Delphes

framework for fast simulation of a generic collider experiment

Jérôme de Favereau, Pavel Demin

Université catholique de Louvain
Louvain-la-Neuve, Belgium

Center for Particle Physics and Phenomenology (CP3)

June 11, 2012

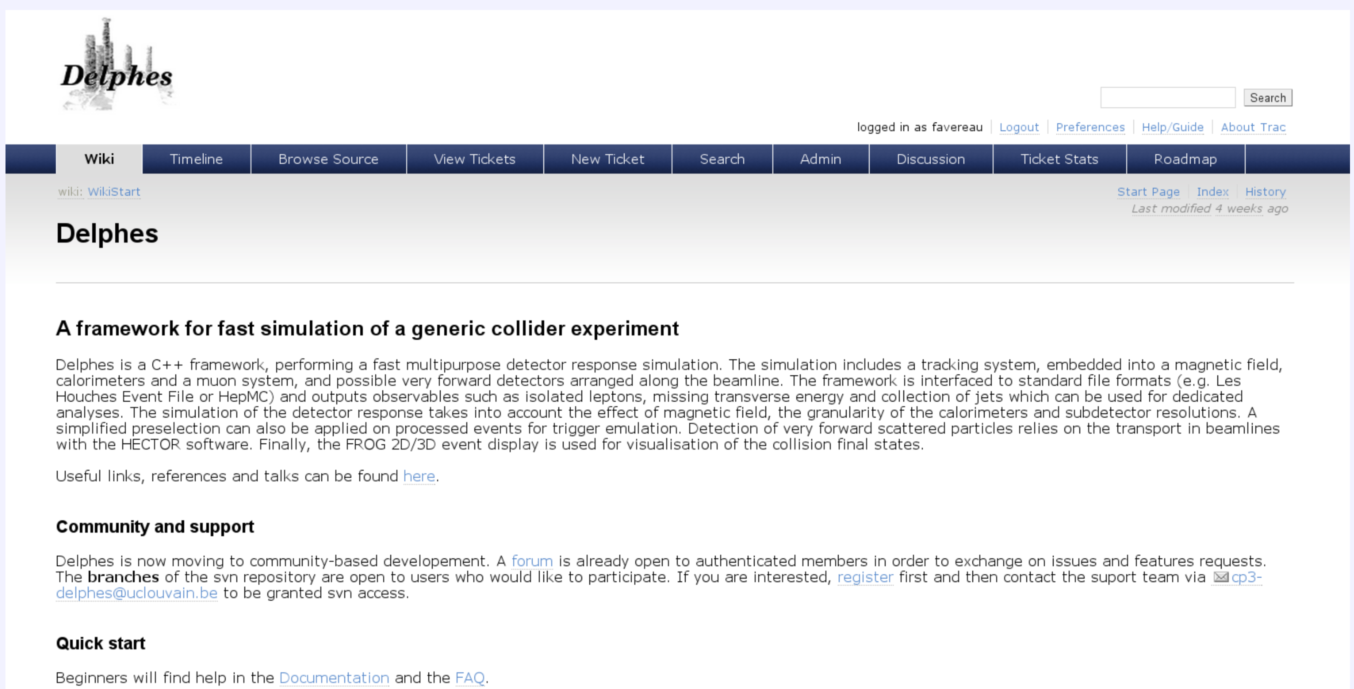
References

- Website:

  https://server06.fynu.ucl.ac.be/projects/delphes

- Paper and User manual:

  arXiv:0903.2225 [hep-ph]



Delphes

Search

logged in as favereau | Logout | Preferences | Help/Guide | About Trac

Wiki | Timeline | Browse Source | View Tickets | New Ticket | Search | Admin | Discussion | Ticket Stats | Roadmap

wiki: WikiStart

Start Page | Index | History

Last modified 4 weeks ago

Delphes

A framework for fast simulation of a generic collider experiment

Delphes is a C++ framework, performing a fast multipurpose detector response simulation. The simulation includes a tracking system, embedded into a magnetic field, calorimeters and a muon system, and possible very forward detectors arranged along the beamline. The framework is interfaced to standard file formats (e.g. Les Houches Event File or HepMC) and outputs observables such as isolated leptons, missing transverse energy and collection of jets which can be used for dedicated analyses. The simulation of the detector response takes into account the effect of magnetic field, the granularity of the calorimeters and subdetector resolutions. A simplified preselection can also be applied on processed events for trigger emulation. Detection of very forward scattered particles relies on the transport in beamlines with the HECTOR software. Finally, the FROG 2D/3D event display is used for visualisation of the collision final states.

