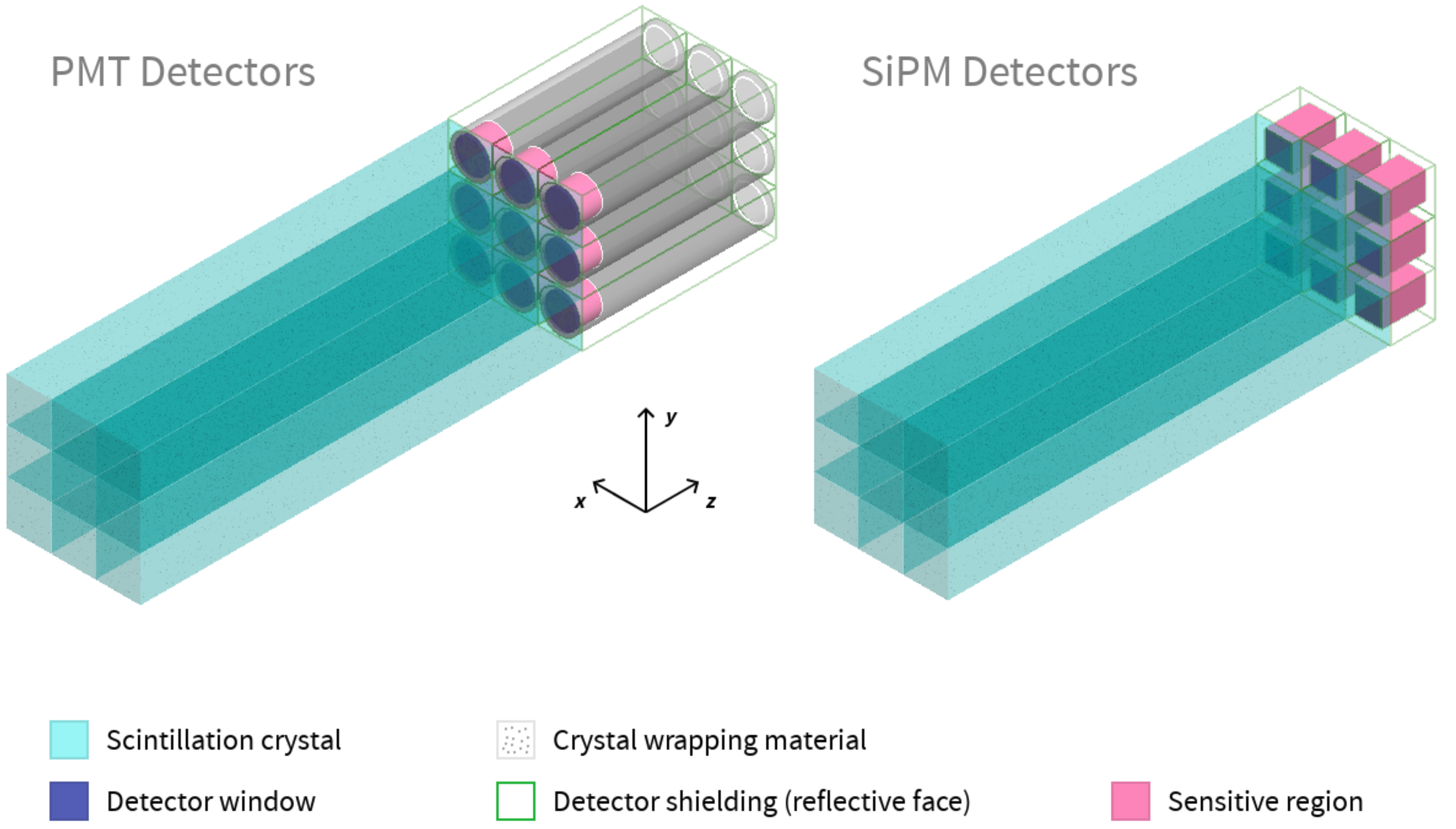
Glass Prototype Simulation

Repository contains Geant4-based program for optical simulations of the SciGlass prototype assembly. Detector geometry is represented by an NxN matrix of a repeated volumes. Each volume contains scintillation material (rectangular cuboid block) paired with the sensitive detector (PMT or MPPC). Please refer to the image below:



Left: PMT-based detector construction. Right: SiPM-setup

To reproduce above detector visualization setup, launch `glass` executable without command line parameters (in Geant4 interactive mode). Next, open the `vis-init.mac` visualization macro.

Installation

Program supports CMake build system. To successfully compile the project it is necessary to:

1. Have Geant4 and ROOT environment installed on the computer.
2. Draw scripts that plot output files require RootUtils and CanvasHelper shared libraries compiled and installed as prerequisites. Please refer to libraries' README for installation instructions.
3. After all the dependencies are satisfied, build can be carried out as follows:

```
# Navigate into the installation folder
mkdir -p ~/Development && cd ~/Development/

# Re-download program source
rm -rf ./glass-prototype
git clone https://github.com/JeffersonLab/glass-prototype

# Build the program externally
rm -rf ./glass-prototype-build
mkdir -p ./glass-prototype-build && cd ./glass-prototype-build
cmake -DGeant4_DIR=$G4LIB/Geant4-$G4DATA_VERSION/ ./../glass-prototype/
```

Macro Commands

Detector setup and the simulation output is controlled with custom commands implemented in corresponding messenger classes. Fundamental commands are outlined below.

Physics Options

Command	Description	Type
`/physics /selectList`	Set Geant4 physics list. Available options: `FTFP_BERT` (default), `QGSP_BERT` , `QGSC_BERT_EMZ` , ... Turn on optical physics for optical simulatinos by passing `withOptical` second parameter.	`String` , `String`

Assembly Parameters

Command	Description	Type

Command	Description	Type
`/detector/setCrystalSize`	Scintillation block size.	`3VectorAndUnit`
`/detector/setCrystalNumberX`	Number of the blocks in the matrix along x axis.	`Int`
`/detector/setCrystalNumberY`	Number of the blocks along y axis.	`Int`
`/detector/setWrapMaterial`	Crystal wrap material. `C10H8O4` is vm2000 (ESR), `G4_TEFLON` , `NONE`	`String`
`/detector/setWrapThickness`	Crystal wrap thickness. Default is 65 μm.	`DoubleAndUnit`

Sensitive Detector

- `/detector/setDetectorType` - type of the sensitive detector placed behind each crystal. Command accepts `string` type.

Available options for the sensitive detectors:

Parameter	PMT Window Diameter	Window Material	Cathode type
`R4125`	18 mm	borosilicate	bialkali
`R1828-01`	53 mm	borosilicate	bialkali
`R2059`	53 mm	quartz (fused silica)	bialkali
`R2257`	52 mm	borosilicate	multialkali

Parameter	MPPC Size	Window Material	Cathode type
`S13360-6025CS`	6x6 mm	silicone resin	silicon crystal
`S13360-6025CS-2x2`	12x12 mm	silicone resin	silicon crystal

Custom PMT-based and SiPM detectors can be added in the code via custom classes inherited from abstract `AbsPMT` and `AbsMPPC` interfaces.

