This course

Structure and logistics - 1

- This course is organized in a mixture of theoretical lectures and practical hands-on sessions
  - The hands-on sessions require real C++ coding to build up a simplified Geant4 application
  - Staged approach in tasks
  - **http://202.122.35.42/introduction**
- A pre-installed virtual machine is provided for the hands-on sessions
  - Includes **Geant4 10.5** on a Linux environment
  - You should already have it downloaded and tested
    - Please let us know ASAP if you have problems with the VM

Structure and logistics - 2

- You can try to install Geant4 on your (Linux/Mac) laptop, if you wish
  - The course is not meant to show that, though
- All lectures (pdf) will be uploaded on-the-fly on the course indico page
  - `http://indico.ihep.ac.cn/event/9624/`
- Please **feel free to ask any question**, either during the lectures , during the exercises or during the breaks
- Solutions of the exercises will be uploaded after the end of each exercise session

Monte Carlo techniques and Geant4 concept

Luciano Pandola

INFN – Laboratori Nazionali del Sud

The 2nd Geant4 School in China,
Shandong University, Qingdao,
March 25th- 29th, 2019

Part I: Monte Carlo and particle tracking

What Monte Carlo (MC) techniques are for?

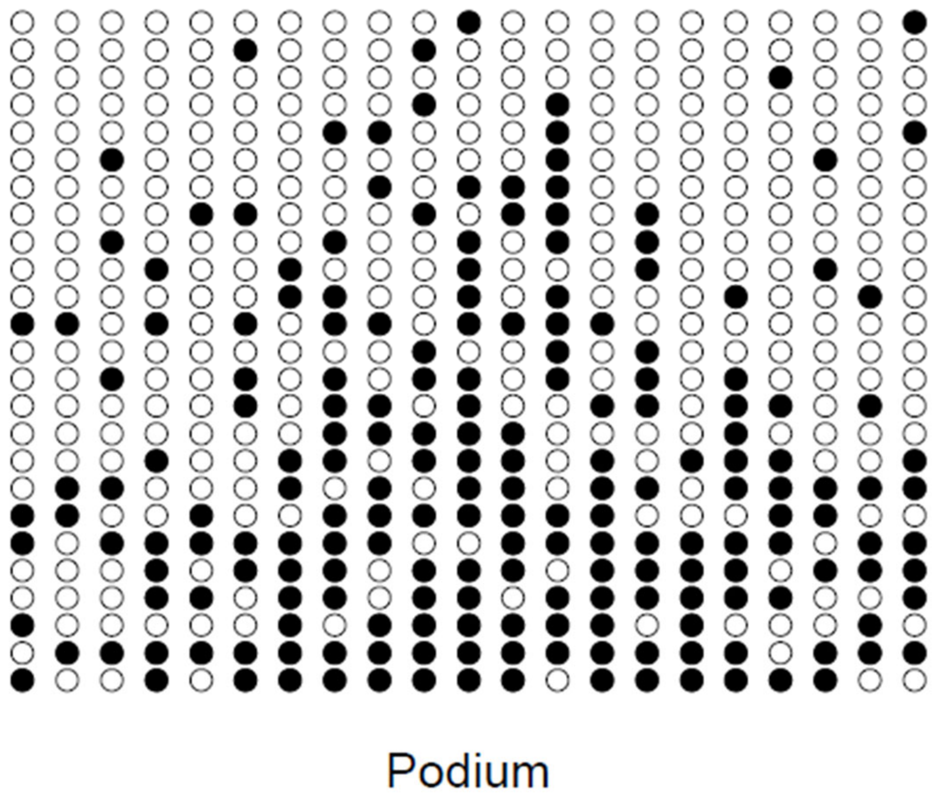
- Numerical solution of a (**complex**) macroscopic problem, by simulating the microscopic interactions among the components
- Uses random sampling, until convergence is achieved
  - Name after Monte Carlo's casino
- Applications not only in physics and science, but also finances, traffic flow, social studies
  - And not only problems that are intrisically probabilistic (e.g. numerical integration)

An example: arrangement in an auditorium

- Produce a configuration (or a "final state"), according to some "laws", e.g.
  - People mostly arrive in pairs
  - Audience members prefer an un-obstructed view of the stage
  - Audience members prefer seats in the middle, and close to the front row
  - Only one person can occupy a seat
- Contrarily e.g. to physics, the laws are not known
  - Rather use "*working assumptions*"
- The math (exact) formulation can be impossible or unpratical → MC is more effective

