Error propagation in Geant4.9.0

Pedro Arce (CIEMAT)

12th GEANT4 WORKSHOP

Hebden Bridge (UK), 13-19 September 2007

Outlook

- What is GEANT4E
- GEANT4E components
  - Trajectory state
  - Target to propagate
  - Track propagator manager
  - Physics
  - Magnetic field
  - Track error propagation
- Comments on backwards tracking
- GEANT4E example in GEANT4
- Future developments
- Summary and plans

What is GEANT4E

- Track reconstruction needs to match signals in two detector parts
  - Propagate tracks from one detector part to another and compare with real measurement there
  - Make the average between the prediction and the real measurement

  ⇒ it needs the track parameter errors

- Many experiments have used in the past GEANE (based on GEANT3) or their 'ad hoc' solution

GEANT4e provides this functionality for the reconstruction software in the context of GEANT4

Released in geant4.9.0

Trajectory state: G4ErrorTrajState

- User defines the initial track parameters in a given point of the trajectory: **G4ErrorTrajState**
  - Particle type
  - Position
  - Momentum
  - Track errors (5x5 HepSymMatrix)
  - Initial surface where parameters are defined
- Two different trajectory states:
  - G4Error**Free**TrajState:
    - 1/p, λ, ϕ, y_perp, z_perp (p_x = p $\cos(\lambda)\cos(\phi)$, p_y = p $\cos(\lambda)\sin(\phi)$, p_z = p $\sin(\lambda)$, x_perp || trajectory, y_perp parallel to x-y plane)
  - G4Error**Surface**TrajState: parameters on a plane in an arbitrary direction
    - 1/p, v', w', v, w (u,v,w is any orthonormal coordiante system, v, w on the plane)

End of propagation: G4ErrorTarget

- User defines up to where the propagation must be done: the target

- G4ErrorSurfaceTarget
  - Track is propagated until the surface is reached
  - The surface is not part of GEANT4 geometry
    - Using a ghost geometry would mean that propagation in field is done twice
    - **G4ErrorPropagationNavigator** takes care of the double navigation: on the full geometry and checking if surface is reached
      - overwrites ComputeStep() and ComputeSafety() to stop the navigation when the surface si reached
  - Several types defined
    - G4ErrorPlaneSurfaceTarget: infinite plane
    - G4ErrorCylSurface: infinite length cylindrical surface
    - ....
- G4ErrorTrackLengthTarget
  - Track is propagated until a certain track length is reached
  - Implemented as a G4VDiscreteProcess

End of propagation: G4ErrorTarget (2)

- G4ErrorGeomVolumeTarget
  - Track is propagated until the surface of a GEANT4 volume
    - Track enters
    - or track exits
    - or both
  - User can choose if volume refers to one or many G4LogicalVolume's, G4VPhysicalVolume's or G4VTouchable's, with a simple syntax:
    - `G4ErrorVolumeTarget("MuonCell")` G4LogicalVolume
    - `G4ErrorVolumeTarget("MuonCell#1")` G4VPhysicalVolume
    - `G4ErrorVolumeTarget("MuonChamber#3/MuonCell#2")` G4VTouchable
- G4ErrorUserDefinedVolumeTarget (TO BE DONE)
  - Track is propagated until the surface of a user-defined volume (outside the GEANT4 geometry)

Managing tracks: G4ErrorPropagatorManager

- User needs to propagate just one track

  ⇒ no need of run and events

- ✓ G4ErrorPropagatorManager creates a track and manages the step propagation
  - Creates a G4Track from the information given in the G4ErrorTrajState
  - Invokes G4SteppingManager to propagate one step

    ```
    fpSteppingManager::Stepping();
    ```
  - And propagates the track errors for this step

    ```
    G4ErrorPropagatorManager::PropagateError( aTrack );
    ```
  - Stops when G4Track stops or when the target is reached
    - If defined target is not reached it returns an error

- ❖ User can choose two ways of propagetion
  - Propagate until target is reached
  - Propagate step by step and return control after each step

Physics: G4ErrorPhysicsList

- Reconstruction software wants the average trajectory followed by the particle: G4ErrorPhysicsList
  - No multiple scattering
  - No secondaries allowed
  - No random fluctuations for energy loss
  - No hadronic processes
  - Huge cuts by default (User can change them with standard GEANT4 methods)
    - AlongStepGetPhysicalInteractionLength only limits step if $\Delta E > 0.2$ E
  - Negative energy loss when propagation is backwards
    - G4ErrorEnergyLoss, based on G4EnergyLossForExtrapolates

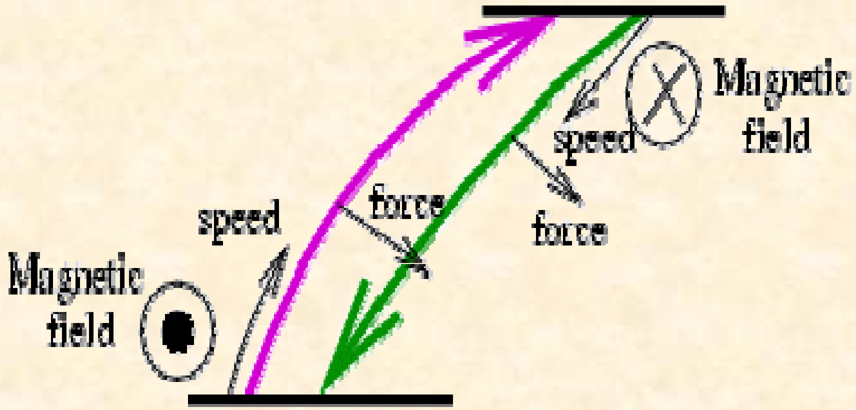
- User could define its own physics list (simply add it to the G4RunManager)
  - But it should account for backwards tracking

- Simple energy loss can be chosen: faster but less precise
  - But time in calculating energy loss is ~1 %

Magnetic field: G4ErrorMag_UsualEqRhs

- User defines the magnetic field in the standard GEANT4 way
- But GEANT4e has to handle the backwards propagation

  ⇒ Magnetic field has to be reversed



- G4ErrorPropagatorManager takes care of replacing G4Mag_UsualEqRhs by G4ErrorMag_UsualEqRhs, that overwrites EvaluateRhsGivenB() to reverse the field

Track error propagation

- Based on the equations of the European Muon Collaboration (same as GEANE)
  - ✓ Error from curved trajectory in magnetic field
  - ✓ Error from multiple scattering
  - ✓ Error from ionisation

- Formulas assume propagation along an helix
  - Need to make small steps to assure magnetic field constantness and not too big energy loss $\Rightarrow$ makes it slower

- Another approach to be studied: propagate the error together with the solving of the Runge-Kutta equations
  - Probably slower per step but could not need so many steps

