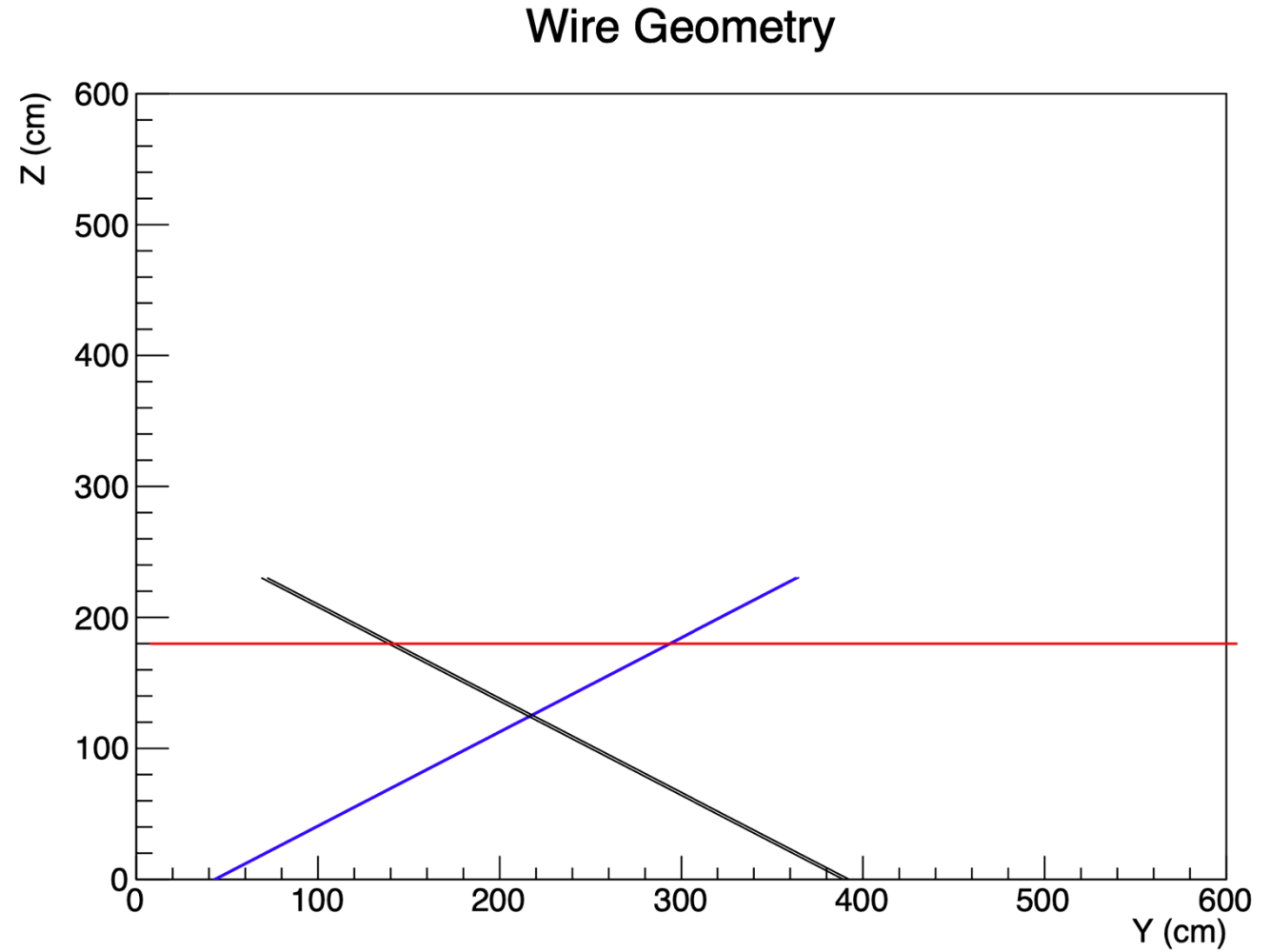


Update for imaging validation

Xuyang Ning

Recap

I am still debugging this...



Problems found

wire-cell-cfg/pgrapher/experiment/pdhd/img.jsonnet

```
// A functio that sets up slicing for an APA.
slicing:: function(anode, aname, span=4, active_planes=[0, 1, 2], masked_planes=[], dummy_planes=[]) {
  ret: g.pnode({
    type: 'MaskSlices',
    name: 'slicing-' + aname,
    data: {
      tick_span: span,
      wiener_tag: 'wiener%d' % anode.data.ident,
      summary_tag: 'wiener%d' % anode.data.ident,
      charge_tag: 'gauss%d' % anode.data.ident,
      error_tag: 'gauss_error%d' % anode.data.ident,
      anode: wc.tn(anode),
      min_tbin: 0,
      max_tbin: 8500,
      active_planes: active_planes,
      masked_planes: masked_planes,
      dummy_planes: dummy_planes,
      // nthreshold: [1e-6, 1e-6, 1e-6],
      //nthreshold: [3.6, 3.6, 3.6],
      nthreshold: [2, 2, 2],
    },
  }, nin=1, nout=1, uses=[anode]),
}.ret,
```

