UCSB, Physics 129L, Computational Physics
Lecture notes, Week 7

Zihang Wang (UCSB), zihangwang@ucsb.edu

February 22, 2025

Contents

1 Random number generation **1**

1.1 Pseudo-Random Number Generators: Linear congruential generator . . . 2

1.2 Quasi-Random Number Generators, Sobol sequence . . . 2

2 Distribution Sampling **5**

2.1 Inverse Transform Sampling . . . 5

2.2 Box–Muller transform . . . 6

2.3 Rejection Sampling . . . 10

3 Monte Carlo Methods **11**

3.1 Monte Carlo integration . . . 11

3.2 Importance Sampling and regularization . . . 13

1 Random number generation

Random number generation (RNG) is the process of producing a sequence of numbers that lack any predictable pattern. There are three main types of random number generators:

- **True Random Number Generators**: These rely on physical phenomena, such as electronic noise or radioactive decay, to produce truly unpredictable numbers.
- **Pseudo-Random Number Generators**: These use deterministic algorithms to generate sequences that **appear random** but are reproducible if the initial seed value is known.
- **Quasi-Random Number Generators**: Quasi-random numbers are low-discrepancy sequences that are **not statistically random**, designed to cover space evenly rather than appearing random.

While on a computer, generating truly random numbers using deterministic algorithms is not possible, pseudo-random numbers can be generated, which appear random for practical purposes.

1.1 Pseudo-Random Number Generators: Linear congruential generator

A **Linear Congruential Generator (LCG)** is a simple and widely used method for generating pseudo-random numbers. It follows the recursive formula:

$$X_{n+1} = (aX_n + c) \mod m, \quad X \in \{0, 1, \cdots, m-1\}. \tag{1}$$

where X_n is the sequence of pseudo-random numbers, a is the multiplier, c is the increment, m is the modulus, X_0 is the seed (initial value). To obtain a **uniform random number** in the range $[0, 1)$, we use the following,

$$U_n = \frac{X_n}{m}. \tag{2}$$

For the generator to have a full period (cover all possible values before repeating), the following conditions should be met,

1. c and m must be relatively prime.
2. $a - 1$ must be a multiple of all prime factors of m.
3. $a - 1$ must be a multiple of 4 if m is a multiple of 4.

A well-known choice of parameters ism

$$a = 1664525, \quad c = 1013904223, \quad m = 2^{32}. \tag{3}$$

1.2 Quasi-Random Number Generators, Sobol sequence

A Sobol sequence is a specific kind of **quasi-random** sequence, which is constructed in such a way that its points are evenly distributed across the unit cube $[0, 1]^d$ for d-dimensional space. These sequences are typically generated by iteratively constructing binary expansions of each coordinate.

Let us define a Sobol sequence in d dimensions as the sequence of points $\mathbf{x}_n = (x_{n,1}, x_{n,2}, \ldots, x_{n,d}) \in [0, 1]^d$. Each coordinate component $x_{n,d}$ of the n-th point in the sequence is generated using a binary expansion. Specifically, the binary expansion of the d-th coordinate of the n-th point is given by the weighted binary sum,

$$x_{n,d} = \sum_{k=1}^{n} b_k \, v_{d,k}, \tag{4}$$

where the index (upto m bit) is given in the binary form $n = b_m b_{m-1} \cdots b_1$, and $v_{d,k}$ is the k^{th} **direction number**,

$$v_{d,k} = \frac{m_k}{2^k}, \tag{5}$$

where m_k is a general coefficient.

For each dimension $d \geq 2$, we choose a **primitive polynomial**, where the coefficients are either one or zero. Those coefficients will be used to construct the directional number. For example, for dimension 2, we might use,

$$p_d(x) = x^3 + a_1x^2 + a_2x + 1 = x^3 + 0 \cdot x^2 + x + 1, \tag{6}$$

where in our case the coefficients are ($a_1 = 0$ and $a_2 = 1$ (the constant term is always 1 by construction).

We first choose initial values for the sequence $m_1, m_2, \ldots, m_s$ (where s is the degree of the polynomial). These are typically odd integers with $m_k < 2^k$. For $\mathbf{k} > \mathbf{s}$, the polynomial provides a recurrence to compute m_k. It uses s number of previous points. A common recurrence used in Sobol sequence construction is the weighted binary sum,

$$m_k = a_1m_{k-1} \oplus a_2m_{k-2} \oplus \cdots \oplus a_{s-1}m_{k-s+1} \oplus (m_{k-s} \ll s), \tag{7}$$

where, $\oplus$ denotes binary addition (bitwise XOR) such that all calculation must be written in bit addition. $\ll s$ denotes a left-shift by s bits, which is equivalent as the multiplication of 2^s. The coefficients a_j come from the polynomial $p(x)$. In calculation, $v_{2,k}$ will be in the binary form.

