Geant 4

Version 10.0-p01

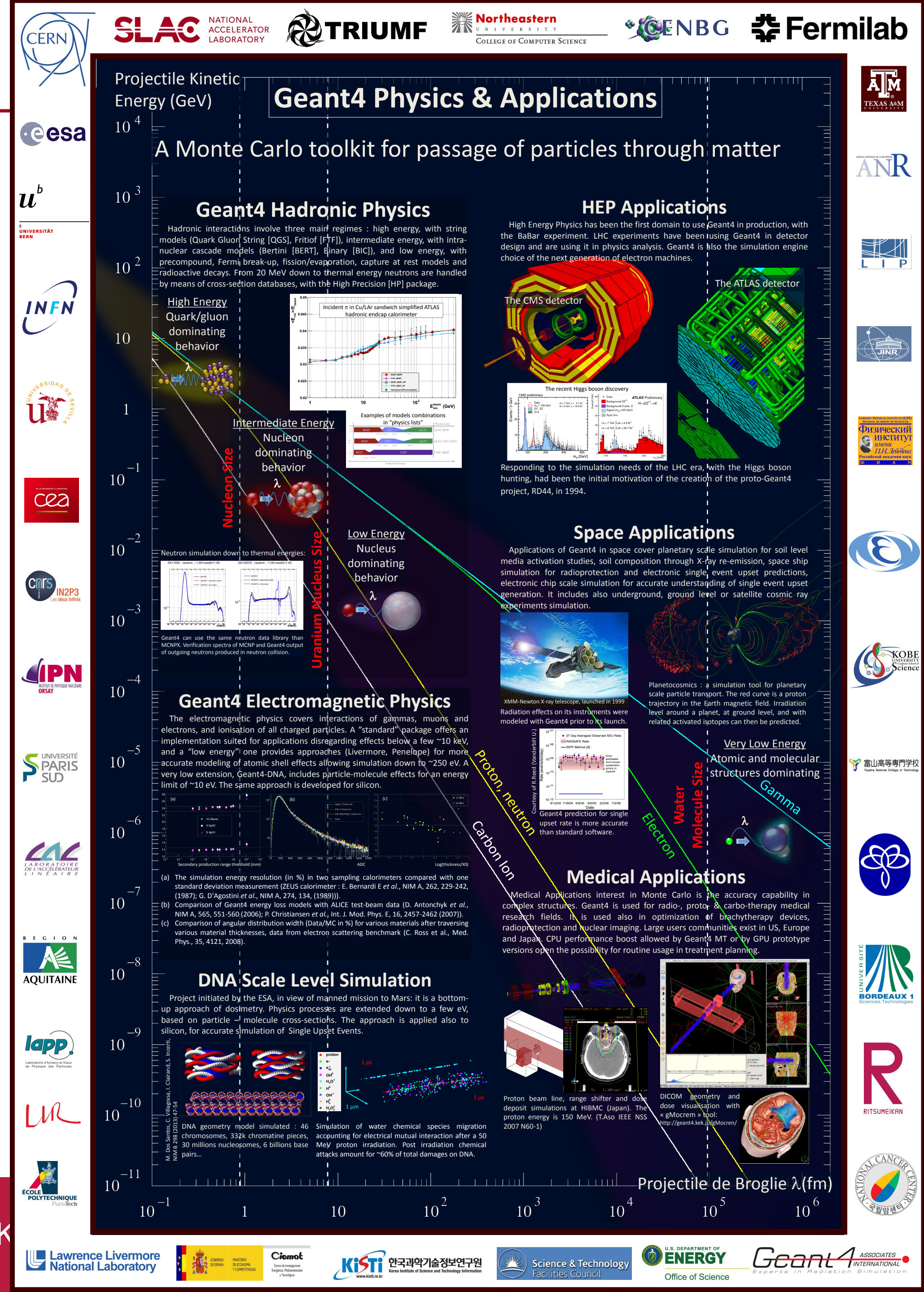
Kernel I

Makoto Asai (SLAC)

Geant4 Tutorial Course

Contents

- General introduction and brief history
- Highlights of user applications
- Geant4 license
- Geant4 kernel
  - Basic concepts and kernel structure
  - User classes



Geant 4

Version 10.0-p01

General introduction and brief history

NATIONAL ACCELERATOR LABORATORY

U.S. DEPARTMENT OF ENERGY

Office of Science

Geant4

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

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.
- 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.

Geant4 – Its history

- Dec ’94 - Project start
- Apr ’97 - First alpha release
- Jul ’98 - First beta release
- Dec ’98 - First Geant4 public release - version 1.0
- …
- Dec 2^{nd}, ’11 – Geant4 version 9.5 release
  - Oct 22^{nd}, ’12 - Geant4 9.5-patch02 release
- Nov 30^{th}, ’12 – Geant4 version 9.6 release
  - May 17^{th}, ’13 - Geant4 9.6-patch02 release
- Dec 6^{th}, ’13 – Geant4 version 10.0 release
  - Feb 28^{th}, ’14 - Geant4 10.0-patch01 release ← Current version

- We currently provide one public release every year.
  - Beta releases are also available.
  - Release announcements on Collaboration Web pages and through the announcement mailing list

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.
  - A few interfaces need to be changed due to multi-threading
- It offers two build options.
  - Multi-threaded mode (including single thread)
  - Sequential mode
    - In case a user depends on thread-unsafe external libraries, he may install Geant4 in sequential mode.

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

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

http://top25.sciencedirect.com/journal/01689002/

Top 25 Hottest Articles

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

July to September 2013

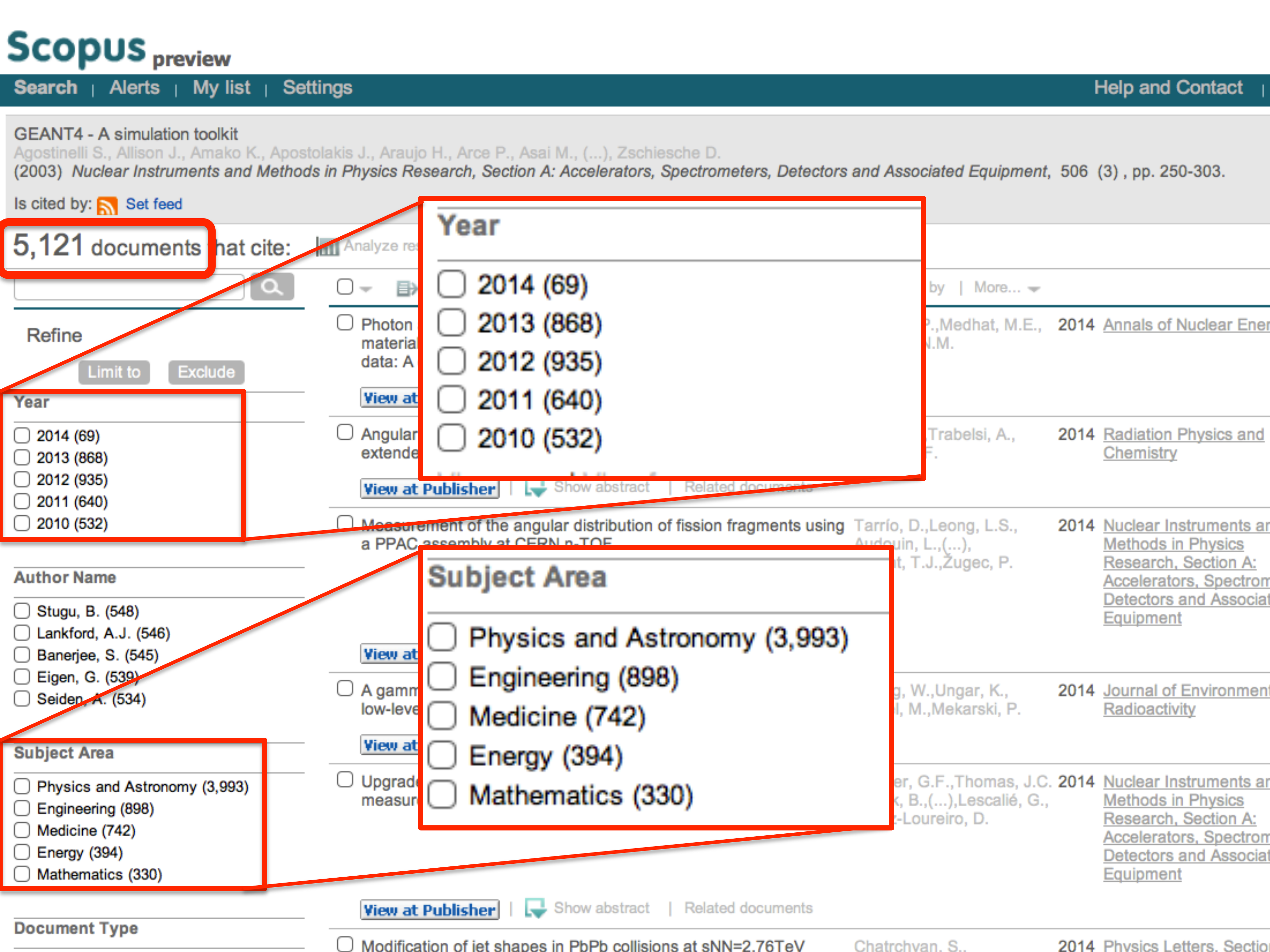
RSS Blog This! Print **Show condensed**

1. **Geant4-a simulation toolkit**
Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 506, Issue 3, July 2003, Pages 250-303
Agostinelli, S.; Allison, J.; Amako, K.; Apostolakis, J.; Araujo, H.; Arce, P.; Asai, M.; Axen, D.; Banerjee, S.; Barrand, G.; Behner, F.; Bellagamba, L.; Boudreau, J.; Broglia, L.; Brunengo, A.; Burkhardt, H.; Chauvie, S.; Chuma, J.; Chytracek, R.; Coope
Cited by Scopus (5016)

2. **The Nuclear Science References (NSR) database and Web Retrieval System**
Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 640, Issue 1, June 2011, Pages 213-218
Pritychenko, B.; Betak, E.; Kellett, M.A.; Singh, B.; Totans, J.
Cited by Scopus (6)

3. **Radioluminescent characteristics of the EJ 299-33 plastic scintillator**
Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 728, November 2013, Pages 36-39
Nyibule, S.; Henry, E.; Schroder, W.U.; Toke, J.; Acosta, L.; Auditore, L.; Cardella, G.; De Filippo, E.; Francalanza, L.; Giani, S.; Minniti, T.; Morgana, E.; Pagano, E.V.; Pirrone, S.; Politi, G.; Quattrocchi, L.; Rizzo, F.; Russotto, P.; Trifiro, A.; T

4. **The BABAR detector** • Review article
Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 479, Issue 1, February 2002, Pages 1-116
Aubert, B.; Bazan, A.; Boucham, A.; Boutigny, D.; De Bonis, I.; Favier, J.; Gaillard, J.-M.; Jeremie, A.; Karyotakis, Y.; Le Flour, T.; Lees, J.P.; Lieunard, S.; Petitpas, P.; Robbe, P.; Tisserand, V.; Zachariadou, K.; Palano, A.; Chen, G.P.; Chen, J.C.;


Scopus preview
Search | Alerts | My list | Settings
Help and Contact
GEANT4 - A simulation toolkit
Agostinelli S., Allison J., Amako K., Apostolakis J., Araujo H., Arce P., Asai M., (...), Zschiesche D.
(2003) Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 506 (3), pp. 250-303.
Is cited by: Set feed
5,121 documents that cite:
Analyze results
More...
Refine
Limit to
Exclude
Year
2014 (69)
2013 (868)
2012 (935)
2011 (640)
2010 (532)
Author Name
Stugu, B. (548)
Lankford, A.J. (546)
Banerjee, S. (545)
Eigen, G. (539)
Seiden, A. (534)
Subject Area
Physics and Astronomy (3,993)
Engineering (898)
Medicine (742)
Energy (394)
Mathematics (330)
Document Type
2014 Annals of Nuclear Energy
2014 Radiation Physics and Chemistry
Measurement of the angular distribution of fission fragments using a PPAC assembly at CERN n-TOF
Tarrío, D.,Leong, L.S., Audouin, L.,(...), Žugec, P.
2014 Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment
2014 Journal of Environmental Radioactivity
2014 Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment
View at Publisher | Show abstract | Related documents
Modification of jet shapes in PbPb collisions at sNN=2.76TeV
Chatrchyan, S.,
2014 Physics Letters, Section

Web Images More...

Google

Geant4

Scholar

About 21,400 results (**0.03** sec)

Articles

Case law

My library New!

Any time

Since 2014

Since 2013

Since 2010

Custom range...

Sort by relevance

Sort by date

include patents

include citations

Create alert

[GEANT4—a simulation toolkit]

S Agostinelli, J Allison, K Amako, J Apostolakis… - Nuclear instruments and …, 2003 - Elsevier

Geant4 is a toolkit for simulating the passage of particles through matter. It includes a complete range of functionality including tracking, geometry, physics models and hits. The physics processes offered cover a comprehensive range, including electromagnetic, **...**

Cited by 9303 Related articles All 20 versions Cite Save

[Geant4 developments and applications]

J Allison, K Amako, J Apostolakis… - Nuclear Science, …, 2006 - ieeexplore.ieee.org

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 **...**

Cited by 2108 Related articles All 23 versions Cite Save

[GATE (Geant4 Application for Tomographic Emission): a PET/SPECT general-purpose simulation platform]

D Strulab, G Santin, D Lazaro, V Breton… - Nuclear Physics B- …, 2003 - Elsevier

We present the development of GATE, the **Geant4** Application for Tomographic Emission, as a new general purpose simulation platform for PET and SPECT applications. Built on top of the **Geant4** simulation toolkit, it provides multiple new features with the objective to ease **...**

Cited by 99 Related articles All 11 versions Cite Save

[GATE, a Geant4-based simulation platform for PET integrating movement and time management]

G Santin, D Strul, D Lazaro, L Simon… - … Record, 2002 IEEE, 2002 - ieeexplore.ieee.org

Emission, is a simulation platform developed for PET and SPECT. It combines a powerful

