MATHEMATICS OF COMPUTATION
Volume 85, Number 298, March 2016, Pages 783–797
http://dx.doi.org/10.1090/mcom/2986
Article electronically published on June 17, 2015

# HOW TO PICK A RANDOM INTEGER MATRIX? (AND OTHER QUESTIONS)

IGOR RIVIN

Abstract. We discuss the question of how to pick a matrix uniformly (in an appropriate sense) at random from groups big and small. We give algorithms in some cases, and indicate interesting problems in others.

## 1. Introduction

In a number of papers (see, for example, [6,9,28,30]) results are proved about the behavior of a typical element of a lattice in a semisimple Lie Group (for example, $\mathrm{SL}(n,\mathbb{Z})$, where "typical" means picked uniformly at random from from all matrices in the group with (for example) Frobenius norm bounded above by a constant $X$. While these results are often enlightening, what is not addressed is how one might actually pick such a matrix. In this paper I try to address this question.

Suppose you are asked to pick a matrix uniformly at random from all the matrices $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ in $\mathrm{SL}(2,\mathbb{Z})$ such that the the Frobenius norm $\|M\|$, defined as $\|M\| = \sqrt{\operatorname{tr} MM^t} = \sqrt{a^2+b^2+c^2+d^2}$, is at most $X$. The simplest method is to pick a random matrix (this is already not quite trivial, and is the content of the function PickMatrix, defined in Algorithm 2.2) in $M^{2\times 2}(\mathbb{Z})$ satisfying the norm bound, check whether the determinant is equal to 1, throw it away if it is not, and return it if it is. Now, note that the number of matrices in $M^{2\times 2}(\mathbb{Z})$ satisfying the norm bound is of order $X^4$. On the other hand, it is known that the number of elements of $\mathrm{SL}(2,\mathbb{Z})$ satisfying the norm bound is asymptotic to $6X^2$ (see [23]), which means that the expected number of attempts before we succeed is of order $O(X^2)$, which is *exponential* in the size of the input (which is $O(\log X)$). Below, we will describe in detail a *polynomial time* algorithm for choosing a matrix from $\mathrm{SL}(2,\mathbb{Z})$ with norm bounded above by $X$. This algorithm is *transcendental,* not discrete, which is a little surprising. It is an approximation algorithm, in the following sense: if in default form the biggest ratio of the probabilities of selecting matrices $A$ and $B$ is $\exp(1+\epsilon)$, we can make the ratio $\exp(1+\epsilon/k)$ at the cost of increasing the running time of the algorithm by a factor of $k$.

The rest of the paper is organized as follows: First, we discuss the baby version of the question (how to write the function PickMatrix in the naïve Algorithm 1.1).

Received by the editor February 15, 2014 and, in revised form, July 6, 2014 and August 19, 2014.

2010 *Mathematics Subject Classification.* Primary 20H05, 20P05, 20G99.

*Key words and phrases.* Groups, lattices, matrices, randomness, probability.

The author would like to thank Nick Katz , Chris Hall, Peter Sarnak, and Hee Oh for helpful conversations. He would also like to thank ICERM and Brown University for their hospitality and generous support. The author would like to thank the anonymous referees for their patience and helpful suggestions.

**Algorithm 1.1** Naive algorithm for picking random elements from $\mathrm{SL}(2,\mathbb{Z})$

**Require:** $X$ is a real number greater than or equal to $\sqrt{2}$.
1: **function** PickSLMatrix1($X$)
2: $M \leftarrow \mathrm{PickMatrix}(X, 2)$ ▷ $\mathrm{PickMatrix}(n, X)$ returns a uniformly distributed $n \times n$ integer matrix with Frobenius norm at most $X$.
3: **while** $\det M \neq 1$ **do**
4: $M \leftarrow \mathrm{PickMatrix}(X, 2)$
5: **end while**
6: **return** $M$
7: **end function**

Then we will discuss $\mathrm{SL}(2,\mathbb{Z})$ in detail, and discuss how the algorithm might be extended to other matrix groups, including $\mathrm{SL}(n,\mathbb{Z})$ for arbitrary $n$.

Finally, we briefly discuss the situation for finite matrix groups.

1.1. **How to produce random numbers with a given density?** Suppose we have a positive function $f$ defined on the interval $(0, R]$, and we want to produce random numbers whose density at $x$ is proportional to $f(x)$ (when all we are given is a source of *uniform* random numbers on $[0, 1]$). This turns out to be easier than one might have thought, and described in Algorithm 1.2. To show that Algorithm 1.2 works, we note that the probability that GenRandom $R$ is between $t$ and $t + \Delta t$ is the probability that $x$ (on line 3 of Algorithm 1.2) is between $F(t)$ and $F(t+\Delta t)$, which is about $f(t)\Delta t$, as advertised.

**Algorithm 1.2** Generating random numbers with a given probability density $f$ on $[0, R]$

**Require:** $R$ is a real number greater than 0, $f$ a positive function on $[0, R]$.
1: **function** GenRandom($f$, $R$)
2: $F(t) \leftarrow \int_0^t f d\lambda$. ▷ $F$ is the anti-derivative of $f$.
3: $x \leftarrow F(R)\,\mathrm{random}()$ ▷ Generate uniform random number between 0 and $F(R)$.
4: **return** $F^{-1}(x)$. ▷ Where $F^{-1}$ is the inverse function.
5: **end function**

## 2. Geometric preliminaries

2.1. **Uniform random points in balls.** Suppose we want to generate a uniformly distributed random point in a ball of radius $R$ in $\mathbb{R}^n$. This point will have a radius and a spherical coordinate, so we generate these separately. For the spherical coordinate, it is well known that a vector whose coordinates are identical independently distributed Gaussians has its direction uniformly distributed on the unit sphere (a fast in practice method of generating this is the Box-Muller method [1]). As for the radius, we invoke Algorithm 1.2, and generate a random number between 0 and $R^n$, then take its $n$-th root, multiplied by $R$.

