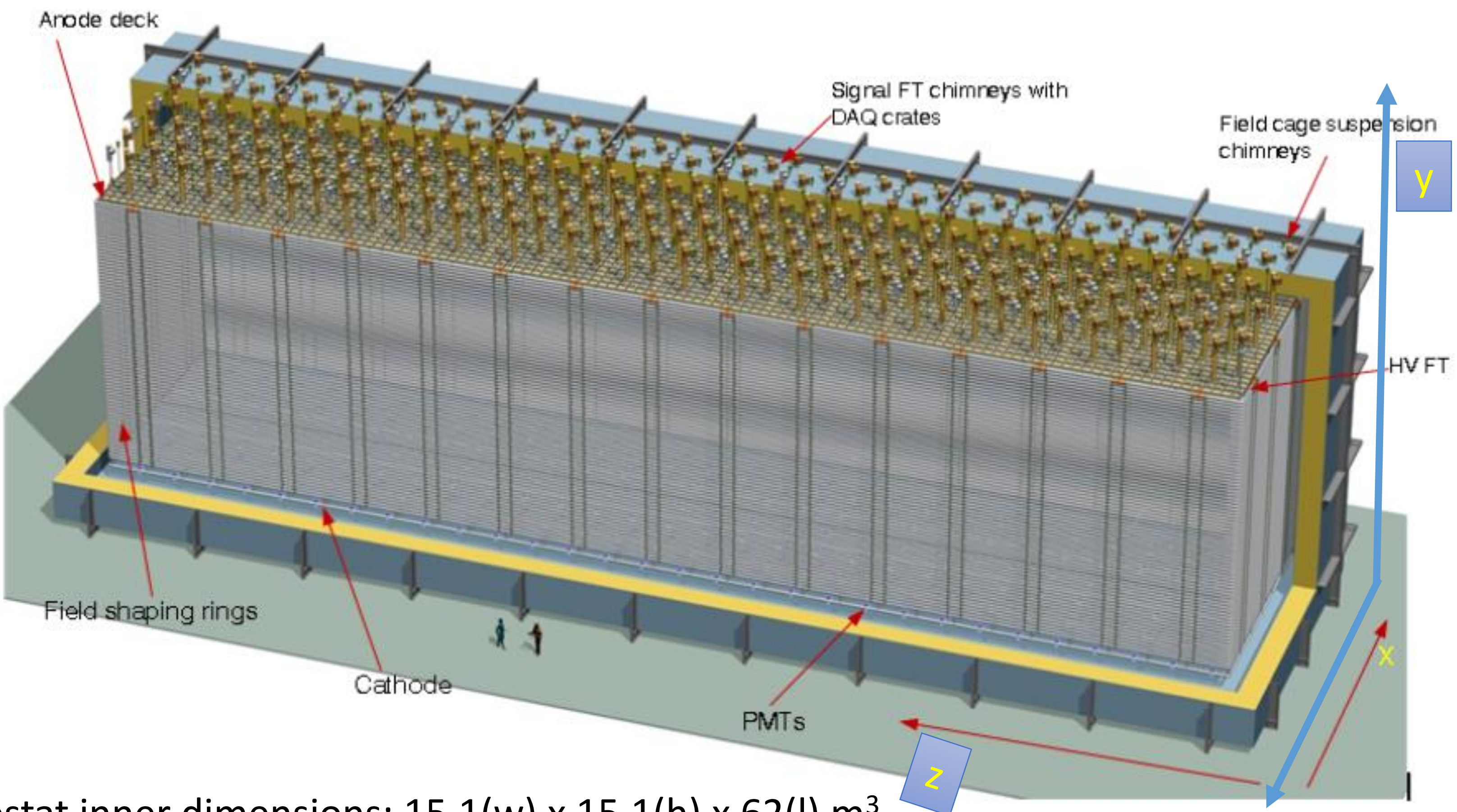
Dual-phase TPC simulation in LArSoft for DUNE FD

