LUXE GEANT4 Simulation Output

Oleksandr Borysov

LUXE S&A Meeting
March 30, 2020

GEANT4 Simulation code

- Typical Geant4 application.
- Can be configured with macro commands file.

Repository:

https://stash.desy.de/projects/LXSIM/repos/lxsim/browse

Environment on naf (bash):

```
. /cvmfs/sft.cern.ch/lcg/releases/LCG_97/Geant4/10.06.p01/x86_64-centos7-gcc8-opt/Geant4-env.sh
. /cvmfs/sft.cern.ch/lcg/releases/LCG_97/Geant4/10.06.p01/x86_64-centos7-gcc8-opt/bin/geant4.sh
alias cmake="/cvmfs/sft.cern.ch/lcg/releases/LCG_97/CMake/3.14.3/x86_64-centos7-gcc8-opt/bin/cmake"
```

```
cmake ./
make
```

```
./lxbeamsim  luxe_e_gamma.mac  1
```

1 — Number of parallel threads

Runs simulation as configured in luxe_e_gamma.mac

```
./lxbeamsim
```

Usually starts qt GUI and then command
/control/execute vis_ev_e_v1.mac
opens geometry viewer/browser

Configuration parameters

Type of primary particles:

- MC from txt file with a specific for IPStrong format, header ignored;
- Gaussian beam with given σx, σy (fixed emittance) and arbitrary initial z position (distance to IP);
- Monoenergetic beam of a given particles;
- Arbitrary energy spectrum from the file with (E, N) pairs;
- Arbitrary initial position (can be combined with spectrum settings);

Output:

- Collection of particles crossing the surface of a given (arbitrary) physical volumes;
- Energy deposited in a given (arbitrary) volume, thought virtual segmentation is predefined. Each cell has a list of particles with individual contributions.
- Trajectories of all particles;
- Position and segmentation of sensitive volumes, those which configured to record deposited energy.
- Histograms (e.g. number of primary simulated, primary particle distribution in phase space x, px);
- GDML geometry file;

Configuration parameters

Geometry and physics list settings:

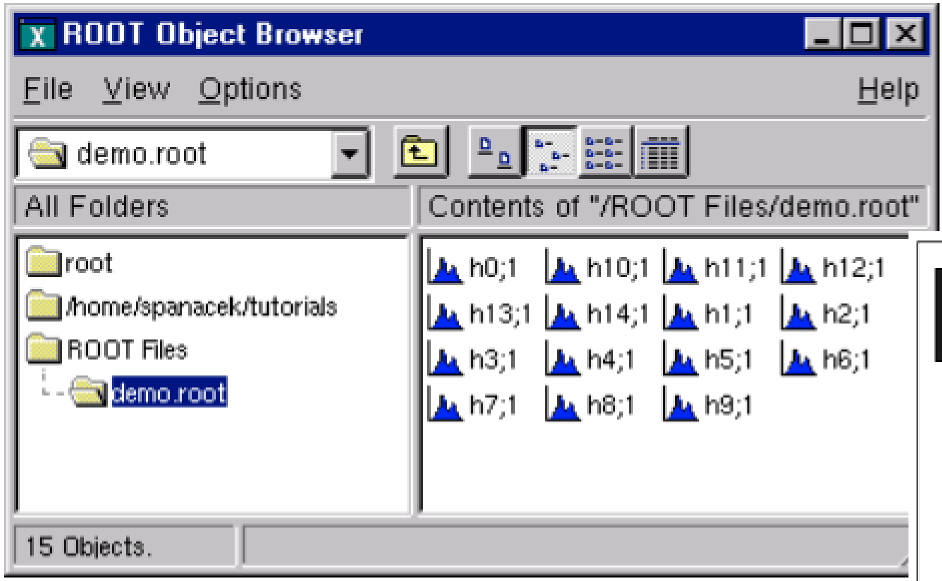
- Quite few and not very useful for typical simulation;
- Bremsstrahlung target type (plane wire), material, position, size in x, y, z;
- World size, material;
- Electromagnetic physics list (options 0-4);
- Production range cut for secondary particles.

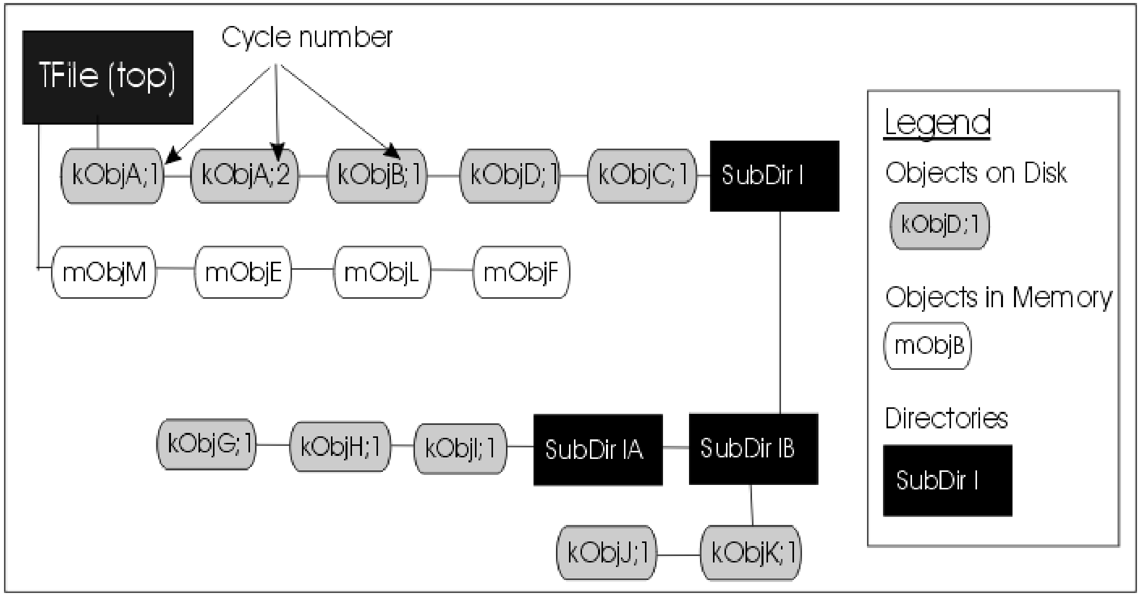
Visualization attributes:

