Montecarlo Methods for Medical Physics

Francesco Longo

(francesco.longo@ts.infn.it)

Summary of the Course

- Part1 (Apr 28)
  - General (and brief) introduction to Monte Carlo methods
  - Montecarlo methods in Medical Physics
- Part2 (Today – May 5)
  - Introduction to the Geant4 toolkit
  - Fundamentals of a Geant4 application
    - Physics, Geometry, Particle Flux, Scoring
- Part3 (May 5 – 12)
  - Realisation of an example relevant to Medical Physics

Evaluation for the "Laboratory"

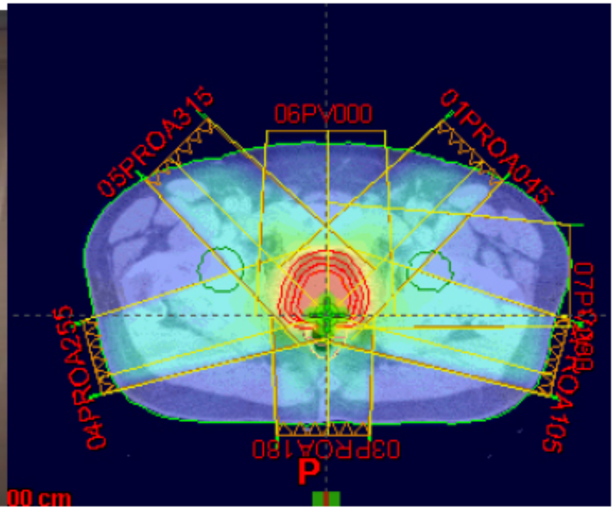
- Discussion of Geant4 example
  - G4 example relevant to Medical Physics
- Discussion of requirements and methods
  - Medical Physics "environment"
  - Geant4 modeling
  - "Basic" analysis of results
- Realization of "new" example – Laboratory
  - Might just be an improvement of an existing G4 example

Part 1

Montecarlo in Medical Physics

Montecarlo for Medical Physics

Overview of Medical Physics Applications

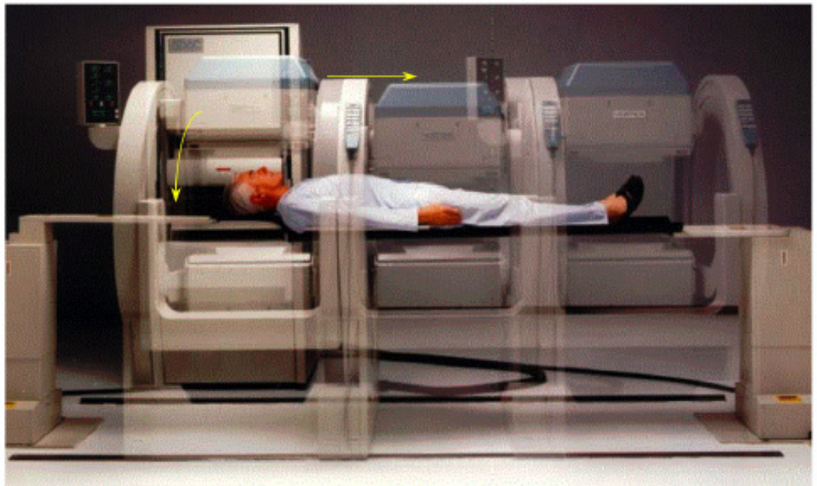


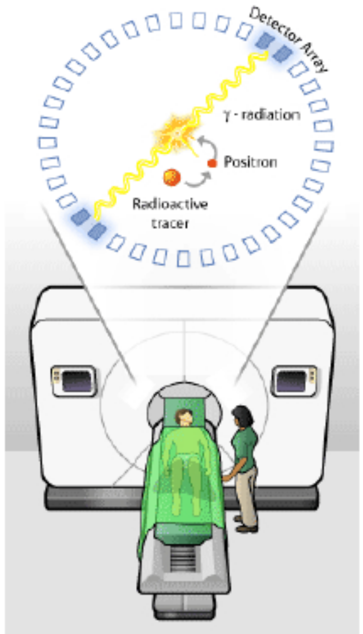
Radiotherapy physics

- external/internal sources and dosimetry
- phantom simulations
- treatment planning

Montecarlo for Medical Physics

Single Photon Emission Computer Tomography (SPECT)

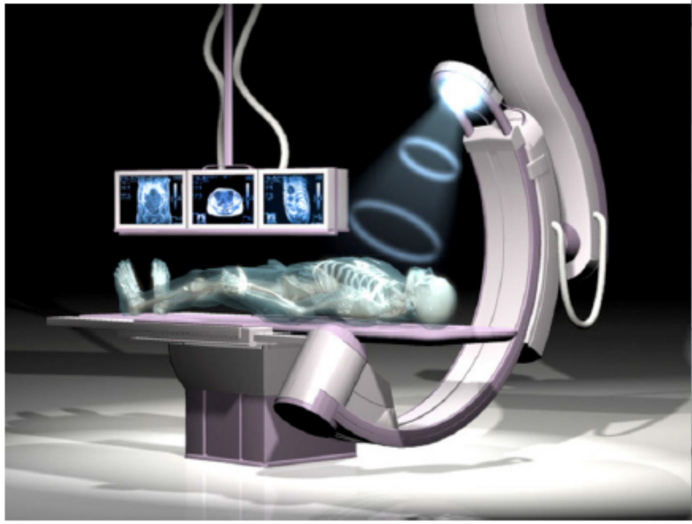




Nuclear medicine

- detectors
- imaging correction
- absorbed dose

Montecarlo for Medical Physics



Diagnostic radiology

- detection systems
- physical quantities
- radiation protection

Part 2

Introduction to Geant4

Geant 4

A Simulation Tool for Multi-disciplinary Applications

http://cern.ch/geant4/

An simple introduction to Geant4 with emphasis on medical physics.

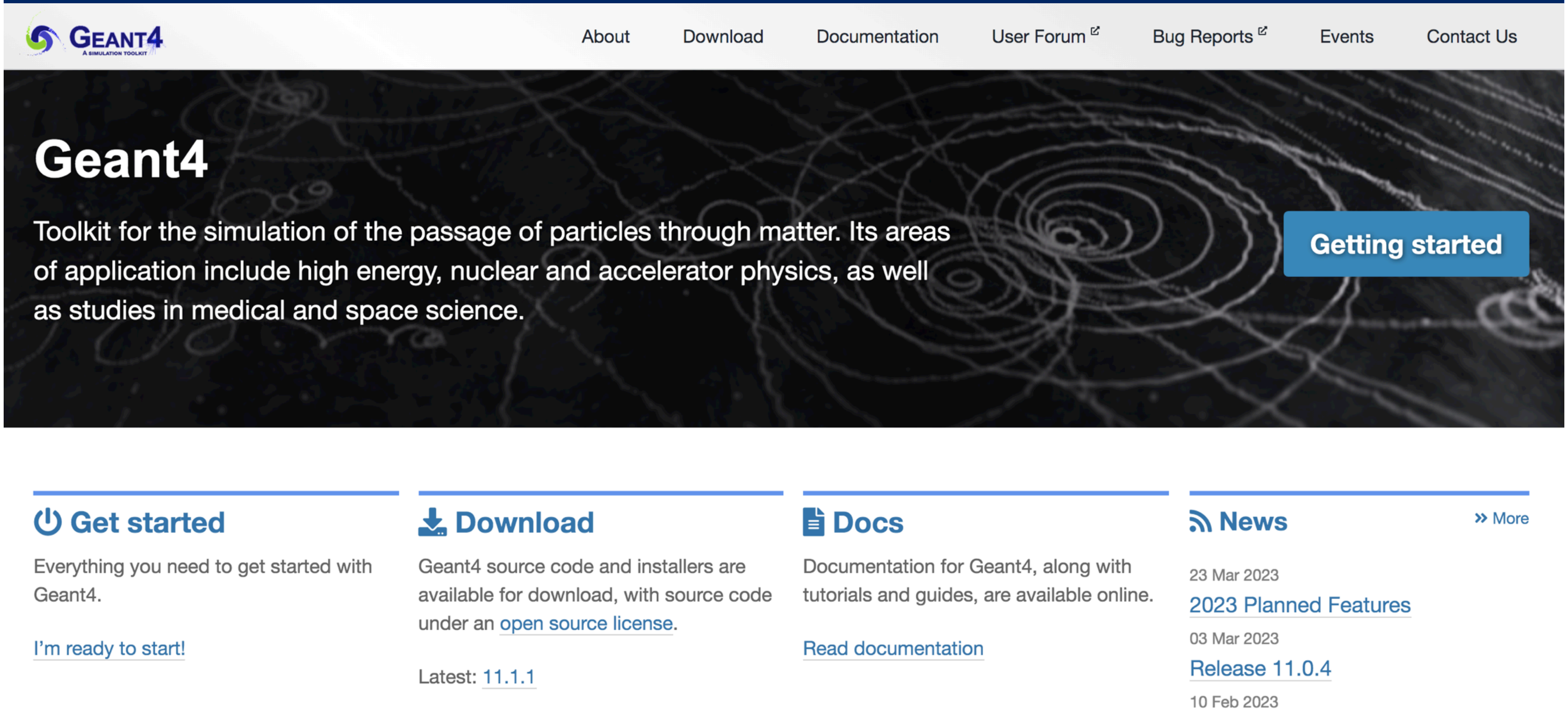
The seminar will touch some aspects of Geant4 from basic description to advanced topics.

The seminar should be of interest to complete novices with no familiarity with Geant4.

Participants are expected to have a basic knowledge of C++.

Based on Training Lectures by the G4 collaboration

The Geant4 toolkit



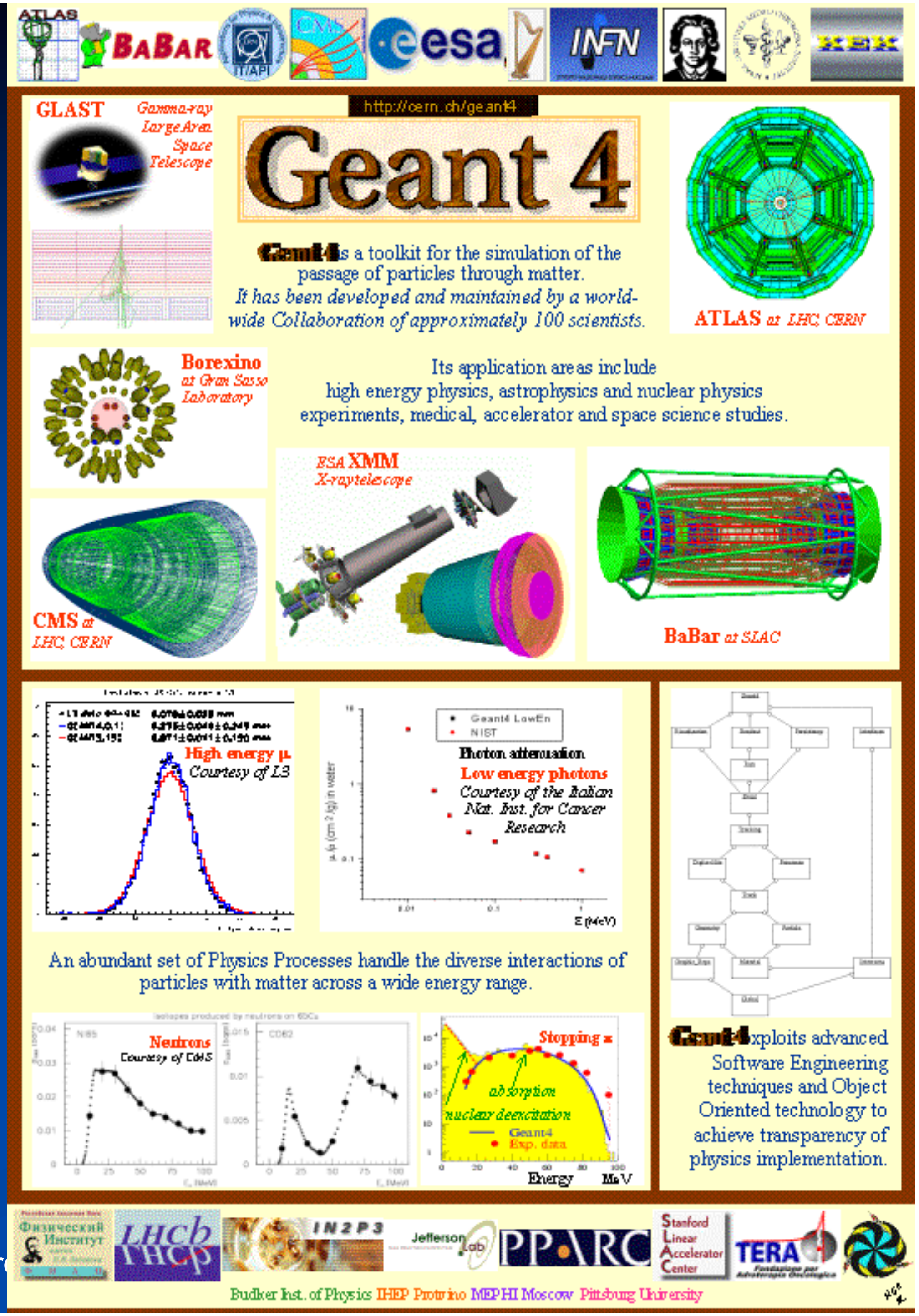
https://geant4.web.cern.ch/

Outline of Part2

- General Introduction to G4
  - What is G4 ?
  - Review of user documentation
  - Geant4 as a toolkit
- Basics of OO programming
- Geant4 Kernel and basics of the toolkit
  - Run, Event, Step
  - Particle and Physics processes
  - User classes

