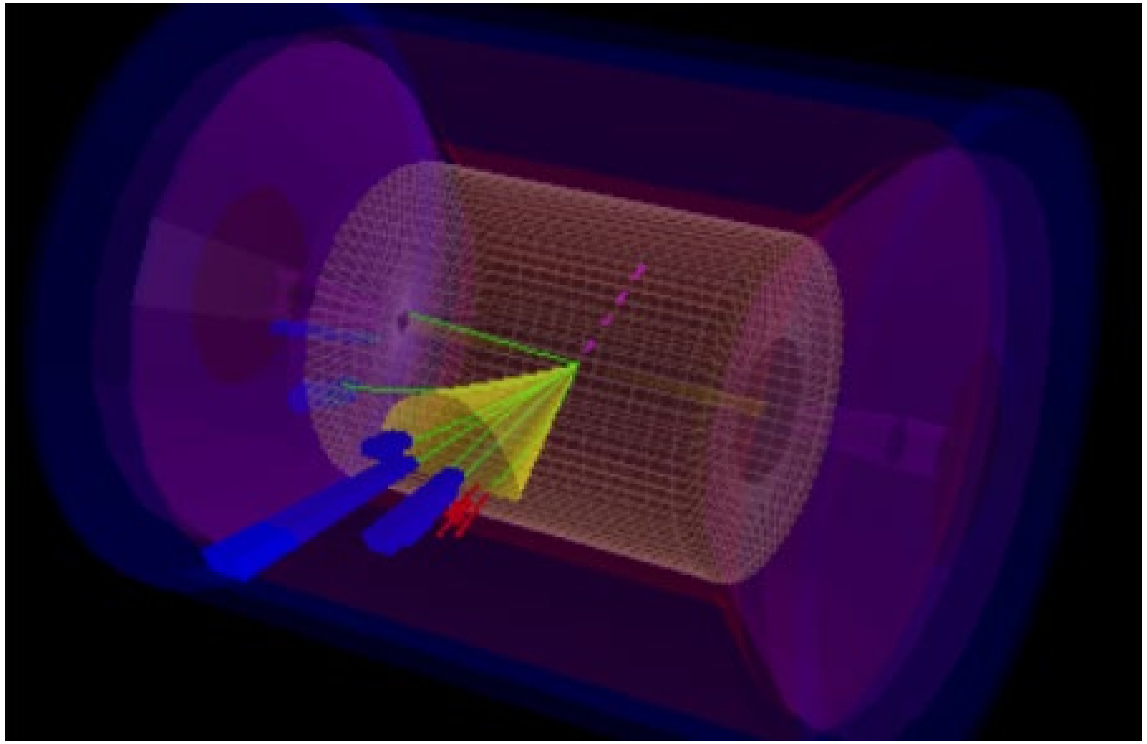
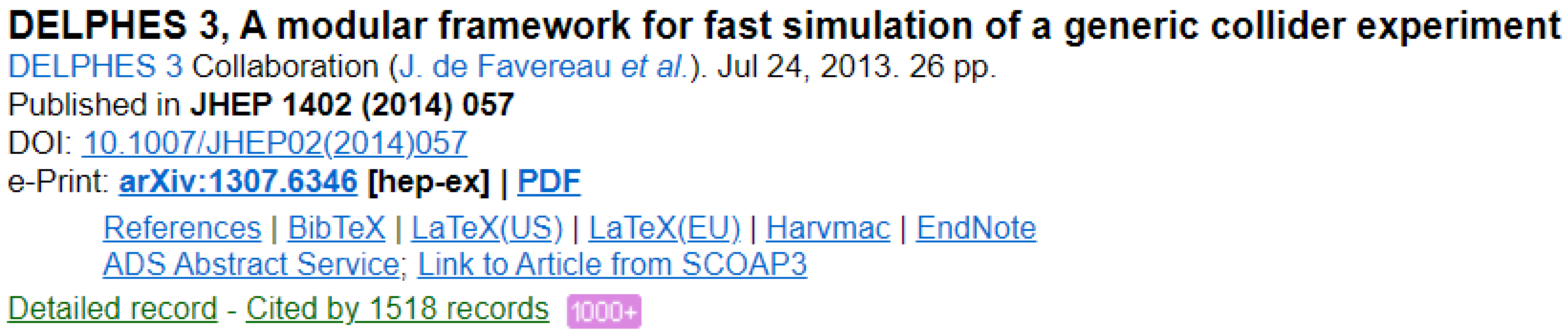
My experience with DELPHES

Miguel Arratia

UNIVERSITY OF CALIFORNIA
UC RIVERSIDE



Fast simulation with Delphes3



DELPHES 3, A modular framework for fast simulation of a generic collider experiment
DELPHES 3 Collaboration (J. de Favereau *et al.*). Jul 24, 2013. 26 pp.
Published in **JHEP 1402 (2014) 057**
DOI: 10.1007/JHEP02(2014)057
e-Print: **arXiv:1307.6346 [hep-ex] | PDF**

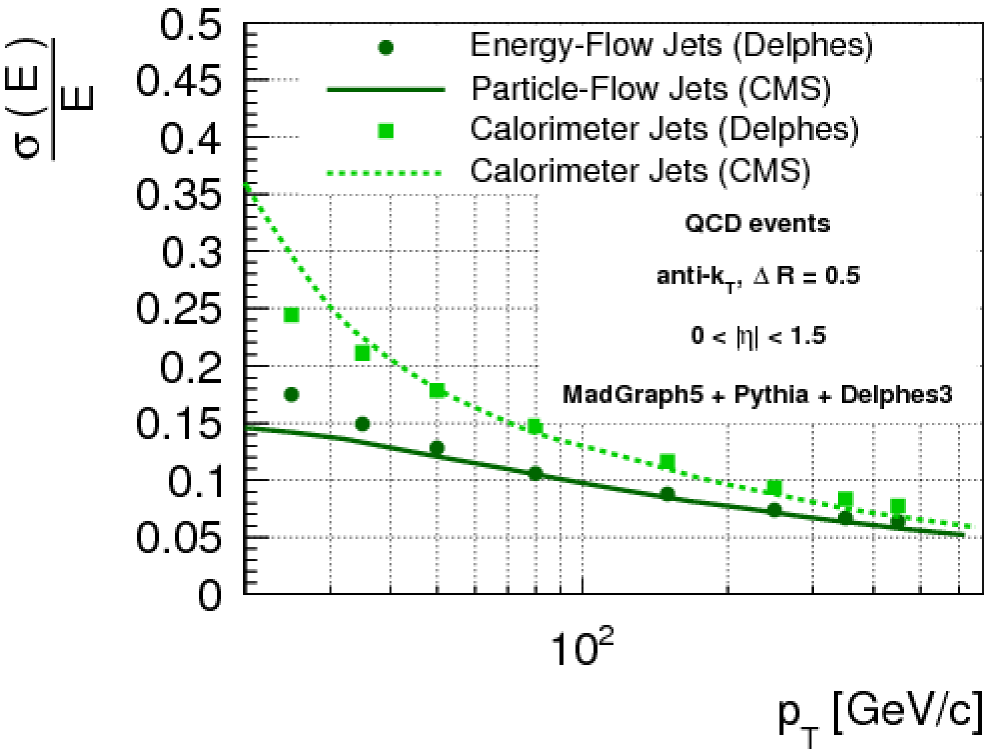
References | BibTeX | LaTeX(US) | LaTeX(EU) | Harvmac | EndNote
ADS Abstract Service; Link to Article from SCOAP3

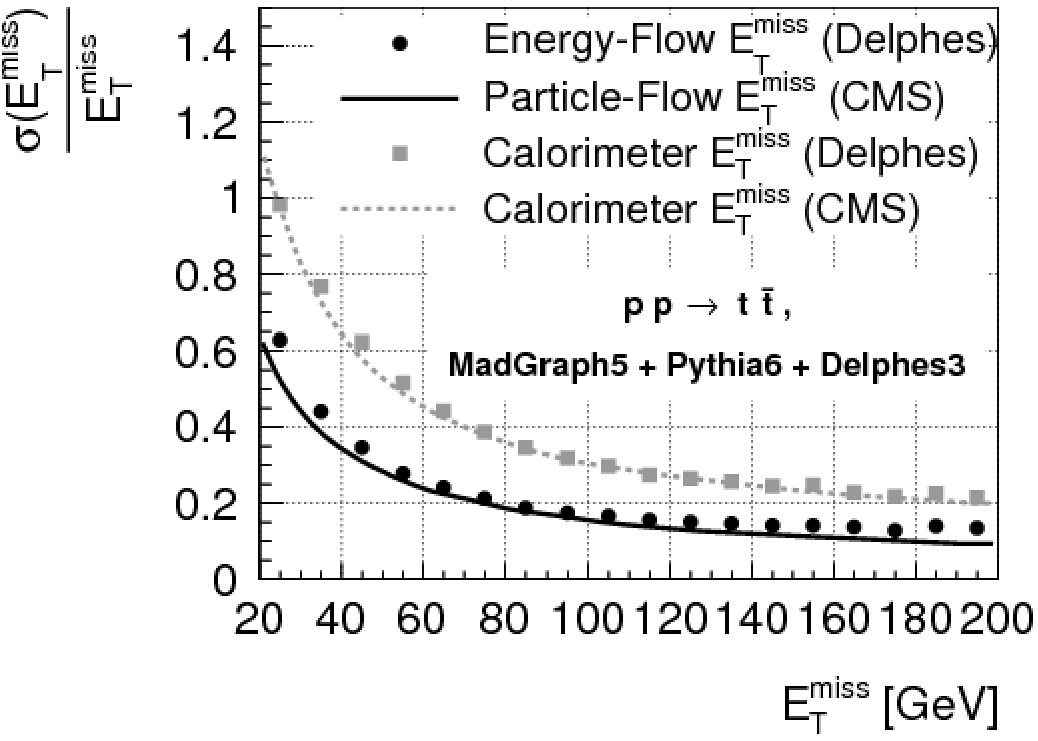
Detailed record - Cited by 1518 records 1000+

Citations include:
“Higgs Physics at the HL-LHC and HE-LHC” - Cepeda, M. et al. **CERN Yellow Rep.Monogr. 7 (2019)**
“Physics at a 100 TeV pp Collider: Standard Model Processes” - Mangano, M.L. et al**. CERN Yellow Rep. (2017)**
“FCC Physics Opportunities : Future Circular Collider Conceptual Design Report Volume 1” Eur.Phys.J. C79 (2019) no.6, 474
“The Compact Linear Collider (CLIC) - 2018 Summary Report” **CERN Yellow Rep.Monogr. 1802 (2018) 1-98**
Also several studies for ILC, CEPC…etc.

- It is based on parametrized tracking and calorimeter resolutions.
- Pythia8-Delphes3 can be run simultaneously. Accepts HEPMC and other formats as well
- It includes bending in magnetic field, granularity of calorimeters (not longitudinal segmentation though). PID efficiency/fake-rate, Jet reconstruction, particle flow, missing-energy, b-tagging, tau-tagging etc.