Controling the Program Output

Following commands control the data in output ROOT files. Command parameters are of a boolean type.

Command	Description	Default
`/detector/usePrimitiveScorer`	Register deposited energy in the crystal volumes with Primitive Scorer	`true`
`/detector/useGlobalScoringMeshes`	Accumulate and visualize deposited energy around crystal matrix and sensitive detector (with Scoring Meshes)	`true`

Command	Description	Default
`/detector /useUnitVolumeScoringMeshes`	Divide crystal into segments (unit volumes) along z-azis. Calculate and visualize energy deposition in each unit volume. This is useful for heavy optical calculations (read below).	`true`
`/detector/saveEnergyWorldEscape`	Save total energy escaped the world. Also save first 1M particles escape the world (types, position, energies)	`true`
`/detector/saveTimeOfFlight`	Save time-of-flight information: local time (from beginning of OP track) and global time (from event start)	`false`

When carrying out an optical simulations with `/physics/addOptical` it is reasonable to turn the above commands off to reduce the computation wall time of the simulation.

User notes

This section is designed for the end users. It describes the process of running the simulation and obtaining the experimental results on the Computing Farm.

Logging to the Computing Farm

Program code supports the running the executable in the Interactive simulation mode. To forward the graphical output from the Computing Farm via ssh protocol, it is necessary to pass the `-Y` parameter to the `ssh` command:

```
ssh -Y <your-username>@login.jalb.org
ssh ifarm
```

After logging in to the Computing Farm it is necessary to source the environment with Geant4 and ROOT frameworks. Current code was tested on environment version 2.6 with Geant4 v.11, ROOT v.6.26:

```
source /site/12gev_phys/softenv.csh 2.6
```

Compilation of the Executable Program

First, it is necessary to compile the dependent ROT Next we check out this GitHub repository under the `Downloads` folder:

```
mkdir -p ~/Downloads && cd ~/Downloads/
rm -rf ./glass-prototype
git clone https://github.com/JeffersonLab/glass-prototype
```

In order to not interfere with the original program code we will create a dedicated build directory outside of the repository tree.

```
mkdir -p ./glass-prototype-bulid && cd ./glass-prototype-bulid
```

```
cmake -DGeant4_DIR=$G4LIB/Geant4-$G4DATA_VERSION/ ./../glass-prototype/
make -j`nproc`
```

Geting Familiar with the Program (GUI)

First we will demonstrate how to run the program in the interactive mode. Ensure the Geant4 environment is sourced, navigate into the build directory and run:

```
./glass
```

Once the user interface is loaded, click "Open" button to load demonstration macros. On the top left of the menu bar locate the "Open" button.

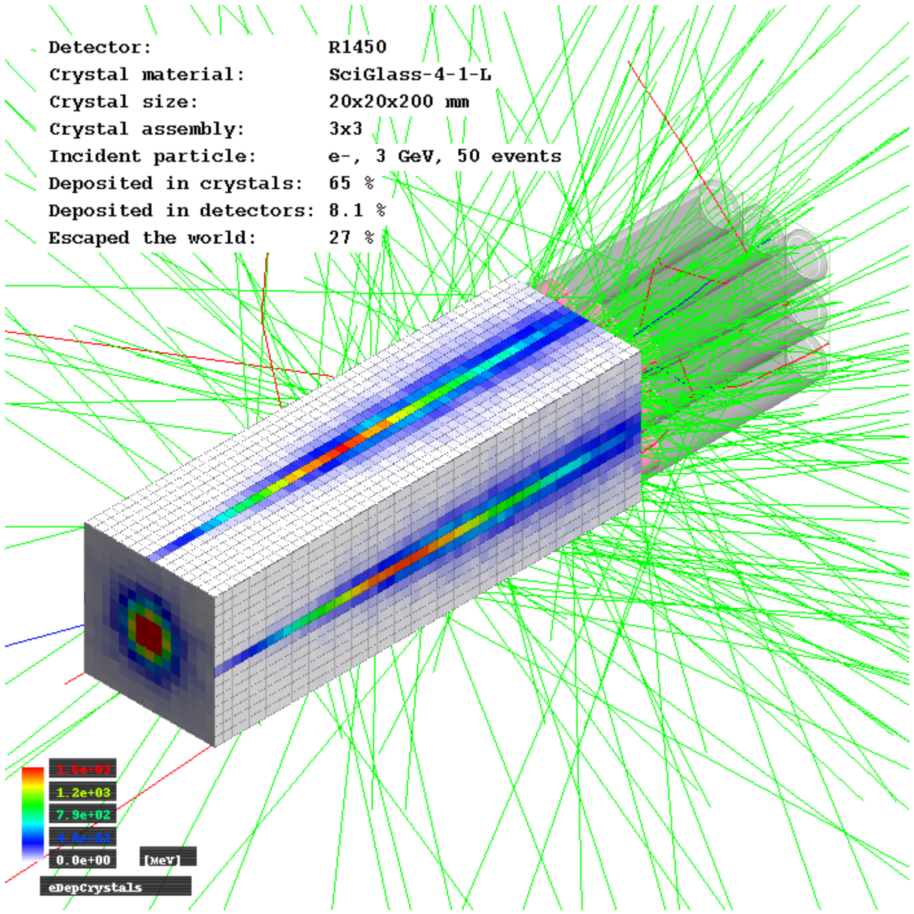
- Load `./macros/init-det.mac` file containing Physics List, Detector Geometry and Particle Source (GPS) information.
- Load `./macros/init-vis.mac` file containing Visualization information.
- Type `/run/beamOn 10` in the GUI command prompt on the bottom to run 10 events. Wait for the simulation to complete and show the result.

To visualize scoring meshes use following commands:

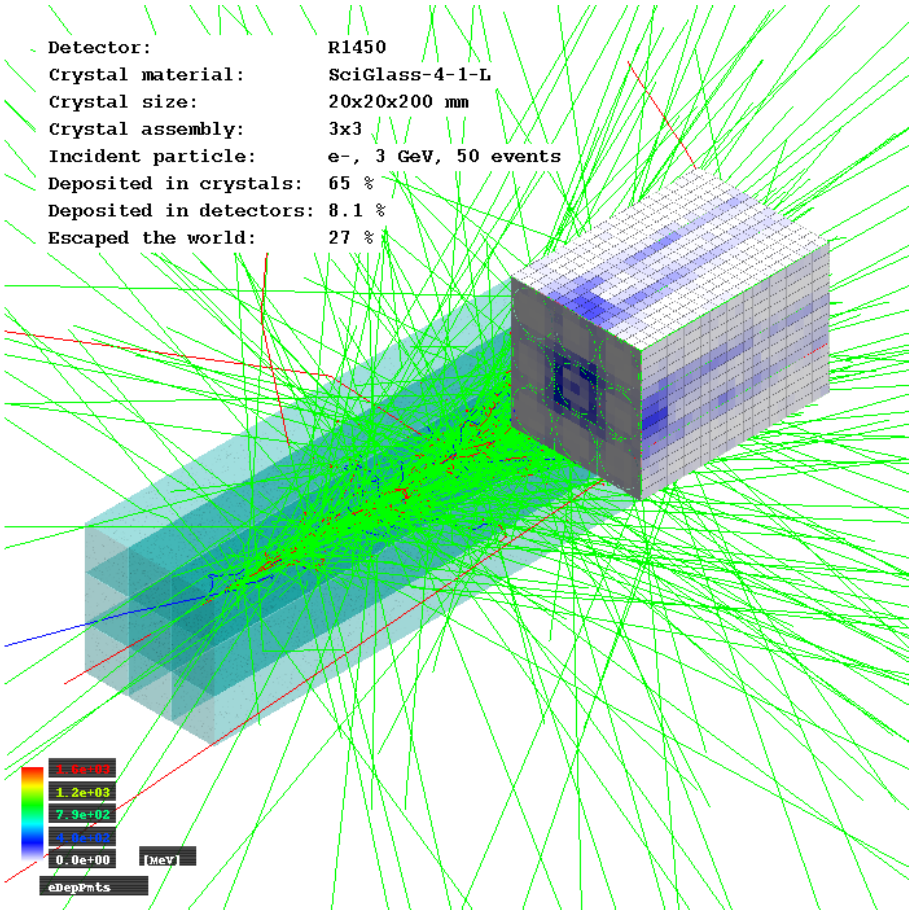
- `/score/drawProjection crystalsMesh eDepCrystals globalColorMap` - draw global mesh around the NxN crystal assembly.
- `/score/drawProjection pmtsMesh eDepPmts globalColorMap` - draw global mesh around all sensitive detectors.
- `/score/drawProjection crystalMesh1 eDepCrystal1 centerCrystalColorMap` - draw unit volume scoring meshes for a particular crystal number (replace 1 with N).

Originally every individual mesh has individual energy deposition color legend. Tweo custom color maps are defined in the program:

- `globalColorMap` has a maximum projection along the z axis across the global crystals mesh and sensitive detector mesh.
- `centerCrystalColorMap` - maximum cololor corresponds to the energy deposition in a unit volume in the center crystal with maximum energy deposition.



```
Detector:               R1450
Crystal material:       SciGlass-4-1-L
Crystal size:           20x20x200 mm
Crystal assembly:       3x3
Incident particle:      e-, 3 GeV, 50 events
Deposited in crystals:  65 %
Deposited in detectors: 8.1 %
Escaped the world:      27 %
```



```
Detector:               R1450
Crystal material:       SciGlass-4-1-L
Crystal size:           20x20x200 mm
Crystal assembly:       3x3
Incident particle:      e-, 3 GeV, 50 events
Deposited in crystals:  65 %
Deposited in detectors: 8.1 %
Escaped the world:      27 %
```

Visualizing energy deposition in the Detector Construction with Scoring Meshes

Extra custom visualization commands include:

- `/myvis/drawStats` - prints statistics in the top right corner.
- `/myvis/ogl/printPNG` and `/myvis/ogl/printEPS` save viewer image to hard drive.

Running Simulation in Batch Mode

Simulation will run in the batch mode if a macro file is passed to the `./glass` program as a parameter (default Geant4 approach).

```
./glass ./macros/<your-macro-file.mac>
```

In the batch mode the visualization driver should be set to `ASCIITree`. All other drivers instantiated with `/vis/open` may be switched off (`OGL`, `OGLIX` etc).

```
/vis/ASCIITree/verbose 13
/vis/drawTree
```

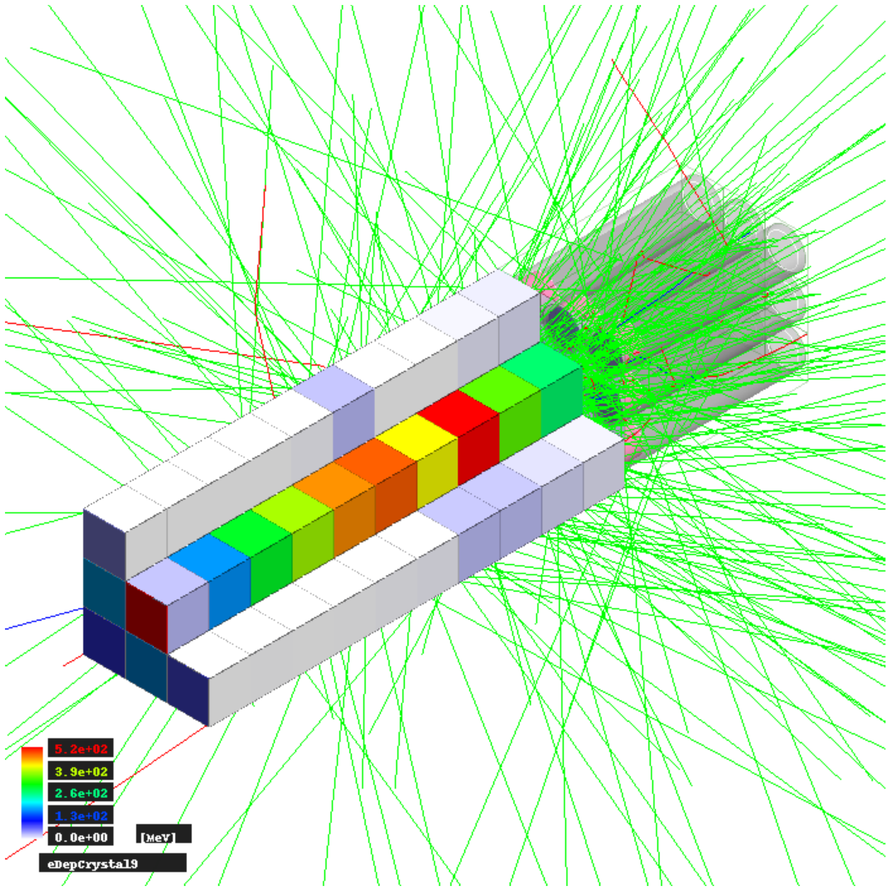
It is possible to create a single macro file that does two runs. An interactive run with low statistics followed by the batch run with high statistics. Corresponding macro file would be:

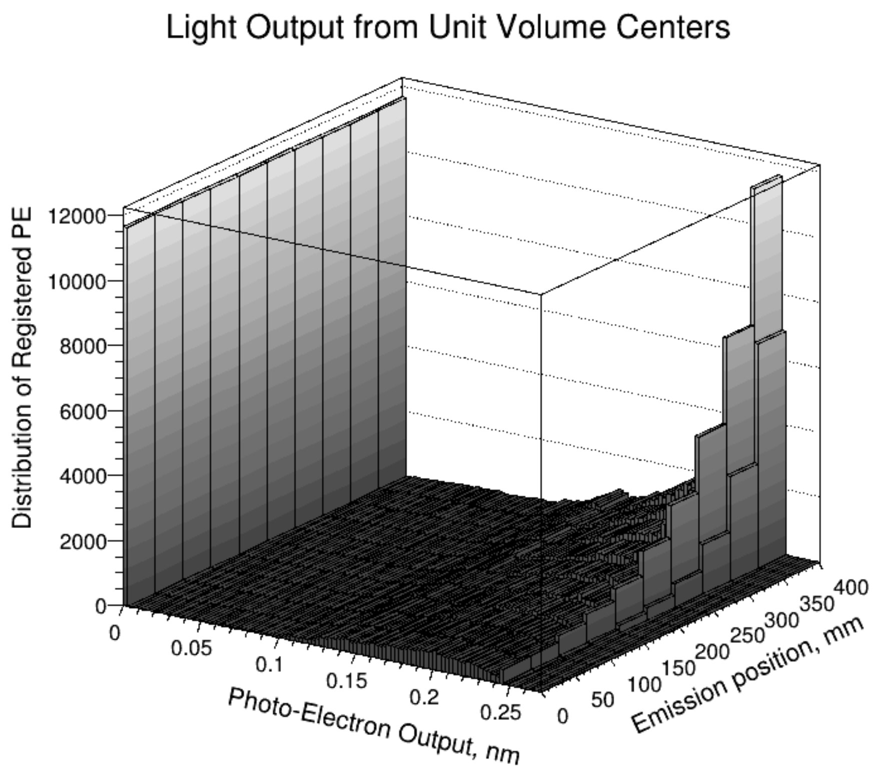
```
# Set up detector, and initialize the Run Manager
...
/run/initialize

# Set visualization driver
/vis/open OGLIX

# Run with low statistics
/run/beamOn 100

# Save images and do stuff
...

# Close visualization
/vis/disable

/vis/ASCIITree/verbose 13
/vis/drawTree

# Second run attempt
/run/initialize

# Run with high statistics
/run/beamOn 100000
```

Facilitating Optical Simulations

Optical simulations are highly resourceful. For instance, simulating one injection event (e-, ~5 GeV energy) on the computing farm in single-threaded mode takes about an hour. Geant4 needs to break down all the deposited energy into optical photons and calculate their trajectories.

To facilitate calculation of the optical resolution, following approach is proposed. Crystal volume is divided into a number of "unit volumes" along the longitudinal axis (refer to the image below). Energy deposition in each unit volume is saved. Next, instead of calculating each emitted photon trajectory, we register optical responces obtained from emittimg 1M photons from the centers of each unit volume. These are photo-electron distribution function obtained from a single photon emitted from the center of each volume. Essentially, for each particular unit volume these area distributiuon of the quantum efficiences (float number of photo-electrons) obtained at the sensitive detactor part.





Left: accumulating energy depositions in unit volumes. Right: optical responces from centers of each unit volume.

First iteration. Each crystal in the NxN assembly is divided into a number of unit volumes (currently hardcoced to 10). Energy deposition is calculated in each crystals' unit volume with the help of Scoring Meshes. To turn the above feature on set `/detector/useUnitVolumeScoringMeshes true` in the macro file. It is enough to have ~5000 events for each incident particle energy run. Please refer to the picture below that demonstrates accumulated energies in the scoring meshes:

Second iteration. For a 1x1 assembly (single block) we and emit a fixed number (currently 1M) of optical photons from the centers of each unit volume. The emission spectrum of the optical photons should match the one of the studied material. This is achieved as follows. Optical photon emission macro files for a particular material are generated via a custom `Materials::generateEmissionMacro(G4Material* material)` function. Next, this macro files are imported in the GPS settings.

Third iteration. Using a ROOT script `calcOpticalResolution` we merge data from first and second itrerations. Knowing the material scintillation yield and deposited energy in each unit volume, we calculate number of photons produced inside each unit volume. Next, we reconstruct cumulative light output from all unit volumes in the following manner. For each single emitted photon from each unit volume we utilize ROOT's `TH1::GetRandom()` function to obtain resulted PE number saved in the second iteration. Finally, optical responces from each photon emitted from each unit volume center are saved in "tree_pe" tree in original file with energy deposition in unit volumes.

```
./calcOpticalResolution <root-file-with-energy-deposition-in-unit-volumes> <directory-path-to files-with-pe-responces-from unit-volumes>
```

Alternatively, it is possible to pass a directory contatining multiple ROOT energy deposition files for a number of incident particle energies as a first parameter:

```
./calcOpticalResolution <root-file-with-energy-deposition-in-unit-volumes> <directory-path-to files-with-pe-responces-from unit-volumes>
```

Running Multiple Simulatinos with Slurm (Advanced)

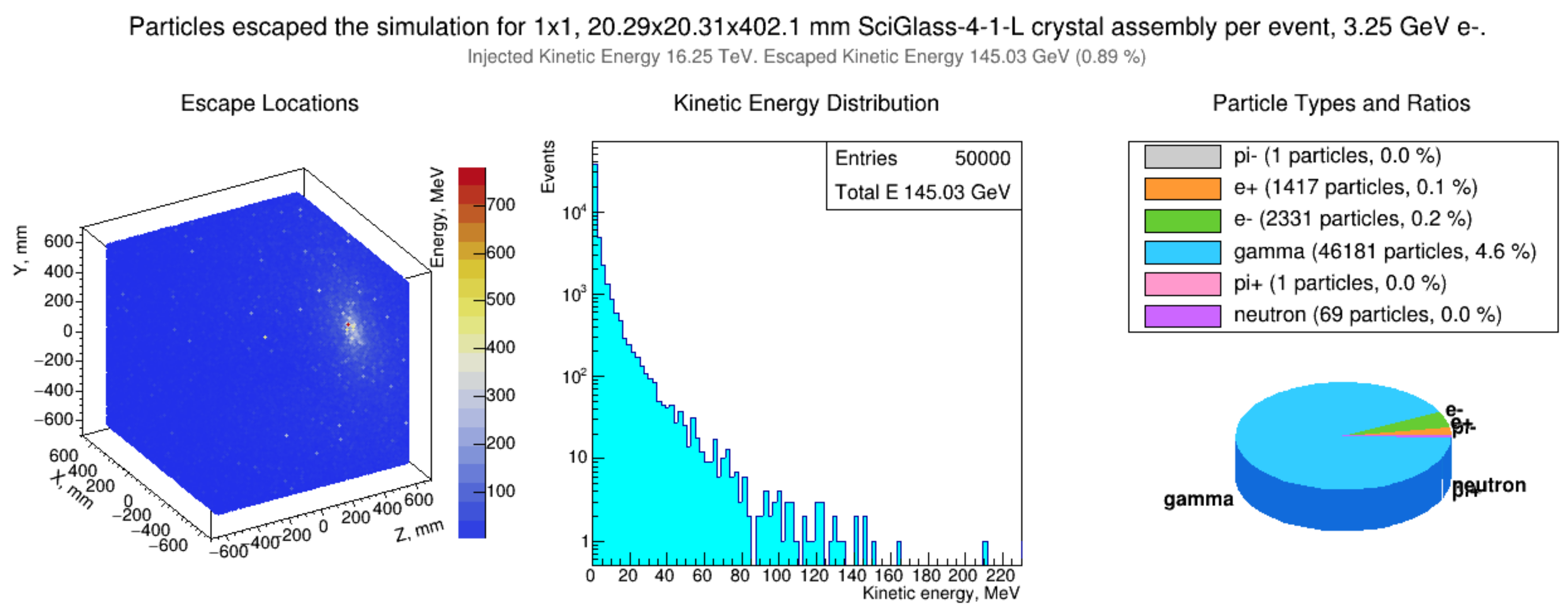
Slurm Workload Manager program on the Supercomputer Environment provides a functionality to schedule multiple Geant4 simulations at the same time. A single slurm script with arrays functionality can generate a series of macro files from a single template macro file. Then each generated macro file is executed in a separate shell process. Progress of the operation can be tracked in the Scientific Computing JLab interface.

