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://202.122.35.46/geant4/introduction`**
- A pre-installed virtual machine is provided for the hands-on sessions
  - Includes **Geant4 10.3.p01** 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/6764/`**
- 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

IHEP, China

What Monte Carlo (MC) techniques are for?

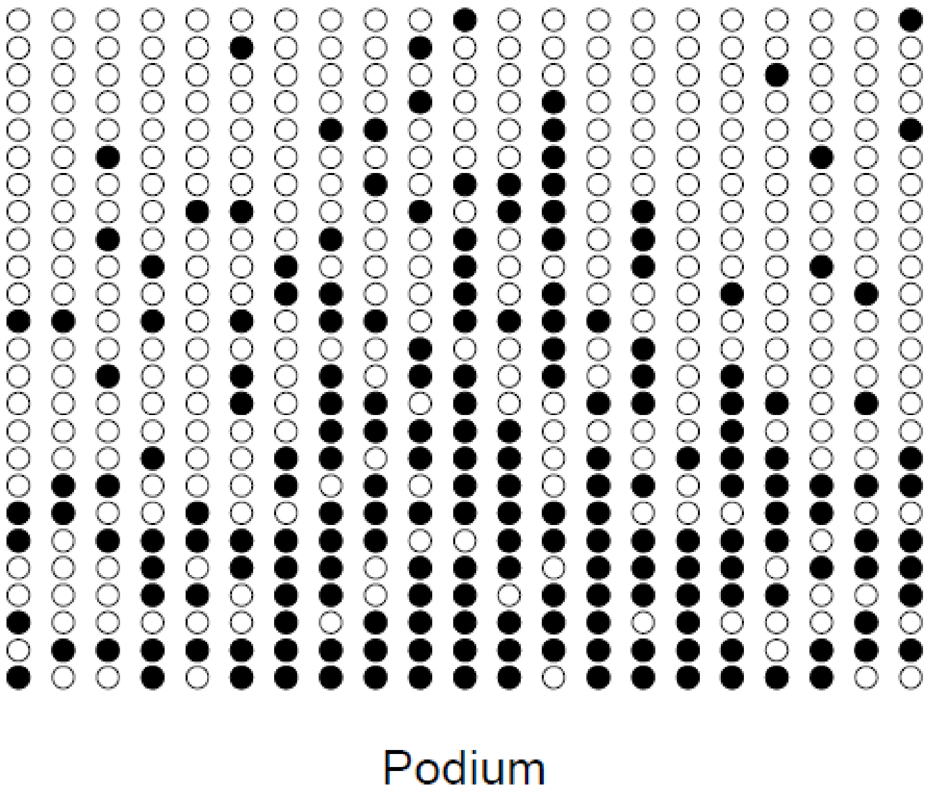
- Numerical solution of a (**complex**) macroscopic problem, by simulating the microscopic interactions among the components
- Uses random sampling, until convergence is achieved
  - Name after Monte Carlo's casino
- Applications not only in physics and science, but also finances, traffic flow, social 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

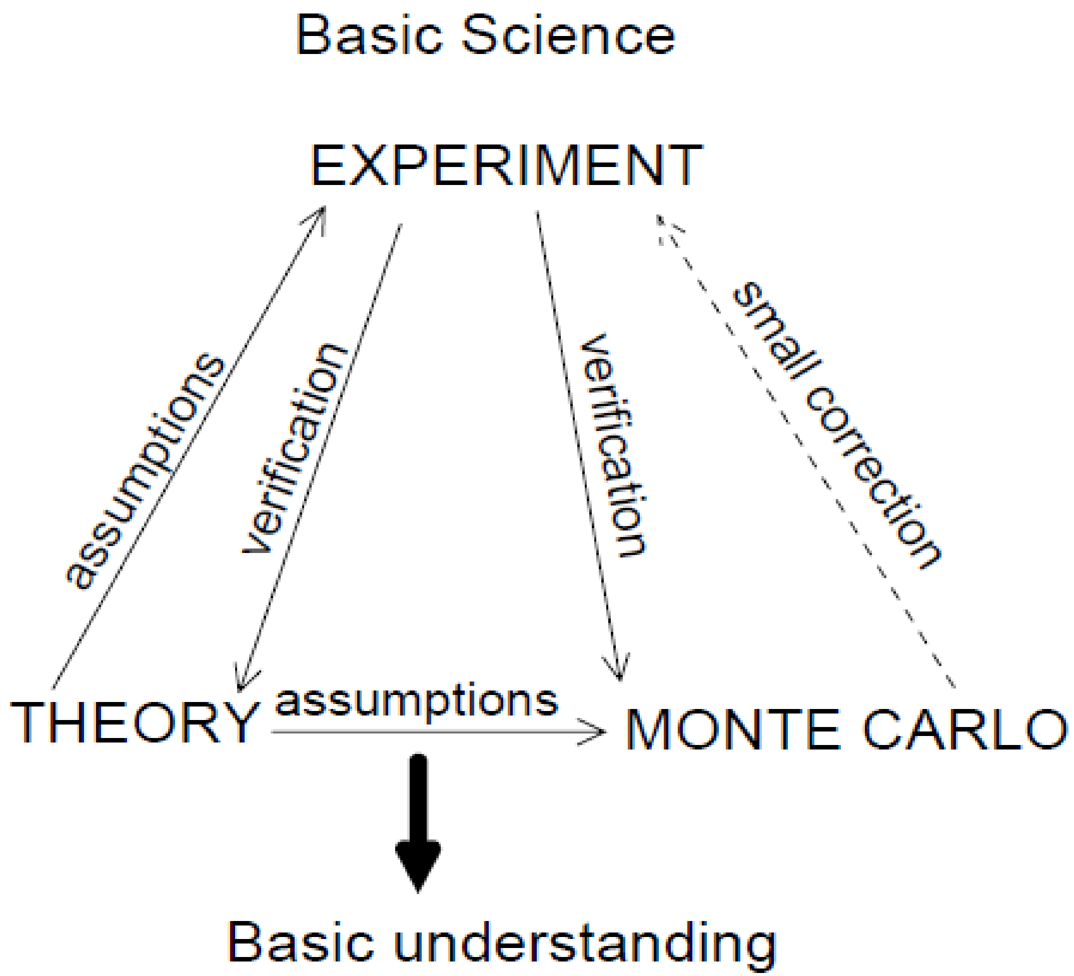
- Reverse logic: find the "laws" that better fit the **observed** distribution
  - Use MC to build a (microscopic) theory of a complex system by comparison with experiments



MC in science

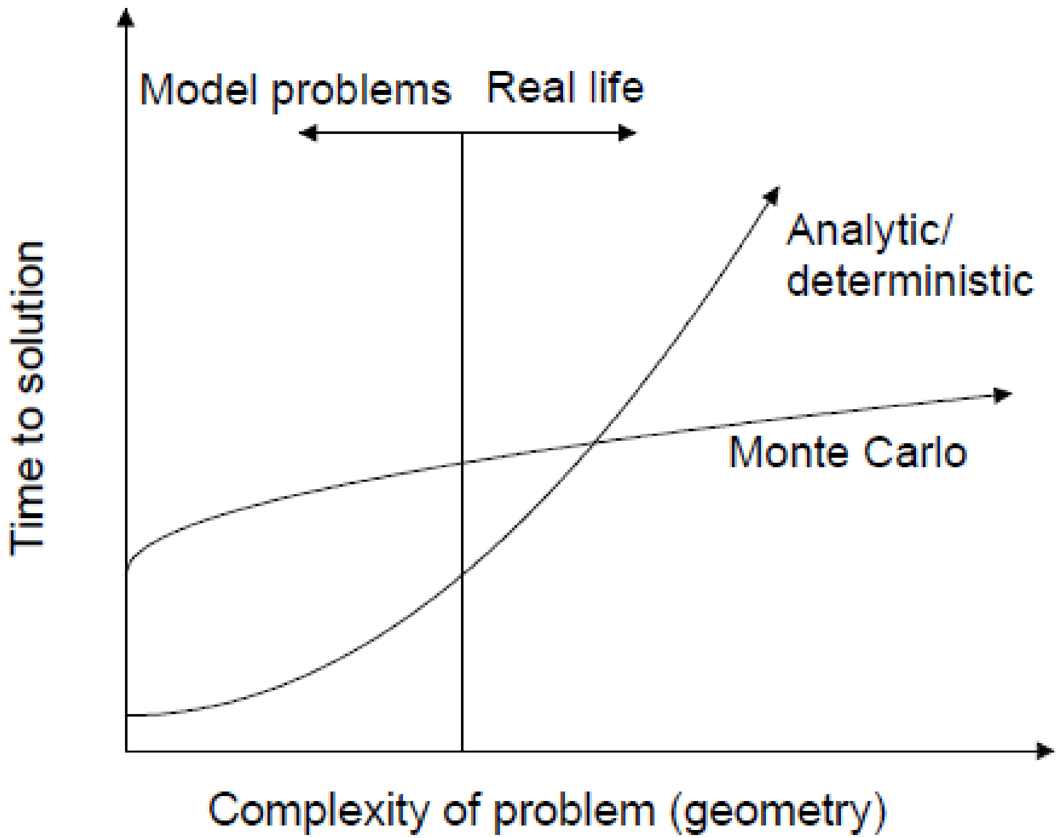
- In physics, elementary laws are (typically) **known** → MC is used to predict the outcome of a (complex) experiment
  - Exact calculation from the basic laws is unpractical
  - Optimize an experimental setup, support data analysis
- Can be used to validate/disproof a theory, and/or to provide small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

