

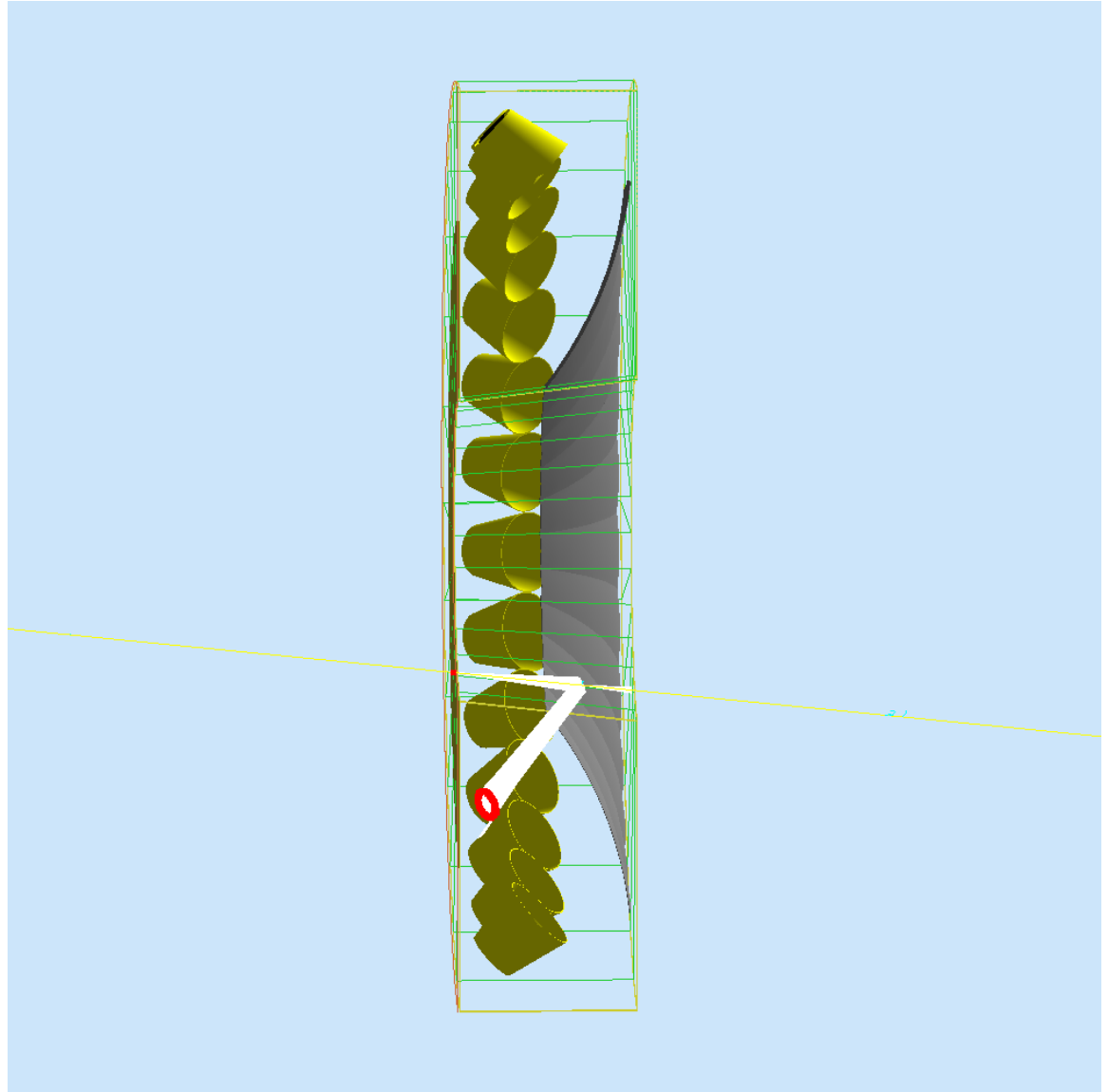
SoLID simulation with GEMC

Zhiwen Zhao

2015/09/11

