

# Aula 6

## Aula passada

- Método do primeiro momento
- Bolas e urnas
- Lei dos grandes números (fraca e forte)
- Erro e confiança

## Aula de hoje

- Método de Monte Carlo
- Estimando somatórios
- Calculando erro
- Estimando  $\pi$
- Erro de  $\pi$
- Integração de Monte Carlo
- Monte Carlo Ray Tracing

# Método de Monte Carlo

- Classe de algoritmos baseado em amostragem aleatória repetida
  - obter solução aproximada para problemas determinísticos
- Ideia central: grande número de amostras repetidas acabam revelando a solução
  - no limite, amostras dão a solução

**Lei dos grandes números!**

$$M_n = \frac{1}{n} \sum_{i=1}^n X_i \quad P[\lim_{n \rightarrow \infty} M_n = \mu] = 1$$

- Arcabouço teórico para método de Monte Carlo

# Calculando Soma

- Calcular o valor de um somatório (problema da aula 1)
  - $N$  (número de parcelas) é muito grande
  - calcular valor de cada parcela é fácil

$$G_N = \sum_{i=1}^N g(i)$$

- Como usar aleatoriedade para resolver (aproximar) este somatório?
- Usando valor esperado!



- Seja  $X$  uma v.a. uniforme em  $[1, N]$ 
  - $P[X = i] = 1/N$  para  $i=1, \dots, N$

$$E[g(X)] = \sum_{i=1}^N P[X = i] g(i) = \frac{1}{N} \sum_{i=1}^N g(i) = \frac{G_N}{N}$$

# Calculando Soma

- Logo, temos que  $G_N = N E[g(X)]$
- Podemos agora estimar  $E[g(X)]$ . Como?

## Gerando amostras, fazendo a média!

- Seja  $X_i$  sequência iid de v.a. uniforme  $[1, N]$ 
  - escolher um valor para  $n$  (número de amostras)

$$M_n = \frac{1}{n} \sum_{i=1}^n g(X_i) \quad \longleftarrow \text{Média amostral com } n \text{ amostras}$$

- Temos que  $E[M_n] = E[g(X)]$ , para todo  $n$
- $M_n \rightarrow E[g(X)]$  quando  $n \rightarrow \infty$  (pela lei dos grds números)
- $M_n$  é estimador de  $E[g(X)]$ , logo  $N * M_n$  é estimador para  $G_N$

# Arestas no Facebook

- Quantas arestas tem a rede de amizade do Facebook?
  - função indicadora de aresta,  $g(i, j) = 1$  se existe aresta entre perfil  $i$  e  $j$
  - número de perfis  $n_p = 2 \cdot 10^9$

$$T = \sum_{i=1}^{n_p} \sum_{j=i+1}^{n_p} g(i, j) \quad \longleftarrow T = \text{número de arestas}$$

**Problema: somatório tem  $\sim 10^{18}$  termos!**

- Como construir um método de Monte Carlo para obter um valor aproximado para  $T$ ?

# Arestas no Facebook

- Seja  $Z = (i, j)$  uma v.a. uniforme no conjunto de pares com  $n_p$  perfis

- $N = n_p(n_p - 1)/2 \rightarrow$  número total de pares

$$E[g(Z)] = \sum_{i=1}^{n_p} \sum_{j=i+1}^{n_p} \frac{1}{N} g(i, j) = \frac{1}{N} T$$

- Seja  $Z_i$  sequência iid de v.a. uniforme sobre pares

$$M_n = \frac{1}{n} \sum_{i=1}^n g(Z_i) \quad \leftarrow \text{Média amostral com } n \text{ amostras, estima } E[g(Z)]$$

- Logo  $T$  é aproximado por  $M_n * N$

# Jogando Paciência

- Qual é a fração de vezes que um algoritmo determinístico  $A$  vence o jogo de paciência?
  - vencer depende apenas da permutação das cartas do baralho (não há aleatoriedade)
- Seja  $s$  uma permutação das cartas
  - $f_A(s) = 1$  se algoritmo  $A$  vence com permutação  $s$ , 0 c.c.
- $N =$  número total de permutações das cartas,  $N = 52!$

$$F_A = \frac{1}{N} \sum_{s=1}^N f_A(s) \quad \leftarrow \text{Fração de vezes que algoritmo } A \text{ vence}$$