- Tracks, background color, etc.
- Volumes visibility and colors (/control/execute vis_lx_color.mac)

Root file

A ROOT file is like a UNIX file directory. It can contain directories and objects organized in unlimited number of levels. It also is stored in machine independent format (ASCII, IEEE floating point, Big Endian byte ordering).





The structure of TFile

GEANT4 simulation output

GEANT4 simulation output:

- File with trees (TTree) and histograms:
  - In case of multithread simulation run, both trees and histograms are merged and saved in single a root file;
  - The name is specified in configuration macro file.

There are 5 trees:

- Tracks contains information about particles crossing the surface of physical volume
- Hits contains energy deposition in sensitive volumes
- HitTtracks information about tracks which deposited energy in sensitive detectors
- DetSettings position, segmentation, mass, material of sensitive volumes
- Trajectory trajectories of all particles step by step

Output configuration

```
CreateNtuple("Tracks", "Tracks hitting volumes marked for track interception");
CreateNtupleDColumn(1, "E");
CreateNtupleDColumn(1, "x");
CreateNtupleDColumn(1, "y");
CreateNtupleDColumn(1, "z");
CreateNtupleDColumn(1, "t");
CreateNtupleDColumn(1, "vtxx");
CreateNtupleDColumn(1, "vtxy");
CreateNtupleDColumn(1, "vtxz");
CreateNtupleDColumn(1, "px");
CreateNtupleDColumn(1, "py");
CreateNtupleDColumn(1, "pz");
CreateNtupleDColumn(1, "theta");
CreateNtupleDColumn(1, "phi");
CreateNtupleIColumn(1, "pdg");
CreateNtupleIColumn(1, "physproc");
umn(1, "detid");
umn(1, "xlocal");
umn(1, "ylocal");
umn(1, "zlocal");
umn(1, "eventid");
umn(1, "trackid");
umn(1, "weight");
```

```
/luxe/run/dump_geometry  false

/luxe/run/add_intercept_volume  OpppDetContainer:1000
/luxe/run/add_intercept_volume  ComptonDetContainer:2000
/luxe/run/add_intercept_volume  GammaCalo:3000
/luxe/run/add_intercept_volume  Shielding:4000

/run/initialize

/luxe/run/add_sensitive_volume  OpppTracker:1000:1:0
/luxe/run/add_sensitive_volume  ComptonTracker:2000:1:0
/luxe/run/add_sensitive_volume  GammaCalo:3000:1:0
```

```
->CreateNtuple("DetSettings", "Sensitive detector settings");
->CreateNtupleIColumn(4, "detid");
->CreateNtupleSColumn(4, "det_name");
->CreateNtupleIColumn(4, "layerid");
->CreateNtupleIColumn(4, "n_cell_x");
->CreateNtupleIColumn(4, "n_cell_y");
->CreateNtupleDColumn(4, "size_x");
->CreateNtupleDColumn(4, "size_y");
->CreateNtupleDColumn(4, "size_z");
->CreateNtupleDColumn(4, "translation_x");
->CreateNtupleDColumn(4, "translation_y");
->CreateNtupleDColumn(4, "translation_z");
->CreateNtupleDColumn(4, "e_phi");
->CreateNtupleDColumn(4, "e_theta");
->CreateNtupleDColumn(4, "e_psi");
->FinishNtuple(4);
```

Output of hits and tracks

```
CreateNtuple("Hits", "Hits in sensitive detectors");
CreateNtupleIColumn(2, "eventid");
CreateNtupleIColumn(2, "detid");
CreateNtupleIColumn(2, "layerid");
CreateNtupleIColumn(2, "cellx");
CreateNtupleIColumn(2, "celly");
CreateNtupleDColumn(2, "edep");
CreateNtupleIColumn(2, "hitid");
CreateNtupleIColumn(2, "track_list", fvHitTrackList);
CreateNtupleDColumn(2, "trackx", ftrackx);
CreateNtupleDColumn(2, "tracky", ftracky);
CreateNtupleDColumn(2, "trackz", ftrackz);
CreateNtupleDColumn(2, "trackt", ftrackt)
CreateNtupleDColumn(2, "trackedep", ftra
CreateNtupleDColumn(2, "weight");
FinishNtuple(2);
```