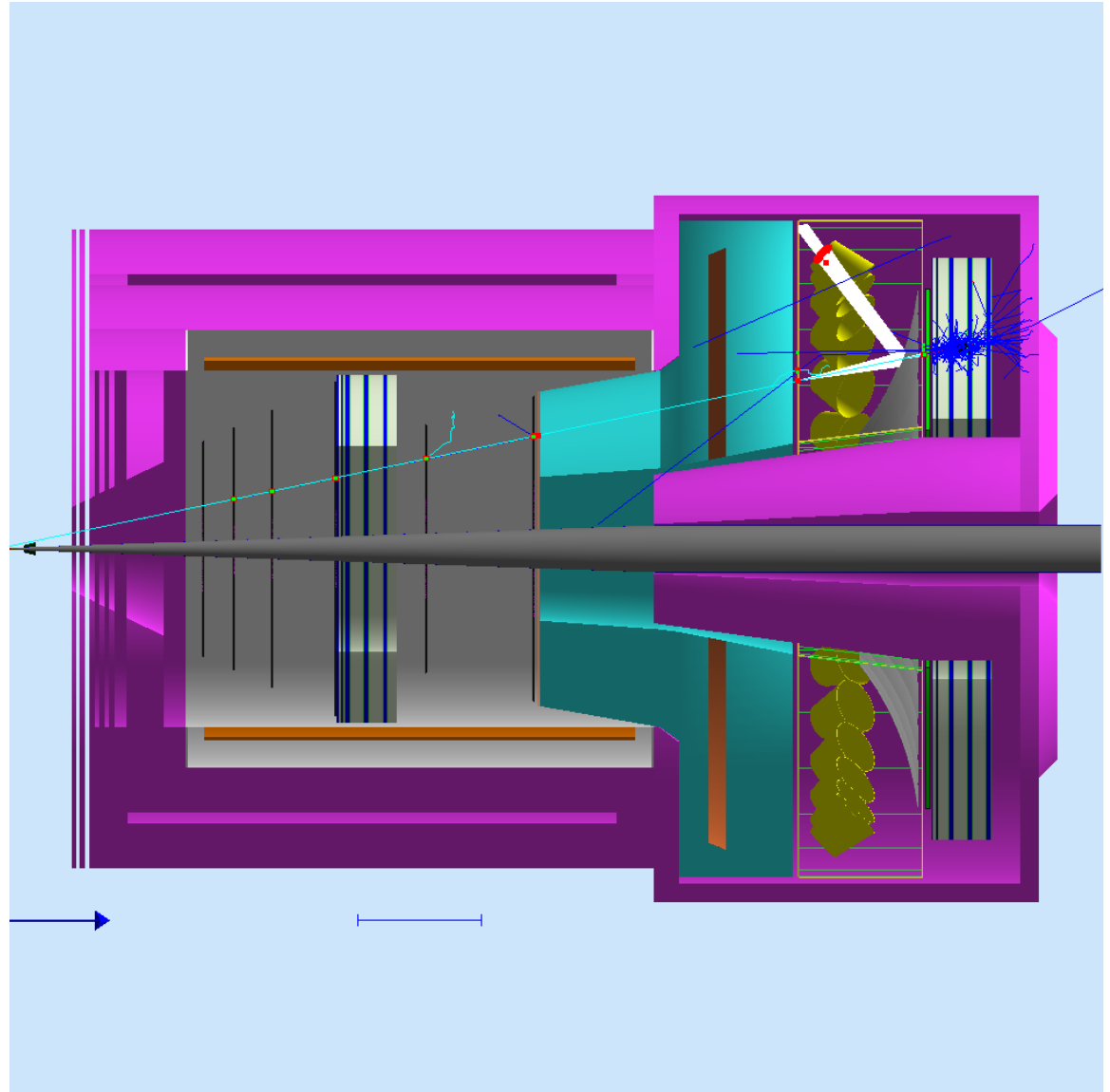
Standalone simulation in GEMC

- HGC



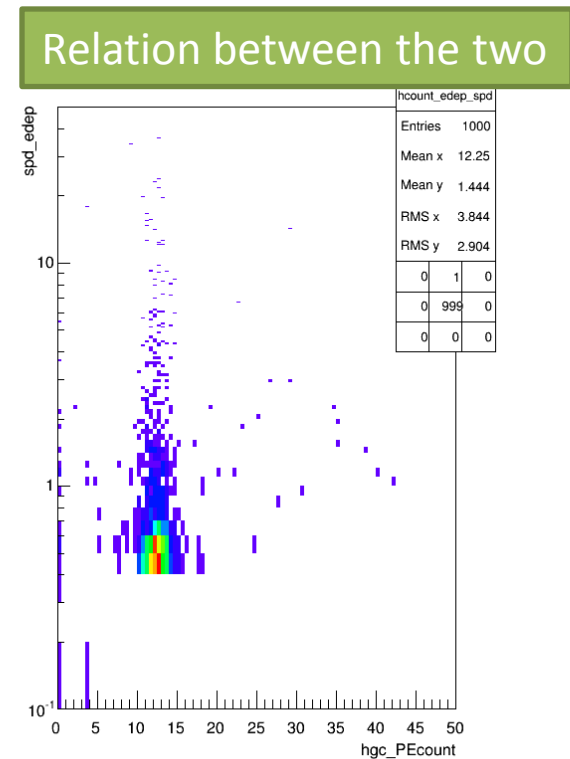
