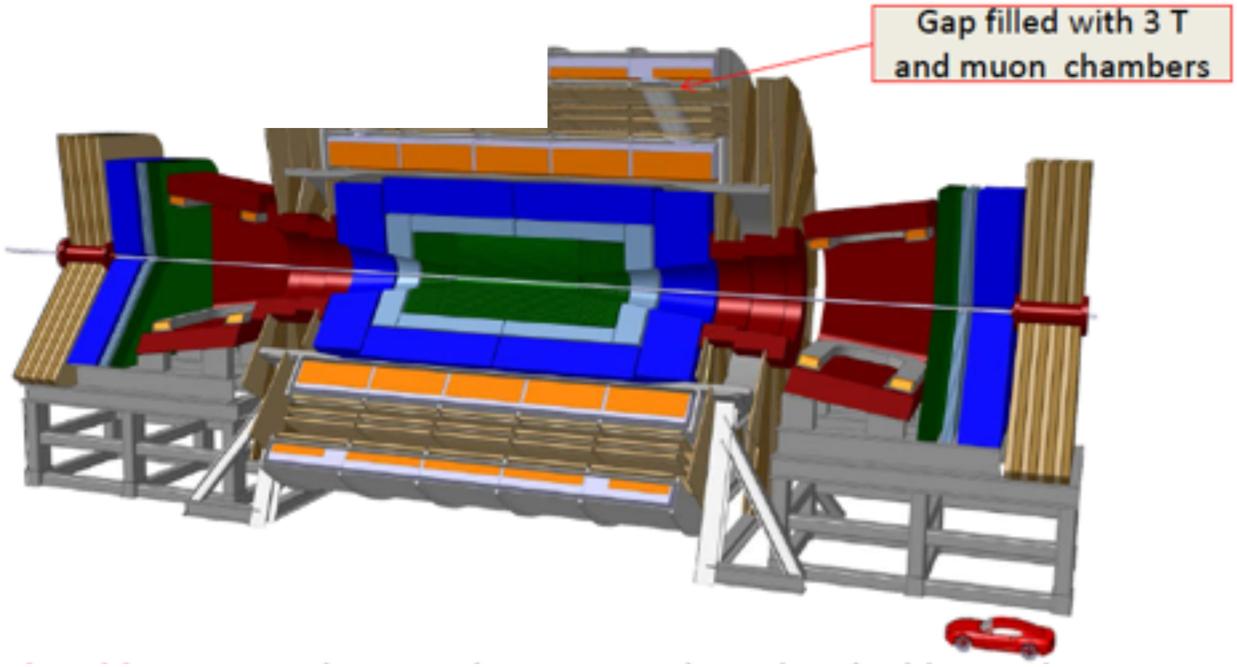
FCC-hh

Towards a DELPHES description for FCC

Heather Gray, Filip Moortgat (CERN)



Introduction

- Which framework to use if we want to a rough simulation of an FCC detector?

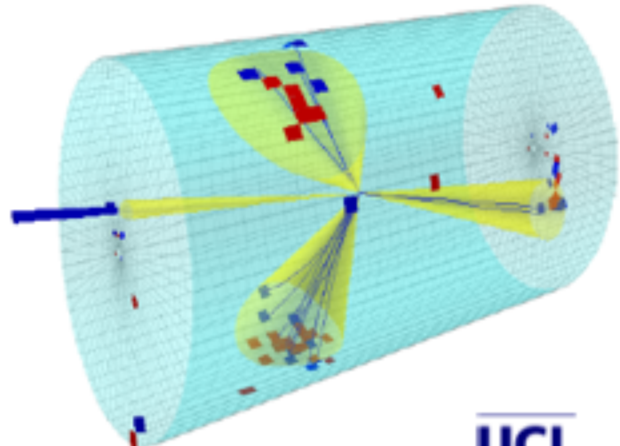
- DELPHES seems the most appropriate short-term simulation package
- FCC software framework will include DELPHES interface, plus other detector simulation options (see talk by Benedikt/Colin later)
  - Long-term goal

