Recent Developments for the CLAS12 Simulation

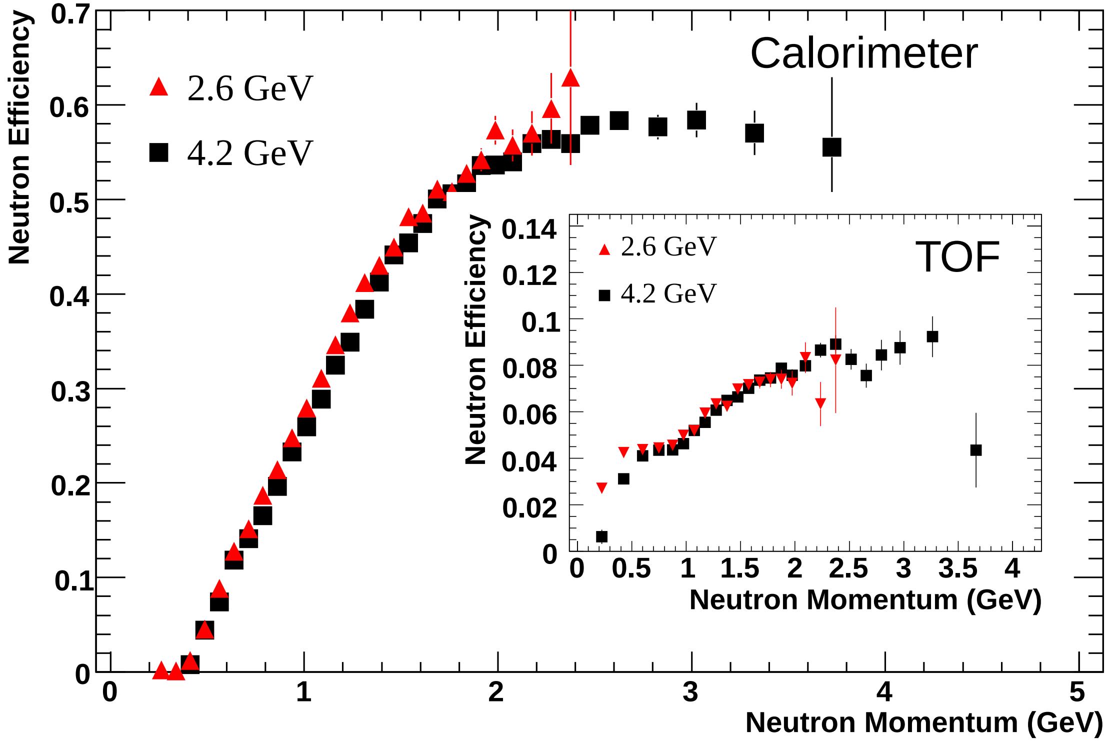
G.P.Gilfoyle, M. Moog, M.Ungaro, et al.

CLAS12 Software Group

Outline:
1. Physics Motivation.
2. Neutron Detection Efficiency in the CLAS12 Outer TOF.
3. Implementing the CLAS6 EC in CLAS12.
4. Future Plans.

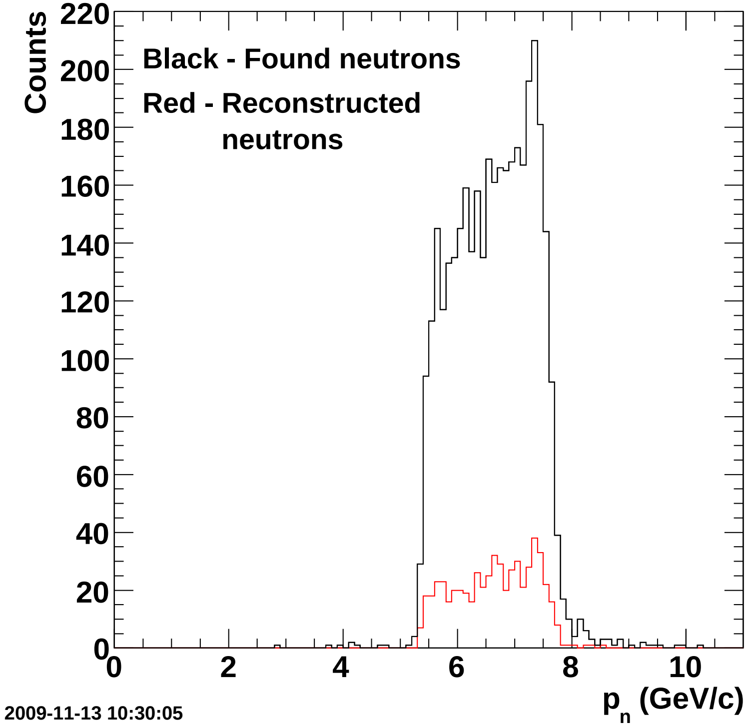
Simulation of CLAS12 OTOFP1A NDE

- Neutron detection efficiency (NDE) needed for E12-07-104 (G_M^n experiment) and others.
- Study first TOF panel (OTOFP1A); we have CLAS6 NDE measurements for these scintillators.



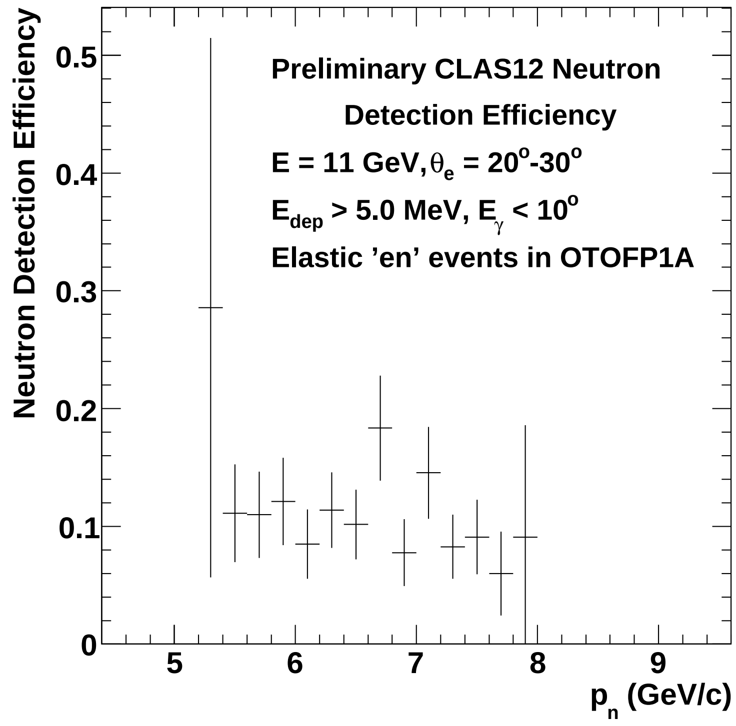
Simulation of CLAS12 OTOFP1A NDE

- Generate 'elastic' en events and process with gemc.
- Reconstruct electrons with Socrat.
- Modify Socrat to include OTOFP1A information.
- Found neutrons - Events where neutron is predicted to hit OTOFP1A using only electron information.
- Reconstructed neutrons - Apply cut on angle between the predicted neutron direction and vertex-to-TOF-hit vector ($\theta_\gamma < 10^\circ$) and require $E_{dep} > 5\ MeV$.



Simulation of CLAS12 OTOFP1A NDE

- Simulated NDE is about 10% at 6-8 GeV/c versus 9% at 3 GeV/c for CLAS6.



