A Geant4 introduction

IX Seminar on , May 28 - June 2, 2012 - Porto Conte, Italy

Where you can find this material?

- Pablo Cirrone, Francesco Romano,
  researchers at the Italian National Institute for Nuclear Physics
  pablo.cirrone@lns.infn.it francesco.romano@lns.infn.it

- We (as Geant4 Collaboration and as other Geant4 Members) regularly offer tutorials and schools - see Official Geant4 pages

- The official Geant4 web pages
  www.cern.ch/geant4

- The Italian Geant4 group:
  http://geant4.lngs.infn.it/

- Fews concepts on Monte Carlo approach
- Geant4 and the Geant4 Collaboration
- Basic concepts and capabilities of Geant4
- Installation tips
- Example of an application (Geometry, physics, tracking, etc)

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:

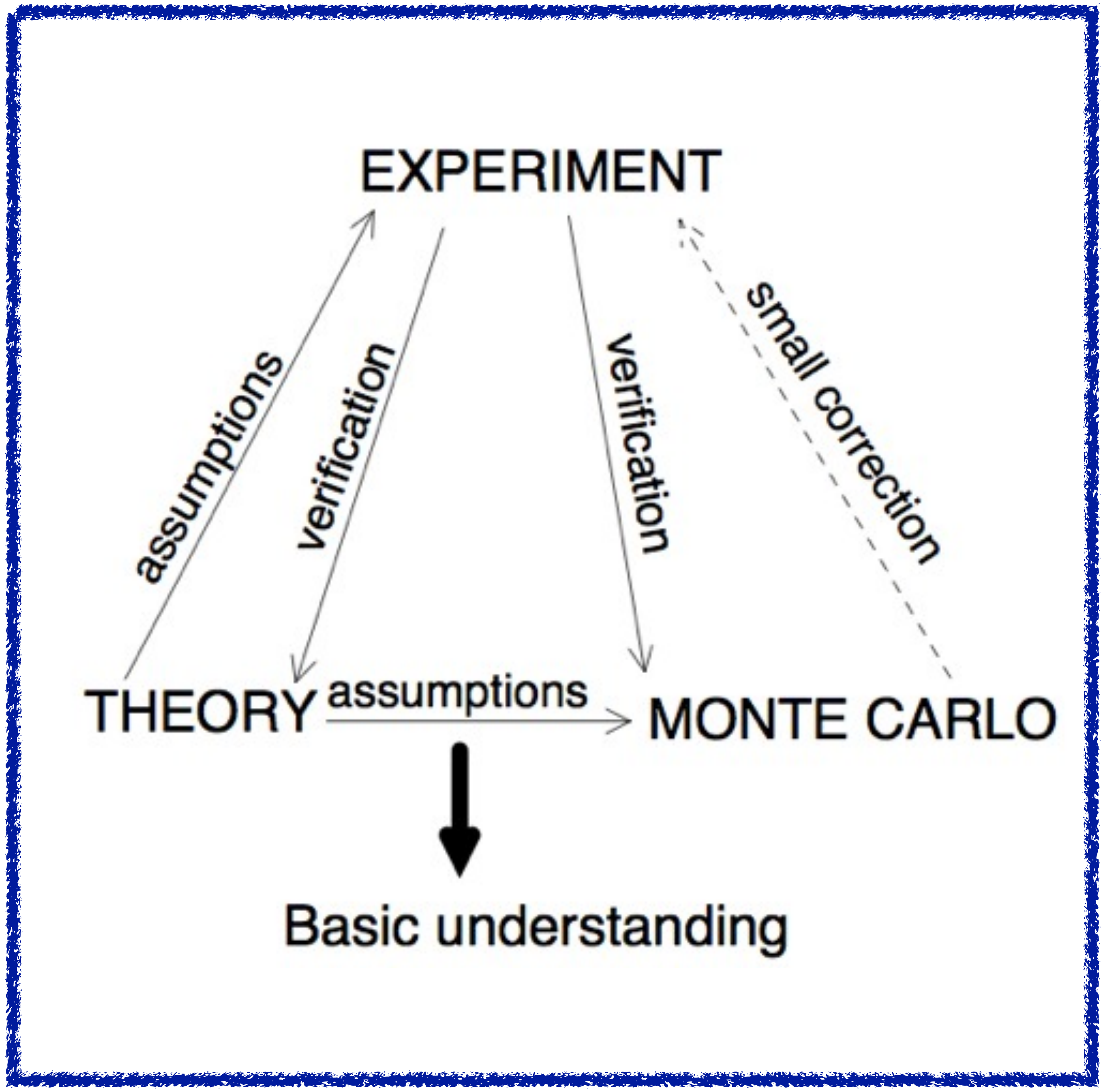
we must construct a big number of this equations, using

different random numbers, and

estimate the parameter and its variance"

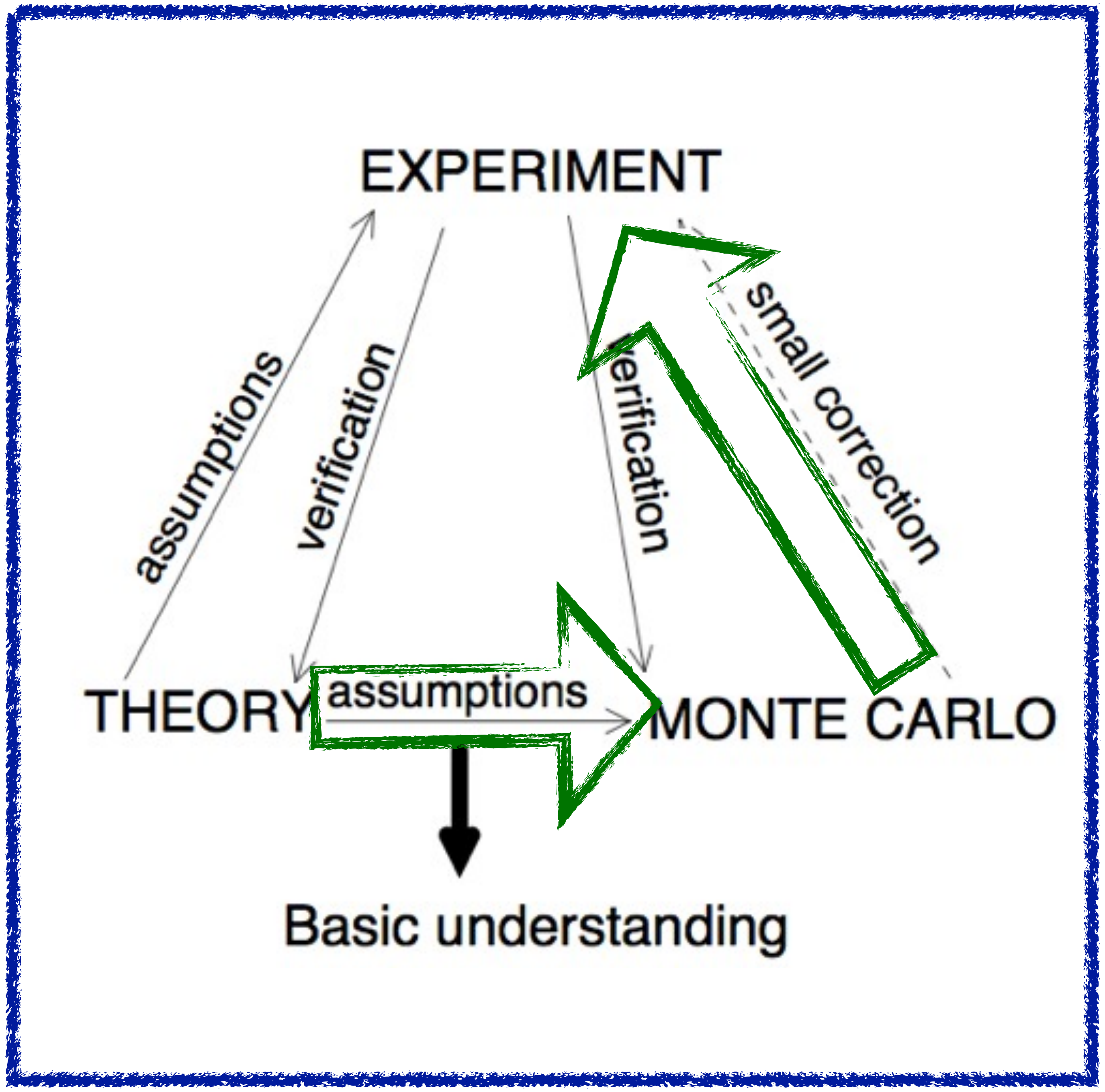
A. F. Bielajew, 2001

The Monte Carlo method



- Monte Carlo helps
  - To verify a theory if physics models are in development
  - To develop or verify an experiment in the other case

The Monte Carlo method



- In particle transport, if particles interaction models are known, MC can be used to calculate the parameters of the motion equations in a given configuration

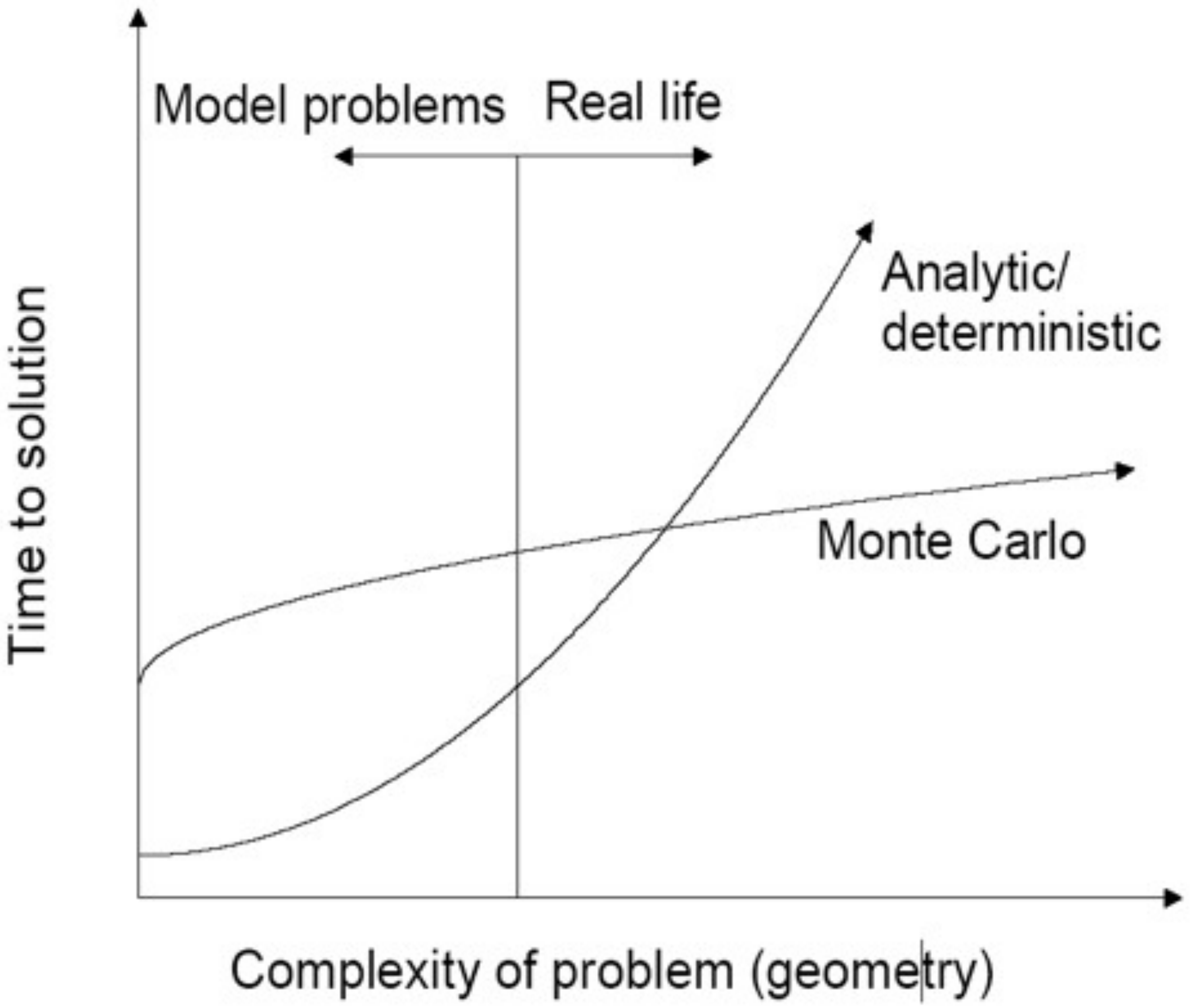
The Monte Carlo method

- 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
  .... sometime necessary and
  ....with many advantages

The Monte Carlo method

Monte Carlo *vs* deterministic/analytic methods



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

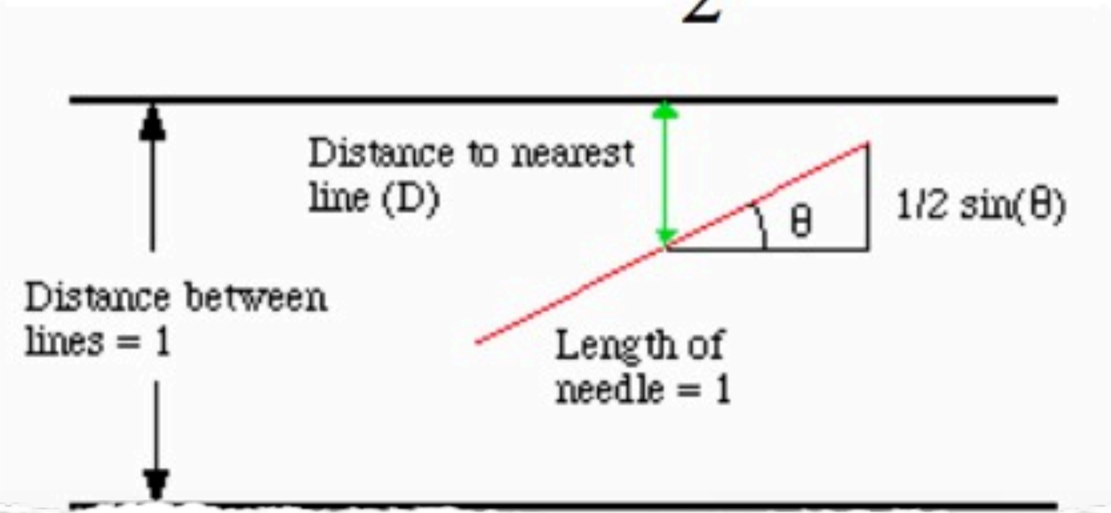
We need a computer for a Monte Carlo calculation?

