Hadronic Fast Simulation Response Smearing and SimHit Timing

Kevin Pedro
(University of Maryland)
May 2, 2013

Response Smearing

The hadronic shower fast simulation distributes the incident hadron’s energy into numerous energy spots according to a longitudinal and transverse shower profile.

After the energy spots are created, they are smeared with random numbers (generated once per event) distributed according to pion energy response distributions from full simulation. This accounts for various detector geometry effects and produces realistic results.

Originally, the energy response distributions were modeled as Gaussians, but this did not account for the tails, and the full simulation samples used to find the Gaussian μ and σ parameters had not been updated in several years.

Full Simulation

- Pions were generated for these energies (GeV): 1, 2, 3, 5, 9, 11, 15, 20, 30, 50, 100, 150, 225, 300, 1000, 3000

- For each energy, a sample of 500,000 pions was generated, uniformly distributed in φ and in $-5 \leq \eta \leq 5$

- Settings:
  GEN, SIM steps
  Standard geometry
  Magnetic field turned off
  No vertex smearing

Energy Response

Energy response is calculated as follows for each event:
$E = E_{ECAL} + f_{HB}(\eta)\cdot E_{HB} + f_{HE}(\eta)\cdot E_{HE} + f_{HO}(\eta)\cdot E_{HO} + \text{poisson}\ (6\cdot E_{HF})/(6\cdot f_{HF}(d))$

E_{ECAL} is the ECAL energy
$E_{HB,HE,HO}$ are the HB, HE, HO energy (sensitive layers)
$f_{HB,HE,HO}$ are the sampling factors (η-dependent, given in SimCalorimetry/HcalSimProducers/python/hcalSimParameters_cfi.py)
E_{HF} is the mean number of photoelectrons in HF, which is smeared according to a Poisson distribution and then converted to GeV using f_{HF} (depth-dependent)

ECAL and HCAL energies come from SimHits, collected inside a cone of $dR < 0.5$ from the incident pion's η and φ coordinates.

Double-sided Crystal Ball Function

$$N = \frac{1}{\sigma \left[\frac{n_L}{a_L} \cdot \frac{1}{n_L - 1} \cdot \exp\left(-\frac{a_L^2}{2}\right) + \sqrt{\frac{\pi}{2}} \left(\mathrm{erf}\left(\frac{a_L}{\sqrt{2}}\right) + \mathrm{erf}\left(\frac{a_R}{\sqrt{2}}\right)\right) + \frac{n_R}{a_R} \cdot \frac{1}{n_R - 1} \cdot \exp\left(-\frac{a_R^2}{2}\right) \right]}$$

$$f(x;\vec{p}) = N \cdot \begin{cases} \left(\frac{n_L}{a_L}\right)^{n_L} \cdot \exp\left(-\frac{a_L^2}{2}\right) \cdot \left(\frac{n_L}{a_L} - a_L - \frac{x-\mu}{\sigma}\right)^{-n_L} & \text{for } \frac{x-\mu}{\sigma} \leq -a_L \\ \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) & \text{for } -a_L < \frac{x-\mu}{\sigma} < a_R \\ \left(\frac{n_R}{a_R}\right)^{n_R} \cdot \exp\left(-\frac{a_R^2}{2}\right) \cdot \left(\frac{n_R}{a_R} - a_R + \frac{x-\mu}{\sigma}\right)^{-n_R} & \text{for } \frac{x-\mu}{\sigma} \geq a_R \end{cases}$$

The Crystal Ball function has a Gaussian core with parameters μ and σ, and power-law tails on each side with parameters a_L, n_L, a_R, and n_R. The parameters a_L and a_R give the locations where the tails start (in units of σ) and the parameters n_L and n_R give the steepness of the tails.

Fitting Details

Energy response distributions are fit with the Crystal Ball function for each energy and each bin of 0.1η. Fit range is typically $[0.1 \cdot E, 2 \cdot E]$.

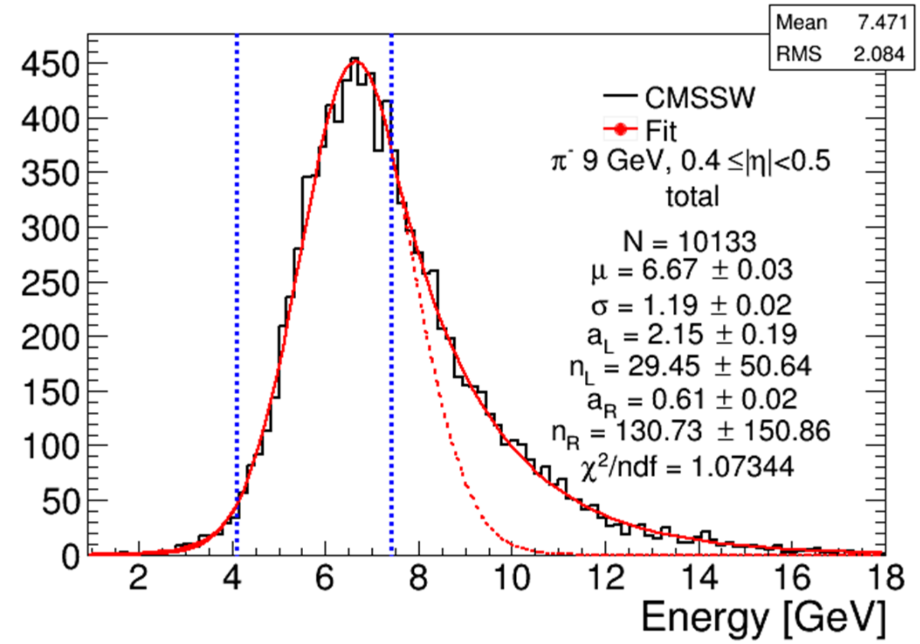
The fit parameters are given limits as follows:
$\mu, \sigma \geq 0$ $0 \leq a_L, a_R \leq 10$ $1.01 \leq n_L, n_R \leq 200$

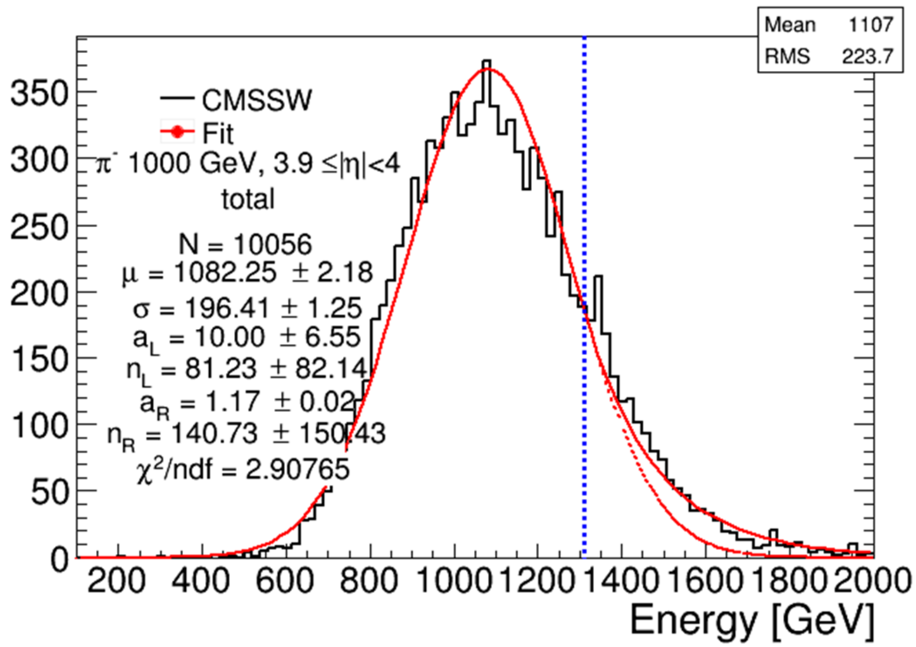
Limits on the tail parameters are based on inspection of the results from fitting without any limits.

