UNIFORM SAMPLING METHODS FOR VARIOUS COMPACT SPACES

UNIFORM SAMPLING METHODS FOR VARIOUS COMPACT SPACES

By

SEAN O'HAGAN, B.SC.

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
AT
MCMASTER UNIVERSITY
HAMILTON, ONTARIO
APRIL 2007

MCMASTER UNIVERSITY

DEGREE:	Master of Science, 2007
DEPARTMENT:	Department of Mathematics and Statistics, Hamilton, Ontario
UNIVERSITY:	McMaster University
TITLE:	Uniform Sampling Methods for various Compact Spaces
AUTHOR:	Sean O'Hagan, B.Sc.(Laurentian University)
SUPERVISOR:	Dr. Maung Min-Oo
PAGES:	vii, 71

Table of Contents

Abstract

We look at methods to generate uniformly distributed points from the classical matrix groups, spheres, projective spaces, and Grassmannians. We motivate the discussion with a number of applications ranging from number theory to wireless communications. The uniformity of the samples and the efficiency of the algorithms are compared.

Acknowledgements

I would like to thank my wife and my parents for their support and encouragement, and I hope that one day, the wonder of mathematics hits them as it has me. Thanks to my supervisor Dr. Maung Min-Oo for his guidance and support, and to committee members Dr. Andy Nicas and Dr. Angelo Canty for their time and comments.

Chapter 1

Introduction

This project looks at sampling uniformly distributed random points from various familiar compact spaces. We begin with two of the classical matrix groups: the orthogonal and unitary groups. On each of these, we look at a number of sampling methods. In particular, we discuss the Gram-Schmidt method, the Householder reflection method, the subgroup method, and the random reflection method. We compare the speeds of each method, and to test the uniformity of the resulting samples, we look at the distribution of eigenvalues on the unit circle, eigenvalue phase histograms, and the distribution of traces.

We next look at spheres and consider the following methods: the rejection method, quaternionic method, Gaussian method, and two uniform methods. Noting that the sphere is a homogeneous space of the orthogonal group, we use its sampling methods to sample points from the sphere. We continue with projective spaces and Grassmannians, in each case comparing the various methods for speed and uniformity.

In most cases, we begin a chapter with a description of an application of random sampling from the space in question. Applications include digitized speech encryption, applied optics, the grand tour, and wireless communications. Finally, we make a few observations regarding suggested methods for each space.

Chapter 2

The Orthogonal Group

A scheme for a speech encryption method attempted to use random orthogonal matrices to scramble digitized speech before transmission [21]. The sender rotates a block of Fourier-transformed, digitized speech (in the form of a vector in $\mathbb{R}^{256}$) using a 256 × 256 orthogonal matrix, and the receiver applies its transpose to decrypt it. The method is slow (i.e. $0(n^3)$) but extremely secure. In an attempt to speed up the process, products of random reflections were looked at, one of the methods considered below.

We begin with some required theory. First, what is the orthogonal group, and how can we hope to sample uniform orthogonal matrices from it?

Definition 1. *An* **orthogonal matrix** *is a square matrix O with real entries such that*

$$O^TO = OO^T = I$$

The set of all $n \times n$ orthogonal matrices is a compact topological group denoted $O(n)$, and is called the **orthogonal group**. *Elements of $O(n)$ are either rotations or reflections. The orthogonal group is a compact Lie group, ie. a smooth manifold with a smooth group operation.*

Every compact Lie group has a unique invariant measure called Haar measure. On Lie groups, the measure is both left and right-invariant. Since the measure is invariant under group multiplication, it assigns the same "volume" to similar regions in $O(n)$. Because of this, Haar measure normalized to 1 is a natural choice for a probability measure for $O(n)$.

Definition 2. **Haar measure** *is a method of assigning an invariant "volume" to Borel subsets of locally compact topological groups. That is, if B is a Borel subset of the locally compact topological group G and the Haar measure of B is written $d\mu(B)$, then*

$$d\mu(KB) = d\mu(BL) = d\mu(B) \qquad K, L \in G$$

2.1 Sampling Methods

2.1.1 The Gram-Schmidt Method

The first orthogonal group-sampling method employs the QR decomposition via the Gram-Schmidt algorithm [8] and makes use of the fact that the joint probability density function of the elements of a matrix from the real Ginibre ensemble, is invariant under orthogonal transformations [16]. This ensemble is named after Jean Ginibre who studied statistical ensembles of complex, quaternionic, and real matrices in the 1960s [12].

Definition 3. *The real $n \times n$* **Ginibre ensemble** *is the collection of all invertible matrices in $GL(n, \mathbb{R})$ where the matrix elements are independent identically distributed standard normal random variates. We will occasionally refer to this ensemble by $G_1(n, n)$.*

Hence, the probability density function of a matrix element is simply $p(x_{jk}) = \frac{1}{\sqrt{2\pi}} e^{-x_{jk}^2/2}$. Since the elements are independent, the joint probability density function is

$$\begin{aligned} P(X) &= \frac{1}{\sqrt{2\pi}^{n^2}} \prod_{j,k=1}^{n,n} e^{-x_{jk}^2/2} \\ &= \frac{1}{\sqrt{2\pi}^{n^2}} \exp\left(-\frac{1}{2}\sum_{j,k=1}^{n,n} x_{jk}^2\right) \\ &= \frac{1}{\sqrt{2\pi}^{n^2}} \exp\left(-\frac{1}{2}\mathrm{Tr}(X^TX)\right) \end{aligned}$$

We can write $d\mu_{G_1}(X)$ or $P(X)dX$ to refer to the measure of sets of the ensemble depending on the context.

Lemma 1. *The measure of the real Ginibre ensemble is invariant under orthogonal transformations.*

Proof. Let $X \in G_1(n,n)$ and $A \in O(n)$. Since $A^TA = I$,

$$\begin{aligned} P(AX) &= \frac{1}{\sqrt{2\pi}^{n^2}} \exp(-\tfrac{1}{2}\mathrm{Tr}(X^TA^TAX)) \\ &= \frac{1}{\sqrt{2\pi}^{n^2}} \exp(-\tfrac{1}{2}\mathrm{Tr}(X^TX)) \\ &= P(X) \end{aligned}$$

Now consider the map

$$X \mapsto AX$$

and let $Y = AX$. The Jacobian of the transformation is

$$\left|\frac{\partial(y_{11},\ y_{12},\ \ldots,\ y_{nn})}{\partial(x_{11},\ x_{12},\ \ldots,\ x_{nn})}\right|$$

which is the determinant of an $n^2 \times n^2$ orthogonal matrix (the matrix is composed of n shuffled copies of A) and is hence equal to 1. Therefore the joint probability density function of the y_{ij} is

$$P(Y) = P(X) \div \left|\frac{\partial(y_{ij})}{\partial(x_{ij})}\right| = P(X)$$

Thus the probability density function and the measure are left-invariant under orthogonal transformations. Right-invariance is proven in a similar way. □

We can now describe the method. Let X be a random matrix from the real $n \times n$ Ginibre ensemble. Next, we orthonormalize X. The standard Gram-Schmidt process is numerically unstable due to rounding errors which cause the vectors to be not quite orthogonal. The following modification introduces smaller errors in the Matlab algorithm.

Label the columns of X as $X_1,\ X_2,\ \ldots,\ X_n$. Set

$$\begin{array}{lcl}
Y_1 & = & X_1 \\
Y_2 & = & X_2 - \mathrm{proj}_{Y_1} X_2 \\
Y_3^{(1)} & = & X_3 - \mathrm{proj}_{Y_1} X_3 \\
Y_3 & = & Y_3^{(1)} - \mathrm{proj}_{Y_2} Y_3^{(1)} \\
\multicolumn{3}{l}{(\text{and for } 4 \leq k \leq n)} \\
Y_k^{(1)} & = & X_k - \mathrm{proj}_{Y_1} X_k \\
Y_k^{(2)} & = & Y_k^{(1)} - \mathrm{proj}_{Y_2} Y_k^{(1)} \\
 & \vdots & \\
Y_k^{(k-2)} & = & Y_k^{(k-3)} - \mathrm{proj}_{Y_{k-2}} Y_k^{(k-3)} \\
Y_k & = & Y_k^{(k-2)} - \mathrm{proj}_{Y_{k-1}} Y_k^{(k-2)} .
\end{array}$$

Normalize the Y_i

$$Q_i = \frac{Y_i}{||Y_i||} \qquad 1 \leq i \leq n$$

and let $Q = [Q_1\ Q_2\ \cdots\ Q_n]$. The claim is that Q is uniformly distributed on $O(n)$.

Theorem 1. *Let X be a random matrix from the $n \times n$ real Ginibre ensemble, and let Q be the random matrix resulting from applying the Gram-Schmidt process to X. Then Q is distributed uniformly on $O(n)$ with respect to Haar measure.*

Proof. Let B be a Borel set in $O(n)$ and let $O \in O(n)$. Write Q_X for Q to specify the source matrix. Then

$$\begin{aligned} d\mu_Q(B) &= P(Q_X \in B) \\ &= P(Q_{OX} \in B) && \text{(from Lemma 1)} \\ &= P(OQ_X \in B) && \text{(linearity of Gram} - \text{Schmidt)} \\ &= P(Q_X \in O^T B) \\ &= d\mu_Q(O^T B) \end{aligned}$$

Hence $d\mu_Q$ is a left-invariant measure on $O(n)$ and is therefore the unique Haar measure. So Q is uniformly distributed on $O(n)$. [10] □

The Matlab code implementing the method is as follows (here, `n` refers to the orthogonal group $O(n)$ and `m` is the number of samples):

```
function gramschmidt(n,m)
    for p=1:m
        X=randn(n,n);
        for j=1:n
            for k=1:(j-1)
                X(:,j)=X(:,j)-dot(X(:,j),X(:,k))*X(:,k);
            end
            X(:,j)=1/norm(X(:,j))*X(:,j);
        end
        Q(:,:,p)=X;
    end
```

The matrix `Q` contains the random samples.

2.1.2 The Householder Reflection Method

In this section, following [22] and [16] we take a look at the inside of a different algorithm used in the generation of random matrices from the classical matrix groups, namely the QR decomposition of a matrix into the product of an orthogonal matrix and an upper-right-triangular matrix. Matlab uses the following Householder technique to implement this decomposition, as the Gram-Schmidt process is not as numerically stable.

Definition 4. *The* **Householder matrix** H_{x} *associated with the vector* $\mathbf{x}$ *is defined as follows. Let* $r = ||\mathbf{x}||$ *and* $\mathbf{u} = \mathbf{x} - r\mathbf{e}_1$ *where* $\mathbf{e}_1$ *is the first standard basis vector of* $\mathbb{R}^n$*. Normalize* $\mathbf{u}$ *by setting* $\mathbf{v} = \frac{\mathbf{u}}{||\mathbf{u}||}$ *and set* $H_{\mathrm{x}} = I - 2\mathbf{v}\mathbf{v}^T$*.*

The matrix H_{x} has the property that it zeroes all but the first entry of $\mathbf{x}$ so that $H_{\mathrm{x}}\mathbf{x} \in span\{\mathbf{e}_1\}$ (see Lemma 11 in the Appendix).

To perform the QR decomposition of X using Householder matrices, do the following. Let $X = [X_1\ X_2\ \ldots\ X_n]$ be an invertible matrix and let $r = ||X_1||$. Further, let $\mathbf{u} = X_1 - r\mathbf{e}_1$ and $\mathbf{v} = \frac{\mathbf{u}}{||\mathbf{u}||}$. Finally, define $H_1 = I - 2\mathbf{v}\mathbf{v}^{\mathbf{T}}$. This is the Householder matrix related to X_1, in that

$$H_1 X = \left[\begin{array}{c|c} r & \bullet \ \cdots \ \bullet \\ \hline 0 & \\ \vdots & X' \\ 0 & \end{array}\right]$$

H_1 knocks out all but the top entry (itself being set to the norm of X_1) in the first column X_1 of X. (Note that H_1 is both symmetric and orthogonal. See Lemma 9 and 10 in the Appendix.) Repeat the above procedure with X', resulting in an $(n-1) \times (n-1)$ matrix H_2'. Now, H_2' knocks out all but the top entry in the first column of X'. Set H_2 equal to

$$H_2 = \left[\begin{array}{c|c} 1 & 0 \ \cdots \ 0 \\ \hline 0 & \\ \vdots & H_2' \\ 0 & \end{array}\right]$$

Continue this process until the matrices $H_1, H_2, \ldots, H_{n-1}$ are obtained. Then it's clear that

$$H_{n-1} \ldots H_2 H_1 X = R$$

where R is upper-right-triangular. Since $H_i^{-1} = H_i^T = H_i$

$$X = H_1 H_2 \dots H_{n-1} R$$

Setting

$$Q = H_1 H_2 \dots H_{n-1}$$

we obtain a QR decomposition of X, namely $X = QR$.

With this procedure in hand, how does one go about generating random elements from $O(n)$? Since any invertible matrix X can be split into the product of an orthogonal matrix and an upper right-triangular matrix via the QR decomposition, perhaps we can generate random elements from $GL(n, \mathbb{R})$ and decompose them to produce orthogonal samples. However, since $GL(n, \mathbb{R})$ is non-compact, we cannot put a uniform measure on it, and hence cannot generate random matrices from it. As an example, on the real line, we can assign a constant probability density function to a closed interval, but we cannot to the whole real line, since the associated cumulative distribution function would diverge. So we return to the normal distribution, and slightly modify the QR decomposition.

Theorem 2. *Let X be an $n \times n$ matrix with each entry picked independently from the standard normal distribution (as in Lemma 1) and let $X = QR$ be the QR^+ decomposition of X (defined right after the proof). Then Q is an orthogonal matrix distributed uniformly with respect to Haar measure.*

Proof. Let $H \in O(n)$. By Lemma 1, the matrix HX has the same distribution as X. Since $X = QR$ (where the QR^+ decomposition is the unique decomposition such that R has positive diagonal entries) it follows that $HX = (HQ)R$ is the QR^+ decomposition of HX. Now Q and HQ are two random matrices resulting from an identical algorithm (QR^+) and whose source matrices X and HX are identically distributed. Hence Q and HQ are identically distributed. This means that the distribution of Q on $O(n)$ is left-invariant since H was arbitrary. Hence Q is uniformly distributed with respect to Haar measure. □

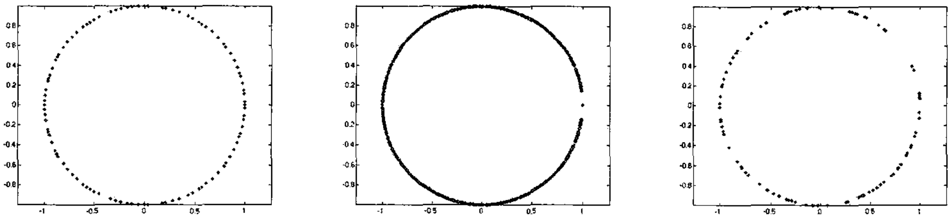

Figure 2.1: Eigenvalue plots from QR^+, QR, and random points on a circle

The QR^+ decomposition involves an extra step to ensure that the decomposition is unique. To see why this step is necessary, and to witness the non-uniformity of the samples from unmodified method, the following numerical experiments were performed. First, matrices were generated from $O(100)$ using both the QR^+ method and the QR method. The eigenvalues of these matrices come in pairs of complex conjugates along with ± 1 with magnitudes of 1, and are distributed on the unit circle in the complex plane. The image on the left of Figure 2.1 is an eigenvalue plot of a Haar-distributed matrix. The eigenvalues appear relatively uniformly distributed. The image in the middle shows a superposition of ten matrices generated using the standard QR method. It is clear that the eigenvalues are not uniformly distributed. For comparison, 100 random points on a circle are displayed at right. In the second experiment, two sets of $50,000$ matrices were randomly generated from $O(40)$ using both QR methods. The phases of the eigenvalues (ie. the angles between $-\pi$ and π between the line joining the origin to the eigenvalue and the real axis) were collected and plotted on two histograms. Since ± 1 are always eigenvalues for even orthogonal matrices, we omit them from our calculations. It is clear from Figure 2.2 that QR^+ produces Haar-distributed orthogonal matrices.

