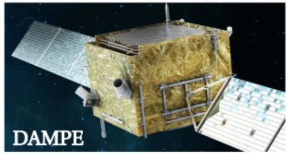
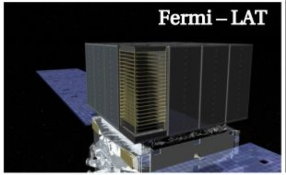
Hands-on session

CORSIKA & Geant4

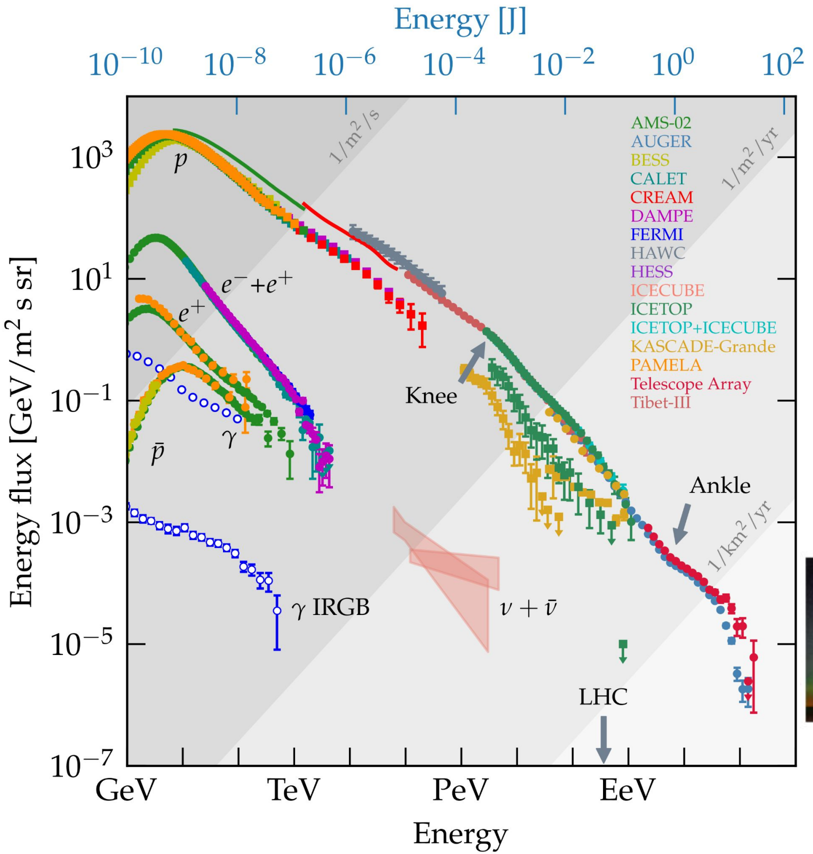
P. Savina

The Cosmic Ray Landscape

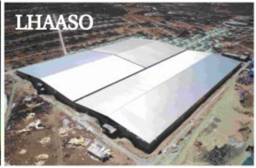




Direct measurements, can identify and precisely measure individual particle species, providing detailed information on the primary composition. Current experiments can measure up to energies of a few hundred TeV. This is achieved despite challenges as:
- containment
- limited effective area.



Spectral features at the highest energies can be probed using indirect detection experiments



Simulations

Computer simulation: reproduction of the behavior of a system using a computer to simulate the outcomes using a model associated to the system.

Complex problems (EAS simulations) broken down in smaller sub-problems.

Mathematical model: description of a system using mathematical concept and language.

used when is impractical to do a full simulation.
Models are based on simplifications, assumptions and approximations.

More simplifications lead to smaller "confidence level" (more verification needed).

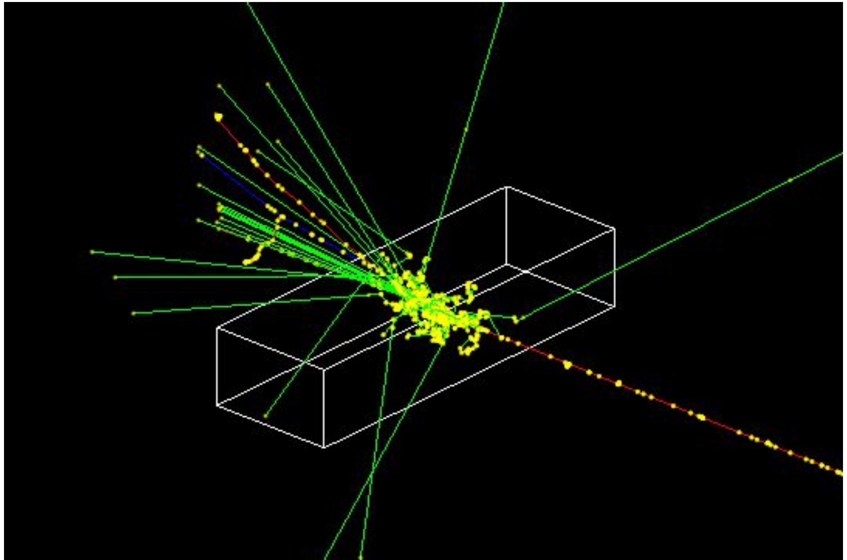
Monte Carlo Techniques: algorithms that rely on repeated random sampling to obtain numerical results. Their essential idea is using randomness to solve problems.

G S
S I

Simplified simulation pipeline in astroparticle physics

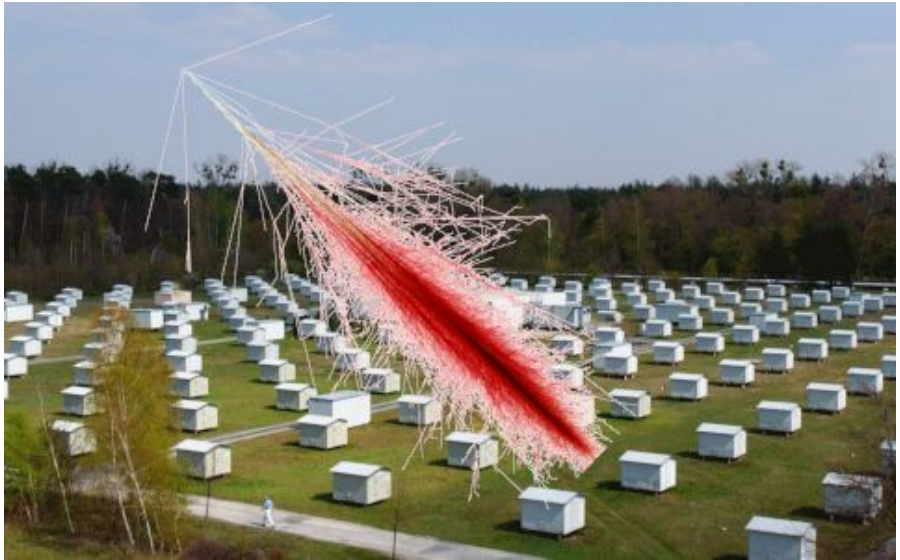
Direct Experiments

CR interactions with the detector simulated through Geant4 (or FLUKA)



Indirect Experiments

CR interactions with the atmosphere (CORSIKA + FLUKA, CONEX)
Secondary products - Detector interactions with Geant4



Outline

- CORSIKA 7 installation
- CORSIKA 7 exercises
- CORSIKA 8 installation
- Geant4 Exercises

CORSIKA

Cosmic Ray Simulation for KASCADE

consistent results in different experiments.