Useful links, references and talks can be found here.

Community and support

Delphes is now moving to community-based developement. A forum is already open to authenticated members in order to exchange on issues and features requests. The **branches** of the svn repository are open to users who would like to participate. If you are interested, register first and then contact the suport team via cp3-delphes@uclouvain.be to be granted svn access.

Quick start

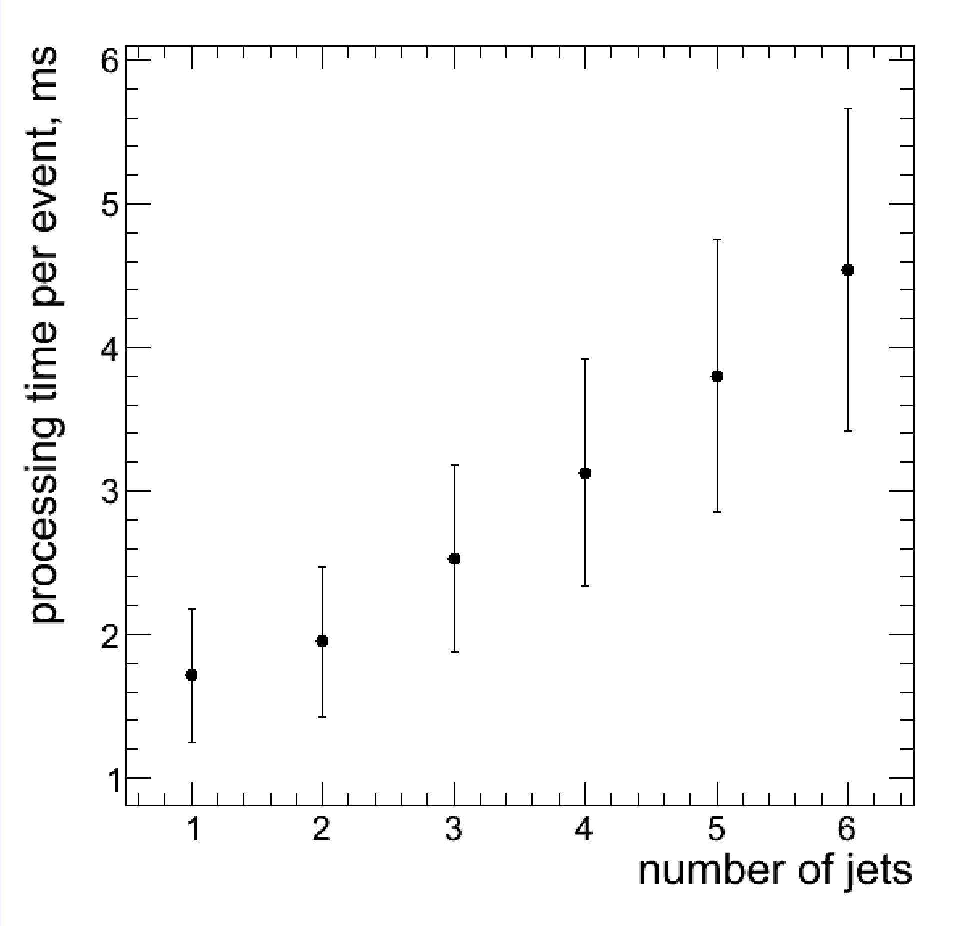
Beginners will find help in the Documentation and the FAQ.

Core technical features

- Code adjustable by users familiar with C++ / ROOT
- Easy interface with existing libraries (file IO, jet finding)
- Interoperability with the ROOT analysis framework

Speed

- few ms / event on a standard laptop
- Ttbar events :



Delphes' foundation

- File IO:
  - HepMC: http://lcgapp.cern.ch/project/simu/HepMC
  - ROOT: http://root.cern.ch
  - ExRootAnalysis: https://server06.fynu.ucl.ac.be/projects/ExRootAnalysis
  - LHEF Reader: http://home.thep.lu.se/~leif/LHEF
  - StdHep: http://cepa.fnal.gov/psm/stdhep
  - MCFIO: http://cepa.fnal.gov/psm/simulation/mcfio

- Jet finding:
  - FastJet: http://fastjet.fr

- Particle transport through beam lines:
  - Hector: http://www.fynu.ucl.ac.be/themes/he/ggamma/hector