As mentioned, the QR decomposition is not unique. Let $X \in GL(n, \mathbb{R})$ and decompose X into QR, where $Q \in O(n)$ and $R \in R(n)$, the group of

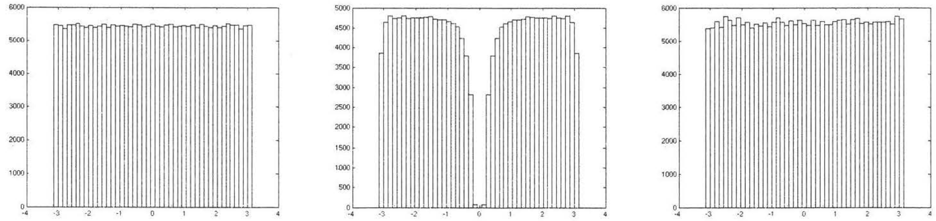

Figure 2.2: Phase histograms from QR^+, QR and random points on a circle

invertible upper-right-triangular matrices. If

$$\Lambda = \begin{bmatrix} \pm 1 & & & \mathbf{0} \\ & \pm 1 & & \\ & & \ddots & \\ \mathbf{0} & & & \pm 1 \end{bmatrix}$$

then $Q' = Q\Lambda$ and $R' = \Lambda R$ are still orthogonal and upper-right-triangular.

$$Q'R' = Q\Lambda\Lambda R = QR = X \qquad \text{since } \Lambda = \Lambda^{-1}$$

so the decomposition represents a multi-valued map

$$QR : GL(n, \mathbb{R}) \to O(n) \times R(n)$$

The goal is to alter the mapping to make it single-valued and one-to-one. This last property will allow one to be confident that regardless of which algorithm is used to implement QR, we always obtain the same Q.

Lemma 2. *If $X = QR = Q'R'$, then $Q' = Q\Lambda$ and $R' = \Lambda R$ where $\Lambda \in \Lambda(n)$, the group of all orthogonal diagonal matrices.*

Proof.

$$QR = Q'R' \quad \Rightarrow \quad Q^{-1}Q' = RR'^{-1}$$

This group element lies in $O(n) \cap R(n)$. If it is both orthogonal and upper-right-triangular, then its inverse is its transpose as well as being upper-right-triangular. So it must be diagonal. Hence

$$\Lambda = Q^{-1}Q' \Rightarrow Q' = Q\Lambda \text{ and } \Lambda = \mathrm{RR}'^{-1} \Rightarrow \mathrm{R}' = \Lambda\mathrm{R}$$

where $\Lambda \in \Lambda(n)$. □

Instead of the multi-valued map QR, this lemma points to the following map

$$QR^+ : GL(n, \mathbb{R}) \quad \longrightarrow \quad O(n) \times \; R(n)/\Lambda(n)$$

We need to define QR^+ in such a way that

$$X \mapsto (Q, \lambda T) \quad \Rightarrow \quad OX \mapsto (OQ, \lambda T) \qquad (*)$$

for any $O \in O(n)$. This would imply that multiplication of X by an orthogonal matrix reduces to the left action of $O(n)$ on itself, post decomposition.

To construct QR^+, choose a class of representatives of $R(n)/\Lambda(n)$ by imposing that the main diagonal of R have only positive entries as previously mentioned. Using Lemma 2, we can uniquely factorize any $X \in GL(n, \mathbb{R})$ such that the main diagonal of R has this property. If $X = QR$, then $OX = OQR$, $O \in O(n)$. This decomposition of OX is unique within the specified class of representatives of $R(n)/\Lambda(n)$. So, our map QR^+ satisfies $(*)$. Recall that Lemma 1 states that $d\mu_G(OX) = d\mu_G(X)$ for any $O \in O(n)$. Thus, the induced measure on the left factor $O(n)$ of QR^+ is also invariant under left multiplication, and hence must be Haar. We summarize this in the following theorem.

Theorem 3. *If the map QR^+ is such that*

$$X \mapsto (Q, \lambda T) \quad \Rightarrow \quad OX \mapsto (OQ, \lambda T)$$

then QR^+ decomposes the measure of the real Ginibre ensemble $d\mu_G$ as

$$d\mu_G(X) = d\mu_{Haar}(Q) \times d\mu_{R(n)/\Lambda(n)}(\lambda T)$$

Proof. The proof is a measure-theoretic one.

$$\begin{aligned}
d\mu_G(X) &= d\mu_G(OX) && \text{(Lemma 1)}\\
&= d\mu(OQ, \lambda T) && \text{(d}\mu\text{ is measure on prod. sp.)}\\
&= d\mu(Q, \lambda T) && \text{(from (}*\text{))}\\
&= d\mu_{Haar}(Q) \times d\mu_{R(n)/\Lambda(n)}(\lambda T) && \text{(uniqueness of Haar measure)}
\end{aligned}$$

□

The Matlab code implementing the method is as follows (here, `n` refers to the orthogonal group $O(n)$ and `m` is the number of samples):

```
function householder(n,m)
    for p=1:m
        X(:,:,p)=randn(n,n);
        [Q(:,:,p),R(:,:,p)]=qr(X(:,:,p));
        J=eye(n);
        for j=1:n
            if (R(j,j,p)<0)
                J(j,j)=-1;
            end
        end
        Q(:,:,p)=Q(:,:,p)*J;
    end
```

The matrix `Q` contains the random samples.

2.1.3 The Subgroup Method

The Householder method is a particular instance of a more general method useful for sampling from groups or compact topological groups. This is the subgroup method, introduced by Diaconis and Shahshahani in [9]. We first look at the finite version.

Let G be a finite group and let

$$G = G_0 \supset G_1 \supset \cdots \supset G_n$$

be a nested chain of subgroups. Let C_i denote coset representatives of G_{i+1} in G_i, $0 \leq i < n$. We can then represent G as

$$G \simeq C_0 \times C_1 \times \cdots \times C_{n-1} \times G_n$$

where each $g \in G$ has a unique representation as $g_0 g_1 \dots g_n$ where $g_i \in C_i$ and $g_n \in G_n$. If the g_i are chosen uniformly from the factors and multiplied together, then the product will be uniformly distributed on G.

This algorithm can be similarly used to produce random samples from any compact topological group. Here we apply it to the orthogonal group. $O(n)$ has the following nested chain of subgroups

$$O(n) \supset O(n-1) \supset \cdots \supset O(2)$$

with $O(k-1)$ the subgroup of $O(k)$ fixing $e_1 \in O(k)$.

We can also simply consider the much shorter chain $O(n-1) \subset O(n)$. If we knew how to generate a random matrix from $O(n-1)$ as well as coset representatives of $O(n-1)$ in $O(n)$ at random, we would have a much more efficient way of generating elements of $O(n)$. Since the coset space $O(n)/O(n-1)$ is equivalent to $\mathbb{S}^{n-1}$ we can pick a point at random from the sphere and use it to generate a coset representative by specifying where e_1 goes. For any $p \in \mathbb{S}^{n-1}$, the Householder reflection $(I - 2x^T x)$ where

$$x = \frac{e_1 - p}{\sqrt{(e_1 - p)(e_1 - p)^T}}$$

takes e_1 to p. This leads to the following.

Lemma 3. *Let p be a random point on $\mathbb{S}^{n-1}$ and let O_{n-1} be a random matrix* *in $O(n-1)$. Then*

$$(I - 2x^T x) \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & O_{n-1} & \\ 0 & & & \end{bmatrix}$$

is uniformly distributed on $O(n)$ where x is as above.

Using induction on n, this becomes Householder's method.

2.1.4 The Random Reflection Method

Matrices from $O(n)$ can be approximated by the product of random reflections (see Theorem 8 in the Appendix). If a point p is sampled uniformly from $\mathbb{S}^{n-1}$ then the reflection in the plane orthogonal to p is

$$R = (I - 2pp^T)$$

Form the product of a number of such R. It is natural to ask how many factors are required so that the resulting orthogonal matrix is close to being distributed with Haar measure. To answer the question it was necessary to look at the distribution of the trace of the matrices, and their powers [6]. If $Q \in O(n)$ is Haar-distributed then $Tr(Q)$ is the sum of a collection of small numbers, whose sum has finite variance, so the central limit theorem should apply.

The method of moments (see Theorem 9 in the Appendix) can be used to show the following:

Lemma 4. *Let Q be a Haar-distributed matrix in $O(n)$. For $0 \leq k \leq 2n+1$,*

$$\int_{O(n)} (TrQ)^k dQ = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x^k e^{-x^2/2} dx$$

The right hand side of this equation is the k^{th} moment of the standard normal distribution. This suggests that the measure of the trace function on $O(n)$ converges to a limiting standard normal distribution.

Comparing the distribution of the traces of the products to that of the random orthogonal matrices Diaconis and Shahshahani [9] concluded that $\frac{1}{2}n\log n + cn$ random reflections were required. However, multiplying these matrices together results in an algorithm with running time $0(n^3\log n)$ since multiplying a matrix by a reflection is an n^2 process. Further, if all that is required is the image of a vector under this product of reflections, then the cost of the algorithm is reduced to $0(n^2\log n)$.

The Matlab code implementing the method is as follows (here, n refers to the orthogonal group $O(n)$, m is the number of samples, c is the constant from above, and sphere() is one of the sphere-sampling methods):

```
function randomrefl(n,m,c)
    numref=ceil((n*log(n))/2+c*n);
    for j=1:m
        X=sphere(n-1,numref);
        Q(:,:,j)=eye(n);
        for k=1:n
            Q(:,:,j)=(eye(n)-2*X(k,:)'*X(k,:))*Q(:,:,j);
        end
    end
```

The matrix Q contains the random samples.

2.1.5 The Normalized Gaussian Method

Another method to approximate random orthogonal matrices which Diaconis alludes to in [9] is to consider matrices from the Ginibre ensemble but whose elements have variance $1/n$. A random matrix X thus defined (although not orthogonal) has most pairs of rows or columns roughly orthogonal. As well, the first $2n$ moments of $Tr(X)$ equal those of $Tr(O)$ where O is a Haar-distributed random orthogonal matrix. However, eigenvalue plots of these matrices show quite uniformly scattered eigenvalues throughout the 1-disc.

2.2 Uniformity Tests

2.2.1 Distribution of Eigenvalues

The joint probability density function of the eigenvalues of a Haar-distributed random matrix in $O(n)$ were given by Weyl in [24]. As an example, consider the odd orthogonal $O(2k+1)$ case. We will split this into its proper part O^+ (also know as $SO(2k+1)$) and its coset O^-. The density function on O^+ is proportional to $\Delta^+\overline{\Delta^+}$ where

$$\Delta^+ = \prod_j s(\phi_j/2) \prod_{j<k} (c(\phi_j) - c(\phi_k))$$

while the density function on O^- is proportional to $\Delta^-\overline{\Delta^-}$ where

$$\Delta^- = \prod_j c(\phi_j/2) \prod_{j<k} (c(\phi_j) - c(\phi_k))$$

and where

$$c(\theta) = e^{i\theta} + e^{-i\theta} \text{ and } s(\theta) = e^{i\theta} - e^{-i\theta}$$

Not much can be gleaned from these formulae, aside from noticing that the eigenvalues will tend to repel each other. (Functions for the even orthogonal case are similar, but their inclusion here would not be illuminating.) That is, as θ_j and θ_k approach each other, the Δ's go to zero. We should expect then, that since the eigenvalues of an orthogonal matrix lie on the unit circle, the eigenvalues of a "typical" element of $O(n)$ for n large, should be quite regularly-spaced. This provides us with a quick visual test to determine whether a given matrix is random. Figures 2.3 to 2.6 show eigenvalue plots of a random

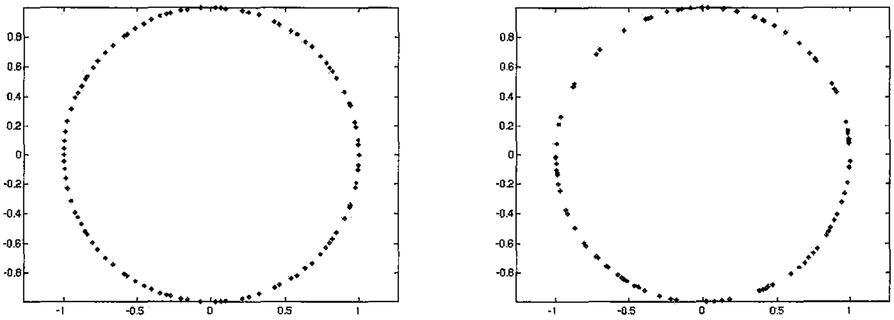

Figure 2.3: Eigenvalues from $O(100)$ using G-S; 100 points on $\mathbb{S}^1$

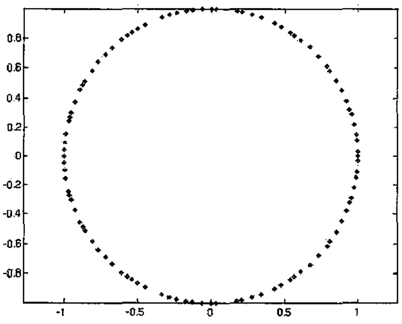

Figure 2.4: Eigenvalues from $O(100)$ using Householder

element generated from $O(100)$ using the various sampling methods compared to 100 randomly plotted points on a circle. It also shows a close-up of a random element from $O(1000)$ compared to that of a thousand random points on a circle. We see noticeable gaps in the random points, and although there are some larger gaps in the eigenvalue plots, they are much smaller.

Thus, a random orthogonal matrix produces near-regularly spaced eigenvalues on the unit circle. What family of matrices (if any) produces randomly-spaced eigenvalues? We will touch briefly on this point further on in our discussion of the unitary group.

