LCDG4 Status

Dhiman Chakraborty, Guilherme Lima,
Jeremy McCormick, Vishnu Zutshi

NORTHERN ILLINOIS
UNIVERSITY

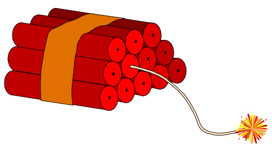
LC Simulations Workshop
Argonne, June 02-05, 2004

Outline

- Overview
- LCDG4 features
  - XML geometry representation
  - SIO contents
- Certification
- Processing times
- Available datasets
- Remaining issues
- Summary

What is LCDG4

- A Geant4-based detector simulator to support detector R&D for the Linear Collider
- Alternatives: Mokka, LCS or Gismo (old, unsupported)
- LCDG4 features
  - Input format: binary STDHEP
  - Output format: only ***.sio*** for now (Gismo compatible)
    .lcio is planed for the near future
  - Some detector geometries are implemented via XML geometry files (e.g. SD, LD, PD)



Simplistic geometry: cylinders, disks and cones only, no cracks, limited representation of support structure

Some LCDG4-specific features

- Development bias towards HCal (also ECal)
  other detectors should work fine, but no active studies by the developers
- Correct MC particle hierarchy, even when V0s and hyperon decays are forced in event generation
- Energies deposited in absorbers are easily available for analysis (ASCII format only)
- Non-projective geometries also available for barrel HCal (forked versions), output into SIO files
- Analysis code and documentation available from CVS (new! Preliminary versions...) and http://nicadd.niu.edu/~lima/lcdg4/

Geometry info in XML

An example: Silicon detector (SDJan03.xml)

```
<?xml version="1.0"  ?>
<!-- Comments
...    -->
<lcdparm>
  <global file="SDJan03.xml" />
  <physical_detector topology="silicon" id = "SDJan03" >

    <volume id = "PIPE_INNER" rad_len_cm="0.00047" inter_len_cm="0.00040">
    ...
    <complex_volume id = "VERTEX_BARREL">
    ...
  </physical_detector>

  <proc_parm>
    <cal_smear em_energy = "0.12 0.01" had_energy = "0.50 0.02"
               em_position = "1.0 0.0" had_position = "5.0 0.0" />
  </proc_parm>
</lcdparm>
```

Geometry info in XML

An example: Hadronic calorimeter barrel in XML (a simple volume). Dimensions are in centimeters.

```
...
<volume id="HAD_BARREL" rad_len_cm="1.133" inter_len_cm="0.1193">
  <tube>
    <barrel_dimensions inner_r = "144.0" outer_z = "286.0" />
    <layering n="34">
      <slice material = "Stainless_steel" width = "2.0" />
      <slice material = "Polystyrene" width = "1.0" sensitive = "yes“ />
    </layering>
    <segmentation cos_theta = "600" phi = "1200" />
  </tube>
  <calorimeter type="had" />
</volume>
...
```

Geometry info in XML

Another example: Vertex detector barrel in XML (a complex volume). Dimensions are in centimeters.

```
...
<complex_volume id = "VERTEX_BARREL">
      <volume id = "VERTEX_BARREL_1" rad_len_cm="0.0011" inter_len_cm="0.00022">
        <tube>
          <barrel_dimensions inner_r = "1.2" outer_z = "2.5" />
          <layering>
            <slice material = "Si" width = "0.01" sensitive = "yes" />
          </layering>
        </tube>
        <tracker />
      </volume >
      <volume id = "VERTEX_BARREL_2" rad_len_cm="0.0011" inter_len_cm="0.00022">
      ...
      <volume id = "VERTEX_BARREL_5" rad_len_cm="0.0011" inter_len_cm="0.00022">
      ...
    </complex_volume>
...
```

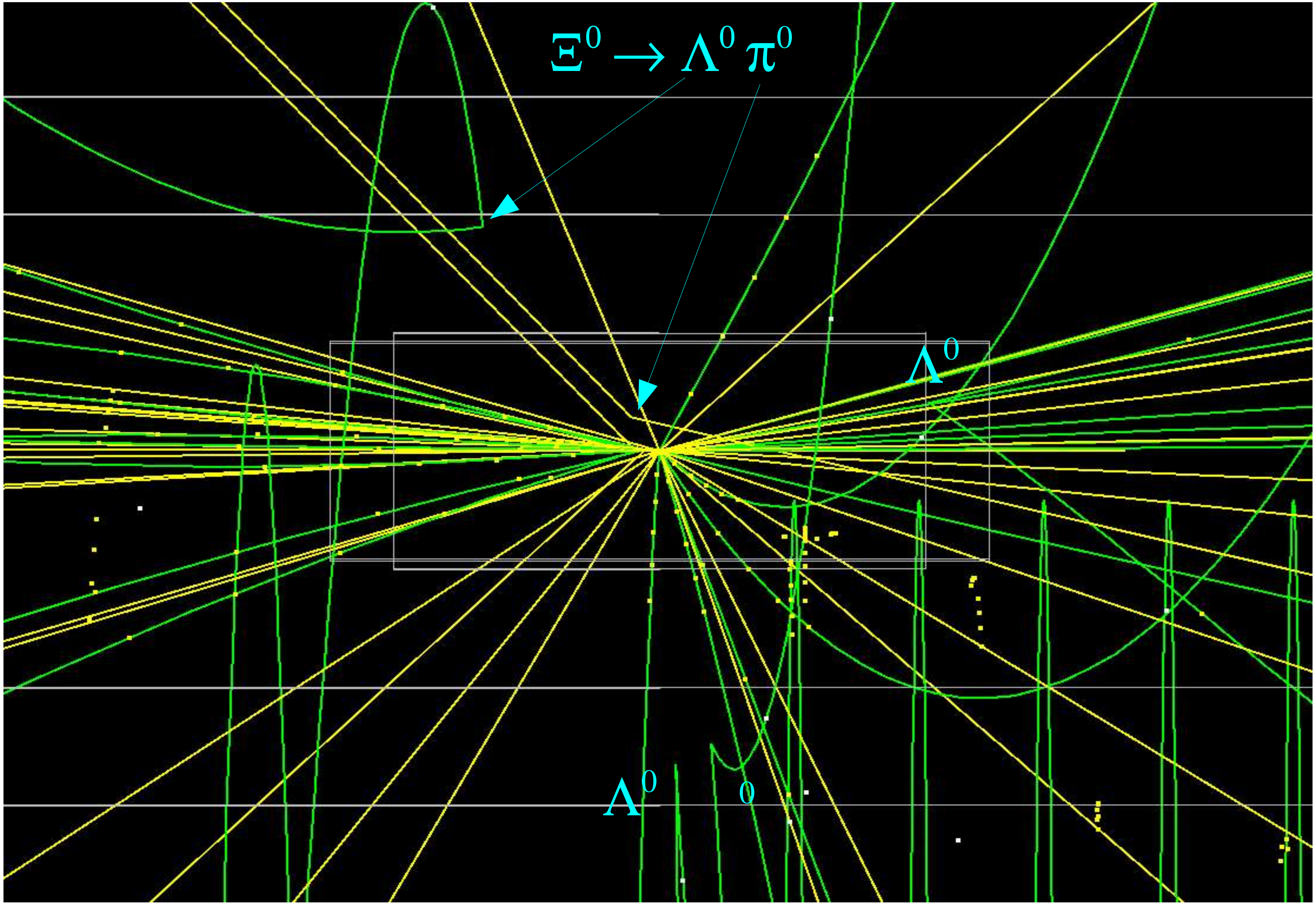
XML-based geometry description

- Flexible, very easy for minor changes (dimensions, materials, layering, segmentation), recompilation is not needed
- Deeper changes are error prone, not many checks for inconsistencies
- Detector construction implementation could be better!!
  (It currently depends too much on the format of the XML file)
- XML files may become large for complex geometries (could split into a top level + one file per subcomponent)
- Slow-access, as compared to a relational database access, but it only happens at initialization stage (not a big deal)

