

# Quasi Monte Carlo algorithm for computing smallest and largest generalised eigenvalues

Behrouz Fathi Vajargah<sup>1</sup>

Farshid Mehrdoust<sup>2</sup>

(Received 27 August 2010; revised 3 March 2011)

## Abstract

The problem of obtaining the smallest and the largest generalised eigenvalues using quasi Monte Carlo algorithm is considered. We first study the results of Dimov and others using three algorithms based on the power method combined with Monte Carlo and quasi Monte Carlo methods for evaluating extremal eigenvalue of real matrices. We present a quasi Monte Carlo algorithm for computing both the smallest and the largest generalised eigenvalues using Sobol, Halton sequences and the rand function in Matlab. We finally compare the efficiency of three employed generators in our algorithm for different pencils.

## Contents

### 1 Introduction

E42

---

<http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/3437> gives this article, © Austral. Mathematical Soc. 2011. Published March 25, 2011. ISSN 1446-8735. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to this URL for this article.

<i>1</i>	<i>Introduction</i>	E42
<b>2</b>	<b>Pseudorandom and quasi-random sequences</b>	<b>E44</b>
2.1	The Halton sequence . . . . .	E44
2.2	The Sobol sequence . . . . .	E45
<b>3</b>	<b>Resolvent quasi Monte Carlo method</b>	<b>E45</b>
<b>4</b>	<b>Numerical results</b>	<b>E49</b>
<b>5</b>	<b>Concluding remarks</b>	<b>E55</b>
	<b>References</b>	<b>E57</b>

# 1 Introduction

Finding eigenvalues of a given matrix is a necessary task in scientific computations. It is used in many scientific procedures such as in discriminant analysis, numerical analysis, variance analysis, and principal component analysis. When analytical or deterministic methods cannot give solutions for this problem, Monte Carlo methods are legitimate alternative. A primary reason for this is that in Monte Carlo and quasi Monte Carlo methods only  $\mathcal{O}(Nl)$  steps are required to find a desired eigenvalue of a given matrix, where  $N$  is the number of Markov chains and  $l$  is an estimate of the chain length in the stochastic process, which are independent of matrix size  $n$ . Secondly, the algorithms for these methods are inherently parallel [3].

Monte Carlo methods based on the simulation of stochastic process whose expectations are equal to computationally interesting quantities. These methods give statistical estimates for the functional of the solution by performing random sampling of a certain random variable such that its expectation is the desired functional [1, 4, 5]. Convergence of Monte Carlo methods is often improved by replacing pseudorandom numbers with more uniformly distributed numbers known as quasi-random numbers. Quasi Monte Carlo methods are an alternative approach based on the idea that random Monte

Carlo techniques are often improved by replacing the underlying source of random numbers with a more uniformly distributed deterministic sequence. In other words, quasi Monte Carlo methods use quasi-random or low discrepancy sequences instead of pseudorandom sequences [7, 8, 9, 10].

Recently, evaluating the dominant eigenvalue of large real sparse matrices on a cluster of workstation using MPI and the Monte Carlo and quasi Monte Carlo methods for obtaining the smallest and largest eigenvalue has been presented [3, 4, 5, 8]. Here, we introduce a quasi Monte Carlo algorithm for computing the smallest and the largest generalised eigenvalue. We first studied previous works [3, 4, 5, 8] and then improved these ideas for generalised eigenvalue based on calculating multiple integrals.

Let  $A, B \in \mathbb{R}^{n \times n}$  be real symmetric matrices, also suppose that the matrix  $B$  is a positive definite matrix. Consider the problem of evaluating the eigenvalues of the pencil  $(A, B)$ , that is, the values for which

$$Ax = \lambda Bx. \quad (1)$$

A generalised eigenvalue problem (1) is said to be symmetric positive definite (SPD) when  $A$  is symmetric and  $B$  is positive definite. We assume these conditions throughout the rest of this paper. This nonzero  $x$  is said to be an eigenvector of (1) and the corresponding value of  $\lambda$  is called an eigenvalue. For simplicity,  $(\lambda, x)$  is called an eigenpair of (1). Problem (1) includes the standard eigenvalue problem when  $B = I_n$ .

Frequently scientists wish to know some eigenvalues of very large matrices. For such problems, Krylov subspace methods are well known. The Lanczos algorithm is a Krylov subspace method for symmetric problems. For non-symmetric matrices, the methods are Arnoldi and non-symmetric Lanczos [6]. An alternative approach which is more useful in the context of large scale matrices are Monte Carlo and quasi Monte Carlo algorithms. These algorithms are almost independent of the dimension of matrix [1, 3, 4].

