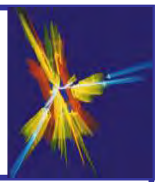


The Monte Carlo method

A numerical technique for calculating probabilities & related quantities using sequences of random numbers.

The usual steps:

(1) generate sequence $r_1, r_2, ..., r_m$ uniform in [0,1].

(2) use them to produce another sequence $x_1, x_2, ..., x_n$ distributed according to the distribution $f(x)$ of interrest

(3) use obtained x values to estimate some property of $f(x)$, e.g. fraction of x values within $[a,b] = \int_a^b f(x)dx$

$\Rightarrow$ MC calculation is a sort of integration (at least formally).

Usually trivial for 1D: $\int_a^b f(x)dx$ obtainable by other methods.

however MC more powerful for multi-dimensional integrals.

MC x values = ”simulated data”
$\rightarrow$ used for testing statistical procedures

MC methods a wide & own field in itself –

Here focus on the generation of arbitrary distributions & spend some time trying to answer the simple question: ”how can I generate the type of distribution I need?”.

More thorough & deeper reviews of field can be found in e.g.

V. Karimäki: Monte Carlo menetelmät – opintomoniste HU-SEFT 1993-01

Random number generators

Goal: to get uniformly distributed values in [0,1].

$\Rightarrow$ ”random number generator”

= computer algorithm to generate $r_1, r_2, \ldots, r_m$.

Example: multiplicative linear congruential generator (MLCG)

$$n_{i+1} = (a n_i) \bmod m, \qquad \text{where}$$

n_i = integer, a = **multiplier**, m = **modulus** & n_0 = **seed**.

NB! mod = modulus (remainder), e.g. 27 mod 5 = 2

n_i follow periodic sequence in $[1, m-1]$ $\Rightarrow$

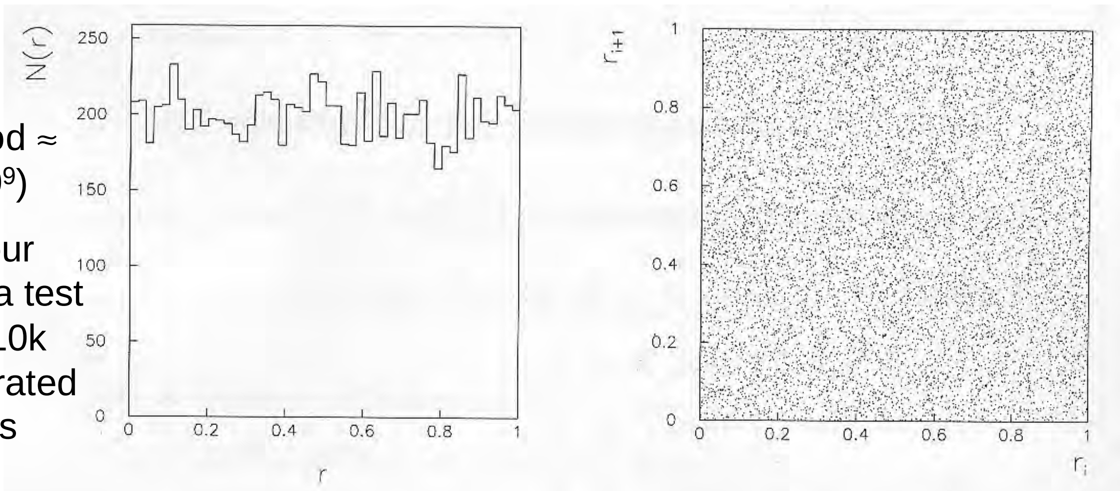
$r_i = n_i / m$ distributed in [0,1].

choose a & m so that r_i's pass various tests of randomness: uniform distribution in [0,1], all pairs independent (no correlations) & period long (maximum = $m-1$)

e.g. L’Ecuyer, *Commun. ACM* 31(1988)742: a = 40692, m = 2147483399

(period ≈ $2 \cdot 10^9$)

on your right a test with 10k generated values



Far better algorithms exist e.g. RANMAR, period ≈ $2 \cdot 10^{43}$.

