

LCDG4 and DigiSim

Simulation activities at NIU

Guilherme Lima

for the ILC software group at NICADD/NIU



NORTHERN ILLINOIS
UNIVERSITY

ILC Simulations Workshop
SLAC, March 16, 2005

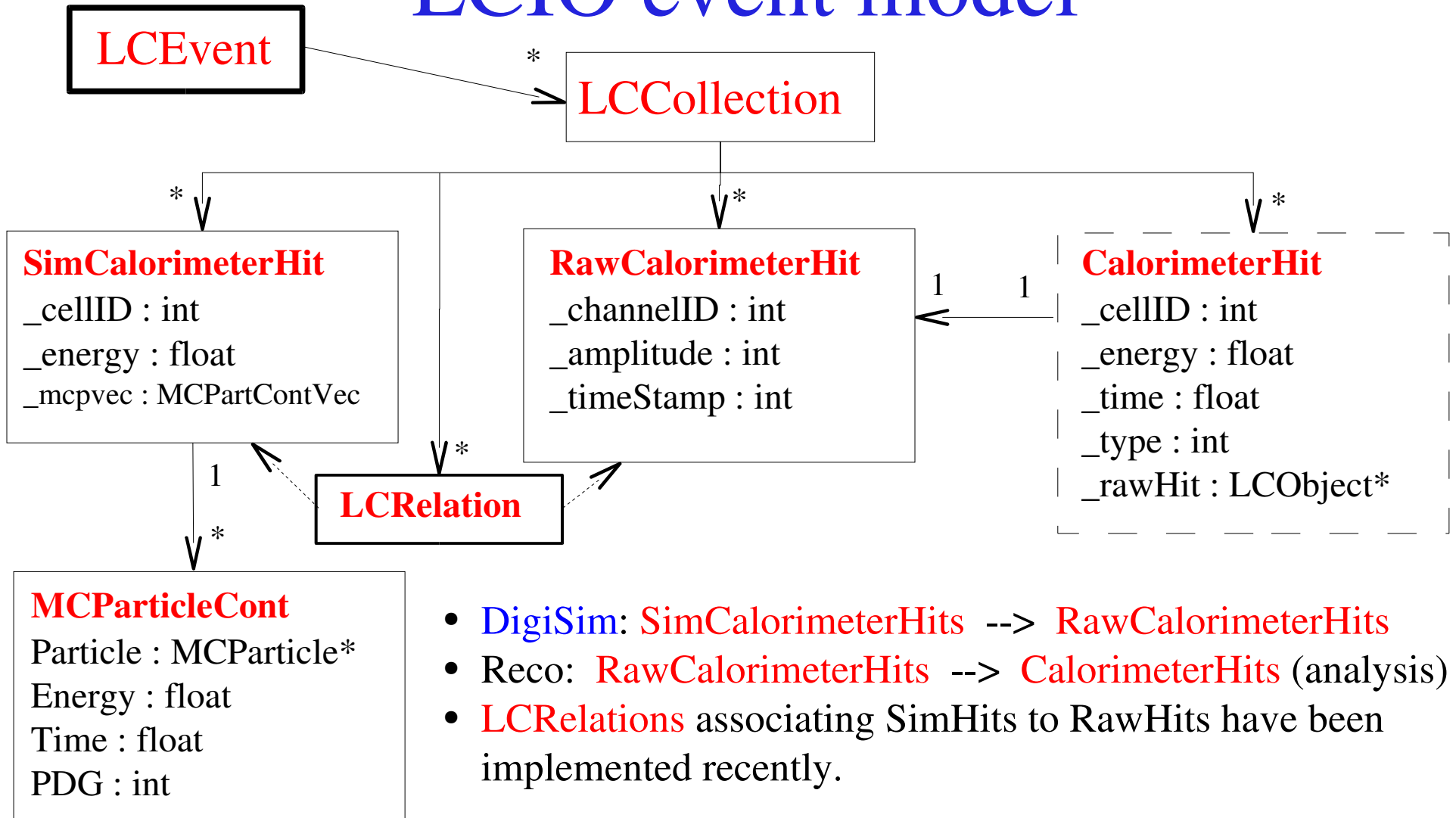
Talk Outline

- DigiSim
 - Very brief: purpose and status
- LCDG4
 - Introduction
 - Simple analyses
 - recent developments
 - Status and plans

DigiSim

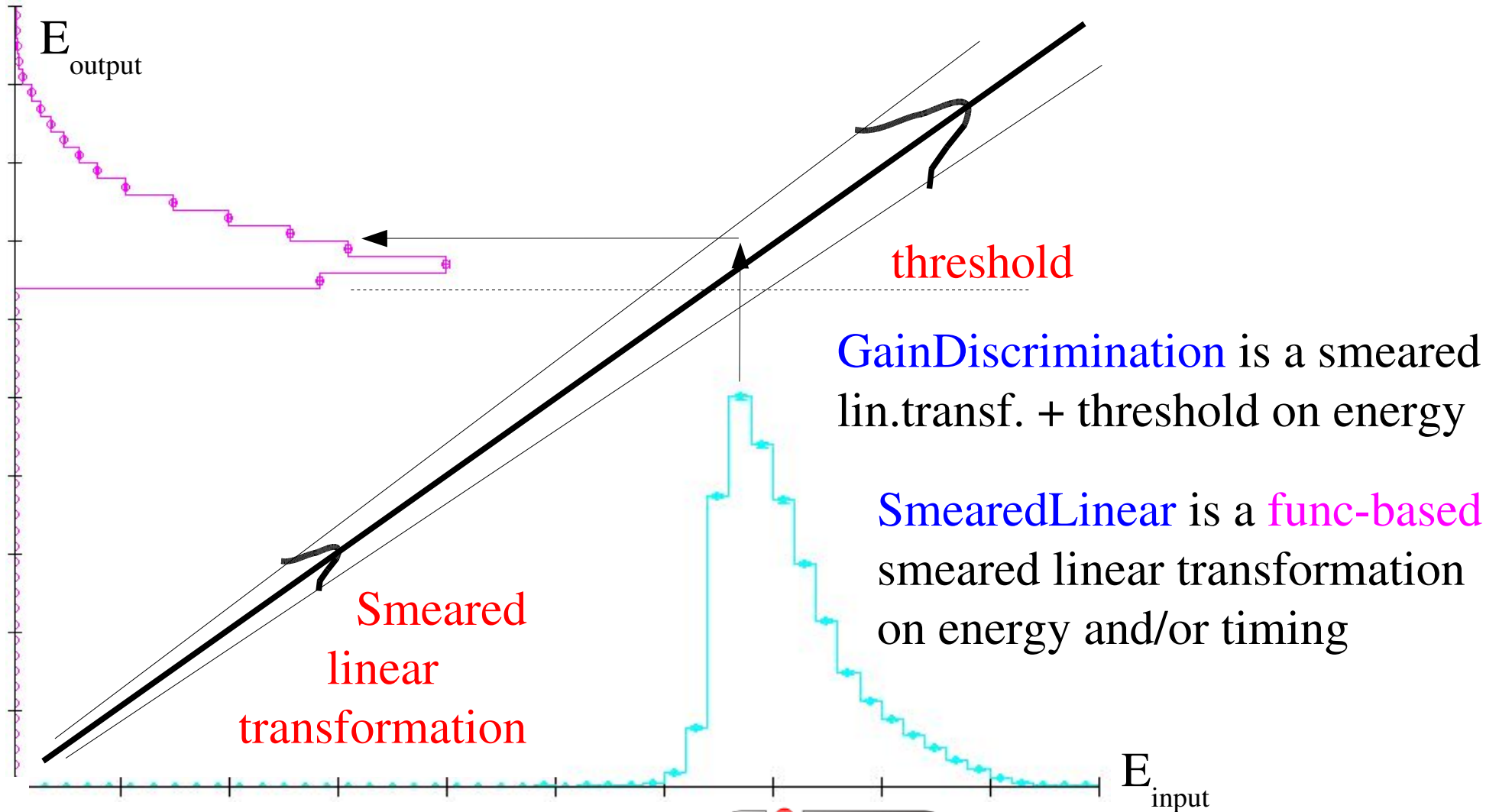
- Goal: a program to simulate the digitization process for the ILC detector simulation
- Basic requirements:
 - Object oriented design to **simplify maintenance and implementation of new functionality**
 - Should be used within the CALICE test beam project, as a testbed for the reconstruction framework
 - All test beam code based on C++ and LCIO
- Gaede's Marlin (v0.6) chosen as the C++ framework