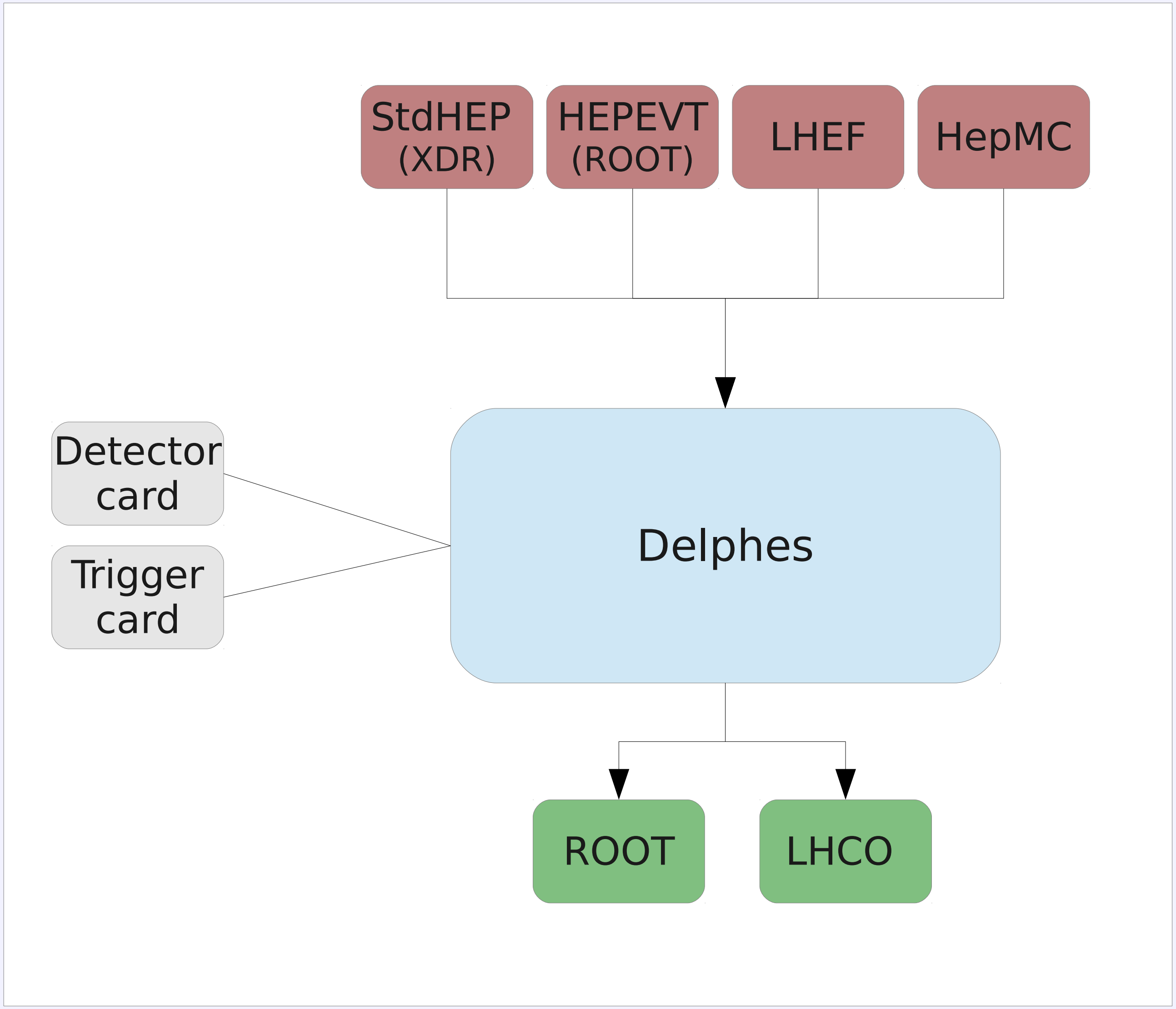
- Event Display:
  - FROG: http://frog.hepforge.org

Features

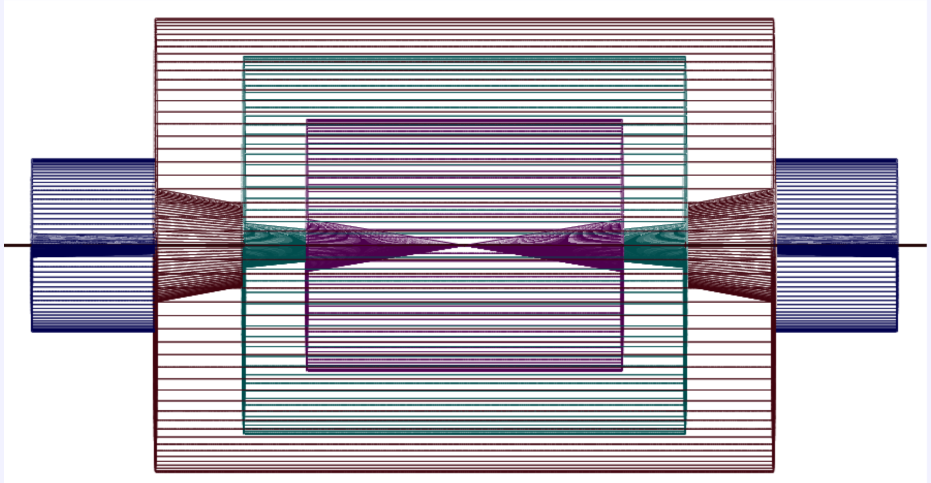
- Fast simulation of the following sub-detectors:
  - Propagation of particles in a magnetic field
  - calorimeters with electromagnetic and hadronic sections
  - muon detectors
  - (very-)forward detectors

