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

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

severine.ovyn@uclouvain.be, xavier.rouby@cern.ch

Abstract

Knowing whether theoretical predictions are visible and measurable in a high energy experiment is always delicate, due to the complexity of the related detectors, data acquisition chain and software. We introduce here a new framework, DELPHES, for fast simulation of a general purpose experiment. The simulation includes a tracking system, embedded into a magnetic field, calorimetry 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) and outputs observable analysis data objects, like missing transverse energy and collections of electrons or jets. The simulation of detector response takes into account the detector resolution, and usual reconstruction algorithms for complex objects, like FASTJET. A simplified preselection can also be applied on processed data 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. An overview of DELPHES is given as well as a few use-cases for illustration.

Keywords: DELPHES, fast simulation, LHC, smearing, trigger, FASTJET, HECTOR, FROG

[a] Now in Physikalisches Institut, Albert-Ludwigs-Universität Freiburg

1 Introduction

Experiments at high energy colliders are very complex systems, in several ways. First, in terms of the various detector subsystems, including tracking, central calorimetry, forward calorimetry, and muon chambers. These detectors differ with their principles, technologies, geometries and sensitivities. Then, due to the requirement of a highly effective online selection (i.e. a *trigger*), subdivided into several levels for an optimal reduction factor, but based only on partially processed data. Finally, in terms of the experiment software, with different data formats (like *raw* or *reconstructed* data), many reconstruction algorithms and particle identification schemes.

This complexity is handled by large collaborations of thousands of people, which restrict the availability of the data, software and documentation to their members. Real data analyses require a full detector simulation, including the various detector inefficiencies, the dead material, the imperfections and the geometrical details. Moreover, detector calibration and alignment are crucial. Such simulation is very complicated, technical and slow. On the other hand, phenomenological studies, looking for the observability of given signals, may require only fast but realistic estimates of the observables.

A new framework, called DELPHES [1], is introduced here, for the fast simulation of a general purpose collider experiment. Using the framework, observables can be estimated for specific signal and background channels, as well as their production and measurement rates, under a set of assumptions. Starting from the output of event generators, the simulation of the detector response takes into account the subdetector resolutions, by smearing the kinematical properties of the visible final particles. Tracks of charged particles and calorimetric towers (or *calotowers* are then created.

DELPHES includes the most crucial experimental features, like (1) the geometry of both central or forward detectors; (2) lepton isolation; (3) reconstruction of photons, leptons, jets, b-jets, τ-jets and missing transverse energy; (4) trigger emulation and (5) an event display (Fig. 1).

Although this kind of approach yields much realistic results than a simple "parton-level" analysis, a fast simulation comes with some limitations. Detector geometry is idealised, being uniform, symmetric around the beam axis, and having no cracks nor dead material. Secondary interactions, multiple scatterings, photon conversion and bremsstrahlung are also neglected.

Three formats of input files can currently be used as input in DELPHES[1]. In order to process events from many different generators, the standard Monte Carlo event structure STDHEP can be used as an input. Besides, DELPHES can also provide detector response for events read in "Les Houches Event Format" (LHEF) and ROOT files obtained using the `h2root` utility from the ROOT framework [2].

DELPHES uses the `ExRootAnalysis` utility [3] to create output data in a `*.root` file format. This output contains a copy of the generator level data (GEN tree), the analysis data objects after reconstruction (Analysis tree), and possibly the results of the trigger emulation (Trigger tree). The program is driven by input cards. The detector card (`data/DataCardDet.dat`) allows a large spectrum of running conditions by modifying basic detector parameters, including calorimeter and tracking coverage and resolution, thresholds or jet algorithm parameters. The trigger card (`data/trigger.dat`) lists the user algorithms for the simplified online preselection.

2 Detector simulation

The overall layout of the general purpose detector simulated by DELPHES is shown in Fig. 2. A

[1] `[code]` See the `HEPEVTConverter`, `LHEFConverter` and `STDHEPConverter` classes.

central tracking system (TRACKER) is surrounded by an electromagnetic and a hadron calorimeters (ECAL and HCAL, resp.). Two forward calorimeters (FCAL) ensure a larger geometric coverage for the measurement of the missing transverse energy. Finally, a muon system (MUON) encloses the central detector volume The fast simulation of the detector response takes into account geometrical acceptance of sub-detectors and their finite resolution, as defined in the smearing data card[2]. If no such file is provided, predifined values are used. The coverage of the various subsystems used in the default configuration are summarised in table 1.

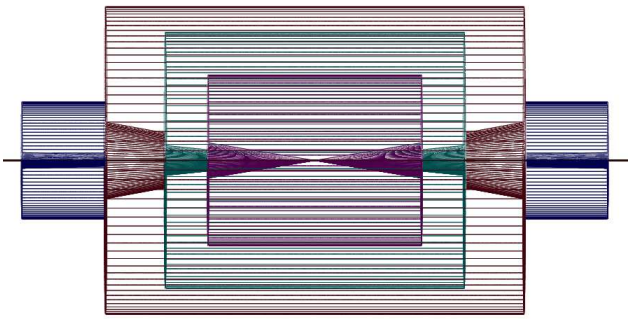

Figure 2: Profile of layout of the generic detector geometry assumed in DELPHES. The innermost layer, close to the interaction point, is a central tracking system (pink). It is surrounded by a central calorimeter volume (green) with both electromagnetic and hadronic sections. The outer layer of the central system (red) consist of a muon system. In addition, two end-cap calorimeters (blue) extend the pseudorapidity coverage of the central detector. The detector parameters are defined in the user-configuration card. The extension of the various subdetectors, as defined in Tab. 1, are clearly visible. The detector is assumed to be strictly symmetric around the beam axis (black line). Additional forward detectors are not depicted.

[2] `[code]` See the `RESOLution` class.

Magnetic field

In addition to the subdetectors, the effects of a dipolar magnetic field is simulated for the charged particles[3]. This simply modifies the corresponding particle direction before it enters the calorimeters.

2.1 Tracks reconstruction

Every stable charged particle with a transverse momentum above some threshold and lying inside the fiducial volume of the tracker provides a track. By default, a track is assumed to be reconstructed with 90% probability[4] if its transverse momentum p_T is higher than 0.9 GeV and if its pseudorapidity $|\eta| \leq 2.5$.

2.2 Simulation of calorimeters

The energy of each particle considered as stable in the generator particle list is smeared, with a Gaussian distribution depending on the calorimeter resolution. This resolution varies with the subcalorimeter (ECAL, HCAL, FCAL) measuring the particle. The response of each sub-calorimeter is parametrised as a function of the energy:

$$\frac{\sigma}{E} = \frac{S}{\sqrt{E}} \oplus \frac{N}{E} \oplus C, \qquad (1)$$

where S, N and C are the *stochastic*, *noise* and *constant* terms, respectively.

The particle four-momentum p^μ are smeared with a parametrisation directly derived from the detector techinal designs[5]. In the default parametrisation, the calorimeter is assumed to cover the pseudorapidity range $|\eta| < 3$ and consists in an electromagnetic and an hadronic

[3] `[code]` See the `TrackPropagation` class.

[4] `[code]` The reconstruction efficiency is defined in the smearing datacard by the `TRACKING_EFF` term.

[5] `[code]` The response of the detector is applied to the electromagnetic and the hadronic particles through the `SmearElectron` and `SmearHadron` functions.

Table 1: Default extension in pseudorapidity η of the different subdetectors. The corresponding parameter name, in the smearing card, is given.

TRACKER	`CEN_max_tracker`	$0.0 \leq \lvert\eta\rvert \leq 2.5$
ECAL, HCAL	`CEN_max_calo_cen`	$0.0 \leq \lvert\eta\rvert \leq 3.0$
FCAL	`CEN_max_calo_fwd`	$3.0 \leq \lvert\eta\rvert \leq 5.0$
MUON	`CEN_max_mu`	$0.0 \leq \lvert\eta\rvert \leq 2.4$

part. Coverage between pseudorapidities of 3.0 and 5.0 is provided by forward calorimeters, with different response to electromagnetic objects ($e^{\pm}, \gamma$) or hadrons. Muons and neutrinos are assumed no to interact with the calorimeters[6]. The default values of the stochastic, noisy and constant terms are given in Table 2.

