Geant4 Tutorial

SciNeGHE – Trieste 2010

An hands-on course based on Geant4 with emphasis on high energy astroparticle physics.

Lectures will cover all aspects of Geant4 from basic installation through advanced topics and will be interspersed with examples that build a progressively more complex application.

The course should be of interest both to complete novices and to those who already have some basic familiarity with Geant4. Participants are expected to have a reasonable knowledge of C++.

Based on Lectures by the G4 collaboration

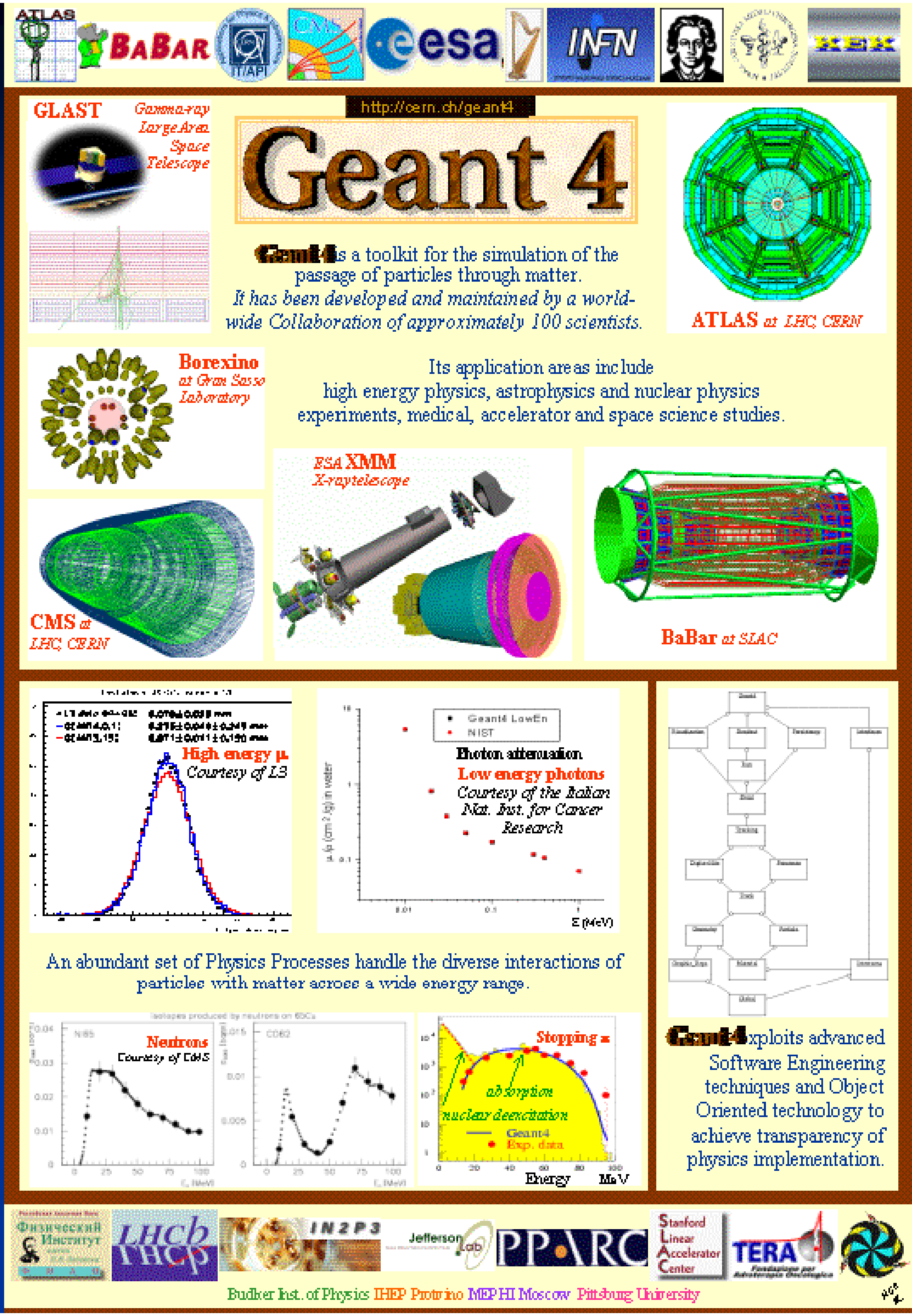
Outline of the G4 course

- General Introduction to G4
  - What is G4 ?
  - Review of user documentation
  - Geant4 as a toolkit
- Geant4 Kernel and basics of the toolkit
  - Run, Event, Step
  - Particle and Physics processes
  - User classes
- Geant4 geometry construction
  - Materials
  - Volumes
- Geant4 physics
- Particle Generation
- Simple Examples

Lecture 0
Geant4 Introduction

Contents

- General introduction and brief history
- Highlights of user applications
- User Documentation
- The main program



General introduction and brief history

What is Geant4?

- Geant4 is the successor of GEANT3, the world-standard toolkit for HEP detector simulation.
- Geant4 is one of the first successful attempt to re-design a major package of HEP software for the next generation of experiments using an Object-Oriented environment.
- A variety of requirements have also taken into account from heavy ion physics, CP violation physics, cosmic ray physics, astrophysics, space science and medical applications.
- In order to meet such requirements, a large degree of functionality and flexibility are provided.
- G4 is not only for HEP but goes well beyond that.

Geant4 – Its history

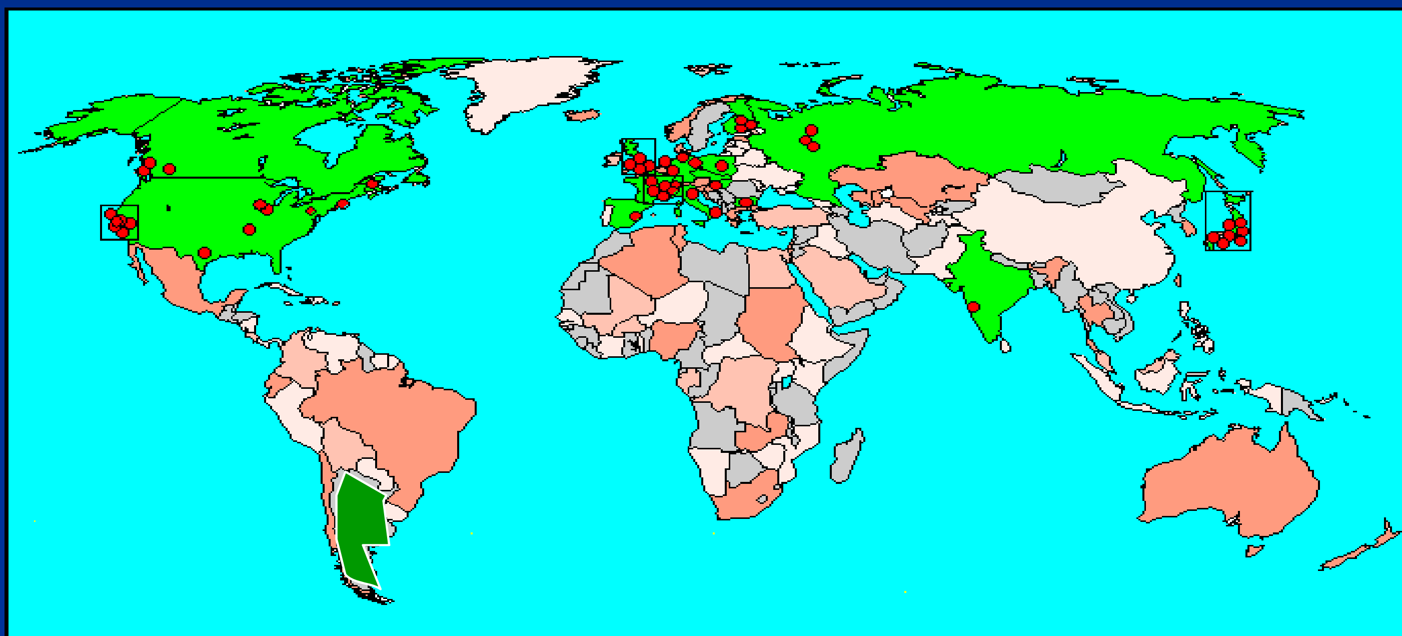
- Dec ’94 - Project start
- Apr ’97 - First alpha release
- Jul ’98 - First beta release
- Dec ’98 - First Geant4 public release
- ...
- June 2007 - Geant4 version 9.0 release
- December 2007 – Geant4 version 9.1 release
- June 2008 – Geant4 9.2 beta: open previewing of developments in 9.2
- December 2008 – Geant4 version 9.2 release
- September 2010 G4 v9.4b01 ← Current BETA version
- We currently provide two public releases every year.
- Bimonthly beta releases are available to the registered beta-testers.

Geant4 Collaboration

Stanford Linear Accelerator Center

TRIUMF

Lebedev



J.W.Goethe Universität

Collaborators also from non-member institutions, including
Budker Inst. of Physics
IHEP Protvino
MEPHI Moscow
Pittsburg University
Cordoba (Argentina)

Technology transfer

Particle physics software aids space and medicine

Geant4 is a showcase example of technology transfer from particle physics to other fields such as space and medical science [...].

CERN Courier, June 2002

http://top25.sciencedirect.com/index.php?subject_area_id=21

SCIENCE DIRECT

TOP 25

HOTTEST ARTICLES ON ScienceDirect

INTRODUCTION

The **ScienceDirect TOP25 Hottest Articles** is a **free** quarterly service from ScienceDirect. When you subscribe to the **ScienceDirect TOP25**, you'll receive an e-mail every three months listing the ScienceDirect users' 25 most frequently downloaded journal articles, from any selected journal among more than 2,000 titles in the ScienceDirect database, or from any of 24 subject areas.

Now you can keep track of the latest trends in your speciality and find out what your colleagues are reading. A simple click on any of the listed articles will take you to the journal abstract, and of course, you have the option to download any article straight to your desktop, depending on your access rights.

Read more about how the TOP25 is generated and what it reflects.

SIGN UP!

Would you like to be

Check out the ScienceDirect TOP25 Hottest Articles within your area of interest.

Subject Area: Physics and Astronomy

Journal: [all journals]

With these drop-down menus, the ScienceDirect TOP25 Hottest Articles are selected. Please refine your selection if necessary. To see the overall TOP25 within a certain subject area or journal, please select 'all subjects' or 'all journals' from the drop-down menus.

TOP25 articles within the subject area:
Physics and Astronomy

1. Nanoscience and engineering in mechanics and materials • **Article**
   Journal of Physics and Chemistry of Solids, Volume 65, Issue 8-9, 1 August 2004, Pages 1501-1506
   Chong, K.P.
2. Geant4-a simulation toolkit • **Article**
   Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 506, Issue 3, 1 July 2003, Pages 250-303
   Agostinelli, S.; Allison, J.; Amako, K.; Apostolakis, J.; Araujo, H.; Arce, P.; Asai, M.; Axen, D.; Banerjee, S.; Barrand, G.; Behner, F.; Bellagamba, L.; Boudreau, J.; Broglia, L.; Brunengo, A.; Burk
3. Radiation pneumonitis and pulmonary fibrosis in non-small-cell lung cancer: Pulmonary function, prediction, and prevention • **Article**
   *International Journal of Radiation Oncology*Biology*Physics, Volume 63, Issue 1, 1 September 2005, Pages 5-24*
   Mehta, V.

Flexibility of Geant4

- In order to meet wide variety of requirements from various application fields, a large degree of functionality and flexibility are provided.
- Geant4 has many types of geometrical descriptions to describe most complicated and realistic geometries
  - CSG, BREP and Boolean solids
  - Placement, replica, divided, parameterized, reflected and grouped
  - XML interface
- Everything is open to the user
  - Choice of physics processes/models
  - Choice of GUI/Visualization/persistency/histogramming technologies

Physics in Geant4

- It is rather unrealistic to develop a uniform physics model to cover wide variety of particles and/or wide energy range.
- Much wider coverage of physics comes from mixture of theory-driven, parameterized, and empirical formulae. Thanks to polymorphism mechanism, both cross-sections and models (final state generation) can be combined in arbitrary manners into one particular process.
- Geant4 offers
  - EM processes
  - Hadronic processes
  - Photon/lepton-hadron processes
  - Optical photon processes
  - Decay processes
  - Shower parameterization
  - Event biasing techniques
  - And you can plug-in more

Physics in Geant4

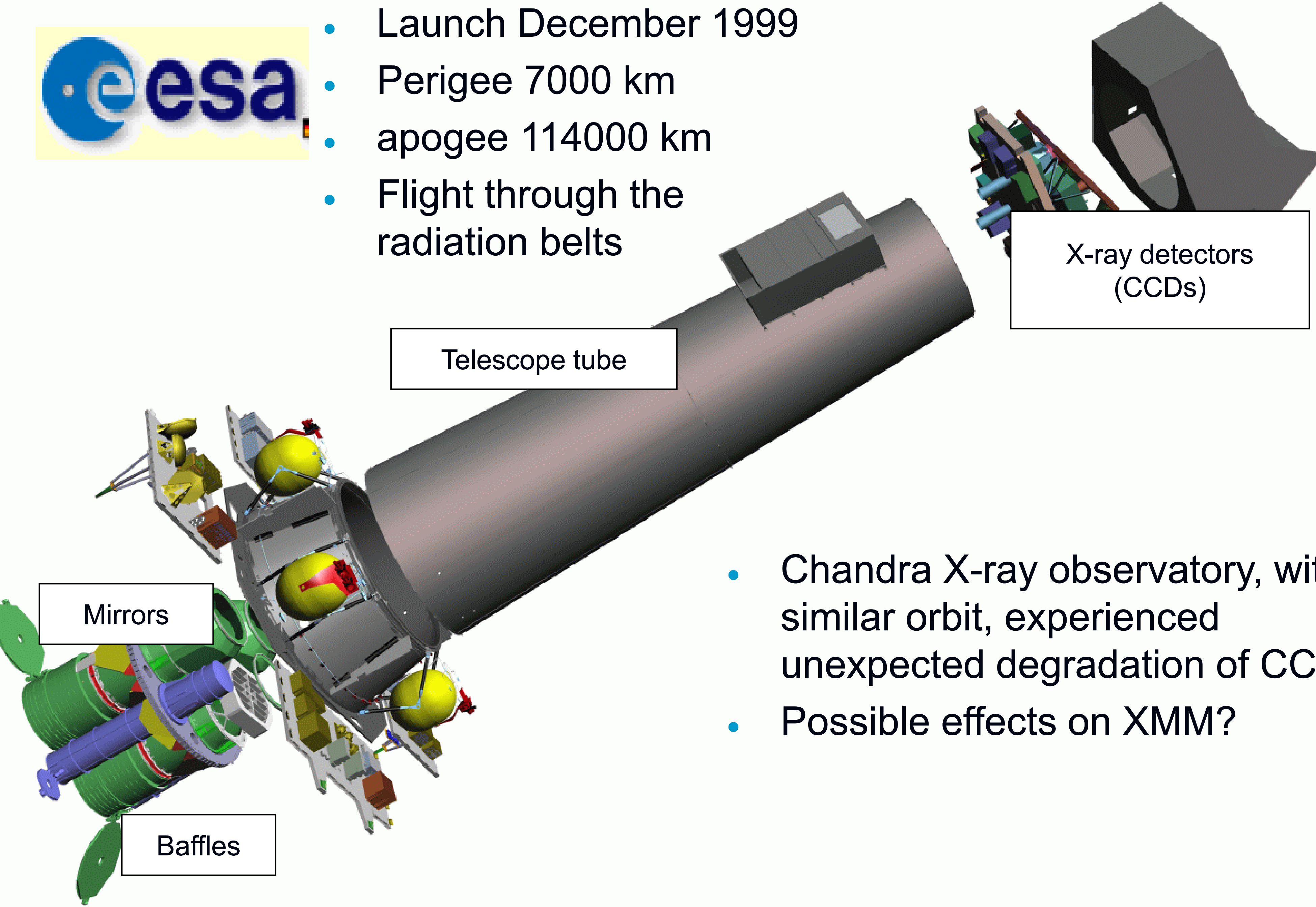
- Each cross-section table or physics model (final state generation) has its own applicable energy range. Combining more than one tables / models, one physics process can have enough coverage of energy range for wide variety of simulation applications.
- Geant4 provides sets of alternative physics models so that the user can freely choose appropriate models according to the type of his/her application.
  - In other words, it is the user's responsibility to choose reasonable set of physics processes/models that fits to his/her needs.
  - For example, some models are more accurate than others at a sacrifice of speed.

