S. Ovyn

Delphes

A framework for fast simulation of a generic collider experiment

Xavier Rouby(a), Séverine Ovyn

Université catholique de Louvain, Belgium

Center for Particle Physics and Phenomenology (CP3)

(a) now in Physikalisches Institut Albert-Ludwigs-Universität Freiburg

References

Website :

http://www.fynu.ucl.ac.be/delphes.html

News / Download / User manual / FAQ

Paper + User manual :

arXiv:0903.2225[hep-ph]

DELPHES, a framework for fast simulation of a generic collider experiment

S. Ovyn, X. Rouby[a] and V. Lemaître
Center for Particle Physics and Phenomenology (CP3)
Université catholique de Louvain
B-1348 Louvain-la-Neuve, Belgium

Contacts :

severine.ovyn@uclouvain.be

xavier.rouby@cern.ch

UCL
Université catholique de Louvain

Delphes

S. Ovyn

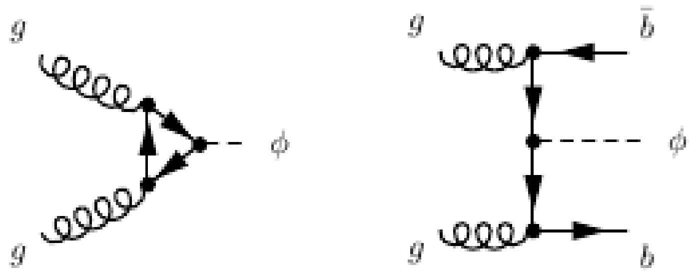
Motivation: from theory to detectors...

From theory to detectors...

Knowing if theoretical predictions will be visible and measurable in a high energy experiment is complex and requires several steps:

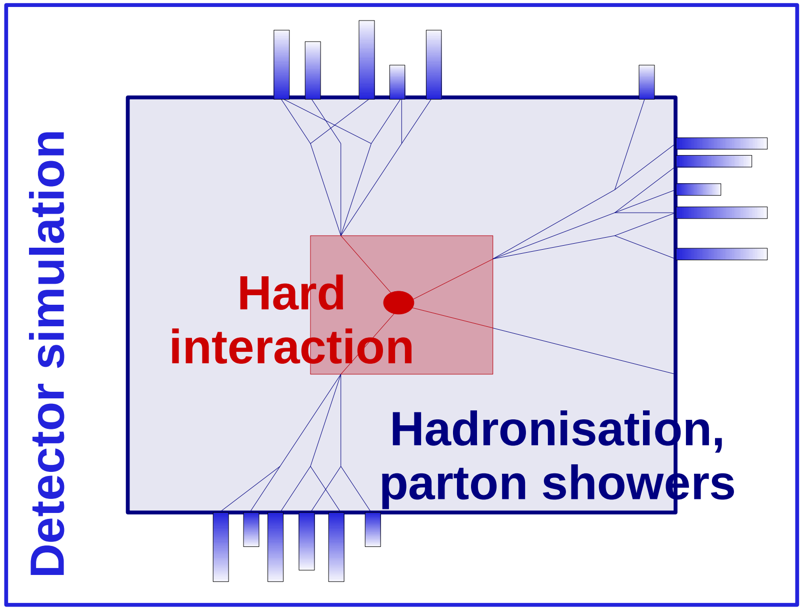
S. Ovyn

1° Development of a new model



2° Implementation and generation of hard interaction

- MadGraph/MadEvent (MG/ME)
- CalcHep



3° Simulation of hadronisation and parton showers

- Pythia
- Herwig

Complexity of HE detectors...

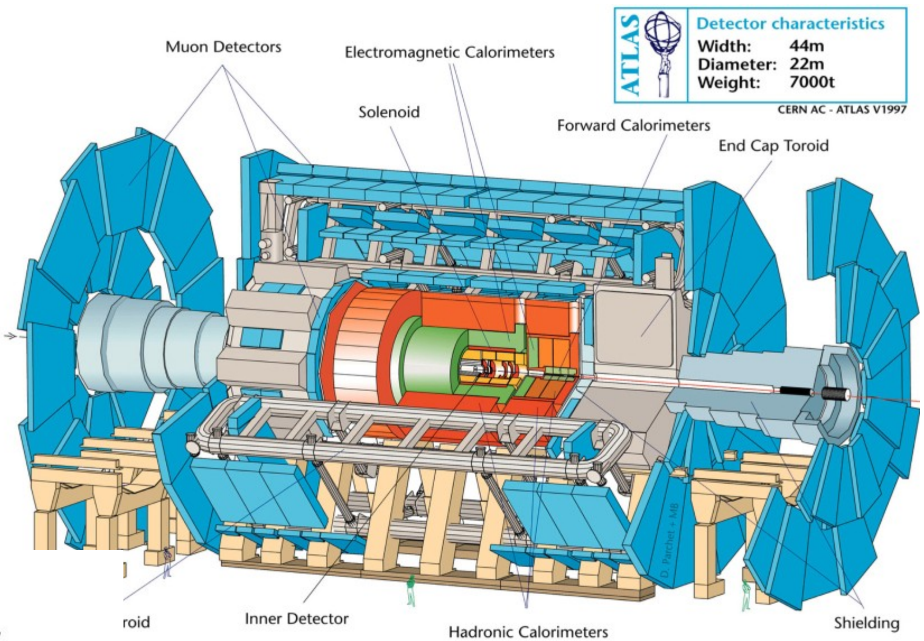
Delphes

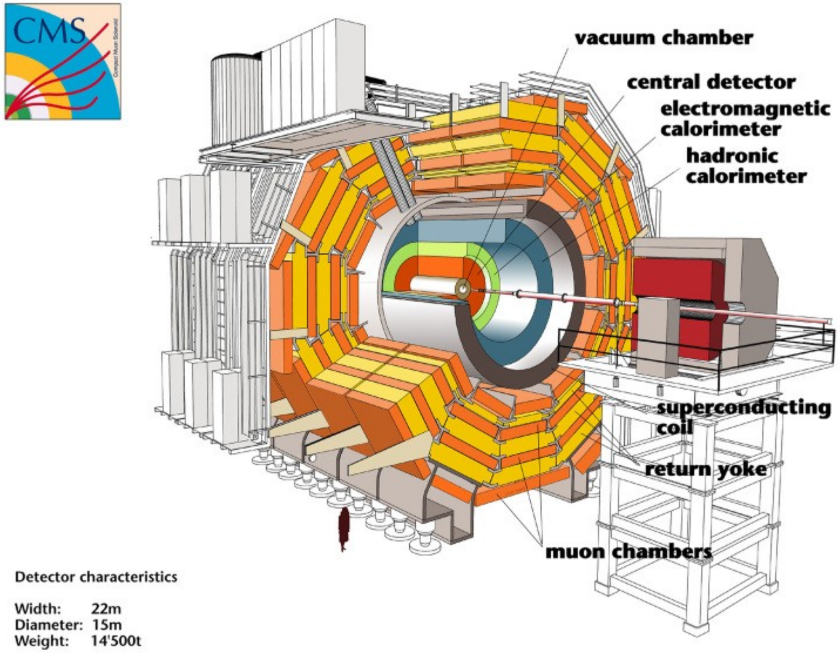
S. Ovyn

4° Simulation of the response of a high energy experiment

- ATLAS
- CMS

ATLAS ➡





⬅ **CMS**

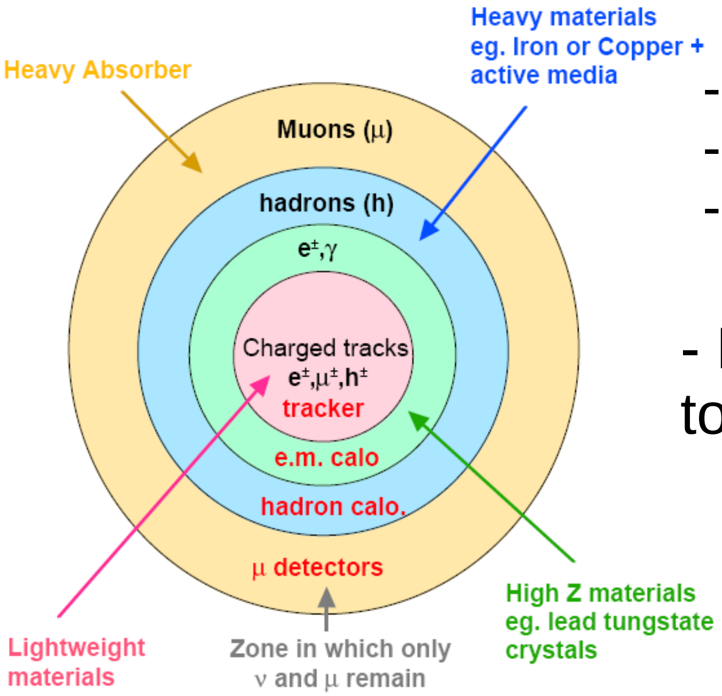
UCL
Université catholique de Louvain

Complexity of HE detectors...

Delphes

S. Ovyn

General structure



- Complexity of the related subdetectors
  - tracker
  - electromagnetic and hadronic calorimeters
  - muon chambers

- Requires the use of complex softwares to simulate
  - detailed energy deposition from ionization, showering
  - secondary interactions
  - detector inefficiencies
  - multiple scattering
  - ...

Such a simulation is very complex and a large CPU per event

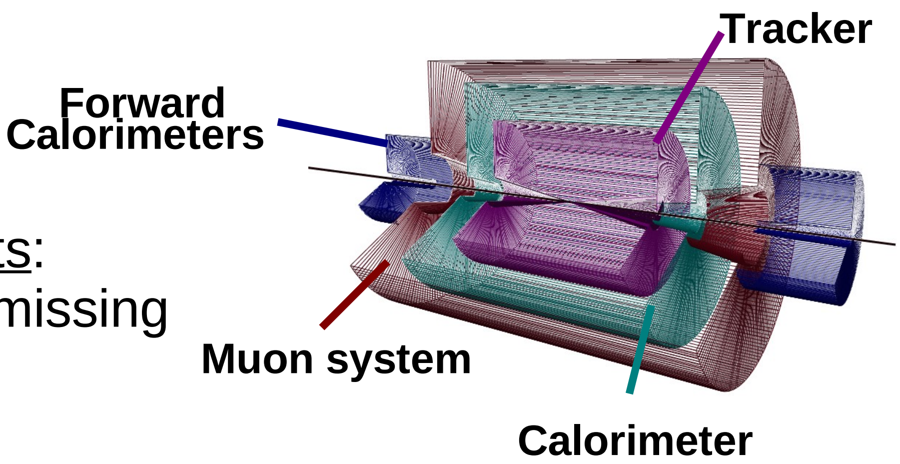
Phenomenological studies may require only <u>fast</u> but <u>realistic</u> estimates of detector response → Delphes

Fast simulation utilitiy

Delphes provides:

- Realistic simulation taking into account subdetector extensions, types, segmentations and resolutions

    - A tracker in a solenoidal magnetic field
    - Calorimeters with electromagnetic and hadronic sections
    - Muon system



- Reconstruction of physics objects: leptons, jets, b-jets, tau-jets, and missing transverse energy

- Trigger emulation

- An event display

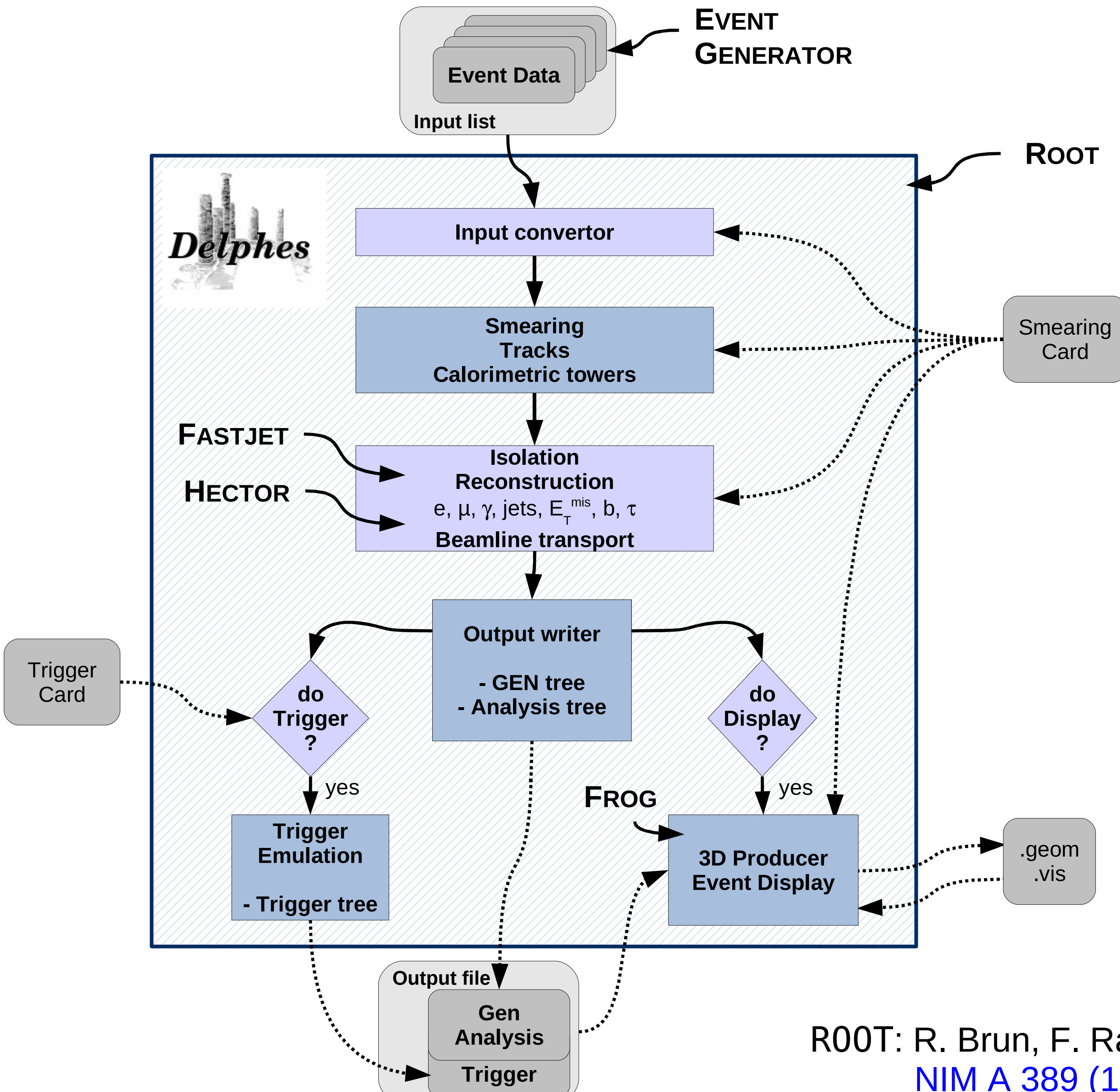
Delphes allows easy connection between theoretical and experimental *(distant)* worlds

