EVD endpoint projection update
+ Geant4 endless loop debugging
+ SingleGen changes
(+ DatabaseUtil update ?)

Andrzej

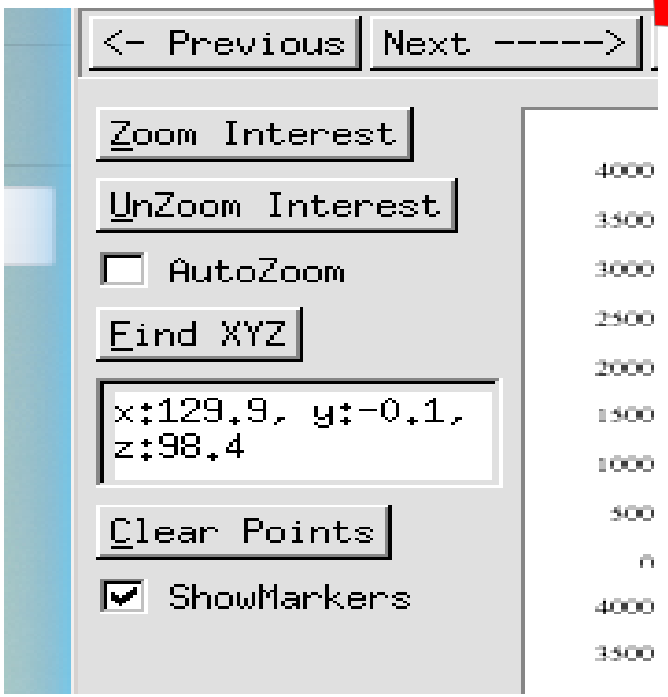
Track End Point projection in the EVD

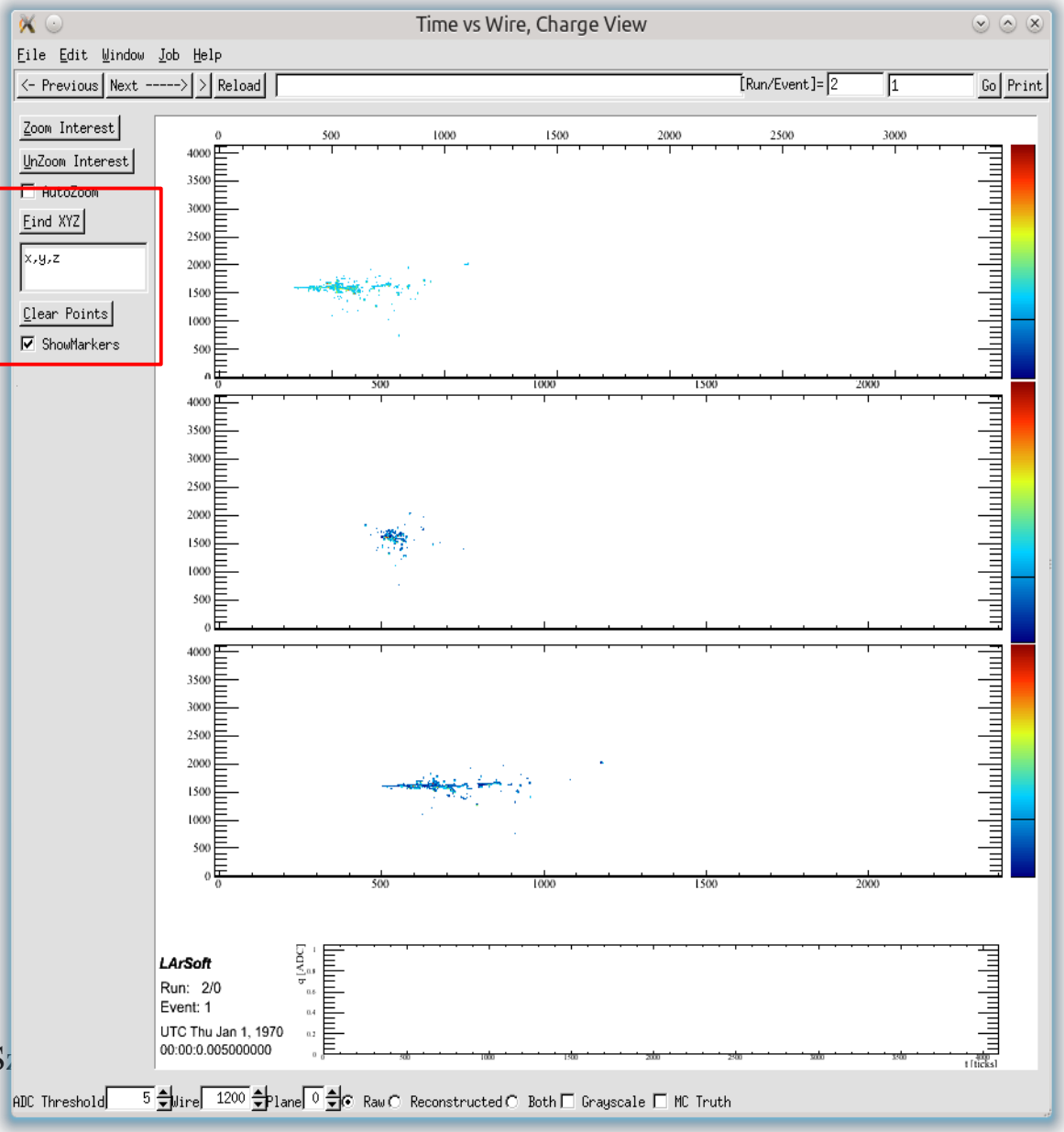
Track EndPoint in a LArTPC

- It's always hard to understand where a given track ends in the TPC (at least for me) due to the wire plane geometry.
- This information is particularly useful for hand scans etc. since knowledge that a track is exiting can help understand the event.
- The idea is that you could click on a point and know if it is close to the edge or not (or where in the chamber).