Geant 4

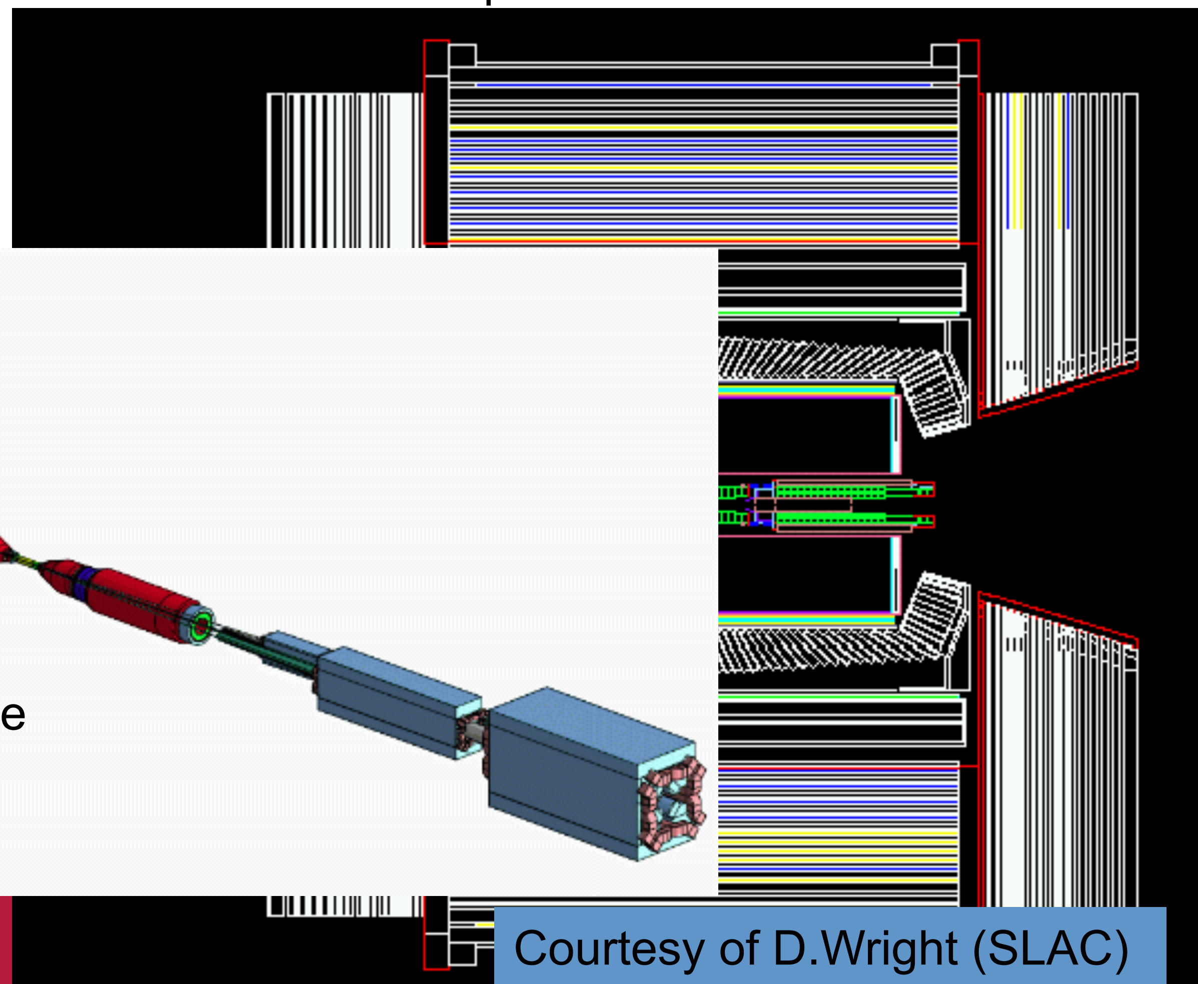
Version 10.0-p01

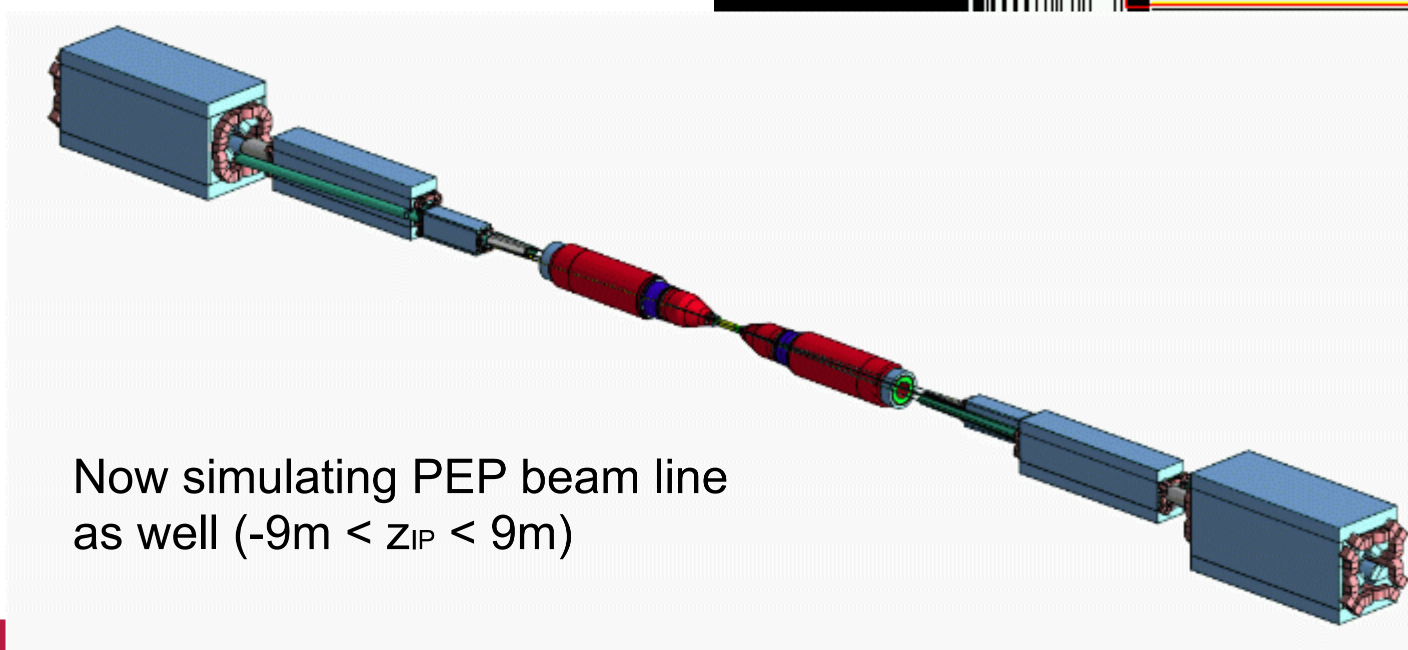
Highlights of Users Applications

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

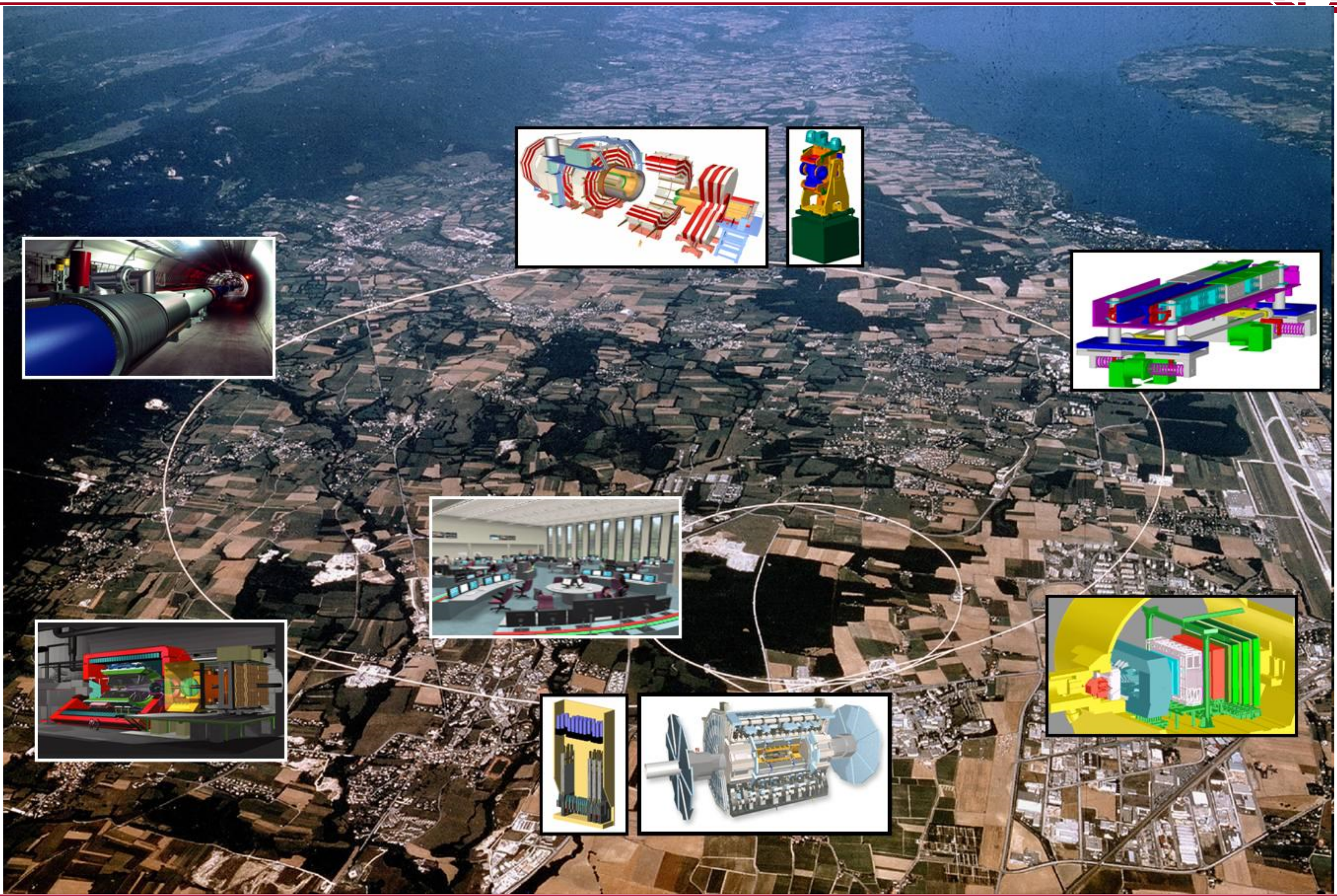




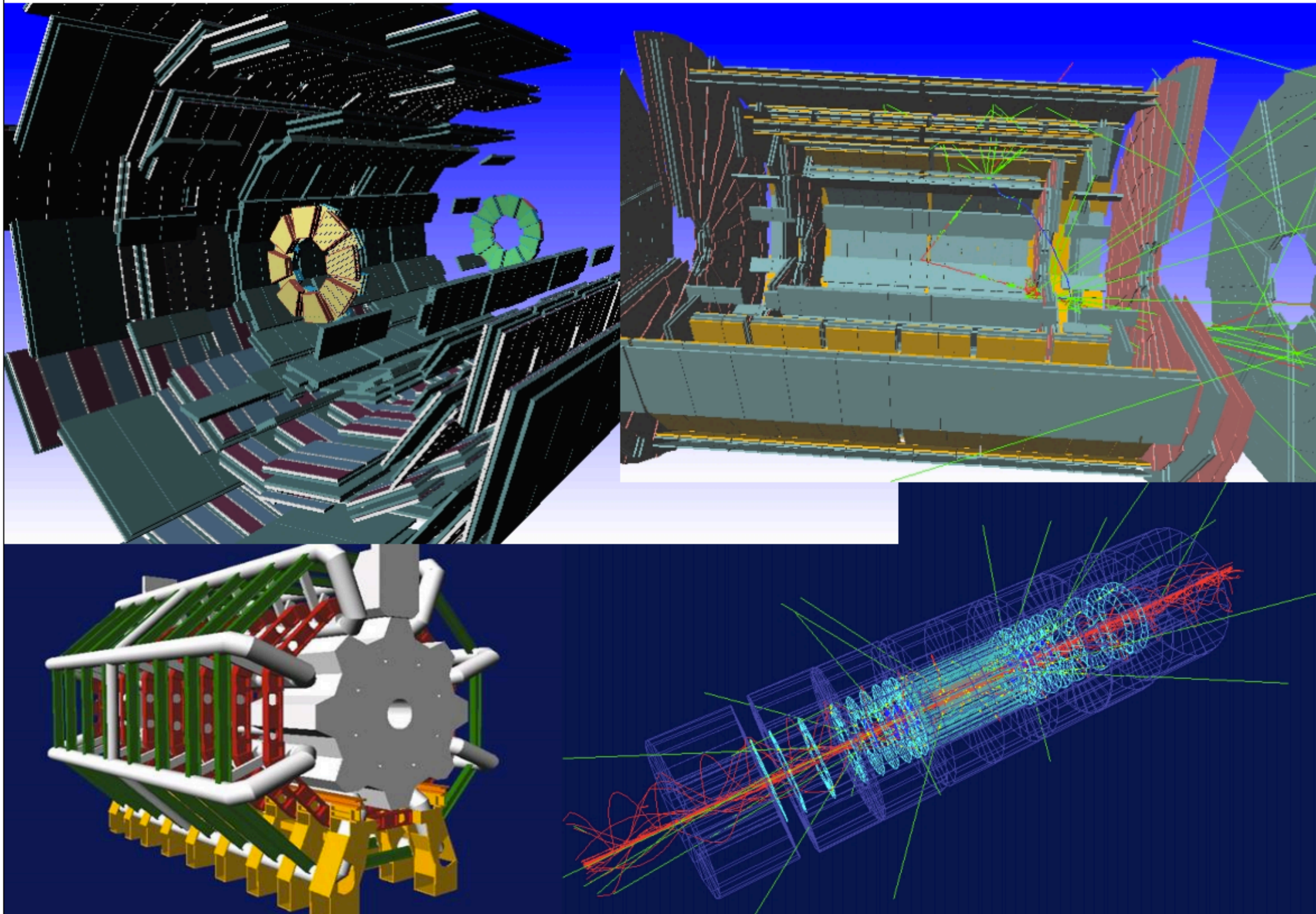
Now simulating PEP beam line as well (-9m < z_{IP} < 9m)

Courtesy of D.Wright (SLAC)

Large Hadron Collider (LHC) @ CERN

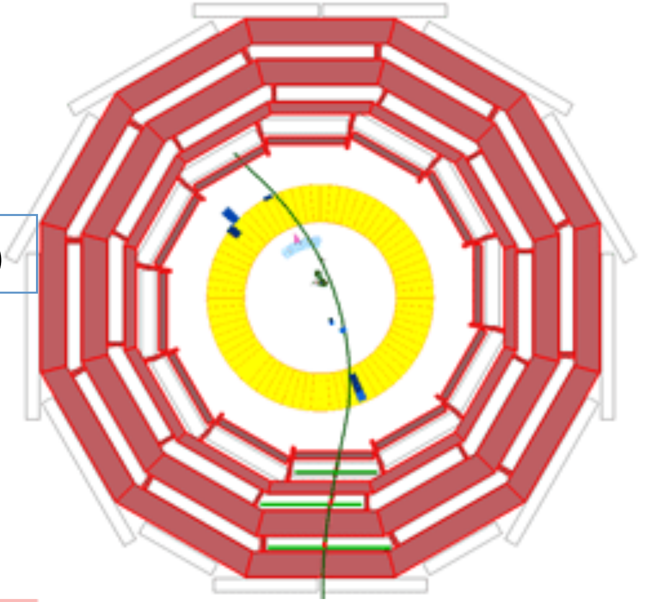


Geant4 in High Energy Physics (ATLAS at LHC)



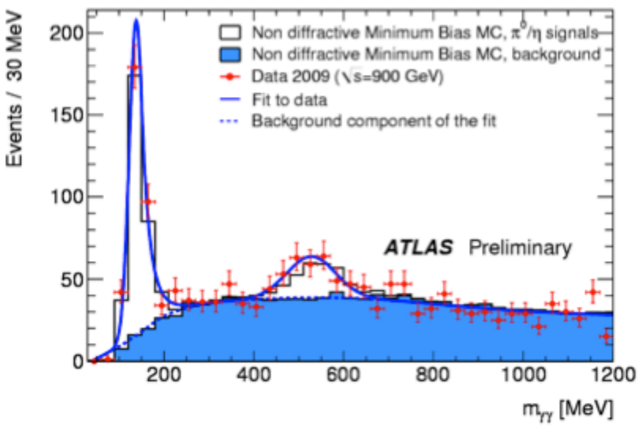
Geant4 has been successfully employed for

- Detector design
- Calibration / alignment
- First analyses



T. LeCompte (ANL)

GEANT4 Comparisons with the Calorimeters



Invariant mass of pairs of well-isolated electromagnetic clusters.

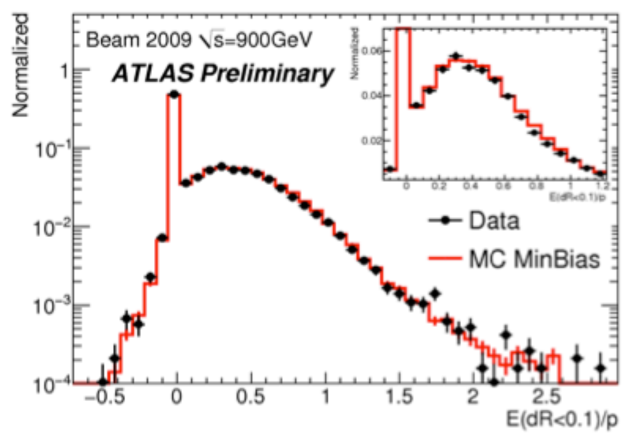
The π^0 mass is within 0.8 ± 0.6% of expectations.

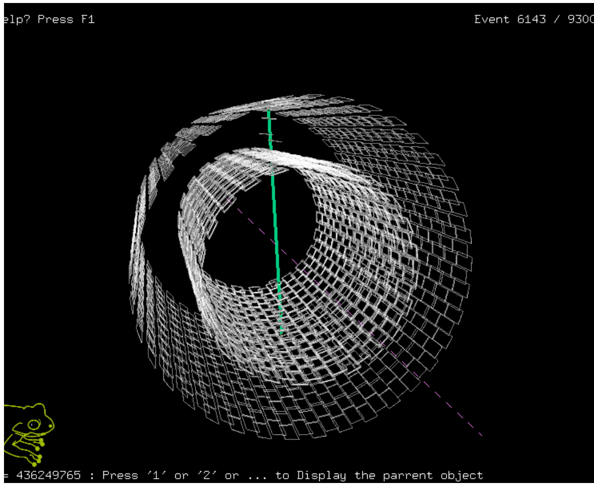
The η^0 mass is within 3 ± 2% of expectations.

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.

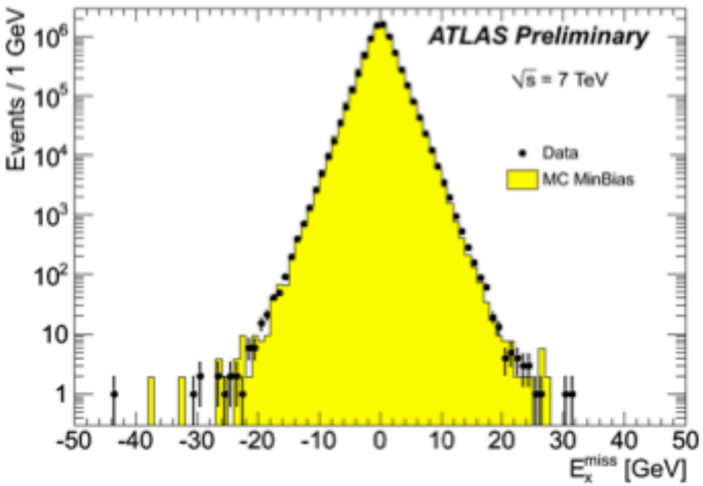
This plot agreed better than we ever expected. (I sent the student who made it back to make sure that they didn't accidentally compare G4 with G4.

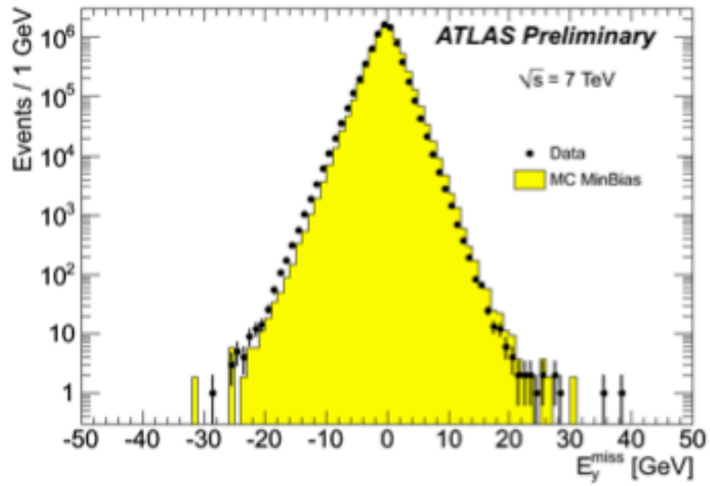




Figures from CMS

Missing ET

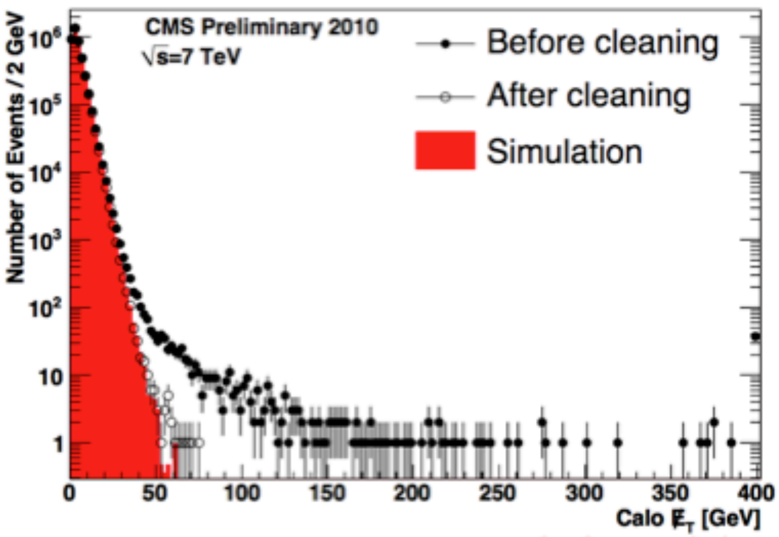


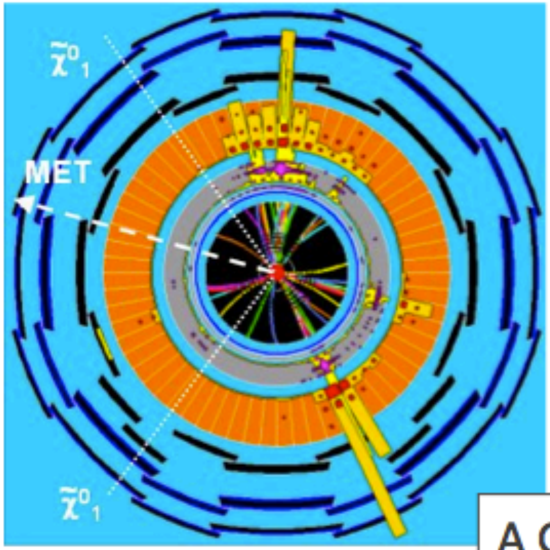


This is one of the hardest things to get right. MET incorporates everything measured in the detector and attempts to identify non-interacting particles, such as neutrinos or dark matter.