Many good algorithms implemented in program libraries e.g. `RANMAR` & `RANLUX` in the CERN program libraries.

NB! r_i's like above in reality **pseudorandom numbers**

for more info see e.g. F. James, *Comput. Phys. Commun.* 60 (1990) 111

Exponential distribution:

Exponential distribution for a continuous random variable x is

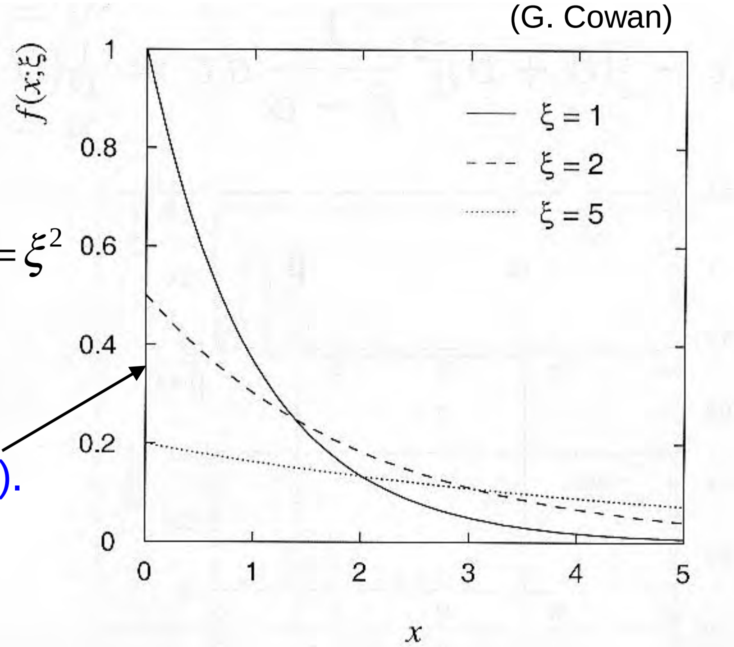
$$f(x;\xi) = \frac{1}{\xi} e^{-x/\xi} \quad (x \geq 0)$$

Exponential distribution characterized by only one parameter ξ.
Expectation value & variance of exponential distribution

$$E[x] = \int_0^\infty x \frac{1}{\xi} e^{-x/\xi} dx = \xi$$

$$V[x] = \int_0^\infty (x-\xi)^2 \frac{1}{\xi} e^{-x/\xi} dx = \xi^2$$

(G. Cowan)



Examples of exponential distributions (ξ = 1, 2 & 5).

Example: proper decay time of an unstable particle/state

$$f(t;\tau) = \frac{1}{\tau} e^{-t/\tau} \quad (t \geq 0)$$

τ = mean life time

Exponential distribution has unique feature – ”lack of memory”

$$f(t - t_0 \,|\, t \geq t_0) = f(t)$$

absolute starting (& end) point (”zero”) irrelevant

Very convient for any lifetime measurement in HEP

Gaussian (or normal) distribution:

Gaussian distribution for a continuous random variable x is

$$f(x;\mu,\sigma)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)\quad(-\infty<x<\infty)$$

Gaussian distribution is characterized by two parameters μ & σ.