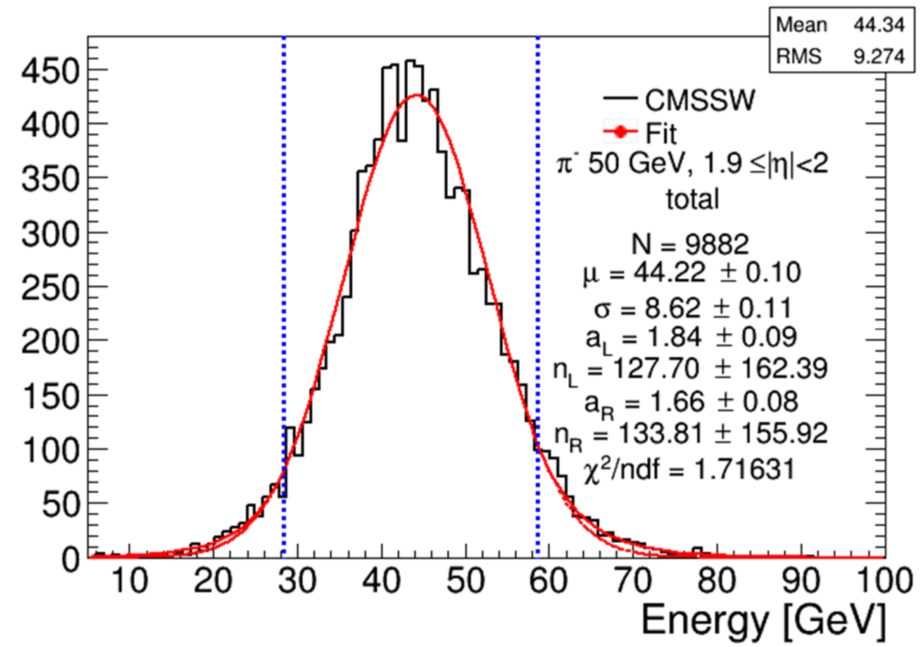
Response distributions for mips ($E_{ECAL} < 0.8$ GeV) and nonmips ($E_{ECAL} > 0.8$ GeV) were also generated and fit separately. These results are stored, but will not be used by default due to inaccuracies in the mip percentage in the hadronic fast sim.

(Each fit was checked manually, and adjusted in those cases where the fitting algorithm fell into a bad local minimum.)

Example of Response Fits







Parameter Interpolation

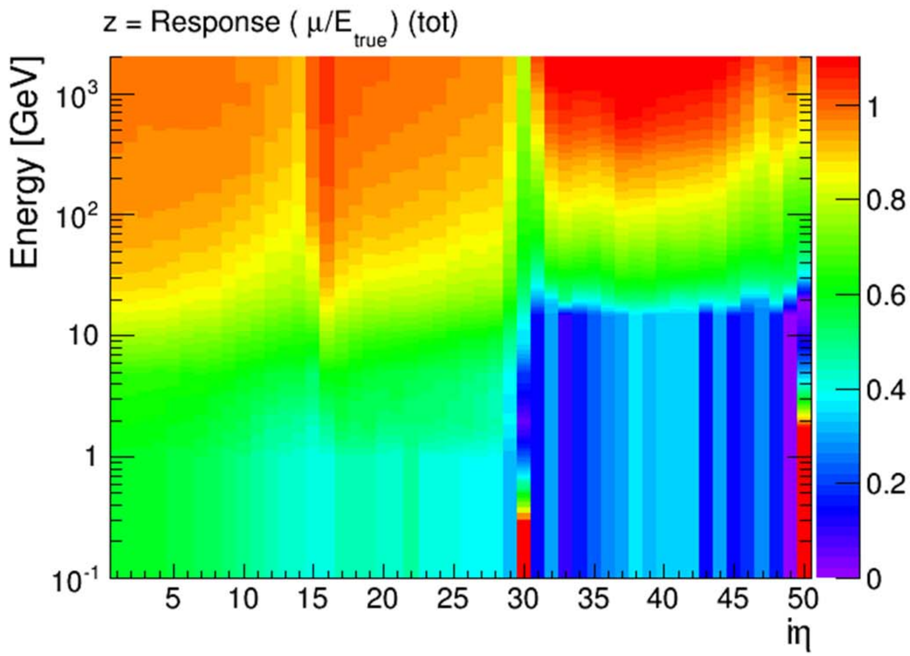
After fitting energy response distributions, the parameters are stored and linearly interpolated (within each η bin) for intermediate energies.

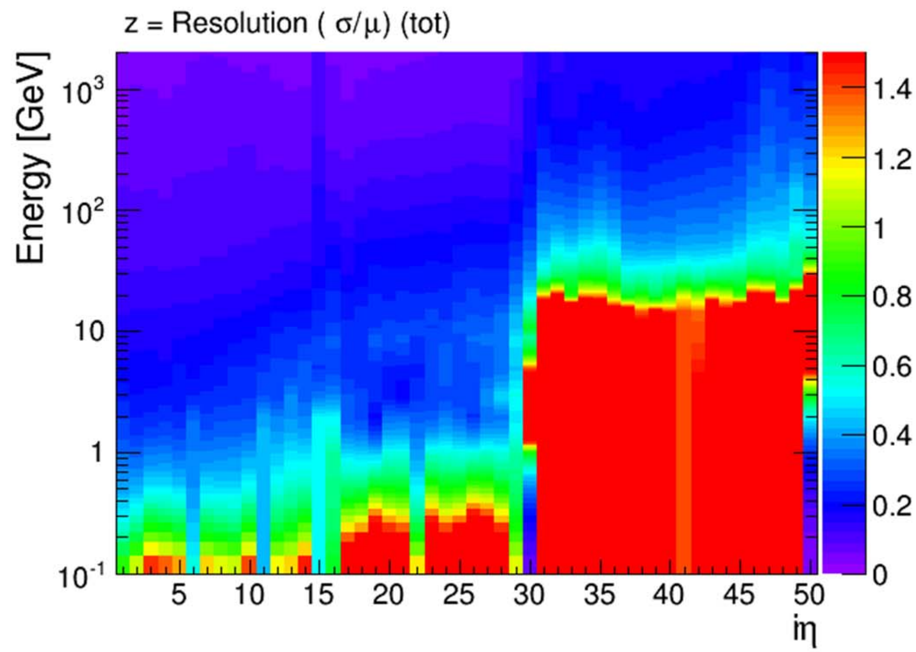
There are various safety checks to ensure that non-physical values of the parameters (μ, σ, a_L, $a_R < 0$; n_L, $n_R < 1$) are not generated.

In the HF region, $|\eta| < 3.0$, low-energy distributions are not well-behaved enough to fit. Parameters are extrapolated down for energies < 15 GeV.

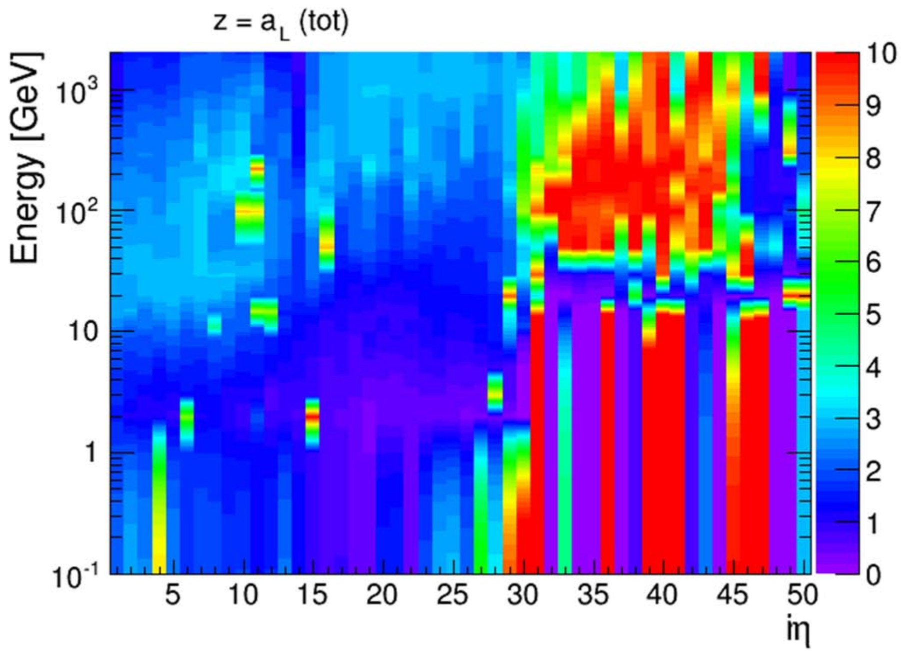
Subsequent plots will show the energy response, resolution, and tail parameters over the entire energy and η ranges, including interpolation for intermediate energies.