Models:

e.m. : **EGS4**

low-E hadronic: **FLUKA**, UrQMD, GHEISHA

high-E hadronic: **QGSJET**, EPOS-LHC, DPMJET, SIBILL

recommended

Models tuned at collider energies then extrapolated in the energy range considered

references:
CORSIKA
physics manual
user guide

Fair agreement from 10^{12} to 10^{20} eV.

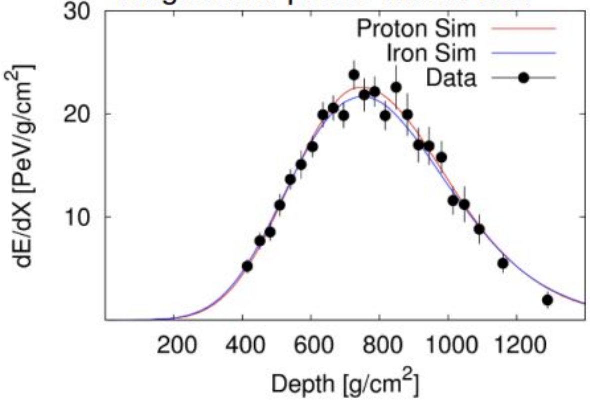
much better agreement at low energies where data constrains extrapolations.
At highest energies considerable extrapolation needed (high uncertainties).

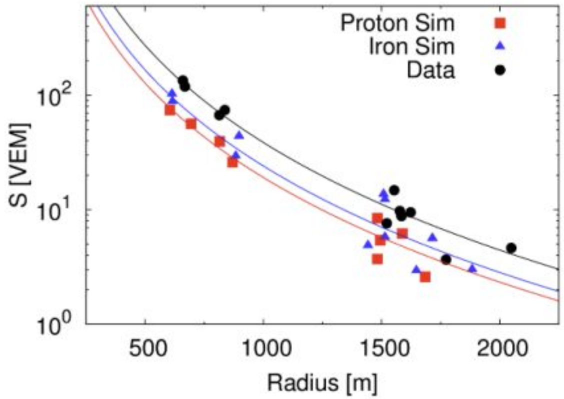
CORSIKA LIMITATIONS

Might be fixed in the newest EPOS-LHC-R model

Still testing

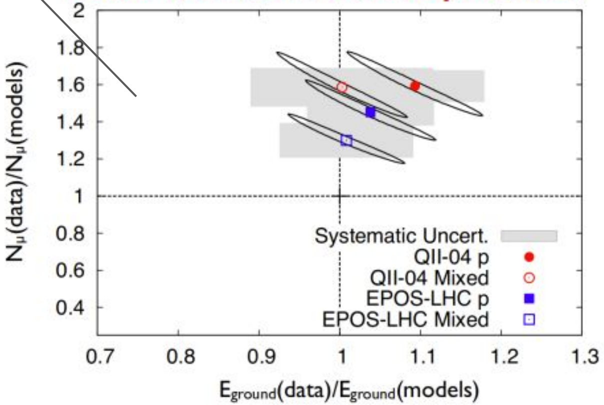
longitudinal profile match well

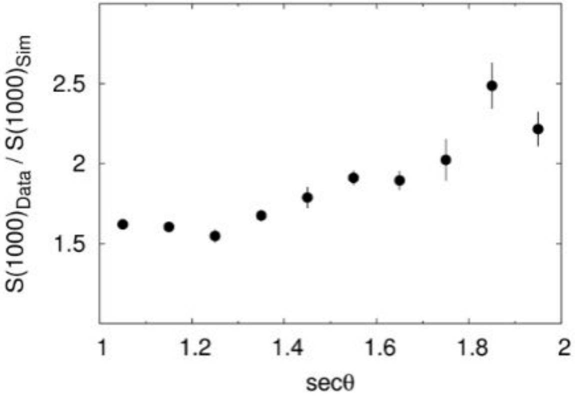




less signals at the ground in simulations

lower number of muons produced





CORSIKA7 vs CORSIKA8

- CORSIKA 7 widely used
- Written in Fortran 77
- **Used in major experiments such as KASCADE, Pierre Auger Observatory, IceCube** …

- Difficult to modify or extend.

- Mature codebase but limited by the aging technology and structure.

- CORSIKA 8 is a **complete redesign written in C++17**,
- Designed as a general-purpose particle propagation framework
- Allows easier integration with detector simulations and external software.

- **Modular architecture**
- Native support for **parallelization**
- Actively developed (beta version available)

Today's focus on CORSIKA7
Go through the installation procedure for CORSIKA8 together

CORSIKA7 Download

Please send an e-mail **requesting access to the CORSIKA program** to T. Pierog. This e-mail should also indicate, for which experiment at which university rsp. laboratory you will perform simulations. Then we will add your e-mail address to the CORSIKA mailing list and you will get an e-mail giving instructions how to download the files belonging to CORSIKA rsp. CONEX including the user name and password for the access.

Download

You may take the CORSIKA77550 version including all files belonging to it using an internet browser going to the URL: https://web.iap.kit.edu/corsika/download/ giving username: **corsika** and password: **(which you get by e-mail)** and change to the appropriate subdirectory by clicking on: **corsika-v780.** Finally you click on: **corsika-78010.tar.gz** to get the most recent version corsika-78000.

