A POWER METHOD WITH MONTE CARLO ITERATIONS

IVAN DIMOV AND ANETA KARAIVANOVA
CLPP – Bulgarian Academy of Sciences
Acad. G.Bonchev Str., bl. 25A, 1113 Sofia, Bulgaria
E-mail: dimov@amigo.acad.bg, anet@amigo.acad.bg,
Web site: http://wwwvsapo.acad.bg

We present Monte Carlo versions of the Direct Power Method, the Shifted Power Method, the Inverse Power Method and the Inverse Shifted Power Method for calculating eigenvalues of matrices. An analysis of the accuracy and the convergence of these versions of the Power Method is done. The advantages and disadvantages of the corresponding Monte Carlo algorithms are discussed.

1 Background

1.1 The Power Method

Suppose $A \in \mathbb{C}^{n\times n}$ is diagonalizable, $X^{-1}AX = diag(\lambda_1, \ldots, \lambda_n)$, $X = (x_1, \ldots, x_n)$, and $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$. Given $f^{(0)} \in \mathbb{C}^n$, the *power method* [3] produces a sequence of vectors $f^{(k)}$ as follows:

$$z^{(k)} = Af^{(k-1)},$$

$$f^{(k)} = z^{(k)}/||z^{(k)}||_2,$$

$$\lambda^{(k)} = [f^{(k)}]^H Af^{(k)}, \;\; k = 1, 2, \ldots .$$

Except for special starting points, the iterations converge to an eigenvector corresponding to the eigenvalue of A with largest magnitude (*dominant eigenvalue*).

Convergence. If $f^{(0)} = a_1x_1 + a_2x_2 + \ldots + a_nx_n$, $a_1 \neq 0$, then

$$A^k f^{(0)} = a_1\lambda_1^k \left(x_1 + \sum_{j=2}^{n} \frac{a_j}{a_1} \left(\frac{\lambda_j}{\lambda_1}\right)^k x_j \right),$$

so that

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

The power method approximates the eigenvalue(s) of largest magnitude. Consider the case we want to compute the smallest eigenvalue. To handle this case and others, the power method is altered in the following way: The iteration matrix A is replaced by B, where A and B have the same eigenvectors, but different eigenvalues. Letting σ denote a scalar, there are three common choices for B: $B = A - \sigma I$ which is called the *shifted power method*, $B = A^{-1}$ which is called the *inverse power method*, and $B = (A - \sigma I)^{-1}$ which is called the *inverse shifted power method*.

Table 1. Relationship between eigenvalues of A and B

B	Eigenvalue of B	Eigenvalue of A
A^{-1}	$1/\lambda_A$	$1/\lambda_B$
$A - \sigma I$	$\lambda_A - \sigma$	$\lambda_B + \sigma$
$(A - \sigma I)^{-1}$	$\frac{1}{\lambda_A - \sigma}$	$\sigma + \frac{1}{\lambda_B}$

On the surface, the inverse shifted power method is an unstable algorithm. To compute the eigenpair (λ, x) of A, we select a shift σ near λ. Although the convergence of the inverse shifted power method accelerates as σ approaches λ, the numerical errors also increase as σ approaches λ. The norm of $(A - \lambda I)^{-1}$ is at least $1/|\lambda - \sigma|$. The condition number for $A - \sigma I$ is at least $\| A - \sigma I \| / |\lambda - \sigma|$. Since the conditioning of the shifted matrix worses as σ approaches λ, we expect that the new f computed by the inverse shifted power method is relatively inaccurate when σ is close to λ. Fortunately, the error in the new f points predominantly in the direction of the desired eigenvector x. In other words,

$$\text{computed } f^{new} \approx \text{correct } f^{new} + sx,$$

where s is some scalar. In theory, the f's approach x as the iteration progress. Computationaly, the error in the correct f^{new} is nearly a multiple of x and the ill conditioning associated with the inverse shifted power method does not impede the convergence to x.

Computational Complexity: Having k iterations, the number of arithmetic operations in Power method is $O(4kn^2 + 3kn)$, so the Power Method is not suitable for large sparse matrices. In order to reduce the computational complexity we propose a Power Method with Monte Carlo iterations.

1.2 Monte Carlo Method

Monte Carlo methods give statistical estimates for the functional of the solution by performing random sampling of a certain random variable whose mathematical expectation is the desired functional.

