Geant4 Toolkit: an introduction

Andrea Dotti (adotti@slac.stanford.edu)

University and INFN Ferrara, Friday 9 May 2014

U.S. DEPARTMENT OF ENERGY Office of Science

http://www.geant4.org

SLAC NATIONAL ACCELERATOR LABORATORY

What will we review in the next hour

- Geant4 is a **toolkit** for the development of your **own simulation** of the interaction of radiation with matter
- We cannot cover a Geant4 tutorial in one hour, the idea here is to give a "taste" of what writing an application for G4 means

Stating the physics problem

The study of the interaction of radiation (e.g. particles, x-rays) with matter has applications in several scientific areas:

- Basic research (e.g. at accelerators to discover new phenomena)
- Medical imaging (e.g. x-rays)
- Medical treatment (e.g. radio-therapy)
- Industrial (e.g. energy production, shielding)

Essential tools for these studies are simulation programs. The most precise are based on Monte Carlo techniques

Several codes exists: Geant4 is one of them

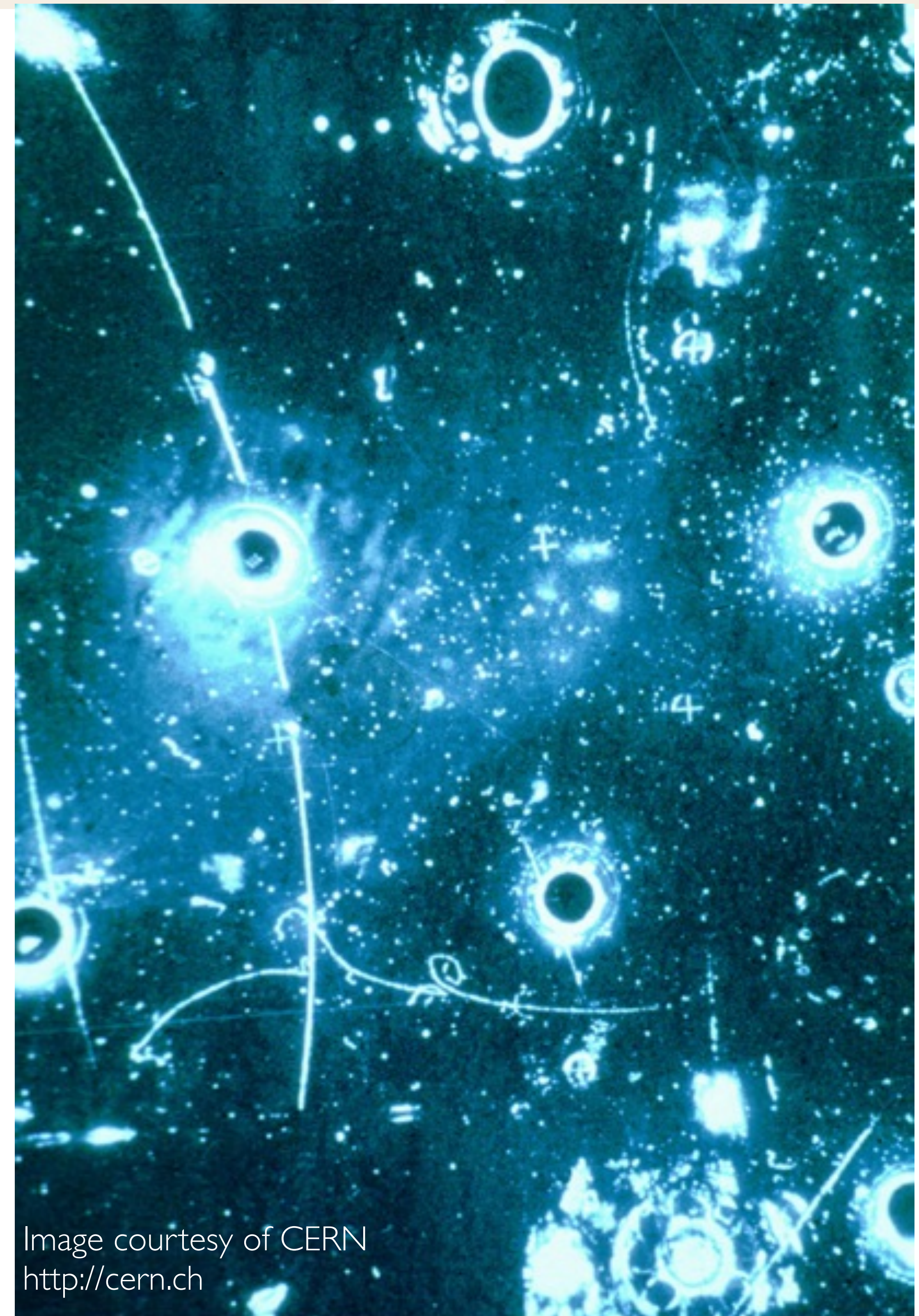


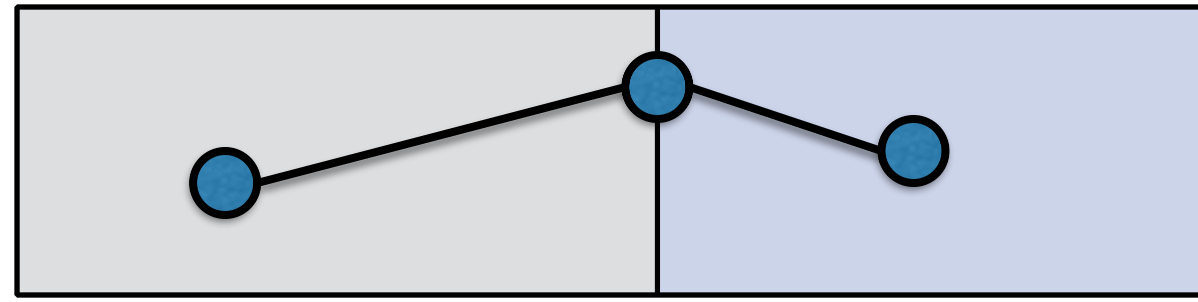
Image courtesy of CERN
http://cern.ch

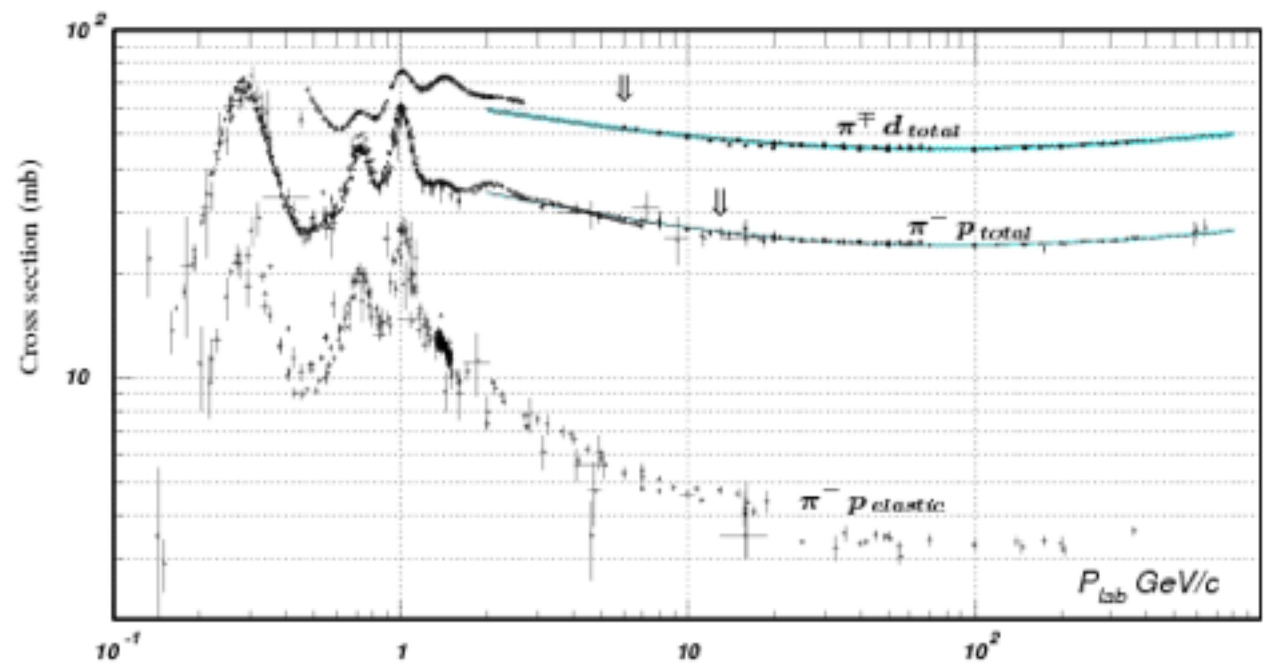
This event shows the real tracks produced in the 1200 litre Gargamelle bubble chamber that provided the first confirmation of a neutral current interaction. A neutrino interacts with an electron, the track of which is seen horizontally, and emerges as a neutrino without producing a muon. The discovery of the neutral current was announced in the CERN main auditorium in July 1973

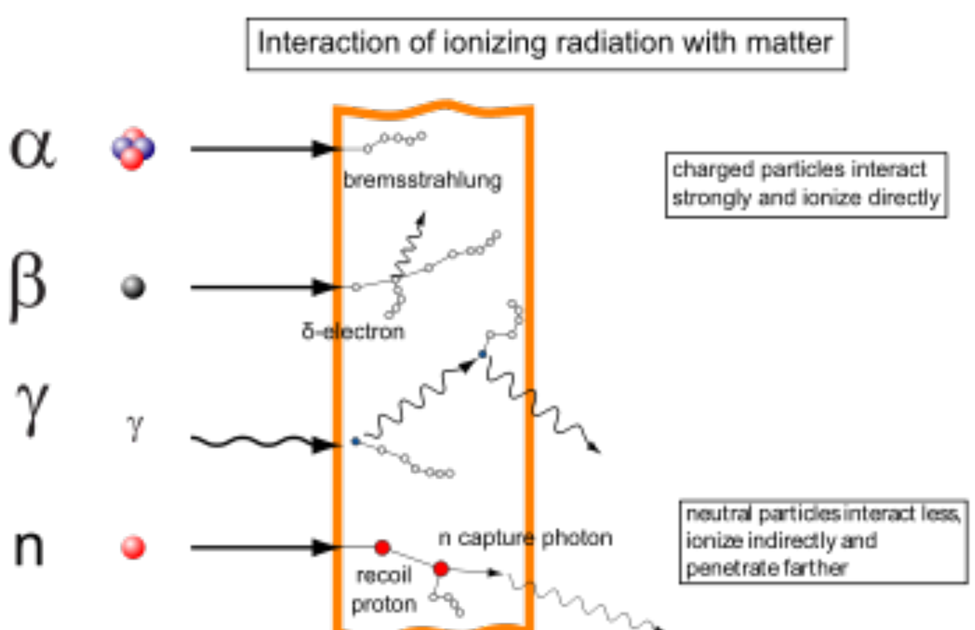
What is Geant4

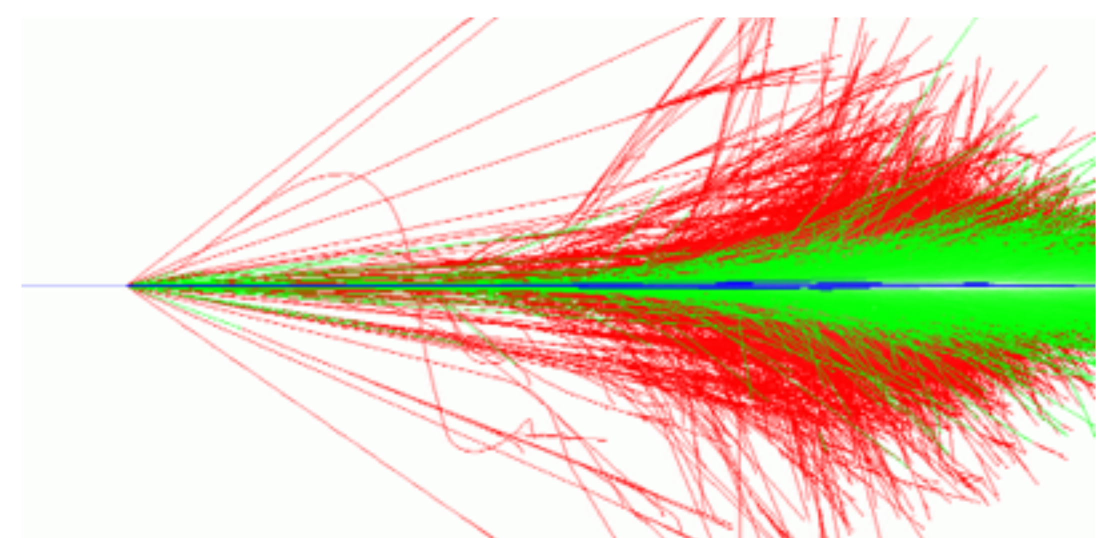
Geant4 design

- A particle is **tracked** in steps in the geometry setup
  - Electro-magnetic fields, if present, bend the particle trajectories
- At each step one or more **physics processes** are applied (e.g. ionization, nuclear interactions)
  - Each particle has its own set of possible processes
- The **probability** of any given process to occur is driven by the cross-section
- **Secondary particles** may be produced in this interactions
  - A "cascade" can be produced





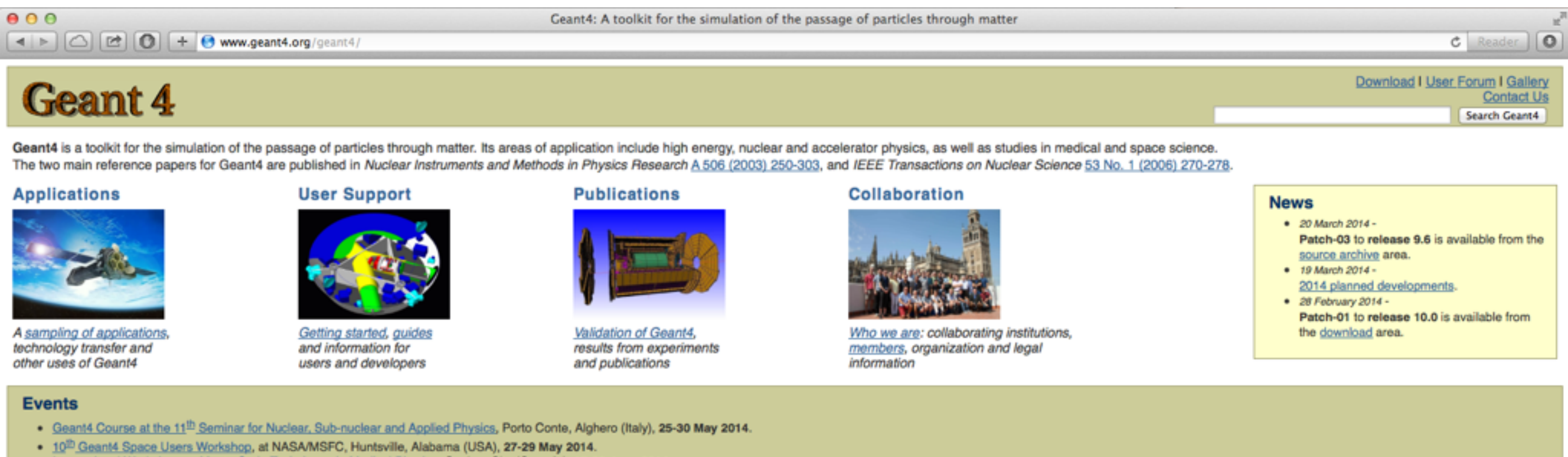




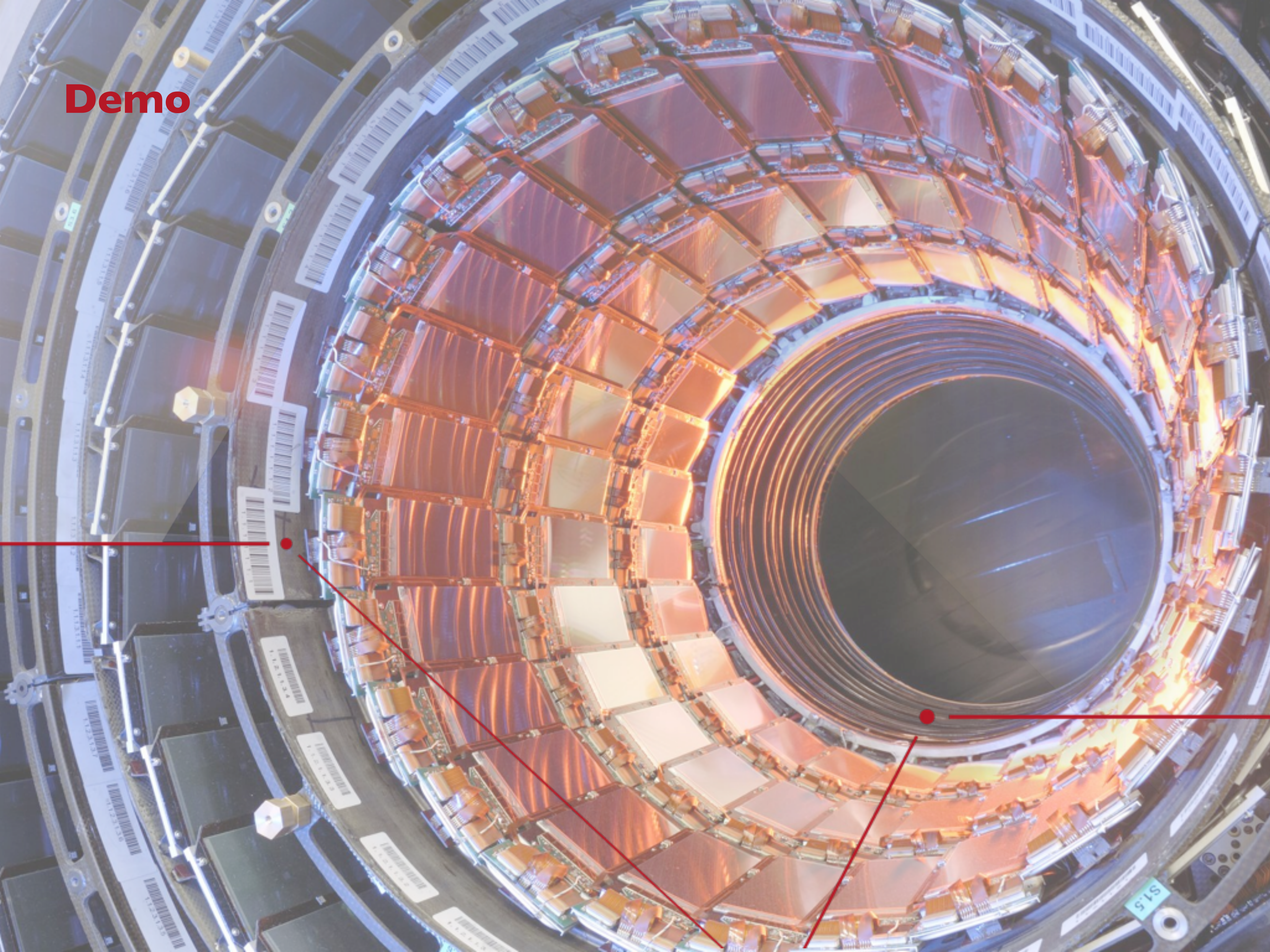
Geant4 Code

- **Open-source** C++ project (~800k LOC)
- International Collaboration (mainly physicists)
- R&D started in 1994, arrived at version 10.0
- Currently available for Linux, Mac, Win OS
- **Geant4 is a toolkit**:
  - Set of libraries on top of which a user creates an application
  - Several examples available
- **Complete**: geometry modeling, physics processes, visualization, scoring, analysis
- Estimated user base O(10k) users, O(100) applications

http://geant4.org



Demo



Let's take a look at a simple application

- What you see now is an example coming with Geant4 re-engineered for our tutorials
- Slides and exercises are available at:
  - http://geant4.slac.stanford.edu/SLACTutorial14/Agenda.html

Anatomy of an application (simplified)

SLAC

```
int main(int argc,char** argv)
{
  // Construct the default run manager
  //
#ifdef G4MULTITHREADED
    G4MTRunManager* runManager = new G4MTRunManager;
    runManager->SetNumberOfThreads(G4Threading::G4GetNumberOfCores());
#else
    G4RunManager* runManager = new G4RunManager;
#endif

    // Geometry and Setup
    runManager->SetUserInitialization(new DetectorConstruction);

    // Physics List
    runManager->SetUserInitialization(new FTFP_BERT);

    // User action initialization: primary generation, results extraction
    runManager->SetUserInitialization(new ActionInitialization());

    // Get the pointer to the User Interface manager
    G4UImanager* UImanager = G4UImanager::GetUIpointer();

    if (argc>1) {
        // execute an argument macro file if exist
        G4String command = "/control/execute ";
        G4String fileName = argv[1];
        UImanager->ApplyCommand(command+fileName);
    }
    else {
       // start interactive session
       G4UIExecutive* ui = new G4UIExecutive(argc, argv);
       ui->SessionStart();
       delete ui;
   }

    // Job termination
   delete runManager;
    return 0;
}
```

Anatomy of an application (simplified)

```
int main(int argc,char** argv)
{
  // Construct the default run manager
  //
#ifdef G4MULTITHREADED
    G4MTRunManager* runManager = new G4MTRunManager;
    runManager->SetNumberOfThreads(G4Threading::G4GetNumberOfCores());
#else
    G4RunManager* runManager = new G4RunManager;
#endif

    // Geometry and Setup
    runManager->SetUserInitialization(new DetectorConstruction);

    // Physics List
    runManager->SetUserInitialization(new FTFP_BERT);

    // User action initialization: primary generation, results extraction
    runManager->SetUserInitialization(new ActionInitialization());

    // Get the pointer to the User Interface manager
    G4UImanager* UImanager = G4UImanager::GetUIpointer();

    if (argc>1) {
        // execute an argument macro file if exist
        G4String command = "/control/execute ";
        G4String fileName = argv[1];
        UImanager->ApplyCommand(command+fileName);
    }
    else {
       // start interactive session
       G4UIExecutive* ui = new G4UIExecutive(argc, argv);
       ui->SessionStart();
       delete ui;
   }

    // Job termination
   delete runManager;
    return 0;
}
```

Instantiate Geant4 RunManager
If built supports MT, set default number of threads

Anatomy of an application (simplified)

```
int main(int argc,char** argv)
{
  // Construct the default run manager
  //
#ifdef G4MULTITHREADED
    G4MTRunManager* runManager = new G4MTRunManager;
    runManager->SetNumberOfThreads(G4Threading::G4GetNumberOfCores());
#else
    G4RunManager* runManager = new G4RunManager;
#endif

    // Geometry and Setup
    runManager->SetUserInitialization(new DetectorConstruction);

    // Physics List
    runManager->SetUserInitialization(new FTFP_BERT);

    // User action initialization: primary generation, results extraction
    runManager->SetUserInitialization(new ActionInitialization());

    // Get the pointer to the User Interface manager
    G4UImanager* UImanager = G4UImanager::GetUIpointer();

    if (argc>1) {
        // execute an argument macro file if exist
        G4String command = "/control/execute ";
        G4String fileName = argv[1];
        UImanager->ApplyCommand(command+fileName);
    }
    else {
       // start interactive session
       G4UIExecutive* ui = new G4UIExecutive(argc, argv);
       ui->SessionStart();
       delete ui;
   }

    // Job termination
   delete runManager;
    return 0;
}
```

Mandatory initialization #1: define experimental setup, fileds, sensitive detectors

Anatomy of an application (simplified)

```
int main(int argc,char** argv)
{
  // Construct the default run manager
  //
#ifdef G4MULTITHREADED
    G4MTRunManager* runManager = new G4MTRunManager;
    runManager->SetNumberOfThreads(G4Threading::G4GetNumberOfCores());
#else
    G4RunManager* runManager = new G4RunManager;
#endif

    // Geometry and Setup
    runManager->SetUserInitialization(new DetectorConstruction);

    // Physics List
    runManager->SetUserInitialization(new FTFP_BERT);

    // User action initialization: primary generation, results extraction
    runManager->SetUserInitialization(new ActionInitialization());

    // Get the pointer to the User Interface manager
    G4UImanager* UImanager = G4UImanager::GetUIpointer();

    if (argc>1) {
        // execute an argument macro file if exist
        G4String command = "/control/execute ";
        G4String fileName = argv[1];
        UImanager->ApplyCommand(command+fileName);
    }
    else {
       // start interactive session
       G4UIExecutive* ui = new G4UIExecutive(argc, argv);
       ui->SessionStart();
       delete ui;
   }

    // Job termination
   delete runManager;
    return 0;
}
```

Mandatory initialization #2: define physics list to be used

Anatomy of an application (simplified)

```
int main(int argc,char** argv)
{
  // Construct the default run manager
  //
#ifdef G4MULTITHREADED
    G4MTRunManager* runManager = new G4MTRunManager;
    runManager->SetNumberOfThreads(G4Threading::G4GetNumberOfCores());
#else
    G4RunManager* runManager = new G4RunManager;
#endif

    // Geometry and Setup
    runManager->SetUserInitialization(new DetectorConstruction);

    // Physics List
    runManager->SetUserInitialization(new FTFP_BERT);

    // User action initialization: primary generation, results extraction
    runManager->SetUserInitialization(new ActionInitialization());

    // Get the pointer to the User Interface manager
    G4UImanager* UImanager = G4UImanager::GetUIpointer();

    if (argc>1) {
        // execute an argument macro file if exist
        G4String command = "/control/execute ";
        G4String fileName = argv[1];
        UImanager->ApplyCommand(command+fileName);
    }
    else {
       // start interactive session
       G4UIExecutive* ui = new G4UIExecutive(argc, argv);
       ui->SessionStart();
       delete ui;
   }

    // Job termination
   delete runManager;
    return 0;
}
```

Mandatory initialization #3: User actions, including primary generation and quantities calculation (analysis)

Not discussed today!

