OLYMPUS Monte Carlo

Still early days

- detector simulation basic rather than detailed
- hits smeared with resolutions but no digitisation
- basic generators available: ep, Möller
- simple reconstruction from hits to p, z, θ, φ

Code needs to be tidied

- problem with people adding code without good knowledge of GEANT4 or close supervision (my fault)
- e.g.
    - ProgOptions - already removed
    - EventInfo
    - PathSwimmer
    - resolution smearing
- code entangled
    - EventInfo and Reconstruction

Fix now or it will get worse and worse

GEANT4

GEANT4 has lots of tools and capabilities

- built in and supported by a large community
  - we should not write our own code when better code exists
- documented, tested, and examples on use

GEANT4 provides

- shapes to build detectors - CSG, BREP, boolean, CAD, …
- materials
- primary event generation - single particle, HEPEvt, HEPMC
- sensitive detectors - hits, scoring, …
- physics - EM, hadronic, production, fragmentation, …
  - reflection, refraction, transition radiation, ...
  - regions, event biasing, ...
- visualisation - OpenGL, HepRep, Dawn, OpenInventor, VRML, …
  - user defined GUI
- user interface - batch and interactive control, modifying, …
- analysis - ROOT, AIDA, …
- scientific software libraries, CLHEP, fitting, random numbers, ...
- ….

GEANT4

Is trivial to use

- compile, run, analyse results, …
- no different from physics experiment
    - don't need to know how wire chamber, PMT, trigger, electronics, DAQ work
    - just analyse data

However, it is not trivial to code properly

- learning curve steep because of the number of different aspects in a complete, fully developed program
- just like an experiment
    - need to know ….

GEANT4 programmer

- should know C++ or at least object oriented programming
- should understand all the manuals and examples
- should attend the GEANT4 workshop
- should talk to an expert before adding / changing an existing code

Plans for OLYMPUS Monte Carlo

Separate Monte Carlo and Reconstruction

- generate events in Monte Carlo - perhaps sufficient for some needs
- reconstruct MC events separately using GEANT4 tools
- could be in same program but separate option controlled by user
  - currently MC runs then reconstruction reads MCData file and produces ReconData

Remove EventInfo

- GEANT4 has UserRun and UserEvent classes which know everything about the run or event
- EventInfo effectively duplicates this and appears in almost every routine plus is entangled with reconstruction

Remove PathSwimmer

- GEANT4 tracks particles in magnetic field - use it
  - better, faster
  - incorporate energy loss in fitting
  - automatically detects hits

Further Plans

General clean-up

- good coding practises
  - guidelines for coding
- comment code (more below)

User interface

- GEANT4 has UI to control physics, event generation, visualisation, …
- can add user defined controls
  - beam energy
  - particle species
  - magnet polarity, field strength
  - number of events
  - reconstruction
  - anything you would like to change without getting silly
- either interactive or through batch command files

Once this is in good shape

- release for general use (can use it now but will change)
- start development to make more realistic

OLYMPUS Monte Carlo

Available from the BLAST computer cluster at MIT-Bates

- need account on BLAST computers
- however, code does not run on BLAST machines
  - hope to change this soon
  - need to have GEANT4, CLHEP, ROOT, gcc, etc. on your machine

Uses SUBVERSION for code management

- similar to CVS but some nicer features
  - will publish a basic description and instructions
- initially checkout latest version from repository
- compile and run
  - change number of events, energy, resolutions, etc.
- periodically check for new versions and update your own version
  - for average user this should be it
- for a programmer
  - discuss with coordinator changes you are planning hopefully using existing tools
  - test, document, and when you are sure commit changes to repository

Strongly recommend coordinate act as librarian as well

Documentation

Use Doxygen

- documentation system for variety of languages including C++
- used by GEANT4 and numerous other large collaborations

Doxygen

- will document code even if people do not comment code
- if people add reasonably simple comments to the code Doxygen provides very nice, complete documentation on usage, etc.
- some special comment formats can be used to enhance and extend documentation
  - provide overview, guidelines
- html and latex output

Monte Carlo Simulations with Collimator

Work by Axel Schmidt (MIT)

