ICRM GSWG : ACTION TO FACILITATE THE USE OF GEANT4 IN GAMMA–SPECTROMETRY

Laboratoire National Henri Becquerel

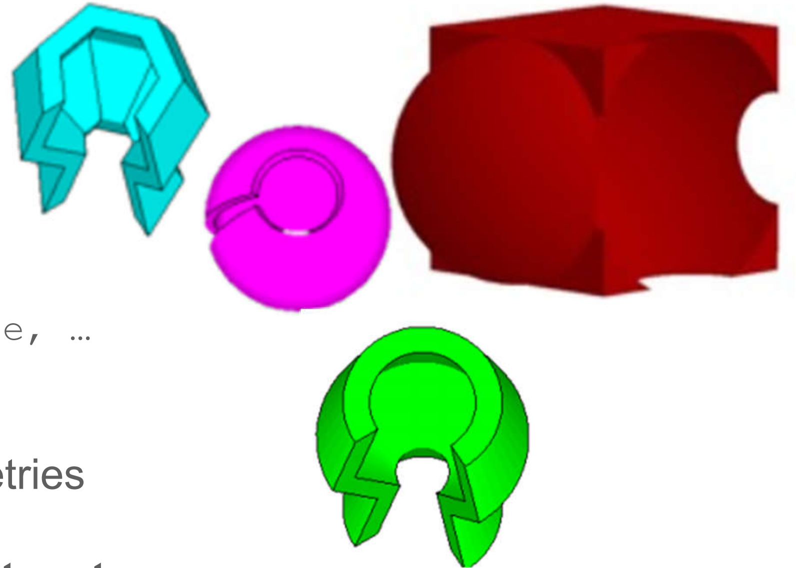
ICRM-GSWG meeting – 14th June 2018 | Cheick THIAM

INTRODUCTION TO GEANT4

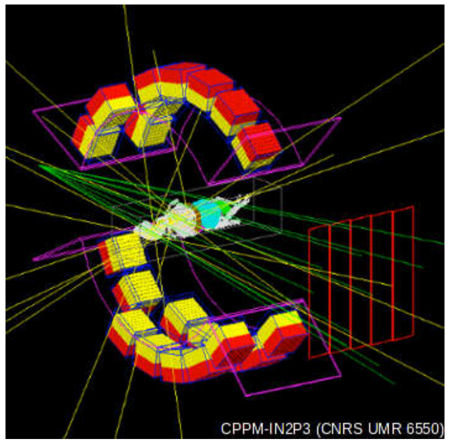
- **Geant = « GEometry ANd Tracking »**
  - General purpose Monte Carlo toolkit (free) for simulating the passage of particles through matter and interacting with it
  - Written in C++ and exploits advanced software-engineering techniques and object-oriented technology to achieve transparency
- **Wide variety of user domains**
  - High energy and nuclear physics
  - Space engineering and medical applications
  - Material science, radiation protection and security…
- **Geant4 offers lots of the functionalities required for the simulation**
  - Geometry and navigation,
  - Physical processes governing particles interactions
  - Visualization of the detector and particle trajectories (OpenGL, VRML….)
  - Data analysis at different levels of detail and refinement
- **User must build his own application by selecting the Geant4 components**
  - Either selecting ready to use tools, or building his own from the base abstract classes
    - ➔ Need a minimal knowledge of the Geant4 structure and base classes
    - ➔ Need a basic knowledge in Linux and C++ programming

GEANT4 : KEY GEOMETRY CAPABILITIES

- **Richest collection of shapes**
  - CSG (Constructed Solid Geometry) :
    - `G4Box, G4Tubs, G4Cons, G4Sphere, …`
  - Specific solids (CSG like)
    - `G4Polycone, G4Polyhedra, G4Hype, …`
  - BREP (Boundary REPresented) solids
    - `G4BREPSolidPolycone, G4BSplineSurface, …`
  - Boolean solids
    - `G4UnionSolid, G4SubtractionSolid, …`
  - The user can easily extend for complex geometries



- **Describing a setup as a hierarchy or 'flat' structure**
  - Describing setups up to billions of volumes
  - Tools for creating & checking complex structures
  - GDML (Geometry Description Markup Language) : XML-based format that enables to describe many aspects of geometry
    ➔ Import CAD (3D modeler for design)



- **Geometry models can be 'dynamic'**
  - Changing the setup at run-time e.g. "moving objects"