Using the the quasi Monte Carlo algorithm we develop further recent ideas [4, 5, 8]. Section 2 briefly reviews the pseudorandom and quasi-random sequences

(that is Halton and Sobol sequences). Section 3 presents the resolvent quasi Monte Carlo method used to the smallest and the largest generalised eigenvalues and Section 4 gives numerical results that confirm the applicability of proposed algorithm. Section 5 concludes and discusses our research.

## 2 Pseudorandom and quasi-random sequences

Quasi Monte Carlo methods are succinctly described as deterministic versions of Monte Carlo methods. Determinism enters in two ways: by working with deterministic points rather than random samples; and by the availability of deterministic error bounds instead of the probabilistic Monte Carlo error bounds. The connections between quasi Monte Carlo methods and uniform pseudorandom numbers arise in the theoretical analysis of various methods for the generation of uniform pseudorandom numbers. The pseudorandom sequences simulate random samples from a  $\mathbf{U}(0, 1)$  distribution and quasi-random sequences correspond to samples from a  $\mathbf{U}(0, 1)$  distribution. One can find several quasi-random sequences, such as the Halton sequences, Sobol sequences, Faure sequences, and Niederreiter sequences [7, 9, 10].

### 2.1 The Halton sequence

One of the most important low discrepancy sequences, used by many researchers, is the Halton sequence. Its definition is based on the radical inverse function

$$\Phi_{\mathbf{p}} = \frac{\mathbf{b}_0}{\mathbf{p}} + \frac{\mathbf{b}_1}{\mathbf{p}^2} + \cdots + \frac{\mathbf{b}_m}{\mathbf{p}^{m+1}}, \quad (2)$$

where  $\mathbf{p}$  is a prime number, and  $\mathbf{n} = \mathbf{b}_0 + \mathbf{b}_1\mathbf{p} + \cdots + \mathbf{b}_m\mathbf{p}^m$ , with integer  $0 \leq \mathbf{b}_j < \mathbf{p}$ . The Halton sequence,  $\mathbf{X}_{\mathbf{n}}$ , in  $s$ -dimensional is then defined as

$$\mathbf{X}_{\mathbf{n}} = (\Phi_{\mathbf{p}_1}, \Phi_{\mathbf{p}_2}, \dots, \Phi_{\mathbf{p}_s}), \quad \mathbf{n} = 0, 1, \dots, \quad (3)$$

where the integer numbers  $p_1, \dots, p_s$  are greater than one and pairwise prime. Most of the times, they are chosen as the first  $s$  primes. We use optimal Halton sequences [2].

## 2.2 The Sobol sequence

The Sobol sequence is defined in base 2. To generate the  $j$ th component of the points in a Sobol sequence, we need to choose a primitive polynomial of some degree  $s_j$  in the field  $Z_2$ :

$$x^{s_j} + a_{1,j}x^{s_j-1} + \dots + a_{s_j-1,j}x + 1, \quad (4)$$

where coefficients  $a_{1,j}, \dots, a_{s_j-1,j} \in \{0, 1\}$ .

The  $j$ th component of the  $i$ th point in a Sobol sequence, is

$$x_{ij} = i_1 v_{1,j} \oplus i_2 v_{2,j} \oplus \dots, \quad (5)$$

where  $v_{k,j} = m_{k,j}/2^k$ , and the sequence of positive integer  $\{m_{1,j}, m_{2,j}, \dots\}$  are defined by the recurrence relation

$$m_{k,j} = 2a_{1,j}m_{k-1,j} \oplus 2^2a_{2,j}m_{k-2,j} \oplus \dots \oplus 2^{s_j}m_{k-s_j,j} \oplus m_{k-s_j,j}, \quad (6)$$

where  $\oplus$  is the bit-by-bit exclusive-or operator and  $i_k$  is the  $k$ th digit from the right when  $i$  is written binary  $i = (\dots i_3 i_2 i_1)$  [10].