S. Ovyn

C++ implementation of the simulation

C++/ROOT implementation

S. Ovyn

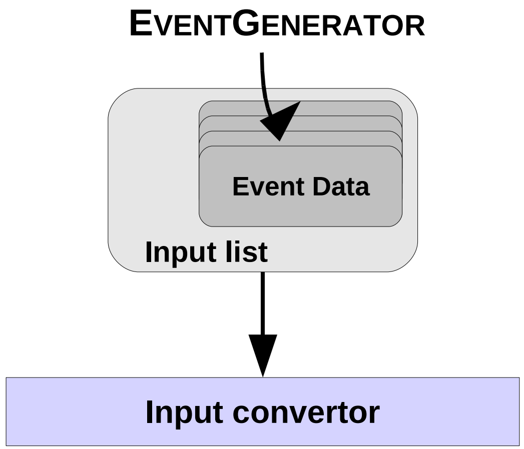


R00T: R. Brun, F. Rademakers, NIM A 389 (1997) 81-86.

Delphes flow

S. Ovyn

Interface:



First step of ***Delphes***: conversion of input events into a ROOT tree readable by the rest of the code

➡ Result of the conversion stored in a `GEN tree`

➡ Allow easy checks between various generators

```
./Delphes inputlist.list OutputRootFileName.root data/DetectorCard.dat
data/TriggerCard.dat
```

- Input events : ***Delphes*** is interfaced to standard file formats

- `StdHEP`
- `ROOT` files obtained with `h2root` (hbook)
- Les Houches Event Format
- `HepMC`

➡ Compatible with - MG/ME, Pythia, Herwig,...

Delphes flow

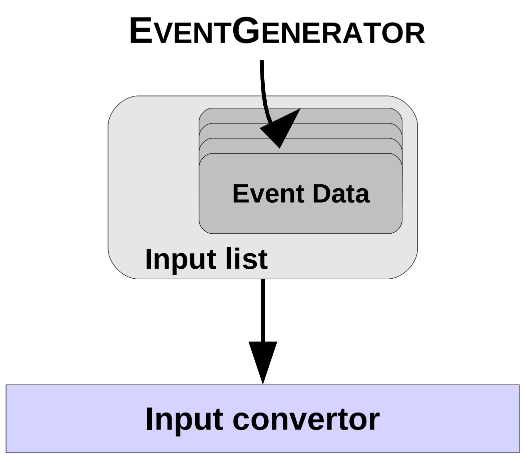
S. Ovyn

Interface:

The input file list contains the location of the input files with one file per line!

```
/nfs/cms/mass10/o/ovyn/Analysis2/Wt_2l/apatWt_an_1.root
/nfs/cms/mass10/o/ovyn/Analysis2/Wt_2l/apatWt_an_2.root
/nfs/cms/mass10/o/ovyn/Analysis2/Wt_2l/apatWt_an_3.root
```

➡ Automatic detection of the file extension (.lhe, .hep, .root, .hepmc)



- ***Delphes*** is driven by two input cards defining

(a) detector parametrisation
(b) trigger definitions
(c) parameters on physics objects (cuts,...)

➡ Default detector cards and trigger tables available for ATLAS & CMS experiments

```
./Delphes inputlist.list OutputRootFileName.root data/DetectorCard.dat data/TriggerCard.dat
```

Delphes flow

Output file format: ***Delphes*** yields realistic observables for all reconstructed high level objects in two formats:

1°) Analysis tree in ROOT files, using ExRootAnalysis, P. Demin

- `GEN` tree (Monte Carlo level information)
- `Analysis` tree (detector level information)
- `Trigger` tree (trigger acceptance)

2°) LHCO http://v1.jthaler.net/olympicswiki/doku.php

Column format

```
#   typ   eta   phi   pt   jmass   ntrk   btag   had/em   dummy   dummy
```

Typ: 0 = photon , 1 = electron , 2 = muon , 3 = tau-jet , 4 = jet , 6 = MET

Ntrk: number of tracks associated with the object. For of a lepton, this number is multiplied by the charge of the lepton.

For muons: The integer part is the identity of the jet that is closest ot this μ in ΔR

had/em: ratio of the hadronic versus electromagnetic energy deposited in the calorimeter cells associated with the object; it is typically > 1 for a jet and << 1 for an electron or γ.

For muons: the format is xxx.yy. The 'xxx' is ptiso, the summed p_T in a R=0.4 cone (excluding the μ). The 'yy' is etrat, is the ratio of the transverse energy in a 3×3 grid surrounding the μ to the p_T of the muon.

S. Ovyn

Trees in the output ROOT file

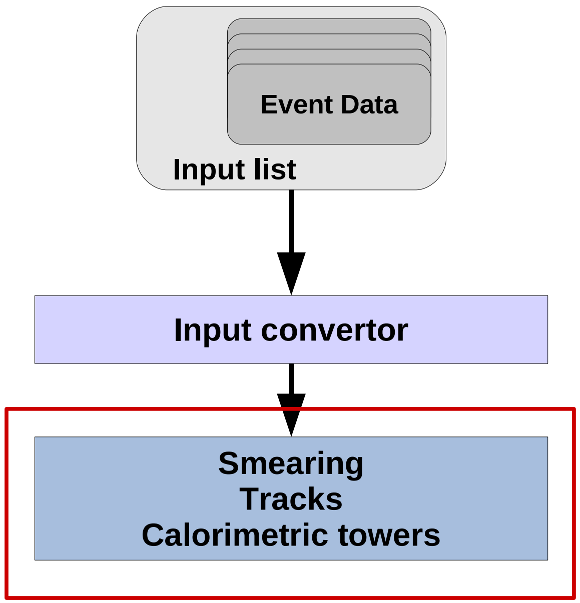
Delphes.root

- GEN
  Monte Carlo level information
  - Event
    - Event.Number
  - Particle
    - Particle.PT
    - Particle.E
    - Particle.Px
    - Particle.D1
    - Particle.D2
    - Particle.M1
    - Particle.M2
    - Particle.Status
    - Particle.PID

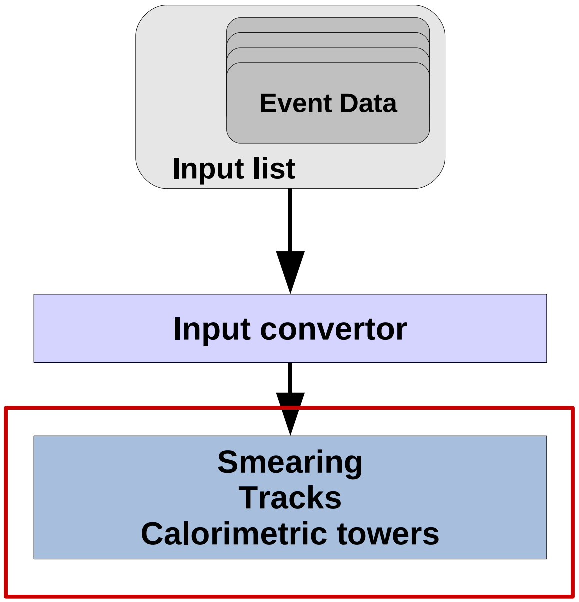
S. Ovyn

S. Ovyn

Low objects, detector level information



Delphes flow



Smearing: Response of each subdetector parametrised as a function of the energy:

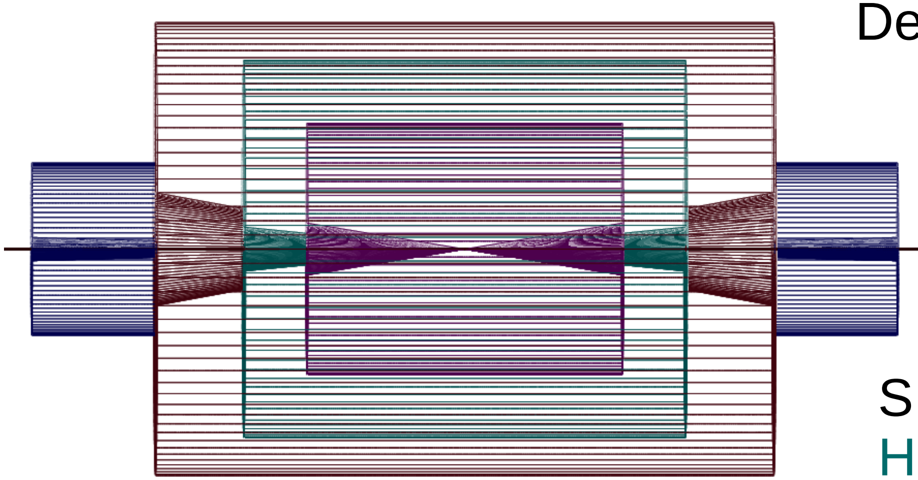
$$\frac{\sigma}{E} = \frac{S}{\sqrt{E}} \oplus \frac{N}{E} \oplus C$$

With different response to
- electromagnetic objects
- hadrons

Muons: smearing on the p_T

Parameters controllable using the input datacard

Schematic view of the *Delphes* detector



Detector extension in pseudorapidity
- tracker coverage
- central calorimeter coverage
- forward calorimeter coverage
- muon chambers coverage

S, N and C term of the ECAL, HCAL, FCAL

S. Ovyn

Low level objects : Tracks

For all charged particles in the tracking coverage, considering « energy flow »

The tracker is embedded in a magnetic B field

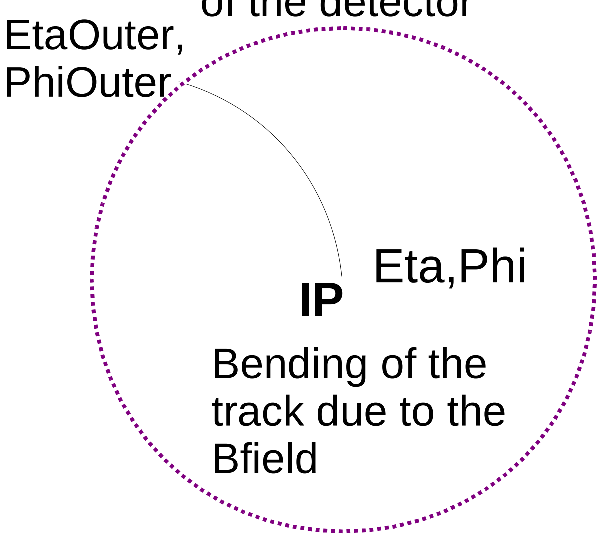
- Position of charged particles is modified
- The values of the length and radius of the tracker are important parameters

Impact of the η modification important when the p_T of the particle is too small to reach the central calorimeters

The inner and outer value of the tracks are stored in the `Tracks` branch of the ***Delphes*** `ROOT` file

➡ Eta, Phi, EtaOuter, PhiOuter

Transversal view of the detector



!! The particle energies are smeared according to the resolution of the calorimeter subdetector they reach !!

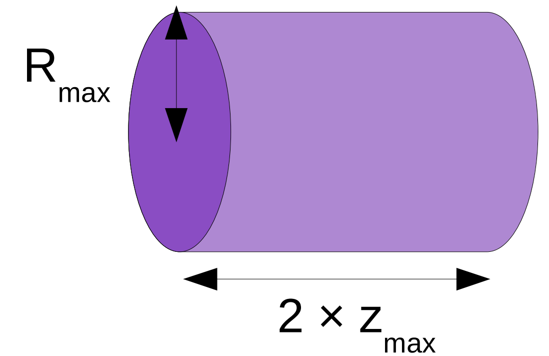
Delphes flow

S. Ovyn

Simulation of the magnetic field

Bx = By = 0 ➡ Exact calculation of the transport of a charged particle

The magnetic field is supposed to be
- homogeneous
- constant inside a cylinder of length $2 \times z_{max}$ and of radius R_{max}.