Assume we have chose the initial sequence,

$$m_1 = 1, \quad m_2 = 3, \quad m_3 = 5, \tag{8}$$

$$v_{2,1} = \frac{1}{2}, \quad v_{2,2} = \frac{3}{4}, \quad v_{2,3} = \frac{5}{8}, \tag{9}$$

with $s = 3$ for $p(x) = x^3 + x + 1$ (where $a_1 = 0, a_2 = 1$). For $k = 4$, the recurrence would be,

$$m_4 = a_1m_3 \oplus a_2m_2 \oplus (m_1 \ll 3). \tag{10}$$

Substituting the values,

$$m_4 = 0 \cdot m_3 \oplus 1 \cdot m_2 \oplus (1 \ll 3). \tag{11}$$

Here, $1 \ll 3$ means shifting 1 left by 3 bits, which is 8 (in decimal). So,

$$m_4 = m_2 \oplus 8 = 3 \oplus 8. \tag{12}$$

In binary, we have $3 = 0011,\ 8 = 1000,\ 3\ \oplus\ 8\ =\ 1011$ (11 in decimal).

Thus, the direction number is,

$$v_{2,4} = \frac{m_4}{2^4} = \frac{11}{16}. \tag{13}$$

The index n can be written in the binary form, e.g. $n = 3 \rightarrow b_3b_2b_1 = 011$ and $n = 5 \rightarrow b_3b_2b_1 = 101$ (recall that in binary form, the integer has the opposite order as those in fractions after the decimal point, i.e. $1/2 \to 100, 1/4 = 010, 3/4 = 110$. The value, $x_{n,2}$, is given by the binary product,

$$x_{3,2} = b_1 v_{2,1} \oplus b_2 v_{2,2} \oplus b_3 v_{2,3} = 1 \cdot \frac{1}{2} \oplus 1 \cdot \frac{3}{4} \oplus 0 \cdot \frac{5}{8} = 100 \oplus 110 \oplus 000 = 010 = \frac{1}{4}. \tag{14}$$

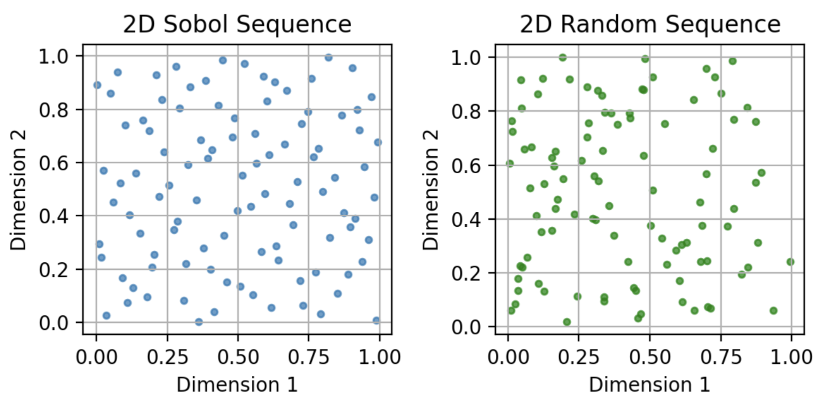
$$x_{5,2} = b_1 v_{2,1} \oplus b_2 v_{2,2} \oplus b_3 v_{2,3} = 1 \cdot \frac{1}{2} \oplus 0 \cdot \frac{3}{4} \oplus 1 \cdot \frac{5}{8} = 100 \oplus 000 \oplus 101 = 001 = \frac{1}{8}. \tag{15}$$

If we make it larger, $n = 9 \to b_4 b_3 b_2 b_1 = 1001$,

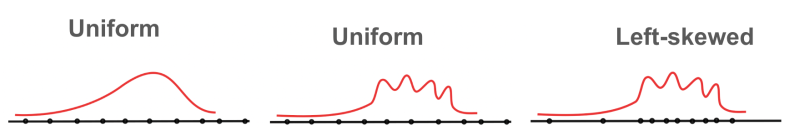
$$\begin{aligned} x_{9,2} &= b_1 v_{2,1} \oplus b_2 v_{2,2} \oplus b_3 v_{2,3} \oplus b_4 v_{2,4} = 1 \cdot \frac{1}{2} \oplus 0 \cdot \frac{3}{4} \oplus 0 \cdot \frac{5}{8} \oplus 1 \cdot \frac{11}{16} \\ &= 1000 \oplus 0000 \oplus 0000 \oplus 1011 = 0011 = \frac{3}{16}. \end{aligned} \tag{16}$$

The Sobol sequence divide the space in a binary way, such that it spreads out points to the uniform hyper-cube.

