What we are trying to do:

(1) Evaluate (very) complicated integrals that represent probabilities of specific events occuring

(2) In the process of doing this, we essentially "simulate" the physical effects that influence the outcome of an experiment.

Examples of probability density functions:

Uniform: $f(x) = 1$ defined for $0 < x < 1$.

Exponential $f(x;\tau) = \frac{1}{\tau} e^{-t/\tau}$ defined for $t > 0$

Gaussian $f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-(x-\mu)^2/2\sigma^2}$ for $-\infty < x < \infty$.

All are normalized so that

$$\int f(x)\,dx = 1$$

where the range of integration is over the domain on which the distribution is defined.

Many other examples.

Angular distribution of muons in $e^+e^- \to \mu^+\mu^-$ scattering: $\frac{dN}{d\cos\theta} \sim 1 + \cos^2\theta$

How can we generate random numbers that are distributed according to a given p.d.f.?

So important that section 33 in 2004 Review of Particle Properties discusses this.

Two useful techniques that ASSUME you are able to generate uniformly distributed random numbers $0 < x < 1$ ...

Inversion method:

Recall that $F(x) = \int_{-\infty}^{x} f(x')\,dx'$ must have

$-\infty$ — or whatever.

a value $0 < F(x) < 1$ for all x on which $f(x)$ is defined.

The recipe is this:

Let u be a random number uniformly distributed between 0 and 1.

Then $x = F^{-1}(u)$ is distributed like $f(x)$.

Truely amazing. No proof given here.

Example: Generating an exponential distribution.

$$f(t) = \frac{1}{\tau} e^{-t/\tau}$$

$$F(x) = \int_0^x f(t)\,dt$$

$$= \int_0^x \frac{1}{\tau} e^{-t/\tau}\,dt$$

$$= -e^{-t/\tau}\Big|_0^x$$

$$= 1 - e^{-x/\tau}$$

Let $u = 1 - e^{-x/\tau}$ and solve for x:

$$e^{-x/\tau} = 1 - u$$

$$-x/\tau = \log(1-u)$$

$$x = -\tau \log(1-u)$$

If u is uniformly distributed between 0 and 1 then so is $(1-u)$.

So $x = -\tau \log u$ will be distributed like $f(x) = \frac{1}{\tau} e^{-x/\tau}$.

Try it out with something like, say,

$f(x) = \frac{1}{6} x(1-x)$ defined for $0 < x < 1$.

Caveats: Sometimes the inversion method is very efficient (ie, fast). Sometimes it is not.

Don't use it to generate a Gaussian:

$$F(x) = \int_{-\infty}^{x} \frac{e^{-u^2/2}}{\sqrt{2\pi}}\, du \quad \text{(sometimes written } \Phi(x)\text{)}$$

$$= \frac{1}{2}\left(\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) + 1\right)$$

$\operatorname{erf}(x)$ is often slow to evaluate ... so is $\operatorname{erf}^{-1}(u)$.

So, for Gaussians, use a library method, eg. in ROOT

TRandom :: Gaus() unit Gaussian

or TRandom :: Gaus(μ, σ)

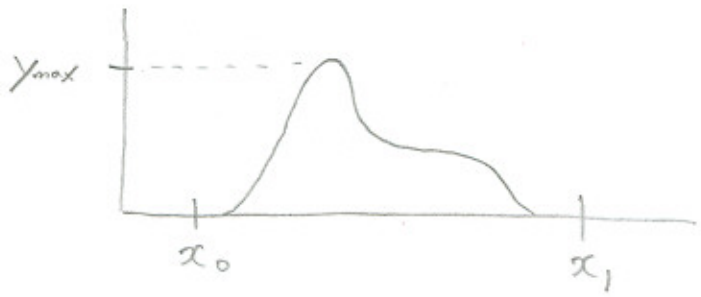
Or use (if you must know) one of the tricks described in the Review of Particle Properties, sec. 33.4.4 (2004 ed.).

The other method:

Acceptance - Rejection method:

Recipe:

Suppose the pdf you are trying to generate looks like this:



do {

generate x, uniformly distributed between x_0 and x_1.

generate u, uniformly distributed between 0 and y_{max}.

}

while ($u > f(x)$)

When x is in a spot where $f(x)$ is large, u will be more likely to be less than $f(x) \Rightarrow x$ is accepted.

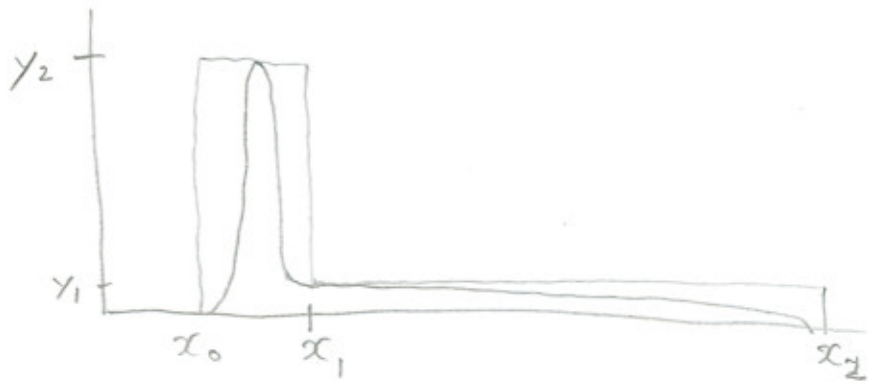
When $f(x)$ is small it will be more likely that $u > f(x) \Rightarrow x$ is rejected.

Sometimes can be very inefficient:

example:

Efficiency is essentially the ratio of the areas.

But you could divide the problem into two regions:



If $x_0 < x < x_1$, let u be uniformly distributed between 0 and y_2.

If $x_1 < x < x_2$, let u be between 0 and y_1.

Some algorithms automatically adjust these regions to optimize efficiency especially in multidimensional problems.

eg. VEGAS algorithm.

The connection between p.d.f.'s and histograms.

We use histograms to represent experimental data that is random by nature.

Example with C++ / ROOT :

```
#include "TH1.h"
  ⋮
TH1F *h = new TH1F("mass", "Invariant mass in Ge
                                  100, 1.0, 2.0 );
```

label

title

"pointer" to the histogram

of "bins"

low end

high end.

```
h->Fill( m, 1 );
```

Finds which bin contains m, adds 1 to that bin.

What is the correct way to label a histogram?

x-axis has the dimensions you specify, eg, GeV for mass, etc.

What is the y-axis?

If you integrate the histogram you should get N, the total number of entries.

$$N = \int_{x_0}^{x_1} h(x)\,dx$$

$$= \sum_{i=1}^{N_{bin}} h_i \,\Delta x_i$$

h_i / # of entries in bin i

Δx_i is the width of bin i

N is dimensionless, so h_i must have dimensions $[\Delta x]^{-1}$, eg. $0.01\ \text{GeV}^{-1} = /10\ \text{MeV}$.

Compare with the dimensions of a p.d.f. probability is dimensionless, so the dimensions of $f(x)$ are $[x]^{-1}$ since

$$\int f(x)\,dx = 1.$$

How can we compare a histogram with a p.d.f.?

Various methods:

(1) Multiply $f(x)$ by $N\Delta x$ so that $Nf(x)\Delta x$ can be compared with h_i when evaluated at the center of bin i.

(2) Integrate $f(x)$ over each bin and multiply by N.

(3) Divide h_i by $N\Delta x_i$. Then yaxis has dimensions purely of $[\Delta x]^{-1}$ rather than Δx^{-1} explicitly.