Anatomy of an application (simplified)

```
int main(int argc,char** argv)
{
  // Construct the default run manager
  //
#ifdef G4MULTITHREADED
    G4MTRunManager* runManager = new G4MTRunManager;
    runManager->SetNumberOfThreads(G4Threading::G4GetNumberOfCores());
#else
    G4RunManager* runManager = new G4RunManager;
#endif

    // Geometry and Setup
    runManager->SetUserInitialization(new DetectorConstruction);

    // Physics List
    runManager->SetUserInitialization(new FTFP_BERT);

    // User action initialization: primary generation, results extraction
    runManager->SetUserInitialization(new ActionInitialization());

    // Get the pointer to the User Interface manager
    G4UImanager* UImanager = G4UImanager::GetUIpointer();

    if (argc>1) {
        // execute an argument macro file if exist
        G4String command = "/control/execute ";
        G4String fileName = argv[1];
        UImanager->ApplyCommand(command+fileName);
    }
    else {
       // start interactive session
       G4UIExecutive* ui = new G4UIExecutive(argc, argv);
       ui->SessionStart();
       delete ui;
   }

    // Job termination
   delete runManager;
    return 0;
}
```

Dealing with user-interface options:

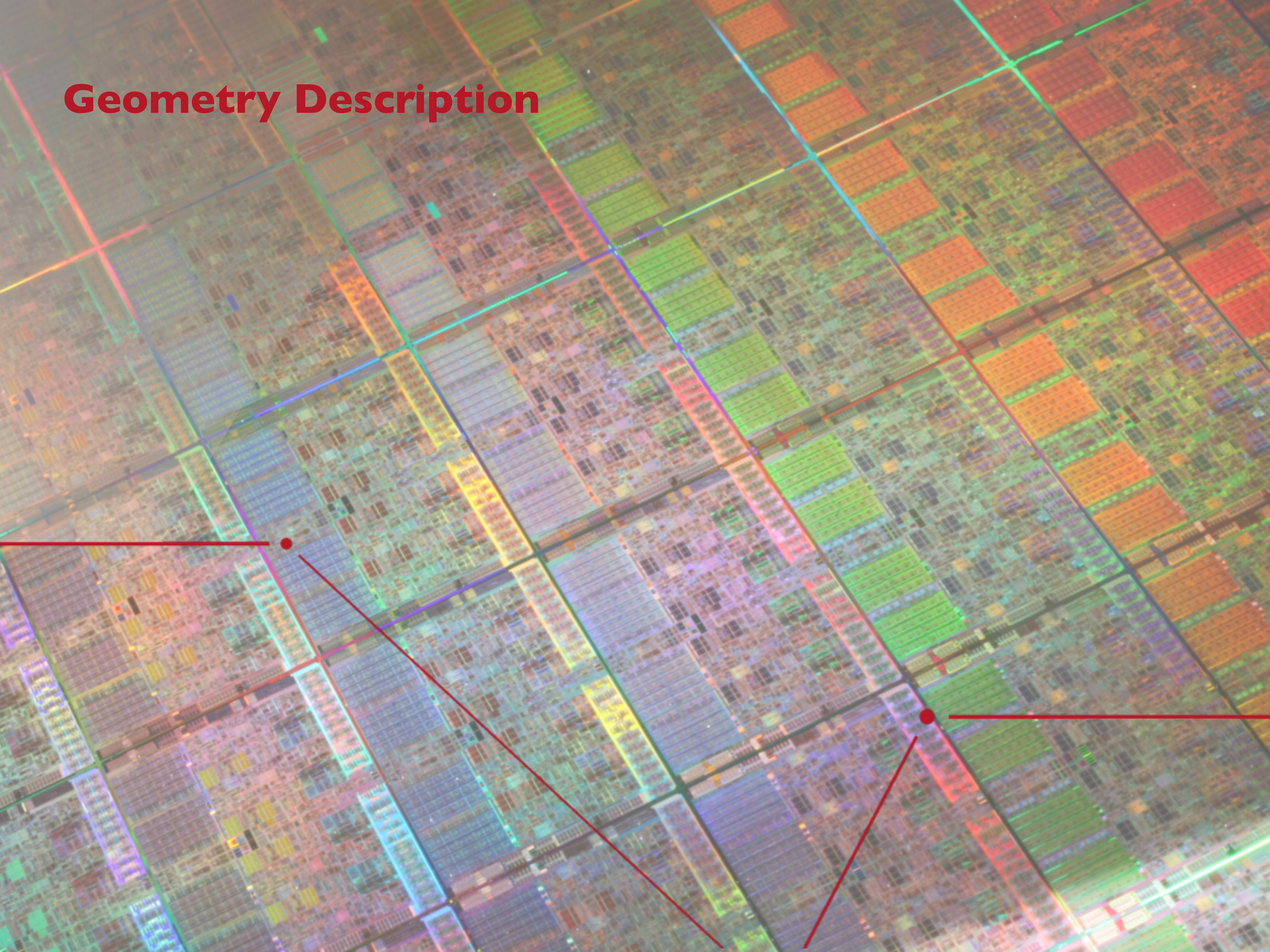
- batch mode (reading commands from a file, no GUI, exit at end)
- interactive mode (start UI session, possibly with GUI)

Anatomy of an application (simplified)

```
int main(int argc,char** argv)
{
  // Construct the default run manager
  //
#ifdef G4MULTITHREADED
    G4MTRunManager* runManager = new G4MTRunManager;
    runManager->SetNumberOfThreads(G4Threading::G4GetNumberOfCores());
#else
    G4RunManager* runManager = new G4RunManager;
#endif

    // Geometry and Setup
    runManager->SetUserInitialization(new DetectorConstruction);

    // Physics List
    runManager->SetUserInitialization(new FTFP_BERT);

    // User action initialization: primary generation, results extraction
    runManager->SetUserInitialization(new ActionInitialization());

    // Get the pointer to the User Interface manager
    G4UImanager* UImanager = G4UImanager::GetUIpointer();

    if (argc>1) {
        // execute an argument macro file if exist
        G4String command = "/control/execute ";
        G4String fileName = argv[1];
        UImanager->ApplyCommand(command+fileName);
    }
    else {
       // start interactive session
       G4UIExecutive* ui = new G4UIExecutive(argc, argv);
       ui->SessionStart();
       delete ui;
   }

    // Job termination
   delete runManager;
    return 0;
}
```

Exit: clean-up everything

Geometry Description

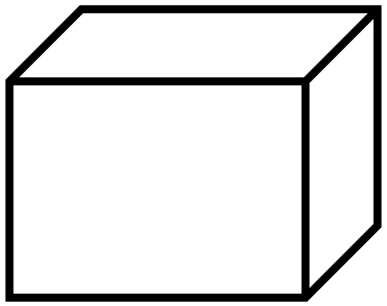


Define detector geometry

Basic strategy

```
G4VSolid* pBoxSolid =
  new G4Box("aBoxSolid",
      1.*m, 2.*m, 3.*m);
```

Solid: shape and size

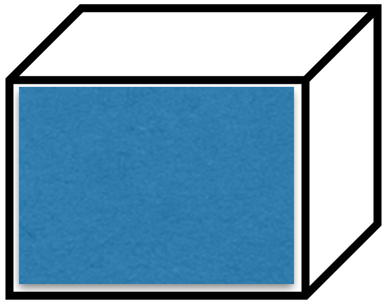


Define detector geometry

Basic strategy

```
G4VSolid* pBoxSolid =
  new G4Box("aBoxSolid",
        1.*m, 2.*m, 3.*m);
G4LogicalVolume* pBoxLog =
  new G4LogicalVolume( pBoxSolid,
    pBoxMaterial, "aBoxLog", 0, 0, 0);
```

Logical: solid + material

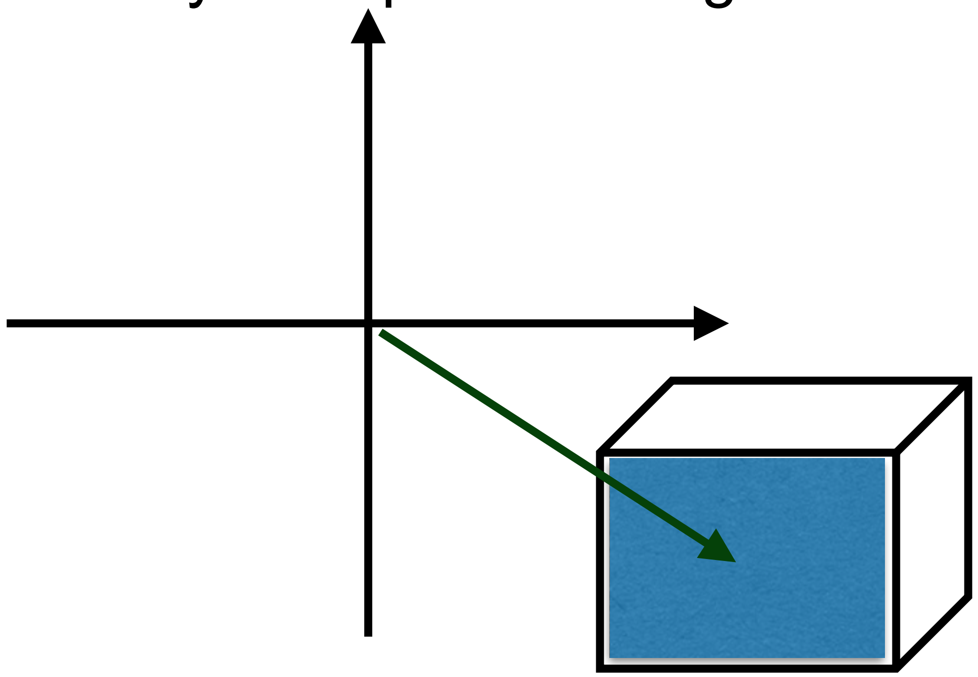


Define detector geometry

Basic strategy

```
G4VSolid* pBoxSolid =
  new G4Box("aBoxSolid",
        1.*m, 2.*m, 3.*m);
G4LogicalVolume* pBoxLog =
  new G4LogicalVolume( pBoxSolid,
   pBoxMaterial, "aBoxLog", 0, 0, 0);
G4VPhysicalVolume* aBoxPhys =
  new G4PVPlacement( pRotation,
   G4ThreeVector(posX, posY, posZ),
   pBoxLog, "aBoxPhys", pMotherLog,
   0, copyNo);
```

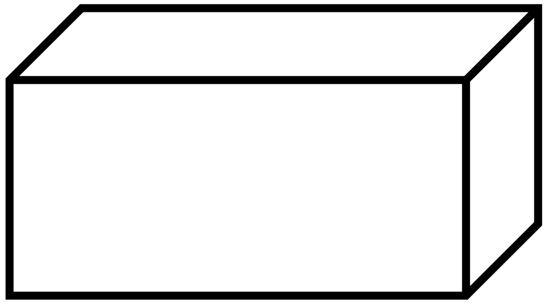
Physical: positionining



A volume is placed in its mother volume. Position and rotation of the daughter volume is described with respect to the local coordinate system of the mother volume. The origin of mother volume's local coordinate system is at the center of the mother volume.

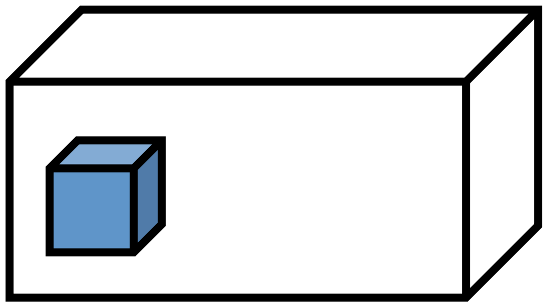
Geometry Hierarchy

- One logical volume can be placed more than once. One or more volumes can be placed to a mother volume.
- If the mother volume is placed more than once, all daughters are by definition appear in all of mother physical volumes.
- The world volume must be a unique physical volume which fully contains all the other volumes.
  - The world volume defines the global coordinate system. The origin of the global coordinate system is at the center of the world volume.
  - Position of a track is given with respect to the global coordinate system.



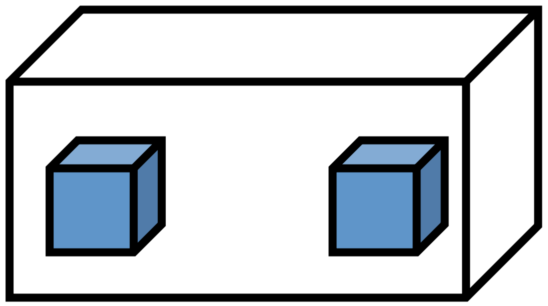
Geometry Hierarchy

- One logical volume can be placed more than once. One or more volumes can be placed to a mother volume.
- If the mother volume is placed more than once, all daughters are by definition appear in all of mother physical volumes.
- The world volume must be a unique physical volume which fully contains all the other volumes.
  - The world volume defines the global coordinate system. The origin of the global coordinate system is at the center of the world volume.
  - Position of a track is given with respect to the global coordinate system.



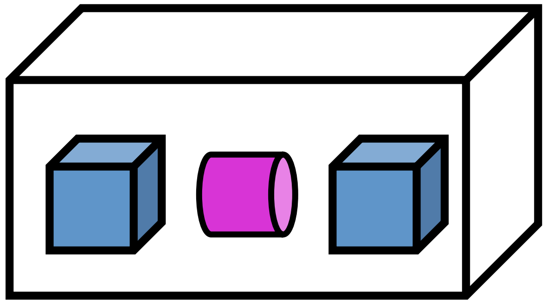
Geometry Hierarchy

- One logical volume can be placed more than once. One or more volumes can be placed to a mother volume.
- If the mother volume is placed more than once, all daughters are by definition appear in all of mother physical volumes.
- The world volume must be a unique physical volume which fully contains all the other volumes.
  - The world volume defines the global coordinate system. The origin of the global coordinate system is at the center of the world volume.
  - Position of a track is given with respect to the global coordinate system.



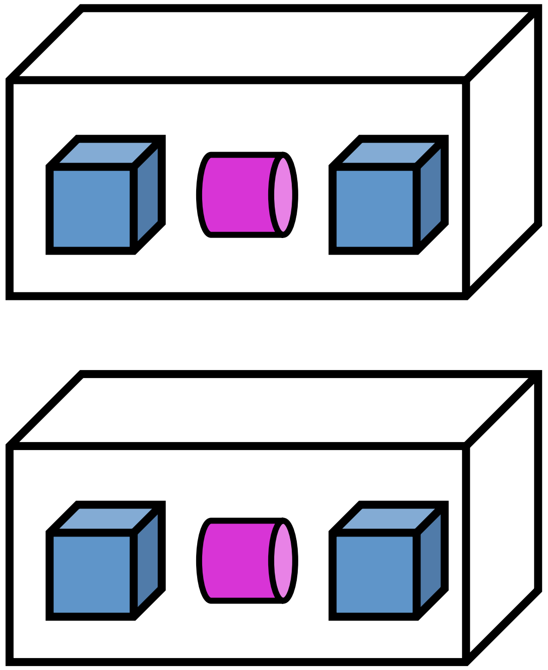
Geometry Hierarchy

- One logical volume can be placed more than once. One or more volumes can be placed to a mother volume.
- If the mother volume is placed more than once, all daughters are by definition appear in all of mother physical volumes.
- The world volume must be a unique physical volume which fully contains all the other volumes.
  - The world volume defines the global coordinate system. The origin of the global coordinate system is at the center of the world volume.
  - Position of a track is given with respect to the global coordinate system.



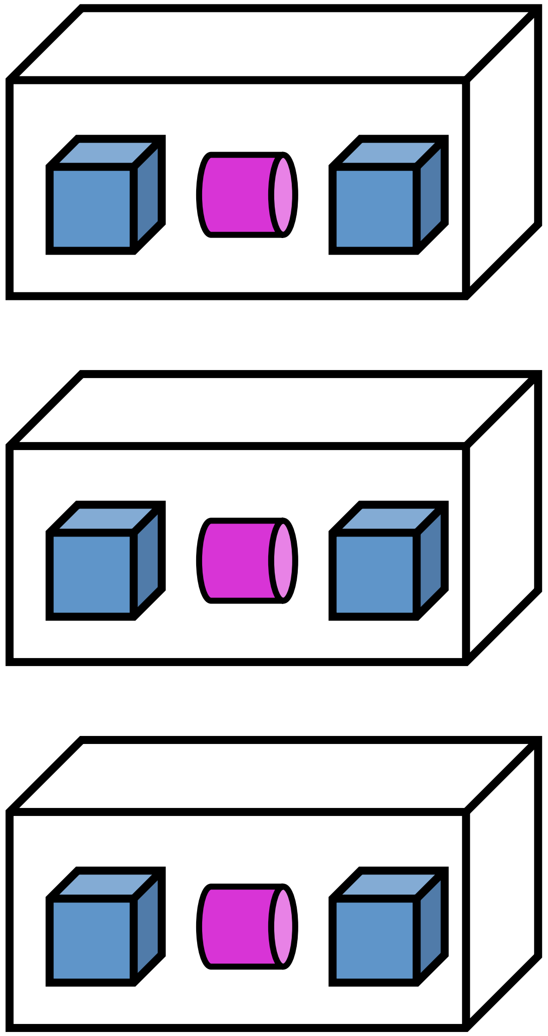
Geometry Hierarchy

- One logical volume can be placed more than once. One or more volumes can be placed to a mother volume.
- If the mother volume is placed more than once, all daughters are by definition appear in all of mother physical volumes.
- The world volume must be a unique physical volume which fully contains all the other volumes.
  - The world volume defines the global coordinate system. The origin of the global coordinate system is at the center of the world volume.
  - Position of a track is given with respect to the global coordinate system.



Geometry Hierarchy

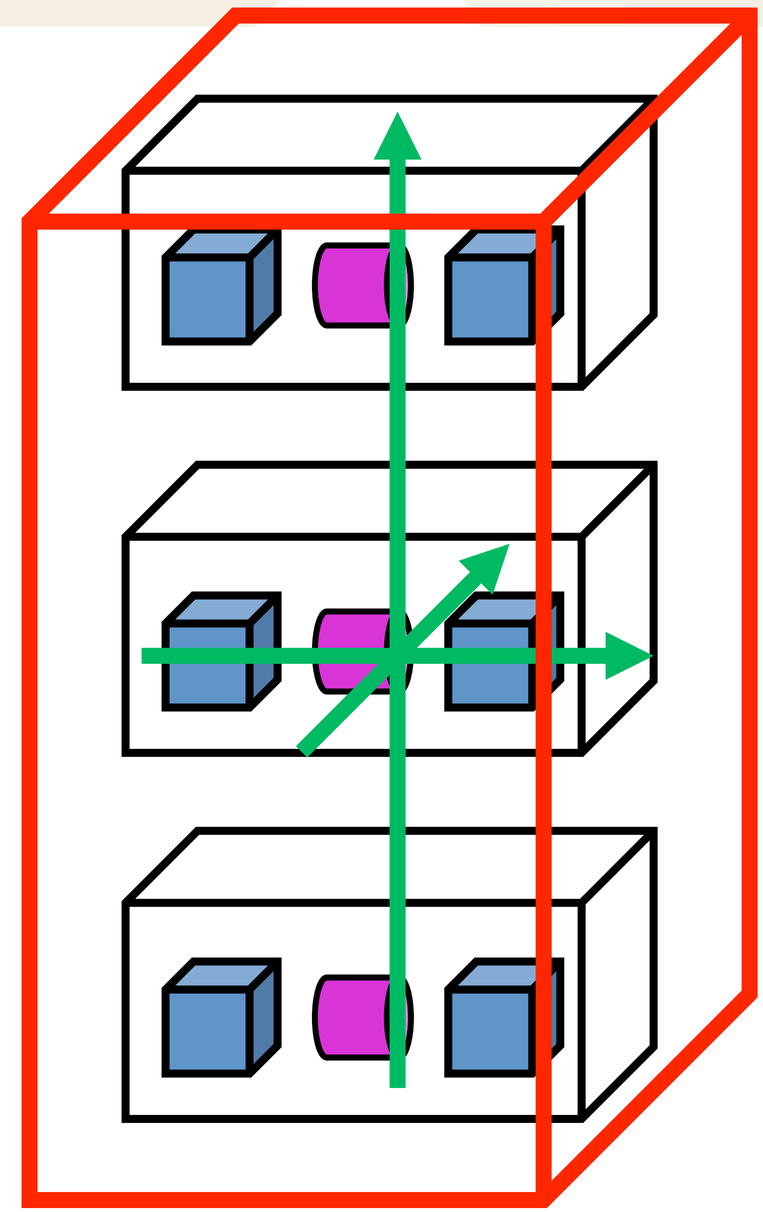
- One logical volume can be placed more than once. One or more volumes can be placed to a mother volume.
- If the mother volume is placed more than once, all daughters are by definition appear in all of mother physical volumes.
- The world volume must be a unique physical volume which fully contains all the other volumes.
  - The world volume defines the global coordinate system. The origin of the global coordinate system is at the center of the world volume.
  - Position of a track is given with respect to the global coordinate system.



Geometry Hierarchy

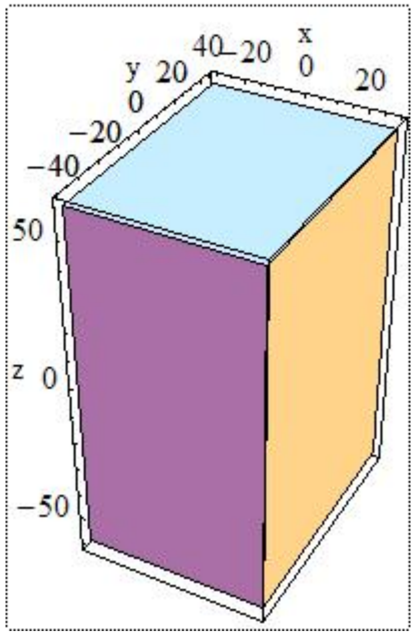
SLAC

- One logical volume can be placed more than once. One or more volumes can be placed to a mother volume.
- If the mother volume is placed more than once, all daughters are by definition appear in all of mother physical volumes.
- The world volume must be a unique physical volume which fully contains all the other volumes.
  - The world volume defines the global coordinate system. The origin of the global coordinate system is at the center of the world volume.
  - Position of a track is given with respect to the global coordinate system.

