ORIGINAL PAPER

NOVEL APPROACH FOR EIGENVALUE PROBLEMS USING THE MONTE CARLO METHOD

FAUZIA SHAHEEN[1], NAJMUDDIN AHMAD[1]

Manuscript received: 16.05.2023; Accepted paper: 02.11.2023;
Published online: 30.12.2023.

Abstract.*In this paper, we have studied various types of Monte Carlo methods along with the Power method to evaluate the maximum and minimum eigenvalue of a linear system of equations. We have studied how the accuracy of the maximum eigenvalue depends on the parameters,* ℓ *(moves in Markova chain),* $\aleph$ *(no of Markova chain),* $\wp$ *(accelerating parameter), and a parameter* $\boldsymbol{m}$ *(the power applied on the resolving matrix). We have applied these methods to the randomly chosen symmetric matrices. We have also made comparisons for the different matrices of different orders depending on the parameters by using the Monte Carlo methods. We are using Matlab 2020R for the calculation.*

Keywords: *Monte Carlo Method; power method; Markov chain; eigenvalues.*

1. INTRODUCTION

Let A be a real symmetric matrix. Let's consider the eigenvalues problem of matrix A

$$Au = \lambda u. \tag{1}$$

The problem of computing the smallest eigenvalue of A is known to be more challenging numerically than the problem of determining the largest eigenvalue. Nonetheless, estimating the smallest eigenvalue is essential for numerous applications in physics and engineering because it usually defines the most stable state of the system described by the relevant matrix. There are numerous fields where we get problems in terms of probability. To deal with such a problem Monte Carlo method followed by the Markov chain was suggested to solve the problems of various areas. Monte Carlo algorithms correspondingly use the concept of the power method combined by the given matrix, the resolvent matrix and the reverse matrix with Monte Carlo iterations [1-8]. Several authors have worked on Advancement of Monte Carlo methods [9-20].

One of the important test elements where high-efficiency parallel algorithms are needed is the problem of plotting spectral matrices pictures. In stability analysis the spectral portraits are used. The fundamental idea of the Monte Carlo methods is that a random process is constructed in order to address a particular problem. A random process can be developed in such a way as to give an approximate solution to the problem under study or estimate an integral function of the solution by a certain random estimator over the samples of this process. Such a random process cannot be developed in general in a single way. Efficient algorithms for assessing smallest eigenvalue are not easy to find for large sparse matrices

[1] Integral University, Department of Mathematics & Statistics, 226026 Lucknow, India.
E-mail:shaheenfauzia97@gmail.com; najmuddinahmad33@gmail.com.

https://doi.org/10.46939/J.Sci.Arts-23.4-a12

although some authors have worked on advanced parallel algorithm to improve the efficiency of the method [21-34].

Numerical algorithms of Monte Carlo method can be divided into two classes - direct method and iterative method. Here we are considering iterative stationary linear Monte Carlo method for the evaluation of eigenvalues. Monte Carlo method is very sensitive about optimal parameters. Wrong choices of the parameters may lead to a divergence in the method. In this research work we worked on optimal parameters to deal the challenges. Organization of the study is as follows: In the first section some introduction along with previous work has been given. In the next section methodology of the Monte Carlo method has been given. In the third section some numerical examples are given to illustrate how the optimal parameters work for Markov chain. Finally results and discussion along with the conclusion have been presented.

2. MATERIALS AND METHODS

2.1. ITERATIVE MONTE CARLO METHODS

Numerical algorithms of Monte Carlo can be divided into two classes – direct and iterative. Here we are presenting stationary linear algorithms from Monte Carlo for the evaluation of eigenvalues.

