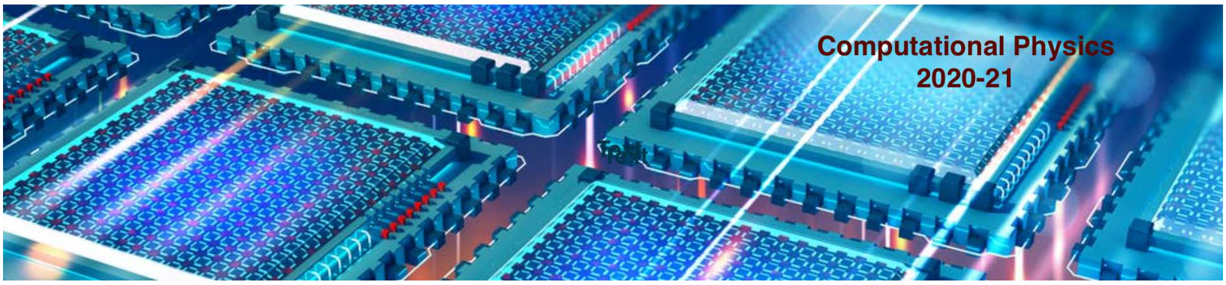
Computational Physics

numerical methods with C++ (and UNIX)

2020-21



Fernando Barao

Instituto Superior Técnico, Dep. Fisica

email: fernando.barao@tecnico.ulisboa.pt

Random numbers

✔ the most common uniform random number generators are based on Linear Congruential relations

$$N_i = (aN_{i-1} + c)\,\%m$$

☞ For example, with the parameters: a=6, c=7, m=5
and a seed: $N_0 = 2$
we get a **period 5** generator: 4, 1, 3, 0, 2, 4, 1, 3, 0, 2, ...

✔ a good uniform random number generator:

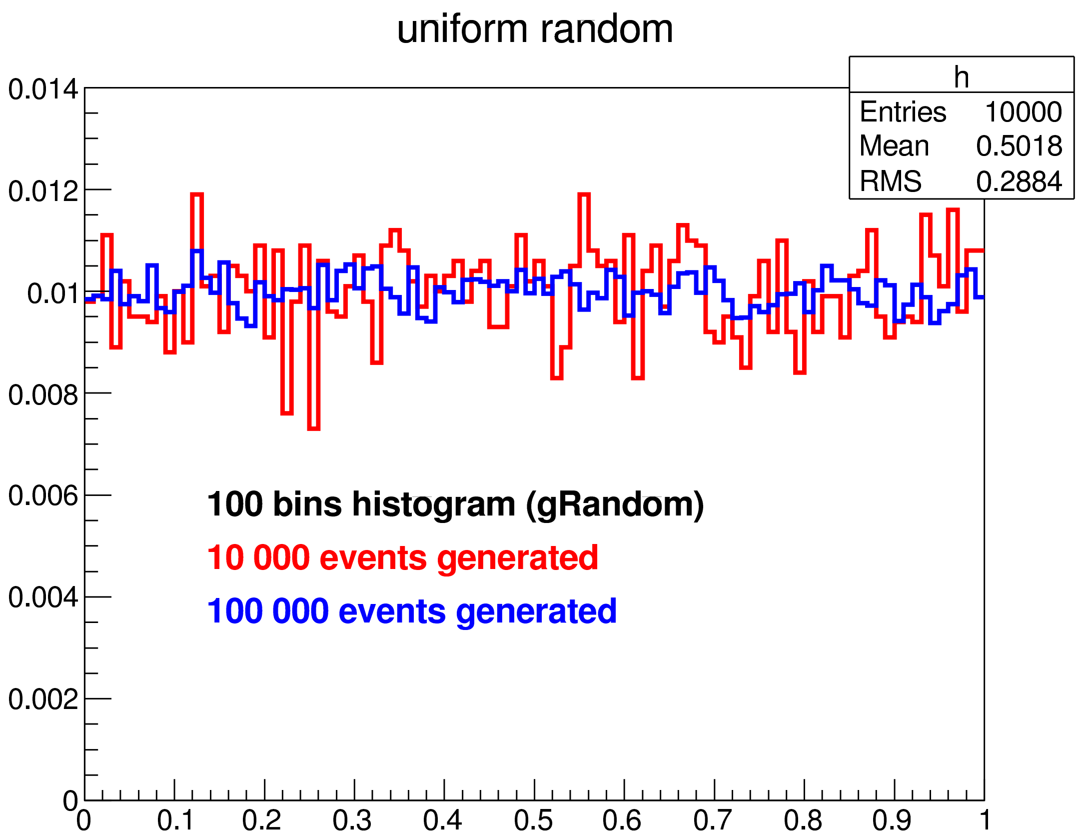
☞ produces a uniform distribution in the all the generation range