The Buffon experiment:
The Monte Carlo approach for the π estimation

Two variables: θ and D

$$0 \leq \vartheta \leq \pi$$

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

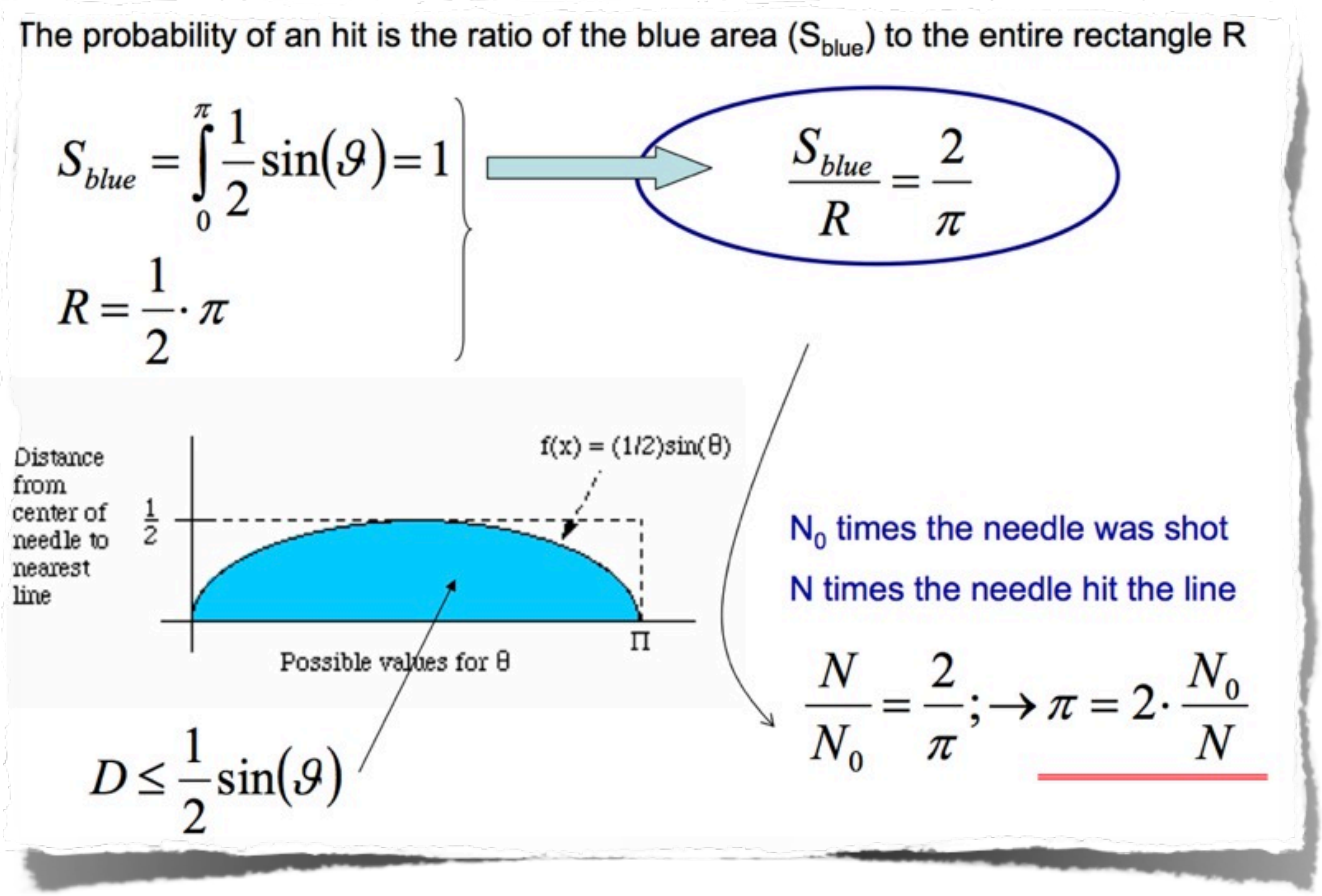


Georges Louis Leclerc
Comte de Buffon
(07.09.1707.-16.04.1788.)

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

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

The Buffon experiment:
The Monte Carlo approach for the π estimation



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

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

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

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

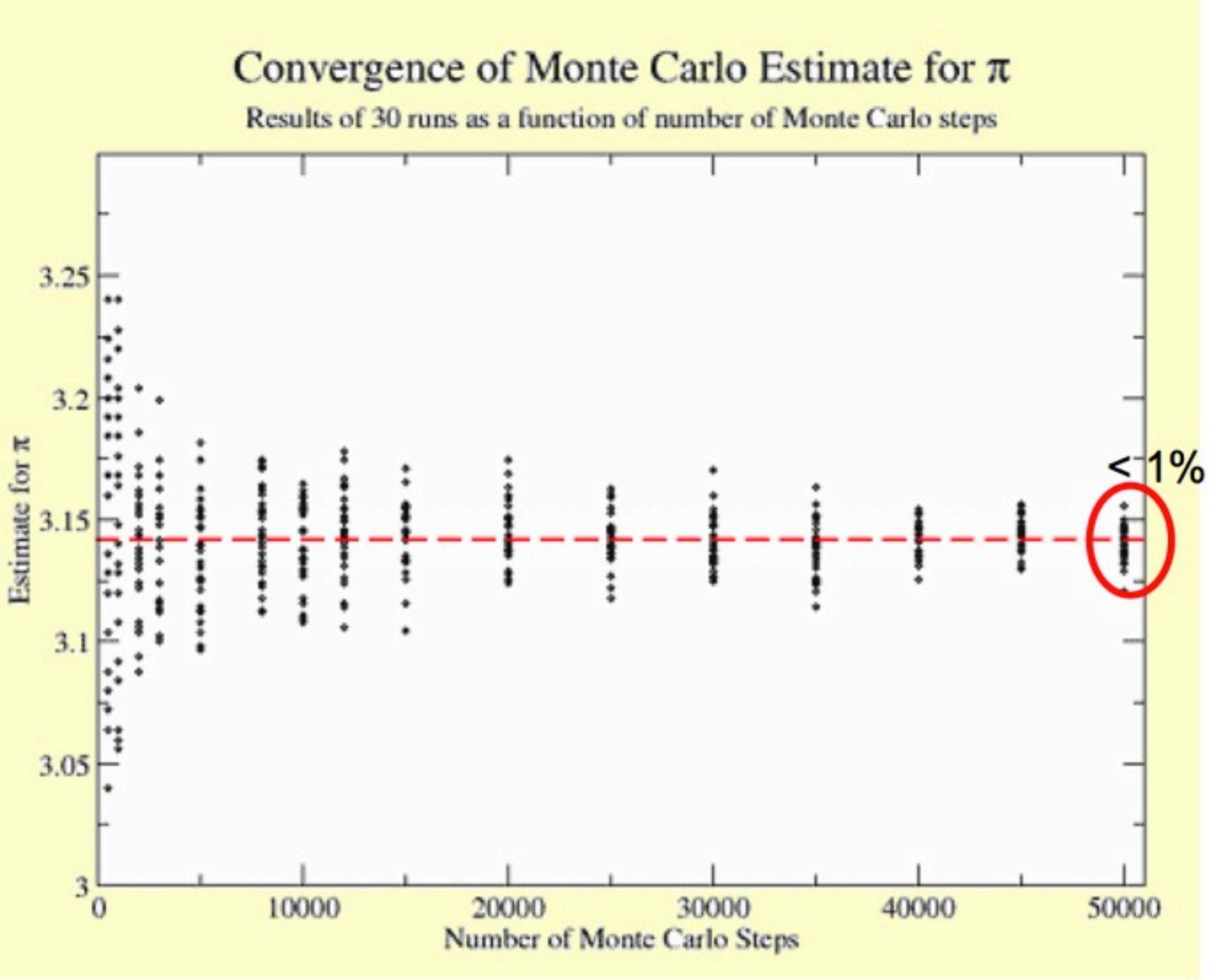
$$D \le \frac{1}{2} \sin(\vartheta)$$

N_0 times the needle was shot

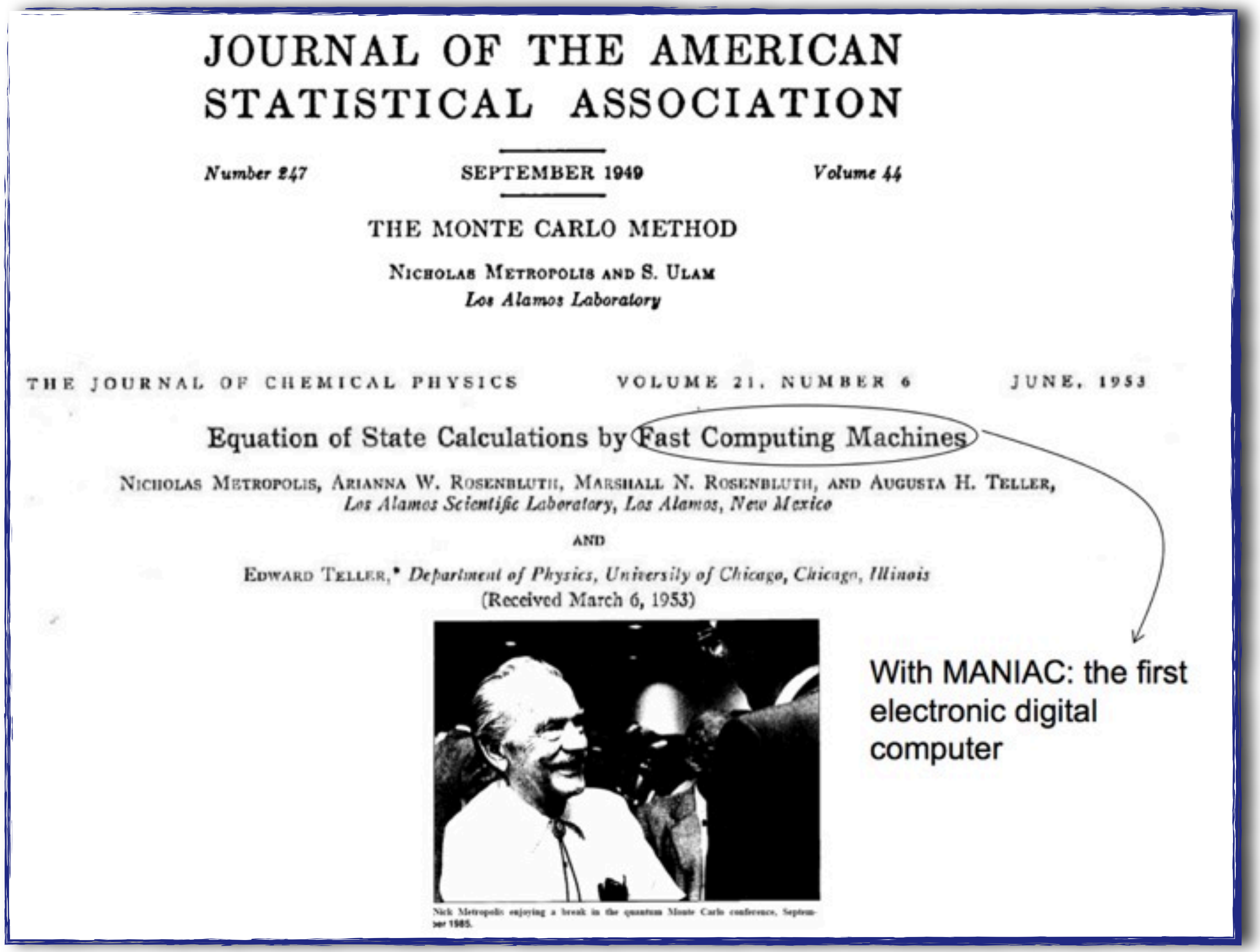
N times the needle hit the line

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

The Buffon experiment:
The Monte Carlo approach for the π estimation



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, September 1985.

