GEANT4: Error propagation for track reconstruction inside the GEANT4 framework

P.Arce, CIEMAT, Madrid, Spain

Abstract

GEANT4e is a package of the GEANT4 Toolkit [1] that allows to propagate a track with its error parameters. It uses the standard GEANT4 code to propagate the track and for the track propagation it makes an helix approximation (with the step controlled by the user) using the same equations as GEANT3/GEANE. We present here a first working prototype of the GEANT4e package and compare its results and performance to the GEANE package.

DESCRIPTION OF GEANT4E

Track reconstruction software needs often to match signals in two detector parts, for what it needs to propagate tracks from one detector part to another and compare with real measurement there. To make then the average between the prediction and the real measurement it is necessary to know the track parameter errors.

Many experiments have used in the past GEANE (based on GEANT3) or their 'ad hoc' solution for this task. GEANT4e provides this functionality for the reconstruction software in the context of GEANT4

MAIN COMPONENTS

We describe here briefly the main components of GEANT4e.

G4eTrajState

This class serves to define the initial track parameters in a given point of the trajectory:

- Particle type
- Position
- Momentum
- Track errors (5x5 HepSymMatrix)
- Initial surface where parameters are defined

Two different trajectory states are available: free trajectory state and trajectory state on a surface

G4eTarget

This is the class to define up to where the propagation must be done: the target. Several types of targets are supported:

- G4eTargetSurface: Track is propagated until the surface is reached. The surface is not part of GEANT4 geometry Several types can be defined
  - G4eTargetPlaneSurface: infinite plane
  - G4eTargetCylindrycalSurface: infinite length cylindrical surface
  - ....
- G4eTargetTrackLength: Track is propagated until a certain track length is reached
- G4eTargetVolumeG4: Track is propagated until the surface of a GEANT4 volume. It can be selected if the track is stopped only when track enters or track exits or both. User can choose if volume refers to one or many G4LogicalVolumes, G4VPhysicalVolumes or G4VTouchables, with a simple syntax:

```
MuonCell                    G4LogicalVolume
MuonCell#1                 G4VPhysicalVolume
MuonChamber#3/MuonCell#2   G4VTouchable
```

G4ePropagator

This is the class that creates a G4Track from the information given in the G4eTrajState, invokes G4SteppingManager to propagate one step and then propagates the track errors for this step. The propagation stops when G4Track stops or when the target is reached. If the defined target is not reached it returns an error.

User can choose two ways of propagation:

- Propagate until target is reached
- Propagate step by step and return control to the user after each step

G4ePhysicsList

G4ePhysicsList takes care of implementing the special physics needed by reconstruction software, that only needs the average trajectory followed by the particle

- No secondaries allowed
- No random fluctuations for energy loss
- No hadronic processes
- Huge cuts by default (the user can change them with standard GEANT4 methods)

- Negative energy loss when propagation is backwards

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

Track error propagation

It is based on the equations of the European Muon Collaboration (same as GEANE) [2].
It takes into account

- Error from curved trajectory in magnetic field
- Error from multiple scattering
- Error from ionisation

The formulas assume propagation along an helix, and therefore it is necessary to make small steps to assure magnetic field constantness and not too big energy loss, what makes teh propagation slower. Another approach to be studied would be to propagate the error together with the solving of the Runge-Kutta equations. This would probably be slower per step but might not need so many steps.

GEANT4E COMPARISON WITH GEANE

We have implemented a simple detector in GEANT4e and GEANE to allow detailed benchmarking. The setup is described in figure 1. The detector is inmersed in a magnetic field 0.1 Tesla. 10000 muons of energy between 5 and 100 GeV along the X axis ±10 degree are propagated from the origin along all detectors until a plane surface, and then from the end point each muon is propagated backwards. Several variables are compared GEANE vs GEANT4E: energy lost, deviation in position and deviation in angle for forward, backward and forward+backward tracking (in this case the above variables should ideally be 0), and trajectory errors for forward and backward.

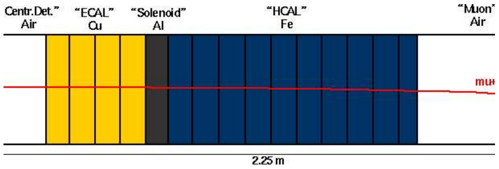

Figure 1: Detector setup

Figure 2, 3 and 4 show a very good agreement between both codes for the energy lost and deviation in position and angle. Figure 5 shows that also the agreement in the error calculated is quite good, considering the different internal precision.

Time comparison

We show in table 1 the time taken to propagate 10000 mu+ of 20 GeV that cross all the detector. The time is given in msec/event on an Athlon 1 GHz CPU. We guarantee that the same number of steps are taken in GEANT3 and GEANT4.

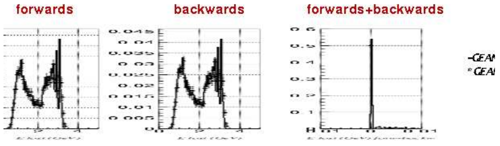

Figure 2: Comparison between GEANT4e and GEANE energy lost for forward, backward and forward+backward propagation

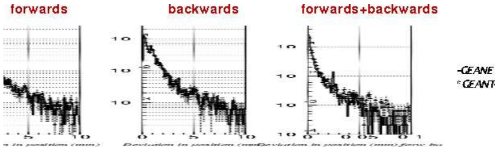

Figure 3: Comparison between GEANT4e and GEANE deviation in position for forward, backward and forward+backward propagation

We can see from this table that GEANT4 is 2.5 times slower than GEANT3, while GEANT4E is 3.5 times slower than GEANE. We have checked by profiling that most of the time is taken by GEANT4 field propagation, while error propagation is about one third of the total time.

The problem is indeed how to make it faster keeping the desired precision. For this there are several things that the user should try for her/his setup:

- 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 IntegratorStepper
- Choose precision parameters
- Simplify geometry

For the part of the GEANT4e developers we should also investigate some optimization options:

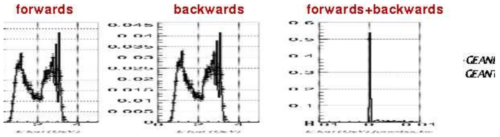

Figure 4: Comparison between GEANT4e and GEANE deviation in angle for forward, backward and forward+backward propagation

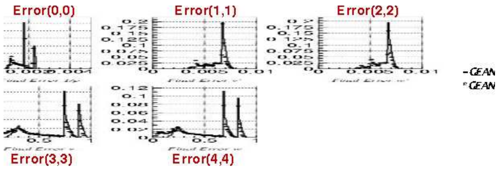

Figure 5: Comparison between GEANT4e and GEANE error matrix elements for forward, backward and forward+backward propagation

GEANT3	0.39	GEANT4	1.22
GEANE: forward or backward	0.45	GEANT4e: Forward or backward	1.65
GEANE: forward or backward (no error)	0.28	GEANT4E: forward or backward (no error)	1.30

Table 1: Time comparison between GEANT4e and GEANE in simple detector

- Optimize the track propagation
- Optimize the error propagation (try different matrix class)
- Propagate error with Runge-Kutta equations to make bigger steps

REFERENCES

[1] http://www.cern.ch/geant4

[2] EMC internal reports EMCSW/80/15, EMCSW/80/39, EMCSW/81/18