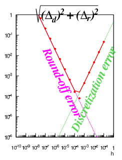
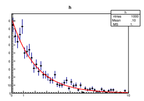
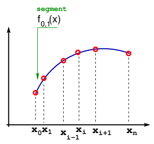
Computational Physics

numerical methods with C++ (and UNIX)







Fernando Barao
Instituto Superior Tecnico, Dep. Fisica
email: barao at lip.pt

Computational Physics
Monte-Carlo methods

Fernando Barao, Phys Department IST (Lisbon)

Generating random variables

- The common problem is to have a variable x distributed according a given distribution function $p(x)$ we are going to make a change of variable such that the number of events (randoms) generated is independent of the used variable

$$dN = p(x)dx = p(y)dy$$

- Suppose that y is a random variable distributed in $[0,1]$ and x starts at x_0

$$\boxed{p(y) = 1 \quad \Rightarrow \quad \int_0^y dy' = \int_{x_0}^{x} p(x')dx' \quad \Rightarrow \quad y = \int_{x_0}^{x} p(x')dx'}$$

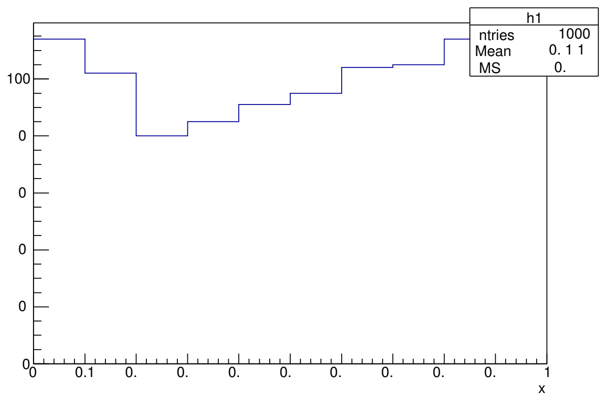
- For generating a random variable x in a interval $[x_0, x_1]$ we just make sure that:

$$\int_{x_0}^{x_1} p(x)dx = 1$$

and we invert the relation above (boxed) giving us $x(y)$.
Provided a random y in $[0,1]$ we generate a random x in $[x_0, x_1]$.

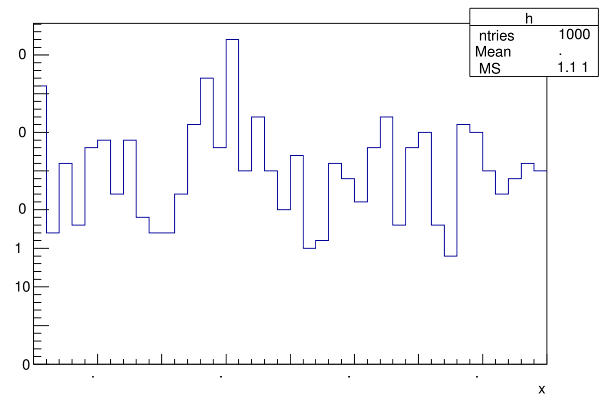
uniform [0,1] $\rightarrow$ uniform [a,b]

- Let's transform a variable
  y uniformly distributed in $[0,1]$
  into a variable
  x uniformly distributed in $[a,b]$



- Normalization of $p(x)$:

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



- Transformation:

$$y = \int_a^x \frac{1}{b-a}dx' = \frac{x-a}{b-a}$$

$$\Rightarrow \quad x - a = (b-a)y$$

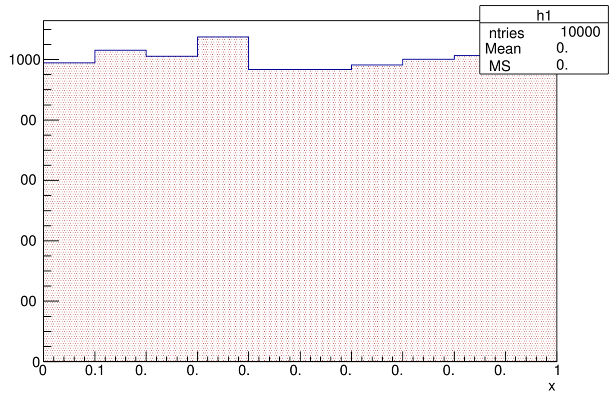
$$\boxed{x = a + (b-a)\, y}$$

uniform [0,1] → exponential $[0, \infty]$

- Let's transform a variable
  - ☞ y uniformly distributed in $[0, 1]$

  into a variable
  - ☞ x distributed according to

  $\boxed{p(x) \propto e^{-x}}$ in $[0, \infty]$

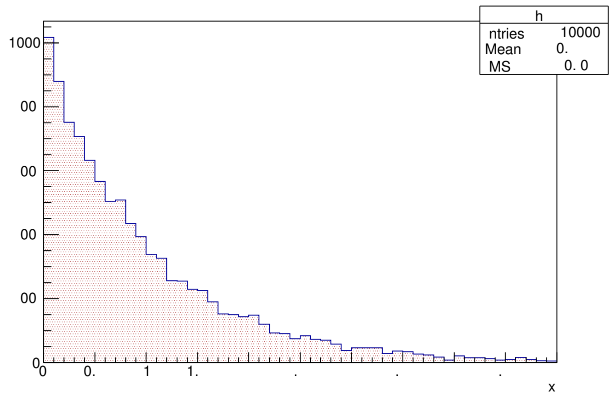


- Normalization of $p(x)$:

$$k \int_0^{+\infty} e^{-x}\, dx = 1$$

$$\Rightarrow k \left[-e^{-x}\right]_0^{\infty} = 1 \Rightarrow \quad k = 1$$

$$\boxed{p(x) = e^{-x}}$$



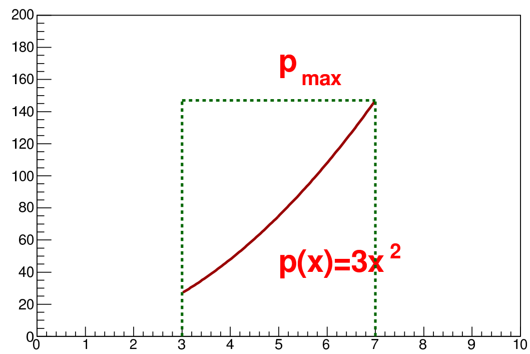
- Transformation:

$$y = \int_0^x e^{-x'} dx' = \left[-e^{-x'}\right]_0^x = 1 - e^{-x}$$

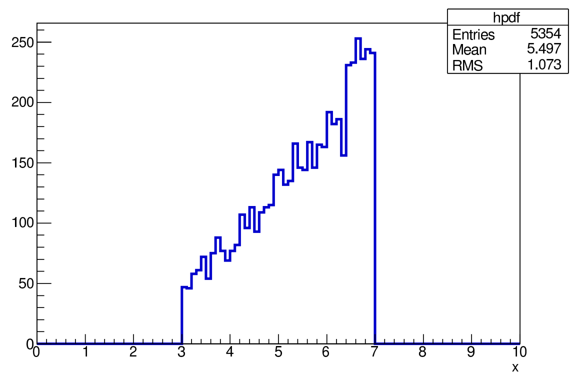
$$\boxed{x = -\ln(1-y)}$$

MC integration: acceptance-rejection

- For applying the method before said we need to integrate the pdf and invert it
- In case it is not possible we can use the more generic but less efficient method of acceptance-rejection



