Experiment Simulation

CERN School of Computing 2005
Saint Malo

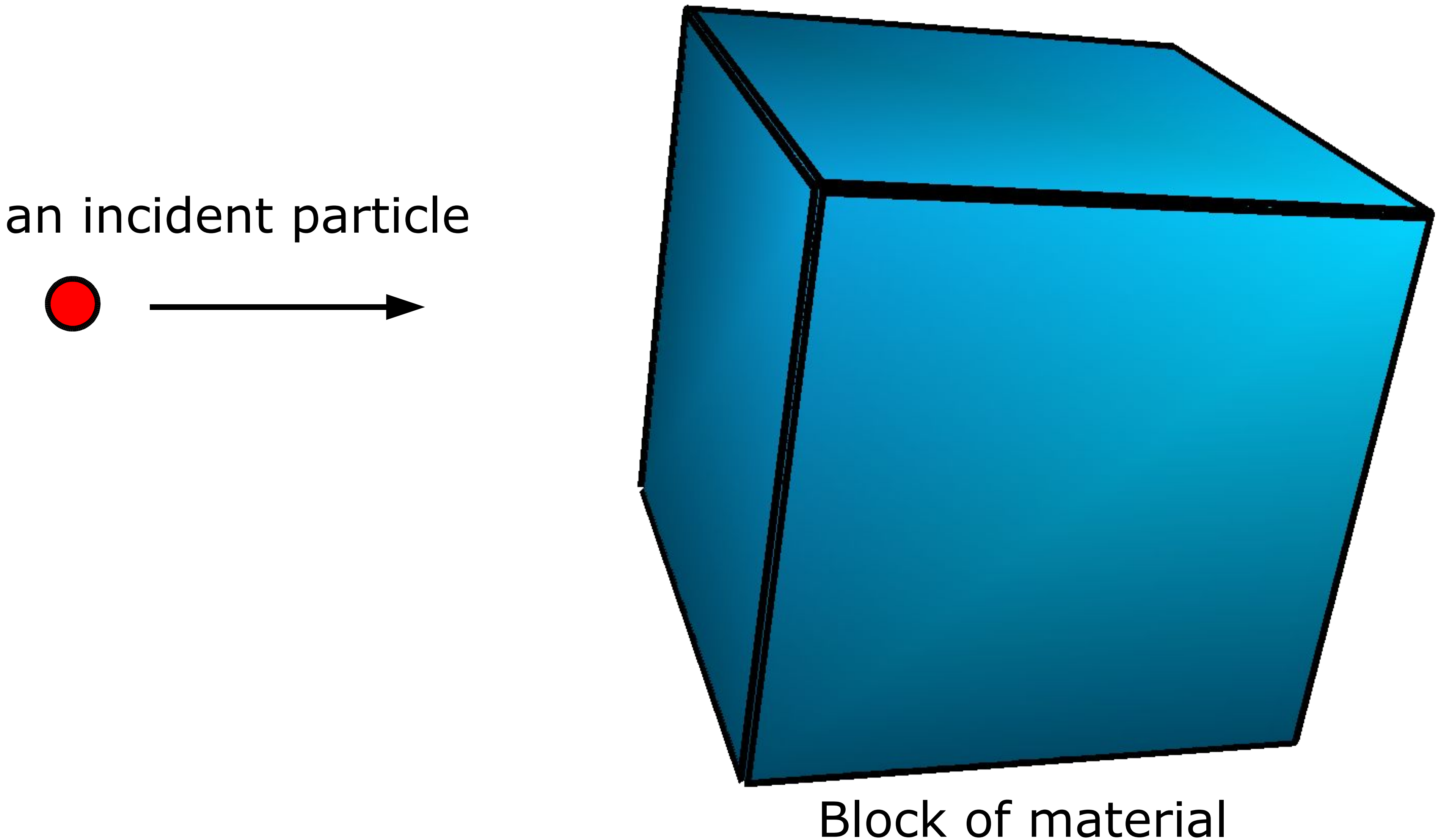
Lecture 2

Martin Liendl
SWX Swiss Exchange

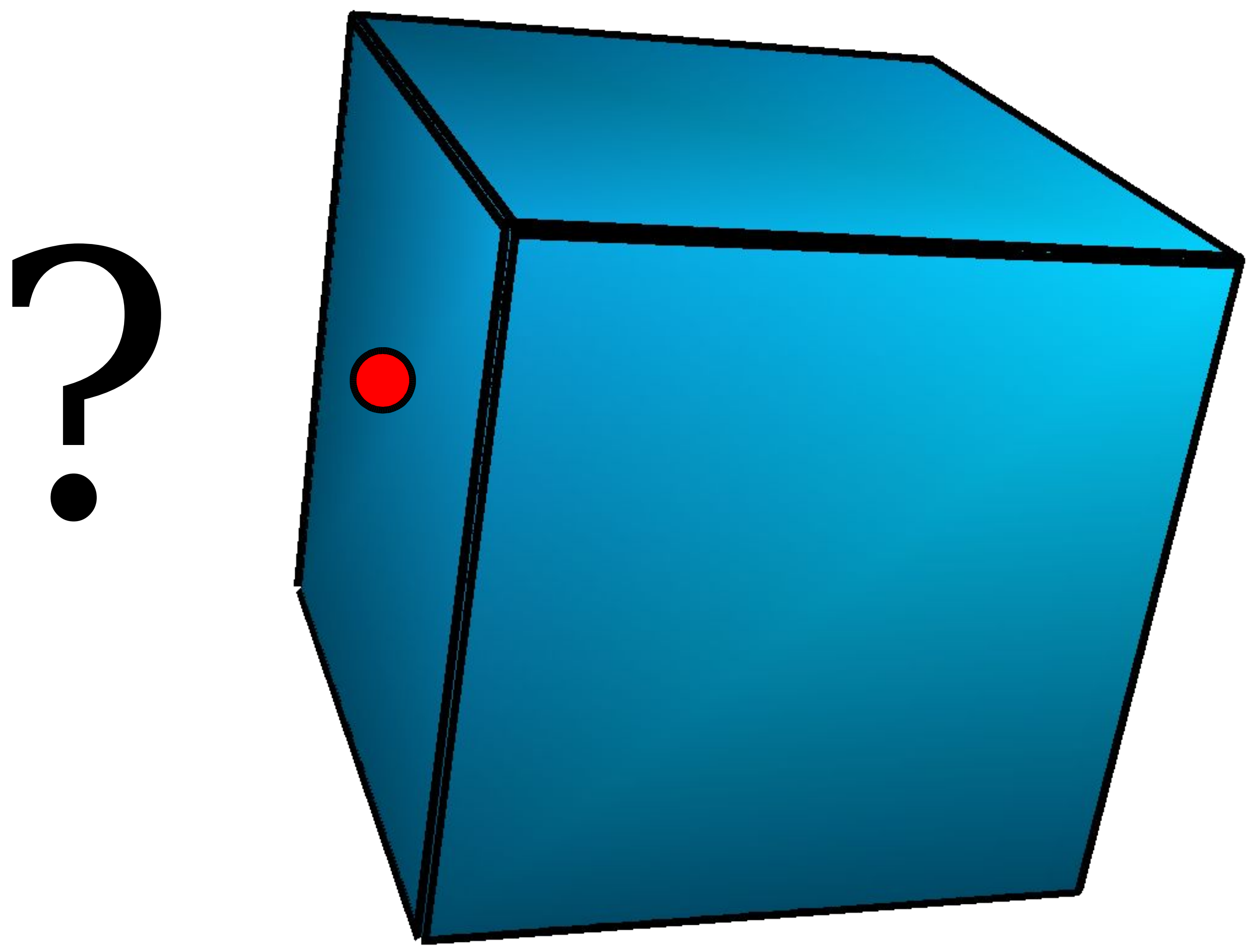
Overview Lecture 2

- Develop an understanding of the Monte Carlo Method
  - Passage of particles through matter
  - Random numbers, distributions
  - Basic tracking algorithm
- Free historic sightseeing tour
  - Beginning of the modern Monte Carlo Method
- Introduction to GEANT4
  - Particles and Materials

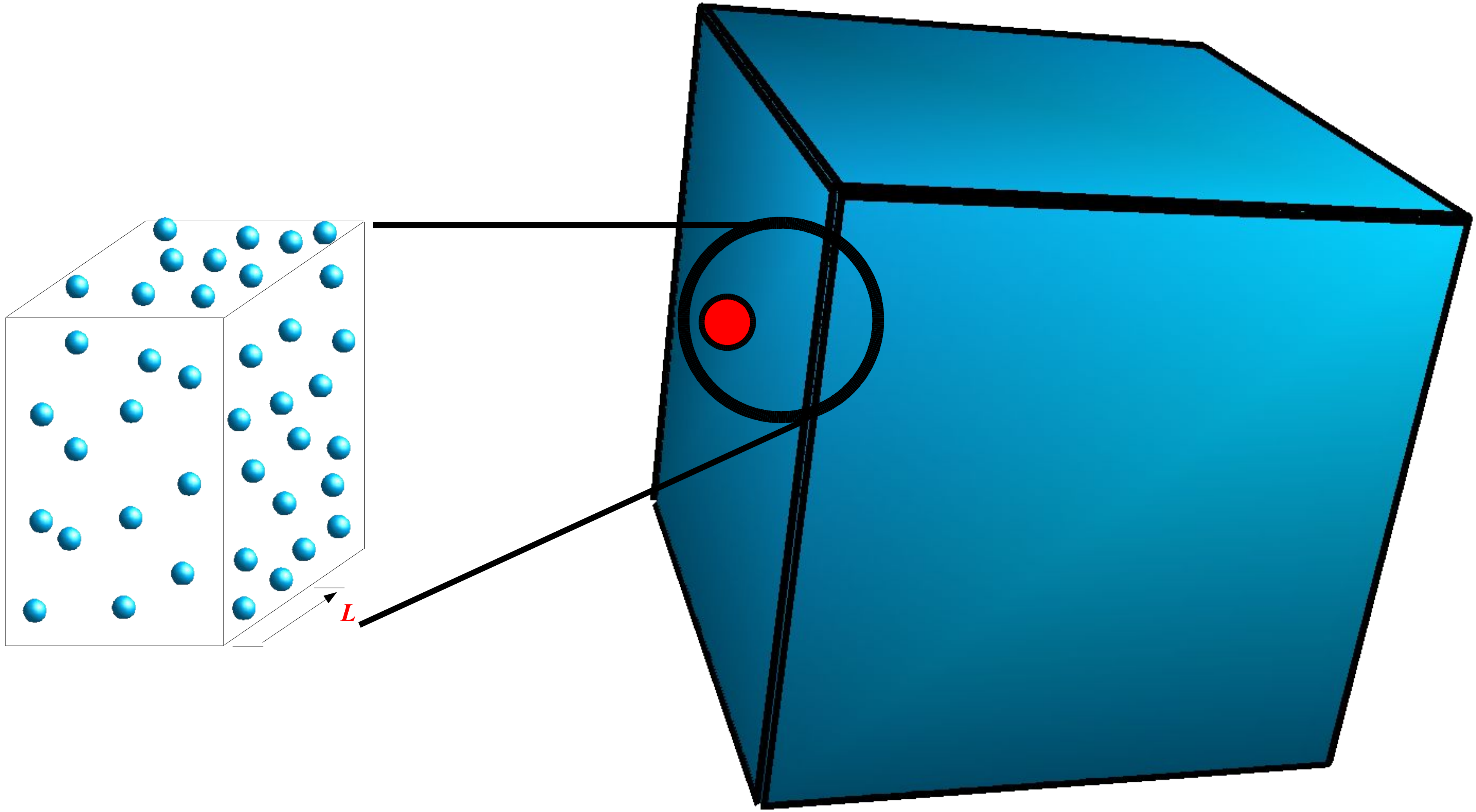
The Problem:



The Problem:

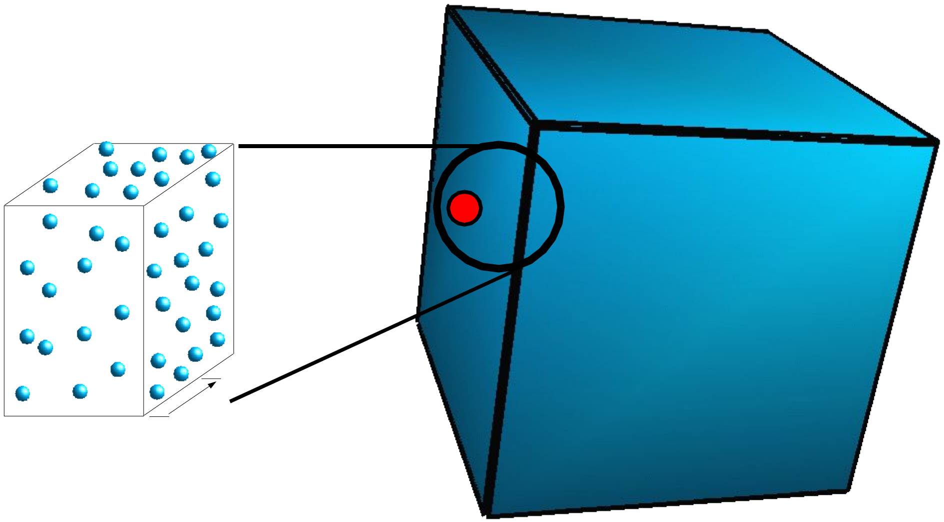


Approach to an answer ..



Block of material

What we know about ..



Block of material

Incident particle - microscopic properties:
type (e^+, μ^-,..), position, energy, momentum, spin, ..

Block of material – microscopic properties:
constituting element types & their properties (atomic charge, ..), mixture ratios (H_2O), molar masses

macroscopic properties:
density (g/cm^3), size/shape, state (solid, gaseous, ..), ..

Don't know any details, where exactly the atoms are!

So, the method to tackle the problem will be:

Monte Carlo Method

for Pedestrians

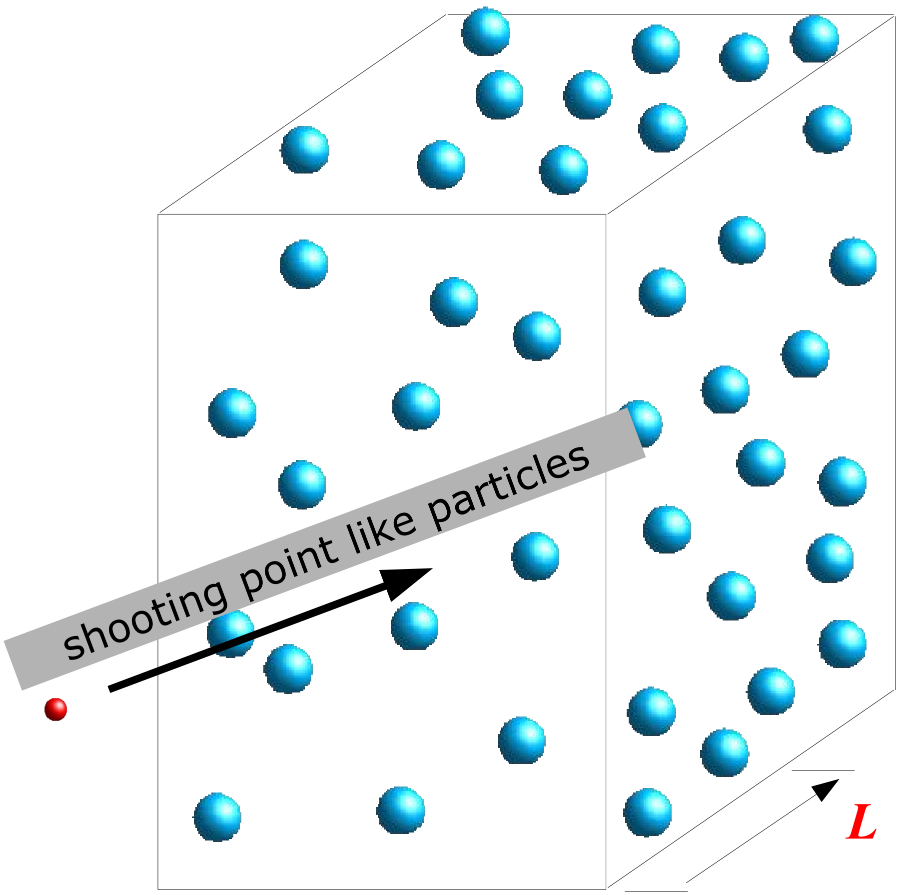
a pedestrian

What follows is a gentle

introduction to the Monte Carlo Method for simulating the passage of particles through matter

