# Monte Carlo Integration I

Digital Image Synthesis

*Yung-Yu Chuang*

*with slides by Pat Hanrahan and Torsten Moller*

# Introduction

$$L_o(\mathrm{p},\omega_\mathrm{o}) = L_e(\mathrm{p},\omega_\mathrm{o}) + \int_{s^2} f(\mathrm{p},\omega_\mathrm{o},\omega_\mathrm{i}) L_i(\mathrm{p},\omega_\mathrm{i}) \left|\cos\theta_\mathrm{i}\right| d\omega_\mathrm{i}$$

- The integral equations generally don't have analytic solutions, so we must turn to numerical methods.
- Standard methods like Trapezoidal integration or Gaussian quadrature are not effective for high-dimensional and discontinuous integrals.

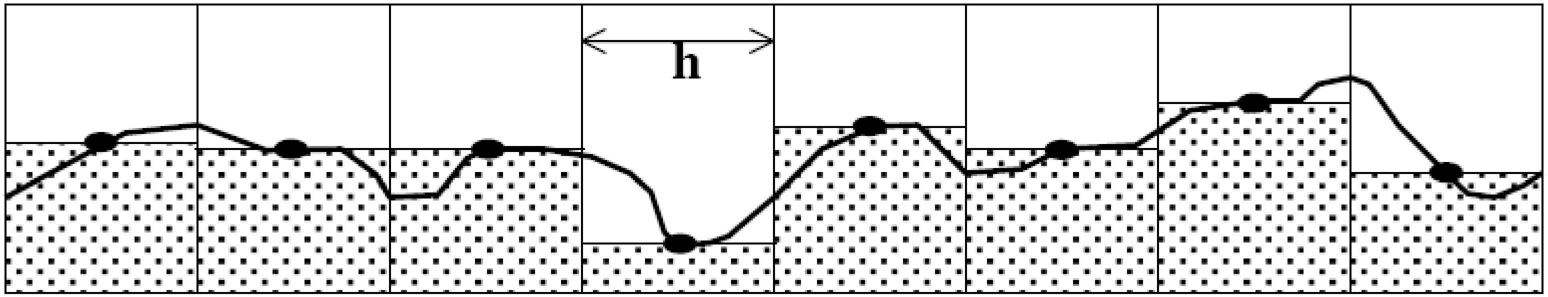
# Numerical quadrature

- Suppose we want to calculate $I = \int_a^b f(x)dx$, but can't solve it analytically. The approximations through quadrature rules have the form

$$\hat{I} = \sum_{i=1}^{n} w_i f(x_i)$$

which is essentially the weighted sum of samples of the function at various points

# Midpoint rule

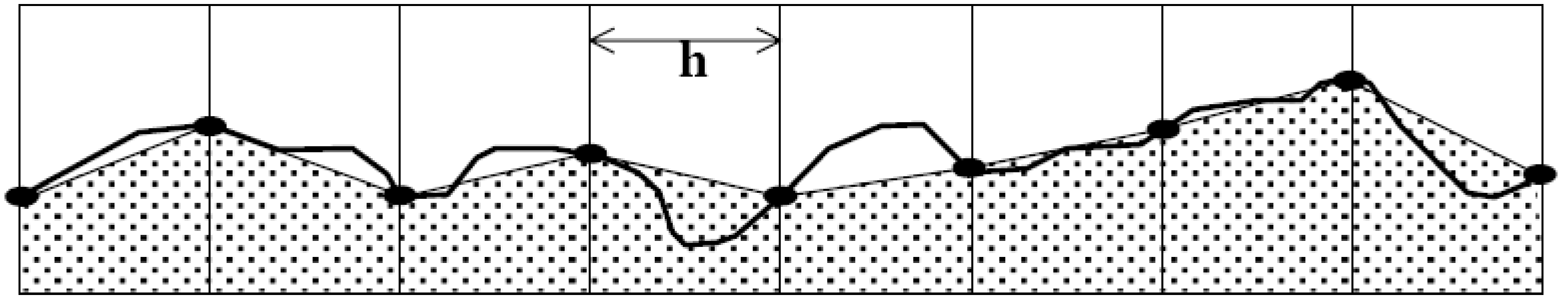



$$\hat{I} = h\sum_{i=1}^{n} f(a + (i - \frac{1}{2})h)$$

$$= h\left[f(a + \frac{h}{2}) + f(a + \frac{3h}{2}) + \cdots + f(b - \frac{h}{2})\right]$$

convergence $\hat{I} - I = -\frac{(b-a)^3}{24n^2} f''(\xi) = O(n^{-2})$

# Trapezoid rule




$$\hat{I} = \sum_{i=1}^{n} \frac{h}{2}\left[f(a+(i-1)h) + f(a+ih)\right]$$

$$= h\left[\frac{1}{2}f(a) + f(a+h) + f(a+2h) + \cdots + f(b-h) + \frac{1}{2}f(b)\right]$$

convergence $\hat{I} - I = \frac{(b-a)^3}{12n^2} f''(\xi^*) = O(n^{-2})$

# Simpson's rule

- Similar to trapezoid but using a quadratic polynomial approximation

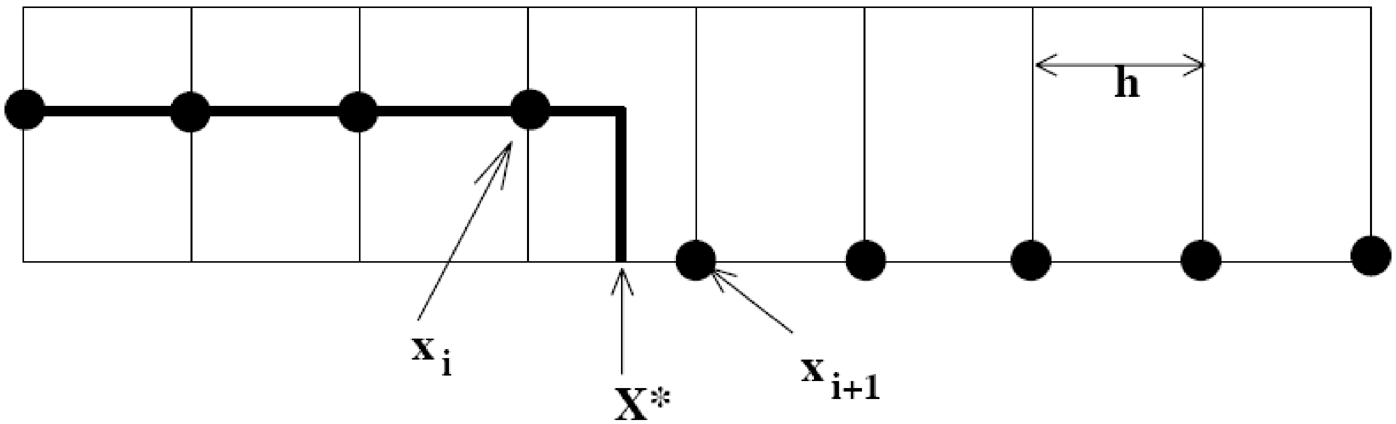
$$\hat{I} = h\left[\frac{1}{3}f(a) + \frac{4}{3}f(a+h) + \frac{2}{3}f(a+2h) + \frac{4}{3}f(a+3h) + \frac{2}{3}f(a+4h) + \cdots + \frac{4}{3}f(b-h) + \frac{1}{3}f(b)\right]$$

convergence $|\hat{I} - I| = \frac{(b-a)^5}{180(2n)^4} f^{(4)}(\xi) = O(n^{-4})$

assuming $f$ has a continuous fourth derivative.

# Curse of dimensionality and discontinuity

- For an $s$d function $f$, $$\hat{I} = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \cdots \sum_{i_s=1}^{n} w_{i_1} w_{i_2} \cdots w_{i_s} f(x_{i_1}, x_{i_2}, \ldots, x_{i_s})$$
- If the 1d rule has a convergence rate of $O(n^{-r})$, the $s$d rule would require a much larger number ($n^s$) of samples to work as well as the 1d one. Thus, the convergence rate is only $O(n^{-r/s})$.
- If f is discontinuous, convergence is $O(n^{-1/s})$ for $s$d.



# Randomized algorithms

- *Las Vegas* v.s. *Monte Carlo*
- *Las Vegas:* always gives the right answer by using randomness.
- *Monte Carlo*: gives the right answer *on the average*. Results depend on random numbers used, but statistically likely to be close to the right answer.

# Monte Carlo integration

- Monte Carlo integration: uses sampling to estimate the values of integrals. It only requires to be able to evaluate the integrand at arbitrary points, making it *easy to implement* and *applicable to many problems*.
- If $n$ samples are used, its converges at the rate of $O(n^{-1/2})$. That is, to cut the error in half, it is necessary to evaluate four times as many samples.
- Images by Monte Carlo methods are often noisy. Most current methods try to reduce noise.

# Monte Carlo methods

- Advantages
  - Easy to implement
  - Easy to think about (but be careful of statistical bias)
  - Robust when used with complex integrands and domains (shapes, lights, …)
  - Efficient for high dimensional integrals
- Disadvantages
  - Noisy
  - Slow (many samples needed for convergence)

# Basic concepts

- $X$ is a random variable
- Applying a function to a random variable gives another random variable, $Y=f(X)$.
- CDF (cumulative distribution function)

$$P(x) \equiv \Pr\{X \leq x\}$$

- PDF (probability density function): nonnegative, sum to 1

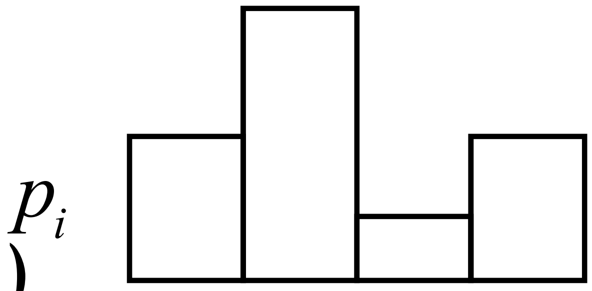
$$p(x) \equiv \frac{dP(x)}{dx}$$

- canonical uniform random variable ξ (provided by standard library and easy to transform to other distributions)

# Discrete probability distributions

- Discrete events $X_i$ with probability $p_i$

$$p_i \geq 0 \qquad \sum_{i=1}^{n} p_i = 1$$




- Cumulative PDF (distribution)

$$P_j = \sum_{i=1}^{j} p_i$$

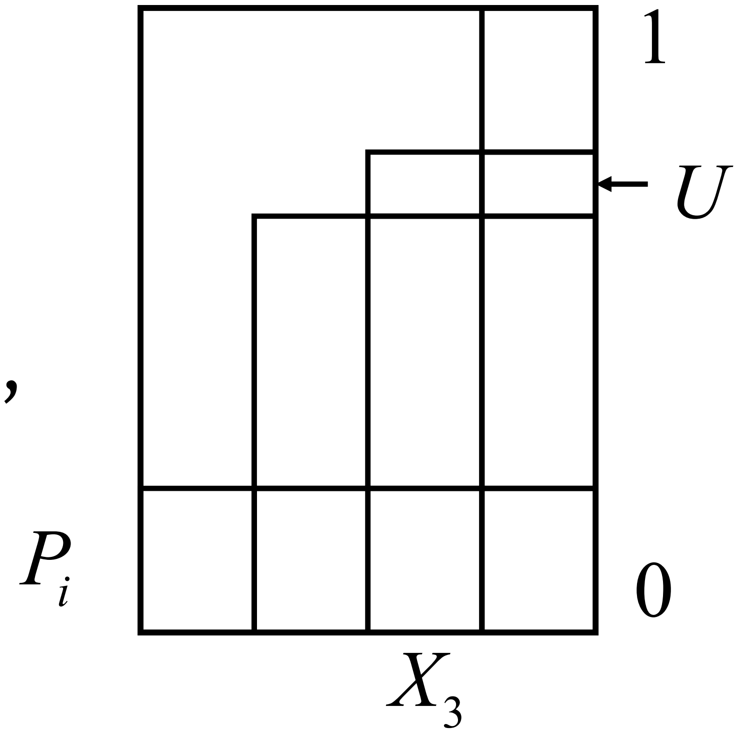
- Construction of samples:

  To randomly select an event,

  Select $X_i$ if $P_{i-1} < U \leq P_i$

  ↑

  **Uniform random variable**



# Continuous probability distributions

- PDF $p(x)$

$$p(x) \geq 0$$

- CDF $P(x)$

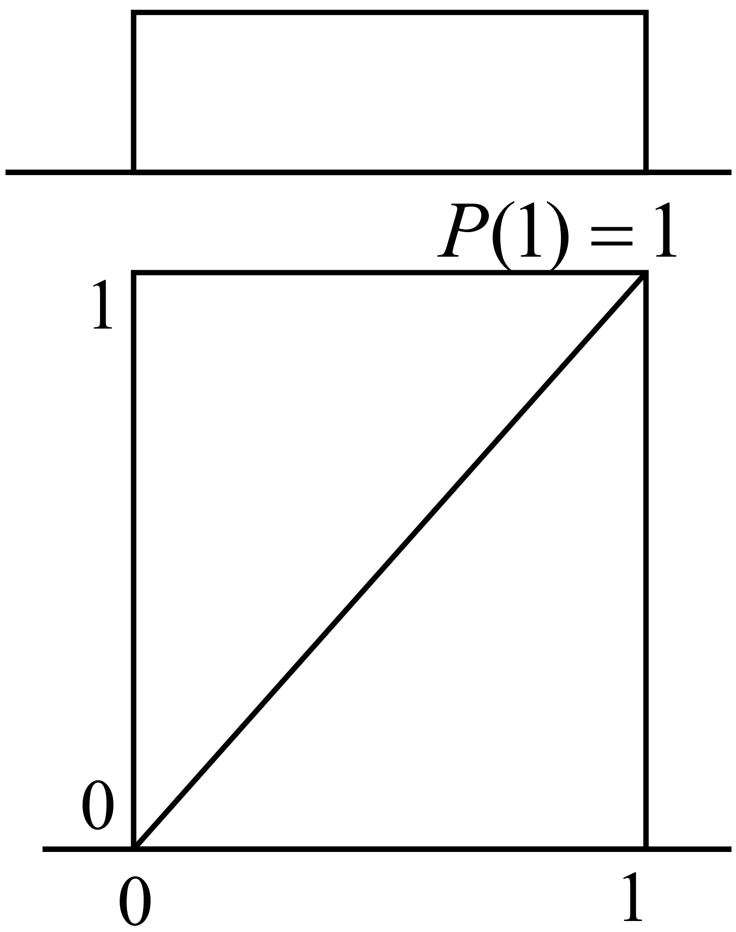
$$P(x) = \int_0^x p(x)dx$$

$$P(x) = \Pr(X < x)$$

$$\Pr(\alpha \leq X \leq \beta) = \int_\alpha^\beta p(x)\,dx$$

$$= P(\beta) - P(\alpha)$$

**Uniform**



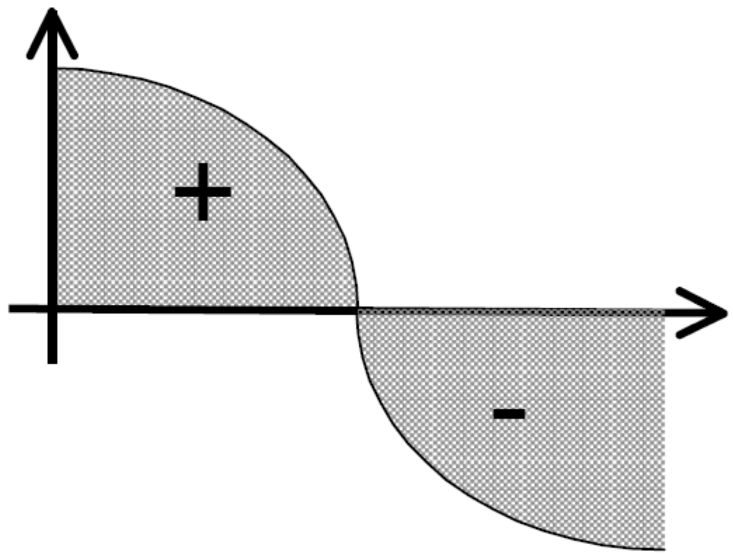
# Expected values

- Average value of a function *f(x)* over some distribution of values *p(x)* over its domain *D*

$$E_p\left[f(x)\right] = \int_D f(x)p(x)dx$$

- Example: *cos* function over $[0, \pi]$, $p$ is uniform

$$E_p\left[\cos(x)\right] = \int_0^\pi \cos x \frac{1}{\pi} dx = 0$$



# Variance

- Expected deviation from the expected value
- Fundamental concept of quantifying the error in Monte Carlo methods

$$V[f(x)] = E\left[\left(f(x) - E[f(x)]\right)^2\right]$$

# Properties

$$E\left[af(x)\right] = aE\left[f(x)\right]$$

$$E\left[\sum_i f(X_i)\right] = \sum_i E\left[f(X_i)\right]$$

$$V\left[af(x)\right] = a^2 V\left[f(x)\right]$$

$$\longrightarrow \quad V\left[f(x)\right] = E\left[\left(f(x)\right)^2\right] - E\left[f(x)\right]^2$$

# Monte Carlo estimator

- Assume that we want to evaluate the integral of $f(x)$ over [a,b] $\int_a^b f(x)dx$
- Given a uniform random variable $X_i$ over [a,b], Monte Carlo estimator

$$F_N = \frac{b-a}{N}\sum_{i=1}^{N} f(X_i)$$

says that the expected value $E[F_N]$ of the estimator $F_N$ equals the integral

$$E[F_N] = E\left[\frac{b-a}{N}\sum_{i=1}^{N} f(X_i)\right]$$

$$= \frac{b-a}{N}\sum_{i=1}^{N} E[f(X_i)]$$

$$= \frac{b-a}{N}\sum_{i=1}^{N} \int_a^b f(x)p(x)dx$$

$$= \frac{1}{N}\sum_{i=1}^{N} \int_a^b f(x)dx$$

$$= \int_a^b f(x)dx$$

# General Monte Carlo estimator

- Given a random variable $X$ drawn from an arbitrary PDF $p(x)$, then the estimator is

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$$

$$E[F_N] = E\left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}\right] = \frac{1}{N}\sum_{i=1}^{N}\int_a^b \frac{f(x)}{p(x)}p(x)dx = \int_a^b f(x)dx$$

- Although the converge rate of MC estimator is $O(N^{1/2})$, slower than other integral methods, its converge rate is independent of the dimension, making it the only practical method for high dimensional integral

# Convergence of Monte Carlo

- Chebyshev's inequality: let X be a random variable with expected value μ and variance $\sigma^2$. For any real number k>0,

$$\Pr\{|X-\mu| \geq k\sigma\} \leq \frac{1}{k^2}$$

- For example, for k=$\sqrt{2}$, it shows that at least half of the value lie in the interval $(\mu-\sqrt{2}\sigma, \mu+\sqrt{2}\sigma)$
- Let $Y_i = f(X_i)/p(X_i)$, the MC estimate $F_N$ becomes

$$F_N = \frac{1}{N}\sum_{i=1}^{N} Y_i$$

# Convergence of Monte Carlo

- According to Chebyshev's inequality,

$$\Pr\left\{ | F_N - E[F_N] | \geq \left( \frac{V[F_N]}{\delta} \right)^{1/2} \right\} \leq \delta$$

$$V[F_N] = V\left[ \frac{1}{N} \sum_{i=1}^{N} Y_i \right] = \frac{1}{N^2} V\left[ \sum_{i=1}^{N} Y_i \right] = \frac{1}{N^2} \sum_{i=1}^{N} V[Y_i] = \frac{1}{N} V[Y]$$

- Plugging into Chebyshev's inequality,

$$\Pr\left\{ | F_N - I | \geq \frac{1}{\sqrt{N}} \left( \frac{V[Y]}{\delta} \right)^{1/2} \right\} \leq \delta$$

So, for a fixed threshold, the error decreases at the rate $N^{-1/2}$.

# Properties of estimators

- An estimator $F_N$ is called unbiased if for all $N$

$$E[F_N] = Q$$

That is, the expected value is independent of $N$.

- Otherwise, the bias of the estimator is defined as

$$\beta[F_N] = E[F_N] - Q$$

- If the bias goes to zero as $N$ increases, the estimator is called consistent

$$\lim_{N\to\infty} \beta[F_N] = 0$$

$$\lim_{N\to\infty} E[F_N] = Q$$

# Example of a biased consistent estimator

- Suppose we are doing antialiasing on a 1d pixel, to determine the pixel value, we need to evaluate $I = \int_0^1 w(x) f(x) dx$, where $w(x)$ is the filter function with $\int_0^1 w(x) dx = 1$
- A common way to evaluate this is

$$F_N = \frac{\sum_{i=1}^{N} w(X_i) f(X_i)}{\sum_{i=1}^{N} w(X_i)}$$

- When N=1, we have

$$E[F_1] = E\left[\frac{w(X_1) f(X_1)}{w(X_1)}\right] = E[f(X_1)] = \int_0^1 f(x) dx \neq I$$

# Example of a biased consistent estimator

- When N=2, we have

$$E[F_2] = \int_0^1 \int_0^1 \frac{w(x_1) f(x_1) + w(x_2) f(x_2)}{w(x_1) + w(x_2)} dx_1 dx_2 \neq I$$

- However, when N is very large, the bias approaches to zero

$$F_N = \frac{\frac{1}{N}\sum_{i=1}^{N} w(X_i) f(X_i)}{\frac{1}{N}\sum_{i=1}^{N} w(X_i)}$$

$$\lim_{N\to\infty} E[F_N] = \frac{\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} w(X_i) f(X_i)}{\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} w(X_i)} = \frac{\int_0^1 w(x) f(x) dx}{\int_0^1 w(x) dx} = \int_0^1 w(x) f(x) dx = I$$

# Choosing samples

- $$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$$
- Carefully choosing the PDF from which samples are drawn is an important technique to reduce variance. We want the $f/p$ to have a low variance. Hence, it is necessary to be able to draw samples from the chosen PDF.
- How to sample an arbitrary distribution from a variable of uniform distribution?
  - Inversion
  - Rejection
  - Transform

# Inversion method

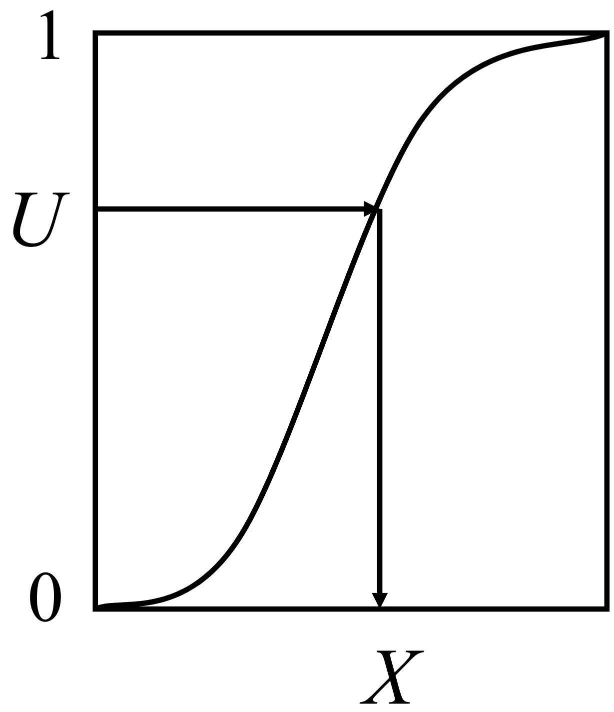
- Cumulative probability distribution function

$$P(x) = \Pr(X < x)$$

- Construction of samples

  Solve for $X = P^{-1}(U)$

- Must know:
  1. The integral of $p(x)$
  2. The inverse function $P^{-1}(x)$



# Proof for the inversion method

- Let U be an uniform random variable and its CDF is $P_u(x)=x$. We will show that $Y=P^{-1}(U)$ has the CDF $P(x)$.