SIO output: general features

- SIO output contents: one particle collection and several hit collections (one collection per subdetector)
- Each hit points to the contributing particles (except tracker hits from calorimeter back-scatterings, as in Gismo)
- All secondaries above an energy threshold (now set at 1 MeV), except for shower secondaries, are saved in output with:
  - Particle id and status codes (generation and simulation)
  - Production momentum and ending position
  - Calorimeter entrance point: position and momentum
  - Pointers to parent particles (decay or interaction)

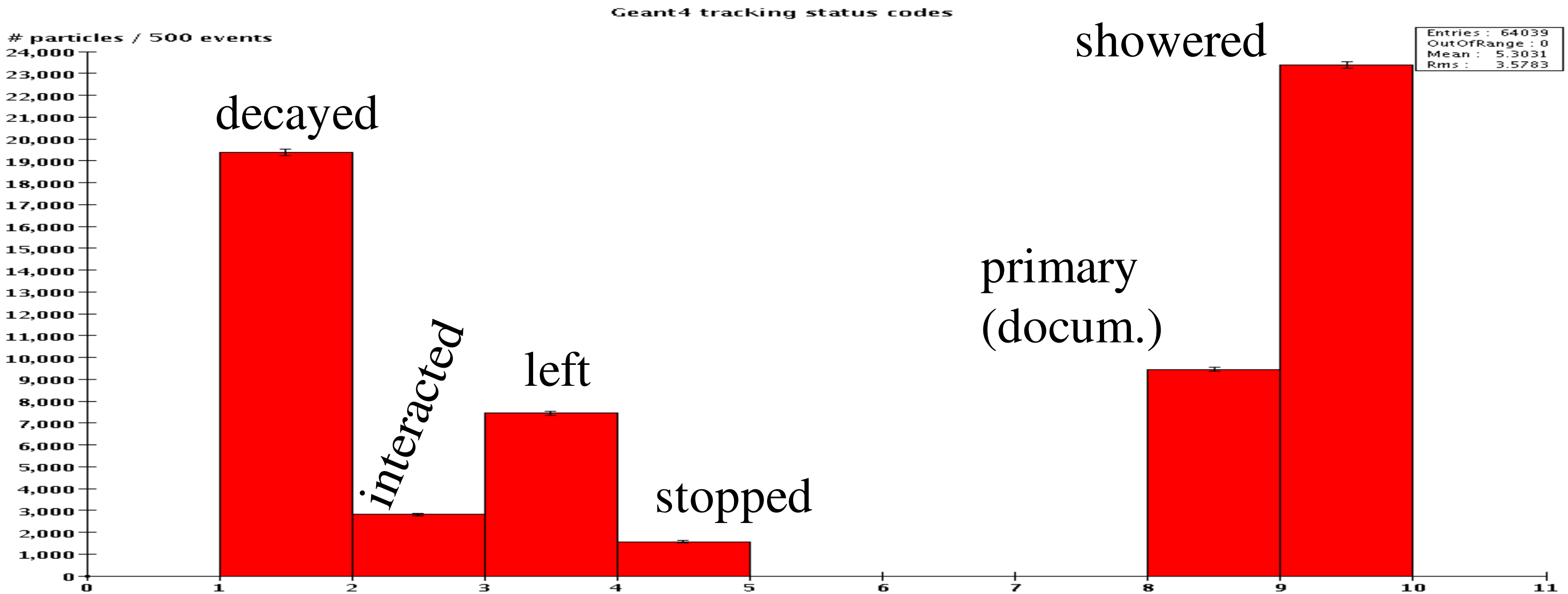
Zoom on the primary interaction



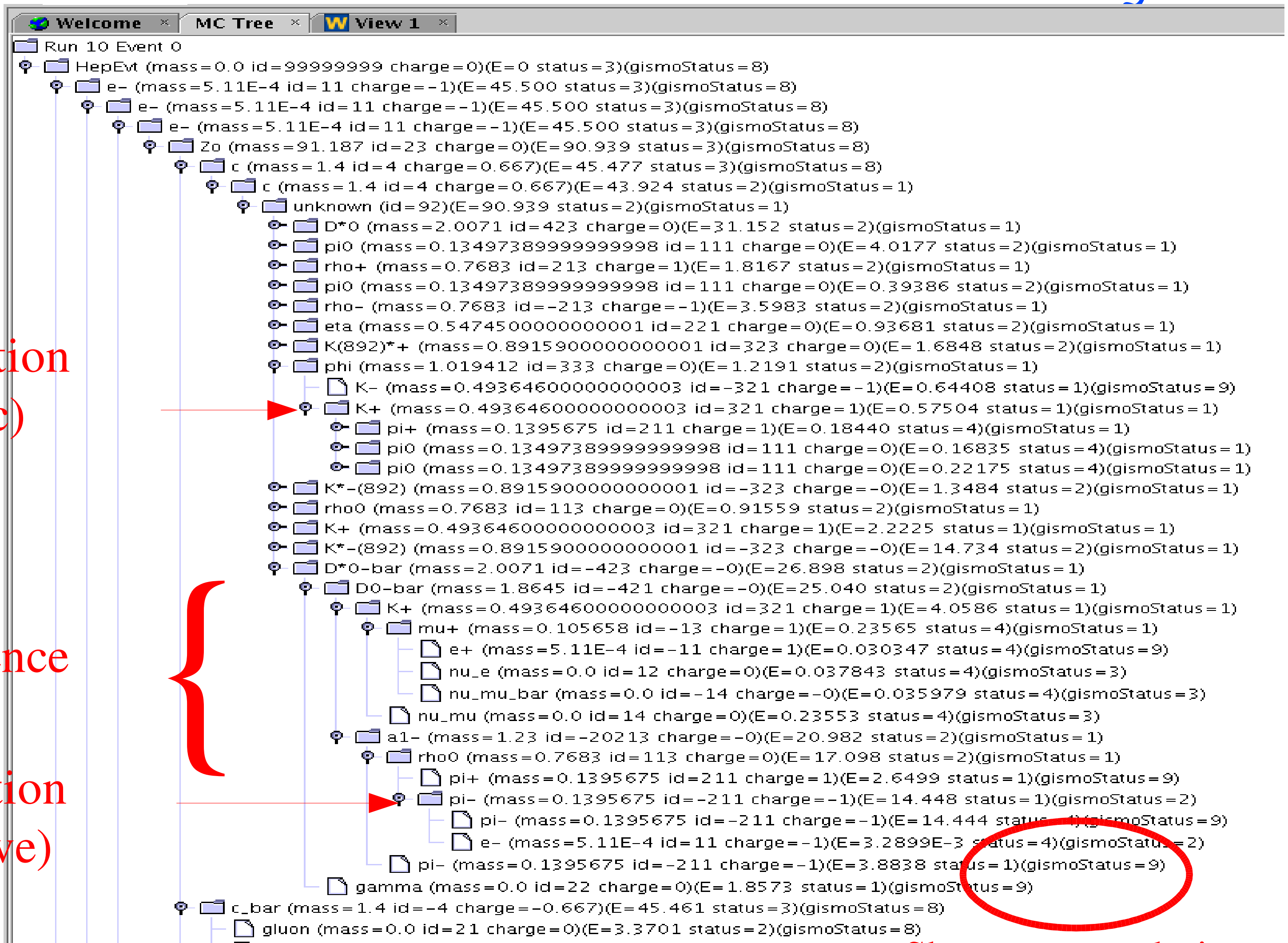
Simulation Status Codes

0 – active
1 – decayed
2 – interacted
3 – left detector
4 – stopped
5 – looping
6 – lost
7 – stuck
8 – primary
9 – showered
10 – maxsteps

(fully implemented, or not to be implemented)



Correct MCParticle hierarchy



pre-Cal interaction (inelastic)