## 3 Resolvent quasi Monte Carlo method

Here, we describe some results on quasi Monte Carlo methods based on resolvent matrix and then extend that for evaluating the generalised eigenvalue problem. We want to solve (1): since  $B$  is symmetric and positive definite, then it admits the Schur decomposition, that is  $u^T B u = D = \text{diag}(d_1, \dots, d_n)$  and

let  $L = uD^{1/2} = \text{diag}(\sqrt{d_1}, \dots, \sqrt{d_n})$ , where  $L$  is a lower triangular matrix. So (1) is converted to

$$(L^{-1}AL^{-T})(L^T x) = \lambda(L^T x).$$

Hence any eigenvalue  $\lambda$  satisfying equation (1) must be the eigenvalue of the symmetric matrix  $R = L^{-1}AL^{-T}$ , and the eigenvectors of which take the form  $y = L^T x$ . Now we consider computing the eigenvalues of the equation

$$R x = \lambda x \tag{7}$$

where the matrix  $R$  is symmetric and therefore the value of  $\lambda$  is real. Let us order all the eigenvalues of  $R$  in descending order

$$\lambda_{\max} = \lambda_1 > \lambda_2 \geq \dots > \lambda_n = \lambda_{\min}.$$

Now, consider an algorithm based on Monte Carlo iterations by the matrix  $R$  resolvent operator  $R_q = [I - qR]^{-1}$ , where  $q$  is a parameter chosen such that  $|q| < 1/\|R\|$ .

We have

$$R_q^m = [I - qR]^{-m} = \sum_{i=0}^{\infty} q^i C_{m+i-1}^i R^i. \tag{8}$$

The eigenvalues of the operators  $[I - qR]^{-1}$  and  $R$  are connected with the equality

$$\mu = \frac{1}{1 - q\lambda}.$$

If  $q > 0$ , then the largest eigenvalue  $R_q$  of the resolvent matrix corresponds to the largest eigenvalue of the matrix  $R$ ; but if  $q < 0$ , then it corresponds to the smallest eigenvalue of the matrix  $R$ . Also, as  $m \rightarrow \infty$  [4],

$$\mu^{(m)} = \frac{([I - qR]^{-m} f, h)}{([I - qR]^{-(m-1)} f, h)} \rightarrow \mu = \frac{1}{1 - q\lambda}, \quad f, h \in \mathbb{R}^n. \tag{9}$$

Similar to the Monte Carlo methods, but in the continuous case, suppose the existence of the following random trajectory (Markov chain)  $T_l$  of length  $l$  starting at state  $x_0$

$$T_l : x_0 \rightarrow x_1 \rightarrow \dots \rightarrow x_l,$$

where  $x_j \in D_j = [j - 1, j)$  means the chosen state, for each  $j = 1, 2, \dots, l$  and  $D = \bigcup_{j=1}^l D_j = [0, n)$ . Assume that

$$p(x) = p_i, \quad x \in G_i, \tag{10}$$

and

$$p(x, y) = p_{ij}, \quad x \in G_i, \quad y \in G_j, \tag{11}$$

are the probability of starting the chain at  $x_0$  and the transition probability from state  $x_i$  to  $y_j$ , respectively.

Now, define the random variable  $W_j^Q$  based on quasi Monte Carlo methods by the recursion equation

$$W_0 = \frac{h(x_0)}{p(x_0)}, \quad W_j^Q = W_{j-1}^Q \frac{r(x_{j-1}, x_j)}{p(x_{j-1}, x_j)}, \quad j = 1, 2, \dots, l. \tag{12}$$

We have [3]

$$E[W_i f_{k_i}] = (A^i f, h), \quad i = 1, 2, \dots \tag{13}$$

We write

$$(R_q^m f, h) = E \left[ \sum_{i=0}^{\infty} q^i C_{m+i-1}^i (R^i f, h) \right], \quad m = 1, 2, \dots \tag{14}$$

Using the Rayleigh quotient [3],

$$\lambda \approx \frac{E \sum_{i=0}^l q^i C_{m+i-1}^i W_{i+1}^Q}{E \sum_{i=0}^l q^i C_{m+i-1}^i W_i^Q}. \tag{15}$$

Therefore we write

$$\lambda \approx \frac{\sum_{i=0}^l q^i C_{m+i-1}^i \int_{D_0} \int_{D_1} \dots \int_{D_i} W_{i+1}^Q dx_0 \dots dx_{i+1}}{\sum_{i=0}^l q^i C_{m+i-1}^i \int_{D_0} \dots \int_{D_i} W_i^Q dx_0 \dots dx_i}. \tag{16}$$

