Delphes fast simulation

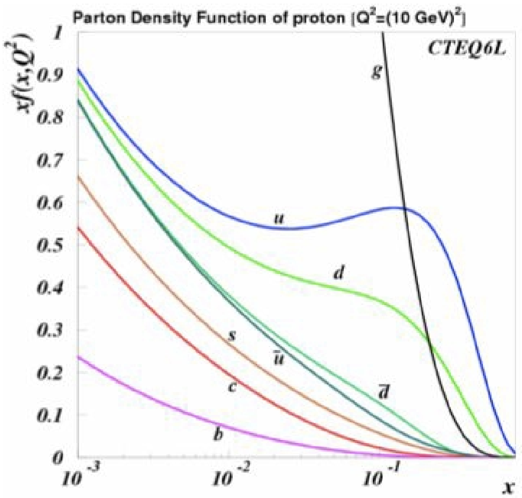
Michele Selvaggi, for the Delphes Team

Université Catholique de Louvain (UCL)
Center for Particle Physics and Phenomenology (CP3)

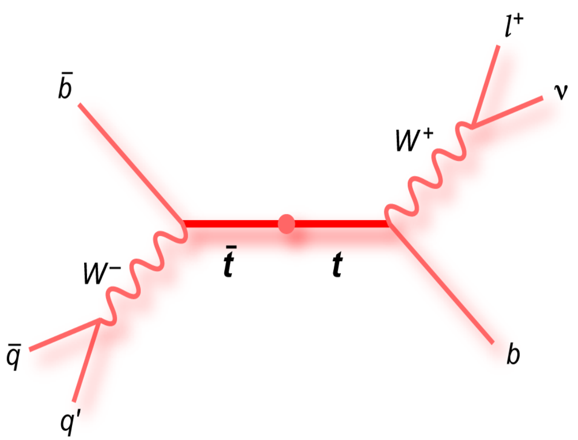
Zurich Phenomenology Workshop 2014
10 January 2014

Motivation

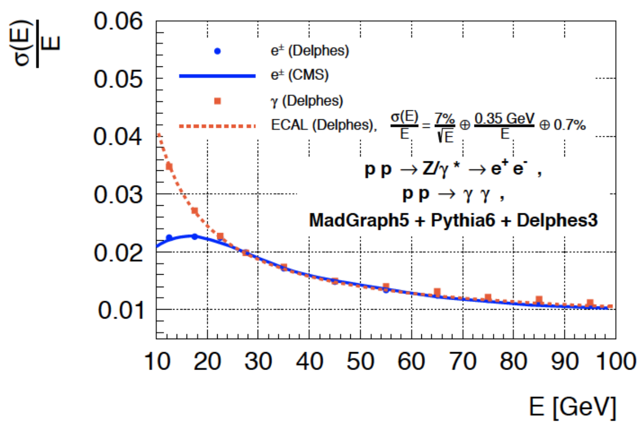
$$P(\mathbf{x}|\alpha) = \frac{1}{\sigma_\alpha} \int dq_1 dq_2 f(q_1) f(q_2) \int d\phi(y) |M_\alpha(q_1, q_2, y)|^2 W(\mathbf{x}|y)$$



PDFs



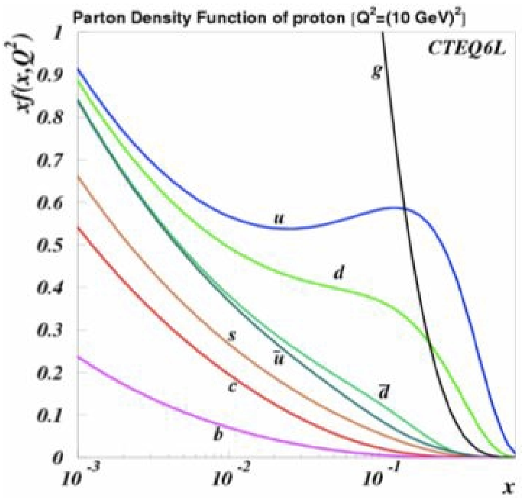
Matrix Element



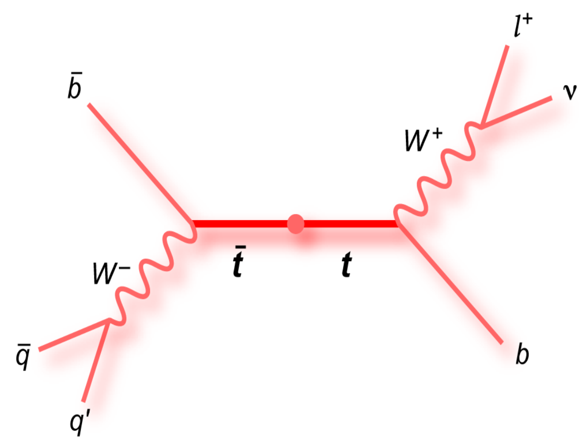
Resolution function

Motivation

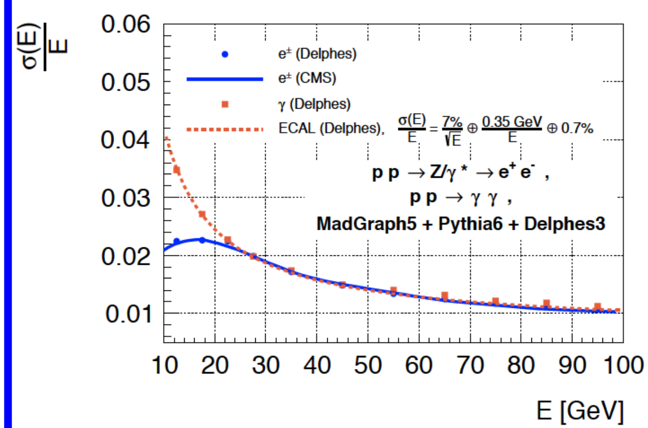
$$P(\mathbf{x}|\alpha) = \frac{1}{\sigma_\alpha} \int dq_1 dq_2 f(q_1) f(q_2) \int d\phi(y) |M_\alpha(q_1, q_2, y)|^2 W(\mathbf{x}|y)$$



