Air Shower Simulations

Johannes Knapp,
Physics & Astronomy
U of Leeds, UK

3rd School on
Cosmic Rays and Astrophysics
Arequipa, Peru
2008

- Part 1: Astroparticle Physics, Air Showers and Simulations
- Part 2, 3: Hadronic & Nuclear Models, CORSIKA, Performance and Limitations
- Part 4: Selected Aspects of EAS Simulations, Simulation Techniques

Air Shower Simulations

Johannes Knapp,
Physics & Astronomy
U of Leeds, UK

3rd School on
Cosmic Rays and Astrophysics
Arequipa, Peru
2008

- Part 1: Astroparticle Physics, Air Showers and Simulations
- Part 2, 3: Hadronic & Nuclear Models, CORSIKA, Performance and Limitations
- Part 4: Selected Aspects of EAS Simulations, Simulation Techniques

The Atmosphere

All other components
(e.g. CO_2, H_2O, ...) are << 1%
and can be neglected.

composition: 78.0% N_2

21.5% O_2

0.5% Ar

barometric formula: $\rho(h) = \rho(0) \exp(-h/h_0)$

$\rho(0) = 0.00123\ g/cm^3$ at sea level

$h_0 \sim 7\ km$

Total (vertical) thickness T of atmosphere: $\sim 1000\ g/cm^2$

Pressure at sea level: 1 bar 1 mbar = 1 hPa

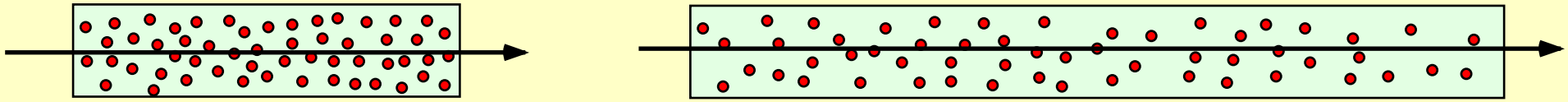
$P \sim T \cdot g = 1000\ g/cm^2 \cdot 9.81\ m/s^2 \sim 10^5\ N/m^2 = 10^5\ Pa$

Chacaltaya (5200 m a.s.l.):
$r = 0.59\ r_0$, $T = 538\ g/cm^2$

(temperature complicates things....
"adiabatic atmosphere")

The natural scale for the shower development is

the atmospheric thickness t (in g/cm^2), not the height (in cm).



(the number of scattering centers matters, not the distance travelled.)

Characteristic lengths: x_0 radiation length, λ_0 interaction length

Atmospheric Thickness (=mass overburden)

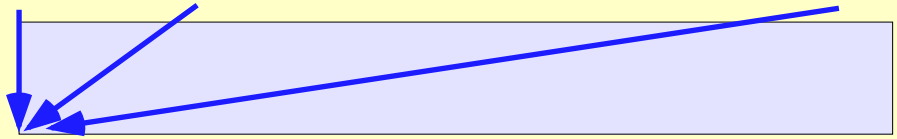
$$T(h) = \int_h^\infty \rho(h)\, dh = \rho(0)\, h_0 \exp(-h/h_0)$$

Shower development: $N(t) \sim c \cdot t^a \cdot \exp(-bt)$

T increases with zenith angle θ

$T(\theta) \sim T(0) / \cos\theta$ for $\theta < 70^o$

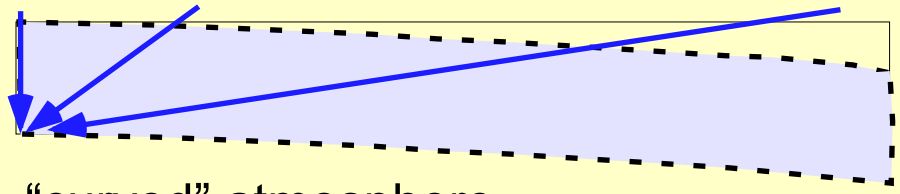
$T(\theta) \sim \infty$ for $\theta = 90^o$



"flat" atmosphere

$T(\theta) < T(0) / \cos\theta$ for $\theta < 70^o$

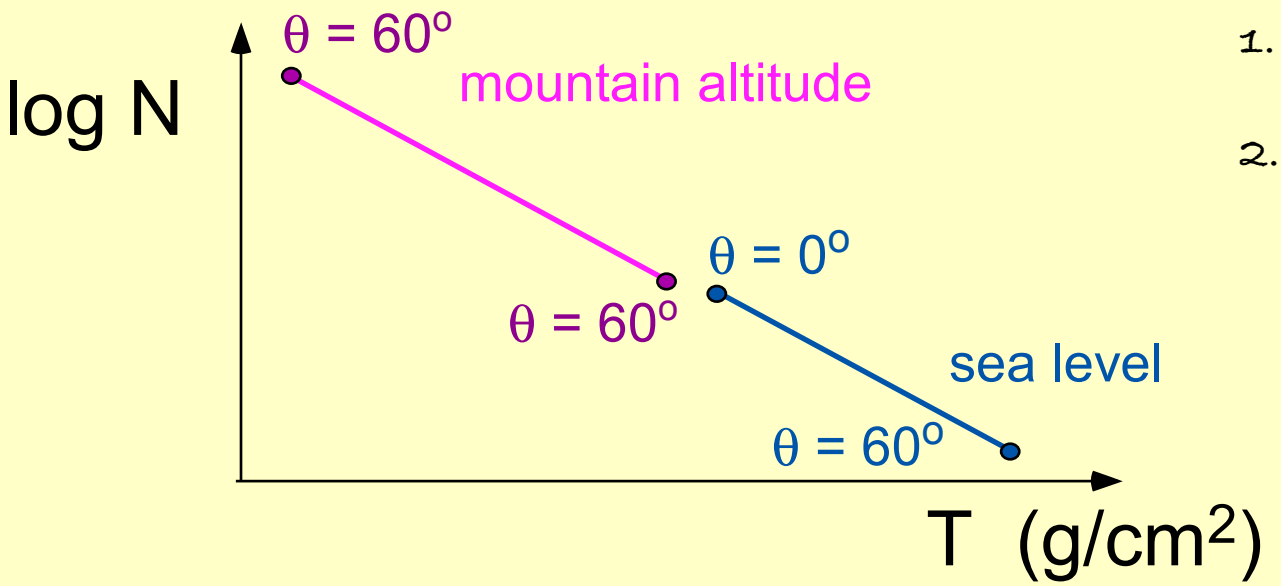
$T(\theta) \sim 36\, T(0)$ for $\theta = 90^o$



"curved" atmosphere

A vertical 10^{20} eV shower has its maximum near sea level ($T_{max} \sim 1000$ g/cm^2).

A 10^{20} eV shower at $>70^o$ ($T > 3000$ g/cm^2) is almost completely absorbed in the atmosphere (just muons survive).



1. Compare experiments at different altitudes.
2. Study the longitudinal shower development by varying q and look for signatures with the same frequency (constant intensity method)

The Atmosphere in CORSIKA

US Standard Atmosphere (parameterized by J. Linsley in 5 layers)

> 100 km	linear
40 ... 100 km	exponential density variation
10 ... 40 km	
4 ... 10 km	
0 ... 4 km	

78.0%	N_2
21.5%	O_2
0.5%	Ar

$\rho(h)$, $T(h)$, $h(T)$ } implemented as functions

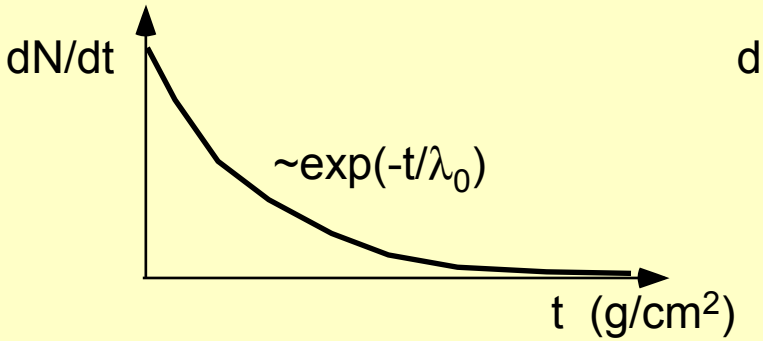
$\rho(0) = 0.00123$ g/cm^3

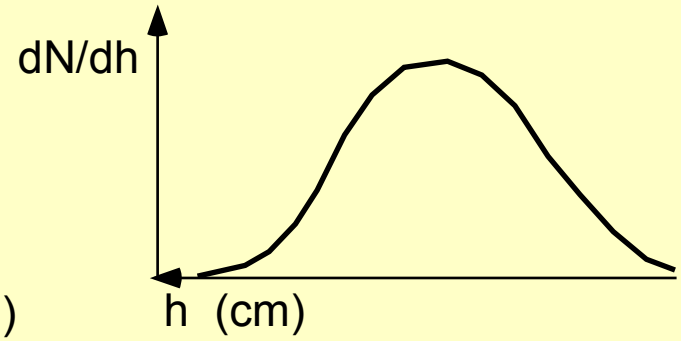
$T(0) = 1036.1$ g/cm^2

$h(T=0) = 112.8$ km

Position of next interaction follows exponential in t

but a small dt may correspond to huge dh since $T \sim \exp(h/h_0)$





Decay or Interaction ?

Position of next interaction of a particle depends on the matter traversed (i.e. T).

$dN/dx \sim \exp(-x/\lambda_o)$

Position of decay of a particle depends on time past (i.e. t) or distance travelled (s).

$dN/dt \sim \exp(-t/\tau)$ and since $t = s/c = x/\rho c$ $dN/dx \sim \exp(-x/\rho c\tau)$

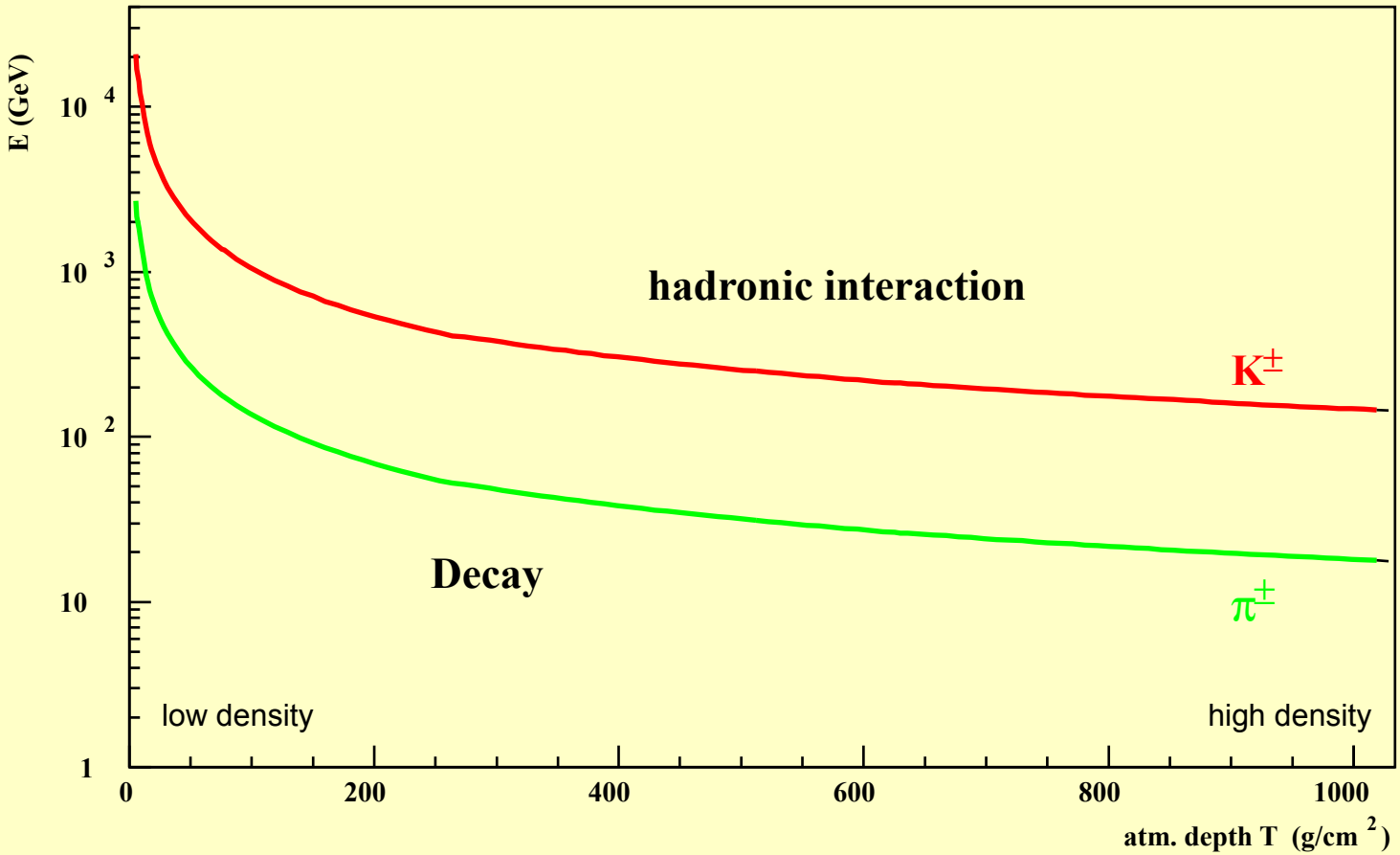
What if interaction and decay compete, e.g. for $\pi^{\pm}$?

in MC very simple: Calculate independently a decay point and an interaction point.
Select the mechanism that happens first.

1.) draw a random path x_i from the distribution $\exp(-x/\lambda_o)$ to get interaction point
2.) draw a random time t_d from the distribution $\exp(-t/\tau)$ to get decay point
3.) convert time t_d into a distance $x_d = t_d\, c\, \rho$.
4.) if $x_i < x_d$, then an interaction happens, otherwise a decay.

This is actually how nature does it.

Decay or Interaction ?

