

MC simulation of test beam events

Fabrizio Salvatore (RHUL)



Simulation of Desy tb events - I

- n First attempt to simulate MC events in test beam mode (Desy06 test beam) at three different institutions: RHUL, B'ham, Manc.
- n Use same version of Mokka (v06-01) and same version of packages against which Mokka is linked:
 - ⊠ Geant4 8.0.p01 ⊠ LCIO v01-07
 - ⊠ CLHEP 2.0.2.3 ⊠ Gear 00-03
 - ⊠ GSL 1.7
- n Note: use same versions of g++ compiler (gcc 3.2.3), but different updates
 - ⊠ RHUL: 3.2.3-56; Bham: 3.2.3-49; Man: 3.2.3-49/56



Simulation of Desy tb events - II

n Small modification needed in Geant4 and in Mokka/Kernel in order to define initial seeds used by the random numbers generator (David Bailey):

n G4: in source/intercoms/include/G4UIcommand.hh

```
G4int Dolt(G4String parameterList); → virtual G4int Dolt(G4String parameterList);
```

n Mokka: Added class 'G4UlcmdWithVariableIntList' defined in Kernel/src/G4UlcmdWithVariableIntList.cc

o In src/ControlMessenger.cc:

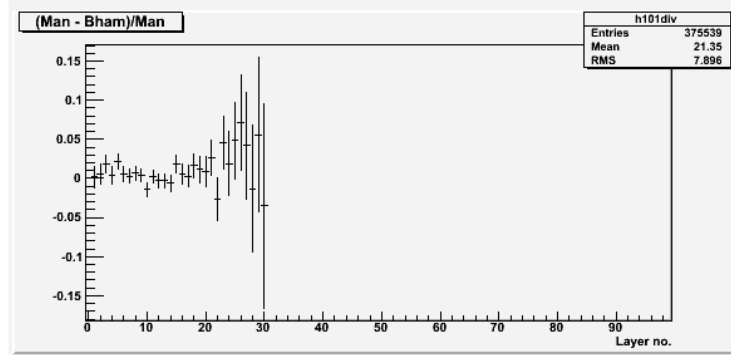
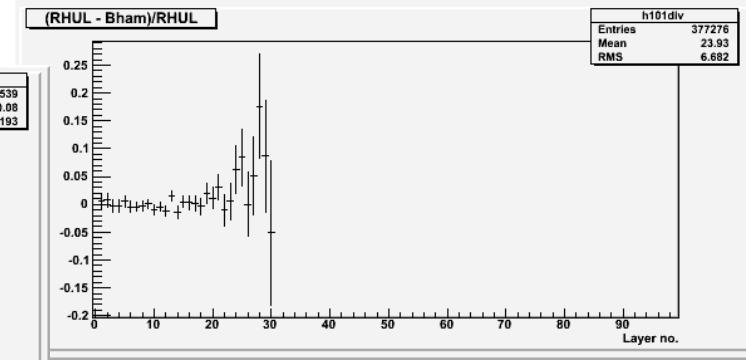
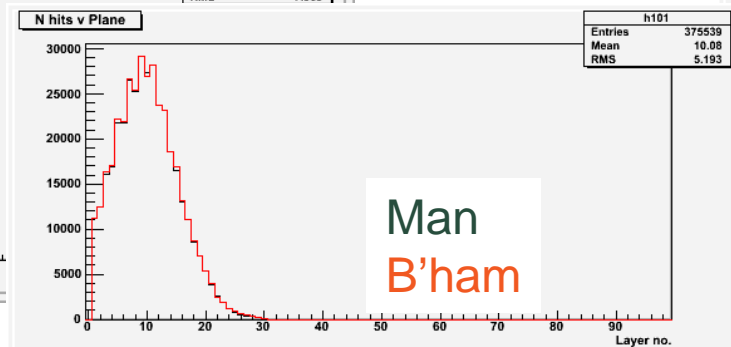
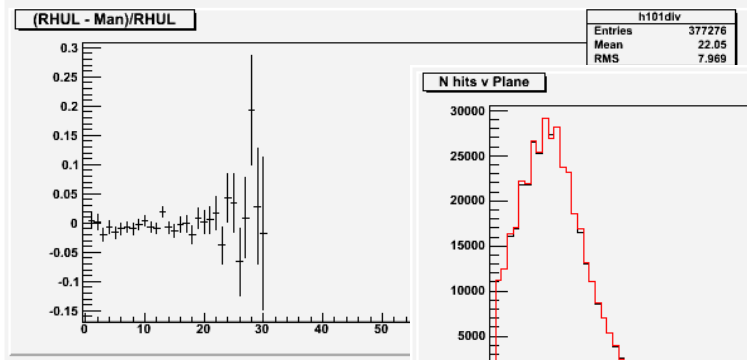
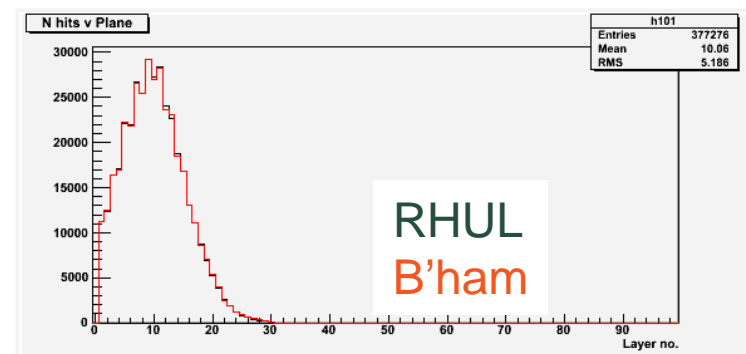
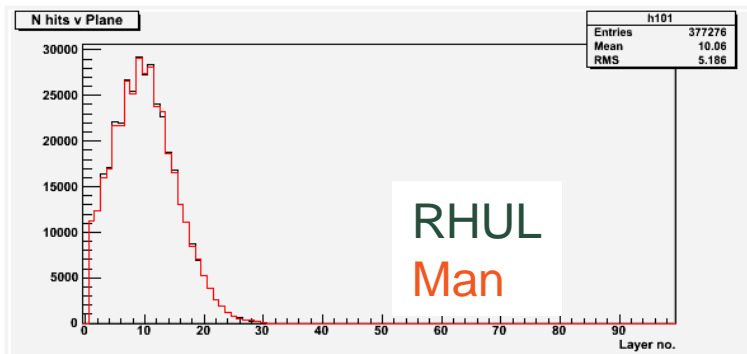
```
//Random number commands  
  
_randomSeedsCommand = new  
G4UlcmdWithVariableIntList("/Mokka/init/RandomNumberSeeds",this);  
  
_randomSeedsCommand->SetGuidance("List of integer seeds to be  
passed to the random number generator.");  
  
_randomSeedsCommand->AvailableForStates(G4State_PreInit);
```



