Introduction to GLG4sim

An introduction to GLG4sim features

2005/06/14

G. Horton-Smith

GLG4sim: what is it?

- GLG4sim stands for "GenericLAND Geant4 simulation"
- LAND stands for Liquid-scintillator Anti-Neutrino Detector (as in KamLAND)
- It's a "Generic" Geant4 simulation for LANDs.
- It can serve as a library or a starting point for specific liquid scintillator antineutrino detectors.

GLG4sim: who wrote it?
Quoting from the ACKNOWLEGEMENTS file:

GLG4sim is derived from the most general parts of KGL4sim, a Geant4-based Monte Carlo for KamLAND started in 1999 at Tohoku University by myself (Glenn Horton-Smith) and Haruo Ikeda.

Contributors to KLG4sim between 1999 and 2004 have included, in alphabetical order, S. Dazeley (Tohoku, LSU), J. Detwiler (Stanford), G. Horton-Smith (Tohoku, Caltech), L. Hsu (LBL), H. Ikeda (Tohoku), T. Iwamoto (Tohoku), K. McKinney (Alabama), D. Markoff (TUNL), D. Ray (LBL), R. Rohm (TUNL), O. Tajima (Tohoku), B. Tipton (Caltech), and Y. Uchida (Stanford). It is important to acknowledge their valuable contributions to KLG4sim.

(see ACKNOWLEGEMENTS file for more.)

GLG4sim features

- Completely functional simulation.
- Supports multiple detector options; cylindrical and spherical “generic examples” provided.
- GLG4DataInputReader class for reading in G4MaterialPropertyVector data from an external file: `materials.dat`
- GLG4param class implements trivial database of detector-defining numbers, read from file: `settings.dat`
- GLG4DebugMessenger adds some useful tools.

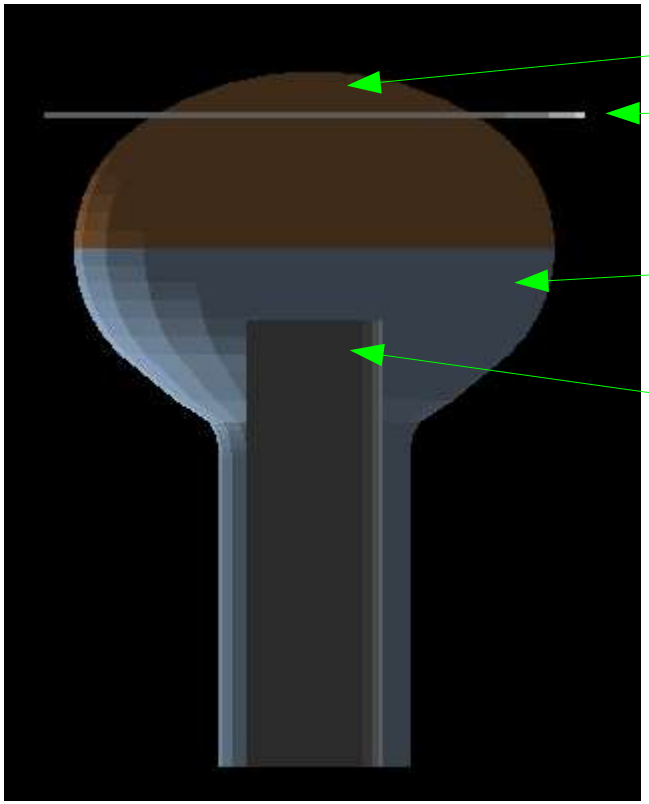
GLG4sim features, continued

- GLG4Hit* classes collect hit data efficiently.
- Support for multiple, easily customizable output formats.
- GLG4VertexGen classes provide a variety of event types, including external HEPEVT-like input.
- GLG4PosGen classes allow fixed point, surface paint, volume fill for event location, and more.
- GLG4PrimaryGenerator, GLG4DeferTrackPos work together to provide event mixing, correlation, separation.

GLG4sim features, continued

- GLG4OpAttenuation includes scalar scattering.
- GLG4Scint class implements scintillation process with more options, “bulletproof” energy deposit catching, optical photon reemission.
- GLG4PMTOpticalModel implements complex semi-transparent thin cathodes.
- GLG4TorusStack special shape for PMTs, domes.
- GLG4_PMT_LogicalVolume implements construction of Hamamatsu 17/20-inch and 8-inch PMTs and ETI 5-inch PMTs; extendable.