☞ shows no correlations between random numbers

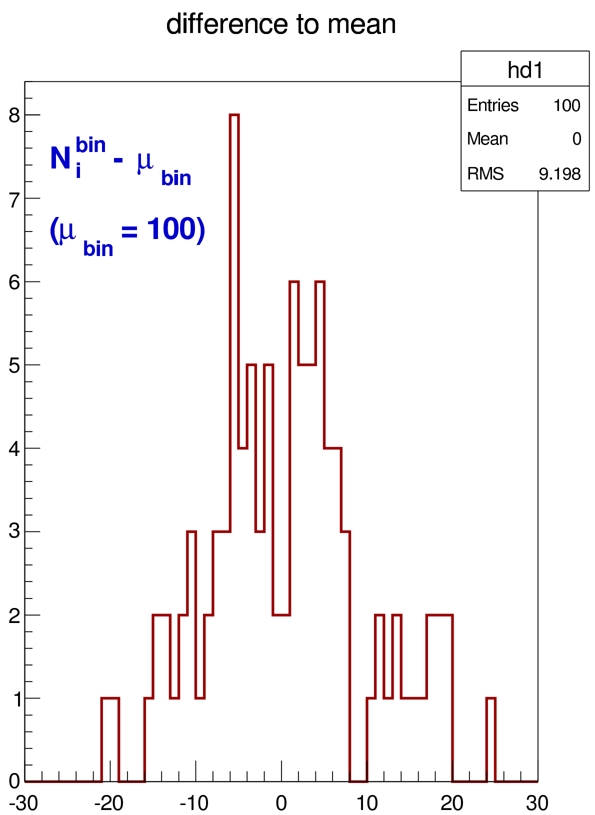
☞ the period of sequence repetition is as large as possible

☞ the generation algorithm shall be fast

Random numbers



random numbers



The differences of every bin statistics to the expected mean (μ) per bin

$N_i^{bin} - \mu$

distributes according to a **normal (gaussian) distribution** with

mean: μ

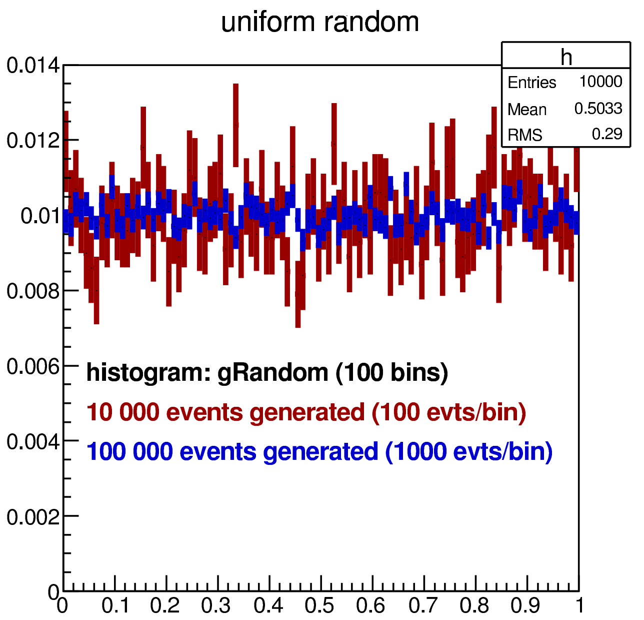
width: $\sigma \sim \sqrt{\mu}$

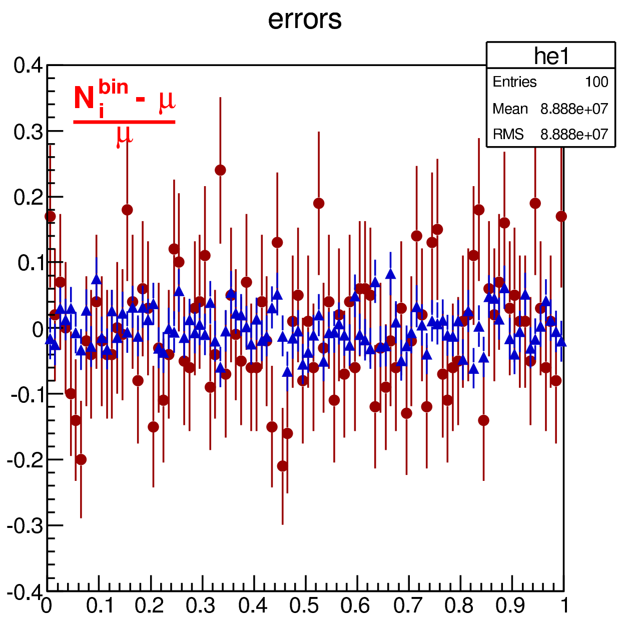
Consequence of the central limit theorem

the statistics accumulated in every bin

Central limit theorem, in probability theory, is a theorem that establishes the normal distribution as the distribution to which the mean (average) of almost any set of independent and randomly generated variables rapidly converges