Jet/Met performance in Delphes vs CMS



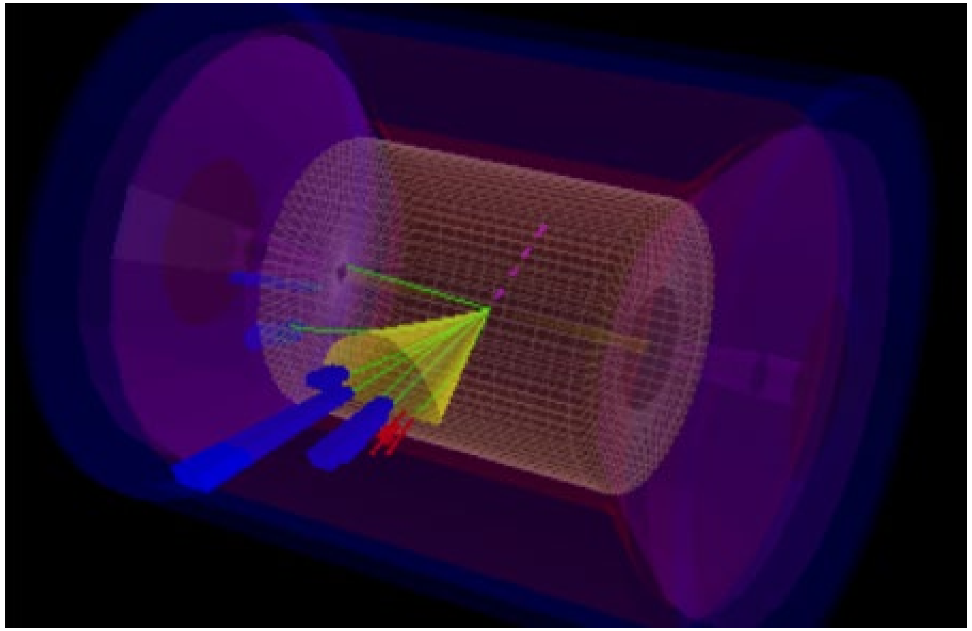


This is **not** by construction, it emerges from tracking and calorimetry resolution and granularity, as well as implementation of “particle flow”

EIC detector in Delphes

https://github.com/miguelignacio/delphes_EIC/blob/master/delphes_card_EIC.tcl

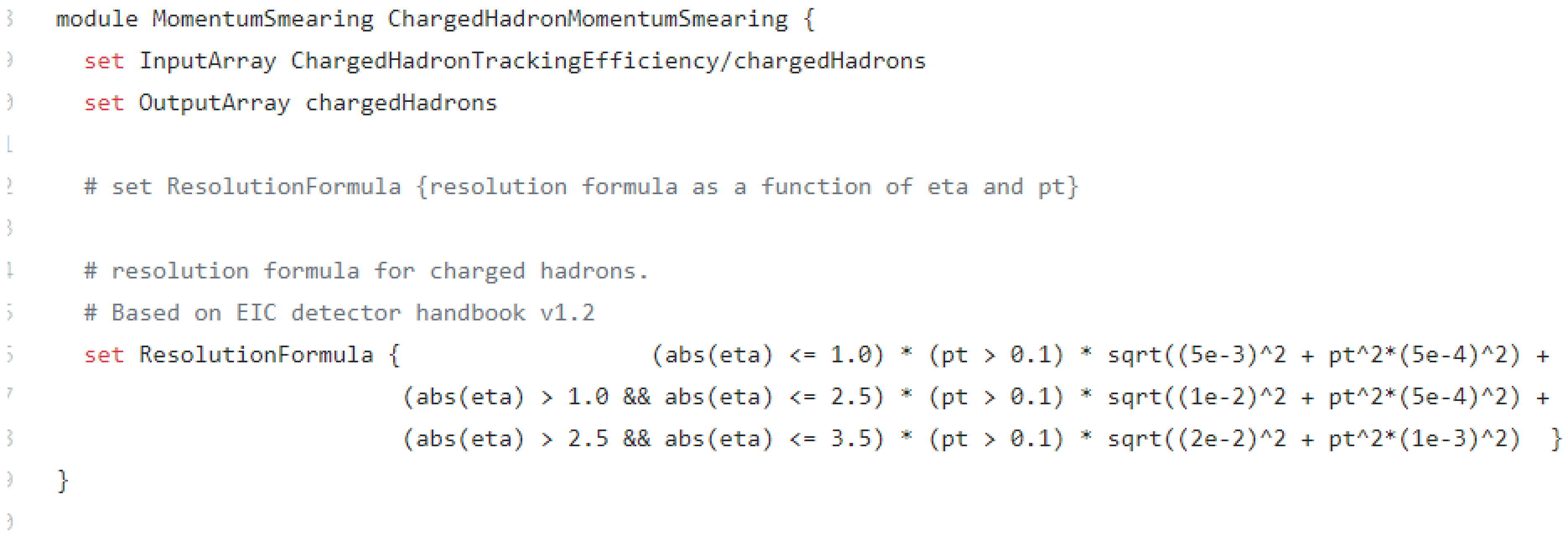
Tracking resolution, EMCAL resolution and HCAL resolution as in detector handbook.



In addition:

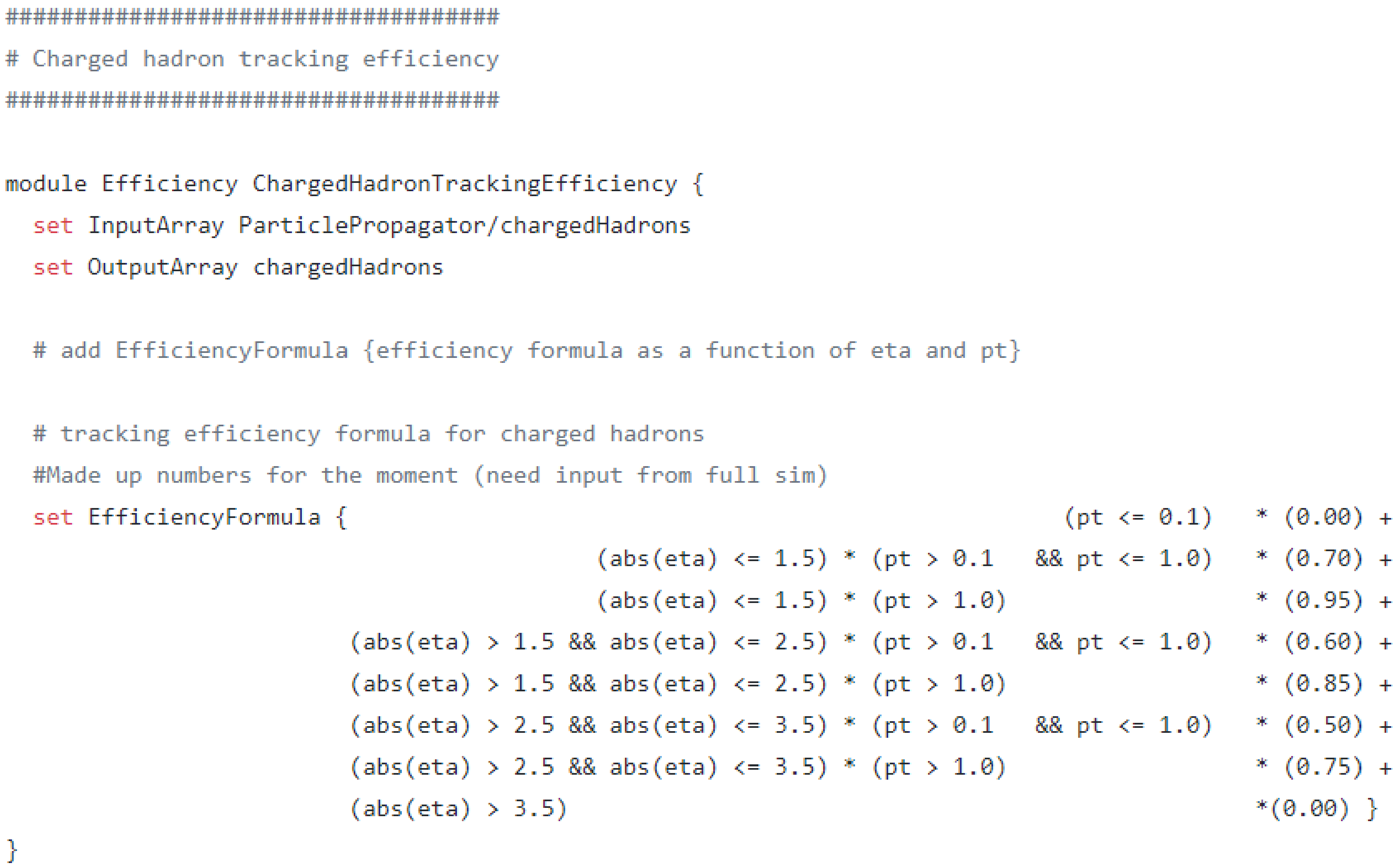
- B=1.5 T, R=0.80 m, L = 1 m
- EMCAL granularity (dphi x deta):
  0.0174 x 0.02 for |eta|<3.5
- HCAL granularity (dphi x deta):
  0.087 x 0.10 for |eta|<1.0
  0.174 x 0.20 for 1.0 |eta|<3.4
- HCAL resolution:
  100%/sqrt(E) + 10% in barrel
  50%/sqrt(E) + 10% in encap
- No PID yet, but it can be included (LHCb is in Delphes).
  Need parametrization of efficiency and mis-identification matrix