With MANIAC: the first electronic digital computer

Fermi's work on pion-proton phase shift analysis

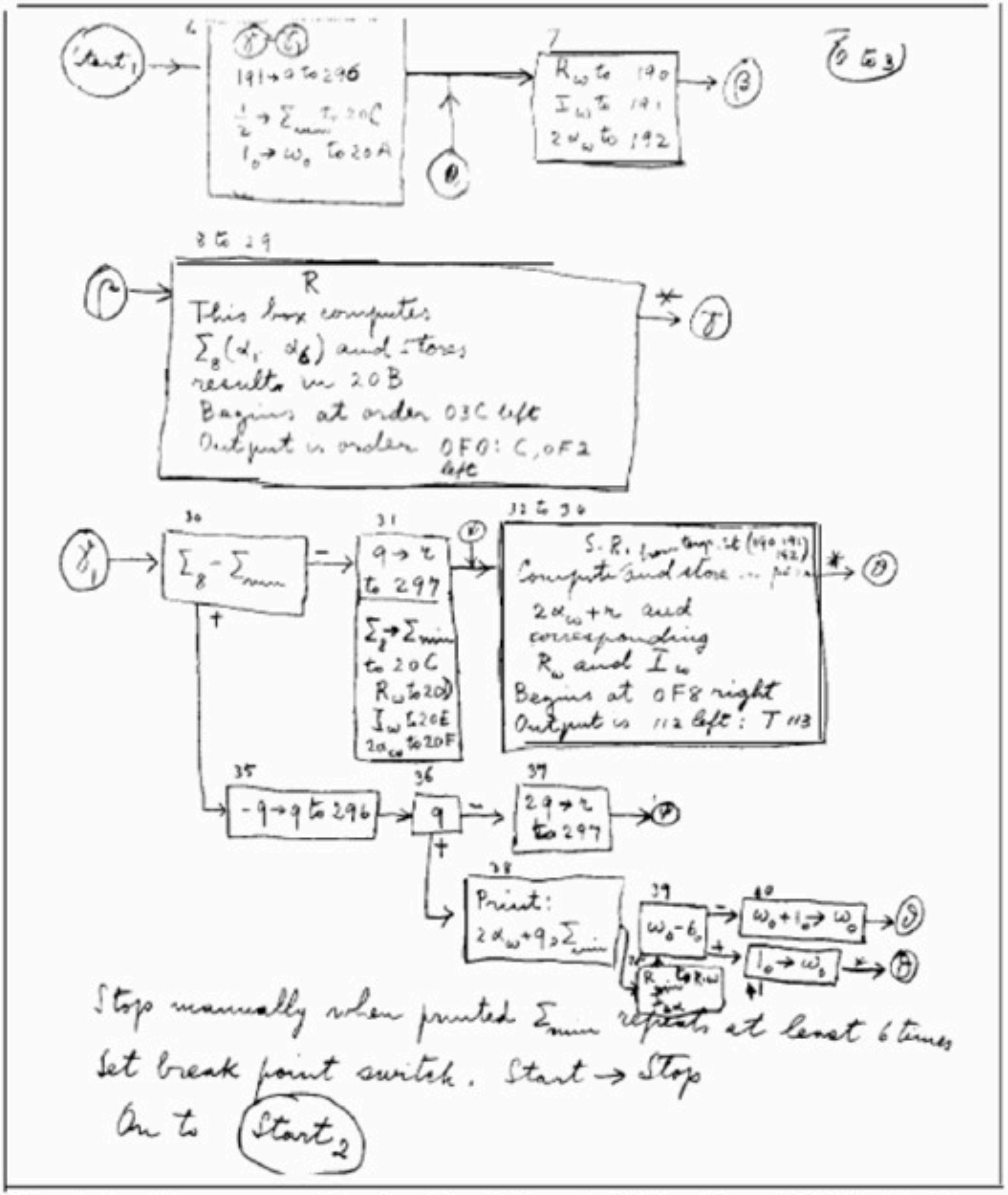
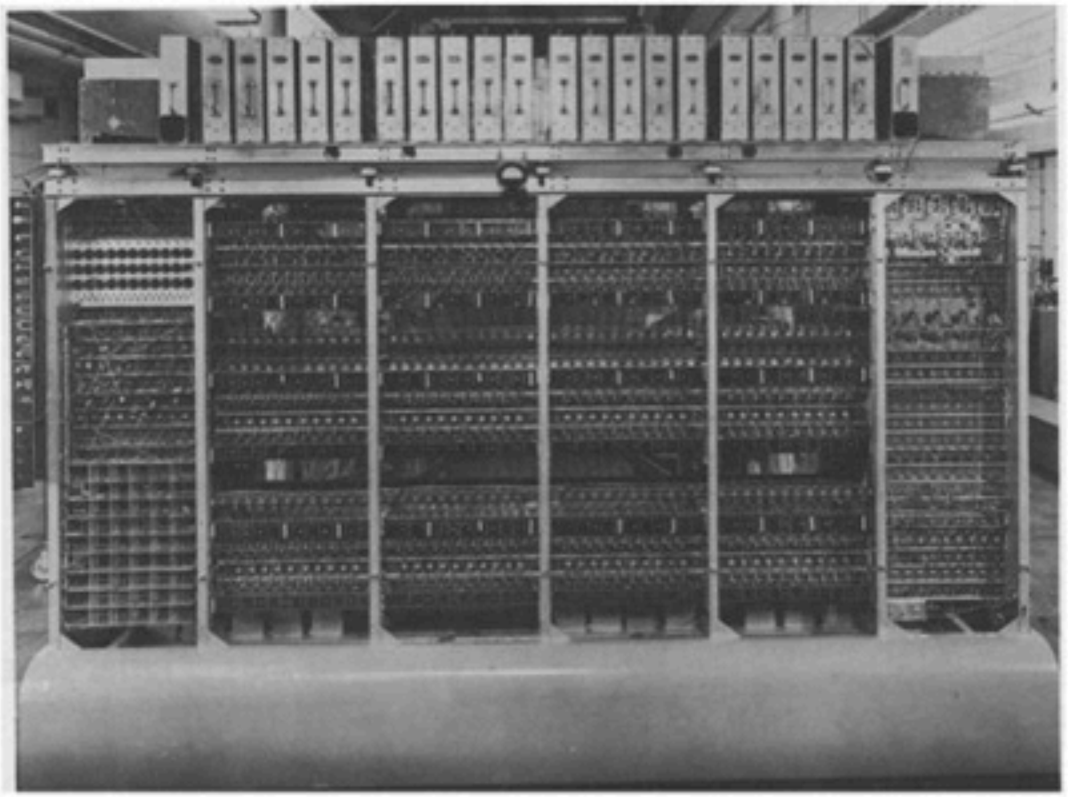
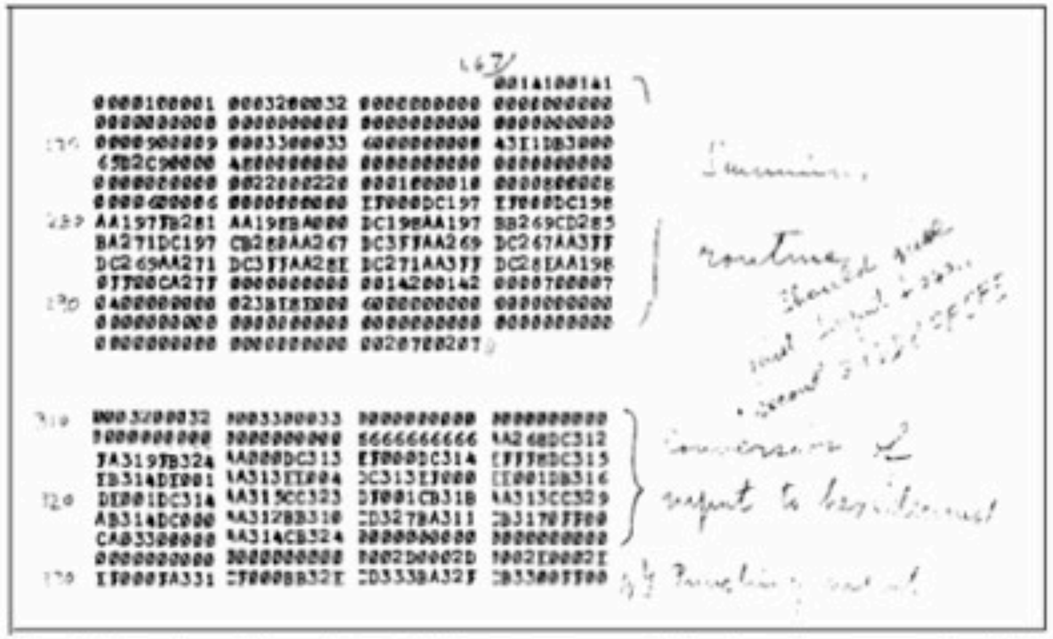


Fig. 4. A subprogram written by Fermi for calculating phase shifts by finding a minimum chi-squared in a fit to the data.





```
                                            167
                                            001A100141
    0000100001 0003200032 0000000000 0000000000
    0000000000 0000000000 0000000000 0000000000
170 0000900009 0003300033 6000000000 43I1DB3000
    65D2C90000 4E00000000 0000000000 0000000000
    0000000000 0022000220 0001000010 0000800008
    0000600006 0000000000 IJ000DC197 IJ000DC198
180 AA197JB281 AA198BA000 DC198AA197 BB269CD285
    BA271DC197 CB280AA267 DC3JJAA269 DC267AA3JJ
    DC269AA271 DC3JJAA28E DC271AA3JJ DC28EAA198
    0JJ00CA2JJ 0000000000 0014200142 0000700007
190 0400000000 023BIBI000 6000000000 0000000000
    0000000000 0000000000 0000000000 0000000000
    0000000000 0000000000 0020700207

310 0003200032 0003300033 0000000000 0000000000
    0000000000 0000000000 6666666666 AA268DC312
    JA319JB324 AA000DC313 EJ000DC314 EJJJ0DC315
    ED314DI001 AA313EI004 DC313EJ000 EE001DB316
320 DE001DC314 AA315CC323 DJ001CB31B AA315CC329
    AB314DC000 AA317BB310 DD327BA311 DB3170JJ00
    CA03300000 AA314CB324 0000000000 0000000000
    0000000000 0000000000 0002D0002D 0002E0002E
330 EJ000JA331 EJ000BB32E DD333BA32J DB3300JJ00
```

Summing routine

Should put ... second ...

Conversion & output to hexadecimal

1/2 Punching ...

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

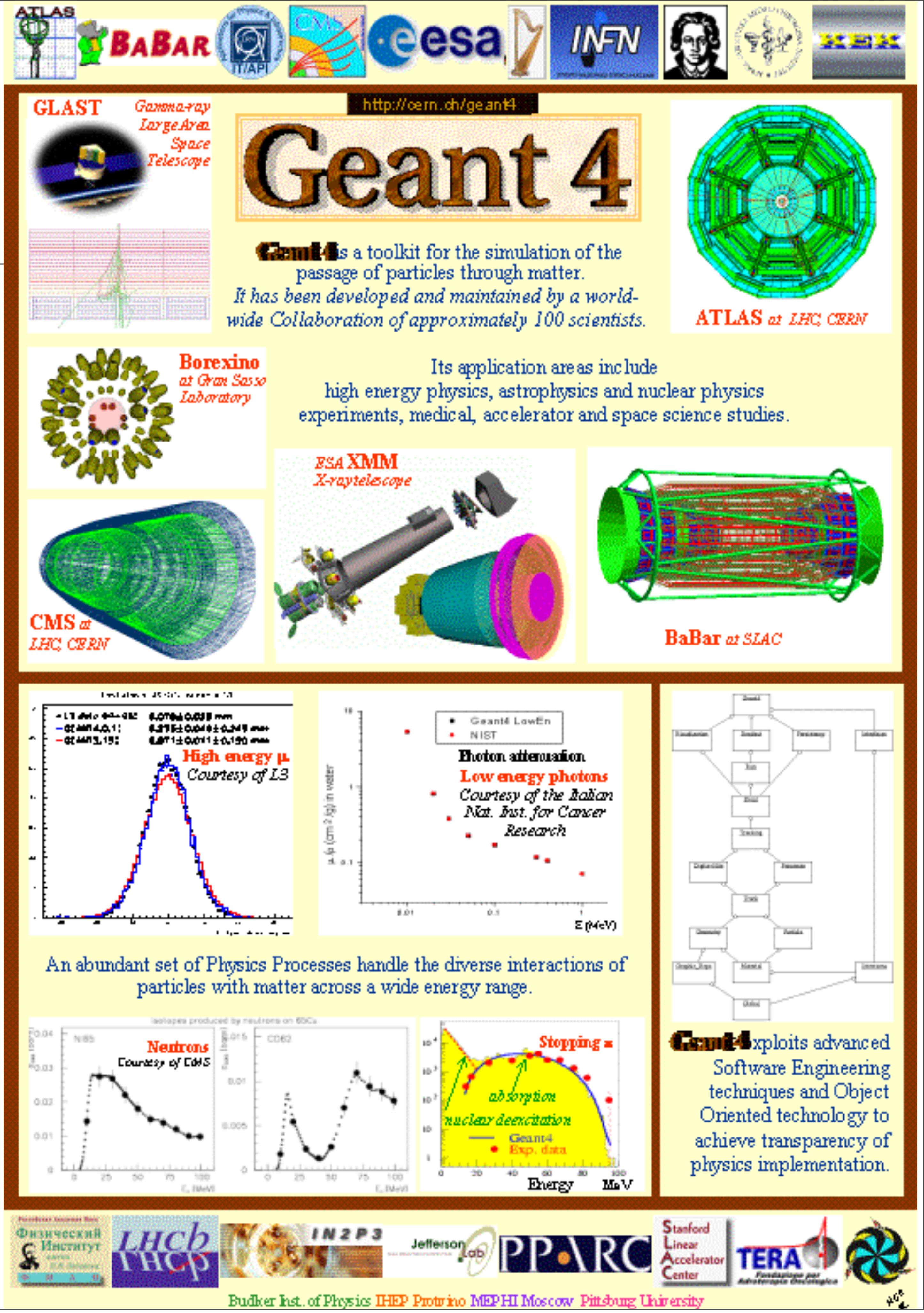
Monte Carlo codes on the market

- 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

- Fews concepts on Monte Carlo approach
- Geant4 and the Geant4 Collaboration
- Basic concepts and capabilities of Geant4
- Ongoing and future developments
- Installation tips
- Example of an application (Geometry, physics, tracking, etc)



http://cern.ch/geant4

Geant 4

Geant4 is a toolkit for the simulation of the passage of particles through matter.

It has been developed and maintained by a world-wide Collaboration of approximately 100 scientists.

Its application areas include high energy physics, astrophysics and nuclear physics experiments, medical, accelerator and space science studies.

An abundant set of Physics Processes handle the diverse interactions of particles with matter across a wide energy range.

