

## Research Article

# A Robust and Accurate Quasi-Monte Carlo Algorithm for Estimating Eigenvalue of Homogeneous Integral Equations

F. Mehrdoust, B. Fathi Vajargah, and E. Radmoghaddam

Department of Applied Mathematics, Faculty of Mathematical Science, University of Guilan, P.O. Box 1945, Rasht, Iran

Correspondence should be addressed to F. Mehrdoust; [fmehrdoust@guilan.ac.ir](mailto:fmehrdoust@guilan.ac.ir)

Received 17 August 2013; Accepted 12 September 2013

Academic Editors: D. S. Corti, Y. Peng, and H. Richter

Copyright © 2013 F. Mehrdoust et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present an efficient numerical algorithm for computing the eigenvalue of the linear homogeneous integral equations. The proposed algorithm is based on antithetic Monte Carlo algorithm and a low-discrepancy sequence, namely, Faure sequence. To reduce the computational time we reduce the variance by using the antithetic variance reduction procedure. Numerical results show that our scheme is robust and accurate.

## 1. Introduction

In Monte Carlo (MC) methods the random variables are simulated by computer generated pseudorandom sequences, and the numerical solution is performed by averaging over a large number of simulations. Pseudorandom numbers mimic the realizations of independent identically distributed random variables. Due to a fundamental result in probability theory, Monte Carlo methods based on such pseudorandom numbers can converge only proportionally to the square root of the number of simulations. It was noticed that the randomness of the simulating sequence is not essential for numerical integration. Deterministic sequences which fill the space uniformly can also be used. One of the most powerful methods for improving Monte Carlo simulation is to use low-discrepancy numbers [1]. Therefore, instead of using random numbers, we can employ a deterministic sequence of numbers. Such a sequence is called a low-discrepancy sequence. The great advantage of low-discrepancy sequences is that their rate of convergence is  $O(N^{-1})$  rather than  $O(N^{-1/2})$ , which greatly increases the competitiveness of Monte Carlo simulations. Using low-discrepancy sequences to carry out Monte Carlo algorithm is sometimes called quasi-Monte Carlo (QMC) algorithm [1]. The Faure sequence is one of the well-known quasirandom sequences used in quasi-Monte Carlo applications. In [2, 3] details of Faure sequence and its various scrambling methods are described.

The problem of using Monte Carlo and quasi-Monte Carlo methods for finding an eigenpair of matrices has been extensively studied [4]. In [5] a numerical method for finding the eigenvalue of the linear homogenous integral equation is proposed by applying antithetic Monte Carlo method to obtain a reliable scheme.

In this paper we present a new numerical method based on antithetic variance reduction method and quasi-Monte Carlo algorithm on scrambled Faure sequence for estimating the eigenvalue of the following linear homogenous integral equation:

$$u(x) = \lambda \int_D k(x, y) u(y) dy, \quad \lambda \neq 0, \quad x \in D. \quad (1)$$

It can be written in the form of

$$u(x) = \lambda K u(x), \quad \lambda \neq 0. \quad (2)$$

If  $u \neq 0$ , then  $\lambda$  and  $u$  are called the eigenvalue and eigenfunction of the above equations, respectively. Also,  $K$  is called the kernel of this integral equation. Throughout this paper, we suppose that the kernel  $K$  is symmetric and positive definite; that is,

$$K(x, y) = K(y, x), \quad \forall x, y \in D. \quad (3)$$

And  $\langle K\psi, \psi \rangle > 0$ , for all  $\psi \neq 0$ .

Now, we consider the following continuous Markov Chain  $T_k$  with length,  $k$ :

$$x_0 \longrightarrow x_1 \longrightarrow \cdots \longrightarrow x_k, \quad (4)$$

where  $x_i \in D \subseteq \mathbb{R}$ ,  $i = 1, 2, \dots, k$ . The statistical nature of constructing the chain (4) is  $p(x_0)$  and  $p(x, y)$ , which are the probability of starting chain from  $x_0 \in D$  and transition probability from state  $x \in D$  to  $y \in D$ , respectively [1].

