A Geant4 introduction

VIII Seminario sul Sftware per la Fisica Nucleare, Subnucleare e Applicata
Hotel Portoconte, Alghero
6 - 10 Giugno 2011

Date

Where you can find this material?

- The official Geant4 web pages
  www.cern.ch/geant4
- This course web pages
  www.lngs.infn.it

cirrone@lns.infn.it,http://workgroup.lngs.infn.it/geant4lns

Outline

- The Monte Carlo approach
- Geant4 Toolkit and Collaboration
- Basic concepts and capabilities

The Monte Carlo method

*"**Monte Carlo methods** are a class of **computational algorithms** that rely on repeated **random sampling** to compute their results. Monte Carlo methods are often used when **simulating physical** and **mathematical systems**. Because of their reliance on repeated computation and **random** or **pseudo-random numbers**, Monte Carlo methods are most suited to calculation by a computer. Monte Carlo methods tend to be used when it is infeasible or impossible to compute an exact result with a deterministic algorithm".*

WIKIPEDIA
The Free Encyclopedia

cirrone@lns.infn.it,http://workgroup.lngs.infn.it/geant4lns

The Monte Carlo method

"**Monte Carlo methods** are a class of **computational algorithms** that rely on repeated **random sampling** to compute their results. Monte Carlo methods are often used when **simulating physical** and **mathematical systems**. Because of their reliance on repeated computation and **random** or **pseudo-random numbers**, Monte Carlo methods are most suited to calculation by a computer. Monte Carlo methods tend to be used when it is infeasible or impossible to compute an exact result with a deterministic algorithm".

WIKIPEDIA
The Free Encyclopedia

The name **"Monte Carlo"** was popularized by physics researchers Stanislaw Ulam, Enrico Fermi, John von Neumann, and Nicholas Metropolis, among others; the name is a reference to the **Monte Carlo Casino** in **Monaco** where Ulam's uncle would borrow money to gamble. The use of **randomness** and the repetitive nature of the process are analogous to the activities conducted at a casino.

cirrone@lns.infn.it,http://workgroup.lngs.infn.it/geant4lns

The Monte Carlo method

- It is a mathematical approach using a sequence of random numbers to solve a problem

"If we are interested in a parameter of, i.e., an equation: we must construct a big number of this equations, using different random numbers, and estimate the parameter and its variance"

A. F. Bielajew, 2001

The Monte Carlo method

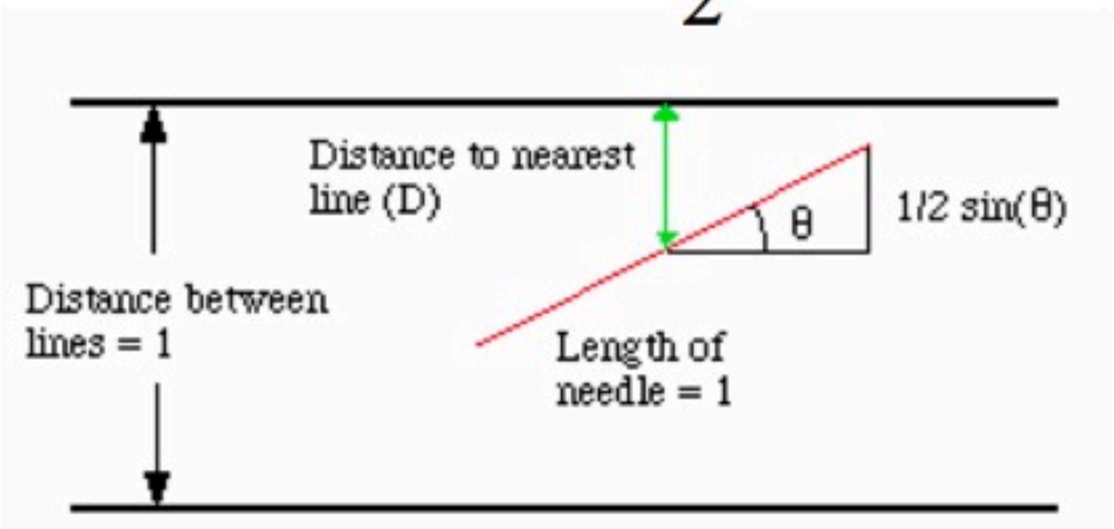
- In particle transport, if interaction physics models are known, MC can be used to calculate the parameters of the motion equations in a given configuration
- Particles are tracked one-by-one, step-by-step and, after a reasonable number, the correct information can be extracted
- MC is very time consuming but it shows many advantages

The Buffon experiment: Monte Carlo evaluation of π

Two variables: θ and D

$$0 \le \vartheta \le \pi$$

$$0 \le D \le \frac{1}{2}$$



Georges Louis Leclerc
Comte de Buffon
(07.09.1707.-16.04.1788.)

The needle will hit the line if the closest distance to a line D is

$$D \le \frac{1}{2}\sin(\vartheta)$$

cirrone@lns.infn.it,http://workgroup.lngs.infn.it/geant4lns

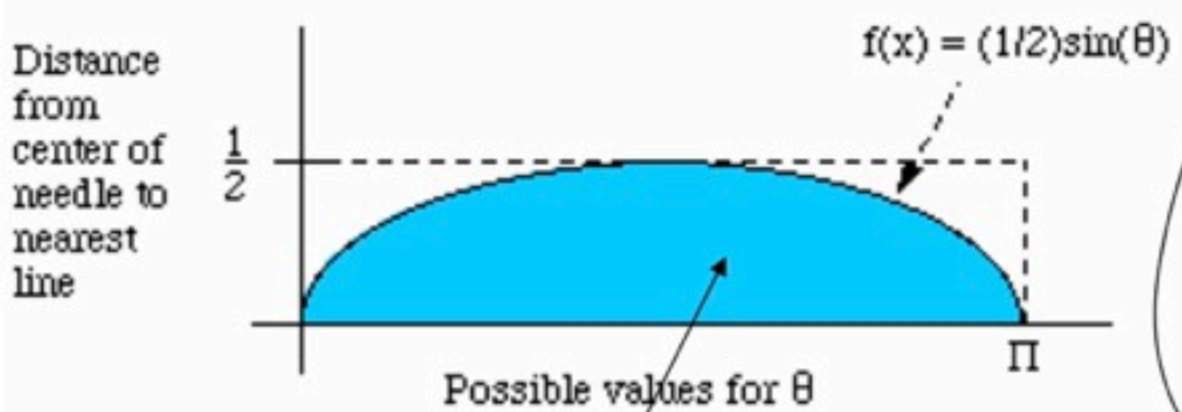
The Buffon experiment: Monte Carlo evaluation of π