PMT shape



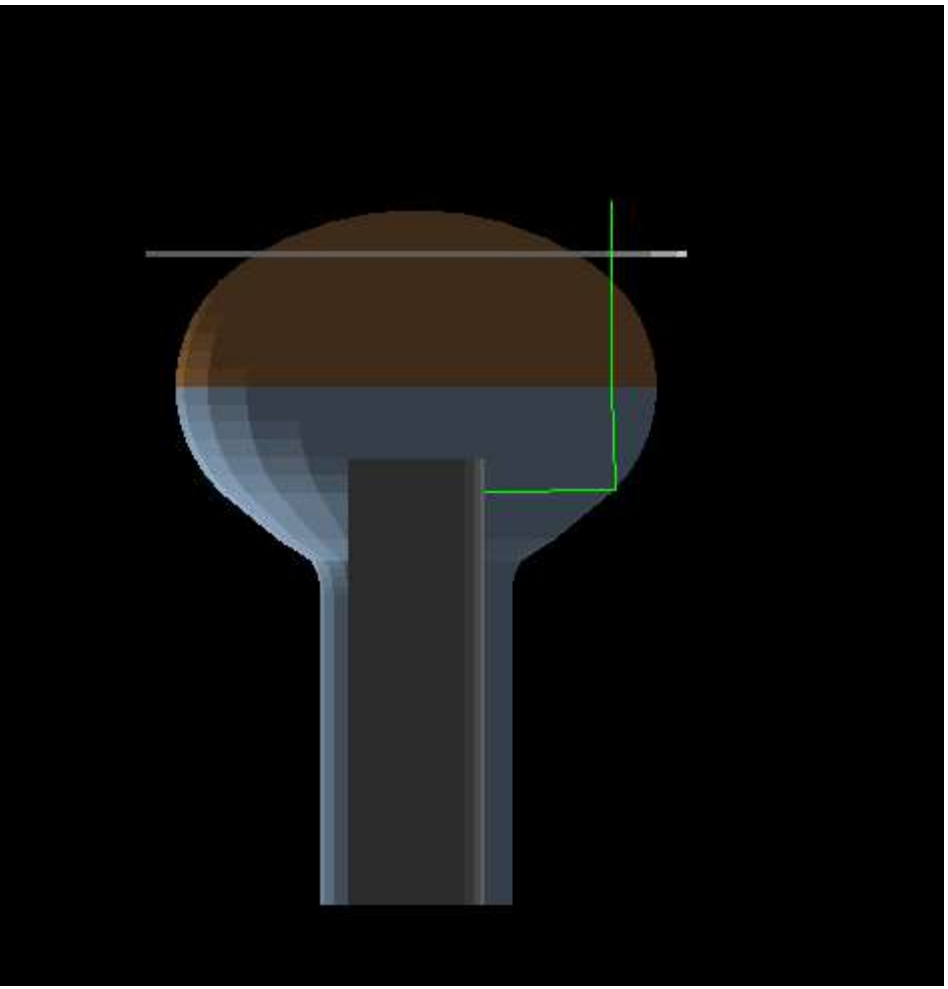
Cathode

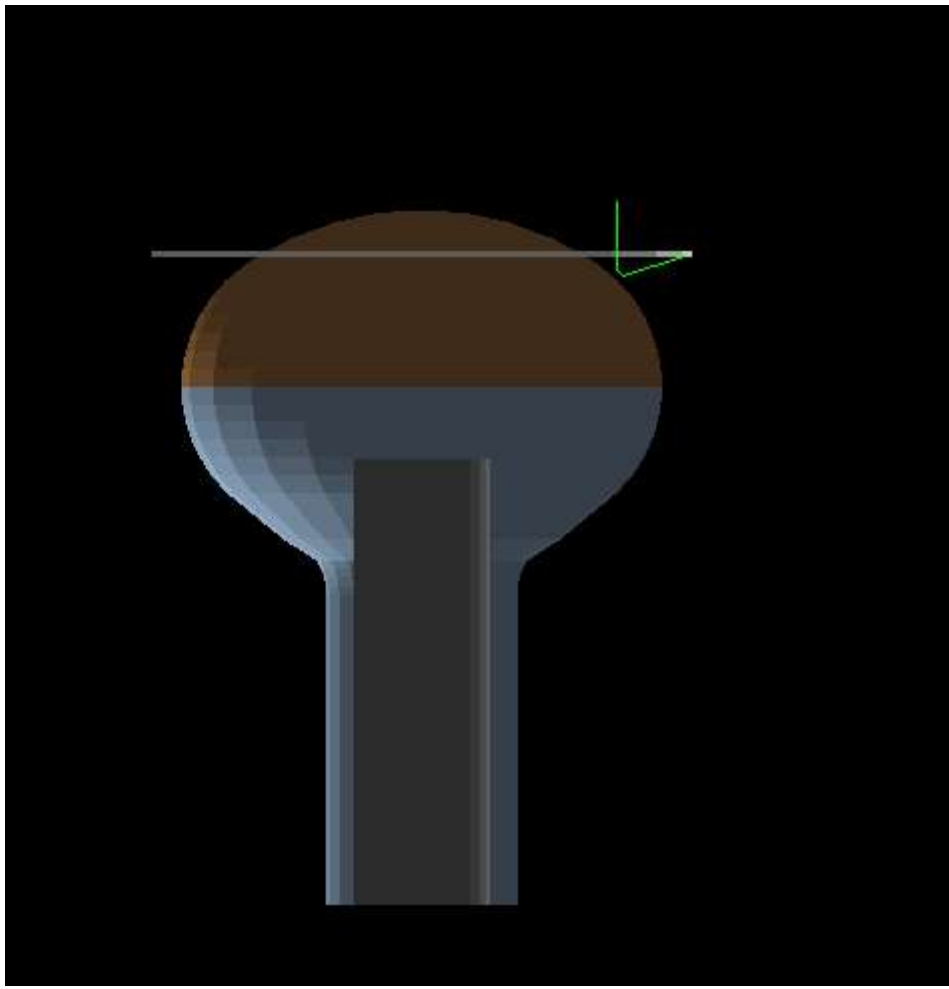
mask

silvered surface

dynode crudely approximated as solid cylinder of low density metal

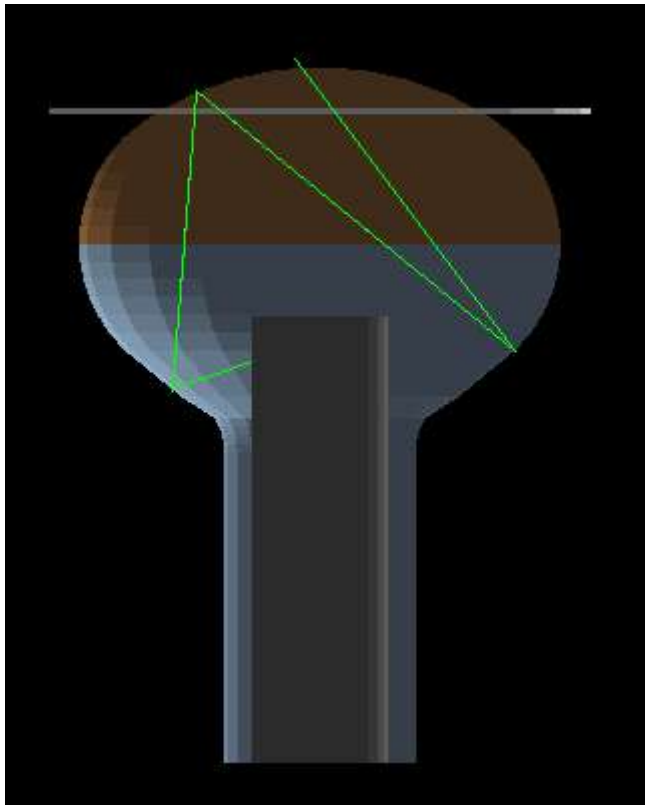
PMT optical model

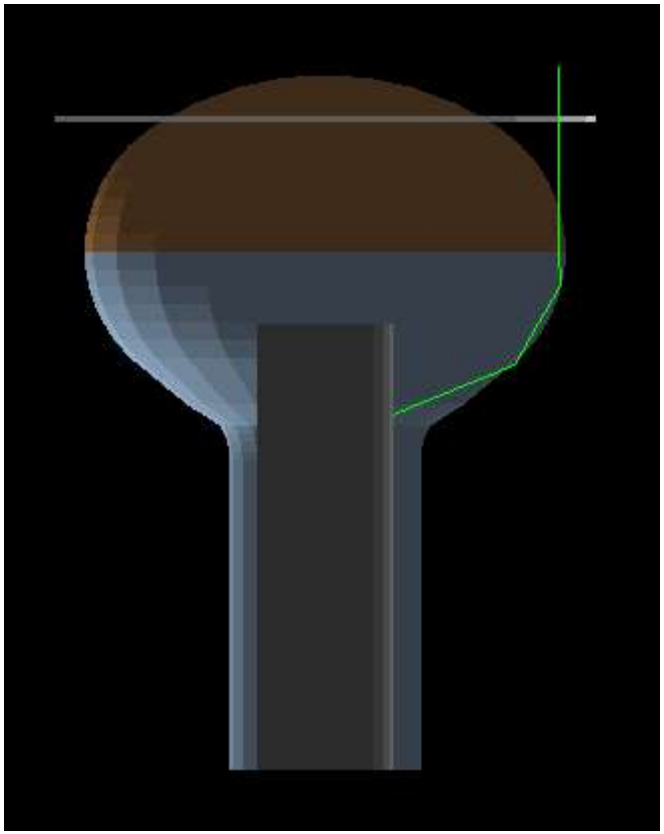




Light reflects or transmits at most dielectric boundaries as per Geant4 process; reflects, transmits, or absorbs at photocathode based on complex index of refraction and thickness. (Math from SNO paper by M.D. Lay, et al.)

PMT optical model





Can have internal reflections in glass, etc.

(And yes, mask is placed too high here.)

GLG4sim speed

- With full optical photon tracking, can simulate 10000 1-MeV events in 80 cpu-min on 3.2 GHz P4/Xeon.
- CPU-time proportional to number of photons tracked, works out to 0.5 cpu-sec/MeV.
- Can run with optical photon tracking disabled, in which case it just tracks total energy deposit in scintillator (with and without quenching) and "centroid" -- orders of magnitude faster.