Interplay between theory, simulation and experiments



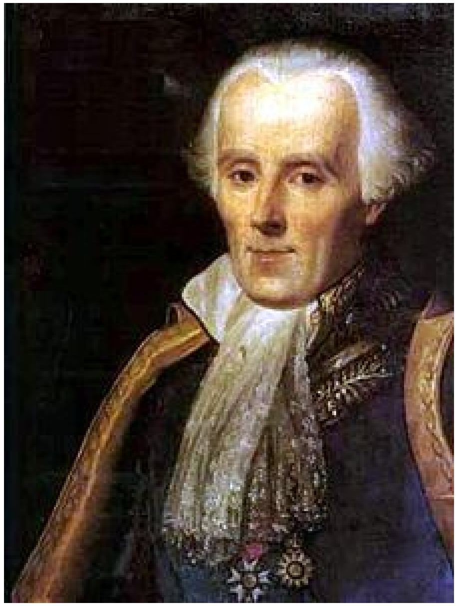
When are MC useful wrt to the math exact solution?

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



A bit of history

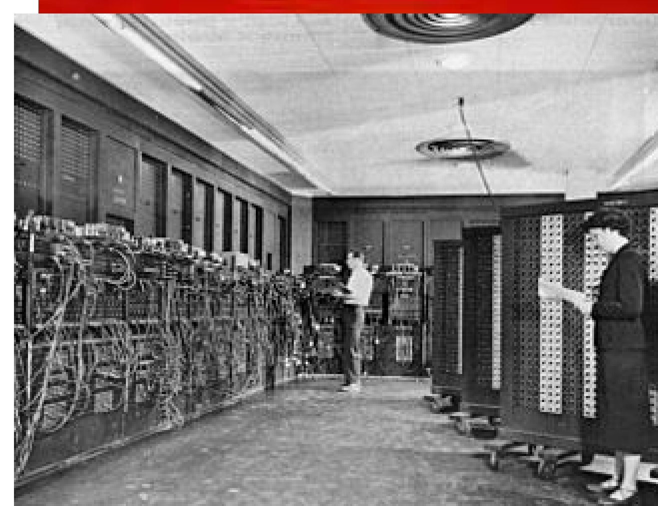
- Very concept of Monte Carlo comes in the XVIII century (Buffon, 1777, and then Laplace, 1786)
  - Monte Carlo estimate of π



- 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

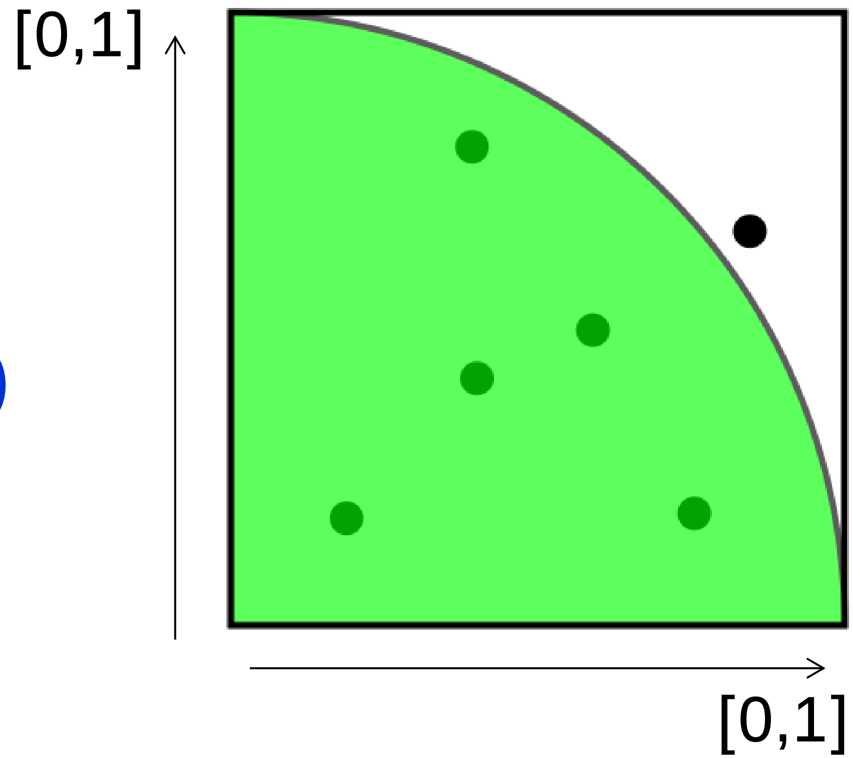
EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*

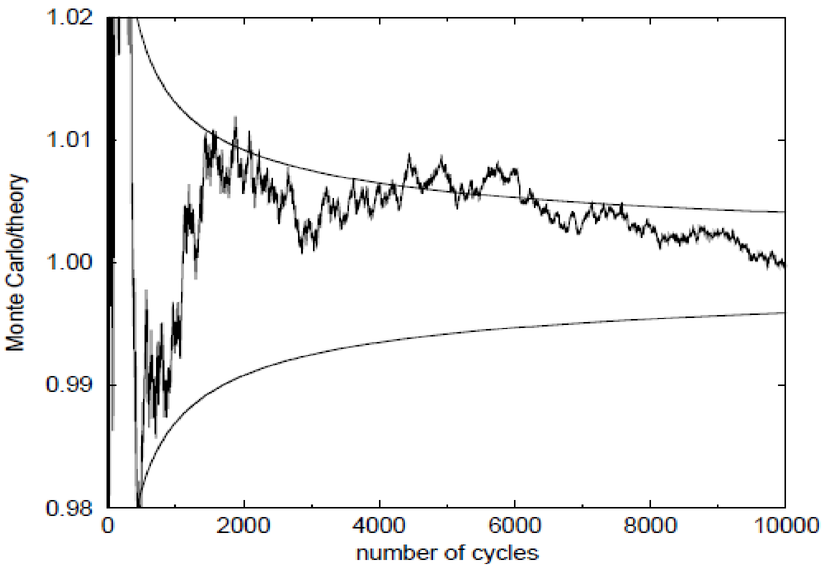
(Received March 6, 1953)