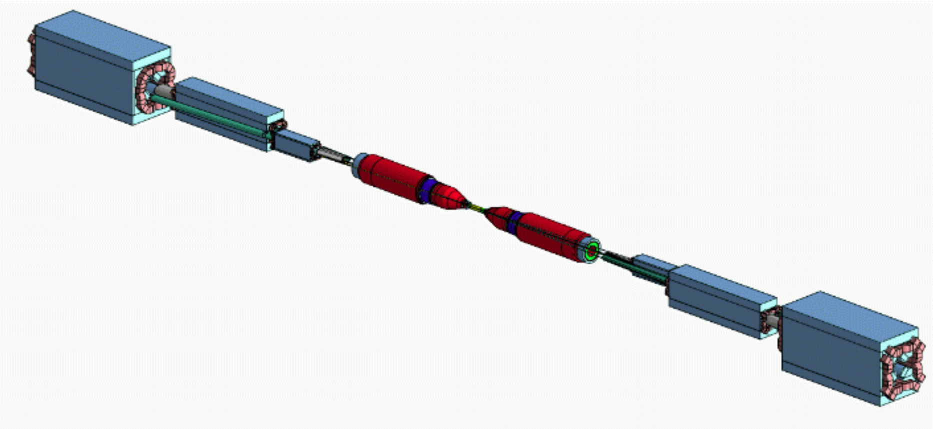
Geant4 exploits advanced Software Engineering techniques and Object Oriented technology to achieve transparency of physics implementation.

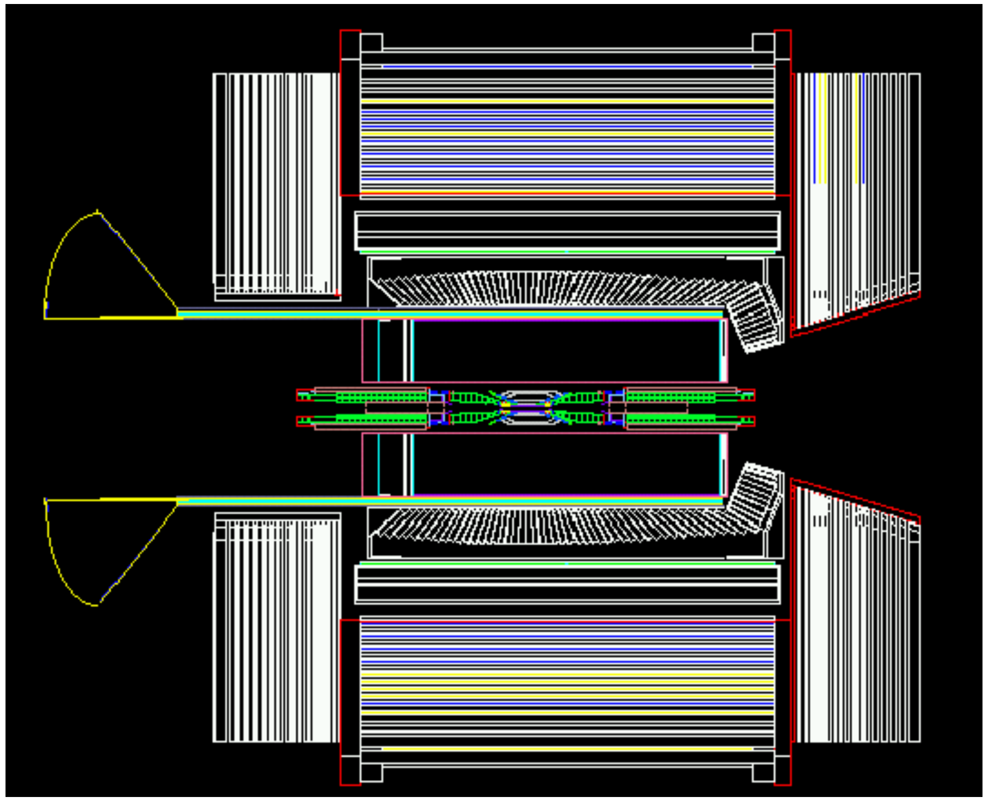
Geant4 - past and present

- Geant4 started at CHEP 1994 @ San Francisco
  - "Geant steps into the future", R Brun et al.
  - "Object oriented analysis and design of a Geant based detector simulator", K Amako et al
- Dec '94 - CERN RD44 project starts
- Apr '97 - First alpha release
- Jul '98 - First beta release
- Dec '98 - First Geant4 public release - version 1.0
- ....
- Dec 2nd, 2011 - Geant4 9.5 release ← Current version
  - Mar 27th, 2012 - Geant4 9.5-patch01 release
- We currently provide one public release every year

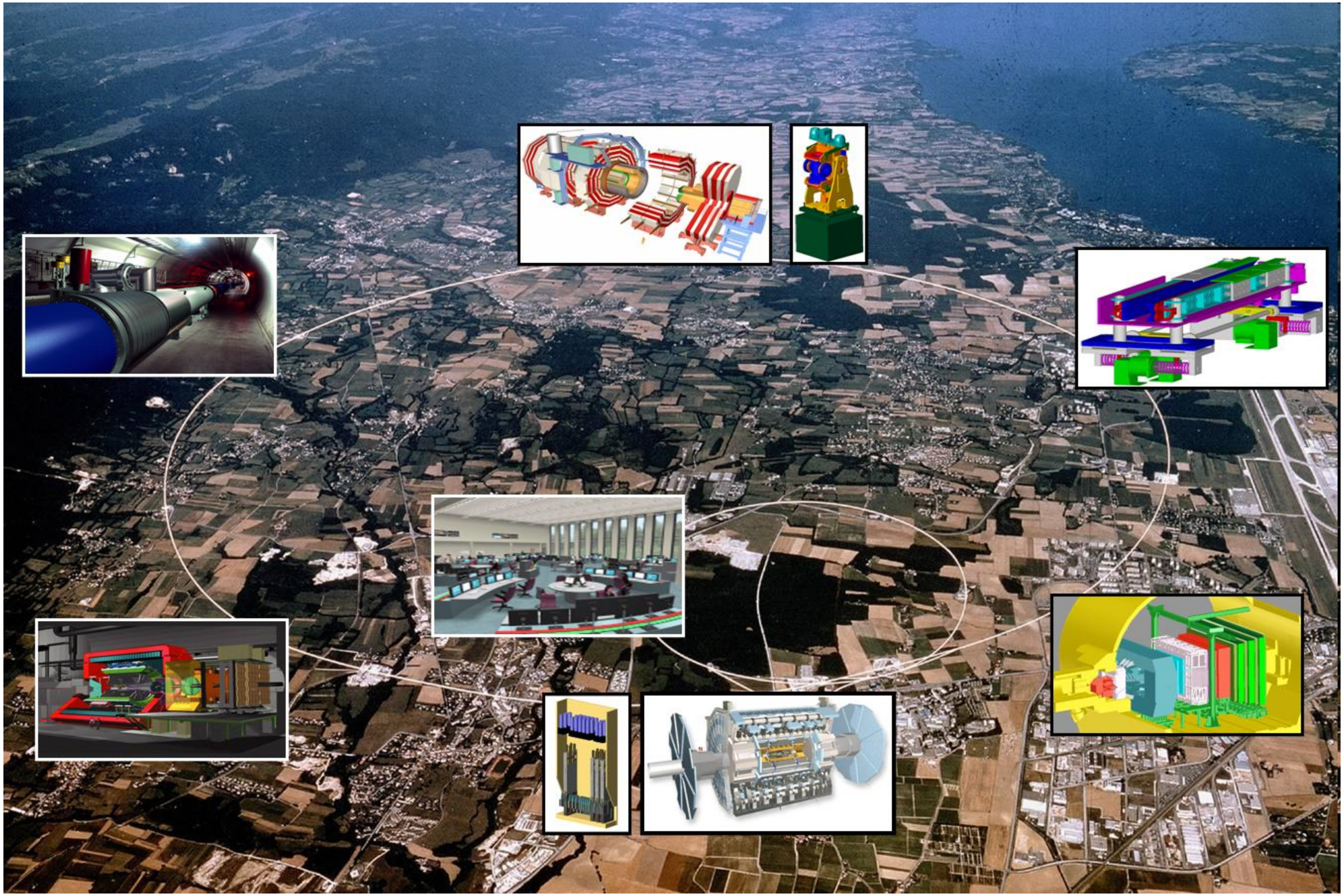
BaBar

- BaBar is the pioneer HEP experiment in use of OO technology and the first customer of Geant4
  - During the R&D phase of Geant4 a lot of evaluable feedbacks were provided
- BaBar started its simulation production in 2000 and had produced more than 10 bilion events at more than 20 sites in Europe and North America.

