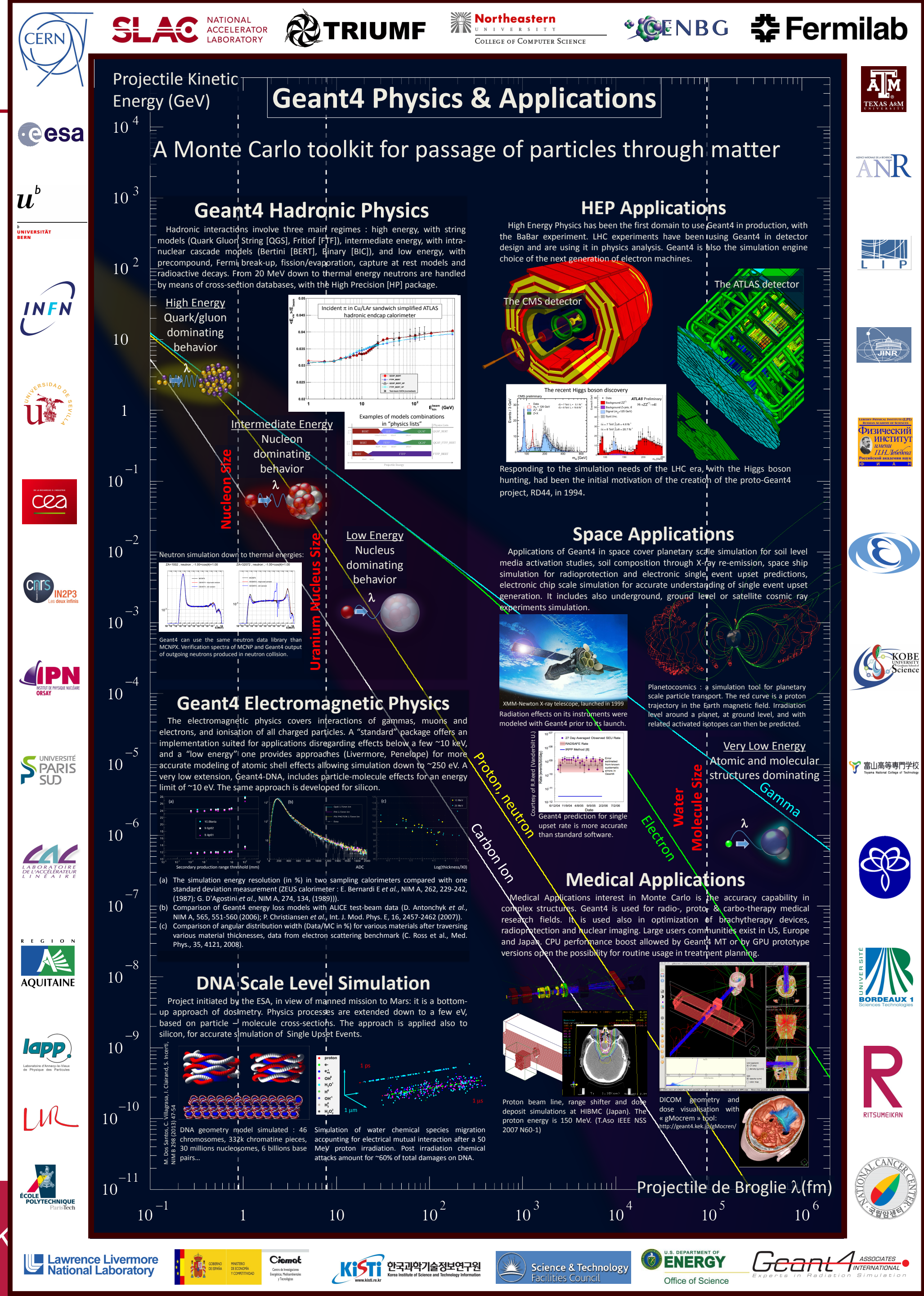
Version 10.4-p02

# Kernel I

Makoto Asai (SLAC)
Geant4 Tutorial Course

# Contents

- General introduction and brief history
- What's new in version 10
- Highlights of user applications
- Geant4 license
- Geant4 kernel
  - Basic concepts and kernel structure
  - User classes



GEANT4

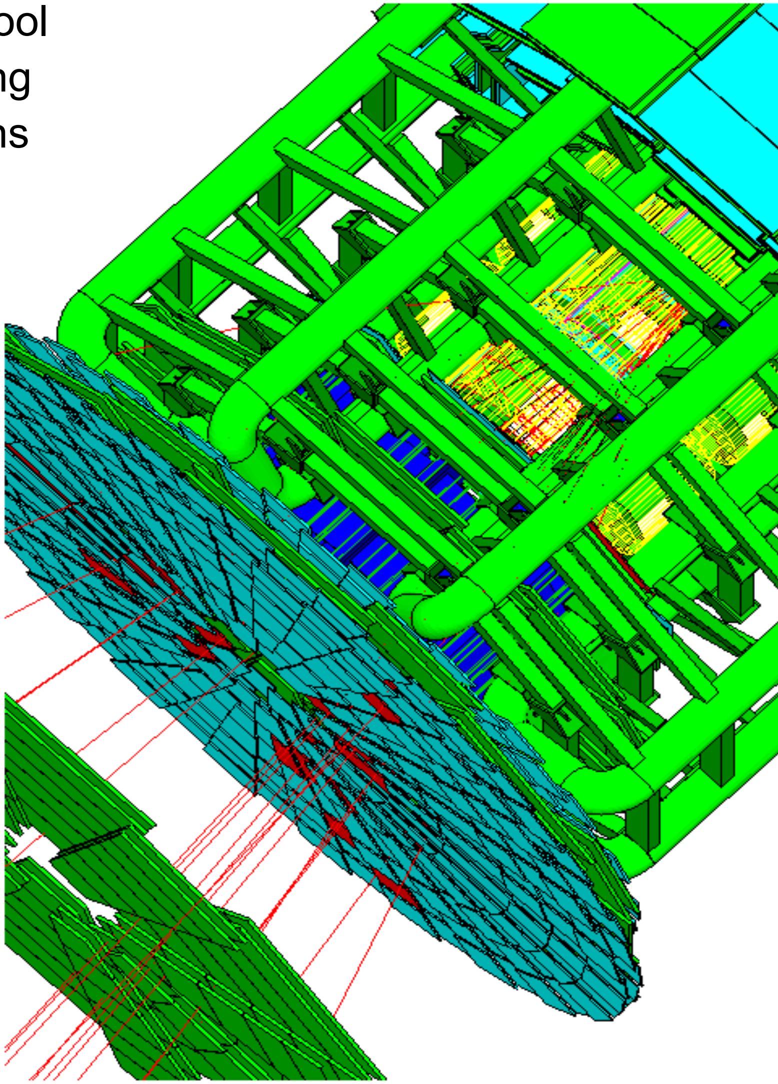
A SIMULATION TOOLKIT

Version 10.4-p02

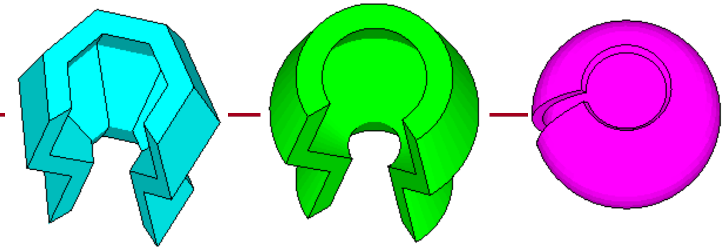
# General introduction and brief history

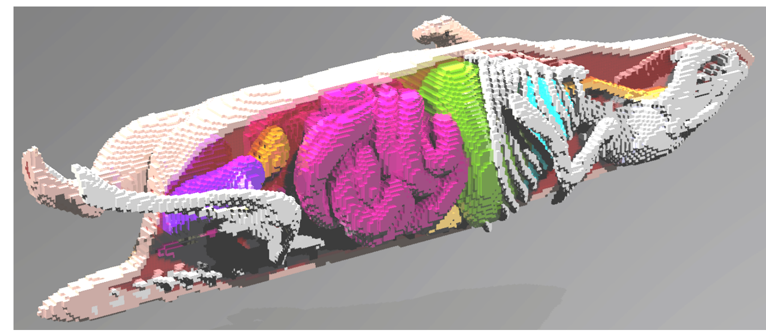
# Key Geant4 functionalities

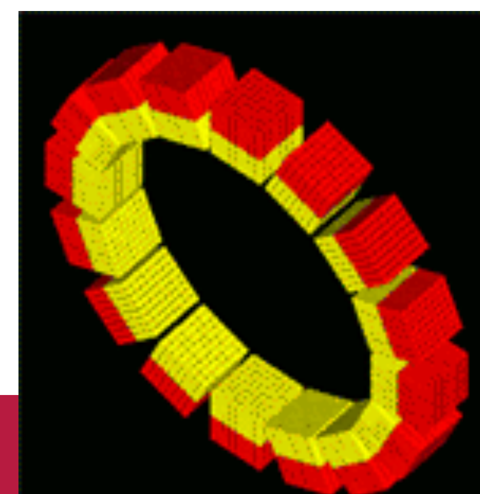
- Geant4 is a general purpose Monte Carlo simulation tool for elementary particles passing through and interacting with matter. It finds quite a wide variety of user domains including high energy and nuclear physics, space engineering, medical applications, material science, radiation protection and security.
- Geant4 offers most, if not all, of the functionalities required for the simulation of elementary particle and nucleus passing through and interacting with matter.
  - Kernel
  - Geometry and navigation
  - Physics processes
  - Scoring
  - GUI and Visualization drivers
- Thanks to the polymorphism mechanism of C++, the users can easily plug-in their extensions without interfering with the other part of Geant4.
- Extensive user guide documents and examples are provided.



# Key geometry capabilities



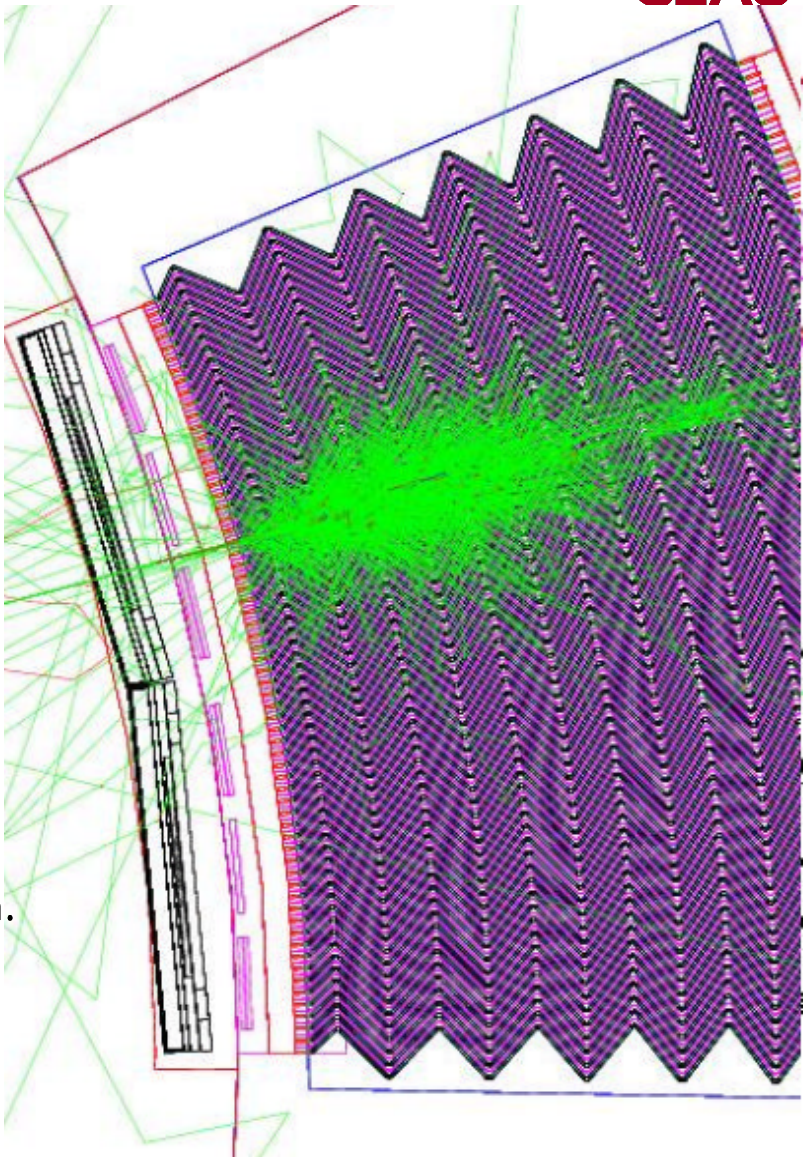




- Richest collection of shapes
  - CSG (Constructed Solid Geometry), Boolean operation, Tessellated solid, etc.
  - The user can easily extend
- Describing a setup as hierarchy or ‘flat’ structure
  - Describing setups up to billions of volumes
  - Tools for creating & checking complex structures
  - Interface to CAD
- Navigating fast in complex geometry model
  - Automatic optimization
- Geometry models can be ‘dynamic’
  - Changing the setup at run-time, e.g. “moving objects”

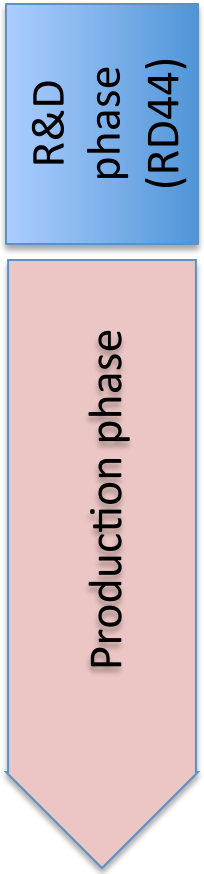
# Physics models in Geant4

- Geant4 offers
  - Electromagnetic processes
  - Hadronic and nuclear processes
  - Photon/lepton-hadron processes
  - Optical photon processes
  - Decay processes
  - Shower parameterization
  - Event biasing techniques
  - And you can plug-in more
- Geant4 provides sets of alternative physics models so that the user can freely choose appropriate models according to the type of his/her application.
  - For example, some models are more accurate than others at a sacrifice of speed.



# Geant4 History

- Early discussions, for example at CHEP 1994 @ San Francisco
  - "Geant steps into the future" R. Brun et al.
  - "Object oriented analysis and design of a GEANT based detector simulator" K. Amako et al.

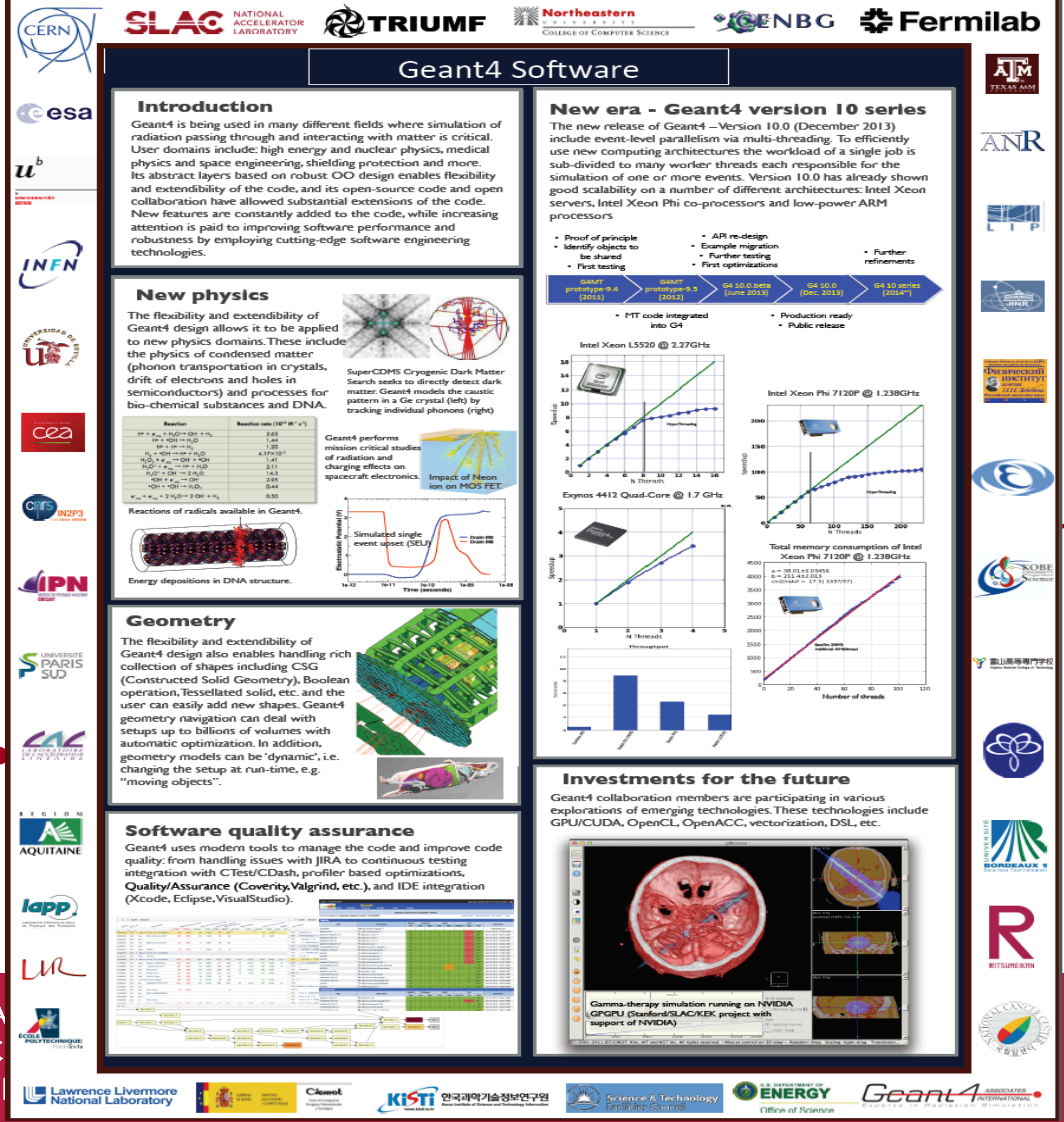



- Dec '94 – R&D project start
- Apr '97 - First alpha release
- Jul '98 - First beta release
- Dec '98 - First Geant4 public release - version 1.0

- Several major architectural revisions
  - E.g. STL migration, "cuts per region", parallel worlds, **multithreading**

- Dec 8th, '17 – Geant4 version 10.4 release
  - May 25th, '18 - Geant4 10.4-patch02 release ← *Current version*
- We currently provide one public release every year.

# What's new in Geant4 version 10

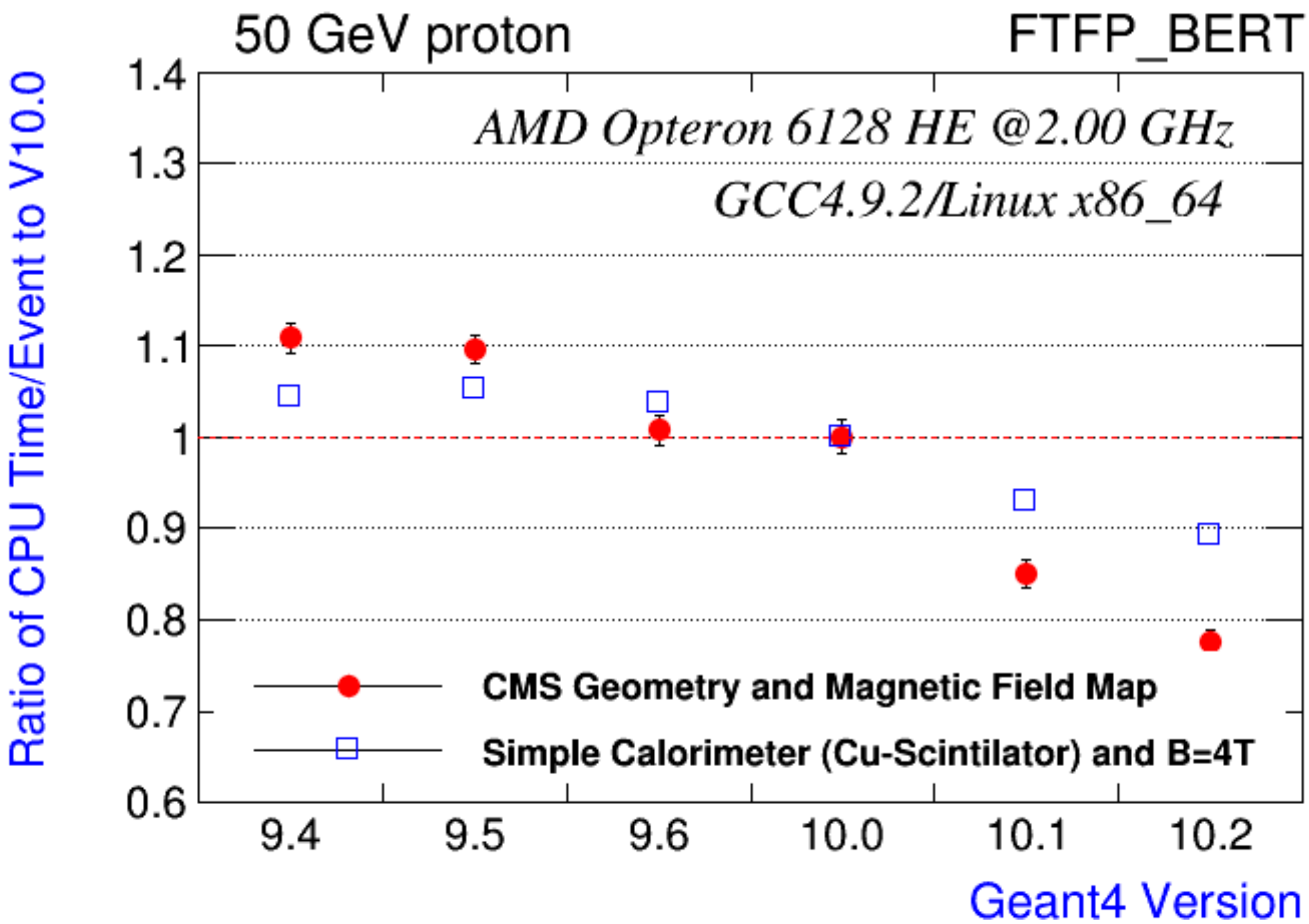


# Geant4 version 10 series

- The release in 2013 was a major release.
  - Geant4 version 10 – release date : Dec. 6, 2013
- The highlight is its multi-threading capability.
  - The world first large-scale physics software fully multithreaded
- Geant4 version 10 series will be evolving.
  - Performance improvements (both in physics and computing)
  - Missing functionalities yet to be migrated to multithreading,
  - Additional APIs
  - Additional functionalities
  - New physics

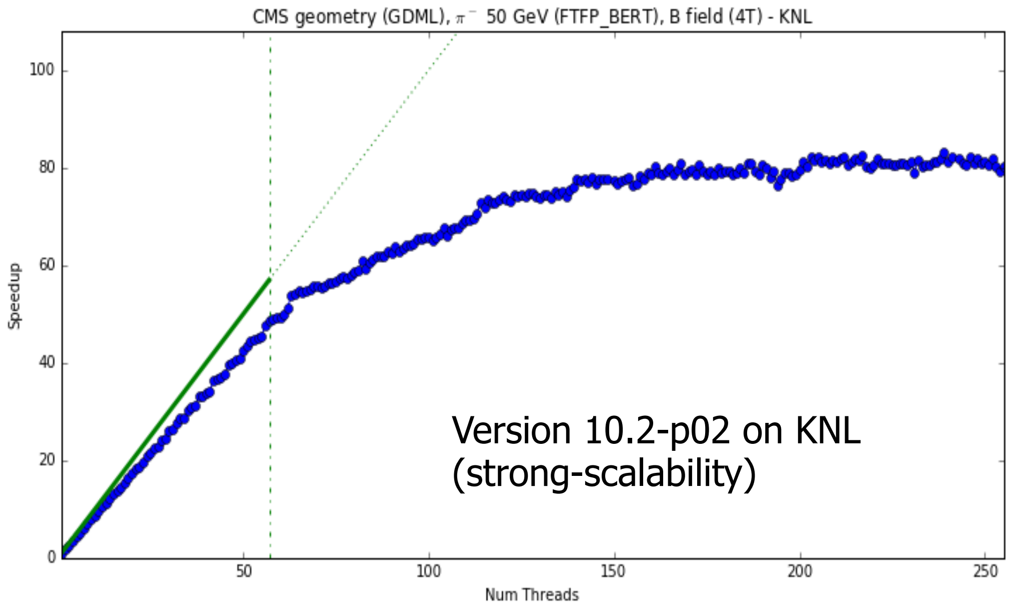
| G4MT prototype-9.4 (2011) | G4MT prototype-9.5 (2012) | G4 10.0.beta (Jun.2013) | G4 10.0 (Dec. 2013) | G4 10 series (2014~) |
| --- | --- | --- | --- | --- |
| • Proof of principle<br>• Identify objects to be shared<br>• First testing | • MT code integrated into G4 | • API re-design<br>• Example migration<br>• Further testing<br>• First optimizations | • Production ready<br>• Public release | • Further refinements |

# We are making it faster!




*ATLAS : "The 10% CPU improvement we gain from the move from G4 9.6 to 10.1 is invaluable to the collaboration."*

# Many core, many thread




- For three years we have provided support for running Geant4 on KNC.
  - ATLAS, CMS successfully multithreaded
- We will soon extend our support to KNL.
  - With KNL, thanks to x86 binary compatibility including the use of gcc, work-flow is tremendously simplified.

| System | Time to completion (5k events) |
|---|---|
| Xeon E5-2620 @ 2.1 GHz<br>(12 cores, 24 threads) | 570 s |
| KNC (31s1P) @ 1.0 GHz (228 threads) | 1000 s |
| KNL (7210, quadrant mode, MCDRAM only) @ 1.3 GHz (255 threads) | 378 s (x3 improvement w.r.t. KNC) |
| KNL (shared library) | 480 s (25% slower than static library) |

# More memory-efficient, more HPC friendly

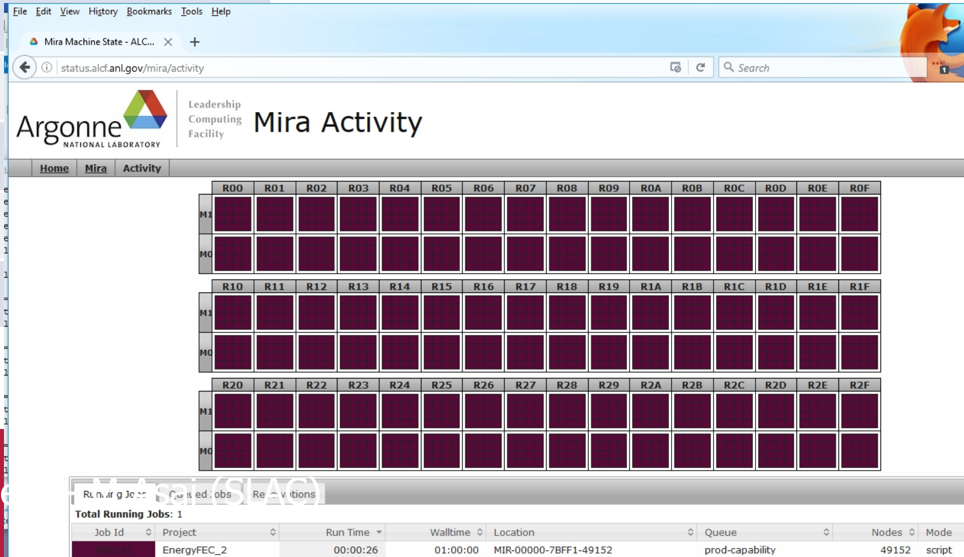
| Version | Intercept | Memory/thread |
|---|---|---|
| 9.6 (seq.) | 113 MB | (113 MB) |
| 10.0.p02-seq | 170 MB | (170 MB) |
| 10.0.p02-MT | 151 MB | 28 MB |
| 10.3.beta-MT | 148 MB | 9 MB |

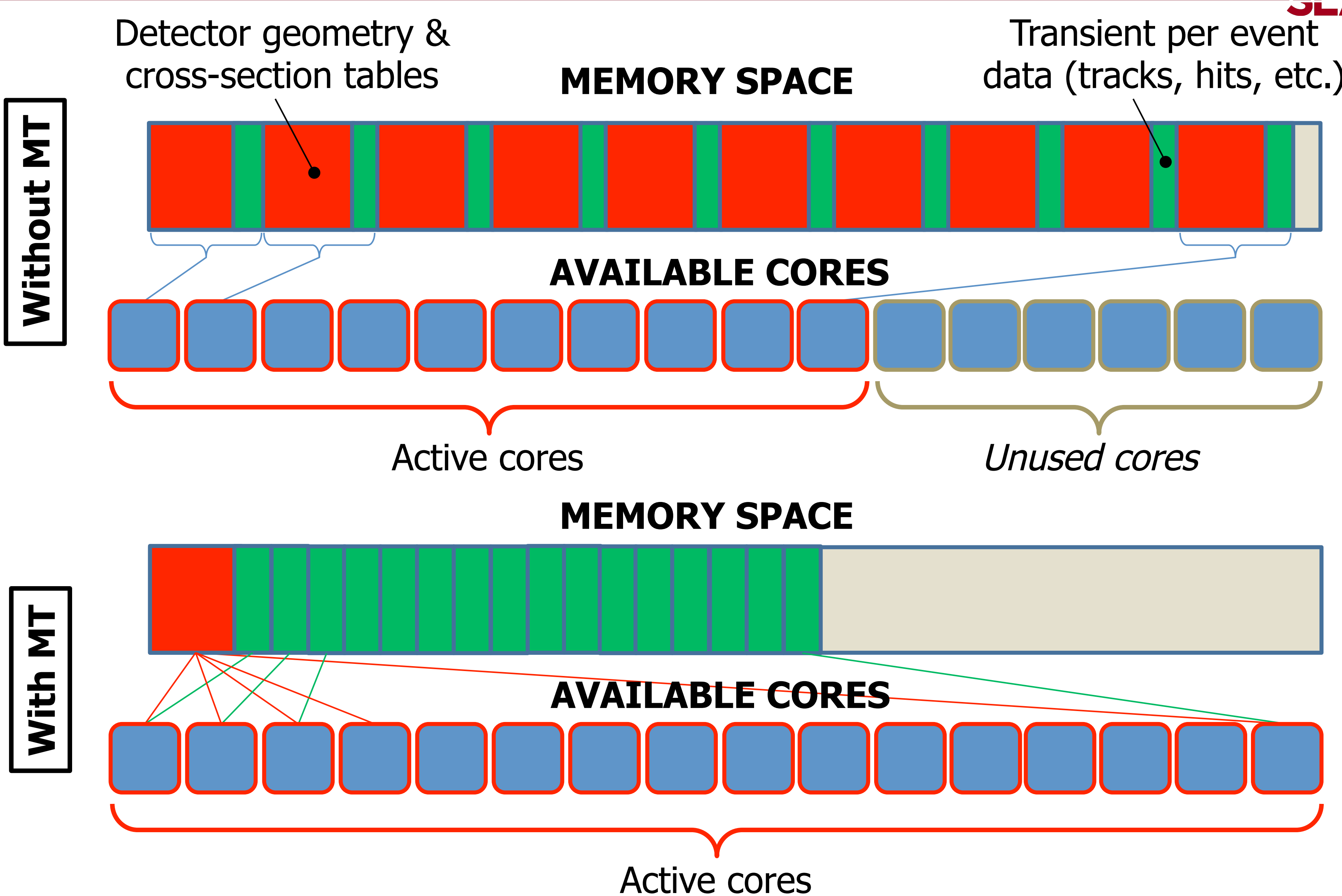
Memory space required for Intel Xeon Phi 3120A
Full-CMS geometry (GDML), 4 Tesla field, 50 GeV pi- (FTFP_BERT)

| # of CPU | # of threads | Speed-up factor | efficiency |
|---|---|---|---|
| 10 | 80 | 79 | 98. 8% |
| 20 | 160 | 158 | 98. 8% |
| 40 | 320 | 317 | 99. 0% |
| 80 | 640 | 626 | 97. 8% |
| 160 | 1280 | 1251 | 97. 7% |
| 320 | 2560 | 2297 | 89. 7% |
| 640 | 5120 | 3555 | 69. 4% |

