CUPS – CMS Upgrade School
17-21 Nov. 2014

DELPHES:

A modular framework for fast simulation of a generic collider experiment

Lisa Borgonovi

Outline

- Introduction to Delphes
    - What is Delphes and why do we use it?
    - Structure of the simulated detector
    - Object reconstruction
    - Delphes for CMS
    - To sum up
- Getting started with Delphes
    - Delphes DataCard

Introduction to Delphes

Input: output of event generators (MadGraph, PYTHIA, HERWIG, ...)

DELPHES

C++ modular framework which performs a fast and parametrized multipurpose detector response simulation

- Tracking system embedded in a magnetic field
- Electromagnetic and hadronic calorimeters
- Muon identification system

Recostruction of the physics objects (γ, $e^{\pm}$, $\mu^{\pm}$, $\tau^{\pm}$, jets, b-jets, MET, ...)

Output: ROOT ntuple (generated and recontstructed objects)

What is Delphes and why do we use it?

Multipurpose detectors are complex and sophisticated

To tune and optimize specific analysis, a **full detector simulation** is requested, with a high level of accuracy:
Large scale computing resources and high level expertise

Phenomenological studies, comparison between different ugraded or aged scenarios, big background samples production require: less precision but a **FASTER** and **SIMPLER** tool

↓

DELPHES

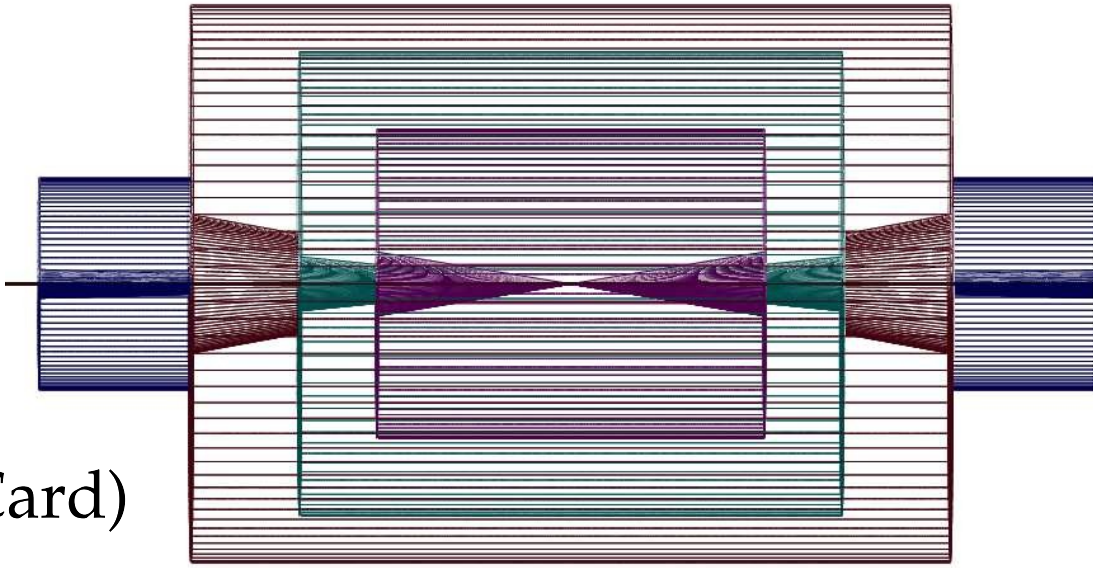
Introduction to Delphes

Structure of the simulated detector

Limitations:

- Detector geometry: uniform, symmetric around the beam axis, without cracks or dead material
- Secondary interactions, multiple scattering, photon conversion and bremmstrahlung are not simulated

The overall layout of a multipurpose detector is simulated thanks to a parametrization (in a DataCard)



Introduction to Delphes
Structure of the simulated detector

- Central tracking system

Propagation of the long-lived particles in a uniform axial magnetic field along the beam direction

- Particles that originate outside the tracker volume are neglected
- NO smearing applied on track parameters (exept for Pt)