To make the code faster, the time of flight needed to exit the cylinder is computed

$$t_{max} = min(t_T, t_z) \quad \begin{cases} t_z \text{ such that } |z(t_z)| = z_{max} \\ t_T \text{ such that } R(t_T) = R_{max} \end{cases}$$

Bx ≠ 0 By ≠ 0 ➡ iterative method step by step until the particle exits the tracker region

Method slower than for a pure solenoidal B field

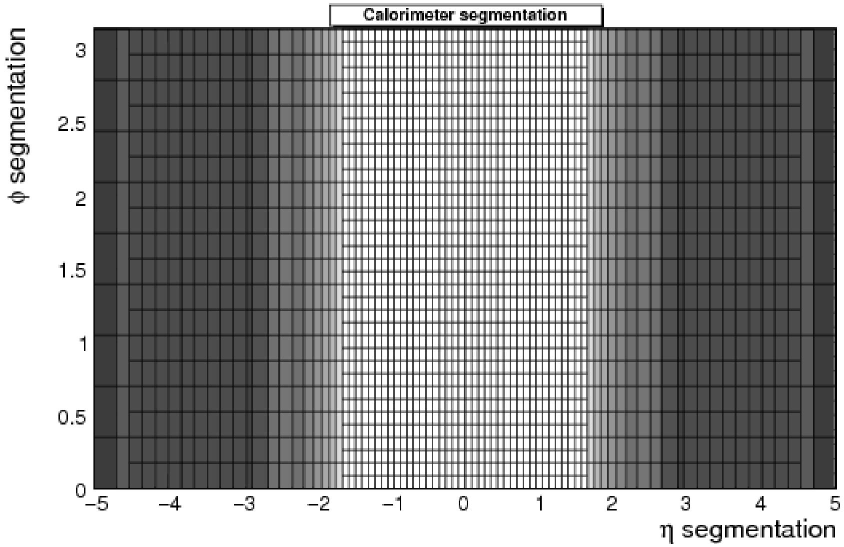
Disclamer: magnetic field of muon chambers such as for ATLAS not simulated with ***Delphes***

Delphes flow

S. Ovyn

Calorimetric towers

Segmentation in eta/phi



Need to enter in the datacard the number of towers in pseudorapidity as well as the edges of the towers in eta/phi

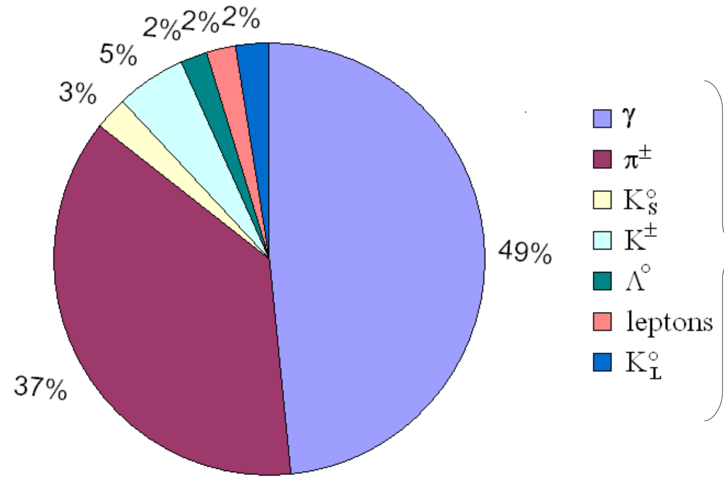
- Definition of positive towers only because the detector is supposed to be symmetric in η!
- ***Delphes*** assumes that all towers are similar in ϕ for a given η value

TOWER_number 40
TOWER_eta_edges 0. 0.087 0.174 0.261 0.348 0.435 0.522 0.609 0.696 0.783 0.870 0.957 1.044 1.131 1.218 1.305 1.392 1.479 1.566 1.653 1.740 1.830 1.930 2.043 2.172 2.322 2.500 2.650 2.868 2.950 3.125 3.300 3.475 3.650 3.825 4.000 4.175 4.350 4.525 4.700 5.000
TOWER_dphi 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 20 20

Delphes flow

S. Ovyn

- Charged and neutral final-state hadrons interact with the ECAL, HCAL and FCAL



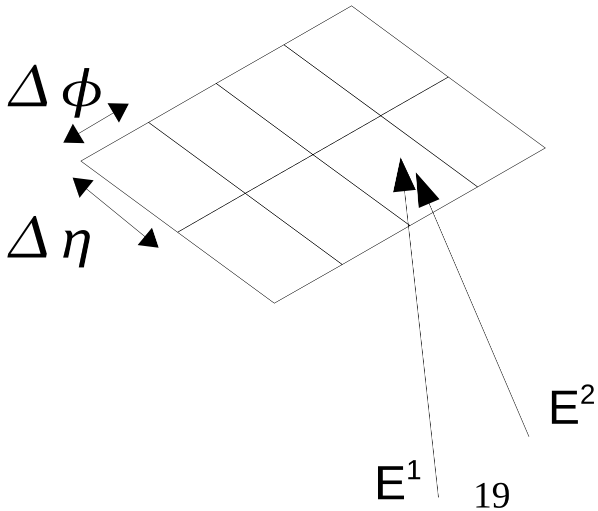
$$E_{HCAL} = E \times F$$

$$E_{ECAL} = E \times (1 - F)$$

Smearing of particles performed using the expected fraction of the energy, determined according to their decay products, that would be deposited into the ECAL (E_{ECAL}) and into the HCAL (E_{HCAL})

- Summing energy of multiple impacts in identical towers

$$E^{tower}_{ECAL} = E^{1}_{ECAL} + E^{2}_{ECAL} \quad \text{and} \quad E^{tower}_{HCAL} = E^{1}_{HCAL} + E^{2}_{HCAL}$$



➡ Smearing of the corresponding energies

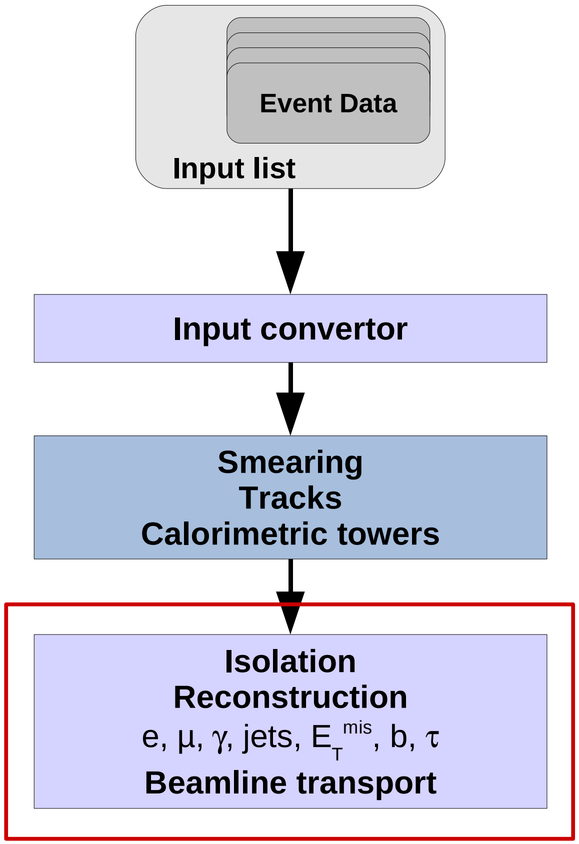
$$E^{tower} = E^{tower}_{SHCAL} + E^{tower}_{SECAL}$$

Trees in the output ROOT file

Delphes.root

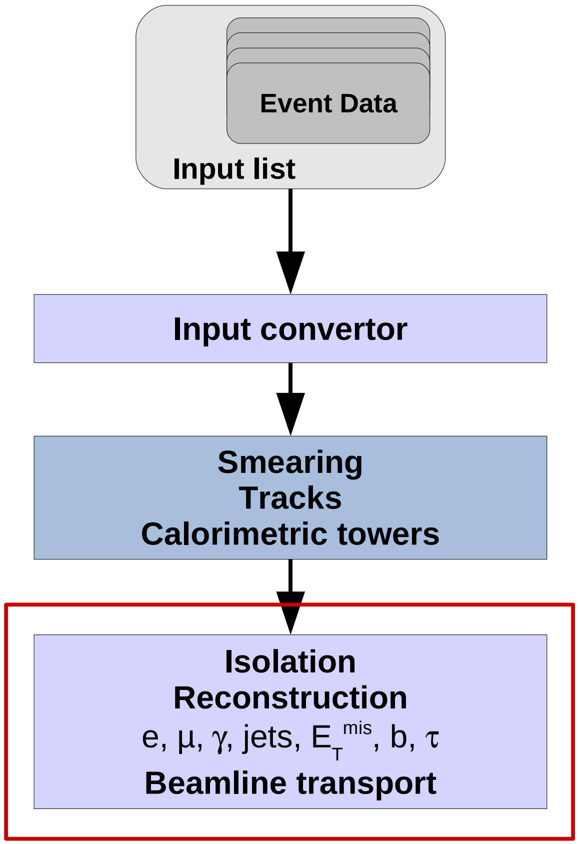
- GEN — Monte Carlo level information
  - Event
  - Particle
- Analysis — Analysis level information
  - CaloTower
    - CaloTower.E
    - CaloTower.ET
    - CaloTower.E_em
    - CaloTower.E_had
    - CaloTower.Eta
    - CaloTower.Phi
  - Tracks
    - Tracks.Charge
    - Tracks.E
    - Tracks.PT
    - Tracks.Phi
    - Tracks.Eta
    - Tracks.PhiOuter
    - Tracks.EtaOuter
    - Tracks.Px
    - Tracks.Py
    - Tracks.Pz

S. Ovyn



High level objects, final reconstructed information

Delphes flow



Photons :

- reconstructed if they fall into the tracker coverage
- eta/phi variables correspond to the impact point in the calorimeter

Electrons and muons :

- reconstructed if they fall into the tracker coverage
- muons do not leave a deposit in the calorimeters

Isolation implemented:

- Isolation of charged particles using tracking information

→ No other charge particles wit p_T > 2 GeV within a cone

$$\Delta R = \sqrt{\Delta \eta^2 + \Delta \phi^2} < 0.5$$

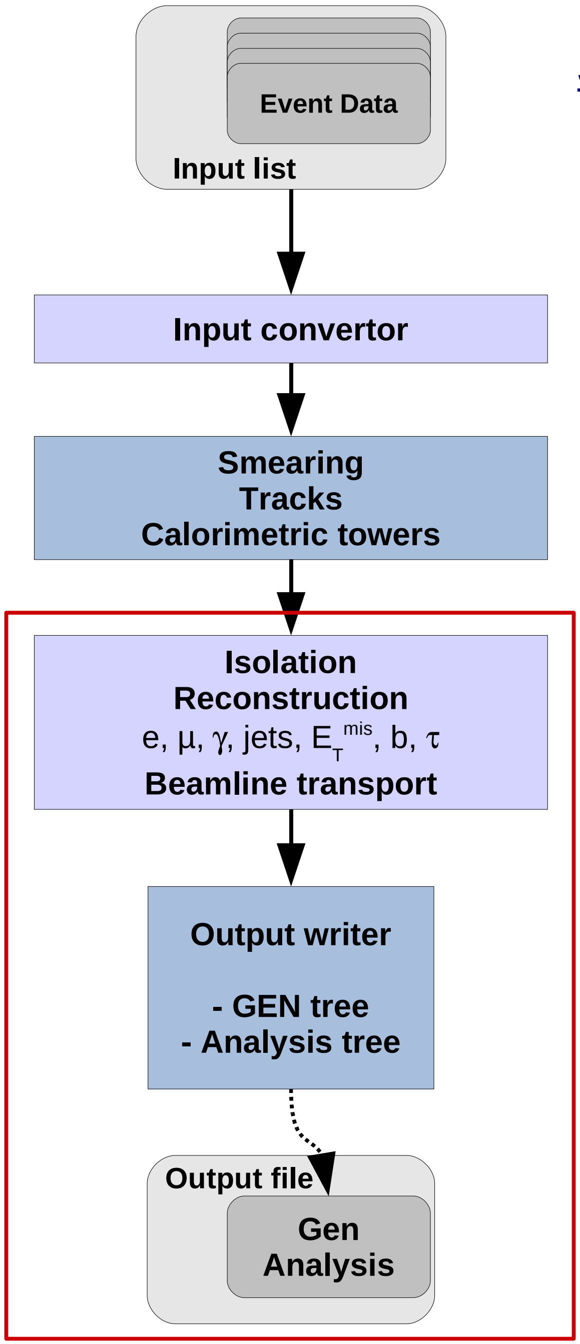
UCL
Université catholique de Louvain

Delphes flow

Delphes

S. Ovyn

Motivations
Simulation
Interface
Tower-tracks
photon-e/μ
jets
tau-jets-MET
Forward det.
Validation
BUT also...
Tutorial
Conclusion



Jets :

- Treatement of particles which hadronise using jet reconstruction

- Uses reconstruction algorithms implemented in FastJet

  - CDF jet algorithm (cone)
  - CDF Midpoint algorithm
  - SIS Cone jets
  - Longitudinally invariant k_t jets
  - Cambridge / Aachen jets
  - Anti k_t jets

Jet algorithms differ
- in their sensitivity to soft particles or collinear splittings
- their computing speed performances.

FastJet: M. Cacciari, G.P. Salam, Phys. Lett. B 641 (2006) 57.