The energy of electrons and photons found in the particle list are smeared using the ECAL resolution terms. Charged and neutral final state hadrons interact with the ECAL, HCAL and FCAL. Some long-living particles, such as the K_s^0, possessing lifetime $c\tau$ smaller than 10 mm are considered as stable particles although they decay before the calorimeters. The energy smearing of such particles is 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}). Defining F as the fraction of the energy leading to a HCAL deposit, the two energy values are given by

$$\begin{cases} E_{\text{HCAL}} = E \times F \\ E_{\text{ECAL}} = E \times (1 - F) \end{cases} \quad (2)$$

where $0 \leq F \leq 1$. The electromagnetic part is handled as the electrons. The resulting final energy given after the application of the smearing is then $E = E_{\text{HCAL}} + E_{\text{ECAL}}$. For K_S^0 and Λ hadrons, the energy fraction is F is assumed to be worth 0.7.

Table 2: Default values for the resolution of the central and forward calorimeters. Resolution is parametrised by the *stochastic* (S), *noise* (N) and *constant* (C) terms (Eq. 1). The corresponding parameter name, in the smearing card, is given.

Resolution Term	Card flag	Value
ECAL		
S	`ELG_Scen`	0.05
N	`ELG_Ncen`	0.25
C	`ELG_Ccen`	0.0055
FCAL, electromagnetic part		
S	`ELG_Sfwd`	2.084
N	`ELG_Nfwd`	0
C	`ELG_Cfwd`	0.107
HCAL		
S	`HAD_Shcal`	1.5
N	`HAD_Nhcal`	0
C	`HAD_Chcal`	0.05
FCAL, hadronic part		
S	`HAD_Shf`	2.7
N	`HAD_Nhf`	0.
C	`HAD_Chf`	0.13

[6] In the current DELPHES version, particles other than electrons ($e^{\pm}$), photons (γ), muons ($\mu^{\pm}$) and neutrinos (ν_e, ν_μ and ν_τ) are simulated as hadrons for their interactions with the calorimeters. The simulation of stable particles beyond the Standard Model should subsequently be handled with care.

2.3 Calorimetric towers

The smallest unit for geometrical sampling of the calorimeters is a *tower*; it segments the (η, ϕ) plane for the energy measurement. All undecayed particles, except muons and neutrinos produce a calorimetric tower, either in ECAL, in HCAL or FCAL. As the detector is assumed to be symmetric in ϕ and with respect to the $\eta = 0$ plane, the smearing card stores the number of calorimetric towers with $\phi = 0$ and $\eta > 0$ (default: 40 towers). For a given η, the size of the ϕ segmentation is also specified. Fig. 3 illustrates the default segmentation of the (η, ϕ) plane.

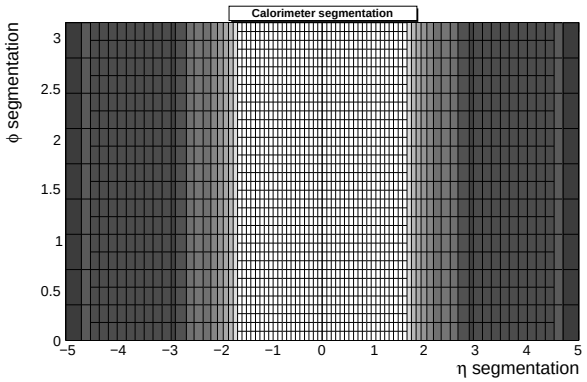

Figure 3: Default segmentation of the calorimeters in the (η, ϕ) plane. Only the central detectors (ECAL, HCAL and FCAL) are considered.

The calorimetric towers directly enter in the calculation of the missing transverse energy (MET), and as input for the jet reconstruction algorithms. No longitudinal segmentation is available in the simulated calorimeters. No sharing between neighbouring towers is implemented when particles enter a tower very close to its geometrical edge.

2.4 Very forward detectors simulation

Most of the recent experiments in beam colliders have additional instrumentation along the beamline. These extend the η coverage to higher values, for the detection of very forward final-state particles. Zero Degree Calorimeters (ZDC) are located at zero angle, i.e. are aligned with the beamline axis at the interaction point, and placed at the distance where the paths of incoming and outgoing beams separate (Fig. 4). These allow the measurement of stable neutral particles (γ and n) coming from the interaction point, with large pseudorapirities (e.g. $|\eta_{n,\gamma}| > 8.3$ in CMS). Forward taggers (called here RP220 and FP420 as at the LHC) are meant for the measurement of particles following very closely the beam path. To be able to reach these detectors, such particles must have a charge identical to the beam particles, and a momentum very close to the nominal value for the beam. These taggers are near-beam detectors located a few millimeters from the true beam trajectory and this distance defines their acceptance (Table 3).

While neutral particles propagate along a straight line to the ZDC, a dedicated simulation of the transport of charged particles is needed for RP220 and FP420. This fast simulation uses the HECTOR software [4], which includes the chromaticity effects and the geometrical aperture of the beamline elements.

Some subdetectors have the ability to measure the time of flight of the particle. This corresponds to the delay after which the particle is observed in the detector, after the bunch crossing. The time of flight measurement of ZDC and FP420 detector is implemented here. For the ZDC, the formula is simply

$$t = t_0 + \frac{1}{v} \times \left(\frac{s - z}{\cos\theta}\right), \tag{3}$$

where t is the time of flight, t_0 is the true time coordinate of the vertex from which the particle originates, v the particle velocity, s is the ZDC distance to the interaction point, z is the longitudinal coordinate of the vertex from which the particle comes from, θ is the particle emission angle. This assumes that the neutral particle observed in the ZDC is highly relativistic, i.e. travelling at the

Table 3: Default parameters for the forward detectors: distance from the interaction point and detector acceptance. The LHC beamline is assumed around the fifth interaction point. For the ZDC, the acceptance depends only on the pseudorapidity η of the particle, which should be neutral and stable. The tagger acceptance is fully determined by the distance in the transverse plane of the detector to the real beam position [4]. It is expressed in terms of the particle energy.

Detector	Distance	Acceptance	
ZDC	140 m	$\lvert\eta\rvert > 8.3$	for n and γ
RP220	220 m	$E \in [6100; 6880]$ (GeV)	at 2 mm
FP420	420 m	$E \in [6880; 6980]$ (GeV)	at 4 mm

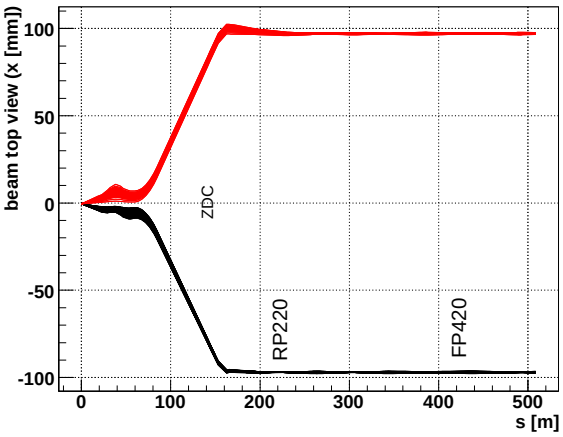

Figure 4: Default location of the very forward detectors, including ZDC, RP220 and FP420 in the LHC beamline. Incoming (red) and outgoing (black) beams on one side of the interaction point ($s = 0$ m). The Zero Degree Calorimeter is located in perfect alignment with the beamline axis at the interaction point, at 140 m, where the beam paths are separated. The forward taggers are near-beam detectors located at 220 m and 420 m.

speed of light c. We also assume that $\cos\theta = 1$, i.e. $\theta \approx 0$ or equivalently η is large. As an example, $\eta = 5$ leads to $\theta = 0.013$ and $1 - \cos\theta < 10^{-4}$. The formula then reduces to

$$t = \frac{1}{c} \times (s - z) \tag{4}$$

Only neutrons and photons are currently assumed to be able to reach the ZDC. All other particles are neglected in the ZDC. To fix the ideas, if the ZDC is located at $s = 140$ m, neglecting z and θ, and assuming that $v = c$, one gets $t = 0.47$ μs.

