How to generate a random unitary matrix

Maris Ozols

March 16, 2009

Contents

1 Introduction

The purpose of this essay is to explain how to sample uniformly from U(n), the set of all $n \times n$ unitary matrices, and why the method described actually works. However, along the way we will learn how to sample form other interesting sets too. This essay was inspired by [1]. A two-page introduction to the subject can be found in [2].

1.1 Definitions

We will write M^T for *transpose* and $M^\dagger$ for *conjugate-transpose* of a matrix M, and I for the *identity matrix*. We will use M^* to denote the entry-wise complex conjugate of M. Here is some common notation that we will use:

- $\mathrm{O}(n) := \{\, O \in \mathbb{R}^{n\times n} \mid O^\mathsf{T}O = I \,\}$ – the *orthogonal group*,
- $\mathrm{SO}(n) := \{\, O \in \mathrm{O}(n) \mid \det O = 1 \,\}$ – the *special orthogonal group*,
- $\mathrm{U}(n) := \{\, U \in \mathbb{C}^{n\times n} \mid U^\dagger U = I \,\}$ – the *unitary group*,
- $\mathrm{SU}(n) := \{\, U \in \mathrm{U}(n) \mid \det U = 1 \,\}$ – the *special unitary group*,
- $\mathrm{GL}(n,\mathbb{C}) := \{\, M \in \mathbb{C}^{n\times n} \mid \det M \neq 0 \,\}$ – the *general linear group* over $\mathbb{C}$,
- $\mathbb{S}^{n-1} := \{\, \boldsymbol{x} \in \mathbb{R}^n \mid x_1^2 + x_2^2 + \cdots + x_n^2 = 1 \,\}$ – the *unit sphere* in $\mathbb{R}^n$ whose surface has dimension $n-1$.

Note that the columns of an orthogonal matrix form an *orthonormal basis* of $\mathbb{R}^n$. Similarly, the columns of a unitary matrix form an orthonormal basis of $\mathbb{C}^n$ (the *inner product* of column vectors $u, v \in \mathbb{C}^n$ is $u^\dagger v \in \mathbb{C}$). Of course, the same holds for rows. In this sense unitary matrix is a natural generalization of an orthogonal matrix. In fact, quantum physicists would say that unitary matrices are "more natural" than orthogonal ones.

1.2 Quantum physics from A to Z[1]

This section is both – an introduction to quantum mechanics and a motivation for studying random unitary matrices.

Quantum mechanics is about solving the *Scrödinger equation*

$$i\frac{\mathrm{d}\psi(t)}{\mathrm{d}t} = H\psi(t). \tag{1}$$

Roughly speaking (1) says that the rate of change of the state ψ at time t is "proportional" to its current value $\psi(t)$ with the "proportionality coefficient" being the *Hamiltonian* H. However, in reality $\psi(t) \in \mathbb{C}^n$ and $H \in \mathbb{C}^{n\times n}$ is a *Hermitian* matrix, i.e., $H^\dagger = H$.

If you forget this all for a minute and imagine that $n = 1$, then you can easily solve (1) and get

$$\psi(t) = e^{-iHt}\psi(0). \tag{2}$$

In fact, the same solution works in general (with the exponential of a matrix properly defined). Now note that $(e^{-iHt})^\dagger = e^{iHt}$, since H is Hermitian. Therefore e^{-iHt} is unitary for all $t \in \mathbb{R}$. Thus physicists care about unitary transformations, since they describe the evolution of a quantum system. When physicists don't understand what is going on, they use *random unitary matrices*.

[1] This title was inspired by an actual paper [3], whose one of many "contributions" to the field is the insight that "Anton (Zeilinger) is a click in an Anton counter".

1.3 Haar measure

Let f be a function defined on $\mathbb{R}$ (note that $(\mathbb{R}, +)$ is a group). Then for any $a \in \mathbb{R}$ we have:

$$\int_{\mathbb{R}} f(x)\,\mathrm{d}x = \int_{\mathbb{R}} f(x+a)\,\mathrm{d}x. \tag{3}$$

We can have a similar translational invariance on many other groups. If we make a good choice of *measure* μ on our group G, then for all $g \in G$:

$$\int_G f(x)\,\mathrm{d}\mu(x) = \int_G f(gx)\,\mathrm{d}\mu(x). \tag{4}$$

A non-zero measure $\mu : G \to [0, \infty]$ such that for all $S \subseteq G$ and $g \in G$:

$$\mu(gS) = \mu(Sg) = \mu(S), \tag{5}$$

where

$$\mu(S) := \int_{g \in S} \mathrm{d}\mu(g), \tag{6}$$

is called *Haar measure*. It exists on every compact topological group (in particular, on unitary and orthogonal group) and is essentially unique [4].

If, in addition, $\mu(G) = 1$, then μ is called *probability measure* on G. Indeed, if f is a probability density function on G and $\mathrm{d}\mu(g) := f(g)\,\mathrm{d}g$, then

$$\mu(S) = \int_{g \in S} \mathrm{d}\mu(g) = \int_{g \in S} f(g)\,\mathrm{d}g \tag{7}$$

is the total probability of S.

