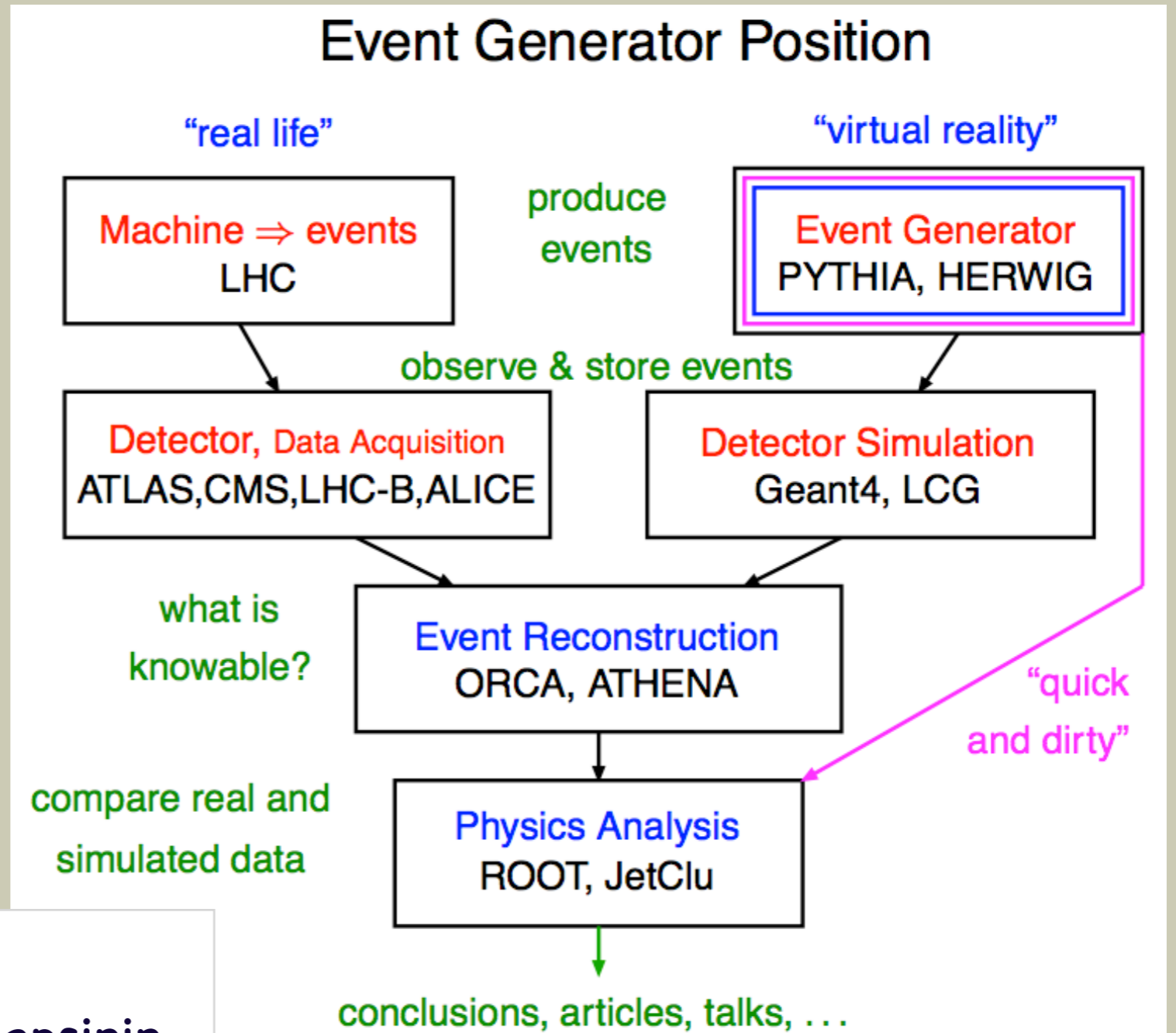
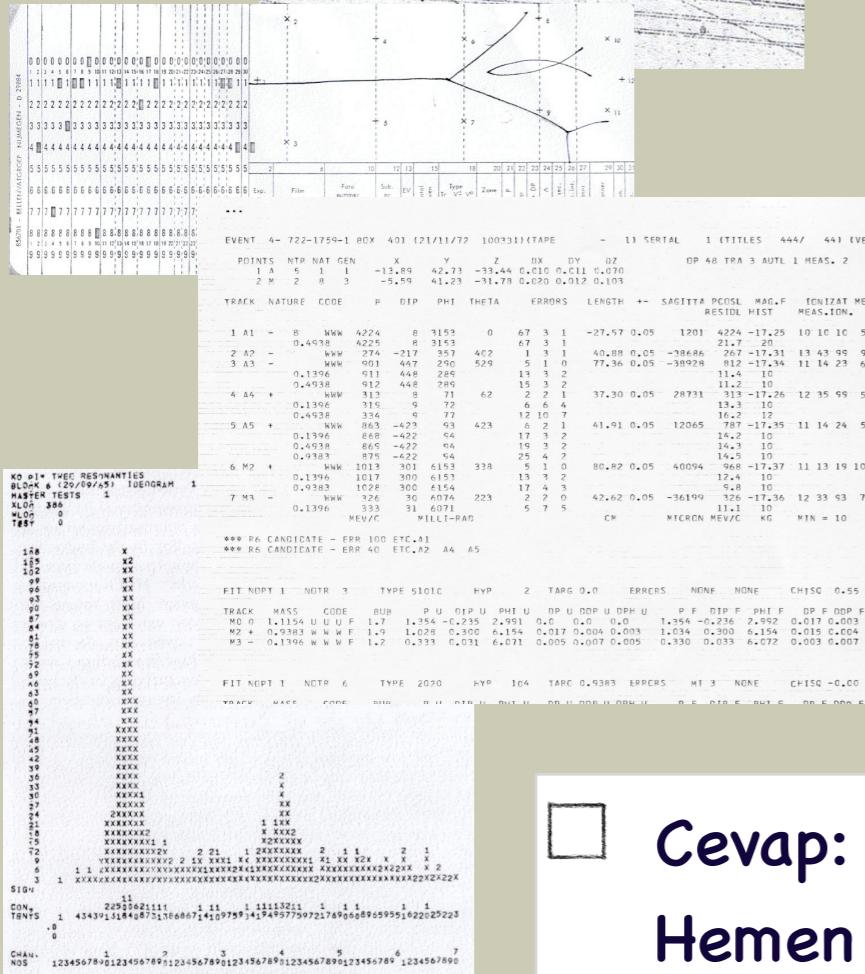
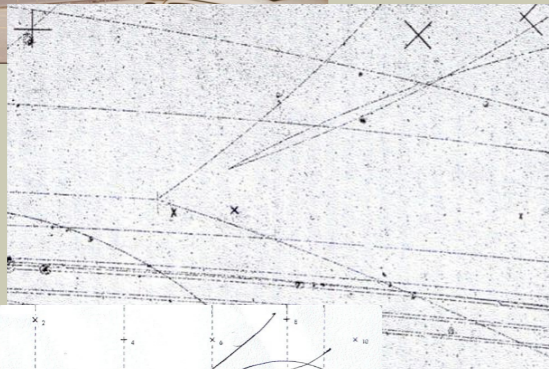