2.2. POWER METHOD

Suppose $A \in \mathbb{R}^{n\times n}$ is invertble, $\mathcal{X}A\mathcal{X} = diag(\lambda_1,\dots,\lambda_n)$, $\mathcal{X} = (\mathcal{X}_1,\dots,\mathcal{X}_n)$, and $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$. Given $\mathbb{f} \in \mathbb{R}^n$, the power method produces a vector sequences $\mathbb{f}^{(k)}$ as follows:

$$\begin{aligned} \mathcal{y}^{(k)} &= A\mathbb{f}^{(k-1)} \\ \mathbb{f}^{(k)} &= \mathcal{y}^{(k)}/||\mathcal{y}^{(k)}||_2 \\ \lambda^{(k)} &= [\mathbb{f}^{(k)}]^H A\mathbb{f}^{(k)}, k = 1,2,\dots. \end{aligned} \tag{2}$$

The iterations converge to an Eigenvalue of A, with the greatest magnitude (dominantly the own value), and with convergence rate, except for the special starting points.

$$\left|\lambda_1 - \lambda^{(k)}\right| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right).$$

We shall take the smallest eigenvalue into consideration. The Power method is modified as follows in order to handle this and other cases. Replace matrix A with $\mathcal{P}$ in such away both the matrices A and $\mathcal{P}$ have different eigenvalues of their own but the same eigenvectors. Matrix P can select one of the three options listed below.

a) $\mathcal{P} = A - \wp I$, referred to as the shifted power method for the appropriate shift $\wp$.

b) $\mathcal{P} = A^{-1}$, referred to as the inverse power method, provided A is invertible.

c) $\mathcal{P} = (A - \wp I)^{-1}$, referred to as the inverse shifted power method provided $\mathcal{P}$ is invertible.

2.3. ALMOST OPTIMAL MONTE CARLO METHOD

Consider a matrix $A = \{a_{ij}\}_{i,j=1}^{n}, A \in \mathbb{R}^{n\times n}$, and vectors $b = (b^1,\dots,b^n)^t \in \mathbb{R}^{n\times 1}$ and h $=(h_1,h_2\dots,h_n)^t \in \mathbb{R}^{n\times 1}$. The matrix transformation $Ab \in \mathbb{R}^{n\times 1}$ is called iteration and has a fundamental role in iterative Monte Carlo methods. Consider the following Markov chain

$$\mathcal{C}_0 \to \mathcal{C}_1 \to \dots \to \mathcal{C}_i \tag{3}$$

where $\mathcal{C}_j = 1,2,\dots,n$ for $j = 1,\dots,i$are natural numbers. The probability matrix P defines a Markov chain with states $\{1,\dots,i,n+1\}$, where $n+1$ is an absorbing state and $p_{ij}, i,j = 1,\dots,n+1$ is the one step transition probability from state i to state j. This type of chain is called a Markov chain. It is also termed as a random walk, as it is homogeneous and finite.

Let the trajectory is denoted as, $\delta = (i_0,i_1,\dots,i_k,n+1)$. The initial state of the trajectory is $i_0 < n+1$. It contains the sequence of states such as $(i_1,\dots,i_{\mathcal{C}})$ ends with a final state which is known as an absorbing state such as $i_{\mathcal{C}} = n+1$. Consider a vector $\alpha = (\alpha_1,\dots,\alpha_n)$, where $\alpha_i, i = 1,\dots,n$ is the probability that a trajectory starts in state i.

Mathematically,

$$P(i_0 = i) = \alpha_i, \alpha_i \geq 0, i = 1,\dots,n, \sum_{i=1}^{n} \alpha_i = 1 \tag{4}$$

The probability to follow trajectory δ is $P(\delta) = \alpha_{i_0} p_{i_0 i_1} \dots p_{i_{(\mathcal{C}-1)} i_{\mathcal{C}}} p_{i\mathcal{C}}$. We consider the space of trajectories and define the estimators

$$\theta_i, i = 1,\dots,n \text{ and } \lambda_i, i = 1,\dots,n .$$

For a single trajectory $\delta = (i_0,i_{\mathcal{C}},\dots,i_{\mathcal{C}},n+1)$, the values of these estimators are defined as

$$\theta_i(\delta) = \mathcal{Z}_k(\delta)\frac{\delta_{i_k i}}{p_{ik}}$$

$$\lambda_i(\delta) = \sum_{m=0}^{k} \mathcal{Z}_m(\delta)\delta_{i_m i}, i = 1,\dots,n,$$

where $\mathcal{Z}_m, m = 0,\dots,\mathcal{C}$ random variables are also termed as weight functions whosevalues for a particular trajectory are given as

$$\mathcal{Z}_0(\delta) = c_{i_0}\alpha_{i_0}$$
$$\mathcal{Z}_m(\delta) = \mathcal{Z}_{m-1}1(\delta)\mathcal{Z}_{i_{m-1} i_m}$$
$$= c_{i_0}\alpha_{i_0}\mathcal{Z}_{i_0 i_1}\mathcal{Z}_{i_1 i_2}\dots\mathcal{Z}_{i_{m-1 0} i_m}, m = 1,\dots,\mathcal{C}.$$