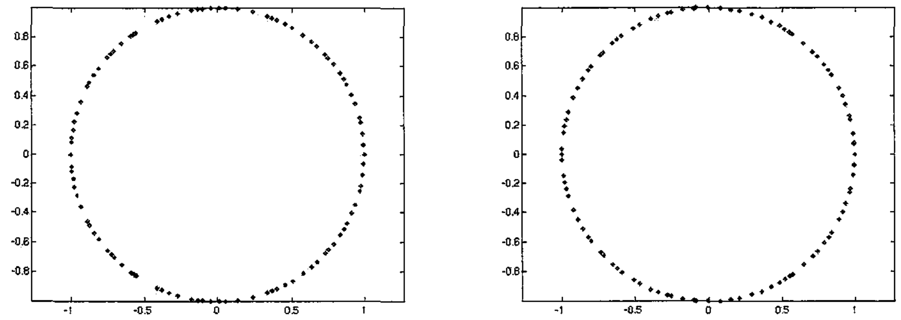

Figure 2.5: Eigenvalues from $O(100)$ using reflections, $c = 5, 50$

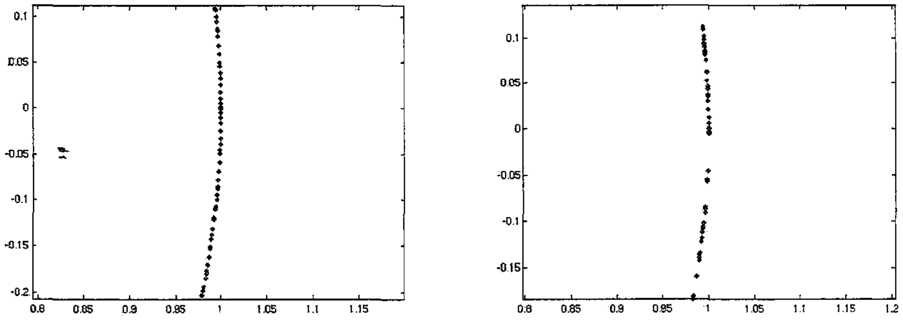

Figure 2.6: Zoom of eigenvalues from $O(1000)$ using G-S

2.2.2 Distribution of Traces

Diaconis and Mallows [7] proved the following:

Theorem 4. *If M is chosen uniformly from $O(n)$, then, as n tends to ∞,*

$$\left| P(trM \leq x) - \int_{-\infty}^{x} \frac{e^{-t^2/2}}{\sqrt{2\pi}} dt \right| \to 0$$

uniformly in x.

Figure 2.7 shows histograms of traces of 1000 random orthogonal matrices generated from $O(100)$, using various methods.

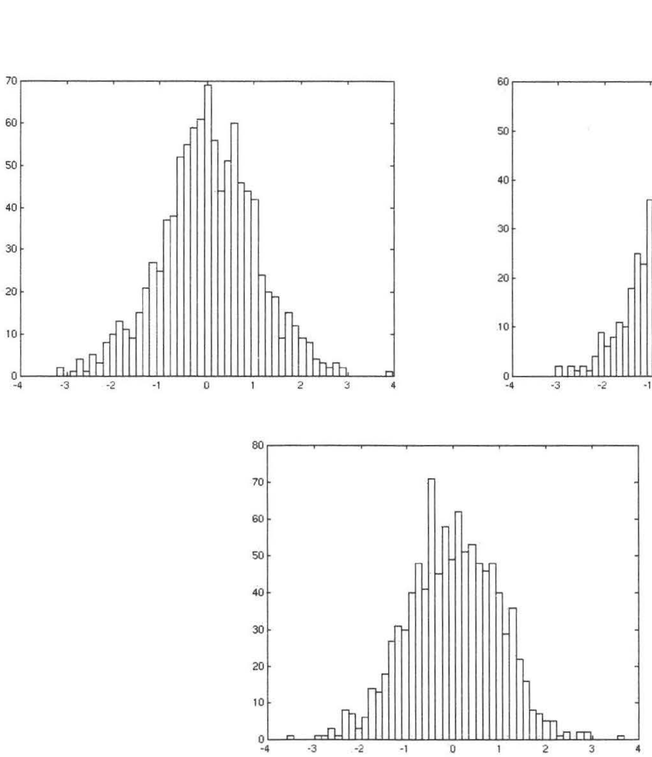

Figure 2.7: Trace plots using G-S, random reflections, and Householder

2.2.3 Eigenvalue Phase Histograms

As we saw in our alteration of the QR-decomposition, the histogram of eigenvalue phases is extremely close to uniform for a random orthogonal matrix. In this section, we compare histograms for the various methods. To create the histograms (see Figure 2.8, we sampled 1000 matrices from $O(50)$. As expected for the Gram-Schmidt and Householder methods, the eigenvalue phase histograms show uniformity in the eigenvalue angles. However, there are large spikes in eigenvalues close to ±1 for the random reflection method. This method seems to generate matrices some of whose eigenvalue phases are very close to 0 and π. The algorithm in fact factors out eigenvalues whose phases are equal to 0 and π, but does not help in the case where the eigenvalues are very close to ±1. It appears that as n increases, the spikes become less prominent, and in fact invert mildly for $O(100)$.

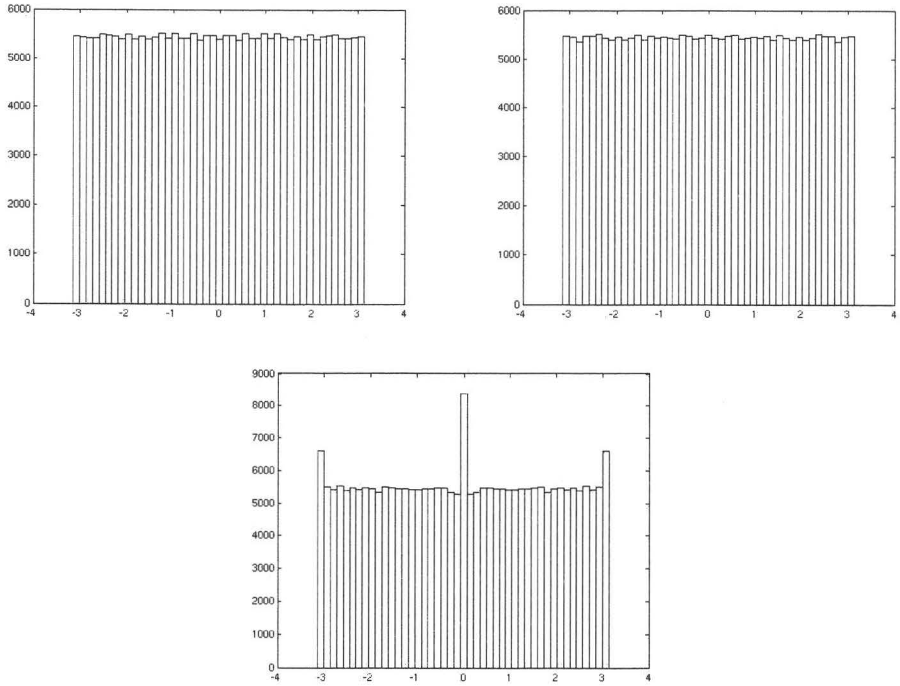

Figure 2.8: Phase histograms for G-S, Householder, and random reflections

2.2.4 Distribution of O_{11}

Another method to test the uniformity of the samples from $O(n)$ is to use a theorem of Borel's [1].

Theorem 5. *If $O \in O(n)$ is a Haar-distributed random matrix, then*

$$P(\sqrt{n}O_{11} \leq x) \rightarrow \int_{-\infty}^{x} e^{-t^2/2} dt$$

where O_{11} is the $(1,1)$ element of O.

Figure 2.9 shows the distribution of this term for each of the orthogonal sampling methods.

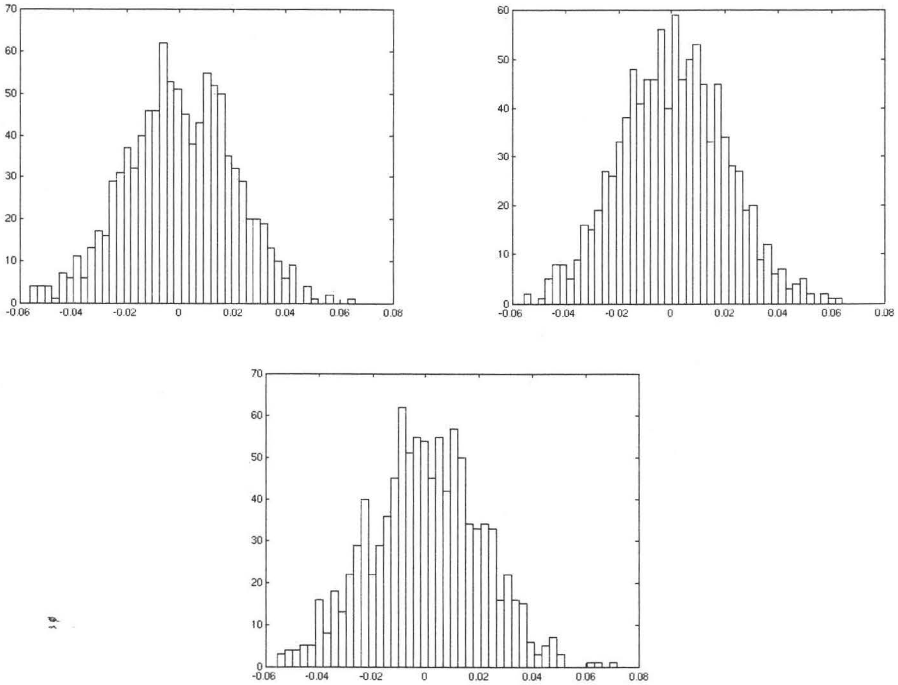

Figure 2.9: $\sqrt{n}O_{11}$ histograms for G-S, Householder, and random reflections

2.3 Speeds of Algorithms

$O(n)$	$O(2)$	$O(3)$	$O(5)$	$O(10)$	$O(20)$	$O(50)$	$O(100)$
Method	**Average time to sample 1000 points**						
Gram-Schmidt	.064s	.14s	.38s	2.30s	11.11s	1m12s	4m15s
Householder	.097s	.14s	.45s	3.54s	17.93s	2m5.9s	7m26s
Reflection	.30s	.44s	.80s	2.91s	11.47s	1m45s	18m19s

Chapter 3

The Unitary Group

Random unitary matrices make an unexpected entrance in the study of the Riemann zeta function [4], [7], [14]. This function is defined as

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = \Pi_{p \in \mathbb{P}} \left(1 - \frac{1}{p^s}\right)^{-1}$$

for $s \in \mathbb{C}$ where $re(s) > 1$. The domain of definition of the zeta function can be extended to the entire complex place (with a simple pole at $s = 1$). The zeros of $\zeta(s)$ are important in that they can provide insight into the distribution of prime numbers. All of the zeros of interest lie in the critical strip $0 < re(s) \leq 1$. The density of the zeros in this strip increases as $im(s)$ increases. Assuming the Riemann hypothesis, all of these zeros lie on the critical line $re(s) = \frac{1}{2}$. The unitary group is brought into play by looking at the density of zeros at any given point high up on the imaginary axis. The local density d of the zeros is calculated, and numerous groups of d successive zeros are tabulated. After some normalization, the groups of zeros are wrapped around the unit sphere. The distribution of the zeros in each group seems to be the same as that of the eigenvalues of $U(d)$, the d-th unitary group. More generally, it may be that random matrix theory can be applied to general families of L-functions. It has also been suggested that the characteristic polynomial of a given matrix

$U \in U(n)$,

$$Z(U, \theta) = \prod_{j=1}^{n} (1 - e^{i\theta_j - \theta})$$

can be used as a model for the zeta function. Moments of the zeta function along $re(s) = \frac{1}{2}$ match moments of $Z(U, \theta)$ over $U(n)$, allowing for very strong predictions about the behaviour of the Riemann zeta function.

Definition 5. *A* **unitary matrix** *is a square matrix U with complex entries such that*

$$U^*U = UU^* = I$$

The set (in fact, group) of all $n \times n$ unitary matrices is denoted $U(n)$, and is called the **unitary group**. *The unitary group is a compact Lie group.*

Again, Haar measure normalized to 1 is a natural choice for a probability measure for $U(n)$.

3.1 Sampling Methods

3.1.1 The Gram-Schmidt Method

This method decomposes random complex matrices from the complex Ginibre ensemble into a unitary matrix and upper right-triangular matrix. The resulting unitary matrix is a Haar-distributed matrix.

Definition 6. *The complex $n \times n$* **Ginibre ensemble** *is the collection of all invertible matrices in $GL(n, \mathbb{C})$ where the matrix elements are independent identically distributed standard complex normal random variates. We will occasionally refer to this ensemble by $G_2(n, n)$.*

Hence, the probability density function of a matrix element is simply $p(z_{jk}) = \frac{1}{\pi} e^{-|z_{jk}|^2}$. Since the elements are independent, the joint probability density function is

$$\begin{aligned} P(Z) &= \tfrac{1}{\pi^{n^2}} \prod_{j,k=1}^{n,n} \exp\left(-|z_{jk}|^2\right) \\ &= \tfrac{1}{\pi^{n^2}} \exp\left(-\sum_{j,k=1}^{n,n} |z_{jk}|^2\right) \\ &= \tfrac{1}{\pi^{n^2}} \exp\left(-\mathrm{Tr} Z^* Z\right) \end{aligned}$$

We can write $d\mu_{G_2}(Z)$ or $P(Z)dZ$ to refer to the measure of sets in the ensemble depending on the context.

Lemma 5. *The measure of the complex Ginibre ensemble is invariant under unitary transformations.*

Proof. Let $Z \in G_2(n,n)$ and $V \in U(n)$. Since $V^*V = I$,

$$\begin{aligned} P(VZ) &= \tfrac{1}{\pi^{n^2}} \exp\left(-\mathrm{Tr}(Z^*V^*VZ)\right) \\ &= \tfrac{1}{\pi^{n^2}} \exp\left(-\mathrm{Tr}(Z^*Z)\right) \\ &= P(Z) \end{aligned}$$

Now consider the map

$$Z \mapsto VZ$$

and let $W = VZ$. The Jacobian of the transformation is again equal to one, being the determinant of an $n^2 \times n^2$ unitary matrix. Therefore the joint probability density function of the w_{ij} is the same as that of the z_{ij}. Thus the probability density function and the measure are left-invariant under unitary transformations. Right-invariance is proven in a similar fashion. □

We now describe the method. Let Z be a random matrix from the complex $n \times n$ Ginibre ensemble. Next, we orthonormalize Z using the modified Gram-Schmidt process as described above such that the resulting Q us uniformly distributed on $U(n)$.