3 High-level object reconstruction

Analysis object data contain the final collections of particles ($e^{\pm}$, $\mu^{\pm}$, γ) or objects (light jets, b-jets, τ-jets, E_T^{miss}) and are stored[7] in the output file created by DELPHES. In addition, some detector data are added: tracks, calorometric towers and hits in ZDC, RP220 and FP420. While electrons, muons and photons are easily identified, some other objects are more difficult to measure, like jets or missing energy due to invisible particles.

For most of these objects, their four-momentum p^{μ} and related quantities are directly accessible in DELPHES output (E, $\vec{p}$, p_T, η and ϕ). Additional properties are available for specific objects (like the charge and the isolation status for $e^{\pm}$ and $\mu^{\pm}$,

[7] `[code]` All these processed data are located under the `Analysis` tree.

the result of application of b-tag for jets and time-of-flight for some detector hits).

3.1 Photon and charged lepton reconstruction

From here onwards, *electrons* refer to both positrons (e^+) and electrons (e^-), and *charged leptons* refer to electrons and muons ($\mu^\pm$), leaving out the $\tau^\pm$ leptons as they decay before being detected.

Electrons and photons

Photon and electron ($e^\pm$) candidates are reconstructed if they fall into the acceptance of the tracking system and have a transverse momentum above a threshold (default $p_T > 10$ GeV). A calorimetric tower will be seen in the detector, an electrons leave in addition a track. Consequently, electrons and photons creates as usual a candidate in the jet collection.

Muons

Generator level muons entering the detector acceptance are considered as candidates for the analysis level. The acceptance is defined in terms of a transverse momentum threshold to overpass (default : $p_T > 10$ GeV) and of the pseudorapidity coverage of the muon system of the detector (default: $-2.4 \leq \eta \leq 2.4$). The application of the detector resolution on the muon momentum depends on a Gaussian smearing of the p_T variable[8]. Neither η nor ϕ variables are modified beyond the calorimeters: no additional magnetic field is applied. In addition, multiple scattering is also neglected. This implies that low energy muons have in DELPHES a better resolution than in a real detector.

[8] `[code]` See the `SmearMuon` method.

Charged lepton isolation

To improve the quality of the contents of the charged lepton collections, additional criteria can be applied to impose some isolation. This requires that electron or muon candidates are isolated in the detector from any other particle, within a small cone. In DELPHES, charged lepton isolation demands that there is no other charged particle with $p_T > 2$ GeV within a cone of $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2} < 0.5$ around the lepton. The result (i.e. *isolated* or *not*) is added to the charged lepton measured properties[9].

3.2 Jet reconstruction

A realistic analysis requires a correct treatment of final state particles which hadronise. Therefore, the most widely currently used jet algorithms have been integrated into the DELPHES framework using the FASTJET tools [5]. Six different jet reconstruction schemes are available[10]. The first three belong to the cone algorithm class while the last three are using a sequential recombinaison scheme. For all of them, the towers are used as input of the jet clustering. Jet algorithms also differ with their sensitivity to soft particles or collinear splittings, and with their computing speed performance.

Cone algorithms

1. *CDF Jet Clusters*: Algorithm forming jets by associating together towers lying within a circle (default radius $\Delta R = 0.7$) in the (η, ϕ) space. The so-called JETCLU cone jet algorithm that was used by CDF in Run II is used. All towers with a transverse energy E_T higher than a given threshold (default:

[9] `[code]` See the `IsolFlag` output of the `Electron` or `Muon` collections in the `Analysis` tree.

[10] `[code]` The choice is done by allocating the `JET_jetalgo` input parameter in the smearing card.

$E_T > 1$ GeV) are used to seed the jet candidates. The existing FastJet code as been modified to allow easy modification or the tower pattern in η, ϕ space. In the following versions of Delphes, a new dedicated plug-in will be created on this purpose[11].

2. *CDF MidPoint*: Algorithm developped for the CDF Run II to reduce infrared and collinear sensitivity compared to purely seed-based cone by adding 'midpoints' (energy barycenters) in the list of cone seeds.

3. *SISCone*: Seedless Infrared Safe Cone [6]: Cone algorithm simultaneously insensitive to additional soft particles and collinear splittings, and fast enough to be used in experimental analysis.

Recombination algorithms

The three following jet algorithms are safe for soft radiations (*infrared*) and collinear splittings. They rely on recombination schemes where neighbouring calotower pairs are successively merged. The definitions of the jet algorithms are similar except for the definition of the *distances* d used during the merging procedure. Two such variables are defined: the distance d_{ij} between each pair of towers (i, j), and a variable d_{iB} (*beam distance*) depending on the transverse momentum of the tower i.

The jet reconstruction algorithm browses the calotower list. It starts by finding the minimum value $d_{\min}$ of all the distances d_{ij} and d_{iB}. If $d_{\min}$ is a d_{ij}, the towers i and j are merged into a single tower with a four-momentum $p^\mu = p^\mu(i) + p^\mu(j)$ (*E-scheme recombination*). If $d_{\min}$ is a d_{iB}, the tower is declared as a final jet and is removed from the input list. This procedure is repeated until no towers are left in the input list. Further information on these jet algorithms is given here below, using k_{ti}, y_i and ϕ_i as the transverse momentum, rapidity and azimuth of calotower i and $\Delta R_{ij} = \sqrt{(y_i - y_j)^2 + (\phi_i - \phi_j)^2}$ as the jet-radius parameter:

4. *Longitudinally invariant k_t jet* [7]:

$$\begin{aligned} d_{ij} &= \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2 \\ d_{iB} &= k_{ti}^2 \end{aligned} \tag{5}$$

5. *Cambridge/Aachen jet* [8]:

$$\begin{aligned} d_{ij} &= \Delta R_{ij}^2/R^2 \\ d_{iB} &= 1 \end{aligned} \tag{6}$$

6. *Anti k_t jet* [9]: where hard jets are exactly circular

$$\begin{aligned} d_{ij} &= \min(1/k_{ti}^2, 1/k_{tj}^2)\Delta R_{ij}^2/R^2 \\ d_{iB} &= 1/k_{ti}^2 \end{aligned} \tag{7}$$

By default, reconstruction uses a cone algorithm with $\Delta R = 0.7$. Jets are stored if their transverse energy is higher[12] than 20 GeV.

3.3 *b*-tagging

A jet is tagged as b-jets if its direction lies in the acceptance of the tracker and if it is associated to a parent b-quark. A b-tagging efficiency of 40% is assumed if the jet has a parent b quark. For c-jets and light jets (i.e. originating in u,d,s quarks or in gluons), a fake b-tagging efficiency of 10% and 1% respectively is assumed[13] . The (mis)tagging relies on the true particle identity (PID) of the most energetic particle within a cone around the observed (η, ϕ) region, with a radius ΔR of 0.7.

[11] `[code]` `JET_coneradius` and `JET_seed` variables in the smearing card.

[12] `[code]` `PTCUT_jet` variable in the smearing card.

[13] `[code]` Corresponding to the `TAGGING_B`, `MISTAGGING_C` and `MISTAGGING_L` constants, for (respectively) the efficiency of tagging of a b-jet, the efficiency of mistagging a c-jet as a b-jet, and the efficiency of mistagging a light jet (u,d,s,g) as a b-jet.

3.4 τ identification

Jets originating from τ-decays are identified using an identification procedure consistent with the one applied in a full detector simulation [10]. The tagging rely on two properties of the τ lepton. First, 77% of the τ hadronic decays contain only one charged hadron associated to a few neutrals (table 4). Tracks are useful for this criterium. Secondly, the particles arisen from the τ lepton produce narrow jets in the calorimeter (*collimation*).

Table 4: Branching rations for τ^- lepton [11]. $h^\pm$ and h^0 refer to charged and neutral hadrons, respectively. $n \geq 0$ and $m \geq 0$ are integers.

