CMS Simulation with VecGeom

Geant4 Collaboration Meeting
September 26, 2017

S. Banerjee

Introduction

- CMS is currently using Geant4.10.2.p02 in its simulation production version
- CMS plans to migrate to the new version of Geant4 (10.4) which will be released by the end of this year. The beta release of this version happened in June.
- It is possible to replace the native Geometry routines of Geant4 using a new set of classes (VecGeom). The 10.4.beta version integrates well with the VecGeom version (v.00.04.00) with almost all the shapes required by CMS coming from VecGeom
- It is worthwhile to examine its performance as well as its effect on physics distributions
- The very first step is to build a CMSSW version with the new Geant4 version with appropriate dependencies

Integration of Geant4+VecGeom

- There are additional external libraries used for this purpose
  - VecGeom package and for vector version the Vc
  - Step1: Build Vc, VecGeom and Geant4 version
    - Build Vc first
    - Use the same compiler, environment and some of the external packages used by CMSSW
      - CLHEP, XERCES-C, EXPAT-ROOT
    - Build VecGeom
      - in scalar mode for the scalar version
      - in vector mode for the vector version with dependence on Vc
    - Build Geant4 using the VecGeom (and Vc) - both shared and static libraries
  - Step2: Build CMSSW
    - Provide configuration files for the new external packages Vc, VecGeom
    - Modify the configuration files for Geant4 within CMSSW
      - There are several such files: the core, data, static, ….
      - Show dependence on VecGeom in the Geant4 build
    - Checkout the code which are affected by change in the tool
    - Inhibit the biglib option
      - Not useful for most CMS applications

Geant 4 Performance of the Scalar Version

- A few physics processes are tried for native Geant4 and Geant4 with VecGeom versions using the release 10.3.ref06
  - 3000 events for single particle events at 50 GeV
  - 500 events for minimum bias events at 14 TeV
  - 110 events for t-tbar events at 14 TeV

	RSS (Native) (GB)	CPU (VecGeom) (wrt Native)	RSS (VecGeom) (GB)
Muon (Barrel)	0.49	0.984	0.49
Muon (Endcap)	0.49	0.945	0.52
Pion (Barrel)	0.60	0.959	0.55
Pion (Endcap)	0.55	0.953	0.60
Elec (Barrel)	0.51	0.998	0.55
Elec (Endcap)	0.51	0.983	0.50
Minimum Bias	0.59	0.919	0.58
t-tbar	0.64	0.932	0.62

Performance of the Vector Build

- Repeat the same samples as done for the version using VecGeom in scalar mode

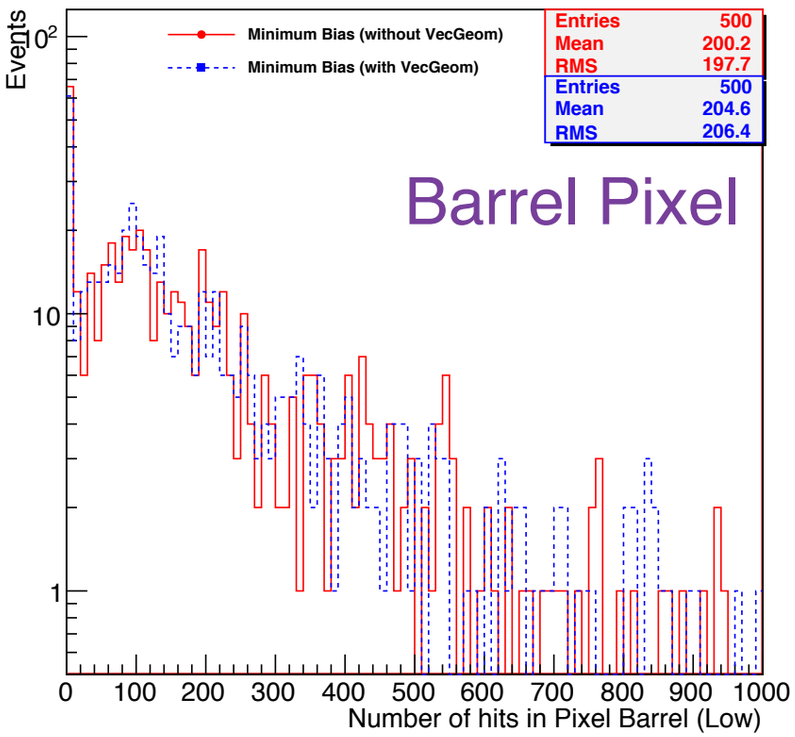
	RSS (Scalar) (GB)	CPU (Vector) (wrt Scalar)	RSS (Vector) (GB)
Muon (Barrel)	0.49	1.016	0.50 GB
Muon (Endcap)	0.52	1.019	0.51 GB
Pion (Barrel)	0.55	1.014	0.55 GB
Pion (Endcap)	0.60	1.028	0.54 GB
Elec (Barrel)	0.50	1.021	0.51 GB
Elec (Endcap)	0.50	1.011	0.51 GB
Minimum Bias	0.58	1.015	0.59 GB
t-tbar	0.62	1.033	0.61 GB

