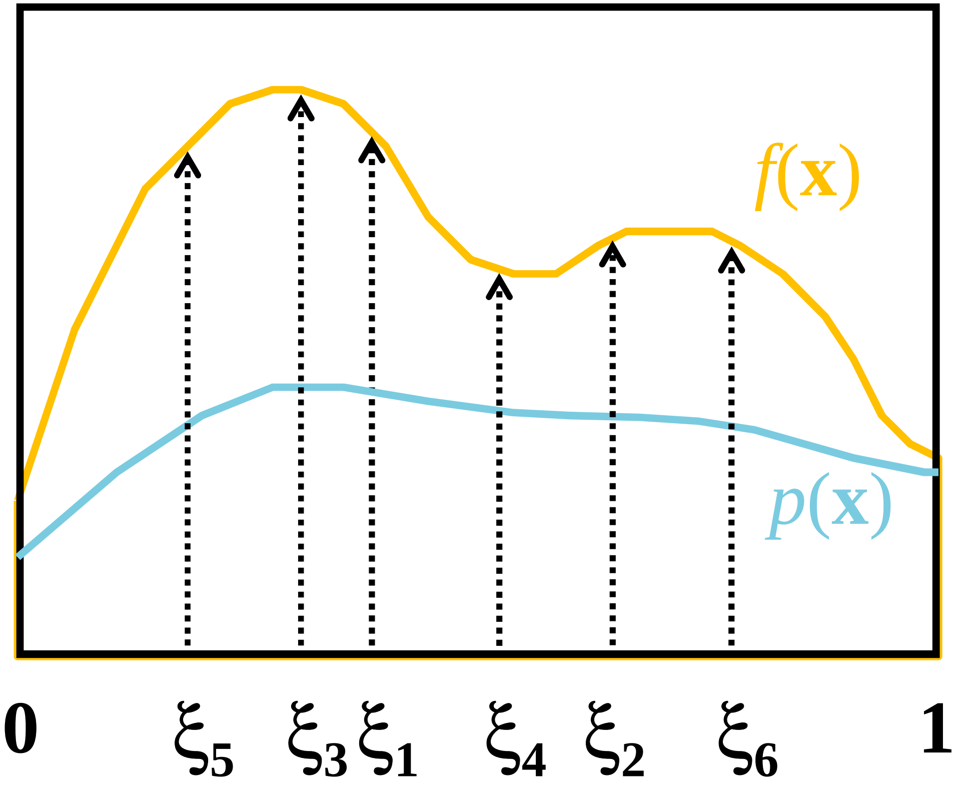
Computer graphics III – Monte Carlo integration II

Jaroslav Křivánek, MFF UK
Jaroslav.Krivanek@mff.cuni.cz

Monte Carlo integration

- General tool for estimating definite integrals



Integral:

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

Monte Carlo estimate I:

$$\langle I \rangle = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\xi_i)}{p(\xi_i)}; \quad \xi_i \propto p(\mathbf{x})$$

Works "on average":

$$E[\langle I \rangle] = I$$

Generating samples from a distribution

Generating samples from a 1D continuous random variable

- Option 1: **Transformation method**

- Option 2: **Rejection sampling**

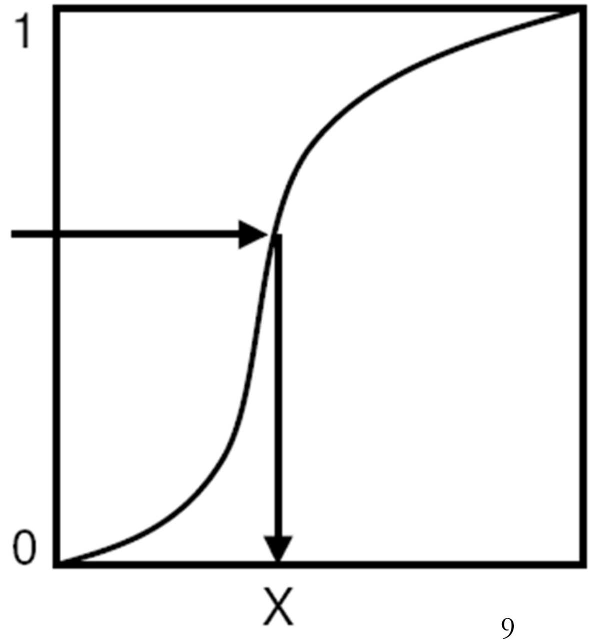
Transformation method

- Consider the random variable *U* from the uniform distribution Uniform(0,1). Then the random variable *X*