2 Sampling uniformly from various sets

In this section we will discuss some simple examples of how to sample uniformly from various sets. This is a warm-up for our main goal – sampling from $\mathrm{U}(n)$.

2.1 Sampling from $\mathbb{S}^1$ and $\mathrm{SO}(2)$

The obvious parameterizations of $\mathbb{S}^1$ and $\mathrm{SO}(2)$ are

$$\mathbb{S}^1 = \left\{ \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \;\middle|\; 0 \le \alpha < 2\pi \right\} \tag{8}$$

and

$$\mathrm{SO}(2) = \left\{ \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \;\middle|\; 0 \le \alpha < 2\pi \right\}. \tag{9}$$

Clearly, we will get a uniform distribution over $\mathbb{S}^1$ and $\mathrm{SO}(2)$ if we choose $\alpha \in [0, 2\pi]$ uniformly at random. The distribution over $\mathrm{SO}(2)$ will clearly have the invariance property (5).

2.2 Sampling from $\mathbb{S}^2$

There are several ways how to sample from $\mathbb{S}^2$ uniformly [5]. We will discuss two of them.

2.2.1 Using spherical coordinates

One can parameterize $(x, y, z) \in \mathbb{S}^2$ using spherical coordinates as follows:

$$x = \sin\theta \cos\varphi, \tag{10}$$
$$y = \sin\theta \sin\varphi, \tag{11}$$
$$z = \cos\theta, \tag{12}$$

where $0 \leq \theta \leq \pi$ and $0 \leq \varphi < 2\pi$. It is important to note that one will *not* obtain a uniform distribution over $\mathbb{S}^2$ by choosing θ and φ uniformly at random. Instead, the points will be "bunched" close to poles [5]. This is because the area element of $\mathbb{S}^2$ is given by

$$\mathrm{d}S = \sin\theta \,\mathrm{d}\theta \,\mathrm{d}\varphi = -\,\mathrm{d}(\cos\theta)\,\mathrm{d}\varphi. \tag{13}$$

Hence, to obtain a uniform distribution over $\mathbb{S}^2$, one has to pick $\varphi \in [0, 2\pi]$ and $t \in [-1, 1]$ uniformly at random and compute θ as follows:

$$\theta = \arccos t. \tag{14}$$

In this way $\cos\theta = t$ will be uniformly distributed in $[-1, 1]$.

2.2.2 Using normal distribution

Another simple way of sampling uniformly from $\mathbb{S}^2$ is to choose each component of $\boldsymbol{r} \in \mathbb{R}^3$ independently from the *standard normal distribution*, i.e., the *normal distribution* with mean 0 and variance 1. Its probability density function is

$$\varphi(x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}. \tag{15}$$

Thus the joint probability density function of $\boldsymbol{r} = (x, y, z)$ is

$$f(\boldsymbol{r}) = \left(\frac{1}{\sqrt{2\pi}}\right)^3 \exp\left(-\frac{x^2 + y^2 + z^2}{2}\right) = \left(\frac{1}{\sqrt{2\pi}}\right)^3 \exp\left(-\frac{\|\boldsymbol{r}\|^2}{2}\right). \tag{16}$$

Note that $f(\boldsymbol{r})$ does not depend on the direction of $\boldsymbol{r}$, but only on its length $\|\boldsymbol{r}\|$. Thus $\frac{\boldsymbol{r}}{\|\boldsymbol{r}\|}$ is uniformly distributed over $\mathbb{S}^2$. This method generalizes in an obvious way to $\mathbb{S}^n$.

2.3 Sampling from $\mathrm{U}(2)$

Now let us come back to the original question that we are trying to answer, i.e., how to sample from $\mathrm{U}(n)$. Let us first consider the case of $\mathrm{U}(2)$.

Let us use a similar approach as for $\mathbb{S}^1$ and $\mathbb{S}^2$ and parametrize $\mathrm{U}(2)$. Then it remains to choose the right distribution for each parameter. Observe that

$$\mathrm{SU}(2) = \left\{ \begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix} \in \mathbb{C}^{2\times 2} \;\middle|\; |a|^2 + |b|^2 = 1 \right\}. \tag{17}$$

Let us set

$$a := e^{i\psi}\cos\phi \quad \text{and} \quad b := e^{i\chi}\sin\phi, \tag{18}$$

and introduce a *global phase* $e^{i\alpha}$. Then we can parametrize $\mathrm{U}(2)$ as follows [6]:

$$U(\alpha,\phi,\psi,\chi) := e^{i\alpha} \begin{pmatrix} e^{i\psi}\cos\phi & e^{i\chi}\sin\phi \\ -e^{-i\chi}\sin\phi & e^{-i\psi}\cos\phi \end{pmatrix}, \tag{19}$$

where $0 \le \phi \le \frac{\pi}{2}$ and $0 \le \alpha, \psi, \chi < 2\pi$. In this case it is not obvious at all what probability distribution to use for each of the parameters to obtain a uniform distribution over $\mathrm{U}(2)$. As before, knowing the expression for the volume element is helpful [6, 7]:

$$\mathrm{d}V = \frac{1}{2}\sin(2\phi)\,\mathrm{d}\alpha\,\mathrm{d}\phi\,\mathrm{d}\psi\,\mathrm{d}\chi = \frac{1}{2}\,\mathrm{d}(\sin^2\phi)\,\mathrm{d}\alpha\,\mathrm{d}\psi\,\mathrm{d}\chi. \tag{20}$$