# HIZLI ALGIÇ BENZETİMİ

V. Erkcan ÖZCAN & Sezen SEKMEN



# Parçacık Deneylerinde Neyin Benzetimini Yapıyoruz?



Cevap:  
Hemen hepsinin...



# Hızlı Benzetim

- \* Neden? Geant4 gibi tam benzetimler çok iyi ama çok yavaş.
  - \* LHC için bir olayın Geant'dan geçirilmesi dakikalar alıyor, bizimse milyarca olay üretmemiz lazım.
- \* Çözüm: hızlı benzetim teknikleri...
  - \* Sadece en yavaş kısımları değiştirip başka şekillerde modelleme.
  - \* Tüm algııcı modelleyip Geant'ı tamamen bırakma.
    - \* Olayın incelenip, izlerin sürülmesi vs. gibi adımları da modelleme.
- \* Eski örnekler: AcerDet, ATLFast, FAMOS, PGS
  - \* Şimdiki standart: Delphes

# Bir Algıcın Cevabının Modellenmesi

## ● İz sürme (İç Algıç) $|\eta| < 2.5$

- 2T solenoid ile
- Silisyum (pixel + strip) izsürücü
- Geçiş ışınımı izsürücüsü (TRT)  $e/\pi$  ayrışımı
- b işaretleme etkinliği %50

## ● Enerji Ölçümü $|\eta| < 5.0$

- EM kalorimetre : Pb-LAr, akordiyon şeklinde
- Hadron kalorimetre : Fe/Sci (merkez), Cu/W-LAr (ön kısım)

## ● Muon Ölçümü $|\eta| < 2.7$

- 4T toroid spektrometre
- MDT ve CSC : iz sürmek için
- RPC ve TGC : hızlı tetiklemek için