Leptonic decays	
$\tau^- \rightarrow e^- \ \bar{\nu}_e \ \nu_\tau$	17.85%
$\tau^- \rightarrow \mu^- \ \bar{\nu}_\mu \ \nu_\tau$	17.36%
Hadronic decays	
$\tau^- \rightarrow h^- \ n \times h^\pm \ m \times h^0 \ \nu_\tau$	64.79%
$\tau^- \rightarrow h^- \ m \times h^0 \ \nu_\tau$	50.15%
$\tau^- \rightarrow h^- \ h^+ h^- m \times h^0 \ \nu_\tau$	15.18%

Electromagnetic collimation

To use the narrowness of the τ-jet, the *electromagnetic collimation* C_τ^{em} is defined as the 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. To be taken into account, a calorimeter tower should have a transverse energy E_T^{tower} above a given threshold. A large fraction of the jet energy is expected in this small cone. This fraction, or collimation factor, is represented in Fig. 6 for the default values (see table 5).

Tracking isolation

The tracking isolation for the τ identification requires that the number of tracks associated to a particle with a significant transverse momentum is one and only one in a cone of radius R^{tracks}. This cone should be entirely pointing to the tracker to be taken into account. Default values of these parameters are given in table 5.

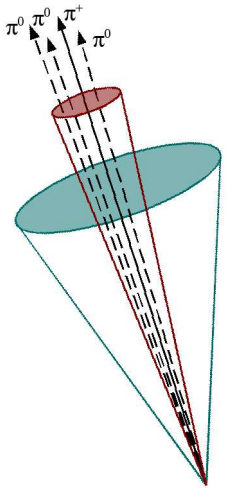

Figure 5: Illustration of the identification of τ-jets. The jet cone is narrow and contains only one track.

Purity

Once both electromagnetic collimation and tracking isolation are applied, a threshold on the p_T of the τ-jet candidate is requested to purify the collection. This procedure selects τ leptons decaying hadronically with a typical efficiency of 60%.

3.5 Missing transverse energy

In an ideal detector, momentum conservation imposes the transverse momentum of the observed final state $\overrightarrow{p_T}^{\text{obs}}$ to be equal to the $\overrightarrow{p_T}$ vector sum of the invisible particles, written $\overrightarrow{p_T}^{\text{miss}}$.

$$\overrightarrow{p_T} = \begin{pmatrix} p_x \\ p_y \end{pmatrix} \text{ and } \begin{cases} p_x^{\text{miss}} = -p_x^{\text{obs}} \\ p_y^{\text{miss}} = -p_y^{\text{obs}} \end{cases} \quad (8)$$

The *true* missing transverse energy, i.e. at generator-level, is calculated as the opposite of

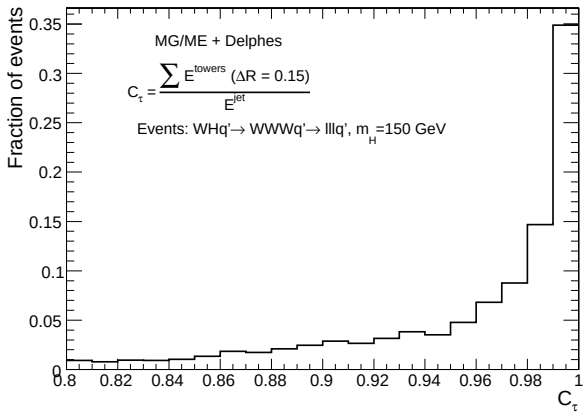

Figure 6: Distribution of the electromagnetic collimation C_τ variable for true τ-jets, normalised to unity. This distribution is shown for associated WH photoproduction [12], where the Higgs boson decays into a W^+W^- pair. Each W boson decays into a $\ell\nu_\ell$ pair, where $\ell = e, \mu, \tau$. Events generated with MadGraph/MadEvent [13]. Final state hadronisation is performed by Pythia [14]. Histogram entries correspond to true τ-jets, matched with generator level data.

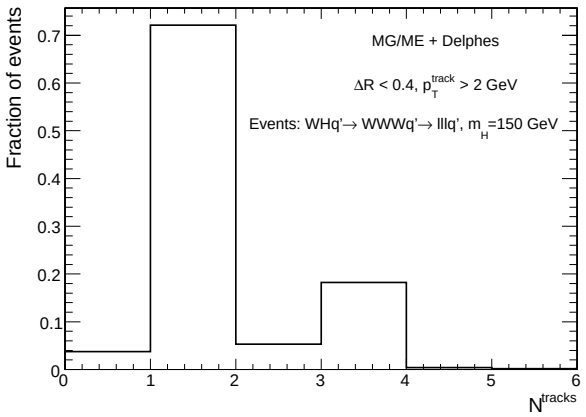

Figure 7: Distribution of the number of tracks N^{tracks} within a small jet cone for true τ-jets, normalised to unity. Photoproduced WH events, where W bosons decay leptonically (e, μ, τ), as in Fig. 6. Histogram entries correspond to true τ-jets, matched with generator level data.

the vector sum of the transverse momenta of all visible particles – or equivalently, to the vector sum of invisible particle transverse momenta. In a real experiment, calorimeters measure energy and not momentum. Any problem affecting the detector (dead channels, misalignment, noisy towers, cracks) worsens directly the measured missing transverse energy $\overrightarrow{E_T}^{\text{miss}}$. In this document, MET is based on the calorimetric towers and only muons and neutrinos are not taken into account for its evaluation:

$$\overrightarrow{E_T}^{\text{miss}} = -\sum_{i}^{\text{towers}} \overrightarrow{E_T}(i) \qquad (9)$$

4 Trigger emulation

New physics in collider experiment are often characterised in phenomenology by low cross-section values, compared to the Standard Model (SM) processes. For instance at the LHC ($\sqrt{s}$ = 14 TeV), the cross-section of inclusive production of $b\bar{b}$ pairs is expected to be 10^7 nb, or inclusive jets at 100 nb ($p_T > 200$ GeV), while Higgs boson cross-section within the SM can be as small as $\ldots \times 10^{-6}$ nb.

High statistics are required for data analyses, consequently imposing high luminosity, i.e. a high collision rate. As only a tiny fraction of the observed events can be stored for subsequent *offline* analyses, a very large data rejection factor should be applied directly as the events are produced. This data selection is supposed to reject only well-known SM events[14]. Dedicated algorithms of this

[14]However, some bandwidth is allocated to random triggers that stores a small fraction of the events without any selection criteria.

Table 5: Default values for parameters used in τ-jet reconstruction algorithm. Electromagnetic collimation requirements involve the inner *small* cone radius R^{em}, the minimum transverse energy for calotowers E_T^{tower} and the collimation factor C_τ. Tracking isolation constrains the number of tracks with a significant transverse momentum p_T^{tracks} in a cone of radius R^{tracks}. Finally, the τ-jet collection is purified by the application of a cut on the p_T of τ-jet candidates.

Parameter	Card flag	Value
Electromagnetic collimation		
R^{em}	`TAU_energy_scone`	0.15
min E_T^{tower}	`JET_M_seed`	1.0 GeV
C_τ	`TAU_energy_frac`	0.95
Tracking isolation		
R^{tracks}	`TAU_track_scone`	0.4
min p_T^{tracks}	`PTAU_track_pt`	2 GeV
τ-jet candidate		
min p_T	`TAUJET_pt`	10 GeV

online selection, or *trigger*, should be fast and very efficient for data rejection, in order to preserve the experiment output bandwidth. They must also be as inclusive as possible to avoid loosing interesting events.

Most of the usual trigger algorithms select events containing objects (i.e. jets, particles, MET) with an energy scale above some threshold. This is often expressed in terms of a cut on the transverse momentum of one or several objects of the measured event. Logical combinations of several conditions are also possible. For instance, a trigger path could select events containing at least one jet and one electron such as $p_T^{\text{jet}} > 100$ GeV and $p_T^e > 50$ GeV.

A trigger emulation is included in DELPHES, using a fully parametrisable *trigger table*[15]. When enabled, this trigger is applied on analysis object data. In a real experiment, the online selection is often divided into several steps (or *levels*). This splits the overall reduction factor into a product of smaller factors, corresponding to the different trigger levels. This is related to the architecture of the experiment data acquisition chain, with limited electronic buffers requiring a quick decision for the first trigger level. First level triggers are then fast and simple but based only on partial data as not all detector front-ends are readable within the decision latency. Later levels are more complex, of finer-but-not-final quality and based on full detector data.