E loss in backwards tracking

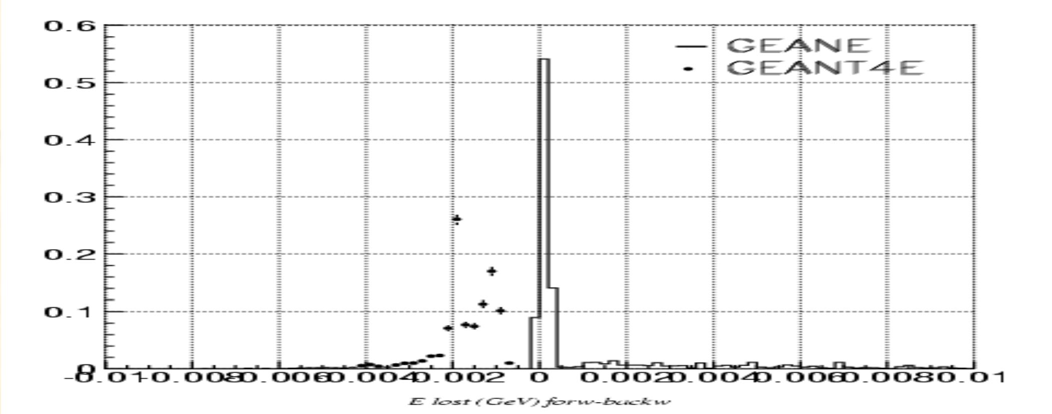
- When reconstruction software wants to know the trajectory that a track has described from a detector part to another, often the track has to be propagated **backwards**
  - ✓ The track has to gain energy instead of losing it
  - ✓ The value of the magnetic field has to be reversed

- But the energy lost (or gained) in one step is calculated
  - Forward tracking: using the energy at the beginning of the step
  - Backward tracking: using **the energy at the end of the step**
  - ❑ And similarly for the curvature in magnetic field

E loss in backward tracking (2)

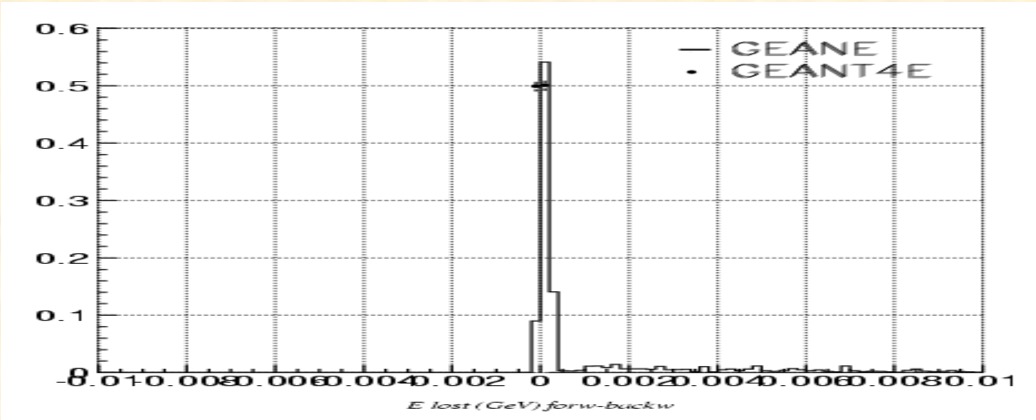
☹ This means that if you propagate a particle forwards and then backwards it would not recover the original energy

Difference in energy when a 20 GeV track is propagated forwards and then backwards
NO CORRECTION



☺ A correction is applied: dEdx is calculated with the energy at the end of step, then the lost energy is added and dEdx is recalculated again (☹ it needs #define private public)

Difference in energy when a 20 GeV track is propagated forwards and then backwards
CORRECTED



Limiting the step

There are three ways to limit the step

- Fixed length value

G4UImanager::GetUIpointer()->ApplyCommand("/geant4e/limits/stepLength MY_VALUE MY_UNIT");

- Maximum percentage of energy loss in the step (or energy gain is propagation is backwards)

G4UImanager::GetUIpointer()->ApplyCommand("/geant4e/limits/energyLoss MY_VALUE");

- Maximum difference between the value of the magnetic field at the beginning and at the end of the step.
  - Indeed what is limited is the curvature, or exactly the value of the magnetic field divided by the value of the momentum transversal to the field.

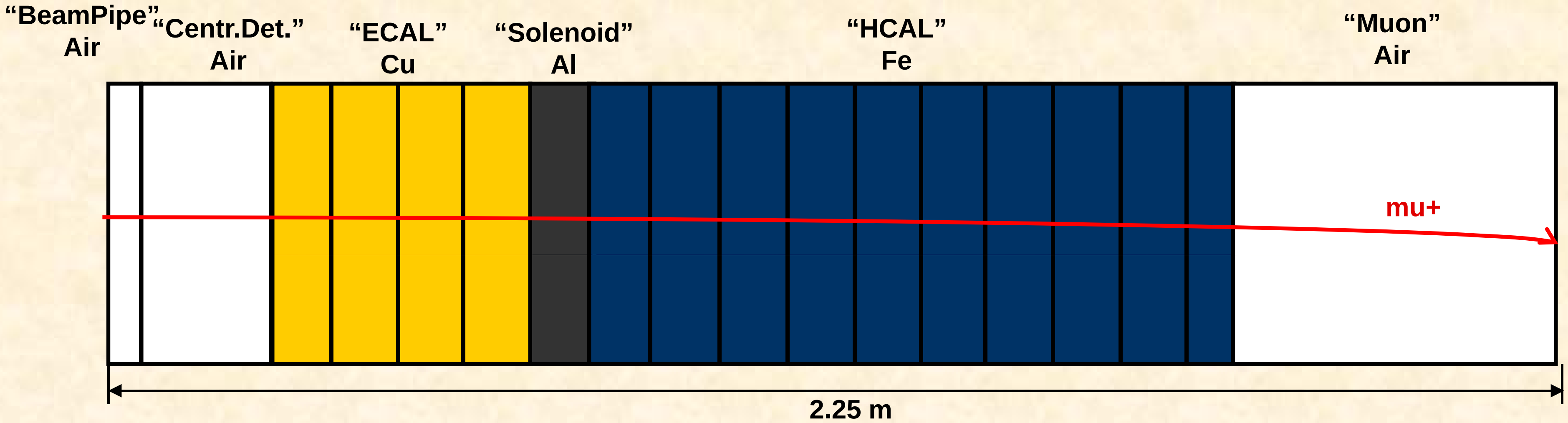
G4UImanager::GetUIpointer()->ApplyCommand("/geant4e/limits/magField MY_VALUE");

GEANTE example in GEANT4

extended/errorpropagation

(same example as the EREXAM1 example in GEANE and GEANT4E)

- Simple detector:



- Magnetic field -1 kGauss (0.01 Tesla)
- A track is propagated from the origin along all detectors
- User can choose target
  - Plane surface
  - Cylindrical surface
  - Volume
  - Track Length

Pulls

- We send 10k muons all along the detector with GEANT4 and compare the final state with the one sended with GEANT4e (only 1, there are no fluctuations)

- We plot the pulls (difference/error) for the five variables:
  - Position Y
  - Position Z
  - Angle phi
  - Angle lambda
  - 1/momentum

- ✓ They should be centered at 0 and with RMS = 1, if distributions were gaussians

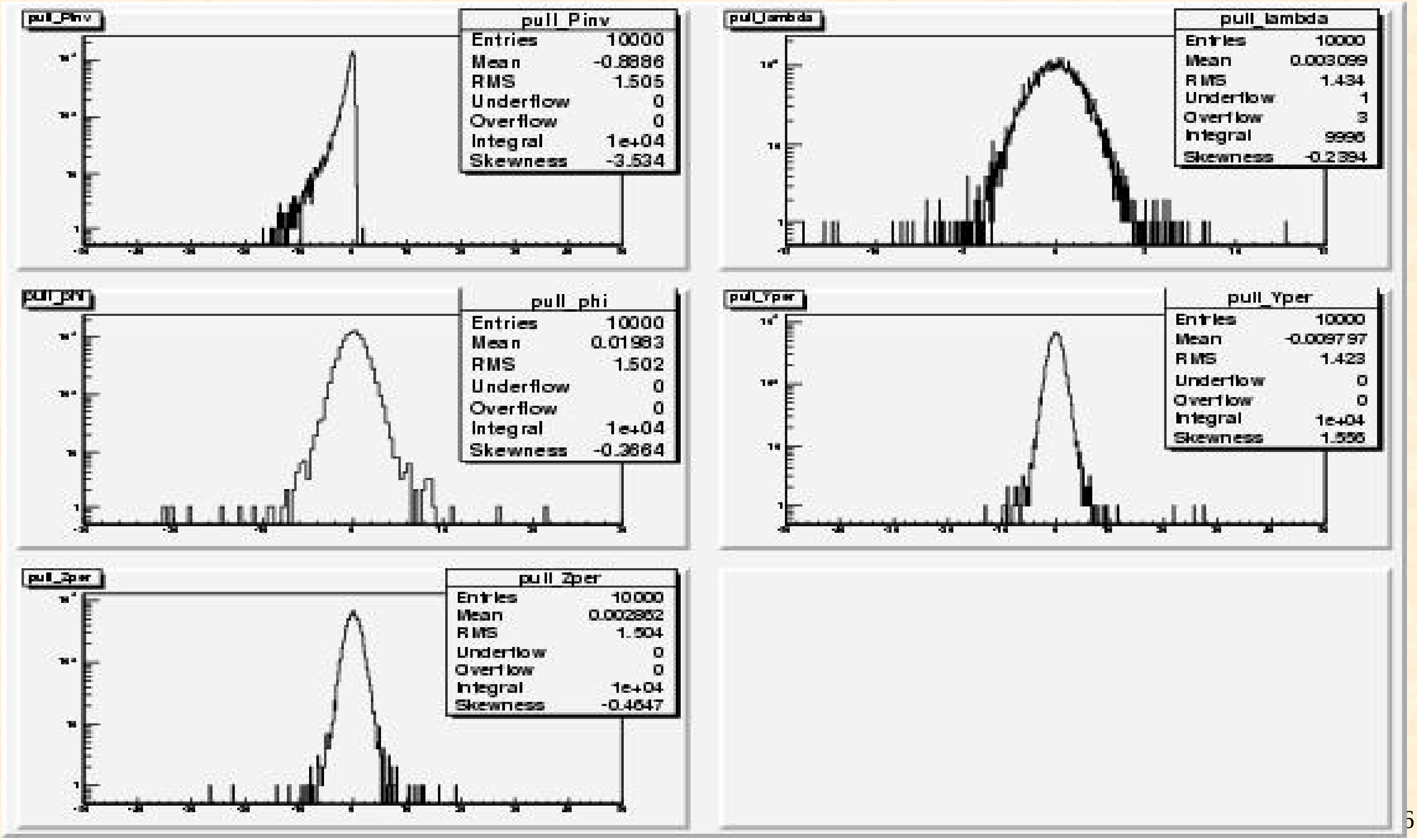
  ((n fact multiple scattering and energy loss are not gaussian)