The probability of an hit is the ratio of the blue area (S_{blue}) to the entire rectangle R

$$S_{blue} = \int_0^{\pi} \frac{1}{2}\sin(\vartheta) = 1$$

$$R = \frac{1}{2}\cdot \pi$$

$$\frac{S_{blue}}{R} = \frac{2}{\pi}$$



$$D \leq \frac{1}{2}\sin(\vartheta)$$

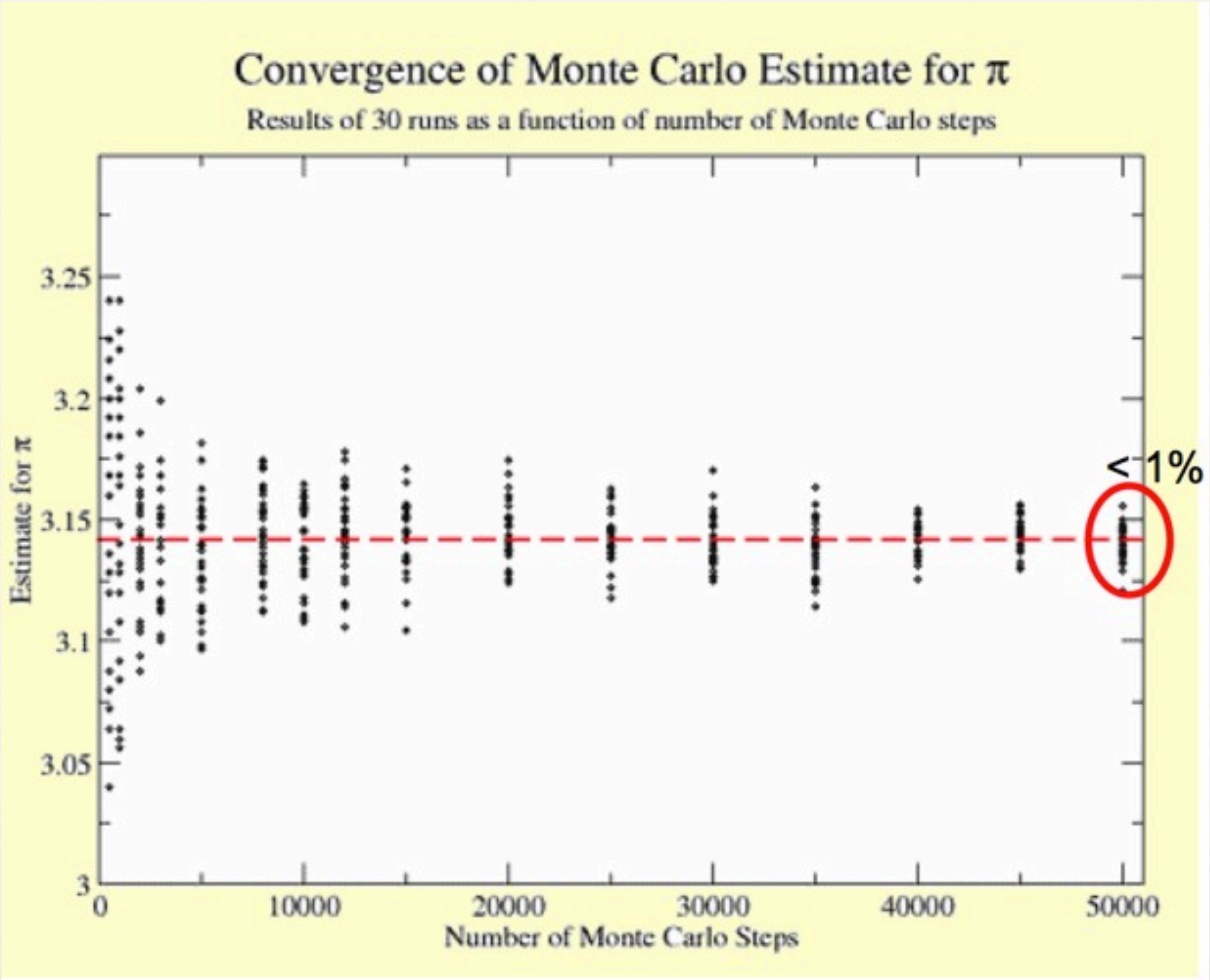
N_0 times the needle was shot

N times the needle hit the line

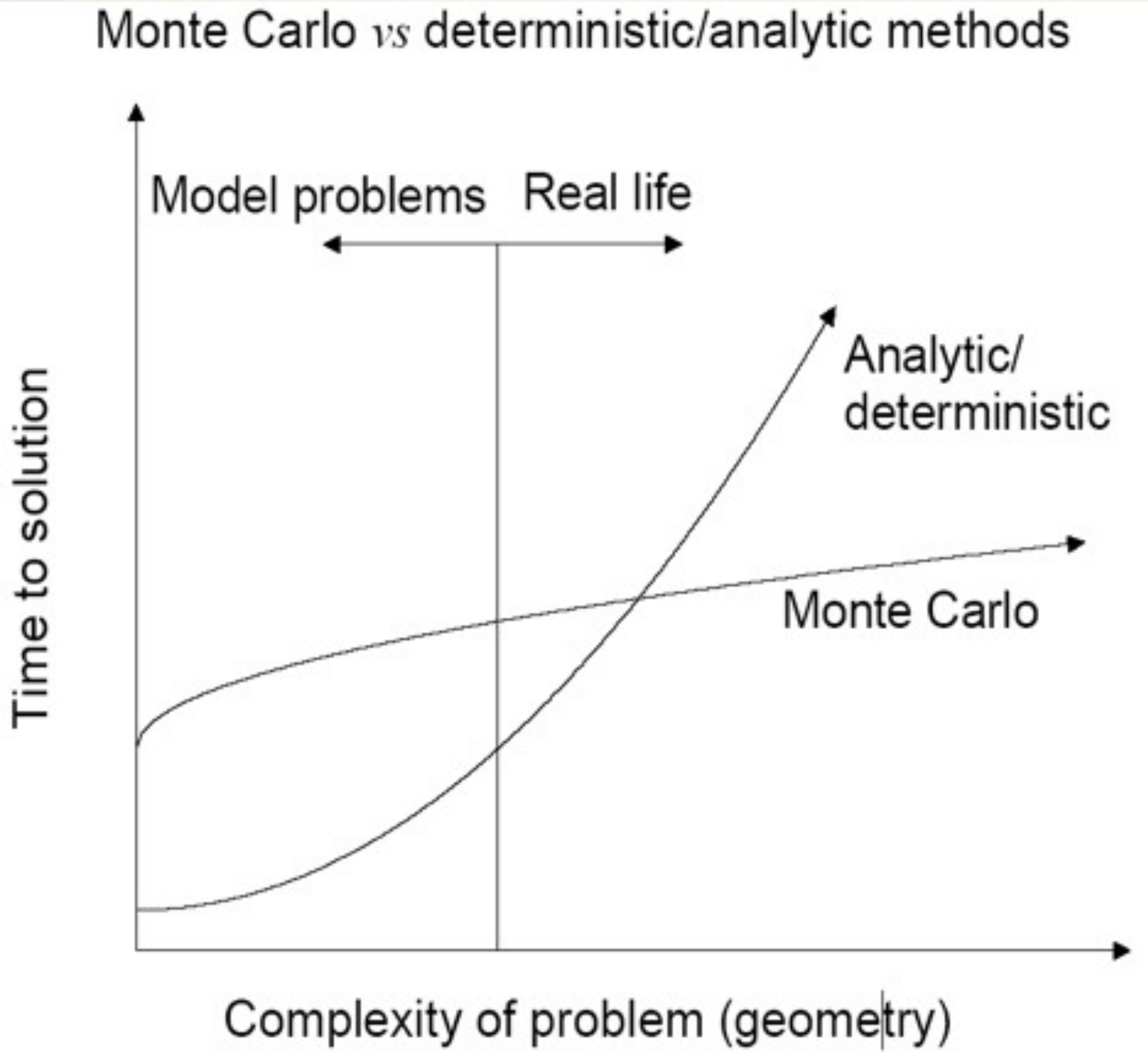
$$\frac{N}{N_0} = \frac{2}{\pi}; \rightarrow \pi = 2\cdot \frac{N_0}{N}$$

cirrone@lns.infn.it,http://workgroup.lngs.infn.it/geant4lns

The Buffon experiment: Monte Carlo evaluation of π



The Monte Carlo method



Plot from Alex F. Bielajew, 2001

Mathematical proofs exist demonstrating that MC is the most efficient way of estimate quantity in 3D when compared to first-order deterministic method