- Electromagnetic and hadronic calorimeters

ECAL	HCAL
$f_{ECAL}(e^{\pm}, \gamma) = 1$	$f_{HCAL}(e^{\pm}, \gamma) = 0$
$f_{ECAL}(h) = 0$	$f_{HCAL}(h) = 1$
$f_{ECAL}(K, \Lambda) = 0.3$	$f_{HCAL}(K, \Lambda) = 0.7$
$f_{ECAL}(\mu^{\pm}, \nu) = 0$	$f_{HCAL}(\mu^{\pm}, \nu) = 0$

Introduction to Delphes
Object reconstruction

- $e^{\pm}$, $\mu^{\pm}$, γ ➡ quite easy to reconstruct
- Jets, Missing Transverse Energy ➡ difficult to reconstruct

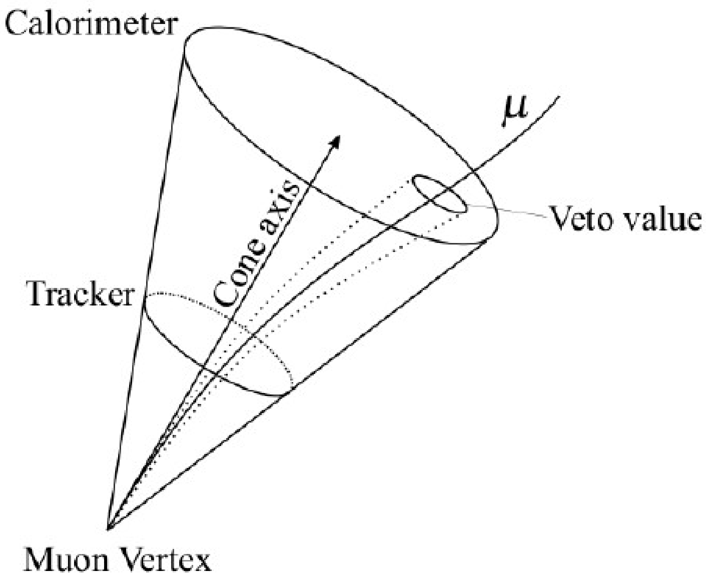
Accessible in the output:
- 4-momentum and related quantities
- Additional specific properties (e.g. Isolation)

Object reconstruction

- Photons and charged leptons (NO τ!)

Only the final-state particles identificated via generator-data and passing cuts (tracker acceptance and p_T threshold)

- Reconstructed with a user-defined Efficiency: $f(\eta, p_T)$
- NO FAKE candidates (they can be added in the analysis)



Isolation:

$e^{\pm}$, $\mu^{\pm}$ isolated if $I < I_{min}$ within a small cone of radius ΔR in the (η,φ) plane

Delphes for CMS

Many changes from the “official” Delphes:

- Timing information
- Different calorimeter configurations
- η dipendent ρ correction

In general: in CMS version there are more and different modules from the official one, to adjust the simulation to the CMS detector

This is the version we are going to use!

To sum up

Delphes is a modular framework that performs a fast and parametrized simulation of a multipurpose detector (e.g. CMS)

This tool has been completely validated (with CMS and ATLAS real data)

Useful in phenomenological studies and to produce big size bakground samples in which a fast simulation is needed

Getting started with Delphes

1) Set the right Scientific Linux CERN version

2) Set the CMS environment

3) Download and install Delphes

4) You need one or more file(s) with Generated Events
(in .hepMC format or others)

5) If you run with PU you need a file with Minimum Bias Events
(MinBias.pileup)

6) Delphes creates a directory “Cards” with DataCards inside

To Run Delphes:

./DelphesHepMC Cards/cardname.tcl path/outputfilename.root path/hepmcfilename.hepmc

Getting started with Delphes

Delphes DataCard

File with ALL the settings of the simulation: efficiencies, resolutions, parameters for isolation, jets, …

Structure:

- Function with all the list of instructions (different modules called one after the other)
- Descriptions of all the modules (Muon Momentum Smearing, Muon efficiency, Electron Energy Resolution, Muon Isolation, …)

⬇

Last Module: TreeWriter

(choice of the Branches to write in the ROOT Tree)

Execution Path

```
#######################################
# Order of execution of various modules
#######################################

set ExecutionPath {

  PileUpMerger
  ModifyBeamSpot
  ParticlePropagator
  StatusPid
  GenBeamSpotFilter

  ChargedHadronTrackingEfficiency
  ElectronTrackingEfficiency
  MuonTrackingEfficiency

  ChargedHadronMomentumSmearing
  ElectronEnergySmearing
  MuonMomentumSmearing

  ModifyBeamSpotNoPU
  ParticlePropagatorNoPU
  ChargedHadronTrackingEfficiencyNoPU
  ElectronTrackingEfficiencyNoPU
  MuonTrackingEfficiencyNoPU
  ChargedHadronMomentumSmearingNoPU
  ElectronEnergySmearingNoPU
  MuonMomentumSmearingNoPU
  TrackMergerNoPU
  CalorimeterNoPU
  EFlowMergerNoPU
  FastJetFinderNoPU

  TrackMerger
  Calorimeter
  TrackPileUpSubtractor
  EFlowMerger

  GlobalRho
  Rho
  FastJetFinder
  GenJetFinder
  JetPileUpSubtractor

  NeutrinoFilter
  GenJetFinderNoNu
  GenMissingET

  EFlowChargedMerger
  RunPUPPI
  PuppiJetFinder
  PuppiRho
  PuppiJetPileUpSubtractor
  PuppiMissingET

  PhotonEfficiency
  PhotonIsolation

  ElectronEfficiency
  ElectronIsolation

  MuonEfficiency
  MuonIsolation

  MissingET

  BTagging
  BTaggingLoose
  TauTagging

  TrackPVSubtractor
  IsoTrackFilter

  UniqueObjectFinderGJ
  UniqueObjectFinderEJ
  UniqueObjectFinderMJ

  ScalarHT

  PileUpJetID

  PileUpJetIDMissingET

  ConstituentFilter
  TreeWriter
}
```

Resolution

```
File  Edit  Options  Buffers  Tools  Tcl  Help

################################
# Momentum resolution for muons
################################

module MomentumSmearing MuonMomentumSmearing {
  set InputArray MuonTrackingEfficiency/muons
  set OutputArray muons

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

  # resolution formula for muons

    set ResolutionFormula {


(pt > 0.0) * ( abs(eta) > 0 && abs(eta) <= 0.1 ) * ( 0.00941022 ) + \
(pt > 0.0) * ( abs(eta) > 0.1 && abs(eta) <= 0.2 ) * ( 0.00958603 ) + \
(pt > 0.0) * ( abs(eta) > 0.2 && abs(eta) <= 0.3 ) * ( 0.0100961 ) + \
(pt > 0.0) * ( abs(eta) > 0.3 && abs(eta) <= 0.4 ) * ( 0.0108526 ) + \
(pt > 0.0) * ( abs(eta) > 0.4 && abs(eta) <= 0.5 ) * ( 0.0112063 ) + \
(pt > 0.0) * ( abs(eta) > 0.5 && abs(eta) <= 0.6 ) * ( 0.0114379 ) + \
(pt > 0.0) * ( abs(eta) > 0.6 && abs(eta) <= 0.7 ) * ( 0.0118331 ) + \
(pt > 0.0) * ( abs(eta) > 0.7 && abs(eta) <= 0.8 ) * ( 0.0124008 ) + \
(pt > 0.0) * ( abs(eta) > 0.8 && abs(eta) <= 0.9 ) * ( 0.0136308 ) + \
(pt > 0.0) * ( abs(eta) > 0.9 && abs(eta) <= 1 ) * ( 0.0159186 ) + \
(pt > 0.0) * ( abs(eta) > 1 && abs(eta) <= 1.1 ) * ( 0.0168638 ) + \
(pt > 0.0) * ( abs(eta) > 1.1 && abs(eta) <= 1.2 ) * ( 0.0177207 ) + \
(pt > 0.0) * ( abs(eta) > 1.2 && abs(eta) <= 1.3 ) * ( 0.0186257 ) + \
(pt > 0.0) * ( abs(eta) > 1.3 && abs(eta) <= 1.4 ) * ( 0.0192152 ) + \
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)
(pt > 0.0)


}
}
```

```
#################################
# Energy resolution for electrons
#################################

module EnergySmearing ElectronEnergySmearing {
  set InputArray ElectronTrackingEfficiency/electrons
  set OutputArray electrons


    set ResolutionFormula {            (abs(eta) <= 2.5) * (energy > 0.1   && energy <= 2.5e1) * (energy*0.025) + \
                                       (abs(eta) <= 2.5) * (energy > 2.5e1)                    * (energy*0.035) + \
                                       (abs(eta) > 2.5 && abs(eta) <= 3.0)                     * (energy*0.035) + \
                                       (abs(eta) > 3.0 && abs(eta) <= 5.0)                     * (energy*0.07)}

}
```

