Detector Simulation

Alberto Ribon
CERN PH/SFT

CERN Summer Student Program, 3 August 2015

Foreword

This lecture is aimed to offer a **simple and general introduction** to detector simulation.

Geant4 will be considered as a concrete example (because it is used by the LHC experiments) but only to illustrate general aspects of detector simulation.

This lecture is **not** a tutorial on Geant4 !

(The best way to learn how to use any simulation package is by starting with an example)

Outline

1. **Introduction**
   - Why do we need to simulate a detector?
   - How does it work?
2. **Geometry**
   - How do we describe an experimental apparatus?
3. **Physics**
   - What is available and what to use?
   - What are the challenges?
4. **Validation**
   - How can we trust a few million lines of code?

Introduction

Introduction

- Simulation is a very useful, essential tool in modern particle physics for:
  - **designing** an experiment (e.g. now ILC/CLIC, FCC)
  - **analysing** the data (e.g. now LHC experiments)

- For the LHC experiments, the simulation is made of two distinct steps:

  1. Simulation of the p-p collision
  - Monte Carlo event generators

  2. Simulation of the passage of the produced particles through the experimental apparatus
  - Monte Carlo radiation transportation, or simply “detector simulation”
  - From the beam pipe to the end of the cavern
  - The output of 1. is the input of 2.

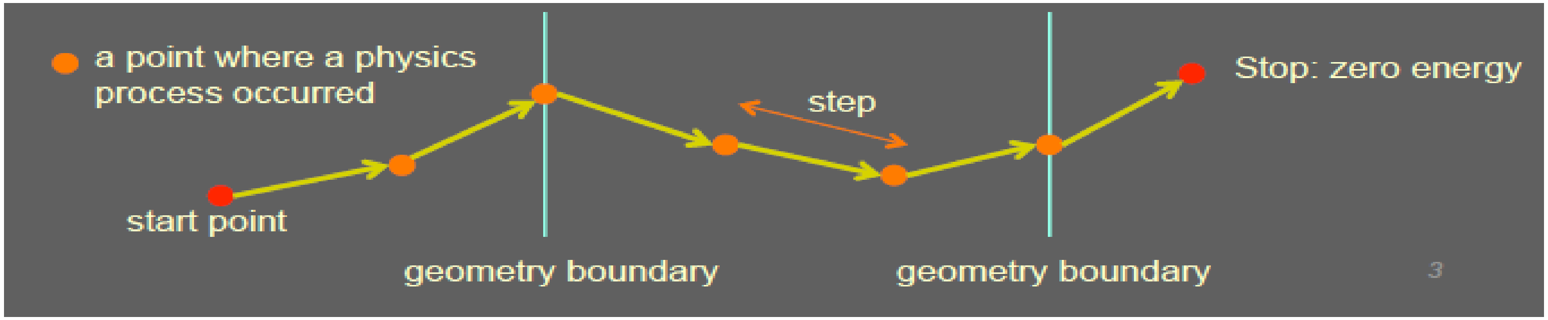
Monte Carlo radiation transportation codes

- The simulation of the p-p collision is the same for different experiments at the same collider, e.g. ATLAS and CMS
- The detector simulation is different for each experiment. However, general codes exist that can be used for simulating any detector
  - An experimental apparatus can be modeled in terms of elementary geometrical objects
  - The physics processes are detector independent

- These general codes, e.g. Geant4, are called "**Monte Carlo radiation transportation codes**"
  - Non-deterministic (e.g. do not solve equations); use random numbers to reproduce distributions
  - Transport particles through matter

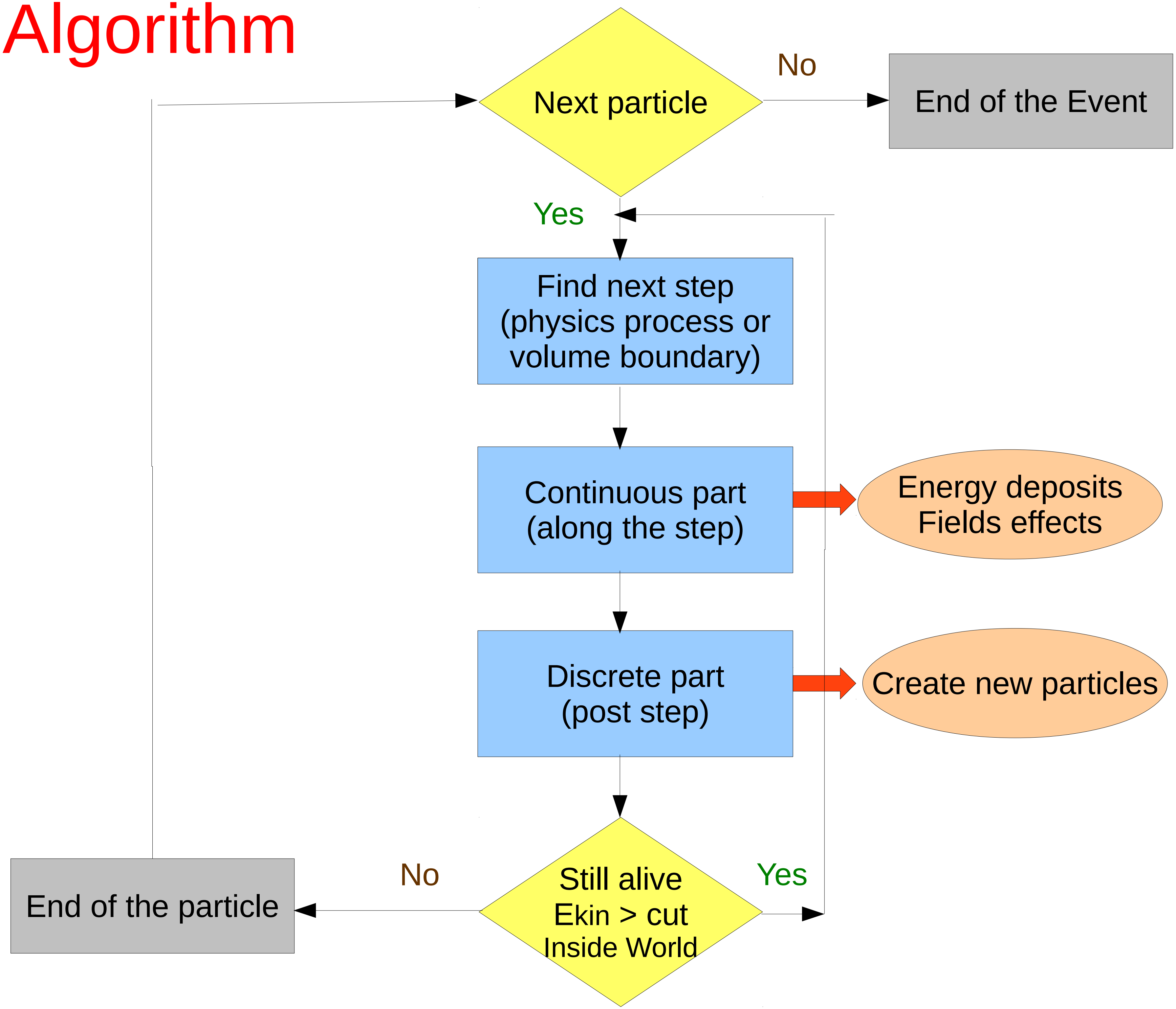
How does it work?

- Treat one particle at the time
- Treat a particle in **steps**



- For each step
  - the step length is determined by the cross sections of the physics processes and the geometrical boundaries; if new particles are created, add them to the list of particles to be transported;
  - local energy deposit; effect of magnetic and electric fields;
  - if the particle is destroyed by the interaction, or it reaches the end of the apparatus, or its energy is below a (tracking) threshold, then the simulation of this particle is over;
    else continue with another step.
- Output - new particles created (indirect)
  - local **energy deposits** throughout the detector (direct)

Algorithm


Next particle
No
End of the Event
Yes
Find next step (physics process or volume boundary)
Continuous part (along the step)
Energy deposits Fields effects
Discrete part (post step)
Create new particles
No
Still alive
Ekin > cut
Inside World
Yes
End of the particle

“Digitization”

- Besides the geometry, another experiment-specific aspect of the detector simulation is the “digitization”
  - It is not part of the general radiation transportation codes
- It consists of producing the detector response in terms of electric current & voltage signals, as in the real experiment
  - The same reconstruction chain can be applied for both real and simulated data
- The general radiation transportation code provides energy deposits in the whole detector;
  from these, the “digitization” simulates the electrical signals induced in the sensitive parts of the detector
- Another detector-specific aspect is the “**pile-up**” ...

Accuracy *vs.* Speed

- Huge samples (billions) of simulated events are needed by the experiments for their physics analyses
- The number of simulated events is limited by CPU
- The simulation time is dominated by the detector simulation
- Tradeoff between **accuracy** and **speed** of the detector simulation
  - More precise physics models are slower and, more importantly, create more secondaries and/or steps
  - Smaller geometrical details slow down the simulation
    - Never model explicitly screws, bolts, cables, etc.
  - Continuous spectrum of types of detector simulations
    - From full, detailed detector simulations (covered in this lecture)
    - To very fast, fully parametrized detector simulations (not covered here!)
- *On-going effort to exploit the latest CPU features*
  - *Multi-threading (e.g. G4 10) ; Vectorization (e.g. Geant-V)*

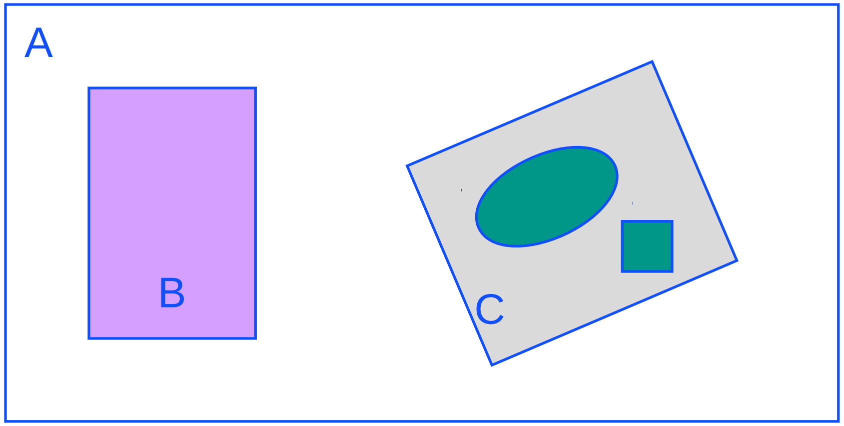
Application domains

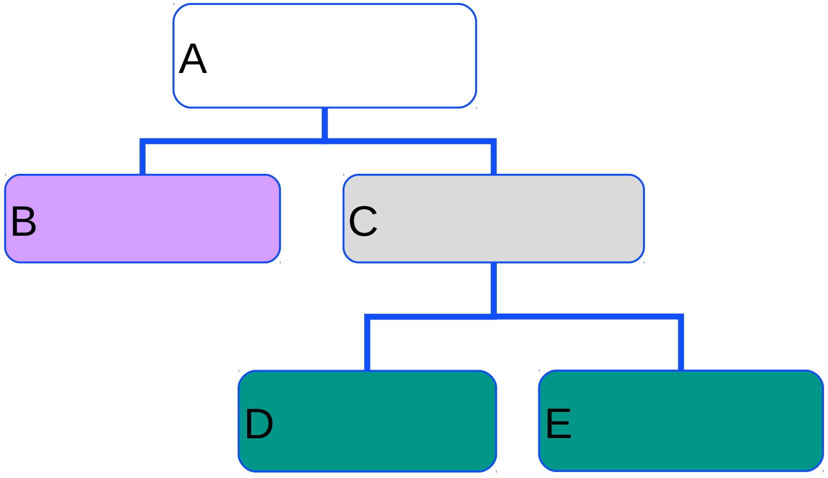
- We are considering here mainly high-energy physics, but...
- There are other domains where the same radiation transportation codes are successfully used:
  - Nuclear physics
  - Accelerator science
  - Astrophysics
  - Space engineering
  - Radiation damage
  - Medical physics
  - Industrial applications
- So, detector simulation is a multi-disciplinary field!

Geometry

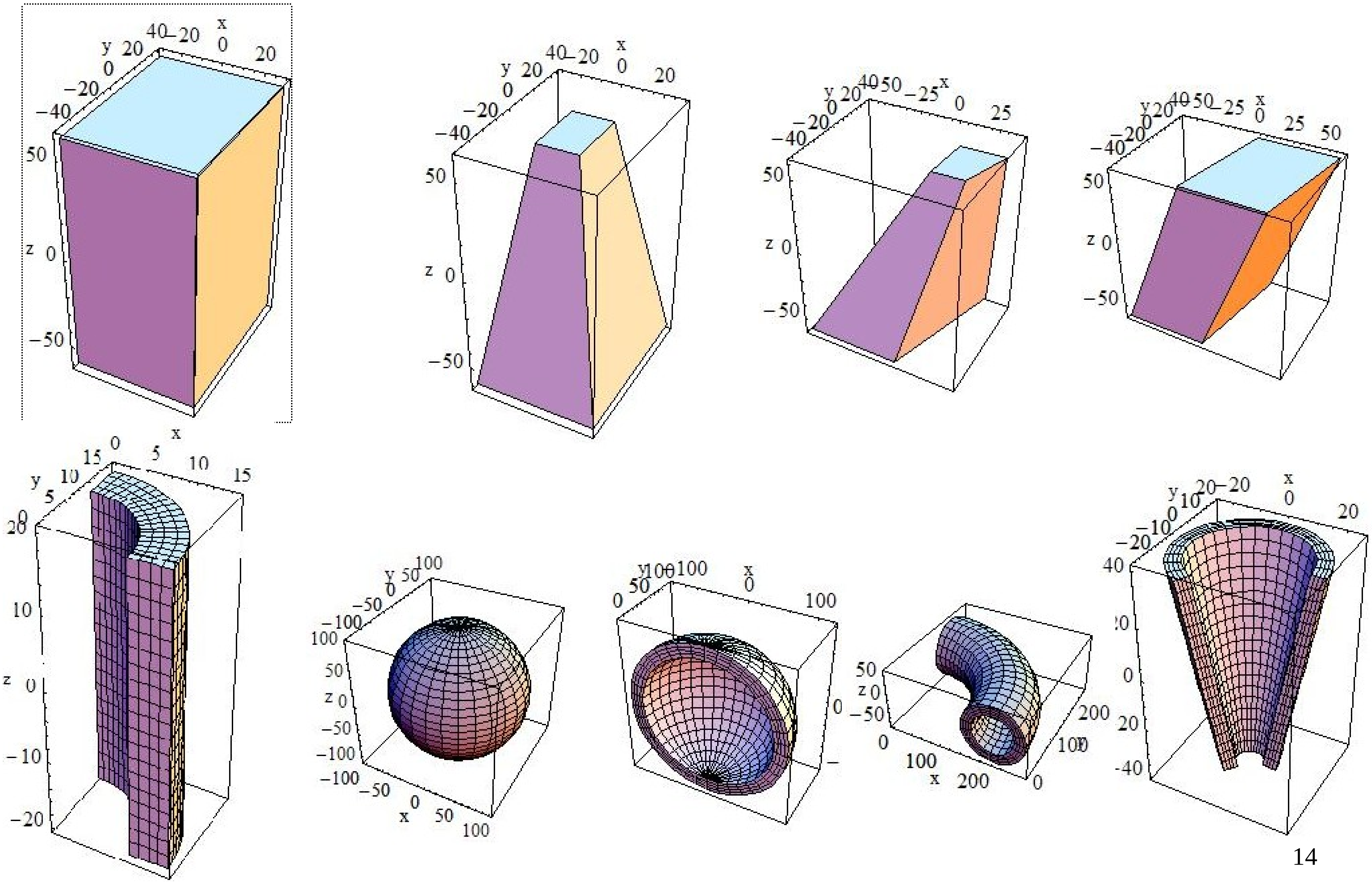
Geometry

- The way to describe the geometry varies widely between the different simulation engines
  - In Geant4, you need to write some C++ code
    - Geometry objects are instances of classes
    - Geometry parameters (e.g. dimensions) are arguments of the constructors
- The geometry can be “flat” or “hierarchical”
  - In Geant4, it is hierarchical: a volume is placed in its mother volume; there are mother-daughter relationships
- A **material** should be assigned to each volume