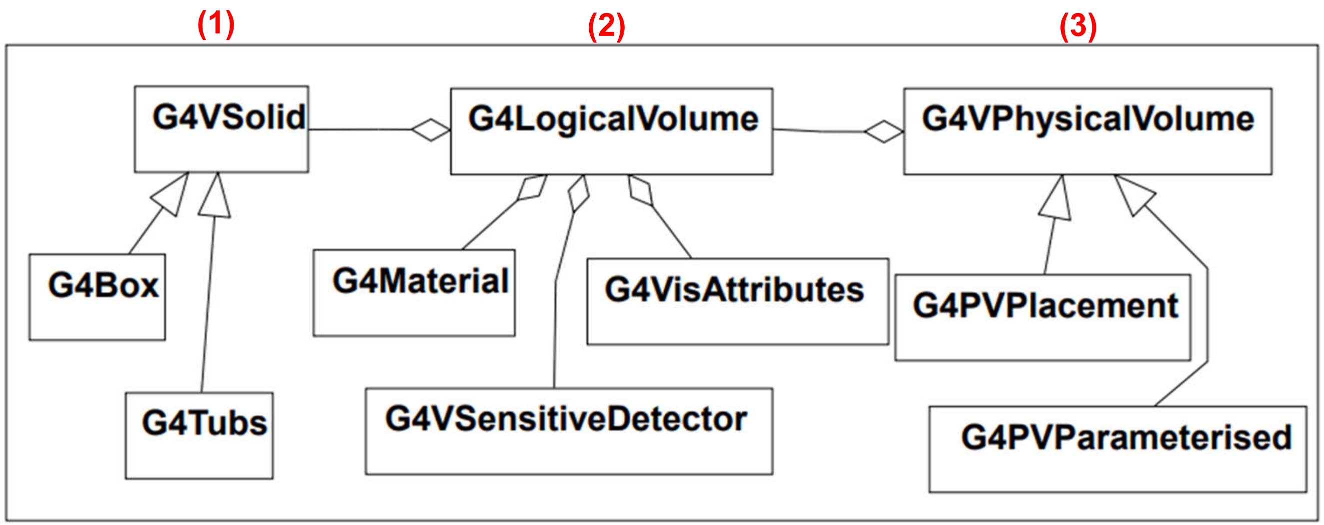
GEANT4 : KEY GEOMETRY CAPABILITIES

- **Three conceptual layers to define volumes and geometries**

`G4VSolid` : *shape, size*

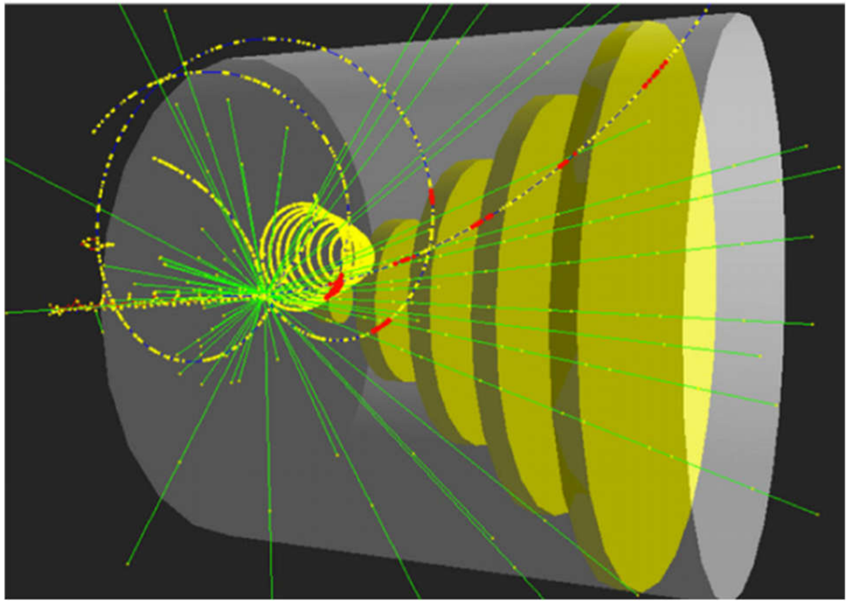
`G4LogicalVolume` : *G4VSolid, material, sensitivity, visualization attributes, daughter physical volumes, user limits, etc.*

`G4VPhysicalVolume` : *position: rotation of mother frame, position in mother frame; repeated Volumes: a single physical volume represents multiple copies of a volume*



➔ A unique physical volume which represents the experimental area must exist and contains all the other components *"The world volume"*