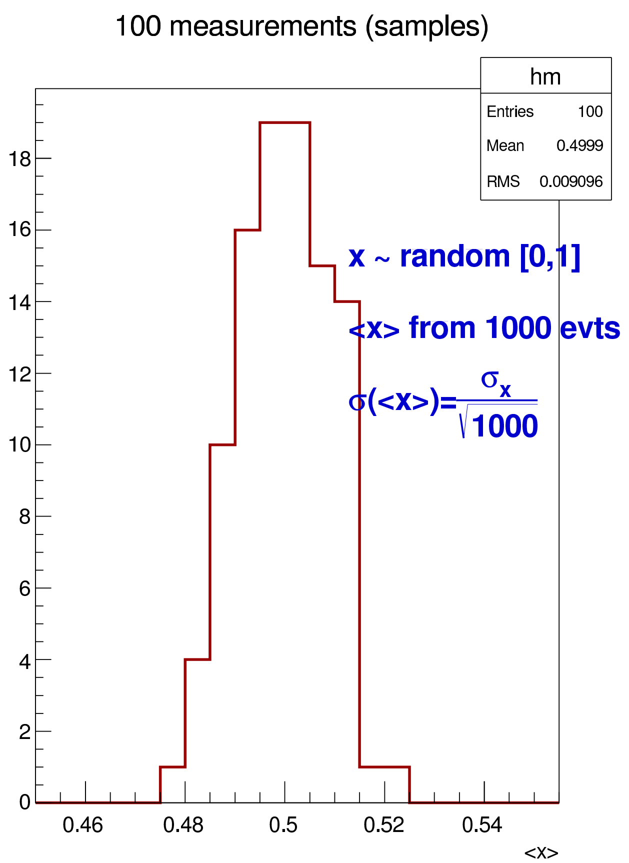
random numbers





- ✔ The number of random numbers per bin fluctuates wrt to the expected nb of events per bin ($Ngen/Nbins = \frac{10\,000}{100} = 100$)
- ✔ The deviation of the number of events per bin wrt to the expected mean of events per bin (μ) in percentage: $\frac{N_i^{bin}-\mu}{\mu}$ relative error: $\frac{\sigma_{N_i}}{<N_i>} \sim \frac{\sqrt{<N_i>}}{<N_i>} = \frac{1}{\sqrt{<N_i>}} = \frac{1}{\sqrt{100}}$

random numbers



Suppose we have N=100 data samples (experiments) and in every sample we throw n=1000 random numbers (measurement)

The mean of every sample $<x>$ is distributed in the plot

$<x> \sim 0.5 \qquad \sigma(<x>) = \frac{\sigma_x}{\sqrt{n}} \sim \frac{0.3}{33} \sim 0.01$