Highlights of Users Applications

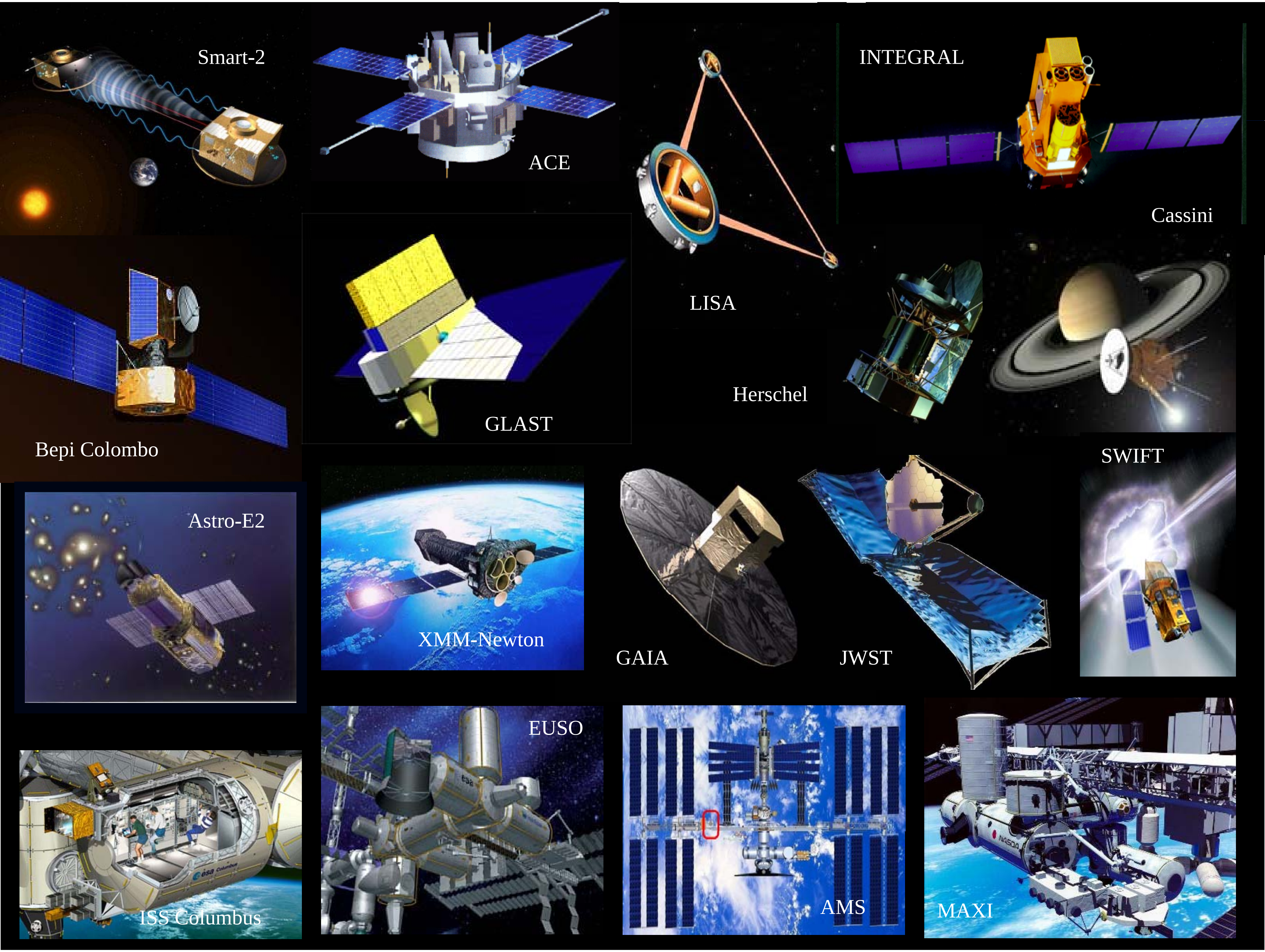
To provide you some ideas how Geant4 would be utilized...

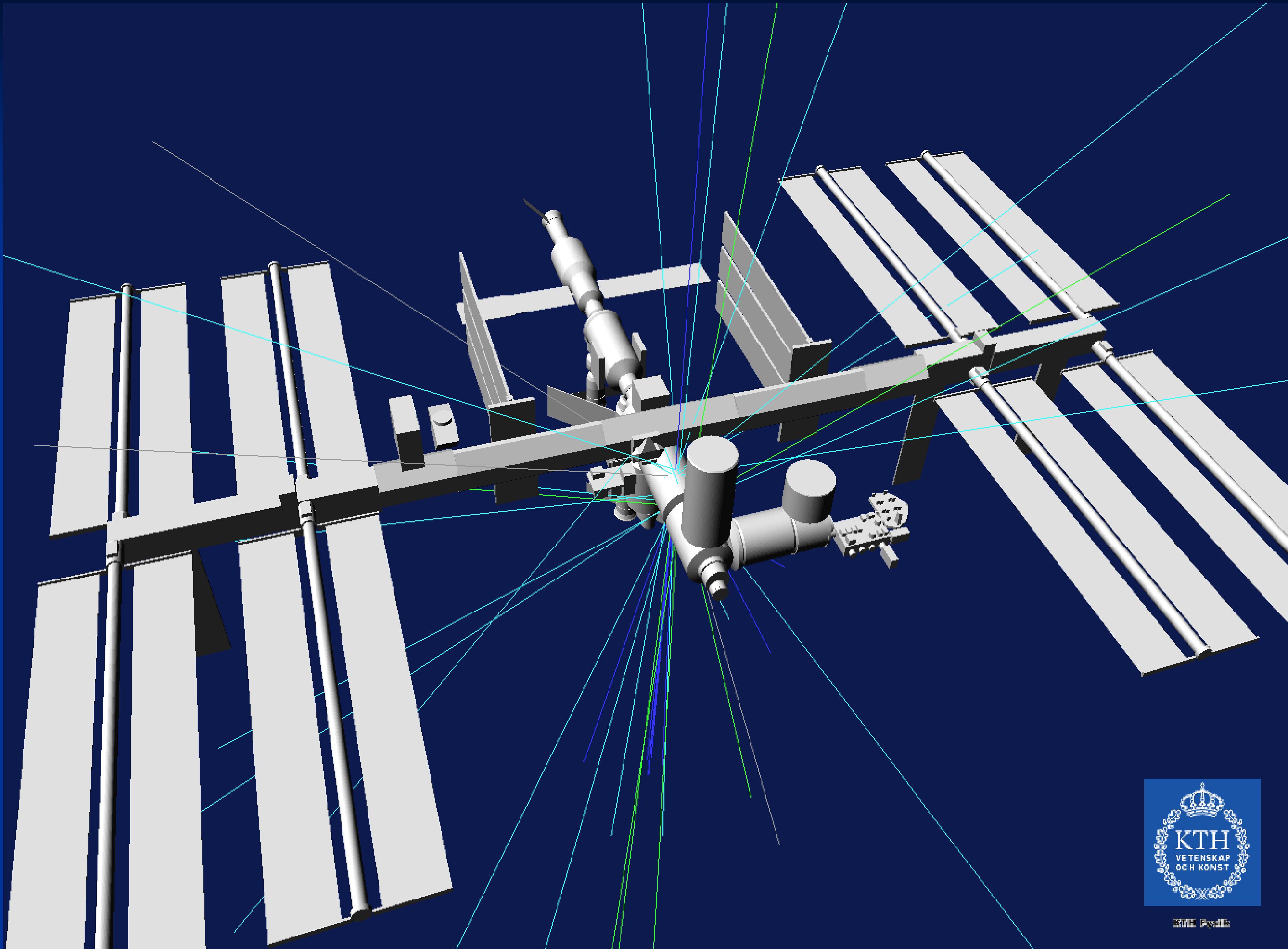
X-ray Multi-Mirror mission (XMM)

- Launch December 1999
- Perigee 7000 km
- apogee 114000 km
- Flight through the radiation belts



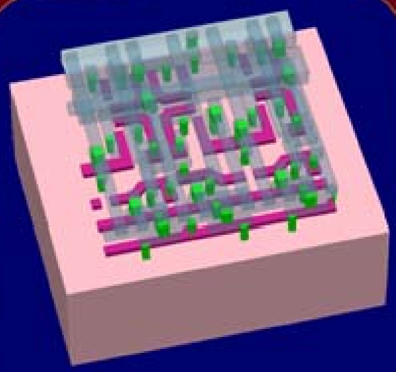
- Chandra X-ray observatory, with similar orbit, experienced unexpected degradation of CCDs
- Possible effects on XMM?


Smart-2
ACE
LISA
INTEGRAL
Cassini
Bepi Colombo
GLAST
Herschel
SWIFT
Astro-E2
XMM-Newton
GAIA
JWST
EUSO
AMS
MAXI
ISS Columbus

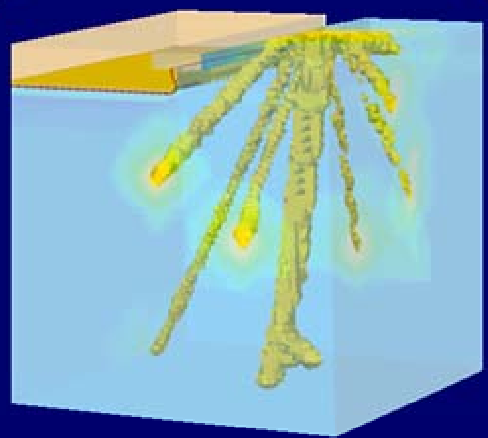


Courtesy T. Ersmark, KTH Stockholm

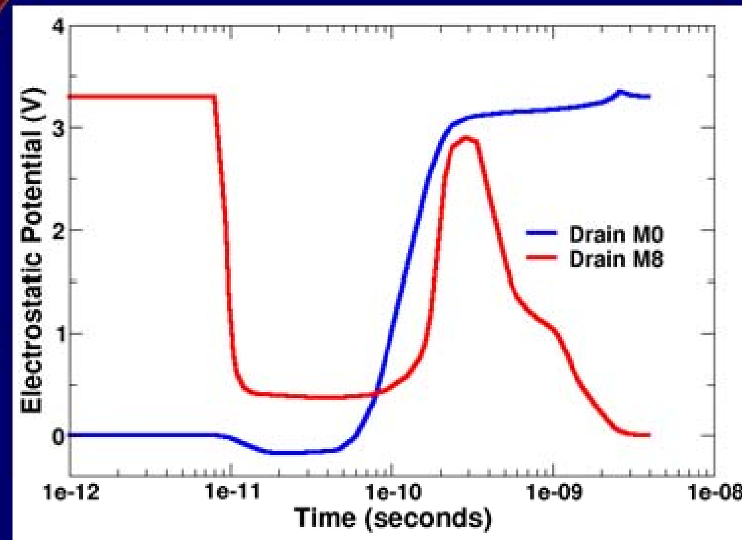
RADSAFE on SEE in SRAMs



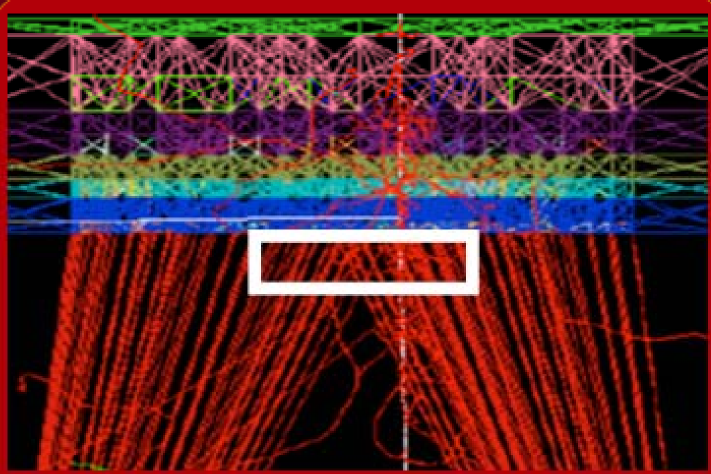
TCAD Cell Structure: SRAM Cell



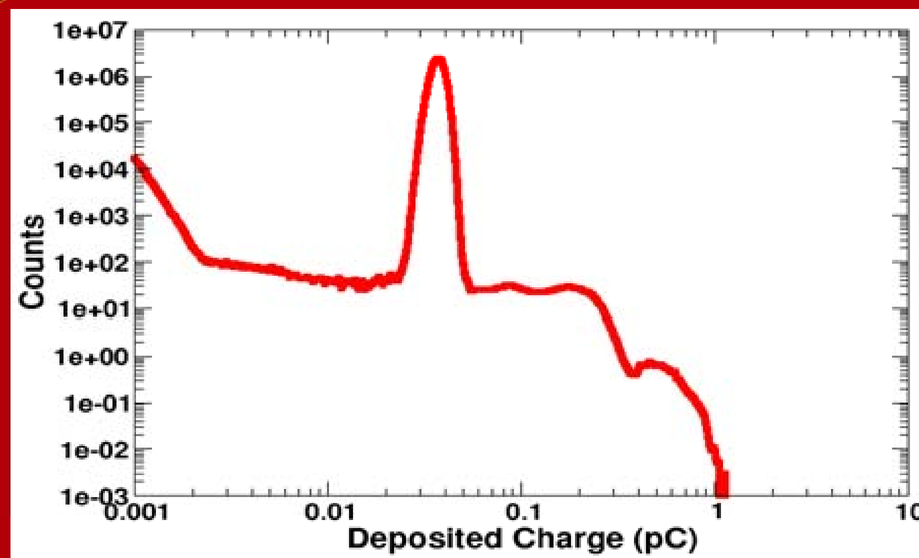
Single Charge Deposition in TCAD: Ne+W Event



SRAM Cell Upset

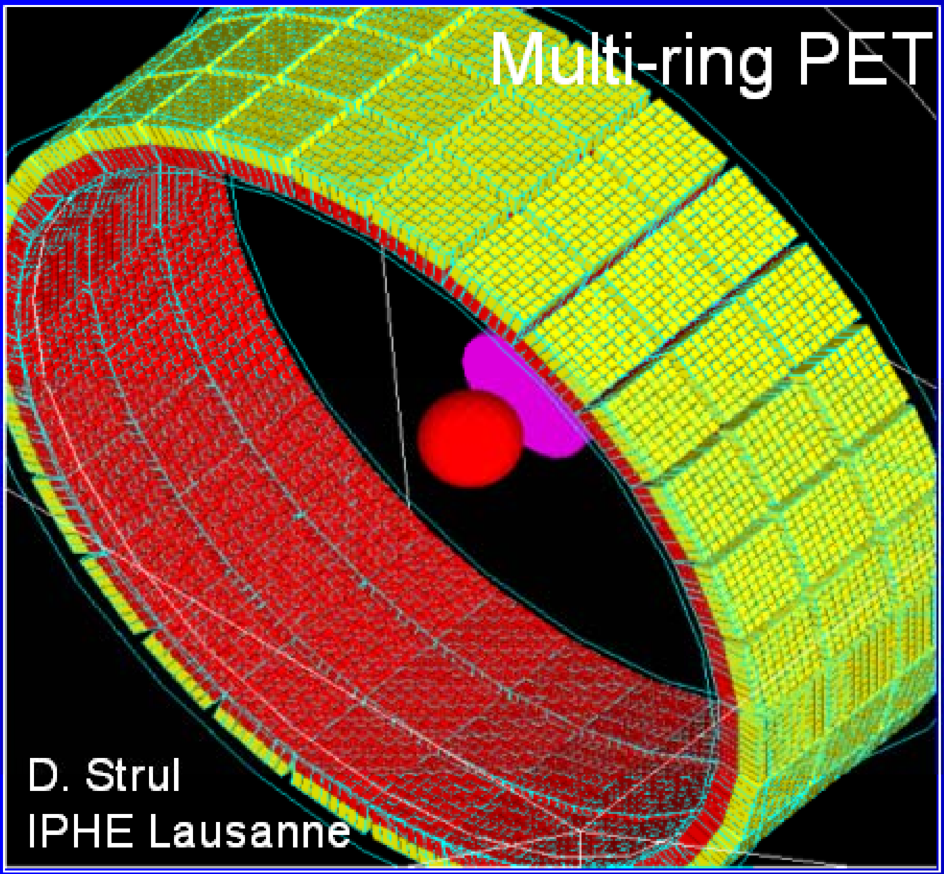


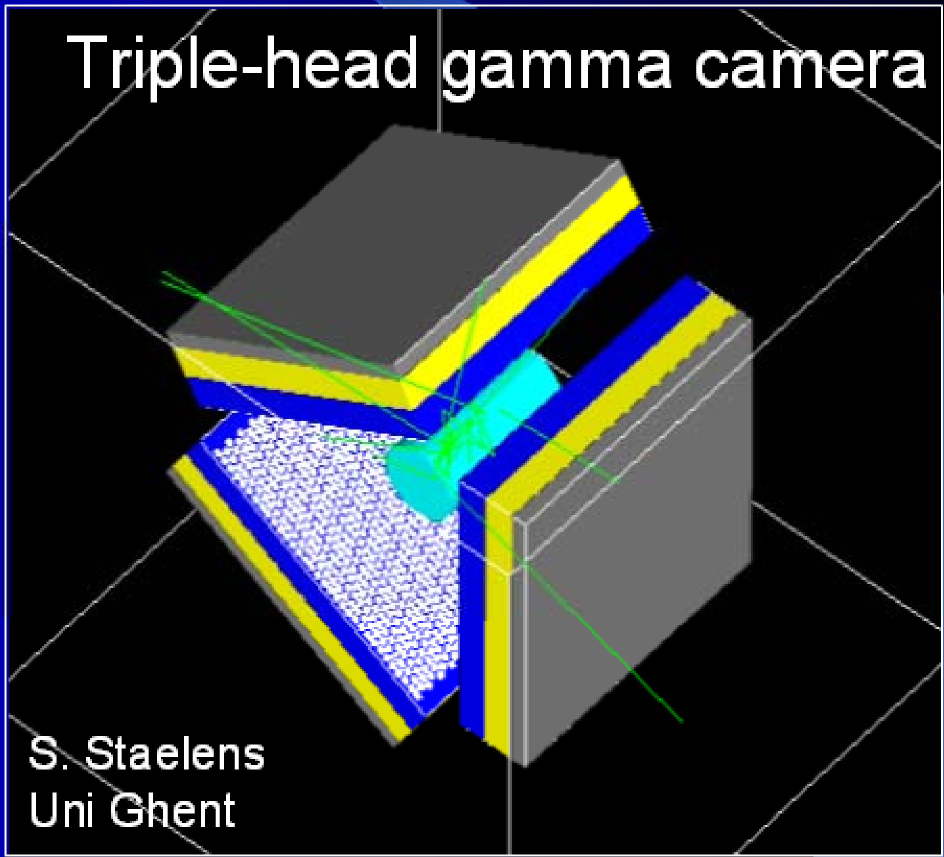
Geant4 Geometry and 523 MeV Neon Event



MRED Energy Deposition for 10^8 Events

Geometry examples of GATE applications





How to?

Installation of Geant4

- Follow instruction on G4 web site: http://www.cern.ch/geant4
- Download the code: http://geant4.web.cern.ch/geant4/support/download.shtml
- Download data tables
- System requirements
- CLHEP issue
- Visualization and Analysis tools

G4 download page

Geant 4

Home > User Support > Download > Beta code download

Beta Development Source Code Download

Geant4 9.4-beta

released 25 June 2010

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

Feedback for any problem found is welcome via the usual support channels.