RPC: Resistive Plate Chamber  
MDT: Monitored Drift Tubes  
CSC: Cathode Strip Chamber  
TGC: Thin Gap Chamber

$$\text{ID: } \sigma/p_T \approx 5 \times 10^{-4} p_T \oplus 0.001$$

$$\sigma(d_0) = 15 \mu\text{m at } 20 \text{ GeV}$$

$$\text{ECAL: } \sigma/E \approx 10\% / \sqrt{E(\text{GeV})} \oplus 0.7\%$$

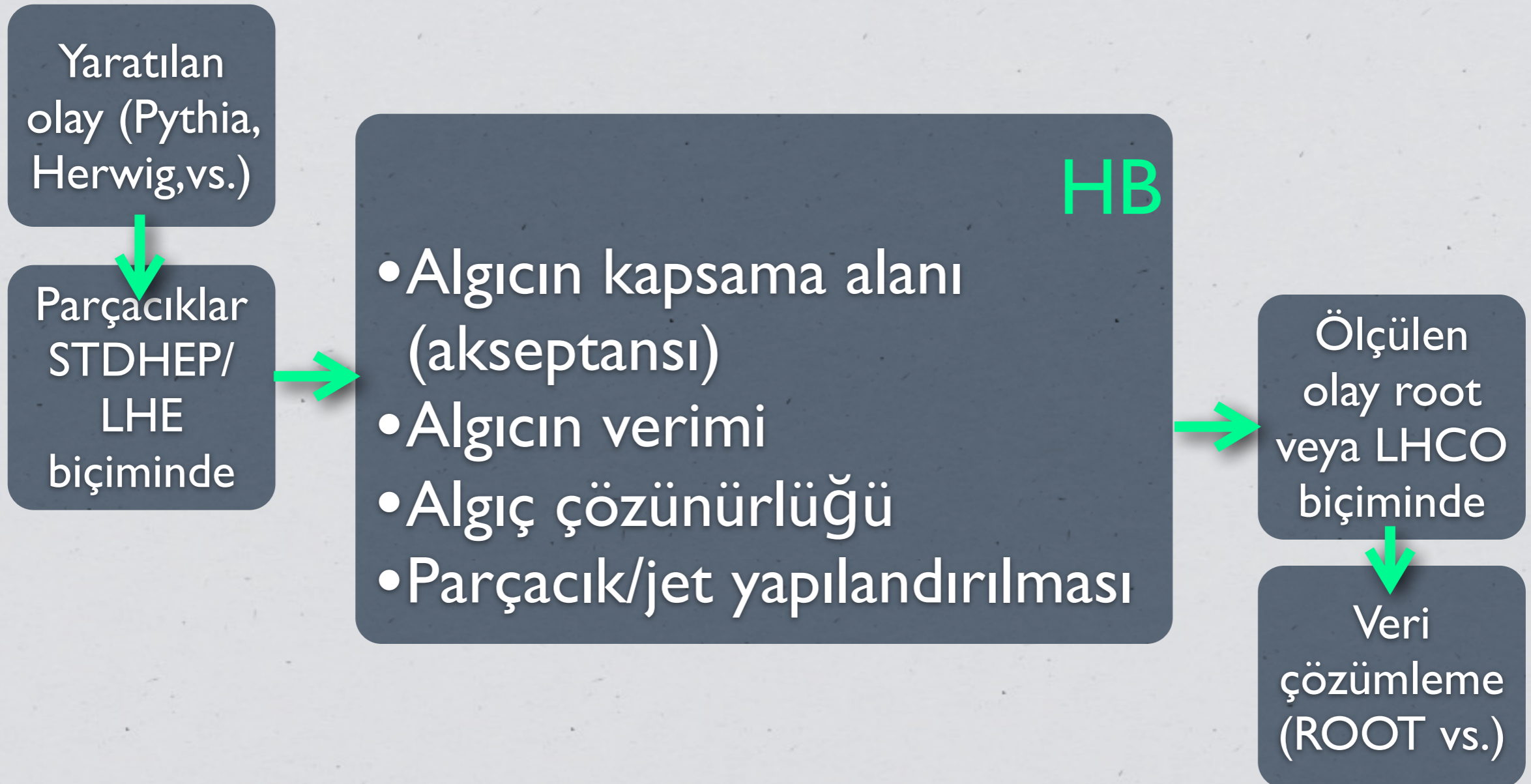
$$\text{HCAL: } \sigma/E \approx 50\% / \sqrt{E(\text{GeV})} \oplus 3\%$$