The Monte Carlo origins

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, Septem-

With MANIAC: the first electronic digital computer

Fermi's work on pion-proton phase shift analysis

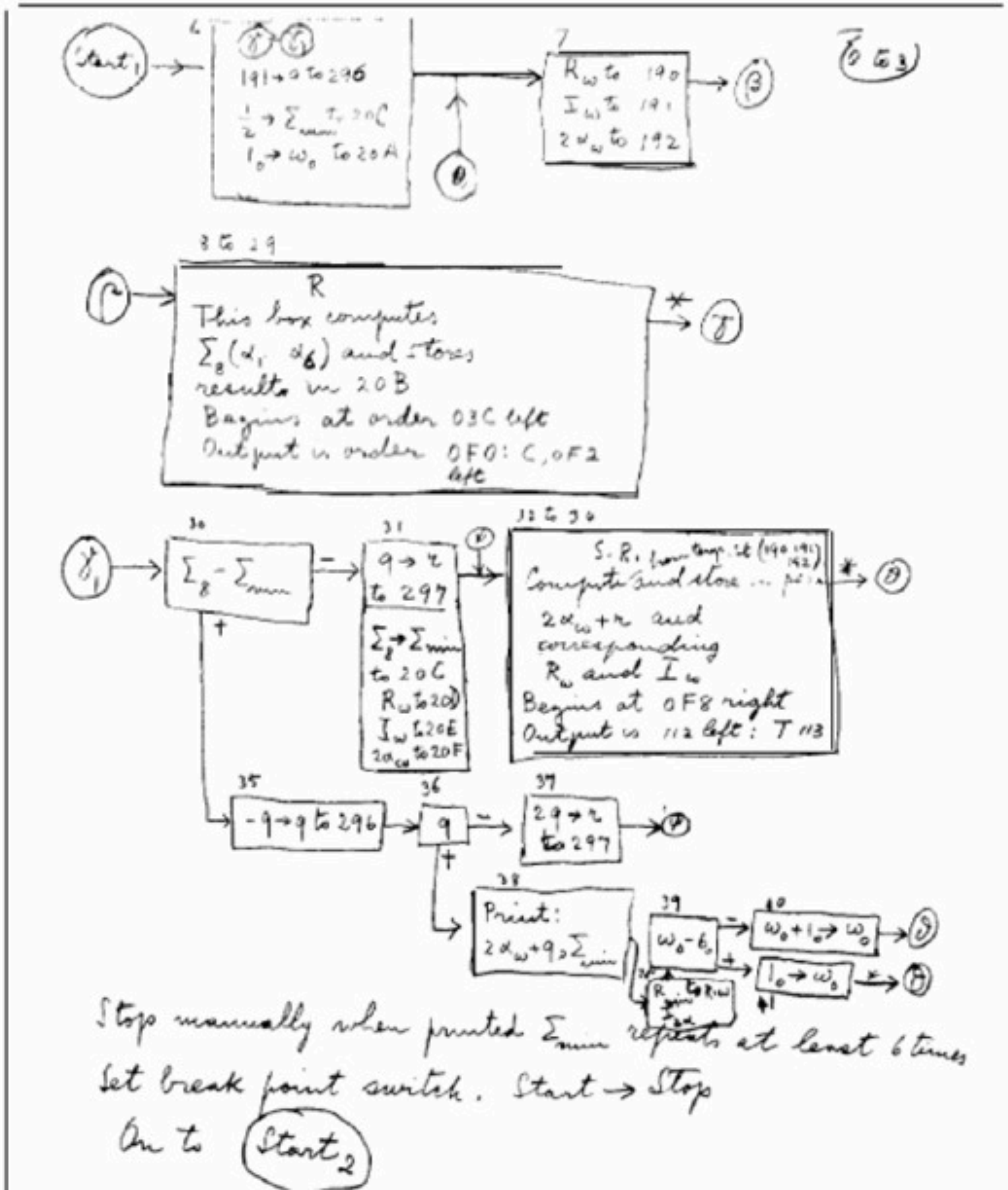
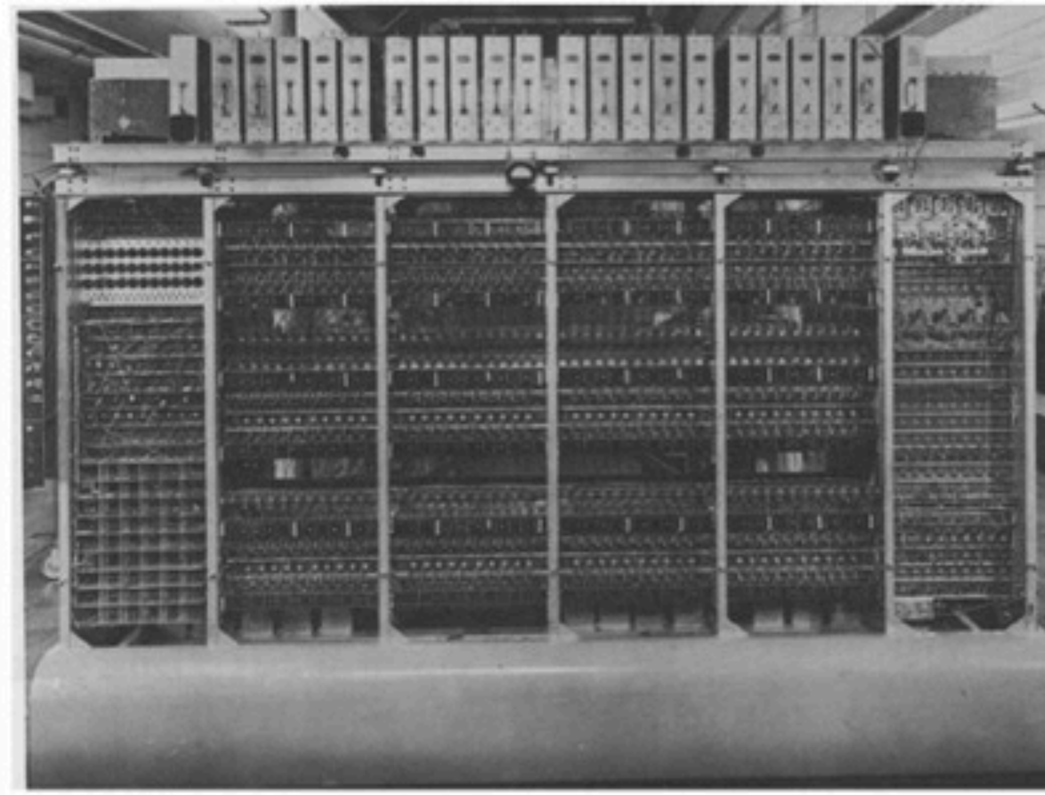


Fig. 4. A subprogram written by Fermi for calculating phase shifts by finding a minimum chi-squared in a fit to the data.