GEANT4 : PHYSICS MODELS IN GEANT4

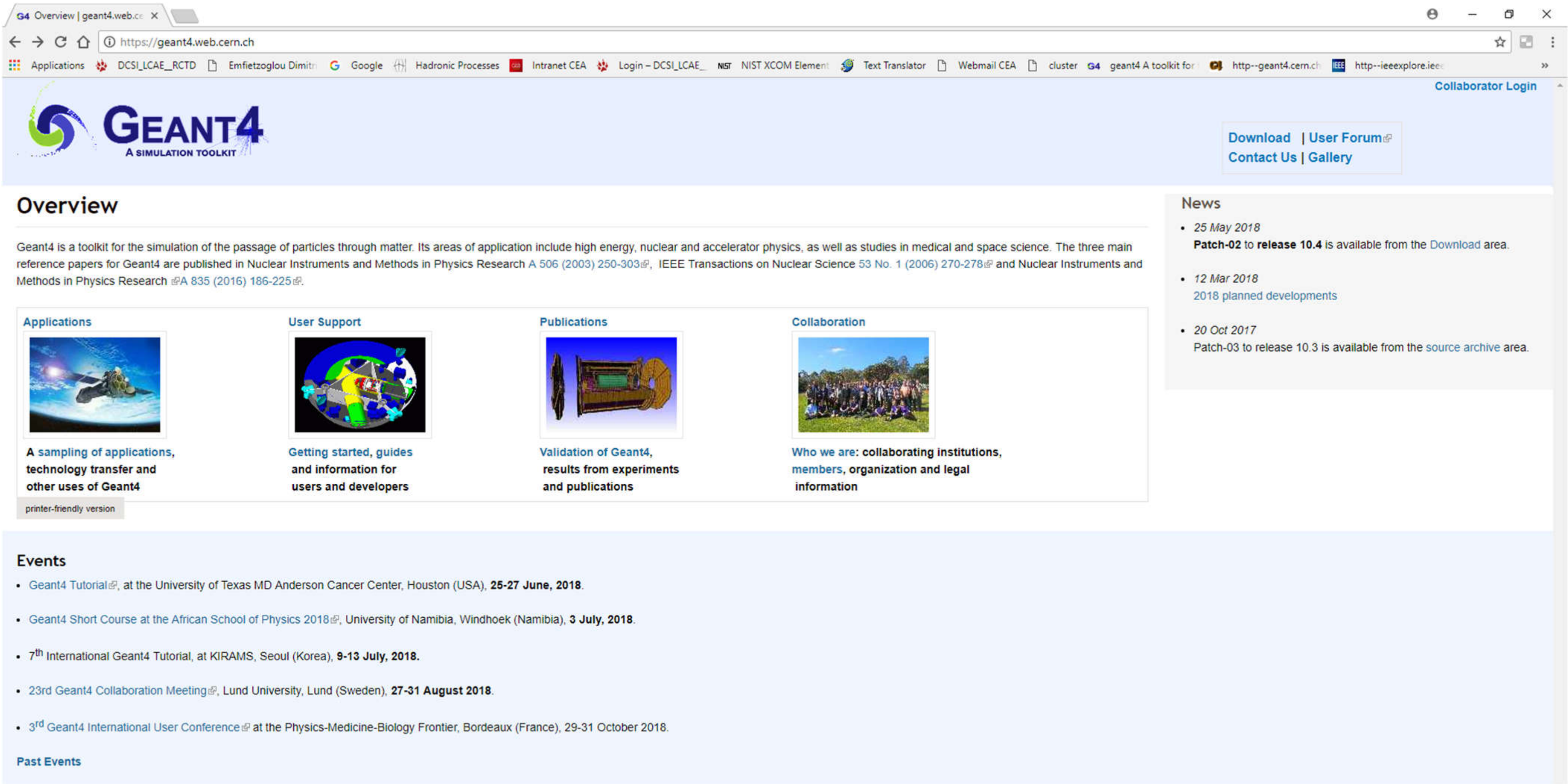


- **Geant4 offers**
  - Electromagnetic processes (EM)
  - Hadronic and nuclear processes
  - Photon/lepton-hadron processes
  - Optical photon processes
  - Decay processes
  - Atomic rearrangement (KLM model)
  - Shower parameterization
  - Event biasing techniques
  - And you can plug-in more…

- **Wide set of physical models provided**
  - Complementary models with different energy range applicability
    - That can be combined to cover a wide range of energy
  - Competing models with same energy range applicability
    - That can be selected by the user (for example, some models are more accurate than others (in that case, speed is sacrificed)