Addition to the EVD sidebar

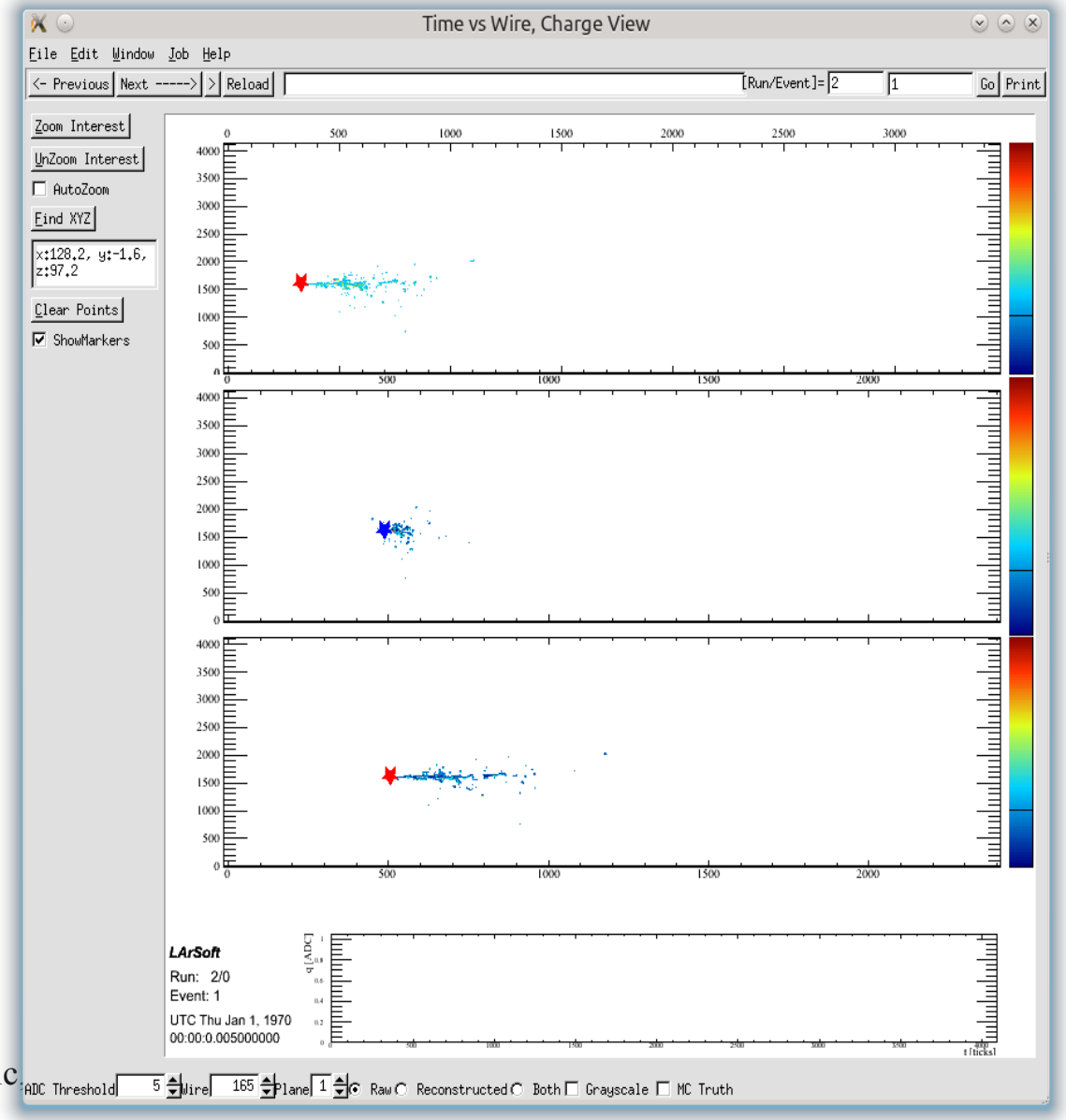
New elements in the sidebar + new mouse action.