The Matlab code implementing the method is as follows (here, `n` refers to the unitary group $U(n)$ and `m` is the number of samples):

```
function gramschmidt(n,m)
  for p=1:m
    X=1/sqrt(2)*randn(n,n);
    Y=1/sqrt(2)*randn(n,n);
    Z=X+i*Y;
    for j=1:n
       for k=1:(j-1)
         pr=pr+dot(U(:,k),Z(:,j))/dot(U(:,k),U(:,k))*U(:,k);
       end
       U(:,j)=Z(:,j)-pr;
       E(:,j)=1/norm(U(:,j))*U(:,j);
    end
    Q(:,:,p)=E;
  end
```

The matrix Q contains the random samples.

3.1.2 The Householder Reflection Method

The Householder method works similarly for unitary matrices [16]. There is a slight change in definition however.

Definition 7. *The* **Householder matrix** H_{z} *associated with the complex vector* $\mathbf{z}$ *is defined as follows. Let* $r = -\mathrm{e}^{\mathrm{i}\ \arg(z_1)}||\mathbf{z}||$ *and* $\mathbf{u} = \mathbf{z} - r\mathbf{e}_1$. *Normalize* $\mathbf{u}$ *by setting* $\mathbf{v} = \frac{\mathbf{u}}{||\mathbf{u}||}$ *and set* $H_{\mathrm{z}} = I - 2\mathbf{v}\mathbf{v}^*$.

The complex method involves a new definition for r and the conjugate transpose. Given a complex matrix Z, this method decomposes Z into a unitary matrix Q and an upper-right-triangular matrix R.

It is again true that the QR decomposition of a complex matrix is not unique. If $Z \in GL(n, \mathbb{C})$ with decomposition $Z = QR$ and

$$\Lambda = \begin{bmatrix} e^{i\theta_1} & & & 0 \\ & e^{i\theta_2} & & \\ & & \ddots & \\ 0 & & & e^{i\theta_n} \end{bmatrix} = diag(e^{i\theta_1}, \ldots, e^{i\theta_n})$$

then $Q' = Q\Lambda$ and $R' = \Lambda^* R$ are still unitary and upper-right-triangular and

$$Z = QR = Q'R'$$

We again obtain a multi-valued map

$$QR_{\mathbb{C}} : GL(n, \mathbb{C}) \to U(n) \times R(n)$$

which we will make single-valued and one-to-one by specifying the exact decomposition algorithm. Again, we obtain the following Lemma:

Lemma 6. *If $Z = QR = Q'R'$, then $Q' = Q\Lambda$ and $R' = \Lambda^* R$ where $\Lambda \in \Lambda(n)$, the group of all unitary diagonal matrices.*

Consider the following map

$$QR^{\mathbb{R}+} : GL(n, \mathbb{C}) \to U(n) \times R(n)/\Lambda(n)$$

We want to define $QR^{\mathbb{R}+}$ in such a way that

$$Z \mapsto (Q, \lambda T) \quad \Rightarrow \quad UZ \mapsto (UQ, \lambda T) \quad (*)$$

for any $U \in U(n)$. This implies that multiplying Z by a unitary matrix pushes forward to a left self-action of $U(n)$, after decomposition.

For the unitary case, the construction of $QR^{\mathbb{R}+}$ involves choosing a class of representatives from $R(n)/\Lambda(n)$ by imposing that the main diagonal of R have only real positive entries. (We could also correct QR by randomly perturbing the phases via `Q=Q*diag(exp(i*2*pi*rand(n,1))`.) Using Lemma

6, we can uniquely factorize any $Z \in GL(n, \mathbb{C})$ such that the main diagonal of R has this property. If $Z = QR$, then $UZ = UQR$, $U \in U(n)$. This decomposition of UZ is unique within the specified class of representatives of $R(n)/\Lambda(n)$. So, our map $QR^{\mathbb{R}+}$ satisfies $(*)$. Recall that Lemma 5 states that $d\mu_{\mathbb{C}G}(UZ) = d\mu_{\mathbb{C}G}(Z)$ for any $U \in U(n)$. Thus, the induced measure on the left factor $U(n)$ of $QR^{\mathbb{R}+}$ is also invariant under left multiplication, and hence must be Haar.

Theorem 6. *If the map $QR^{\mathbb{R}+}$ is such that*

$$Z \mapsto (Q, \lambda T) \quad \Rightarrow \quad UZ \mapsto (UQ, \lambda T)$$

then $QR^{\mathbb{R}+}$ decomposes the measure of the complex Ginibre ensemble $d\mu_{\mathbb{C}G}$ as

$$d\mu_{\mathbb{C}G}(Z) = d\mu_{Haar}(Q) \times d\mu_{R(n)/\Lambda(n)}(\lambda T)$$

The Matlab code implementing the method is as follows (here, `n` refers to the unitary group $U(n)$ and `m` is the number of samples):

```
function householder(n,m)
    for p=1:m
        Z(:,:,p)=1/sqrt(2)*randn(n,n)+1/sqrt(2)*i*randn(n,n);
        [Q(:,:,p),R(:,:,p)]=qr(Z(:,:,p));
        J=eye(n);
        for j=1:n
            J(j,j)=R(j,j,p)/norm(R(j,j,p));
        end
        Q(:,:,p)=Q(:,:,p)*J;
    end
```

The matrix `Q` contains the random samples.

3.1.3 The Subgroup Method

We again apply the subgroup method as introduced in [9], this time to the unitary group. $U(n)$ has the following nested chain of subgroups

$$U(n) \supset U(n-1) \supset \cdots \supset U(2)$$

with $U(k-1)$ the subgroup of $U(k)$ fixing $e_1 \in U(k)$. So, a matrix $U_{k-1} \in U(k-1)$ can be written

$$U_k = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & U_{k-1} & \\ 0 & & & \end{bmatrix} \quad \text{with } U_{k-1}^* U_{k-1} = I_{k-1,k-1}$$

Again, considering the shorter chain $U(n-1) \subset U(n)$, we can choose coset representatives of $U(n-1)$ in $U(n)$ by specifying where e_1 goes. This is equivalent to choosing an element at random from

$$\mathcal{S}_n = \{z \in \mathbb{C}^n : zz^* = 1\}$$

For any $z \in \mathcal{S}_n$ with $z_1 = |z_1|e^{i\theta}$, $0 \leq \theta < 2\pi$, the map

$$\phi(z) = -e^{i\theta}(I - 2u^T\bar{u})$$

where

$$u = v/\sqrt{vv^*}, \quad v = z + e^{i\theta}e_1$$

is unitary and sends e_1 to z. We can now mimic Lemma 3 with

Lemma 7. *Let z be a random point in $\mathcal{S}_n$ and let U_{n-1} be a random matrix in $U(n-1)$. Then*

$$-e^{i\theta}(I - 2u^T\bar{u}) \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & U_{n-1} & \\ 0 & & & \end{bmatrix}$$

is uniformly distributed on $U(n)$ where u is defined as above.

As with the orthogonal group, this becomes Householder's method proceeding inductively on n.

Another method to generate samples from $U(n)$ would be to consider the short tower $U(n) \supset O(n)$. If a method to obtain random coset representatives of the factor group was known, this element could be multiplied by a random element of $O(n)$, producing a Haar-distributed element in $U(n)$.

3.2 Uniformity Tests

3.2.1 Distribution of Eigenvalues

The joint probability density function of the eigenvalues of a Haar-distributed random matrix in $U(n)$ is the Weyl denominator formula [7]

$$f_2(\theta_1, \ldots, \theta_n) = \frac{1}{(2\pi)^n n!} \prod_{j<k} \left| e^{i\theta_j} - e^{i\theta_k} \right|^2$$

As with the orthogonal group, not much can be inferred from this formula regarding the eigenvalue distribution, aside from noticing that the eigenvalues will tend to repel each other. That is, as θ_j and θ_k approach each other, f_2 goes to zero. We should expect then, that since the eigenvalues of a unitary matrix lie on the unit circle, the eigenvalues of a "typical" element of $U(n)$ for n large, should be quite regularly-spaced. This provides us with a quick visual test to determine whether a given matrix is random. Figure 3.1 show eigenvalue plots of a random element generated from $U(100)$ using the Gram-Schmidt method and the Householder method.

3.2.2 Distribution of Traces

Diaconis and Mallows [7] proved the following:

Theorem 7. *If M is chosen uniformly from $U(n)$ and B is any ball in $\mathbb{C}$,*

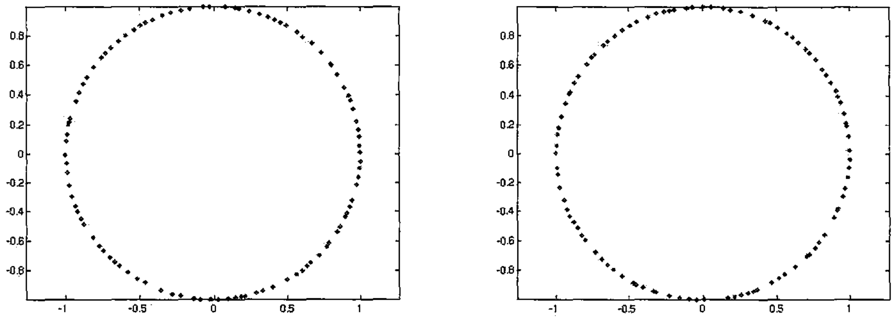

Figure 3.1: Eigenvalues from $U(100)$ using G-S and Householder

then, as n tends to ∞,

$$\left|P(\mathrm{Tr}(M) \in B) - \int_B \frac{1}{\pi} e^{-|z|^2} dz\right| \to 0$$

uniformly in Borel sets B.

It has been shown that for given n, the error term is super-exponential [13]. That is

$$\left|P(\mathrm{Tr}(M) \in B) - \int_B \frac{1}{\pi} e^{-|z|^2} dz\right| \leq \frac{c}{n^{\sigma n}}$$

for universal constants $c, \sigma > 0$. This is far from the typical order of n for the error term of n random points on the unit circle using the classical central limit theorem.

This first result allows us to test the uniformity of our data as we did with the orthogonal group. Figure 3.2 shows histograms of traces of 1000 random unitary matrices generated from $U(100)$ using the above methods.

3.2.3 Eigenvalue Phase Histograms

In this section, we compare histograms of eigenvalue phases for the above methods. To create the histograms (see Figure 3.3, we sampled 1000 matrices from $U(50)$. As expected, given the regular spacing of eigenvalues on the unit circle, we see a uniform distribution of the phases.

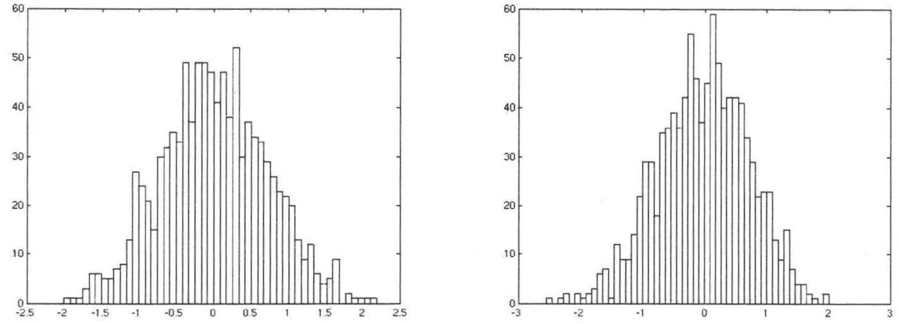

Figure 3.2: Trace plots using G-S, and Householder

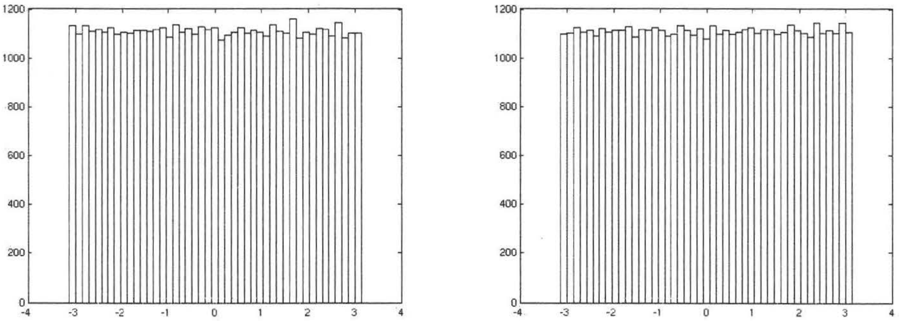

Figure 3.3: Phase histograms for G-S, Householder

3.3 Speeds of Algorithms

$U(n)$	$U(2)$	$U(3)$	$U(5)$	$U(10)$	$U(20)$	$U(50)$	$U(100)$
Method	**Average time to sample 1000 points**						
Gram-Schmidt	.15s	.29s	.85s	4.3s	18.6s	2m 4s	8m 33s
Householder	.19s	.28s	1.0s	5.8s	26.3s	2m 49s	11m 34s

3.4 High Powers of $U(n)$

Rains in [20] has studied the distribution of the eigenvalues and trace of powers of a random matrix $U \in U(n)$, and has noted an interesting shift from regularity to randomness. In fact, it appears that for powers n and higher, the eigenvalues of U^n are distributed as n points uniformly distributed on the unit circle. We asked the question earlier: what kind of matrix results in eigenvalues which resemble random points on a circle. One answer is high powers of random unitary matrices. Taking high powers of unitary matrices seems to produce matrices which are quite far from "typical" Haar-distributed unitary matrices.

Figure 3.4 shows an eigenvalue plot for a random matrix from $O(100)$, for the 100-th power of a random matrix from $O(100)$, and 100 uniformly distributed points on the circle.

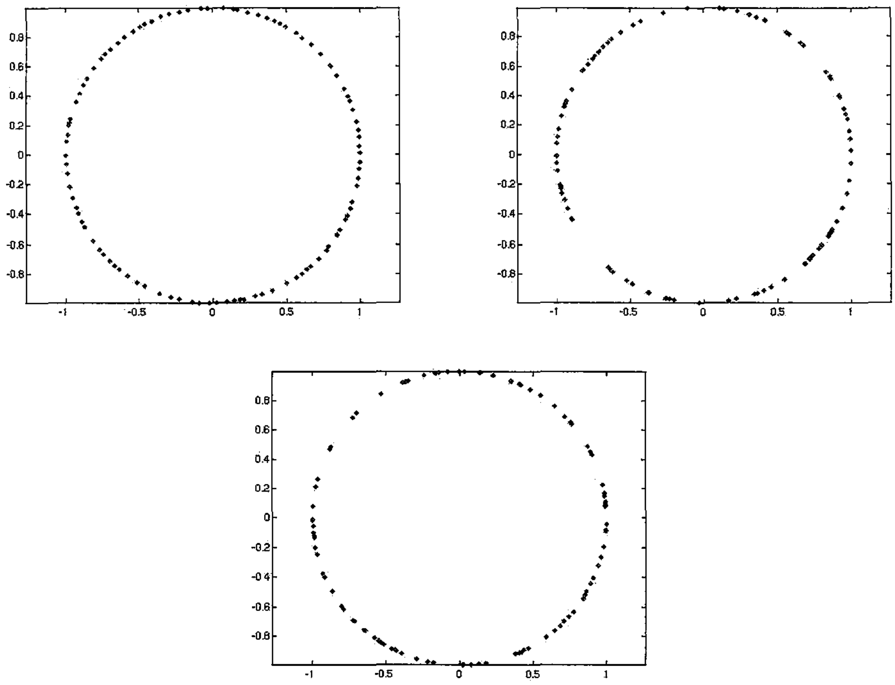

Figure 3.4: Eigenvalue plots for $O \in O(100)$, $O^n \in O(100)$, and 100 random points on $\mathbb{S}^1$

3.5 Other Matrix Ensembles

Following Edelman and Rao [11] we introduce some further matrix ensembles. For completeness we also include their joint eigenvalue probability density functions. Note that the following ensembles are non-compact ensembles, and thus cannot be assigned an invariant measure as with the compact Lie groups.