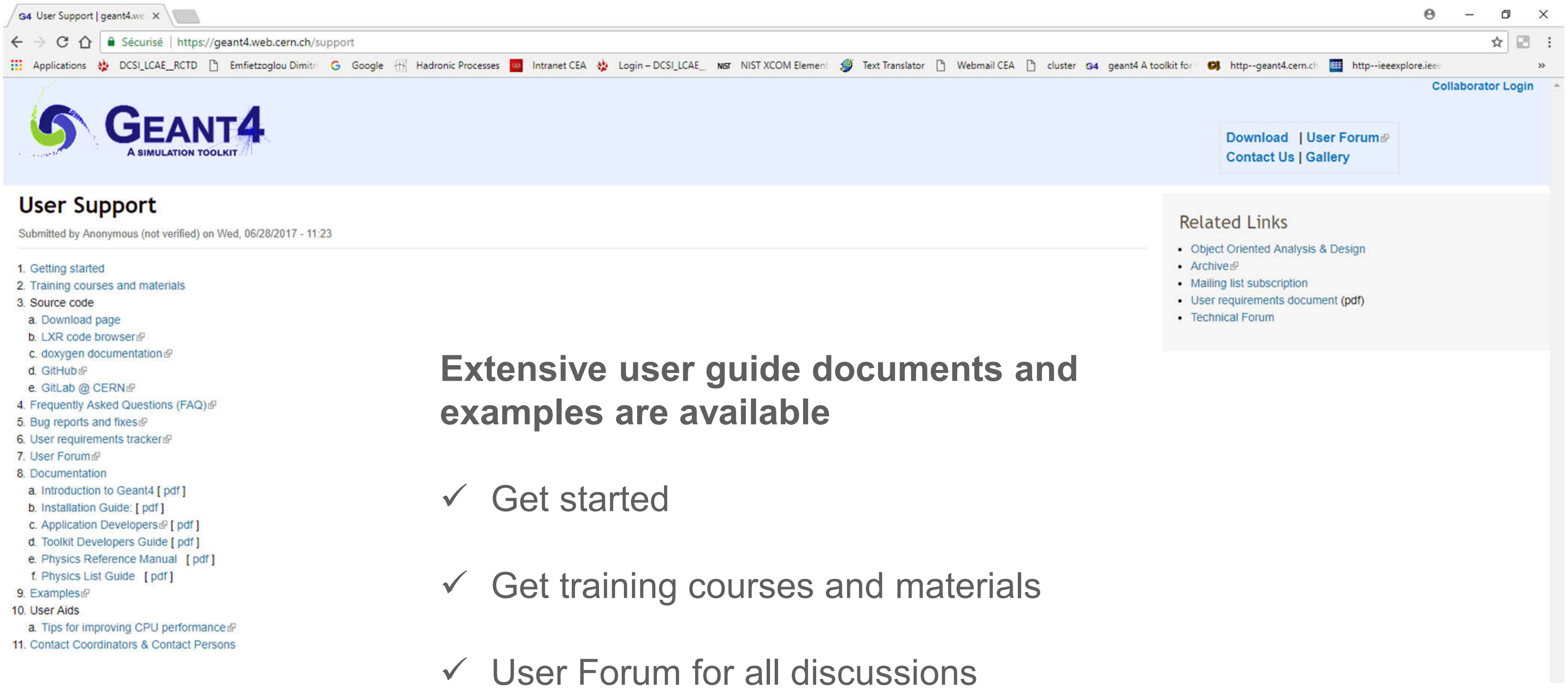
GEANT4 : THE TOOLKIT AND SUPPORT (1)

➔ https://geant4.web.cern.ch



GEANT4 : THE TOOLKIT AND SUPPORT (2)

➔ https://geant4.web.cern.ch/support



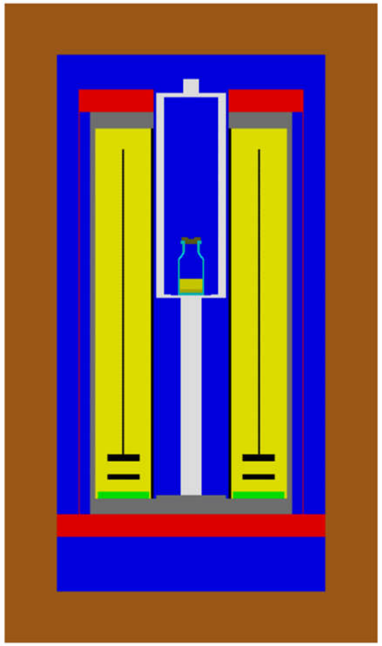
Extensive user guide documents and examples are available

- ✓ Get started
- ✓ Get training courses and materials
- ✓ User Forum for all discussions

RADIATION DETECTORS USED IN METROLOGY

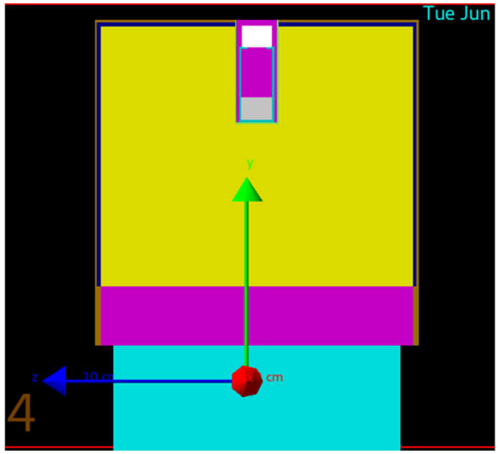
Ionization chambers

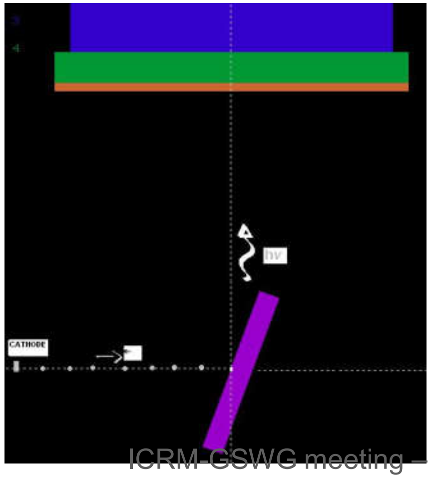
→ calibration of coefficients calculation