Since this is **beta** software, please consider that:

- It is distributed "**as is**", and full support cannot be provided
- Some code may be new or enhanced, therefore still experimental and not fully tested
- Interfaces may have changed since the previous version and may change further in the final release
- User documentation and manuals are **not** updated (will be updated only for the official release)
- Only source code and no pre-built libraries are provided

Please read the Notes before downloading or using this code.

Source files

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

[Download] GNU or Linux tar format, compressed using gzip (19Mbytes, 19415368 bytes).
After downloading, gunzip, then unpack using GNU tar.

[Download] ZIP format (29Mbytes, 30613152 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.

[Download] Neutron data files with thermal cross sections - version 3.13 (54Mbytes, 54904967 bytes)

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

[Download] Evaluated neutron cross section data on natural composition of elements - version 1.0 (1.1Mbytes, 1186619 bytes) NEW

CLHEP

CLHEP - A Class Library for High Energy Physics

Shortcuts to: Documentation Download CLHEP editors Mailing List CLHEP Workshops News and Bug Reports

---

The CLHEP project was proposed by Leif Lönnblad at CHEP 92. It is intended to be a set of HEP-specific foundation and utility classes such as random generators, physics vectors, geometry and linear algebra. CLHEP is structured in a set of packages independent of any external package (interdependencies within CLHEP are allowed under certain conditions).

A large fraction of contributions (mainly to the Random, Vector, Geometry and Matrix packages) came from using CLHEP within (in alphabetical order):

- the BaBar experiment @ SLAC
- the Geant4 Collaboration
- the ZOOM Project @ Fermilab

Latest Release:

The latest release is **1.9.4.2/2.0.4.2**, released on Nov. 20th, 2008.

Notes:

- LCG is formally maintaining CLHEP.
  No further development is expected for the CLHEP library, except for bug fixes.
- HepPDT, HepMC, and StdHep packages are no longer included.
  HepMC and HepPDT have migrated to standalone external LCG packages.

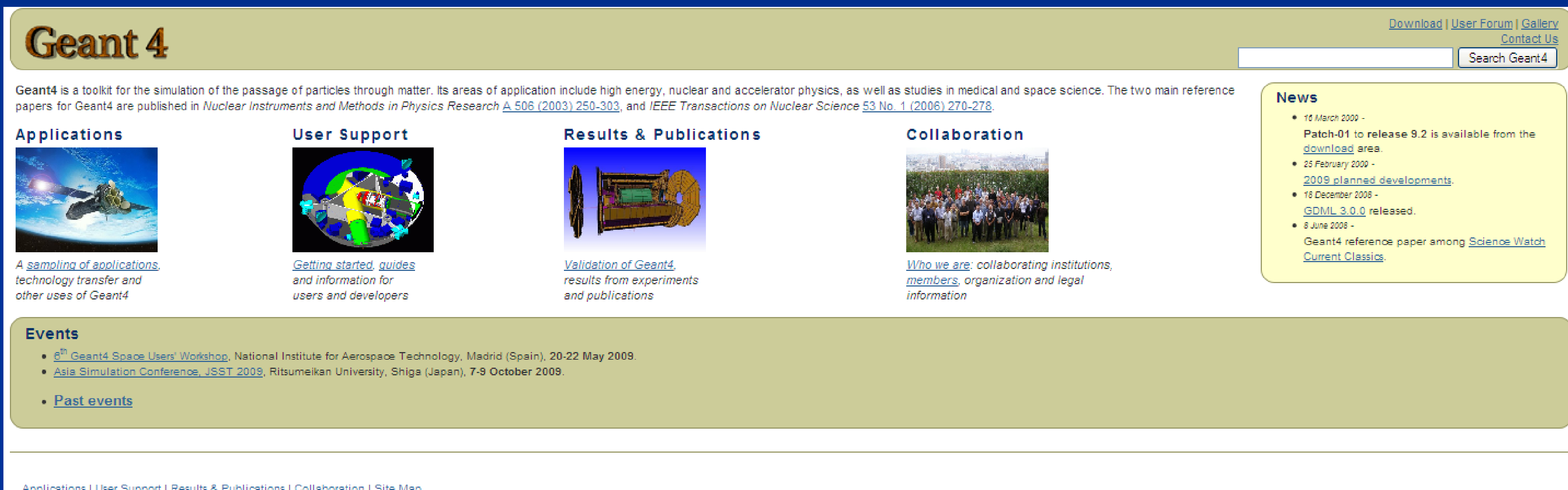
Visualization, Analysis, Geometry …

- Geant4 is a toolkit
- Many interfaces have been developed
  - Analysis (ROOT, jAIDA, OpenScientist)
  - Visualization (OpenGL, VRML, Dawn/eps …)
  - Geometry (XML description, CAD interfaces…)
- The basics you need is:
  - g++
  - CLHEP
  - A visualization driver
- All you need is setup in Configure script
- Much easier now …

G4 documentation basics

G4 home page

- http://geant4.web.cern.ch/geant4/



G4 Application Developer

Geant4 User's Guide for Application Developers

Geant4 Collaboration

Version: geant4 9.2

19 December, 2008

Table of Contents

G4 Physics manual

Contents

-10

Geant4 Software Manual

G4DMesonZero
G4DMmessenger
G4DNACrossSectionDataSet
G4DNAGenericIonsManager
G4DNAProcess
G4DopplerProfile
G4DrawVoxels
G4DsMesonMinus
G4DsMesonPlus
G4Dton
G4DummyFinalState
G4DummyModel
G4DummyProbability
G4DummyScalingAlgorithm
DumpFrames
G4DynamicParticle

E

G4E1Probability
G4E1Probability001
G4E1Probability01
G4E1Probability10
G4E1Probability100
G4E1SingleProbability001
G4E1SingleProbability01
G4E1SingleProbability1
G4E1SingleProbability10
G4E1SingleProbability100
E_isoAng
E_P_E_isoAng
G4eBremsstrahlung
G4eBremsstrahlungModel
G4eBremsstrahlungRelModel
G4eBremsstrahlungSpectrum
G4Ec2sub
G4EcId
G4eCoulombScatteringModel
G4ecpssrCrossSection
G4eCrossSectionExcitationEmfietzoglou
G4eCrossSectionScreenedRutherford
G4ee2KChargedModel
G4ee2KNeutralModel
G4eeCrossSections
G4Eenuc

G4PSTermination
G4PSTermination3D
G4PSTrackCounter
G4PSTrackCounter3D
G4PSTrackLength
G4PSTrackLength3D
G4PVDivision
G4PVDivisionFactory
G4PVParameterised
G4PVPlacement
G4PVReplica

Q

G4QANuANuNuclearCrossSection
G4QANuENuclearCrossSection
G4QANuMuNuclearCrossSection
G4QAOLowEnergyLoss
G4QAtomicElectronScattering
G4QBesIKJY
G4QCandidate
G4QCaptureAtRest
G4QChipolino
G4QCHIPSWorld
G4QCoherentChargeExchange
G4QCollision
G4QContent
G4QDecayChan
G4QDiffraction
G4QDiffractionRatio
G4QDiscProcessMixer
G4QElastic
G4QElasticCrossSection
G4QElectronNuclearCrossSection
G4QEmExtraPhysics
G4QEnvironment
G4QException
G4QFragmentation
G4QGluonString
G4QGSBinaryNeutronBuilder
G4QGSBinaryPiKBuilder
G4QGSBinaryProtonBuilder
G4QGSCEflowNeutronBuilder
G4QGSCEflowPiKBuilder
G4QGSCEflowProtonBuilder

Geant4 Software Manual

Cross-Referencing Geant4

Geant4/

License

Version: [ReleaseNotes] [1.0] [1.1] [2.0] [3.0] [3.1] [3.2] [4.0] [4.0.p1] [4.0.p2] [4.1] [4.1.p1] [5.0] [5.0.p1] [5.1] [5.1.p1] [5.2] [5.2.p1] [5.2.p2] [6.0] [6.0.p1] [6.1] [6.2] [6.2.p1] [6.2.p2] [7.0] [7.0.p1] [7.1] [7.1.p1] [8.0] [8.0.p1] [8.1] [8.1.p1] [8.1.p2] [8.2] [8.2.p1] [8.3] [8.3.p1] [8.3.p2] [9.0] [9.0.p1] [9.0.p2] [9.1] [9.1.p1] [9.1.p2] [9.1.p3] [9.2] [*9.2.p1*]

- [***source navigation***] - [identifier search] - [freetext search] - [file search] -

Name	Size	Last modified (GMT)	Description
digits_hits/		2009-03-16 14:08:14	
environments/		2009-03-16 14:08:12	
error_propagation/		2009-03-16 14:08:14	
event/		2009-03-16 14:08:14	
examples/		2009-03-16 14:08:12	
g3tog4/		2009-03-16 14:08:14	
geometry/		2009-03-16 14:08:14	
global/		2009-03-16 14:08:14	
graphics_reps/		2009-03-16 14:08:14	
intercoms/		2009-03-16 14:08:14	
interfaces/		2009-03-16 14:08:14	
materials/		2009-03-16 14:08:14	
parameterisations/		2009-03-16 14:08:14	
particles/		2009-03-16 14:08:14	
persistency/		2009-03-16 14:08:14	
physics_lists/		2009-03-16 14:08:14	
processes/		2009-03-16 14:08:14	
readout/		2009-03-16 14:08:14	

User Forum

GEANT4 at hypernews.slac.stanford.edu Forum List by Category

Geant 4

Forums by Category
Forums by Time Order
Request a New Forum

Recent Postings
Search in Forums
Subscribe to Forums

Member Info
Members List
New Member

Overview
Contact Admin

Category: Applications

Educational Applications	Industrial instruments	Medical Applications	Space Applications

Category: Control of runs, events, tracks, particles

Event and Track Management	Particles	Run Management

Category: Experimental Setup

Biasing and Scoring	Fields: Magnetic and Otherwise	Geometry	Hits, Digitization and Pileup

Category: General matters

Documentation and Examples	HyperNews System Announcements	Hypernews Testing	Installation and Configuration	User Requirements

Category: Interfaces

(Graphical) User Interfaces	Analysis	Persistency	Visualization

Category: Physics

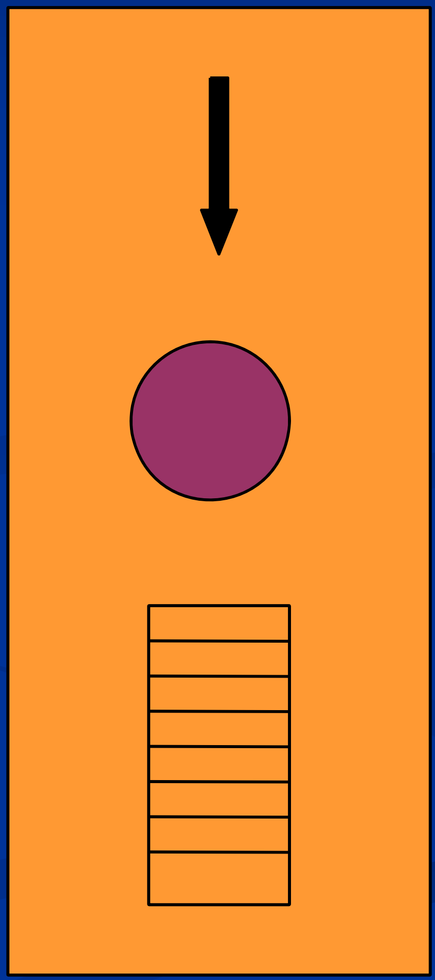
Biasing and Scoring	Electromagnetic Processes	Fast Simulation, Transportation & Others	Hadronic Processes	Physics List
Processes Involving Optical Photons				

This site runs SLAC HyperNews version 1.11-slac-98, derived from the original HyperNews

Geant4 Simple Examples

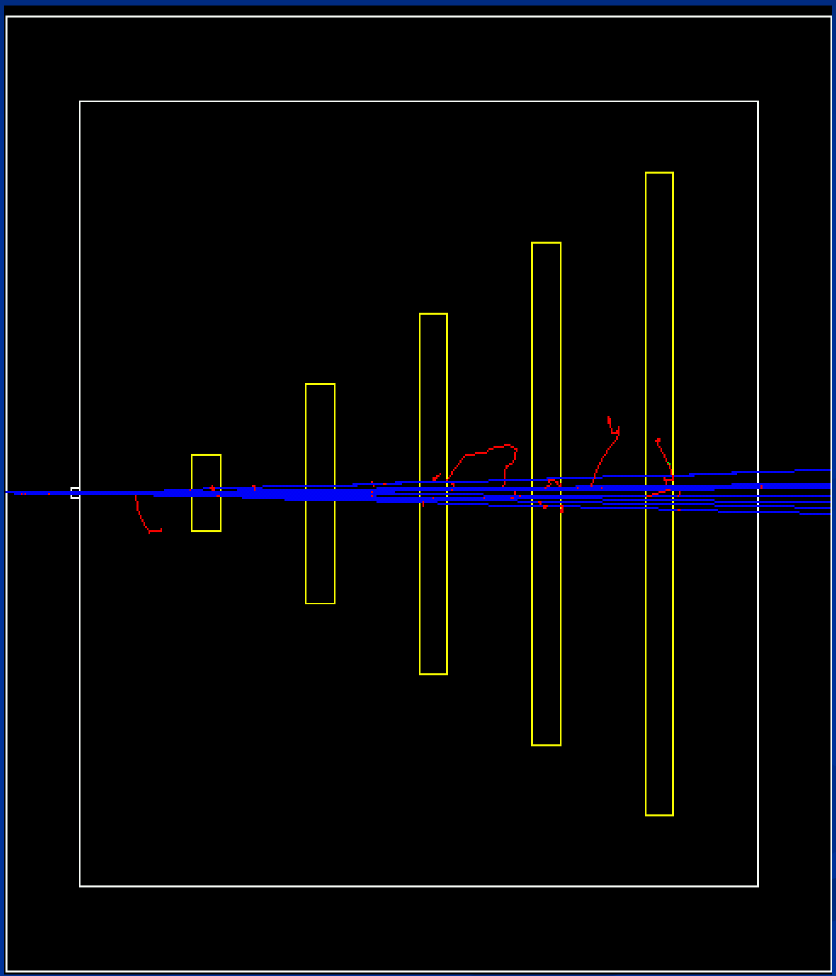
Novice Example N01

- Fixed geometry: Ar gas mother volume with Al cylinder and Pb block with Al slices
- Incident particle is a geantino – no physics interactions
- No magnetic field and only the transportation process is enabled
- Hard coded batch job and verbosity



Novice Example N02

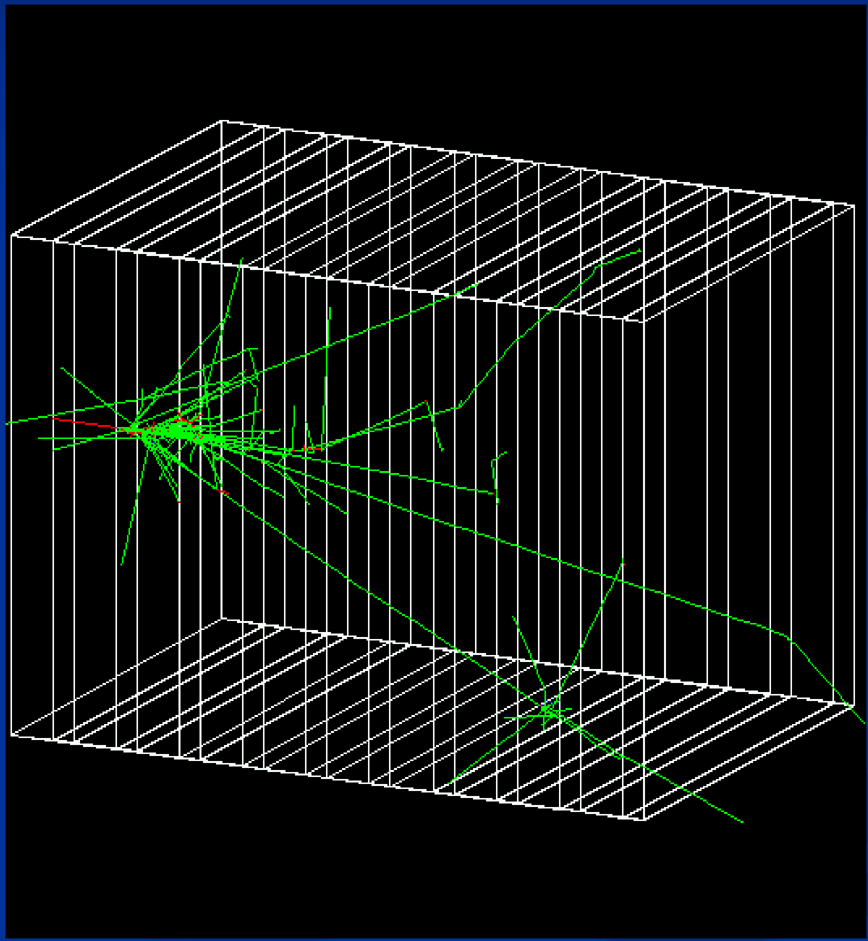
- Pb target, Xe gas chambers (parameterized volumes)
- All EM processes + decay included for γ, charged leptons and charged hadrons
- Detector response
  - Trajectories and chamber hit collections may be stored
- Visualization of detector and event
- Command interface introduced
  - Can change target, chamber materials, magnetic field, incident particle type, momentum, etc. at run time