- Here we discuss a proposal for a DELPHES implementation of a generic FCC detector

DELPHES

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

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



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

M. Selvaggi, A. Mertens, P. Demin

- Website and manual: https://cp3.irmp.ucl.ac.be/projects/delphes

- Paper: JHEP 02 (2014) 057

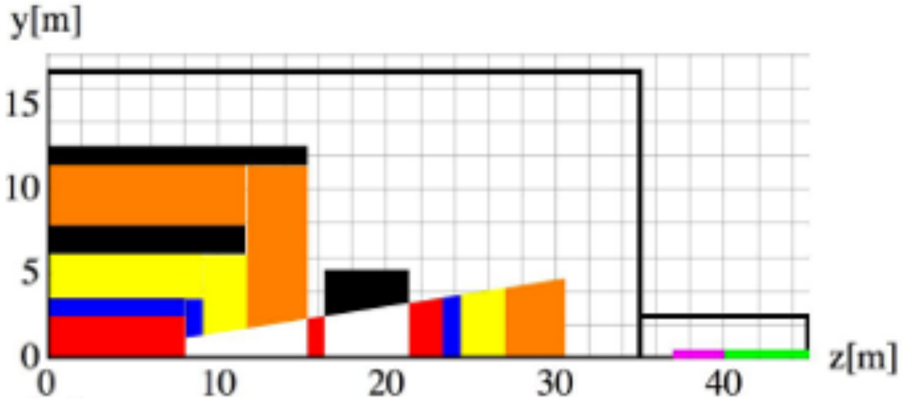
Ingredients

What do we need for a DELPHES description of an FCC detector?

- basic geometry
  - radius of tracking system (before the magnet or calorimetry)
  - length of magnetic field coverage
  - magnetic filed intensity
  - tracking coverage in eta/theta
  - eta/phi granularity of hcal and ecal
- momentum resolution formula for charged tracks
- energy resolution for electrons and photons
- momentum resolution for muons
- impact parameters resolution (optional)
- identification and mis-identification efficiency for particles: muons, electron, pions, kaons, ...
- neutral hadron energy fraction lost in hcal and ecal (sum =1)
- energy resolution formula for jets (optional)
- b-tag efficiency (optional)

Magnet choice

First choice: which magnet configuration?
→ pick Twin Solenoid option as baseline



Twin Solenoid
+ Dipole ?

Delphes card:

radius of the magnetic field coverage, in m
set Radius 6.0

half-length of the magnetic field coverage, in m
set HalfLength 11.5

magnetic field
set Bz 6.0

Note: dipole field not yet implemented in Delphes. Is this urgent?

Twin Solenoid design

"System"	"ATLAS"	"CMS"	"Twin Sol0"	"Twin Sol1"
"DrTracker"	1.23	1.29	2.5	2.5
"DrCoil0"	0.05	0	0	0
"DrEcal"	1	0.48	1.1	1.1
"DrHcal"	1.97	1.23	2.6	2.6
"DrCoil"	0	0.312	1.73	1.73
"DrMuon"	6	4.2	3.57	3.57
"DrCoil2"	0	0	1.1	1.1
"LTracker"	2.8	2.9	8	8
"LEecal"	1	0.98	1.1	1.1
"LEhcal"	2.6	1.8	2.6	2.6
"LEmuon"	17	5.18	3.57	3.57
"LUtracker"	0	0	0	1
"LDipole"	0	0	0	5
"DrDipole"	0	0	0	2
"LDtracker"	0	0	0	2
"LFecal"	0	0	0	1.1
"LFhecal"	0	4	0	2.6
"LFmuon"	0	0	0	3.57
"EtaForward"	10	3	10	2.56
"Lstar"	23	23	40	40
"LTAS"	3	3	3	3
"Lcavern"	23.5	23.5	35	35
"Rcavern"	15	15	17	17
"Rtunnel"	2	2	2.5	2.5
"Rtriplet"	0.5	0.5	0.5	0.5
"xmax"	45	45	45	45
"ymax"	18	18	18	18
"eps"	0.1	0.1	0.1	0.1
"BBarrel"	2	3.8	6	6
"SigBarrel"	20 * 10 ^ (-6)	30 * 10 ^ (-6)	20 * 10 ^ (-6)	20 * 10 ^ (-6)
"NBarrel"	15	15	15	15
"ForwardBFlag"	False	False	False	True
"ToroidFlag"	False	False	False	False
□	□	□	□	□
□	□	□	□	□