Delphes flow

Choice of the jet algorithm, jet parameters in the detector datacard

In addition to the standard E, Px, Py, Pz, Eta, Phi variables, the jet collections also contains

- The number of tracks associated to the jet
- The ECAL value ont the HCAL value of the jet
- A b-flag indicated if the jet has been b-tagged

- *b*-tagging
  - identical in the entire tracker coverage
  - independent of the p_T of the jet
  - efficiency controlable in the datacard (default= 40%)
  - mis-identification of *c* (10%) and light jets (1%)

<u>Final remark</u>

The user can choose if a perfect energy recontruction is applied in the tracker coverage (perfect energy flow).
If not, jets are taking as input the calorimetric towers

Delphes flow

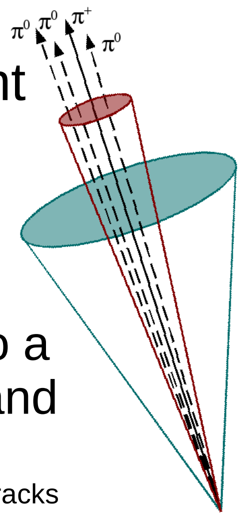
UCL Université catholique de Louvain

Delphes

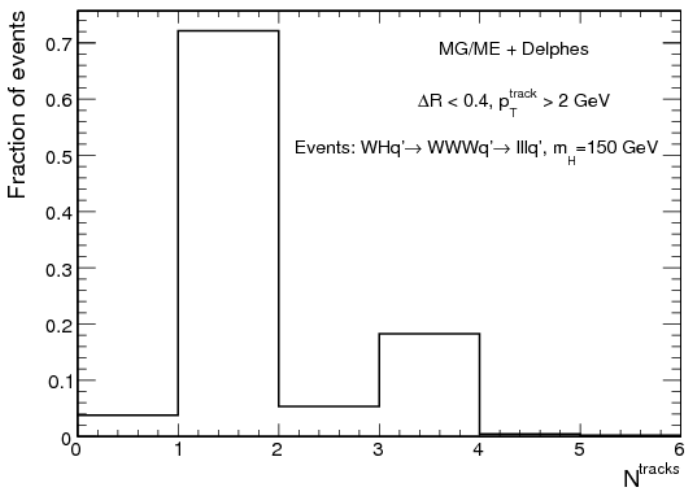
S. Ovyn

Tau-jets reconstruction:

Selected from the jet collection using a procedure consistent with the one applied in a full detector simulation



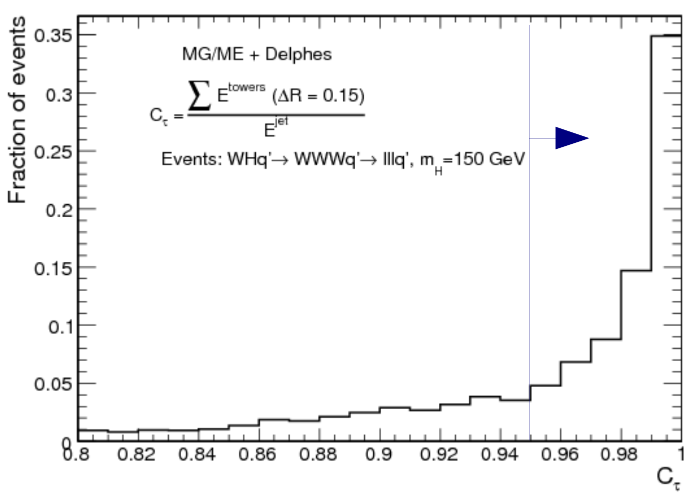
2) Requirement of tracking isolation



Number of tracks associated to a particle with p_T>2 GeV is one and only one in a cone of radius R_{tracks}

- 3-prong τ dropped.
- Cone should be entirely incorporated into the tracker

1) Use of the narrowness of the tau-jet



C_τ = sum of the energy of towers in a small cone of radius R_{em} around the jet axis, divided by the energy of the reconstructed jet.

C_τ expected to be large

Delphes flow

- Measurement of Missing Transverse Energy (MET) in an ideal detector

Momentum conservation imposes the transverse momentum of the observed final state (p_T^{obs}) to be equal to the vector sum of the invisible particles,

$$\vec{p}_T = \begin{pmatrix} p_x \\ p_y \end{pmatrix} \text{ and } \begin{cases} p_x^{miss} = -p_x^{obs} \\ p_y^{miss} = -p_y^{obs} \end{cases}$$

- MET reconstruction in ***Delphes***

Missing Transverse Energy (MET) calculation based on the calorimetric towers:

$$\vec{E}_T^{\,miss} = -\sum_i^{towers} \vec{E}_T(i)$$

Remarks:

- dead channels, misalignment, noisy towers, cracks of the detector that worsen directly the MET not taken into account
- based on the calorimetric towers only

➡ muons not used to reconstruct MET

S. Ovyn

Trees in the output ROOT file

Delphes.root

- **GEN** — Monte Carlo level information
  - Event
  - Particle
- **Analysis** — Analysis level information
  - CaloTower
  - Tracks
  - Electron
    - Charge
    - IsolFlag
    - PhiCalo
    - EtaCalo
  - Muon
    - Charge
    - IsolFlag
    - PhiCalo
    - EtaCalo
  - Jet
    - NTracks
    - Btag
  - TauJet
    - NTracks
    - Charge
  - ETmis
    - ET
    - Phi
    - Px
    - Py
  - Photon

Common leaves to Electron, Muon, Jet, TauJet and Photon branches

- E
- PT
- Phi
- Eta
- Px
- Py
- Pz
- EHoverEE

S. Ovyn

Motivations
Simulation
Interface
Tower-tracks
photon-e/μ
jets
tau-jets-MET
Forward det.
Validation
BUT also...
Tutorial
Conclusion

Very forward detector information

Near-beam components

Most of recent experiments in HE physics have additional instrumentation along the beamline

➡ In addition to the central detector, ***Delphes*** includes

- Forward detectors to extend the eta coverage to higher values

  e.g. : Zero Degree Calorimeters

- (very) forward near-beam detectors

Central detector coverage

CMS tracking : $0 < |\eta| < 2.5$

CMS calorimetry : $0 < |\eta| < 5$

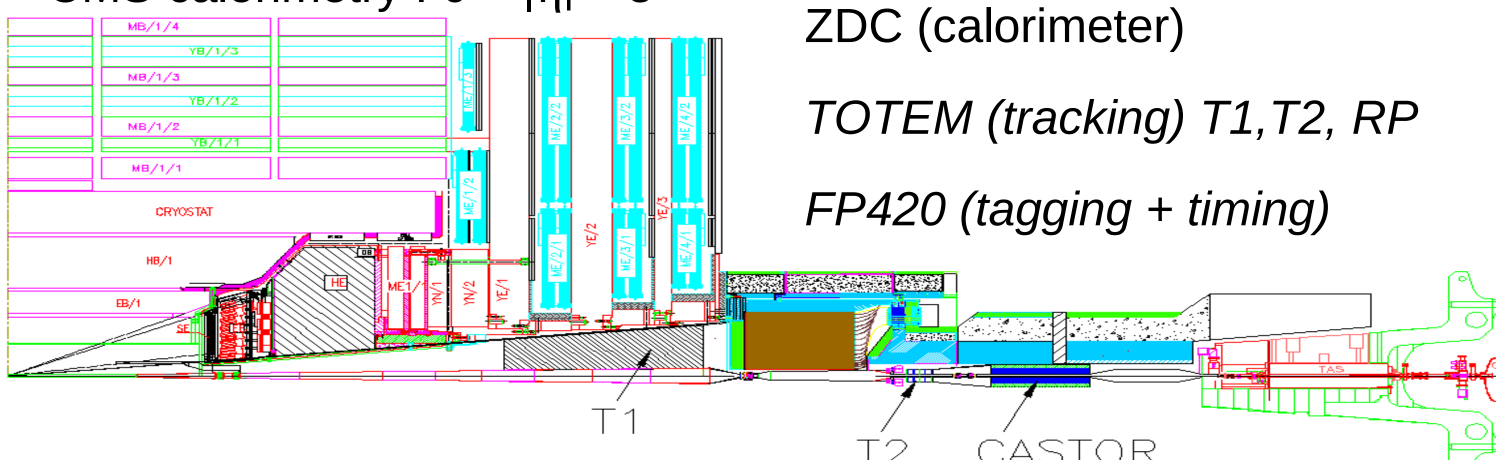
Very forward extensions

CASTOR (calorimeter)

ZDC (calorimeter)

TOTEM (tracking) T1,T2, RP

FP420 (tagging + timing)



CMS & TOTEM Coll [CERN/LHCC 2006-039/G-124]

Near-beam components

S. Ovyn

Motivations
Simulation
Interface
Tower-tracks
photon-e/μ
jets
tau-jets-MET
Forward det.
Validation
BUT also...
Tutorial
Conclusion

Most of recent experiments in HE physics have additional instrumentation along the beamline

➡ In addition to the central detector, ***Delphes*** includes

- Forward detectors to extend the eta coverage to higher values

e.g. : Zero Degree Calorimeters

- (very) forward near-beam detectors

Central detector coverage

CMS tracking : $0 < |\eta| < 2.5$

CMS calorimetry : $0 < |\eta| < 5$

Very forward extensions

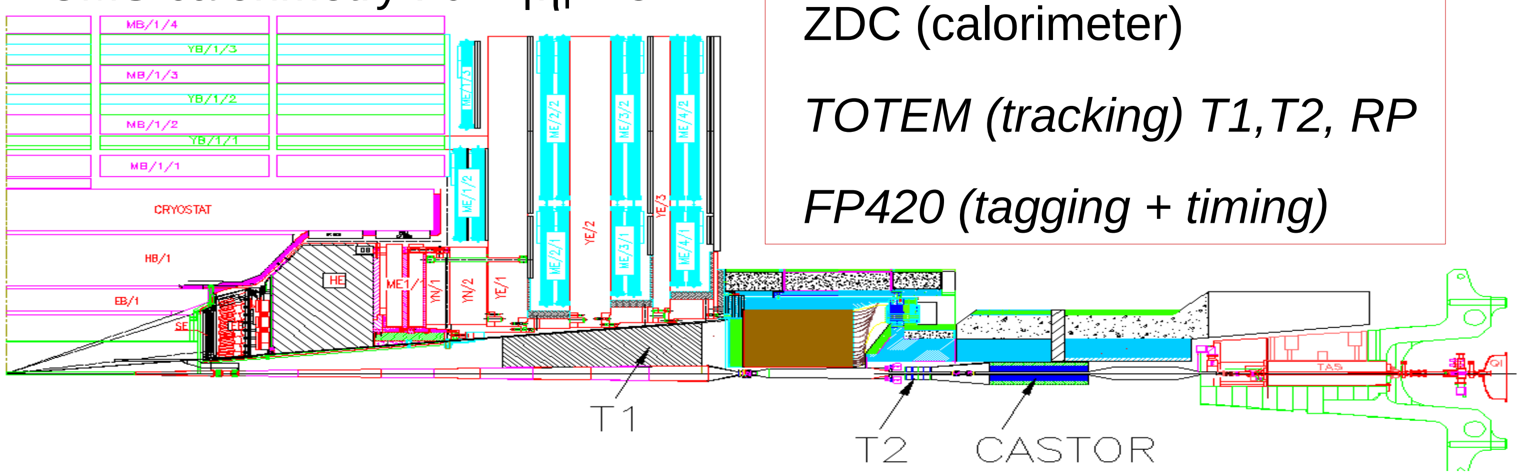
CASTOR (calorimeter)

Delphes

ZDC (calorimeter)

TOTEM (tracking) T1,T2, RP

FP420 (tagging + timing)



CMS & TOTEM Coll [CERN/LHCC 2006-039/G-124]

UCL
Université catholique de Louvain

Delphes

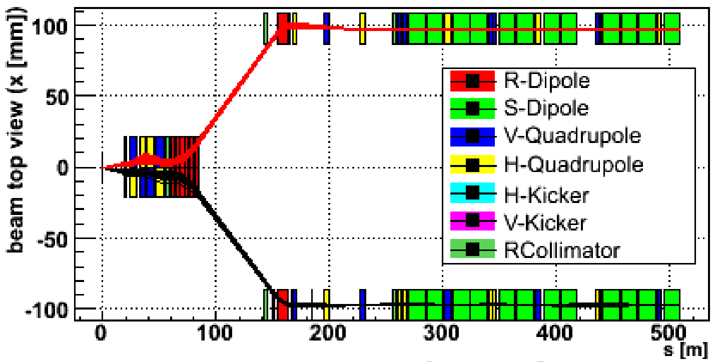
HECTOR implementation

S. Ovyn

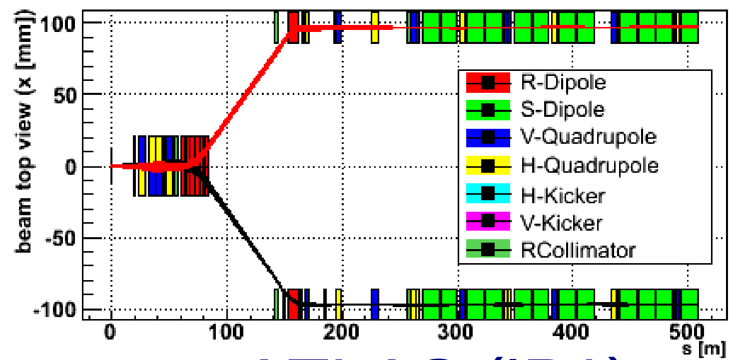
Motivations
Simulation
Interface
Tower-tracks
photon-e/μ
jets
tau-jets-MET
Forward det.
Validation
BUT also...
Tutorial
Conclusion