Download from here:
https://drive.google.com/drive/folders/1dYYBpZjT94b5LqLYiUR3HlLVCEiMiz7y?usp=sharing

Let’s setup the environment

if miniconda is not installed yet:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh
```

then

```
conda create -n geant4-env
conda activate geant4-env
conda install  -c conda-forge geant4 root gfortran_linux-64 make gcc_linux-64
```

CORSIKA7 INSTALL

Download from here:

Choose compilation mode of the machine:

[2] if you don’t care about compatibility

Must be the same used for FLUKA or ROOT if used

```
$ ./coconut


|==========================================================================|
|                        Welcome to COCONUT (v3.1)                         |
|               -- the CORSIKA CONfiguration UTility --                    |
|==========================================================================|
|                                                                          |
|                                                                          |
|           create an executable of a specific CORSIKA version            |
|                                                                          |
|                                                                          |
|        Please read the documentation for a detailed description         |
|                 of the options and how to use it.                        |
|                                                                          |
|            Try './coconut -h' to get some help about COCONUT            |
|   Use './coconut --expert' to enable additional configuration steps.    |
|                                                                          |
| (press 'Enter' to select an option followed by "[DEFAULT]" or "[CACHED]") |
|                                                                          |
|==========================================================================|



-------------------------------------------------------------------------
Compile in 32 or 64bit mode ?
   1 - Force 32bit mode
   2 - Use compiler default ('-m64' on a 64bit machine) [DEFAULT]

   r - restart (reset all options to cached values)
   x - exit make

   (only one choice possible):
```

CORSIKA7 INSTALL II

```
-----------------------------------------------------------------------
Which high energy hadronic interaction model do you want to use ?
   1 - DPMJET-III (2017.1) with PHOJET 1.20.0
   2 - EPOS LHC [CACHED]
   3 - NEXUS 3.97
   4 - QGSJET 01C (enlarged commons)
   5 - QGSJETII-04
   6 - SIBYLL 2.3d
   7 - VENUS 4.12

   r - restart (reset all options to cached values)
   x - exit make
```

HIGH-E Hadronic

Up to date:
EPOS-LHC, QGSJetII-04, Sybill2.3c (DPMJETIII to come)

Reference:
QGSJet01

Others for special use.

LOW-E Hadronic

GHEISHA: too old (only for test)

FLUKA(recommended): can be installed defining **$FLUPRO** to point to the fluka installation path. Subscription to FLUKA needed.

```
-----------------------------------------------------------------------
Which low energy hadronic interaction model do you want to use ?
   1 - GHEISHA 2002d (double precision)
   2 - FLUKA-CERN
   3 - FLUKA  [CACHED]
   4 - URQMD 1.3cr

   r - restart (reset all options to cached values)
   x - exit make

   (only one choice possible):
```

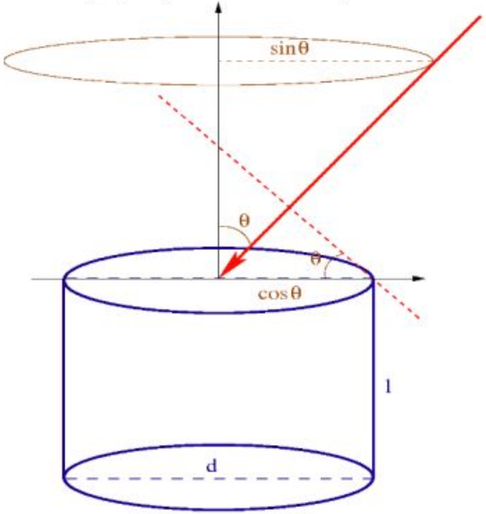
CORSIKA7 INSTALL III

Detector geometry only change the angular distribution of showers.

```
Which detector geometry do you have ?
   1 - horizontal flat detector array [DEFAULT]
   2 - non-flat (volume) detector geometry
   3 - vertical string detector geometry

   r - restart (reset all options to cached values)
   x - exit make

   (only one choice possible): 
```

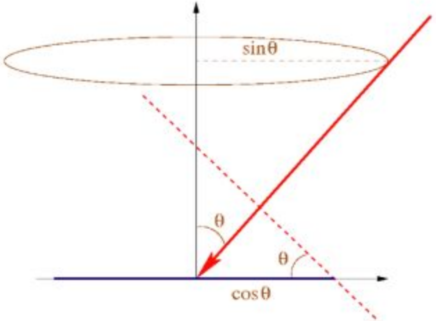
Vertical string detector

$$I \propto (d/2)^2 \pi \sin\theta \cdot (\cos\theta + 4/\pi l/d \sin\theta)$$



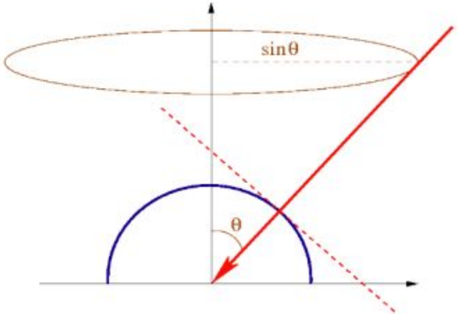
Flat experiment

$$I \propto \cos(\theta)\sin(\theta)$$



Non-flat experiment

$$I \propto \sin(\theta)$$



G S
S I

CORSIKA7 INSTALL IV

```
Which additional CORSIKA program options do you need ?
   1a - Cherenkov version
   1b - Cherenkov version using Bernlohr IACT routines (for telescopes)
   1c - apply atm. absorption, mirror reflectivity & quantum eff.
   1d - Auger Cherenkov longitudinal distribution
   1e - TRAJECTory version to follow motion of source on the sky
   2  - LPM-effect without thinning
   2a - THINning version (includes LPM)
   2b - MULTIple THINning version (includes LPM)
   3  - PRESHOWER version for EeV gammas
   4  - NEUTRINO version
   4a - NUPRIM primary neutrino version with HERWIG
   4b - ICECUBE1 FIFO version
   4c - ICECUBE2 gzip/pipe output
   5  - STACK INput of secondaries, no primary particle
   6  - CHARMed particle/tau lepton version with PYTHIA
   6a - TAU LEPton version with PYTHIA
   7  - SLANT depth instead of vertical depth for longi-distribution
   7a - CURVED atmosphere version
   7b - UPWARD particles version
   7c - VIEWCONE version
   8a - shower PLOT version (PLOTSH) (only for single events)
   8b - shower PLOT(C) version (PLOTSH2) (only for single events)
   8c - ANALysis HISTos & THIN (instead of particle file)
   8d - Auger-histo file & THIN
   8e - MUON-histo file
   9  - external atmosphere functions (table interpolation)
        (using bernlohr C-routines)
   9a - EFIELD version for electrical field in atmosphere
   9b - RIGIDITY Ooty version rejecting low-energy primaries entering Earth-magnetic field
   10a - DYNamic intermediate particle STACK
   10b - Remote Control for Corsika
   a  - CONEX for high energy MC and cascade equations
   b  - PARALLEL treatment of subshowers (includes LPM)
   c  - CoREAS Radio Simulations
   d1 - Inclined observation plane
   e  - interaction test version (only for 1st interaction)
   f  - Auger-info file instead of dbase file
   g  - COMPACT particle output file
   h  - MUPROD to write decaying muons
   h2 - prEHISTORY of muons: mother and grandmother
   k  - annitest cross-section version (obsolete)
   l  - hit Auger detector (steered by AUGSCT)
      - ----------------------------------------
   y  - *** Reset selection ***
   z  - *** Finish selection *** [DEFAULT]