Suppose now that we want to generate a random uniform point from a disk in the hyperbolic plane. The angle here is even easier (a uniform random number between 0 and $2\pi$ will do). As for the radius, we know that the area of a disk of

radius $R$ in the hyperbolic plane is $2\pi(\cosh R - 1)$, so to generate our radius, we compute a random number $x$ between 0 and $\cosh R - 1$, then use $\operatorname{arccosh}(x+1)$ as the radius. We summarize this as follows:

**Algorithm 2.1** Picking a point uniformly at random from a hyperbolic disk of radius $R$

**Require:** $X$ is a positive real number.

```
function PickHyperbolic(R)   ▷ Returns the ordered pair of radius, angle.
    x ← (cosh R − 1) random()
    θ ← 2π random()
    return (arccosh(x + 1), θ)
end function
```

2.2. **Computing a random integer matrix.** How do we write our procedure PickMatrix? The first observation is that a random $n \times n$ matrix with Frobenius norm bounded by $X$ is simply an $n^2$-tuple of integers $a_{11}, a_{12}, \ldots, a_{nn}$ with

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2 \le X^2,$$

so we are looking for a uniformly distributed integer lattice point in the ball of radius $X$ in $\mathbb{R}^{n^2}$. The simplest (combinatorial) way to pick such a point is to pick a lattice point in the cube $[-X, X]^{n^2}$, and then throw out those points with norm bigger than $X$. This is a perfectly fine algorithm in small dimensions, but it degrades horribly in high dimensions, since the ratio of the volume of the ball to the ratio of circumscribed cube goes to zero super-exponentially as dimension goes to infinity.[1] Instead, the following is an efficient algorithm:

**Algorithm 2.2** Picking a random lattice vector of $L^2$ norm bounded by $X$ in $\mathbb{R}^n$

**Require:** $X$ is a positive real number.

```
function PickLatticeVector(n,X)
    loop
        x ← Random vector in R^n of norm bounded above by X + √n.
        v ← closest lattice point to x.
        if ‖v‖ ≤ X then
            return v
        end if
    end loop
end function
function PickMatrix(n, X)
    return PickLatticeVector(n^2, X)
end function
```

Note that the additive constant of $\sqrt{n}$ (the length of a diagonal of a unit cube in $\mathbb{R}^n$ (and the consequent possible resampling) is added to eliminate "edge effect"—without it, the probabilities of choosing numbers close to the norm bound would be

[1] In particular, for $4 \times 4$ matrices, we will reject around 300000 matrices for each one accepted.

different from that of choosing smaller numbers. The additive constant is chosen in such a way that for every candidate lattice point its Voronoi cell is contained in the ball.

## 3. Action of $\mathrm{SL}(2,\mathbb{R})$ and $\mathrm{SL}(2,\mathbb{Z})$ on the upper half-plane

Recall that $\mathrm{SL}(2,\mathbb{R})$ acts on the upper half-plane $H = \{z \,|\Im z > 0\}$ by

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} z = \frac{az+b}{cz+d}.$$

Recall also that we can define a metric on $H$ by setting

$$d(z,w) = \operatorname{arccosh}\left(1 + \frac{|z-w|^2}{2\Im z \Im w}\right),$$

and, equipped with this metric, $H$ is isometric to the hyperbolic plane $\mathbb{H}^2$. Infinitesimally, the area form of this metric is given $\frac{dxdy}{y^2}$, where $y = \Im z$. In addition, the action of $\mathrm{SL}(2,\mathbb{R})$ by linear fractional transformations described above is isometric, and, indeed, every orientation preserving isometry of $\mathbb{H}^2$ is obtained this way, so

$$\operatorname{Isom}\mathbb{H}^2 \simeq P\,\mathrm{SL}(2,\mathbb{R}) = \mathrm{SL}(2,\mathbb{R})/\{\pm I\},$$

where the quotient by plus and minus identity is needed because $(-I)z = \frac{-z}{-1} = z$, for all $z \in H$. We will also need the *singular value decomposition*. Recall that every matrix $A$ in $M^{m\times n}$ can be written as $A = PDQ$, where $P \in O(m)$, $Q \in O(n)$, and $D$ is diagonal $m \times n$ matrix with nonnegative diagonal elements (see, e.g., [12]). The diagonal elements of $D$ are known as the *singular values* of $A$. It is well known (and easy to verify) that the Frobenius norm of $A$ equals the Euclidean ($L^2$) norm of the vector of its singular values.

In the special case where $n = m = 2$, and $\det A = 1$, it is easy to see that the above implies that $A$ can be written as

$$A = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} x & 0 \\ 0 & \frac{1}{x} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix},$$

for some $x > 1$. Further, as noted above, $\|A\|^2 = x^2 + 1/x^2$.

3.1. **Translation distance.** A big part of the reason for introducing the singular value decomposition above is to give a palatable answer to the following question:

**Question 3.1.** How far (in hyperbolic metric) does the matrix $A = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}(2,\mathbb{R})$ move the point $i$?

