Recent Developments for the CLAS12 Simulation


CLAS12 Software Group

Outline:
1. Physics Motivation.
2. Neutron Detection Efficiency in the CLAS12 Outer TOF.
3. Implementing the CLAS6 EC in CLAS12.
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 $E_{dep} > 5 \text{ MeV}$. 

![Graph showing neutron counts vs. p_n (GeV/c) with black and red histograms, indicating found and reconstructed neutrons, respectively.]

2009-11-13 10:30:05

CLAS12 Workshop - Nov 18, 2009 – p. 3/2
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

   \[ y_-(L) = A_- + B_- (L - 1) \quad (\text{vertex}) \quad y_+(L) = A_+ + B_+ (L - 1) \quad (\text{top}) \]

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

   \[ y + x \tan \theta = y_-(L_-) \quad y - x \tan \theta = y_-(L) \]

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:

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETot</td>
<td>1</td>
</tr>
<tr>
<td>&lt;x&gt;</td>
<td>2</td>
</tr>
<tr>
<td>&lt;y&gt;</td>
<td>3</td>
</tr>
<tr>
<td>&lt;z&gt;</td>
<td>4</td>
</tr>
<tr>
<td>&lt;lx&gt;</td>
<td>5</td>
</tr>
<tr>
<td>&lt;ly&gt;</td>
<td>6</td>
</tr>
<tr>
<td>&lt;lz&gt;</td>
<td>7</td>
</tr>
<tr>
<td>&lt;t&gt;</td>
<td>8</td>
</tr>
<tr>
<td>pid</td>
<td>9</td>
</tr>
<tr>
<td>vx</td>
<td>10</td>
</tr>
<tr>
<td>vy</td>
<td>11</td>
</tr>
<tr>
<td>vz</td>
<td>12</td>
</tr>
<tr>
<td>E</td>
<td>13</td>
</tr>
<tr>
<td>mpid</td>
<td>14</td>
</tr>
<tr>
<td>mvx</td>
<td>15</td>
</tr>
<tr>
<td>mvy</td>
<td>16</td>
</tr>
<tr>
<td>mvz</td>
<td>17</td>
</tr>
<tr>
<td>sector</td>
<td>18</td>
</tr>
<tr>
<td>layer</td>
<td>19</td>
</tr>
<tr>
<td>ADCL</td>
<td>20</td>
</tr>
<tr>
<td>ADCR</td>
<td>22</td>
</tr>
</tbody>
</table>
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.
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.

14. First histograms from gemc → gemc_evio2root for electrons.
Adding the EC to the CLAS12 Simulation

15. First histograms from gemc→gemc_evio2root for neutrons.

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

gemc simulation
4.2 GeV neutrons
θ_θ=30°

7.5 GeV electrons

layer

2009-11-17 23:29:48

2009-11-18 10:21:54
Distribution of electrons within the EC

7.5 GeV electrons in CLAS12

7.5 GeV electrons gemc simulation
<table>
<thead>
<tr>
<th>Generated</th>
<th>1</th>
<th>Generated Particles Information</th>
<th>0.0 * MeV</th>
<th>0 * ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTOF</td>
<td>50</td>
<td>paddle 60</td>
<td>200.0 * KeV</td>
<td>4 * ns</td>
</tr>
<tr>
<td>CND</td>
<td>55</td>
<td>layer 4 paddle 30</td>
<td>200.0 * KeV</td>
<td>4 * ns</td>
</tr>
<tr>
<td>OTOF_1a</td>
<td>60</td>
<td>sector 6 paddle 30</td>
<td>200.0 * KeV</td>
<td>4 * ns</td>
</tr>
<tr>
<td>OTOF_1b</td>
<td>70</td>
<td>sector 6 paddle 120</td>
<td>200.0 * KeV</td>
<td>4 * ns</td>
</tr>
<tr>
<td>OTOF_2b</td>
<td>80</td>
<td>sector 6 paddle 20</td>
<td>200.0 * KeV</td>
<td>4 * ns</td>
</tr>
<tr>
<td>BST</td>
<td>400</td>
<td>superlayer 4 type 2 segment 24 module 3 strip 300</td>
<td>2.0 * KeV</td>
<td>132 * ns</td>
</tr>
<tr>
<td>FST</td>
<td>410</td>
<td>superlayer 3 type 2 segment 24 strip 300</td>
<td>2.0 * KeV</td>
<td>132 * ns</td>
</tr>
<tr>
<td>FMT</td>
<td>450</td>
<td>superlayer 3 type 2 segment 12 strip 1500</td>
<td>2.0 * KeV</td>
<td>132 * ns</td>
</tr>
<tr>
<td>BMT</td>
<td>460</td>
<td>superlayer 3 type 2 segment 6 strip 3000</td>
<td>2.0 * KeV</td>
<td>132 * ns</td>
</tr>
<tr>
<td>DC</td>
<td>500</td>
<td>sector 6 superlayer 6 layer 8 wire 120</td>
<td>0.1 * KeV</td>
<td>500 * ns</td>
</tr>
<tr>
<td>IC</td>
<td>600</td>
<td>ih 30 iv 30</td>
<td>0.1 * KeV</td>
<td>500 * ns</td>
</tr>
<tr>
<td>Bonus</td>
<td>700</td>
<td>bonus 4</td>
<td>0.1 * KeV</td>
<td>0 * ns</td>
</tr>
<tr>
<td>TREK</td>
<td>200</td>
<td>layer 60</td>
<td>0.001 * KeV</td>
<td>1 * ns</td>
</tr>
<tr>
<td>EC</td>
<td>300</td>
<td>sector 6 layer 64 paddle 64</td>
<td>1.0 * KeV</td>
<td>200 * ns</td>
</tr>
</tbody>
</table>
// %%%%%%%%%%%
// 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();
// 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();
  }
// 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);
```
// %%%%%%%%%%%%%%%%%
// 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, g4part* aPart)
{
    return id;
}
```
#!/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 genc detector
$detector{"rmin"} = $rmin;
$detector{"rmax"} = $rmax;

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

my $inches = 2.54;
my $NUM_BARS = 10;
# 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);
# 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;
# 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 1909.28*mm";
$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 ncop 0 layer manual $i paddle manual 1";

print_det(\%detector, $file);
# 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"} = "";

    print_det(\%detector, $file);
}
# scintillator layer

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

$detector("name") = $subnetname;
$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);