- Much more to be done: timing cuts, p_n dependence, simulate $^1H(e,e'\pi^+n)$ reaction for tagged neutrons.

Adding the EC to the CLAS12 Simulation

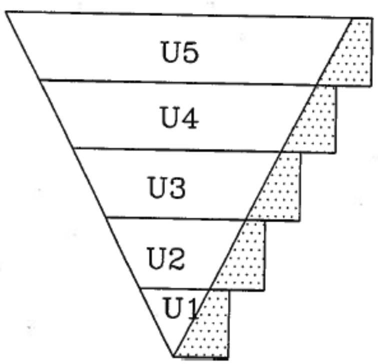
1. The CLAS6 EC geometry - defined by R. Minehart in `http://www.jlab.org/~gilfoyle/CLAS12software/CLAS6ECgeometry.pdf`

2. Active region defined by the following equations. The vertex of the triangle at small scattering angle and the top of the triangle at large scattering angle are

$$y_-(L) = A_- + B_-(L-1) \ \ (vertex) \qquad y_+(L) = A_+ + B_+(L-1) \ \ (top)$$

   where L is the layer number, $A_- = -1829.74\,mm$, $B_- = -4.3708\,mm$, $A_+ = 1899.56\,mm$, and $B_+ = 4.5419\,mm$. The other two lines defining the active region are the following where $\tan\theta = 1.95325$.

$$y + x\tan\theta = y_-(L_-) \qquad y - x\tan\theta = y_-(L)$$

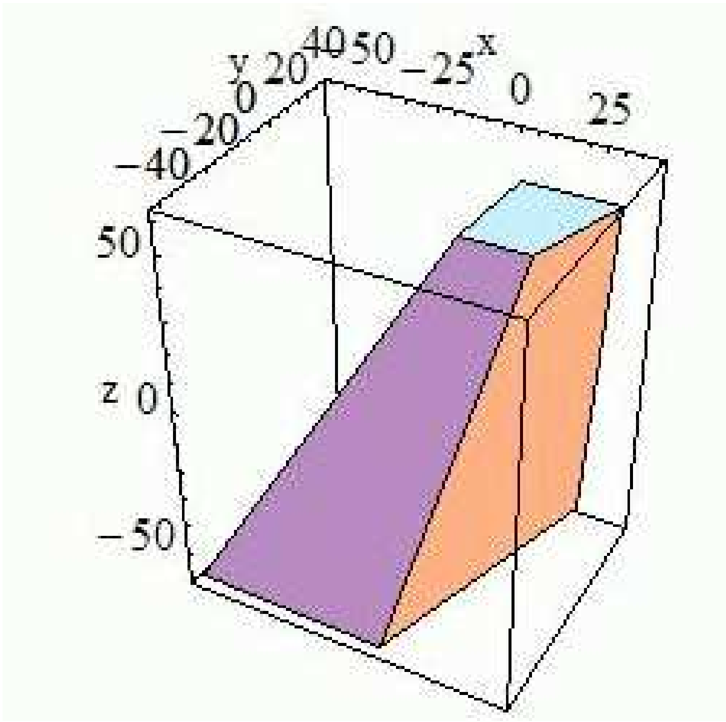


3. EC still at $\theta = 25^\circ$ to a perpendicular to the beamline and the same distance from the beam.

4. EC farther downstream from the target center than CLAS6. Normal distance from target center to upstream face is 7217.23 mm (JJ Roberts and D. Kashy). See `http://clasweb.jlab.org/wiki/index.php/Clas12_EC#Geometry`

Adding the EC to the CLAS12 Simulation

5. To add a new volume (the EC) to gemc:
   (a) Generate the parameters for a generic trapezoid (G4Trap) in Geant4.
   (b) Insert into a perl script (see `gemc/production/database_io/clas12/geo/ec/ec_build.pl`).
   (c) Execute '`go_tables user_geometry`' which reads the geometry values in `ec_build.pl` and inserts the values in the `user_geometry` database.
   (d) Use the `-DATABASE=user_geometry` option when running gemc.
   (e) BEWARE! Spaces versus tabs can muck up the mysql database.

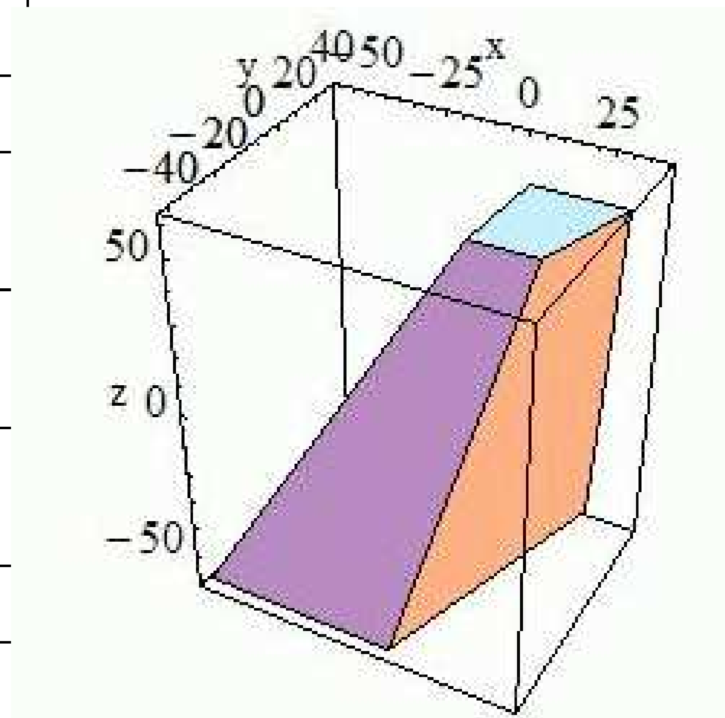


6. Treating each active layer in the EC as a single trapezoid. The strips will be defined during digitization.

Adding the EC to the CLAS12 Simulation

7. Geant4 G4trap geometry parameters:

pDx1	Half x length of the side at y=-pDy1 of the face at -pDz	pDx2	Half x length of the side at y=+pDy1 of the face at -pDz
pDz	Half z length	pTheta	Polar angle of the line joining the centres of the faces at -/+pDz
pPhi	Azimuthal angle of the line joining the centre of the face at -pDz to the centre of the face at +pDz	pDx3	Half x length of the side at y=-pDy2 of the face at +pDz
pDy2	Half y length at +pDz		
pDx4	Half x length of the side at y=+pDy2 of the face at +pDz		
pAlp2	Angle relative to the y axis from the centre of the side (upper endcap)		
pAlp1	Angle with respect to the y axis from the centre of the side (lower endcap)		
pDy1	Half y length at -pDz		



Adding the EC to the CLAS12 Simulation

8. Add the EC banks to the simulations.

   (a) In `gemc/production/database_io/clas12/banks/` create `ec.bank` (see below) and edit `clas12_hits_def.txt`.

   (b) Execute `go_tables` and `go_hits` in the same area to make entries in `clas12_banks` database.

   (c) In `gemc/production/hitprocess/` create `EC_hitprocess.cc` and `EC_hitprocess.h` and edit `Clas12_HitProcess_MapRegister.cc` to include the EC.