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

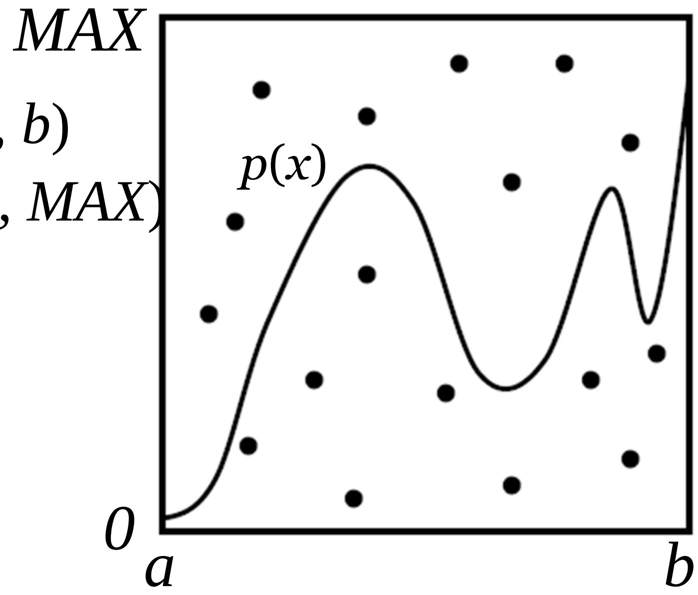
  has the distribution given by the **cdf** *P*.

- To generate samples according to a given pdf *p*, we need to:
  - calculate the cdf $P(x)$ from the pdf $p(x)$
  - calculate the inverse cdf $P^{-1}(u)$



Rejection sampling in 1D

- Algorithm
  - Choose random u_1 from Uniform $R(a, b)$
  - Choose random u_2 from Uniform $R(0, MAX)$
  - Accept the sample only if $p(u_1) > u_2$
  - Repeat until a sample is accepted



- The accepted samples have the distribution given by the pdf $p(x)$

- Efficiency = % of accepted samples
  - Area under the pdf graph / area of the bounding rectangle

Transformation method vs. Rejection sampling

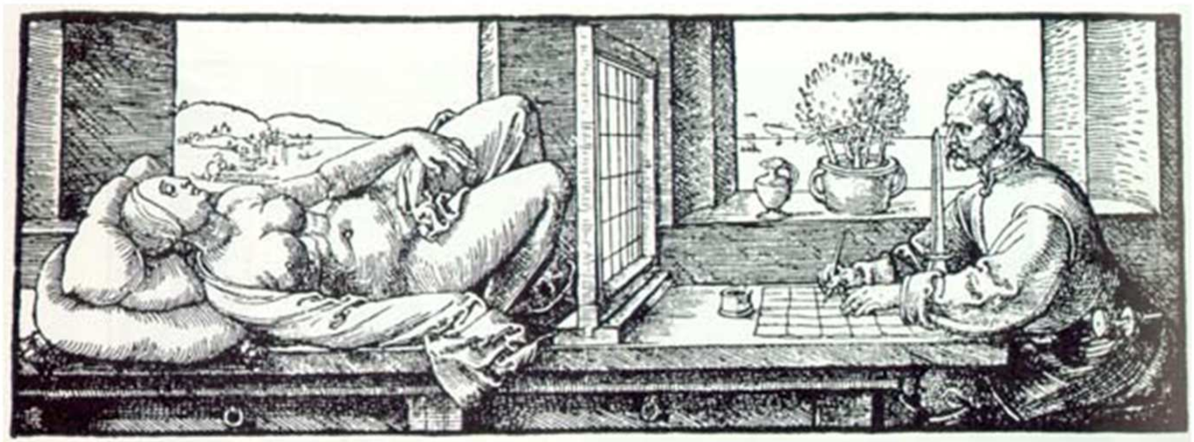
- Transformation method **Pros**
  - Almost always more efficient than rejection sampling (unless the transformation formula $x = P^{-1}(u)$ turns our extremely complex)
  - Has a constant time complexity and the random number count is known upfront
- Transformation method **Cons**
  - May not be feasible (we may not be able to find the suitable form for $x = P^{-1}(u)$), but rejection sampling is always applicable as long as we can evaluate the pdf (i.e. rejection sampling is more general)
- Smart rejection sampling can be very efficient (e.g. the Ziggurat method, see Wikipedia)

Transformation formulas for common cases in light transport

- P. Dutré: **Global Illumination Compendium**, http://people.cs.kuleuven.be/~philip.dutre/GI/

Global Illumination Compendium

The Concise Guide to Global Illumination Algorithms



Albrecht Duerer, *Underweysung der Messung mit dem Zirkel und Richtscheyt* (Nurenberg, 1525), Book 3, figure 67.

Importance sampling from the physically-plausible Phong BRDF

- Ray hits a surface with a Phong BRDF. How do we generate the ray for continuing the light path?

- Procedure

  1. Choose the BRDF component (diffuse reflection, specular reflection, refraction)

  2. Sample the chosen component

  3. Evaluate the total PDF and BRDF

Physically-plausible Phong BRDF

$$f_r^{\text{Phong}}(\omega_{\mathrm{i}} \to \omega_{\mathrm{o}}) = \frac{\rho_{\mathrm{d}}}{\pi} + \frac{n+2}{2\pi}\rho_{\mathrm{s}}\cos^n\theta_{\mathrm{r}}$$

- Where

$$\cos\theta_{\mathrm{r}} = \omega_{\mathrm{o}} \cdot \omega_{\mathrm{r}}$$

$$\omega_{\mathrm{r}} = 2(\omega_{\mathrm{i}} \cdot \mathbf{n})\mathbf{n} - \omega_{\mathrm{i}}$$