The distribution is gaussian (central limit theorem) with a mean

$\mu = \frac{<x_1>+<x_2>+\cdots+<x_N>}{N}$

and a width

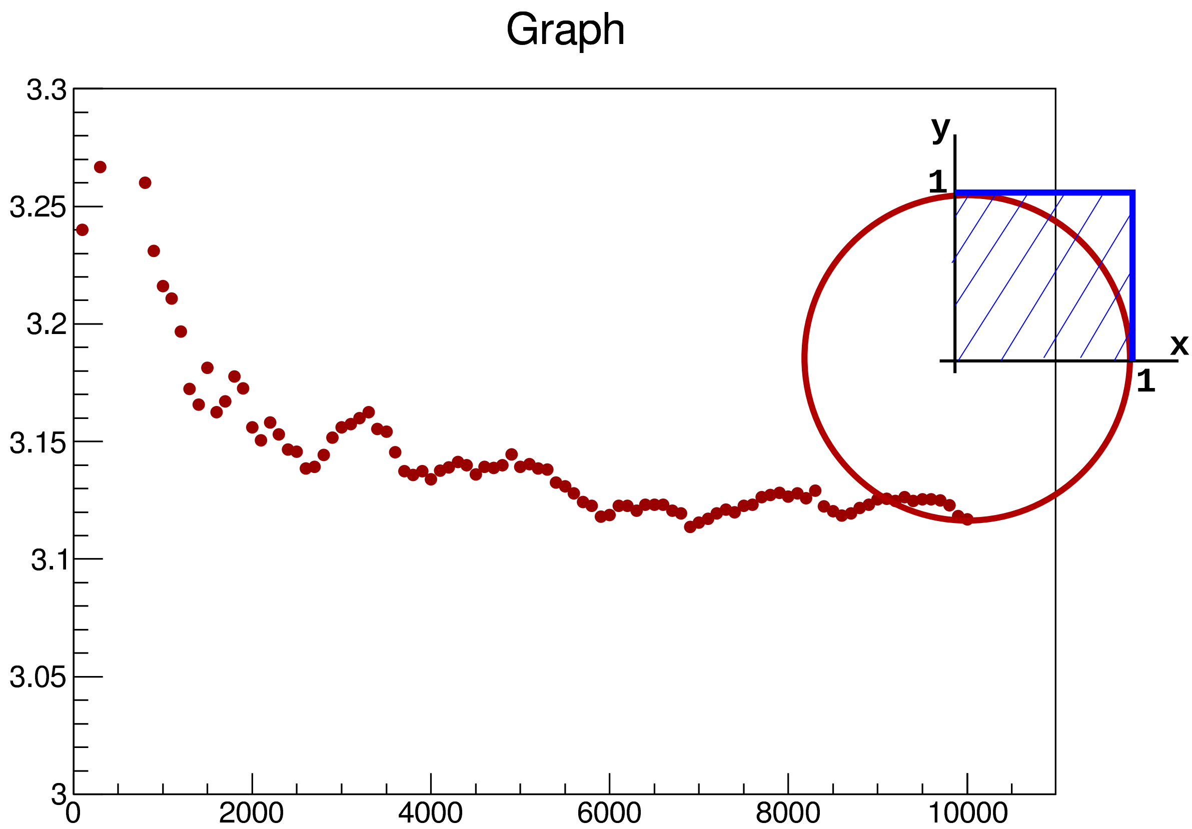
$\sigma <x> \sim \frac{\sigma_x}{\sqrt{n}} \sim 0.01$

the average of our 100 measurements is very precise

$\sigma_\mu = \frac{\sigma(<x>)}{\sqrt{100}} \sim \frac{0.01}{10} = 0.001$

This is equivalent to have 100*1000 measurements!

Pi evaluation



Monte Carlo integration

✔ we want to evaluate the following integral:

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

✔ remember that the expectation value of the function $f(x)$ for x distributed according to a PDF $p(x)$

$$\langle f \rangle = \int_a^b f(x)\, p(x)\, dx \quad \text{with: } \int_a^b p(x)\, dx = 1$$

✔ choosing x to be uniformly distributed in the interval $[a, b]$, one has:

$$\begin{aligned} p(x) &= \tfrac{1}{b-a} \\ \langle f \rangle &= \int_a^b f(x)\, p(x)\, dx = \tfrac{1}{b-a} \int_a^b f(x)\, dx \end{aligned}$$

MC integration

$$\begin{aligned} F &= \int_a^b f(x)\, dx \\ &= (b-a)\langle f \rangle \\ &= \tfrac{(b-a)}{N} \textstyle\sum_{i=1}^N f(x_i) \end{aligned}$$

x_i is a random variable uniformly distributed in the interval $[a, b]$

error estimation

$$\sigma_F = (b-a)\sigma_{\langle f \rangle}$$

$$\sigma_f^2 = \left\langle f^2 \right\rangle - \langle f \rangle^2$$

$$\sigma_{\langle f \rangle}^2 = \frac{\sigma_f^2}{N}$$

$$\boxed{\sigma_F = (b-a)\frac{\sigma_f}{\sqrt{N}} = \frac{(b-a)}{\sqrt{N}} \sqrt{\frac{1}{N}\sum_{i=1}^N \left(f(x_i)\right)^2 - \left(\frac{1}{N}\sum_{i=1}^N f(x_i)\right)^2}}$$

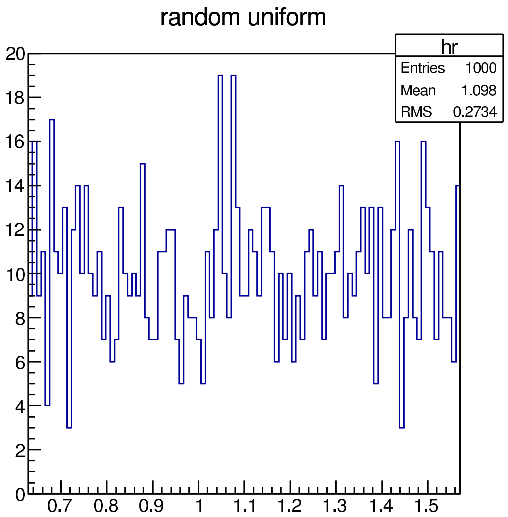
MC integration (cont.)