Now it becomes evident that one has to pick $\alpha, \psi, \chi \in [0, 2\pi]$ and $\xi \in [0, 1]$ uniformly at random and compute

$$\phi = \arcsin\sqrt{\xi}. \tag{21}$$

It turns out that one can generalize this idea to sample uniformly from $\mathrm{U}(n)$ (see [6]). The main idea is to decompose an $n \times n$ unitary as a product of two-level unitaries $G(i,j) \in \mathrm{U}(n)$ known as *Givens rotations* [8, 12]. Each Givens rotation $G(i,j)$ is just the $n\times n$ identity matrix with elements at positions $\left(\begin{smallmatrix} ii & ij \\ ji & jj \end{smallmatrix}\right)$ replaced by an arbitrary 2×2 unitary matrix.

3 Sampling uniformly from $\mathrm{U}(n)$

3.1 *Mathematica* code

Here is a "quick-and-dirty" way to produce a uniformly distributed unitary matrix using *Mathematica*[2]:

```
RR:=RandomReal[NormalDistribution[0,1]];
RC:=RR+I*RR;
RG[n_]:=Table[RC,{n},{n}];
RU[n_]:=Orthogonalize[RG[n]];
```

[2] This code works in *Mathematica 6.0* or any newer version, but *does not* work in older versions. That is because the generation of random numbers and orthogonalization in previous versions were done in a slightly different way.

This code is self-explanatory, except that `Orthogonalize` actually returns an ortho*normal* basis. To use it for producing a random 4×4 unitary, one calls `RU[4]`. It can be easily modified to sample from the orthogonal group:

```
RR:=RandomReal[NormalDistribution[0,1]];
RG[n_]:=Table[RR,{n},{n}];
RO[n_]:=Orthogonalize[RG[n]];
```

This is a quick but "dirty" implementation, since the resulting distribution over $\mathrm{U}(n)$ and $\mathrm{O}(n)$ depends on the orthogonalization method used by the *Mathematica*'s function `Orthogonalize`. By default *Mathematica* uses the Gram-Schmidt method [9], which happens to give the correct distribution [1, 2]. However, the built-in Householder method does *not* produce the correct distribution [1]. In the next few sections we will try to explain why the above code actually works.

3.2 Ginibre ensemble

The *Ginibre ensemble* consists of matrices $Z \in \mathbb{C}^{n\times n}$, whose elements z_{jk} are independent identically distributed standard normal complex random variables. Such a matrix is generated by function `RG` in the above code.

The probability density function of z_{jk} is

$$f(z_{jk}) = \frac{e^{-|z_{jk}|^2}}{\pi}. \tag{22}$$

Thus the joint probability density function of Z is

$$f_G(Z) = \frac{1}{\pi^{n^2}} \exp\left(-\sum_{j,k=1}^{n} |z_{jk}|^2\right) = \frac{1}{\pi^{n^2}} \exp\left(-\operatorname{Tr}(Z^\dagger Z)\right). \tag{23}$$

Let us use the probability density function f_G to define a measure

$$\mathrm{d}\mu_G(Z) := f_G(Z)\,\mathrm{d}Z, \tag{24}$$

with $\mathrm{d}Z$ defined as follows:

$$\mathrm{d}Z = \prod_{j,k=1}^{n} \mathrm{d}x_{jk}\,\mathrm{d}y_{jk}, \quad \text{where} \quad x_{jk} + iy_{jk} = z_{jk}. \tag{25}$$

Lemma 1. The measure $\mathrm{d}\mu_G$ is invariant under $\mathrm{U}(n)$, i.e., for all $U \in \mathrm{U}(n)$ we have: $\mathrm{d}\mu_G(UZ) = \mathrm{d}\mu_G(Z) = \mathrm{d}\mu_G(ZU)$.

Proof. We will prove the left invariance (the proof of the right invariance is similar). We have to show that $f_G(UZ) = f_G(Z)$ and the Jacobian of the map $Z \mapsto UZ$ (seen as a transformation in $\mathbb{C}^{n^2}$) is one. From equation (23) we have:

$$f_G(UZ) = \frac{1}{\pi^{n^2}} \exp\Big(-\operatorname{Tr}\big((UZ)^\dagger UZ\big)\Big) \tag{26}$$

$$= \frac{1}{\pi^{n^2}} \exp\Big(-\operatorname{Tr}\big(Z^\dagger U^\dagger UZ\big)\Big) = f_G(Z). \tag{27}$$

Note that in UZ the matrix U acts on each column of Z independently. If we think of Z as a vector in $\mathbb{C}^{n^2}$, we can decompose the action of U as an n-fold direct sum $U' := \underbrace{U \oplus \cdots \oplus U}_{n} \in \mathrm{U}(n^2)$. Clearly, $|\det U'| = 1$. □

Now we know that the measure $\mathrm{d}\mu_G$ is invariant under $\mathrm{U}(n)$. It remains to understand why we will still obtain an invariant measure after we orthogonalize the matrix Z from the Ginibre ensemble (see the code in Sect. 3.1).

