Mestrado Integrado
Engenharia Informática

Integração de Monte Carlo

Visualização e Iluminação II

Luís Paulo Peixoto dos Santos

Motivação

$$L(p \to \omega_r) = L_e(p \to \omega_r) + \int_{\Omega_s} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i) \cos(N_p, \omega_i) d\omega_i$$

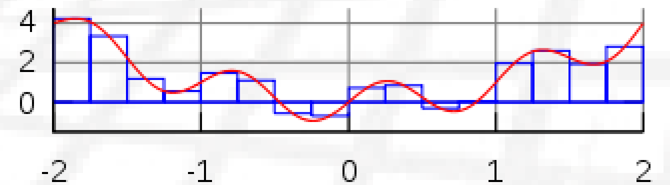
- A equação de rendering não tem solução analítica
- Os algoritmos de rendering calculam soluções numéricas aproximadas
- São seleccionadas N amostras (direcções ω_i) do domínio de integração (semiesfera)
- A contribuição de cada amostra é adicionada ao resultado final com um determinado peso, w_i: o integral transforma-se numa soma

$$L(p \to \omega_r) = L_e(p \to \omega_r) + \sum_{i=0}^{N-1} w_i f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i) \cos(N_p, \omega_i)$$

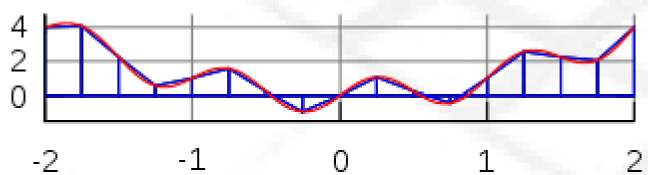
- A selecção de quais e quantas as direcções a amostrar e qual o peso w_i de cada amostra determinam a qualidade e desempenho dos algoritmos de iluminação global

Quadratura numérica determinística

Regra do rectângulo

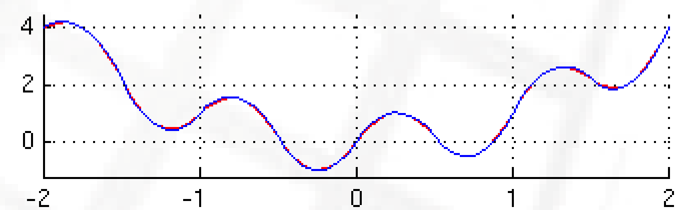


$$F(f(.), x_{i-1}, x_i) = (x_i - x_{i-1})\, f\left(\frac{x_{i-1} + x_i}{2}\right)$$



Regra do trapézio

$$F(f(.), x_{i-1}, x_i) = (x_i - x_{i-1})\, \frac{f(x_{i-1}) + f(x_i)}{2}$$



Regra de Simpson

$$F(f(.), x_{i-1}, x_i) = \frac{(x_i - x_{i-1})}{6}\left[f(x_{i-1}) + 4 f\left(\frac{x_{i-1} + x_i}{2}\right) + f(x_i) \right]$$

Quadratura numérica determinística – Exemplo: método dos rectângulos

- O domínio é **uniformemente** subdividido em N subdomínios de extensão h, [a_i, a_i+h[
- Para cada subdomínio é calculado um **estimador primário** dado pelo produto da extensão do domínio e o valor da função no centro do subdomínio
- O **estimador final** é a **média aritmética** dos estimadores primários

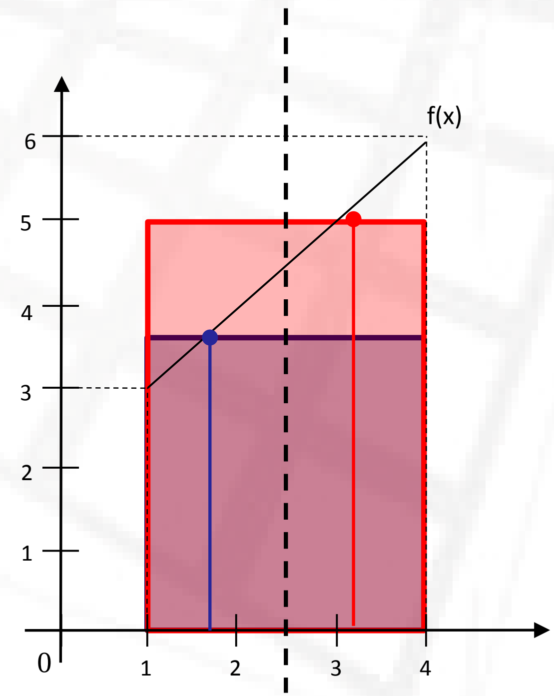
$$I = \int_1^4 (x+2)dx = \left.\frac{x^2}{2}\right|_1^4 + 2x\Big|_1^4 = 13{,}5$$

$$N = 2; h = 1.5; a_o = 1; a_1 = 2.5$$

$$f(x_0) = f(1.75) = 3.75 \quad \langle I_{x0} \rangle = 3 * f(1.75) = 11.25$$

$$f(x_1) = f(3.25) = 5.25 \quad \langle I_{x1} \rangle = 3 * f(3.25) = 15.75$$

$$< I >= \frac{1}{2} * (11.25 + 15.75) = 13.5$$



Integração Numérica Determinística

- Desvantagens:
  - A localização das amostras no domínio está perfeitamente definida e não varia
    Isto implica que há pontos do domínio da função que não contribuem NUNCA para o valor do estimador do integral
    E se o valor da função nestes pontos variar significativamente?