Using some simple calculations,

$$\lambda \approx \frac{\sum_{i=0}^l q^i C_{m+i-1}^i \int_{D_{i+1}} \cdots \int_{D_0} h(x_0) r(x_0, x_1) \cdots r(x_i, x_{i+1}) dx_0 \cdots dx_{i+1}}{\sum_{i=0}^l q^i C_{m+i-1}^i \int_{D_i} \cdots \int_{D_0} h(x_0) r(x_0, x_1) \cdots r(x_{i-1}, x_i) dx_0 \cdots dx_i}. \quad (17)$$

The above integrals are multi-dimensional integrals and therefore we compute an approximation using quasi Monte Carlo integration [3, 7, 9]. The following theorems give the error bound for this approximation. Before stating, it is necessary to explain some preliminaries about discrepancy.

**Definition 1** Let  $S$  be a point set consisting of  $x_1, x_2, \dots, x_N \in [0, 1]^s$ . For an arbitrary subset  $A$  of  $[0, 1]^s$ , we define

$$\#(A; S) = \sum_{n=1}^N \chi_A(x_n)$$

where  $\chi_A$  is the the characteristic function of  $A$ . Suppose that  $A$  is a nonempty family of Lebesgue measurable subsets of  $[0, 1]^s$ . Define the discrepancy of a point set  $S$  by

$$D_N(A; S) = \sup_{A \in \mathcal{A}} \left| \frac{\#(A; S)}{N} - \lambda_s(A) \right|$$

where  $\lambda_s(A)$  denotes the Lebesgue measure of  $A$  in  $\mathbb{R}^n$ .

**Definition 2** The star discrepancy  $D_N^*(A; S)$ , when  $s = 1$  is defined by

$$D_N^* = D_N^*(x_1, \dots, x_N) = \sup_{0 \leq u \leq 1} \left| \frac{\#(A; S)}{N} - u \right|.$$

Two theorems relate error bounds in the quasi Monte Carlo methods.

**Theorem 3 ([7])** If  $f$  has bounded variation  $V(f)$  on  $[0, 1]$ , then for any  $x_1, \dots, x_N \in [0, 1]$ ,

$$\left| \frac{1}{N} \sum_{n=1}^N f(x_n) - \int_0^1 f(x) dx \right| \leq V(f) D_N^*(x_1, \dots, x_N).$$



**Theorem 4 ([7])** Suppose that  $f$  is continuous on  $[0, 1]$ , then for any  $\mathbf{u}, \mathbf{v} \in [0, 1]$  and  $x_1, \dots, x_N \in [0, 1]$ ,

$$\left| \frac{1}{N} \sum_{n=1}^N f(x_n) - \int_0^1 f(x) dx \right| \leq \sup_{|\mathbf{u}-\mathbf{v}| \leq D_N^*} |f(\mathbf{u}) - f(\mathbf{v})|.$$

**Remark 5** Based on the above theorems, let

$$f(x_0, \dots, x_i) = h(x_0)r(x_0, x_1) \cdots r(x_{i-1}, x_i),$$

then

$$\left| \frac{1}{N} \sum_{j=1}^N f(x_0^{(j)}, \dots, x_i^{(j)}) - \int_{D_i} \cdots \int_{D_0} f(x_0, \dots, x_i) dx_0 \cdots dx_i \right| \leq |h| \|R^i\| D_N^*.$$

## 4 Numerical results

We present some numerical results to show the performances of quasi Monte Carlo algorithm in our implementation. We ran our algorithm on a workstation Intel(R) 1.83 GHz Dual CPU, 2.00 GB RAM and MATLAB software 7.6(R2008a). In this software, we employed the function `eig(A,B)` to contrast to our algorithm results. In the first example, we choose the pencil  $(\mathbf{A}, \mathbf{B})_{128 \times 128}$ , where the matrix  $\mathbf{A}$  is ill conditioned, ( $\text{cond}(\mathbf{A}) = 5.7710 \times 10^{14}$ ). Our method is applied for a general matrix of dimension  $1000 \times 1000$  whose elements are generated by a uniformly distributed random number generator, in the second example. Figures 1–3 plot the PRNs and QRNs sequences. Figure 4 compares the relative error of the largest and smallest eigenvalues.