Tachyon-2 supercomputer @ KISTI (South Korea)
FTFP_BERT physics validation benchmark

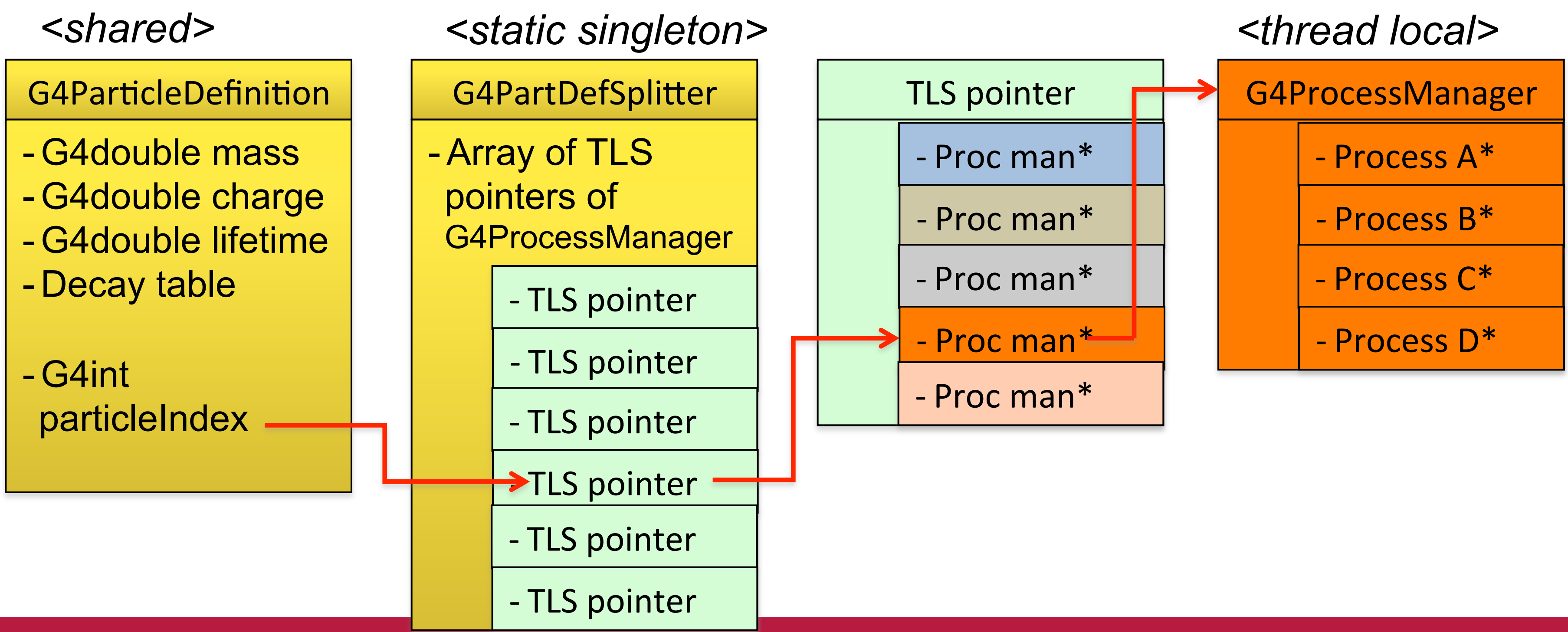
- Geant4 has successfully run with a combination of MT and MPI on Mira Bluegene/Q Supercomputer (@ANL) with all of its 3 million threads
  - Full-CMS geometry & field
- I/O is the limiting factor to scale large concurrent threads:
  - Granular input data files, output data/histograms, etc.
  - 2017 work item
  - Targeting also Cori @ NERSC




Without MT
Detector geometry & cross-section tables
MEMORY SPACE
Transient per event data (tracks, hits, etc.)
AVAILABLE CORES
Active cores
*Unused cores*
With MT
MEMORY SPACE
AVAILABLE CORES
Active cores

# Split class – case of particle definition

- In Geant4, each particle type has its own dedicated object of G4ParticleDefinition class.
  - Static quantities : mass, charge, life time, decay channels, etc.,
    - To be shared by all threads.
  - Dedicated object of G4ProcessManager : list of physics processes this particular kind of particle undertakes.
    - Physics process object must be thread-local.

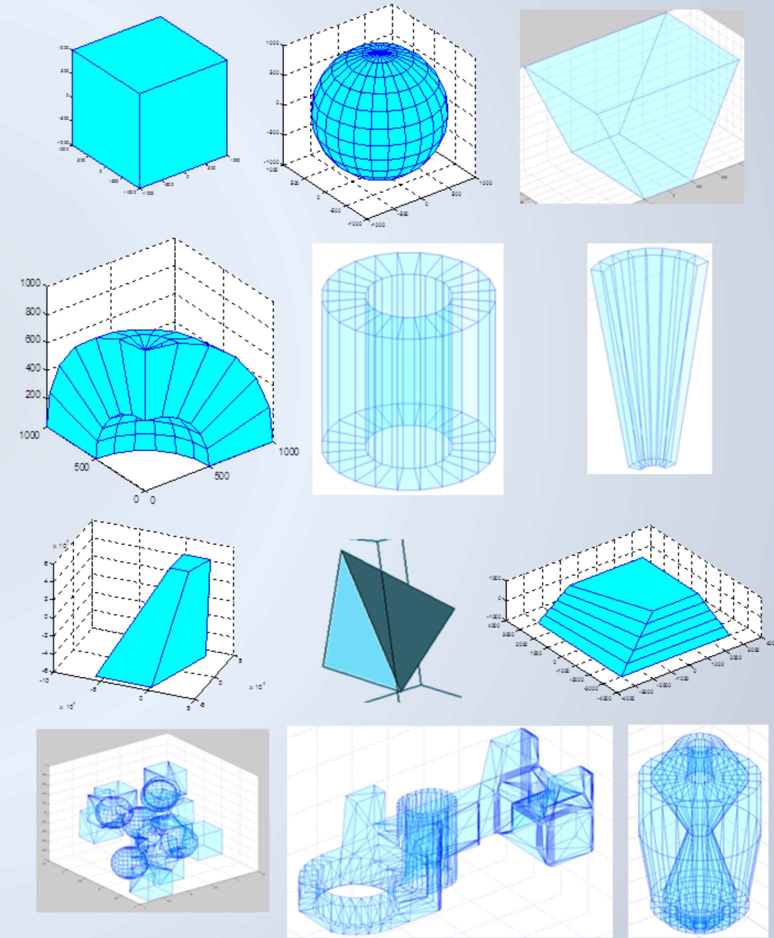


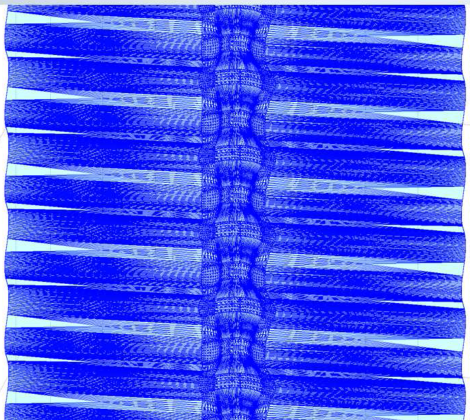
# User's code migration

- If you have a running code with version 9.6 and you want to stick to sequential mode, you do not need to migrate. It should run with version 10.0.
  - Except for a few obsolete interfaces that you had already seen warning messages in v9.6.
- Migration of user's code to multi-threading mode of Geant4 version 10.0 should be fairly easy and straightforward.
  - Migration guide is available.
  - Geant4 users guides are updated with multi-threading features.
  - Most examples have been migrated to multi-threading.
  - Geant4 tutorials based on version 10.0 has already started.
- G4MTRunManager collects run objects from worker threads and "reduces".
- Toughest part of the migration is making user's code thread-safe.
  - It is always a good idea to clearly identify which class objects are thread-local.
- Every file I/O for local thread is a challenge
  - Input : primary events : examples are offered in the migration guide.
  - Output : event-by-event hits, trajectories, histograms

# New features in geometry

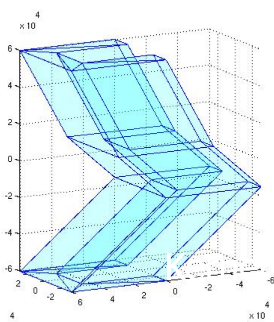
- New command-based geometry overlap checks
  - Built-in check with random points on surfaces
  - Resolution and tolerance are tunable
- Introduction of gravity field and magnetic field gradient
- Optional new AIDA Unified Solid library
  - Optimized implementation of a large number of primitives and constructs
  - For example, the time required to compute intersections with the tessellated solid was dramatically reduced.
- Several new shapes including extruded solid
- New multi-union structure
  - Replace multiple use of binary Boolean unions
  - Far better performance for combining more than O(10) unions





| Method | Speedup |
|---|---|
| Inside | 2423x |
| DistanceToIn | 1334x |
| DistanceToOut | 1976x |

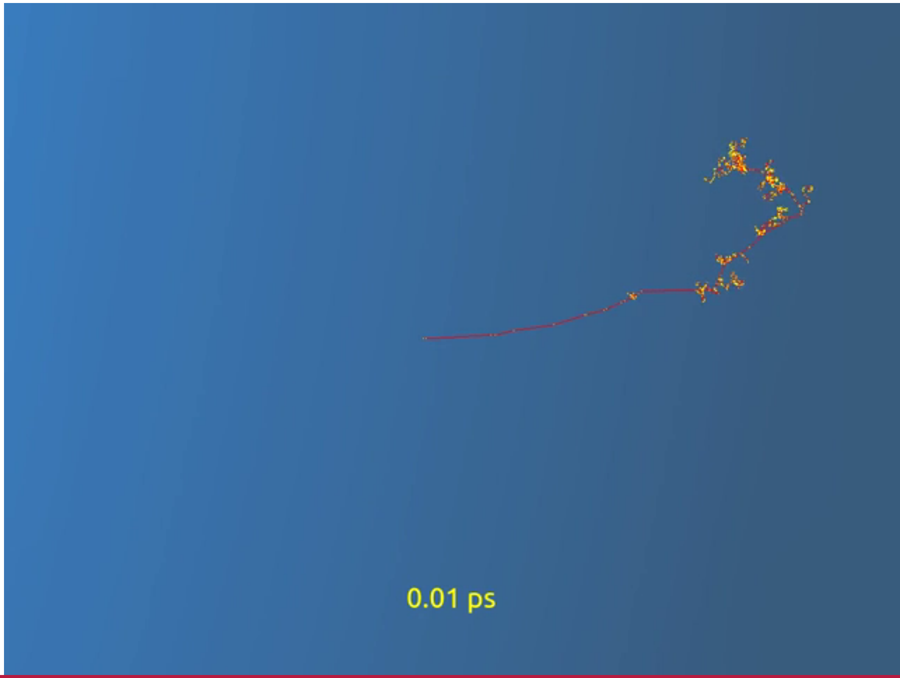
| Information | Value |
|---|---|
| Number of facets | 164.149 |
| Number of voxels | 100.000 |
| Memory saved compared with original Geant4 | 22% (51MB) |



SLAC

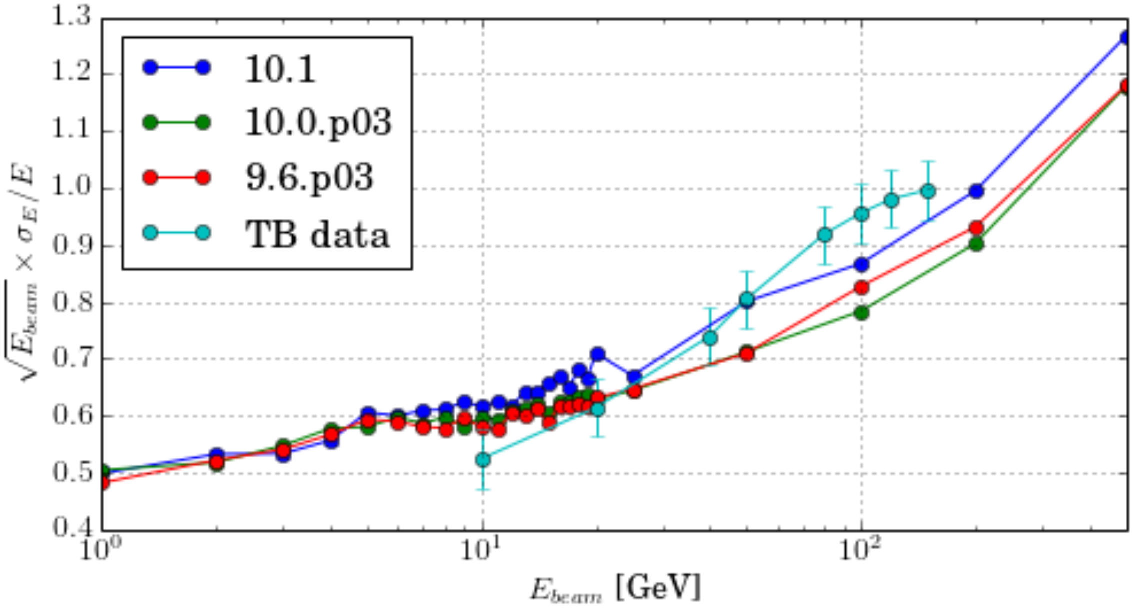
# New features in EM physics

- Multiple/single scattering
  - Introduction of optional displacement on geometrical boundary
  - New G4LowEWenzalVIModel for low-energy applications
- Gamma processes
  - Photo-effect and Compton cross-sections at low-energy integrated
- High-energy models
  - Improvements in gamma->muons, positron->hadrons and positron->muons
  - Synchrotron radiation for all particle types
- Atomic de-excitation
  - New alternative fluorescence dataset (Bearden)
- New radiolysis process for water and silicon
  - Physics stage followed by physico-chemical and chemical stage
- Introduction of phonon transport with a new concept of crystal
- Channeling effect in straight and bent crystal
- Lots of code refinements along with MT

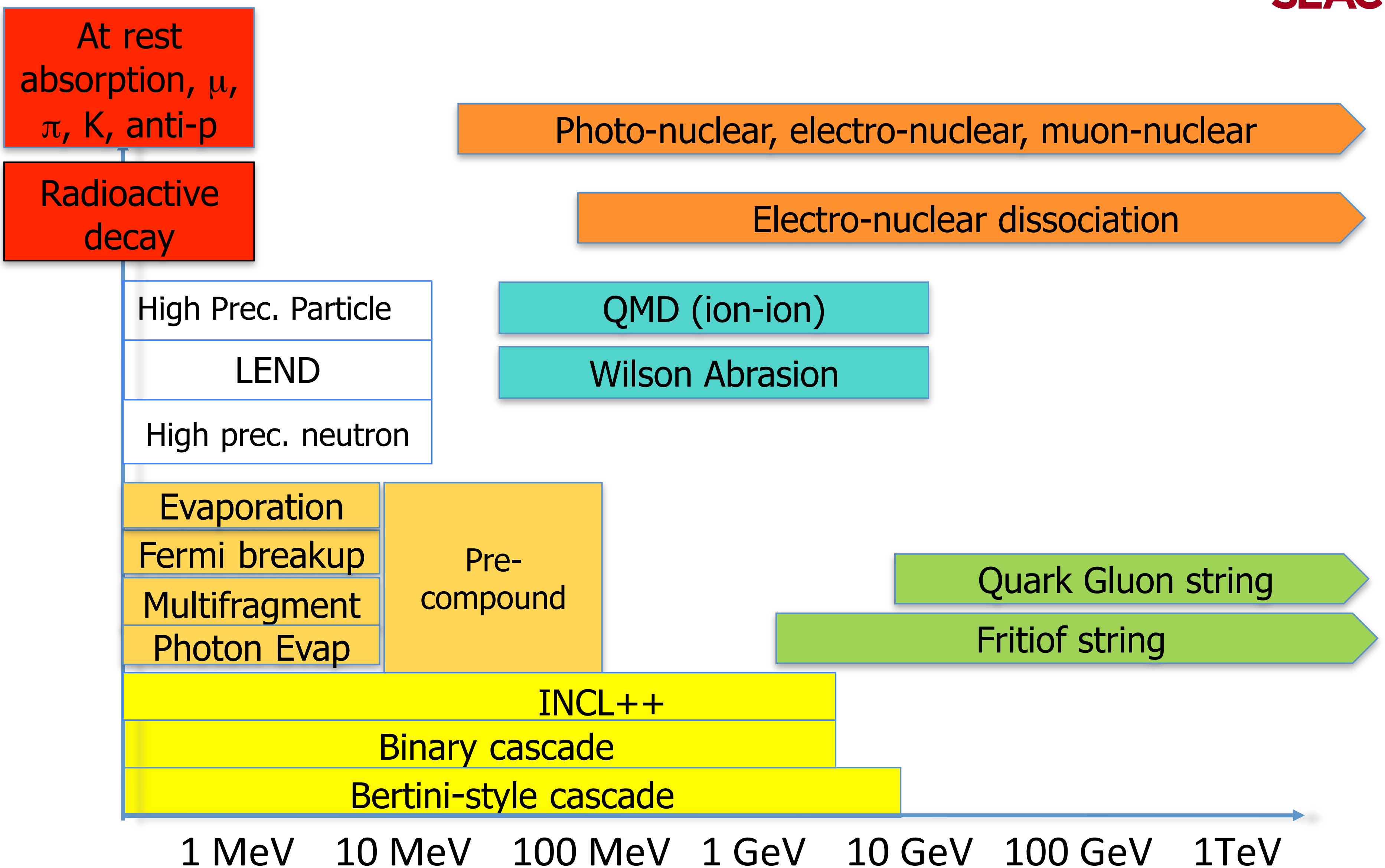


# New features in hadronic physics

- FTFP_BERT is now the recommended physics list for most of high-energy use-cases
- Generation of Isomer (a.k.a. metastable nuclides)
  - by default lifetime > 1 nsec
- Neutron_HP is extended to Particle_HP to cover p, d, t, $\alpha$
- Alternative low-energy neutron model with GND (Generalized Nuclear Data) format
- Liege intra-nuclear cascade model (INCLXX) extended up to 20 GeV
- FTF model extended to nucleus-nucleus and antinucleus-nucleus interactions
- Radioactive decay redesigned with rare decay channels
- New hadron stopping models based on Bertini
- Decommission of LHEP and CHIPS models

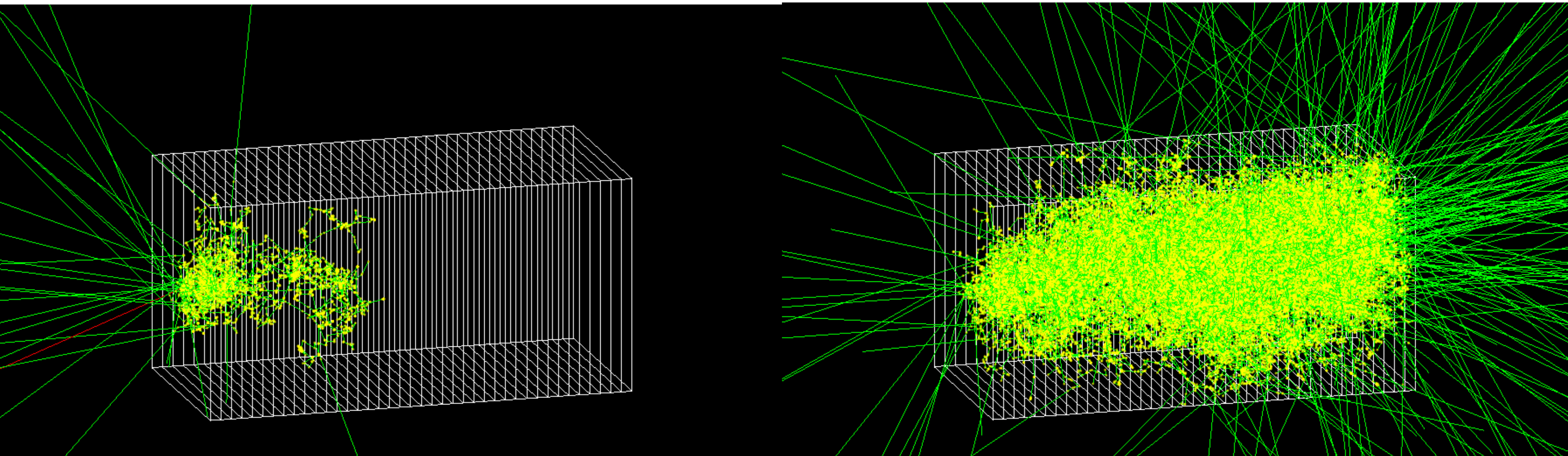


# Hadronic Model Inventory



# New biasing scheme

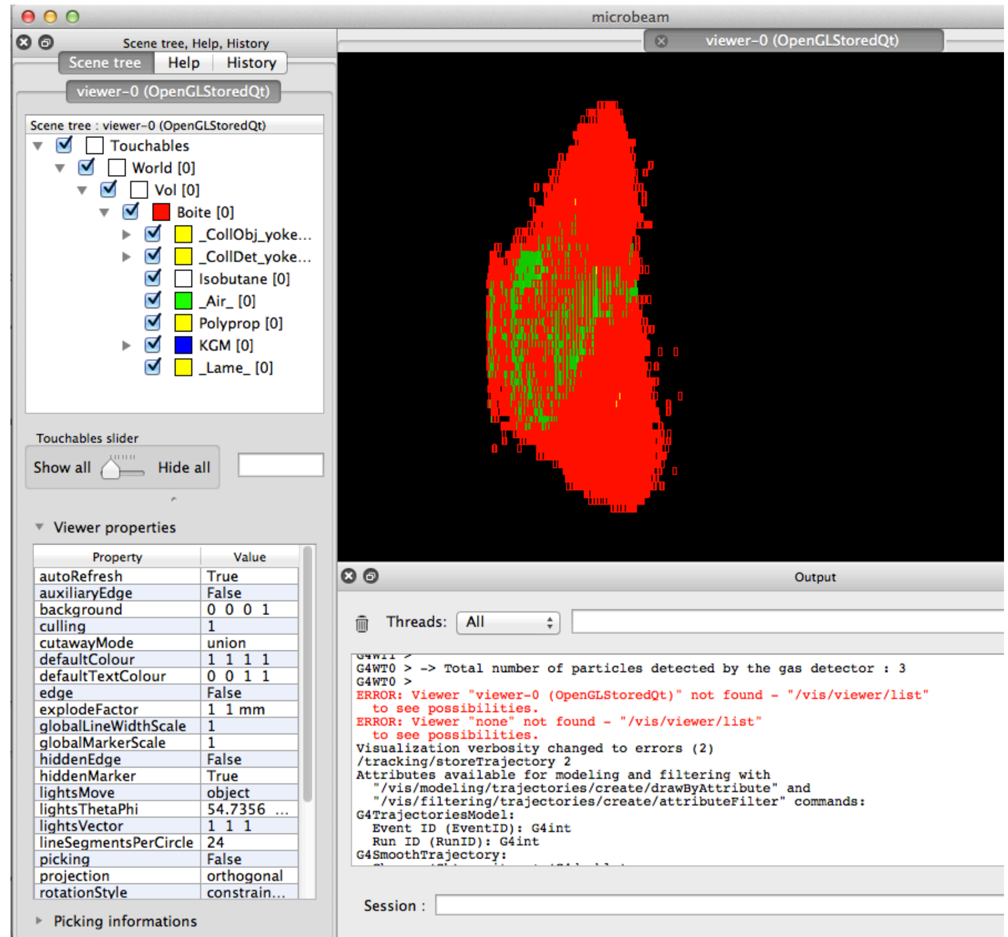
- Event biasing (a.k.a. variance reduction) scheme has been fully revised at version 10.
- It allows treating many biasing options in coherent manner.
- Such options include:
  - Physics process biasing : alters physics process
    - Cross-section biasing, forced interaction, forced passage, etc.
    - Biasing final products of an interaction, e.g. distribution
  - Non-physics biasing : alters the transportation of particle
    - Geometrical importance, splitting / Russian roulette, weight window, etc.
- Easily extensible to new (or user-defined) options
- Well-integrated with built-in scoring functionalities.
- New examples are available.

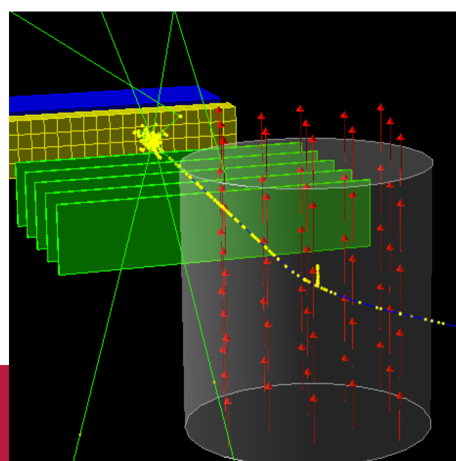


Five of 100 MeV neutrons on Pb/Scinti calorimeter.
Left : analogue simulation Right : splitting with geometrical importance and probability control.

# New features in analysis, GUI and visualization

- New built-in fully-multithreaded histogramming tool
  - 1-D and 2-D histograms and scatter plots, n-tuples
  - Data format compatible with ROOT, XML, AIDA, CSV
    - Extensible to other format
- GUI and visualization
  - New Qt driver with OpenGL
    - Viewer properties and picking panel, dock-able widgets
    - Multithread output filtering
  - More than 30% faster drawing on OpenGL
  - Magnetic field lines



# Geant4 – A Simulation Toolkit

**http://www.geant4.org/**

S. Agostinelli et al.
**Geant4: a simulation toolkit**
NIM A, vol. 506, no. 3, pp. 250-303, 2003

J. Allison et al.
**Geant4 Developments and Applications**
IEEE Trans. Nucl. Sci., vol. 53, no. 1, pp. 270-278, 2006

Scopus

Search Sou

S. Agostinelli et al.

**Geant4: a simulation toolkit**

NIM A, vol. 506, no. 3, pp. 250-303, 2003

## 10,258 documents have cited:

Back

GEANT4 - A simulation tooli

Agostinelli S., Allison J., Amak

(2003) Nuclear Instruments a

Set feed

Search within results...

Refine results

Limit to Exclude

Access type

Year

2018

Cited by

0

### Filter by year

| | | | |
|---|---|---|---|
| 2018 | (505) | 2008 | (427) |
| 2017 | (1,112) | 2007 | (450) |
| 2016 | (1,094) | 2006 | (315) |
| 2015 | (1,189) | 2005 | (238) |
| 2014 | (941) | 2004 | (181) |
| 2013 | (996) | 2003 | (36) |
| 2012 | (971) | 2002 | (1) |
| 2011 | (688) | | |
| 2010 | (548) | | |
| 2009 | (566) | | |

### Filter by subject area

| | | | | | |
|---|---|---|---|---|---|
| Physics and Astronomy | (8,313) | Multidisciplinary | (249) | Neuroscience | (4) |
| Medicine | (1,635) | Environmental Science | (186) | Decision Sciences | (3) |
| Engineering | (1,338) | Chemistry | (173) | Economics, Econometrics and Finance | (1) |
| Energy | (688) | Agricultural and Biological Sciences | (58) | | |
| Mathematics | (590) | Chemical Engineering | (47) | | |
| Health Professions | (525) | Social Sciences | (20) | | |
| Biochemistry, Genetics and Molecular Biology | (459) | Pharmacology, Toxicology and Pharmaceutics | (14) | | |
| Materials Science | (380) | Nursing | (7) | | |
| Earth and Planetary Sciences | (356) | Immunology and Microbiology | (6) | | |
| Computer Science | (283) | Arts and Humanities | (5) | | |

ScienceDirect

Journals Bo

J. Allison et al.

**Recent developments in Geant4**

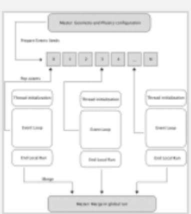
NIM A, vol. 835, pp. 186-225, 2016

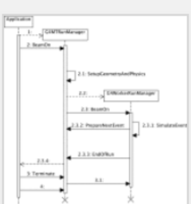
Advanced search

**Article outline** Show full outline



**Figures and tables**






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],

Show more



https://doi.org/10.1016/j.nima.2016.06.125 Get rights and content




## Highlights

- Multithreading resulted in a smaller memory footprint and nearly linear speed-up.
- Scoring options, faster geometry primitives, more versatile visualization were added.
- Improved electromagnetic and hadronic models and cross sections were developed.

# Download of recent 90 days

ELSEVIER

SEARCH MENU




## Nuclear Instruments and Methods in Physics Research Section A:

Most Downloaded | Recent Articles | Most Cited | Open Access Articles

Recent developments in Geant4 J. Allison | K. Amako | ...

Geant4—a simulation toolkit S. Agostinelli | J. Allison | ...

The gas electron multiplier (GEM): Operating principles and applications Fabio Sauli

View All Articles >

accelerators, ion sources, beam transport systems and target arrangements...

Read more

Guide for Authors

Abstracting/ Indexing

Track Your Paper

Order Journal

Most Downloaded | Recent Articles | Most Cited | Open Access Articles

Recent developments in Geant4 J. Allison | K. Amako | ...

Geant4—a simulation toolkit S. Agostinelli | J. Allison | ...