Vyacheslav Galymov

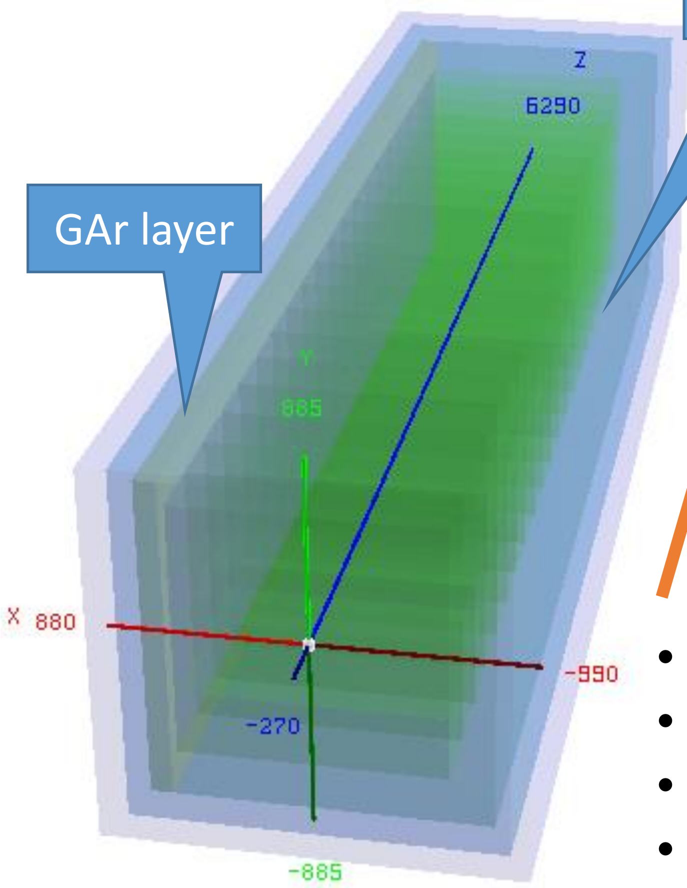
LArSoft Meeting 17.11.2015

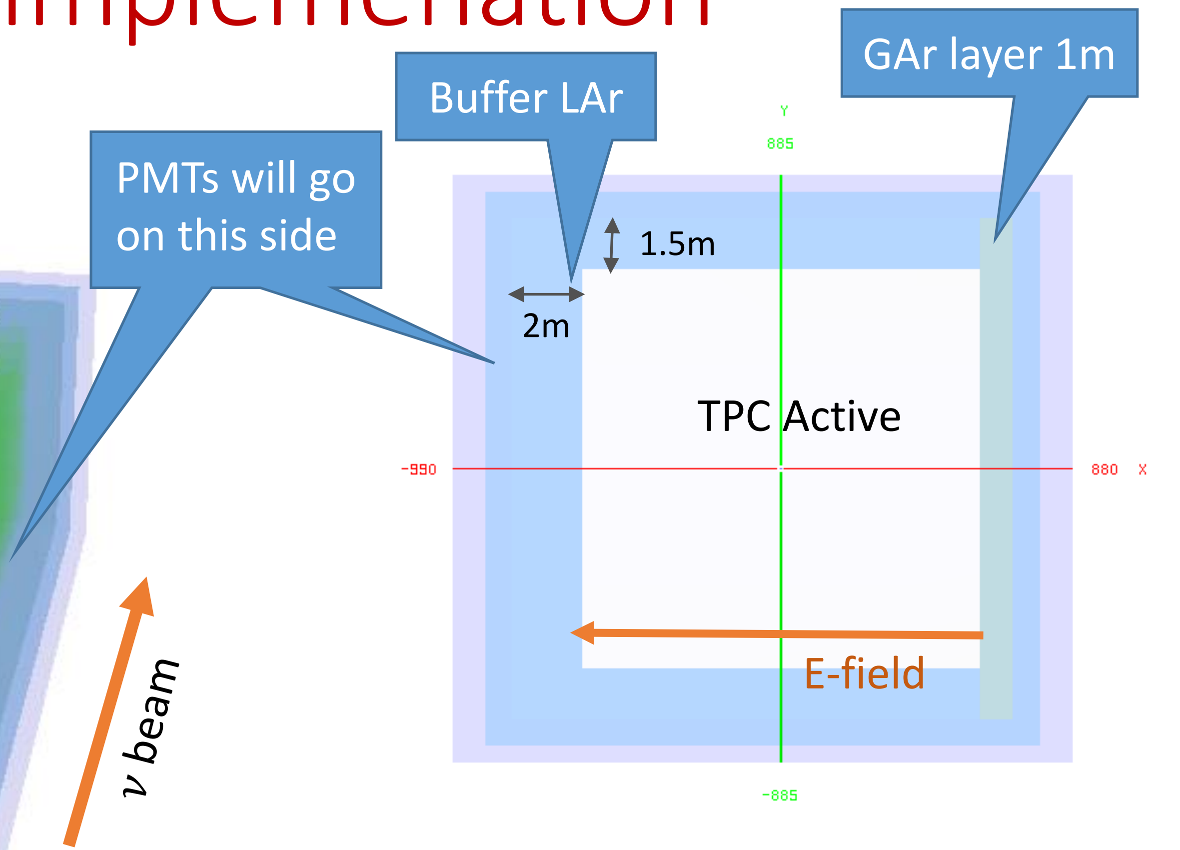
DUNE dual-phase TPC



- Cryostat inner dimensions: 15.1(w) x 15.1(h) x 62(l) m^3
- Active TPC dimensions: 12(w) x 12(drift) x 60 (l) m^3
- CRP unit = CRM (Charge Readout Module): 3 x 3 m^2 with 960 channels per view and pitch of 3.125 mm, border size for each CRM < 5 mm