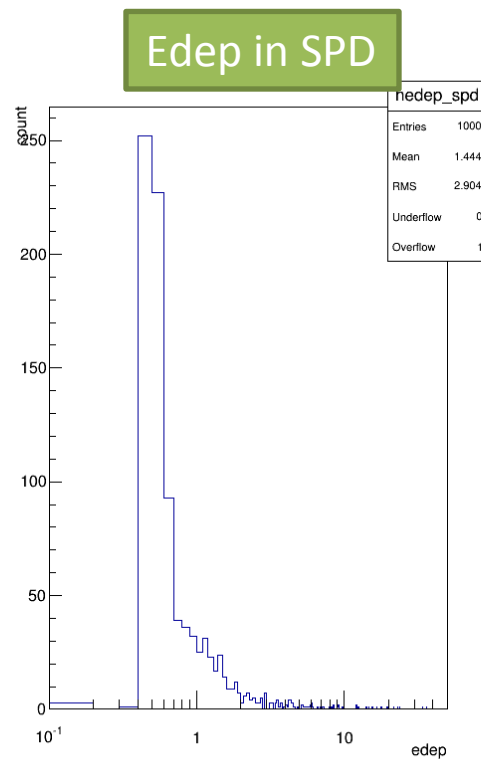
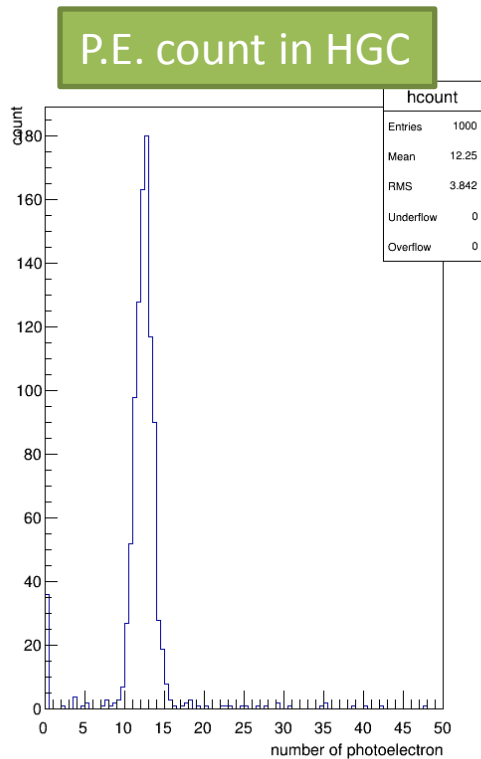
Full SoLID simulation in GEMC