Agreement is astounding.

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



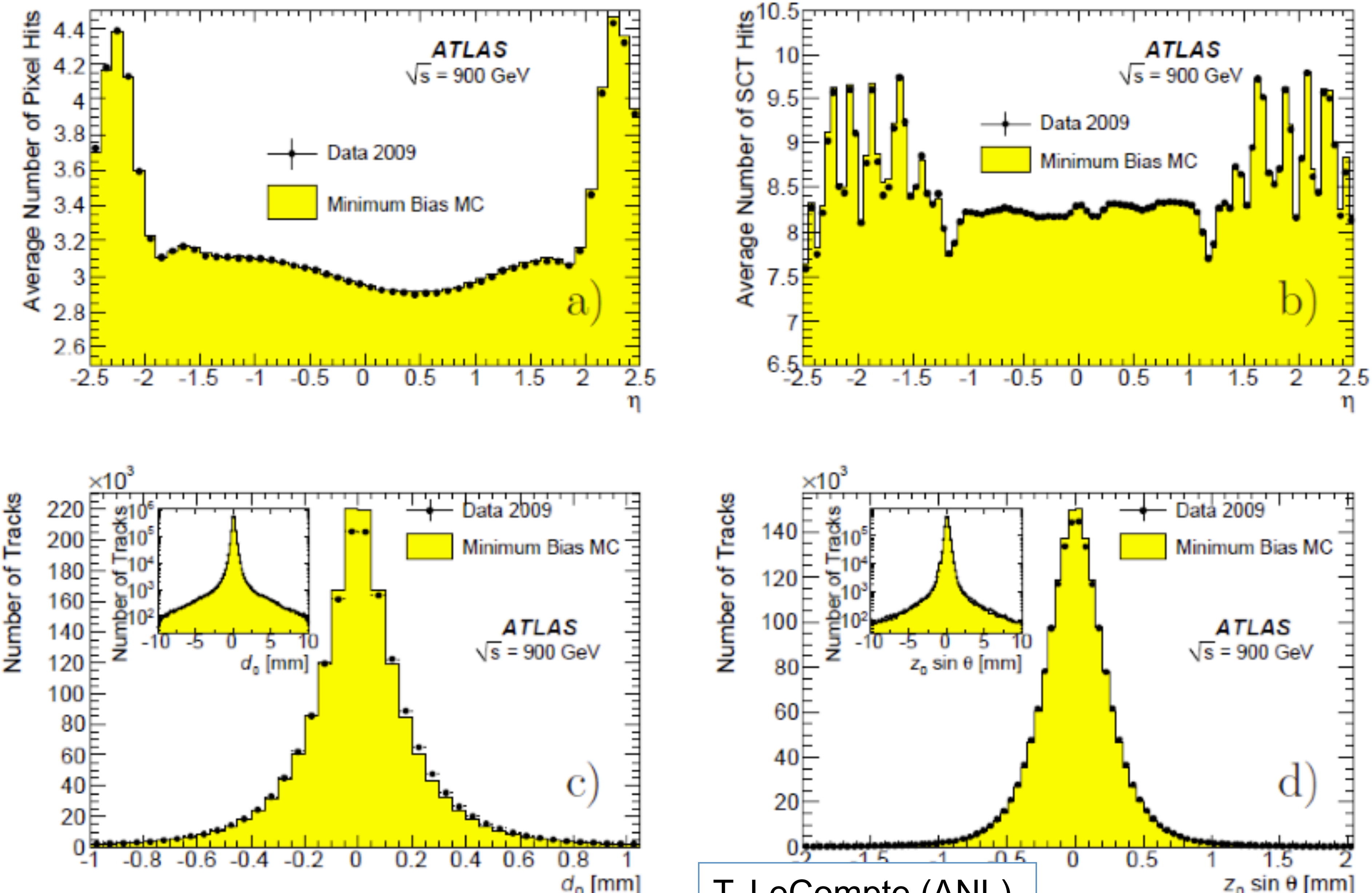


A GEANT4 event.

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

T. LeCompte (ANL)

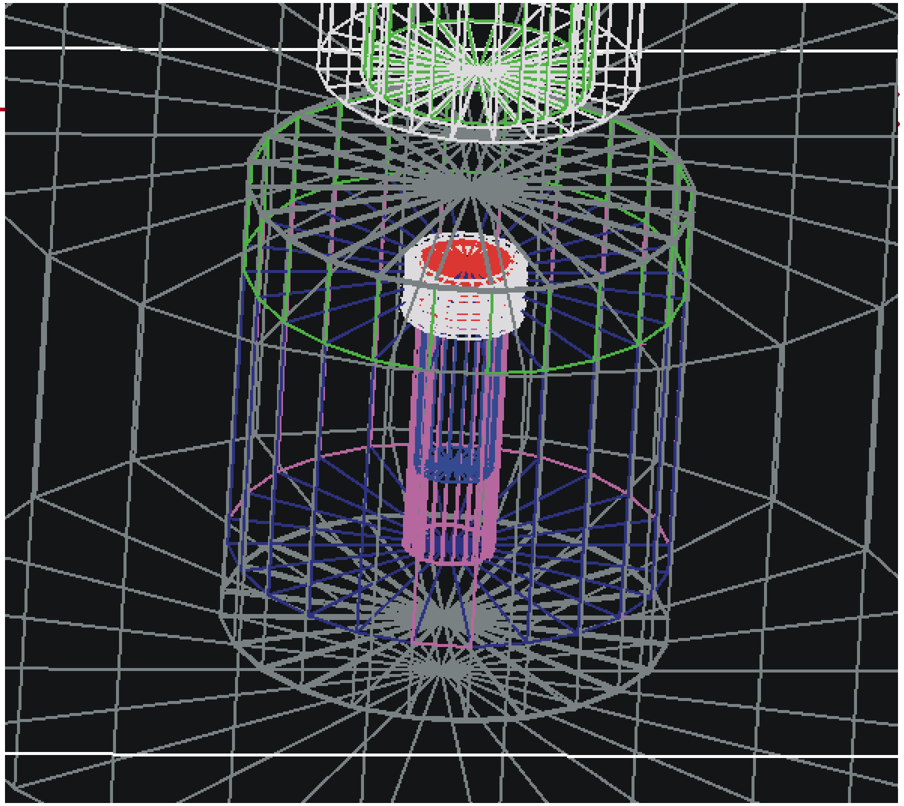
Data and simulation agreements



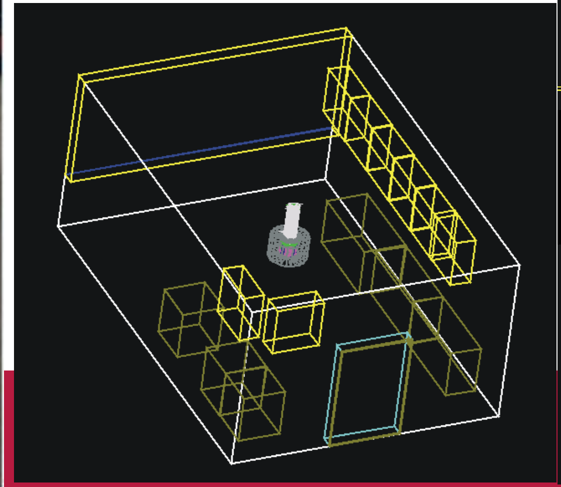
T. LeCompte (ANL)

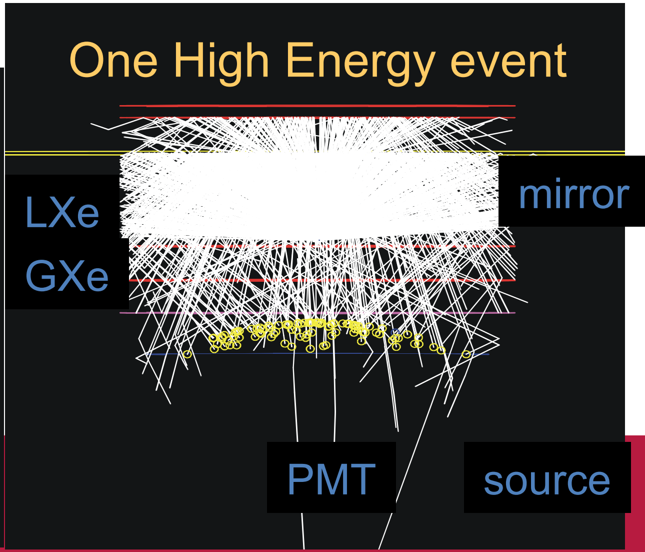
Boulby Mine dark matter search

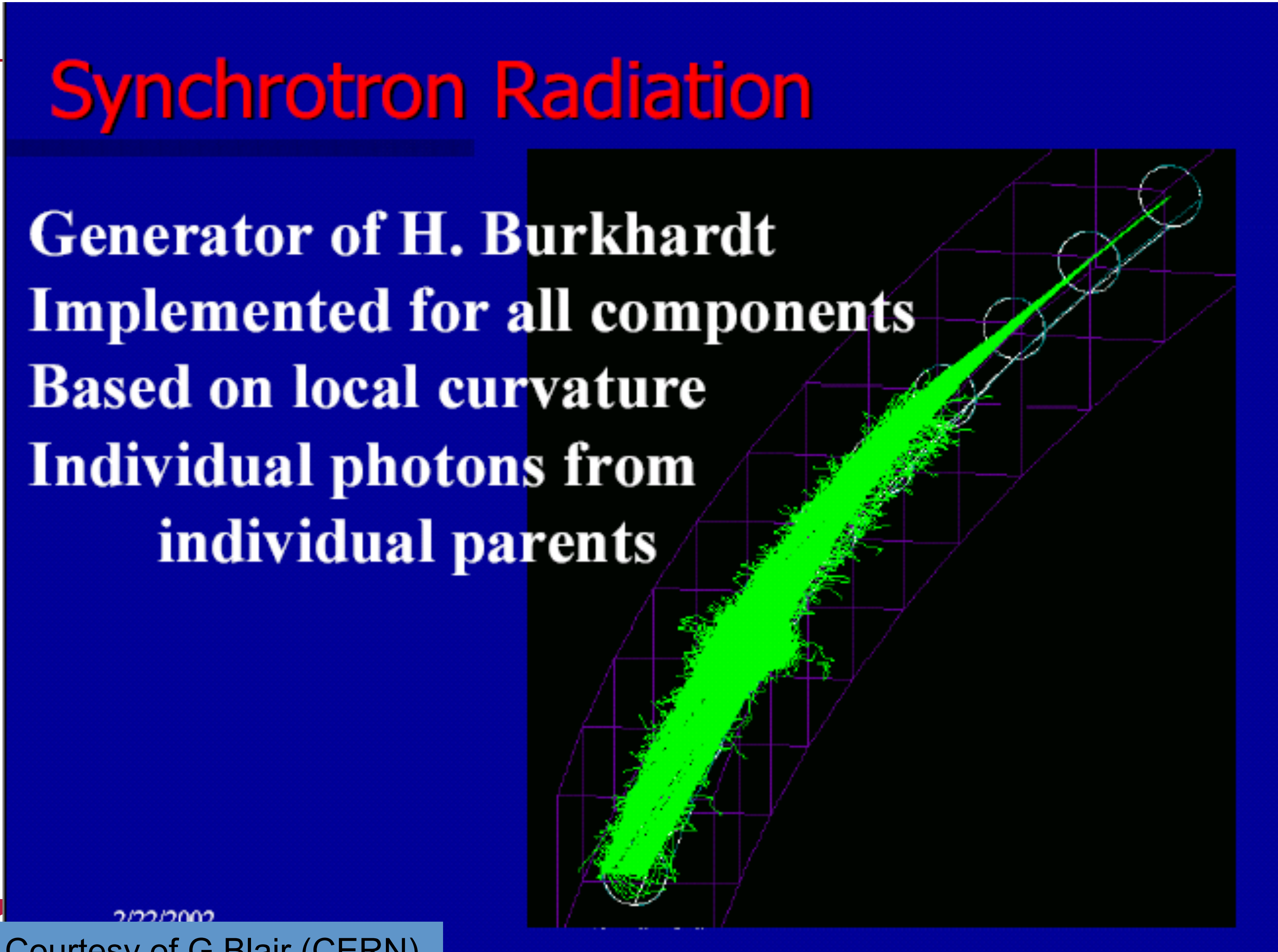
Prototype Simulation



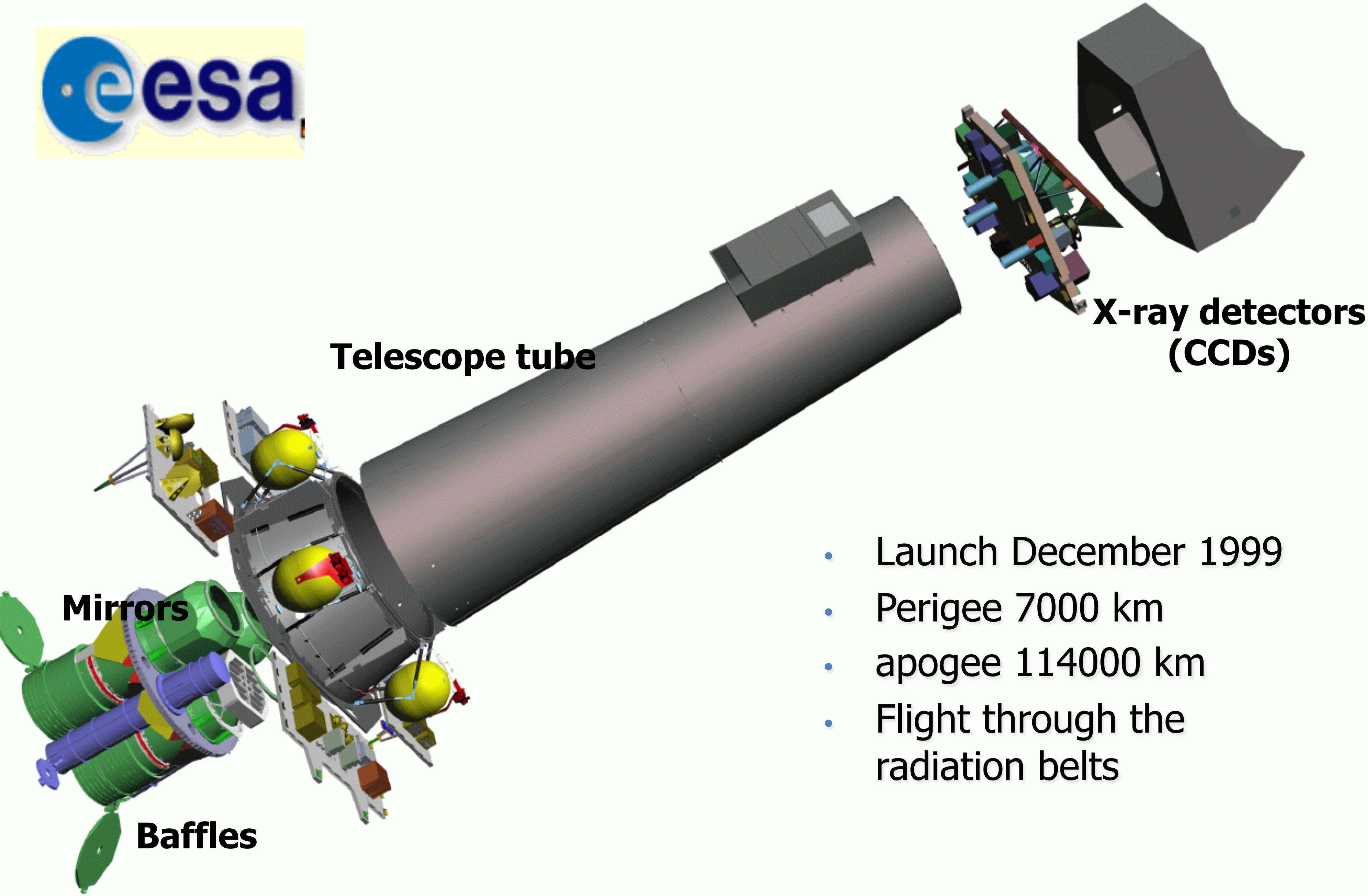
Courtesy of H. Araujo, A. Howard, IC London







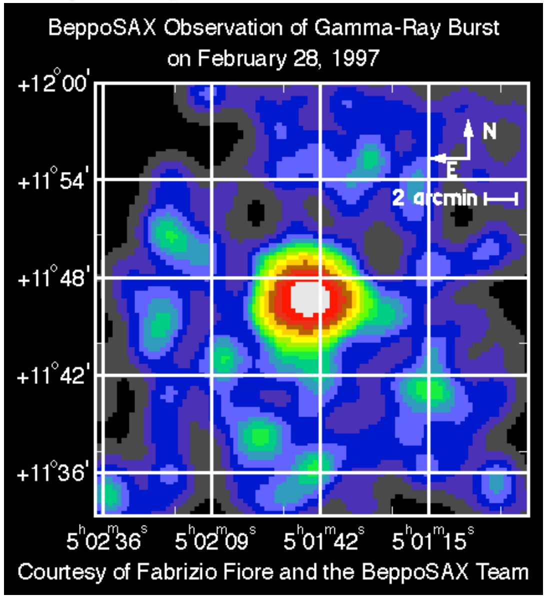
Courtesy of G.Blair (CERN)

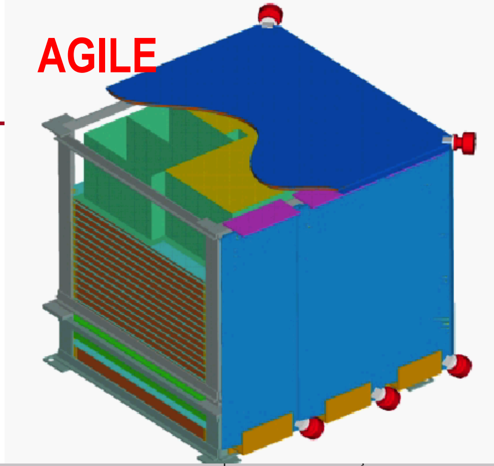


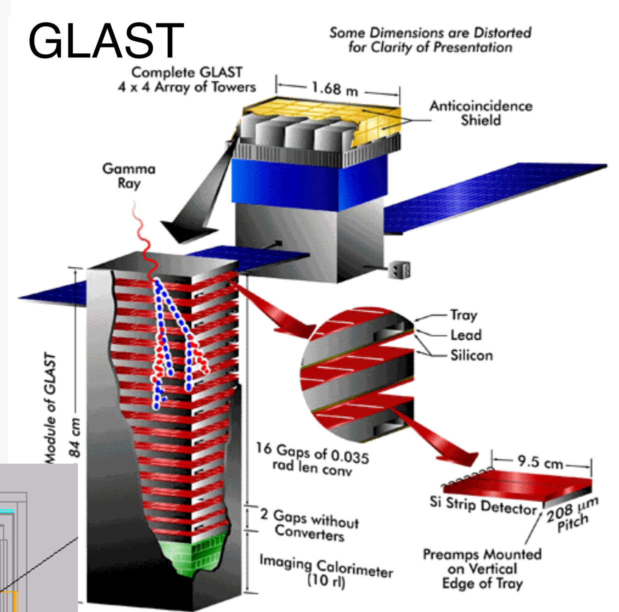
- Launch December 1999
- Perigee 7000 km
- apogee 114000 km
- Flight through the radiation belts