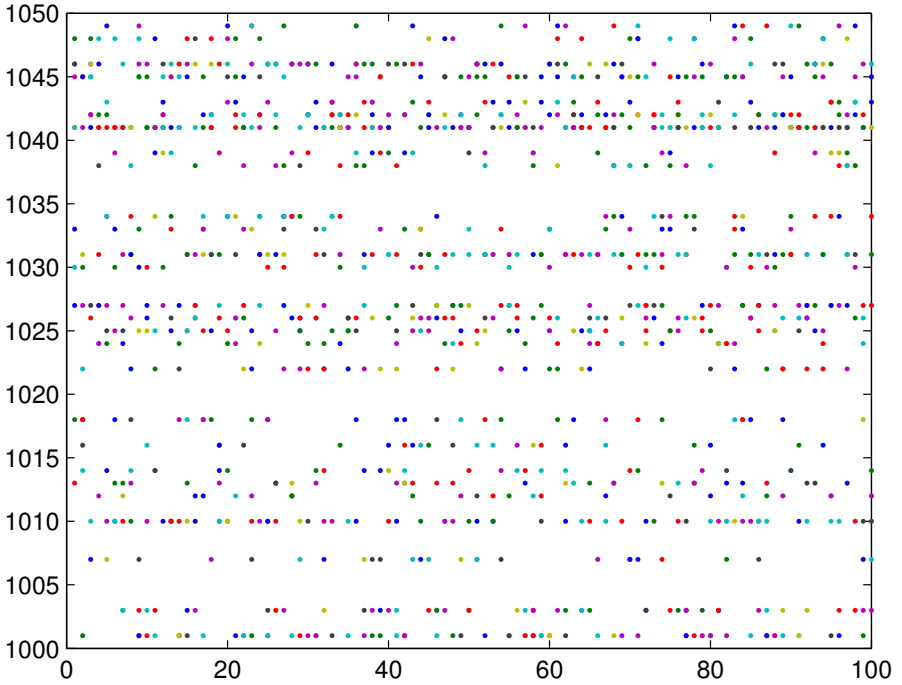


FIGURE 1: Random points generated by rand function,  $n = 5000$ .

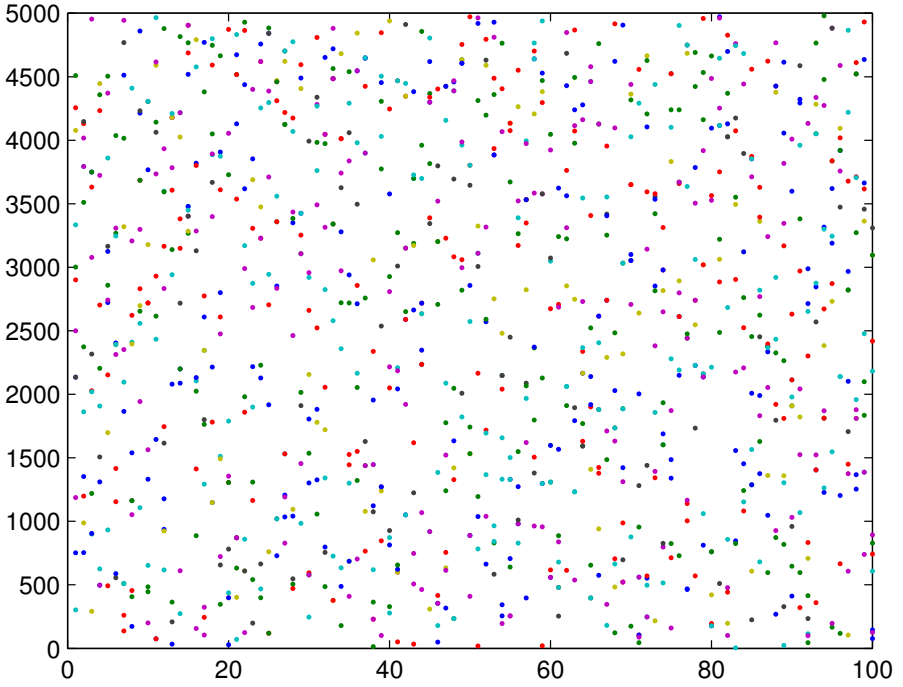


FIGURE 2: Random points generated by Halton sequence,  $n = 5000$ .