- Reconstruction of physics objects:
  - Isolated electrons and muons
  - photons
  - jets
  - b-jets
  - tau-jets
  - missing transverse energy

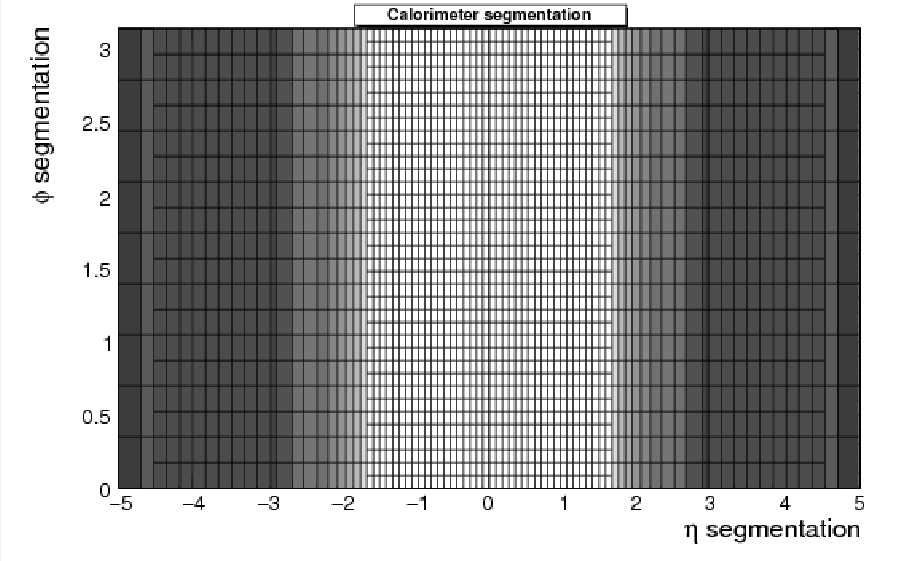
Input and output files



- Detector extension in pseudorapidity:
  - tracker
  - central calorimeter
  - forward calorimeter
  - muon detectors



- Calorimeter segmentation
  - calorimeters are symmetric in η
  - all cells have identical size in ϕ for given η
  - identical segmentation for EM and HAD



Remark:

Unless otherwise stated, all given values can be set in the config cards

Detector simulation (I)

- Particles with P_T > 0.9 GeV/c are propagated within a magnetic field until they reach the calorimeter
- Track reconstruction efficiency is 90% by default
- Particle energies (except muons) are smeared according to the resolution of the calorimeter they reach

$$\sigma^2(\eta) = N^2(\eta) + S^2(\eta)\cdot E + C^2(\eta)\cdot E^2$$

Detector (CMS default)	S	N	C
ECAL	0.05	0.25	0.0055
ECAL endcaps	0.05	0.25	0.0055
FCAL (e-m)	2.08	0	0.1070
HCAL	1.50	0	0.0500
HCAL endcaps	1.50	0	0.0500
FCAL (had)	2.70	0	0.1300

- For muons, transverse momenta are smeared (C=0.01): $\sigma^2 = C^2\cdot p_T^{\ 2}$

- Fraction of energy deposited in EM and HAD calorimeters is taken into account:

$$E_{smeared} = gaus(E \cdot f_{EM}, \sigma_{EM}(\eta)) + gaus(E \cdot f_{HAD}, \sigma_{HAD}(\eta))$$

- All energies deposited in a given η-ϕ cell are summed to form a tower:

$$E_{tower} = gaus\left(\sum (E_i \cdot f_{EM\ i}), \sigma_{EM}(\eta)\right) + gaus\left(\sum (E_i \cdot f_{HAD\ i}), \sigma_{HAD}(\eta)\right)$$

particles	f_{EM}	f_{HAD}
e γ π^0	1	0
Long-lived neutral hadrons (K^0_S , Λ^0)	0.3	0.7
ν μ	0	0
others	0	1