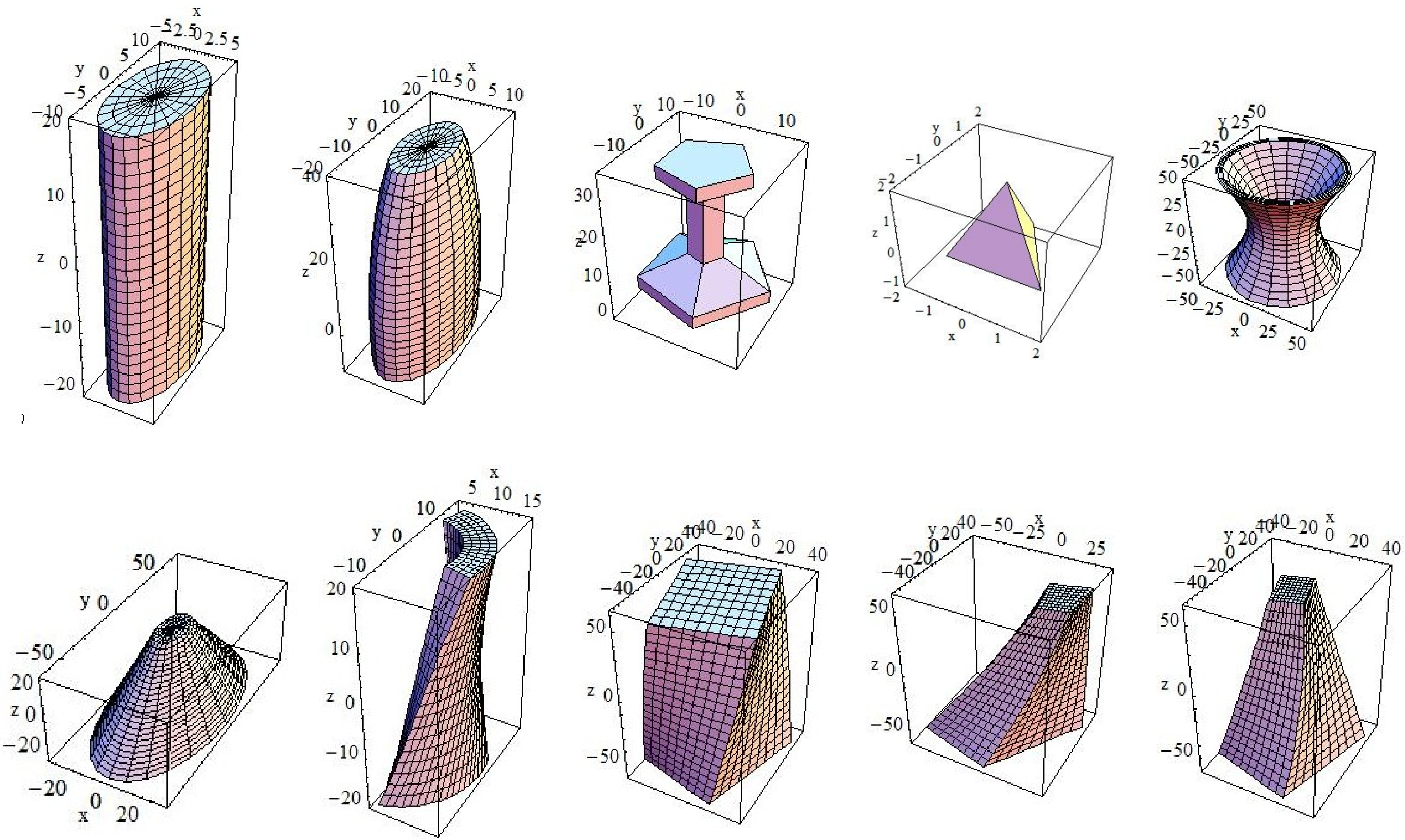




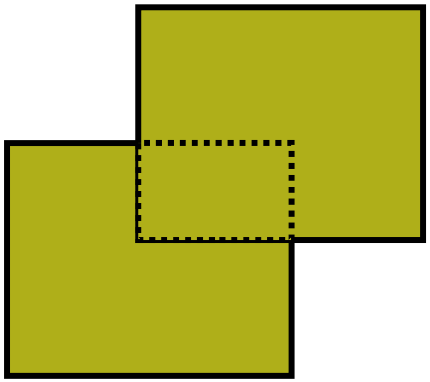
CGS (Constructed Geometry) Solids



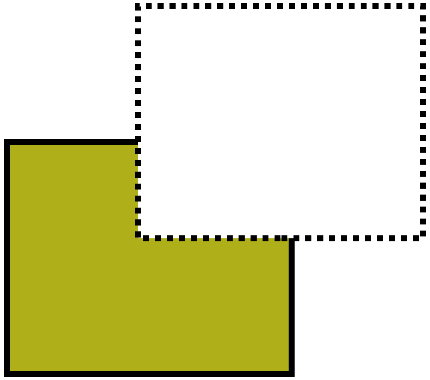
Other CGS solids



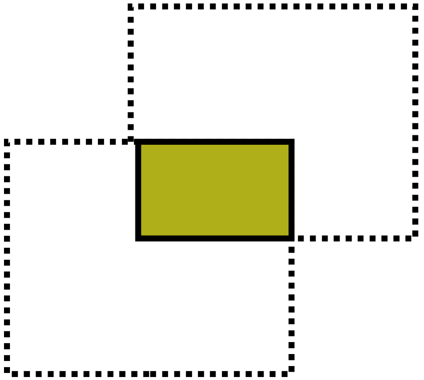
Boolean solids



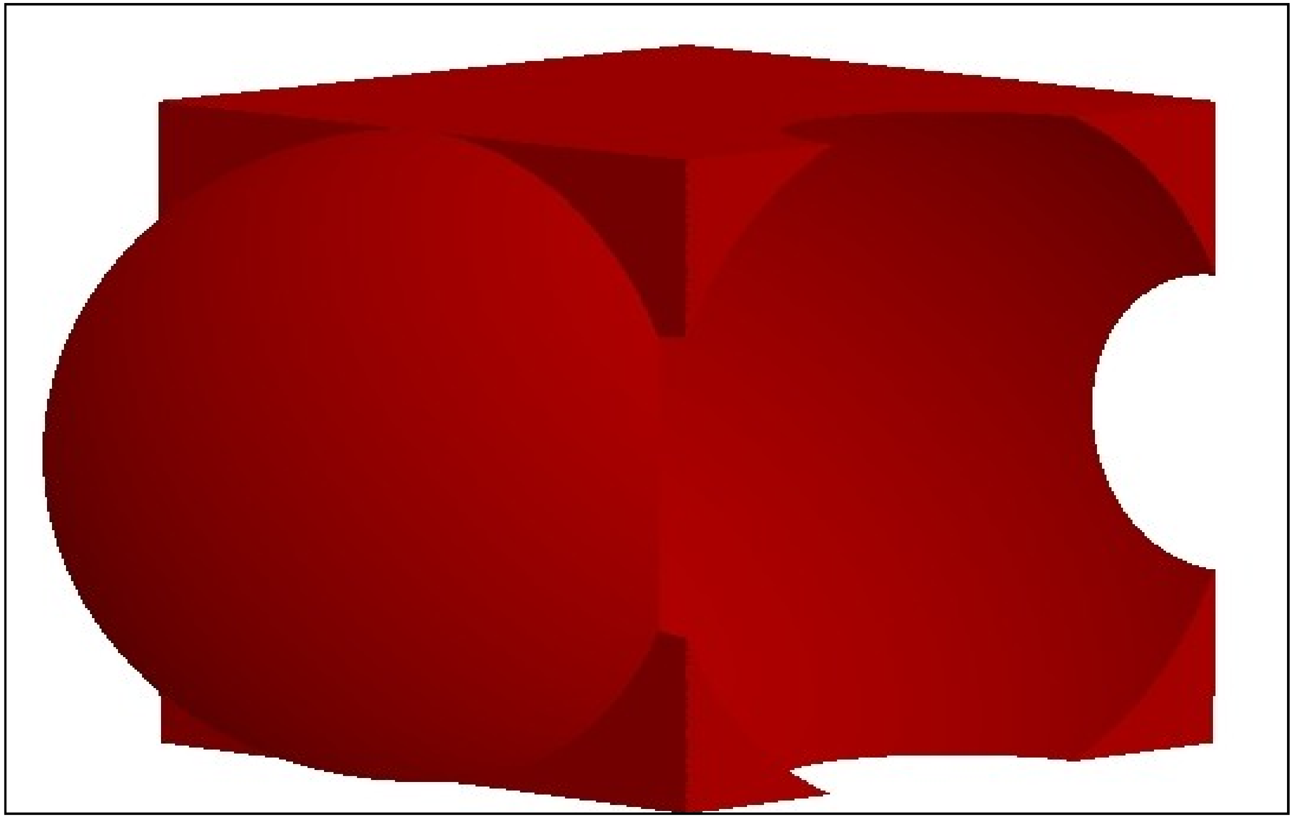
Union



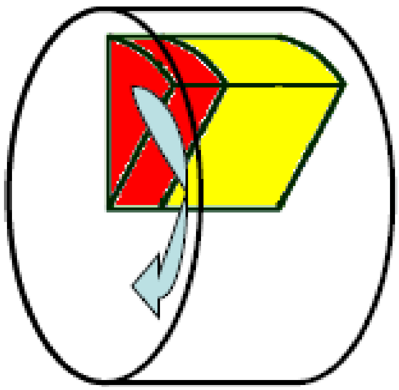
Subtraction

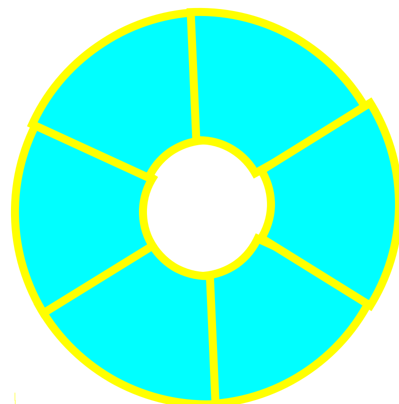


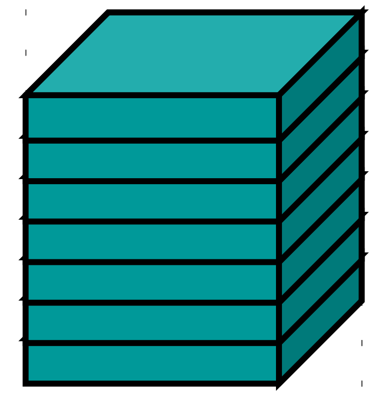
Intersection

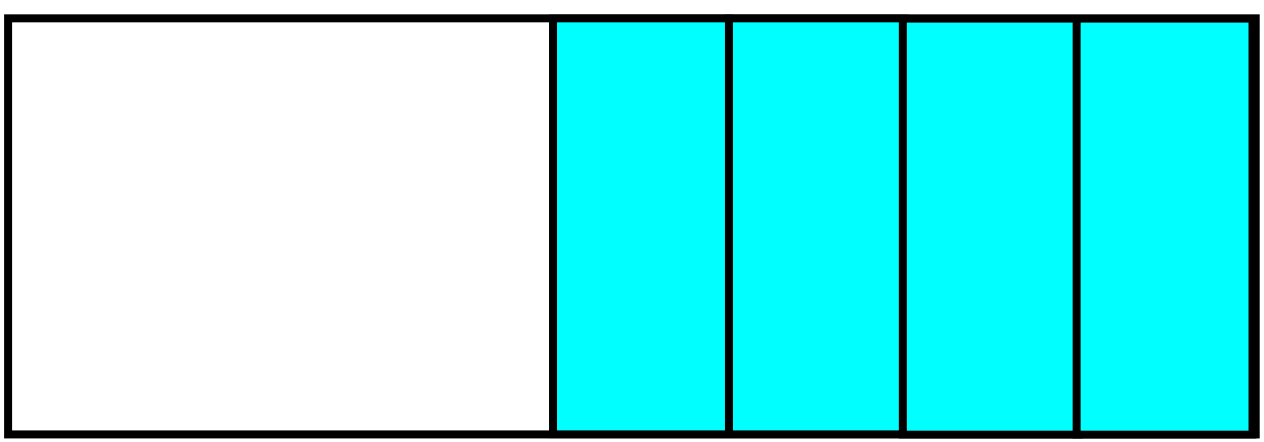


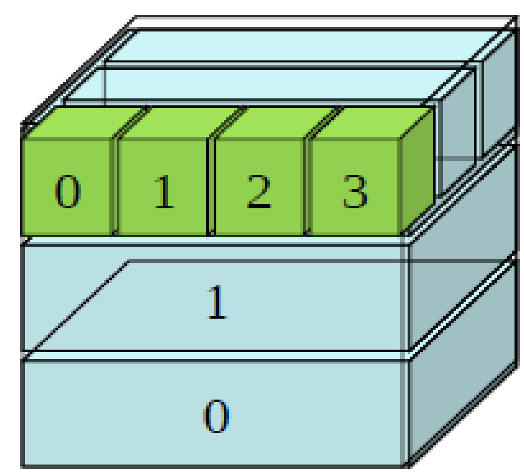
Geometrical symmetries



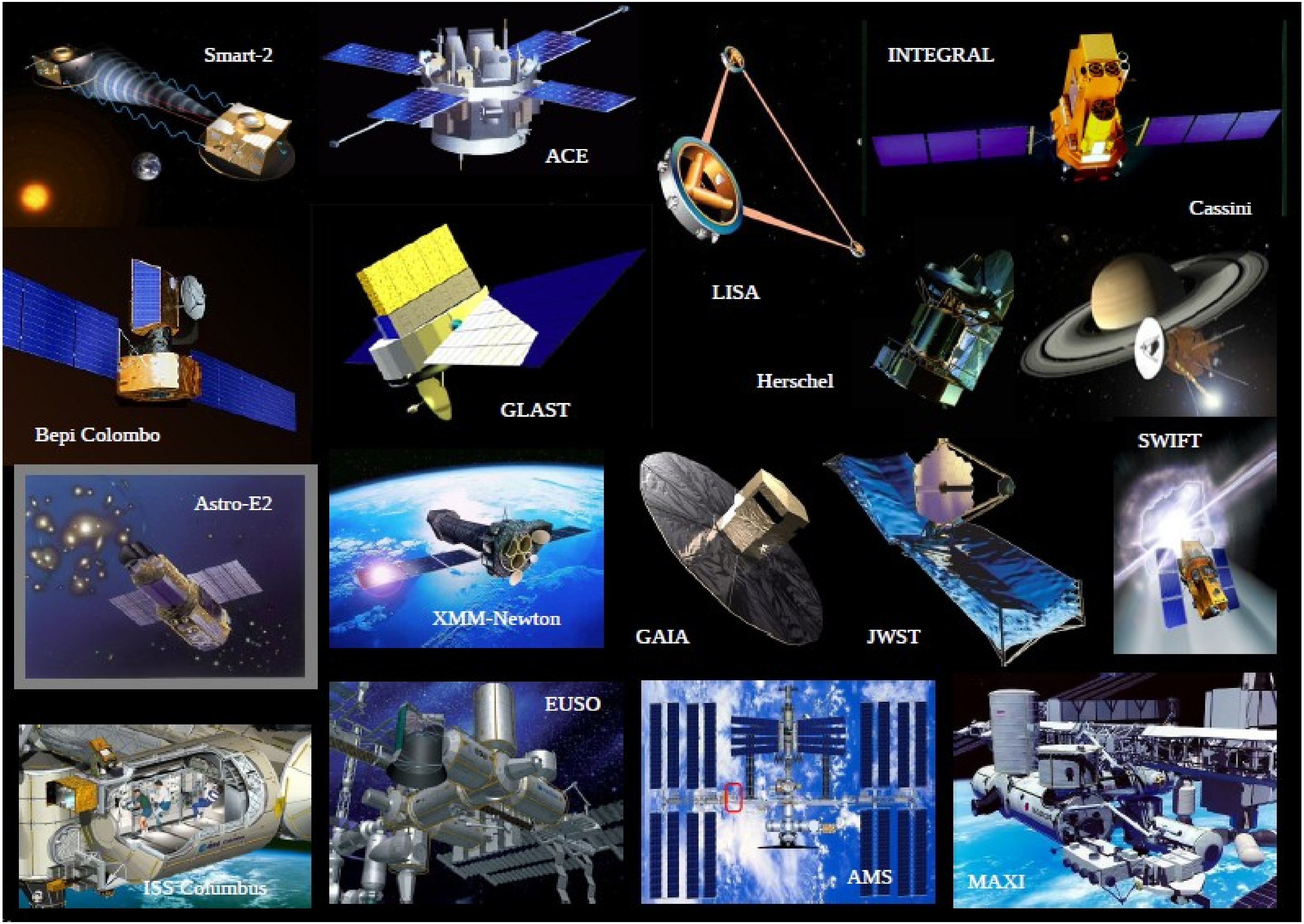




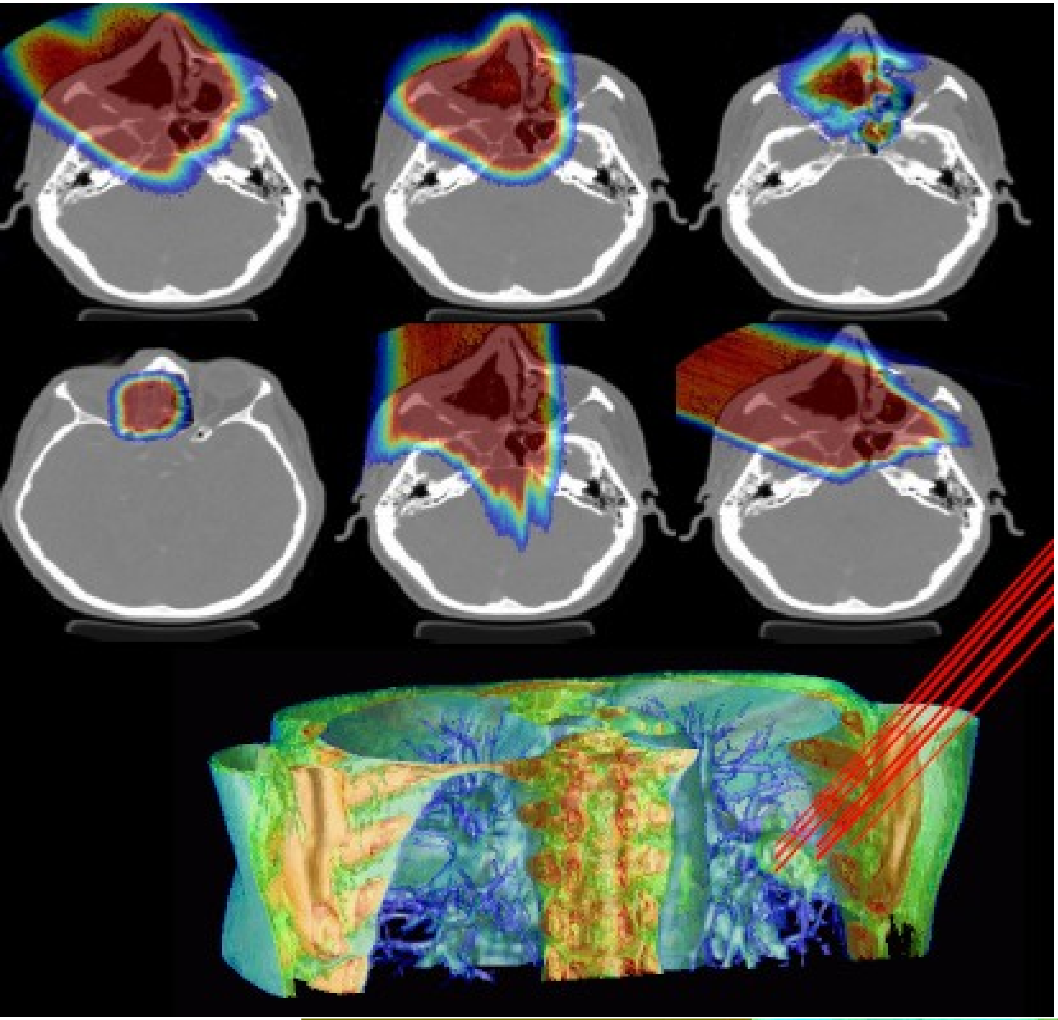


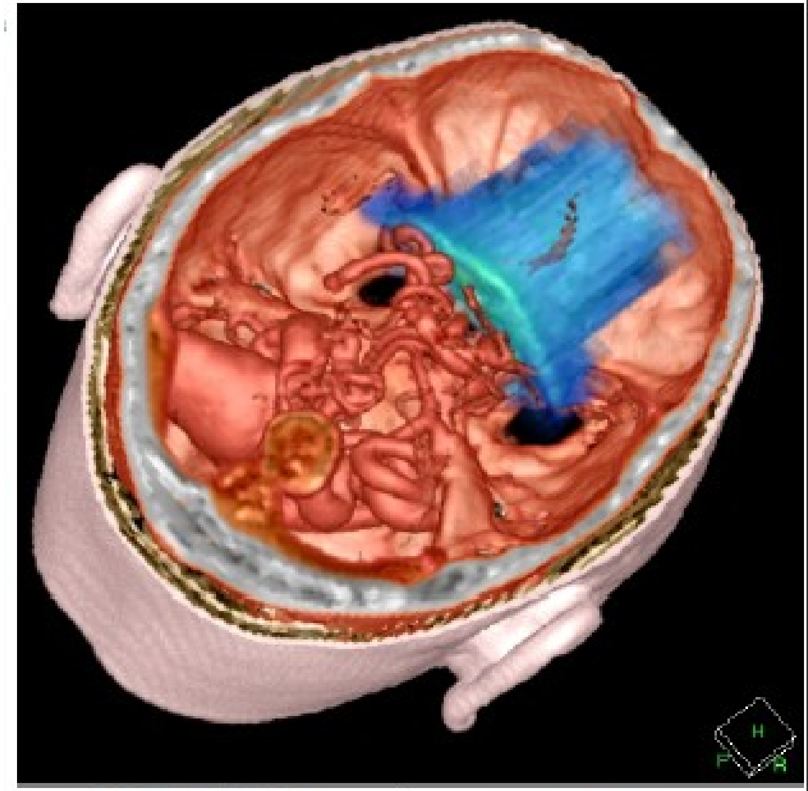


Space applications

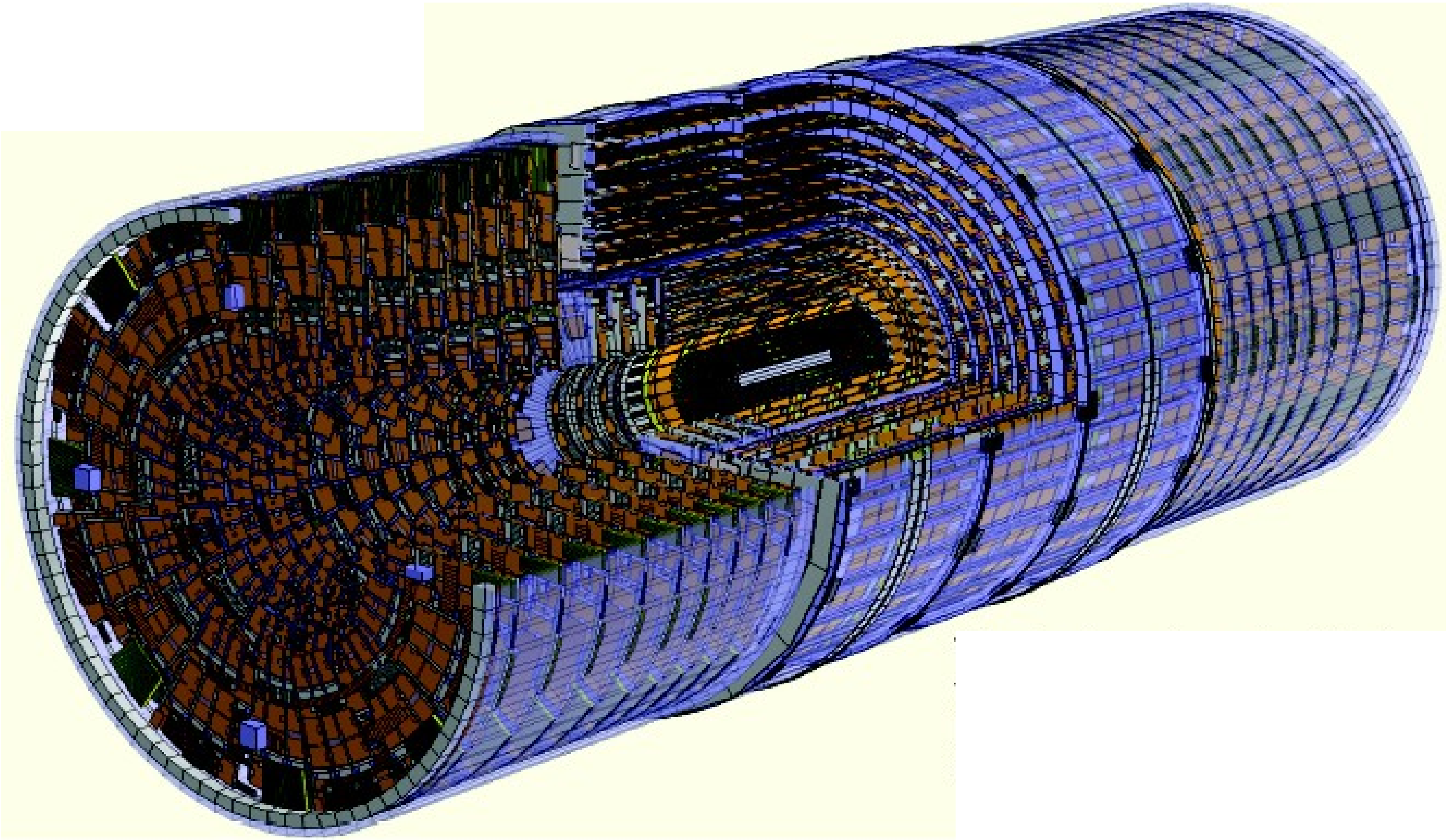


Medical applications

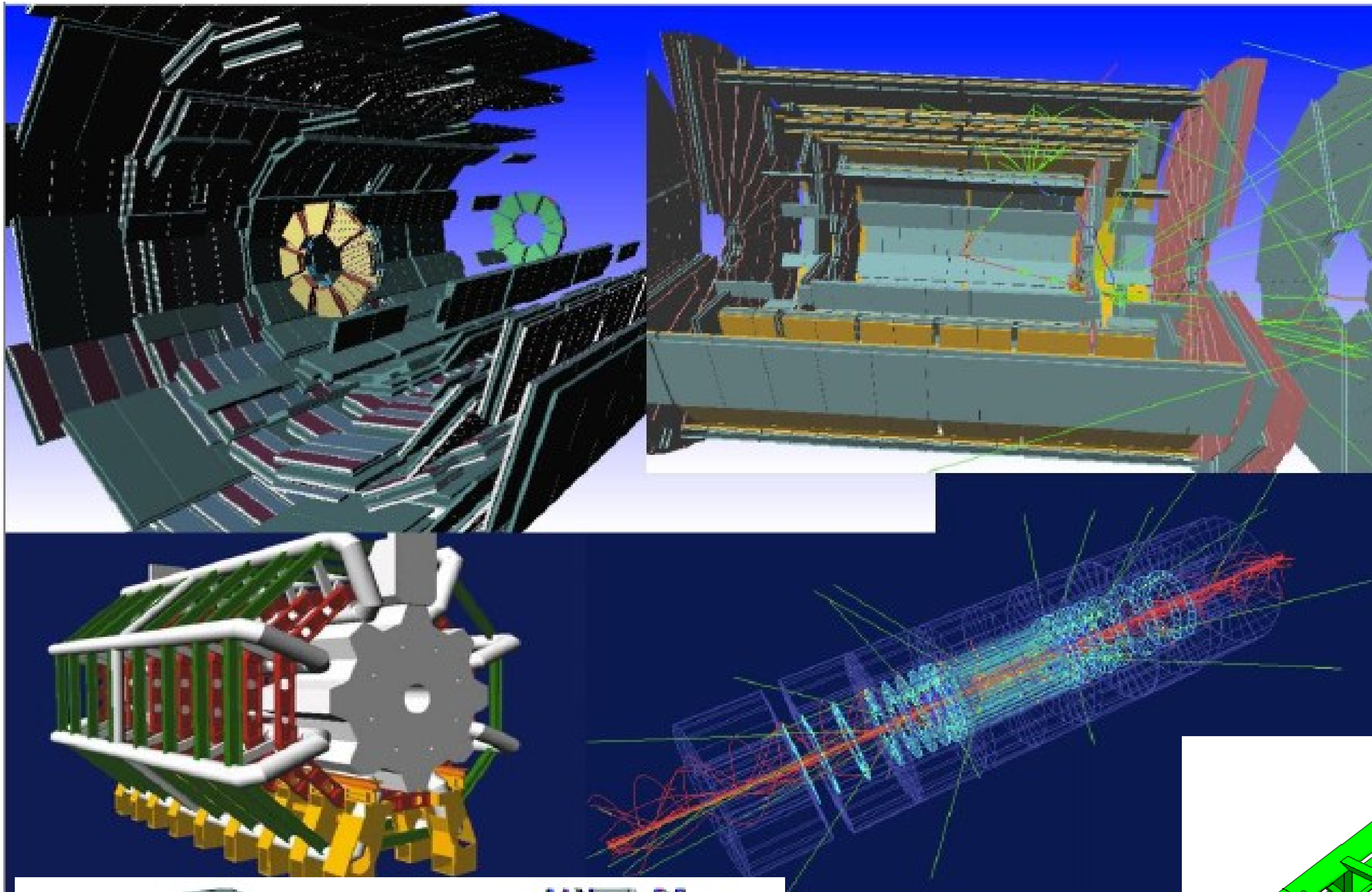


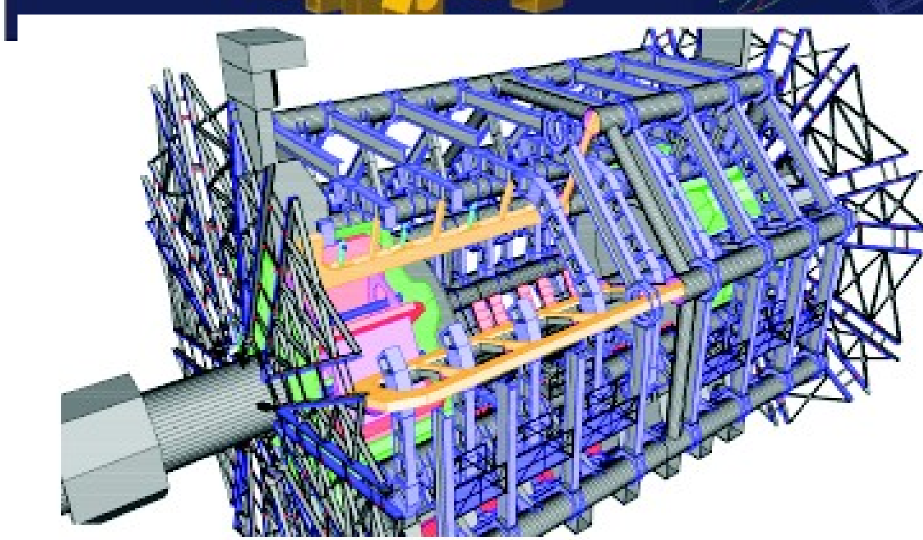


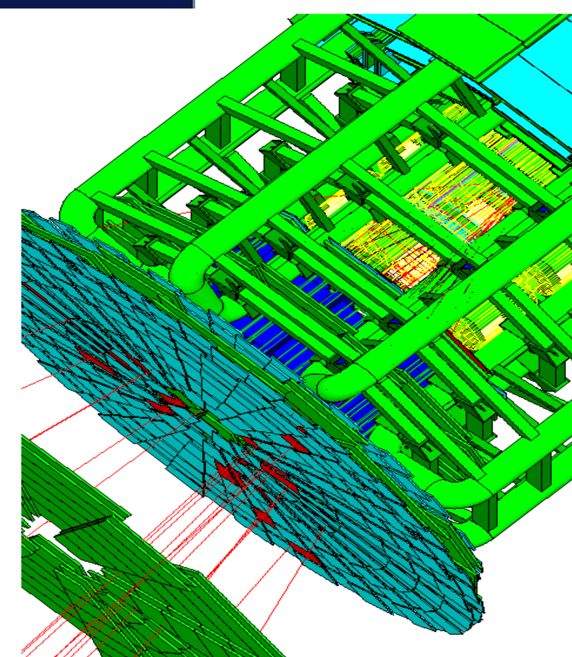
HEP : CMS tracker



HEP : ATLAS



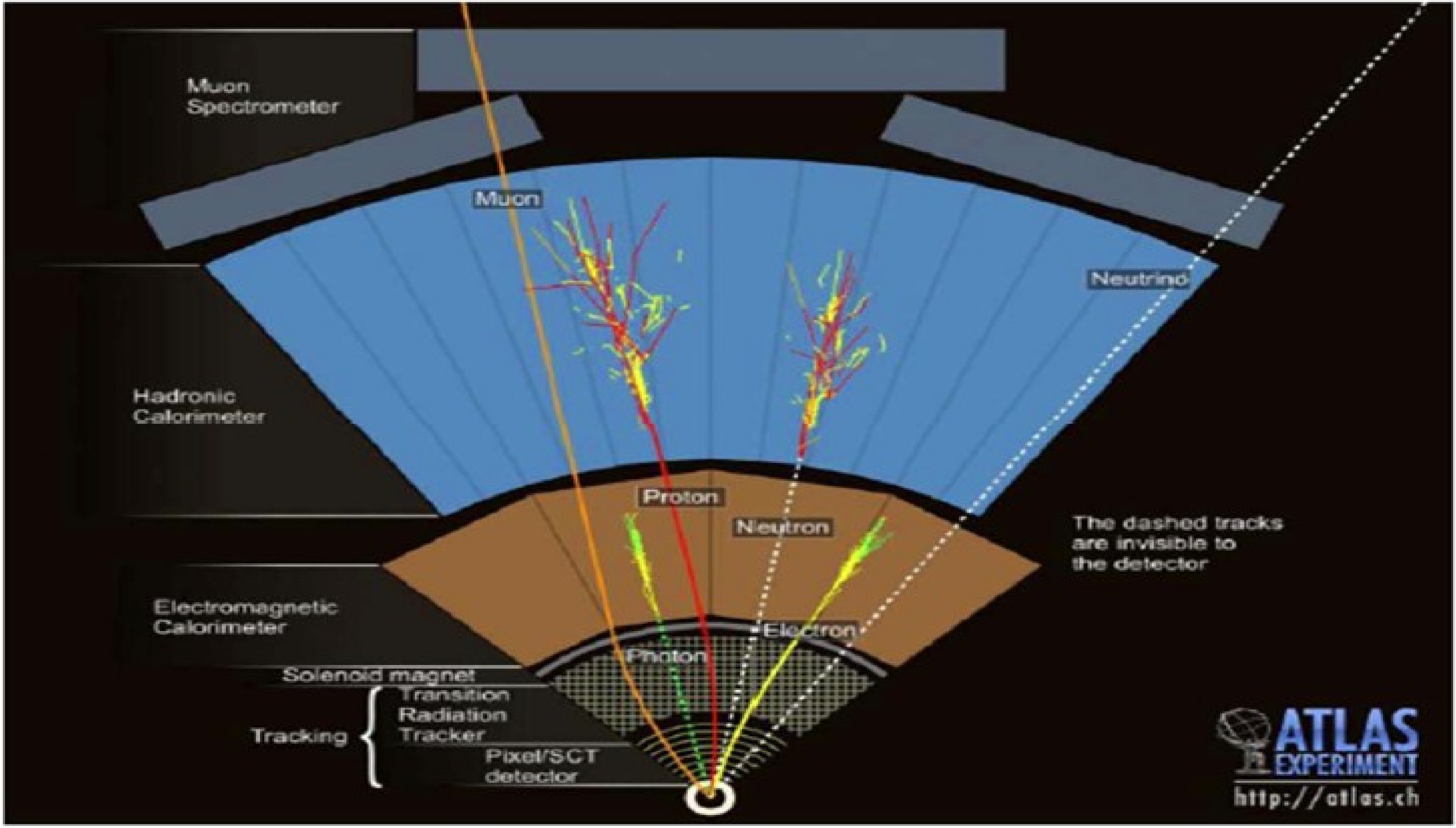




Physics

Particle interactions

Each particle type has its own set of physics processes.
Only electromagnetic effects are directly measurable



Main electromagnetic processes

Gamma

- Conversion
  γ -> e+ e- , μ+ μ-
- Compton scattering
  γ *(atomic)*e- -> γ *(free)*e-
- Photo-electric
  γ material -> *(free)*e-
- Rayleigh scattering
  γ atom -> γ atom

Muon

- Pair production
  μ- atom -> μ- e+ e-
- Bremsstrahlung
  μ- atom -> μ- γ
- MSC (Coulomb scattering)
  μ- atom -> μ- atom
- Ionization :
  μ- atom -> μ- ion+ e-

Total cross section:
⇨ step length

Differential & partial cross sections :
⇨ final state (multiplicity & spectra)

Electron, Positron

- Bremsstrahlung
  e- atom -> e- γ
- MSC (Coulomb scattering)