Slurm script examples are located in the `./slurm` project folder. Few essential Slurm scripts are following:

- `./slurm/eres-optical.sh` - calculates energy depositions for a particular NxN matrix geometry. Also obtains energy depositions in unit volumes (optical resolution, iteration 1).
- `./slurm/eres-optical-uv-emission.sh` - obtains light output from photons emitted from the unit volume centers for a single block (optical resolution, iteration 2).

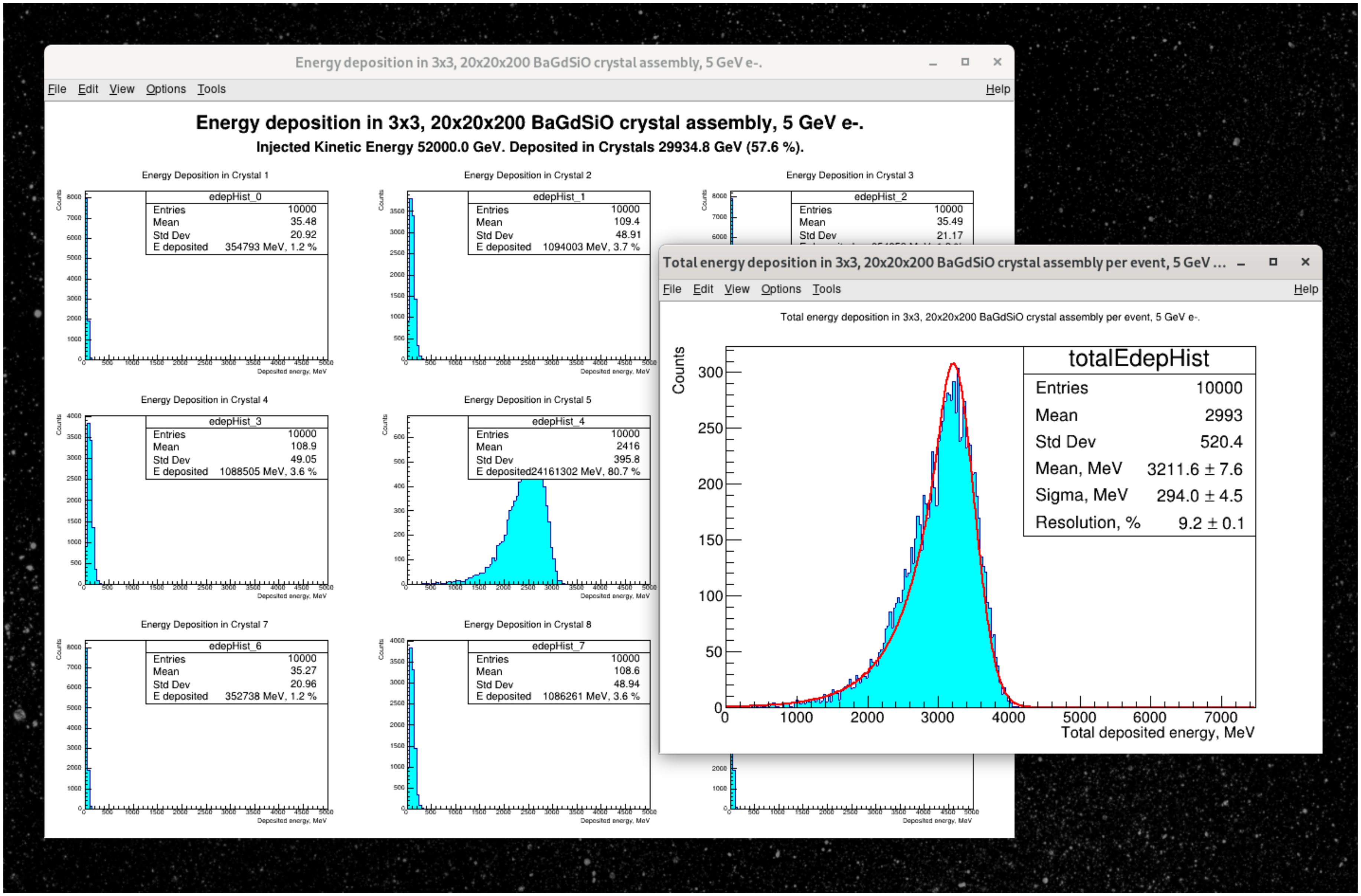
Analysis of the Output Files

By default, simulation produces series or ROOT files saved under the `./output` folder. ROOT scripts that plot simulatuion results are located in the `./draw` folder. Every plotting ROOT script is additionally compiled into ROOT-based executable (mostly for the ease of debugging).