In the figure below, we can see Sobol sequence has a better coverage than a pseudo random number,



Random generator is critical not only for mesh generation (or grid), but serving as the first step in Monte-Carlo methods for generating non-uniform grids. For example, as shown in the figure below, we need to consider the variation of the function itself when preforming sampling in integration. Direct uniform sampling may not be optimal.



2 Distribution Sampling

2.1 Inverse Transform Sampling

Let x be a continuous random variable with cumulative distribution function (CDF) $F(x)$, i.e.,

$$y = F(x) = P(X \leq x). \tag{17}$$

If we sample the random variable y from a uniform distribution from 0 to 1, $y \sim \mathrm{U}(0,1)$, and assume F is invertible, we rewrite the random variable x as the following,

$$x = F^{-1}(y). \tag{18}$$

The probability density function (PDF) is given by the derivative of the CDF,

$$\frac{dy}{dx} = \frac{d}{dx}F(x) = f(x). \tag{19}$$

Since x, y have the same CDF, and we want to find $f(x)$ in terms of $U(y)$, we insert a differential,

$$f(x)\frac{dx}{dy}dy = U(y)dy, \tag{20}$$

where $U(y)$ is the PDF of y, i.e. the uniform distribution from 0 to 1. If no singular values, two sides

$$f(x) = U(y)\left|\frac{dy}{dx}\right| = U(y)\left|\frac{dx}{dy}\right|^{-1}. \tag{21}$$

This idea is illustrated in the figure below.

Let's consider an example, We are given that the CDF,

$$F(x) = 1 - e^{-\lambda x}, \quad x \geq 0 \tag{22}$$

The inverse CDF is,

$$x = F^{-1}(y) = -\frac{\ln(1-y)}{\lambda} \tag{23}$$

where y is uniformly distributed in $[0, 1]$. In this way, we are able to generate samples of x, that follows the PDF of $F(x)$. To see x follows the desire PDF, we can first calculate the following derivative,

$$\frac{dx}{dy} = \frac{1}{\lambda}\frac{1}{1-y}. \tag{24}$$

Since we know, $y = F(x) = 1 - e^{-\lambda x}$, we substitute this into the formula,

$$\frac{dx}{dy} = \frac{1}{\lambda}\frac{1}{e^{-\lambda x}} \tag{25}$$

Thus, the probability density function of X is,

$$f(x) = U(y) \left| \frac{dx}{dy} \right|^{-1} = \frac{dy}{dx} = \lambda e^{-\lambda x}, \quad x \geq 0. \tag{26}$$

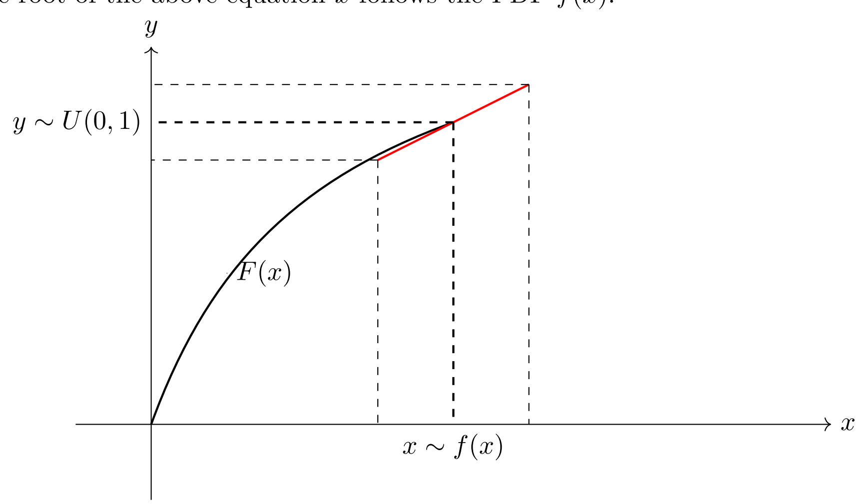
As you can see, by assuming a uniform distribution, we are able to generate random variable that follows a particular PDF via its analytical CDF. Since CDF is a monotonically increasing function, we are able to integrate PDF numerically and approximate the inverse. The **general inverse transform sampling** can be used without an explicit inverse CDF. Let's approximate the CDF F numerically via the PDF $f(x)$,

$$F(x) = \int_{-\infty}^{x} f(x')\, dx', \tag{27}$$

and, for a uniform random variable $y \sim U(0, 1)$, we solve the following equitation,

$$g(x) = F(x) - y = 0. \tag{28}$$

The root of the above equation x follows the PDF $f(x)$.