  e- atom -> e- atom
- Ionization
  e- atom -> e- ion+ e-
- Positron annihilation
  e+ e- -> γ γ

Charged hadron, ion

- Bremsstrahlung
  h- atom -> h- γ
- MSC (Coulomb scattering)
  h- atom -> h- atom
- Ionization
  h- atom -> h- ion+ e-

Production and tracking cuts

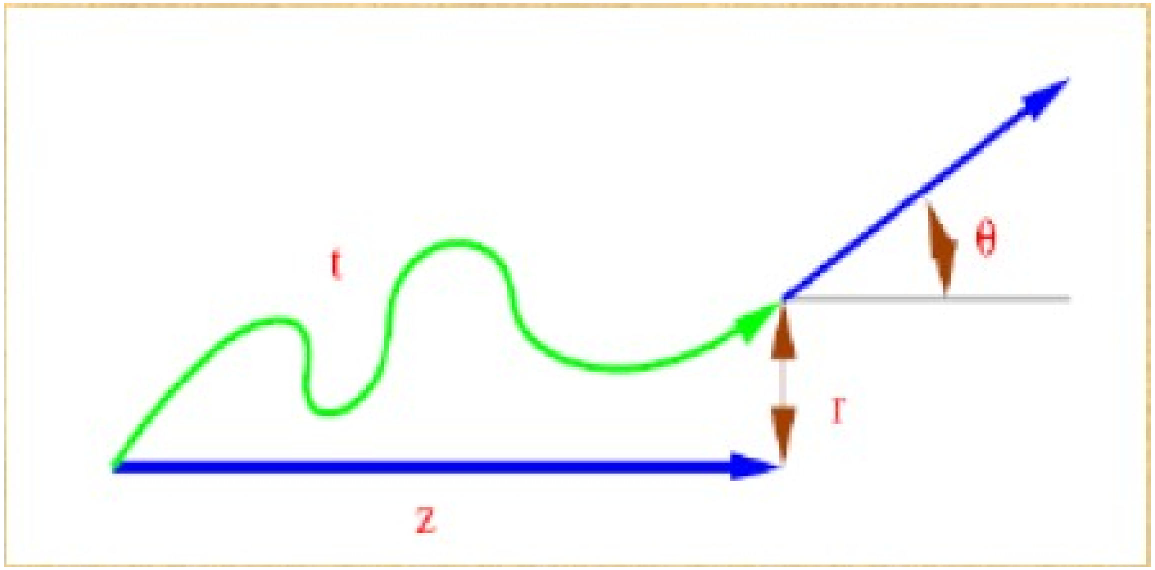
- Ionization & bremsstrahlung processes produce an increasing number of secondaries as the secondary energy decreases, so we need to set a production cut
  - Above the cut, new particles (e-, γ) are created
  - Below the cut, “continuous” energy loss of the primary

- Once a charged particle is created, it can be reliably transported down to E_{kin} ~ 1 keV
  - Either, stop it below a tracking cut and deposit its energy locally
  - Or, go down to E_{kin} -> 0 using its approximated range

- Production and tracking cuts can be expressed directly as kinetic energy thresholds or indirectly as equivalent range thresholds

Multiple (Coulomb) scattering (MSC)

- Charged particles traversing a finite thickness of matter suffer a huge number (millions) of elastic Coulomb scatterings
- The cumulative effect of these small angle scatterings is mainly a net deflection from the original particle direction
- In most cases, to save CPU time, these multiple scatterings are not simulated individually, but in a "condensed" form
- Various algorithms exist, and new ones under development. One of the main differences between codes



Electromagnetic physics

- Typical validity of electromagnetic physics **≥ 1 keV** ;
  for a few processes, extensions to lower energies
- CPU performance of electromagnetic physics is critical : continuous effort to improve it
