

*Reviewing past & present practice, and discussing what to do in the future*

---

# When to abandon looping particles?

J. Apostolakis

---

---

# Issue

---

- ❖ Particles trapped in a **strong magnetic field** in a **low-density** material are a CPU time sink - their **tight helical trajectories** require thousands of integration steps
- ❖ Typically they are **low-energy electrons**
- ❖ FNAL experiment experience shows they all be unstable particles in decay volumes - which must be kept!
- ❖ Tracks have been dropped if their energy thresholds were used

---

# Background

---

- ❖ An integration segment can be limited by a sagitta exceeding  $d\_chord$  or by integration accuracy needs - due to oscillatory solutions this occur for angle  $\sim \pi/3$
- ❖ Integration 'failure' occurs if
- ❖ They are identified by `G4PropagatorInField` and marked as '**looping**'
- ❖ The **Transportation** process must use its broader knowledge of particle type

---

# The problem

---

- ❖ Krzysztof Genser reported that 'signal' particles were being silently killed by transportation before they could decay in the decay volume



---

# Historic & recent behaviour

---

- ❖ Transportation has two thresholds (since 7.0, Dec 2004)
  - ❖ 'Warning' energy - below which a track was killed without report/warning, in the first step it is reported as 'looping'
  - ❖ 'Important' energy - above this a track is given multiple looping steps before being killed
    - ❖ number of step = 'fThresholdTrials' parameter
- ❖ Until release 10.5-β, the values were:
  - ❖  $E_{\text{warning}} = 100 \text{ MeV}$
  - ❖  $E_{\text{important}} = 250 \text{ MeV}$
  - ❖  $N_{\text{trials}} = 10$

---

# Latest revision in 10.5- $\beta$

---

- ❖ After Krzysztof's report, changed thresholds
  - ❖  $E_{\text{warning}} = 1 \text{ keV}$  (from 100 MeV)
  - ❖  $E_{\text{important}} = 1 \text{ MeV}$  (from 250 MeV)
- ❖ Improved printing of information for 'loopers'. Added
  - ❖ Momentum direction (already had position, energy)
  - ❖ Volume name, material & its density.