2.2 Box–Muller transform

The **Box–Muller transform** uses the relationship between uniform distributions and polar coordinates. Given two uniform random variables u_1 and u_2 $u_1, u_2 \sim U(0, 1)$, it computes two normally distributed variables z_0 and z_1 using the equations,

$$z_0 = \sqrt{-2\ln(u_1)}\cos(2\pi u_2), \quad z_1 = \sqrt{-2\ln(u_1)}\sin(2\pi u_2) \tag{29}$$

These outputs are **independent** and follow a **standard normal distribution**, $z_1, z_2 \sim \mathcal{N}(0, 1)$.

Consider the joint PDF of two independent standard normal variables z_1 and z_2,

$$f(z_1, z_2) = \frac{1}{2\pi} e^{-\frac{z_1^2+z_2^2}{2}}. \tag{30}$$

This density is radially symmetric, making it natural to switch to polar coordinates,

$$r = \sqrt{z_1^2 + z_2^2}, \quad \phi = \arctan\left(\frac{z_2}{z_1}\right). \tag{31}$$

In polar coordinates, the joint PDF is separable,

$$f(r, \theta) = R(r)\Phi(\phi), \tag{32}$$

where $\Phi(\phi) = \frac{1}{2\pi}$ and $R(r) = re^{-r^2/2}$ for $r > 0$.

To generate samples from $\Phi(\phi) = \frac{1}{2\pi}$, we use the **inverse transform sampling** for the CDF,

$$u_2 = F_\Phi(\phi) = \frac{\phi}{2\pi}, \quad \to \quad \phi = 2\pi u_2, \tag{33}$$

where $u_2 \sim U(0, 1)$.

Similarly, to sample from $R(r) = re^{-r^2/2}$, note that its CDF is,

$$u_1 = F_R(r) = 1 - e^{-r^2/2}. \tag{34}$$

we use the **inverse transform sampling**, we get,

$$r = \sqrt{-2\ln(1 - u_1)}, \tag{35}$$

where $u_1 \sim U(0, 1)$.

Using the sampled values of r and ϕ, convert back to Cartesian coordinates,

$$z_1 = r\cos(\phi), \quad z_2 = r\sin(\phi). \tag{36}$$

Substituting the expressions for r and ϕ,

$$z_1 = \sqrt{-2\ln(u_1)}\cos(2\pi u_2), \quad z_2 = \sqrt{-2\ln(u_1)}\sin(2\pi u_2). \tag{37}$$

Since u_1, u_2 are independent and uniformly distributed, the transformations preserve independence. The resulting variables z_1, z_2 are independent standard normal random variables because they follow the desired joint PDF.

Using

$$z_1 = r\cos\phi, \tag{38}$$

we have,

$$z_1 = -2\ln(u_1)\cos(2\pi u_2). \tag{39}$$

To find the PDF of $z = z_1$, we integrate over all possible values of ϕ and radius while treating z as an independent random variable,

$$f(z) = \int_0^\infty \int_0^{2\pi} f(r, \phi)\delta(z - r\cos\phi)\, d\phi\, dr. \tag{40}$$

We can look at the first integral in ϕ. The Dirac delta function $\delta(z - r\cos\phi)$ selects the values of ϕ such that $z = r\cos\phi$. We should note that z, ϕ, r are independent from each other.