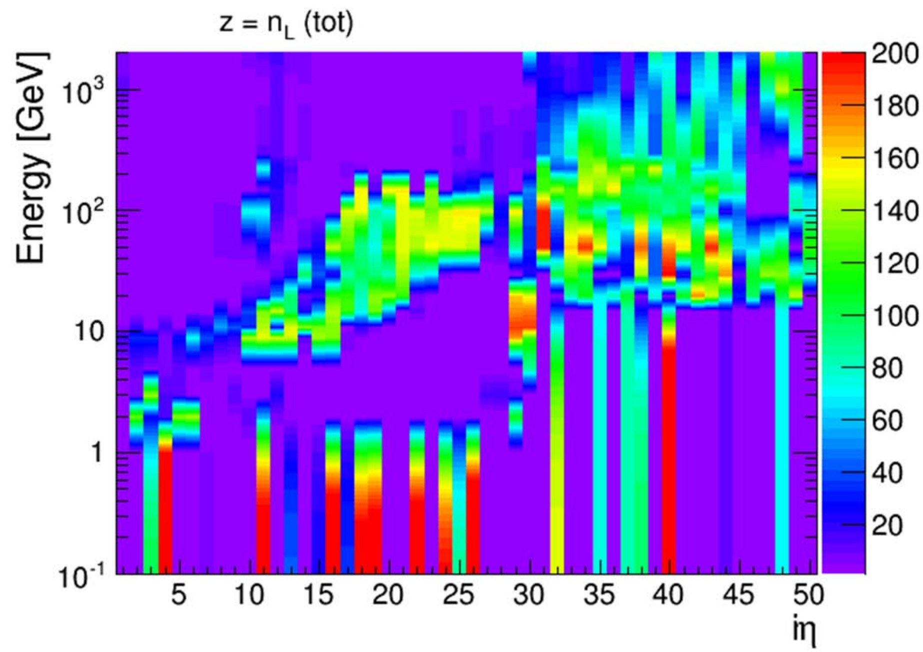
Energy Response and Resolution



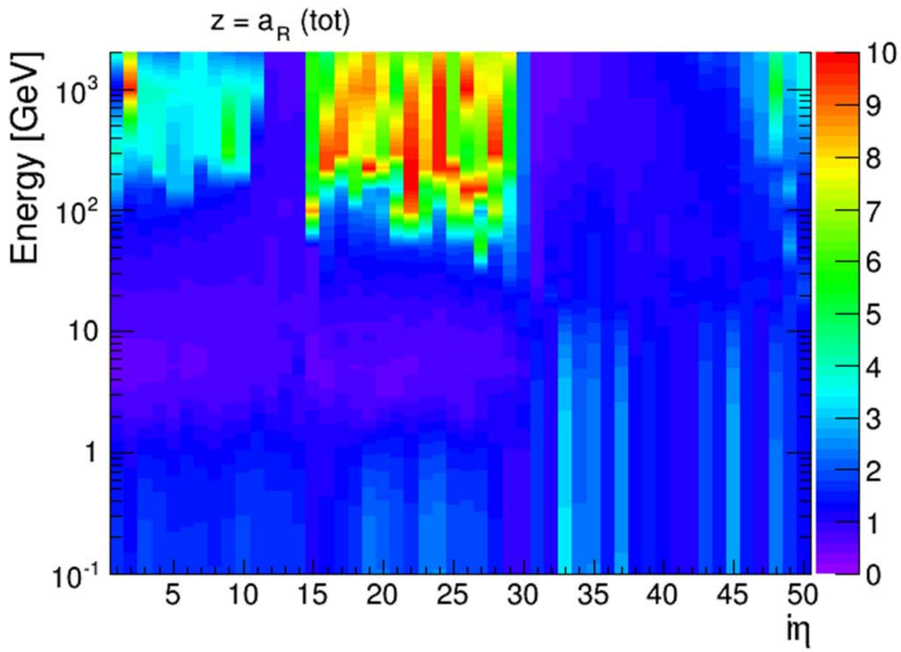


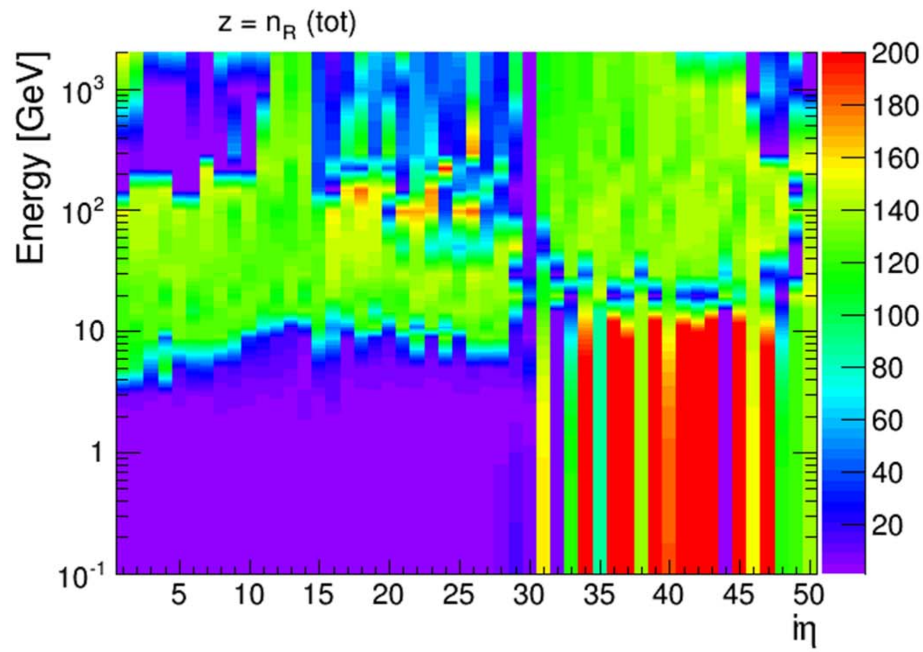
Left Tail





Right Tail





Technical Details

In the course of this update, structural changes were made to HCALResponse. All of the random shooting for hadron response smearing (as well as muons, and electrons in HF) is now done in HCALResponse rather than CalorimetryManager. Also, the Python parameters are now stored in a more generalized way.

Instead of gaussShoot, a class DoubleCrystalBallGenerator has been written, which uses inversion sampling to generate random numbers according to a double-sided Crystal Ball distribution with user-provided parameters. Gaussian smearing is still available as a “fallback” option for users.

Speed Test

The performance of these changes were tested, following the instructions at https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideFastSimTimingHowTo. Results:

Old fast sim:
Total time = 4.34e+03 +/- 2.5e+03 ms/event

New fast sim: (still Gaussian smearing)
Total time = 5.14e+03 +/- 2.63e+03 ms/event

New fast sim: (Crystal Ball smearing)
Total time = 4.93e+03 +/- 2.74e+03 ms/event

These all agree very well within errors bars, which are ~50%.