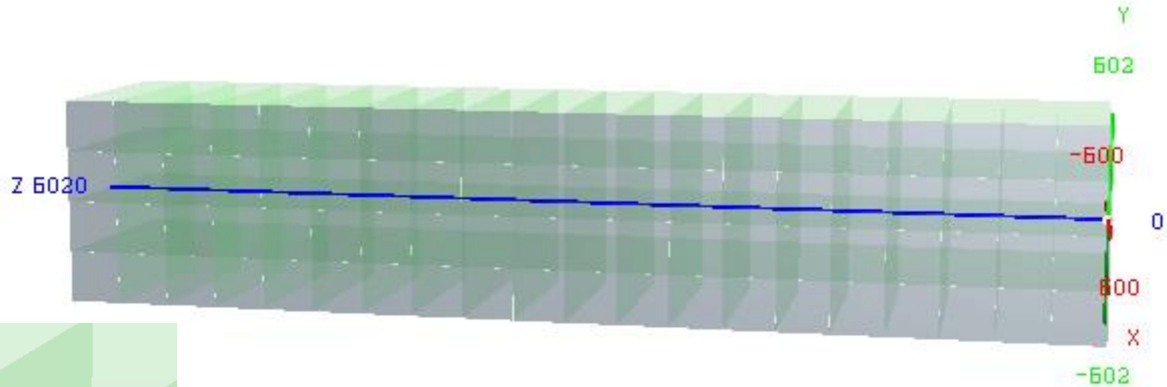
Geometry Implemenation

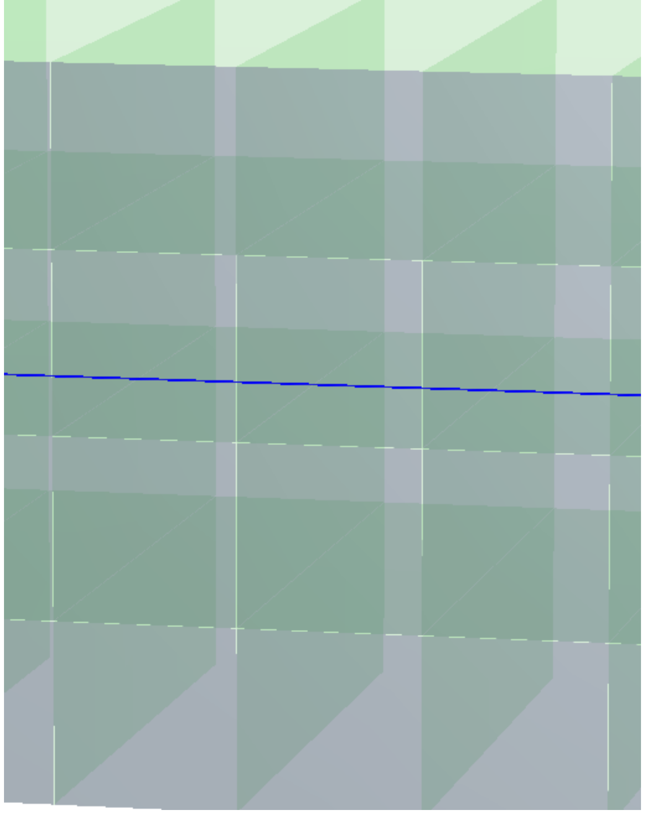




- Inner cryostat dimensions: 15.1 x 15.1 x 62 m^3
- Active volume dimensions: 12.0 x 12.04 x 60.2 m^3
- GAr layer is on +x side (e^- drift to +x)
- Origin of coordinate system is the center of the TPC active volume at the upstream end