- ***Delphes*** uses HECTOR to perform particle transport in beamlines

X. Rouby, J. de Favereau and K. Piotrzkowski, JINST 2(2007) P09005

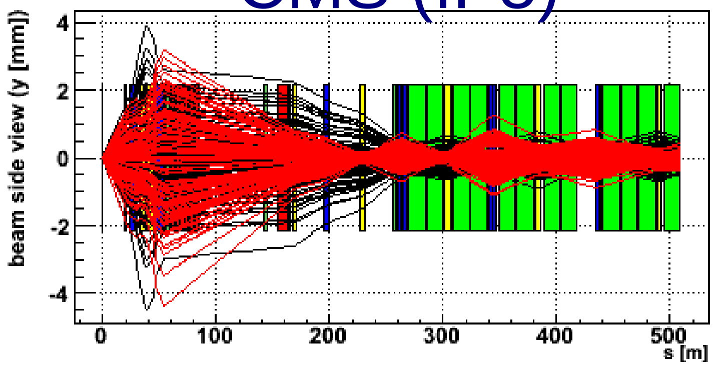


top

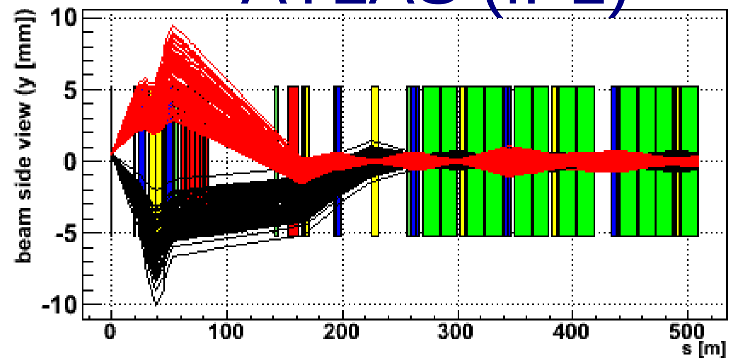


CMS (IP5)

ATLAS (IP1)



side



Horizontal crossing plane

Vertical crossing plane

Input needed:

- effective field strength / length
- magnetic position/aperture

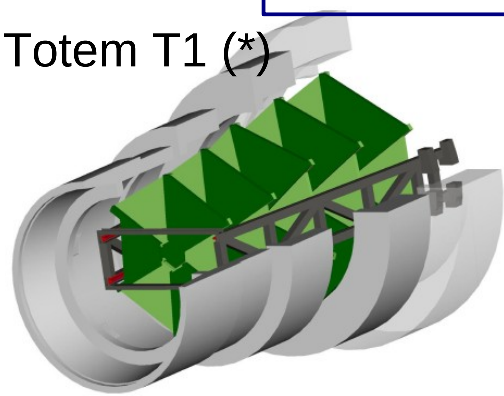
→ data/LHCB1IR5_v6.500.tfs
data/LHCB2IR5_v6.500.tfs

Acceptance of the very forward and near-beam detectors are easily modified using the Detector card

Forward detectors around CMS

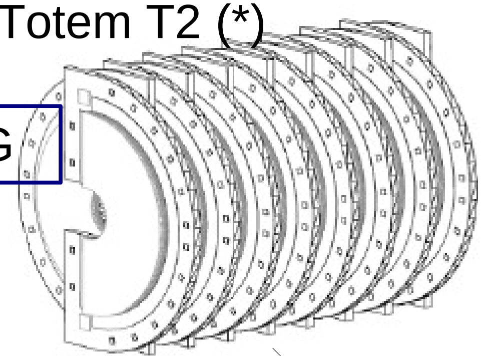
S. Ovyn

Totem T1 (*)

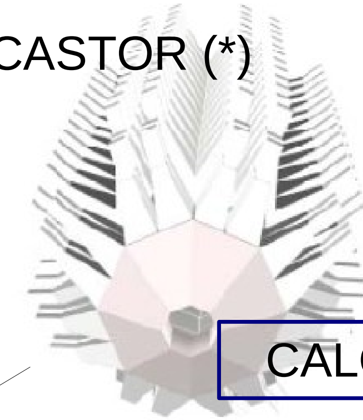


TRACKING

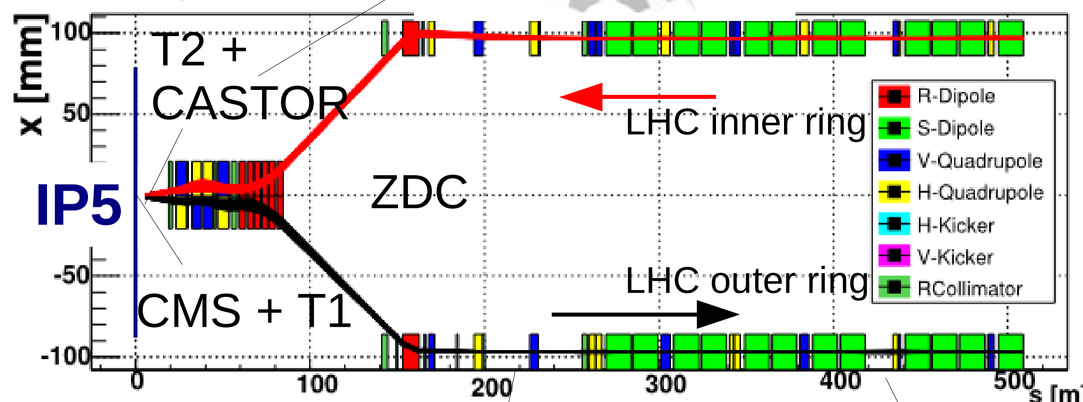
Totem T2 (*)



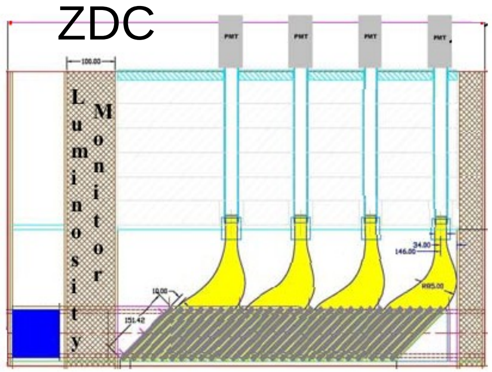
CASTOR (*)



CALORIMETRY



ZDC

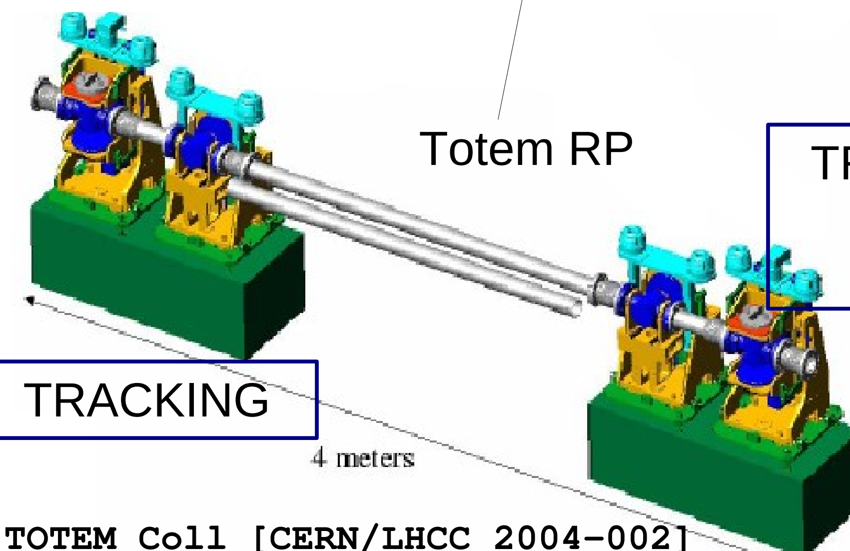


CALORIMETRY
neutrals

Totem RP

FP420

TRACKING
&
TIMING



TRACKING

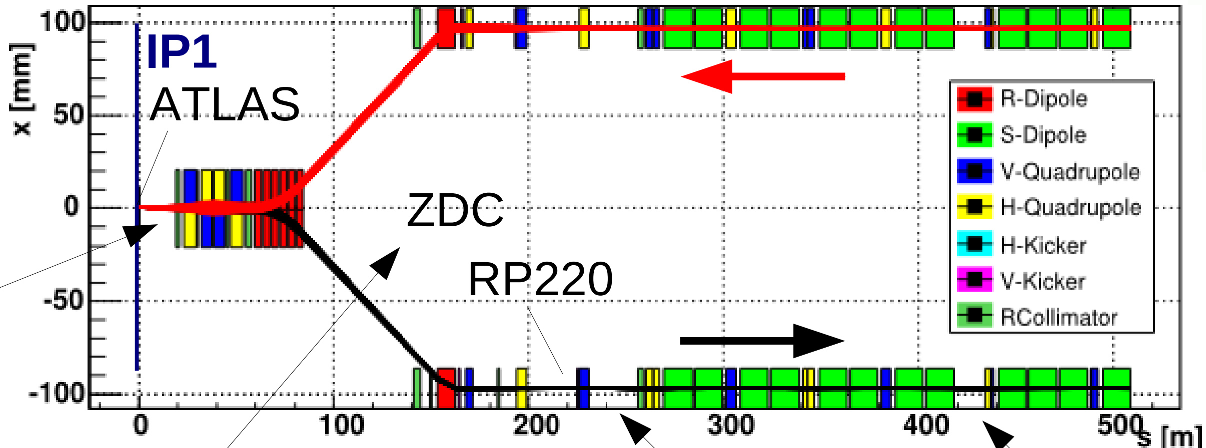
TOTEM Coll [CERN/LHCC 2004-002]

(*) Not implemented in Delphes

Forward detectors around ATLAS

S. Ovyn

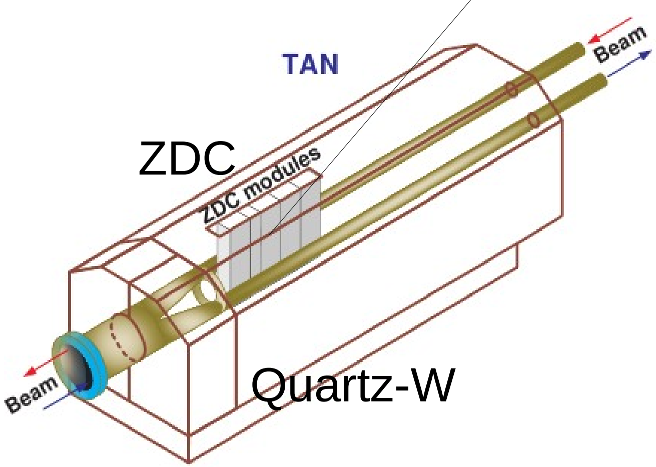
Adapted from
Rouby, de Favereau, Piotrzkowski [JINST 2 P09005]



LUMI

LUCID (*)

Cerenkov

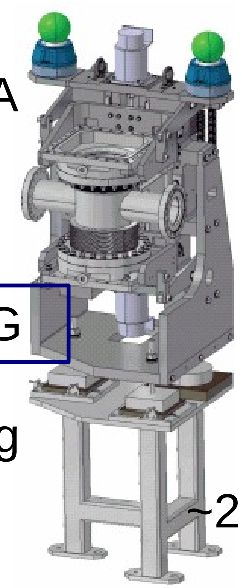


ZDC

Quartz-W

CALORIMETRY
neutrals

ALFA



TRACKING

Scintillating
fibers

~2009

FP420

Si + Cerenkov

TRACK
&
TIMING

~2010

(*) Not implemented in Delphes

Trees in the output ROOT file

Delphes.root

- **GEN** — Monte Carlo level information
  - Event
  - Particle
- **Analysis** — Analysis level information
  - CaloTower
  - Tracks
  - Electron
  - Muon
  - Jet
  - TauJet
  - ETmis
  - Photon
  - FP420hits
  - RP220hits
  - ZDChits
    - ZDChits.E
    - ZDChits.Eta
    - ZDChits.Phi
    - ZDChits.PT
    - ZDChits.Px
    - ZDChits.Py
    - ZDChits.Pz
    - ZDChits.T
    - ZDChits.pid
    - ZDChits.side

Common leaves to FP420hits and RP220hits branches

- E
- S
- T
- side
- Tx
- Ty
- X
- Y

Validation: jet resolution

S. Ovyn

Validation procedures using CMS-like detector parameters

CMS resolution from: The CMS Collaboration, CERN/LHCC 2006-001.

The majority of interesting processes contain jets in the final state.