3.3 Group products

Before we proceed, let us recall some facts from the group theory. Let G be a group with two subgroups H_1 and H_2. Assume we have the following situation:

$$G = H_1 H_2 \quad \text{and} \quad H_1 \cap H_2 = \{\, e \,\}. \tag{28}$$

Then we would like to say that G is a "product" of H_1 and H_2. It turns out that there are three different names used for such product, depending on whether H_1 and H_2 are *normal* subgroups of G:

- $G = H_1 \times H_2$ – *direct product*, when $H_1 \lhd G$ and $H_2 \lhd G$,
- $G = H_1 \rtimes H_2$ – *semidirect product*, when $H_1 \lhd G$,
- $G = H_1 \bowtie H_2$ – *Zappa-Szép product*[3] (or *knit product*).

The reader might be familiar with the first two kids of products. However, for our discussion the third of them will be the most relevant. In particular, we will need the following decomposition[10]:

$$\mathrm{GL}(n, \mathbb{C}) = \mathrm{U}(n) \bowtie \mathrm{T}(n), \tag{29}$$

where $\mathrm{T}(n)$ is the set of invertible $n \times n$ upper-triangular complex matrices with positive diagonal entries. As a reality check, note that both $\mathrm{U}(n)$ and $\mathrm{T}(n)$ are indeed subgroups of $\mathrm{GL}(n, \mathbb{C})$. Also note that we really have to use "$\bowtie$", since none of $\mathrm{U}(n)$ and $\mathrm{T}(n)$ is a normal subgroup of $\mathrm{GL}(n, \mathbb{C})$. We will discuss why condition (28) is satisfied in the next section.

To understand why the decomposition (29) is important to us, we need the following observation:

Lemma 2. If Z is a random matrix from the Ginibre ensemble, then we can assume that Z is invertible. In other words, $Z \in \mathrm{GL}(n, \mathbb{C})$ with probability 1.

Proof. Assume we pick the first $n-1$ vectors (columns of Z) from $\mathbb{C}^n$ at random. They span a linear subspace which is not the entire $\mathbb{C}^n$. Clearly, the probability that the last vector will lie in the same subspace is zero (every proper subspace of $\mathbb{C}^n$ has measure zero in the whole $\mathbb{C}^n$). Thus singular matrices form a measure zero subset of $\mathbb{C}^{n \times n}$. Hence Z is invertible. □

[3] I don't know if there is any standard notation for this, so I made it up and use "$\bowtie$".

Now we can think of the Ginibre ensemble as a probability distribution over $\mathrm{GL}(n,\mathbb{C})$. Then decomposition (29) states that every matrix Z from the Ginibre ensemble can be written as $Z = UT$ for some unique $U \in \mathrm{U}(n)$ and $T \in \mathrm{T}(n)$. To see why equation (29) holds, we will consider the QR decomposition of Z.

3.4 QR decomposition

Any matrix $Z \in \mathrm{GL}(n,\mathbb{C})$ can be decomposed as

$$Z = QR, \tag{30}$$

where $Q \in \mathrm{U}(n)$ and R is upper-triangular and invertible. Expression (30) is called *QR decomposition* of Z. QR decomposition is actually the same as orthogonalization, just from a different point of view. Since R is invertible, we can rewrite (30) as

$$ZR^{-1} = Q, \tag{31}$$

which simply says that the columns of Z can be made orthonormal by multiplying it with an upper-triangular matrix on the right. This should not be too surprising if one knows the *Gram-Schmidt process* for making an orthonormal basis [11]. Let us recall how it works.

Let us denote the ith column of Z by $\boldsymbol{v}_i \in \mathbb{C}^n$ and apply the Gram-Schmidt process to vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$. In the first iteration we normalize $\boldsymbol{v}_1$. In the second iteration we subtract from $\boldsymbol{v}_2$ the component that is along the direction of $\boldsymbol{v}_1$ and normalize the result. Similary, in the kth iteration we take a linear combination of vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ that gives a unit vector orthogonal to the subspace spanned by $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k-1}\}$. Since the kth iteration involves only vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$, the Gram-Schmidt process can be realized by multiplying with an upper-triangular matrix on the right:

$$\left(\begin{array}{c|c|c|c|c} & & & & \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 & \ldots & \boldsymbol{v}_n \\ & & & & \end{array}\right) \begin{pmatrix} * & * & * & \ldots & * \\ & * & * & \ldots & * \\ & & * & \cdots & * \\ & & & \ddots & \vdots \\ & & & & * \end{pmatrix}. \tag{32}$$

Thus for every $Z \in \mathrm{GL}(n,\mathbb{C})$ there exists R that is upper-triangular and invertible, such that equation (31) holds. Therefore QR decomposition (30) exists for every $Z \in \mathrm{GL}(n,\mathbb{C})$.

