Matrix Computations Using Quasirandom Sequences

Michael Mascagni[1] and Aneta Karaivanova[1,2]

[1] Department of Computer Science, Florida State University
203 Love Building, Tallahassee, FL 32306-4530, USA
mascagni@cs.fsu.edu
http://www.cs.fsu.edu/~mascagni

[2] Central Laboratory for Parallel Processing, Bulgarian Academy of Sciences
Acad. G. Bonchev St.,bl. 25 A, 1113, Sofia, Bulgaria
aneta@scri.fsu.edu

Abstract. The convergence of Monte Carlo method for numerical integration can often be improved by replacing pseudorandom numbers (PRNs) with more uniformly distributed numbers known as quasirandom numbers(QRNs). Standard Monte Carlo methods use pseudorandom sequences and provide a convergence rate of $O(N^{-1/2})$ using N samples. Quasi-Monte Carlo methods use quasirandom sequences with the resulting convergence rate for numerical integration as good as $O((logN)^k)N^{-1})$.

In this paper we study the possibility of using QRNs for computing matrix-vector products, solving systems of linear algebraic equations and calculating the extreme eigenvalues of matrices. Several algorithms using the same Markov chains with different random variables are described. We have shown, theoretically and through numerical tests, that the use of quasirandom sequences improves both the magnitude of the error and the convergence rate of the corresponding Monte Carlo methods. Numerical tests are performed on sparse matrices using PRNs and Soboĺ, Halton, and Faure QRNs.

1 Introduction

Monte **C**arlo **m**ethods (MCMs) are based on the simulation of stochastic processes whose expected values are equal to computationally interesting quantities. Despite the universality of MCMs, a serious drawback is their slow convergence, which is based on the $O(N^{-1/2})$ behavior of the size of statistical sampling errors. This represents a great opportunity for researchers in computational science. Even modest improvements in the MCM can have substantial impact on the efficiency and range of applicability for MCM. Much of the effort in the development of Monte Carlo methods has been in construction of variance reduction methods which speed up the computation by reducing the constant in front of the $O(N^{-1/2})$. An alternative approach to acceleration is to change the choice of sequence and hence improve the behavior with N. **Q**uasi-**M**onte **C**arlo

L. Vulkov, J. Waśniewski, and P. Yalamov (Eds.): NAA 2000, LNCS 1988, pp. 552–559, 2001.

methods (QMCMs) use quasirandom (also known as low-discrepancy) sequences instead of pseudorandom sequences.

QRNs are constructed to minimize a measure of their deviation from uniformity called discrepancy. There are many different discrepancies, but let us consider the most common, the star discrepancy. Let us define the star discrepancy of a one-dimensional point set, $\{x_n\}_{n=1}^{N}$, by

$$D^\star = D^\star(x_1, \dots, x_N) = \sup_{0 \le u \le 1} |\frac{1}{N} \sum_{n=1}^{N} \chi_{[0,u)}(x_n) - u| \qquad (1)$$

where $\chi_{[0,u)}$ is the characteristic function of the half open interval $[0, u)$. The mathematical motivation for quasirandom numbers can be found in the classic Monte Carlo application of numerical integration. We detail this for the trivial example of one-dimensional integration for illustrative simplicity.

Theorem (Koksma-Hlawka, [6]): if $f(x)$ has bounded variation, $V(f)$, on $[0, 1)$, and $x_1, \dots, x_N \in [0, 1]$ have star discrepancy $D^\star$, then:

$$|\frac{1}{N} \sum_{n=1} N f(x_n) - \int_0^1 f(x)\, dx| \le V(f) D^\star, \qquad (2)$$

The star discrepancy of a point set of N truly random numbers in one dimension is $O(N^{-1/2}(\log\log N)^{1/2})$, while the discrepancy of N quasirandom numbers can be as low as N^{-1}. [1] In $s > 3$ dimensions it is rigorously known that the discrepancy of a point set with N elements can be no smaller than a constant depending only on s times $N^{-1}(\log N)^{(s-1)/2}$. This remarkable result of Roth, [10], has motivated mathematicians to seek point sets and sequences with discrepancies as close to this lower bound as possible. Since Roth's remarkable results, there have been many constructions of low discrepancy point sets that have achieved star discrepancies as small as $O(N^{-1}(\log N)^{s-1})$. Most notably there are the constructions of Hammersley, Halton, [5], Soboĺ, [11], Faure, [3], and Niederreiter, [9].

While QRNs do improve the convergence of applications like numerical integration, it is by no means trivial to enhance the convergence of all MCMs. In fact, even with numerical integration, enhanced convergence is by no means assured in all situations with the näive use of quasirandom numbers, [1,8].