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 $(x^2+y^2\leq 1)$

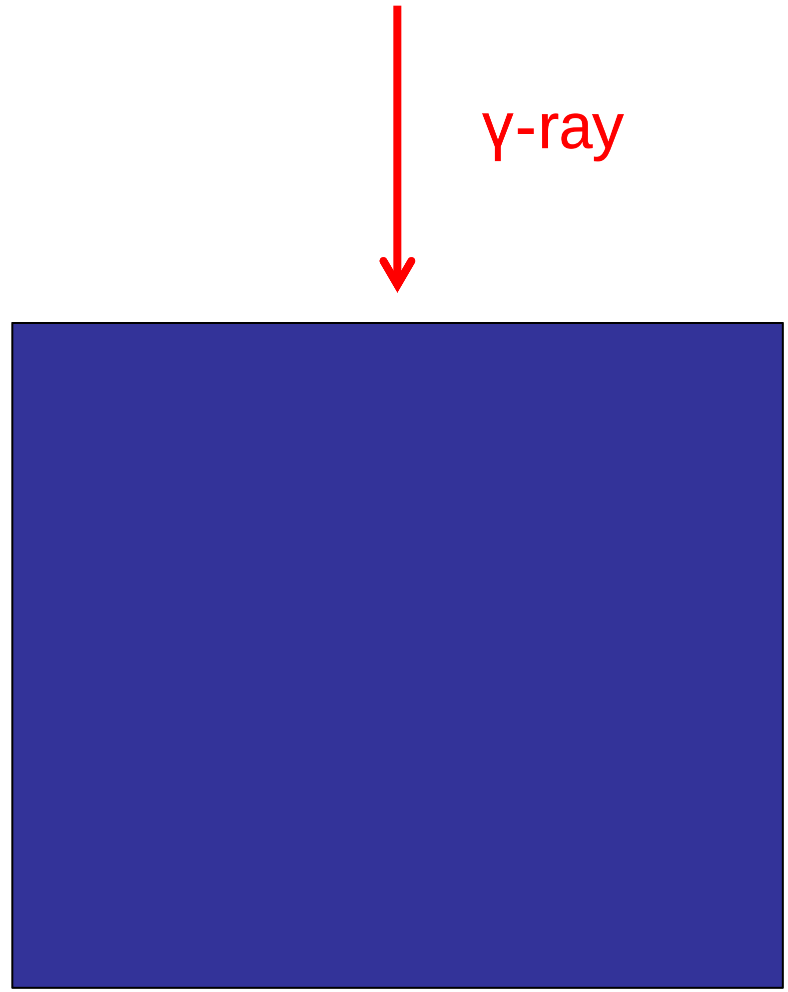




- n/N = π/4 (ratio of areas)
- Convergence as $1/\sqrt{N}$

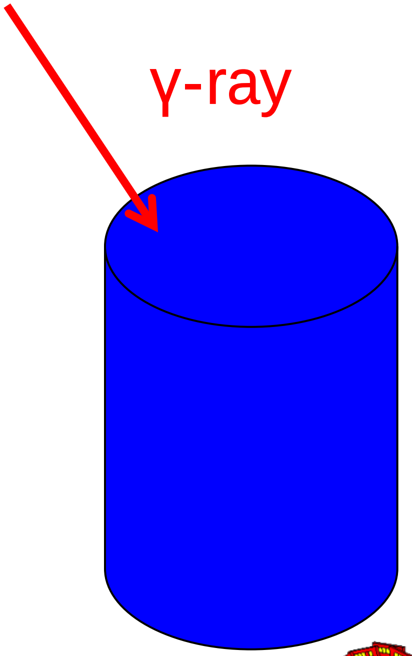
Most common application in particle physics: particle tracking

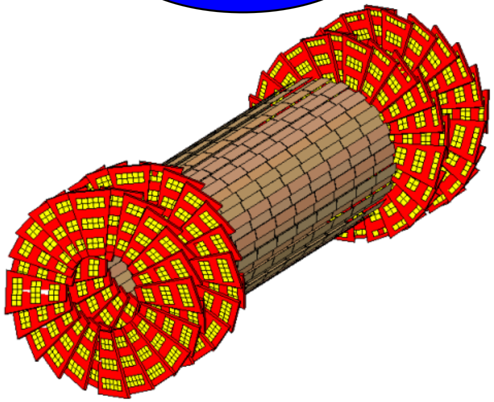
- Problem: track a γ-ray in a semi-infinite detector and determine the energy spectrum deposited
  - Still, a **model case**
- All physics is known from textbook (Compton scattering, photoelectric effect, etc.)
- Yet, the analytical calculation is a nightmare (while still possible)



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



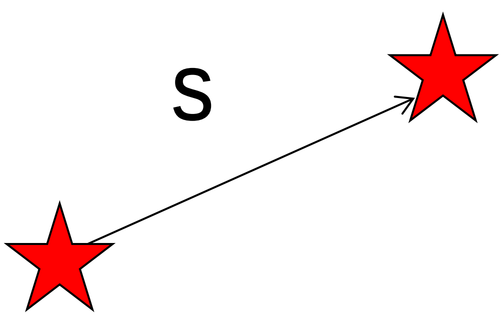


Particle tracking

- 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

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

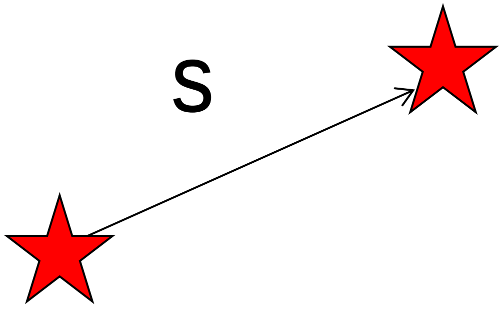
- Transition between two homogeneous materials

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

Particle tracking

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

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

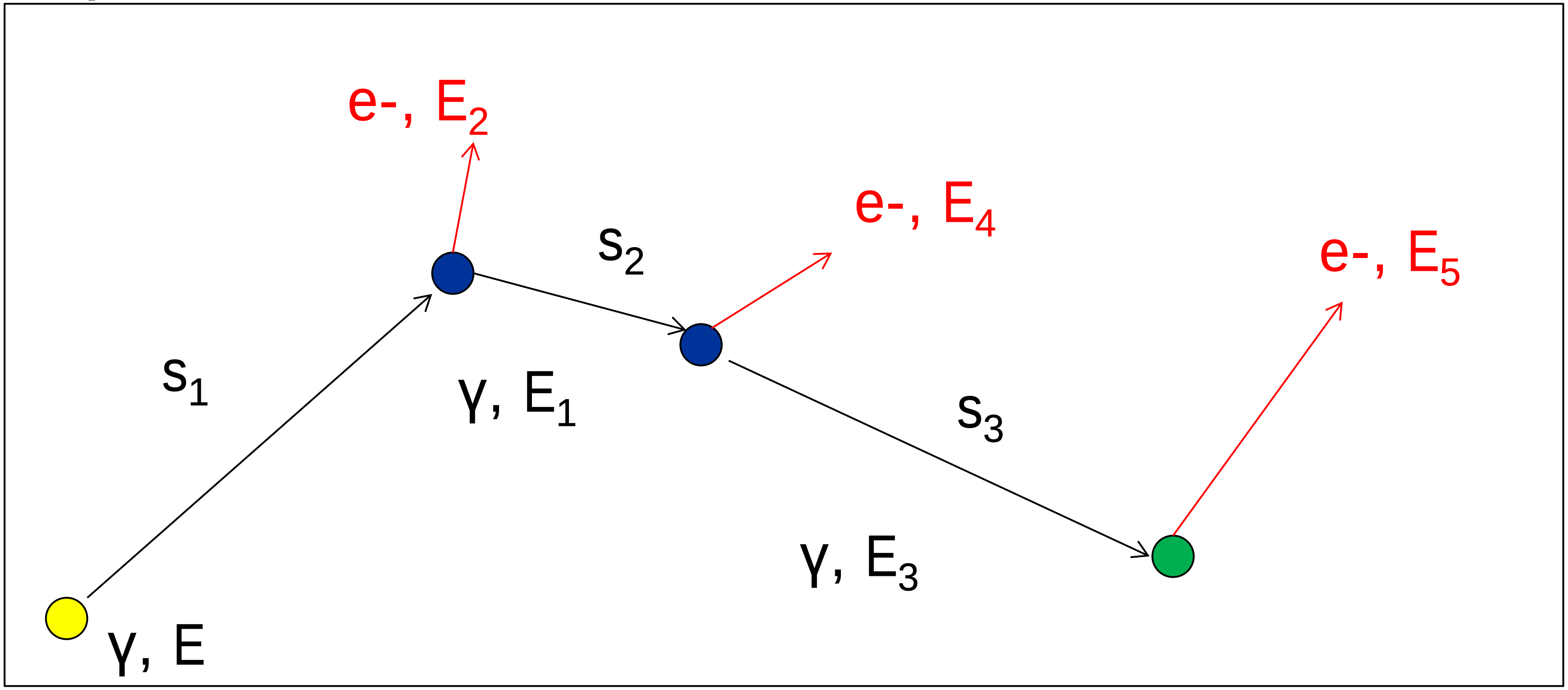


- All competing processes contribute with their own μ_i
- Each process takes place with probability μ_i/μ → i.e. proportionally to the partial cross sections

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 cumulatively
  - 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 **$\mu_h << \mu$** 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)