**Theorem 1.** *Under the above assumptions, the random variable*

$$\Theta_n(h) = \frac{h(x_0)}{p(x_0)} W_n \psi(x_n), \quad (5)$$

where

$$W_n = W_{n-1} \frac{K(x_{n-1}, x_n)}{p(x_{n-1}, x_n)}, \quad W_0 = 1, \quad (6)$$

is unbiased estimator for  $\langle K^n \psi, h \rangle$ ; that is,

$$E[\Theta_n(h)] = \langle K^n \psi, h \rangle. \quad (7)$$

*Proof.* See [5]. □

**Theorem 2.** *Under the above conditions, one has*

$$\lambda = \lim_{n \rightarrow \infty} \frac{E[\Theta_n(h)]}{E[\Theta_{n+1}(h)]} = \lim_{n \rightarrow \infty} \frac{\langle K^n \psi, h \rangle}{\langle K^{n+1} \psi, h \rangle}. \quad (8)$$

*Proof.* See [5]. □

## 2. Monte Carlo and Quasi-Monte Carlo Methods

The Monte Carlo method was invented by Stanislaw Ulam, a Polish born mathematician who worked for John von Neumann on the United States' Manhattan Project during World War II, in 1946 [6]. Monte Carlo methods are based on the simulation of stochastic processes whose expected values are equal to computationally interesting quantities. MC methods offer simplicity of construction and are often designed to mirror some process whose behavior is only understood in a statistical sense. However, there are wide classes of problems where MC methods are the only known computational method of solution. Despite the universality of MC methods, a serious drawback is their slow convergence, which is based on the  $O(N^{-1/2})$  behavior of the size of statistical sampling errors. One generic approach to improve the convergence of MC methods has been the use of highly uniform random numbers in place of the usual pseudorandom numbers. While pseudorandom numbers are constructed to mimic the behavior of truly random numbers, these highly uniform numbers, called quasirandom numbers (or low-discrepancy sequences), are constructed to be as evenly distributed as mathematically possible. The use of quasirandom numbers in MC leads to quasi-Monte Carlo methods [7, 8]. Quasi-Monte Carlo methods are a variant of ordinary Monte

Carlo methods that employ highly uniform quasirandom numbers in place of Monte Carlo's pseudorandom numbers. QMC methods can be viewed as deterministic versions of Monte Carlo methods [7]. Determinism enters in two ways: by working with deterministic points rather than random samples and by the availability of deterministic error bounds instead of probabilistic MC error bounds. It could be argued that most practical implementations of MC methods are, in fact, quasi-Monte Carlo methods since the purportedly random samples that are used in Monte Carlo calculation are often generated in the computer by the deterministic algorithm. In QMC methods deterministic nodes are selected in such a way that the error bound is as small as possible. The very nature of the QMC methods with its completely deterministic procedures implies that we get deterministic and thus guaranteed error bounds [7]. In principle, it is therefore always possible to determine in advance an integration rule that yields a given accuracy. For example, for MC it is necessary to increase the number of simulations 100 times to reduce the error by a factor of 10, whereas the QMC requires less (in general much less) than 100 times, and only 10 times in optimal cases. Monte Carlo methods offer statistical error estimates; however, while quasi-Monte Carlo has a faster convergence rate than normal Monte Carlo, one cannot obtain error estimates from quasi-Monte Carlo sample values by any practical way. A recently proposed method, called randomized quasi-Monte Carlo method, takes advantage of MC and QMC methods. Randomness can be brought to bear on quasirandom sequences through scrambling and other related randomization techniques in randomized QMC methods, which provide an elegant approach to obtain error estimates for quasi-Monte Carlo based on treating each scrambled sequence as a different and independent random sample. The core of randomized quasi-Monte Carlo method is to find an effective and fast algorithm to scramble (randomize) quasirandom sequences. Besides obtaining error estimates for quasi-Monte Carlo, scrambling techniques provide a natural way to parallelize quasirandom sequences. This scheme is especially suitable for distributed or grid computing. By scrambling a quasirandom sequence we can produce a family of related quasirandom sequences.