In this paper we study the applicability of quasirandom sequences for solving some linear algebra problems. We have already produced encouraging theoretical and empirical results with QMCMs for linear algebra problems and we believe that this initial work can be improved.

Solving Systems of Linear Algebraic Equations via Neumann Series

Assume that a system of linear algebraic equations (SLAE) can be transformed into the following form: $x = Ax + \varphi$, where A is a real square, $n \times n$, matrix,

[1] Of course, the N optimal quasirandom points in $[0, 1)$ are the obvious: $\frac{1}{(N+1)}, \frac{2}{(N+1)}, \dots \frac{N}{(N+1)}$.

$x = (x_1, x_2, ..., x_n)^t$ is the $1 \times n$ solution vector and $\varphi = (\varphi_1, \varphi_2, ..., \varphi_n)^t$ is the given right-hand side vector.[2] In addition, assume that A satisfies either the condition $\max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}| < 1$, or, that all the eigenvalues of A lie within the unit circle.

Now consider the sequence $x^{(1)}, x^{(2)}, \ldots$ defined by the following recursion:

$$x^{(k)} = Ax^{(k-1)} + \varphi, \quad k = 1, 2, \ldots .$$

Given initial vector $x^{(0)}$, the approximate solution to the system $x = Ax + \varphi$ can be developed via a truncated Neumann series:

$$x^{(k)} = \varphi + A\varphi + A^2\varphi + \ldots + A^{(k-1)}\varphi + A^k x^{(0)}, \quad k > 0 \tag{3}$$

with a truncation error of $x^{(k)} - x = A^k(x^{(0)} - x)$.

This iterative process (3) of applying the matrix A repeatedly is the basis for deriving a Monte Carlo approach for this problem.

The Monte Carlo Method

Consider the problem of evaluating the inner product of a given vector, g, with the vector solution of the considered system

$$(g, x) = \sum_{\alpha=1}^{n} g_\alpha x_\alpha . \tag{4}$$

To solve this problem via a MCM (see, for example, [12]) one has to construct a random process with mean equal to the solution of the desired problem. This requires the construction of a finite-state Markov chain. Consider the following Markov chain:

$$k_0 \to k_1 \to \ldots \to k_i , \tag{5}$$

where $k_j = 1, 2, \ldots, n$ for $j = 1, \ldots, i$ are natural numbers. The rules for constructing the chain (5) are: $P(k_0 = \alpha) = p_\alpha, P(k_j = \beta | k_{j-1} = \alpha) = p_{\alpha\beta}$ where p_α is the probability that the chain starts in state α and $p_{\alpha\beta}$ is the transition probability from state α to state β . Probabilities $p_{\alpha\beta}$ define a transition matrix P. We require that $\sum_{\alpha=1}^{n} p_\alpha = 1$, $\sum_{\beta=1}^{n} p_{\alpha\beta} = 1$ for any $\alpha = 1, 2, ..., n$, and that the distribution $(p_1, ..., p_n)^t$ is permissible to the vector g and similarly the distribution $p_{\alpha\beta}$ is permissible to A [12]. Common constructions are to choose $p_{\alpha\beta} = \frac{|a_{\alpha\beta}|}{\sum_\beta |a_{\alpha\beta}|}$ for $\alpha, \beta = 1, 2, ..., n$, which corresponds to an *importance sampling* MCM (MCM with a reduced variance), or to choose $p_{\alpha\beta} = 1/n$ for $\alpha, \beta = 1, 2, ..., n$ which corresponds to standard MCM.

Now define the random variables $\theta[g]$:

$$\begin{aligned} \theta[g] \quad &= \frac{g_{k_0}}{p_{k_0}} \sum_{j=0}^{\infty} W_j \varphi_{k_j} \\ \text{where } W_0 = 1 \ , \quad &W_j = W_{j-1} \frac{a_{k_{j-1} k_j}}{p_{k_{j-1} k_j}} , \end{aligned} \tag{6}$$

[2] If we consider a given system $Lx = b$, then it is possible to find a non-singular matrix, M, such that $ML = I - A$ and $Mb = \varphi$. Thus without loss of generality the system $Lx = b$ can always be recast as $x = Ax + \varphi$.