- Just drop-in with others
- Run with same field file, same condition, same output
- Nothing needs change



An example

- Shoot 4GeV pi- from target into forward detector



Everything external

- “Geometry, parameter, material, optical properties, output content” are all external
- They can be stored in file (regular version control) or database (CCDB)
- “parameter” can be anything, particularly useful to reconstruction/calibration/analysis
- “output content” defines data flow
- Both “Parameter” and “output content” can evolve

solid_gemc

- GEMC compiled as a lib during installation by default
- Then add customized hit process routing and other things and link to libgemc.so to keep all GEMC features
- This becomes “solid_gemc” with matching version of GEMC

solid/solid_gemc2/source

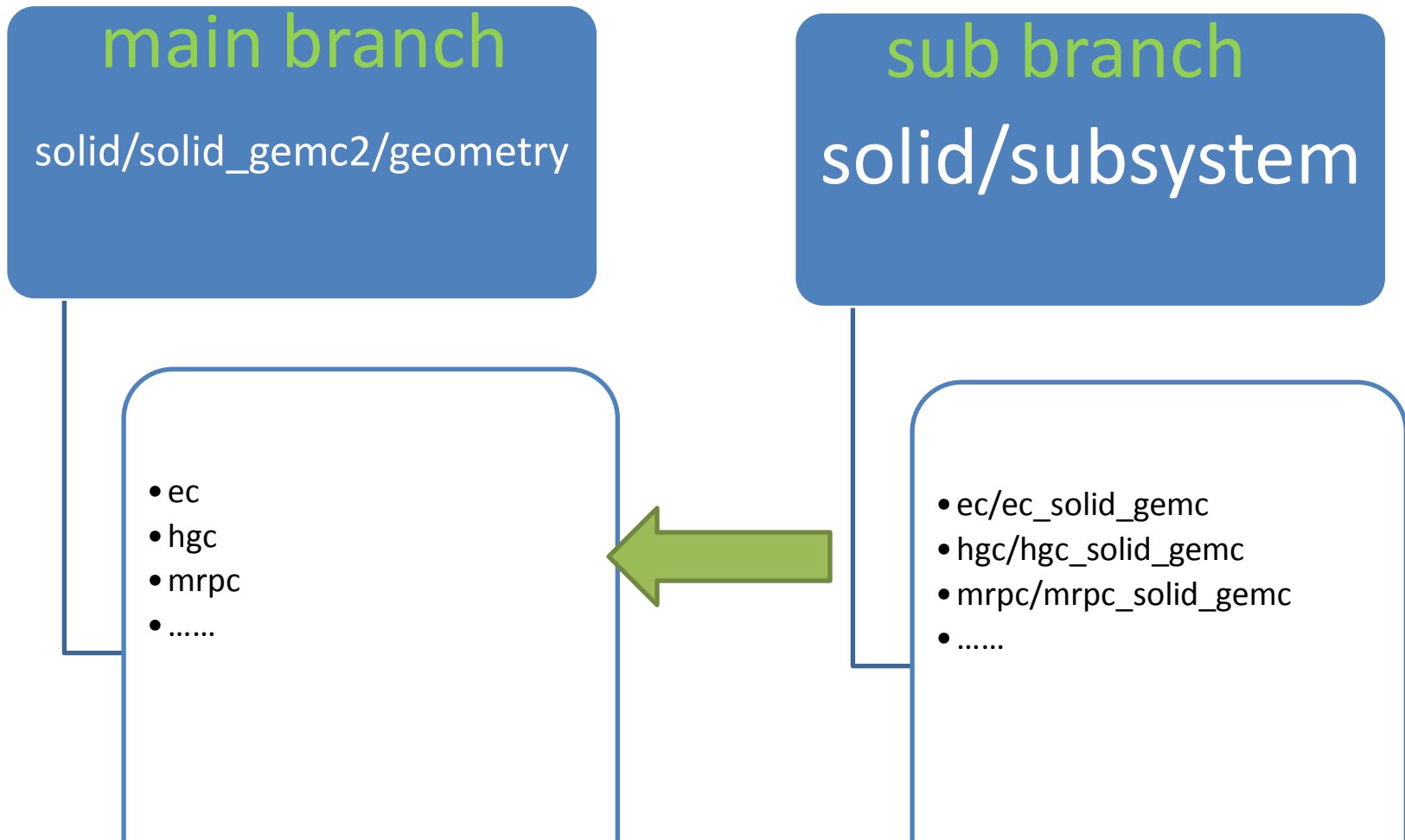
2.1

2.2

- solid_gemc.cc
- hitprocess
 - solid_hitprocess.h
 - solid_ec_hitprocess.cc
 - solid_ec_hitprocess.h
 -

Development model

- test standalone in sub branch -> test with others from main branch in sub branch -> push from sub branch into main branch



Other features of GEMC 2.x

- GEMC 1.x to 2.x, a major code rewrite
- use factory method as much as possible, easy to add plug-in and expand functionality
- Field map
 - Field info embedded in field map file, no separated definition needed
 - Map Reading 3-4x faster Swimming 30% faster
- Modular physics list (hadron+EM+optical)
- Built-in hit type “flux” and step-by-step hit info
- customized event generator input
- All codes are packaged by a single release number “jlab_version” with streamlined installation
- Took a lot suggestion from SoLID and MEIC simulation and we contribute to its code and structure also

Other features of GEMC 2.x

The screenshot displays the GEMC 2.x GUI. The top panel is the 'Generator' section, which includes controls for 'N. Events: 1', 'Run', 'Cycle', 'Stop', and 'Exit' buttons. It features a 'Momentum' section with 'Particle Type: e-' and input fields for p (8 GeV), θ (25 deg), and ϕ (0 deg). The 'Vertex' section has input fields for vX, vY, and vZ (10 cm). The bottom panel is the 'Signals' section, showing 'N. Events: 1' and 'Run', 'Cycle', 'Stop', 'Exit' buttons. It includes a 'Hits List' table with columns for Hit ID, nsteps, and E Dep. The table shows three hits: Hit n. 1 (195 hits, 207 nsteps), Hit n. 2 (134 hits, 134 nsteps), and Hit n. 3 (118 hits, 118 nsteps). Below the table is a 'Signal' plot for 'id 310000' showing E Dep. vs time [ns].