Each hit is one entry;

eventid links them to the primary particle.

```
>CreateNtuple("HitTracks", "Tracks which produced hits in sensitive detectors");
>CreateNtupleIColumn(3, "eventid");
>CreateNtupleIColumn(3, "trackid", fvTracks);
>CreateNtupleDColumn(3, "vtxx", fVtxx);
>CreateNtupleDColumn(3, "vtxy", fVtxy);
>CreateNtupleDColumn(3, "vtxz", fVtxz);
>CreateNtupleDColumn(3, "px", fPx);
>CreateNtupleDColumn(3, "py", fPy);
Column(3, "pz", fPz);
Column(3, "E", fE);
Column(3, "pdg", fPDG);
Column(3, "pproc", fPhysProc);
Column(3, "ptid", fPTId);
Column(3, "weight");
3);
```

```
CreateNtuple("DetSettings", "Sensitive detector settings");
CreateNtupleIColumn(4, "detid");
CreateNtupleSColumn(4, "det_name");
CreateNtupleIColumn(4, "layerid");
CreateNtupleIColumn(4, "n_cell_x");
CreateNtupleIColumn(4, "n_cell_y");
CreateNtupleDColumn(4, "size_x");
CreateNtupleDColumn(4, "size_y");
CreateNtupleDColumn(4, "size_z");
CreateNtupleDColumn(4, "translation_x");
CreateNtupleDColumn(4, "translation_y");
CreateNtupleDColumn(4, "translation_z");
CreateNtupleDColumn(4, "e_phi");
CreateNtupleDColumn(4, "e_theta");
CreateNtupleDColumn(4, "e_psi");
CreateNtupleDColumn(4, "mass");
CreateNtupleSColumn(4, "material");
CreateNtupleDColumn(4, "density");
```

ID of parent track,
For primary contains MC_ID

MC weight with scaling possibility

```
#------------------------------------------------------------------------------------------------------------------------
#  Final state particles.
#------------------------------------------------------------------------------------------------------------------------
#  First interacting species: electron     Second interacting species: laser
#  First initial particle energy = 16.5000 +/- .16 GeV,  Sigma_xyz = 5.00 5.00 24.00  microns,  Emit_xy = 1.401.40 mm mrad
#  Laser Intensity = 113.18 x 10^18 W/cm^2,  Wavelength = 800.00 nm,  pulse length = 25.00 fs,  spot size = 28.27 micron^2
#  Pulse peak xi =5.1769, Pulse peak chi =.9169, Misalignment = .0000 microns
#------------------------------------------------------------------------------------------------------------------------
#    E (GeV)        x (um)         y (um)        z(um)          beta_x                 beta_y                  beta_z              PDG_NUM    MP_Wgt    MP_ID
#------------------------------------------------------------------------------------------------------------------------
   16.509726     -10.304388     10.886293     63.219523    0.1444821543661777E-04 -0.1423474399418547E-04  0.99999999931531640107     11     750000.      1
   16.447808       1.9583259     2.7707411     55.681578    0.1518019409418836E-04 -0.1219624261428225E-05  0.99999999940142977493     11     750000.      2
   16.493717      -9.4561718    -4.7788727     36.038030    0.1591573848410204E-04 -0.6042725754170378E-05  0.99999999937516291547     11     750000.      3
   16.491767       9.2804767     2.0283782     87.417279    0.6121435756231020E-05 -0.8145950469357662E-05  0.99999999946804781186     11     750000.      4
   16.493315      -2.2033272    -3.1911421     26.844245    0.6608088942154653E-05 -0.3288602050925281E-05  0.99999999949281126753     11     750000.      5
```

Detector Settings Tree

Settings for the sensitive detectors

Translation from local detector element to global reference frame:

```
TRotation drt;
drt.SetXEulerAngles(phi, theta, psi);

TVector3 trns(translation.x, translation.y, translation.z);
xl = (cellx+0.5) * sizex / static_cast<double>(ncellx) - 0.5*sizex;
yl = (celly+0.5) * sizey / static_cast<double>(ncelly) - 0.5*sizey;
zl = 0.5*sizez;
TVector3 hitl(xl, yl, zl);

hitg = drt * hitl + trns;
```

Mass, material and density for dose estimation and test.

```
CreateNtuple("DetSettings", "Sensitive detector settings");
CreateNtupleIColumn(4, "detid");
CreateNtupleSColumn(4, "det_name");
CreateNtupleIColumn(4, "layerid");
CreateNtupleIColumn(4, "n_cell_x");
CreateNtupleIColumn(4, "n_cell_y");
CreateNtupleDColumn(4, "size_x");
CreateNtupleDColumn(4, "size_y");
CreateNtupleDColumn(4, "size_z");
CreateNtupleDColumn(4, "translation_x");
CreateNtupleDColumn(4, "translation_y");
CreateNtupleDColumn(4, "translation_z");
CreateNtupleDColumn(4, "e_phi");
CreateNtupleDColumn(4, "e_theta");
CreateNtupleDColumn(4, "e_psi");
CreateNtupleDColumn(4, "mass");
CreateNtupleSColumn(4, "material");
CreateNtupleDColumn(4, "density");
```

```
root [4] DetSettings->Scan("detid:det_name:layerid:n_cell_x:n_cell_y:size_x:size_y:mass:material:density:translation_x:translation_y:translation_z")
```