It is known [12] that the mathematical expectation $E[\theta[g]]$ of the random variable $\theta[g]$ is:

$$E[\theta[g]] = (g, x).$$

The partial sum corresponding to (6) is defined as $\theta_i[g] = \frac{g_{k_0}}{p_{k_0}} \sum_{j=0}^{i} W_j \varphi_{k_j}$. Thus the Monte Carlo estimate for (g, x) is $(g, x) \approx \frac{1}{N} \sum_{s=1}^{N} \theta_i[g]_s$, where N is the number of chains and $\theta_i[g]_s$ is the value of $\theta_i[g]$ taken over the s-th chain, and a statistical error of size $O(Var(\theta_i)^{1/2} N^{-1/2})$.

Computing the Extremal Eigenvalues

Let A be an $n \times n$ large, sparse, matrix. Consider the problem of computing one or more eigenvalues of A, i.e., the values of λ for which $Au = \lambda u$ holds. Suppose the eigenvalues are ordered $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_{n-1}| > |\lambda_n|$. There are two deterministic numerical methods that can efficiently compute only the extremal eigenvalues - the *power method* and *Lanczos-type methods*. (Note that, the Lanczos method is applicable to only *symmetric* eigenproblems, [4].)
Computational Complexity: If k iterations are required for convergence, the number of arithmetic operations is $O(kn^2)$ for the power method and $O(n^3 + kn^2)$ for both the inverse and inverse shifted power method.

The Monte Carlo Method

Consider MCMs based on the power method. When computing eigenvalues, we work with the matrix A and its *resolvent* matrix $R_q = [I - qA]^{-1} \in I\!R^{n \times n}$. If $|q\lambda| < 1$, R_q may be expanded as a series via the binomial theorem:

$$[I - qA]^{-m} = \sum_{i=1}^{\infty} q^i C_{m+i-1}^{i}, \quad |q\lambda| < 1. \tag{7}$$

The eigenvalues of the matrices R_q and A are connected by the equality $\mu = \frac{1}{1-q\lambda}$, and the eigenvectors of the two matrices coincide[3]. Let $f \in I\!R^n, h \in I\!R^n$. Applying the power method, ([2]), leads to the following iterative processes:

$$\lambda^{(m)} = \frac{(h, A^i f)}{(h, A^{i-1} f)} \underset{m \to \infty}{\longrightarrow} \lambda_{max} \tag{8}$$

$$\mu^{(m)} = \frac{([I - qA]^{-m} f, h)}{([I - qA]^{-(m-1)} f, h)} \underset{m \to \infty}{\longrightarrow} \mu_{max} = \frac{1}{1 - q\lambda}. \tag{9}$$

Construct the same Markov chain as before with the initial density vector, $p = \{p_\alpha\}_{\alpha=1}^{n}$, and the transition density matrix, $P = \{p_{\alpha\beta}\}_{\alpha\beta=1}^{n}$. Define the following

[3] If $q > 0$, the largest eigenvalue μ_{max} of the resolvent matrix corresponds to the largest eigenvalue, λ_{max}, of the matrix A, but if $q < 0$, then μ_{max}, corresponds to the smallest eigenvalue, λ_{min}, of the matrix A.

random variable: $W_0 = \frac{h_{k_0}}{p_{k_0}}$, $W_j = W_{j-1}\frac{a_{k_{j-1}k_j}}{p_{k_{j-1}k_j}}$, $j = 1, \dots, i$. This has the desired expected values ([2]):

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

$$E[\sum_{i=0}^{\infty} q^i C^i_{i+m-1} W_i f(x_i)] = (h, [I - qA]^{-m} f), \;\; m = 1, 2, \dots,$$

and allows us to estimate the desired eigenvalues as:

$$\lambda_{max} \approx \frac{E[W_i f_{k_i}]}{E[W_{i-1} f_{k_{i-1}}]}. \tag{10}$$

and

$$\lambda \approx \frac{1}{q}\left(1 - \frac{1}{\mu^{(m)}}\right) = \frac{E[\sum_{i=1}^{\infty} q^{i-1} C^{i-1}_{i+m-2} W_i f(x_i)]}{E[\sum_{i=0}^{\infty} q^i C^i_{i+m-1} W_i f(x_i)]}. \tag{11}$$

We remark that in (10) the length of the Markov chain, l, is equal to the number of iterations, i, in the power method. However in (11) the length of the Markov chain is equal to the number of terms in truncated series for the resolvent matrix. In this second case the parameter m corresponds to the number of iterations.