**Problema: somatório tem 52! termos!**

- *Monte Carlo to the rescue!*

# Jogando Paciência

- Seja  $S$  uma v.a. uniforme em  $[1, N]$ 
  - uniforme entre todas possíveis permutações das cartas

$$E[f_A(S)] = \sum_{s=1}^N \frac{1}{N} f_A(s) = \frac{1}{N} \sum_{s=1}^N f_A(s) = F_A$$

- Valor esperado já é a fração que queremos!
- Seja  $S_i$  sequência iid de v.a. uniforme em  $[1, N]$ 
  - $S_i$  é uma permutação das cartas

$$M_n = \frac{1}{n} \sum_{i=1}^n f_A(S_i) \quad \longleftarrow \text{Média amostral da fração de vezes que algoritmo vence}$$

- Logo,  $F_A$  é aproximado por  $M_n$



# Vantagens e Desvantagens



• Trocar  $G_N = \sum_{i=1}^N g(i)$  por  $M_n = \frac{1}{n} \sum_{i=1}^n g(X_i)$

• **Quando esta ideia é boa?**

- Comparação entre  $N$  e  $n$
- Se  $N$  for pequeno, ideia não é boa
  - podemos calcular  $G_N$  diretamente
- Se calcular  $g(\cdot)$  é muito caro, boa ideia mesmo quando  $N$  pequeno
- Se  $g(i)$  for muito “errática”, ideia não é boa
  - ex. um valor de  $g(i)$  é maior que todo o resto da soma
  - estimador pode ser muito ruim se  $n$  não for muito grande
- Qualidade da aproximação depende de  $N, n, g()$

# Calculando o Erro

- Podemos usar Chebyshev para calcular  $n$ 
  - para precisão  $\epsilon$  e confiança  $\beta$ , temos

$$P[M_n \in [\mu - \epsilon, \mu + \epsilon]] > 1 - \frac{\sigma^2}{\epsilon^2 n} = \beta$$

- onde  $\mu$  e  $\sigma^2$  são o valor esperado e variância da v.a. que será aproximada pelo estimador

$$\mu = E[g(X)] \quad \sigma^2 = \text{Var}[g(X)]$$

- onde  $g(\cdot)$  é geralmente uma função indicadora (para contar coisas) e  $X$  é uma v.a. uniforme nos valores que  $g$  pode assumir
- **Problema:** muitas vezes não sabemos  $\sigma^2$ 
  - temos que estimar com o estimador!

# Erro da Paciência

- Seja  $F_A = 0.1$  fração de vezes que algoritmo A vence
  - $S$  é v.a. uniforme nas permutações
  - $\mu = E[f_A(S)] = 0.1$ ,  $\sigma^2 = \text{Var}[f_A(S)] = 0.1*0.9 = 0.09$
- Supor  $\epsilon = 10^{-4}$  e  $\beta = 0.99$

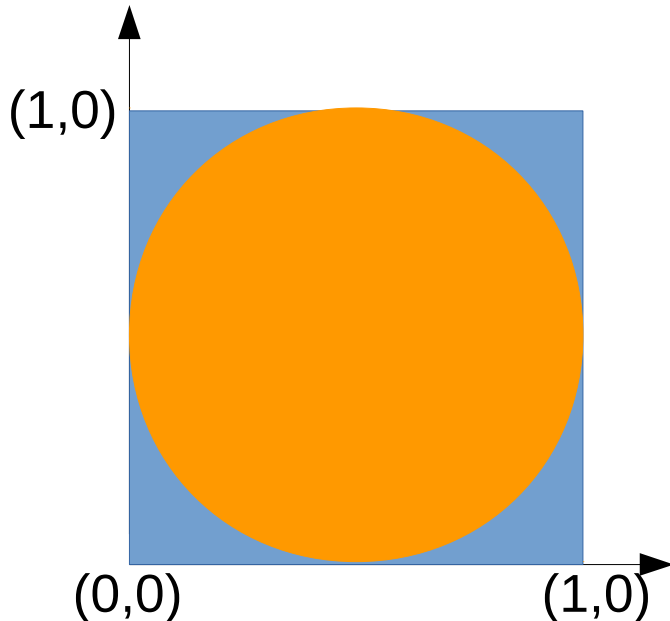
$$P[M_n \in [\mu - \epsilon, \mu + \epsilon]] > 1 - \frac{\sigma^2}{\epsilon^2 n} = \beta$$