Tracking resolution

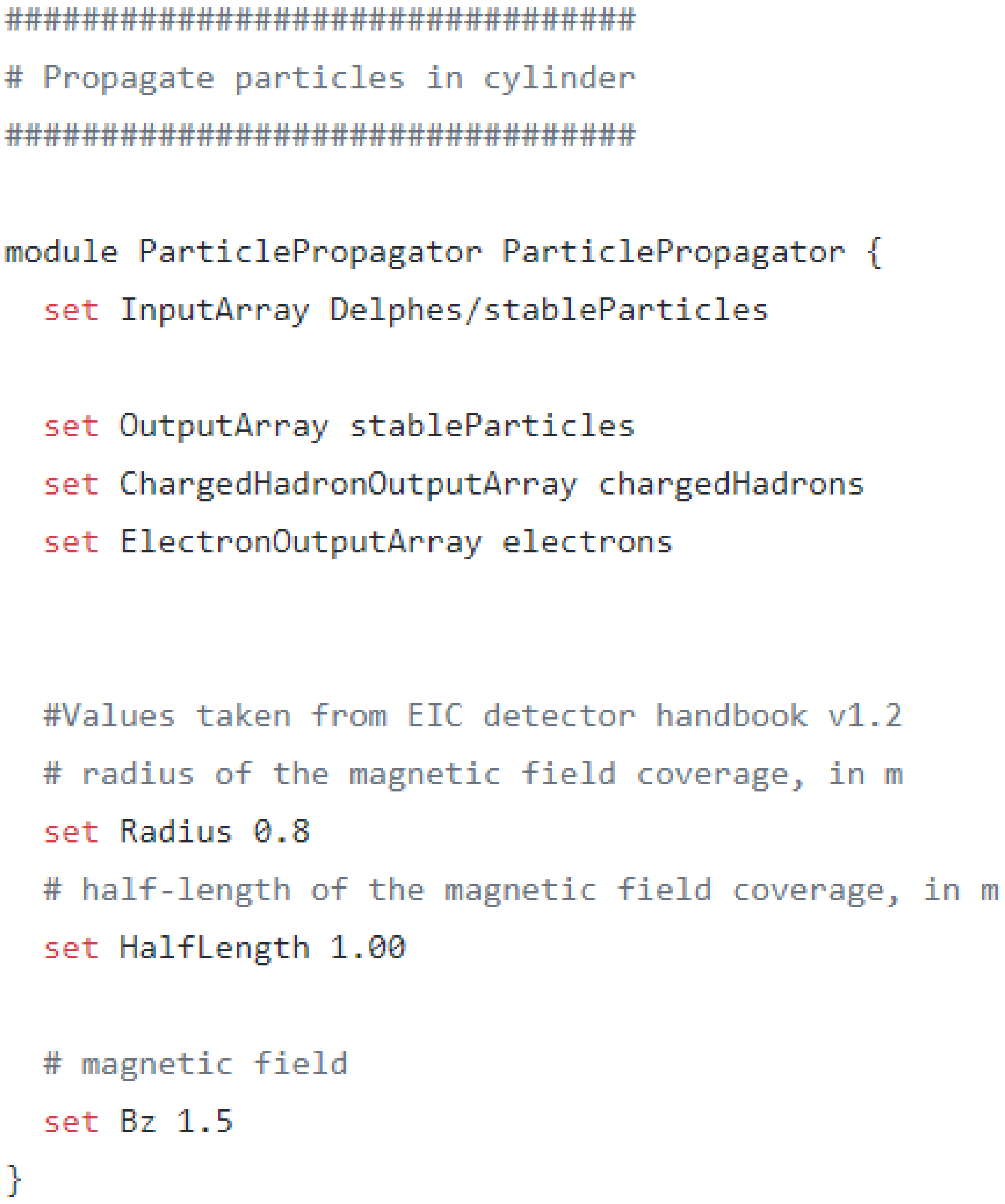


```
module MomentumSmearing ChargedHadronMomentumSmearing {
  set InputArray ChargedHadronTrackingEfficiency/chargedHadrons
  set OutputArray chargedHadrons

  # set ResolutionFormula {resolution formula as a function of eta and pt}

  # resolution formula for charged hadrons.
  # Based on EIC detector handbook v1.2
  set ResolutionFormula {                  (abs(eta) <= 1.0) * (pt > 0.1) * sqrt((5e-3)^2 + pt^2*(5e-4)^2) +
                         (abs(eta) > 1.0 && abs(eta) <= 2.5) * (pt > 0.1) * sqrt((1e-2)^2 + pt^2*(5e-4)^2) +
                         (abs(eta) > 2.5 && abs(eta) <= 3.5) * (pt > 0.1) * sqrt((2e-2)^2 + pt^2*(1e-3)^2)  }
}
```

Tracking efficiency



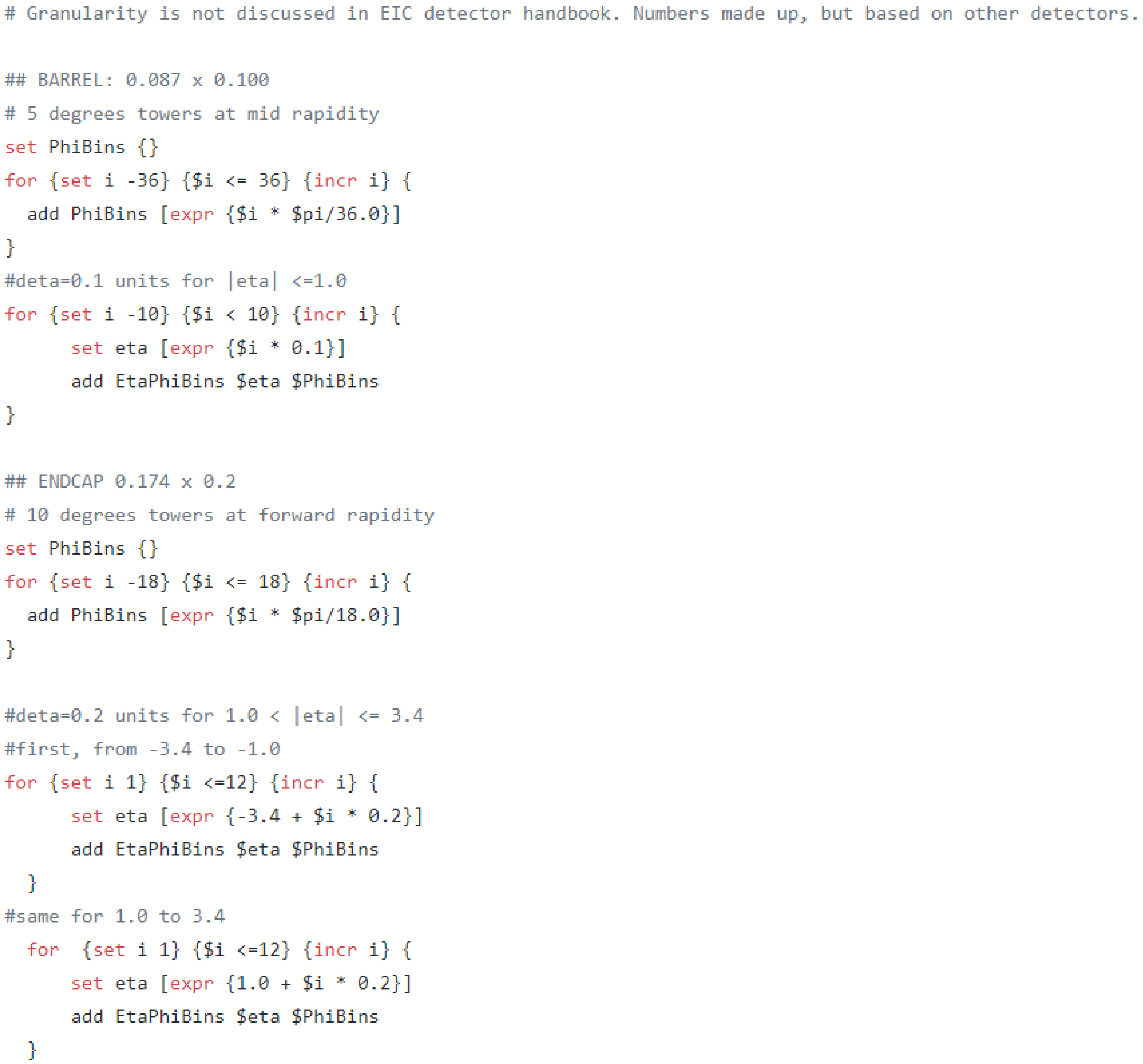
```
####################################
# Charged hadron tracking efficiency
####################################

module Efficiency ChargedHadronTrackingEfficiency {
  set InputArray ParticlePropagator/chargedHadrons
  set OutputArray chargedHadrons

  # add EfficiencyFormula {efficiency formula as a function of eta and pt}

  # tracking efficiency formula for charged hadrons
  #Made up numbers for the moment (need input from full sim)
  set EfficiencyFormula {                                                    (pt <= 0.1)   * (0.00) +
                                           (abs(eta) <= 1.5) * (pt > 0.1   && pt <= 1.0)   * (0.70) +
                                           (abs(eta) <= 1.5) * (pt > 1.0)                  * (0.95) +
                         (abs(eta) > 1.5 && abs(eta) <= 2.5) * (pt > 0.1   && pt <= 1.0)   * (0.60) +
                         (abs(eta) > 1.5 && abs(eta) <= 2.5) * (pt > 1.0)                  * (0.85) +
                         (abs(eta) > 2.5 && abs(eta) <= 3.5) * (pt > 0.1   && pt <= 1.0)   * (0.50) +
                         (abs(eta) > 2.5 && abs(eta) <= 3.5) * (pt > 1.0)                  * (0.75) +
                         (abs(eta) > 3.5)                                                  *(0.00) }
}
```