An example: arrangement in an auditorium

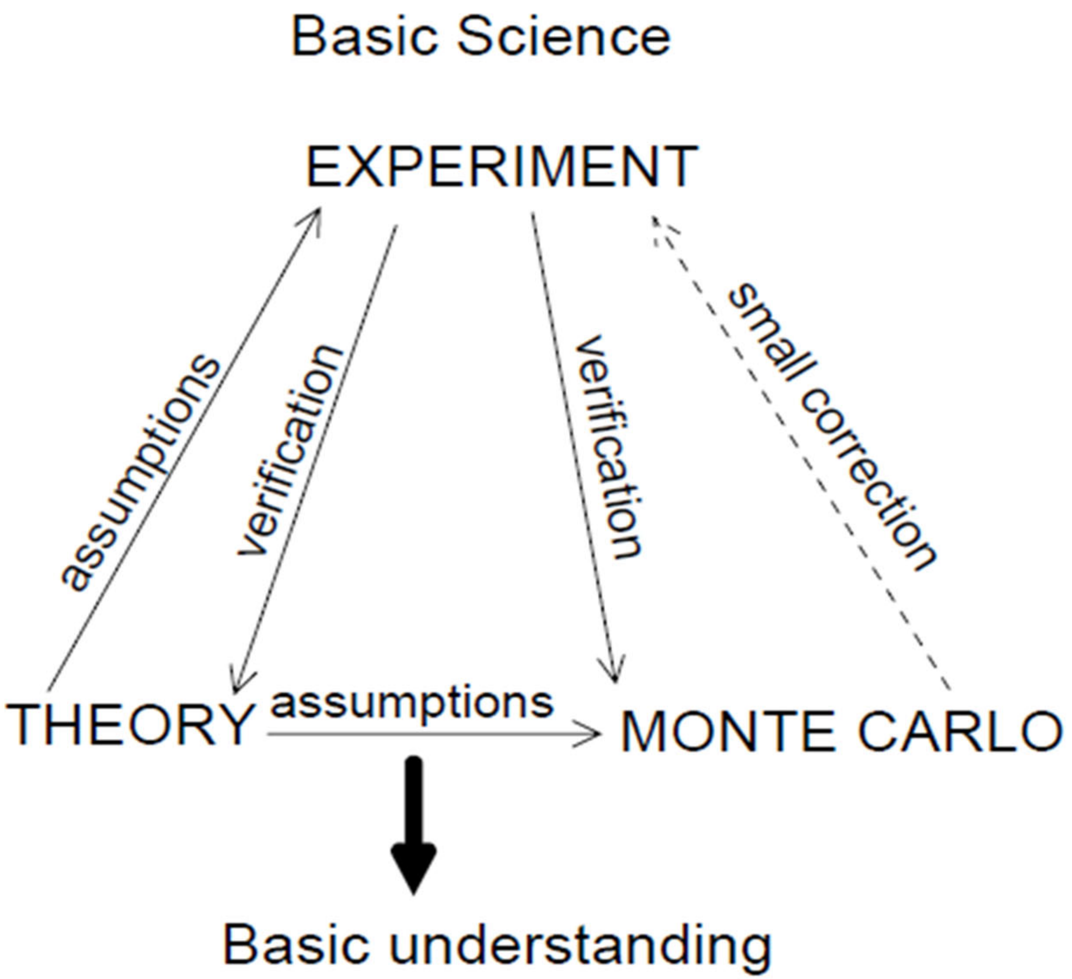
- Reverse logic: find the "laws" that better fit the **observed** distribution
  - Use MC to build a (microscopic) theory of a complex system by comparison with experiments