LCIO event model



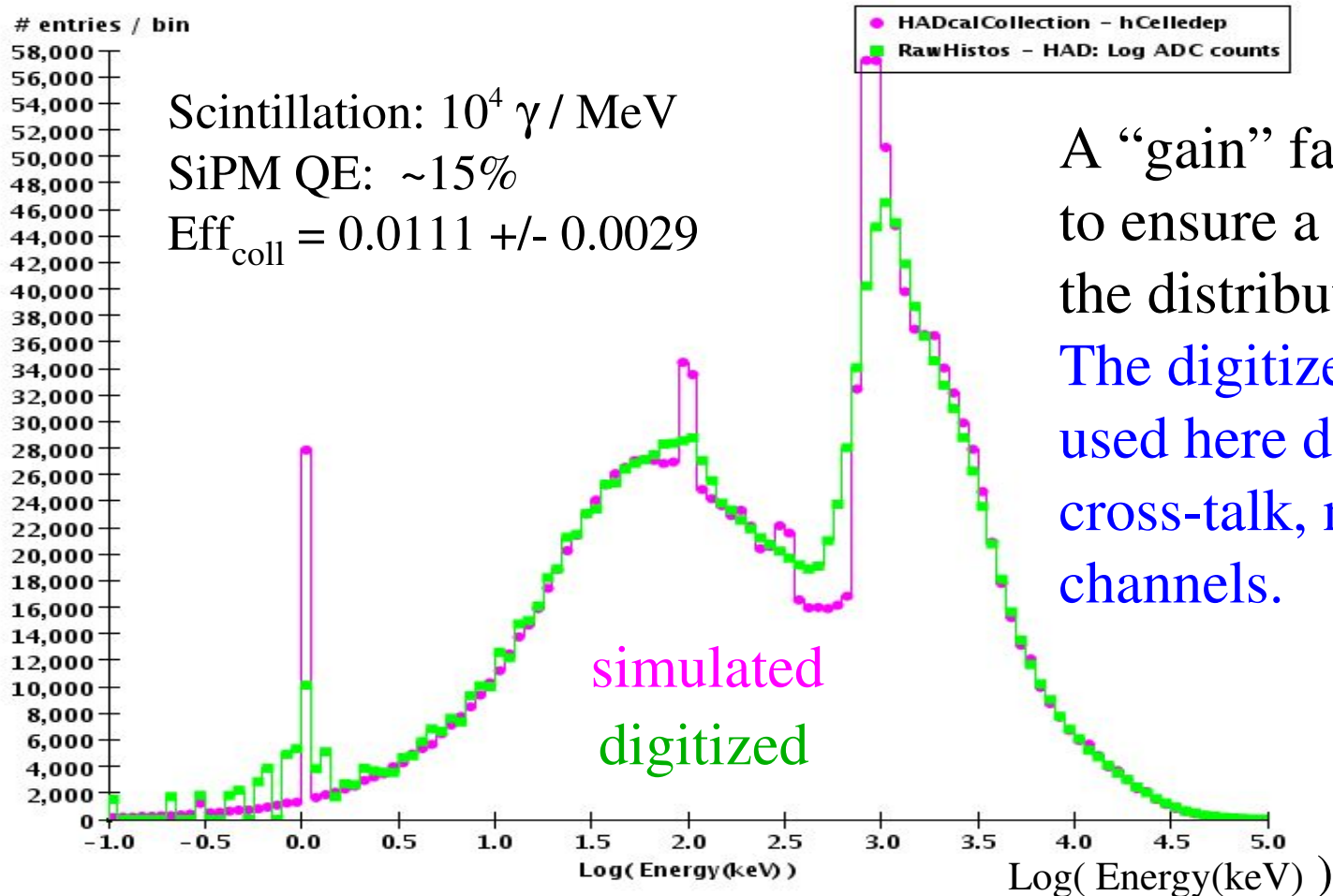
- **DigiSim**: **SimCalorimeterHits** --> **RawCalorimeterHits**
- **Reco**: **RawCalorimeterHits** --> **CalorimeterHits** (analysis)
- **LCRelations** associating SimHits to RawHits have been implemented recently.

Existing modifiers



HCal digitization (1st attempt)

50GeV pions - DigiSim for HADCal



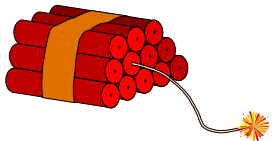
A “gain” factor was used to ensure a good overlap of the distributions shown. The digitizer configuration used here does not include cross-talk, noise or hot/dead channels.

DigiSim Status

- A digitization simulation package, DigiSim, has been developed at NICADD/NIU
 - Generic modifiers available: GainDiscrimination and function-based SmearedLinear
 - LCRelations implemented to associate raw hits to corresponding simulated hits
- DigiSim can be run either standalone (output file) or as an on-the-fly preprocessor
- EMraw, HADraw (raw hits) and LCRelations collections are appended to LCEvent, in addition to the (untouched) simulated calorimeter hits collections.
- A first version of a real (though incomplete) tile HCal digitizer exists. It can be used as an example to implement ECal or TCMT.
- When reuse of existing generic modifiers is not appropriate, the creation of new modifiers is quite easy, by just copying one of the existing modifier classes and implementing the desired transformation.
- Crosstalk and cell ganging, require neighborhood information from geometry-aware classes, which are currently under development.

LCDG4: Full detector simulation

- A Geant4-based detector simulator to support detector R&D for the Linear Collider (up-to-date with latest version, Geant4.7.0.p01)
- Alternatives: Mokka (Europe) or SLIC (new, under development)
- LCDG4 features
 - Input format: binary STDHEP
 - Output format: *.lcio* (standard) or *.sio* (Gismo compatible)
 - Some detector geometries are implemented via XML geometry files (e.g. SiD, LD and variants)



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

Some LCDG4-specific features

- Development bias towards calorimeters
other subdetectors probably work fine, but no active studies have been done by the developers. If you find problems, please let us know.
- Correct MC particle hierarchy, even when V^0 s and hyperon decays are forced at the event generation stage
- Xml-based geometry is very easy to make interesting changes
- Non-projective calorimeter geometries also available via XML (e.g. see `HAD_Barrel` in `ParFiles/SDNPHOct04.xml` for an example)
- Simple analysis code (java) and documentation available from CVS and <http://nicadd.niu.edu/~lima/lcdg4/>.

More LCDG4 features

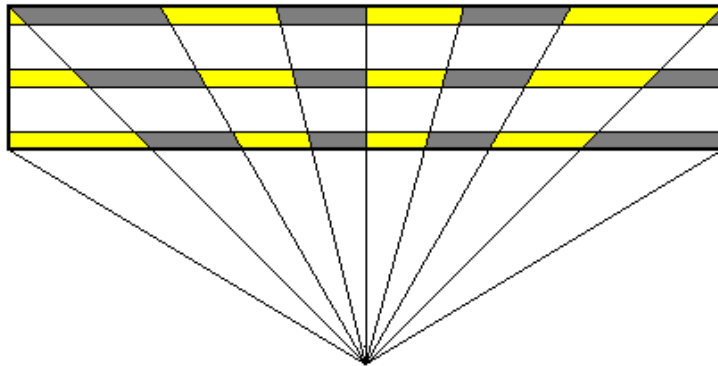
- Many hadronic physics lists available, including ILC-oriented [LCPhys](#)
- Energies deposited in absorbers are easily available for cross-check analysis (ASCII format only)
- Useful debugging tools are available, like:
 - Run-time control (RTC) of debugging printout
 - RTC of random seed initialization, either one-time or same seed for all events
 - RTC of skipping events from input file

Projective or non-projective

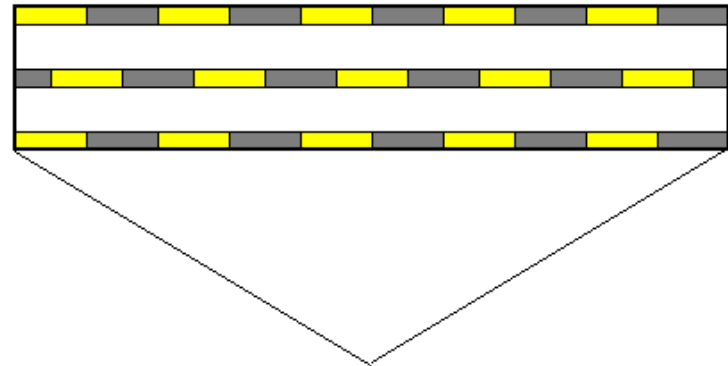
Projective or non-projective virtual calorimeter cells available

- barrel only for now (see e.g. SDNPHOct04.xml for a version with Non-Projective Hcal), but nonproj endcaps may be available later
- Rectangular virtual cells with fixed size, controlled at run time (XML)
- ECal or HCal barrels can be made projective or NP independently

projective



non-projective



Geometry description in XML

An example: Hadronic calorimeter barrel.