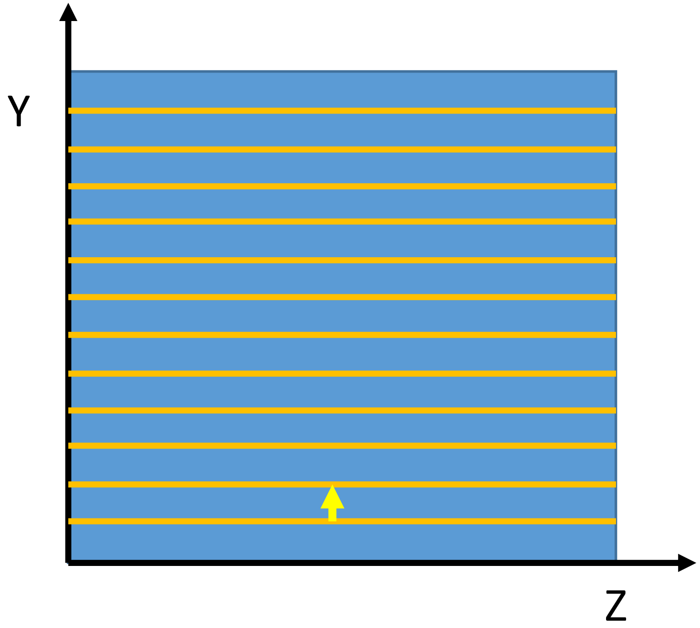
Charge readout modules (CRM)

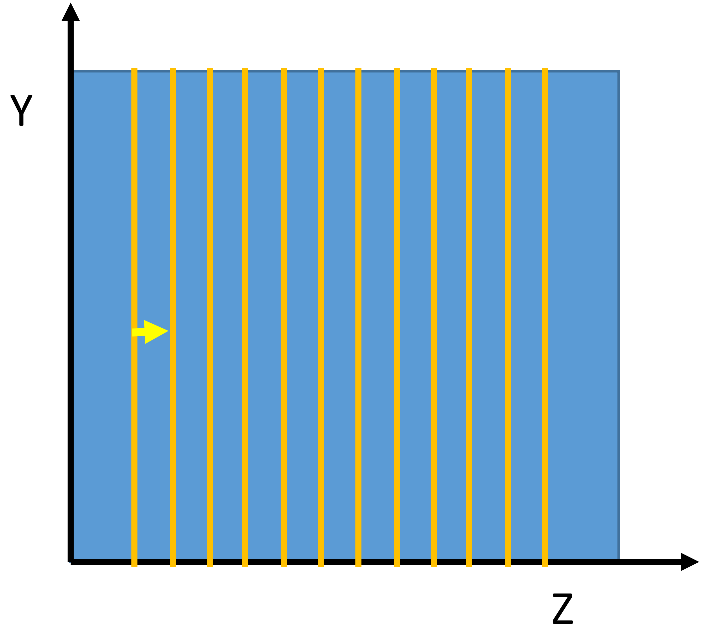




- Active volume consists of copy of identical rectangular prisms to be read out by 960 ch in “Y” and 960 ch in Z collection views
- The dimensions of each module are 3x3 m^2
- There is a dead space between each module of 1 cm associated with a border size is 0.5 cm

Wire planes





1st Vplane (assigned geo::kU in larsoft)

2nd Zplane (assigned geo::kV in larsoft)

Including geometry

geometry_dune.fcl

```
dunedphase10kt_geo:
{
 Name:       "dunedphase10kt_v2"

 GDML:       "dunedphase10kt_v2.gdml"
 ROOT:       "dunedphase10kt_v2.gdml"

 SurfaceY: 147828                   # Underground option. 4850 feet to cm. from DocDb-3833

 DisableWiresInG4:   true
}
```

```
dune_geometry_helper:
{
 service_provider : DUNEGeometryHelper
}
```

Keep the same service

DUNEGeometryHelper

```
//
// DUNE 10kt
//
else if (( detectorName.find("dune10kt") != std::string::npos )
  || ( detectorName.find("lbne10kt") != std::string::npos ))
{
  fChannelMap = std::make_shared<geo::ChannelMapAPAAlg>(sortingParameters);
}
//
// DUNE 10kt dual-phase
//
else if ( detectorName.find("dunedphase10kt") != std::string::npos )
  {
    fChannelMap = std::make_shared<geo::ChanneMapCRMAlg>(sortingParameters);
  }
```