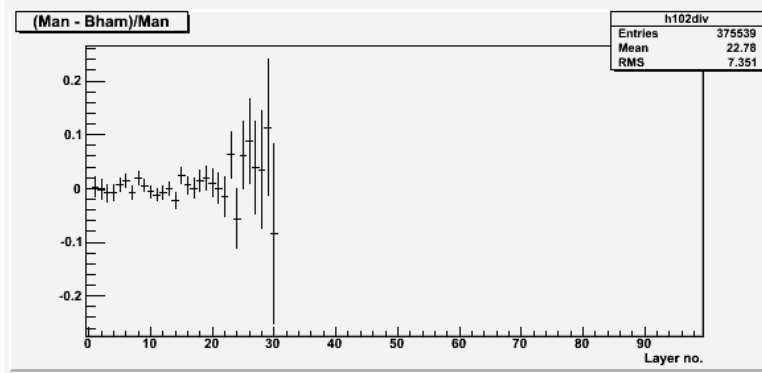
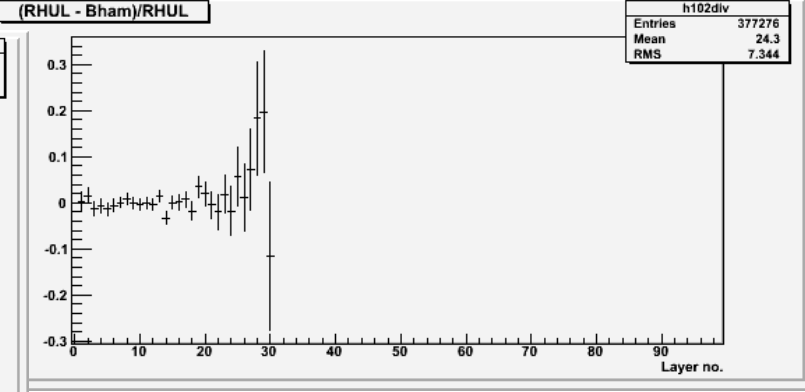
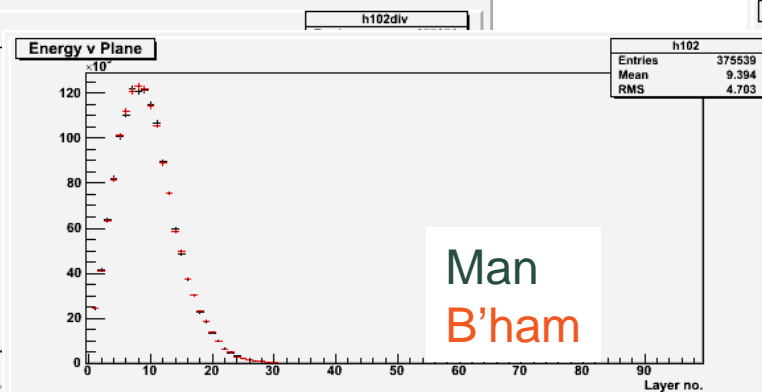
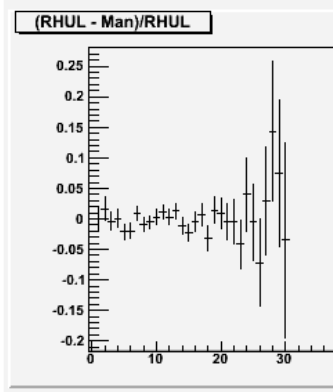
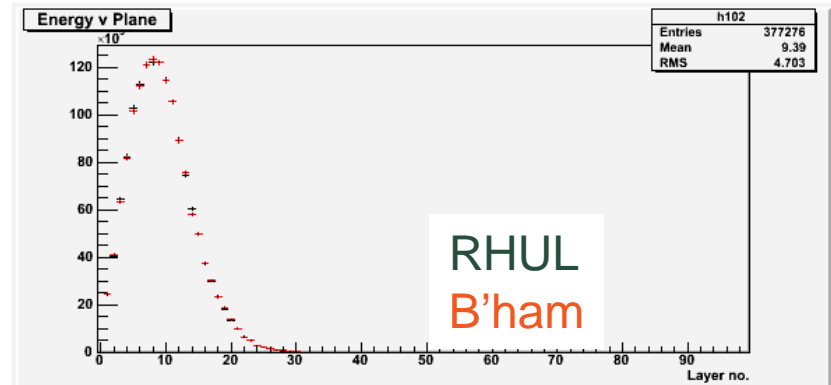
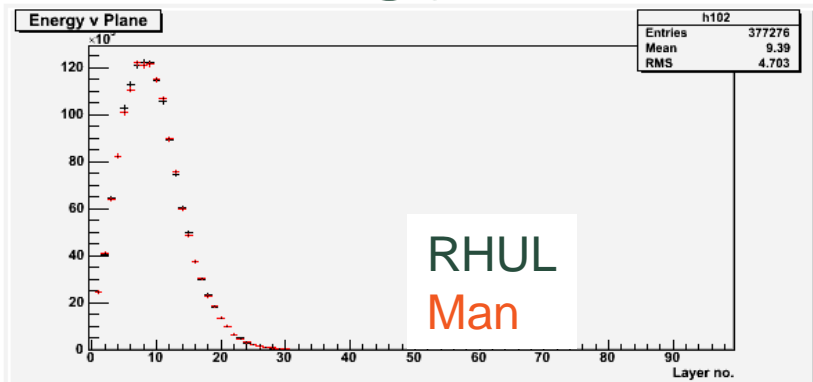
Simulation of Desy tb events - III