Geant 4

Pulls results

Ciemat

10 GeV muons

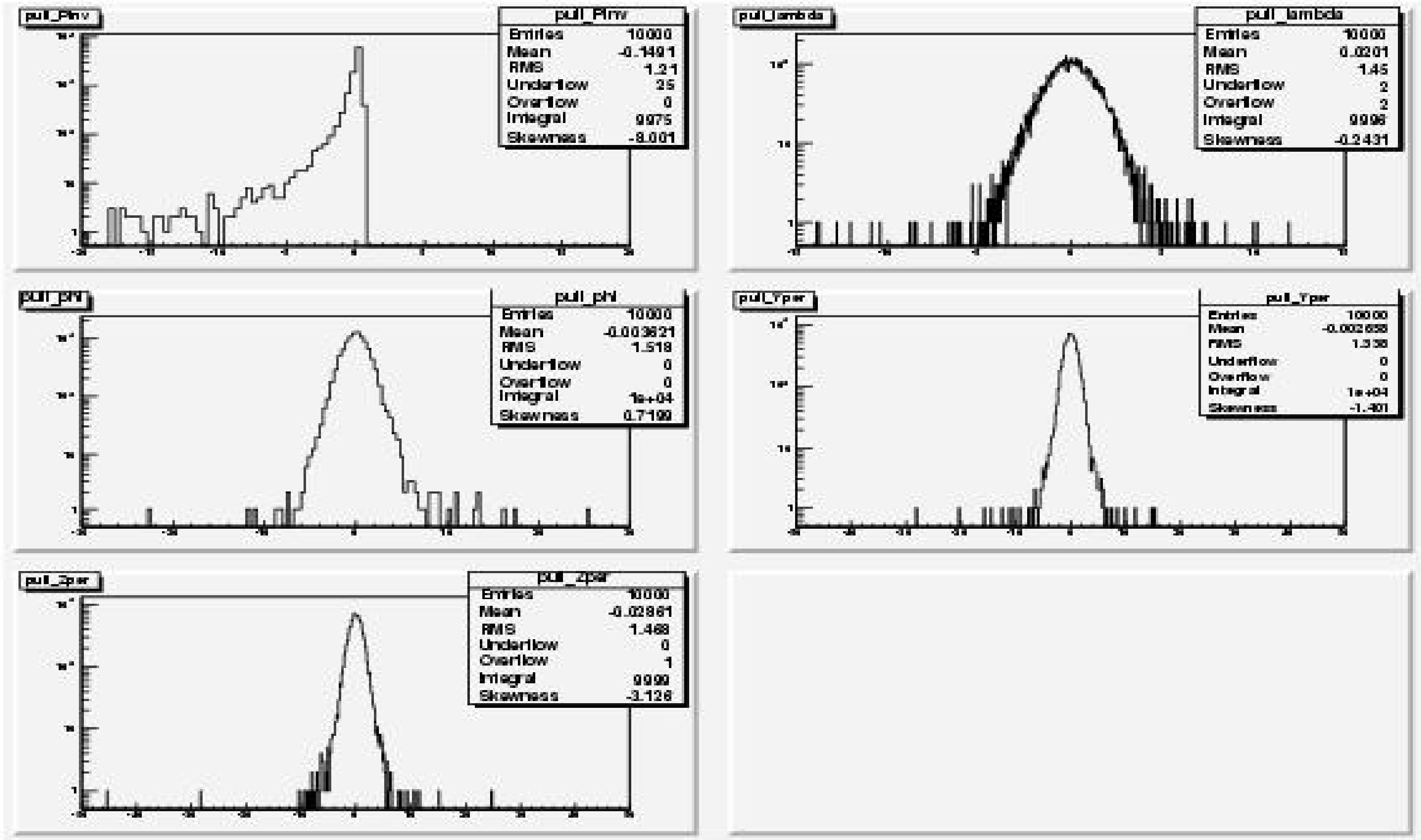


Geant 4

Pulls results

Ciemat

100 GeV muons

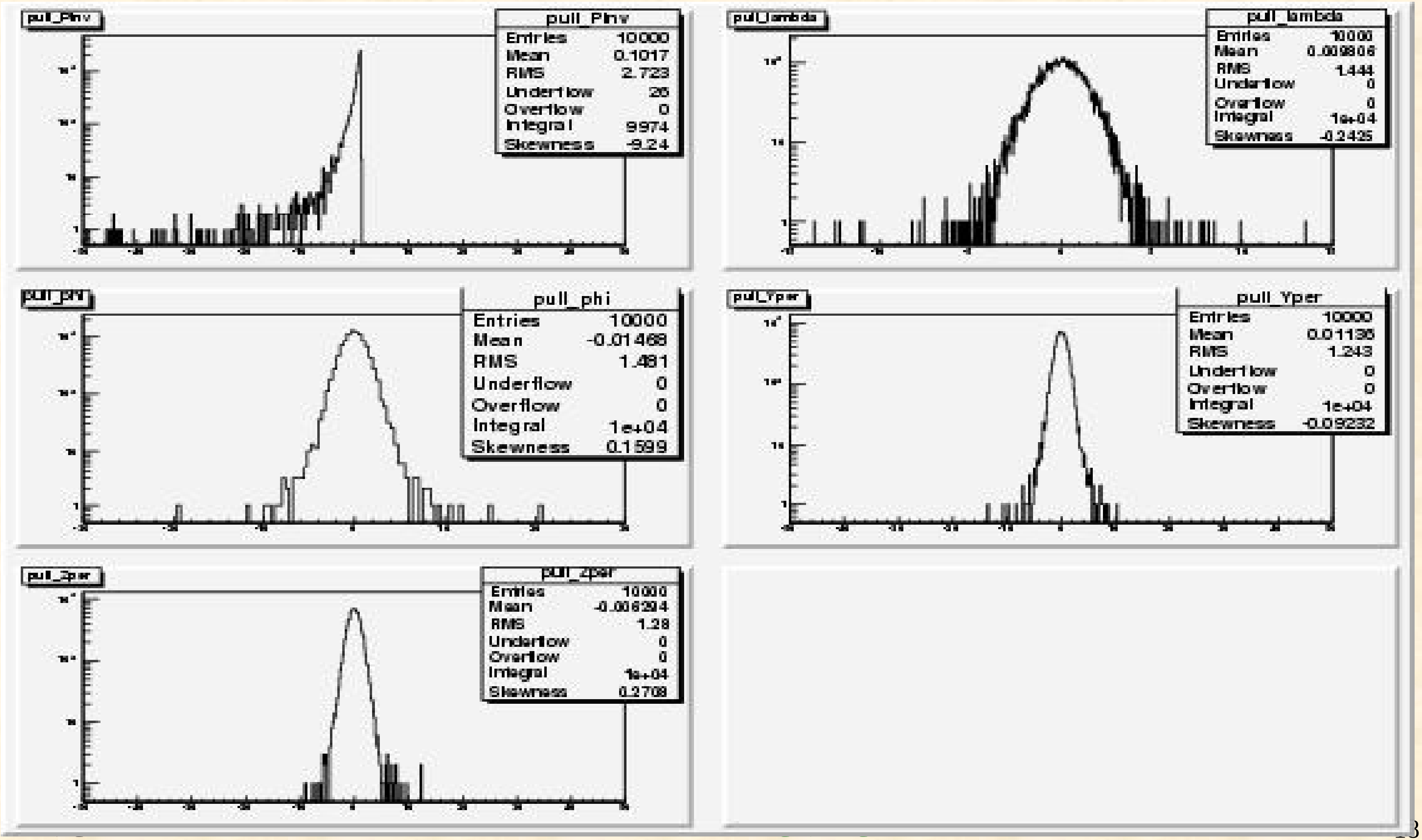


Geant 4

Pulls results

Ciemat

1000 GeV muons

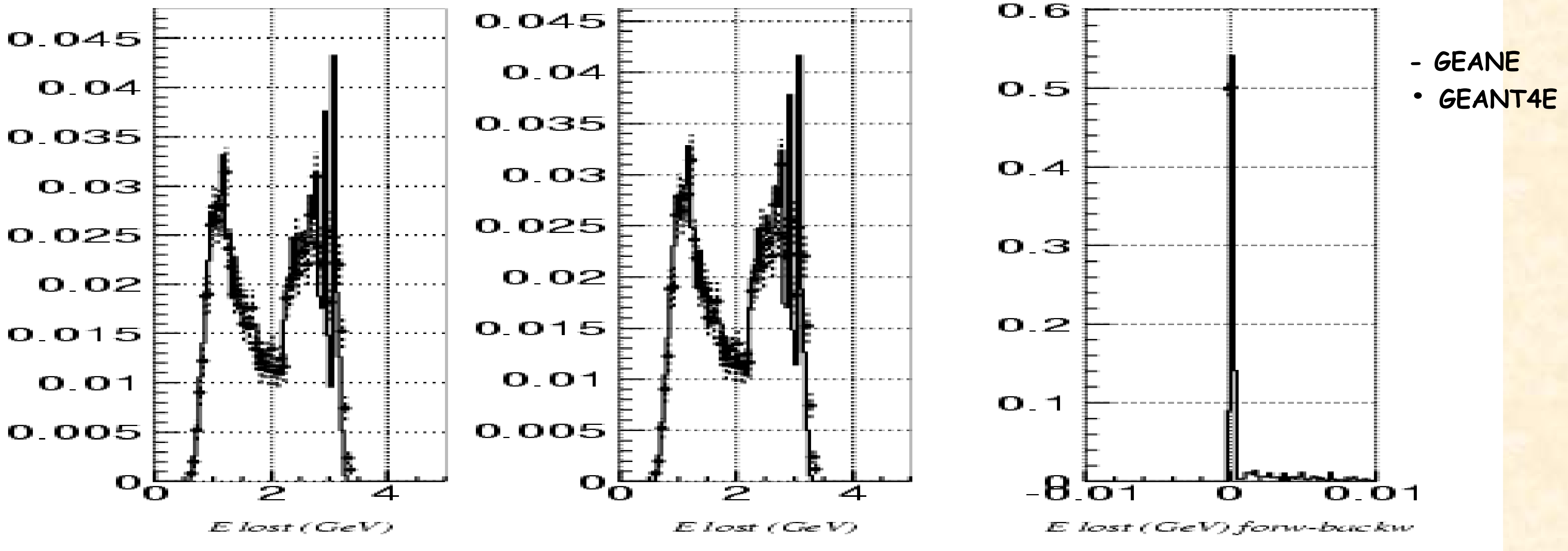


Comparison with GEANE

10000 mu+: 5-100 GeV, along X ± 10 degrees:

- **Energy lost (GeV)**

forwards backwards forwards+backwards



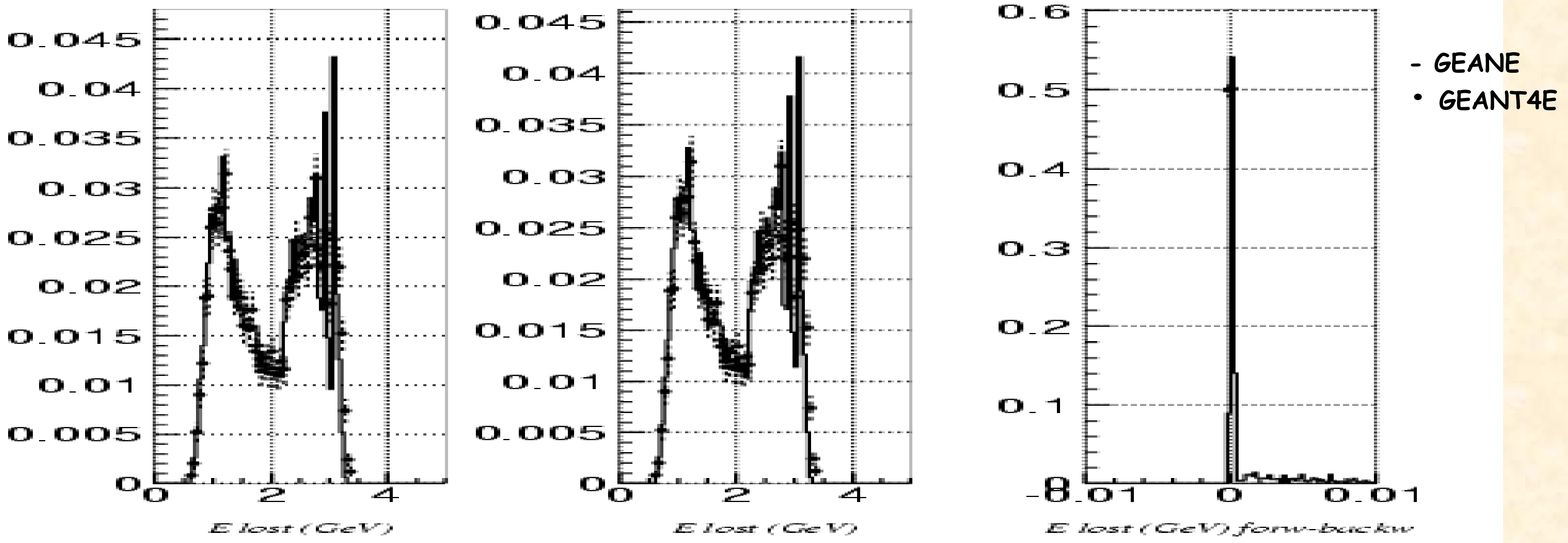
Not the same because GEANT4 propagation is more precise

Comparison with GEANE

10000 mu+: 5-100 GeV, along X ± 10 degrees:

- **Deviation in position (mm)**

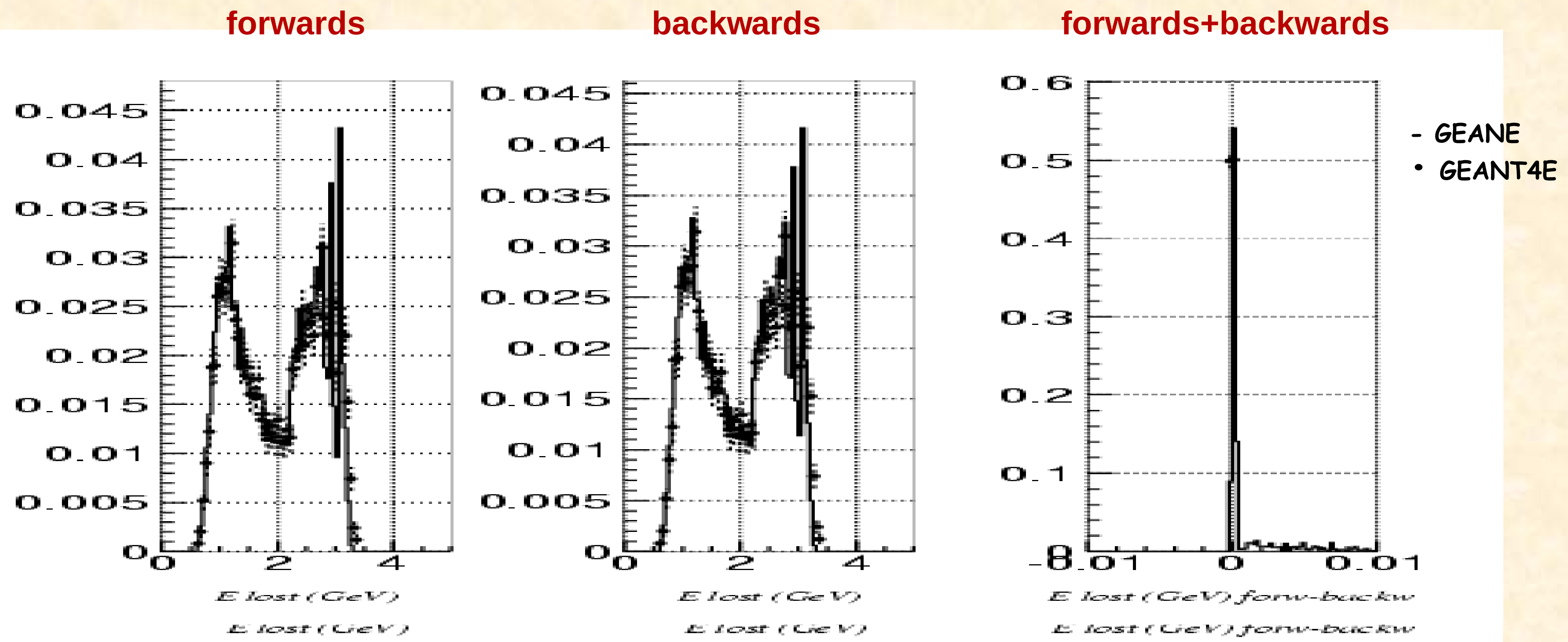
forwards **backwards** **forwards+backwards**