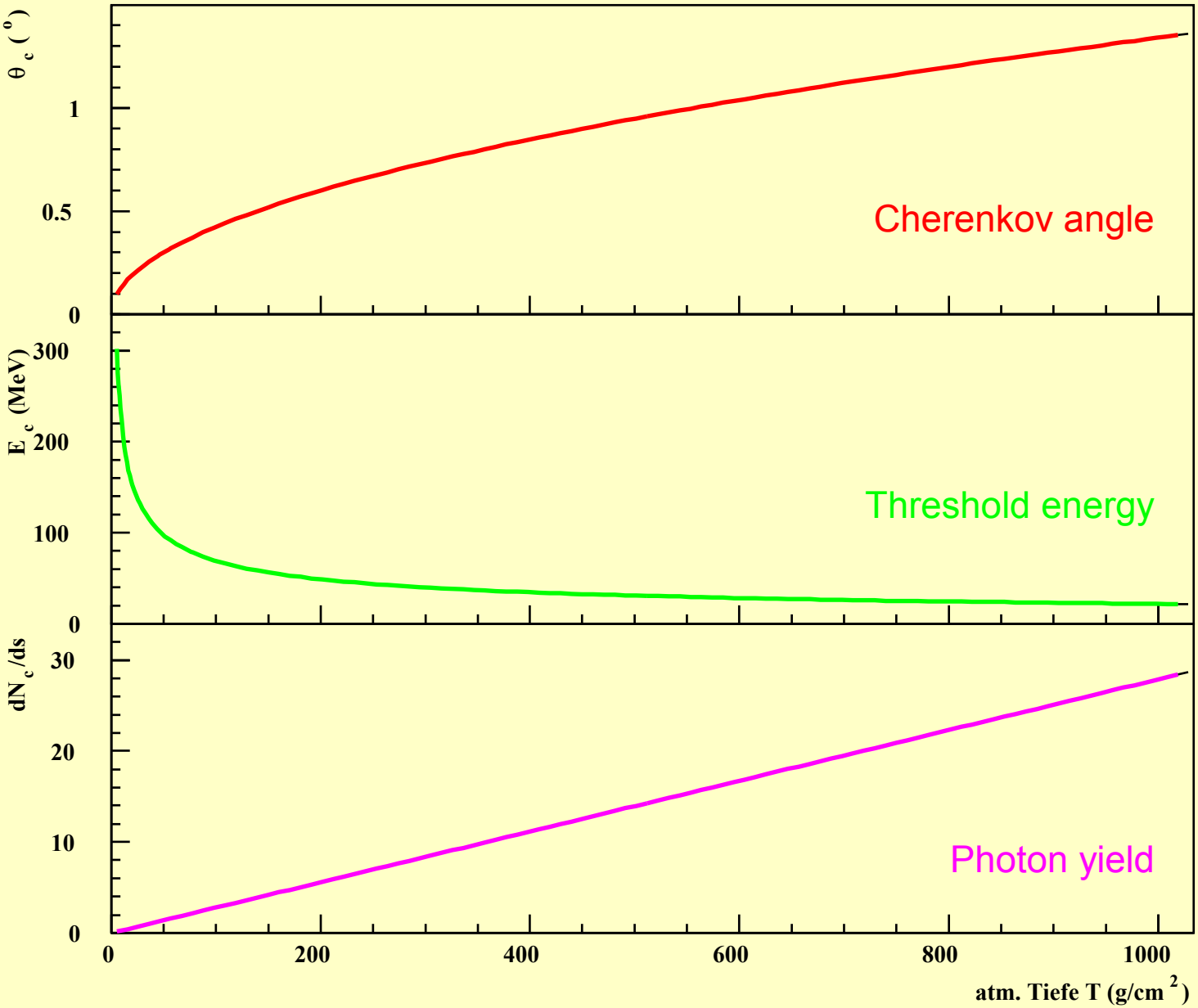


decay: $p_d(t) \sim \exp(t/\tau)$ τ = lifetime

interaction: $p_i(x) \sim \exp(x/\lambda)$ λ = interaction length

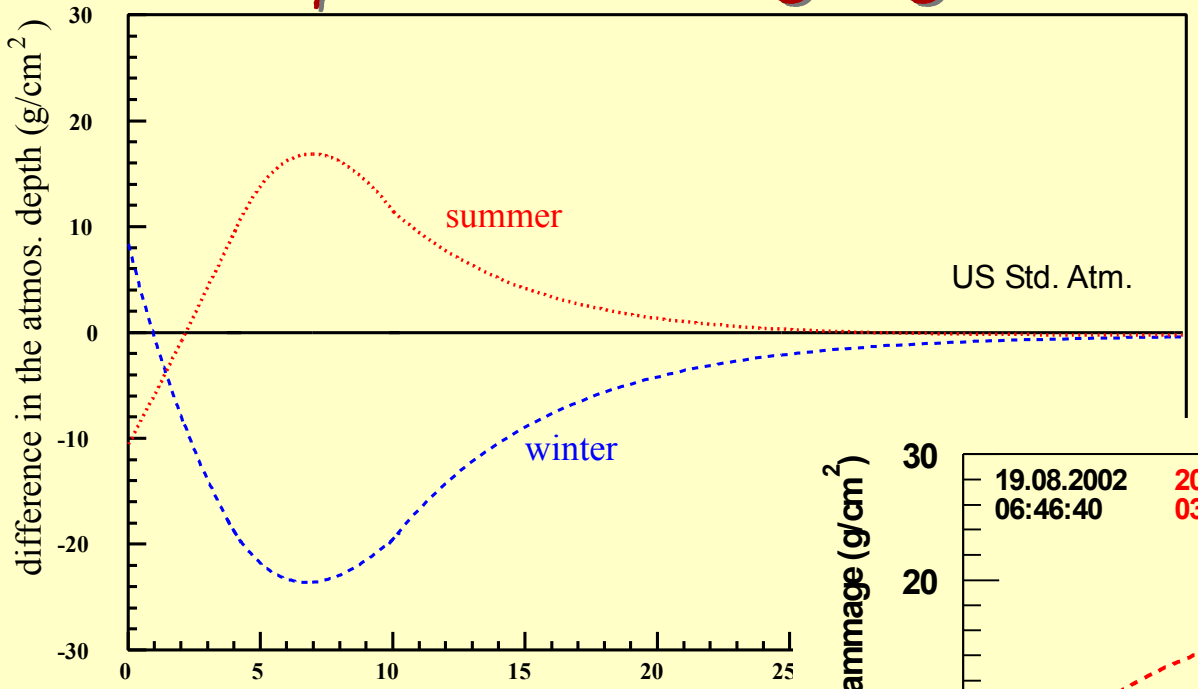
analytic: $p_{tot}^{-1} = p_d^{-1} + p_i^{-1}$ difficult, since exponentially varying atmospheric density

Cherenkov Light Production in Air



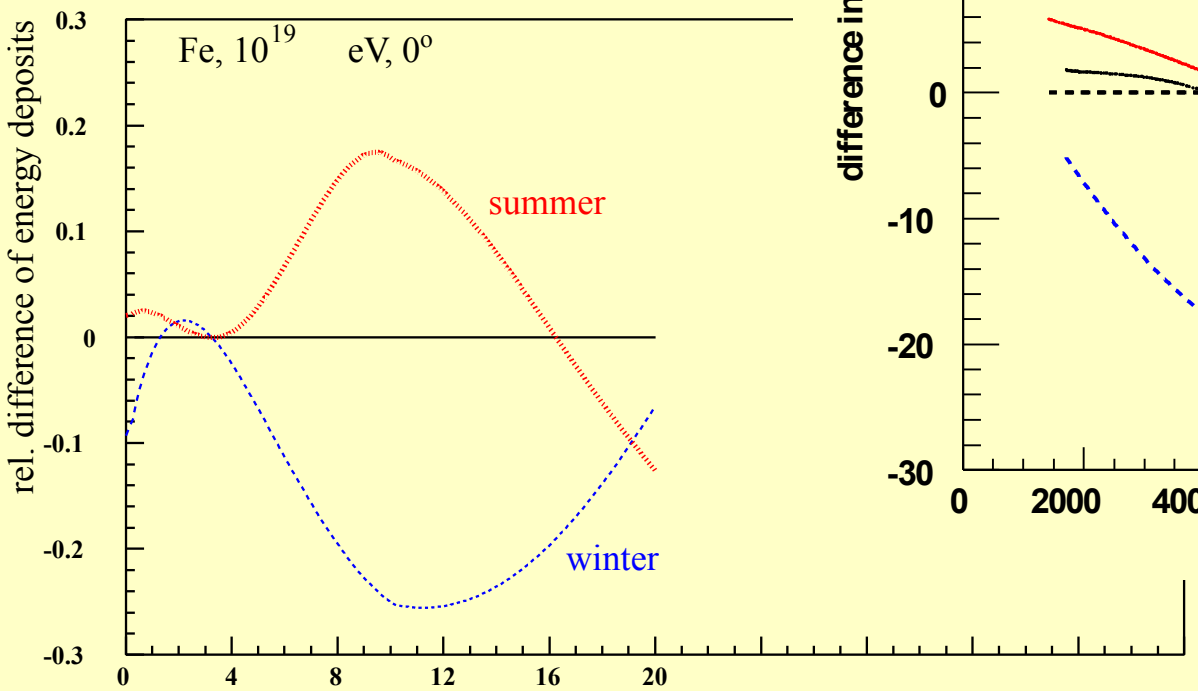
needs to be folded with the particle distribution

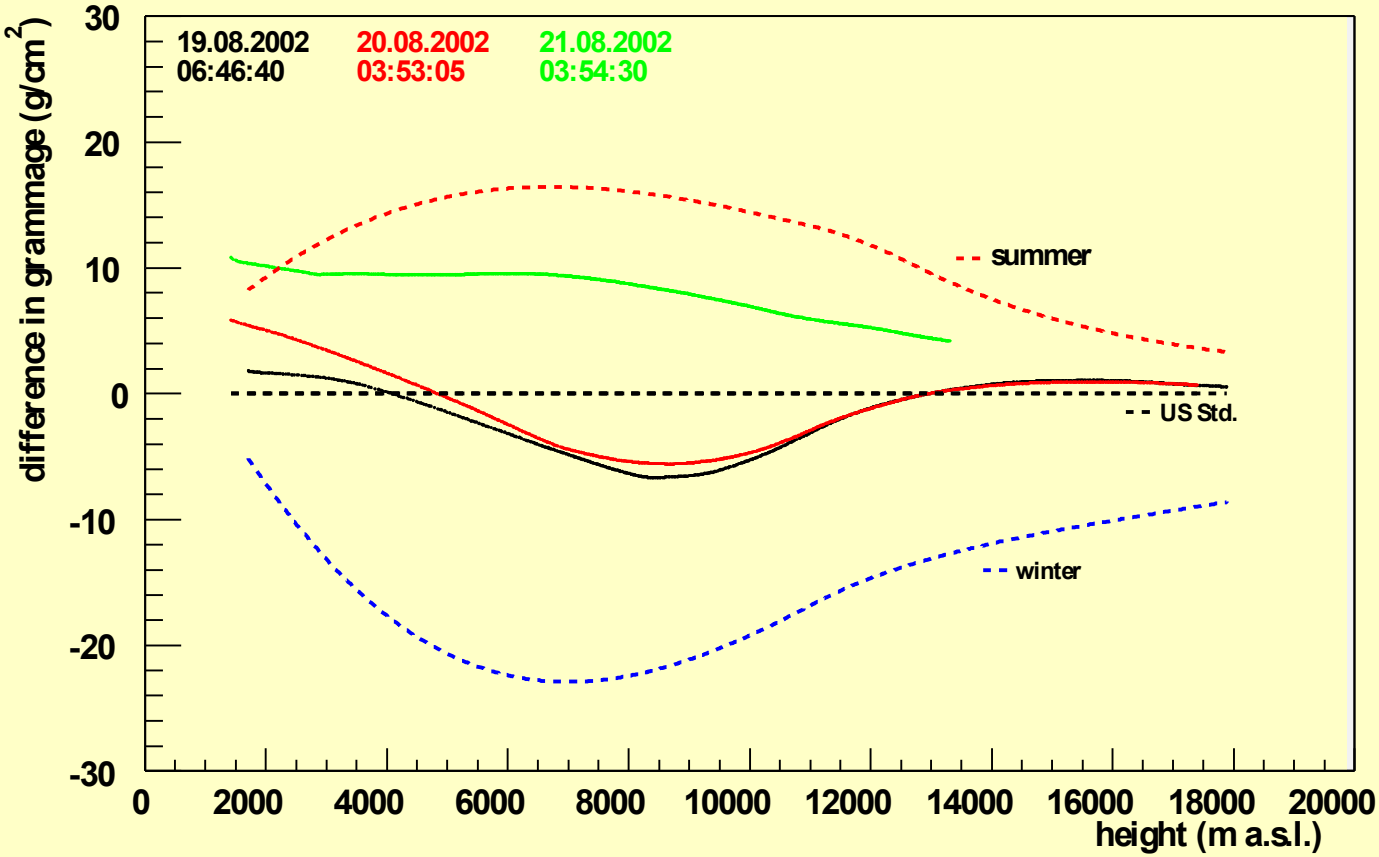
Atmosphere is changing all the time ....



Total thickness of absorber varies as well as the layering.
Variation of atmospheric conditions:

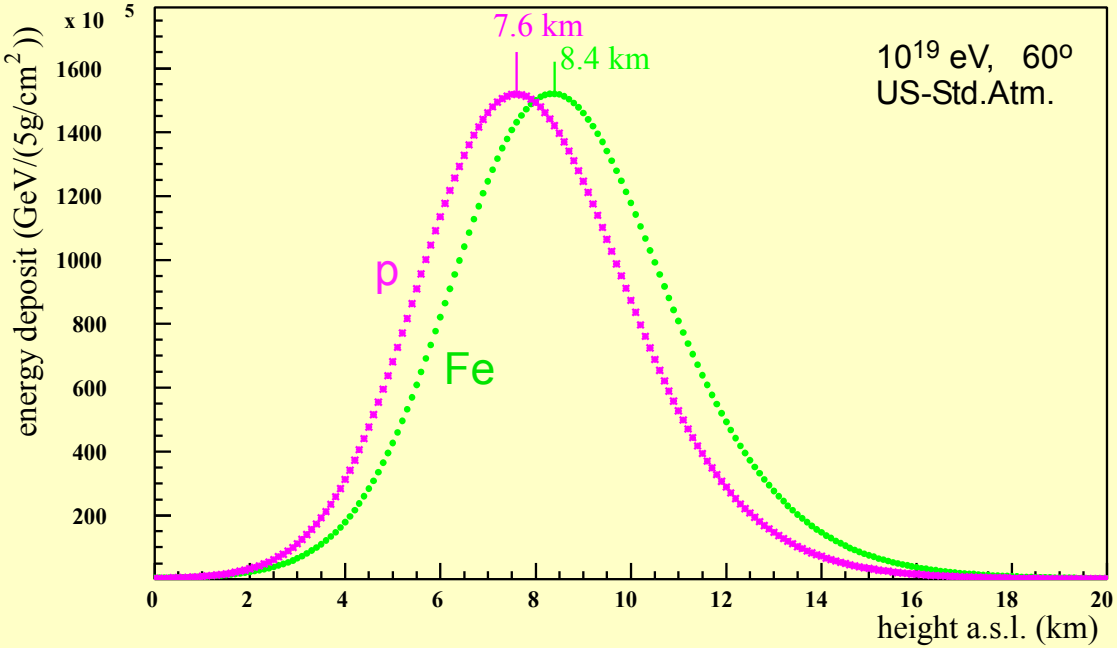
(to be measured by balloon sondes at the site,
e.g. after each > 10^{20} eV event ?)

