Chamber and Tracking House Cleaning....

Several people have been working on reconstruction in the chambers (BC's, DC1-4, MWPC's)

Time to put this all into a common framework we can all use

Andre L. and I sat down over the last two days and worked out what structure for the geometry and tracking makes the most sense. Idea is to move common code that everyone interested in looking at the wire chambers needs into a common area

Geometry

Currently have two geometry classes for BC+DC and MWPCs
These have been unified into a single class (GChamGeo) for both
After some warnings, the old classes (GDCGeo and GMWPCGeo) will become obsolete. You've been warned!!

The new geometry adds some nice features several of which that have been duplicated in several places:

- No longer assumes perfect geometry for chambers
- wire to u mapping (u is distance to beam axis along plane measuring direction
- track (x,y) to u conversion (also accounts for chamber rotations)
- wire1,wire2 -> (x,y) crossing point

If you use these type of functions I would encourage you to move to using this class instead of rolling your own...

Tracking Finding and Fitting

Again, several people are working on this and I thought it was time to add some more structure to the code. This will help us make better use of each other's work and reduce duplication.

Start a new package “TrkRBase” (Track Reco. Base)

This will contain classes for:

- “Hit wires” - wires with calibrated times relative to t0, selected as not being in the bad channel lists for the run/subrun. Stored in the evt.Calib() branch.
- “Wire Clusters” - one or more wires from the same plane and same chamber which together measure the track position in a given view
- “Wire Crossings” - one or more wire clusters from different views which together measure an x,y point
- Fitters for straight lines in 2 and 3D for sets of wire clusters