However, note that QR decomposition (30) it is not unique, since for any diagonal matrix $\Lambda \in \mathrm{U}(n)$, we have

$$QR = (Q\Lambda)(\Lambda^* R) = Q'R', \tag{33}$$

where $Q' = Q\Lambda$ is unitary and $R' = \Lambda^* R$ is upper-triangular. Thus $Z = Q'R'$ is also a valid QR decomposition of Z. However, we can make it unique by demanding that $R \in \mathrm{T}(n)$, i.e., R has positive diagonal entries. Then the only

Λ that can be used in (33) is the identity matrix I, since $\mathrm{U}(n) \cap \mathrm{T}(n) = \{ I \}$. In this way we obtain the decomposition of $\mathrm{GL}(n, \mathbb{C})$ as a Zappa-Szép product of $\mathrm{U}(n)$ and $\mathrm{T}(n)$ as in equation (29).

There are several practical considerations that one must take into account, when implementing the decomposition (29). One possibility is to use the built-in QR decomposition routine that is available in most contemporary programming languages. However, then one must do some post-processing to make sure that $R \in \mathrm{T}(n)$ and hence the decomposition is unique. In particular, given a pair (Q, R) that is returned by a standard QR decomposition routine, one has to compute a diagonal unitary matrix

$$\Lambda = \begin{pmatrix} \frac{r_{11}}{|r_{11}|} & & & \\ & \frac{r_{22}}{|r_{22}|} & & \\ & & \ddots & \\ & & & \frac{r_{nn}}{|r_{nn}|} \end{pmatrix}, \tag{34}$$

where r_{ii} are matrix elements of R and replace (Q, R) by

$$(Q', R') := (Q\Lambda, \Lambda^* R) \tag{35}$$

so that $R' \in \mathrm{T}(n)$ (see [1] for more details). This can be achieved by the following modification of the function `RU` defined in Sect. 3.1:

```
RU[n_]:=Module[{Q,R,r,L},
  {Q,R}=QRDecomposition[RG[n]];
  r=Diagonal[R];
  L=DiagonalMatrix[r/Abs[r]];
  Q.L
];
```

Compared to the previous code, this code is "clean" [1], since it does not depend on the way `QRDecomposition` is implemented in *Mathematica*.

Another option is to use the Gram-Schmidt process to find Q as shown in equation (31). This is what implicitly happens in our code in Sect. 3.1, when we use the *Mathematica*'s built-in function `Orthogonalize` [9]. When one uses the Gram-Schmidt algorithm, it is guaranteed that $R^{-1} \in \mathrm{T}(n)$, since in the kth iteration one computes a linear combination that contains $\boldsymbol{v}_k$ with a positive coefficient. However, the Gram-Schmidt algorithm is not numerically stable. Thus one might prefer to use the Householder's method. Actually, it is implicitly used in the above code, since `QRDecomposition` is implemented via Householder reflections (see Sect. 4.1). For more details see [1] and [12].

3.5 Resulting distribution

Now we know that decomposition (29) indeed holds and Lemma 1 tells us that the measure $\mathrm{d}\mu_G$ of the Ginibre ensemble is invariant under $\mathrm{U}(n)$. It remains to show that after we multiply Z (which is chosen from the Ginibre ensemble)

by $R^{-1} \in \mathrm{T}(n)$ as in equation (31), the obtained measure over $\mathrm{U}(n)$ will still be invariant. In other words, we want to show that the last line of our code in Sect. 3.1 does not spoil the invariance property of measure $\mathrm{d}\mu_G$.

Let us define an equivalence relation in $\mathrm{GL}(n, \mathbb{C})$ as follows:

$$Z \sim Z' \quad \Leftrightarrow \quad Z' = UZ, \quad \text{where } U \in \mathrm{U}(n). \tag{36}$$

Then we can use R to represent the equivalence class

$$[R] := \{\, QR \mid Q \in \mathrm{U}(n) \,\}. \tag{37}$$

Note that every equivalence class is a right coset of $\mathrm{U}(n)$ in $\mathrm{GL}(n, \mathbb{C})$. Since the measure $\mathrm{d}\mu_G$ is invariant under left-multiplication by $\mathrm{U}(n)$ and equivalence classes (37) are closed under left-multiplication by $\mathrm{U}(n)$, the restriction of $\mathrm{d}\mu_G$ to equivalence class $[R]$ is also left-invariant for every $R \in \mathrm{T}(n)$.

Now we can think of the Gram-Schmidt process as a map $\mathrm{GL}(n, \mathbb{C}) \to \mathrm{U}(n)$ that acts follows: $QR \mapsto Q$, i.e., it throws away the R part of $Z = QR$. In other words, it "forgets" to which equivalence class Z belongs to. When we collapse all equivalence classes to a single class isomorphic to $\mathrm{U}(n)$ via Gram-Schmidt process, the resulting measure will be invariant, since $\mathrm{d}\mu_G$ was invariant in each equivalence class. Another way to think about this is that the measure $\mathrm{d}\mu_G$ decomposes as a product of measures on $\mathrm{U}(n)$ and $\mathrm{T}(n)$, see Theorem 1 in [1].

This concludes our discussion on the correctness of the algorithm presented in Sect. 3.1 for generating unitaries uniformly at random. However, there are two more useful things that are related to our discussion. First, we will give more details on how to implement the QR decomposition using Householder reflections. Second, we will see how our algorithm for generating random unitaries fits into the "big picture".