Let J be any functional that we estimate by Monte Carlo method; θ_N be the estimator, where N is the number of trials. The probable error for the usual Monte Carlo method (which does not use any additional a priori information about the regularity of the solution) [6] is defined as parameter r_N for which:

$$Pr\{|J - \theta_N| \geq r_N\} = 1/2 = Pr\{|J - \theta_N| \leq r_N\}. \tag{1}$$

If the standard deviation is bounded, i.e. $D(\theta_N) < \infty$, the normal convergence in the central limit theorem holds.

$$Pr\{|J - \theta_N| \leq x\sigma(\theta_N)N^{-1/2}\} \approx \Phi(x).$$

Obviously, from Eq.(1) and $\Phi(0.6745) \approx \frac{1}{2}$ we have

$$r_N \approx 0.6745 D(\theta_N) N^{-1/2}. \tag{2}$$

In [7], [6], [4], [5] one can find Monte Carlo methods for evaluation the dominant (maximal by modulus) eigenvalue of an integral operator. In this paper we continue our works [1],[2], (where we consider real symmetric matrices) on Monte Carlo methods for calculation eigenvalues of complex matrices.

2 Monte Carlo iterations

Consider a matrix $A = \{a_{ij}\}_{i,j=1}^n$, $A \in I\!R^{n\times n}$, and vectors $f = (f_1, \ldots, f_n)^t \in I\!R^{n\times 1}$ and $h = (h_1, \ldots, h_n)^t \in I\!R^{n\times 1}$. The algebraic transformation $Af \in I\!R^{n\times 1}$ is called *iteration* and plays a fundamental role in iterative Monte Carlo methods.

Consider the following Markov chain:

$$k_0 \to k_1 \to \ldots \to k_i, \tag{3}$$

where $k_j = 1, 2, \ldots, n$ for $j = 1, \ldots, i$ are natural numbers. The rules for constructing the chain such as Eq.(3) are:

$$Pr(k_0 = \alpha) = \frac{|h_\alpha|}{\sum_{\alpha=1}^{n} |h_\alpha|},$$

$$Pr(k_j = \beta | k_{j-1} = \alpha) = \frac{a_{\alpha\beta}}{\sum_{\beta=1}^{n} |a_{\alpha\beta}|}, \; \alpha = 1, \ldots, n.$$

Such a choice of the initial density vector and the transition density matrix leads to *almost optimal* Monte Carlo algorithms for matrix computations.

Now define the random variables W_j using the following recursion formula:

$$W_0 = \frac{h_{k_0}}{p_{k_0}}, \;\; W_j = W_{j-1} \frac{a_{k_{j-1}k_j}}{p_{k_{j-1}k_j}}, \;\; j = 1, \ldots, i. \tag{4}$$

It is easy to show (see, for example [6], [7]) that

$$(h, A^i f) = E\{W_i f_{k_i}\}, \;\; i = 1, 2, \ldots. \tag{5}$$

and (see, [4], [5])

$$([I - qA]^{-m} f, h) = E\left\{ \sum_{i=0}^{\infty} q^i C_{m+i-1}^i (A^i f, h) \right\}.$$

The following Monte Carlo versions of the method under consideration are studied:

2.1 The Power Method with Monte Carlo Iterations

$$\lambda_{max} \approx \frac{E\{W_i f_{k_i}\}}{E\{W_{i-1} f_{k_{i-1}}\}}.$$

2.2 The Inverse Power Method with Monte Carlo Iterations

$$\lambda_{min} \approx \frac{E \sum_{i=0}^{l} C_{i+m-1}^i W_{i+1} f(x_{i+1})}{E \sum_{n=0}^{l} C_{i+m-1}^i W_i f(x_i)}.$$

2.3 The Inverse Shifted Power Method with Monte Carlo Iterations (Resolvent MC Method)

$$\lambda_{min} \approx \frac{1}{q}\left(1 - \frac{1}{\mu^{(m)}}\right) = \frac{(A[I-qA]^{-m}f,h)}{([I-qA]^{-m}f,h)} =$$

$$\frac{E\sum_{i=1}^{\infty} q^{i-1}C_{i+m-2}^{i-1}W_i f(x_i)}{E\sum_{i=0}^{\infty} q^i C_{i+m-1}^{i} W_i f(x_i)} = \frac{E\sum_{i=0}^{l} q^i C_{i+m-1}^{i} W_{i+1} f(x_{i+1})}{E\sum_{n=0}^{l} q^i C_{i+m-1}^{i} W_i f(x_i)}, \quad (6)$$

where $W_0 = \frac{h_{k_0}}{p_{k_0}}$ and W_i are defined by Eq.(4). The coefficients C_{n+m}^{n} are calculated using the presentation

$$C_{i+m}^{i} = C_{i+m-1}^{i} + C_{i+m-1}^{i-1}.$$

2.4 Balancing of errors

There are two kind of errors in the Power method with Monte Carlo iterations:

- systematic error (from the Power method):

  $$O\left(\left|\frac{\mu_2}{\mu_1}\right|^k\right),$$

  where $\mu_i = \lambda_i$ if $B = A$, $\mu_i = \frac{1}{\lambda_i}$ if $B = A^{-1}$, $\mu_i = \lambda_i - \sigma$ if $B = A - \sigma I$, $\mu_i = \frac{1}{\lambda_i - \sigma}$ if $B = (A-\sigma I)^{-1}$ and λ_i and μ_i are the eigenvalues of A and B correspondingly;

- stochastic error (because we calculate mathematical expectations approximately): $O(D(\theta_N)N^{-1/2})$.

The problem of balancing of errors is very important. It is clear that for obtaining good results the stochastic error must be approximately equal to the systematic one. (It is not necessary to use "a large" number of realizations N in order to have "a small" stochastic error if the systematic error is "large".)

Remark. Monte Carlo approach for calculating the dominant eigenvalue is based on computing the scalar products $(h, A^k h), k = 1, 2, \ldots$. Let $\| h \| = 1$. Having

$$(h, A^m h) = E(W_m h_{k_m}),$$

we consider the estimation

$$(h, A^m h) \approx \frac{1}{N} \sum_{i=1}^{N} (W_m h_{k_m})_i$$

with a statistical error $c\sqrt{D(W_m h_{k_m})}/\sqrt{N}$. In this case

$$|W_m h_{k_m}| = \frac{h_{k_0}}{p_{k_0}} . \frac{a_{k_0 k_1} \dots a_{k_{m-1} k_m}}{p_{k_0 k_1} \dots p_{k_{m-1} k_m}}$$

$$= \sum_{i=1}^{n} |h_i| . \sum_{i=1}^{n} |a_{k_0 i}| . \sum_{i=1}^{n} |a_{k_1 i}| \dots \sum_{i=1}^{n} |a_{k_{m-1} i}|$$

$$\leq \sum_{i=1}^{n} |h_i| . \left(max_k \left\{ \sum_{i=1}^{n} |a_{ki}| \right\} \right)^m .$$

Let us estimate the variance of the random variable $W_m h_{k_m}$:

$$D(W_m h_{k_m}) = E(W_m h_{k_m})^2 - (E(W_m h_{k_m}))^2$$

$$\leq E(W_m h_{k_m})^2 \leq \left(\sum_{i=1}^{n} |h_i| \right)^2 . \left(max_k \sum_{i=1}^{n} |a_{ki}| \right)^{2m} .$$

One can see, that

$$minD(W_m h_{k_m}) = 0,$$

if

$$\sum_{i=1}^{n} |a_{ki}| = c, \quad \forall k = 1, \dots, n, \tag{7}$$

where c is a constant.

The last remark shows that to minimize the stochastic error it is important to deal with *well-balanced matrices*, that is matrices with not too large variation of $\sum_{i=1}^{n} |a_{ki}|$. It is easy to see, that the smallest stochastic error can be obtained for the so called *stochastic* matrices for which the condition given by Eq.(7) is fulfilled.

2.5 Computational Complexity

The mathematical expectation of the total number of operations for the Resolvent Monte Carlo Method (RMC) [1], [2], (the most time consuming method) is:

$$ET_1(RMC) \approx 2\left[(k+\gamma_A)l_A + \frac{1}{2}dl_L\right] lN + 2n(1+d), \tag{8}$$

where l is the numbers of moves in every Markov chain, N is the number of Markov chains, d is the mean value of the number of non-zero elements per row, γ_A is the number of arithmetic operations for calculation the random variable (in our code-realization of the algorithm $\gamma_A = 6$), l_A and l_L are arithmetic and logical suboperations in one move of the Markov chain, k is the number of arithmetic operations for generating the random number (k is equal to 2 or 3).