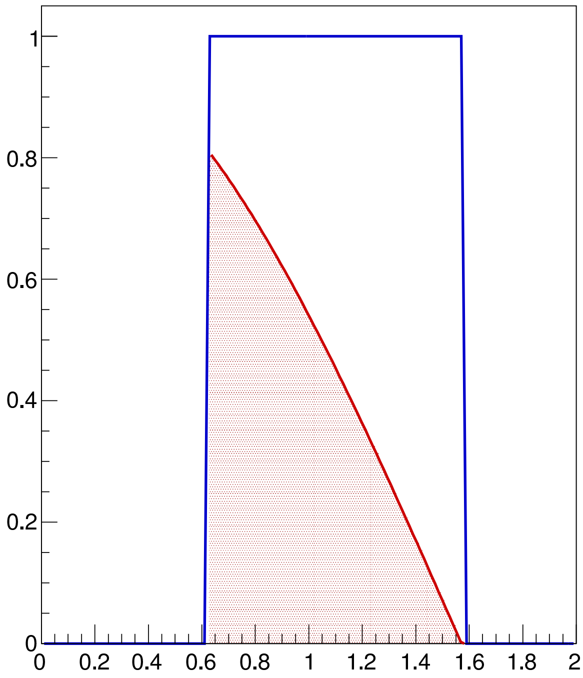
- For generating a x variable in the interval $[a, b]$ distributed according to a pdf $p(x)$ wo do the following:
  - ☞ generate a uniform random x_R between $[a, b]$
  - ☞ compute $p(x_R)$ and the ratio $\frac{p(x_R)}{p_{max}}$
  - ☞ generate a second random u_R from $U(0, 1)$
  - ☞ if $u_R \leq \frac{p(x_R)}{p_{max}}$ accept the variable x_R



MC integration example: $cos(x)$

- method introduced by von Neumann
- for integrating the function we define an envelope with an area $A = (x_{max} - x_{min}) * f_{max}$
- generate two random variables
  - x_R in in range $[x_{min}, x_{max}]$
  - f_R in in range $[0, f_{max}]$
- count the number of events N_R that $\boxed{f_R \leq f(x_R)}$
- the integral $\boxed{I = (x_{max} - x_{min})\ f_{max}\ \frac{N_R}{N}}$
- the integral error

$$\boxed{\sigma_I = \frac{(x_{max}-x_{min})\ f_{max}}{N} \sqrt{N_R\left(1 - \frac{N_R}{N}\right)}}$$



Using 100 randoms:
$\int_{0.2\pi}^{0.5\pi} cos(x)\ dx = 0.435 \pm 0.037$

MC integration example: $f(x) = 3x^2$

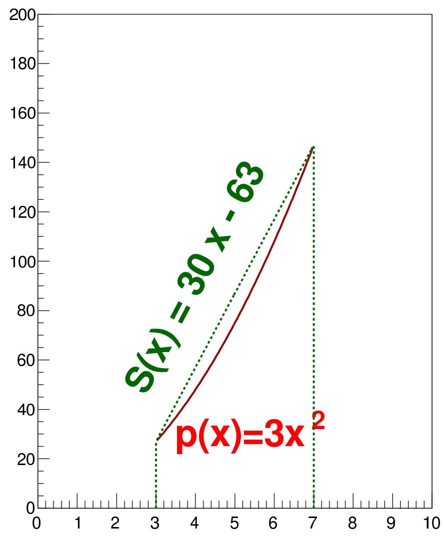
- The acceptance-rejection method is inefficient if the function varies quickly
- The number of randoms accepted (efficiency) is given by:

$$\varepsilon = \frac{\int_a^b f(x)\ dx}{(b-a)\ f_{max}}$$

- The efficiency can be improved using an auxiliar function $S(x)$ that has a shape close to the one we want to sample

$$\varepsilon_S = \frac{\int_a^b f(x)\ dx}{\int_a^b q(x)\ dx}$$

where $q(x) = C\ S(x) \geq f(x)$ for x in $[a, b]$



efficiencies:

$\varepsilon = \frac{N_R}{N} \simeq 0.535$

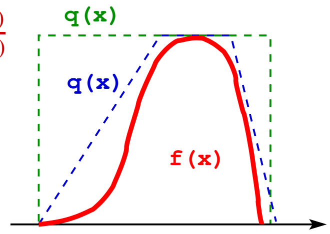
$\varepsilon_S = \frac{N_R}{N} \simeq 0.91$

MC integration: acc-rej with auxiliar function

For generating a random variable x distributed according to $f(x)$ in the interval $[a, b]$