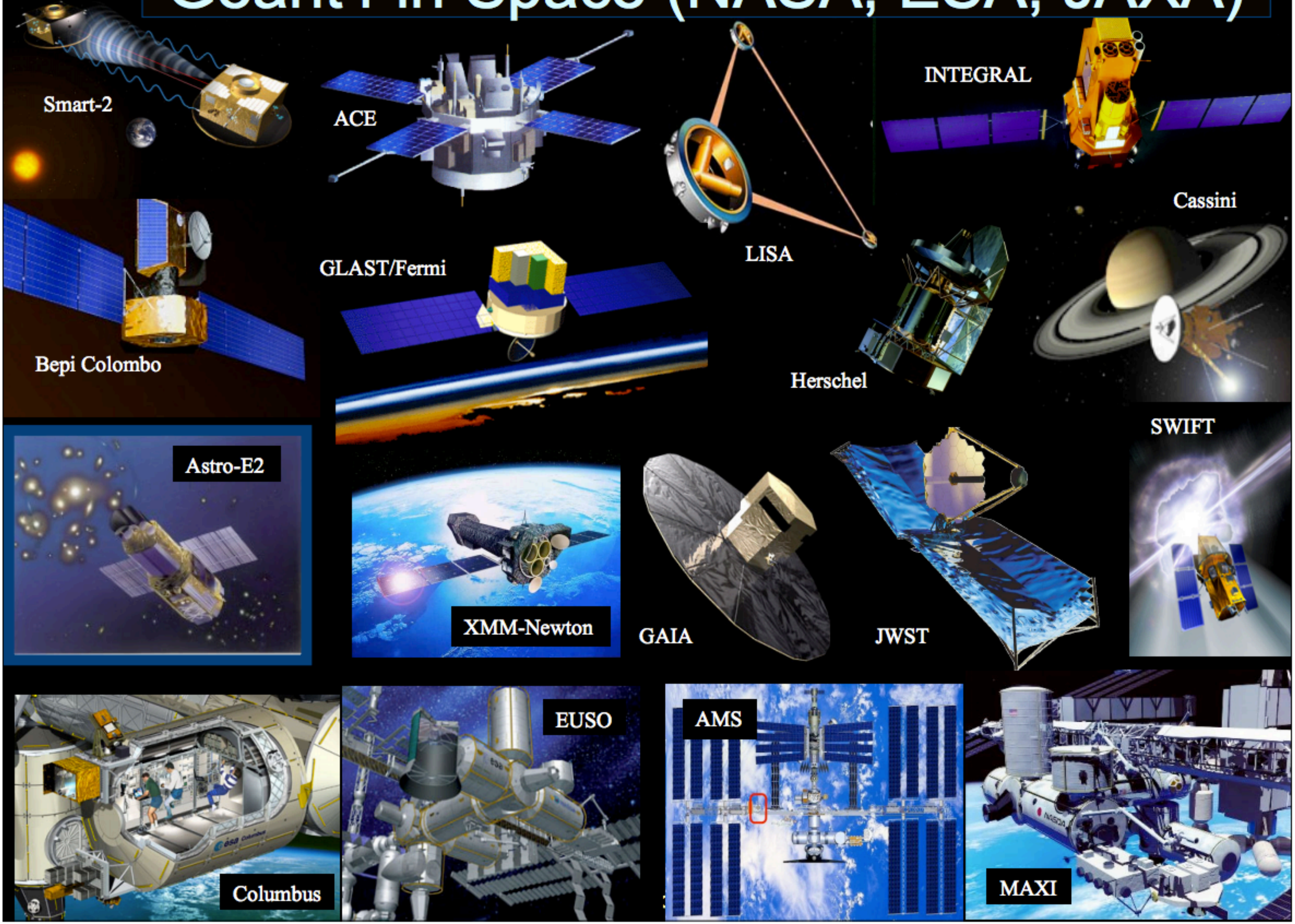




Large Hadron Collider @ CERN

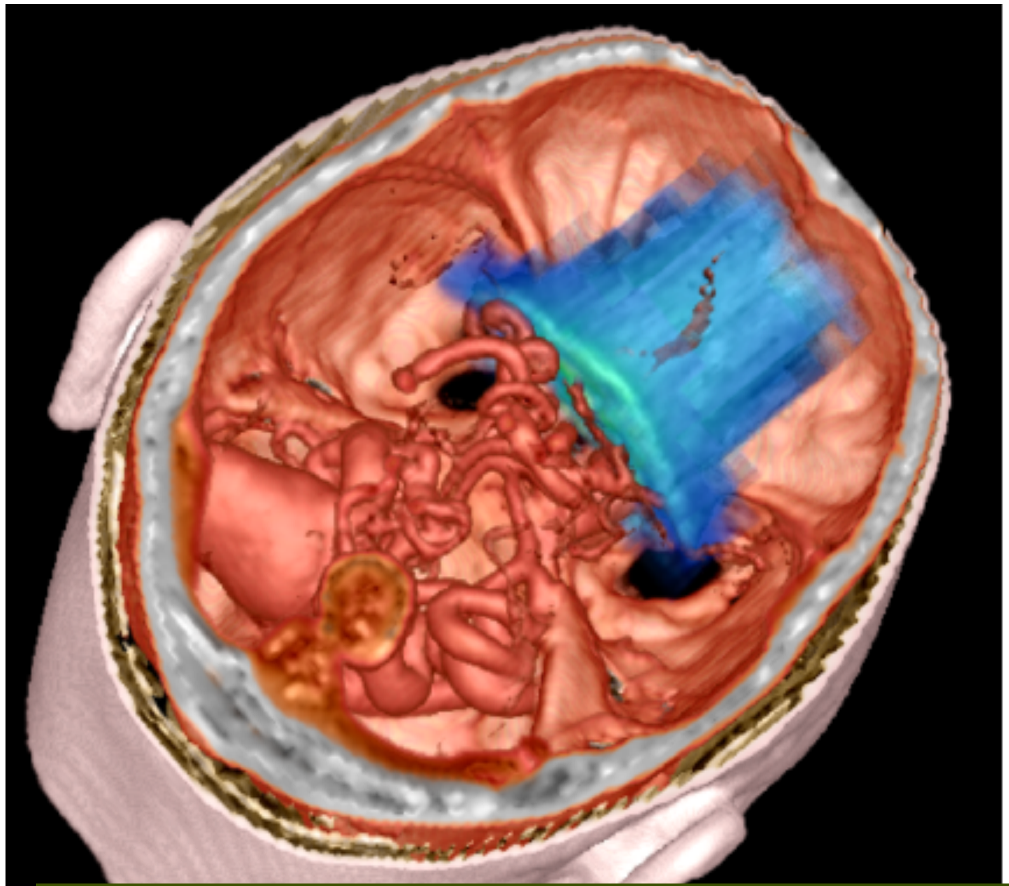


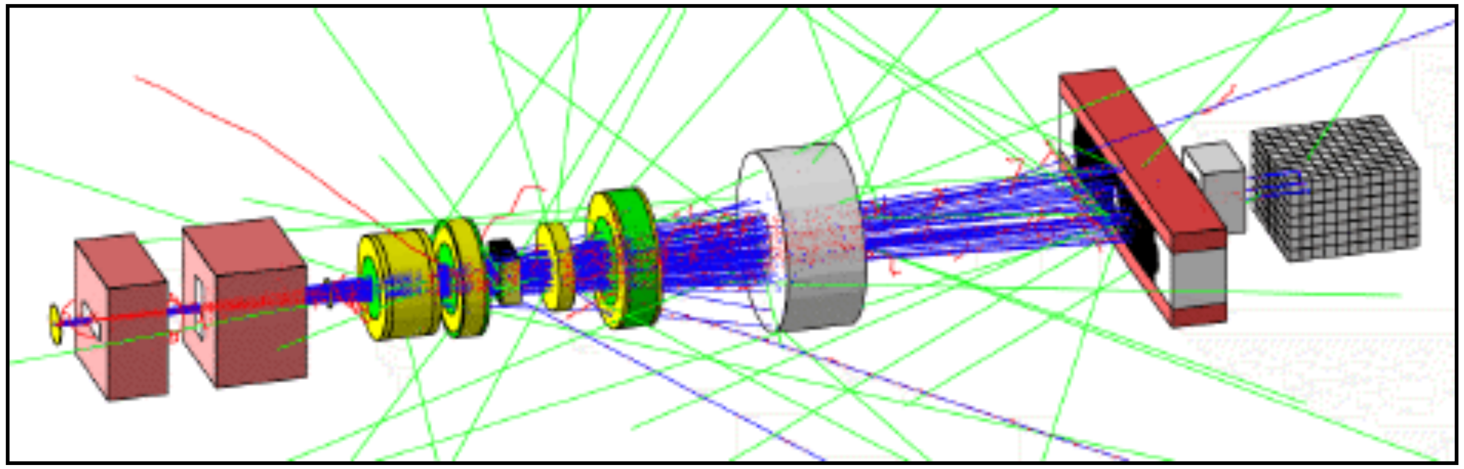
Geant4 in Space (NASA, ESA, JAXA)

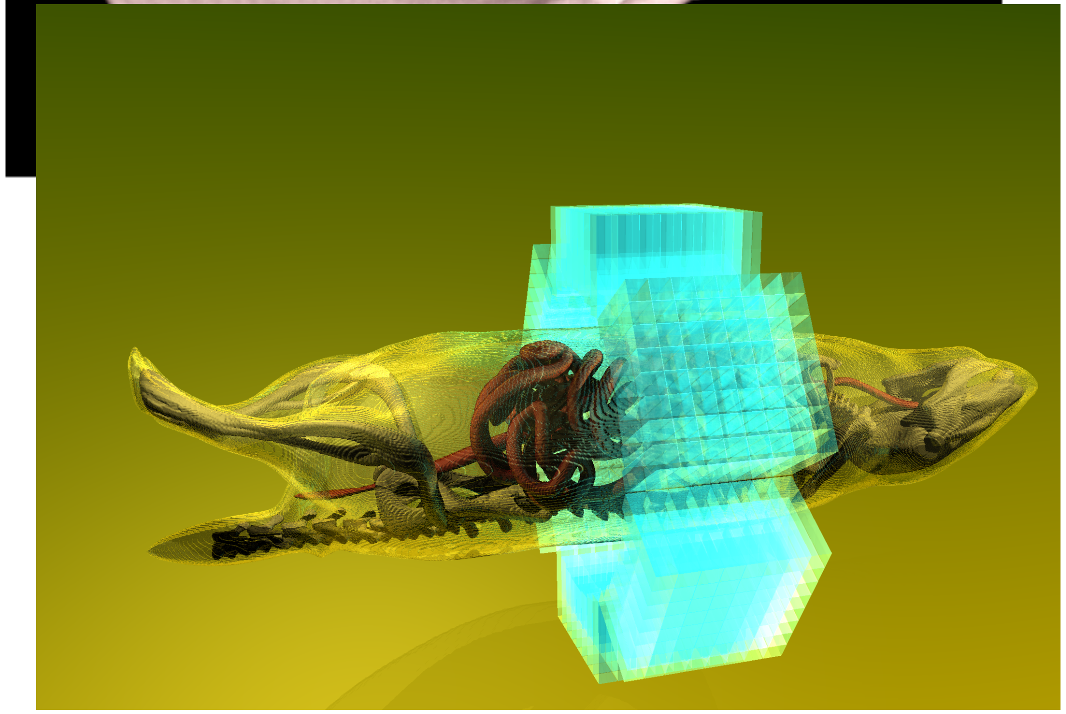


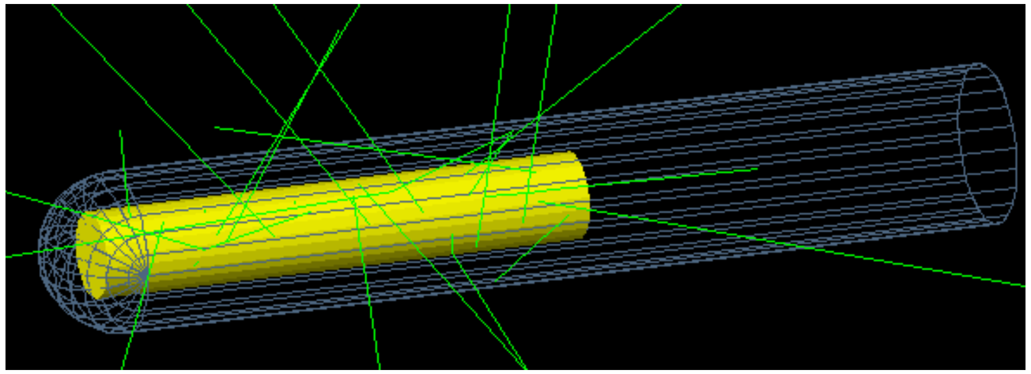
Geant4 and medical science

- Four major use cases
  - Beam therapy
  - Brachytherapy
  - Imaging
  - Irradiation study









- Fews concepts on Monte Carlo approach
- Geant4 and the Geant4 Collaboration
- Basic concepts and capabilities of Geant4
- Ongoing and future developments
- Installation tips
- Example of an application (Geometry, physics, tracking, etc)

GEANT4: www.cern.ch/Geant4

- C++ language
- Object Oriented
- Open Source
- Twice per year released
- 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

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
- Unix/Linux
  - These are the standard OSs for Geant4 and a basic knowledge is required
  - Principal shell commands
  - How to compile a program
  - How to install from source code

Supported and tested platforms

- Linux with gcc 4.1.2 or 4.3 and Intel icc 11 or 12
  - Tested on Scientific Linux CERN 5(SLC5) but also successfuly compiled on other Linux distributions, including Debian, Ubuntu and OpenSUSE

- Mac OSX 10.7 (Lion) and 10.6 (Snow Leopard) with gcc 4.2.1

- Windows 7 and XP with Visual Studio 9 and 10

Main Geant4 capabilities

- 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

Main Geant4 capabilities

•Geant4 permits to 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 particle reaches a sensitive detector;

- others ......

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

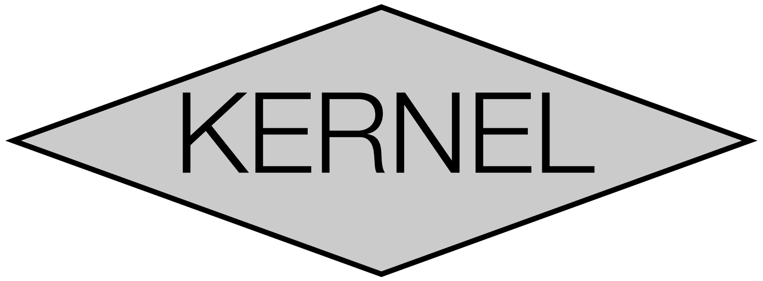
Geant4: basic concepts

- 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

Files composing a Geant4 app

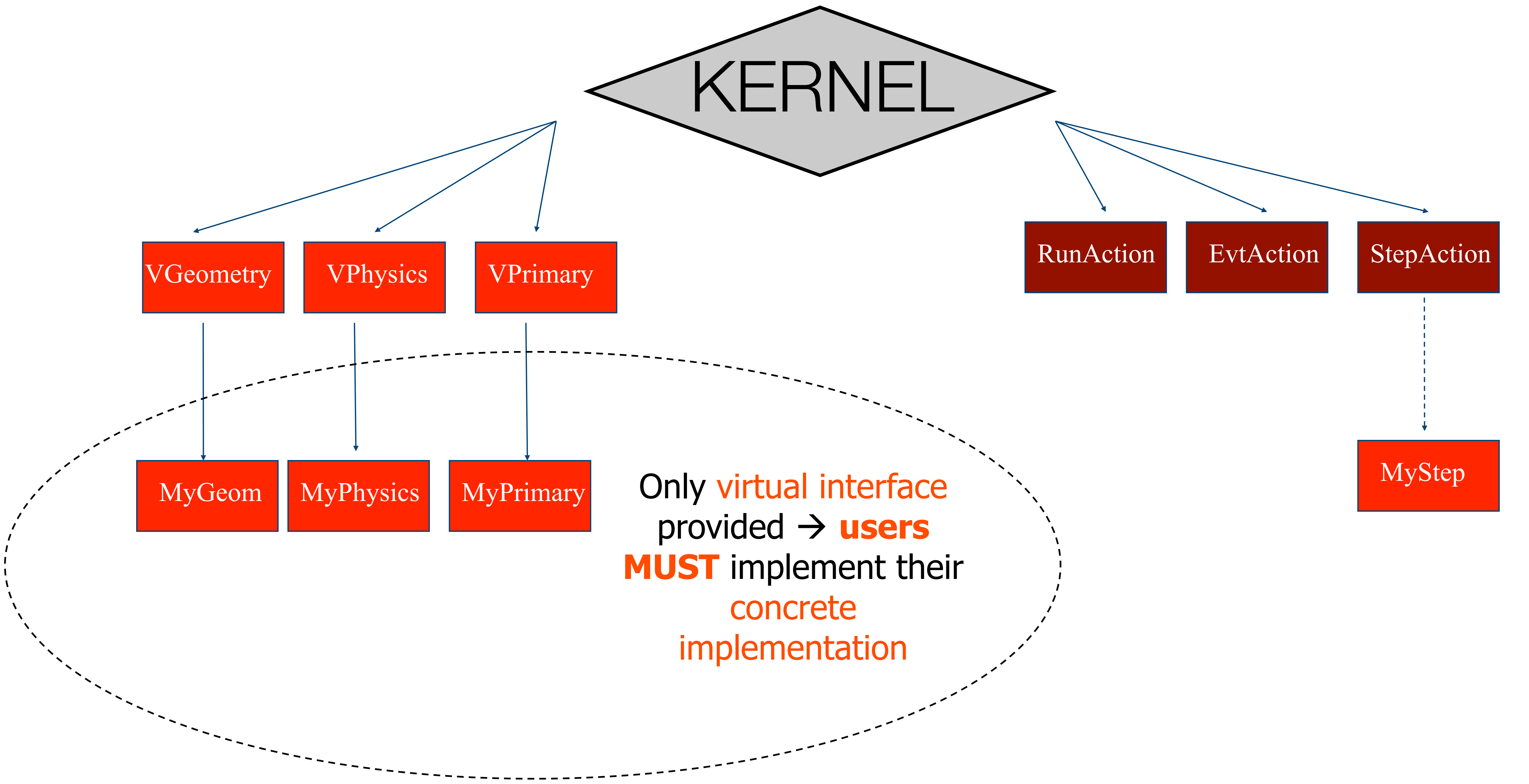
- Main() file
- Sources files (*.cc)
  - usually included in the /src folder
- Header files (*.hh)
  - usually included in the /include files
- Three couples of files are necessary (with the Main.cc ons)
  - The PrimarygeneratorAction (.cc and .hh)
  - The DetectorConstruction (.cc and .hh)
  - The PhysicsList (.cc and .hh)

Geant4 general concept



KERNEL

Geant4 general concept



Only virtual interface provided → **users MUST** implement their concrete implementation