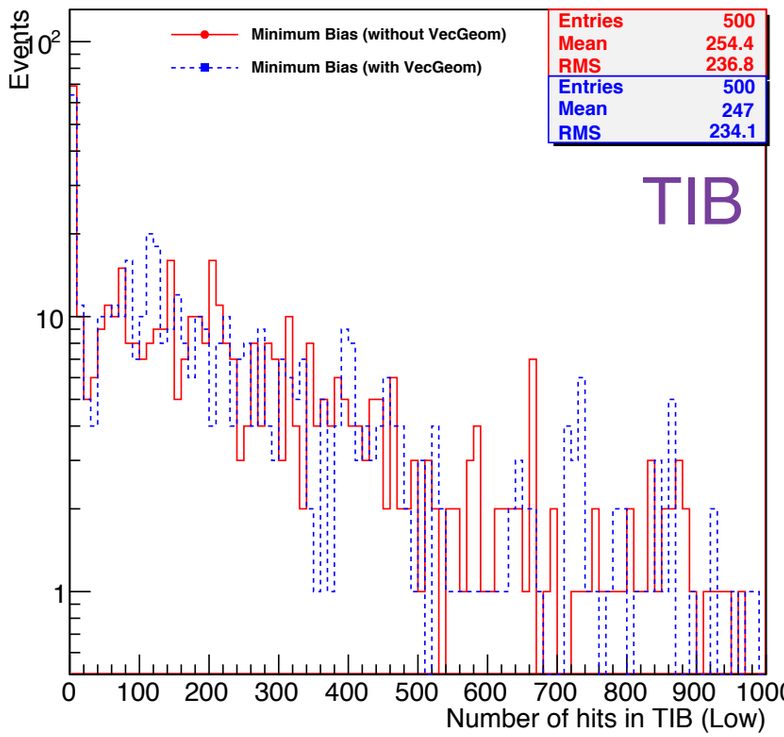
- The vectorized build is less performant than the scalar build
- Physics performance is identical to that of the scalar build

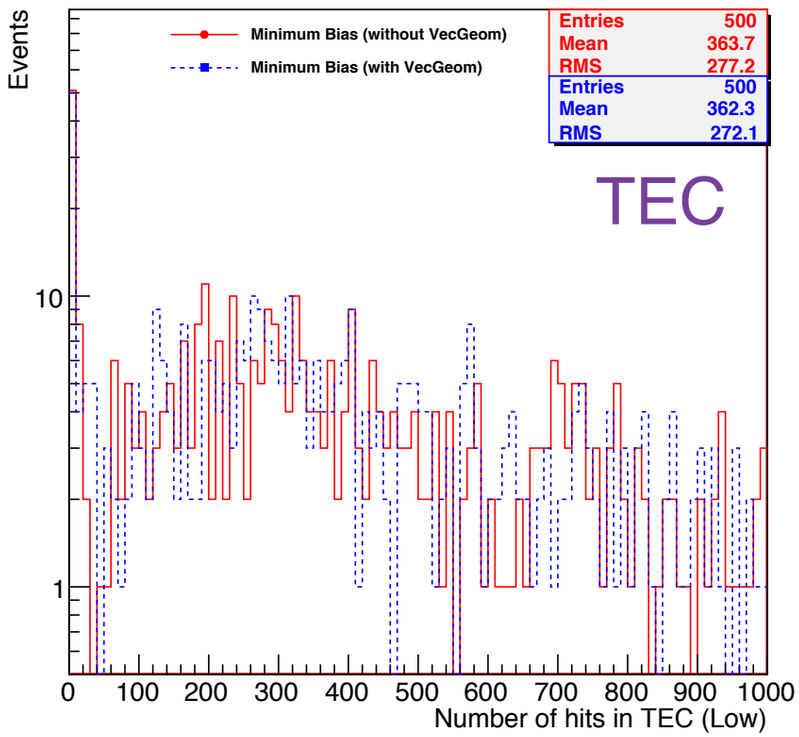
Quality of Simulation

- Compared some of the basic quantities in 500 minimum bias events from
  - version using native Geant4 geometry routines
  - version using Geant4 + VecGeom package