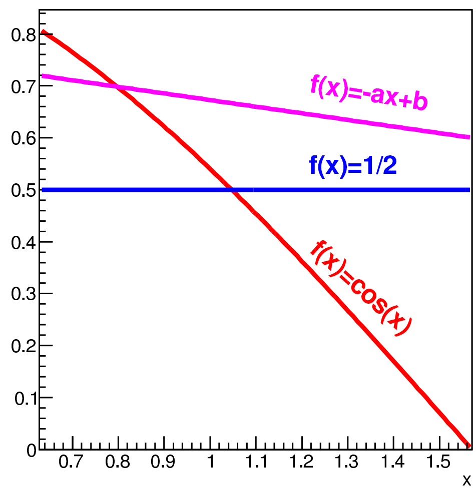
Let's compute the integrals of the functions:

$$\int_{0.2\pi}^{0.5\pi} \frac{dx}{2} = 0.471239$$
$$\int_{0.2\pi}^{0.5\pi} cos(x)\ dx = 0.412215$$
$$\int_{0.2\pi}^{0.5\pi} (ax+b)\ dx = 0.622035$$

Throwing 100 random variable uniformly distributed we obtain the following results:

$$\int_{0.2\pi}^{0.5\pi} \frac{dx}{2} = 0.471239 \pm 0.000000$$
$$\int_{0.2\pi}^{0.5\pi} cos(x)\ dx = 0.413671 \pm 0.007098$$
$$\int_{0.2\pi}^{0.5\pi} (ax+b)\ dx = 0.622280 \pm 0.001037$$





MC integration: algorithm

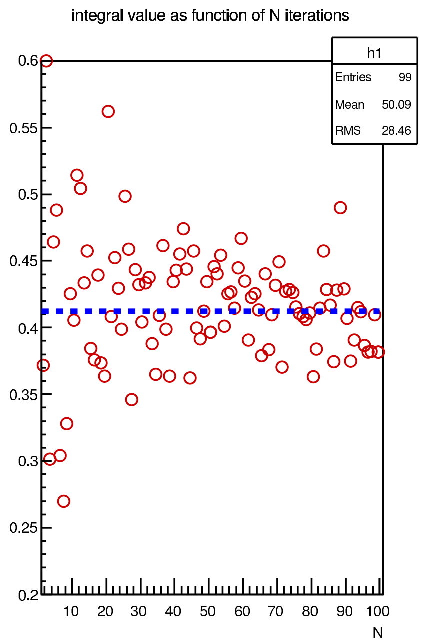
```
double xmin=TMath::Pi()*0.2;
double xmax=TMath::Pi()*0.5;
int N = 1000;

TF1 *f1 = new TF1("f1","TMath::Abs(cos(x))",xmin,xmax);
TF1 *f2 = new TF1("f2","0.5",xmin,xmax);
TF1 *f3 = new TF1("f3","-0.4/TMath::Pi()*x+0.8",xmin,xmax);
(...)
for (int i=0; i<N; i++) {
   double x = xmin + (xmax-xmin)*gRandom->Uniform();
   F1 += f1->Eval(x);
   F2 += f2->Eval(x);
   F3 += f3->Eval(x);
   f1s += f1->Eval(x) * f1->Eval(x);
   f2s += f2->Eval(x) * f2->Eval(x);
   f3s += f3->Eval(x) * f3->Eval(x);
 }
double f1m = F1/N; //mean
double f2m = F2/N;
double f3m = F3/N;
```