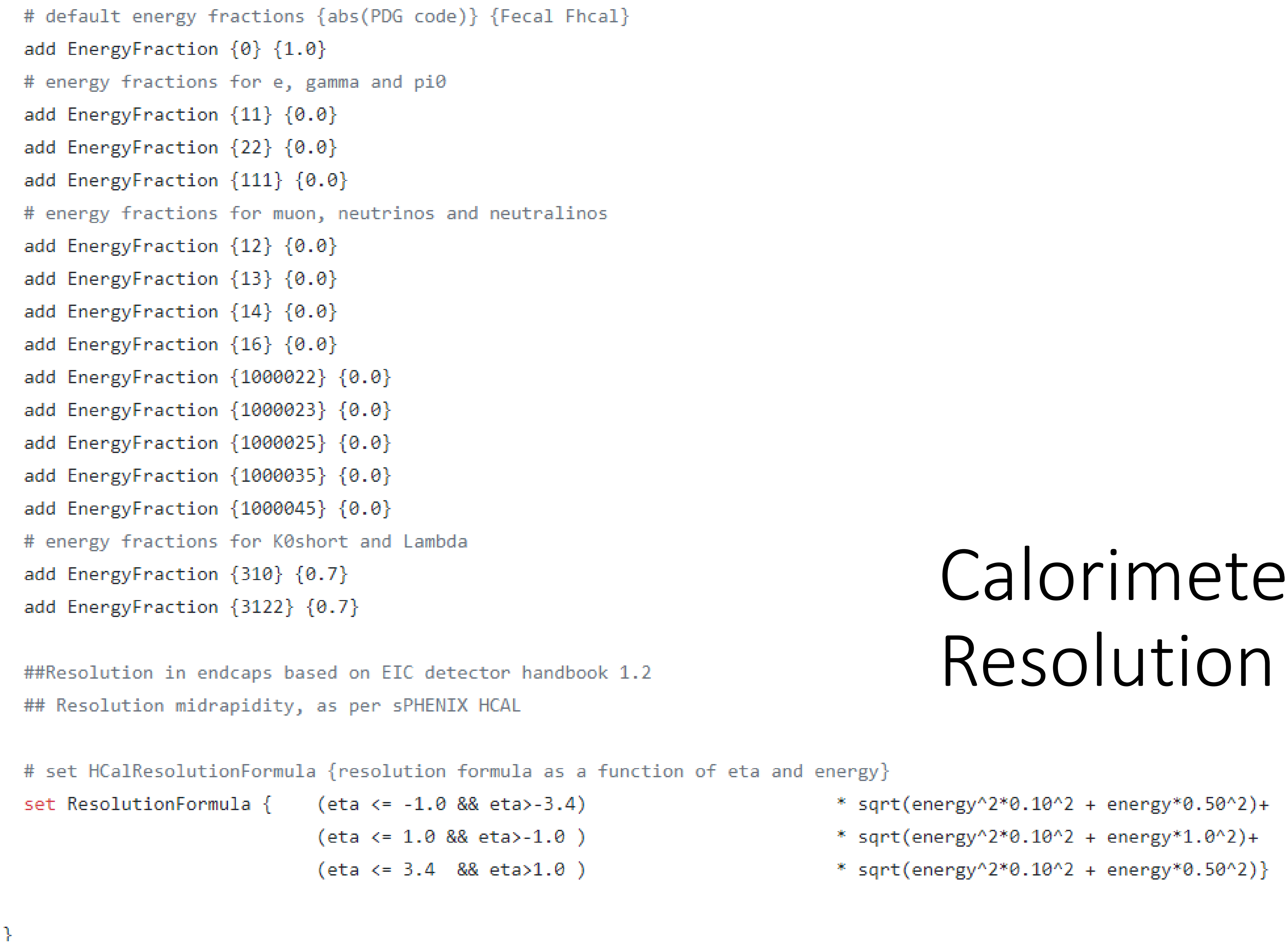
```
#################################
# Propagate particles in cylinder
#################################

module ParticlePropagator ParticlePropagator {
  set InputArray Delphes/stableParticles

  set OutputArray stableParticles
  set ChargedHadronOutputArray chargedHadrons
  set ElectronOutputArray electrons



  #Values taken from EIC detector handbook v1.2
  # radius of the magnetic field coverage, in m
  set Radius 0.8
  # half-length of the magnetic field coverage, in m
  set HalfLength 1.00

  # magnetic field
  set Bz 1.5
}
```

```
#############
#   HCAL
#############

module SimpleCalorimeter HCal {
  set ParticleInputArray ParticlePropagator/stableParticles
  set TrackInputArray ECal/eflowTracks

  set TowerOutputArray hcalTowers
  set EFlowTrackOutputArray eflowTracks
  set EFlowTowerOutputArray eflowNeutralHadrons

  set IsEcal false

  set EnergyMin 1.0
  set EnergySignificanceMin 1.0

  set SmearTowerCenter true

  set pi [expr {acos(-1)}]

  # lists of the edges of each tower in eta and phi
  # each list starts with the lower edge of the first tower
  # the list ends with the higher edged of the last tower

  # Granularity is not discussed in EIC detector handbook. Numbers made up, but based on other detectors.

  ## BARREL: 0.087 x 0.100
  # 5 degrees towers at mid rapidity
  set PhiBins {}
  for {set i -36} {$i <= 36} {incr i} {
    add PhiBins [expr {$i * $pi/36.0}]
```

Calorimeter granularity



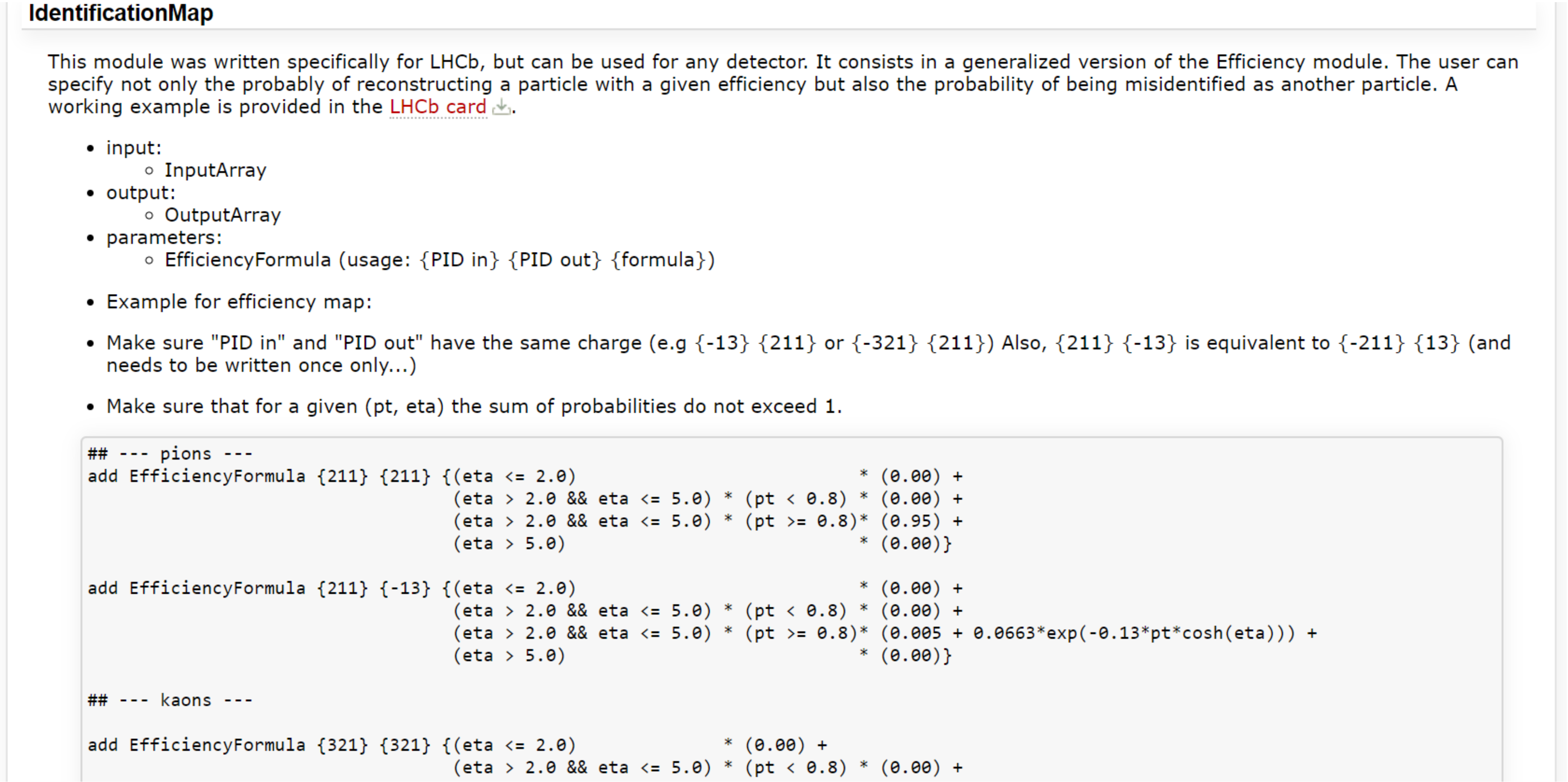
```
# Granularity is not discussed in EIC detector handbook. Numbers made up, but based on other detectors.

## BARREL: 0.087 x 0.100
# 5 degrees towers at mid rapidity
set PhiBins {}
for {set i -36} {$i <= 36} {incr i} {
  add PhiBins [expr {$i * $pi/36.0}]
}
#deta=0.1 units for |eta| <=1.0
for {set i -10} {$i < 10} {incr i} {
      set eta [expr {$i * 0.1}]
      add EtaPhiBins $eta $PhiBins
}

## ENDCAP 0.174 x 0.2
# 10 degrees towers at forward rapidity
set PhiBins {}
for {set i -18} {$i <= 18} {incr i} {
  add PhiBins [expr {$i * $pi/18.0}]
}

#deta=0.2 units for 1.0 < |eta| <= 3.4
#first, from -3.4 to -1.0
for {set i 1} {$i <=12} {incr i} {
      set eta [expr {-3.4 + $i * 0.2}]
      add EtaPhiBins $eta $PhiBins
  }
#same for 1.0 to 3.4
  for  {set i 1} {$i <=12} {incr i} {
      set eta [expr {1.0 + $i * 0.2}]
      add EtaPhiBins $eta $PhiBins
  }
```



```
# default energy fractions {abs(PDG code)} {Fecal Fhcal}
add EnergyFraction {0} {1.0}
# energy fractions for e, gamma and pi0
add EnergyFraction {11} {0.0}
add EnergyFraction {22} {0.0}
add EnergyFraction {111} {0.0}
# energy fractions for muon, neutrinos and neutralinos
add EnergyFraction {12} {0.0}
add EnergyFraction {13} {0.0}
add EnergyFraction {14} {0.0}
add EnergyFraction {16} {0.0}
add EnergyFraction {1000022} {0.0}
add EnergyFraction {1000023} {0.0}
add EnergyFraction {1000025} {0.0}
add EnergyFraction {1000035} {0.0}
add EnergyFraction {1000045} {0.0}
# energy fractions for K0short and Lambda
add EnergyFraction {310} {0.7}
add EnergyFraction {3122} {0.7}

##Resolution in endcaps based on EIC detector handbook 1.2
## Resolution midrapidity, as per sPHENIX HCAL

# set HCalResolutionFormula {resolution formula as a function of eta and energy}
set ResolutionFormula {    (eta <= -1.0 && eta>-3.4)                       * sqrt(energy^2*0.10^2 + energy*0.50^2)+
                           (eta <= 1.0 && eta>-1.0 )                       * sqrt(energy^2*0.10^2 + energy*1.0^2)+
                           (eta <= 3.4  && eta>1.0 )                       * sqrt(energy^2*0.10^2 + energy*0.50^2)}

}
```

Calorimeter Resolution

PID, which I have not implemented ….



IdentificationMap

This module was written specifically for LHCb, but can be used for any detector. It consists in a generalized version of the Efficiency module. The user can specify not only the probably of reconstructing a particle with a given efficiency but also the probability of being misidentified as another particle. A working example is provided in the LHCb card.