The main reason why the singular value decomposition helps is that

$$\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} i = i,$$

so with $A$ as above, we have

$$d(i, A(i)) = d(i, ix^2) = 2\log x.$$

Since

$$\|A\|^2 = x^2 + \frac{1}{x^2},$$

it follows that

$$\|A\|^2 = 2\cosh d(i, A(i)). \tag{1}$$

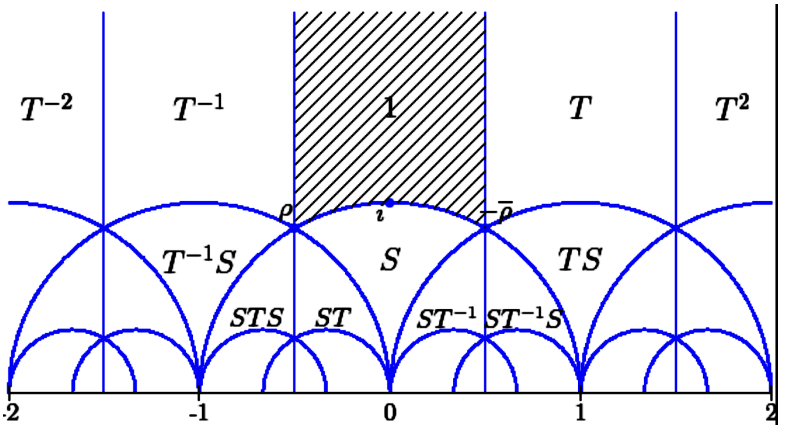


FIGURE 1. The modular tessellation; fundamental domain shaded, other copies labeled by the elements sending the shaded domain to the copy. Here $T(z) = z + 1, S(z) = -1/z$.

As a minor bonus, we can now modify our procedure PickHyperbolic to return a point in the upper half-plane in procedure PickHalfplane (see Algorithm 3.1).

**Algorithm 3.1** Picking a random point in the disk around $i$ in the Poincaré half-plane model

**Require:** $R$ a positive real number.
1: **function** PICKHALFPLANE($R$)
2: $(r, \theta) \leftarrow$ PickHyperbolic($R$)
3: **return** $\dfrac{ie^r \cos\theta + \sin\theta}{-ie^r \sin\theta + \cos\theta}$
4: **end function**

3.2. **The fundamental domain and orbits of the** $\mathrm{SL}(2,\mathbb{Z})$ **action.** The action of $\mathrm{SL}(2,\mathbb{Z})$ on $H$ is discrete, and its fundamental domain $\Lambda$ is one of the best known images in all of mathematics (the reader can see it again in Figure 1, which shows the *modular tessellation*—the tiling of the upper half-plane by the images of $\Lambda$ by elements of $\mathrm{SL}(2,\mathbb{Z})$). Geometrically, $\Lambda$ is a triangle with angles $\frac{\pi}{3}, \frac{\pi}{2}, 0$. The vertex corresponding to the last angle is an *ideal vertex* (the point $\infty$ in the Figure 1), also known as a *cusp*. A cusp neighborhood at height $w$ is the region $H_w = \{z \in H \,|\, 0 < \Re z < 1; \Im z > w\}$. A simple integration shows that

$$\mathrm{area}(H_w) = \frac{1}{w}.$$

This immediately implies the following observation:

*Fact* 3.2. The area of the complement in $\Lambda$ of a circle of radius $R$ around $i$, for large $R$ has area approximately $\exp(-R)$.

The points in the fundamental domain index the orbit of the $\mathrm{SL}(2,\mathbb{Z})$ action, and gives rise to the following natural question:

**Question 3.3.** Given a point $z \in H$, which orbit is it in? In other words, which point of $\Lambda$ gets mapped to $z$?

This question is so natural it was asked and answered in the 18th century by Legendre and Gauss. Of course, for them, the question was a little different: they were given two linearly independent vectors in the plane. These vectors generate a lattice, and the question is: what is the canonical form for that lattice? In other words, Gauss and Legendre posed (and solved) the two-dimensional *lattice reduction* problem (a very nice reference is the paper [33]). Gauss's algorithm (which is basically the continued fraction algorithm) proceeds as follows:

**Algorithm 3.2** Gauss's lattice reduction algorithm

**Require:** A complex number $z$ with $\Im z \geq 0$.
1: **function** REDUCE($z$)
2: **while** $|z| \leq 1$ **do**
3: $z \leftarrow -1/z$
4: $q \leftarrow \text{round}\, \Re z$
5: $z \leftarrow z - q$
6: **end while**
7: **return** $z$
8: **end function**

In fact, Algorithm Reduce can be made to do more: give the point $z \in H$, we can return not just the point $z_0 \in \Lambda$ such that $z$ is in the orbit of $z_0$, but also the matrix $A \in \mathrm{SL}(2,\mathbb{Z})$ such that $z_0 = Az$, as done in Algorithm 3.3.