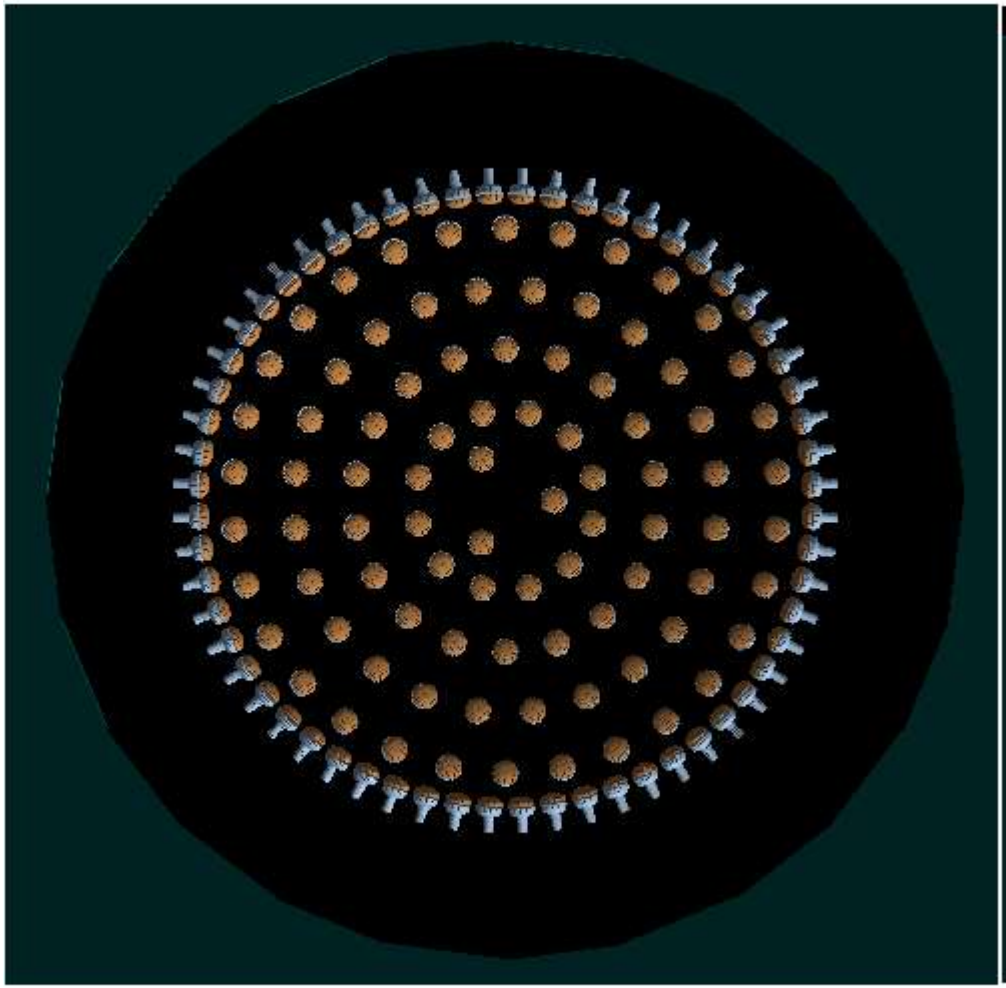
User support

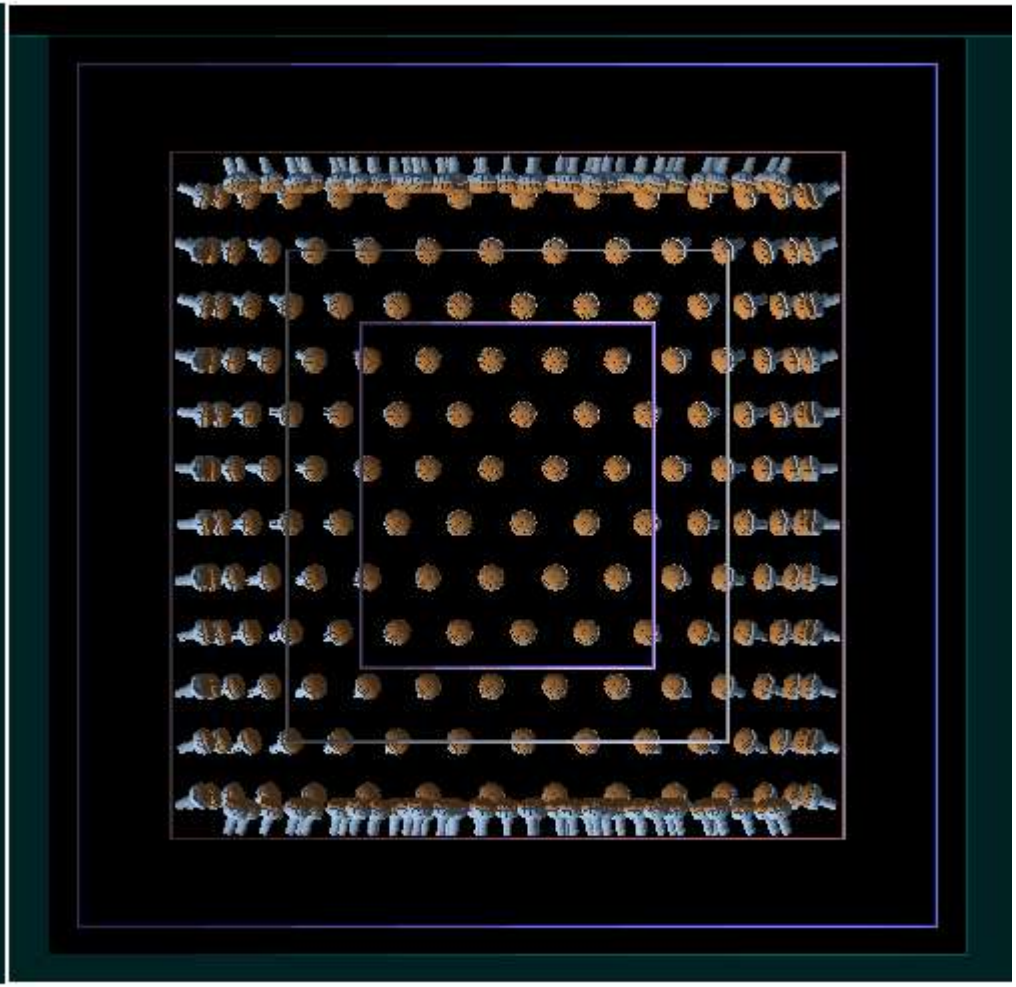
- User supported.
- Website at http://neutrino.phys.ksu.edu/~GLG4sim/
- Current user base:
  - Daya Bay (L. Hsu, LBL, used as starting point)
  - Braidwood (D. Thompson, KSU, used for inspiration)
  - Double Chooz (Motta, LoSecco, etc., used as library)
  - KamLAND (“organ donations”)
- Bugs are being found, fixed!