Muon Efficiency

```
#################
# Muon efficiency
#################

module Efficiency MuonEfficiency {
  set InputArray TrackPileUpSubtractor/muons
  set OutputArray muons

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

  # efficiency formula for muons


    set EfficiencyFormula {

(pt < 5.0) * (0.00) + \
(pt > 5.0 && pt <= 10.0) * ( abs(eta) > 0.0 && abs(eta) <= 0.5 ) * ( 0.98
(pt > 5.0 && pt <= 10.0) * ( abs(eta) > 0.5 && abs(eta) <= 1.0 ) * ( 0.98
(pt > 5.0 && pt <= 10.0) * ( abs(eta) > 1.0 && abs(eta) <= 1.5 ) * ( 0.97
(pt > 5.0 && pt <= 10.0) * ( abs(eta) > 1.5 && abs(eta) <= 2.0 ) * ( 0.99
(pt > 5.0 && pt <= 10.0) * ( abs(eta) > 2 && abs(eta) <= 2.4 ) * ( 0.9714
(pt > 5.0 && pt <= 10.0) * ( abs(eta) > 2.4  ) * ( 0 ) + \
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 0.0 && abs(eta) <= 0.5 ) * ( 0.9
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 0.5 && abs(eta) <= 0.9 ) * ( 0.9
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 0.9 && abs(eta) <= 1.1 ) * ( 0.9
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 1.1 && abs(eta) <= 1.4 ) * ( 0.9
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 1.4 && abs(eta) <= 2.0 ) * ( 0.9
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 2.0 && abs(eta) <= 2.4 ) * ( 0.9
(pt > 10.0 && pt <= 20.0) * ( abs(eta) > 2.4 ) * ( 0 ) + \
(pt > 20.0 && pt <= 40.0) * ( abs(eta) > 0.0 && abs(eta) <= 0.5 ) * ( 0.9
(pt > 20.0 && pt <= 40.0) * ( abs(eta) > 0.5 && abs(eta) <= 1.0 ) * ( 0.9
(pt > 20.0 && pt <= 40.0) * ( abs(eta) > 1.0 && abs(eta) <= 1.5 ) * ( 0.9
(pt > 20.0 && pt <= 40.0) * ( abs(eta) > 1.5 && abs(eta) <= 2.0 ) * ( 0.9
(pt > 20.0 && pt <= 40.0) * ( abs(eta) > 2.0 && abs(eta) <= 2.4 ) * ( 0.9
(pt > 20.0 && pt <= 40.0) * ( abs(eta) > 2.4 ) * ( 0) + \
(pt > 40.0 ) * ( abs(eta) > 0.0 && abs(eta) <= 0.5 ) * ( 0.9949498) + \
(pt > 40.0 ) * ( abs(eta) > 0.5 && abs(eta) <= 1.0 ) * ( 0.9952686) + \
(pt > 40.0 ) * ( abs(eta) > 1.0 && abs(eta) <= 1.5 ) * ( 0.9886918) + \
(pt > 40.0 ) * ( abs(eta) > 1.5 && abs(eta) <= 2.0 ) * ( 0.9919946) + \
(pt > 40.0 ) * ( abs(eta) > 2.0 && abs(eta) <= 2.4 ) * ( 0.964623) + \
(pt > 40.0) * ( abs(eta) > 2.4 ) * ( 0)
```