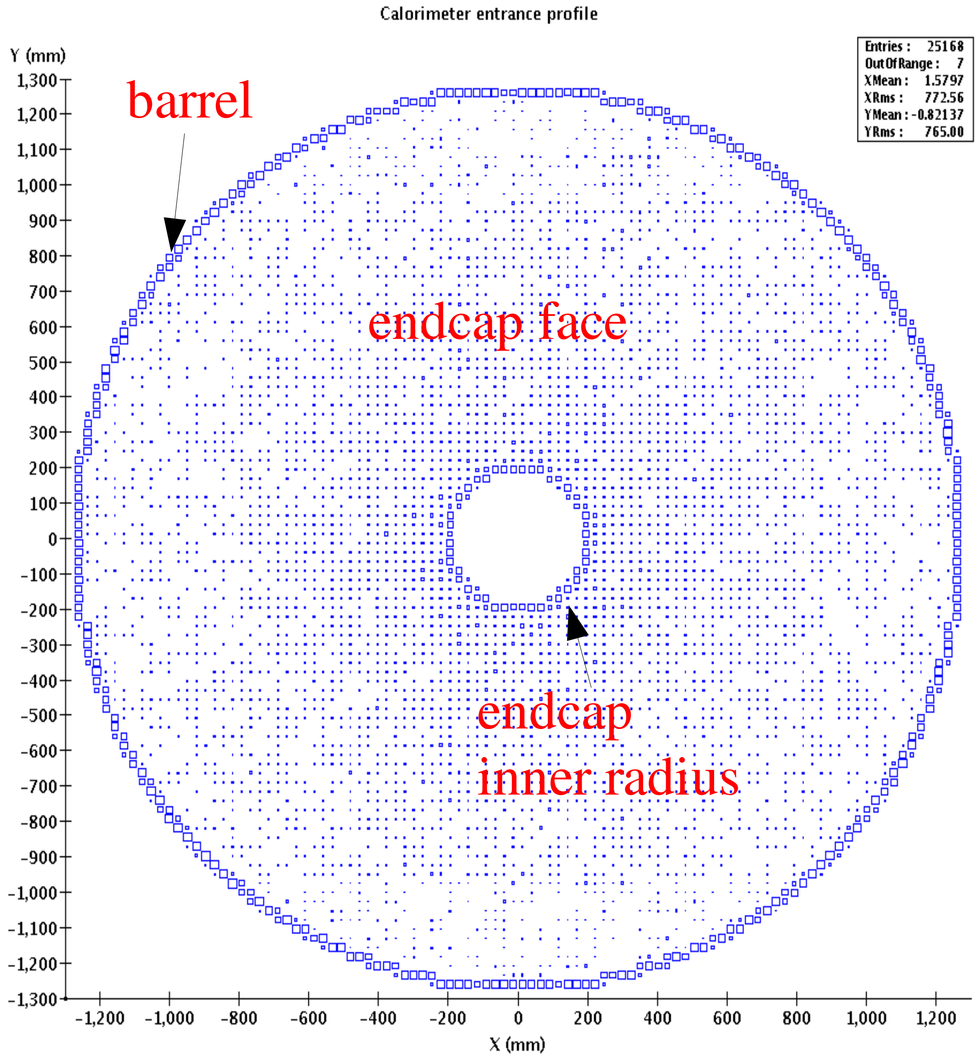
decay sequence

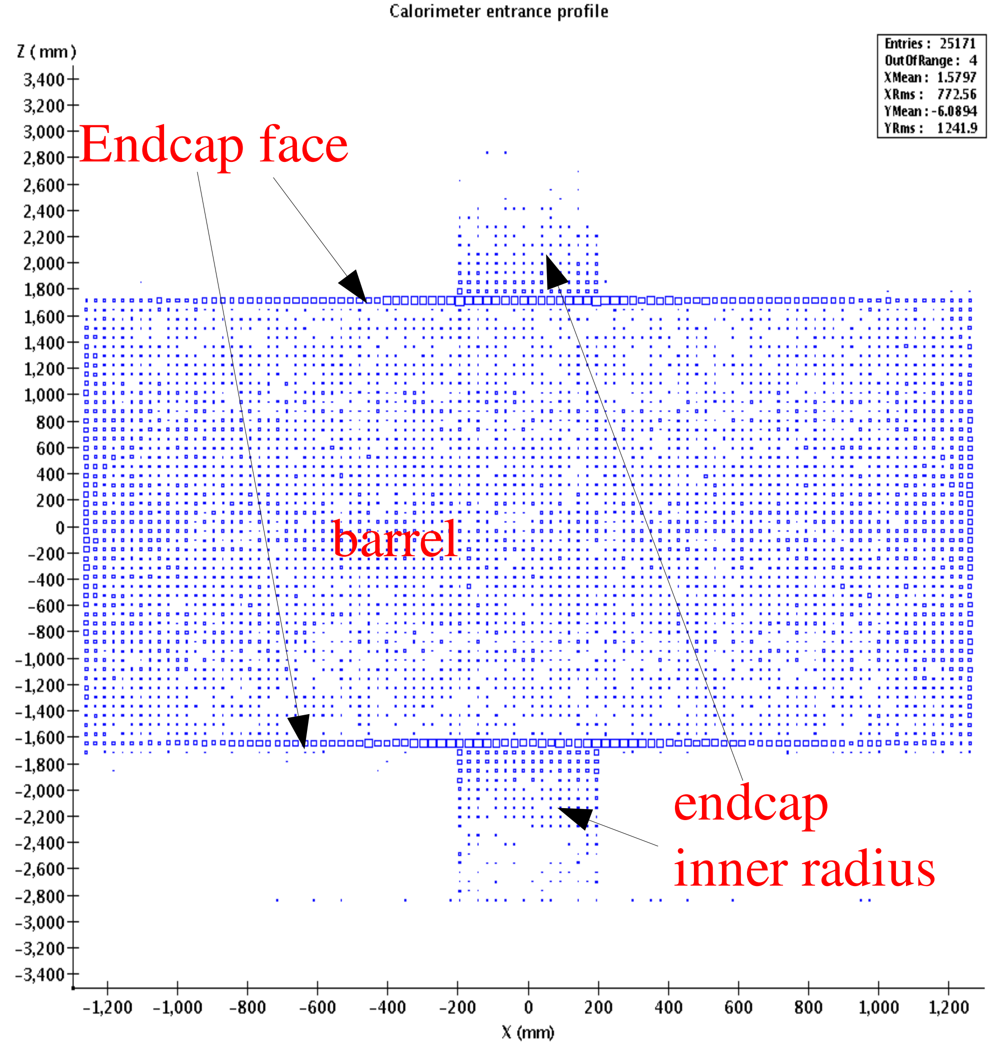
pre-Cal interaction (non-destructive)

Shower secondaries are not saved

nicadd

Cal entrance: position, momentum

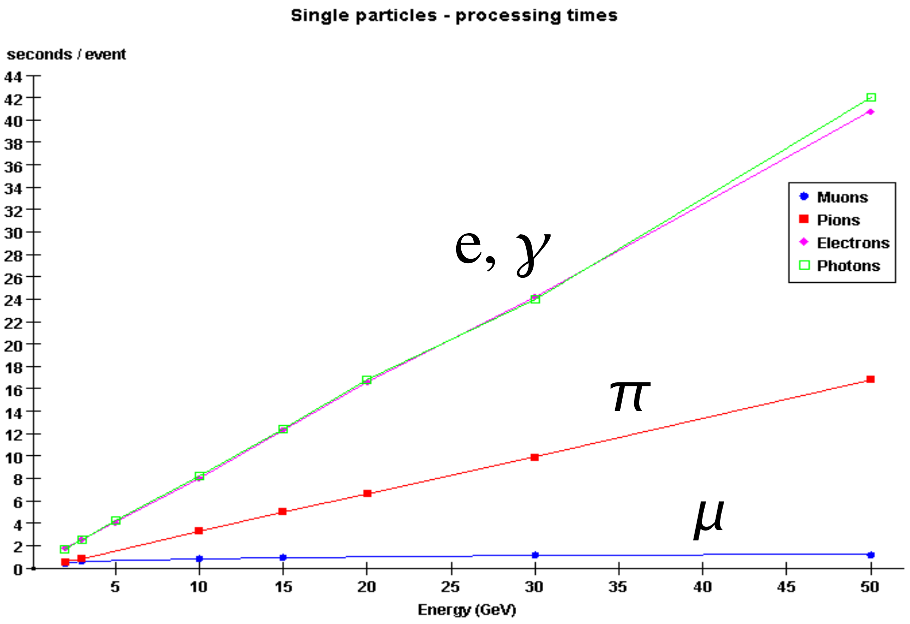




LCDG4 processing times

(in a 2.4 GHz CPU)