**Algorithm 3.3** Lattice reduction keeping track of the matrix

**Require:** A complex number $z$ with $\Im z \geq 0$.
**function** REDUCE2($z$)
$A \leftarrow I$
**while** $|z| \leq 1$ **do**
$z \leftarrow -1/z$
$A \leftarrow \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} A$
$q \leftarrow \text{round}\, \Re z$
$z \leftarrow z - q$
$A \leftarrow \begin{pmatrix} 1 & q \\ 0 & 1 \end{pmatrix} A$
**end while**
**return** $(A, z)$
**end function**

## 4. Selecting a random element of $\mathrm{SL}(2,\mathbb{Z})$ almost uniformly.

We are now ready to describe the algorithm for selecting a random matrix $M$ from the set of matrices in $\mathrm{SL}(2,\mathbb{Z})$ with Frobenius norm bounded above by $X$. Aside from the observations above, the key remark is that since the hyperbolic plane is the homogeneous space of $\mathrm{SL}(2,\mathbb{R})$ (that is, the quotient of $\mathrm{SL}(2,\mathbb{R})$ by its maximal compact subgroup $\mathrm{SO}(2,\mathbb{R})$) the Haar measure on $\mathrm{SL}(2,\mathbb{R})$ projects to the hyperbolic metric (see the discussion in [2, 4] for more on the subject, and A. Knapp's book [16] for everything you ever wanted to know). This suggests the following algorithm:

**Algorithm 4.1** Returns a matrix in $\mathrm{SL}(2,\mathbb{Z})$ with Frobenius norm bounded by $X$. The ratio of the probabilities of any two matrices is between $e^{\epsilon}$ and $e^{-\epsilon}$

**Require:** A pair of positive real numbers $X, \epsilon$
1: **function** PickFancy($X$,$\epsilon$)
2: $R \leftarrow f(X,\epsilon)$ ▷ $f$ is a function to be named later
3: **loop**
4: $z \leftarrow$ PickHalfplane($R$)
5: $(A, z_0) \leftarrow$ Reduce2($z$)
6: **if** $\|A\| \le X$ **then**
7: **return** $A$
8: **end if**
9: **end loop**
10: **end function**

What should $f(X,\epsilon)$ be? First, it is obviously necessary that the disk of radius $f(X,\epsilon)$ intersect *all* of the images of $i$ by matrices $A$ with $\|A\| < X$. As we have seen (eq. (1)), in order for this to be true, we must have $f(X,\epsilon) > \operatorname{arccosh} X^2/2$. On the other hand, the fundamental domain $\Lambda$ of $\mathrm{SL}(2,\mathbb{Z})$ has a cusp, which is bad, since no disk can contain $\Lambda$, but not *so* bad, since the part of $\Lambda$ which lies outside the disk of radius $R$ around $i$ is asymptotic to $\exp(-R)$. This means that if $f(X) > t + \operatorname{arccosh} X^2/2$, the ratio of the areas of the intersections of fundamental domains we are interested in is of order $1 + e^{-t}$. On the other hand, the area of the set of points that lie in fundamental domains we *do not* want is proportional to $e^t$. This is so, since the total area of "good" points is bounded (by $\pi X^2$) independently of $t$. Therefore, as claimed in the introduction, the amount of excess computation is proportional to the error.

4.1. **Complexity estimates and implementation.** Picking the random complex number in the half-plane in the function PickHalfplane has been made unnecessarily expensive. Unwinding what we are doing, we see that in the first step we pick a random number $x$ between 0 and $g_C(X) = \cosh(C + \operatorname{arccosh} 2X^2) - 1$. Note that

$$\begin{aligned} g_C(X) &= \cosh(C + \operatorname{arccosh} X^2/2) - 1 \\ &= \cosh C \cosh \operatorname{arccosh} X^2/2 - \sinh C \sinh \operatorname{arccosh} X^2/2 - 1 \\ &= \cosh CX^2/2 - \sinh C\sqrt{x^4/4 - 1} - 1 = O_C(X^2). \end{aligned}$$

In the next step we compute $\operatorname{arccosh}(x+1) = \log(x+1+\sqrt{x^2+2x})$. Since the number of fundamental domains is exponential in the radius, we need roughly $\log X$ bits of precision, and the final step (Reduce2) then takes a logarithmic number of steps (see [17, 34]), each of which is of logarithmic complexity, so the running time is of the order of $O(t \log^2 X)$.

## 5. Extensions to other Fuchsian and Kleinian groups

Suppose that instead of $\mathrm{SL}(2,\mathbb{Z})$ we want to generate random elements of bounded norm from other subgroups of $\mathrm{SL}(2,\mathbb{R})$ or, even more ambitiously, $\mathrm{SL}(2,\mathbb{C})$. The general approach described above works. Suppose $H$ is our (discrete) subgroup. To pick a random element, we pick a random point $x$ in $\mathbb{H}^2$ or $\mathbb{H}^3$ (our radius computation goes through unchanged), then find the matrix $A \in H$ which moves $x$ to the "canonical" fundamental domain of $H$. This last part, however, is not so obvious, because both questions (constructing the fundamental domain and "reducing" the point $x$ to that fundamental domain) are nontrivial.

5.1. **Constructing the fundamental domain.** The first observation is that if the group $H$ is not geometrically finite, it does not have a finite-sided fundamental domain at all, so constructing one may be too much. It is, however, conceivable that deciding whether $x$ is reduced (that is, lies in the canonical fundamental domain) is still decidable. Since no algorithm leaps to mind, we shall state this as a question:

**Question 5.1.** Is there a decision procedure to determine whether $x \in \mathbb{H}^n$ lies in the canonical fundamental domain for a not-necessarily-geometrically finite group $H$?