4 Other interesting stuff

4.1 Implementing QR decomposition

QR decomposition is usually implemented using Householder reflections. It works as follows. Assume that for given $\boldsymbol{v} \in \mathbb{C}^n$ we construct a transformation $H(\boldsymbol{v}) \in \mathrm{U}(n)$ that depends on $\boldsymbol{v}$, such that

$$H(\boldsymbol{v}) \cdot \boldsymbol{v} = \|\boldsymbol{v}\|\, \boldsymbol{e}_1, \tag{38}$$

where $\boldsymbol{e}_1$ is the first basis vector of the standard basis of $\mathbb{C}^n$ (we will discuss the implementation of $H(\boldsymbol{v})$ in the next section). Let $Z_n \in \mathrm{GL}(n, \mathbb{C})$ and $\boldsymbol{v}_n$ be its first column. Then the first column of $H(\boldsymbol{v}_n) \cdot Z_n$ is $\|\boldsymbol{v}_n\|\, \boldsymbol{e}_1$, i.e.,

$$H(\boldsymbol{v}_n) \cdot \left(\begin{array}{c|c} \boldsymbol{v}_n & \cdots \end{array}\right) = \left(\begin{array}{c|c} \|\boldsymbol{v}_n\| & \\ \hline 0 & \\ \vdots & Z_{n-1} \\ 0 & \end{array}\right). \tag{39}$$

Similarly, let the first column of the lower $(n-1)\times(n-1)$ block Z_{n-1} in (39) be $\boldsymbol{v}_{n-1}\in\mathbb{C}^{n-1}$. Then we can find $H(\boldsymbol{v}_{n-1})\in \mathrm{U}(n-1)$, such that

$$\left(\begin{array}{c|ccc} 1 & 0 & \dots & 0 \\ \hline 0 & & & \\ \vdots & & H(\boldsymbol{v}_{n-1}) & \\ 0 & & & \end{array}\right)\cdot\left(\begin{array}{c|c|c} \|\boldsymbol{v}_n\| & & \\ \hline 0 & & \\ \vdots & \boldsymbol{v}_{n-1} & \\ 0 & & \end{array}\right)= \tag{40}$$

$$\left(\begin{array}{c|c|c} \|\boldsymbol{v}_n\| & & \\ \hline 0 & \|\boldsymbol{v}_{n-1}\| & \\ \hline 0 & 0 & \\ \vdots & \vdots & Z_{n-2} \\ 0 & 0 & \end{array}\right). \tag{41}$$

If we continue in a similar way, we end up with an upper-triangular matrix whose diagonal entries are positive, i.e.,

$$\tilde{H}(\boldsymbol{v}_1)\cdot\tilde{H}(\boldsymbol{v}_2)\cdot\ldots\cdot\tilde{H}(\boldsymbol{v}_{n-1})\cdot H(\boldsymbol{v}_n)\cdot Z = R, \tag{42}$$

where $R\in\mathrm{T}(n)$ and $\tilde{H}(\boldsymbol{v}_i) = I_{n-i}\oplus H(\boldsymbol{v}_i)$ as in equation (40). Thus

$$Z = \Big(H(\boldsymbol{v}_n)^\dagger\cdot\tilde{H}(\boldsymbol{v}_{n-1})^\dagger\cdot\ldots\cdot\tilde{H}(\boldsymbol{v}_2)^\dagger\cdot\tilde{H}(\boldsymbol{v}_1)^\dagger\Big)\cdot R \tag{43}$$

is the unique QR decomposition of Z. It remains to understand how one implements the transformation $H(\boldsymbol{v})$ satisfying equation (38).

4.2 Householder reflections

In this section we will describe how to implement a transformation that satisfies equation (38). To make thing simpler, let us first consider the real Euclidean space $\mathbb{R}^n$, instead of $\mathbb{C}^n$. It will be simple to go from $\mathbb{R}^n$ to $\mathbb{C}^n$ afterwards.

Given a unit vector $\boldsymbol{u}\in\mathbb{R}^n$, consider the following transformation

$$H := I - 2\,\boldsymbol{u}\cdot\boldsymbol{u}^{\mathsf{T}}. \tag{44}$$

Note that $H^{\mathsf{T}} = H$ and $H^2 = I$, thus the matrix H is symmetric and orthogonal. It is called *Householder reflection*, since H performs a reflection with respect to a hyperplane orthogonal to $\boldsymbol{u}$:

$$H\cdot\boldsymbol{w} = \begin{cases} -\boldsymbol{w} & \text{when } \boldsymbol{w} = c\,\boldsymbol{u}, \\ +\boldsymbol{w} & \text{when } \boldsymbol{w}\perp\boldsymbol{u}. \end{cases} \tag{45}$$

The nice thing about this transformation is that one does not need to explicitly compute the matrix representation of H in order to apply it on $\boldsymbol{w}$, since [12]

$$H\cdot\boldsymbol{w} = \boldsymbol{w} - 2\,\boldsymbol{u}\left(\boldsymbol{u}^{\mathsf{T}}\cdot\boldsymbol{w}\right). \tag{46}$$