Geant4 toolkit

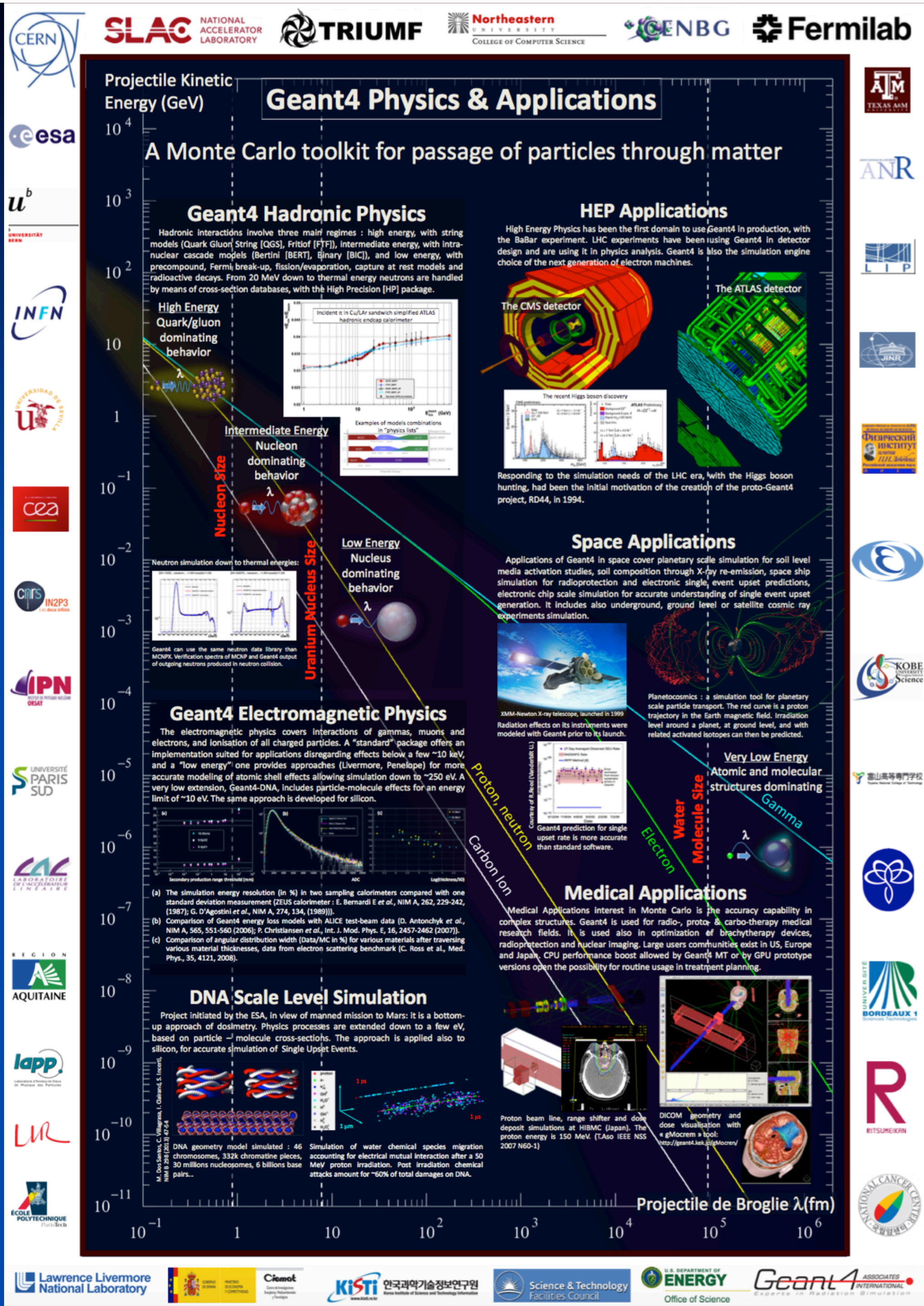
- Highlights of user applications
- General introduction and brief history
- User Documentation
- The main program



Geant4

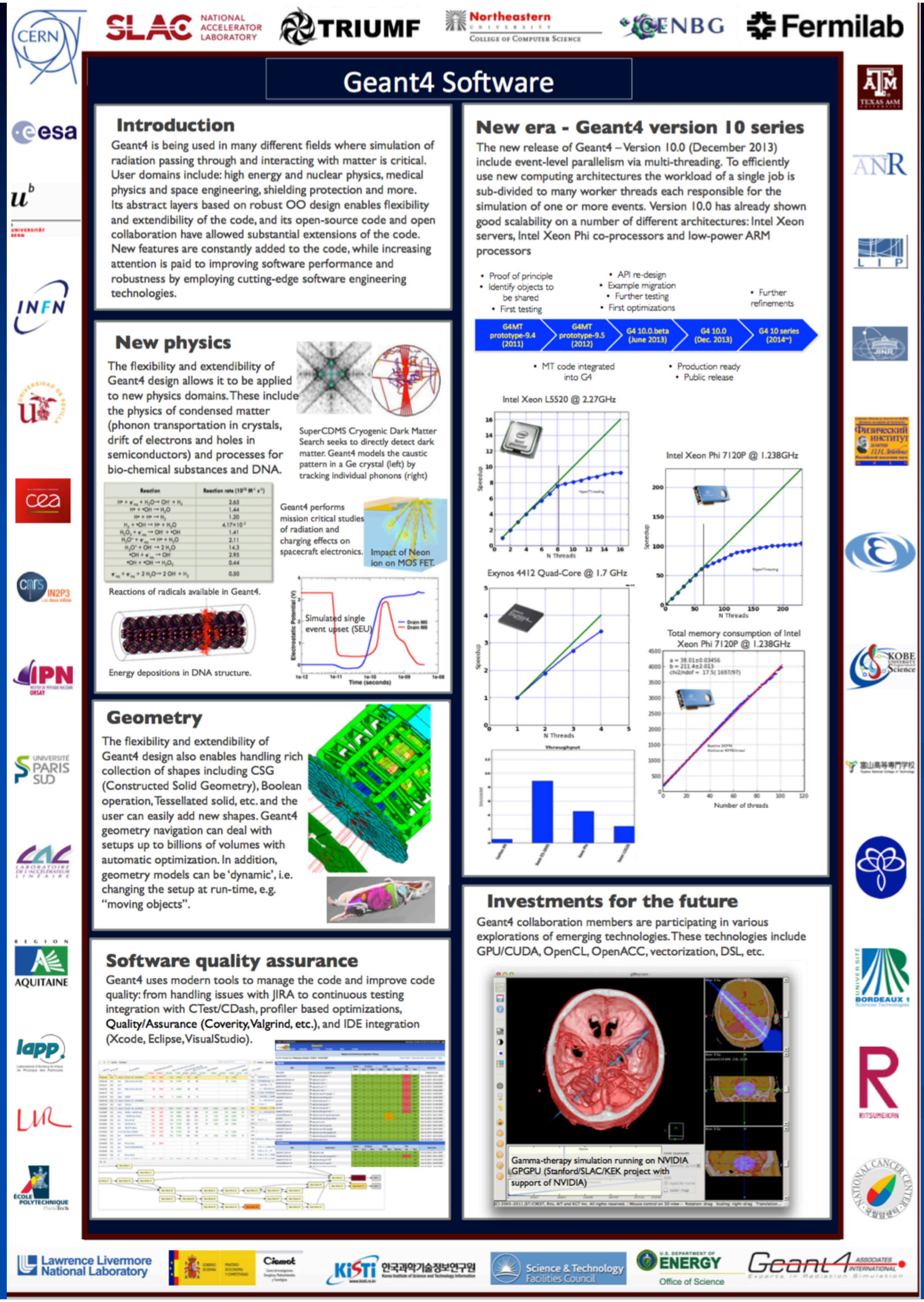

European Organization for Nuclear Research
Geant4
A toolkit to simulate the interaction of particles with matter
Collaborators also from non-member institutions, including
Budker Inst. of Physics
IHEP Protvino
MEPHI Moscow
Technical questions
J. Apostolakis
PH Department
CERN - European Organization for Nuclear Research
CH-1211 Geneva 23
Tel. +41 22 767 72 39
E-mail: John.Apostolakis@cern.ch
http://cern.ch/geant4
Licensing questions
Technology Transfer Service/ETT
CERN - European Organization for Nuclear Research
CH-1211 Geneva 23
Tel. +41 22 767 84 44
Fax: +41 22 767 35 40
E-mail: HelpDesk-TT@cern.ch
http://cern.ch/TTdb
Concept
Geant4 simulates the passage of particles through matter.
It provides a complete set of tools for all domains of radiation transport:
. Geometry and Tracking
. Physics processes and models
. Biasing and Scoring
. Graphics and User Interfaces
. Propagation in fields.
Geant4 physics processes describe electromagnetic and nuclear interactions of particles with matter, at energies from eV to TeV.
A choice of physics models exists for many processes providing options for applications with different accuracy and time requirements.
A visualization of the CMS detector (Courtesy of I. Osborne, CMS Collaboration)
The toolkit is developed, maintained and supported by Geant4, a world-wide collaboration of about 100 scientists from many institutions, contributing in their area of expertise. Developers interact constantly with users, and combine efforts to validate physics results for application in high energy physics experiments, space and medical studies.
Applications
High energy and nuclear physics detectors
. ATLAS, CMS, HARP and LHCb at CERN and BaBar at SLAC
Accelerator and shielding
. Linacs for medical use
Medicine
. Radiotherapy
. photon, proton and light ion beams
. brachytherapy
. boron and gadolinium neutron capture therapy
. Simulation of scanners
. PET & SPECT with GATE (Geant4 Application for Tomographic Emission)
Space
. Satellites
. effect of space environment on components (especially electronics)
. shielding of instruments
. charging effects
. Space environment
. cosmic ray cut-offs
. Astronauts
. dose estimates
The BepiColombo Mercury orbiter (Courtesy of ESA)
Simulation of small PET scanner using GATE (Courtesy of the OpenGATE collaboration)
A view of the ATLAS detector (Courtesy of S. Tanaka, ATLAS collaboration)
Advantages
. Simulates the geometries of complex setups efficiently
. Provides configurations of physics processes for application areas
. Enables user to tailor simulation components and address accuracy needs
. Performant and adaptable
. Easy to embed into specific applications
XMM-Newton X-ray telescope: the effects of the radiation environment on its instruments was modeled with Geant4 prior to launch in 1999 (Courtesy of ESA)
The European Organization for Nuclear Research (CERN), one of the world's foremost particle physics laboratories, has introduced an active Technology Transfer policy to establish its competence in European industrial and scientific environments, and to demonstrate clear benefits of the results obtained from the considerable resources made available to particle physics research.
Technology Transfer is an integral part of CERN's principal mission of fundamental research.
CERN Technology Transfer
http://knowledgetransfer.web.cern.ch/technology-transfer/external-partners/geant4

Geant4



Geant4 Physics & Applications
A Monte Carlo toolkit for passage of particles through matter
Projectile Kinetic Energy (GeV)
Geant4 Hadronic Physics
High Energy Quark/gluon dominating behavior
Intermediate Energy Nucleon dominating behavior
Low Energy Nucleus dominating behavior
Nucleon Size
Uranium Nucleus Size
HEP Applications
The CMS detector
The ATLAS detector
Space Applications
Geant4 Electromagnetic Physics
Very Low Energy Atomic and molecular structures dominating
Water Molecule Size
Proton, neutron
Carbon Ion
Electron
Gamma
Medical Applications
DNA Scale Level Simulation
Projectile de Broglie λ(fm)