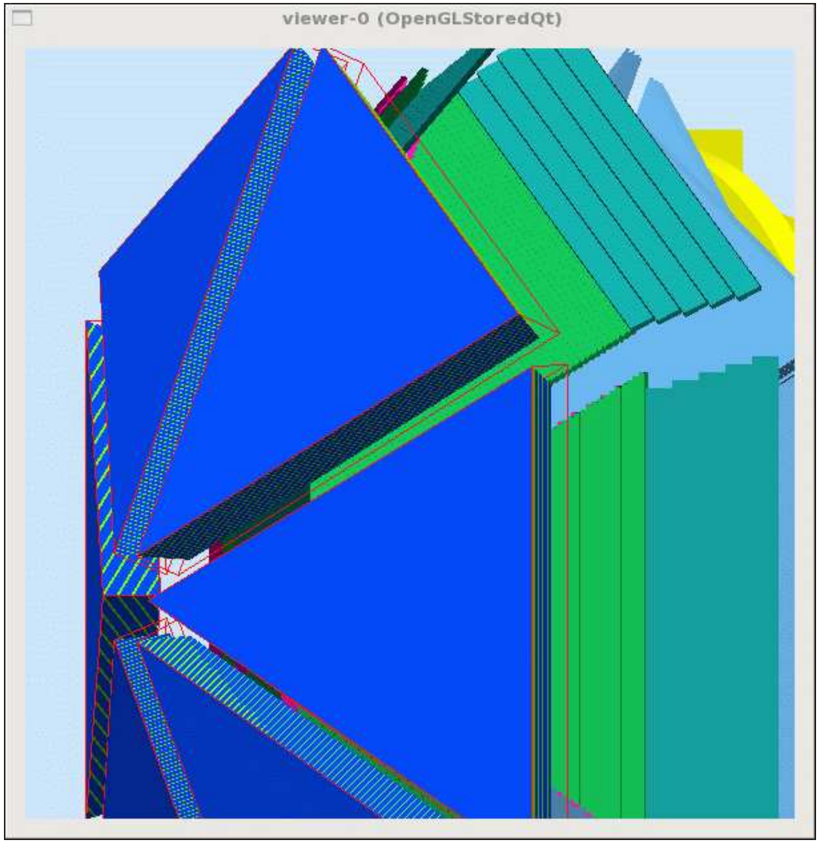
   (d) Recompile.

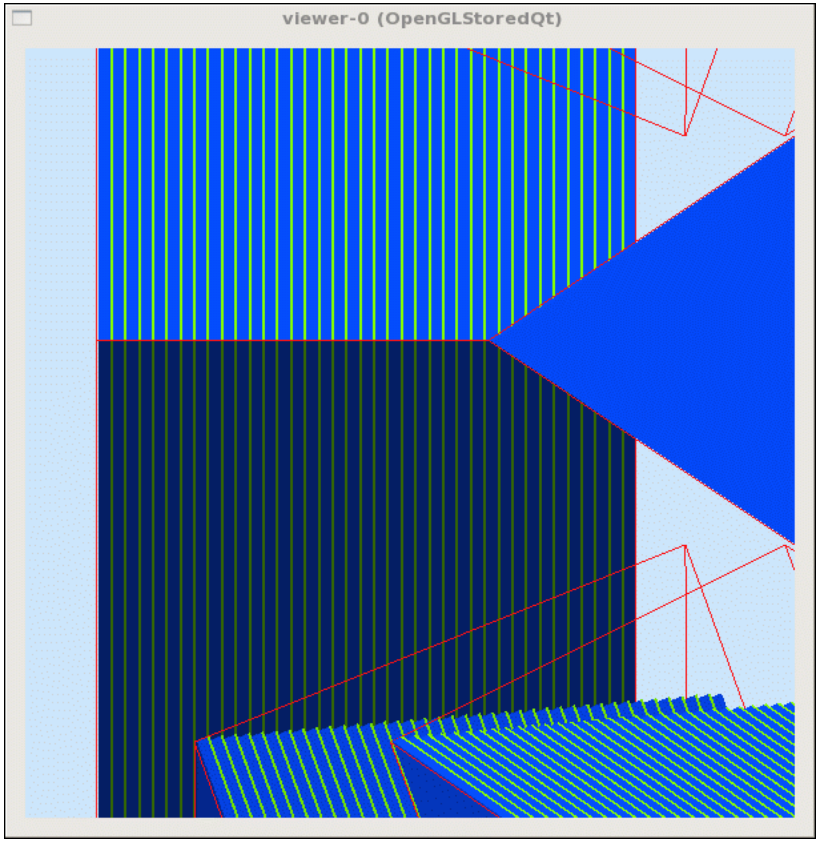
   (e) Talk to Mauri, fix your mistakes, and recompile...

ETot	1	1	1	Total Energy Deposited
<x>	2	1	1	Average global x position
<y>	3	1	1	Average global y position
<z>	4	1	1	Average global z position
<lx>	5	1	1	Average local x position
<ly>	6	1	1	Average local y position
<lz>	7	1	1	Average local z position
<t>	8	1	1	Average time
pid	9	1	1	Particle ID
vx	10	1	1	x coordinate of primary vertex
vy	11	1	1	y coordinate of primary vertex
vz	12	1	1	z coordinate of primary vertex
E	13	1	1	Energy of the track at the entrance point
mpid	14	1	1	Mother Particle ID
mvx	15	1	1	x coordinate of mother vertex
mvy	16	1	1	y coordinate of mother vertex
mvz	17	1	1	z coordinate of mother vertex
sector	18	1	1	sector
layer	19	1	1	layer
ADCL	20	0	1	ADCL
ADCR	22	0	1	ADCR

Adding the EC to the CLAS12 Simulation

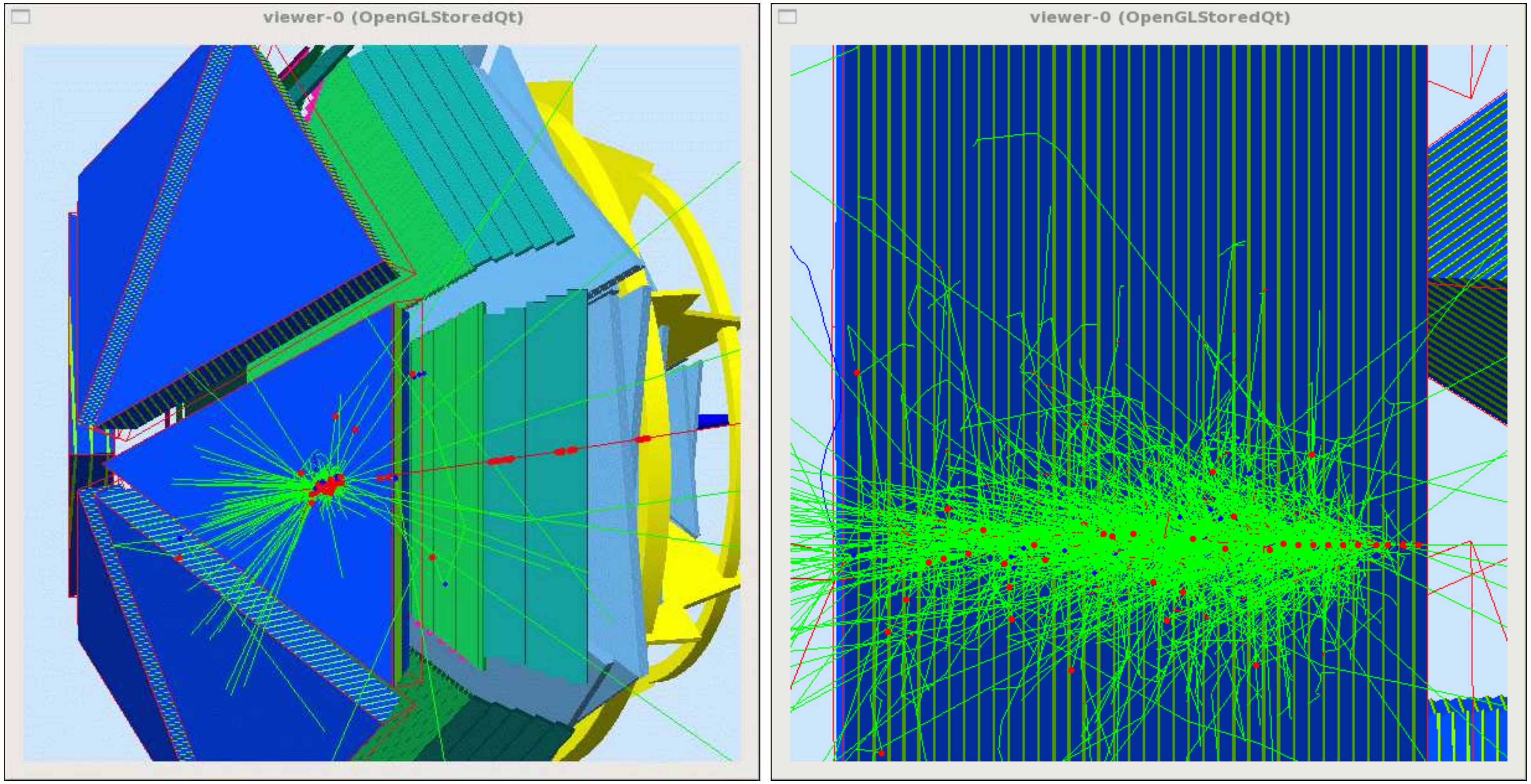
9. Images of the EC active region showing alternating layers of scintillator (blue) and lead (green).