The above values are taken with probability $P(\delta)$, where (δ_{ij} is the kronecker symbol, i.e., $\delta_{ij} = 1$ if $i = j$ and 0 otherwise). It can be proved that $\theta_i$and λ_i are unbiased estimators of $x_i, i.e.$

$$\mathscr{E}(\theta_i) = \mathscr{E}(\lambda_i) = x_i, i = 1,\dots,n.$$

The chain (3) is constructed according to the rule as

$$P_r(\mathcal{C}_0 = \alpha) = \frac{|h_\alpha|}{\sum_{\alpha=1}^{n}|h_\alpha|},$$
$$P_r\left(\mathcal{C}_j = \beta \,|\, \mathcal{C}_{j-1} = \alpha\right) = \frac{|a_{\alpha\beta}|}{\sum_{\beta=1}^{n}|a_{\alpha\beta}|}, \alpha = 1,2,\ldots.n. \tag{5}$$

One can use the initial density vector $P_r(\mathcal{C}_0 = \alpha)$ and the transition density matrix $P_r\left(\mathcal{C}_j = \beta \,|\, \mathcal{C}_{j-1} = \alpha\right)$ to apply Almost Monte Carlo method.
Now define the random variables $\mathcal{Z}_j$ using the following recursion formula.

$$\mathcal{Z}_0 = \frac{h_{\mathcal{C}_0}}{p_{\mathcal{C}_0}}, \mathcal{Z}_j = \frac{\mathcal{Z}_{j-1}a_{k_j-1}a_{k_j}}{p_{\mathcal{C}_{j-1}\mathcal{C}_j}}, j = 1,\ldots,\mathrm{i}$$
$$\mathcal{E}\left(\mathcal{Z}_i f_{k_i}\right) = \left(h, A^i f\right), \tag{6}$$

$$\lambda_{max} \approx \frac{\mathcal{E}\left\{\mathcal{Z}_i f_{\mathcal{C}_i}\right\}}{\mathcal{E}\left\{\mathcal{Z}_{i-1} f_{\mathcal{C}_{i-1}}\right\}}, \text{ for sufficiently large i} \tag{7}$$

$$\lambda_{max} \approx \frac{\frac{1}{\aleph}\left[\sum_{j=1}^{\aleph}(\mathcal{Z}_i f_{\mathcal{C}_i})\right]}{\frac{1}{\aleph}\left[\sum_{j=1}^{\aleph}(\mathcal{Z}_{i-1} f_{\mathcal{C}_{i-1}})\right]}. \tag{8}$$

2.4. INVERSE SHIFTED MONTE CARLO METHOD (THE RESOLVENT MONTE CARLO METHOD)

Now consider an algorithm based on Monte Carlo iterations by the resolvent matrix $R_\wp = [I - \wp A]^{-1} \in \mathbb{R}^{n\times n}$ where $\wp$ is an applied appropriate shift. The following presentation holds

$$[I - \wp A]^{-m} = \sum_{i=0}^{\infty} \wp^i B_{i+m-1}^i A^i, |\wp A| < 1.$$
$$([I - \wp A]^{-m} f, h) = \mathcal{E}\left\{\sum_{i=0}^{\infty} \wp^i B_{i+m-1}^i A^i\right\}.$$