Number of hits in different Tracker layers

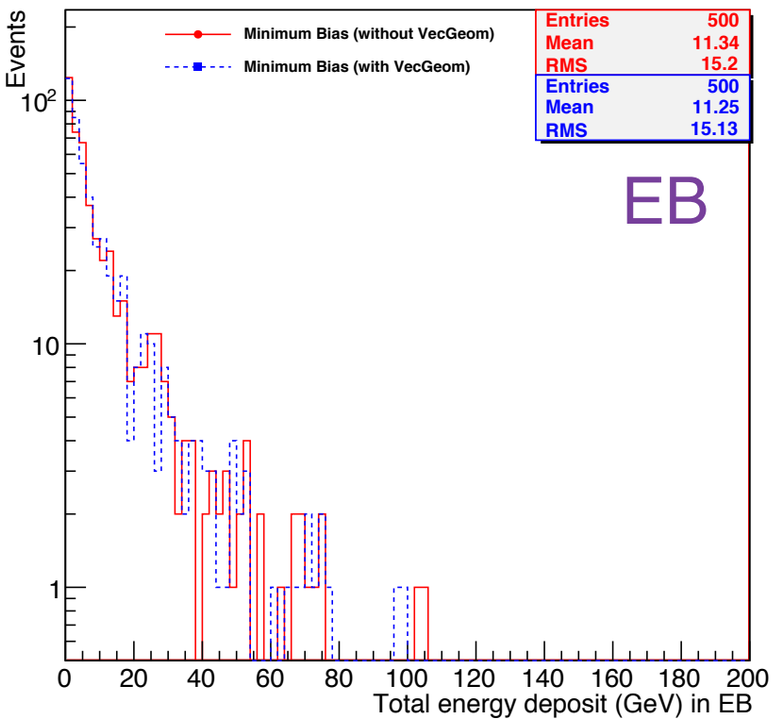


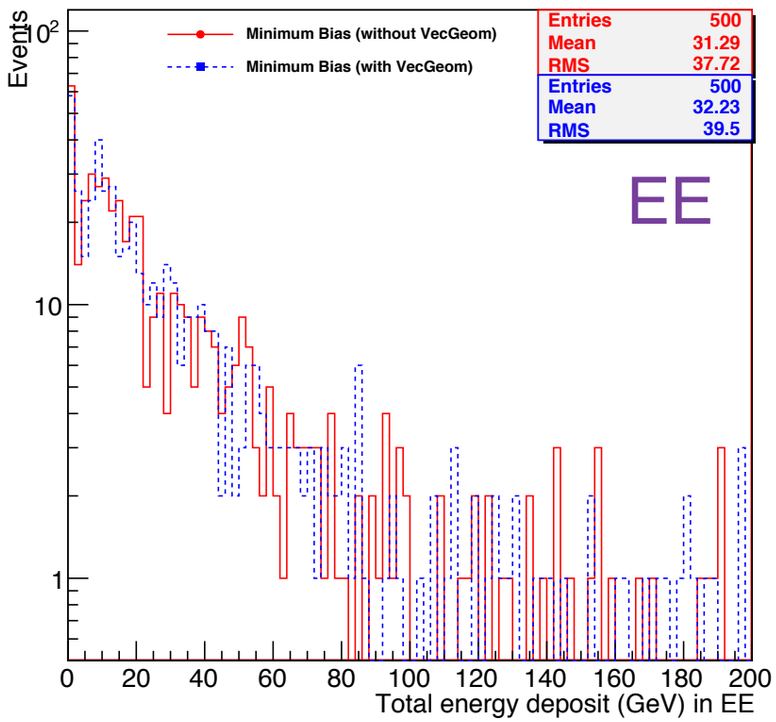


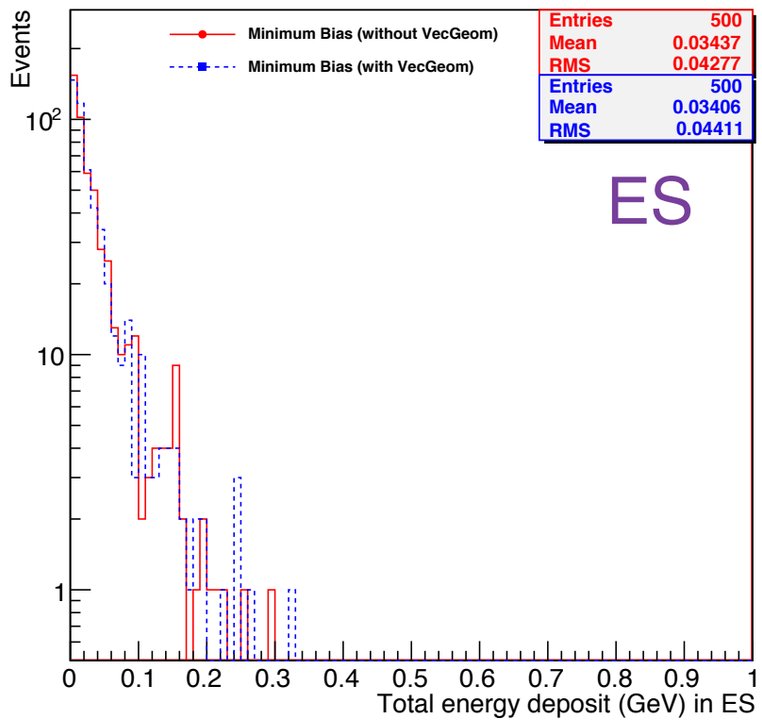


Energy and Timing in Calorimeter

