Generating uniform unit random vectors in $\mathbb{R}^n$

Andersen Ang

Mathématique et recherche opérationnelle
UMONS, Belgium

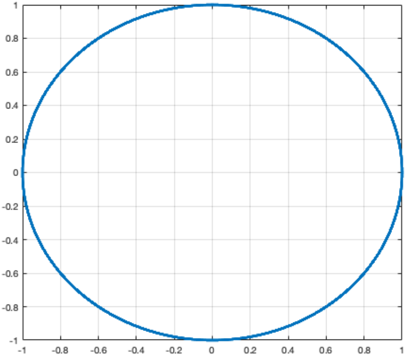
manshun.ang@umons.ac.be Homepage: angms.science

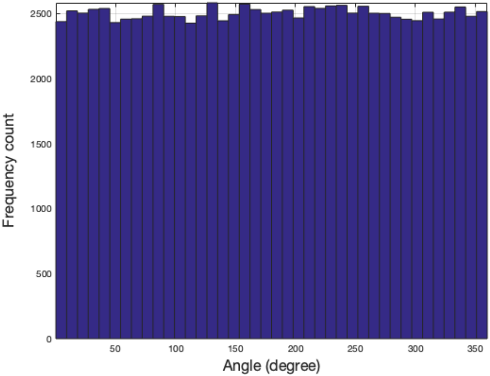
First draft : December 24, 2018
Last update : June 21, 2019

Generating uniform distributed random unit vector in $\mathbb{R}^2$

Generating a random unit vector $\mathbf{x}$ in $\mathbb{R}^2$ is the same as randomly picking a point on the circumference of a circle centred at the origin.

With the expression $\mathbf{x} = [\cos\theta \;\; \sin\theta]$, we can sample $\theta \sim \mathcal{U}[0, 2\pi]$, where $\mathcal{U}$ stands for uniform distribution.

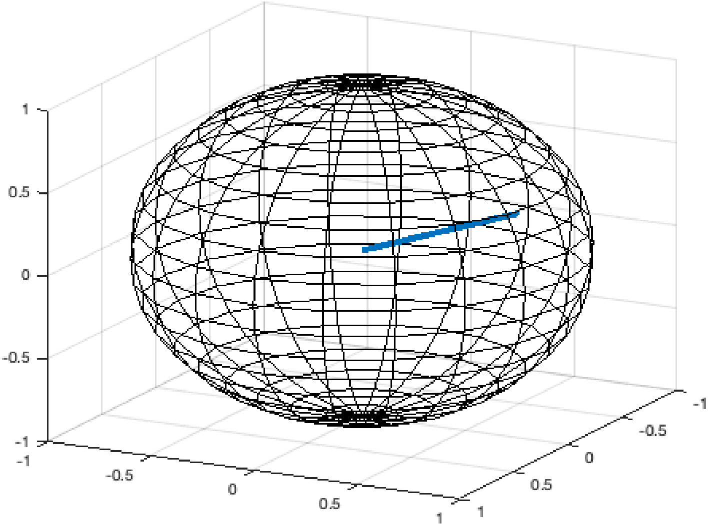




The points are uniformly distributed on the circumference.

Random unit vector in $\mathbb{R}^3$

In $\mathbb{R}^3$, the same idea holds : generating a unit vector in $\mathbb{R}^3$ is the same as randomly picking a point on the surface of a 2-sphere[1].



[1] A n-shpere is defined as $S^n = \{\mathbf{x} \in \mathbb{R}^{n+1} \mid \|\mathbf{x}\| = r\}$.

Expression of points on the surface of 2-sphere

The point on the surface of unit 2-sphere in $\mathbb{R}^3$ can be expressed as

$$\mathbf{x} = [\sin\phi\cos\theta \quad \sin\phi\sin\theta \quad \cos\phi],$$

where

- θ is the azimuthal angle ranged in $[0\ 2\pi]$
- ϕ is the polar angle ranged in $[0\ \pi]$
- for unit 2-shpere the radius $r = 1$