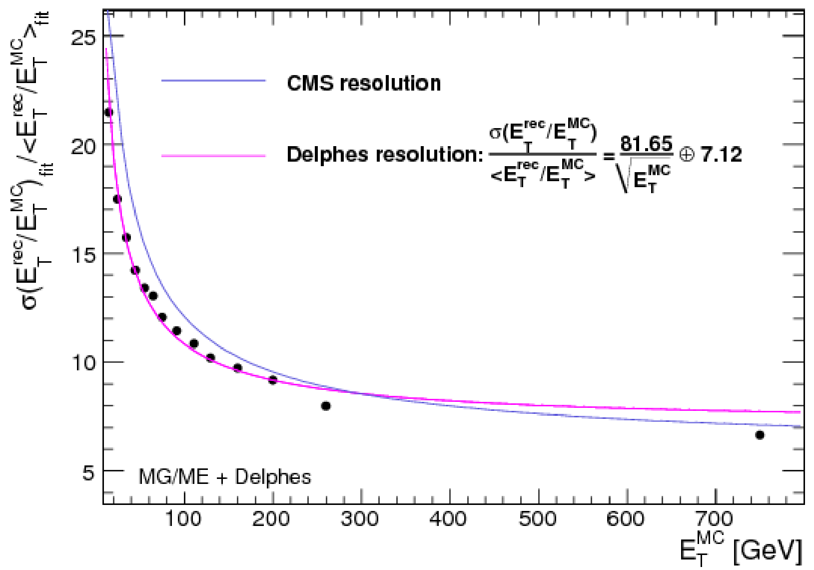
➡ The jet resolution is therefore a crucial point

Sample used: pp→gg

- Arranged in 14 bins of gluon p_T .
- In each p_T bin, Delphes jets are matched to the closest GEN jet using

$$\Delta R = \sqrt{(\eta^{rec} - \eta^{MC})^2 + (\phi^{rec} - \phi^{MC})^2} < 0.25$$

jet clustering algorithm (jetclu) with R = 0.7



- E_T^{rec}/E_T^{MC} histograms fitted with a Gaussian distribution in the interval ±2 rms centred around the mean value. The resolution in each p_T bin is obtained by

$$\frac{\sigma\left(\frac{E_T^{rec}}{E_T^{MC}}\right)_{fit}}{\left\langle\frac{E_T^{rec}}{E_T^{MC}}\right\rangle_{fit}}(\hat{p}_T(i))$$

An excellent agreement is obtained comparing values of ***Delphes*** with the expectations of the general purpose CMS detector

Validation: MET resolution

Delphes

S. Ovyn

Validation procedures using CMS-like detector parameters

CMS resolution from: The CMS Collaboration, CERN/LHCC 2006-001.

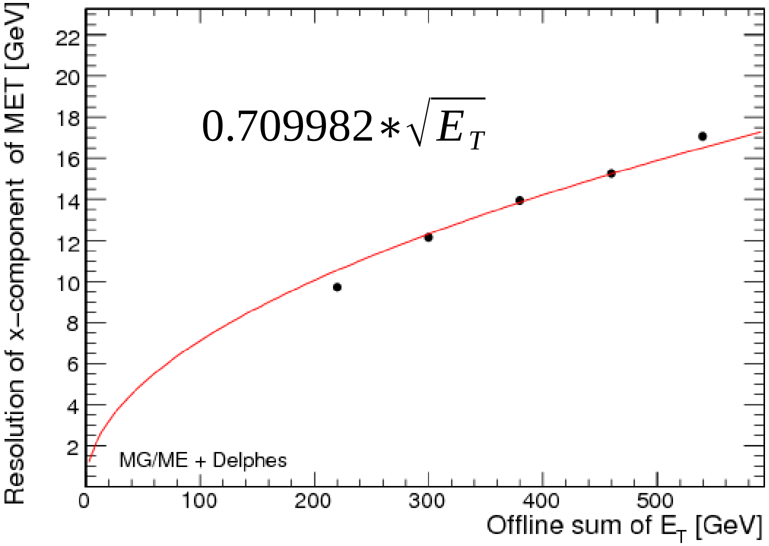
HEP detectors designed to be as much hermetic as possible

➡ MET resolution is a crucial point

Sample used: pp→gg : muon contribution is negligeable

- Arranged in 5 bins of of scalar E_T sum

Quality of the MET checked via the resolution on its horizontal component E_x^{miss}



- Difference between the ***Delphes*** and the GEN E_x^{miss} fitted with a Gaussian distribution in the interval ± 1 rms centred around the mean value.

Value expected by CMS: $\sigma_x = (0.6-0.7)\ \sqrt{E_T}\ GeV^{1/2}$

An excellent agreement is obtained comparing values of ***Delphes*** with the expectations of the general purpose CMS detector

S. Ovyn

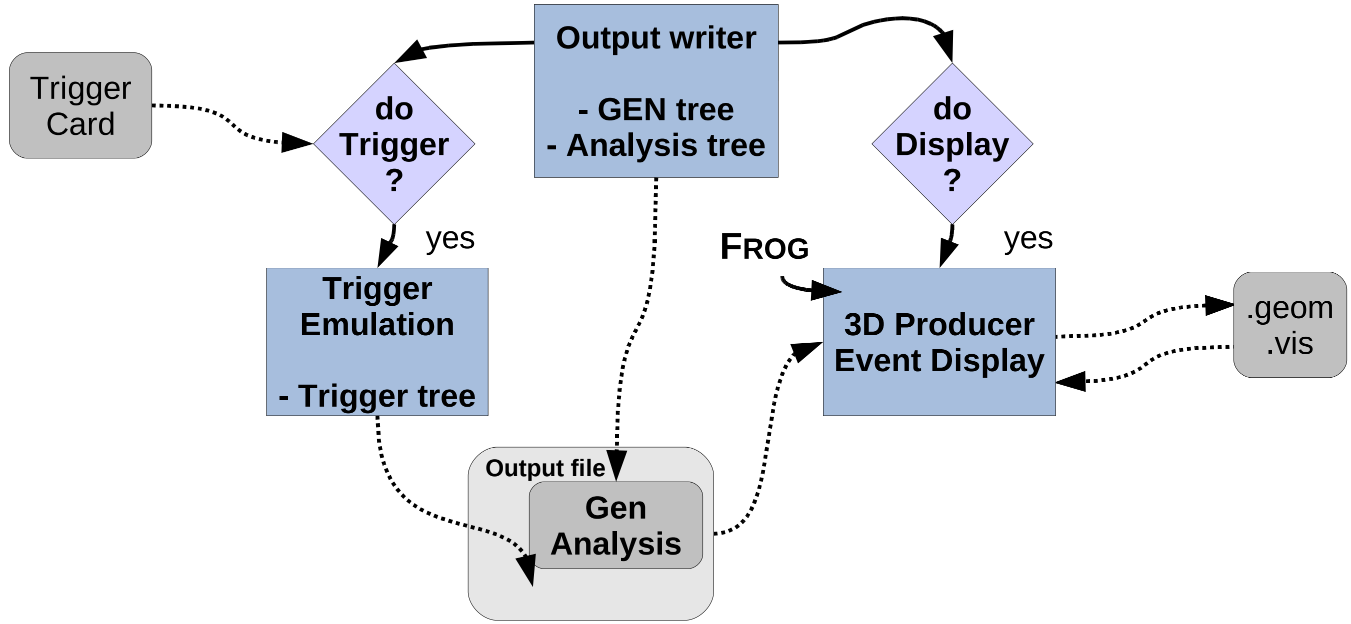
Additional features

UCL
Université catholique de Louvain

Delphes

S. Ovyn

Additional features



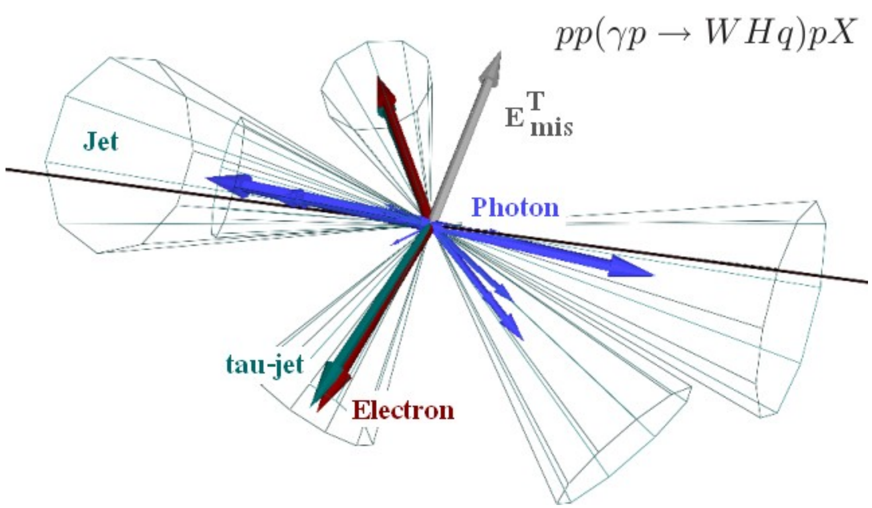
Trigger emulation

Application of user-defined trigger selection using the Trigger card

Result of the ***Delphes*** trigger selection is stored in the « Trigger tree » in the output root file

3D Event Display

FROG interfaced to ***Delphes***



FROG: L. Quertenmont, V. Roberfroid, arXiv:0901.2718v1[hep-ex]

Trigger emulation

- Trigger selection in a real experiment

New physics often characterised by low σ of new physics compared to values of Standard Model ones

➡ High statistics are required for data analyses ➡ high luminosity

BUT only a tiny fraction of the observed events can be stored for subsequent offline analyses,

↳ - large data rejection factor using dedicated algorithms
- Selection should be fast and very efficient

A trigger emulation is included in ***Delphes***, using a fully parametrisable trigger table

```
Inclusive electron  >> ELEC1_PT:   '29'
di-electron         >> ELEC1_PT:   '17'   &&   ELEC2_PT:   '17'
```

- select events containing objects (i.e. jets, particles,met) with a p_T above some threshold.
- Logical combinations (AND) of several conditions are also possible.
- Default trigger tables available for ATLAS & CMS experiments

3D Event Display

Delphes

S. Ovyn

FROG: L. Quertenmont, V. Roberfroid, arXiv:0901.2718v1[hep-ex]

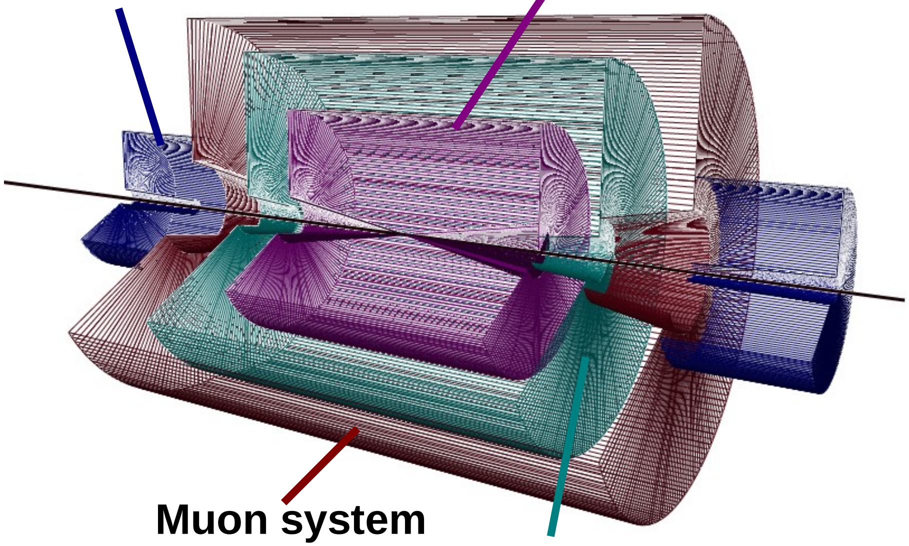
Visualisation is useful to convey information about the detector layout and the event topology in a simple way.

➡ The *Fast and Realistic OpenGL Displayer* FROG interfaced in ***Delphes***

Reminder

- Detector assumed to be strictly symmetric around the beam axis.

- Only the geometrical coverage: towers are not drawed



Utility of the detector visualisation

- Communication purpose
- Geometric coverage of the different detector subsystems clearly visible.

3D Event Display

FROG: L. Quertenmont, V. Roberfroid, arXiv:0901.2718v1[hep-ex]

Visualisation is useful to convey information about the detector layout and the event topology in a simple way.

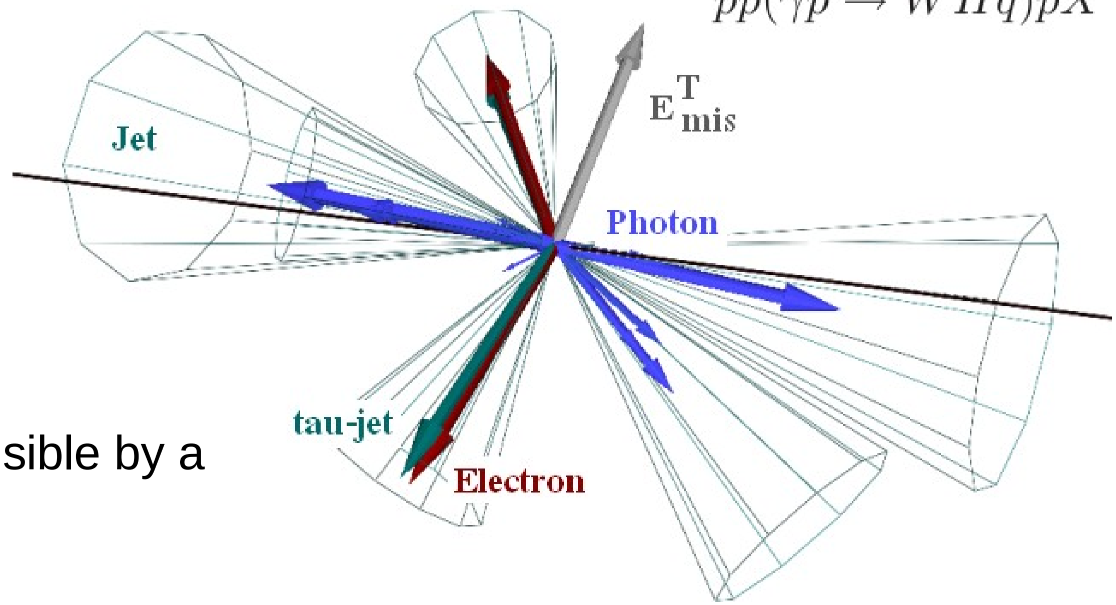
➡ The *Fast and Realistic OpenGL Displayer* FROG interfaced in ***Delphes***

S. Ovyn

- Visibility of each objects ($e^{\pm}$, $\mu^{\pm}$, $\tau^{\pm}$, jets, MET) enhanced by a colour coding.