- Energy conservation:

$$\rho_d + \rho_s \le 1$$

Selection of the BRDF component

```
pd = max(rhoD.r, rhoD.g, rhoD.b);
ps = max(rhoS.r, rhoS.g, rhoS.b);
pd /= (pd + ps);   // prob of choosing the diffuse component
ps /= (pd + ps);   // prob of choosing the specular comp.

if (rand(0,1) <= pd)
  genDir = sampleDiffuse();
else
  genDir = sampleSpecular(incDir);

pdf = evalPdf(incDir, genDir, pd, ps);
```

Sampling of the diffuse reflection

- Importance sampling with the density $p(\theta) = \cos(\theta) / \pi$
  - θ…angle between the surface normal and the generated ray
  - Generating the direction:

$$\varphi = 2\pi r_1 \qquad \theta = \operatorname{acos}(\sqrt{r_2})$$

$$x = \cos(2\pi r_1)\sqrt{1-r_2}$$
$$y = \sin(2\pi r_1)\sqrt{1-r_2}$$
$$z = \sqrt{r_2}$$

    - r1, r2 … uniform random variates on <0,1)
  - Reference: Dutre, Global illumination Compendium (on-line)
  - Derivation: Pharr & Huphreys, PBRT

sampleDiffuse()

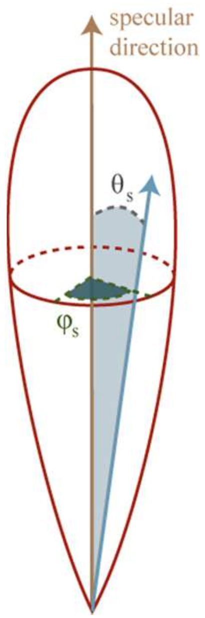
```
// generate spherical coordinates of the direction
float r1 = rand(0,1), r2 = rand(0,1);
float sinTheta = sqrt(1 – r2);
float cosTheta = sqrt(r2);
float phi      = 2.0*PI*r1;
float pdf      = cosTheta/PI;

// convert [theta, phi] to Cartesian coordinates
Vec3 dir (cos(phi)*sinTheta, sin(phi)*sinTheta, cosTheta);

return dir;
```

Sampling of the glossy (specular) reflection

- Importance sampling with the pdf $p(\theta) = (n+1)/(2\pi) \cos^n(\theta)$
  - θ…angle between the ideal mirror reflection of ω_o and the generated ray
  - Formulas for generating the direction:



$$\varphi = 2\pi r_1$$

$$\theta = \operatorname{acos}\left(r_2^{\frac{1}{n+1}}\right)$$

$$x = \cos(2\pi r_1)\sqrt{1 - r_2^{\frac{2}{n+1}}}$$

$$y = \sin(2\pi r_1)\sqrt{1 - r_2^{\frac{2}{n+1}}}$$

$$z = r_2^{\frac{1}{n+1}}$$

- r1, r2 … uniform random variates on <0,1)

sampleSpecular()

```
// build a local coordinate frame with R = z-axis
Vec3 R = 2*dot(N,incDir)*N – incDir;  // ideal reflected direction
Vec3 U = arbitraryNormal(R);          // U is perpendicular to R
Vec3 V = crossProd(R, U);             // orthonormal basis with R and U

// generate direction in local coordinate frame
Vec3 locDir = rndHemiCosN(n);  // formulas form prev. slide, n=phong exp.

// transform locDir to global coordinate frame
Vec3 dir = locDir.x * U + locDir.y * V + locDir.z * R;

return dir;
```

evalPdf(incDir, genDir, pd, ps)

```
return
       pd * getDiffusePdf(genDir) +
       ps * getSpecularPdf(incdir, genDir);
```