- Single particles:



- Physics events
  - Z to X @ 91 GeV: 0.55 min/evt
  - tt to X @ 350 GeV: 1.66 min/evt
  - ZH to Xbb @ 500 GeV: 2.33 min/evt
  - WW to qqbb @ 500 GeV: 2.22 min/evt

About ~15% faster after some memory debugging (valgrind)

Mokka and LCDMokka

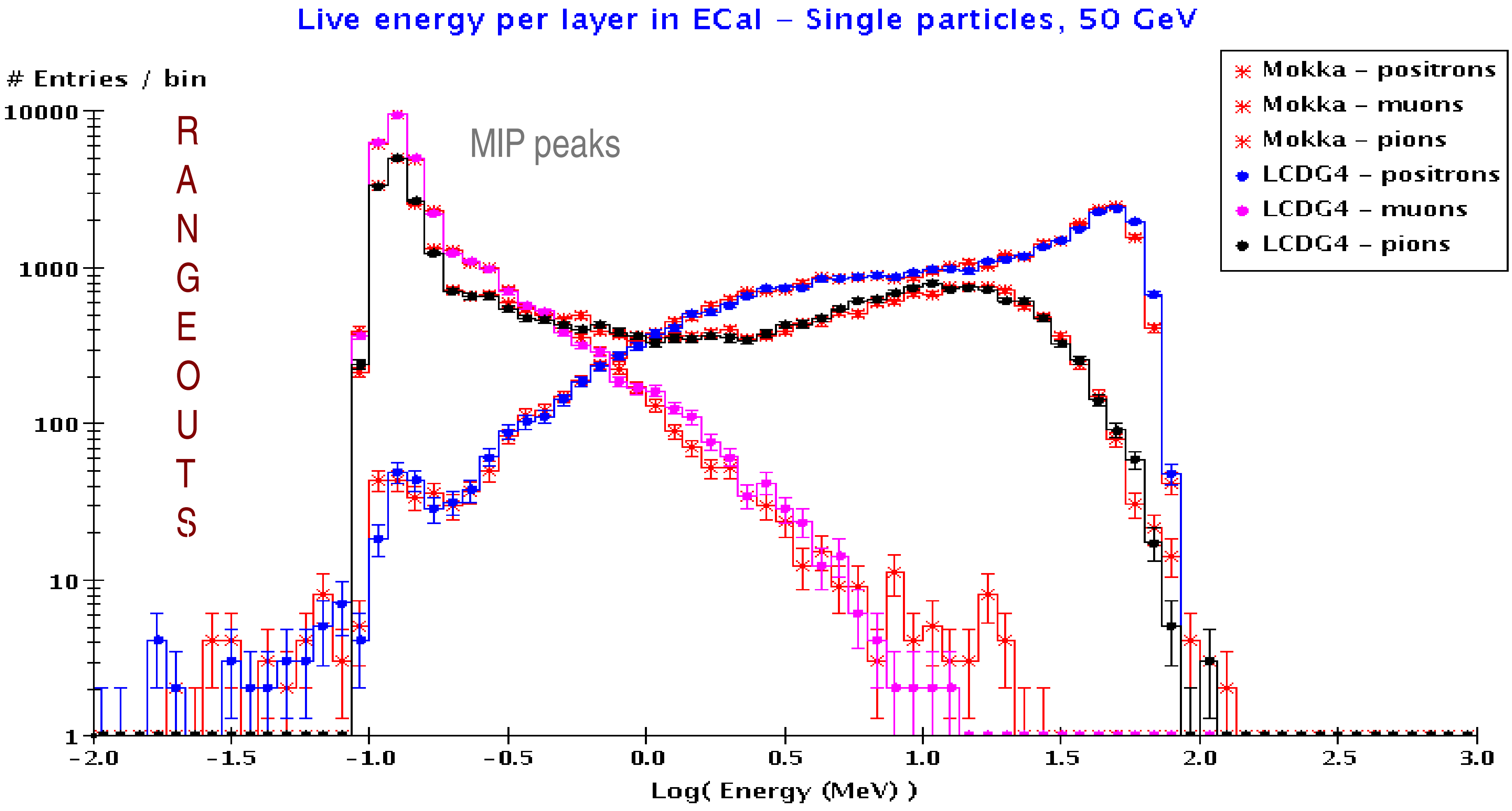
- Mokka is another Geant4-based simulation framework for Linear Collider R&D
- Detector geometry is described using a MySQL database
- Based on Tesla model, many other models and prototypes have been added into the geometry database
- Input: ASCII StdHEP / Ouptut: ASCII or LCIO
- For more info, please visit Mokka web site:

  http://polywww.in2p3.fr/geant4/tesla/www/mokka/mokka.html
- LCDMokka: XML capabilities into Mokka v01-05 (latest version is v02-03), while LCDG4 is not able to use MySQL geometry files (e.g. Tesla)
- Used LCDMokka for comparisons with LCDG4

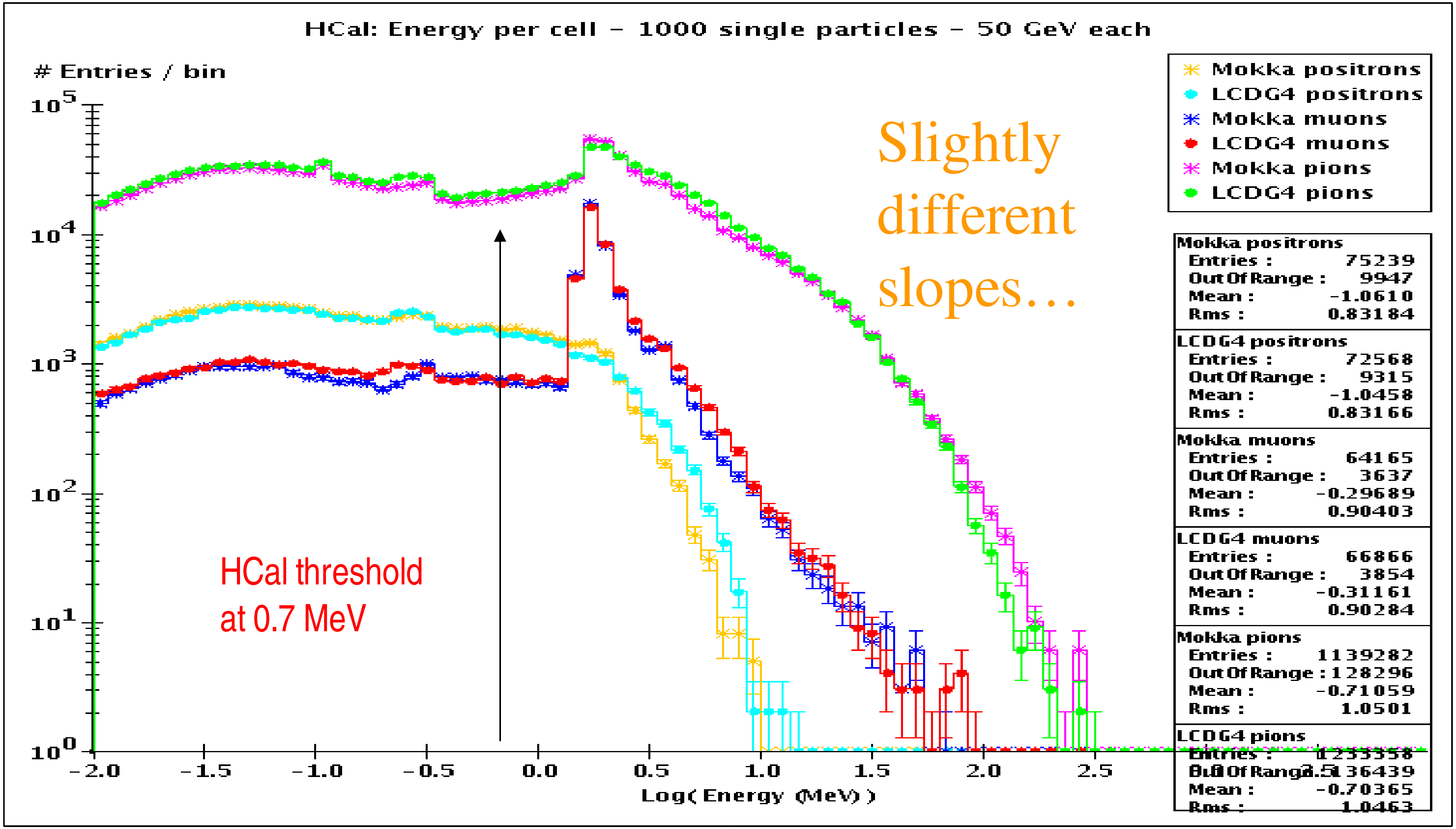
Fair comparison

- Geant4 version 5.2
- SDJan03 geometry
  (cylindrical layers with virtual cells)
- Physics list from Mokka v01.05
- Range cut of 0.1mm
- Identical I/O formats (binary stdhep input, text output) implemented into both simulators
- Same events processed in both detector simulators
  single particles: 50 GeV $e^{\pm}$, $\mu^{\pm}$, $\pi^{\pm}$, $\theta = 90°$, flat in φ
- Same materials in sub-detectors (look at X_0, λ_I)