   r  - restart (reset all options to cached values)
   x  - exit make
```

no additional option will be used for the exercises.

2 useful options will be described.

2a) THINNING: save time computation by reducing the number of particles; a particle randomly selected carry a weight related to all particles produced at the same time to conserve energy.

a) CONEX: use cascade equations to reduce simulation time.

CORSIKA7 INSTALL V

```
--------------------------------------------------------------------------------
Configuration is finished. How do you want to proceed ?
   f - Compiling and remove temporary files [DEFAULT]
   k - Compile and keep extracted CORSIKA source code
   n - Just extract source code. Do not compile!

   r - restart (reset all options to cached values)
   x - exit make

   (only one choice possible):
```

source not saved by default.
using “k” source can be saved to check what is used in the code.

```
checking parallel computation with MPI... (cached) no
checking do not compile binaries, just extract CORSIKA compilefile... (cached) no
checking to keep the CORSIKA compilefile... (cached) no
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating baack/Makefile
config.status: creating bernlohr/Makefile
config.status: creating conex/Makefile
config.status: creating dpmjet/Makefile
config.status: creating epos/Makefile
config.status: creating pythia/Makefile
```

incompatible option or missing declaration reported here

CORSIKA7 SUMMARY

- Install CORSIKA from tar file.
- Produce different binaries.
- Edit a steering card.
- Run a simulation.
- Analyze the output.

Simulating a shower

```
~/ISAPP_2025/Ex1/corsika-77550/run$ ./corsika77550Linux_EPOS_urqmd < all-inputs-epos
```

```
RUNNR   18                            run number
NSHOW   1                            number of showers to generate
PRMPAR  14                           prim. particle (1=gamma, 14=proton, ...)
ESLOPE  -2.7                         slope of primary energy spectrum
ERANGE  1.E6  1.E6                   energy range of primary particle (GeV)
THETAP  0.  0.                       range of zenith angle (degree)
PHIP    -180.  180.                  range of azimuth angle (degree)
SEED    1   0   0                    seed for 1. random number sequence
SEED    2   0   0                    seed for 2. random number sequence
*THIN    1.E-2  1.E2  0.              thinning definition
*THINH   10.  10.                     relative threshold and weight for hadron thinning
OBSLEV  100000.                          observation level (in cm)
MAGNET  20.0  42.8                   magnetic field centr. Europe
HADFLG  0  0  0  0  0  2             flags hadr.interact.&fragmentation
ECUTS   1.  1.  0.001  0.001         energy cuts for particles
MUADDI  T                            additional info for muons
MUMULT  T                            muon multiple scattering angle
ELMFLG  T   F                        em. interaction flags (NKG,EGS)
STEPFC  1.0                          mult. scattering step length fact.
RADNKG  200.E2                       outer radius for NKG lat.dens.distr.
EPOPAR input ../epos/epos.param       !initialization input file for epos
EPOPAR fname inics ../epos/epos.inics  !initialization input file for epos
EPOPAR fname iniev ../epos/epos.iniev  !initialization input file for epos
EPOPAR fname initl ../epos/epos.initl  !initialization input file for epos
EPOPAR fname inirj ../epos/epos.inirj  !initialization input file for epos
EPOPAR fname inihy ../epos/epos.ini1b  !initialization input file for epos
EPOPAR fname check none               !dummy output file for epos
EPOPAR fname histo none               !dummy output file for epos
EPOPAR fname data  none               !dummy output file for epos
EPOPAR fname copy  none               !dummy output file for epos
LONGI   T  10.  T  T                 longit.distr. & step size & fit & outfile
ECTMAP  1.E11                         cut on gamma factor for printout
MAXPRT  1                            max. number of printed events
DIRECT  ./                           output directory
USER    you                          user
*PAROUT F F                           suppress DAT file
DEBUG   F  6  F  1000000             debug flag and log.unit for out
EXIT                                 terminates input
```

