pECal fiber implementations in DD4hep

Zhongling Ji

UCLA

February 8, 2022

Problem

UCLA

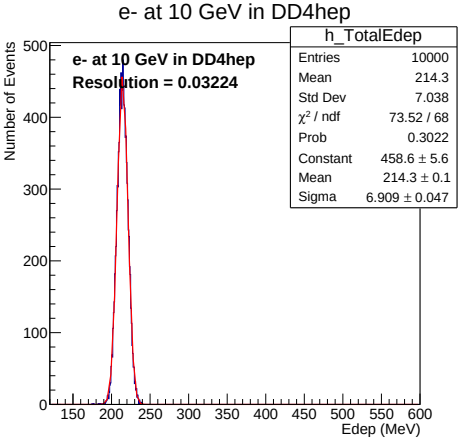
- Problem: pECal with fibers uses $\sim$6 GB memory in NPSim and $\sim$8.8 GB memory in Juggler. It will use more memory when other detectors were included.
- Reason: storing *PlacementPath–VolumeID* mapping in *m_geo.g4Paths* uses large memory.
- Solution: group fibers in each module as a single readout channel.
- NB: This is the reverse logic of segmentation. In segmentation, we have one physical structure with many readout channels. However, in pECal fibers, we have many physical structures (fibers) with a single readout channel (each module).
- Result: reduce memory usage to $<$700 MB, which is the same as that without fibers.

Two methods of implementations

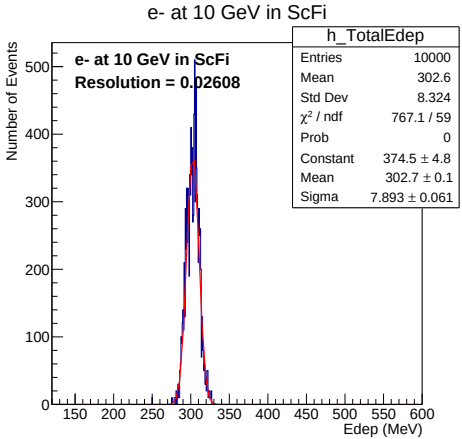
UCLA

- Method I: Change DD4hep to let it only store the *VolumeID* for each module instead of each fiber.
  - Pros: Easy to build fibers.
  - Cons: Need a small change in DD4hep.
- Method II: Set the whole module as a sensitive detector and cover the insensitive areas by daughter radiators.
  - Pros: No need to change DD4hep.
  - Cons: Much more coding when building the fibers and runs $\sim$20% slower than method I.
- ~~Method III~~: Use *UnionSolid* to build the shape for all fibers. In principle, it works. In practical, it takes infinite time to build the shape for a large number of fibers.
- ~~Method IV~~: Use *Assembly* to group fibers in each module. It doesn't work since DD4hep requires the sensitive detector to be a real *Volume* instead of a *Assembly*.

Default materials

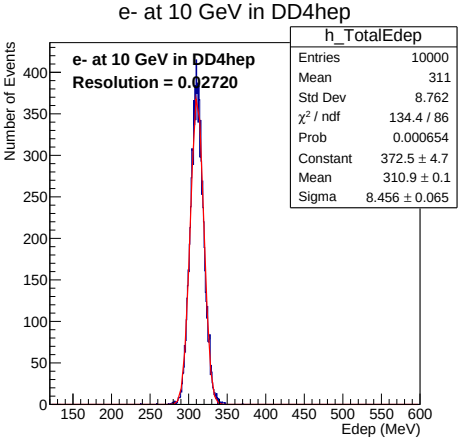


DD4hep: fibers by covering (Method II)
Polystyrene: $C_{19}H_{21}$
Tungsten: $W_{0.93}Ni_{0.061}Fe_{0.009}$