Comparison with GEANE

10000 mu+: 5-100 GeV, along X ± 10 degrees:

- **Deviation in angle (mrad)**

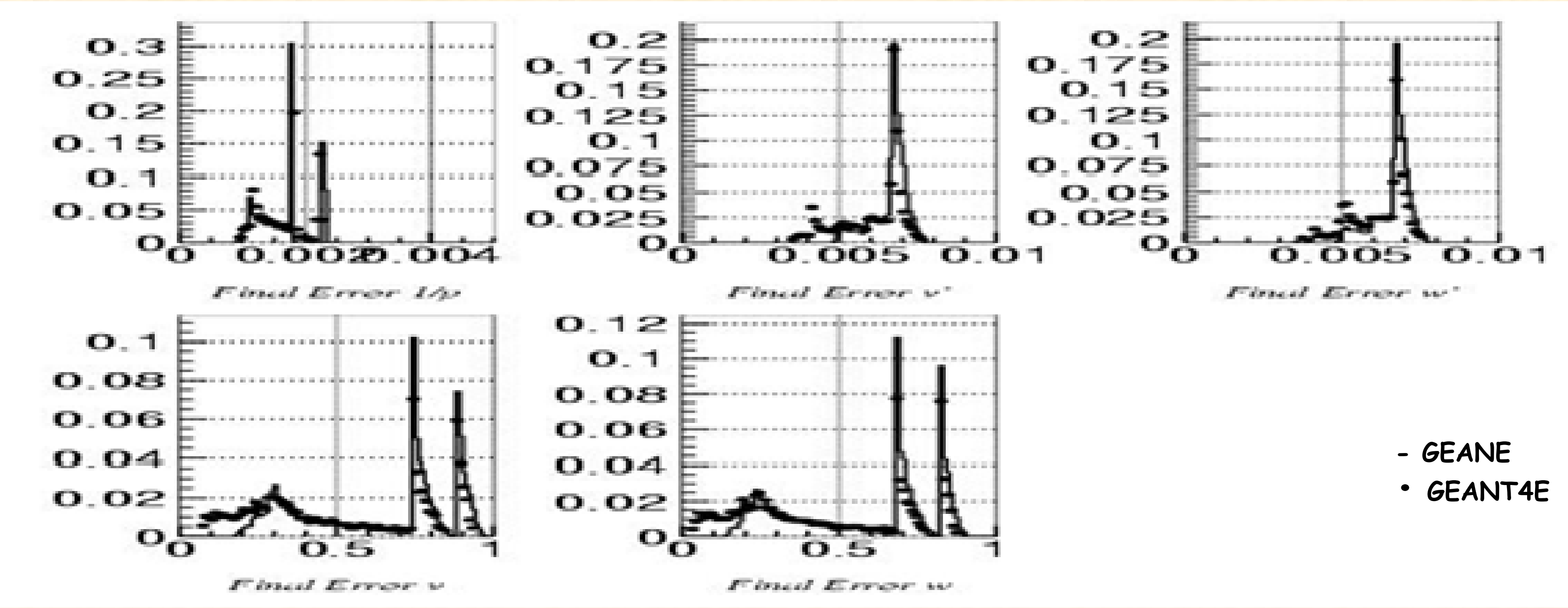


Comparison with GEANE

10000 mu+: 5-100 GeV, along X ± 10 degrees:

- **Trajectory errors (if target is reached)**

Error(0,0) Error(1,1) Error(2,2)



Error(3,3) Error(4,4)

Timing GEANE vs GEANT4E

10k mu+ 20 GeV cross all the detector (time in msec/evt CPU: Athlon 1 GHz)

- Same number of steps in GEANT3 and GEANT4

GEANT3		GEANT4	
GEANT3	0.39	GEANT4	1.22
GEANE: Forward or backward	**0.45**	**GEANT4E: Forward or backward**	**1.65**
GEANE: no error Forward or backward	0.28	GEANT4E: no error Forward or backward	1.30

- ➢ GEANT4 is 2.5 times slower than GEANT3
- ➢ GEANT4E is 3.5 times slower than GEANE
- ➢ Most of the time is taken by GEANT4 field propagation
- ➢ Error propagation is ~1/3 of total time

☺ Results have been checked by profiling

Geant 4

Precision GEANT3 vs GEANT4

Ciemat

Tracking in GEANT4 has been improved w.r.t. GEANT3 to make it more precise

Make a simple exercise: track the same muons in the whole CMS

- Once with detector full: all the volumes
- Once with detector empty: only the outer volume