and it's application in GEANT4

Enters: Probability

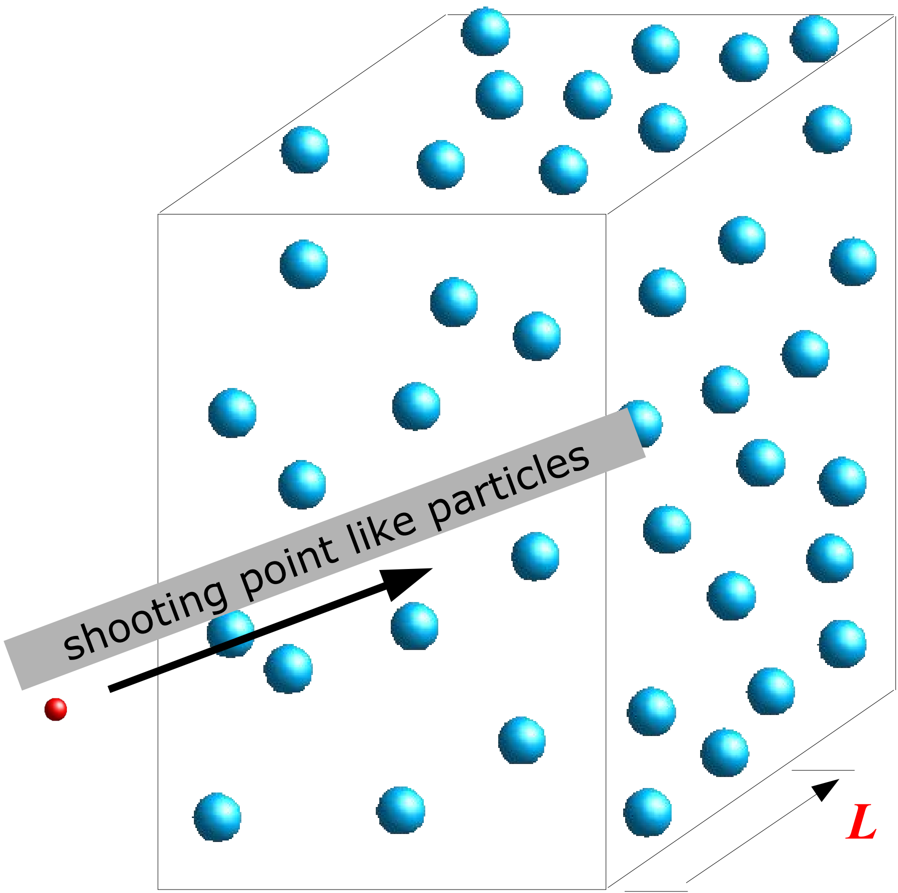


macroscopic
distance L

We want to find a method for efficiently using the computer to simulate tracks of single particles through bulk materials.

Because we don't know all the gory details of the bulk material, we will come up with a statistical method: the Monte Carlo method.

Enters: Probability

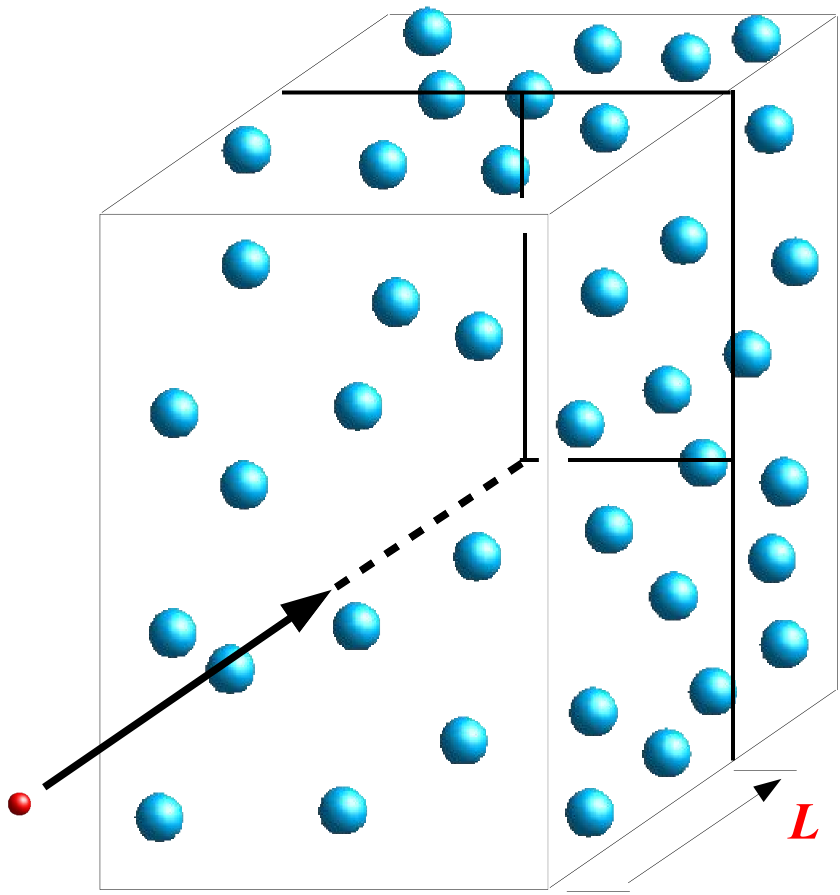


macroscopic distance L

We don't want to know all these details - for the sake of a feasible and efficient computational method!

Because we don't know all the gory details of the bulk material, we will come up with a statistical method: the Monte Carlo method.

Enters: Probability



macroscopic
distance L

We assume that the obstacles making up the bulk material are equally distributed at random!

How far can an incident particle travel without interacting with the bulk material?

What is the probability for an incident particle to travel undisturbed for a distance L?

What´s the probability for an incident particle to travel undisturbed for a distance L?

$$P(L) = \frac{\text{number particles not hitting anything until } L}{\text{total number of particles shot}}$$

(frequentist's definition)

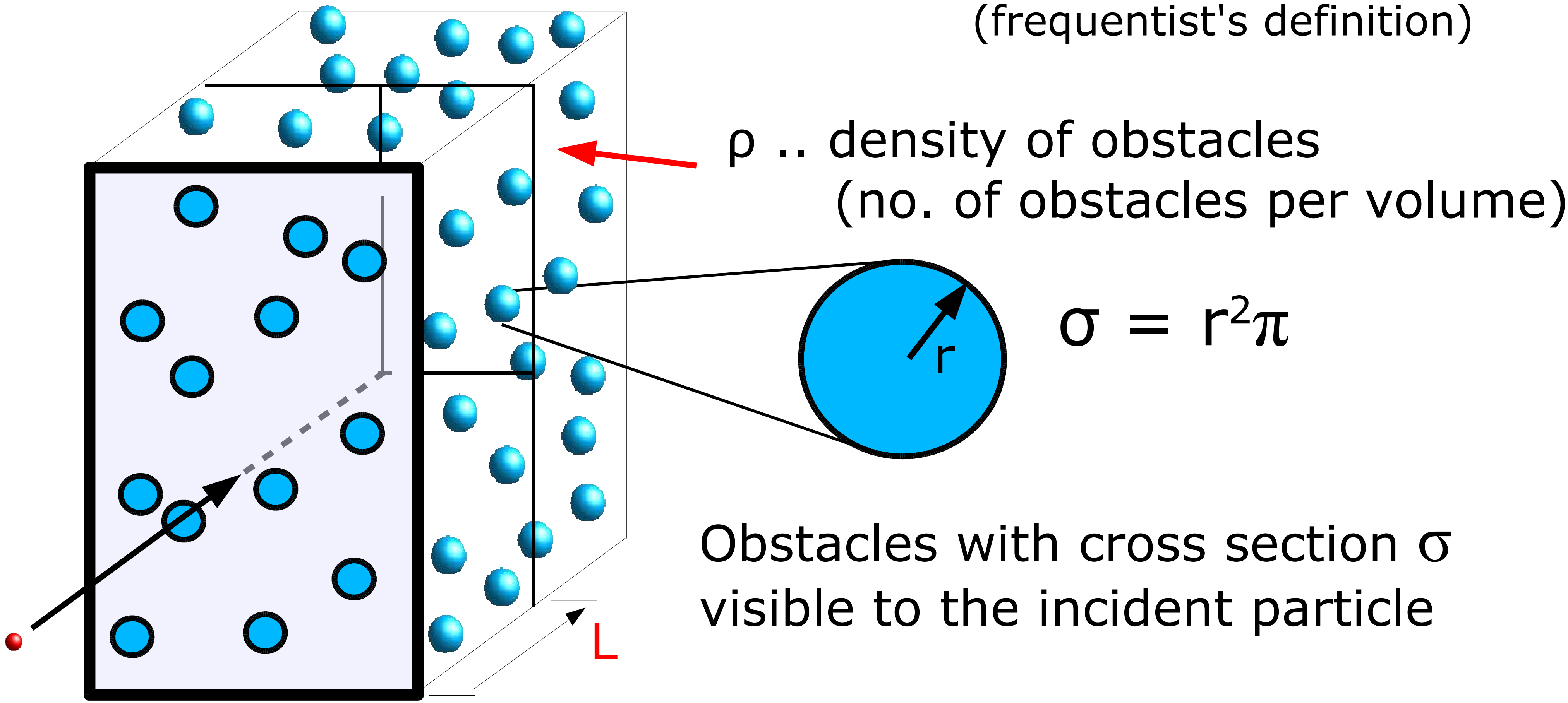
We have to ask for a probability, because we don't know

- about the exact distribution of obstacles in the bulk material
- even if we knew, the interaction itself is governed by the laws of quantum mechanics, which are probabilistic!

Find a formula for:

$$P(L) = \frac{\text{number particles not hitting anything until } L}{\text{total number of particles shot}}$$

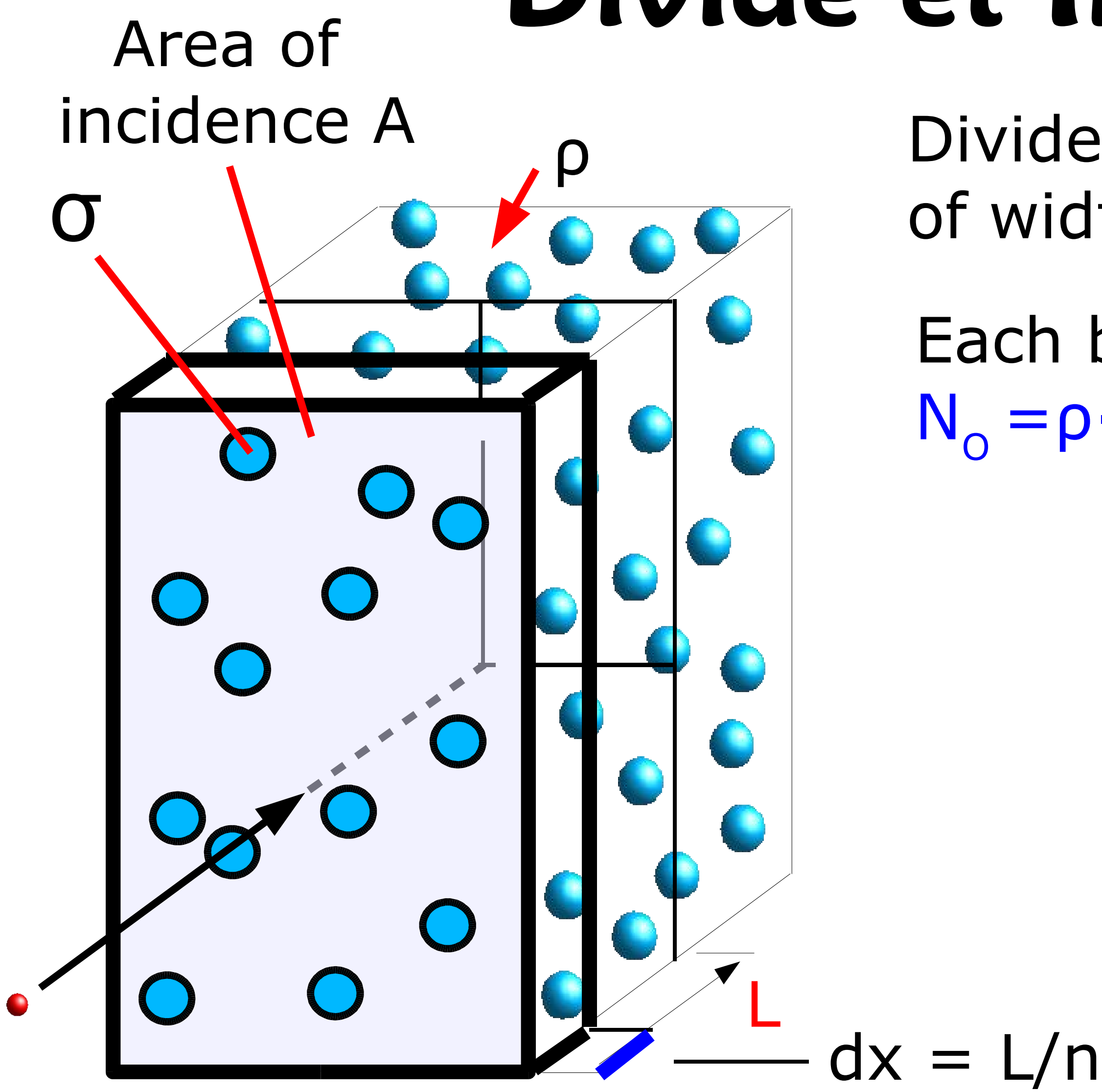
(frequentist's definition)



Obstacles with cross section σ visible to the incident particle

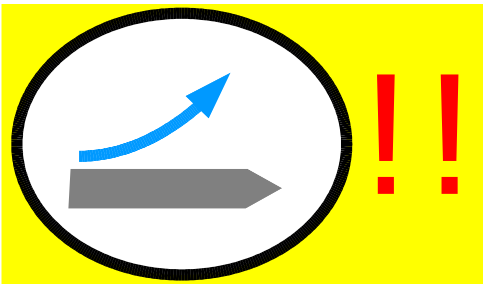
σ .. <u>cross section</u> ~ some interaction with the incident particle takes place if it hits the area of the cross section of one of the obstacles – otherwise it remains undisturbed.

"Divide et Impera!"