## 3. Quasirandom or Low-Discrepancy Sequences

Quasirandom numbers are constructed to minimize a measure of their deviation from uniformity called discrepancy. A quasirandom sequence is normally generated in the unit  $s$ -dimensional hypercube,  $I^s = [0, 1]^s$ , and attempts to fill the hypercube as uniformly as possible. The original construction of quasirandom sequences was related to the van der Corput sequence, which is a one-dimension quasirandom sequence based on digital inversion. This digital inversion method is a central idea behind the construction of many current quasirandom sequences in arbitrary bases and dimensions. Following that, Halton [9] generalized the van der Corput sequence to  $s$  dimensions, and Sobol [10] introduced

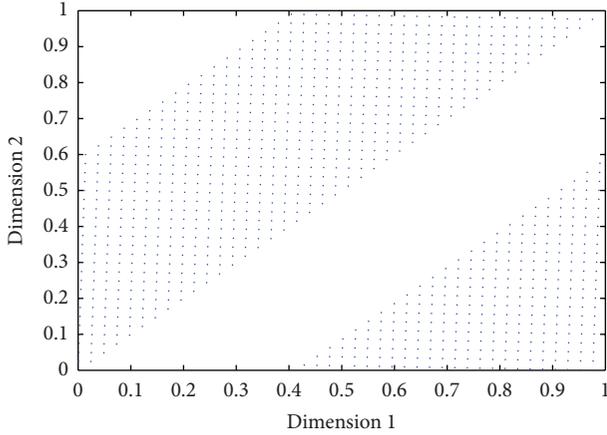


FIGURE 1: Projection of the first 1024 points from a 40-dimensional Faure sequence in base 41.

the sequences that now bear his name. As significant generalization of these methods was proposed by Faure [6]. Niederreiter [7] generalized the existing construction of the Sobol and Faure sequences to arbitrary bases. These are now called Niederreiter sequences. Tezuka [11] further generalized Niederreiter sequences by using the polynomial arithmetic analogue of Halton sequences. The construction of quasirandom sequences is based on minimizing their discrepancy. Quasirandom sequences aim to have the fraction of their points within any subset  $J(\nu) = [0, \nu_1) \times [0, \nu_2) \times \dots \times [0, \nu_s)$  as close as possible to the subset's volume fraction. Based on stardiscrepancy, we expose that the following definition of a low-discrepancy sequence in  $[0, 1)^s$  is as follows.

*Definition 3.* For any  $N > 1$  and sequence  $\{x_i\}$ , let  $\{x_i\}_{1 \leq i \leq N}$  denote the first  $N$  points of the sequence  $\{x_i\}$ . If one has

$$D_N^* \leq C_s \frac{(\log N)^s}{N}, \quad (9)$$

where the constant  $c_s$  depends only on the dimension  $s$ , then the sequence  $\{x_i\}$  is called a low-discrepancy sequence.

There are some variations for random and quasirandom sequences. Tezuka formulated the generalized Faure sequence, based on Halton sequence but using polynomials for reordering. According to Traub, generalized Faure method converges significantly faster than Monte Carlo and simple Faure method. In [2] we represent two methods that mix Faure sequence with the best scrambling Faure sequence and called them MFaure and M2Faure sequences. Figures 1, 2, and 3 represent Faure, MFaure, and M2Faure sequences for dimension 40. Figure 1 illustrates a disadvantage of the Faure sequence: the Faure construction leads to a sequence that has correlations between its individual coordinates. This leads, among others, to bad two-dimensional projections and also has its consequences when the sequence is used for numerical integration. We must stress that its performance in low dimensions is better. In Figures 2 and 3, we see the advantages of the MFaure and M2Faure sequences [3]. In this paper, we employ M2Faure sequence in the proposed algorithm.

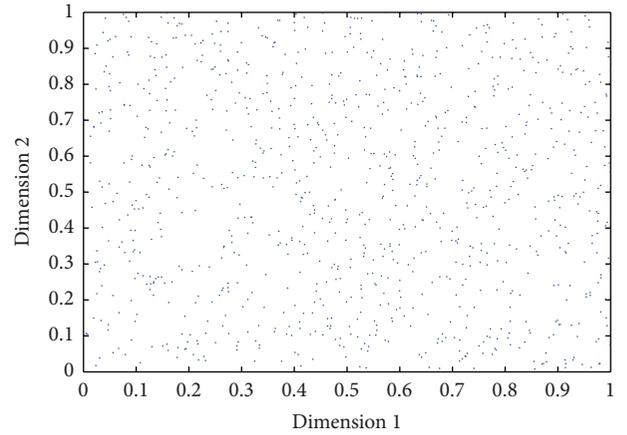


FIGURE 2: Projection of the first 1024 points from a 40-dimensional MFaure sequence in base 41.