Geant4

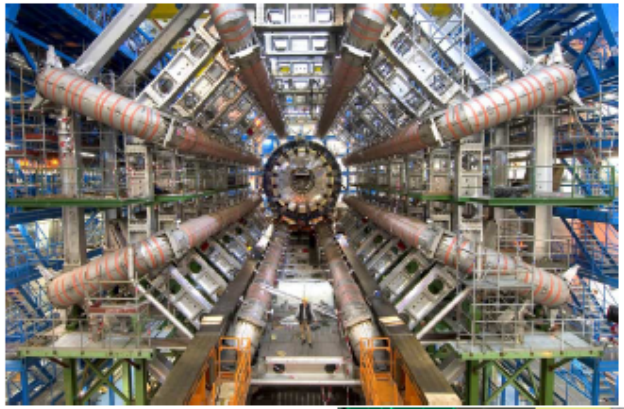


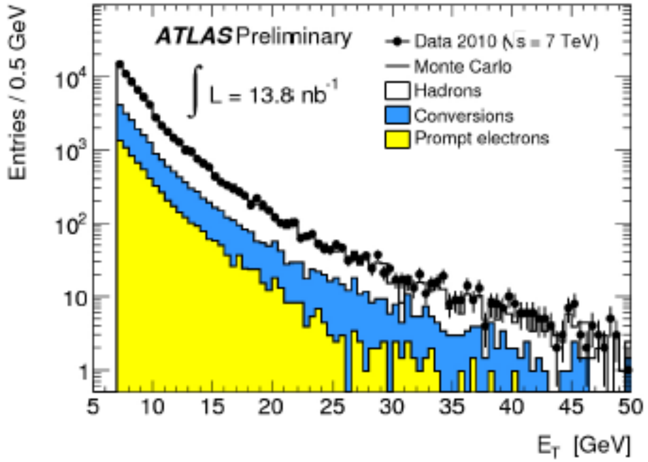
Geant4 Software
Introduction
Geant4 is being used in many different fields where simulation of radiation passing through and interacting with matter is critical. User domains include: high energy and nuclear physics, medical physics and space engineering, shielding protection and more. Its abstract layers based on robust OO design enables flexibility and extendibility of the code, and its open-source code and open collaboration have allowed substantial extensions of the code. New features are constantly added to the code, while increasing attention is paid to improving software performance and robustness by employing cutting-edge software engineering technologies.
New physics
The flexibility and extendibility of Geant4 design allows it to be applied to new physics domains. These include the physics of condensed matter (phonon transportation in crystals, drift of electrons and holes in semiconductors) and processes for bio-chemical substances and DNA.
SuperCDMS Cryogenic Dark Matter Search seeks to directly detect dark matter. Geant4 models the caustic pattern in a Ge crystal (left) by tracking individual phonons (right)
Geant4 performs mission critical studies of radiation and charging effects on spacecraft electronics.
Impact of Neon ion on MOS FET.
Reactions of radicals available in Geant4.
Simulated single event upset (SEU)
Energy depositions in DNA structure.
Geometry
The flexibility and extendibility of Geant4 design also enables handling rich collection of shapes including CSG (Constructed Solid Geometry), Boolean operation, Tessellated solid, etc. and the user can easily add new shapes. Geant4 geometry navigation can deal with setups up to billions of volumes with automatic optimization. In addition, geometry models can be 'dynamic', i.e. changing the setup at run-time, e.g. "moving objects".
Software quality assurance
Geant4 uses modern tools to manage the code and improve code quality: from handling issues with JIRA to continuous testing integration with CTest/CDash, profiler based optimizations, Quality/Assurance (Coverity, Valgrind, etc.), and IDE integration (Xcode, Eclipse, VisualStudio).
New era - Geant4 version 10 series
The new release of Geant4 – Version 10.0 (December 2013) include event-level parallelism via multi-threading. To efficiently use new computing architectures the workload of a single job is sub-divided to many worker threads each responsible for the simulation of one or more events. Version 10.0 has already shown good scalability on a number of different architectures: Intel Xeon servers, Intel Xeon Phi co-processors and low-power ARM processors
G4MT prototype-9.4 (2011)
G4MT prototype-9.5 (2012)
G4 10.0.beta (June 2013)
G4 10.0 (Dec. 2013)
G4 10 series (2014~)
Intel Xeon L5520 @ 2.27GHz
Intel Xeon Phi 7120P @ 1.238GHz
Exynos 4412 Quad-Core @ 1.7 GHz
Total memory consumption of Intel Xeon Phi 7120P @ 1.238GHz
Investments for the future
Geant4 collaboration members are participating in various explorations of emerging technologies. These technologies include GPU/CUDA, OpenCL, OpenACC, vectorization, DSL, etc.
Gamma-therapy simulation running on NVIDIA GPGPU (Stanford/SLAC/KEK project with support of NVIDIA)