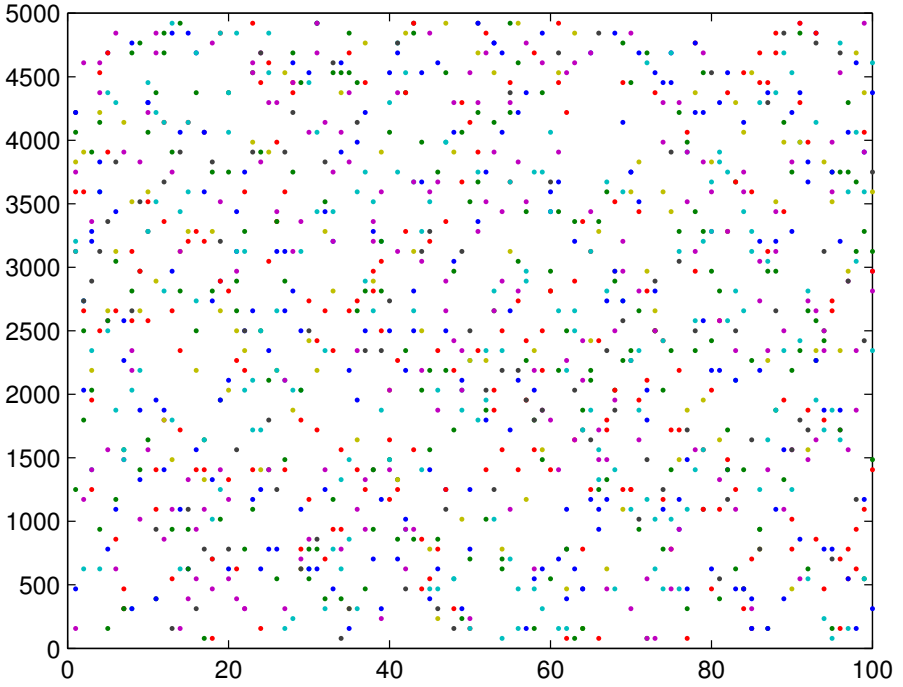


FIGURE 3: Random points generated by Sobol sequence,  $n = 5000$ .

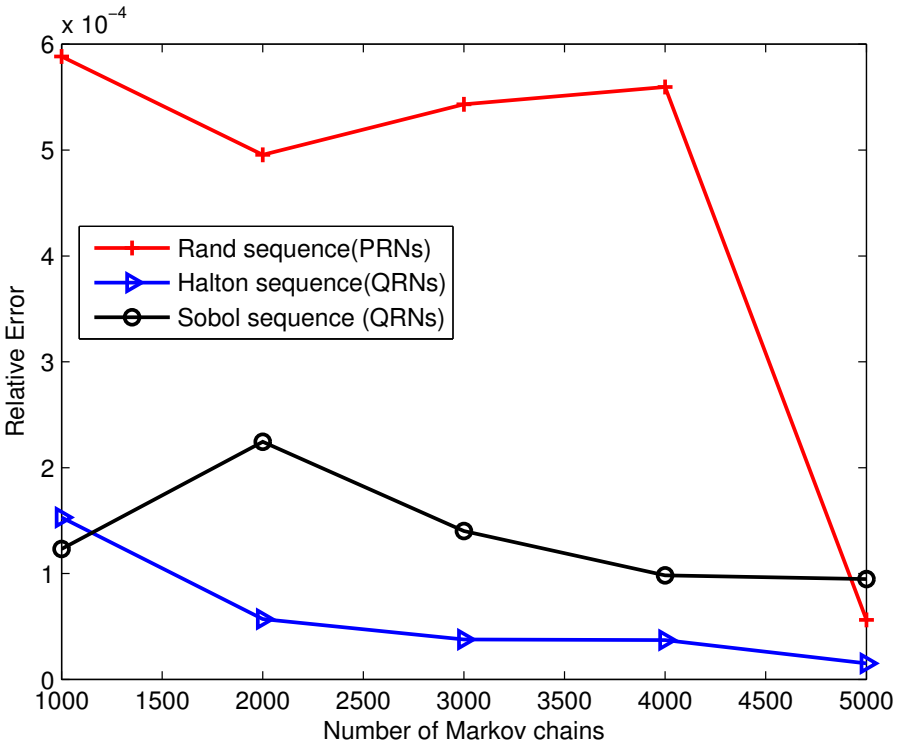


FIGURE 4: Comparison relative error for three cases.

TABLE 1: Numerical results for Example 6.

Random seq.	$\lambda_{\max}$	Error	$\lambda_{\min}$	Error	Time (S)
PRNs-rand	145.1295	0.0089	-384.0085	0.0085	0.07
QRNs-Halton	145.1383	$1.00 \times 10^{-4}$	-384.0041	0.0041	0.09
QRNs-Sobol	145.1378	$6.00 \times 10^{-4}$	-384.0074	0.0074	0.1

TABLE 2: Relative error for computing  $\lambda_{\max}$ .