Until Question 5.1 is resolved, we will assume that $H$ is geometrically finite. Now, we can construct the fundamental domain by generating a chunk of the orbit of the base point, and then computing the Voronoi diagram of that point set; the resulting domains are the so-called *Dirichlet* fundamental domains. Computing the Voronoi diagram can be reduced to a Euclidean computation (see the elegant exposition in [26], and H. Edelsbrunner's recent classic [3] for background on the various diagrams). However, a much harder problem is of figuring out how much of an orbit needs to be computed. For Fuchsian groups, this was addressed by Jane Gilman in her monograph [8] (at least for two-generator Fuchsian groups). For Kleinian groups the question is that much harder, but has been studied at least for *arithmetic* Kleinian groups in [27]. All we can say in general is that the computation is finite (since at every step we check the conditions for the Poincaré polyhedron theorem), so after waiting for a finite (though possibly long) time, we are good to go. Now, the question is: lacking the number theory underlying the continued fraction algorithm, how do we *reduce* our random point to the canonical fundamental domain? There are a number of ways to try to emulate the continued fraction algorithm. Algorithm 5.1 is one.

Algorithm 5.1 will terminate in *at most* exponential time (exponential in $d(b,x)$, that is), and it seems very plausible (for reasons of hyperbolicity) that it will actually terminate in time *linear* in $d(b,x)$, but this seems difficult to show.

## 6. Higher rank

6.1. $\mathrm{SL}(n,\mathbb{Z})$**.** The algorithms for $\mathrm{SL}(2,\mathbb{Z})$ use, in essence, the $KAK$ decomposition of the group (which is in this case the singular value decomposition). This

**Algorithm 5.1** Greedy reduction algorithm

**Require:** $x, b \in \mathbb{H}^n$, side-pairing transformation of the Dirichlet domain $\Gamma = \{\gamma_0 = I(n), \gamma_1, \ldots, \gamma_k\}$

```
function GreedyReduce(x, b, Γ)                    ▷ b is the basepoint.
    M ← I(n)
    loop
        Loop over Γ to find the i ∈ [0, k] for which d(γ_i(x), b) is minimal.
        if i = 0 then
            return M
        end if
        M ← γ_i M
        b ← γ_i b
    end loop
end function
```

exists, and is easy to describe geometrically, in the higher rank case as well (this construction is due to Minkowski). We first introduce the *positive definite cone*

$$\mathrm{PSD}(n) = \{M \,|\, M = M^t, v^t M v \geq 0, \forall v \in \mathbb{R}^n\}.$$

The general linear group $\mathrm{GL}(n, \mathbb{R})$ acts on $\mathrm{PSD}(n)$ by $g(M) = gMg^t$. It is not immediate that the subset $\mathrm{PSD}_1(n) = \{M \in \mathrm{PSD}(n) \,|\det M = 1\}$ is invariant under $\mathrm{SL}(n, \mathbb{R})$. We can define a family of (Finsler) metrics on $\mathrm{PSD}(n)$ by

$$d_p(A, B) = \left( \sum_{i=1}^{n} |\log \sigma_i(B^{-1}A)|^p \right)^{1/p},$$

where $\sigma_i(M)$ denotes the $i$-th singular value of $M$. When $p = 2$ this defines a Riemannian metric, which makes $\mathrm{PSD}_1(n)$ into the symmetric space for $\mathrm{SL}(n, \mathbb{Z})$. In particular, when $n = 2$ it is easy to check that $\mathrm{PSD}_1(2)$ is the hyperbolic plane $\mathbb{H}^2$ with the usual metric. With this in place, the algorithm we described for $\mathrm{SL}(2, \mathbb{Z})$ goes through *mutatis mutandis*. The hard part is the reduction algorithm. In the setting of $\mathrm{SL}(n, \mathbb{Z})$ we have the *lattice reduction* problem, which has been heavily studied starting with L. Lovasz's foundational LLL algorithm in [18]. The LLL algorithm is generally used as an *approximation* algorithm: it reduces a point not into the fundamental domain but into a point near the fundamental domain, which begs the question:

**Question 6.1.** Are the matrices obtained in the LLL algorithm uniformly distributed?

In any case, one can also perform *exact* lattice reduction, but in that case the running time is exponential in dimension (see [24]); for dimensions up to four there is an extension of the Legendre-Gauss algorithm, described above, which is exact and quadratic in terms of the bit-complexity of the input; see [25].

6.2. $\mathrm{Sp}(2n, \mathbb{Z})$**.** For $\mathrm{Sp}(2n, \mathbb{R})$ the symmetric space is the Siegel half-space, where the metric is defined in the same way as for $\mathrm{SL}(n, \mathbb{R})$, while the underlying space is not the positive semi-definite cone, but instead the set $S(2n)$ of all *complex* symmetric matrices with positive-definite imaginary part. A symplectic matrix

$X \in \mathrm{Sp}(2n, \mathbb{R})$ has the form $X = \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)$ where $A, B, C, D$ are $n \times n$ matrices satisfying the conditions that

$$A^t C = C^t A,$$
$$B^t = D^t B.$$

The action of $\mathrm{Sp}(2n, \mathbb{R})$ on $S(2n)$ is then given by:

$$X(Z) = (AZ + B)(CZ + D)^{-1}.$$

For more details on this, see [5, 31]. In any case, the action of $\mathrm{Sp}(2n, \mathbb{Z})$ on the Siegel half-space is fairly well understood, and the algorithm we gave for $\mathrm{SL}(2, \mathbb{Z})$ (which is also known as $\mathrm{Sp}(2, \mathbb{Z})$) goes through, with the usual question of lattice reduction, which has *not* been studied very extensively; the only reference I have found was [7], which is, however, quite thorough.

