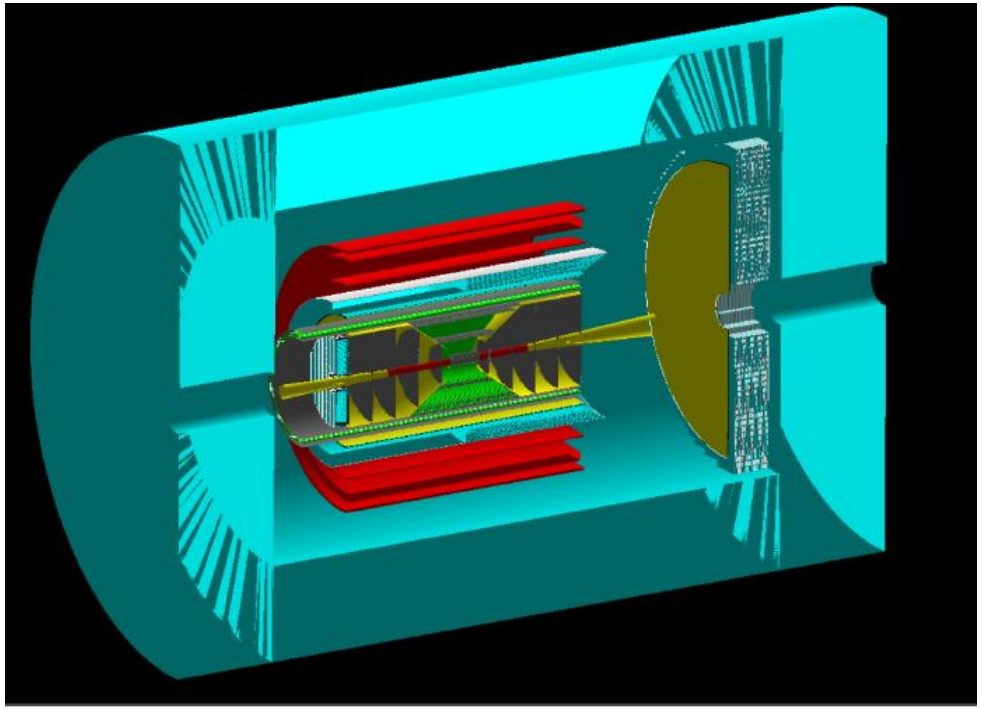
# CORE Implementation in Delphes: Progress Update

Joseph Grassi

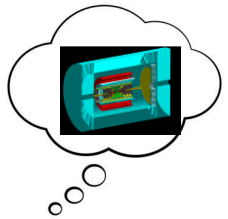


DELPHES
fast simulation

# Reasons to use:

**Low requirements:**
- ROOT
- Delphes
- Event file (hepmc format)
- Detector Card (a single .tcl) file

**Fast:**
1 million events takes approx. 8 minutes. . .
. . .on a virtual machine. . .
. . .running on a laptop.




**Complementary to Full Simulation:**
- Detector card parameters determined from full simulation
- Future full simulation informed by fast simulation

**Easy Analysis:**
- Data in ROOT tree, easy to analyze with macros

# Implementation In Progress:

- **Silicon tracker: Mostly implemented**
  - Needs extensions to d0 and dZ parametrizations for eta>2.5
  - Temporarily extended with constant values-poor approximation
  - Interpolation of B field between 1.4 and 3 T implemented

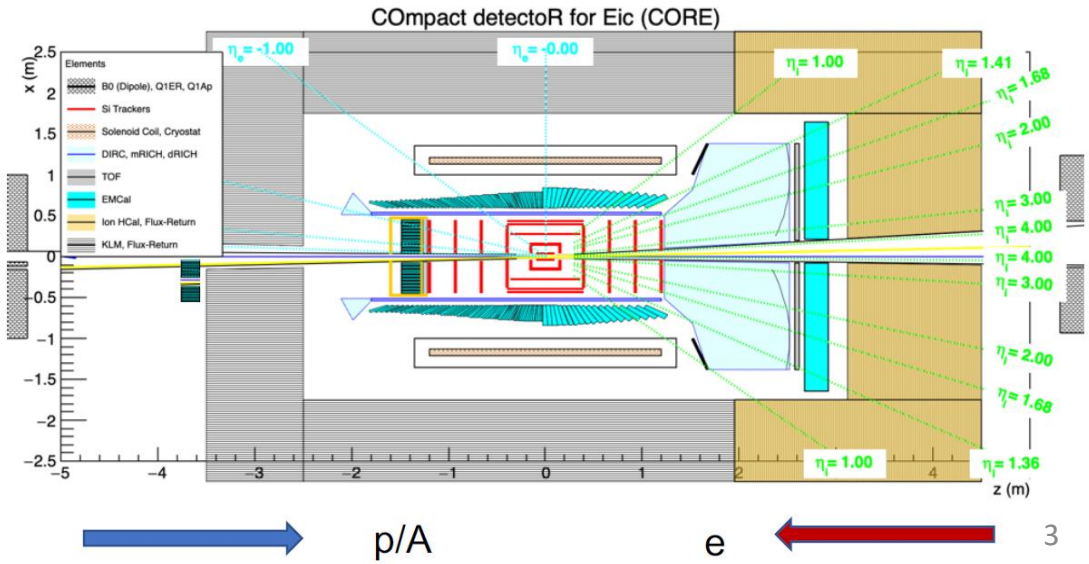
- **PID: Mostly Implemented**
  - DIRC and dRICH implemented: Regions without efficiency parametrization were extended with constants
  - Efficiency doesn't vary *that* much- reasonable approx. for fast simulation