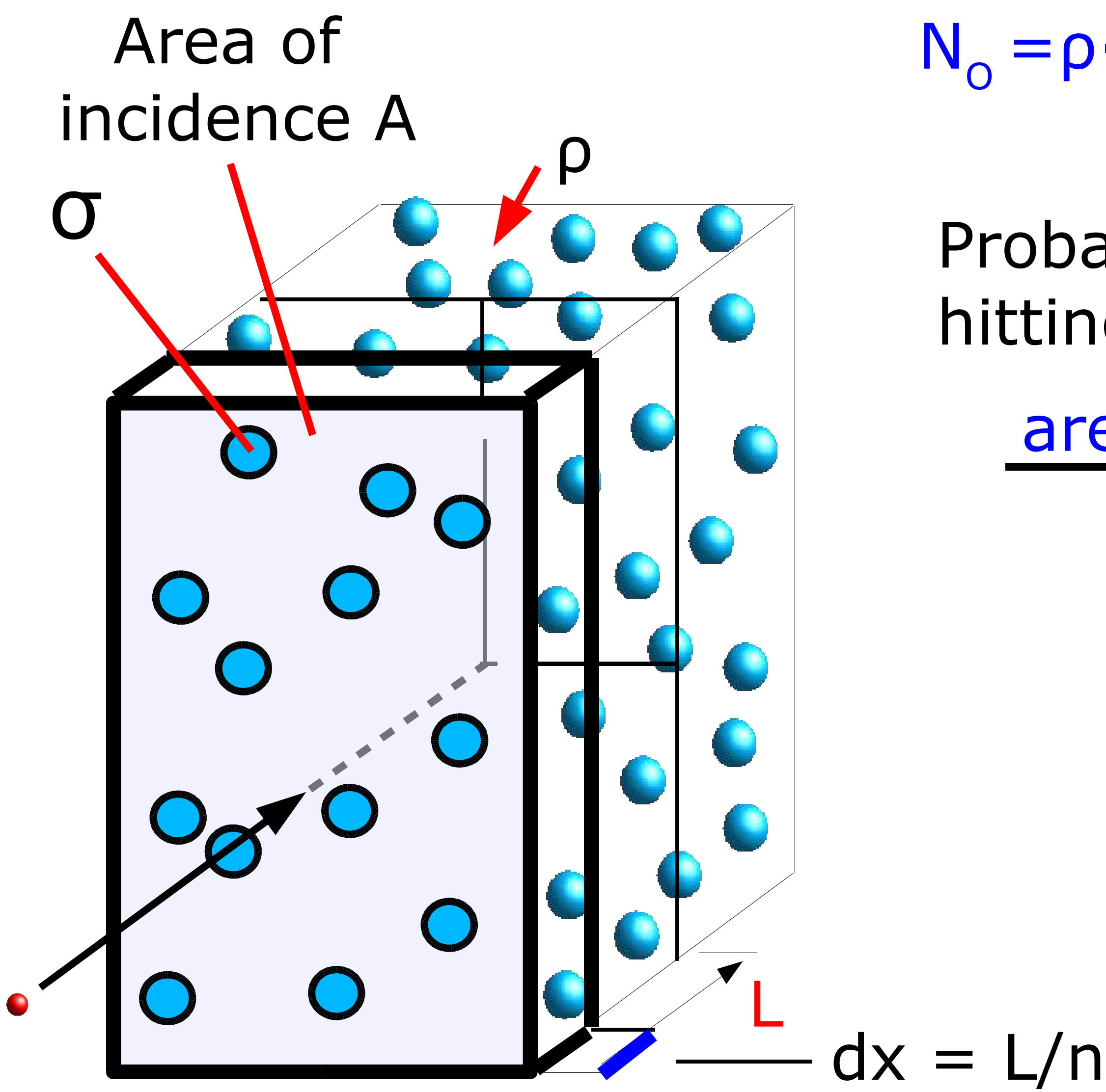
Divide the block into n slices of width $dx = L/n$

Each block then contains $N_o = \rho \cdot A \cdot dx = \rho \cdot A \cdot L/n$ obstacles



$$P(L) = \frac{\text{number particles not hitting anything until } L}{\text{total number of particles shot}}$$

(frequentist's definition)



Each block contains

$N_o = \rho \cdot A \cdot dx = \rho \cdot A \cdot L/n$ obstacles

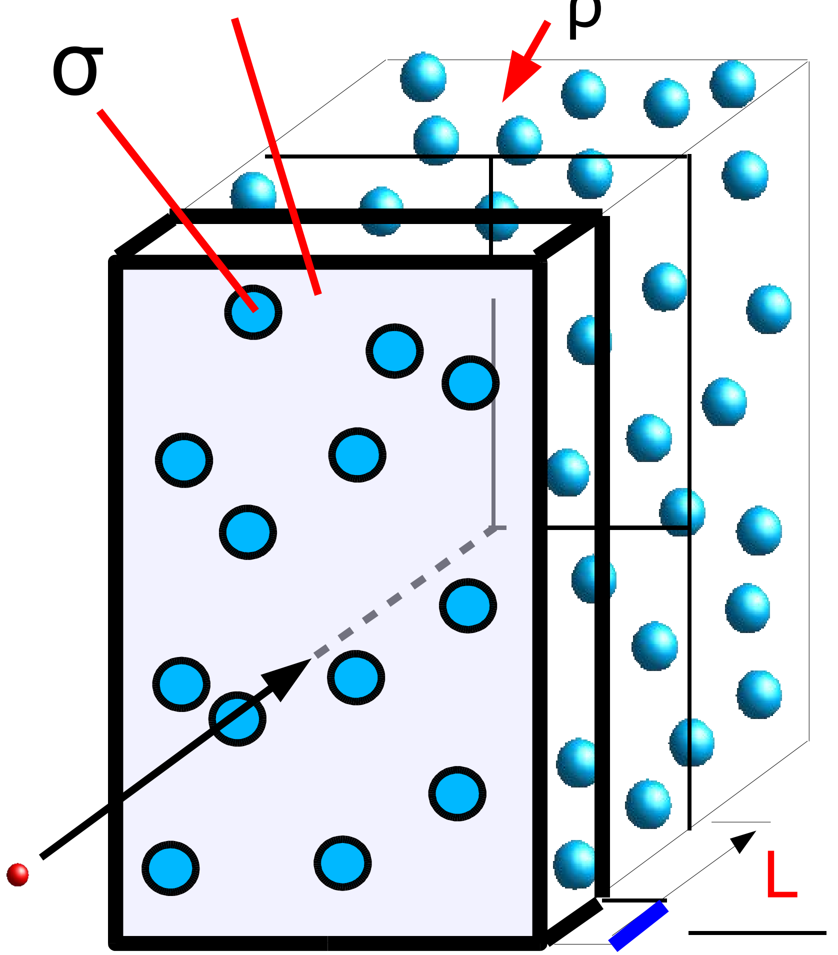
Probability to cross the slice hitting and obstacle =

$$\frac{\text{area of obstacles in slice}}{\text{area of incidence A}} =$$

$$\frac{N_o \cdot \sigma}{A} = \frac{\rho \cdot L \cdot \sigma}{n}$$

$$P(L) = \frac{\text{number particles not hitting anything until } L}{\text{total number of particles shot}}$$

(frequentist's definition)



dx = L/n

Probability to cross the slice hitting and obstacle =

$$\frac{\text{area of obstacles in slice}}{\text{area of incidence A}} =$$

$$P_{hit} = \frac{N_o \cdot \sigma}{A} = \frac{\rho \cdot L \cdot \sigma}{n}$$

Probability to cross the slice undisturbed:

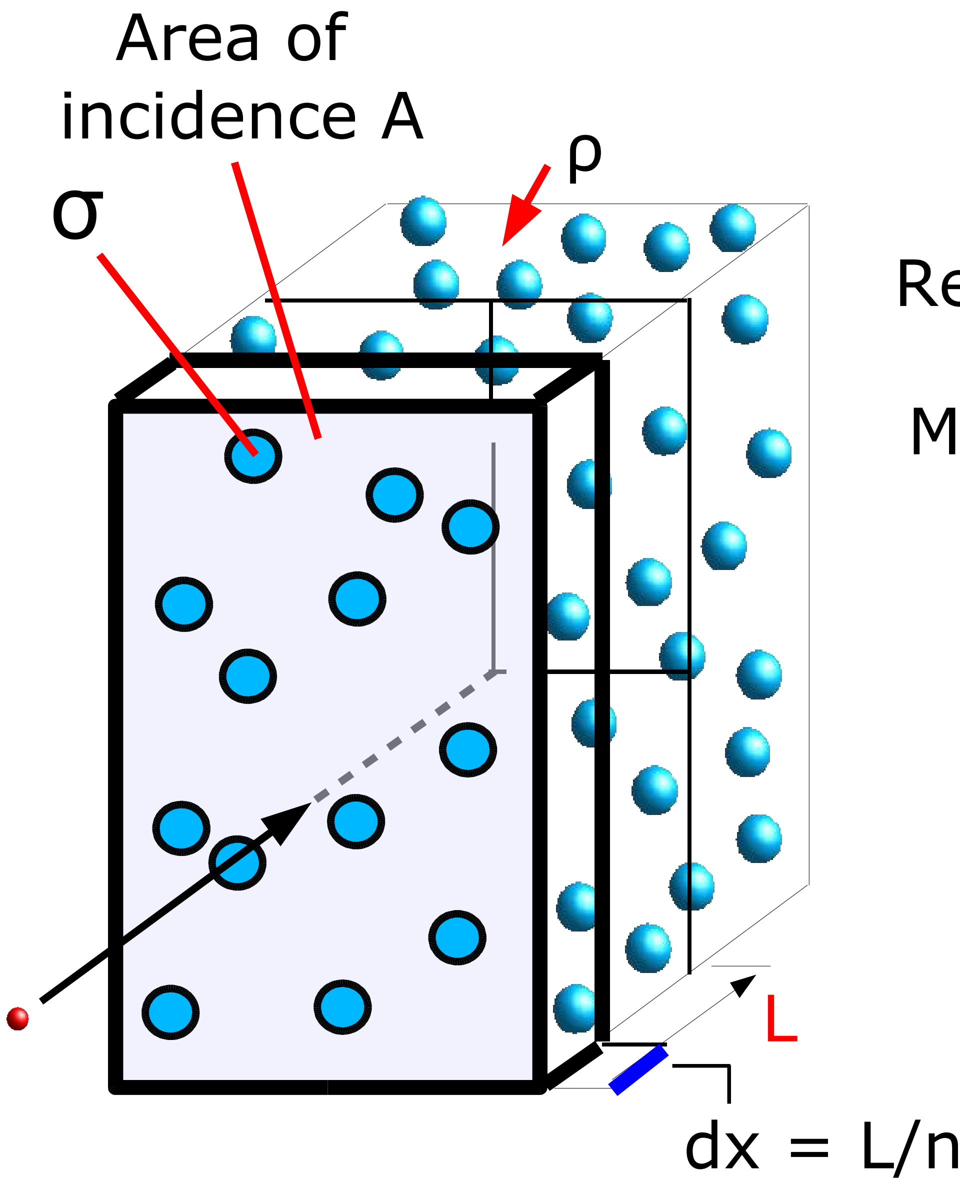
$$P_{no\text{-}hit} = 1 - P_{hit}$$

$$= 1 - \frac{\rho \cdot L \cdot \sigma}{n}$$

$$P(L) = \frac{\text{number particles not hitting anything until } L}{\text{total number of particles shot}}$$

(frequentist's definition)

Probability to cross the slice dx undisturbed:



$$P_{no\text{-}hit} = 1 - \frac{\rho \cdot L \cdot \sigma}{n}$$

Recall, we have n slices: $L = n \cdot dx$

Multiply $P_{no\text{-}hit}$ n times:

$$P(L) = P_{no\text{-}hit} \cdot P_{no\text{-}hit} \cdot \ldots \cdot P_{no\text{-}hit}$$

$$= (P_{no\text{-}hit})^n$$

$$= \left(1 - \frac{\rho \cdot L \cdot \sigma}{n}\right)^n$$

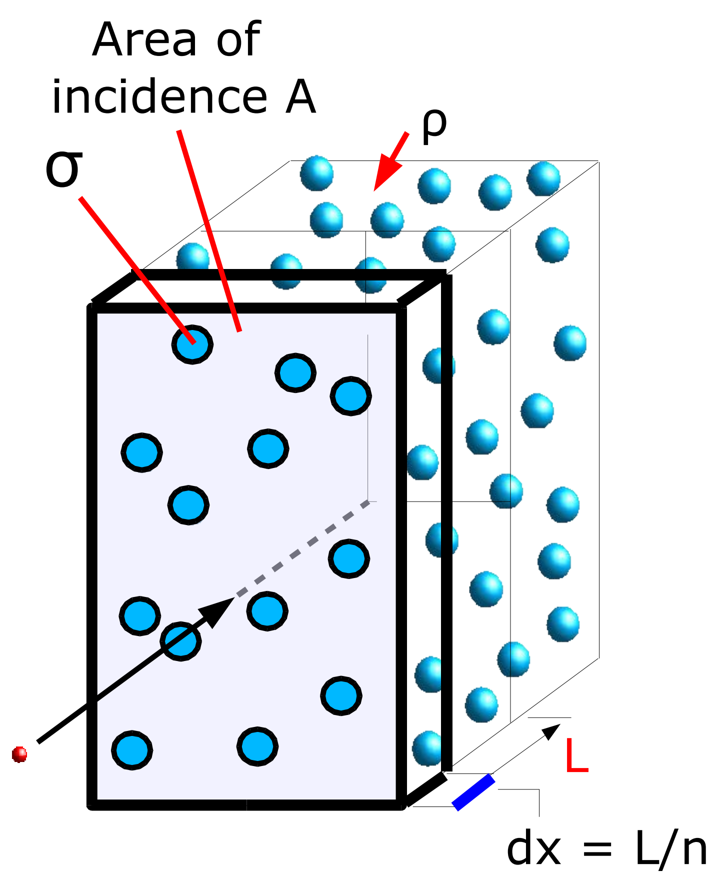
$$\sim \exp(-\rho \cdot L \cdot \sigma)$$

$$P(L) = \frac{\text{number particles not hitting anything until } L}{\text{total number of particles shot}}$$

(frequentist's definition)

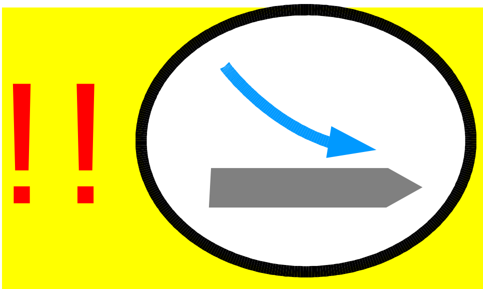
$$P(L) = \frac{\text{number particles not hitting anything until } L}{\text{total number of particles shot}}$$

(frequentist's definition)



$$P(L) = \exp(-\rho \cdot L \cdot \sigma)$$

Note: The probabilistic character of this result is due ONLY to the fact of our lack of knowledge concerning the exact distribution of interaction centers in the bulk material!

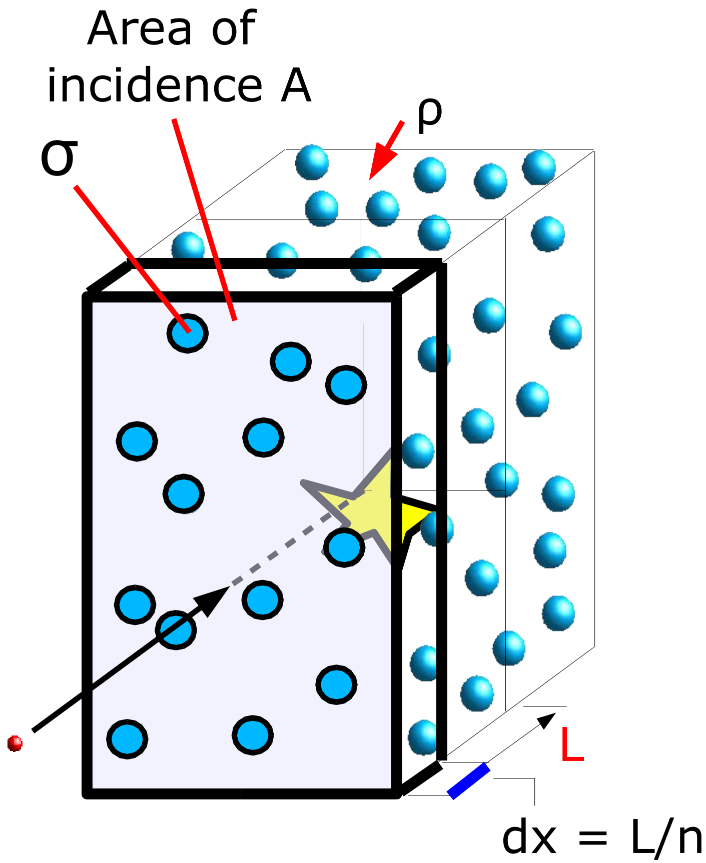


Traveling undisturbed until L:

$$P(L) = \exp(-\rho \cdot L \cdot \sigma)$$

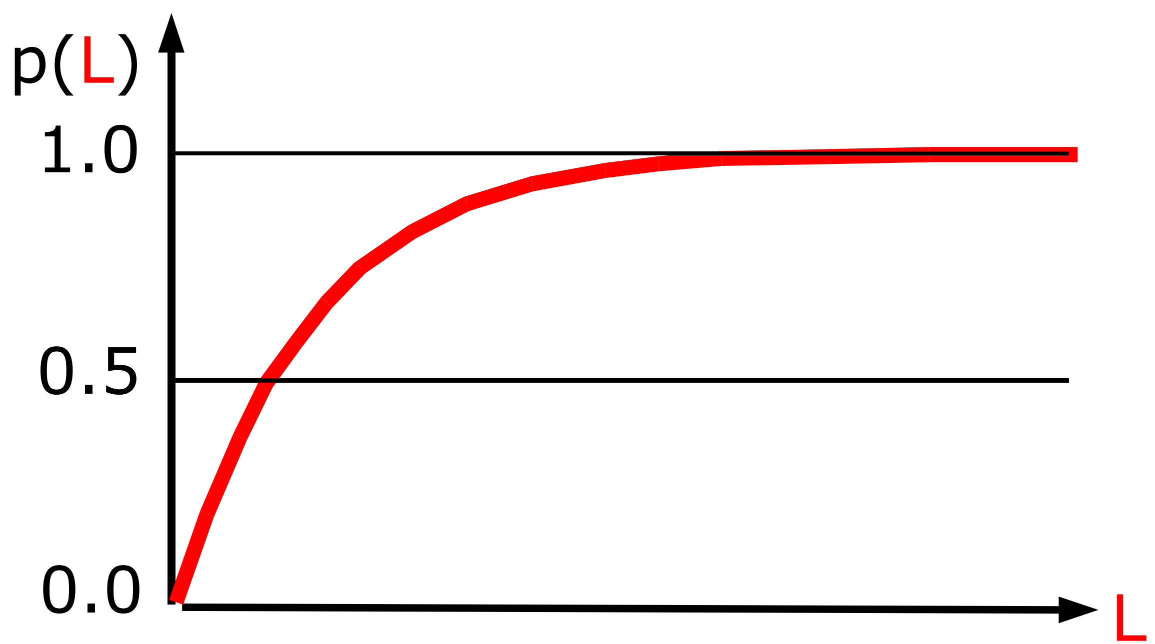
$$1 = P(L) + p(L)$$

undisturbed — 1st interaction



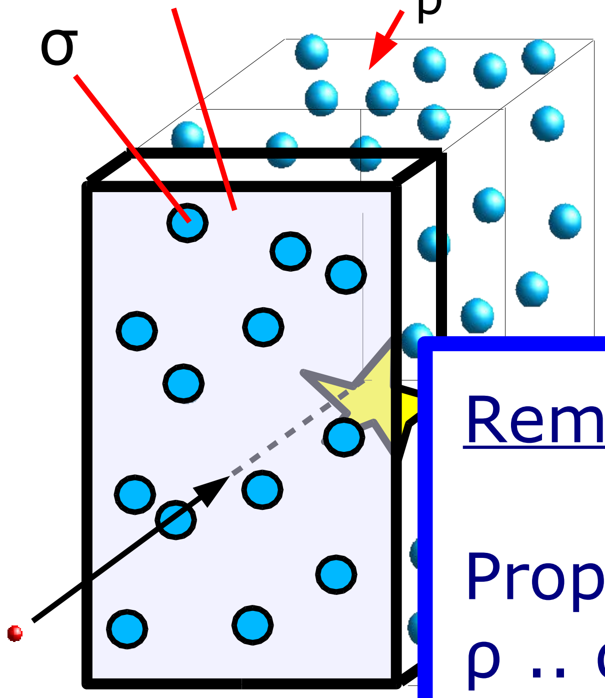
Probability to have an interaction at distance L:

$$p(L) = 1 - P(L)$$
$$= 1 - \exp(-\rho \cdot L \cdot \sigma)$$



Traveling undisturbed until L:

$$P(L) = \exp(-\rho \cdot L \cdot \sigma)$$



Probability to have an interaction at distance L:

$$p(L) = 1 - P(L) = 1 - \exp(-\rho \cdot L \cdot \sigma)$$

Reminder:

Properties of the bulk material:
ρ .. densities of interaction centers

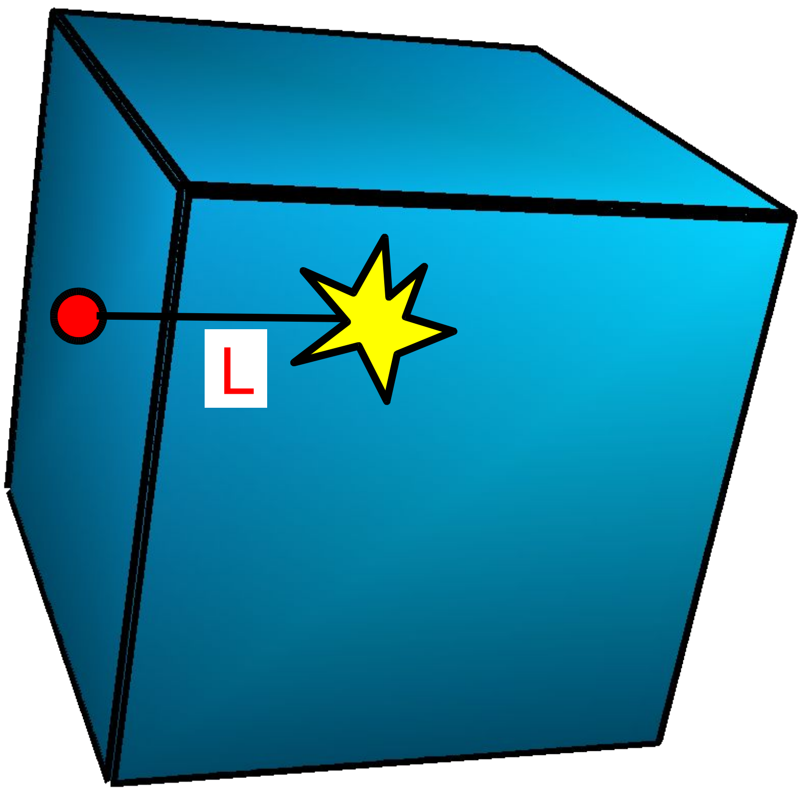
Nature of the microscopic particle interaction:
σ .. cross section of the particle interaction
σ = σ(incident particle, momentum, ..., type of interaction center)

A "probable" Answer!

The probability of having an interaction at distance L:

$$p(L) = 1 - \exp(-\rho \cdot L \cdot \sigma)$$

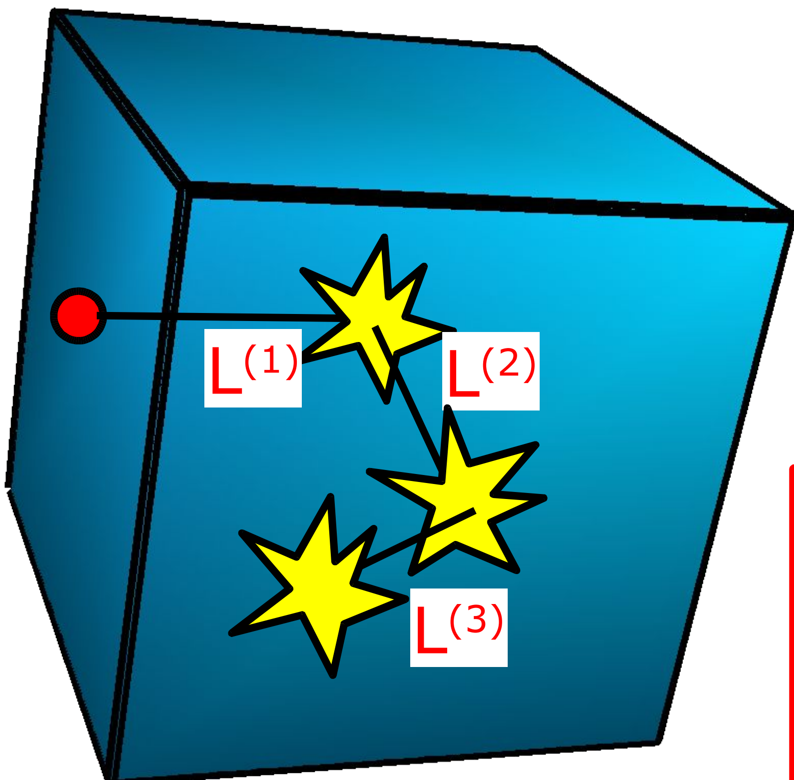
(the exponential distribution!)



Block of material

Basic Algorithm:

The probability of having an interaction at distance L:



$$p(L) = 1 - \exp(-\rho \cdot L \cdot \sigma)$$

(the exponential distribution!)

Very basic algorithm:

(1) start values for incident particle
(2) get values for ρ and σ
(3) sample $L^{(n)}$ from p(L)
(4) transport particle undisturbed by $L^{(n)}$
(5) simulate interaction
(6) if particle still exists: goto (1)

Core of the Monte Carlo Method: drawing of random numbers!

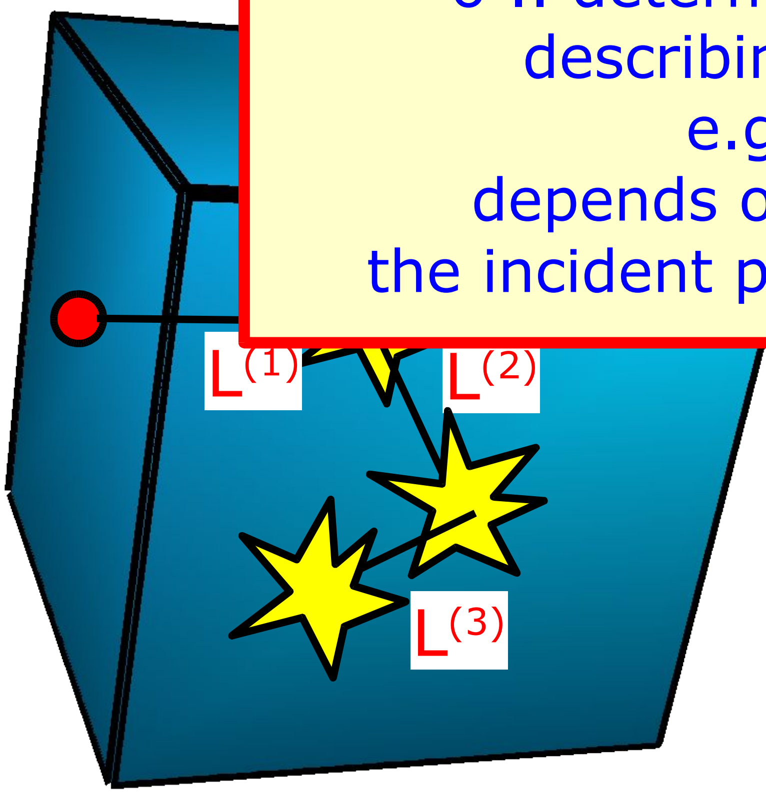
ρ .. determined from the material description and the type of process, e.g. density of charged nuclei with Z=16

σ .. determined by the quantum theory describing the interaction process, e.g. ionization process; depends on type of process & state of the incident particle (energy, momentum, ..)

The pro

!)

cle



(2) get values for ρ and σ

(3) sample $L^{(n)}$ from p(L)

(4) transport particle undisturbed by $L^{(n)}$

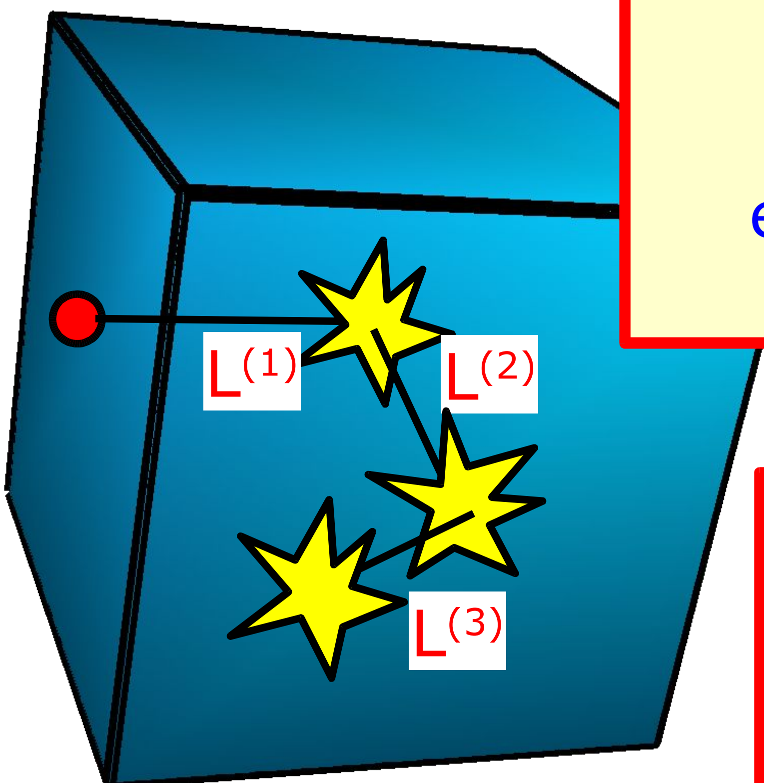
(5) simulate interaction

(6) if particle still exists: goto (1)

Core of the Monte Carlo Method: drawing of random numbers!

Basi

The probability of ha



interaction described by quantum mechanics determining σ,
→ distributions, random numbers, Monte Carlo method again!!

Outcome: new state of the incident particle (change of energy, momentum, ...) AND other particles!!!

e

(2) get values for ρ and σ
(3) sample $L^{(n)}$ from p(L)
(4) transport particle
undisturbed by $L^{(n)}$
(5) simulate interaction
(6) if particle still exists:
goto (1)

Core of the Monte Carlo Method:
drawing of random numbers!

Mean Free Path Length

$$p(L) = 1 - \exp(-\rho \cdot L \cdot \sigma)$$
$$= 1 - \exp(-L/\lambda)$$

$\lambda := 1/(\rho \cdot \sigma)$ ... mean free path length,

average distance a particle moves undisturbed with respect to process i

$\lambda = \lambda$(particle type, momentum, ..., density of interaction centers, type of interaction center)

microscopic and macroscopic properties!

Mean Free Path Length, more than one interaction process

$$p_i(L) = 1 - \exp(-\rho_i \cdot L \cdot \sigma_i)$$

$$= 1 - \exp(-L / \lambda_i)$$

$\lambda_i := 1/(\rho_i \cdot \sigma_i)$ ... mean free path length,

average distance a particle moves undisturbed with respect to process i

Process 1 has λ_1:

Process 2 has λ_2:

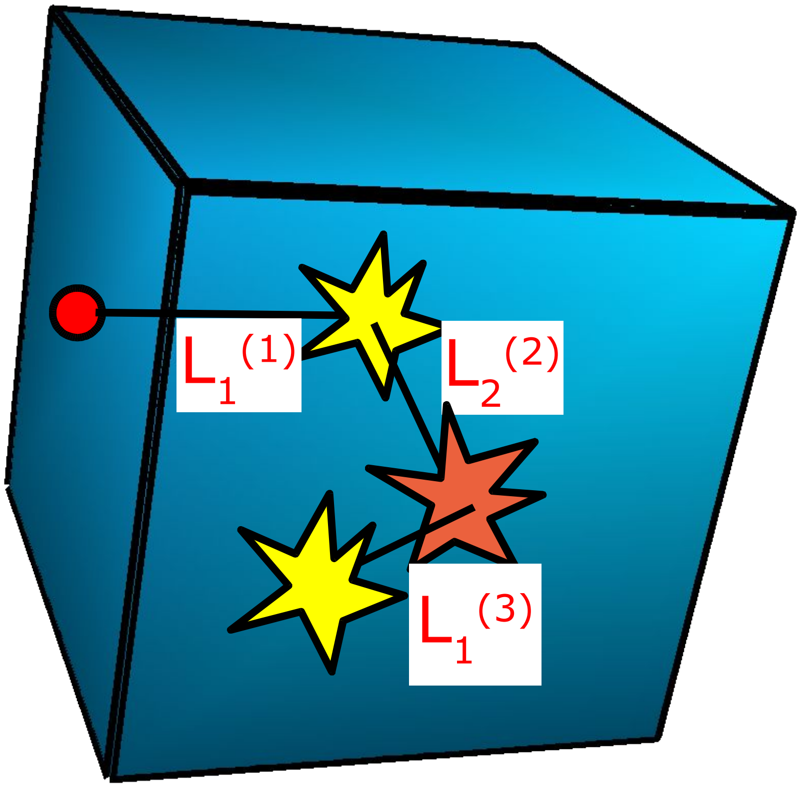
...

Process m has λ_m:

$\lambda_i = \lambda_i$(particle type, momentum, ..., density of interaction centers, type of interaction center)

microscopic and macroscopic properties!

Back to the "probable" Answer ..