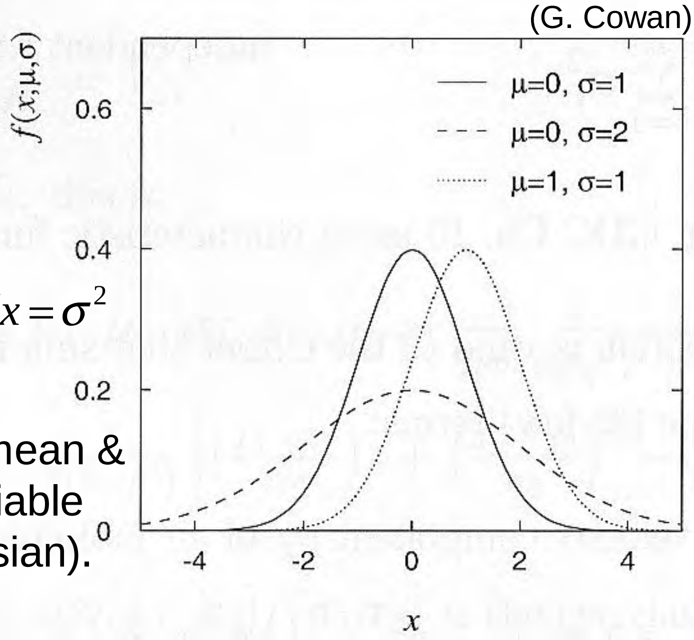
Expectation value & variance of gaussian distribution:

$$E[x]=\int_{-\infty}^{\infty}x\ f(x;\mu,\sigma)dx=\mu$$

$$V[x]=\int_{-\infty}^{\infty}(x-\mu)^2\ f(x;\mu,\sigma)dx=\sigma^2$$

NB! μ & σ often used for mean & spread of any random variable (i.e. not necessarily Gaussian).

(G. Cowan)



Special case: $\mu = 0$, $\sigma = 1$ ("standard Gaussian")

$$\varphi(x)=\frac{1}{\sqrt{2\pi}}\exp\left(-x^2/2\right),\qquad \Phi(x)=\int_{-\infty}^{x}\varphi(x')dx'$$

if y Gaussian distributed with μ & σ, then $x = (y-\mu)/\sigma$ follows $\varphi(x)$ & the cumulative distribution $F(y)$ related to $\Phi(x)$. No analytic expression for the cumulative distribution $\Phi(x)$ exists. Numerical evaluations of $\Phi(x)$ are tabulated & available in program libraries e.g. 68.3 % within 1σ, 90 % within 1.645σ, 95 % within 1.960σ, 99.7 % within 3σ etc… (for two-tailed Gaussians).

Why are errors often Gaussian ?

A consequence of the Central Limit Theorem (CLT). Look at behaviour of a variable that is the sum of several others. Irrespective of the distribution of the orginal variables, if one takes the sum X of N independent variables x_i, i = 1,..., N, each taken from a distribution with mean μ_i & variance V_i, the distribution for X has an expectation value & variance

$$E[X] = \sum_i \mu_i \qquad V[X] = \sum_i V_i$$

& becomes gaussian $N \to \infty$.

Note $V[X]$ equation above holds only for independent variables, formal proof of CLT tedious so we'll give a MC "proof" instead:

1 random number $\mu = 0.5$ & $\sigma^2 = 1/12$

2 random numbers $\mu = 1.0$

(R.J. Barlow)

3 random numbers $\mu = 1.5$

12 random numbers $\mu = 6.0$ & $\sigma^2 = 1.0$ (~gaussian)