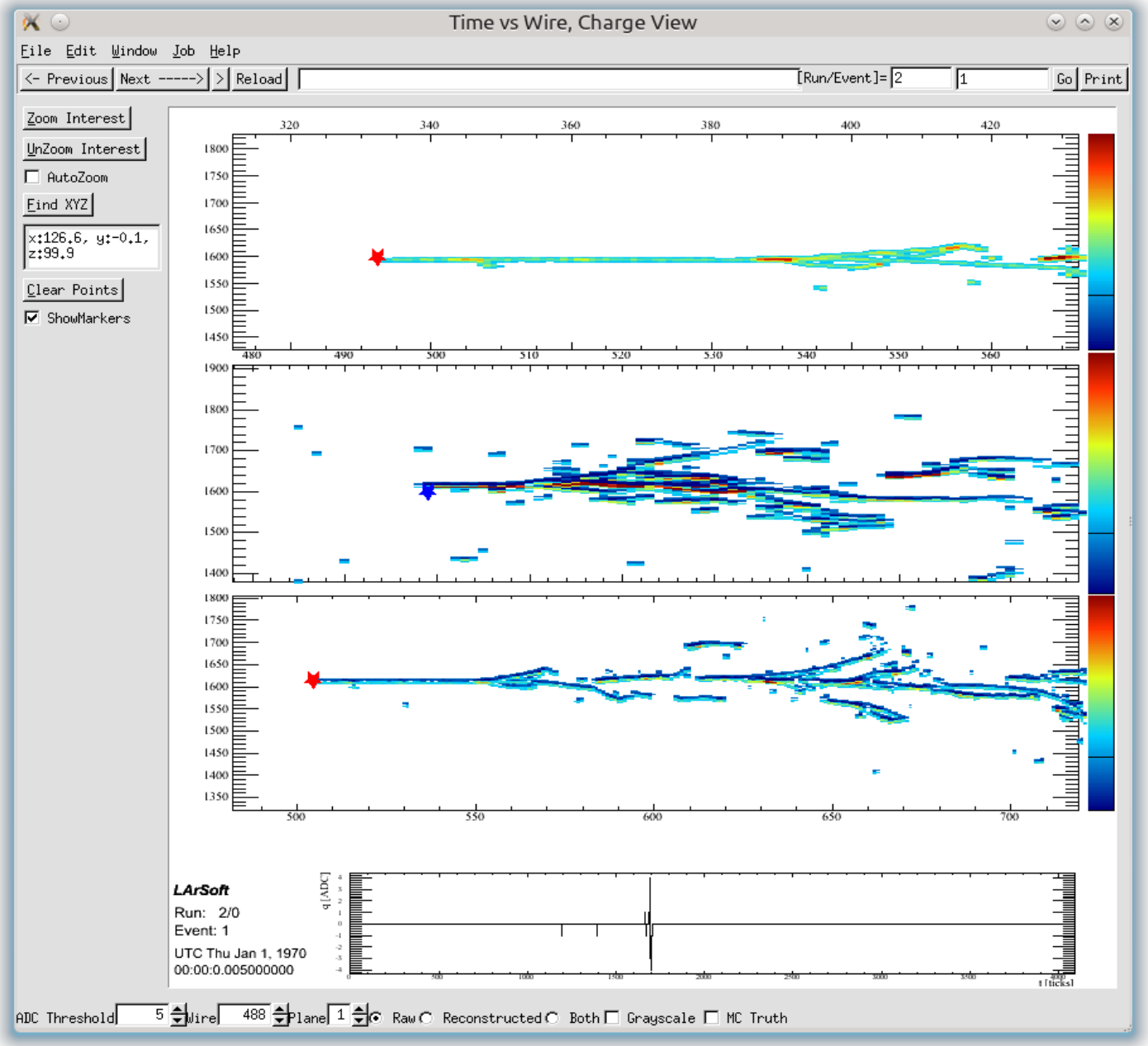


HowTo:

- Shift+L Mouse Click on points in two planes: red markers appear
- Click on Find XYZ button: real world x,y,z coordinates appear in the text display and a blue projection marker appears in the third plane.
- Any combination of two planes is ok.
- If wires don't cross or time distance is too long an error message gets printed in the text window.



Zoom:


Time vs Wire, Charge View
File Edit Window Job Help
<- Previous
Next ----->
Reload
[Run/Event]=
Go
Print
Zoom Interest
UnZoom Interest
AutoZoom
Find XYZ
x:126.6, y:-0.1, z:99.9
Clear Points
ShowMarkers
LArSoft
Run: 2/0
Event: 1
UTC Thu Jan 1, 1970
00:00:0.005000000
q [ADC]
t [ticks]
ADC Threshold
Wire
Plane
Raw
Reconstructed
Both
Grayscale
MC Truth

Status:

- Checked in to SVN
- New options in evdservices.fcl:

```
standard_evdlayoutopt:

{
 ShowSideBar:      1           # toggle extra sidebar visibility option
 AutoZoomInterest: 1           # toggle the auto zoom to interesting
                                 region option
 PrintTotalCharge: 1           # Print out the sum of collected charge
 ShowEndPointSection:  1         # Show Sidebar section with EndPoint
                                   extrapolation
 ShowEndPointMarkers:  1         # toggle visbility of markers for
                                   EndPoint finding

}
```

- Turned off by default for ArgoNeut.

Geant4 endless loop debugging

When running jobs (usually e^-)

- Jobs sometimes seem to hang.
- When pouncing on them with gdb, the symptoms showed that the Geant4 stepsize was infinitesimally small ~10 e-14.
- It seems that a particle gets stuck at a voxel boundary (one of coordinates pos. is a multiplier of ~ 0.3 mm)
- This particle is usually a photon (but not always)
- This problem has apparently been solved in the newer versions of Geant?