This equation has two roots in $[0, 2\pi)$,

$$\phi_1 = \arccos\left(\frac{z}{r}\right), \quad \phi_2 = 2\pi - \arccos\left(\frac{z}{r}\right). \tag{41}$$

Using the property of the Dirac delta function under a change of variables,

$$\delta(g(\phi)) = \sum_i \frac{\delta(\phi - \phi_i)}{|\partial_\phi g|_{\phi_i}|}, \tag{42}$$

where $g(\phi) = z - r\cos\phi$ and $\partial_\phi g|_{\phi_i} = r(-\sin\phi_{\phi_i})$, we compute,

$$|g'(\phi_1)| = r|\sin(\phi_1)| = r\sqrt{1 - \cos^2(\phi_1)} = r\sqrt{1 - (z/r)^2} = \sqrt{r^2 - z^2}. \tag{43}$$

The same holds for ϕ_2, and we have,

$$\int_0^{2\pi} \delta(z - r\cos\phi)d\phi = \frac{1}{|g'(\phi_1)|} + \frac{1}{|g'(\phi_2)|} = \frac{2}{\sqrt{r^2 - z^2}}, \tag{44}$$

and this is valid only when $r \geq |z|$.

Substituting this result back into the integral for $f(z)$,

$$f(z) = \int_{|z|}^{\infty} \frac{1}{2\pi} r e^{-r^2/2} \frac{2}{\sqrt{r^2 - z^2}} dr = \frac{1}{\pi} \int_{|z|}^{\infty} \frac{r e^{-r^2/2}}{\sqrt{r^2 - z^2}} dr. \tag{45}$$

To simplify the square root in the denominator, we perform the substitution,

$$r = \sqrt{z^2 + s^2}, \quad dr = \frac{s}{\sqrt{z^2 + s^2}} ds. \tag{46}$$

Substituting these into the integral for $f(z)$,

$$f(z) = \frac{1}{\pi} \int_0^{\infty} e^{-(z^2+s^2)/2} \frac{\sqrt{z^2 + s^2}}{\sqrt{s^2}} \left(\frac{s}{\sqrt{z^2 + s^2}}\right) ds. \tag{47}$$

The term $\sqrt{s^2}/s$ cancels out,

$$f(z) = e^{-z^2/2} \cdot \frac{1}{\pi} \int_0^{\infty} e^{-s^2/2} ds = e^{-z^2/2} \cdot \frac{1}{\pi} \cdot \sqrt{\frac{\pi}{2}} = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}. \tag{48}$$

This is the standard normal distribution.

In general, **Box–Muller transform** can be adapted to **sample correlated normal random variables**, z_1 and z_2. To see this, consider two independent standard normal random variables v_1 and v_2, we define the transformation,

$$z_1 = v_1, \quad z_2 = \rho v_1 + \sqrt{1 - \rho^2} v_2, \tag{49}$$

where $-1 \leq \rho \leq 1$ is a correlation coefficient. The joint PDF of v_1 and v_2 is,

$$f(v_1, v_2) = \frac{1}{2\pi} e^{-\frac{v_1^2+v_2^2}{2}}. \tag{50}$$

To find the joint PDF of z_1 and z_2, we use the change-of-variables formula. The Jacobian matrix of the transformation is,

$$J = \begin{bmatrix} \frac{\partial z_1}{\partial v_1} & \frac{\partial z_1}{\partial v_2} \\ \frac{\partial z_2}{\partial v_1} & \frac{\partial z_2}{\partial v_2} \end{bmatrix}. \tag{51}$$

Substituting the expressions for $z_1 = v_1$ and $z_2 = \rho v_1 + \sqrt{1-\rho^2} v_2$, we compute,

$$J = \begin{bmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{bmatrix}. \tag{52}$$

The determinant of this matrix is,

$$|J| = 1 \cdot \sqrt{1-\rho^2} - 0 \cdot \rho = \sqrt{1-\rho^2}. \tag{53}$$

Using the change-of-variables formula, the joint PDF of z_1 and z_2 is,

$$f(z_1, z_2) = f(v_1, v_2) \cdot |J|^{-1}. \tag{54}$$

Substituting $f(v_1, v_2)$, we have,

$$f(z_1, z_2) = \frac{1}{2\pi} e^{-\frac{v_1^2+v_2^2}{2}} \cdot \frac{1}{|J|}. \tag{55}$$

Since the transformation is invertible, we can express $v_1 = z_1$ and $v_2 = (z_2 - \rho z_1)/\sqrt{1-\rho^2}$. Substituting these into the exponent gives,

$$v_1^2 + v_2^2 = z_1^2 + \left(\frac{z_2 - \rho z_1}{\sqrt{1-\rho^2}} \right)^2. \tag{56}$$

We expand and simplify the terms as follows,

$$z_1^2 + \frac{\rho^2 z_1^2}{1-\rho^2} = z_1^2 \left(1 + \frac{\rho^2}{1-\rho^2} \right) = z_1^2 \cdot \frac{1-\rho^2+\rho^2}{1-\rho^2} = \frac{z_1^2}{1-\rho^2}. \tag{57}$$