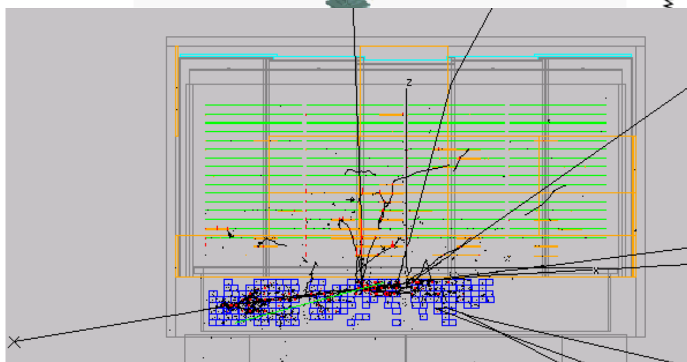
γ astrophysics

γ-ray bursts

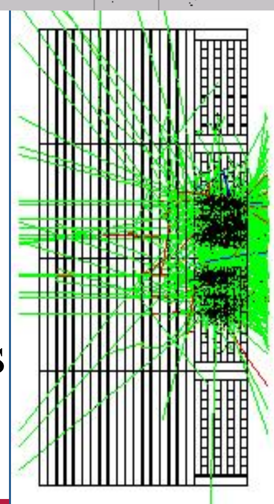








GLAST



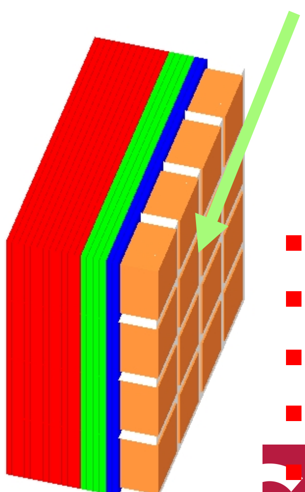
Typical telescope:

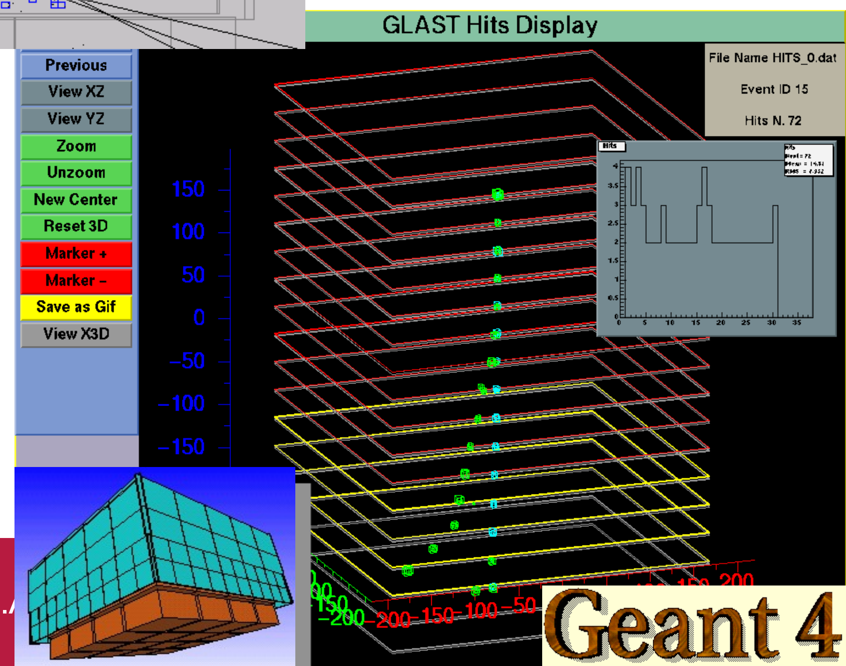
Tracker

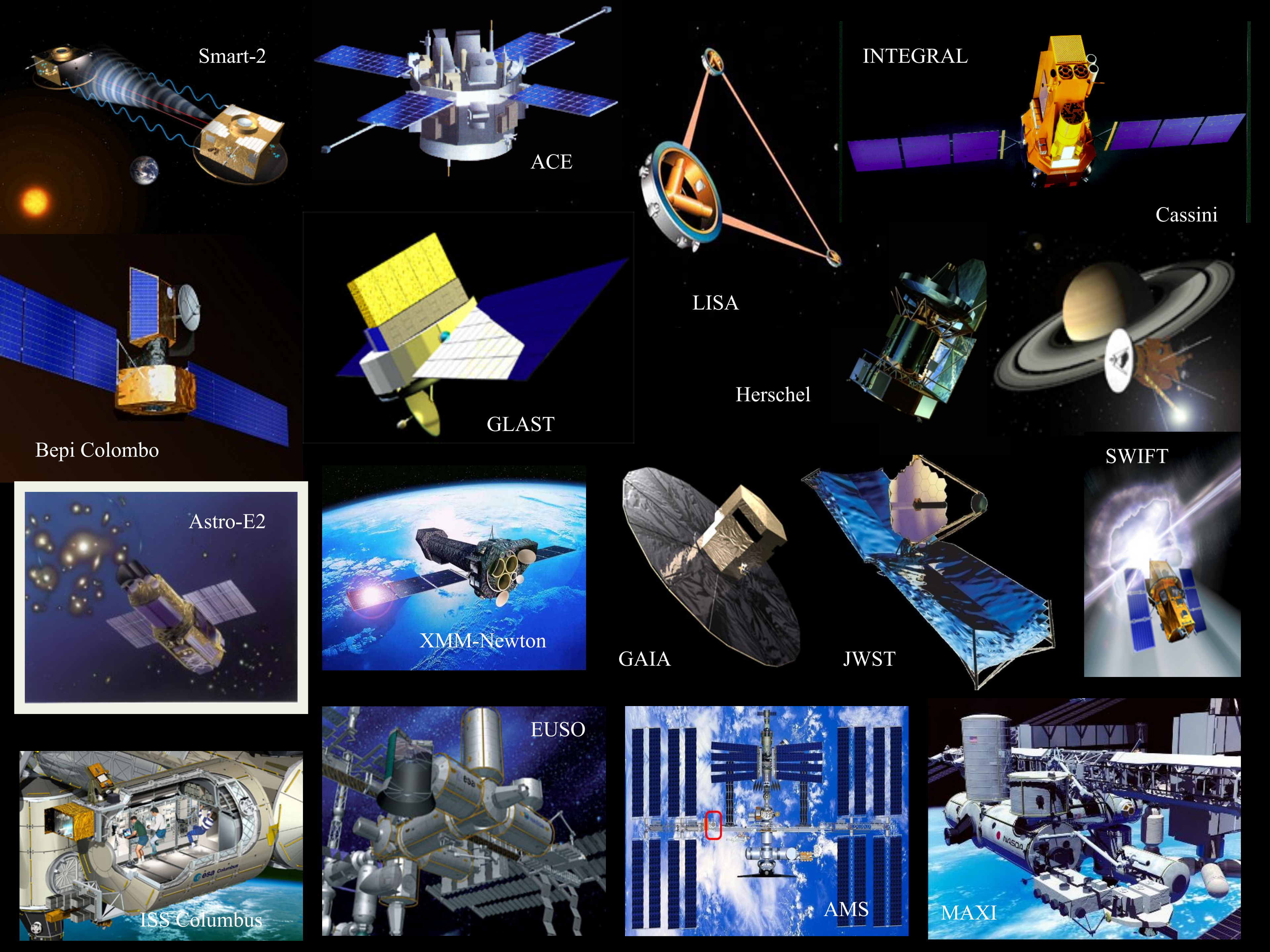
Calorimeter

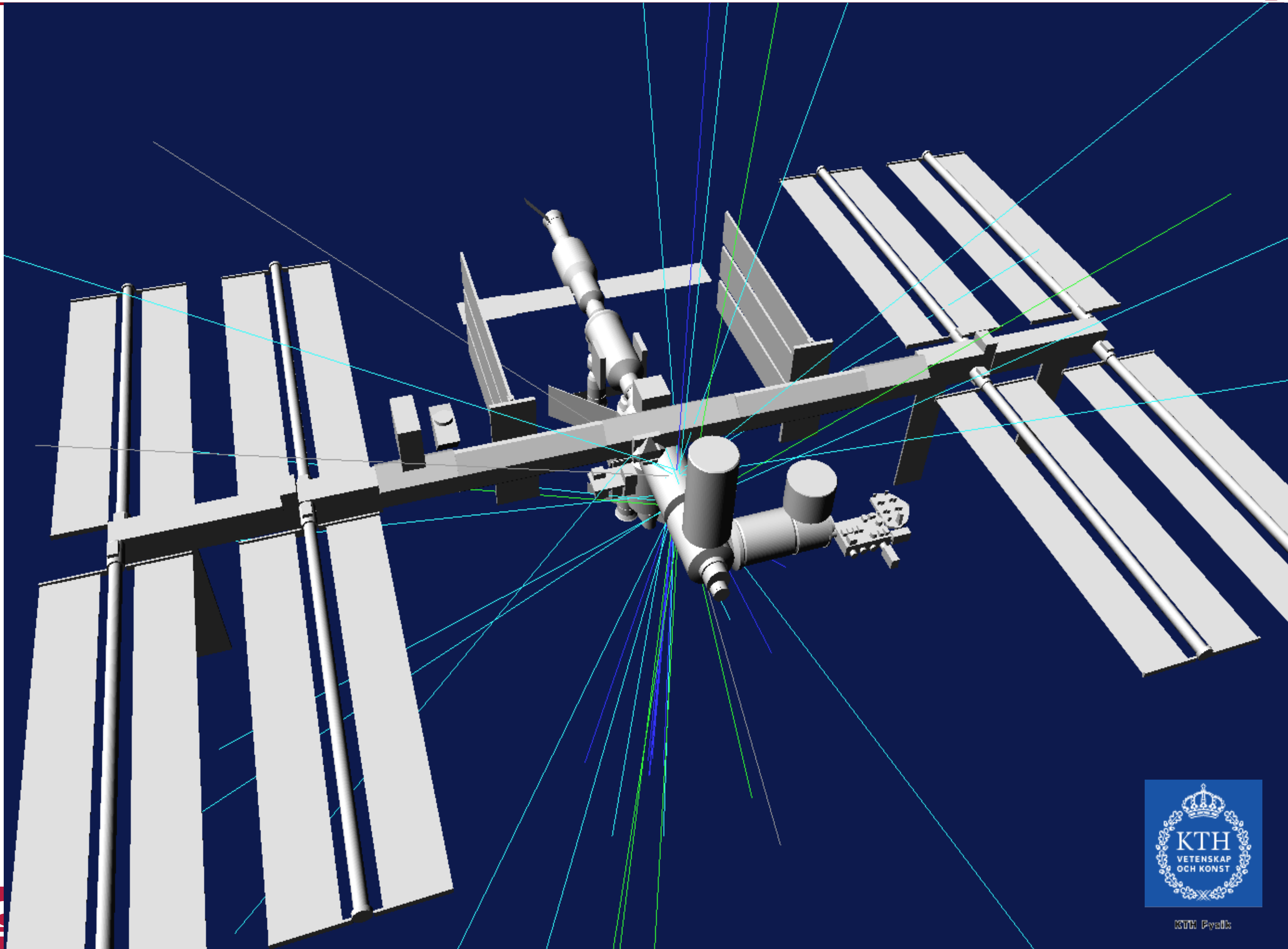
Anticoincidence

- γ conversion
- electron interactions
- multiple scattering
- δ-ray production
- charged particle tracking





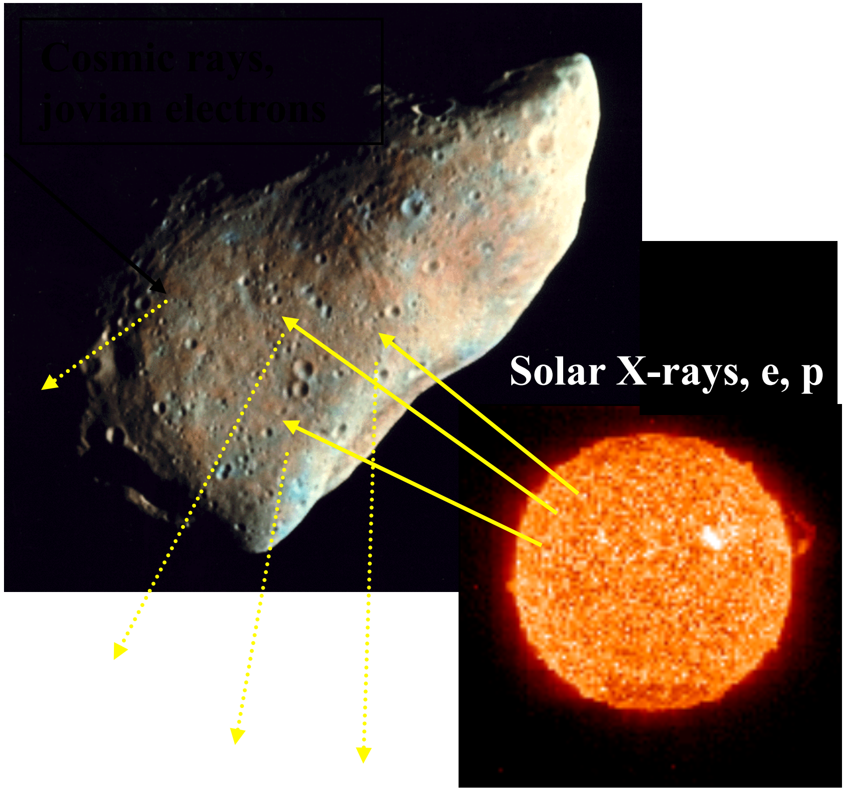

Smart-2
ACE
LISA
INTEGRAL
Cassini
Bepi Colombo
GLAST
Herschel
SWIFT
Astro-E2
XMM-Newton
GAIA
JWST
EUSO
ISS Columbus
AMS
MAXI



Courtesy T. Ersmark, KTH Stockholm

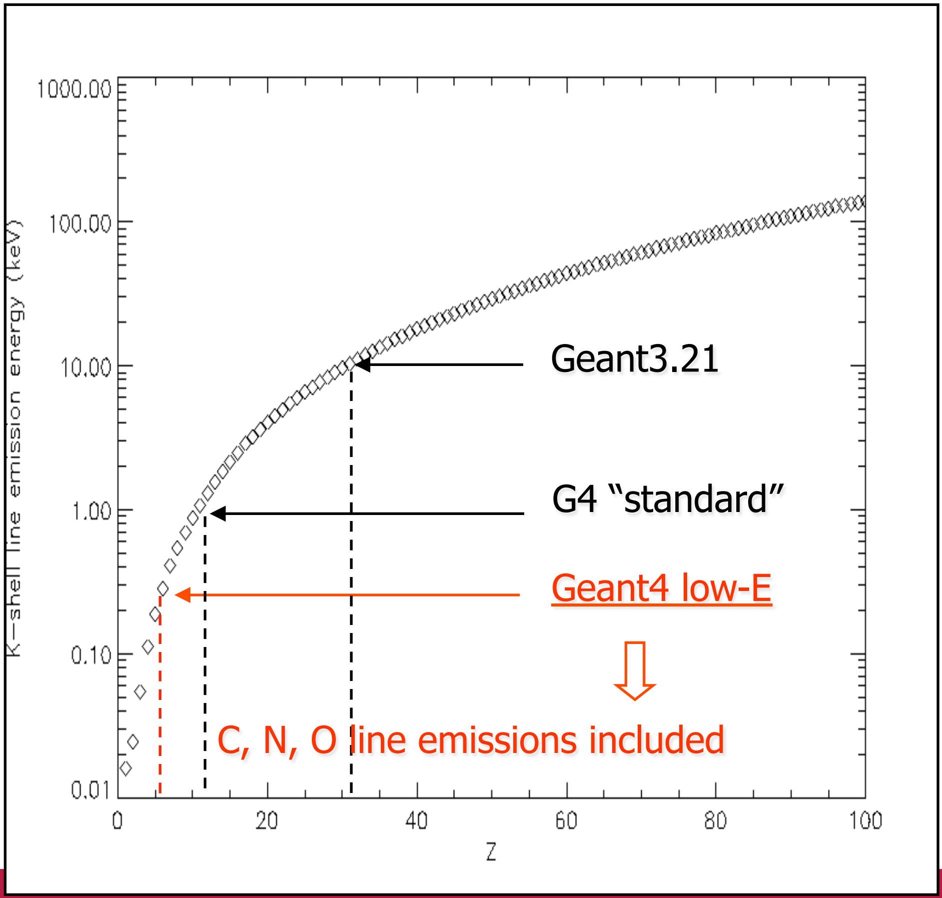
ESA Space Environment & Effects Analysis Section