MC in science

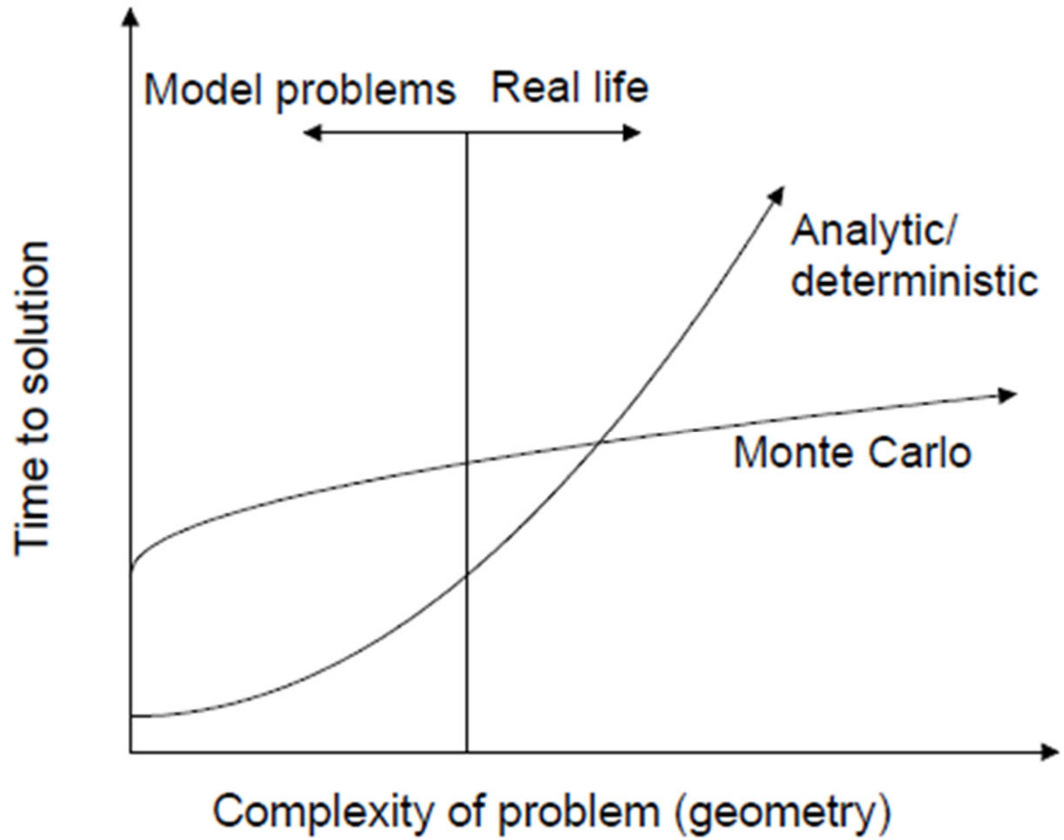
- In physics, elementary laws are (typically) **known** → MC is used to predict the outcome of a (complex) experiment
  - Exact calculation from the basic laws is unpractical
  - Optimize an experimental setup, support data analysis
- Can be used to validate/disproof a theory, and/or to provide small corrections to the theory
- In this course: Monte Carlo for particle tracking (interaction of radiation with matter)

Interplay between theory, simulation and experiments



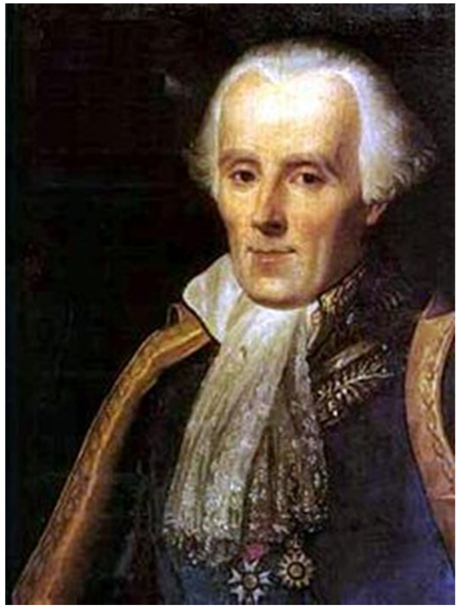
When are MC useful wrt to the math exact solution?

- Usually the Monte Carlo wins over the exact (mathematical) solution for **complex** problems



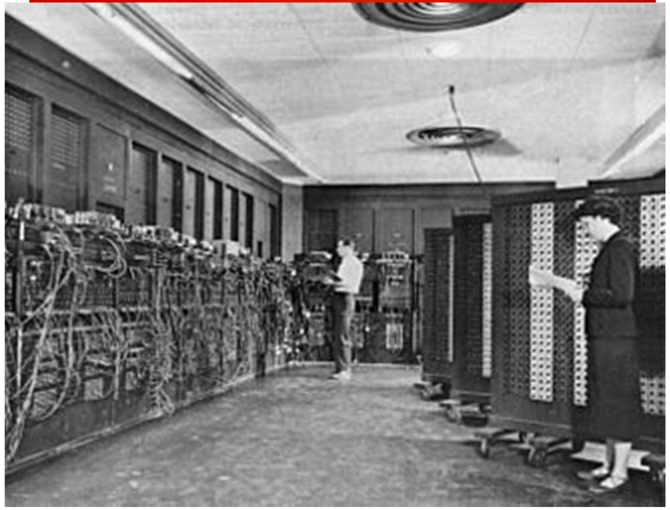
A bit of history

- Very concept of Monte Carlo comes in the XVIII century (Buffon, 1777, and then Laplace, 1786)
  - Monte Carlo estimate of π
- Concept of MC is much older than real computers
  - one can also implement the algorithms manually, with dice (= Random Number Generator)