Reconstruction

- Photons / Electrons / Muons:
  - Identification: MC particle PID
  - 10 GeV P_T cut
  - Photon position from calorimeter cell
  - Muons do not leave energy in calorimeters
- Not simulated: fakes, punch-through, Bremsstrahlung, conversions
- Electrons and muons isolation:
  - No Tracks with P_T > 2 Gev/c in a 0.5 Cone

Reconstruction: Jets

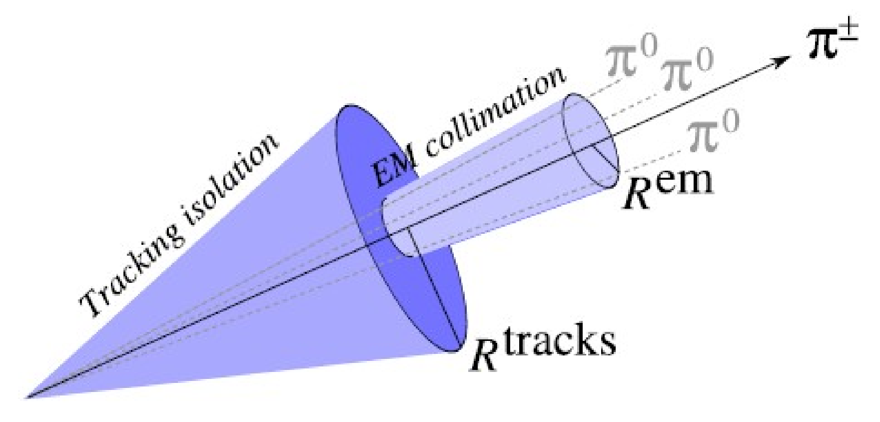
- Based on calorimeter towers
- Using FastJet Library with:
  - CDF Jets, CDF Midpoint
  - SISCone
  - Kt, anti-Kt
  - Etc...
- Energy flow (optional): jets are reconstructed using:
  - Charged particles momenta without smearing (in tracker)
  - Smeared tower energy for neutrals (in tracker)
  - Smeared tower energy for all (outside tracker)
- Electromagnetic and hadronic fractions, as well as number of tracks, are stored in the jet collection

B-tagging:

- Based on most energetic parton in jet:
  - Functions of η & P_T can be defined in config file
  - Flat default: b quark → 40 % tag, c quark → 10% mistag, light → 1% mistag

Tau-jets:

- Only one track with P_T > 2 GeV /c in 0.4 cone (rejection of "3-prong")
- 95% of the energy in a 0.15 cone
- Jet P_T > 10 GeV



Ideal Missing ET reconstruction based on calo towers and muons:

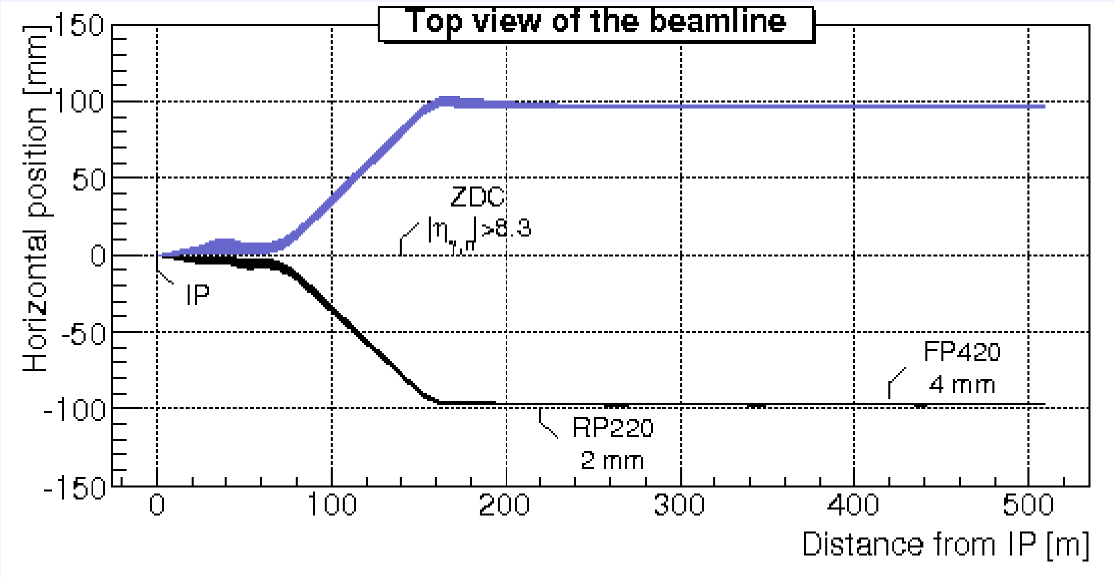
$$\vec{E}_T^{miss} = - \sum_{i}^{towers} E_T(i) - \sum_{j}^{muons} E_T(j)$$

Effects not simulated:

- Dead channels
- Noisy towers
- Cracks
- Etc...