X-Ray Surveys of Asteroids and Moons

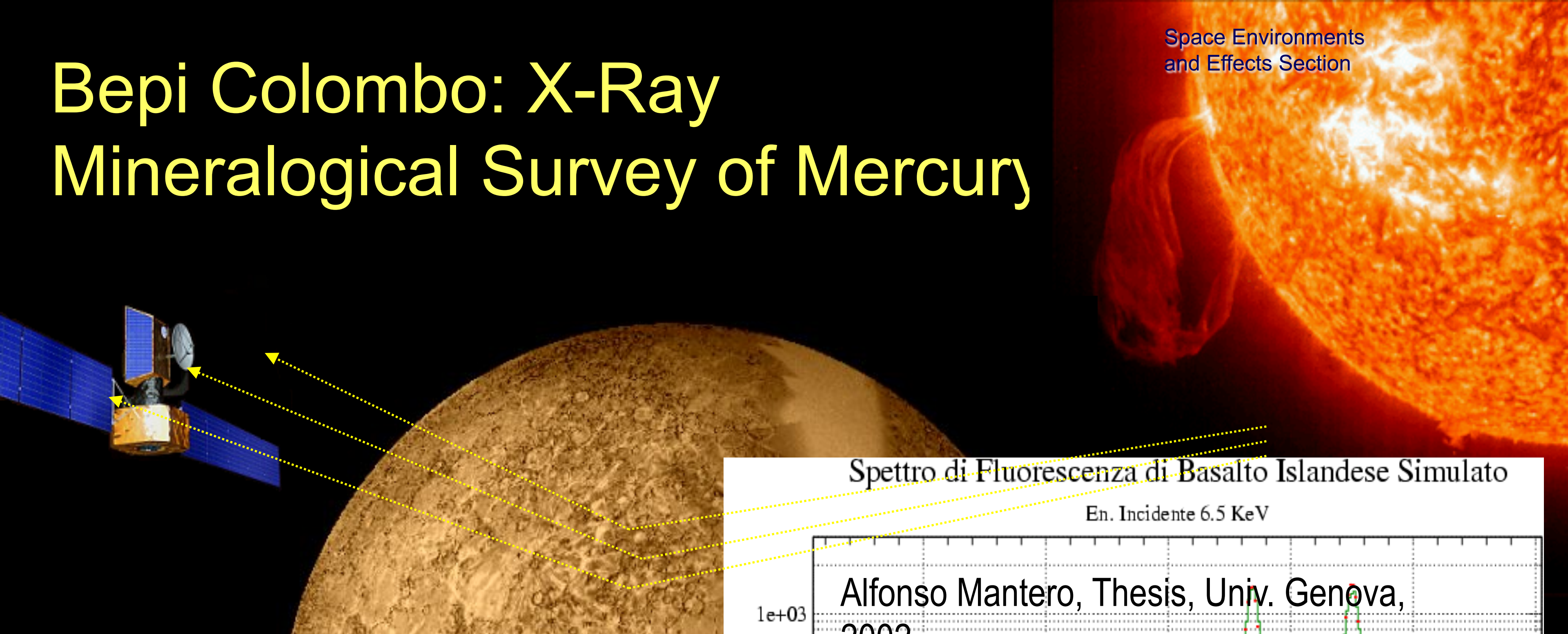


Courtesy SOHO EIT

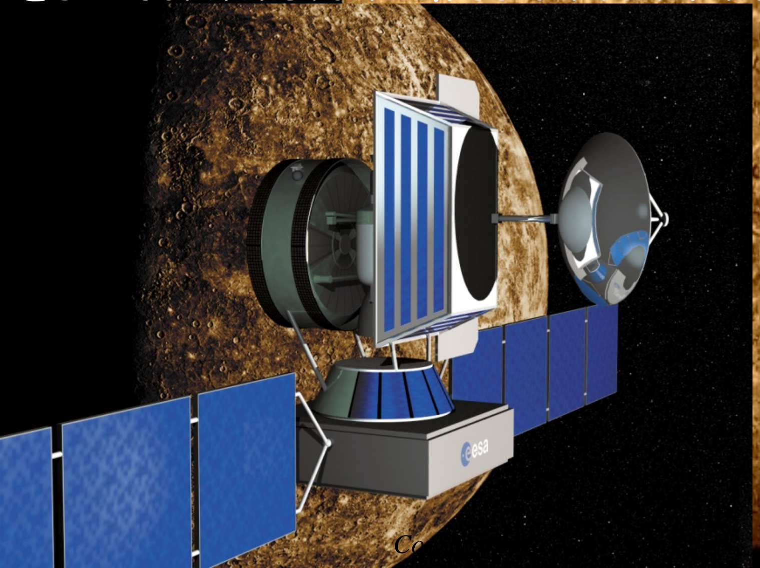
Induced X-ray line emission: indicator of target composition
(~100 μm surface layer)

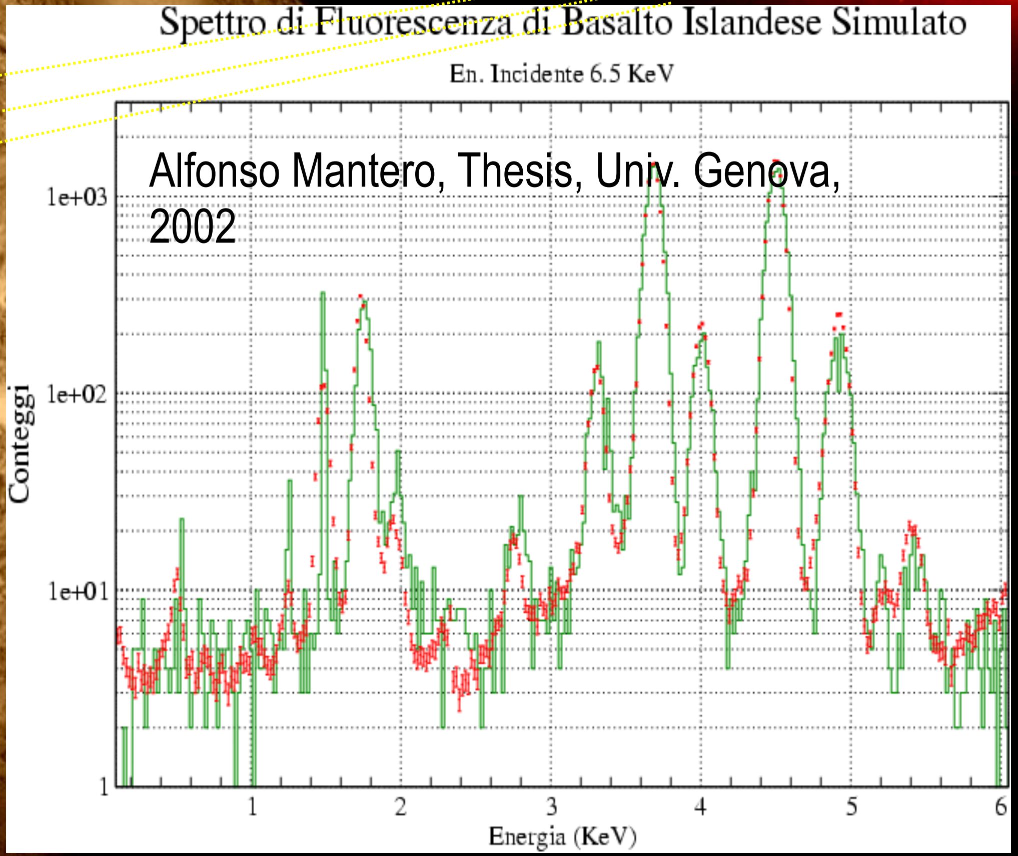


Bepi Colombo: X-Ray Mineralogical Survey of Mercury



BepiColombo
ESA cornerstone mission to Mercury





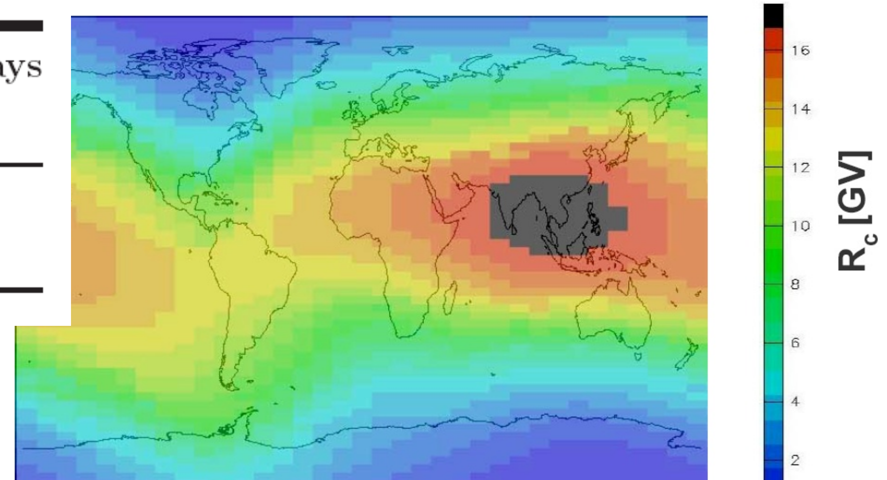
PlanetoCosmics
Geant4 simulation of Cosmic Rays in planetary Atmo-/Magneto- spheres

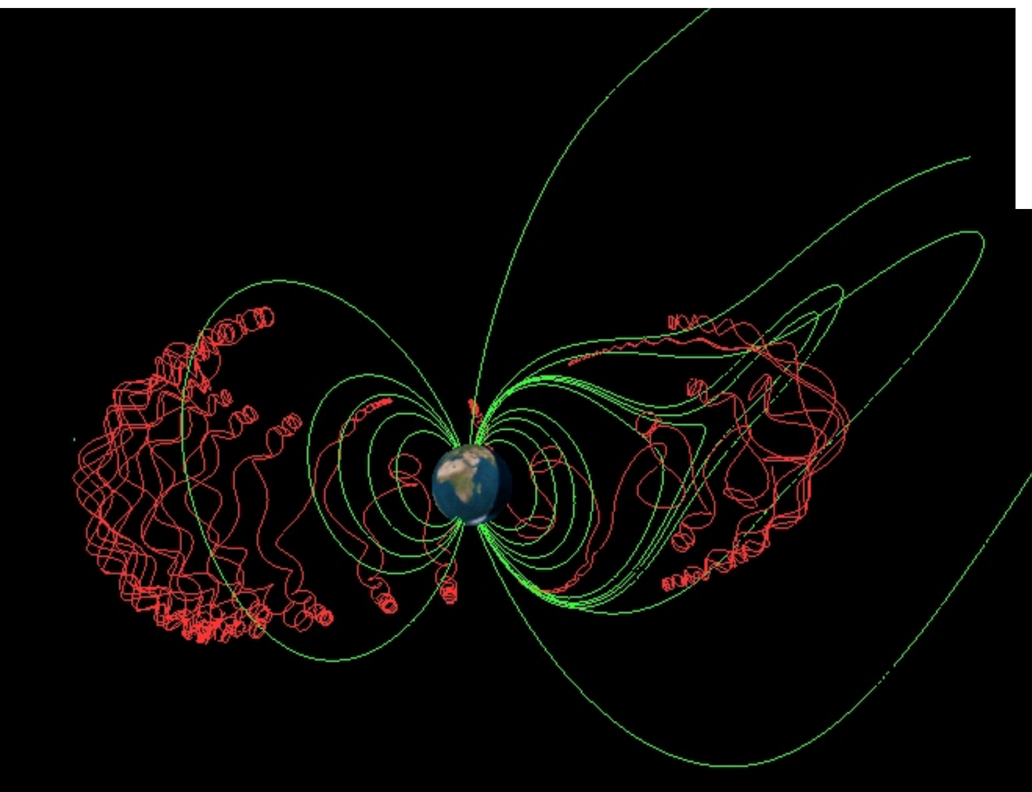
28th International Cosmic Ray Conference — 4277

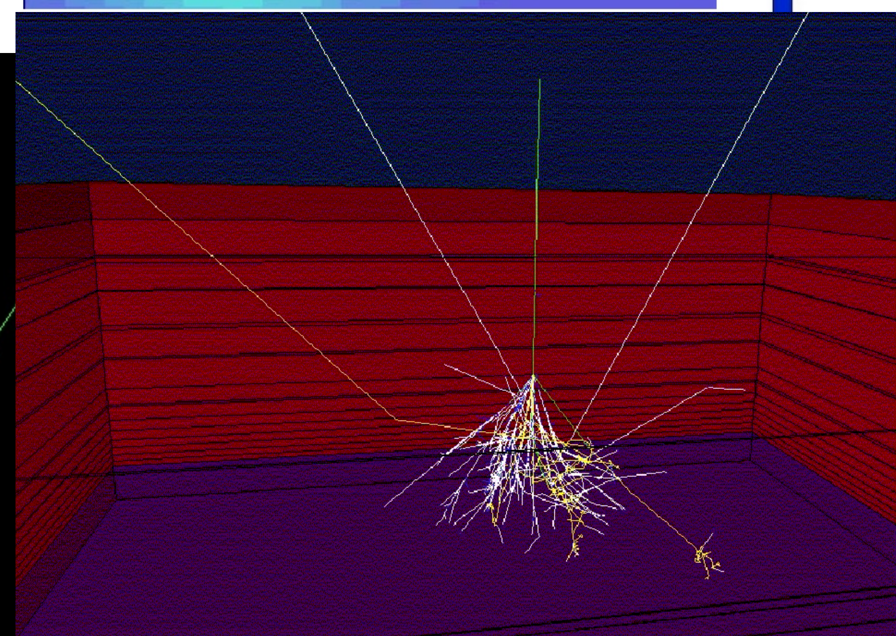
Geant4 Simulation of the Propagation of Cosmic Rays through the Earth's Atmosphere

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

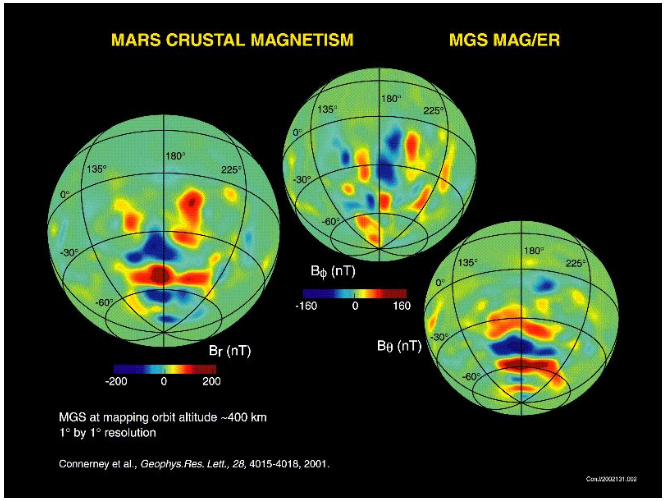
Cutoff Rigidities vs position

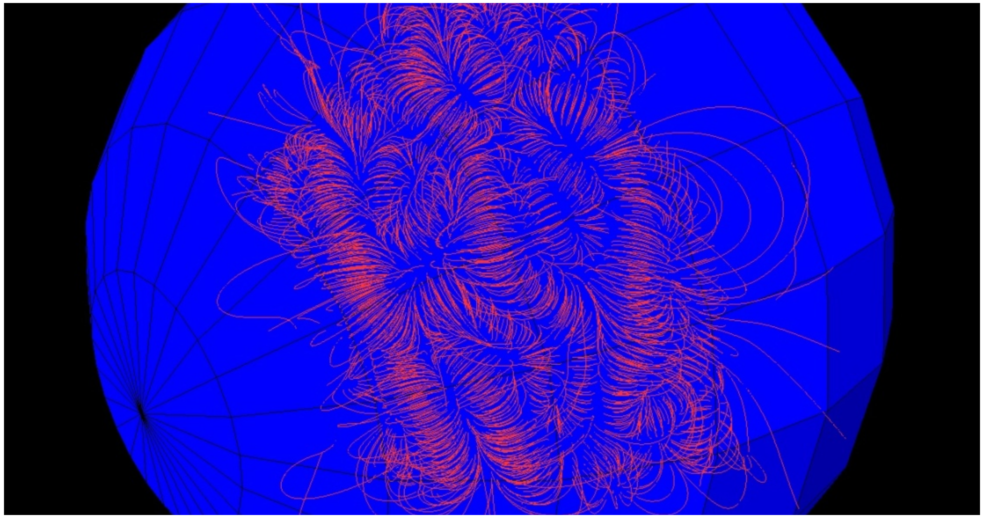




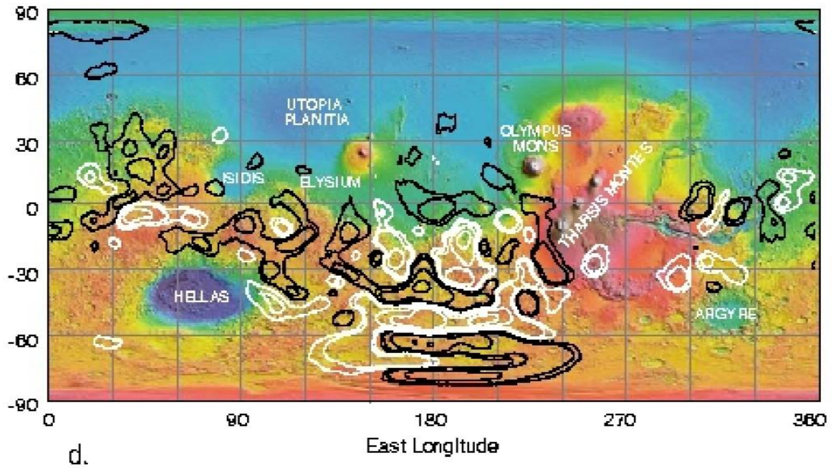


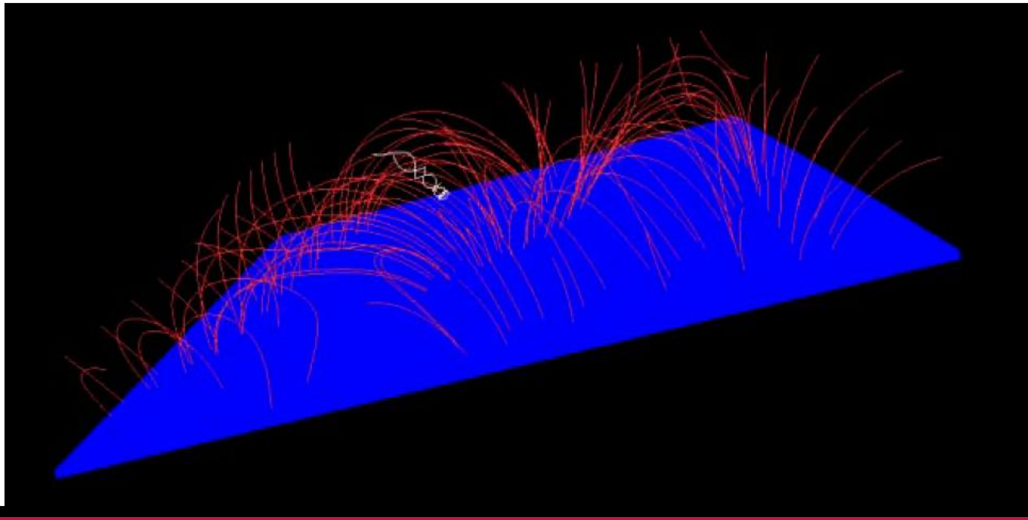
PlanetoCosmics
Mars field and atmosphere