Muon Isolation

```
################
# Muon isolation
################


module Isolation MuonIsolation {
  set CandidateInputArray MuonEfficiency/muons
  set NeutralIsolationInputArray Calorimeter/eflowTowers
  set ChargedIsolationInputArray TrackPileUpSubtractor/eflowTrac
  set RhoInputArray Rho/rho

  set OutputArray muons

  set DeltaRMax 0.4

  set PTMin 0.4

  set PTRatioMax 9999.
}

###################
```

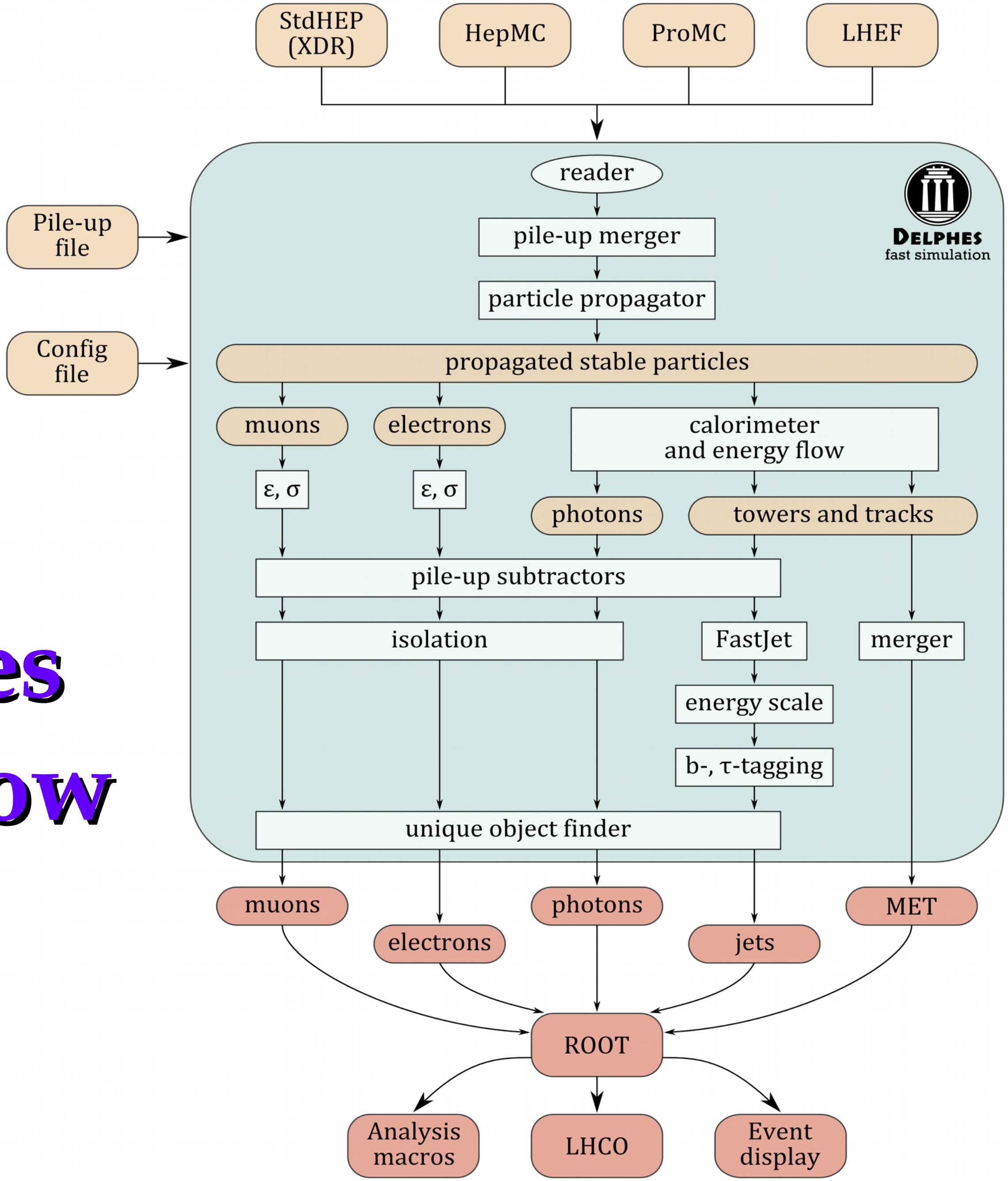
Root Tree Writer

```
}

##################
# ROOT tree writer
##################

module TreeWriter TreeWriter {
 add Branch StatusPid/filteredParticles Particle GenParticle
 ### add Branch Delphes/allParticles Particle GenParticle



  add Branch GenBeamSpotFilter/beamSpotParticles BeamSpotParticle GenParticle

  add Branch FastJetFinder/jets RawJet Jet
  add Branch FastJetFinderNoPU/jets RawJetNoPU Jet

  add Branch GenJetFinder/jets GenJetWithNu Jet
  add Branch GenJetFinderNoNu/jets GenJet Jet