Ecal: energies per layer



Hcal: energies per cell



LCDG4: Remaining issues

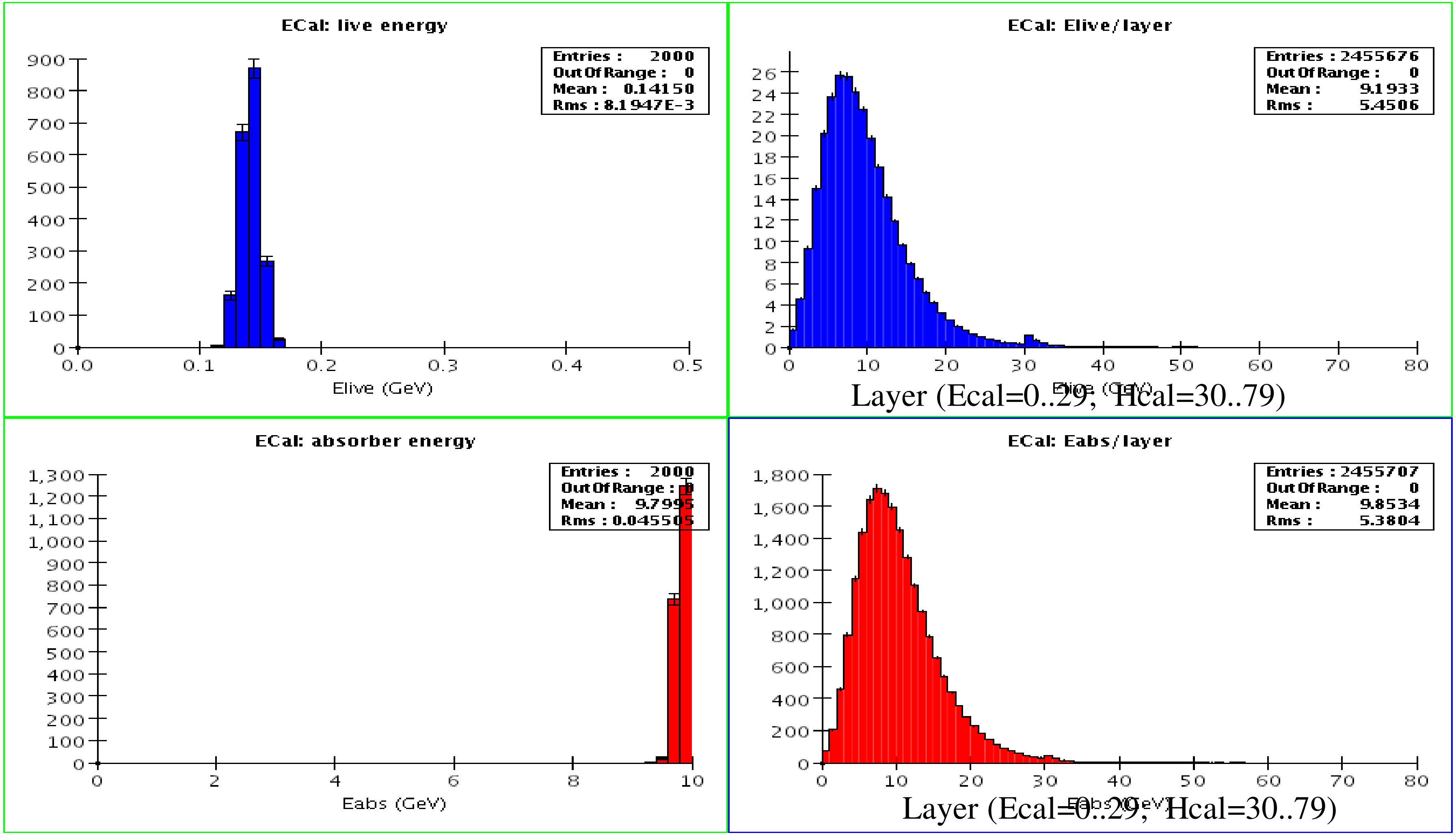
(1) Discrepancy in EM live energy (LCDG4 & Gismo)

(2) Discrepancy in number of calorimeter hits

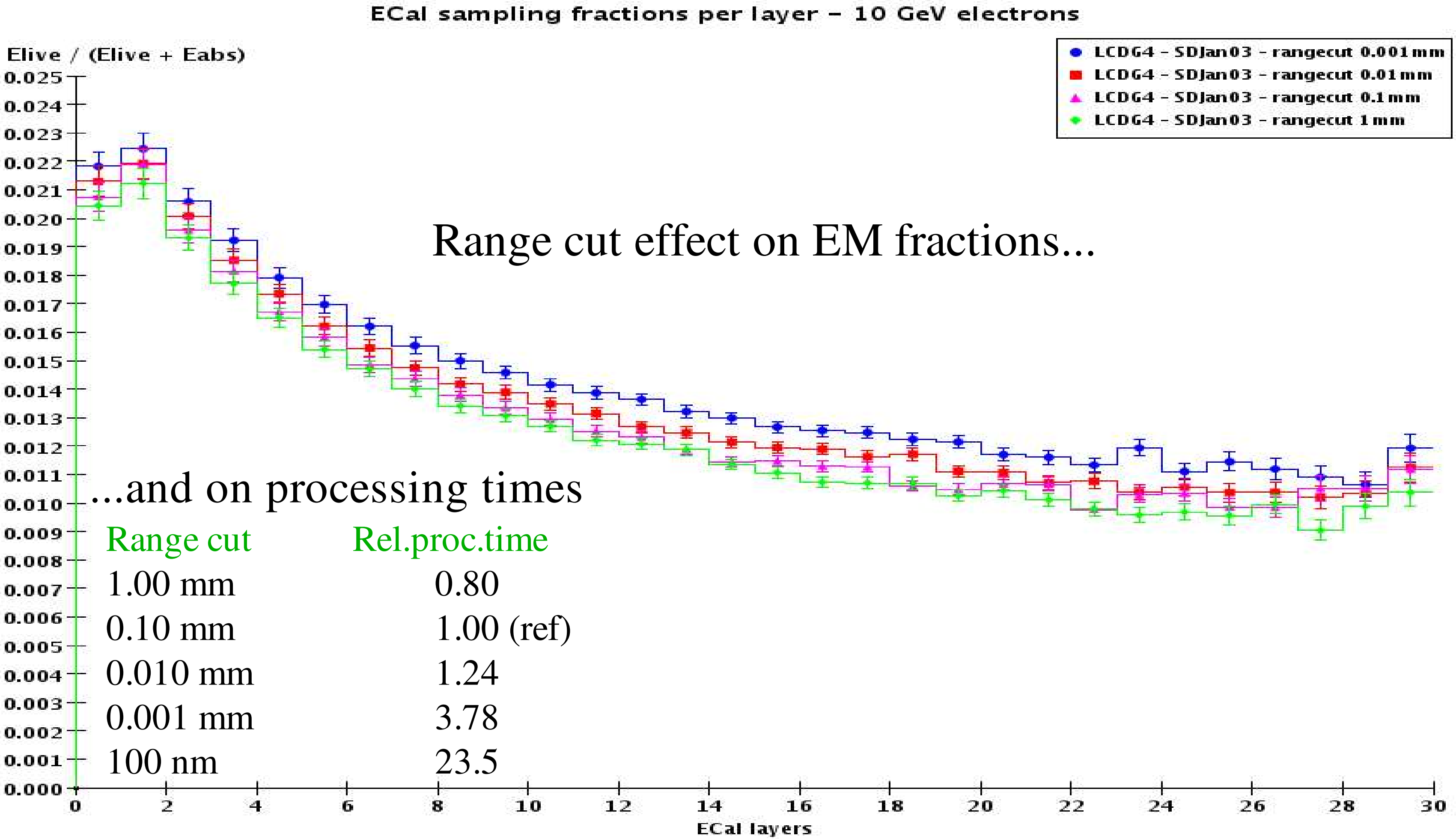
(3) Large number of HCal hits in the ECal/HCal interface