As we have seen, the real Ginibre ensemble $G_1(n,n)$ is an $n \times n$ matrix of independent and identically distributed standard real random normals. More generally, we write $G_\beta(n,n)$ where $\beta = 2$ refers to complex normals and $\beta = 4$ refers to quaternion normals. Matlab can generate an element from $G_4(n,n)$ via

```
X=1/sqrt(4)*randn(n,n)+1/sqrt(4)*i*randn(n,n);
Y=1/sqrt(4)*randn(n,n)+1/sqrt(4)*i*randn(n,n);
G=[X Y; -conj(Y) conj(X)]
```

The following classical random matrix ensembles are derived from the Gaussian random matrices, and hence are invariant to orthogonal transformations:

Gaussian orthogonal ensemble (GOE): This ensemble is composed of symmetric $n \times n$ matrices $(A + A^T)/2$ where $A \in G_1(n,n)$. The diagonal elements are i.i.d. $N(0,1)$ and the remaining elements are i.i.d. $N(0,1/2)$ subject to the symmetry.

Gaussian unitary ensemble (GUE): This ensemble is composed of Hermitian $n \times n$ matrices $(A + A^*)/2$ where $A \in G_2(n,n)$. The diagonal elements are i.i.d. $N(0,1)$ and the remaining elements are i.i.d. $N_2(0,1/2)$ subject to being Hermitian.

Gaussian symplectic ensemble (GSE): This ensemble is composed of self-dual $n \times n$ matrices $(A + A^D)/2$ where $A \in G_4(n,n)$ and D is the dual-transpose of a quaternion matrix. The diagonal elements are i.i.d. $N(0,1)$ and the remaining elements are i.i.d. $N_4(0,1/2)$ subject to being self-dual.

The Gaussian ensembles have the following joint eigenvalue probability density functions:

$$f_\beta(\lambda) = c_H^\beta \prod_{i<j} |\lambda_i - \lambda_j|^\beta \exp\left(-\sum_{i=1}^{n} \lambda_i^2/2\right)$$

where $\beta = 1$ refers to the GOE, $\beta = 2$ refers to the GUE, and $\beta = 4$ refers to the GSE, and where

$$c_H^\beta = (2\pi)^{-n/2} \prod_{j=1}^{n} \frac{\Gamma(1+\frac{\beta}{2})}{\Gamma(1+\frac{\beta}{2}j)}$$

The following ensembles are not part of the classical ensembles but are important in their own right:

Wishart ensemble ($W_\beta(m,n), m \geq n$): symmetric/Hermitian/self-dual $n \times n$ matrix $A'A$, where $A \in G_\beta(m,n)$ and A' denotes A^T, A^* or A^D, as A is real, complex, or quaternionic, respectively.

The Wishart ensemble is named after John Wishart who studied these matrices as sample covariance matrices, as they relate to statistics applications. This ensemble has joint eigenvalue probability density function

$$g_\beta(\lambda) = c_L^{\beta,a} \prod_{i<j} |\lambda_i - \lambda_j|^\beta \prod_{i} \lambda_i^{a-p} \exp\left(-\sum_{i=1}^{n} \lambda_i/2\right)$$

where $a = \frac{\beta}{2}m$ and $p = 1 + \frac{\beta}{2}(n-1)$, and β is as above, and where

$$c_L^{\beta,a} = 2^{-na} \prod_{j=1}^{n} \frac{\Gamma(1+\frac{\beta}{2})}{\Gamma(1+\frac{\beta}{2}j)\Gamma(a-\frac{\beta}{2}(n-j))}$$

MANOVA ensemble ($J_\beta(m_1, m_2, n), m_1, m_2 \geq n$): symmetric/Hermitian/self-dual $n \times n$ matrix which can be obtained as $A/(A+B)$, where A and B are $W_\beta(m_1, n)$ and $W_\beta(m_2, n)$, respectively.

The MANOVA ensembles arise in statistics in the multivariate analysis of variance, and are also known as the Jacobi ensembles. Their joint eigenvalue probability density function is

$$h_\beta(\lambda) = c_J^{\beta,a_1,a_2} \prod_{i<j} |\lambda_i - \lambda_j|^\beta \prod_{j=1}^{n} \lambda_i^{a_1-p}(1-\lambda_i)^{a_2-p}$$

where $a_1 = \frac{\beta}{2}m_1$, $a_2 = \frac{\beta}{2}m_2$, and $p = 1 + \frac{\beta}{2}(n-1)$, and $\beta = 1, 2$, and where

$$c_J^{\beta,a_1,a_2} = \prod_{j=1}^{n} \frac{\Gamma(1+\frac{\beta}{2})\Gamma(a_1+a_2-\frac{\beta}{2}(n-j))}{\Gamma(1+\frac{\beta}{2}j)\Gamma(a_1-\frac{\beta}{2}(n-j))\Gamma(a_2-\frac{\beta}{2}(n-j))}$$

Chapter 4

Spheres and Hyperspheres

The uniform sampling from spheres is required in numerous applications (molecular simulations, optics modeling, earthquake source modeling, etc.) In optics, an integrating sphere reflectometer is a component with small input and output holes, and whose interior surface is engineered to create a high proportion of diffuse reflections (ie. scattering) relative to specular reflections (ie. mirror-type). To model a ray of light travelling within an integrating sphere, a sequence of diffuse reflection points is required. Due to the physical properties of an integrating sphere reflectometer, it is sufficient to sample uniform points from the surface of a sphere [19].

In this section we present a number of ways to sample random points from spheres. The relative speeds of the methods are analyzed, as well as the uniformity of the generated points.

4.1 Sampling Methods

4.1.1 The Rejection Method

This method involves generating uniform points inside an $(n+1)$-cube (C^{n+1}) of side length 2 centred at the origin, discarding points lying outside of the n-sphere, and projecting the remaining points onto its surface. Let $x_1, x_2, \ldots, x_{n+1}$ be independent uniform random variates on $(-1, 1)$ and set $\mathbf{x} = (x_1, x_2, \ldots, x_{n+1})$. Reject samples if $||\mathbf{x}|| > 1$. Project $\mathbf{x}$ onto $\mathbb{S}^n$ to the point

$$\mathbf{y} = \left(\frac{x_1}{||\mathbf{x}||}, \frac{x_2}{||\mathbf{x}||}, \cdots, \frac{x_{n+1}}{||\mathbf{x}||}\right)$$

which will be uniformly distributed on the sphere. The probability that $\mathbf{x}$ is on or within $\mathbb{S}^n$ is the ratio of the volume of the sphere to that of the cube which is

$$\frac{V(\mathbb{S}^n)}{V(C^{n+1})} = \frac{\left(\frac{\pi^{(n+1)/2}}{\Gamma(\frac{n+1}{2}+1)}\right)}{2^{n+1}}$$

where Γ is the gamma function defined as

$$\begin{aligned} \Gamma(\tfrac{\nu}{2}+1) &= (\tfrac{\nu}{2})! && \text{when } \nu \text{ is even, and} \\ \Gamma(\tfrac{\nu}{2}+1) &= \sqrt{\pi}\frac{\nu!!}{2^{(\nu+1)/2}} && \text{when } \nu \text{ is odd} \end{aligned}$$

Hyperspherical volume attains a maximum of approximately 5.26 when $n = 5$ causing the ratio of volumes to tend to zero very quickly as n grows large. Hence, the probability that points will be discarded is high. For $\mathbb{S}^2$, the volume ratio is $\left(\frac{4}{3}\pi\right)/2^3 = \pi/6$, so an average of $6/\pi$ points are required to generate one point in $\mathbb{S}^2$, or equivalently $18/\pi \approx 5.73$ uniform variates. For $\mathbb{S}^3$, an average number of 13 uniform variates are required to generate a point in its interior, and for $\mathbb{S}^4$, an average of 30 are required. Hence this method should not be used as a general method for sampling from $\mathbb{S}^n$. Figure 4.1 shows 5000 points on $\mathbb{S}^2$ using this method.

The Matlab code implementing the method is as follows (here, `m` is the number of samples and `n` is the dimension of the sphere):

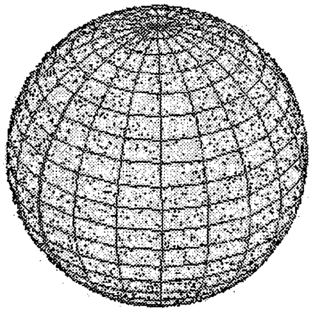

Figure 4.1: 5000 points on $\mathbb{S}^2$ using the rejection method

```
function rejection(n,m)
    j=0;
    while j<m
        x=2*rand(n+1,1)-1;
        l=norm(x);
        if (l<=1)
            j=j+1;
            y(j,:)=1/l*x;
        end
    end
```

The matrix y contains the random samples.

4.1.2 The Quaternionic Method

The quaternionic method is also a rejection method, but rather than simply projecting points to the surface of the sphere, points are transformed by rational formulae obtained using quaternions [2]. We first look at $\mathbb{S}^2$.

Generate x_1, x_2, x_3, x_4 uniformly from $[-1, 1]$ until

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 < 1$$

and consider the quaternion

$$\mathbf{q} = x_1 + x_2\mathbf{i} + x_3\mathbf{j} + x_4\mathbf{k}$$

This $\mathbf{q}$ is uniformly distributed on the set $|\mathbf{q}| < 1$. Since a non-zero quaternion effects a rotation in $\mathbb{R}^3$ via conjugation, we can rotate the north pole $\mathbf{k}$ by $\mathbf{qkq}^{-1}$. (Note that every rotation can be represented by a quaternion. The map $z = \cos(\alpha/2)+\sin(\alpha/2)\mathbf{v}$ is a counter-clockwise rotation through an angle α about the axis defined by the unit vector $\mathbf{v}$.)

The claim is that the distribution of $\mathbf{qkq}^{-1}$ on $\mathbb{S}^2$ is uniform. The uniform distribution is the unique rotation-invariant distribution on the surface of the sphere. Let $\mathbf{p}$ be a non-zero quaternion and form:

$$\begin{aligned}\mathbf{p}(\mathbf{qkq}^{-1})\mathbf{p}^{-1} &= \tfrac{|\mathbf{p}|}{|\mathbf{p}|}\mathbf{p}(\mathbf{qkq}^{-1})\mathbf{p}^{-1} \\ &= \left(\tfrac{\mathbf{p}}{|\mathbf{p}|}\mathbf{q}\right)\mathbf{k}\left(\tfrac{\mathbf{p}}{|\mathbf{p}|}\mathbf{q}\right)^{-1}\end{aligned}$$

Multiplying $\mathbf{q}$ by a quaternion of unit norm is a rotation of quaternion space and therefore the distribution of $\mathbf{q}$ is preserved. Hence the distribution of $\mathbf{qkq}^{-1}$ is unchanged by rotations, and is thus the uniform distribution.

So where does this rotation send the north pole? Set

$$\mathbf{qkq}^{-1} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

and equate the coefficients if $\mathbf{i}, \mathbf{j}$, and $\mathbf{k}$. The computations are trivial and we spare the reader the details but one; $\mathbf{q}^{-1} = \mathbf{q}^*/|\mathbf{q}|^2$. The coordinates of the

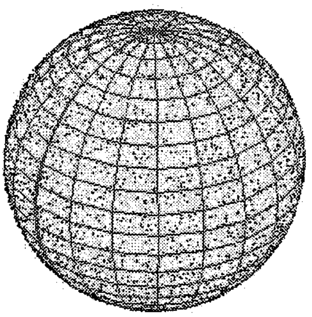

Figure 4.2: 5000 points on $\mathbb{S}^2$ using the quaternionic method

uniformly distributed point on $\mathbb{S}^2$ are

$$\begin{aligned} x &= \frac{2(x_2x_4 + x_1x_3)}{x_1^2 + x_2^2 + x_3^2 + x_4^2} \\ y &= \frac{2(x_3x_4 - x_1x_2)}{x_1^2 + x_2^2 + x_3^2 + x_4^2} \\ z &= \frac{x_1^2 + x_4^2 - x_2^2 - x_3^2}{x_1^2 + x_2^2 + x_3^2 + x_4^2} \end{aligned}$$

Although the method extends to higher dimensions (and to spaces with an action by a compact group of transformations) the formulae are apparently quite complicated and involve Clifford Algebras [3].

This method, although mathematically interesting, is comparatively inefficient. Since we reject points outside the interior of $\mathbb{S}^3$, on average, 30 uniform variates are required to produce an acceptable starting point. Figure 4.2 shows 5000 points on $\mathbb{S}^2$ using this method.

The Matlab code implementing the method is as follows (here, `m` is the number of samples):

```
function quaternionic(m)
    j=0;
    while j<m
        x=2*rand(4,1)-1;
        l=norm(x);
        if (l<1)
            j=j+1;
            y(j,1)=2*(x(2)*x(4)+x(1)*x(3))/l^2;
            y(j,2)=2*(x(3)*x(4)-x(1)*x(2))/l^2;
            y(j,3)=(x(1)^2+x(4)^2-x(2)^2-x(3)^2)/l^2;
        end
    end
```

The matrix y contains the random samples.

4.1.3 The Gaussian Method

This sphere-sampling method utilizes the following Lemma.

Lemma 8. *If* $\mathbf{x} = (x_1, x_2, \ldots, x_{n+1})$ *is such that* $x_i \sim N(0,1)$, *then* $\mathbf{x}/||\mathbf{x}||$ *is uniformly distributed on* $\mathbb{S}^n$.