Novice Example N03

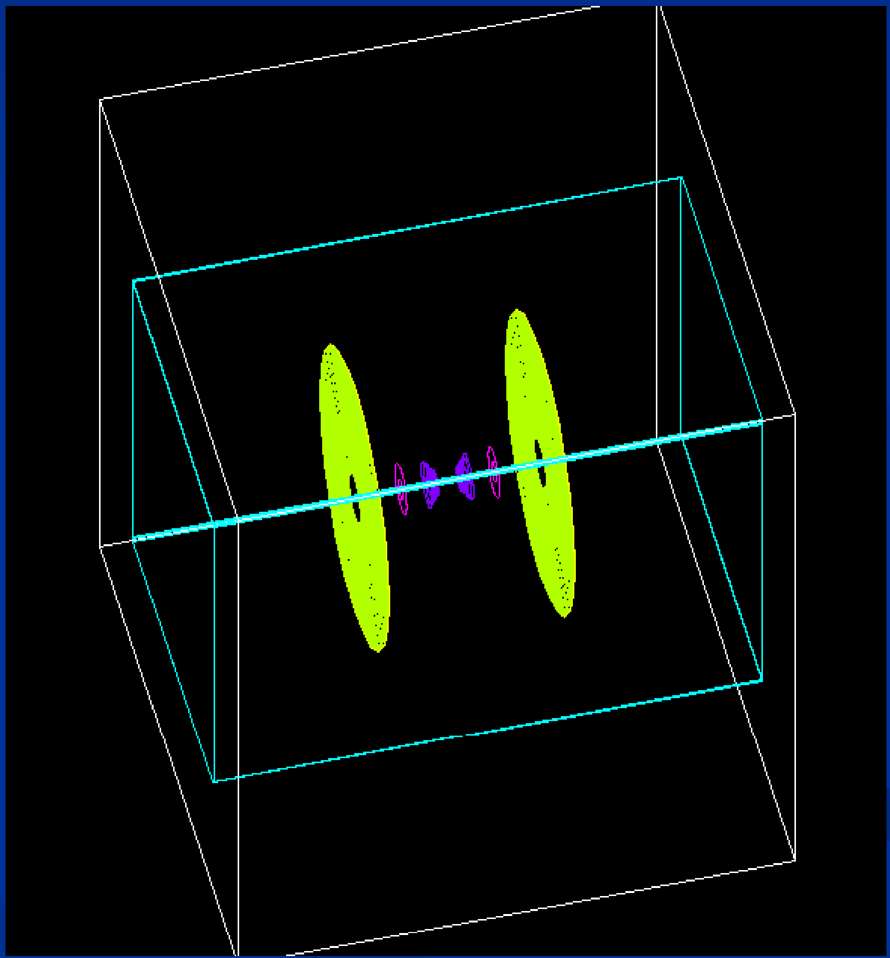
- Sampling calorimeter with layers of Pb absorber and liquid Ar detection gaps (replicas)
- Exhaustive material definitions
- Command interface
- Randomization of incident beam

- All EM processes + decay, with separate production cuts for γ, e+, e- (use for shower studies)
- Detector response: E deposit, track length in absorber and gap
- Visualization tutorial
- Random number seed handling



Novice Example N04

- Simplified collider detector
  - all kinds of volume definitions
- Magnetic field
- PYTHIA primary event generator
  - Higgs decay by Z0, lepton pairs
- Full set of EM + hadronic processes
  - Should use updated hadronic physics lists
- Event filtering by using stacking mechanism

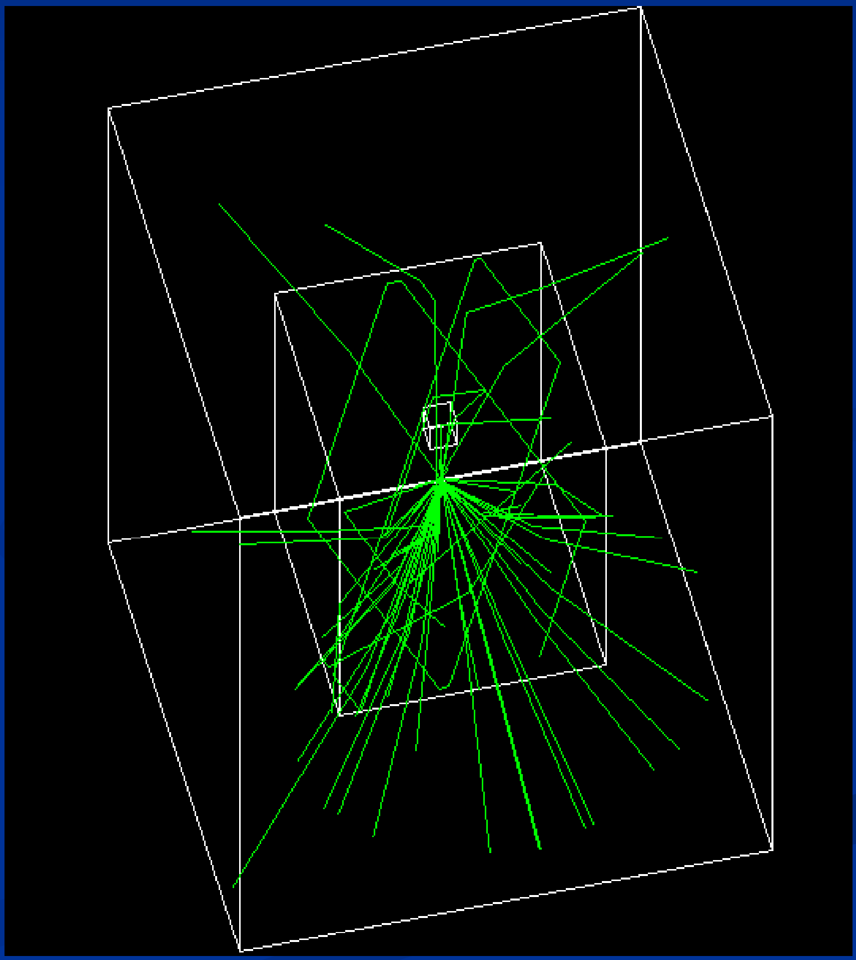


Novice Example N05

- Fast simulation with parameterized showers
  - EM showers (derived from G4VFastSimulationModel)
  - Pion showers (for illustration only – not used)
- EM physics only
  - Use of G4FastSimulationManagerProcess
- Simplified collider detector geometry
  - Drift chamber
  - EM, hadronic calorimeter
  - Ghost volume

Novice Example N06

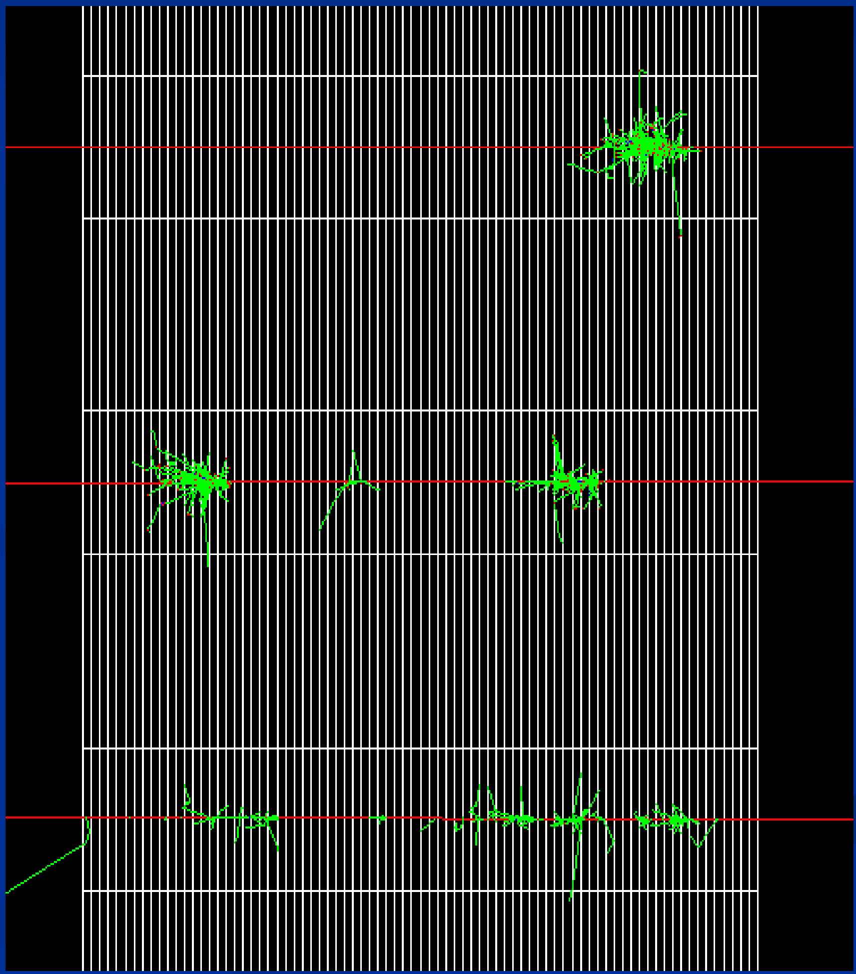
- Water Cerenkov detector with air “bubble”
- Materials
  - Specification of optical properties
  - Specification of scintillation spectra
- Physics
  - Optical processes
  - Generation of Cerenkov radiation, energy loss collected to produce scintillation



Novice Example N07

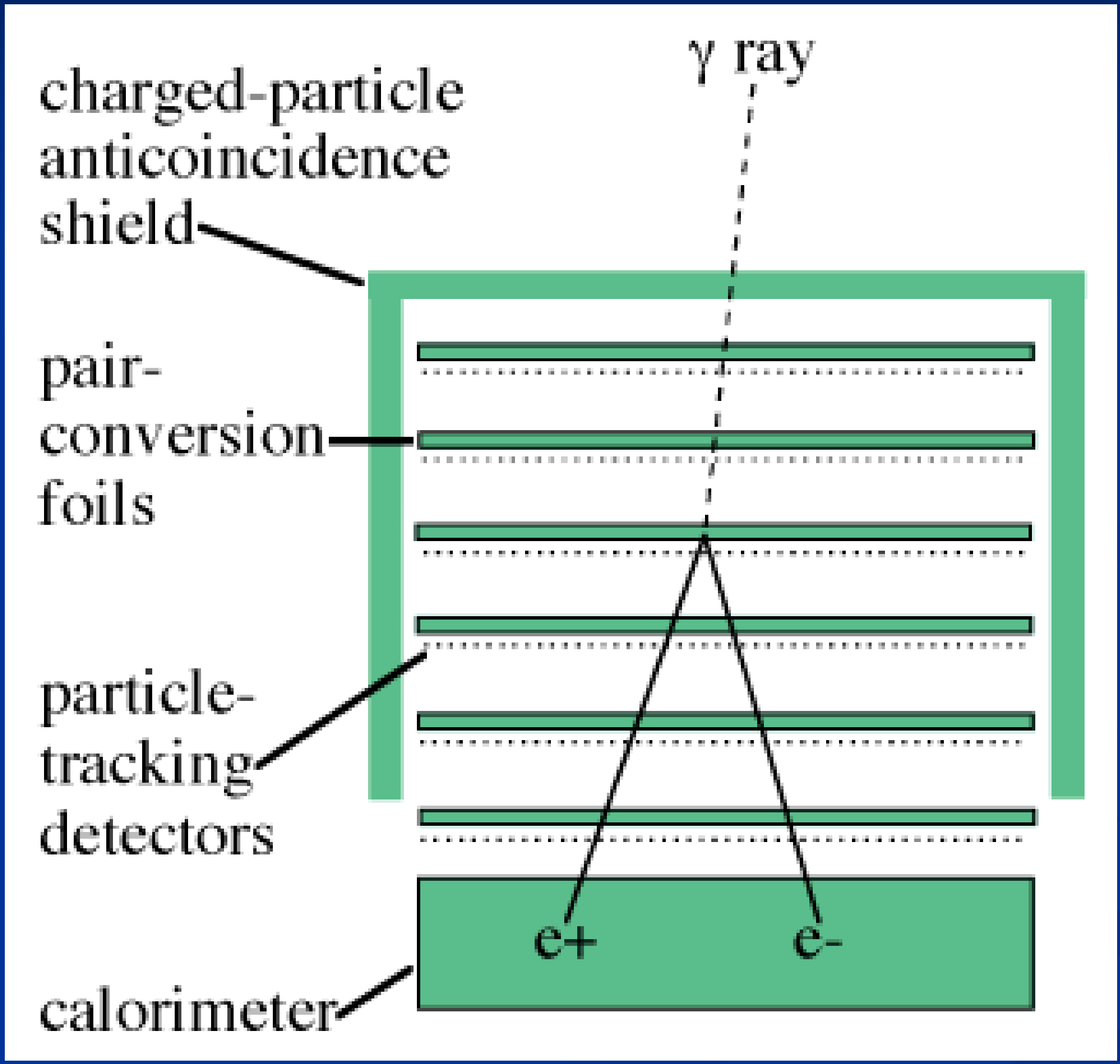
- 3 simplified sandwich calorimeters (Pb, Al, Ar)
- Run-based (as opposed to event-based) hit accumulation

- Changing geometries without re-building world
- Setting different secondary production cuts for each calorimeter using G4Region

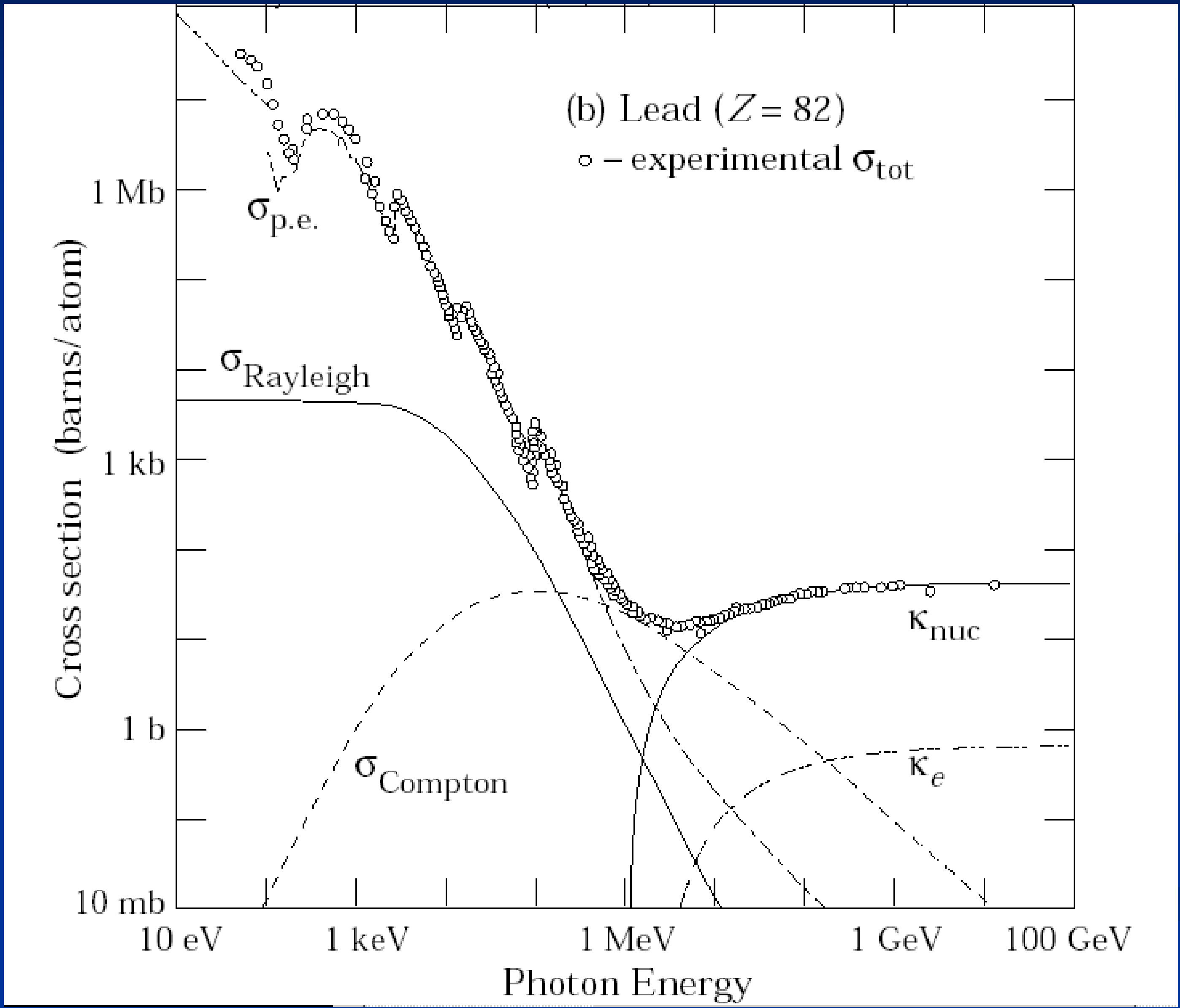


Fermi LAT Example

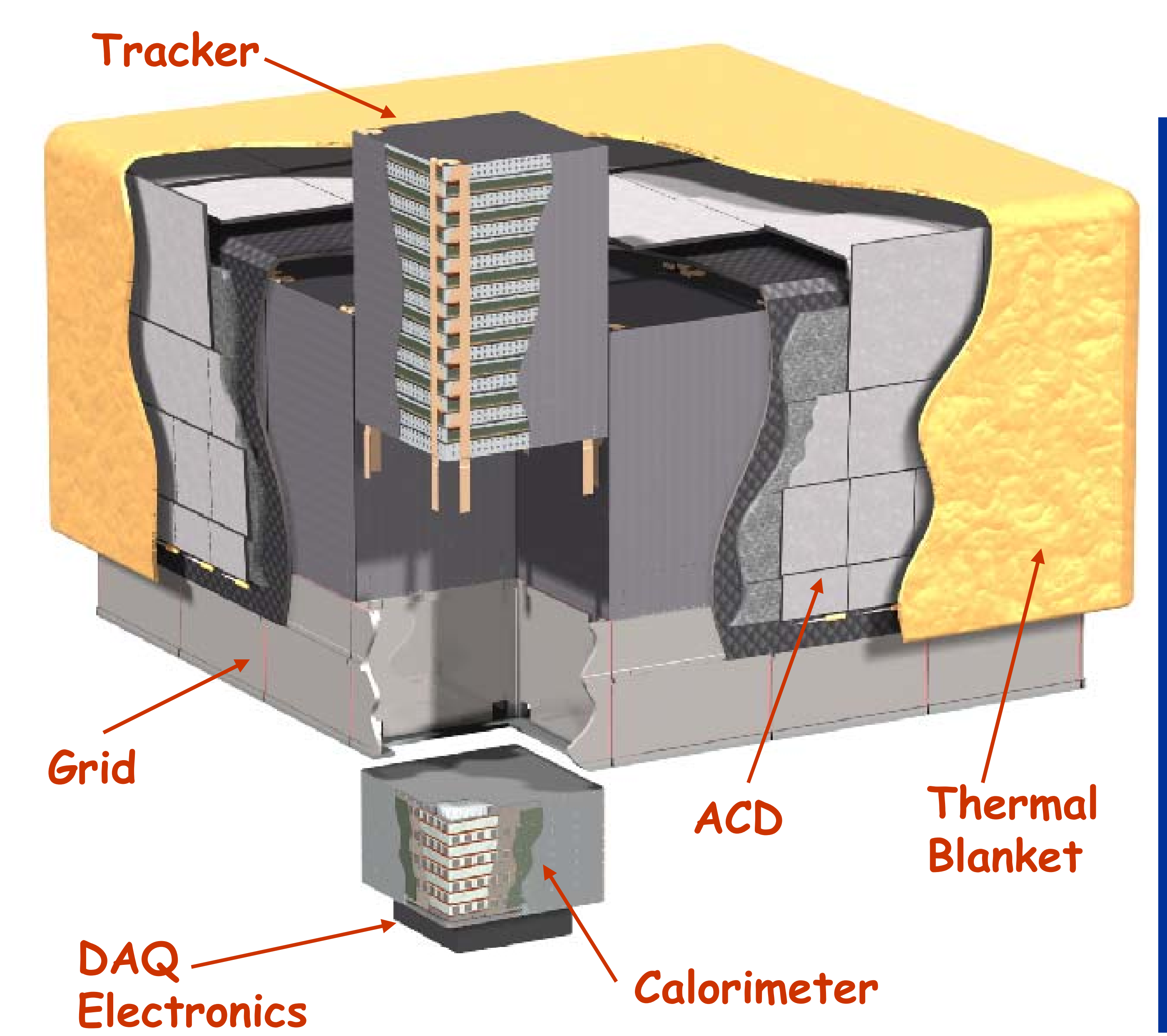
The need of Simulation



Photon Interactions



The Large Area Telescope (LAT)