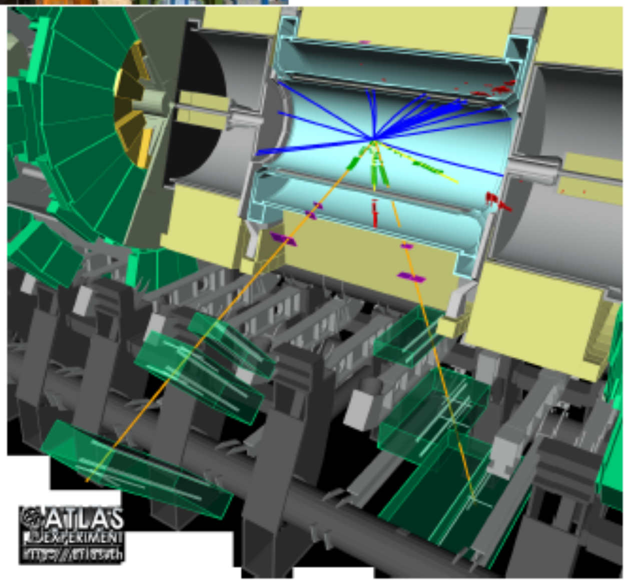
Highlights of Users Applications

To provide you some ideas how Geant4 would be utilized…

Experiments …

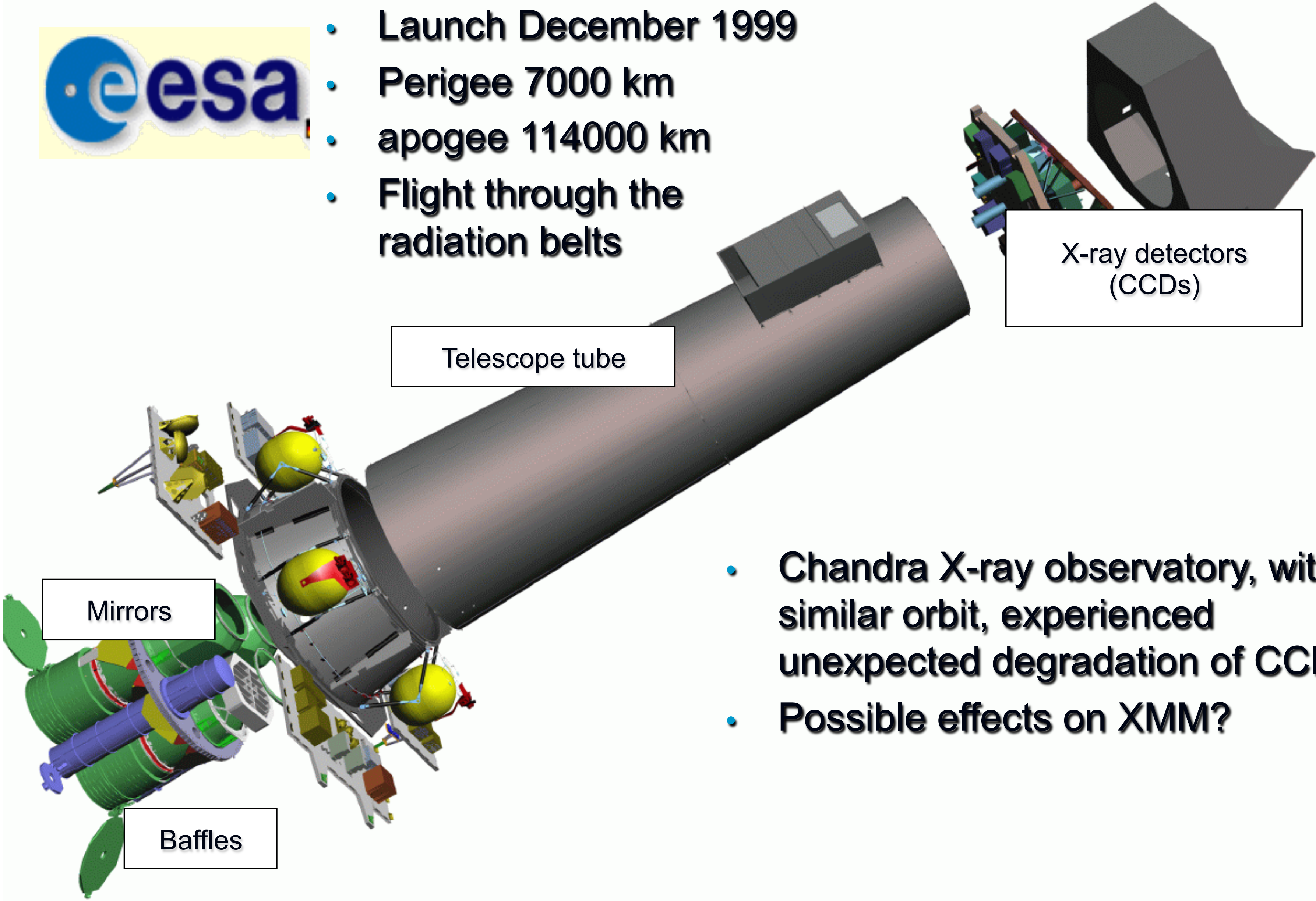




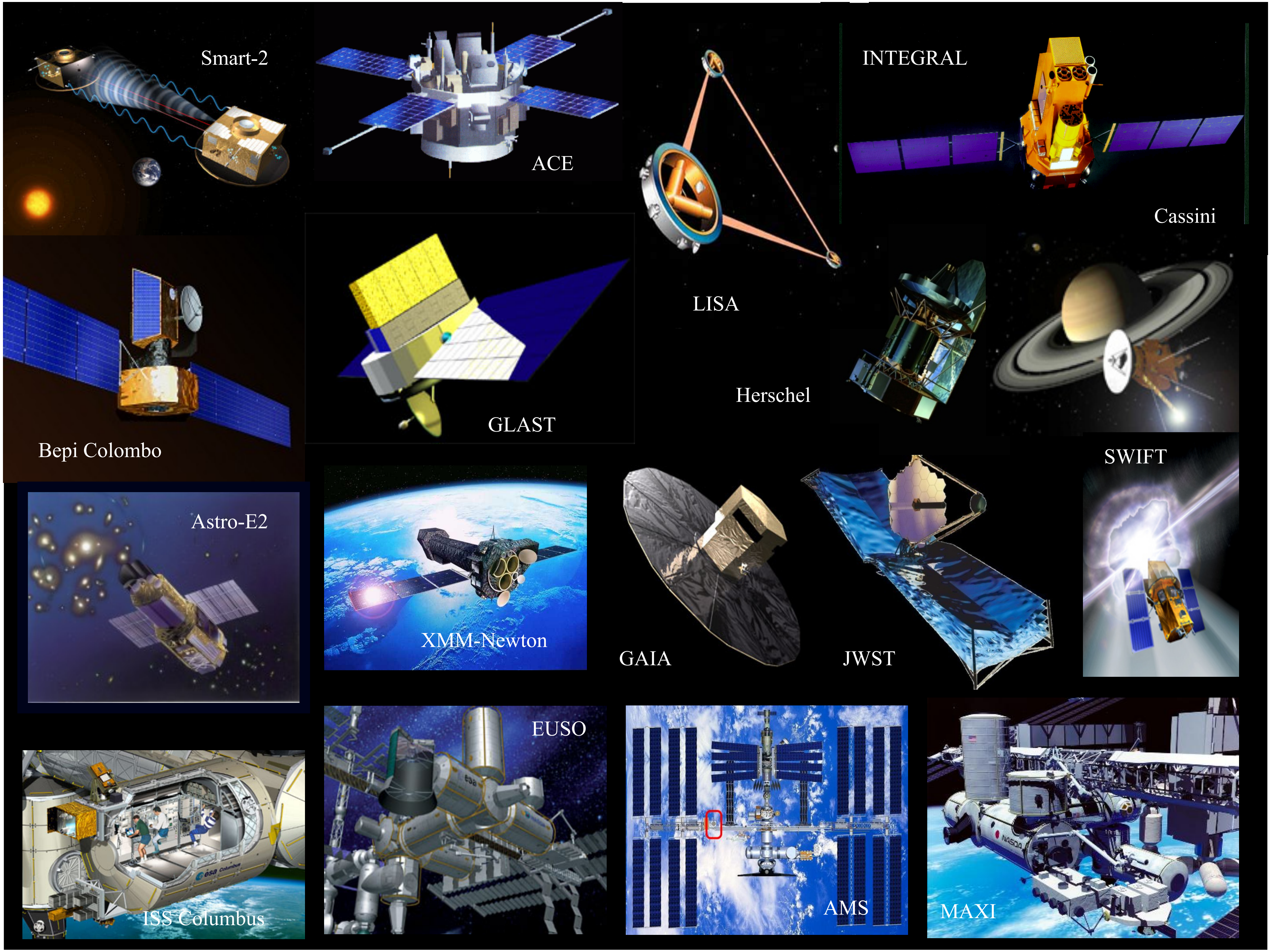


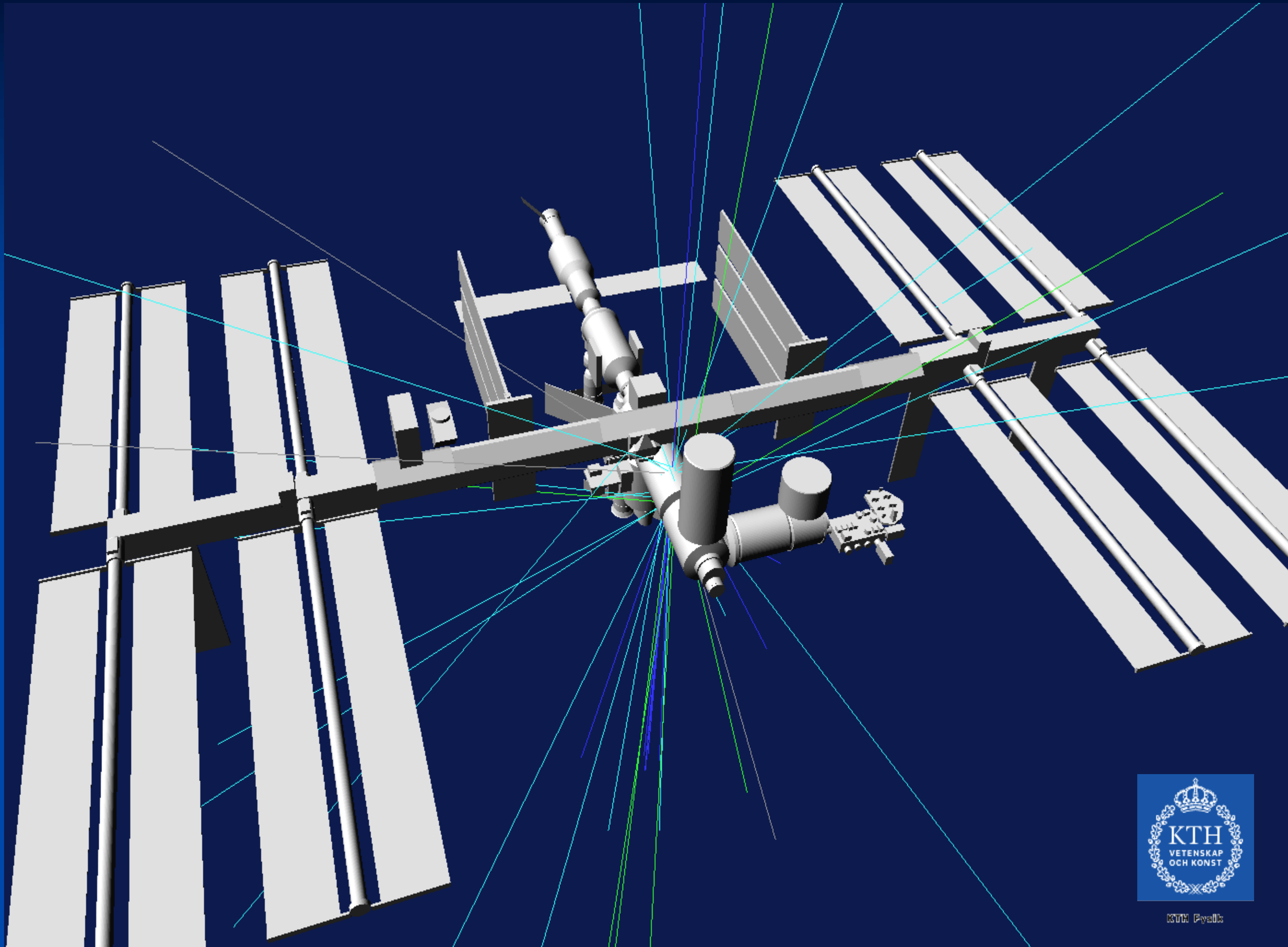
X-ray Multi-Mirror mission (XMM)

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



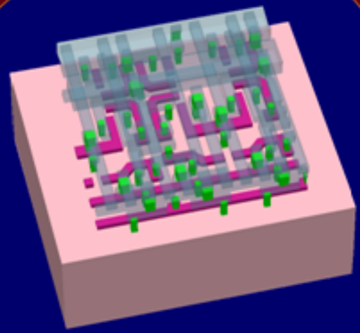
- Chandra X-ray observatory, with similar orbit, experienced unexpected degradation of CCDs
- Possible effects on XMM?


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

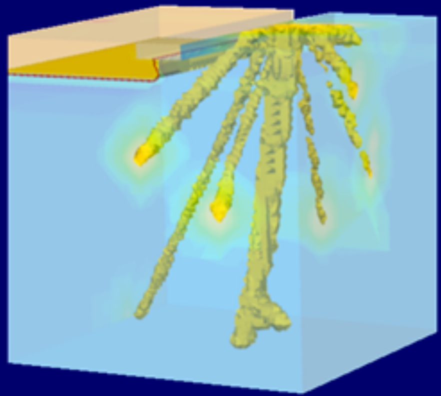


Courtesy T. Ersmark, KTH Stockholm

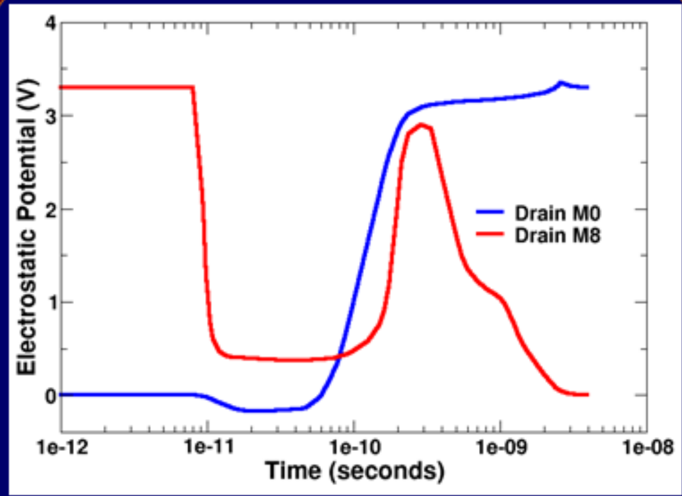
RADSAFE on SEE in SRAMs



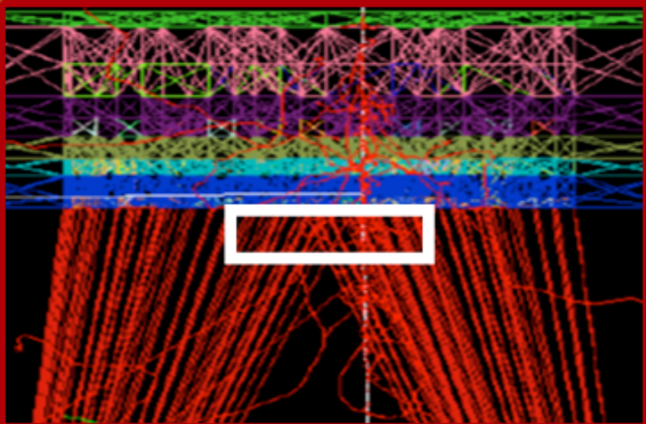
TCAD Cell Structure: SRAM Cell



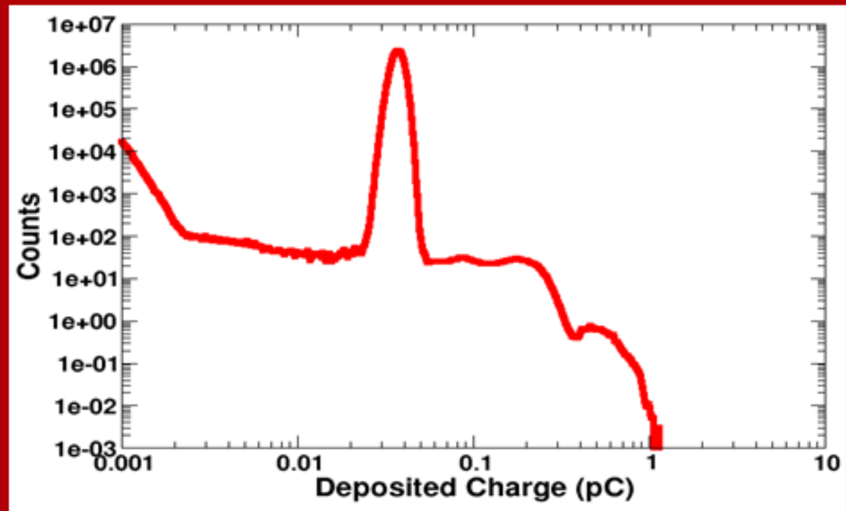
Single Charge Deposition in TCAD: Ne+W Event



SRAM Cell Upset

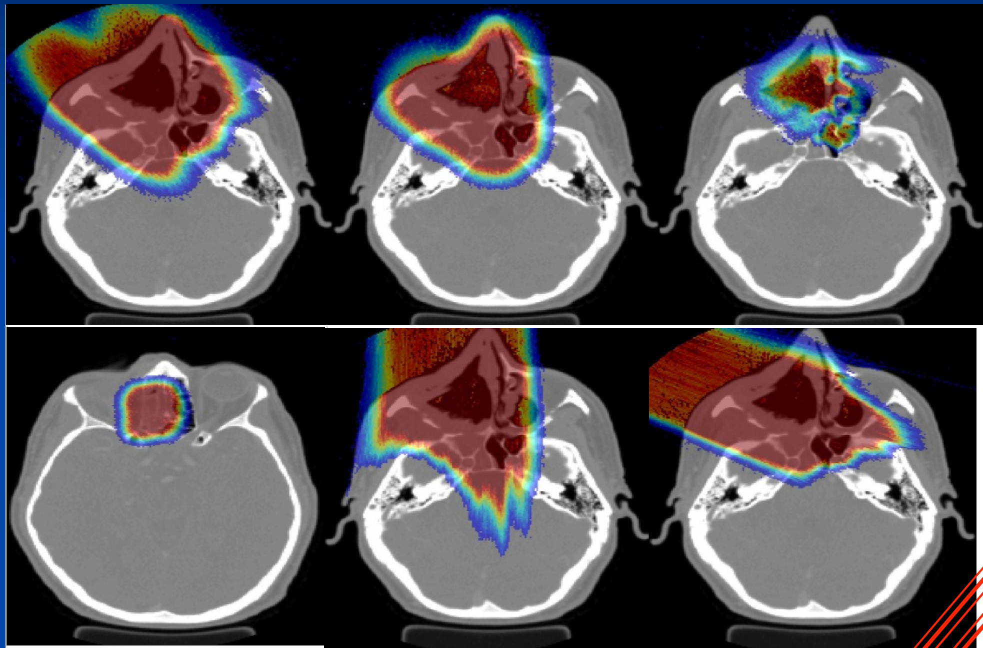


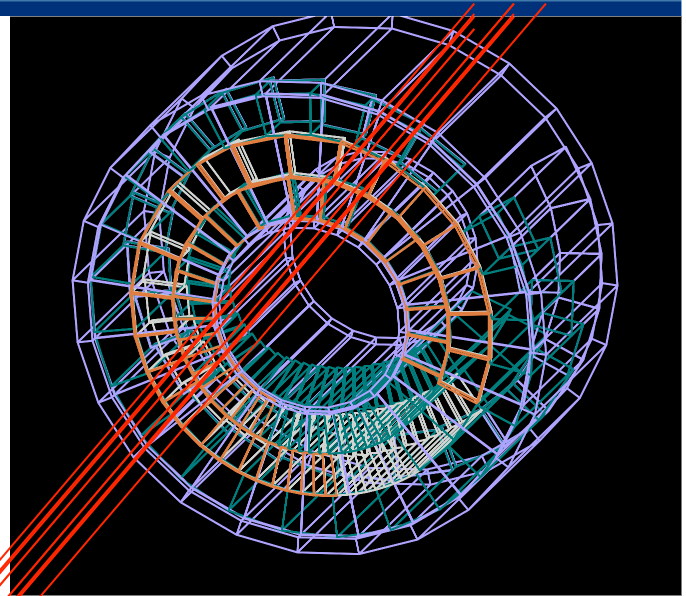
Geant4 Geometry and 523 MeV Neon Event

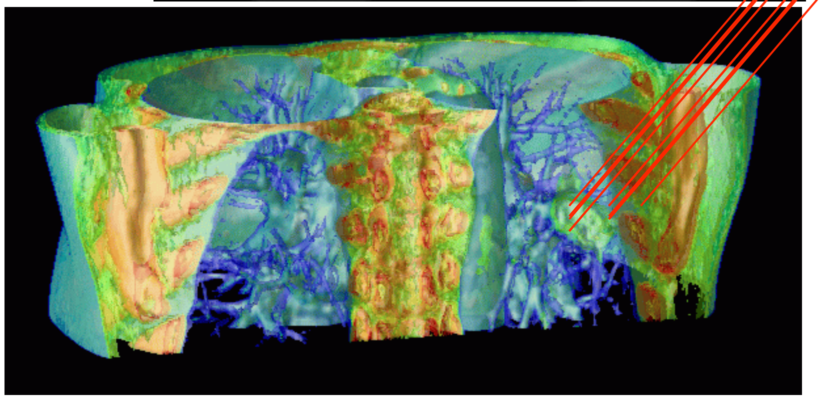


MRED Energy Deposition for 10^8 Events

GEANT4 based proton dose calculation in a clinical environment: technical aspects, strategies and challenges