```
DAT000018.long
DAT000018
```

Exercise

```
RUNNR 1
NSHOW 50
ESLOPE -1
ERANGE 1E2 1E4
THETAP 20. 70.
OBSLEV 410000
ELMFLG F T
```

Read the output

git clone https://gitlab.com/psavina_public_projects/corsika-hands_on
Compile coast (go underder corsika-77800/coast and then type make ; make install)
Modify the Makefile (COAST_DIR has to point to your current corsica folder)

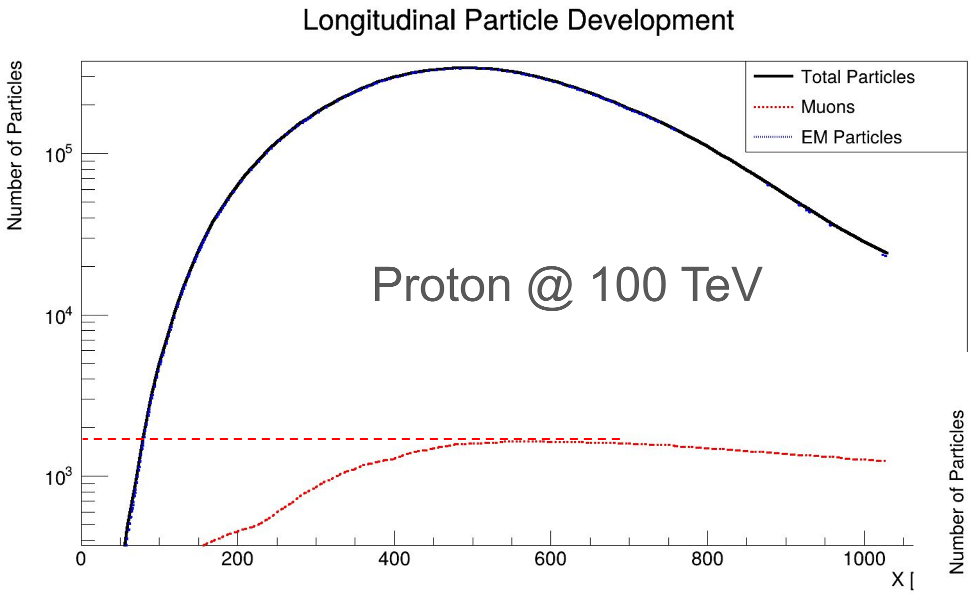
different examples to read the output files:

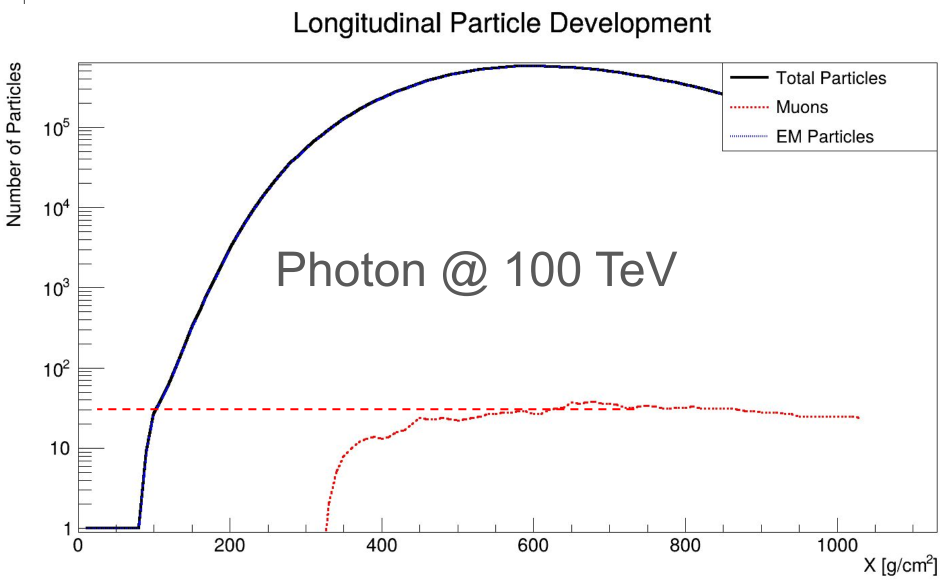
- energySpectra.cc → energy spectrum of the generated showers
- angularDistribution.cc → angular distribution of the generated shower
- groundMomenta.cc → momentum distribution of the ground particles*
- footprint.cc → plot the footprint*
- dummySim.cc → simulation of a over-simplified detector*
- longReader.cc → plot of the longitudinal development of the shower*

Compile
type: **make**

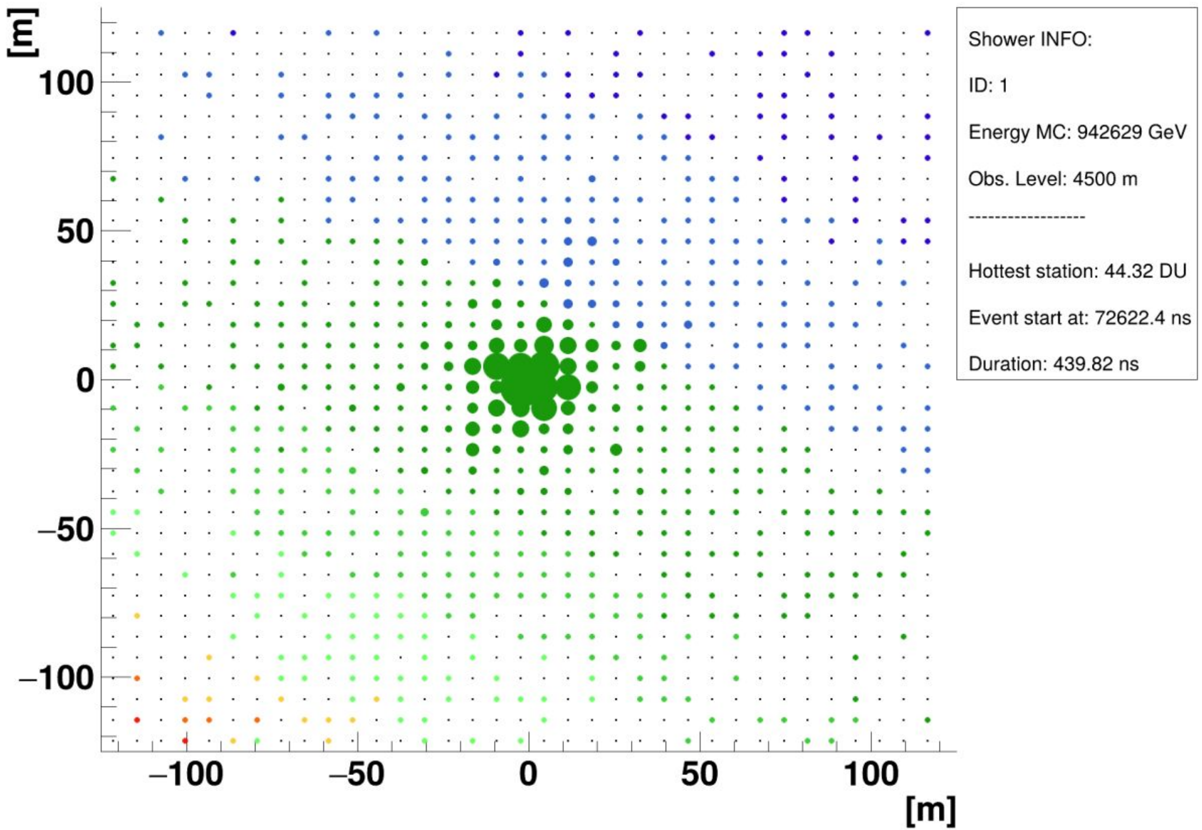
* only for a chosen shower

Protons vs Photons



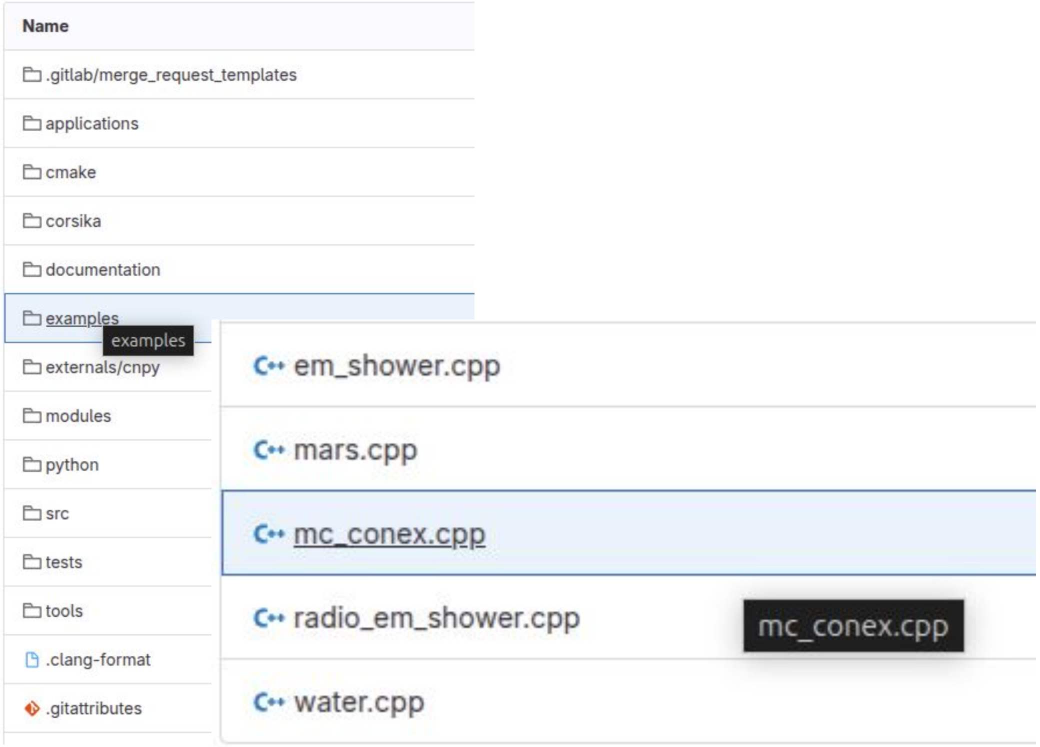


Read the output



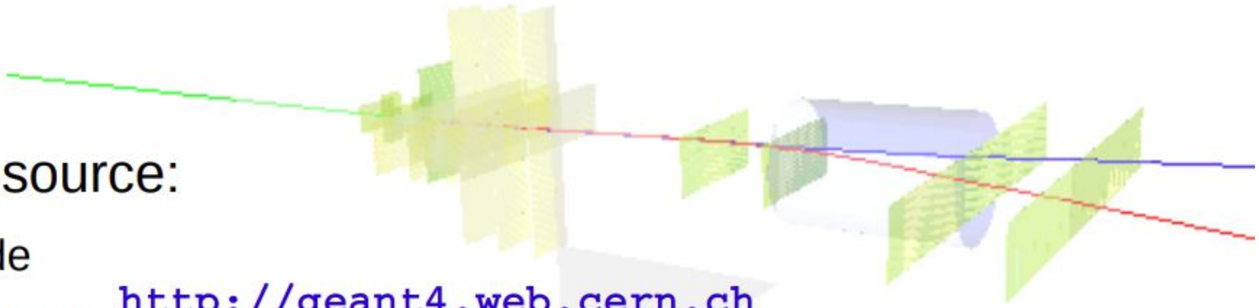
CORSIKA 8

https://gitlab.iap.kit.edu/AirShowerPhysics/corsika/-/tree/corsika8-v1.0-beta1?ref_type=tags



Simulation toolkit in C++:

- Variety of geometries → choose your own setup
- Variety of materials → choose your own materials
- Variety of particles → choose particle type energy position direction
- Variety of physics processes → available physics models, cuts



Open source:

- Code
- Manuals
- Examples

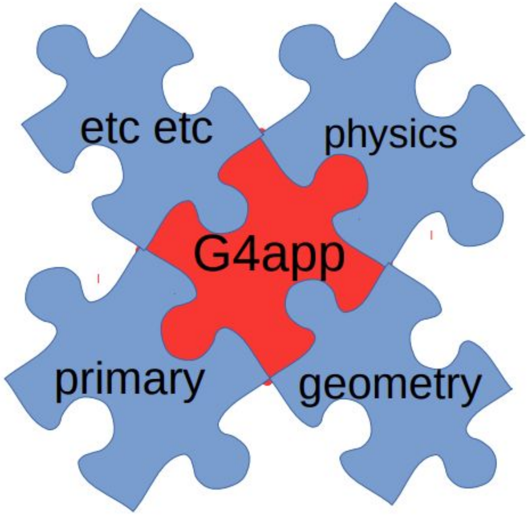
http://geant4.web.cern.ch

Nucl. Inst. and Methods Phys. Res. A, 506, 250