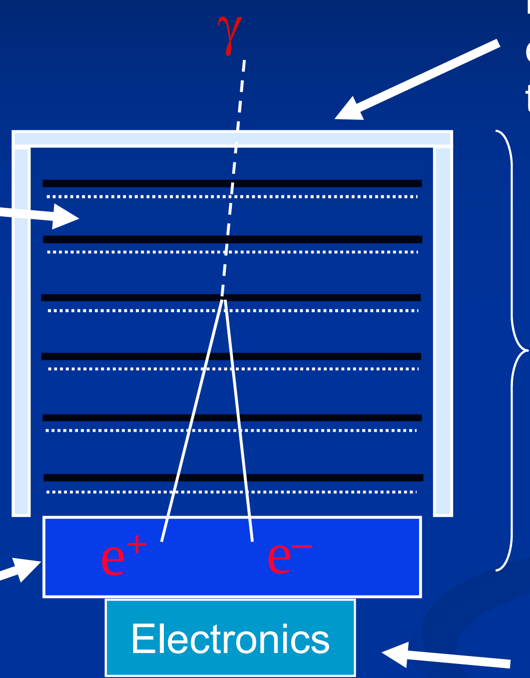


- Array of 16 identical "Tower" Modules, each with a tracker (Si strips) and a calorimeter (CsI with PIN diode readout) and DAQ module.
- Surrounded by finely segmented ACD (plastic scintillator with PMT readout).
- Aluminum strong-back "Grid," with heat pipes for transport of heat to the instrument sides.

Fermi LAT Detector Project

Effective area and PSF requirements drive the converter thicknesses and layout. PSF requirements also drive the sensor performance, layer spacings, and drive the design of the mechanical supports.

Background rejection requirements drive the ACD design (and influence the calorimeter and tracker layouts).



Field of view sets the aspect ratio (height/width)

Energy range and energy resolution requirements bound the thickness of calorimeter

Time accuracy provided by electronics and intrinsic resolution of the sensors.

On-board transient detection requirements, and on-board background rejection to meet telemetry requirements, are relevant to the electronics, processing, flight software, and trigger design.

Instrument life has an impact on detector technology choices.
Derived requirements (source location determination and point source sensitivity) are a result of the overall system performance.

Detector Project

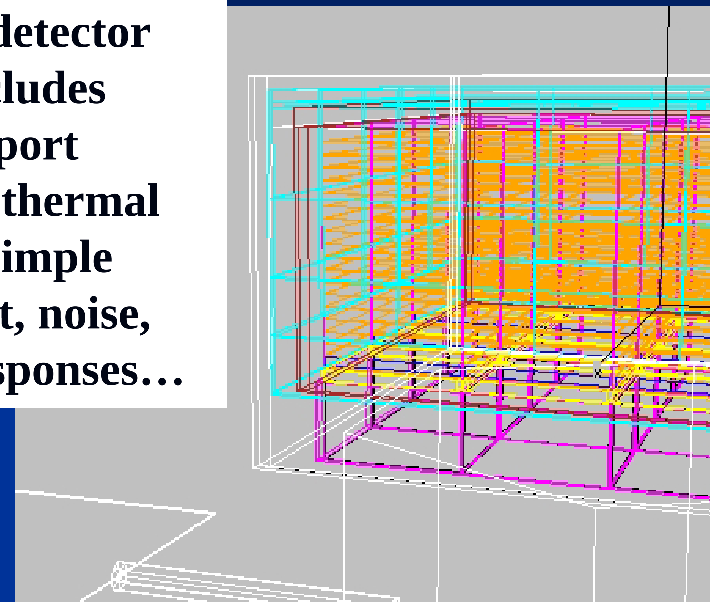
The LAT design is based on detailed Monte Carlo simulations.

Integral part of the project from the start.

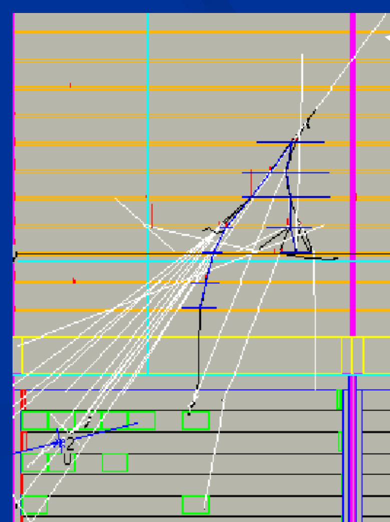
- **Background rejection**
- **Calculate effective area and resolutions (computer models now verified by beam tests). Current reconstruction algorithms are existence proofs -- many further improvements under development.**
- **Trigger design.**
- **Overall design optimization.**

Simulations and analyses are all C++, based on standard HEP packages.

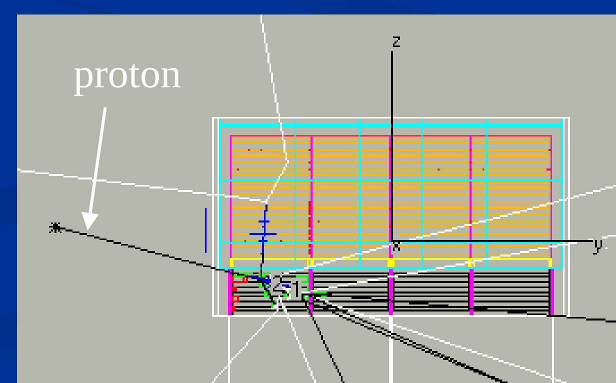
Detailed detector model includes gaps, support material, thermal blanket, simple spacecraft, noise, sensor responses…



Instrument naturally distinguishes gammas from backgrounds, but details matter.

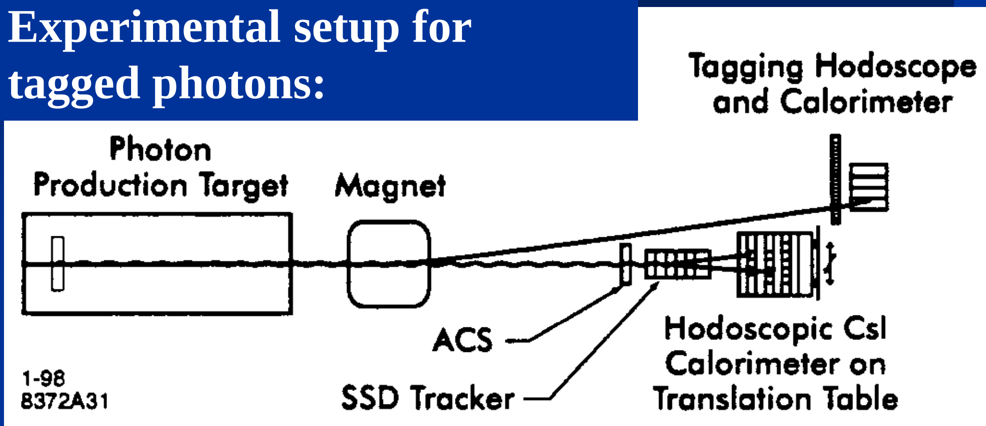


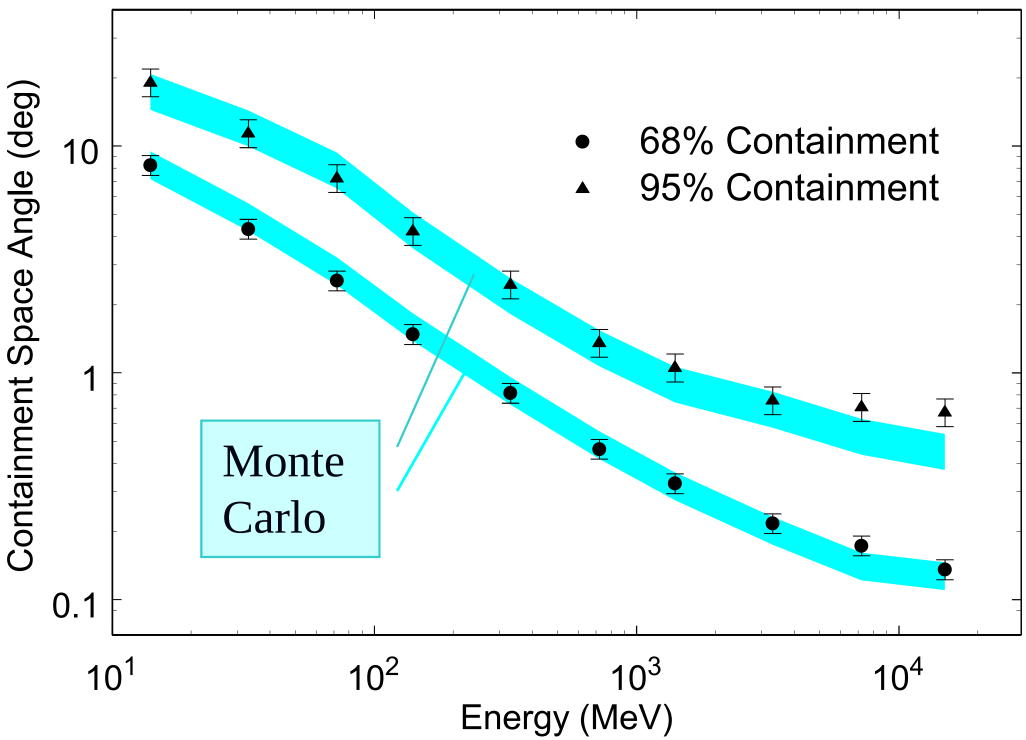
gamma ray



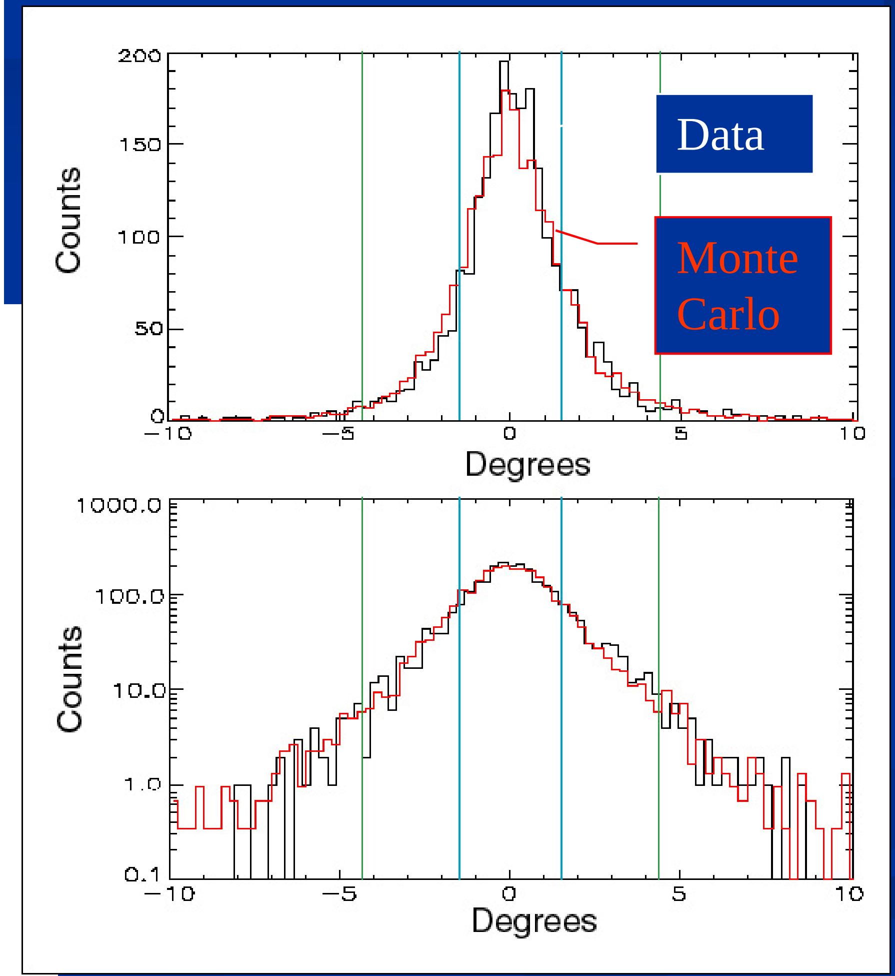
Detector Project

Experimental setup for tagged photons:



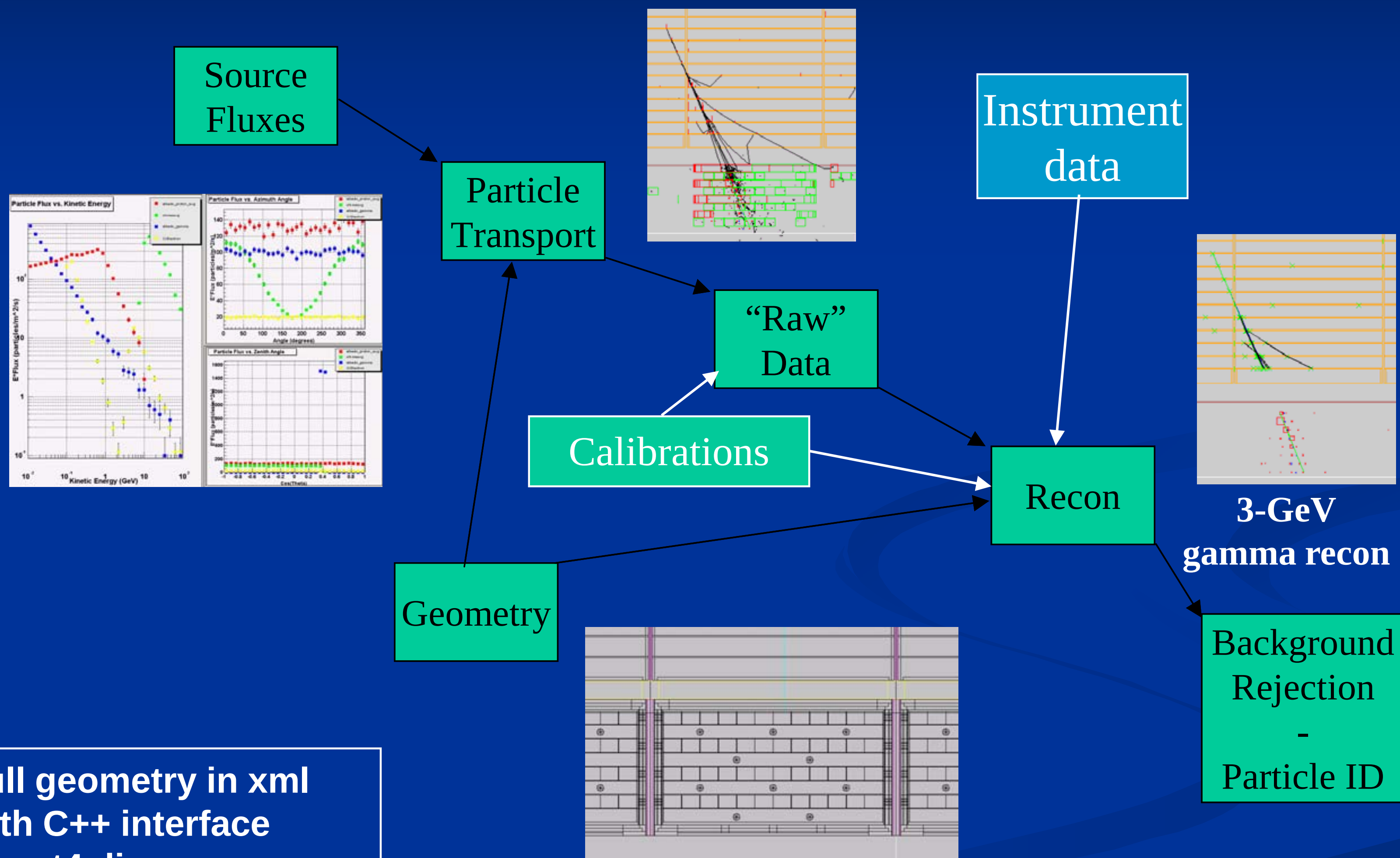


X Projected Angle
3-cm spacing, 4% foils, 100-200 MeV



Published in NIM A446(2000), 444.