- n Generate 10K 1GeV electron events using TBDesy0506, at normal incidence
 - o Use same initial random numbers seeds
- n Use caliceMarlin to extract distribution of variables related to ECAL and tracking to compare the events (Icio output) generated in the three different sites (thanks to Michele and David W. for the help on setting up caliceMarlin)
- n Plot variable distribution and ratio of
$$\frac{\text{variable_at_site_i} - \text{variable_at_site_j}}{\text{variable_at_site_i}}$$

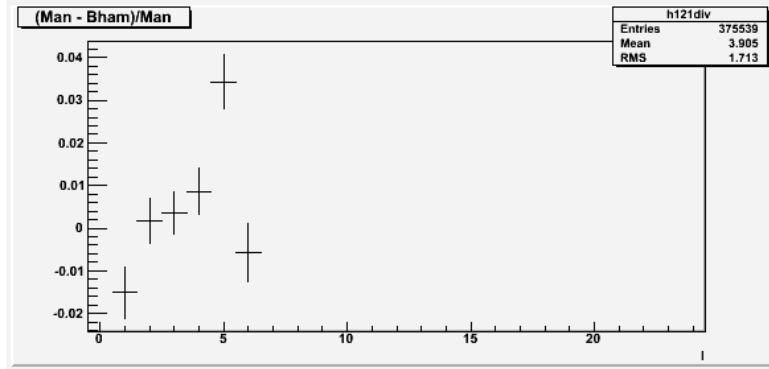
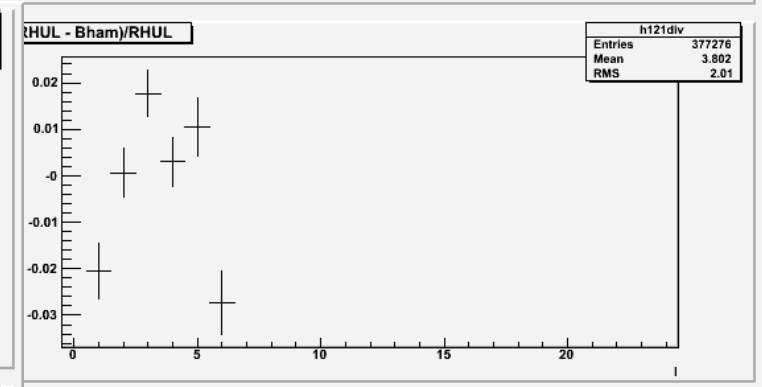
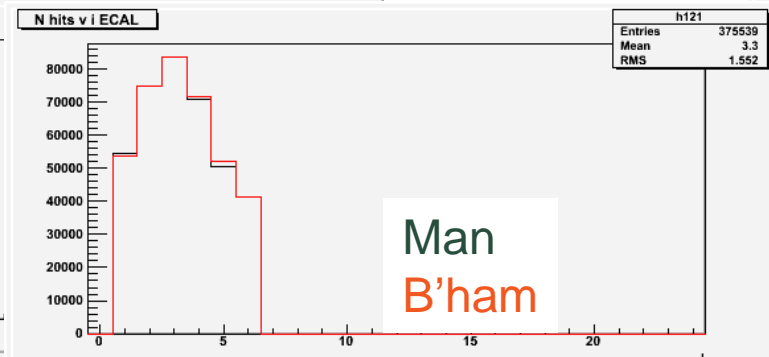
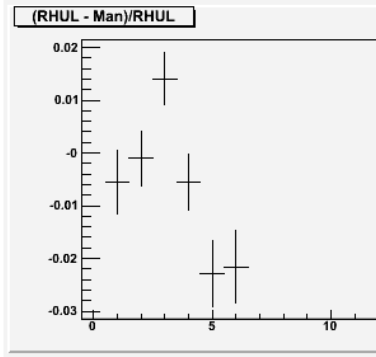
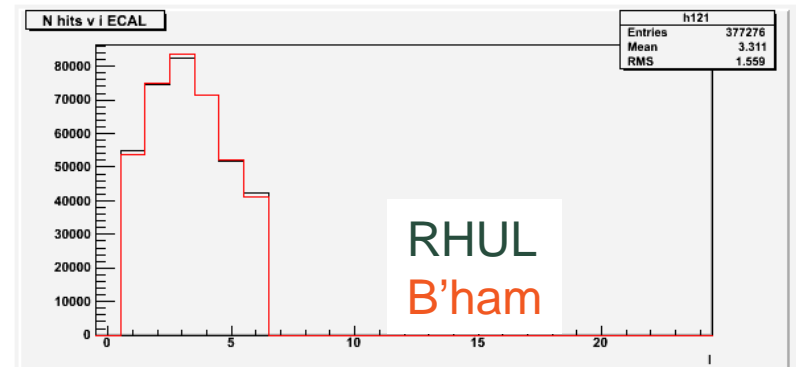
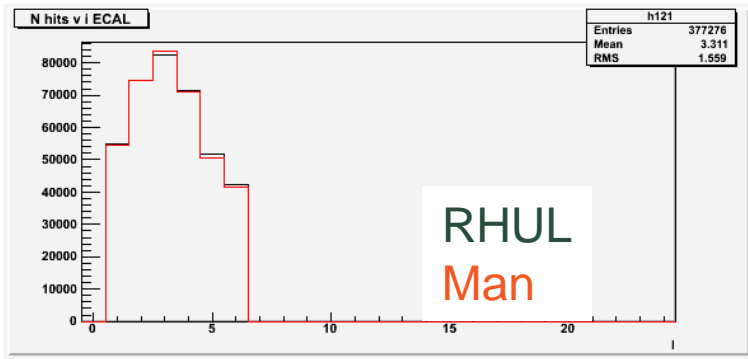
N(hits) vs Plane