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)`**

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);
}
```

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

User classes

- ACTION CLASSES (Invoked during the execution of the loop)

  - **G4VUserPrimaryGeneratorAction** - Mandatory

  - **G4UserRunAction** - Optional

  - **G4UserEventAction** - Optional

  - **G4UserTrackingAction** - Optional

  - **G4UserSteppingAction** - Optional

- 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 - continue

- **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)

Selection 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: (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
  - ....

SUMMARY: 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

SUMMARY: 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

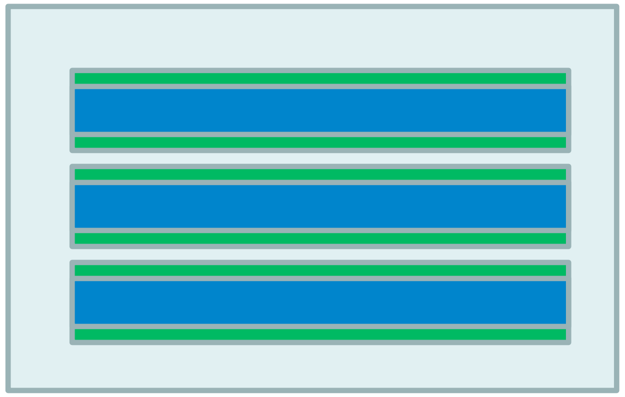
Experienced users may do much more,

but the conceptual process is still the same...

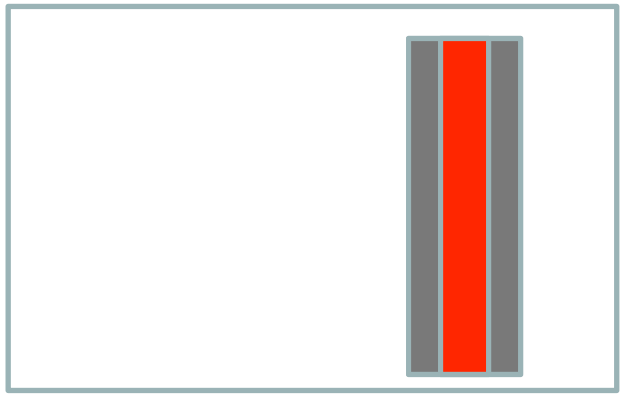
Recent and ongoing developments

Recent and ongoing developments

- Multi-threaded prototype (9.5 version)
- Layered mass geometries in parallel word (since 8.2 version)
- Parallel geometry may be stacked on top of mass geometry or other parallel word geometry, allowing a user to define more than one word with materials (and region/cuts)



Mass world

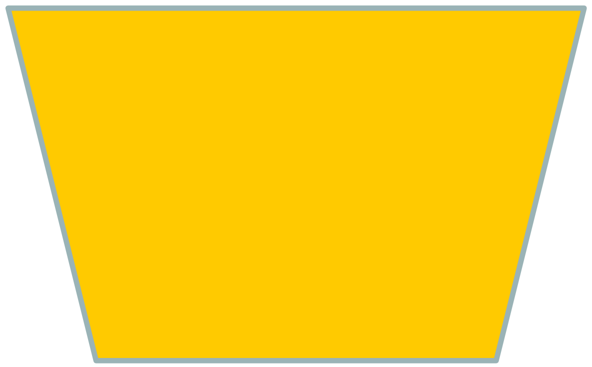


Parallel world

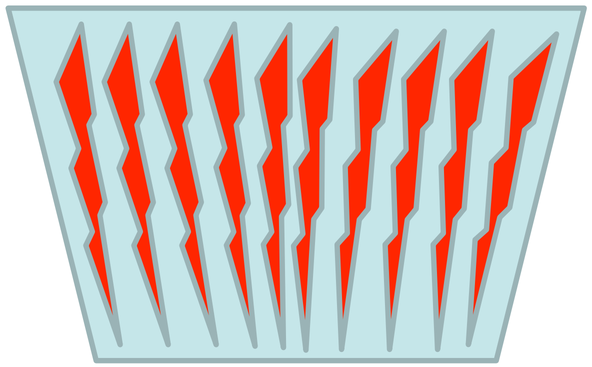
Parallel word - continue

- A parallel word can be associated to a limited number of particles
  - You may define geometries of different level of details for different particle types
  - Example: a sampling calorimeter: the mass word define only the crude geometry with averaged material, while a parallel word contains the detailed geometry.The materials in the detailed parallel word are associated with all particle except e+, e- and gamma

✓e+, e- and gamma do not see volume boundaries defined in the parallel word (i.e. steps won't be limited)

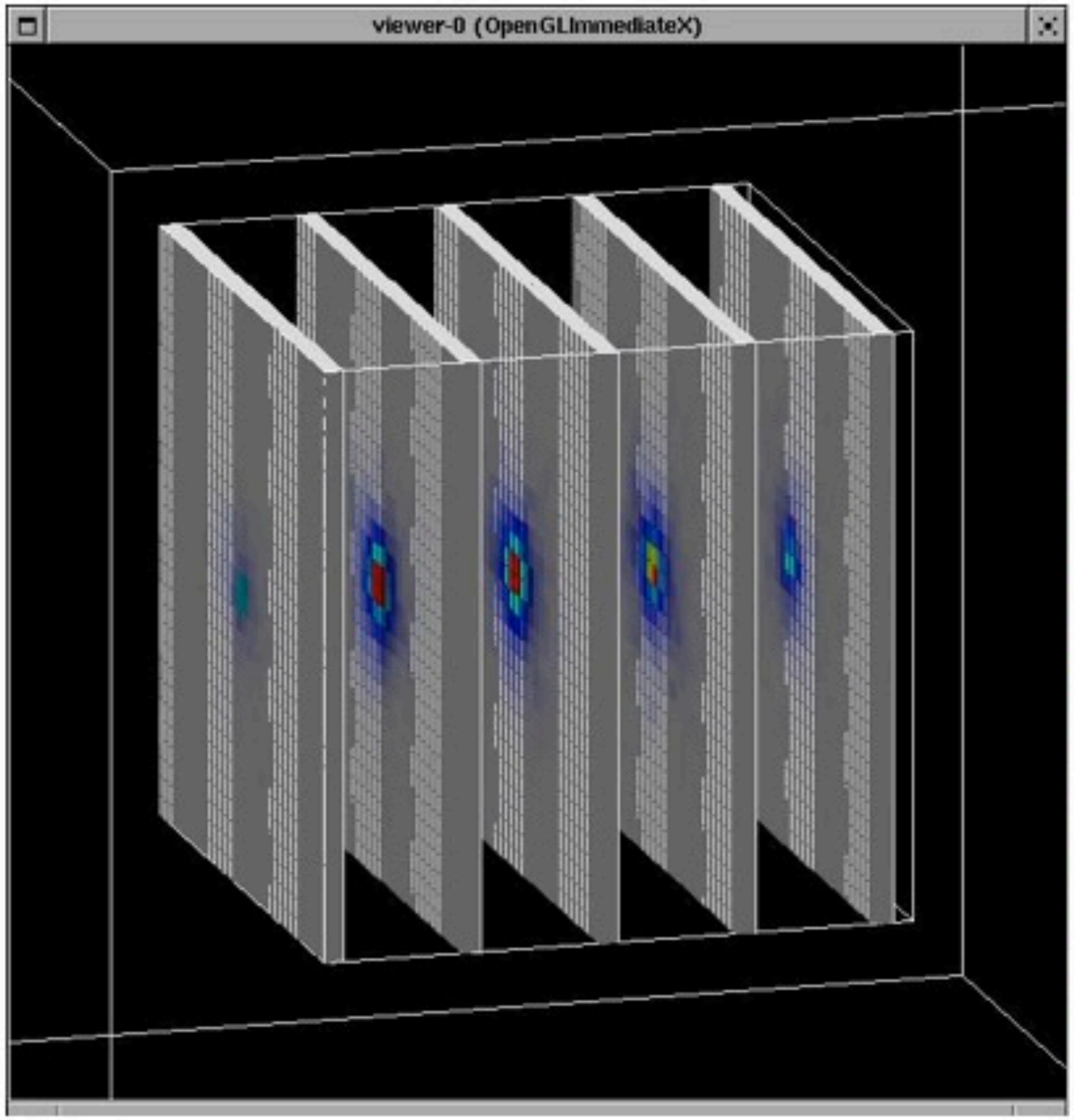


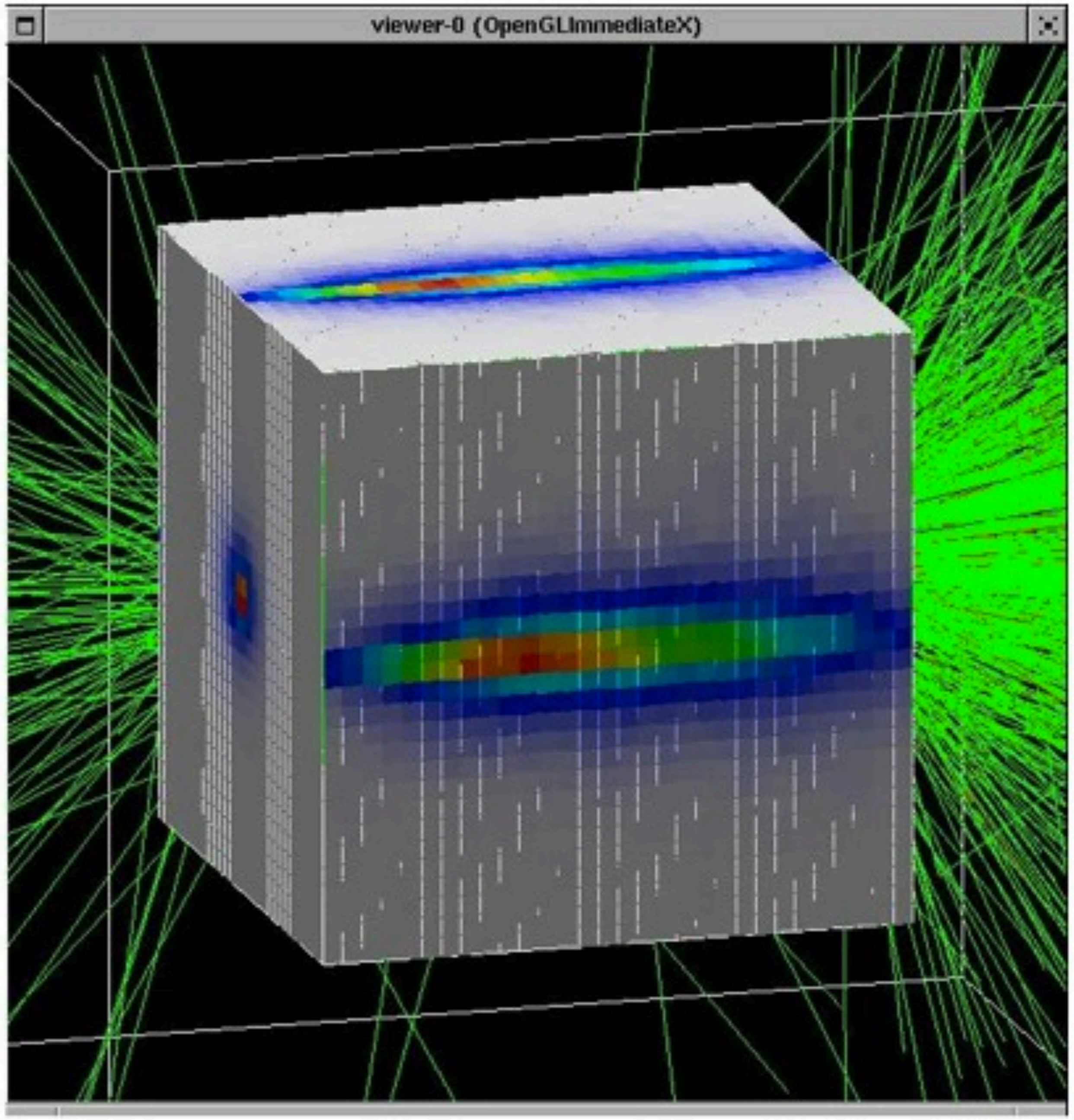
Geometry seen by e+, e-, γ



Geometry seen by other particles

Scoring volumes: an example of parallel word





Improvement in usability

- Removal of ordering nembers in physics list
  - Automatic consistency check
  - Easiness of combining physics builder
- Unifying error/warning message format
- Restructuring and polishing examples
- cmake and CLHEP
  - New installation procedure
  - A subset of CLHEP is included

A much more straightforward installation procedure

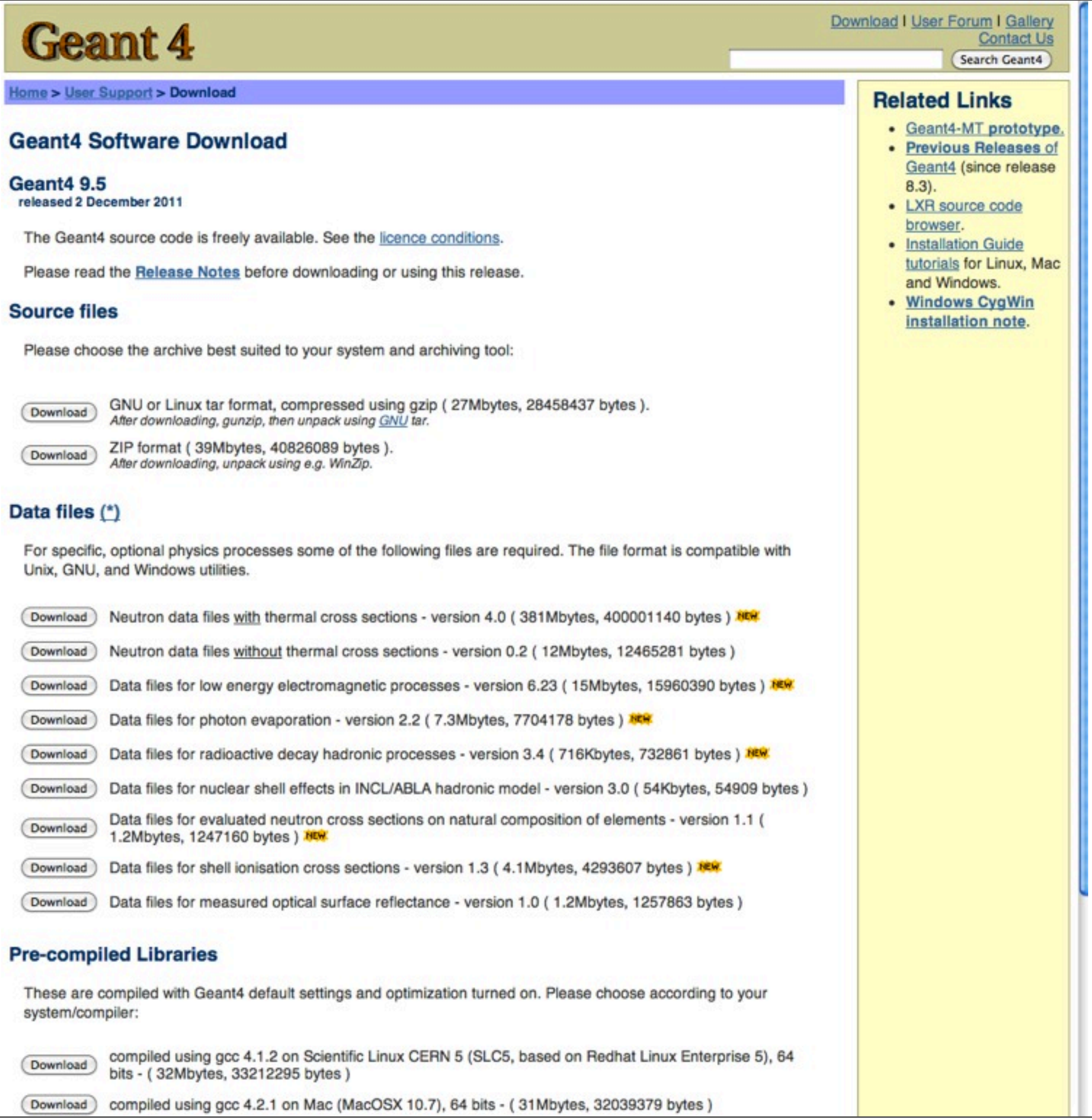
Collaboration-wide developments 2012-2013

- Performance improvments
- Review implementations of physics and transportation
  - A lot of code implemented without code performance
- Event bias options
  - Review, unify and enrich existing biasing options
  - Review interface
- Geant4-MT
  - Will process multiple events simultaneously
  - G4MT v9.6 at the end of 2012 or early 2013 will be the final prototype release
  - In 2013 we will merge G4MT into the main development repository

Collaboration-wide developments 2012-2013

- Version 9.6 will be released at the end of this year
- In November 2013 we'll produce the last v9 release
- The next Geant4 version X (name t.b.c)
  - Multi-thread capable
  - Minimal migration cost
  - (First) beta release in June 2013

- Fews concepts on Monte Carlo approach
- The Geant4 Collaboration
- Basic concepts and capabilities of Geant4
- Ongoing and future developments
- Installation tips
- Example of an application (Geometry, physics, tracking, etc)


Geant 4
Download | User Forum | Gallery
Contact Us
Search Geant4
Home > User Support > Download
Download

Geant4 Software Download

Geant4 9.5

released 2 December 2011

The Geant4 source code is freely available. See the licence conditions.

Please read the **Release Notes** before downloading or using this release.

Source files

Please choose the archive best suited to your system and archiving tool:

GNU or Linux tar format, compressed using gzip (27Mbytes, 28458437 bytes).
After downloading, gunzip, then unpack using GNU tar.

ZIP format (39Mbytes, 40826089 bytes).
After downloading, unpack using e.g. WinZip.

Data files (*)

For specific, optional physics processes some of the following files are required. The file format is compatible with Unix, GNU, and Windows utilities.

Neutron data files with thermal cross sections - version 4.0 (381Mbytes, 400001140 bytes) NEW

Neutron data files without thermal cross sections - version 0.2 (12Mbytes, 12465281 bytes)

Data files for low energy electromagnetic processes - version 6.23 (15Mbytes, 15960390 bytes) NEW

Data files for photon evaporation - version 2.2 (7.3Mbytes, 7704178 bytes) NEW

Data files for radioactive decay hadronic processes - version 3.4 (716Kbytes, 732861 bytes) NEW

Data files for nuclear shell effects in INCL/ABLA hadronic model - version 3.0 (54Kbytes, 54909 bytes)

Data files for evaluated neutron cross sections on natural composition of elements - version 1.1 (1.2Mbytes, 1247160 bytes) NEW

Data files for shell ionisation cross sections - version 1.3 (4.1Mbytes, 4293607 bytes) NEW

Data files for measured optical surface reflectance - version 1.0 (1.2Mbytes, 1257863 bytes)

Pre-compiled Libraries

These are compiled with Geant4 default settings and optimization turned on. Please choose according to your system/compiler:

compiled using gcc 4.1.2 on Scientific Linux CERN 5 (SLC5, based on Redhat Linux Enterprise 5), 64 bits - (32Mbytes, 33212295 bytes)

compiled using gcc 4.2.1 on Mac (MacOSX 10.7), 64 bits - (31Mbytes, 32039379 bytes)

Related Links

- Geant4-MT prototype.
- **Previous Releases of** Geant4 (since release 8.3).
- LXR source code browser.
- Installation Guide tutorials for Linux, Mac and Windows.
- **Windows CygWin installation note.**

Download and installation tips for the 9.5 version

- You can download the compiled libraries of Geant4 but **the compilation in your computer is strongly suggested**

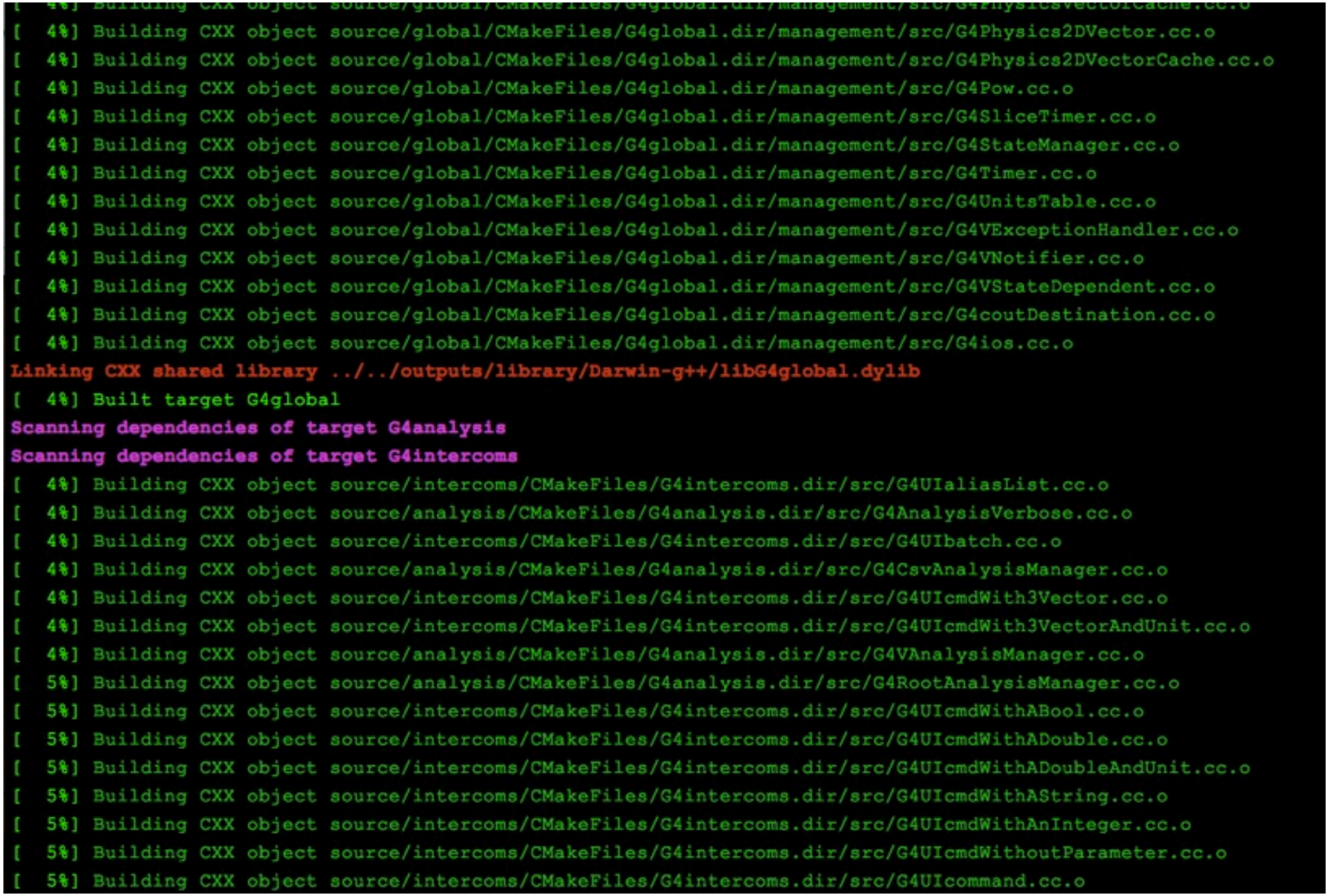
- **Download the source file** from the Geant4 web site

- Two way to proceed:
  - Using cmake via terminal
  - Using the GUI version of cmake

cmake Geant4 installation

- **cmake** version greater than 2.8.3
- Locate the **source folder**
  Ex: `/home/Username/geant4-09-05`
- Create the **build folder**
  Ex: `/home/Username/geant4-09-05-build`
- Create the **install folder**
  Ex: `/home/Username/geant4-09-05-install`
- `cmake -DCMAKE_INSTALL_PREFIX=/home/Username/geant4-09-05-install/`
- Define and/or activate the **additional features/package you require** using the same cmake interface
- make -jN
- make install

cmake Geant4 installation