---

# Changing default values

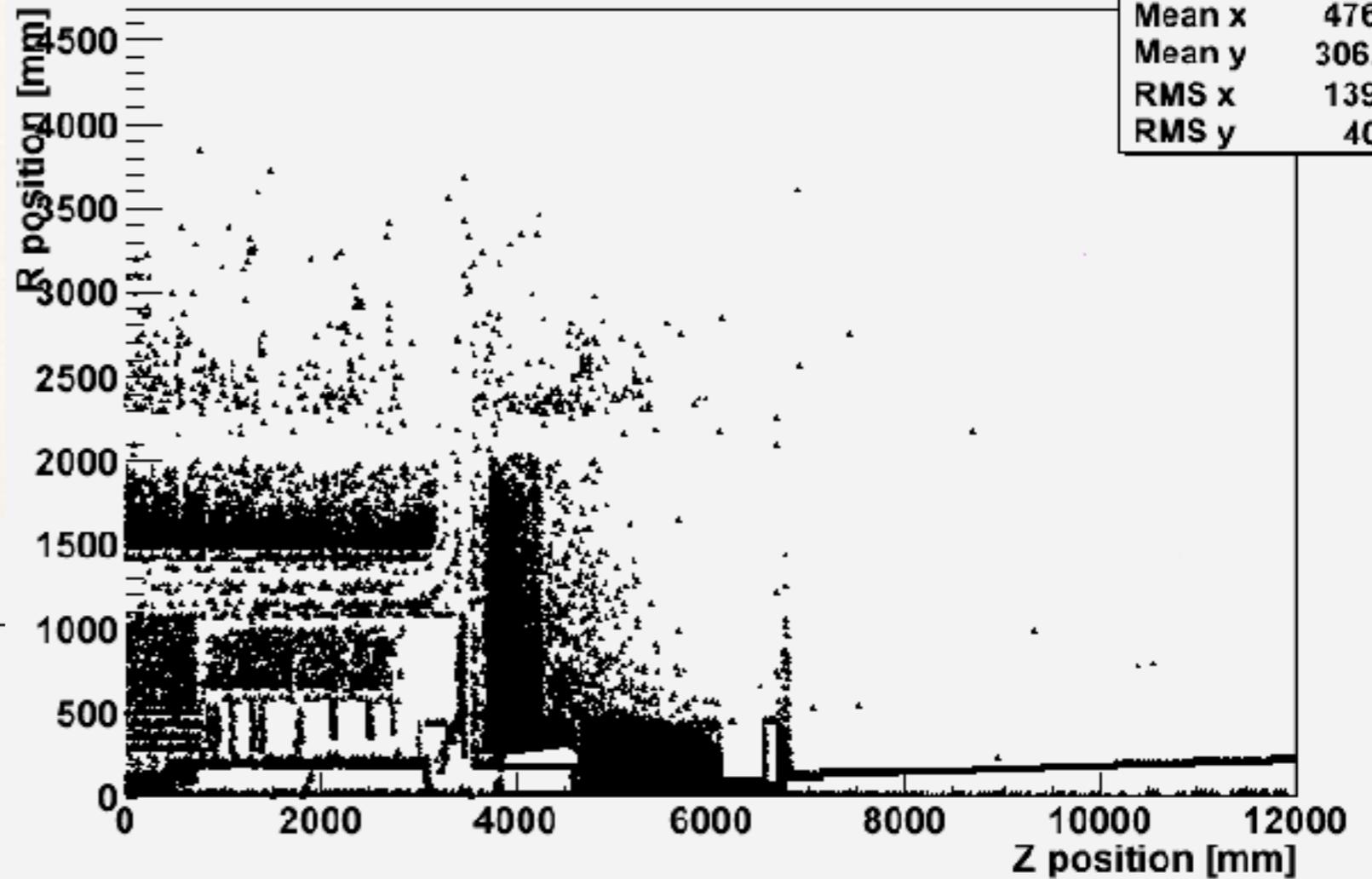
---

- ❖ Thresholds can be changed with methods of G4Transportation's
- ❖ Today Transportation object is shared by all particles when using modular physics lists.
- ❖ Obtaining pointer to Transportation object
  - ❖ is challenging and long-winded.
  - ❖ can be simplified greatly by adding new method in `GetPhysicsListHelper`
- ❖ Different values for each particle type are NOT current supported
  - ❖ Of course they could be achieved by Start/End OfTracking action.

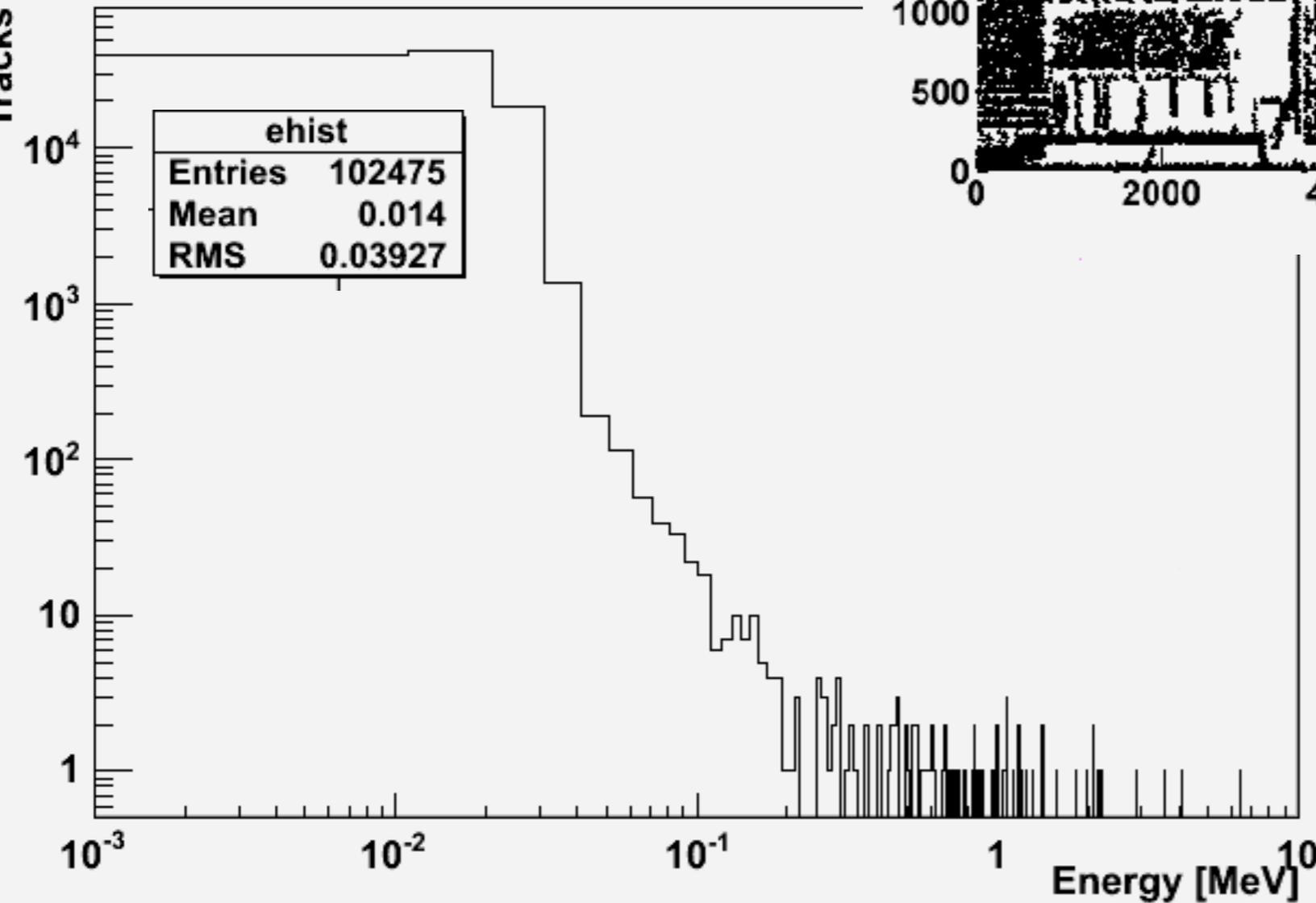
Sample of killed looping particles from ATLAS in 2009.