A bit of history

- Boost in the '50 (Ulam and Von Neumann) for the development of thermonuclear weapons
- Von Neumann invented the name "Monte Carlo" and settled a number of basic theorems
- First (proto)computers available at that time
  - MC mainly **CPU load**, minimal I/O



A bit of history

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION

Number 247 SEPTEMBER 1949 *Volume 44*

THE MONTE CARLO METHOD

Nicholas Metropolis and S. Ulam
Los Alamos Laboratory

THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

Equation of State Calculations by Fast Computing Machines

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, and Augusta H. Teller,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

and

Edward Teller,* *Department of Physics, University of Chicago, Chicago, Illinois*

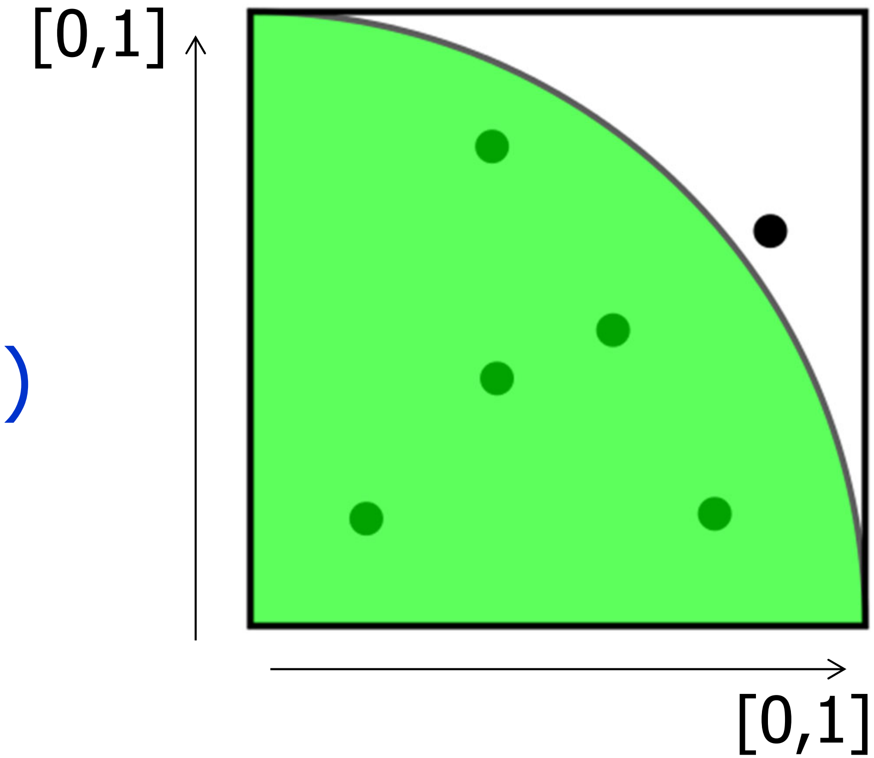
(Received March 6, 1953)

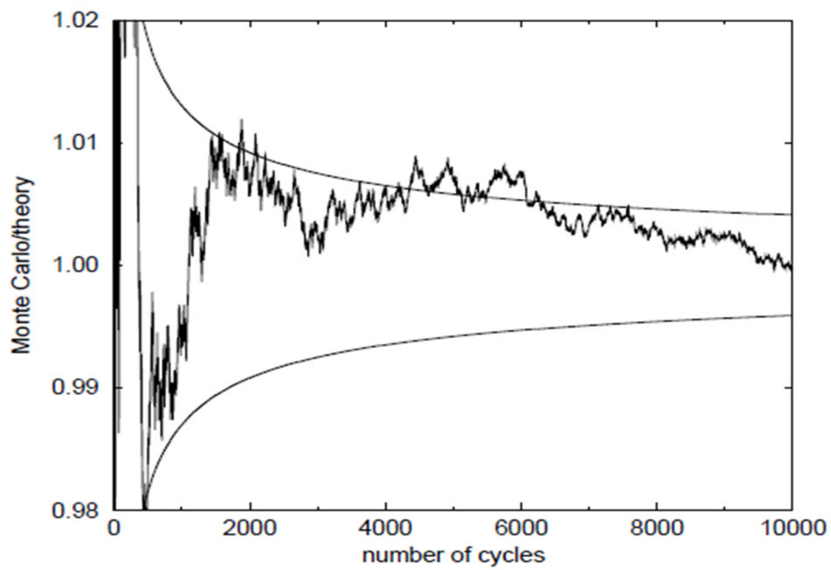
Nick Metropolis enjoying a break in the quantum Monte Carlo conference, September 1985.