Row	detid	det_name	layerid	n_cell_x	n_cell_y	size_x	size_y	mass	material	density	translati	translati	translati
0	3000	LysoCal	0	300	25	300	50	2.25	LANEX	7500	160	0	10890
1	3001	LysoCal	0	300	25	300	50	2.25	LANEX	7500	-160	0	10890
2	4000	LeadGlass	0	1	1	38	38	2.508228	LG_TF1	3860	8.512e-15	-139.016	13279.992
3	4001	LeadGlass	0	1	1	38	38	2.508228	LG_TF1	3860	98.299156	-98.29915	13279.992
4	4002	LeadGlass	0	1	1	38	38	2.508228	LG_TF1	3860	139.016	0	13279.992
5	4003	LeadGlass	0	1	1	38	38	2.508228	LG_TF1	3860	98.299156	98.299156	13279.992
6	4004	LeadGlass	0	1	1	38	38	2.508228	LG_TF1	3860	8.512e-15	139.016	13279.992
7	4005	LeadGlass	0	1	1	38	38	2.508228	LG_TF1	3860	-98.29915	98.299156	13279.992
8	4006	LeadGlass	0	1	1	38	38	2.508228	LG_TF1	3860	-139.016	1.702e-14	13279.992
9	4007	LeadGlass	0	1	1	38	38	2.508228	LG_TF1	3860	-98.29915	-98.29915	13279.992
10	2000	ECalSenso	0	110	11	550	55	0.0225544	G4_Si	2330	304.13	0	4258.54
11	2000	ECalSenso	1	110	11	550	55	0.0225544	G4_Si	2330	304.13	0	4263.042
12	2000	ECalSenso	2	110	11	550	55	0.0225544	G4_Si	2330	304.13	0	4267.544

Tracks TTree

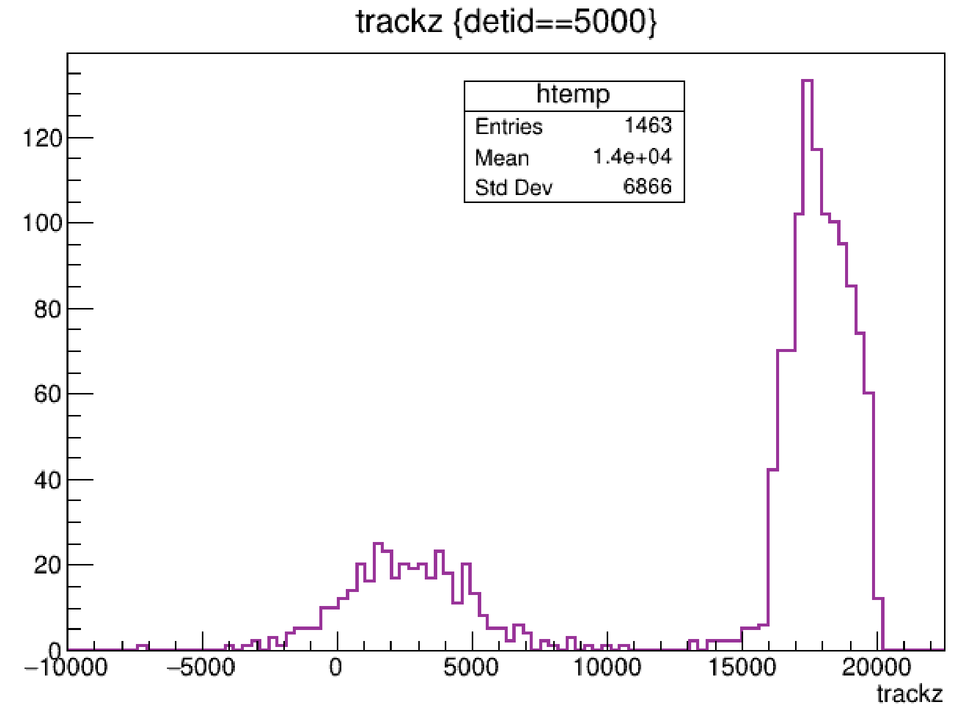
```
CreateNtuple("Tracks", "Tracks hitting volumes marked for track interception");
CreateNtupleIColumn(1, "eventid");                                  //0
CreateNtupleIColumn(1, "trackid",  fvolTrackIMap[16]);              //1
CreateNtupleIColumn(1, "detid",    fvolTrackIMap[17]);              //2
CreateNtupleIColumn(1, "pdg",      fvolTrackIMap[18]);              //3
CreateNtupleIColumn(1, "physproc", fvolTrackIMap[19]);              //4
CreateNtupleDColumn(1, "E", fvolTrackDMap[0]);                      //5
CreateNtupleDColumn(1, "x", fvolTrackDMap[1]);                      //6
CreateNtupleDColumn(1, "y", fvolTrackDMap[2]);                      //7
CreateNtupleDColumn(1, "z", fvolTrackDMap[3]);                      //8
CreateNtupleDColumn(1, "t", fvolTrackDMap[4]);                      //9
CreateNtupleDColumn(1, "vtxx", fvolTrackDMap[5]);                   //10
CreateNtupleDColumn(1, "vtxy", fvolTrackDMap[6]);                   //11
CreateNtupleDColumn(1, "vtxz", fvolTrackDMap[7]);                   //12
CreateNtupleDColumn(1, "px",   fvolTrackDMap[8]);                   //13
CreateNtupleDColumn(1, "py",   fvolTrackDMap[9]);                   //14
CreateNtupleDColumn(1, "pz",   fvolTrackDMap[10]);                  //15
CreateNtupleDColumn(1, "theta",  fvolTrackDMap[11]);                //16
CreateNtupleDColumn(1, "phi",    fvolTrackDMap[12]);                //17
CreateNtupleDColumn(1, "xlocal", fvolTrackDMap[13]);                //18
CreateNtupleDColumn(1, "ylocal", fvolTrackDMap[14]);                //19
CreateNtupleDColumn(1, "zlocal", fvolTrackDMap[15]);                //20
CreateNtupleDColumn(1, "weight");                                   //21
CreateNtupleIColumn(1, "ptrackid", fvolTrackIMap[20]);              //22
CreateNtupleIColumn(1, "nsecondary", fvolTrackIMap[21]);            //23
CreateNtupleDColumn(1, "esecondary", fvolTrackDMap[22]);            //24
FinishNtuple(1);
```