Real triggers are thus intrinsically based on reconstructed data with a worse resolution than final analysis data. On the contrary, same data are used in DELPHES for trigger emulation and for final analyses.

5 Validation

DELPHES performs a fast simulation of a collider experiment. Its quality and validity are assessed by comparing to resolution of the reconstructed data to the CMS detector expectations.

Electrons and muons match by construction to the experiment designs, as the Gaussian smearing of their kinematical properties is defined according to the experiment resolution. Similarly, the b-tagging efficiency (for real b-jets) and misidentification rates (for fake b-jets) are taken from the expected values of the experiment. Unlike these simple objects, jets and missing transverse energy should be carefully cross-checked.

5.1 Jet resolution

The majority of interesting processes at the LHC contain jets in the final state. The jet resolution obtained using DELPHES is therefore a crucial point for its validation. Even if DELPHES contains six algorithms for jet reconstruction, only the jet clustering algorithm (JETCLU) with $R = 0.7$ is used to validate the jet collection.

[15] `[code]` The trigger card is the `data/trigger.dat` file.

This validation employs $pp \to gg$ events produced with MadGraph/MadEvent and hadronised using Pythia [13, 14]. The events were arranged in 14 bins of gluon transverse momentum $\hat{p}_T$. In each $\hat{p}_T$ bin, every jet in Delphes is matched to the closest jet of generator-level particles, using the spatial separation between the two jet axes

$$\Delta R = \sqrt{\left(\eta^{\rm rec} - \eta^{\rm MC}\right)^2 + \left(\phi^{\rm rec} - \phi^{\rm MC}\right)^2} < 0.25. \quad (10)$$

The jets made of generator-level particles, or MC jets, are obtained by applying the same clustering algorithm to all particles considered as stable after hadronisation. Jets produced by Delphes and satisfying the matching criterium are called hereafter *reconstructed jets*.

The ratio of the transverse energies of every reconstructed jet $E_T^{\rm rec}$ and its corresponding MC jet $E_T^{\rm MC}$ is calculated in each $\hat{p}_T$ bin. The $E_T^{\rm rec}/E_T^{\rm MC}$ histogram is fitted with a Gaussian distribution in the interval ± 2 RMS centered around the mean value. The resolution in each $\hat{p}_T$ bin is obtained by the fit mean $\langle x \rangle$ and variance $\sigma^2(x)$:

$$\frac{\sigma\left(\frac{E_T^{rec}}{E_T^{MC}}\right)_{\rm fit}}{\left\langle \frac{E_T^{rec}}{E_T^{MC}} \right\rangle_{\rm fit}} \left(\hat{p}_T(i)\right), \text{ for all } i. \quad (11)$$

The resulting jet resolution as a function of $E_T^{\rm MC}$ is shown in Fig. 8. This distribution is fitted with a function of the following form:

$$\frac{a}{E_T^{\rm MC}} \oplus \frac{b}{\sqrt{E_T^{\rm MC}}} \oplus c, \quad (12)$$

where a, b and c are the fit parameters. It is then compared to the resolution obtained with a recent version of the simulation package of the CMS detector [15]. The resolution curves from Delphes and CMS are in good agreement.

5.2 MET resolution

All major detectors at hadron colliders have been designed to be as much hermetic as possible in order to detect the presence of one or more neutrinos through apparent missing transverse energy. The resolution of the $\overrightarrow{E_T}^{\rm miss}$ variable, as obtained with Delphes, is then crucial.

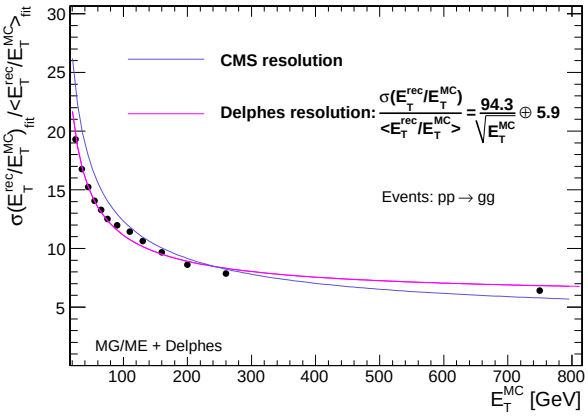

Figure 8: Resolution of the transverse energy of reconstructed jets $E_T^{\rm rec}$ as a function of the transverse energy of the closest jet of generator-level particles $E_T^{\rm MC}$. The maximum separation between the reconstructed and MC jets is $\Delta R = 0.25$. Pink line is the fit result for comparison to the CMS resolution, in blue.

The samples used to study the MET performance are identical to those used for the jet validation. It is worth noting that the contribution to $E_T^{\rm miss}$ from muons is negligible in the studied sample. The[16] input samples are divided in five bins of scalar E_T sums (ΣE_T). This sum, called *total visible transverse energy*, is defined as the scalar sum of transverse energy in all towers. The quality of the MET reconstruction is checked via the resolution on its horizontal component $E_x^{\rm miss}$.

The $E_x^{\rm miss}$ resolution is evaluated in the following way. The distribution of the difference between $E_x^{\rm miss}$ in Delphes and at generator-level is fitted with a Gaussian function in each (ΣE_T) bin. The fit mean gives the MET bias in each bin. The

[16] je n'ai pas tout compris. Ce que j'ai deviné est en rouge.

resulting value is plotted in Fig. 9 as a function of the total visible transverse energy.[17]

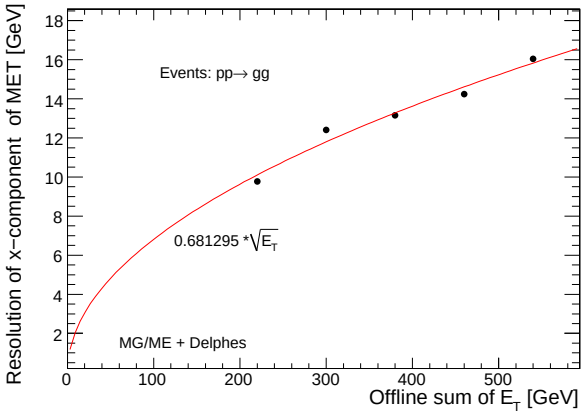

Figure 9: $\sigma(E_x^{\text{miss}})$ as a function on the scalar sum of all towers (ΣE_T) for $pp \to gg$ events.

The resolution σ_x of the horizontal component of MET is observed to behave like

$$\sigma_x = \alpha \ (\Sigma E_T) \quad (\text{GeV}^{1/2}), \tag{13}$$

where the α parameter is depending on the resolution of the calorimeters.

The MET resolution expected for the CMS detector for similar events is $\sigma_x = (0.6 - 0.7)\ (\Sigma E_T)\ \text{GeV}^{1/2}$ with no pile-up[18] [15]. The same quantity obtained by DELPHES is in excellent agreement with the expectations of the general purpose detector, as $\alpha = 0.68$.

5.3 τ-jet efficiency

Due to the complexity of their reconstruction algorithm, τ-jets have also to be checked. Table 6 lists the reconstruction efficiencies for the hadronic τ-jets in the CMS experiment and in DELPHES. Agreement is good enough to validate this reconstruction.

Table 6: Reconstruction efficiencies of τ-jets in decays from Z or H bosons, in DELPHES and in the CMS experiment [16].

CMS		
$Z \to \tau^+\tau^-$	38%	
$H \to \tau^+\tau^-$	36%	$m_H = 150$ GeV
$H \to \tau^+\tau^-$	47%	$m_H = 300$ GeV
DELPHES		
$H \to \tau^+\tau^-$	42%	$m_H = 140$ GeV

6 Visualisation

When performing an event analysis, it can be usefull to convey informations about the detector layout or the event topology in a simple way. With this aim in view, a visualisation tool can be of great interest. Hence, the Fast and Realistic OpenGl Displayer FROG has been interfaced in DELPHES allowing an easy display of the defined detector configuration[19].