'Candidate' G4 release in range

- [Geant4 8.3](#) (June 2007)
- [Geant4 9.2](#) (Dec 2008)



post	
Entries	102470
Mean x	4760
Mean y	306.7
RMS x	1398
RMS y	400



ehist	
Entries	102475
Mean	0.014
RMS	0.03927

---

# Requests

---

- ❖ Enable filtering of 'looping' particles by particle type.
- ❖ This could enable different choices, including:
  - ❖ never kill unstable particles; or
  - ❖ kill only electrons or stable particles.



---

# Topics for discussion & decision

---

- ❖ Need to agree
  - ❖ new default behaviour - for release 10.5
  - ❖ what tests should be undertaken (with experiments)
  - ❖ whether to hand the user full control of 'warning' and killing - potentially via a new filter interface
    - ❖ allowing an experiment/user to tailor it fully, using knowledge of geometry, key particle types, etc.

Backup slides

- 
- 
- ❖ Thresholds can be changed with G4Transportation's
    - ❖ SetThresholdWarningEnergy( G4double val )
    - ❖ SetThresholdImportantEnergy( G4double val )
    - ❖ SetThresholdTrials( int num)
  - ❖ Yet obtaining pointer to Transportation object (shared by all processes) is long-winded:
    - ❖ auto pm= G4Electron::G4Electron->GetProcessManager();
    - ❖ auto transportProc = pm->GetProcess( );
  - ❖ Could be simplified greatly by adding new method:
    - ❖ G4VProcess\* G4PhysicsListHelper()::GetTransportationProcess() const;
    - ❖ bool G4PhysicsListHelper()::IsCoupledTransportationUsed();
    - ❖ G4VProcess candProc= GetPhysicsListHelper::GetPhysicsListHelper()->GetTransportationProcess();

```
----- WWW ----- G4Exception-START ----- WWW -----
*** G4Exception : Transport-001-ExcessSteps
    issued by : G4Transportation::AlongStepDoIt()
Transportation is killing track that is looping or stuck.
Track is e- and has 0.448242 MeV energy ( pre-Step = 0.448242 )
momentum = (0.397018,-0.683593,-0.184662) mag= 0.811802
position = (-321.83,-998.959,446.186) volume = TOB material Air density = 0.001214 g/cm^3
Number of Steps by this track: 8
Number of propagation trials = 1
( Number of *calls* of Transport/AlongStepDoIt = 538290 )
----- WWW ----- G4Exception-END ----- WWW -----
```

```
===== Recommendations / advice =====
Recommendations to address this issue (Transport-001-ExcessSteps)
This warning is controlled by the SetThresholdWarningEnergy method of G4Transportation.
Current value of 'warning' threshold= 0 MeV
- If 'unimportant' particles (with energy low enough not to matter in your application, then increase its value.
- If particles of high-enough energy to be important are being killed, you can
  a) Increase the trial steps using the method SetThresholdTrials(). The default value was 10, and the current
value is 10
  b) Increase the energy which you consider 'important' (above this they are never killed), using the method
SetThresholdImportantEnergy()
    Note: this can incur a potentially high cost in extra simulation time if tracks require very large number of
integration steps .
  c) investigate alternative integration methods
    e.g. Helical methods for uniform or almost uniform fields or else higher order RK methods such as DormandPrince78
=====
```