- Detailed validation of electromagnetic physics is necessary before the validation of hadronic physics
- Typical precision in electromagnetic physics is **~1%**
  - QED is extremely precise for elementary processes,
    but atomic and medium effects, important for detector simulations, bring larger uncertainties...
  - Moreover, the “condensed” description of multiple scattering introduces further approximations...
  - Continuous effort to improve the models

Hadronic physics

- Hadrons ($\pi\pm$, $K\pm$, K°_{L} , p, n, α, *etc.*), produced in jets and decays, traverse the detectors (H,C,Ar,Si,Al,Fe,Cu,W,Pb...)
- Therefore we need to model hadronic interactions

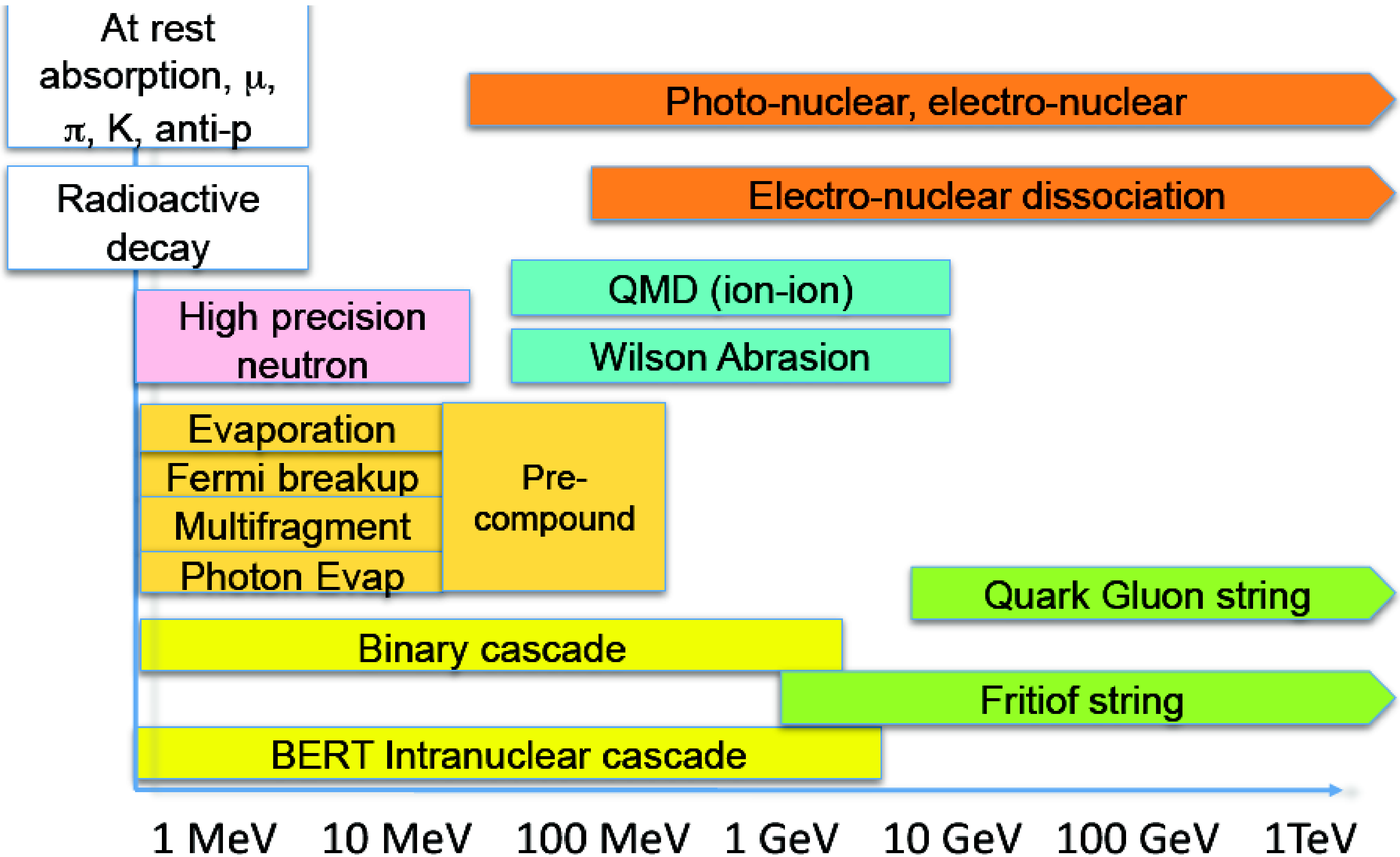
  hadron – nucleus -> *anything*

  in our detector simulations
- In principle, QCD is the theory that describes all hadronic interactions; in practice, perturbative calculations are applicable only in a tiny (but important!) phase-space region
  - the hard scattering at high transverse momentum

  whereas for the rest, i.e. most of the phase space
  - soft scattering, re-scattering, hadronization, nucleus de-excitation

  only approximate models are available
- Hadronic models are valid for limited combinations of
  - **particle type - energy - target material**

Hadronic models available in Geant4

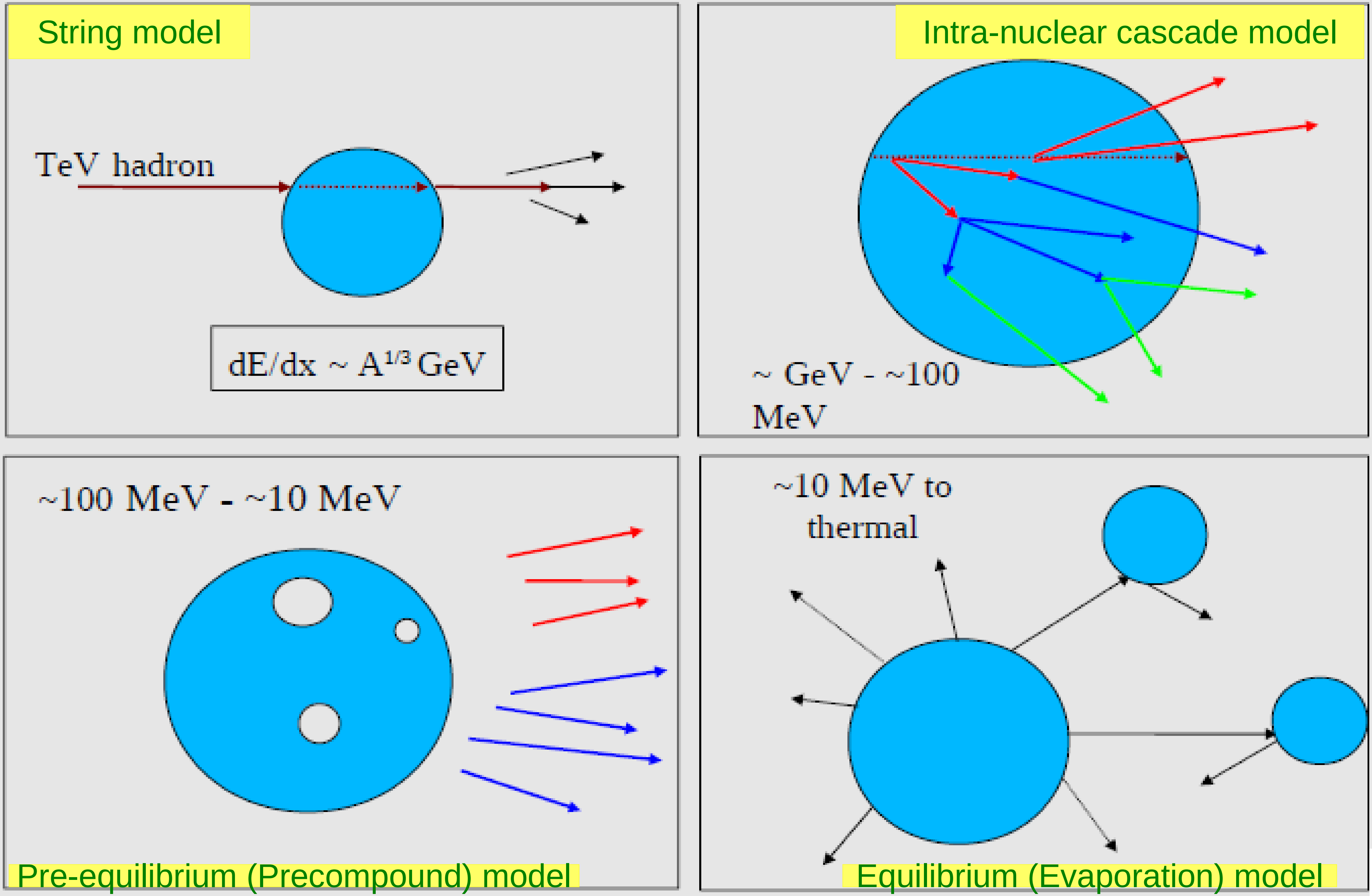


Physics configuration

- No “unified” hadronic model: need to choose a set of hadronic models able to cover all possible interactions
  - The choice depends on the use-case, because of:
    - The energy scale involved
    - The compromise between accuracy and CPU speed
  - The choice is often done by the developers
    - Options can be proposed according to use-cases