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" /-->
    <cell_info length="1.0" height="1.0" offset="0." outer_r="246.5"/>
  </tube>
  <calorimeter type="had" />
</volume>
...
```

Proj or NP selection:
either one, not both.
To do: better tag names

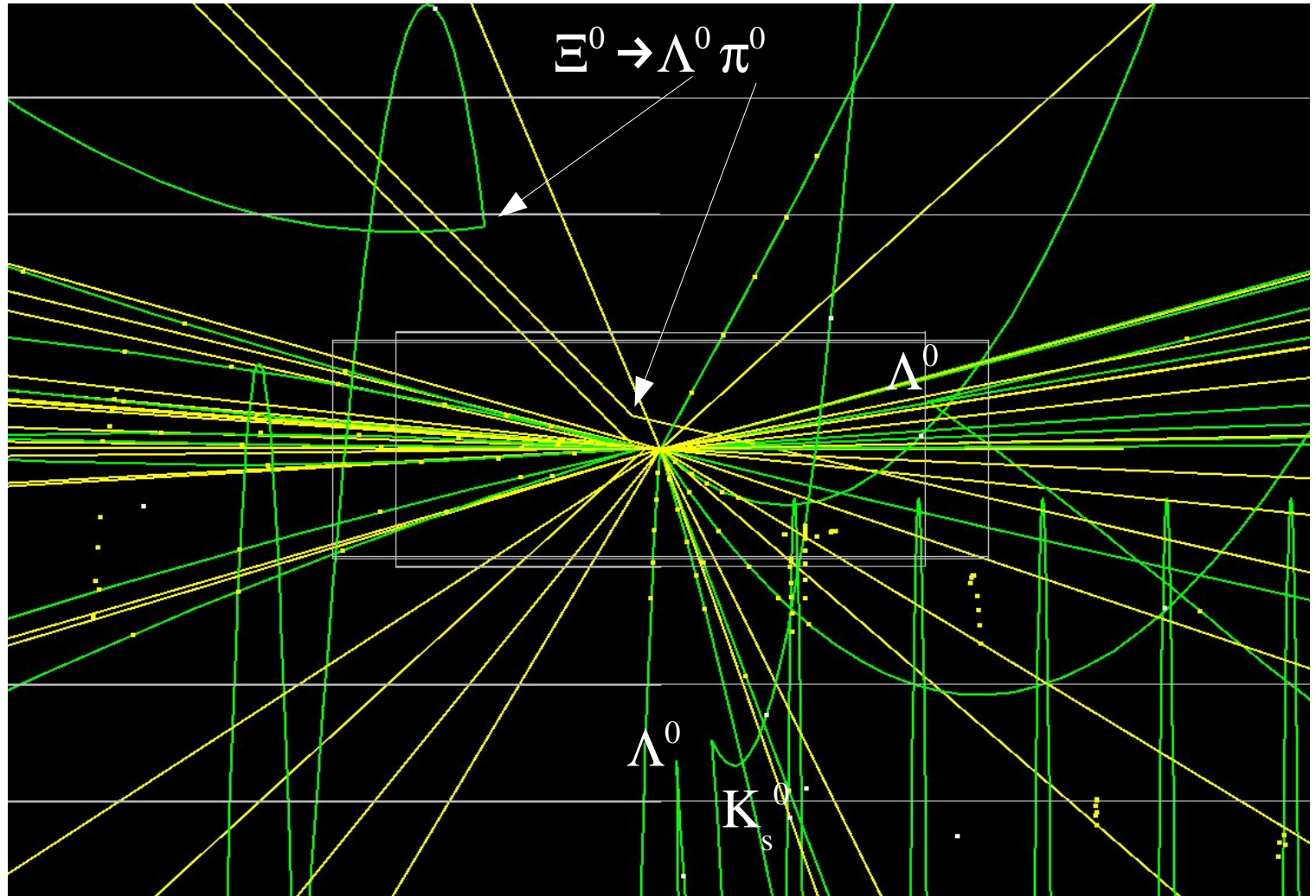
Most of detector changes are very easy to make!

SIO / LCIO output: general features

- One particle collection and several hit collections (one hit collection per subdetector)
- Each hit points to the contributing particles (except tracker hits from calorimeter back-scatterings, as in Gismo)
 - LCIO only: (x,y,z) coordinates stored for every hit
- All secondaries above an energy threshold (now set at 1 MeV), except for shower secondaries, are saved in output with:
 - Particle id and generation/simulation status codes
 - Production momentum, production* and ending position (*LCIO only)
 - Calorimeter entrance point: position and momentum (SIO only)
 - Pointers to parent particles

Zoom on the primary interaction

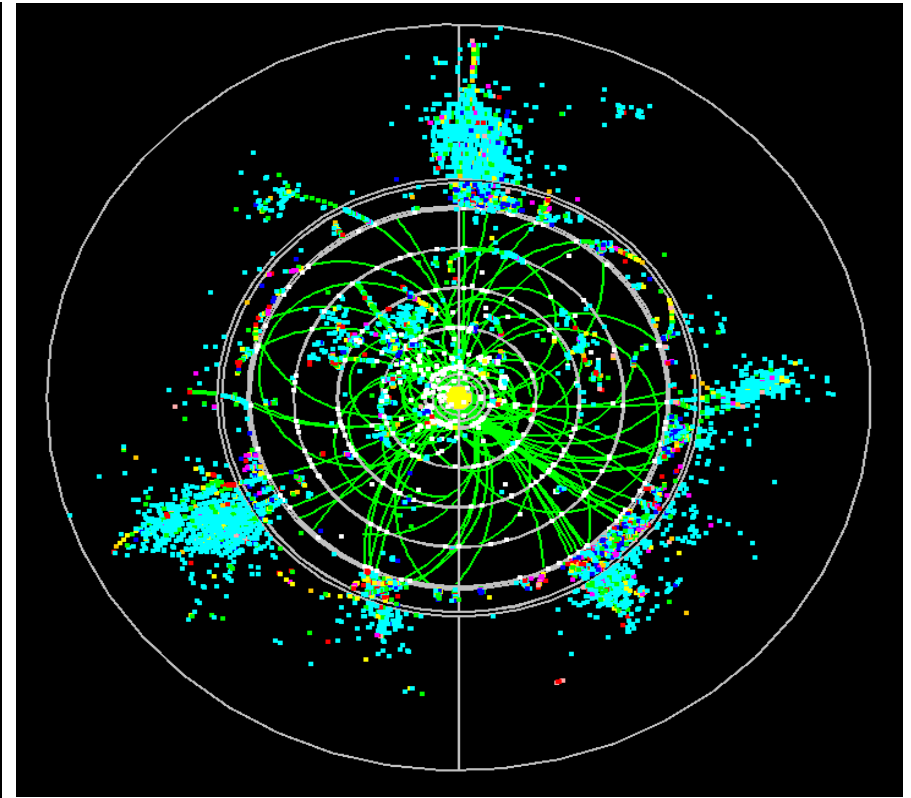
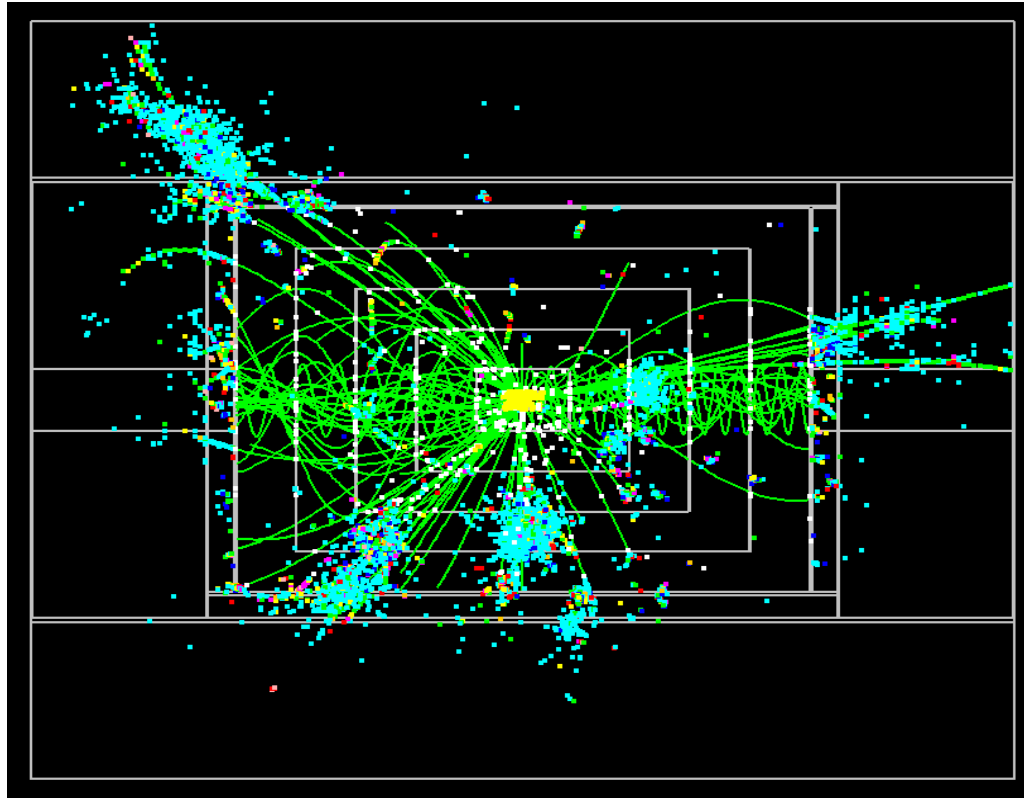
All decays forced in the generator are correctly processed by the LCDG4 simulation



Green: charged
Yellow: neutrals

Event displays (JAS3)

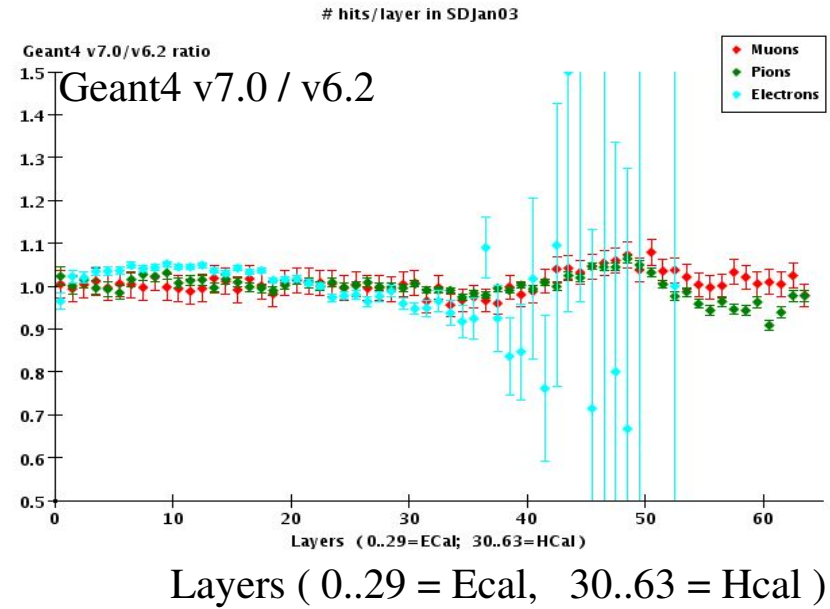
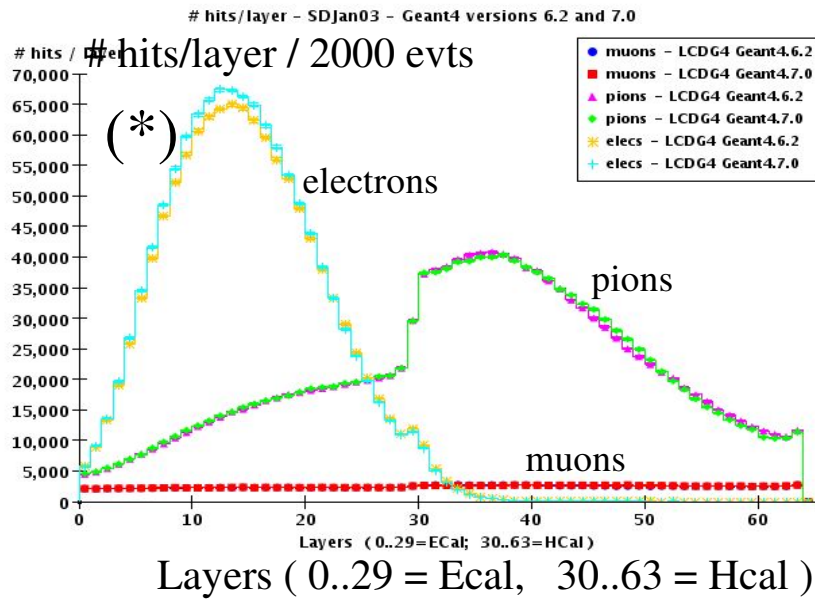
e^+e^- into $t\bar{t}$ event (SDJan03)



Data after Geant4 upgrade

- A comparison between Geant4.6.2 (v03-18) and Geant4.7.0 (v03-19) based on normally incident, 50GeV muons, electrons and pions

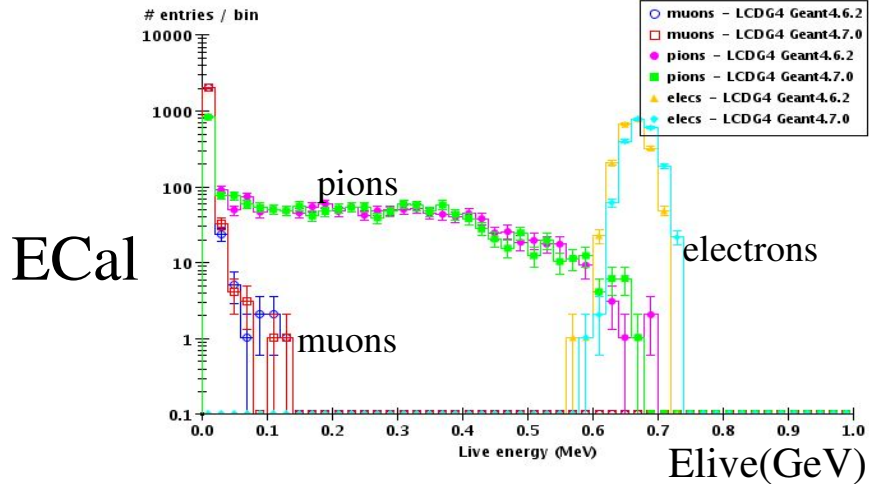
Comparing Geant4.7.0 to previous version



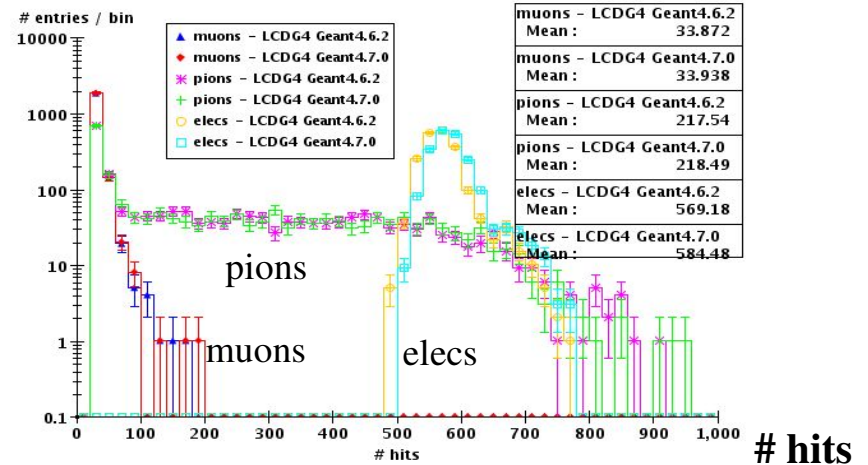