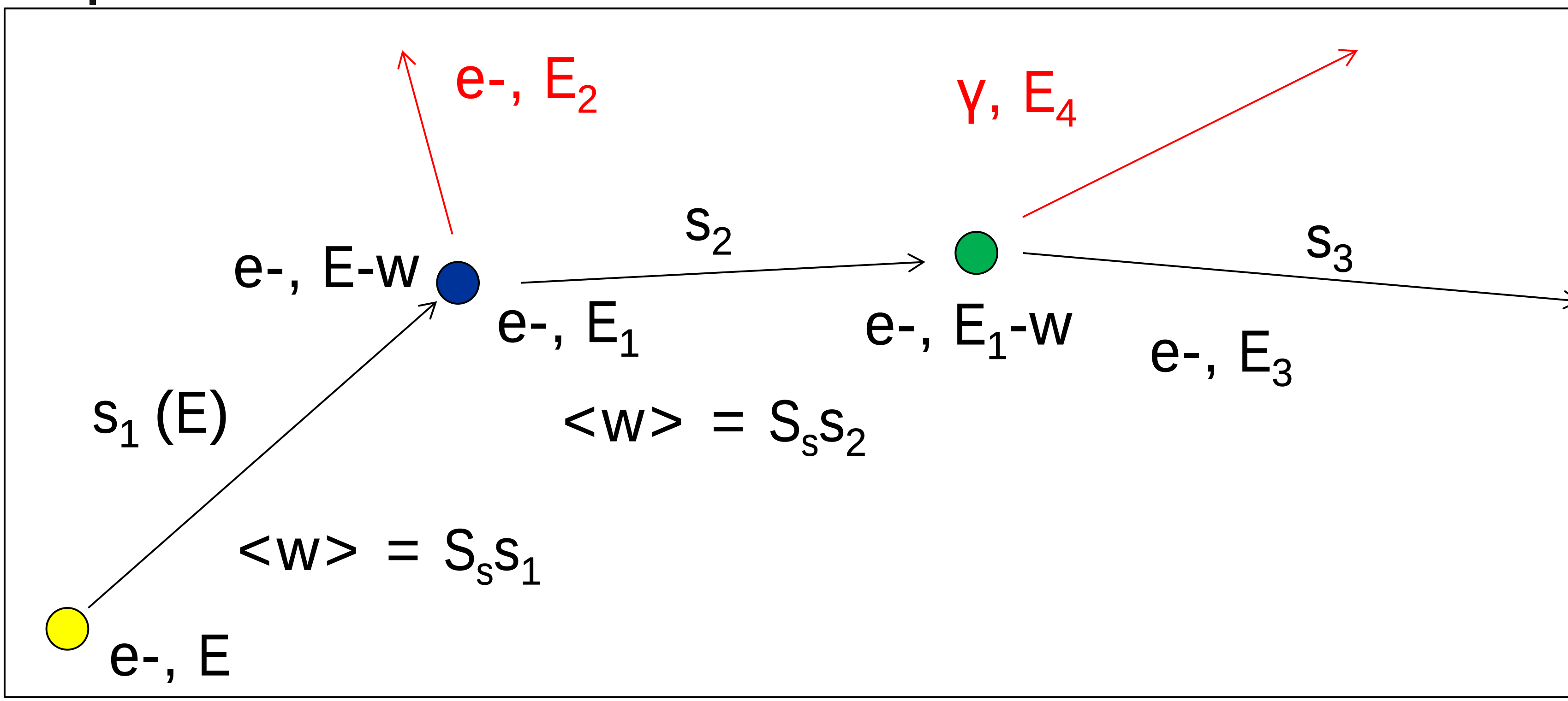
$$S_s = N \int_0^{W_0} W \frac{d\sigma}{dW}(E)\, dW$$

- Average energy lost along the step: $\langle w \rangle = sS_s$
  - Must be $\langle w \rangle \ll E$
- Fluctuations around the average value $\langle w \rangle$ have to be taken into account
  - Appropriate random sampling of w with mean value $\langle w \rangle$ 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)

S. Agostinelli et al., Nucl. Instr. Meth. A ***506*** *(2003) 250*
J. Allison et al., IEEE Trans. Nucl. Scie. ***53*** *(2006) 270*
J. Allison et al., Nucl. Instr. Meth. A ***835*** *(2016) 186*

What is Geant 4

- Toolkit for the **Monte Carlo simulation** of the interaction of **particles** with **matter**
  - physics processes (EM, hadronic, optical) cover a comprehensive set of particles, materials and over a wide energy range
  - offers a complete set of support functionalities (tracking, geometry)
  - **Distributed** software production and management: developed by an **international Collaboration**
    - Established in 1998
    - Approximately 100 members, from Europe, America and Asia
- Written in **C++** language
  - Takes advantage from the Object Oriented software technology
- **Open source**

http://geant4.org

Geant 4

Download | User Forum | Gallery
Contact Us

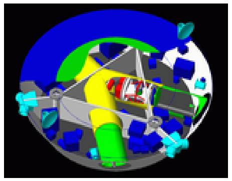
Search Geant4

Geant4 is a toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science. The three main reference papers for Geant4 are: *Nuclear Instruments and Methods in Physics Research* A **506** (2003) 250-303, *IEEE Transactions on Nuclear Science* **53** No. 1 (2006) 270-278, *Nuclear Instruments and Methods in Physics Research* A **835** (2016) 186-225.

Applications

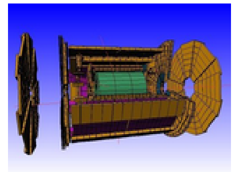
A sampling of applications, technology transfer and other uses of Geant4

User Support



Getting started, guides and information for users and developers

Publications



Validation of Geant4, results from experiments and publications

Collaboration

Who we are: collaborating institutions, members, organization and legal information

News

- *8 March 2017* - 2017 planned developments.
- *28 February 2017* - **Patch-01** to **release 10.3** is available from the Download area.
- *27 January 2017* - **Patch-03** to **release 10.2** is available from the source archive area.
- *17 February 2016* - **Patch-03** to **release 10.1** is available from the source archive area.

http://geant4.org

Events

- IN2P3 Geant4 School 2017, LAL, Orsay (France), **15-19 May 2017**.
- Geant4 Course at the 14th Seminar on Software for Nuclear, Sub-nuclear and Applied Physics, Porto Conte, Alghero (Italy), **4-9 June 2017**.
- Geant4 Tutorial and International Multidisciplinary User Workshop, Ton Duc Thang University, Ho Chi Minh City (Vietnam), **15-16 June 2017**.
- Geant4 User Workshop, Sage Hotel, Wollongong (Australia), **19-22 September 2017**.
- 22nd Geant4 Collaboration Meeting, UOW Campus, Wollongong (Australia), **25-29 September 2017**.

- **Past events**

- **Code** and **documentation** available in the main **web page**
- Regular **tutorial courses** held worldwide

Geant 4 versions and releases

- First release (Geant4 1.0) in December 1998
  - ~Two releases per year since then
  - Major releases (**x**.y) or minor releases (x.**y**) or beta releases
  - Patches regularly issued
- Last version: Geant4 10.3
  - Released December 9th, 2016
  - Now patch **10.3.p01** (February 24th, 2017)
  - This is the version of Geant4 that is installed in the VM used for this course
- Requires C++11 since 10.2 (gcc > 4.8.x)
  - Native C+11 features in-place (= compilation with old compilers **fails**)

Basic concept of Geant4

Toolkit and User Application

- Geant4 is a **toolkit** (= a collection of tools)
  - i.e. you cannot “run” it out of the box
  - You must **write an application**, which uses Geant4 tools
- Consequences:
  - There are no such concepts as “Geant4 defaults”
  - You must provide the necessary information to configure your simulation
  - You must deliberately choose which Geant4 tools to use
- Guidance: many **examples** are provided
  - **Basic Examples**: overview of Geant4 tools
  - **Advanced Examples**: Geant4 tools in real-life applications

Basic concepts

- What you **MUST** do:
  - Describe your **experimental set-up**
  - Provide the **primary particles** input to your simulation
  - Decide which **particles** and **physics models** you want to use out of those available in Geant4 and the precision of your simulation (cuts to produce and track secondary particles)

- You **may also want**
  - To interact with Geant4 kernel to **control** your simulation
  - To **visualise** your simulation configuration or results
  - To produce **histograms, tuples** etc. to be further analysed