Transaction on Nuclear Science 53, 270

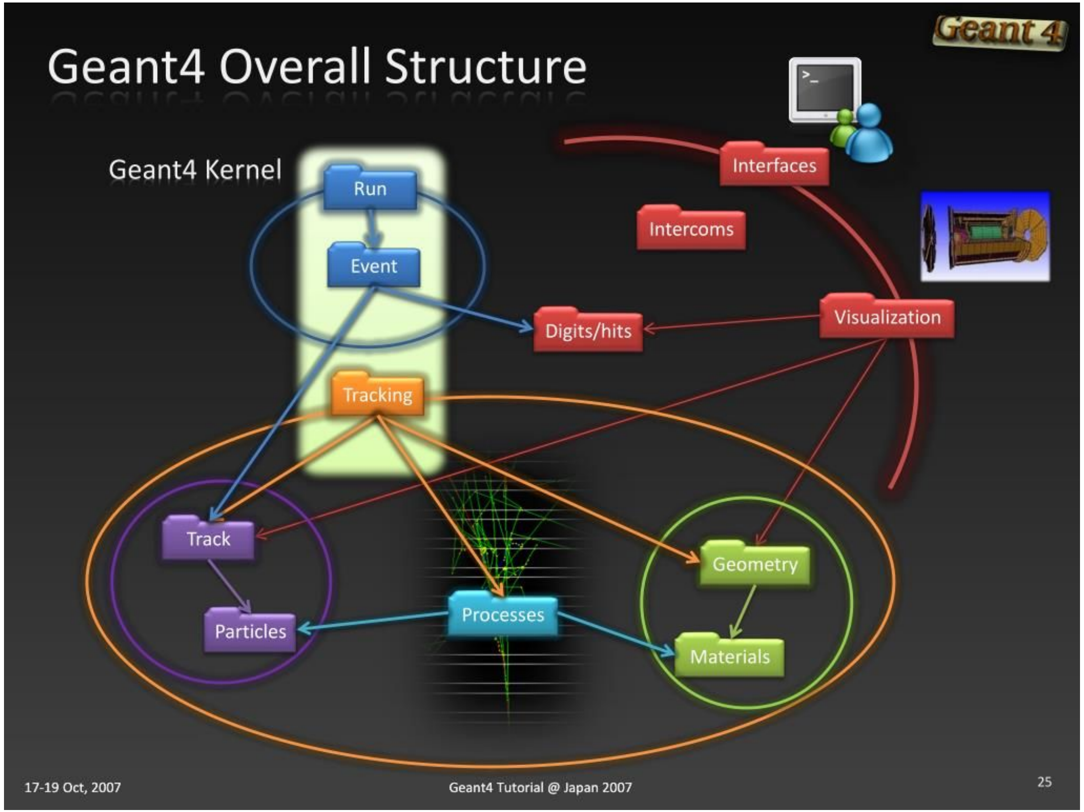
G4 Components

particles processes



particles
generator

detector
geometries

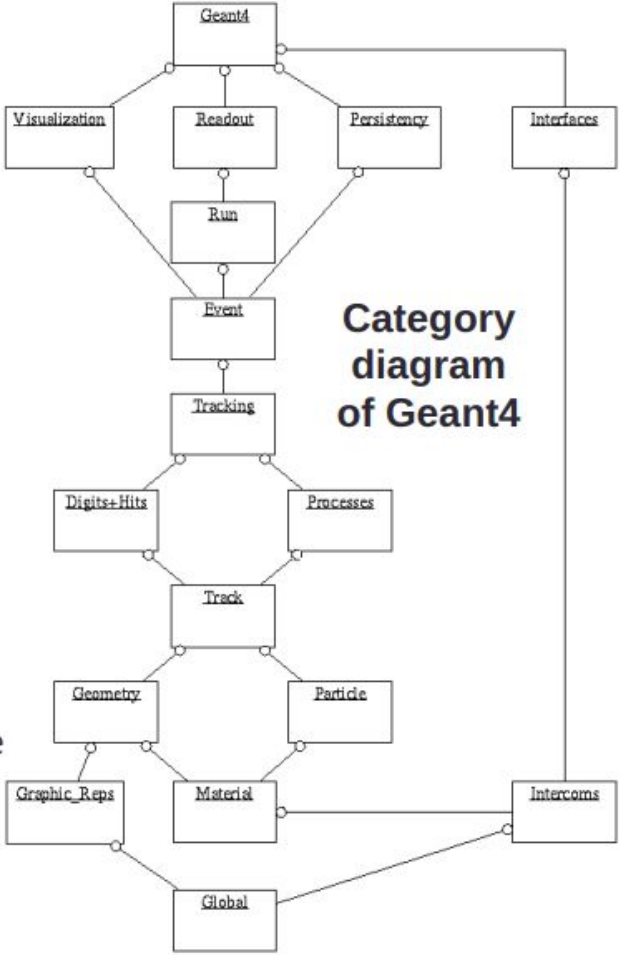


The Toolkit philosophy

- **<u>Geant4 is not an application</u>**
  - applications : eg powerpoint, root, etc.

Geant4 is a toolkit

- Which means:
  - **Geant4 provides tools / components**
    - Many of them are defined from abstract classes
    - All are open to the users (☞ you)
  - **You build your own application selecting the Geant4 components you need**
    - Either selecting ready to use tools
    - Or building your own, if needed, from the base abstract classes
  - **You instantiate the components in your own main program**
    - That you then compile and link
- You need a minimal knowledge of the Geant4 structure
  - And of the Geant4 base classes and existing tools
- Which is all what this tutorial is about !



Category diagram of Geant4

Geometry

Very complex geometries can be described at three levels:

- **Solid**: shapes and dimensions, boolean operations
- **LogicalVolume**: materials, sensitivity, mother and daughter volumes, visualization etc
- **PhysicalVolume**: position in space and rotation

  a logical volume can be placed multiple times originating different physical volumes

  → single or repeated placements (replicas,parameterizations)

Physics

- electromagnetic interactions for all particles
- inelastic interactions
- elastic scattering
- capture
- decay of unstable particles

plug&play Geant4 physics list

`$G4SOURCE/ source/physics_lists/lists`