Usually, there are more (i=1,2,..,m) processes responsible for interactions. For each of them we have the probability:

$$p_i(L) = 1 - \exp(-\rho_i \cdot L \cdot \sigma_i)$$

ρ_i .. density of interaction centers for physics process i

σ_i .. cross section of physics process i

Monte Carlo Algorithm

The free path lengths for an incident particle for each process are distributed according to the exponential distribution determined by the mean free path length of this process and material:

$$p_i(x) = 1 - \exp(-x / \lambda_i)$$

(1) set properties for incident particle (energy, momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

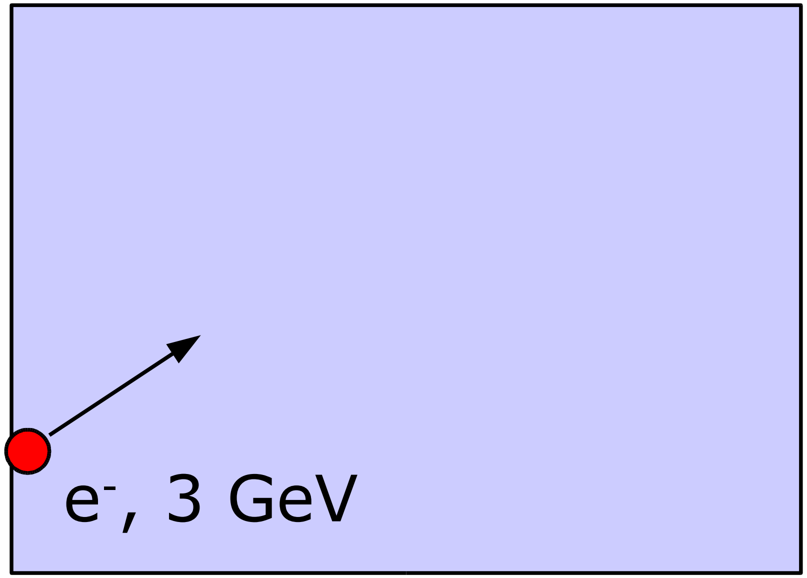
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

A little Example



(1) set properties for incident particle (energy, momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

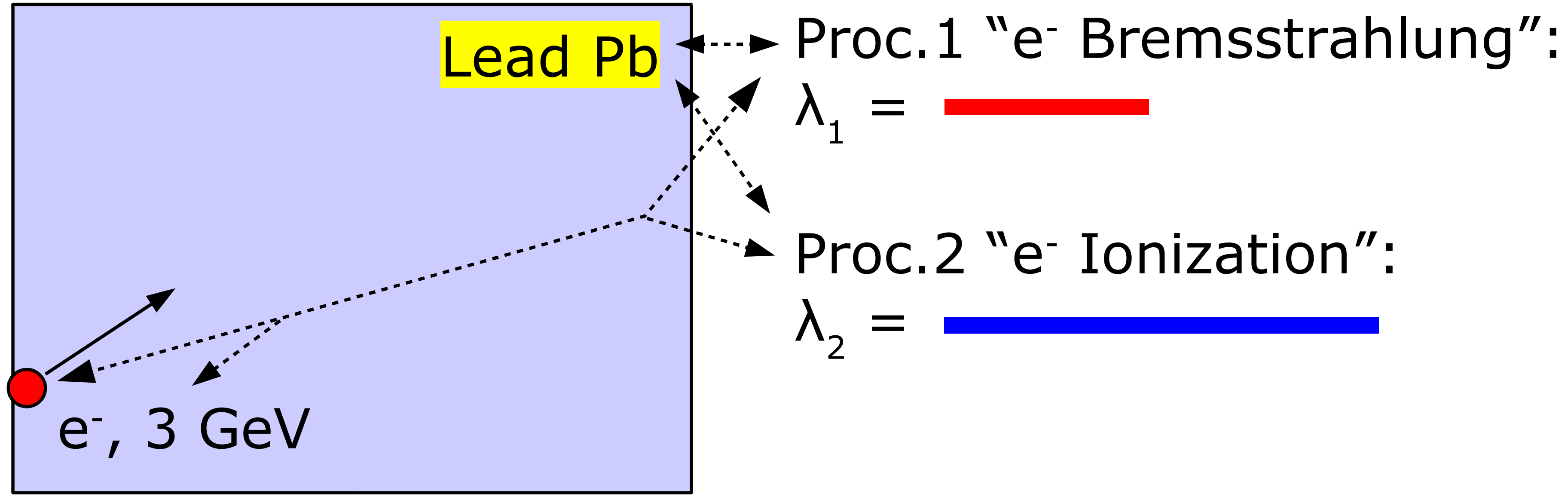
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

Let's take only 2 processes:



(1) set properties for incident particle (energy, momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

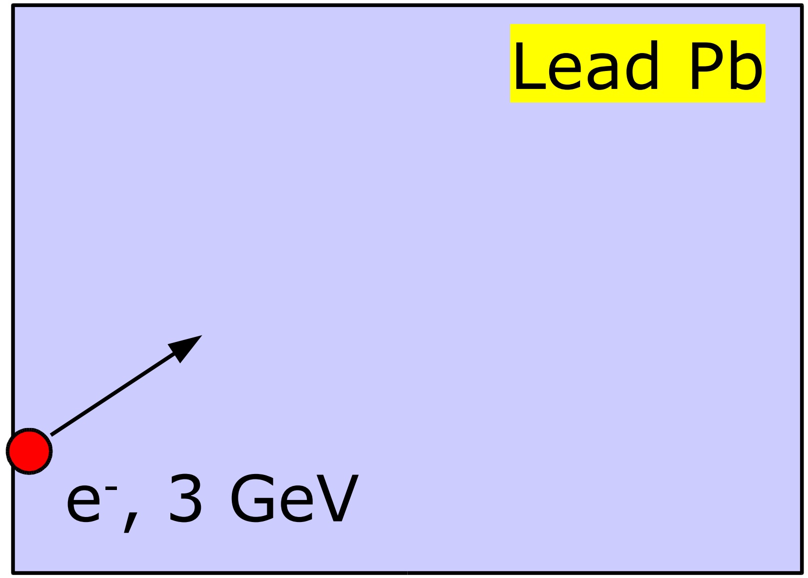
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

Roll the Dice!



Proc.1 “e- Bremsstrahlung”:
λ_1 =

Proc.2 “e- Ionization”:
λ_2 =

$$p_i(L) = 1 - \exp(-L/\lambda_i)$$

$L_1^{(1)}$ =

$L_2^{(1)}$ =

(1) set properties for incident particle (energy, momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):
    sample L_i from $p_i(x)$

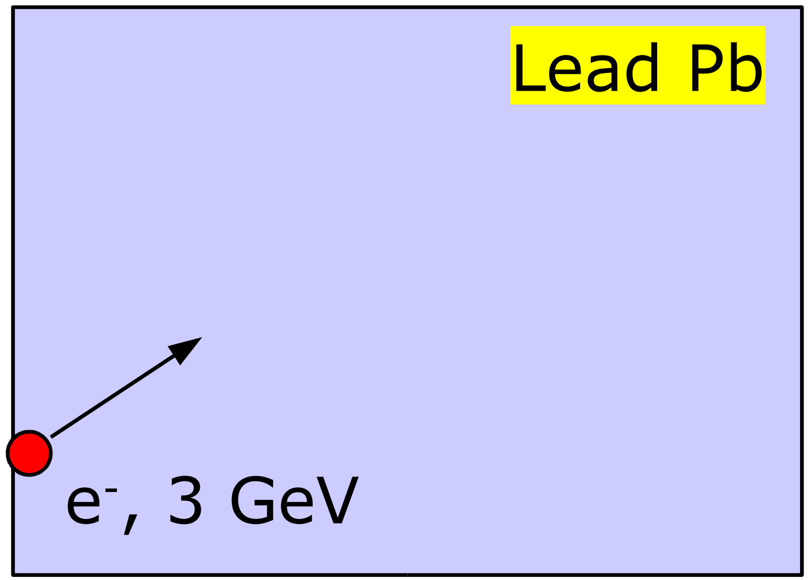
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

First come, first served ..



Proc.1 "e^- Bremsstrahlung":
λ_1 =

Proc.2 "e^- Ionization":
λ_2 =

$$p_i(L) = 1 - \exp(-L/\lambda_i)$$

$L_1^{(1)}$ =

$L_2^{(1)}$ =

(1) set properties for incident particle (energy, momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

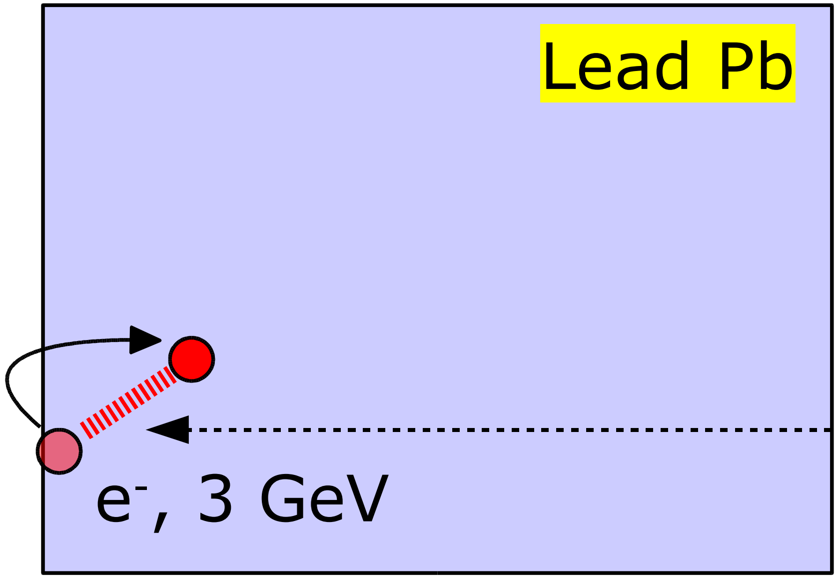
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

Move it!



Proc.1 "e^- Bremsstrahlung":
λ_1 =

Proc.2 "e^- Ionization":
λ_2 =

$$p_i(L) = 1 - \exp(-L/\lambda_i)$$

$L_1^{(1)}$ =

$L_2^{(1)}$ =

(1) set properties for incident particle (energy, momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

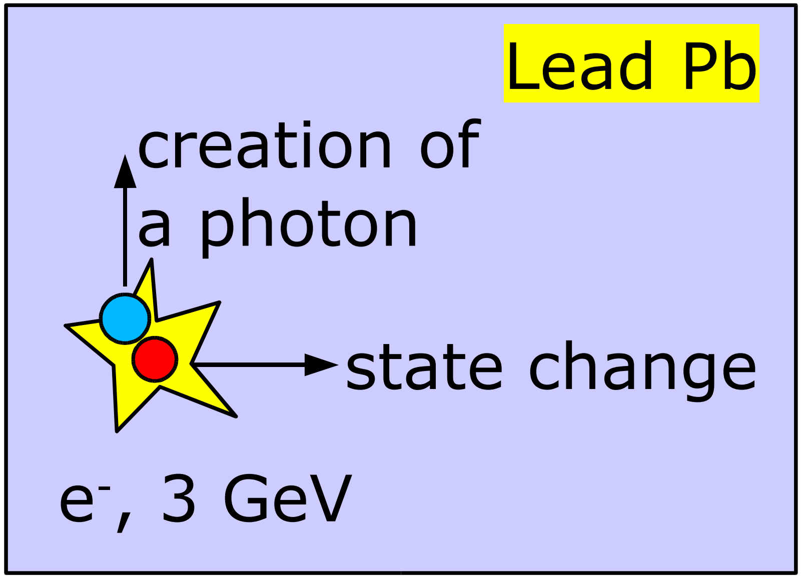
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

"... and (Inter)Action!"



Proc.1 "e⁻ Bremsstrahlung":
λ_1 =

Proc.2 "e⁻ Ionization":
λ_2 =

$$p_i(L) = 1 - \exp(-L/\lambda_i)$$

$L_1^{(1)}$ =

$L_2^{(1)}$ =

(1) set properties for incident particle (energy, momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

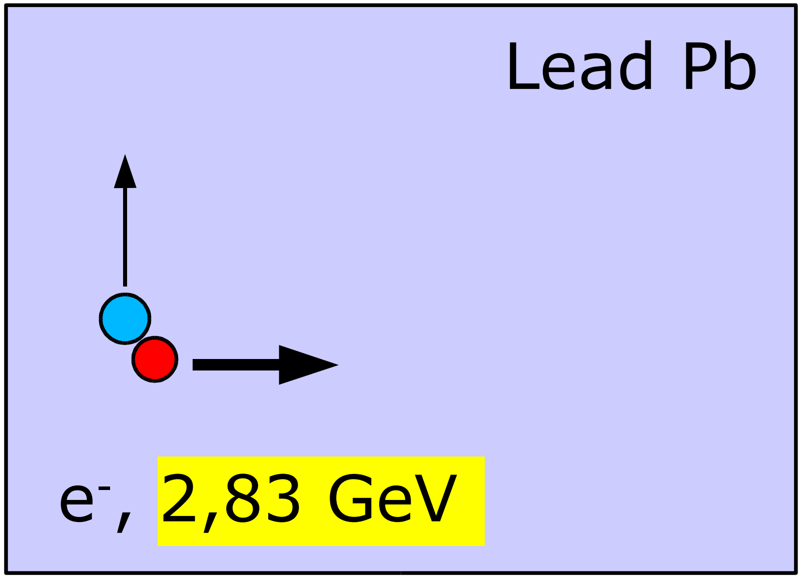
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

Da Capo!



Proc.1 "e^- Bremsstrahlung":
λ_1 =

Proc.2 "e^- Ionization":
λ_2 =

(1) set properties for incident particle (energy, momentum, ..)

(2) get values for λ_i for all relevant processes i=1,2,..,m

(3) for each process i (i=1,2,..,m):

sample L_i from $p_i(x)$

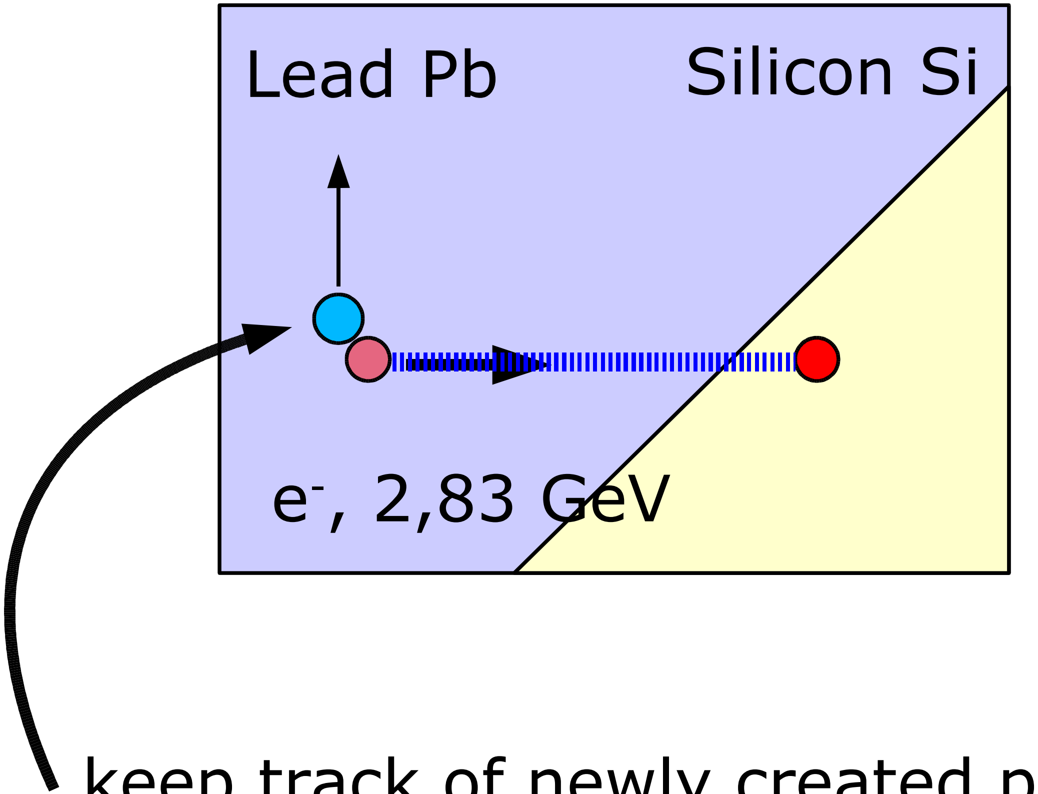
(4) L_c = min(L_i) from all sampled L_i , c in (1,2,..,m)

(5) transport incident particle by L_c

(6) simulate interaction

(7) if particle still exists: goto (1)

What GEANT4 does for you (additionally)



re-scale the free path length once a particle enters another material during an undisturbed push forward

keep track of newly created particles ("secondaries")
- they will be simulated once the "primary" is done

We will come back to describing how the before mentioned concepts map to GEANT4 classes (Lecture 3, 4).

Some thoughts about random numbers ...

Some thoughts about Random Numbers

When do we need random numbers?