Charged hadron efficiency

Reference: http://indico.cern.ch/event/340703/session/80/contribution/167/material/slides/1.pdf

```
set EfficiencyFormula {                                        (pt <= 0.5)   * (0.00) + \
                          (abs(eta) <= 1.5) * (pt > 0.5   && pt <= 1)   * (0.90) + \
                          (abs(eta) <= 1.5) * (pt > 1)                  * (0.99) + \
               (abs(eta) > 1.5 && abs(eta) <= 4) * (pt > 0.5   && pt <= 1)   * (0.85) + \
               (abs(eta) > 1.5 && abs(eta) <= 4) * (pt > 1)                  * (0.98) + \
 (abs(eta) > 4 && abs(eta) <= 6) * (pt > 0.5   && pt <= 1)   * (0.80) + \
               (abs(eta) > 4 && abs(eta) <= 6) * (pt > 1.0)                  * (0.95) + \
               (abs(eta) > 6.0)                                              * (0.00)}}
```

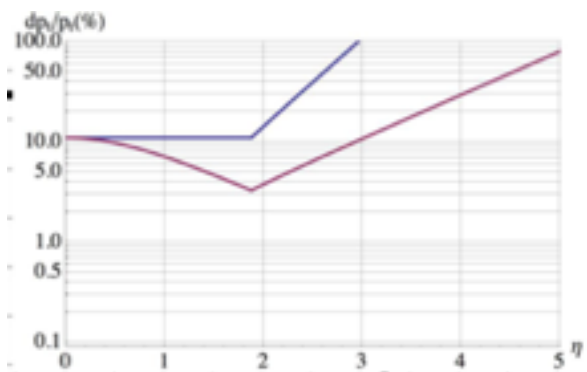
- $|\eta| < 6$
- Minimum p_T cut of 500 MeV?
  - Depends on magnetic field, computing requirements
- Material distribution will drive inefficiency

Charged hadron resolution

```
set ResolutionFormula {                 (abs(eta) <= 1.5) * (pt > 0.5   && pt <= 1.0)   * (0.02) + \
                                 (abs(eta) <= 1.5) * (pt > 1.0   && pt <= 1.0e1) * (0.01) + \
                                 (abs(eta) <= 1.5) * (pt > 1.0e1 && pt <= 2.0e2) * (0.03) + \
                                 (abs(eta) <= 1.5) * (pt > 2.0e2)                * (0.05) + \
                       (abs(eta) > 1.5 && abs(eta) <= 4) * (pt > 0.5   && pt <= 1.0)   * (0.03) + \
                       (abs(eta) > 1.5 && abs(eta) <= 4) * (pt > 1.0   && pt <= 1.0e1) * (0.02) + \
                       (abs(eta) > 1.5 && abs(eta) <= 4) * (pt > 1.0e1 && pt <= 2.0e2) * (0.04) + \
                       (abs(eta) > 1.5 && abs(eta) <= 4) * (pt > 2.0e2)                * (0.05)}
    (abs(eta) > 4 && abs(eta) <= 6) * (pt > 0.5   && pt <= 1.0)   * (0.06) + \
                       (abs(eta) > 4 && abs(eta) <= 6) * (pt > 1.0   && pt <= 1.0e1) * (0.04) + \
                       (abs(eta) > 4 && abs(eta) <= 6) * (pt > 1.0e1 && pt <= 2.0e2) * (0.08) + \
                       (abs(eta) > 4 && abs(eta) <= 6) * (pt > 2.0e2)                * (0.08)}
```