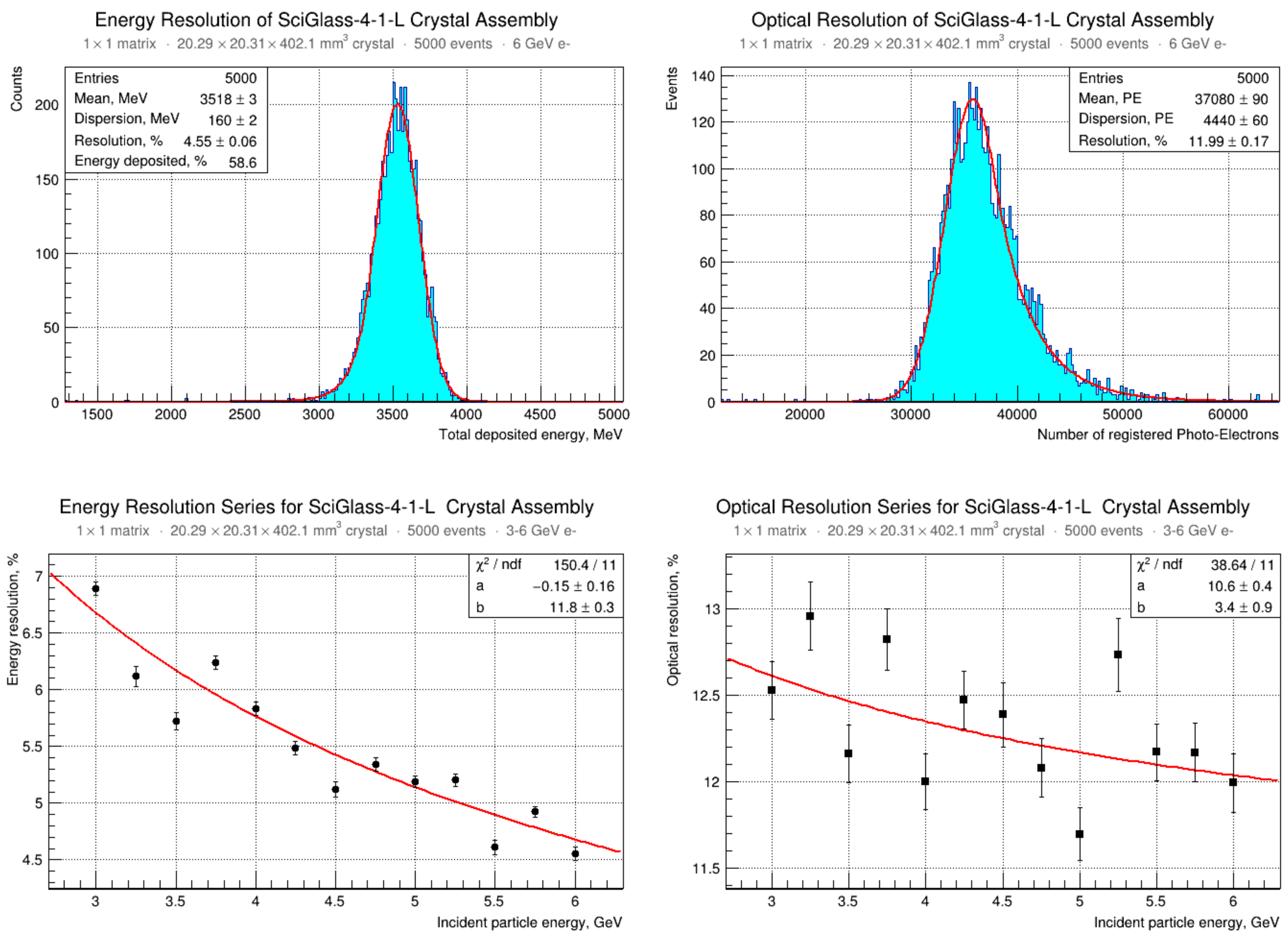
Particles escaping the Geant4 world.

Script `escapeParticles.cpp` visualizes the locations, energies and particle types that have escaped the Geant4 simulation world boundaries.


Energy deposition in 3x3, 20x20x200 BaGdSiO crystal assembly, 5 GeV e-.
File Edit View Options Tools Help
Energy deposition in 3x3, 20x20x200 BaGdSiO crystal assembly, 5 GeV e-.
Injected Kinetic Energy 52000.0 GeV. Deposited in Crystals 29934.8 GeV (57.6 %).
Energy Deposition in Crystal 1
edepHist_0
Entries 10000
Mean 35.48
Std Dev 20.92
E deposited 354793 MeV, 1.2 %
Energy Deposition in Crystal 2
edepHist_1
Entries 10000
Mean 109.4
Std Dev 48.91
E deposited 1094003 MeV, 3.7 %
Energy Deposition in Crystal 3
edepHist_2
Entries 10000
Mean 35.49
Std Dev 21.17
Energy Deposition in Crystal 4
edepHist_3
Entries 10000
Mean 108.9
Std Dev 49.05
E deposited 1088505 MeV, 3.6 %
Energy Deposition in Crystal 5
edepHist_4
Entries 10000
Mean 2416
Std Dev 395.8
E deposited24161302 MeV, 80.7 %
Energy Deposition in Crystal 7
edepHist_6
Entries 10000
Mean 35.27
Std Dev 20.96
E deposited 352738 MeV, 1.2 %
Energy Deposition in Crystal 8
edepHist_7
Entries 10000
Mean 108.6
Std Dev 48.94
E deposited 1086261 MeV, 3.6 %
Counts
Deposited energy, MeV
Total energy deposition in 3x3, 20x20x200 BaGdSiO crystal assembly per event, 5 GeV ...
Total energy deposition in 3x3, 20x20x200 BaGdSiO crystal assembly per event, 5 GeV e-.
totalEdepHist
Entries 10000
Mean 2993
Std Dev 520.4
Mean, MeV 3211.6 ± 7.6
Sigma, MeV 294.0 ± 4.5
Resolution, % 9.2 ± 0.1
Counts
Total deposited energy, MeV

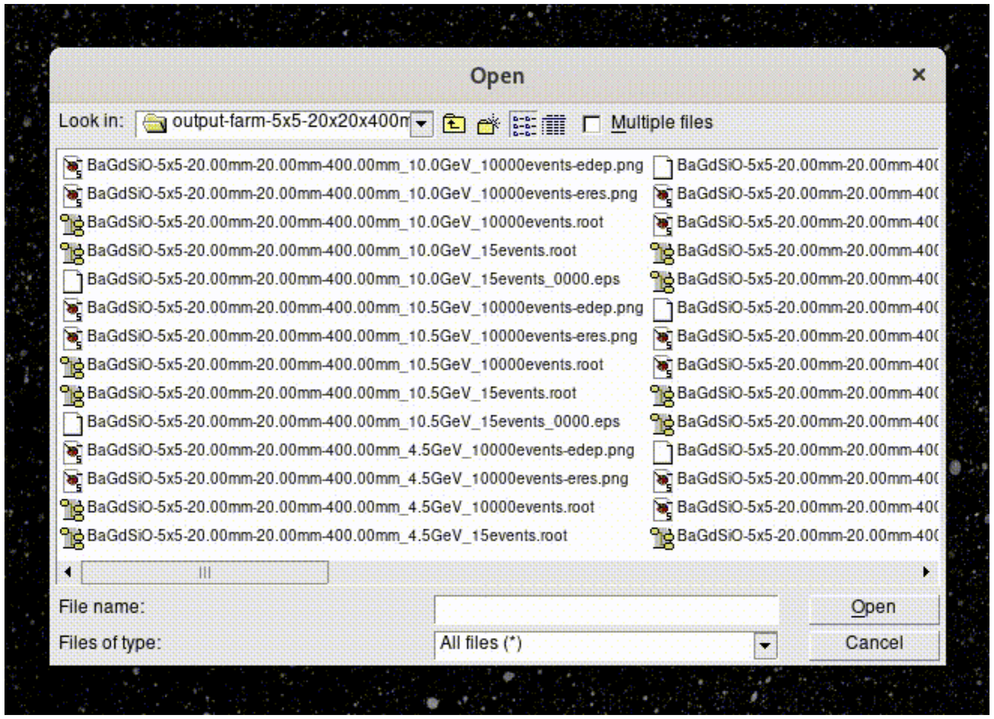
Particles escaping the Geant4 world.