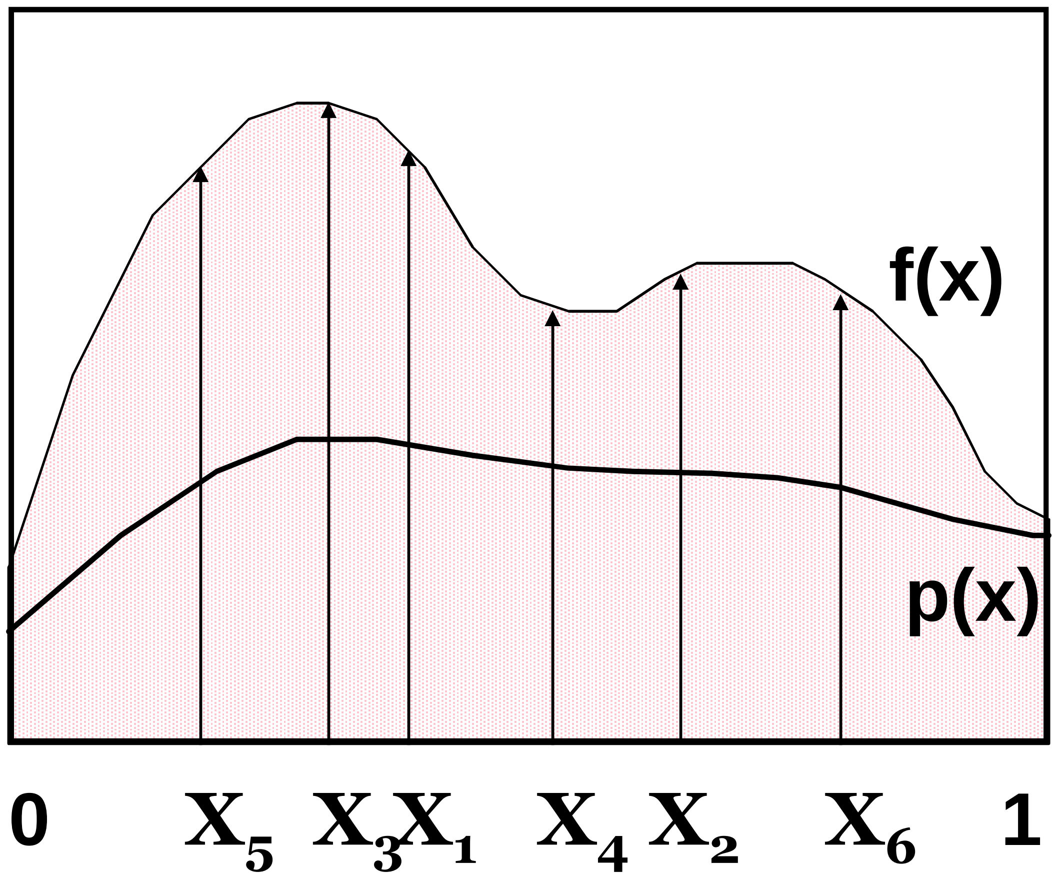
formulas from prev. slides

Variance reduction methods for MC estimators

Variance reduction methods

- **Importance sampling**
  - The most commonly used method in light transport (most often we use BRDF-proportional importance sampling)
- **Control variates**
- **Improved sample distribution**
  - Stratification
  - quasi-Monte Carlo (QMC)