Adding the EC to the CLAS12 Simulation

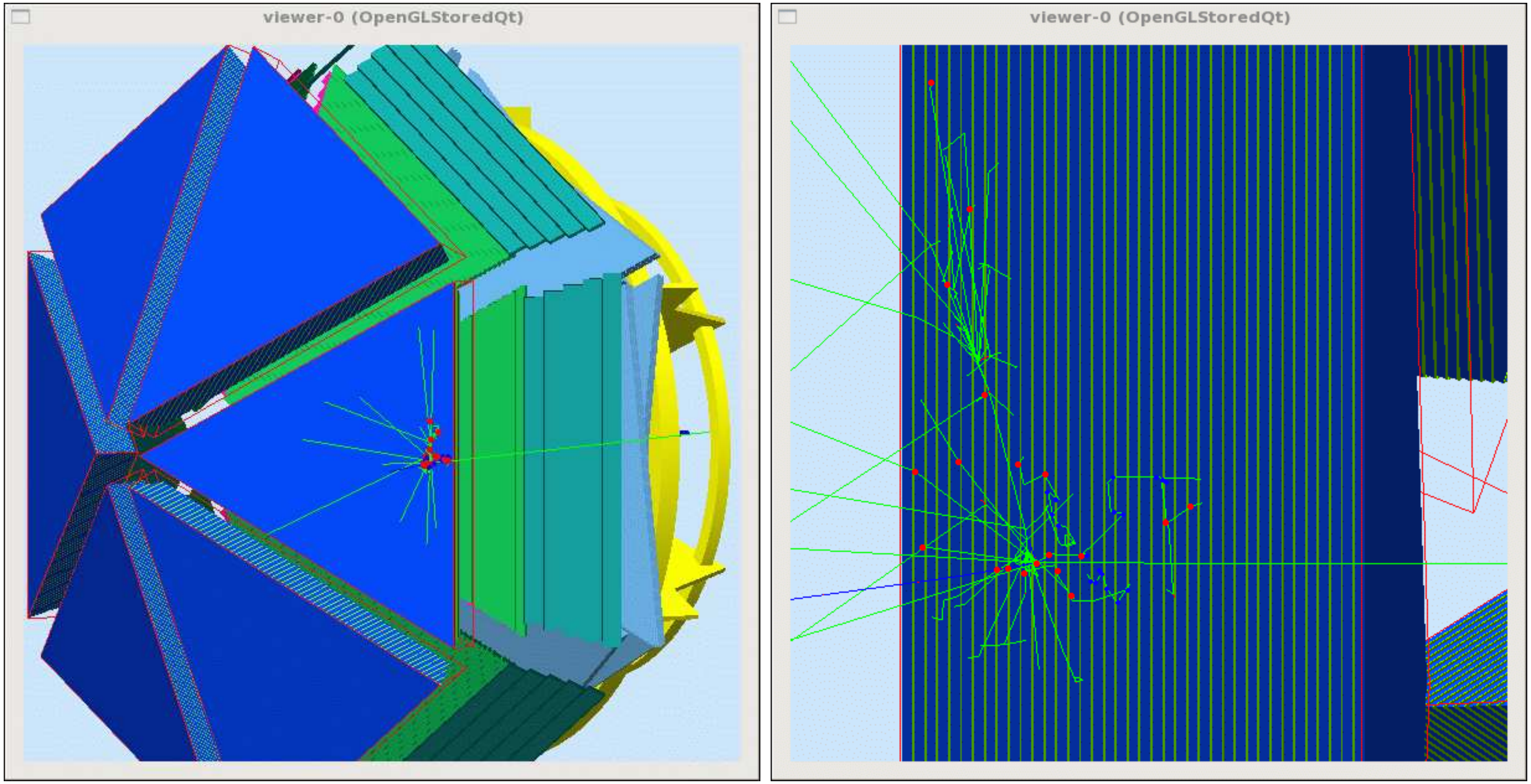
10. A 7.5-GeV electron event (left) and a close-up of the same event (right).



Red tracks are negatively charged particles; green are neutrals; blue are positive; red dots are above-threshold hits; blue dots are below-threshold hits.

Adding the EC to the CLAS12 Simulation

11. A 4.3-GeV neutron event (left) and a close-up of the same event (right).



Red tracks are negatively charged particles; green are neutrals; blue are positive; red dots are above-threshold hits; blue dots are below-threshold hits.

Adding the EC to the CLAS12 Simulation

12. Add the code to gemc to process an EC event; simplified version now.

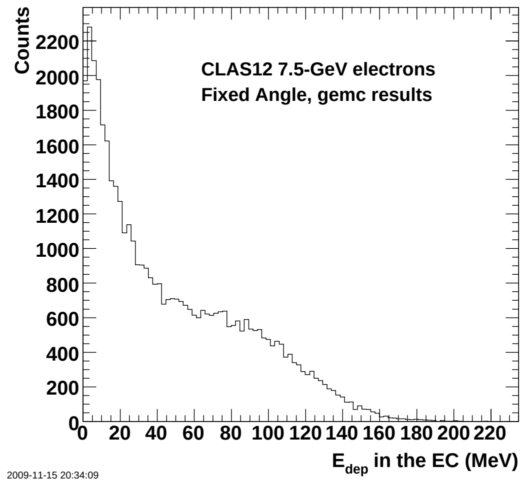
13. Modify and add functions to `gemc_evio2root` to include EC events in `gemc_evio2root/src/`:

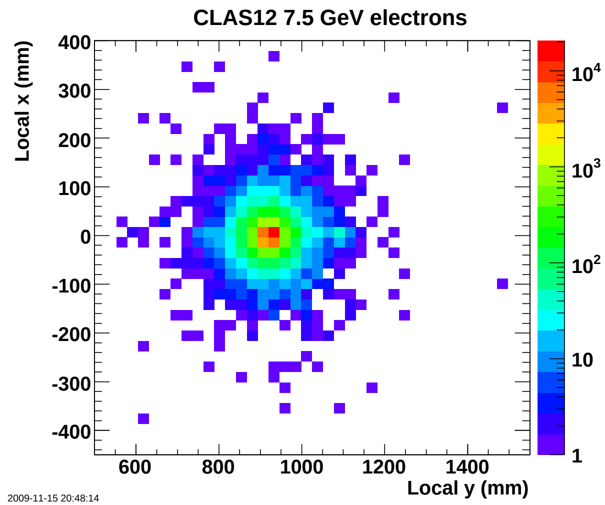
    (a) Create `ec_hit.cc` and `ec_hit.h` to handle Geant4 hit information.