According to Monte Carlo algorithm,

$$\lambda_{min} \approx \frac{1}{\wp}\left(1 - \frac{1}{\mu^{(m)}}\right) = \frac{(A[I-\wp A]^{-m} f, h)}{([I-\wp A]^{-m} f, h)},$$
$$= \frac{\mathcal{E}\sum_{i=1}^{\infty} \wp^{i-1} B_{i+m-2}^{i-1} \mathcal{Z}_i f(x_i)}{\mathcal{E}\sum_{i=1}^{\infty} \wp^i B_{i+m-1}^i \mathcal{Z}_i f(x_i)},$$
$$= \frac{\mathcal{E}\sum_{i=0}^{\ell} \wp^i B_{i+m-1}^i \mathcal{Z}_{i+1} f(x_{i+1})}{\mathcal{E}\sum_{i=0}^{\ell} \wp^i B_{i+m-1}^i \mathcal{Z}_i f(x_i)},$$

where $\mathcal{Z}_0 = \frac{h_{k_0}}{p_{k_0}}$, $\mathcal{Z}_i$ are defined by (6) and B_i^j are binomial coefficients. Let us note that if $\wp > 0$ the algorithm evaluates maximum eigenvalue, if $\wp < 0$ the algorithm computes the minimum eigenvalue without inverting the matrix. This parameter could be used to control

the convergency. The following recursion formula is used to calculate the binomil coefficients B^{n}_{m+n}

$$B^{i}_{i+m} = B^{i}_{i+m-1} + B^{i-1}_{i+m-1}.$$

The RMC algorithm's matrices should be carefully chosen in order to be applicable.The value of the parameter m is determined by the absolute error from the Power method applied to the resolvent matrix.The absolute error caused by representing the resolvent matrix as a series specifies the value of the variable ℓ, and the values of m and ℓ are not independent because they establish the binomial coefficients $B^{\ell}_{\ell-1=m}$ which grow exponentially with ℓ.

2.5. THEOREM

Let $\lambda^{'}_{max}$ be the largest eigenvalue of the matrix $A^{'} = \{|a_{ij}|\}^{n}_{i,j=1}$ If parameter $\wp$ is chosen in such a way that $|\lambda^{'}_{max}\wp| < 1$, then

$$([I - \wp A]^{-m} f, h) = \mathcal{E}\left\{\sum_{i=0}^{\infty} \wp^{i} B^{i}_{i+m-1} \mathcal{Z}_i f(x_i)\right\}. \tag{9}$$

Proof: Since the expansion (8) converges in uniform operator topology it converges for any vector f

$$([I - \wp A]^{-m} f, h) = \mathcal{E}\left\{\sum_{i=0}^{\infty} \wp^{i} B^{i}_{i+m-1} (A^{i} f, h)\right\}. \tag{10}$$

For obtaining (10) from (9) one needs to apply (7) and to average every term of the presentation (10). Such averaging will be correct if A, f, h and $\wp$ in (9) are replaced by their absolute values. If it is done the sum (9) will be finite since the condition $|\lambda^{'}_{max}\wp| < 1$ is fulfilled. Thus, for a finite sum (9) there is a finite major ant summed over all terms and the expansion can be average over all terms. The theorem is proved. After some calculations one can obtain

$$\lambda_{min} \approx \frac{1}{\wp}\left(1 - \frac{1}{\mu^{(m)}}\right) = \frac{(A[I-\wp A]^{-m} f, h)}{([I-\wp A]^{-m} f, h)},$$
$$= \frac{\mathcal{E}\{\sum_{i=1}^{\infty} \wp^{i-1} C^{i-1}_{i+m-2} \mathcal{Z}_i f(x_i)\}}{\mathcal{E}\{\sum_{i=1}^{\infty} \wp^{i} C^{i}_{i+m-1} \mathcal{Z}_i f(x_i)\}},$$
$$= \frac{\mathcal{E}\{\sum_{i=0}^{l} \wp^{i} C^{i}_{i+m-1} \mathcal{Z}_{i+1} f(x_{i+1})\}}{\mathcal{E}\{\sum_{i=0}^{l} \wp^{i} C^{i}_{i+m-1} \mathcal{Z}_i f(x_i)\}},$$

where $\mathcal{Z}_0$ and $\mathcal{Z}_i$ are defined by (6). The parameter $\wp < 0$ must be chosen in order to minimize the following expression

$$\mathcal{L}(\wp, A) = \frac{1 + |\wp|\lambda_1}{1 + |\wp|\lambda_2},$$

or if $\lambda_1 = \alpha\lambda_2 (0 < \alpha < 1)$,

$$\mathcal{L}(\wp, A) = 1 - \frac{|\wp|\lambda_2(1-\alpha)}{1 + |\wp|\lambda_2},$$

$$\wp = -\frac{1}{2\|A\|}.$$

However, a slightly different value of $\wp = \left(-\frac{\alpha}{\|A\|}\right)$ may yield better results, when a number of algorithm realisations are considered.