With MANIAC: the first electronic digital computer

The simplest MC application: numerical estimate of π

- Shoot N couples (x,y) randomly in [0,1]
- Count n: how many couples satisfy $(x^2+y^2\leq 1)$

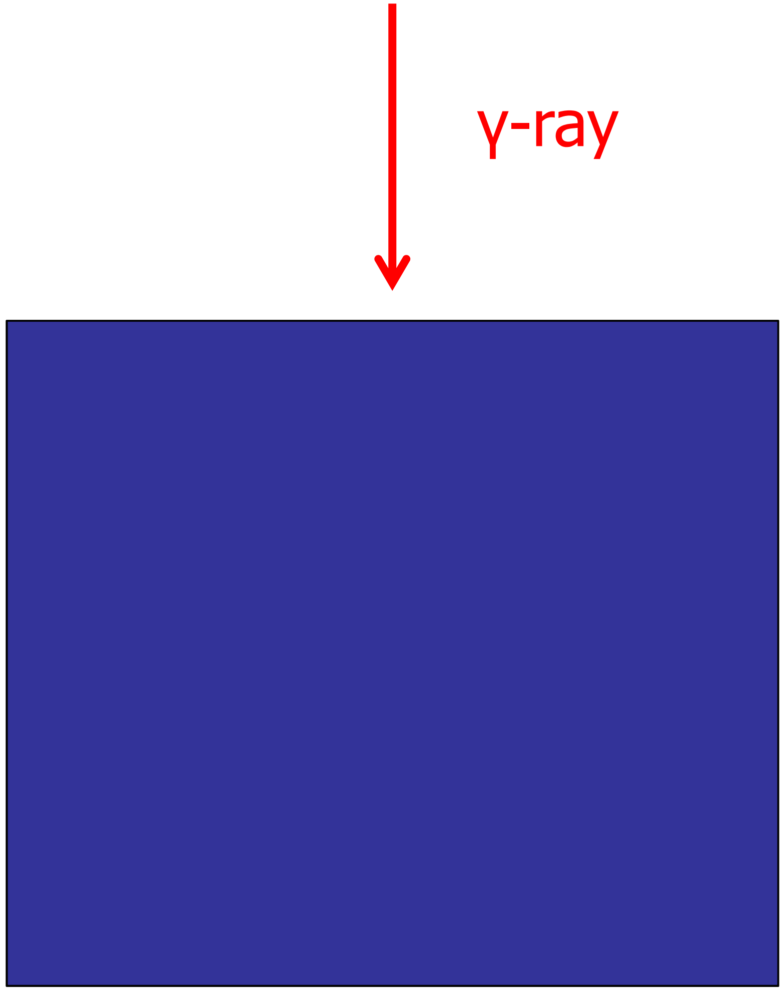




- $n/N = \pi/4$ (ratio of areas)
- Convergence as $1/\sqrt{N}$

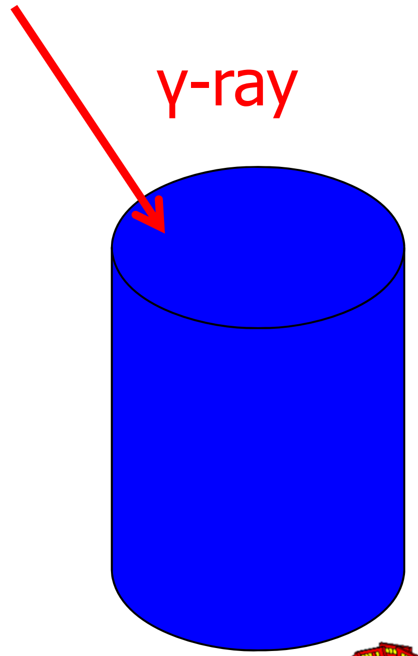
Most common application in particle physics: particle tracking