    (b) Edit `root_tree.cc`, `root_tree.h`, and `gemc_evio2root.cc` to include EC information.

Defining an electron: $t < 50\ ns$, $N_{EC} < 40$, $E_{dep} > 5\ MeV$, $PID = 11$

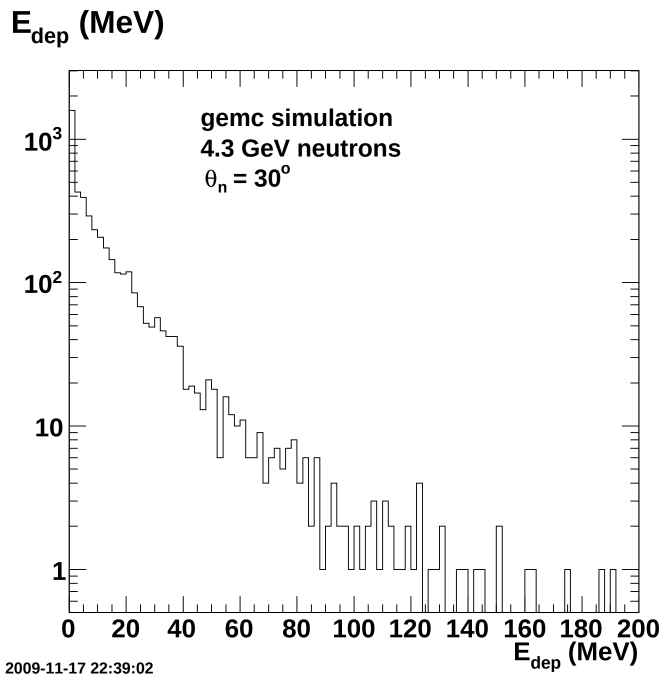
14. First histograms from gemc$\longrightarrow$ gemc_evio2root for electrons.

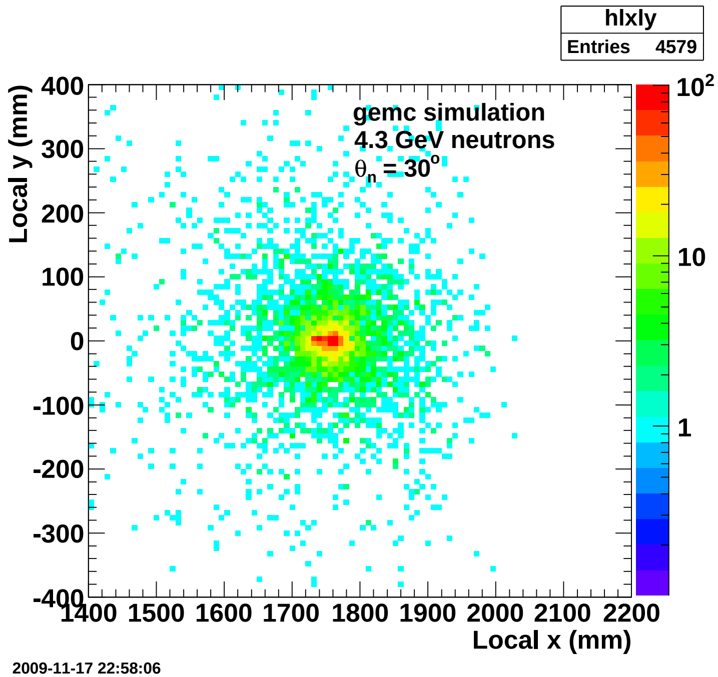




Adding the EC to the CLAS12 Simulation

15. First histograms from gemc→gemc_evio2root for neutrons.

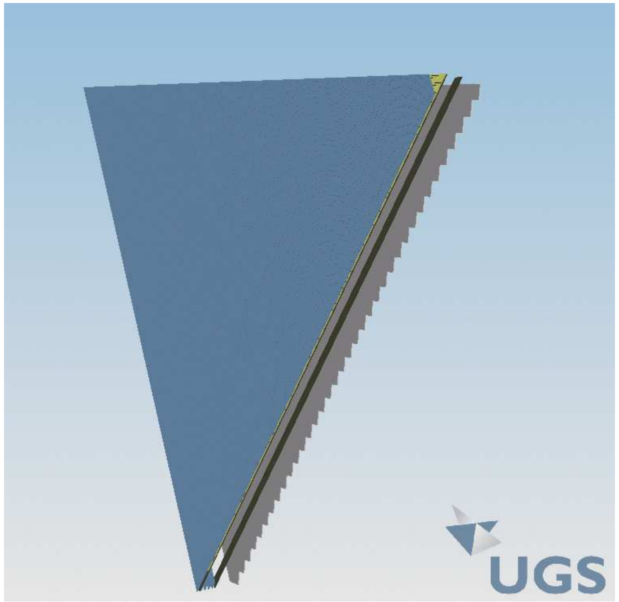




Defining a neutron: $t < 50\ ns$, $N_{EC} < 40$, $E_{dep} > 5\ MeV$, $PID = 2112$