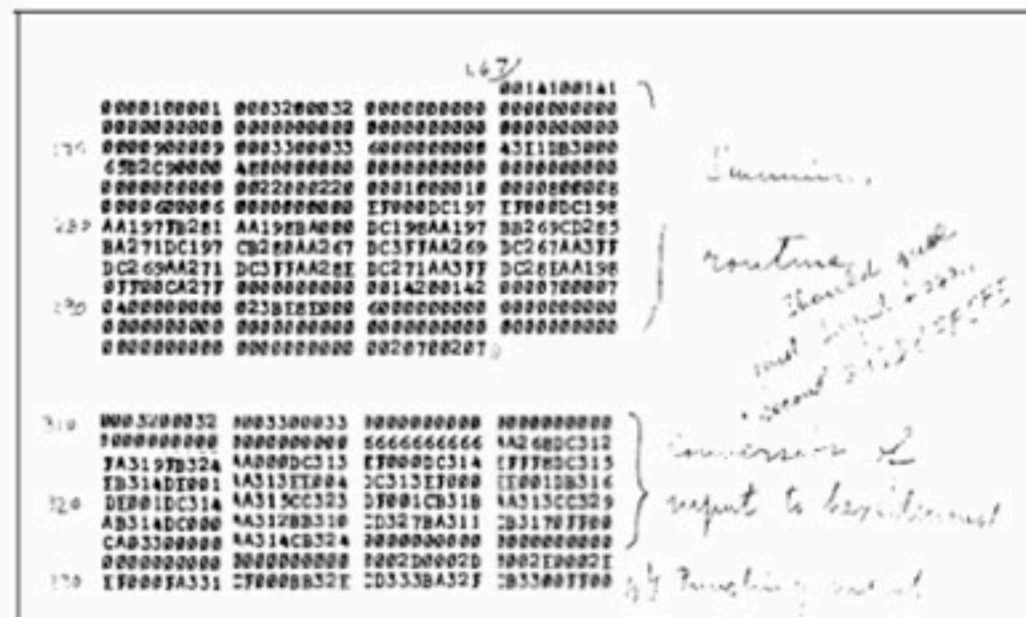


Fig. 5. A portion of the printout of the program containing the subprograms described in Figs. 3 and 4. The program is written in machine language in hexadecimal numbers.

LOS ALAMOS SCIENCE Fall 1986

Monte Carlo codes

- MCNP (neutrons mainly)
- Penelope (e- and gamma)
- PETRA (protons)
- EGSnrc (e- and gammas)
- PHIT (protons/ions)
- FLUKA (any particle)
- Geant4
  - GEometry ANd Traking
  - Geant4 - a simulation toolkit
    Nucl. Inst. and Methods Phys. Res. A, 506:250:303
  - Geant4 developments and applications
    Transaction on Nuclear Science 53, 270-278

GEANT4: www.cern.ch/Geant4

- C++ language
- Object Oriented
- Open Source
- It is a toolkit, i.e. a collection of tools the User can use for his/her simulation
- Consequences:
  - There are not such concepts as "Geant4 defaults"
  - You must provide the necessary the necessary information to configure your simulation
  - You must choose the Geant4 tool to use
- Guidance: many examples are provided:
  - Novice examples: overview of the Geant4 tools
  - Advanced Examples: Geant4 tools in real-life applications

The Geant4 Collaboration

MoU based

Distribution, Development and User Support of Geant4

CERN, ESA, KEK, SLAC, TRIUMF, TJNL

INFN, IN2P3, PPARC

Barcelona Univ., Budker Inst., Frankfurt Univ., Karolinska Inst., Helsinki Univ., Lebedev Inst., LIP, Northeastern Univ. *etc.*

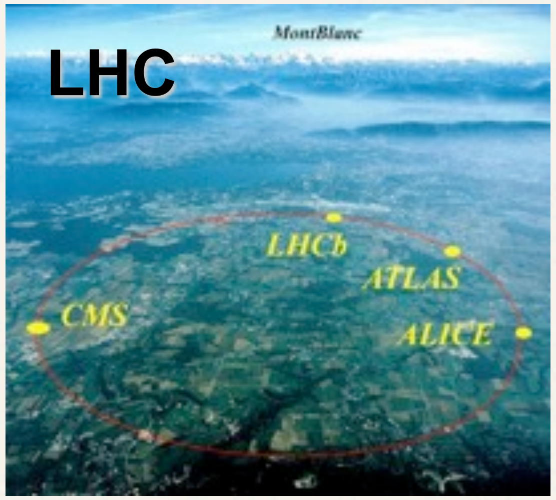
cirrone@lns.infn.it,http://workgroup.lngs.infn.it/geant4lns

Geant4 applications

Geant 4

R&D phase: **RD44**, 1994 - 1998
1st release: December 1998
2 new releases/year since then

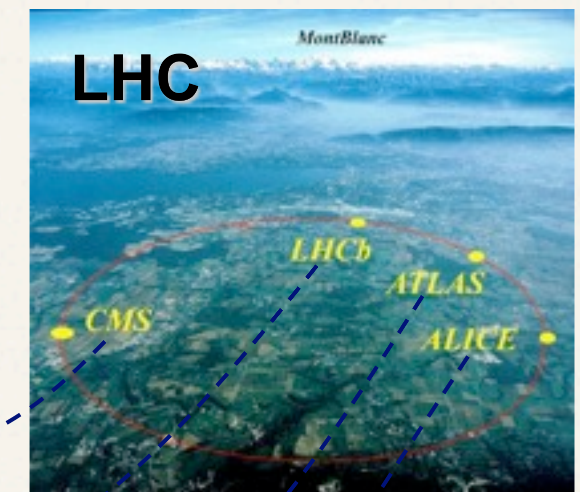
Object Oriented Toolkit (C++) born for the simulation of large scale HEP experiments at CERN (Geneva)