- GEANT4 gives = results than with full geometry
- GEANT3 shows big differences

Geant 4

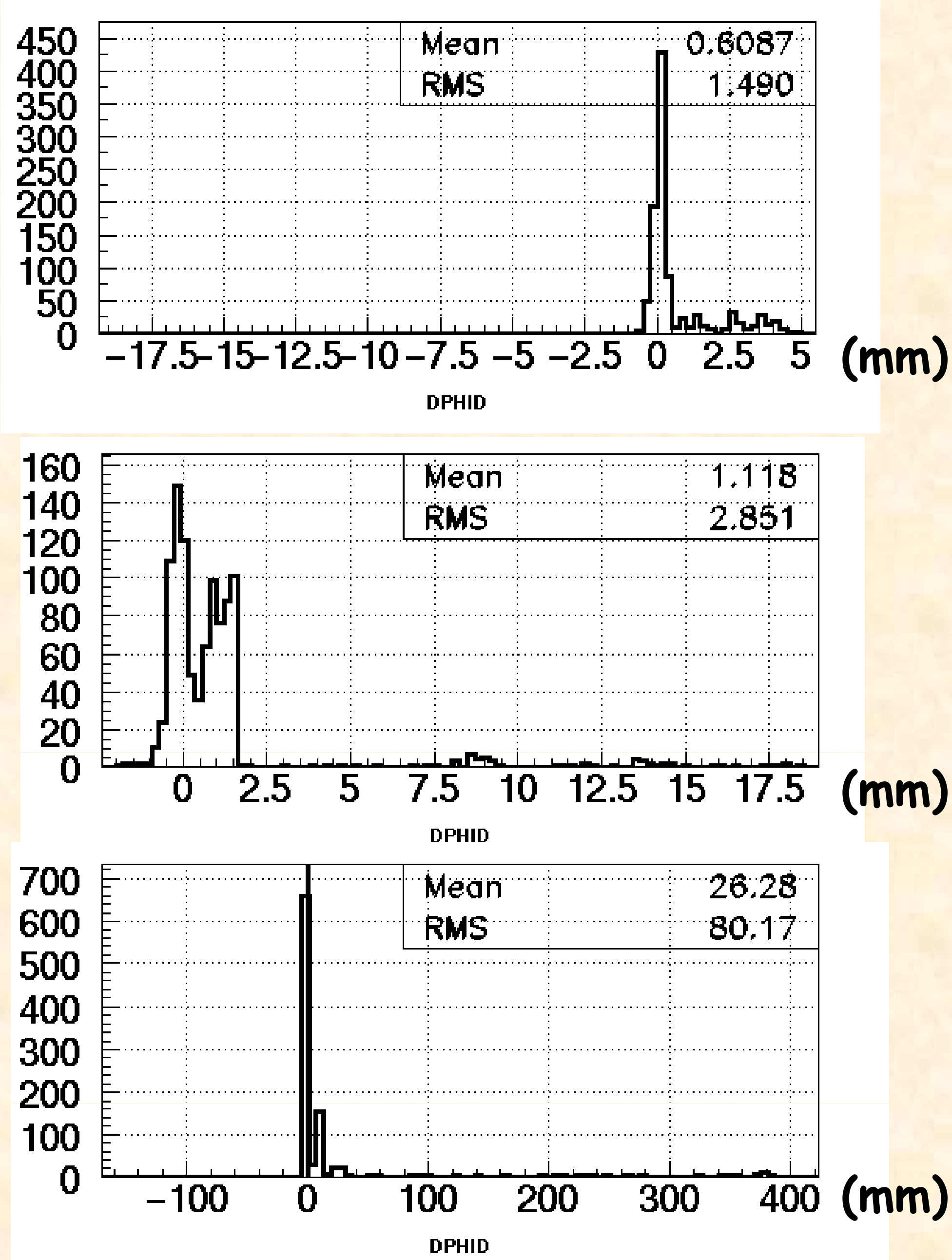
GEANT3: full geometry - no geometry

Ciemat

P = 500 GeV

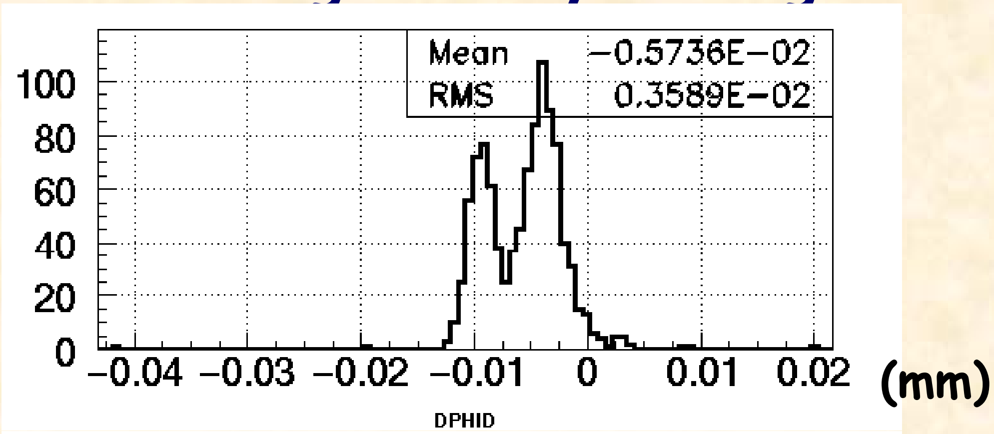
P = 50 GeV

P = 5 GeV

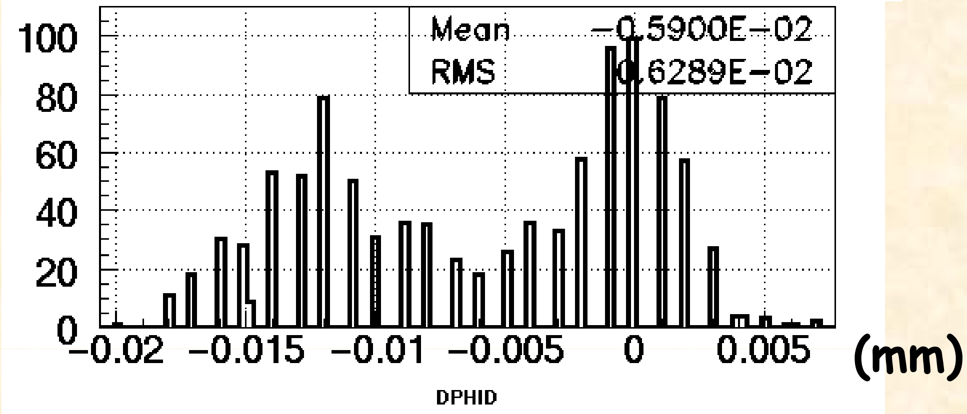


GEANT4: full geometry - no geometry

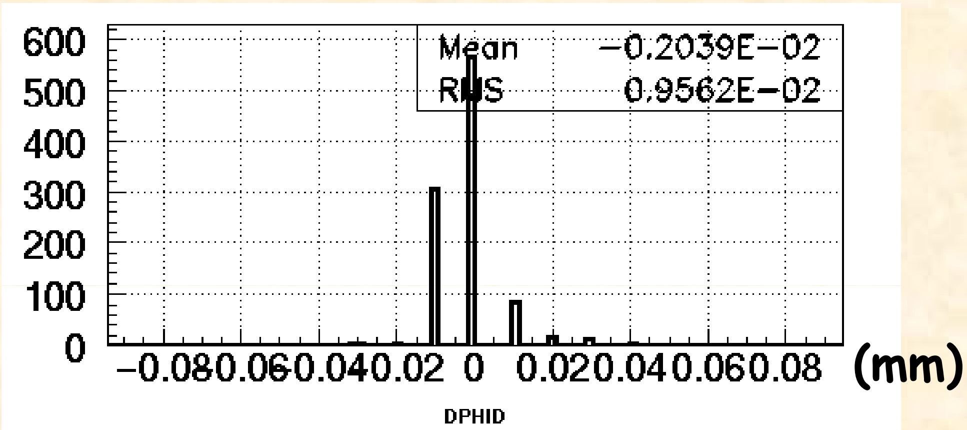
P = 500 GeV



P = 50 GeV



P = 5 GeV



GEANTE example in GEANT4 (2)