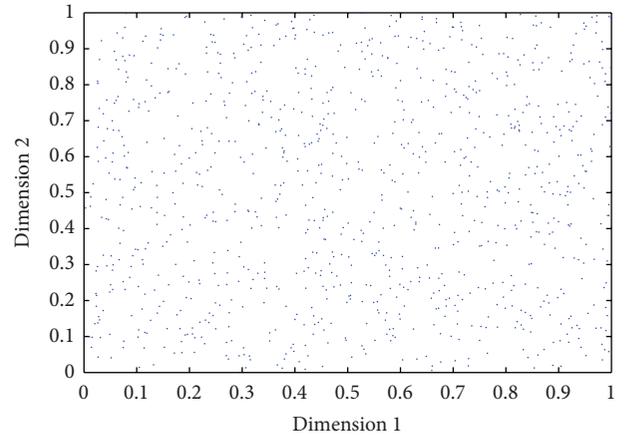


FIGURE 3: Projection of the first 1024 points from a 40-dimensional M2Faure sequence in base 41.

#### 4. Variance Reduction Techniques

As Monte Carlo simulation is slow to converge and there are various methods for increasing the speed of convergence, this essentially comes down to simulate the estimator in such a way that their variance is reduced. One simple method is antithetic sampling. With antithetic sampling one draws samples in pairs. If  $U$  is uniformly distributed, then so is  $1 - U$ . For every random, draw two paths that are therefore simulated: one with  $U$  and one with  $1 - U$ . This ensures that the mean of the drawn paths is zero, and the symmetry of the normal distribution is achieved by construction [12].

Assume that we want to compute  $E[f(X)]$  with  $X$  a random variable uniformly distributed on  $[0, 1]$ . The Naive Monte Carlo (NMC) is as follows:

$$\bar{f}(X) = \frac{1}{N} \sum_{i=1}^N f(X_i), \quad (10)$$

where  $X_i$ ,  $i = 1, \dots, N$ , are independent copies of  $X$ . In the antithetic variates (AV) method we use the numbers

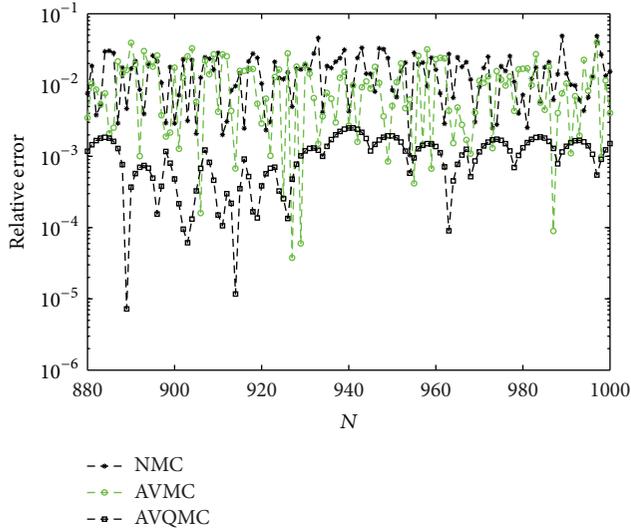


FIGURE 4: Relative error against the number of sample points with  $N = 1000$  for Example 4.

$1 - X_1, \dots, 1 - X_N$  and introduce the antithetic Monte Carlo estimator

$$\bar{f}_{AV}(X) = \frac{1}{2N} \left[ \sum_{i=1}^N f(X_i) + f(1 - X_i) \right]. \quad (11)$$

We note that the AV estimator is unbiased [13]. Suppose that  $\sigma^2 = \text{var}(f(X))$ . Then the variance of AV estimator is given by

$$\bar{f}_{AV}(X) = \frac{\sigma^2}{2N} + \frac{1}{2N} \text{cov}(f(X), f(1 - X)). \quad (12)$$

If  $f(X)$  and  $f(1 - X)$  are negatively correlated, then we have a reduction of the variance compared to Naive Monte Carlo estimator based on  $2N$  random numbers.