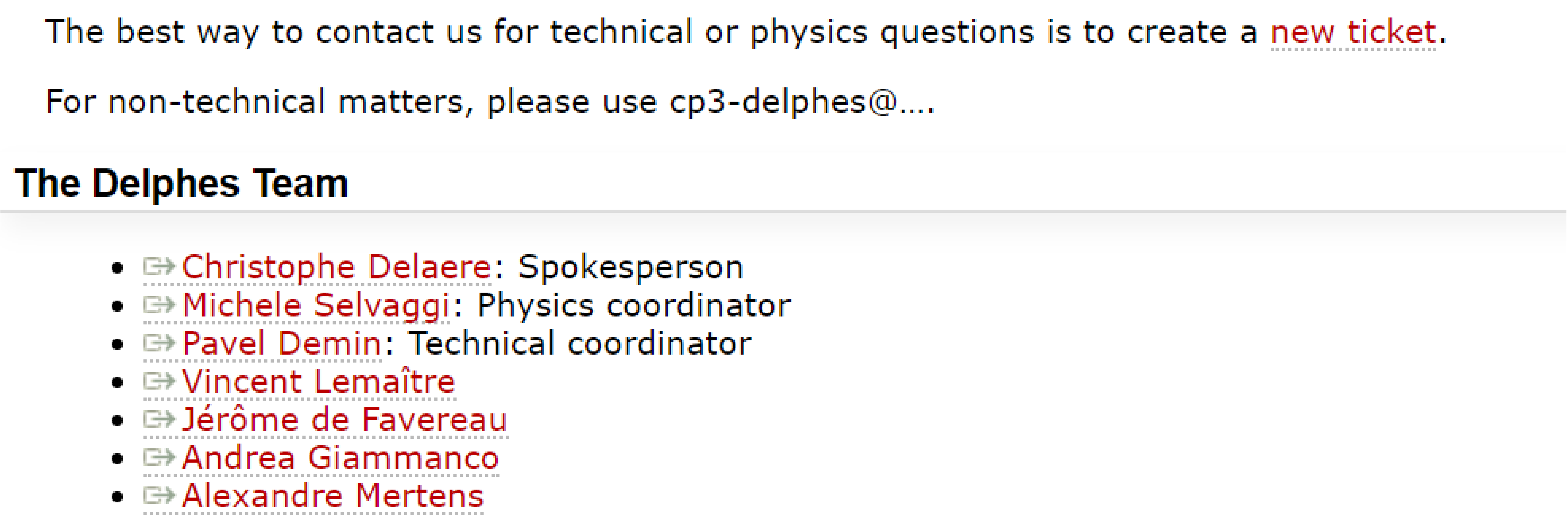
- input:
  - InputArray
- output:
  - OutputArray
- parameters:
  - EfficiencyFormula (usage: {PID in} {PID out} {formula})

- Example for efficiency map:

- Make sure "PID in" and "PID out" have the same charge (e.g {-13} {211} or {-321} {211}) Also, {211} {-13} is equivalent to {-211} {13} (and needs to be written once only...)

- Make sure that for a given (pt, eta) the sum of probabilities do not exceed 1.

```
## --- pions ---
add EfficiencyFormula {211} {211} {(eta <= 2.0)                          * (0.00) +
                                   (eta > 2.0 && eta <= 5.0) * (pt < 0.8) * (0.00) +
                                   (eta > 2.0 && eta <= 5.0) * (pt >= 0.8)* (0.95) +
                                   (eta > 5.0)                            * (0.00)}

add EfficiencyFormula {211} {-13} {(eta <= 2.0)                          * (0.00) +
                                   (eta > 2.0 && eta <= 5.0) * (pt < 0.8) * (0.00) +
                                   (eta > 2.0 && eta <= 5.0) * (pt >= 0.8)* (0.005 + 0.0663*exp(-0.13*pt*cosh(eta))) +
                                   (eta > 5.0)                            * (0.00)}

## --- kaons ---

add EfficiencyFormula {321} {321} {(eta <= 2.0)             * (0.00) +
                                   (eta > 2.0 && eta <= 5.0) * (pt < 0.8) * (0.00) +
```

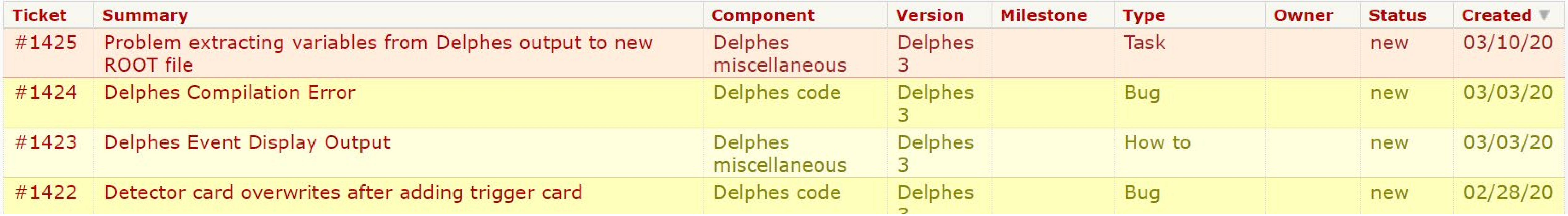


The best way to contact us for technical or physics questions is to create a new ticket.

For non-technical matters, please use cp3-delphes@....

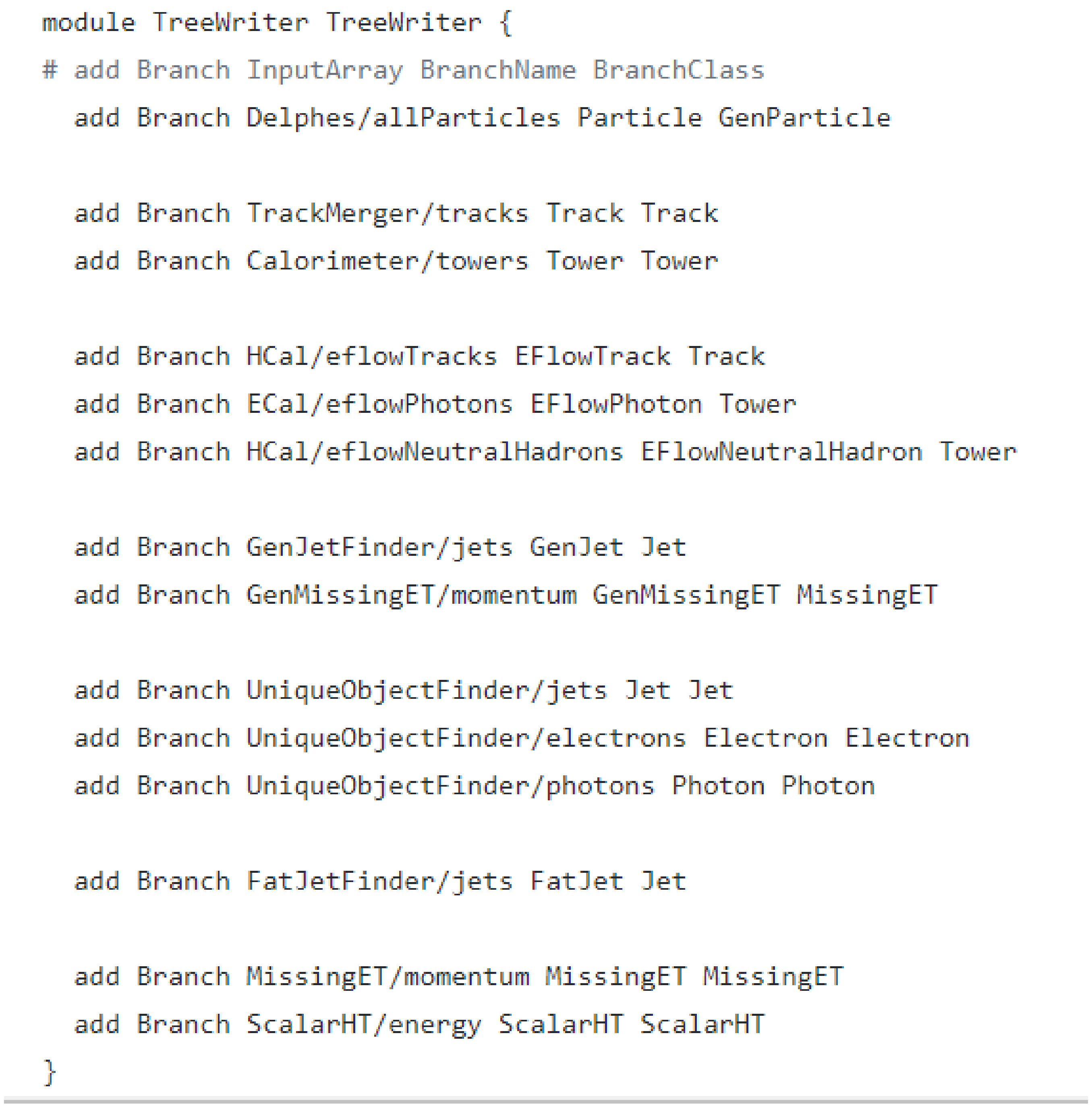
The Delphes Team

- Christophe Delaere: Spokesperson
- Michele Selvaggi: Physics coordinator
- Pavel Demin: Technical coordinator
- Vincent Lemaître
- Jérôme de Favereau
- Andrea Giammanco
- Alexandre Mertens



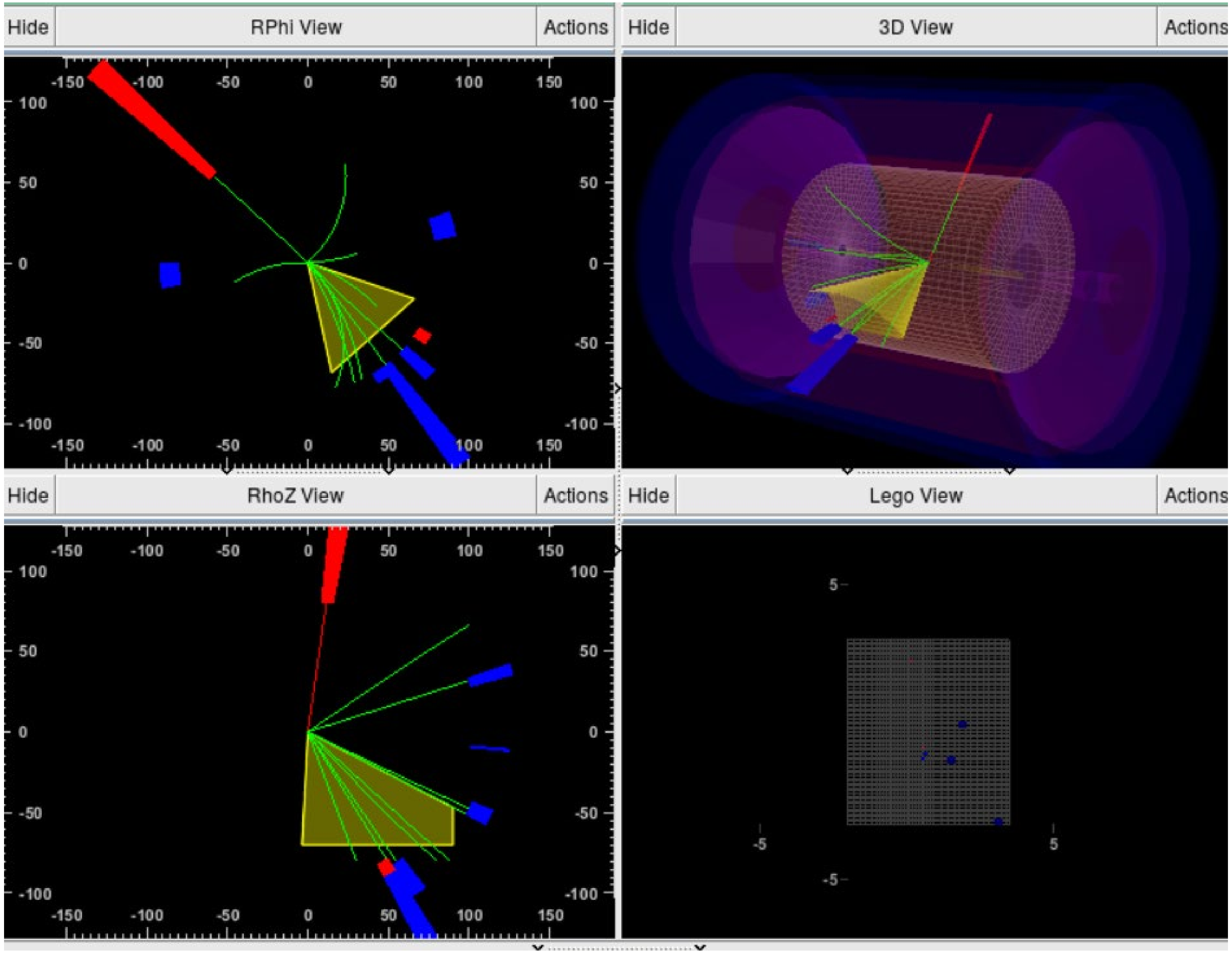
Ticket	Summary	Component	Version	Milestone	Type	Owner	Status	Created ▼
#1425	Problem extracting variables from Delphes output to new ROOT file	Delphes miscellaneous	Delphes 3		Task		new	03/10/20
#1424	Delphes Compilation Error	Delphes code	Delphes 3		Bug		new	03/03/20
#1423	Delphes Event Display Output	Delphes miscellaneous	Delphes 3		How to		new	03/03/20
#1422	Detector card overwrites after adding trigger card	Delphes code	Delphes 3		Bug		new	02/28/20