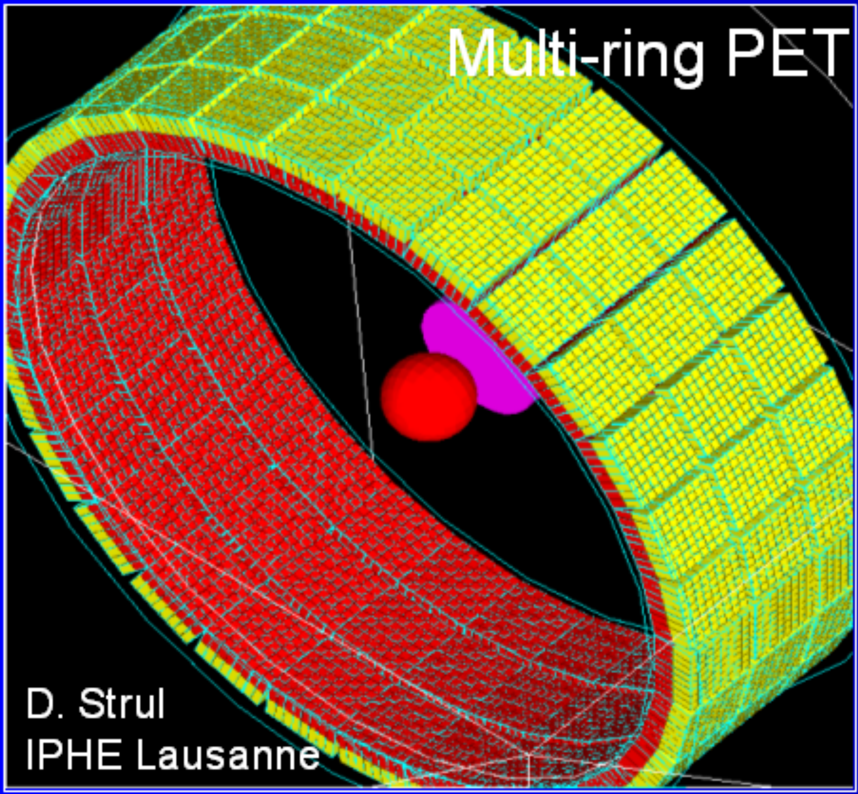
Harald Paganetti

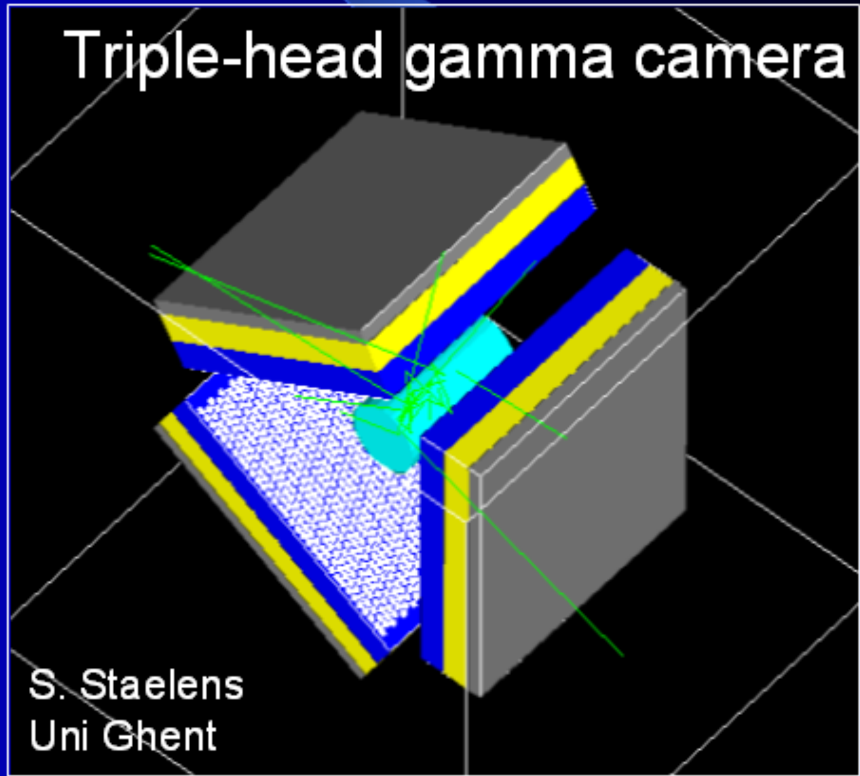
MASSACHUSETTS GENERAL HOSPITAL

HARVARD MEDICAL SCHOOL

March

Geometry examples of GATE applications





Advanced Topics

GATE

Simulations of Preclinical and Clinical Scans in Emission Tomography, Transmission Tomography and Radiation Therapy

Home | Download | Documentation | Collaborative Wiki | Mailing-list | Training | Publications | Meetings | Opportunities | Awards | About GATE

User login

Username *

Password *

Request new password

Log in

PMB Citations Prize

Members of the OpenGATE collaboration have won the Physics in Medicine & Biology Citations Prize twice, in 2009 for their paper 'GATE: a simulation toolkit for PET and SPECT' and in 2015 for their paper 'GATE V6: a major enhancement of the GATE simulation platform enabling modelling of CT and radiotherapy'.

Forewords

GATE is an advanced opensource software developed by the international OpenGATE collaboration and dedicated to numerical simulations in medical imaging and radiotherapy. It currently supports simulations of Emission Tomography (Positron Emission Tomography - PET and Single Photon Emission Computed Tomography - SPECT), Computed Tomography (CT), Optical Imaging (Bioluminescence and Fluorescence) and Radiotherapy experiments. Using an easy-to-learn macro mechanism to configurate simple or highly sophisticated experimental settings, GATE now plays a key role in the design of new medical imaging devices, in the optimization of acquisition protocols and in the development and assessment of image reconstruction algorithms and correction techniques. It can also be used for dose calculation in radiotherapy experiments.

If you are interested in contributing to GATE, here are a few tips regarding what you can do to be part of this collaborative effort:

- Reply to the mailing list
- Contribute to the documentation: ask for a login/password and then modify the documentation on the wiki
- Report bugs

GATE project is now publicly available on GitHub. So, any people identified as a GATE contributor on GitHub can create, assign and close an issue

- Add/modify the source code or fix bugs
  - Start by copying the GATE public repository from GitHub

```
git clone https://github.com/OpenGATE/Gate.git
```

  - Create a specific branch on your repository copy and commit your modifications in that branch
  - Create your own copy (fork) of GATE public repository inside your GitHub account so as to be able to push your branch onto this copy
  - Once your code is ok,

1. Create a pull-request from your Gate repository to the official Gate repository

Shortcuts

Subscribe to GATE-users mailing-list

Request account on GATE collaborative wiki

GitHub

Access to GATE project on GitHub

SurveyMonkey

GATE users survey

Download

http://www.opengatecollaboration.org/

G4 documentation basics

G4 home page

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

GEANT4 A SIMULATION TOOLKIT

Download | User Forum
Contact Us | Bug Reports

Geant4

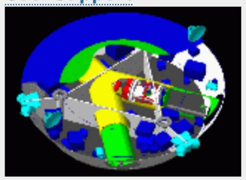
Overview

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

Applications

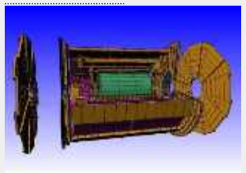
A sampling of applications, technology transfer and other uses of Geant4

User Support



Getting started, guides and information for users and developers

Publications



Validation of Geant4, results from experiments and publications

Collaboration

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

News

2021-03-10
2021 planned developments.

2021-02-05
Patch-01 to **release 10.7** is available from the Download area.

2020-11-06
Patch-03 to release **10.6** is available from the Download archive area.

G4 Application Developer

Book For Application Developers

10.5

Search docs

- Introduction
- Getting Started with Geant4 - Running a Simple Example
- Toolkit Fundamentals
- Detector Definition and Response
- Tracking and Physics
- User Actions
- Control

Docs » Geant4 Book For Application Developers

Geant4 Book For Application Developers

Scope of this manual

The User's Guide for Application Developers is the first manual the reader should consult when learning about GEANT4 or developing a GEANT4 -based detector simulation program. This manual is designed to:

- introduce the first-time user to the GEANT4 object-oriented detector simulation toolkit,
- provide a description of the available tools and how to use them, and
- supply the practical information required to develop and run simulation applications which may be used in real experiments.

This manual is intended to be an overview of the toolkit, rather than an exhaustive treatment of it. Related physics discussions are not included unless required for the description of a particular tool. Detailed discussions of the physics included in GEANT4 can be found in the Physics Reference Manual. Details of the design and functionality of the GEANT4 classes can be found in the User's Guide for Toolkit Developers.

G4 Application Developer

GEANT4
A SIMULATION TOOLKIT

Book For Application Developers

Release 10.5

Geant4 Collaboration

G4 Physics manual

Physics Reference Manual

10.5

Search docs

General Information
Particle Decay
Electromagnetic Interactions
Solid State Physics
Hadronic Physics in GEANT4
Gamma- and Lepto-Nuclear Interactions
Solid State Physics

Docs » Physics Reference Manual

Physics Reference Manual

Scope of this Manual

The Physics Reference Manual provides detailed explanations of the physics implemented in the GEANT4 toolkit.

The manual's purpose is threefold:

- to present the theoretical formulation, model, or parameterization of the physics interactions included in GEANT4,
- to describe the probability of the occurrence of an interaction and the sampling mechanisms required to simulate it, and
- to serve as a reference for toolkit users and developers who wish to consult the underlying physics of an interaction.

This manual does not discuss code implementation or how to use the implemented physics interactions in a simulation. These topics are discussed in the *User's Guide for Application Developers*. Details of the object-oriented design and functionality of the GEANT4 toolkit are given in the *User's Guide for Toolkit Developers*. The *Installation Guide for Setting up |Geant4| in Your Computing Environment* describes how to get the GEANT4 code, install it, and run it.

G4 Physics manual

GEANT4
A SIMULATION TOOLKIT

Physics Reference Manual

Release 10.5

Geant4 Collaboration

Geant4 Cross Reference

GEANT4 A SIMULATION TOOLKIT — Geant4 LXR

Geant4 Cross Reference

Cross-Referencing Geant4
Geant4/

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] [9.2.p2] [9.2.p3] [9.2.p4] [9.3] [9.3.p1] [9.3.p2] [9.4] [9.4.p1] [9.4.p2] [9.4.p3] [9.4.p4] [9.5] [9.5.p1] [9.5.p2] [9.6] [9.6.p1] [9.6.p2] [9.6.p3] [9.6.p4] [10.0] [10.0.p1] [10.0.p2] [10.0.p3] [10.0.p4] [10.1] [10.1.p1] [10.1.p2] [10.1.p3] [10.2] [10.2.p1] [10.2.p2] [10.2.p3] [10.3] [10.3.p1] [10.3.p2] [10.3.p3] [10.4] [10.4.p1] [10.4.p2] [10.4.p3] [10.5] [*10.5.p1*]

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

Name	Size	Last modified (GMT)	Description
analysis/		2019-04-17 07:34:32	
digits_hits/		2019-04-17 07:34:31	
environments/		2019-04-17 07:34:33	
error_propagation/		2019-04-17 07:34:32	
event/		2019-04-17 07:34:32	
examples/		2019-04-17 07:34:31	
externals/		2019-04-17 07:34:32	
g3tog4/		2019-04-17 07:34:32	
geometry/		2019-04-17 07:34:32	
global/		2019-04-17 07:34:33	
graphics_reps/		2019-04-17 07:34:32	

User Forum

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

GEANT4 A SIMULATION TOOLKIT

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	Multithreading	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	DNA/Very Low Energy	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

Laboratory - 1

- Check the G4 documentation
- Find the Manual for Application Developers
- Find the documentation on how to build an example
- Find the link for the Basic examples
- Find the source code for Example Basic B3

Basics of OO programming in C++

Class and Object

Class and Object

Object: is characterized by attributes (which define its state) and operations

A **class** is the blueprint of objects of the same type

```
class Rectangle {
  public:
  Rectangle (double,double); // constructor
  ~Rectangle() { // empty; } // destructor
  double area () { return (width * height); } // member function
  private:
  double width, height; // data members
};
```

```
Rectangle  rectangleA (3.,4.); // instantiate an object of type "Rectangle"
Rectangle* rectangleB = new Rectangle(5.,6.);
cout << "A area: "  <<  rectangleA.area()   << endl;
cout << "B area: "  <<  rectangleB->area() << endl;
delete rectangleB;              // invokes the destructor
```

Inheritance

Inheritance

- A key feature of C++
- Inheritance allows to create classes derived from other classes
- Public inheritance defines an **“is-a”** relationship
  - *In other words: what applies to a base class applies to its derived classes*

```
class Base {
  public:
    virtual ~Base() {}
    virtual void f() {…}
  protected:
    int a;
  private:
    int b; …
};
```

```
class Derived : public Base {
 public:
   virtual ~Derived() {}
   virtual void f() {…}
   …
};
```

Header File

How a Header File looks like

- header file
- begin header guard
- forward declaration
- class declaration
- constructor
- destructor
- member functions
- member variables
- need semi-colon
- end header guard

Segment.h

```
#ifndef SEGMENT_HEADER
#define SEGMENT_HEADER

class Point;
class Segment
{
public:
    Segment();
    virtual ~Segment();
    double length();
private:
    Point* p0,
    Point* p1;
}
#endif // SEGMENT_HEADER
```

Implementation

Header file and implementation

File Segment.hh

```
#ifndef SEGMENT_HEADER
#define SEGMENT_HEADER

class Point;
class Segment
{
public:
    Segment();
    virtual ~Segment();
    double length();
private:
    Point* p0,
    Point* p1;
};
#endif // SEGMENT_HEADER
```

File Segment.cc

```
#include "Segment.hh"
#include "Point.hh"
Segment::Segment() // constructor
{
    p0 = new Point(0.,0.);
    p1 = new Point(1.,1.);
}
Segment::~Segment() // destructor
{
    delete p0;
    delete p1;
}
double Segment::length()
{
    function implementation ...
}
```