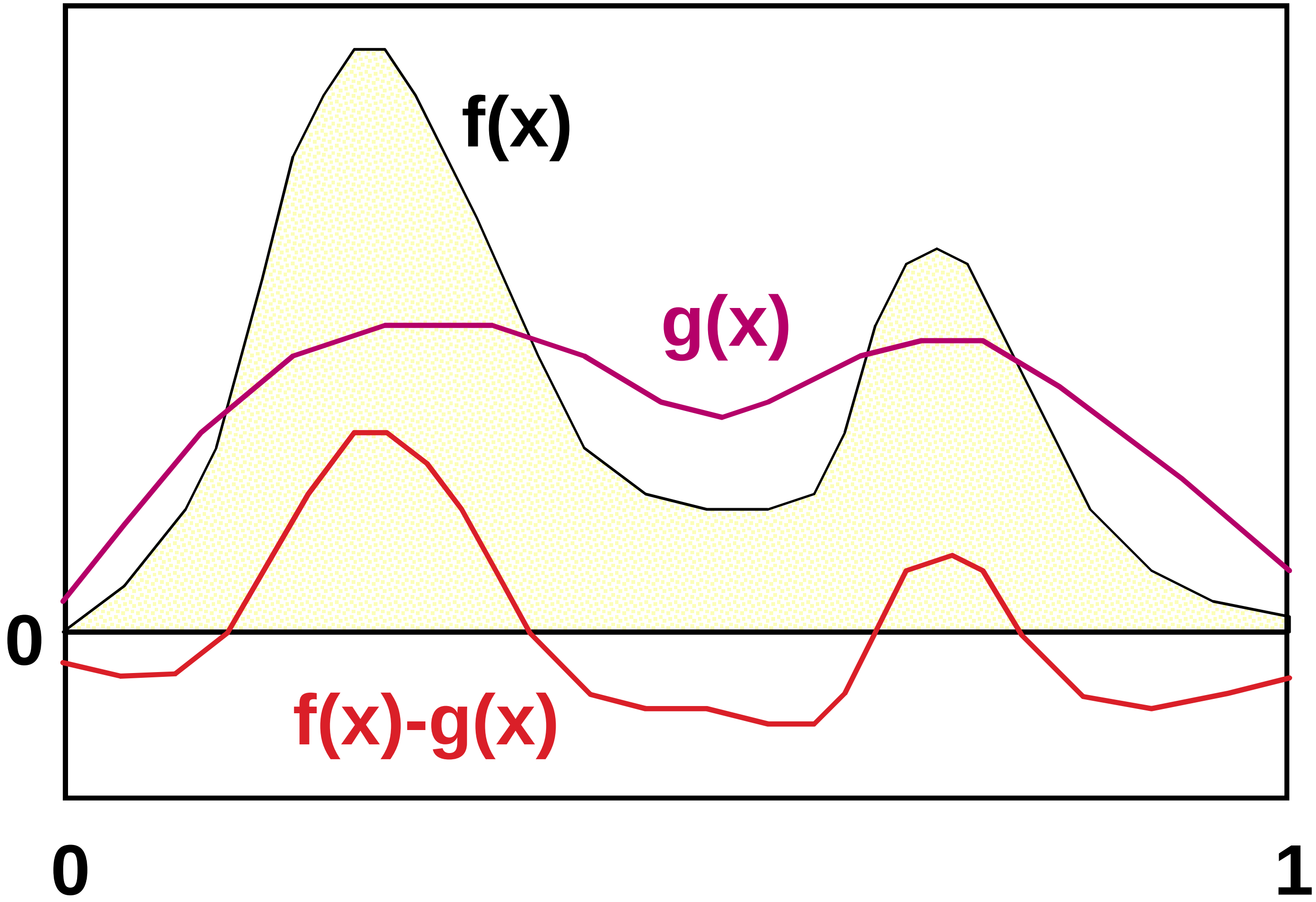
Importance sampling



Importance sampling

- Parts of the integration domain with high value of the integrand *f* are more important
  - Samples from these areas have higher impact on the result
- **Importance sampling** places samples preferentially to these areas
  - I.e. the **pdf** *p* is “similar” to the integrand *f*
- **Decreases variance** while keeping unbiasedness

Control variates



Control variates

Consider a function **g(x)**, that **approximates the integrand** and we can integrate it analytically:

$$I = \int f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \underbrace{\int \left[f(\mathbf{x}) - g(\mathbf{x}) \right] \mathrm{d}\mathbf{x}}_{\text{Numerical integration (MC)}} + \underbrace{\int g(\mathbf{x})\,\mathrm{d}\mathbf{x}}_{\text{We can integrate analytically}}$$

Numerical integration (MC)
Hopefully with less variance than integrating $f(\mathbf{x})$ directly.

We can integrate analytically

Control variates vs. Importance sampling

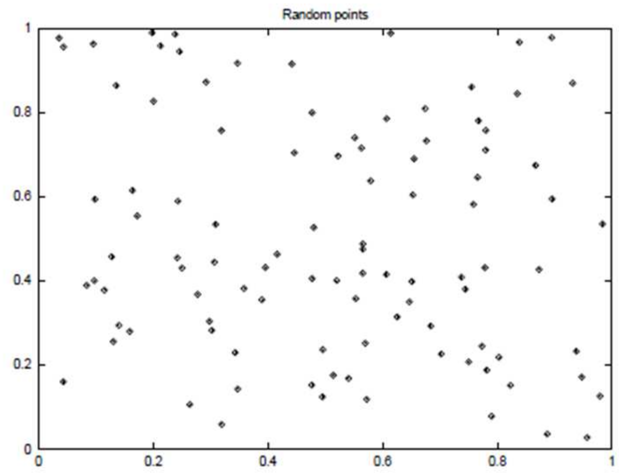
- **Importance sampling**
  - Advantageous whenever the function, according to which we can generate samples, appears in the integrand as a **multiplicative factor** (e.g. BRDF in the reflection equation).

- **Control variates**
  - Better if the function that we can integrate analytically appears in the integrand as an **additive term**.

- This is why in light transport, we almost always use importance sampling and almost never control variates.

Better sample distribution

- Generating independent samples often leads to clustering of samples
  - Results in high estimator variance

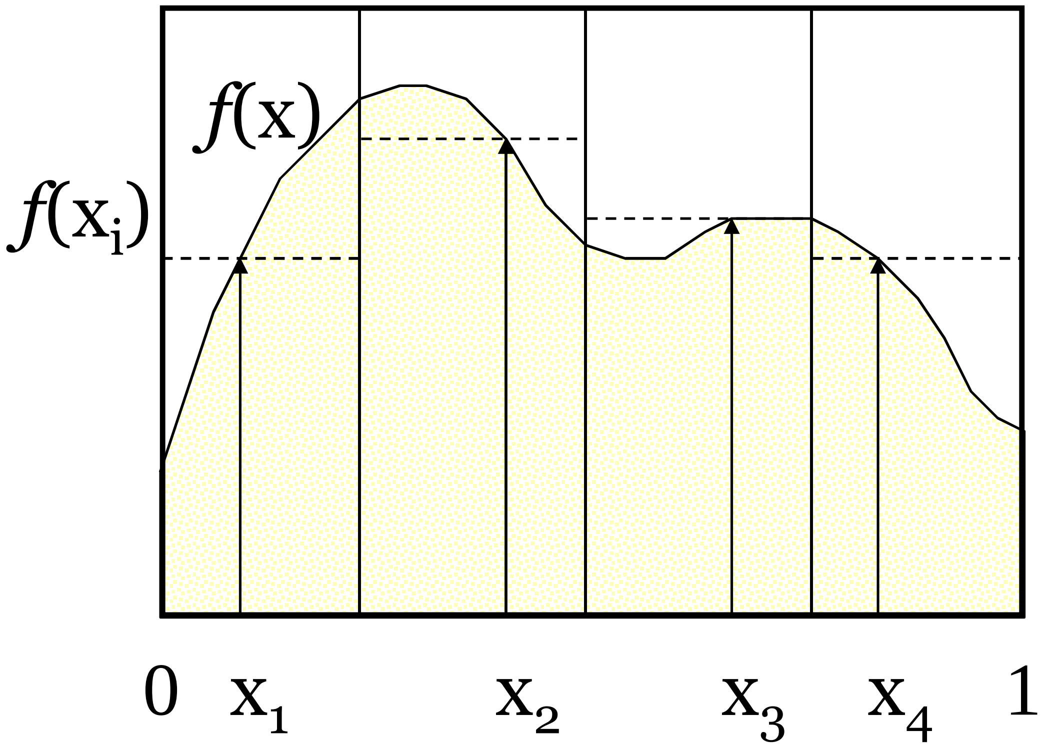


- Better sample distribution => better coverage of the integration domain by samples => lower variance