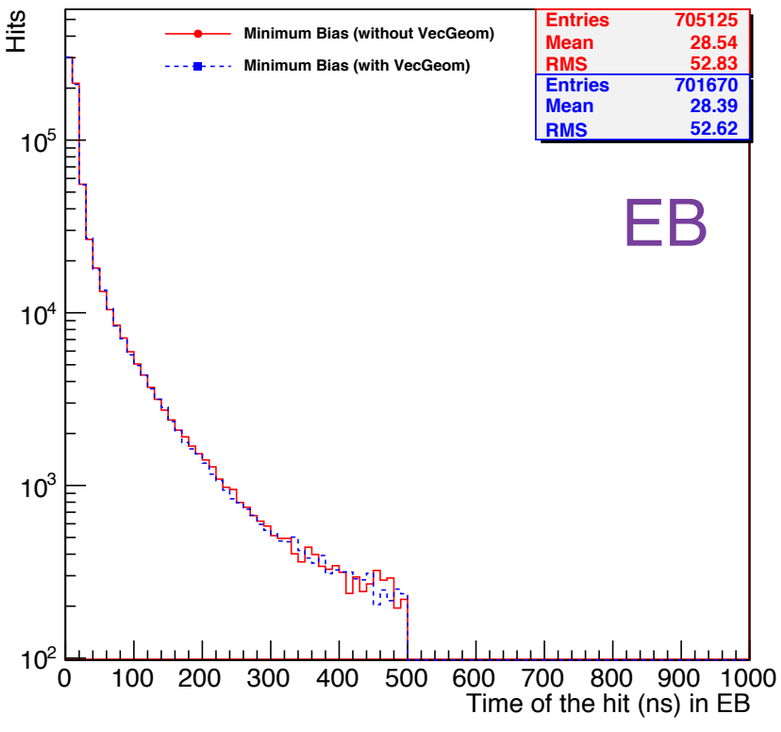
Total Energy Deposit

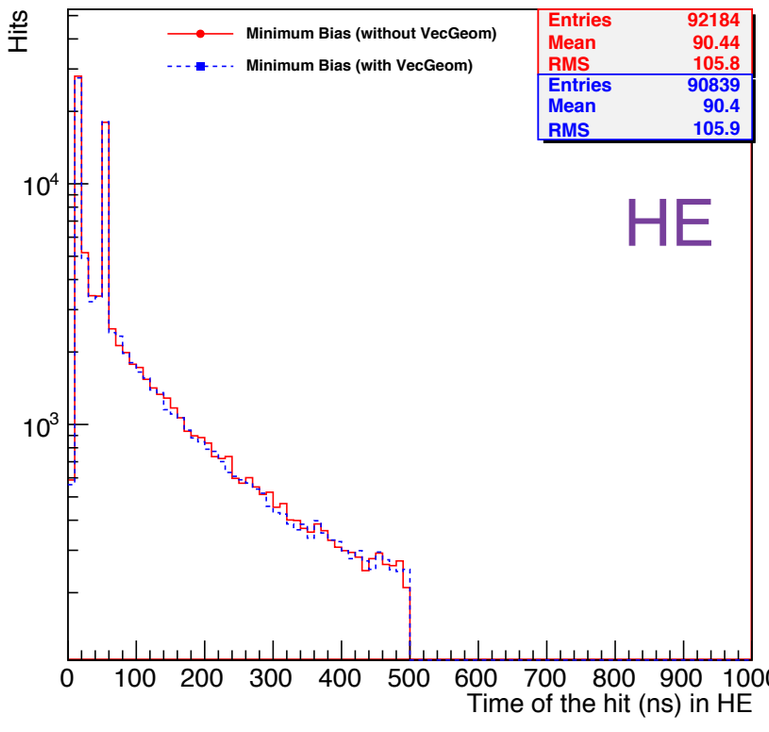


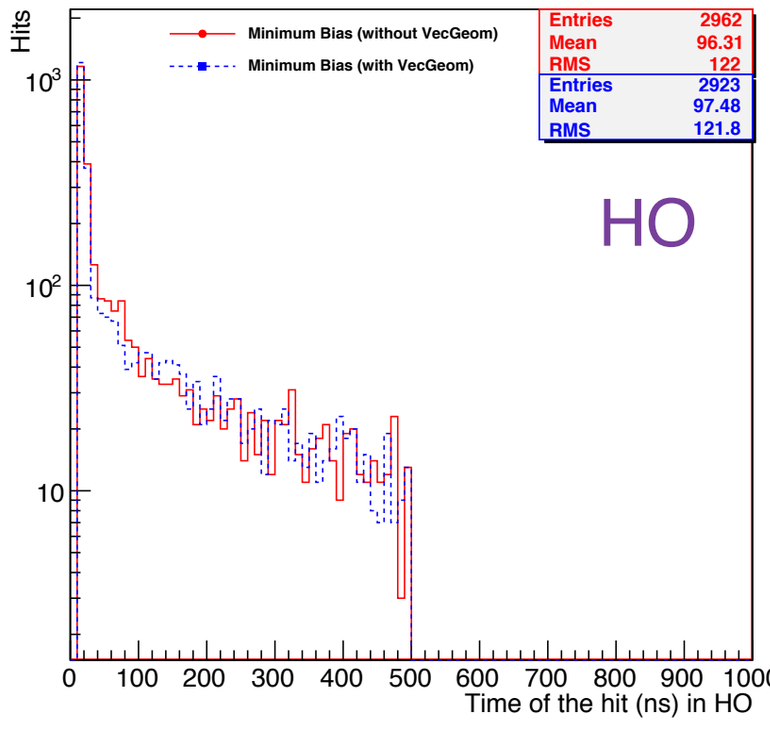




Hit Time