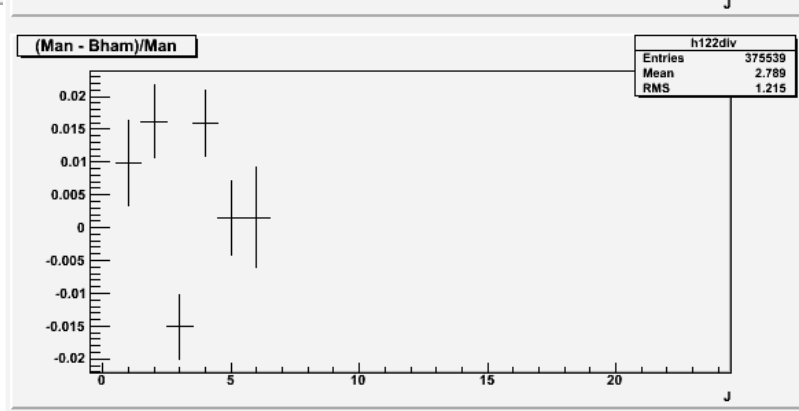
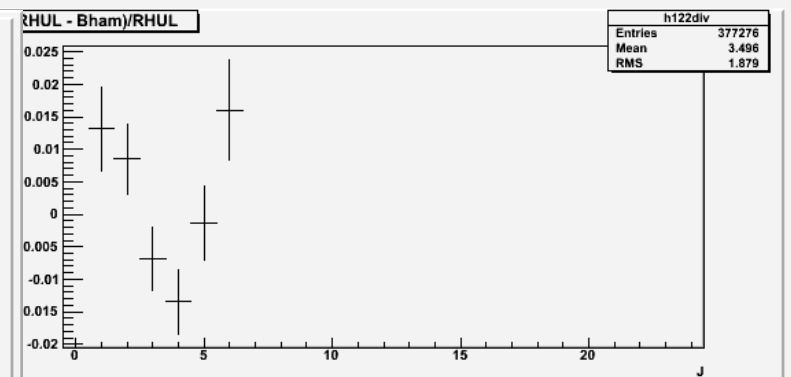
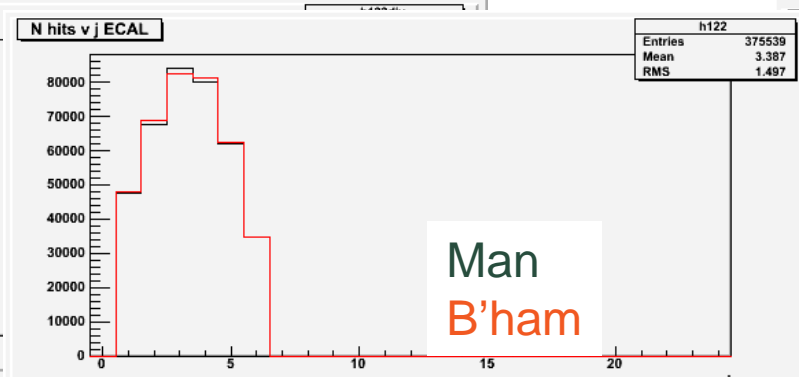
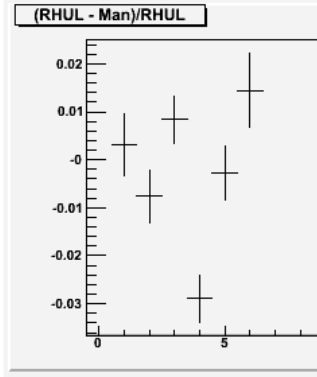
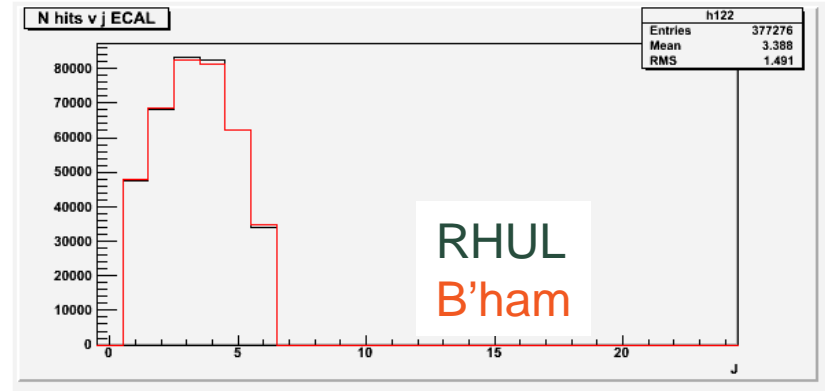
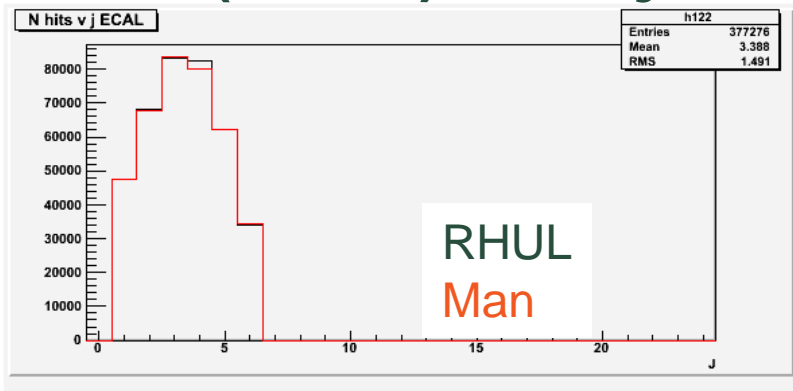
Energy vs Plane