- Approaches
  - **Stratified sampling**
  - **quasi-Monte Carlo (QMC)**

Stratified sampling

- Sampling domain subdivided into disjoint areas that are sampled independently



Stratified sampling

Subdivision of the sampling domain Ω into N parts Ω_i:

$$I = \int_{\Omega} f(x)\,\mathrm{d}x = \underline{\sum_{i=1}^{N} \int_{\Omega_i} f(x)\,\mathrm{d}x} = \sum_{i=1}^{N} I_i$$

Resulting estimator:

$$\underline{\hat{I}_{\text{strat}} = \frac{1}{N} \sum_{i=1}^{N} f(X_i),} \quad X_i \in \Omega_i$$

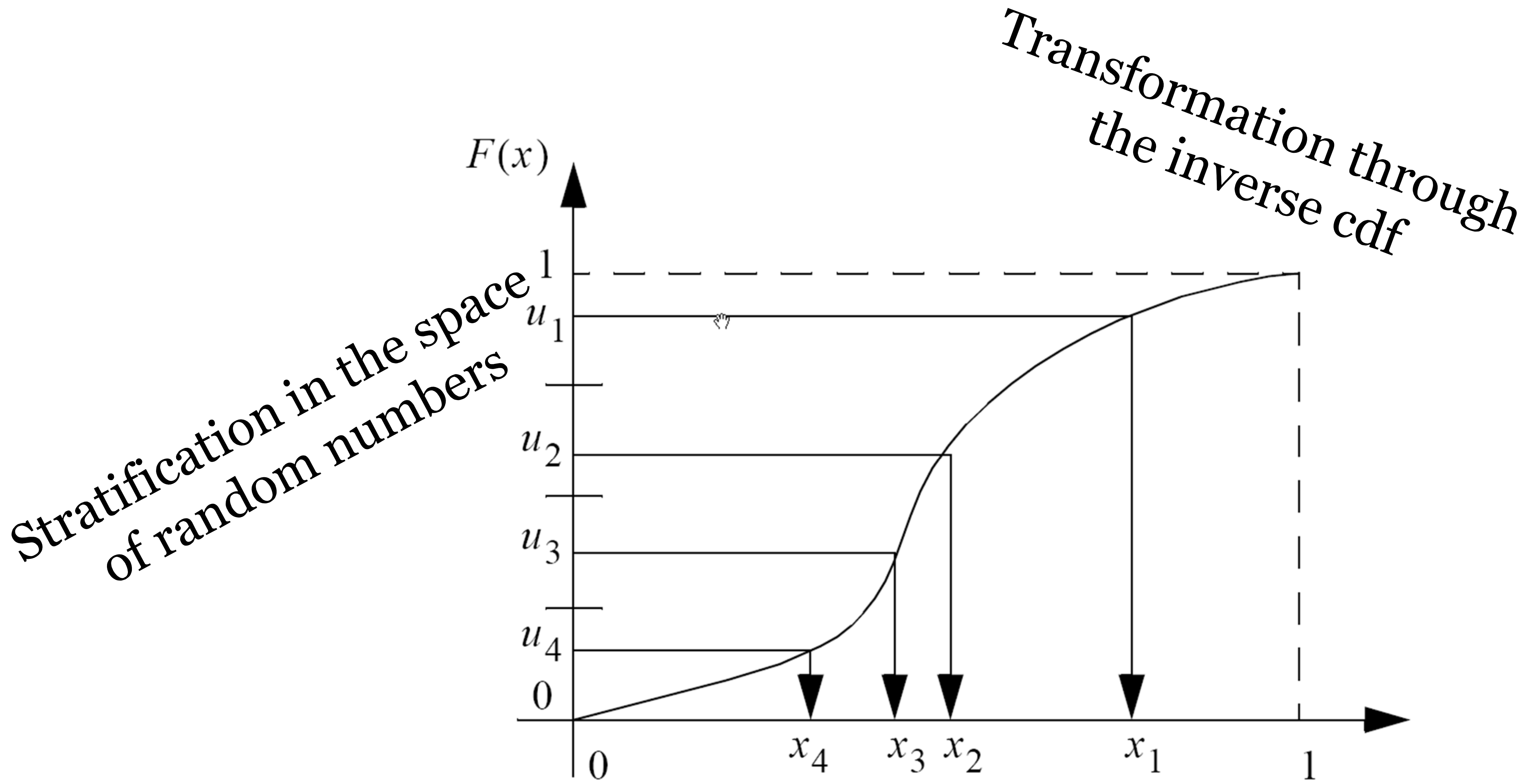
Stratified sampling

- Suppresses sample clustering
- Reduces estimator variance
  - Variance is provably less than or equal to the variance of a regular secondary estimator
- Very effective in low dimension
  - Effectiveness deteriorates for high-dimensional integrands

How to subdivide the interval?