PDFs



Matrix Element



Resolution function

→ simulation plays key role for asserting resolution functions

Detector simulation

- Full simulation (GEANT):

  - **simulates** particle-matter interaction (including e.m. showering, nuclear int., brehmstrahlung, photon conversions, etc ...) $\rightarrow$ 10-100 s /ev

- Experiment Fast simulation (ATLAS, CMS ...):

  - **simplifies** and makes faster simulation and reconstruction $\rightarrow$ 1 s /ev

- Parametric simulation:

  Delphes, PGS**:**

  - **parameterize** detector response, reconstruct complex objects $\rightarrow$ 10 ms /ev

  TurboSim

  - **no detector**,parameterize object response, parton $\leftrightarrow$ reco

Detector simulation

- Full simulation (GEANT):

  - **simulates** particle-matter interaction (including e.m. showering, nuclear int., brehmstrahlung, photon conversions, etc ...) $\rightarrow$ 10-100 s /ev

- Experiment Fast simulation (ATLAS, CMS ...):

  - **simplifies** and makes faster simulation and reconstruction $\rightarrow$ 1 s /ev

- Parametric simulation:

  Delphes:

  - **parameterize** detector response, reconstruct complex objects $\rightarrow$ 10 ms /ev

  TurboSim

  - **no detector**,parameterize object response, parton $\leftrightarrow$ reco

DELPHES

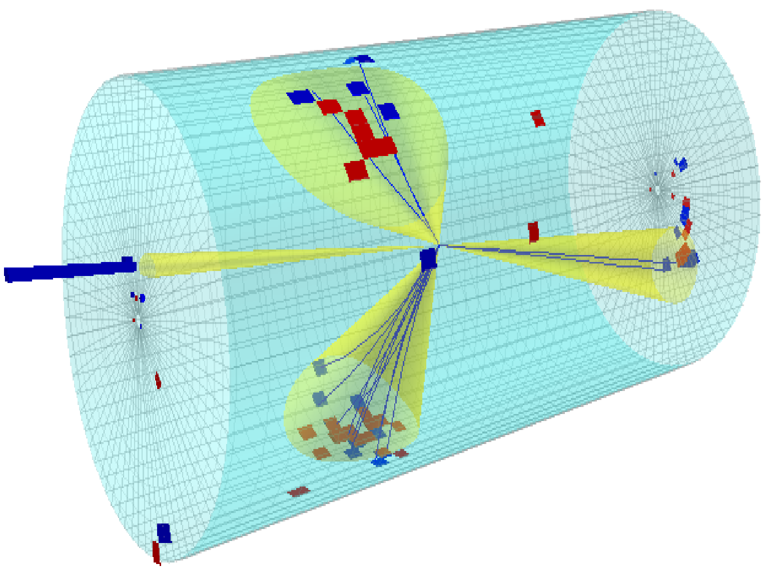
Development

- Delphes project started back in 2007 at UCL
- Since 2009, its development is **community-based**
  - **ticketing system** for improvement and bug-fixes
    → user proposed patches
  - **Quality control** and **core development** is done at the UCL
- In 2013, **DELPHES 3** was released:
  - modular software
  - new features
  - included in MG/ME suite
- **Widely** tested and used by the community (mainly pheno)
- Website and manual: https://cp3.irmp.ucl.ac.be/projects/delphes
- Paper: `arXiv:1307.6346`

DELPHES in a nutshell

- **Delphes** is a **modular framework** that simulates of the response of a multipurpose detector

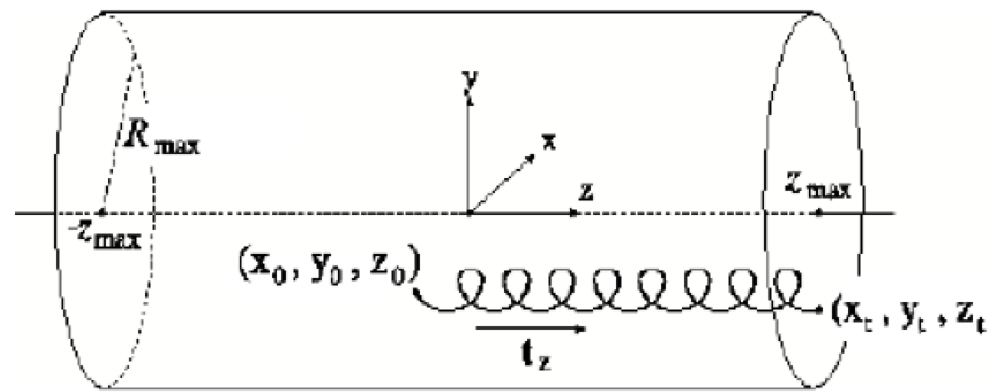
- **Simulates**:
  - pile-up
  - charged particle propagation in magnetic field: **tracking**
  - electromagnetic and hadronic **calorimeters**
  - **muon** system