# Proof for the inversion method

- Let U be an uniform random variable and its CDF is $P_u(x)=x$. We will show that $Y=P^{-1}(U)$ has the CDF $P(x)$.

$$\Pr\{Y \le x\} = \Pr\{P^{-1}(U) \le x\} = \Pr\{U \le P(x)\} = P_u(P(x)) = P(x)$$

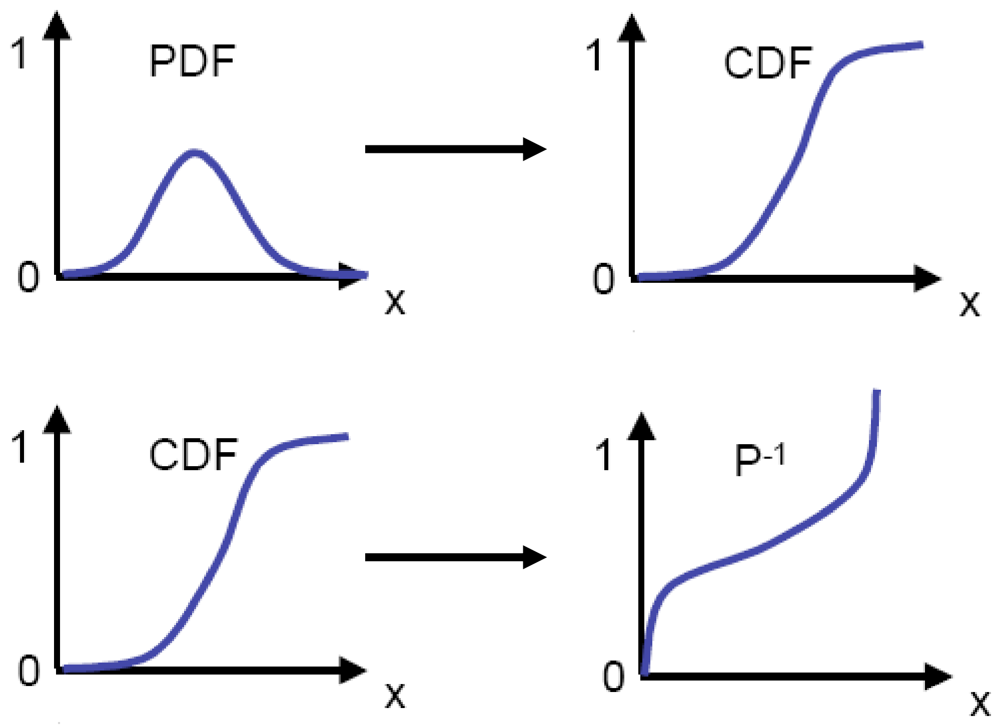
because $P$ is monotonic,

$$x_1 \le x_2 \Rightarrow P(x_1) \le P(x_2)$$

Thus, Y's CDF is exactly $P(x)$.

# Inversion method

- Compute CDF P(x)
- Compute $P^{-1}(x)$




- Obtain ξ
- Compute $X_i=P^{-1}(\xi)$

# Example: power function

It is used in sampling Blinn's microfacet model.

$$p(x) \propto x^n$$

# Example: power function

It is used in sampling Blinn's microfacet model.

- Assume

$$\int_0^1 x^n dx = \left.\frac{x^{n+1}}{n+1}\right|_0^1 = \frac{1}{n+1}$$

$$p(x) = (n+1)x^n$$

$$P(x) = x^{n+1}$$

$$X \sim p(x) \Rightarrow X = P^{-1}(U) = \sqrt[n+1]{U}$$

**Trick** (It only works for sampling power distribution)

$$Y = \max(U_1, U_2, \cdots, U_n, U_{n+1})$$

$$\Pr(Y < x) = \prod_{i=1}^{n+1} \Pr(U < x) = x^{n+1}$$

# Example: exponential distribution

$p(x) = ce^{-ax}$ useful for rendering participating media.

- Compute CDF P(x)
- Compute $P^{-1}(x)$
- Obtain ξ
- Compute $X_i = P^{-1}(\xi)$

# Example: exponential distribution

$$p(x) = ce^{-ax}$$ useful for rendering participating media.

$$\int_0^\infty ce^{-ax}\,dx = 1 \longrightarrow c = a$$

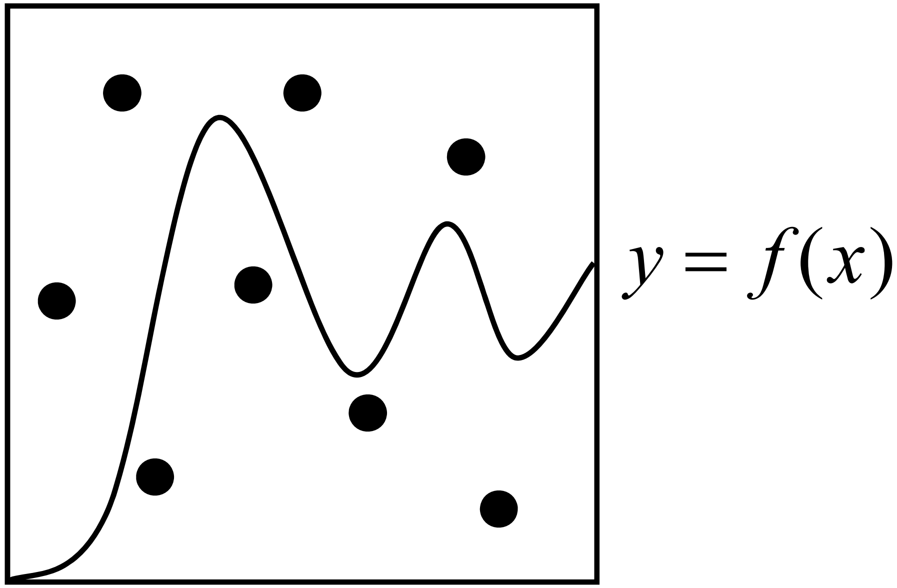
- Compute CDF P(x) $$P(x) = \int_0^x ae^{-as}\,ds = 1 - e^{-ax}$$
- Compute $P^{-1}(x)$ $$P^{-1}(x) = -\frac{1}{a}\ln(1-x)$$
- Obtain ξ
- Compute $X_i = P^{-1}(\xi)$ $$X = -\frac{1}{a}\ln(1-\xi) = -\frac{1}{a}\ln\xi$$

# Rejection method

- Sometimes, we can't integrate into CDF or invert CDF

$$I = \int_0^1 f(x)\,dx$$

$$= \iint_{y<f(x)} dx\,dy$$



- Algorithm

  Pick $U_1$ and $U_2$

  Accept $U_1$ if $U_2 < f(U_1)$

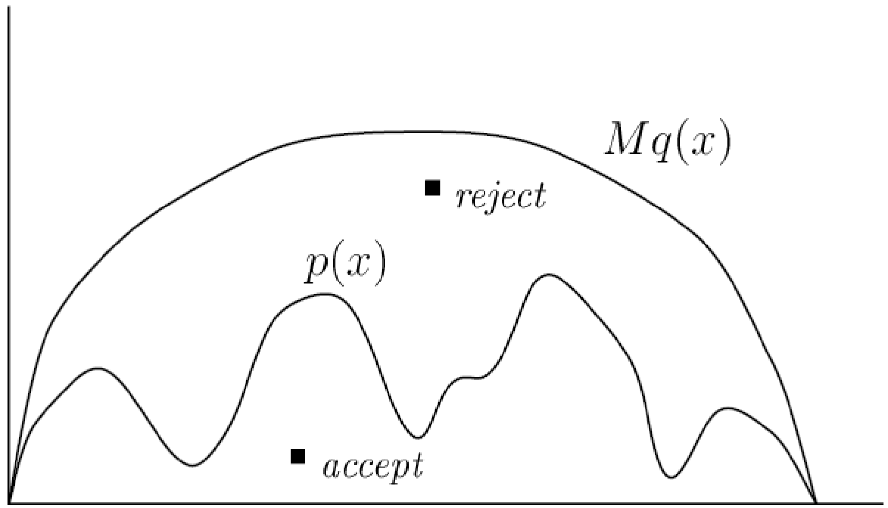
- Wasteful? **Efficiency = Area / Area of rectangle**

# Rejection method

- Rejection method is a dart-throwing method without performing integration and inversion.

1. Find $q(x)$ so that $p(x)<Mq(x)$
2. Dart throwing
   a. Choose a pair $(X, \xi)$, where $X$ is sampled from $q(x)$
   b. If $(\xi<p(X)/Mq(X))$ return $X$

- Equivalently, we pick point $(X, \xi Mq(X))$. If it lies beneath $p(X)$ then we are fine.



# Why it works

- For each iteration, we generate $X_i$ from $q$. The sample is returned if $\xi < p(X)/Mq(X)$, which happens with probability $p(X)/Mq(X)$.
- So, the probability to return $x$ is

$$q(x)\frac{p(x)}{Mq(x)} = \frac{p(x)}{M}$$

whose integral is 1/M

- Thus, when a sample is returned (with probability 1/M), $X_i$ is distributed according to $p(x)$.

# Example: sampling a unit disk

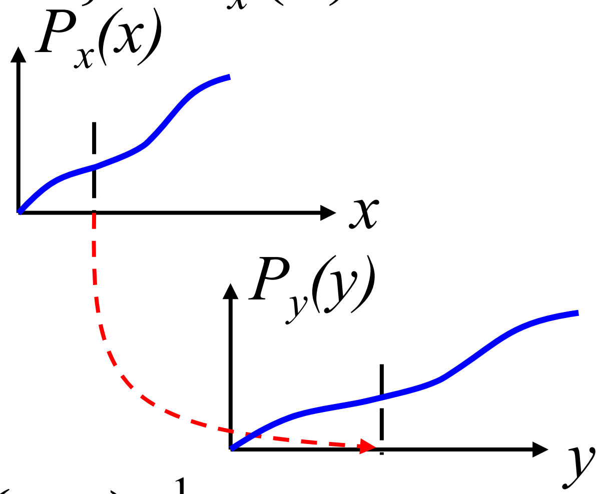
```
void RejectionSampleDisk(float *x, float *y) {
  float sx, sy;
  do {
    sx = 1.f -2.f * RandomFloat();
    sy = 1.f -2.f * RandomFloat();
  } while (sx*sx + sy*sy > 1.f)
  *x = sx;  *y = sy;
}
```

$\pi/4 \sim 78.5\%$ good samples, gets worse in higher dimensions, for example, for sphere, $\pi/6 \sim 52.4\%$

# Transformation of variables

- Given a random variable $X$ from distribution $p_x(x)$ to a random variable $Y=y(X)$, where $y$ is one-to-one, i.e. monotonic. We want to derive the distribution of $Y$, $p_y(y)$.
- $P_y(y(x)) = \Pr\{Y \le y(x)\} = \Pr\{X \le x\} = P_x(x)$
- PDF:

$$\frac{dP_y(y(x))}{dx} = \frac{dP_x(x)}{dx}$$




$$p_y(y)\frac{dy}{dx} = \frac{dP_y(y)}{dy}\frac{dy}{dx} \qquad p_x(x)$$

$$p_y(y) = \left(\frac{dy}{dx}\right)^{-1} p_x(x)$$

# Example

$$p_x(x) = 2x$$

$$Y = \sin X$$

$$p_y(y) = (\cos x)^{-1} p_x(x) = \frac{2x}{\cos x} = \frac{2 \sin^{-1} y}{\sqrt{1-y^2}}$$

# Transformation method

- A problem to apply the above method is that we usually have some PDF to sample from, not a given transformation.
- Given a source random variable $X$ with $p_x(x)$ and a target distribution $p_y(y)$, try transform $X$ into to another random variable $Y$ so that $Y$ has the distribution $p_y(y)$.
- We first have to find a transformation $y(x)$ so that $P_x(x)=P_y(y(x))$. Thus,

$$y(x) = P_y^{-1}(P_x(x))$$

# Transformation method

- Let's prove that the above transform works.