## 5. The Proposed Algorithm (AVQMC Algorithm)

- (1) Choose a value of Markov chain length  $n$  and a value of number of Markov chain  $N$ .
- (2) Create a continuous quasirandom walk (Markov chain)  $x_0 \rightarrow x_1 \rightarrow \dots \rightarrow x_n$ , where  $x_i$  for  $i = 0, \dots, n$  are elements of M2Faure sequence.
- (3) For each quasirandom walk  $s = 1, \dots, N$  simulate the following random variable:

$$\Theta_n^{(s,1)}(h) = \frac{h(x_0^{(s)})}{P(x_0^{(s)})} W_n^{(s)} \psi(x_n^{(s)}). \quad (13)$$

(3.1) Create  $y_0 \rightarrow y_1 \rightarrow \dots \rightarrow y_n$  such that  $y_i = 1 - x_i$  for  $i = 0, \dots, n$ .

(3.2) Set the following random variable:

$$\Theta_n^{(s,2)}(h) = \frac{h(y_0^{(s)})}{P(y_0^{(s)})} W_n^{(s)} \psi(y_n^{(s)}). \quad (14)$$

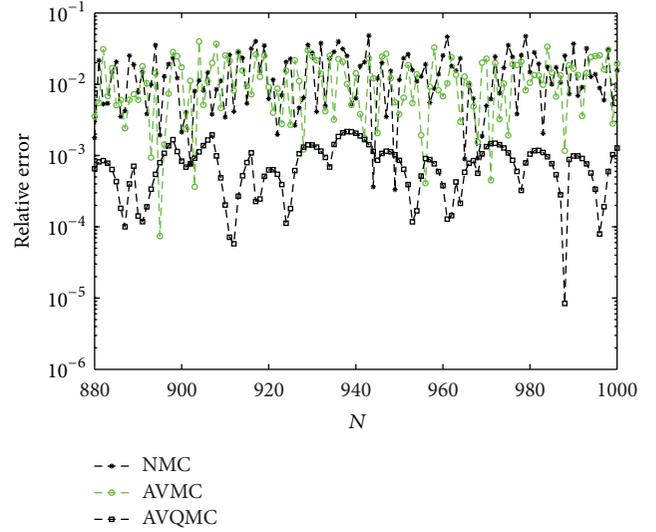


FIGURE 5: Relative error against the number of sample points with  $N = 1000$  for Example 5.

- (4) Obtain the quasi-Monte Carlo estimation

$$\widehat{\Theta}_n = \frac{1}{2N} \left\{ \sum_{s=1}^N \Theta_n^{(s,1)}(h) + \sum_{s=1}^N \Theta_n^{(s,2)}(h) \right\}. \quad (15)$$

- (5) Print  $\widehat{\Theta}_n \approx \lambda$ .

- (6) End of algorithm.

## 6. Simulation Results

In this section we compare three approaches of Monte Carlo methods, namely, Naive Monte Carlo (NMC), antithetic variates Monte Carlo (AVMC) [13], and the proposed algorithm (AVQMC) methods. We run algorithms for evaluating the eigenvalue of the following examples. We compare the relative error of AVQMC and AVMC algorithms and the Naive Monte Carlo (NMC) algorithm for each example. Figures 4 and 5 illustrate the relative error against number of sample sizes with  $N = 1000$  for Examples 4 and 5, respectively. All figures show that, when the sample size is small, AVQMC and AVMC algorithms are better than NMC algorithm. Also, when the sample size is large, AVQMC is better than all, and another advantage is that relative errors of method with Faure sequence are deterministic. Also, we obtain the estimation of the eigenvalue, with standard deviation (STD) estimation for different values of the number of Markov chains ( $N$ ). For each example the Monte Carlo simulation result is summarized in a figure. The numerical results in Figures 6 and 7 confirm that the AVQMC algorithm is a reliable stochastic algorithm for computing the eigenvalue of the homogeneous linear integral equations. Also, we outline the standard deviation (STD) of the unbiased Monte Carlo estimator, the approximated eigenvalue ( $\bar{\lambda}$ ), and the relative errors of three methods for Examples 4 and 5 in Table 1.