Voltage Signal

This screenshot shows the 'Voltage Signal' feature. It includes a 'Hits List' table with columns for Hit ID, nsteps, and E Dep. The table shows three hits: Hit n. 1 (1 hit, 8 nsteps), Hit n. 2 (5 hits, 5 nsteps), and Hit n. 3 (8 hits, 8 nsteps). Below the table is a 'Signal' plot for 'sector 1 paddle 48' showing Voltage vs time [ns]. The plot shows a series of step-like signals. To the right of the plot, the following parameters are listed: Rise time: 1ns, Fall time: 2ns, Delay: 50ns, and 1 MeV = 100 mV. Below the plot is a 'Total Signal' plot for 'sector 1 paddle 48' showing Voltage vs time [ns].

12

- Updated GUI
- voltage signal and FADC support

GEMC future

- Large user base: CLAS12, MEIC, SoLID, many smaller projects
- In github now, “github.com/gemc”, has nice tool for debugging, feature adding, contributing
- Started “gemc collaboration” to gain input and help from developers and users

Summary

- status
 - EC, GEM, MRPC, SPD, HGC have initial implementation at various stages
 - LGC is under work (in a modified GEMC 1.x now)
- advantage
 - fast MC and full MC within one framework
 - standalone and full simulation within one framework
 - Things can evolve with new input format, new reconstruction/analysis framework

backup

GEMC Update: Outline

- GEMC 1.x to 2.x, a major code rewrite
- use factory method as much as possible, easy to add plug-in and expand functionality
- Field map
 - Field info embedded in field map file, no separated definition needed
 - Map Reading 3-4x faster Swimming 30% faster
- Modular physics list (hadron+EM+optical)
- Updated GUI
- geometry and parameters, material, optical properties, hit process and output are all external
- Built-in hit type “flux” and step-by-step hit info
- New features like customized event generator input, voltage signal and FADC support
- Took a lot suggestion from SoLID and MEIC simulation and we contribute to its code and structure also