2.6. THE INVERSE MONTE CARLO ITERATIVE ALGORITHM (IMCI):

When A is a non-singular matrix, the IMCI algorithm can be used. The smallest by modulus eigenvalue of A is much smaller than the other eigenvalues, the algorithm becomes more efficient. This algorithm can be implemented in two ways.

We begin by computing the inversion of matrix A. Then, using iterations with the inverse matrix, apply the direct Monte Carlo algorithm. When we apply Resolvent Monte Carlo algorithm with $\wp = -1$, i.e.

$$\lambda_{min} \approx \frac{\mathcal{E}\left\{\sum_{i=0}^{\ell} \wp^i B_{i+m-1}^i \mathcal{Z}_{i+1} f(x_{i+1})\right\}}{\mathcal{E}\left\{\sum_{i=0}^{\ell} \wp^i B_{i+m-1}^i \mathcal{Z}_i f(x_i)\right\}}.$$

3. NUMERICAL EXAMPLES

3.1. EXAMPLE 1

Let us consider A x = b to be a linear system of equations, where $A \in \mathbb{R}^{4\times4}$ is a coefficient matrix, $b \in \mathbb{R}^4$ is a column vector and h is a also a column vector, choosen randomly from the any four dimensional real space. The exact maximum eigenvalue of the matrix A is 1.341869385397356.

$$A = \begin{pmatrix} 0.51 & 0.12 & 0.12 & 0.12 \\ 0.12 & 0.51 & 0.31 & 0.31 \\ 0.12 & 0.31 & 0.51 & 0.61 \\ 0.12 & 0.31 & 0.61 & 0.51 \end{pmatrix}, b = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \end{pmatrix}, h = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.1 \\ 0.2 \end{pmatrix}$$

Table 1. Calculation of Eigen values for different values of m & ℓ by using RMC method when $\aleph$=6, ℓ =9.

No.	$\wp$	m	Max Eigen value	Absolute error
1.	0.365	3	1.613798889864895	0.272444913967647
2.	0.438	3	1.255612556953081	0.085741418944167
3.	0.510	3	1.258495739613585	0.082858236283663
4.	0.583	3	1.290223849765392	0.051130126131856
5.	0.656	3	1.343632363054793	0.002278387157545

Table 2. Calculation of Eigen values for different values of ℵ when m &ℓ are fixed by using RMC method. When ℓ = 9, m = 3, $\wp$ = 0.656.

No.	ℵ	Max Eigen value	Absolute error
1.	2	1.372919664841931	0.031565688944683
2.	3	1.393132340671615	0.051778364774367
3.	4	1.363465382862886	0.022111406965638
4.	5	1.343632363054793	0.002278387157545
5.	6	1.345167683205606	0.003813707308358

3.2. EXAMPLE 2

Let us consider A x = b be a linear system of equations, where $A \in \mathbb{R}^{n\times n}$ is a coefficient matrix, b $\in \mathbb{R}^5$ is a column vector and h is a also a column vector, choosen randomly from the any five dimensional real space The exact maximum eigenvalue of the matrix A is 5.800000000000002.

$$A = \begin{pmatrix} 5 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 5 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 5 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 5 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 5 \end{pmatrix}, b = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \end{pmatrix}, h = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.1 \\ 0.2 \\ 0.1 \end{pmatrix}$$

Table 3. Calculation of Eigen values for different values of m &ℓ by using RMC method, when ℵ=5, ℓ =16.