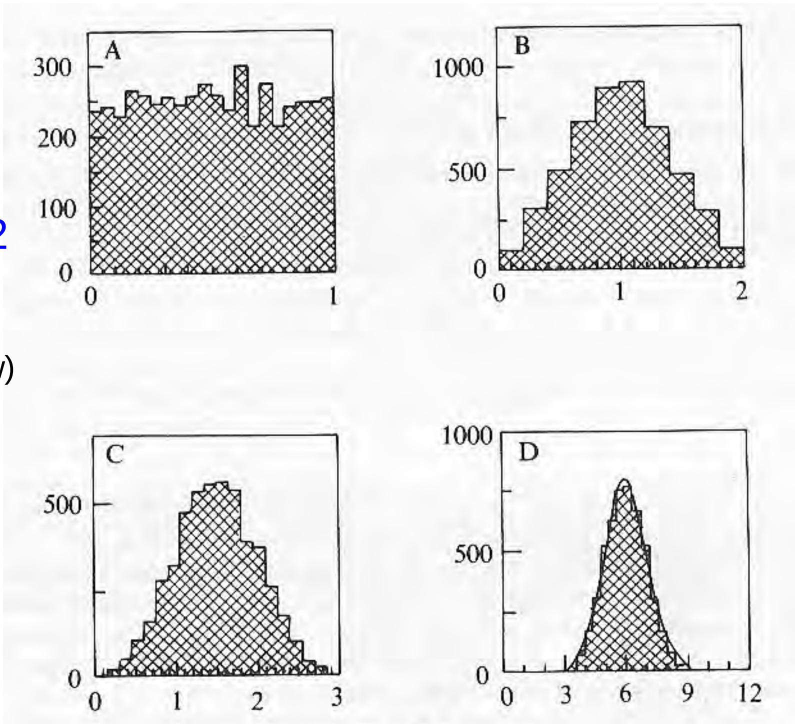


Fig. 4.1. The CLT at work.

Already after summing ~12 evenly distributed random numbers in [0,1] one obtains a Gaussian like distribution

Breit-Wigner distribution:

Cauchy distribution for a continuous random variable x is

$$f(x) = \frac{1}{\pi} \frac{1}{1+x^2}$$

Special case: the Breit-Wigner (common in particle & nuclear physics)

$$f(x;\Gamma,x_0) = \frac{1}{\pi} \frac{\Gamma/2}{(\Gamma/2)^2 + (x-x_0)^2}$$

where parameters x_0 & Γ are mass & width of a resonant state

Breit-Wigner distribution has a peculiar mathematical behaviour

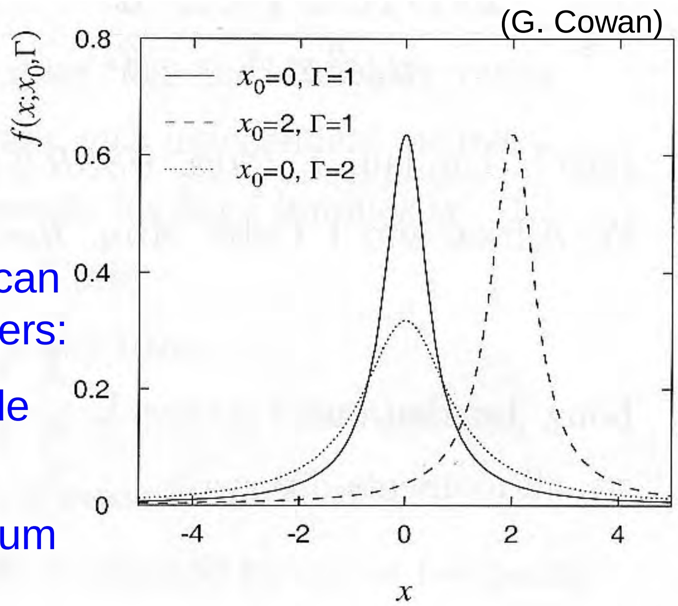
$E[x]$ = not well defined

$V[x] = \infty$

However the Breit-Wigner can be described by 2 parameters:

x_0 = peak position (i.e. mode or most probable value)