$$1 - \frac{0.09}{(10^{-4})^2 n} = 0.99 \rightarrow n = 9 * 10^8$$

- Muito, muito menor do que  $52! \sim 10^{68}$

# Calculando $\pi$

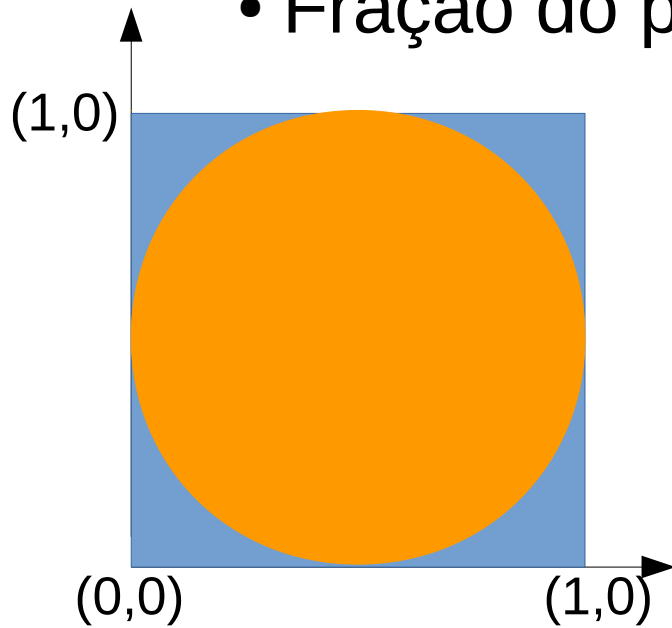
- Como estimar o valor de  $\pi$ ?
  - ou qualquer outro valor que tenha relação com geométrica
- **Ideia**
  - Escrever  $\pi$  como relação entre áreas
  - Usar Monte Carlo para estimar relação



- $A_q = \text{Área do quadrado} = 1$
- $A_c = \text{Área do círculo} = \pi r^2 = \pi/4$
- $\pi = 4 * A_c / A_q$
- Estimar  $A_c / A_q$

# Calculando $\pi$

- Ideias para estimar  $A_c / A_q$  ?
- Gerar  $n$  pontos uniforme no quadrado
- Fração do pontos que estão dentro do círculo!



- Seja  $X$  e  $Y$  duas v.a. uniformes contínuas em  $[0, 1]$
- Seja  $g(x, y)$  indicadora do ponto  $(x, y)$  estar dentro do círculo
  - $g(x, y) = 1$  se  $(x-0.5)^2 + (y-0.5)^2 \leq 1$
- Temos  $E[g(x, y)] = A_c / A_q = \pi/4$
- Como estimar  $E[g(x, y)]$  ?

# Calculando $\pi$

- Seja  $X_i, Y_i$  sequência iid uniforme em  $[0,1]$ 
  - v.a. contínua

$$M_n = \frac{1}{n} \sum_{i=1}^n g(X_i, Y_i) \quad \longleftarrow \text{Fração de pontos que estão dentro do círculo}$$

- $M_n$  converge para  $\pi/4$  (pela lei dos grds números)
- Logo  $\pi$  pode ser estimado por  $4 * M_n$

# O Erro de $\pi$

- Podemos usar Chebyshev para calcular  $n$ 
  - para precisão  $\epsilon$  e confiança  $\beta$ , temos

$$P[M_n \in [\mu - \epsilon, \mu + \epsilon]] > 1 - \frac{\sigma^2}{\epsilon^2 n} = \beta$$

- $\mu = E[g(X, Y)] = P[g(X, Y) = 1] = \pi/4$
- $\sigma^2 = \text{Var}[g(X, Y)] = \pi/4 (1 - \pi/4)$
- Não sabemos  $\pi$  mas  $\sigma^2$  de uma v.a. indicadora tem valor máximo quando  $p = 1/2 \rightarrow \sigma^2 = 1/4$
- Supor  $\epsilon = 10^{-6}$  e  $\beta = 0.99$ 
  - então temos que  $n = 2.5 * 10^{16}$