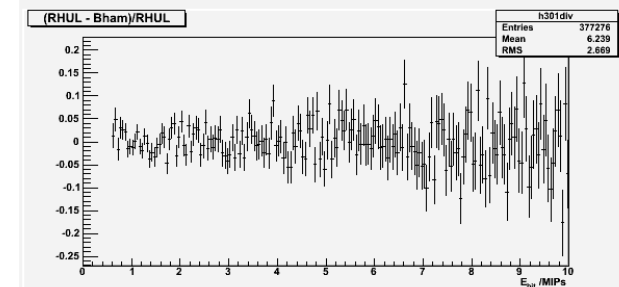
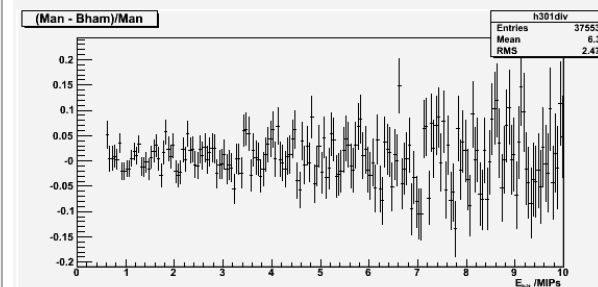
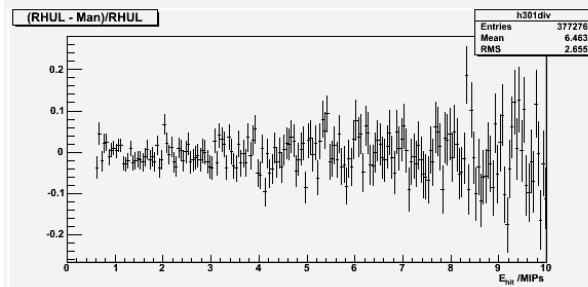
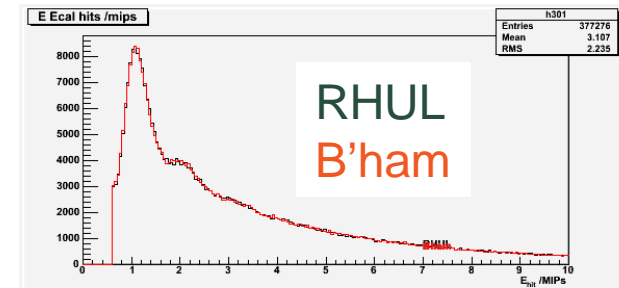
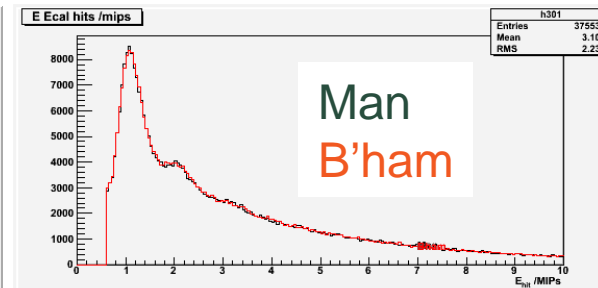
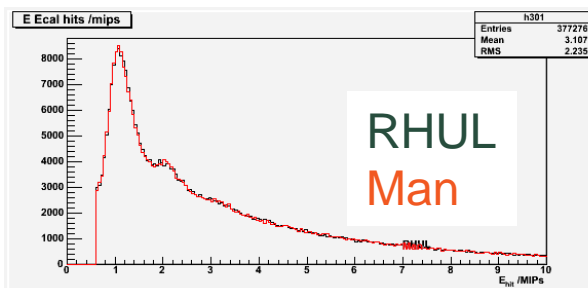
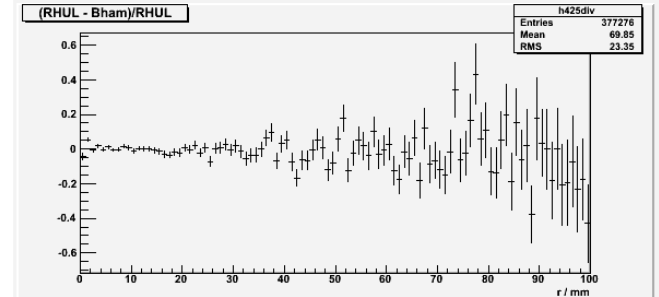
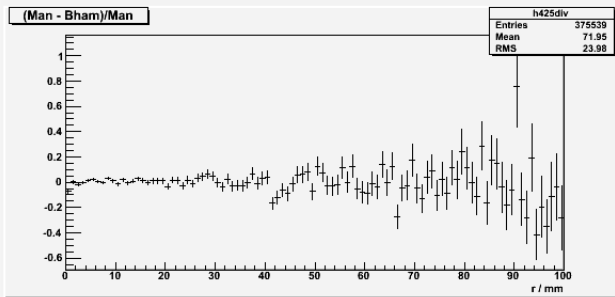
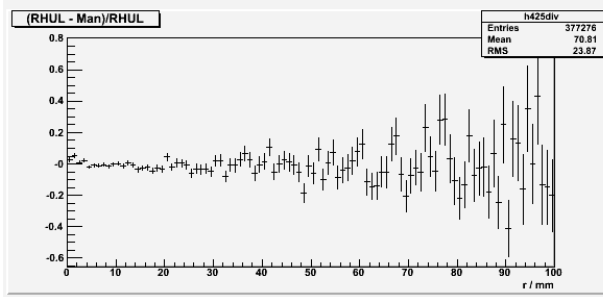
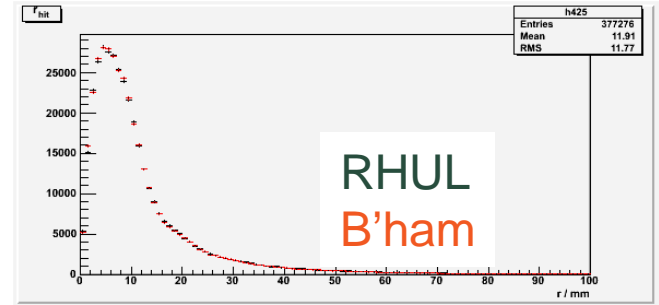
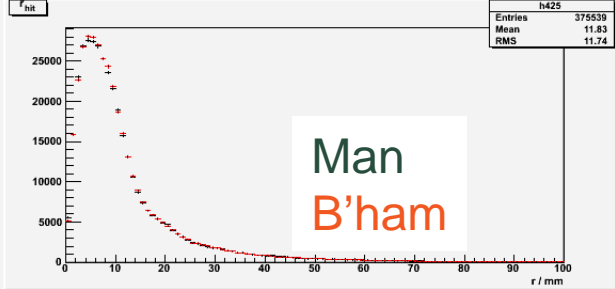
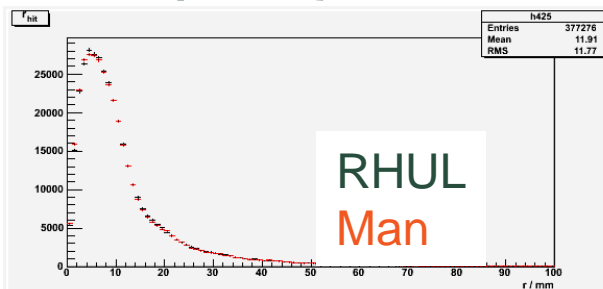
N(hits) vs i ECAL



N(hits) vs j ECAL



r(hit) and E(hit/mips)





Summary - I

- n Full comparison plots available at:
<http://www.pp.rhul.ac.uk/~calice/fab/Mokka/test/>
- n Although same versions of all programs and same initial seeds for the random numbers generator are used, resulting MC events are not exactly the same (as one would expect)
 - α Differences in the distributions of several kinematic variables are almost all within statistical uncertainties
 - α No big systematic bias is clearly visible



Summary - II

- n Modified code to handle random number seeds not yet in official Mokka release
 - o Modification in Geant4 needs to be included in standard G4 release
- n Need to setup a strategy for future massive MC production
 - o Compile executable at one single site and distribute the executable to all institutes involved in MC production
 - o Use different executables and quote differences as systematic effect on results