Very forward detectors

- CMS: Castor, ZDC, TOTEM detectors, HPS
- Atlas: Alfa, Lucid, ZDC, AFP
- In Delphes: only ZDC + 2 sets of near-beam detectors
- Beamline propagation using Hector
- Acceptances in config cards.



Incoming beam

outgoing beam

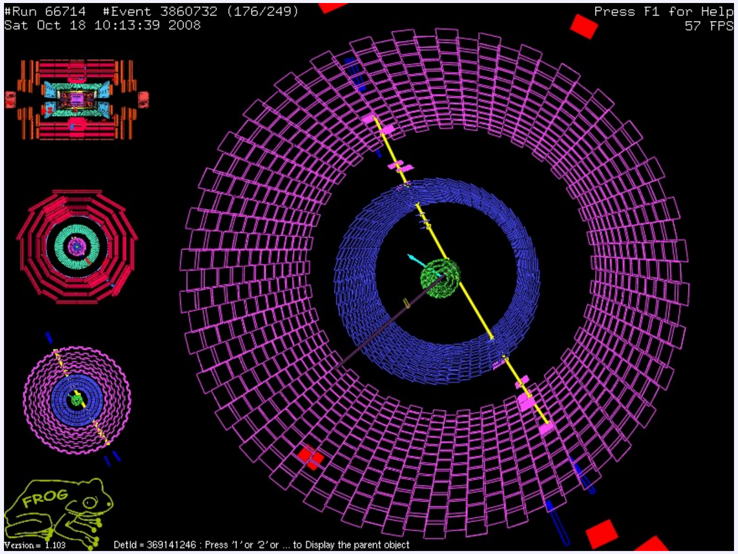
Trigger emulation & event visualisation

Trigger:

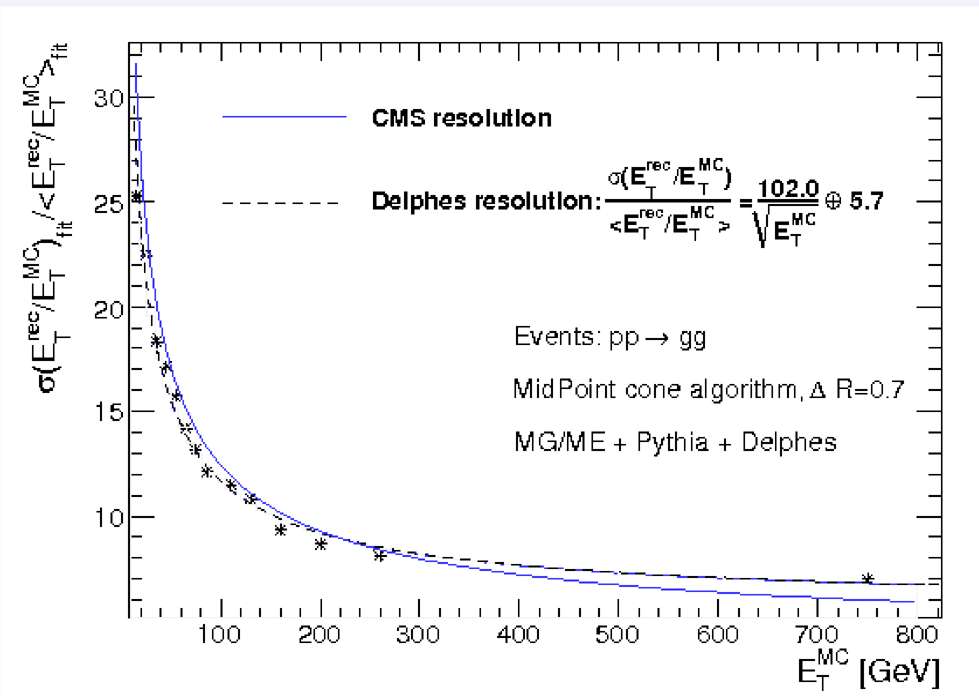
- Cut-based preselection on off-line objects
- Full trigger table through config card
- Logical combinations (AND)