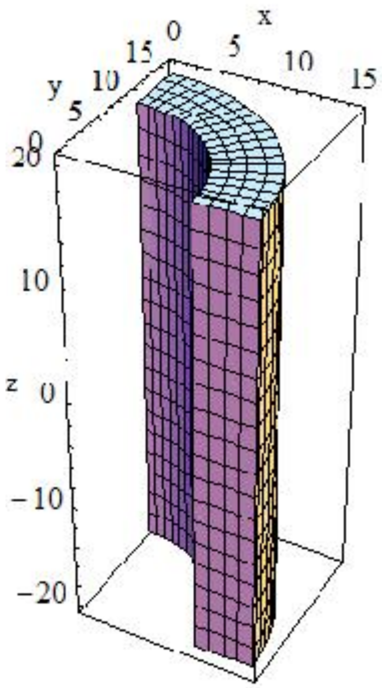


CSG: Box, Tubs

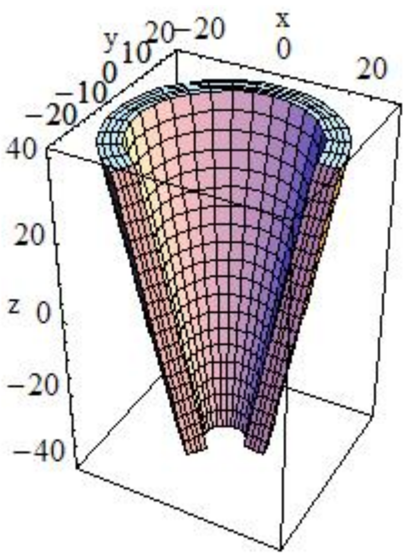
```
G4Box(const G4String &pname,   // name
            G4double half_x,   // X half size
            G4double half_y,   // Y half size
            G4double half_z);  // Z half size
```



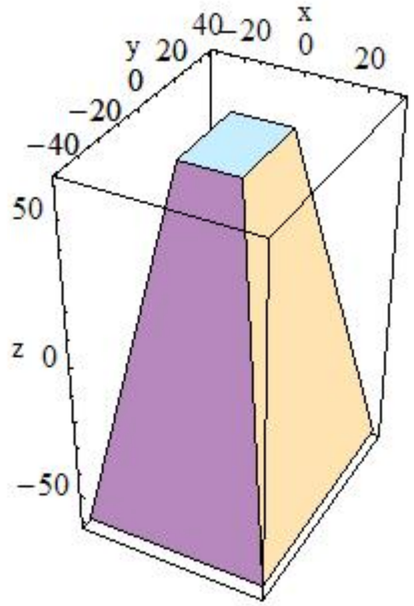

```
G4Tubs(const G4String &pname,  // name
             G4double  pRmin,  // inner radius
             G4double  pRmax,  // outer radius
             G4double  pDz,    // Z half length
             G4double  pSphi,  // starting Phi
             G4double  pDphi); // segment angle
```



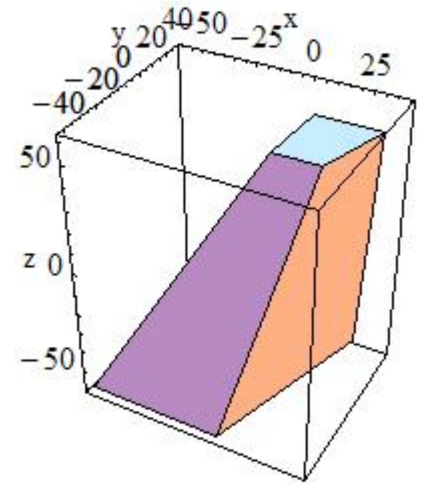
Other CSG Solids



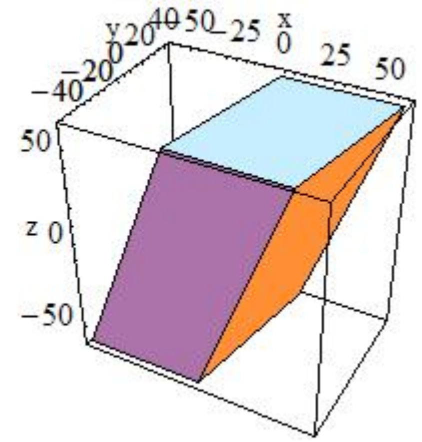
G4Cons



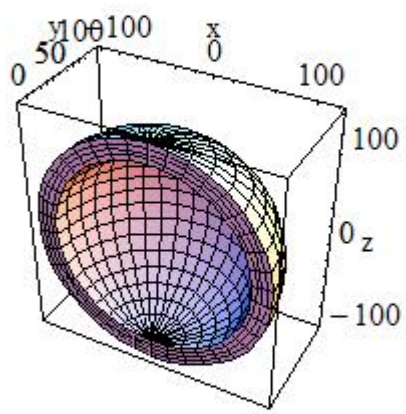
G4Trd



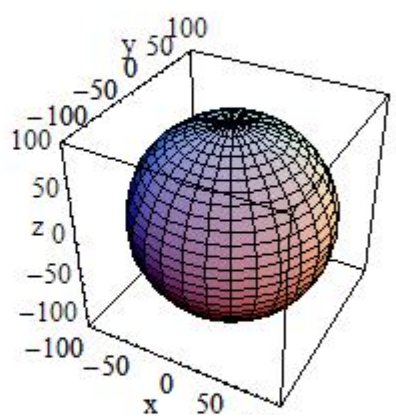
G4Trap



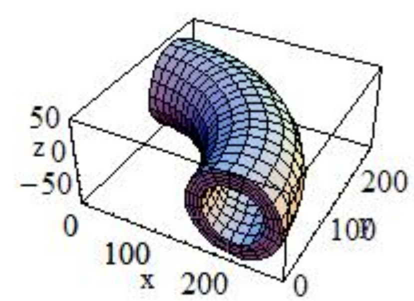
G4Para
(parallelepiped)



G4Sphere



G4Orb
(full solid sphere)

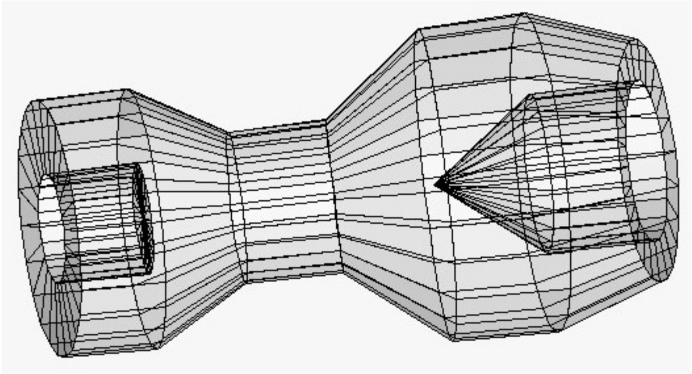


G4Torus

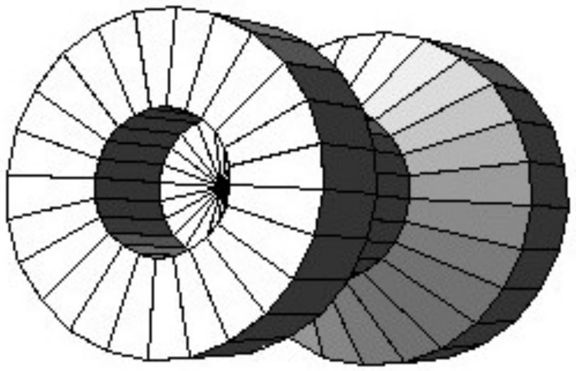
Consult to
[Section 4.1.2 of Geant4 Application Developers Guide]for all available shapes.

Specific CSG: Polycones

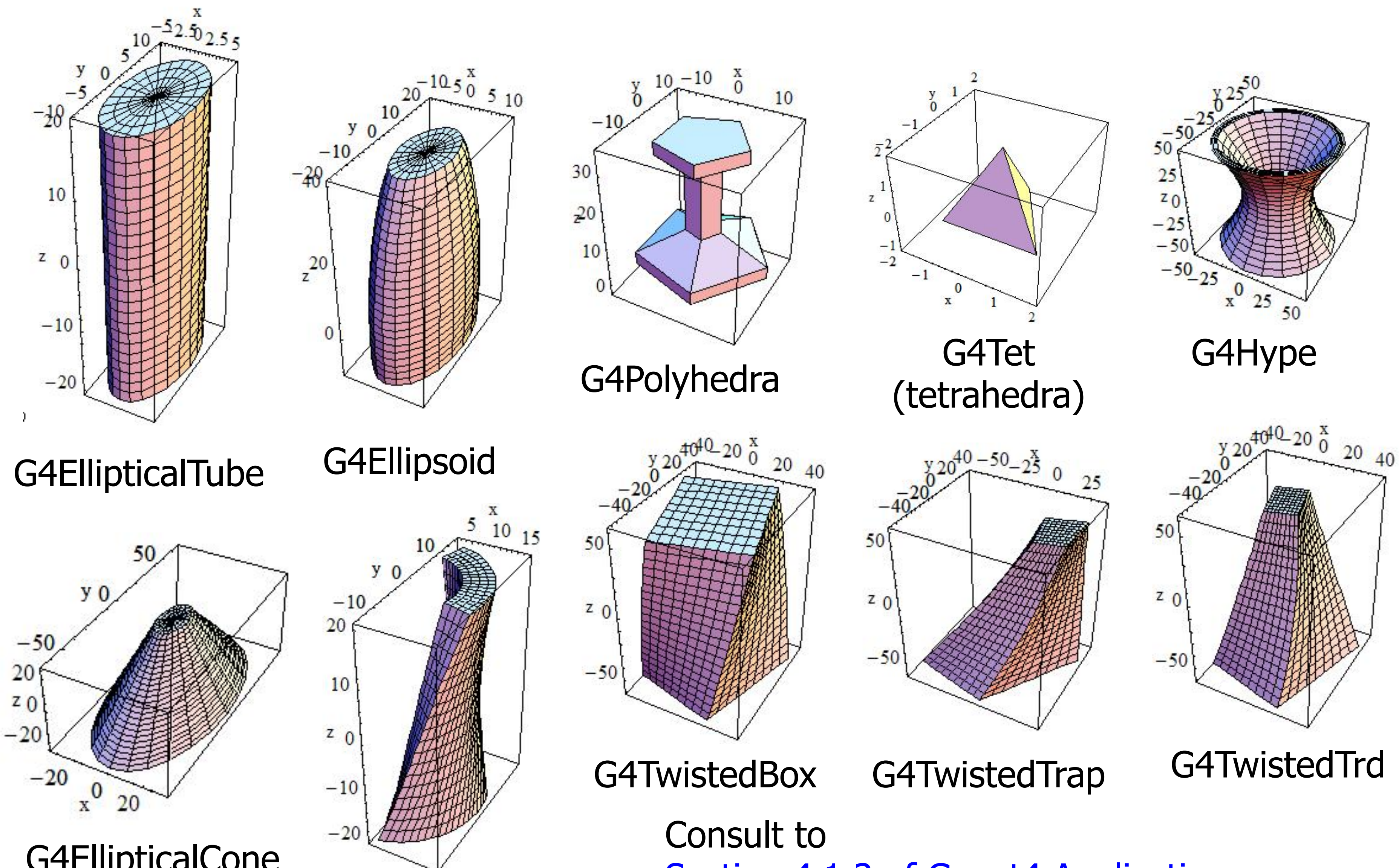
```
G4Polycone(const G4String& pName,
           G4double phiStart,
           G4double phiTotal,
           G4int numRZ,
           const G4double r[],
           const G4double z[]);
```



- **numRZ** - numbers of corners in the **r,z** space
- **r, z** - coordinates of corners



Even more CSGs



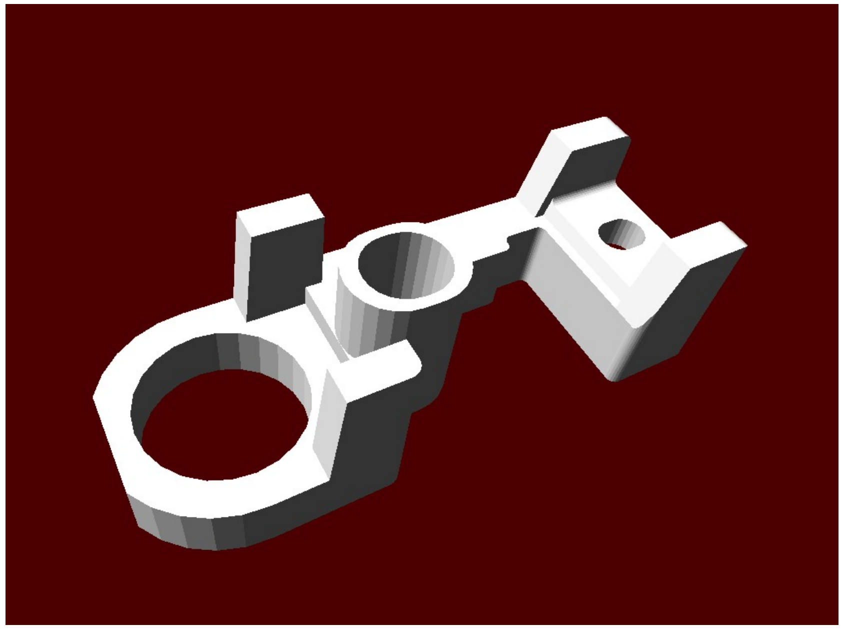
G4EllipticalCone

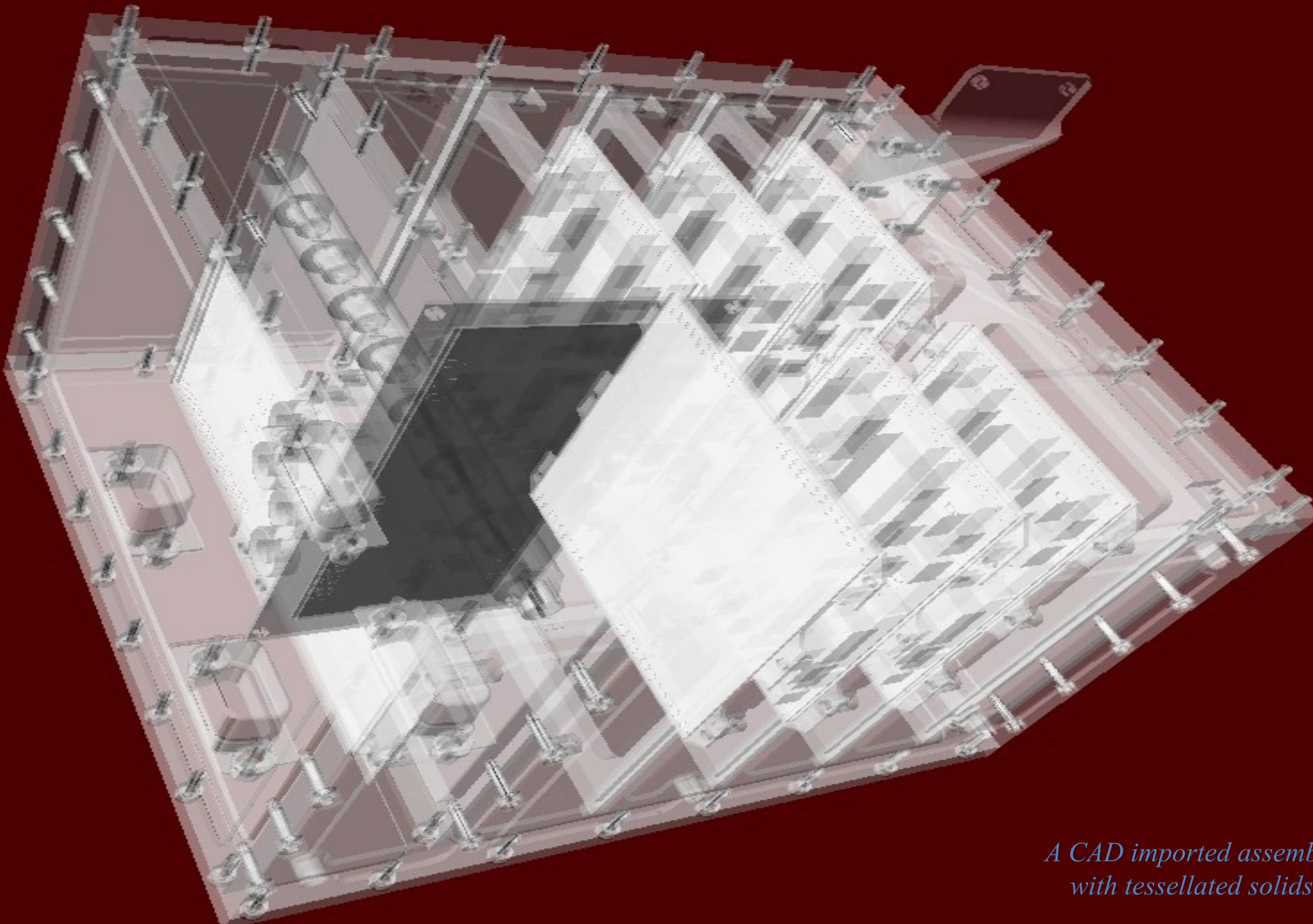
G4TwistedTubs

Consult to
[Section 4.1.2 of Geant4 Application Developers Guide](#) for all available shapes.

Tessellated Solids

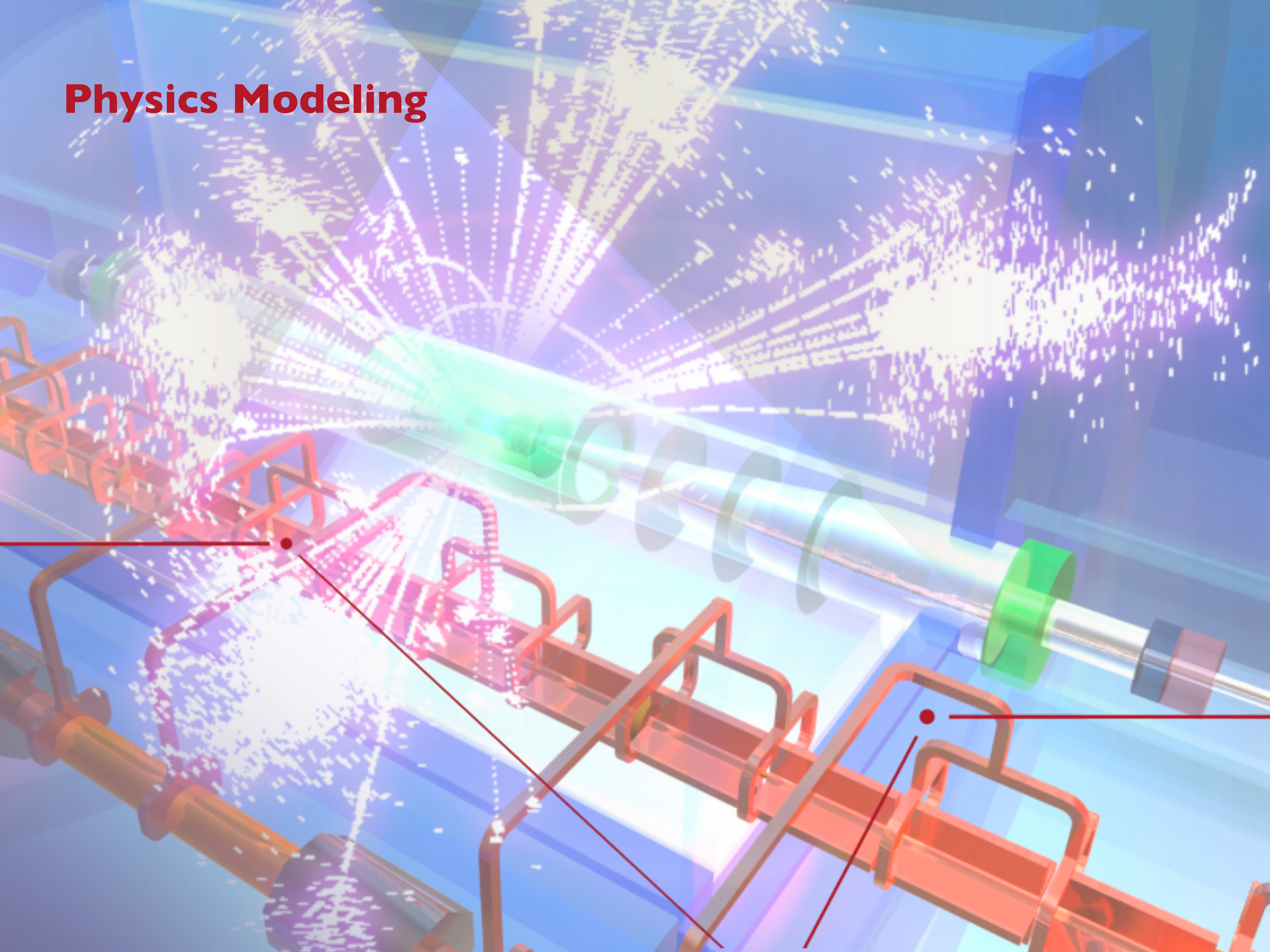
- **`G4TessellatedSolid`** (since 8.1)
  - Generic solid defined by a number of facets (**`G4VFacet`**)
    - Facets can be triangular (**`G4TriangularFacet`**) or quadrangular (**`G4QuadrangularFacet`**)
  - Constructs especially important for conversion of complex geometrical shapes imported from CAD systems
  - But can also be explicitly defined:
    - By providing the vertices of the facets in *anti-clock wise* order, in *absolute* or *relative* reference frame
  - GDML binding





A CAD imported assembly with tessellated solids

Physics Modeling



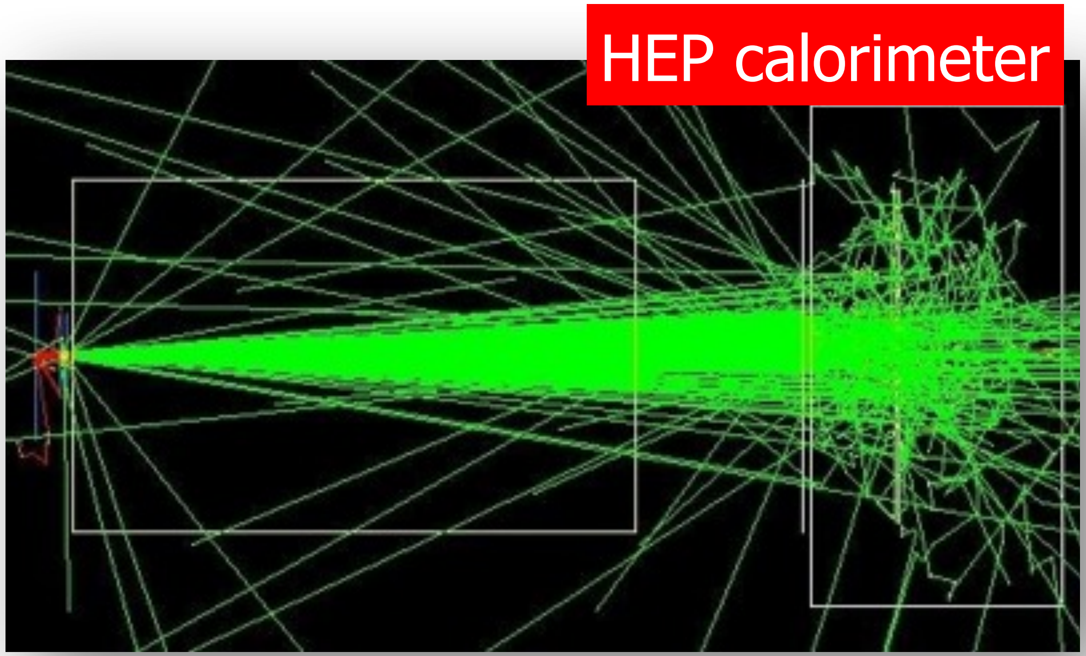
Located in sources/processes/electromagnetic