Thus, the exponent becomes,

$$-\frac{1}{2} \left[\frac{z_1^2}{1-\rho^2} + \frac{z_2^2}{1-\rho^2} - \frac{2\rho z_1 z_2}{1-\rho^2} \right] = -\frac{1}{2(1-\rho^2)} \left[z_1^2 + z_2^2 - 2\rho z_1 z_2 \right]. \tag{58}$$

The term inside the brackets, $z_1^2 + z_2^2 - 2\rho z_1 z_2$, can be written as a quadratic form,

$$[z_1, z_2] \begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}. \tag{59}$$

Thus, the joint PDF becomes,

$$f(z_1, z_2) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}[z_1, z_2]\begin{bmatrix}1 & -\rho\\ -\rho & 1\end{bmatrix}\begin{bmatrix}z_1\\ z_2\end{bmatrix}\right). \tag{60}$$

We see that the covariance matrix and its inverse,

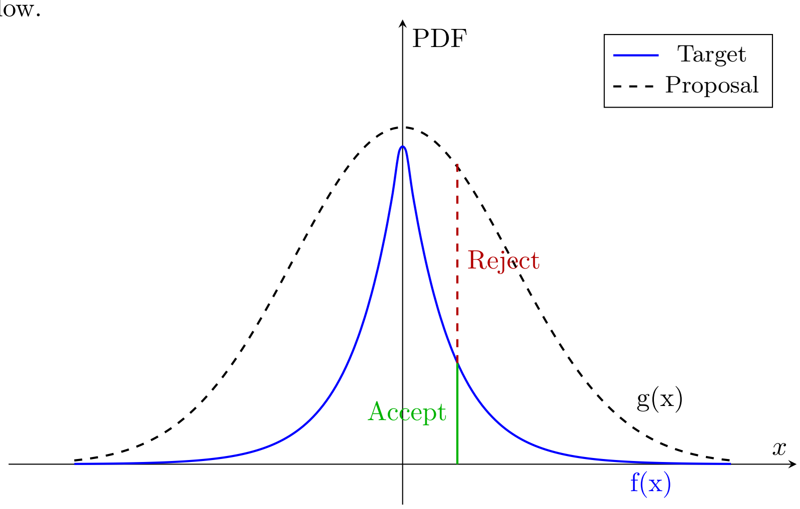
$$\Sigma = \begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix}, \quad \Sigma^{-1} = \frac{1}{1-\rho^2}\begin{bmatrix}1 & -\rho\\ -\rho & 1\end{bmatrix}. \tag{61}$$

And finally, we have,

$$f(z_1, z_2) = \frac{1}{2\pi\sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}[z_1, z_2](\Sigma^{-1})\begin{bmatrix}z_1\\ z_2\end{bmatrix}\right). \tag{62}$$

2.3 Rejection Sampling

Let's say we want to generate data points that follow a particular complicated PDF $f(x)$. For example, we know how to generate the PDF $g(x)$ and CDF $G(x)$ of distribution. We use it as the **proposal function**, and we want to get the samples from a complicated PDF (Target function, $f(x)$), showing in the figure below.



Let's start with a given x, and calculate the ratio between two PDF at that point,

$$R(x) = \frac{f(x)}{Mg(x)} \le 1, \tag{63}$$

where M is a scaling constant. We must ensure that **the proposal function encloses the target function**. Then, using the inverse sampling, we use the proposal function to generate a random variable,

$$x = G^{-1}(u_1), \tag{64}$$

where $u_1 \sim U(0,1)$ and G is the CDF of the proposal function g. For generating a We then sample a different random variable $u_2 \sim U(0,1)$, if $u_2 \leq R$, accept the sample. Otherwise, we reject the sample. In other words, the sample drawn from the rejection sampling has a binary form,

$$x = \begin{cases} F^{-1}(u_1), & u_2 \leq R \\ G^{-1}(u_1), & \text{else} \end{cases}. \tag{65}$$

While we use $u_2 \sim U(0,1)$, in general, u_2 can follow a different distribution and the rejection criteria could be a function of the ratio R. This process ensures that we generate samples from the target distribution $g(x)$ by using a simpler proposal distribution $f(x)$.

3 Monte Carlo Methods

Monte Carlo methods are a class of computational algorithms that rely on **random sampling**.