γ-ray spectrometry

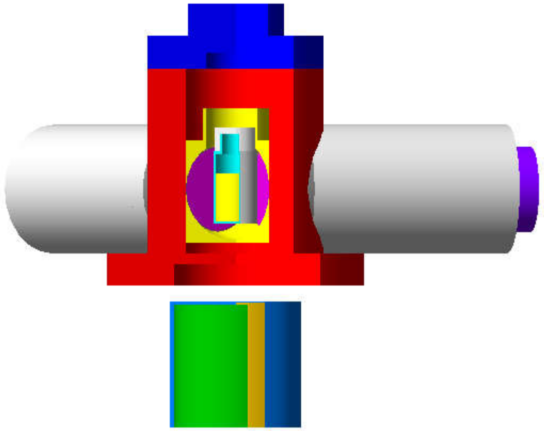
→ calculation of detection efficiencies





TDCR counter (Triple to Double Coincidence Radio)

→ calculation of detection efficiencies



X-ray tub

→ calculation of correction factors

GAMMA-RAY SPECTROMETRY BENCHMARK

- **« `GammaRaySpec` » : γ-ray spectrometry dedicated benchmark**

- **Use the Geant4 Monte Carlo code to investigate the response of a HPGe-detector**

  ➔ How to compute full-energy-peak and total-energy efficiencies

  ➔ How to compute coincidence-summing corrections from an energy sprectrum

  ➔ Plot histograms (ex. energy deposit in Ge-crystal)

  ➔ Investigate the dead zone effect of Ge-Cristal, variation of physics models etc

GAMMA-RAY SPECTROMETRY BENCHMARK

- « `GammaRaySpec` » : γ-ray spectrometry dedicated benchmark

(1) Geometry: defined in the dedicated class `GRSDetectorConstruction.cc /.hh`

➔ Contains a set of well-defined HPGe detector and sample geometries

➔ User can selects a given configuration with UI commands lines

Detector parameters

Parameter	Detector A	Detector B
Crystal material	Ge	Ge
Crystal diameter (including the side dead slayer)	60	60
Crystal length (including the top dead layer)	60	60
Dead layer thickness (top and side)	1	0
Hole diameter	10	10
Hole depth	40	40
Window diameter	80	80
Window thickness	1	1
Window material	Al	Al
Crystal-to-window distance	5	5
Housing length	80	80
Housing thickness	1	1
Housing material	Al	Al

8 cases source-to-detector configuration

Name	Detector	Source
AP	A	Point
AW	A	Water
AS	A	Soil
AF	A	Filter
BP	B	Point
BW	B	Water
BS	B	Soil
BF	B	Filter

Sample parameters

Parameter	Water	Point	Soil	Filter
Sample diameter	90	-	60	80
Sample thickness	40	-	20	3
Sample material	Water	-	Dirt	Cellulose
Sample-to-window distance	1.0	1.0	1.0	1.0

GAMMA-RAY SPECTROMETRY BENCHMARK

- « `GammaRaySpec` » : γ-ray spectrometry dedicated benchmark

(2) Physics: defined in the dedicated class `GRSPhysicsList.cc /.hh`

➔ **Physics lists are based on modular design. Several modules are instantiated:**

- Transportation
- EM physics (electron photon mode)
- Decays
- Atomic relaxation

➔ EM physics builders are from G4 kernel physics_lists subdirectory. Physics lists and options can be (re)set with UI commands, or in the input macro file

➔ Four choices are offered:

`"emstandard"` standard EM physics with current 'best' options setting,

`"emlivermore"` low-energy EM physics using Livermore data,

`"emlowenergy"` low-energy EM physics implementing experimental low-energy models,

`"empenelope"` low-energy EM physics implementing Penelope models.

GAMMA-RAY SPECTROMETRY BENCHMARK

- « `GammaRaySpec` » : γ-ray spectrometry dedicated benchmark

(3) Primary generator: defined in the class `GRSPrimaryGeneratorAction.cc /.hh`

➔ The primary kinematic consists of a single particle which hits the Ge detector

➔ The type of the particle and its energy can be set in the macro file (ex. detectorA.mac) or via the G4 build-in commands of G4ParticleGun

➔ User can also set the *Point source* or Volume source (*Water, Soil or Filter*)

(4) Detector response: defined in the class `GRSAnalysisManager.cc /.hh`

➔

```
--------------------End of Global Run-----------------------

 ----> print histograms statistic

     ---Gamma-Ray-Spectrometry--- RunStatus

       - total events generated  :       100000
     [ Peak Efficiency ]
       - peak count              :         2521
       - peak efficiency         :      0.02521
       - sigma (%)               :      1.99165
     [ Total Efficiency ]
       - total count             :         2727
       - total efficiency        :      0.02727
       - sigma (%)               :      1.91495

... write Root file : GammaRaySpec.root - done
... close Root file : GammaRaySpec.root - done

 ... histograms and ntuples are saved
```