It is worth noting that the main term of Eq.(8) does not depend on the size n of the matrix. This result means that the time required for calculating the eigenvalue by RMC practically does not depend n. The parameters l and N depend on the spectrum of the matrix, but not depend on the size n. The above mentioned result was confirmed for a wide range of matrices during the realized numerical tests.

2.6 Computation of a Complex Conjugate Pair of Eigenvalues

A procedure similar to that employed for a single real eigenvalue can be used to compute a complex conjugate pair of eigenvalues $\lambda_1 = a+bi$ and $\lambda_2 = a-bi$. It can be shown that λ_1 and λ_2 are the roots of the quadratic equation

$$\lambda^2 + c_1\lambda + c_2 = 0,$$

where c_1 and c_2 have the property that for any three consecutive iterations x_k, x_{k+1} and x_{k+2} in the power method, we have

$$x_{k+2} + c_1 x_{k+1} + c_2 x_k \approx 0.$$

To estimate c_1 and c_2, the least squares solution d_1 and d_2 to the overdetermined system

$$d_1 x_{k+1} + d_2 x_k = -x_{k+2}, \tag{9}$$

or

$$\begin{bmatrix} x_{k+1}.x_{k+1} & x_{k+1}.x_k \\ x_{k+1}.x_k & x_k.x_k \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = - \begin{bmatrix} x_{k+2}.x_{k+1} \\ x_{k+2}.x_k \end{bmatrix}$$

can be used. Solving this equation gives us the coefficients of a quadratic and the zeros of this quadratic approximate the two dominant eigenvalues. When these dominant eigenvalues are a complex conjugate pair $\lambda_1 = a + bi$ and $\lambda_2 = a - bi$, it can be shown that the corresponding eigenvectors are approximately equal to

$$bx^{old} \pm i(ax^{old} - x^{new})$$

provided the power iterations are real.

The Monte Carlo Algorithm

Let $h \in I\!R^{1\times n}$ be n-dimensional vector. One can prove that the system (9) is equivalent to the following one

$$d_1(h, Ax_k) + d_2(h, x_k) = -(h, A^2x_k).$$

Suppose, we have the values of four MC iterations:

$$(h, x_{k-3});\ (h, Ax_{k-3}) = (h, x_{k-2});$$

$$(h, A^2x_{k-3}) = (h, x_{k-1});\ (h, A^3x_{k-3}) = (h, x_k).$$

Now, one can prove that the dominant complex conjugate eigenvalues λ_1 and λ_2 of the matrix A are the solution of the following quadratic equation:

$$\lambda^2 + d_1\lambda + d_2 = 0,$$

where

$$d_1 = \frac{(h, x_{k-1})(h, x_{k-2}) - (h, x_k)(h, x_{k-3})}{(h, x_{k-1})(h, x_{k-3}) - (h, x_{k-2})^2} \tag{10}$$

and

$$d_2 = \frac{(h, x_k)(h, x_{k-2}) - (h, x_{k-1})^2}{(h, x_{k-1})(h, x_{k-3}) - (h, x_{k-2})^2}. \tag{11}$$

3 Conclusion

Monte Carlo versions of the the Power method are presented. We propose to use them when one has to calculate the dominant eigenvalue of large sparse matrices because the computational time is almost independent of the dimension of the matrix.

A special case when the dominant eigenvalues are complex conjugated is also considered. A Monte Carlo version in this case is also proposed.

References

1. I. Dimov, A. Karaivanova and P. Yordanova in: *Proceedings of MC & QMC 96*, **Springer Lecture Notes in Statistics, v. 127** (1998) (H. Niederreiter, P. Hellekalek, G. Larcher and P. Zinterhof, Eds), pp.205-220.
2. I. Dimov, A. Karaivanova, Journal of Monte Carlo Method and Applications **4, No. 1**, 33 (1998).
3. G. H. Golub, Ch. F. Van Loon, *Matrix Computations*, **The Johns Hopkins Univ. Press**, Baltimore and London, 1996.
4. G.A. Mikhailov, Docl. Acad. Nauk SSSR **191, No 5**, 993 (1970).
5. G.A. Mikhailov, *Optimization of the "weight" Monte Carlo methods*, **Nauka**, Moscow, 1987.
6. I.M. Sobol, *Monte Carlo numerical methods*, **Nauka**, Moscow, 1973.
7. V.S.Vladimirov, Teoriya Veroyatnostej i Yeye Primenenie **1**, 113 (1956).