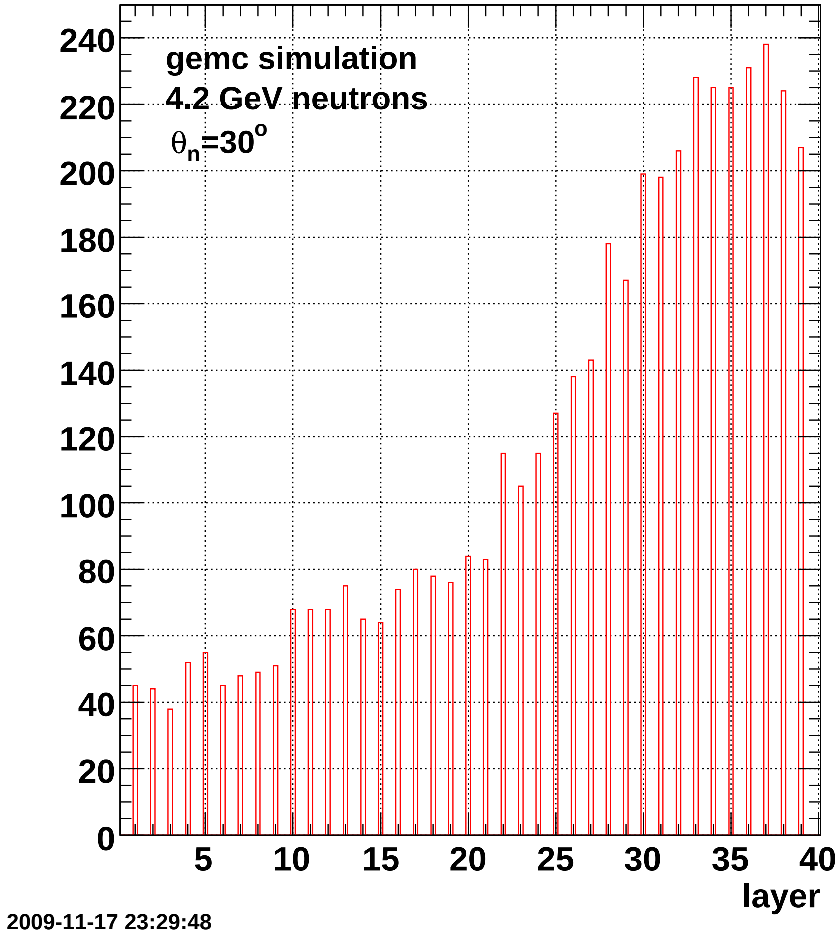
Plans

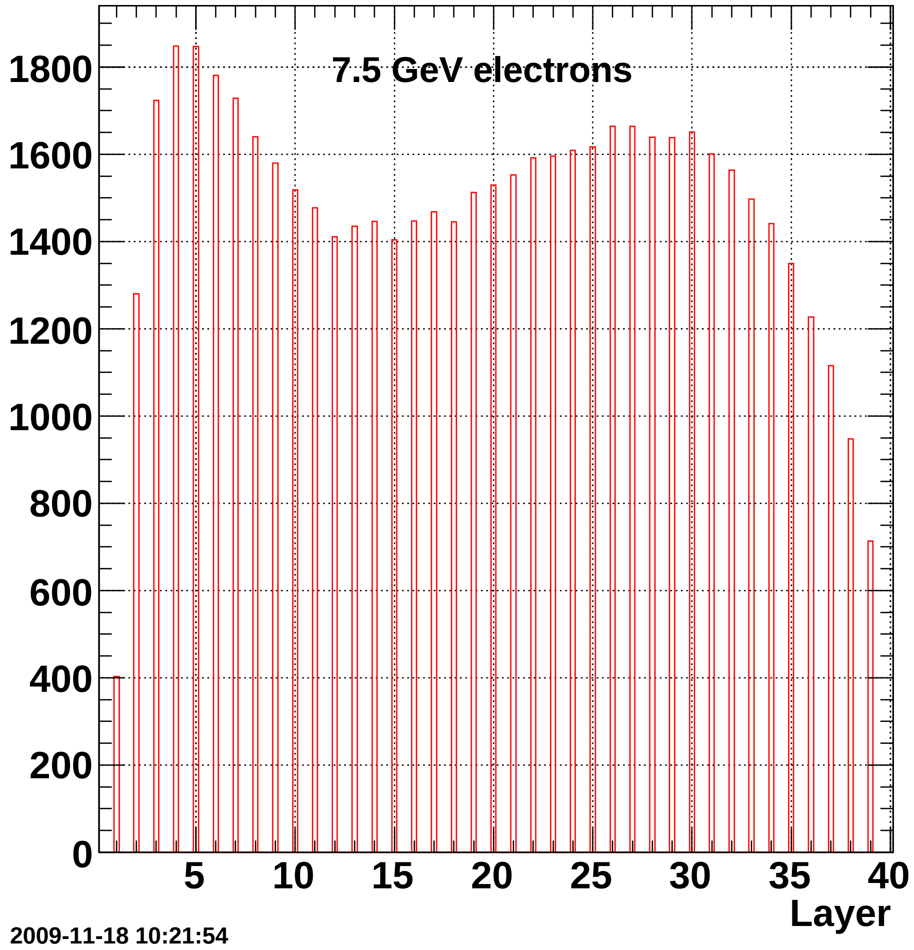
1. EC digitization.
2. Testing.
3. Implement the EC in Socrat (electron reconstruction code) and add(?) to reconstruction algorithm.
4. Extract preliminary neutron efficiency of EC and compare with E5 results.
5. Simulate $ep \rightarrow e'\pi^+n$ reaction.
6. More complete EC simulation (cover plates, ...).
7. Add PCAL (Mike Woods).