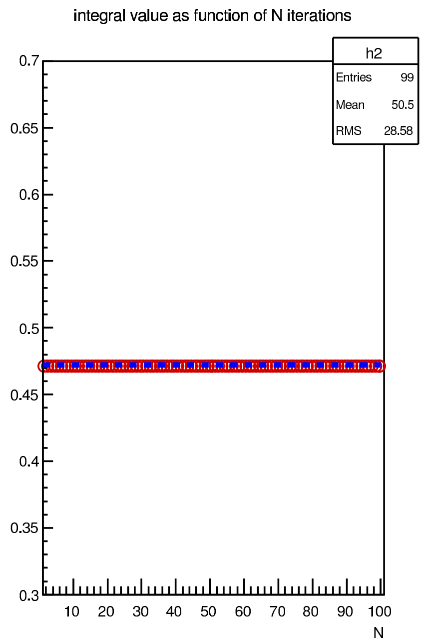
algorithm

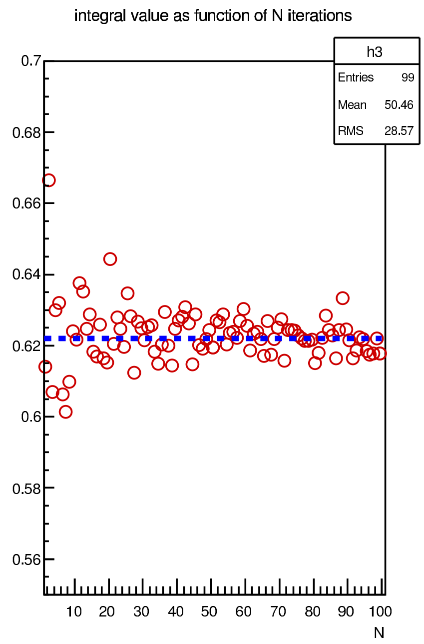
```
// integrals
double I1 = f1m*(xmax-xmin);
double I2 = f2m*(xmax-xmin);
double I3 = f3m*(xmax-xmin);

// variances
double Var1 = f1s/N - f1m*f1m;
double Var2 = f2s/N - f2m*f2m;
double Var3 = f3s/N - f3m*f3m;

// errors
double E1 = (xmax-xmin)/sqrt(N)*sqrt(Var1);
double E2 = (xmax-xmin)/sqrt(N)*sqrt(Var2);
double E3 = (xmax-xmin)/sqrt(N)*sqrt(Var3);
```

MC integration (cont.)

Let's check the integral value as function of the number of random variables generated N







$$F = \int_{0.2\pi}^{0.5\pi} cos(x)\, dx$$

$$F = \int_{0.2\pi}^{0.5\pi} \frac{dx}{2}$$

$$F = \int_{0.2\pi}^{0.5\pi} (ax + b)\, dx$$

Empty Slide

Reduction variance techniques

- ✔ The $cos(x)$ function varies much more in the interval of integration than the others
- ✔ Its integral value evaluation presents the largest variance. Why?
- ☞ Because we are sampling uniformly and the regions close to zero where the function is more important are sampled with the same importance as others where the function is smaller!
- ☞ In the framework of the **importance sampling technique** an additional pdf $p(x)$ can be used to rend the integrand smooth!

Importance sampling

- ✔ Rend smooth our integrand by applying a pdf $p(x)$

$$F = \int_a^b f(x)\, dx = \int_a^b \frac{f(x)}{p(x)}\, p(x)\, dx$$

- ✔ If the pdf is normalized in the integral interval $[a, b]$

$$\int_a^b p(x)\, dx = 1$$

and x is a variable distributed according to $p(x)$, then

$$\left\langle \frac{f}{p} \right\rangle = \int_a^b \frac{f(x)}{p(x)}\, p(x)\, dx$$

- ✔ Let's make a variable change

$$\int_a^b \frac{f(x)}{p(x)}\, \underbrace{p(x)\, dx}_{p(y)dy}$$

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

if y is distributed uniformly in $[0, 1]$ then

$\int_0^1 p(y)dy = 1 \Rightarrow p(y) = 1$

The transformation between $\mathbf{x}$ and $\mathbf{y}$ can be obtained by:

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

Importance sampling (cont.)

✔ From the transformation of variables we have a relation between x and y

$$\boxed{y = \int_a^x p(x')dx' \quad \Rightarrow \quad x(y)}$$

Generating a random variable y uniformly between $[0,1]$ and applying the transformation relation $x(y)$ we get random variables x distributed according to $p(x)$

$$\boxed{F = \int_a^b f(x)\,dx = \int_a^b \frac{f(x)}{p(x)}\,p(x)\,dx = \int_0^1 \frac{f[x(y)]}{p[x(y)]}\,dy = \left\langle \frac{f}{p} \right\rangle_y = \frac{1}{N}\sum_{i=1}^{N} \frac{f[x(y_i)]}{p[x(y_i)]}}$$