Macro `resolution.cpp` plots energy resolutions for series of output files. If output files were processed to contain the optical information, optical resolutions will be calculated and plotted as well. Regular and reversed Crystal Ball functions implemented in the `RootUtils` library are used to fit the energy and optical resolution distributions. Resolutions are calculated upon the first and second Crystal Ball function momenta (formula obtained in Wolfram Mathematica).



Energy and optical resolutions for a single ROOT file and series of files for different iuncident particle energies.

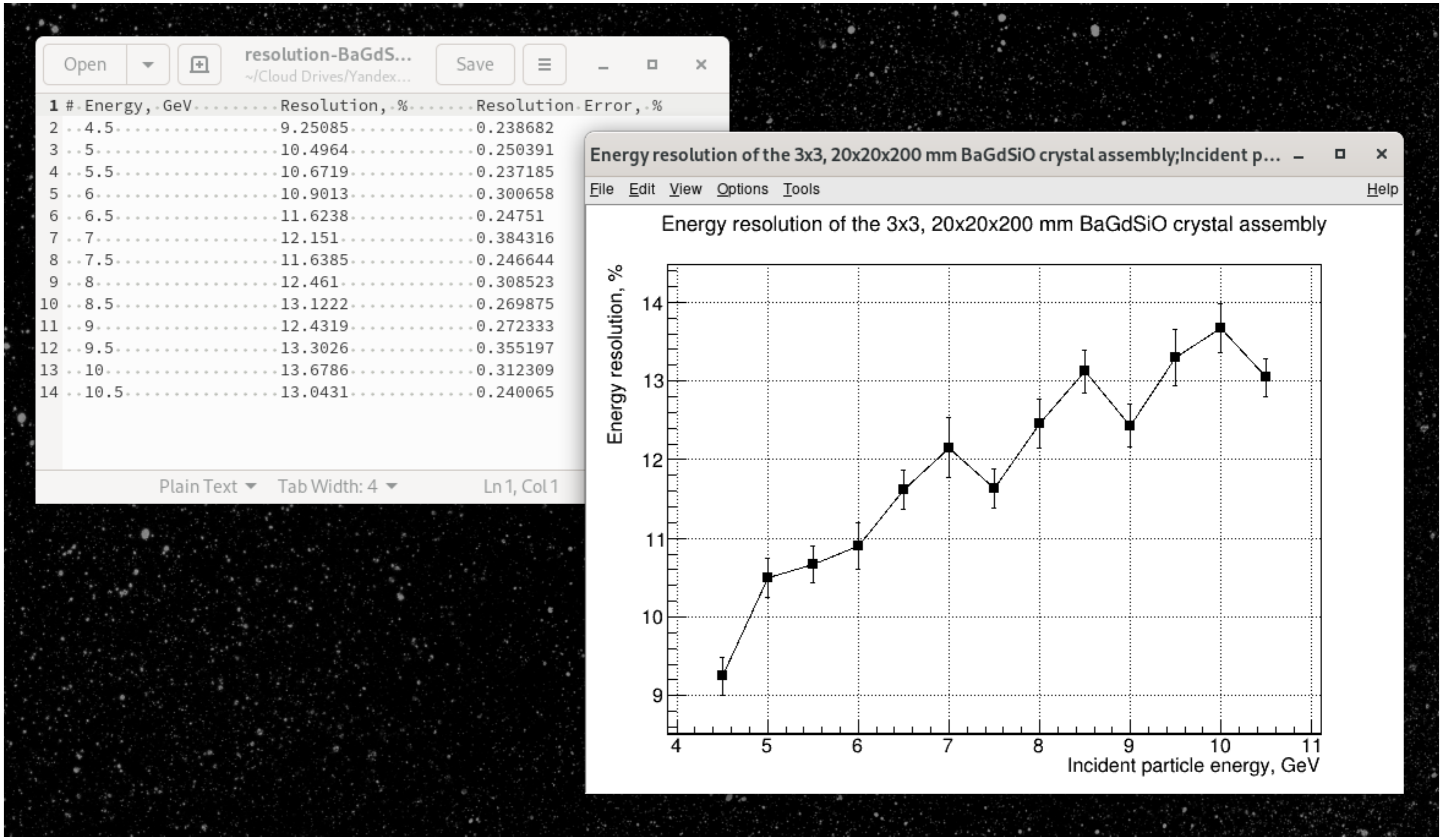
Locate the `.root` output files that correspond to the simulations of the same detector geometry for different energies. It is possible to select multiple files in ROOT dialog box by checking the `Multiple files` checkbox on the top right of the dialog box and holding the `CTRL` or `CMD` key on the keyboard:



```
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_10.0GeV_10000events-edep.png   BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_10.0GeV_10000events-eres.png   BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_10.0GeV_10000events.root       BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_10.0GeV_15events.root          BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_10.0GeV_15events_0000.eps      BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_10.5GeV_10000events-edep.png   BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_10.5GeV_10000events-eres.png   BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_10.5GeV_10000events.root       BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_10.5GeV_15events.root          BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_10.5GeV_15events_0000.eps      BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_4.5GeV_10000events-edep.png    BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_4.5GeV_10000events-eres.png    BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_4.5GeV_10000events.root        BaGdSiO-5x5-20.00mm-20.00mm-40
BaGdSiO-5x5-20.00mm-20.00mm-400.00mm_4.5GeV_15events.root           BaGdSiO-5x5-20.00mm-20.00mm-40
```

Selecting multiple files with ROOT File dialog.

The ASCII data file with energy resolution values for every energy of the incident particle is created. Additionally the energy resolution graph is plotted.


Open
resolution-BaGdS...
~/Cloud Drives/Yandex...
Save
Plain Text
Tab Width: 4
Ln 1, Col 1
Energy resolution of the 3x3, 20x20x200 mm BaGdSiO crystal assembly;Incident p...
File Edit View Options Tools Help
Energy resolution of the 3x3, 20x20x200 mm BaGdSiO crystal assembly
Energy resolution, %
Incident particle energy, GeV
9
10
11
12
13
14
4
5
6
7
8
9
10
11

```
# Energy, GeV          Resolution, %         Resolution Error, %
  4.5                  9.25085               0.238682
  5                    10.4964               0.250391
  5.5                  10.6719               0.237185
  6                    10.9013               0.300658
  6.5                  11.6238               0.24751
  7                    12.151                0.384316
  7.5                  11.6385               0.246644
  8                    12.461                0.308523
  8.5                  13.1222               0.269875
  9                    12.4319               0.272333
  9.5                  13.3026               0.355197
  10                   13.6786               0.312309
  10.5                 13.0431               0.240065
```

Energy resolution plot for multiple incident particle energies.

