

# Status on making GEMC compatible with LCSim

Ebrahim Ebrahim

UNH

July 7, 2011

The goal is to eventually be able to do tracking with gemc output.  
Why?

- provides an excellent test of software
- extends the functionality of gemc and makes it available as an alternative simulation
- gives me an opportunity to familiarize myself with lcsim tracking

# Filling up the SimTrackerHits

Right now I can fill everything in a SimTrackerHit except for an MCParticle.

- Not straight-forward because the information gemc extracts from a G4Event is different from SLIC
- You can tell gemc how to behave each event by writing a custom hit\_process class
- GEMC output for tracker hits now *identical* to SLIC output.

# Importing Detector Geometry from LCSim

It's important to ensure that the detector geometry in gemc is identical to the one lcsim knows about.

- Wrote a script that reads an LCDD file and writes its contents to a gemc database table
- preserves all information except magnetic fields
- performs rotation to gemc coordinates
- *Now we can simulate with identical geometries*

# Importing Detector Geometry from LcSim

The screenshot displays the **gemc** software interface. On the left, a terminal window shows the execution of the program with the following output:

```
@ebrahim-laptop: ~/HPS/gemc_dir/gemc/prc
Terminal Help
Settings >> Beam polarization: 0%
Settings >> Polarization Direction: (theta
>> Initializing Event Action...
>> Initializing Stepping Activ
>> Init
>> Init
tic Field
struction:
t File: >
>> Exec
>> Exec
g: /vis/s
g: /vis/o
g: /vis/s
g: /vis/v
g: /vis/s
g: /vis/v
g: /vis/s
```

The main window shows a 3D visualization of detector components. A grey, multi-layered detector structure is visible at the bottom left. A series of red rectangular frames are arranged in a diagonal path, leading to a green rectangular plane at the top right. A blue background is visible behind these elements.

On the right side, the **gemc** control panel is shown. It includes tabs for **Primary Particle**, **Primary Beam**, and **Secondary Beam**. The **Primary Particle** is set to **e-**. Below this, there are sliders for **Value** and **Dispersion**. The **Vertex Values** section displays the following parameters:

(x,y,z):	(0, -10, 50) mm
radius:	0 mm
delta z:	0 mm

Below the vertex values, there are more sliders for **Value** and **Dispersion**, with labels for **radius** and **dvz**. The **Number of Events** is set to **1**. At the bottom, there is a **Beam On** button and an **Exit** button.

Now that we have the same geometry and (almost) the same output, we can try to do tracking with gemc.

- We could try to work around using MCParticles in lcsim (seed tracking driver)
- Maybe it would be easier to get gemc to keep track of the MCParticle Tree.

- If it's worth moving forward with this project, I need to figure out in which direction I'm going to take it.
- If not, the LCDD-GEMC conversion will probably prove useful anyway.