What is happening

- Presumably the particle gets stuck at the edge of a Geant4 intrinsic voxel and does not calculate the step correctly.
- Since this happens most often for photons it is most often present in EM shower events.
- Worst case: up to 3-4 events in 20 jobs of 500 showers (ArgoNeut).
- This sometimes happens for particles where fparticle==0 and so do not enter SteppingAction(...) in ParticleActionList.cxx

Temporary Solution in BadIdeaAction.cxx

- The solution is thanks to Bill Seligman.
- Applied only if the step number is over 50k AND the step number is smaller ~10e-12 mm (usual values are 2-3k and ~0.1 mm)
- The particle gets a kick by (0.001,0.001,0.001) mm and a warning is printed out:

```
##### In endless loop. Kicking particle by (+0.001,+0.001,+0.001)--
fparticle: 0x1eaba7f0 PDG and encoding 22 gamma current step number:
50001 stepsize: 2.32731e-14 x,y,z  265.628 106.5 521.55
```

- Don't try this at home! The fix is temporary and should go away, once we migrate to a newer geant4 version (update: this happened yesterday, need to run some events and test that it does not happen anymore).
- I have run over a 100 jobs with over a third e showers. As of this writing the problem did not repeat.

SingleGen update
(with Eric and Brian)

The “Bug”

- When running the SingleGen single particle gun all of the parameters are defined as vectors. These are: `PDG,P0, SigmaP, X0, Y0, Z0, SigmaX, SigmaY SigmaZ, Theta0XZ Theta0YZ, SigmaThetaXZ, SigmaThetaYZ.`
- This lets you run either different particle configurations in a job or multiple particles in an event depending on:
  `ParticleSelectionMode: 0 # 0 = use full list, 1 = randomly select a single listed particle`
- The code assumed that all of the above vectors would be of the same size as fPDG and did not warn when they weren't (it also did not crash most of the time).

The Fix

- The Code now checks the sizes of all of the above vectors and a new variable called PadOutVectors.
- If PadOutVectors == false, all vectors must be the same size. i.e. PDG: [13,13,13] → all vectors must have three fields.
- If PadOutVectors == true, the other vectors will be padded out provided they are of size one. i.e. → all vectors must be either, the size of PDG or size = 1.

Example:

- Ok:
  - `PadOutVectors: true`
  - `PDG:  [ 13,11,-11]`
  - `P0:  [ 6.,5.5,6. ]`
  - `SigmaP:  [ 0. ]`
  - `X0:  [ 25.,50.,60. ]`
- Not ok:
  - `PadOutVectors: true`
  - `PDG:  [ 13,11,-11]`
  - `P0:  [ 6.,5.5 ]`
  - `SigmaP:  [ 0. ]`
  - `X0:  [ 25.,50.,60. ]`

DatabaseUtil update

The DatabaseUtil + LArProperties -quick updates

- Temperature and Electric field can now also be downloaded from the DB (although they do not change much). Again they are caught by LArProperties, so the user doesn't need to call DatabaseUtil explictly.
- Electric Field is now a vector Efield(int = 0) is the drift field as before. Efield (I) gives the Electric Field value in the consecutive plane gaps (Collection being last)
- I have realized that the way LArProperties picked up the values from DatabaseUtil resulted in it being after all of the modules BeginJobs. Trying to do it the right way unearthed a bug in ART, which was propagated into LarSoft yesterday. The fix to LArProperties + DatabaseUtil is coming.
- As of Monday we have the three databases for the three experiments.
- Wiki page is halfway done. Will try to finish after the upgrades above.
- Added a variable called ToughErrorTreatment to the databaseutil.fcl if set to true, you will get an exception when the database connection fails. Currently set to false.
- Other changes soon to come, are tricks to avoid too many connections at once using ideas from Nova.
- I'm still following on the DB cloning ideas - NOVA is going through this currently – apparently PostgreSQL 9 allows real time synching of Databases. Ok if the Database stays LarSoft related. If other stuff gets thrown into it, like slow control or other might need cloning of just parts of the DB. Again NOVA + CD are researching a solution using Ruby on Rails.