Table 1. Monte Carlo estimations using PRNs and QRN sequences for the dominant eigenvalue of matrices of size 128 and 2000

	PRN	*QRN(Faur)*	*QRN(Sobol)*	*QRN(Halton)*
Estimated $\lambda 128_{max}$	61.2851	63.0789	63.5916	65.1777
Relative Error	0.0424	0.0143	0.0063	0.0184
Estimated $\lambda 2000_{max}$	58.8838	62.7721	65.2831	65.377
Relative Error	0.0799	0.01918	0.0200	0.0215

Quasi-Monte Carlo Methods for Matrix Computations

Recall that power method iterations are based on computing $h^T A^i f$ (see (8) and (9)). Even if we are interested in evaluating the inner product (4), substituting x with $x^{(k)}$ from (3) will give $(g, x) \approx g^T\varphi + g^T A\varphi + g^T A^2\varphi + \dots + g^T A^{(k-1)}\varphi + g^T A^k x^{(0)}$, $k > 0$. Define the sets $G = [0, n)$ and $G_i = [i-1, i)$, $i = 1, \dots, n$, and likewize define the piecewise continous functions $f(x) = f_i$, $x \in G_i, i = 1, \dots, n$, $a(x, y) = a_{ij}, x \in G_i, y \in G_j$, $i, j = 1, \dots, n$ and $h(x) = h_i$, $x \in G_i, i = 1, \dots, n$.

Computing $h^T A^i f$ is equivalent to computing an $(i+1)$-dimensional integral. Thus we may analyze using QRNs in this case with bounds from numerical integration. We do not know A^i explicitly, but we do know A and can use a random walk on the elements of the matrix to compute approximately $h^T A^i f$.

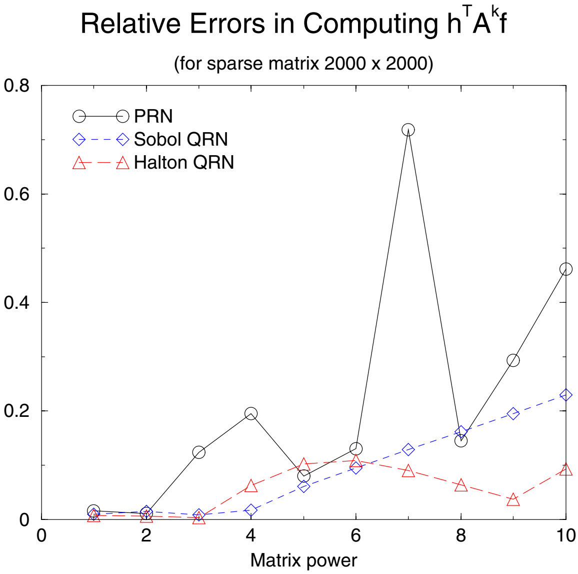

Fig. 1. Relative errors in computing $h^T A^k f$ for $k = 1, 2, \ldots, 10$ for a sparse matrix 2000 × 2000. The corresponding Markov chains are realized using PRN, Soboĺ and Halton sequences

Consider $h^T A^i f$ and an $(i+1)$-dimensional QRN sequence. Normalizing A with $\frac{1}{n}$, and h and f with $\frac{1}{\sqrt{n}}$, we have the following error bound (for proof see [7]):

$$|h_N^T A_N^l f_N - \frac{1}{N}\sum_{s=1}^{N} h(x_s)a(x_s,y_s)\ldots a(z_s,w_s)f(w_s)| \leq |h|^T |A|^l |f| D_N^*.$$

If A is a general sparse matrix with d nonzero elements per row, and $d \ll n$, then *importance sampling method* can be used; the normalizing factors in the error bound (3) are then $1/d$ for the matrix and $\frac{1}{\sqrt{(n)}}$ for the vectors.

2 Numerical Results

Why are we interested in quasi-MCMs for the eigenvalue problem? Because the computational complexity of QMCMs is bounded by $O(lN)$ where N is the number of chains, and l is the mathematical expectation of the length of the Markov chains, both of which are independent of matrix size n. This makes QMCMs very efficient for large, sparse, problems, for which deterministic methods are not computationally efficient.

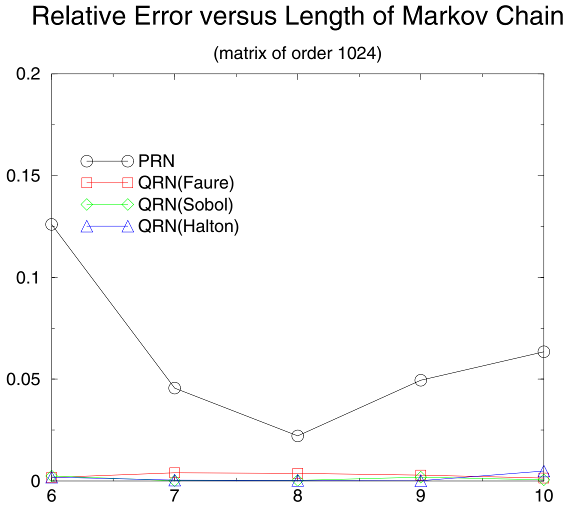

Fig. 2. Relative errors in computing λ_{max} using different length of Markov chains for a sparse matrix 1024 × 1024. The random walks are realized using PRN, Faure, Soboĺ and Halton sequences