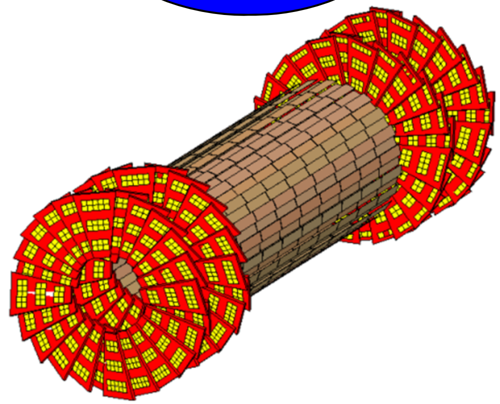
- Problem: track a γ-ray in a semi-infinite detector and determine the energy spectrum deposited
  - Still, a **model case**
- All physics is known from textbook (Compton scattering, photoelectric effect, etc.)
- Yet, the analytical calculation is a nightmare (while still possible)



Most common application in particle physics: particle tracking

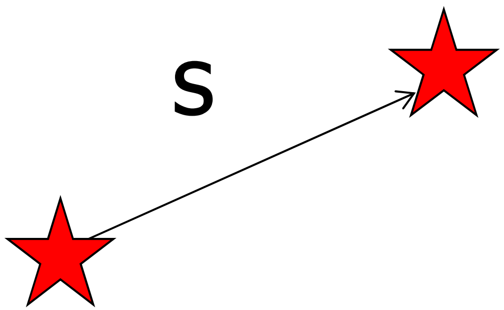
- Problem v2: track a γ-ray in a finite detector (e.g. a NaI)
  - Real-life (simplified) case
- Analytical computation nearly impossible
  - Monte Carlo clearly wins
- Now make the detector more complicate, as in modern physics





Particle tracking

- Distance s between two subsequent interactions distributed as $$p(s) = \mu e^{-\mu s}$$
- μ is a property of the *medium* (supposed to be homogeneous) and of the *physics*



- If medium is not homogeneous

$$p(s) = \mu(z)\exp(-\int_0^z \mu(z')dz')$$

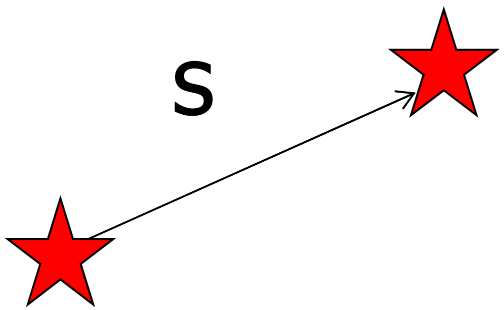
- Transition between two homogeneous materials

$$\mu(z) = \theta(b-z)\mu_1 + \theta(z-b)\mu_2$$

Particle tracking

- μ is proportional to the **total** cross section and depends on the density of the material