- **Uniform** subdivision of the interval
  - Natural approach for a completely unknown integrand ***f***
- If we know at least roughly the shape of **the integrand *f***, we aim for a subdivision with the lowest possible variance on the sub-domains
- Subdivision of a ***d*-dimensional interval** leads to N^d samples
  - A better approach in high dimension is ***N*-rooks** sampling

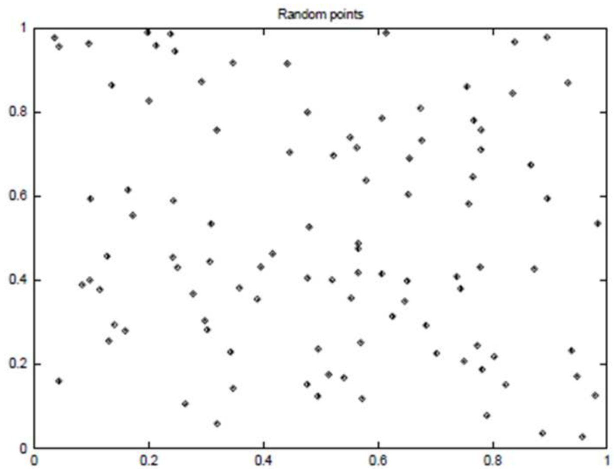
Combination of stratified sampling and the transformation method



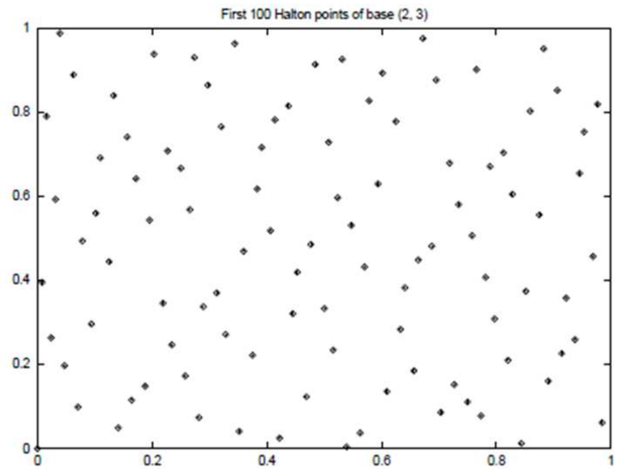
Quasi-Monte Carlo methods (QMC)

- Use of strictly deterministic sequences instead of (pseudo-)random numbers
- Pseudo-random numbers replaced by **low-discrepancy sequences**
- Everything works as in regular MC, but the underlying math is different (nothing is random so the math cannot be built on probability theory)