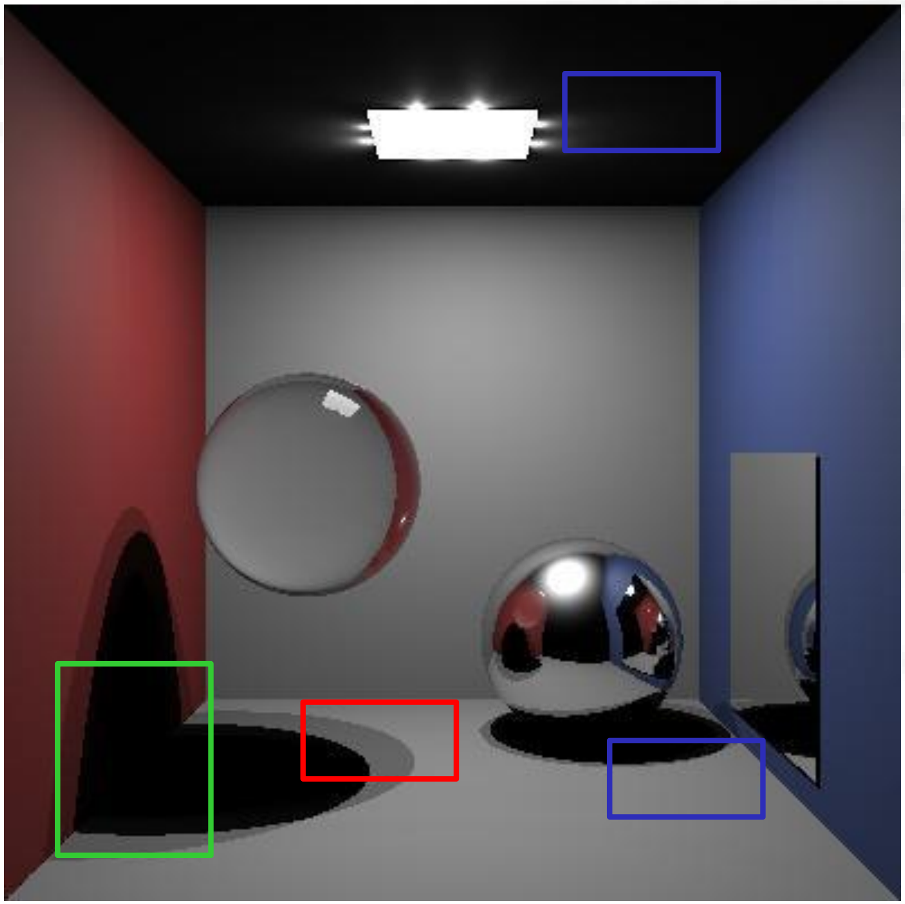
    NOTA: no contexto do rendering isto significa que as direcções a amostrar na semiesfera estão pré-determinadas o que pode levar a ignorar direcções importantes

  - Estes métodos implicam N amostras por dimensão.
    Uma função k-dimensional requer portanto N^k amostras

    NOTA: O número de dimensões k num algoritmo de rendering é infinito

Ray Tracing Clássico

A selecção determinística de um número limitado de direcções de amostragem resulta em:



- Descontinuidades abruptas, principalmente as resultantes de efeitos de iluminação
- *aliasing*
- Simulação de um número limitado de efeitos de iluminação

Integração de Monte Carlo: Formulação

- Objectivo: calcular o integral de $f(x)$ no domínio *D*:

$$I = \int_D f(x)dx$$

- O valor esperado de uma função aleatória é o valor médio da função, calculado sobre um conjunto de valores x pertencentes ao domínio da função e distribuídos com probabilidade *p(x):*

$$E(f(x)) = \int_D f(x)p(x)dx$$

- O valor esperado de uma função é aproximado pela média de um número de amostras aleatórias - converge para o valor correcto quando o número de amostras se aproxima de infinito (*Lei dos Grandes Números*):

$$E(f(x)) \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Integração de Monte Carlo: Formulação

- Combinando estes dois resultados podemos construir um integrador numérico <*I*> para estimar o valor do integral:

$$E(f(x)) = \int_D f(x)p(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

$$\int_D \frac{f(x)p(x)dx}{p(x)} \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

$$\int_D f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

- O integrador é $\langle I \rangle = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} \approx I$ com $\lim_{N\to\infty} \langle I \rangle = I$

O integral pode ser estimado pelo somatório pesado de amostras de *f(x)* geradas **aleatoriamente**

Integração de Monte Carlo: Distribuição Uniforme

$$\int_D f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

- Este integrador numérico permite aproximar o valor do integral seleccionando **estocasticamente** pontos (amostras) no domínio da função!

- Se a distribuição de probabilidade usada para seleccionar os pontos de amostragem é uniforme, então todos os pontos têm a mesma probabilidade

$$p(x_i) = p, \forall x_i \in D$$

- O integrador passa a ser:

$$\int_D f(x)dx \approx \frac{1}{N * p}\sum_{i=1}^{N} f(x_i)$$

Integração de Monte Carlo: Cálculo de *p(x)*

- Distribuição uniforme

$$p(x)=c; \quad \int_a^b p(x)dx=1;$$

$$\int_a^b c\,dx=1; \quad cx\Big|_a^b=1;$$

$$c(b-a)=1;$$

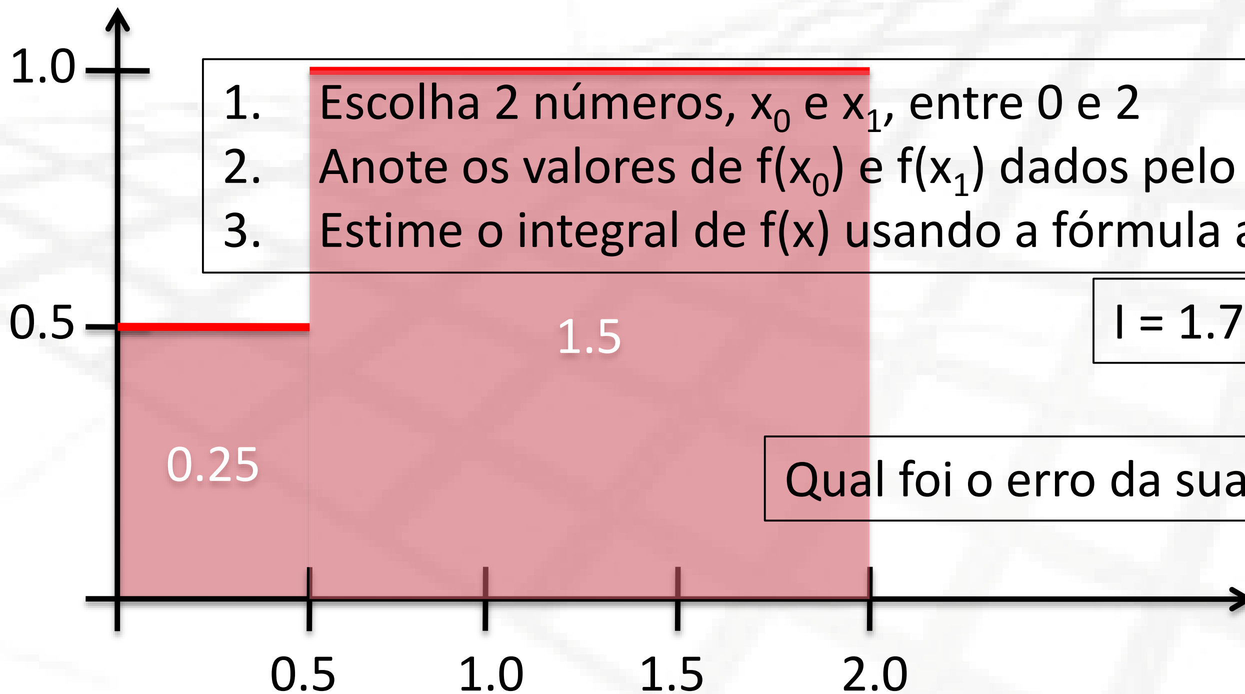
$$c=\frac{1}{b-a} \Leftrightarrow p(x)=\frac{1}{b-a}$$