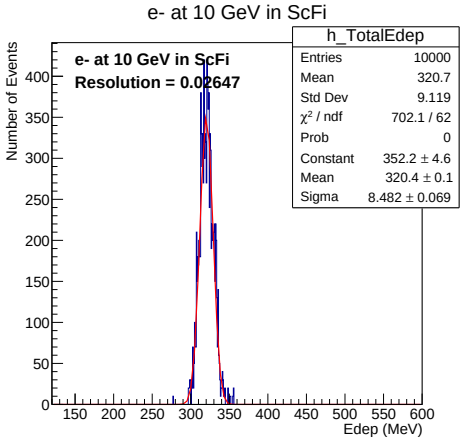


Geant4
Polystyrene: C_8H_8
Tungsten: $W_{0.97}\text{Polystyrene}_{0.03}$

Same Geant4 materials



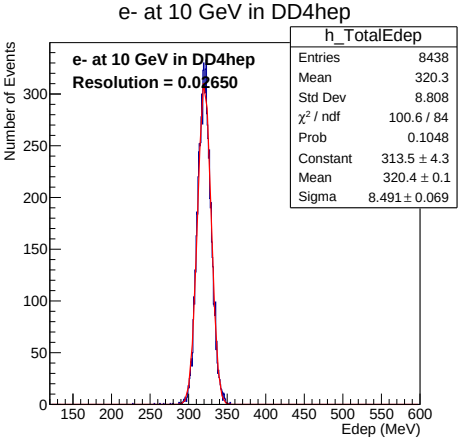
DD4hep: fibers by covering (Method II)
Geant4 materials
Birks = 0



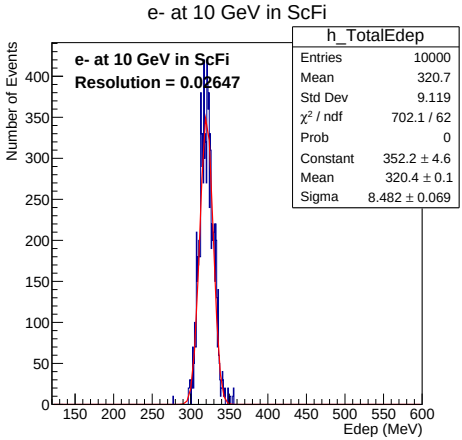
Geant4
Explicitly defined Geant4 materials
Birks = 0