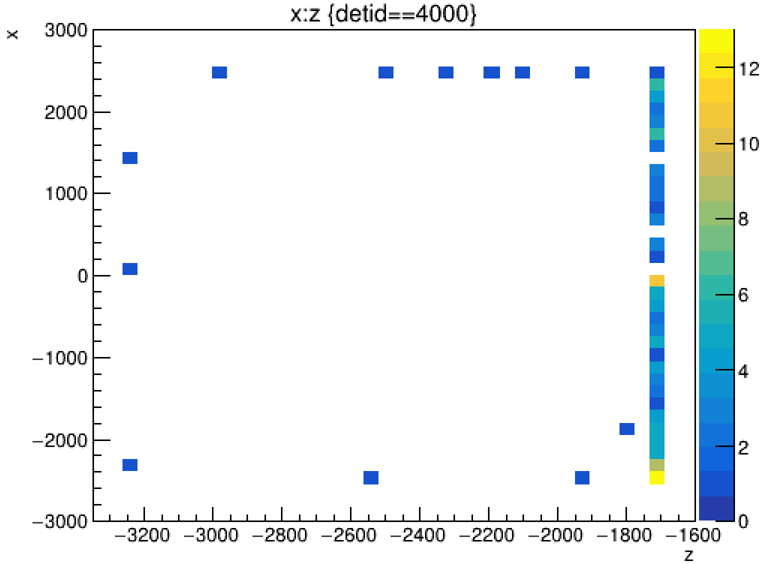
Simulation example

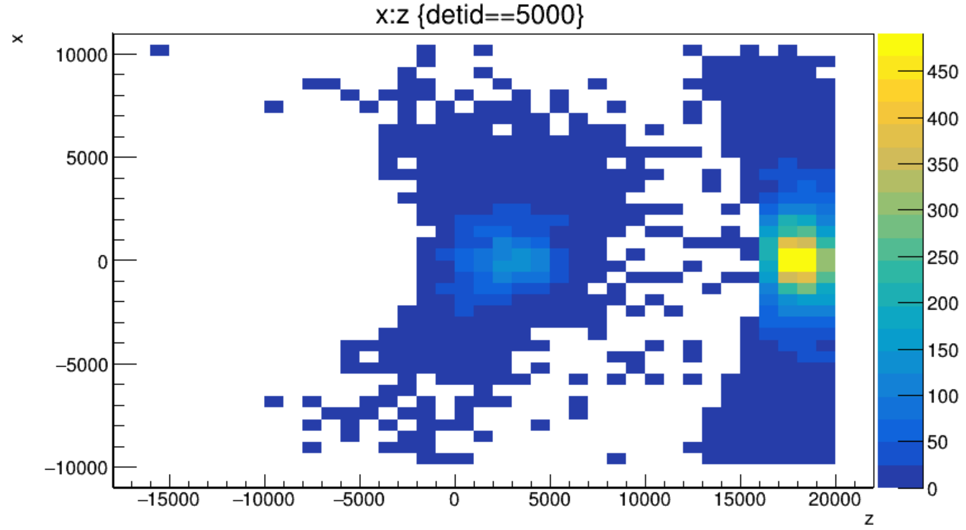
```
/luxe/run/add_intercept_volume  OpppDetContainer:1000
/luxe/run/add_intercept_volume  ComptonDetContainer:2000
/luxe/run/add_intercept_volume  GammaCalo:3000
/luxe/run/add_intercept_volume  Shielding:4000
/luxe/run/add_intercept_volume  Floor:5000

/run/initialize

/luxe/run/add_sensitive_volume  OpppTracker:1000:1:0
/luxe/run/add_sensitive_volume  ComptonTracker:2000:1:0
/luxe/run/add_sensitive_volume  GammaCalo:3000:1:0
/luxe/run/add_sensitive_volume  Shielding:4000:1:0
/luxe/run/add_sensitive_volume  Floor:5000:1:0
```







Summary

In experiment typically:

- specific settings of experiment (detectors, calibration, beam, etc.) are assigned to Run;
- Each BX can produce a Trigger with a certain ID which is assigned to an event;
- Event - Collection of recorded detectors response assigned to the trigger.