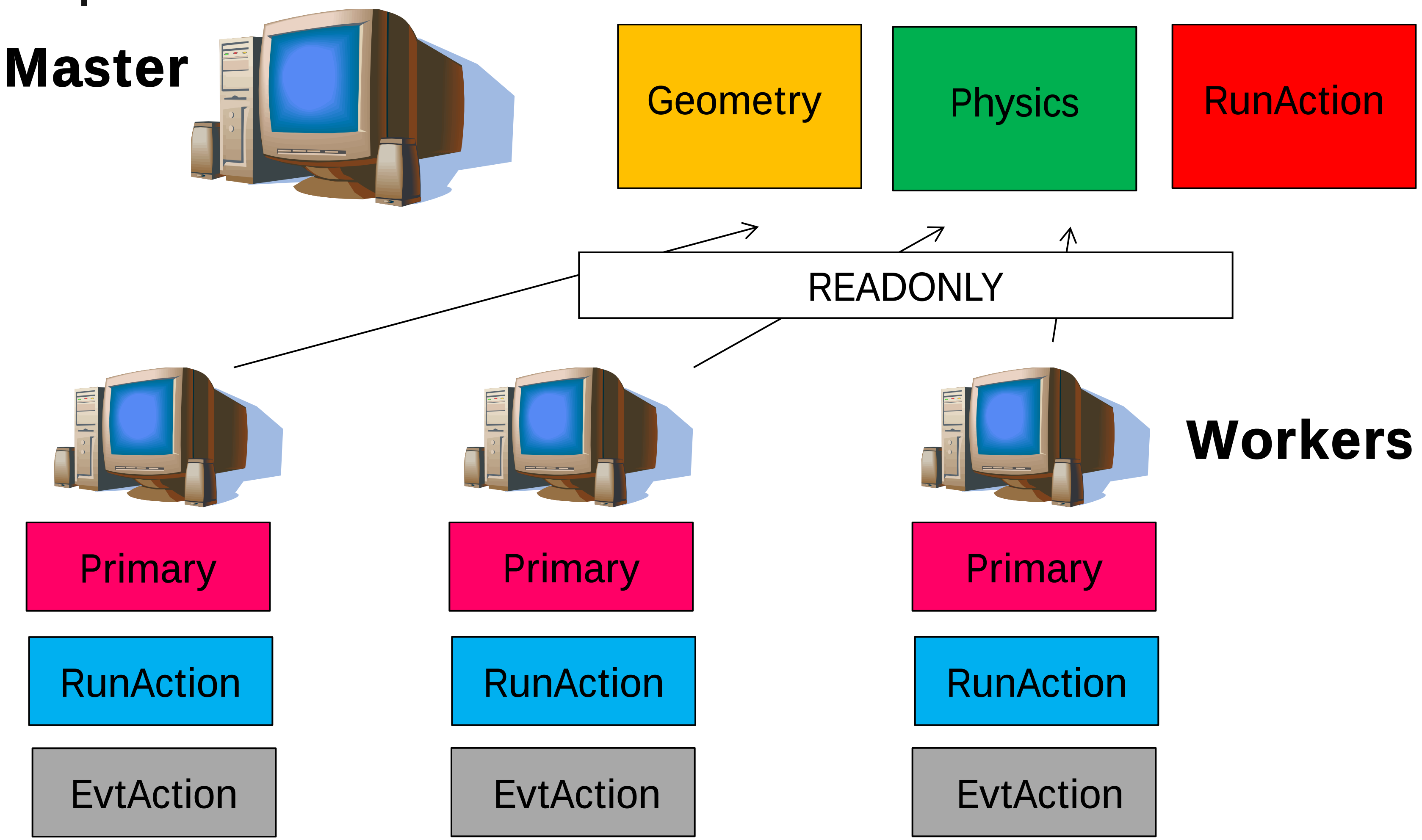
Main Geant4 capabilities

- **Transportation of a particle 'step-by-step'** taking into account all possible interactions with materials and fields
- **The transport ends** if the particle
  - is slowed down to zero kinetic energy (and it doesn't have any interaction at rest)
  - disappears in some interaction
  - reaches the end of the simulation volume
- Geant4 allows the User to access the transportation process and retrieve the results (USER ACTIONS)
  - at the beginning and end of the transport
  - at the end of each step in transportation
  - if a particle reaches a sensitive detector
  - Others...

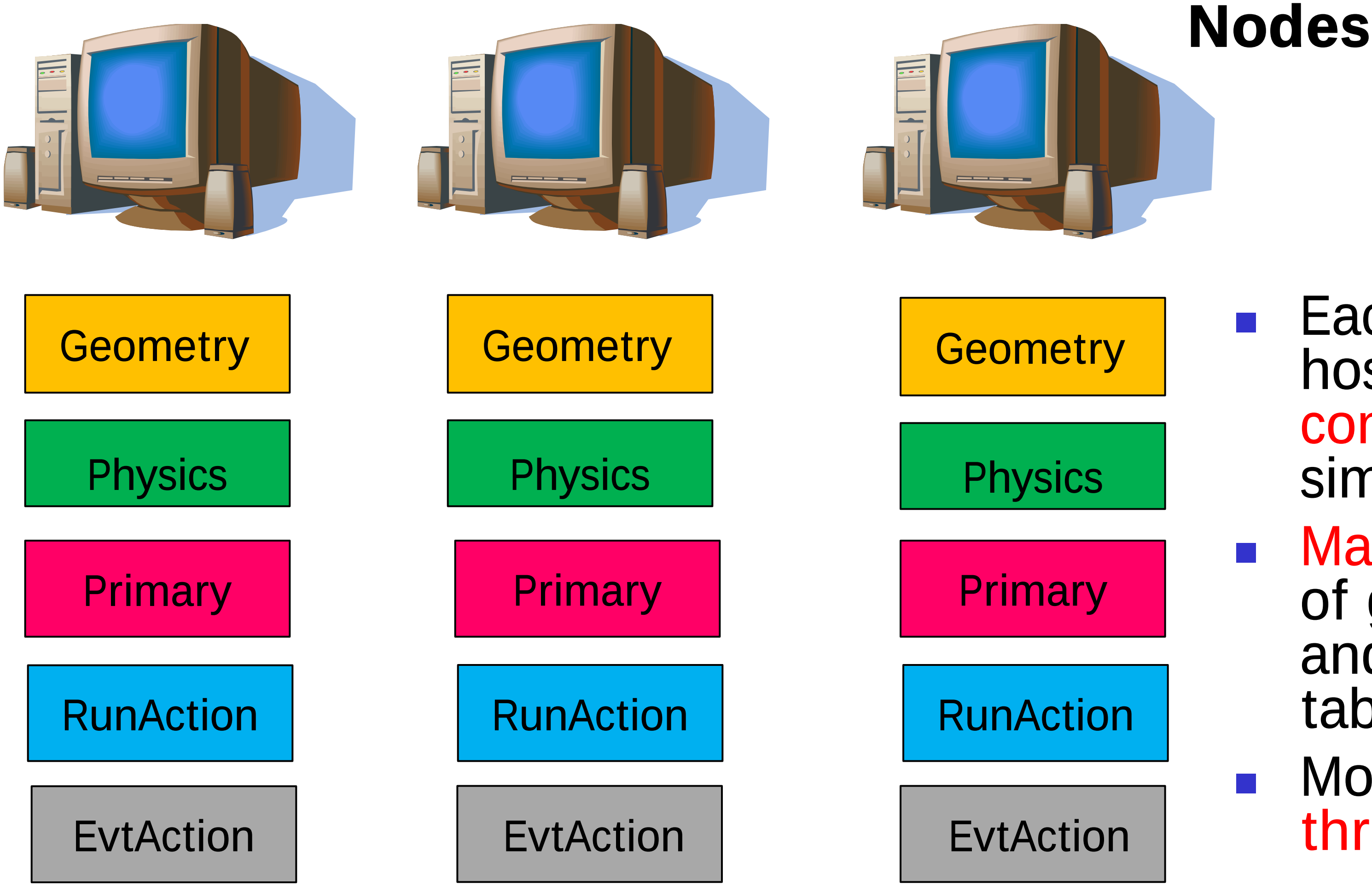
Multi-thread mode

- Geant4 10.0 (released Dec, 2013) supports multi-thread approach for multi-core machines
  - Simulation is automatically split on an event-by-event basis
    - different events are processed by different cores
  - Can fully profit of **all cores** available on modern machines → substantial **speed-up** of simulations
  - **Unique** copy (master) of geometry and physics
    - All cores have them as read-only (saves memory)
- Backwards compatible with the **sequential** mode
  - The MT programming requires some care: need to avoid conflicts between threads
  - Some modification and porting required

Concept for multi-thread ...