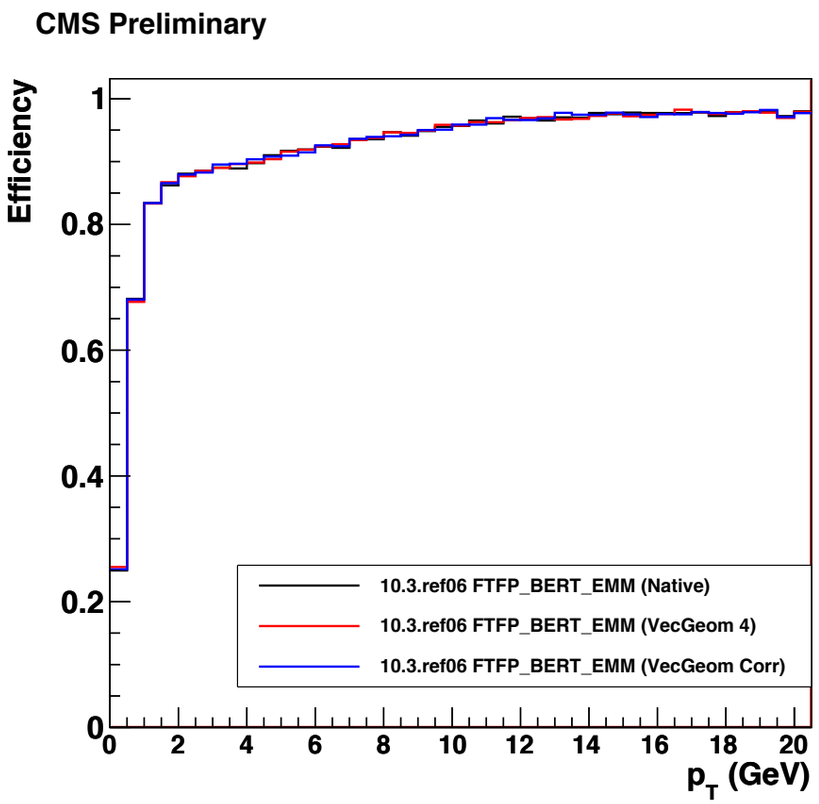


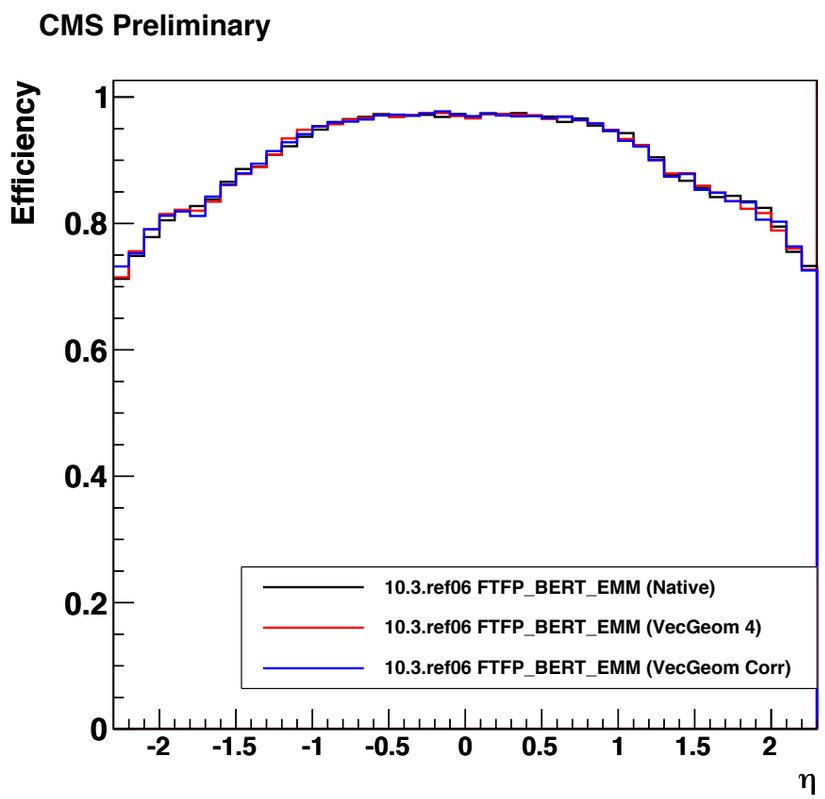


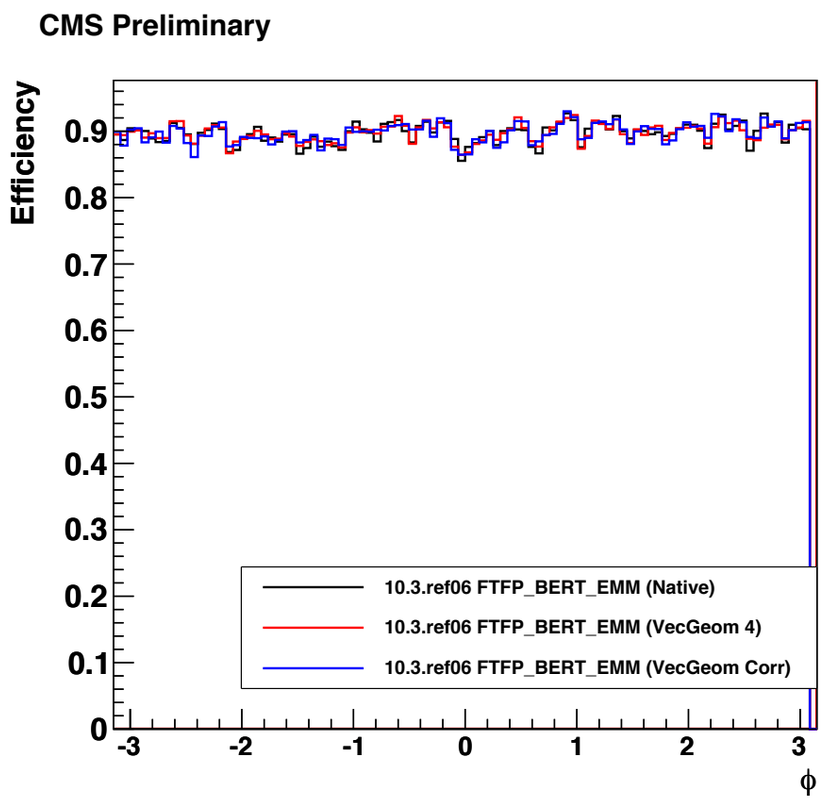


Track Reconstruction Efficiency

- Generate single particle events with a mixture of pions, kaons, protons and anti-protons
- Do a complete simulation going all the way up to reconstruction
- Study track reconstruction efficiency as a function of p, p_T, η and ϕ