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

Integração de Monte Carlo: Distribuição Uniforme

1. Escolha 2 números, x_0 e x_1, entre 0 e 2
2. Anote os valores de $f(x_0)$ e $f(x_1)$ dados pelo professor
3. Estime o integral de f(x) usando a fórmula abaixo (N=2)

I = 1.75



Qual foi o erro da sua estimativa?

$$\int_0^2 f(x)dx \approx \frac{(2-0)}{N}\sum_{i=1}^{N} f(x_i)$$

Integração de Monte Carlo: quadratura

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

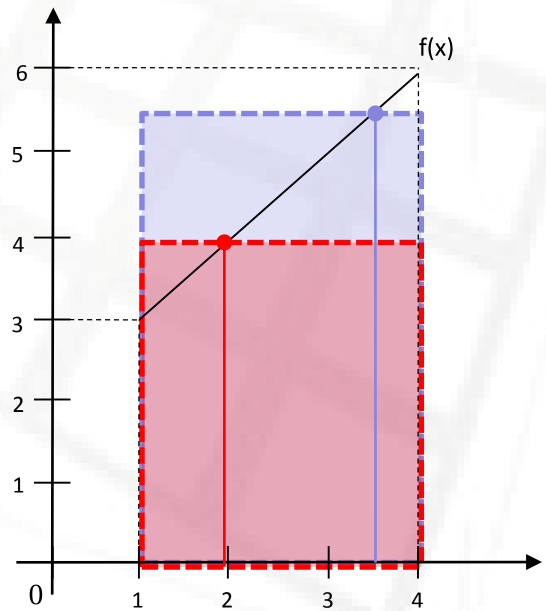
O valor do integral é aproximado, para cada x_i, por um volume cuja altura é o valor de $f(x_i)$ e cuja largura é dada pelo tamanho do domínio de integração *(b-a)*

$$I = \int_1^4 (x+2)dx = \left.\frac{x^2}{2}\right|_1^4 + 2x\Big|_1^4 = 13{,}5$$

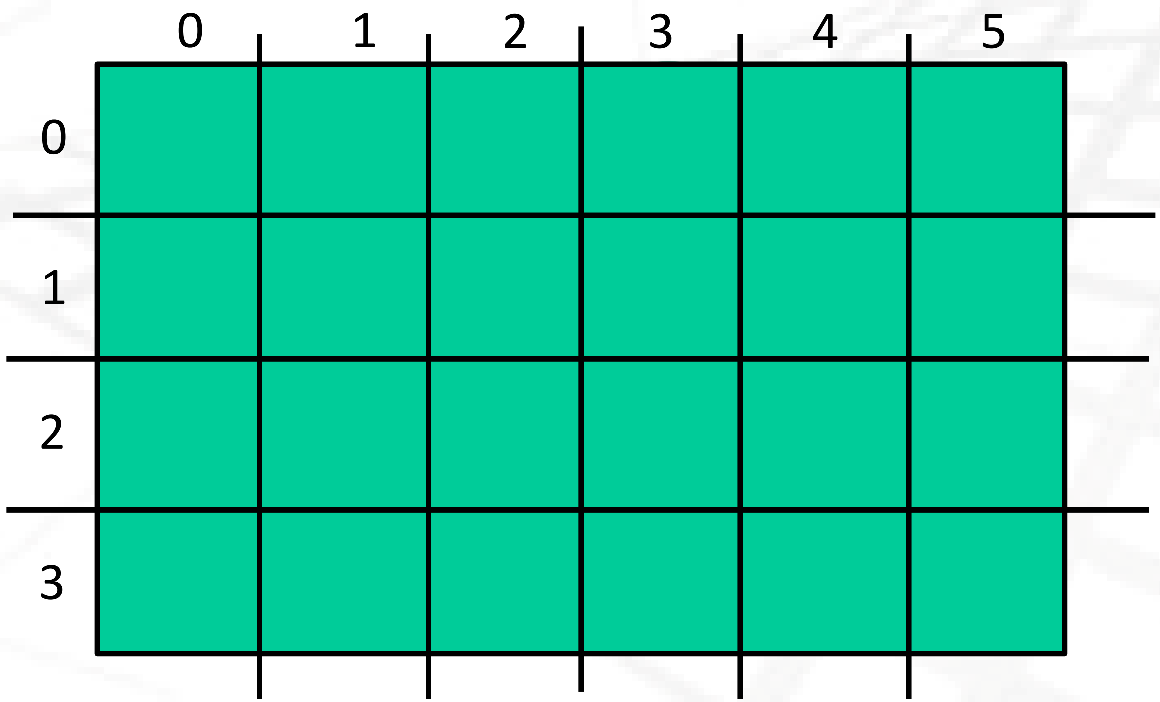
$x_0 = 2$ $\quad f(x_0) = f(2) = 4 \quad \langle I_0 \rangle = (4-1)\, f(x_0) = 12$

$x_1 = 3{,}5$ $\quad f(x_1) = f(3{,}5) = 5{,}5 \quad \langle I_1 \rangle = (4-1)\, f(x_1) = 16{,}5$

$\langle I \rangle = ½ * (\langle I_0 \rangle + \langle I_1 \rangle) = (12+ 16{,}5) / 2 = 14.25$