η: 0,1.5,4,6

p_T: 0.5, 1, 10, >20



$$\frac{\sigma(p_T)}{p_T} = \frac{\sigma_x \cdot p_T}{0.3BL^2}\sqrt{\frac{720}{(N+4)}}$$

Needs some update depending on tracker design assumptions

Electrons

```
# tracking efficiency formula for electrons
 set EfficiencyFormula {
(pt <= 10)   * (0.00) + \
(abs(eta) <= 1.5) * (pt > 10   && pt <= 50)   * (0.80) + \
(abs(eta) <= 1.5) * (pt > 50)                 * (0.90) + \
(abs(eta) > 1.5 && abs(eta) <= 4) * (pt > 10   && pt <= 50)  * (0.80) + \
(abs(eta) > 1.5 && abs(eta) <= 4) * (pt > 50) * (0.90) + \
(abs(eta) > 4 && abs(eta) <= 6) * (pt > 10   && pt <= 50)  * (0.70) + \
(abs(eta) > 4 && abs(eta) <= 6) * (pt > 50) * (0.80) + \
(abs(eta) > 6)                                * (0.00)}
```

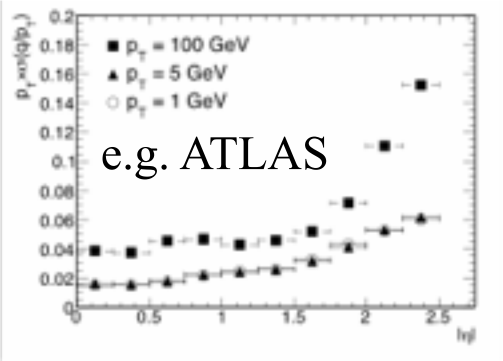
Identification efficiency included

```
# resolution formula for electrons
 set ResolutionFormula {
(abs(eta) <= 4.0) * (energy > 0.1   && energy <= 2.0e1) * (energy*0.007) + \
(abs(eta) <= 4.0) * (energy > 2.0e1) * sqrt(energy^2*0.005^2 + energy*0.02^2)
+ \
(abs(eta) > 4.0 && abs(eta) <= 6.0)  * sqrt(energy^2*0.05^2 + energy*1.00^2)}
```

What resolution to assume ?

Muons

```
set EfficiencyFormula {
(pt <= 5)   * (0.00) + \
(abs(eta) <= 6) * (pt > 5 )   * (0.99) + \
(abs(eta) > 6) * (0.00)}
```




```
set ResolutionFormula {
(abs(eta) <= 0.5) * (pt > 0.1   && pt <= 5.0)   * (0.02) + \
(abs(eta) <= 0.5) * (pt > 5.0   && pt <= 1.0e2) * (0.015) + \
(abs(eta) <= 0.5) * (pt > 1.0e2 && pt <= 2.0e2) * (0.03) + \
(abs(eta) <= 0.5) * (pt > 2.0e2)                * (0.05 + pt*1.e-4) + \
(abs(eta) > 0.5 && abs(eta) <= 1.5) * (pt > 0.1   && pt <= 5.0)   * (0.03) + \
(abs(eta) > 0.5 && abs(eta) <= 1.5) * (pt > 5.0   && pt <= 1.0e2) * (0.02) + \
(abs(eta) > 0.5 && abs(eta) <= 1.5) * (pt > 1.0e2 && pt <= 2.0e2) * (0.04) + \
(abs(eta) > 0.5 && abs(eta) <= 1.5) * (pt > 2.0e2)                * (0.05 + pt*1.e-4) + \
(abs(eta) > 1.5 && abs(eta) <= 6.0) * (pt > 0.1   && pt <= 5.0)   * (0.04) + \
(abs(eta) > 1.5 && abs(eta) <= 6.0) * (pt > 5.0   && pt <= 1.0e2) * (0.035) + \
(abs(eta) > 1.5 && abs(eta) <= 6.0) * (pt > 1.0e2 && pt <= 2.0e2) * (0.05) + \
(abs(eta) > 1.5 && abs(eta) <= 6.0) * (pt > 2.0e2)                * (0.05 + pt*1.e-4)}}
```