- **reconstructs**:
  - leptons (electrons and muons)
  - photons
  - jets and missing transverse energy (particle-flow)
  - taus and b's

Charged Particle Propagation

- Charged particles are propagated in the magnetic field until they reach the calorimeters



- Propagation parameters:
  - magnetic field B
  - radius and half-length (R_{max}, z_{max})

- Efficiency/resolution depends on:
  - particle ID
  - transverse momentum
  - pseudorapidity

```
# efficiency formula for muons
add EfficiencyFormula {13} {                                         (pt <= 0.1)   * (0.000) + \
                                  (abs(eta) <= 1.5) * (pt > 0.1   && pt <= 1.0)   * (0.750) + \
                                  (abs(eta) <= 1.5) * (pt > 1.0)                  * (1.000) + \
                  (abs(eta) > 1.5 && abs(eta) <= 2.5) * (pt > 0.1   && pt <= 1.0)   * (0.700) + \
                  (abs(eta) > 1.5 && abs(eta) <= 2.5) * (pt > 1.0)                  * (0.975) + \
                  (abs(eta) > 2.5)                                                  * (0.000)}
```

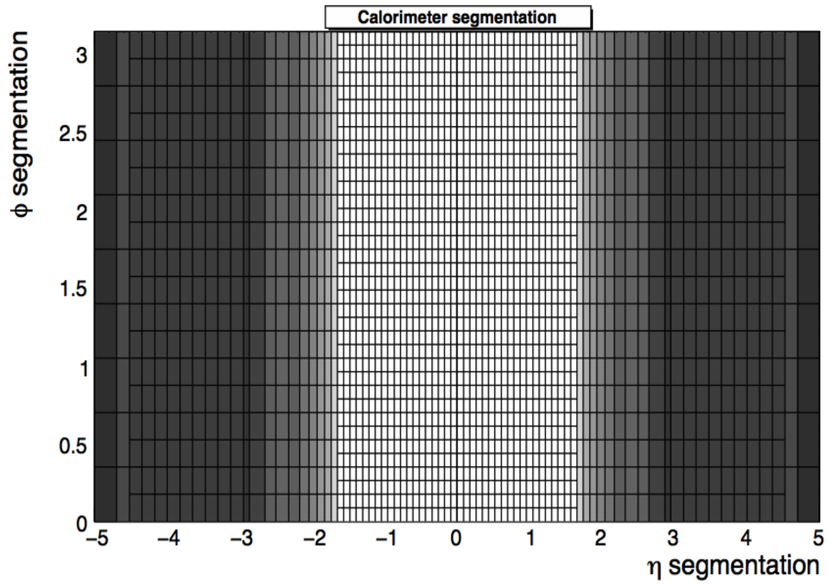
Not real tracking/vertexing !!
→ no fake tracks/ conversions (but can be easily implemented)
→ no dE/dx measurements

Calorimetry

- em/had calorimeters have same **segmentation** in eta/phi

- Each particle that reaches the calorimeters **deposits a fraction of its energy** in one ECAL cell (f_{EM}) and HCAL cell (f_{HAD}), depending on its type:

particles	f_{EM}	f_{HAD}
$e\ \gamma\ \pi^0$	1	0
Long-lived neutral hadrons (K^0_S, Λ^0)	0.3	0.7
$\nu\ \mu$	0	0
others	0	1



- Particle energy is **smeared** according to the calorimeter cell it reaches

$$E_{Tower} = \sum_{particles} \ln\mathcal{N}\left(f_{ECAL}\cdot E, \sigma_{ECAL}(E,\eta)\right) + \ln\mathcal{N}\left(f_{HCAL}\cdot E, \sigma_{HCAL}(E,\eta)\right)$$

$$\left(\frac{\sigma}{E}\right)^2 = \left(\frac{S(\eta)}{\sqrt{E}}\right)^2 + \left(\frac{N(\eta)}{E}\right)^2 + C(\eta)^2$$

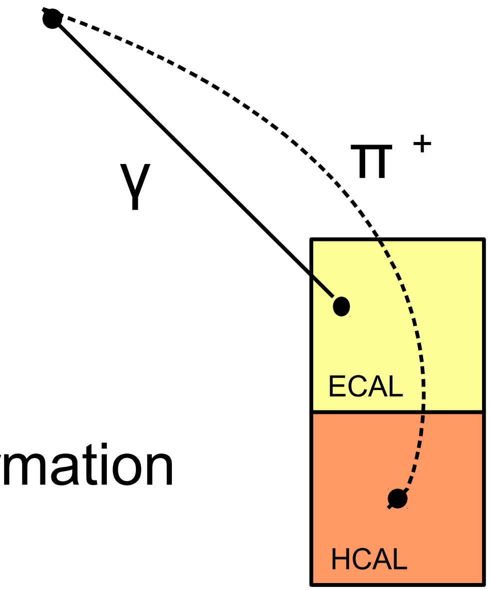
Leptons, photons

- Muons/photons/electrons
  - **identified** via their PDG id
  - inside the **tracker coverage** for electrons and muons
  - muons do not deposit energy in calo (independent smearing parameterized in p_T and η)
  - electrons and photons smeared according to electromagnetic calorimeter resolution
- Isolation: $$\text{rel.Iso} = \frac{\sum_{\Delta R<0.5} p_T^{track}}{p_T}$$ → modular structure allows to easily define different isolation

If rel.Iso ~ 0, the lepton is isolated

- Not taken into account:
  - fakes, punch-through, brehmstrahlung, conversions

Jets / E_T^{miss} / H_T



- Inputs can be formed from:

  - **calorimeter towers**

  - **"particle-flow"** candidates (tracks and towers):

    → optimally combine calorimeter and tracking information
    → compare track momentum with tower energy:

    - if $E_{tower} \leq E_{track}$, PF track with energy E_{track} is created
    - if $E_{tower} > E_{track}$, PF tower with energy $E_{tower} - E_{track}$ and PF track with energy E_{track} are created

- Jets, E_T^{miss} and H_T quantities can be computed from both calorimeter towers and particle-flow candidates.

b and tau jets

- b-jets

  - if **b** parton is found in a cone ΔR w.r.t jet direction
    → apply **efficiency**
  - if **c** parton is found in a cone ΔR w.r.t jet direction
    → apply **c-mistag rate**
  - if **u,d,s,g** parton is found in a cone ΔR w.r.t jet direction
    → apply **light-mistag rate**

b-tag **flag** is then stored in the jet collection

- tau-jets

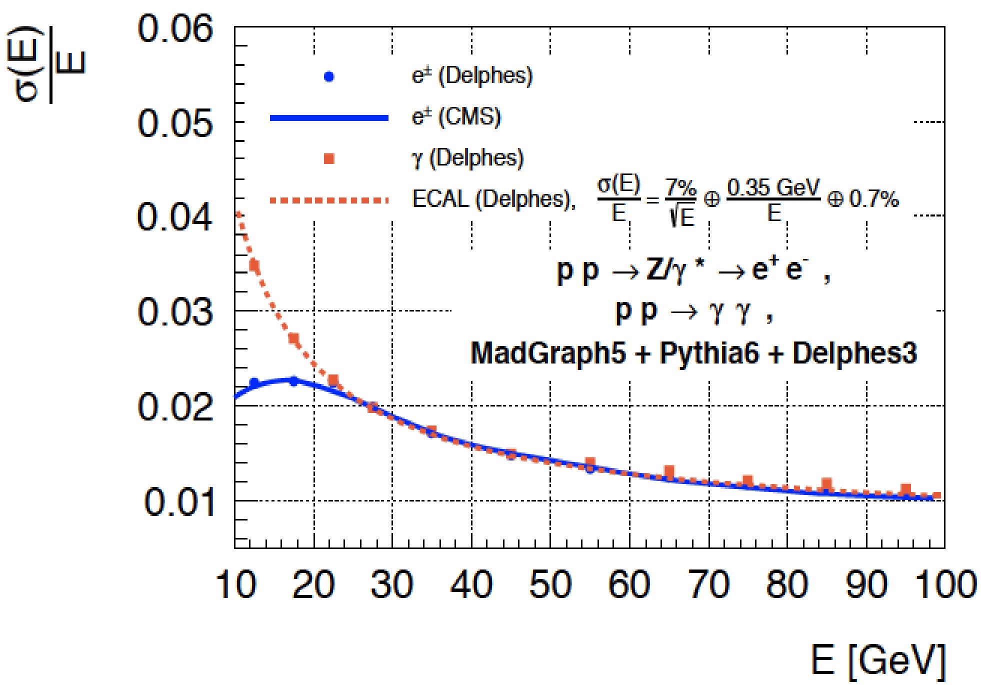
  - if tau lepton is found in a cone ΔR w.r.t jet direction
    → apply **efficiency**
  - else
    → apply **tau-mistag rate**

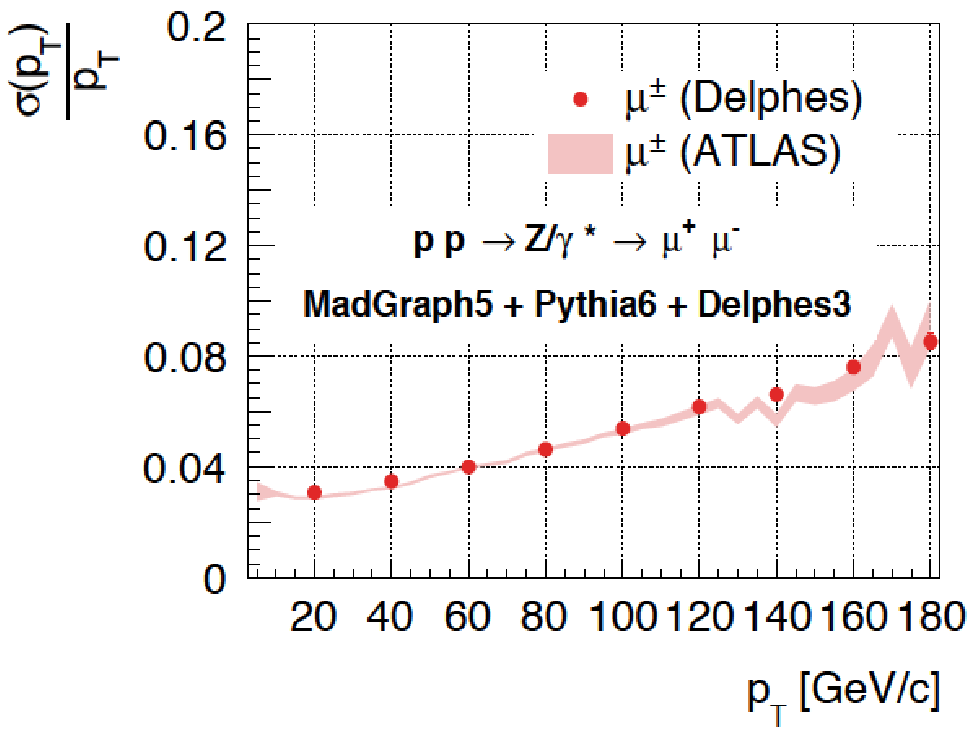
tau jets have their own collection (no leptonic tau decays)