- incomplete due to illness

Collimator

- 10 cm diameter tungsten cylinder, 10 cm thick
- elliptical hole 25 mm horizontal, 7 mm vertical
  - 1 mm smaller in all directions from target cell

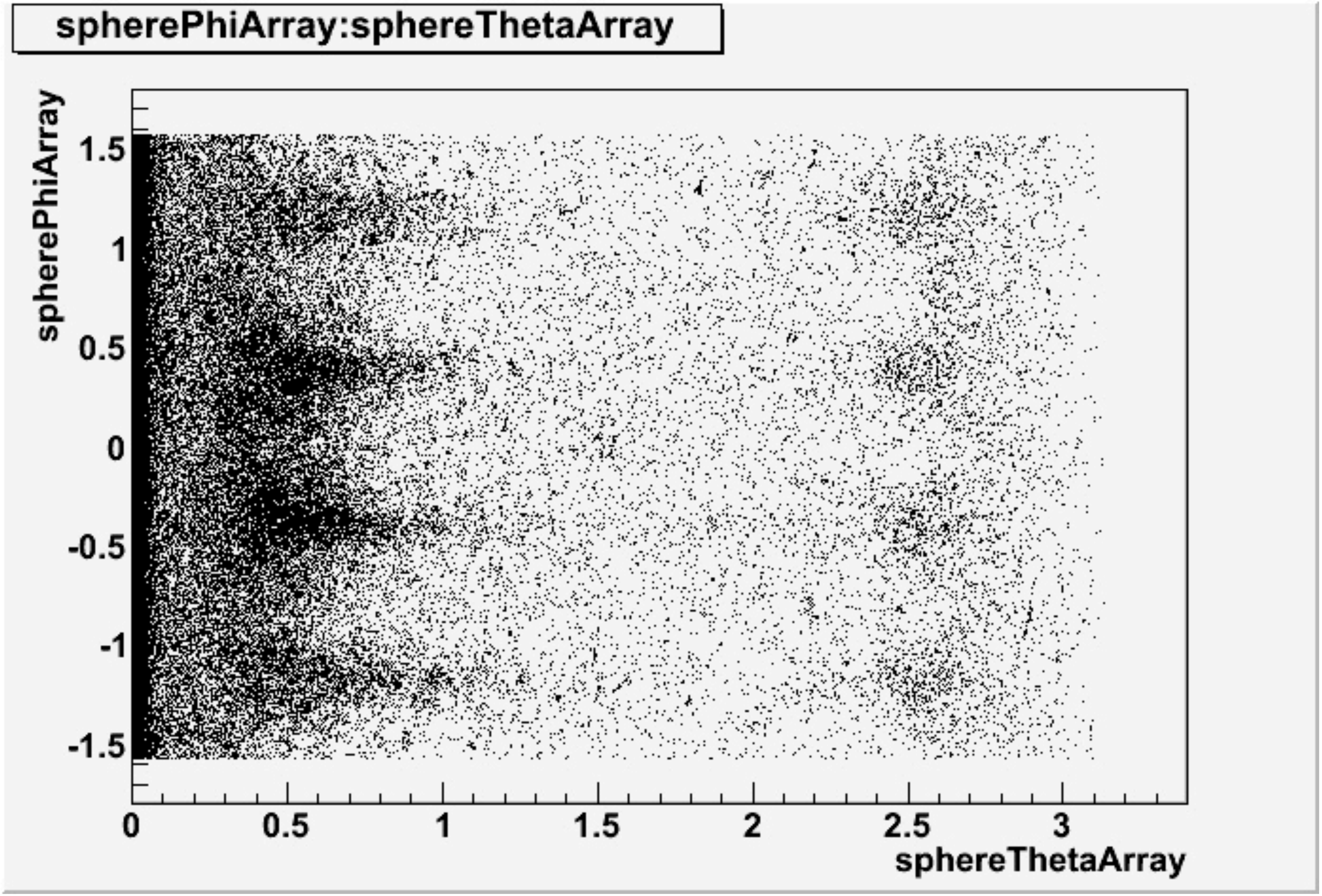
Simulate beam electrons striking upstream face

- distributed as tail of gaussian distribution at 10 sigma
- start 100 microns inside hole and then out from there

