LArSoft - Bug #22431

Inconsistent location of wire plane reported by TPCGeo

04/22/2019 05:03 PM - Gianluca Petrillo

Status:	Feedback	**Start date:**	04/22/2019
Priority:	Normal	**Due date:**	
Assignee:	Gianluca Petrillo	**% Done:**	30%
Category:	Geometry	**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.50 hour
Occurs In:	v0_00_04	**Co-Assignees:**	
Experiment:	-		

Description

Let's see if I can break a record here.

Premise: GEANT4 deals with volumes, and the wire plane is in fact not a geometric plane, but rather a box volume with its depth, which contains all the volumes associates with the wires on the plane. On the detector, though, there is no such a box and the wire plane is defined by the wires (GEANT4 "tubes").

geo::TPCGeo::PlaneLocation(p) attempts to provide the location of the plane p of the TPC. Matter of fact, it returns the center of the geometry box describing the plane and containing the wires. In this sense, it is equivalent to the more modern geo::TPCGeo::Plane(p).GetBoxCenter().
Unfortunately, PlaneLocation() is used to calculate the drift distance, which only works under the assumption that it returns a coordinate on the wire plane.
Matter of another fact, that is not necessarily the case. In ArgoNeuT geometry, for example, the wires are on one side of the plane box, say at *x* 0 cm for the front plane, while the plane box goes from -0.4 to 0 cm. The code using this feature, which includes LArG4 and some reconstruction, considers ArgoNeuT wire planes 2 mm off. I haven't checked the geometry of the other detectors.

My recommendation is to just ditch geo::TPCGeo::PlaneLocation() and all what is related to it, and to replace it with a call to geo::PlaneGeo::GetCenter() where needed. In alternative, geo::TPCGeo::PlaneLocation() can be reimplemented to do that internally. My personal preference goes to the first option though. Note that geo::PlaneGeo is well aware of the distinction, and it provides two different methods, GetCenter() and GetBoxCenter(), for the two different quantities.

I discovered this while refreshing the code of larsim:source:larsim/LArG4/LArVoxelReadout.cxx (feature/gp_refreshLArVoxelReadout), when I noticed one of ArgoNeuT integration tests showing different results. DUNE appeared to like the changes better, so maybe their wires *are* in the middle of the plane box. I can't swear for other experiments, though.

For what it's worth, for the drift distance I recommend the use of geo::PlaneGeo::DistanceFromPlane().

This feature was introduced, together with its implicit defect, with larcorealg:505c223 (that's the record-breaker).

Associated revisions

Revision 4794ece9 - 05/06/2019 07:25 PM - Gianluca Petrillo

`geo::TPCGeo::PlaneLocations()` now refers to `geo::PlaneGeo::GetCenter()`.

This is the first part of the solution of issue #22431.

History

#1 - 04/29/2019 10:55 AM - Kyle Knoepfel

- Status changed from New to Under Discussion

#2 - 04/29/2019 10:58 AM - Kyle Knoepfel

- Status changed from Under Discussion to Feedback

This sounds like a reasonable suggestion. Can you prepare a feature branch and present the change at a future LArSoft coordination meeting?

#3 - 04/29/2019 11:14 AM - Gianluca Petrillo

So, what about a twofold solution:

1. fix geo::TPCGeo::PlaneLocation() to give a consistent result
2. deprecate it (in the sense of C++ [[deprecate("use `geo::PlaneGeo::DistanceFromPlane()` or `geo::PlaneGeo::GetCenter()` instead")]]); I can provide LArSoft feature branches, but I will not chase the uses of it in foreign experiment code.
3. present the change when it's ready to go in, including a update guide

#4 - 05/06/2019 10:35 AM - Kyle Knoepfel

- Assignee set to Gianluca Petrillo

- Status changed from Feedback to Assigned

Sounds good. Fire away.

#5 - 05/06/2019 07:29 PM - Gianluca Petrillo

- % Done changed from 0 to 30

First part of the solution pushed as larcorealg:4794ece9 in larcorealgdevelop.

#6 - 05/13/2019 11:16 AM - Kyle Knoepfel

- Status changed from Assigned to Feedback

Does the change pushed to develop affect physics results? Either way, this needs to be discussed. This change should have been in a feature branch, as requested.

#7 - 05/13/2019 04:14 PM - Gianluca Petrillo

It's a bug fix: it changes physics results bringing them from wronger to less wrong.
The policy used to be that bug fixes (as in point 1 above) go in directly, so I believed the feature branch request to be for the deprecation at points 2 and 3.

#8 - 05/14/2019 09:09 AM - Lynn Garren

We realize that this is a bug fix, but it changes physics results. Anything that changes physics results, even to make them better, needs to be managed.