Number of chains N	Relative error		
	rand sequence	Halton sequence	Sobol sequence
1000	$5.8835 \times 10^{-4}$	$1.5295 \times 10^{-4}$	$1.2315 \times 10^{-4}$
2000	$4.9545 \times 10^{-4}$	$5.7002 \times 10^{-5}$	$2.2448 \times 10^{-4}$
3000	$5.4322 \times 10^{-4}$	$3.7768 \times 10^{-5}$	$1.4016 \times 10^{-4}$
4000	$5.5958 \times 10^{-4}$	$3.7179 \times 10^{-5}$	$9.8268 \times 10^{-5}$
5000	$5.6136 \times 10^{-5}$	$1.5136 \times 10^{-5}$	$9.4723 \times 10^{-5}$

**Example 6** Consider the generalised eigenvalue problem (1), where

$$A = \begin{bmatrix} 3n-1 & -1 & \dots & & -1 \\ -1 & 3n-1 & -1 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ -1 & \dots & -1 & 3n-1 & -1 \\ -1 & \dots & & -1 & 3n-1 \end{bmatrix}_{3n \times 3n},$$

$$B = \begin{bmatrix} C & & & \\ & C & & \\ & & \ddots & \\ & & & C \end{bmatrix}_{3n \times 3n} \quad \text{and} \quad C = \begin{bmatrix} -2 & 1 & 0 \\ 1 & 0 & -2 \\ 0 & -2 & 1 \end{bmatrix}.$$

**Example 7** In Table 2 the relative errors for computing  $\lambda_{\max}$  and  $\lambda_{\min}$  on randomised pencil  $(A, B)_{1000 \times 1000}$  is outlined.

TABLE 3: Results for computing  $\lambda_{\min}$  and  $\lambda_{\min}$ .

Martix dimension n	QRNS-Halton		
	$\lambda_{\max}$	$\lambda_{\min}$	Time (s)
100	$1.5101 \times 10^{-4}$	$0.9300 \times 10^{-3}$	0.023915
200	$1.2108 \times 10^{-4}$	$0.8312 \times 10^{-3}$	0.029173
400	$6.3472 \times 10^{-5}$	$2.7768 \times 10^{-4}$	0.039850
800	$9.4072 \times 10^{-6}$	$3.7179 \times 10^{-5}$	0.083722
1600	$4.5975 \times 10^{-6}$	$1.9136 \times 10^{-5}$	0.153430
3200	$1.2118 \times 10^{-6}$	$1.0516 \times 10^{-5}$	0.332421

**Example 8** In Table 3 the relative errors and computational time using quasi Monte Carlo algorithm (QRNS-Halton) for computing  $\lambda_{\min}$  on randomised pencils  $(A, B)_{n \times n}$  is outlined.

## 5 Concluding remarks

The computational results in Table 1 are obtained based on the rand function in MATLAB software and two Halton and Sobol quasi-random numbers, for  $\lambda_{\max}$  and  $\lambda_{\min}$  of a given matrix. Table 2 shows the relative error of the employed algorithm using various number of Markov chains N. The results in Table 1 and 2 show that the Halton sequence of quasi-random numbers has better improvement, in both the error and convergence, than Sobol quasi-random number and rand random generators. However, the Sobol sequence up to  $N = 3000$  is slightly better than the rand generator, but with increasing the number of Markov chains N, the accuracy of the rand generator is better than the Sobol sequence. Halton sequence in error and accuracy is the most important generator between above employed generators. Also, the proposed algorithm is efficient when each element of the pencil  $(A, B)$  is ill conditioned (Example 6).