## 7. Other miscellaneous groups

7.1. **The orthogonal group.** Even without integrality assumptions, it is not immediately obvious how to sample a uniformly random matrix from the orthogonal group. This question got a very elegant one-line answer from G. W. Stewart in his paper [32]. Stewart's basic method is as follows: First, we remark that it is well known that every matrix $M$ possesses a $QR$ decomposition, where $Q$ is orthogonal, while $M$ is upper triangular, and this decomposition is unique up to post-multiplying $Q$ by a diagonal matrix whose elements are $\pm 1$. This indeterminacy can be normalized away by requiring the diagonal elements of $R$ to be positive. The algorithm is now the following (Algorithm 7.1):

**Algorithm 7.1** Generating matrices in $O(n)$ uniform with respect to the Haar measure

**Require:** $n$ is a positive integer.
  **function** RandomOrthogonal($n$)
    $X \leftarrow$ an $n \times n$ matrix whose entries are independent $N(0, 1)$ random variables
    $(Q, R) \leftarrow$ the QR decomposition of $X$
    **return** $Q$
  **end function**

This algorithm works because the distribution of $KX$ is the same as the distribution of $X$ for a matrix $X$ with i.i.d. normal entries, and so the distribution of $KQ$ is the same as the distribution of $Q$, which is exactly what we seek (notice that this method is morally a slight extension of the method described in Section 2.1), and is also morally related to our algorithms for $\mathrm{SL}(n, \mathbb{Z})$.

Now generating random *integral* matrices in $O(n)$ is easy—they are just the signed permutation matrices, and generating a random permutation is easy (in a quest for self-containment we give the algorithm below as Algorithm 7.2, as is assigning random signs. However, as far as I know there is no known way to generate uniformly random *rational* orthogonal matrices. We ask this as a question:

**Question 7.1.** How do we generate a random element of $O(n)$ whose elements have greatest common denominator bounded above by $N$?

There is a natural companion question:

**Question 7.2.** Let $O_q(n)$ be the set of those elements of $O(n)$ with rational entries, such that the size of the greatest common denominator is bounded above by $q$. Is there any exact or asymptotic formula for the order of $|Q_q(n)|$?

Another natural question is:

**Question 7.3.** Let $\mu_q$ be the normalized counting measure on $O_q(n)$ (as above). Do the measures $\mu_q$ converge weakly to the Haar measure on the orthogonal group?

Questions related to Questions 7.2 and 7.3 are considered in the paper [10], and it is quite plausible that the methods extend, but it is not completely obvious as of this writing. The only thing we know with certainty is how to address the case of $SO(2)$. Here, the elements have the form $\left(\begin{smallmatrix} a & b \\ -b & a \end{smallmatrix}\right)$, with $a^2 + b^2 = 1$. Thus, if $a$ and $b$ have denominator $q$, we are counting the representations of $q$ as a sum of two squares. For this there is the explicit formula of Dirichlet:

> If $q = p_1^{2a_1} \dots p_k^{2a_k} q_1^{b_1} \dots q_l^{b_l}$, where $p_i = 4k_i + 3$, in which $q_j = 4k_j + 1$, then the number of ways to write $q$ as a sum of two squares is $\prod_{j=1}^{l}(b_j + 1)$.

To get an asymptotic result, it is necessary to consider all $q \leq Q$, when we see that the number of elements in $SO(2)$ with the greatest common divisor of coefficients equals the number of *visible* lattice points in the disk $\|x\| \leq Q$ (a visible point $(a, b)$ is a lattice point with relatively prime $a, b$). Since the probability of a lattice point being relatively prime for $Q \gg 1$ approaches $6/\pi^2$, and the number of lattice points in the disk is asymptotic to $\pi Q^2$, we see that the cardinality of $SO_Q(2)$ is asymptotic to $\frac{6}{\pi} Q^2$, so we have a rather satisfactory answer to Question 7.2 in this setting.

Question 7.3 is also easy (but already deep) in this setting. It is equivalent to the equidistribution of rational numbers with bounded denominator in the interval, and that, in turn, is not hard to show it is equivalent to the prime number theorem (both statements are equivalent to the statement that $\sum_{k=1}^{x} \mu(x) = o(x)$, where $\mu$ is the Möbius function).

Finally, in view of the answer to Question 7.2, Question 7.1 is equivalent to the question of generating a lattice point in a ball, which we have already discussed in Section 2.1.

**Algorithm 7.2** Generating a random permutation uniformly

**Require:** $n > 0$
1: **function** GenPerm($n$)
2: $a \leftarrow [1, 2, 3, \ldots, n]$
3: **for** $i = 1 \rightarrow n$ **do**
4: swap $a[1]$ and $a[n - i + 1]$
5: **end for**.
6: **return** $a$
7: **end function**

7.2. **Finite Linear Groups.** Our final remarks are on finite linear groups. The simplest class of groups to deal with is $\mathrm{SL}(n, p)$ How do we get a random element? This is quite easy; see Algorithm 7.3:

**Algorithm 7.3** Generating a random element of $\mathrm{SL}(n, p)$.

**Require:** $n > 0$
1: **function** GenRandSL($n$)
2: **loop**
3: $a \leftarrow$ a uniformly random element of $M^{n\times n}(p)$.
4: **if** $\det(a) \neq 0$ **then**
5: **return** $a$ with the first column divided by $\det(a)$.
6: **end if**
7: **end loop**
8: **end function**

We pick every element independently at random from $F_p$. If the resulting matrix $M$ is singular, we try again, if not, let the determinant be $d$. We then divide the first column of $M$ by $d$. It is easy to see that the resulting matrix $M'$ will be uniformly distributed in $\mathrm{SL}(n, p)$. It is easy to see that the complexity of this method is $O(n^\omega \log p)$, where $\omega$ is the optimal matrix multiplication exponent. Unfortunately, this simple method only works for $\mathrm{SL}(n, q)$. For $\mathrm{Sp}(2n, q)$ there is Algorithm 7.4, which is due to Chris Hall. It is not hard to see that Chris Hall's algorithm has time complexity $O(n^3 \log p)$.