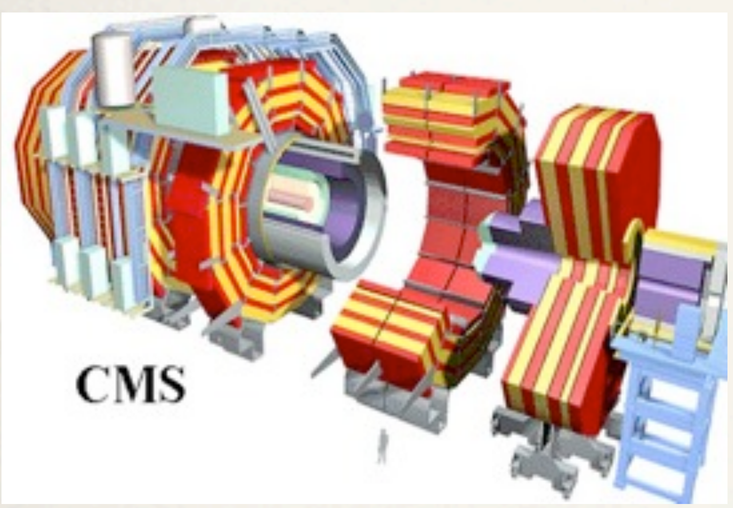


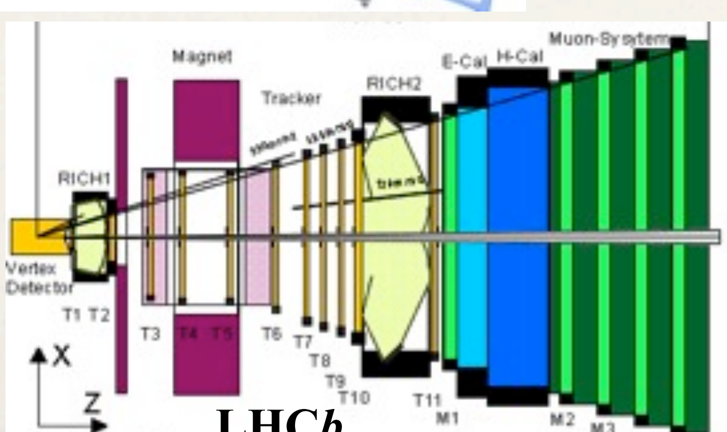
Geant4 applications

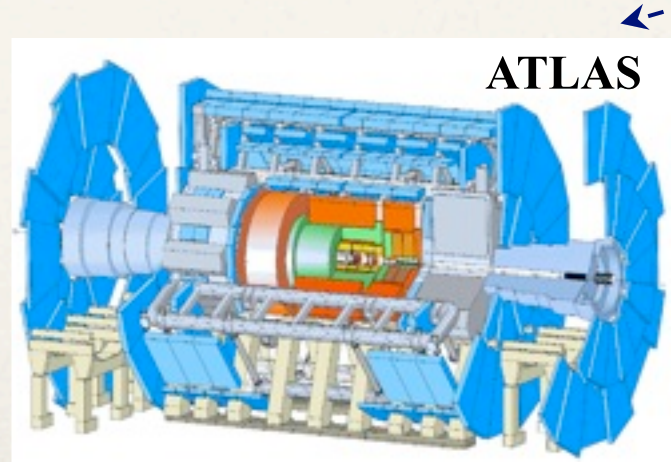
Object Oriented Toolkit (C++) born for the simulation of large scale HEP experiments at CERN (Geneva)

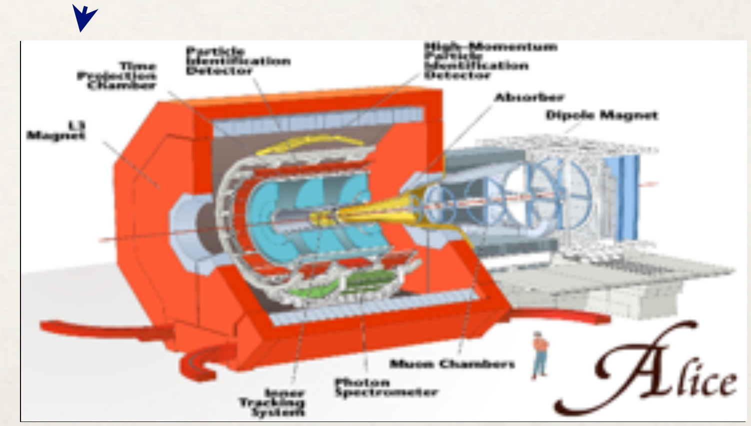
R&D phase: **RD44**, 1994 - 1998
1st release: December 1998
2 new releases/year since then









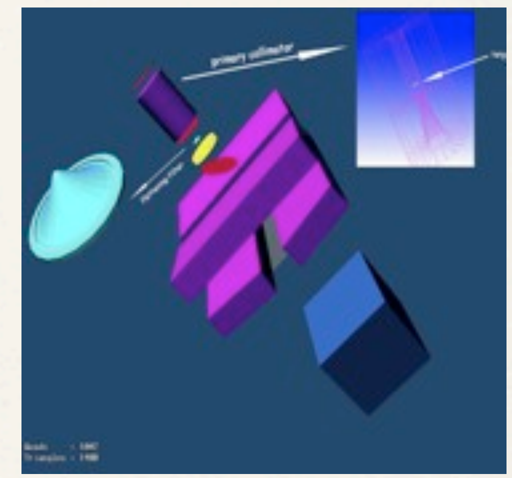


cirrone@lns.infn.it,http://workgroup.lngs.infn.it/geant4lns

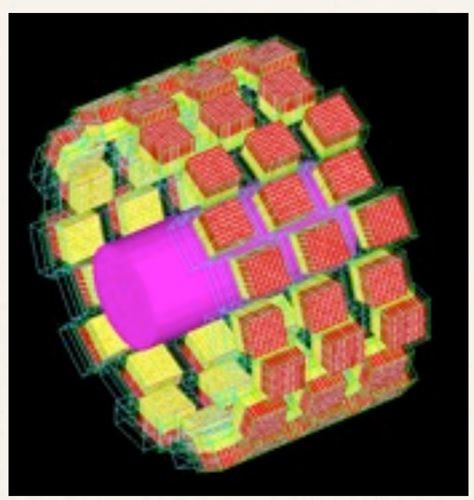
Geant4 applications

Space applications

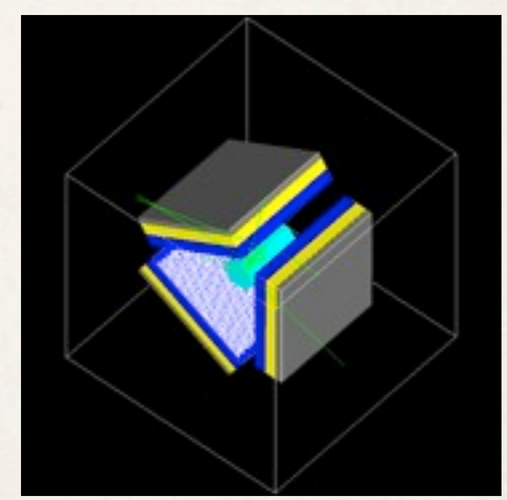
Linear accelerator



LSO/LuYAP ClearPET prototype design



3-head SPECT



Some capability of Geant4

- Transportation of a particle 'step-by-step' taking into account all the possible interactions with materials and fields
- The transport ends if the particle
  - reaches a zero kinetic energy
  - disappears in some interaction
  - reaches the end of the simulation volume
- Geant4 permits to the User to access the transportation process and retrieve the results (USER ACTIONS)