The gas electron multiplier (GEM): Operating principles and applications Fabio Sauli

> View All Most Downloaded Articles

Feedback

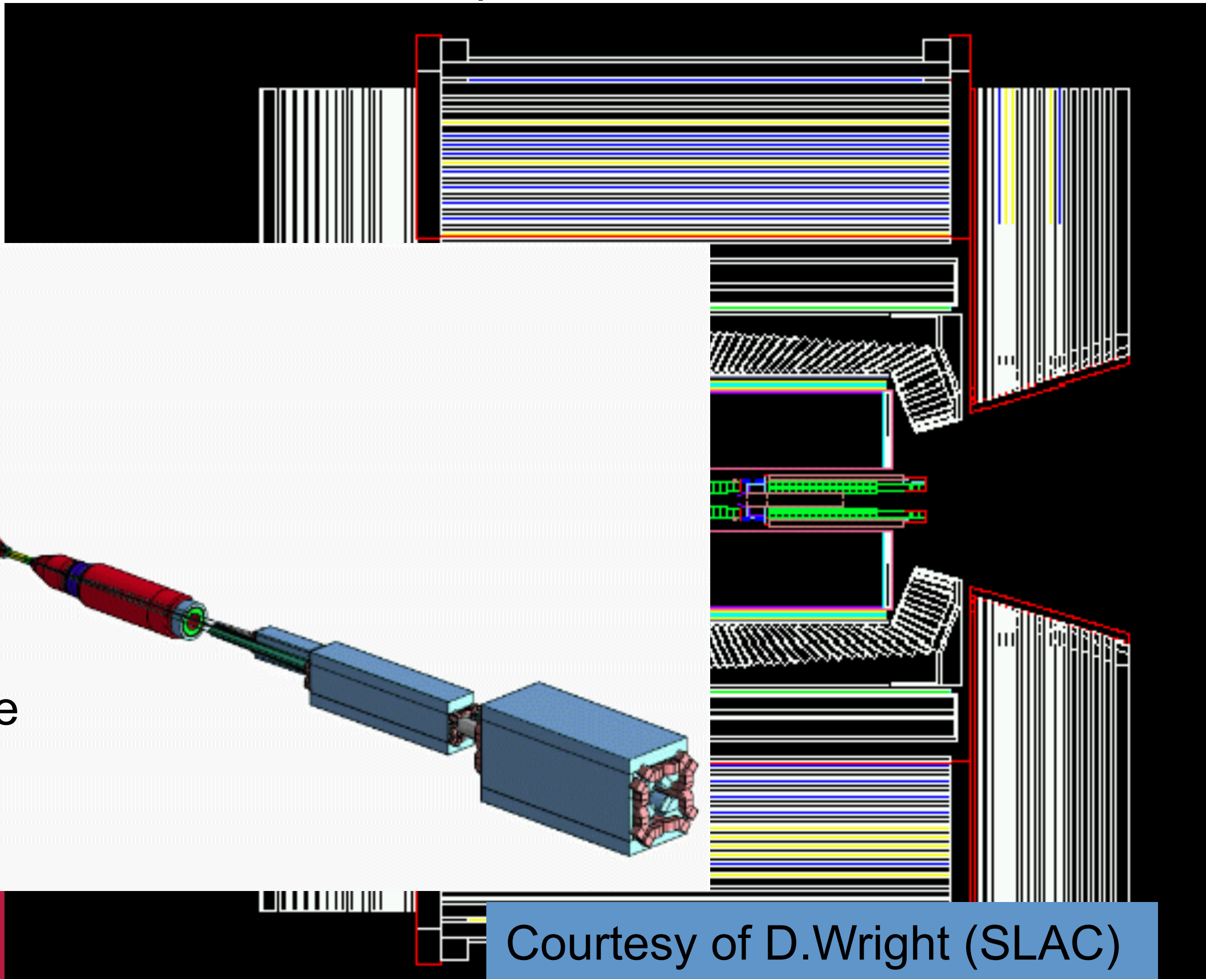
Version 10.4-p02

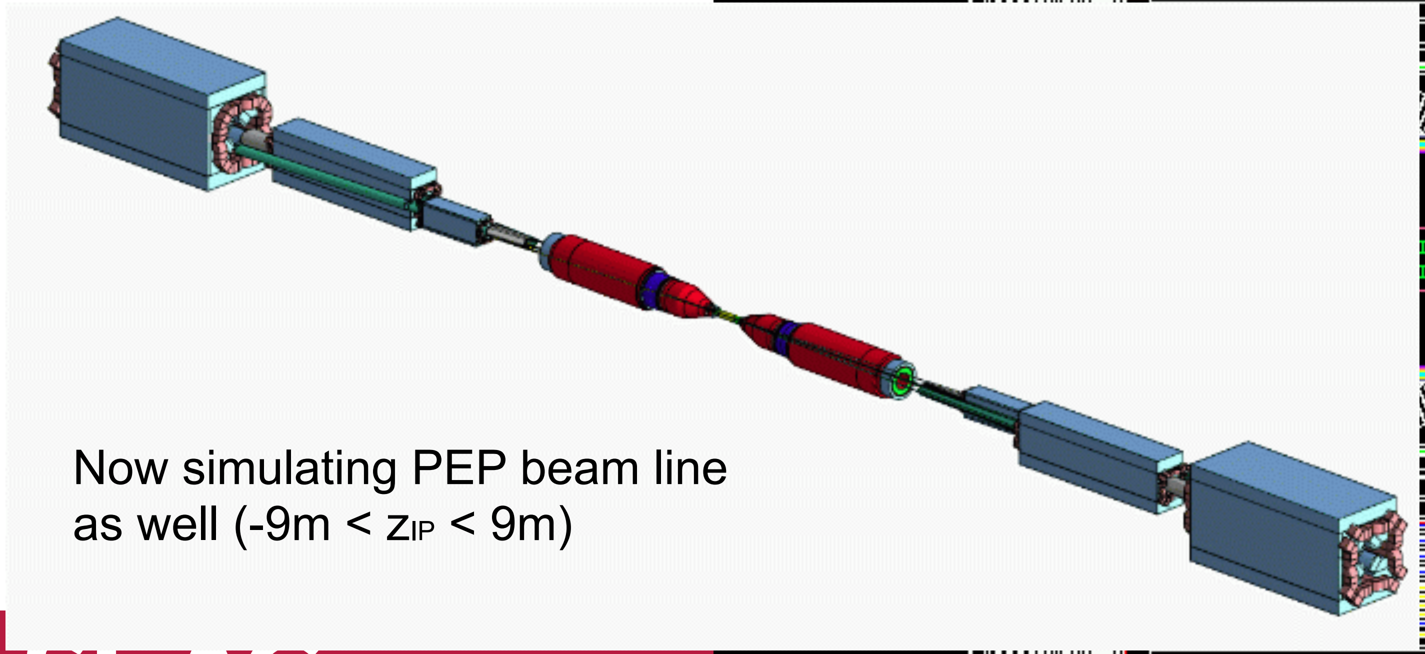
# Highlights of Users Applications

To provide you some ideas how Geant4 would be utilized…

# BaBar

- BaBar at SLAC is the pioneer experiment in HEP in use of Geant4
  - Started in 2000
  - Simulated ~2*10$^{10}$ events so far
  - Produced at 20 sites in North America and Europe

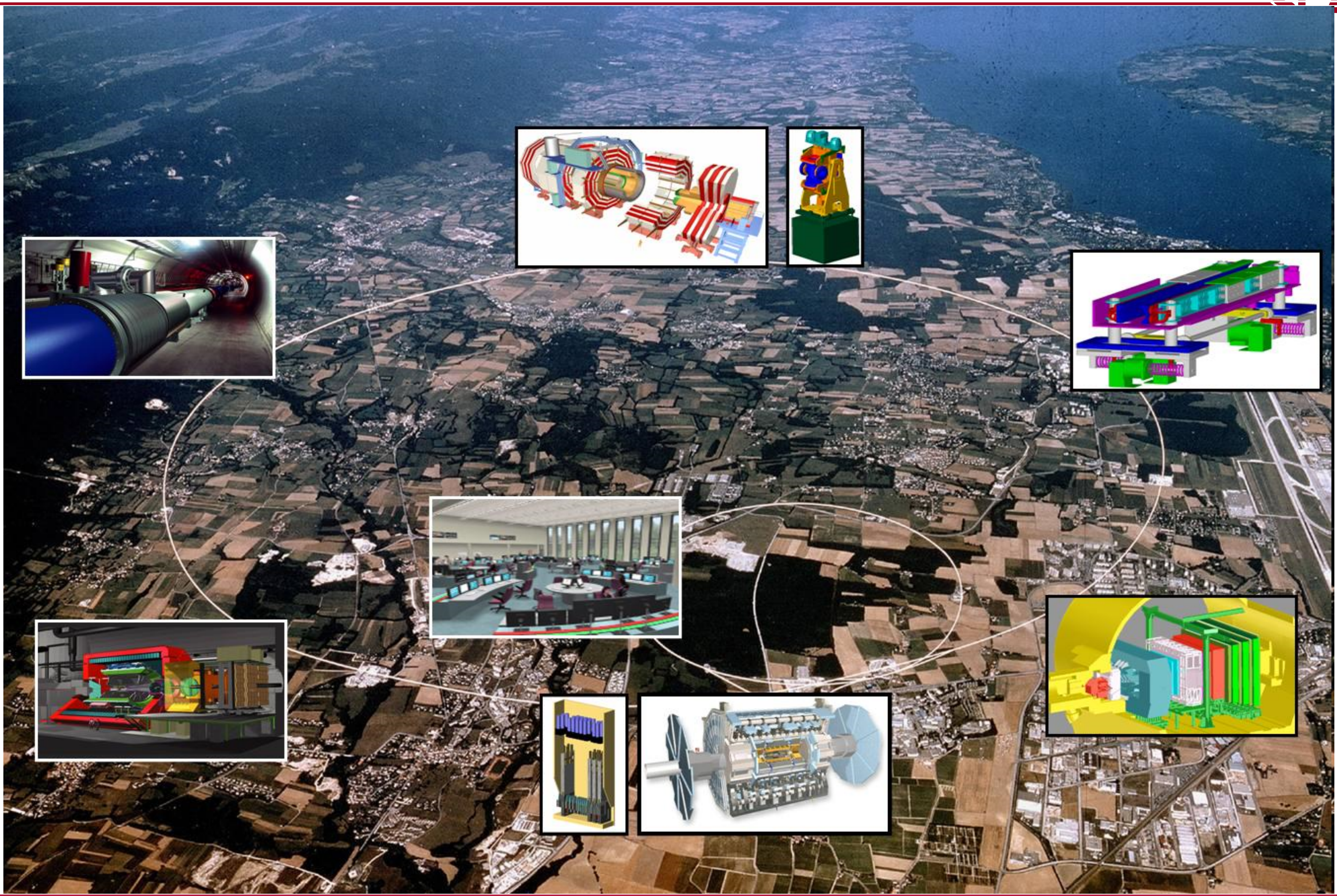




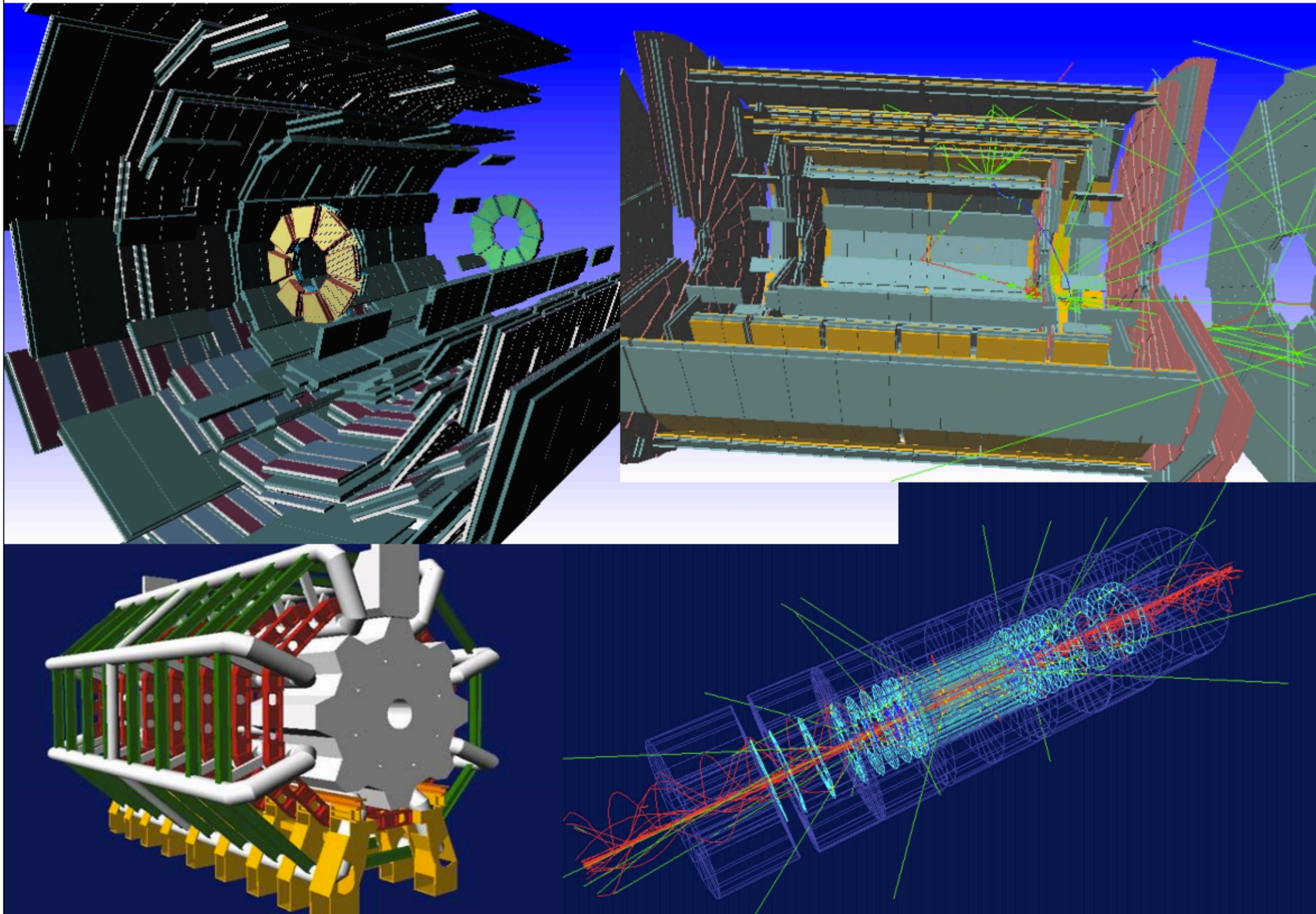
Now simulating PEP beam line as well ($-9m < z_{IP} < 9m$)

Courtesy of D.Wright (SLAC)

# Large Hadron Collider (LHC) @ CERN



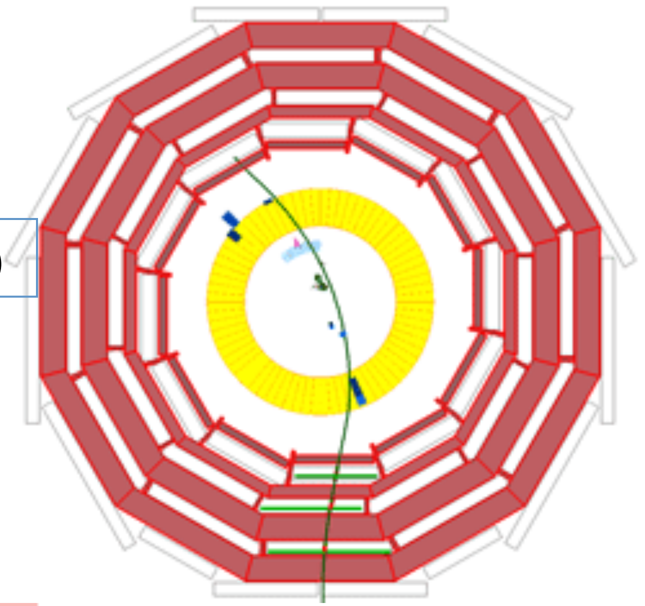
# Geant4 in High Energy Physics (ATLAS at LHC)



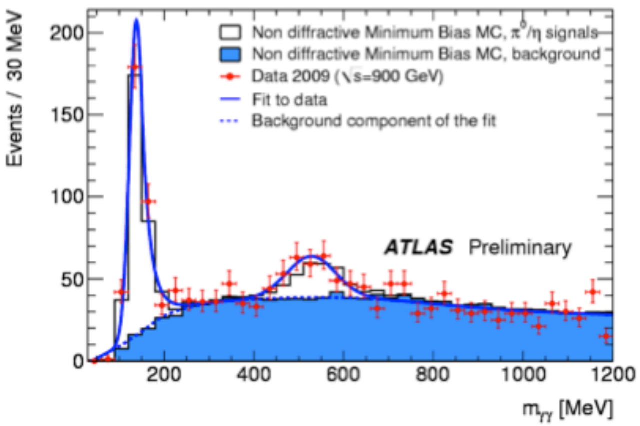
# Geant4 has been successfully employed for

- Detector design
- Calibration / alignment
- First analyses

T. LeCompte (ANL)



## GEANT4 Comparisons with the Calorimeters




Invariant mass of pairs of well-isolated electromagnetic clusters.

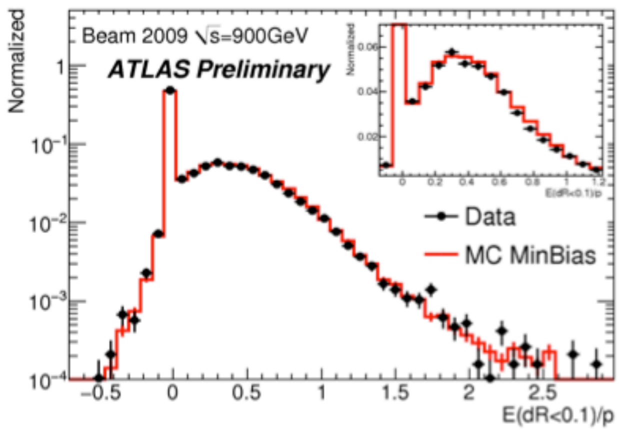
The $\pi^0$ mass is within 0.8 ± 0.6% of expectations.

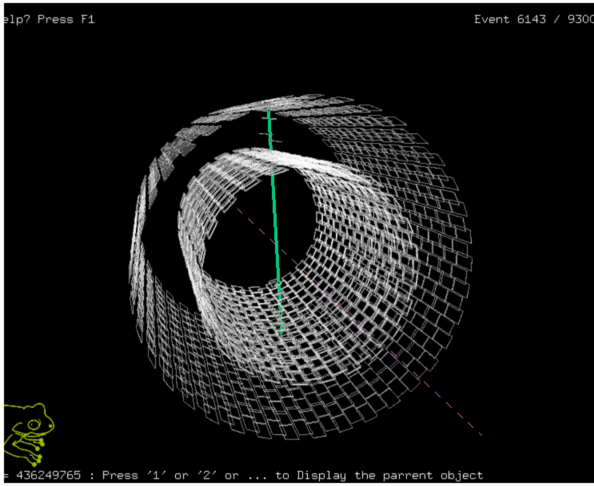
The $\eta^0$ mass is within 3 ± 2% of expectations.

The detector uniformity is better than 2%.

Response of the calorimeter to single isolated tracks. To reduce the effect of noise, topological clusters are used in summing the energy.

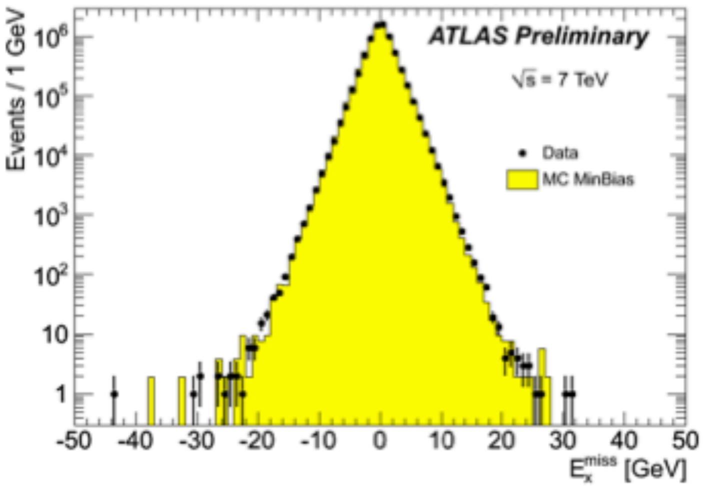
This plot agreed better than we ever expected. (I sent the student who made it back to make sure that they didn't accidentally compare G4 with G4.

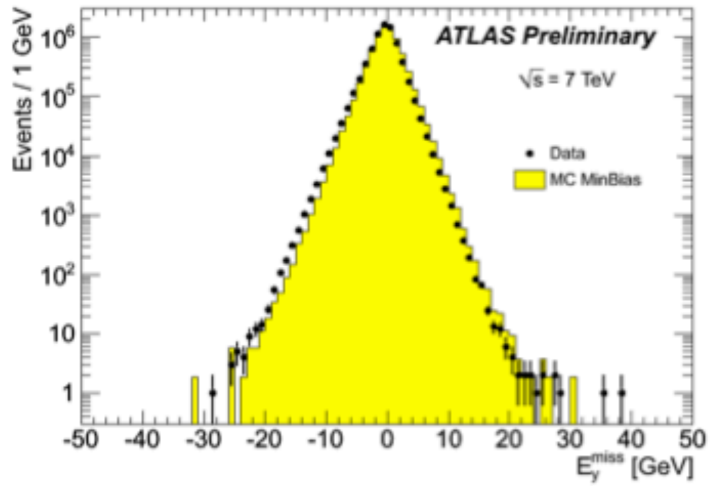


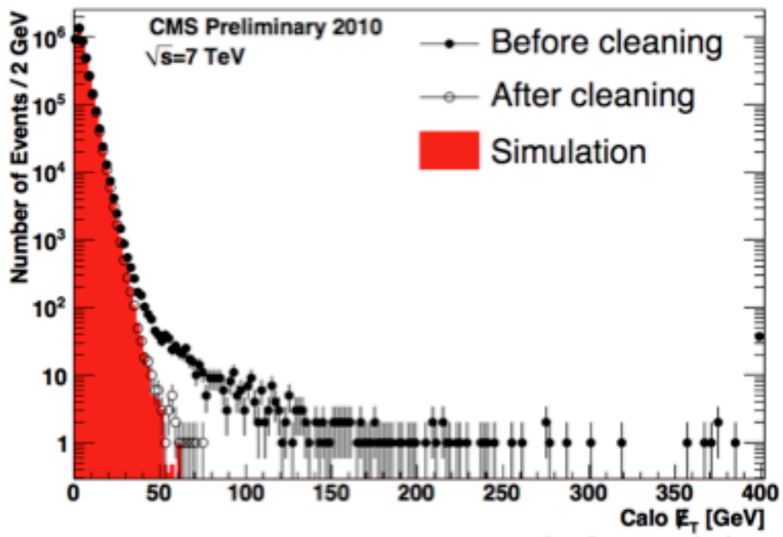
Figures from CMS

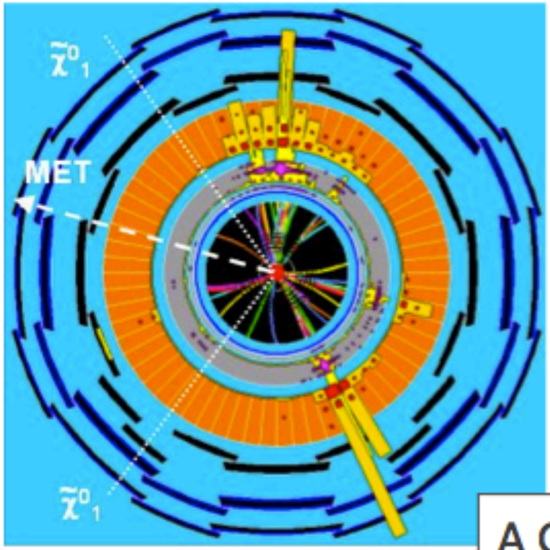
# Missing $E_T$




This is one of the hardest things to get right. MET incorporates everything measured in the detector and attempts to identify non-interacting particles, such as neutrinos or dark matter.




Agreement is astounding.

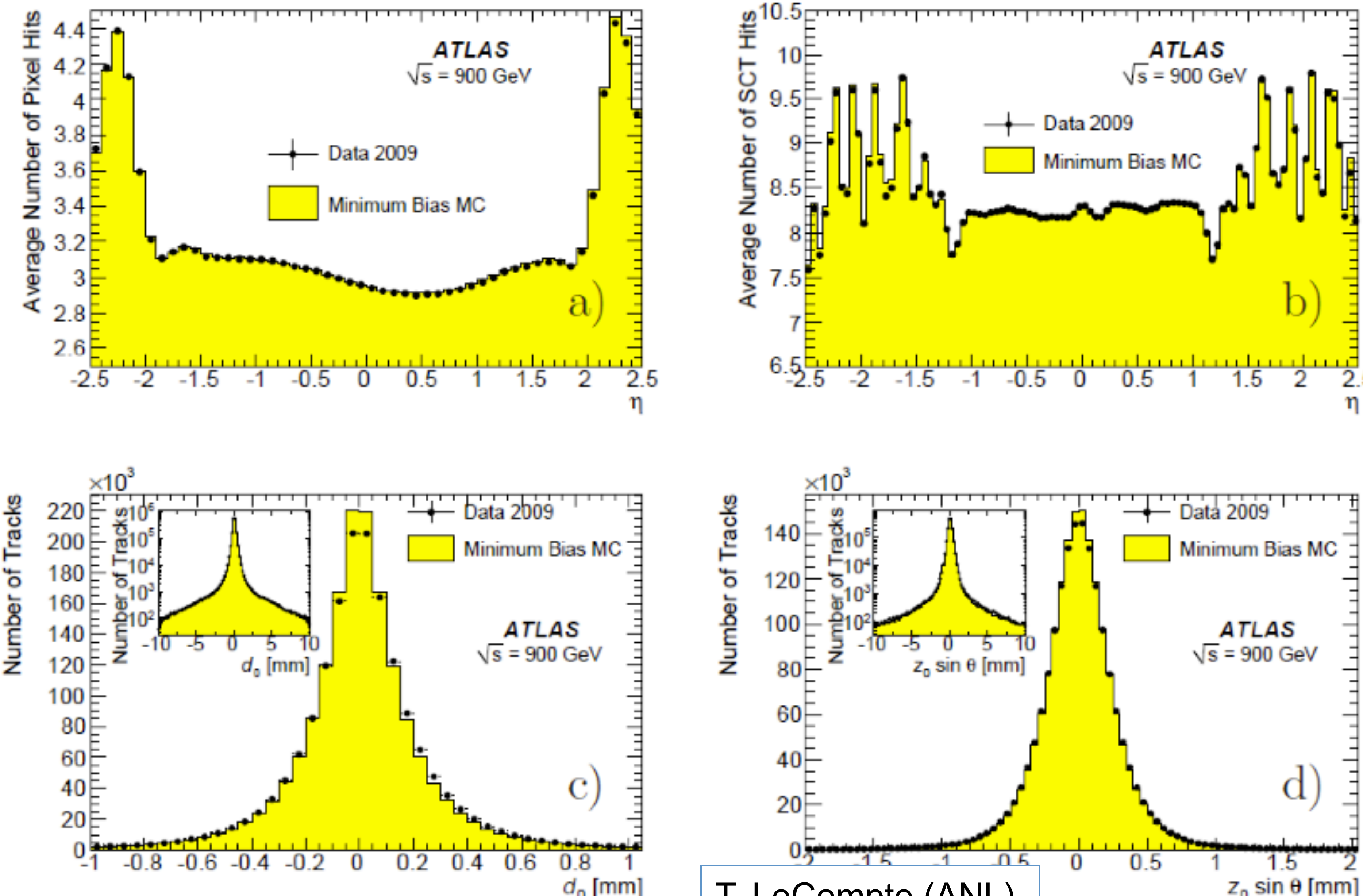
You can even see that the ATLAS detector is not quite centered – in both data and MC.

A GEANT4 event.

Both ATLAS and CMS plots are made from a tiny piece of the very earliest data.

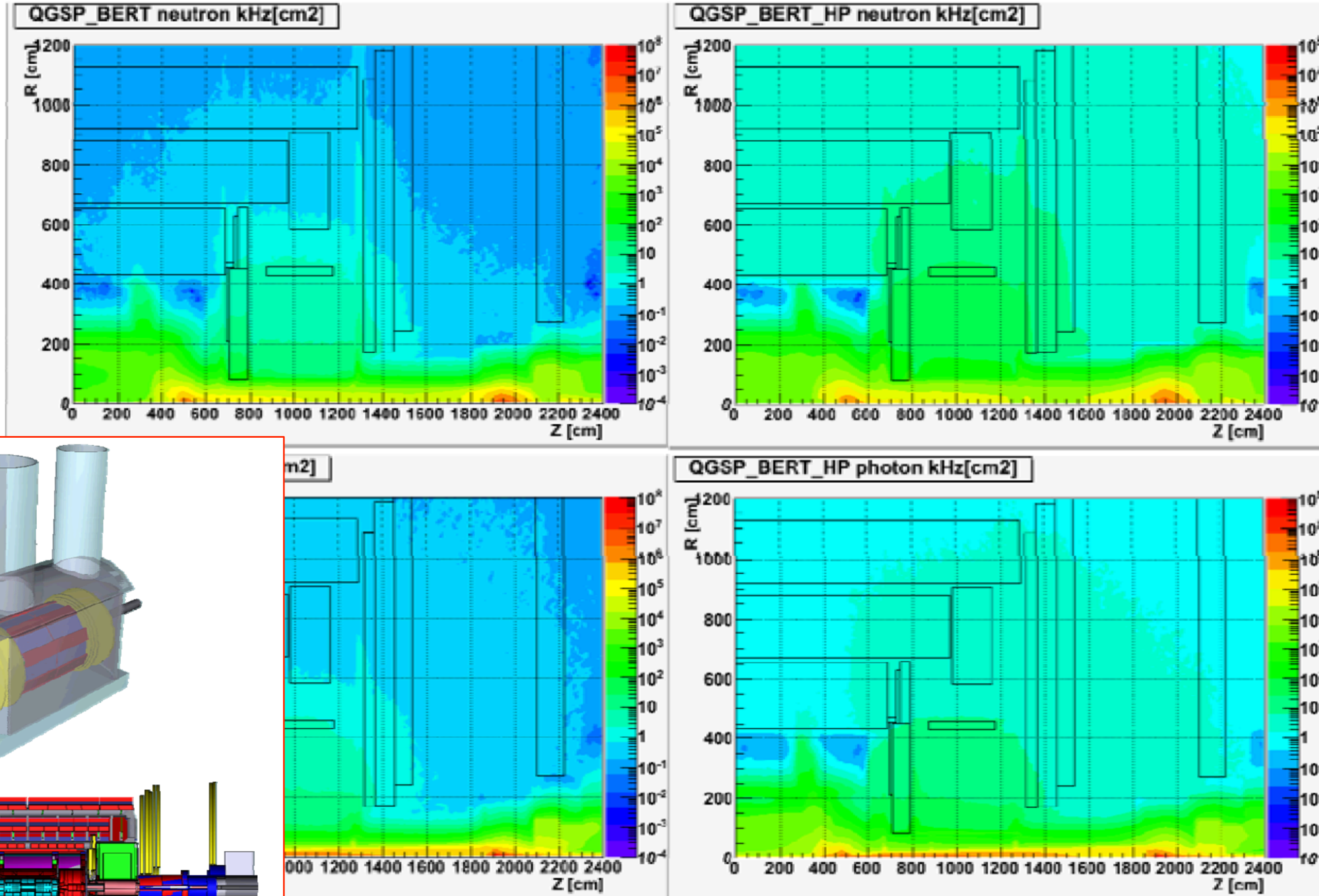
T. LeCompte (ANL)