GAMMA-RAY SPECTROMETRY BENCHMARK

- « `GammaRaySpec` » : γ-ray spectrometry dedicated benchmark

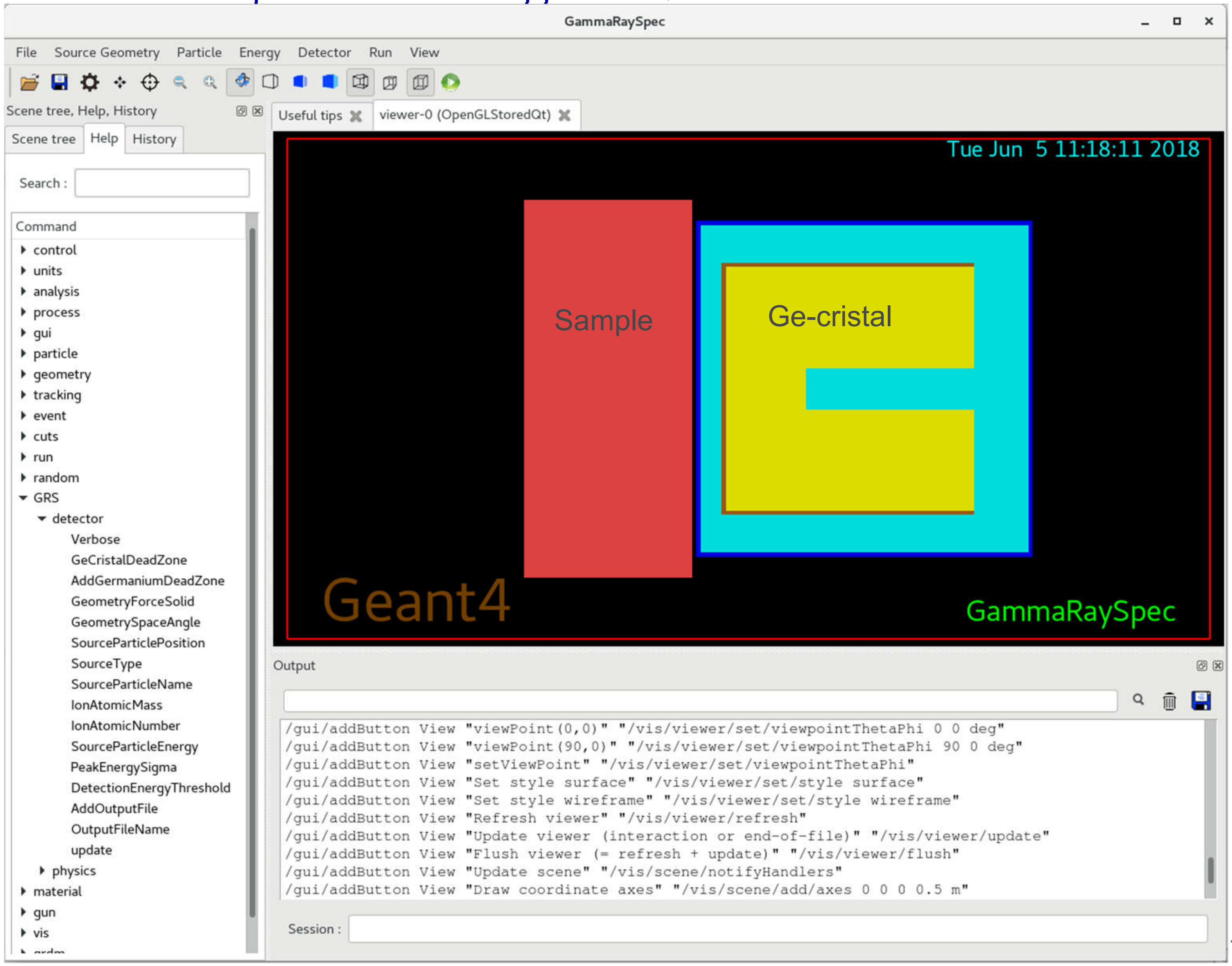
Typical macro file (input)

```
#=============================================================
# Geant4 - an Object-Oriented Toolkit for Simulation in HEP
#=============================================================
#
# Macro file to execute detector A in GammaRaySpec example
#
# Can be run in batch, without graphic
# or interactively: Idle> /control/execute detectorA.mac
#
# Nnumber of workers
#/run/numberOfWorkers 4                         # set nb of workers (in multi-threading mode)
#
# Verbose
/control/verbose 1
/run/verbose 1
/event/verbose 0
/tracking/verbose 0
/process/verbose 0
# Activate physics model
/GRS/physics/addPhysics emlowenergy             # set EM physics
/cuts/setLowEdge 250 eV                         # set range cuts
# Initialize kernel
/run/initialize
#
# Detector commands
/GRS/detector/Verbose 0                         # GRS detector verbosity
/GRS/detector/AddGermaniumDeadZone true         # set dead zone true (A) or false (B)
/GRS/detector/GeCristalDeadZone 1 mm            # set dead layer layer
/GRS/detector/SourceType Filter                 # set source type (Point Water Filter Soil)
/GRS/detector/SourceParticlePosition 0 0 1 mm   # set source position from the top of detector
/GRS/detector/SourceParticleName gamma          # set particle name
/GRS/detector/SourceParticleEnergy 1000 keV     # set particle energy
/GRS/detector/PeakEnergySigma 2.0 keV           # set peak energy sigma
/GRS/detector/DetectionEnergyThreshold 1.0 keV  # set detection threshold
/GRS/detector/AddOutputFile 1                   # add root output file
/GRS/detector/OutputFileName detector_AF_1000keV # set output file name
/GRS/detector/update                            # update geometry
#
# Print progress status
/run/printProgress 100000                       # Display event information at given frequency
#
# Run GO !
/run/beamOn 1000000                             # set number of events to be generated
```