```
class MyPhysicsList: public G4VUserPhysicsList {
public:
MyPhysicsList();
~MyPhysicsList();
void ConstructParticle();
void ConstructProcess();
void SetCuts();
}
```

user can implement the methods to define particles processes and cuts (range based) on generation of secondaries ex. delta rays from ionization, or gamma from bremsstrahlung

Tracking

Transportation of a particle ‘step-by-step’taking into account all possible interactions with materials and fields

The transport ends if the particle:

- is slowed down to zero kinetic energy (and it doesn't have any interaction at rest)
- disappears in some interaction
- reaches the end of the simulation volume

Geant4 allows the User to access the transportation process and retrieve the results (USER ACTIONS)

- at the beginning and end of the transport
- at the end of each step in transportation
- if a particle reaches a sensitive detector

Compile your first example

examples can be found under:

path-to-miniconda3/envs/geant4-env/share/Geant4/examples/

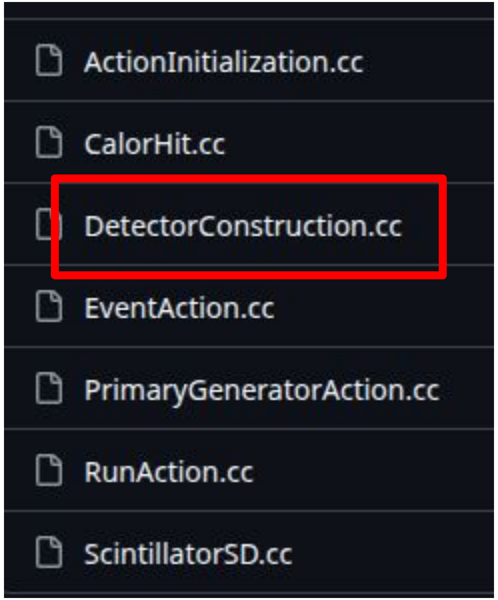
to compile

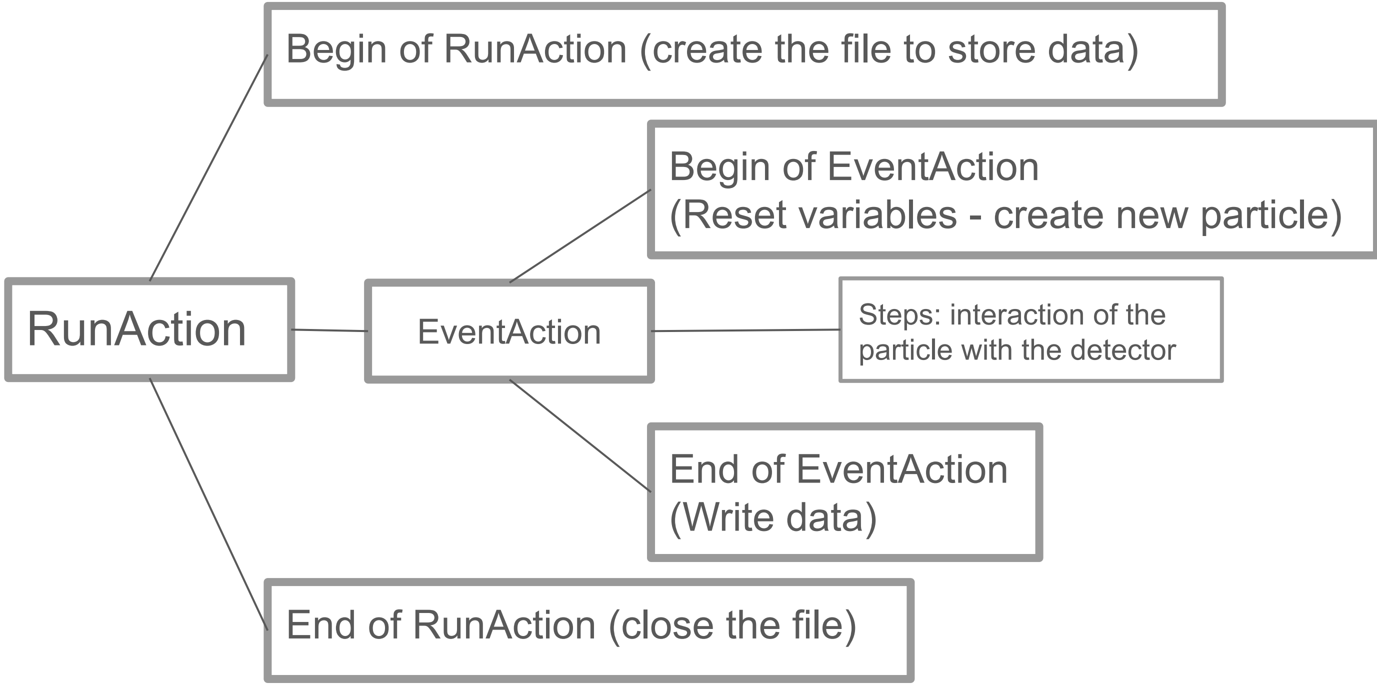
```
mkdir build
cd build
cmake path-to-src -DCMAKE_PREFIX_PATH=$CONDA_PREFIX
```

Exercise repositoris:
https://github.com/psavina/ISAPP2025_G4_SimpleSim.git
https://github.com/psavina/G4CORSIKA.git * needs CORSIKA 8

Ex1 - Simple Scintillator

Heavily inspired on example/basic/B4/B4c





Ex1 - Geometry

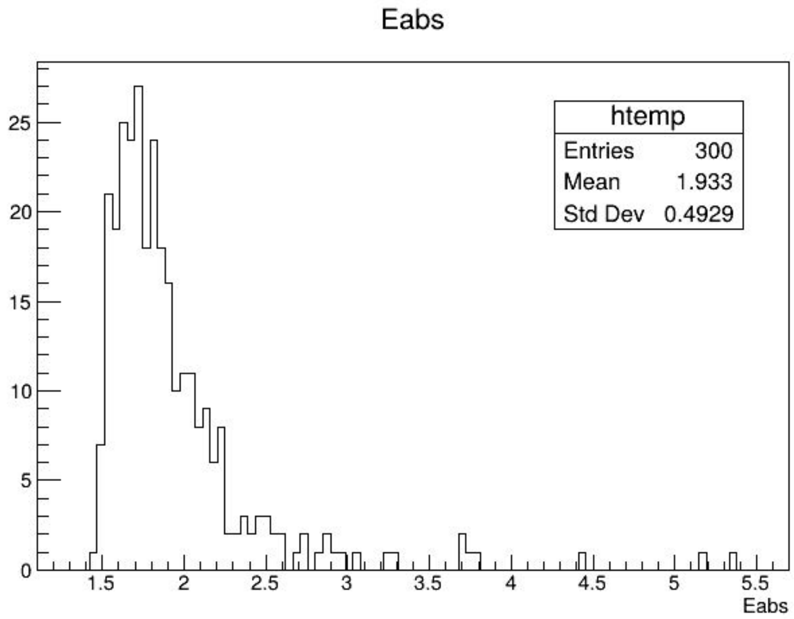
```
auto worldS
  = new G4Box("World",           // its name
           worldSizeXY/2, worldSizeXY/2, worldSizeZ/2); // its size