What Geant4 offers to start a simulation

- Multiple choices to describe the geometry
  - Basic geometry shapes
  - Representation by surface planes
  - Boolean operations, etc.
- Many possibilities to define elements and materials
- A huge variety of particles
  - from standard to unstable also including ions

Minimum software requirements

- C++
  - A basic knowledge is required being Geant4 a collection of C++ libraries
  - It is complex but also no C++ experts can use Geant4
- Object oriented technology (OO)
  - Very basic knowledge
  - Expertise needed for the development of complex applications

Geant4: 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 the Geant4 kernel to control your simulation
  - To visualise your simulation set-up and particles
  - To produce histograms, tuples, etc. to be further analysed

User mandatory classes

- Mandatory classes in any Geant4 User Application
  - **G4VUserDetectorConstruction**
    describes the experimental set-up
  - **G4VUserPhysicsList**
    selects the physics you want to activate
  - **G4VUserPrimaryGeneratorAction**
    generates primary events
- Sets of PHYSICS MODELS in a very big energy range (250 eV – 20 TeV)
  - Electromagnetic
  - Decay processes
  - Hadronic elastic
  - Hadronic inelastic

User classes

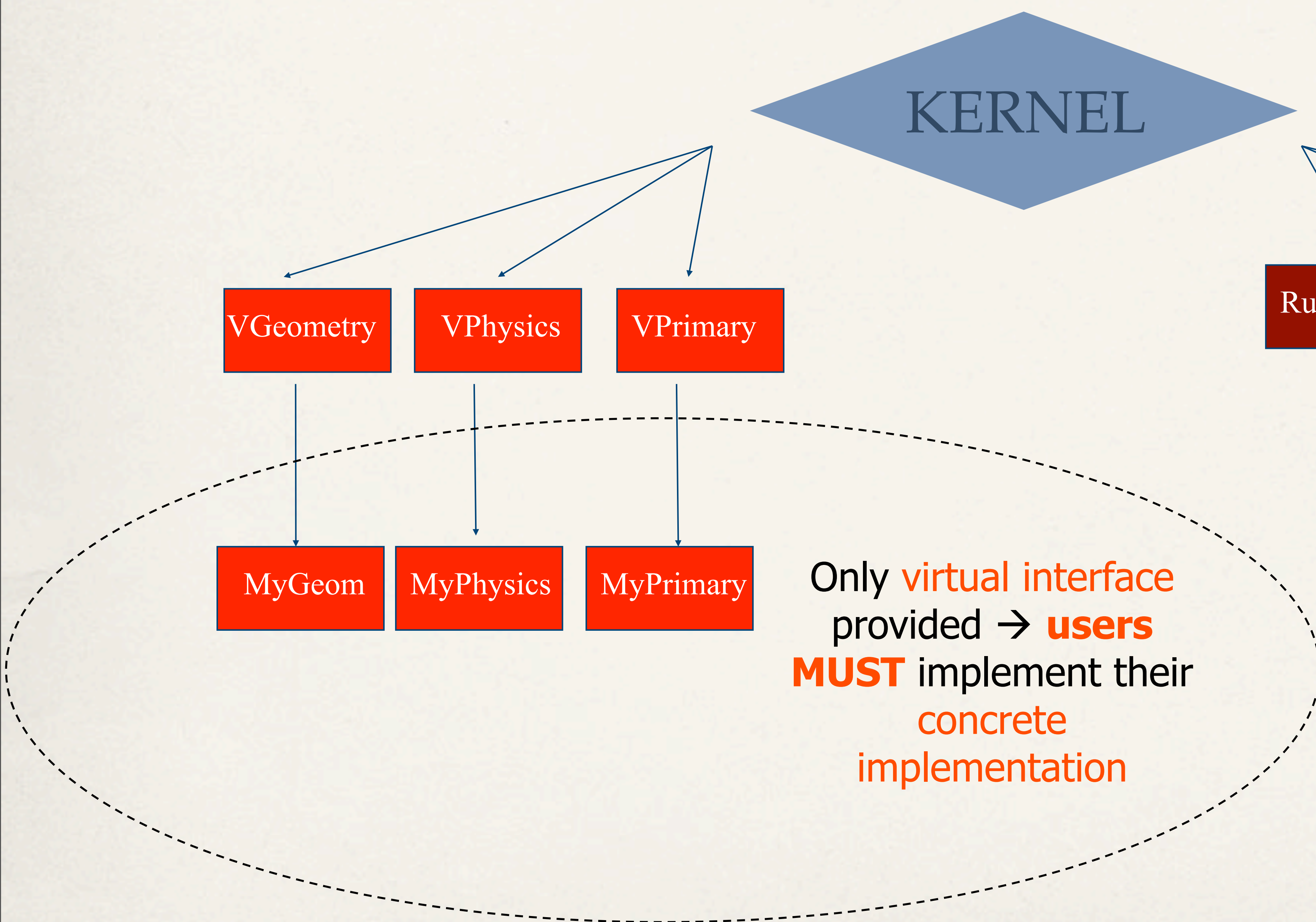
- ACTION CLASSES (Invoked during the execution of the loop)
  - G4VUserPrimaryGeneratorAction
  - G4UserRunAction
  - G4UserEventAction
  - G4UserTrackingAction
  - G4UserSteppingAction