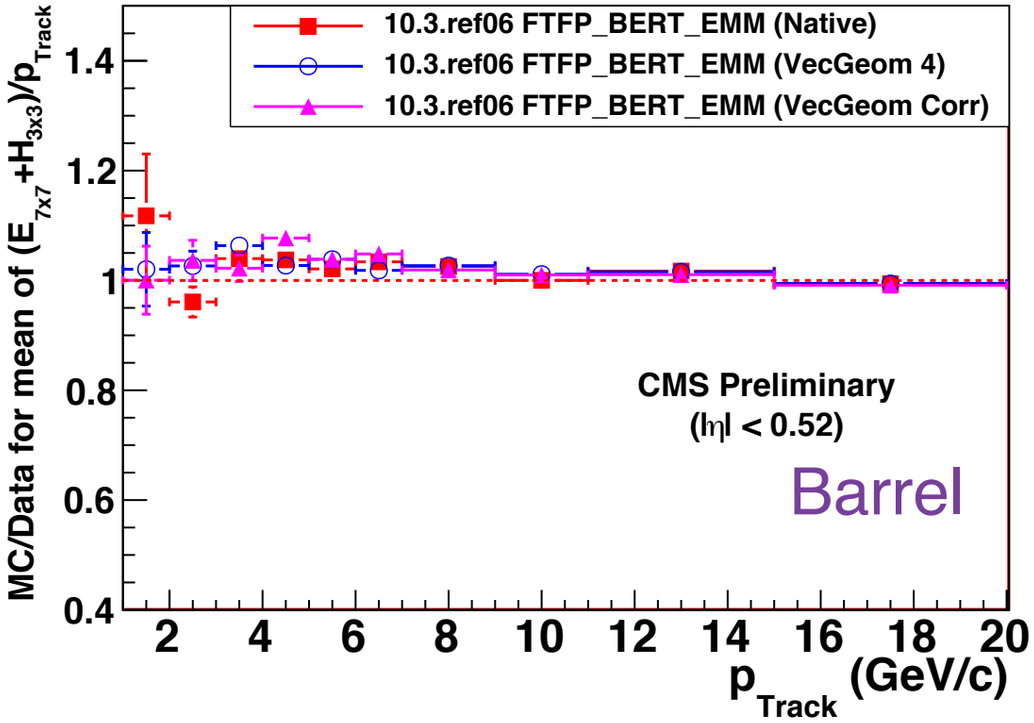


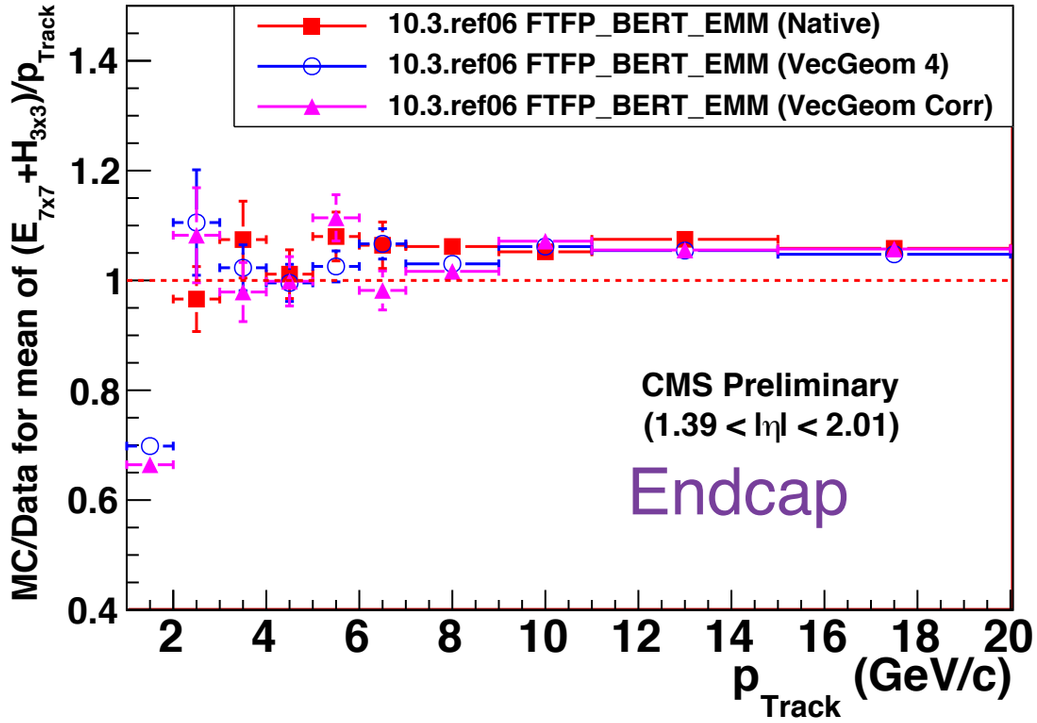


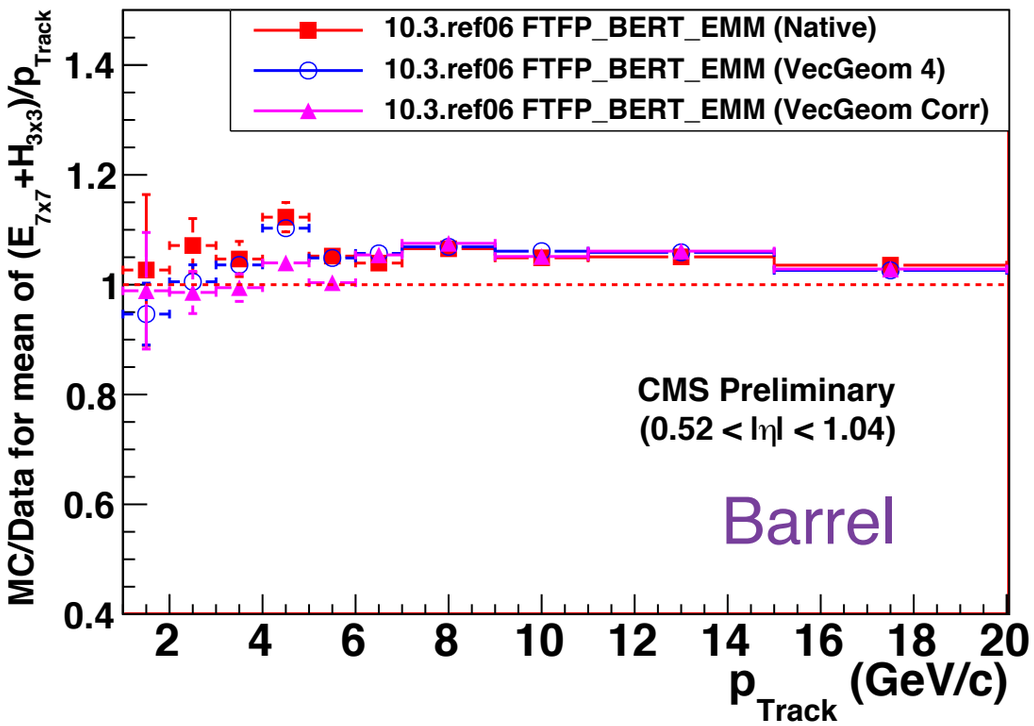
- There is no difference observed among the versions using native Geant4 geometry package or using the VecGeom package