Output



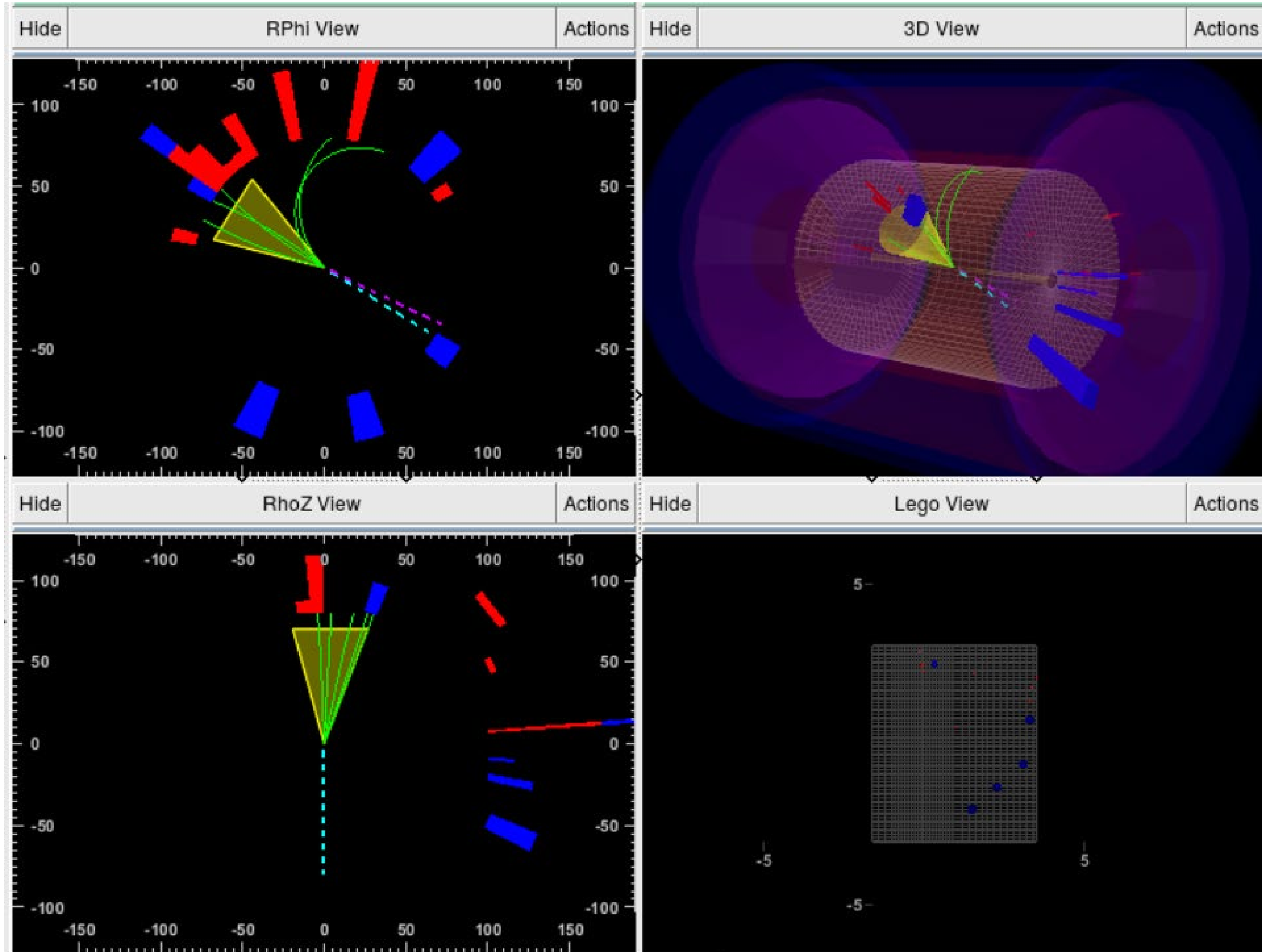
```
module TreeWriter TreeWriter {
# add Branch InputArray BranchName BranchClass
  add Branch Delphes/allParticles Particle GenParticle

  add Branch TrackMerger/tracks Track Track
  add Branch Calorimeter/towers Tower Tower

  add Branch HCal/eflowTracks EFlowTrack Track
  add Branch ECal/eflowPhotons EFlowPhoton Tower
  add Branch HCal/eflowNeutralHadrons EFlowNeutralHadron Tower

  add Branch GenJetFinder/jets GenJet Jet
  add Branch GenMissingET/momentum GenMissingET MissingET

  add Branch UniqueObjectFinder/jets Jet Jet
  add Branch UniqueObjectFinder/electrons Electron Electron
  add Branch UniqueObjectFinder/photons Photon Photon

  add Branch FatJetFinder/jets FatJet Jet

  add Branch MissingET/momentum MissingET MissingET
  add Branch ScalarHT/energy ScalarHT ScalarHT
}
```

Neutral-current interaction, 100 GeV

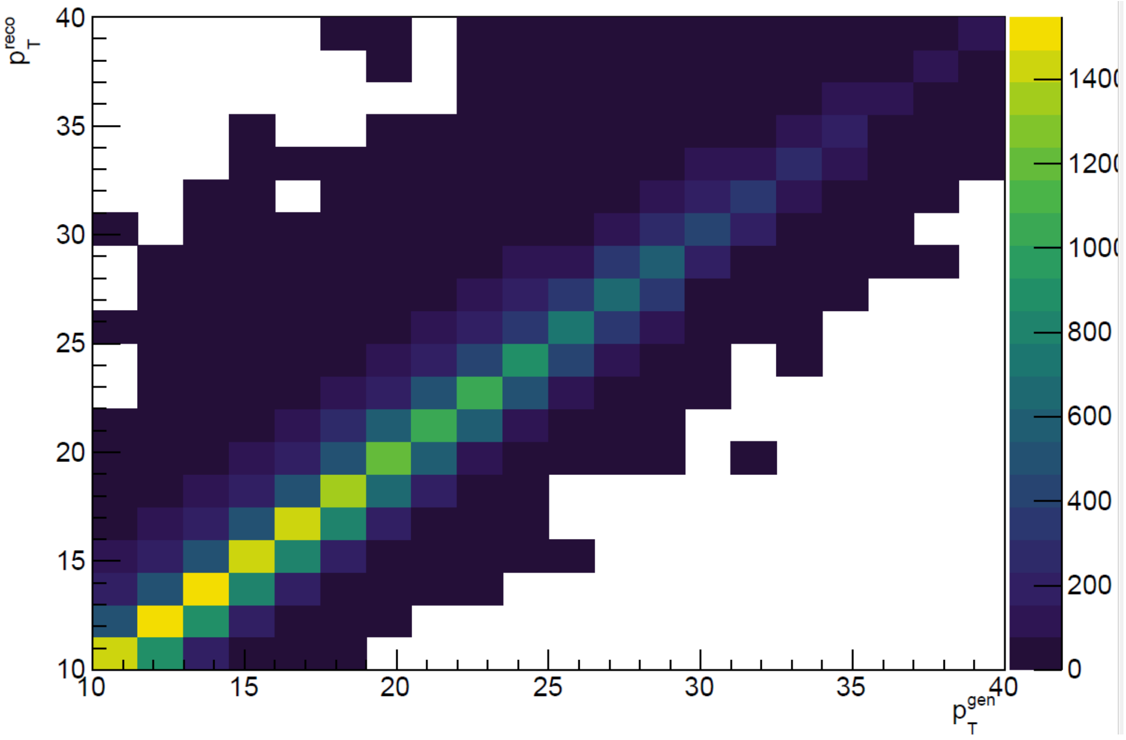


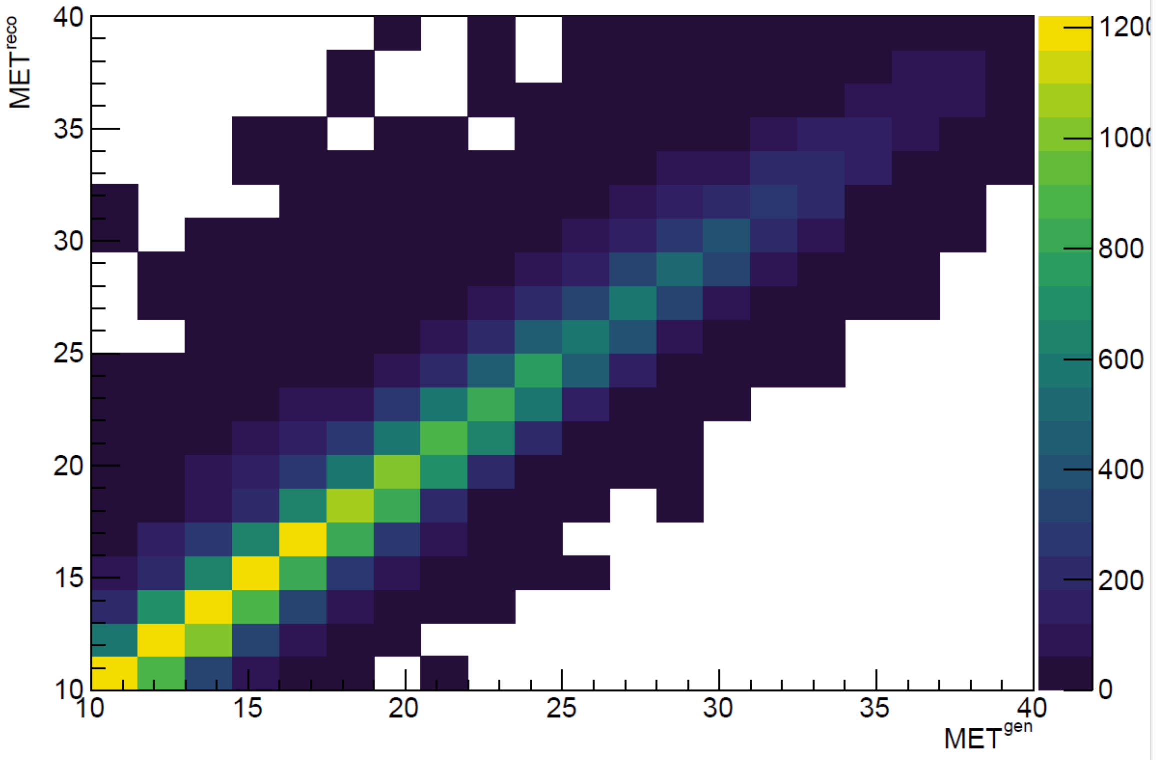
Charged-current interaction, 100 GeV

Reconstructed “missing transverse energy”