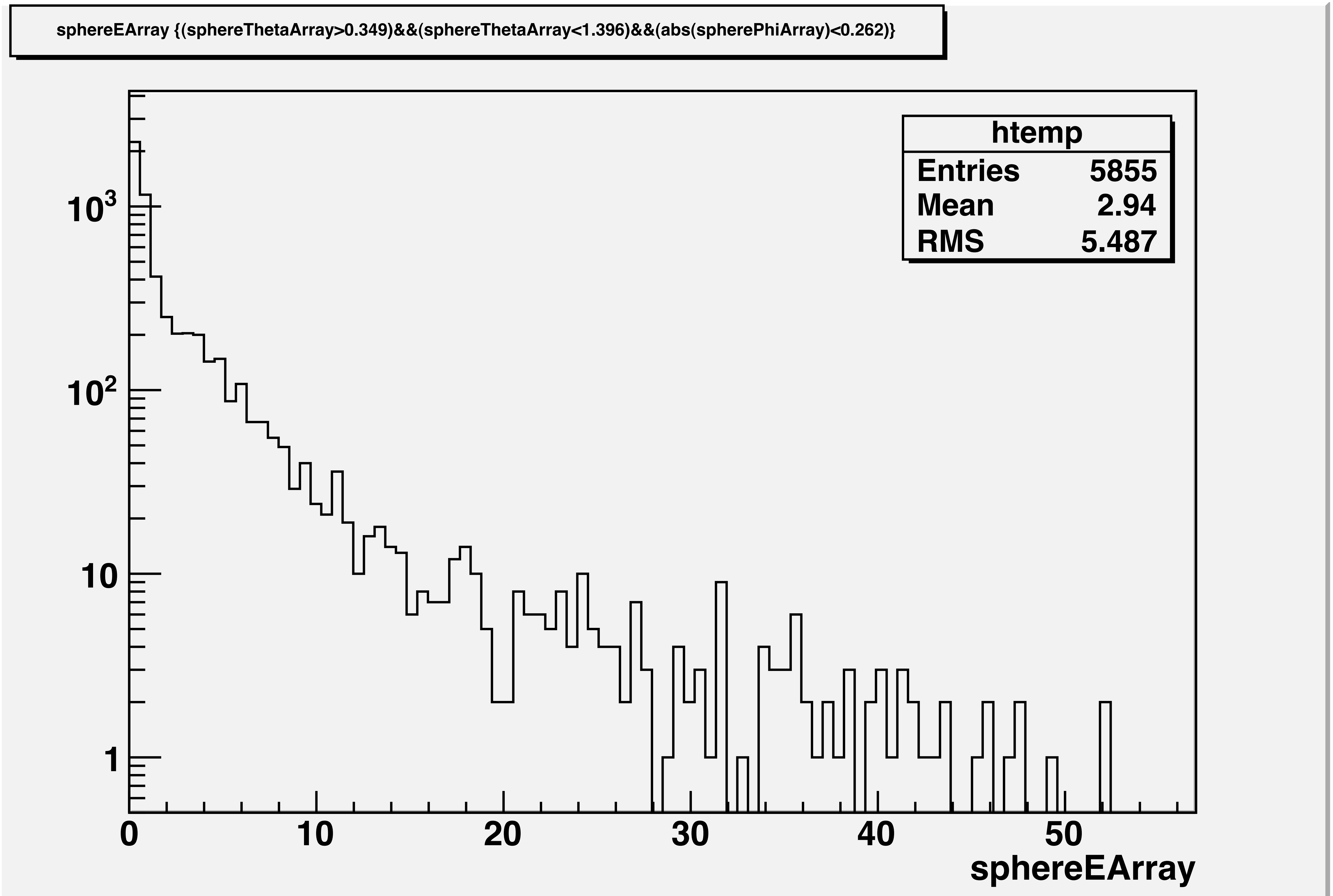
Measure energy flow into sphere 1 m in radius