# MSE e SEM

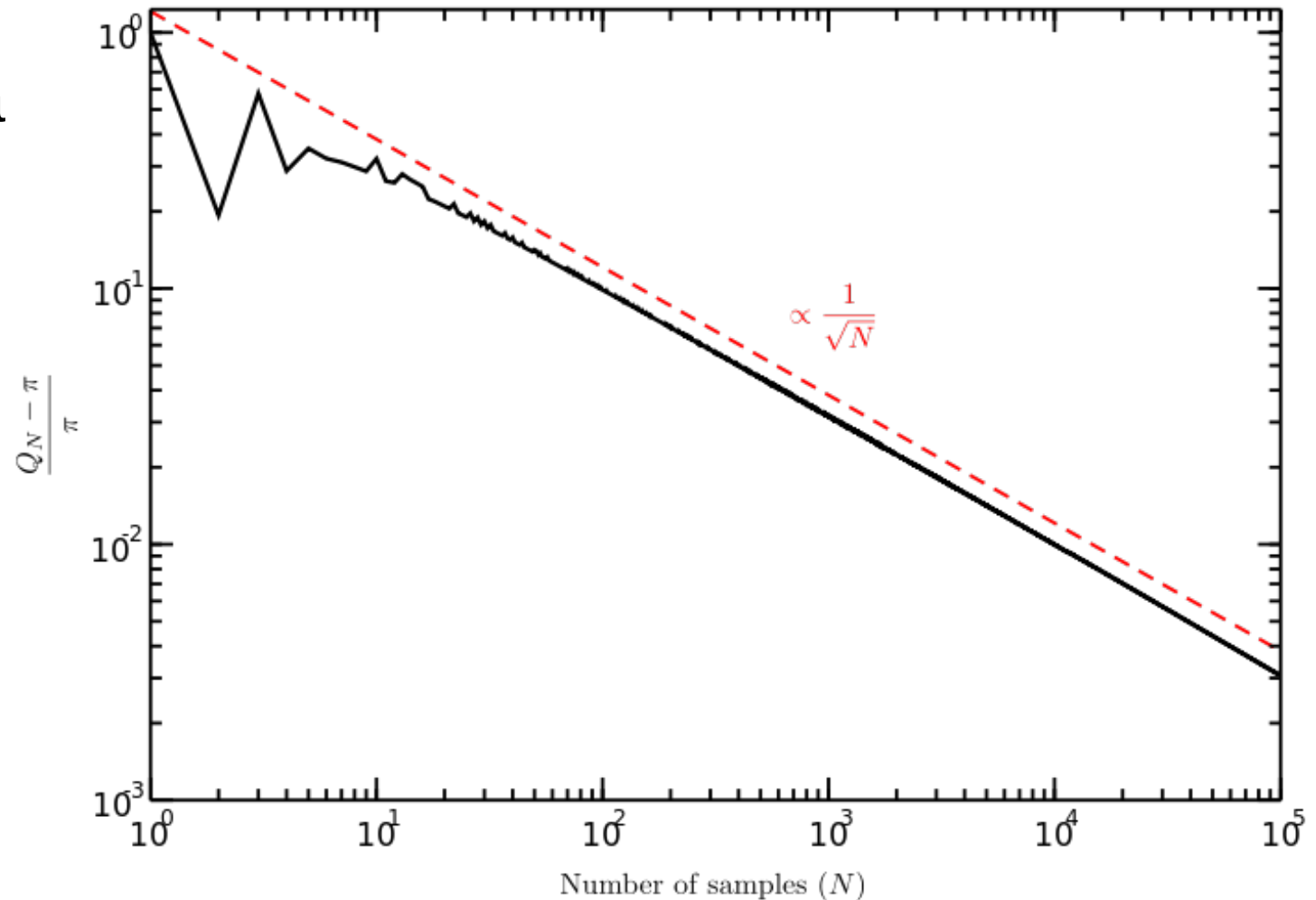
- Mean Squared Error (MSE)
  - medida clássica para erro de preditores ou estimadores
  - $MSE(\phi') = E_{\phi'}[(\phi' - \phi)^2]$ , onde  $\phi'$  é o estimador e  $\phi$  o valor a ser estimado
- Seja  $M_n$  (média amostral) o estimador para o valor  $\mu$ 
  - $MSE(M_n) = \text{Var}(M_n) = \sigma^2/n$
- Standard Error of the Mean (SEM)
  - medida de erro relativo
  - $SEM(M_n) = \sigma/\text{sqrt}(n) = \text{sqrt}(MSE(M_n))$
- Para qualquer  $M_n$ , sempre decresce como  $\text{sqrt}(n)$ 
  - não é muito rápido!