... vs. parallelisation



- Each node hosts a complete simulation
- Many copies of geometry and physics tables
- More memory-thristy

Who/why is using Geant4?

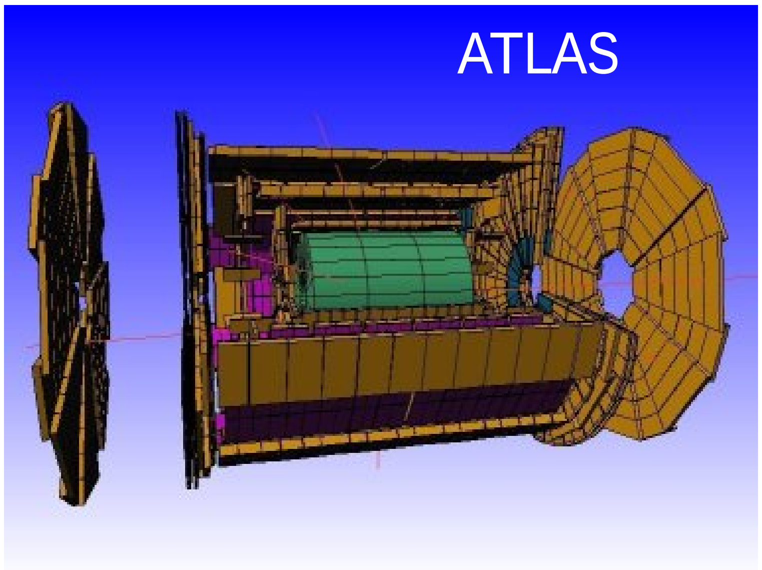
Experiments and MC

- In my knowledge, **all experiments** have a (more or less detailed) full-scale Monte Carlo simulation
- Design phase
  - Evaluation of background
  - Optimization of setup to maximize scientific yield
    - Minimize background, maximize signal efficiency
- Running/analysis phase
  - Support of data analysis (e.g. provide efficiency for signal, background, coincidences, tagging, ...).
    - Often, Monte Carlo is the only way to convert *relative rates* (events/day) in *absolute yields*

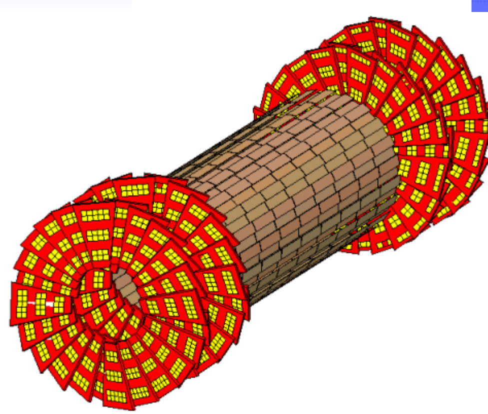
Why Geant4 is a common choice in the market

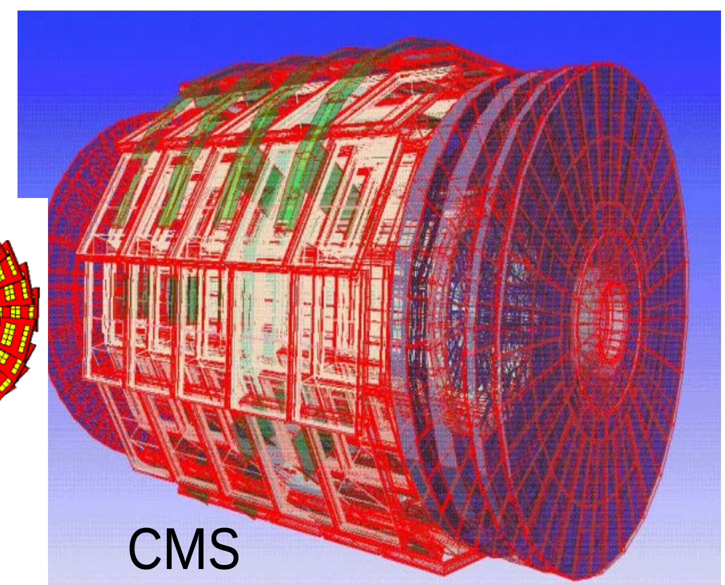
- Open source and object oriented/C++
  - No black box
  - Freely available on all platforms
  - Can be easily extended and customized by using the existing interfaces
    - New processes, new primary generators, interface to ROOT analysis, ...
- Can handle complex geometries
- Regular development, updates, bug fixes and validation
- Good physics, customizable per use-cases
- End-to-end simulation (all particles, including optical photons)

LHC @ CERN



- All four big **LHC experiments** have a Geant4 simulation
  - M of volumes
  - Physics at the TeV scale



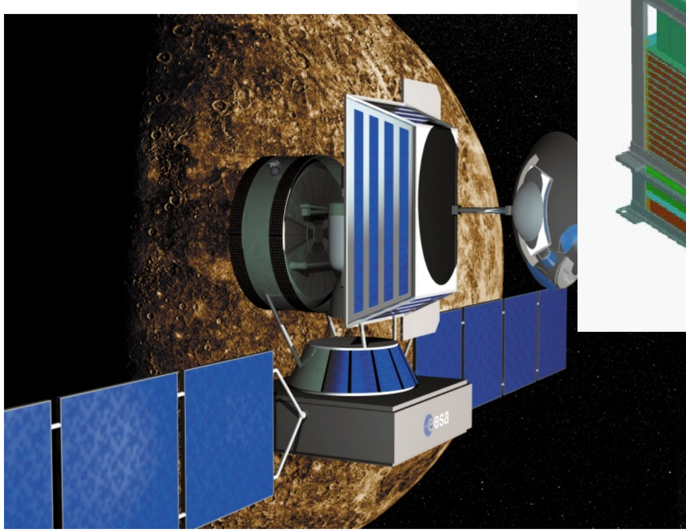


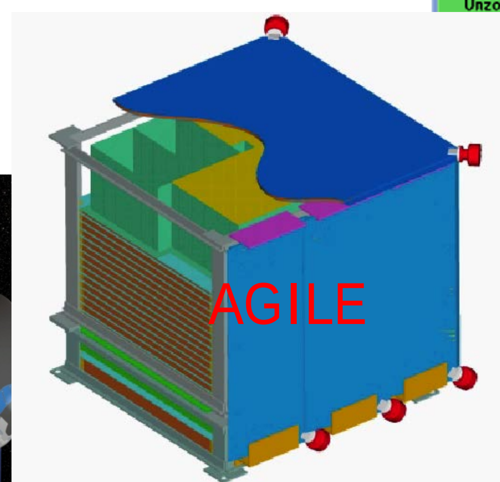
- Benchmark with test-beam data
- Key role for the Higgs searches