Flow of Simulation/Reconstruction



CAL Detail

Full geometry in xml with C++ interface Geant4 discovers instrument from the xml

The role of simulation

Simulation plays a fundamental role in various domains and phases of an experimental physics project

- *design of the experimental set-up*
- *evaluation and definition of the potential physics output of the project*
- *evaluation of potential risks to the project*
- *assessment of the performance of the experiment*
- *development, test and optimisation of reconstruction and physics analysis software*
- *contribution to the calculation and validation of physics results*

The scope of Geant4 encompasses the simulation of the passage of particles through matter

- *there are other kinds of simulation components, such as physics event generators, detector/electronics response generators, etc.*
- *often the simulation of a complex experiment consists of several of these components interfaced to one another*

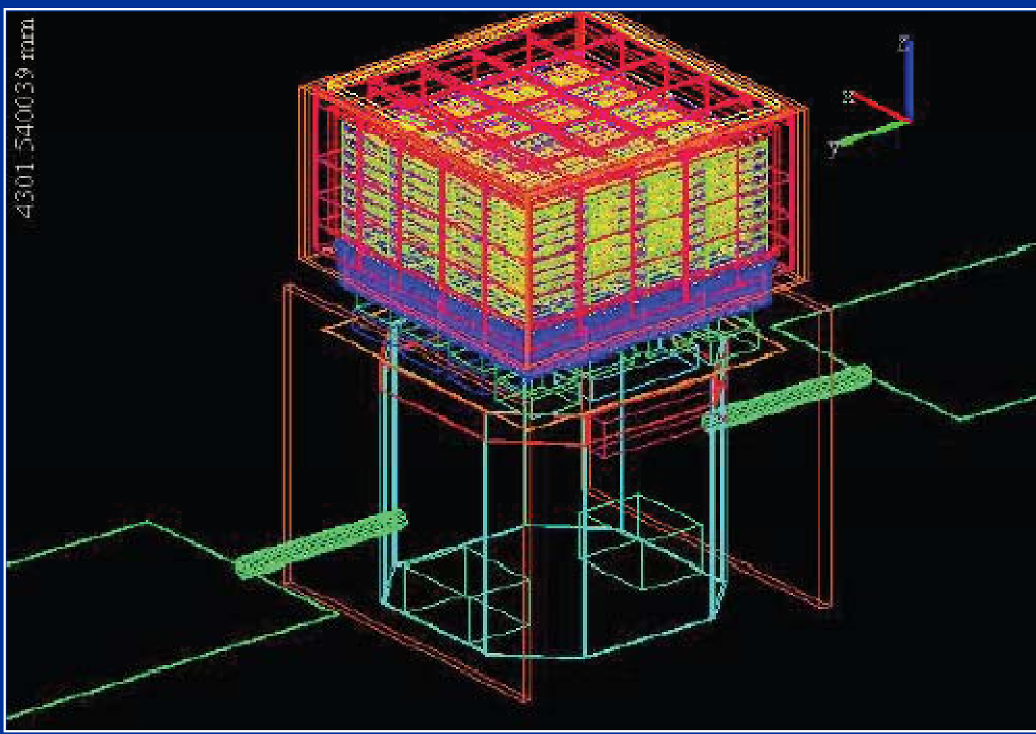
What is required

- Modeling the experimental set-up
- Tracking particles through matter
- Interaction of particles with matter
- Modeling the detector response
- Run and event control
- Accessory utilities *(random number generators, PDG particle information, physical constants, system of units etc.)*
- User interface
- Interface to event generators
- Visualisation *(of the set-up, tracks, hits etc.)*
- Persistency
- Analysis

Simulation and validation

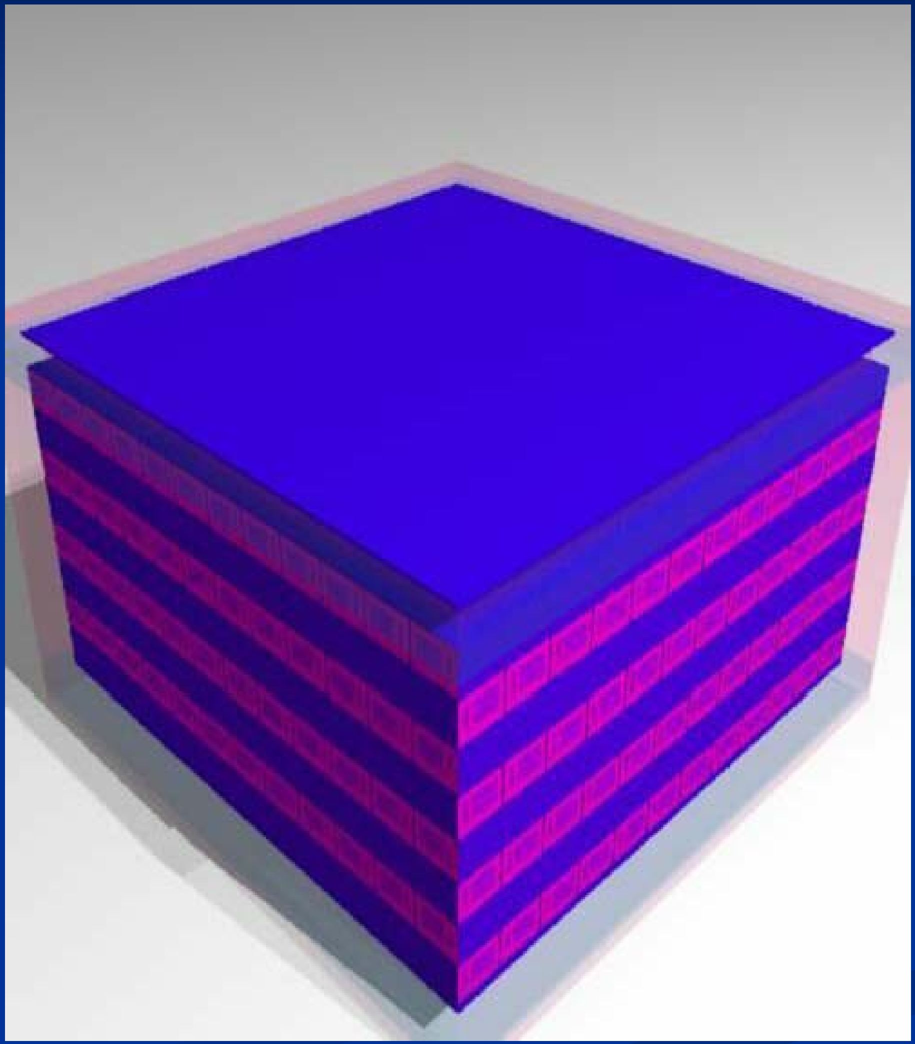
- Accurate detector model
  - >45k volumes
- Physical interactions modeled with Geant4

- MC validation
  - ground test with CR muons
  - beam test
    - 100M evts of γ, e, p, e+, C, Xe between 50MeV and 300GeV colleted at CERN and GSI in 2006

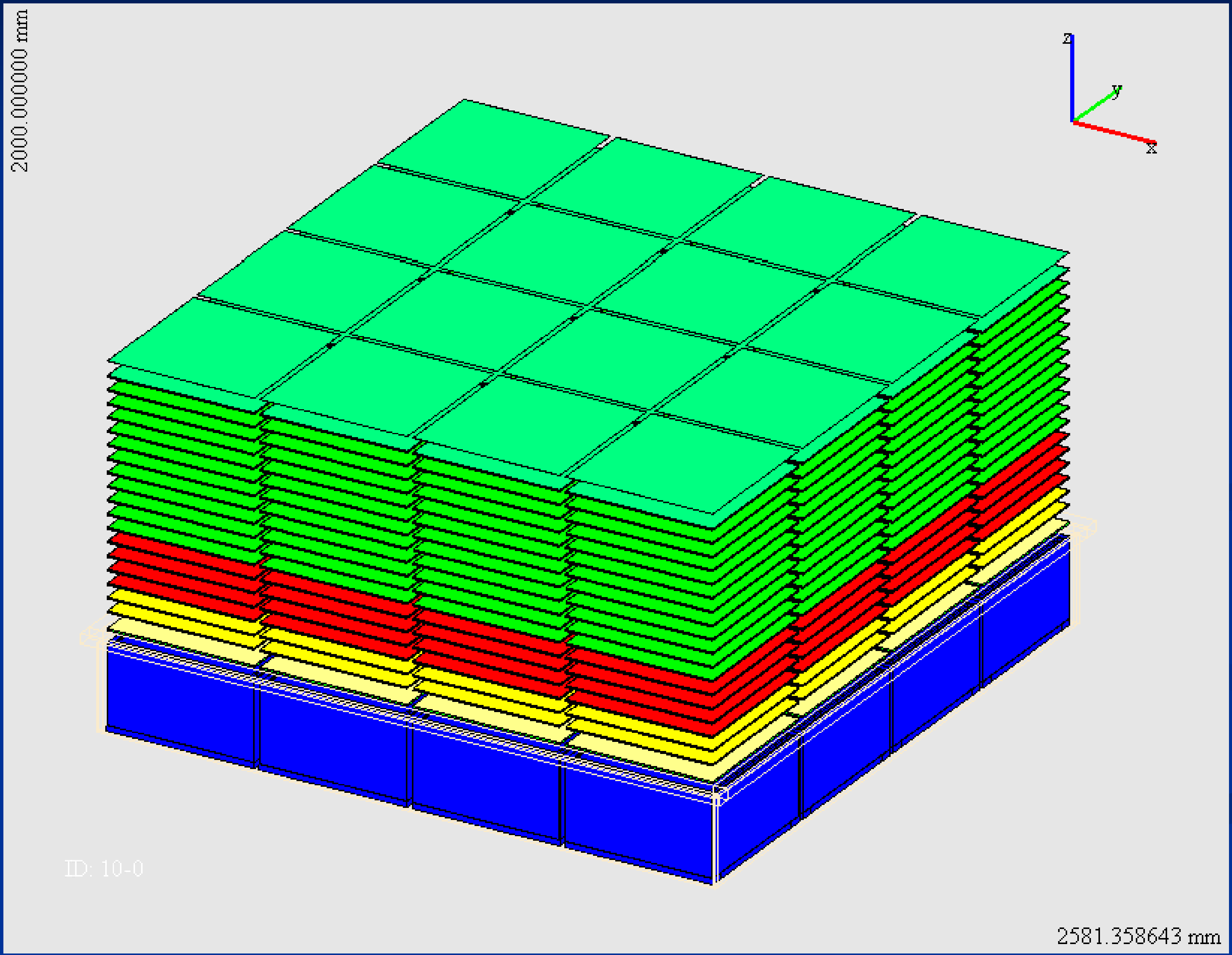


Geometry Repository

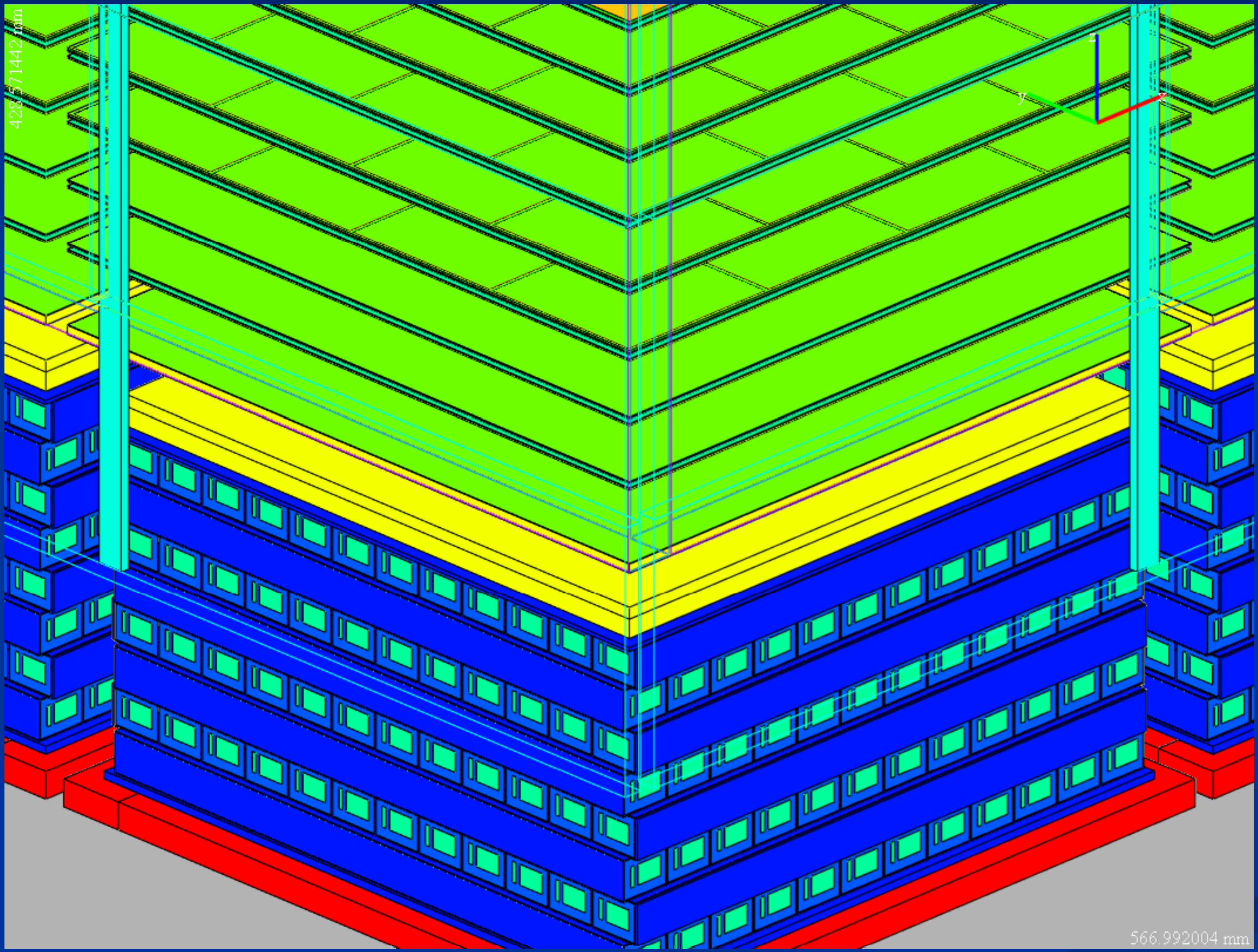
- A complex instrument: Fermi geometry is formed by more than 40000 different kind of volumes.
  - A typical problem in HEP experiments (and Fermi is not so big)
- The geometry database in Fermi is in XML, a quite common choice (LHCb, Atlas and more) today
- detModel is a set of C++ classes to parse and represent in memory such XML description
- Used by various clients (reconstruction algorithms, MonteCarlo simulation, graphics etc. etc)