Whenever states of processes can "only" be described by means of probability distributions

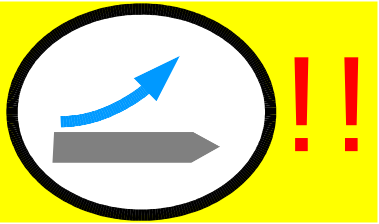
and

we want to study particular samples of states described by these distributions

Definitions

Disclaimer: This section gives only a very brief and incomplete introduction in order to gain some understanding of what happens behind the scenes in a Monte Carlo simulation toolkit such as GEANT4.

x_i ... random variables describing the state of a system



$g_j(x)$ ... observables as functions of the x_i's

$f(x)$ ... probability density function – our best knowledge of the system

$\int_{x^{(1)}}^{x^{(2)}} f(x)dx$ ... probability of x being between $x^{(1)}$ and $x^{(2)}$

special case: cumulative distribution function $F(x)$

$F(a) = \int_{-\infty}^{a} f(x)dx$ ... probability that $x \leq a$ $\left(-\infty \ldots \begin{array}{l}\text{start of the interval}\\ \text{on which x is defined}\end{array}\right)$

$\int f(x) \cdot g(x)dx$ ... expectation (mean) value

The Problem

Given:

f(x) ... probability density function – our best knowledge of the system, or equivalently F(a)

Task:

Find a statistical ensembles $\{x^{(m)}\}$ having F(a) as its underlying distribution!

Or, the other way round:

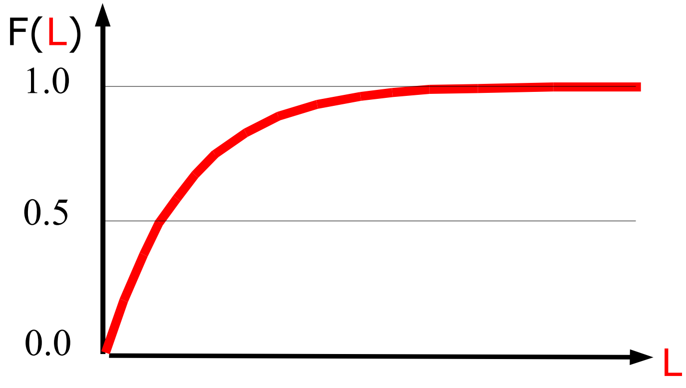
How can we produce a series $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ such that, given this series, we can conclude that it stems from the distribution f(x) whenever m gets large.

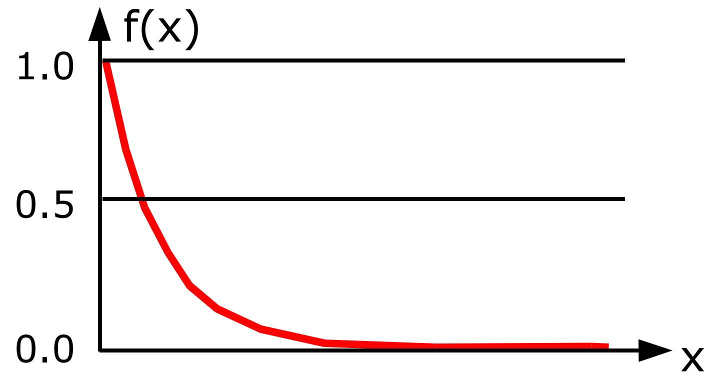
Our Example

$F(L) = 1 - \exp(-L/\lambda)$ ... probability that $x \leq L$,
i.e. probability of having the first interaction before L when the incident particle starts off at x=0

$f(x) = 1/\lambda \cdot \exp(-x/\lambda)$

Find a series $\{x^{(i)}\} = x^{(1)}, x^{(2)}, \ldots$ where each $x^{(i)}$ "could" be a distance to the first interaction!





The Solution ...

It turns out, being able to generate random number of only one distribution solves the problem related to the randomness-aspect!

If we are able, for example, to generate samples of of equally distributed numbers in the interval [0,1), we can re-use them to calculate random numbers of any other distribution!

f(x) = 1 defined for x in [0,1)

F(x) = x defined for x in [0,1)

Inverse Method

random number generator: RG.generate() spits out a random number of equally distributed numbers in [0,1)

Want to have a random number of any other distribution f(x):

$$x^{(i)} = F^{-1}(\text{RG.generate}())$$

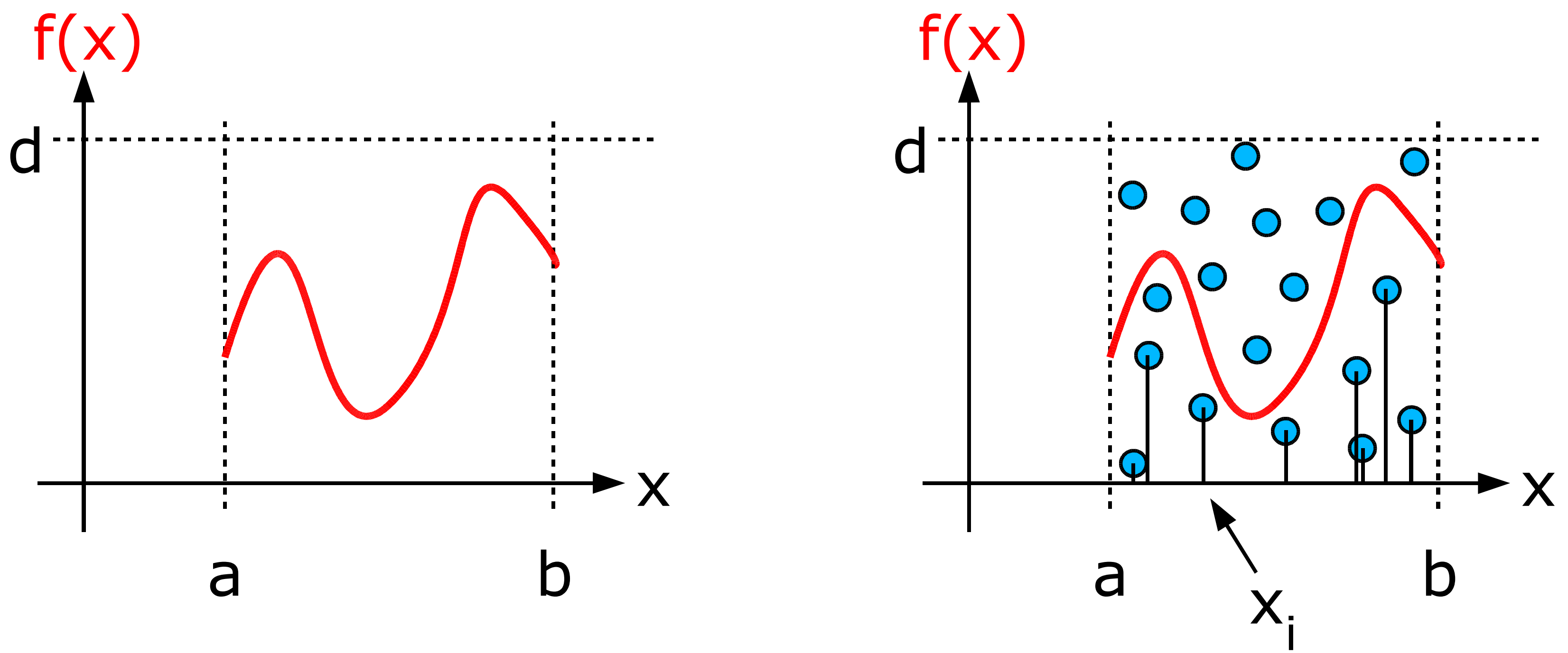
By applying the inverse cumulative distribution function on a set of [0,1) equally distributed random numbers produces random numbers being f(x)-distributed!

Accept – Reject Method

f(x) ... density function
x ... vector of m [a,b) uniformly distributed random numbers
y ... vector of m [0,d) uniformly distributed random numbers

accept x_i for which $y_i < f(x_i)$

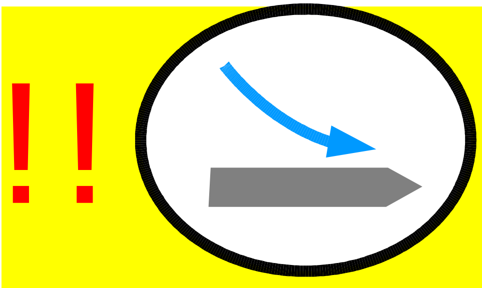
→ the accepted x_i are f(x)-distributed!



Random Number Conclusion

If you have random numbers of one particular distribution (of course, you must know which distribution), then:

- random numbers of any other distribution can be derived from the given ones
- several methods exists, we have seen two:
  - inverse method
  - accept – reject method



Typically …

Typically, Monte Carlo toolkits such as GEANT4 have one single instance of a random number generator/engine for [0,1) equally distributed random numbers.

- single source of “randomness”
- apply methods such as the accept-reject or inverse methods or more sophisticated versions thereof to generate all other kinds of random numbers

Typically …

Typically, Monte Carlo toolkits such as GEANT4 have one single instance of a random number generator/engine for [0,1) equally distributed random numbers.

- single source of "randomness"
- apply methods such as the accept-reject or inverse methods or more sophisticated versions thereof to generate all other kinds of random numbers

The engine produces <u>pseudo random</u> numbers!

- random numbers that are not random, but generated by a well defined algorithm
- <u>reproducibility!</u>
- pseudo random numbers must not be
  - biased
  - not re-occur in cycles/patterns

Ty… …e
on… …engine
fo…

- s…
- a… …e
  m…
  g…

Not an easy task!

"Good" number generators
- must pass various statistical tests
- should be algorithmically "fast"

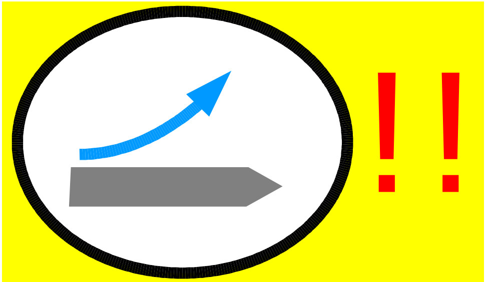
Usually configurable engines
with different speed/quality ratios

Remember: finally ALL your simulation results
depend on these numbers!

The engine produces <u>pseudo random</u> numbers!
- random numbers that are not random, but generated by a well defined algorithm
- <u>reproducibility!</u>
- pseudo random numbers must not be
  - biased
  - not re-occur in cycles/patterns

Uff,
time to relax a bit!

The Monte Carlo Method
a free historic sight-seeing tour

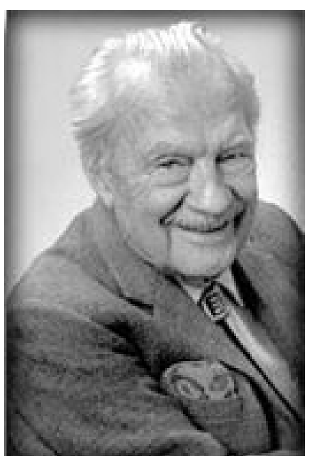


- The law of particle physics are based on quantum mechanics
  - Non deterministic in the sense that one can "only" calculate the probabilities of certain things to happen
  - Results are valid not for a single particle/interaction, but for an (statistical) ensemble
- An experimental measurement can be regarded as drawing a sample from the ensemble of possible outcomes (both, in reality and in simulation)
- When quantum mechanics became "applicable" to large projects, people had to invent tools to estimate parameters for their design of reactors, bombs, ...
  - the Monte Carlo Method was born / re-invented / applied at large scale

Monte Carlo

This wonderful method is indeed named after:

Ulam, Stanislaw
(1909-1986)



Metropolis, Nicholas Constantine
(1915-1999)

**Metropolis Nicholas and Stanislaw Ulam (1949).
The Monte Carlo method,
Journal of the American Statistical Association,
44 (247), 335-341.**

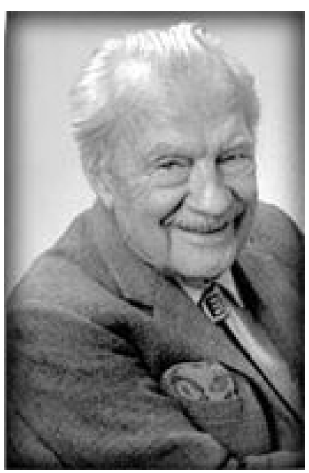
Neumann, John von
(1903-1957)

Monte Carlo

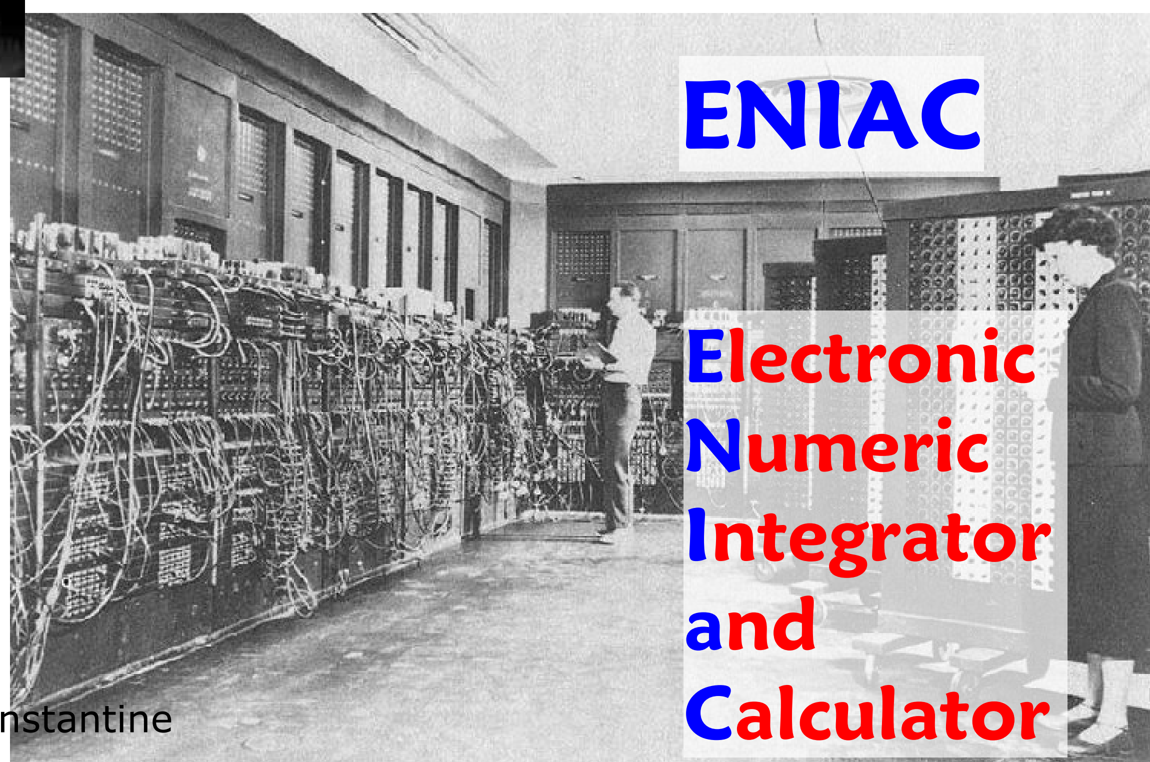
This wonderful method is indeed named after:

Ulam, Stanislaw
(1909-1986)

Statistical problems were pushing the construction of the first computers!!