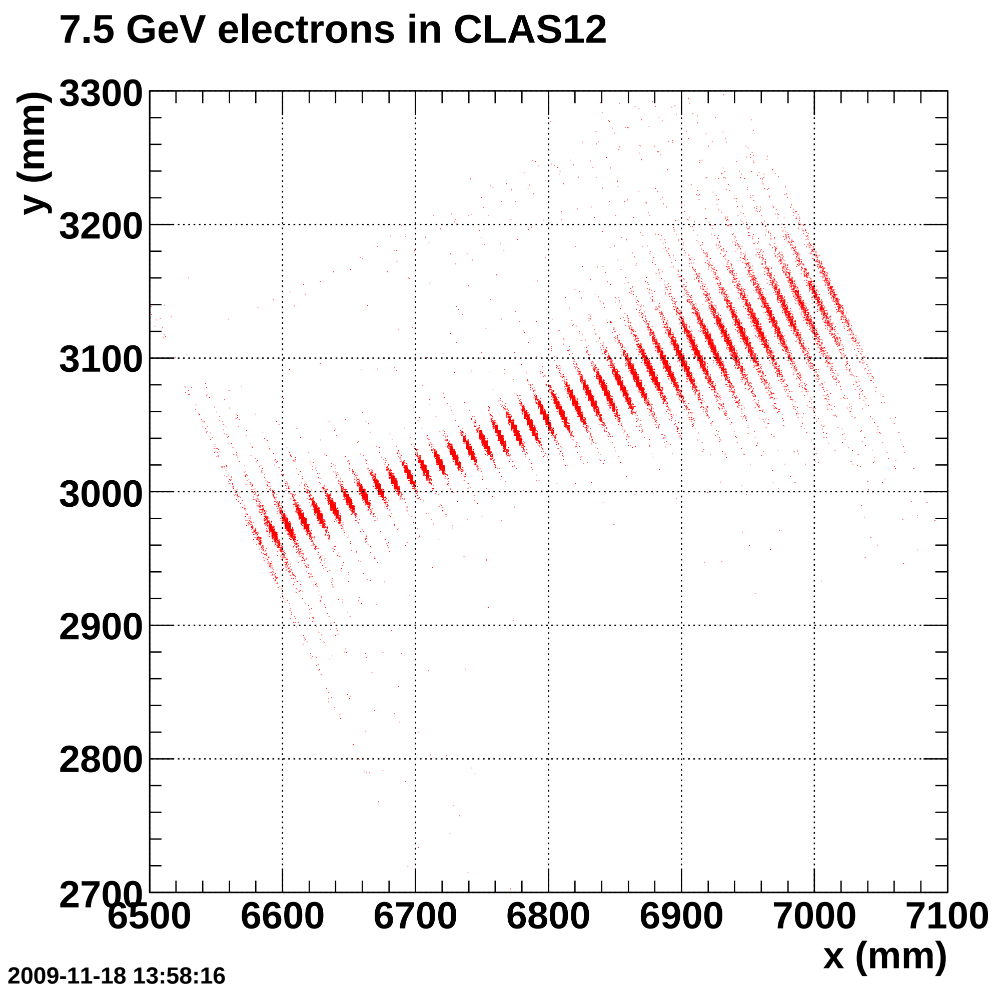
Additional Slides

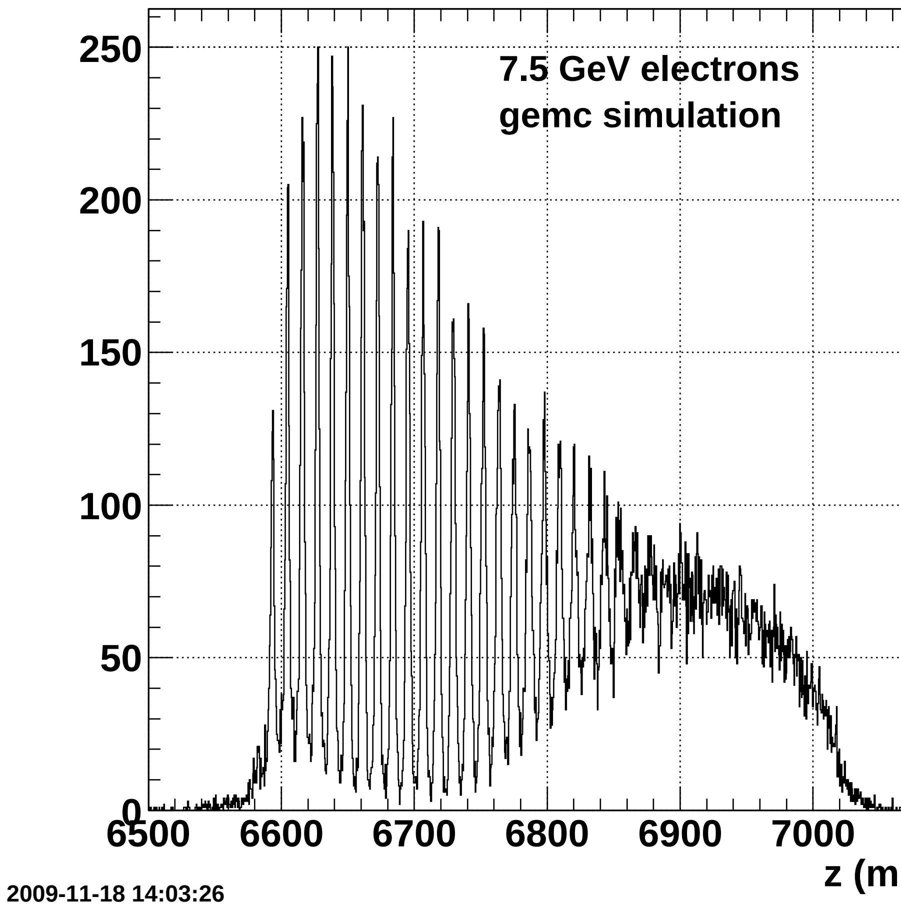
Distribution of Layers





Distribution of electrons within the EC





clas12_hits_def.txt

Generated	1	Generated Particles Information	0.0*MeV	0*ns
CTOF	50	paddle 60	200.0*KeV	4*ns
CND	55	layer 4 paddle 30	200.0*KeV	4*ns
OTOF_1a	60	sector 6 paddle 30	200.0*KeV	4*ns
OTOF_1b	70	sector 6 paddle 120	200.0*KeV	4*ns
OTOF_2b	80	sector 6 paddle 20	200.0*KeV	4*ns
BST	400	superlayer 4 type 2 segment 24 module 3 strip 300	2.0*KeV	132*ns
FST	410	superlayer 3 type 2 segment 24 strip 300	2.0*KeV	132*ns
FMT	450	superlayer 3 type 2 segment 12 strip 1500	2.0*KeV	132*ns
BMT	460	superlayer 3 type 2 segment 6 strip 3000	2.0*KeV	132*ns
DC	500	sector 6 superlayer 6 layer 8 wire 120	0.1*KeV	500*ns
IC	600	ih 30 iv 30	0.1*KeV	500*ns
Bonus	700	bonus 4	0.1*KeV	0*ns
TREK	200	layer 60	0.001*KeV	1*ns
EC	300	sector 6 layer 64 paddle 64	1.0*KeV	200*ns

EC_hitprocess.cc

```
// %%%%%%%%%%
// G4 headers
// %%%%%%%%%%
#include "G4UnitsTable.hh"

// %%%%%%%%%%%%%
// gemc headers
// %%%%%%%%%%%%%
#include "EC_hitprocess.h"

PH_output EC_HitProcess :: ProcessHit(MHit* aHit, gemc_opts)
{
 PH_output out;
 out.identity = aHit->GetId();
 HCname = "EC Hit Process";
 // get layer and sector.
 int sectorint = out.identity[0].id;
 int layerint  = out.identity[1].id;
 double sector = (double)sectorint;
 double layer  = (double)layerint;

 //cout << "sector = " << sector << " layer = " << layer << endl;

 // %%%%%%%%%%%%%%%%%%%
 // Raw hit information
 // %%%%%%%%%%%%%%%%%%%
 int nsteps = aHit->GetPos().size();
```

EC_hitprocess.cc

```
// Get Total Energy deposited
double Etot = 0;
vector<G4double> Edep = aHit->GetEdep();
for(int s=0; s<nsteps; s++) Etot = Etot + Edep[s];

// average global, local positions of the hit
double x, y, z;
double lx, ly, lz;
x = y = z = lx = ly = lz = 0;
vector<G4ThreeVector> pos  = aHit->GetPos();
vector<G4ThreeVector> Lpos = aHit->GetLPos();

if(Etot>0)
for(int s=0; s<nsteps; s++)
{
   x  = x  +  pos[s].x()*Edep[s]/Etot;
   y  = y  +  pos[s].y()*Edep[s]/Etot;
   z  = z  +  pos[s].z()*Edep[s]/Etot;
   lx = lx + Lpos[s].x()*Edep[s]/Etot;
   ly = ly + Lpos[s].y()*Edep[s]/Etot;
   lz = lz + Lpos[s].z()*Edep[s]/Etot;
}
else
{
  x  = pos[0].x();
  y  = pos[0].y();
  z  = pos[0].z();
  lx = Lpos[0].x();
  ly = Lpos[0].y();
  lz = Lpos[0].z();
}
```

EC_hitprocess.cc

```
// average time
double time = 0;
vector<G4double> times = aHit->GetTime();
for(int s=0; s<nsteps; s++) time = time + times[s]/nsteps;

// Energy of the track
double Ene = aHit->GetE();

out.raws.push_back(Etot);
out.raws.push_back(x);
out.raws.push_back(y);
out.raws.push_back(z);
out.raws.push_back(lx);
out.raws.push_back(ly);
out.raws.push_back(lz);
out.raws.push_back(time);
out.raws.push_back((double) aHit->GetPID());
out.raws.push_back(aHit->GetVert().getX());
out.raws.push_back(aHit->GetVert().getY());
out.raws.push_back(aHit->GetVert().getZ());
out.raws.push_back(Ene);
out.raws.push_back((double) aHit->GetmPID());
out.raws.push_back(aHit->GetmVert().getX());
out.raws.push_back(aHit->GetmVert().getY());
out.raws.push_back(aHit->GetmVert().getZ());

out.raws.push_back(sector);
out.raws.push_back(layer);
```