GAMMA-RAY SPECTROMETRY BENCHMARK

- « `GammaRaySpec` » : γ-ray spectrometry dedicated benchmark

Screenshot of OpenGL viewer wrapped in Qt visualization driver

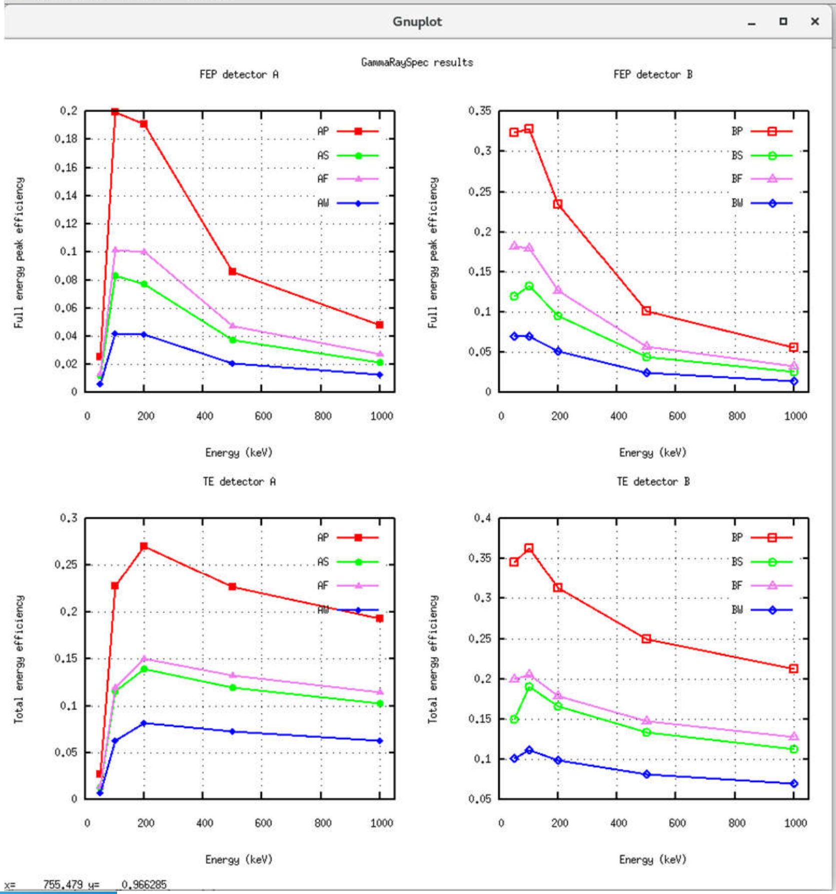


GAMMA-RAY SPECTROMETRY BENCHMARK

- « GammaRaySpec » : γ-ray spectrometry dedicated benchmark

Typical output for 8 cases configuration (detector A, B; and seletected energy: 50, 100, 200, 500, 1000 keV)

See in ${Dir}/GammaRaySpec/results



grs_user@localhost:~/GammaRaySpec/results

Detector B

Point source

Full energy peak efficiency					Total energy efficiency				
E(kev)	Events	Counts	FEP	U(%)	E(kev)	Events	Counts	TE	U(%)
0050.000	010000000	003235281	0.323528	0.055596	0050.000	010000000	003453322	0.345332	0.053812
0100.000	010000000	003283304	0.328330	0.055188	0100.000	010000000	003630534	0.363053	0.052483
0200.000	010000000	002340399	0.234040	0.065366	0200.000	010000000	003129380	0.312938	0.056529
0500.000	010000000	001015338	0.101534	0.099242	0500.000	010000000	002499002	0.249900	0.063258
1000.000	010000000	000565382	0.056538	0.132993	1000.000	010000000	002122524	0.212252	0.068639