(4) Calorimeter entrance point coords in mm (not cm)
this has been fixed in v02-25

Issue 1: ECal E_{live} and E_{abs}



EM Sampling fraction per layer

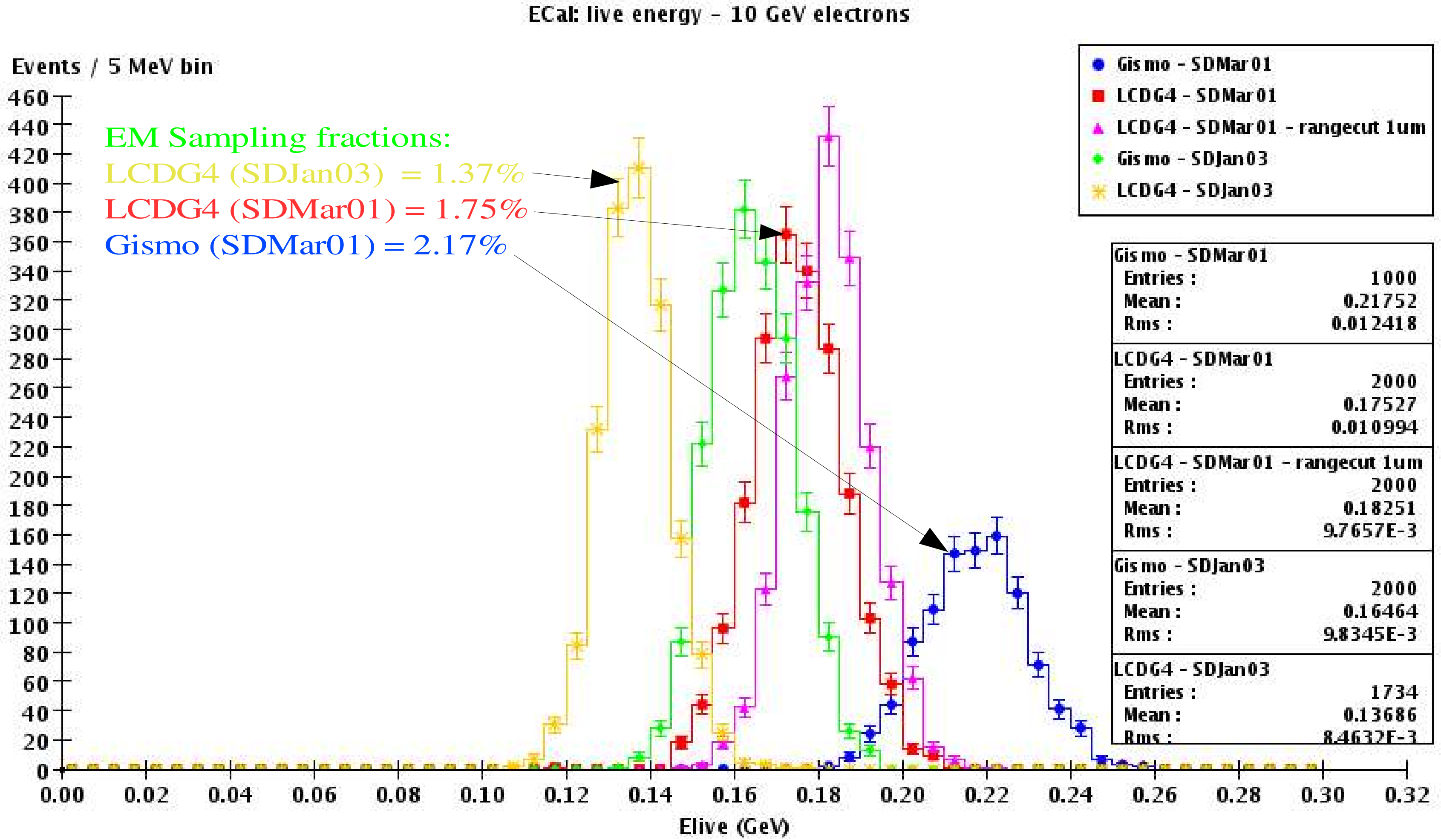


Range cut effect on EM fractions...

