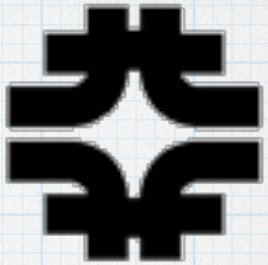


The option for Pythia 8 decays of tau and charm-hadrons in Legacy and Refactorized LArG4

GENIE, Geant4, Pythia{6,8}: tau and charm hadron decays

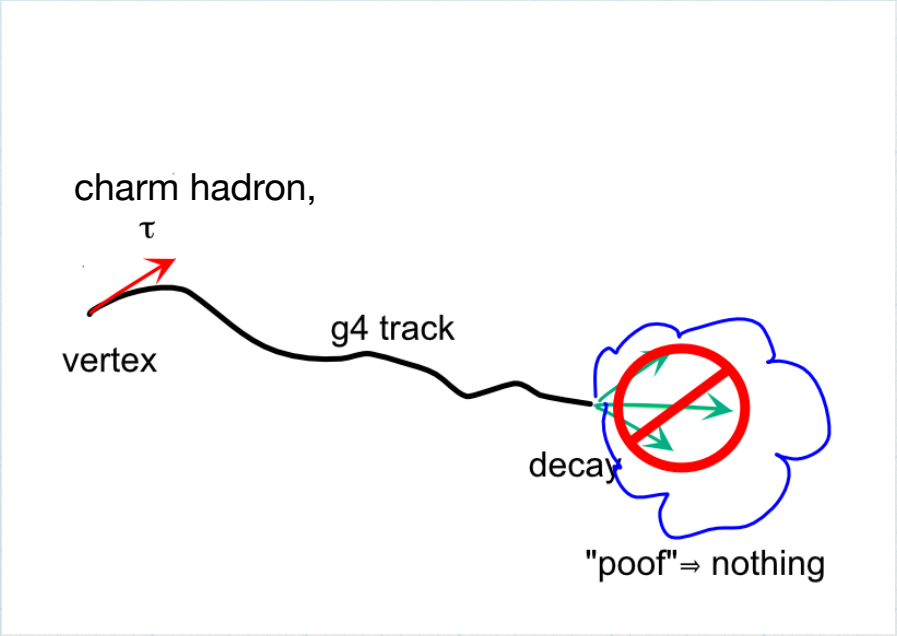
Robert Hatcher, Julia Yarba

Fermilab Computing Division

LArSoft Coordination Mtg 2021-11-30

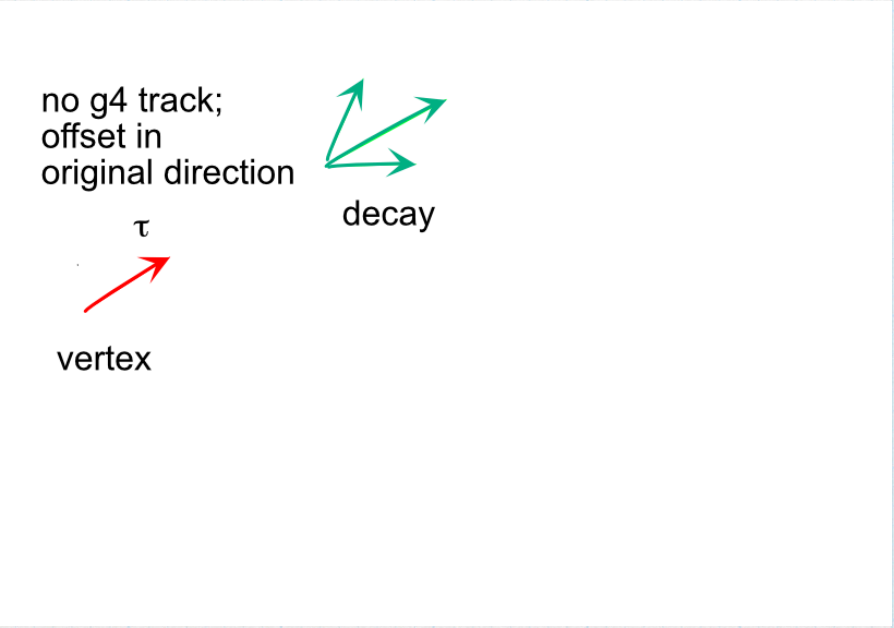
Past

- For a high enough energy in the flux spectrum, GENIE will produce charm hadrons in final state.
- Additionally for an oscillated flux, tau leptons.
- The default handling would be to hand these over to Geant4 for propagation and decay.
- circa ~ 2012 users were surprised to find that Geant4 didn't generate secondaries for charm or taus...