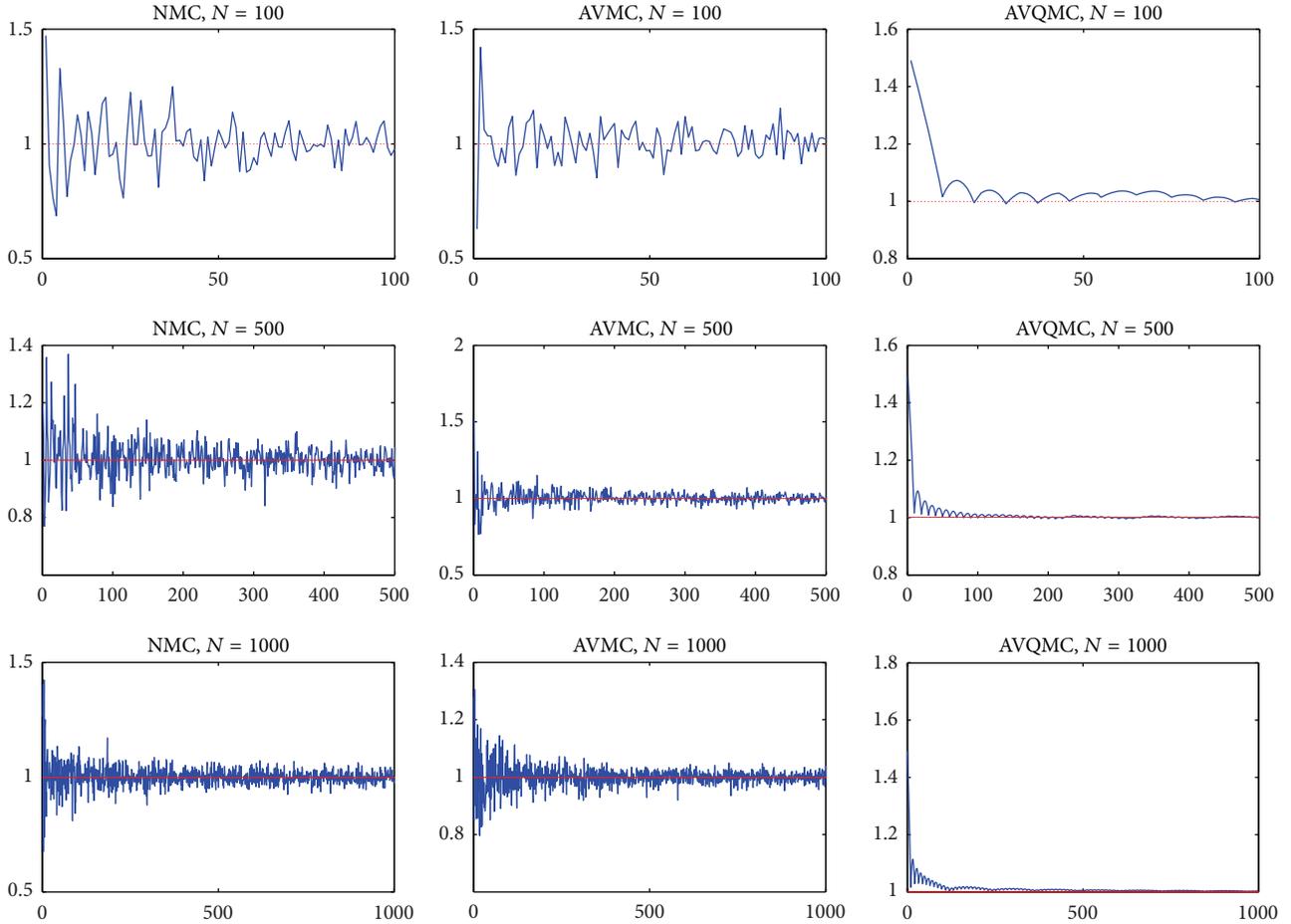


FIGURE 6: Comparison between NMC, AVMC, and AVQMC methods regarding Example 4.

TABLE 1: Relatives errors, STD, and  $\hat{\lambda}$ .