Jets/MET

Two options:

- A simple smearing of jet and MET (made up from genparticles)
- Or an implementation of a simple calorimeter

  - granularity
  - resolution
  - fraction of energy deposit per particle

Output: towers (full deposit) or EflowTowers (full deposit minus charged deposit)

First option: - easy to change the jet resolution
- but no way to study jet substructure, PU mitigation, …

Second option: - can also do substructure, PU mitigation, …
- but less direct to change resolution

Calorimeters

First option (simple calorimeter):

ECAL: granularity : 5 degrees (in phi) for eta<1.6,
10 degrees for eta<4.4,
20 degrees for eta<6.0

```
# set ECalResolutionFormula {resolution formula as a function of eta and energy}
 set ResolutionFormula {      (abs(eta) <= 4.0) * sqrt(energy^2*0.005^2 + energy*0.02^2) + \
                        (abs(eta) > 4.0 && abs(eta) <= 6.0) * sqrt(energy^2*0.05^2  + energy*1.00^2)}
```

HCAL: granularity : 5 degrees (in phi) for eta<1.6,
10 degrees for eta<4.4,
20 degrees for eta<6.0

```
# set HCalResolutionFormula {resolution formula as a function of eta and energy}
 set ResolutionFormula {       (abs(eta) <= 4.0) * sqrt(energy^2*0.03^2 + energy*0.50^2) + \
                        (abs(eta) > 4.0 && abs(eta) <= 6.0) * sqrt(energy^2*0.05^2 + energy*1.00^2)}
```

Do we want longitudinal segmentation ?

Jet & MET

Then:

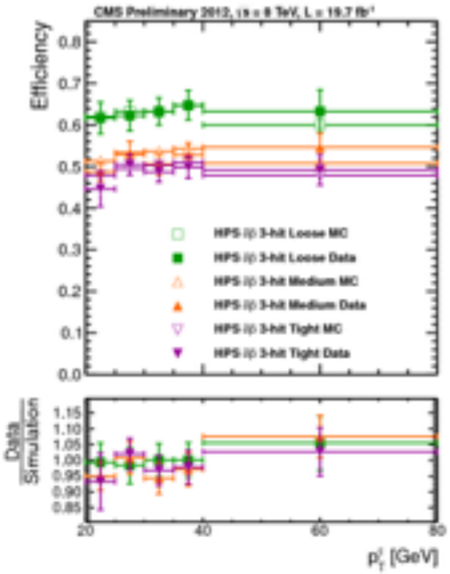
jets:
- default Anti-Kt (FastJet)
- using Eflow objects as input
- default cone 0.4?
- default PT>30 GeV?

MET = negative vector sum of Eflow objects
if smearing, would be negative vector sum of jets, leptons and photons

Note: no PU at the moment. PU configurations can be foreseen, but requires a separate card. Which PU scenario?