OOP programming

OOP basic concepts

- **Object, Class**
  - A class defines the abstract characteristics of a thing (object), including the thing's attributes and the thing's behaviour
- **Inheritance**
  - "Subclasses" are more specialized versions of a class, which ***inherit*** attributes and behaviours from their parent classes (and can introduce their own)
- **Encapsulation**
  - Each object exposes to any class a certain ***interface*** (i.e. those members accessible to that class)
  - Members can be **public**, **protected** or **private**
- **Abstraction**
  - Simplifying complex reality by modelling classes appropriate to the problem
  - One works at the most appropriate level of inheritance for a given aspect of the problem
- **Polymorphism**
  - It allows one to treat derived class members just like their parent class' members

Maria Grazia Pia

Laboratory - 2

- Find at least three classes in the source code of example B3
- Find a few methods of the classes

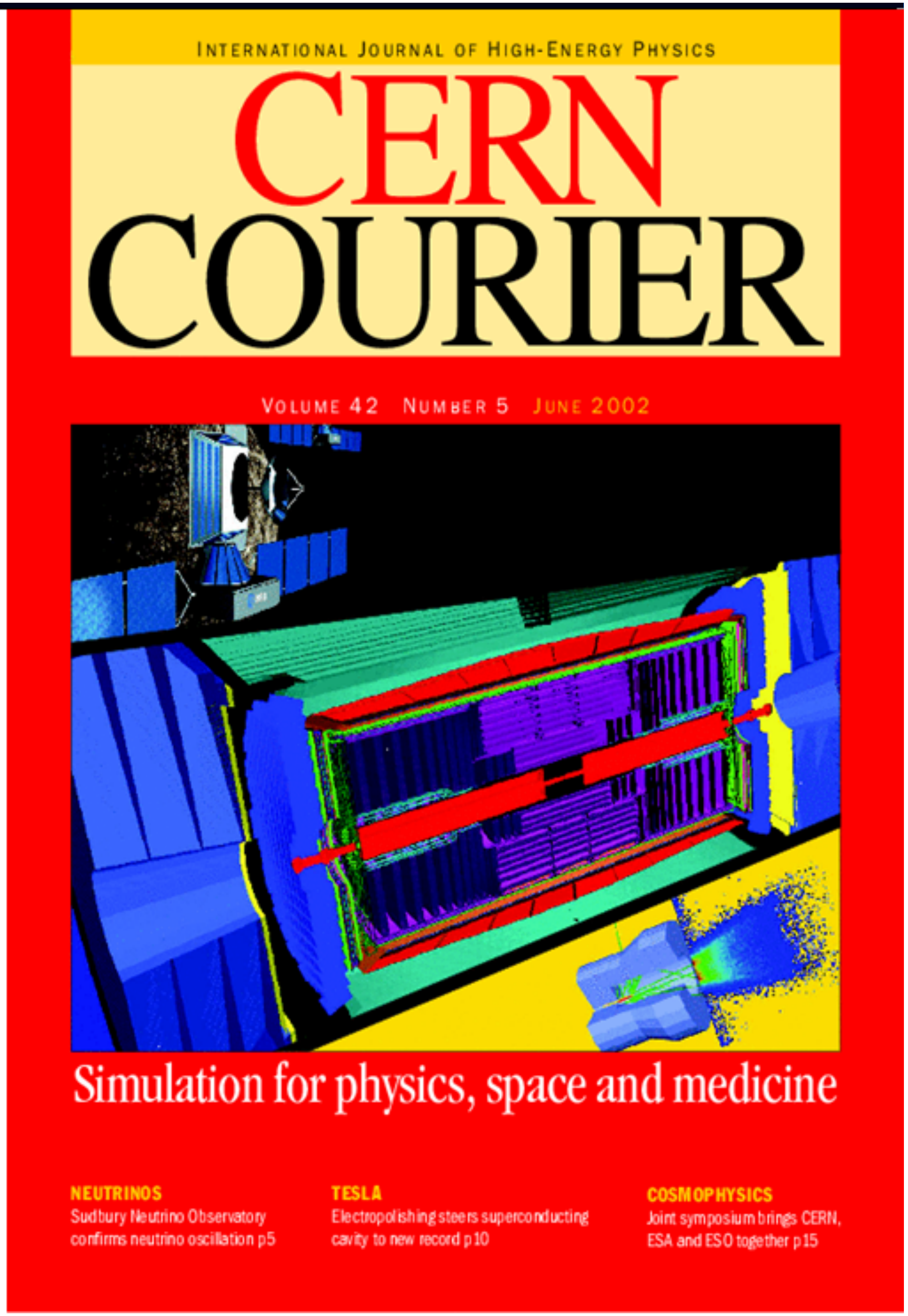
General Introduction to Geant4

Technology transfer

Particle physics software aids space and medicine

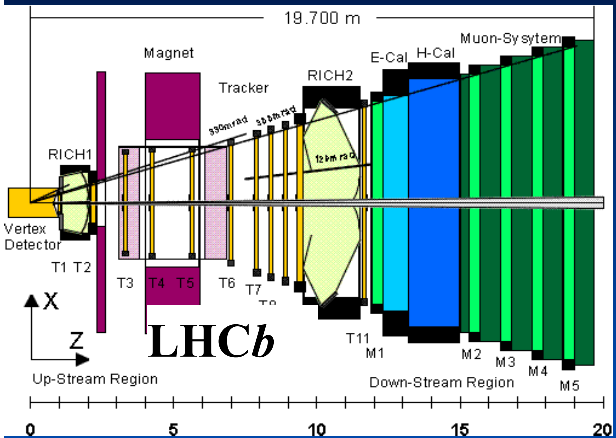
Geant4 is a showcase example of technology transfer from particle physics to other fields such as space and medical science [...].

CERN Courier, June 2002

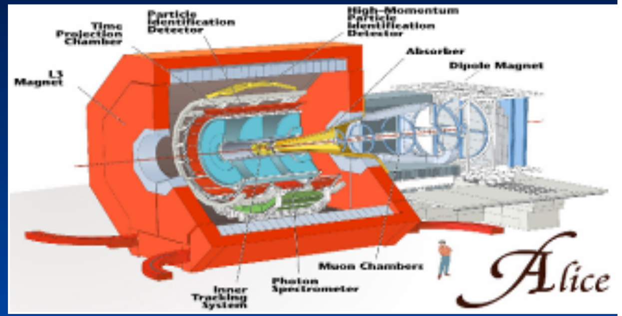


Globalisation

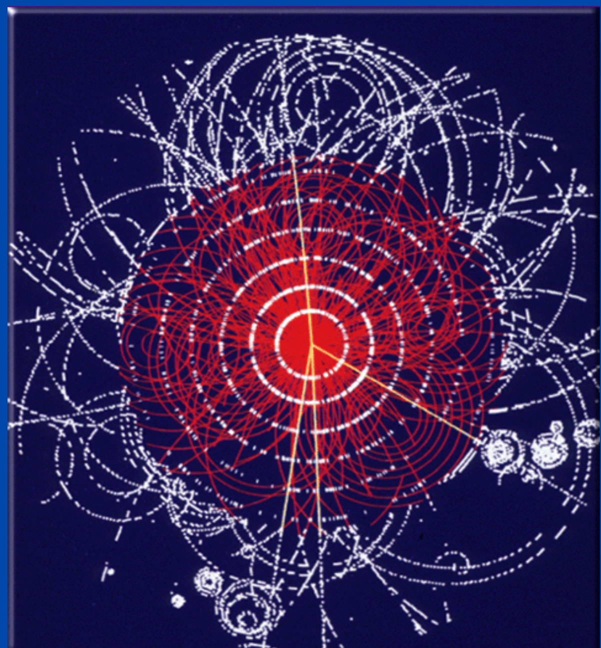
Sharing requirements and functionality across diverse fields

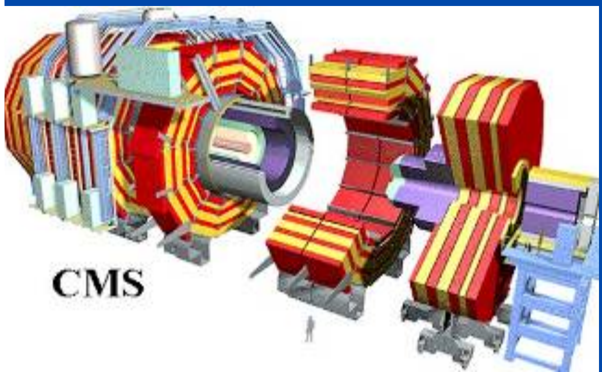


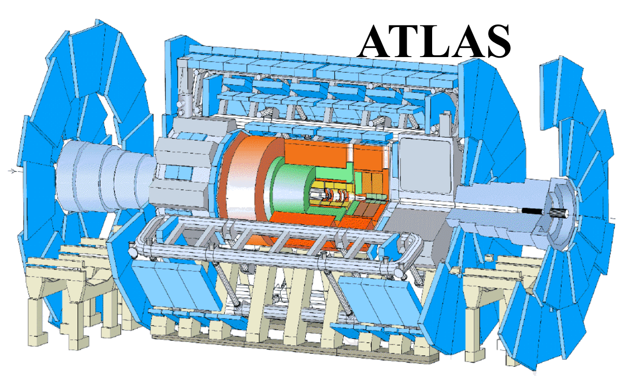
Complex physics
Complex detectors
20 years
software life-span



LHC

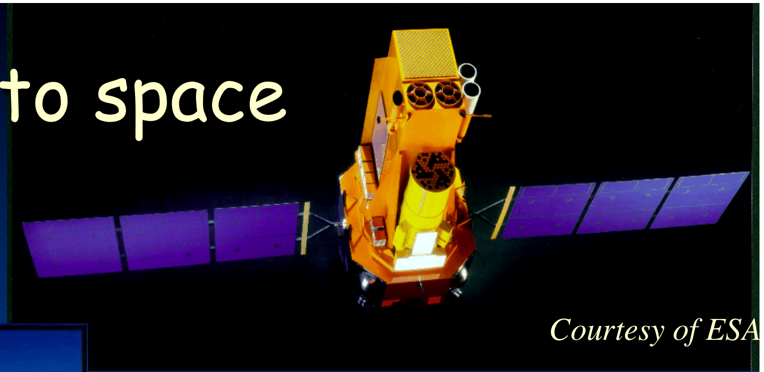






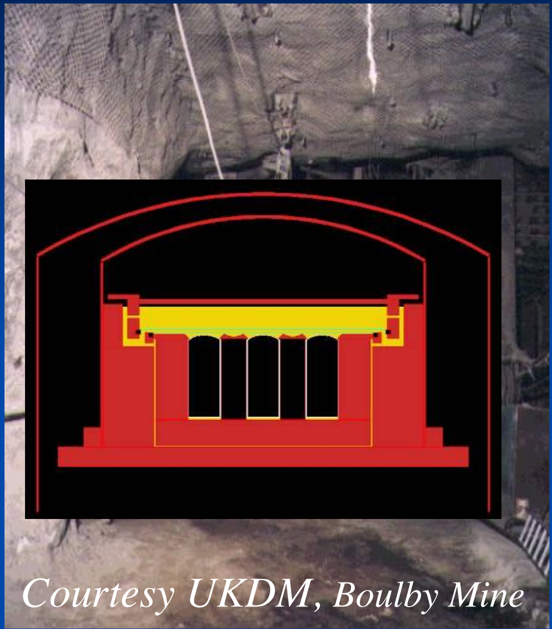
From deep underground…

…to space



Courtesy of ESA

Dark matter and ν experiments



Courtesy UKDM, Boulby Mine

Courtesy of Auger

X and γ astronomy, gravitational waves, radiation damage to components etc.