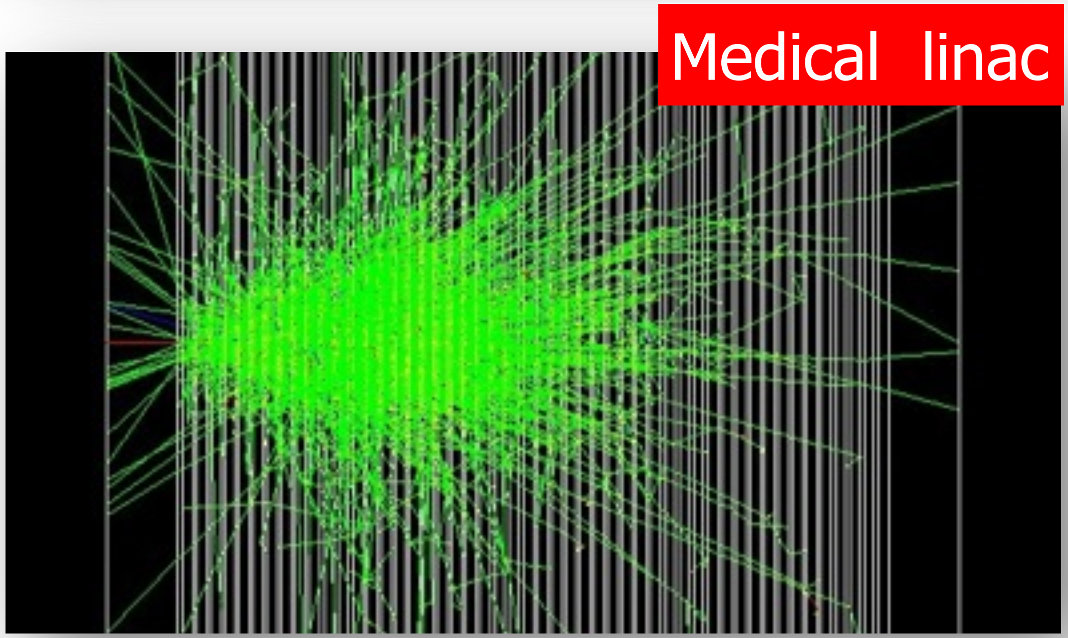
Geant4 EM packages

- Standard
  - γ, e± up to 100 TeV
  - hadrons up to 100 TeV
  - ions up to 100 TeV
- Muons
  - up to 1 PeV
  - energy loss propagator
- X-rays
  - X-ray and optical photon production proc.
- High-energy
  - processes at high energy (E>10GeV)
  - physics for exotic particles
- Polarisation
  - simulation of polarized beams
- Optical
  - optical photon interactions
- Low-energy
  - Livermore library γ, e- from 10 eV up to 1 GeV
  - Livermore library based polarized processes
  - PENELOPE code rewrite , γ, e- , e+ from 100 eV up to 1 GeV (2008 version)
  - hadrons and ions up to 1 GeV
  - atomic de-excitation (fluorescence + Auger)
- Geant4-DNA
  - microdosimetry models for radiobiology (Geant4-DNA project) from 0.025 eV to 10 MeV
- Adjoint
  - New sub-library for reverse Monte Carlo simulation from the detector of interest back to source of radiation
- Utils
  - general EM interfaces

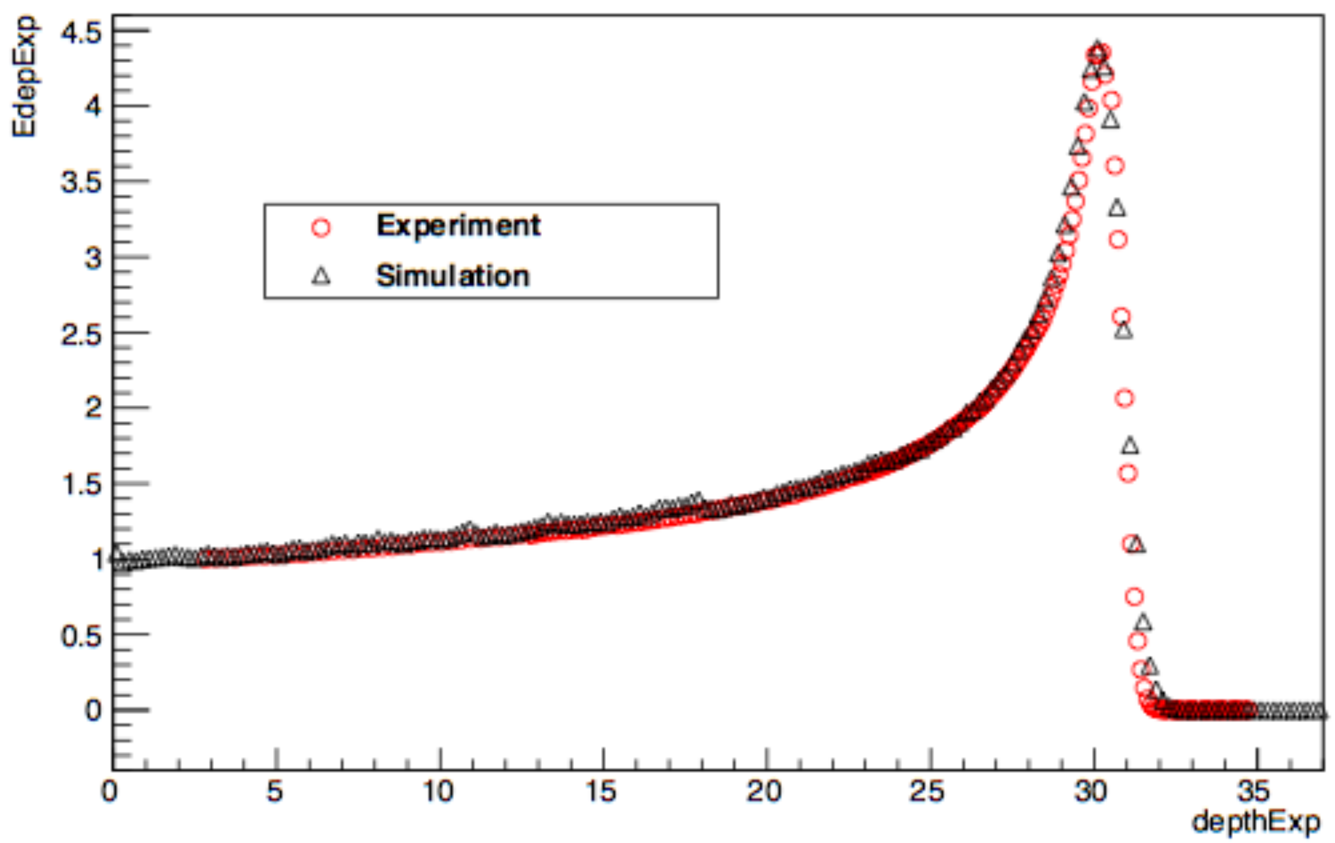
Gamma and electron transport

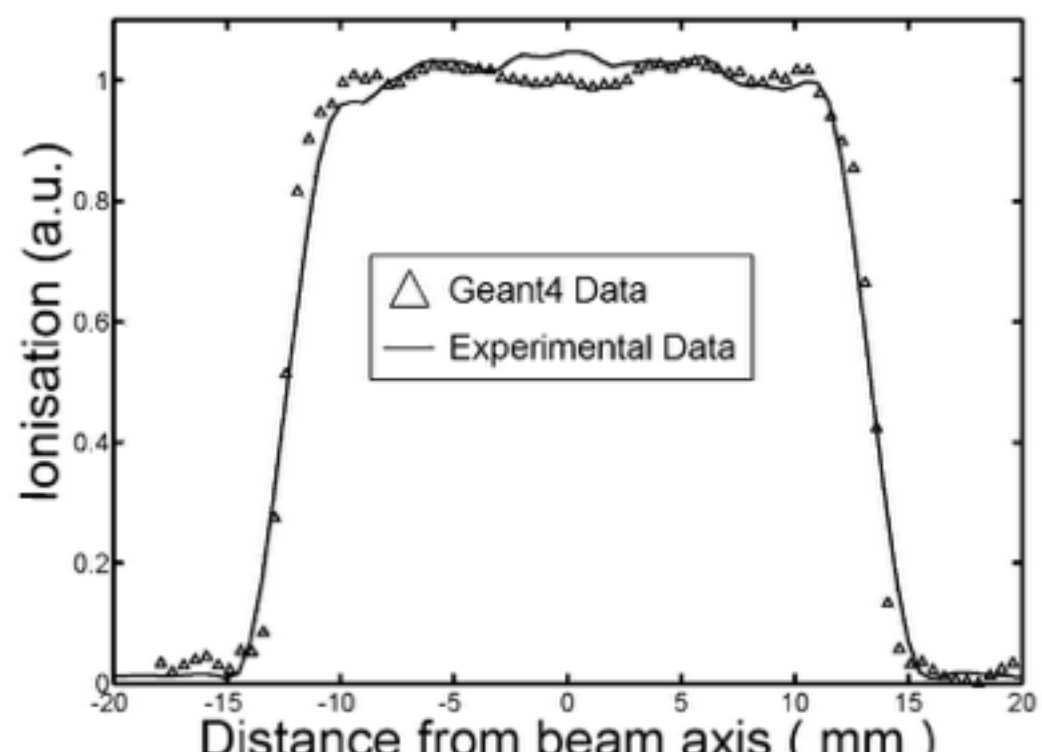
- Photon processes
  - γ conversion into e+e- pair
  - Compton scattering
  - Photoelectric effect
  - Rayleigh scattering
  - *Gamma-nuclear interaction in hadronic sub-package*
- Electron and positron processes
  - Ionisation
  - Coulomb scattering
  - Bremsstrahlung
  - Positron annihilation
  - *Nuclear interaction in hadronic sub-package*
- Suitable for HEP & many other Geant4 applications with electron and gamma beams





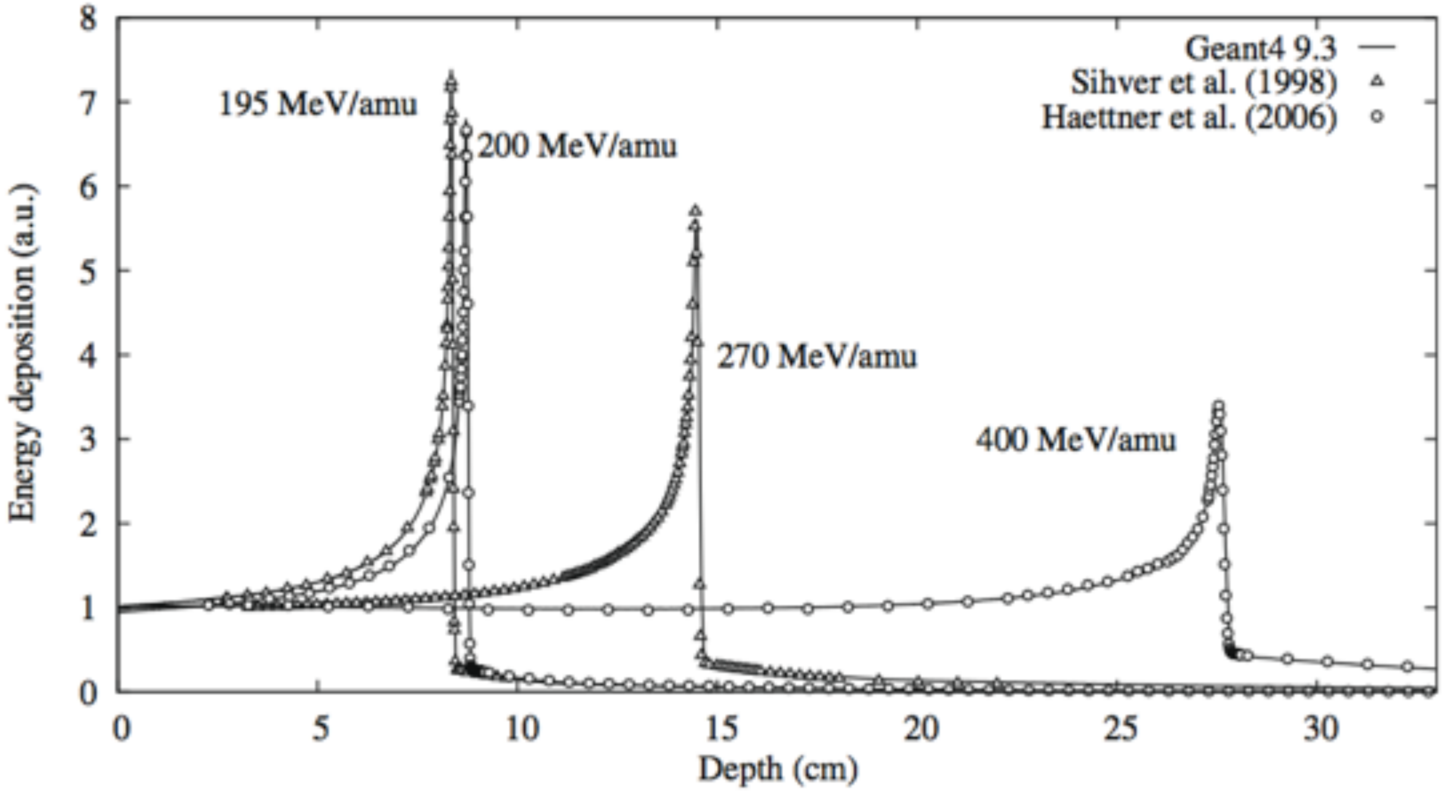
- CATANA (Italy) **proton therapy line** for the treatment of ocular melanoma
- **Good agreement with data O(%)**





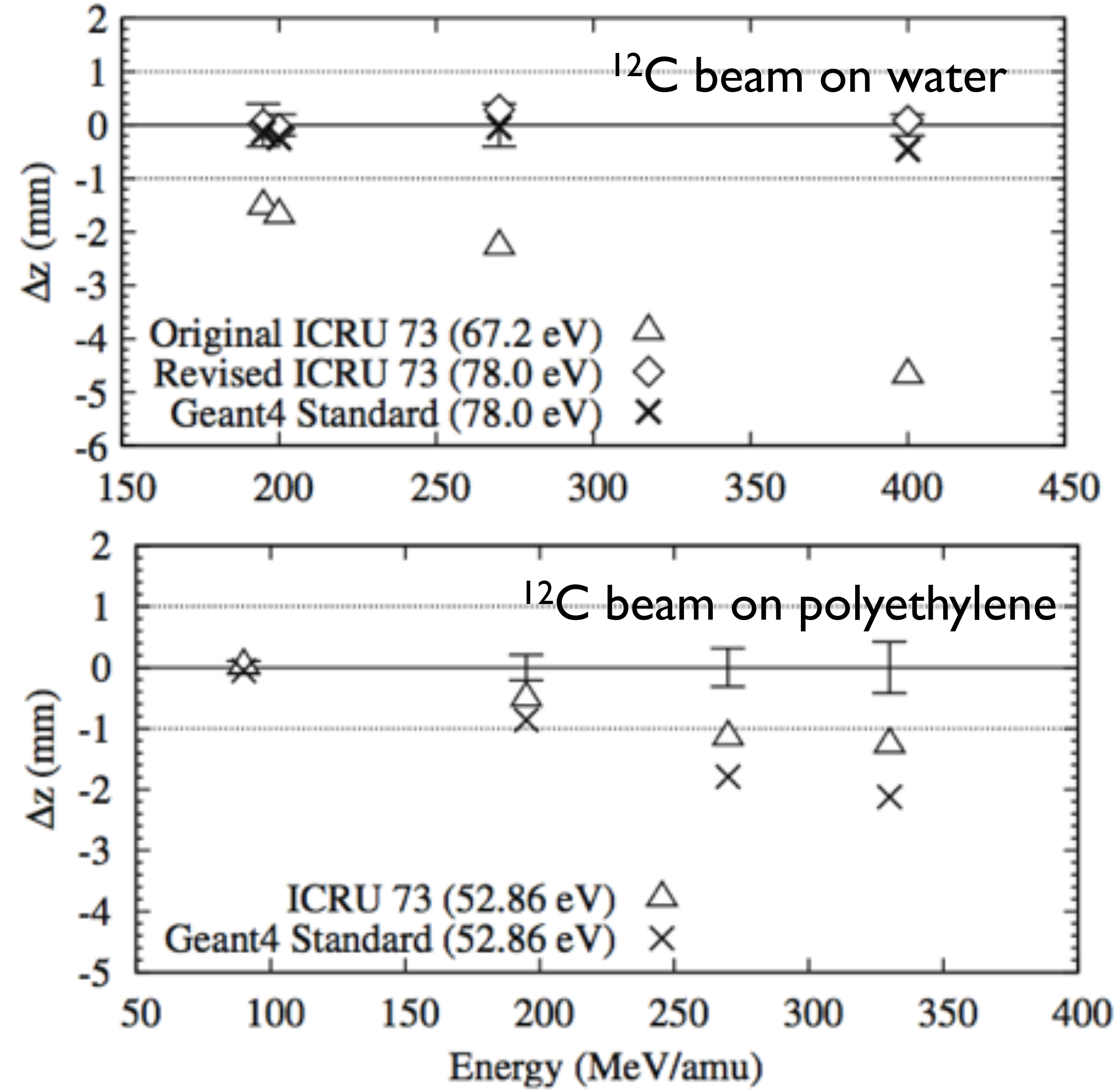
IEEE TNS Vol. 52 No. 1 (2005) 262-265

^{12}C beam on Water



Nuclear Instruments and Methods in Physics Research B 268 (2010) 2343–2354

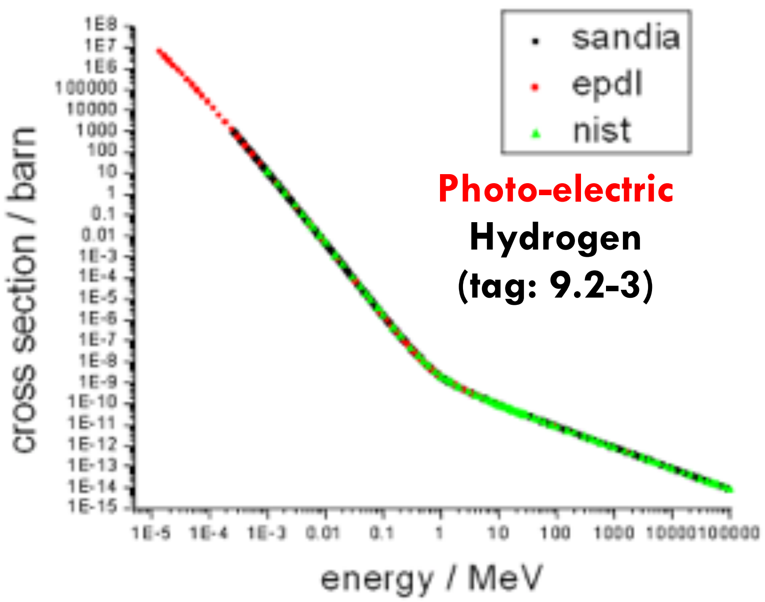
Bragg Peak position: MC Vs data

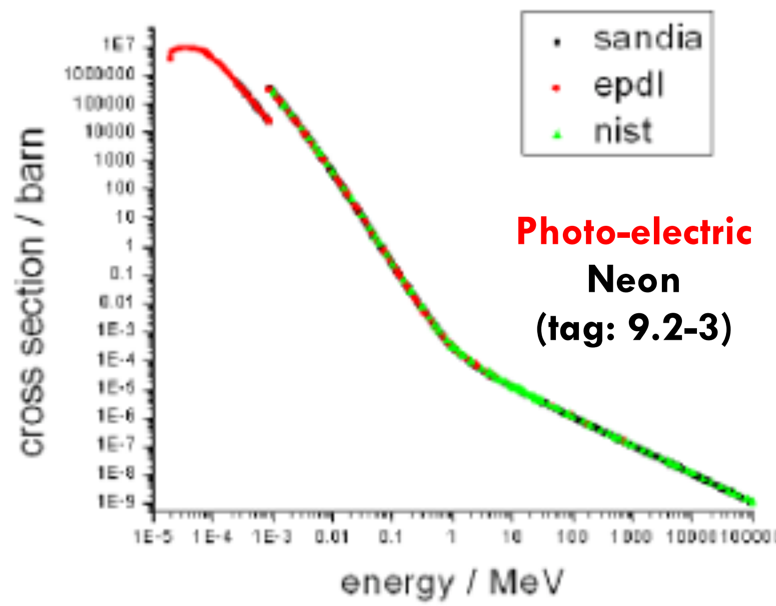


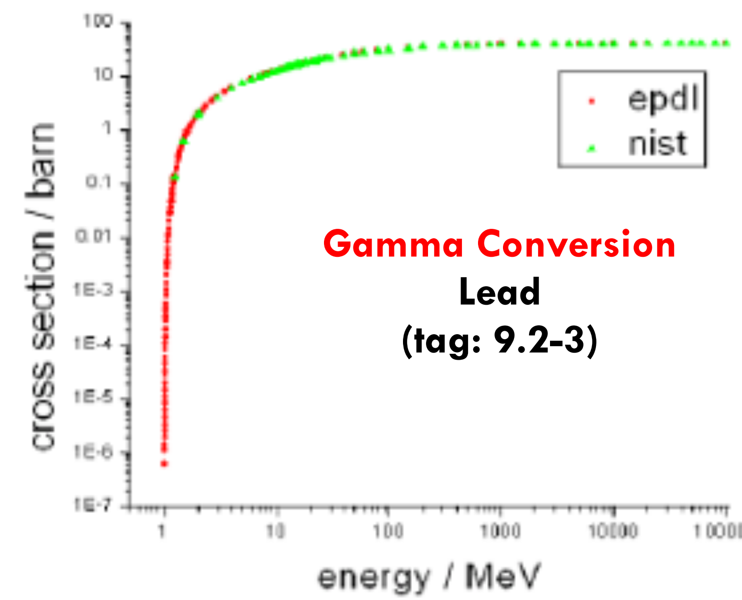
- Absolute measurement of Bragg Peak position
- **Good agreement** with data
  - Alternative ionization potentials for water: better description of data
  - Example of Geant4 extendibility (thanks to OO architecture)