**Summary Table**

| | | $\delta p/p = Ap \oplus B$ | | $\mathrm{DCA}_z = A/p_T \oplus B$ | | $\mathrm{DCA}_T = A/p_T \oplus B$ | |
|---|---|---|---|---|---|---|---|
| | | A [%/GeV] | B [%] | A [$\mu$m GeV] | B [$\mu$m] | A [$\mu$m GeV] | B [$\mu$m] |
| $0.0 < \eta < 0.5$ | B = 3.0T | 0.018 | 0.369 | 26.6 | 3.24 | 25.0 | 4.87 |
| | B=1.4T | 0.038 | 0.816 | 27.1 | 3.33 | 26.2 | 3.88 |
| $0.5 < \eta < 1.0$ | B = 3.0T | 0.016 | 0.428 | 36.8 | 3.79 | 28.5 | 4.49 |
| | B=1.4T | 0.035 | 0.898 | 35.1 | 3.79 | 31.2 | 4.04 |
| $1.0 < \eta < 1.5$ | B = 3.0T | 0.016 | 0.427 | 55.9 | 5.89 | 33.1 | 5.46 |
| | B=1.4T | 0.035 | 0.921 | 56.2 | 5.41 | 35.2 | 5.10 |
| $1.5 < \eta < 2.0$ | B = 3.0T | 0.012 | 0.462 | 108.2 | 8.74 | 39.1 | 5.46 |
| | B=1.4T | 0.026 | 0.997 | 106.3 | 8.36 | 39.8 | 5.21 |
| $2.0 < \eta < 2.5$ | B = 3.0T | 0.018 | 0.719 | 207.3 | 19.77 | 45.1 | 9.54 |
| | B=1.4T | 0.041 | 1.548 | 201.8 | 21.50 | 46.3 | 9.31 |
| $2.5 < \eta < 3.0$ | B = 3.0T | 0.039 | 1.336 | | | | |
| | B=1.4T | 0.088 | 2.830 | | | | |
| $3.0 < \eta < 3.5$ | B = 3.0T | 0.103 | 2.428 | | | | |
| | B=1.4T | 0.217 | 5.234 | | | | |
| $3.5 < \eta < 4.0$ | B = 3.0T | 0.295 | 4.552 | | | | |
| | B=1.4T | 0.610 | 9.797 | | | | |

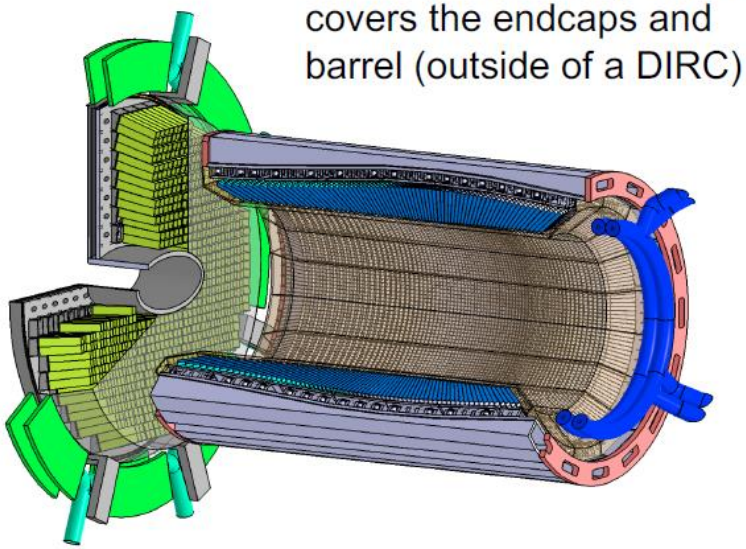
???