- Kinematics information is visible by a simple mouse action.

$pp(\gamma p \to WHq)pX$



Utility of the event visualisation

- Deeper understanding of interesting physics processes

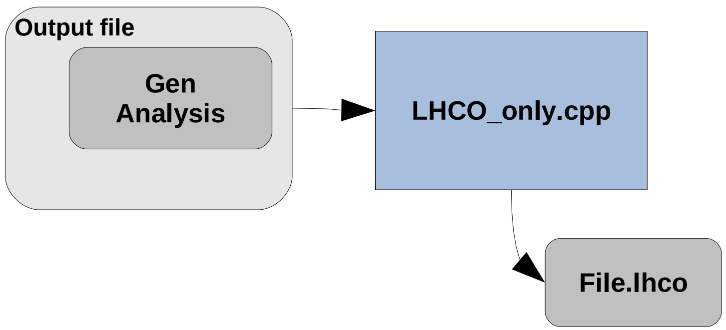
Additional features

S. Ovyn

Forgotten to run the trigger selection, the LHCO output or the preparation for the event visualitsation?

➡ Stand-alone running programs are available

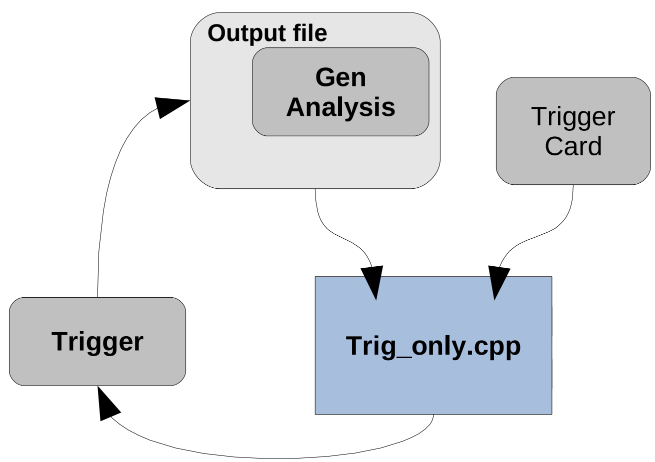
LHCO running



The input Delphes root file should contain all the GEN and Analysis information

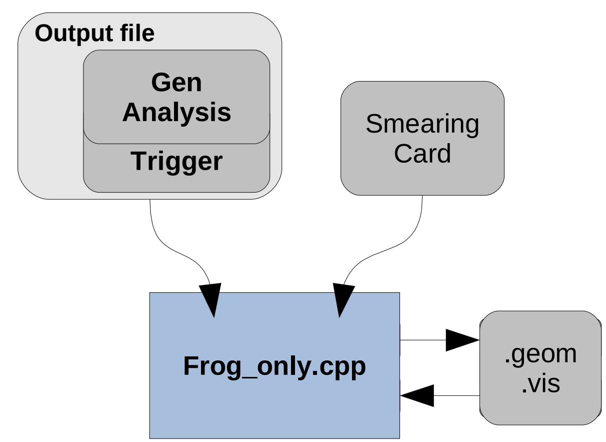
➡ Creation of the text file

Running the trigger



Using the ***Delphes*** file, the code adds the Trigger branch in the input file

Running the preparation for FROG

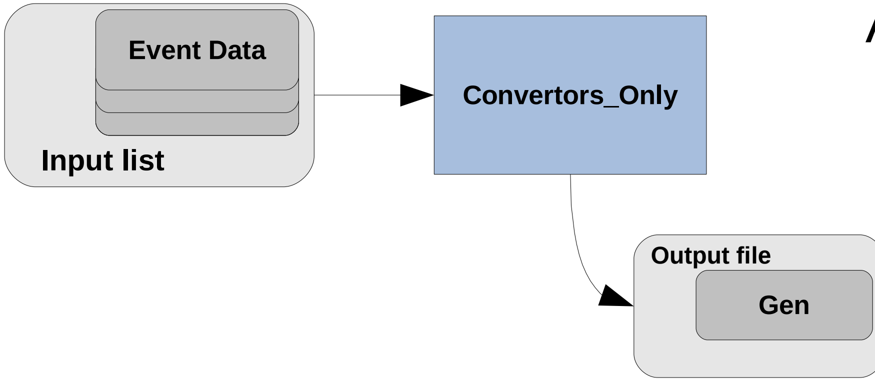


Using the ***Delphes*** file, the code creates the `.vis` files

Additional features

S. Ovyn

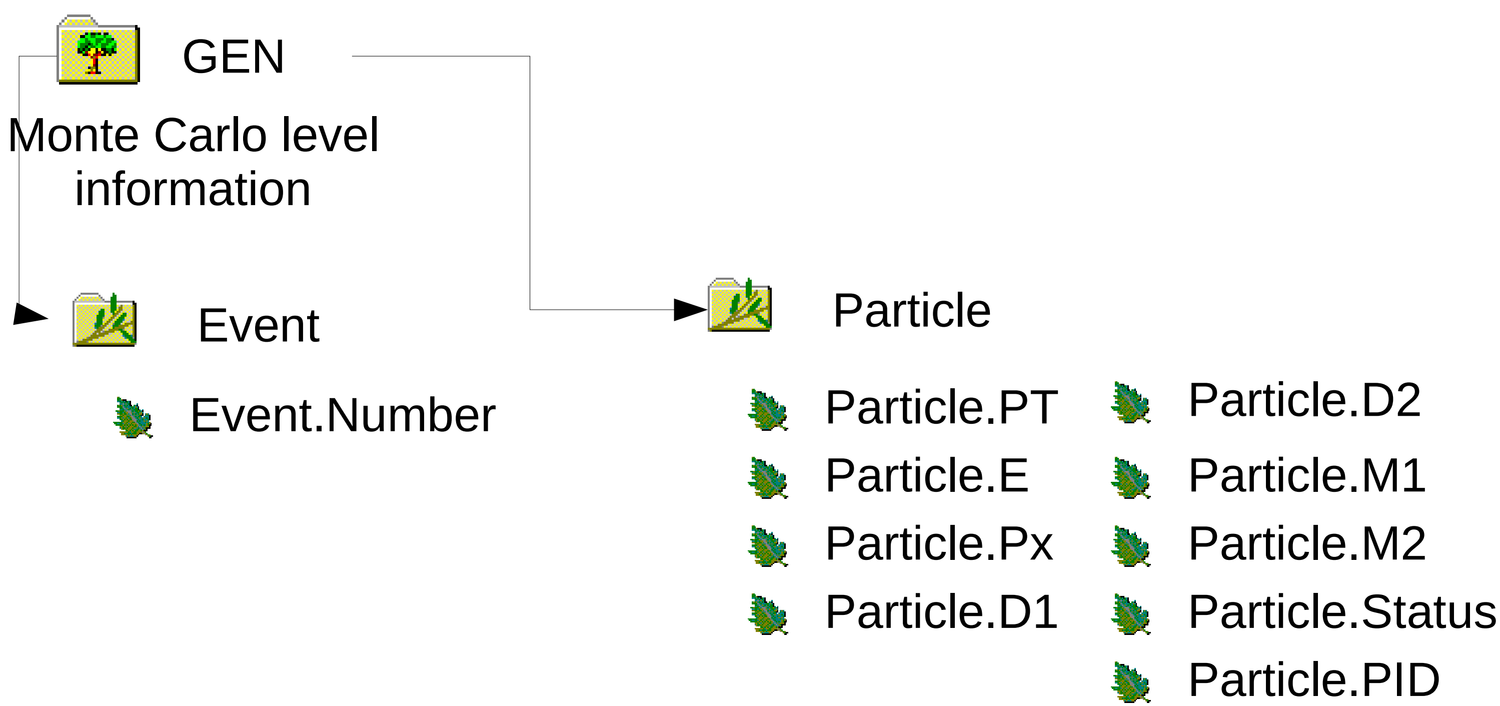
Convertor



All types of input files accepted by ***Delphes***

- StdHEP
- ROOT files
- Les Houches Event Format
- HepMC

Allow easy checks between various generators



Small tutorial...

Getting started : download

S. Ovyn

Code download

from the website : « *download* » link

Delphes tar-ball is self-sufficient, it contains every dependencies needed for the physics.

or from a command line :

```
wget http://www.fynu.ucl.ac.be/users/s.ovyn/Delphes/files/Delphes_V_1.7.tar.gz
```

Requirements

A recent working ROOT version (http://root.cern.ch)

ROOT: R. Brun, F. Rademakers, NIM A 389 (1997) 81-86.

Delphes has been developped on ROOT > 5.18 on Linux with GNU gcc/g++ > 4.1.2, but any recent version should be fine.

For Mac-OSX users:

In ***Delphes***' genMakefile.tcl, you should add "`-Dmacos` " in the `CXXFLAGS` definition (line 219):

```
CXXFLAGS += $(ROOTCFLAGS) -Dmacos -DDROP_CGAL -I. ...
```

Getting started : download

S. Ovyn

Requirements

Checking ROOT installation

```
echo $ROOTSYS
```

If empty, check that the environment variables are defined.

In *bash* shell, check that these variables are in the .bashrc:

(e.g. assuming ROOT is in /usr/bin/root

```
export ROOTSYS=/usr/bin/root
export PATH=$PATH:$ROOTSYS/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ROOTSYS/lib
```

Test it :

```
root
```

If the FROG event display is to be run:

3D-OpenGL libraries are not included in the `tar.gz`, but required only if FROG is used. These libraries can be downloaded from here: http://curl.haxx.se/download.html

More on FROG requirements:

http://projects.hepforge.org/frog/index.php?page=Starting.php

Getting started : compile

Untar – decompress the code sources

```
tar -xzf Delphes_V_1.7.tar.gz
```

Compile the sources

```
cd Delphes_V_1.7
./genMakefile.tcl > Makefile
make
>> Compiling tmp/Utilities/ExRootAnalysis/src/BlockClassesDict.cc
>> Compiling tmp/src/TreeClassesDict.cc
...
>> Building Analysis_Ex
Delphes has been compiled
Ready to run
```

Many lines are printed during the compilation.

In particular, the dependencies (like FastJet, mcfio, stdhep) lead to a few warning messages. This is normal and harmless.

Getting started : samples

S. Ovyn

Input files from MC generator

Suggested samples for this introduction:

```
wget http://www.fynu.ucl.ac.be/users/s.ovyn/Delphes/files/tt_jj_small.hep.tar.gz
tar -xzf tt_jj_small.hep.tar.gz
mv samples/* .
```

These events are $\gamma p \rightarrow ttX$,

- generated with MadGraph/MadEvent and
- hadronised with Pythia
- saved into StdHEP file format (*hep).