#  add Branch UniqueObjectFinderMJ/jets Jet Jet
#  add Branch UniqueObjectFinderEJ/electrons Electron Electron
#  add Branch UniqueObjectFinderGJ/photons Photon Photon
#  add Branch UniqueObjectFinderMJ/muons Muon Muon
  add Branch JetPileUpSubtractor/jets Jet Jet
  add Branch ElectronIsolation/electrons Electron Electron
  add Branch PhotonIsolation/photons Photon Photon
  add Branch MuonIsolation/muons Muon Muon

  add Branch PileUpJetIDMissingET/momentum PileUpJetIDMissingET MissingET
  add Branch GenMissingET/momentum GenMissingET MissingET
  add Branch PuppiMissingET/momentum PuppiMissingET MissingET


  add Branch MissingET/momentum MissingET MissingET
  add Branch ScalarHT/energy ScalarHT ScalarHT
  add Branch Rho/rho Rho Rho
  add Branch GlobalRho/rho GlobalRho Rho
  add Branch PileUpMerger/NPU NPU ScalarHT
  add Branch IsoTrackFilter/IsoTrack IsoTrack IsoTrack

  add Branch PuppiJetFinder/jets PuppiJet Jet

  set OffsetFromModifyBeamSpot 0

#  add Branch RunPUPPI/weightedparticles PuppiWeightedParticles GenParticle
#  add Branch Delphes/allParticles Particle GenParticle
--:**  CMS_PhaseI_PU50_muPOGeff.tcl      (Tcl)--L1346--95%-------------------------------------
```

Thank you for the attention!