cirrone@lns.infn.it,http://workgroup.lngs.infn.it/geant4lns

Geant4 general concept

KERNEL

Geant4 general concept



cirrone@lns.infn.it,http://workgroup.lngs.infn.it/geant4lns

The main() file

- Geant4 does not provide a main() file
  - Geant4 is a toolkit!
  - The main() is part of the User application
- In his/her main(), the user must:
  - Construct the `G4RunManager`
  - Notify the `G4RunManager` the mandatory user classes derived from:
    - `runManager -> SetUserInitialization(new MyApplicationDetectorConstruction)`

cirrone@lns.infn.it,http://workgroup.lngs.infn.it/geant4lns

The main() file

- The user MAY define in his/her main():
  - Optional user action classes
  - VisManager, (G)UI session
- The User has also to take care of retrieve and save the relevant information from the simulation (Geant4 will not do that by default)
- Do not forget to delete the `G4RunManager` at the end

An example of main()

```
{
 // Construct the default run manager
  G4RunManager* runManager = new G4RunManager;
 // Set mandatory user initialization classes
  MyDetectorConstruction* detector = new MyDetectorConstruction;
  runManager->SetUserInitialization(detector);
  MyPhysicsList* physicsList = new MyPhysicsList;
  runManager->SetUserInitialization(myPhysicsList);

  // Set mandatory user action classes
  runManager->SetUserAction(new MyPrimaryGeneratorAction);

  // Set optional user action classes
  MyEventAction* eventAction = new MyEventAction();
   runManager->SetUserAction(eventAction);
  MyRunAction* runAction = new MyRunAction();
   runManager->SetUserAction(runAction);
}
```

Select physics processes

- Geant4 doesn't have any default particles or processes
- Derive your own concrete class from the G4VUserPhysicsList abstract base class
  - Define all necessary particles
  - Define all necessary processes and assign them to proper particles
  - Define particles production threshold (in terms of range)
- Methods of G4VUserPhysicsList:
  - ContructParticles()
  - ConstructProcesses()
  - SetCuts()

→ Must be implemented by the user in his/her concrete class

Optional user classes

- Five concrete base classes, whose virtual member functions the User may override, to gain control of the simulation at various stages
  - G4User**Run**Action
  - G4User**Event**Action
  - G4User**Tracking**Action
  - G4User**Stacking**Action
  - G4User**Stepping**Action
- The User may implement any function he/she desires
  - E.g. one may want to perform some action at each step
- Objects of user action classes must be registered with G4RunManager
  - `runMnager -> SetUserAction(new MyEventActionClass);`

Methods of User classes

G4UserRunAction

- `BeginOfRunAction(const G4Run*)` // book histos
- `EndOfRunAction(const G4Run*)` //store histos

G4UserEventAction

- `BeginOfEventAction(const G4Event*)` //initialize event
- `EndOfEventAction (const G4Event*)` // analyze event

G4UserTrackingAction

//decide to store/not store a given track

- `PreUserTrackingAction(const G4Track*)`
- `PostUserTrackingAction(const G4Track*)`

Methods of User classes

G4UserSteppingAction

- `UserSteppingAction(const G4Step*)`

  //kill, suspend, pospone the track, draw the step, …

G4UserStackingAction

- `PrepareNewEvent()` //reset priority control
- `ClassifyNewTrack(const G4Track*)`

  // Invoked when a new track is registered (e.g. kill, pospone)
- `NewStage()` **//** Invoked when the Urgent stack becomes empty (re-classify, abort event)

cirrone@lns.infn.it,http://workgroup.lngs.infn.it/geant4lns

Optional: (G)UI

- In your main(), taking into account your computer environment, instantiate a **G4UISession** provided by Geant4 and invoke its **SessionStart()** method:
  - `mysession -> SessionStart();`
- Geant4 provides:
  - G4UIterminal;
  - csh or tcsh like shell
  - G4UIBatch
  - Bach job with macro files

Optional: visualisation

- In your main(), taking into account your computer environment, instantiate a **G4VisExecutive** and invoke its **Initialize()** method
- Geant4 provides interfaces to various graphics drivers:
  - Dawn
  - Wired
  - RayTracer
  - OpenGL
  - OpenInventor
  - VRML
  - ....

General recipe for novice users

- Design your application .... requires preliminary thinking (what is supposed to do?)
- Create your derived mandatory user classes
  - `MyDetectorConstruction`
  - `MyPhysicsList`
  - `MyPrimaryGeneratorAction`
- Create optional derived user action classes
  - `MyUserRunAction, MyUserEventAction`
- Create your main() file
  - Instantiate `G4RunManager`
  - Notify the RunManager of your mandatory and optional user classes
  - Optionally initialise your favourite User Interface and Visualisation

General recipe for novice users

- Design your application .... requires preliminary thinking (what is supposed to do?)
- Create your derived mandatory user classes
  - `MyDetectorConstruction`
  - `MyPhysicsList`
  - `MyPrimaryGeneratorAction`
- Create optional derived user action classes
  - `MyUserRunAction, MyUserEventAction`
- Create your main() file
  - Instantiate `G4RunManager`
  - Notify the RunManager of your mandatory and optional user classes
  - Optionally initialise your favourite User Interface and Visualisation

Experienced users may do much more, but the conceptual process is still the same…

Geant4 download and installation

- You can download the compiled libraries of Geant4 but the compilation in your computer is strongly suggested
- Download the source file from the Geant4 web site
- Download all the external data libraries
- Download and install the CLHEP libraries
- Two way to proceed:
    - Use the configure script (./configure) to define the enviroment variables
    - Define in a text file the the correct environment variables before to start with the Geant4 compilation