List of input files:

```
ls *list
TEST_small_tt.list
```

→

TEST_small_tt.list :

```
tt_jj_small.hep
```

- text file containing one input data file per line
- all data files must be of the same type

Getting started : run

S. Ovyn

Running ***Delphes*** :

```
./Delphes
Usage: ./Delphes input_file output_file [detector_card] [trigger_card]
input_list - list of files in Ntpl, StdHep of LHEF format,
output_file - output file.
detector_card - Datacard containing resolution variables for the detector
  simulation (optional)
trigger_card - Datacard containing the trigger algorithms (optional)
```

Main things needed:

- input_list : e.g. TEST_small_tt.list — List of input MC files
- output_file : e.g. test.root — Output ROOT filename

Some options:

- detector_card : e.g. data/DetectorCard_CMS.dat — Detector parameters
- trigger_card : e.g. data/TriggerCard_CMS.dat — Trigger definitions

Try it:

```
./Delphes TEST_small_tt.list test.root
```

Trigger Card

S. Ovyn


```
# trigger_name         >> algorithm       #comments
Inclusive electron     >> ELEC1_PT:  '29'
di-electron            >> ELEC1_PT:  '17'   && ELEC2_PT:  '17'
Inclusive Photon       >> GAMMA1_PT: '80'
di-Photon              >> GAMMA1_PT: '40'   && GAMMA2_PT: '25'
Inclusive muon         >> MUON1_PT:  '19'
di-muon                >> MUON1_PT:  '7'    && MUON2_PT:  '7'
Taujet and ETmis       >> TAU1_PT:   '86'   && ETMIS_PT:  '65'
di-Taujets             >> TAU1_PT:   '59'   && TAU2_PT:   '59'
Jet and ETmis          >> JET1_PT:   '180'  && ETMIS_PT:  '123'
Taujet and electron    >> TAU1_PT:   '45'   && ELEC1_PT:  '19'
Taujet and muon        >> TAU1_PT:   '40'   && ELEC1_PT:  '15'
Inclusive b-jet        >> Bjet1_PT:  '237'
Inclusive 1 jet        >> JET1_PT:   '657'
Inclusive 3 jets       >> JET1_PT:   '247'  && JET2_PT:   '247' && JET3_PT: '247'
```

Cuts on p_T in GeV

logical AND operator

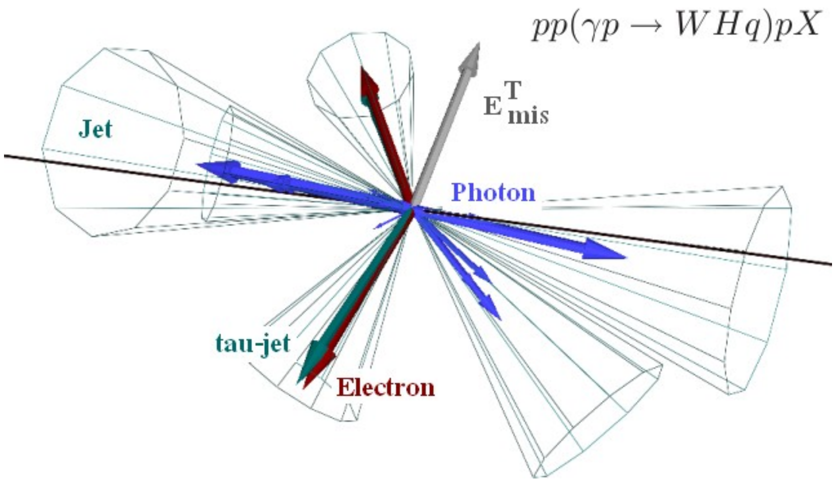
Trigger code	Corresponding object
ELEC_PT	electron
IElec_PT	isolated electron
MUON_PT	muon
IMuon_PT	isolated muon
JET_PT	jet
TAU_PT	τ -jet
ETMIS_PT	missing transverse energy
GAMMA_PT	photon
Bjet_PT	b-jet

Delphes

S. Ovyn

3D Event Display

FROG interfaced to ***Delphes***



FROG: L. Quertenmont, V. Roberfroid, arXiv:0901.2718v1[hep-ex]

To run FROG, some libraries are needed, as it uses the OpenGL free libraries, which are not in Delphes

```
OpenGL:
    xorg-x11-Mesa-libGL-6.8.2-1.EL.33.0.2
    xorg-x11-Mesa-libGLU-6.8.2-
1.EL.33.0.2
GLUT:
    freeglut-2.2.0-14
    freeglut-devel-2.2.0-14
X11-devel:
    xorg-x11-devel-6.8.2-1.EL.33.0.2
CURL:
    libcurl
```

FROG runs on two files : *geom and *vis, which are created if the flag is ON

```
FLAG_frog        1   //1 to run the FROG event display
```

Then compile the source (NOT done automatically when Delphes is compiled)

```
cd Utilities/FROG
make
...
cd ../..
./Utilities/FROG/frog
```

Compile it...

...Run it

Summary and Outlook

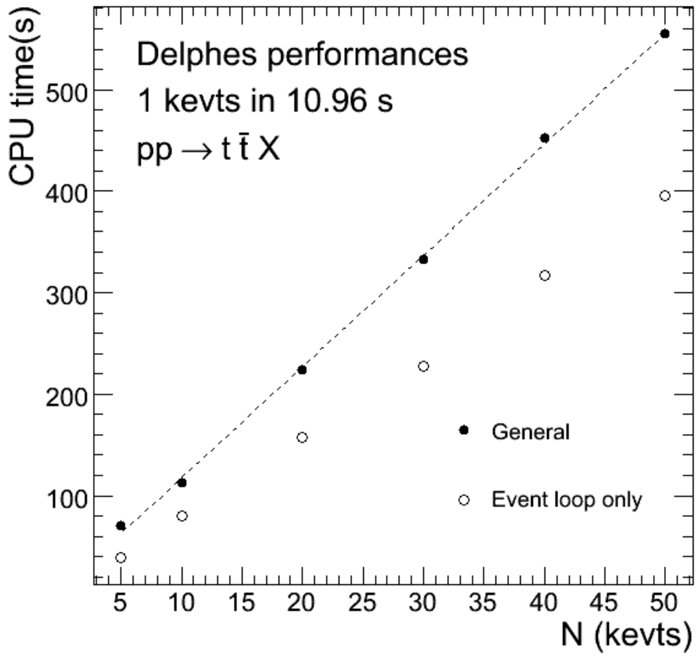
Delphes

S. Ovyn

We present here a new framework for the fast simulation of a generic collider experiment

- Includes Trigger, forward near-beam detectors, 3D Event display

- Several input file types accepted by ***Delphes***
  - StdHEP
  - ROOT files
  - Les Houches Event Format
  - HepMC



- ***Delphes*** stores output information in
  - ASCII file of LHCO type
  - ROOT format

- ***Delphes*** performs a *fast* simulation:
  - 10 000 events, 10.96 s (regular laptop), 240 MB (physics dependent)

Can be used for fast evaluation of observability of new signals in phenomenology, as an illustration tool for tutorial sessions, ...

http://www.fynu.ucl.ac.be/delphes.html

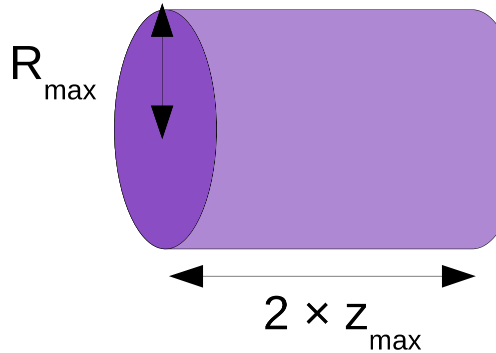
S. Ovyn

Backup slides

Delphes flow

Simulation of the magnetic field

Bx = By = 0 ➡ Exact calculation of the transport of a charged particle



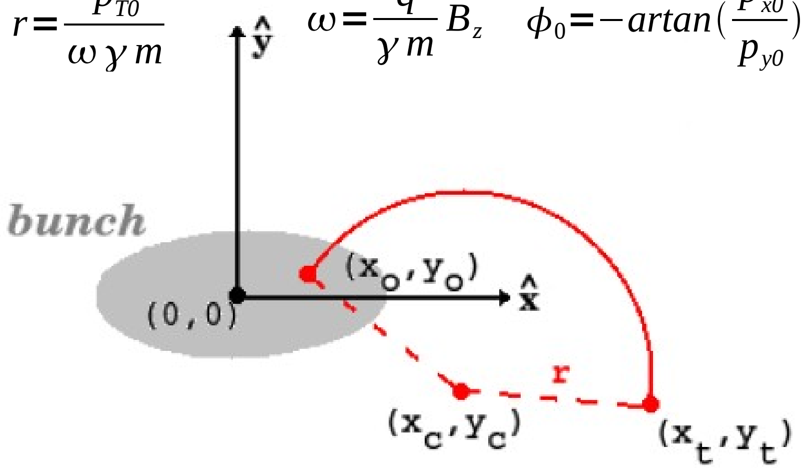
The magnetic field is supposed to be

- homogeneous
- constant inside a cylinder of length $2 \times z_{max}$ and of radius R_{max}.

Based on $$\frac{d\vec{p}}{dt} = q\vec{v}\times\vec{B},\quad \frac{d\vec{x}}{dt} = \vec{v} \text{ and } \vec{p} = \gamma m \vec{v} = \gamma m \frac{d\vec{x}}{dt}.$$

Expression of the position and momentum of the charged particle at any time t.

$$r = \frac{p_{T0}}{\omega\gamma m} \qquad \omega = \frac{q}{\gamma m} B_z \qquad \phi_0 = -\arctan\left(\frac{p_{x0}}{p_{y0}}\right)$$



$$\begin{cases} x(t) = x_c + r\cos(\omega t + \phi_0) \\ y(t) = y_c + r\sin(\omega t + \phi_0) \\ z(t) = z_0 + \dfrac{p_{z0}}{\gamma m} t \end{cases}$$

$$\begin{cases} R(t) = \sqrt{R_c^2 + r^2 + 2 r R_c \cos(\Phi_c - (\phi_0 + \omega t))} \\ \Phi(t) = \arctan\left(\dfrac{y(t)}{x(t)}\right) \\ \Theta(t) = \arctan\left(\dfrac{R(t)}{z(t)}\right) \end{cases}$$

Delphes flow

S. Ovyn

To make the code faster, the time of flight needed to exit the cylinder is computed

1°) t_z : time needed to reach the end of the tracker longitudinally

$$t_z = \frac{\gamma m}{p_{z0}} \left(-z_0 + z_{max} \times sign(p_{z0}) \right)$$

2°) t_z :time to exit the volume by the side once R(t) = R_{max}

$$t_T = \frac{1}{\omega} \left[\Phi_c - \phi_0 + \arctan\left(\frac{R^2_{max} - (R^2_c + r^2)}{2rR_c} \right) \right]$$

$$t_{max} = min(t_T, t_z) \quad \begin{cases} t_z \; such\, that \, |z(t_z)| = z_{max} \\ t_T \; such\, that \; R(t_T) = R_{max} \end{cases}$$

Bx ≠ 0 By ≠ 0 tracker region → iterative method step by step until the particle exits the

Method slower than for a pure solenoidal B field

Disclamer: magnetic field of muon chambers such as for ATLAS not simulated with ***Delphes***

Jets algorithms

S. Ovyn

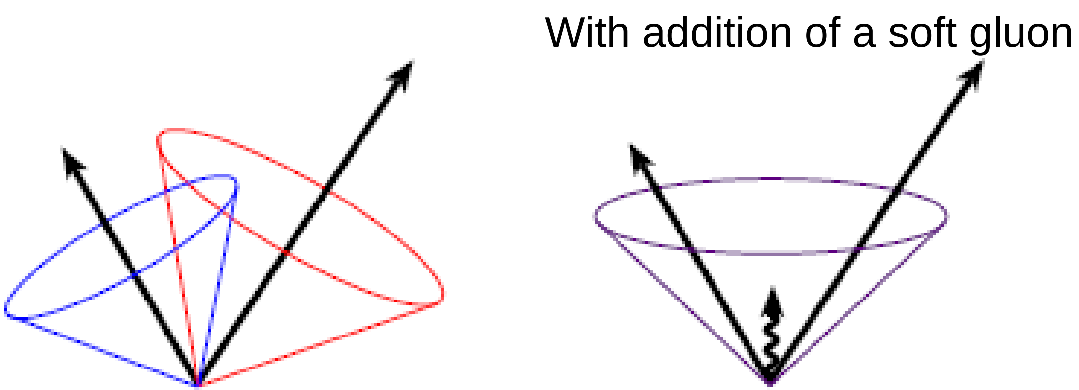
Cone algorithms

1°) CDF jet algorithm - cone (also named « *JetClu cone jet algorithm* »)

- Associates together towers lying within a circle in the (η, φ) space.
- Used by the CDF experiment in Run II.
- Towers with a E_T higher than a given threshold (default: $E_T > 1$ GeV) used as seeds for the jet candidates.

The existing FastJet code has been modified to allow easy modification of the tower pattern in η, φ space.

JetClu is not <u>infrared safe</u>



Identified as 2 jets   Identified as 1 jets

➡ More performant algorithms are also available in Delphes

Jets algorithms

S. Ovyn

Cone algorithms

1°) CDF jet algorithm - cone (also named « *Jetclu cone jet algorithm* »)

- Associates together towers lying within a circle in the (η, φ) space.
- Used by the CDF experiment in Run II.
- Towers with a E_T higher than a given threshold (default: $E_T > 1$ GeV) used as seeds for the jet candidates.

The existing FastJet code has been modified to allow easy modification of the tower pattern in η, φ space.

2°) CDF Midpoint algorithm

- Identical jet procedure than the CDF jet cone algorithm
- Algorithm that reduces infrared sensitivity
- Adds ‘midpoints’ (energy barycentres) in the list of cone seeds.

3°) SIS Cone jets : NO seed!

- Simultaneously insensitive to additional soft particles and collinear splittings,
- Fast enough to be used in experimental analysis.

Jets algorithms

Recombination algorithms

- Infrared and colinear safe

- Merge successive calorimeter tower pairs

- Similar jet running except for the definition of distances *d*

→ distance d_{ij} between each pair of towers (i, j)

variable d_{iB} (beam distance) depending on the p_T of the tower i.

Algorithm:

- Browses the calotower list
- Starts by finding the minimum value d_{min} of all d_{ij} and d_{iB}.
- $d_{min} = d_{ij}$ towers i and j merged into a single tower with $p^\mu = p^\mu(i) + p^\mu(j)$
- $d_{min} = d_{iB}$ the tower is declared as a final jet

4°) k_t jets: $d_{ij} = min(k_{ti}^2, k_{tj}^2)\,\Delta R_{ij}^2 / R^2$ and $d_{iB} = k_{ti}^2$

5°) Cambridge / Aachen jets: $d_{ij} = \Delta R_{ij}^2 / R^2$ and $d_{iB} = 1$

6°) Anti k_t jets: $d_{ij} = min(1/k_{ti}^2, 1/k_{tj}^2)\,\Delta R_{ij}^2 / R^2$ and $d_{iB} = 1/k_{ti}^2$