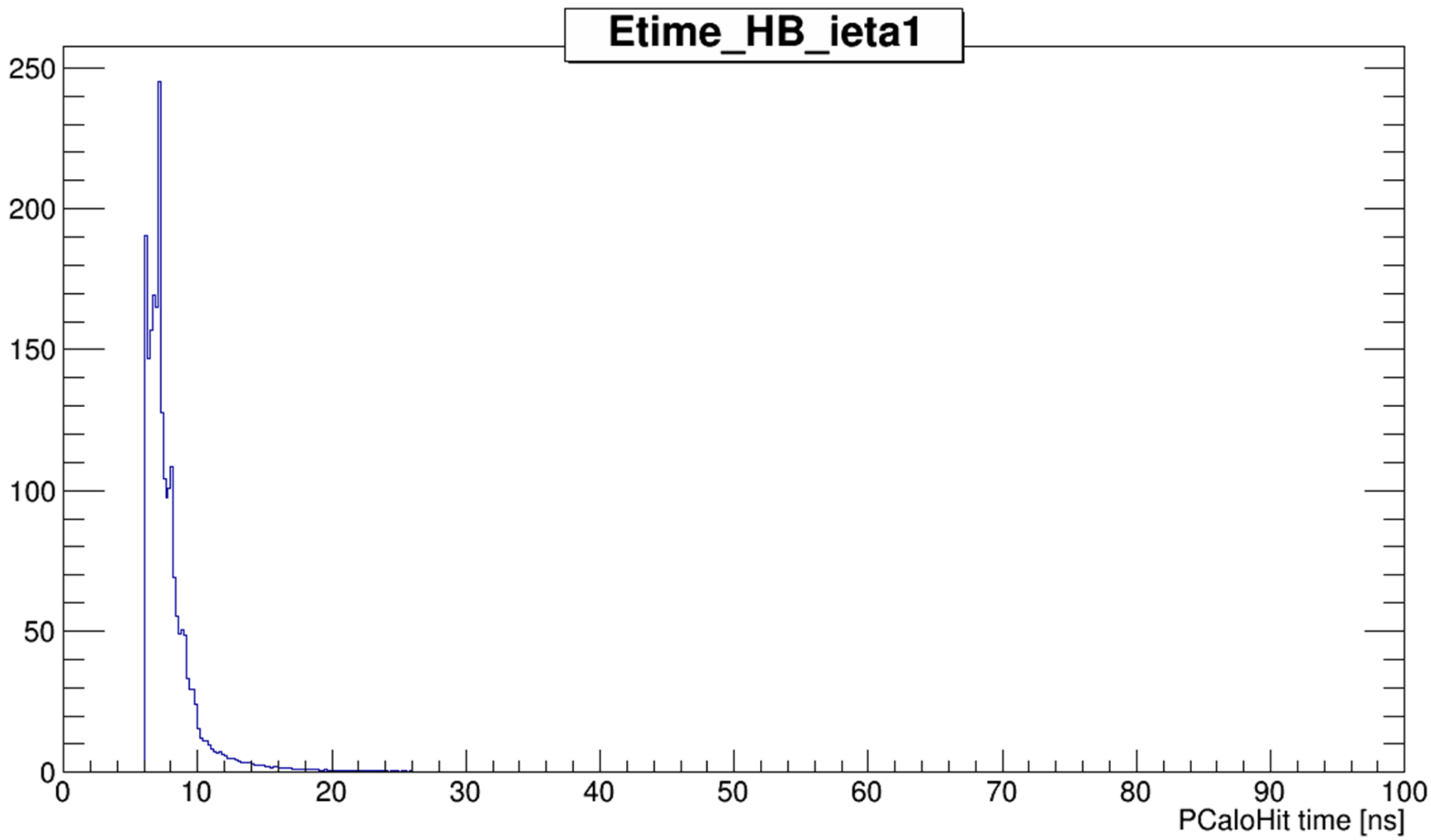
Timing

Applying appropriate timing to HCAL SimHits is important for the digitization process. Calculating simple time of flight is not quite good enough; in Geant4, shower development time and other effects are simulated, which causes the actual timing of hits to have a deviation from the time of flight, called jitter.

Looking at a simple histogram of PCaloHit timing makes it appear that there is a very broad distribution, but most of the tail consists of hits with very low energy. Constructing a histogram of PCaloHit timing with each entry weighted by the energy of the PCaloHit isolates the timing peak very clearly.

The timing peak is found for each cell ieta of each HCAL subdetector and used for the time of the appropriate fast sim hits. This is “good enough” because it means all of the energy will be deposited in the time region which is used by the digitizer.

Sample Energy-weighted Timing Histogram



Technical Details (2)

In the course of this update, structural changes were made to CalorimetryManager and the CaloHitMaker classes.

The CaloHitMaker classes now use the standard CaloHitID class from full sim. This class keeps track of the DetID, timing, and associated track ID, and has well-defined comparison operators so it can be used as the key value in a map. Timing of hits is now set by default in the CaloHitMaker classes, although it can be overwritten in CalorimetryManager (currently done for HCAL).

The processing of hits in CalorimetryManager has been streamlined and standardized to reduce unnecessary computation and to ensure that all hits (from the various simulation processes EMShowerSimulation, HDShowerSimulation, reconstructHCAL, and MuonMipSimulation) are treated correctly.

(Previously, hits from reconstructHCAL were not calibrated in the proper way for the digitizer, which led to significant problems.)

Speed Test (2)

The performance of these changes were tested, following the instructions at https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideFastSimTimingHowTo. Results:

Old fast sim:
Total time = 4.34e+03 +/- 2.5e+03 ms/event

New fast sim: (new response and timing)
Total time = 5.39e+03 +/- 2.71e+03 ms/event

These agree within errors bars, which are ~50%.

Validation

The goal of this update was to retune the hadronic response, in order to give accurate results (without any ad hoc correction factors) when using full sim reconstruction after digitizing fast sim SimHits.

For the validation, this update was combined with the preshower digitization process done by Federica Primavera.

Samples of 9000 $t\bar{t}$ events were generated with a “clean” fast sim (using CMSSW_6_2_0_pre3) and the modified fast sim (using CMSSW_6_2_0_pre5). These were compared to full sim relvals.