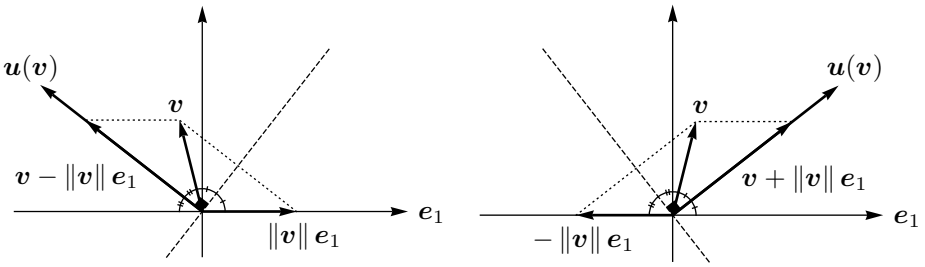

Figure 1: Householder reflection (48) with respect to a hyperplane that is orthogonal to $\boldsymbol{u}(\boldsymbol{v})$ defined in equation (49). When $v_1 < 0$, it sends $\boldsymbol{v}$ to $\|\boldsymbol{v}\|\,\boldsymbol{e}_1$ (the picture on the left), but when $v_1 > 0$, it sends $\boldsymbol{v}$ to $-\|\boldsymbol{v}\|\,\boldsymbol{e}_1$ (the picture on the right), so the sign must be corrected as in equation (50).

Let us use the above idea to construct a transformation $H(\boldsymbol{v}) \in \mathrm{O}(n)$ for given $\boldsymbol{v} \in \mathbb{R}^n$, such that $H(\boldsymbol{v}) \cdot \boldsymbol{v} = \|\boldsymbol{v}\|\,\boldsymbol{e}_1$. Clearly, any $n \times n$ orthogonal matrix whose first row is $\boldsymbol{v}/\|\boldsymbol{v}\|$ does the job. However, we would like $H(\boldsymbol{v})$ to be a Householder reflection, so that it can be implemented as in equation (46).

Let $\boldsymbol{u}(\boldsymbol{v}) \in \operatorname{span}\{\boldsymbol{v}, \boldsymbol{e}_1\}$ be a unit vector defined as follows[4]:

$$\boldsymbol{u}(\boldsymbol{v}) := \frac{\boldsymbol{v} - \|\boldsymbol{v}\|\,\boldsymbol{e}_1}{\big\|\boldsymbol{v} - \|\boldsymbol{v}\|\,\boldsymbol{e}_1\big\|}. \tag{47}$$

Note that $\boldsymbol{u}(\boldsymbol{v})$ is orthogonal to $\boldsymbol{v} + \|\boldsymbol{v}\|\,\boldsymbol{e}_1$ – the interior bisector of the angle between $\boldsymbol{v}$ and $\boldsymbol{e}_1$ (see the picture on the left in Fig. 1). We choose $H(\boldsymbol{v})$ to be the Householder reflection with respect to a hyperplane orthogonal to $\boldsymbol{u}(\boldsymbol{v})$:

$$H(\boldsymbol{v}) := I - 2\,\boldsymbol{u}(\boldsymbol{v}) \cdot \boldsymbol{u}(\boldsymbol{v})^{\mathsf{T}}. \tag{48}$$

One can check that $H(\boldsymbol{v})$ satisfies $H(\boldsymbol{v}) \cdot \boldsymbol{v} = \|\boldsymbol{v}\|\,\boldsymbol{e}_1$.

However, there is one problem with the definition (47) of $\boldsymbol{u}(\boldsymbol{v})$ – it does not work when $\boldsymbol{v} = \|\boldsymbol{v}\|\,\boldsymbol{e}_1$. In addition, it is not numerically stable as well. To see this, let $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$, where $v_1 > 0$ is large and all other components have small absolute value. Then the norm of $\boldsymbol{v} - \|\boldsymbol{v}\|\,\boldsymbol{e}_1$ is very small and thus the denominator of (47) is close to zero.

To avoid this, for $v_1 > 0$ we choose $\boldsymbol{u}(\boldsymbol{v})$ to be orthogonal to the bisector between $\boldsymbol{v}$ and $-\boldsymbol{e}_1$ instead (see the picture on the right in Fig. 1). In general we define $\boldsymbol{u}(\boldsymbol{v})$ as follows [1, 12]:

$$\boldsymbol{u}(\boldsymbol{v}) := \frac{\boldsymbol{v} + \operatorname{sgn}(v_1)\,\|\boldsymbol{v}\|\,\boldsymbol{e}_1}{\big\|\boldsymbol{v} + \operatorname{sgn}(v_1)\,\|\boldsymbol{v}\|\,\boldsymbol{e}_1\big\|}. \tag{49}$$

Note that this definition is consistent with (47) when $v_1 < 0$. It is also numerically stable, since the denominator of (49) will not be small, unless $\|\boldsymbol{v}\|$ is small.

[4]Our convention of signs in equations (47) and (48) differs from that of [1] in order to be consistent with the definition (44) of the Householder reflection.

One must be careful of how the sign function "sgn" is implemented though (e.g., `Sign[0]=0` in *Mathematica*). The choice of the value for sgn(0) is irrelevant, unless it is fixed and either $+1$ or -1.