Channel mapping algorithm

Note: in this scheme should not use geometry names like “dune10ktdphase”

Channel map & geo sorter

ChannelMapCRMAlg Essentially a copy of larsoft *ChannelMapStandardAlg* with some minor modification & the geo sorter **GeoObjectSorterCRM**

1. Ensure that only 2 planes are detected in ChannelMapCRMAlg

```
for(unsigned int PlaneCount = 0; PlaneCount != PlanesThisTPC; ++PlaneCount){

  if(PlaneCount >= 2) //should only find two views for dual-phase
    throw cet::exception("Geometry") <<"CANNOT HAVE more than two readout planes for dual-phase"
                                     << "\n";
```

2. Sort wires GeoObjectSorterCRM

```
//------------------------------------------------------------------------
bool sortWireCRM(WireGeo* w1, WireGeo* w2){
  double xyz1[3] = {0.};
  double xyz2[3] = {0.};

  w1->GetCenter(xyz1); w2->GetCenter(xyz2);

  // for dual-phase we have to planes with wires perpendicular to each other
  // sort wires in the increasing coordinate order

  if( fabs(xyz1[2]-xyz2[2]) < 1.0E-6 ) // for wires along y-axis
    {
      if(xyz1[1] < xyz2[1]) return true;
      else return false;
    }
  else if( fabs(xyz1[1]-xyz2[1]) < 1.0E-6 ) // for wires along z-axis
    {
      if(xyz1[2] < xyz2[2]) return true;
      else return false;
    }
  else //don't know what to do
    throw cet::exception("TPCGeo") << "Uknown sorting situation for the wires in a plane\n";


  return false;
}
```

Y: down → up

Z: upstream → downstream

CRMs (TPCs) sorting

The sorting order in y-z is the same as for DUNE single-phase

```
//----------------------------------------------------------
// Define sort order for tpcs in dual-phase configuration
static bool sortTPCCRM(const TPCGeo* t1, const TPCGeo* t2)
{
  double xyz1[3] = {0.};
  double xyz2[3] = {0.};
  double local[3] = {0.};
  t1->LocalToWorld(local, xyz1);
  t2->LocalToWorld(local, xyz2);

  // First sort all TPCs into same-z groups
  if(xyz1[2]<xyz2[2]) return true;

  // Within a same-z group, sort TPCs into same-y groups
  if(xyz1[2] == xyz2[2] && xyz1[1] < xyz2[1]) return true;

  return false;
}
```

Checking geometry

- Basic check of the dual-phase GDML file interpretation in larsoft using "CheckGeometry_module" from Tingjun
  - Simply print information from geo::Geometry to stdout

```
Total number of TPC 80
TPC 0 has found 2 planes
Drift direction : geo::kPosX
Drift distance  : 1200

   View type geo::kU
   View is geo::kInduction
   Number of wires : 960
   Wire pitch      : 0.3125
   Theta Z         : 0

   View type geo::kV
   View is geo::kCollection
   Number of wires : 960
   Wire pitch      : 0.3125
   Theta Z         : 1.5708
```

...

```
TPC 79 has found 2 planes
Drift direction : geo::kPosX
Drift distance  : 1200
   View type geo::kU
   View is geo::kInduction
   Number of wires : 960
   Wire pitch      : 0.3125
   Theta Z         : 0

   View type geo::kV
   View is geo::kCollection
   Number of wires : 960
   Wire pitch      : 0.3125
   Theta Z         : 1.5708

Total number of channel wires = 153600
```

- ✓ Total number of CRM (4 x 20 = 80)
- ✓ Drift direction & length
- ✓ Number of readout planes per TPC
- ✓ Number of wires & pitch
- ✓ Angle of the wire with respect to Z axis in each plane
- ✓ Total number of wires/channels (2 x 960 x 80 = 153600)

The first two plane are labelled U and V. The last plane is tagged as collection while other(s) are induction. Set within *geo::TPCGeo::SortSubVolumes()* function

For dual-phase will need to ignore these signal type assignments at the level of SimWire