LAT Geometry
Simulation and Reconstruction

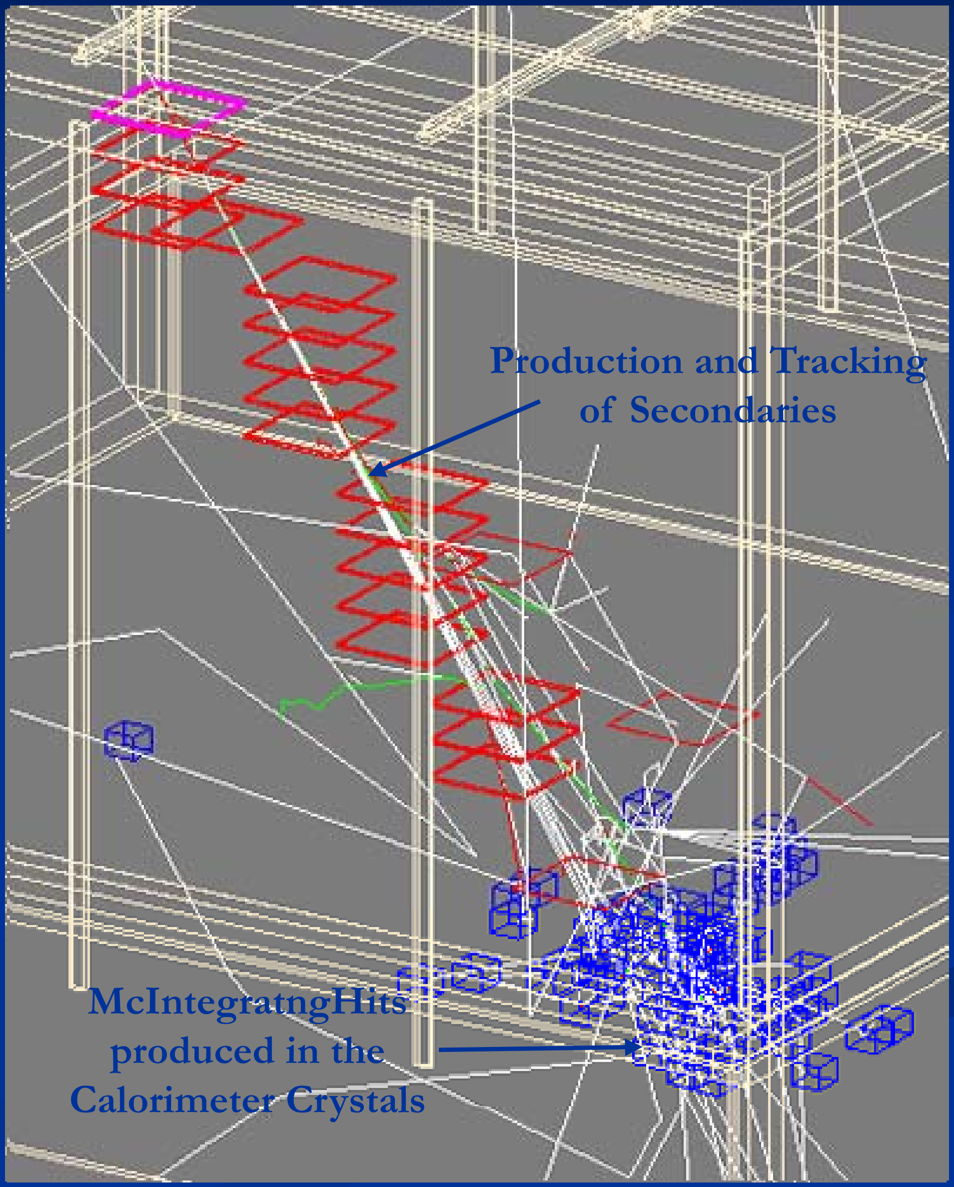


LAT Geometry
Simulation and Reconstruction



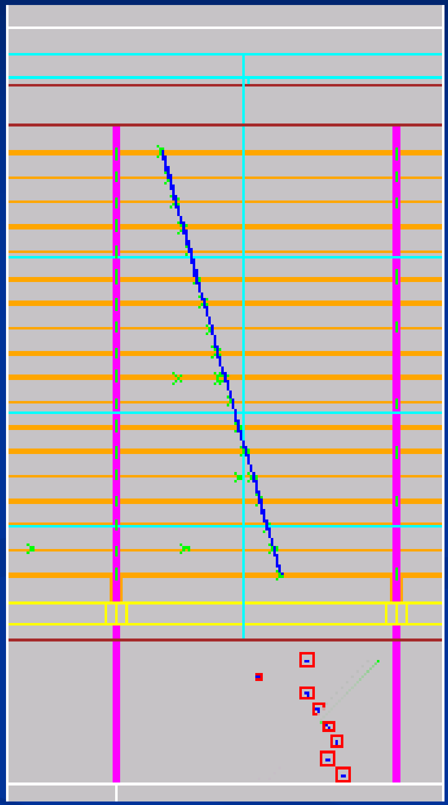
Overview of Geant4 Simulation

- Geant4 determines the evolution of the event
  - G4 given "source" which is propagated through the LAT
  - G4 output to TDS:
    - McParticles – particles produced during trackinig
    - McPositionHits – positions and energy deposited in Silicon and ACD
    - McIntegratingHits – energy deposited in "sub-cubes" of CsI crystals
    - Passed on to Digitization stage
- We use G4 v8.0.p01
  - "Vanilla" G4 for everything…
  - However, we continue to experience "problems" with the multiple scattering
    - Native G4 v6.2.2 Multiple Scattering "too small"
    - Continue to use formalism circa G4 v3



Event Reconstruction

- Sequence of operations, each implemented by one or more Algorithms, using TDS for communication
  - Trigger analysis: is there a valid trigger?
  - Preliminary CAL to find seed for tracker
  - Tracker recon: pattern recognition and fitting to find tracks and then photons in the tracker (uses Kalman filter)
  - Full CAL recon: finds clusters to estimate energies and directions
    - Must deal with significant energy leakage since only 8.5 X_0 thick
  - ACD recon: associate tracks with hit tiles to allow rejection of events in which a tile fired in the vicinity of a track extrapolation
  - Background rejection: consistency of patterns:
    - Hits in tracker
    - Shower in CAL: alignment with track, consistency with EM shower



An easy rejection

A simple example

Geant4 examples

- A number of ready-to-use examples are available in Geant4
- Located in **`$G4INSTALL/examples`**
- Three categories
  - Novice : basic functionalities of Geant4
  - Extended : specific functionalities (specific Physics processes, biasing, magnetic fields…)
  - Advanced : full simulation of realistic use cases (medical physics, space, calorimetry…)

Running without macros

- Once compiled and linked, you can run the executable using simply :

$G4WORKDIR/bin/Linux-g++/exampleN03

- You get the following screen output

```
*************************************************************
 Geant4 version Name: geant4-09-01-patch-01    (25-January-2008)
                      Copyright : Geant4 Collaboration
                      Reference : NIM A 506 (2003), 250-303
                            WWW : http://cern.ch/geant4
*************************************************************



***** Table : Nb of materials = 13 *****
[...]
-------------------------------------------------------------
---> The calorimeter is 10 layers of: [ 10mm of Lead + 5mm of liquidArgon ]
-------------------------------------------------------------

[...]
You have successfully registered the following graphics systems.
Current available graphics systems are:
  ASCIITree (ATree)
[...]
Idle>
```

Running without macros

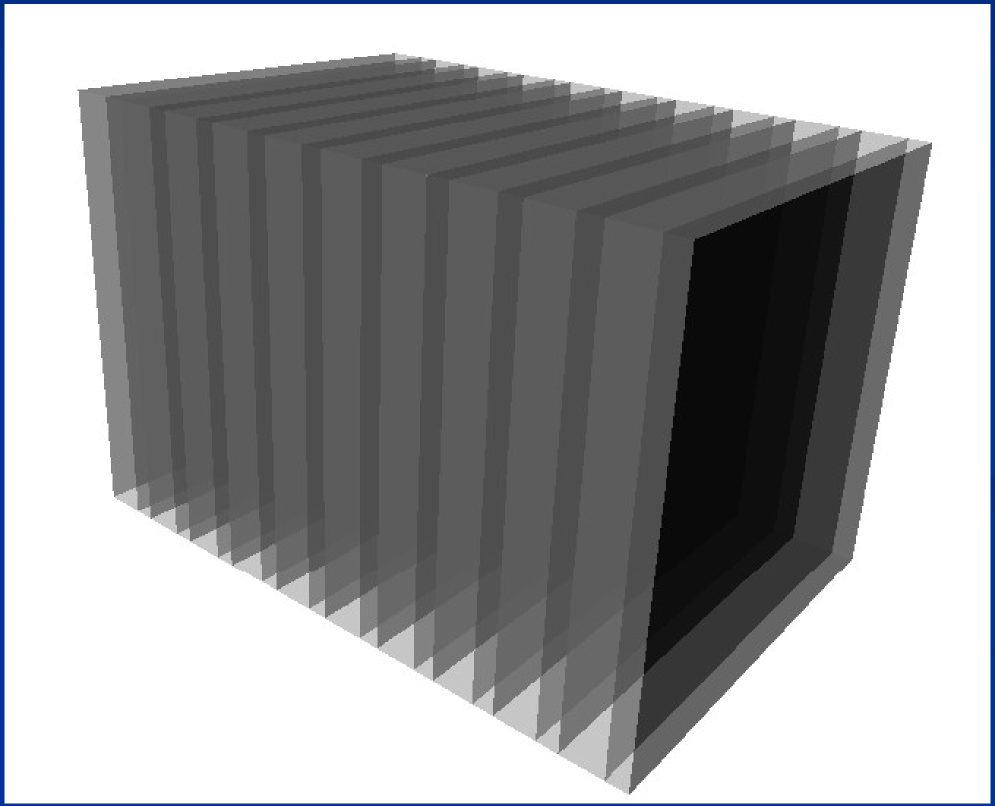
What happened ?

1. the run is initialized
   1. definition of materials,
   2. build of geometry
   3. set physics processes
   4. set production cuts
   5. ...

2. a macro file vis.mac is automatically read to register the visualization drivers (default OGLIX) and the set-up is shown on a graphic window (OpenGL)

3. you get the **`Idle>`** prompt where you can enter commands interactively
   - e.g. change geometry, decide which particle to shoot, which energy, execute another macro, shoot a particle, ..

The default geometry



What you get with the VRMLFILE visualization driver

- 10 layers : 10 mm Lead + 5 mm Liquid Argon
- no magnetic field

Running without macros

Let's try to shoot a particle:

Idle> /run/beamOn 1

By default, a 50 MeV e^- is shot impinging perpendicularly on the calorimeter

```
phot: Total cross sections from Sandia parametrisation.
Sampling according PhotoElectric model
[...]

========= Table of registered couples ========================
Index : 1 used in the geometry : Yes recalculation needed : No
Material : Lead
cuts : gamma 1 mm e- 1 mm e+ 1 mm
Energy thresholds : gamma 100.511 keV e- 1.37814 MeV e+ 1.28002 MeV
Region(s) which use this couple :
DefaultRegionForTheWorld
```

Initialization of physics tables

Calculation of energy cuts

Running without macros

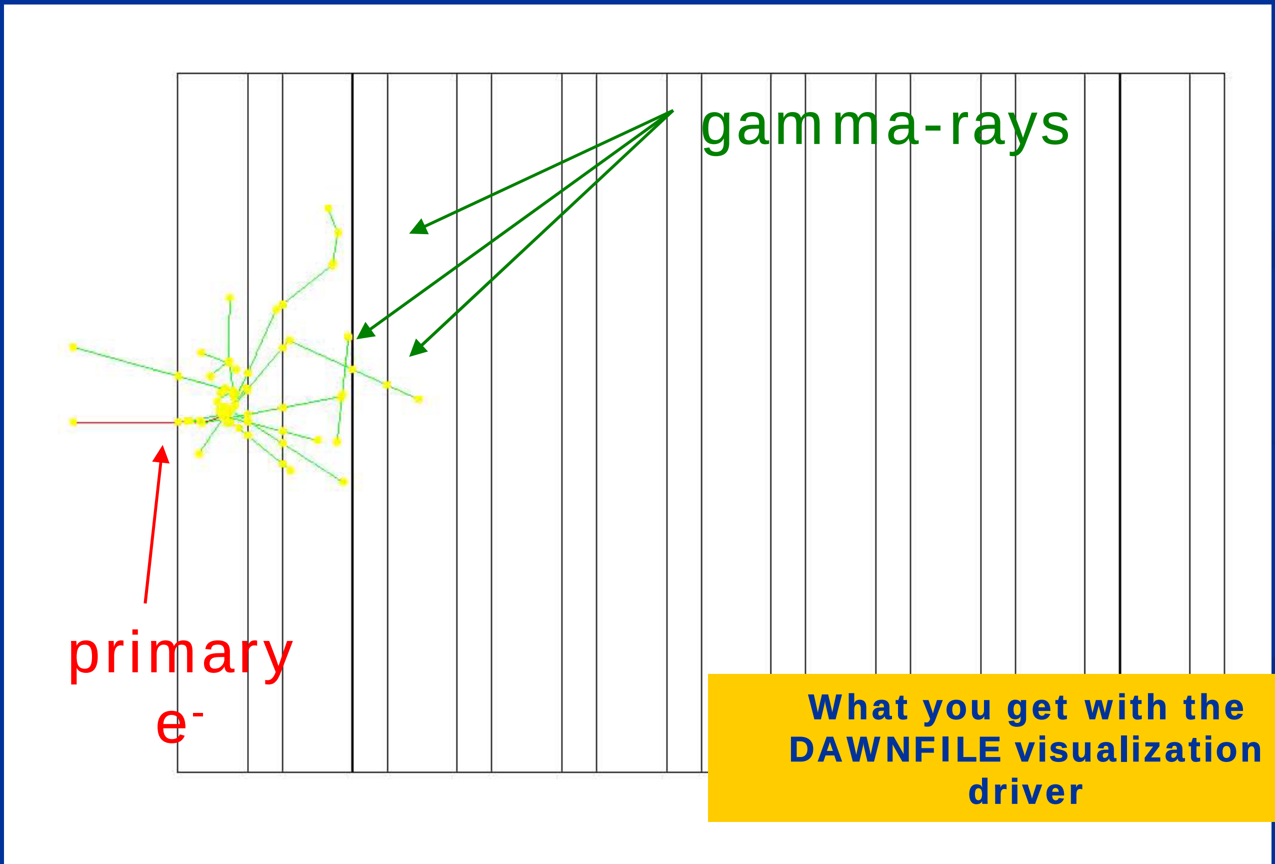
```
---------------------------------------
---> End of event: 0
   Absorber: total energy: 44.555829 MeV       total track length: 2.9979304 cm
        Gap: total energy: 809.13499 keV       total track length: 3.6520688 mm
Run terminated.
Run Summary
  Number of events processed : 1
  User=0.01s Real=0.35s Sys=0.09s

--------------------End of Run-------------------------------

 mean Energy in Absorber : 44.555829 MeV +- 0.12511652 eV
 mean Energy in Gap      : 809.13499 keV +- 0.005999626 eV

 mean trackLength in Absorber : 2.9979304 cm  +- 2.0295141 Ang
 mean trackLength in Gap      : 3.6520688 mm  +- 0 fm
------------------------------------------------------------
```

Event summary

Run summary

Running without macros

You also get a visualization of the event you have just shot (50 MeV e^-)



What you get with the DAWNFILE visualization driver

Default color code:

red = negative charge

blue = positive charge

green = neutral

Running with macros

The argument following the executable name is taken as a macro name, e.g. run1.mac

$G4WORKDIR/bin/Linux-g++/exampleN03 run1.mac

Geant4 macros are ASCII files containing a sequence of Geant4 commands:

```
#
/run/verbose 2
/event/verbose 0
/tracking/verbose 1
#
/gun/particle mu+
/gun/energy 300 MeV
/run/beamOn 3
```

Shoot 3 μ^+ of energy 300 MeV

Running with macros

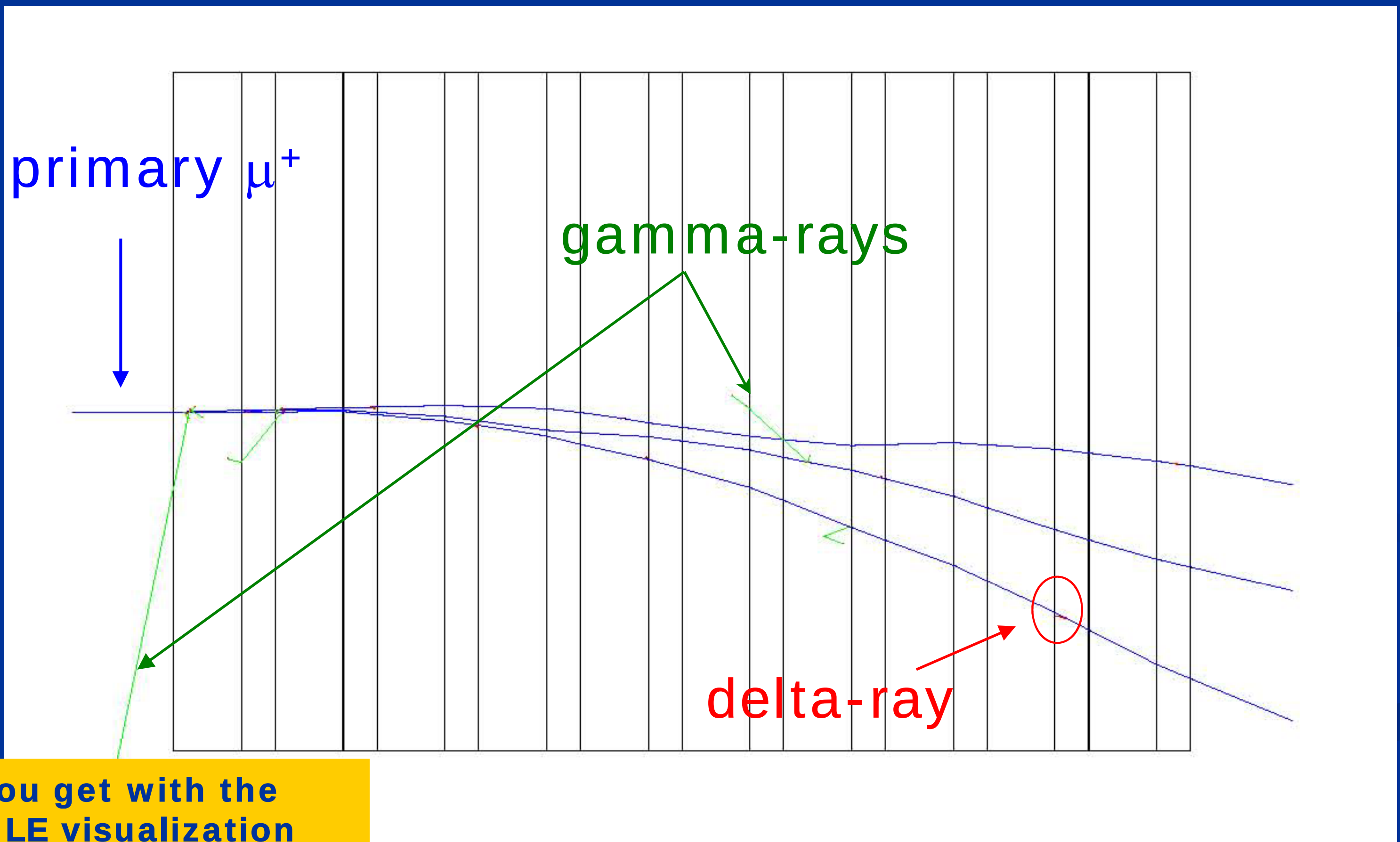
```
---> End of event: 2
   Absorber: total energy: 125.1375 MeV       total track length: 10.637355 cm
        Gap: total energy: 9.5302526 MeV       total track length: 5.3417345 cm
Run terminated.
Run Summary
  Number of events processed : 3
  User=0.01s Real=3.78s Sys=0.06s

--------------------End of Run-------------------------------

 mean Energy in Absorber : 133.48424 MeV +- 7.4232944 MeV
 mean Energy in Gap      : 9.8882883 MeV +- 274.14222 keV

 mean trackLength in Absorber : 10.969526 cm  +- 4.7680992 mm
 mean trackLength in Gap      : 5.3210426 cm  +- 640.82289 um
------------------------------------------------------------
```

Summary of event #2 (the 3rd one!)