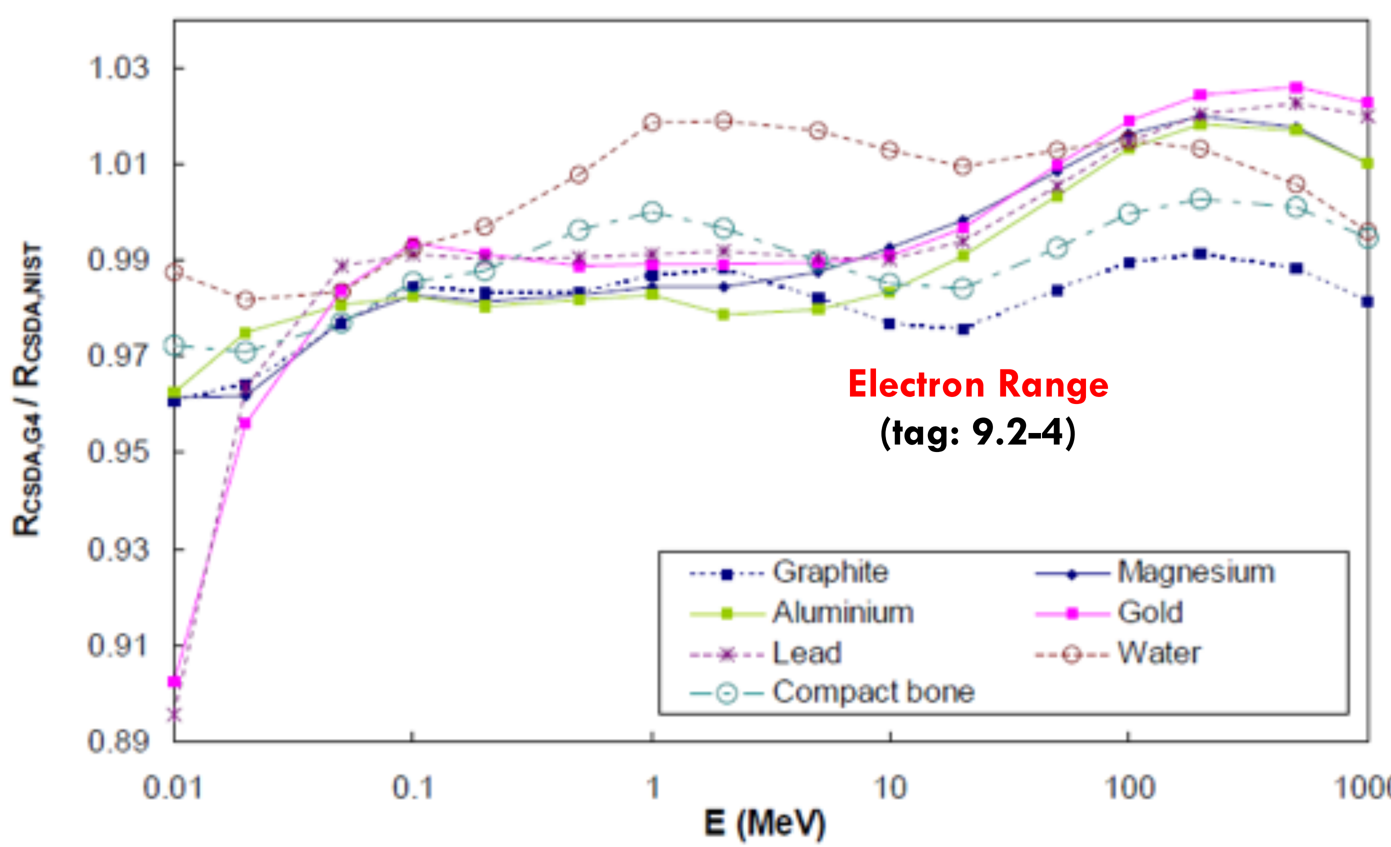
Eg. of verification of Livermore models

See Nucl. Instrum. and Meth. A 618 (2010) 315-322



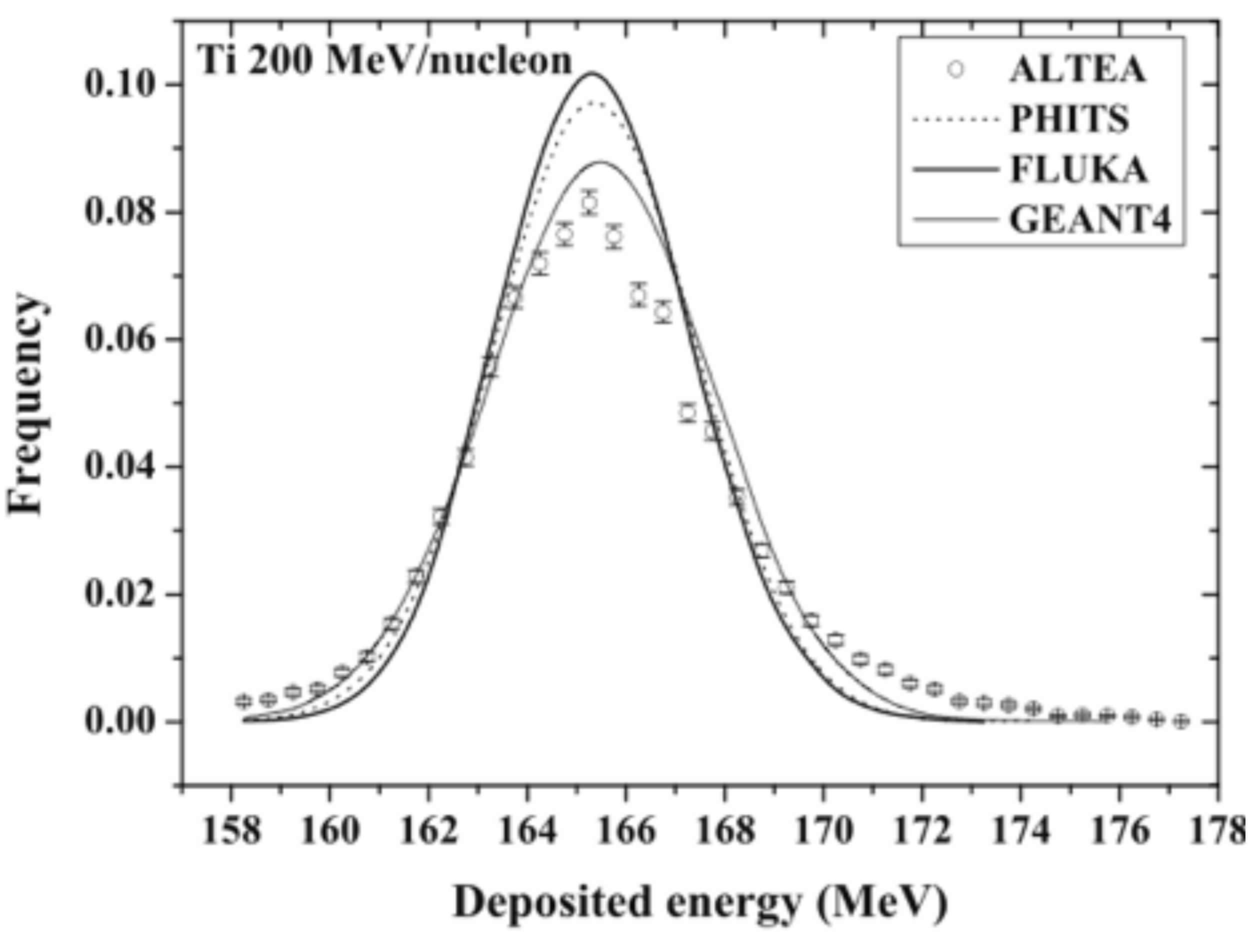


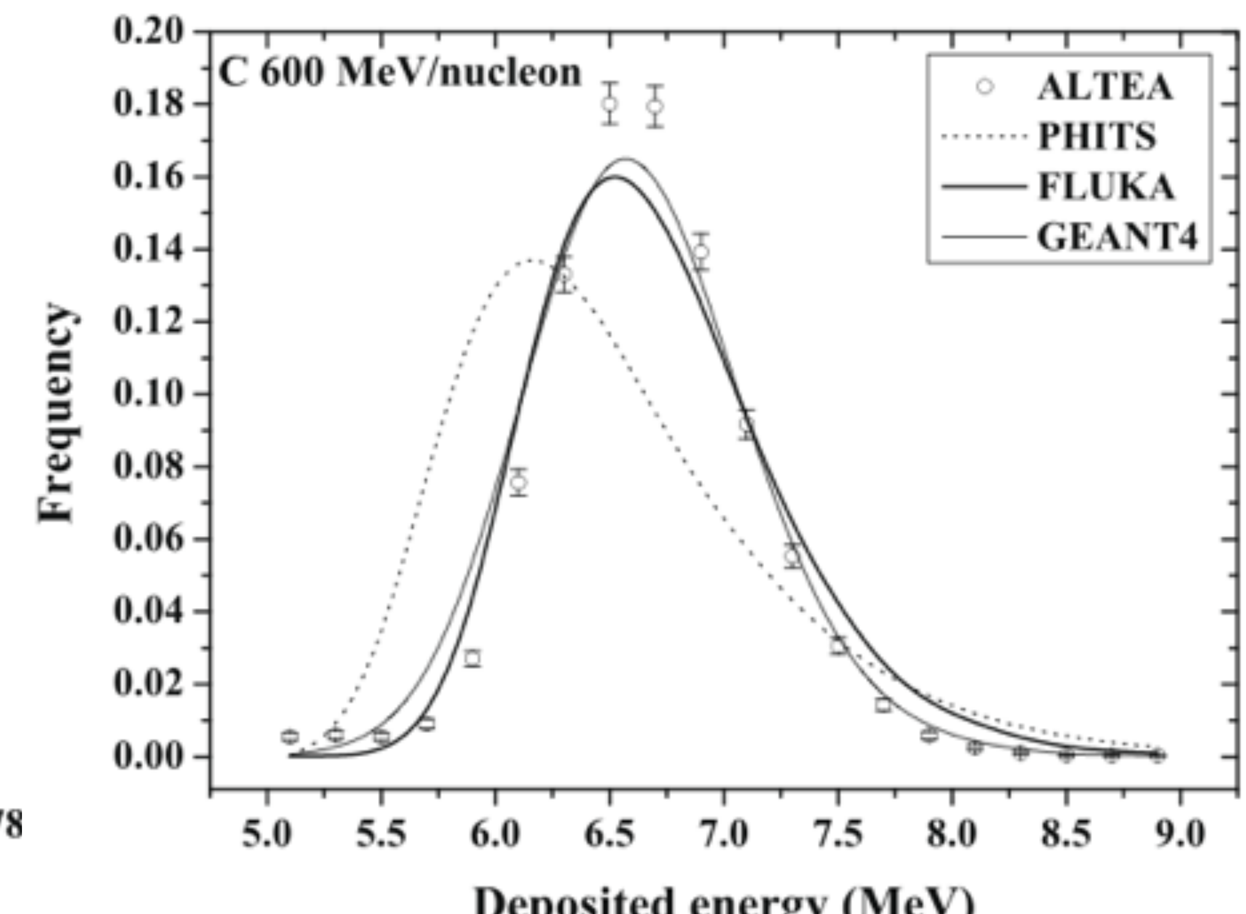




- ALTEA: Si detector on ISS
  - Goal: **measure ion fluxes and direction in spacecrafts**
- Comparison of signal with simulation codes: ions different E
- **Good level of agreement with data**

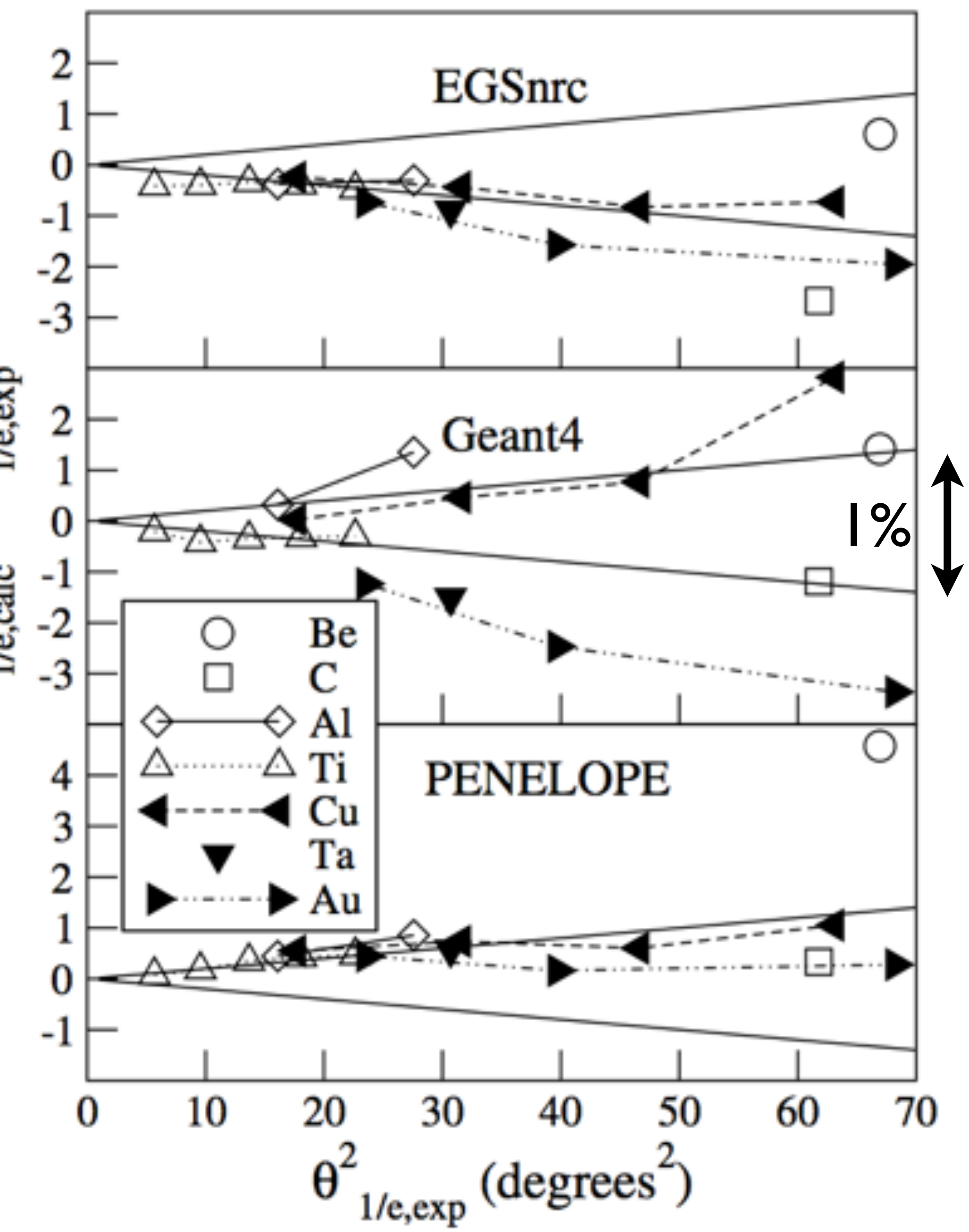
NIMB267(2009)3549-3557





Performance summary

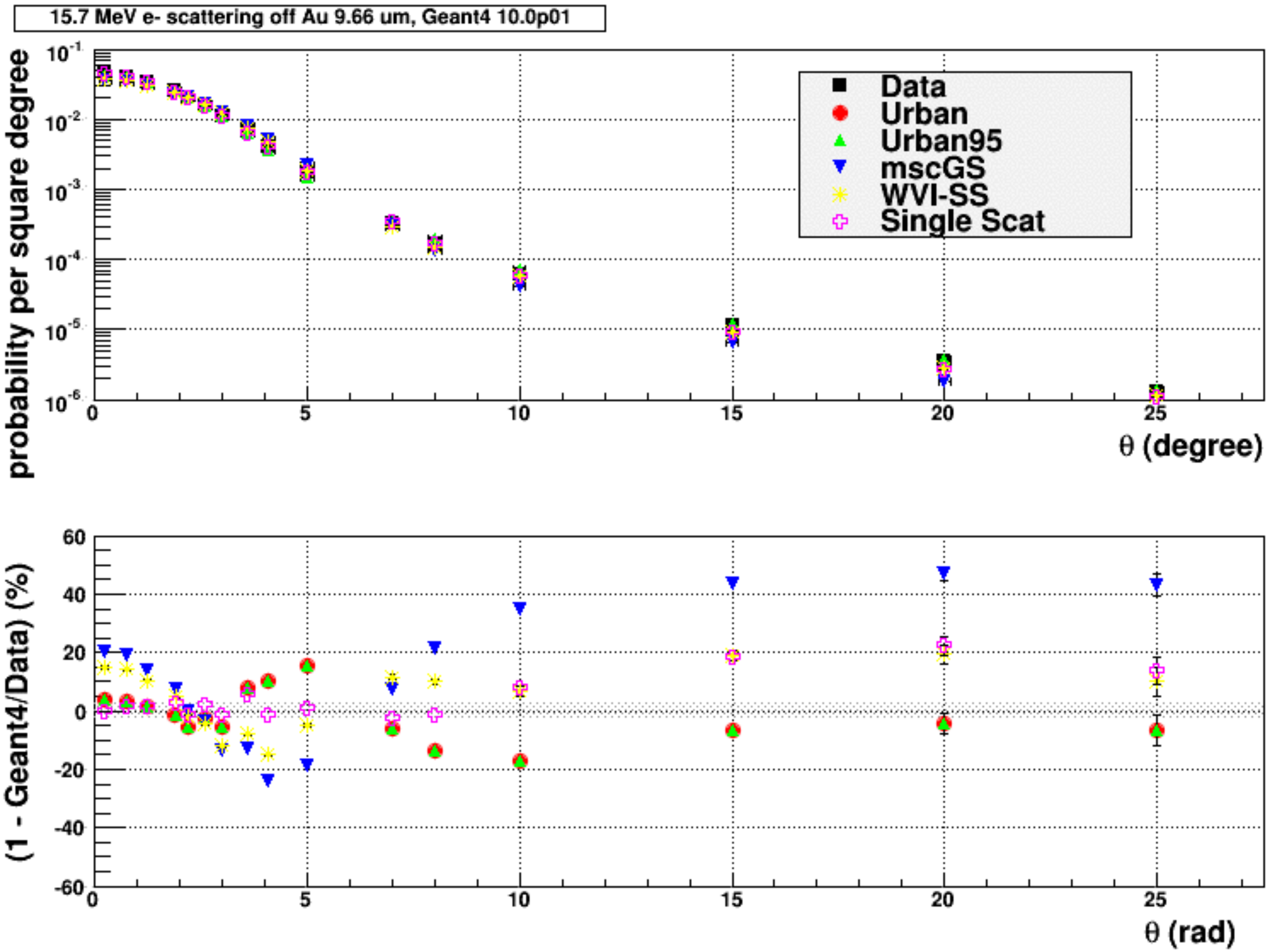
13 MeV e- beam on thin foil



Phys. Med. Biol. **54** 6151

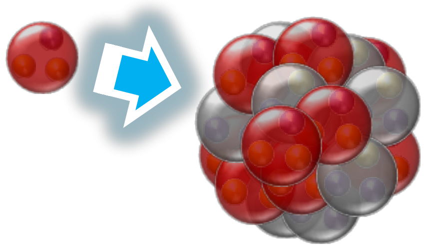
- **Typical precision ~1%**
  - Atomic and medium effects bring larger uncertainties
- Condensed description of **multiple scattering** introduces further approximations
  - MSC models are one of the main development areas

Multiple Scattering: Hanson data



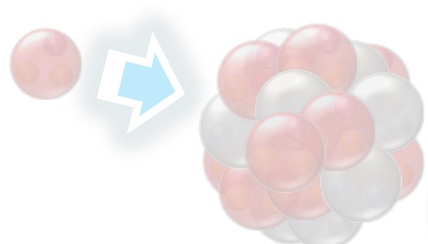
Hanson et al., 'Measurement of multiple scattering of 15.7 MeV electrons,' Phys. Rev. 84 (1951) 634-637

<u>One</u> hadronic collision = sequence of many hadronic interaction<u>s</u>

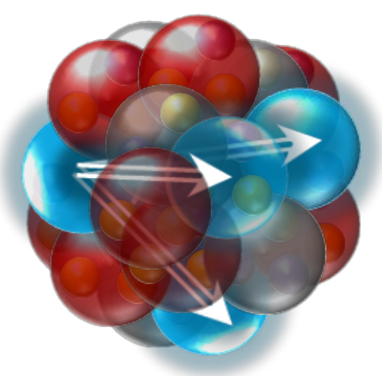


Projectile hitting
a target nucleus

One hadronic collision = sequence of many hadronic interactions

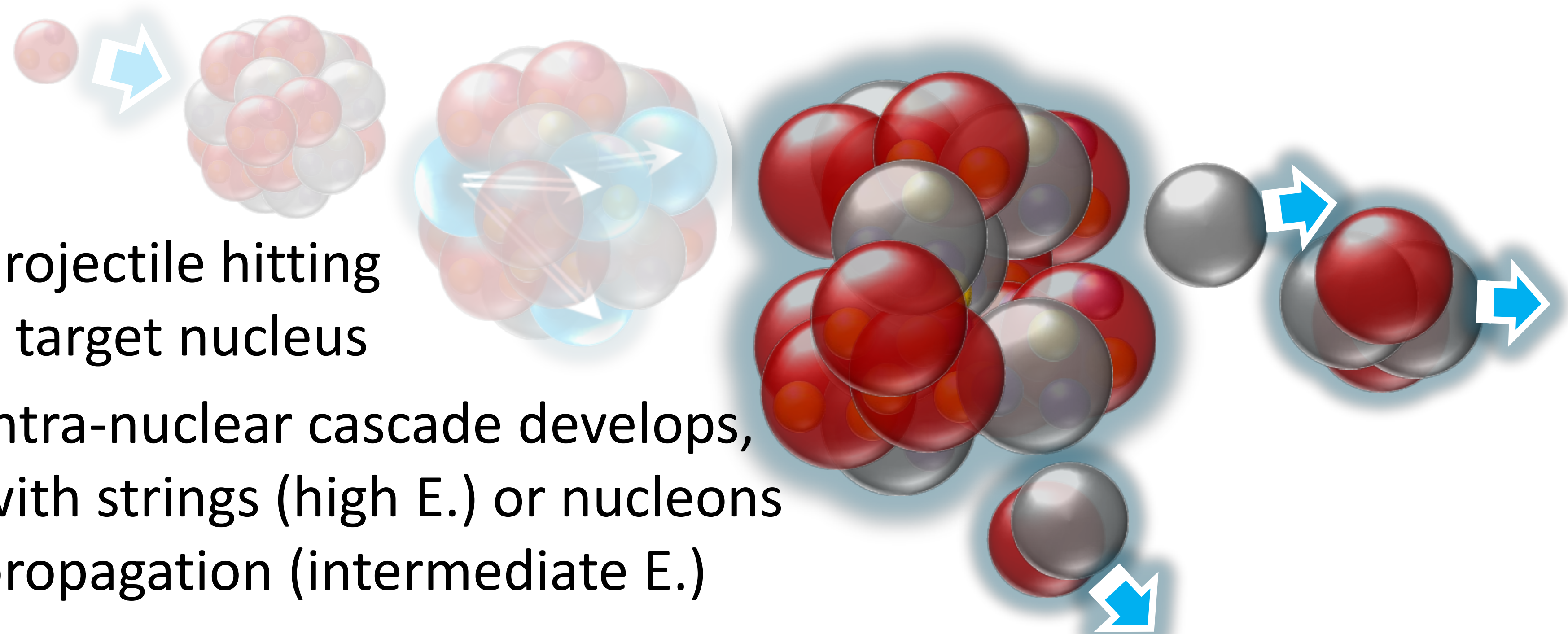


Projectile hitting
a target nucleus



Intra-nuclear cascade develops, with strings (high E.) or nucleons propagation (intermediate E.)

<u>One</u> hadronic collision = sequence of many hadronic interaction<u>s</u>

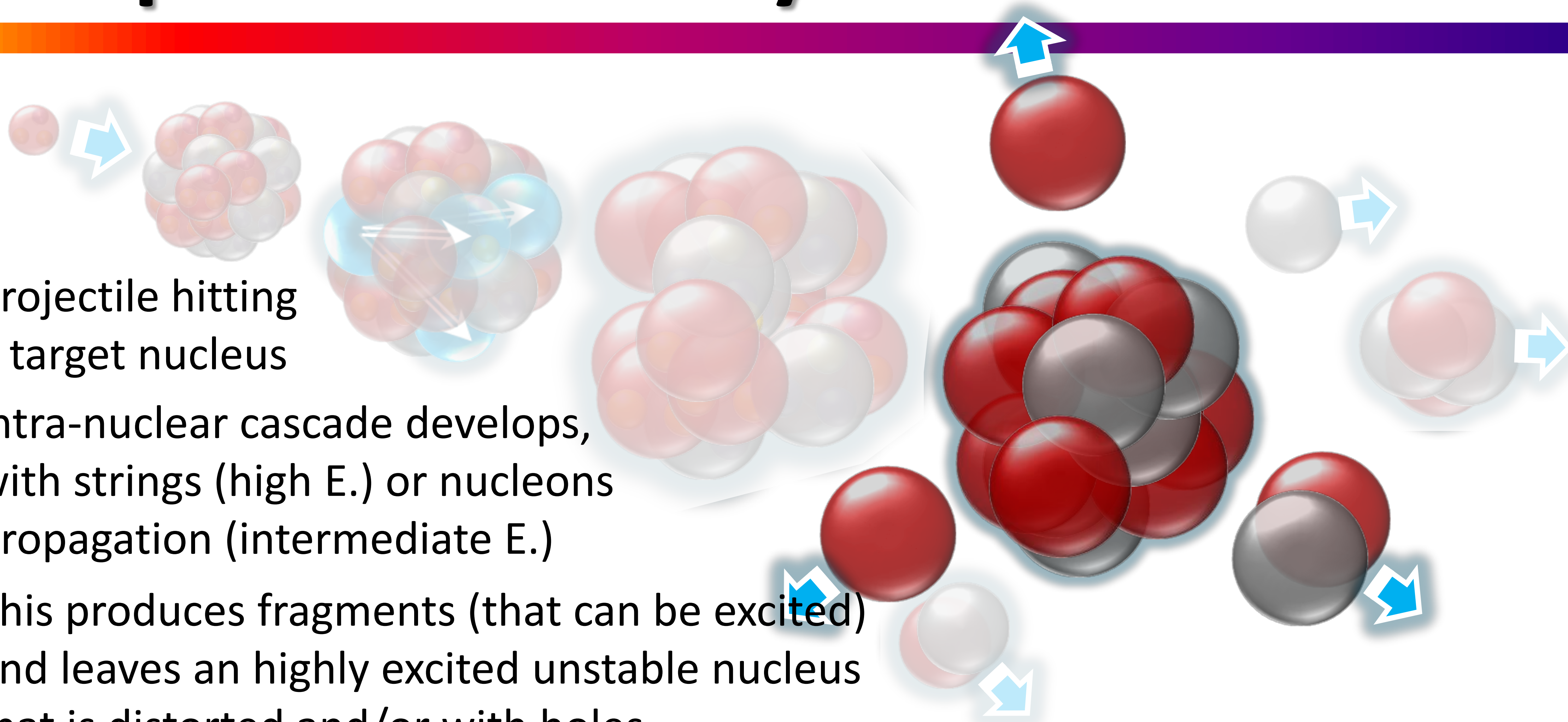


Projectile hitting
a target nucleus

Intra-nuclear cascade develops,
with strings (high E.) or nucleons
propagation (intermediate E.)