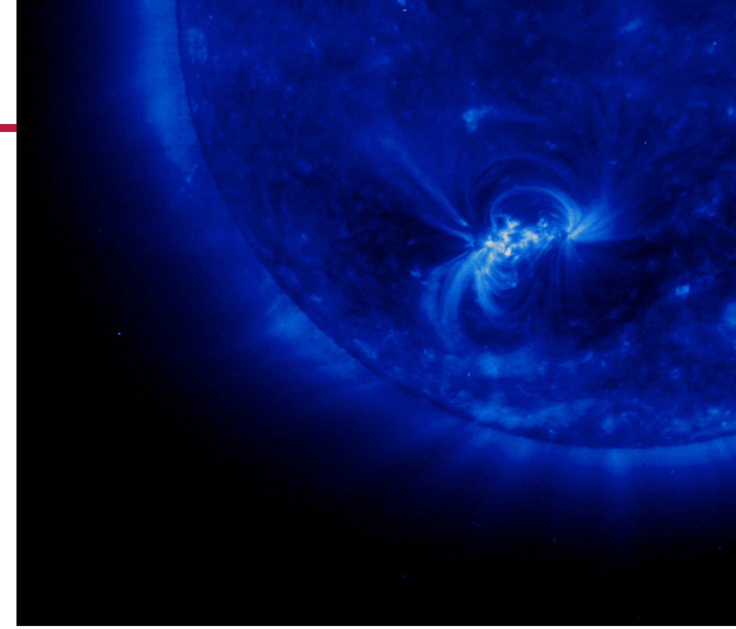


- NASA Mars-GRAM2001 model

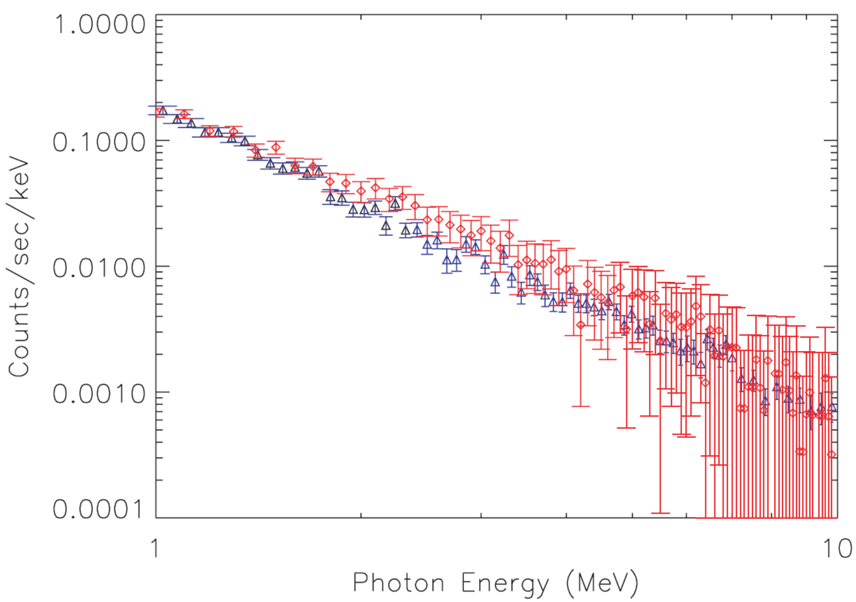




Solar event gamma-rays



- 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

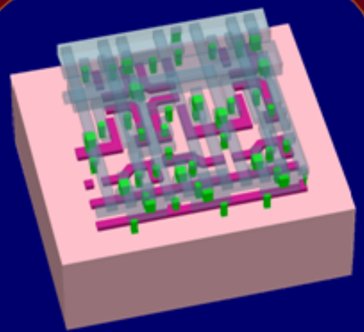
Jun'ichi Kotoku
National Astronomical Observatory of Japan, 2-21-1 Osawa, Mitaka, Tokyo 181-8588, JAPAN
junichi.kotoku@nao.ac.jp
Kazuo Makishima[1] and Yukari Matsumoto[2]
Department of Physics, University of Tokyo, Bunkyo-ku, Tokyo, 113-0022
and
Mitsuhiro Kohama, Yukikatsu Terada and Toru Tamagawa
RIKEN (Institute of Physical and Chemical research), Wako-shi, Saitama
[1] *Also at RIKEN*
[2] *Present address: Mitsubishi Electric Co., Ltd.*

(Received ; accepted)

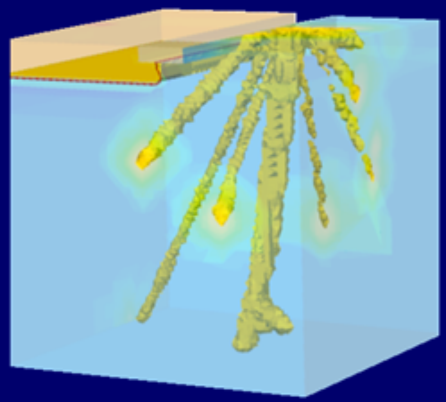
Abstract

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-

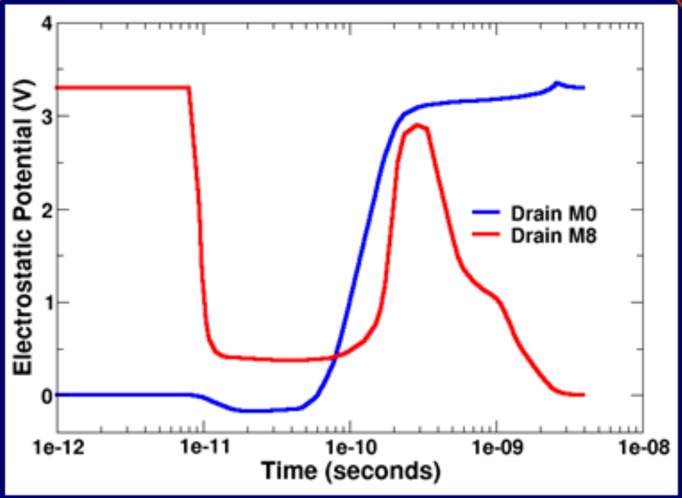
RADSAFE on SEE in SRAMs



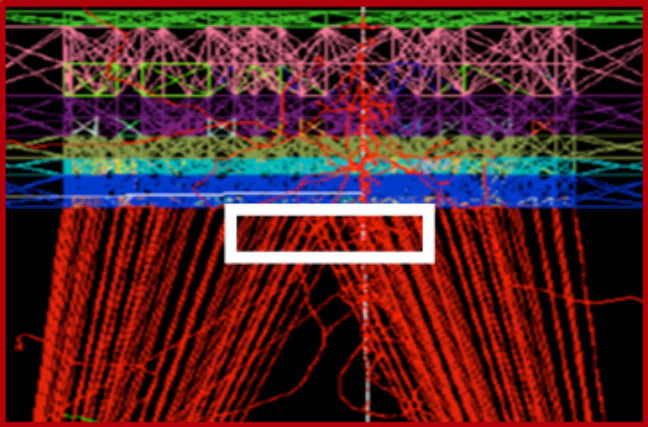
TCAD Cell Structure: SRAM Cell



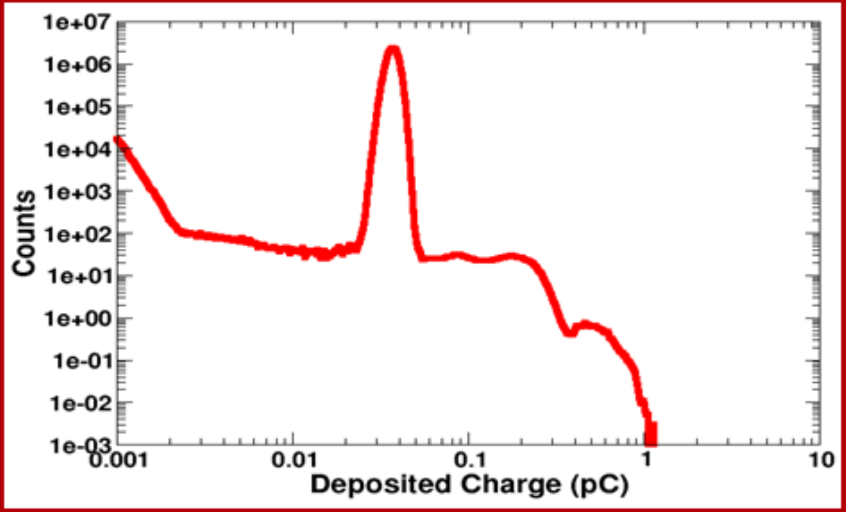
Single Charge Deposition in TCAD: Ne+W Event



SRAM Cell Upset



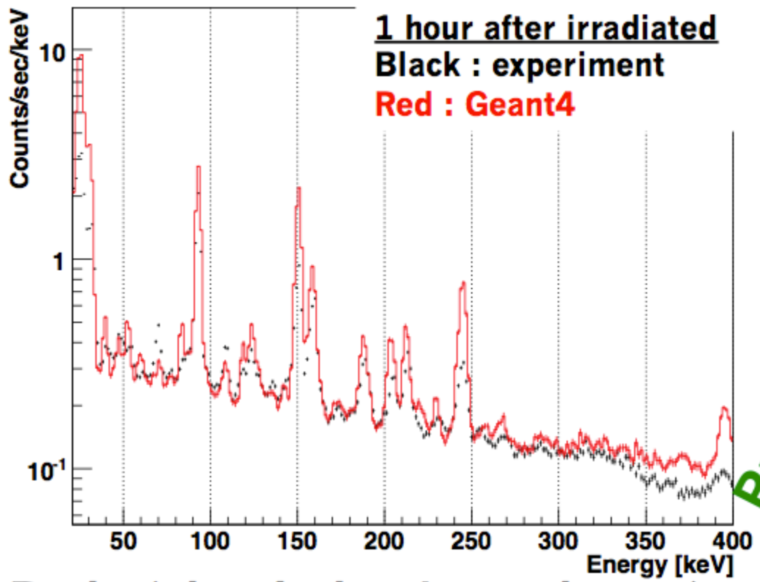
Geant4 Geometry and 523 MeV Neon Event

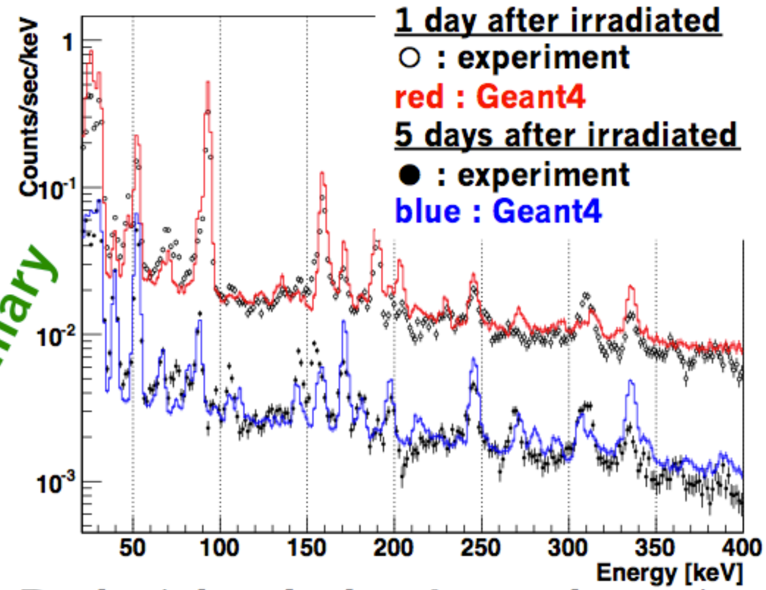


MRED Energy Deposition for 10^8 Events

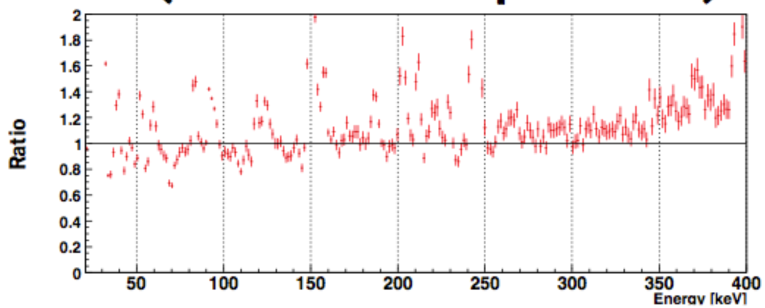
Time evolution of the activation background

Comparison with Geant4

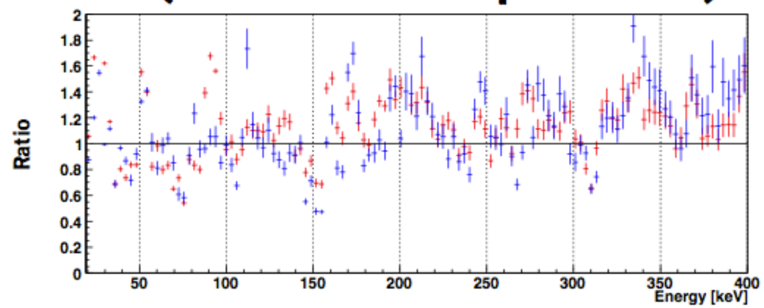




Ratio (simulation/experiment)

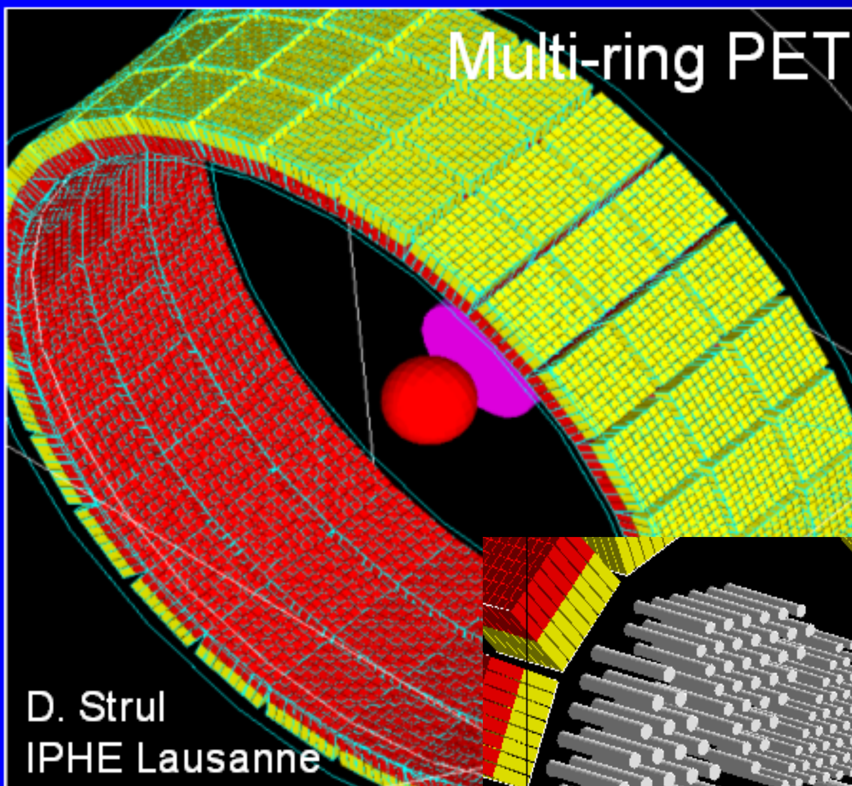


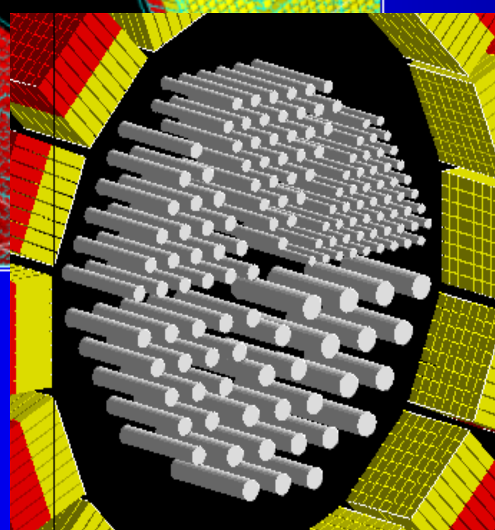
Ratio (simulation/experiment)

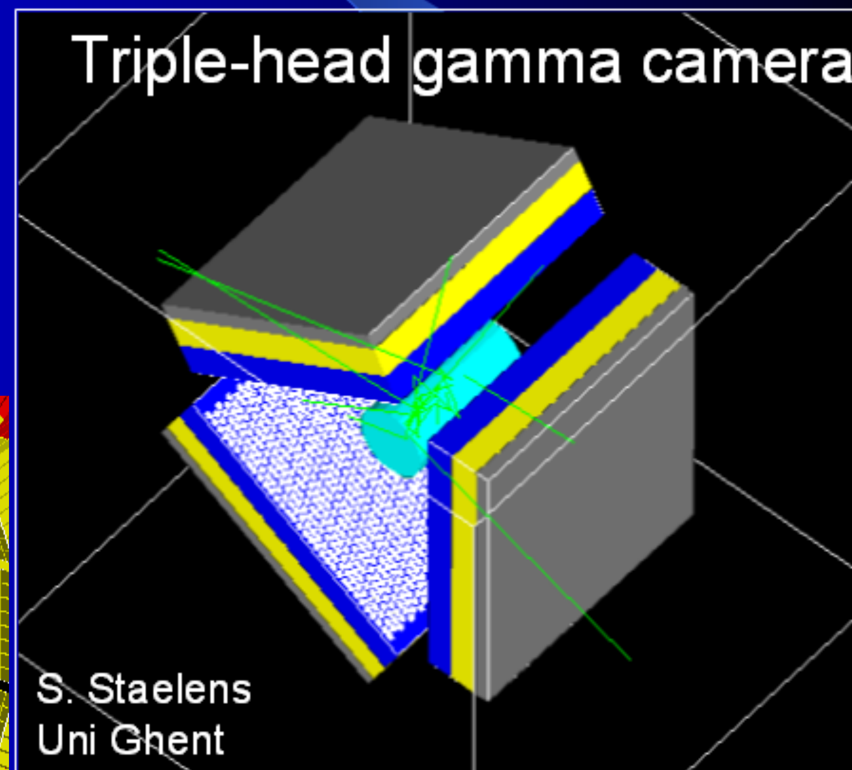


❖ Simulation results agrees with experimental data within a factor of two in terms of the line intensities