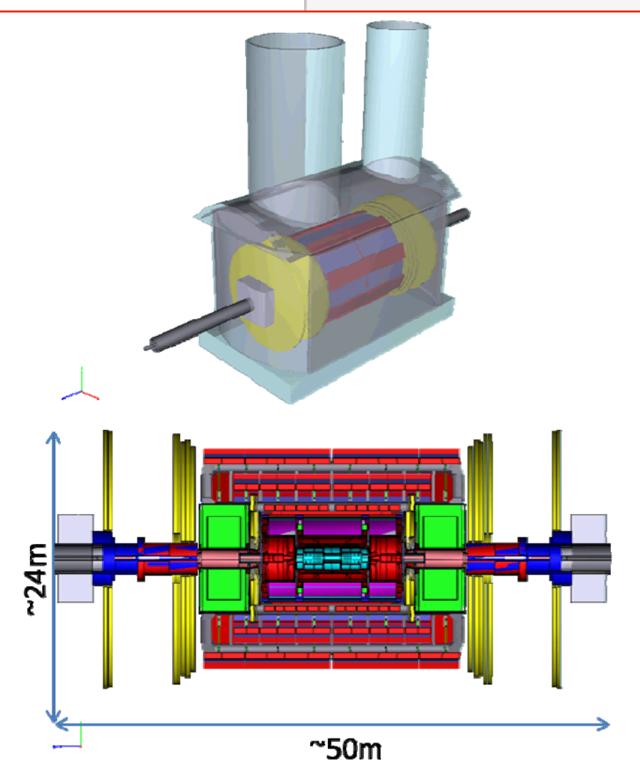
# Data and simulation agreements

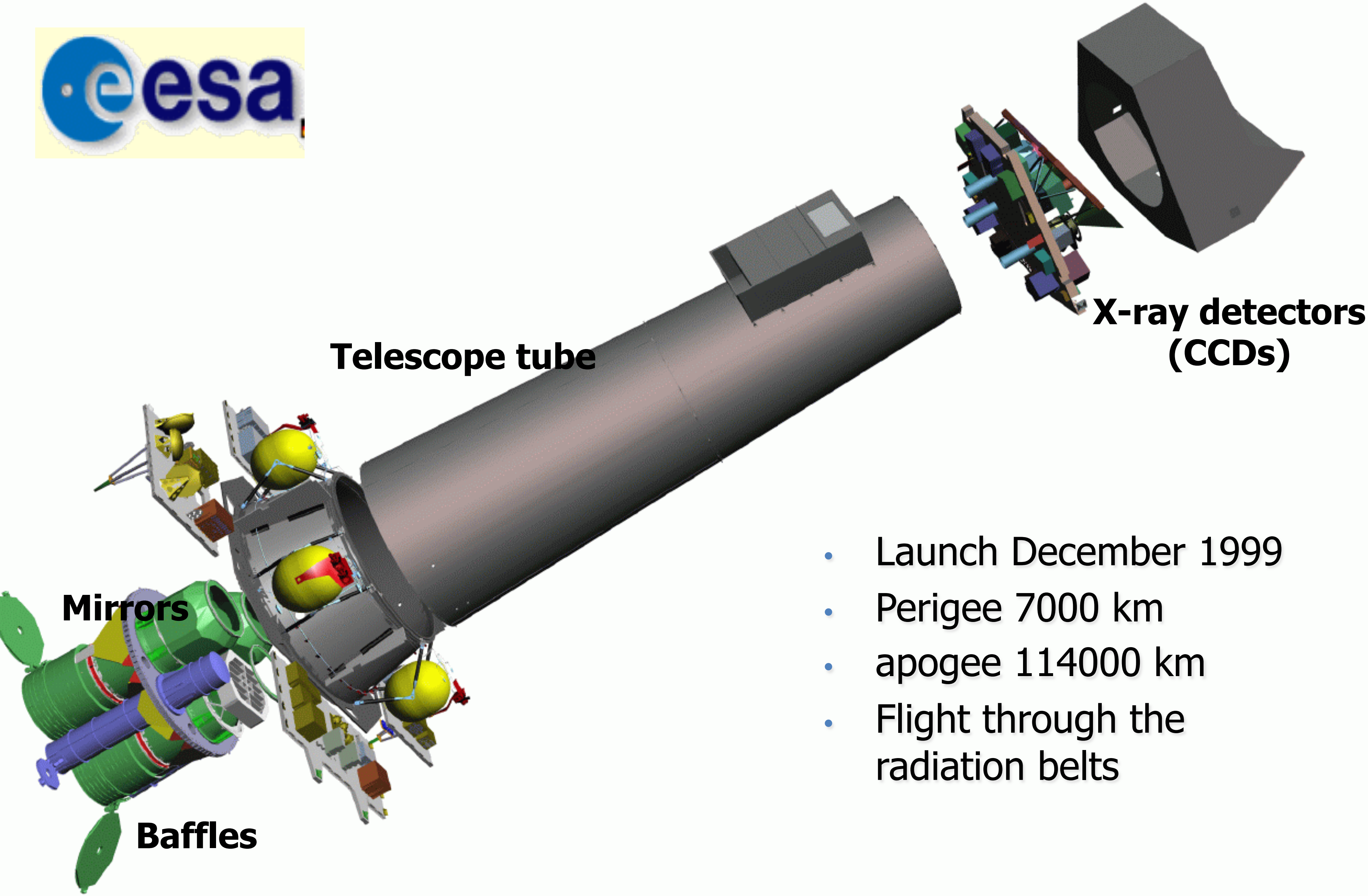



T. LeCompte (ANL)

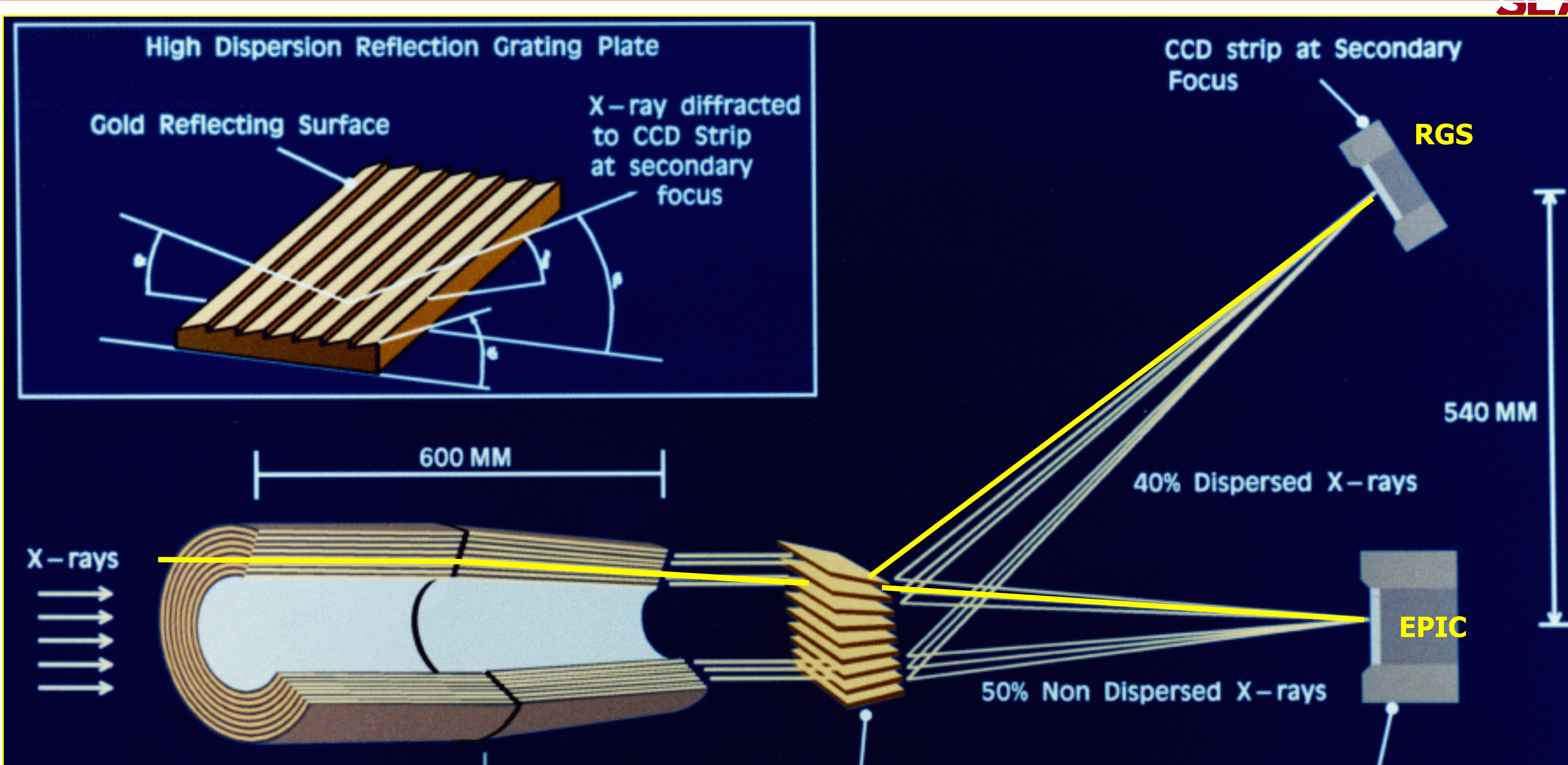
# ATLAS cavern background study






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




Low-E (~100 keV to few MeV), low-angle (~0°-5°) proton scattering:

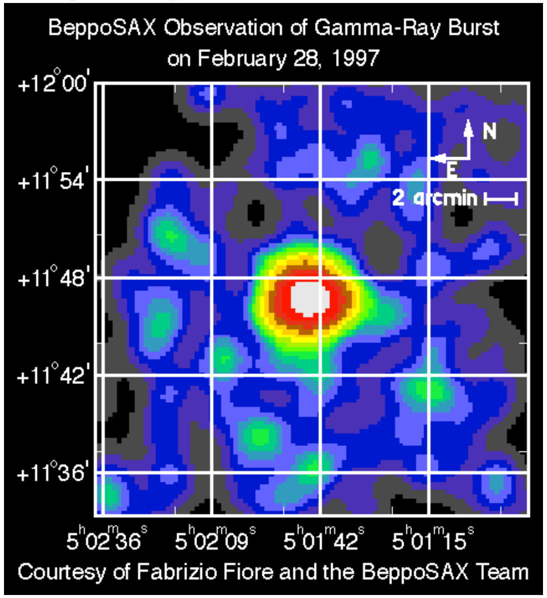
Obscure problem; not much analysed

As the outcome of the simulation, beam shutter was introduced.

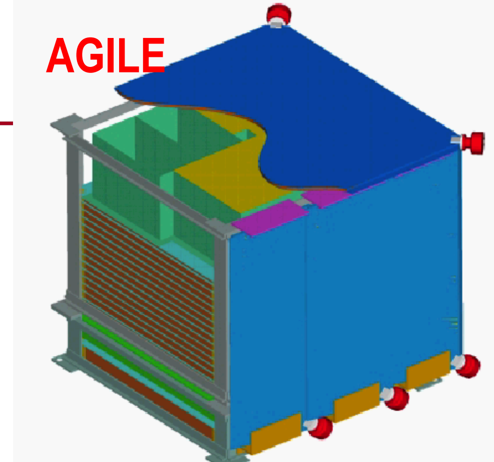
The shutter is closed while XMM is passing through the radiation belt.

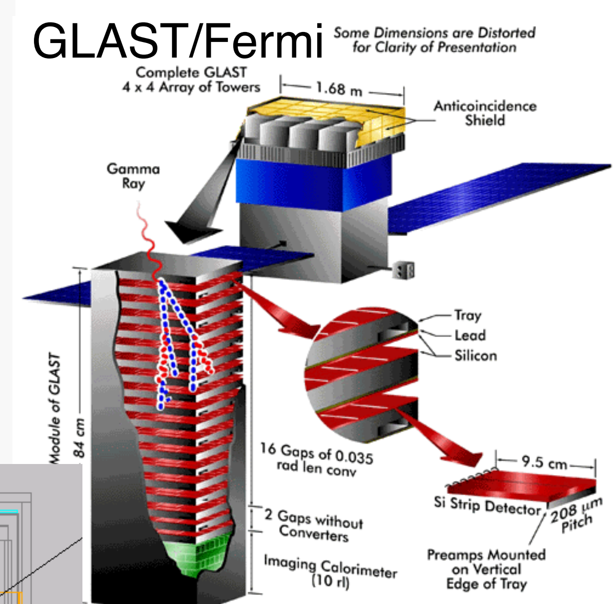
# $\gamma$ astrophysics

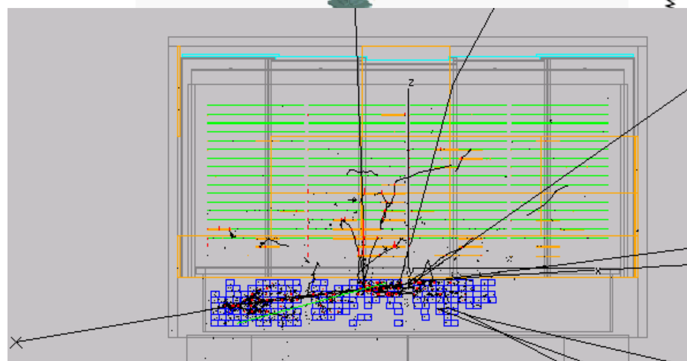
## $\gamma$-ray bursts



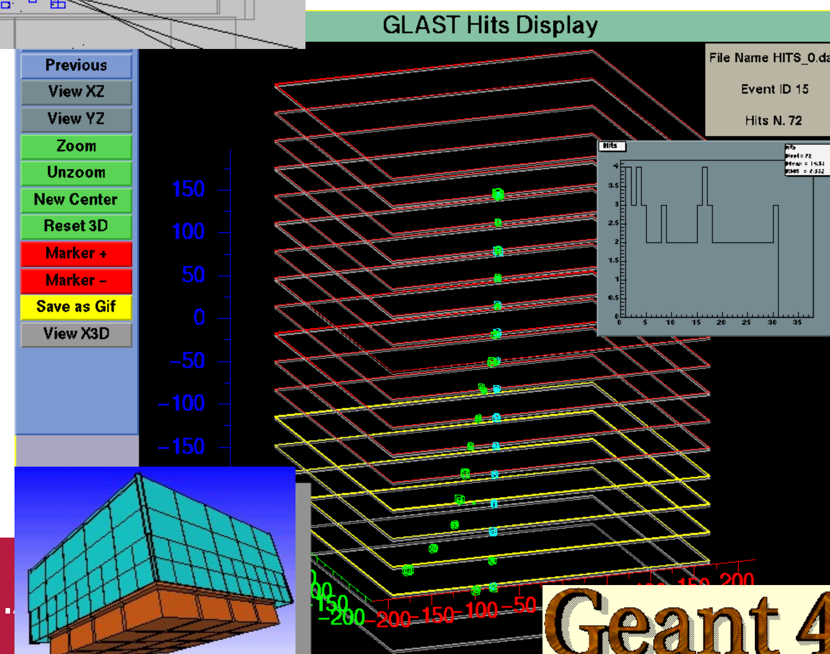

AGILE

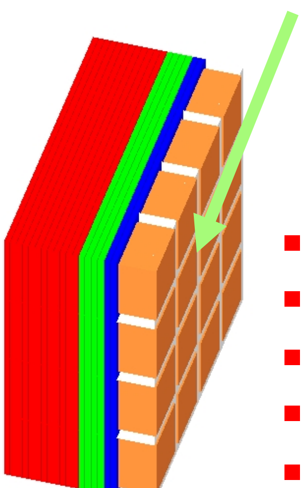




GLAST / Fermi



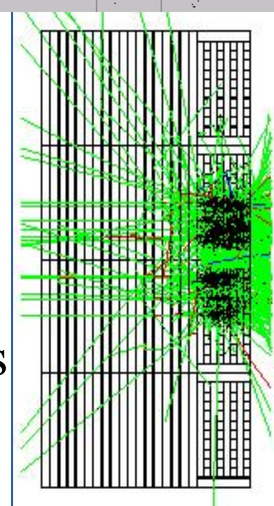
Typical telescope:

*Tracker*

*Calorimeter*

*Anticoincidence*

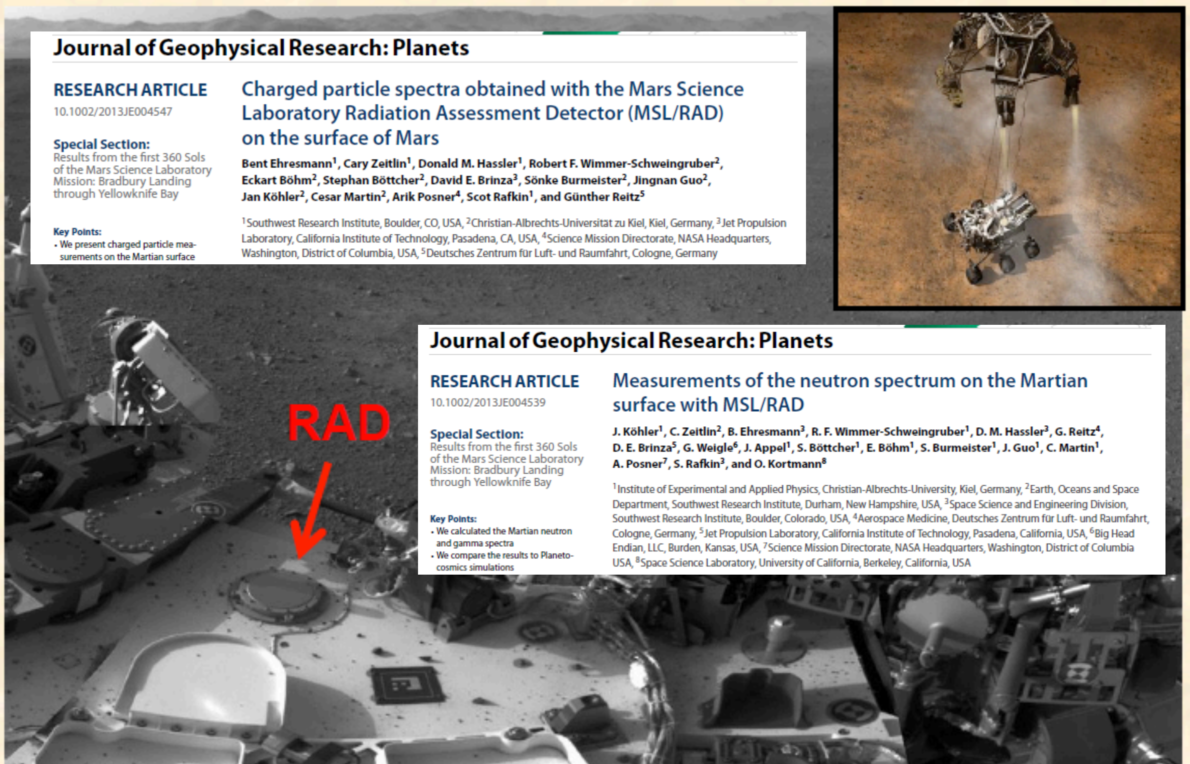
- $\gamma$ conversion
- electron interactions
- multiple scattering
- $\delta$-ray production
- charged particle tracking



# Geant4 in space

# MSL Radiation Assessment Detector (RAD)

- RAD is a compact, highly capable radiation analyzer to characterize the full spectrum of space radiation (both charged & neutral particle).
- MSL RAD is currently characterizing the radiation environment on the surface of Mars.




Journal of Geophysical Research: Planets

RESEARCH ARTICLE
10.1002/2013JE004547

**Special Section:** Results from the first 360 Sols of the Mars Science Laboratory Mission: Bradbury Landing through Yellowknife Bay

**Key Points:**
- We present charged particle measurements on the Martian surface

Charged particle spectra obtained with the Mars Science Laboratory Radiation Assessment Detector (MSL/RAD) on the surface of Mars

Bent Ehresmann[1], Cary Zeitlin[1], Donald M. Hassler[1], Robert F. Wimmer-Schweingruber[2], Eckart Böhm[2], Stephan Böttcher[2], David E. Brinza[3], Sönke Burmeister[2], Jingnan Guo[2], Jan Köhler[2], Cesar Martin[2], Arik Posner[4], Scot Rafkin[1], and Günther Reitz[5]

[1]Southwest Research Institute, Boulder, CO, USA, [2]Christian-Albrechts-Universität zu Kiel, Kiel, Germany, [3]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA, [4]Science Mission Directorate, NASA Headquarters, Washington, District of Columbia, USA, [5]Deutsches Zentrum für Luft- und Raumfahrt, Cologne, Germany

Journal of Geophysical Research: Planets

RESEARCH ARTICLE
10.1002/2013JE004539

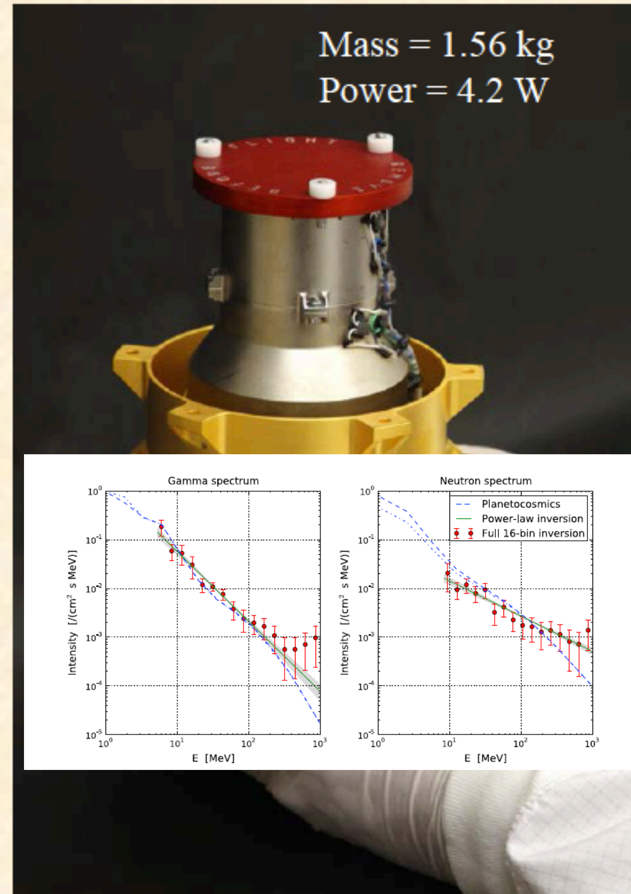
**Special Section:** Results from the first 360 Sols of the Mars Science Laboratory Mission: Bradbury Landing through Yellowknife Bay

**Key Points:**
- We calculated the Martian neutron and gamma spectra
- We compare the results to Planetocosmics simulations

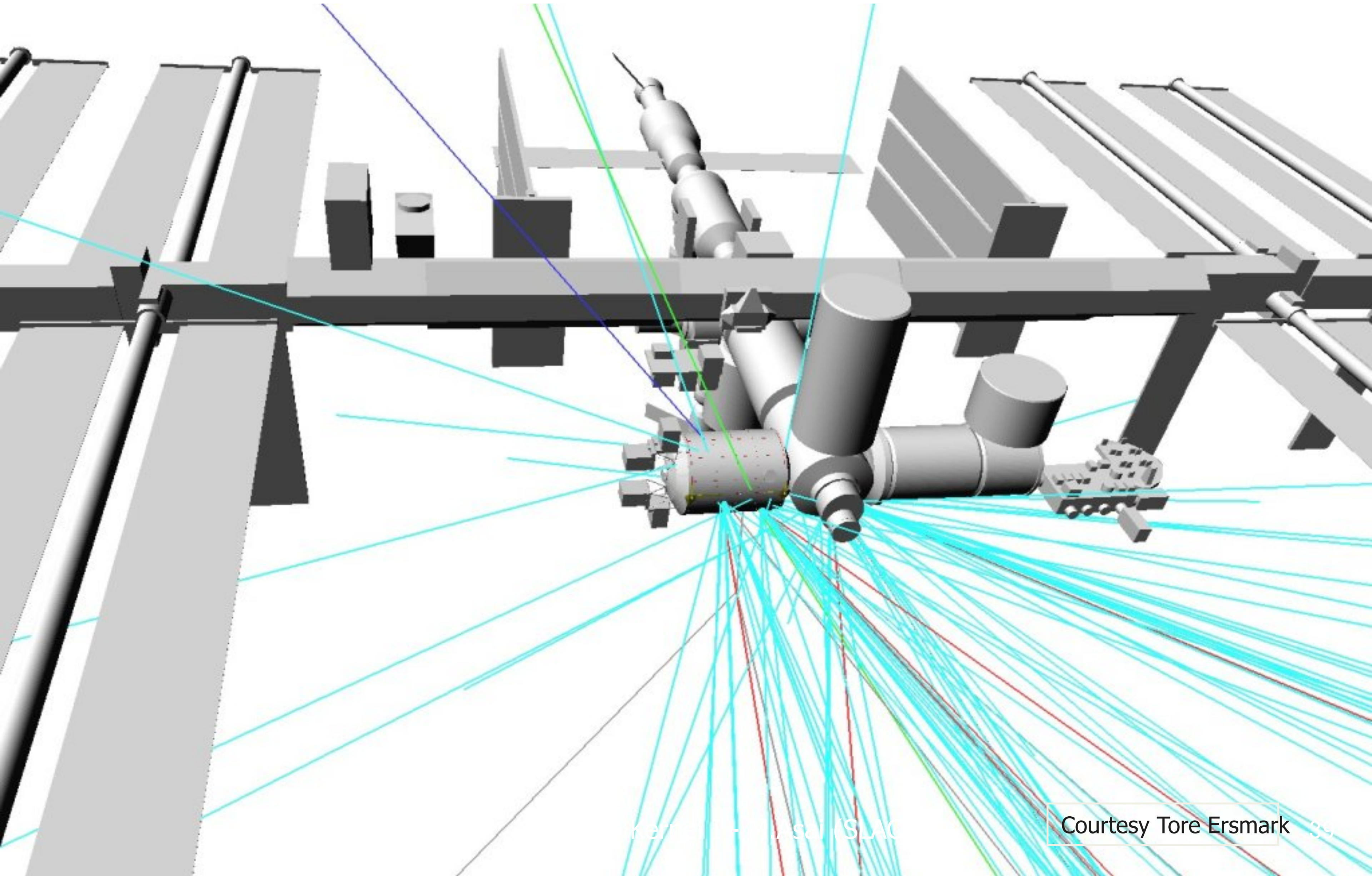
Measurements of the neutron spectrum on the Martian surface with MSL/RAD

J. Köhler[1], C. Zeitlin[2], B. Ehresmann[2], R. F. Wimmer-Schweingruber[1], D. M. Hassler[3], G. Reitz[4], D. E. Brinza[5], G. Weigle[6], J. Appel[1], S. Böttcher[1], E. Böhm[1], S. Burmeister[1], J. Guo[1], C. Martin[1], A. Posner[7], S. Rafkin[3], and O. Kortmann[8]

[1]Institute of Experimental and Applied Physics, Christian-Albrechts-University, Kiel, Germany, [2]Earth, Oceans and Space Department, Southwest Research Institute, Durham, New Hampshire, USA, [3]Space Science and Engineering Division, Southwest Research Institute, Boulder, Colorado, USA, [4]Aerospace Medicine, Deutsches Zentrum für Luft- und Raumfahrt, Cologne, Germany, [5]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, USA, [6]Big Head Endian, LLC, Burden, Kansas, USA, [7]Science Mission Directorate, NASA Headquarters, Washington, District of Columbia, USA, [8]Space Science Laboratory, University of California, Berkeley, California, USA



# International Space Station



Courtesy Tore Ersmark

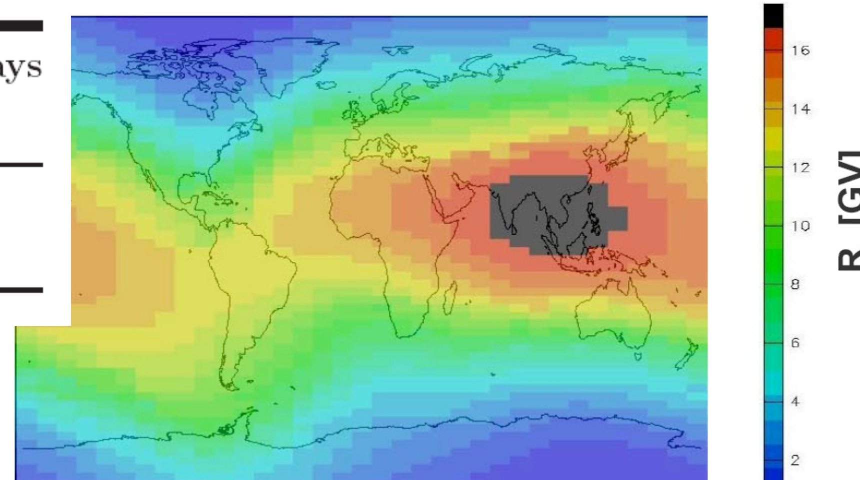
# PlanetoCosmics
# Geant4 simulation of Cosmic Rays in planetary Atmo-/Magneto- spheres

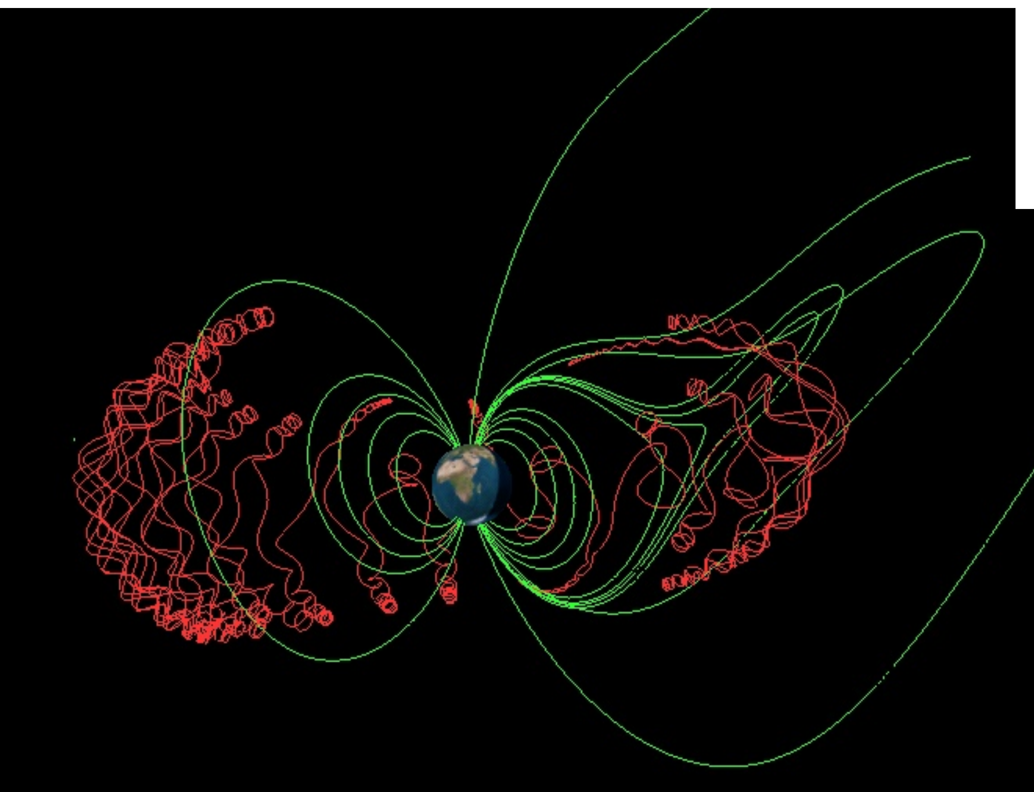
*28th International Cosmic Ray Conference* — 4277