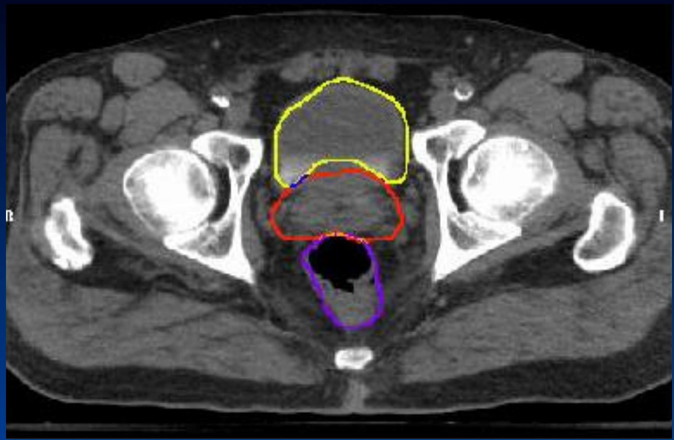
Cosmic ray experiments

Variety of requirements from diverse applications

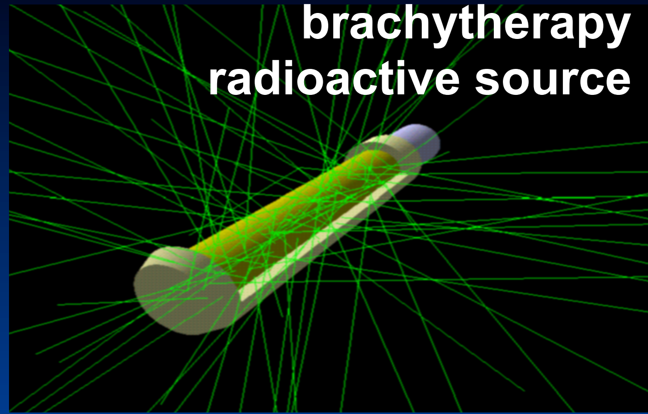
Physics from the eV to the PeV scale	Detectors, spacecrafts and environment

For such experiments software is often **mission critical**

Require reliability, rigorous software engineering standards



Medical Physics

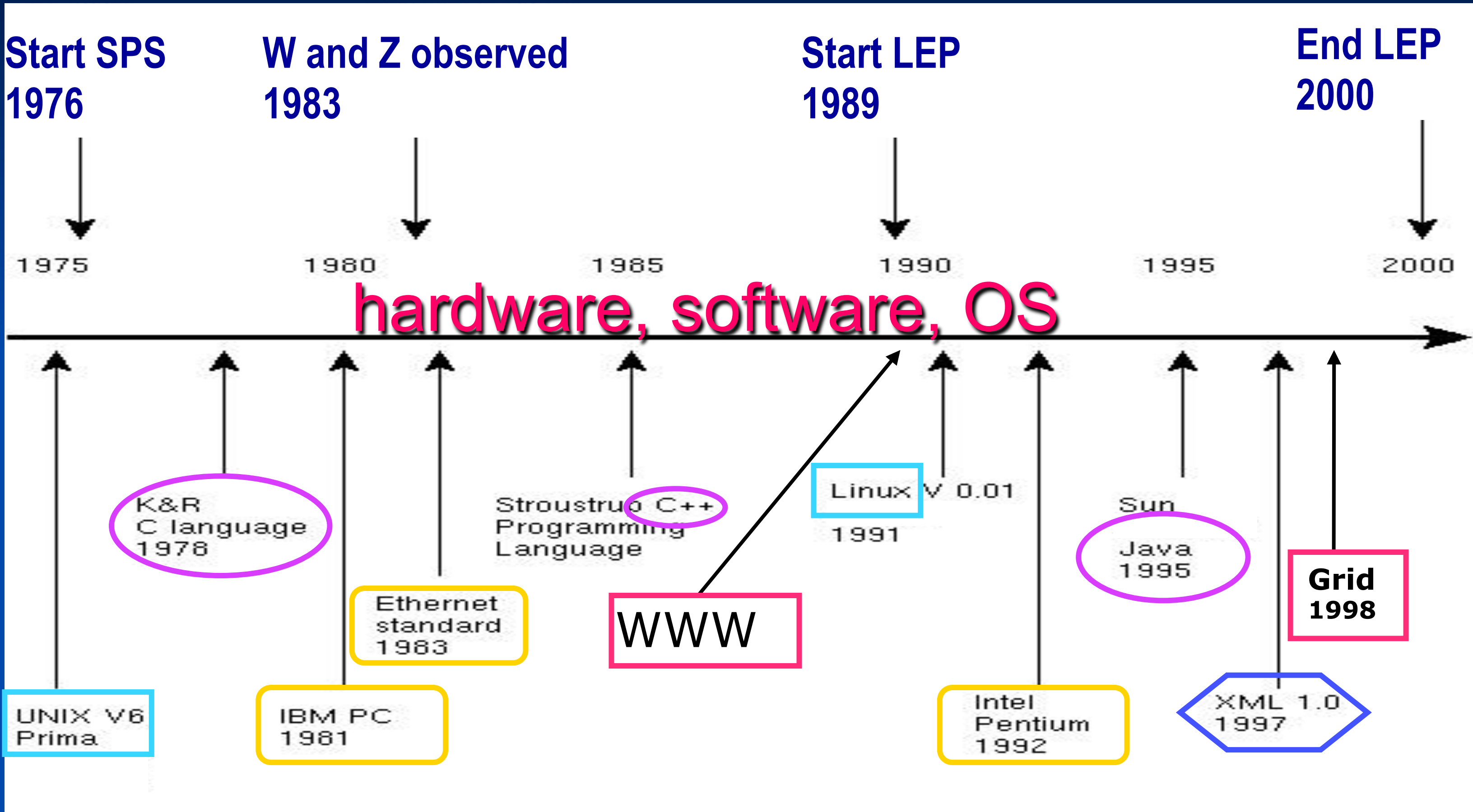


from hospitals…

…to Mars

- Accurate modelling of radiation sources, devices and human body
- Precision of physics
- Reliability
  - Easy configuration and friendly interface
  - Speed

…in a fast changing computing environment



…and don't forget changes of requirements!

Evolution towards greater diversity ➡ we must anticipate changes

Strategic vision

OO technology

- Openness to **extension** and **evolution**

new implementations can be added w/o changing the existing code

- Robustness and ease of **maintenance**

protocols and well defined dependencies minimize coupling

Toolkit

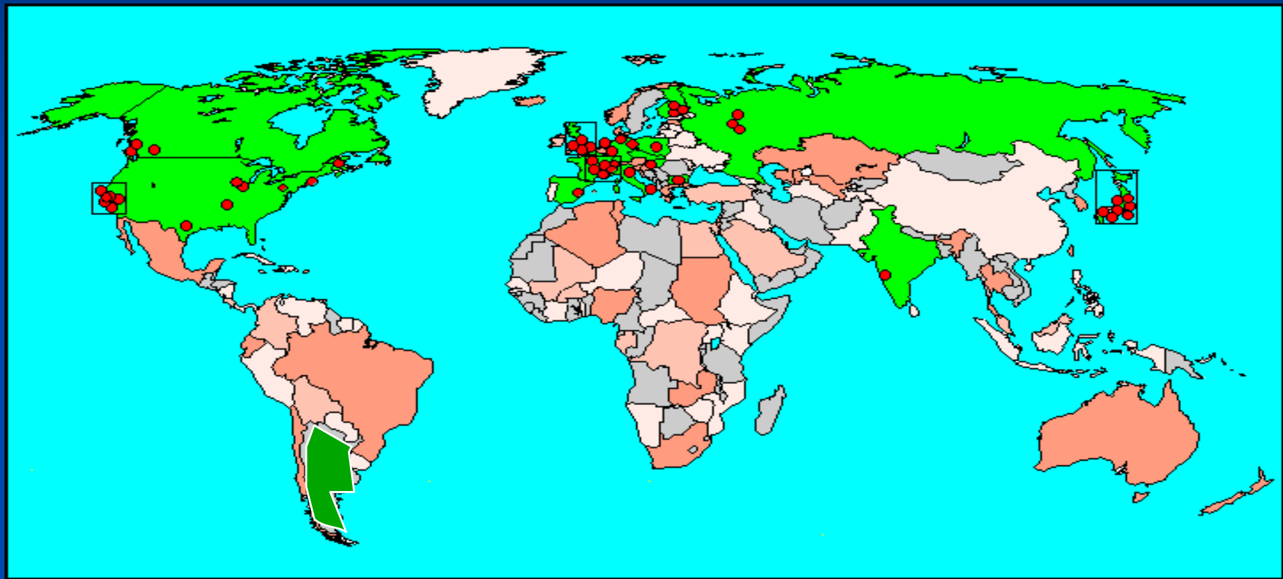
A set of compatible components

- each component is **specialised** for a specific functionality
- each component can be **refined** independently to a great detail
- components can be **integrated** at any degree of complexity
- it is easy to provide (and use) **alternative** components
- the user application can be **customised** as needed

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 Collaboration (~2010)



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)

Geant4 Collaboration (now)

Home > **Collaboration**

Collaboration

Since 1999 the production service, user support and development of Geant4 have been managed by the international Geant4 Collaboration. The collaboration is based on a Collaboration Agreement among the participating laboratories, experiments and national institutes. Many specialized working groups are responsible for the various domains of the toolkit.

Collaboration Members

List of Collaboration members with contact information

Learn More

Steering and Oversight Boards

Steering board and Oversight board

Learn More

Technical Forum

Role and scope

Learn More

https://geant4.web.cern.ch/collaboration/

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
- May 2023 – G4 v11.1p1 ← Current version
- We currently provide one public release every year (and several patches) + one beta public release (soon G4 v10.8.beta).

Geant4 source version

GEANT4 A SIMULATION TOOLKIT — About | Download | Documentation | User Forum

Home > Download > **Download Geant4-11.1.1**

Download Geant4-11.1.1

First released 13 Feb 2023 **All releases**

License

See the license conditions.

RELEASE NOTES

See:
Main Release Notes - Patch-1 -

Source code

Source code is freely available from CERN GitLab or through GitHub.
Source code can also be browsed through the LXR source code browser.

Download zip | **Download tar.gz** | **Download tar.bz2** | **Download tar**

https://geant4.web.cern.ch/download/11.1.1.html

Geant4 official publications

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

GEANT4—a simulation toolkit

S. Agostinelli [ae], J. Allison [as], K. Amako [e], J. Apostolakis [a], H. Araujo [aj], P. Arce [l, m, x, a], M. Asai [g, ai], D. Axen [i, t], S. Banerjee [bi, l], G. Barrand [an], F. Behner [l], L. Bellagamba [c], J. Boudreau [bd], L. Broglia [ar], A. Brunengo [c], H. Burkhardt [a], S. Chauvie [bj, bl], J. Chuma [h] … D. Zschiesche [af]

⊞ **Show more**

https://doi.org/10.1016/S0168-9002(03)01368-8

Get rights and content

Geant4 official publications

Geant4 developments and applications

Publisher: IEEE

Cite This

PDF

73 Author(s) J. Allison ; K. Amako ; J. Apostolakis ; H. Araujo ; P. Arce Dubois ; M. Asai ; G. Barrand ; R. Capra ; S. Ch... View All Authors

Abstract

Document Sections

Abstract:

Geant4 is a software toolkit for the simulation of the passage of particles through matter. It is used by a large number of experiments and projects in a variety of application domains, including high energy physics, astrophysics and space science, medical physics and radiation protection. Its functionality and modeling capabilities continue to be extended, while its performance is enhanced. An overview of recent developments in diverse areas of the toolkit is presented. These include performance optimization for complex setups; improvements for the propagation in fields; new options for event biasing; and additions and improvements in geometry, physics processes and interactive capabilities

Published in: IEEE Transactions on Nuclear Science (Volume: 53 , Issue: 1 , Feb. 2006)

Page(s): 270 - 278

INSPEC Accession Number: 8999460

Date of Publication: 27 March 2006

DOI: 10.1109/TNS.2006.869826

▸ **ISSN Information:**

Publisher: IEEE

Geant4 official publications

ELSEVIER

Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment

Volume 835, 1 November 2016, Pages 186-225

Recent developments in GEANT4

J. Allison [a, b], K. Amako [c, a], J. Apostolakis [d], P. Arce [e], M. Asai [f], T. Aso [g], E. Bagli [h], A. Bagulya [i], S. Banerjee [j], G. Barrand [k], B.R. Beck [l], A.G. Bogdanov [m], D. Brandt [n], J.M.C. Brown [o], H. Burkhardt [d], Ph. Canal [j], D. Cano-Ott [p], S. Chauvie [q] … H. Yoshida [bs, a]

⊞ Show more

https://doi.org/10.1016/j.nima.2016.06.125

Get rights and content

Simulation basics

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*

Geant4 simulation toolkit

- 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

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 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.

Physics

From the Minutes of LCB (LHCC Computing Board) meeting on 21 October, 1997:

"It was noted that experiments have requirements for **independent, alternative physics models**. In Geant4 these models, differently from the concept of packages, allow the user to **understand** how the results are produced, and hence improve the **physics validation**. Geant4 is developed with a modular architecture and is the ideal framework where existing components are integrated and new models continue to be developed."

What is a Physics List?

- A class which collects all the particles, physics processes and production thresholds needed for your application
- It tells the run manager how and when to invoke physics

- It is a very flexible way to build a physics environment
  - user can pick the particles he wants
  - user can pick the physics to assign to each particle

- But, user must have a good understanding of the physics required
  - omission of particles or physics could cause errors or poor simulation

Why Do We Need a Physics List?

- Physics is physics – shouldn't Geant4 provide, as a default, a complete set of physics that everyone can use?
- No:
  - there are many different physics models and approximations
    - very much the case for hadronic physics
    - but also the case for electromagnetic physics
  - computation speed is an issue
    - a user may want a less-detailed, but faster approximation
  - no application requires all the physics and particles Geant4 has to offer
    - e.g., most medical applications do not want multi-GeV physics

Why Do We Need a Physics List?