- In the case of Geant4
  - These physics configurations are called “physics lists”
    - The particles to be considered in the simulation are also specified
  - There is no default
  - Ready-to-use “physics lists” exist, for different use-cases
  - Users can also tailor/modify any of these, or write their own

Hadronic Interactions from TeV to meV

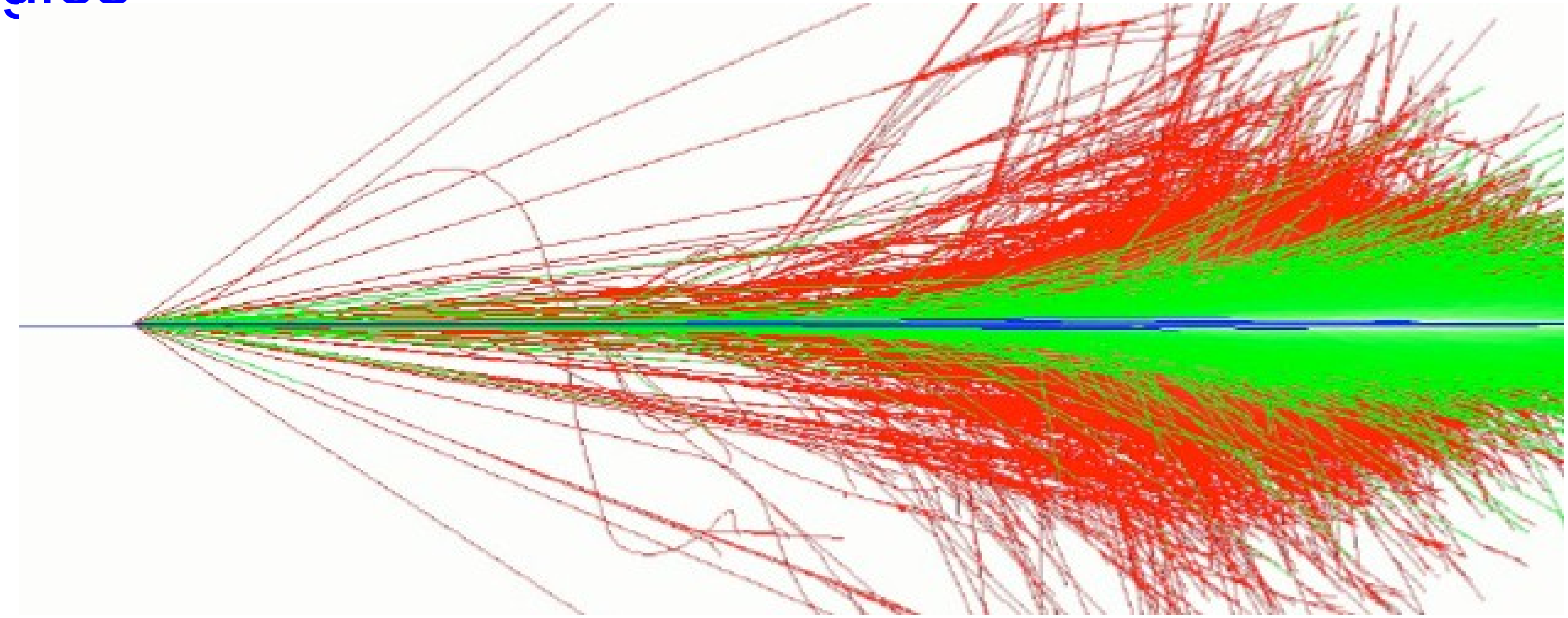


An interesting complication: Neutrons

- Neutrons are abundantly produced
  - Mostly “soft” neutrons, produced by the de-excitations of nuclei, after hadron-nucleus interactions
  - It is typically the 3rd most produced particle (after e-, γ)
- Before a neutron “disappears” via an inelastic interaction, it can have many elastic scatterings with nuclei, and eventually it can “thermalize” in the environment
- The CPU time of the detector simulation can vary by an order of magnitude according to the physical accuracy of the neutron transportation simulation
  - For typical high-energy applications, a simple treatment is enough (luckily!)
  - For activation and radiation damage studies, a more precise, data-driven and isotope-specific treatment is needed, especially for neutrons of kinetic energy below **~ MeV**

Hadronic showers

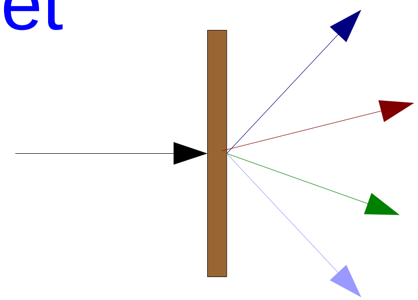
- A single hadron impinging on a large block of matter (e.g. a calorimeter) produces secondary hadrons of lower energies, which in turn can produce other hadrons, and so on: the set of these particles is called a **hadronic shower**
  - e-/e+/γ (electromagnetic component) are also produced copiously because of **$\pi^0 \to \gamma\gamma$** and ionization of charged particles
- The development of a hadronic shower involves many energy scales, from hundreds of GeV down to thermal energies



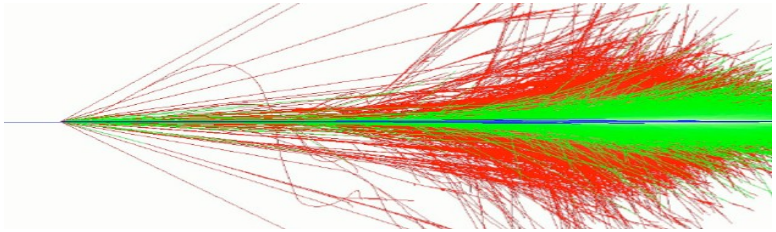
Validation

Validation & tuning of hadronic models

- The developers of the hadronic models are responsible of the tuning & validation of these models with thin-target (microscopic, single-interaction) measurements



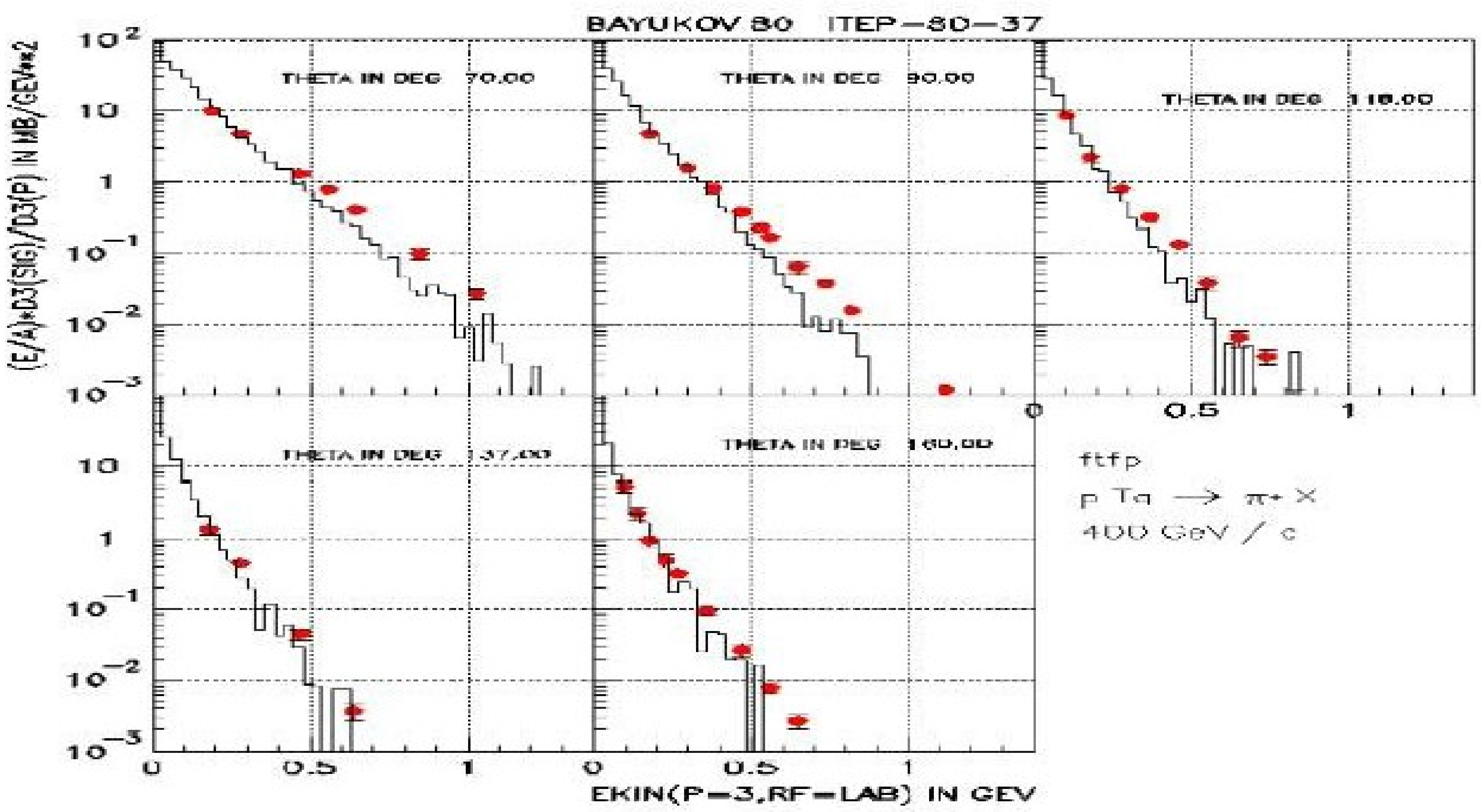
- Validation of complete physics configurations is performed by users mostly via measurements of hadronic showers in calorimeter test-beam setups (thick targets)



- The most important application of the hadronic models for collider experiments is the **simulation of jets**, which involves
    1. the Monte Carlo event generator
    2. the convolution of the showers for each constituent hadron
    3. experiment specific: geometry & materials, digitization, etc.