Proof. If $Q \in O(n+1)$, then $\frac{Q\mathbf{x}}{||\mathbf{x}||}$ has the same distribution as $\frac{Q\mathbf{x}}{||Q\mathbf{x}||}$. Since the x_i are identically distributed normal random variates, $\mathbf{x}$ is radially symmetric and so $\frac{Q\mathbf{x}}{||Q\mathbf{x}||}$ has the same distribution as $\frac{\mathbf{x}}{||\mathbf{x}||}$. Therefore $\frac{\mathbf{x}}{||\mathbf{x}||}$ is also radially symmetric and $\left|\left|\frac{\mathbf{x}}{||\mathbf{x}||}\right|\right| = 1$ with probability 1. Hence $\frac{\mathbf{x}}{||\mathbf{x}||}$ is uniformly distributed on $\mathbb{S}^n$. [5] □

We now use Lemma 8 to generate random points on the n-sphere. Let $\mathbf{x} = (x_1,\ x_2, \ldots,\ x_{n+1})$ where the x_i are i.i.d. standard normal random

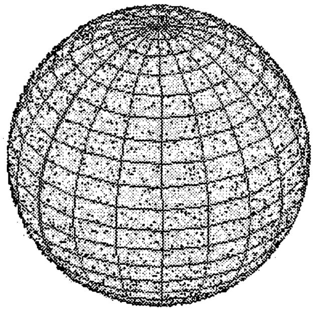

Figure 4.3: 5000 points on $\mathbb{S}^2$ using the Gaussian method

variates. Project $\mathbf{x}$ onto $\mathbb{S}^n$ to the point

$$\mathbf{y} = (y_1, \ldots, y_{n+1}) = \frac{1}{||\mathbf{x}||}(x_1,\ x_2, \ldots,\ x_{n+1})$$

Due to Lemma 8, the y_i are uniformly distributed on the sphere [18]. Figure 4.3 shows 5000 points on $\mathbb{S}^2$ using this method.

The Matlab code implementing the method is as follows (here, `m` is the number of samples and `n` is the dimension of the sphere):

```
function gaussian(n,m)
    x=randn(m,n+1);
    for i=1:m
        y(i,:)=1/norm(x(i,:))*x(i,:);
    end
```

The matrix `y` contains the random samples.

4.1.4 Two Uniform Methods

In his paper on choosing points from spheres, Marsaglia [15] presents methods for $\mathbb{S}^2$ and $\mathbb{S}^3$ which use the uniform distribution on $(-1,1)$. The first method is for $\mathbb{S}^2$. Let x_1, x_2 be independent uniform random variates on $(-1,1)$ and reject samples if $r = x_1^2 + x_2^2 \geq 1$. Form the point

$$\mathbf{x} = \left(2x_1\sqrt{1-r},\ 2x_2\sqrt{1-r},\ 1-2r\right)$$

The probability that x_1, x_2 are accepted is the ratio of the area of the unit circle to that of the square of side length 2, which is $\pi/4$. Hence, on average, $4/\pi$ points are required to generate a point in $\mathbb{S}^2$, or equivalently $8/\pi \approx 2.55$ uniform variates.

To prove that the method works, we follow Marsaglia [15] and combine the following two facts:

A. If (z_1, z_2, z_3) is uniform on the surface of the unit 2-sphere, then each z_i is uniform on $(-1,1)$, (the area of a spherical cap is a multiple of its height), and (z_1, z_2), for given z_3, is uniform on the circumference of the circle of radius $\sqrt{1-z_3^2}$.

B. If (x_1, x_2) is uniform over the interior of the unit circle, then $r = x_1^2 + x_2^2$ is uniform on $(0,1)$ and independent of the point $(x_1/\sqrt{r}, x_2/\sqrt{r})$.

Combining A and B we conclude that if z_3 is uniform on $(-1,1)$ and independent of $(x_1/\sqrt{r}, x_2/\sqrt{r})$ then

$$\left(\frac{x_1}{\sqrt{r}}\sqrt{1-z_3^2}, \frac{x_2}{\sqrt{r}}\sqrt{1-z_3^2}, z_3\right)$$

is uniform on the surface of the 2-sphere. But $1-2r$ is uniform on $(-1,1)$ and independent of $(x_1/\sqrt{r}, x_2/\sqrt{r})$. Substituting $1-2r$ for z_3 then yields the sample point $\mathbf{x}$.

Figure 4.4 shows 5000 points on $\mathbb{S}^2$ using the method for the 2-sphere.

The Matlab code implementing the method is as follows (here, `m` is the number of samples):

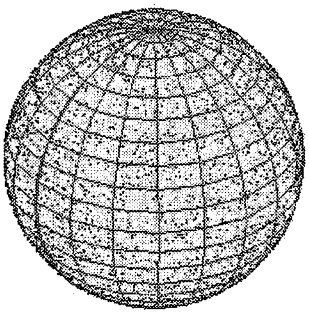

Figure 4.4: 5000 points on $\mathbb{S}^2$ using the Gaussian method

```
function uniform2(m)
    j=0;
    while j<m
        x=2*rand(2,1)-1;
        l=x(1)^2+x(2)^2;
        if (l<1)
            j=j+1;
            f=2*sqrt(1-S);
            y(j,:)=[x(1)*f x(2)*f 1-2*l];
        end
    end
```

The matrix y contains the random samples.

For $\mathbb{S}^3$, let x_1, x_2 be independent uniform random variates on $(-1, 1)$ and reject samples if $r_1 = x_1^2 + x_2^2 \geq 1$. Let x_3, x_4 be independent uniform random variates on $(-1, 1)$ and reject samples if $r_2 = x_3^2 + x_4^2 \geq 1$. Then the point

$$(x_1, x_2, x_3\sqrt{(1-r_1)/r_2}, x_4\sqrt{(1-r_1)/r_2})$$

is uniformly distributed on $\mathbb{S}^3$. The probability that x_1, x_2, x_3, x_4 are all accepted is half that of the method above. That is, approximately 5.09 uniform variates are required to generate one point on $\mathbb{S}^3$. Compared to the general rejection method, this is a more than 2-fold increase in efficiency in this dimension.

The Matlab code implementing the method is as follows (here `m` is the number of samples):

```
function uniform3(m)
    j=0;
    while j<m
        x1=2*rand(2,1)-1;
        l1=x1(1)^2+x1(2)^2;
        if (l1<1)
            x2=2*rand(2,1)-1;
            l2=x2(1)^2+x2(2)^2;
            if (l2<1)
                j=j+1;
                f=sqrt((1-l1)/l2);
                y(j,:)=[x1(1) x1(2) x2(1)*f x2(2)*f];
            end
        end
    end
```

The matrix y contains the random samples.

Marsaglia does not discuss why the method for $\mathbb{S}^3$ works and hints at similar methods for higher-dimensional spheres, but does not elaborate on either.

4.1.5 The Homogeneous Method

We have already sampled random points from a few homogeneous spaces, without exploiting their homogeneity. $\mathbb{S}^n$, $\mathbb{C}P^n$, and the real and complex Grassmannian spaces are all homogeneous spaces. We now define a homogeneous space and then introduce methods to sample uniformly distributed random points from them.

Definition 8. *A* **homogeneous space** *is a topological space X on which there is a transitive group action by a Lie group G. Since the action is transitive, there is only one group orbit, which implies that all of the isotropy groups are conjugate. Therefore, X is isomorphic to G/G_x where G_x is the isotropy group of the point x.*

The Lie group $O(n+1)$ acts on $\mathbb{S}^n$ by rotating or reflecting a point of $\mathbb{S}^n$ about or through an axis. The action is transitive since for any distinct p, $q \in \mathbb{S}^n$ there exists an element of $O(n+1)$ taking p to q. (In fact, there are numerous such elements. For example, the two rotations defined by the great circle joining p and q, or the reflection defined by p, q and one of their midpoints on this same circle.) Therefore, $\mathbb{S}^n$ is homogeneous and the group action possesses only one orbit. Elements of the isotropy group of $(1, 0, \ldots, 0)^T$ are matrices of the form

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & O_n & \\ 0 & & & \end{bmatrix}$$

where $O_n \in O(n)$. The group of all such elements is clearly isomorphic to $O(n)$. In fact, since there is only one group orbit, all of the isotropy groups are isomorphic to $O(n)$, and thus

$$\mathbb{S}^n \sim O(n+1)/O(n)$$

We have considered a number of methods to sample uniformly from orthogonal groups. Choose one of these methods, and generate a matrix O from $O(n+1)$. Since $O(n)$ has measure zero in $O(n+1)$, the probability of generating an element from one of the augmented isotropy groups is almost zero. Apply O to the point $p = (0, \ldots, 0, 1)^T$. Continue sampling new matrices from $O(n+1)$ and applying them to the previously obtained point. Since the action is transitive, we can follow the point through its orbit, resulting in uniformly distributed points on $\mathbb{S}^n$.

The Matlab code implementing the method is as follows (here, `m` is the number of samples and `n` is the dimension of the sphere):

```
function homogeneous(n,m)
    Q=orthogonal(n+1,m);
    x(:,1)=Q(:,:,1)*eye(n+1,1);
    for j=2:m
        x(:,j)=Q(:,:,j)*X(:,i-1);
    end
```

The matrix `x` contains the random samples.

4.2 Uniformity Tests

4.2.1 Distributions of Sample Coordinates

Pick a coordinate x_i from the sample points. It can be shown [17] that its distribution is proportional to $(1-x^2)^{n/2-1}$. Plots of these functions are shown in Figure 4.5 for various n. Figures 4.6 to 4.9 show how the distribution of the first coordinate of the hypersphere matches up to the plotted functions for various n for each of the sphere-sampling methods.

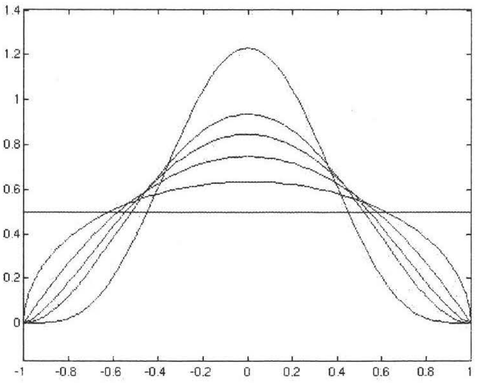

Figure 4.5: Plots of $(1-x^2)^{n/2-1}$ for $n = 2, 3, 4, 5, 6, 10$ ($n = 2$ is the horizontal line and n increases as the midpoint of the graph increases)

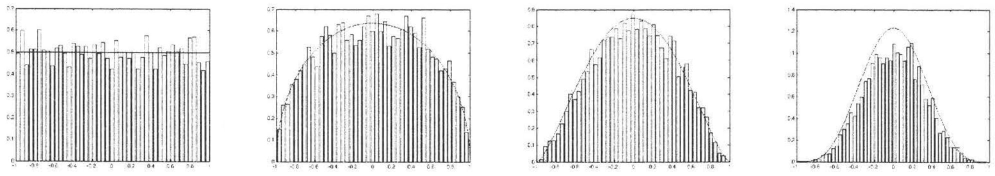

Figure 4.6: Gaussian method with $m = 5000$, $n = 2, 3, 5, 10$

4.2.2 Variances of Sample Coordinates

The above distribution has variance $\frac{1}{n+1}$ as shown in [17]. Therefore we can test our sample data for each method for various n. We do so by taking the average of the sample variances from 5 sample populations of $m = 1000$ points each.

$$\hat{S}^2 = \frac{1}{m-1} \sum_{i=1}^{m} x_i^2$$

Sample variances were not calculated for the rejection method for high n, as even after 5 minutes not one point had been accepted.

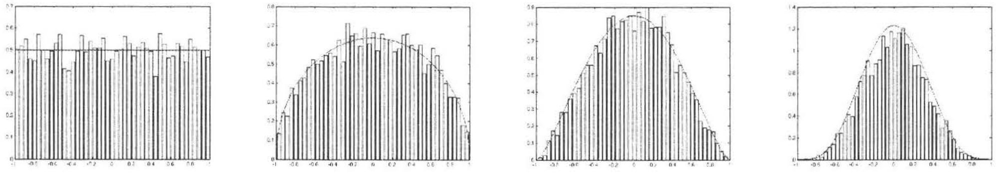

Figure 4.7: Rejection method with $m = 5000$, $n = 2, 3, 5, 10$

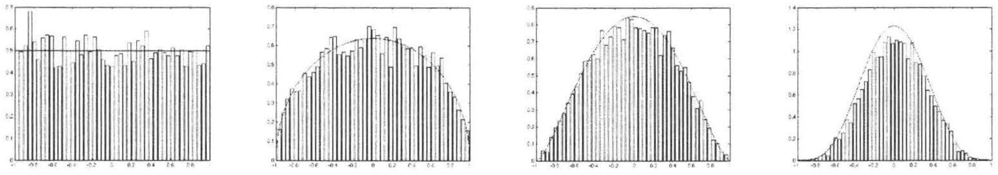

Figure 4.8: Homogeneous method with $m = 5000$, $n = 2, 3, 5, 10$

Sphere	$\mathbb{S}^2$	$\mathbb{S}^3$	$\mathbb{S}^4$	$\mathbb{S}^5$	$\mathbb{S}^{10}$	$\mathbb{S}^{20}$	$\mathbb{S}^{40}$
σ^2	.333	.250	.200	.167	.091	.048	.024

Method	**Sample variances $\hat{S}^2$**						
Rejection	.333	.246	.197	.171	.089	?	?
Quaternionic	.344	N/A	N/A	N/A	N/A	N/A	N/A
Gaussian	.341	.251	.206	.167	.091	.047	.025
Homogeneous	.338	.253	.200	.162	.092	.045	.025
Uniform	.336	.251	N/A	N/A	N/A	N/A	N/A

4.3 Speeds of Algorithms

In this section we look at the running-times of each of the above algorithms. For each method, we sample 1000 points.

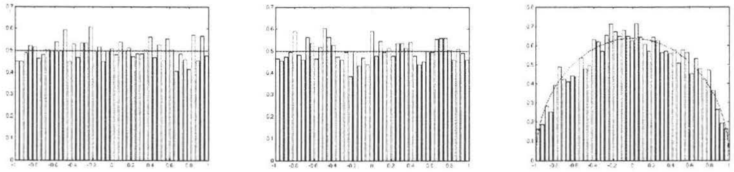

Figure 4.9: Various methods for $m = 5000$ (Quaternionic $n = 2$, Uniform $n = 2, 3$)

Sphere	$\mathbb{S}^2$	$\mathbb{S}^3$	$\mathbb{S}^5$	$\mathbb{S}^{10}$	$\mathbb{S}^{20}$	$\mathbb{S}^{50}$	$\mathbb{S}^{100}$
Method	**Average time to sample 1000 points**						
Rejection	.011s	.017s	.053s	3.58s	$< \infty$	$< \infty$	$< \infty$
Quaternionic	.015s	N/A	N/A	N/A	N/A	N/A	N/A
Gaussian	.0063s	.0076s	.011s	.019s	.080s	.24s	.78s
Uniform	.0077s	.013s	N/A	N/A	N/A	N/A	N/A
Homogeneous	.15s	.25s	.64s	2.66s	10.57s	1m 7.53s	4m 33s

Chapter 5

Real Projective Space

In this section, we look at a method for generating random points from real projective space using the sphere methods described in the previous section.