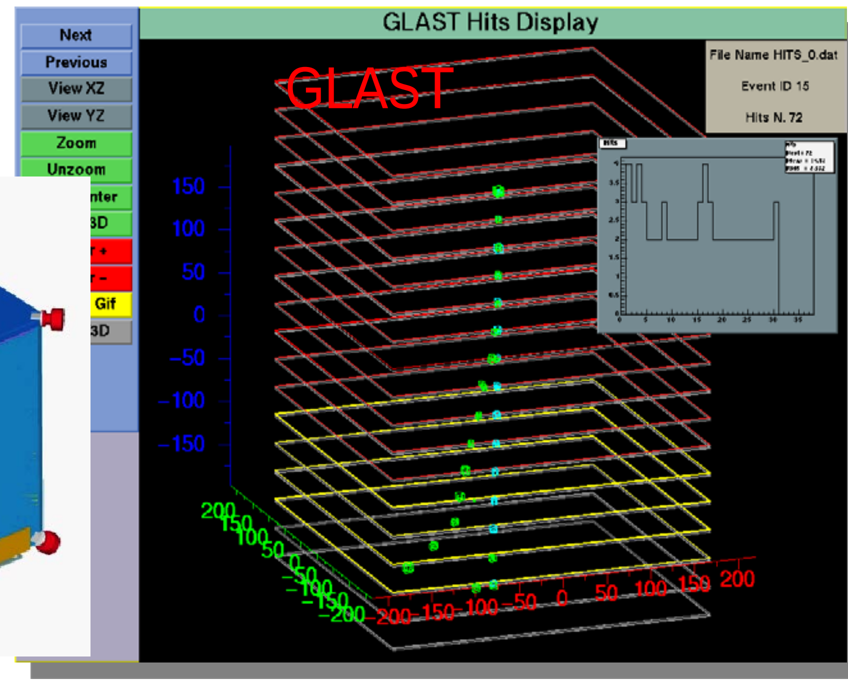
Space applications

- Satellites (γ astrophysics, planetary sciences)
- Funding from ESA

Typical telescope:
Tracker
Calorimeter
Anticoincidence

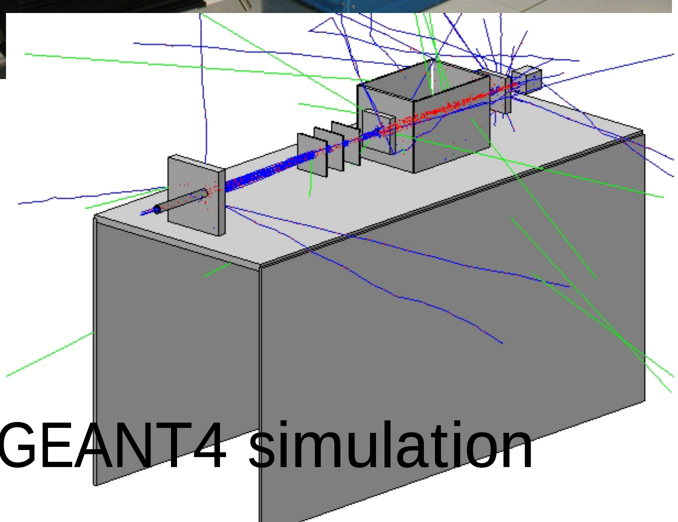






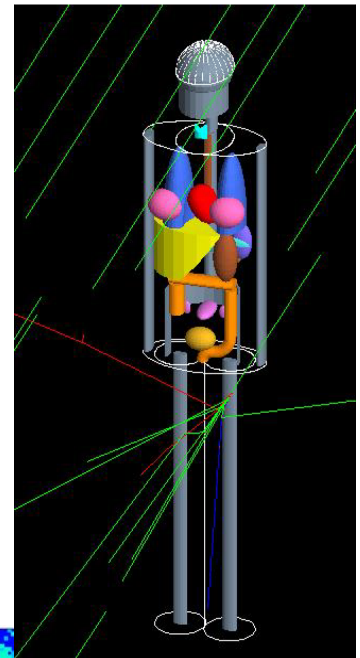
Medical applications

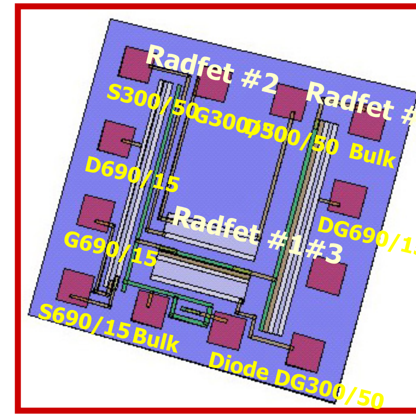

Proton-therapy beam line


GEANT4 simulation

- **Treatment planning** for hadrontherapy and proton-therapy systems
  - <u>Goal</u>: deliver dose to the tumor while sparing the healthy tissues
  - Alternative to less-precise (and commercial) TP software
- Medical imaging
- Radiation fields from medical accelerators and devices
  - medical_linac
  - gamma-knife
  - brachytherapy