3.1 Monte Carlo integration

Monte Carlo integration is a numerical method used to approximate definite integrals, particularly in high-dimensional spaces where traditional methods (like **Riemann sums or Gaussian quadrature**, which we will be talking in the next lecture) become inefficient. For example, if a CDF does not have any analytical form, we can preform Monte Carlo integration via a PDF distribution.

Instead of summing over small slices (like in traditional numerical integration), Monte Carlo integration **estimates the integral using randomly sampled points**. Given an integral of the form in one dimension,

$$I = \int_a^b f(x)\,dx = \int_a^b \frac{f(x)}{q(x)} q(x)\,dx = \int_a^b w(x)q(x)\,dx, \tag{66}$$

Monte Carlo integration is based on the idea of estimating an expected value. If we define the **uniform PDF** over $[a,b]$ as,

$$q(x) = \frac{1}{b-a}, \quad \text{for } x \in [a,b] \tag{67}$$

we can rewrite the integral as,

$$I = \int f(x)\,dx = \int f(x)\frac{b-a}{b-a}\,dx = (b-a)\int f(x)q(x)\,dx \tag{68}$$

and we note that the last equality is the expectation value, which is just,

$$I = (b-a)\mathbb{E}[f(x)] \tag{69}$$

where $\mathbb{E}[f(x)]$ is the expected value of $f(x)$ when x is sampled from the uniform distribution.

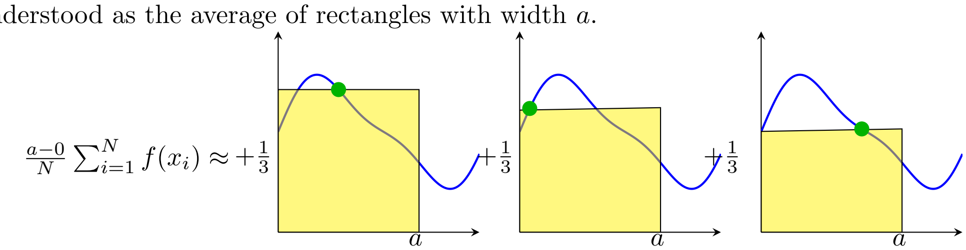
Monte Carlo sampling estimates the expected value using the sample mean, and we approximate,

$$\mathbb{E}[f(x)] = \int f(x)q(x)\,dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{70}$$

Thus, the Monte Carlo estimate for I is,

$$I \approx (b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i), \quad x_i \sim q(x) = U(0, b-a). \tag{71}$$

where x_i are now **random variable drawn from the uniform distribution** $x_i \sim q(x) = U(0, b-a)$. The accuracy improves as N increases, following the law of large numbers. The $\frac{1}{N}$ appears because we approximate the expectation using the sample mean. As illustrated in the figure below, the sum can be understood as the average of rectangles with width a.



For multi-dimensional integrals, the volume integral is given by,

$$I = \int_V f(x)\,dV, \tag{72}$$

and Monte Carlo estimation is,

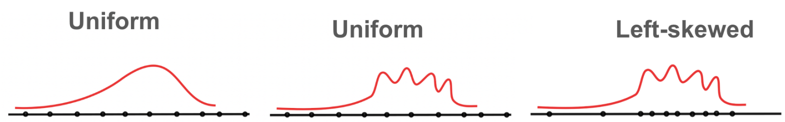
$$I \approx \frac{V}{N}\sum_{i=1}^{N} f(x_i), \quad x_i \sim q(x) = U(0, V) \tag{73}$$

where x_i are randomly drawn **from the d-dimensional hypercube** with volume V, $x_i \sim q(x) = U(0, V)$.

In other words, we effectively transform the d-dimensional integrator from a given volume into a hyper-cube with differential $D\mathbf{u} = \prod_i^d du_i$, such that,

$$dV = q(u_1, u_2, \cdots, u_d)\prod_{i=0}^{d} du_i = q(\mathbf{u})D\mathbf{u}, \tag{74}$$

where $q(\mathbf{u})$ is the "**density of state**" for a given parametrization $\mathbf{u}$. In the above case, we assume a uniform distribution $q \sim U(0, b-a)$, but in general, any well-behaved distribution can be used to further improve the number of samples needed. For example, if we sample the function below with uniform distribution, satisfactory convergence requires large number of samples (middle figure). On the other hand, if we sample with different distribution, a faster convergence may achieve (right side figure). This concept is called the **Importance Sampling**.