Summary of the full run

Running with macros

Screenshot of the 3 events



What you get with the DAWNFILE visualization driver

Running with macros

Note that

```
$G4WORKDIR/bin/Linux-g++/exampleN03 run1.mac
```

is equivalent to

```
$G4WORKDIR/bin/Linux-g++/exampleN03
[...]
Idle> /control/execute run1.mac
```

⬆ command to run an external macro

(but in the second case you will get the **Idle>** prompt back)

Change geometry on-the-fly

```
$G4WORKDIR/bin/Linux-g++/exampleN03

Idle> /control/execute newgeom.mac

Idle> /control/execute run1.mac
```

1) First macro changes geometry:

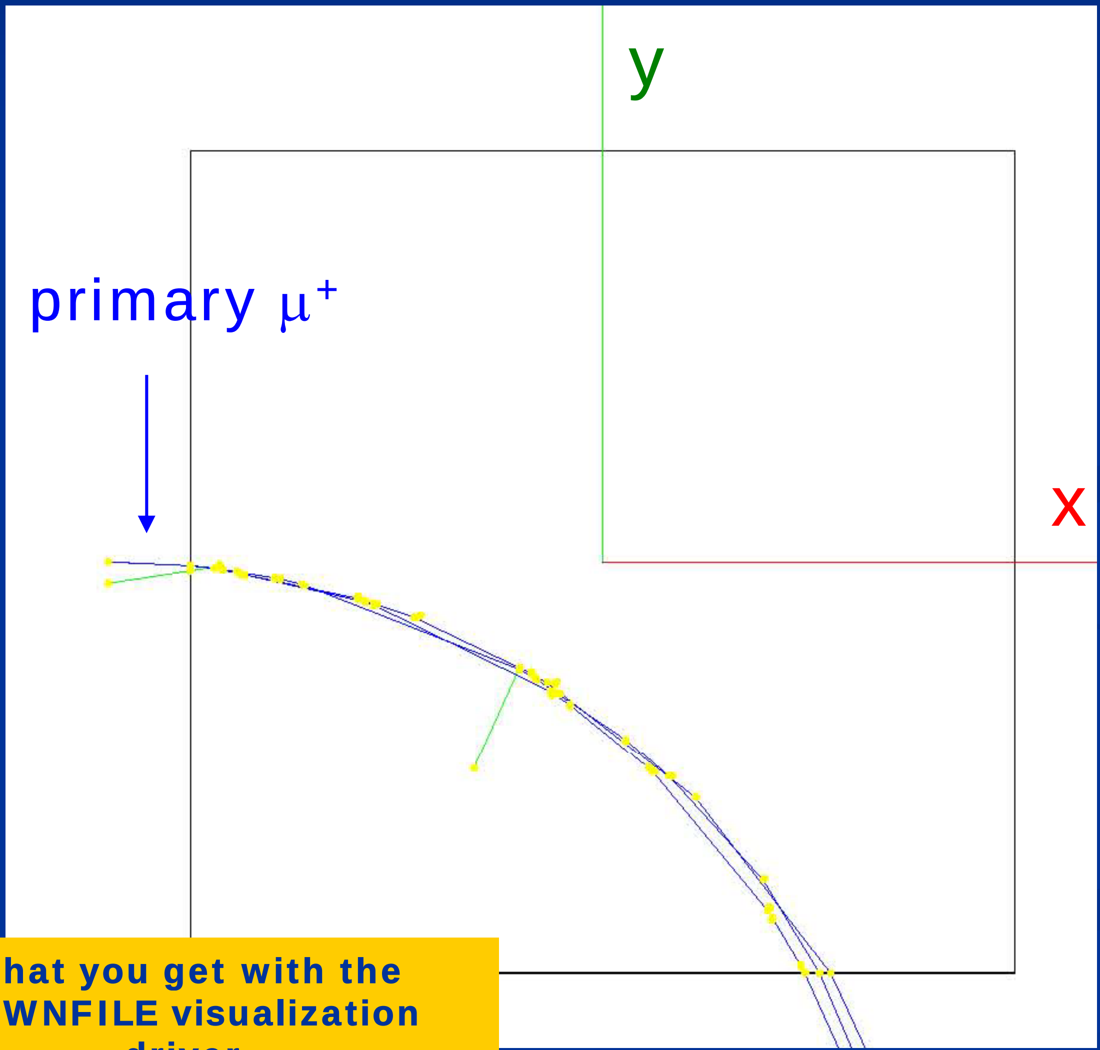
only one layer of absorber (40 cm of water), no gap (thickness = 0 cm) → practically a **solid block of water**

Change transverse dimensions, set a 3 T magnetic field

```
/N03/det/setNbOfLayers 1
/N03/det/setAbsMat Water
/N03/det/setAbsThick 40 cm
/N03/det/setGapMat Air
/N03/det/setGapThick  0 cm
/N03/det/setSizeYZ   40 cm
/N03/det/setField 3 tesla
/N03/det/update
```

2) Second macro shoots the 3 300-MeV μ^+, as before

Change geometry on-the-fly



What you get with the DAWNFILE visualization driver

Geometry, materials, magnetic field and primary particles can be tuned by ASCII macros, without recompiling the code !

User classes

To use Geant4, you have to…

- Geant4 is a toolkit. You have to build an application.
- To make an application, you have to
  - Define your geometrical setup
    - Material, volume
  - Define physics to get involved
    - Particles, physics processes/models
    - Production thresholds
  - Define how an event starts
    - Primary track generation
  - Extract information useful to you
- You may also want to
  - Visualize geometry, trajectories and physics output
  - Utilize (Graphical) User Interface
  - Define your own UI commands
  - etc.

User classes

- main()
  - Geant4 does not provide *main().*
- Initialization classes
  - Use G4RunManager::SetUserInitialization() to define.
  - Invoked at the initialization
    - G4VUserDetectorConstruction
    - G4VUserPhysicsList
- Action classes
  - Use G4RunManager::SetUserAction() to define.
  - Invoked during an event loop
    - G4VUserPrimaryGeneratorAction
    - G4UserRunAction
    - G4UserEventAction
    - G4UserStackingAction
    - G4UserTrackingAction
    - G4UserSteppingAction

Note : classes written in yellow are mandatory.

The main program

- Geant4 does not provide the *main*().
- In your *main(),* you have to
  - Construct G4RunManager (or your derived class)
  - Set user mandatory classes to RunManager
    - G4VUserDetectorConstruction
    - G4VUserPhysicsList
    - G4VUserPrimaryGeneratorAction
- You can define VisManager, (G)UI session, optional user action classes, and/or your persistency manager in your *main()*.

Describe your detector

- Derive your own concrete class from G4VUserDetectorConstruction abstract base class.
- In the virtual method *Construct()*,
  - Instantiate all necessary materials
  - Instantiate volumes of your detector geometry
  - Instantiate your sensitive detector classes and set them to the corresponding logical volumes
- Optionally you can define
  - Regions for any part of your detector
  - Visualization attributes (color, visibility, etc.) of your detector elements

Select physics processes

- Geant4 does not have any default particles or processes.
  - Even for the particle transportation, you have to define it explicitly.
- Derive your own concrete class from G4VUserPhysicsList abstract base class.
  - Define all necessary particles
  - Define all necessary processes and assign them to proper particles
  - Define cut-off ranges applied to the world (and each region)
- Geant4 provides lots of utility classes/methods and examples.
  - "Educated guess" physics lists for defining hadronic processes for various use-cases.

Generate primary event

- Derive your concrete class from G4VUserPrimaryGeneratorAction abstract base class.
- Pass a G4Event object to one or more primary generator concrete class objects which generate primary vertices and primary particles.
- Geant4 provides several generators in addition to the G4VPrimaryParticlegenerator base class.
  - G4ParticleGun
  - G4HEPEvtInterface, G4HepMCInterface
    - Interface to /hepevt/ common block or HepMC class
  - G4GeneralParticleSource
    - Define radioactivity

Optional user action classes

- All user action classes, methods of which are invoked during “Beam On”, must be constructed in the user’s *main*() and must be set to the RunManager.
- G4UserRunAction
  - G4Run* GenerateRun()
    - Instantiate user-customized run object
  - void BeginOfRunAction(const G4Run*)
    - Define histograms
  - void EndOfRunAction(const G4Run*)
    - Analyze the run
    - Store histograms
- G4UserEventAction
  - void BeginOfEventAction(const G4Event*)
    - Event selection
  - void EndOfEventAction(const G4Event*)
    - Output event information

Optional user action classes

- G4UserStackingAction
  - void PrepareNewEvent()
    - Reset priority control
  - G4ClassificationOfNewTrack ClassifyNewTrack(const G4Track*)
    - Invoked every time a new track is pushed
    - Classify a new track -- priority control
      - Urgent, Waiting, PostponeToNextEvent, Kill
  - void NewStage()
    - Invoked when the Urgent stack becomes empty
    - Change the classification criteria
    - Event filtering (Event abortion)

Optional user action classes

- G4UserTrackingAction
  - void PreUserTrackingAction(const G4Track*)
    - Decide trajectory should be stored or not
    - Create user-defined trajectory
  - void PostUserTrackingAction(const G4Track*)
    - Delete unnecessary trajectory
- G4UserSteppingAction
  - void UserSteppingAction(const G4Step*)
    - Kill / suspend / postpone the track
    - Draw the step (for a track not to be stored as a trajectory)

Let me remind you...

- Define material and geometry
  - ➔ G4VUserDetectorConstruction
    - Material and Geometry lectures
- Select appropriate particles and processes and define production threshold(s)
  - ➔ G4VUserPhysicsList
    - Physics lectures
- Define the way of primary particle generation
  - ➔ G4VUserPrimaryGeneratorAction
    - Primary particle lecture
- Define the way to extract useful information from Geant4
  - ➔ G4UserSteppingAction, G4UserTrackingAction, etc.
  - ➔ G4VUserDetectorConstruction, G4UserEventAction, G4Run, G4UserRunAction
  - ➔ G4SensitiveDetector, G4VHit, G4VHitsCollection

Homework

- Review G4 web pages
- Find Appropriate documentation
- Define your own project
  - Simple geometry
  - Particle distributions
  - Scoring needs

Lecture 1
Geant4 Kernel

Basic concepts and kernel structure

Terminology (jargons)

- Run, event, track, step, step point
- Track ←→ trajectory, step ←→ trajectory point
- Process
  - At rest, along step, post step
- Cut = production threshold
- Sensitive detector, hit, hits collection

Run in Geant4

- As an analogy of the real experiment, a run of Geant4 starts with "Beam On".
- Within a run, the user cannot change
  - detector setup
  - settings of physics processes
- Conceptually, a run is a collection of events which share the same detector and physics conditions.
  - A run consists of one event loop.
- At the beginning of a run, geometry is optimized for navigation and cross-section tables are calculated according to materials appear in the geometry and the cut-off values defined.
- G4RunManager class manages processing a run, a run is represented by G4Run class or a user-defined class derived from G4Run.
  - A run class may have a summary results of the run.
- G4UserRunAction is the optional user hook.

Event in Geant4

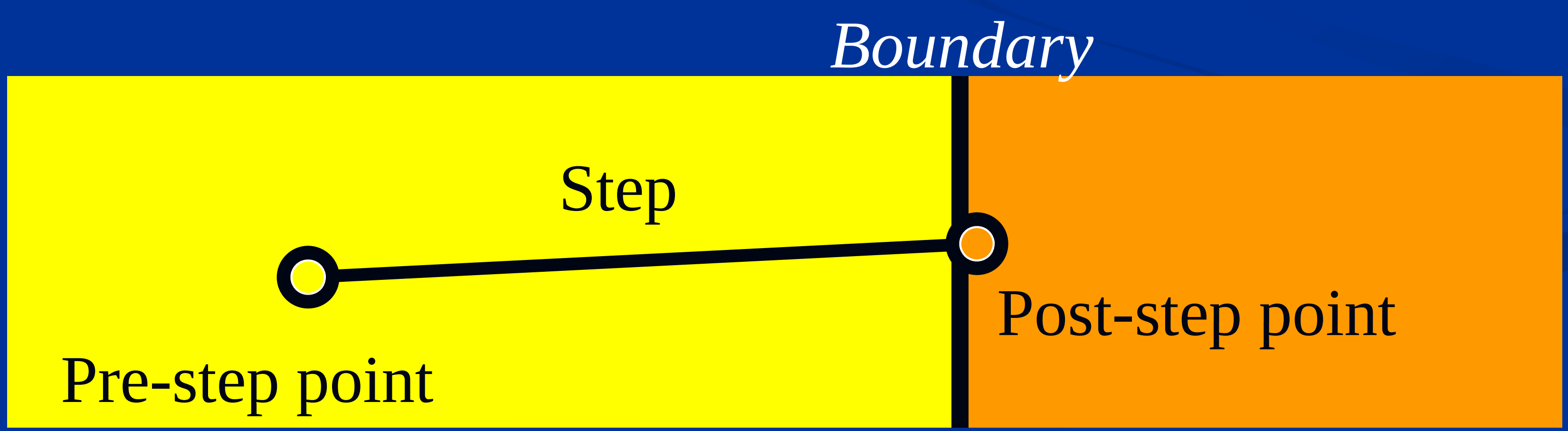
- An event is the basic unit of simulation in Geant4.
- At beginning of processing, primary tracks are generated. These primary tracks are pushed into a stack.
- A track is popped up from the stack one by one and "tracked". Resulting secondary tracks are pushed into the stack.
  - This "tracking" lasts as long as the stack has a track.
- When the stack becomes empty, processing of one event is over.
- G4Event class represents an event. It has following objects at the end of its (successful) processing.
  - List of primary vertices and particles (as input)
  - Hits and Trajectory collections (as output)
- G4EventManager class manages processing an event. G4UserEventAction is the optional user hook.

Track in Geant4

- Track is a snapshot of a particle.
  - It has physical quantities of current instance only. It does not record previous quantities.
  - Step is a “delta” information to a track. Track is not a collection of steps. Instead, a track is being updated by steps.
- Track object is deleted when
  - it goes out of the world volume,
  - it disappears (by e.g. decay, inelastic scattering),
  - it goes down to zero kinetic energy and no “AtRest” additional process is required, or
  - the user decides to kill it artificially.
- No track object persists at the end of event.
  - For the record of tracks, use trajectory class objects.
- G4TrackingManager manages processing a track, a track is represented by G4Track class.
- G4UserTrackingAction is the optional user hook.

Step in Geant4

- Step has two points and also “delta” information of a particle (energy loss on the step, time-of-flight spent by the step, etc.).
- Each point knows the volume (and material). In case a step is limited by a volume boundary, the end point physically stands on the boundary, and it logically belongs to the next volume.
  - Because one step knows materials of two volumes, boundary processes such as transition radiation or refraction could be simulated.
- G4SteppingManager class manages processing a step, a step is represented by G4Step class.
- G4UserSteppingAction is the optional user hook.



Particle in Geant4

- A particle in Geant4 is represented by three layers of classes.
- G4Track
  - Position, geometrical information, etc.
  - This is a class representing a particle to be tracked.
- G4DynamicParticle
  - "Dynamic" physical properties of a particle, such as momentum, energy, spin, etc.
  - Each G4Track object has its own and unique G4DynamicParticle object.
  - This is a class representing an individual particle.
- G4ParticleDefinition
  - "Static" properties of a particle, such as charge, mass, life time, decay channels, etc.
  - G4ProcessManager which describes processes involving to the particle
  - All G4DynamicParticle objects of same kind of particle share the same G4ParticleDefinition.

Processes in Geant4

- In Geant4, particle transportation is a process as well, by which a particle interacts with geometrical volume boundaries and field of any kind.
  - Because of this, shower parameterization process can take over from the ordinary transportation without modifying the transportation process.
- Each particle has its own list of applicable processes. At each step, all processes listed are invoked to get proposed physical interaction lengths.
- The process which requires the shortest interaction length (in space-time) limits the step.
- Each process has one or combination of the following natures.
  - AtRest
    - e.g. muon decay at rest
  - AlongStep (a.k.a. continuous process)
    - e.g. Cerenkov process
  - PostStep (a.k.a. discrete process)
    - e.g. decay on the fly

Extract useful information

- Given geometry, physics and primary track generation, Geant4 does proper physics simulation “silently”.
  - You have to add a bit of code to extract information useful to you.
- There are two ways:
  - Use user hooks (G4UserTrackingAction, G4UserSteppingAction, etc.)
    - You have an access to almost all information
    - Straight-forward, but do-it-yourself
  - Use Geant4 scoring functionality
    - Assign G4VSensitiveDetector to a volume
    - Hits collection is automatically stored in G4Event object, and automatically accumulated if user-defined Run object is used.
    - Use user hooks (G4UserEventAction, G4UserRunAction) to get event / run summary

Unit system

- Internal unit system used in Geant4 is completely hidden not only from user's code but also from Geant4 source code implementation.
- Each hard-coded number must be multiplied by its proper unit.

```
radius = 10.0 * cm;
kineticE = 1.0 * GeV;
```

- To get a number, it must be divided by a proper unit.

```
G4cout << eDep / MeV << " [MeV]" << G4endl;
```

- Most of commonly used units are provided and user can add his/her own units.
- By this unit system, source code becomes more readable and importing / exporting physical quantities becomes straightforward.
  - For particular application, user can change the internal unit to suitable alternative unit without affecting to the result.

Geant4 kernel

- Geant4 consists of 17 categories.
  - Independently developed and maintained by WG(s) responsible to each category.
  - Interfaces between categories (e.g. top level design) are maintained by the global architecture WG.
- Geant4 Kernel
  - Handles run, event, track, step, hit, trajectory.
  - Provides frameworks of geometrical representation and physics processes.