In simulation:

- Specific settings of experiment (detectors, beam, etc.) can be assigned to Run;
- Collection of (primary) MC particles with initial state (particle ID, position, momentum):
  - could be 1BX weighted (signal also signal+background);
  - could be specific distribution of primaries, not weighted 1.5e9 per BX (beam background);
- Simulation of each particle in geometry is independent on others:
  - Detector response recorded for to each individual particle
- The concept of trigger is not considered in simulation, at least for the time being;
- Event – as a collection of detector responses assigned to trigger is a bit artificial:
  - Can be composed of signal and background

Summary

Link configuration with simulation results:

- Geometry:
  - TTree for selected volumes: position size weight, misalignment;
  - Complete geometry in gdml (implemented) or root.
- Primary particles configuration:
  - Laser intensity;
  - Beam settings: type, energy, etc.
- Fields;

Consider intermediate software layer to combine simulated results to Event data:

- Geant4 simulation optimized focused on good timing not on data processing;
- Flexibility in implementing different detector response models.

Summarize experience with present data model and combine wish list.

Backup

Primary beam settings

```
# Primary beam settings
############################################
#
#/lxphoton/gun/setDefault
#/lxphoton/gun/beamType    mono

#/lxphoton/gun/beamType    gaussian
#/lxphoton/gun/setSigmaX   5 um
#/lxphoton/gun/setSigmaY   5 um
#/lxphoton/gun/setFocus    8.1 m

/lxphoton/gun/beamType    mc
/lxphoton/gun/MCParticlesFile   test_data_0.out

#/lxphoton/gun/SpectraFile   spectra_test1.txt
#/lxphoton/gun/SpectraFile   spectra_test_compt.txt

/gun/particle    e-
/gun/energy      17.5 GeV
```

Fixed momentum beam

XFEL beam with fixed emittance and given $\sigma_{x,y}$ and focusing distance.

MC, typically Anthony's out file

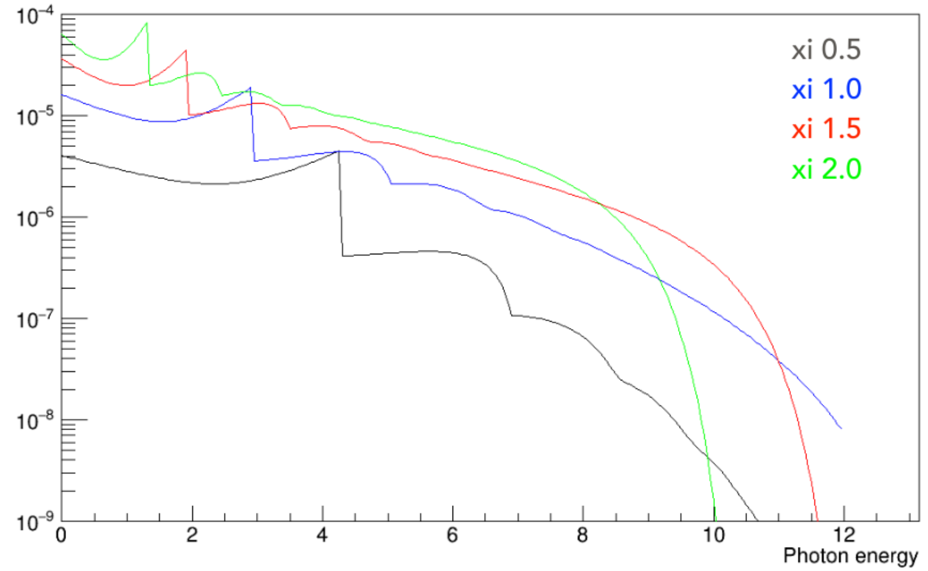
Arbitrary spectra which will be interpolated and used for generating particles