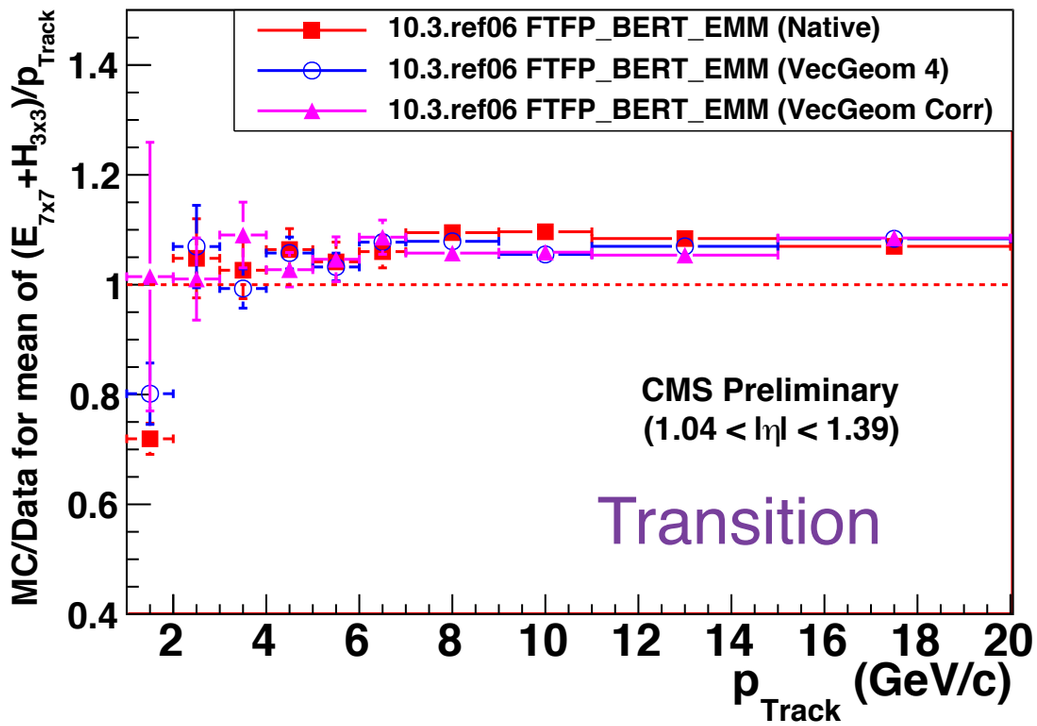
Physics Validation

- Propagate charged hadrons to the calorimeter surface. Demand them to be isolated in ECAL and HCAL. Measure energy in the calorimeter and compare that to the momentum measured in the Tracker. Study at different detector regions.









Issues with VecGeom Version

- There are reports of “stuck track” during track propagation with equal frequency in the Scalar and Vector builds. Geant4 with native geometry package does not report any stuck track for the same statistics
  - Two CMS geometry is tried so far - 2016 and 2018. Reports are similar
  - Stuck tracks relate to PolyCone or PolyHedra (in the statistics tried so far)
  - Summary of stuck tracks (# of occurrence in event samples):

Muon Barrel	Muon Endcap	Pion Barrel	Pion Endcap	Electron Barrel	Electron Endcap	Minimum Bias	Top Pair
0/3000	0/3000	0/3000	1/3000	0/3000	1/3000	29/500	83/300

- There were a few segmentation violations after tracks with direction along z-axis getting stuck
  - The features of these failures are reported to the VecGeom team
  - The VecGeom team (Raman) provided a fix which handles these cases
  - The version from Raman is integrated with CMSSW version
    - No crashes with 300 ttbar and 100k single particle events
    - Other stuck track feature is maintained with equal frequency as in the version 4.0
    - Physics quality and timing performance are unaffected with the fix

Summary

- CMSSW can now run with Geant4 version using VecGeom library replacing the Geant4 native geometry routines.
- There are a few remaining issues need to be addressed
  - Need to wait till the version becomes more robust
- The additional memory requirement is marginal
- There is some improvement in CPU performance for the scalar version and less improvement for the vectorized version. For the scalar version:
  - Muons and electrons show small improvement (~2-3%)
  - Pions show between 5-9% improvement
  - Minimum bias events show up to 8% improvement
  - t-tbar events show up to 8% improvement
- Physics validation do not show any significant differences