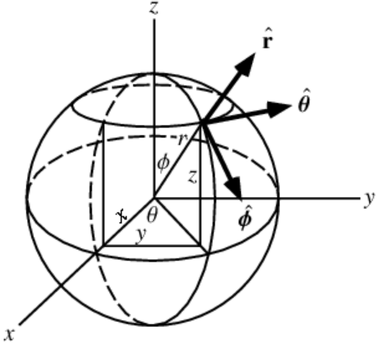
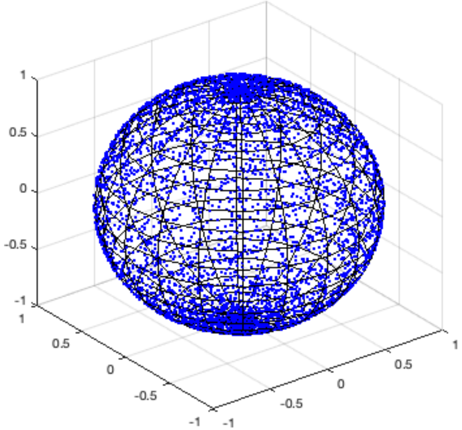


Image from : wolfram.com/SphericalCoordinates.html

Wrong way to generate uniform random unit vector in $\mathbb{R}^3$

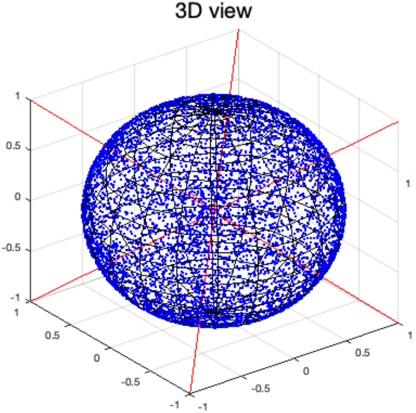
As $\mathbf{x} = [\sin\phi\cos\theta \quad \sin\phi\sin\theta \quad \cos\phi]$, $\theta \in [0, 2\pi], \phi \in [0, \pi]$. At first glance, to generate $\mathbf{x}$, one may pick $\phi \sim \mathcal{U}[0, \pi]$ and $\theta \sim \mathcal{U}[0, 2\pi]$.
This method does not work : the resulting points will not be uniform on the surface of the sphere, there will be more points at the two poles.

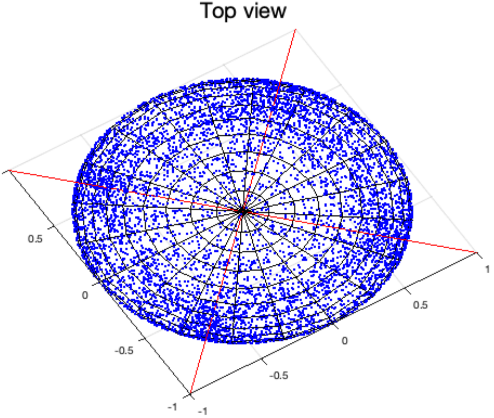


See also http://mathworld.wolfram.com/SpherePointPicking.html.

Wrong way to generate uniform random unit vector in $\mathbb{R}^3$

One may also generate $\mathbf{x}$ by the *cube method* : randomly pick a point in $[-1, 1]^3$, then normalize it to have unit norm.
This method does not work : there will be more points on the diagonals.





Interesting fact : for $\mathbf{x} = [x\,y\,z]$, if $\mathbf{x}$ is uniformly generated on the surface of the unit sphere, then (x, y, z) are all uniform in $[-1, 1]$, but the converse here is not true.

The modified cube method in $\mathbb{R}^3$ using rejection sampling

We can modify the cube method to generate $\mathbf{x}$:

1. Generate 3 random number $x, y, z \sim \mathcal{U}[-1, 1]$ (the cube method)
2. What's new : if $x^2 + y^2 + z^2 \leq 1$, let $\mathbf{x} = \dfrac{[x \quad y \quad z]}{\sqrt{x^2 + y^2 + z^2}}$.

   Otherwise, reject the point and re-sample again.

The distribution of $\mathbf{x}$ would be uniform on surface of the unit 2-sphere.

Why it works : only the points inside of unit ball are normalized, the points outside the sphere are rejected. This technique is called *Rejection sampling*.

Drawback of this approach : inefficient/slow, which is the fundamental disadvantage of all methods that use rejection sampling.

Why the modified cube method is inefficient

There are 2 step to generate $\mathbf{x}$:

1. Generate random number $x, y, z \sim \mathcal{U}[-1, 1]$
2. Rejection : if $x^2 + y^2 + z^2 > 1$, re-sample again

Rejecting sampled point means some computer resources are wasted to generate a useless point. That is, this method requires some random number generations before $x^2 + y^2 + z^2 \leq 1$ is true.