```
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4PhysicsVectorCache.cc.o
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4Physics2DVector.cc.o
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4Physics2DVectorCache.cc.o
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4Pow.cc.o
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4SliceTimer.cc.o
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4StateManager.cc.o
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4Timer.cc.o
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4UnitsTable.cc.o
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4VExceptionHandler.cc.o
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4VNotifier.cc.o
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4VStateDependent.cc.o
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4coutDestination.cc.o
[  4%] Building CXX object source/global/CMakeFiles/G4global.dir/management/src/G4ios.cc.o
Linking CXX shared library ../../outputs/library/Darwin-g++/libG4global.dylib
[  4%] Built target G4global
Scanning dependencies of target G4analysis
Scanning dependencies of target G4intercoms
[  4%] Building CXX object source/intercoms/CMakeFiles/G4intercoms.dir/src/G4UIaliasList.cc.o
[  4%] Building CXX object source/analysis/CMakeFiles/G4analysis.dir/src/G4AnalysisVerbose.cc.o
[  4%] Building CXX object source/intercoms/CMakeFiles/G4intercoms.dir/src/G4UIbatch.cc.o
[  4%] Building CXX object source/analysis/CMakeFiles/G4analysis.dir/src/G4CsvAnalysisManager.cc.o
[  4%] Building CXX object source/intercoms/CMakeFiles/G4intercoms.dir/src/G4UIcmdWith3Vector.cc.o
[  4%] Building CXX object source/intercoms/CMakeFiles/G4intercoms.dir/src/G4UIcmdWith3VectorAndUnit.cc.o
[  4%] Building CXX object source/analysis/CMakeFiles/G4analysis.dir/src/G4VAnalysisManager.cc.o
[  5%] Building CXX object source/analysis/CMakeFiles/G4analysis.dir/src/G4RootAnalysisManager.cc.o
[  5%] Building CXX object source/intercoms/CMakeFiles/G4intercoms.dir/src/G4UIcmdWithABool.cc.o
[  5%] Building CXX object source/intercoms/CMakeFiles/G4intercoms.dir/src/G4UIcmdWithADouble.cc.o
[  5%] Building CXX object source/intercoms/CMakeFiles/G4intercoms.dir/src/G4UIcmdWithADoubleAndUnit.cc.o
[  5%] Building CXX object source/intercoms/CMakeFiles/G4intercoms.dir/src/G4UIcmdWithAString.cc.o
[  5%] Building CXX object source/intercoms/CMakeFiles/G4intercoms.dir/src/G4UIcmdWithAnInteger.cc.o
[  5%] Building CXX object source/intercoms/CMakeFiles/G4intercoms.dir/src/G4UIcmdWithoutParameter.cc.o
[  5%] Building CXX object source/intercoms/CMakeFiles/G4intercoms.dir/src/G4UIcommand.cc.o
```

cmake Geant4 installation

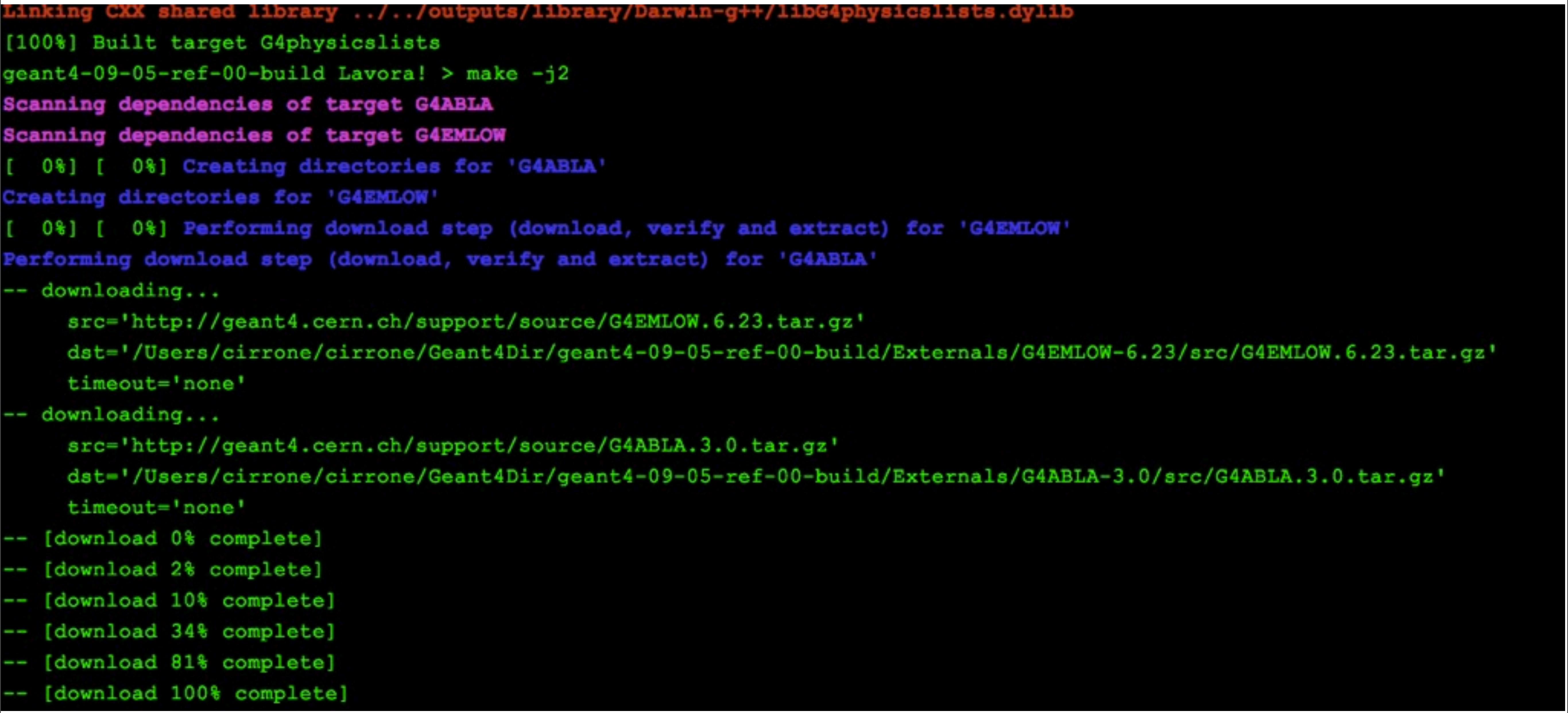
- If GEANT4_INSTALL_DATA is ON
  the additional external data libraries are automatically downloaded

- If GEANT4_INSTALL_EXAMPLES is ON
  Examples are installed

- If GEANT4_USE_SYSTEM_CLHEP is ON
  external CLHEP are searched

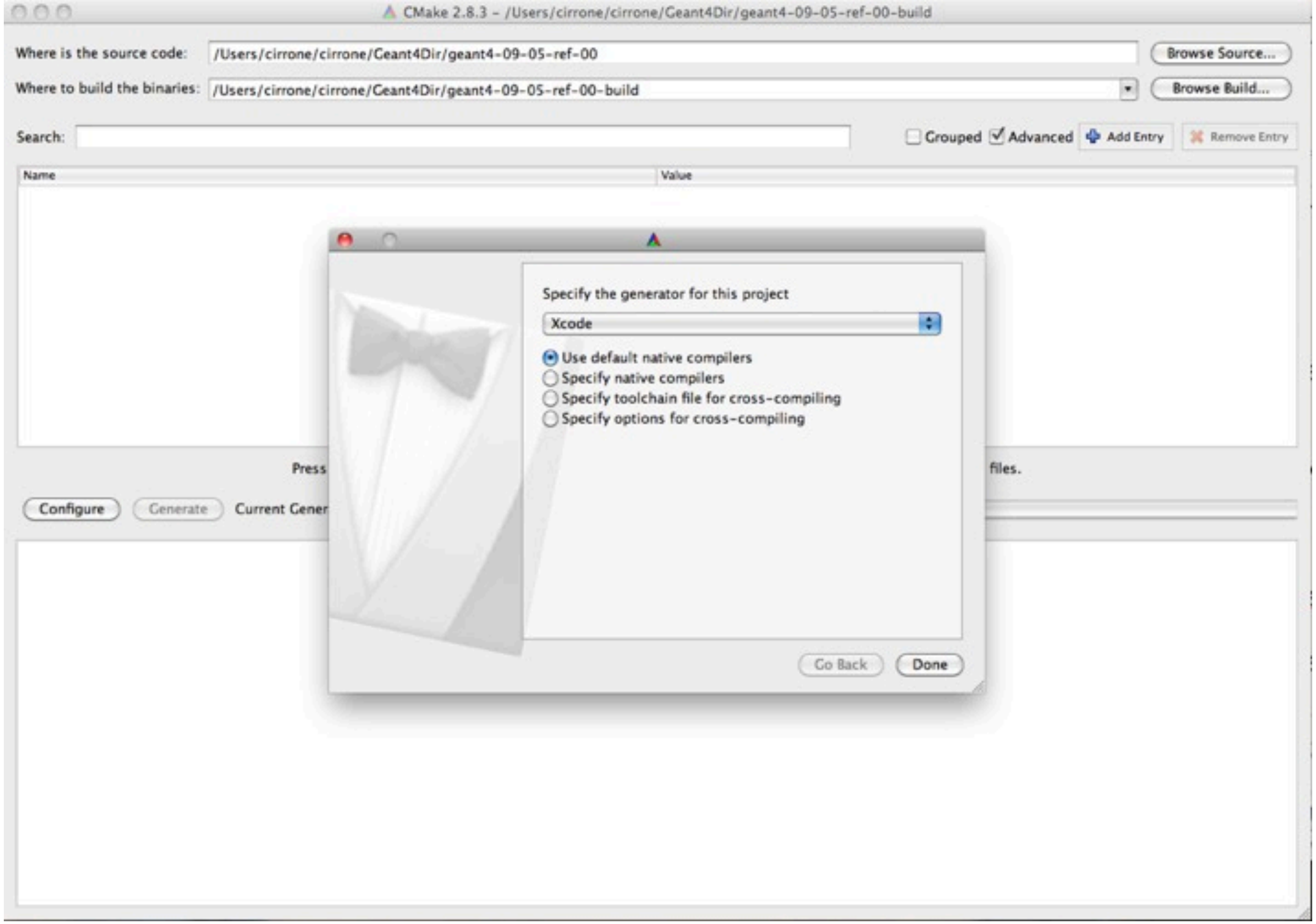
- See documentation for details for the complete variables list and explanation

cmake Geant4 installation