can define **p_T and η dependent** efficiency and mistag rate

Validation

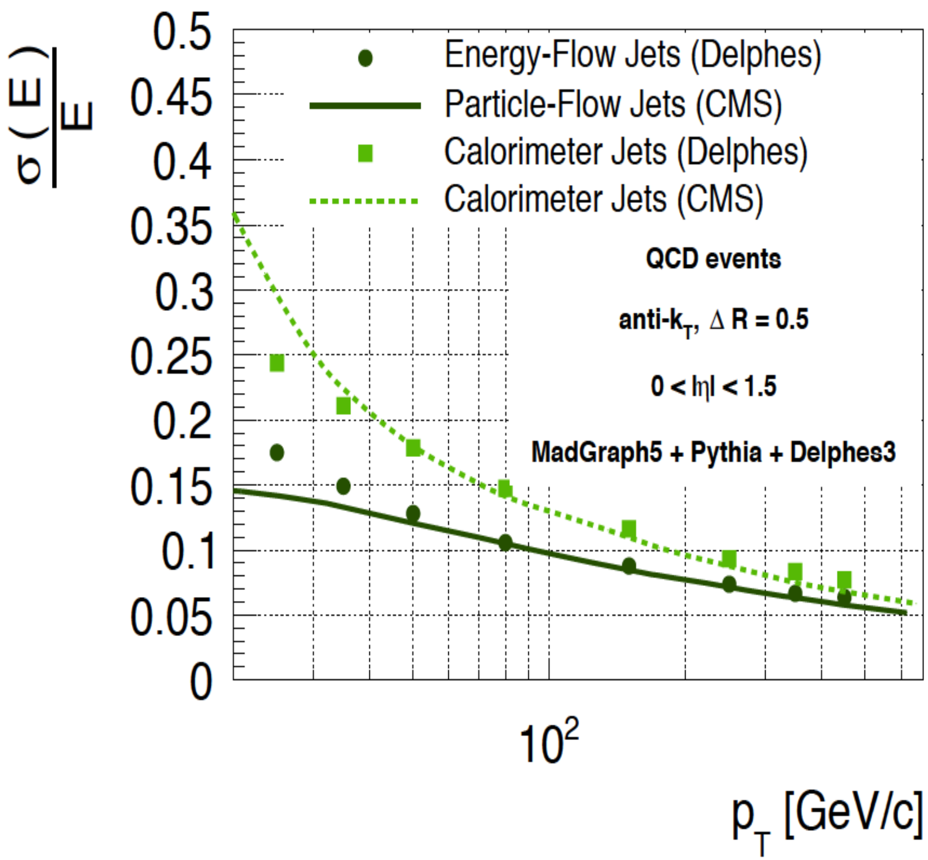
Validation: electrons and muons

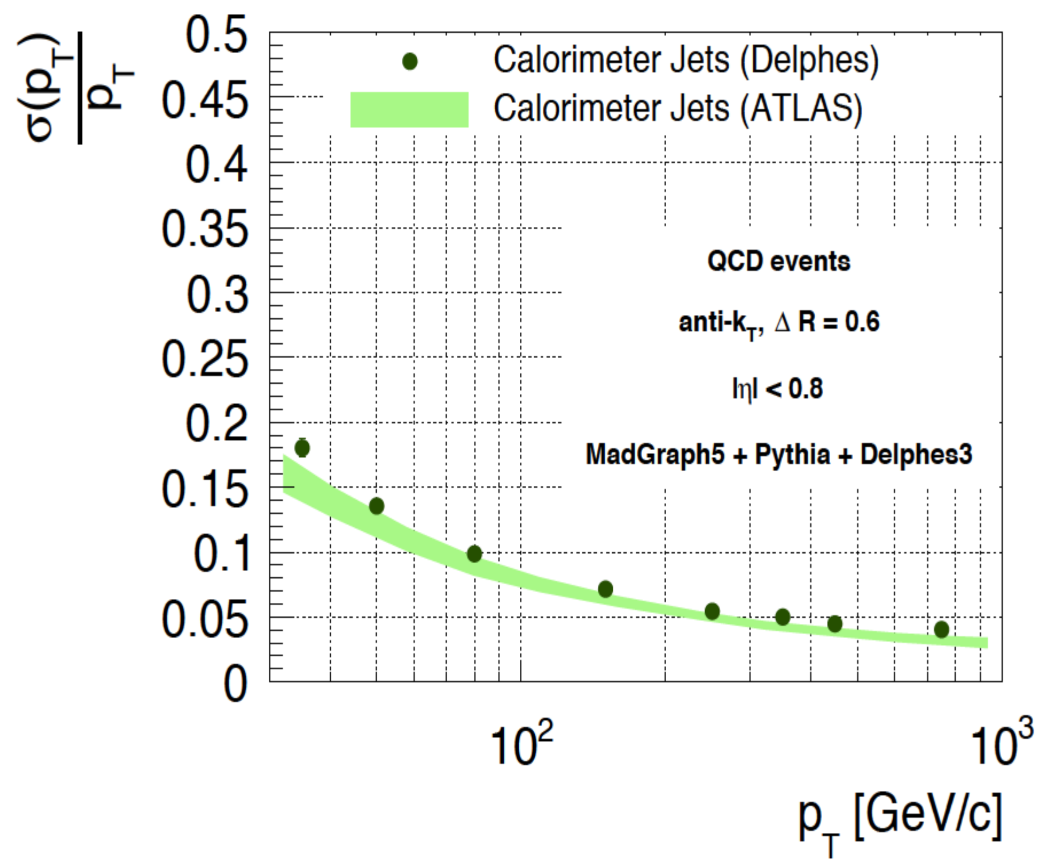




→ **excellent agreement**

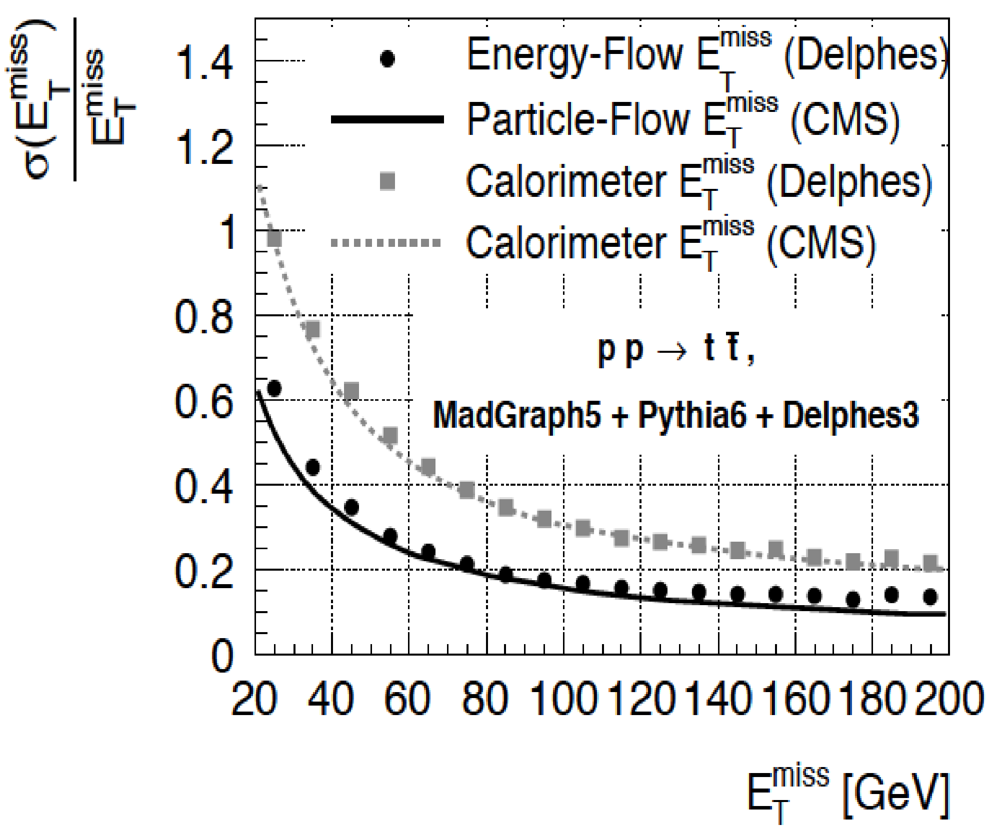
Validation: jets

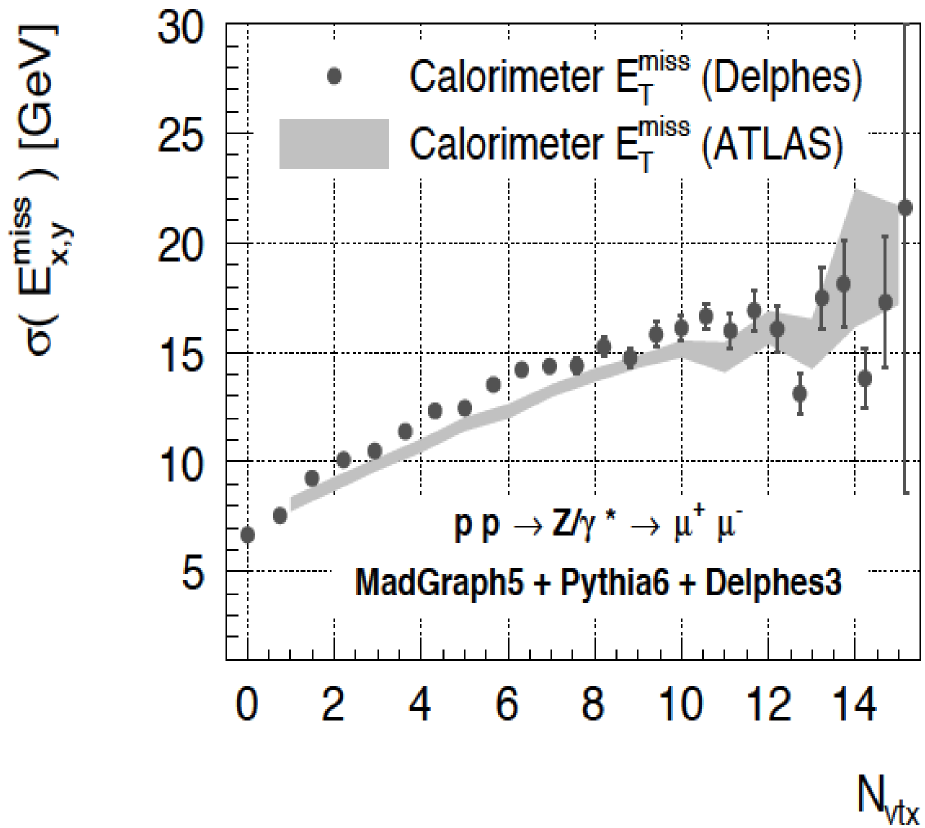




→ **good agreement**

Validation: E_T^{miss}



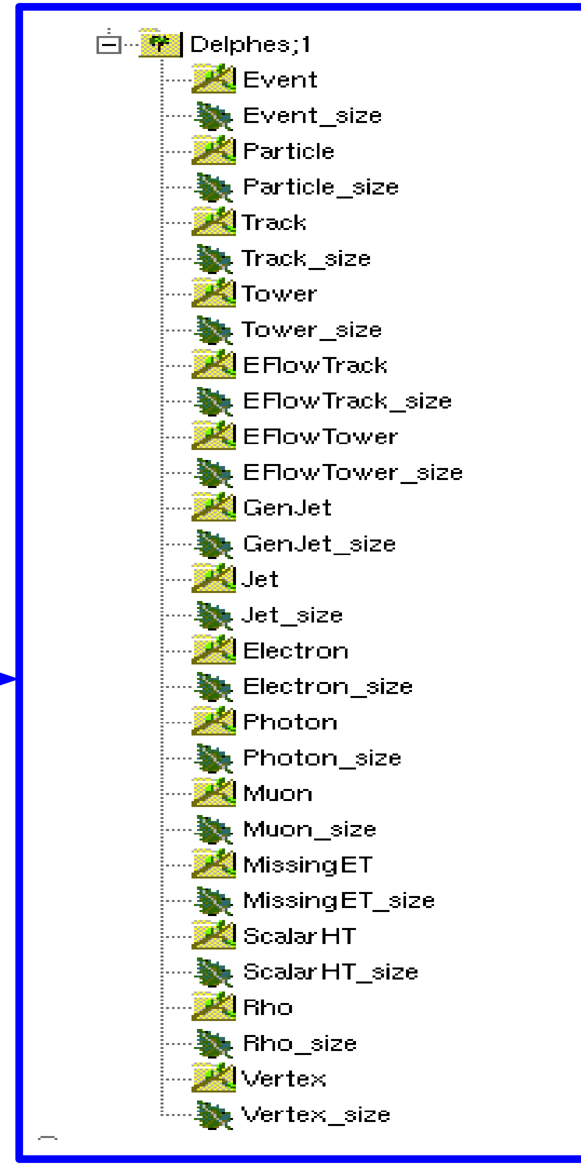


→ **excellent agreement**

Technical features

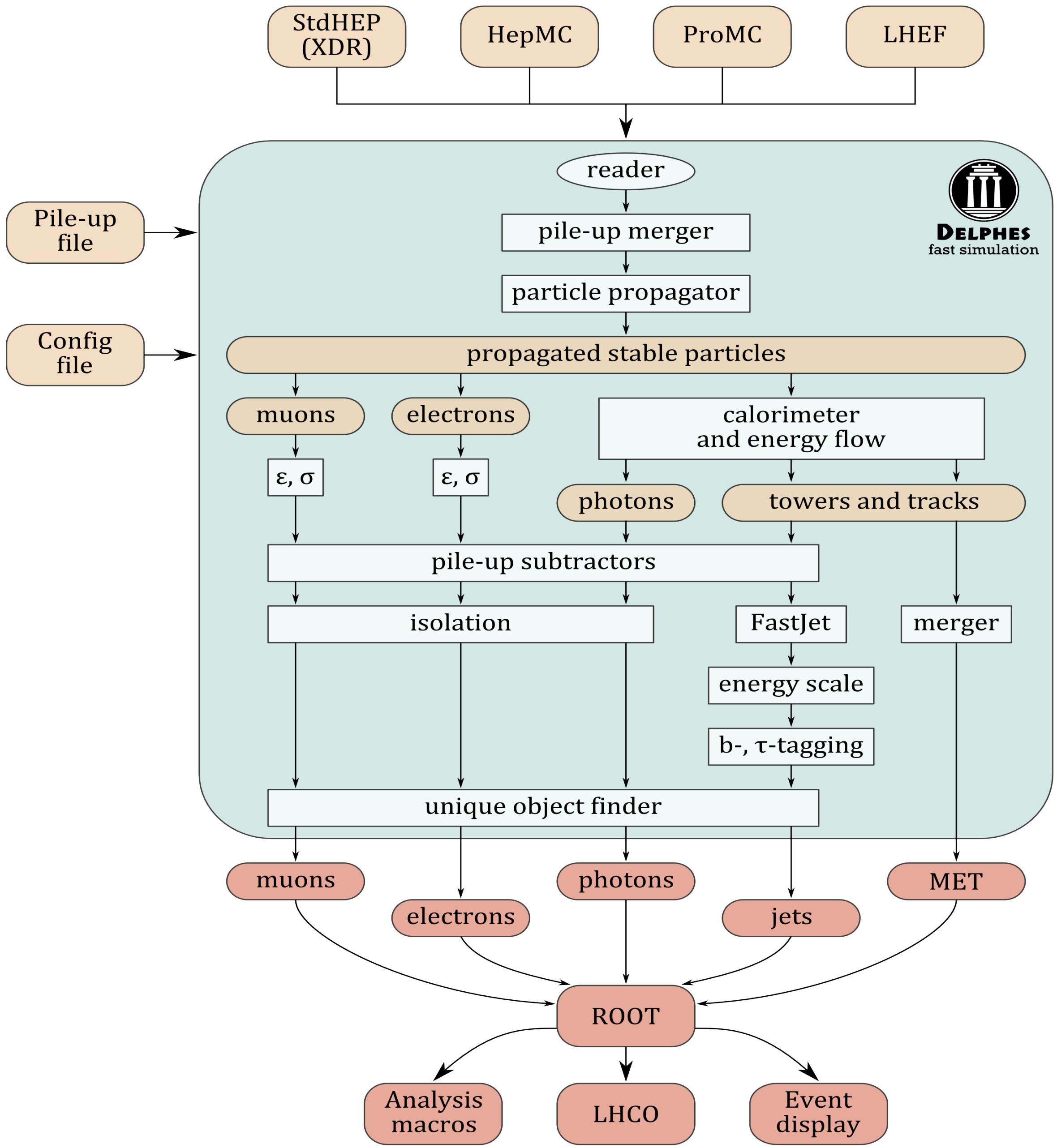
Technical features

- **modular** C++ code, uses ROOT classes
- Input
  - Pythia/Herwig output (HepMC,STDHEP)
  - LHE (MadGraph/MadEvent)
  - ProMC
- Output
  - ROOT trees
- Configuration file
  - define geometry
  - resolution/reconstruction/selection criteria
  - output object collections