Method	Number of simulation	Example 4			Example 5		
		STD	$\hat{\lambda}$	Relative error	STD	$\hat{\lambda}$	Relative error
NMC	$N = 100$	0.1060	1.1526	0.1526	0.1307	1.3568	0.0656
AVMC	$N = 100$	0.0743	1.0446	0.0446	0.0996	1.3424	0.0543
AVQMC	$N = 100$	0.0885	1.0055	0.0055	0.1486	1.2733	$4.5863 \times 10^{-5}$
NMC	$N = 500$	0.0579	1.0242	0.0242	0.0856	1.2997	0.0208
AVMC	$N = 500$	0.0390	1.0110	0.0110	0.0554	1.2632	0.0079
AVQMC	$N = 500$	0.0439	1.0010	0.0010	0.0681	1.2752	0.0016
NMC	$N = 1000$	0.0447	0.9541	0.0459	0.1307	1.2972	0.0188
AVMC	$N = 1000$	0.0743	0.9934	0.0066	0.0931	1.2673	0.0046
AVQMC	$N = 1000$	0.0324	1.0015	0.0015	0.0497	1.2749	0.0013

*Example 4.* Consider equation (2) with kernel  $K(x, y) = e^{x-y}$ ,  $D = [0, 1]$ . We can easily show that the exact eigenvalue is 1.

*Example 5.* Consider equation (2) with kernel  $K(x, y) = \sin(x) \sin(y)$ ,  $D = [0, \pi/2]$ . We can easily show that the exact eigenvalue is  $4/\pi = 1.2732$ .

## 7. Conclusion

In this paper we have proposed an efficient algorithm and analyzed the performance of the algorithm for computing the eigenvalue of the linear homogenous integral equations. We studied the accuracy and robustness of the proposed algorithm. The numerical results in Table 1 show that AVQMC