Putting fibers directly

UCLA

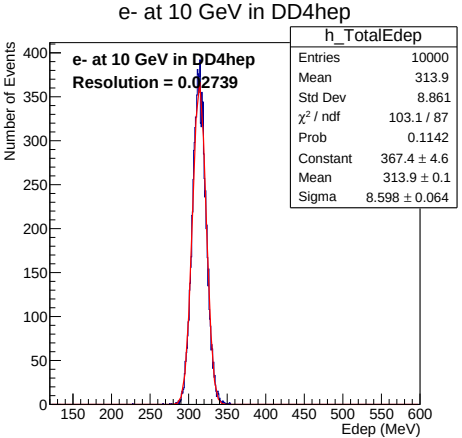


DD4hep: put fibers directly (Method I)
Geant4 materials
Birks = 0

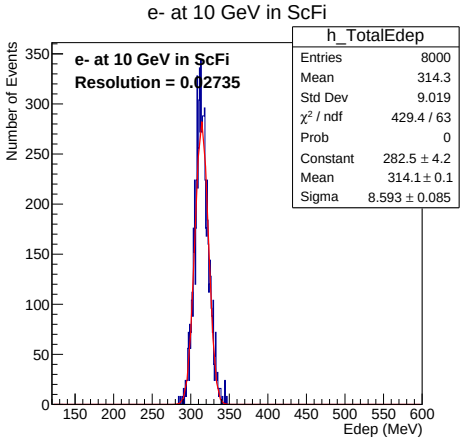


Geant4
Explicitly defined Geant4 materials
Birks = 0

Grouping fibers in each layer



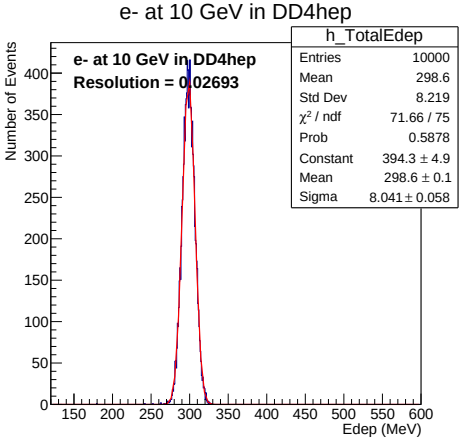
DD4hep: group fibers in each layer (Method I)
Geant4 materials
Birks = 0



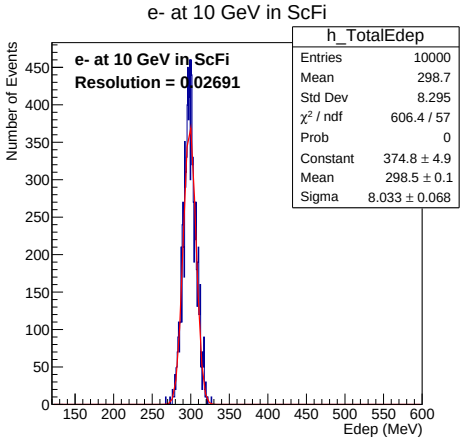
Geant4: group fibers in each layer
Explicitly defined Geant4 materials
Birks = 0

Adding Birks constant

UCLA

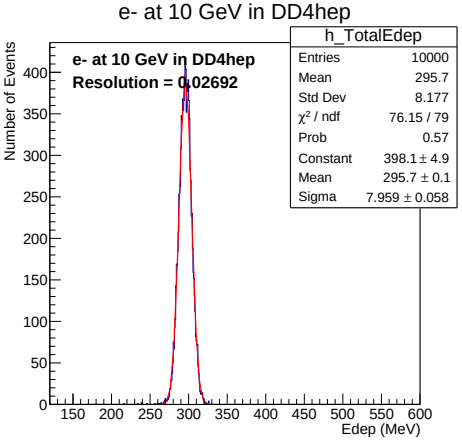


DD4hep: group fibers in each layer (Method I)
Geant4 materials
Birks = 0.126*mm/MeV

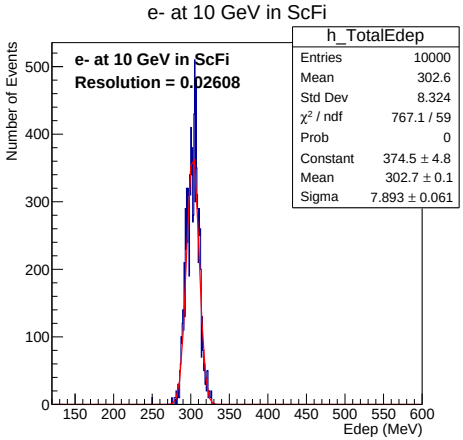


Geant4: group fibers in each layer
Explicitly defined Geant4 materials
Birks = 0.126*mm/MeV

Final result



DD4hep: fibers by covering (Method II)
Geant4 materials
Birks = 0.126*mm/MeV



Geant4
Implicitly defined Geant4 materials
Birks = 0.126*mm/MeV

Summary

UCLA

- The default materials are different in Geant4 and DD4hep. It is important to use the correct one though they have the same name.
- The default range cuts are different in Geant4 and DD4hep.
- The default implementation of birks' law in DD4hep, which is implemented in Geant4, is slightly different from Ryan's implementation. However, it doesn't change the result when using the single electron beam.
- Grouping fibers in each layer by a volume will slightly change the results. When tuning the simulation by the test beam data, we should keep this in mind.