Numerical tests were performed on general sparse matrices using PRNs and Soboĺ, Halton and Faure QRNs. The relative errors in computing $h^T A^k f$ with A a sparse matrix of order 2000 and $h = f = (1, 1, \dots, 1)$, are presented in *Figure 1.* The results confirm that the QRNs produce higher precision results than PRNs. The more important fact is the smoothness of the quasirandom "iterations" with k. This is important because these eigenvalue algorithms compute a Raleigh quotient which requires the division of values from consecutive iterations.
The estimated λ_{max} and the corresponding relative errors using MCM and QMCM are presented in *Table 1.* The exact value of λ_{max} for all test matrices is 64.0000153. The results show improvement of the accuracy. Numerical experiments using resolvent MCM and resolvent QMCM have been also performed - the relative errors in computing λ_{max} using Markov chains with different lengths are presented in *Figures 2 and 3.*

Acknowledgements

This paper is based upon work supported by the North Atlantic Treaty Organization under a Grant awarded in 1999.

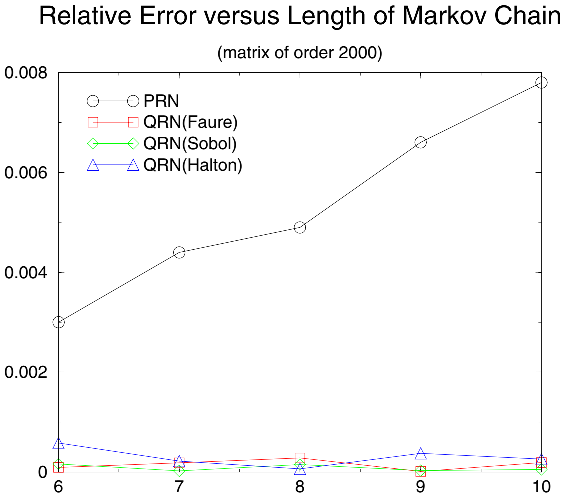

Fig. 3. Relative errors in computing λ_{max} using different length of Markov chains for a sparse matrix 2000 × 2000. The random walks are realized using PRN, Faure, Sobol´ and Halton sequences

References

1. R. E. Caflisch, "Monte Carlo and quasi-Monte Carlo methods," *Acta Numerica*, **7**: 1–49, 1998. 553
2. I. Dimov, A. Karaivanova, "Parallel computations of eigenvalues based on a Monte Carlo approach," *J. of MC Methods and Appl.*, **4**, Num.1: 33–52, 1998. 555, 556
3. H. Faure, "Discrépance de suites associées à un système de numération (en dimension s)," *Acta Arithmetica*, **XLI**: 337–351, 1992. 553
4. G. H. Golub, C. F. Van Loan, "Matrix computations", *The Johns Hopkins Univ. Press*, Baltimore, 1996. 555
5. J. H. Halton, "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numer. Math.*, **2**: 84–90, 1960. 553
6. J. F. Koksma, "Een algemeene stelling uit de theorie der gelijkmatige verdeeling modulo 1," *Mathematica B (Zutphen)*, **11**: 7–11, 1942/43. 553
7. M. Mascagni, A. Karaivanova, "Are Quasirandom Numbers Good for Anything Besides Integration?" to appear in *Proc. of Advances in Reactor Physics and Mathematics and Computation into the Next Millennium (PHYSOR2000)*, 2000. 557
8. B. Moskowitz, R. E. Caflisch, "Smoothness and dimension reduction in quasi-Monte Carlo methods", *J. Math. Comput. Modeling*, **23**: 37–54, 1996. 553
9. H. Niederreiter, *Random number generation and quasi-Monte Carlo methods*, SIAM: Philadelphia, 1992. 553
10. K. F. Roth, "On irregularities of distribution," *Mathematika*, **1**: 73–79, 1954. 553
11. I. M. Sobol´, "The distribution of points in a cube and approximate evaluation of integrals," *Zh. Vychisl. Mat. Mat. Fiz.*, **7**: 784–802, 1967. 553
12. I. M. Sobol´ *Monte Carlo numerical methods*, Nauka, Moscow, 1973. 554, 555