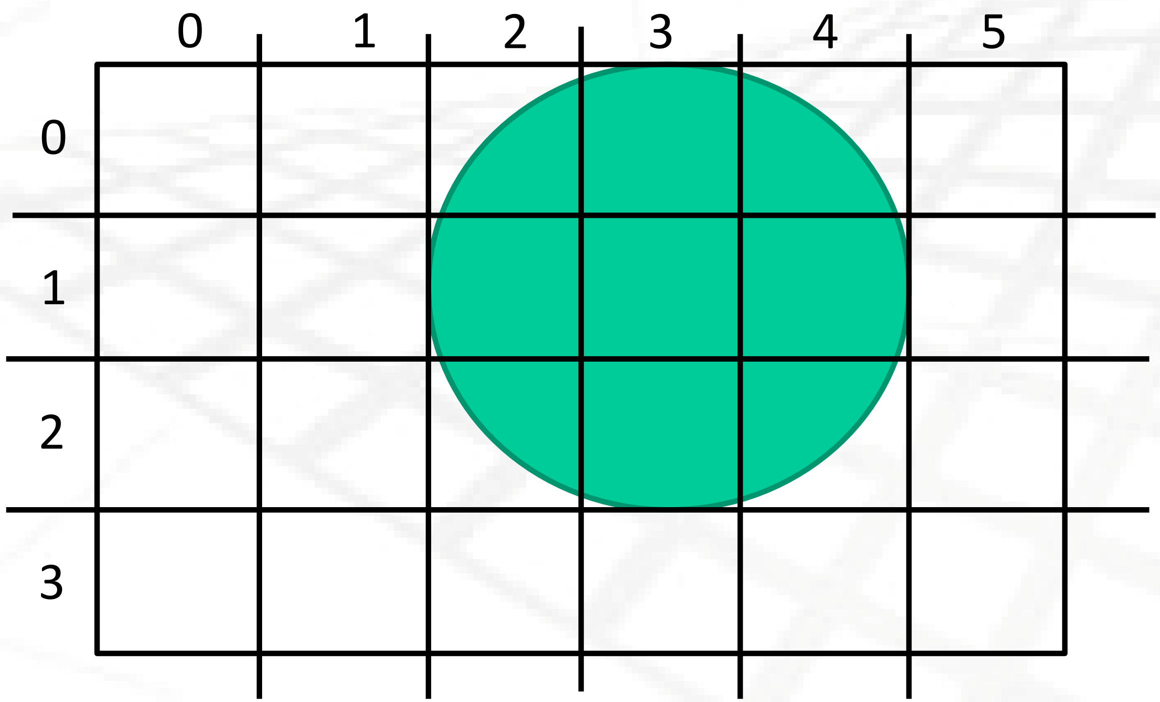


Calcule a área da superfície escondida



$$A = \int_0^3 \int_0^5 f(x,y)\,dxdy \approx \frac{24}{N} \sum_{i=0}^{N-1} f(x_i, y_i)$$

Calcule a área da superfície escondida



$$A = \int_0^3 \int_0^5 f(x,y)\,dx\,dy = \pi * r^2 = 7.0686$$

Integração de Monte Carlo: Semi-Esfera Uniforme

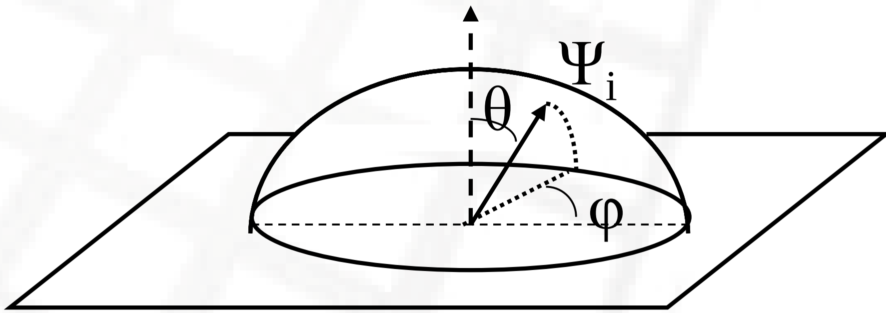
AMOSTRAGEM UNIFORME DA SEMI-ESFERA

$$L_r(p \to \omega_r) = \int_{\Omega_s} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i) \cos(N_p, \omega_i) d\omega_i$$

O ângulo sólido da semi-esfera é 2π, logo p(ω)=1/ 2π

Seleccionar as direcções aleatoriamente e seguir o raio com direcção

$$\Psi_i = (\theta, \varphi) = \left(\cos^{-1} \xi_1, 2\pi \xi_2\right)$$



$$\langle L_r(p \to \omega_r) \rangle = \frac{2\pi}{N} \sum_{i=1}^{N} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i) \cos(\theta_i)$$

Monte Carlo *ray tracing*

$$\left\langle L(p \to \omega_r) \right\rangle = L_e(p \to \omega_r) + L_d(p \to \omega_r) + \frac{2\pi}{N} \sum_{i=1}^{N} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i) \cos(\theta_i)$$

$\left\langle L(p \to \omega_r) \right\rangle$ Radiância reflectida por *p* na direcção ω_r

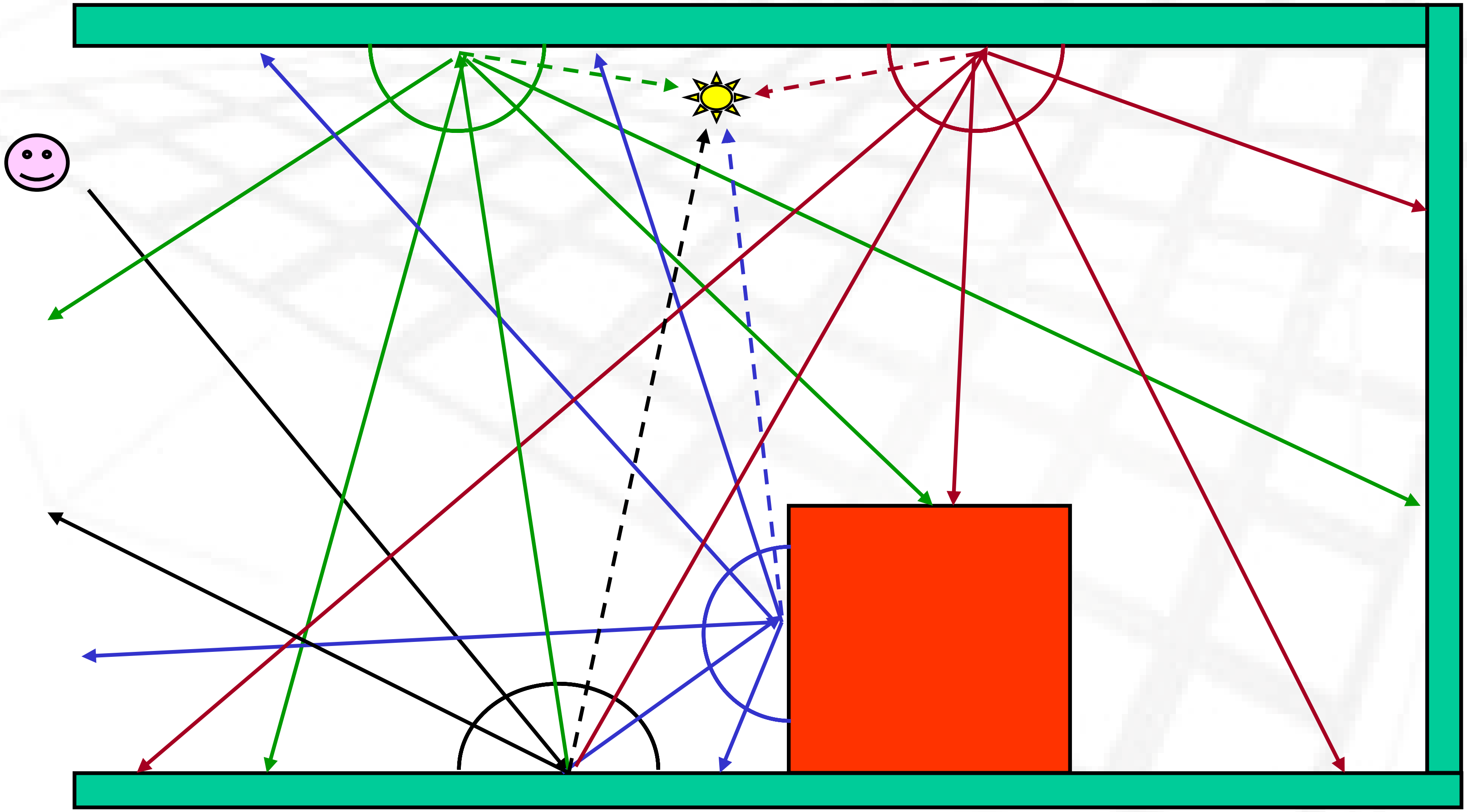
$L_e(p \to \omega_r)$ Radiância auto-emitida por *p* na direcção ω_r