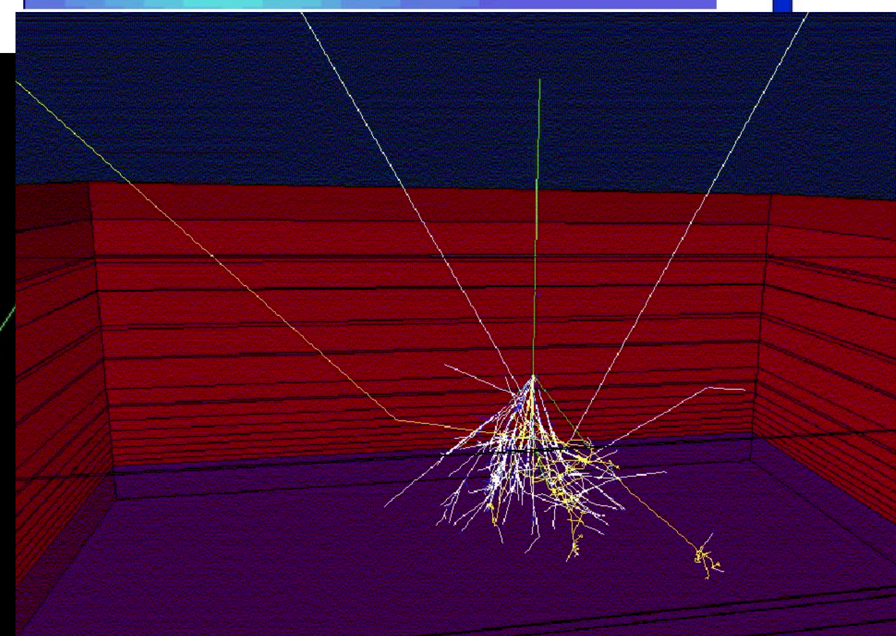
**Geant4 Simulation of the Propagation of Cosmic Rays through the Earth's Atmosphere**

L. Desorgher, E. O. Flückiger, M. R. Moser, and R. Bütikofer
*Physikalisches Institut, University of Bern, CH-3012 Bern, Switzerland*

***Cutoff Rigidities vs position***





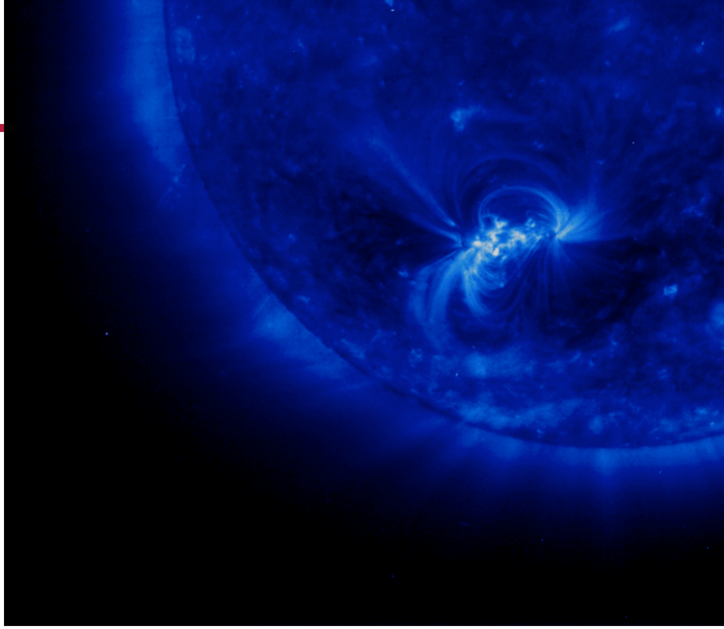
# Solar event gamma-rays

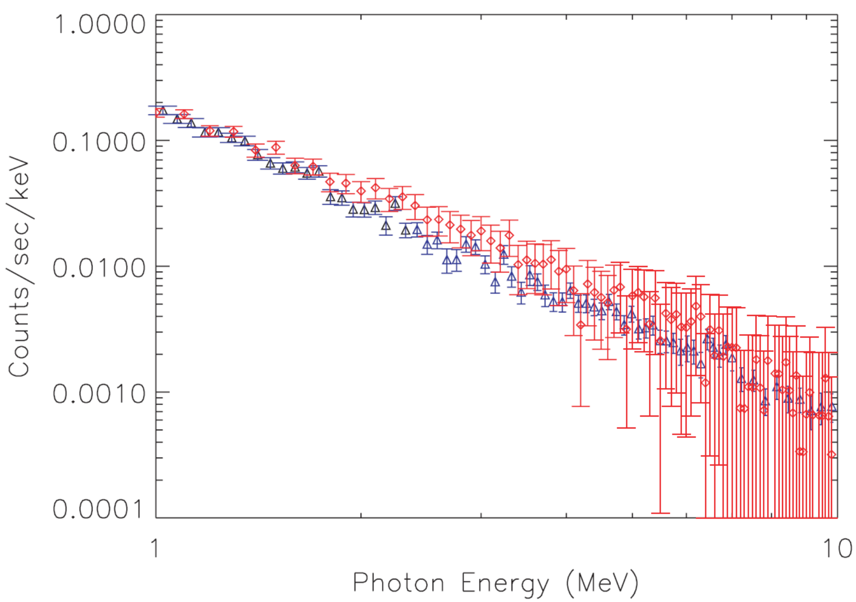
- Electron Bremsstrahlung – induced gammas in solar flares
- Compton back-scattering

  → observable gamma-ray spectrum

  much softer than predicted by simple

  analytic calculations






## Effects of Compton scattering on the Gamma Ray Spectra of Solar flares


Jun'ichi KOTOKU

*National Astronomical Observatory of Japan, 2-21-1 Osawa, Mitaka, Tokyo 181-8588, JAPAN*

*junichi.kotoku@nao.ac.jp*

Kazuo MAKISHIMA[1] and Yukari MATSUMOTO[2]

*Department of Physics, University of Tokyo, Bunkyo-ku, Tokyo, 113-0022*

and

Mitsuhiro KOHAMA, Yukikatsu TERADA and Toru TAMAGAWA

*RIKEN (Institute of Physical and Chemical research), Wako-shi, Saitama*

[1] *Also at RIKEN*

[2] *Present address: Mitsubishi Electric Co., Ltd.*



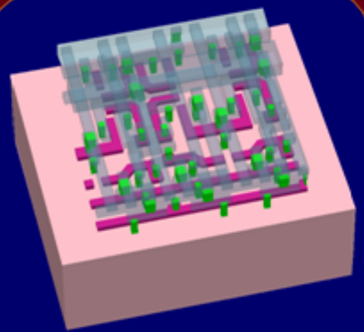
(Received ; accepted )



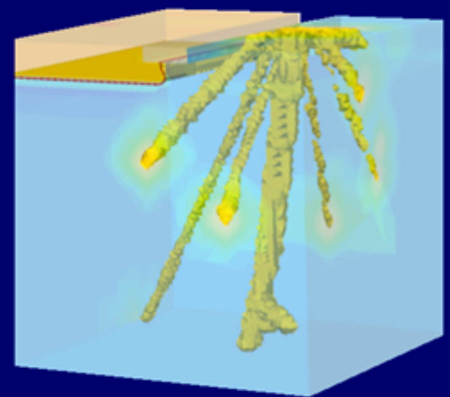
**Abstract**

Using fully relativistic GEANT4 simulation tool kit, the transport of energetic electrons generated in solar flares was Monte-Carlo simulated, and resultant bremsstrahlung gamma-ray spectra were calculated. The solar atmosphere was ap-

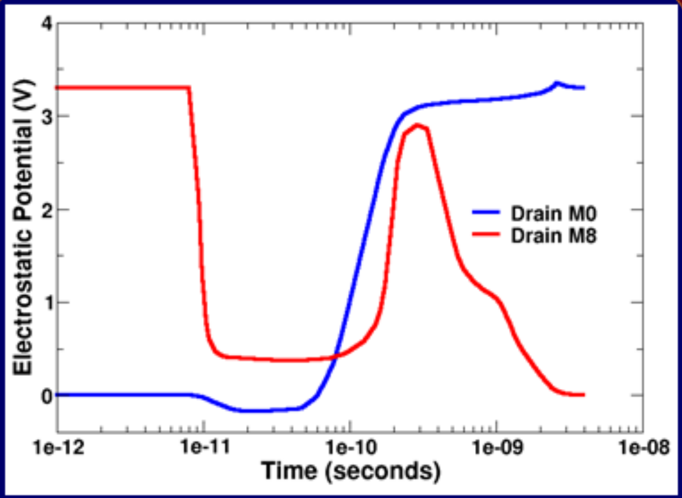
# RADSAFE on SEE in SRAMs



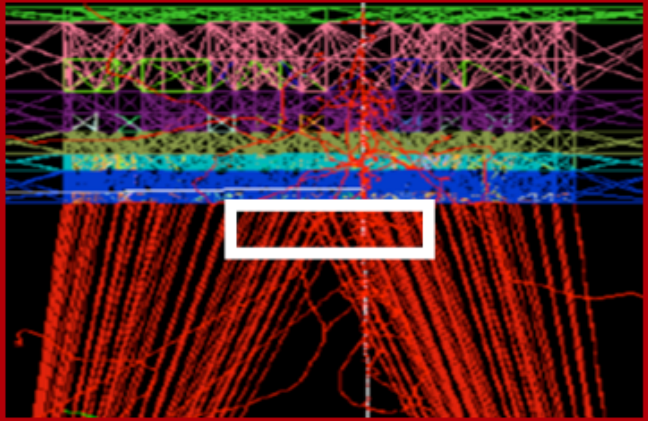
**TCAD Cell Structure: SRAM Cell**



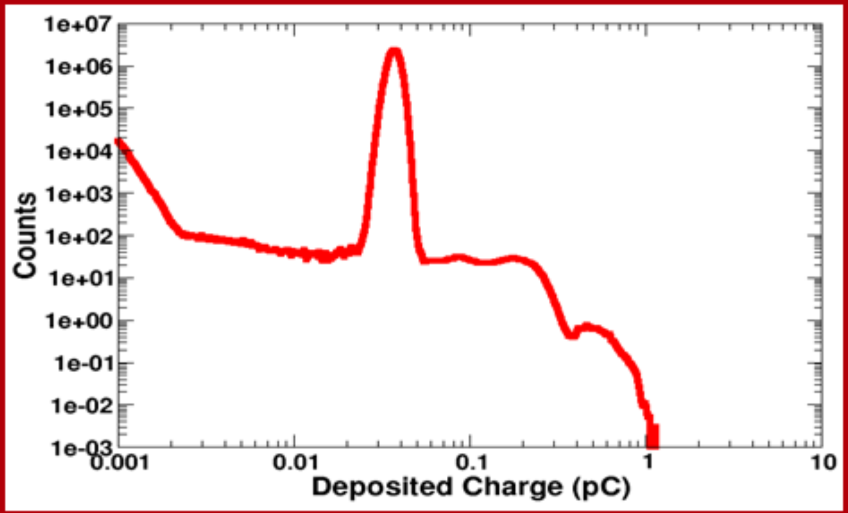
**Single Charge Deposition in TCAD: Ne+W Event**




**SRAM Cell Upset**



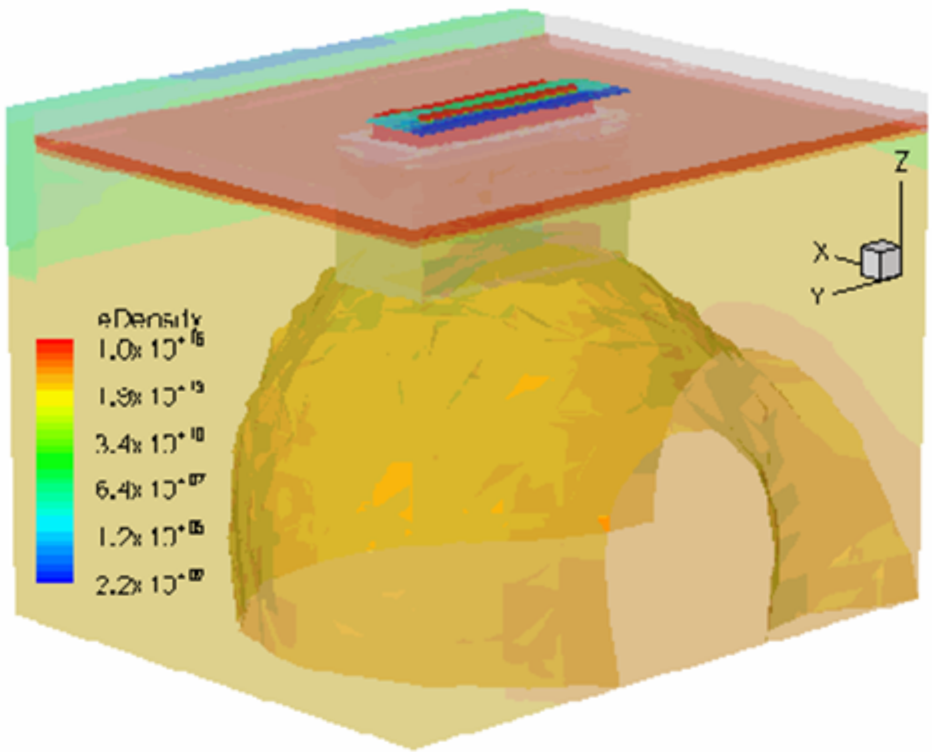
**Geant4 Geometry and 523 MeV Neon Event**

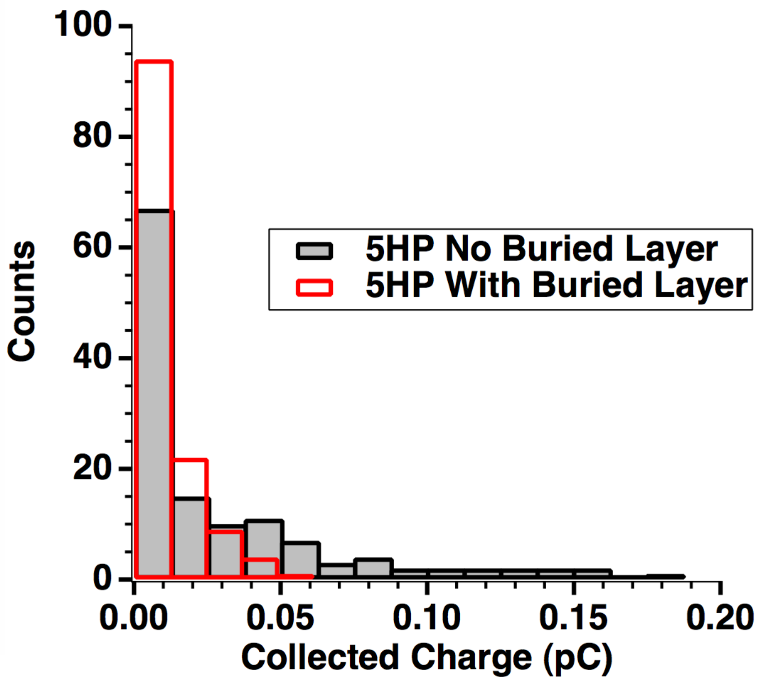



**MRED Energy Deposition for $10^8$ Events**

# Simulation of Radiation Events

- 63-MeV proton incident on a SiGe Heterojunction Bipolar Transistor (HBT)
- Iso-charge surfaces following a nuclear reaction

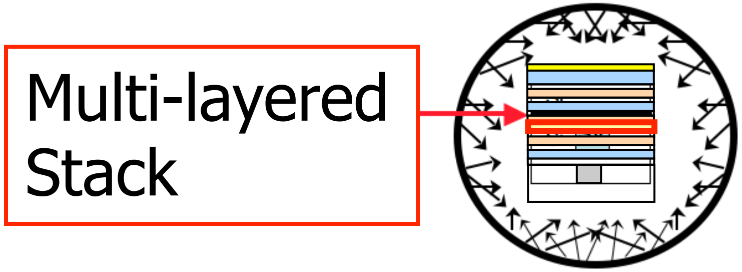


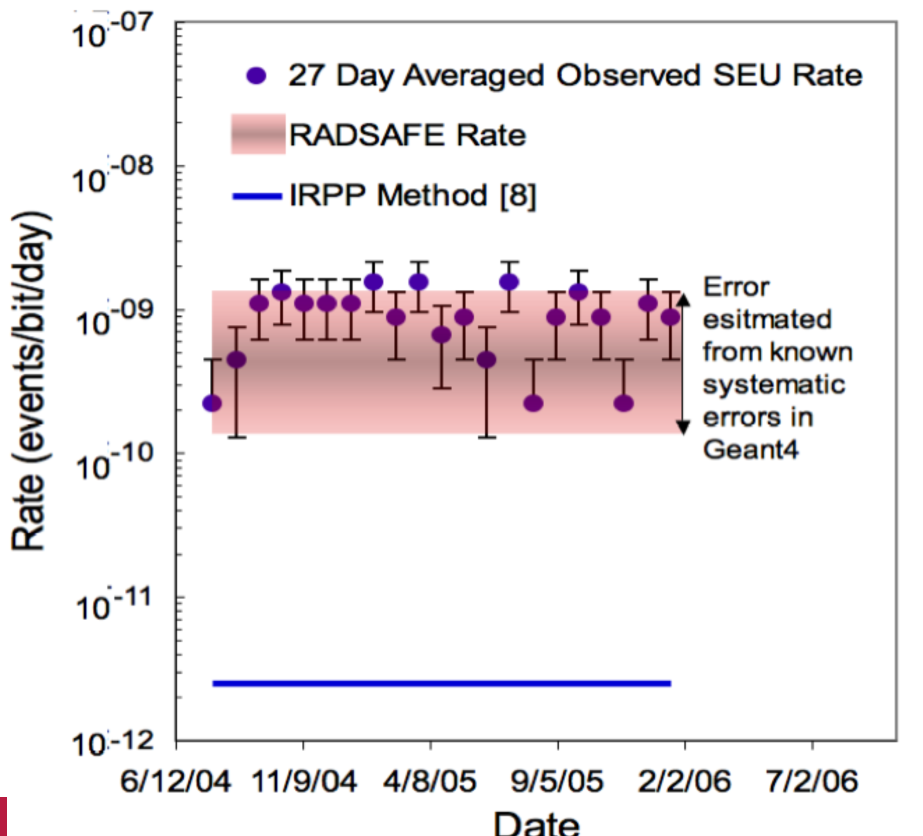




Courtesy of R.Reed (Vanderbilt U.)

# Observed and Predicted SEU Rate for an SRAM

- SRAM used on NASA Messenger spacecraft
- Observed Average SEU Rate:
  - $1 \times 10^{-9}$ Events/Bit/Day
- Vendor predicted rate using CREME96:
  - $2 \times 10^{-12}$ Events/Bit/Day
  - Classical Method nearly a factor 500 lower than observed rate
- MRED rate (includes reaction products):
  - Between $1.3 \times 10^{-10}$ and $1.3 \times 10^{-9}$ Errors/Bit/Day

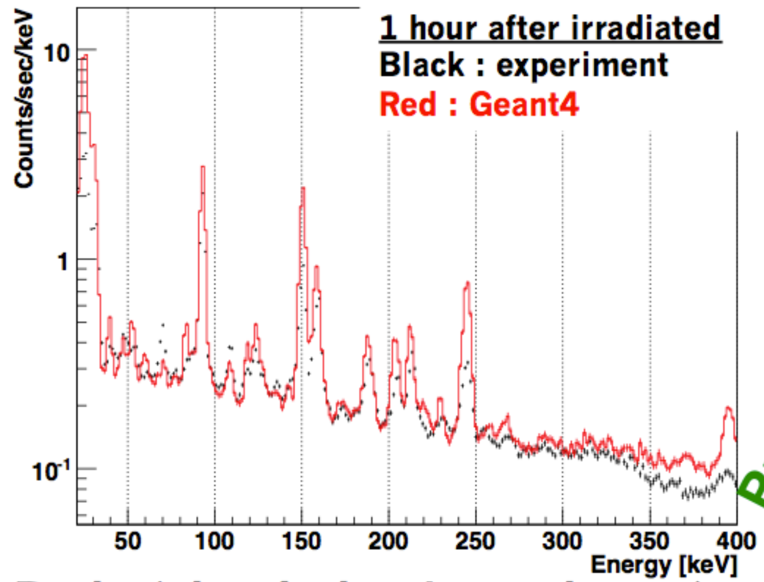


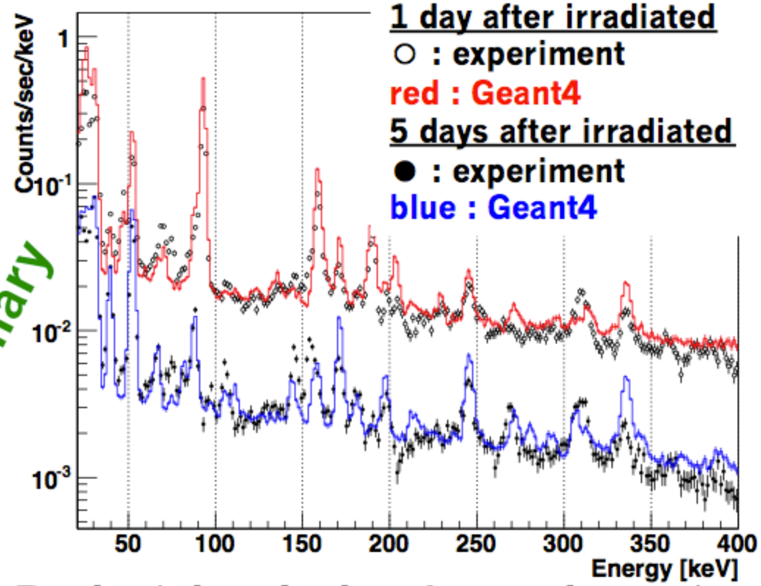

Courtesy of R.Reed (Vanderbilt U.)

# Time evolution of the activation background

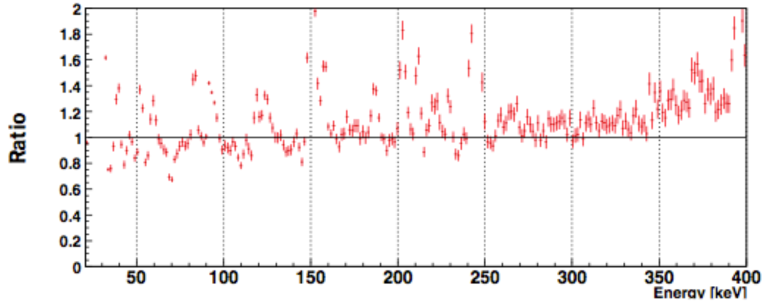
## Comparison with Geant4



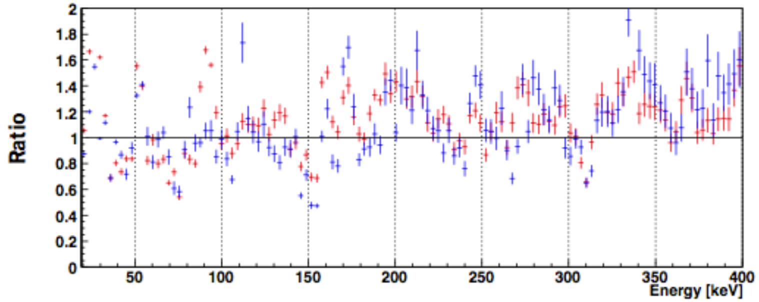

Preliminary




## Ratio (simulation/experiment)



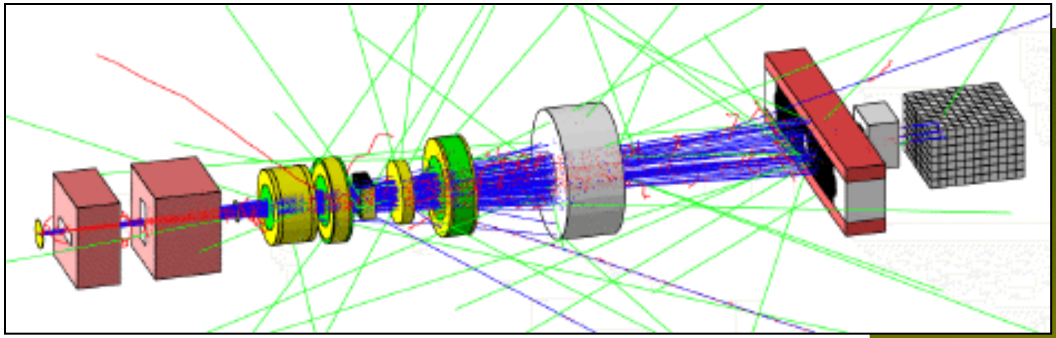

## Ratio (simulation/experiment)

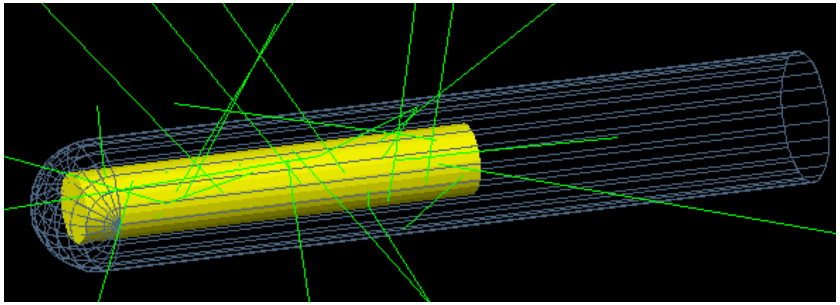


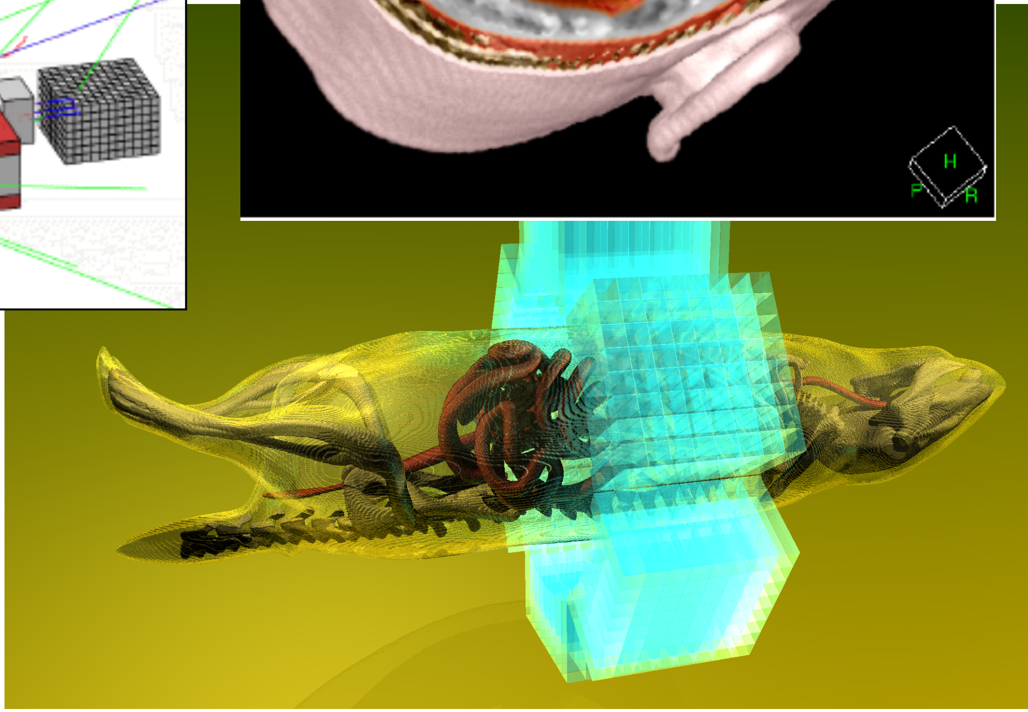

❖ Simulation results agrees with experimental data within a factor of two in terms of the line intensities