Status
as known as of 2005/06/12

- Current release (0.9.4) is functional, with some known bugs, none too serious.
  - photon hits before front of distribution not merged properly in HitPMT
  - stainless opsurfaces not used in some calibration device surfaces
  - arbitrary hard-coded constants for PMT “mirror” optical surface
  - hbook option does not compile
- Works with Geant4.7.0 and 4.6.2-p02.
- Documentation is a slow work in progress.
  - Getting it working and understanding it still requires way too many e-mails to me. I am motivated to fix that. Please help me if you can!
- “Dusty corners” in code unimportant to KL...

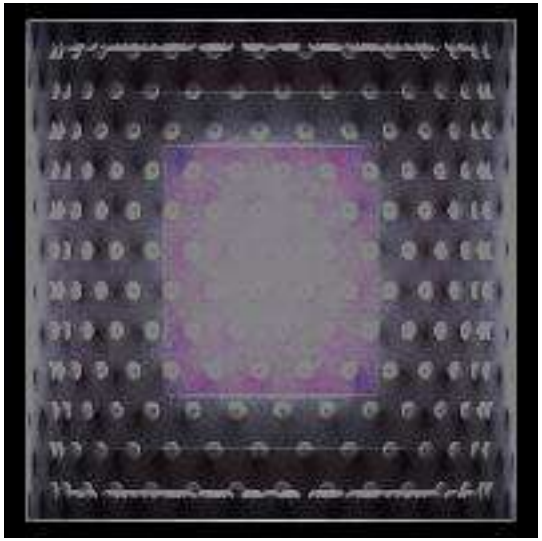
Section views



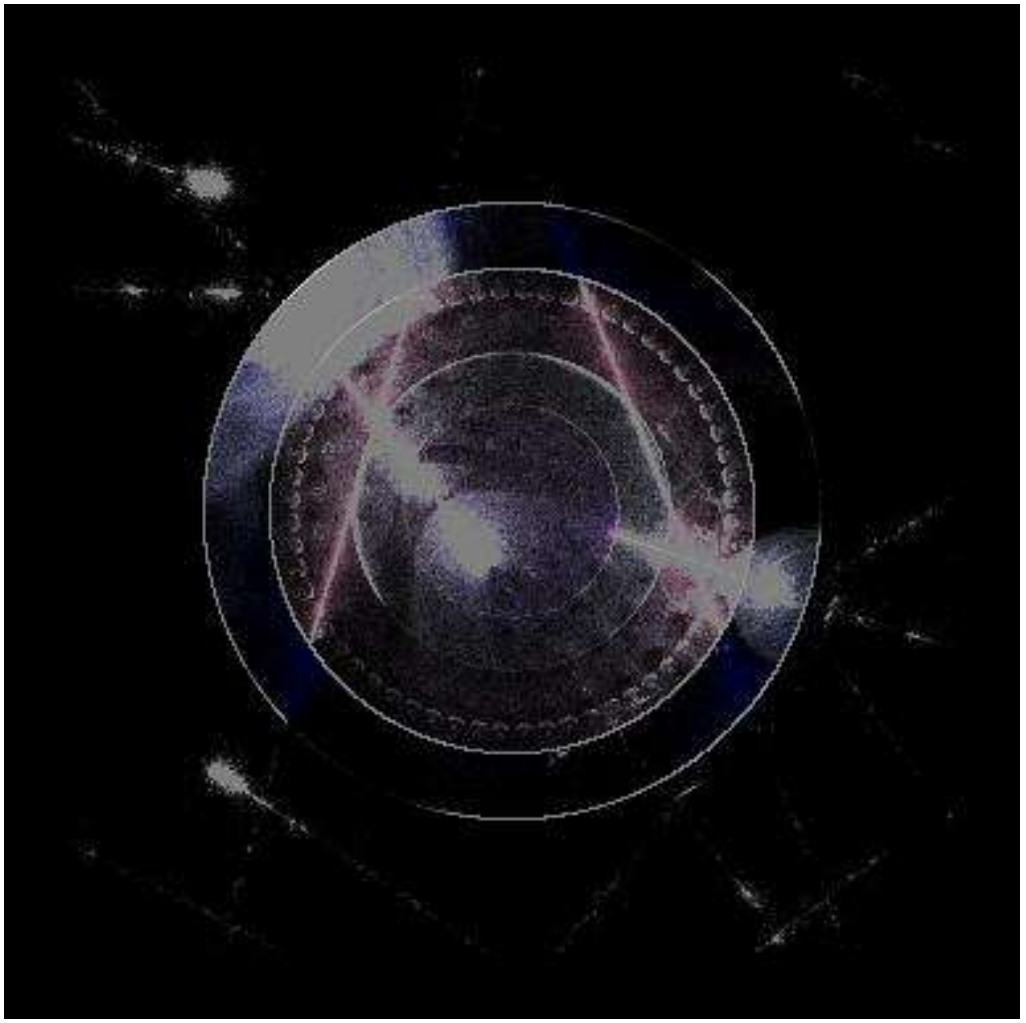


“Self-illuminated” views

To create, compile with G4DEBUG option, and use /glg4debug/dump_illumination_map command after simulating some events. Output in map?.ppm.