GEMC Update: installation

- software installation streamlined by a set of scripts
- part of the general jlab software framework (including jana, ccdb and more)
- Everything in release package, no SVN or github download
- a single version control by env “Jlab_version”
- Just an “App” to download on Mac, no install needed

Jlab_version	1.0	1.1	1.2	devel
GEMC	1.8	2.1	2.2	2.3?
geant4	9.5	9.5/9.6	10.0	10.1?

GEMC update: GUI

- qt5 and openGL

The screenshot displays the GEMC GUI interface. On the left, a vertical sidebar contains icons for Generator, Camera, Detector, Infos, G4Dialog, Signals, and Physics. The main window is divided into several panels:

- Top Panel:** N. Events: 1, Run, Cycle, Stop, Exit buttons.
- Generator Panel:** Beam 1, Beam 2 tabs. Momentum: Particle Type: e-. p: 8 ± 0 GeV, θ : 25 ± 0 deg, ϕ : 0 ± 0 deg.
- Vertex Panel:** vx: 0 Δ x: 0, vy: 0 Δ z: 0, vz: 10 Units: cm.
- Bottom Panel:** N. Events: 1, Run, Cycle, Stop, Exit buttons.
- Hits List Panel:** E Dep. dropdown, Hits List table, and Signal table.
- Signal Panel:** A scatter plot titled "id 310000" showing E Dep. vs time [ns].

Hit n.	nsteps
Hit n. 1	nsteps: 207
Hit n. 2	nsteps: 134
Hit n. 3	nsteps: 118

Signal	solid_ec	Hit n.	nsteps
solid_ec	Hit n. 1	nsteps: 207	
	E Dep.	pid	Time[ns]
	1.54252	11	11.6000
	0.66429	11	11.6141

A 3D visualization of particle tracks in a detector. The tracks are represented by blue lines originating from a central point and spreading outwards. A vertical black bar represents the detector's sensitive volume. The tracks are shown in a perspective view, with a light blue background.

GEMC update: Input

- All generators are still fully independent, interface by txt file in customized LUND format
- Fully pass customized header info into output

GEMC

Home Documentation Gallery Support Downloads Web Interface

THE LUND FORMAT

The LUND format is a text file that can be used feed gemc with physics events. It has a *header* to input the number N of generated particle for each event and other kinematic properties. Then it has N lines describing each particle property, as described in these tables.

In **BOLD**, the quantities actually used by GEMC.

A * indicate a quantity not actually used by GEMC, but kept in the output stream. User defined meanings (not LUND) can be assigned to these variables.

Header Infos

Column	Quantity
1	Number of particles
2*	Number of target nucleons
3*	Number of target protons
4*	Target Polarization
5	Beam Polarization
6*	x
7*	y
8*	W
9*	Q2
10*	nu

Particle Infos

Column	Quantity
1	index
2	charge
3	type (= 1 is active)
4	particle Id
5	parent id (decay bookkeeping)
6	daughter (decay bookkeeping)
7	p_x [GeV]
8	p_y [GeV]
9	p_z [GeV]
10	E [GeV]
11	mass (not used)
12	x vertex [cm]
13	y vertex [cm]
14	z vertex [cm]

Example of 2 DVCS events (ePy)

```

3 1. 1. 0 -1 0.209 0.336 6.373 1.448 -1.000
1 -1. 1 11 0 0 0.9636 -0.1675 7.2357 7.3015 0.0005 0.0000 0.0000 0.0000
2 1. 1. 2212 0 0 -0.6536 0.0604 0.3367 1.1935 0.9383 0.0000 0.0000 0.0000
    
```