# Geant4 @ Medical Science

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

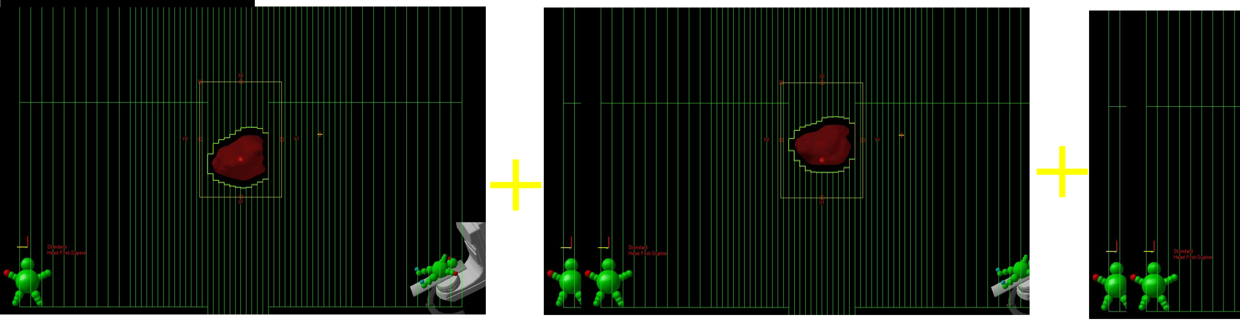



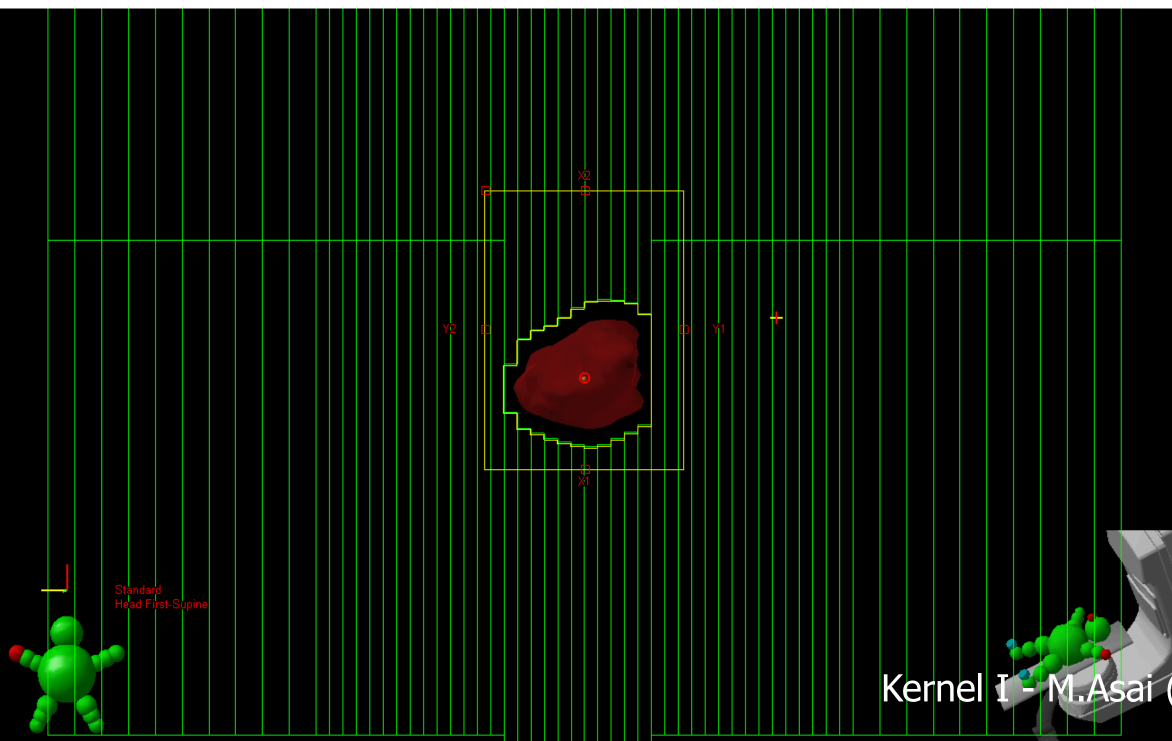


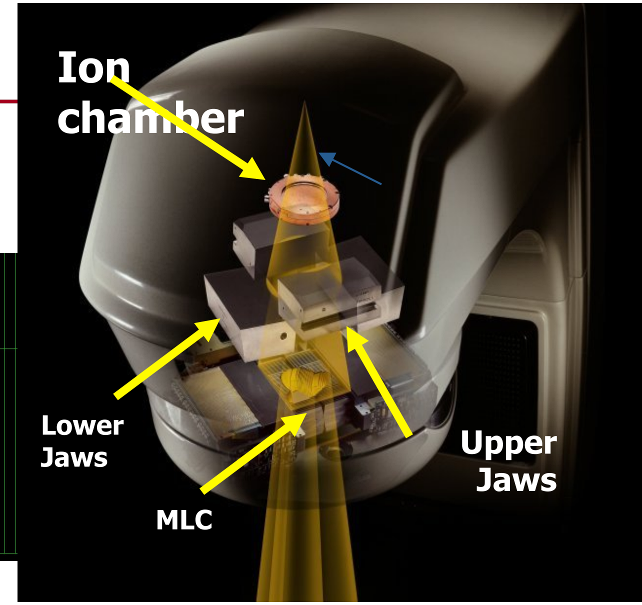


# 4D RT Treatment Plan

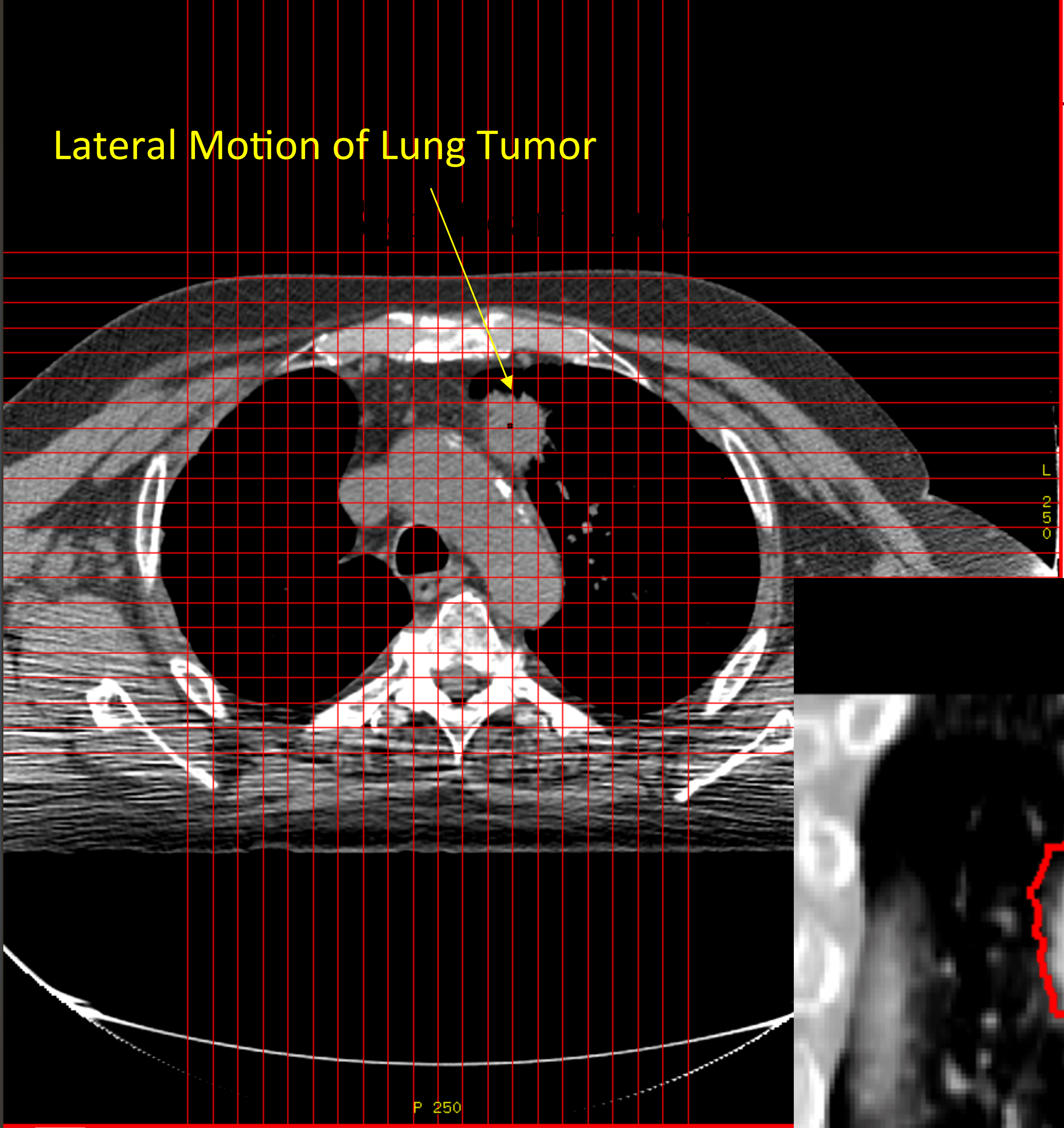
Source: Lei Xing, Stanford University

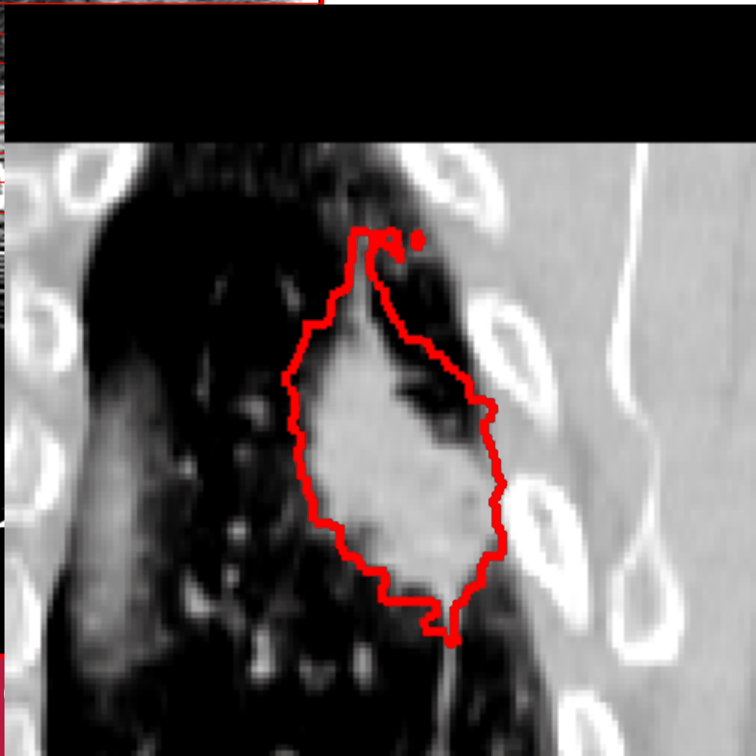




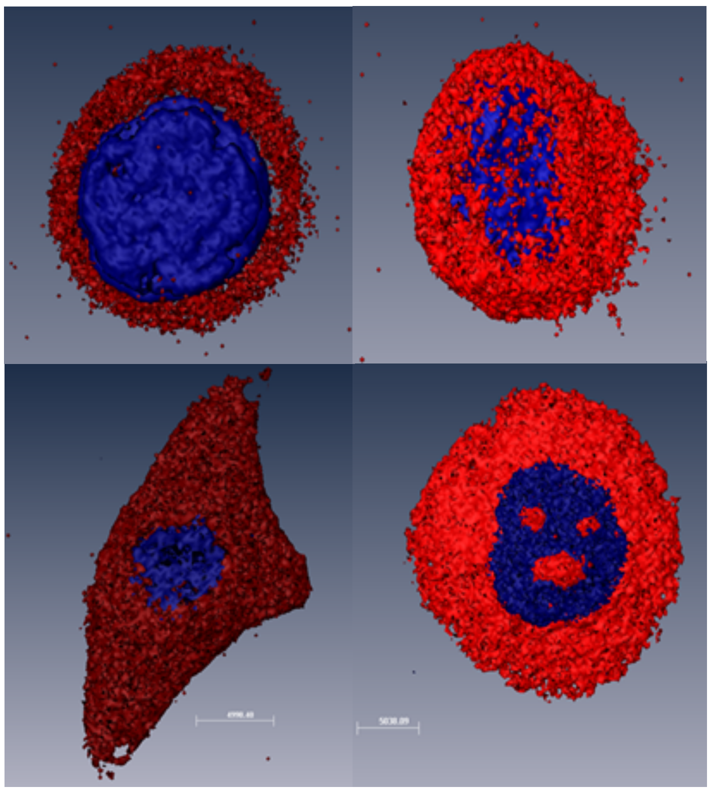

Y. Yang, S. Huq, L Xing, Med. Phys, 2006

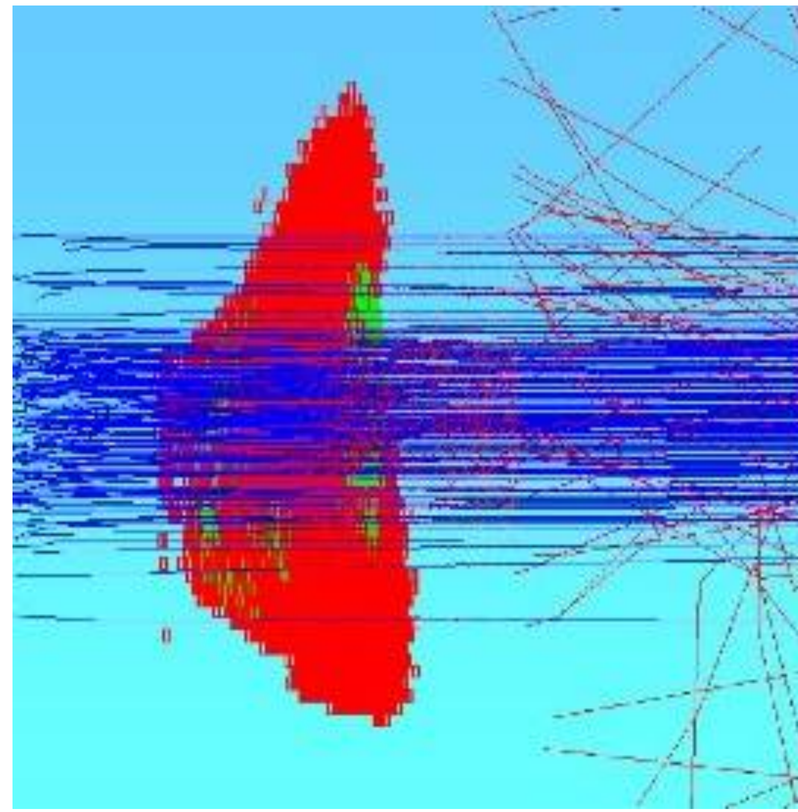

Lateral Motion of Lung Tumor
L
250
P 250



# Single cell irradiation

- Example of single cell irradiation by 3 MeV alpha particles in a high-resolution cellular phantom
  - 4h or 24h incubated cell
  - 64 x 64 x 60 resolution
  - 0.36 x 0.36 x 0.16 $\mu m^3$ voxel size
- Full CENBG microbeam irradiation setup simulated

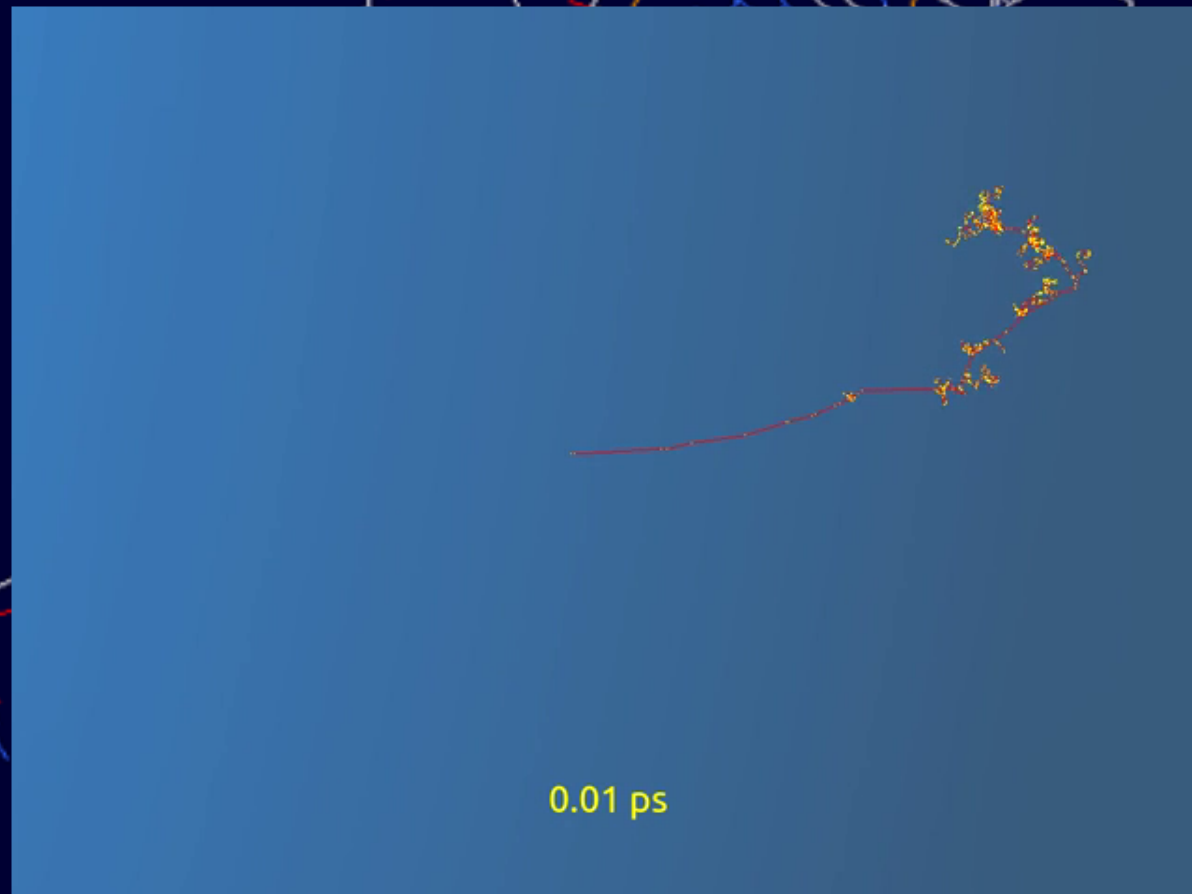




Courtesy of Sebasien Incerti (IN2P3-CNRS / CENBG)

# DNA in Geant4

most common DNA conformation in cells

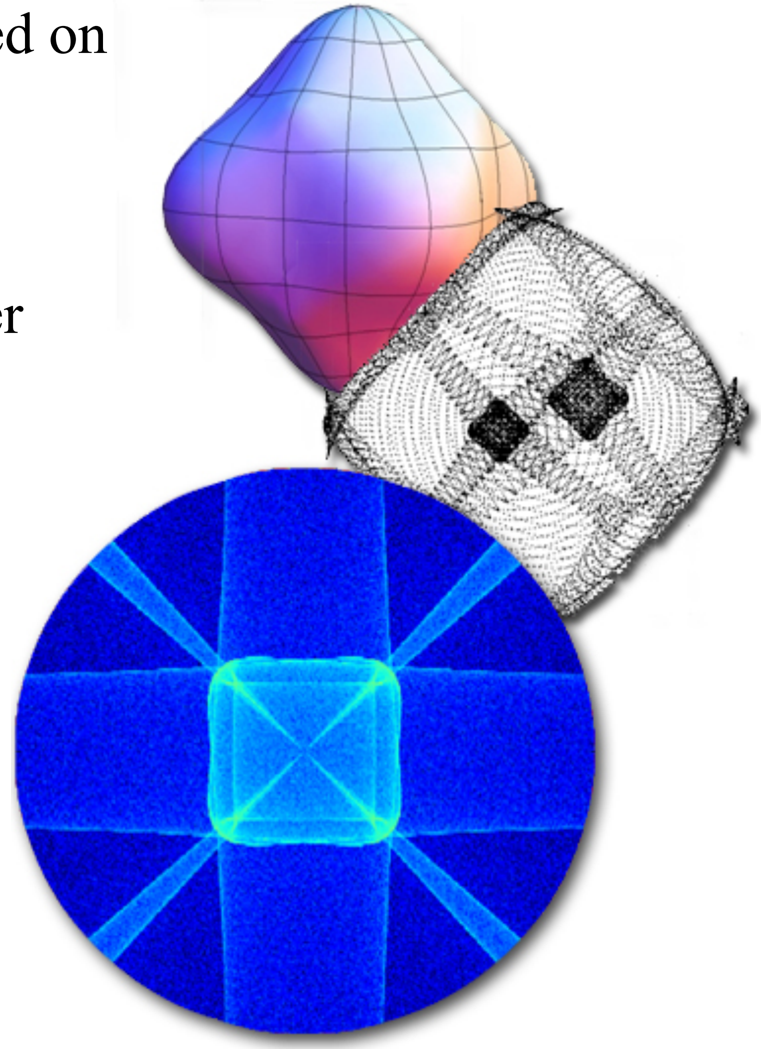



**Conversion of ProteinDataBank-format file ▸ Geant4**

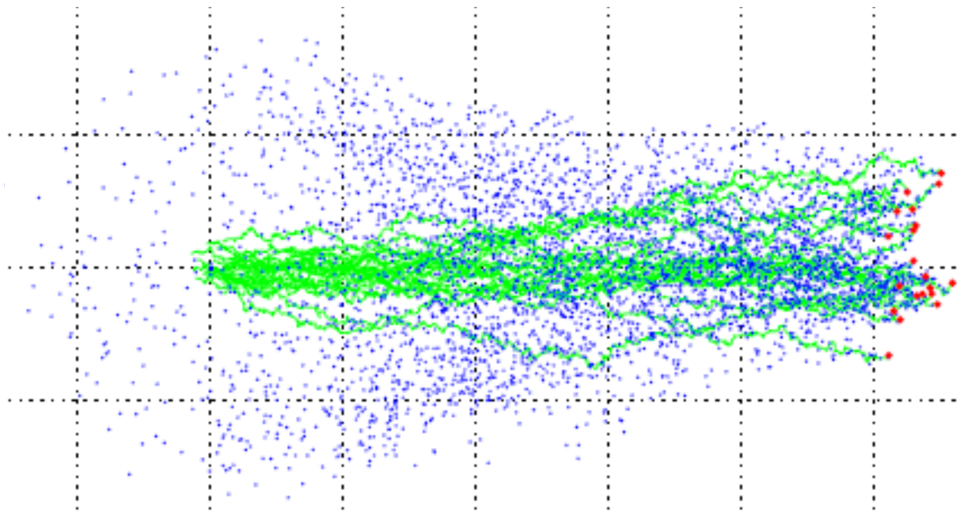
Courtesy of Sebasien Incerti (IN2P3-CNRS / CENBG)

# Condensed Matter Physics in Geant4

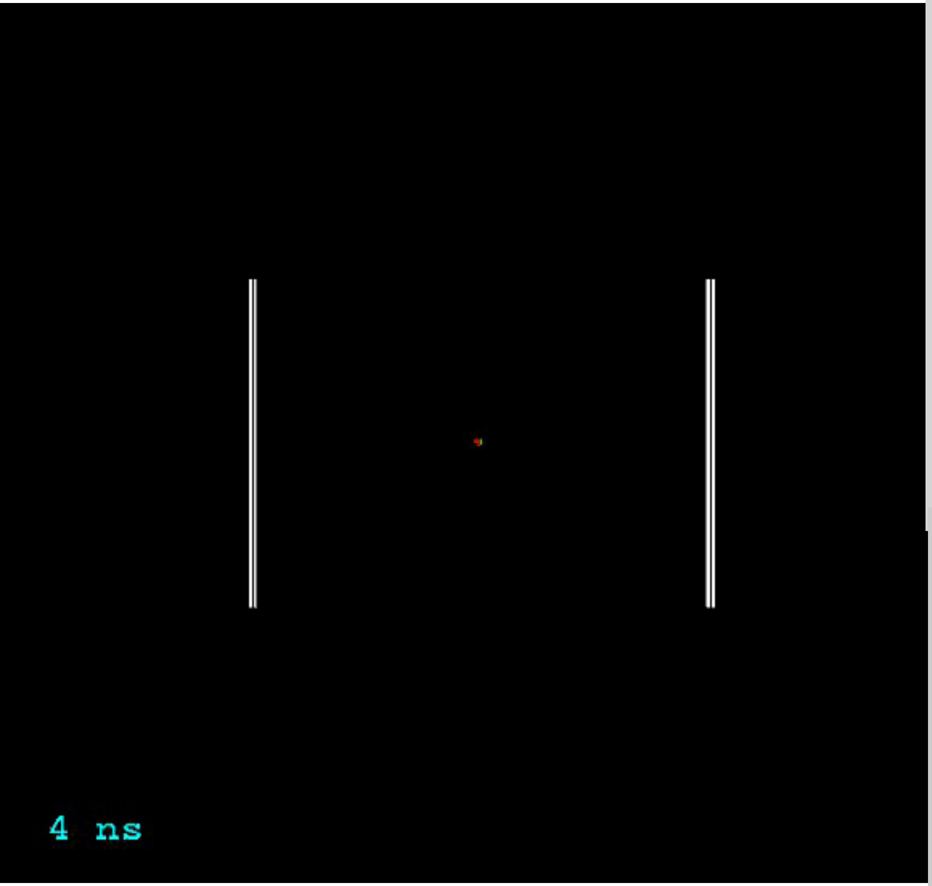
- Phonon propagation, including focusing based on elasticity tensor (right)

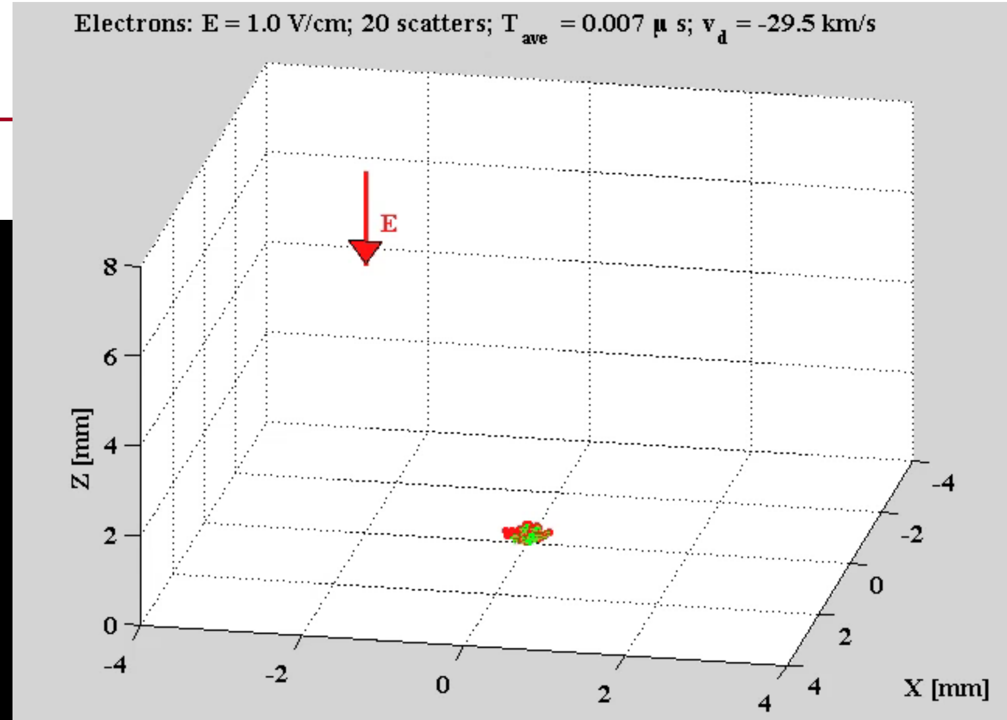


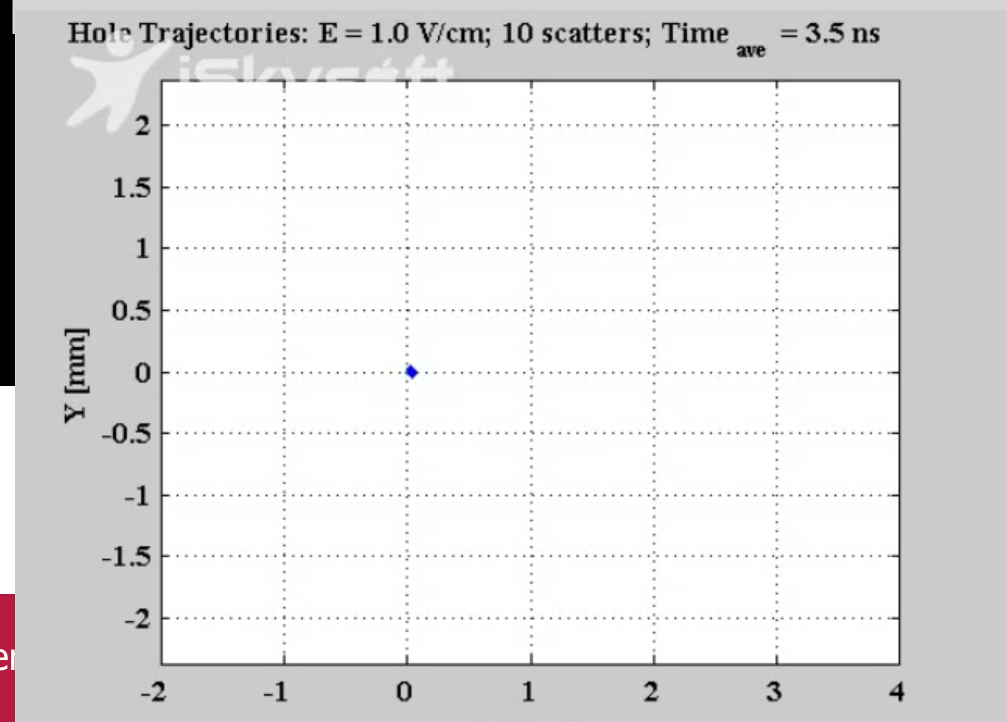
- e-/h+ transport, including conduction band anisotropy and Luke-Neganov emission, under development (below)



# e-/h propagation with Luke phonon emission in Ge crystal

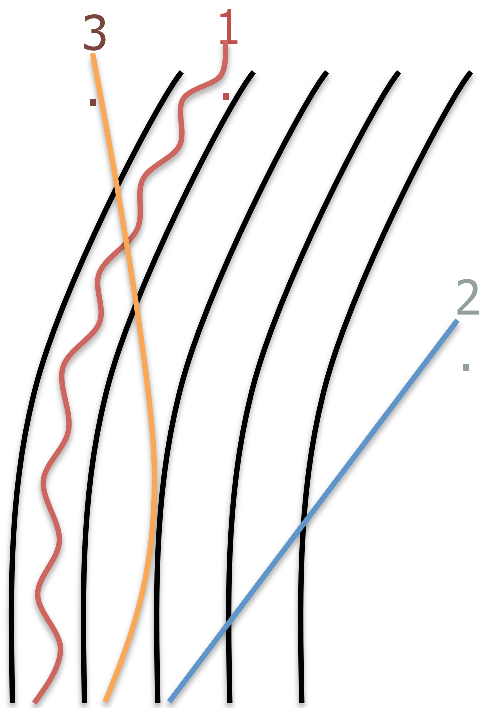


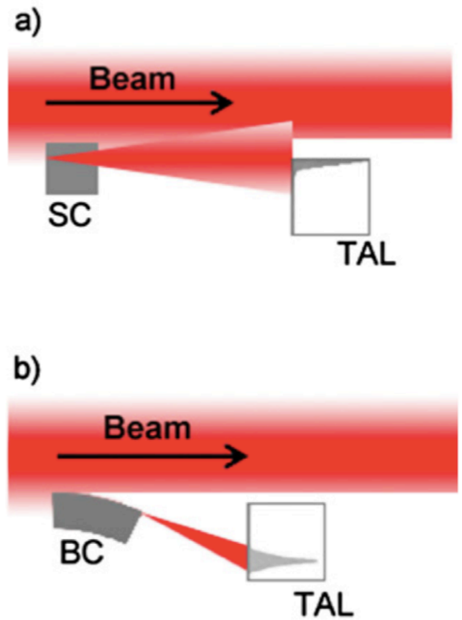



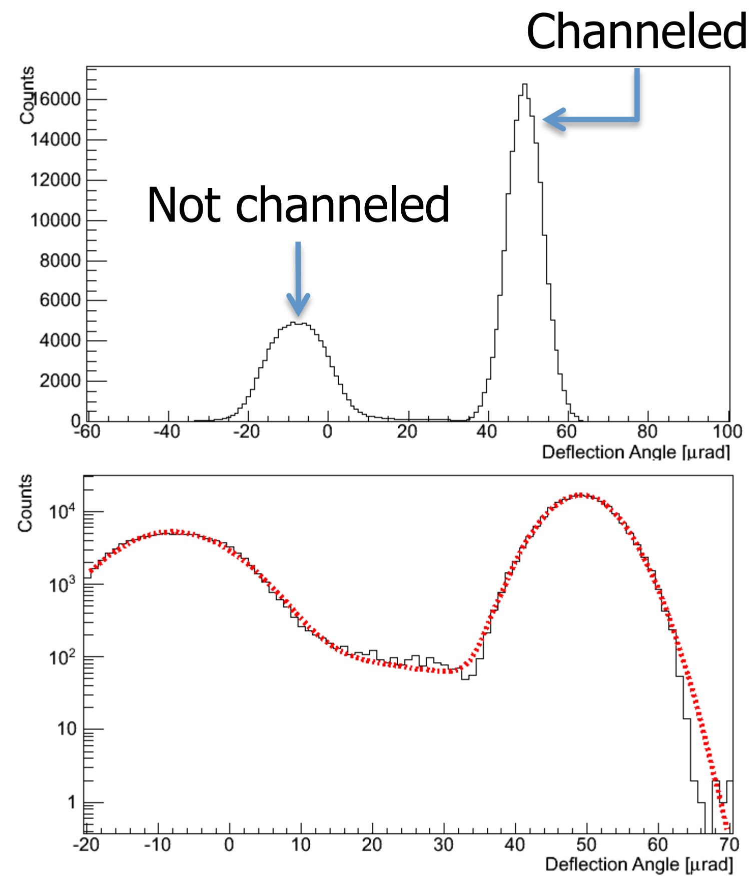



# Bent crystal as a collimator

- Bent crystal can be used as a collimator to deflect particles of beam halo.
- This study will be extended for T-513 experiment at SLAC LCLS ESTB