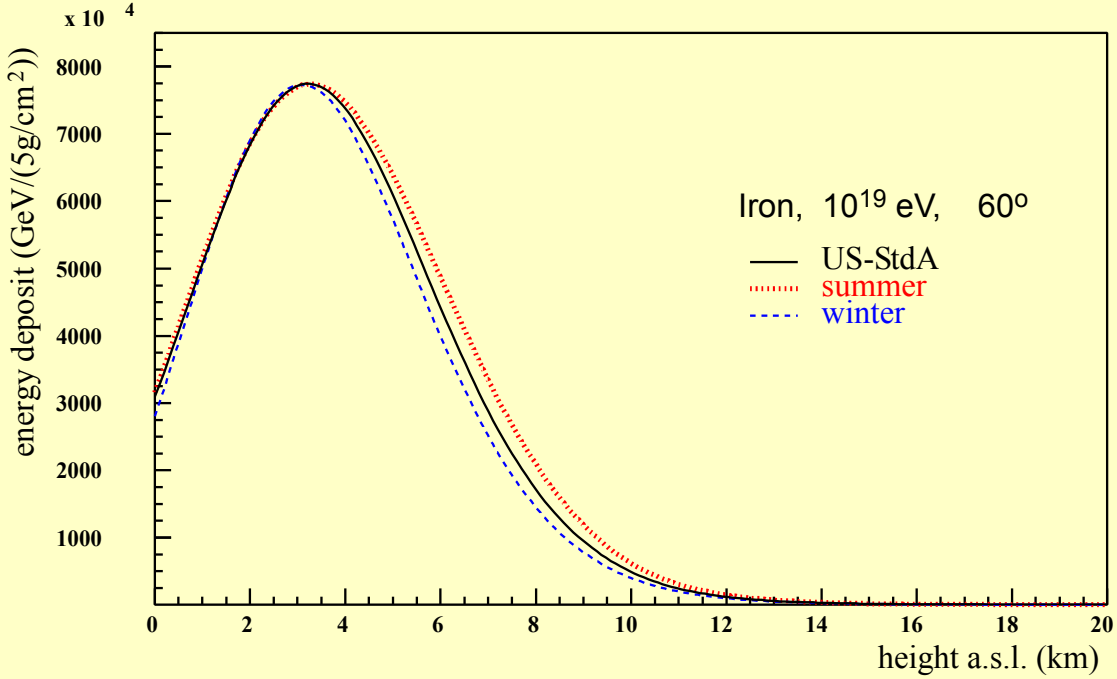


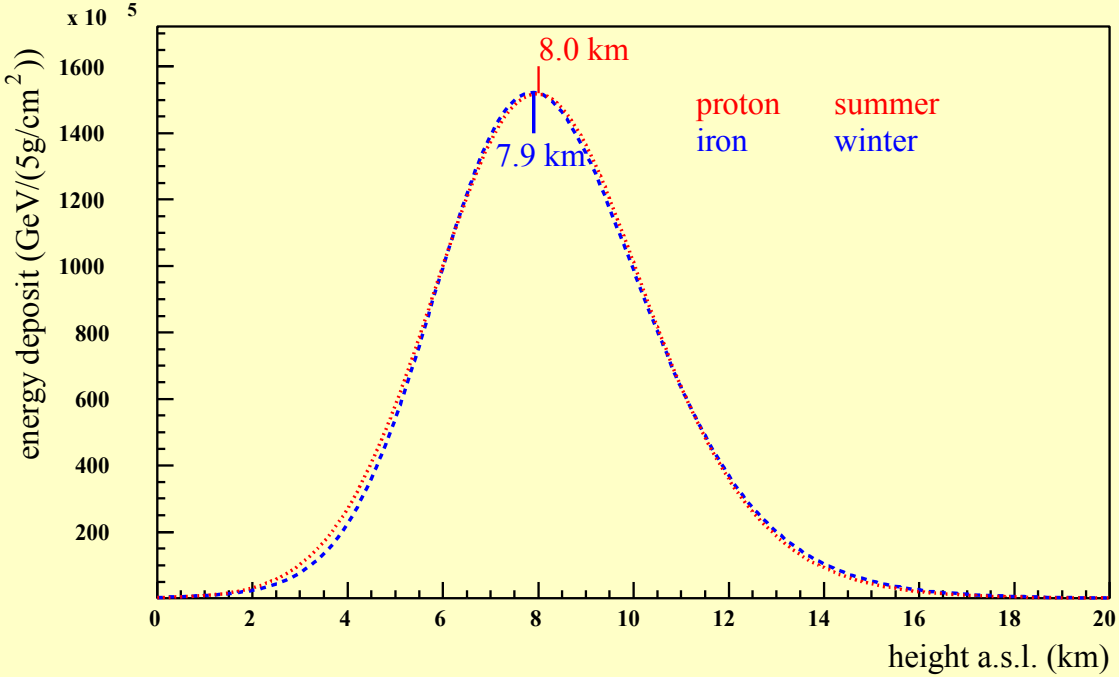


B. Keilhauer

X_{max} and the atmosphere







Simulation Speed-Up

Computing time ~ 1 h x $E/10^{15}$ eV Disc space ~ 300 MB x $E/10^{15}$ eV per shower.

At 10^{20} eV : more than 10^{11} secondaries !!!

excessive resources needed per shower : ~ 10^5 h = 11 years ~ 30 Tera Bytes

No way (nor need?) to follow them all: statistical sampling "Thinning"

Standard method:

(by A.M. Hillas)

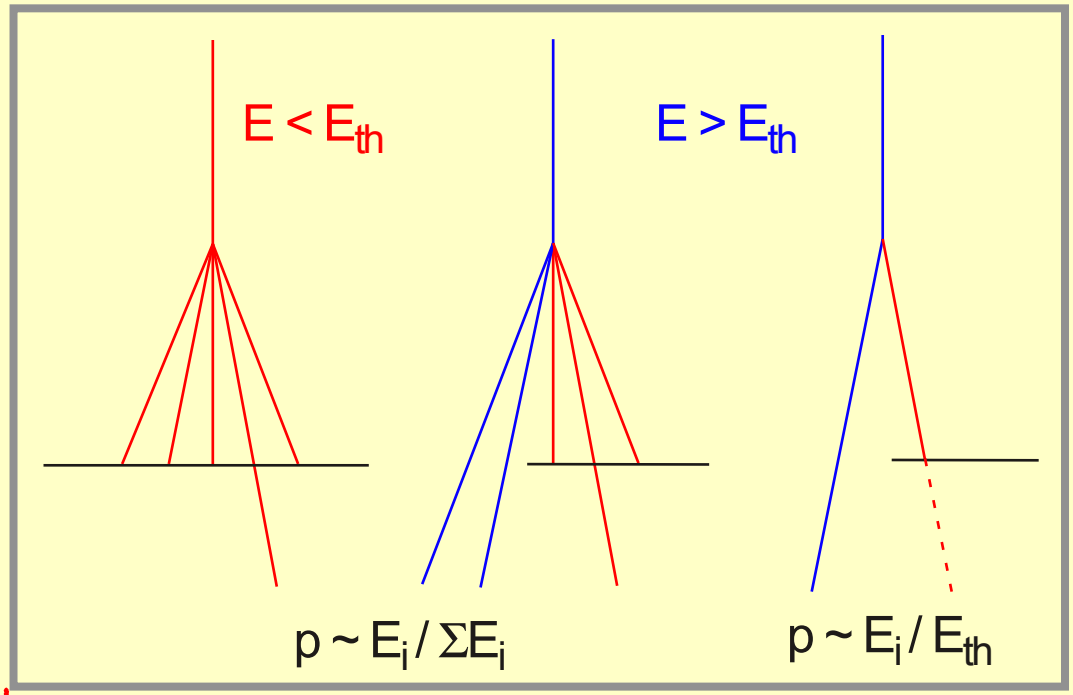
define the thinning threshold $E_{th} = \varepsilon_{th} E_0$

$E > E_{th}$ follow each particle

$E < E_{th}$ follow only one (or few),

but give it a weight to account for the discarded particles: $w_i' \sim w_i \, 1/p$

+ energy is conserved
+ mean $N_{e,\gamma,\mu}$ are preserved
- but : fluctuations are enlarged artificially



Added weight increases output per particle (8 words instead of 7),

but computing time and number of particles to be written out are greatly reduced.

Which is the right thinning level?

longitudinal development N(t):

very many particles in the shower core, i.e.

low thinning level is sufficient ($E_{th} \sim 10^{-4} E_0$)

The smaller E_{th},

the better the shower is modelled

& the larger the computing time.

particles far from shower core, e.g. S(r) in Auger:

small particle density requires good thinning ($E_{th} < 10^{-7} E_0$)

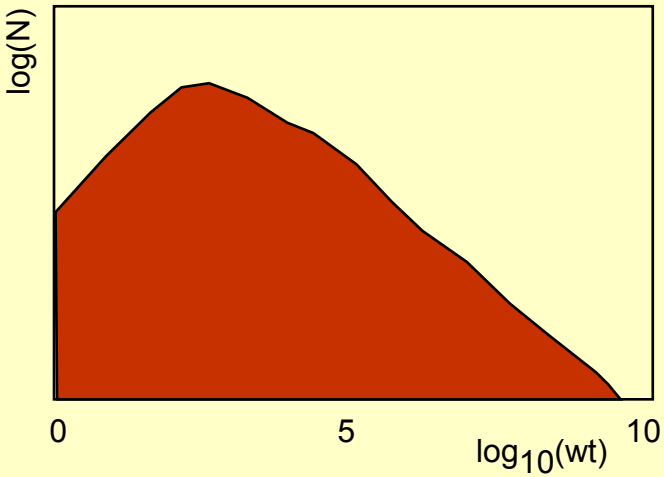
Artificial fluctuations due to thinning should be smaller than the intrinsic shower fluctuations

(to be checked for the variable in question...)

Computing time and disk space are reduced, but do still grow proportional to E_0.

Particle weights can go up to $w_{max} = E_{th} / E_{min} \sim 10^{20} \times 10^{-6} / 10^5 = 10^9$

start end of thinning (E_{min} = low energy cut-off)



High weights are problematic:

Is there a way to avoid them ?

Weight Limitation and Optimum Thinning

Avoid too high weights by setting a maximum allowed weight w_{max} (e.g. 10^5).
If weight gets larger, follow all particles again.

Of course, weight limitation increases run time again.

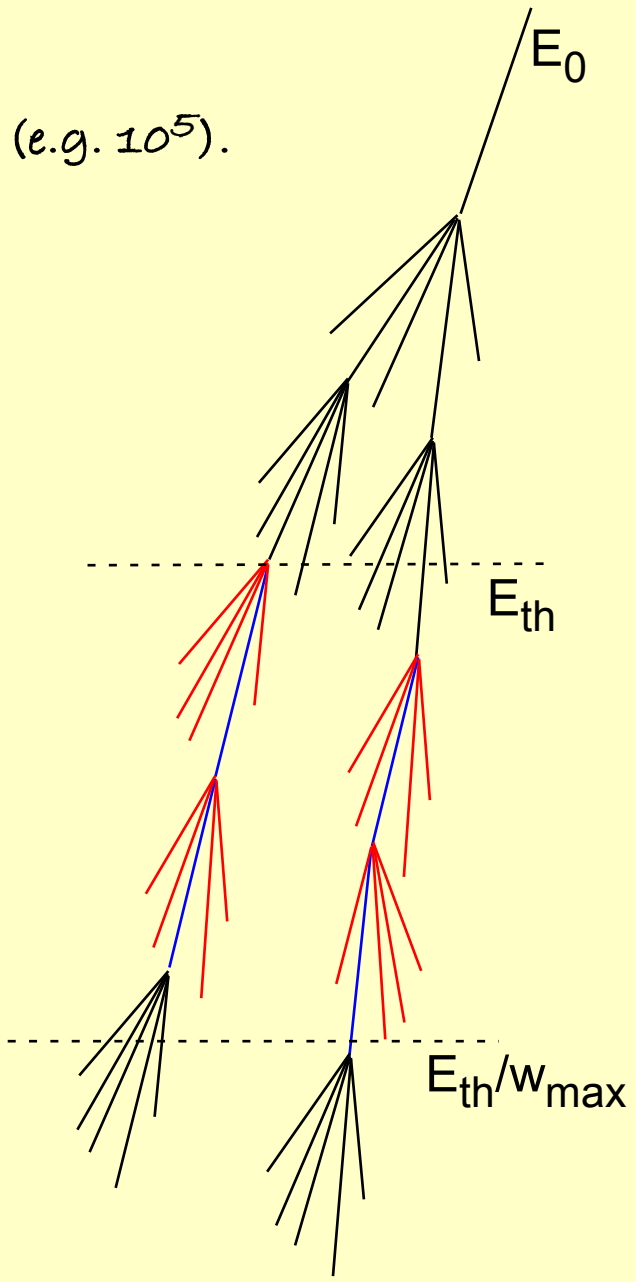
Which is the best setting of E_{th} and w_{max} ?

i.e. which minimizes the statistical error for a given run time?

Optimum thinning for a given ε_{th}

$$w_{max} = E_0 \text{ (in GeV)} \cdot \varepsilon_{th}$$

10^{18} eV	$\varepsilon_{th} = 10^{-6}$	$w_{max} = 10^3$
10^{19} eV	$\varepsilon_{th} = 10^{-6}$	$w_{max} = 10^4$
10^{20} eV	$\varepsilon_{th} = 10^{-6}$	$w_{max} = 10^5$



The run time is only dependent on ε_{th}, no longer on energy.

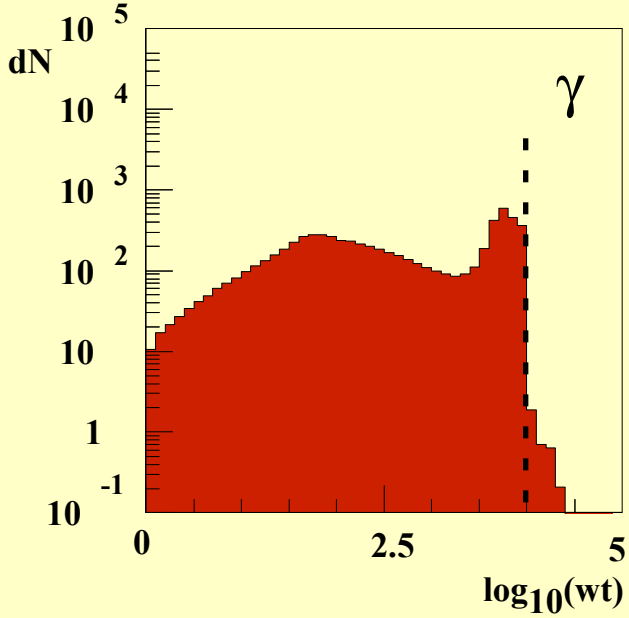
10^{-5} optimum thinning is about as good as 10^{-7} thinning without weight limitation.