(*) Main difference w.r.t. Geant4.6.2 is in the EM distributions. There were some changes to EM processes for Geant4.7.0.

Comparing Geant4.7.0 to previous version

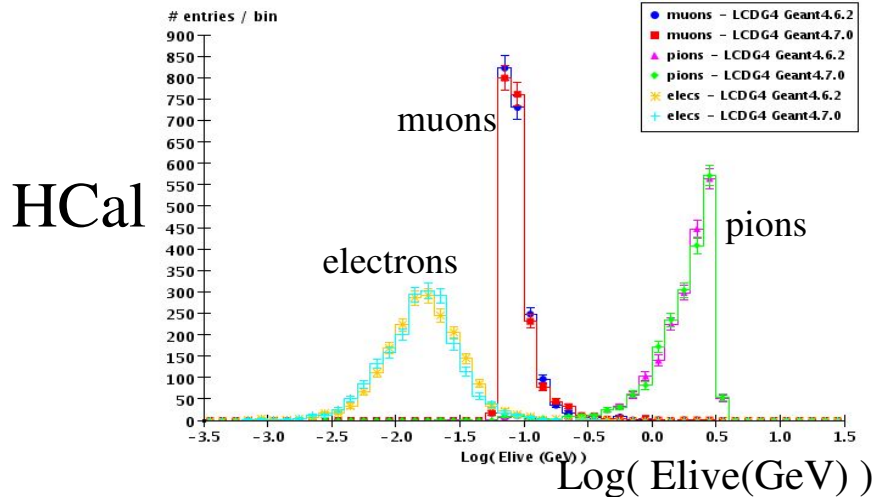
Total ECal Elive - SDJan03 - Geant4 versions 6.2 and 7.0



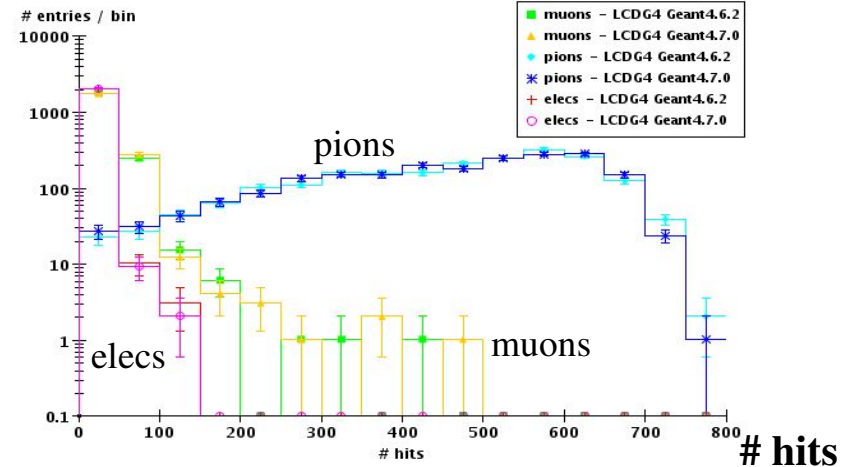
ECal hits - SDJan03 - Geant4 versions 6.2 and 7.0



Total HCal Elive - SDJan03 - Geant4 versions 6.2 and 7.0



HCal hits - SDJan03 - Geant4 versions 6.2 and 7.H

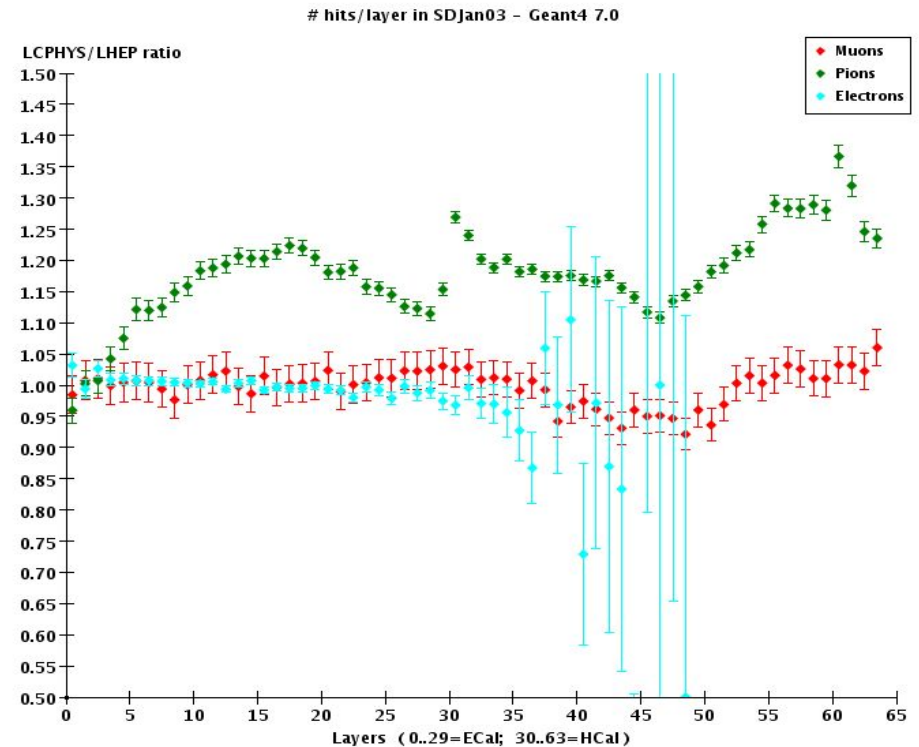
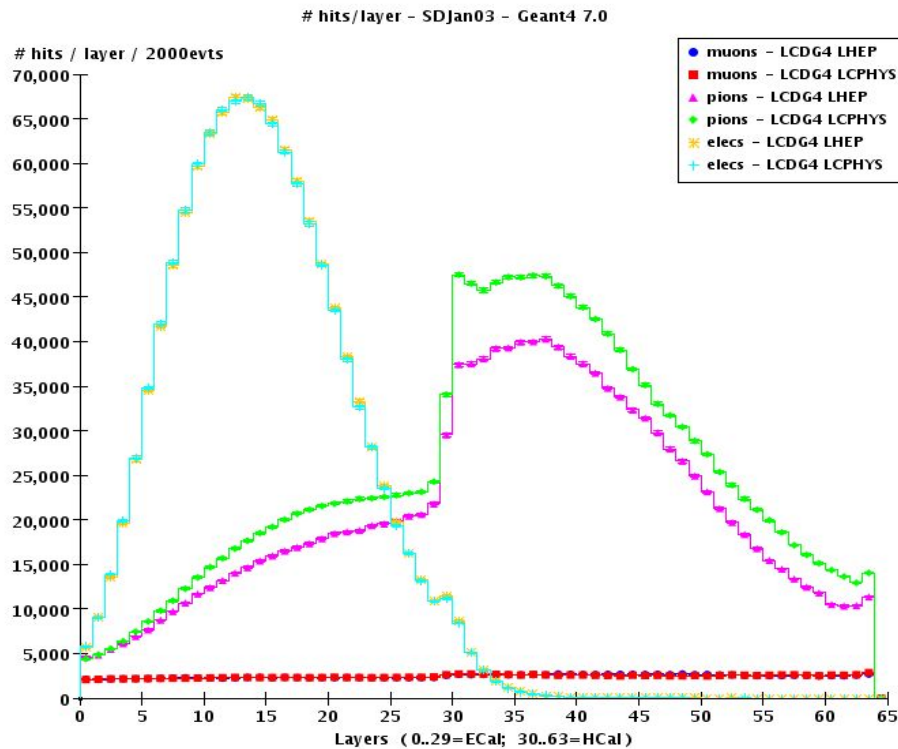


Comparing LCPHYS to LHEP

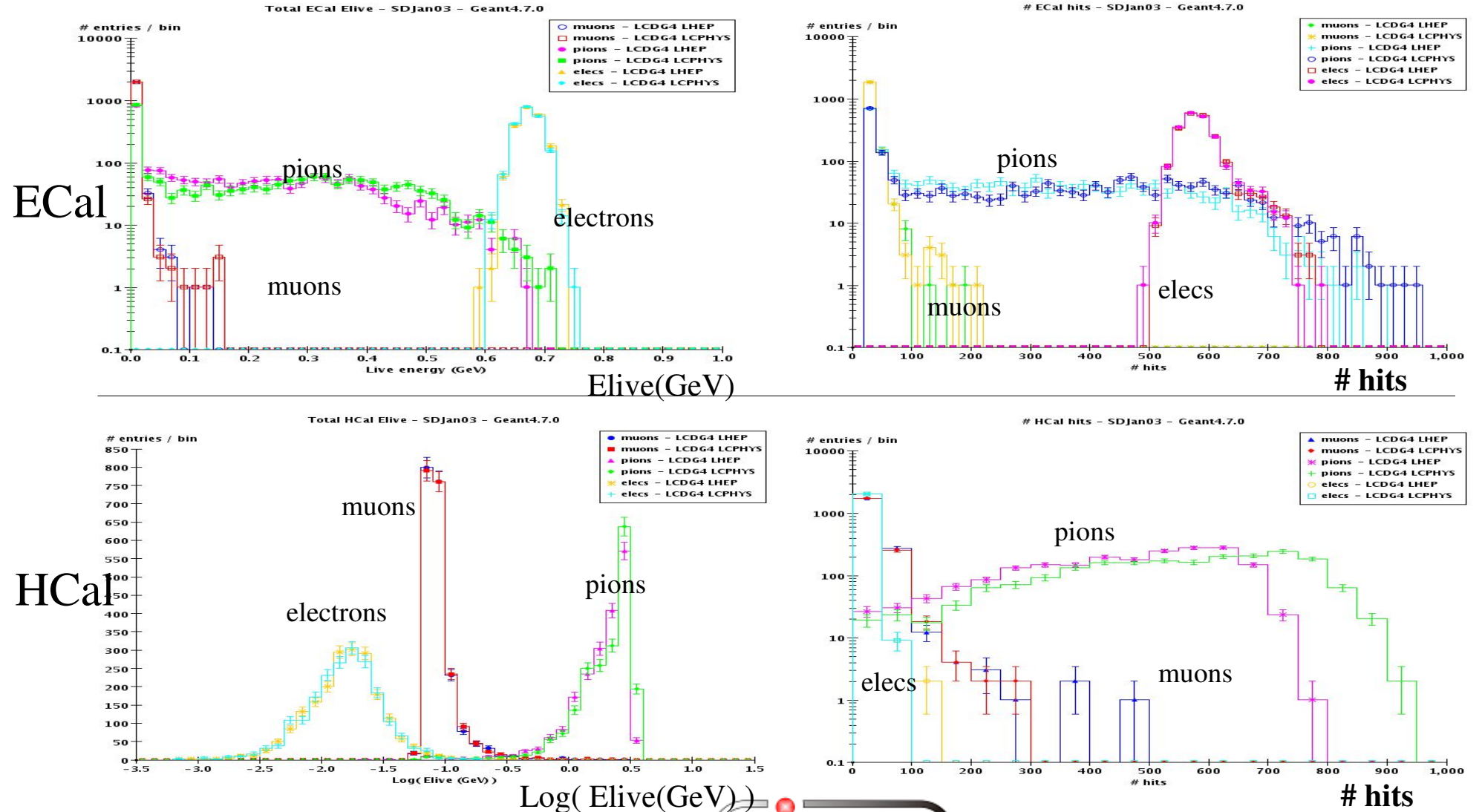
- A *comparison* of new hadronic physics list LCPHYS and the default LHEP, using Geant4.7.0 and normally incident, 50 GeV pions, electrons and muons.

Shower profiles

Now comparing physics lists LCPHYS vs. LHEP

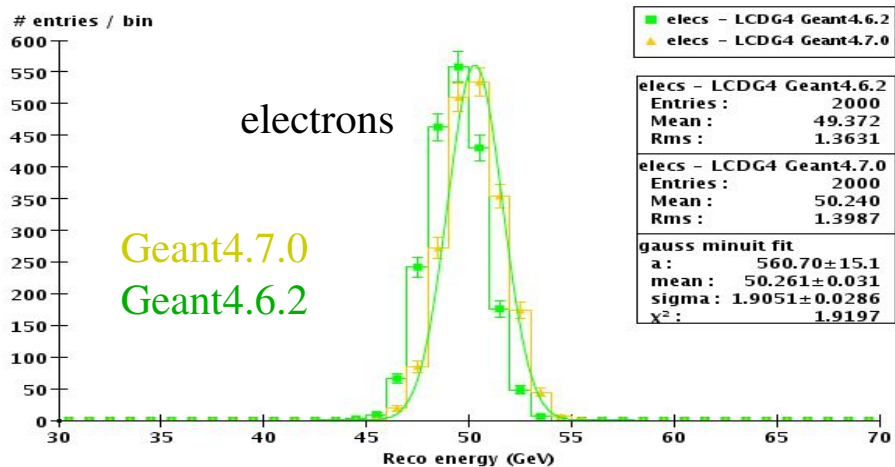


Ecal Nhits

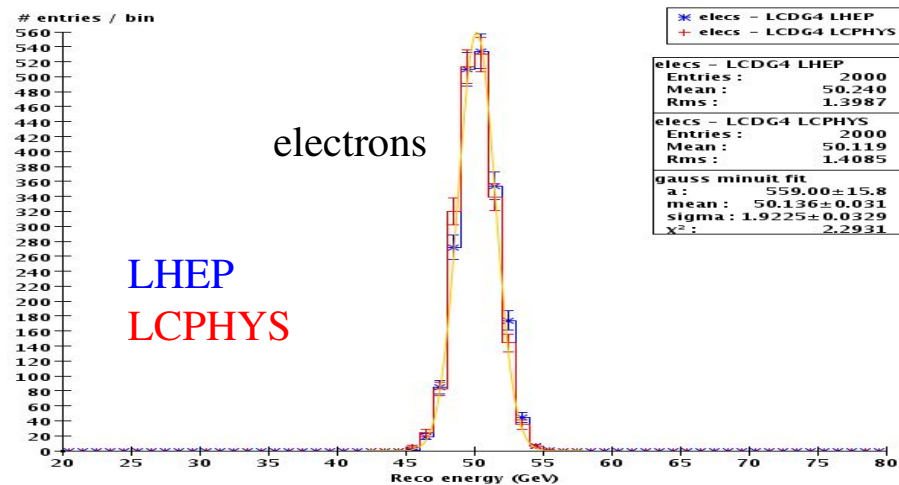


Effects on reconstructed energy

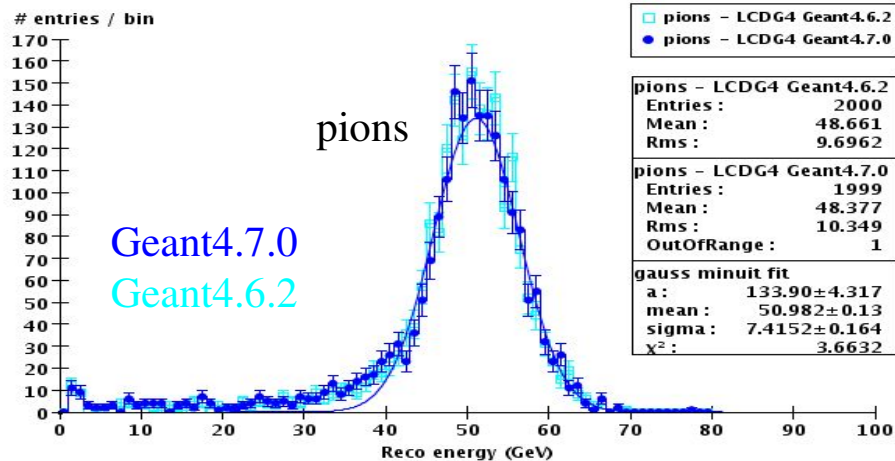
50GeV electrons - SDJan03 - Geant4 versions 6.2 and 7.0



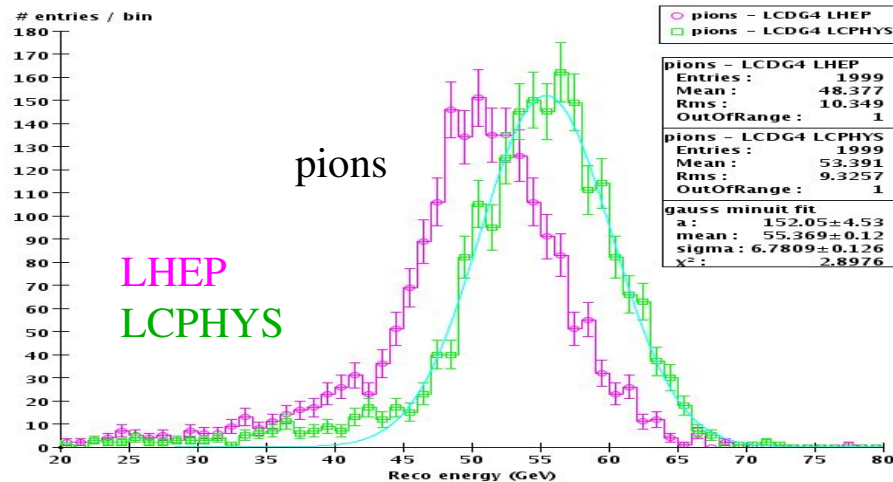
50GeV electrons - SDJan03 - LCPHYS vs. LHEP



50GeV pions - SDJan03 - Geant4 versions 6.2 and 7.0



50GeV pions - SDJan03 - LCPHYS vs. LHEP



MC Samples for general use

- Samples currently available at NIU through sftp:
scpuser@k2.nicadd.niu.edu (lcd_2004): </pub/lima/lcdg4/v03-19>
 - 2K each of $e^\pm, \mu^\pm, \pi^\pm, \gamma, n$ at $\theta = 90^\circ$ 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 decays
 - 5K 1..10GeV Λ^0 , inclusive decays
 - 10K Z into (hadrons) at 91 GeV
 - 5K $t\bar{t}$ inclusive at 350 GeV
 - 10K WW into (hadrons)(any) at 500 GeV
 - 10K $\tau^+\tau^-$ at 500 GeV