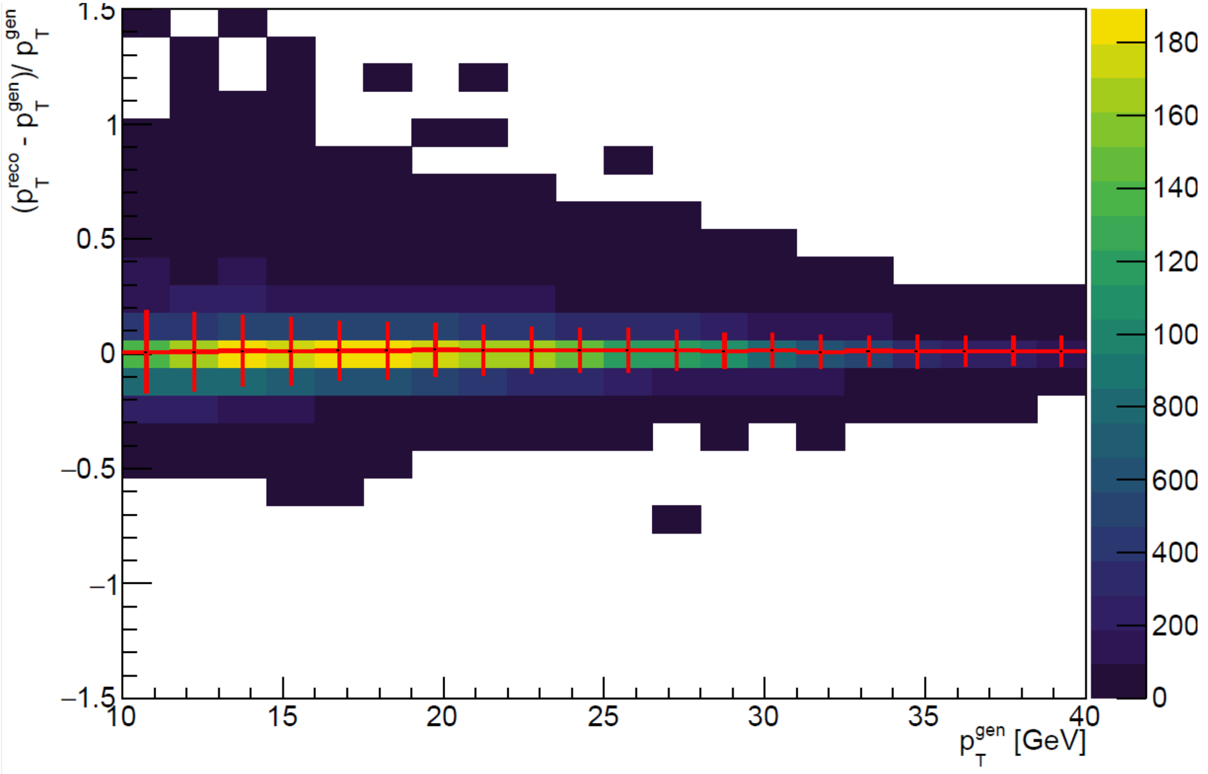


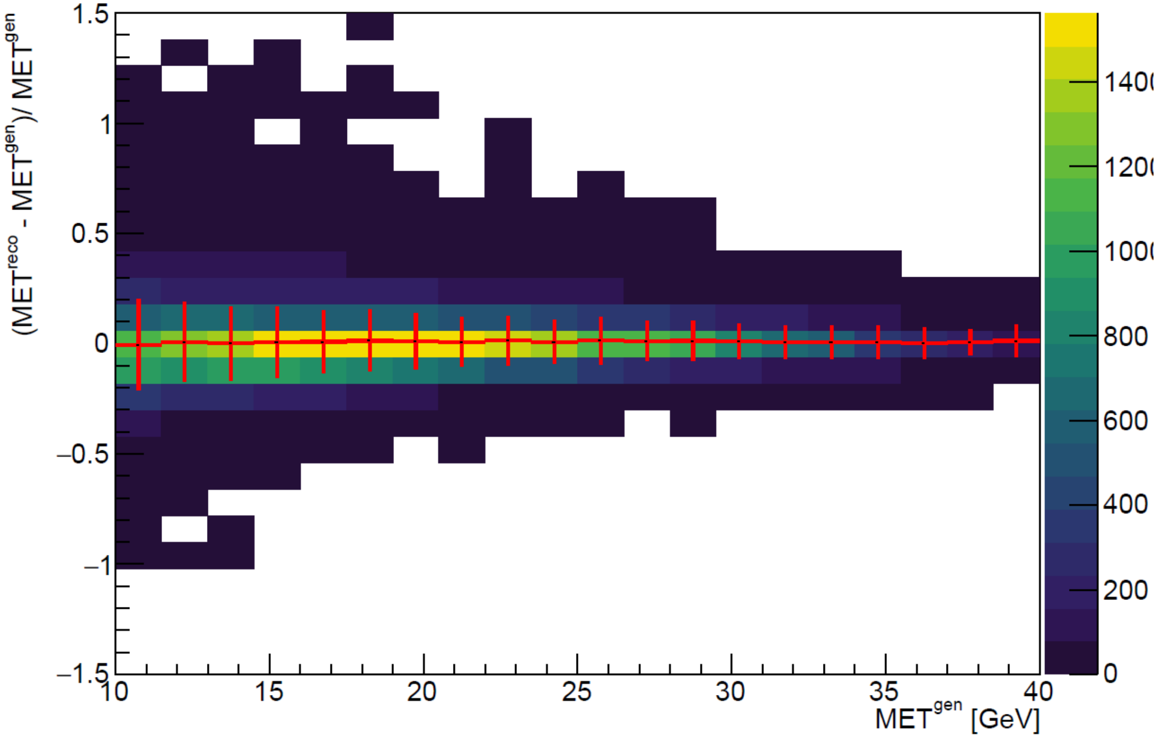
Jet/MET performance (particle-flow)