$$\mu = N\sigma = N\sum_i \sigma_i = \sum_i \mu_i$$

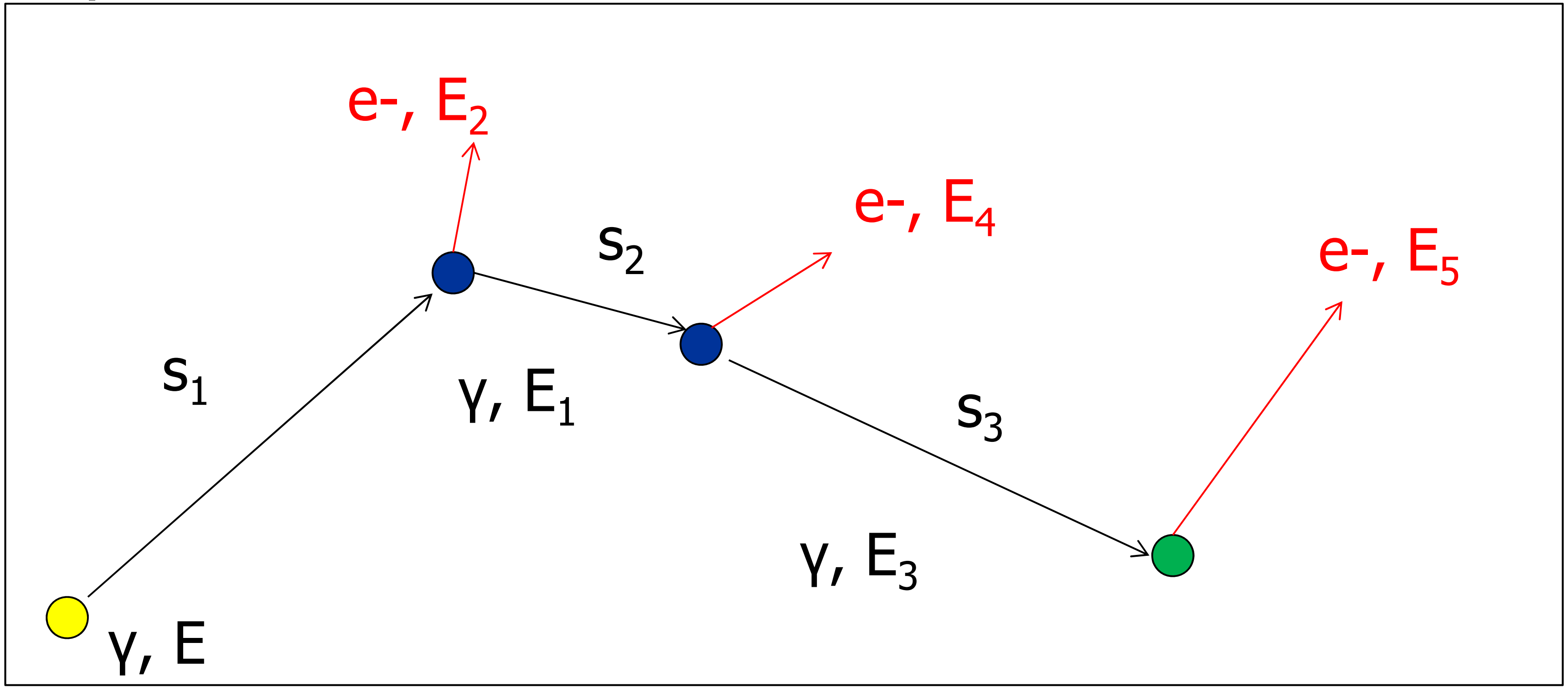


- All competing processes contribute with their own μ_i
- Each process takes place with probability μ_i/μ → i.e. proportionally to the partial cross sections

Particle tracking: basic recipe

- Divide the trajectory of the particle in **"steps"**
  - Straight free-flight tracks between consecutive physics interactions
  - Steps can also be limited by geometry boundaries
- Decide the step length s, by sampling according to $p(s) = \mu e^{-\mu s}$, with the proper μ (material+physics)
- Decide which interaction takes place at the end of the step, according to μ_i/μ
- Produce the final state according to the **physics** of the interaction ($d^2\sigma/d\Omega dE$)
  - Update direction of the primary particle
  - Store somewhere the possible secondary particles, to be tracked later on

Particle tracking: basic recipe



- Follow **all secondaries**, until absorbed or leave volume
- Notice: μ depends on energy (cross sections do!)

Well, not so easy

- This basic recipe works fine for γ-rays and other neutral particles (e.g. neutrons)
- Not so well for $e^{\pm}$: the cross section (ionization & bremsstrahlung) is very high, so the steps between two consecutive interactions are very small
  - CPU intensive: viable for low energies and thin material
- Even worse: in each interaction only a small fraction of energy is lost, and the angular displacement is small
  - A lot of time is spent to simulate interactions having **small effect**
  - The interactions of γ are "catastrophics": large change in energy/direction

Solution: the mixed Monte Carlo

- Simulate explicitly (i.e. force step) interactions **only if** energy loss (or change of direction) is above threshold W_0
  - Detailed simulation
  - "**hard**" interaction (like γ interactions)
- The effect of all sub-threshold interactions is described statistically
  - Condensed simulation
  - "**soft**" interactions
- Hard interactions occur much less frequently than soft interactions
  - Fully detailed simulation restored for $W_0=0$

The mixed Monte Carlo

- Has some technical tricks:
  - since energy is lost along the step due to soft interactions, the sampled step s cannot be too long ($s < s_{max}$)
- Parameter μ_h between hard collisions

$$\mu_h = N \int_{W_0}^{E} \frac{d\sigma}{dW}(E)\,dW$$

  - Has **μ_h << μ** because the differential cross section is strogly peaked at low W (= soft secondaries)
  - Much longer step length

The mixed Monte Carlo

- Stopping power due to **soft** collisions (dE/dx)