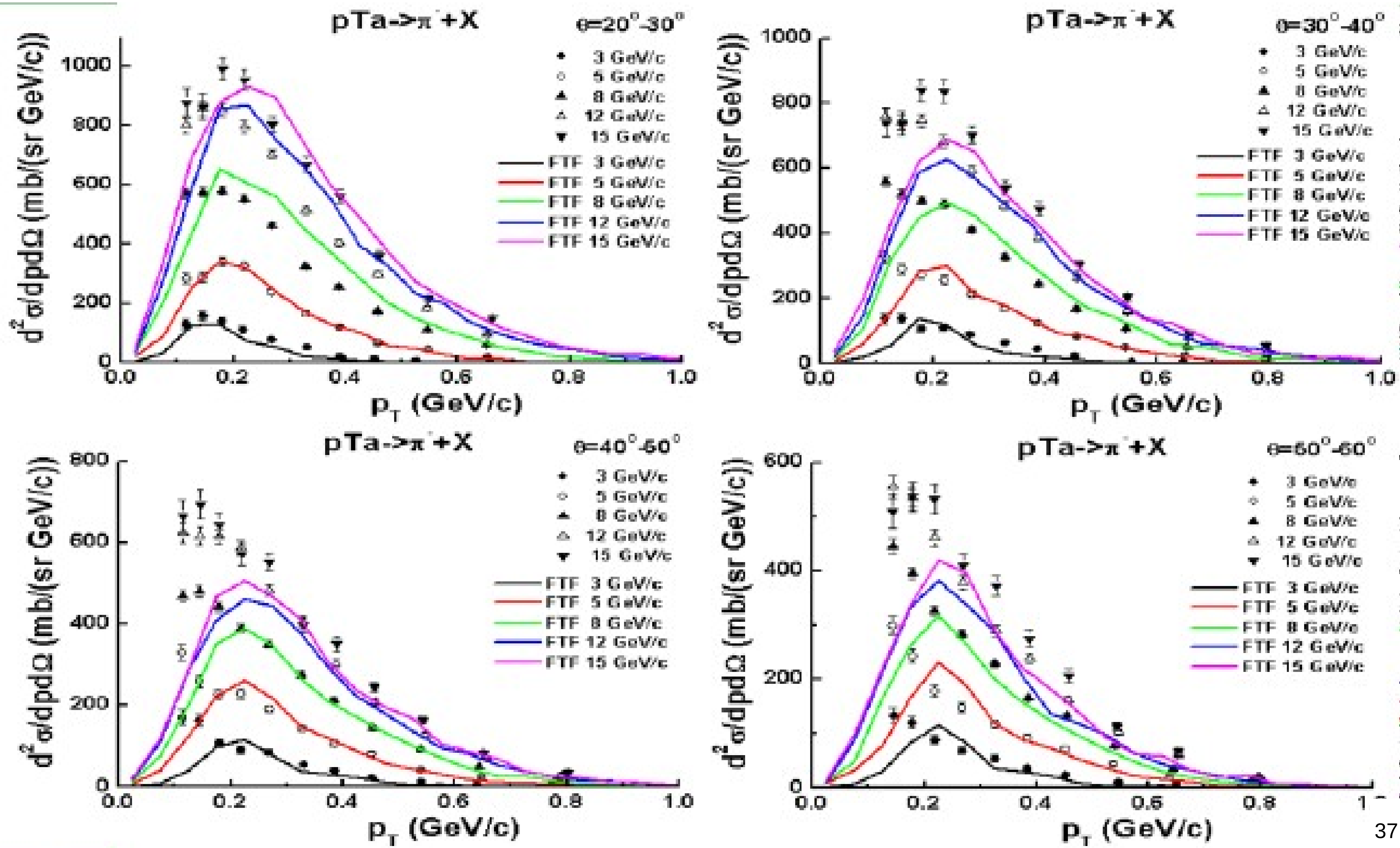
Model-level thin-target tests

FTF Results at 400 GeV/c
p Ta -> pi+ X



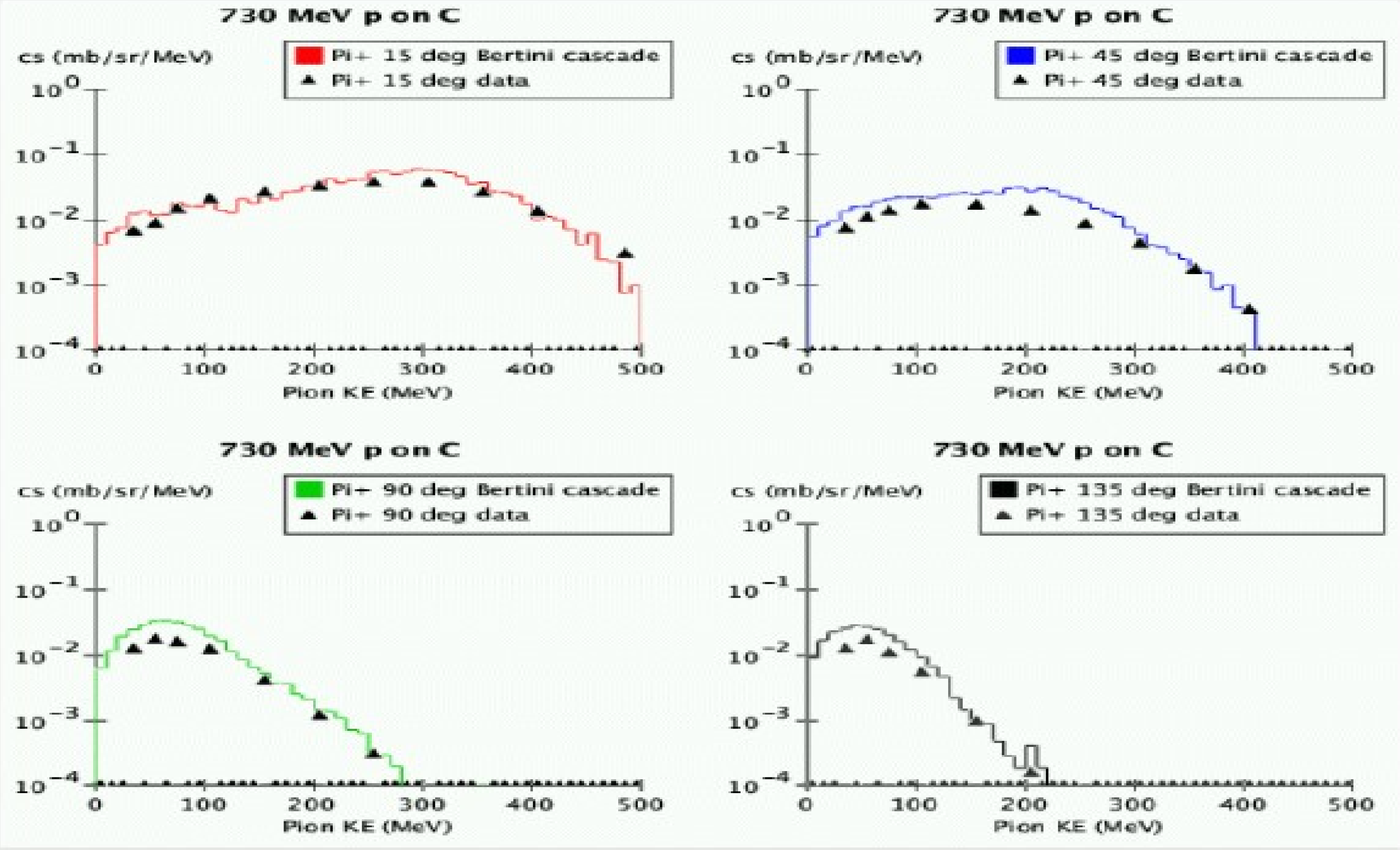
Model-level thin-target tests

FTF validation, HARP-CDP data



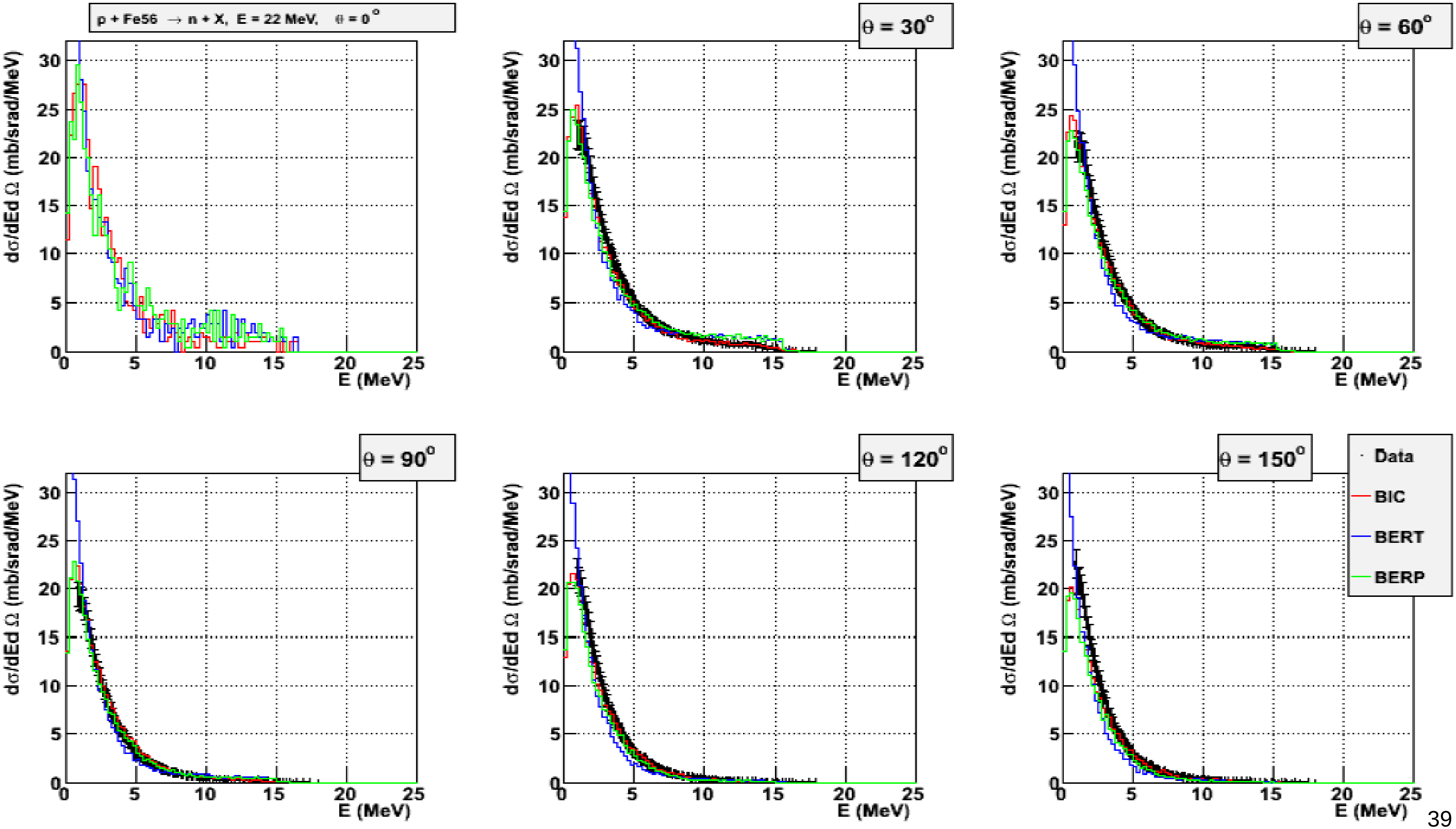
Model-level thin-target tests

Validation of the Bertini Cascade

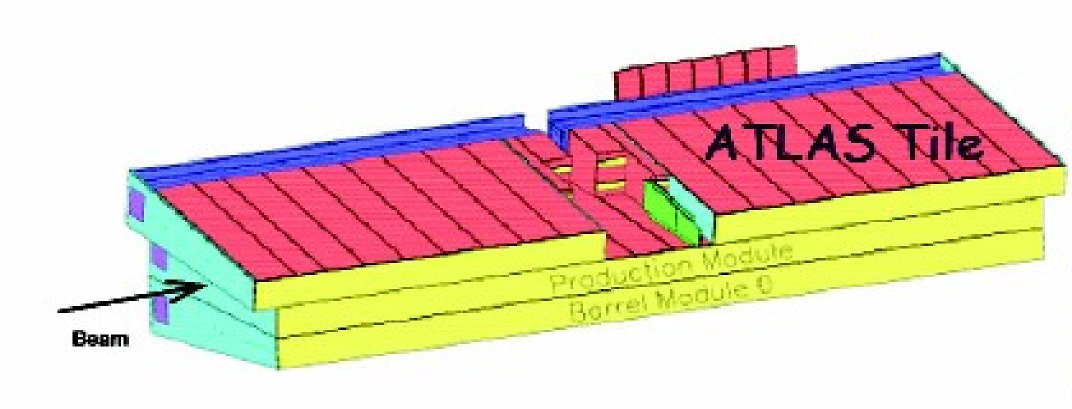


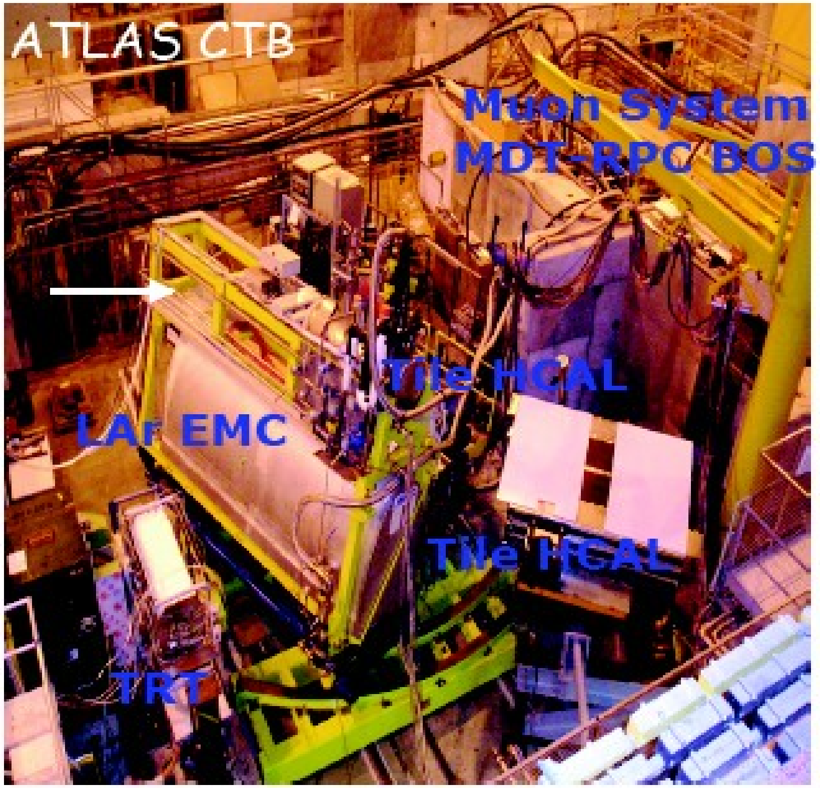
Model-level thin-target tests

Preco validation, 22 MeV p – Fe -> n



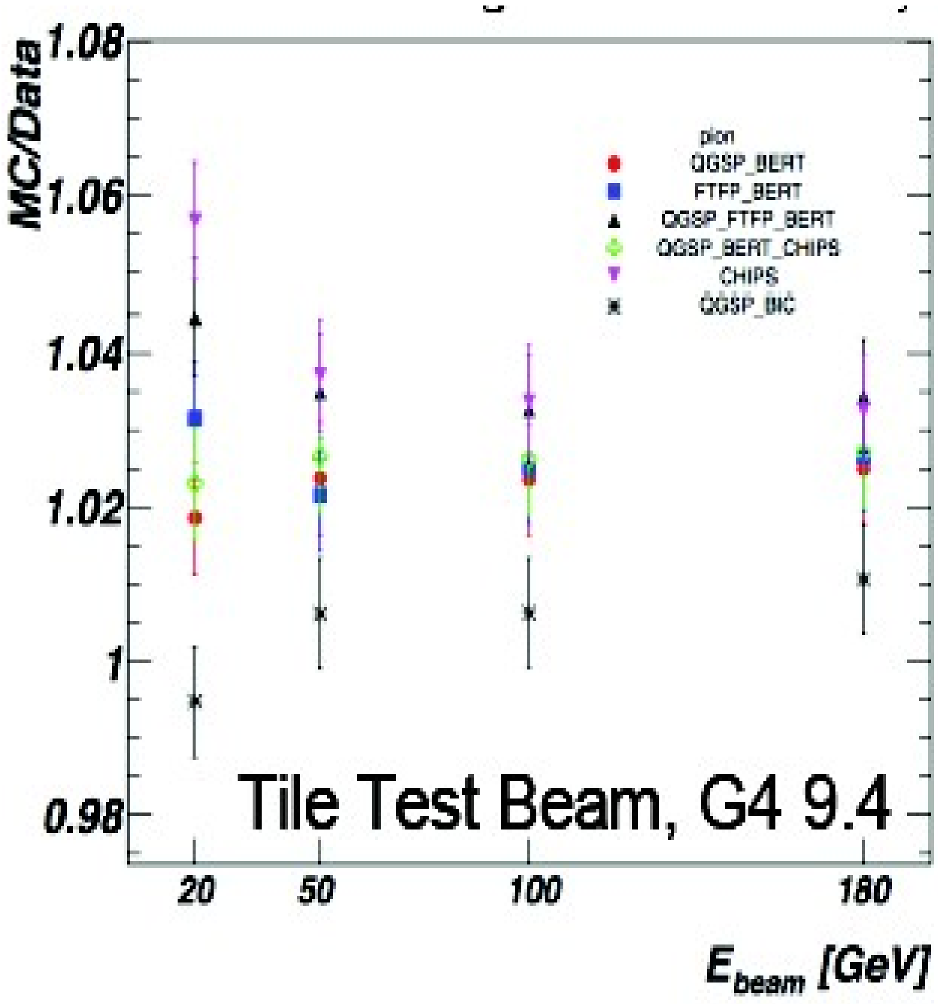
LHC calorimeter test-beams

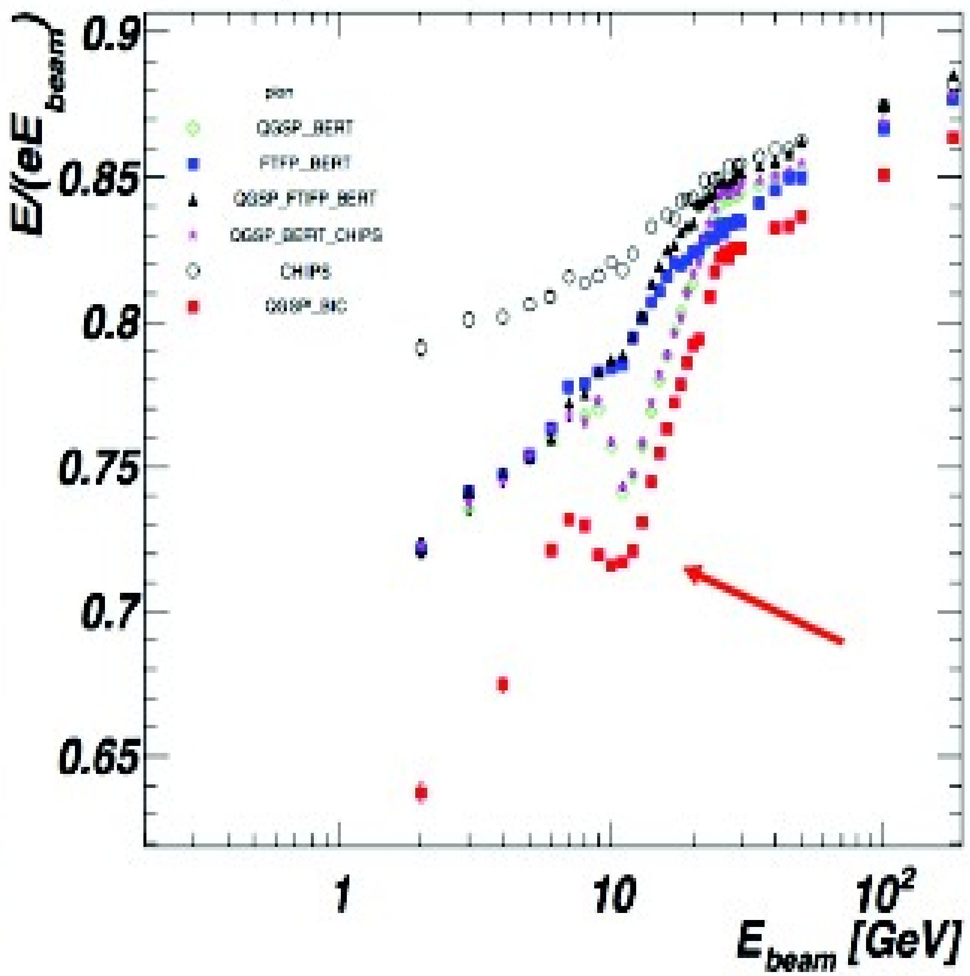




Energy response

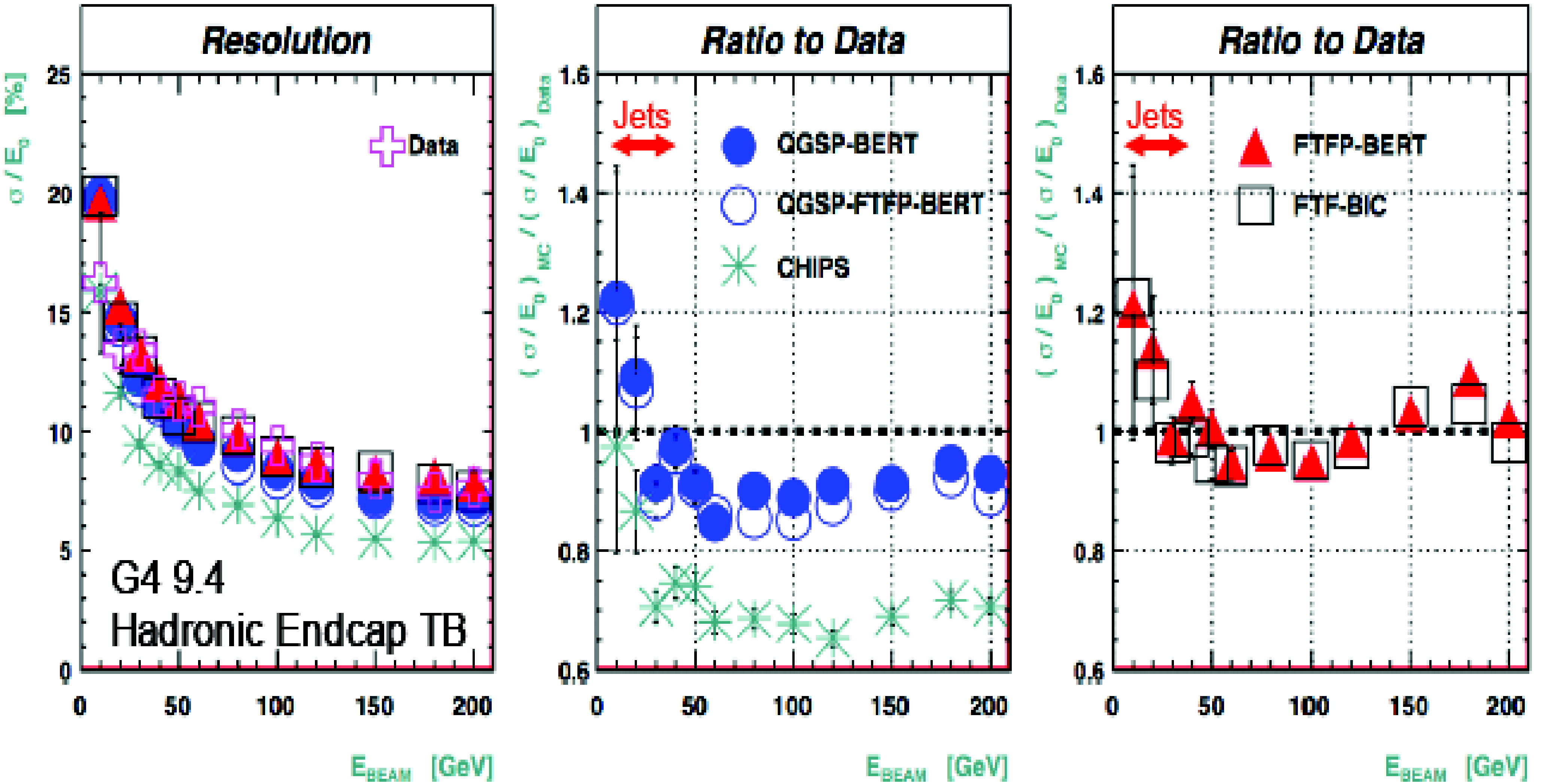
ATLAS TileCal test-beam





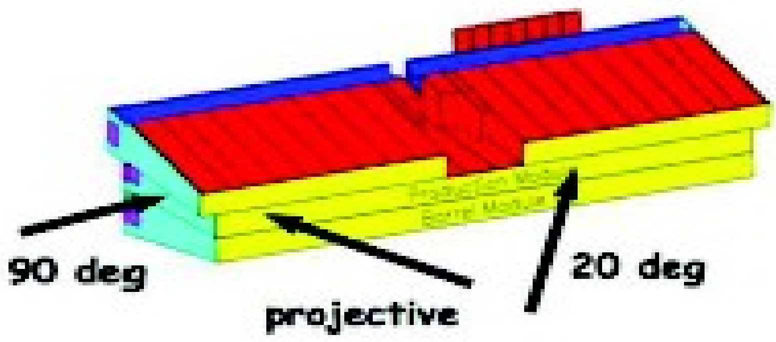
Energy resolution

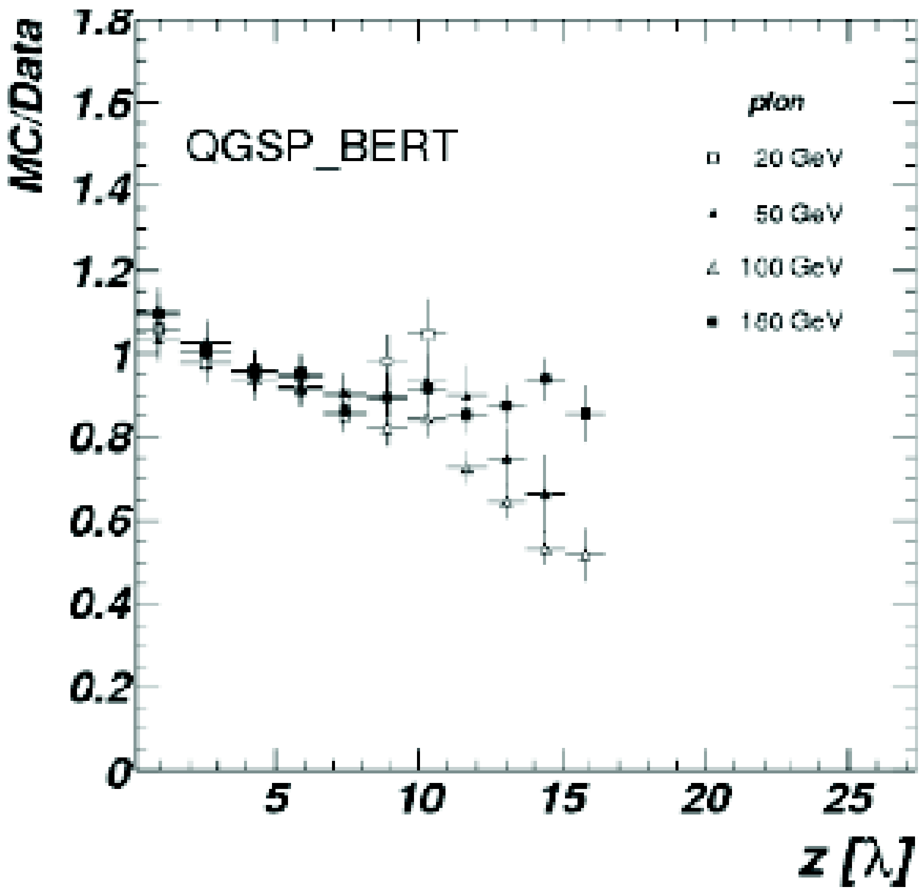
ATLAS HEC test-beam

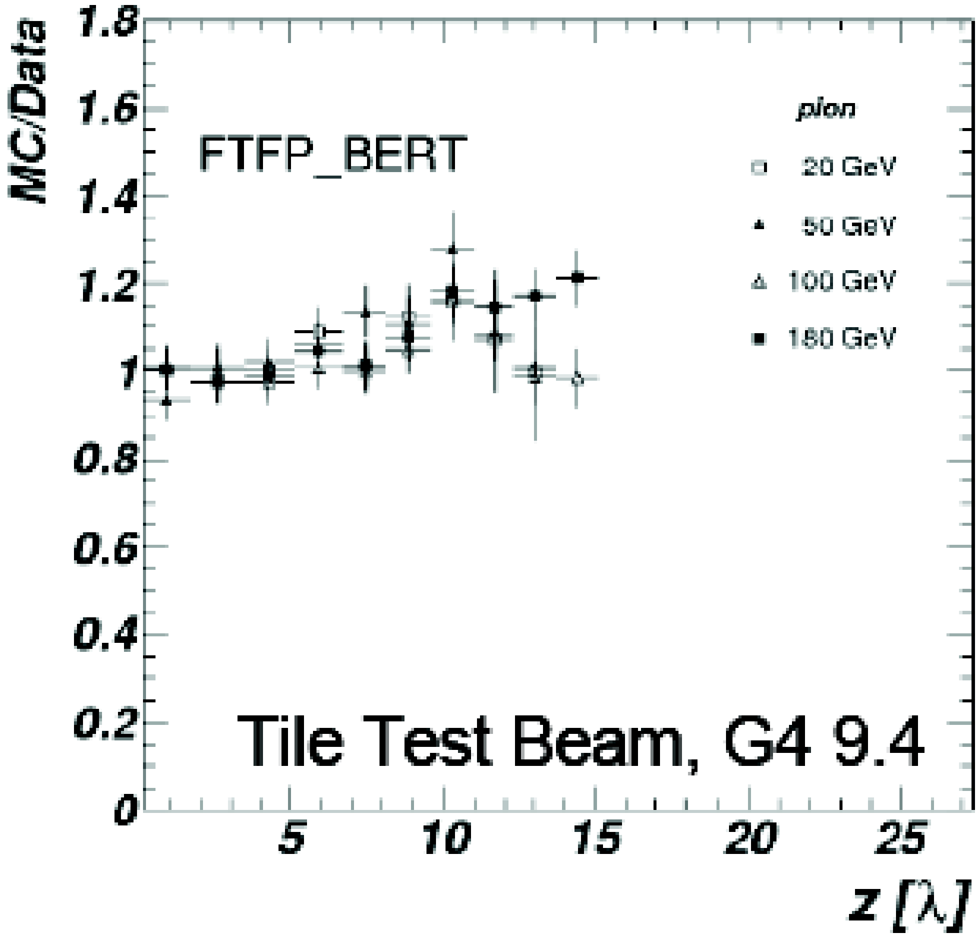


Longitudinal shower shape

ATLAS TileCal test-beam @90°