For the purpose of publication and talks, the two and three-dimentional representation of the used detector configuration can be used as it clearly show the geometric coverage of the different detector subsystems. An an illustration, the obtained representation of the generic detector geometry assumed in DELPHES is shown in Fig.10 and 11 As pointed before, the detector is assumed to be strictly symmetric around the beam axis. The extention in pseudorapidity of the central tracking system, the central calorimeters are displayed. In addition, the two end-cap calorimeters ad defined in the Datacard extend

[17] Entre nous, ca ressemble plus à un biais (= une différence entre le vrai et le simulé) plus qu'a une résolution! Mais je suppose que c'est la definition que tu as trouvée dans le CMS TDR.

[18] *Pile-up* events are extra simultaneous pp collision occuring at the same bunch crossing.

[19] `[code]` To prepare the visualisation, the `FLAG_frog` parameter should be equal to 1.

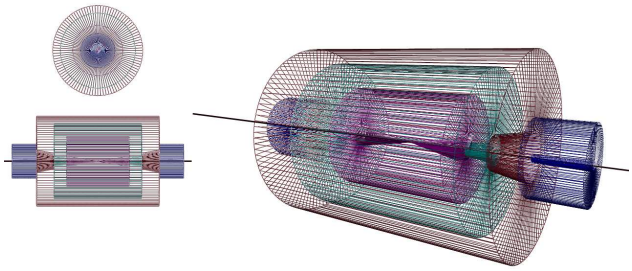

Figure 10: Layout of the generic detector geometry assumed in DELPHES. The innermost layer, close to the interaction point, is a central tracking system (pink). It is surrounded by a central calorimeter volume (green) with both electromagnetic and hadronic sections. The outer layer of the central system (red) consist of a muon system. In addition, two end-cap calorimeters (blue) extend the pseudorapidity coverage of the central detector. The actual detector granularity and extension is defined in the user-configuration card. The detector is assumed to be strictly symmetric around the beam axis (black line). Additional forward detectors are not depicted.

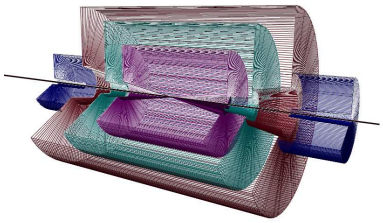

Figure 11: Layout of the generic detector geometry assumed in DELPHES. Open 3D-view of the detector with solid volumes. Same colour codes as for Fig. 10 are applied. Additional forward detectors are not depicted.

the pseudorapidity coverage of the central detector until $|\eta| = 5$. Nevertheless, it should be noticed that only the geometry coverage is represented and that the calorimeter segmentation is not taken into account in the draw of the detector. Morevocer, the radius as well as the length of the different sub-detectors are insignifiant

A more deep understanding of interesting physics processes is obtained using the display of the events. The visibility of each set of objects (e.g. electrons, muons, taus, jets, transverse missing energy) is enhanced by a color coding. Moreover, each object is toggled on by a simple mouse action allowing to access its four-momentum infomration. As an illustration, an associated photoproduction of a W boson and a t quark is shown in Fig. 12. This corresponds to a $pp \to Wt + p + X$ process, where the Wt couple is induced by an incoming photon emitted by one interacting proton. This leading proton survives from the photon emission and subsequently from the pp interaction, and is present in the final state. The experimental signature is a lack of hadronic activity in one forward hemisphere, where the surviving proton escapes. The t quark decays into a W and a b. Both W bosons decay into leptons ($W \to \mu\nu_\mu$ and $W \to \tau\nu_\tau$).

7 Conclusion and perspectives

7.1 version 1

We have described here the major features of the DELPHES framework, introduced for the fast simulation of a collider experiment. It has already been used for several phenomenological studies, in particular in photon interactions at the LHC.

DELPHES takes the output of event generators, in various formats, and yields analysis object data. The simulation applies the resolutions of central and forward detectors by smearing the kinematical properties of final state particles. It yields tracks in a solenoidal magnetic field and calorimetric towers. Realistic reconstruction algorithms are run, including the FASTJET package, to produce collections of $e^\pm$, $\mu^\pm$, jets and τ-jets. b-tag and missing transverse energy are also evaluated. The output is validated by comparing to

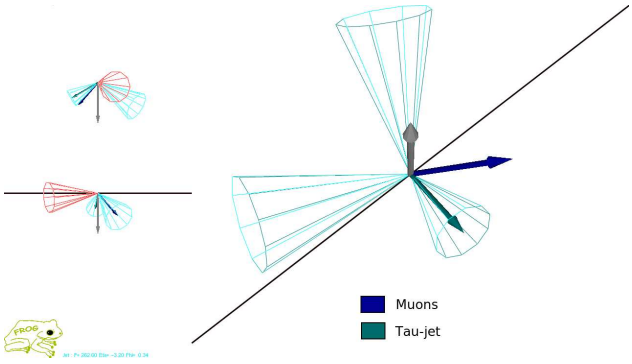

Figure 12: Example of $pp(\gamma p \to Wt)pY$ event. One W boson decays into a μ ν_μ pair and the second one into a τ ν_τ pair. The surviving proton leaves a forward hemisphere with no hadronic activity. The isolated muon is shown as the blue vector. The τ-jet is the cone around the green vector, while the reconstructed missing energy is shown in gray. One jet is visible in one forward region, along the beamline axis, opposite to the direction of the escaping proton.

the CMS expected performances. A trigger stage can be emulated on the output data. At last, event visualisation is possible through the FROG 3D event display.

DELPHES has been developped using the parameters of the CMS experiment but can be easily extended to ATLAS and other non-LHC experiments, as at Tevatron or at the ILC. Further developments include a more flexible design for the subdetector assembly and possibly the implementation of an event mixing module for pile-up event simulation. c'est complet, mais ca ressemble fort a l'abstract et a l'intro.

7.2 version 2

We have described here the major features of the DELPHES framework, introduced for the fast simulation of a collider experiment. This framework is a tool meant for feasibility studies in phenomenology, probing the observability of models in collider experiments. It has already been used for several analyses, in particular in photon interactions at the LHC.

DELPHES takes the output of event generators and yields analysis object data. The simulation includes central and forward detectors to produce realistic observables using standard reconstruction algorithms. Moreover, the framework allows trigger emulation and 3D event visualisation.

DELPHES has been developped using the parameters of the CMS experiment but can be easily extended to ATLAS and other non-LHC experiments, as at Tevatron or at the ILC. Further developments include a more flexible design for the subdetector assembly and possibly the implementation of an event mixing module for pile-up event simulation.

Acknowledgements

The author would like to thank Vincent Lemaitre, Muriel Vander Donckt and David d'Enterria for useful discussions and comments, and Loic Quertenmont for support in interfacing FROG. Part of this work was supported by the Belgian Federal Office for Scientific, Technical and Cultural Affairs through the Interuniversity Attraction Pole P6/11.

References

[1] DELPHES, hepforge:

[2] R. Brun, F. Rademakers, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86.

[3] P. Demin, (2006), unpublished. Now part of MADGRAPH/MADEVENT.

[4] X. Rouby, J. de Favereau, K. Piotrzkowski, JINST 2 P09005 (2007).

[5] M. Cacciari, G. Salam, Phys. Lett. B 641 (2006) 57.

[6] G.P. Salam, G. Soyez, JHEP0705:086 (2007).

[7] S. Catani, Y. L. Dokshitzer, M. H. Seymour and B. R. Webber, Nucl. Phys. B 406 (1993) 187. S. D. Ellis and D. E. Soper, Phys. Rev. D 48 (1993) 3160.

[8] Y.L. Dokshitzer, G.D. Leder, S. Moretti and B.R. Webber, JHEP 9708 (1997) 001. M. Wobisch and T. Wengler, arXiv:hep-ph/9907280.

[9] M. Cacciari, G. P. Salam and G. Soyez, JHEP 0804 (2008) 063.

[10] Tau reconstruction in CMS

[11] C. Amsler et al. (Particle Data Group), PL B667, 1 (2008).