 - 4K ZH into (any)($b\bar{b}$) at 500 GeV and $M_H=120, 160$ GeV (2K events each)
- Other samples can be requested to lima at fnal.gov. Please read <http://nicadd.niu.edu/~jeremy/lcd/simreq/> for guidelines.

Both SDJan03 and
SDNPHOct04 geometries,
plus some variants

Other interesting samples available

- Some samples have been simulated with alternative geometries:
 - [SDJan03-ANL](#) in v03-06, [SDNPHOct04-85W](#) in v03-19:
[ANL](#): 60 proj HCal layers, each with 0.7cm tungsten + 1cm scintillator
[85W](#): 85 NP HCal layers, each with 0.7cm tungsten + 0.5cm scintillator
 - 2K each of μ^\pm, π^\pm at $\theta = 90^\circ$ and flat in φ
energies = 2, 3, 5, 10, 15, 20, 30, 50 GeV
 - 10K Z into (hadrons) at 91 GeV
 - [SDNPHOct04-BXX-EcalRminYYY](#) in v03-16:
Non-projective HCal barrels with Ecal inner radius varying from 127 to 164cm, and magnetic fields (B) ranging from 3.0 to 5.0 GeV, while keeping $BR^2 = \text{constant}$
 - 2K μ^\pm at $\theta = 90^\circ$ and flat in φ , energies = 2, 3, 5, 10, 15, 20, 30, 50 GeV
 - 2K e+e- into ZH into (qqbar)(bbbar), at 500 or 1000 GeV

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/
Sftp> ls      (to see a list of versions available)
Sftp> cd <some-version>
sftp> ls      (to see a list of files available)
sftp> mget muons-10gev*.slcio      (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.

Preassigned decays from generator

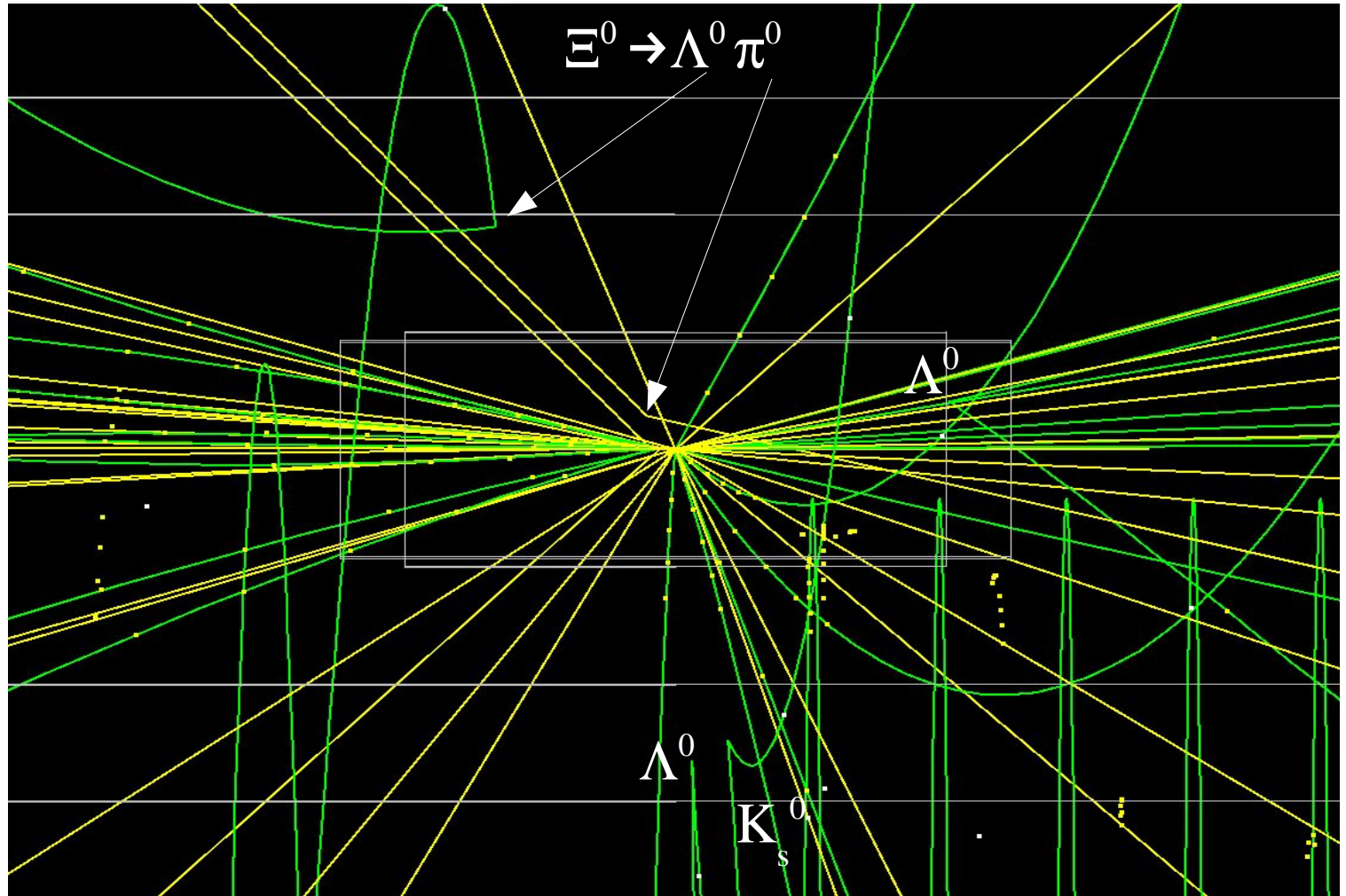
Usually: \pm , V^0 s,
charged hyperons
and short-lived
b,c-hadrons

Examples:

$$B^0 \rightarrow J/\psi K_s^0,$$

$$K_s^0 \rightarrow \pi^+ \pi^-$$

Not only FINAL
STATES, but also
INTERMEDIATES
need to be given for
simulation.



Recurrent problem

Parent-daughter inconsistencies present in input binary stdhep files