Most practical implementations of QR decomposition use the definition (48) of $H(\boldsymbol{v})$ together with the definition (49) of $\boldsymbol{u}(\boldsymbol{v})$, see [1]. Unfortunately, in such case equation (38) no longer holds, since $H(\boldsymbol{v}) \cdot \boldsymbol{v} = \pm \|\boldsymbol{v}\| \boldsymbol{e}_1$. Thus the matrix R returned by QR decomposition (43) no longer has positive diagonal entries. This causes a problem when QR decomposition is used to generate unitaries uniformly at random [1].

This can be fixed by post-processing the output of the QR decomposition routine as described in Sect. 3.4. However, if one implements the QR decomposition from scratch using Householder reflections, one can simply incorporate the sign in the definition (48) of $H(\boldsymbol{v})$ as follows [1]:

$$H(\boldsymbol{v}) := -\operatorname{sgn}(v_1)\big(I - 2\, \boldsymbol{u}(\boldsymbol{v}) \cdot \boldsymbol{u}(\boldsymbol{v})^{\mathsf{T}}\big). \tag{50}$$

Note that this definition is consistent with (48) when $v_1 < 0$. Thus, when implementing the unique QR decomposition via Householder reflections, one should use the definition (50) of $H(\boldsymbol{v})$ with $\boldsymbol{u}(\boldsymbol{v})$ defined in (49).

It is straightforward to generalize equations (50) and (49) for $\boldsymbol{v} \in \mathbb{C}^n$ and $H(\boldsymbol{v}) \in \mathrm{U}(n)$:

$$H(\boldsymbol{v}) := -e^{-i\theta}\big(I - 2\, \boldsymbol{u}(\boldsymbol{v}) \cdot \boldsymbol{u}(\boldsymbol{v})^{\dagger}\big) \tag{51}$$

and

$$\boldsymbol{u}(\boldsymbol{v}) := \frac{\boldsymbol{v} + e^{i\theta}\, \|\boldsymbol{v}\|\, \boldsymbol{e}_1}{\big\|\boldsymbol{v} + e^{i\theta}\, \|\boldsymbol{v}\|\, \boldsymbol{e}_1\big\|}, \tag{52}$$

where $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ and $v_1 = e^{i\theta}\, |v_1|$. Note that this implementation is also numerically stable.

4.3 Subgroup algorithm

Let us conclude our discussion with an approach for generating random elements, that is common for both finite and compact groups. This approach is known as the *subgroup algorithm* and was developed by Diaconis and Shahshahani [13]. Sorry, I did not have time to finish this. You can find more information on this here: [1, 13, 14].

References

[1] Francesco Mezzadri, "How to Generate Random Matrices from the Classical Compact Groups," *Notices of the AMS*, Volume 54, Issue 5, pp. 592-604, May 2007. arXiv:math-ph/0609050v2
http://www.ams.org/notices/200705/fea-mezzadri-web.pdf

[2] Persi Diaconis, "What is... a random matrix?" *Notices of the AMS*, Volume 52, Issue 11, pp. 1348-1349, December 2005.
http://www.ams.org/notices/200511/what-is.pdf

[3] A bunch of authors, "Quantum Physics from A to Z". arXiv:quant-ph/0505187v4

[4] Simon Rubinstein-Salzedo, "On the Existence and Uniqueness of Invariant Measures on Locally Compact Groups," 2004. http://www.artofproblemsolving.com/LaTeX/Examples/HaarMeasure.pdf

[5] Eric W. Weisstein, "Sphere Point Picking," MathWorld. http://mathworld.wolfram.com/SpherePointPicking.html

[6] Karol Życzkowski, Marek Kuś, "Random unitary matrices," *J. Phys. A: Math. Gen.*, Volume 27, Number 12, pp. 4235–4245, 1994.

[7] Luis J. Boya, E. C. G. Sudarshan, Todd Tilma, "Volumes of Compact Manifolds," *Reports on Mathematical Physics*, Volume 52, Issue 3, pp. 401–422, December 2003. arXiv:math-ph/0210033v3

[8] George Cybenko, "Reducing Quantum Computations to Elementary Unitary Operations," *Computing in Science and Engineering*, vol. 3, no. 2, pp. 27–32, 2001.

[9] *Mathematica* documentation on `Orthogonalize`. http://reference.wolfram.com/mathematica/ref/Orthogonalize.html

[10] "Zappa-Szép product," Wikipedia. http://en.wikipedia.org/wiki/Zappa-Szép_product

[11] "Gram-Schmidt process," Wikipedia. http://en.wikipedia.org/wiki/Gram-Schmidt_process

[12] Lenka Čížková, Pavel Čížek, "Matrix Decompositions", Chapter II.4.1 in "Handbook of Computational Statistics: Concepts and Methods," pp. 104, Birkhäuser, 2004. Editors: James E. Gentle, Wolfgang Härdle, Yuichi Mori. http://mars.wiwi.hu-berlin.de/ebooks/html/csa/node36.html

[13] Persi Diaconis, Mehrdad Shahshahani, "The subgroup algorithm for generating uniform random variables," *Prob. Eng. Inf. Sc.* **1**, pp. 15–32, 1987.

[14] G. W. Stewart "The Efficient Generation of Random Orthogonal Matrices with an Application to Condition Estimators," *SIAM Journal on Numerical Analysis*, Vol. 17, No. 3, pp. 403–409, June 1980.