5.1 Sampling Method

Definition 9. Real projective space *(or $\mathbb{R}P^n$) is the projective space of lines in $\mathbb{R}^{n+1}$. More precisely, it is obtained by forming the quotient of $\mathbb{R}^{n+1} \setminus \{0\}$ under the equivalence relation $x \sim \lambda x$ for all nonzero $\lambda \in \mathbb{R}$.*

5.1.1 The Spherical Method

$\mathbb{R}P^n$ can also be constructed by identifying antipodal points of $\mathbb{S}^n$. Hence, we can use any of our sphere-sampling methods to sample random points from $\mathbb{R}P^n$ by applying an antipodal reflection to points from the lower hemisphere, as well as to points lying on any of the lower-dimensional spheres in standard position within that hemisphere.

The Matlab code implementing this method is as follows (here, `m` is the number of samples, `n` is the dimension of the sphere, and `sphere()` is a sphere-sampling function):

```
function spherical(n,m)
    x=sphere(n,m);
    for j=1:m
        if (x(j,n+1)<0)
            x(j,:)=x(j,:)*(-1);
        end
        for k=1:n
            p=n+2-k;
            if (x(j,p)~=0) break
            elseif (x(j,p-1)<0)
                x(j,:)=x(j,:)*(-1);
            end
        end
    end
```

The matrix x contains the random samples.

5.2 Uniformity Tests

To test the uniformity of points generated in $\mathbb{R}P^n$, we can use the same tests as we used for spheres. We expect the same distributions, aside from those involving the last coordinate, as we only choose points in the upper hemisphere.

5.2.1 Distributions of Sample Coordinates

The plots of the coordinate distributions are identical to those of the sphere. However, we show the distributions of the last coordinate in Figure 5.1, which as expected, show the positive half of the spherical distributions.

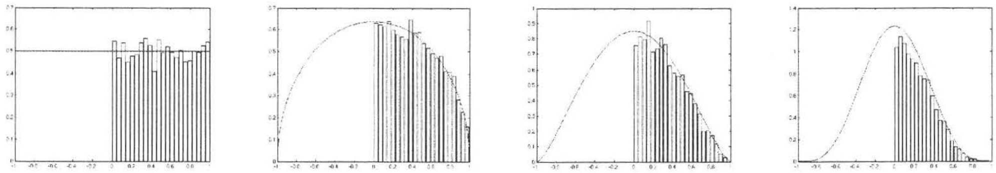

Figure 5.1: Antipodal method with $m = 5000,\ n = 2, 3, 5, 10$

5.2.2 Variances of Sample Coordinates

Here, the theoretical and sample variances are calculated for the first coordinate.

$\mathbb{R}P^n$	2	3	4	5	10	20	40
σ^2	.333	.250	.200	.167	.091	.048	.024
$\hat{S}^2$	.325	.258	.201	.167	.090	.049	.024

5.3 Speed of Algorithm

The following table shows running times for 1000 points sampled from some real projective spaces. The sphere-sampling method used is the Gaussian sphere-sampling method, the fastest of the general methods.

Average time to sample 1000 **points**						
$\mathbb{R}P^2$	$\mathbb{R}P^3$	$\mathbb{R}P^5$	$\mathbb{R}P^{10}$	$\mathbb{R}P^{20}$	$\mathbb{R}P^{50}$	$\mathbb{R}P^{100}$
.0065s	.0081s	.010s	.020s	.058s	.26s	.77s

Chapter 6

Complex Projective Space

In this section, we look at methods for generating random points from complex projective space using the spherical and homogeneous methods.

Definition 10. Complex projective space *(or $\mathbb{C}P^n$) is the projective space of complex lines in $\mathbb{C}^{n+1}$. More precisely, it is obtained by forming the quotient of $\mathbb{C}^{n+1} \setminus \{0\}$ under the equivalence relation $z \sim \lambda z$ for all nonzero $\lambda \in \mathbb{C}$.*

6.1 Sampling Methods

6.1.1 The Spherical Method

We can also think of $\mathbb{C}P^n$ as the quotient of $\mathbb{S}^{2n+1} \subset \mathbb{C}^{n+1}$ by the action of $U(1)$. So generating points in $\mathbb{S}^{2n+1}$ will suffice. Since $U(1)$ is of measure zero in $\mathbb{C}^{n+1}$, the likelihood of generating two points $\mathbf{w_1}, \mathbf{w_2}$ in $\mathbb{C}P^n$ such that $\mathbf{w_1} = \lambda \mathbf{w_2}$ for some $\lambda \in \mathbb{C}$ is almost zero. Once we have our points in $\mathbb{S}^{2n+1}$ of the form

$$(x_1, x_2, \dots, x_{2n+2})$$

we can set $z_k = x_{2k-1} + ix_{2k}$ for $1 \leq k \leq n+1$ to get the following point in $\mathbb{C}^{n+1}$

$$\mathbf{z} = (z_1, z_2, \dots, z_{n+1})$$

The probability that $x_{2n+1} = x_{2n+2} = 0$ is again almost zero, so we can normalize the last coordinate of $\mathbf{z}$ to obtain

$$\hat{\mathbf{z}} = \left(\frac{z_1}{z_{n+1}}, \frac{z_2}{z_{n+1}}, \ldots, \frac{z_n}{z_{n+1}}\right)$$

to express the point in homogeneous coordinates.

The Matlab code implementing this method is as follows (here, `m` is the number of samples, `n` refers to $\mathbb{C}P^n$, and `sphere()` is a sphere-sampling function):

```
function spherical(n,m)
    p=sphere(2*n+1,m);
    x=p(:,1:(n+1));
    y=p(:,(n+2):(2*n+2));
    z=x+i*y;
    z=1./z(:,n+1)*ones(1,n+1).*z;
```

The matrix `z` contains the random samples.

6.1.2 The Homogeneous Method

The Lie group $U(n+1)$ acts transitively on $\mathbb{C}P^n$ and so complex projective space is a homogeneous space. In fact,

$$\mathbb{C}P^n \sim U(n+1)/U(n) \times U(1)$$

Choose one of the methods to sample from the unitary group, and generate a matrix U from $U(n+1)$. Since the dimension of $U(n+1)$ is n^2+2n+1 and that of $U(n) \times U(1)$ is n^2+1, the measure of the latter is zero in the former, and thus the probability of generating an element from an isotropy group is almost zero. Apply U to the point $p = (0, \ldots, 0, 1)^T$. Continue sampling new matrices from $U(n+1)$ and applying them to the previously obtained point. Since the action is transitive, we can follow this point through its orbit, resulting in uniformly distributed points on $\mathbb{C}P^n$.

The Matlab code implementing this method is as follows (here, m is the number of samples and n refers to $\mathbb{C}P^n$):

```
function homogeneous(n,m)
    q=unitary(n+1,m);
    z(:,1,:)=q(:,:,1)*eye(n+1,1);
    for j=2:m
        z(:,j,:)=q(:,:,j)*z(:,j-1,:);
    end
```

The matrix z contains the random samples.

6.2 Uniformity Tests

To test the uniformity of points generated in $\mathbb{C}P^n$, we can use the same tests as we used for spheres.

6.2.1 Distributions of Sample Coordinates

Here we plot the distributions of the real part of the first coordinate $Re(z_1)$ along with the associated theoretical distribution functions for $n = 2, 3, 4, 5$.

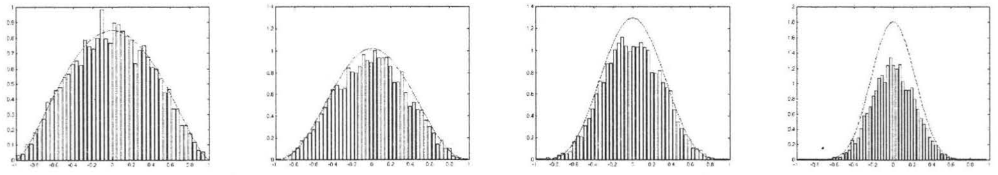

Figure 6.1: Spherical method with $m = 5000$, $n = 2, 3, 5, 10$

6.2.2 Variances of Sample Coordinates

Here, the theoretical and sample variances are calculated for the real part of the first coordinate $Re(z_1)$.

$\mathbb{C}P^n$	2	3	4	5	10	20	40
σ^2	.167	.125	.100	.083	.046	.024	.012
Method	**Sample variances $\hat{S}^2$**						
Spherical	.165	.124	.101	.082	.045	.024	.013
Homogeneous	.161	.126	.098	.085	.046	.023	.013

6.3 Speeds of Algorithms

The following table shows running times for 1000 points sampled from some complex projective spaces. The sphere-sampling method used is the Gaussian sphere-sampling method, the fastest of the general methods.

Average time to sample 1000 points							
Method	$\mathbb{C}P^2$	$\mathbb{C}P^3$	$\mathbb{C}P^5$	$\mathbb{C}P^{10}$	$\mathbb{C}P^{20}$	$\mathbb{C}P^{50}$	$\mathbb{C}P^{100}$
Spherical	.010s	.014s	.021s	.056s	.19s	.73s	1.88s
Homogeneous	.31s	.50s	1.26s	5.31s	20.59s	2m 8.7s	8m 24.7s

Chapter 7

Real Grassmannian Spaces

The grand tour is a method of viewing multivariate statistical data (say, n-dimensional data) on a computer screen by projecting this data orthogonally onto a sequence of two-dimensional subspaces. To implement the grand tour, a sequence of pairs of orthonormal vectors spanning a plane in $\mathbb{R}^n$ are required. It is desirable that the sequence of planes be uniformly distributed. This is an example of sampling from the Grassmannian $Gr(n, 2, \mathbb{R})$.

Definition 11. *The* **real Grassmannian space** $Gr(n, k, \mathbb{R})$ *is the space of all k-dimensional subspaces of* $\mathbb{R}^n$ $(0 < k < n)$.

Grassmannians are generalizations of projective spaces (in fact, $\mathbb{R}P^n$ is equivalent to $Gr(n+1, 1, \mathbb{R})$).

7.1 Sampling Methods

7.1.1 The Spherical Method

We can easily sample points from $Gr(n, k, \mathbb{R})$ by using our $\mathbb{R}P^{n-1}$-sampling method. Generate k points in $\mathbb{R}P^{n-1}$. Since the probability of generating two identical points is almost zero, the k points will span a k-dimensional subspace of $\mathbb{R}^n$. If need be, the vectors can be orthogonalized.

The Matlab code implementing this method is as follows (here, `m` is the number of samples, `n` is the dimension of the ambient space, `k` is the dimension of the subspaces, and `realproj()` is the $\mathbb{R}P^{n-1}$-sampling function):

```
function projective(n,k,m)
    for j=1:m
        x(:,:,j)=realproj(n-1,k);
    end
```

The matrix `x` contains the random samples.

7.1.2 The Homogeneous Method

The Lie group $O(n)$ acts transitively on $Gr(n,k)$ and so real Grassmannians are homogeneous spaces. In fact,

$$Gr(n,k) \sim O(n)/O(n-k) \times O(k)$$

Choose one of the methods to sample from the orthogonal group, and generate a matrix O from $O(n)$. Since the dimension of $O(n)$ is $n(n-1)/2$ and that of $O(n-k) \times O(k)$ is smaller, the measure of the latter in the former is zero, and thus the probability of generating an element from an isotropy group is almost zero. Apply O to the $n \times k$ identity matrix. Continue sampling new matrices from $O(n)$ and applying them to the previously obtained point. Since the action is transitive, we can follow this point through its orbit, resulting in uniformly distributed points on $Gr(n,k)$.

The Matlab code implementing this method is as follows (here, `m` is the number of samples, `n` is the dimension of the ambient space, `k` is the dimension of the subspaces, and `orthogonal()` is an $O(n)$-sampling function):

```
function grassmannian(n,k,m)
    Q=orthogonal(n,m);
    x(:,:,1)=Q(:,:,1)*eye(n,k);
    for j=2:m
        x(:,:,j)=Q(:,:,j)*X(:,:,i-1);
    end
```

The matrix **x** contains the random samples.

7.2 Note on the Stiefel manifold $V_k(\mathbb{R}^n)$

If the results of the previous sampling methods are orthogonal sets of vectors, then they can be interpreted as random k-frames from the Stiefel manifold $V_k(\mathbb{R}^n)$. Real Stiefel manifolds are homogeneous spaces of the orthogonal group:

$$V_k(\mathbb{R}^n) \sim O(n)/O(n-k)$$

Since $O(n-k)$ has measure zero in $O(n)$, the probability of generating a matrix which fixes a particular k-frame is zero. So even though the Stiefel manifold $V_k(\mathbb{R}^n)$ and the Grassmannian $Gr(n,k,\mathbb{R})$ are of different dimensions, these sampling methods will generate random points from both spaces.

7.3 Uniformity Tests

To test the uniformity of points generated in $Gr(n,k,\mathbb{R})$, we project the first k standard basis vectors of $\mathbb{R}^n$ onto an orthogonalized set of the k vectors (g_{ij}) generated by the algorithm.

$$< e_1, g_{i1} > g_{i1} + \cdots + < e_k, g_{ik} > g_{ik}$$

We then look at the distribution of elements in the resulting $n \times k$ matrix. We plot the distributions of the $(1,1),(2,2)$ elements and of the $(1,2),(2,1)$ elements in Figures 7.1 and 7.2 for $Gr(6,2,\mathbb{R})$ using the two methods. We see similar distributions for the two methods. Although not definitive evidence of the uniformity of the samples, it suggests that the samples have the same

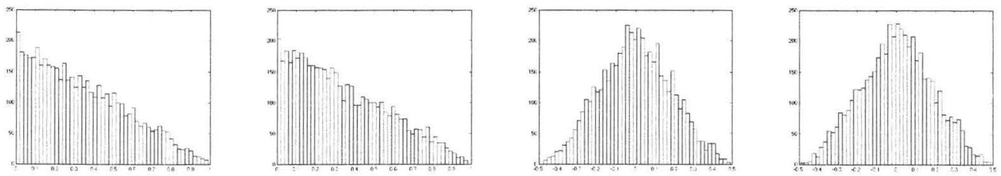

Figure 7.1: Histograms for $Gr(6, 2, \mathbb{R})$ (projective method)

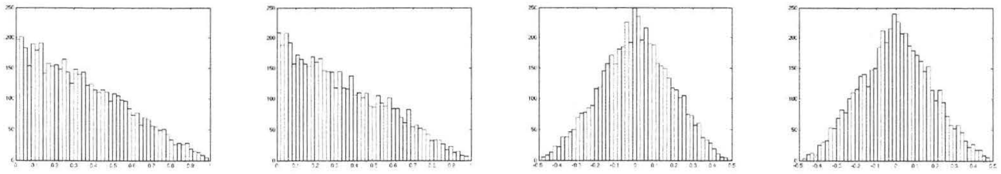

Figure 7.2: Histograms for $Gr(6, 2, \mathbb{R})$ (homogeneous method)

distribution. The homogeneous method is much too slow to be useful for this space, but if it indeed generates uniform samples, then it serves to demonstrate the uniformity of the other method.