In fact, step 1 has to be run on average 2 to 3 time to generate a feasible point :

$$\mathbb{P}(x^2 + y^2 + z^2 \leq 1) = \frac{\text{Volume of the sphere}}{\text{Volume of the cube}} = \frac{\frac{4\pi r^3}{3}}{(2r)^3} = \frac{\pi}{6} \in \left[\frac{1}{3}, \frac{1}{2}\right].$$

The modified cube method is very inefficient in high dimension

The modified cube method works even worse when dimensions n is large. As the volume of unit ball shrinks fast in high dimension, it will require many random number generations before $x_1^2 + x_2^2 + \cdots + x_n^2 \leq 1$ is true.

For example, when $n = 8$, we have volume of the unit 7-sphere as $4.06r^8$. Then

$$\mathbb{P}(x_1^2 + \cdots + x_n^2 \leq 1) \quad = \quad \frac{4.06r^8}{(2r)^8} \approx \frac{1}{2^6} = \frac{1}{64}.$$

It takes roughly 64 random generations to form a feasible point.

If $n = 11$:

$$\mathbb{P}(x_1^2 + \cdots + x_n^2 \leq 1) \quad = \quad \frac{1.884r^{11}}{(2r)^{11}} \approx \frac{1}{2^{10}} = \frac{1}{1024}.$$

It takes about 1000 generations to form a feasible point.

Efficient and simple way to pick random unit vector in $\mathbb{R}^3$

The following is an efficient and simple way to generate $\mathbf{x}$ by using Gaussian random variable[2] : generate 3 independent Gaussian random variables $x, y, z \sim \mathcal{N}(0,1)$. Then the distribution of

$$\mathbf{x} = \frac{[x \quad y \quad z]}{\sqrt{x^2 + y^2 + z^2}}$$

will be uniform over the surface of the sphere.

This works as multivariate normal distribution is *spherical*

- Further more, multivariate normal distribution is *symmetric* : it is invariant under rotation, so such approach is not the same as the cube method, points will not be concentrated on the diagonals
- Also, this method works for any dimension n

---

[2] M. E. Muller "A Note on a Method for Generating Points Uniformly on N-Dimensional Spheres." 1959.

Other ways to pick random unit vector in $\mathbb{R}^3$

There are other (more complicated) methods to generate $\mathbf{x}$ in $\mathbb{R}^3$:

Using equal-area projection of sphere onto rectangle surface of the bounding cylinder, we have

$$\mathbf{x} = \left[\sqrt{1 - z^2} \cos\theta \quad \sqrt{1 - z^2} \sin\theta \quad z \right],$$

where $z \sim \mathcal{U}[-1, 1]$, $\theta \sim \mathcal{U}[0, 2\pi]$ and here $z = \cos\phi$ for the azimuthal angle ϕ.

On spherical coordinate, we can set

$$\theta \sim \mathcal{U}[0, 2\pi], \quad \phi = \cos^{-1} a,$$

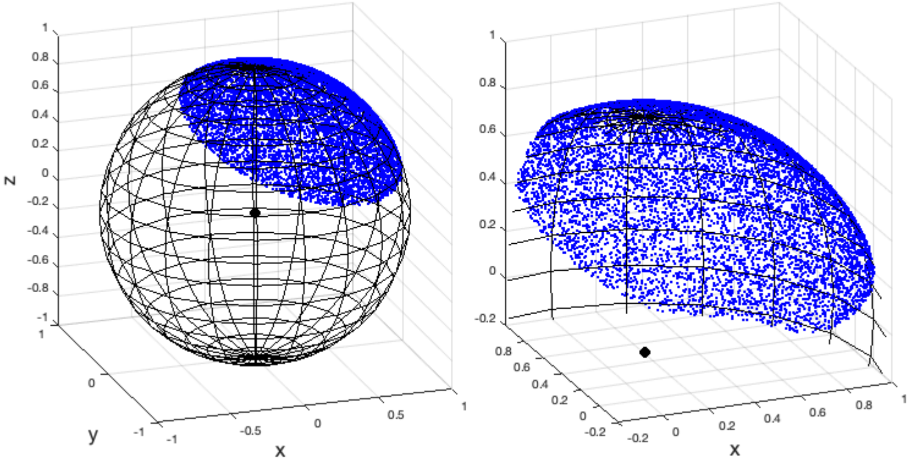
where $a \sim \mathcal{U}[-1, 1]$. Then we set $\mathbf{x} = [\sin\phi \cos\theta \;\; \sin\phi \sin\theta \;\; \cos\phi]$. Note that $\phi = \cos^{-1} a, a \sim \mathcal{U}[-1, 1]$ is different from directly picking $\phi \sim \mathcal{U}[0, \pi]$.