Γ = full width at half maximum

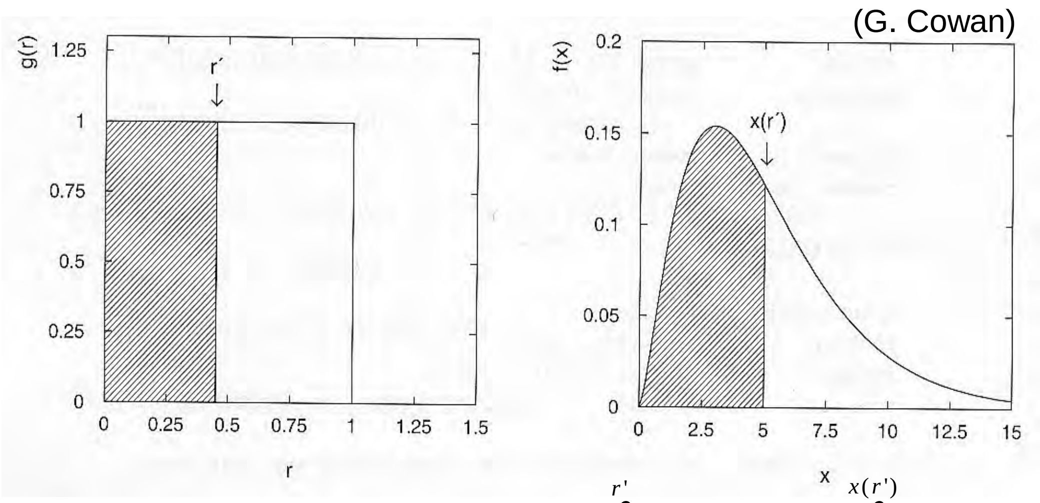


Example: describes a resonance (an unstable particle or state) e.g. the W gauge boson responsible for radioactive decays (or weak decays). Γ = decay width ($\propto$ inverse of mean life time).

NB! in practice the mean & variance are calculable for a physical phenomena described by a Breit-Wigner since in reality the tails of the distribution are finite due to energy conservation.

Inverse transform method

given r_1, r_2, ..., r_n uniform in [0,1], find x_1, x_2,...,x_n, which follow $f(x)$ by finding a suitable transformation $x(r)$.

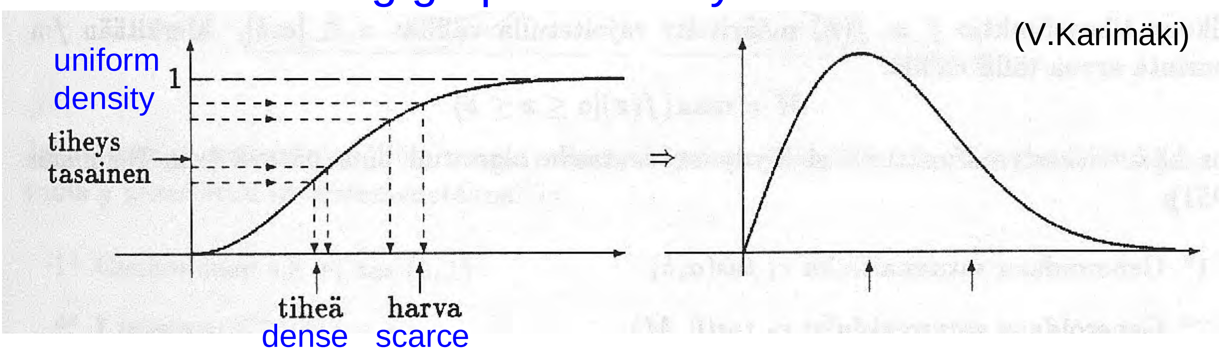


Require: $P(r \le r') = P(x \le x(r'))$ i.e. $\int_{-\infty}^{r'} g(r)dr = r' = \int_{-\infty}^{x(r')} f(x')dx' = F(x(r'))$

A general method that always works in case the inverse function of the cumulative distribution function $F(x)$ can be tabularized or is known. The generation steps for random number with the inverse transform method are:

- sample r from a uniform distribution [0,1]
- calculate $x = F^{-1}(r)$

Generated random numbers x obey the distribution $f(x)$. From the following graph it is easy to see that it works.



Inverse transform method

The inverse transform method can also be used for discrete distributions. One can tabularize the cumulative distribution function $F_j = \Sigma_{i=0}^{j}\, p_i$, j = 0, ..., N. If there are infinite number of probabilities p_i, then N has to be set so large that $F_N \approx 1$. The generation algorithm:

(i) sample r from a uniform distribution [0,1]

(ii) find k so that $F_{k-1} < r < F_k$.

The algorithm will generate integer numbers k whose distribution is proportional to the probability p_k.