GEMC Update: Optical

- All properties defined external in “table” format similar to geometry

Optical Properties

- surface
- type
- **optical properties:**
 - ▶ photonEnergy
 - ▶ indexOfRefraction
 - ▶ reflectivity
 - ▶ efficiency
 - ▶ specularlobe
 - ▶ specularspike
 - ▶ backscatter

```
Table of optical photon energies (wavelengths) from 190–650 nm:
my $photonEnergy =
" 1.9074494eV 1.9372533eV 1.9680033eV 1.9997453eV 2.0325280eV " .
" 2.0664035eV 2.1014273eV 2.1376588eV 2.1751616eV 2.2140038eV " .
" 2.2542584eV 2.2960039eV 2.3393247eV 2.3843117eV 2.4310630eV " .
" 2.4796842eV 2.5302900eV 2.5830044eV 2.6379619eV 2.6953089eV " .
" 2.7552047eV 2.8178230eV 2.8833537eV 2.9520050eV 3.0240051eV " .
" 3.0996053eV 3.1790823eV 3.2627424eV 3.3509246eV 3.4440059eV " .
" 3.5424060eV 3.6465944eV 3.7570973eV 3.8745066eV 3.9994907eV " .
" 4.1328070eV 4.2753176eV 4.4280075eV 4.5920078eV 4.7686235eV " .
" 4.9593684eV 5.1660088eV 5.3906179eV 5.6356459eV 5.9040100eV " .
" 6.1992105eV ";

Reflectivity of AlMgF2 coated on thermally shaped acrylic sheets, measured by AJRP,
10/01/2012:
my $reflectivity =
" 0.8331038 0.8309071 0.8279127 0.8280742 0.8322623 " .
" 0.837572 0.8396875 0.8481834 0.8660284 0.8611336 " .
" 0.8566167 0.8667431 0.86955 0.8722481 0.8728122 " .
" 0.8771635 0.879907 0.879761 0.8831943 0.8894673 " .
" 0.8984234 0.9009531 0.8910166 0.8887382 0.8869093 " .
" 0.8941976 0.8948479 0.8877356 0.8926919 0.8999685 " .
" 0.9101617 0.8983005 0.8991694 0.8990987 0.9000493 " .
" 0.9065833 0.9028855 0.8985184 0.9009736 0.9088968 " .
" 0.9015145 0.8914838 0.8816829 0.8666895 0.8496298 " .
" 0.9042583 ";
```


GEMC Update: Outlook

- FADC banks will emulate FADC output, including translation tables
- Hit process routines will be a plugin - completely independent from GEMC, to be loaded at run time. The routines can be stored together with the geometry / materials scripts.
- multithreading, following G4
- C++ 11

EC simulation with solid_gemc

- solid - Revision 746: /subsystem/ec/ec_solid_gemc
- ..
- config_solid_PVDIS_ec_forwardangle.dat
- readme
- solid_PVDIS_ec_forwardangle.gcard
- solid_PVDIS_ec_forwardangle.pl
- solid_PVDIS_ec_forwardangle__bank.txt
- solid_PVDIS_ec_forwardangle__geometry_Original.txt
- solid_PVDIS_ec_forwardangle__hit_Original.txt
- solid_PVDIS_ec_forwardangle__materials_Original.txt
- solid_PVDIS_ec_forwardangle__parameters_Original.txt
- solid_PVDIS_ec_forwardangle_real.pl
- solid_ec_bank.pl
- solid_ec_hit.pl
- solid_ec_materials.pl
- solid_slice.vis

configuration file for generating

input file to run simulation

file to generate geomtry,bank,hit

generated bank

generated geometry

generated hit

generated material

parameters defines geometry

file to generate geometry

file to generate bank

file to generate hit

file to generate material

- Purple and orange files needed to generate files for simulation
- Red files needed to run simulation
- Red and orange files are in “table” format, they can be in
 - txt file
 - Database (mysql now, CCDB soon)
 - expanded to more sources

An example line of a txt file

```
solid_PVDIS_ec_forwardangle_real_shower | root
|solid_PVDIS_ec_forwardangle_real_shower | 0*cm 0*cm
350*cm | 0*deg 0*deg 0*deg | ff0000 | Tube | 110*cm
365*cm 21.728*cm 0*deg 360*deg | G4_AIR | no | 1 | 1 | 1
| 1 | 1 | no | no | no
```