- Conflict between `default_config` and `config` in MaskSlice.cxx.
- WireCell::Configuration Img::MaskSliceBase::default_configuration() const
- void Img::MaskSliceBase::configure(const WireCell::Configuration& cfg)

wire-cell-toolkit/img/src/MaskSlice.cxx

```
void Img::MaskSliceBase::configure(const WireCell::Configuration& cfg)
{
  m_anode = Factory::find_tn<IAnodePlane>(cfg["anode"].asString()); // throws
  m_tick_span = get(cfg, "tick_span", m_tick_span);
  m_wiener_tag = get<std::string>(cfg, "wiener_tag", m_wiener_tag);
  std::cout<<"wiener_tag: "<<m_wiener_tag<<std::endl;
  std::cout<<"summary_tag: "<<m_summary_tag<<std::endl;
  m_summary_tag = get<std::string>(cfg, "summary_tag", m_wiener_tag); // if not given, use wiener_tag
  std::cout<<"summary_tag: "<<m_summary_tag<<std::endl;
  std::cout<<"wiener_tag: "<<m_wiener_tag<<std::endl;
  m_charge_tag = get<std::string>(cfg, "charge_tag", m_charge_tag);
  m_error_tag = get<std::string>(cfg, "error_tag", m_error_tag);
  m_dummy_charge = get<double>(cfg, "dummy_charge", m_dummy_charge);
  m_dummy_error = get<double>(cfg, "dummy_error", m_dummy_error);
  log->debug("double dummy_error = {}", m_dummy_error);
  m_masked_charge = get<double>(cfg, "masked_charge", m_masked_charge);
  m_masked_error = get<double>(cfg, "masked_error", m_masked_error);
  m_min_tbin = get<int>(cfg, "min_tbin", m_min_tbin);
  m_max_tbin = get<int>(cfg, "max_tbin", m_max_tbin);
  if (cfg.isMember("active_planes")) {
    m_active_planes.clear();
    for (auto id : cfg["active_planes"]) {
      m_active_planes.push_back(id.asInt());
    }
  }
}
```

Thank for Wenqiang and Haiwang

Wire geometry

One Blob:

Blob 0: Time [368,372], U [1056:1057], V [449:450], W [453:454], Charge: [597,2266,3452,850]

Extract map from [GridTiling.cxx](#)

```
for (const auto& ch_id : channel_ids) {
    auto channel = m_anode->channel(ch_id);
    if (!channel) continue;

    // outfile << "Channel ID: " << ch_id << std::endl;
    std::cout << "Channel ID: " << ch_id << std::endl;

    const auto& wires = channel->wires();
    // outfile << " Wire count: " << wires.size() << std::endl;
    std::cout << " Wire count: " << wires.size() << std::endl;

    for (const auto& wire : wires) {
        int wire_id = wire->ident();
        int plane_id = wire->planeid().ident();
        int segment = wire->segment();

        // Get wire geometry
        const auto& ray = wire->ray();
        const auto& p1 = ray.first;
        const auto& p2 = ray.second;

        outfile << ch_id << " " << faceid << " " << plane_id << " " << wire_id << " "
            << p1.x() << " " << p1.y() << " " << p1.z() << " "
            << p2.x() << " " << p2.y() << " " << p2.z() << "\n";
        std::cout << ch_id << ", " << wire_id << ", "
            << plane_id << ", " << faceid << ", "
            << p1.x() << ", " << p1.y() << ", " << p1.z() << ", "
            << p2.x() << ", " << p2.y() << ", " << p2.z() << "\n";
    }
    outfile.close();
}
// exit(0);
```

Geometry file: [protodunehd-wires-larsoft-v6.txt](#)