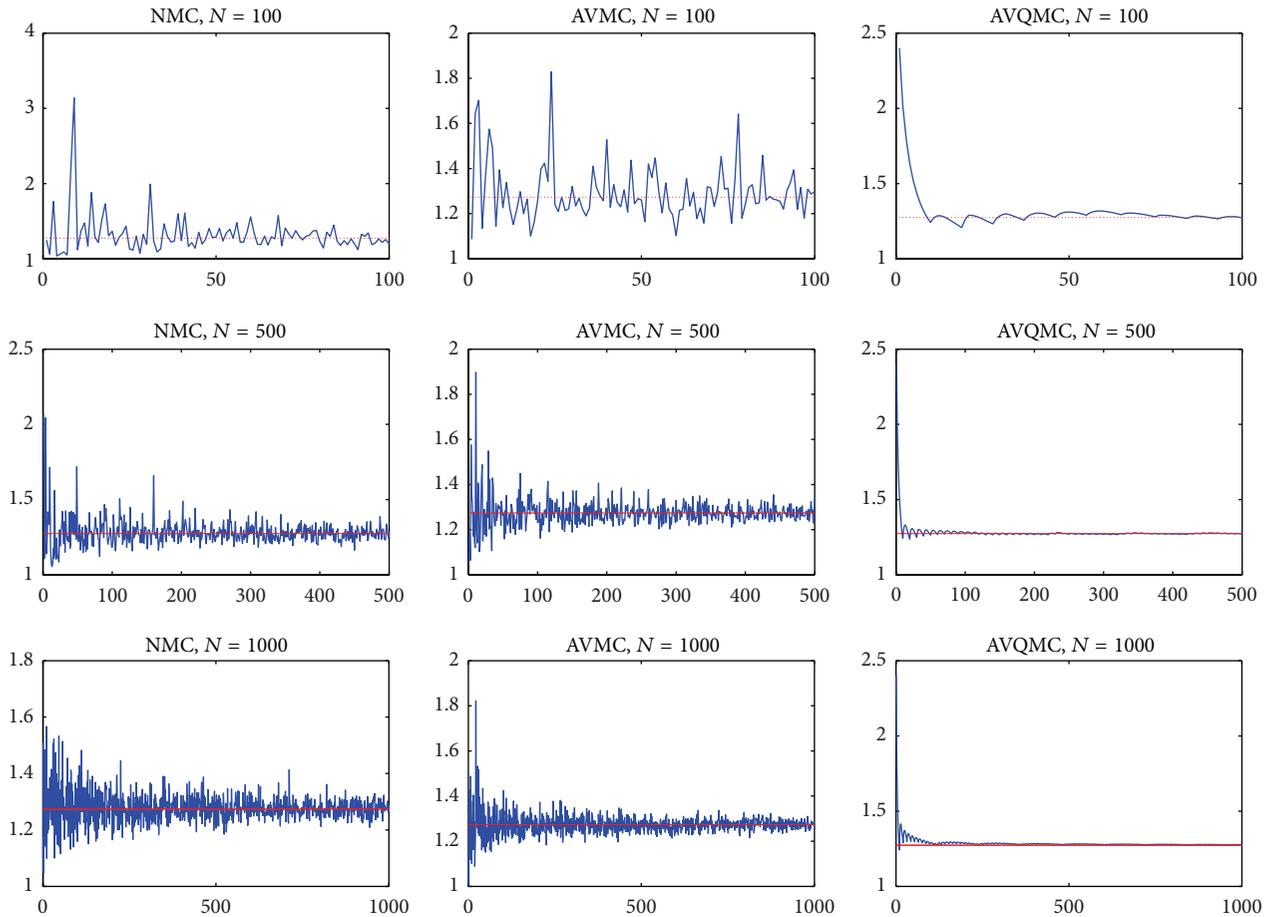


FIGURE 7: Comparison between NMC, AVMC, and AVQMC methods regarding Example 5.

algorithm has better improvement, in both error and convergence, than AVMC and NMC algorithms.

## References

- [1] I. T. Dimov and T. V. Gurov, "Monte Carlo algorithm for solving integral equations with polynomial nonlinearity: parallel implementation," *Pliska Studia Mathematica Bulgarica*, vol. 13, pp. 117–132, 2000.
- [2] B. Fathi and E. Radmoghaddam, "Optimal Faure sequence via mix Faure with the best scrambling schemes," *Computer Science and Its Applications*, vol. 2, no. 1, pp. 2166–2924, 2012.
- [3] B. Vandewoestyne, H. Chi, and R. Cools, "Computational investigations of scrambled faure sequences," *Mathematics and Computers in Simulation*, vol. 81, no. 3, pp. 522–535, 2010.
- [4] B. Fathi and F. Mehrdoust, "Quasi Monte Carlo algorithm for computing smallest and largest generalized eigenvalues," *ANZIAM Journal E*, vol. 52, pp. 41–58, 2011.
- [5] F. Mehrdoust and B. Fathi, "A reliable stochastic algorithm for estimating eigenvalue of homogeneous integral equations," *Journal of Advanced Research in Applied Mathematics*, vol. 5, no. 2, p. 13, 2013.
- [6] H. Faure, "Discrepancy of sequences associated with a number system (in dimension  $s$ )," *Acta Arithmetica*, vol. 41, no. 4, pp. 337–351, 1982 (French).
- [7] H. Niederreiter, *Random Number Generations and Quasi-Monte Carlo Methods*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 1992.
- [8] J. Spanier and E. H. Maize, "Quasi-random methods for estimating integrals using relatively small samples," *SIAM Review*, vol. 36, no. 1, pp. 18–44, 1994.
- [9] F. J. Hickernell, "Mean square discrepancy of randomized nets," *ACM Transactions on Modeling and Computer Simulation*, vol. 6, pp. 274–296, 1996.
- [10] I. M. Sobol, "Uniformly distributed sequences with an additional uniform property," *USSR Computational Mathematics and Mathematical Physics*, vol. 16, no. 5, pp. 236–242, 1976.
- [11] S. Tezuka, "Polynomial arithmetic analogue of halton sequences," *ACM Transactions on Modeling and Computer Simulation*, vol. 3, no. 2, pp. 99–107, 1993.
- [12] M. S. Joshi, *The Concepts and Practice of Mathematical Finance*, Cambridge University Press, 2nd edition, 2003.
- [13] H. Chi, *Scrambled Quasi-Random Sequences and their Application*, The Florida state university college of arts and science, 2004.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