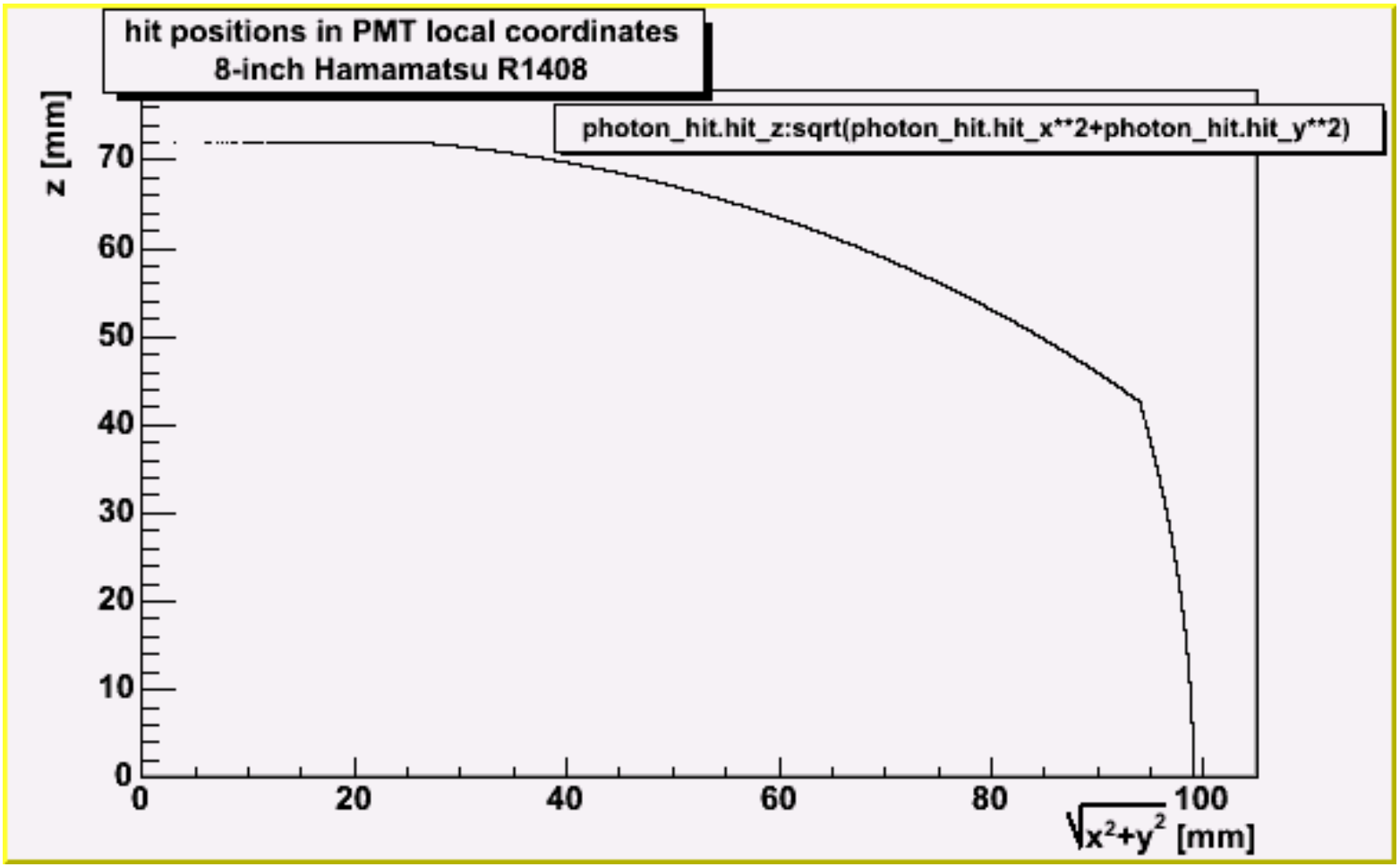


Side view, positrons uniformly filling central volume



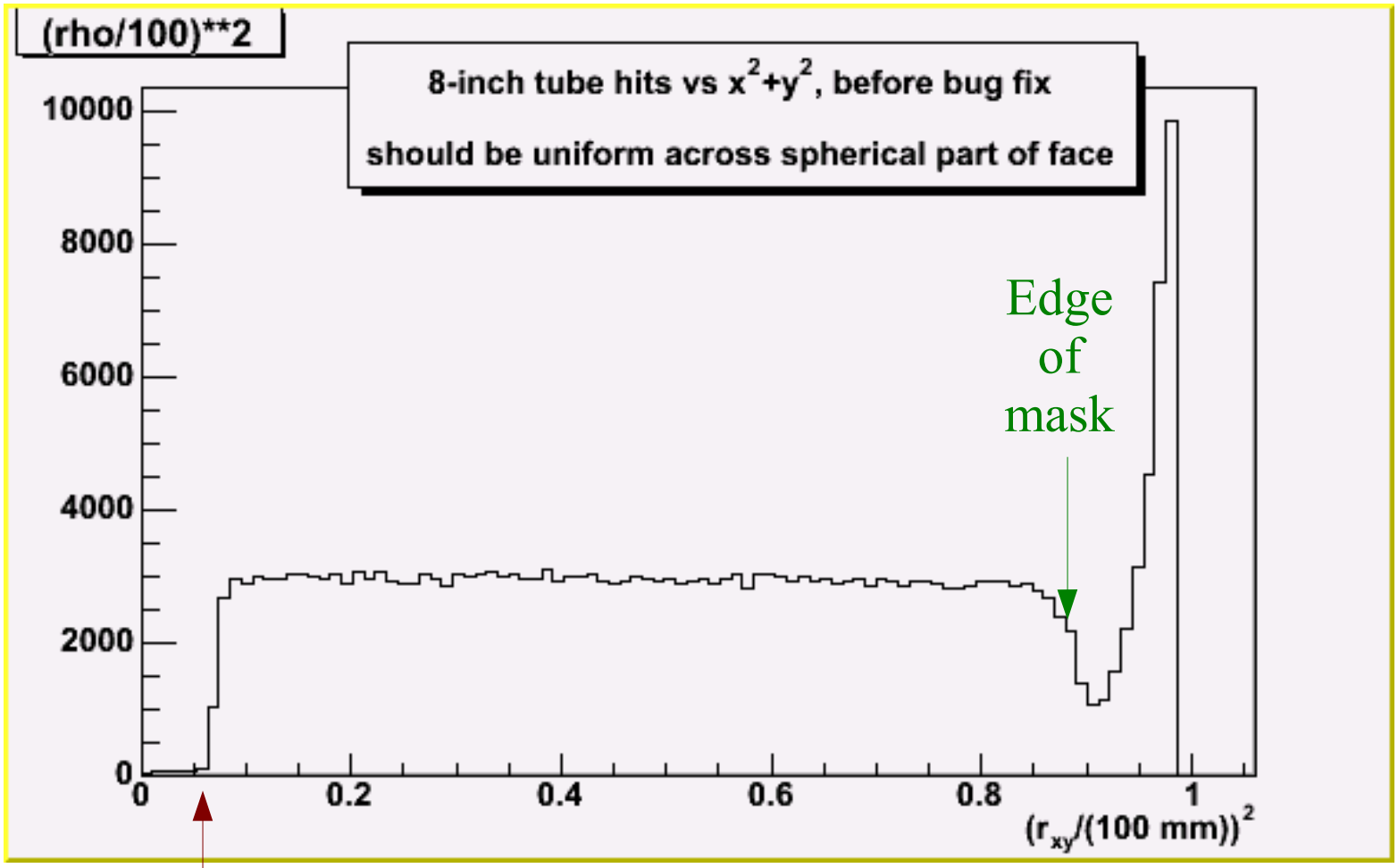
Top half, muon_test_short.mac

Bug discovered (rel 0.93) in plot of position of "hits" in PMT local coordinate system



Apparently due to error in R1408 diagram or my reading of it.

Plot x^2+y^2 should be uniform over hemispherical part...