DUNE services

```
dunefd_services:
{
 ExptGeoHelperInterface:       @local::dune_geometry_helper
 Geometry:                     @local::dune10kt_geo
 TimeService:                  @local::dunefd_timeservice
 DetectorProperties:           @local::dunefd_detproperties
 LArProperties:                @local::dunefd_properties
 LArFFT:                       @local::dunefd_larfft
 DatabaseUtil:                 @local::dunefd_database
 BackTracker:                  @local::dunefd_backtracker
 SeedService:                  @local::dune_seedservice
 SignalShapingServiceDUNE10kt: @local::dunefd_signalshapingservice
}
```

```
#
# for dual-phase implementation
#
dunefddphase_services:                           @local::dunefd_services
dunefddphase_services.Geometry:                  @local::dunedphase10kt_geo
dunefddphase_services.TimeService:               @local::dunefddphase_timeservice
dunefddphase_services.DetectorProperties:        @local::dunefddphase_detproperties
dunefddphase_services.LArProperties:             @local::dunefddphase_properties
```

Time & detector properties

```
dunefddphase_timeservice: @local::standard_timeservice

# dunefddphase_timeservice.TrigModuleName:      ""
dunefddphase_timeservice.InheritClockConfig:   false
dunefddphase_timeservice.G4RefTime:             0.  # G4 time [us] where electronics clock counting start
dunefddphase_timeservice.TriggerOffsetTPC:      0.  # Time [us] for TPC readout start w.r.t. trigger time
dunefddphase_timeservice.FramePeriod:        8000.  # Frame period [us]
dunefddphase_timeservice.ClockSpeedTPC:         2.5 # TPC clock speed in MHz
dunefddphase_timeservice.ClockSpeedOptical:    65.  # Optical clock speed in MHz
dunefddphase_timeservice.ClockSpeedTrigger:    16.  # Trigger clock speed in MHz
dunefddphase_timeservice.DefaultTrigTime:       0.  # Default trigger time [us].
dunefddphase_timeservice.DefaultBeamTime:       0.  # Default beam time [us].
```

```
dunefd_detproperties:                     @local::standard_detproperties
# dunefd_detproperties.SamplingRate:        500.        #in ns
dunefd_detproperties.ElectronsToADC:      6.8906513e-3 # 1fC = 43.008 ADC counts for DUNE fd
dunefd_detproperties.NumberTimeSamples:   4492         # drift length/drift velocity*sampling rate = (359.4 cm)/(0.16 cm/us)*(2 MHz)
dunefd_detproperties.ReadOutWindowSize:   4492         # drift length/drift velocity*sampling rate = (359.4 cm)/(0.16 cm/us)*(2 MHz)
dunefd_detproperties.TimeOffsetU:         0.
dunefd_detproperties.TimeOffsetV:         0.
dunefd_detproperties.TimeOffsetZ:         0.

dunefddphase_detproperties:                     @local::standard_detpropertie
# dunefddphase_detproperties.SamplingRate:      400.        # in n
dunefddphase_detproperties.ElectronsToADC:      5.1267e-04  #
dunefddphase_detproperties.NumberTimeSamples:   20000       # drift length/drift velocity*sampling rate
dunefddphase_detproperties.ReadOutWindowSize:   20000       # drift length/drift velocity*sampling rate
dunefddphase_detproperties.TimeOffsetU:         0.
dunefddphase_detproperties.TimeOffsetV:         0.
dunefddphase_detproperties.TimeOffsetZ:         0.
```

Calculated assuming mip at 100 ADC with 31.25 fC/strip @ Gain = 10 per view : 1fC to ADC = 3.2

Dual-phase gain factor will be put in detsimmodules_dune.fcl

Question of Efield in view planes

LAr properties

```
dunefd_properties:                      @local::standard_properties
dunefd_properties.Temperature:          87
dunefd_properties.Electronlifetime:     3.0e3
dunefd_properties.Efield:               [0.5,0.666,0.8]  #(predicted for microBooNE)

dunefddphase_properties:                      @local::standard_properties
dunefddphase_properties.Temperature:          87
dunefddphase_properties.Electronlifetime:     3.0e3             # us
dunefddphase_properties.Efield:               [0.5,10.0,0.0]    #?
```

Drift UV VZ

How do the last two field values affect electron propagation in view planes?

Current status & future plans

- Dual-phase geometry description appears to be ok

- Working on completing SimWire for dual-phase detector
- Once done try to generate through-going muons to check
  - Signal normalization
  - Wire / TPC assignment

- From there move on to including light detectors (PMTs)