  We first prove that the random variable $Z = P_x(x)$ has a uniform distribution. If so, then $P_y^{-1}(Z)$ should have distribution $P_y$ from the inversion method.

$$\Pr\{Z \le x\} = \Pr\{P_x(X) \le x\} = \Pr\{X \le P_x^{-1}(x)\} = P_x(P_x^{-1}(x)) = x$$

  Thus, Z is uniform and the transformation works.

- It is an obvious generalization of the inversion method, in which $X$ is uniform and $P_x(x)=x$.

# Example

$$p_x(x) = x \xrightarrow{y} p_y(y) = e^y$$

# Example

$$p_x(x) = x \xrightarrow{\ y\ } p_y(y) = e^y$$

$$P_x(x) = \frac{x^2}{2} \qquad P_y(y) = e^y$$

$$P_y^{-1}(y) = \ln y$$

$$y(x) = P_y^{-1}(P_x(x)) = \ln(\frac{x^2}{2}) = 2\ln x - \ln 2$$

Thus, if X has the distribution $p_x(x) = x$ , then the random variable $Y = 2\ln X - \ln 2$ has the distribution $p_y(y) = e^y$

# Multiple dimensions

- Easily generalized - using the Jacobian of Y=T(X) $$p_y(T(x)) = |J_T(x)|^{-1} p_x(x)$$

- Example - polar coordinates $$x = r\cos\theta$$ $$y = r\sin\theta$$

$$J_T(x) = \begin{pmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{pmatrix}$$

$$p(x,y) = r^{-1} p(r,\theta)$$

We often need the other way around, $p(r,\theta) = r\,p(x,y)$

# Spherical coordinates

- The spherical coordinate representation of directions is $x = r\sin\theta\cos\phi$

$$y = r\sin\theta\sin\phi$$

$$z = r\cos\theta$$

$$|J_T| = r^2\sin\theta$$

$$p(r,\theta,\phi) = r^2\sin\theta p(x,y,z)$$

# Spherical coordinates

- Now, look at relation between spherical directions and a solid angles

$$d\omega = \sin\theta d\theta d\phi$$

- Hence, the density in terms of $\theta, \phi$

$$p(\theta,\phi) d\theta d\phi = p(\omega) d\omega$$

$$\boxed{p(\theta,\phi) = \sin\theta p(\omega)}$$

# Multidimensional sampling

- Separable case: independently sample X from $p_x$ and Y from $p_y$. $p(x,y)=p_x(x)p_y(y)$
- Often, this is not possible. Compute the marginal density function $p(x)$ first.

$$p(x) = \int p(x,y)dy$$

- Then, compute the conditional density function

$$p(y \mid x) = \frac{p(x,y)}{p(x)}$$

- Use 1D sampling with $p(x)$ and $p(y|x)$.

# Sampling a hemisphere

- Sample a hemisphere uniformly, *i.e.* $p(\omega) = c$

# Sampling a hemisphere

- Sample a hemisphere uniformly, *i.e.* $p(\omega) = c$

$$1 = \int_\Omega p(\omega) \qquad c = \frac{1}{2\pi} \longrightarrow p(\omega) = \frac{1}{2\pi}$$

$$\downarrow$$

$$p(\theta, \phi) = \frac{\sin\theta}{2\pi}$$

- Sample θ first

$$p(\theta) = \int_0^{2\pi} p(\theta, \phi) d\phi = \int_0^{2\pi} \frac{\sin\theta}{2\pi} d\phi = \sin\theta$$

- Now sampling ϕ

$$p(\phi \mid \theta) = \frac{p(\theta, \phi)}{p(\theta)} = \frac{1}{2\pi}$$

# Sampling a hemisphere

- Now, we use inversion technique in order to sample the PDF’s

$$P(\theta) = \int_0^{\theta} \sin\theta' \, d\theta' = 1 - \cos\theta$$

$$P(\phi \mid \theta) = \int_0^{\phi} \frac{1}{2\pi} \, d\phi' = \frac{\phi}{2\pi}$$

- Inverting these:

$$\theta = \cos^{-1} \xi_1$$

$$\phi = 2\pi\xi_2$$

# Sampling a hemisphere

- Convert these to Cartesian coordinate

$$\theta = \cos^{-1} \xi_1 \qquad x = \sin\theta\cos\phi = \cos(2\pi\xi_2)\sqrt{1-\xi_1^2}$$

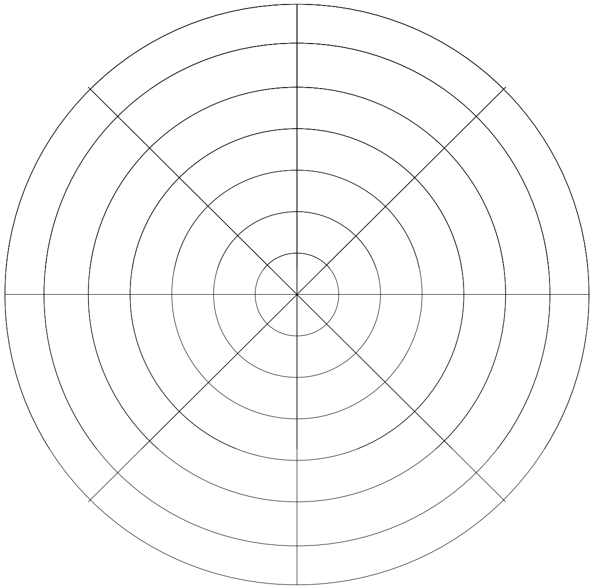
$$\phi = 2\pi\xi_2 \quad \longrightarrow \quad y = \sin\theta\sin\phi = \sin(2\pi\xi_2)\sqrt{1-\xi_1^2}$$

$$z = \cos\theta = \xi_1$$

- Similar derivation for a full sphere

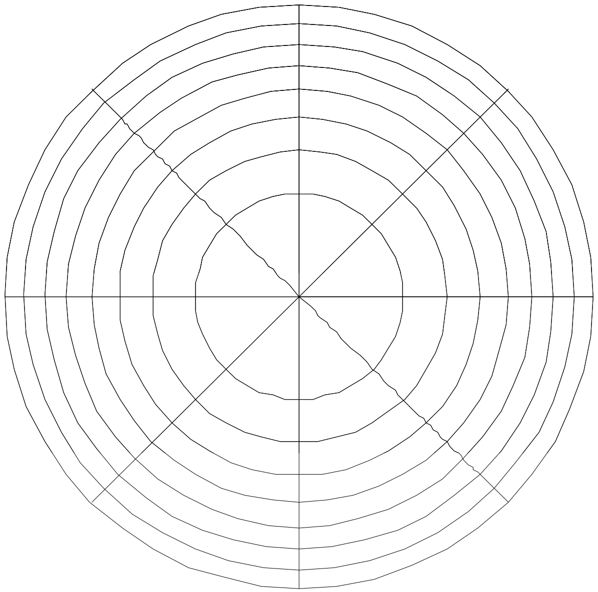
# Sampling a disk

**WRONG $\neq$ Equi-Areal**



$$\theta = 2\pi U_1$$
$$r = U_2$$

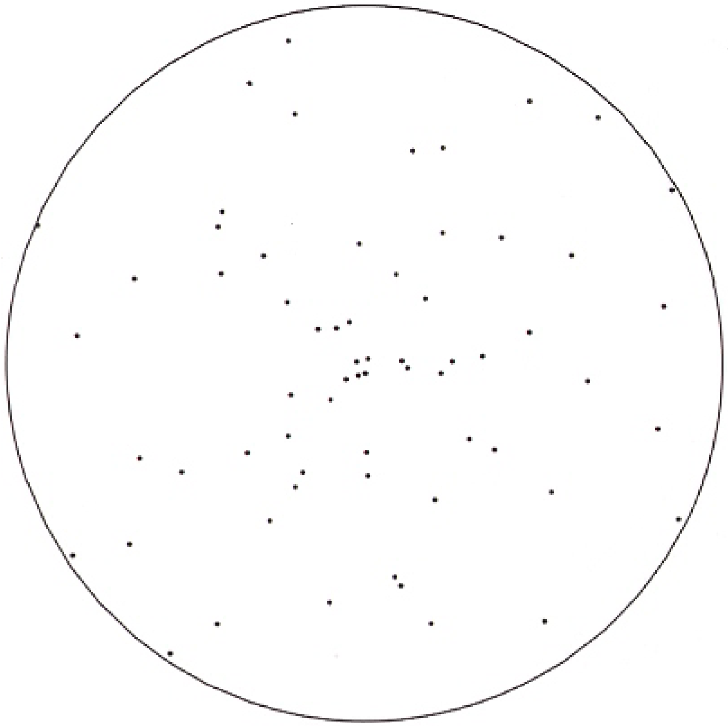
**RIGHT = Equi-Areal**



$$\theta = 2\pi U_1$$
$$r = \sqrt{U_2}$$

# Sampling a disk

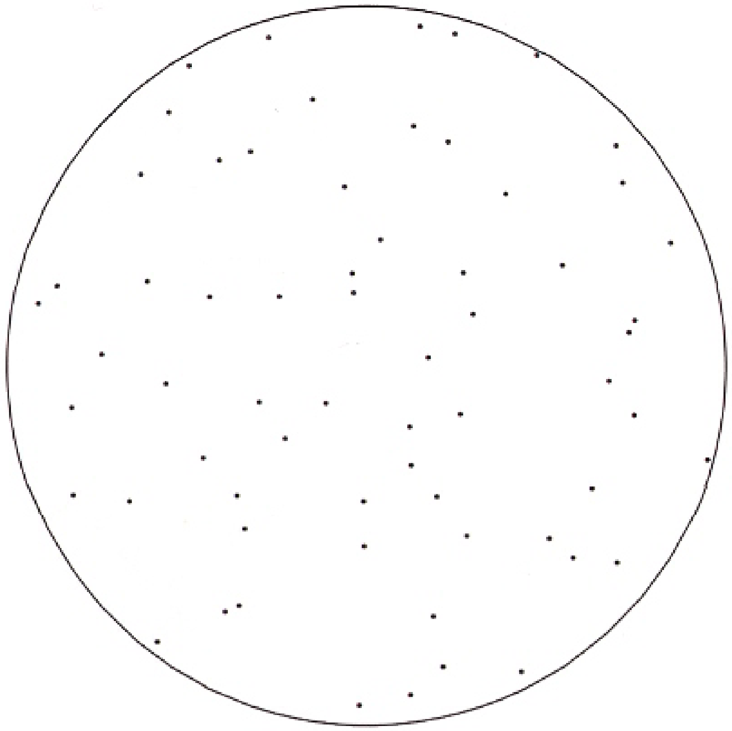
**WRONG ≠ Equi-Areal**



$$\theta = 2\pi U_1$$

$$r = U_2$$

**RIGHT = Equi-Areal**



$$\theta = 2\pi U_1$$

$$r = \sqrt{U_2}$$

# Sampling a disk

- Uniform $p(x,y) = \frac{1}{\pi}$ $p(r,\theta) = rp(x,y) = \frac{r}{\pi}$
- Sample $r$ first.

$$p(r) = \int_0^{2\pi} p(r,\theta)d\theta = 2r$$

- Then, sample $\theta$.