default **CMS** and **ATLAS** configurations are included in any Delphes release

Modularity in action



When and when not DELPHES?

- When do you need Delphes?

  → more advanced than parton-level studies
  → **testing analysis methods (multivariate/Matrix Element)**
  → test your model (CheckMATE)
  → scan big parameter space (SUSY-like)
  → preliminary tests of new geometries/resolutions (upgrades, Snowmass)
  → educational purpose (bachelor/master thesis)

- When not to use Delphes?

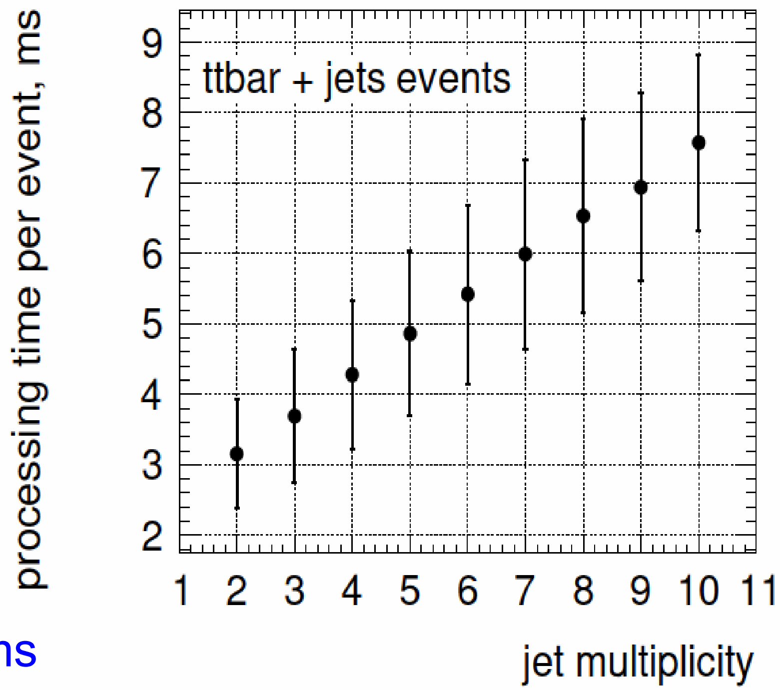
  → high precision studies
  → very exotic topologies (heavy stable charged particles)
  → study is sensitive to tails

Why use DELPHES for ME studies?

- **speed**

  - event generation 1ms – 10s
  - reconstruction 1ms (0 PU) – 1s (150 PU)
  - ME calculation 1s – 100s