In general, there is a completely different polynomial-time algorithm based on the fact that the Cayley graphs of simple groups of Lie type are expanders—uniform expansion bounds have been obtained by a number of people; see [13,14,19,20]. The main significance of the expansion for our purposes is that the random walk on the Cayley graph is very rapidly mixing (see [11, Section 3]), and so a random walk of polylogarithmic length will be equidistributed over the group. Of course, this will be slower than Algorithm 7.3 , and will only generate *approximately* uniform random elements. To be precise, the diameter of the Cayley graph of (for example) $\mathrm{SL}(n, p)$ will be $O(n^2 \log p)$, so the expander-based algorithm will have time complexity $O(\log^2 p n^{\omega+2})$.

7.3. $\mathrm{SL}(n, \mathbb{R})$**.** The method described in Section 4 can be easily adapted to uniformly select a matrix from $\mathrm{SL}(n, \mathbb{R})$ of bounded Frobenius norm (indeed, this is much easier than if the coefficients are constraints to be integers). The basic observation (see [15, p. 142]) is to use the $KAK$ (singular value) decomposition. The

**Algorithm 7.4** Chris Hall's algorithm to generate a random element of $\mathrm{Sp}(2n, p)$.

**Require:** $n > 0$
$V \leftarrow$ symplectic vector space of dimension $2n$.
**function** GenRandSp($n$)
  $W \leftarrow \{0\}$
  **for** $i = 1 \to n; i \leftarrow i + 1$ **do**
    **repeat**
      $x, y \leftarrow$ random vectors in $V$.
      $x', y' \leftarrow$ projections of $x, y$ onto $W$.
      $x'' \leftarrow x - x'$
      $c \leftarrow \langle x'', y'' \rangle$
    **until** $c \neq 0$
    $x_i \leftarrow x''$
    $y_i \leftarrow y''/c$
    $W \leftarrow$ span of $W$ and $x_i, y_i$
  **end for**
  **return** $x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n$
**end function**

two orthogonal factors are equidistributed, while the measure on the $A$ (diagonal) factor is given by the product of sinh factors corresponding to the roots (this is an immediate generalization of the fact that the perimeter of a circle of radius $R$ in the hyperbolic plane is $\sinh R$.).

7.4. **Other groups?** In the work by the author [28, 29] and Joseph Maher ([22]) the model of a random element is the random walk model, since this seemed to be the only natural model for the mapping class group. However, in view of the discussion above it makes sense to define the norm of an element $\gamma$ of a mapping class group as the Teichmüller distance from some fixed base surface $S$ to $\gamma(S)$ (one can also use the Weil-Petersson distance, or the distance from a fixed curve to its image in the curve complex, and then pick a random element by analogy with the construction in this note. In fact, this has been done by Joseph Maher in [21].

## References


[1] G.E.P. Box and M.E. Muller, *A note on the generation of random normal deviates*, The Annals of Mathematical Statistics **29** (1958), no. 2, pp. 610–611.

[2] W. Duke, Z. Rudnick, and P. Sarnak, *Density of integer points on affine homogeneous varieties*, Duke Math. J. **71** (1993), no. 1, 143–179, DOI 10.1215/S0012-7094-93-07107-4. MR1230289 (94k:11072)

[3] H. Edelsbrunner, *Geometry and Topology for Mesh Generation*, Cambridge Monographs on Applied and Computational Mathematics, vol. 7, Cambridge University Press, Cambridge, 2001. MR1833977 (2002k:65206)

[4] A. Eskin and C. McMullen, *Mixing, counting, and equidistribution in Lie groups*, Duke Math. J. **71** (1993), no. 1, 181–209, DOI 10.1215/S0012-7094-93-07108-6. MR1230290 (95b:22025)

[5] P. J. Freitas, *On the Action of the Symplectic Group on the Siegel Upper Half Plane*, ProQuest LLC, Ann Arbor, MI, 1999. Thesis (Ph.D.), University of Illinois at Chicago. MR2699568

[6] E. Fuchs and I. Rivin, *How thin is thin*, in preparation, 2012.

[7] N. Gama, N. Howgrave-Graham, and P. Q. Nguyen, *Symplectic lattice reduction and NTRU*, Advances in cryptology—EUROCRYPT 2006, Lecture Notes in Comput. Sci., vol. 4004, Springer, Berlin, 2006, pp. 233–253, DOI 10.1007/11761679_15. MR2423546 (2009f:94042)