Discrepancy

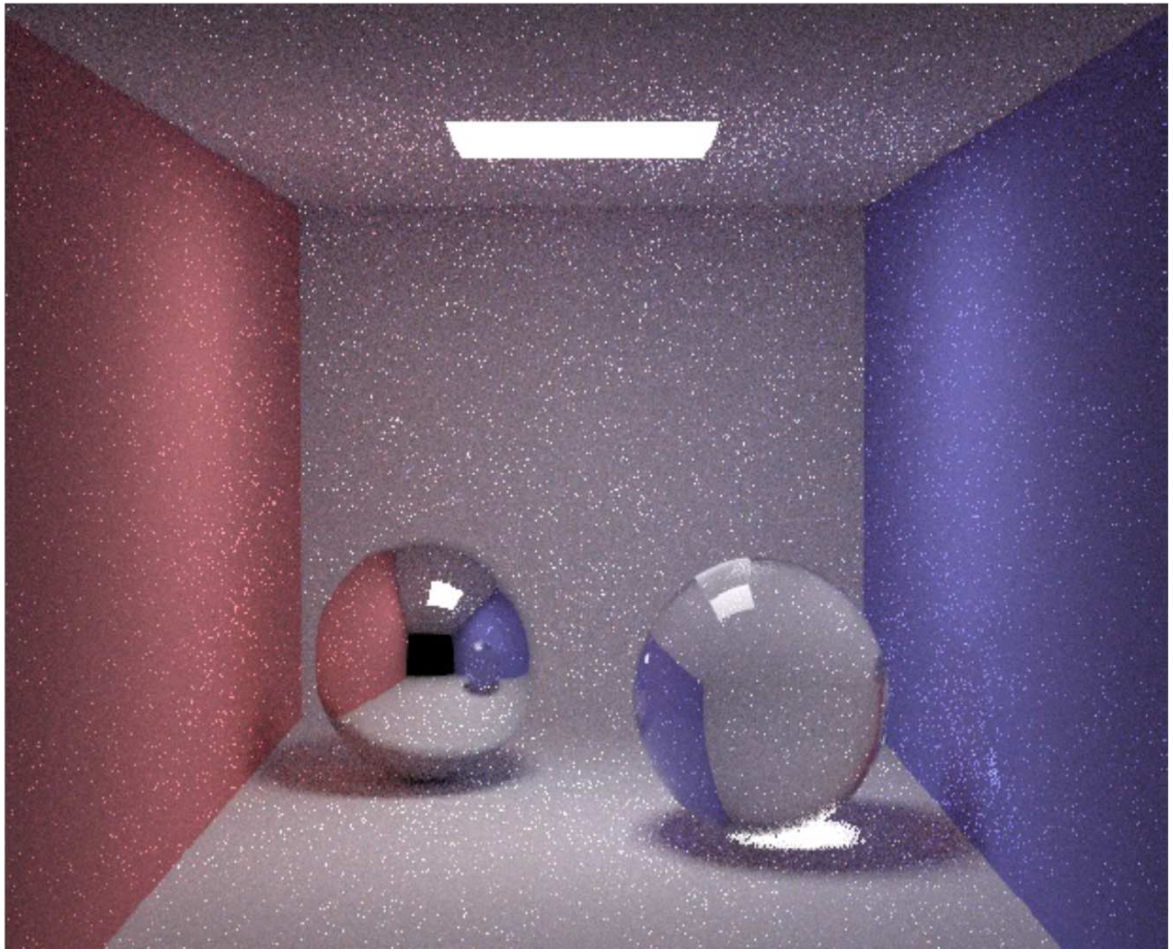


High Discrepancy
(clusters of points)



Low Discrepancy
(more uniform)

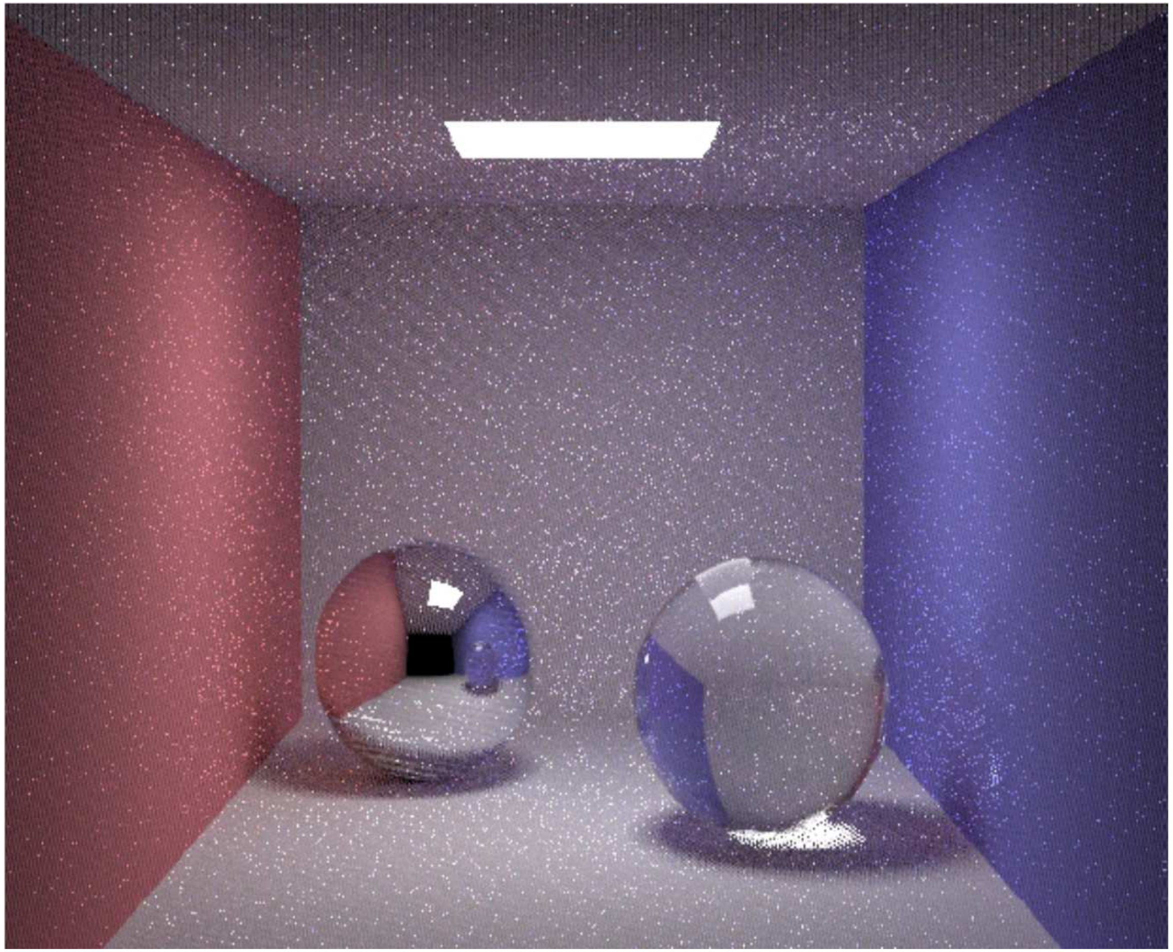
Stratified sampling



Henrik Wann Jensen

10 paths per pixel

Quasi-Monte Carlo



Henrik Wann Jensen

10 paths per pixel

Same random sequence for all pixels

Henrik Wann Jensen

10 paths per pixel