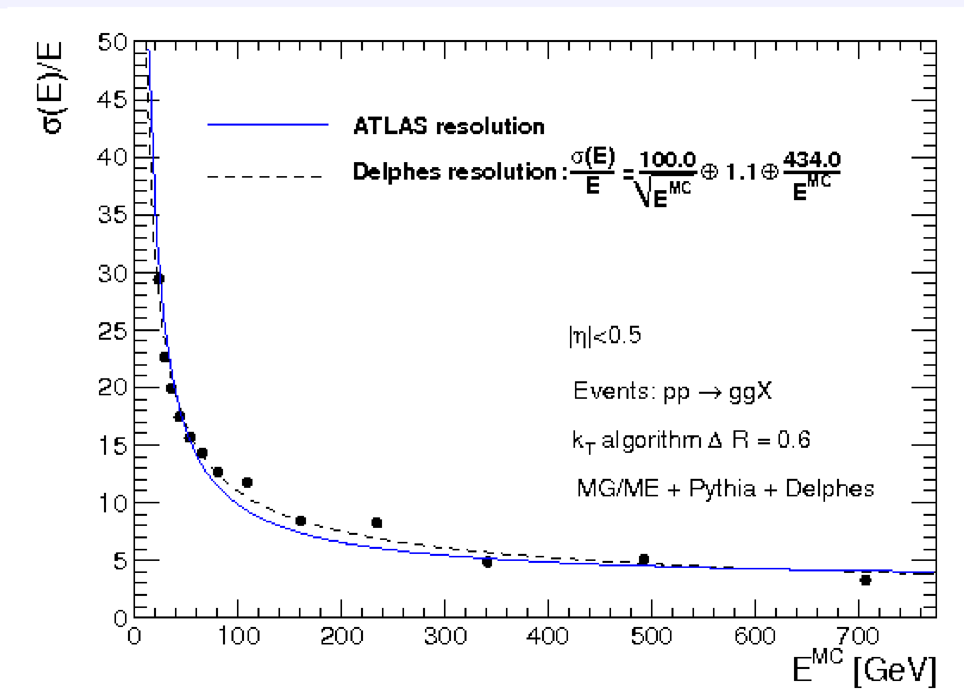
Event display:

- FROG interface for 2D & 3D visualisation
- Geometry defined in config card



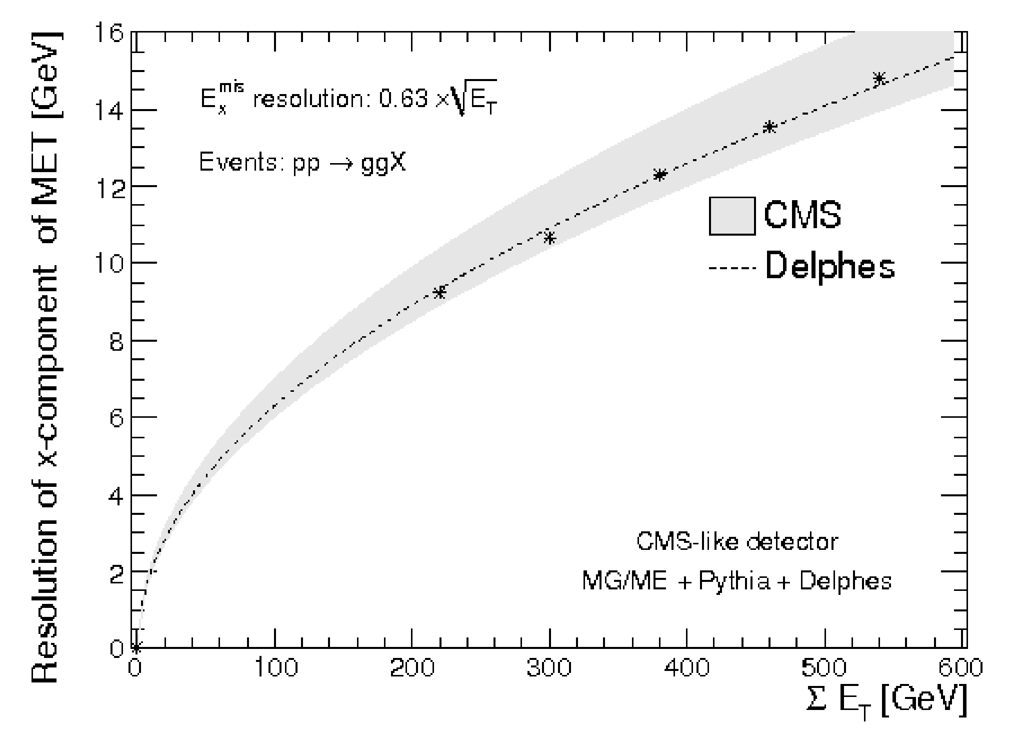
CMS resolution from: The CMS Collaboration, CERN/LHCC 2006-001.
ATLAS resolution from: The ATLAS Collaboration, CERN-OPEN 2008-020.