This produces fragments (that can be excited)
and leaves an highly excited unstable nucleus
that is distorted and/or with holes.

One hadronic collision = sequence of many hadronic interactions



Projectile hitting a target nucleus

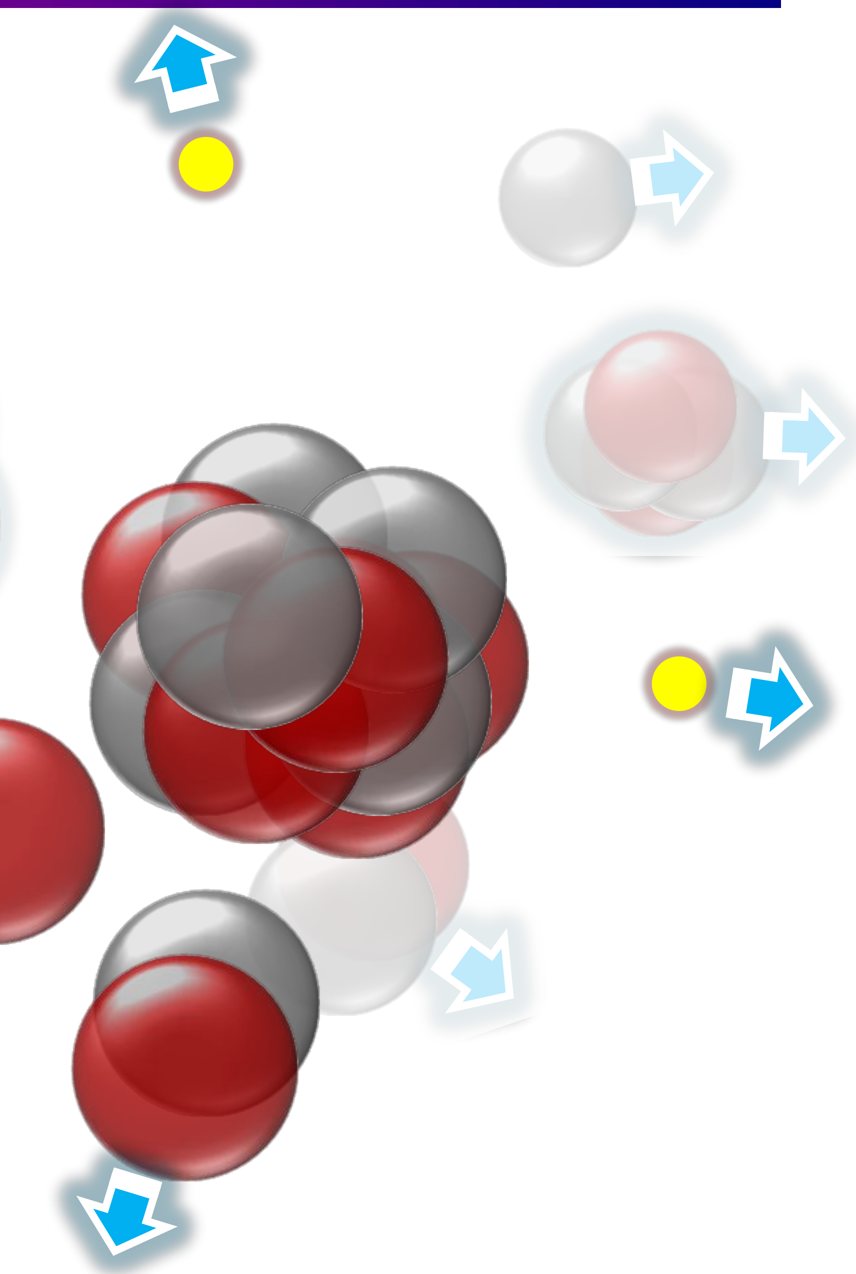
Intra-nuclear cascade develops, with strings (high E.) or nucleons propagation (intermediate E.)

This produces fragments (that can be excited) and leaves an highly excited unstable nucleus that is distorted and/or with holes.

Nucleus rearranges itself by evaporation* and/or fragmentation : this leads to a still excited nucleus, but stable (metastable), and with no memory of the collision history.

(*) **Evaporation** = de-excitation by emission of light nuclei $\in$ {n, p, d, 3d, 3He, a} or photon

<u>One</u> hadronic collision = sequence of many hadronic interaction<u>s</u>



Projectile hitting a target nucleus

Intra-nuclear cascade develops, with strings (high E.) or nucleons propagation (intermediate E.)

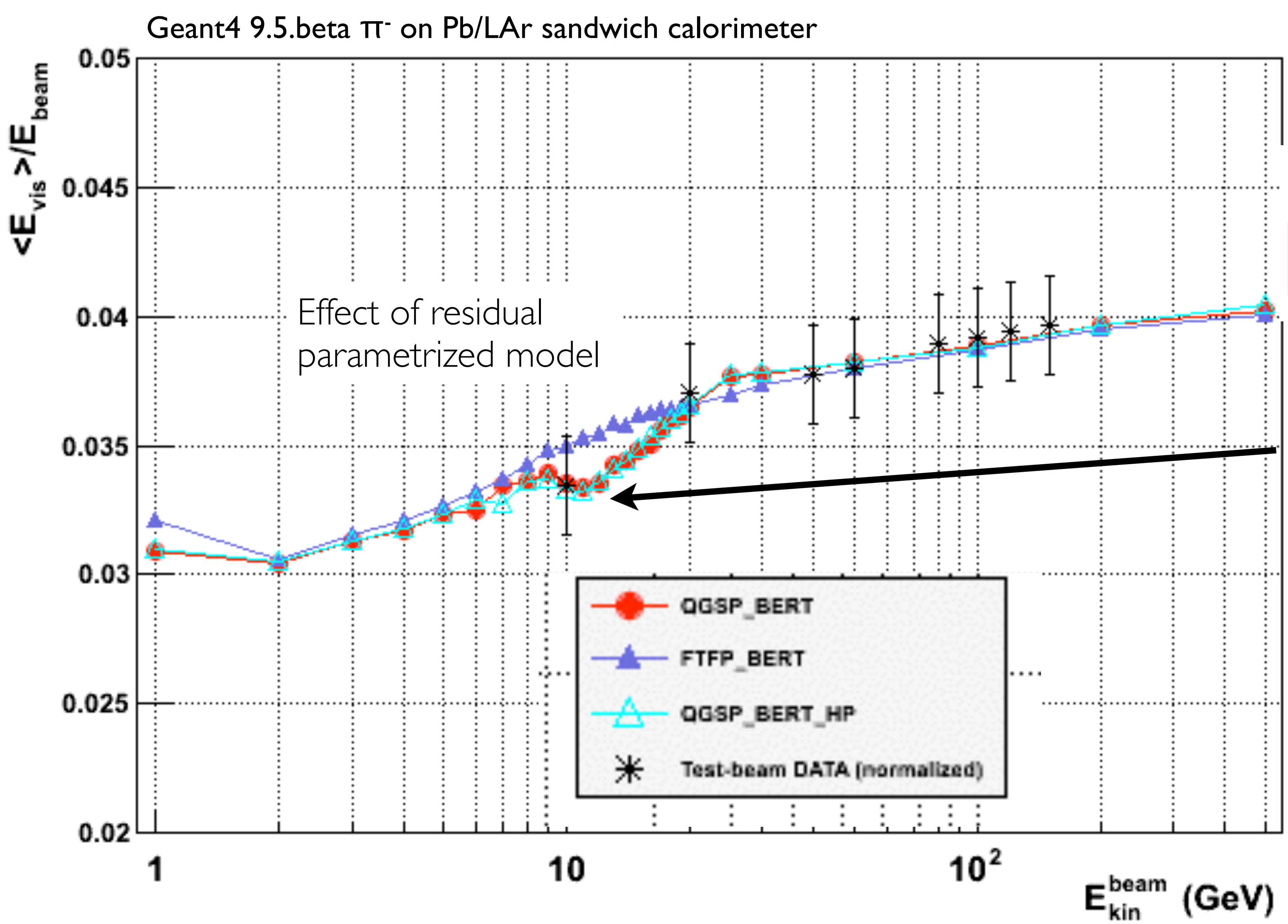
This produces fragments (that can be excited) and leaves an highly excited unstable nucleus that is distorted and/or with holes.

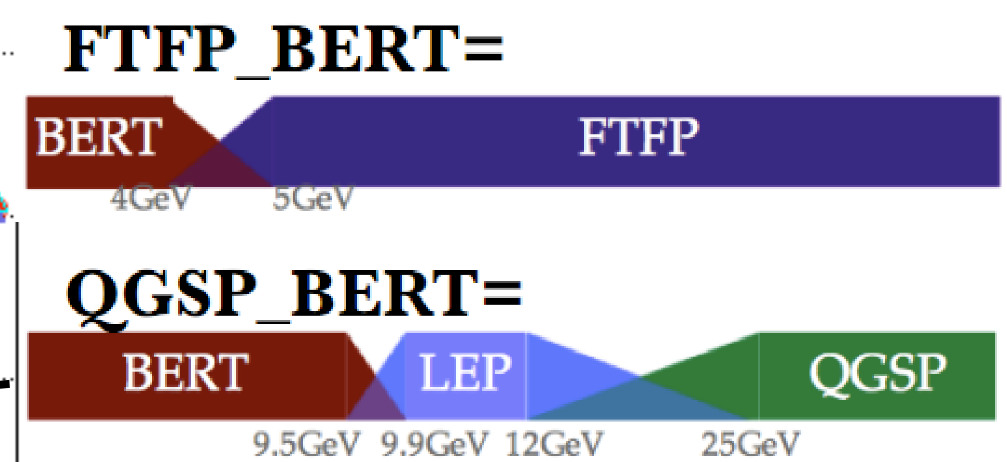
Nucleus rearranges itself by evaporation* and/or fragmentation : this leads to a still excited nucleus, but stable (metastable), and with no memory of the collision history.

The nucleus undergoes final de-excitation by evaporation* or fission and ends-up in its ground state. In case of fission, further de-excitation of fragments may occur.

(*) **Evaporation** = de-excitation by emission of light nuclei $\in$ {n, p, d, t, 3He, α} or photon

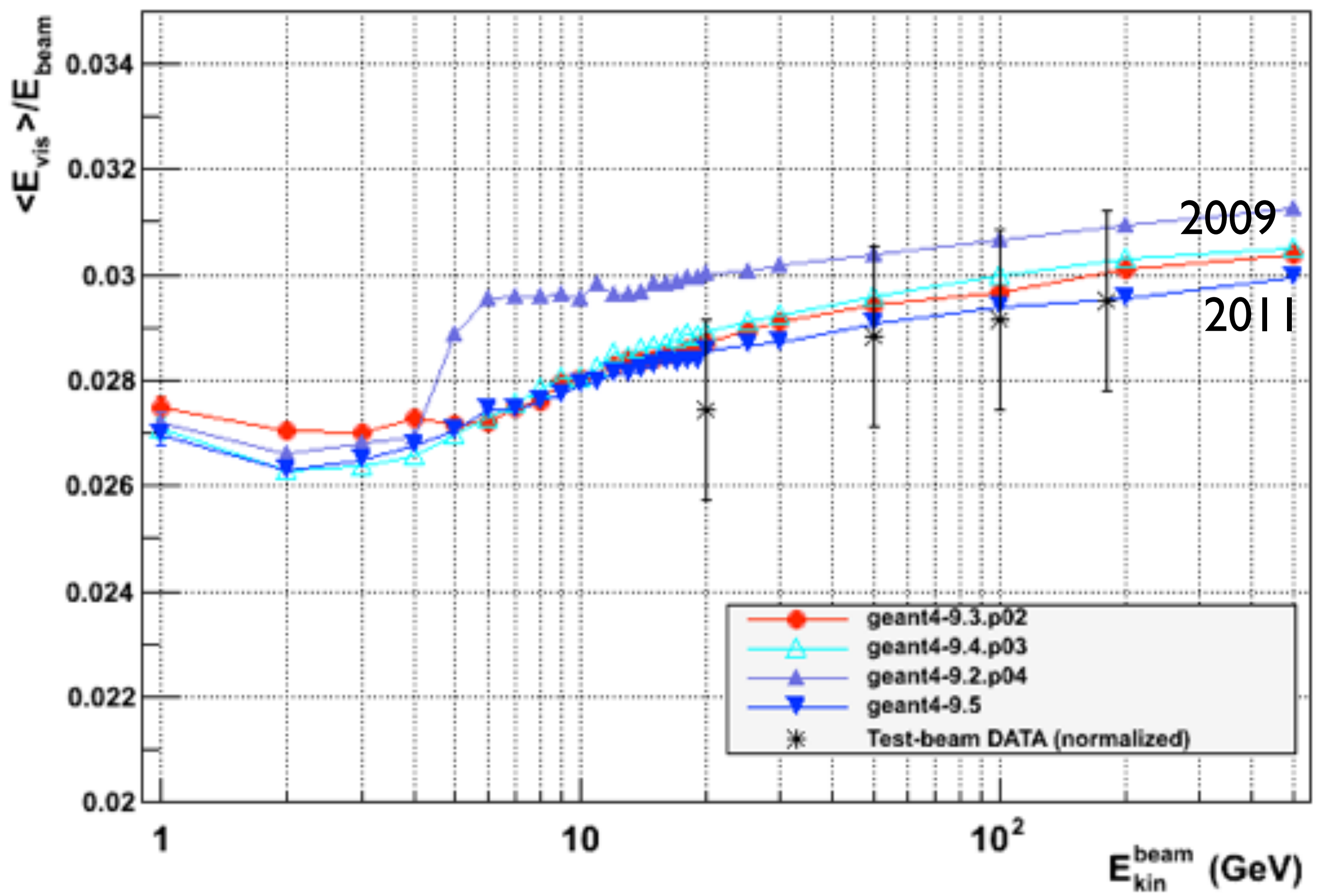
LHC-type calorimeter response





- **Remove LEP** (GHEISHA) → Remove discontinuities
- Reminder of Geant4 strategy: tune on thin-target data, validate with full setups

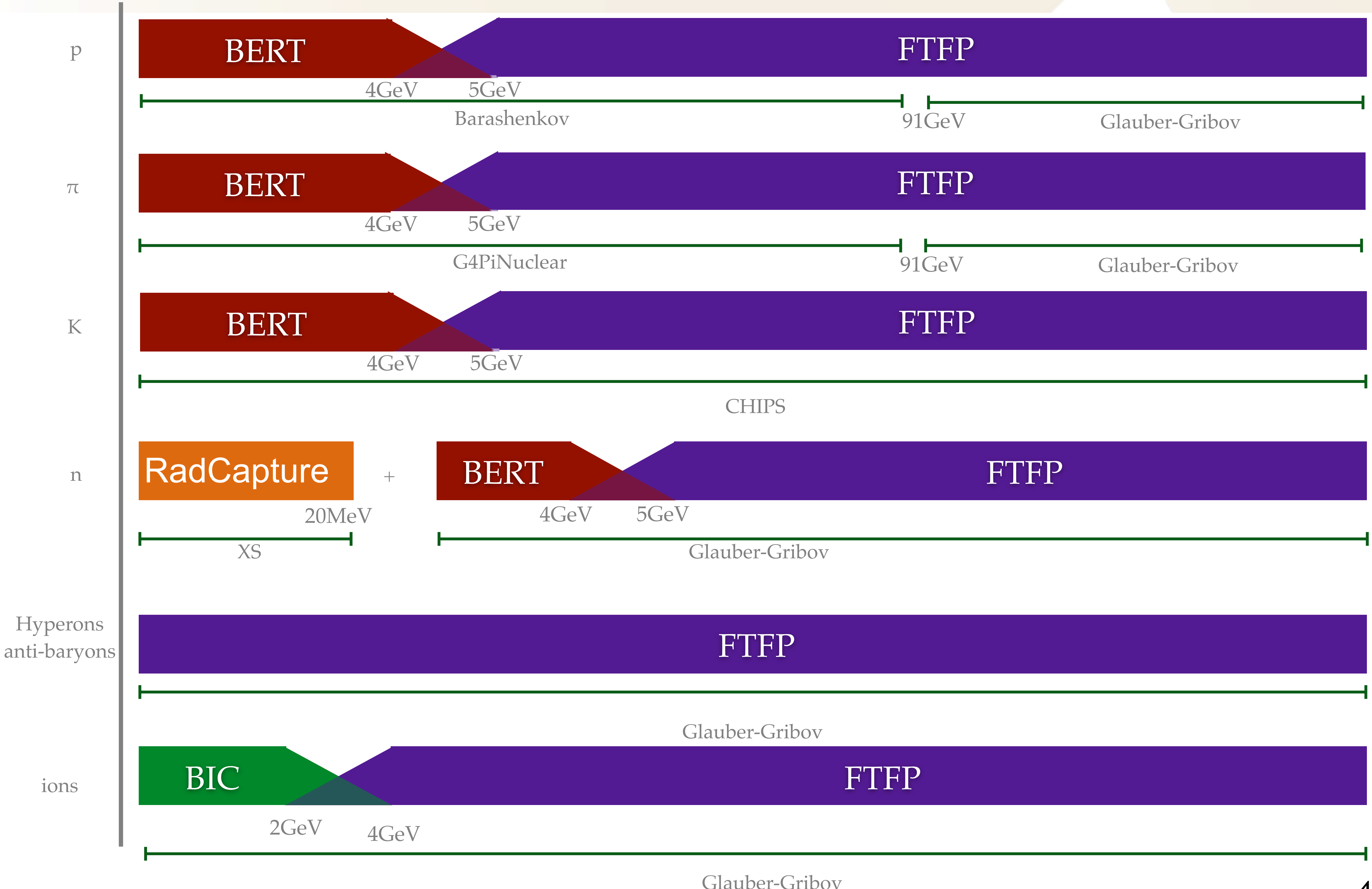
Geant4 π^- on Fe/Sci sandwich calorimeter (ATLAS TileCal)



- Best combination describing simultaneously various calorimeters data (LHC, CALICE, ZEUS)

- Main developments:
  - **Fritiof model** (high energy >4GeV)
  - **Bertini intra-nuclear cascade** (moderate energy <4GeV)
  - **Pre-compound** de-excitation (low-energy <100MeV)

Physics List: FTFP_BERT



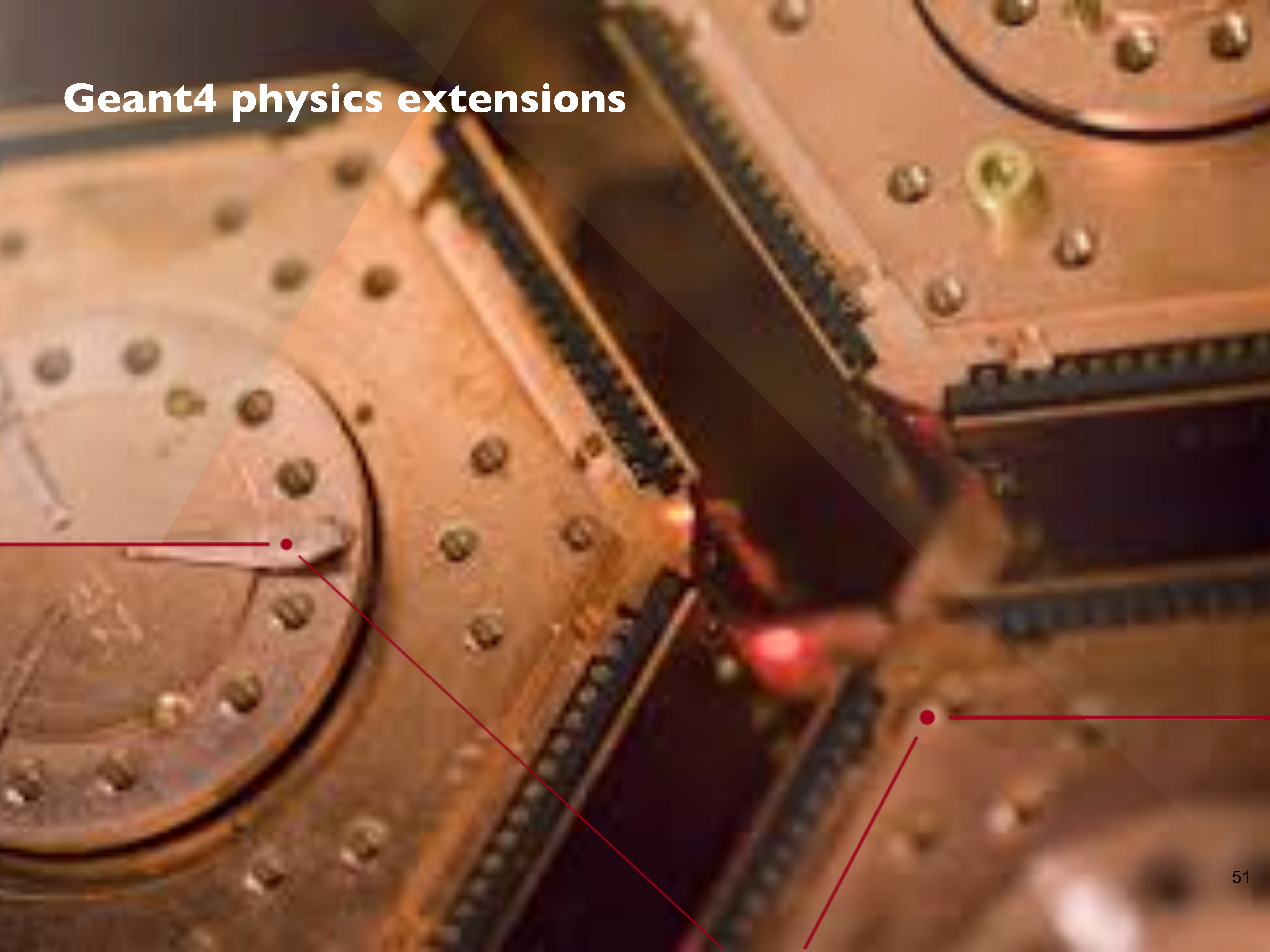
Summary (G4 9.4.p01)

	Response	Resolution	Smoothness	Lateral Shape	Longitudinal Shape @10λ	Notes
QGSP_BERT	+(1-3)%	-(10-5)%	Bad	-(20-10)%	π: -10% p: -20%	anti-nucleons, hyperons via LHEP
FTFP_BERT	+(3-5)%	-(7-3)%	Good	π: -(20-10)% p: -(10-3)%	π: +10% p: +(10-20)%	anti-nucleons, hyperons via CHIPS(*)
CHIPS	+(10-5)%	-(20-10)%	Very Good	π: -(10-3)% p: -(20-10)%	π: -10% p: -20%	native anti-nucleons, hyperons
FTF_BIC(**)	+(3-5)%	-(6-2)%	Bad	-	π: +10%	Implements re-scattering

(*): Native FTF model under development
(**): Much less tested at LHC

OLD results @ LHC first beam
Since then we expect improvements in FTFP_BERT
Obtained with Test-Beam data