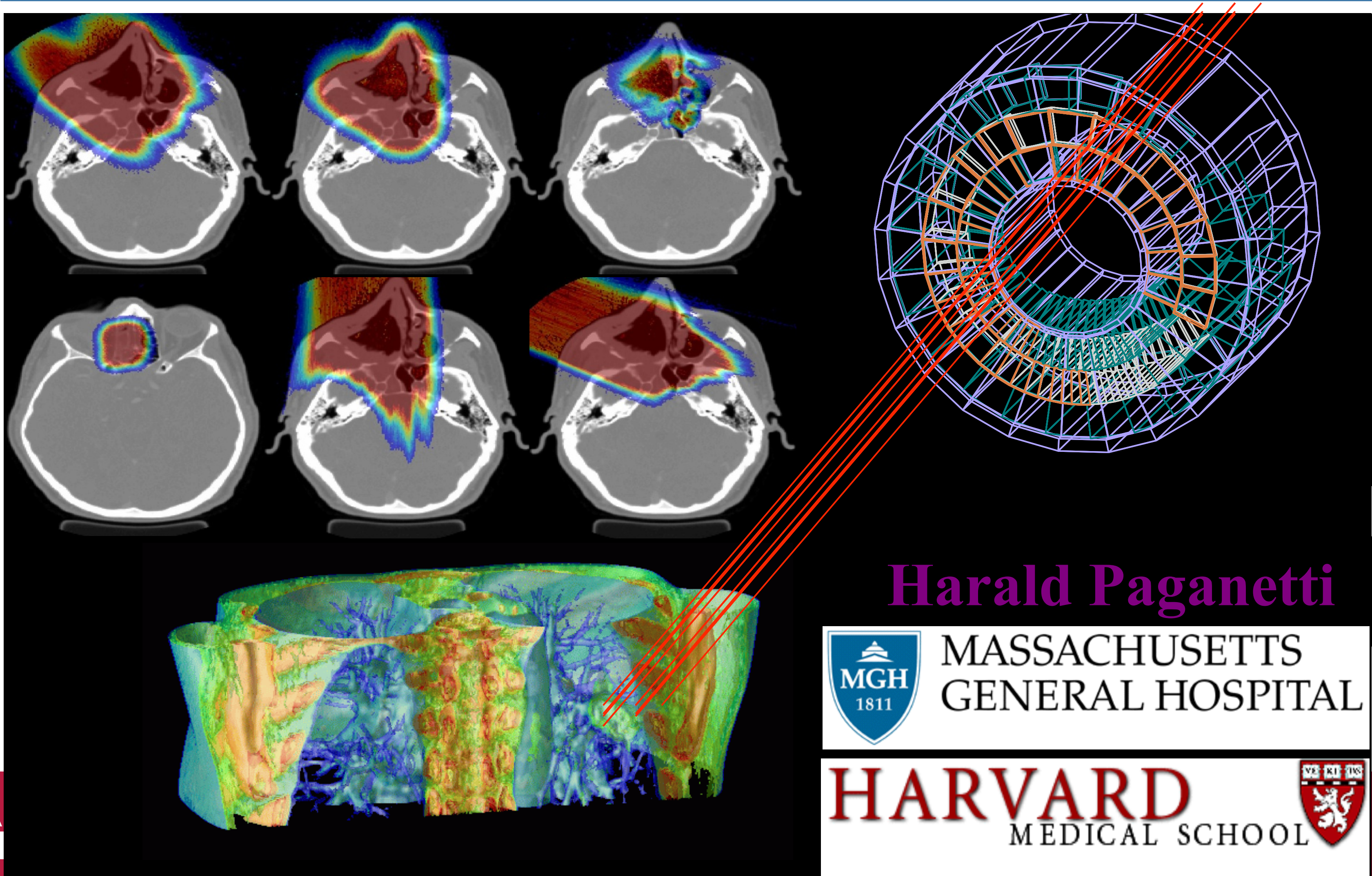
Geometry examples of GATE applications







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

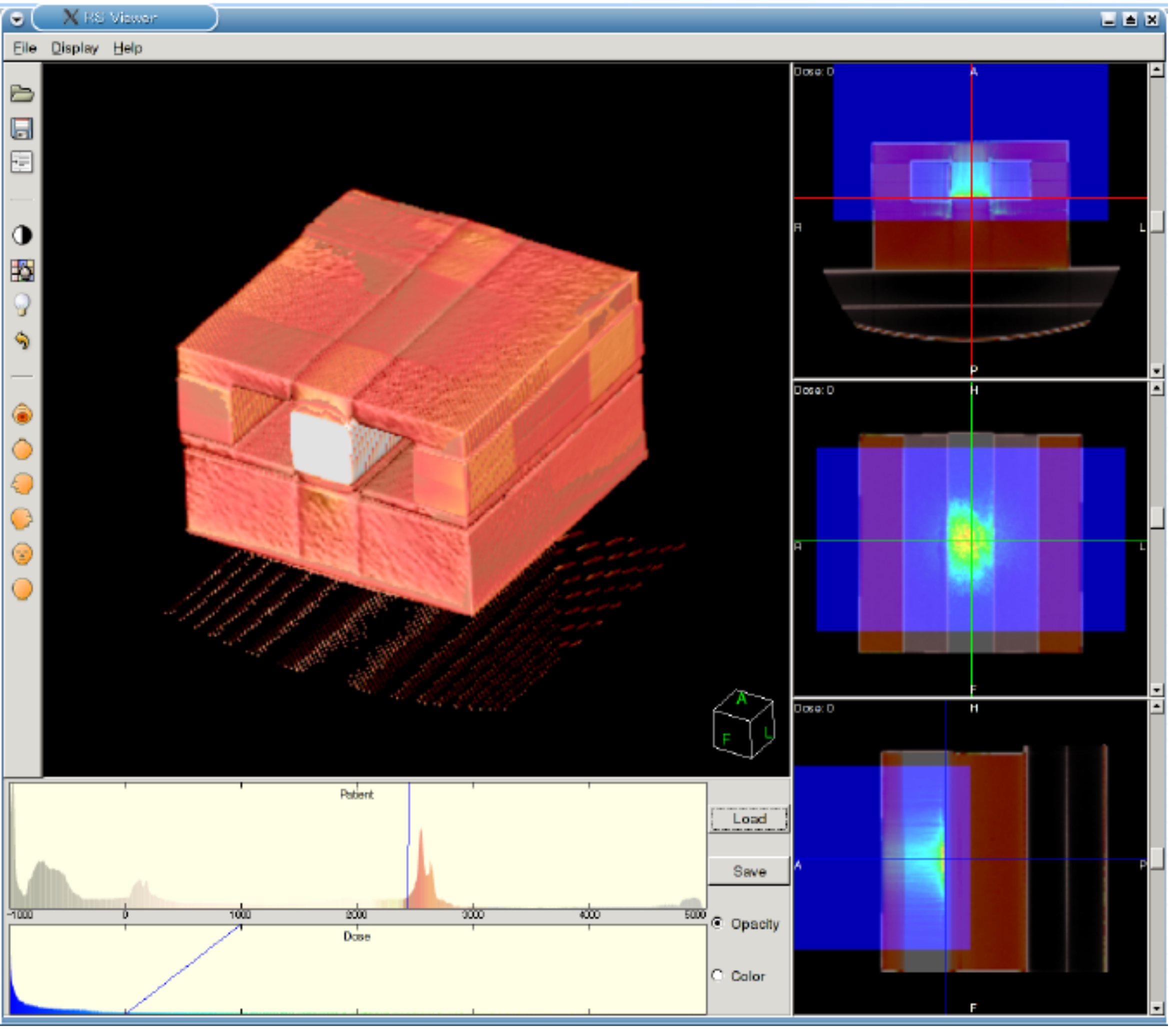


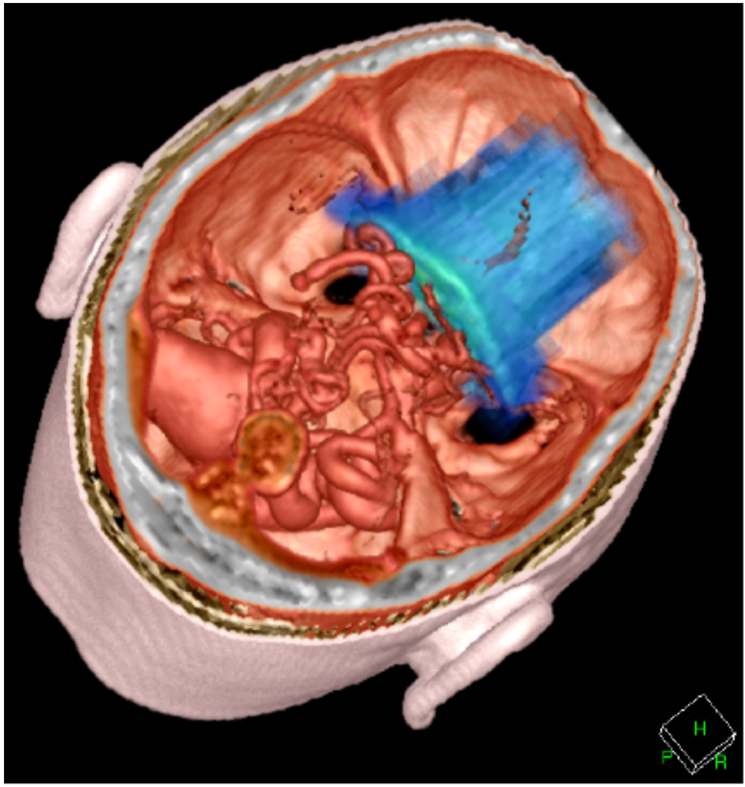
Harald Paganetti

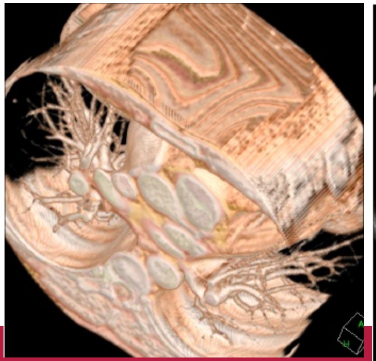
MASSACHUSETTS GENERAL HOSPITAL

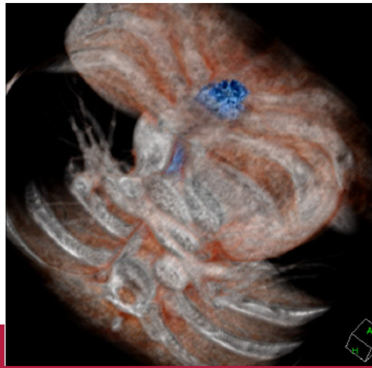
HARVARD MEDICAL SCHOOL

Screen shots of gMocren









Geant 4

Version 10.0-p01

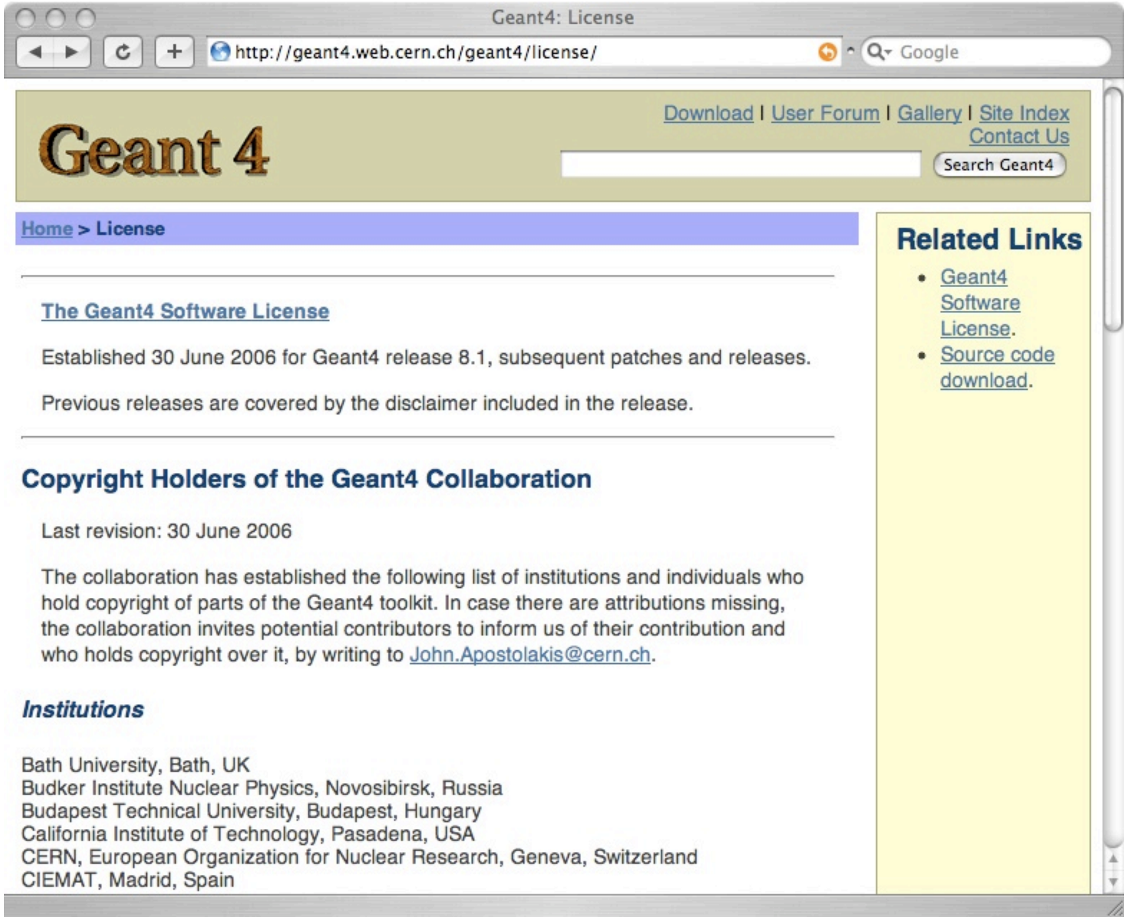
Geant4 license

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.

Geant 4

Version 10.0-p01

Basic concepts
and kernel structure

Terminology (jargons)

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

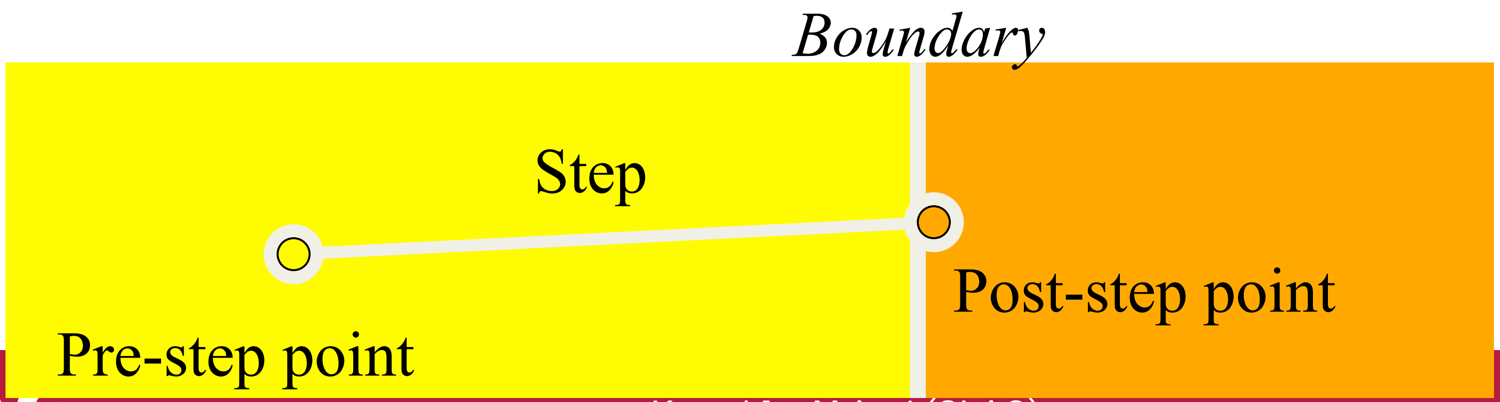
- An event is the basic unit of simulation in Geant4.
- At beginning of processing, primary tracks are generated. These primary tracks are pushed into a stack.
- A track is popped up from the stack one by one and "tracked". Resulting secondary tracks are pushed into the stack.
  - This "tracking" lasts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

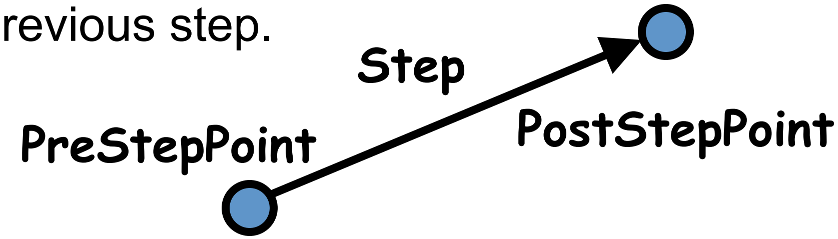
- A Cut in Geant4 is a production threshold.
  - Not tracking cut, which does not exist in Geant4 as default.
    - All tracks are traced down to zero kinetic energy.
  - It is applied only for physics processes that have infrared divergence
- Much detail will be given at later talks on physics.

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

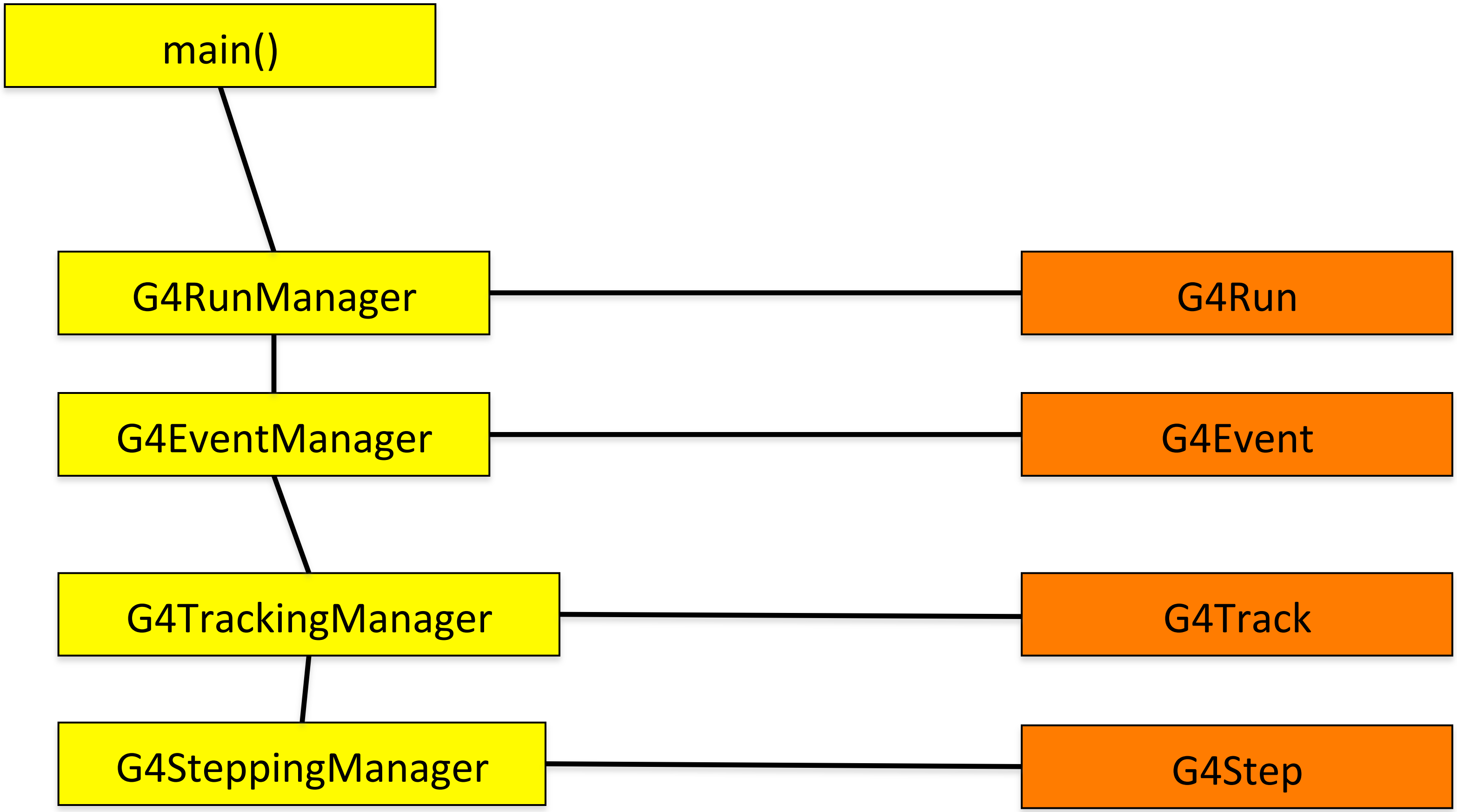
- Step status is attached to G4StepPoint to indicate why that particular step was determined.
  - Use "PostStepPoint" to get the status of this step.
  - "PreStepPoint" has the status of the previous step.