Water source

Full energy peak efficiency					Total energy efficiency				
E(kev)	Events	Counts	FEP	U(%)	E(kev)	Events	Counts	TE	U(%)
0050.000	010000000	000703065	0.070306	0.119262	0050.000	010000000	001016555	0.101655	0.099182
0100.000	010000000	000695178	0.069518	0.119937	0100.000	010000000	001119930	0.111993	0.094494
0200.000	010000000	000513869	0.051387	0.139500	0200.000	010000000	000991666	0.099167	0.100419
0500.000	010000000	000249588	0.024959	0.200165	0500.000	010000000	000813437	0.081344	0.110876
1000.000	010000000	000148882	0.014888	0.259167	1000.000	010000000	000698058	0.069806	0.119689

Filter source

Full energy peak efficiency					Total energy efficiency				
E(kev)	Events	Counts	FEP	U(%)	E(kev)	Events	Counts	TE	U(%)
0050.000	010000000	001824958	0.182496	0.074024	0050.000	010000000	001999399	0.199940	0.070721
0100.000	010000000	001803304	0.180330	0.074467	0100.000	010000000	002062089	0.206209	0.069638
0200.000	010000000	001273505	0.127351	0.088613	0200.000	010000000	001793335	0.179334	0.074674
0500.000	010000000	000569975	0.056997	0.132456	0500.000	010000000	001479049	0.147905	0.082226
1000.000	010000000	000325465	0.032546	0.175286	1000.000	010000000	001282881	0.128288	0.088289

Soil source

Full energy peak efficiency					Total energy efficiency				
E(kev)	Events	Counts	FEP	U(%)	E(kev)	Events	Counts	TE	U(%)
0050.000	010000000	001197041	0.119704	0.091400	0050.000	010000000	001503976	0.150398	0.081542
0100.000	010000000	001326530	0.132653	0.086824	0100.000	010000000	001902432	0.190243	0.072501
0200.000	010000000	000951370	0.095137	0.102524	0200.000	010000000	001661449	0.166145	0.077581
0500.000	010000000	000439256	0.043926	0.150883	0500.000	010000000	001335451	0.133545	0.086534
1000.000	010000000	000255478	0.025548	0.197844	1000.000	010000000	001135747	0.113575	0.093834

[grs_user@localhost results]$

GAMMA-RAY SPECTROMETRY BENCHMARK

➔ **How to get « `GammaRaySpec` »**

(1) Download from this link : **ftp://ftp.cea.fr/incoming/y2k01/ICRM_GRS/GammaRaySpec.tar.gz**

(2) Get a virtual machine containing the latest version of Geant4 and the `GammaRaySpec` benchmark, in Scientific Linux system

➔ Operating with *Virtual Box* in Windows system

➔ The distribution contains the following software already fully installed for the user :
- Operating system : Scientific Linux 7, 64 bits version
- Geant4 version 10.4 with all sets of data files.
- Visualisation tools : Qt OpenGL and Raytracer
- Analysis tools : ROOT, gnuplot
- GammaRaySpec benchmark

➔ **Download from this link:** **ftp://ftp.cea.fr/incoming/y2k01/ICRM_GRS/G4-GRS-VM.tar.gz**

➔ **Detailed description in** ***`${DIR/}GammaRaySpec/README`*** **`file.`**

GAMMA-RAY SPECTROMETRY BENCHMARK

➔ How to start:

Copy and unzip the archive (GammaRaySpec.tar.gz) in your working repository

```
>> cd %HOMEPATH%
>> tar -xzvf GammaRaySpec.tar.gz
>> cd ./GammaRaySpec
>>
```

Compile the code (2 possible modes)

```
>> source $(G4build)/geant4make.sh    # Set the Geant4 environment
>>
```

(1) Compile using make

```
>> cd %HOMEPATH%/GammaRaySpec
>> make clean
>> make -j<number of threads>         # Run using executable created ${DIR}/GammaRaySpec
>>
```

(2) Compile using the CmakeList.txt

➔ Create a building directory and go inside, then execute cmake .. and make

```
>> cd %HOMEPATH%
>> mkdir Build_rep                    # Create a building directory
>> cd Build_rep
>> cmake -DGeant4_DIR="%HOMEPATH%\G4-install\lib\Geant4-G4VERSION"  "%HOMEPATH%\GammaRaySpec"
```

➔ In 'batch' mode

```
>> GammaRaySpec detectorA.mac         # Run using executable created ${DIR}/GammaRaySpec  in 'batch' mode
```

➔ In 'interactive' mode (with visualization)

```
>> GammaRaySpec
    Idle> control/execute detectorB.mac
    Idle> run/beamOn 5
```

Commissariat à l'énergie atomique et aux énergies alternatives
Institut List | CEA SACLAY NANO-INNOV | BAT. 861 – PC142
91191 Gif-sur-Yvette Cedex - FRANCE
www-list.cea.fr

Établissement public à caractère industriel et commercial | RCS Paris B 775 685 019