Status of Geometry service changes to accommodate pixel readouts

Kyle J. Knoepfel
LArSoft coordination meeting
20 September 2022

Motivation

- **LArSoft intends to support pixel geometries**
  - To do this, some adjustments to the Geometry service/system are required.
    - *Will likely be separating readout-specific concepts from those of geometry.*
  - A few of us are meeting weekly to determine how best to proceed.
  - While analyzing geometry code, it became apparent that much of the interface serves as “legacy” code to support older coding patterns

Motivation

- **LArSoft intends to support pixel geometries**
  - To do this, some adjustments to the Geometry service/system are required.
    - *Will likely be separating readout-specific concepts from those of geometry.*
  - A few of us are meeting weekly to determine how best to proceed.
  - While analyzing geometry code, it became apparent that much of the interface serves as “legacy” code to support older coding patterns

- **Maintenance issues**
  - We will need to rearrange some parts of the code to support pixel geometries—***it’s less work to adjust only the code that’s required***.
  - I suggest we address the “deprecated” code first.

Deprecated geometry functionality

- There are many functions listed with the `@deprecated` Doxygen tag (e.g.):

```
     std::size_t AuxDet::FindSensitiveVolume(geo::Point_t const& point) const;
-    /// @deprecated Use the version with `geo::Point_t` argument instead
-    std::size_t AuxDet::FindSensitiveVolume(double const worldLoc[3]) const;
```

```
-    /**
-     * @brief Returns the auxiliary detector at specified location
-     * @param worldLoc 3D coordinates of the point (world reference frame)
      ...
-     * @deprecated Use the version with `geo::Point_t`.
-     */
-     const AuxDetSensitiveGeo&
-     GeometryCore::PositionToAuxDetSensitive(double const worldLoc[3],
-                                             size_t     & ad,
-                                             size_t     & sv,
-                                             double tolerance = 0) const;
-
```

Proposal: *Remove deprecated geometry interface*

- While it may not be possible to easily remove all uses of deprecated geometry functions, classes, and types, I would like to remove as much as possible.
- Reason for not removing this interface in the past: not enough time.

Proposal: *Remove deprecated geometry interface*

- While it may not be possible to easily remove all uses of deprecated geometry functions, classes, and types, I would like to remove as much as possible.
- Reason for not removing this interface in the past: not enough time.

- **Status**

  I have LArSoft feature branches ready, where most of the deprecated functionality has been removed and is no longer used.

  I have not yet created feature branches for the experiments—I will do so assuming approval from the LCM.

  I prefer to create PRs after LArSoft has adopted `clang-format`, but I can create them now if necessary or desirable.