https://indico.bnl.gov/event/7913/ Rey Cruz-Torres Slides



# Implementation In Progress Contd:

- **EMcal: Preliminary Implementation**
  - Efficiencies taken from matrix detector
- **Hcal: Preliminary Implementation**
  - Efficiencies taken from matrix detector
- **KLM: Not implemented yet**
  - Using 100% Hcal from allsilicon detector
  - ETA 2 weeks
- Matrix and allsilicon detector card can still be used as well
- Will be available on CORE wiki within a few days: https://eic.jlab.org/core/index.php/Main_Page



# Installation

Linux:

```
wget http://cp3.irmp.ucl.ac.be/downloads/Delphes-3.5.0.tar.gz
tar -zxf Delphes-3.5.0.tar.gz

cd Delphes-3.5.0
make
```

Test run with:

./DelphesHepMC (or ./DelphesHepMC3)

Output should be:

```
(base) joseph@joseph-VirtualBox:~/Documents/delphes$ ./DelphesHepMC
 Usage: DelphesHepMC config_file output_file [input_file(s)]
 config_file - configuration file in Tcl format,
 output_file - output file in ROOT format,
 input_file(s) - input file(s) in HepMC format,
 with no input_file, or when input_file is -, read standard input.
```

Windows:
Please don't.

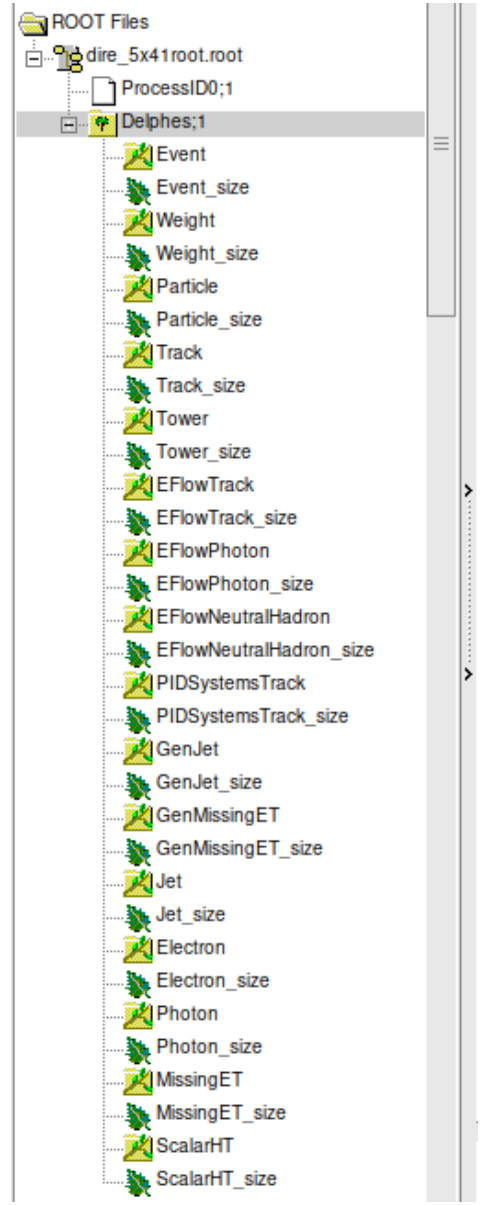
If otherwise, you may need to configure ROOT:
Simply cd into directory root-6.24.00-bin/root/bin and use command source thisroot.sh

```
./DelphesHepMC: error while loading shared libraries: libCore.so.6.24: cannot op
en shared object file: No such file or directory
```

# The detector card

- A .tcl file consisting of an execution path, and modules
- Execution path specifies order of execution of the modules
- Modules specify efficiencies, binning, smearing, and create ROOT branches

```
set ExecutionPath {
  ParticlePropagator

  ChargedHadronTrackingEfficiency
  ElectronTrackingEfficiency
  #MuonTrackingEfficiency
  #May or may not be necessary. Get KLM info before implementing

  ChargedHadronMomentumSmearing
  ElectronMomentumSmearing



  TrackMerger
  TrackSmearing

  ECal
  HCal

  Calorimeter
  EFlowMerger
  EFlowFilter

  PhotonEfficiency
  PhotonIsolation

  ElectronFilter
```



# Running delphes:

Simple command from main delphes directory. Make sure to get the directories correct.