# SEM de $\pi$

- $SEM(M_n) = \sigma/\sqrt{n}$
- Erro relativo =  $|4M_n - \pi| / \pi$

- Teoria e prática estão bem de acordo!
- Repare  $1/\sqrt{n}$  é devagar !

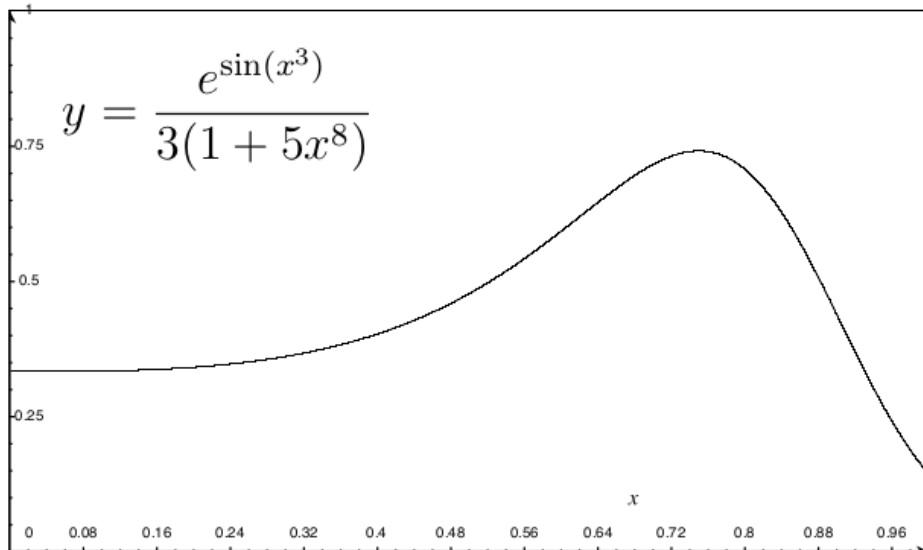


# Integração Numérica

- Encontrar a integral definida de uma função

$$\int_{x=a}^{x=b} f(x) dx$$

- **Problema:** função pode não ser integrável (ou muito difícil de integrar)!



- **Solução:** integração numérica
- Conjunto de algoritmos para calcular valor da integral
  - muito usado na física, química, engenharia, etc

- Abordagem via Monte Carlo é uma classe de algoritmos
  - *Monte Carlo integration*

# Integração de Monte Carlo

- Generalização da ideia de computar o valor de  $\pi$
- Calcular razão entre área de “baixo da curva” (integral) e de um “quadrado” (limites de integração)
- Estimar este valor usando amostras uniformes
- Supor  $0 \leq f(x) \leq 1$ , para  $x$  em  $[0, 1]$

$$I = \int_{x=0}^{x=1} f(x) dx$$

- Definir função indicadora para ponto de baixo da curva
  - $g(x, y) = 1$  se  $f(x) \leq y$ , e zero c.c.
- Seja  $X, Y$  v.a. contínuas uniformes em  $[0, 1]$

# Integração de Monte Carlo

- Valor esperado

$$E[g(X, Y)] = \int_{x=0}^{x=1} \int_{y=0}^{y=1} f_{XY}(x, y) g(x, y) dx dy$$

- $f_{XY}(x, y) = f_X(x) f_Y(y) = 1$  (densidade conjunta de duas v.a. contínuas uniformes e independentes)
- $E[g(X, Y)] = \text{área embaixo da curva} / \text{área do quadrado } [0,1] \times [0,1] = I$
- Como estimar  $E[g(X, Y)]$  ?
- Seja  $X_i, Y_i$  sequência iid de uniformes em  $[0, 1]$

$$M_n = \frac{1}{n} \sum_{i=1}^n g(X_i, Y_i) \quad \longleftarrow \text{Fração de pontos que estão baixo da curva}$$

# Integração de Monte Carlo

- Método alternativo: relacionar valor esperado diretamente com valor da integral

$$I = \int_{x=0}^{x=1} f(x) dx$$

- Seja  $X$  v.a. contínua uniforme em  $[0, 1]$

$$E[f(X)] = \int_{x=0}^{x=1} f_X(x) f(x) dx = \int_{x=0}^{x=1} f(x) dx = I$$

- Pois  $f_X(x) = 1$  (densidade da v.a. uniforme em  $[0,1]$ )
- Como estimar  $E[f(X)]$  ?
- Seja  $X_i$  sequência iid uniforme em  $[0, 1]$

