

TPC simulation: position prior to the SCE

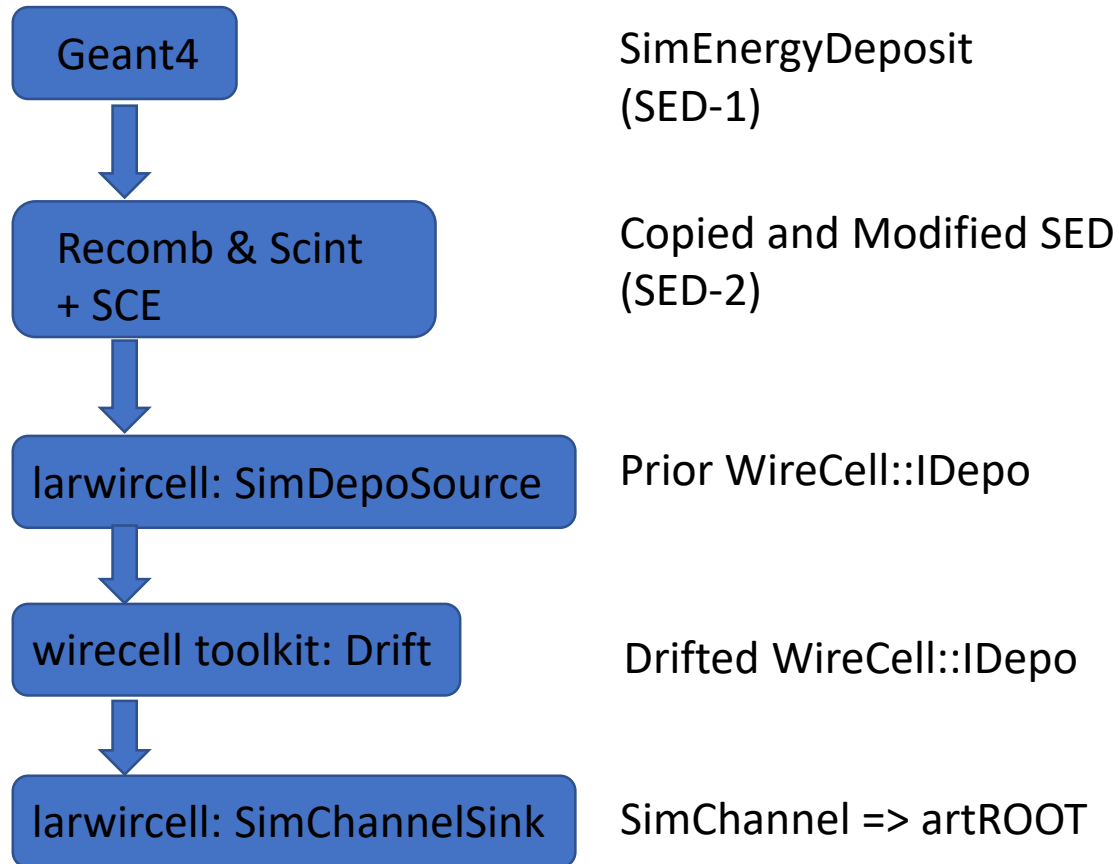
Wenqiang Gu

Brookhaven National Laboratory

Problem description

- The current protoDUNE-SP simulation chain saves the truth information in SimChannel
 - ▶ Larwirecell::SimChannelSink saves energy depo's original XYZ information in SimChannel::TrackIDE
- The issue is that the space charge effect (SCE) is simulated before the Wire-Cell TPC simulation, as a result, Wire-Cell doesn't know the original XYZ before SCE
 - ▶ Although a backward SCE correction can be applied to the saved XYZ, it's not the original position people would like to get from a MC "backtracker"

Simulation chain



- In the WireCell toolkit (WCT), the motherhood of a depo is recorded in a chain
- Two possible solutions:
 - ▶ We can move the SCE simulation into the WireCell, the original “IDepo” provides the initial XYZ
 - ▶ (Recommended) read in two sets of energy depos from artROOT, one is with SCE, the other is without SCE

Implementation in larwirecell::SimDepoSource

```
if (assn_sedv.size() == 0) {
    WireCell::IDepo::pointer depo
        = std::make_shared<WireCell::SimpleDepo>(wt, wpt, wq, nullptr, 0.0, 0.0, wid, pdg, we);
    m_depos.push_back(depo);
}
else {
    auto const& sed1 = assn_sedv.at(ind);
    auto pt1 = sed1.MidPoint();
    const WireCell::Point wpt1(pt1.x()*units::cm, pt1.y()*units::cm, pt1.z()*units::cm);
    double wt1 = sed1.Time()*units::ns;
    double wq1 = (*m_adapter)(sed1);
    int wid1 = sed1.TrackID();
    int pdg1 = sed1.PdgCode();
    double we1 = sed1.Energy()*units::MeV;

    WireCell::IDepo::pointer assn_depo
        = std::make_shared<WireCell::SimpleDepo>(wt1, wpt1, wq1, nullptr, 0.0, 0.0, wid1, pdg1, we1);

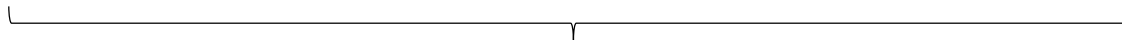
    WireCell::IDepo::pointer depo
        = std::make_shared<WireCell::SimpleDepo>(wt, wpt, wq, assn_depo, 0.0, 0.0, wid, pdg, we);
    m_depos.push_back(depo);
}
```

Set a shared_ptr as “mother” of this depo

Implementation in larwirecell::SimChannelSink

```
WireCell::IDepo::pointer orig = depo_chain(depo).back(); // first depo in the chain
xyz[0] = orig->pos().x()/units::cm;
xyz[1] = orig->pos().y()/units::cm;
xyz[2] = orig->pos().z()/units::cm;
```

- A predefined function “depo_chain()” can loop over the chain of the related depos and find the first one
 - ▶ (geant4)=> first depo => (SCE) => second depo => (WCT) => drifted depo



larsoft

Validation

<https://github.com/LArSoft/larwirecell/pull/7>

<https://github.com/LArSoft/larsim/pull/39>

G4Stage2....	TriggerResults	art::TriggerResults.....
G4Stage2....	IonAndScint...	priorSCE.....	std::vector<sim::SimEnergyDeposit>.....	15482
G4Stage2....	rns.....	std::vector<art::RNGsnapshot>.....3
G4Stage2....	IonAndScint...	std::vector<sim::SimEnergyDeposit>.....	15482
G4Stage2....	PDFastSim.....	pvs.....	std::vector<sim::OpDetBacktrackerRecord>.....	...90

Two energy depo from larsoft as input, they should have the same length and same order


// larwirecell::SimDepoSrouce prints out the XYZ of a depo

Before SCE: (x,y,z) = (-128.846 , 366.834, 386.813)

After SCE: (x,y,z) = (-121.282 , 364.624 , 386.149)

// larwirecell::SimChannelSink prints out the XYZ of a depo

Depo_chain(depo).back(): (x,y,z) = (- 128.846, 366.834, 386.813)  Before SCE (first depo)

Depo->prior(): (x,y,z) = (-121.282 , 364.624, 386.149)  After SCE (second depo)

First depo => (SCE) => Second depo => (WCT) => Drifted depo