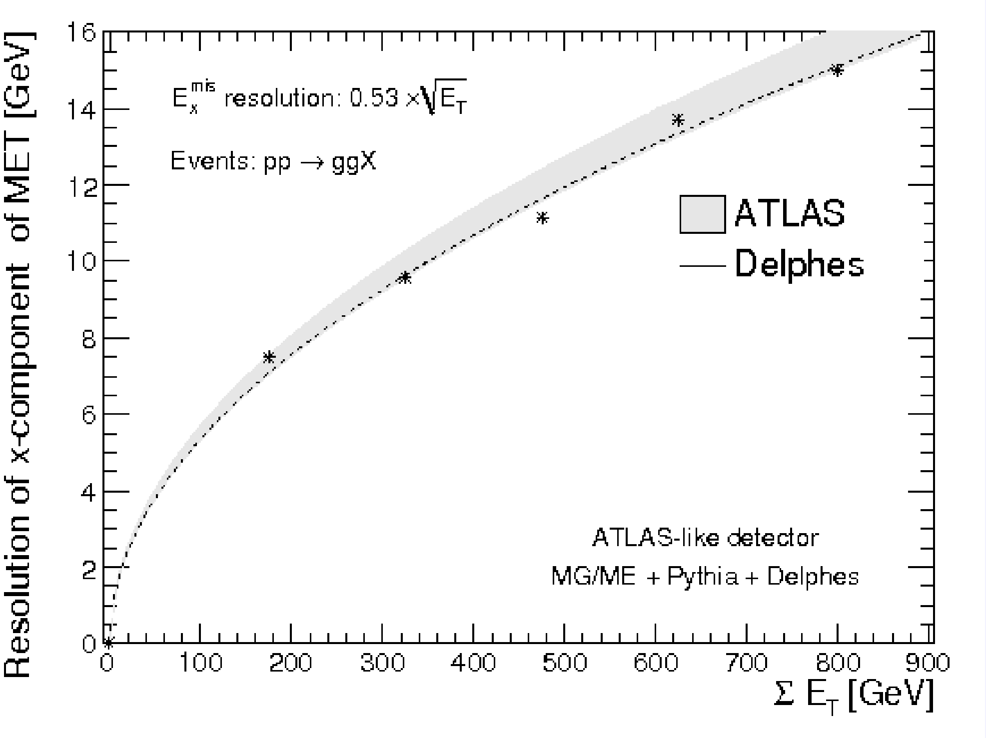




→ Reasonable agreement

CMS resolution from: The CMS Collaboration, CERN/LHCC 2006-001.
ATLAS resolution from: The ATLAS Collaboration, CERN-OPEN 2008-020.





→ Reasonable agreement

Development

- Community-based development
  - Improvements and bugfixes requested through the ticketing system
  - Users (you) propose patches
  - CP3/UCL (us) test and commit, release new version often
  - Users get a mention on the frontpage
- Status
  - > 10 patches in the last 5 months, most by external users
  - Current developments in b-tagging ongoing @ CP3 (A.Mertens)
  - Code optimisation ongoing @ CP3 (P.Demin)
- Conclusions
  - This model works well
  - Delphes development is alive thanks to you

Backup slides

Possible improvements

- Pile-up effects
- Calorimeter noise
- Multiple scattering → smearing of tracks
- Improved lepton isolation criteria
- B-tagging improvements: do not use most energetic parton
- Allow for JES tuning
- Use non-gaussian HCAL smearing
- Implementation of efficiency functions
- Vertex position smearing
- Generic objects improvements
- Generic detector parts improvements

Input files

- Delphes reads the following file formats
  - StdHEP (XDR)
  - ROOT files obtained with h2root (hbook)
  - Les Houches Event Format
  - HepMC

- Delphes is driven by two input cards defining
  - detector card
  - trigger card

- Default detector cards and trigger tables for ATLAS & CMS based on published material

Output files

- Delphes outputs results in two file formats:

  - ROOT file containing three trees
    - GEN tree (generated particles)
    - Analysis tree (reconstructed objects)
    - Trigger tree (trigger acceptance)

  - LHCO file containing information about reconstructed objects