Ignored for MC

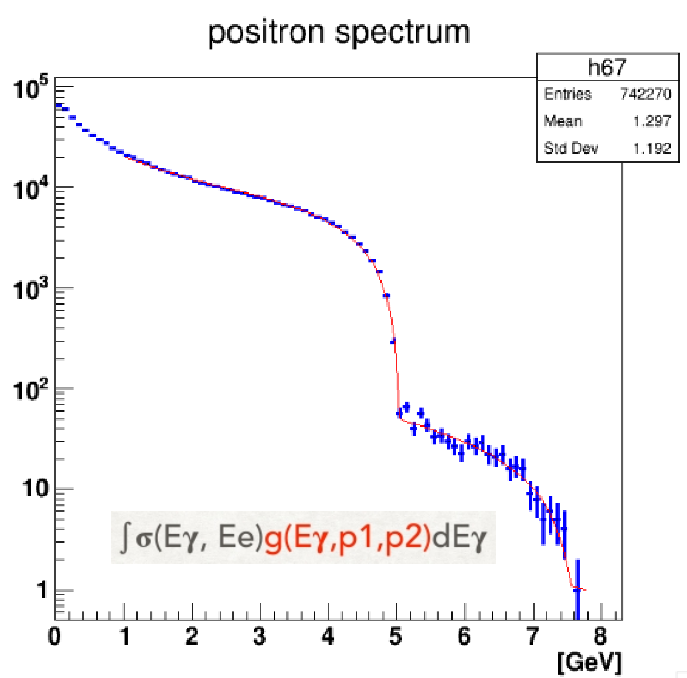
Primaries with a given spectra

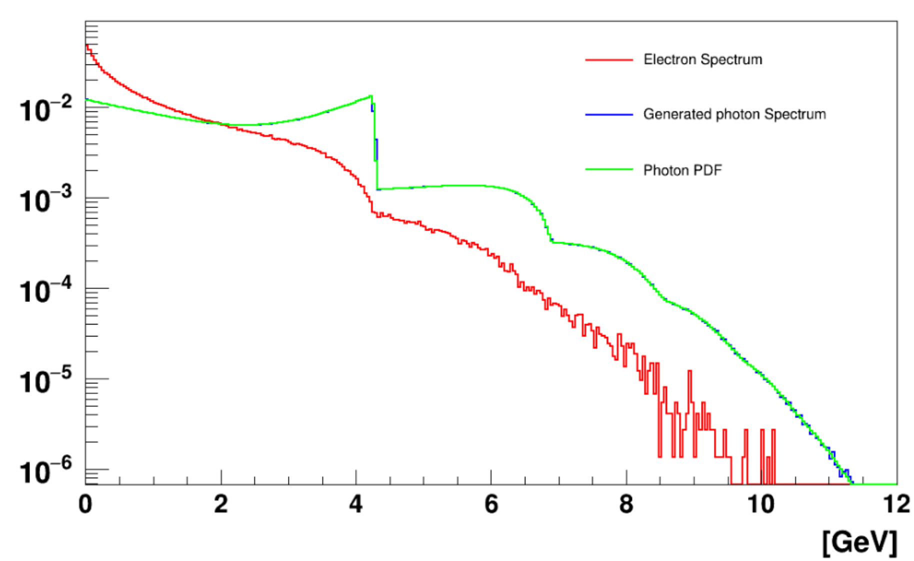
Photon detector performance study w/o MC

HICS spectra as tabulated function calculated by Anthony



GEANT4, primary and positrons after the target

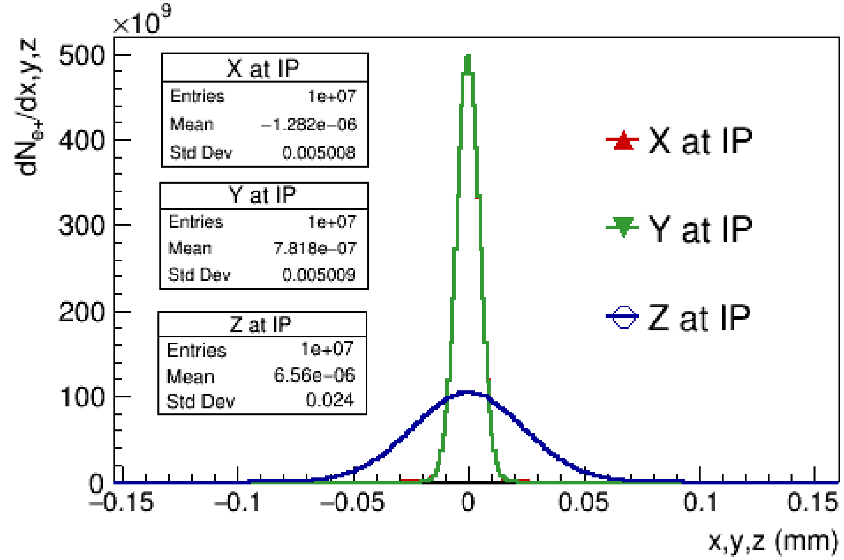




Gaussian beam

$$f(x,x',z) = \frac{1}{2\pi\varepsilon}\exp\left[-\frac{(x-zx')^2}{2\varepsilon\beta^*} - \frac{\beta^* x'^2}{2\varepsilon}\right]$$

$$\beta^* = \frac{\sigma^{*2}}{\varepsilon},$$



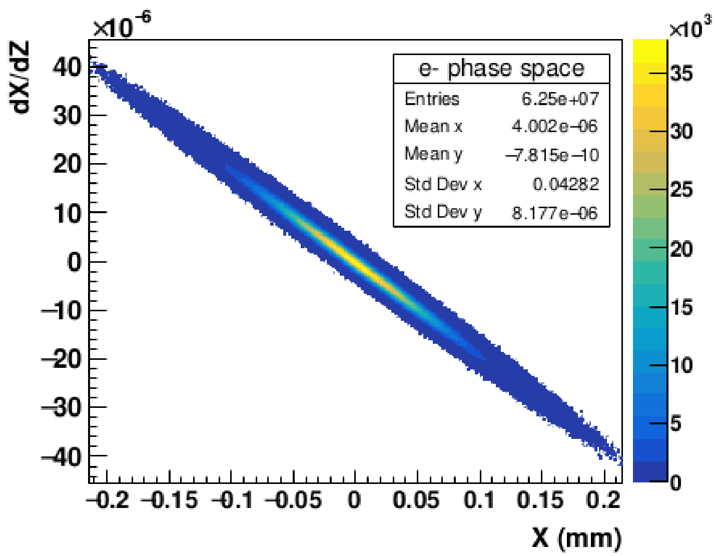


Figure 2: Phase space distribution of the primary electron beam 5.2 m upstream from interaction point.