Tau-tagging

e.g. from CMS



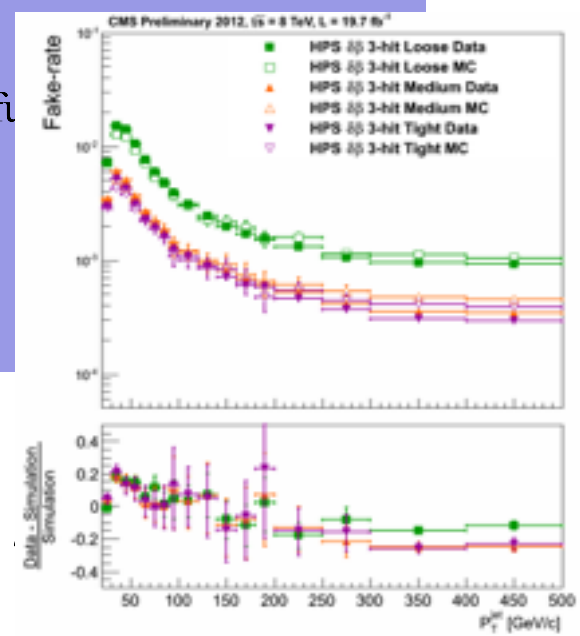
Parametrized description (40% efficiency, 0.1% fake rate)

```
module TauTagging TauTagging {
  set ParticleInputArray Delphes/allParticles
  set PartonInputArray Delphes/partons
  set JetInputArray JetEnergyScale/jets

  set DeltaR 0.4
  set TauPTMin 15.0
  set TauEtaMax 6

  # add EfficiencyFormula {abs(PDG code)} {efficiency formula as a fu

 add EfficiencyFormula {0} {0.01} (misid)
 add EfficiencyFormula {11} {0.005} (electron)
 add EfficiencyFormula {15} {0.6} (taus)
}
```



Need to check if this is only hadronic taus

Would a tighter working point make sense

B-tagging

- Delphes by default is a simple track counting algorithm for b-tagging
- Current LHC algorithms are far more sophisticated and obtain significant performance gains
- Suggest instead using parametrised efficiency and fake rate: (75% b-tagging, 25 c-jet rejection, 1000 l-jet rejection)
- set DeltaR 0.4
- set BPTMin 30.0
- set BEtaMax 4.0
- add EfficiencyFormula {1,2,3} {0.001}
- add EfficiencyFormula {4} {0.04}
- add EfficiencyFormula {5} {0.75}

Requirement on impact parameter resolution?

Add a separate charm tagger?

FCC event database

Clement Helsens has set up a database that collects FCC samples:

https://test-fcc.web.cern.ch/test-FCC/FCCevents.php

It contains two types of samples:
- LHE files
- generated events in FCC event data format

	name	nevents	nfiles	outputdir	mainprocess	finalstates
1	B-4p-0-1	91000	10	/eos/fcc/hh/generation/snowmass/FCCEvents/v0_0/B_4p/B-4p-0-1_100TEV/	vector boson + jets	V+0J
2	BB-4p-0-300	55000	10	/eos/fcc/hh/generation/snowmass/FCCEvents/v0_0/BB_4p/BB-4p-0-300_100TEV/	divector boson + jets	V+nJ
3	BB-4p-1400-2900	55000	10	/eos/fcc/hh/generation/snowmass/FCCEvents/v0_0/BB_4p/BB-4p-1400-2900_100TEV/	divector boson + jets	V+nJ
4	BB-4p-2900-5300	46177	10	/eos/fcc/hh/generation/snowmass/FCCEvents/v0_0/BB_4p/BB-4p-2900-5300_100TEV/	divector boson + jets	V+nJ
5	BB-4p-300-1400	55000	10	/eos/fcc/hh/generation/snowmass/FCCEvents/v0_0/BB_4p/BB-4p-300-1400_100TEV/	divector boson + jets	V+nJ
6	BB-4p-5300-8800	31498	10	/eos/fcc/hh/generation/snowmass/FCCEvents/v0_0/BB_4p/BB-4p-5300-8800_100TEV/	divector boson + jets	V+nJ
7	BB-4p-8800-100000	9392	10	/eos/fcc/hh/generation/snowmass/FCCEvents/v0_0/BB_4p/BB-4p-8800-100000_100TEV/	divector boson + jets	V+nJ
8	BBB-4p-0-1200	55000	10	/eos/fcc/hh/generation/snowmass/FCCEvents/v0_0/BBB_4p/BBB-4p-0-1200_100TEV/	tri-vector + jets, Higgs associated + jets	(VVV+nJ),(

Please help in the validation and provide feedback!

Conclusion

- Outlined components needed to put together a baseline Delphes card
- Tried to make it as independent as possible of the detector technology
- Input needed to finalise appropriate performance goals
  - All numbers are currently preliminary and up for discussion