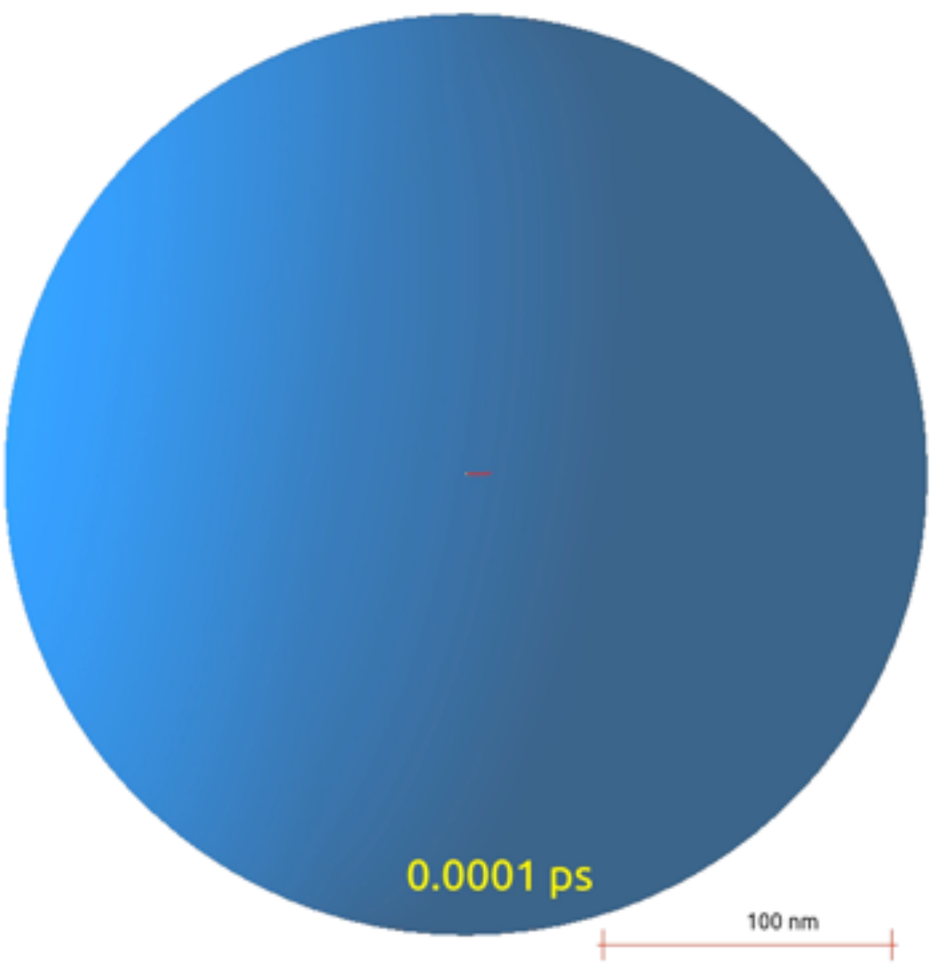
Geant4 physics extensions



Geant4-DNA

http://geant4-dna.org

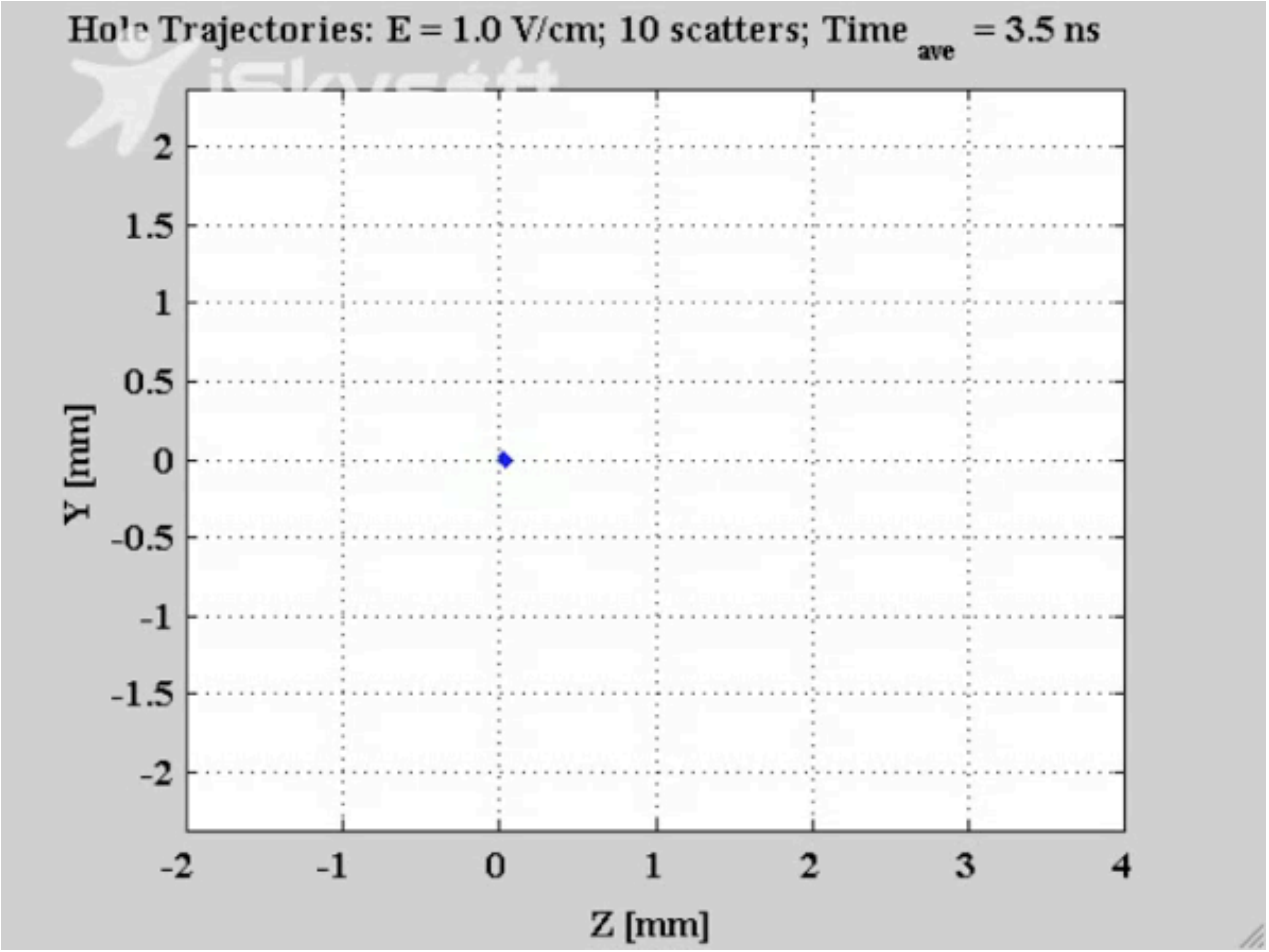
- **Physics phase**: primary radiation interacting with matter (DNA) and producing radicals
- **Chemistry phase**: Brownian motion of radicals (further cell level damage)



Electronic state of H2O	Decay Channel	Fraction (%)
All ionization states	$H_3O^+ + \bullet OH$	100
Excitation state A1B1: (1b1) → (4a1/3s)	$\bullet OH + H\bullet$ $H_2O + DE$	65 35
Excitation state B1A1: (3a1) → (4a1/3s)	$H_3O^+ + \bullet OH + e^-_{aq}$ $\bullet OH + \bullet OH + H_2$ $H_2O + DE$	55 15 30
Excitation state: Rydberg, diffusion bands	$H_3O^+ + \bullet OH + e^-_{aq}$ $H_2O + DE$	50 50

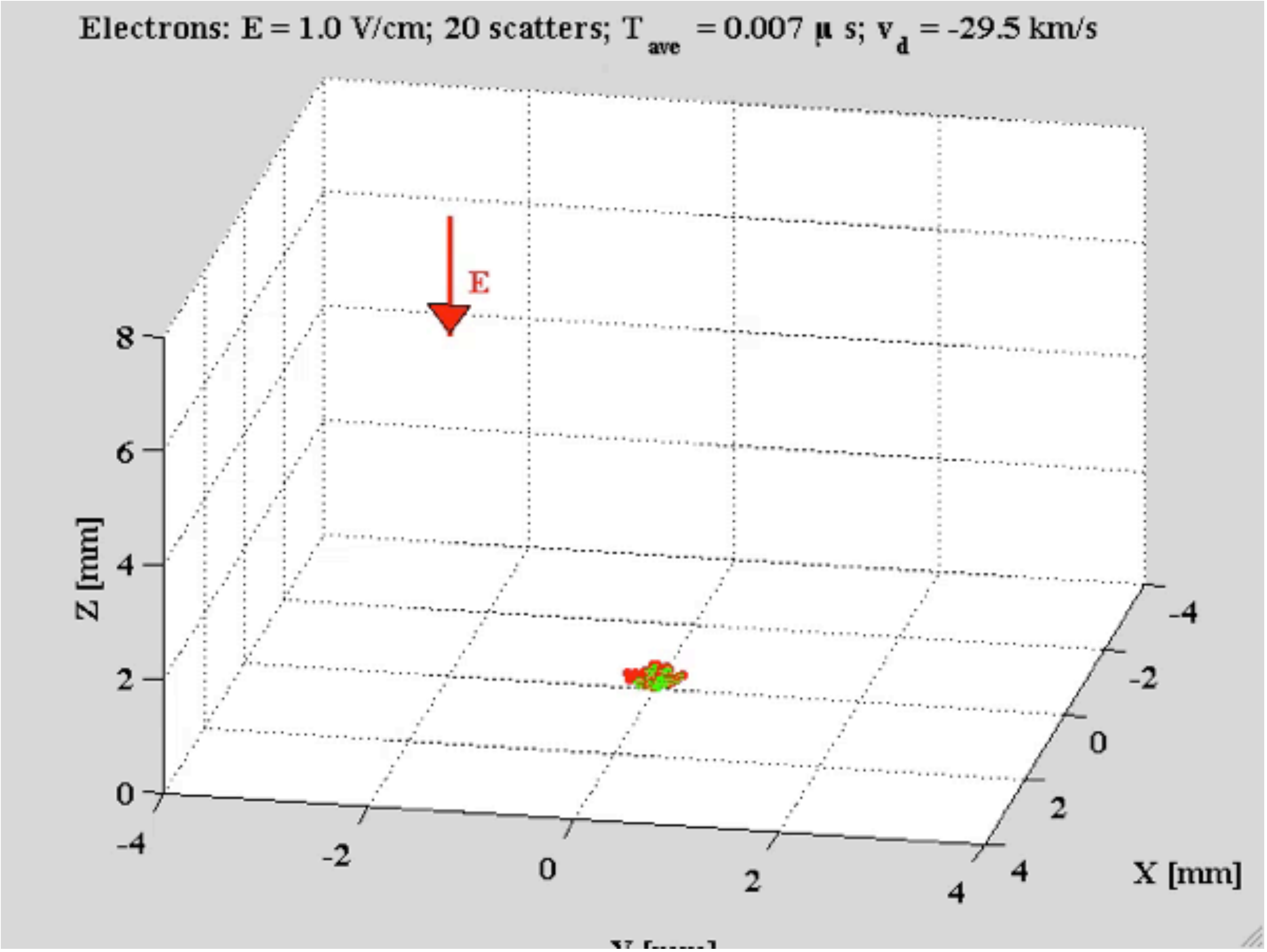
Holes / Phonons simulation

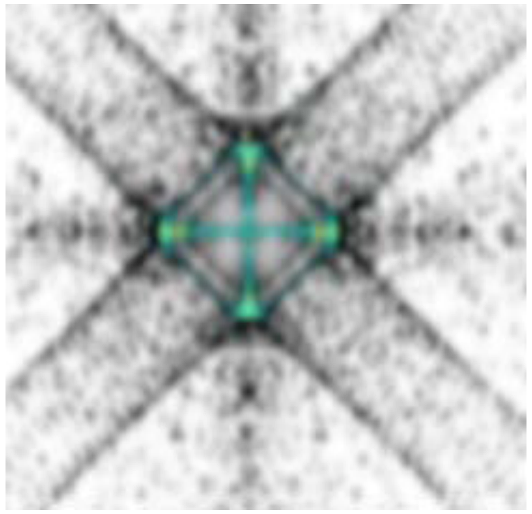
- Simulation of **phonons** in ultra-cold crystals (example available): isotope scattering and mode mixing ; anharmonic down conversion ; reflection processes implemented



Electrons in crystals

- Simulation of **drifting carriers in conduction bands**
  - Oblique propagation implemented ; Luke Scattering under testing ; Inter-valley scattering to be implemented

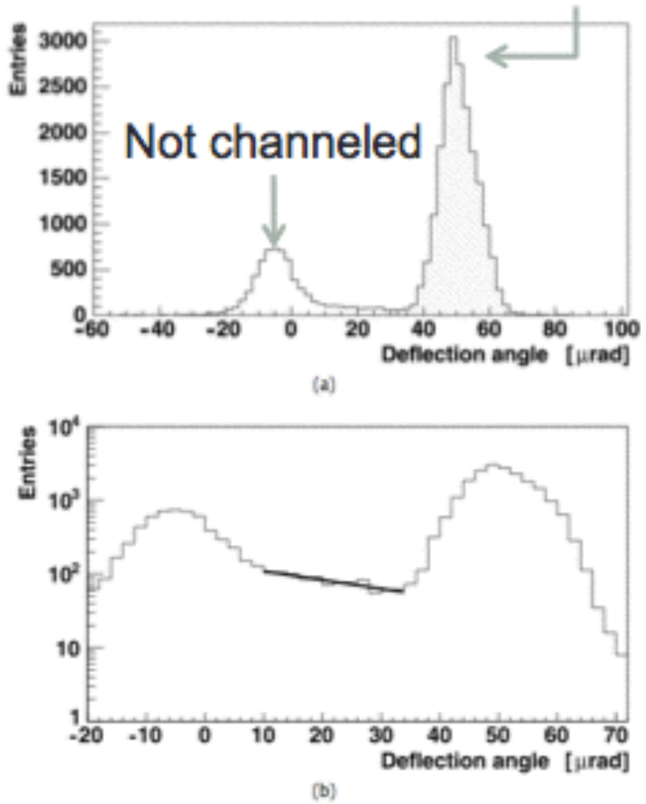




Crystal channeling effect

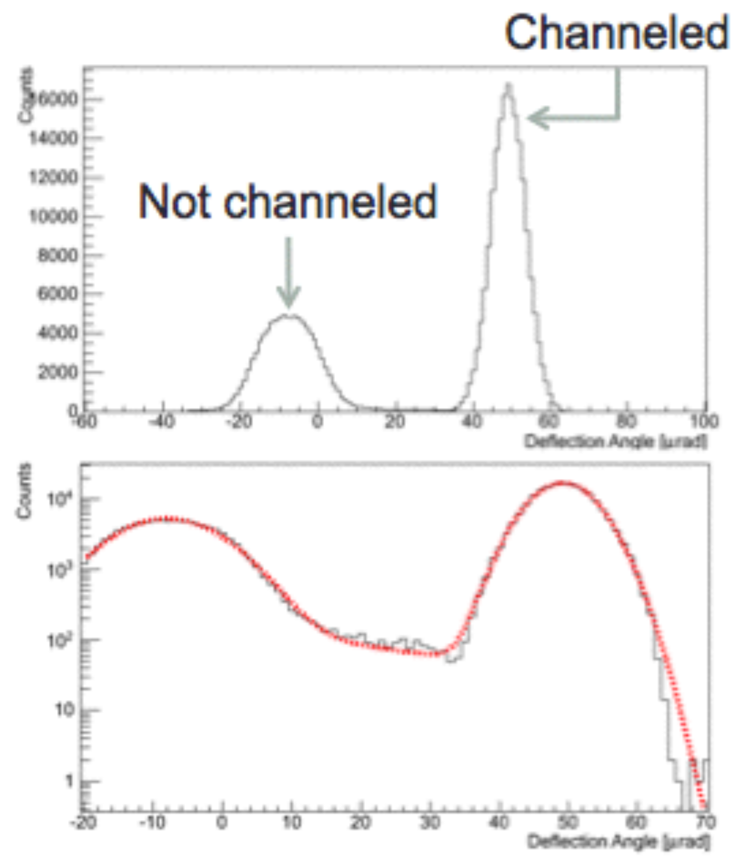
SLAC

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



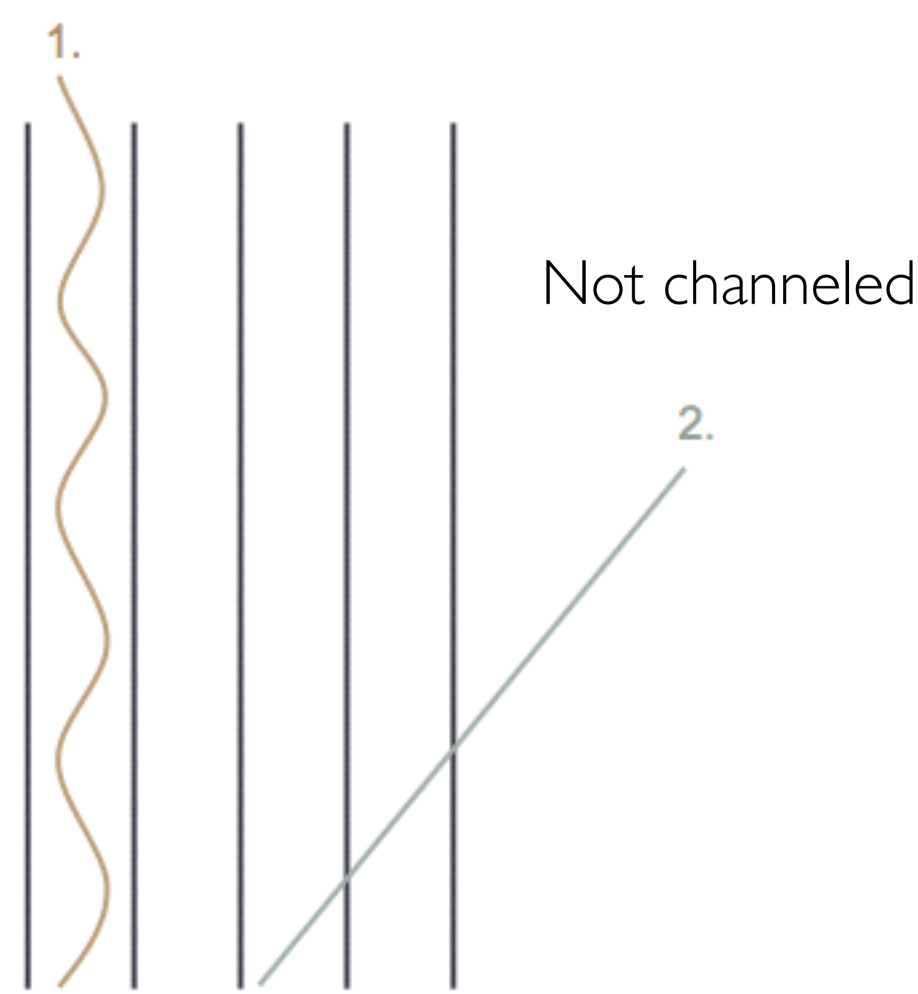
L_n = (1.53 ± 0.35 ± 0.20) mm

Geant4 Channeling



L_n = (1.31 ± 0.05) mm

Particles channeled in atomic crystal planes

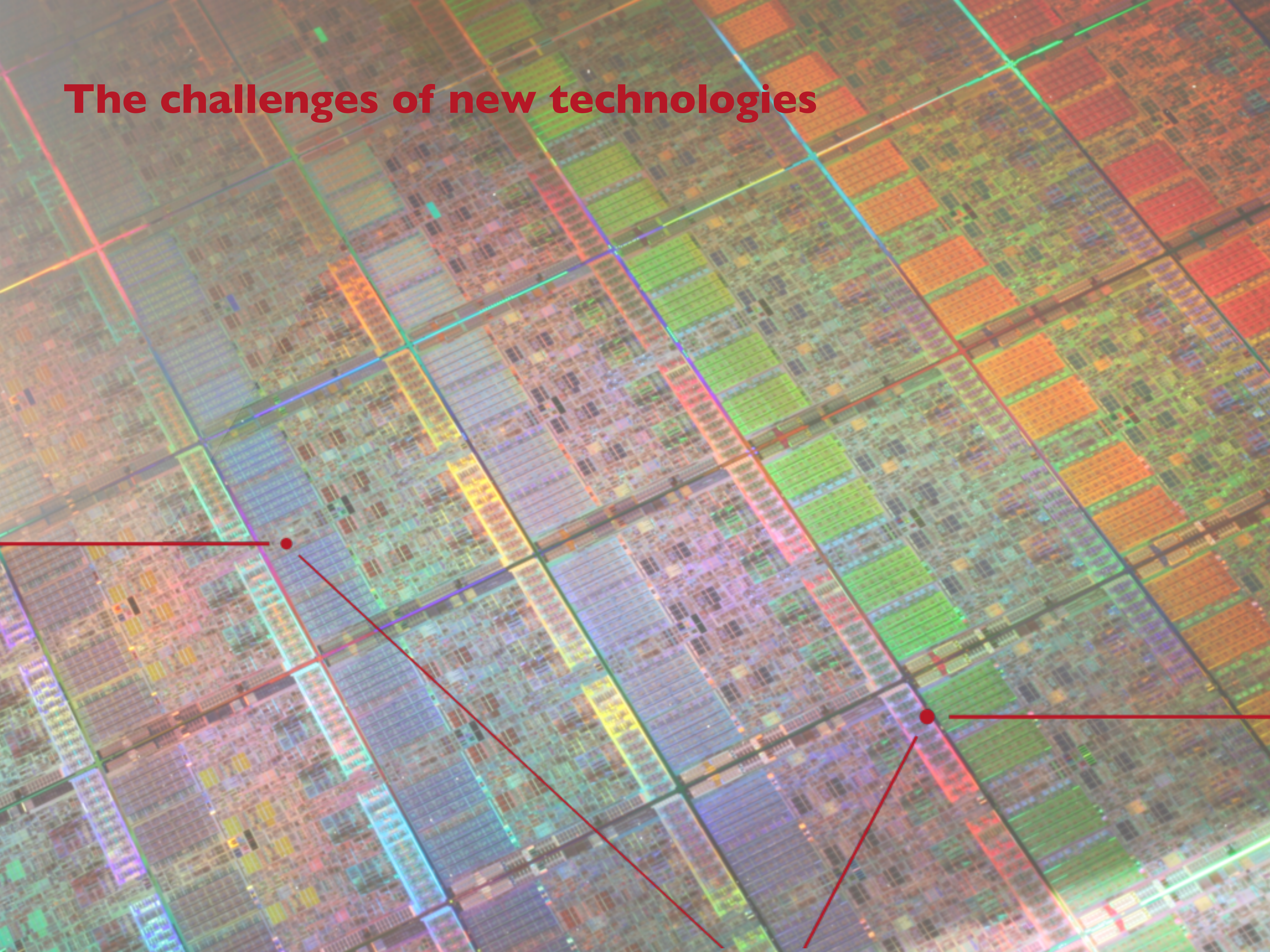


Courtesy of E. Bagli (Ferrara)

