Ben Locke's Legacy Documentation on Software Development for

- **MTS Alignment in Data**
- **GEM Spatial Resolution Measurements in Data,**
- **GEANT4 "Grand" Simulation of MTS**

Fall 2011

All software is available on the Grid cluster. Please do not modify the code in Ben's area, but copy it to your own working directory first and then modify only your own copy.

1. ALIGNMENT

The most up-to-date version of the alignment code is found in

/mnt/nas0/home/g4hep/geant4/examples/mytestapps/benL/cubicFootRealAnalysis/alignmentAndResolution/realData20Dec2011_SideDetectors

This is run as a compiled C++ program, so run it with:

make clean

make

./alignment config.txt

That's it.

The config file is self-explanatory for the most part. Several parts of the configuration file are depreciated and are marked as such; I have not deleted them in case they need to be resurrected in the future. The detectors will automatically be detected from the data.

I have switched to using the closed-form 2D fit for the lines. I have been unsuccessful in writing a chi^2 function for this. So, don't trust any ch^2 results. Perhaps this would be something easily fixable in the future.

If you want to modify something, I suggest looking in alignment.cc and/or Analysis.cpp first, as this is where the meat of the code resides.

As in the case with resolution, be sure you have a strong understanding of the code from reading it before you do anything major. Every part is interconnected.

2. RESOLUTION

The resolution studies for the cubic foot station were performed here:

/mnt/nas0/home/g4hep/geant4/examples/mytestapps/benL/cubicFootRealAnalysis/alignmentAndResolution

Each subdirectory in here is different and contains various versions of the code. The best one to look at would probably be realData17Sept2011_BestResolution/

The resolution studies were performed with a ROOT script, not a compiled C program. In this case, you want to look at alignmentFinalForRealData.C. Before you change anything or go off and try to modify this for future studies, make sure you understand everything in this script.

To run it, type

root –l

.L alignmentFinalForRealData.C+

align(args)

That's it. The args can be found by looking at the definition of the function align() in alignmentFinalForRealData.C.

The output is a series of pngs of residual histograms and various other data. Also, text files containing the means from the iterations are produced for each detector. Generally, I put these data into an excel plot, and one can see the evolution of the means as they are shifted. Other data, in addition to the means, are also available in these files and are labeled therein.

Helpful hints:

Make sure your data is formatted correctly.

Try a different version of the code if you can't get the one above to work.

This version of the code does not work with side detectors.

If you get weird results, make sure the detectors are declared properly in the script. You should see a series of

Detector(…);
Detector(…);
…
Detector(…);

Change the above series to what you need for your applications.

3. GEANT GRAND SIMULATION OF MTS

The most current working version of the grand simulation can be found here

/mnt/nas0/home/g4hep/geant4/examples/mytestapps/benL/grandSimulation4

Note that POCA is still not output and this must be implemented. Detector hits are output and coverage data re output when coverage is chosen as the simulation type.

The configuration file is included in the above directory and is well-labeled.

To produce the visualizations, uncomment all /vis/ options in the config file.

When adding materials to objects in the volume, be sure to use only materials found here:

http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/apas10.html

or create your own in the Materials class.

The following are how to set up the various options configuration file.

/myDet/setActiveVolume 300 300 300 G4_AIR 0 0 0 1 0 0 0 1 0 0 0 1

Label sizex[mm] sizey[mm] sizez[mm] material rotationvetorx rotationvectory rotationvectorz

/myDet/setScintillator top 604 456 20 G4_POLYVINYL_CHLORIDE 0 81 507.5 1 0 0 0 1 0 0 0 1

Label name[string] sizex[mm] sizey[mm] sizez[mm] material positionx[mm]wrtActiveVolumeCenter positiony[mm]wrtActiveVolumeCenter positionz[mm]wrtActiveVolumeCenter rotationMatrix

Targets are set up the same was as scintillators.

/myDet/setDetector top1 200 200 1.3720 43.62 318.95 1 0 0 0 1 0 0 0 1

Label name[string] sizex[mm] sizey[mm] position[mm] position[mm] position[mm] rotationMatrix

Ensure you follow all other instructions in the configuration file.

Helpful hints:

If your simulation crashes, check for overlapping volumes.

If you are running coverage, remember you can only parameterize the active volume, and there can be nothing inside the active volume. If a coverage simulation crashes, ensure your active volume size is an integral multiple of your voxel size.

Also look in my directory to fins the file makeScript.cc. This is very useful when you want to distribute your simulation throughout many smaller simulations. Remember, coverage simulations cannot be distributed.

If you are getting weird results, ensure your CRY plane is significantly larger than your station and rests just above the uppermost scintillator. If now, you will not get optimal POCA reconstructions, and if you are running coverage, your volume will not be "filled out."

Things to do to finish the simulation:

Add POCA to the output. Try using the 3D fit algorithm in the alignment and resolution studies.

Decouple coverage from the parameterizing the active volume with physical volumes. It can be done mathematically by getting the center of every voxel and testing the position of the muon at each step and seeing which voxel the muon is closest to.

Add shielded targets to the config file.