✔ Exercise: make the following integral

$$\int_{0.2\pi}^{0.5\pi} cos(x)\,dx$$

expected = 0.412215

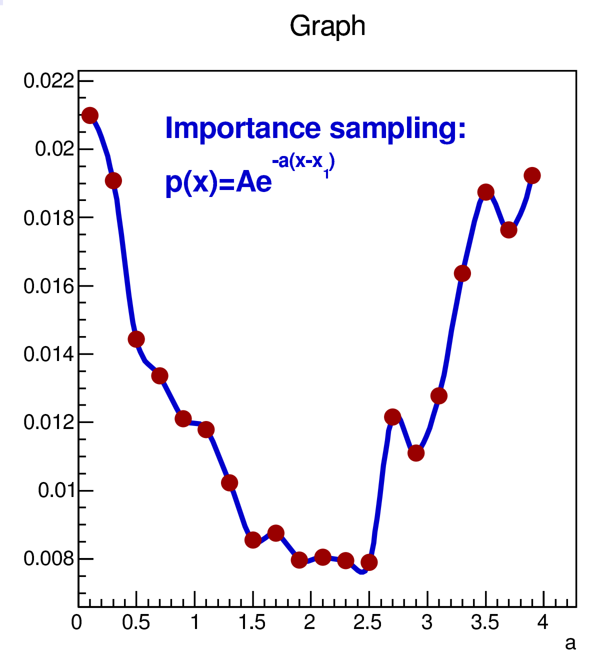
MC = 0.432225 +/- 0.025083 (100 deviates generated)

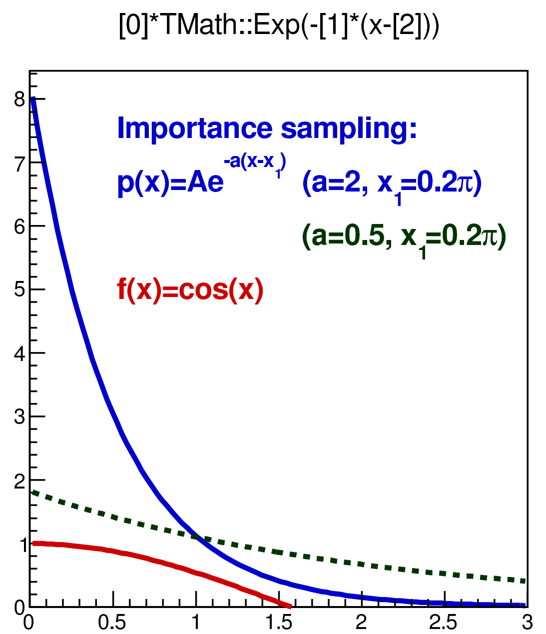
What about using importance sampling with a pdf: $p(x) \propto e^{-ax}$?

Importance sampling (cont.)

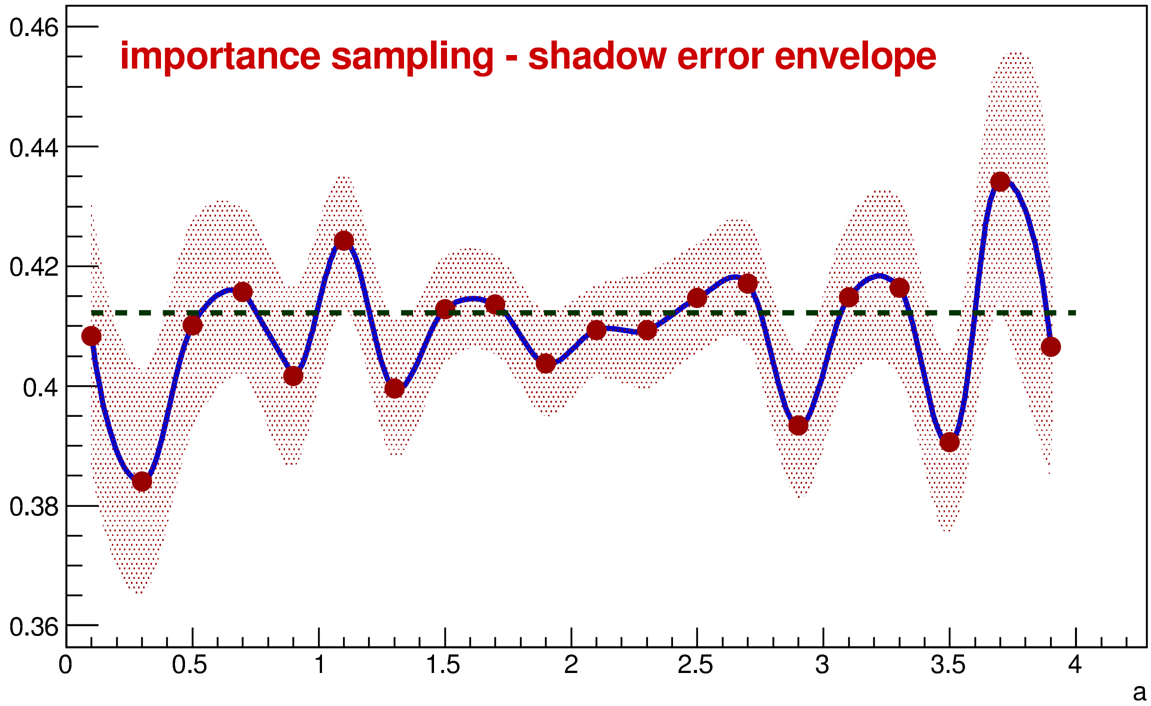
The function PDF shape matters?