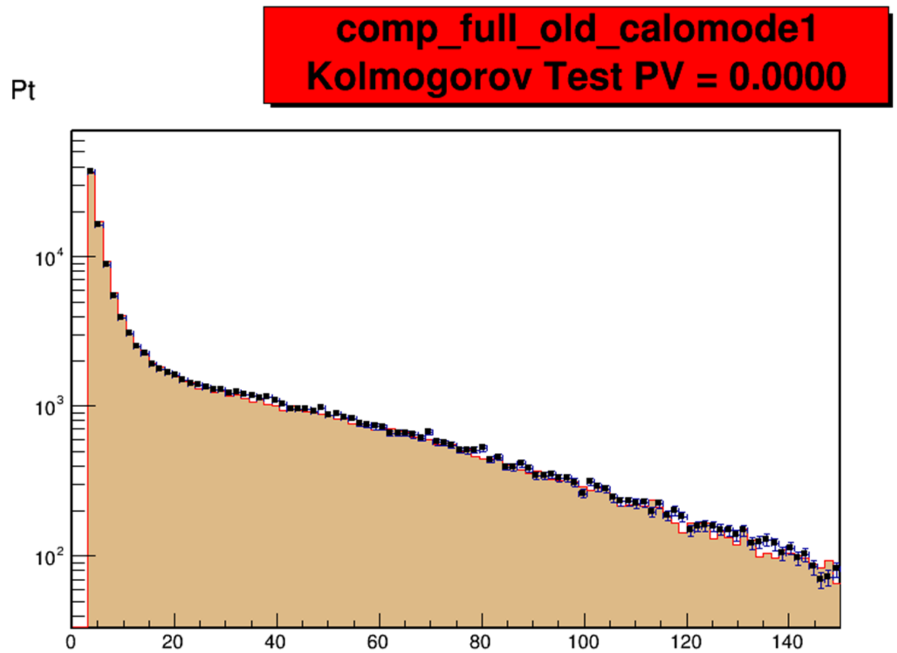
Validation

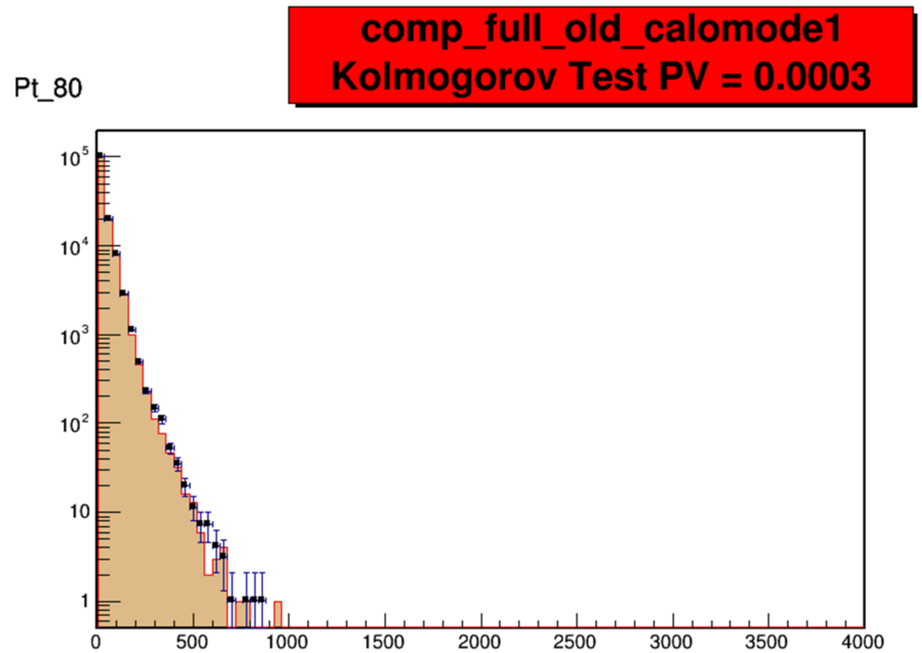
The next slides show comparisons of ak5 PFJet quantities, between:

1) full sim and clean fast sim (CaloMode=1)
http://kjplanet.com/comp/comp_full_old_calomode1/PFlowAntiKt5/htmlTemplate_PF.html
2) full sim and modified fast sim (CaloMode=1)
http://kjplanet.com/comp/comp_full_new_v4-2_calomode1/PFlowAntiKt5/htmlTemplate_PF.html
3) full sim and clean fast sim (CaloMode=3)
http://kjplanet.com/comp/comp_full_old_calomode3/PFlowAntiKt5/htmlTemplate_PF.html
4) full sim and modified fast sim (CaloMode=3)
http://kjplanet.com/comp/comp_full_new_v4-2_calomode3/PFlowAntiKt5/htmlTemplate_PF.html

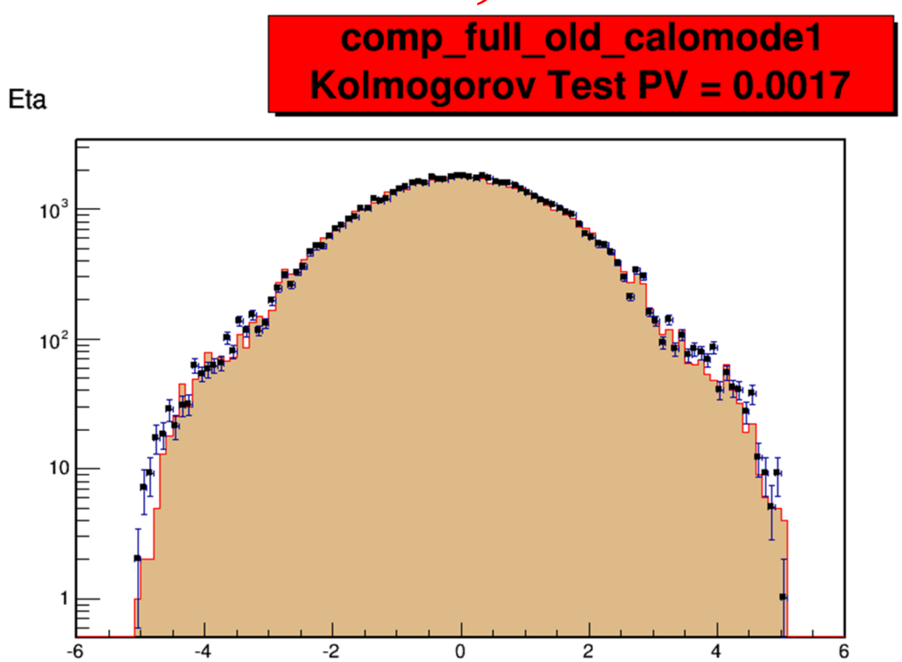
The η and p_T spectrum agreement with full sim are both improved from (3) to (4), comparable with (1).

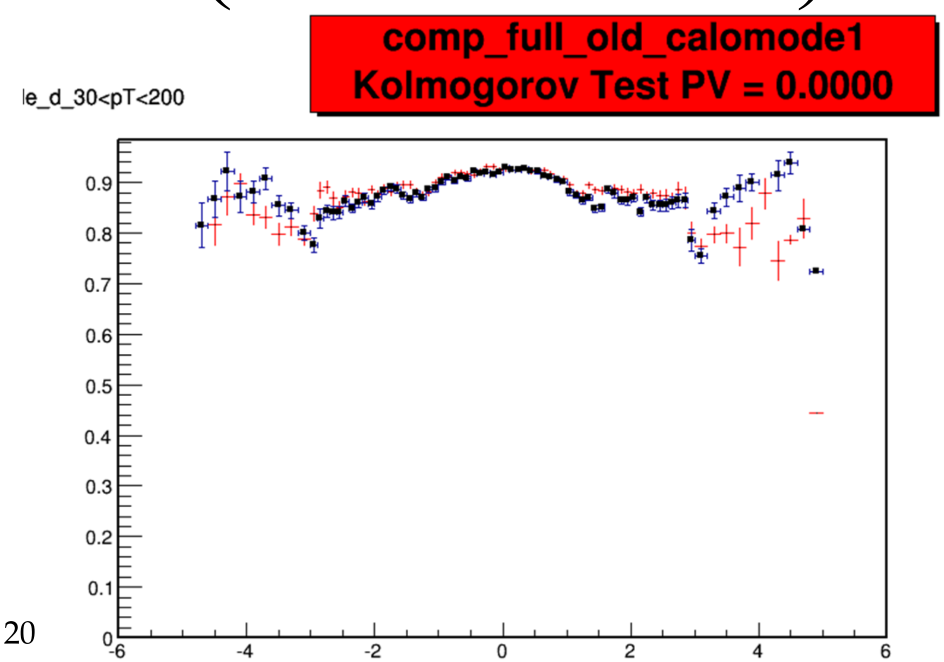
There is some disagreement in the endcap p_T scale in (4) which could be investigated further.