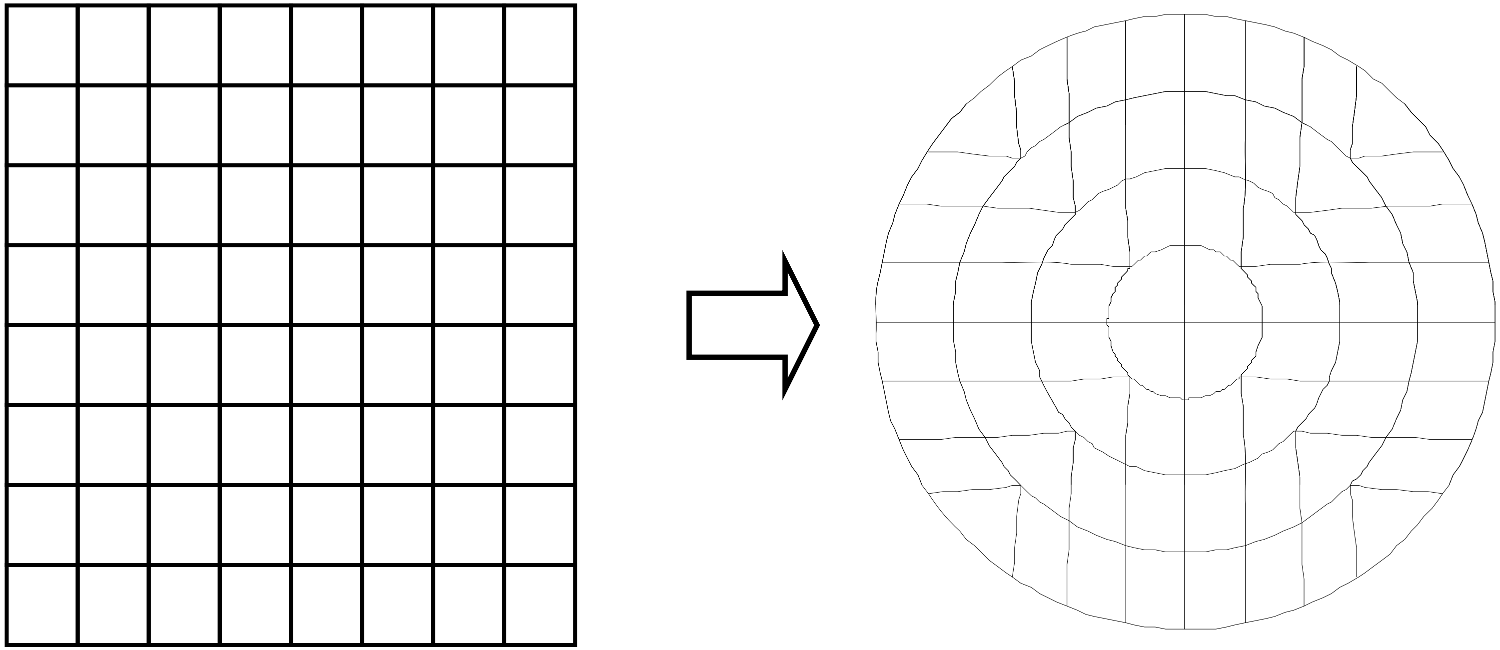
$$p(\theta \mid r) = \frac{p(r,\theta)}{p(r)} = \frac{1}{2\pi}$$

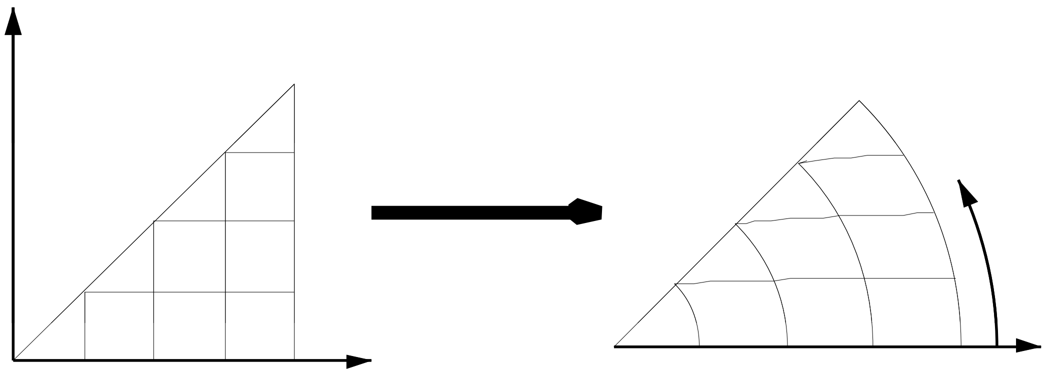
- Invert the CDF.

$$P(r) = r^2 \qquad P(\theta \mid r) = \frac{\theta}{2\pi}$$

$$r = \sqrt{\xi_1} \qquad \theta = 2\pi\xi_2$$

# Shirley's mapping





$$r = U_1$$

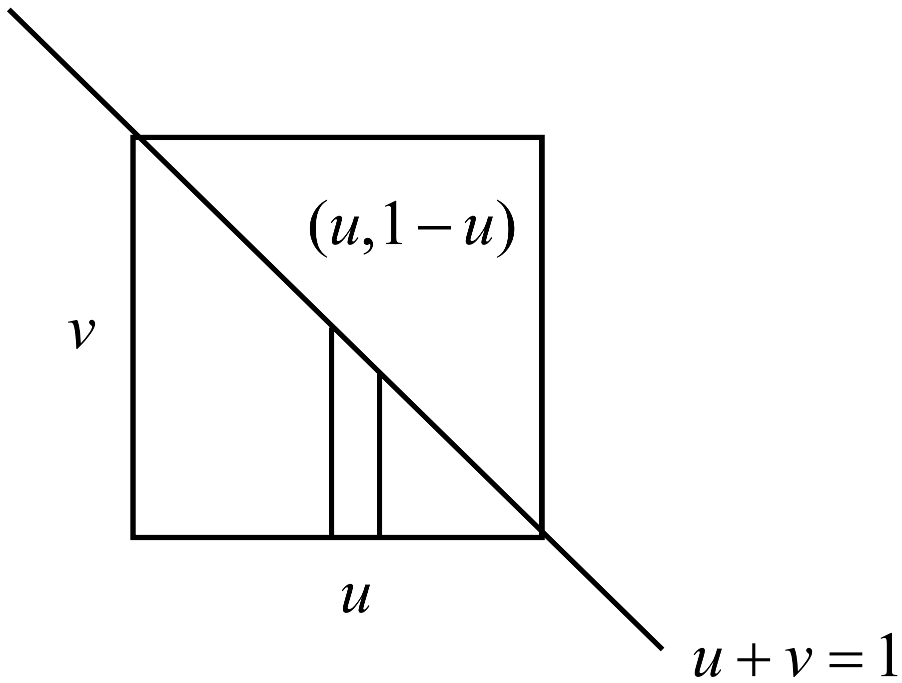
$$\theta = \frac{\pi}{4}\frac{U_2}{U_1}$$

# Sampling a triangle

$$u \geq 0$$

$$v \geq 0$$

$$u + v \leq 1$$




$$A = \int_0^1 \int_0^{1-u} dv\, du = \int_0^1 (1-u)\, du = -\frac{(1-u)^2}{2}\Bigg|_0^1 = \frac{1}{2}$$

$$p(u,v) = 2$$

# Sampling a triangle

- Here $u$ and $v$ are not independent! $p(u,v) = 2$
- Conditional probability

$$p(u) \equiv \int p(u,v)dv \qquad p(u \mid v) \equiv \frac{p(u,v)}{p(v)}$$

$$p(u) = 2\int_0^{1-u} dv = 2(1-u)$$

$$u_0 = 1 - \sqrt{U_1}$$

$$P(u_0) = \int_0^{u_0} 2(1-u)\,du \quad = (1-u_0)^2$$

$$p(v \mid u) = \frac{1}{(1-u)}$$

$$v_0 = \sqrt{U_1}U_2$$

$$P(v_0 \mid u_0) = \int_0^{v_o} p(v \mid u_0)\,dv \quad = \int_0^{v_o} \frac{1}{(1-u_0)}dv \quad = \frac{v_0}{(1-u_0)}$$

# Cosine weighted hemisphere

$$p(\omega) \propto \cos\theta$$

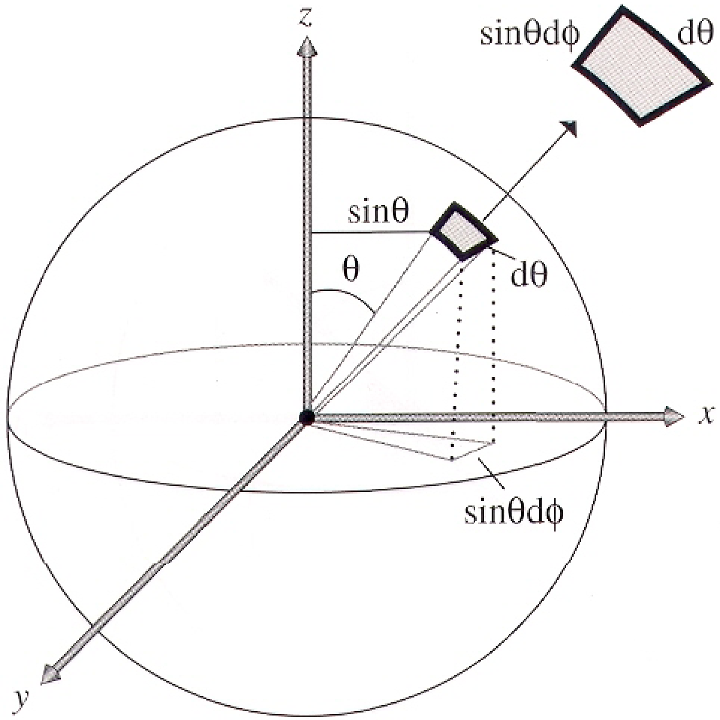
$$1 = \int_{\Omega} p(\omega) d\omega$$

$$1 = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} c \cos\theta \sin\theta d\theta d\phi$$

$$1 = c 2\pi \int_0^{\frac{\pi}{2}} \cos\theta \sin\theta d\theta$$

$$c = \frac{1}{\pi}$$

$$p(\theta, \phi) = \frac{1}{\pi} \cos\theta \sin\theta$$




$$d\omega = \sin\theta d\theta d\phi$$

# Cosine weighted hemisphere

$$p(\theta,\phi) = \frac{1}{\pi}\cos\theta\sin\theta$$

$$p(\theta) = \int_0^{2\pi} \frac{1}{\pi}\cos\theta\sin\theta d\phi = 2\cos\theta\sin\theta = \sin 2\theta$$

$$p(\phi \mid \theta) = \frac{p(\theta,\phi)}{p(\theta)} = \frac{1}{2\pi}$$

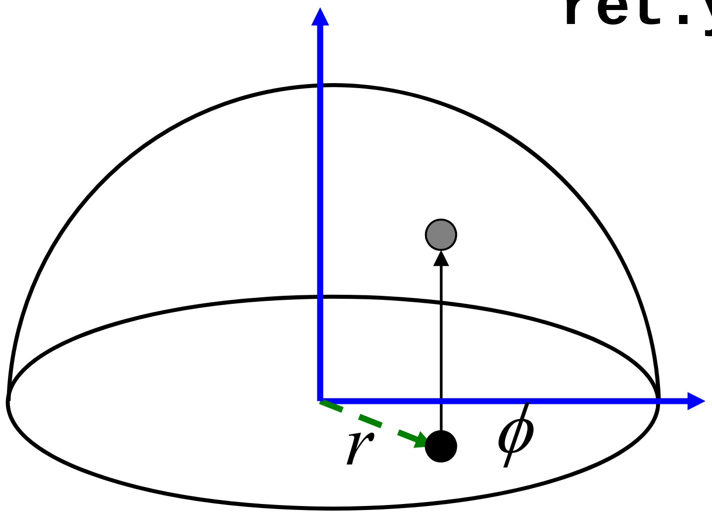
$$P(\theta) = -\frac{1}{2}\cos 2\theta + \frac{1}{2} = \xi_1 \qquad \theta = \frac{1}{2}\cos^{-1}(1-2\xi_1)$$

$$P(\phi \mid \theta) = \frac{\phi}{2\pi} = \xi_2 \qquad \phi = 2\pi\xi_2$$

# Cosine weighted hemisphere

- Malley's method: uniformly generates points on the unit disk and then generates directions by projecting them up to the hemisphere above it.

```
Vector CosineSampleHemisphere(float u1,float u2){
  Vector ret;
  ConcentricSampleDisk(u1, u2, &ret.x, &ret.y);
  ret.z = sqrtf(max(0.f,1.f - ret.x*ret.x -
                            ret.y*ret.y));
  return ret;
}
```