1. Find auxiliar function $S(x)$ with a shape close to the function $f(x)$ we want to sample
   - integrable, invertible
   - from $S(x)$ we define a pdf $p(x)$ and we apply the transformation $y \in U[0,1] \rightarrow x \in U(p(x))$
   - invert transformation equation: $x(y)$
   - using $x(y)$ and generating y uniformly between $[0, 1]$ we obtain the random variable x_R distributed according to $p(x)$
2. Define a function $q(x) = C\ S(x)$ such that $q(x) \geq f(x)$ in the interval $[a, b]$
3. Generate random variable x_R according to $p(x)$ and compute $\frac{q(x_R)}{f(x_R)}$
4. Generate random $u_R \in U[0, 1]$ and accept x_R if:

$$\boxed{u_R \leq \frac{q(x_R)}{f(x_R)}}$$



MC integration: $f(x) = 3x^2$

✔ We want to generate a variable according to a function

$$\boxed{f(x) = 3x^2}$$

1. Define the auxiliar function $S(x)$ to improve acceptance-rejection efficiency

$$\boxed{S(x) = 30x - 63}$$

generate random in [3,7] interval according to auxiliar function

get pdf(x): $S(x) \rightarrow p(x)$ with $\int_3^7 p(x)dx = 1$

$k \int_3^7 S(x)\, dx = 348 \quad \Rightarrow \quad k = \frac{1}{348}$

$$\boxed{p(x) = \frac{1}{348}(30x - 63)}$$

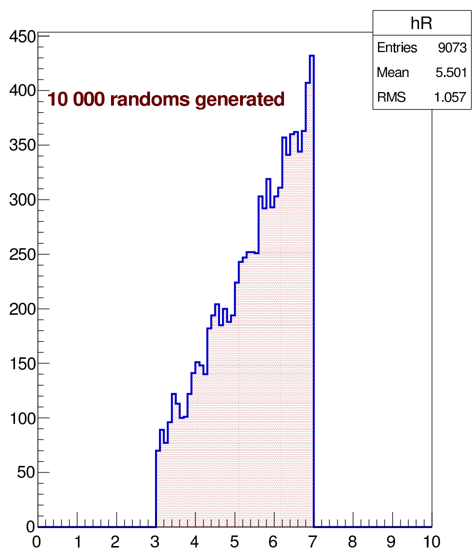
make transformation: $y[0, 1] \rightarrow x$ according to $p(x)$

$y = \int_3^x p(x')\, dx' = 348$

$$\boxed{\text{Solve } x(y)\text{: } 15x^2 - 63x + 54 - 348y = 0}$$

2. define $q(x)$ which in this case is $= S(x)$ and generate x_R according to $p(x)$

3.,4. generate $u_R \in u[0, 1]$ and accept x_R if $u_R \leq \frac{q(x_R)}{f(x_R)}$



Multidimensional MC integration

A multi-dimensional integral:

$$I = \int_{a_1}^{b_1} dx_1 \int_{a_2}^{b_2} dx_2 \int_{a_3}^{b_3} \cdots \int_{a_n}^{b_n} dx_n \; f(x_1, x_2, \cdots, x_n)$$

✔ **Brute-force**

- generate random variables $x_1, x_2, \cdots, x_n \in U[0,1]$ in the interval $[a_i, b_i]$

$$x_i = a_i + (b_i - a_i) x_R \qquad \text{with } x_R \in [0,1]$$

- calculate the function at the random variables: $f(x_1, x_2, \cdots, x_n)$
- the Monte-Carlo integration:

$$I = \frac{\prod_{k=1}^{n}(b_k - a_k)}{N} \sum_{i=1}^{N} f(x_1, x_2, \cdots, x_n)_i$$

N = number of random set of n variables

Multidimensional MC integration (cont.)

✔ **Importance sampling**

$$I = \int_{a_1}^{b_1} dx_1 \int_{a_2}^{b_2} dx_2 \int_{a_3}^{b_3} \cdots \int_{a_n}^{b_n} dx_n \; \frac{f(x_1,x_2,\cdots,x_n)}{p(x_1,x_2,\cdots,x_n)} \; p(x_1, x_2, \cdots, x_n)$$