The type of target can be chosen with the enviromental variable **G4ERROR_TARGET:**

- PLANE_SURFACE : use a G4ErrorPlaneSurfaceTarget perpendicular to X at (2241. mm, 0, 0)
- CYL_SURFACE : use a G4ErrorCylSurfaceTarget parallel to Z of radius 2220mm
- VOLUME : use a G4ErrorGeomVolumeTarget with volume name "MUON"
- TRKLEN : use a G4ErrorTrackLengthTarget with track length 2230 mm

The user may also choose if the propagation is done forwards (the natural way, loosing energy) or backwards (in opposite direction, gaining energy), with the enviromental variable **G4ERROR_MODE**:

- FORWARDS : propagate in the forward direction
- BACKWARDS : propagate in the backward direction

GEANTE example in GEANT4 (3)

There are two modes of propagation, that can be chosen with the enviromental variable **G4ERROR_PROP**:

- UNTIL_TARGET : propagate until target, all steps in one go
- STEP_BY_STEP propagate until target, returning control to the user at each step

Tuning GEANT4E

- The problem is how to make it faster keeping the desired precision

Some ideas:

- Tune the step length to your desired precision
  - Define a fixed step length
  - Define the allowed variation in magnetic field
  - Define the allowed proportion of energy loss
- Tune propagation in magnetic field
  - Choose G4IntegratorStepper
  - Choose precision parameters
- Simplify geometry
  - Probably you do not need so much precision for reconstruction as for simulation

Future developments

- **Correction of multiple scattering fluctuations for low energy particles in gas detectors (see Andrea Fontana's talk)**

http://indico.cern.ch/getFile.py/access?contribId=37&resId=0&materialId=slides&confId=13356

GEANT4e:

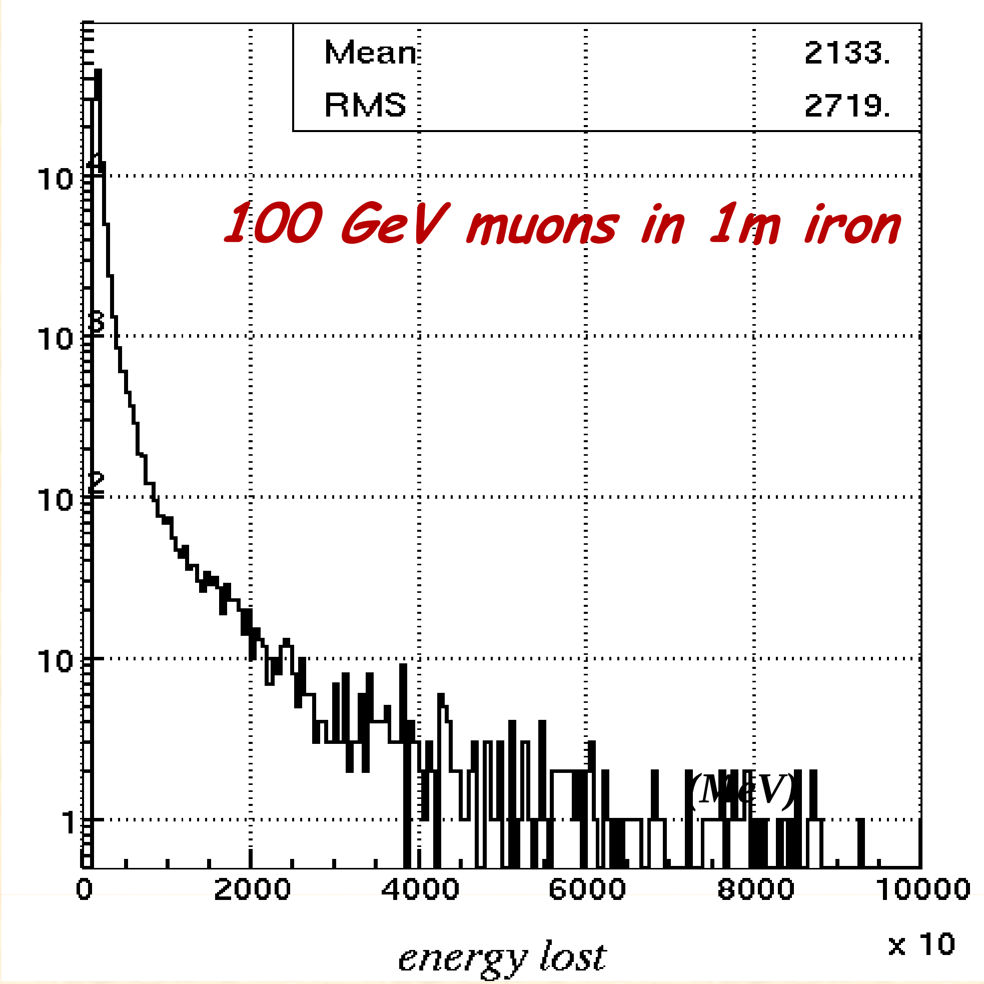
$$< \theta_p^2 >= \frac{184.96 \cdot 10^{-6}}{p^2} \frac{d}{\beta^2 X_0} .$$

R.Fruehwith and M.Regler, NIM A456(2001)369

$$< \theta_p^2 >= \frac{225 \cdot 10^{-6}}{p^2} \frac{d}{\beta^2 X_s} , \qquad X_s = X_0 \frac{Z+1}{Z} \frac{\ln(287\, Z^{-1/2})}{\ln(159\, Z^{-1/3})}$$

Future developments

- **Energy loss straggling**
  - We use in GEANT4e the mean energy loss
  - Energy loss is not gaussian!
  - Probably the events that have very big energy loss will never be reconstructed
- Why not rejecting those events and use the mean of the truncated?
  - You should tune it for your application



- Under discussion with V.Ivantchenko

Future developments

- **Optimize the error propagation**
  - Try different matrix class (now CLHEP)

- **Propagate error with Runge-Kutta equations to make bigger steps**
  - Each step may take longer…

- **Being checked in a real detector reconstruction (CMS)**

Geant 4 | Ciemat

Summary and plans

- First prototype of GEANT4E is ready with similar functionality as GEANE
- Already released in geant4.9.0
  - Including example and detailed documentation
- Many optimisation options available

<u>Next steps:</u>

- Refine multiple scattering and energy loss errors
- Try different optimisation options
- Check in a real detector reconstruction (CMS)
  - On progress..