- For this reason Geant4 takes an atomistic, rather than an integral approach to physics
  - provide many physics components (processes) which are de-coupled from one another
  - user selects these components in custom-designed physics lists in much the same way as a detector geometry is built

Physics Processes Provided by Geant4

- EM physics
  - "standard" processes valid from ~ 1 keV to ~ PeV
  - "low-energy" Livermore/ Penelope valid from 250 eV to ~ PeV
  - optical photons
- Weak physics
  - decay of subatomic particles
  - radioactive decay of nuclei
- Hadronic physics
  - pure hadronic processes valid from 0 to ~100 TeV
  - γ−, μ−nuclear valid from 10 MeV to ~TeV
- Parameterized or "fast simulation" physics

Pre-packaged Physics Lists (1)

- Our example deals mainly with electromagnetic physics
- A complete and realistic set of EM physics lists are provided
  - add to it according to your needs
- Adding hadronic physics is more involved
  - for any one hadronic process, user may choose from several hadronic models to choose from
  - choosing the right models for your application requires care
  - to make things easier, hadronic physics lists are now provided according to some use cases

Pre-packaged Physics Lists (2)

- Originally referred to as "hadronic physics lists" but include electromagnetic physics already
- Can be found on the Geant4 web page at
  - PhysicsList Guide
- Caveats:
  - these lists are provided as a "best guess" of the physics needed in a given case
  - The user is responsible for validating the physics for his own application and adding (or subtracting) the appropriate physics
    - "Trust, but verify."
  - they are intended as starting points or templates

Reference Physics Lists

- Reference physics lists attempt to cover a wide range of use cases
  - Extensive validation by LHC experiments for simulation hadronic showers
  - Comparison experinents for neutron production and transport demonstrates good agreement
    - QGSP_BIC_HP, QGSP_BERT_HP
  - user feedback, e.g. vi hypernews, is welcome
- Users responsible for validating results

- Documentation available from G4 Physics List manual
- Physics Lists User forum for questions and feedback

Laboratory - 3

- Find the information about Physics processes in G4
- Find the documentation about "Physics Lists"

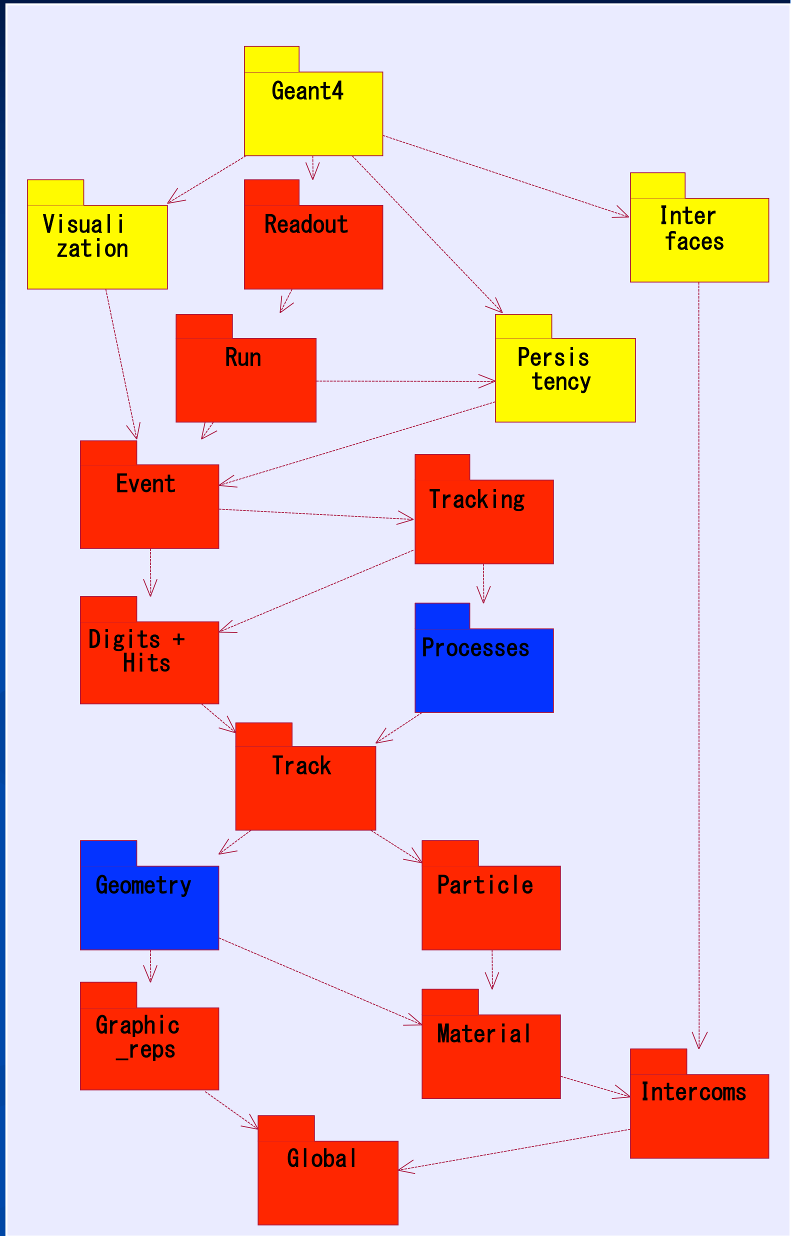
Part 2

Main ingredients of a G4 application

Basic concepts and kernel structure

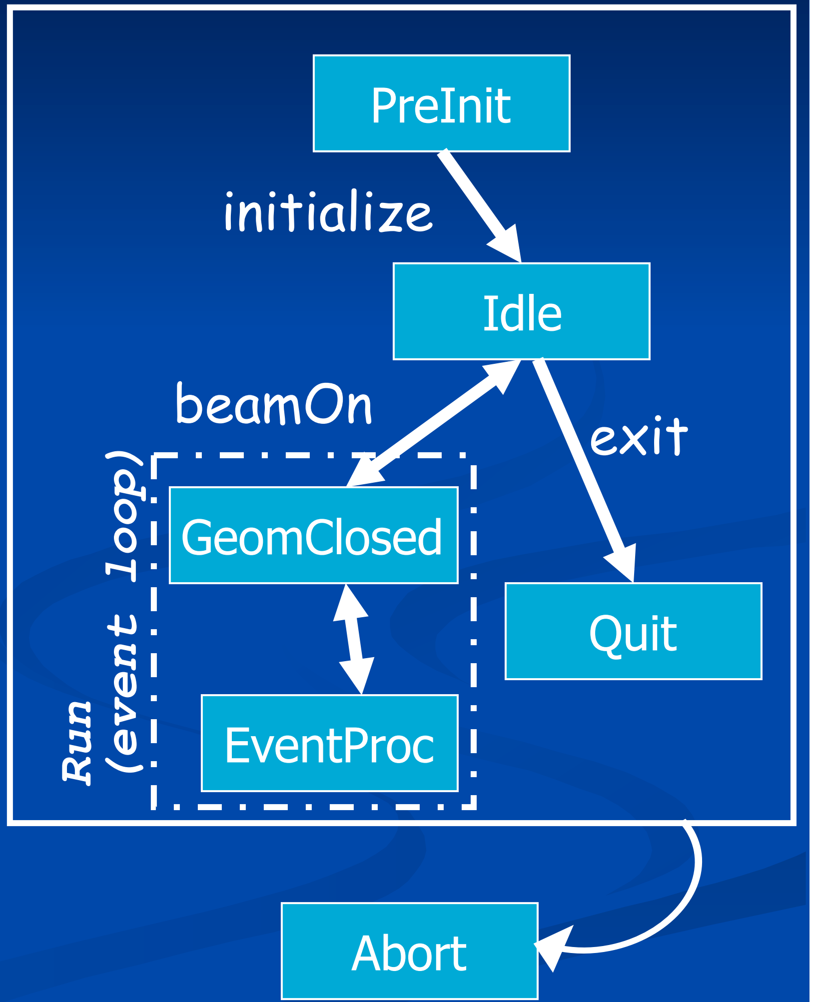
Geant4 kernel

- 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.



Geant4 as a state machine

- Geant4 has six application states.
  - G4State_PreInit
    - Material, Geometry, Particle and/or Physics Process need to be initialized/defined
  - G4State_Idle
    - Ready to start a run
  - G4State_GeomClosed
    - Geometry is optimized and ready to process an event
  - G4State_EventProc
    - An event is processing
  - G4State_Quit
    - (Normal) termination
  - G4State_Abort
    - A fatal exception occurred and program is aborting



The main program

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.

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()*.

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.

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
- In the virtual method *ConstructSDandField*(),
  - 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
  - G4GeneralParticleSource

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 Geant4 native scorers to get specified quantities (dose, energy release, flux, path length, etc.)

User Classes needs

- Define material and geometry
  - ➔ G4VUserDetectorConstruction
- Select appropriate particles and processes and define production threshold(s)
  - ➔ G4VUserPhysicsList
- Define the way of primary particle generation
  - ➔ G4VUserPrimaryGeneratorAction
- Define the way to extract useful information from Geant4
  - ➔ G4UserSteppingAction, G4UserTrackingAction, etc.
  - ➔ G4VUserDetectorConstruction, G4UserEventAction, G4Run, G4UserRunAction
  - ➔ G4SensitiveDetector, G4VHit, G4VHitsCollection
  - ➔ G4PrimitiveScorers

The kernel

Run and event

- Multiple events
  - possibility to handle the pile-up
- Multiple runs in the same job
  - with different geometries, materials etc.
- Powerful stacking mechanism
  - three levels by default: handle trigger studies, loopers etc.

Tracking

- Decoupled from physics
  - all processes handled through the same abstract interface
- Independent from particle type
- New physics processes can be added to the toolkit without affecting tracking

Geant4 has only production thresholds, **no tracking cuts**

- all particles are tracked down to zero range
- energy, TOF ... cuts can be defined by the user

Run in Geant4

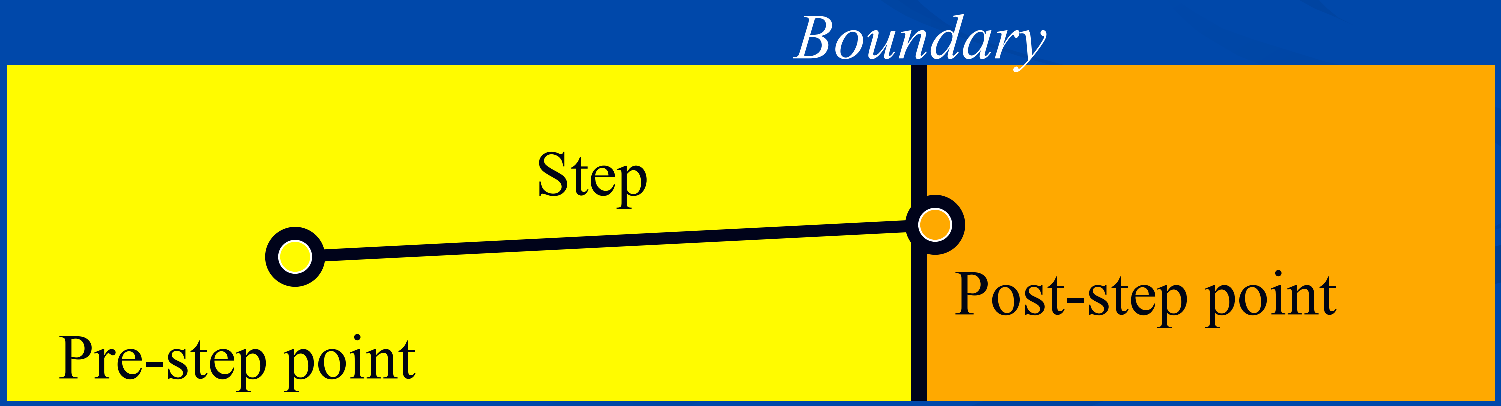
- As an analogy of the real experiment, a run of Geant4 starts with "Beam On".
- Within a run, the user cannot change
  - detector setup
  - settings of physics processes
- Conceptually, a run is a collection of events which share the same detector and physics conditions.
  - A run consists of one event 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.

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.



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.

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.

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.

User Interface in G4

Geant4 UI command

- A command consists of
  - Command directory
  - Command
  - Parameter(s)

```
/run/verbose 1
/vis/viewer/flush
```

- A parameter can be a type of string, boolean, integer or double.
  - Space is a delimiter.
  - Use double-quotes (“”) for string with space(s).
- A parameter may be “omittable”. If it is the case, a default value will be taken if you omit the parameter.
  - Default value is either predefined default value or current value according to its definition.
  - If you want to use the default value for your first parameter while you want to set your second parameter, use “!” as a place holder.

```
/dir/command ! second
```

Command submission

- Geant4 UI command can be issued by
  - (G)UI interactive command submission
  - Macro file
  - Hard-coded implementation
    - Slow but no need for the targeting class pointer
    - Should not be used inside an event loop

```
G4UImanager* UI = G4UImanager::GetUIpointer();
UI->ApplyCommand("/run/verbose 1");
```

- The availability of individual command, the ranges of parameters, the available candidates on individual command parameter may vary according to the implementation of your application and may even vary dynamically during the execution of your job.
- some commands are available only for limited Geant4 application state(s).
  - E.g. **`/run/beamOn`** is available only for *Idle* states.

Macro file

- Macro file is an ASCII file contains UI commands.
- All commands must be given with their full-path directories.
- Use "#" for comment line.
  - First "#" to the end of the line will be ignored.
  - Comment lines will be echoed if **`/control/verbose`** is set to 2.
- Macro file can be executed
  - interactively or in (other) macro file

    ```
    /control/execute file_name
    ```

  - hard-coded

    ```
    G4UImanager* UI = G4UImanager::GetUIpointer();
    UI->ApplyCommand("/control/execute file_name");
    ```

Laboratory - 4

- Find the macro files for example B3
- Check which are the categories the commands is refereeing to …

G4 Virtual Machine

LP2i Bordeaux

GEANT4 VIRTUAL MACHINE

G4VM with RockyLinux 8.5 Geant4.11.1.1

Presentation News What is included Tutorial Hints Publications Download Contacts

Since 2004, LP2i Bordeaux, CNRS / IN2P3 / Bordeaux University laboratory provides free of charge and licensing to Geant4 users a **Geant4 Virtual Machine** with several visualisation, analysis and development tools.

Mode sombre : Off

https://extra.lp2ib.in2p3.fr/G4/

Laboratory - 5

- Check the installation of G4 through the Virtual Machine