- OLYMPUS magnetic field is present but no coil or other material

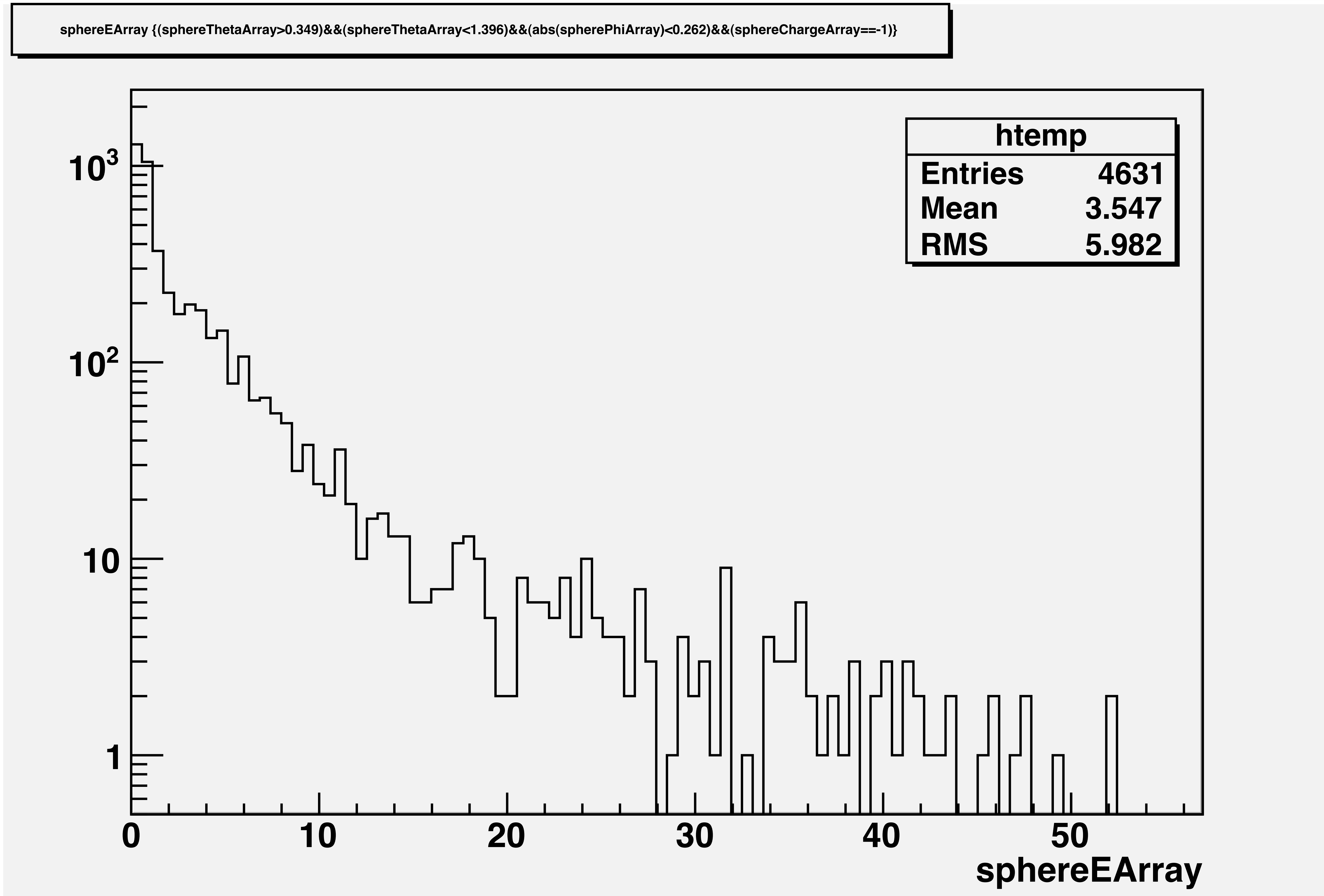
Azimuthal versus Polar Angle



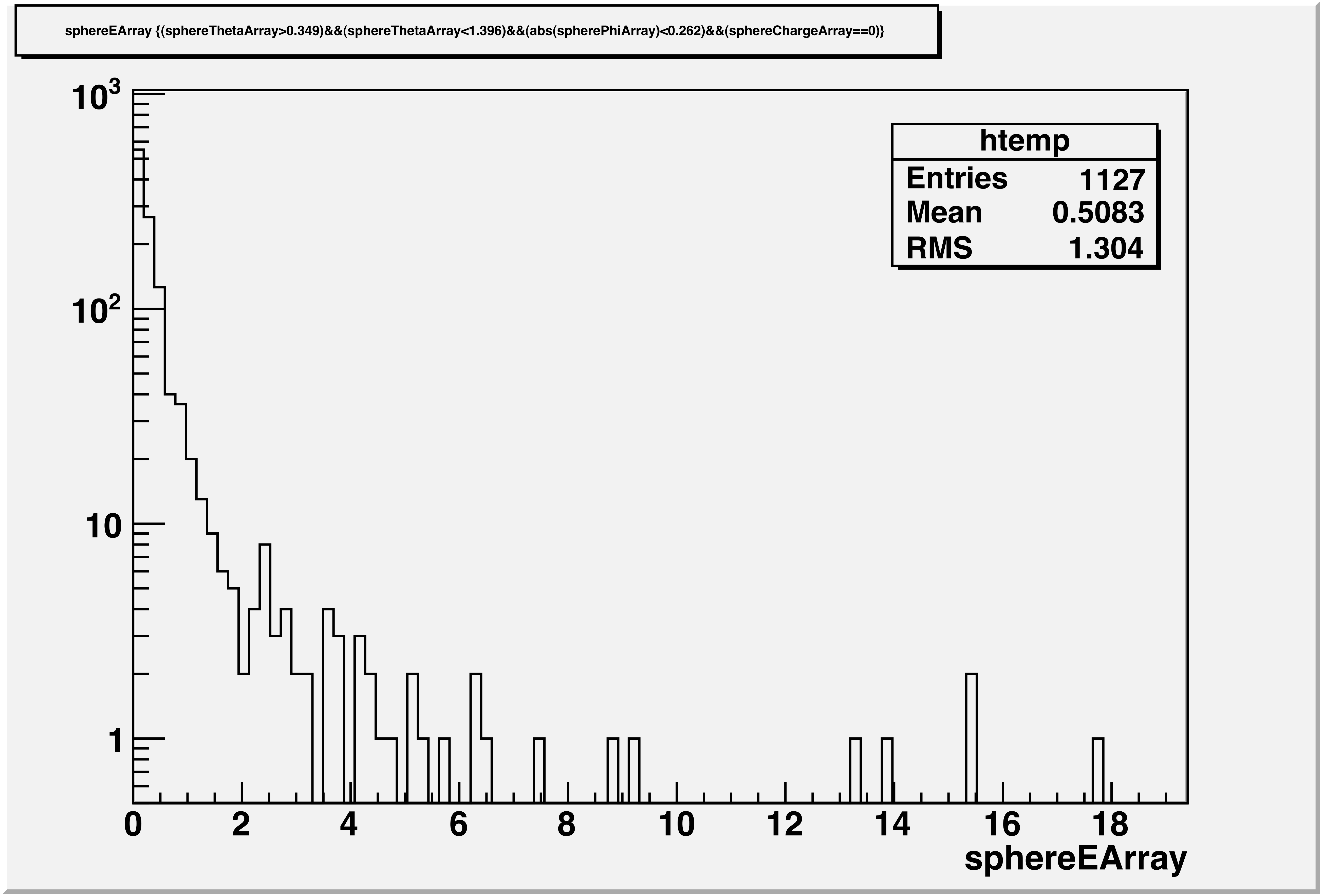
In the Wire Chamber Acceptance



Electrons Only



Photons Only



Collimator Studies

Still lots to do and check

- determine source of events into wire chambers
- simulate dispersion of beam
- determine time distribution
- check rate
  - assuming beam lifetime due to losses on collimator
  - calculate number hitting collimator per second to determine rate in detector
- taper and flare collimator aperture

Synchrotron Radiation

2 GeV

- 1.4 x 10^{16} photons/s
- E_C=1.6 keV
- power on collimator
  - 5 W
  - 0.16 W with horiz. scrapper

4.5 GeV

- 3.2 x 10^{16} photons/s
- E_C=16 keV
- power on collimator
  - 120 W
  - 38 W with horiz. scrapper
  - 21 W with vert.->horiz. scrapper

Sync. fan passes through target cell and beam line.