- Ex1: panpytautau
(Ron / Norman)

- DOCUMENTATION particles (istat=3) should not be given to simulation. But taus should be propagated

- Only reasonable solution seems to be change istat=3 to =2 (INTERMEDIATE).

- Other files with inconsistencies led me to some set of “fixing stages”.

tag	id	istat	moms	daus
1	11	3	0- 0	0- 0
2	-11	3	0- 0	0- 0
3	11	3	1- 1	0- 0
4	-11	3	2- 2	0- 0
5	15	3	3- 4	7- 8
6	-15	3	3- 4	12- 14
7	16	1	5- 5	0- 0
8	-20213	2	5- 5	9- 11
9	111	2	8- 8	15- 16
10	111	2	8- 8	17- 18
11	-211	1	8- 8	0- 0
12	-16	1	6- 6	0- 0
13	-11	1	6- 6	0- 0
14	12	1	6- 6	0- 0
(...)				

Recurrent problem

Another example: M.Ronan's data file (ronan.stdhep) shows “decays” with identical parents and daughters, but distinct istats. Setting taus istat from 3 to 2 can be confusing here...

Solution seems to be get rid of the redundant particles.

tag	id	istat	moms		daus		tag	id	istat	moms		daus	
1	11	3	0-	0	0-	0	13	-24	2	7-	7	15-	16
2	-11	3	0-	0	0-	0	14	24	2	8-	8	17-	24
3	11	3	1-	1	0-	0	15	15	2	9-	9	25-	26
4	-11	3	2-	2	0-	0	16	-16	1	10-	10	0-	0
5	11	3	3-	3	0-	0	17	2	3	11-	11	27-	27
6	-11	3	4-	4	0-	0	18	21	3	11-	11	27-	27
7	-24	3	5-	6	0-	0	19	21	3	11-	11	27-	27
8	24	3	5-	6	0-	0	20	21	3	11-	11	27-	27
9	15	3	7-	7	0-	0	21	21	3	12-	12	27-	27
10	-16	3	7-	7	0-	0	22	21	3	12-	12	27-	27
11	2	3	8-	8	0-	0	23	21	3	12-	12	27-	27
12	-1	3	8-	8	0-	0	24	-1	3	12-	12	27-	27

(...)

Fixing inconsistencies

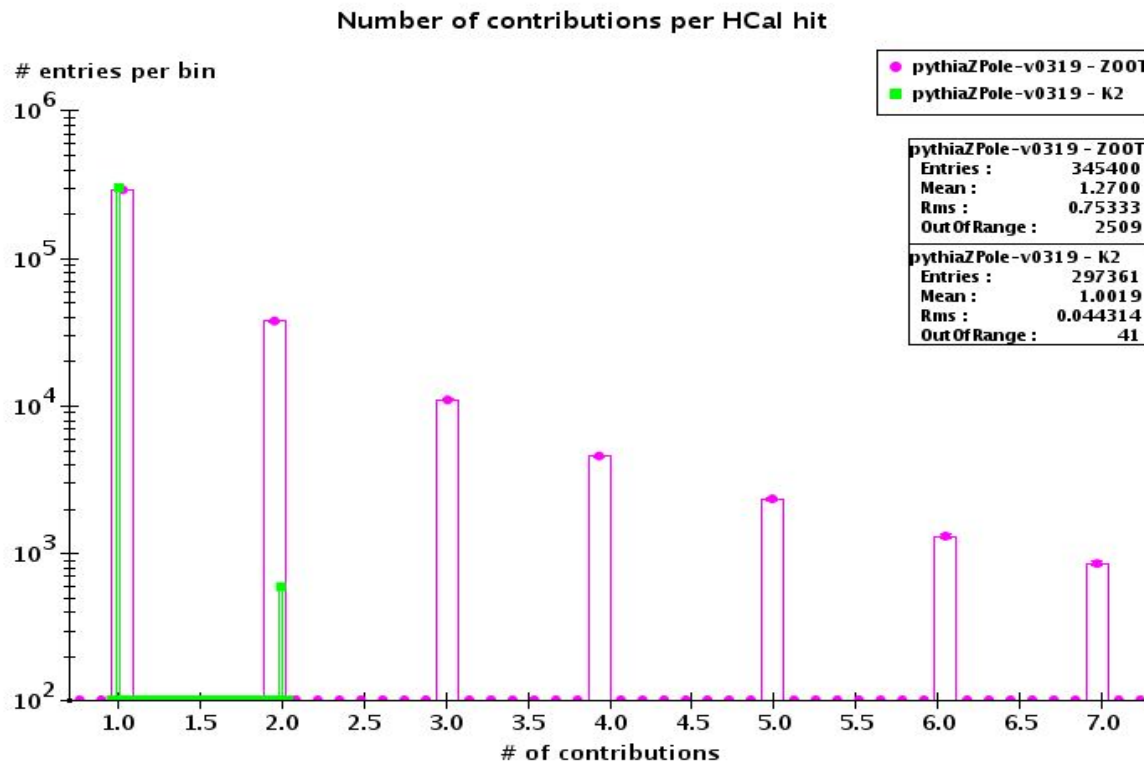
- **Stage 0:** fixes to generator status codes (all q,g,strings -> DOCUMENTATION; also, sometimes the only \pm s in event are labeled DOCUMENTATION -> INTERMEDIATE)
- **Stage 1:** confirm reciprocal relationships