Metropolis, Nicholas Constantine
(1915-1999)



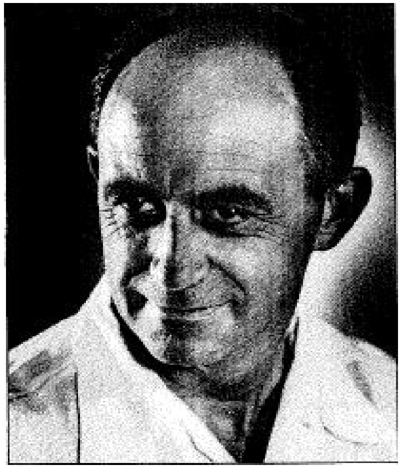
ENIAC

Electronic
Numeric
Integrator
and
Calculator

Neumann, John von
(1903-1957)

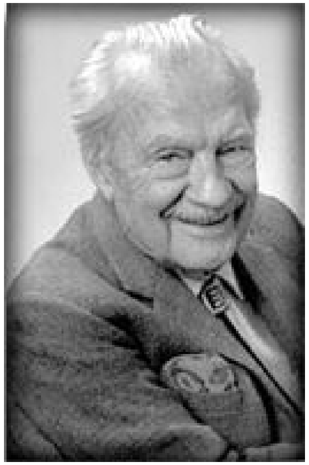
Monte Carlo

Ulam, Stanislaw
(1909-1986)

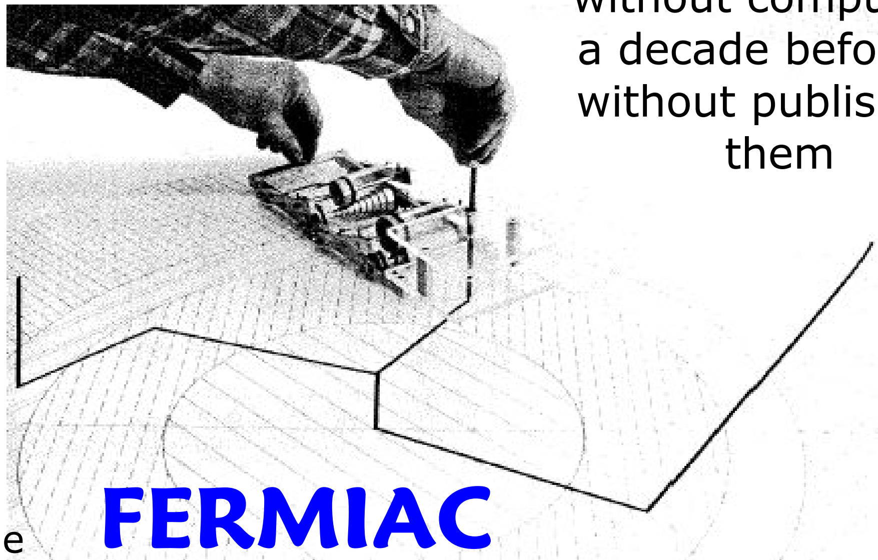


Fermi, Enrico
(1901-1954)

Fermi used MC methods without computers a decade before ... without publishing them



Metropolis, Nicholas Constantine
(1915-1999)



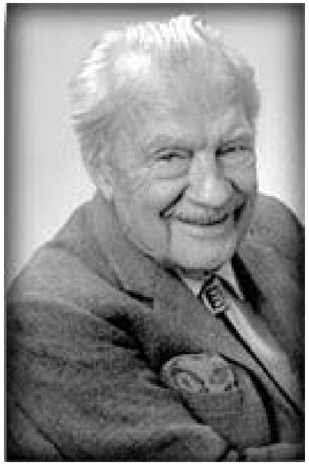
FERMIAC

purely mechanical!

Neumann, John von
(1903-1957)

Monte Carlo

Ulam, Stanislaw
(1909-1986)



Metropolis, Nicholas Constantine
(1915-1999)

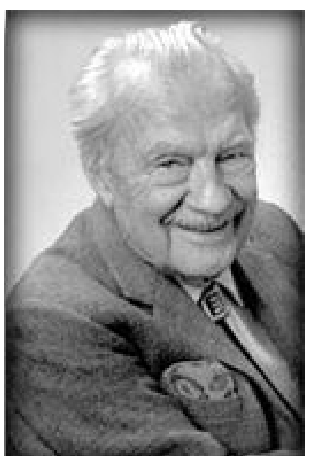
"Anyone who considers arithmetic methods of producing random digits is, of course, in a state of sin."

Neumann, John von
(1903-1957)

von Neumann contributed a lot to the mathematical foundations of MC.

Monte Carlo

Ulam, Stanislaw
(1909-1986)



Metropolis, Nicholas Constantine
(1915-1999)

sin."

Neumann, John von
(1903-1957)

von Neumann contributed a lot to
the mathematical foundations of MC.

Letter: von Neuman to Ulam, May 1947

... will follow today.

I am very glad that preparations for the random numbers work are to begin soon. In this connection, I would like to mention this: Assume that you have several random number distributions, each equidistributed in $0, 1$: (x^i), (y^i), (z^i), ... Assume that you want one with the distribution function (density) $f(\xi)\,d\xi$: (ξ^i). One way to form it is to form the cumulative distribution function: $g(\xi) = \int^{\xi} f(\xi)\,d\xi$ to invert it $h(x) = \xi \rightleftarrows x = g(\xi)$, and to form $\xi^i = h(x^i)$ with this $h(x)$, or some approximant polynomial. This is, as I see, the method that you have in mind.

An alternative, which works if ξ and all values of $f(\xi)$ lie in $0, 1$, is this: Scan pairs x^i, y^i and use or reject x^i, y^i according to whether $y^i \leq f(x^i)$ or not. In the first case, put $\xi^j = x^i$ in the second case form no ξ^j at that step.

The ...

Letter: von Neuman to Ulam, May 1947

Usage of the inverse-method

... will follow today.

I am very glad that preparations for the random numbers work are to begin soon. In this connection, I would like to mention this: Assume that you have several random number distributions, each equidistributed in $0, 1$: (x^i), (y^i), (z^i), ... Assume that you want one with the distribution function (density) $f(\xi)\,d\xi$: (ξ^i). One way to form it is to form the cumulative distribution function: $g(\xi) = \int^{\xi} f(\xi)\,d\xi$ to invert it $h(x) = \xi \rightleftarrows x = g(\xi)$, and to form $\xi^i = h(x^i)$ with this $h(x)$, or some approximant polynomial. This is, as I see, the method that you have in mind.

An alternative, which works if ξ and all values of $f(\xi)$ lie in 0, 1, is this: Scan pairs x^i, y^i and use or reject x^i, y^i according to whether $y^i \leq f(x^i)$ or not. In the first case, put $\xi^j = x^i$, in the second case form no ξ^j at that step.

The ...

Invention of the accept-reject method!

Monte Carlo

This wonderful method in words:

The spirit of Monte Carlo is best conveyed by the example discussed in von Neumann's letter to Richtmyer. Consider a spherical core of fissionable material surrounded by a shell of tamper material. Assume some initial distribution of neutrons in space and in velocity but ignore radiative and hydrodynamic effects. The idea is to now follow the development of a large number of individual neutron chains as a consequence of scattering, absorption, fission, and escape.

At each stage a sequence of decisions has to be made based on statistical probabilities appropriate to the physical and geometric factors. The first two decisions occur at time $t = 0$, when a neutron is selected to have a certain velocity and a certain spatial position. The next decisions are the position of the first collision and the nature of that collision. If it is determined that a fission occurs, the number of emerging neutrons must be decided upon, and each of these neutrons is eventually followed in the same fashion as the first. If the collision is decreed to be a scattering, appropriate statistics are invoked to determine the new momentum of the neutron. When the neutron crosses a material boundary, the parameters and characteristics of the new medium are taken into account. Thus, a genealogical history of an individual neutron is developed. The process is repeated for other neutrons until a statistically valid picture is generated.

from N.Metropolis: The Beginning of the Monte Carlo Method, Los Alamos Science, Special Issue 1987

Monte Carlo

This wonderful method in words:

The spirit of Monte Carlo is best conveyed by the example discussed in von Neumann's letter to Richtmyer. Consider a spherical core of fissionable material surrounded by a shell of tamper material. Assume some initial distribution of neutrons in space and in velocity but ignore radiative and hydrodynamic effects. The idea is to now follow the development of a large number of individual neutron chains as a consequence of scattering, absorption, fission, and escape.

At each stage a sequence of decisions has to be made based on statistical probabilities appropriate to the physical and geometric factors. The first two decisions occur at time $t = 0$, when a neutron is selected to have a certain velocity and a certain spatial position. The next decisions are the position of the first collision and the nature of that collision. If it is determined that a fission occurs, the number of emerging neutrons must be decided upon, and each of these neutrons is eventually followed in the same fashion as the first. If the collision is decreed to be a scattering, appropriate statistics are invoked to determine the new momentum of the neutron. When the neutron crosses a material boundary, the parameters and characteristics of the new medium are taken into account. Thus, a genealogical history of an individual neutron is developed. The process is repeated for other neutrons until a statistically valid picture is generated.

from N.Metropolis: The Beginning of the Monte Carlo Method, Los Alamos Science, Special Issue 1987

Monte Carlo

This wonderful method in words:

Geometry, Materials

Event Generator

Tracking

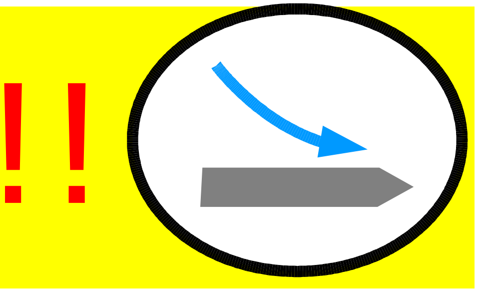
Physics Processes

The spirit of Monte Carlo is best conveyed by the example discussed in von Neumann's letter to Richtmyer. Consider a spherical core of fissionable material surrounded by a shell of tamper material. Assume some initial distribution of neutrons in space and in velocity but ignore radiative and hydrodynamic effects. The idea is to now follow the development of a large number of individual neutron chains as a consequence of scattering, absorption, fission, and escape.

At each stage a sequence of decisions has to be made based on statistical probabilities appropriate to the physical and geometric factors. The first two decisions occur at time $t = 0$, when a neutron is selected to have a certain velocity and a certain spatial position. The next decisions are the position of the first collision and the nature of that collision. If it is determined that a fission occurs, the number of emerging neutrons must be decided upon, and each of these neutrons is eventually followed in the same fashion as the first. If the collision is decreed to be a scattering, appropriate statistics are invoked to determine the new momentum of the neutron. When the neutron crosses a material boundary, the parameters and characteristics of the new medium are taken into account. Thus, a genealogical history of an individual neutron is developed. The process is repeated for other neutrons until a statistically valid picture is generated.

from N.Metropolis: The Beginning of the Monte Carlo Method, Los Alamos Science, Special Issue 1987

So, what then is GEANT4?



"I want you to find a bold and innovative way to do everything exactly the same way it's been done for ~~25~~ years."

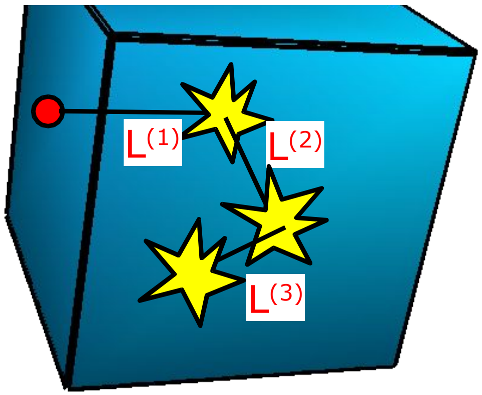
~60 years!

Monte Carlo in GEANT4

- Talk about C++ classes of GEANT4
  - implementing the concepts introduced until now
  - introduction to GEANT4 from the physics point of view
  - will talk later about the other important G4 features
- It´s also a summary of what we have heard so far
  - particles traversing bulk material
  - tracking free path length, physics processes
  - this lecture: primary particles and materials in G4
  - “all” the rest in the following lectures
- Show you puzzle pieces of G4
  - later: show you, how we put them together to get the “big picture”

Primary Particles

Remember?



Very basic algorithm:

(1) start values for incident particle
(2) get values for ρ and σ
(3) sample $L^{(n)}$ from $p(L)$
(4) transport particle undisturbed by $L^{(n)}$
(5) simulate interaction
(6) if particle still exists: goto (1)

We have to tell G4 the starting conditions of all the particles we want to track through the material!

Need these classes:

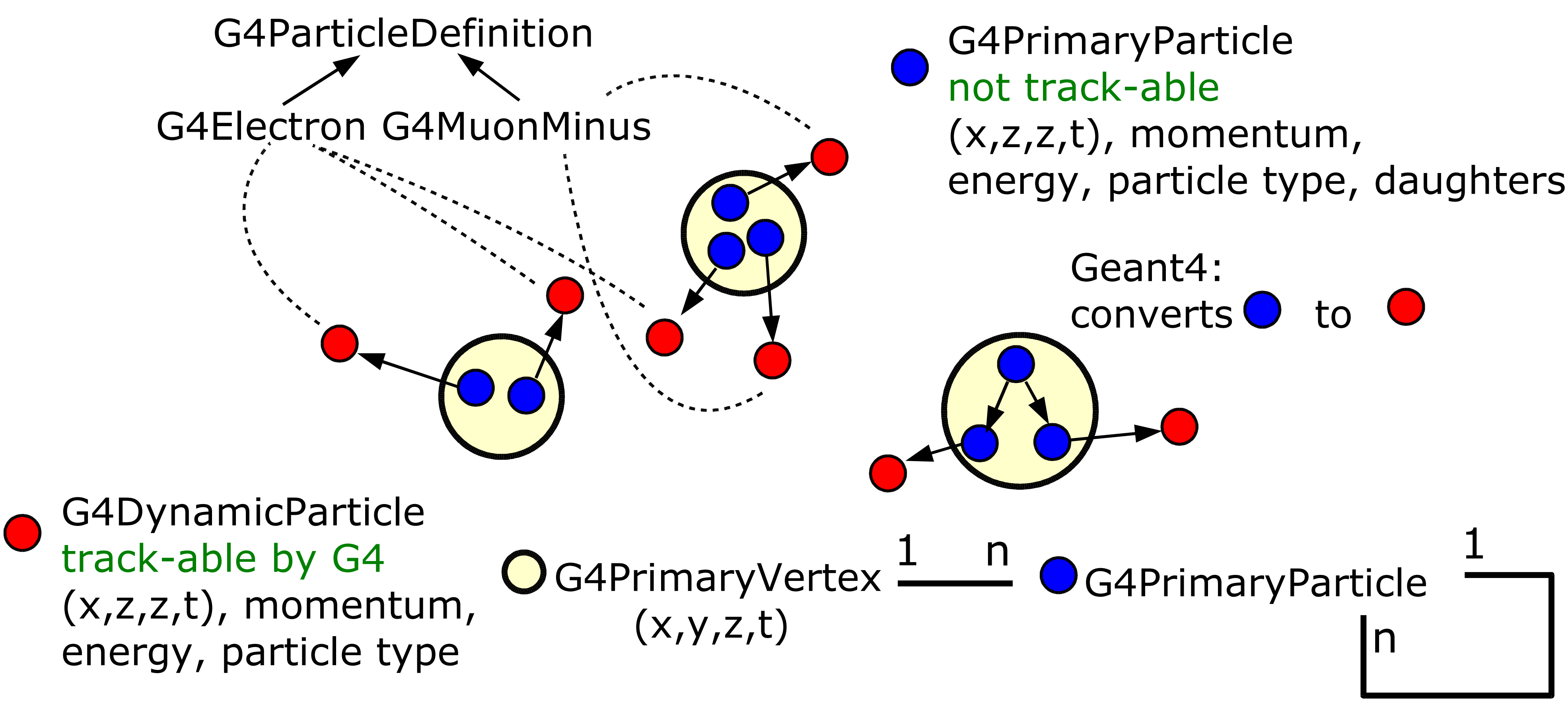
G4ParticleDefinition and subclasses
|
particle type such as e^-, μ^+ and its static data such as charge, mass, spin, ..

G4PrimaryParticle
|
momentum, kinetic energy of a particle described by

G4PrimaryVertex
|
position and time of a list of

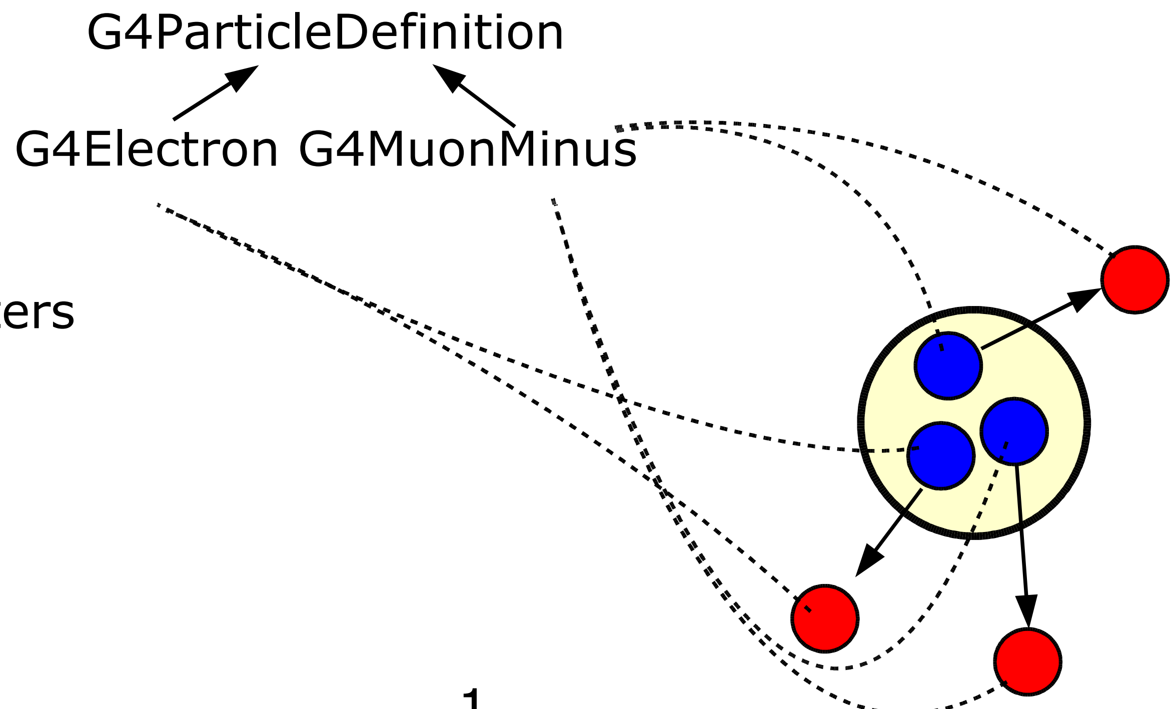
Primary Particles

When starting the simulation, G4 takes all the information of the instances of G4PrimaryVertex, G4PrimaryParticle, and G4ParticleDefinition, and creates G4DynamicPartice instances. These correspond to the particles being tracked.