No.	$\wp$	m	Max Eigen value	Absolute error
1.	0.086	3	6.375886477253936	0.575886477253935
2.	0.103	3	6.144106292513895	0.344106292513893
3.	0.120	3	6.084206321399225	0.284206321399224
4.	0.137	3	6.168443761220227	0.368443761220226
5.	0.155	3	6.421559800643887	0.621559800643885

Table 4. Calculation of Eigen values for different values of ℵ when m &ℓ are fixed by using RMC method, ℓ =16, m=3, $\wp$ =0.120.

No.	ℵ	Max Eigen value	Absolute error
1.	5	6.084206321399225	0.284206321399224
2.	10	6.020614482144025	0.220614482144024
3.	15	5.992782487839938	0.192782487839937
4.	20	5.988353488256137	0.188353488256135
5.	25	5.982723388980260	0.182723388980259

3.3. EXAMPLE 3

Let us consider A x = b be a linear system of equations, where $A \in \mathbb{R}^{6\times6}$ is a coefficient matrix, $b \in \mathbb{R}^{6}$ is a column vector and h is a also a column vector, chosen randomly from the any six dimensional real space. The exact maximum eigenvalue of the matrix A is 20.999999999999993.

$$A = \begin{pmatrix} 11 & 2 & 2 & 2 & 2 & 2 \\ 2 & 11 & 2 & 2 & 2 & 2 \\ 2 & 2 & 11 & 2 & 2 & 2 \\ 2 & 2 & 2 & 11 & 2 & 2 \\ 2 & 2 & 2 & 2 & 11 & 2 \\ 2 & 2 & 2 & 2 & 2 & 11 \end{pmatrix}, b = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \\ 0.6 \end{pmatrix}, h = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.1 \\ 0.2 \\ 0.1 \\ 0.2 \end{pmatrix}$$

Table 5. Calculation of Eigen values for different values of m &ℓ by using RMC method, when א=5, ℓ =16.

No.	$\wp$	m	Max Eigen value	Absolute error
1.	0.023	3	22.764707533517416	1.764707533517424
2.	0.028	3	21.794505896943459	0.794505896943466
3.	0.033	3	21.613634579141195	0.613634579141202
4.	0.038	3	22.167539215877234	1.167539215877241
5.	0.042	3	23.121942269419300	2.121942269419307

Table 6. Calculation of Eigen values for different values of א when m &ℓ are fixed by using RMC method, when ℓ = 16, m= 3, $\wp$=0.033.

No.	א	Max Eigen value	Absolute error
1.	5	21.626732990665886	0.626732990665893
2.	10	21.613634579141195	0.613634579141202
3.	15	21.575979991094780	0.575979991094787
4.	20	21.418615804831859	0.418615804831866
5.	25	21.252610488358055	0.252610488358062

3.4. EXAMPLE 4

Let us consider another linear system of equations A x = b, where $A \in \mathbb{R}^{7\times7}$ is a coefficient matrix, $b \in \mathbb{R}^{7}$ is a column vector and h is a also a column vector, chosen randomly from the any seven dimensional real space. The maximum eigen value of the matrix A is 1.40000000000.

$$A = \begin{pmatrix} 0.8 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.8 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.8 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.8 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.8 \end{pmatrix}, b = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \\ 0.6 \\ 0.7 \end{pmatrix}, h = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.1 \\ 0.2 \\ 0.1 \\ 0.2 \\ 0.1 \end{pmatrix}$$

Table 7. Calculation of Eigen values for different values of m &ℓ by using RMC method, when א=25, ℓ =7.

No.	$\wp$	m	Max Eigen value	Absolute error
1.	0.357	3	1.635180872535320	0.235180872535320
2.	0.428	3	1.612654955585784	0.212654955585784
3.	0.500	3	1.638723010749928	0.238723010749928
4.	0.521	3	1.653269274102666	0.253269274102666
5.	0.571	3	1.695505263739988	0.295505263739988

Table 8. Calculation of Eigen values for different values of א when m &ℓ are fixed by using RMC ℓ =7, when m=3, $\wp$=0.428 method.