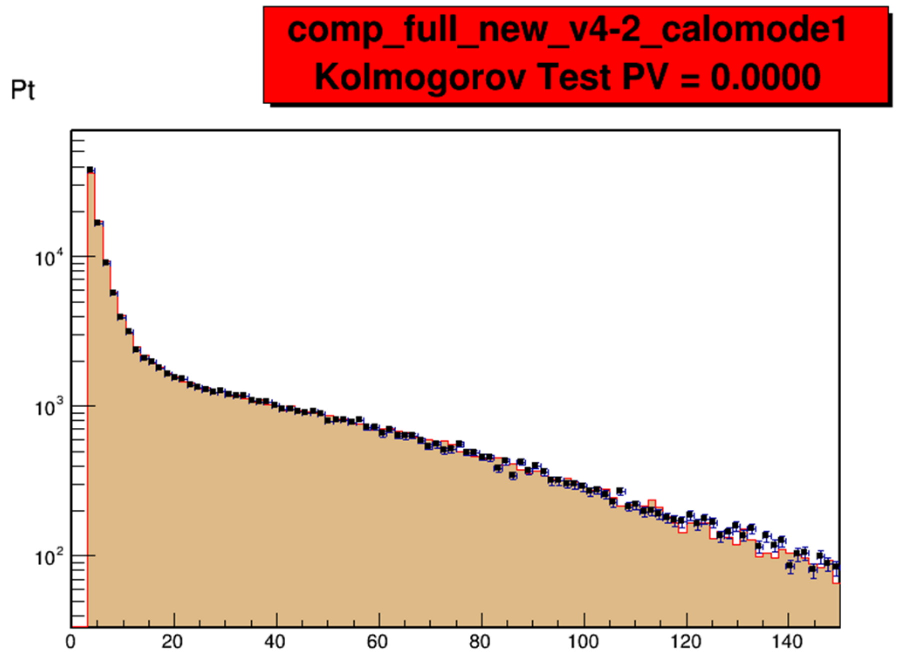


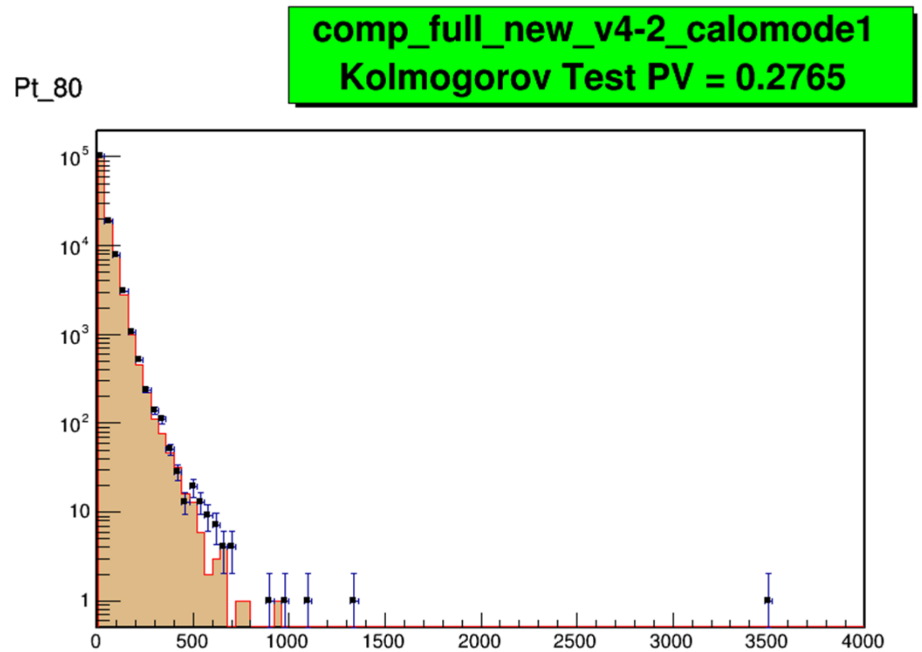


Full Sim, Old Fast Sim (CaloMode=1)