EC_hitprocess.cc

```
// %%%%%%%%%%%%
 // Digitization
 // %%%%%%%%%%%%

//  // Get the paddle length: in TOF paddles are boxes, it’s the x
//  double length = aHit->GetDetector().dimensions[0];

//  // Distances from left, right
//  double dLeft  = length + lx;
//  double dRight = length - lx;

 // dummy formulas for now, parameters could come from DB
 int ADCL = 1;
 int ADCR = 1;

 // speed of light is 30 cm/s
 // int TDCL = (int) (100*(time/ns + dLeft/cm/30.0));
//  int TDCR = (int) (100*(time/ns + dRight/cm/30.0));

 out.dgtz.push_back(ADCL);
 out.dgtz.push_back(ADCR);
//  out.dgtz.push_back(TDCL);
//  out.dgtz.push_back(TDCR);

 return out;
}

vector<identifier>  EC_HitProcess :: ProcessID(vector<identifier> id, G4Step* aStep, detector Detector, g
{
 return id;
}
```

EC_build.pl

```
#!/usr/bin/perl -w

use strict;

use lib ("$ENV{GEMC}/database_io");
use geo;
use geo qw($pi);

my $envelope = 'EC';
my $file     = 'EC.txt';

my $rmin = 1;
my $rmax = 1000000;

my %detector = ();    # hash (map) that defines the gemc detector
$detector{"rmin"} = $rmin;
$detector{"rmax"} = $rmax;

use Getopt::Long;
use Math::Trig;

my $inches   = 2.54;
my $NUM_BARS = 10;
```

EC_build.pl

```
# Mother Volume

$detector{"name"}        = "EC";
$detector{"mother"}      = "sector";
$detector{"description"} = "Forward Calorimeter";
$detector{"pos"}         = "0*mm 2156.71*mm 2217.23*mm";
$detector{"rotation"}    = "25*deg 0*deg 0*deg";
$detector{"color"}       = "ff1111";
$detector{"type"}        = "G4Trap";
$detector{"dimensions"}  = "241.4295*mm 0*deg 0*deg 2033.99*mm 0.001  2082.67*mm  0*deg  2033.99*mm 0.001
$detector{"material"}    = "Air";
$detector{"mfield"}      = "no";
$detector{"ncopy"}       = 1;
$detector{"pMany"}       = 1;
$detector{"exist"}       = 1;
$detector{"visible"}     = 1;
$detector{"style"}       = 0;
$detector{"sensitivity"} = "no";
$detector{"hit_type"}    = "";
$detector{"identifiers"} = "";

print_det(\%detector, $file);
```

EC_build.pl

```
# now start to do the alternating layers of scintillator and lead. Set up inputs first.

my $i;
my $subname;
my $submother = "EC";;
my $description;
my $pos;
my $rotation = "0*deg 0*deg 0*deg";
my $color = "0147FA";
my $type = "G4Trap";
my $dimensions;
my $material ="Air";
my $mfield = "no";
my $ncopy = 1;
my $pMany = 1;
my $exist = 1;
my $visible = 1;
my $style = 1;
my $sensitivity = "no";
my $hit_type = "";
my $identifiers = "";
my $dlead = 2.381;
my $dscint= 10.0;
my $tantheta = 1.95325;
my $nlayers = 39;
my $totaldepth = $nlayers*($dscint+$dlead);
my $z0scint;
my $z0lead;
my $pDy1;
my $pDx2;
my $pDx4;
```

EC_build.pl

```
# a scintillator layer first.

$i = 1;
$subname = "make_EClayerScint${i}";
$description ="Forward Calorimeter scintillator layer ${i}";
$z0scint = -$totaldepth/2 + ($i - 1)*($dscint + $dlead) + $dscint/2;
$pos = "0*mm 0.0*mm ${z0scint}*mm";

$detector{"name"}        = $subname;
$detector{"mother"}      = "EC";
$detector{"description"} = $description;
$detector{"pos"}         = $pos;
$detector{"rotation"}    = $rotation;
$detector{"color"}       = "0147FA";
$detector{"type"}        = "G4Trap";
$detector{"dimensions"}  = "5.0*mm 0*deg 0*deg 1864.65*mm 0.001  1909.28*mm  0*deg  1864.65*mm 0.001  190
$detector{"material"}    = "Scintillator";
$detector{"mfield"}      = "no";
$detector{"ncopy"}       = 1;
$detector{"pMany"}       = 1;
$detector{"exist"}       = 1;
$detector{"visible"}     = 1;
$detector{"style"}       = 1;
$detector{"sensitivity"} = "EC";
$detector{"hit_type"}    = "EC";
$detector{"identifiers"} = "sector ncopy 0 layer manual $i paddle manual 1";

print_det(\%detector, $file);
```

EC_build.pl

```
# loop over the remaining layers.

for ($i = 2; $i < $nlayers+1; $i++) {
# lead layer
    $subname = "EClayerLead${i}";
    $description = "Forward Calorimeter lead layer ${i}";
    $z0lead = -$totaldepth/2 + ($i - 2)*($dscint + $dlead) + $dscint/2 + ($dscint+$dlead)/2;
    $pos = "0*mm 0.0*mm ${z0lead}*mm";
    $pDy1 = (1899.56 + 4.5419*($i-1) - (-1829.74 - 4.3708*($i-1)))/2;
    $pDx2 = (1899.56 + 4.5419*($i-1) - (-1829.74 - 4.3708*($i-1)))/$tantheta;
    $pDx4 = $pDx2;

    $detector{"name"}        = $subname;
    $detector{"mother"}      = $submother;
    $detector{"description"} = $description;
    $detector{"pos"}         = $pos;
    $detector{"rotation"}    = $rotation;
    $detector{"color"}       = "7CFC00";
    $detector{"type"}        = "G4Trap";
    $detector{"dimensions"}  = "1.1935*mm 0*deg 0*deg  ${pDy1}*mm 0.001*mm  ${pDx2}*mm  0*deg  ${pDy1}*mm
    $detector{"material"}    = "Lead";
    $detector{"mfield"}      = "no";
    $detector{"ncopy"}       = 1;
    $detector{"pMany"}       = 1;
    $detector{"exist"}       = 1;
    $detector{"visible"}     = 1;
    $detector{"style"}       = 1;
    $detector{"sensitivity"} = "no";
    $detector{"hit_type"}    = "";
    $detector{"identifiers"} = "";
```

EC_build.pl

```
# scintillator layer

    $subname = "make_EClayerScint${i}";
    $description ="Forward Calorimeter scintillator layer ${i}";
    $z0scint = -$totaldepth/2 + ($i - 1)*($dscint + $dlead) + $dscint/2;
    #print("z0scint = $z0scint  \n");
    $pos = "0*mm 0.0*mm ${z0scint}*mm";

    $detector{"name"}        = $subname;
    $detector{"mother"}      = $submother;
    $detector{"description"} = $description;
    $detector{"pos"}         = $pos;
    $detector{"rotation"}    = $rotation;
    $detector{"color"}       = "0147FA";
    $detector{"type"}        = "G4Trap";
    $detector{"dimensions"}  = "5.0*mm 0*deg 0*deg ${pDy1}*mm 0.001  ${pDx2}*mm  0*deg  ${pDy1}*mm 0.001
    $detector{"material"}    = "Scintillator";
    $detector{"mfield"}      = "no";
    $detector{"ncopy"}       = 1;
    $detector{"pMany"}       = 1;
    $detector{"exist"}       = 1;
    $detector{"visible"}     = 1;
    $detector{"style"}       = 1;
    $detector{"sensitivity"} = "EC";
    $detector{"hit_type"}    = "EC";
    $detector{"identifiers"} = "sector ncopy 0 layer manual $i paddle manual 1";

    print_det(\%detector, $file);
   }
```