Example of inverse transform method:

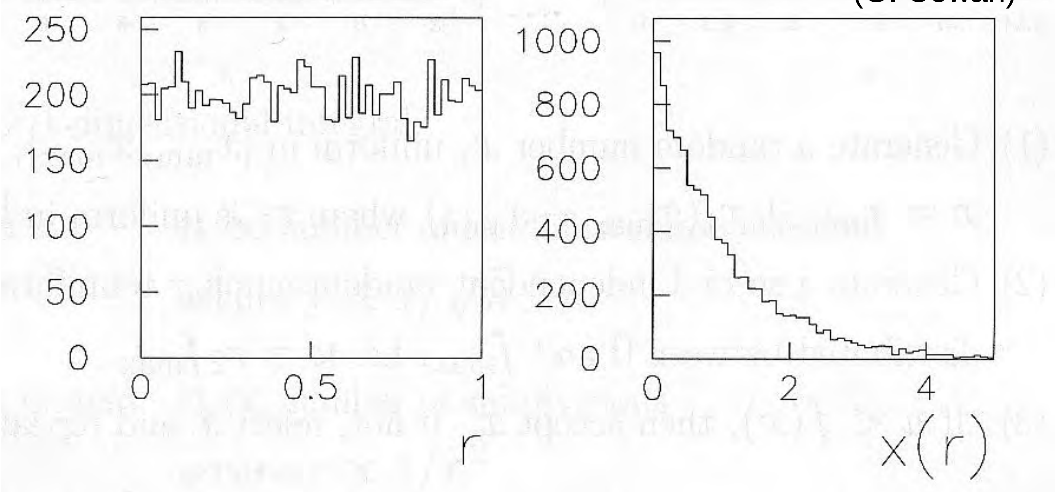
exponential pdf: $f(x;\xi) = \xi^{-1} e^{-x/\xi} \quad (x \geq 0)$

Cumulative distribution function: $F(x) = \int_0^x \xi^{-1} e^{-x'/\xi} dx' = 1 - e^{-x/\xi}$

Assume $r \in$ [0,1], now can set $r = F(x)$ & solve for $x(r)$ $\Rightarrow$

$$x(r) = -\xi \ln(1-r) \quad (\text{NB!} \quad x(r) = -\xi \ln r \quad \text{works too.})$$

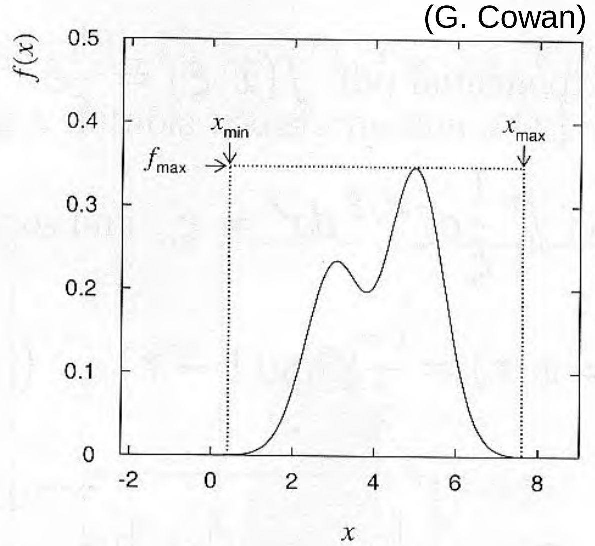
(G. Cowan)



Acceptance-rejection method (von Neumann)

Often an analytical solution impossible or impractical $\Rightarrow$ acceptance–rejection method (or **hit-or-miss**): enclose distribution in a box with height $f_{max} = \max(f(x))$

(G. Cowan)



(i) generate a random number x, uniform in $[x_{min}, x_{max}]$, i.e. $x = x_{min} + r_1(x_{max} - x_{min})$ where r_1 is uniform in [0,1]

(ii) generate a second independent random u uniformly distributed between 0 and $f_{max} = \max(f(x))$, i.e. $u = r_2\, f_{max}$.