$L_d(p \to \omega_r)$ Radiância reflectida por *p* na direcção ω_r devida a iluminação directa

$$\frac{2\pi}{N} \sum_{i=1}^{N} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i) \cos(\theta_i)$$

Radiância reflectida por *p* na direcção ω_r devida a iluminação indirecta

Monte Carlo *ray tracing*



Uniform Monte Carlo *ray tracing*: algoritmo (1/3)

```
radiance renderPixel (camera, x,y, scene) {
  ray = generateRay (camera, x, y, PRIMARY);
  rad = MC_RT (ray, scene, 0);
}
radiance MC_RT (ray, scene, depth) {
  rad = {0., 0., 0.};    // RGB
  inter = trace (ray, scene);
  if inter==NULL return (rad);  // no intersection
  rad += directLighting (inter, ray, scene)
  if (depth < MAX_DEPTH)
    rad += shootSecondaryRays (inter, ray, scene, depth);
  return (rad);
}
```

Uniform Monte Carlo *ray tracing*: algoritmo (2/3)

```
radiance directLighting (int, ray, scene) {
  rad = {0., 0., 0.}
  for l in scene.lights {
    radLight = {0., 0., 0.}
    for sample= 0 to spl-1 {  // spl: samples per light
      sRay, pdf = generateRay (int.point, l, SHADOW);
      visible = traceShadow (shadowRay, scene);
      if (visible) // light source contribution
          N = int.normal; brdf = int.brdf (ray, sRay);
          radLight += brdf * l.L * cos (N, sRay.dir) / pdf;
    }
    rad += (radLight / spl);
  } return (rad);
}
```

Uniform Monte Carlo *ray tracing*: algoritmo (3/3)

```
radiance shootSecondaryRays (int, ray, scene, depth)  {
  rad = {0., 0., 0.};
  for i = 0 to num_SecondaryRays-1 {
    (r1, r2) = get2randomNumbers ();
    // uniform sampling of the hemisphere
    (theta, phi) = (arccos (r1), 2 * PI * r2);
    pdf = 1. / (2. * PI);
    secRay = generateRay (int, ray, (theta,phi), SEC);
    N = int.normal; brdf = int.brdf (ray, secRay);
    secRay_rad = MC_RT (secRay, scene, depth+1);
    secRay_rad *= brdf * cos (N, secRay.dir) / pdf;
    rad + = secRay_rad ;
  }   return (rad / num_SecondaryRays); }
```

Integração de Monte Carlo

- Redução da variância
  - Força bruta: Aumento do número de amostras N
  - Amostragem Estratificada
  - Amostragem por Importância

Integração de Monte Carlo: uniforme

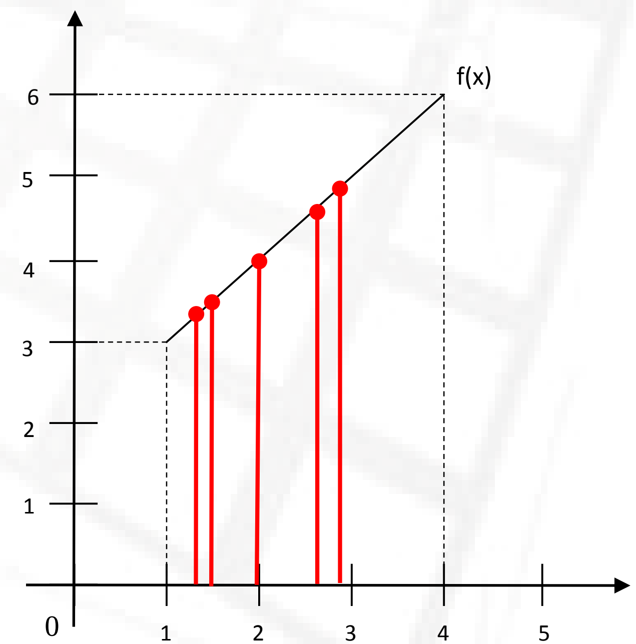
$$I = \int_1^4 (x+2)dx = 13{,}5$$

$$p(x) = c,\ \int_1^4 c\,dx = 1 \Leftrightarrow cx\Big]_1^4 = 1 \Leftrightarrow c = \frac{1}{3},\ p(x) = \frac{1}{3}$$

$$\langle I \rangle = \frac{3}{N=5} \sum_{i=0}^{N-1=4} (x_i + 2)$$

Valores para N=5

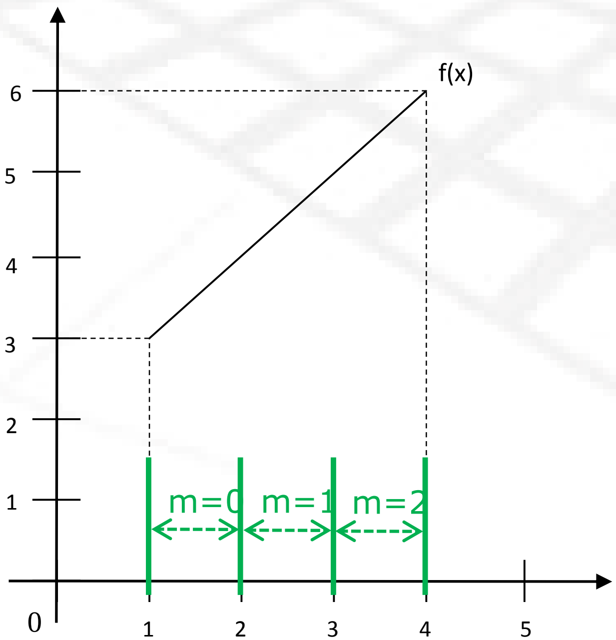
ξ	xi	f(xi)
0,31	1,93	3,93
0,59	2,77	4,77
0,61	2,83	4,83
0,09	1,27	3,27
0,15	1,45	3,45



$$\langle I \rangle = \frac{3}{5}(3{,}93 + 4{,}77 + 4{,}83 + 3{,}27 + 3{,}45) = 12{,}16$$