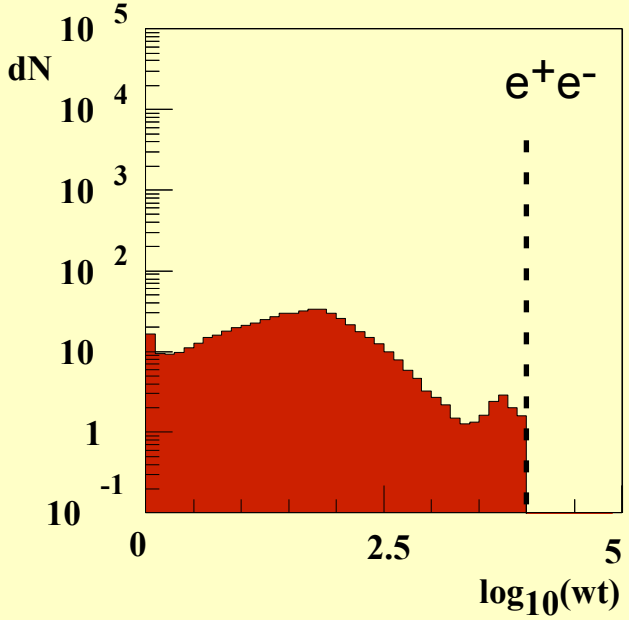
Since γ and e^+e^- are much more abundant (x100) than $\mu^+\mu^-$ or hadrons
we set different ε_{th} and w_{max} for γ & e^+e^- than for $\mu^+\mu^-$ or hadrons.

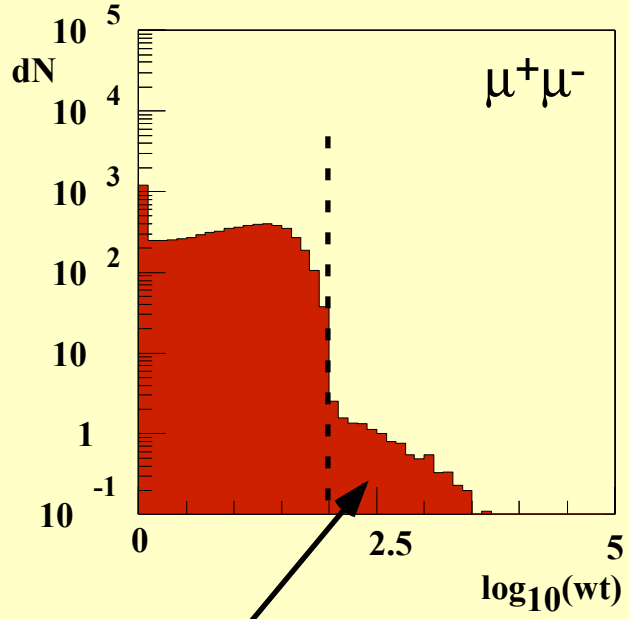
e.g.: proton 10^{19} eV 37^0

γ, e^+e^-: $\varepsilon_{th} = 10^{-6}$ $w_{max} = 10^4$

$\mu^+\mu^-$, h: $\varepsilon_{th} = 10^{-8}$ $w_{max} = 10^2$







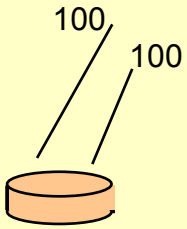
1 particles with weight 10^5 stands for
10 particles with weight 10^4 or
10^5 particles with weight 1

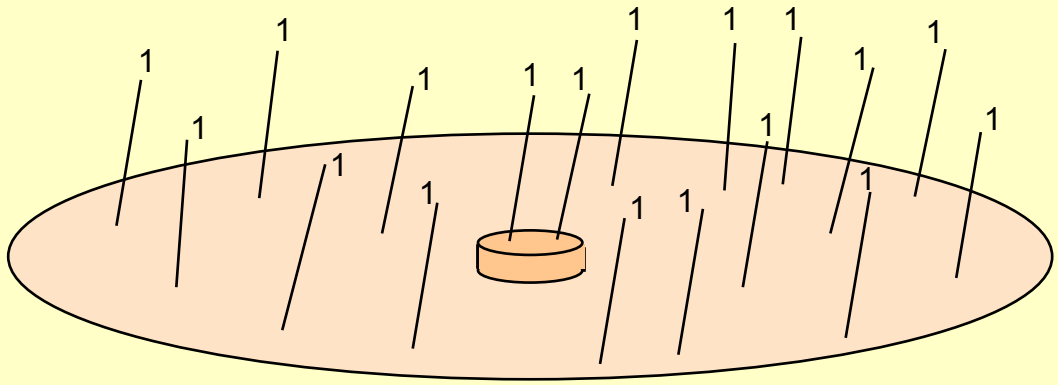
muons produced
in photo production
from photons with weight $>10^2$

How to treat a particle with a high weight once it hits a detector ???

A particle with weight stands for many otherswith different
masses, energies, angles, impact points.

Some sort of un-thinning is needed (e.g. method used in Auger)





Instead of N particles with high weights wt on a small area A,

use N·wt particles, each with weight 1, from a larger area A·wt.

Problem: For wt > 10^5 the area becomes so large (~ distance between detectors)
that particle densities, energies, ... will change from one side to the other.

There is no ideal method to get rid of the weights again.
Some of the information has been lost in the thinning process.

Other speed-up tricks?

All tricks come at a cost !
Speed-up mechanisms cause biases and distortions.
Careful checks are needed that they don't invaluate the results.

Example: early KASCADE simulations:

> Low-energy electromagnetic subshowers created high in the atmosphere were discarded (x3 speed-up, with bias of less than 3%)
>
> When higher energies were simulated,
> this speed-up caused a large bias which led to wrong results.
> Massive simulations had to be repeated after the problem was found.

Use parallel processing?

In principle possible: do first interaction & distribute secondaries on different processors

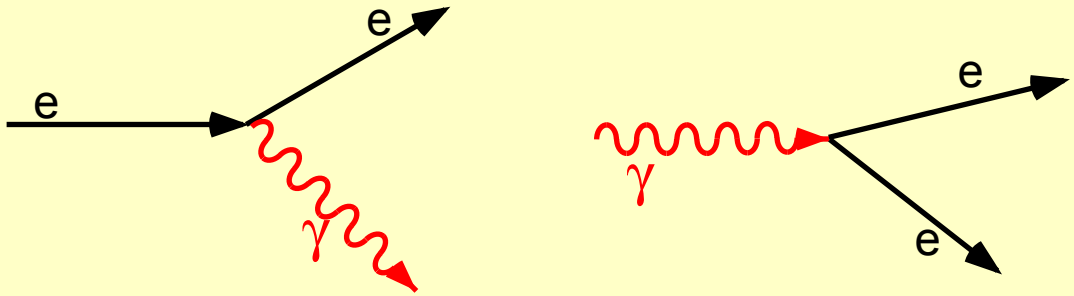
But we usually need more than one shower to evaluate fluctuations.

Best way to parallelize: run on each processor one shower
to avoid any inter-processor communication.

e.g. processor farm in Lyon (~1000 processors + lots of storage space)
70,000 showers for Auger, ~150 processor years, 20 TB

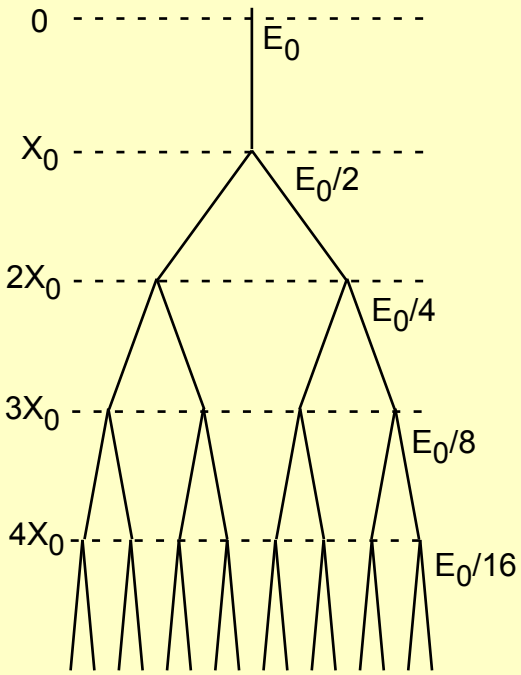
Electromagnetic Showers: from Toy Model to EGS4

basic reactions: photons: pair production
electrons: bremsstrahlung



Both reactions have the same scale length (X_0) and have two outgoing particles per incoming particle.

Toy Model (one-dimensional, very simplified, yet qualitatively correct):



particle multiplication (x2) in each step (X_0) until $E < E_{crit}$, then particle losses due to ionisation dominant.

$t = k X_0$, $\quad k = 1,2, \ldots$

$N = 2^k \qquad E = E_0/N$

k_{max}: $\quad E_0 / 2^{k_{max}} = E_{crit} \qquad k_{max} = \ln(E_0/E_{crit}) / \ln(2)$

grows only logarithmically with E_0

$t_{max} = k_{max} \cdot X_0$

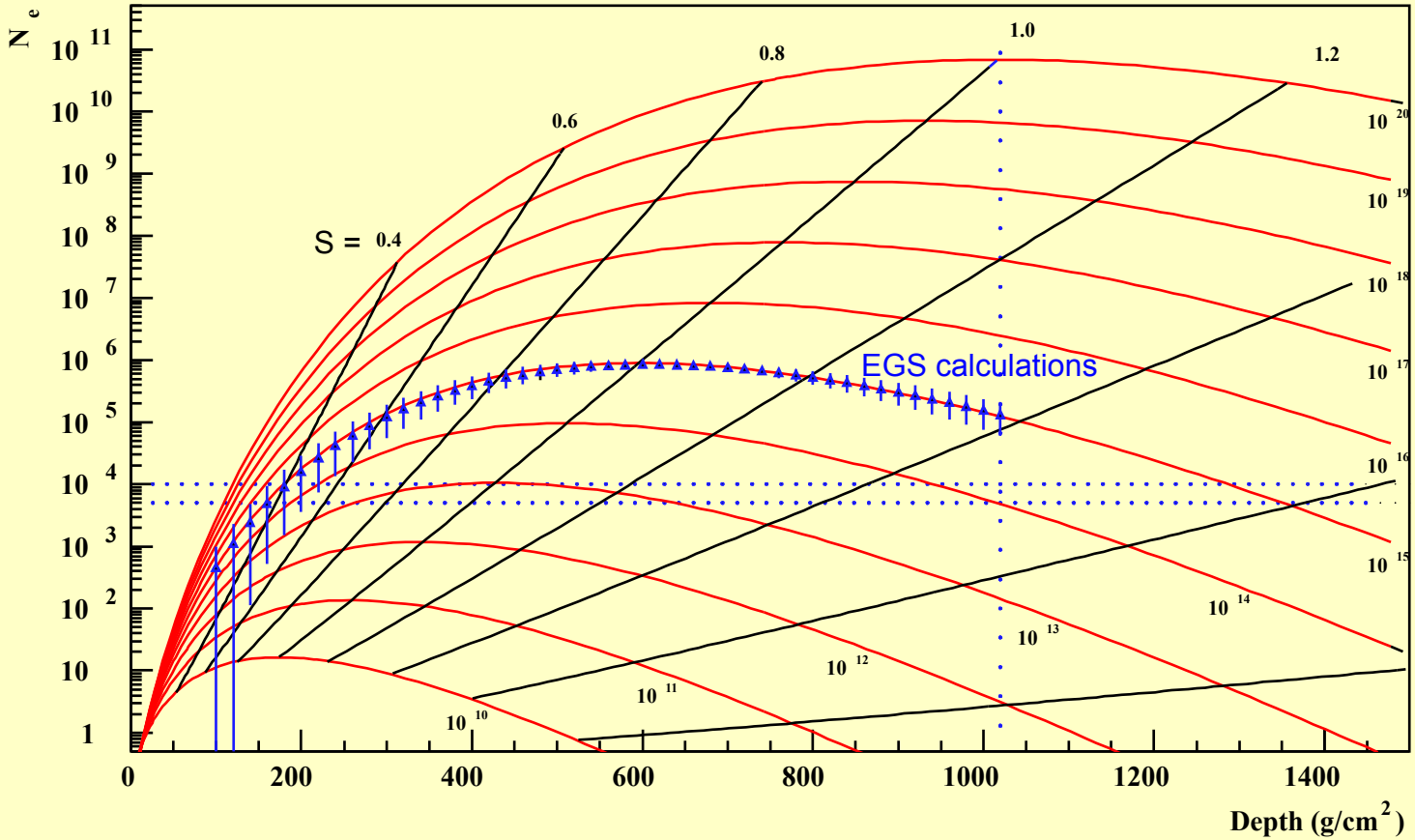
$N_{max} = E_0/E_{crit}$

Measure t_{max} or N_{max} and estimate E_0.

Nishimura Kamata Greisen (NKG): Longitudinal Shower Development

analytic description of purely electromagnetic showers:

$$N_e = \frac{0.31 \ \exp(\, t\,(1-1.5 \ln s))}{\sqrt{\ln(E_0/E_{crit})}} \qquad s = \frac{3\,t}{t + 2\ln(E_0/E_{crit})}$$



N_e :
number of electrons down to energy 0 ?
(unphysical)

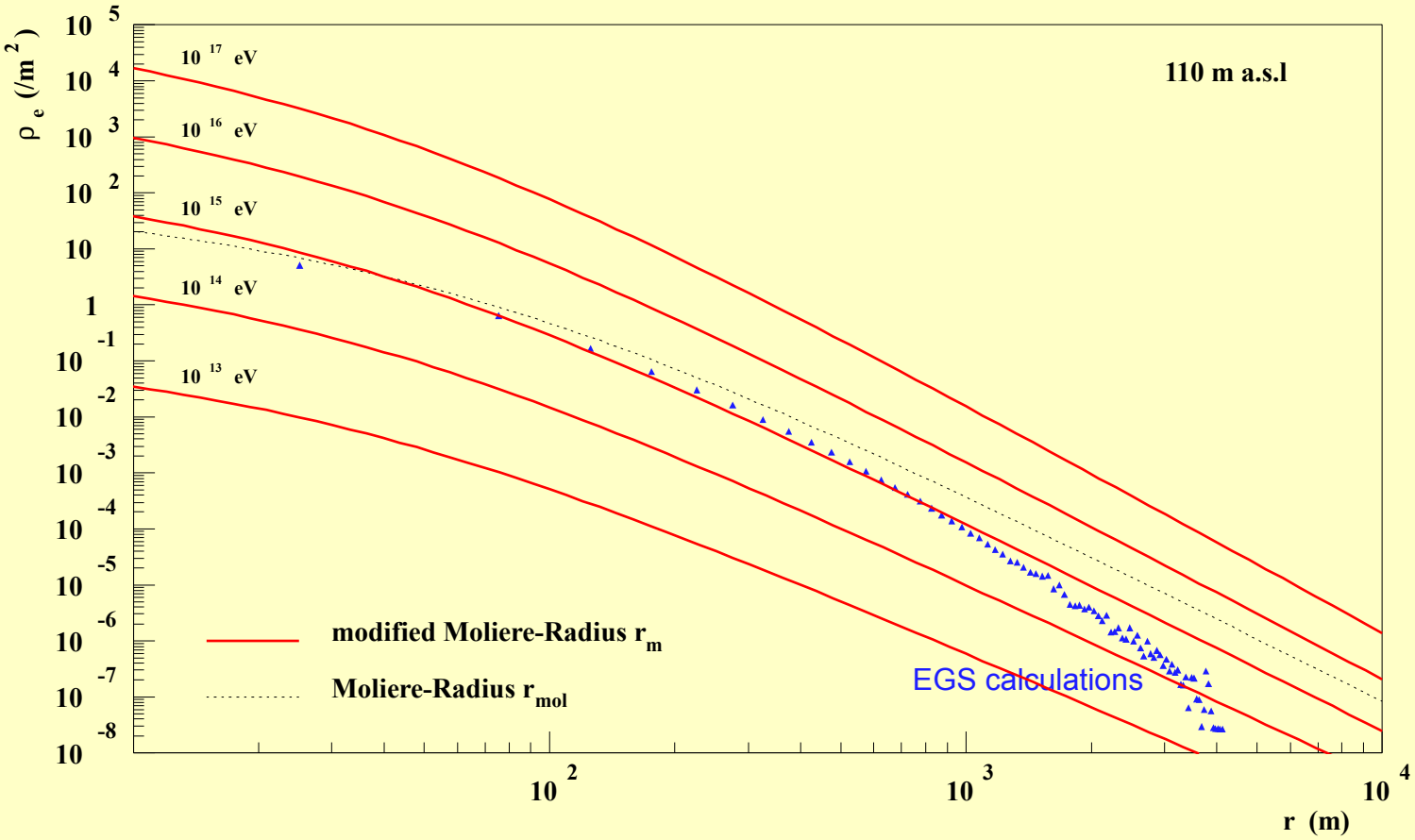
Just average,
no fluctuations.