Dosimetry with Geant4

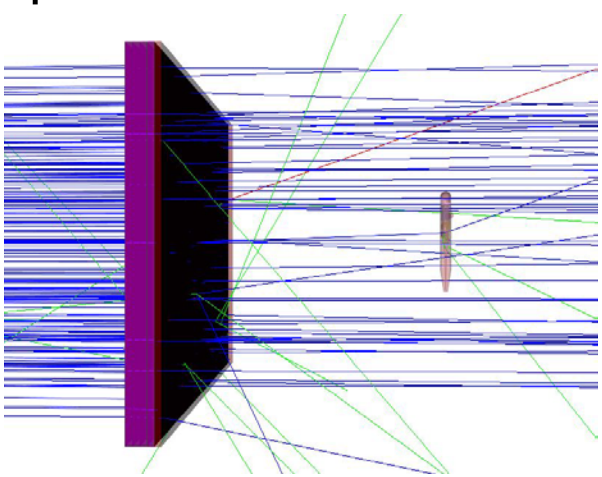


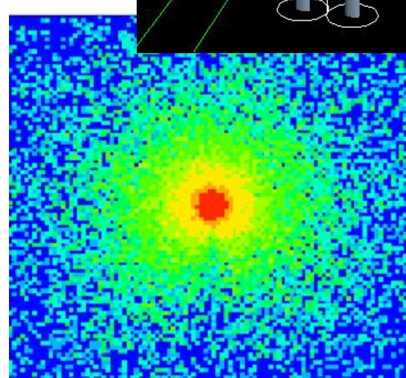


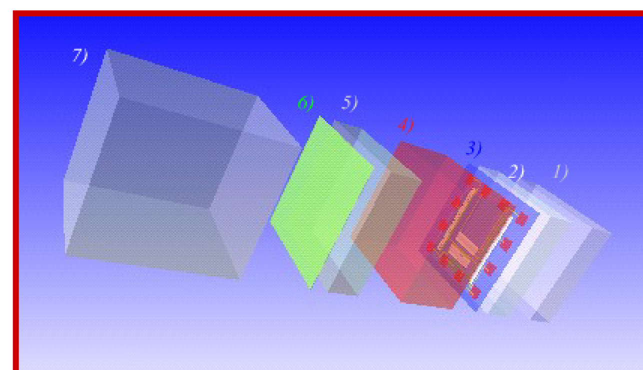
Space science

Radiotherapy

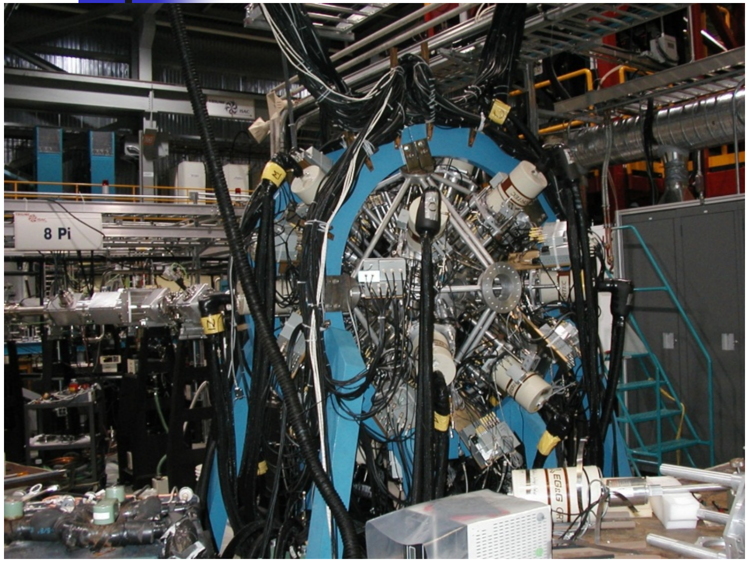
Effects on electronics components

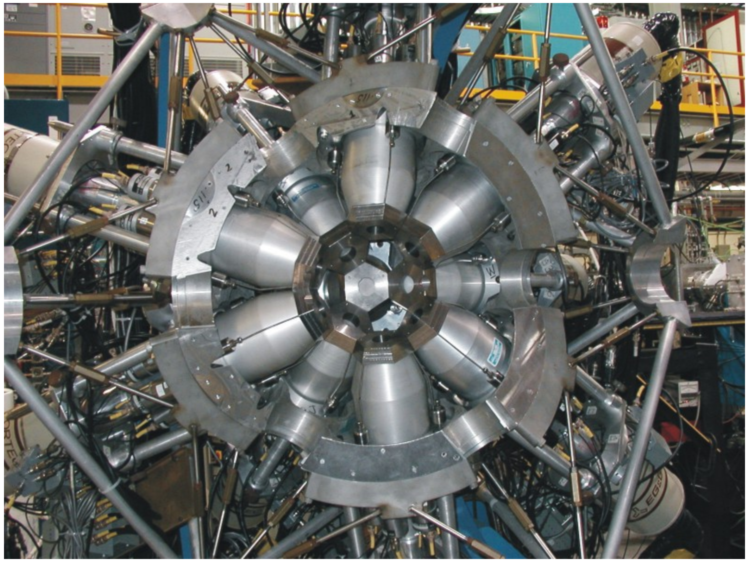


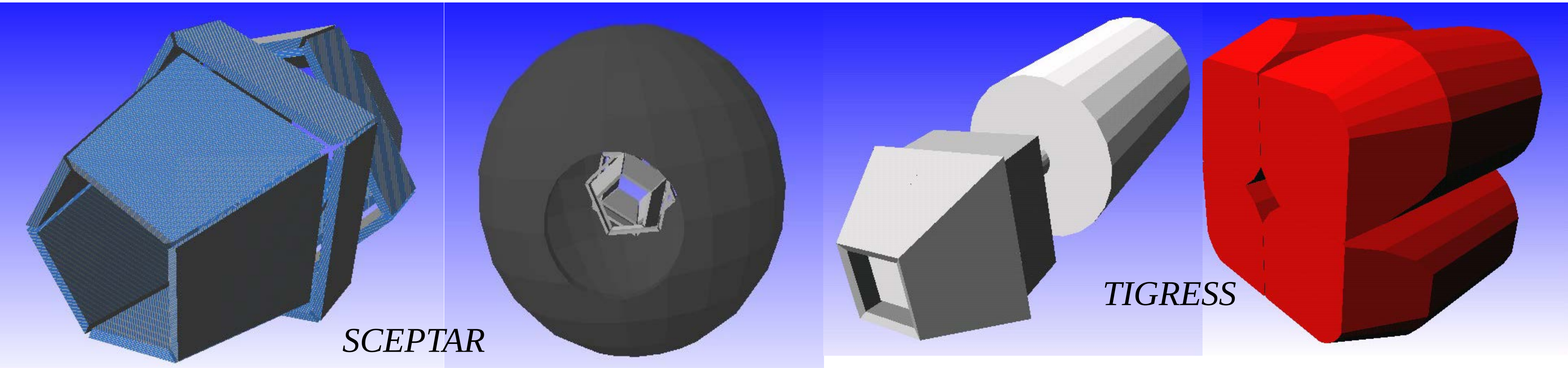




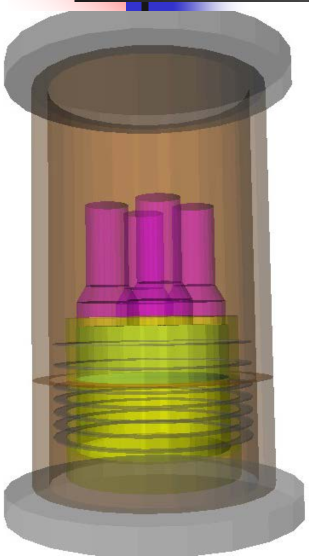
Nuclear spectroscopy





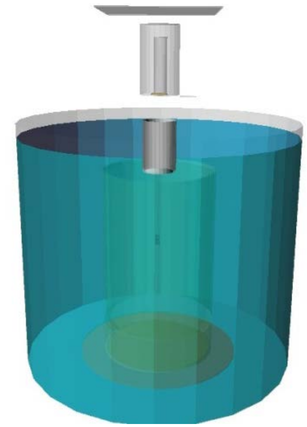


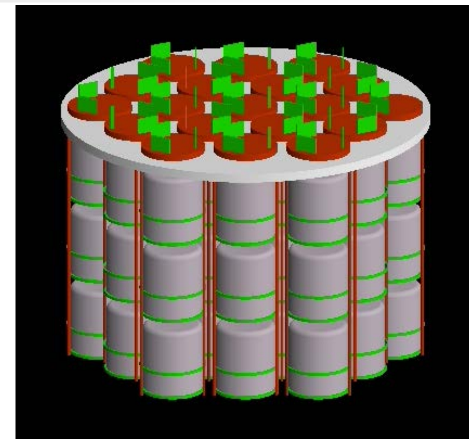
Low background experiments

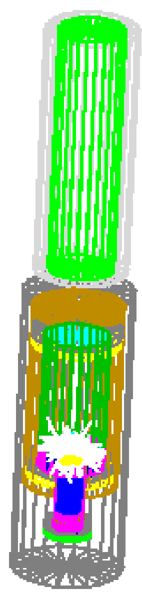


Neutrinoless ββ decay:

GERDA, Majorana
COBRA, CUORE, EXO

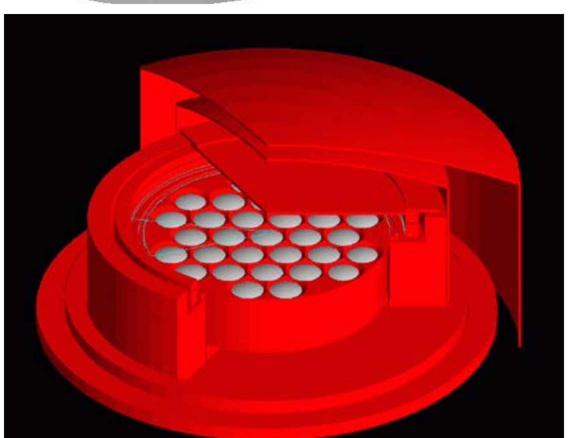






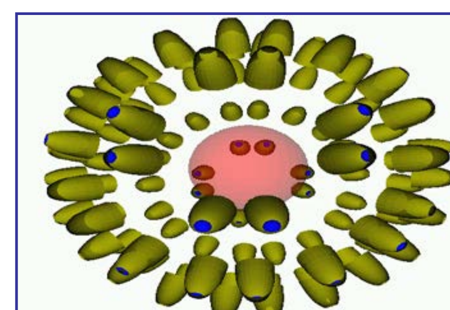
Dark matter detection:

Zeplin-II/III, Drift, Edelweiss, ArDM,
Xenon, CRESST, Lux, Elixir,



Solar neutrinos:

Borexino, …



Applications in the rare-event physics

Experiment backgrounds
internal detector radioactivity
rock radioactivity
μ-induced neutron production
shielding and veto systems

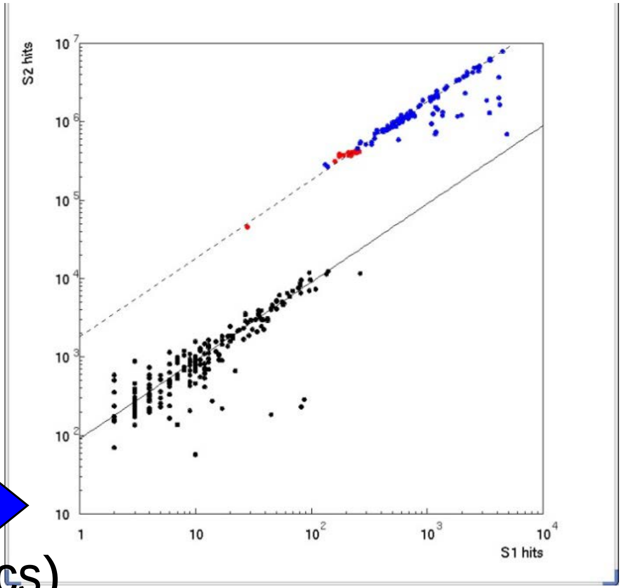
Geant4 is uniquely suited for **integrated** simulations of underground and low-background detectors (e.g. dark matter)

Optics
Photon generation
Light collection

Detector response
Scintillation
Ionisation

Simulated Data
Visualisation
Run-time analysis
Input to data analysis software

Calibration
Neutrons
Gammas



A dedicated advanced example (underground_physics) is **released** with Geant4 (ZEPLIN experiment)

Geant4-based frameworks in astroparticle/neutrino physics

- Geant4 is a **toolkit** → can be used in software projects of **wider scope**
  - Flexibility in selecting geometries, physics, outputs, ...
- A few examples in astroparticle physics:
  - MaGe (GERDA/Majorana): double beta decay
  - LUXSim (LUX): dark matter and undeground experiments
  - DCGLG4sim (Double Chooz): liquid scintillator and reactor neutrinos
  - artG4 (FermiLab)
  - VENOM (COBRA): double beta decay
  - Just google "Geant4-based"
- (Many more for HEP, space physics, medical physics)

Geant4-based frameworks in the medical physics