# Cosine weighted hemisphere

- Why does Malley's method work?
- Unit disk sampling $p(r,\phi) = \frac{r}{\pi}$
- Map to hemisphere $(r,\phi) \Rightarrow (\sin\theta, \phi)$

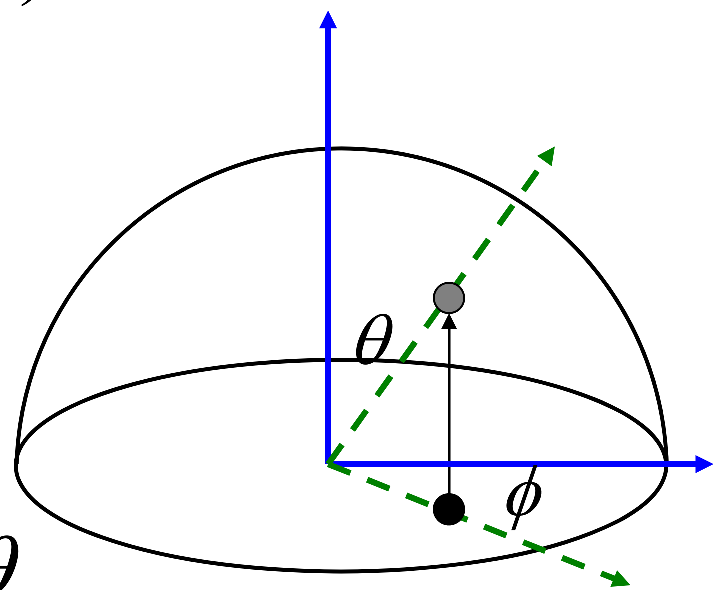
$$Y = (r,\phi) \xleftarrow{T} X = (\theta,\phi)$$

$$r = \sin\theta$$

$$\phi = \phi$$

$$p_y(T(x)) = \left|J_T(x)\right|^{-1} p_x(x)$$

$$\left|J_T(x)\right| = \left|\begin{pmatrix} \cos\theta & 0 \\ 0 & 1 \end{pmatrix}\right| = \cos\theta$$



# Cosine weighted hemisphere

$$Y = (r, \phi) \xleftarrow{T} X = (\theta, \phi)$$

$$r = \sin\theta$$

$$\phi = \phi$$

$$p_y(T(x)) = \left|J_T(x)\right|^{-1} p_x(x)$$

$$\left|J_T(x)\right| = \left|\begin{pmatrix} \cos\theta & 0 \\ 0 & 1 \end{pmatrix}\right| = \cos\theta$$

$$p(\theta, \phi) = \left|J_T\right| p(r, \phi) = \frac{\cos\theta \sin\theta}{\pi}$$

# Sampling Phong lobe

$$p(\omega) \propto \cos^n \theta$$

$$p(\omega) = c\cos^n \theta \rightarrow \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} c\cos^n \theta \sin\theta d\theta d\phi = 1$$

$$\rightarrow -2\pi c \int_{\cos\theta=1}^{0} \cos^n \theta d\cos\theta = 1 \rightarrow \frac{2\pi c}{n+1} = 1$$

$$\rightarrow c = \frac{n+1}{2\pi}$$

$$p(\theta, \phi) = \frac{n+1}{2\pi} \cos^n \theta \sin\theta$$

# Sampling Phong lobe

$$p(\theta,\phi)=\frac{n+1}{2\pi}\cos^n\theta\sin\theta$$

$$p(\theta)=\int_{\phi=0}^{2\pi}\frac{n+1}{2\pi}\cos^n\theta\sin\theta d\phi=(n+1)\cos^n\theta\sin\theta$$

$$P(\theta')=\int_{\theta=0}^{\theta'}(n+1)\cos^n\theta\sin\theta d\theta$$

$$=-(n+1)\int_{\theta=0}^{\theta'}\cos^n\theta d\cos\theta=-(n+1)\left.\frac{\cos^{n+1}\theta}{n+1}\right|_{\cos\theta=1}^{\cos\theta'}$$

$$=1-\cos^{n+1}\theta'$$

$$\theta=\cos^{-1}\left(\sqrt[n+1]{\xi_1}\right)$$

# Sampling Phong lobe

$$p(\theta,\phi)=\frac{n+1}{2\pi}\cos^n\theta\sin\theta$$

$$p(\phi\,|\,\theta)=\frac{p(\theta,\phi)}{p(\theta)}=\frac{\frac{n+1}{2\pi}\cos^n\theta\sin\theta}{(n+1)\cos^n\theta\sin\theta}=\frac{1}{2\pi}$$

$$P(\phi'\,|\,\theta)=\int_{\phi=0}^{\phi'}\frac{1}{2\pi}\,d\phi=\frac{\phi'}{2\pi}$$

$$\phi=2\pi\xi_2$$

# Sampling Phong lobe

When n=1, it is actually equivalent to cosine-weighted hemisphere

$$\mathrm{n}=1,\ (\theta,\phi)=(\cos^{-1}\sqrt{\xi_1},2\pi\xi_2) \qquad (\theta,\phi)=\left(\frac{1}{2}\cos^{-1}(1-2\xi_1),2\pi\xi_2\right)$$

$$P(\theta)=1-\cos^{n+1}\theta=1-\cos^2\theta \qquad P(\theta)=-\frac{1}{2}\cos 2\theta+\frac{1}{2}$$

$$-\frac{1}{2}\cos 2\theta+\frac{1}{2}=-\frac{1}{2}(1-2\sin^2\theta)+\frac{1}{2}=\sin^2\theta=1-\cos^2\theta$$

# Piecewise-constant 2d distributions

- Sample from discrete 2D distributions. Useful for texture maps and environment lights.
- Consider $f(u, v)$ defined by a set of $n_u \times n_v$ values $f[u_i, v_j]$.
- Given a continuous $[u, v]$, we will use $[u', v']$ to denote the corresponding discrete $(u_i, v_j)$ indices.

# Piecewise-constant 2d distributions

integral $$I_f = \iint f(u,v)\,dudv = \frac{1}{n_u n_v}\sum_{i=0}^{n_u-1}\sum_{j=0}^{n_v-1} f[u_i,v_j]$$

pdf $$p(u,v) = \frac{f(u,v)}{\iint f(u,v)\,dudv} = \frac{f\left[u',v'\right]}{1/(n_u n_v)\sum_i \sum_j f\left[u_i,v_j\right]}$$

marginal density $$p(v) = \int p(u,v)\,du = \frac{(1/n_u)\sum_i f\left[u_i,v'\right]}{I_f}$$

conditional probability $$p(u\,|\,v) = \frac{p(u,v)}{p(v)} = \frac{f[u',v']/I_f}{p[v']}$$

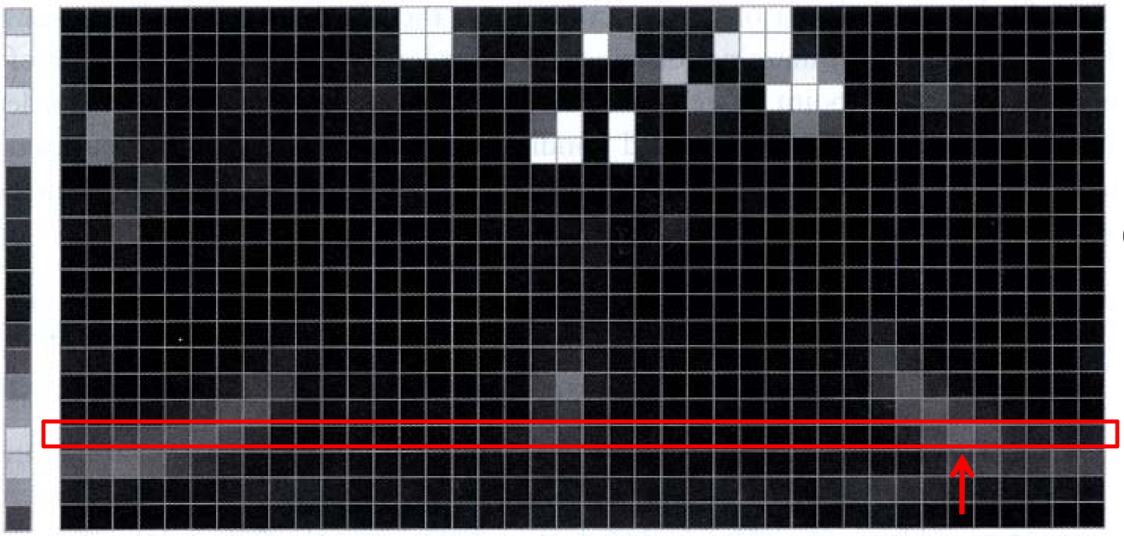
# Piecewise-constant 2d distributions

$f(u,v)$

(a)

`Distribution2D`

low-res marginal density $p(v)$



low-res conditional probability $p(u \mid v)$

# Metropolis sampling

- Metropolis sampling can efficiently generate a set of samples from any non-negative function $f$ requiring only the ability to evaluate $f$.
- Disadvantage: successive samples in the sequence are often correlated. It is not possible to ensure that a small number of samples generated by Metropolis is well distributed over the domain. There is no technique like stratified sampling for Metropolis.

# Metropolis sampling

- Problem: given an arbitrary function

$$f(x) \to \mathbb{R},\ x \in \Omega$$

assuming

$$\mathbf{I}(f) = \int_\Omega f(x) \mathrm{d}\Omega$$

$$f_{\mathrm{pdf}} = f / \mathbf{I}(f)$$

generate a set of samples

$$X = \{x_i\},\ x_i \sim f_{\mathrm{pdf}}$$

# Metropolis sampling

- Steps
  - Generate initial sample $x_0$
  - mutating current sample $x_i$ to propose $x'$
  - If it is accepted, $x_{i+1} = x'$

    Otherwise, $x_{i+1} = x_i$
- Acceptance probability guarantees distribution is the stationary distribution $f$

# Metropolis sampling

- Mutations propose $x'$ given $x_i$
- $T(x \to x')$ is the tentative transition probability density of proposing $x'$ from $x$
- Being able to calculate tentative transition probability is the only restriction for the choice of mutations
- $a(x \to x')$ is the acceptance probability of accepting the transition
- By defining $a(x \to x')$ carefully, we ensure