auto worldLV
  = new G4LogicalVolume(
           worldS,           // its solid
           defaultMaterial,  // its material
           "World");         // its name

auto worldPV = new G4PVPlacement(nullptr,  // no rotation
  G4ThreeVector(),                         // at (0,0,0)
  worldLV,                                 // its logical volume
  "World",                                 // its name
  nullptr,                                 // its mother  volume
  false,                                   // no boolean operation
  0,                                       // copy number
  fCheckOverlaps);                         // checking overlaps
```

```
auto scintillatorS
  = new G4Box("Scintillator",     // its name
           sizeXY/2, sizeXY/2, thickness/2); // its size

auto scintillatorLV
  = new G4LogicalVolume(
           scintillatorS,     // its solid
           scintMaterial,  // its material
           "ScintillatorLV");   // its name

new G4PVPlacement(nullptr,  // no rotation
  G4ThreeVector(0., 0., 0.0*cm),          //
  scintillatorLV,                 // its logical volume
  "Scintillator",           // its name
  worldLV,                 // its mother  volume
  false,                   // no boolean operation
  0,                       // copy number
  fCheckOverlaps);         // checking overlaps
```

Ex1 - Read Output

Output is in root format so you can easily read it with python or root:

root B4.root
B4->Draw(“Eabs”)



conda install -c conda-forge uproot matplotlib

```
import uproot as ur
import matplotlib.pyplot as plt


f = ur.open("B4.root")
t = f['B4']
e = t['Eabs'].array()

plt.hist(e, bins=100)
plt.xlabel("E$_{abs}$")
plt.tight_layout()
plt.show()
```