No.	א	Max Eigen value	Absolute error
1.	5	1.645738997710506	0.245738997710506
2.	10	1.615343848580442	0.215343848580442
3.	15	1.633170305676856	0.233170305676856
4.	20	1.647414583060610	0.247414583060610
5.	25	1.635180872535320	0.235180872535320

3.5. EXAMPLE 5

Let us consider another linear system of equations A x = b, where $A \in \mathbb{R}^{8\times 8}$ is a coefficient matrix, $b \in \mathbb{R}^{8}$ is a column vector and h is a also a column vector, chosen randomly from the any eight dimensional real space. The maximum eigen value of matrix A is 8.350000000000001.

$$A = \begin{pmatrix} 8 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 8 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 8 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 8 & 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 & 8 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 8 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 8 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 8 \end{pmatrix}, b = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \\ 0.6 \\ 0.7 \\ 0.8 \end{pmatrix} \quad h = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.1 \\ 0.2 \\ 0.1 \\ 0.2 \\ 0.1 \\ 0.2 \end{pmatrix}$$

Table 9. Calculation of Eigen values for different values of m &ℓ by using RMC method when א=6, ℓ =16.

No.	$\wp$	m	Max Eigen value	Absolute error
1.	0.059	3	8.723307565754229	0.373307565754228
2.	0.071	3	8.369707564900617	0.019707564900616
3.	0.083	3	8.399291153619075	0.049291153619073
4.	0.095	3	8.713310371717281	0.363310371717279
5.	0.107	3	9.254281417603707	0.904281417603706

Table 10. Calculation of Eigen values for different values of א when m &ℓ are fixed by using RMC method when ℓ =16, m=3, ℘=0.071.

No.	א	Max Eigen value	Absolute error
1.	2	8.388775113953232	0.038775113953230
2.	3	8.332847707251782	0.017152292748220
3.	4	8.333394730772749	0.016605269227252
4.	5	8.353886995684995	0.003886995684994
5.	6	8.369707564900617	0.019707564900616

4. DISCUSSION

Here we have some symmetric matrices of a different order as given in the examples. We will first apply the Almost Optimal Monte Carlo method to find the maximum eigenvalue of the coefficient matrix A as well as the Absolute error from the exact eigenvalue. For this, we first obtain a probability distribution matrix P as given in (5). Then we make a random absorbing Markov chain (δ_i) of length $i_k$on the space set S = (1, 2, 3,...) according to the order of given matrices with 1 as the primary or initial stage according to equation (3). Then we find weights $\mathcal{Z}_i$(i=0, 1, 2.....) for every Markov chain as (6). An expectation value according to equation (7) will give a maximum eigenvalue for matrix A for a sufficiently large value of i. For the resolvent Monte Carlo method we choose a suitable positive value of parameters ℘ and m for the maximum eigenvalues.

5. CONCLUSION

This study presents different types of Monte Carlo methods to deal with probabilistic situations. Tables of comparison for the different order of matrices have been presented for maximum eigenvalues (depending upon the moves, length, and accelerating parameters of the Markov chain). We have concluded that by increasing the no of moves (in Markov chain) and no of trajectories we get results with more accuracy. We have found that a small change in parameter ℘ produces a change in the maximum eigenvalue. So finally we must be very careful about the choice of the accelerating parameter ℘ because the accuracy of the maximum eigenvalues strongly depends on it.

REFERENCES

[1] Sobol, I.M., *Monte Carlo numerical methods*, Nauka, Moscow, 1973.
[2] Groh, D.J., Marshall, R.A., Kunz, A.B., Givens, C.R., *Journal Scientific Computing*, **6**, 251, 1991.
[3] Dimov, I., Karaivanova, A., *Recent Advances Numerical Methods Applications II*, 239, 1999.
[4] Dimov, I.T., *Monte Carlo methods for applied scientists*. World Scientific, Singapore, 2008.