3.2 Importance Sampling and regularization

Let's say we have a set of data points following a given PDF $f(x)$, and we want to find the CDF associated with this PDF,

$$F(x) = \int_{x_0}^{x} f(x)\, dx, \tag{75}$$

where x_0 is set by the **support** of the random variable. While PDF $f(x)$ is normalizable, the PDF itself may diverge. For example, let's consider the following PDF $f(x) = x^{-\alpha}e^{-x}/\Gamma(1-\alpha)$ where $0 < \alpha < 1$ with support on $x \geq 0$ and $\Gamma(1-\alpha)$ is the gamma function for normalization. The CDF is given by,

$$F(x) = \frac{1}{\Gamma(1-\alpha)} \int_{0}^{x} x'^{-\alpha} e^{-x'}\, dx', \quad \Gamma(1-\alpha) = \int_{0}^{\infty} x'^{-\alpha} e^{-x'}\, dx'. \tag{76}$$

However, when $x \approx 0$, the PDF,

$$f(x) = \frac{1}{\Gamma(1-\alpha)} x^{-\alpha} e^{-x}, \tag{77}$$

experiences divergence, $f(x) \to \infty$, which generates numerical instability, e.g. When performing the rejection sampling, evaluating the PDF near 0 may yield excessively large values that distort results. Therefore, we must preform **regularization** that suppresses the divergence behavior in the PDF. We should note that this problem is very common in numerical integration, not limited to CDF calculations.

One regularization methods is the **importance sampling**, it rewrites the CDF integral as follows,

$$F(x) = \int_{x_0}^{x} f(x')\, dx' = \int_{x_0}^{x} \frac{f(x')}{q(x')} q(x')\, dx' = \int_{x_0}^{x} w(x') q(x')\, dx', \tag{78}$$

where $q(x')$ is a **proposal function** or regulatory function, and $w(x')$ is the **weight function**. This not only allows us to evaluate the CDF integral using samples from $q(x')$, but also remove any singularities within the original PDF via the weighted ratio $\frac{f(x)}{q(x)}$.

Consider the CDF integral we discussed earlier,

$$F(x) = \frac{1}{\Gamma(1-\alpha)} \int_0^x x'^{-\alpha} e^{-x'}\, dx'. \tag{79}$$

Let's introduce a proposal distribution $q(x)$,

$$q(x) = \frac{1}{1-\alpha} x^{-\alpha}, \tag{80}$$

and rewriting the integral using importance sampling,

$$F(x) = \frac{1}{\Gamma(1-\alpha)} \int_0^x x'^{-\alpha} e^{-x'} \cdot \left([1-\alpha] x'^{\alpha}\right) g(x')\, dx'. \tag{81}$$

This gives the following CDF integral,

$$F(x) = \frac{1-\alpha}{\Gamma(1-\alpha)} \int_0^x e^{-x'}\, q(x')\, dx'. \tag{82}$$

You may see a similarity in rejection sampling if we set the proposal function to be $g(x') = q(x')$ and the target function to be $f(x') = e^{-x'} q(x')$ such that $g(x') \geq f(x')$. The ratio,

$$R(x') = \frac{e^{-x'} q(x')}{M q(x')} = e^{-x'} = w(x'). \tag{83}$$

However, there is a **fundamental difference between rejection sampling and importance sampling**, the former is a binary decision problem where we reject samples based on some conditions or probabilities; The latter does not reject any sample, but only rescale the sample by the weight $w(x')$, where the sample follows the proposal function $x' \sim q$, and it can be sampled using inverse sampling techniques discussed earlier,

$$x' = Q^{-1}(u_1), \tag{84}$$

where $u_1 \sim U(0,1)$. Since $q(x') = q(x) = \frac{1}{1-\alpha} x^{-\alpha}$, the CDF is given by,

$$u_1 = Q = \int_0^x q(x) dx = \frac{1}{1-\alpha} x'^{-\alpha} dx' = x^{1-\alpha}, \quad x' = Q^{-1}(u_1) = (u_1)^{\frac{1}{1-\alpha}}. \tag{85}$$

Therefore, we have,

$$F(x) = \frac{1-\alpha}{\Gamma(1-\alpha)} \int_0^x e^{-x'}\, q(x')\, dx' = \frac{1-\alpha}{\Gamma(1-\alpha)} \sum_i e^{-x_i}, \quad x_i = (u_1)^{\frac{1}{1-\alpha}}, \quad x_i \leq x, \quad u_1 \sim U(0,1). \tag{86}$$