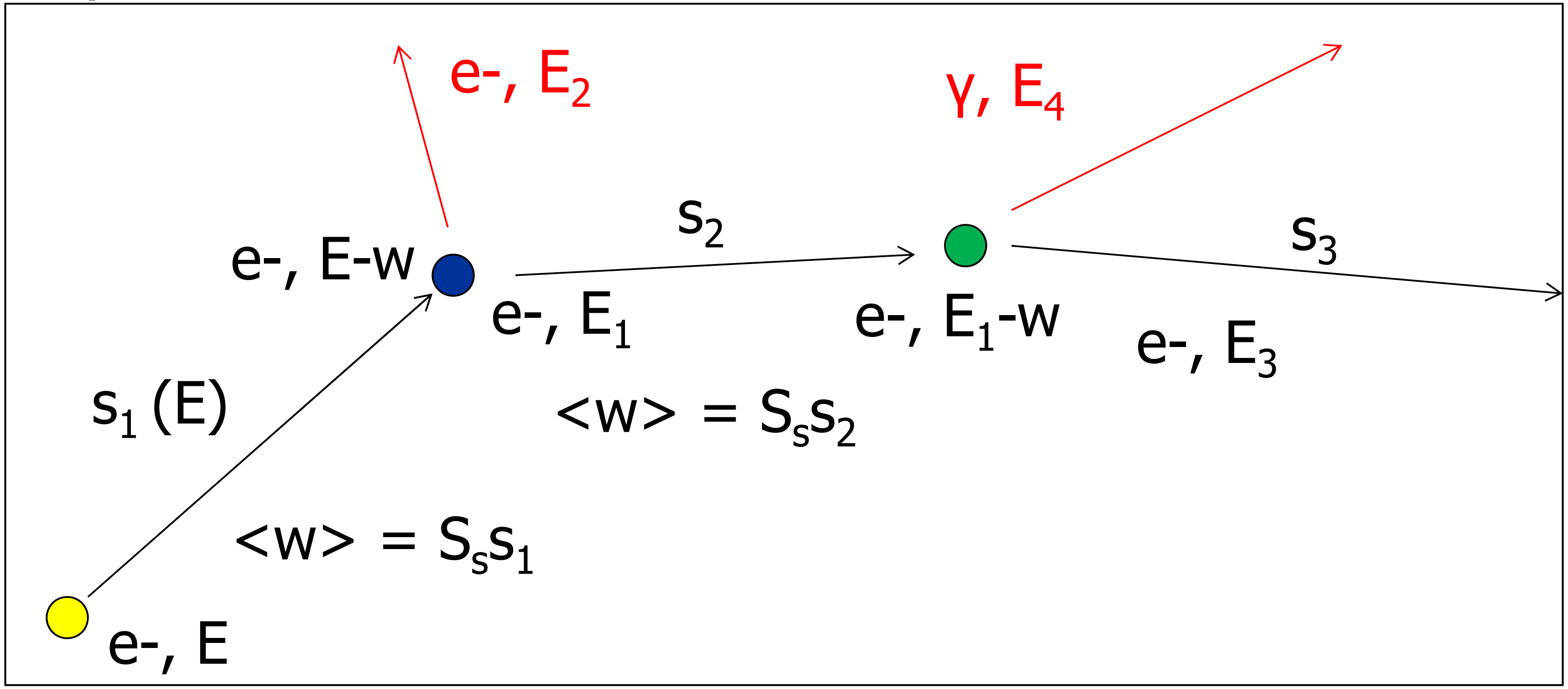
$$S_s = N \int_0^{W_0} W \frac{d\sigma}{dW}(E)\,dW$$

- Average energy lost along the step: <w>=sS_s
  - Must be <w> << E
- Fluctuations around the average value <w> have to be taken into account
  - Appropriate random sampling of w with mean value <w> and variance (straggling)

Extended recipe

1. Decide the step length s, by sampling according to $p(s)= \mu_h e^{-\mu_h s}$, with the proper μ_h
2. Calculate the cumulative effect of the **soft** interactions along the step: sample the energy loss w, with $\langle w \rangle = sS_s$, and the displacement
3. Update *energy and direction* of the primary particle at the end of the step $E \rightarrow E-w$
4. Decide which interaction takes place at the end of the step, according to $\mu_{i,h}/\mu_h$
5. Produce the final state according to the **physics** of the interaction ($d^2\sigma/d\Omega dE$)

Particle tracking: mixed recipe



- Follow **all secondaries**, until absorbed or leave volume

Geometry

- Geometry also enters into the tracking
  - A step can never cross a geometry boundary
  - **Always stop** the step when there is a boundary, then re-start in the new medium
- Navigation in the geometry can be CPU-intensive
  - One must know to which volume each point (x,y,z) belongs to, and how far (and in which direction) is the closest boundary
- Trajectories can be affected also by EM fields, for charged particles

…luckily enough, somebody else already implemented the tracking algorithms for us (and much more)