[5] Kalos, M.H., Whitlock, P.A., *Monte Carlo Methods*. John Wiley Sons, Hoboken, New Jersey, United States, 2009.
[6] Mordechai, S., *Applications of Monte Carlo method in science and engineering*. InTech, 2011.
[7] Kroese, D.P., Rubinstein, R.Y., *Wiley Interdisciplinary Reviews: Computational Statistics*, **4**(1), 48, 2012.
[8] Dunn, W.L., Shultis, J.K., *Exploring Monte Carlo Methods*. Elsevier,Amsterdam, Netherlands, 2022.
[9] Danilov, D.L., Ermakov, S.M.,Halton, J.H., *Journal of Statistical Planning and Inference*, **85**(1–2), 5, 2000.
[10] Tan, C.J.K., *Journal Supercomputing*, **22**, 113, 2002.
[11] Barone, P., Sebastiani, G., Stander, J., *Statistics Computing*,**12**(1), 17, 2002.
[12] Drineas, P., Kannan, R., Mahoney, M.W., *SIAM Journal Computing*, **36**(1), 158, 2006.
[13] Halton , J.H., *Sequential Monte Carlo techniques for solving non-linear systems*, 2006.
[14] Dimov, I.T., Philippe, B., Karaivanova, A., Weihrauch, C., *Applied Mathematical Modelling*, **32**(8), 1511, 2008.
[15] Srinivasan, A., *Mathematics Computers Simulation*, **80**(6), 1133, 2010.
[16] Karaivanova, A., *Serdica Journal Computing*, **4**(1), 57, 2010.
[17] Wu, S.J., Chu, M.T., *Linear Algebra and its Applications*, **487**, 184, 2015.
[18] Figueiredo, L.P., Grana, D., Roisenberg, M., Rodrigues, B.B., *Geophysics*, **84**(3),463, 2019.
[19] Mikhailov, G.A., *A new algorithm of the Monte Carlo method for estimation of the maximal eigenvalue of an integral operator*, Doklady Akademii Nauk, Russian Academy of Sciences, **191**(5), 993, 1970.
[20] Mikhailov, G.A., *Minimax algorithms of weight Monte Carlo methods*, 1989.
[21] Dimov, I.T., Karaivanova, A.N., Yordanova, P.I., *Monte Carlo Algorithms for calculating eigenvalues, in Monte Carlo and Quasi-Monte Carlo Methods*, Springer, 205–220, 1998.
[22] Dimov, I., Karaivanova, A., *Monte Carlo Methods and Applications*, **4**(1), 33, 1998.
[23] Dimov, I., Alexandrov,V., Karaivanova, A., Implementation of Monte Carlo algorithms for eigenvalue problem using MPI. In: Alexandrov, V., Dongarra, J. (Eds.), *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, Lecture Notes in Computer Science, Vol. 1497, pp. 346–353, Springer, Berlin, Heidelberg, 1998. https://doi.org/10.1007/BFb0056594.
[24] Mascagni, M., Karaivanova, A., *A parallel Quasi-Monte Carlo method for solving systems of linear equations*, Computational Science, Springer Berlin Heidelberg Berlin, Heidelberg, pp. 598–608, 2002.
[25] Fathi, B., Liu, B., Alexandrov, V., *Computational Science*, **2**, 609, 2002.
[26] Alexandov, V., Atanassov, E.I., Dimov, I., *Monte Carlo Methods Applications*, **10**(3–4), 213, 2004.
[27] Alexandrov, V.N., Atanassov E.I., Dimov, I., Branford, S., Thandavan, A., Weihrauch, C., *Computational Science* , **5**, 752, 2005.
[28] Shonkwiler, R., Lefton, W.L., *An introduction to parallel and vector scientific computation*, Vol. 41, Cambridge University Press, 2006.

[29] Wang, T.L., *Linear Algebra and Its Applications*, **322**(1–3), 1, 2001.
[30] Grigoriu, M., *Monte Carlo Methods Applications*, **20**(2), 121, 2014.
[31] Ahmad, N., Shaheen, F., *Communications in Mathematics and Applications*, **12**(4), 853, 2021.
[32] Ahmad, N., Shaheen, F., *Journal of Science and Arts*, **21**(2), 365, 2021.
[33] Ahmad, N., Singh, V.P., *Journal of Science and Arts*, **16**(4), 329, 2016.
[34] Ahmad, N., Singh, V., *British Journal of Mathematics & Computer Science*, **19**(2), 1, 2016.