./DelphesHepMC cards/detectorcardname.tcl rootoutputname.root eventfilename.hepmc
Or
./DelphesHepMC3 cards/detectorcardname.tcl rootoutputname.root eventfilename.hepmc

```
#--------------------------------------------------------------------------
#                         FastJet release 3.3.4
#                 M. Cacciari, G.P. Salam and G. Soyez
#     A software package for jet finding and analysis at colliders
#                           http://fastjet.fr
#
# Please cite EPJC72(2012)1896 [arXiv:1111.6097] if you use this package
# for scientific work and optionally PLB641(2006)57 [hep-ph/0512210].
#
# FastJet is provided without warranty under the GNU GPL v2 or higher.
# It uses T. Chan's closest pair algorithm, S. Fortune's Voronoi code
# and 3rd party plugin jet algorithms. See COPYING file for details.
#--------------------------------------------------------------------------
** INFO: initializing module  GenMissingET
** INFO: initializing module  FastJetFinder
** INFO: initializing module  JetEnergyScale
** INFO: initializing module  JetFlavorAssociation
** INFO: initializing module  GenJetFlavorAssociation
** INFO: initializing module  UniqueObjectFinder
** INFO: initializing module  ScalarHT
** INFO: initializing module  TrackCountingBTagging
** INFO: initializing module  mRICH
** INFO: initializing module  barrelDIRC
** INFO: initializing module  dualRICH_aerogel
** INFO: initializing module  dualRICH_c2f6
** INFO: initializing module  TreeWriter
** Reading eic-core/Coherent_HepMC2-1H.hepmc
** [##################################################################] (100.00%)
** Exiting...
```

Extremely simple! ROOT file can now be analyzed as usual.

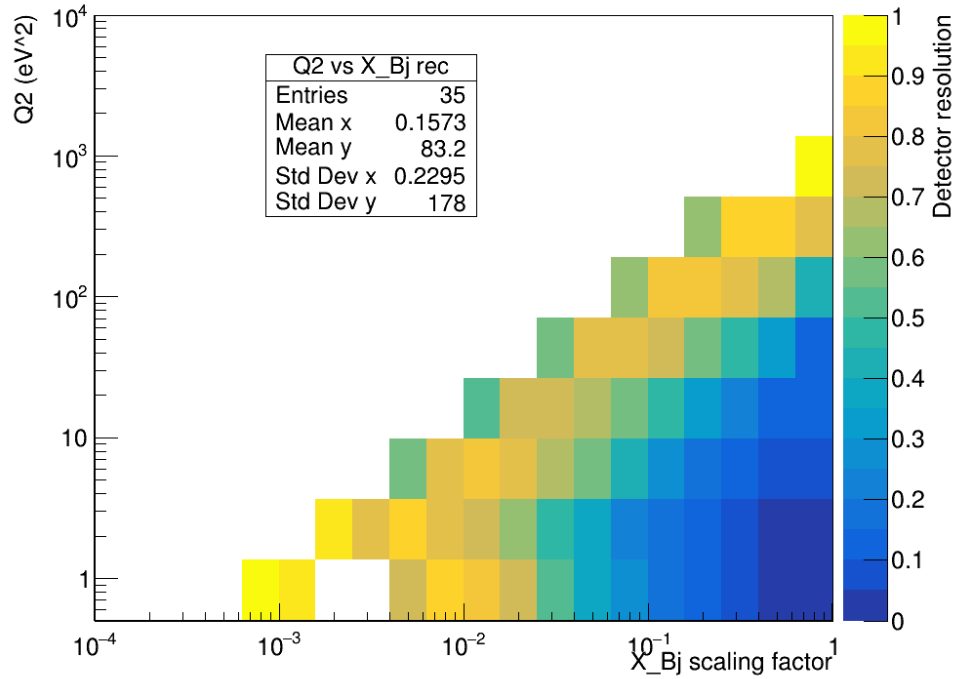
# Physics Analysis: An example

Example ROOT Macro:

- Available soon along with detector card on CORE wiki

- Initializes ROOT file
- Iterates through all events and finds scattered electron
- Calculates $Q^2$ and $X_{Bj}$ for original and reconstructed electron
- Logarithmically binned histogram
- Final output is probability of reconstructed electron being found in same bin as true electron

More interesting physics analysis can also be performed, utilizing Jets and PID




```
176
177     //Root tree initialization
178     TChain chain("Delphes");
179     chain.Add("/home/joseph/Documents/delphes/dire_5x41root.root");
180     ExRootTreeReader *treeReader = new ExRootTreeReader(&chain);
181     TClonesArray *branchEvent=treeReader->UseBranch("Event");
182     TClonesArray *branchParticle = treeReader->UseBranch("Particle");
183     TClonesArray *branchElectron = treeReader->UseBranch("Electron");
184     Long64_t eventCount = treeReader->GetEntries();
185     GenParticle *recElectronParticle;
186     Long64_t entry;
```

# Resources:

- Delphes github: https://github.com/delphes/delphes
- Delphes homepage: https://cp3.irmp.ucl.ac.be/projects/delphes
- In depth Delphes explanation (pdf): https://www.tcm.phy.cam.ac.uk/~mjh261/pdfs/delphes.pdf
- delphes_EIC github: https://github.com/eic/delphes_EIC
  - Includes two detector cards: matrix and allsilicon. Former is direct from Yellow Report requirements. The latter contains silicon tracking, EMcal and HCal.
- CORE wiki: https://eic.jlab.org/core/index.php/Main_Page

Thank you!