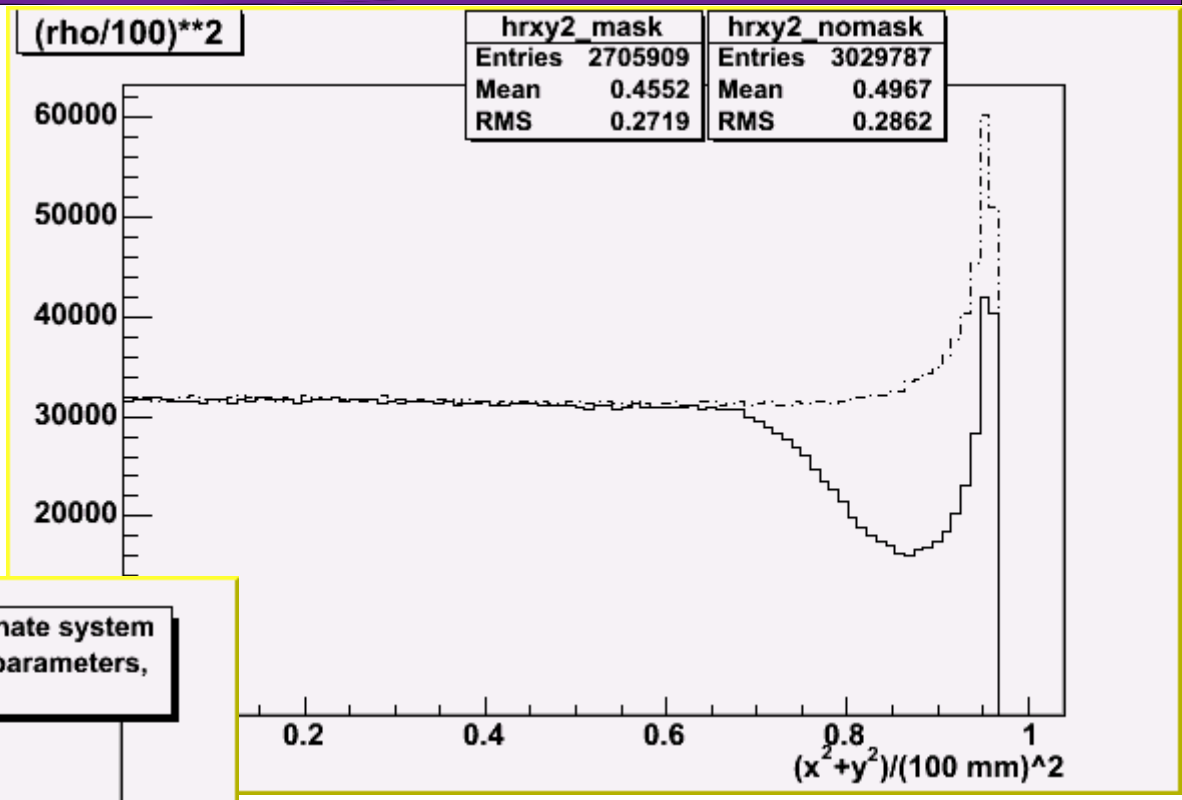


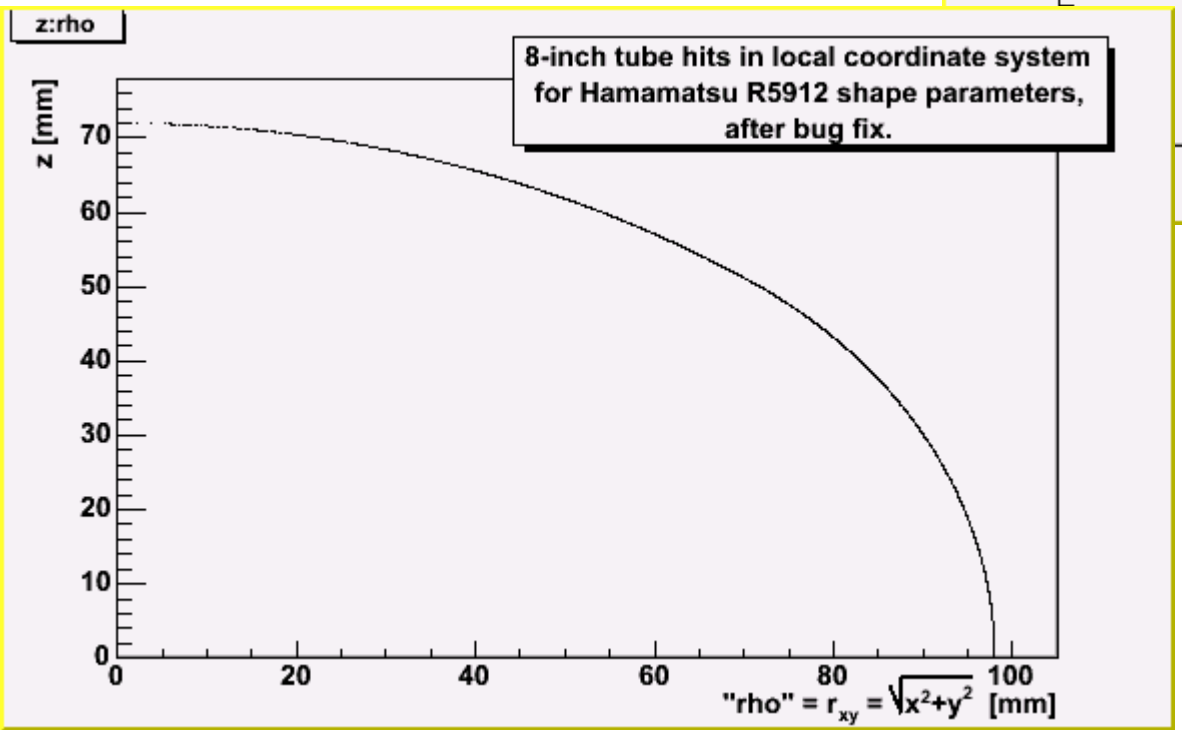
Face of tube exiting bounding cylinder?

Yuck!

After fixing 8-inch PMT shape error

Uniformity of hits across tube now consistent with geometry (but not LSU measurements, needs new code feature to implement)





Position of hits consistent with desired shape.

On the "To Do" list...
(desired new features, in no guaranteed order)

- GLG4Scint upgrade to use better Birk's model for shape-dE/dX dependence
- improve the generators interface
- PMT photocathode response vs. position as explicit table (charge collection efficiency in addition to M.D. Lay model)
- Improve code structure and documentation!