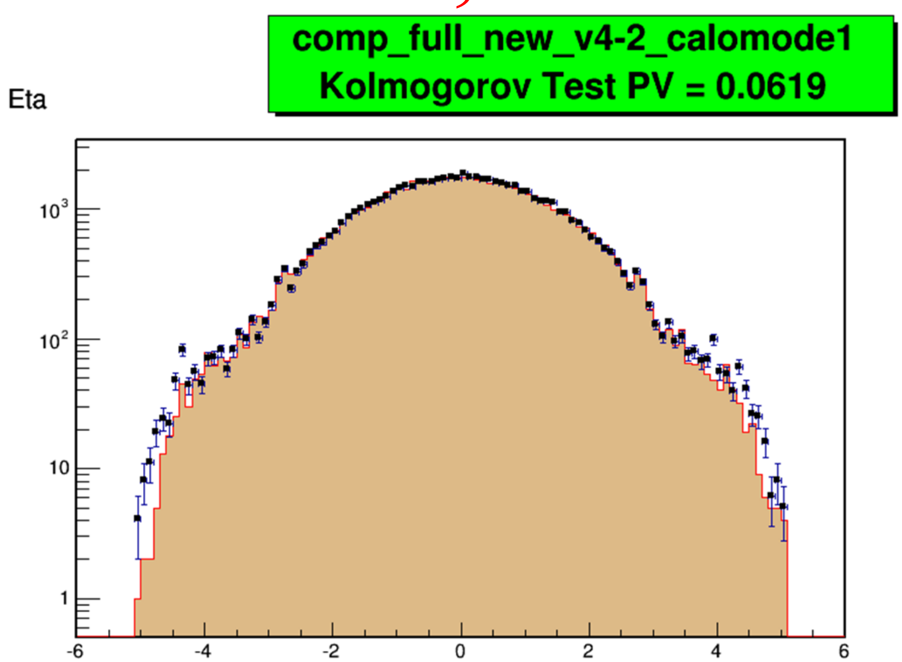


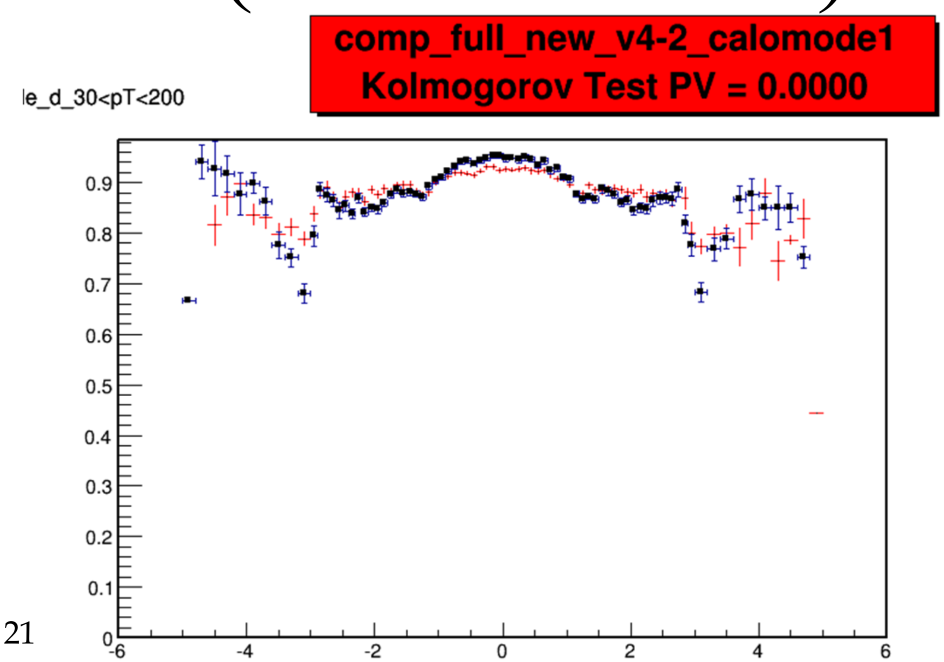


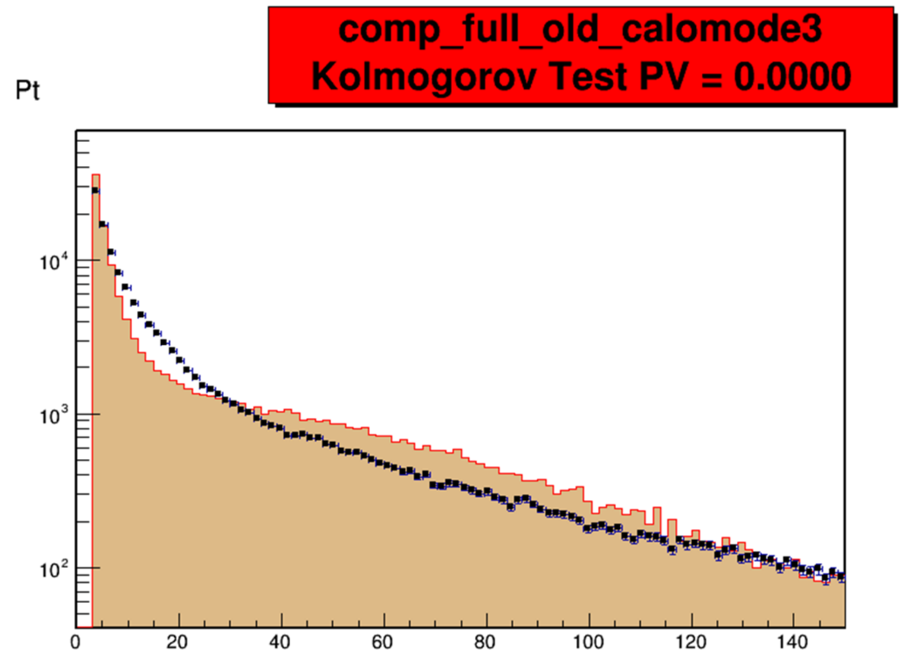


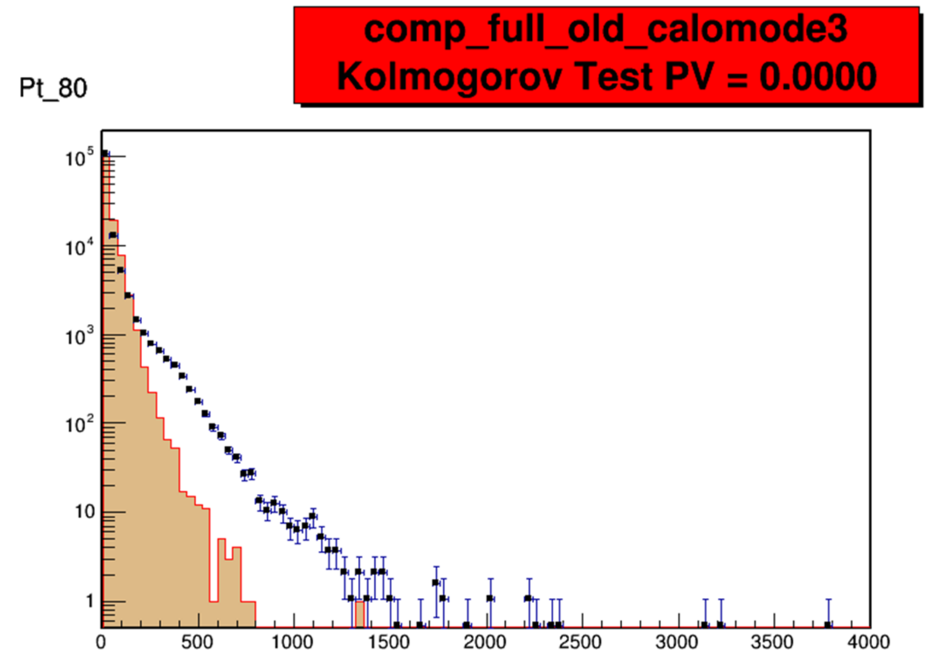


Full Sim, New Fast Sim (CaloMode=1)