Customized hit process routine

“solid_ec”

```
insert_bank_variable(\%configuration, $bankname, "bankid", $bankId, "Di", "$bankname bank ID");
insert_bank_variable(\%configuration, $bankname, "pid", 1, "Di", "ID of the first particle entering the sensitive volume");
insert_bank_variable(\%configuration, $bankname, "mpid", 2, "Di", "ID of the mother of the first particle entering the sensitive volume");
insert_bank_variable(\%configuration, $bankname, "tid", 3, "Di", "Track ID of the first particle entering the sensitive volume");
.....
insert_bank_variable(\%configuration, $bankname, "id", 24, "Di", "id number");
insert_bank_variable(\%configuration, $bankname, "hitn", 99, "Di", "Hit Number");
```

solid_ec_bank.pl

```
dgtz["pid"] = (double) aHit->GetPID();
dgtz["mpid"] = (double) aHit->GetmPID();
dgtz["tid"] = (double) aHit->GetTId();
.....
dgtz["id"] = id;
dgtz["hitn"] = hitn;
```

solid_ec_hitprocess.cc

- Totally flexible to any raw and digitized hit processing and output, fit any level of simulation and digitization need
- As far as the two match each other to give consistent result, “solid_gemc” need recompile if any change

Hit process control

solid_ec_hit.pl

```
$hit{"name"} = "solid_ec";  
$hit{"description"} = "solid ec hit definition";  
$hit{"identifiers"} = "id";  
$hit{"signalThreshold"} = "0*MeV";  
$hit{"timeWindow"} = "400*ns";  
$hit{"prodThreshold"} = "1*mm";  
$hit{"maxStep"} = "1*cm";  
$hit{"delay"} = "10*ns";  
$hit{"riseTime"} = "1*ns";  
$hit{"fallTime"} = "1*ns";  
$hit{"mvToMeV"} = 100;  
$hit{"pedestal"} = -20;
```

- Fine tuning hit processing without source code change
- No need to recompile solid_gemc

External parameters

solid_PVDIS_ec_forwardangle__parameters_
Original.txt

```
Nlayer | 194 | | Nlayer | - | - | - | - | - | - | -  
Thickness_lead | 0.05 | cm | Thickness_lead | - | - | - | - | - | - | -  
Thickness_scint | 0.15 | cm | Thickness_scint | - | - | - | - | - | - | -  
Thickness_gap | 0.024 | cm | Thickness_gap | - | - | - | - | - | - | -  
Thickness_shield | 1.0274 | cm | Thickness_shield | - | - | - | - | - | - | -  
Thickness_prescint | 2 | cm | Thickness_prescint | - | - | - | - | - | - | -  
Thickness_support | 2 | cm | Thickness_support | - | - | - | - | - | - | -  
z_shower | 350 | cm | z_shower | - | - | - | - | - | - | -  
Rmin | 110 | cm | Rmin | - | - | - | - | - | - | -  
Rmax | 365 | cm | Rmax | - | - | - | - | - | - | -  
Sphi | 0 | deg | Sphi | - | - | - | - | - | - | -  
Dphi | 360 | deg | Dphi | - | - | - | - | - | - | -
```

- It can take source from survey data
- And can be part of calibration database

Other things

- Record simulation condition
 - GEMC record all input options into EVIO file
 - We can think of ways to record detector related input (as SVN or github version or database entry with index like run number? It will depends on where we store them)
- Output file format
 - “evio2root”, convert evio to root tree, included with framework
 - “clas-root”, read evio like a root tree, will include <https://userweb.jlab.org/~gavalian/clas12docs/sphinx/html/rootio/introduction.html>
- Documentation
 - Nice GEMC tutorials available
 - Doxygen for source code
 - Wiki [https://hallaweb.jlab.org/wiki/index.php/Solid Software](https://hallaweb.jlab.org/wiki/index.php/Solid_Software)