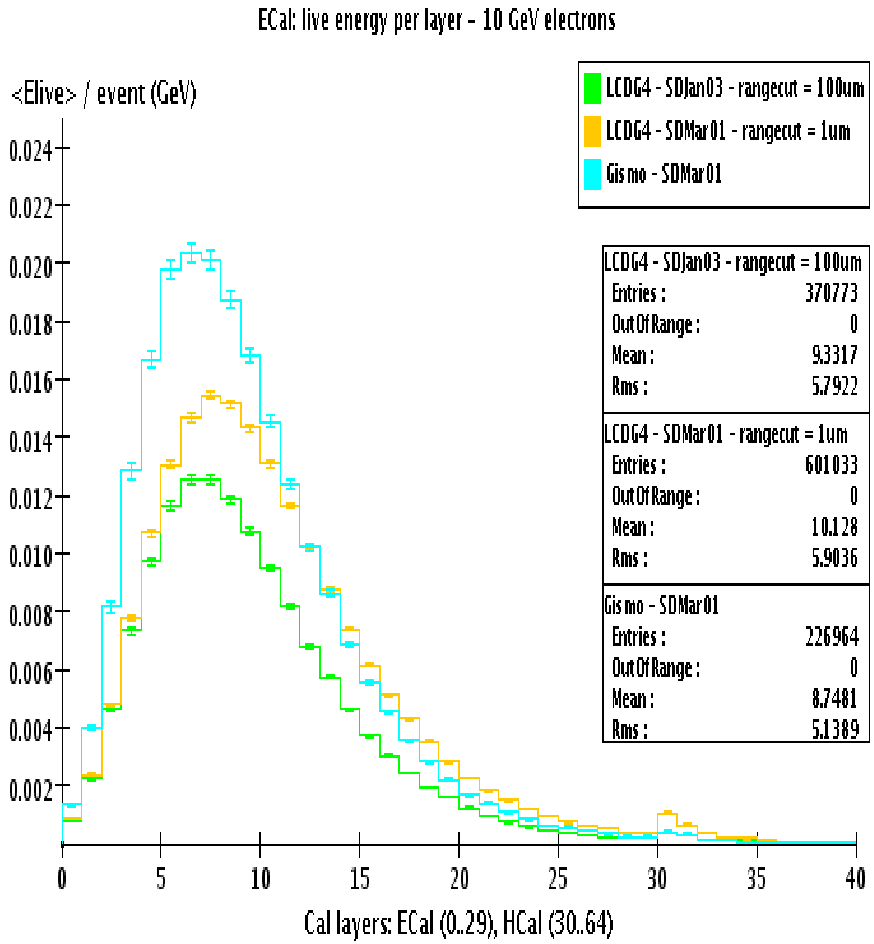
...and on processing times

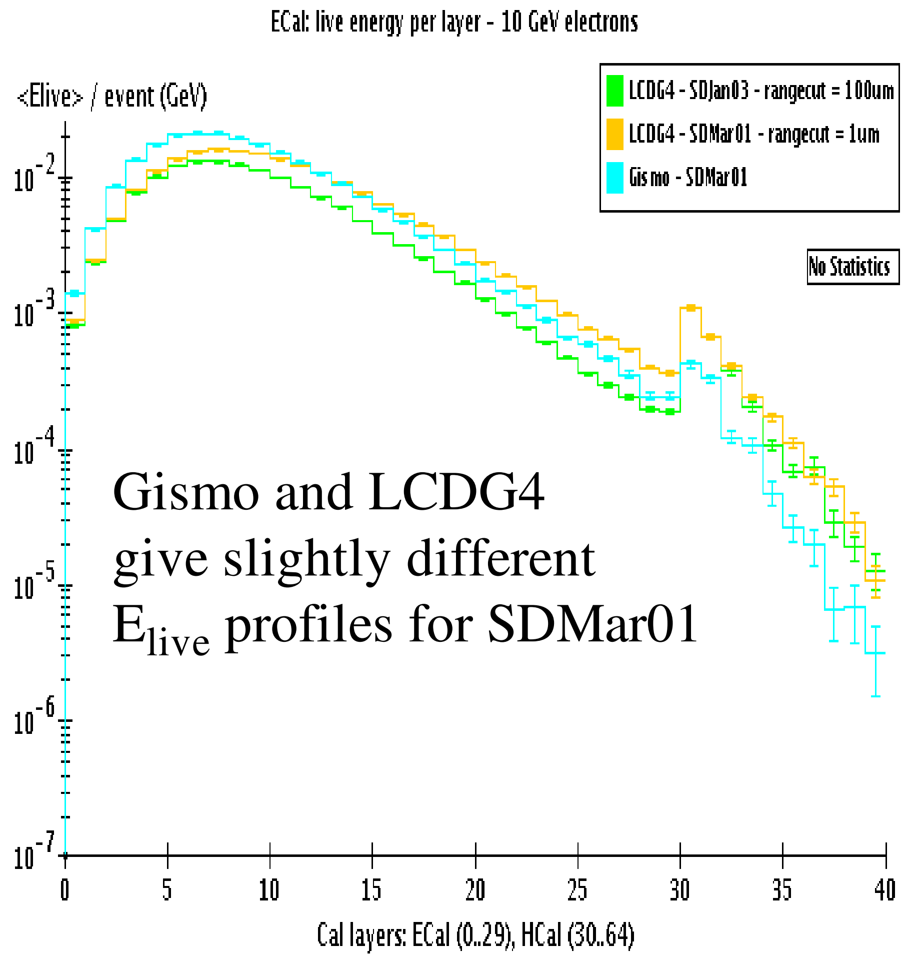
Range cut	Rel.proc.time
1.00 mm	0.80
0.10 mm	1.00 (ref)
0.010 mm	1.24
0.001 mm	3.78
100 nm	23.5

Issue 1: Ecal E_{live}



Issue 1: ECal E_{live} per layer





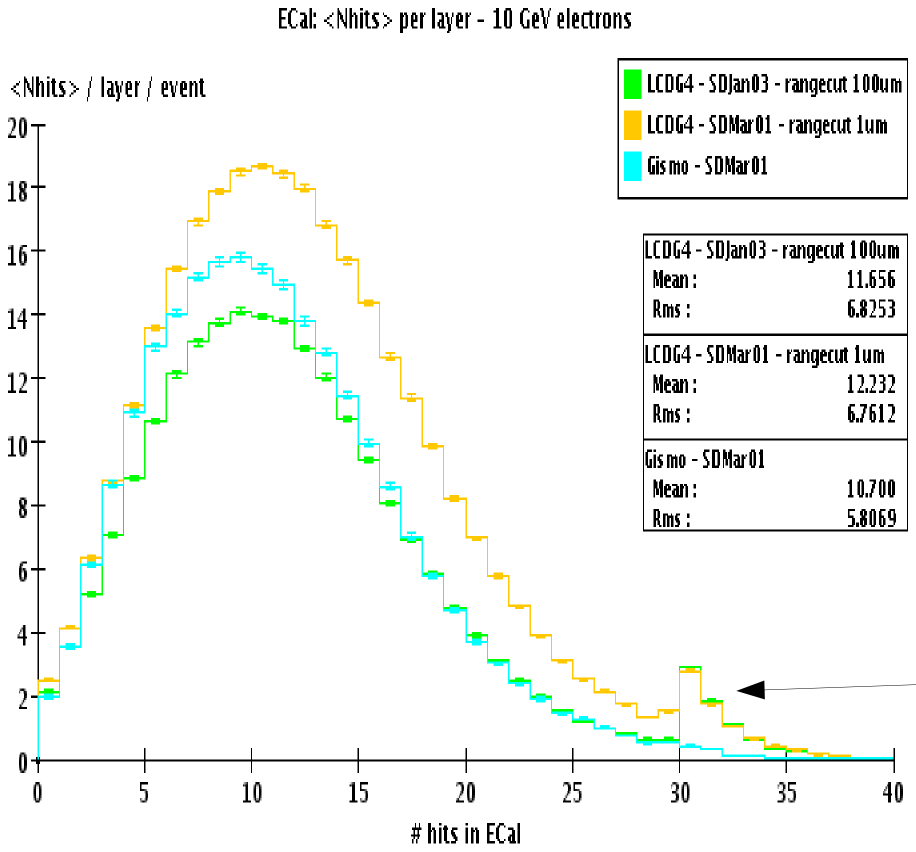
Gismo and LCDG4 give slightly different E_{live} profiles for SDMar01

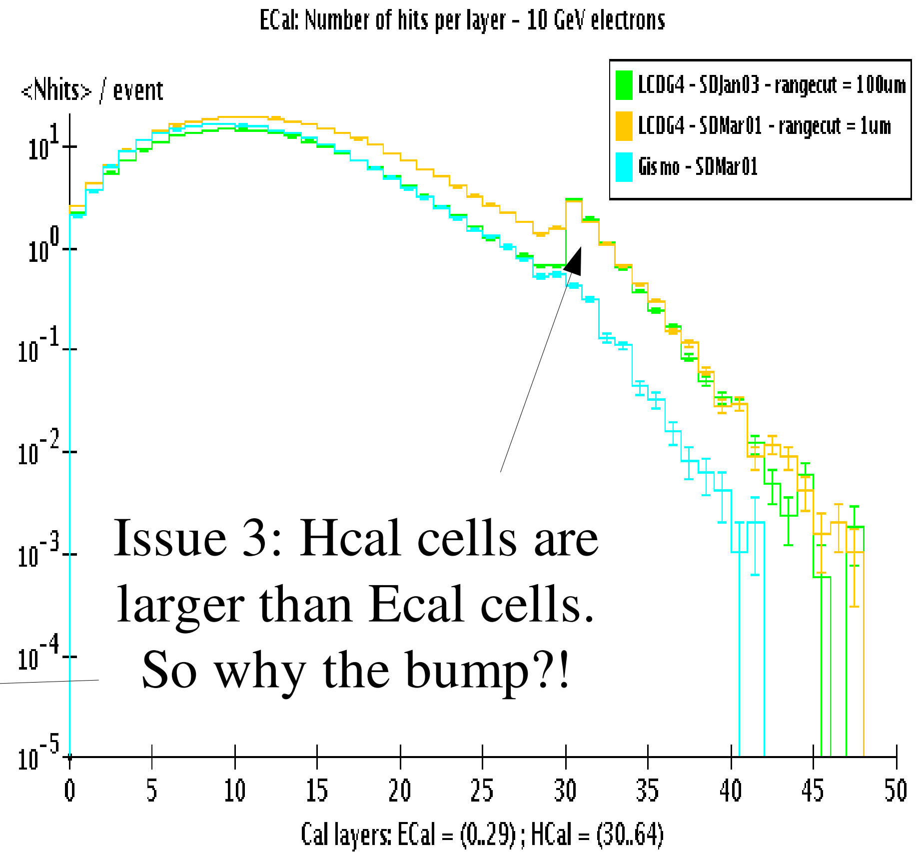
Issue 1: radiation lengths?

Material	Radiation lengths (cm)			
	PDG	Geant	G4-PDG	Gismo
W	0.35	0.350	0.00%	?
Si	9.36	9.094	2.84%	?
G10	19.40	16.716	13.84%	?
Cu	1.43	1.440	-0.70%	?
Air	30420	30551	-0.43%	?