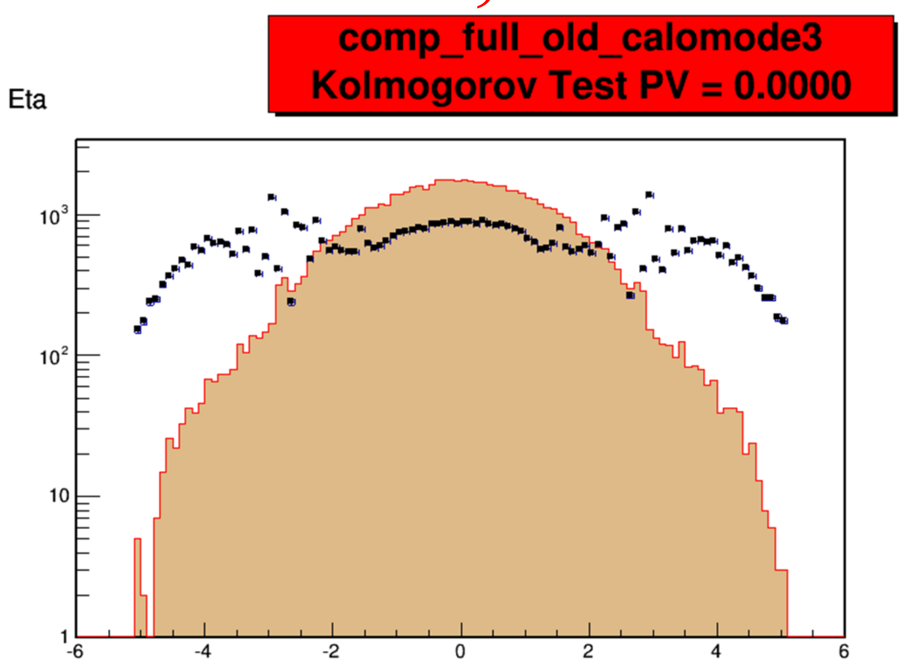


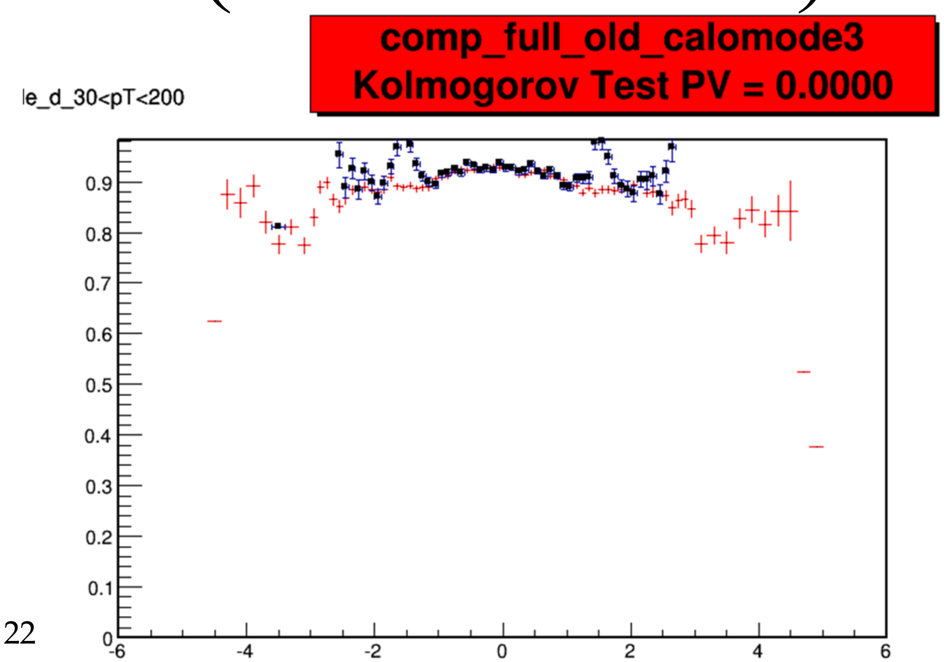


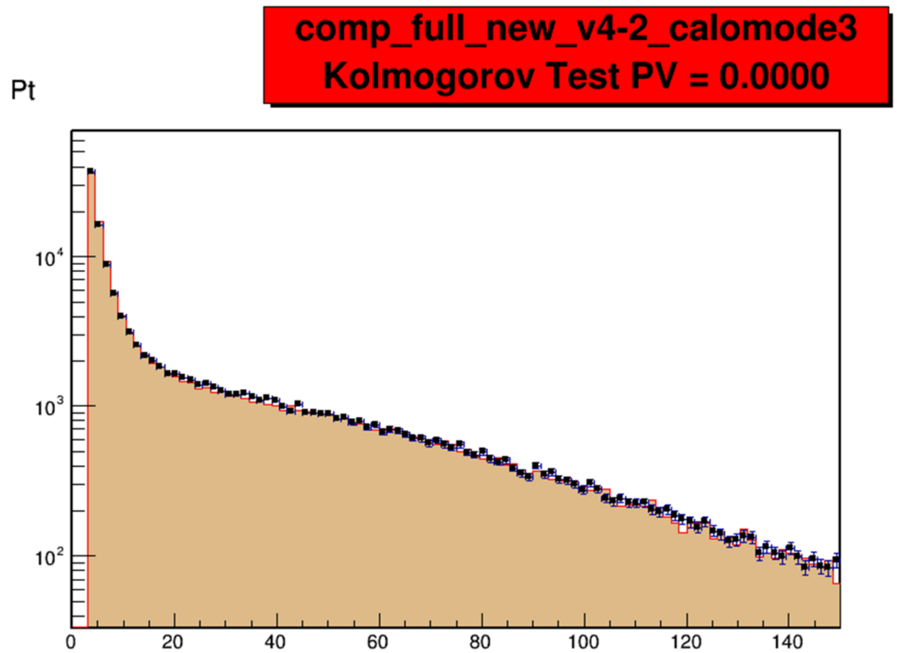


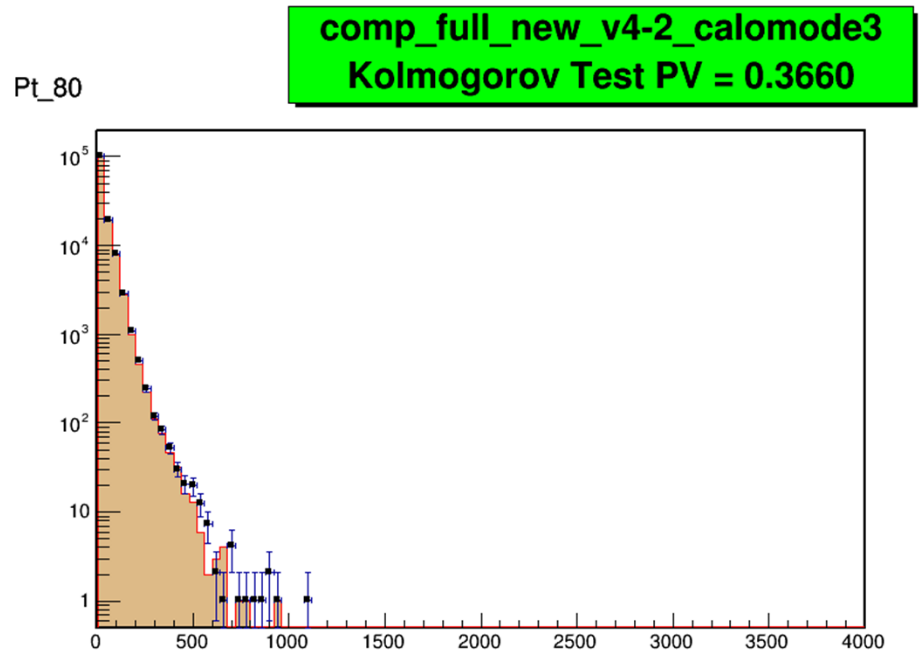


Full Sim, Old Fast Sim (CaloMode=3)

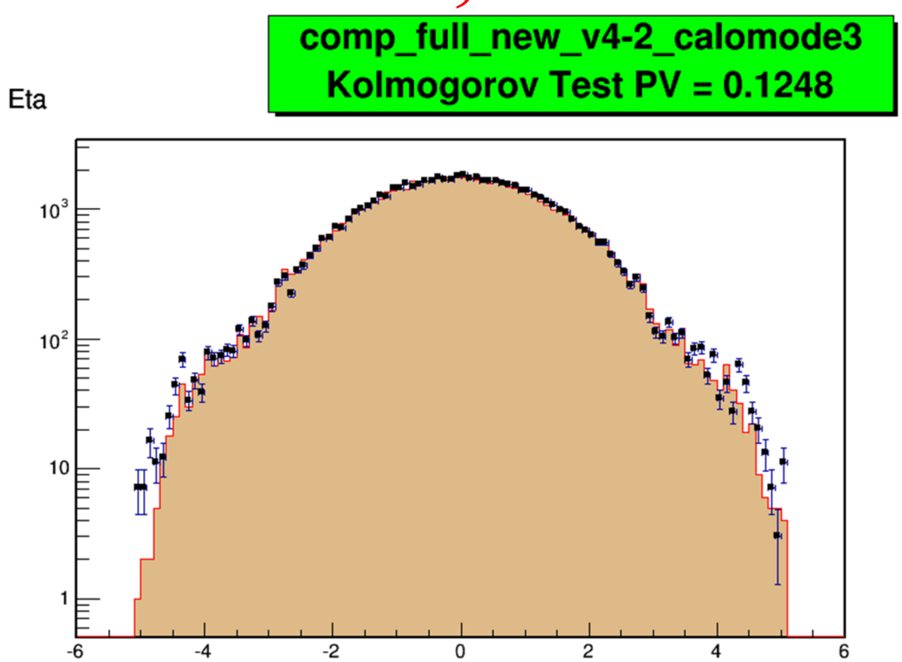


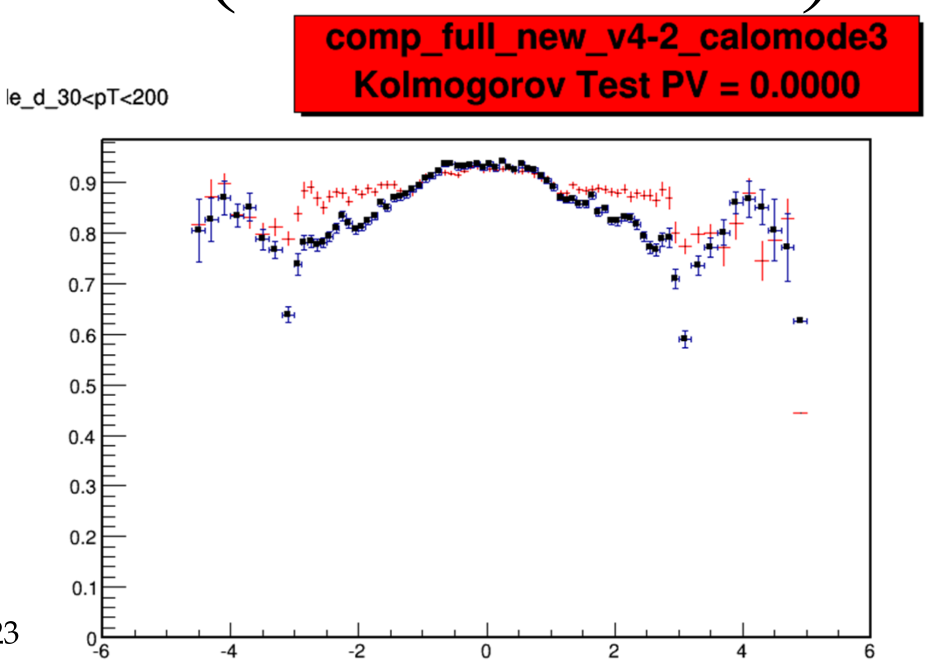






Full Sim, New Fast Sim (CaloMode=3)





Conclusions

This update to the hadronic fast simulation response smearing captures more physics and presents better agreement with full simulation when using the digitization process.

The code will be included in CMSSW_6_2_0_pre6.

The configuration files, analyzer, and ROOT macros used to create and analyze the full simulation pion samples will be available in (once finished): UserCode/pedrok/PionResponse

I would like to thank Salavat Abdoulline, Andrea Giammanco, and Federica Primavera for their help and contributions!

Backup