  → bottleneck is ME



- **simple/flexible**

  - reconstructed objects contain reference to their parton-level counterpart
    → very easy to build transfer functions
  - modular structure easily allows to alter
    → work-flow, output tree, reconstruction algorithms

- **reliable**

  - well validated and tested software

Delphes and MEM contributions

- *Top-squark search with MEM,* **ZPW 2014 Monte Carlo Simulation**, (Artoisenet)
- *Unravelling ttH via the Matrix Element Method*, **Phys.Rev.Lett. 111 (2013) 091802** (Artoisenet, de Aquino, Maltoni, Mattelaer)
- *Determination of differential cross sections from t anti-t fully leptonic, using the matrix element method*, **Nuovo Cim. C035N3 (2012) 229-232** (Pin, Mattelaer)
- *Top B Physics at the LHC*, **Phys.Rev.Lett. 110 (2013) 232002** (Gedalia, Isidori, Maltoni, Perez, M.S., Soreq)
- *The automated matrix element methods and its applications at LHC*, **ACAT2013 Workshop** (Mertens)

Conclusions

- **Delphes 3** has been out for one year now, with **major improvements**:

  - modularity
  - pile-up implementation
  - revamped particle flow algorithm
  - new visualization tool based on ROOT EVE
  - default cards giving results on par with published performance from LHC experiments
  - now fully integrated within MadGraph5

- Delphes is a great tool for preliminary MEM studies

- Delphes 2 is no longer supported!!

- Test it, and give us feedback!

People

Severine Ovyn
Xavier Rouby
Jerome de Favereau
Christophe Delaere
Pavel Demin
Andrea Giammanco
Vincent Lemaitre
Alexandre Mertens
M.S.

the community ...