Converge para valor esperado  
em  $n$  (lei dos grds números)

$$M_n = \frac{1}{n} \sum_{i=1}^n f(X_i)$$

# Generalização

- Funciona para qualquer limite de integração, e qualquer número de dimensões
  - grande vantagem da abordagem Monte Carlo

$$I = \int_{\omega} f(\vec{x}) d\vec{x} \quad V = \int_{\omega} d\vec{x}$$

- onde  $x$  é um ponto em um espaço de  $k$  dimensões,  $\omega$  um “pedaço” fechado deste espaço, e  $V$  é o volume de  $\omega$
- Seja  $X_i$  uma sequência iid de v.a. uniforme no espaço  $\omega$ 
  - $X_i$  tem  $k$  dimensões

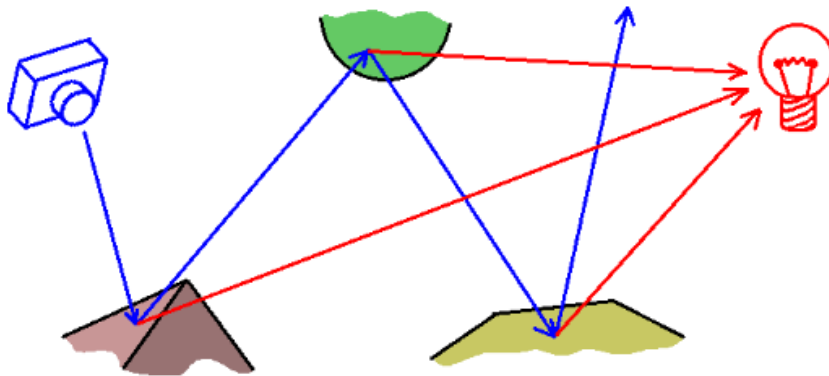
- Temos

$$M_n = \frac{1}{n} \sum_{i=1}^n f(\vec{X}_i) \quad \text{converge para } I/V \text{ quando } n \rightarrow \text{infinito}$$

# Monte Carlo *Ray Tracing*

- Integração de Monte Carlo em Computação Gráfica
- **Problema:** determinar cor (intensidade de luz) em um pixel em uma cena construída

Integral formulation (“rendering equation”)



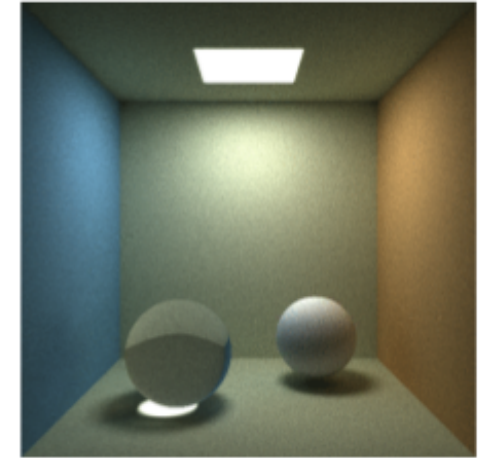
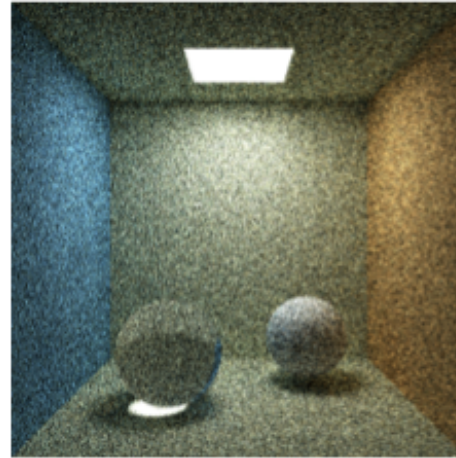
$$R = \int_{\Omega} L(y) dy$$

← Resolvida numericamente usando método de Monte Carlo

- Integrate over all possible light paths  $y$  ← Muitos, muitos caminhos!

# Monte Carlo *Ray Tracing*

$$\int_{\Omega} L(y) dy = E \left[ \frac{L(Y)}{\text{pdf}[Y]} \right]$$



## Advantage of Monte-Carlo

- Good compromise between cost and precision:

$$\epsilon = O\left(1/\sqrt{N}\right)$$

- in contrast, integration by e.g. trapezoid rule has an error of  $\epsilon = O(1/N^{2/d})$ , for  $d$  dimensions

## Disadvantage of Monte-Carlo

- Output is random