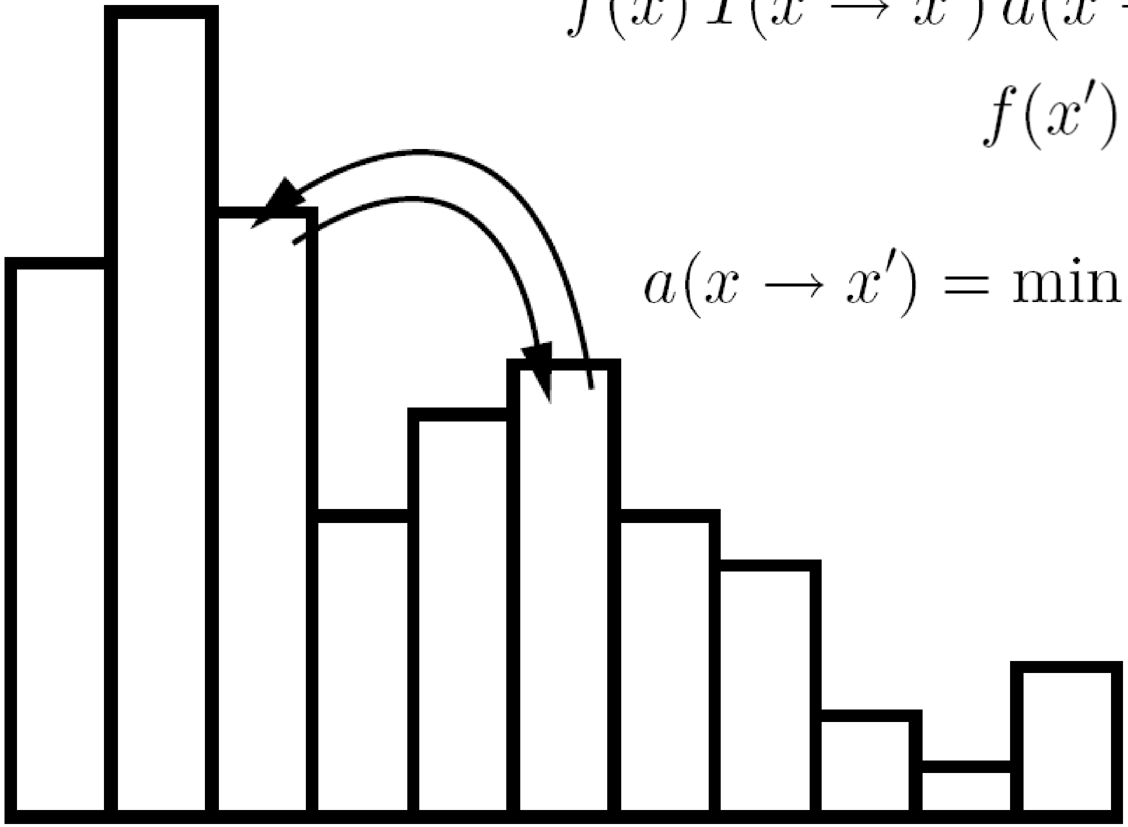
$$x_i \sim f(x)$$

# Metropolis sampling

- Detailed balance

$$f(x)\,T(x \to x')\,a(x \to x') = f(x')\,T(x' \to x)\,a(x' \to x)$$

$$a(x \to x') = \min\left(1, \frac{f(x')\,T(x' \to x)}{f(x)\,T(x \to x')}\right)$$



stationary distribution

# Pseudo code

```
x = x0
for i = 1 to n
    x' = mutate(x)
    a = accept(x, x')
    if (random() < a)
        x = x'
    record(x)
```

# Pseudo code (expected value)

```
x = x0
for i = 1 to n
    x' = mutate(x)
    a = accept(x, x')
    record(x, (1-a) * weight)
    record(x', a * weight)
    if (random() < a)
        x = x'
```

# Binary example I

$\Omega = a, b$ and $f(a) = 9$, $f(b) = 1$

$$\text{mutate}(x) = \begin{cases} a & : \quad \xi < 0.5 \\ b & : \quad \text{otherwise} \end{cases}$$

Then transition densities are

$$T(\{a, b\} \rightarrow \{a, b\}) = 1/2$$

It directly follows that

$$a(a \rightarrow b) = \min\left(1, f(b)/f(a)\right) = .1111\ldots$$

$$a(a \rightarrow a) = a(b \rightarrow a) = a(b \rightarrow b) = 1$$

# Binary example II

$\Omega = a, b$ and $f(a) = 9,\ f(b) = 1$

$$\text{mutate}(x) = \begin{cases} a & : \quad \xi < 8/9 \\ b & : \quad \text{otherwise} \end{cases}$$

transition densities

$$T(\{a,b\} \to a) = 8/9$$

$$T(\{a,b\} \to b) = 1/9$$

Acceptance probabilities

$$a(a \to b) = .9/.9 = 1$$

$$a(b \to a) = .9/.9 = 1$$

Better transitions improve acceptance probability

# Acceptance probability

- Does not affect unbiasedness; just variance
- Want transitions to happen because transitions are often heading where $f$ is large
- Maximize the acceptance probability
  - Explore state space better
  - Reduce correlation

# Mutation strategy

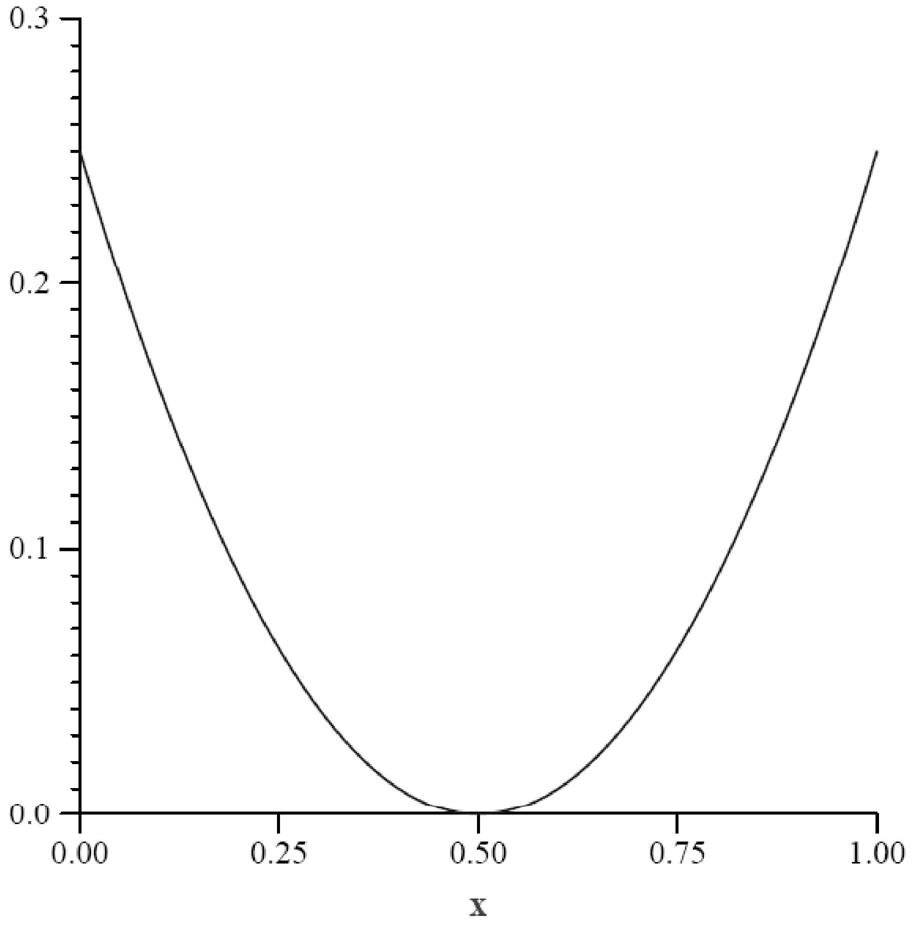
- Very free and flexible; the only requirement is to be able to calculate transition probability
- Based on applications and experience
- The more mutation, the better
- Relative frequency of them is not so important

# Start-up bias

- Using an initial sample not from f's distribution leads to a problem called start-up bias.
- Solution #1: run MS for a while and use the current sample as the initial sample to re-start the process.
  - Expensive start-up cost
  - Need to guess when to re-start
- Solution #2: use another available sampling method to start

# 1D example

$$f^1(x) = \begin{cases} (x - .5)^2 & : \quad 0 \le x \le 1 \\ 0 & : \quad \text{otherwise} \end{cases}$$



# 1D example (mutation)

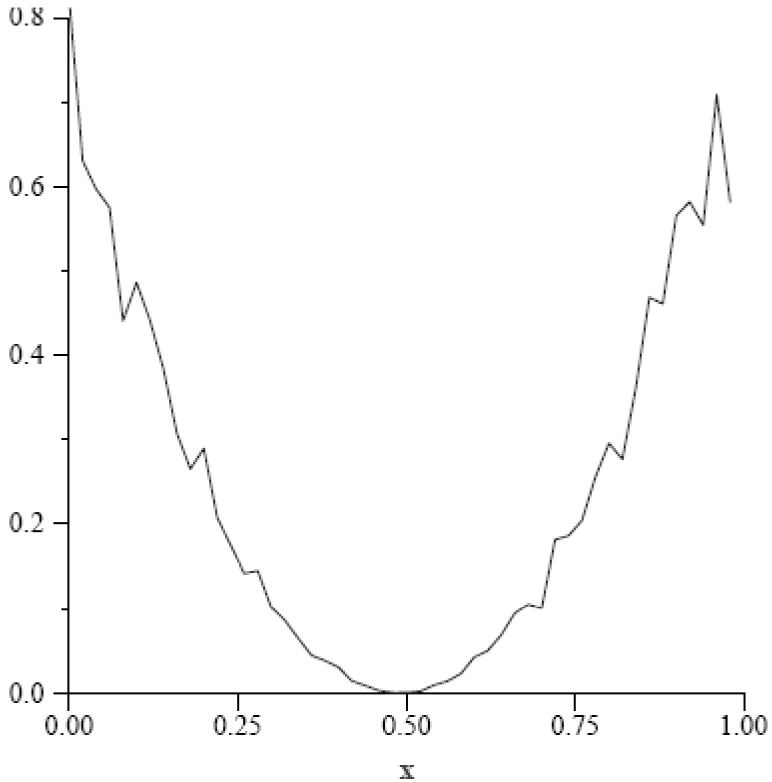
$$\mathrm{mutate}_1(x) \;\rightarrow\; \xi$$

$$T_1(x \rightarrow x') \;=\; 1$$

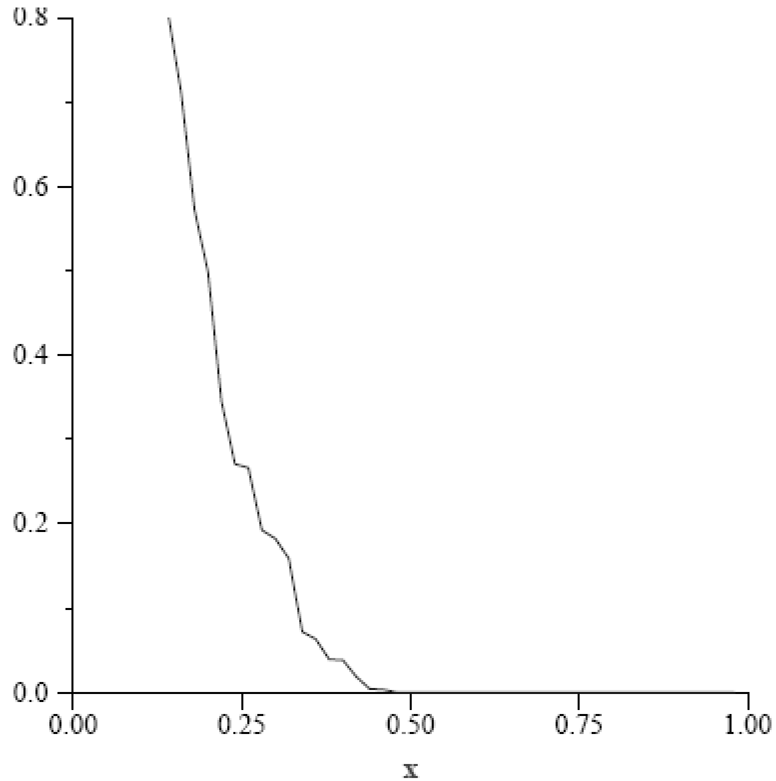
$$\mathrm{mutate}_2(x) \;\rightarrow\; x + .1 * (\xi - .5)$$

$$T_2(x \rightarrow x') \;=\; \begin{cases} \frac{1}{0.1} & : \quad |x - x'| \leq .05 \\ 0 & : \quad \text{otherwise} \end{cases}$$