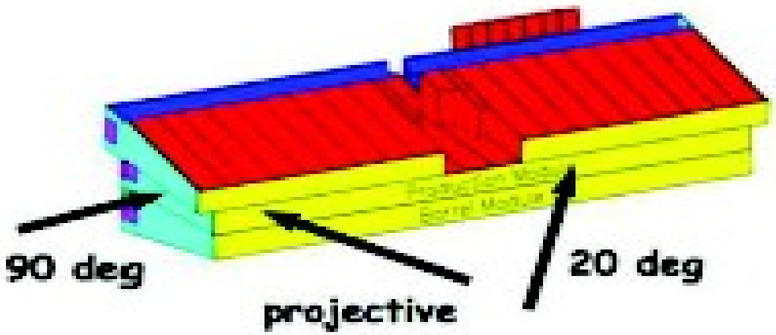


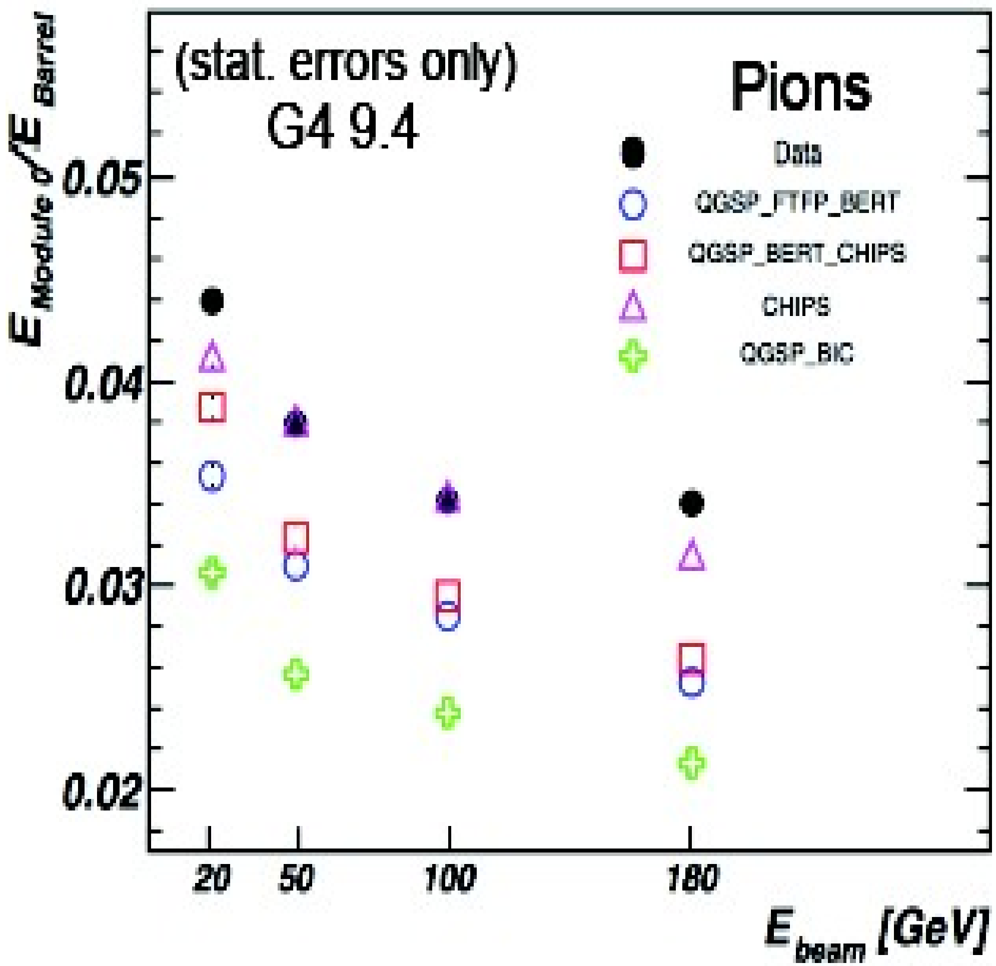


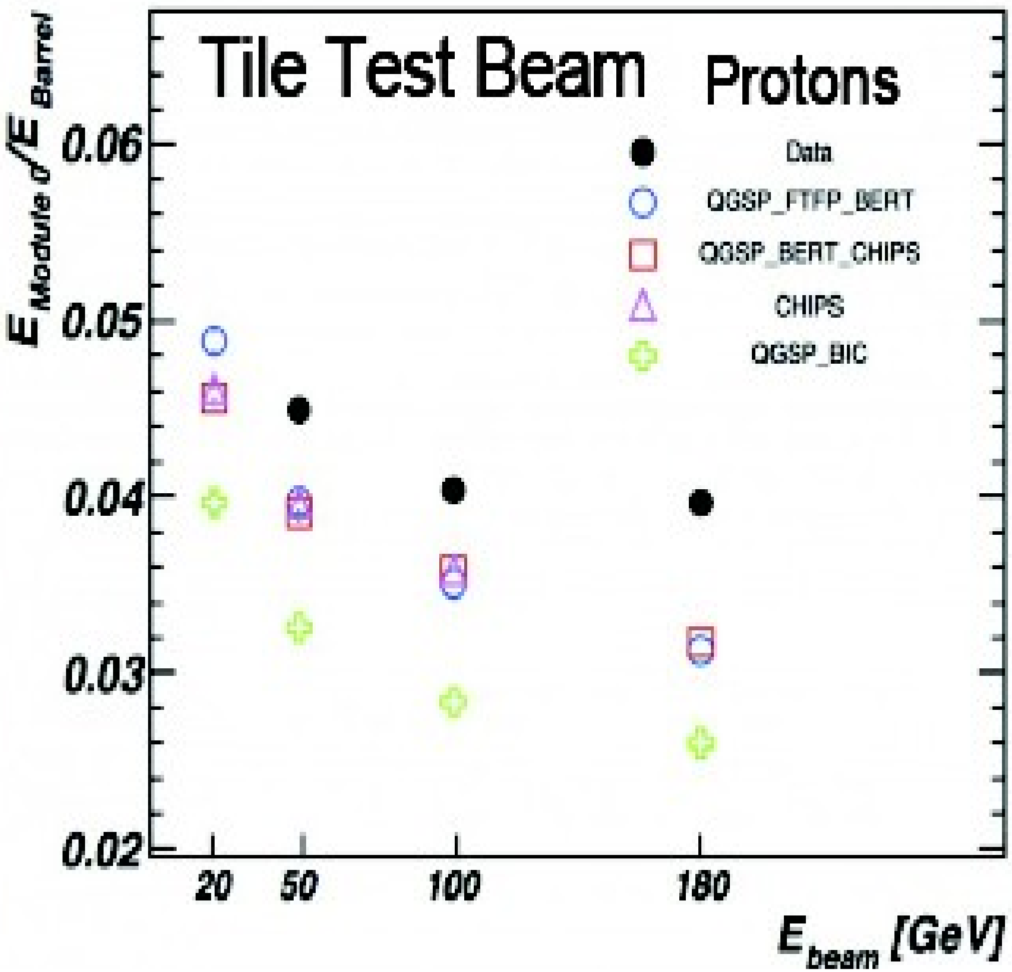


Lateral shower shape

ATLAS TileCal test-beam @90°

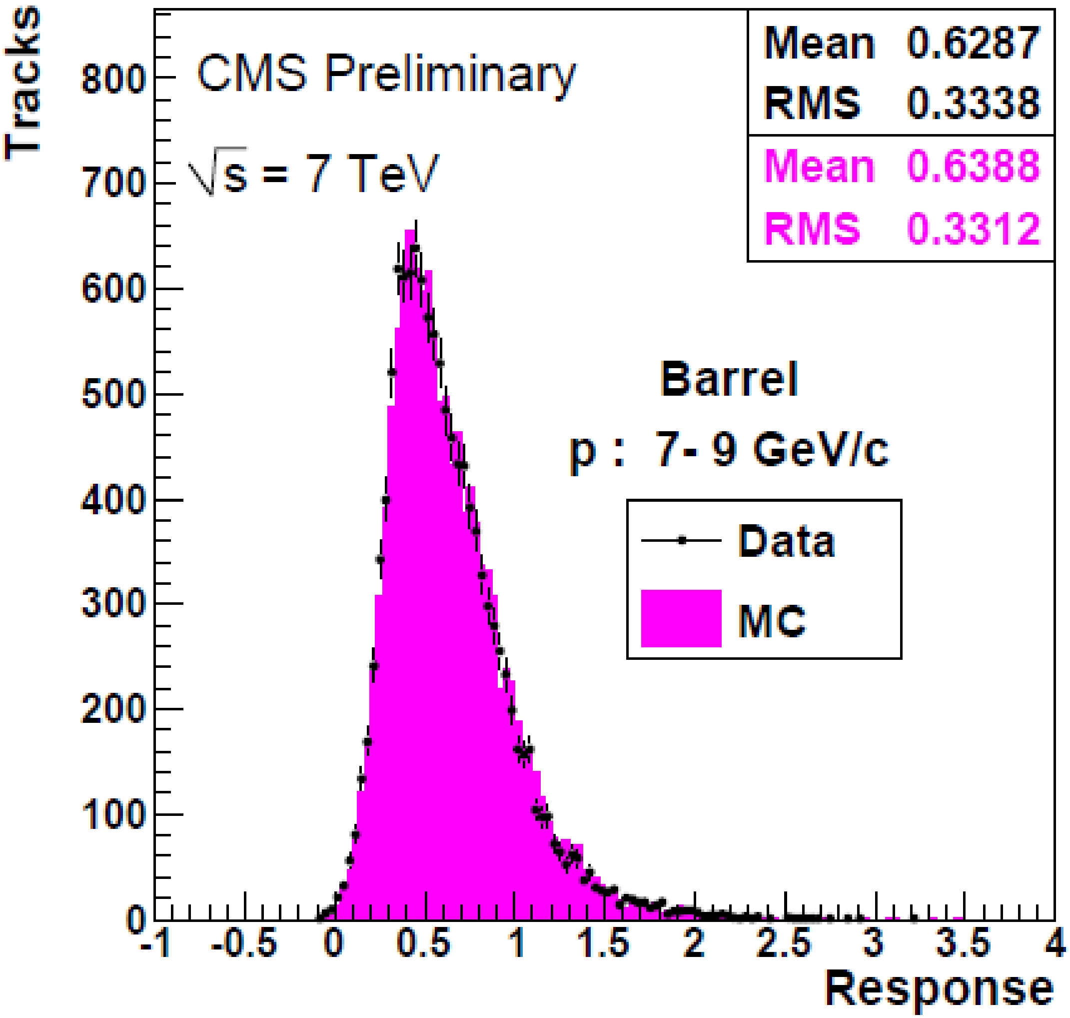


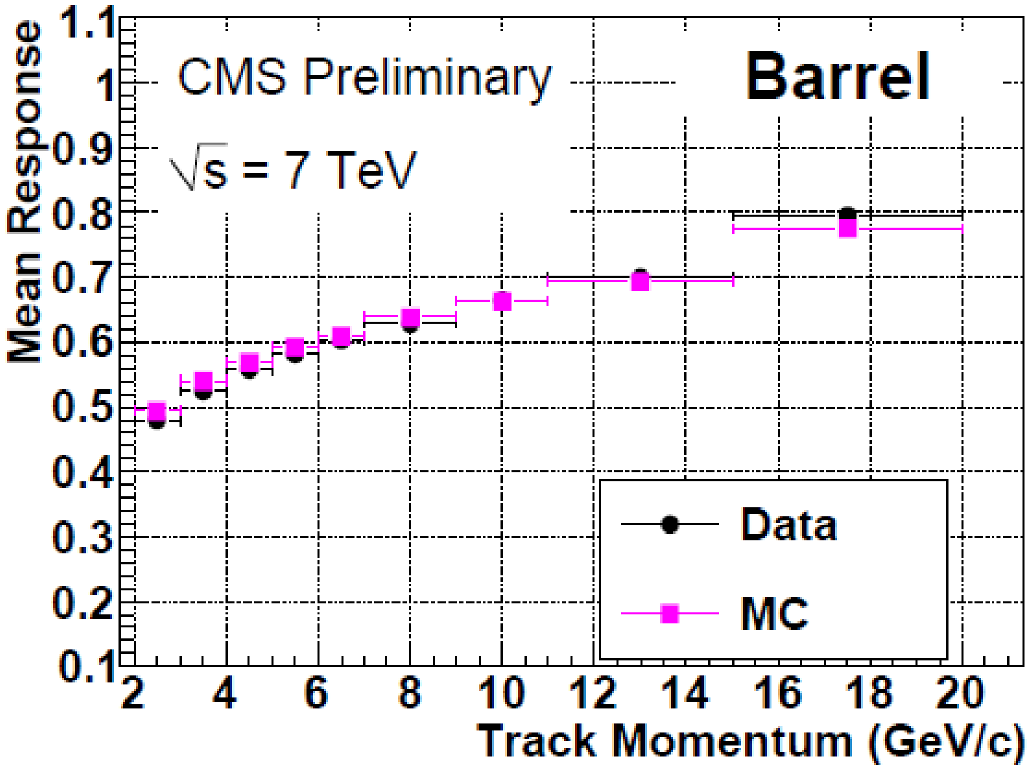




Isolated single hadron response: simulation *vs.* CMS p-p data

Agreement is better than ±3% between 2-20 GeV/c





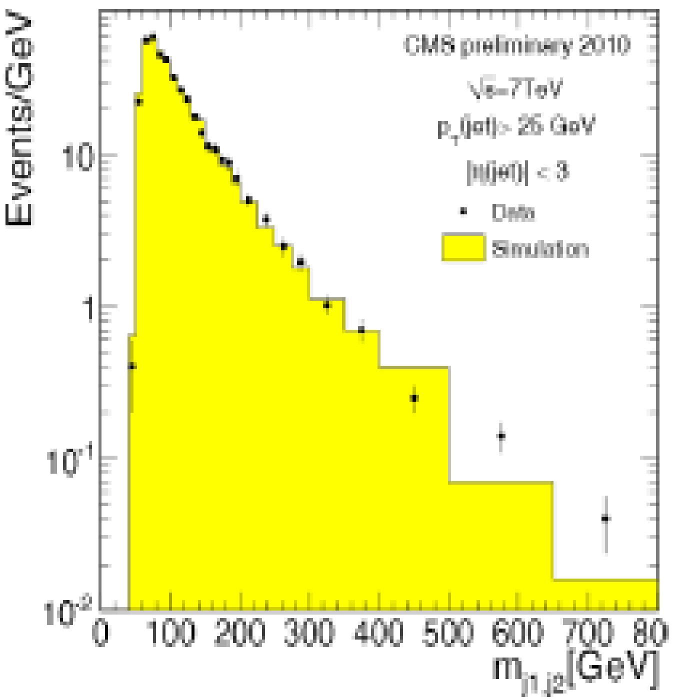
Di-jet invariant mass: simulation *vs.* CMS p-p data

Very good agreement between simulation and collision data!

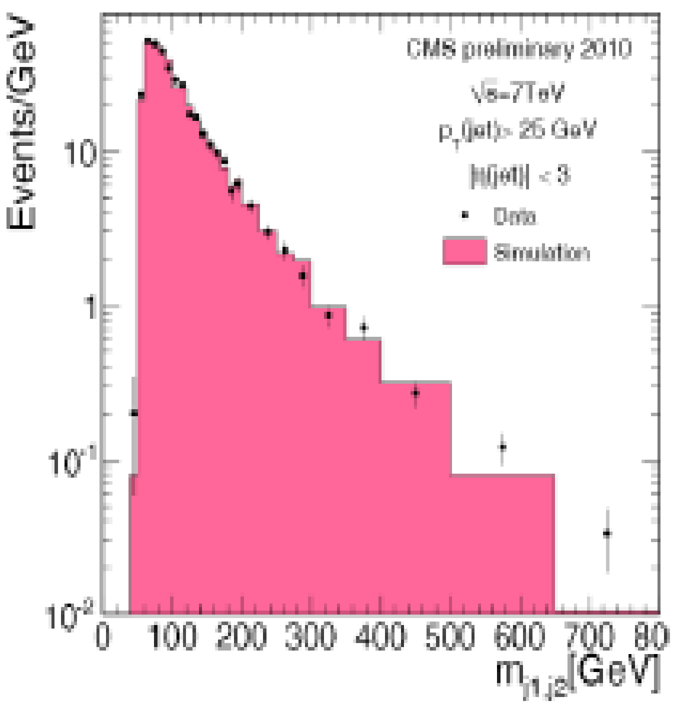
Three ingredients are convoluted in the simulation:

- Monte Carlo event generator: Pythia
- Detector simulation engine: Geant4
- Experiment-specific aspects: geometry/materials, digitization, calibration, rec.

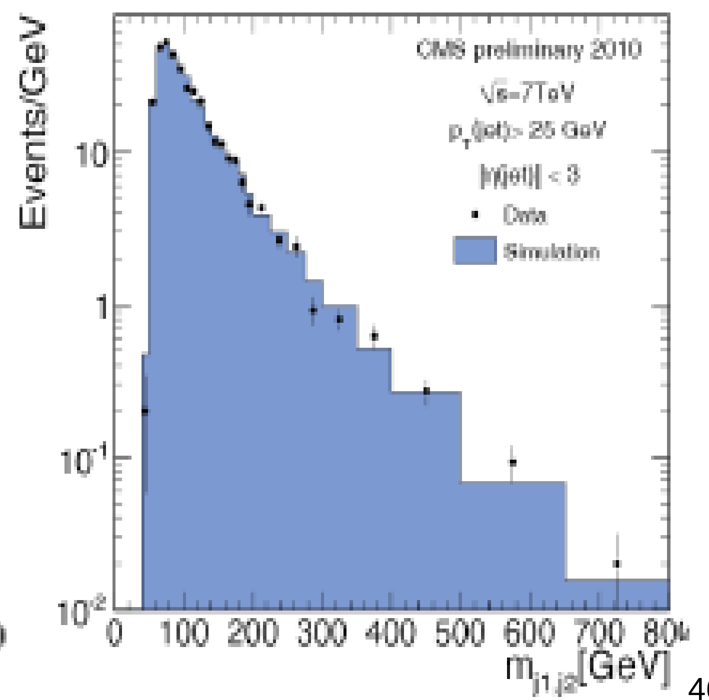
Calo jets



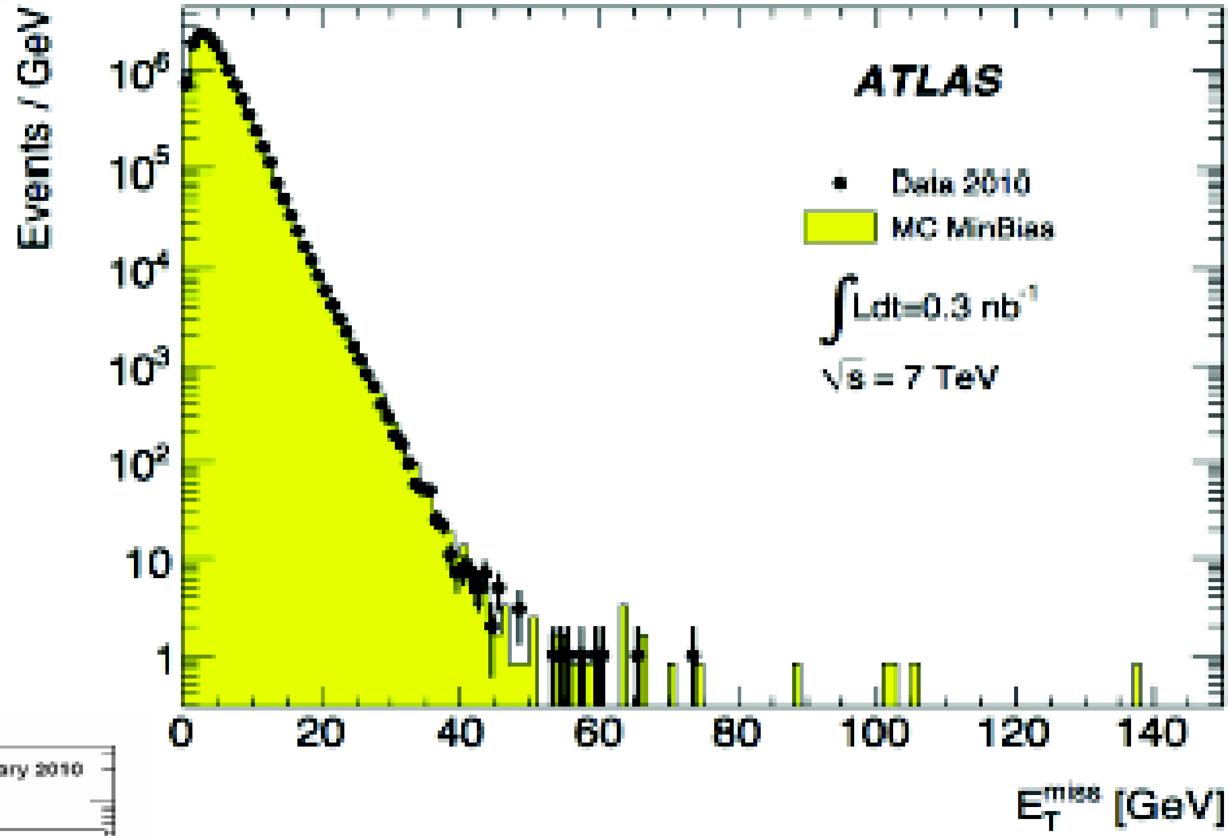
JPT jets