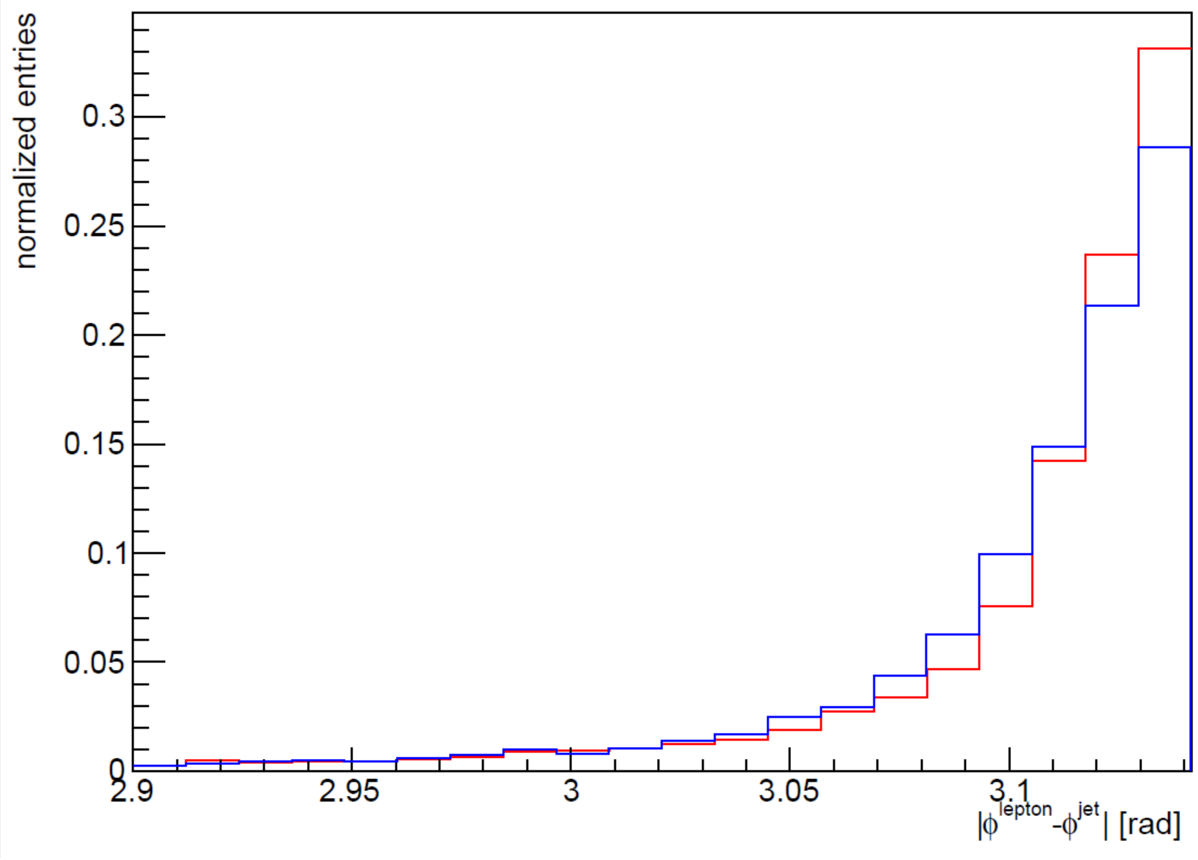


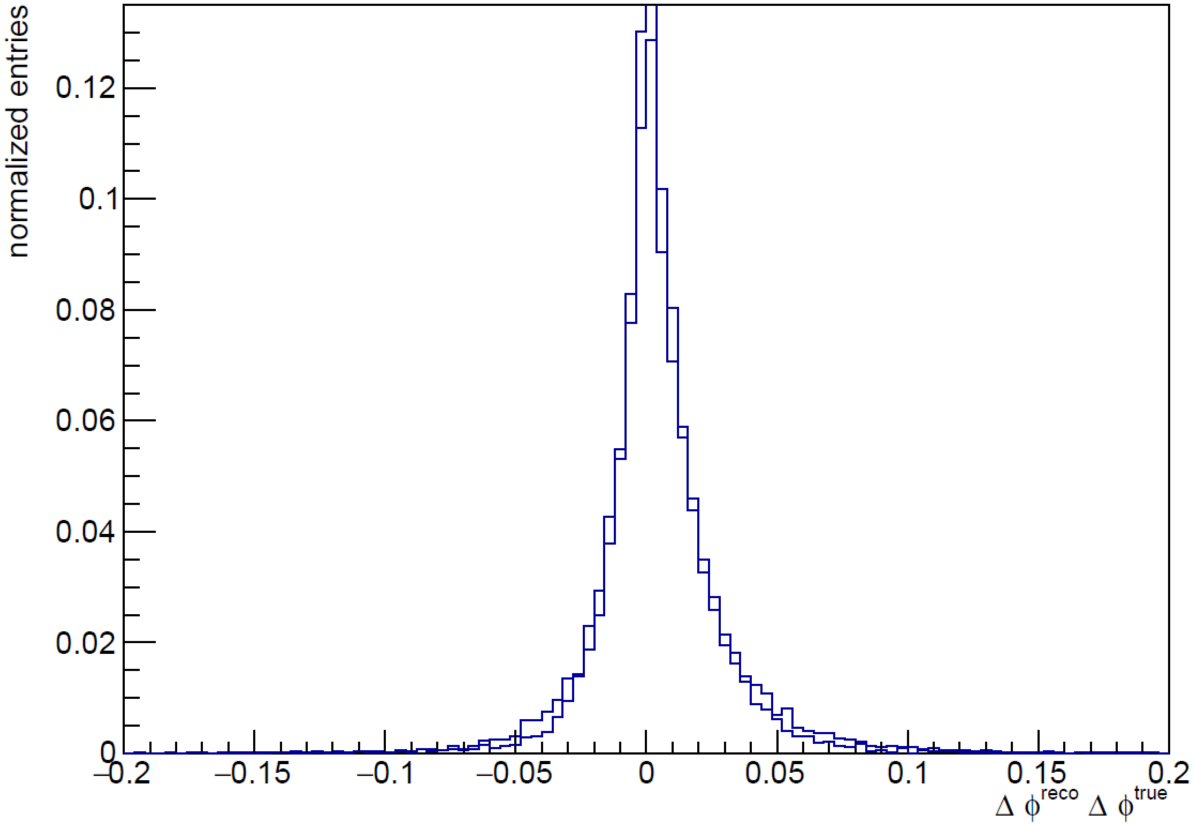
Jet/MET performance (particle-flow)



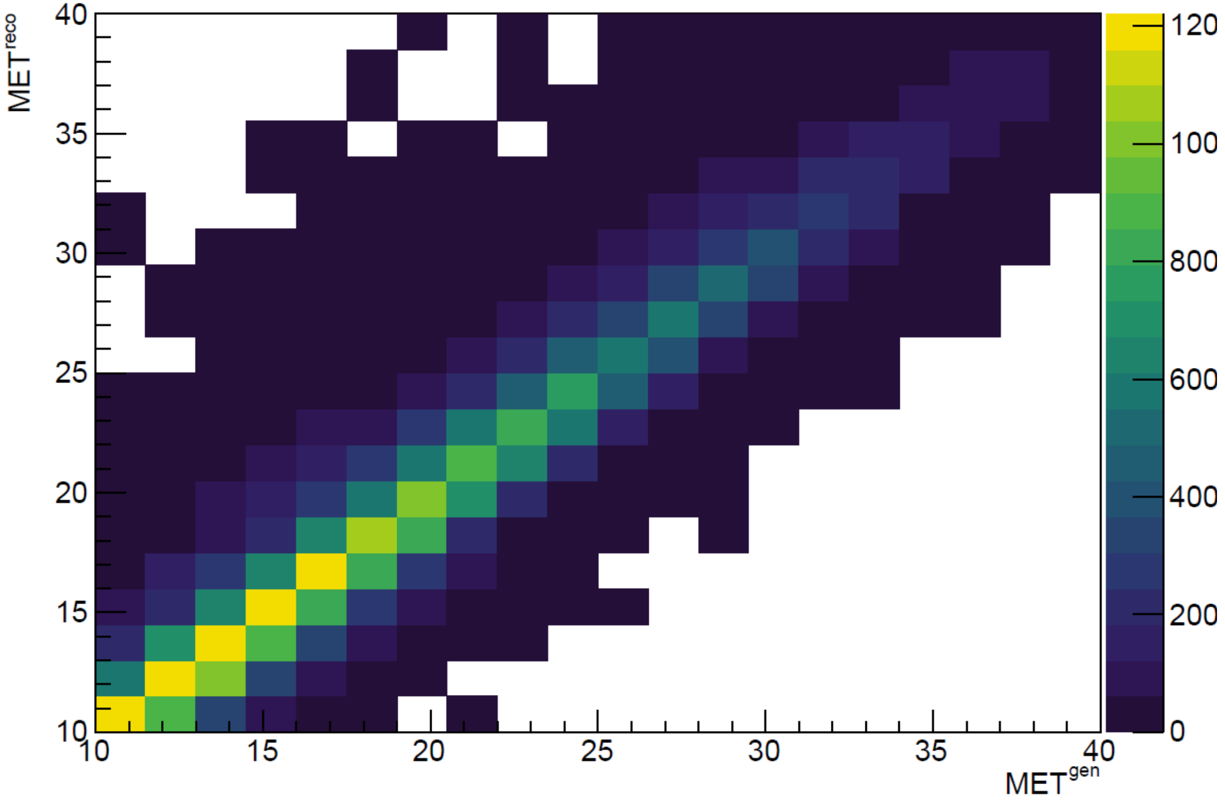


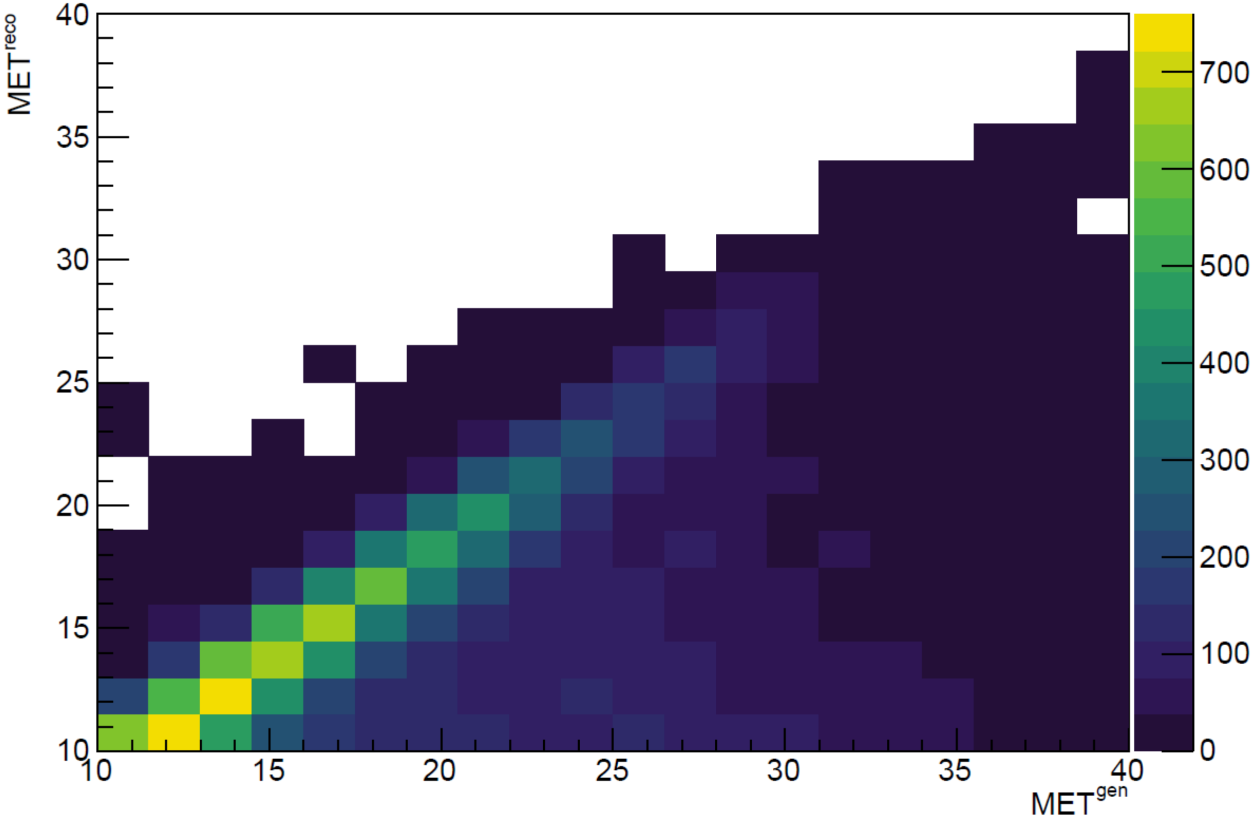
Azimuthal correlations (generated, reconstructed)





MET performance,
with and without barrel HCAL





- Delphes for EIC:
  - Why have you looked to Delphes?
    - Have you used it before? **No**
    - Has / does eic_smear not met / meet your requirements? If so, what are you missing?
      Pythia8. B-field. Efficiency losses. Granularity of calorimeters, PID.
  - Could you summarize the features of Delphes (at least the one that are relevant for you)?
    See above, and more.
- Delphes for an EIC Detector:
  - Could you give us an idea about your work to get an EIC detector into Delphes? Here, the steps would be helpful for everyone not familiar with Delphes. From your comments, it appears to be that Delphes is well-documented and easy to use. It would be interesting if you can show that in your example.
    See above and: (is very easy just pick CMS/ATLAS and change parametriations)
    https://github.com/miguelignacio/delphes_EIC/blob/master/delphes_card_EIC.tcl

- Regarding the concerns that came up in the Q&A event:
  - Do you know what Monte Carlo formats are supported? For EIC studies, we need LUND format (Pythia6 format), HepMC2 and also the BeAGLE format (which provides also information about the remnants and has its own format).
    HEPMC is supported.
  - Delphes appears to be its own framework. However, is there anything stopping us from reading in existing Monte Carlo files and steering Delphes from script? If so, the workflow would be similar to eic_smear and users could just switch between Delphes and/or eic_smear.
    I guess…
  - Do you have an idea what maintenance would be needed:
    - I would assume that we will store all parameterizations of detector responses etc. as text files. So we should be able to find a common format that works for both Delphes and eic_smear.
      I would let the user play with parametrizations and keep a website with text. We are grown ups, after all.
    - We would need a validation script to verify the fast simulation results (something also needed for eic_smear).
      …
    - Whom could we ask for help if needed (assuming that none of us is a Delphes developer)?
      Suggestion: keep centralized text with parametrizations, then let users pick and chose. Then work would be minimal. Then force users to give the parametrization card for reproducibility.