Integração de Monte Carlo: estratificação

- As amostras podem ser melhor distribuídas no domínio se este for subdividido em subdomínios disjuntos (strata) e o integral calculado separadamente em cada estrato



$$\int_1^4 (x+2)dx = \sum_{m=0}^{m=2} \int_{1+m}^{1+m+1} (x+2)dx$$

$$\left\langle \int_1^4 (x+2)dx \right\rangle = \sum_{m=0}^{m=2} \frac{1}{n_m} \sum_{i=1}^{n_m} \frac{f(x_{m,i})}{p(x_{m,i})}$$

Integração de Monte Carlo: estratificação

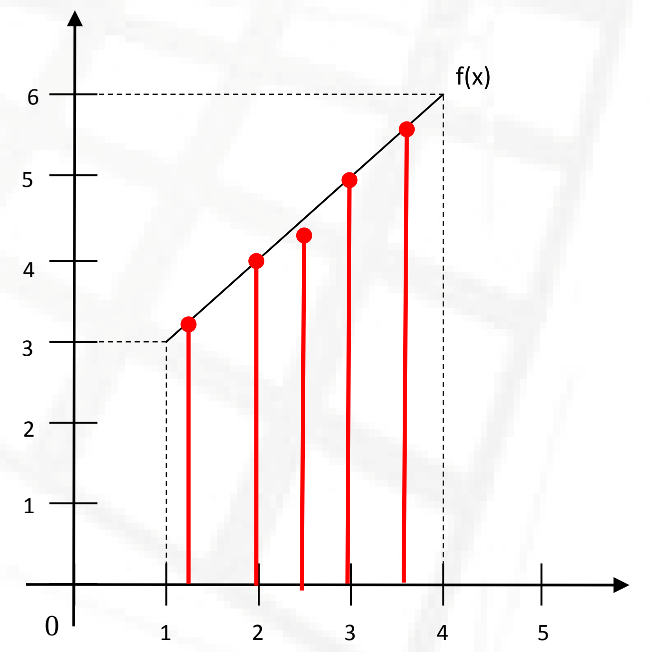
$$I = \int_1^4 (x+2)dx$$

$$p(x) = c,\ \int_1^4 c\,dx = 1 \Leftrightarrow cx\Big]_1^4 = 1 \Leftrightarrow c = \frac{1}{3},\ p(x) = \frac{1}{3}$$

Valores para m=N=5 (1 sample per stratum)

$$\langle I \rangle = \frac{3}{N}\sum_{i=1}^{N(=m)} (x_i + 2)$$

ξ	aj	xi	f(xi)
0,31	1	1,19	3,19
0,59	1,6	1,95	3,95
0,61	2,2	2,57	4,57
0,09	2,8	2,85	4,85
0,15	3,4	3,49	5,49



$$\langle I \rangle = \frac{3}{5}(3{,}19 + 3{,}95 + 4{,}57 + 4{,}85 + 5{,}49) = 13{,}24$$

Nota: I=13,5 e <I>=12.16 com uniforme

Integração de Monte Carlo: estratificação

AMOSTRAGEM ESTRATIFICADA DA SEMIESFERA

$$L_r(p \to \omega_r) = \int_{\Omega_s} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i) \cos(N_p, \omega_i) d\omega_i$$

O ângulo sólido da semiesfera é 2π, logo p(ω)=1/ 2π

Seleccionar as direcções com M *strata* em θ e N *strata* em ψ:

$$(\theta, \psi) = \left(\cos^{-1} \frac{\xi_1 + j}{M}, 2\pi \frac{\xi_2 + i}{N} \right)$$

O integrador é:

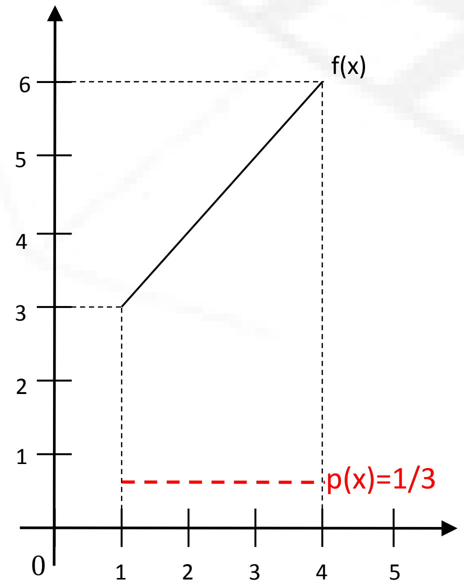
$$\langle L_r(p \to \omega_r) \rangle = \frac{2\pi}{M * N} \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i) \cos(\theta_i)$$

Stratified Uniform Monte Carlo *ray tracing*:
algoritmo (3/3)

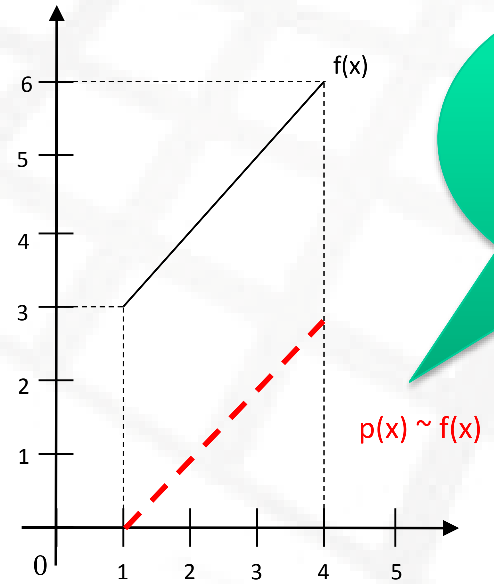
```
radiance shootSecondaryRays (int, ray, scene, depth)  {
  rad = {0., 0., 0.}; pdf = 1. / (2. * PI);
  for j = 0 to M-1 {
    for i = 0 to N-1 {
      (r1, r2) = get2randomNumbers ();
      // stratified uniform sampling of the hemisphere
      (theta, phi) = (arccos ((r1+j)/M), 2 * PI * (r2+i)/N);
      secRay = generateRay (int, ray, (theta,phi), SEC);
      N = int.normal; brdf = int.brdf (ray, secRay);
      secRay_rad = MC_RT (secRay, scene, depth+1);
      secRay_rad *= brdf * cos (N, secRay.dir) / pdf;
      rad + = secRay_rad ;  } }
  return (rad / (N * M)); }
```