Nishimura Kamata Greisen: lateral shower development

purely electromagnetic showers:

$$\rho_e = \frac{Ne}{2\pi r_m^2} \frac{\Gamma(4.5-s)}{\Gamma(s)\,\Gamma(4.5-2s)} \left(\frac{r}{r_m}\right)^{s-2} \left(1+\frac{r}{r_m}\right)^{s-4.5}$$

$$r_m = (0.78 - 0.21\ s)\ r_{mol}$$



$r_{mol} = X_0\, E_s/E_{crit}$

$\sim$ 9.6 g/cm^2

$\sim$ 78 m at sea level

$E_s = m_e c^2\,(4p/a)^{1/2} \sim 21$ MeV

A cylinder around the shower axis with radius r_{mol} contains 90% of the shower energy.

NKG formalism allows a fast, semi-analytical simulation of electromagnetic sub-showers.

Limitations of NKG:

N_e : number of electrons down to energy 0 is unphysical.

When realistic detector thresholds are used up to 30% difference.

Also purely electromagnetic showers contain some muonic / hadronic component.

$\gamma \longrightarrow p$ $\gamma \longrightarrow \mu^+\mu^-$ (suppressed, but nevertheless there)

no fluctuations

no info on gammas

no info on particle times and directions

Full 4-dim simulation with EGS

Electron Gamma Shower Code Nelson et al. ~1970

ALL processes of electrons and gammas are included.

bremsstrahlung, ionisation, d-electrons, Bhabha & Moeller scattering, multiple scattering, annihilation, ...

e^+e^- pair production, Compton effect, photo effect, Rayleigh scattering, ...

based on QED calculations and is very well checked and verified.

extended by LPM effect (> TeV in dense materials; > 10^{18} eV in atmosphere)

EGS gives precise predictions of all sorts of electromagnetic interactions in materials.

negative: 40 x slower than NKG version,
but the quality of the results is worth the while.

Random Numbers

... are of central importance in Monte Carlo methods.

Use random physical processes to create real random numbers?
(e.g. radio active decay, electronic noise, ...)

In principle yes, but ....

MC programs must be reproduceable (e.g. for bug fixing).

Pseudo-Random numbers:

they are produced by a predictable algorithm
but behave in all respects like random numbers.

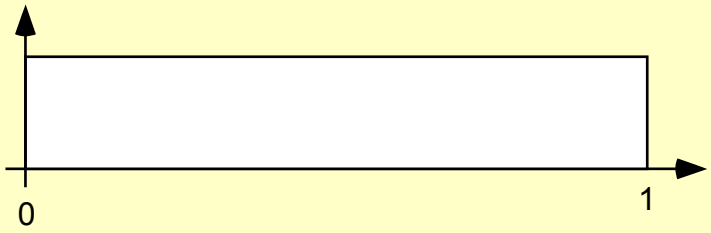
(all digits, all combinations of digits appear with equal probability,
there are no correlations within the sequence)

Computers are deterministic machines,

i.e. computer generated random number sequences are not really random
and Computer sequences have a finite sequence length (period)

It is an art to produce good random numbers !

Uniform Random Number Generator



uniform probability in range 0, ... 1.

Simple example: linear congruent generator

$l_j = a\ (l_{j-1} + c) \bmod m$ $\quad u_j = l_j / m$

3 parameters: a, c, m

l_0 : seed

$l_1, l_2, \ldots$ are integers between 0 and m-1

u_j is a real number between 0 and 1

The maximum period is m, but real period depends on a and c. typical: $10^5 \ldots 10^{11}$

not enough for serious applications.

Random numbers from this generator are not uncorrelated:

k-tupels of random numbers lie in k-dim space on (k-1)-dim hyper planes

Less significant bits are usually less random

better: Fibonacci generator

$u_n = (u_{n-24} + u_{n-55}) \bmod 1$ $\quad n > 55$ $\quad$ period : $> 2^{55} \sim 10^{18}$

to be initialised with $u_1 \ldots u_{55}$ $\quad$ still not sufficient.

modern generators combine simpler methods:

1) Combine two random numbers (from different generators) with "+", "-", or "exclusive OR".

2) A sequence of random numbers from generator 1 is stored in a memory.
A random number of generator 2 is used as address of the next random number in the memory.

Used in CORSIKA: RANMAR (CERNLIB)

32-bit floating point numbers between 0 and 1.

900.000.000 different sequences of $\sim 2^{144} = 10^{43}$ period length

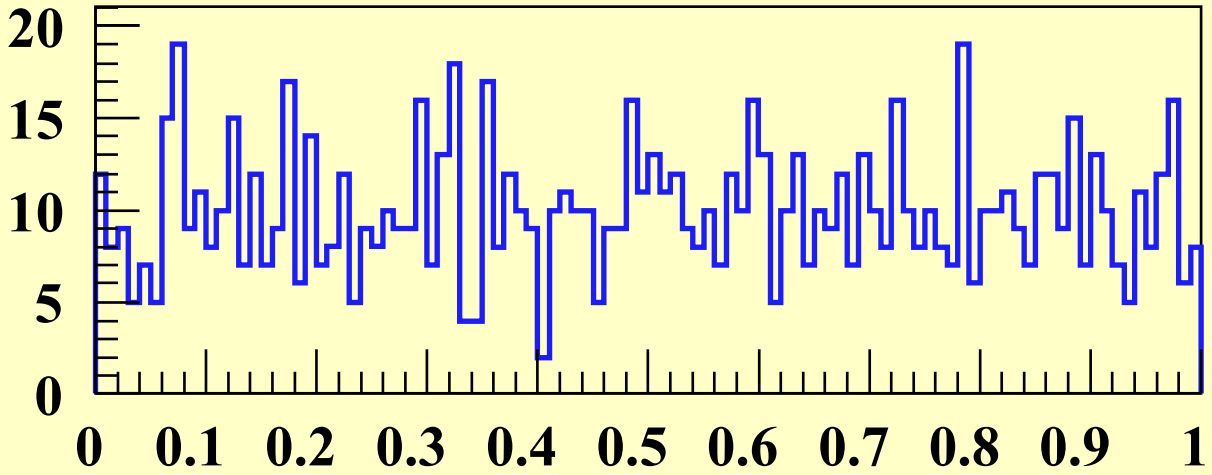
There are even better ones:

... but the better the random number generator, the slower it is.

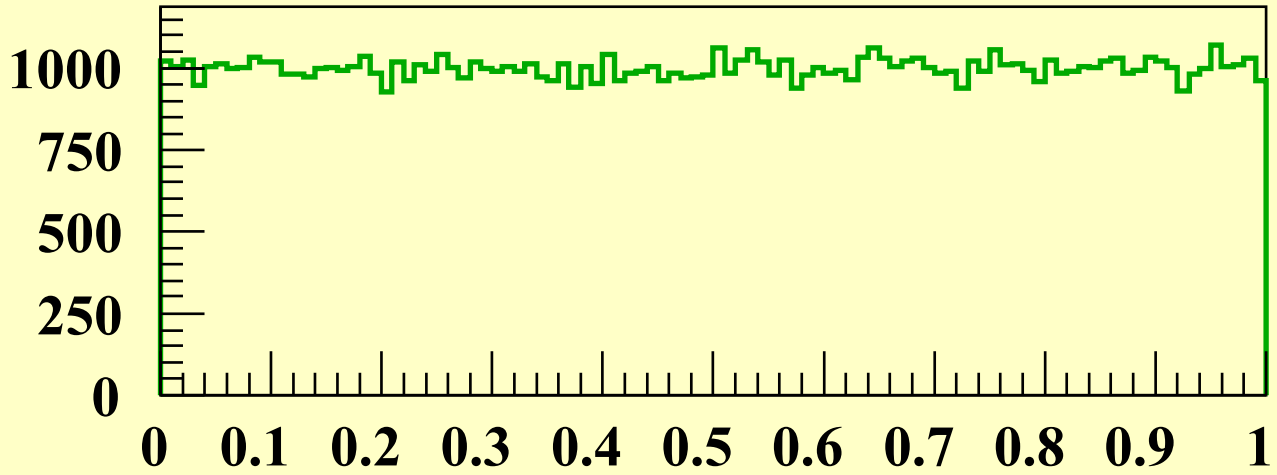
About 30% of the computing time of CORSIKA goes into calculation of random numbers !

