

LArSoft - Bug #17123

GeometryCore::WireDslIntersect() gives wrong intersection points in TVector3 version

07/07/2017 11:16 PM - Christopher Backhouse

Status:	Closed	Start date:	07/07/2017
Priority:	Urgent	Due date:	
Assignee:	Gianluca Petrillo	% Done:	100%
Category:	Geometry	Estimated time:	0.00 hour
Target version:		Spent time:	12.00 hours
Occurs In:		Co-Assignees:	
Experiment:	DUNE		

Description

The output from the version returning WireIDIntersection seems right, but the documentation says it's deprecated (because it assumes drift is in X?). The output from the Point3D_t (ie TVector3) version differs substantially, and doesn't seem right.

eg I have an X-wire, a U-wire, and a V-wire. XU and XV intersect, and the intersection points are near each other, but the UV intersection is far away. Looks like a coordinate system issue to me. The documentation says "(global coordinates)", which I assume means the usual whole-detector system with z along the beam etc.

This is in the DUNE SP FD geometry.

Associated revisions

Revision a81893ae - 07/10/2017 06:55 PM - Gianluca Petrillo

Fix on one of geo::GeometryCore::WireDslIntersect() methods.

This should solve issue #17123.

A new implementation is coded, after the suggestion in the ticket. The unit test has been extended to check the reverse intersection (that is, both (w1,w2) and (w2,w1)). Also a test of consistency with the legacy implementation has been added when drift direction is +/-x. There is still a risk that the unit test is too self-righteous and fails to detect errors.

History

#1 - 07/07/2017 11:47 PM - Christopher Backhouse

A simpler demonstration: the result depends on the order of the two WireID arguments.

#2 - 07/10/2017 09:26 AM - Gianluca Petrillo

Which geometry are you running on?

[edit] I had missed the information about ProtoDUNE Single Phase. Still, could you specify where you pick the geometry configuration from?

#3 - 07/10/2017 10:45 AM - Lynn Garren

- Status changed from New to Assigned

- Assignee set to Gianluca Petrillo

#4 - 07/10/2017 02:51 PM - Christopher Backhouse

DUNE SP FD. I have @table::dunefd_services in my fcl

Seems like a simpler implementation of the wire intersection function would be to just compute the point of closest approach between the two line segments and return that (assuming it's within each line). No changes of coordinates necessary.

#5 - 07/10/2017 03:57 PM - Gianluca Petrillo

- Category set to Geometry

- Priority changed from Normal to Urgent

Err... I am very embarrassed here.

There is a problem (which I haven't started studying yet).

And the unit test that was supposed to test this function was not run for an idiotic bug (for (unsigned int iD = -stepsD; iD <= stepsD; ...)... iD should not have been unsigned).

#6 - 07/11/2017 09:43 AM - Gianluca Petrillo

Following Christopher's suggestion, I have reimplemented the method to use native 3D geometry. The inline comments in the implementation describe the math used.

No change in interface nor in protocol have been made. One day we'll stop using loathed TVector3 and this method will even be kind-of-fast.

The fix is pushed in larcorealgp develop branch (because of issue [#17099](#)) and a few feature/gp_Issue17099 branches disabling some part of the geometry unit test.

Still to do:

- assess the impact of the bug, and announce the fix
- fix the unit test

Impact of the bug

The buggy method is relatively new, introduced a few months ago. I need to see where in the code base it is already used, to warn about possible wrong results.

Unit testing

Beside the bug mentioned in a previous note, the unit test would still succeed. That was likely due to a non-positive dependence of both the tested algorithm *and of the expected result* from the same buggy code.

This would make the test self-consistent even in case of bugs.

I have added a part to the test, to check that the result is symmetric with swapping the order of the wires.

In the current state, ArgoNeuT, LArIAT and ProtoDUNE single phase geometry tests still fail. For the former two, this might be due to their buggy geometry with uneven wire spacing. For the latter, it is still not clear: it is the only failing test.

I will need to verify the algorithm for the test expected result, which may present an ambiguity (a crossing of two wires defines two pairs of regions, opposite to the crossing point; it may be that the prediction is using the wrong one of the two pairs).

#7 - 07/11/2017 09:48 AM - Gianluca Petrillo

- % Done changed from 0 to 80

#8 - 07/11/2017 10:03 AM - Gianluca Petrillo

- % Done changed from 80 to 90

Not surprisingly, I can't find any place where the newer `geo::GeometryCore::WireIDsIntersect()` is used... (about which I am a bit disappointed).

My survey included: 2 LArSoftObj repositories, 12 LArSoft repositories, larcorealgp and 5 experiment repositories (ArgoNeuT, DUNE, MicroBooNE, SBND and LArIAT; no Icarus).

#9 - 07/11/2017 02:32 PM - Christopher Backhouse

Looks good to me. Thanks!

#10 - 07/11/2017 05:50 PM - Gianluca Petrillo

- Status changed from Assigned to Resolved

- % Done changed from 90 to 100

I have reworked the formula for the expected distance from intersection of a point, avoiding the use of angles. It should be more robust now, and it passes also for ProtoDUNE Single Phase geometry test (which is not included in the official dune, and still fails the third plane slope test).

LArIAT and ArgoNeuT tests would fail (they are disabled) because of a different bug in their geometry.

No residual problem known.

The new version of the test is pushed in larcorealgp develop branch, with no particular hurry of going in.

#11 - 07/26/2017 03:56 PM - Gianluca Petrillo

- Status changed from Resolved to Closed