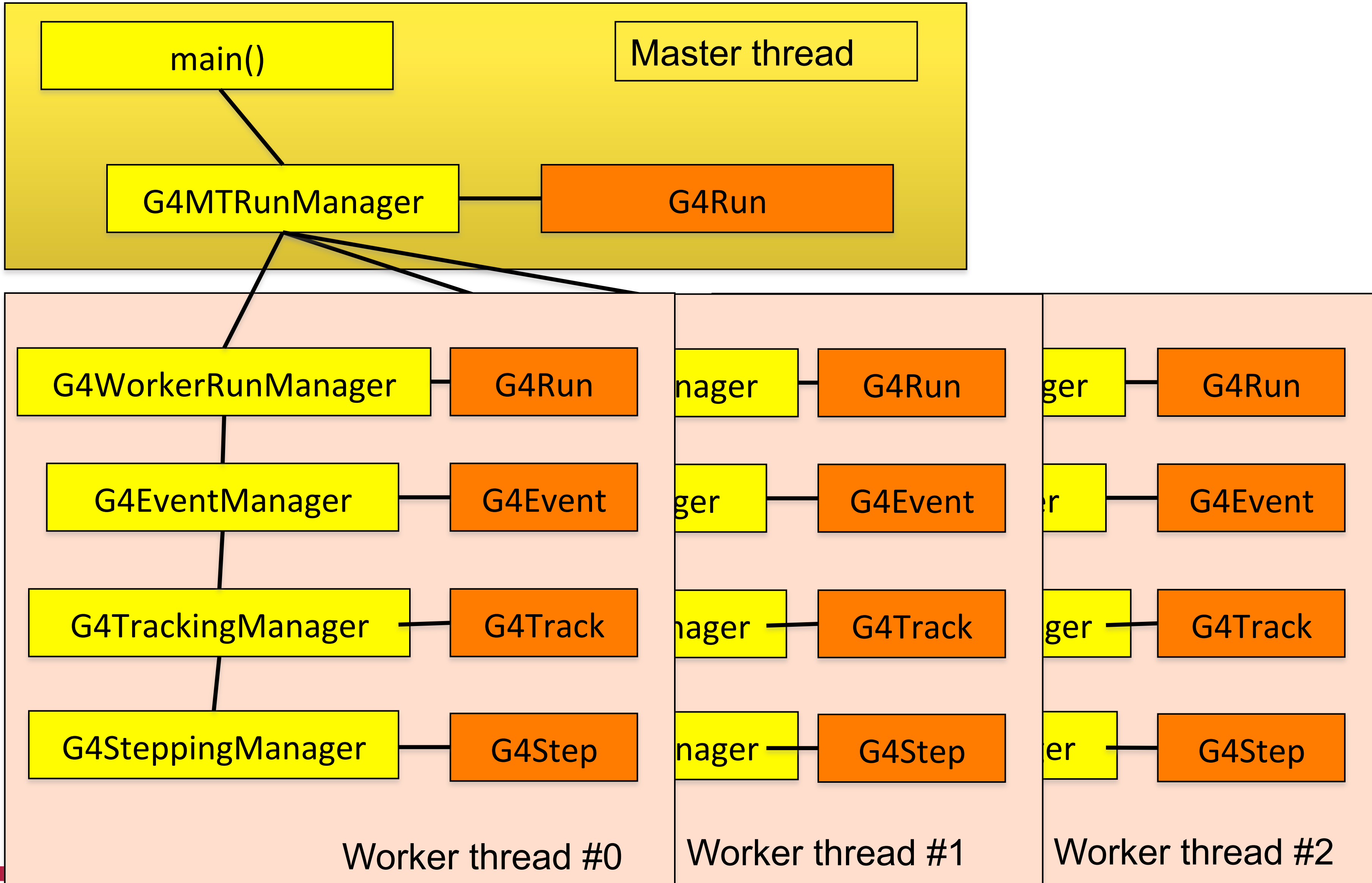
- fWorldBoundary
  - Step reached the world boundary
- fGeomBoundary
  - Step is limited by a volume boundary except the world
- fAtRestDoItProc, fAlongStepDoItProc, fPostStepDoItProc
  - Step is limited by a AtRest, AlongStep or PostStep process
- fUserDefinedLimit
  - Step is limited by the user Step 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.

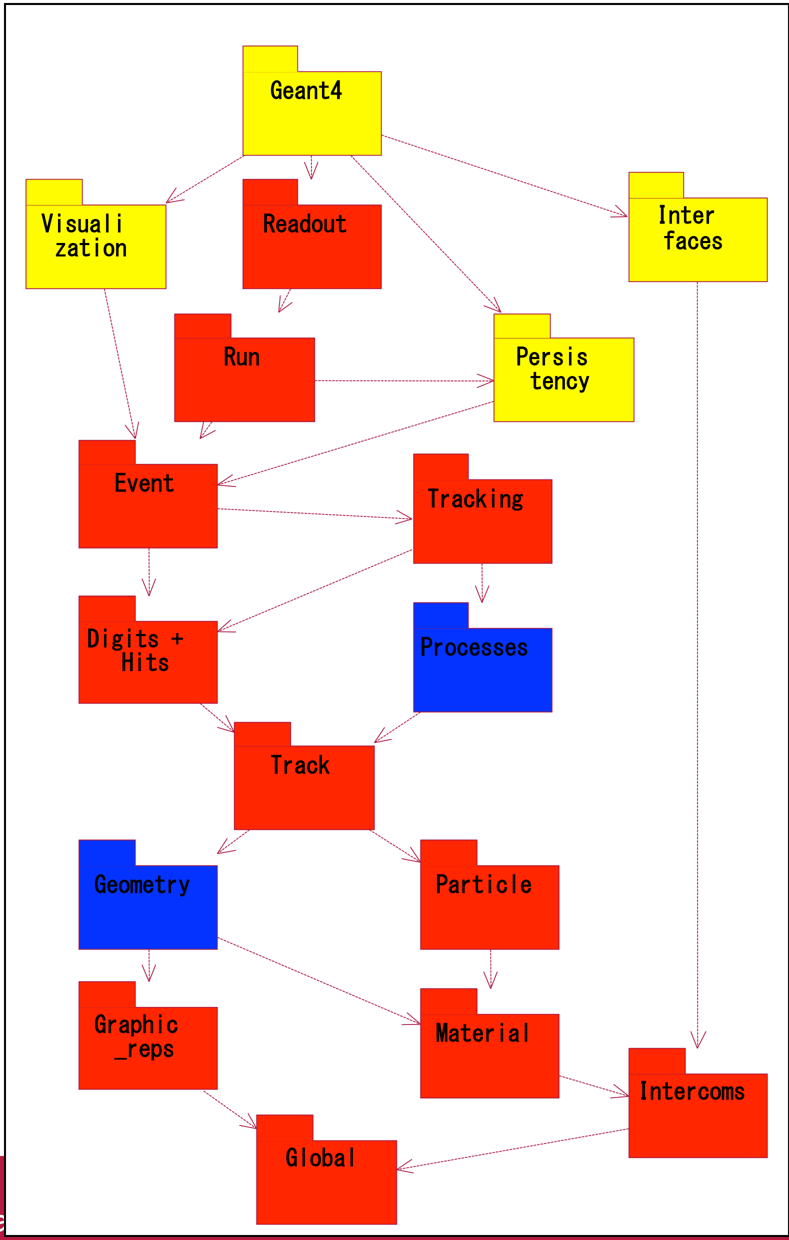
  /control/cout/ignoreThreadsExcept <threadID>
  - Omit cout from worker threads except the specified one.
  - If specified thread ID is greater than the number of threads, no cout is displayed from worker threads. -1 to reset.

  /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**".

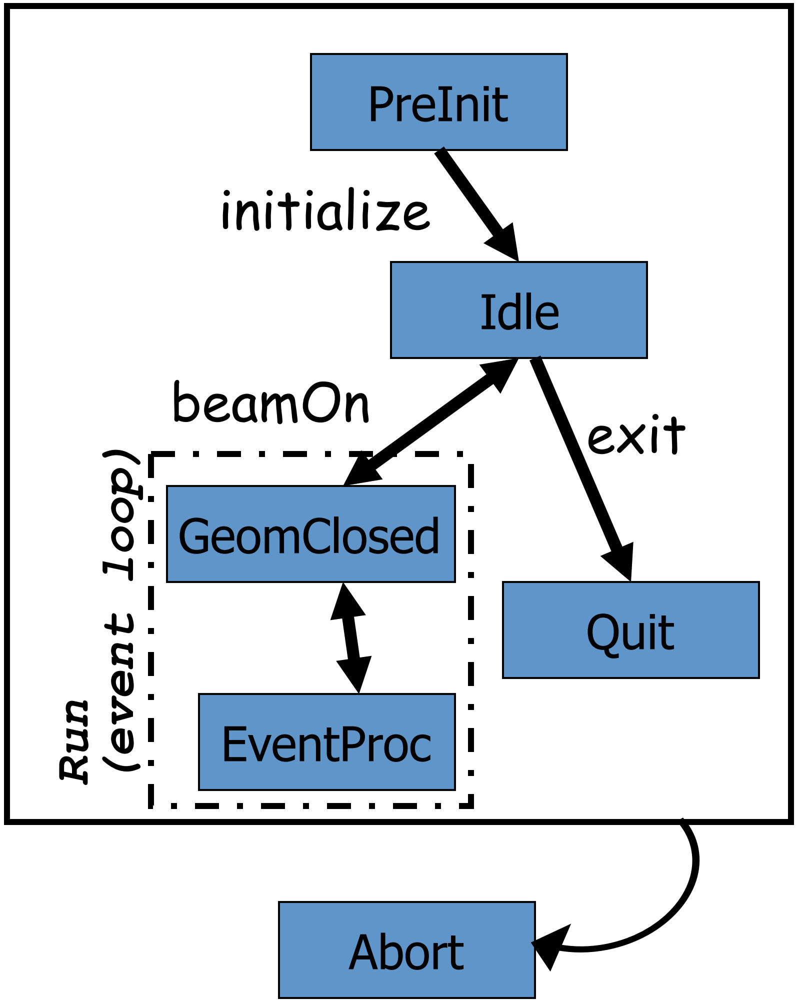
Geant4 kernel

- 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 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



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

Geant 4

Version 10.0-p01

User classes

NATIONAL ACCELERATOR LABORATORY

U.S. DEPARTMENT OF ENERGY

Office of Science

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

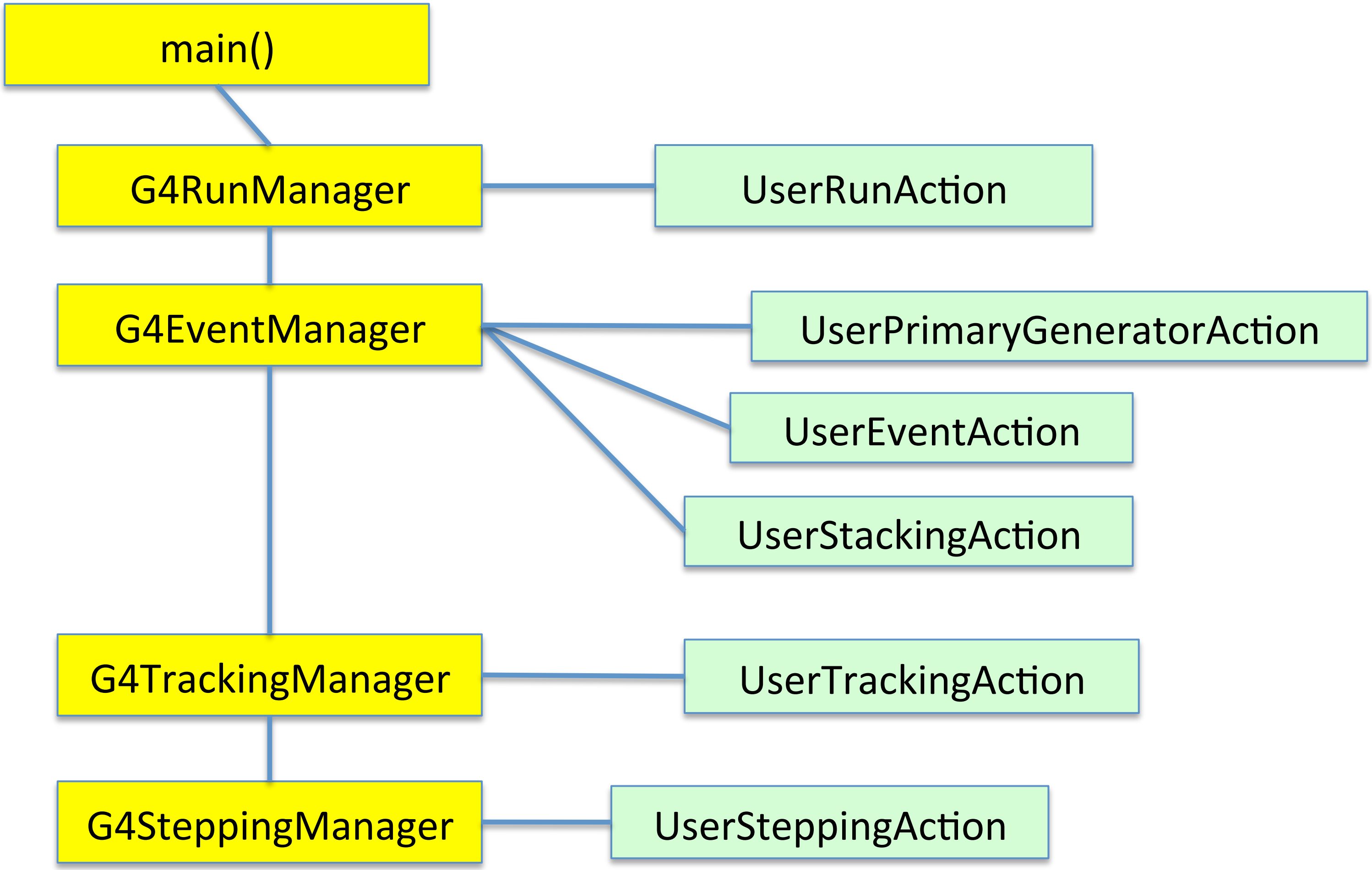
Optional user action classes

- 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)

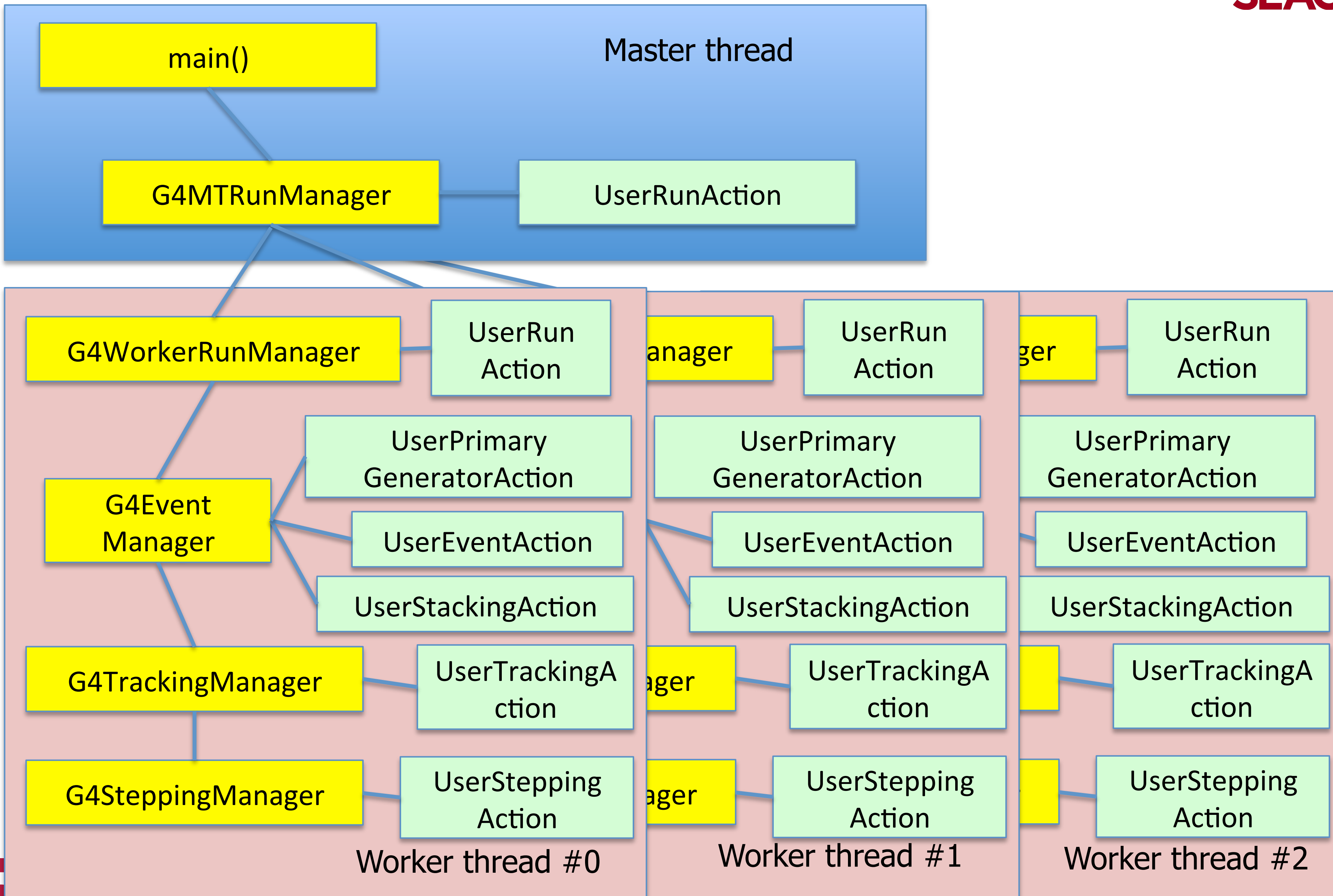
Optional user action classes

- 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