[12] S. Ovyn

[13] J. Alwall, P. Demin, S. de Visscher, R. Frederix, M. Herquet, F. Maltoni, T. Plehn, D.L. Rainwater, T. Stelzer, JHEP 0709:028 (2007).

[14] T. Sjostrand, S. Mrenna and P. Skands, JHEP 05 (2006) 026.

[15] CMS IN 2007/053.

[16] R. Kinnunen, CMS NOTE 1997/002.

[17] Frog,

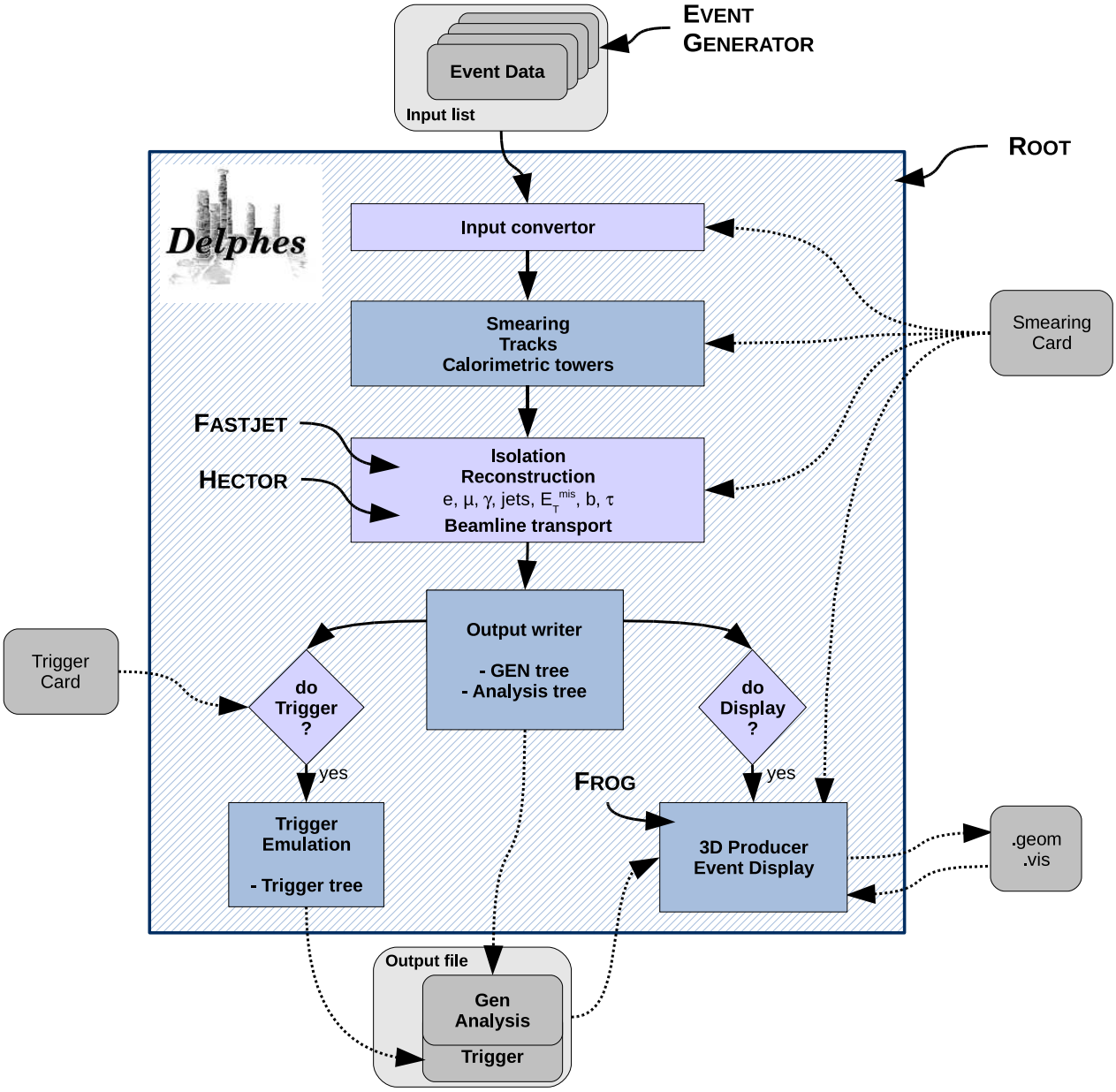

Figure 1: Flow chart describing the principles behind DELPHES. Event files coming from external Monte Carlo generators are read by a convertor stage. The kinematical variables of the final state particles are then smeared according to the subdetector resolutions. Tracks are reconstructed in a simulated dipolar magnetic field and calorimetric towers sample the energy deposits. Based on these, dedicated algorithms are applied for particle identification, isolation and reconstruction. The transport of very forward particle to the near-beam detectors is also simulated. Finally, an output file is written, including generator level and analysis object data. If requested, a fully parametrisable trigger can be emulated. Optionnally, the geometry and visualisation files for the 3D event display can also be produced. All user parameters are set in the *Smearing Card* and the *Trigger Card*.

A User manual

The available code is a tar file which comes with everything you need to run the DELPHES package. Nevertheless in order to visualise the events with the FROG program, you need to install libraries as explained in *href= "http://projects.hepforge.org/frog/"*

A.1 Getting started

In order to run DELPHES on your system, first download is sources and compile it:

```
me@mylap:~$ http://www.fynu.ucl.ac.be/users/s.ovyn/Delphes/files/Delphes_V_*.tar.gz
me@mylap:~$ tar -xvf Delphes_V_*.tar.gz
me@mylap:~$ cd Delphes_V_*.*
me@mylap:~$ ./genMakefile.tcl > Makefile
me@mylap:~$ make
```

A.2 Running Delphes on your events

In this chapter, we will explain how to use DELPHES to perform a fast simulation of a general purpose detector on your event files. The first step to use DELPHES is to create the list of input event files (e.g. `inputlist.list`) file. As an important comment, don't forget that all the files comprised in the list file should have the same type (`*.hep`, `*.lhe` or `*.root`). In the simplest way of running DELPHES, you need this input file and you need to specify the name of the output of DELPHES that will contain the particle-level information (`GEN tree`), the analysis data objects after reconstruction (`Analysis tree`), and the results of the trigger emulation (`Trigger tree`).

```
me@mylaptop:~$ ./Delphes inputlist.list OutputRootFileName.root
```

A.2.1 Setting the run configuration

The program is driven by two datacards (default cards are `data/DataCardDet.dat` and `data/trigger.dat`) which allow a large spectrum of running conditions. Please note that the either you provide those two datacards, either the running will be done using the default parameters defined in the constructor of the class `RESOLution()}`.If you chose a different detector or running configuration you will need to edit the datacards accordingly.

1. **The run card**

   Contains all needed information to run DELPHES

   - The following parameters are available: detector parameters, including calorimeter and tracking coverage and resolution, transverse energy thresholds allowed for reconstructed objects, jet algorithm to use as well as jet parameters.
   - Four flags, `FLAG_bfield`, `FLAG_vfd`, `FLAG_trigger` and `FLAG_frog` should be assigned to decide if the magnetic field propagation, the very forward detectors acceptance, the trigger selection and the preparation for FROG display respectively are running by DELPHES.