There are some discrepancies with LCDG4 and PDG, but below the 3% level.
Maybe Gismo's radlens are different from PDG numbers by more than 10% level?

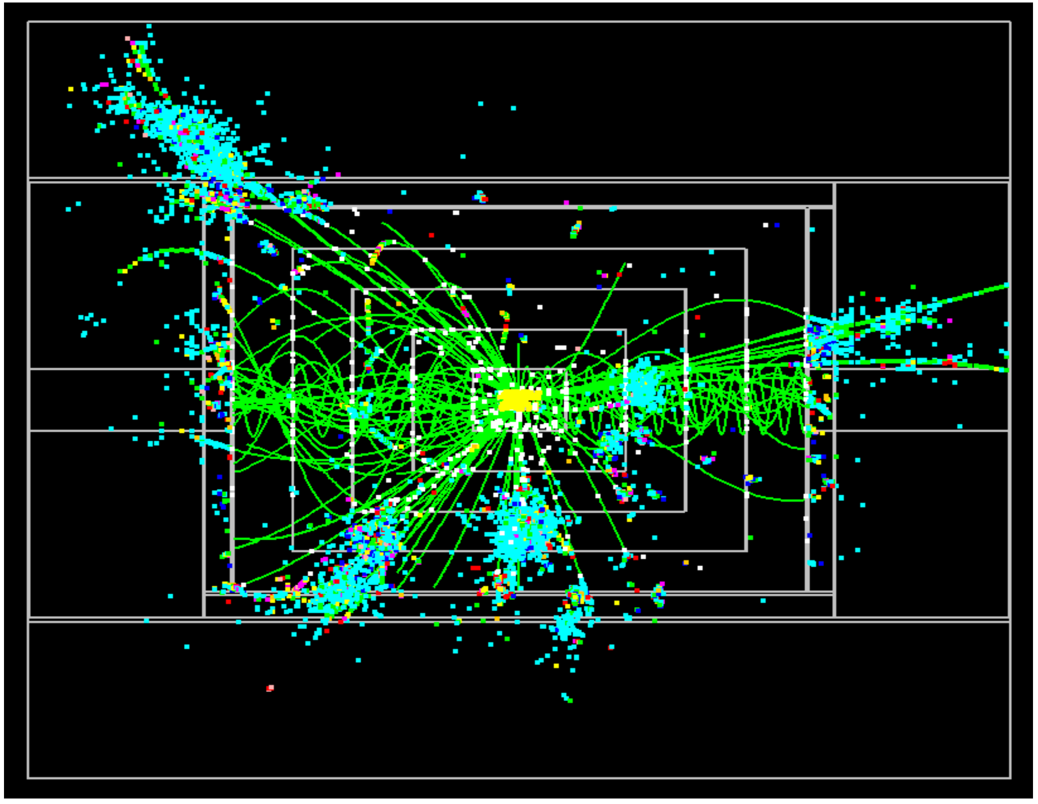
Issues 2 and 3: Nhits per layer

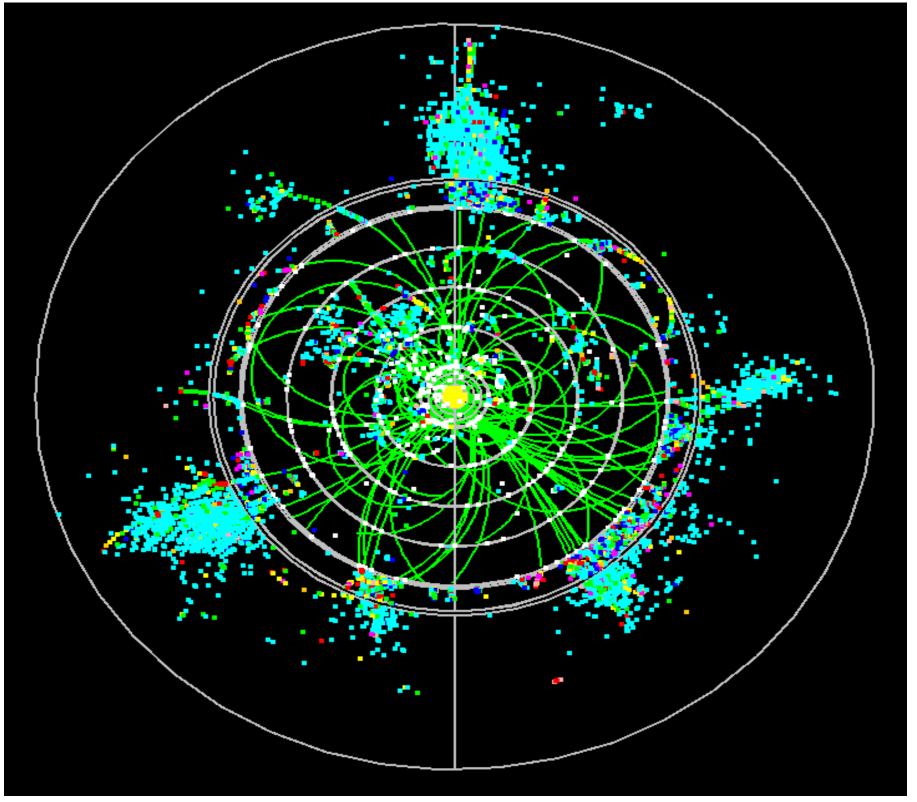




Issue 3: Hcal cells are larger than Ecal cells. So why the bump?!

e^+e^- into ttbar event (SDJan03)





MC Samples for general use

- Samples currently available at NIU through sftp:
  scpuser@k2.nicadd.niu.edu (lcd_2004): /pub/lima/lcdg4/v02-24
  - 2K each of $e^{\pm}$, $\mu^{\pm}$, $\pi^{\pm}$, γ, n at $\theta = 90^{o}$ and flat in φ
    energies = 2, 3, 5, 10, 15, 20, 30, 50 GeV
  - 5K 10 GeV K_s^0 into $\pi^0\pi^0$
  - 10K 10 GeV K_s^0 into $\pi^+\pi^-$
  - 5K 10 GeV Σ+ inclusive
  - 5K 1..10GeV lambdas inclusive
  - 10K Z into (hadrons) at 91 GeV
  - 5K ttbar inclusive at 350 GeV
  - 5K WW into (hadrons)(any) at 500 GeV
  - 2K ZH into (any)(bbbar) at 500 GeV and M_H=120 GeV
  - 2K ZH into (any)(bbbar) at 500 GeV and M_H=160 GeV

- Other samples can be requested to lima at fnal.gov. Please read http://nicadd.niu.edu/~jeremy/lcd/simreq/ for guidelines.

How to access the MC samples

Several single-particle and physics data samples available from NIU data server using secure ftp:

```
 % sftp scpuser@k2.nicadd.niu.edu
password: lcd_2004
sftp> cd pub/lima/lcdg4/v02-23
sftp> ls   (to see a list of .sio files available)
sftp> mget muons-10gev*.sio    (for example)
sftp> quit
%
```

See http://nicadd.niu.edu/~jeremy/admin/scp/index.html for more detailed access instructions, including instructions for windows winscp utility.

LCDG4 status summary

- Detailed comparisons between LCDG4 v02-11 and LCDMokka 01-05 are in good agreement (discrepancies of ~20% to Gismo have been observed)
- LCDG4 faster than Mokka, but it cannot be used for Tesla geometry
- Only cylinders, disks and cones supported by current LCDG4 version (like Gismo). More realistic geometries to be implemented in the medium term
- Several MC physics samples have been generated for algorithm development and studies (SDJan03, SIO format)
- Source code available from SLAC or NIU CVS repositories
- For more information please check the LCDG4 documentation web page: http://nicadd.niu.edu/~lima/lcdg4/, or under subdirectory doc of CVS module

Next steps suggested

- Brief investigation on the ECal/HCal interface
- Improve documentation
- LCIO output
- Merge projective and non-projective versions
- Visualization improvement (virtual cells)
- Different range cuts per subcomponent (ECal)
- More realistic geometries (medium term)