- **Stage 2:** confirm parents for those without a parent after stage 1
No reciprocity required, provided both parent,daughter are DOCUMENTATIONS
- **Stage 3:** For those hadrons with a single quark parent after stage 2, add other quark/gluon parents starting from the stage 2 parent. Stop adding new parents when total charge of parents is zero, or before adding a parent which is neither a quark nor a gluon
- **Stage 4:** Only beam particles are OK to have no parents.
Use the generator-assigned parent info for all other (non-DOC) particles which still don't have links to their parents after the previous stages.
- **Stage 5:** Check for confusing Pythia feature: parents identical to daughters. Just keep one instance at the output. (implemented in v03-20 and 21, taken off for v03-22 for further tests.)

Note: The sequence above reflects current (mis)understanding.

New cases keep showing up, which are not fixed by this sequence...

A new problem

- Number of MCParticles contributions to a few calorimeter hits looks different depending on which cluster was used in the job...



Closer look: the same MCParticle have multiple contributions to the same hit!!

Different behaviour in different machines!?
More investigation is needed!

Some other things to do

- Fixes/improvements to LCIO SimStatusCode
pending on LCIO to make more flags available, like [showerInitiator](#),
[directContributionToSimCalHit](#). Could new flags be made available in LCIO?
- Non-projective fixed-size calorimeter cells for endcaps (SLIC?)
(Needs some thinking on cell numbering scheme)
- Simplify MCParticle handling (more portable, useful for SLIC?)
- Some XML extensions submitted by Shenjian Chen (UofC), to