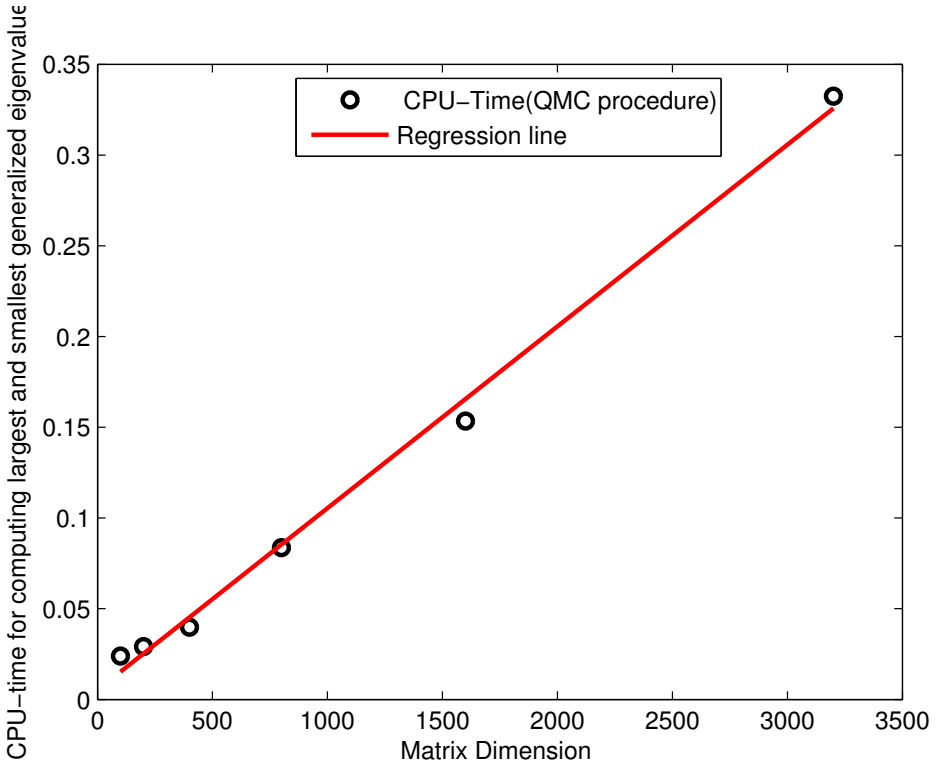


FIGURE 5: CPU time for various matrices



## References

- [1] Alexandrov V. N., Efficient parallel Monte Carlo Methods for Matrix Computation, *Mathematics and computers in Simulation*, Elsevier **47** (1998) 113–122, [doi:10.1016/S0378-4754\(98\)00097-4](https://doi.org/10.1016/S0378-4754(98)00097-4) [E42](#), [E43](#)
- [2] Chi H., Mascagni M. and Warnock T., On the optimal Halton sequence, *Mathematics and Computers in Simulation*, **70** (2005) 9–21, [doi:10.1016/j.matcom.2005.03.004](https://doi.org/10.1016/j.matcom.2005.03.004) [E45](#)
- [3] Dimov I., *Monte Carlo methods for applied scientists*, World Scientific Publishing Co., 2008. [E42](#), [E43](#), [E47](#), [E48](#)
- [4] Dimov I., Alexandrov V. and Karaivanova A., Implementation of Monte Carlo Algorithms for Eigenvalue Problem Using MPI, *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, Springer, 1998. [E42](#), [E43](#), [E46](#)
- [5] Fathi Vajargah B. and Mehrdoust F., New Monte Carlo algorithm for obtaining three dominant eigenvalues, *Int. J. Appl. Math.* **22** (2009), no. 4, 553–559. [E42](#), [E43](#)
- [6] Kressner D., *Numerical Methods for General and Structured Eigenvalue Problems*, Springer–Verlag Berlin Heidelberg, 2005. [E43](#)
- [7] Lemieux C., *Monte Carlo and Quasi Monte Carlo Sampling*, Springer Science, 2009. [E43](#), [E44](#), [E48](#), [E49](#)
- [8] Mascagni M. and A. Karaivanova, A Parallel Quasi Monte Carlo Method for Computing Extremal Eigenvalues, *Monte Carlo and Quasi Monte Carlo Methods*, Springer, **12** (2002) 369–380. [E43](#)
- [9] Sobol I. M., On quasi Monte Carlo integrations, *Mathematics and Computers in Simulation*, **47** (1998) 103–112. [doi:10.1016/S0378-4754\(98\)00096-2](https://doi.org/10.1016/S0378-4754(98)00096-2) [E43](#), [E44](#), [E48](#)

- [10] Sobol I. M., Quasi Monte Carlo methods, *Progress in Nuclear Energy*, **24** (1990) 55–61. [E43](#), [E44](#), [E45](#)

## Author addresses

1. **Behrouz Fathi Vajargah**, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran.  
<mailto:fathi@guilan.ac.ir>
2. **Farshid Mehrdoust**, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran.  
<mailto:fmehrdoust@guilan.ac.ir>