```
Linking CXX shared library ../../outputs/library/Darwin-g++/libG4physicslists.dylib
[100%] Built target G4physicslists
geant4-09-05-ref-00-build Lavora! > make -j2
Scanning dependencies of target G4ABLA
Scanning dependencies of target G4EMLOW
[  0%] [  0%] Creating directories for 'G4ABLA'
Creating directories for 'G4EMLOW'
[  0%] [  0%] Performing download step (download, verify and extract) for 'G4EMLOW'
Performing download step (download, verify and extract) for 'G4ABLA'
-- downloading...
     src='http://geant4.cern.ch/support/source/G4EMLOW.6.23.tar.gz'
     dst='/Users/cirrone/cirrone/Geant4Dir/geant4-09-05-ref-00-build/Externals/G4EMLOW-6.23/src/G4EMLOW.6.23.tar.gz'
     timeout='none'
-- downloading...
     src='http://geant4.cern.ch/support/source/G4ABLA.3.0.tar.gz'
     dst='/Users/cirrone/cirrone/Geant4Dir/geant4-09-05-ref-00-build/Externals/G4ABLA-3.0/src/G4ABLA.3.0.tar.gz'
     timeout='none'
-- [download 0% complete]
-- [download 2% complete]
-- [download 10% complete]
-- [download 34% complete]
-- [download 81% complete]
-- [download 100% complete]
```

GUI version of cmake



CMake 2.8.3 - /Users/cirrone/cirrone/Geant4Dir/geant4-09-05-ref-00-build

Where is the source code: /Users/cirrone/cirrone/Geant4Dir/geant4-09-05-ref-00

Where to build the binaries: /Users/cirrone/cirrone/Geant4Dir/geant4-09-05-ref-00-build

Search:

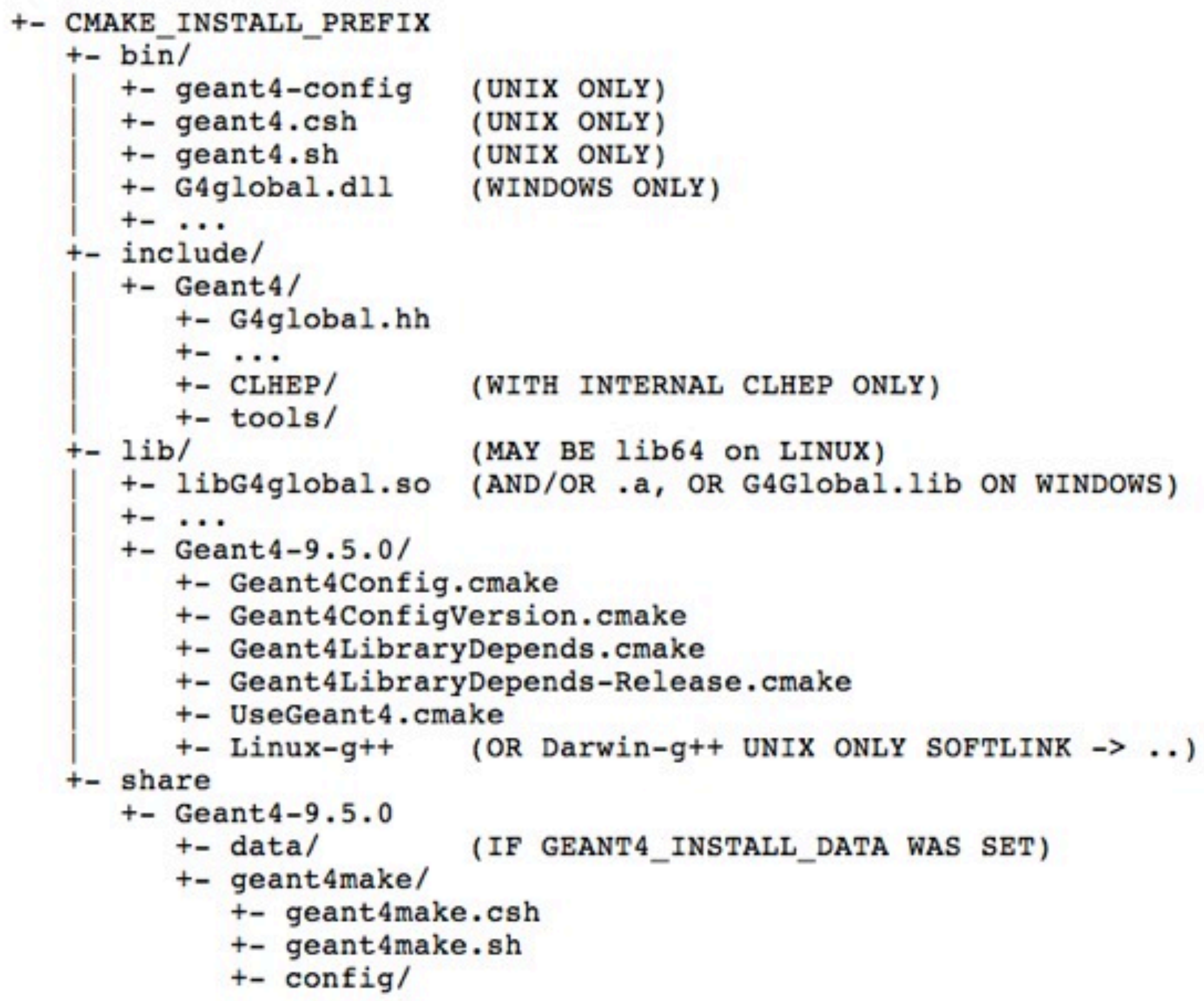
□ Grouped ☑ Advanced

Specify the generator for this project: Xcode

● Use default native compilers
○ Specify native compilers
○ Specify toolchain file for cross-compiling
○ Specify options for cross-compiling

A friendly way to do the same things

If everything is ok install tree should appear so structured



```
+- CMAKE_INSTALL_PREFIX
   +- bin/
   |  +- geant4-config   (UNIX ONLY)
   |  +- geant4.csh      (UNIX ONLY)
   |  +- geant4.sh       (UNIX ONLY)
   |  +- G4global.dll    (WINDOWS ONLY)
   |  +- ...
   +- include/
   |  +- Geant4/
   |     +- G4global.hh
   |     +- ...
   |     +- CLHEP/       (WITH INTERNAL CLHEP ONLY)
   |     +- tools/
   +- lib/               (MAY BE lib64 on LINUX)
   |  +- libG4global.so  (AND/OR .a, OR G4Global.lib ON WINDOWS)
   |  +- ...
   |  +- Geant4-9.5.0/
   |     +- Geant4Config.cmake
   |     +- Geant4ConfigVersion.cmake
   |     +- Geant4LibraryDepends.cmake
   |     +- Geant4LibraryDepends-Release.cmake
   |     +- UseGeant4.cmake
   |     +- Linux-g++    (OR Darwin-g++ UNIX ONLY SOFTLINK -> ..)
   +- share
      +- Geant4-9.5.0
         +- data/        (IF GEANT4_INSTALL_DATA WAS SET)
         +- geant4make/
            +- geant4make.csh
            +- geant4make.sh
            +- config/
```

cmake Geant4 installation

- How to **compile a User application**:
  - `source CMAKE_INSTALL_PREFIX/share/geant4make/geant4make.(c)sh`

- Where **data** and **examples** are located
  - data libraries CMAKE_INSTALL_PREFIX/share/Geant4Version/data
  - examples CMAKE_INSTALL_PREFIX/share/Geant4Version/examples