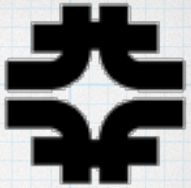


Cheap Solution

- Let GENIE decay the particle, using already integrated `pythia6`, before hand-off to Geant4.
- This can be controlled by a GENIE XML configuration file
  - packaged as `genie_phyopt` UPS product
  - either `dkcharmtau` or `dkcharm` qualifier



Limits to this Solution



- This was sufficient for NOvA -- offset was small, in general, due to low momenta of tau or charm; and energy was once again conserved.
- NOvA's spatial resolution wasn't sufficient to see such offset.

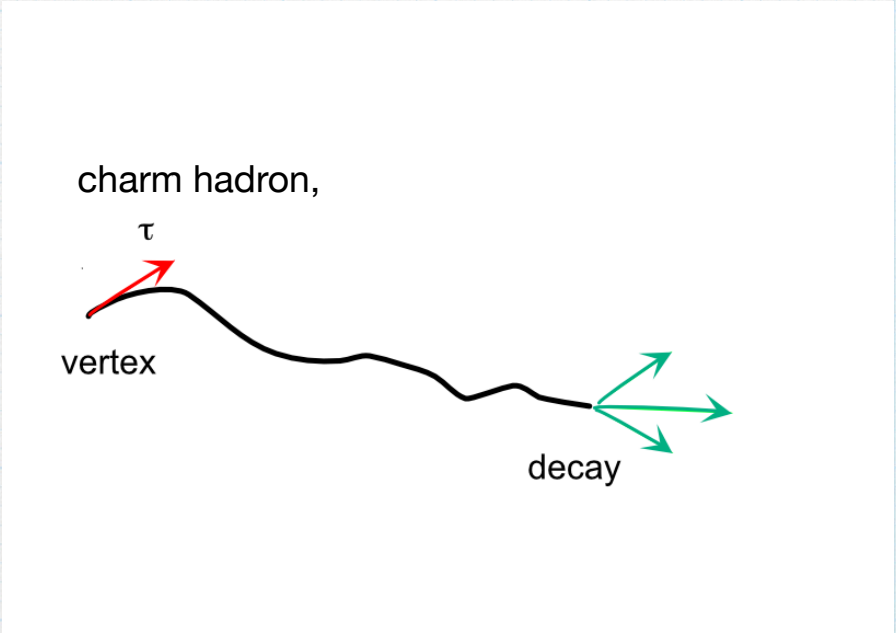
- Then LArSoft came along and the solution was similarly available to them.
  - eventually used by DUNE (upon Robert's insistence).
  - apparently not implemented by microBooNE
    - perhaps not an issue if never tau or charm production
  - ?? other LArSoft experiments ??

Past

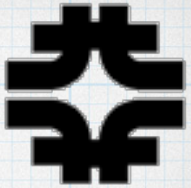
- At some point, Geant4 implemented a rudimentary tau decay (but still not charm hadrons)
  - not sophisticated (phase space only?)
- And … apparently … broken in some cases
  - occasionally generate absurd momenta
- Noticed when running without `genie_phyopt` UPS product setup.

Future is Now: Better Solution

- Let Geant4 do its job, but use Geant4's ability to plug in alternatives to do the decays of charm and tau.



Future is Now: Better Solution

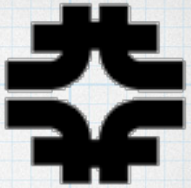


- Let Geant4 do its job, but use Geant4's ability to plug in alternatives to do the decays of charm and tau.

- Make use of the G4 extensible physics list factory to supplement regular standard physics lists.

  - `FTFP_BERT` ⇒ `FTFP_BERT+PY8TAUDK+PY8CHARMDK`

What Does PY8TAUDK Do?



- Removes Geant4's default decay table for taus
- Register taus to be decayed by `Py8TauDecayer` which is a `G4VPhysicsConstructor` with an interface to pythia8
  - via an instance of a `G4VExtDecayer` : `Py8Decayer`
  - code lives in `nug4/AdditionalG4Physics`

- Similarly PY8CHARMDK does that same for PDG code signaling charm (or bottom) quarks
  - e.g. 411 = D^+, 4122 = Λ_c^+
- Both also register any other particles that don't already have a decay table in Geant4 to use pythia8 as well