Integração de Monte Carlo: importância

- Distribuição de probabilidade uniforme : para seleccionar as amostras não é usada informação acerca do integrando, *f(x)*;
- Algumas partes de *f(x)* contribuem mais para o valor do integral: aquelas onde o valor de *f(x)* é maior;
- Se *p(x)* tem uma forma similar a *f(x)*, então as amostra ficarão localizadas, com maior probabilidade, nas partes mais importantes do domínio.



Amostragem Uniforme



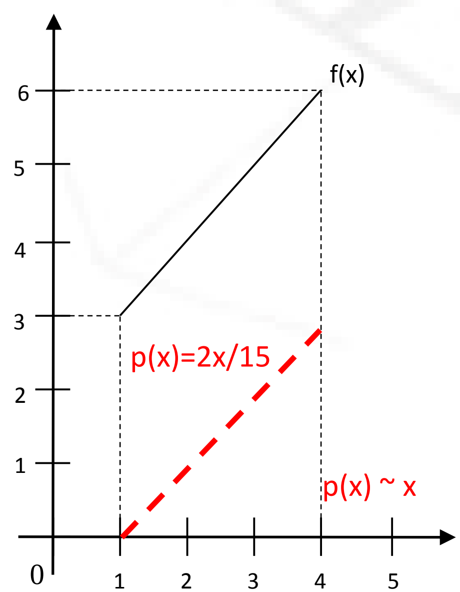
Amostragem por Importância

Mas como normalizar e gerar amostras com esta pdf?

Integração de Monte Carlo: importância

- Normalizar

$$p(x) = cx; \quad c\int_1^4 x\,dx = 1; \quad c = 2/15; \quad p(x) = \frac{2x}{15}$$

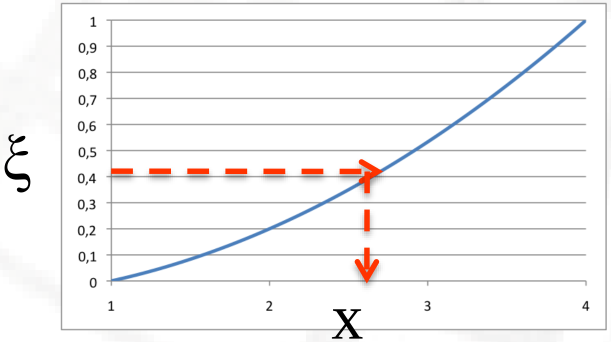


- Para extrair amostras que seguem esta pdf:

1. Calcular a cdf:

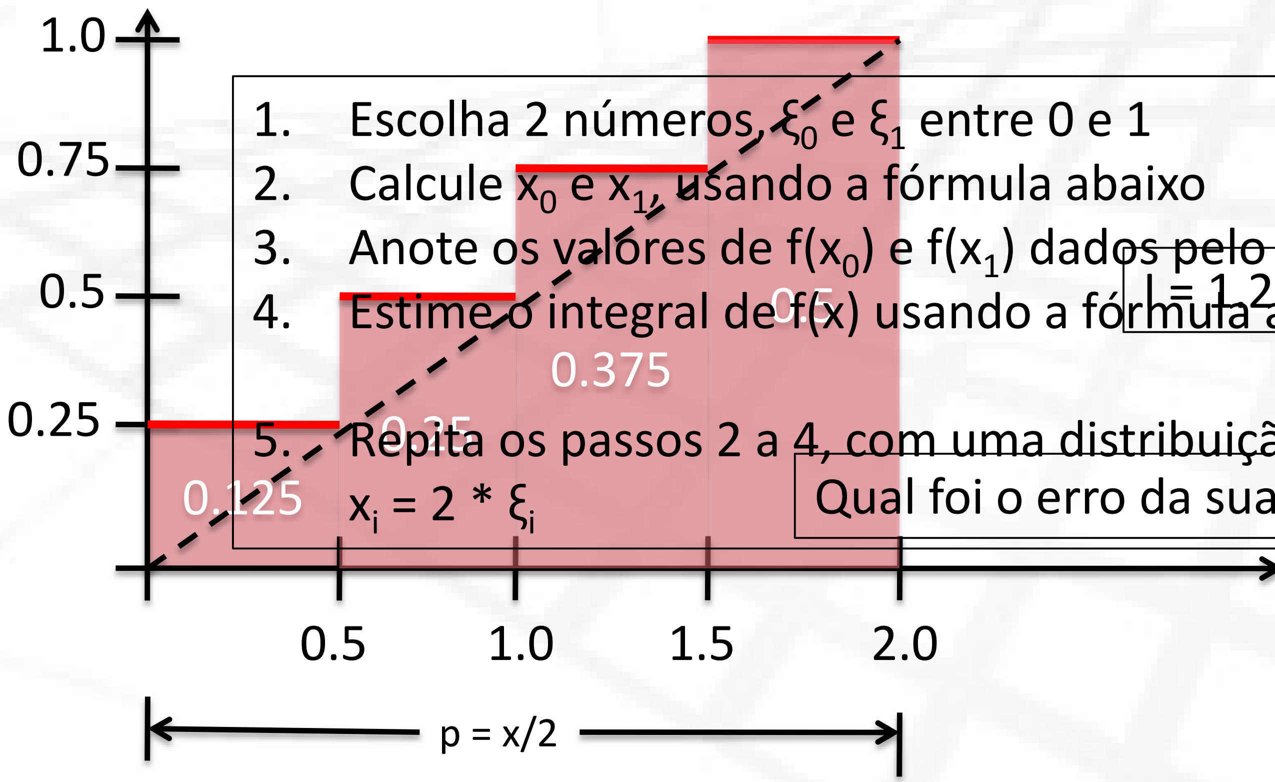
$$cdf(x) = \int_1^x p(x)dx = \int_1^x \frac{2x}{15}dx = \frac{2}{15}\left.\frac{x^2}{2}\right|_1^x = \frac{x^2-1}{15}$$