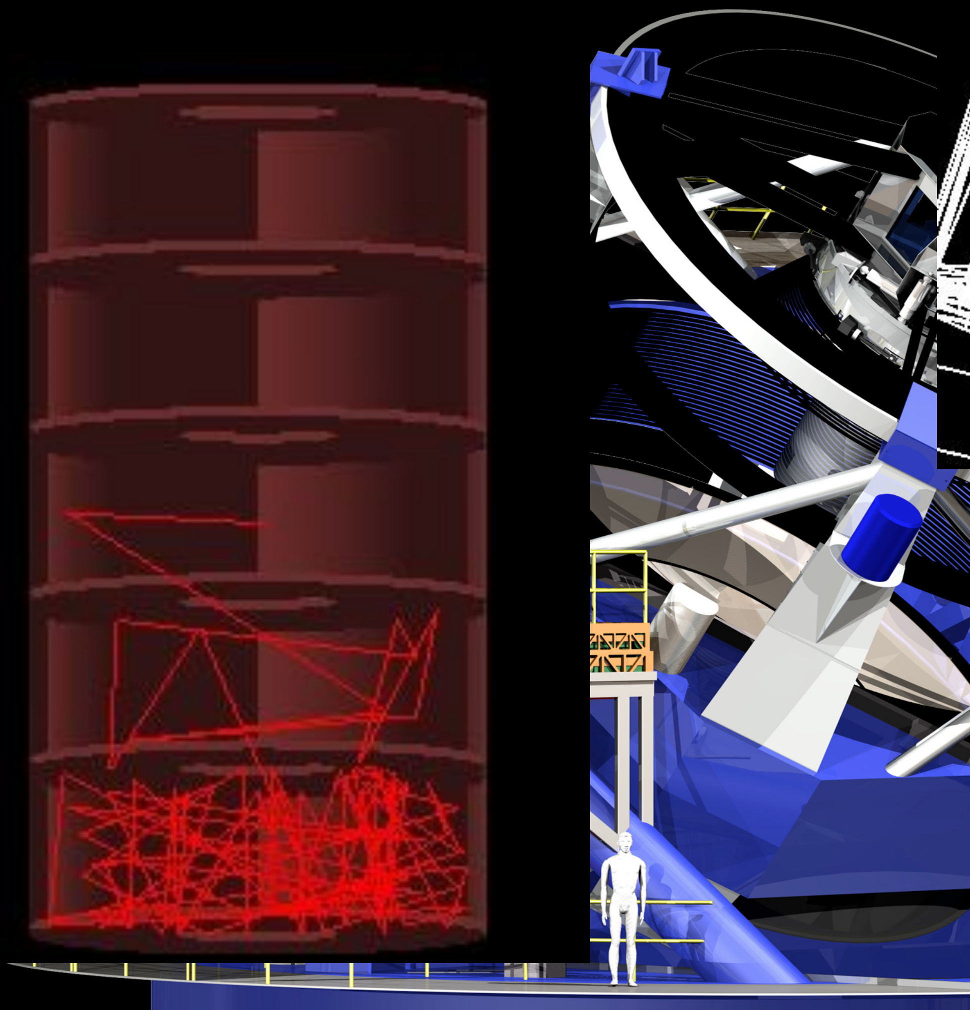



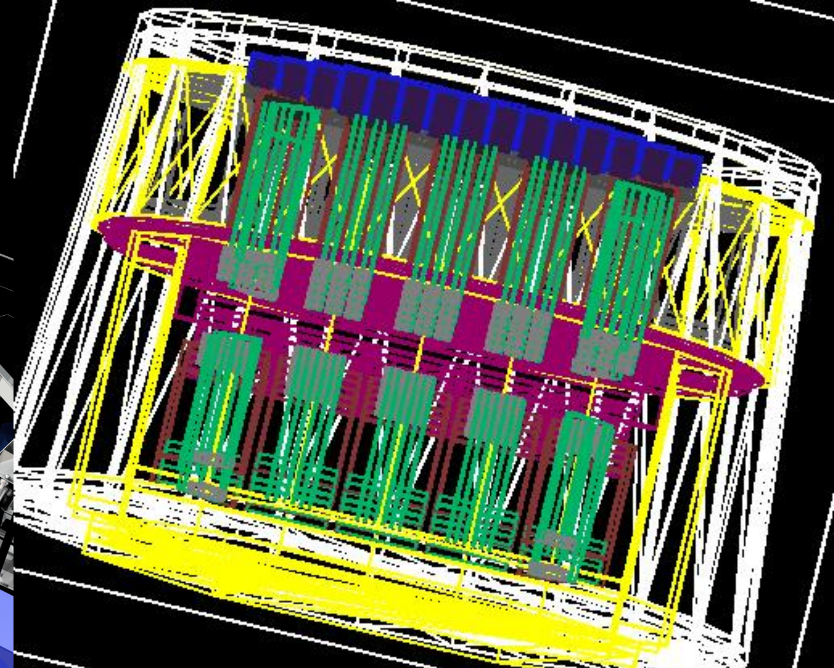



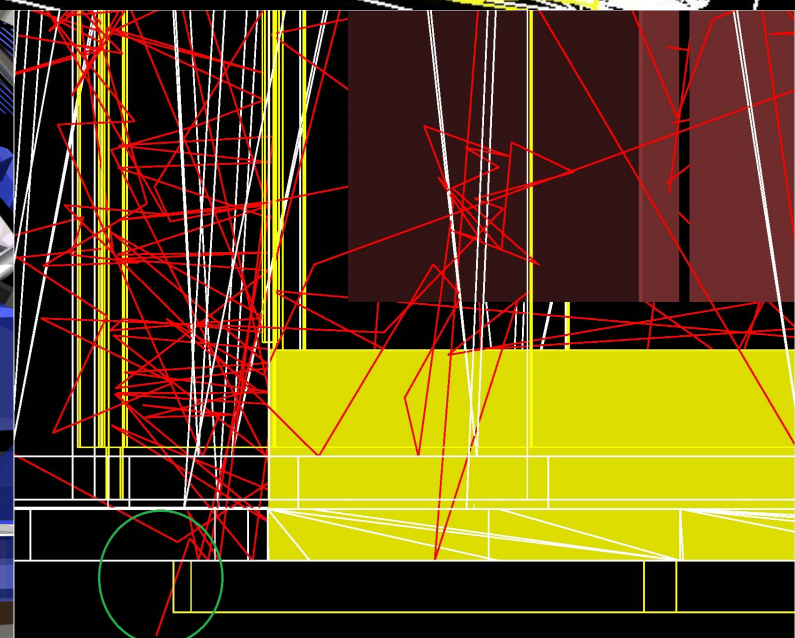

Enrico Bagli (INFN/Ferrara)

- W. Scandale et al., Phys. Lett. B 680 (2009) 129

# LSST (Large Synoptic Survey Telescope)



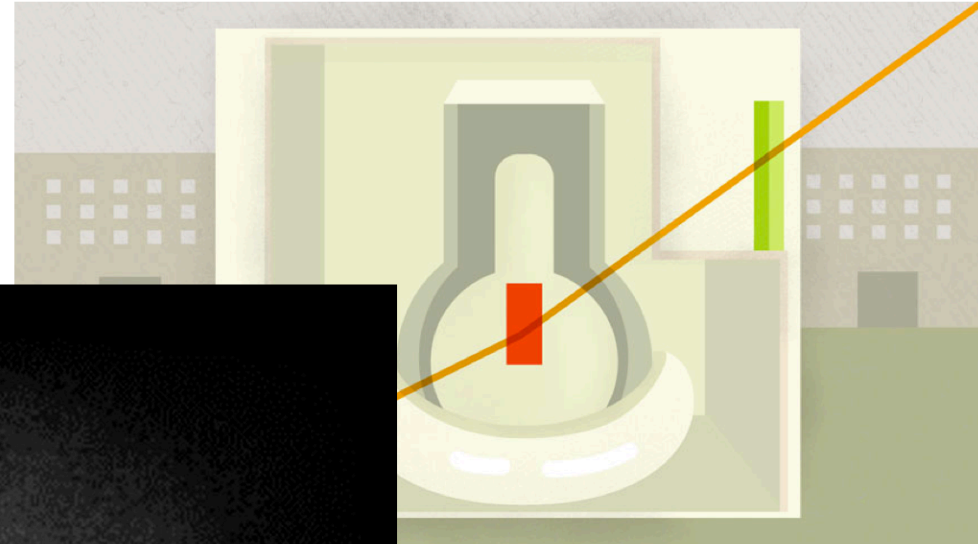




"Geant4 Applications for Modeling Molecular Transport in Complex Vacuum Geometries." J.Singal, J. Langton, R. Schindler, Int J Mod Sim Sci Comp, in press (arXiv:1302.2963) symmetry of particle physics

Those exterior walls, made of concrete 10 feet thick, offer their own challenge. Based on computer simulations run with the particle physics software GEANT4, the walls are expected to reduce the resolution to about 30 centimeters.

In addition, the team must also prepare for the high radiation levels present just outside of the reactor units.



**…ors (shown here in green) on either side of … record the path of muons (represented by the … rough the reactor. By determining how the … ctors, scientists will compile the first picture of**

*… with Shawna X.*

As time ticks down to the restart of the Large Hadron Collider, scientists are making sure their detectors run like clockwork.

age

ELSEVIER

Journal of Environmental Radioactivity

Volumes 162–163, October 2016, Pages 118–128

# Evaluating remediation of radionuclide contaminated forest near Iwaki, Japan, using radiometric methods

D.C.W. Sanderson[a], A.J. Cresswell[a], K. Tamura[b], T. Iwasaka[c], K. Matsuzaki[d]

[a] Scottish Universities Environmental Research Centre, East Kilbride, Glasgow G75 0QF, United Kingdom

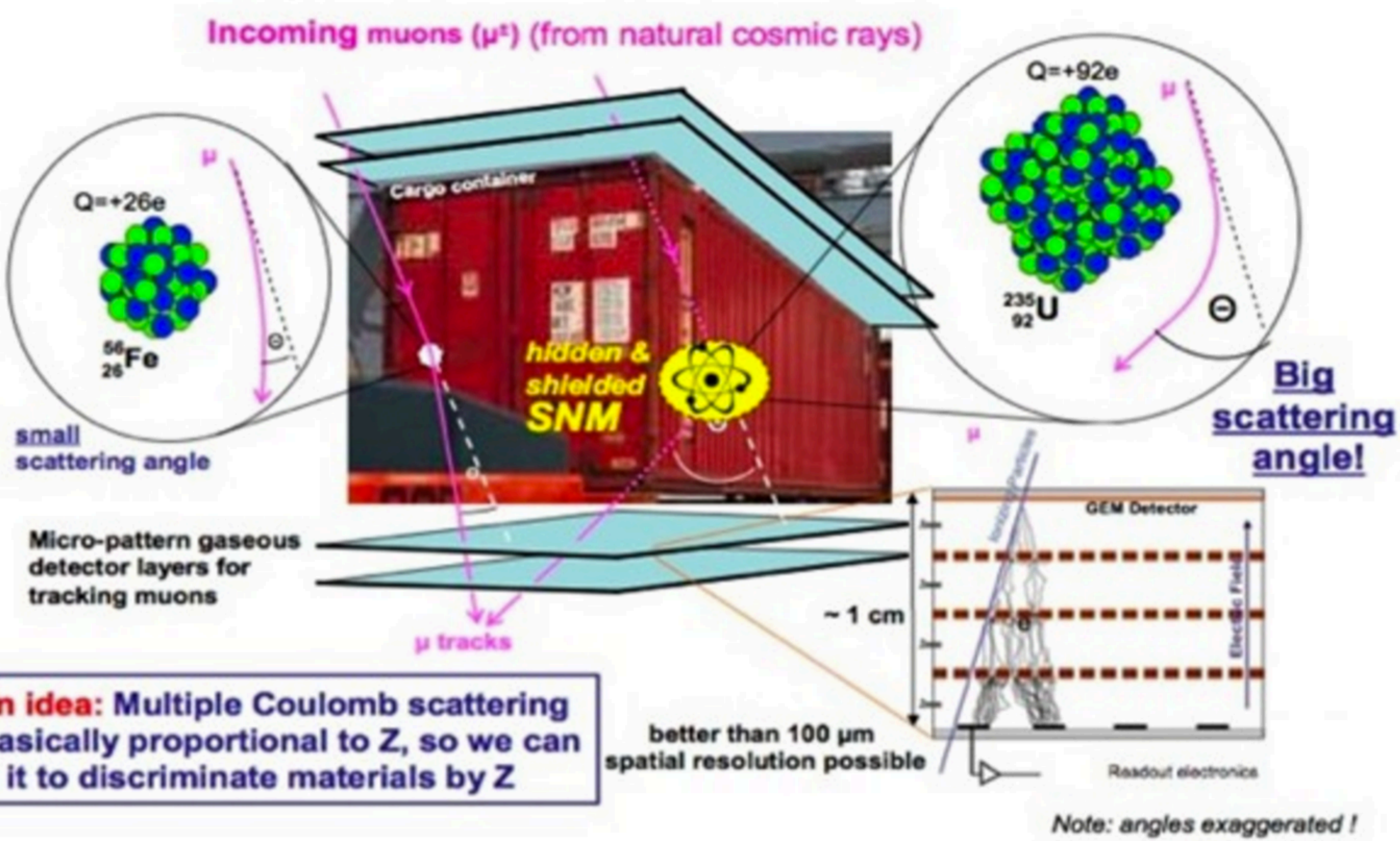
[b] Faculty of Life and Environmental Sciences, University of Tsukuba, Japan

[c] Miraishiko Inc., Kanegaya, Asahi-ku, Yokohama, Japan

[d] Yunodakesansonai, Iwaki, Japan

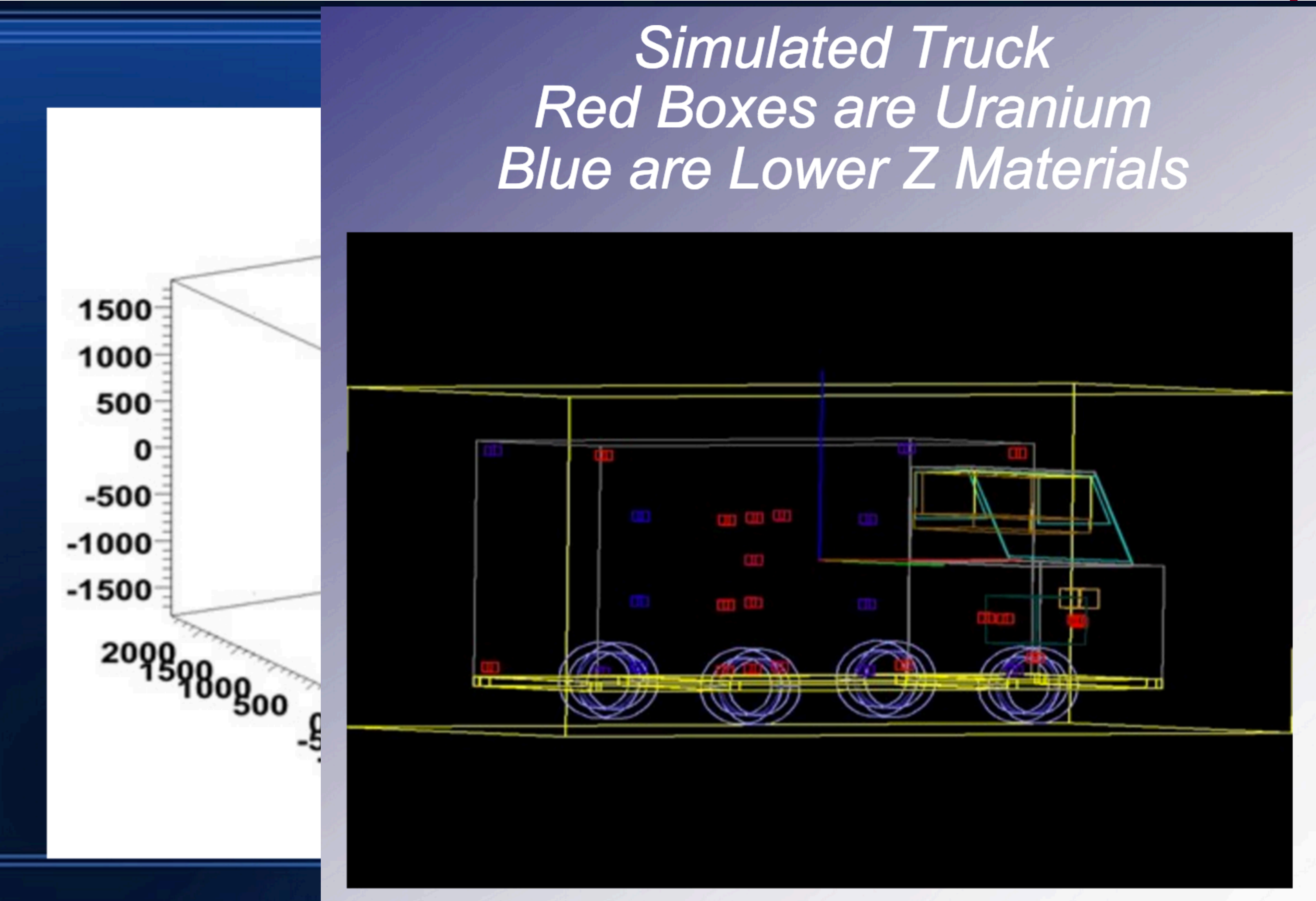
Received 24 December 2015, Revised 10 May 2016, Accepted 15 May 2016, Available online 24 May 2016

# Muon tomography for nuclear threat detection




**Main idea:** Multiple Coulomb scattering is basically proportional to Z, so we can use it to discriminate materials by Z

# Muon tomography for nuclear threat detection



# Simulating x-ray cargo radiography

MENU

SCIENTIFIC REPORTS

SLAC



Article | OPEN

# Uncovering Special Nuclear Materials by Low-energy Nuclear Reac

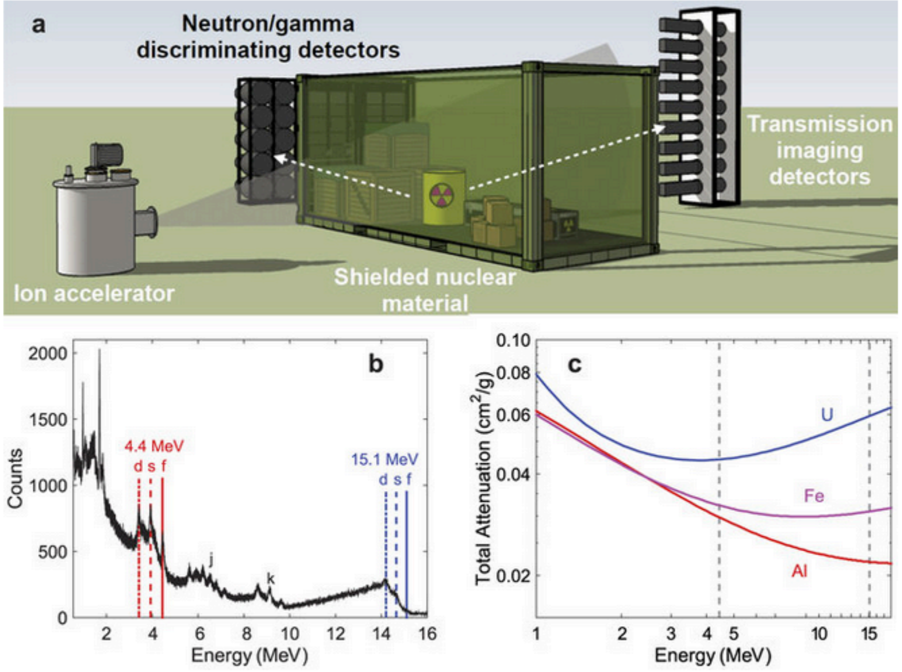
P. B. Rose, A. S. Erickson, M. Mayer, J. Nattress & I. Jovano

*Scientific Reports* **6**, Article number: 24388 (2016)
doi:10.1038/srep24388
Download Citation

Receiv
Accep
Publis

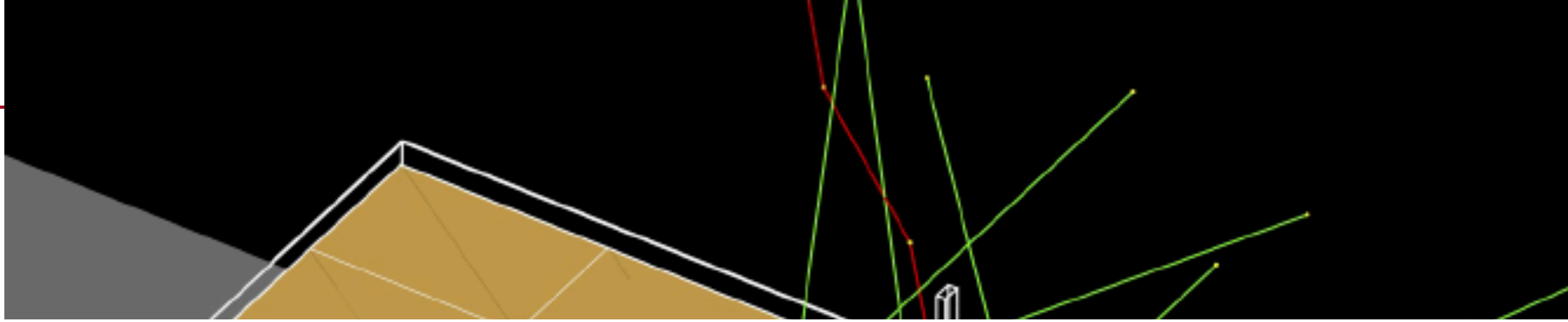
Applied physics | Imaging techniques

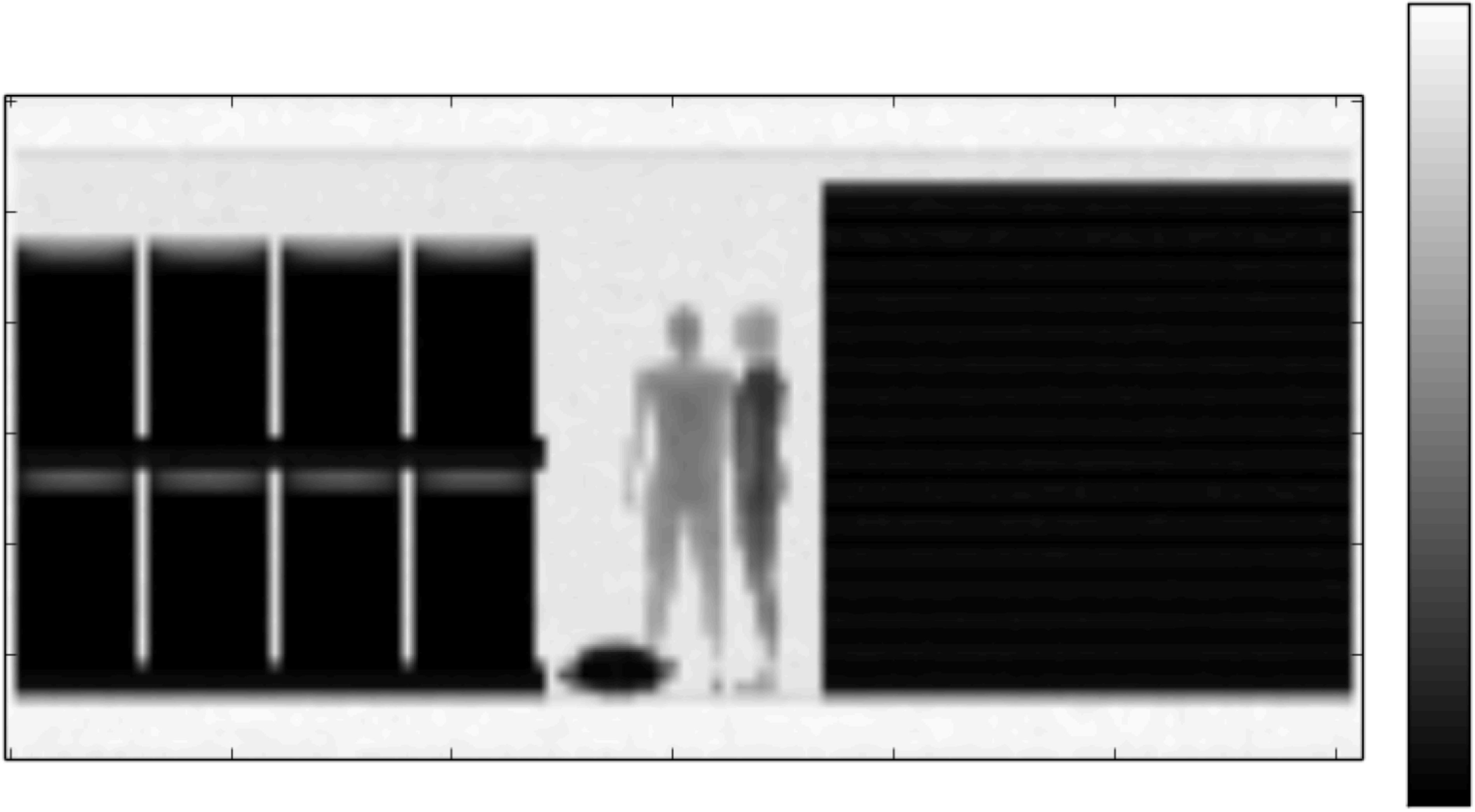
**Figure 1: Illustration of the imaging method using a low-energy nuclear reaction radiation source.**




SLAC

Kerne





Version 10.4-p02

# Geant4 license

NATIONAL ACCELERATOR LABORATORY

U.S. DEPARTMENT OF ENERGY

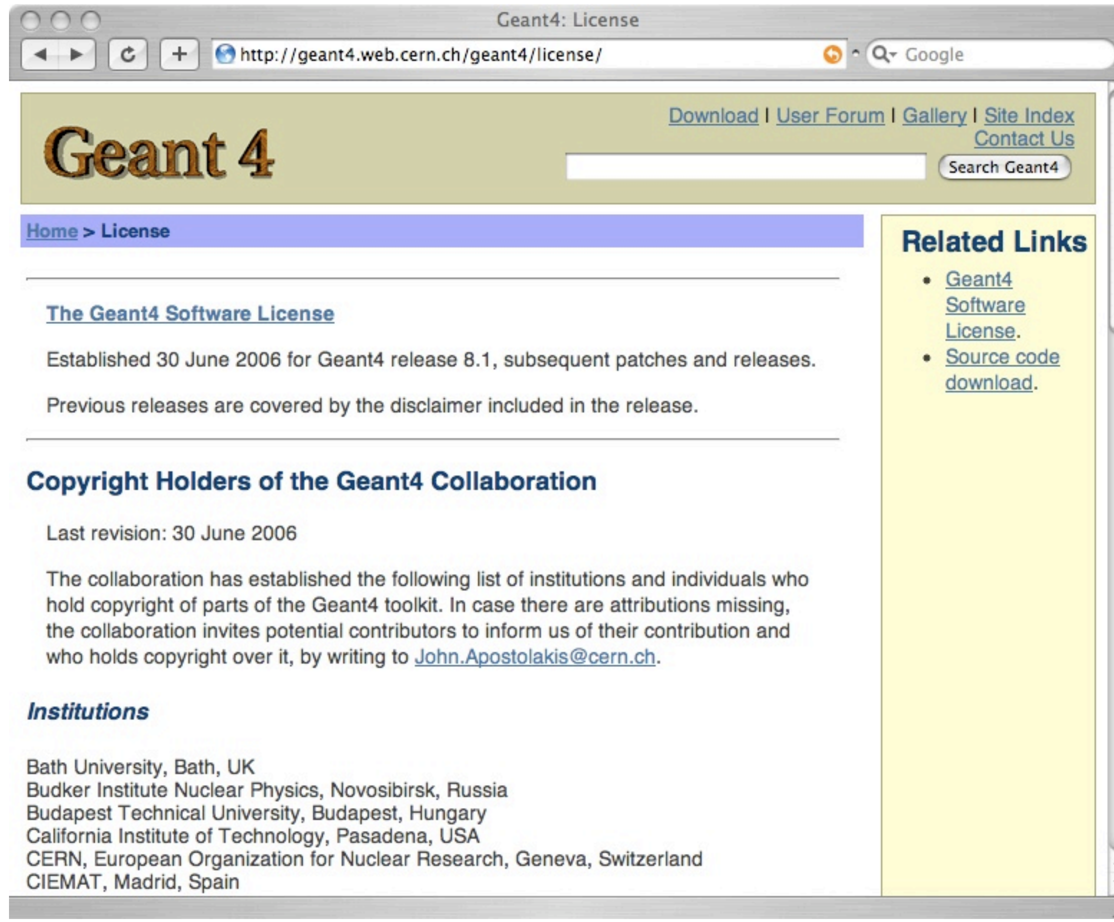
Office of Science

# The New Geant4 License

In response to user requests for clarification of Geant4's distribution policy, the collaboration recently announced a new license.

- Makes clear the user's wide-ranging freedom to use, extend or redistribute Geant4, even as part of some for-profit venture.
- The license was released along with the latest Geant4 release 8.1.
- Simple enough that you can read and understand it.

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



Geant4: License

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

Geant 4

Download | User Forum | Gallery | Site Index
Contact Us

Search Geant4

Home > License

**The Geant4 Software License**

Established 30 June 2006 for Geant4 release 8.1, subsequent patches and releases.

Previous releases are covered by the disclaimer included in the release.

**Copyright Holders of the Geant4 Collaboration**

Last revision: 30 June 2006

The collaboration has established the following list of institutions and individuals who hold copyright of parts of the Geant4 toolkit. In case there are attributions missing, the collaboration invites potential contributors to inform us of their contribution and who holds copyright over it, by writing to John.Apostolakis@cern.ch.

***Institutions***

Bath University, Bath, UK
Budker Institute Nuclear Physics, Novosibirsk, Russia
Budapest Technical University, Budapest, Hungary
California Institute of Technology, Pasadena, USA
CERN, European Organization for Nuclear Research, Geneva, Switzerland
CIEMAT, Madrid, Spain

**Related Links**

- Geant4 Software License.
- Source code download.

# The Geant4 License

License has 8 points.  The points are written clearly and simply.

1,2 and 3) Tell the world who the software came from, and don't claim you are us.

Installation, use, reproduction, display, modification and redistribution of this software, with or without modification, in source and binary forms, are permitted on a non- exclusive basis. Any exercise of rights by you under this license is subject to the following conditions:

1. Redistributions of this software, in whole or in part, with or without modification, must reproduce the above copyright notice and these license conditions in this software, the user documentation and any other materials provided with the redistributed software.

2. The user documentation, if any, included with a redistribution, must include the following notice: "This product includes software developed by Members of the Geant4 Collaboration ( http://cern.ch/geant4 )."

If that is where third-party acknowledgments normally appear, this acknowledgment must be reproduced in the modified version of this software itself.

3. The names "Geant4" and "The Geant4 toolkit" may not be used to endorse or promote software, or products derived therefrom, except with prior written permission by license@geant4.org. If this software is redistributed in modified form, the name and reference of the modified version must be clearly distinguishable from that of this software.

# The Geant4 License

4) If you choose to give it away free to everyone, we can have it for free too.

5) You can't patent the parts we did.

4. You are under no obligation to provide anyone with any modifications of this software that you may develop, including but not limited to bug fixes, patches, upgrades or other enhancements or derivatives of the features, functionality or performance of this software. However, if you publish or distribute your modifications without contemporaneously requiring users to enter into a separate written license agreement, then you are deemed to have granted all Members and all Copyright Holders of the Geant4 Collaboration a license to your modifications, including modifications protected by any patent owned by you, under the conditions of this license.

5. You may not include this software in whole or in part in any patent or patent application in respect of any modification of this software developed by you.

# The Geant4 License

We don't claim that it works, and we're not responsible if it doesn't.