Let's study the variation of the integral error with the a parameter of the exponential





Importance sampling (cont.)



Simulation

- ✔ **Simulation** is very important for understanding real situations or for modelling the behaviour of a system

  it is largely used on particle and astroparticle physics for designing the instruments used for particles detection

- ✔ the various real conditions the system has can be introduced easily in a simulated process

- ✔ Suppose you had to design a detector system for detecting photons coming from Compton scattering on a material?

  I assume my gamma source emits a beam very colimated along an axis (x for instance) and in between I have a block of material where Compton is going to happen...

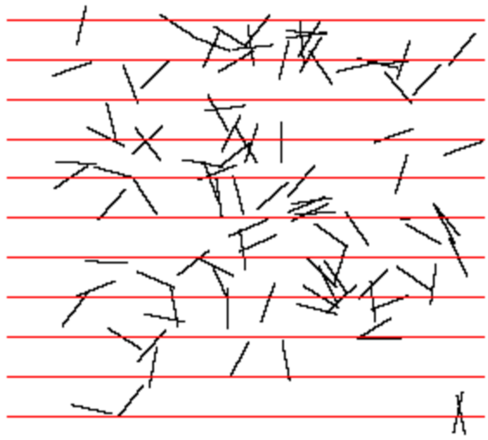
  What we need to know?

Buffon's needle problem

Buffon's needle problem is a question first posed in the 18th century by Georges-Louis Leclerc where simulation can help us a lot!

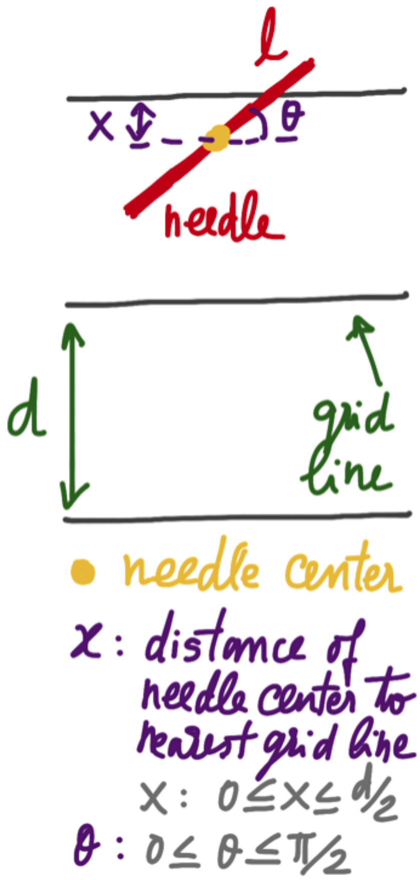
A needle of length ℓ is thrown randomly onto a grid of parallel lines, separated by a distance d, with $d > \ell$

What is the probability that a needle intersects a line?



Buffon's needle modelling

- ✔ Our phase-space is made of variables x and θ which are defined in the ranges, $x : [0, d]$ and $\theta : [0, \pi/2]$
their phase-space make a rectangle of sides d and $\pi/2$
- ✔ A needle crossing a line has to fullfill the condition,
$x < \frac{\ell}{2} \sin\theta$
where x is the needle center distance wrt nearest grid line
- ✔ The probability of crossing a line is calculated as the ratio between the two areas:
1) the area of the function $f(\theta) = \int_0^{\pi/2} \frac{\ell}{2} \sin\theta d\theta$
2) the area of the rectangle: $\pi/2 \; d/2$
$p = \frac{\int_0^{\pi/2} \ell \; \sin\theta \; d\theta}{\pi/2 \; d} = \frac{2\ell}{\pi d}$
- ✔ A more elaborated reasoning can be made based on joint probability density $p(x, \theta)$ which corresponds to the probability of having a given (x, θ) pair of values.
The fact that these variables are independent allow us to write: $p(x, \theta) = p(x) \; p(\theta)$



Buffon’s needle modelling

Making a simulation experiment:

- ✔ throw a needle: generate a random angle θ and a random needle center x
  $x : [0, d/2]$ and $\theta : [0, \pi/2]$
- ✔ a needle do cross a line if $x < \frac{\ell}{2} \sin(\theta)$
- ✔ count how many times a needle do cross a line and compute probability,
  $P_{crossing} = \frac{N^{events}_{crossing}}{N^{events}_{total}}$
- ✔ to be explored: π calculation !
  it comes from the fact that the probability of crossing a line depends on π