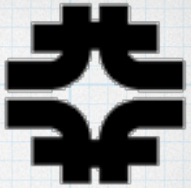
Present

- Works for both Legacy LArG4 and refactorized LArG4; relies on the extensible physics list factory supplied by Geant4 (now available to LegacyG4)
- No longer setup `genie_phyopt -q dkcharmtau` instead use `-q dkstandard:resfixfix` .
  - GENIE then leaves charm and taus undecayed, but importantly fixes an issue where a resonance, P33(1600), could be generated but GENIE didn't know how to decay it, leaving events with just the lepton but no hadronic recoil.
  - `FTFP_BERT+PY8TAUDK == FTFP_BERT+PY8TAUDK+PY8CHARMDK`
  - `FTFP_BERT+PY8CHARMDK !=`

Legacy LArG4 - changes to fcl

```
physics: {
   end_paths: [
      "stream1"
   ]
   producers: {
      largeant: {
         G4PhysListName: "larg4::PhysicsList+PY8TAUDK"
         CheckOverlaps: false
         DebugVoxelAccumulation: 0
         DumpLArVoxelList: false
         DumpParticleList: false
         GeantCommandFile: "LArG4.mac"
         KeepParticlesInVolumes: []
         SmartStacking: 0
         UseModLarqlRecomb: false
         VisualizeEvents: false
         module_label: "largeant"
         module_type: "LArG4"
      }
```

Refactorized LArG4 - changes to fcl

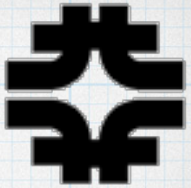


```
-
-       larg4Main: {
-          enableVisualization: false
-          macroPath: ".:./macros"
-          module_label: "larg4Main"
-          module_type: "larg4Main"
-          visMacro: "vis.mac"
+       largeant: {
+          CheckOverlaps: false
+          DebugVoxelAccumulation: 0
+          DumpLArVoxelList: false
+          DumpParticleList: false
+          GeantCommandFile: "LArG4.mac"
+          KeepParticlesInVolumes: []
+          SmartStacking: 0
+          UseModLarqlRecomb: false
+          VisualizeEvents: false
+          module_label: "largeant"
+          module_type: "LArG4"
       }
-

    simulate: [
       "rns",
-      "larg4Main"
+      "largeant"
    ]
```

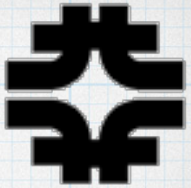
```
-###### start added services for larg4Main
-   ActionHolder: {
-      service_type: "ActionHolder"
-   }
-   DetectorHolder: {
-      service_type: "DetectorHolder"
-   }
-   ExampleGeneralAction: {
-      name: "exampleGeneral"
-      service_type: "ExampleGeneralAction"
-   }
-   LArG4Detector: {
-      category: "world"
-      gdmlFileName_: "dune10kt_v4_1x2x6.gdml"
-      service_type: "LArG4Detector"
-   }
-   MCTruthEventAction: {
-      service_type: "MCTruthEventAction"
-   }
-   ParticleListAction: {
-      SparsifyMargin: 0.1
-      service_type: "ParticleListAction"
-   }
-   PhysicsList: {
-      DumpList: false
-      PhysicsListName: "FTFP_BERT+PY8TAUDK+PY8CHARMDK"
-      ScintillationByParticleType: false
-      enableAbsorption: false
-      enableBoundary: false
-      enableCerenkov: false
-      enableMieHG: false
-      enableRayleigh: false
-      enableScintillation: false
-      enableWLS: false
-      service_type: "PhysicsList"
-   }
-   PhysicsListHolder: {
-      service_type: "PhysicsListHolder"
-   }
-###### end added services for larg4Main
```

Better Solution Status



- Julia Yarba implemented the interface to Pythia8 in a stand-alone Geant4 app
- Robert Hatcher integrated it into the factory system so that physics process can be added by configuring the PhysicsList name and integrated it into nug4 (with a link in larg4)
- Issues:
  - Integrated into NuG4 package — only avail in art 3.09
    - added dependence on Pythia8
  - GENIE sets tau polarization (for RES + DIS, but not QES) in its event record -- to be fixed in v3_02
    - polarization info is transferred to MCTruth in LArSoft record and from MCTruth to G4PrimaryParticle (nug4 ConvertMCToG4 and/or larg4/pluginActions/ MCTruthEventAction_service.cc)

Better Solution Status



- Issues continued:
  - Linked to LArG4main to get library _loaded_ at run time
    - nothing needs be done on user side other than load the library to make it available; no explicit compile/link time references
  - Possibility to do the same kind of interface for the `tauola` package. `Pythia8` documentation implies that the decay models are slightly different in some cases.
  - https://indico.cern.ch/event/300387/contributions/686167/attachments/565826/779490/ilten140915.pdf
  -