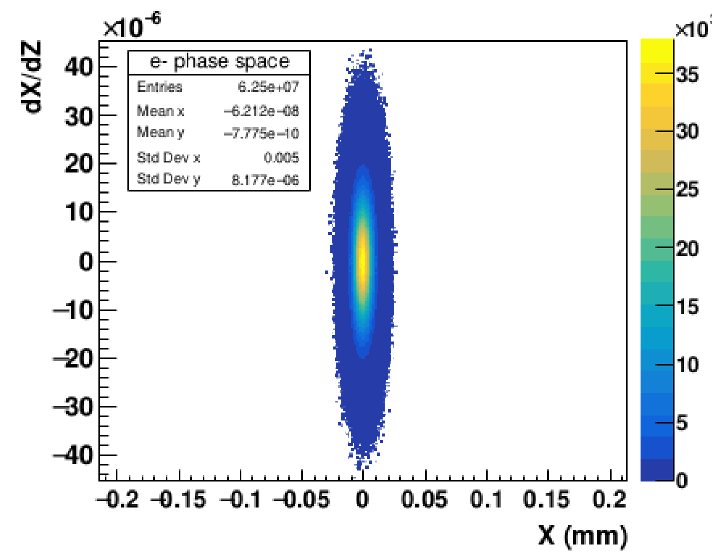


Figure 3: Phase space distribution of the electron beam at the interaction point.

Physics list settings

```
# Simulation settings
############################################
/lxphoton/phys/addPhysics  emstandard_opt0
#/lxphoton/phys/addPhysics local

/lxphoton/stepMax        1 um
#/lxphoton/stepMax        10 um
#
/run/setCut              1 um
#
```

```
if (name == "local") {
  AlterPhysicsList(name, new PhysListEmStandard(name));

} else if (name == "emstandard_opt0") {
  AlterPhysicsList(name, new G4EmStandardPhysics());

} else if (name == "emstandard_opt1") {
  AlterPhysicsList(name, new G4EmStandardPhysics_option1());

} else if (name == "emstandard_opt2") {
  AlterPhysicsList(name, new G4EmStandardPhysics_option2());

} else if (name == "emstandard_opt3") {
  AlterPhysicsList(name, new G4EmStandardPhysics_option3());

} else if (name == "emstandard_opt4") {
  AlterPhysicsList(name, new G4EmStandardPhysics_option4());

} else if (name == "emstandardSS") {
  AlterPhysicsList(name, new G4EmStandardPhysicsSS());

} else if (name == "standardSSM") {
  AlterPhysicsList(name, new PhysListEmStandardSSM());

} else if (name == "emstandardWVI") {
  AlterPhysicsList(name, new G4EmStandardPhysicsWVI());

} else if (name == "standardGS") {
  AlterPhysicsList(name, new PhysListEmStandardGS());

} else if (name == "empenelope"){
  AlterPhysicsList(name, new G4EmPenelopePhysics());

} else if (name == "emlowenergy"){
  AlterPhysicsList(name, new G4EmLowEPPhysics());

} else if (name == "emlivermore"){
  AlterPhysicsList(name, new G4EmLivermorePhysics());

} else {

  G4cout << "PhysicsList::AddPhysicsList: <" << name << ">"
         << " is not defined"
         << G4endl;
}
```

Output of hits and tracks

```
->CreateNtuple("Hits", "Hits in sensitive detectors");
->CreateNtupleIColumn(2, "eventid");
->CreateNtupleIColumn(2, "detid");
->CreateNtupleIColumn(2, "layerid");
->CreateNtupleIColumn(2, "cellx");
->CreateNtupleIColumn(2, "celly");
->CreateNtupleDColumn(2, "edep");
->CreateNtupleIColumn(2, "hitid");
->CreateNtupleIColumn(2, "track_list", fvHitTrackList);
->CreateNtupleDColumn(2, "trackx", ftrackx);
->CreateNtupleDColumn(2, "tracky", ftracky);
->CreateNtupleDColumn(2, "trackz", ftrackz);
->CreateNtupleDColumn(2, "weight");
->FinishNtuple(2);
```

```
->CreateNtuple("HitTracks", "Tracks which produced hits in sensitive detectors");
->CreateNtupleIColumn(3, "eventid");
->CreateNtupleIColumn(3, "trackid", fvTracks);
->CreateNtupleDColumn(3, "vtxx", fVtxx);
->CreateNtupleDColumn(3, "vtxy", fVtxy);
->CreateNtupleDColumn(3, "vtxz", fVtxz);
->CreateNtupleDColumn(3, "px", fPx);
->CreateNtupleDColumn(3, "py", fPy);
->CreateNtupleDColumn(3, "pz", fPz);
->CreateNtupleDColumn(3, "E", fE);
->CreateNtupleIColumn(3, "pdg", fPDG);
->CreateNtupleIColumn(3, "pproc", fPhysProc);
->CreateNtupleIColumn(3, "ptid", fPTId);
->CreateNtupleDColumn(3, "weight");
->FinishNtuple(3);
```

MC weight with scaling possibility

SQL query would be:

SELECT *
FROM Hits **INNER JOIN** HitTracks
ON (Hits.eventid == HitTracks.eventid) **AND** (Hits.track_list == HitTracks.trackid)

Add detector

```
#include "LxDetector.hh"

class LxDetectorGammaCalo: public LxDetector
{
  public:
    LxDetectorGammaCalo(DetectorConstruction *detc = 0): LxDetector(detc) {};
    virtual ~LxDetectorGammaCalo() {};
    virtual void Construct();

  protected:
    void AddSegmentation();
};
```

```
void DetectorConstruction::ConstructLuxeDetectors()
{
  fDetList["OpppDet"] = new LxDetectorOPPP(this);
  fDetList["ComptonDet"] = new LxDetectorCompton(this);
  fDetList["GammaDet"] = new LxDetectorGammaCalo(this);

  for (auto &nd : fDetList) { nd.second->Construct(); }
}
```