Uniform

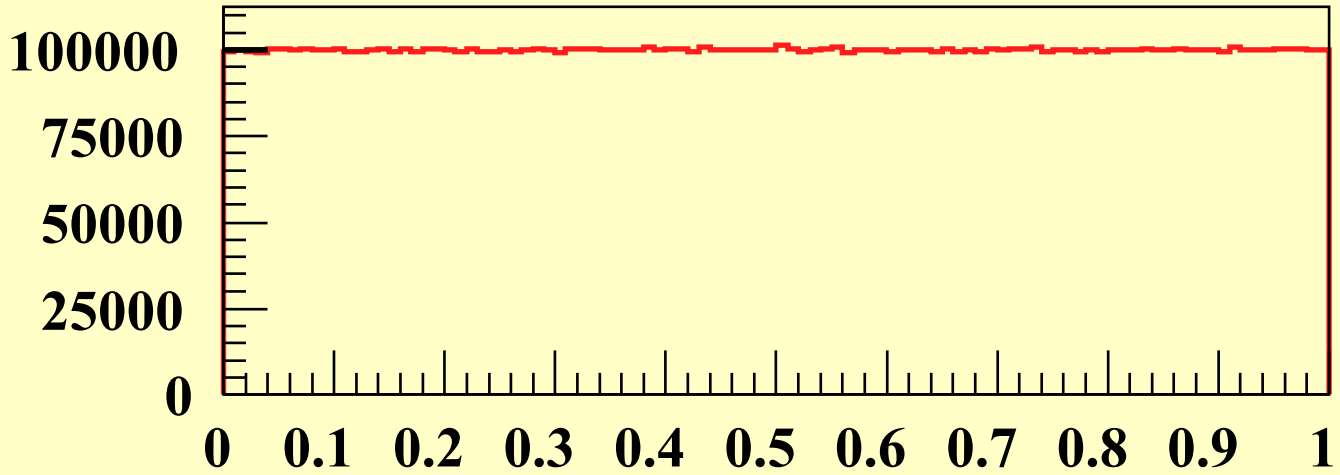
10^3 events

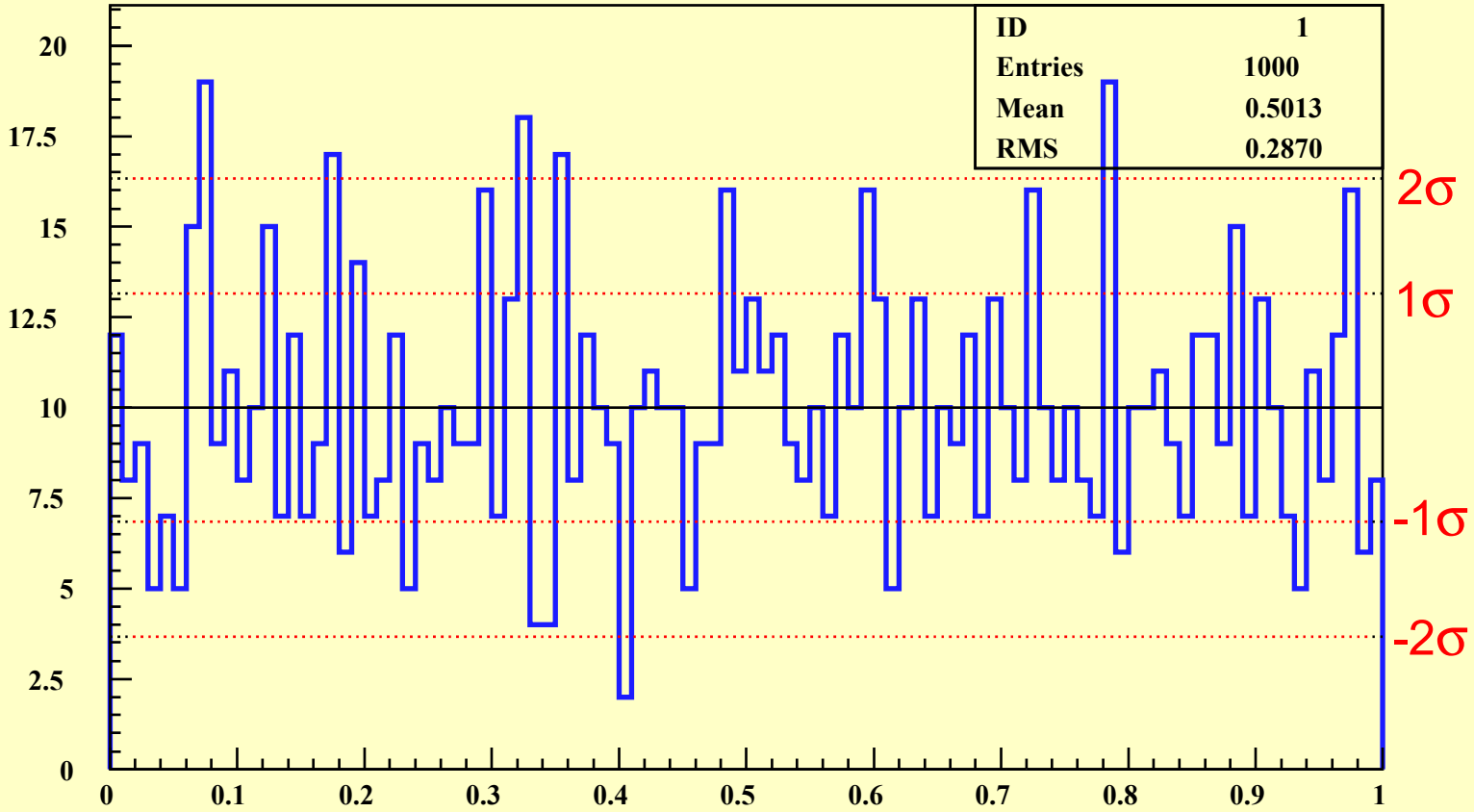


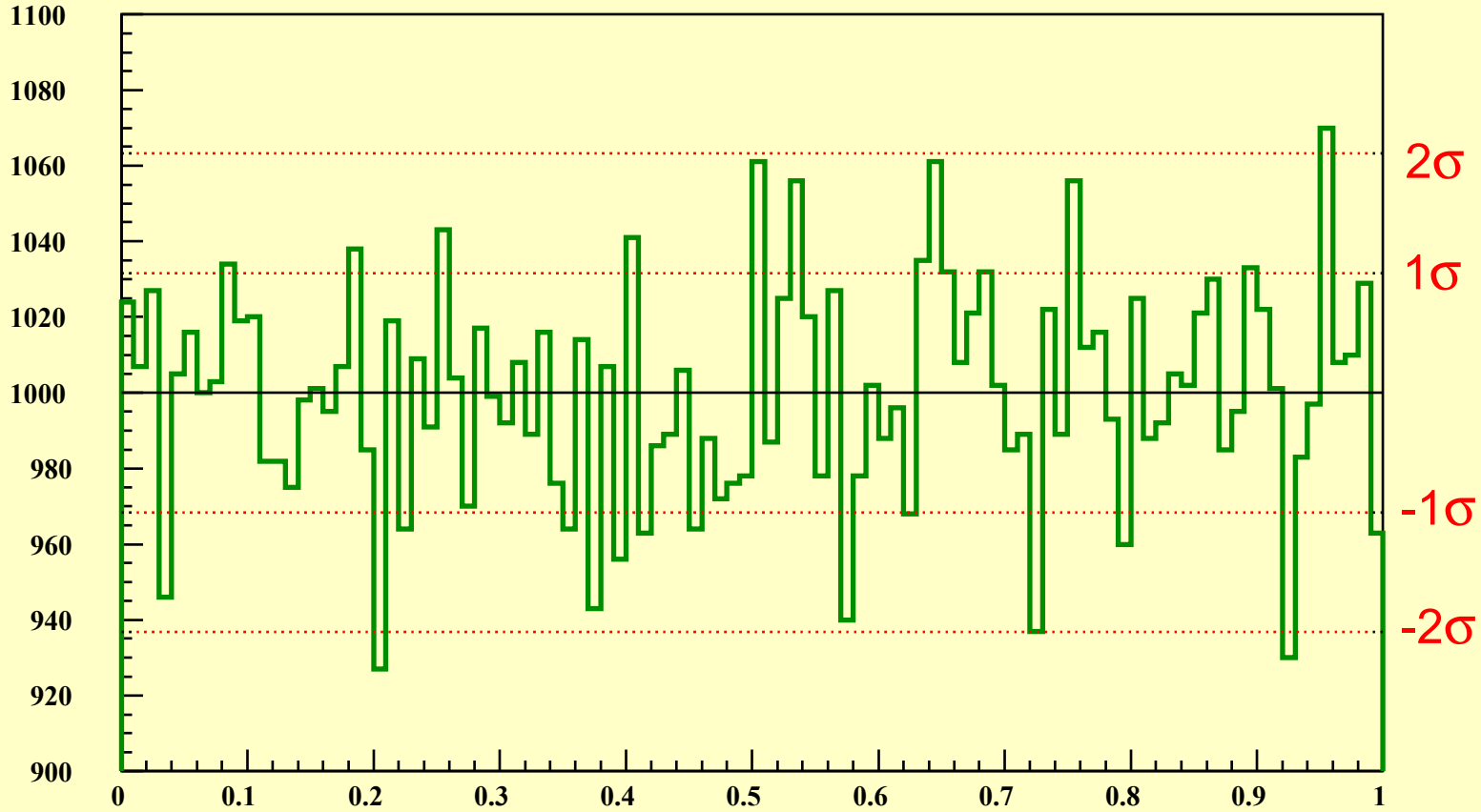
10^5 events



10^7 events




ID 1
Entries 1000
Mean 0.5013
RMS 0.2870
2σ
1σ
-1σ
-2σ
20
17.5
15
12.5
10
7.5
5
2.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

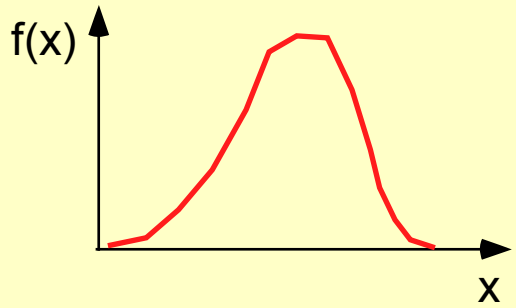

1100
1080
1060
1040
1020
1000
980
960
940
920
900
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
2σ
1σ
-1σ
-2σ

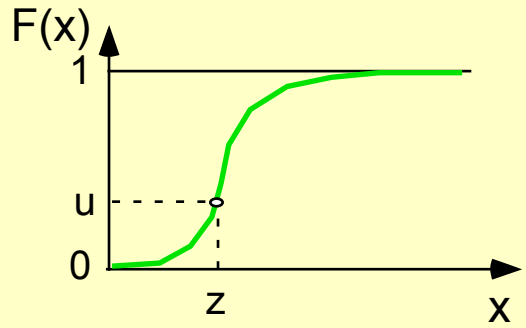
Generators for Arbitrary Distributions

create random numbers according to the probability distribution f(x)

Method 1: inverse integral method

$$F(x) = \int_{-\infty}^{x} f(x)\,dx$$





by construction F(x) ranges between 0 and 1, if f(x) is a probability density.

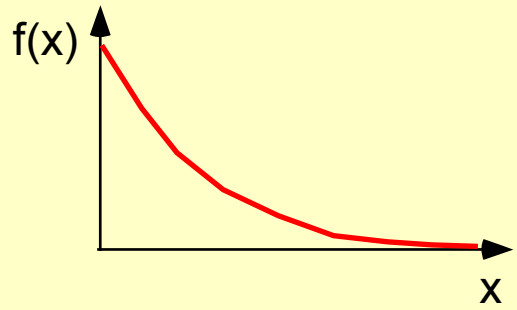
If we draw $u = F(x)$ at random from a uniform distribution and find $z = F^{-1}(u)$ then z is distributed like f(x).

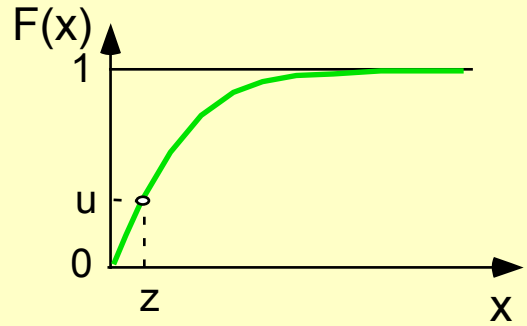
e.g. exponential distribution: $f(x) = \lambda \exp(-\lambda x)$

$$u = F(x) = 1 - \exp(-\lambda x)$$

$$z = -\ln(1-u) / \lambda \quad \text{or}$$

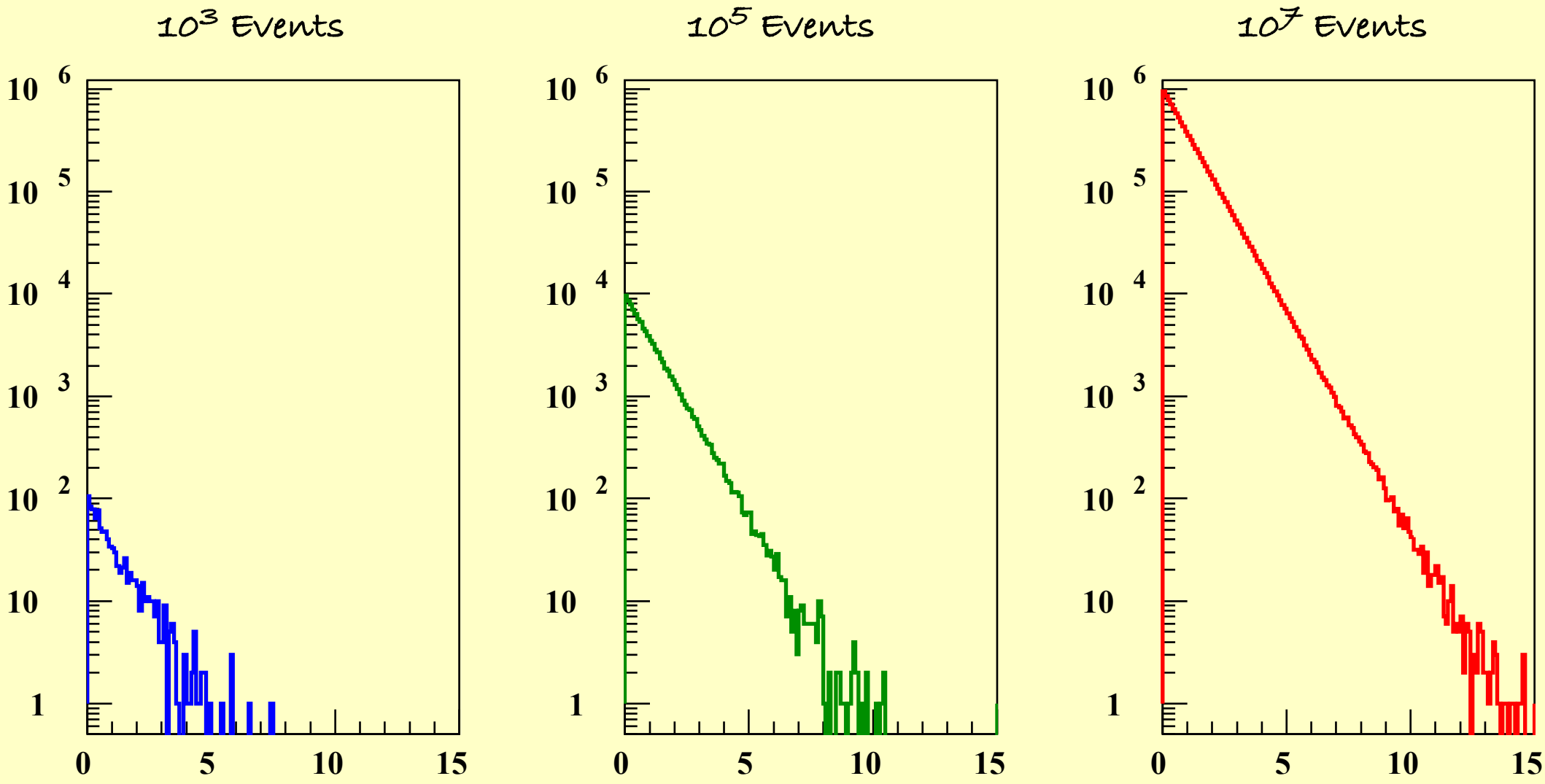
$$z = -\ln(u) / \lambda$$





Only one call to uniform generator gives u
then z is exponentially distributed.

Exponential



What if we cannot compute $F^{-1}(x)$?

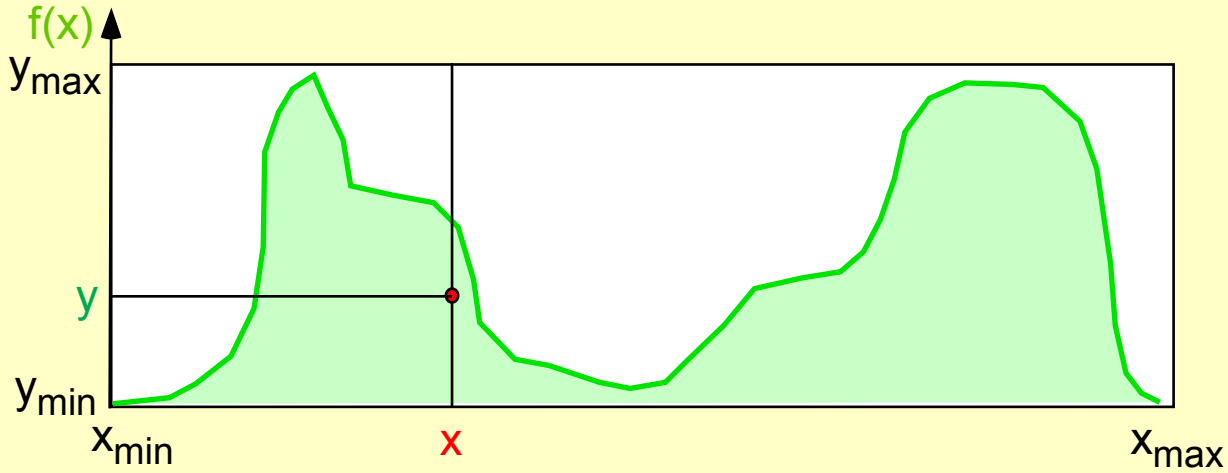
Method 2: brute-force method

draw x at random from a uniform distribution between (x_{min}, x_{max})

draw y at random from a uniform distribution between (y_{min}, y_{max})

if y < f(x) accept x as random number

if y > f(x) reject x and start over again



x will be distributed like f(x), but at least 2 calls for uniformly distributed random numbers are necessary.

The rejection makes this method even more inefficient.

The larger the white area the more rejections will happen.

Gaussian Distribution

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

Gaussian with mean μ and standard deviation σ

... is very important: if many independent little variations contribute to one variable
then it is following a Gaussian distribution

i.e. $z = \sum_{j=1,12} u_j - 6$ will produce a distribution of z that is about Gaussian,
but needs 12 calls for uniform random numbers.