#	channel	tpc	plane	wire	sx	sy	sz	ex	ey	ez
0	0	0	0	-362.772	606.67	229.781	-362.772	605.569	230.572	
0	1	0	400	-352.219	285.577	-0.1	-352.219	606.509	230.572	
0	0	0	800	-362.772	286.518	-0.1	-362.772	7.61	200.367	
1	0	0	1	-362.772	606.67	229.206	-362.772	604.769	230.573	
1	1	0	401	-352.219	284.777	-0.1	-352.219	605.709	230.572	
1	0	0	801	-362.772	285.718	-0.1	-362.772	7.61	199.792	
2	0	0	2	-362.772	606.67	228.631	-362.772	603.969	230.573	
2	1	0	402	-352.219	283.977	-0.1	-352.219	604.909	230.572	
2	0	0	802	-362.772	284.918	-0.1	-362.772	7.61	199.217	
3	0	0	3	-362.772	606.67	228.056	-362.772	603.169	230.572	
3	1	0	403	-352.219	283.177	-0.1	-352.219	604.109	230.572	
3	0	0	803	-362.772	284.118	-0.1	-362.772	7.61	198.642	
4	0	0	4	-362.772	606.67	227.481	-362.772	602.369	230.572	
4	1	0	404	-352.219	282.377	-0.1	-352.219	603.309	230.572	
4	0	0	804	-362.772	283.318	-0.1	-362.772	7.61	198.067	
5	0	0	5	-362.772	606.67	226.906	-362.772	601.569	230.573	
5	1	0	405	-352.219	281.577	-0.1	-352.219	602.509	230.572	
5	0	0	805	-362.772	282.518	-0.1	-362.772	7.61	197.492	
6	0	0	6	-362.772	606.67	226.331	-362.772	600.769	230.573	
6	1	0	406	-352.219	280.777	-0.1	-352.219	601.709	230.572	
6	0	0	806	-362.772	281.718	-0.1	-362.772	7.61	196.917	
7	0	0	7	-362.772	606.67	225.756	-362.772	599.969	230.572	
7	1	0	407	-352.219	279.977	-0.1	-352.219	600.909	230.572	
7	0	0	807	-362.772	280.918	-0.1	-362.772	7.61	196.342	
8	0	0	8	-362.772	606.67	225.181	-362.772	599.169	230.572	
8	1	0	408	-352.219	279.177	-0.1	-352.219	600.109	230.572	
8	0	0	808	-362.772	280.118	-0.1	-362.772	7.61	195.767	
9	0	0	9	-362.772	606.67	224.606	-362.772	598.369	230.573	
9	1	0	409	-352.219	278.377	-0.1	-352.219	599.309	230.572	
9	0	0	809	-362.772	279.318	-0.1	-362.772	7.61	195.192	

Blob 0: Time [368,372], U [1056:1057], V [449:450], W [453:454], Charge: [597,2266,3452,850]

Geometry file:

[protodunehd-wires-larsoft-v6.txt](#)

u Wire position: (-352.219, 7.61, 177.303) -> (-352.219, 81.723, 230.572)
u Wire position: (-352.219, 7.61, 177.878) -> (-352.219, 80.923, 230.572)
v Wire position: (-352.711, 566.449, 0.235) -> (-352.711, 246.449, 230.237)
v Wire position: (-352.711, 565.649, 0.235) -> (-352.711, 245.649, 230.237)
w Wire position: (-353.202, 7.61, 217.538) -> (-353.202, 606, 217.538)
w Wire position: (-353.202, 7.61, 218.017) -> (-353.202, 606, 218.017)

Extract map from [GridTiling.cxx](#)

u Wire position: (-352.219, 7.61014, 177.303) -> (-352.219, 81.723, 230.572)
u Wire position: (-352.219, 7.61015, 177.878) -> (-352.219, 80.923, 230.572)
v Wire position: (-352.711, 246.449, 230.237) -> (-352.711, 566.449, 0.234643)
v Wire position: (-352.711, 245.649, 230.237) -> (-352.711, 565.649, 0.234652)
w Wire position: (-353.202, 7.61, 217.538) -> (-353.202, 606, 217.538)
w Wire position: (-353.202, 7.61, 218.017) -> (-353.202, 606, 218.017)

Print something out in wire-cell-python

```
{'desc': 27, 'code': 'w', 'chid': 658, 'head': [-3522.189999999952, 801.2310510751832, 2305.719928942551], 'ident': 1058, 'index': 1058, 'seg': 2, 'tail': [-3522.189999999952, 76.10150384226132, 1784.5279077154623], 'wpid': 1}
{'desc': 129, 'code': 'w', 'chid': 659, 'head': [-3522.189999999952, 793.2311230390479, 2305.719828819827], 'ident': 1059, 'index': 1059, 'seg': 2, 'tail': [-3522.189999999952, 76.10154883125509, 1790.2778451227769], 'wpid': 1}
{'desc': 23, 'code': 'w', 'chid': 660, 'head': [-3522.189999999952, 785.2311950029122, 2305.7197286971036], 'ident': 1060, 'index': 1060, 'seg': 2, 'tail': [-3522.189999999952, 76.1015938202481, 1796.027782530092], 'wpid': 1}
{'desc': 19, 'code': 'w', 'chid': 661, 'head': [-3522.189999999952, 777.2312669667771, 2305.71962857438], 'ident': 1061, 'index': 1061, 'seg': 2, 'tail': [-3522.189999999952, 76.10163880924141, 1801.7777199374063], 'wpid': 1}
{'desc': 2, 'code': 'w', 'chid': 1103, 'head': [-3527.1099999999606, 3632.538002026343, 2.3472202385006633], 'ident': 703, 'index': 444, 'seg': 1, 'tail': [-3527.1099999999606, 432.54089738183046, 2302.3684659383616], 'wpid': 2}
{'desc': 125, 'code': 'w', 'chid': 2533, 'head': [-3532.0200000000145, 6060.000000000299, 2175.375528818724], 'ident': 453, 'index': 453, 'seg': 0, 'tail': [-3532.0200000000145, 76.1000000003122, 2175.3751371520575], 'wpid': 4}
```