[8] J. Gilman, *Two-generator discrete subgroups of* $\mathrm{PSL}(2,\mathbf{R})$, Mem. Amer. Math. Soc. **117** (1995), no. 561, x+204, DOI 10.1090/memo/0561. MR1290281 (97a:20082)
[9] A. Gorodnik and A. Nevo, *Splitting fields of elements in arithmetic groups*, Math. Res. Lett. **18** (2011), no. 6, 1281–1288, DOI 10.4310/MRL.2011.v18.n6.a16. MR2915481
[10] A. Gorodnik, F. Maucourant, and H. Oh, *Manin's and Peyre's conjectures on rational points and adelic mixing* (English, with English and French summaries), Ann. Sci. Éc. Norm. Supér. (4) **41** (2008), no. 3, 383–435. MR2482443 (2010a:14047)
[11] S. Hoory, N. Linial, and A. Wigderson, *Expander graphs and their applications*, Bull. Amer. Math. Soc. (N.S.) **43** (2006), no. 4, 439–561 (electronic), DOI 10.1090/S0273-0979-06-01126-8. MR2247919 (2007h:68055)
[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original. MR1084815 (91i:15001)
[13] M. Kassabov, *Symmetric groups and expander graphs*, Invent. Math. **170** (2007), no. 2, 327–354, DOI 10.1007/s00222-007-0065-y. MR2342639 (2008g:20009)
[14] M. Kassabov, *Universal lattices and unbounded rank expanders*, Invent. Math. **170** (2007), no. 2, 297–326, DOI 10.1007/s00222-007-0064-z. MR2342638 (2009b:20079)
[15] A. W. Knapp, *Representation Theory of Semisimple Groups*: *An Overview Based on Examples*, Princeton Landmarks in Mathematics, Princeton University Press, Princeton, NJ, 2001. Reprint of the 1986 original. MR1880691 (2002k:22011)
[16] A. W. Knapp, *Lie Groups Beyond an Introduction*, 2nd ed., Progress in Mathematics, vol. 140, Birkhäuser Boston, Inc., Boston, MA, 2002. MR1920389 (2003c:22001)
[17] J. C. Lagarias, *Worst-case complexity bounds for algorithms in the theory of integral quadratic forms*, J. Algorithms **1** (1980), no. 2, 142–186, DOI 10.1016/0196-6774(80)90021-8. MR604862 (83e:90112)
[18] A. K. Lenstra, H. W. Lenstra, and L. Lovász. *Factoring polynomials with rational coefficients*, Mathematische Annalen **261** (1982), no. 4, pp. 515–534.
[19] M. W. Liebeck, N. Nikolov, and A. Shalev, *Groups of Lie type as products of* $\mathrm{SL}_2$ *subgroups*, J. Algebra **326** (2011), 201–207, DOI 10.1016/j.jalgebra.2008.12.030. MR2746060 (2011m:20037)
[20] A. Lubotzky, *Finite simple groups of Lie type as expanders*, J. Eur. Math. Soc. (JEMS) **13** (2011), no. 5, 1331–1341, DOI 10.4171/JEMS/282. MR2825166
[21] J. Maher, *Asymptotics for pseudo-Anosov elements in Teichmüller lattices*, Geom. Funct. Anal. **20** (2010), no. 2, 527–544, DOI 10.1007/s00039-010-0064-9. MR2671285 (2011h:37061)
[22] J. Maher, *Random walks on the mapping class group*, Duke Math. J. **156** (2011), no. 3, 429–468, DOI 10.1215/00127094-2010-216. MR2772067 (2012j:37069)
[23] M. Newman, *Counting modular matrices with specified Euclidean norm*, J. Combin. Theory Ser. A **47** (1988), no. 1, 145–149, DOI 10.1016/0097-3165(88)90048-9. MR924457 (89b:15025)
[24] P. Nguyen, *Lattice reduction algorithms: Theory and practice*, Advances in Cryptology–EUROCRYPT 2011, pages 2–6, 2011.
[25] P. Q. Nguyen and D. Stehlé, *Low-dimensional lattice basis reduction revisited*, ACM Trans. Algorithms **5** (2009), no. 4, Art. 46, 48, DOI 10.1145/1597036.1597050. MR2571909 (2011c:68236)
[26] F. Nielsen and R. Nock, *Hyperbolic voronoi diagrams made easy*, Computational Science and Its Applications (ICCSA), 2010 International Conference on, pages 74–80. IEEE, 2010.
[27] A. Page, *Computing arithmetic kleinian groups*, arXiv preprint arXiv:1206.0087, 2012.
[28] I. Rivin, *Walks on groups, counting reducible matrices, polynomials, and surface and free group automorphisms*, Duke Math. J. **142** (2008), no. 2, 353–379, DOI 10.1215/00127094-2008-009. MR2401624 (2009m:20077)
[29] I. Rivin, *Walks on graphs and lattices—effective bounds and applications*, Forum Math. **21** (2009), no. 4, 673–685, DOI 10.1515/FORUM.2009.034. MR2541479 (2010j:20103)
[30] I. Rivin, *Generic phenomena in groups–some answers and many questions*, arXiv preprint arXiv:1211.6509, 2012.
[31] C. L. Siegel, *Symplectic geometry*, Amer. J. Math. **65** (1943), 1–86. MR0008094 (4,242b)
[32] G. W. Stewart, *The efficient generation of random orthogonal matrices with an application to condition estimators*, SIAM J. Numer. Anal. **17** (1980), no. 3, 403–409, DOI 10.1137/0717034. MR581487 (83e:65018)

[33] B. Vallée, A. Vera, et al., *Lattice reduction in two dimensions: analyses under realistic probabilistic models*, Proceedings of the 13th Conference on Analysis of Algorithms, AofA, volume 7, 2007.

[34] B. Vallée, *Gauss' algorithm revisited*, J. Algorithms **12** (1991), no. 4, 556–572, DOI 10.1016/0196-6774(91)90033-U. MR1130316 (93b:11166)

School of Mathematics, University of St Andrews, St Andrews, UK
*Current address*: Department of Mathematics, Temple University, Philadelphia, Pennsylvania
*E-mail address*: rivin@temple.edu