allow for cell offsets between adjacent projective layers.
Release after tests with SDJan03, extensions to non-projective geometries.

LCDG4 status summary

- Only cylinders, disks and cones supported by current LCDG4 version (like in Gismo).
More realistic geometries to be implemented in another package (SLIC). Work is progressing well.
- Visualization framework within JAS for both .sio or .slcio files
- Source code available from SLAC or NIU CVS repositories
- Several MC physics samples have been generated for algorithm development and studies (both SIO or LCIO formats are available)
Anonymous access via ssh / scp:
[scpuser@k2.nicadd.niu.edu](ssh://scpuser@k2.nicadd.niu.edu) (pwd=lcd_2004): /pub/lima/lcdg4/v03-19
See <http://nicadd.niu.edu/~jeremy/admin/scp/> for detailed access instructions.
- For more information please check the LCDG4 documentation web page: <http://nicadd.niu.edu/~lima/lcdg4/>, or under subdirectory doc of CVS module

Summary

- Geant4-based full detector simulation tools available for detector R&D comparing different detector geometries and/or technologies.
- LCDG4 provides very fast response for basic changes to supported geometries.
- LCDG4 (XML-based) and Mokka (MySQL-based) can be used for standard LCIO format. LCDG4 also produces output in SIO format (old ALCPG format, still in use)
- LCDG4 follows preassigned decays from simulator. Some data samples cause problems though. A simpler MC handling scheme could help to prevent problems.
- Several data samples are available at NIU server for general use:
Anonymous access via ssh / scp:
scpuser@k2.nicadd.niu.edu (pwd=lcd_2004): /pub/lima/lcdg4/v03-19
See <http://nicadd.niu.edu/~jeremy/admin/scp/> for detailed access instructions.