Dynamic Particles

A G4DynamicPartice contains the current, dynamic state of the particle during a simulation: position, time, momentum, energy; it also has a pointer to the particle type information modeled by a subclass of G4ParticleDefintion, e.g. G4Electron



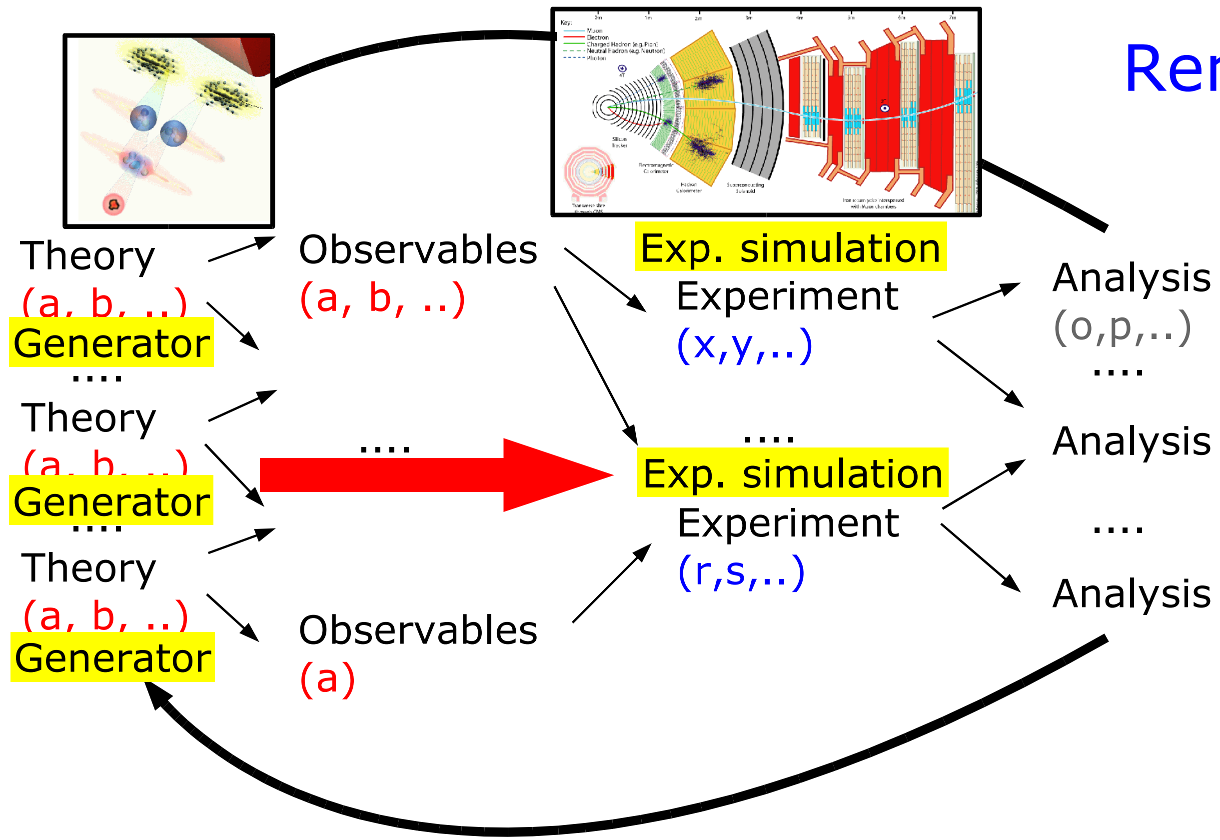
G4PrimaryParticle
not track-able
(x,z,z,t), momentum,
energy, particle type, daughters

G4DynamicParticle
track-able by G4
(x,z,z,t), momentum,
energy, particle type

G4PrimaryVertex (x,y,z,t) 1 — n G4PrimaryParticle 1 — n

.. and Event Generators

Remember?

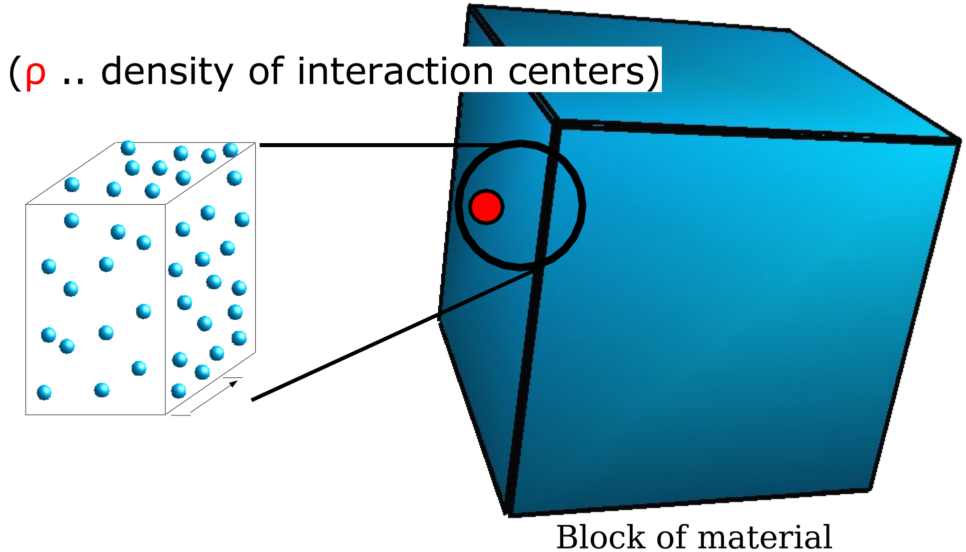


Use G4ParticleDefintion, -PrimaryParticle, -PrimaryVertex to transform the output of Event Generator into the GEANT4 world!

Materials

You have to describe the properties of bulk materials to G4!

Remember?



G4 provides you a quite comfortable way to describe materials like "in the textbook"

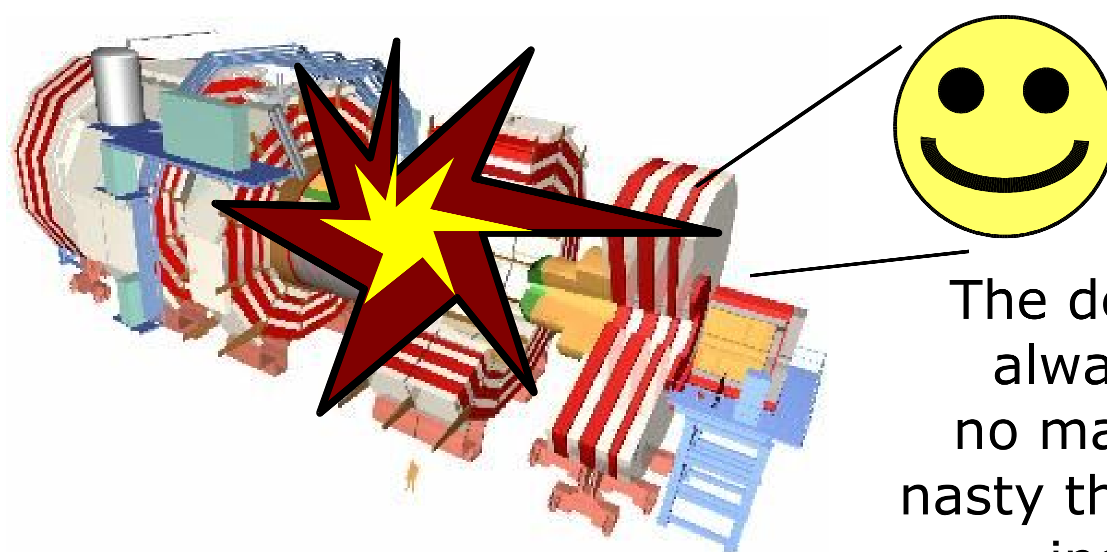
What are the main features of a material in G4?

Properties of Materials in G4

- Materials are not affected by time
  - Gas does not diffuse, liquids don't flow
  - No thermodynamics: no heat transfer, no state changes (water <-> ice), ...

Properties of Materials in G4

- Materials are not affected by time
  - Gas does not diffuse, liquids don't flow
  - No thermodynamics: no heat transfer, no state changes (water <-> ice), ...
- Particle interaction does not harm!
  - No irradiation effects or radiation damages



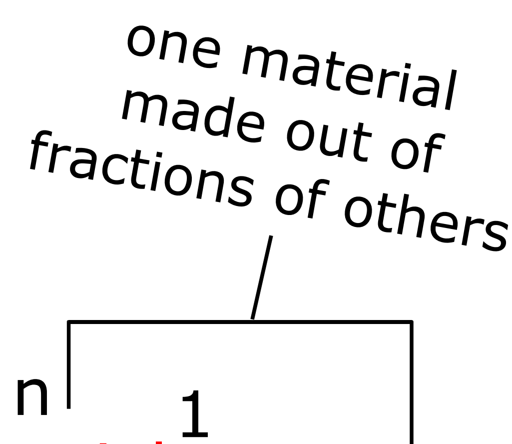
The detector will
always smile,
no matter which
nasty things happen
inside ;-)

Properties of Materials in G4

- Materials are not affected by time
  - Gas does not diffuse, liquids don't flow
  - No thermodynamics: no heat transfer, no state changes (water <-> ice), ...
- Particle interaction does not harm!
  - No irradiation effects or radiation damages
- Not affected by external influences. Materials don't change physical properties under
  - Mechanical influences (stresses, pressures)
  - External electromagnetic fields, i.e. materials don't get magnetized, electrostatically charged (-> see later: description of external fields)

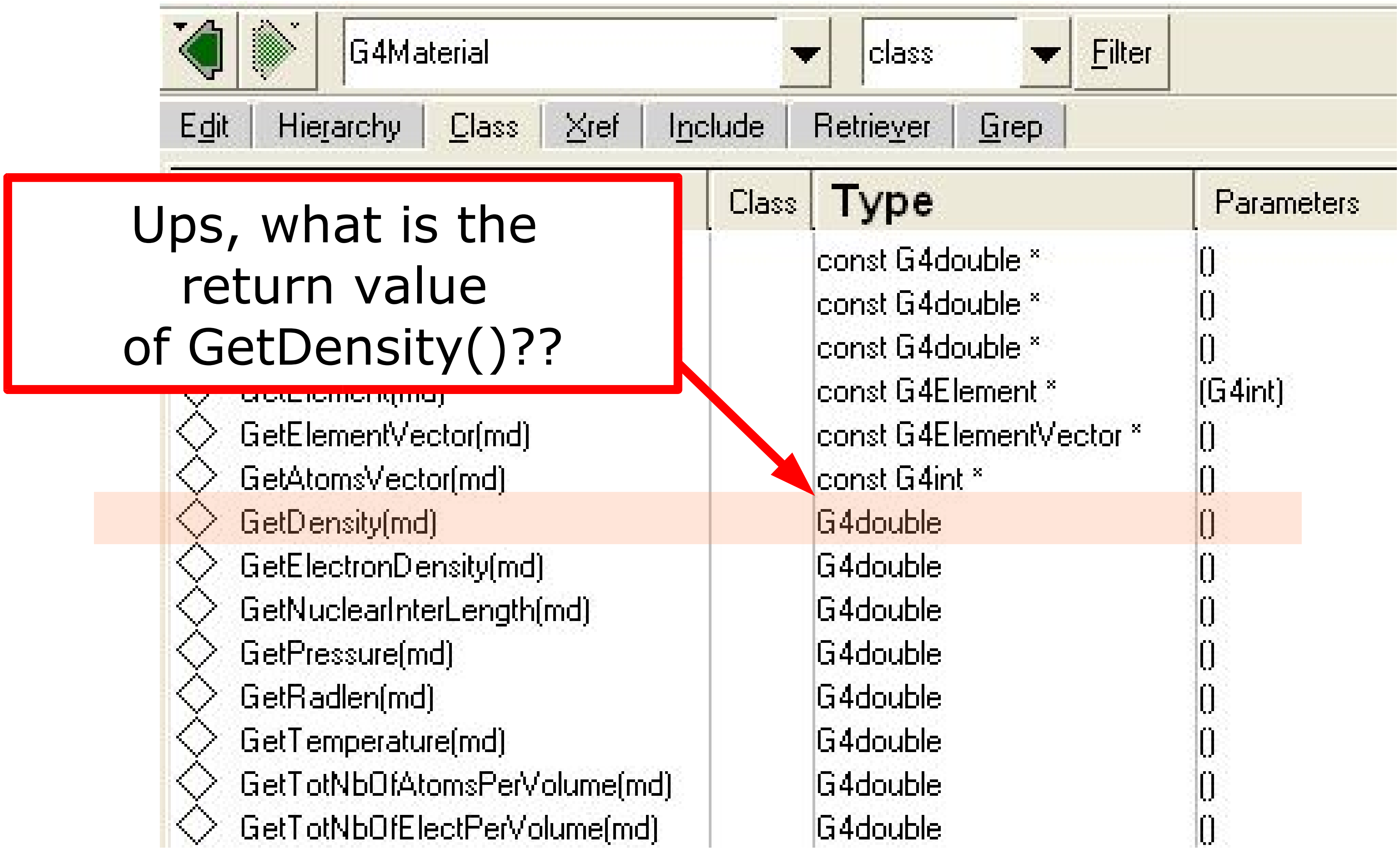
So, what are G4 Materials?

- All materials are represented as objects of a single class: G4Material
- Objects of G4Material is a concrete class that
  - are the only instances used by G4 to ask about material properties
  - represent a homogeneous material made of elements, compounds, or mixtures of other materials
- There are several ways to define a material in a user-friendly way through helper classes;
  - isotopes <-> G4Isotope
  - elements <-> G4Element
  - molecules <-> G4Material
  - compounds and mixtures <-> G4Material



class G4Material

excerpt from the public interface of G4Material:



A word on Quantities & Units

"For those who want some proof that physicists are human, the proof is in the idiocy of all the different units which they use for measuring energy."

The Character of Physical Law (1967)
R.P.Feynman

Quantities & Units in G4

Code:

```
double energy = 1.*GeV;
cout << "Energy of 1 GeV:" << endl;
cout << " in ev     : " << energy/eV << endl
     << " in Joule: " << energy/joule << endl;

double h = 60.*60.*s;
double velocity = 20.*km/h;
cout << "Velocity of 20 km/h:" << endl
     << "  internal : " << velocity << endl
     << "  km/h     : " << velocity/(km/h) << endl
     << "  m/sec    : " << velocity/(m/s) << endl;
```

Result:

```
Energy of 1 GeV:
 in ev    : 1e+09
 in Joule: 1.60218e-10

Velocity of 20 km/h:
  internal : 5.55556e-06
  km/h     : 20
  m/sec    : 5.55556
```

Simple Rules:

- all quantities are represented by a double.
- to pass/create a physical quantity, multiply the numerical value by its unit
- to read/use a physical unit, divide the numerical value by the unit you want to have

Quick Summary

- Mapping of concepts ...
  - incident particles – primary particles
  - bulk material
- ... to GEANT4 class implementations
  - G4ParticleDefinition, G4PrimaryVertex, G4PrimaryParticle, G4DynamicParticle
  - G4Material & helpers
- Easy convention on physical units & quantities
  - different quantities not differently typed in C++, all is "double"
  - multiply / divide by predefined constants

Summary -2-

- Passage of particles through bulk matter
  - cross section, free path length
  - exponential distribution
  - multiple physics processes
- Monte Carlo method
  - random numbers & sampling from distributions
  - basic tracking algorithm
    - for one active physical process
    - for multiple physical processes
  - a bit of history – the bomb
- Gentle introduction to GEANT4
  - primary particles
  - materials
  - physical units & quantities