(iii) if $u < f(x)$, then accept x. If not, reject x and repeat.

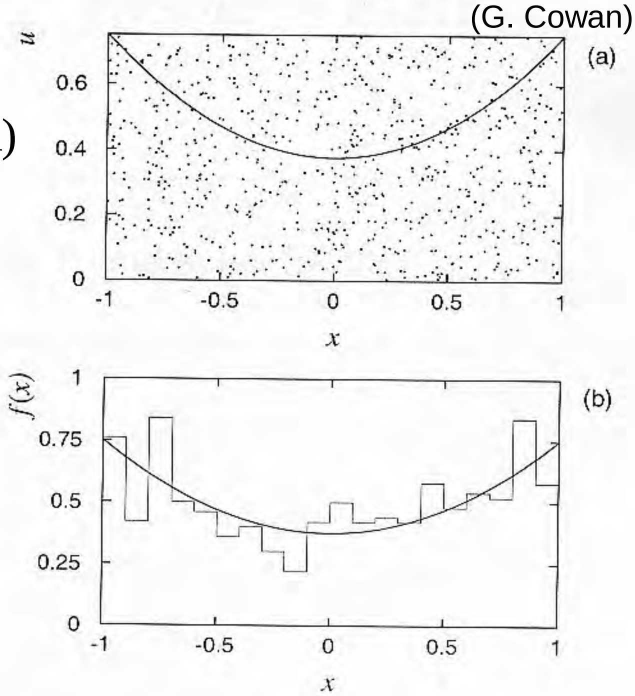
Example:

$$f(x) = \tfrac{3}{8}(1 + x^2) \quad (-1 \leq x \leq 1)$$

f_{max} = ¾; points that lie below curve are accepted. Distribution of accepted combinations shown below.

Efficiency of the algorithm depends on the area ratio of distribution to enclosing box. Algorithm inefficient for very "peaky" distributions.

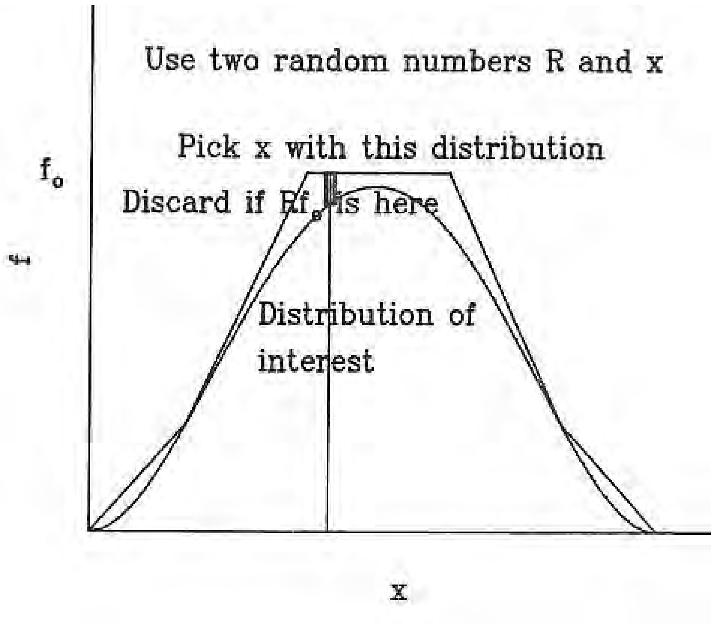
(G. Cowan)



Importance sampling:

To improve efficiency of acceptance-rejection method, generate first random number according to a distribution $g(x)$ such that $f_0(x) = Cg(x) > f(x)$ throughout interval. x choosen according to $g(x)$ are rejected if $uf_0(x) > f(x)$, where $u \in [0,1]$.