# 1D example

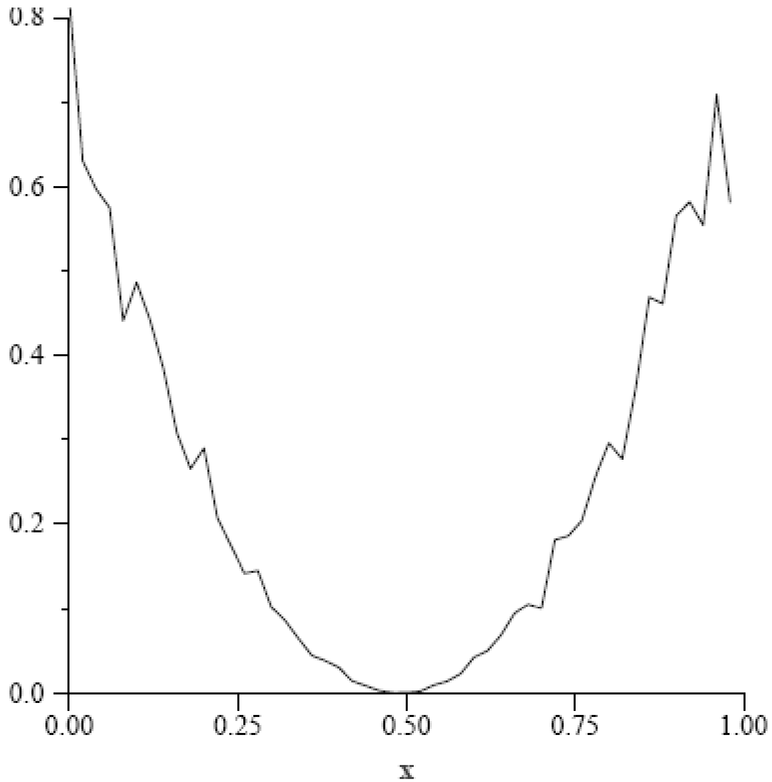



mutation 1

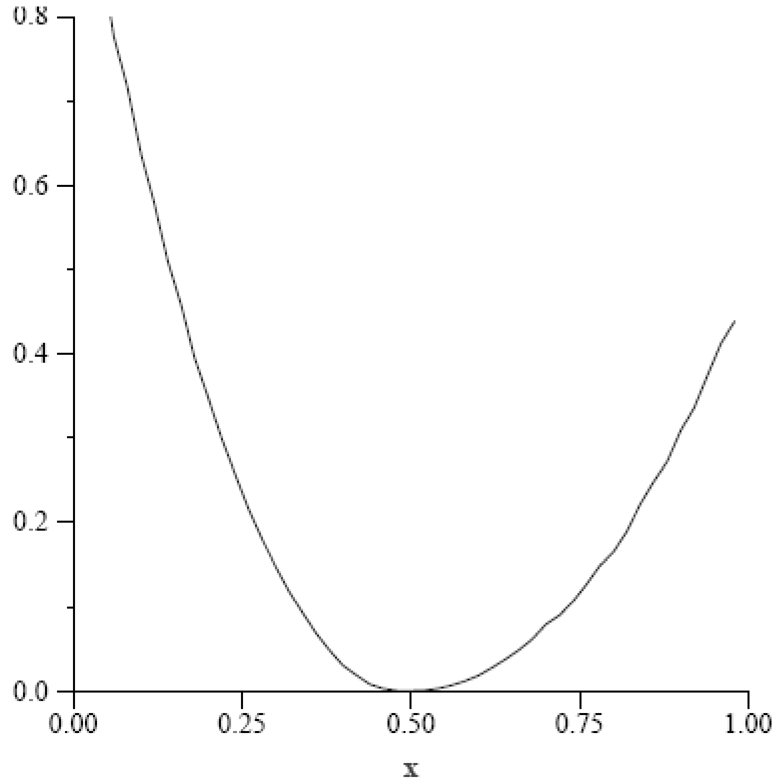



mutation 2
10,000 iterations

# 1D example

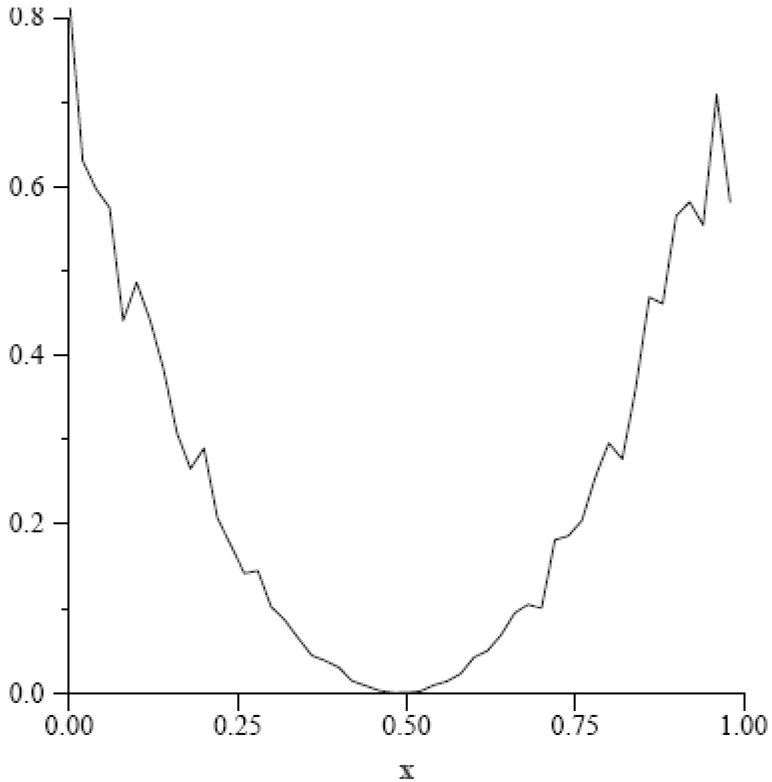



mutation 1

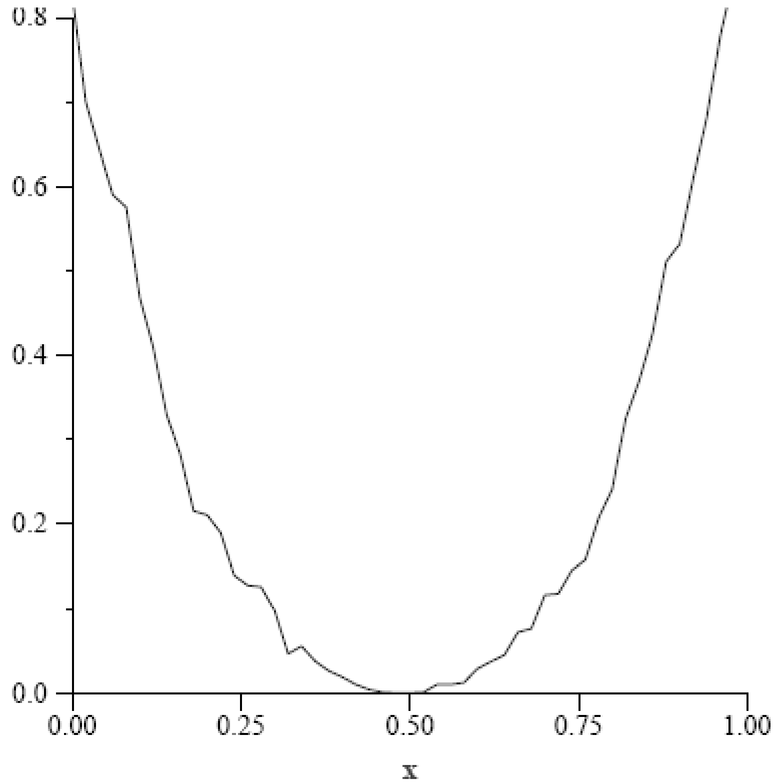



mutation 2
300,000 iterations

# 1D example




mutation 1




90% mutation 2
+ 10% mutation 1

Periodically using uniform mutations increases ergodicity

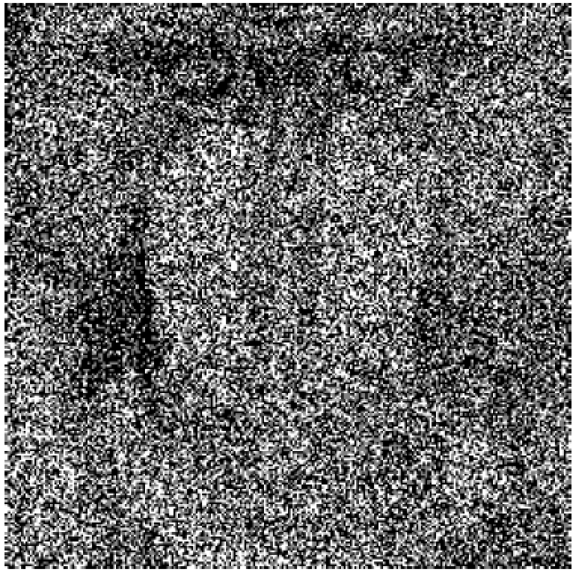
# 2D example (image copy)

```
void makeHistogram(float F[w][h], float histogram[w][h], int mutations)
{
        int i, x0, x1, y0, y1;
        float Fx, Fy, Txy, Tyx, Axy;

        // Create an initial sample point
        x0 = randomInteger(0, w-1);
        x1 = randomInteger(0, h-1);
        Fx = F[x0][x1];

        // In this example, the tentative transition function T simply chooses
        // a random pixel location, so Txy and Tyx are always equal.
        Txy = 1.0 / (w * h);
        Tyx = 1.0 / (w * h);
```

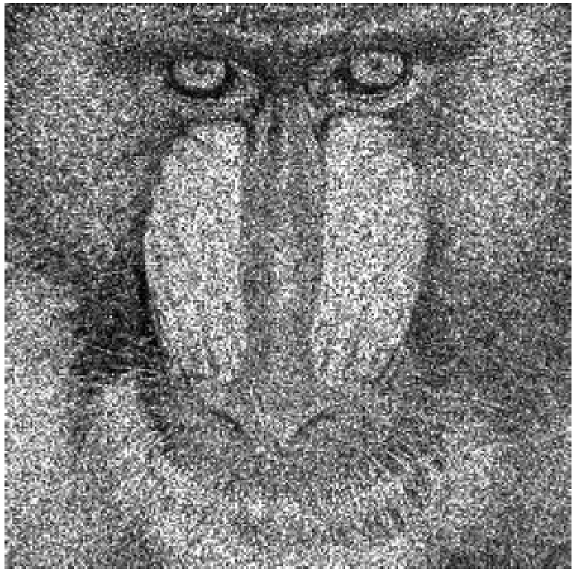
# 2D example (image copy)

```
// Create a histogram of values using Metropolis sampling.
for (i=0; i < mutations; i++) {
        // choose a tentative next sample according to T.
        y0 = randomInteger(0, w-1);
        y1 = randomInteger(0, h-1);
        Fy = F[y0][y1];
        Axy = MIN(1, (Fy * Txy) / (Fx * Tyx)); // equation 2.
        if (randomReal(0.0, 1.0) < Axy) {
                x0 = y0;
                x1 = y1;
                Fx = Fy;
        }
        histogram[x0][x1] += 1;
}
}
```

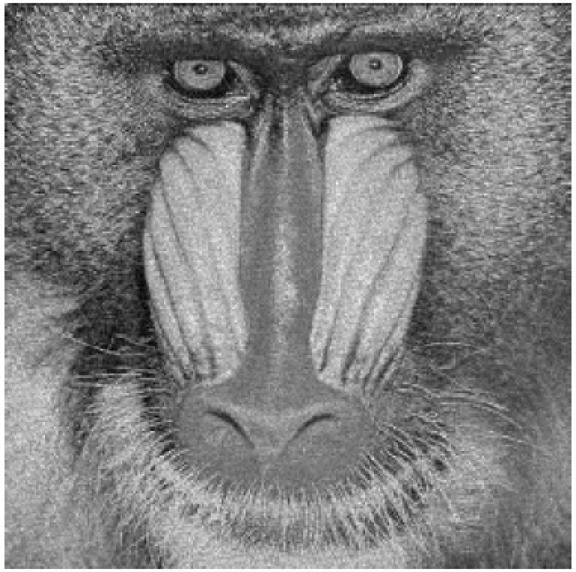
# 2D example (image copy)



1 sample
per pixel



8 samples
per pixel



256 samples
per pixel