All above graphs are automatically saved in the same folder where the input data files are located.

Plotting Energy Resolution for Multiple Geometries

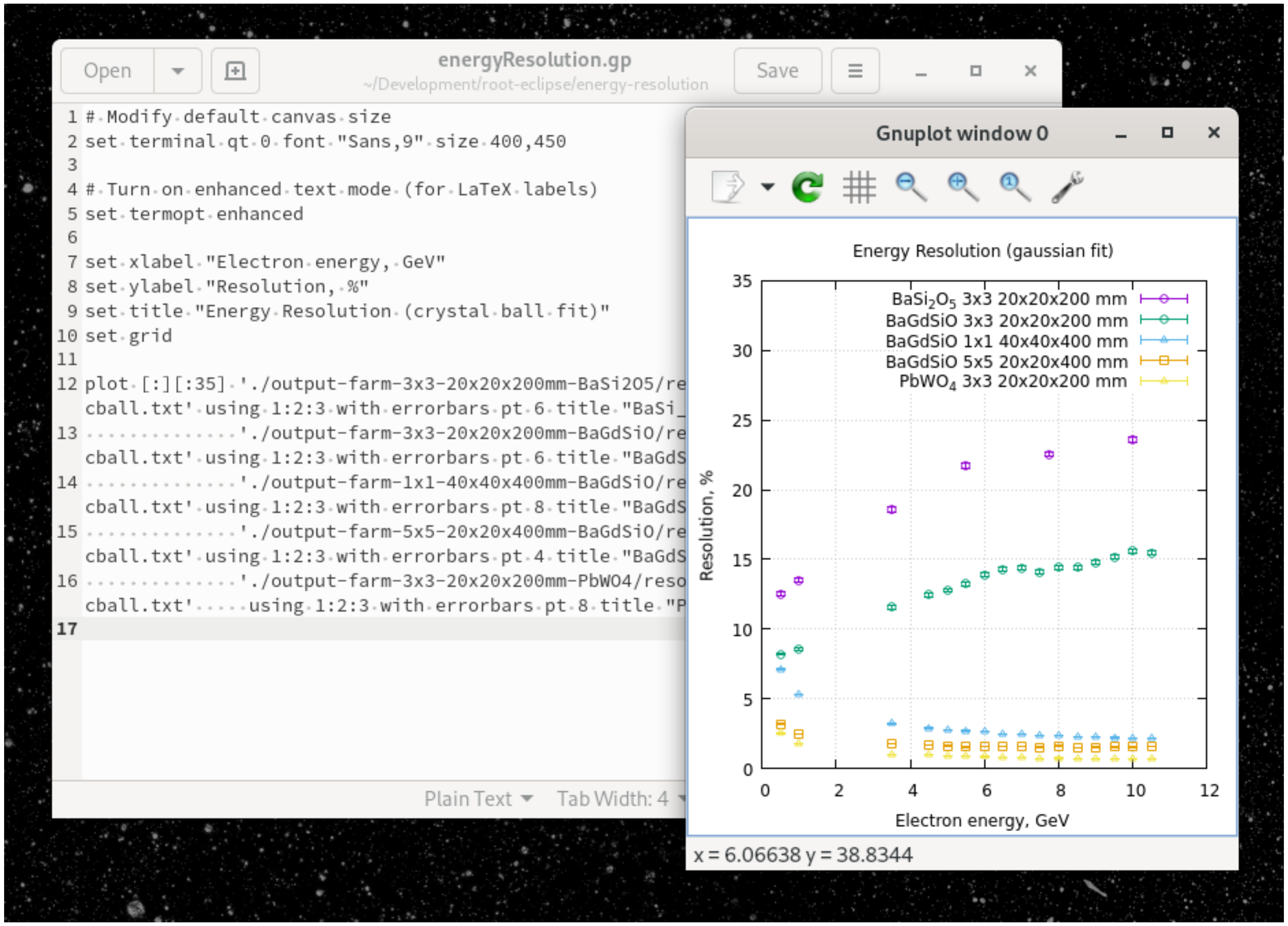
A special Gnuplot script `./glass_prototype/draw/energy-resolution/energyResolution.gp` can plot series of the energy resolutions for various detector geometries. open the script with a text editor of your choice:

```
nano ./draw/energy-resolution/energyResolution.gp
```

Update the data file names in the Gnuplot script to correspond to the energy resolution data points output by the `energyResolution.cpp` script. Then the energy resolution series can can be plotted with the following command:

```
gnuplot -p -c ./draw/energy-resolution/energyResolution.gp
```

A demonstration output of the Gnuplot script is presented on the picture below:



```
# Modify default canvas size
set terminal qt 0 font "Sans,9" size 400,450

# Turn on enhanced text mode (for LaTeX labels)
set termopt enhanced

set xlabel "Electron energy, GeV"
set ylabel "Resolution, %"
set title "Energy Resolution (crystal ball fit)"
set grid

plot [:][:35] './output-farm-3x3-20x20x200mm-BaSi2O5/re
cball.txt' using 1:2:3 with errorbars pt 6 title "BaSi_
              './output-farm-3x3-20x20x200mm-BaGdSiO/re
cball.txt' using 1:2:3 with errorbars pt 6 title "BaGdS
              './output-farm-1x1-40x40x400mm-BaGdSiO/re
cball.txt' using 1:2:3 with errorbars pt 8 title "BaGdS
              './output-farm-5x5-20x20x400mm-BaGdSiO/re
cball.txt' using 1:2:3 with errorbars pt 4 title "BaGdS
              './output-farm-3x3-20x20x200mm-PbWO4/reso
cball.txt'     using 1:2:3 with errorbars pt 8 title "P
```

x = 6.06638 y = 38.8344

Series of Energy resolutions for multiple detector geometries.

Copying Results to a Local Computer

In order to copy the results from the Computing Farm to the local computer the `scp` command can be used. On Windows one must install the PuTTY software to execute the below command. Native Terminal applications can be used on MacOS and Linux. Use following command to copy the Geant4 `output` folder to the local machine. Replace `<your-username>` with your JLab username:

```
scp -rO <your-username>@login.jlab.org:/home/<your-username>/Development/glass-prototype-build/output ~/
```

In order to copy a single file, remove the `-r` flag from the above command.

Notes from the Former Developers

For Hall C DVCS (DVCS_evt_gen/), see https://wiki.jlab.org/cuawiki/images/f/fa/User_Guide.pdf for a short description on how to run on JLab/ifarm