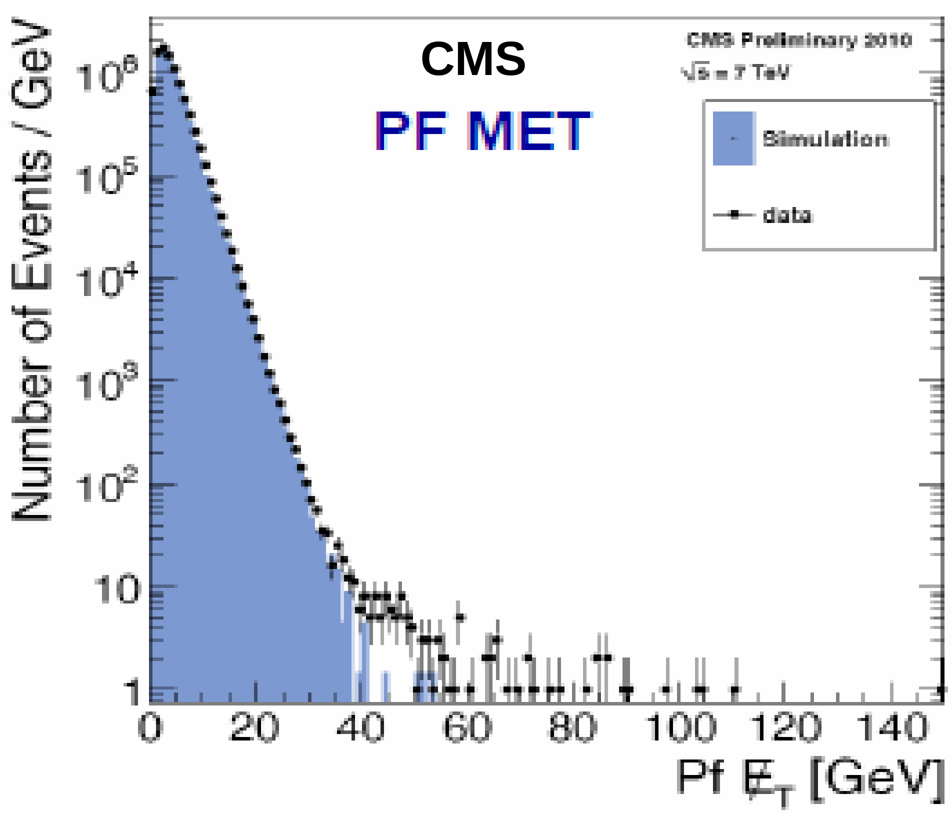


PF jets



Missing E_T : simulation *vs.* collision data





Missing ET is a very complex (global) variable

Good agreement over 6 orders of magnitudes!

Summary

- Detector simulation is one of the main tools of modern high-energy physics
- The main challenges of detector simulation are:
  - Physics accuracy
  - CPU performance
  - Validation

- *Suggestions for you:*
  - *Learn by studying and playing with existing examples*
  - *Be critical and pragmatic when using simulations*
  - *Contribute to the validation and provide feedback*

Other codes

- General
  - **Fluka**
  - Geant3
  - MARS
  - **MCNP / MCNPX**

- Dedicated to electromagnetic physics
  - EGS4
  - EGS5
  - EGSnrc
  - ETRAN
  - Penelope

Acknowledgment

Thanks to:

- Katsuya Amako
- Makoto Asai
- Andrea Dotti
- Michel Maire
- Dennis Wright

for some of the figures used in this lecture