Backup

Delphes work-flow chart



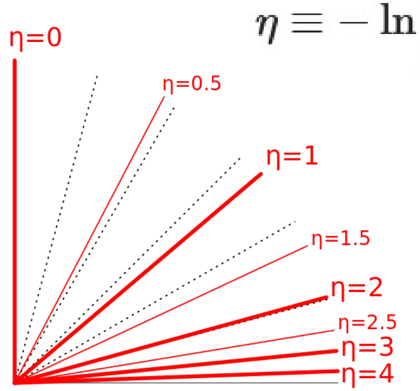
Introduction to Delphes

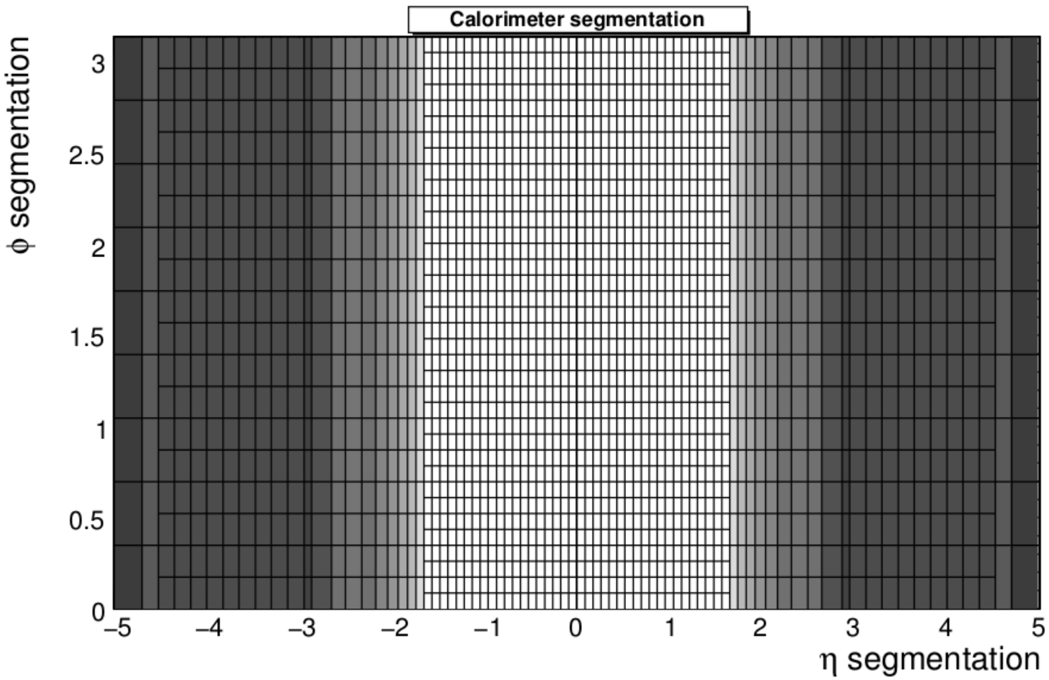
Structure of the simulated detector

- Electromagnetic and hadronic calorimeters

Segmented in the (η,φ) plane in user-defined size cells

$$\eta \equiv -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$$





- Same granularity for ECAL and HCAL
- Uniform segmentation in φ
- Response simulated through a Gaussian smearing of the cell energy

Introduction to Delphes
Object reconstruction

- Photon and charged leptons (NO !)

$\mu^{\pm}$ Reconstructed thanks to user-defined efficiencies: f(η, p_T)

p_T: Gaussian smearing of the initial 4-momentum vector with resolution as a function of eta and p_T

$e^{\pm}$ Reconstructed with user-defined efficiencies (ECAL + tracker recontruction): f(η, p_T)

Energy Resolution: f(TRACKER Res, ECAL Res)

γ Reconstructed only by the ECAL energy deposit

Pairs production neglected

$e^{\pm}$ with no reconstructed tracks that reach the ECAL

Object reconstruction

b-jets and τ-jets

Parametric approach:

A jet is a potential b/τ jet if a generated b/τ is found within some distance:

$$\Delta R = \sqrt{(\eta^{jet} - \eta^{b,\tau})^2 + (\phi^{jet} - \phi^{b,\tau})^2}$$

along the jet axis

User-defined efficiency to identify that candidate as a real b/τ jet intervenes (Mistagging efficiency can be specified)

Introduction to Delphes

Object reconstruction

- Jets (b/τ jets)

Final-states dominated by Jets: need an accurate reconstruction

It is possible to produce Jets starting from different input collections:

- Generated Jets: clustered from generator-level particles after parton-shower and hadronization
- Calorimeter Jets: use calorimeter towers
- Energy-flow Jets: result of clustering energy-flow tracks and towers

Many clustering algorithms (Longitudinally invariant k_t jet, Cambridge/Aachen jet, Anti k_t jet, CDF MidPoint, ...)