7.4 Speeds of Algorithms

The following table shows running times for 1000 points sampled from some real Grassmannian spaces.

$Gr(n, k, \mathbb{R})$	(3,1)	(5,2)	(10,4)	(20,8)	(50,20)	(100,40)
Method	**Average time to sample 1000 points**					
Spherical	.056s	.082s	.19s	1.27s	14.18s	59.38s
Homogeneous	.34s	.40s	2.41s	14.0s	1m52.3s	6m8.1s

Chapter 8

Complex Grassmannian Spaces

In the theory of wireless communications, with multiple inputs and multiple outputs (MIMO), packings of real and complex Grassmannian spaces are often required. There are a number of methods used to generate these packings. One method uses the homogeneous method to sample complex k-planes from $\mathbb{C}^n$ as an initial guess [23].

Definition 12. *The* **complex Grassmannian space** $Gr(n, k, \mathbb{C})$ *is the space of all k-dimensional complex subspaces of* $\mathbb{C}^n$ $(0 < k < n)$.

8.1 Sampling Methods

8.1.1 The Spherical Method

We can easily sample points from $Gr(n, k, \mathbb{C})$ by using our $\mathbb{C}P^n$-sampling technique. Generate k points from $\mathbb{C}P^{n-1}$. Since the probability of generating two points z_1, z_2 such that $z_1 = \lambda z_2$ $(\lambda \in \mathbb{C})$ is almost zero, the k points will span a k-dimensional subspace in $\mathbb{C}^n$.

The Matlab code implementing this method is as follows (here, `m` is the number of samples, `n` is the dimension of the ambient space, `k` is the dimension of the subspaces, and `compproj()` is one of the $\mathbb{C}P^n$-sampling functions):

```
function projective(n,k,m)
    for j=1:m
        x(:,:,j)=compproj(n-1,k);
    end
```

The matrix **x** contains the random samples.

8.1.2 The Homogeneous Method

The Lie group $U(n)$ acts transitively on $Gr(n,k,\mathbb{C})$ and so complex Grassmannians are homogeneous spaces. In fact,

$$Gr(n,k,\mathbb{C}) \sim U(n)/U(n-k) \times U(k)$$

Choose one of the methods to sample from the unitary group, and generate a matrix U from $U(n)$. Since the dimension of $U(n)$ is n^2 and that of $U(n-k) \times U(k)$ is smaller, the measure of the latter in the former is zero, and thus the probability of generating an element from an isotropy group is almost zero. Apply U to the $n \times k$ identity matrix. Continue sampling new matrices from $U(n)$ and applying them to the previously obtained point. Since the action is transitive, we can follow this point through its orbit, resulting in uniformly distributed points on $Gr(n,k,\mathbb{C})$.

The Matlab code implementing this method is as follows (here, `m` is the number of samples, `n` is the dimension of the ambient space, `k` is the dimension of the subspaces, and `unitary()` is a $U(n)$-sampling function):

```
function cgrassmannian(n,k,m)
    Q=unitary(n,m);
    x(:,:,1)=Q(:,:,1)*eye(n,k);
    for j=2:m
        x(:,:,j)=Q(:,:,j)*X(:,:,i-1);
    end
```

The matrix **x** contains the random samples.

8.2 Note on the Stiefel manifold $V_k(\mathbb{C}^n)$

If the results of the previous sampling methods are orthogonal sets of vectors, then they can be interpreted as random k-frames from the Stiefel manifold $V_k(\mathbb{C}^n)$. Complex Stiefel manifolds are homogeneous spaces of the unitary group:

$$V_k(\mathbb{C}^n) \sim U(n)/U(n-k)$$

Since $U(n-k)$ has measure zero in $U(n)$, the probability of generating a matrix which fixes a particular complex k-frame is almost zero. So even though the Stiefel manifold $V_k(\mathbb{C}^n)$ and the complex Grassmannian $Gr(n, k, \mathbb{C})$ are of different dimensions, these sampling methods will generate random points in both spaces.

8.3 Speeds of Algorithms

The following table shows running times for 1000 points sampled from some complex Grassmannian spaces. The `unitary()` function used for the homogeneous test is the faster Gram-Schmidt method.

$Gr(n,k,\mathbb{C})$	(3,1)	(5,2)	(10,4)	(20,8)	(50,20)	(100,40)
Method	**Average time to sample 1000 points**					
Spherical	.047s	.14s	.53s	3.30s	22.4s	1m30s
Homogeneous	.31s	1.13s	6.24s	29.5s	2m56s	11m41s

Chapter 9

Conclusion

This study surveyed a number of the current methods used to sample random points from various compact spaces. The Gram-Schmidt and Householder reflection methods were very similar in their speeds and uniformity. However, the numerical instability of even the modified Gram-Schmidt method, suggests that the latter method is in fact a better choice. For the remainder of the spaces considered (the symmetric spaces: spheres, projective spaces and Grassmannians,) the Gaussian method is by far the best option. As the dimension of the space increases, it remains very fast compared to the other methods. As some of these spaces are homogeneous spaces of the matrix groups, a homogeneous method of generating random points was looked at. Although extremely inefficient, it produced uniform samples, and may possibly be better applied to other homogeneous spaces which do not have a faster method available, such as the Gaussian method.

Appendix A

Supplementary Lemmas and Theorems

Lemma 9. *A Householder matrix (i.e. $H = I - 2vv^T$) is a symmetric matrix.*

Proof.

$$\begin{aligned} H^T &= (I - 2vv^T)^T \\ &= I - 2(vv^T)^T \\ &= I - 2vv^T \quad \text{since } vv^T \text{ is symmetric} \\ &= H \end{aligned}$$

□

Lemma 10. *A Householder matrix (i.e. $H = I - 2vv^T$) is an orthogonal matrix.*

Proof.

$$\begin{aligned}
HH^T &= (I - 2vv^T)(I - 2vv^T)^T \\
&= (I - 2vv^T)(I - 2(vv^T)^T) \\
&= I - 2vv^T - 2(vv^T)^T + 4vv^T(vv^T)^T \\
&= I - 4vv^T + 4v(v^Tv)v^T \quad \text{since } vv^T \text{ is symmetric} \\
&= I - 4vv^T + 4vv^T \quad \text{since } v^Tv = ||v||^2 = 1 \text{ (by definition)} \\
&= I
\end{aligned}$$

□

Lemma 11. *If* $\mathbf{x} \in \mathbb{R}^n$ *then* $H_{\mathbf{x}}\mathbf{x}$ *(where* $H_{\mathbf{x}}$ *is the Householder matrix associated with* $\mathbf{x}$*) has the same direction as* $\mathbf{e}_1$*.*

Proof. We will write $H_{\mathbf{x}}$ in its non-normalized format

$$H_{\mathbf{x}} = I - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}$$

and determine $\mathbf{v}$ such that $H_{\mathbf{x}}\mathbf{x} \in span\{\mathbf{e}_1\}$. Assuming the latter, we write

$$H_{\mathbf{x}}\mathbf{x} = \left(I - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right)\mathbf{x} = \mathbf{x} - 2\frac{\mathbf{v}(\mathbf{v}^T\mathbf{x})}{\mathbf{v}^T\mathbf{v}} = \mathbf{x} - 2\frac{\mathbf{v}^T\mathbf{x}}{\mathbf{v}^T\mathbf{v}}\mathbf{v}$$

Looking at the far left and right sides of this equation, we see that $\mathbf{v} \in span\{\mathbf{x}, \mathbf{e}_1\}$. Let $\mathbf{v} = \mathbf{x} + \rho\mathbf{e}_1$ so that

$$\mathbf{v}^T\mathbf{x} = \mathbf{x}^T\mathbf{x} + \rho\mathbf{e}_1{}^T\mathbf{x} = \mathbf{x}^T\mathbf{x} + \rho\eta_1$$

and

$$\mathbf{v}^T\mathbf{v} = (\mathbf{x}^T + \rho\mathbf{e}_1{}^T)(\mathbf{x} + \rho\mathbf{e}_1) = \mathbf{x}^T\mathbf{x} + 2\rho\eta_1 + \rho^2$$

Now rewrite $H_{\mathbf{x}}\mathbf{x}$ as

$$\begin{aligned}
H_{\mathbf{x}}\mathbf{x} &= \mathbf{x} - 2\frac{\mathbf{v}^T\mathbf{x}}{\mathbf{v}^T\mathbf{v}}\mathbf{v} = \mathbf{x} - 2\frac{\mathbf{x}^T\mathbf{x} + \rho\eta_1}{\mathbf{x}^T\mathbf{x} + 2\rho\eta_1 + \rho^2}(\mathbf{x} + \rho\mathbf{e}_1) \\
&= \left(1 - 2\frac{\mathbf{x}^T\mathbf{x} + \rho\eta_1}{\mathbf{x}^T\mathbf{x} + 2\rho\eta_1 + \rho^2}\right)\mathbf{x} - 2\rho\frac{\mathbf{v}^T\mathbf{x}}{\mathbf{v}^T\mathbf{v}}\mathbf{e}_1
\end{aligned}$$

To have $H_{\mathbf{x}}\mathbf{x}$ point in the direction of $\mathbf{e}_1$ we want the first term to vanish. So let

$$1 - 2\frac{\mathbf{x}^T\mathbf{x} + \rho\eta_1}{\mathbf{x}^T\mathbf{x} + 2\rho\eta_1 + \rho^2} = 0$$

$$\mathbf{x}^T\mathbf{x} + 2\rho\eta_1 + \rho^2 - 2\mathbf{x}^T\mathbf{x} - 2\rho\eta_1 = 0$$

$$\mathbf{x}^T\mathbf{x} = \rho^2 \Leftrightarrow ||\mathbf{x}|| = \pm\rho$$

Therefore $\rho = \pm||\mathbf{x}||$ and so $\mathbf{v} = \mathbf{x} \pm ||\mathbf{x}||\mathbf{e}_1||$. Finally

$$\begin{aligned} H_{\mathbf{x}}\mathbf{x} &= -2\rho\frac{\mathbf{v}^T\mathbf{x}}{\mathbf{v}^T\mathbf{v}}\mathbf{e}_1 = -2\rho\frac{\mathbf{x}^T\mathbf{x} \pm \rho\eta_1}{\mathbf{x}^T\mathbf{x} \pm 2\rho\eta_1 + \mathbf{x}^T\mathbf{x}} \\ &= \rho\mathbf{e}_1 = \mp||\mathbf{x}||\mathbf{e}_1 \end{aligned}$$

□

Theorem 8. *(Cartan-Dieudonné Theorem) Let (V, b) be an n-dimensional, non-degenerate symmetric bilinear space over a field with characteristic not equal to 2. Then, every element of the orthogonal group $O(V, b)$ is a composition of at most n reflections.*

Definition 13. *The k^{th}* **moment** *of a probability measure μ (defined on the Borel sets of $\mathbb{R}$) is defined as*

$$\mu(x^k) := \int x^k \mu(dx)$$

Theorem 9. *Let μ_n be a sequence of probability measures with moments of all orders. Suppose that for each k, $\mu_n(x^k)$ converges to a number μ_k. Then, there is a measure μ with $\mu(x^k) = \mu_k$. If μ is determinate, i.e., is uniquely determined by its moments, then for every bounded continuous function*

$$\int f d\mu_n \longrightarrow \int f d\mu$$

(where the convergence is weak star). [6]

Bibliography

[1] E. Borel. Sur les principes de la theorie cinetique des gaz. *Annales, L'Ecole Normale Sup.*, 23:9–32, 1906.

[2] J.M. Cook. Technical notes and short papers: Rational formulae for the production of a spherically symmetric probability distribution. *Mathematical Tables and Other Aids to Computation*, 11(58):81–82, 1957.

[3] J.M. Cook. Remarks on a recent paper. *Communications of the Association for Computing Machinery*, 2:26, 1959.

[4] M. Coram and P. Diaconis. New tests of the correspondence between unitary eigenvalues and the zeros of Riemann's Zeta function. *Journal of Physics A: Mathematical and Theoretical*, 36(12):2883–2906, 2002.

[5] L. Devroye. *Non-Uniform Random Variate Generation.* Springer-Verlag, New York, 1986.

[6] P. Diaconis. Applications of the method of moments in probability and statistics. *Moments in Mathematics: Proceedings of Symposia in Applied Mathematics, Amer. Math. Soc.*, 37:125–1422, 1987.

[7] P. Diaconis. Patterns in eigenvalues: The 70th Josiah Willard Gibbs lecture. *Bulletin of the Amer. Math. Soc.*, 40(2):155–178, 2003.

[8] P. Diaconis. What is... a random matrix? *Notices of the Amer. Math. Soc.*, 52(11):1348–1349, 2005.

[9] P. Diaconis and M. Shahshahani. The subgroup algorithm for generating uniform random variables. *Probability in the Engineering and Informational Sciences*, 1:15–32, 1987.

[10] M.L. Eaton. *Multivariate Statistics: A Vector Space Approach.* Wiley, New York, 1983.

[11] A. Edelman and R. Rao. Random matrix theory. *Acta Numerica*, 14(1):233–239, 2005.

[12] J. Ginibre. Statistical ensembles of complex, quaternion, and real matrices. *Journal of Mathematical Physics*, 6(3):440–449, 1965.

[13] K. Johansson. On random matrices from the compact classical groups. *Annals of Mathematics*, 145:519–545, 1997.

[14] J. Keating and N. Snaith. Random matrix theory and $\zeta(\frac{1}{2}+it)$. *Commun. Math. Phys.*, 214:57–89, 2000.

[15] G. Marsaglia. Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics*, 43(2):645–646, 1972.

[16] F. Mezzadri. How to generate random matrices from the classical compact groups. *ArXiv Mathematical Physics e-prints,* `arXiv:math-ph/0609050`, 2006.

[17] M. Min-Oo and J.A. Toth. The Levy concentration phenomenon for special functions on rank-one symmetric spaces. *Methods and Applications of Analysis*, 7(1):151–164, 2000.

[18] M.E. Muller. A note on a method for generating points uniformly on n-dimensional spheres. *Comm. Assoc. Comp. Mach.*, 2:19–20, 1959.

[19] A. Prokhorov, S. Mekhontsev, and L. Hanssen. Monte Carlo modeling of an integrating sphere reflectometer. *Applied Optics*, 42(19):3835, 2003.

[20] E. Rains. High powers of random elements of compact Lie groups. *Probab. Theory Related Fields*, 107:219–241, 1997.

[21] N.J.A. Sloane. Encrypting by random rotations. *Cryptography: Lecture Notes in Computer Science, Springer-Verlag*, 149:71–128, 1983.

[22] G.W. Stewart. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM J. Numer. Anal.*, 17(3):403–409, 1980.

[23] J.A. Tropp, I.S. Dhillon, R.W. Heath Jr., and T. Strohmer. Constructing packings in Grassmannian manifolds via alternating projection. *Submitted Nov. 2006 to Experimental Mathematics.*

[24] H. Weyl. *The Classical Groups.* Princeton University Press, 1946.