A better way:

1) create 2 uniformly distributed random numbers u_1 and u_2
2) $v_1 = 2\,u_1 - 1$ uniformly distributed in (-1,1)
   $v_2 = 2\,u_2 - 1$
3) $r^2 = v_1^2 + v_2^2$

if $r > 1$ then goto 1)
if $r < 1$ continue

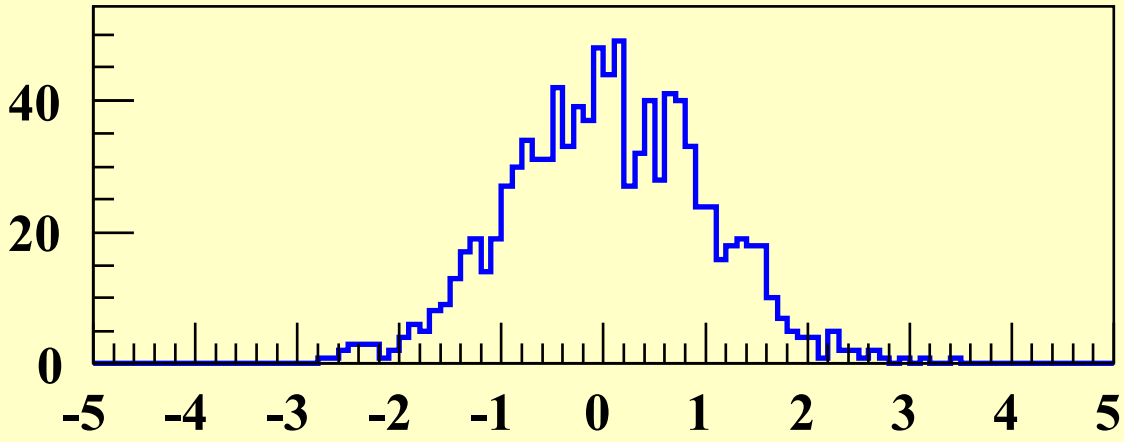
4) $z_{1/2} = v_{1/2} \sqrt{\frac{-2 \ln r^2}{r^2}}$ are two independent and Gaussian distributed random numbers
with mean 0 and standard deviation 1