Version 10.4-p02

# Basic concepts and kernel structure

NATIONAL ACCELERATOR LABORATORY

U.S. DEPARTMENT OF ENERGY

Office of Science

# 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, score, 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

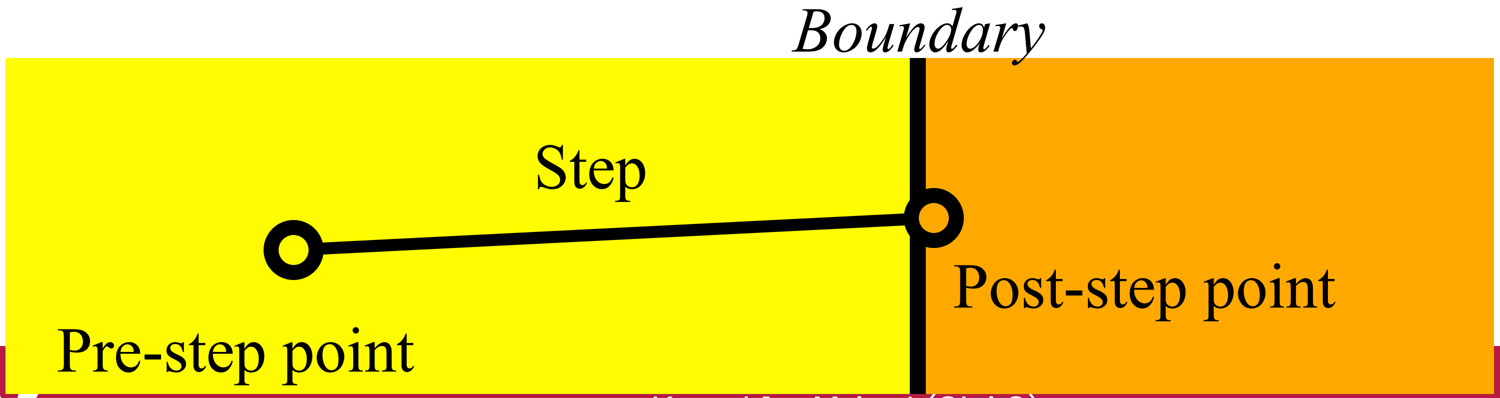
- An event is the basic unit of simulation in Geant4.
- At beginning of processing, primary tracks are generated. These primary tracks are pushed into a stack.
- A track is popped up from the stack one by one and "tracked". Resulting secondary tracks are pushed into the stack.
  – This "tracking" lasts 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.



# Trajectory and trajectory point

- Track does not keep its trace. No track object persists at the end of event.
- G4Trajectory is the class which copies some of G4Track information. G4TrajectoryPoint is the class which copies some of G4Step information.
  - G4Trajectory has a vector of G4TrajectoryPoint.
  - At the end of event processing, G4Event has a collection of G4Trajectory objects.
    - /tracking/storeTrajectory must be set to 1.
- Keep in mind the distinction.
  - G4Track ←→ G4Trajectory, G4Step ←→ G4TrajectoryPoint
- Given G4Trajectory and G4TrajectoryPoint objects persist till the end of an event, you should be careful not to store too many trajectories.
  - E.g. avoid for high energy EM shower tracks.
- G4Trajectory and G4TrajectoryPoint store only the minimum information.
  - You can create your own trajectory / trajectory point classes to store information you need. G4VTrajectory and G4VTrajectoryPoint are base classes.

# 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.

# Tracking and processes

- Geant4 tracking is general.
  - It is independent to
    - the particle type
    - the physics processes involving to a particle
  - It gives the chance to all processes
    - To contribute to determining the step length
    - To contribute any possible changes in physical quantities of the track
    - To generate secondary particles
    - To suggest changes in the state of the track
      - e.g. to suspend, postpone or kill it.

# 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. Celenkov process
  - PostStep (a.k.a. discrete process)
    - e.g. decay on the fly

# Cuts in Geant4

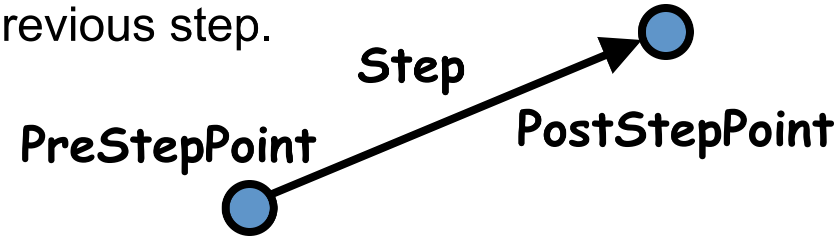
- A Cut in Geant4 is a production threshold.
  - Not tracking cut, which does not exist in Geant4 as default.
    - All tracks are traced down to zero kinetic energy.
  - It is applied only for physics processes that have infrared divergence
- Much detail will be given at later talks on physics.

# Track status

- At the end of each step, according to the processes involved, the state of a track may be changed.
  - The user can also change the status in UserSteppingAction.
  - Statuses shown in green are artificial, i.e. Geant4 kernel won't set them, but the user can set.
- fAlive
  - Continue the tracking.
- fStopButAlive
  - The track has come to zero kinetic energy, but still AtRest process to occur.
- fStopAndKill
  - The track has lost its identity because it has decayed, interacted or gone beyond the world boundary.
  - Secondaries will be pushed to the stack.
- fKillTrackAndSecondaries
  - Kill the current track and also associated secondaries.
- fSuspend
  - Suspend processing of the current track and push it and its secondaries to the stack.
- fPostponeToNextEvent
  - Postpone processing of the current track to the next event.
  - Secondaries are still being processed within the current event.

# Step status

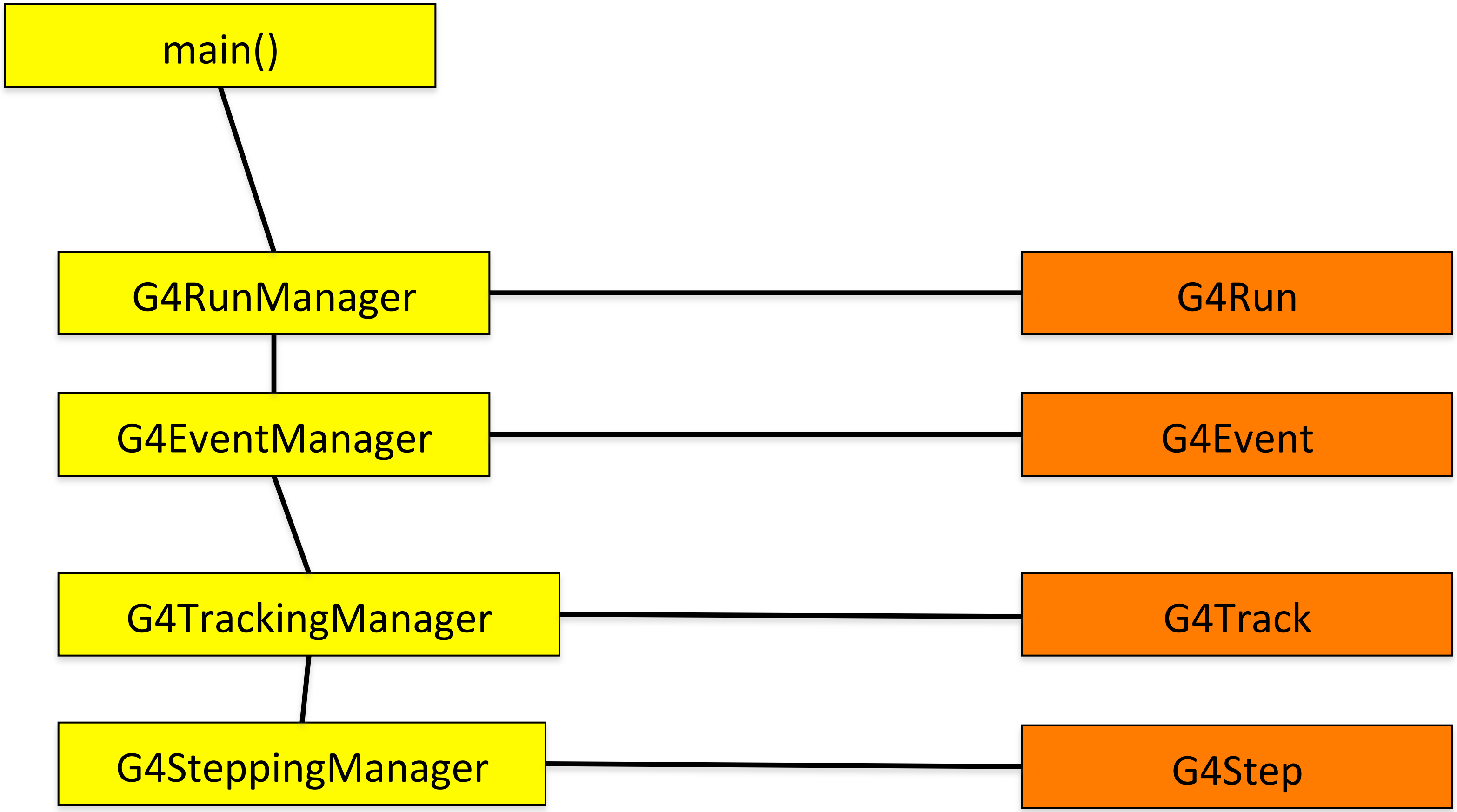
- Step status is attached to G4StepPoint to indicate why that particular step was determined.
  - Use "PostStepPoint" to get the status of this step.
  - "PreStepPoint" has the status of the previous step.



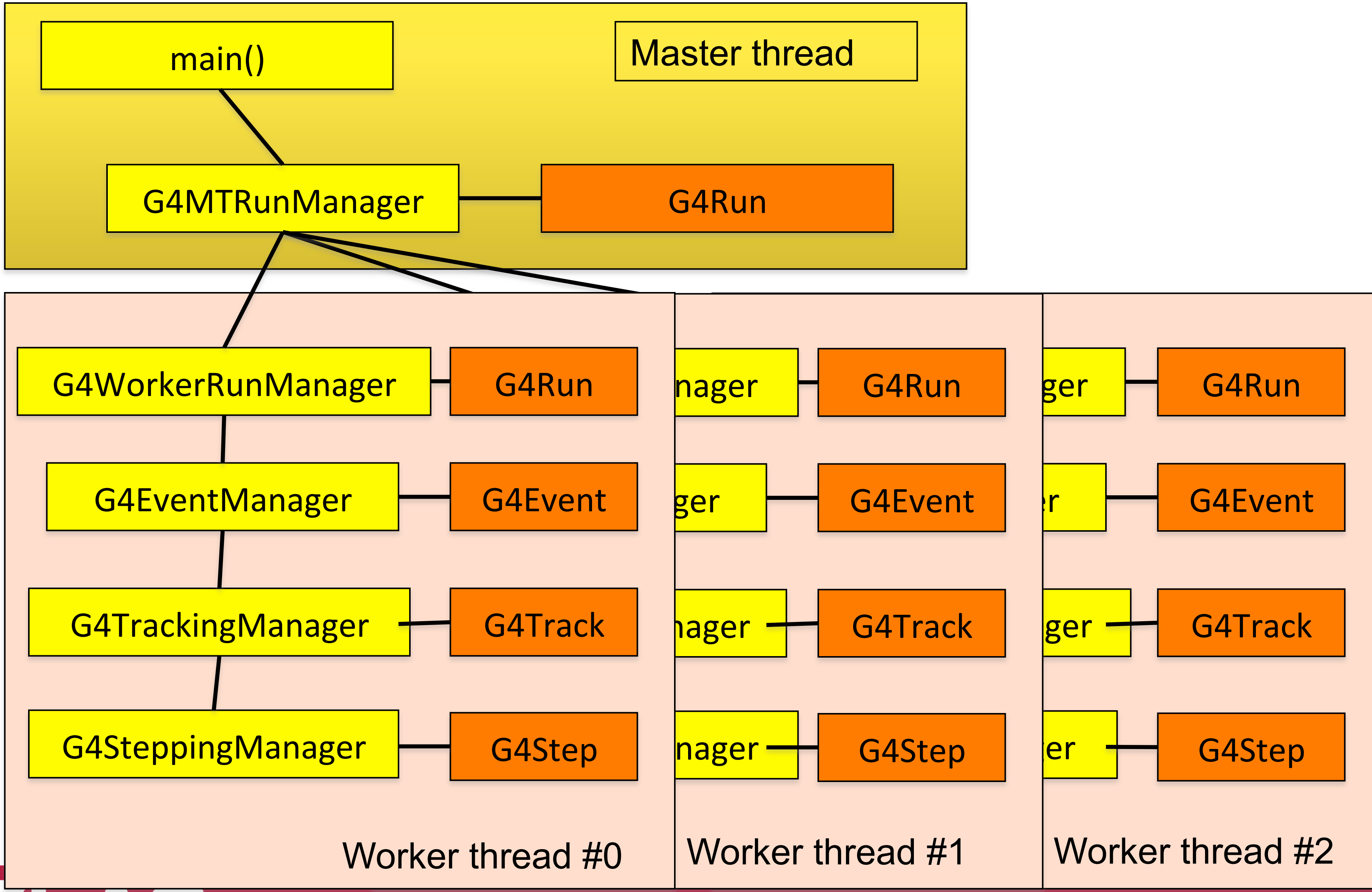

- fWorldBoundary
  - Step reached the world boundary
- fGeomBoundary
  - Step is limited by a volume boundary except the world
- fAtRestDoItProc, fAlongStepDoItProc, fPostStepDoItProc
  - Step is limited by a AtRest, AlongStep or PostStep process
- fUserDefinedLimit
  - Step is limited by the user Step limit
- fExclusivelyForcedProc
  - Step is limited by an exclusively forced (e.g. shower parameterization) process
- fUndefined
  - Step not defined yet

- If you want to identify the first step in a volume, pick fGeomBoudary status in PreStepPoint.
- If you want to identify a step getting out of a volume, pick fGeomBoundary status in PostStepPoint

# Sequential mode



# Multi-threaded mode



# Shared? Thread-local?

- In the multi-threaded mode, generally saying, data that are stable during the event loop are shared among threads while data that are transient during the event loop are thread-local.
- In general, geometry and physics tables are shared, while event, track, step, trajectory, hits, etc., as well as several Geant4 manager classes such as EventManager, TrackingManager, SteppingManager, TransportationManager, FieldManager, Navigator, SensitiveDetectorManager, etc. are thread-local.
- Among the user classes, user initialization classes (G4VUserDetectorConstruction, G4VUserPhysicsList and newly introduced G4VUserActionInitialization) are shared, while all user action classes and sensitive detector classes are thread-local.
  - It is not straightforward (and thus not recommended) to access from a shared class object to a thread-local object, e.g. from detector construction to stepping action.
  - Please note that thread-local objects are instantiated and initialized at the first *BeamOn*.
- To avoid potential errors, it is advised to always keep in mind which class is shared and which class is thread-local.

# Extract useful information

- Given geometry, physics and primary track generation, Geant4 does proper physics simulation "silently".
  - You have to do something to extract information useful to you.
- There are three ways:
  - Built-in scoring commands
    - Most commonly-used physics quantities are available.
  - Use scorers in the tracking volume
    - Create scores for each event
    - Create own Run class to accumulate scores
  - Assign G4VSensitiveDetector to a volume to generate "hit".
    - Use user hooks (G4UserEventAction, G4UserRunAction) to get event / run summary
- You may also use user hooks (G4UserTrackingAction, G4UserSteppingAction, etc.)
  - You have full access to almost all information
  - Straight-forward in sequential mode, but do-it-yourself

# 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.

# G4cout, G4cerr

- G4cout and G4cerr are *ostream* objects defined by Geant4.
  - G4endl is also provided.

```
G4cout << "Hello Geant4!" << G4endl;
```

- Some GUIs are buffering output streams so that they display print-outs on another window or provide storing / editing functionality.
  - The user should not use std::cout, etc.
- The user should not use std::cin for input. Use user-defined commands provided by intercoms category in Geant4.
  - Ordinary file I/O is OK.

# G4cout in multithreaded mode

- By default, every G4cout string is displayed on the screen in the order as it is generated.
  - A line made by a worker thread is preceded by the worker identifier.
- It is not very readable if lines of several worker threads interleave.

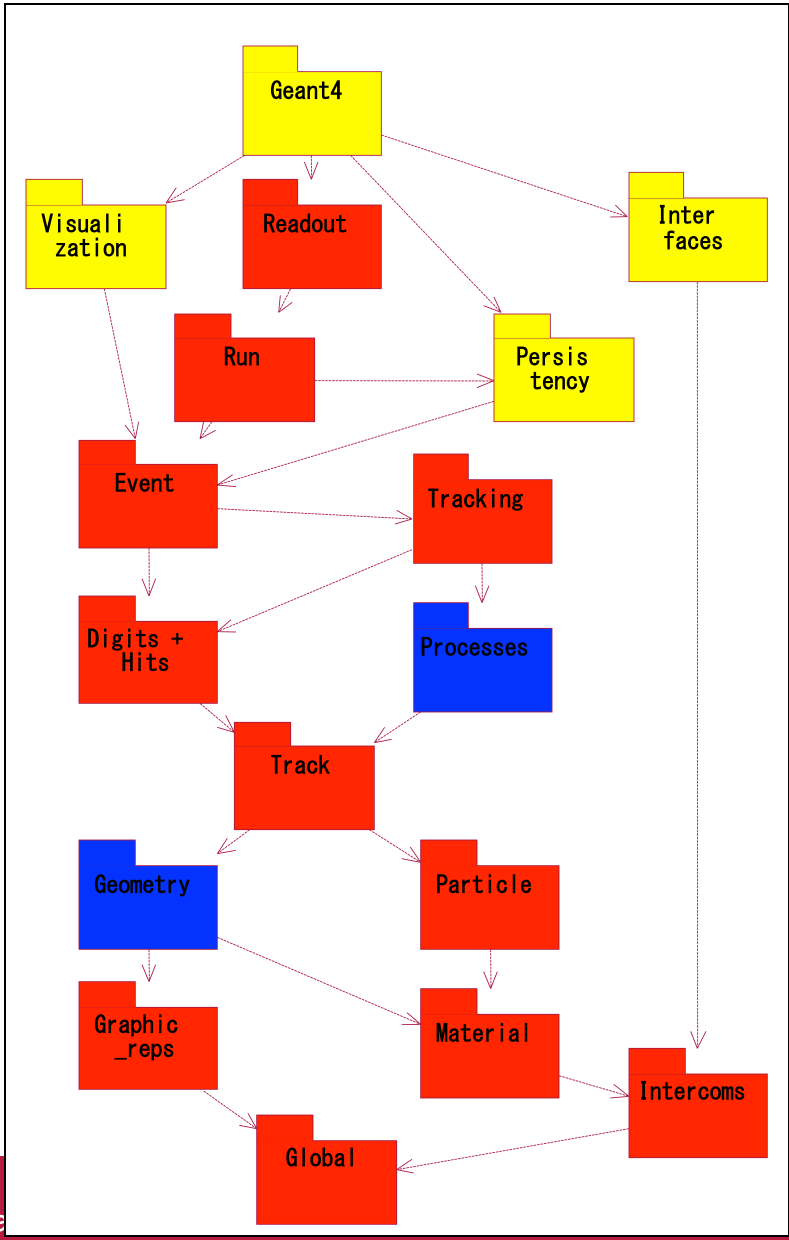
  /control/cout/ignoreThreadsExcept <threadID>
  - Omit cout from worker threads except the specified one.
  - If specified thread ID is greater than the number of threads, no cout is displayed from worker threads. -1 to reset.

  /control/cout/useBuffer <true/false>
  - Send cout stream to a buffer dedicated to each worker thread.
  - The buffered text will be printed at the end of the job for each thread at a time, so that output of each thread is grouped.

  /control/cout/setCoutFile <fileName> <appendFlag>
  - Send G4cout stream to a file dedicated to a thread.
  - If append flag is true output is appended to the existing file, otherwise file output is overwritten.
  - To return to a display output, use special file name "**Screen**".

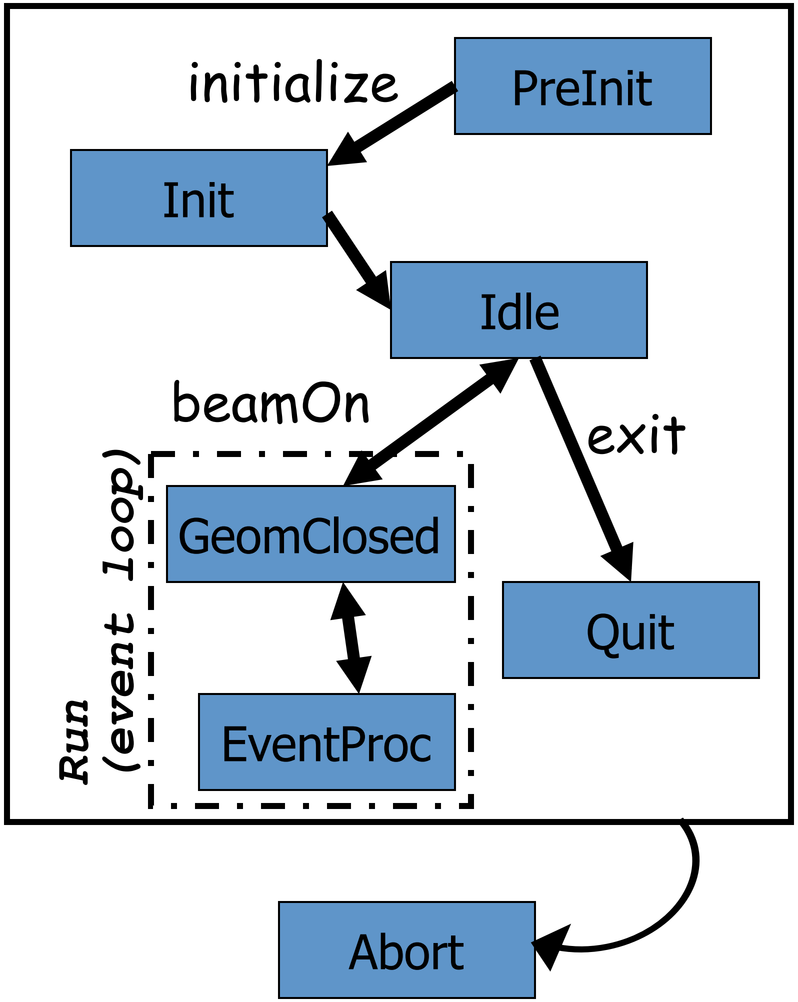
# Geant4 kernel

- 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 seven application states.
  - G4State_PreInit
    - Initial condition
  - G4State_Init
    - During initialization
  - 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




Note: Toggles between GeomClosed and EventProc occur for each thread asynchronously in multithreaded mode.

A SIMULATION TOOLKIT

Version 10.4-p02

# 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
    - G4VUserActionInitialization
- Action classes
  - Instantiate in your G4VUserActionInitialization.
  - Invoked during an event loop
    - G4VUserPrimaryGeneratorAction
    - G4UserRunAction
    - G4UserEventAction
    - G4UserStackingAction
    - G4UserTrackingAction
    - G4UserSteppingAction

Note : classes written in red are mandatory.

# The main program

- Geant4 does not provide a *main*().
- In your *main(),* you have to
  - Construct G4RunManager (sequential mode) or G4MTRunManager (multithreaded mode)
  - Set user mandatory initialization classes to RunManager
    - G4VUserDetectorConstruction
    - G4VUserPhysicsList
    - G4VUserActionInitialization
- 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()*, that is invoked in the master thread (and in sequential mode)
  - Instantiate all necessary materials
  - Instantiate volumes of your detector geometry
- In the virtual method *ConstructSDandField()*, that is invoked in each worker thread (and in sequential mode)
  - Instantiate your sensitive detector classes and field classes and set them to the corresponding logical volumes and field managers, respectively.

# 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)
- Primarily, the user's task is choosing a "pre-packaged" physics list, that combines physics processes and models that are relevant to a typical application use-cases.
  - If "pre-packaged" physics lists do not meet your needs, you may add or alternate some processes/models.
  - If you are brave enough, you may implement your physics list.

# Generate primary event

- This is the only mandatory user action class.
- 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

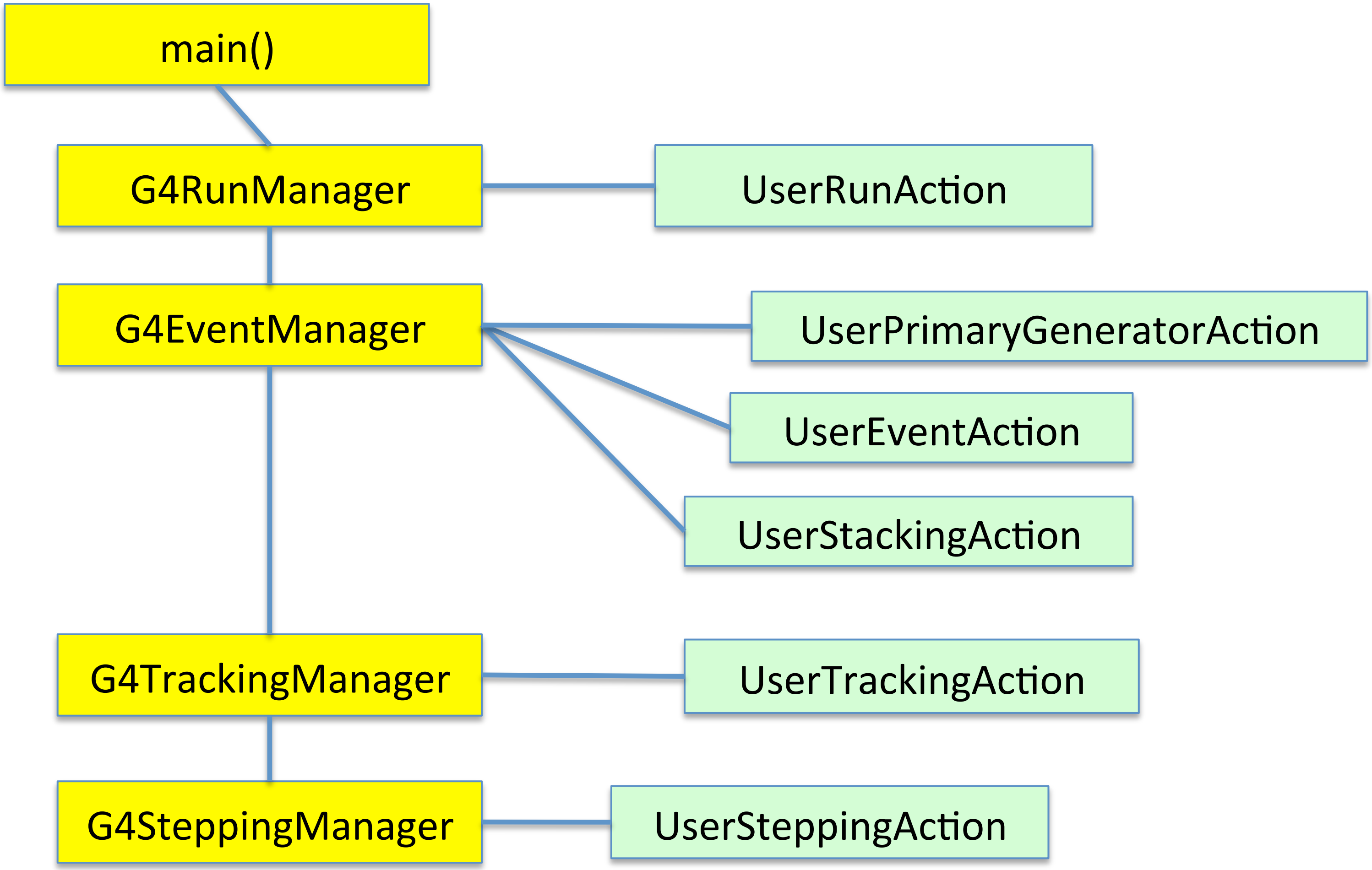
# Optional user action classes

- 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)

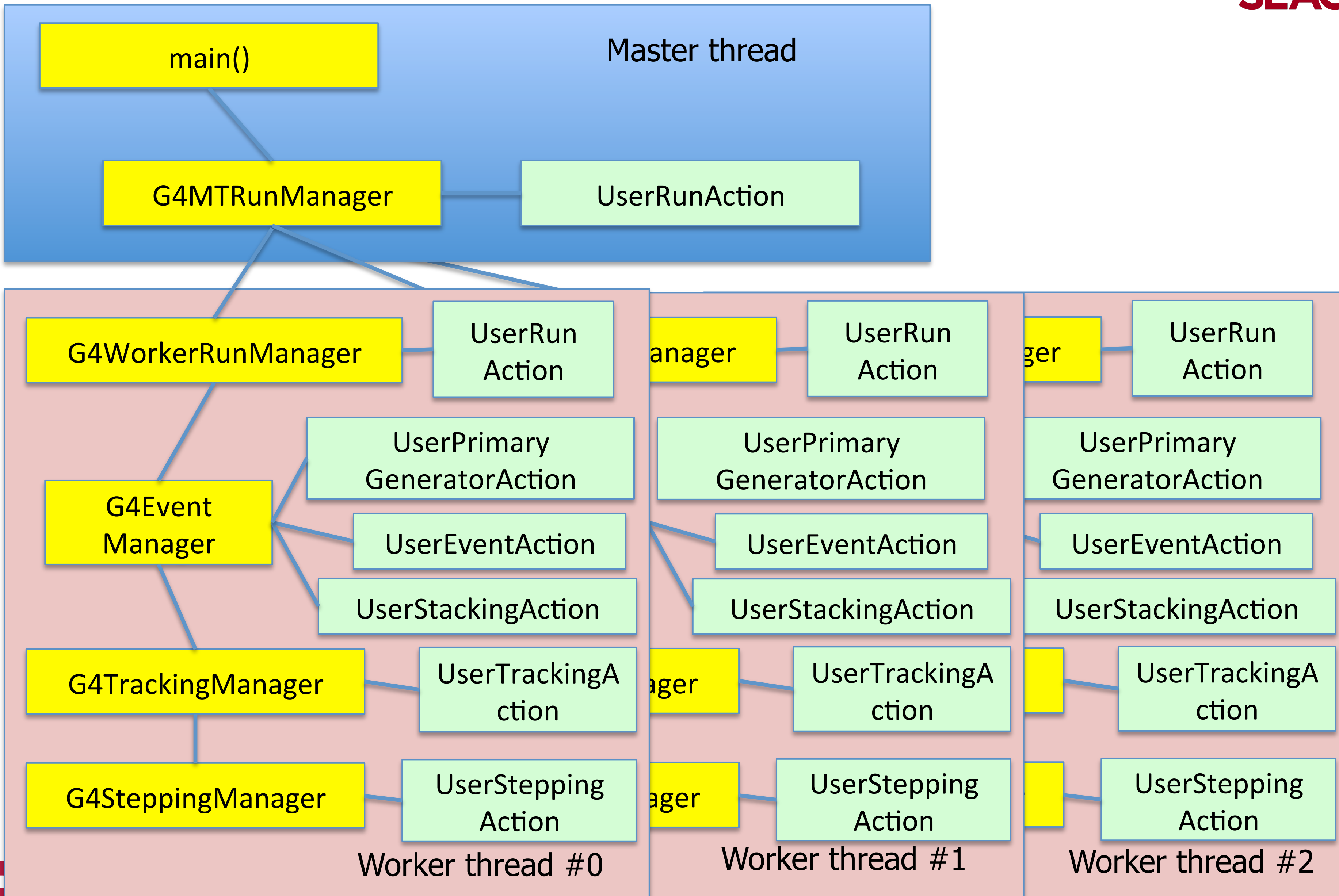
# Optional user action classes

- 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)

# Sequential mode



# Multi-threaded mode



# Instantiate user action classes

- G4VUserActionInitialization has two virtual methods.
- *Build()*
  – Invoked at the beginning of each worker thread as well as in sequential mode
  – Use *SetUserAction()* method to register pointers of all user actions.
  – In multithreaded mode, all user action class objects instantiated in this method are thread-local.
    - User run action instantiated in this method is for thread-local run
- *BuildForMaster()*
  – Invoked only at the beginning of the master thread in multithreaded mode
  – Use *SetUserAction()* method to register pointer of user run action for the global run.

# 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
- Instantiate user action classes
  - ➔ G4VUserActionInitialization
    - Hands-on
- Define the way of primary particle generation
  - ➔ G4VUserPrimaryGeneratorAction
    - Primary particle lecture
- Define the way to extract useful information from Geant4
  - ➔ G4VUserDetectorConstruction, G4UserEventAction, G4Run, G4UserRunAction
  - ➔ G4SensitiveDetector, G4VHit, G4VHitsCollection
    - Scoring lectures