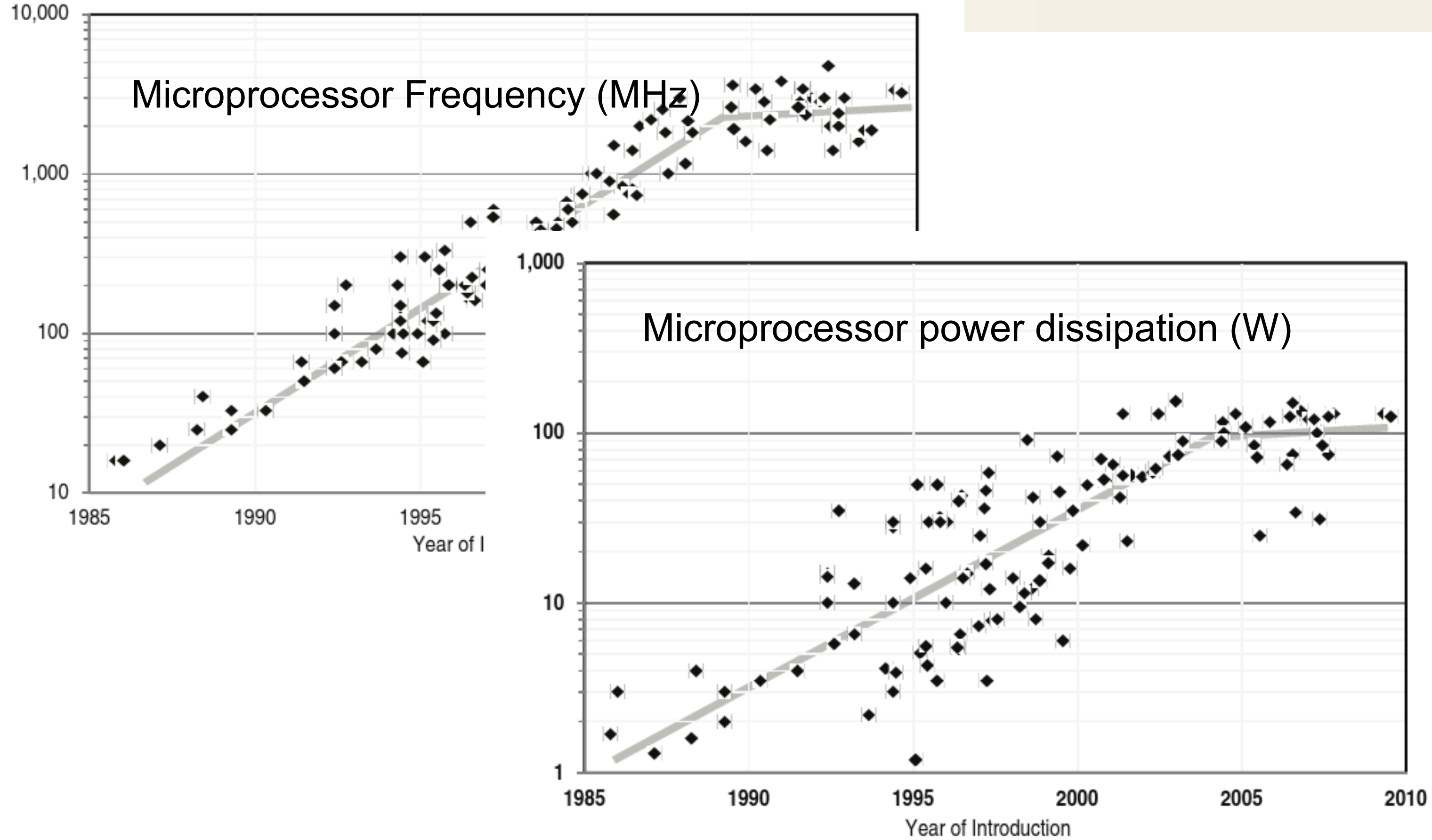
Courtesy of E. Bagli (Ferrara)

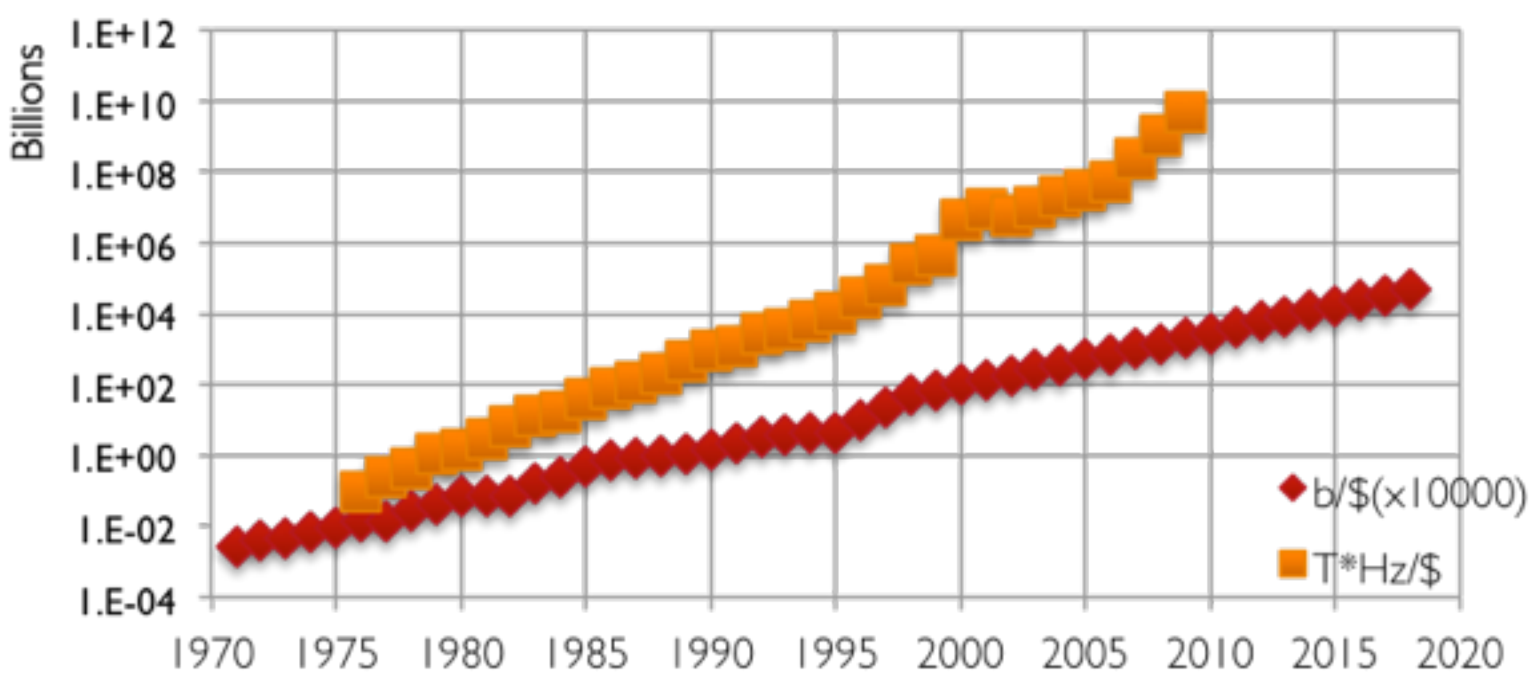
https://indico.cern.ch/getFile.py/access?contribId=105&sessionId=5&resId=0&materialId=slides&confId=250021

The challenges of new technologies



Technology trends

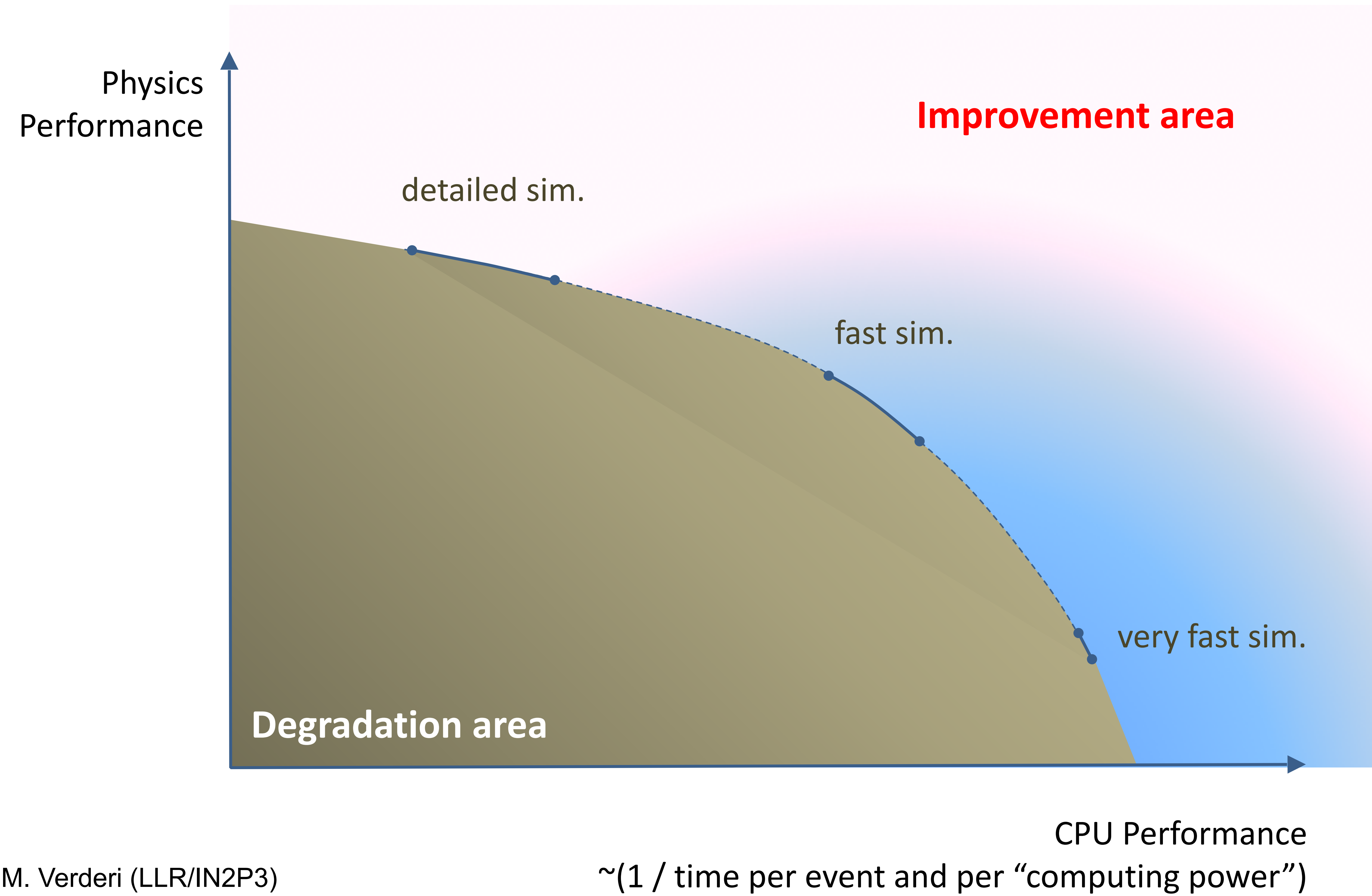




- Increase frequency of CPU causes **increase of power needs**
- Reached plateau around 2005
  - No more increase in CPU frequency
- However number of transistors /$ you can buy continues to grow
  - Multi/May-core era
- Note: quantity memory you can buy with same $ scales slower
- **Expect:**
  - Many core (double/2yrs?)
  - Single core performance will not increase as we were used to
  - Less memory/core
- New software models need to take these into account: increase parallelism

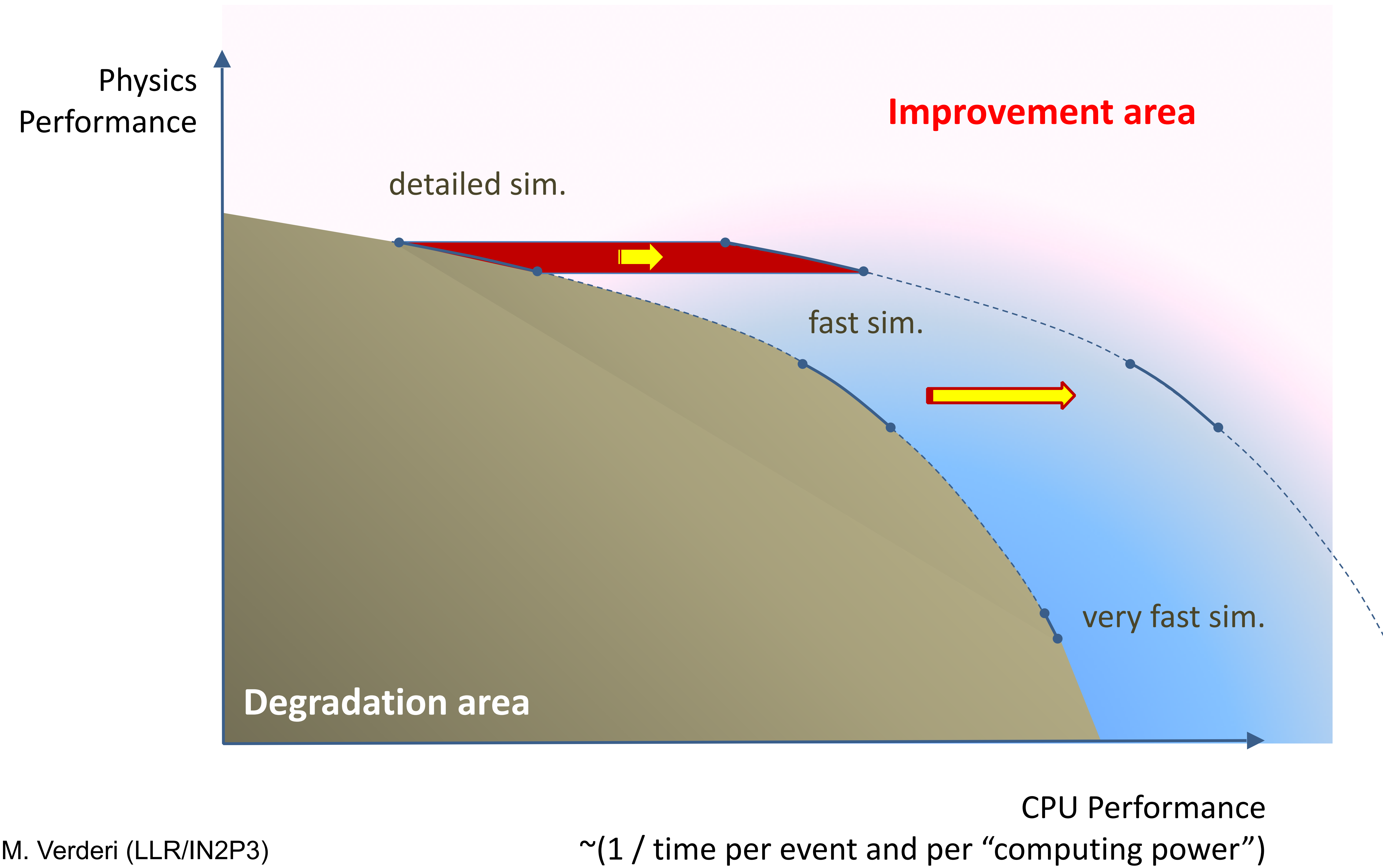
CPU Clock Frequecy 1and usage: The Future of Computing Performance: Game Over or Next Level?
DRAM cost: Data from 1971-2000:VLSI Research Inc. Data from 2001-2002: ITRS, 2002 Update, Table 7a, Cost-Near-Term Years, p. 172. Data from 2003-2018: ITRS, 2004 Update, Tables 7a and 7b, Cost-Near-Term Years, pp. 20-21.
CPU cost: Data from 1976-1999: E. R. Berndt, E. R. Dulberger, and N. J. Rappaport, "Price and Quality of Desktop and Mobile Personal Computers: A Quarter Century of History," July 17, 2000, ;Data from 2001-2016: ITRS, 2002 Update, On-Chip Local Clock in Table 4c: Performance and Package Chips: Frequency On-Chip Wiring Levels -- Near-Term Years, p. 167. ; Average transistor price: Intel and Dataquest reports (December 2002), see Gordon E. Moore, "Our Revolution,"

CPU versus Physics performances



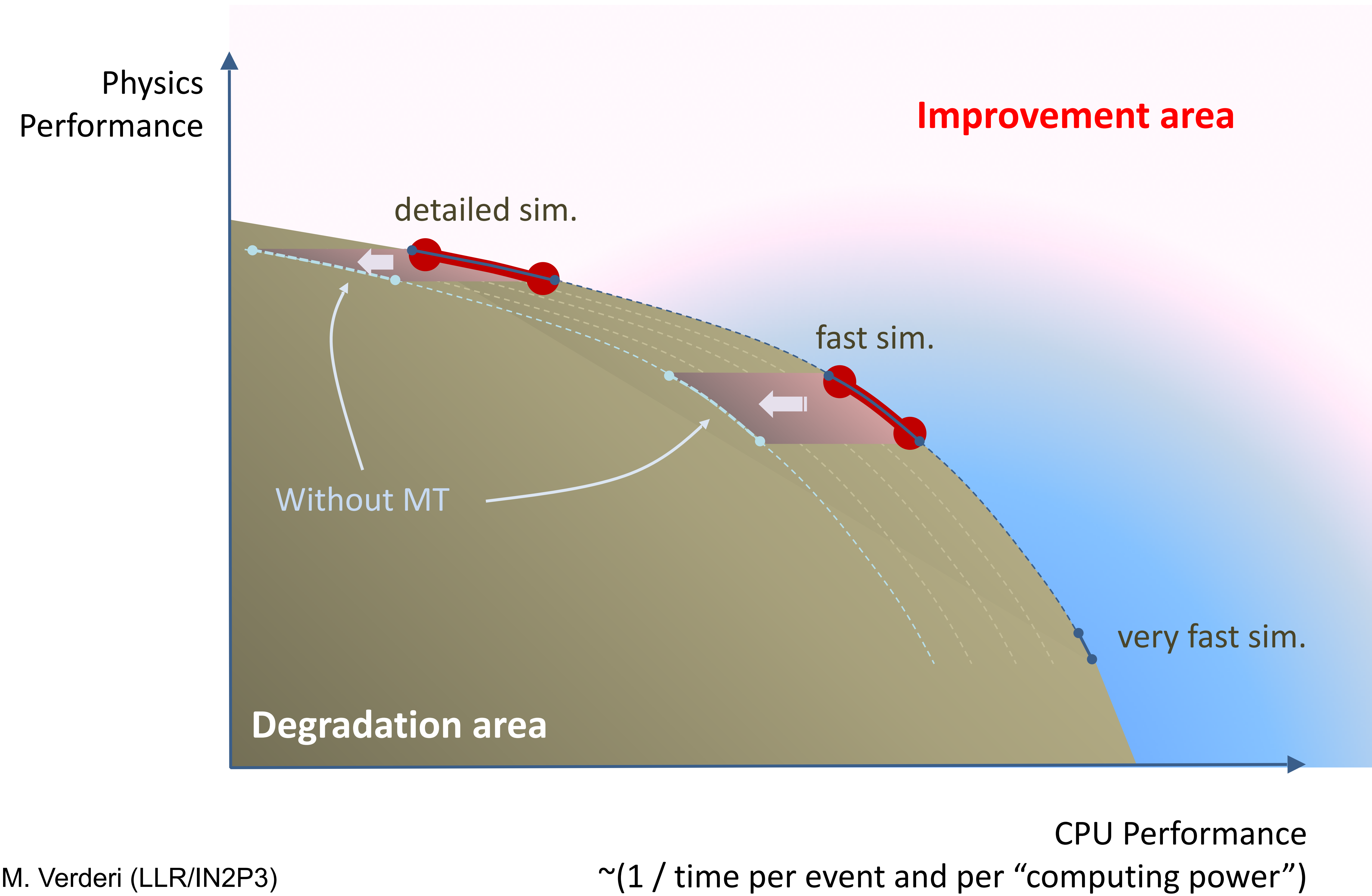
M. Verderi (LLR/IN2P3)

CPU versus Physics performances & new technologies



M. Verderi (LLR/IN2P3)

CPU versus Physics performances & Multi-threading



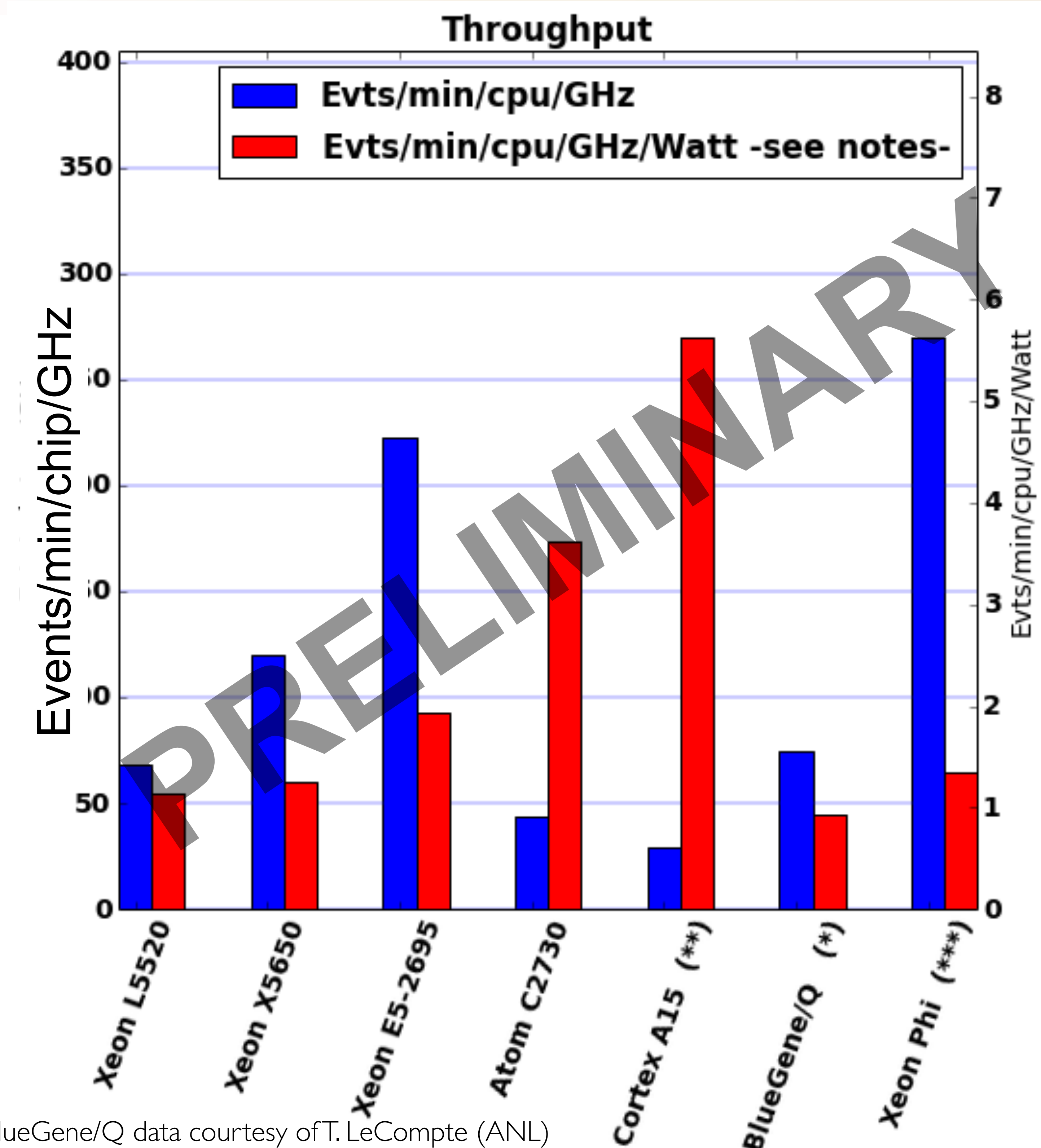
M. Verderi (LLR/IN2P3)

Event Level Parallelism

- **Version 10 supports (optional) event-level parallelism**
  - Can now take advantage of the full CPU power of your machine which likely has more than 1 core
  - You may still opt for a sequential (non-multi-threaded) build (e.g. if you rely on non thread-safe external code)

- Installation
  - No new dependencies, see the Geant4 Installation Guide accessible from the Geant4 web page (User Support -> Documentation -> Installation Guide)
  - Turn on MT via cmake switch
  - See also latest developments and performance at http://twiki.cern.ch/twiki/bin/view/Geant4/MultiThreadingTaskForce

Different Architectures



Geant4 has been run with success on a variety of hardware architectures:

- Intel / AMD
- MIC
- PowerPC (BG/Q)
- ARM / Intel Atom

BlueGene/Q data courtesy of T. LeCompte (ANL)
ARM tests in collaboration with P.Elmer (Princeton;CMS)
Hardware courtesy of OpenLab (CERN)

Obtained with "CMS-style" geometry

Not covered today

- Visualization module: many options are available (shown Qt in demo)
- Scoring: accumulate physics quantities in mesh
- User actions: fine-control of simulation internals
- Analysis module: easy and portable histograms and n-tuple data storing
- Event biasing: focus physics simulation only on important processes or geometry elements (under development)

Conclusions

SLAC

- Geant4 is used in several scientific fields
  - HEP, medical, space
- Rich electro-magnetic and hadronic physics modeling
  - New extensions being developed: DNA-level relevant physics, condensed matter physics (phonons, e/hole transport)
- Powerful geometry, visualization, analysis modules
  - Sorry not time for these today
- First large scale toolkit being migrated to multi-threading technology

- Celebrate with us our 20th anniversary at the "Geant4 Collaboration Meeting, Okinawa (Japan), 29 September - 4 October 2014"