To get a Gaussian with mean m and standard deviation s

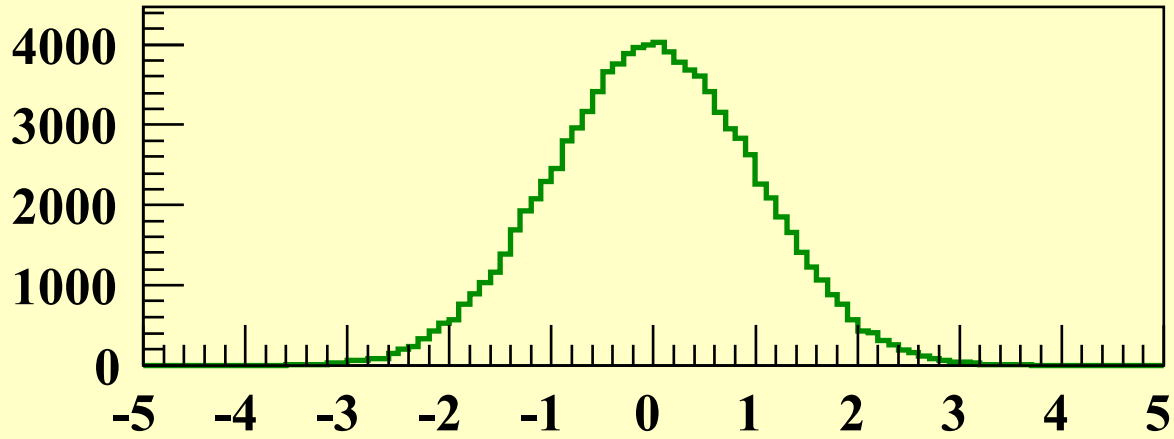
compute $z_{1/2} = z_{1/2} \cdot s + m$

Gaussian

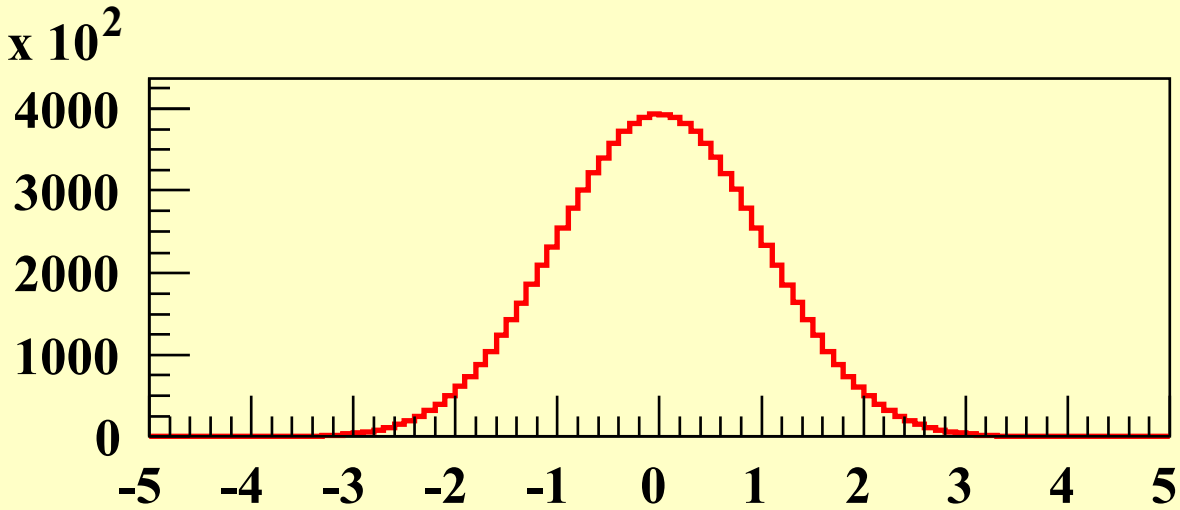
10^3 events

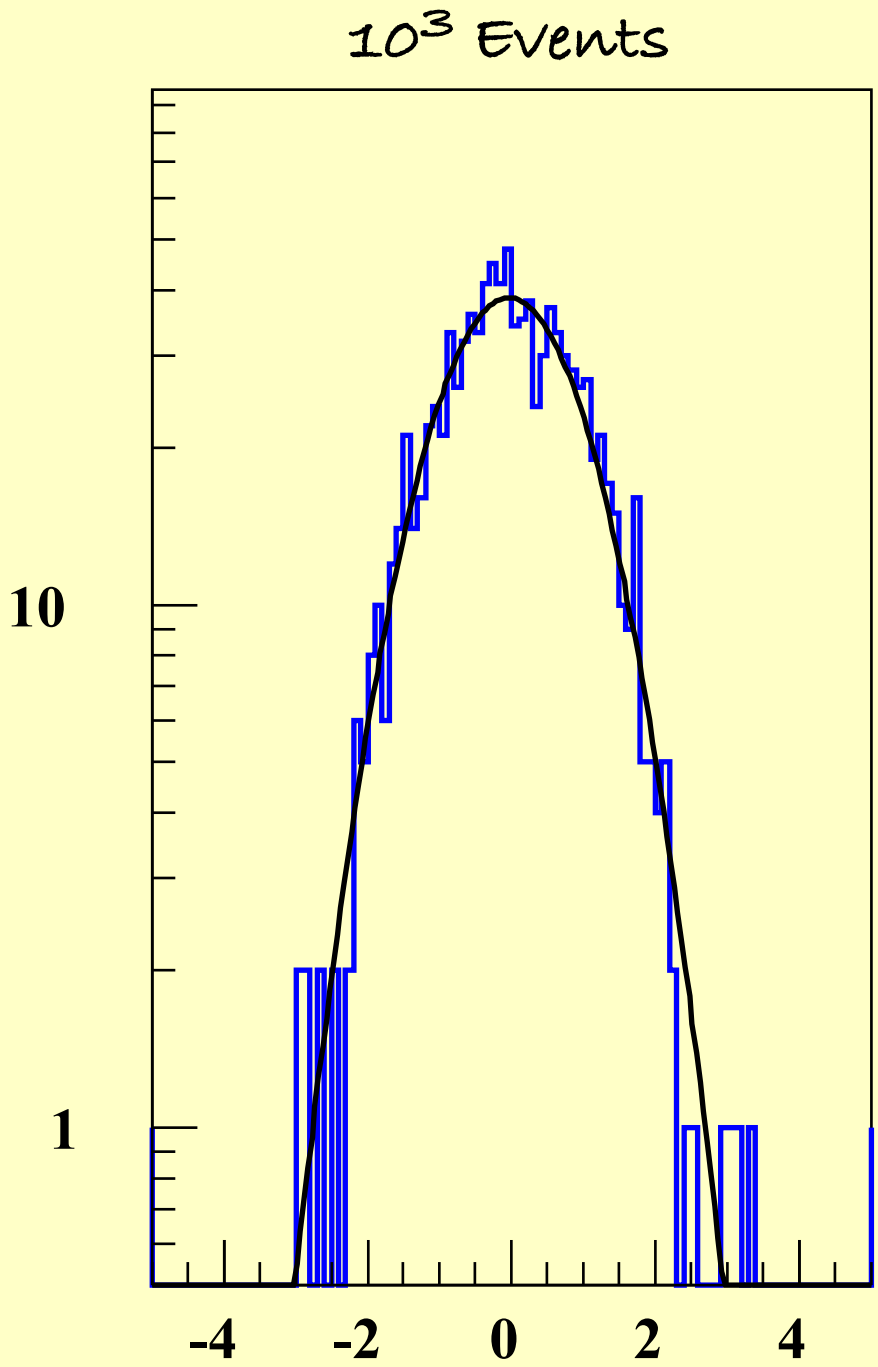


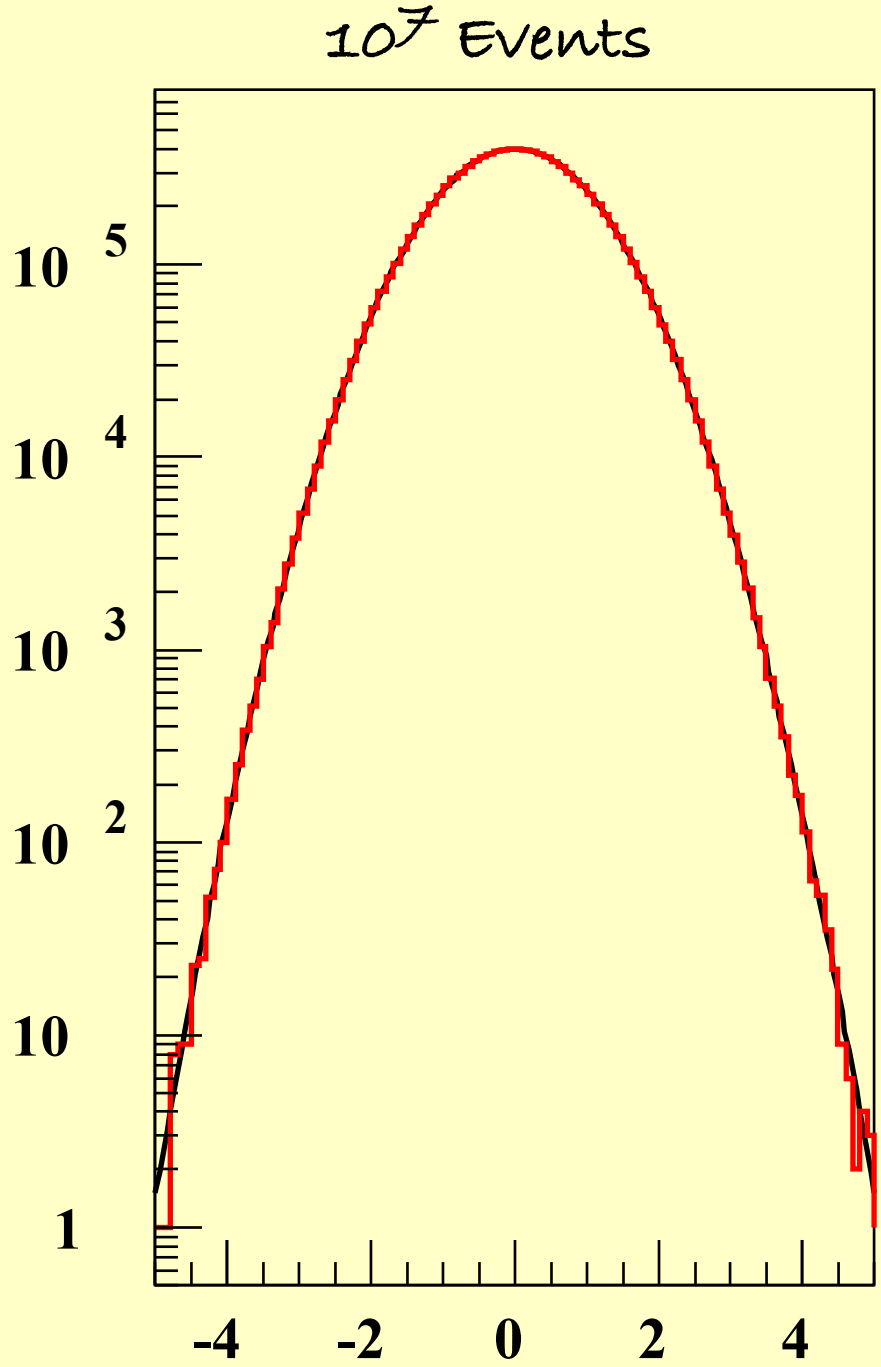
10^5 events



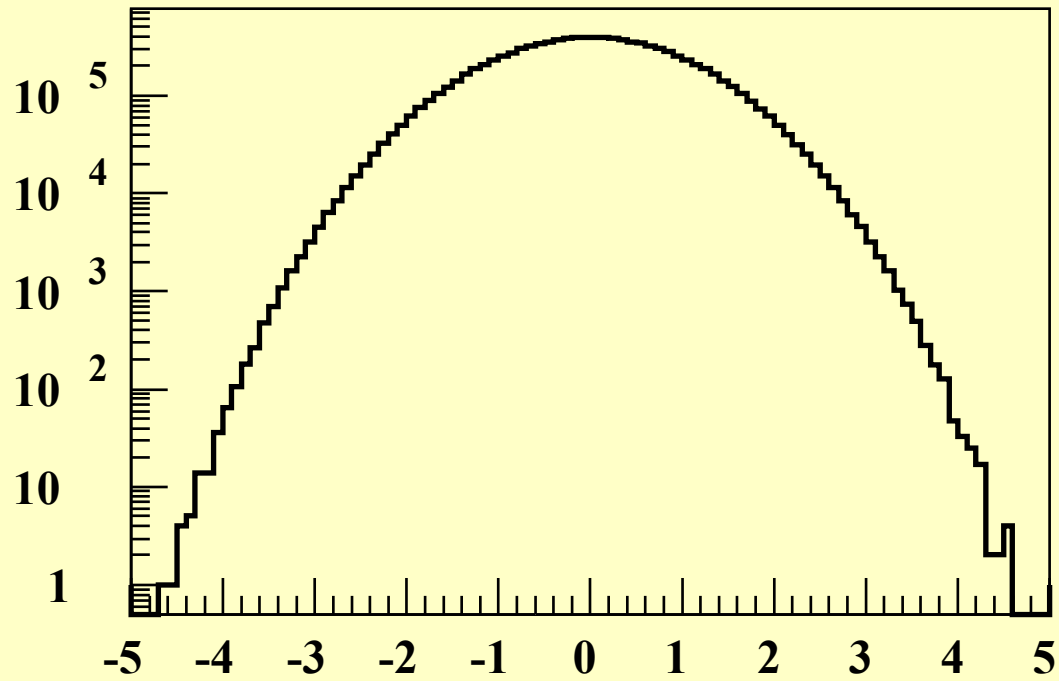
10^7 events



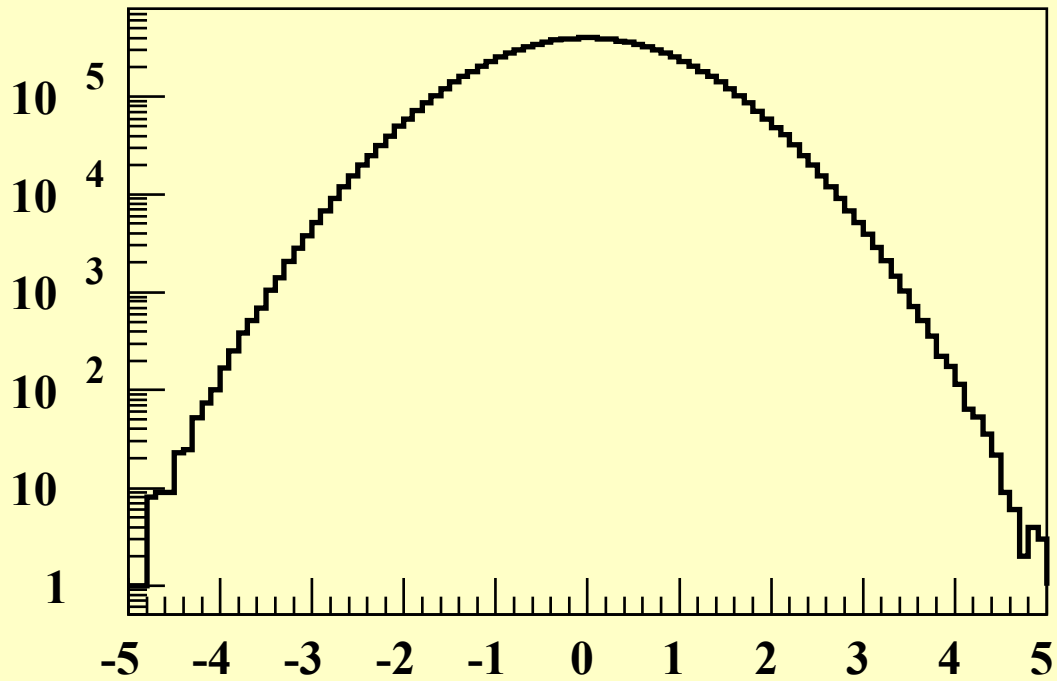

10^3 Events
10
1
-4
-2
0
2
4


10^7 Events
10^5
10^4
10^3
10^2
10
1
-4
-2
0
2
4

Simple Gaussian



Advanced Gaussian

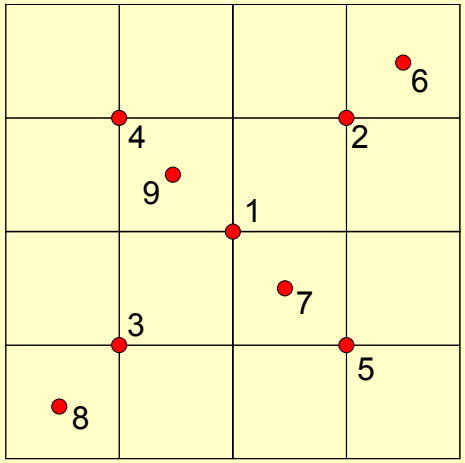


Quasi-Random Number Generators

Sometimes one wants to map out a value range faster and more uniform than with pseudo-random numbers.

Quasi-Random numbers are constructed to fill a given space as uniformly as possible.
(i.e. to avoid clusters)

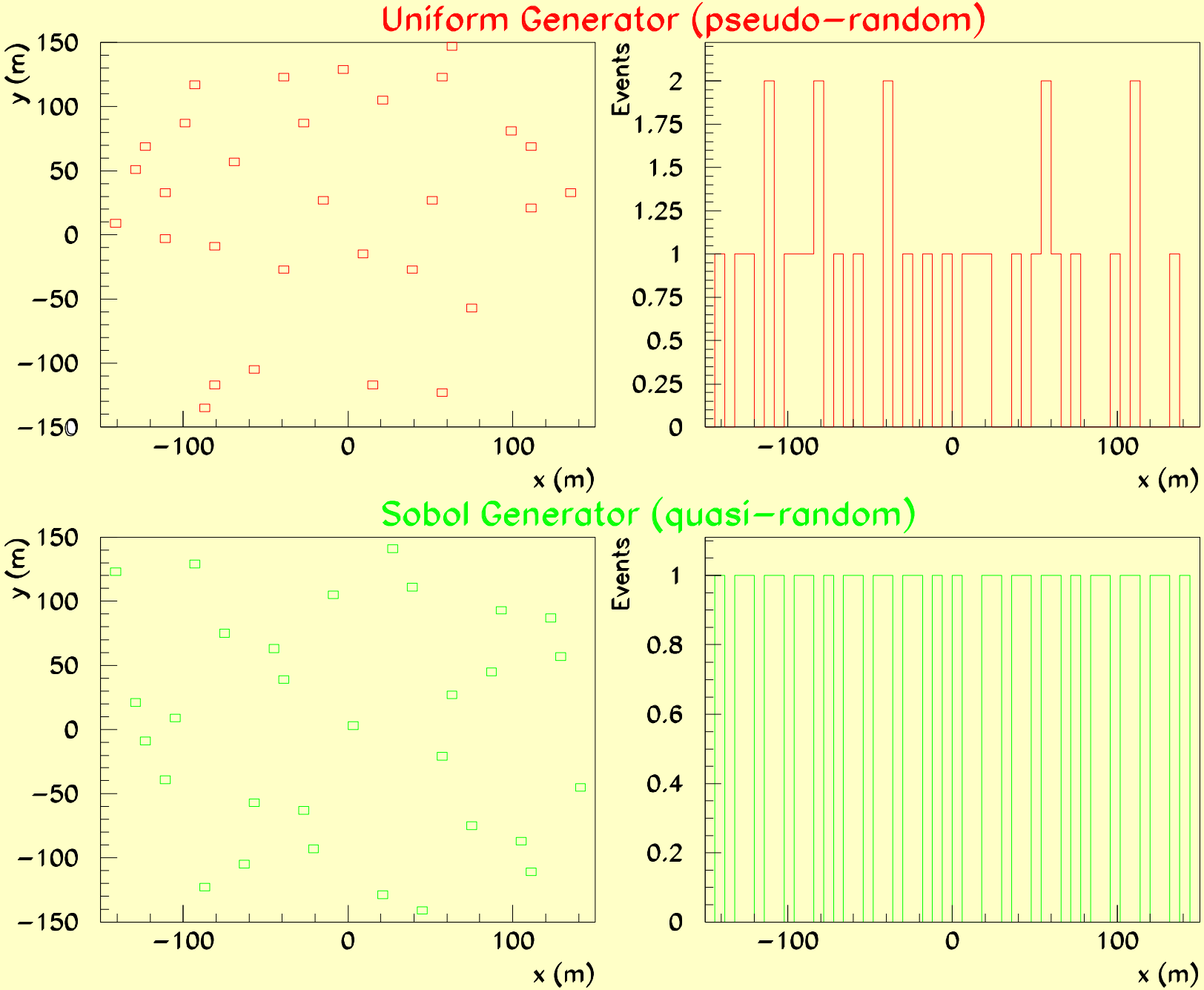
e.g. Sobol Generator:



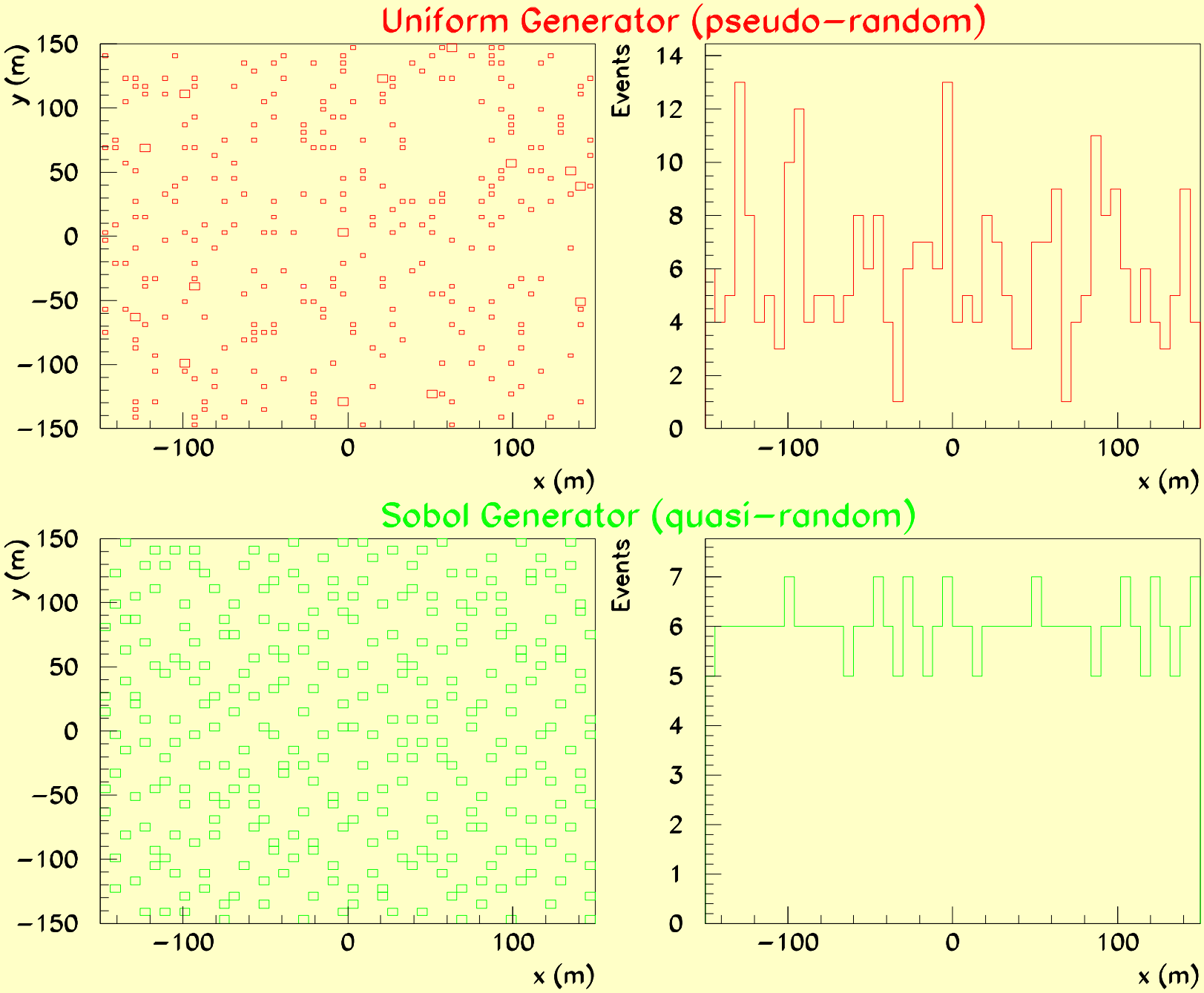
Exhibits reduced fluctuations as comapred to pseudo-random generators.

Therefore usually not used often in Monte Carlo.

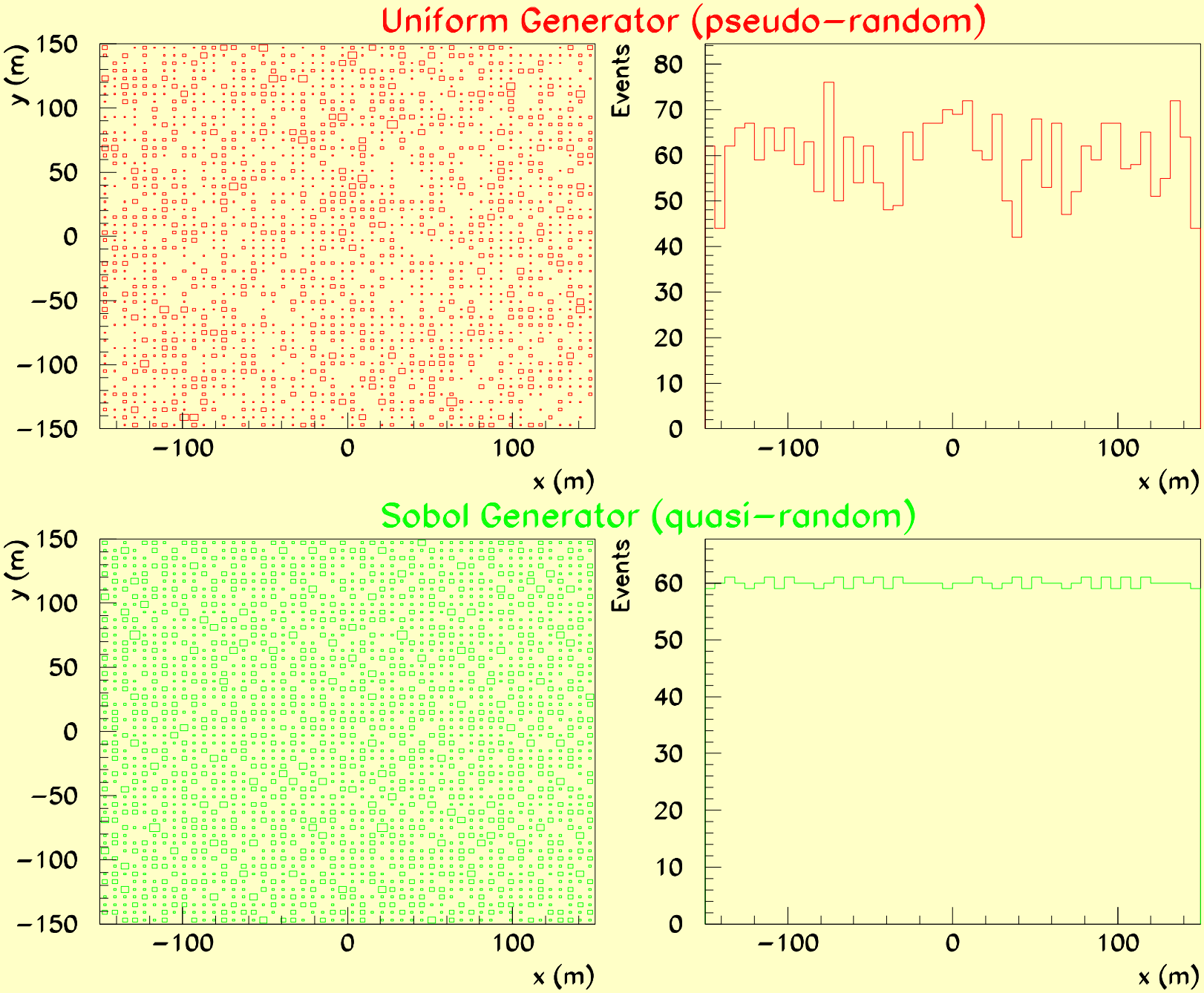
Sobol Quasi-Random Generator vs Pseudo-Random Generator: 30 evts



Sobol Quasi-Random Generator vs Pseudo-Random Generator: 300 evts



Sobol Quasi-Random Generator vs Pseudo-Random Generator: 3000 evts



Monte Carlo Pitfalls

beware of bad random number generators

not random, too short sequence, correlations

rounding errors

e.g. emission angles at high energies

steeply falling distributions

Summary:

Monte Carlo Simulations provide a powerful tool
for many parts of science and engineering,
especially when statistical processes are involved.

MC Methods are prone to subtle errors,
due to random numbers and finite numerical precision.
Therefore, they need careful testing.

Nevertheless, MCs are undispensible for
Air Shower analysis and experiment design.