If no datacard is provided ny the user, the default one is used that contains the followings smearing and running parameters:

```
# Detector characteristics
CEN_max_tracker    2.5     // Maximum tracker coverage
CEN_max_calo_cen   3.0     // central calorimeter coverage
CEN_max_calo_fwd   5.0     // forward calorimeter pseudorapidity coverage
CEN_max_mu         2.4     // muon chambers pseudorapidity coverage

# Energy resolution for electron/photon
# \sigma/E = C + N/E + S/\sqrt{E}
ELG_Scen          0.05     // S term for central ECAL
ELG_Ncen          0.25     // N term for central ECAL
ELG_Ccen          0.005    // C term for central ECAL
ELG_Cfwd          0.107    // S term for FCAL
ELG_Sfwd          2.084    // C term for FCAL
ELG_Nfwd          0.0      // N term for FCAL

# Energy resolution for hadrons in ecal/hcal/hf
# \sigma/E = C + N/E + S/\sqrt{E}
HAD_Shcal         1.5      // S term for central HCAL
HAD_Nhcal         0.       // N term for central HCAL
HAD_Chcal         0.05     // C term for central HCAL
HAD_Shf           2.7      // S term for FCAL
HAD_Nhf           0.       // N term for FCAL
HAD_Chf           0.13     // C term for FCAL

# Muon smearing
MU_SmearPt        0.01

# Tracking efficiencies
TRACK_ptmin       0.9      // minimal pT
TRACK_eff         100      // efficiency associated to the tracking

# Calorimetric towers
TOWER_number         40
### list of the edges of each tower in eta for eta>0 assuming
###a symmetric detector in eta<0
### the list starts with the lower edge of the most central tower
### the list ends with the higher edged of the most forward tower
### there should be NTOWER+1 values
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

### list of the tower size in phi (in degrees), assuming that all
### towers are similar in phi for a given eta value
### the list starts with the phi-size of the most central tower (eta=0)
### the list ends with the phi-size of the most forward tower
### there should be NTOWER values
#TOWER_dphi 5 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

# Thresholds for reconstructed objetcs
PTCUT_elec       10.0
PTCUT_muon       10.0
PTCUT_jet        20.0
PTCUT_gamma      10.0
PTCUT_taujet     10.0

# General jet variable
JET_coneradius   0.7      // generic jet radius
JET_jetalgo      1        // Jet aglorithm selection
JET_seed         1.0      // minimum seed to start jet reconstruction

# Tagging definition
BTAG_b           40
BTAG_mistag_c    10
BTAG_mistag_l    1

# FLAGS
FLAG_bfield      0        // 1 to run the bfield propagation else 0
FLAG_vfd         1        // 1 to run the very forward detectors else 0
FLAG_trigger     1        // 1 to run the trigger selection else 0
FLAG_frog        1        // 1 to run the FROG event display

# In case BField propagation allowed
TRACK_radius      129     // radius of the BField coverage
TRACK_length      300     // length of the BField coverage
TRACK_bfield_x    0       // X composant of the BField
TRACK_bfield_y    0       // Y composant of the BField
TRACK_bfield_z    3.8     // Z composant of the BField

# In case Very forward detectors allowed
VFD_min_calo_vfd  5.2     // very forward calorimeter (if any) like CASTOR
VFD_max_calo_vfd  6.6
```

```
VFD_min_zdc       8.3
VFD_s_zdc         140     // distance of the ZDC, from the IP, in [m]

RP_220_s          220     // distance of the RP to the IP, in meters
RP_220_x          0.002   // distance of the RP to the beam, in meters
RP_420_s          420     // distance of the RP to the IP, in meters
RP_420_x          0.004   // distance of the RP to the beam, in meters

# In case FROG event display allowed
NEvents_Frog      100
```

2. **The trigger card**

   Contains the definition of all trigger bits. Cuts can be applied on the transverse momentum of electrons, muons, jets, tau-jets, photons and transverse missing energy. The following "code-name" should be used so that DELPHES can correctly translate the input list of trigger bit into selection algorithms:

Trigger flag	*Corresponding object*
`ELEC_PT`	electron
`MUON_PT`	muon
`JET_PT`	jet
`TAUJET_PT`	tau-jet
`ETMIS_PT`	transverse missing energy
`GAMMA_PT`	photon

   Moreover, each line in the trigger datacard is allocated to exactly one trigger bit and start with the name of the correcponding trigger. Logical combinaison of several conditions is also possible. If the trigger bit uses the presence of multiple identical objects, the order of their thresholds is not meaningless: they must be defined in decreasing order. Finally, the different requirements on the objects must be separated by a `&&` flag. The default trigger card can be found in the data repository of DELPHES. An exemple of trigger table consistent with the previous rules is given here:

```
DoubleElec                  >> ELEC_PT: '20' && ELEC_PT: '10'   SingleElec and Single
```

   An example (the default trigger card) can be found in `files/trigger.dat`.

A.2.2 Running the code

Create the above cards (data/mydetector.dat and data/mytrigger.dat). Create a text file containing the list of input files that will be used by DELPHES (with extension *.lhe, *.root or *.hep) To run the code, type the following

```
me@mylaptop:~$ ./Delphes inputlist.list OutputRootFileName.root data/mydetector.dat data/mytr
```

A.3 Getting the Delphes information

A.3.1 Contents of the Delphes root trees

As said upwards, the DELPHES ROOT file is subdivided into three TREES. All the branches available in those TREES together with the reconstructed objects they correspond to are summarised here:

GEN tree		
Particle	generator particles from HEPEVT	`TRootGenParticle`
Analysis tree		
Jet	Jet collection	`TRootJet`
TauJet	Collection of jets tagged as τ-jets	`TRootTauJet`
Electron	Collection of electrons	`TRootElectron`
Muon	Collection of muons	`TRootMuon`
Photon	Collection of photons	`TRootPhoton`
Tracks	Tracker tracks	`TRootTracks`
ETmis	Transverse missing energy information	`TRootETmis`
CaloTower	Calorimetric towers	`TRootCalo`
ZDChits	?????	`TRootZdcHits`
RP220hits	?????	`TRootRomanPotHits`
FP420hits	?????	`TRootRomanPotHits`
Trigger		
TrigResult	Acceptance of different trigger bits	`TRootTrigger`

The third column shows the names of the corresponding classes to be written in a ROOT tree. All classes except the `TRootTrigger`, the `TRootETmis` and the `TRootRomanPotHits` inherit from the class `TRootParticle}`which includes the following member functions for accessing the components:

```
float E;  // particle energy in GeV
float Px; // particle momentum vector (x component) in GeV
float Py; // particle momentum vector (y component) in GeV
float Pz; // particle momentum vector (z component) in GeV

float PT; // particle transverse momentum in GeV
float Eta; // particle pseudorapidity  float Phi; // particle azimuthal angle in rad
```

In addition to their four-momentum and related quantities, additional properties are available for specific objects. Those are summarized in the following table:

```
Particle leave
 int PID;                  // particle HEP ID number
 int Status;               // particle status
 int M1;                   // particle 1st mother
 int M2;                   // particle 2nd mother
 int D1;                   // particle 1st daughter
 int D2;                   // particle 2nd daughter
 float Charge;             // electrical charge
 float T;                  // particle vertex position (t component)
 float X;                  // particle vertex position (x component)
 float Y;                  // particle vertex position (y component)
 float Z;                  // particle vertex position (z component)
 float M;                  // particle mass
Electron and Muon leaves
 int Charge
 bool IsolFlag
Jet leave
 bool Btag
ZDChits leave
float T;                   // time of flight [s]
 int side;                 // -1 or +1
```

A.4 Running an analysis on your Delphes events

To analyze the `Root TTree` ntuple of DELPHES, the simplest way is to use the `Analysis_Ex.cpp` code which is coming in the `Examples` repository of DELPHES. Note that all of this is optional and done to facilitate the analysis, as the output from DELPHES is viewable with the standard TBrowser or ROOT and can be analyzed using the MakeClass facility. To run the `Examples/Analysis_Ex.cpp` code, the two following arguments are required: a text file containing the input DELPHES root files to run, and the name of the output root file. To run the code:

```
./Analysis_Ex input_file.list output_file.root
```

A.4.1 sdflksdjf

The `Examples/Trigger_Only.cpp` code permits to run the trigger selection separately from the general detector simulation on output DELPHES root files. An input DELPHES root file is mandatory as argument. The new tree containing the trigger information will be added in these file. The trigger datacard is also necessary. To run the code:

```
./Trigger_Only input_file.root data/trigger.dat
```

A.5 Running the Frog event display

- If the `FLAG_frog` was switched on, two files were created during the run of DELPHES: `DelphesToFrog.vis` and `DelphesToFrog.geom`. They contain all the needed information to

run frog.

- To display the events and the geometry, you first need to compile FROG. Go to the `Utilities/FROG` and type `make`.
- Go back into the main directory and type `./Utilities/FROG/frog`.