2. Inverter a cdf:



$$\frac{x^2-1}{15} = \xi \Leftrightarrow x = \sqrt{15\xi + 1}$$

Integração de Monte Carlo: Importância



1. Escolha 2 números, ξ_0 e ξ_1 entre 0 e 1
2. Calcule x_0 e x_1, usando a fórmula abaixo
3. Anote os valores de $f(x_0)$ e $f(x_1)$ dados pelo professor
4. Estime o integral de f(x) usando a fórmula abaixo (N=2)

I = 1.25

5. Repita os passos 2 a 4, com uma distribuição uniforme $x_i = 2 * \xi_i$

Qual foi o erro da sua estimativa?

$$\int_0^x x/_2 \, dx = \left.\frac{x^2}{4}\right|_0^x = \frac{x^2}{4} = \xi \Rightarrow x = 2\sqrt{\xi}$$

$$\int_0^2 f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

Integração de Monte Carlo: importância

$$I=\int_1^4 (x+2)dx=13.5 \quad p(x)=cx,\ \int_1^4 cx\,dx=1 \Leftrightarrow c\frac{x^2}{2}\Big]_1^4=1 \Leftrightarrow c=\frac{15}{2},\ p(x)=\frac{2x}{15}$$

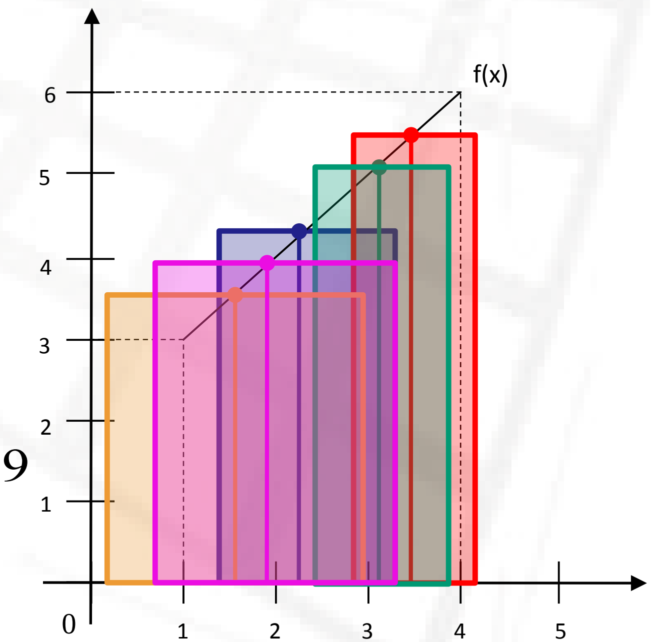
$$\langle I\rangle=\frac{1}{N}\sum_{i=1}^{N}\frac{(x_i+2)}{2x_i/15},\quad x_i=\sqrt{15\xi+1}$$

Valores para N=5

ξ	x_i	$p(x_i)$	$f(x_i)$
0,31	2.37	0.316	4,37
0,59	3.13	0.417	5,13
0,61	3.19	0.425	5,19
0,09	1.53	0.204	3.53
0,15	1.80	0.240	3.80

$$\langle I\rangle=\frac{1}{5}\left(\frac{4.37}{.316}+\frac{5.13}{.417}+\frac{5.19}{.425}+\frac{3.53}{.204}+\frac{3.80}{.240}\right)=14.29$$

Uniforme e mesmos ξ <I>= 12.16



As amostras com maior probabilidade têm um peso menor (1/p(x))

Integração de Monte Carlo: importância

$$L_r(p \rightarrow \omega_r) = \int_{\Omega_s} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i) \cos(N_p, \omega_i) d\omega_i$$

Integrando é um produto: seleccionar um dos factores.

COSINE WEIGTHED IMPORTANCE SAMPLING:

$$p(\omega) = \frac{\cos\theta}{\pi} \qquad (\theta, \psi) = \left(\sin^{-1}\sqrt{\xi_1}, 2\pi\xi_2\right)$$

$$\langle L_r(p \rightarrow \omega_r) \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i) \cos(\theta_i)}{\cos(\theta_i) / \pi}$$

$$\langle L_r(p \rightarrow \omega_r) \rangle = \frac{\pi}{N} \sum_{i=1}^{N} f_r(p, \omega_i \leftrightarrow \omega_r) L(p \leftarrow \omega_i)$$

Importance Sampling Monte Carlo *ray tracing*: algoritmo (3/3)

```
radiance shootSecondaryRays (inter, ray, scene, depth)  {
  rad = {0., 0., 0.};
  for i = 0 to num_SecondaryRays-1 {
    (r1, r2) = get2randomNumbers ();
    // uniform sampling of the hemisphere
    (theta, phi) = (arcsin (sqrt(r1)), 2 * PI * r2);
    secRay = generateRay (point, ray, (theta,phi), SEC);
    N = int.normal; brdf = int.brdf (ray, secRay);
    secRay_rad = MC_RT (secRay, scene, depth+1);
    secRay_rad *= brdf;
    rad + = secRay_rad ;  }
  return (rad / num_SecondaryRays);
}
```

Monte Carlo: bias

- Quando parar a emissão de raios secundários *(path length)?*
  - Usar uma profundidade máxima fixa
  - Quando a contribuição esperada de um raio é inferior a um dado limite
- Estes são métodos determinísticos que afectam o valor do integral (*bias*)!

not biased	*biased*
$\lim\limits_{N\to\infty}\langle I\rangle = I$	$\lim\limits_{N\to\infty}\langle I\rangle = I + \varepsilon$

Monte Carlo: roleta russa

- Definir a probabilidade α de terminar a travessia (não disparar um raio secundário)
- Antes de disparar um raio gerar um número aleatório, ξ, uniformemente distribuído em [0, 1[
- Se ξ <= α então não disparar o raio
- Se ξ > α então disparar o raio
- Uma vez que há uma probabilidade α de não disparar um raio, a contribuição dos raios disparados deve ser multiplicada por 1/(1- α), para compensar aqueles que não são disparados

$$L(p \leftarrow \Psi) \approx \frac{1}{(1-\alpha)}(\xi \leq \alpha \ ? \ 0 : L(p \leftarrow \Psi_i))$$

Russian Roulette Monte Carlo *ray tracing*:

algoritmo (1/3)

```
radiance MC_RT (ray, scene, depth) {
  rad = {0., 0., 0.};   // RGB
  inter = trace (ray, scene);
  if inter==NULL return (rad);  // no intersection
  rad += directLighting (inter, ray, scene)
  // Russian Roulette
  r = getRandomNumber ();
  if (r > TERMINATE_PROBABILITY)
    contrib = shootSecondaryRays (inter, ray, scene, depth);
    rad += (contrib / (1. - TERMINATE_PROBABILITY));
  return (rad);
}
```