(B. Roe)



Isotropic direction in 3 dimensions:

Isotropy means the density is proportional to the solid angle, the differential angle element $d\Omega = d\cos\theta d\phi$. Hence generate uniform distribution in [–1,1] for $\cos\theta$ & uniform distribution in [0, 2π] for ϕ.

$$\cos\theta = (2u_1 - 1) \quad \& \quad \phi = 2\pi u_2, \text{ where } u_1 \,\&\, u_2 \text{ uniform in } [0,1]$$

Gaussian distributed random numbers: if u_1 & u_2 uniform in [0,1]

(a) $z_1 = \sin 2\pi u_1 \sqrt{-2\ln u_2}$ and $z_2 = \cos 2\pi u_1 \sqrt{-2\ln u_2}$

(b) construct $v_1 = 2u_1 - 1$ & $v_2 = 2u_2 - 1$ (uniform in [-1,1]), if $r^2 = v_1^2 + v_2^2 > 1$ start over again, otherwise

$$z_1 = v_1\sqrt{-2\ln r^2 / r^2} \quad \text{and} \quad z_2 = v_2\sqrt{-2\ln r^2 / r^2}$$

z_1 & z_2 are independent & Gaussian distributed with $\mu = 0$ & $\sigma = 1$

$z_i' = \mu + \sigma z_i$ are Gaussian distributed with mean μ & variance σ^2

NB! many Gaussian algorithms implemented in program libraries.

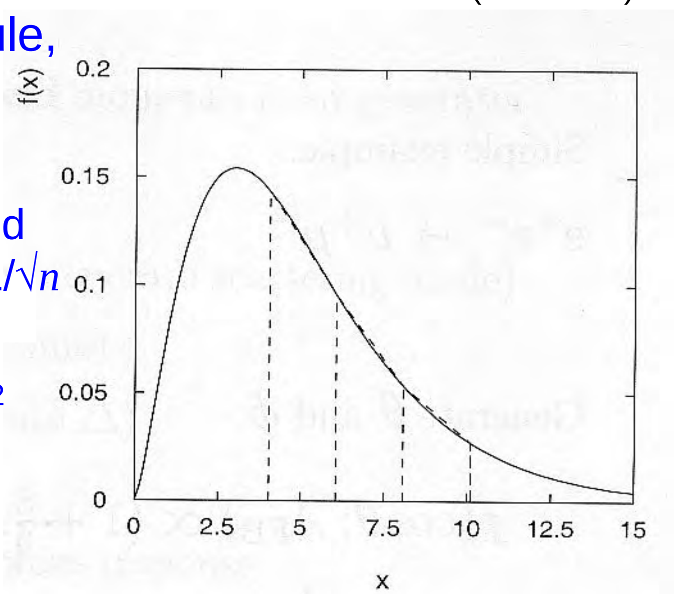
Poisson distribution: (iterate until a successful choice made):
begin with $k = 1$ and set $A = 1$ to start
(i) generate u. replace A with uA
(ii) if $A < \exp(-\nu)$, where ν is the mean of the Poisson distribution, accept $n_k = k - 1$ and stop.
(iii) increment k by 1 and repeat (i).

For large ν (> ~10) it may be satisfactory (& much faster) to approximate Poisson distribution by a Gaussian distribution. Generate z from $f(z;0,1)$ & then accept $x = \max(0,[\nu + z\sqrt{\nu} + 0.5])$ where [ ] signifies greatest integer ≤ the expression in [ ].

Accuracy of Monte Carlo methods:

MC calculation = integration.
compare to trapezoidal rule,
n = # of computing steps

(G. Cowan)



For 1-dimensional integral:
MC: $n \propto$ number of generated random values, accuracy $\propto 1/\sqrt{n}$

trapezoid: $n \propto$ number of subdivisions, accuracy $\propto 1/n^2$

In 1D trapezoid wins!
But in d dimensions:

MC: accuracy $\propto 1/\sqrt{n}$ ← independent of d!
trapezoid: accuracy $\propto 1/n^{2/d}$

MC wins for $d > 4$. Gaussian quadrature better than trapezoid but for high enough d, MC always wins!!
(see e.g. F. James, *Rep. Prog. Phys.* 43 (1980) 1145).