Generating uniform random unit vector in $\mathbb{R}^n$

To generate unit vector $\mathbf{x} = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n$ such that $\mathbf{x}$ is uniformly distributed on the surface of the unit $(n-1)$-sphere :

1. Generate n i.i.d. Gaussian random variable : $x_i \sim \mathcal{N}(0, 1)$.
2. Form $\mathbf{x} = [x_1, x_2, \ldots, x_n]$.
3. Normalize $\mathbf{x}$ to have unit l_2 norm.

"Generating random unit vector in $\mathbb{R}^n$ inside a cone" is the same as "picking a point on the partial surface of a $(n-1)$-sphere defined by the solid angle of the cone".



Points on the surface of 2-sphere inside a cone

The points on the surface of 2-sphere are

$$\mathbf{x} = [\sin\phi\cos\theta \quad \sin\phi\sin\theta \quad \cos\phi],$$

where

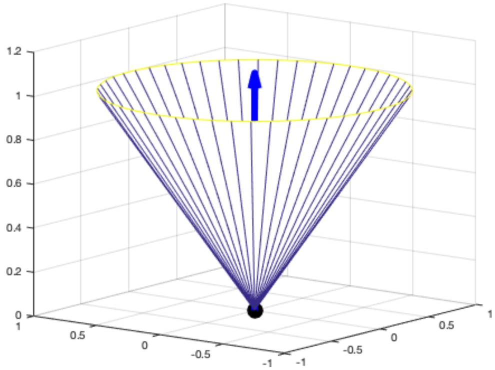
- θ is the azimuthal angle ranged in $[\theta_1, \theta_2] \subseteq [0, 2\pi]$
- ϕ is the polar angle ranged in $[\phi_1, \phi_2] \subseteq [0, \pi]$
- $\theta_1, \theta_2, \phi_1, \phi_2$ are all unknown value yet to be determined based on the information on the circular cone C

The characterization of circular cone $C(\mathbf{p}, \mathbf{d}, \psi)$

A circular cone C is completely determined by three parameters :

- the origin of the cone $\mathbf{p} \in \mathbb{R}^n$
- the direction of the cone $\mathbf{d} \in \mathbb{R}^n$
- the side angle of the cone $\psi \in [0, \pi]$

An example in $\mathbb{R}^3$: $\mathbf{p} = [0, 0, 0]$ (the black dot), $\mathbf{d} = [0, 0, 1]$ (the blue arrow), $\psi = \dfrac{\pi}{4}$ (the angles between all the line on the yellow circle and $\mathbf{d}$ are all equal to $45°$)



Unit vector inside a cone

Now consider $n = 3$ and $\mathbf{p} = [0, 0, 0]$. (If the cone is not centred at origin, we can just translate it by adding $\mathbf{p}$). We are given a cone $C(\mathbf{d}, \psi)$, we want to generate unit vector such that these vectors are uniformly distributed inside the cone C.

How to generate these vectors:

$$\mathbf{x} = \left[\sqrt{1 - z^2}\cos\theta \quad \sqrt{1 - z^2}\sin\theta \quad z\right],$$

where $\theta \sim \mathcal{U}[0, 2\pi]$ and $z \sim \mathcal{U}[\cos\phi, 1]$, the polar angle ϕ is obtained by the z-component of the direction vector $\mathbf{d}$ of the cone C, that is :

$$\phi = \cos^{-1}\langle[0, 0, 1], \hat{\mathbf{d}}\rangle = \cos^{-1}\left(\frac{\mathbf{d}_z}{\|\mathbf{d}\|_2}\right).$$

Last page - summary

- Generating random unit vector in $\mathbb{R}^n$ is the same as picking a point randomly on the surface of a unit $(n-1)$-sphere
- Wrong ways to generating random unit vector in $\mathbb{R}^n$
- Inefficient rejection sampling based method to generating random unit vector in $\mathbb{R}^n$
- Efficient way to generating random unit vector in $\mathbb{R}^n$
- A way to generating random unit vector within a cone in $\mathbb{R}^3$

Not discussed : how to generate random unit vector within a cone in $\mathbb{R}^n$

End of document