$$\text{Muon: } \sigma/p_T \approx 10\% \text{ at } 1 \text{ TeV}/c$$

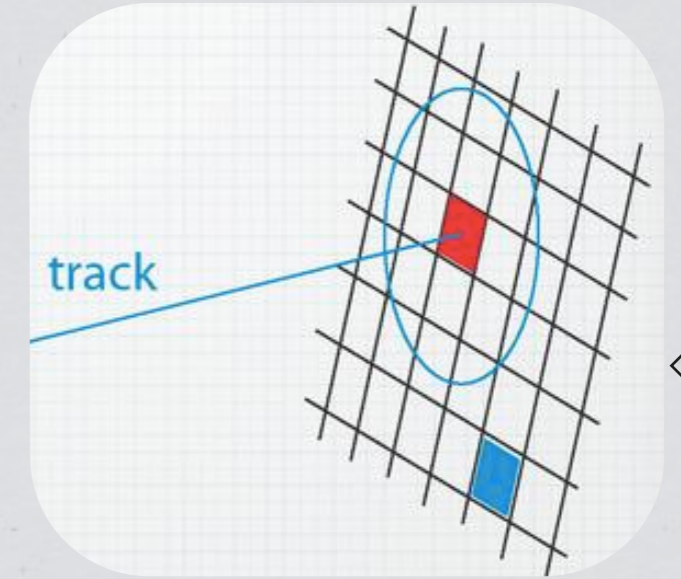




# Hızlı benzetimden ne bekliyoruz?



# Kabaca Ne Yapıyor?



- \* Tüm son durum parçacıklarını işle, eğer algıcın sınırları içindeyse:
- \* Eğer yüklü parçacıksa, onun için izleri çıkar, manyetik alanda izin sagittasının çözünürlüğünü hesaba al.
- \* Kalorimetre  $\eta$  ve  $\phi$  düzleminde karelere ayrılmış. Parçacıklar hangi kareye yönlenmişlerse orada enerjilerini yerleştir.
- \* elektron/foton: hemen tüm enerji EM kalorimetresine.
- \* hadronlar: çoğu enerji hadronik kalorimetreye.
- \* muonlar: azami iyonlaşma.
- \* Enerji çözünürlüklerini uygula:  $\Delta E/E = a \oplus b/\sqrt{E}$
- \* Jetleri yapılandır, gerekirse ağır jetleri işaretle.



# Örnek HB kodu - Kalorimetrelerde Müonlar

```
call pgs_rannor(r1,r2)
mu_ecal = 0.5 + r1*0.1 ! estimated from Mon
if(mu_ecal.lt.0.) mu_ecal = 0.
e_ecal = 0.
if (e.gt.mu_ecal) then
  e_ecal = mu_ecal
  e = e - mu_ecal
else
  e_ecal = e
  e = 0.
endif
mu_hcal = 2.0 + r2*0.4 ! estimated from Mo
if(mu_hcal.lt.0.) mu_hcal = 0.
e_hcal = 0.
if (e.gt.mu_hcal) then
  e_hcal = mu_hcal
  e = e - mu_hcal
else
  e_hcal = e
  e = 0.
endif
```

İki tane Gaussian dağılımlı rastgele sayı üret

EM-kalorimetresinde kaybedilen enerji :  
ortalaması 0.5 GeV, standart sapması 0.1 GeV  
olan Gaussian, ancak negatif olamıyor

Eğer kaybedilen enerji, parçacığın enerjisinden  
fazlaysa: kalorimetreye parçacığın tüm  
enerjisini koy, parçacığın enerjisini sıfırla.

Hadron kalorimetresinde de aynı hikaye:  
kaybedilen enerji ortalaması 2.0 GeV, standart  
sapması 0.4 GeV olan ve negatif olamayan bir  
Gaussian.

Fortran eğreti-koda benzediği için PGS4'den aldık.

# Örnek HB kodu - Kalorimetrelerde Hadronlar

```
ecal_frac = 0.25           ! 25% of energy in electromagnetic
call pgs_rannor(r1,r2)      ! calorimeter...
ecal_frac = ecal_frac + 0.05*r1 ! ...with some spread in fraction...

if (ecal_frac.le.0.) ecal_frac = 0.0
if (ecal_frac.ge.1.) ecal_frac = 1.0

if(pgs_ranmar().lt.0.5) then           ! ...but only half the time does this occur
  e_ecal = 0.5 + 0.1*r2                ! in which case we treat it like a muon
  if(e_ecal.ge.e) then                 ! in the e.m. calorimeter
    ecal_frac = 1.
  elseif(e_ecal.lt.0.) then
    ecal_frac = 0.
  else
    ecal_frac = e_ecal/e
  endif
endif

e_ecal = ecal_frac * e
e_hcal = (1.0-ecal_frac) * e

call pgs_rannor(res_ecal,res_hcal)

e_ecal = e_ecal + hcal_res*sqrt(e_ecal)*res_ecal ! ~80%/sqrt(E) for hadrons
if (e_ecal.lt.0.) e_ecal = 0.                   ! in e.m. calorimeter!?

e_hcal = e_hcal + hcal_res*sqrt(e_hcal)*res_hcal ! ~80%/sqrt(E)
if (e_hcal.lt.0.) e_hcal = 0.
```

Yüzde elli ihtimalle EM kalorimetresinde bir müon gibi enerji bırakıyor. Geri kalan yüzde elli durumda, EM kalorimetresinde enerjinin rastgele bir yüzdesi bırakılıyor.

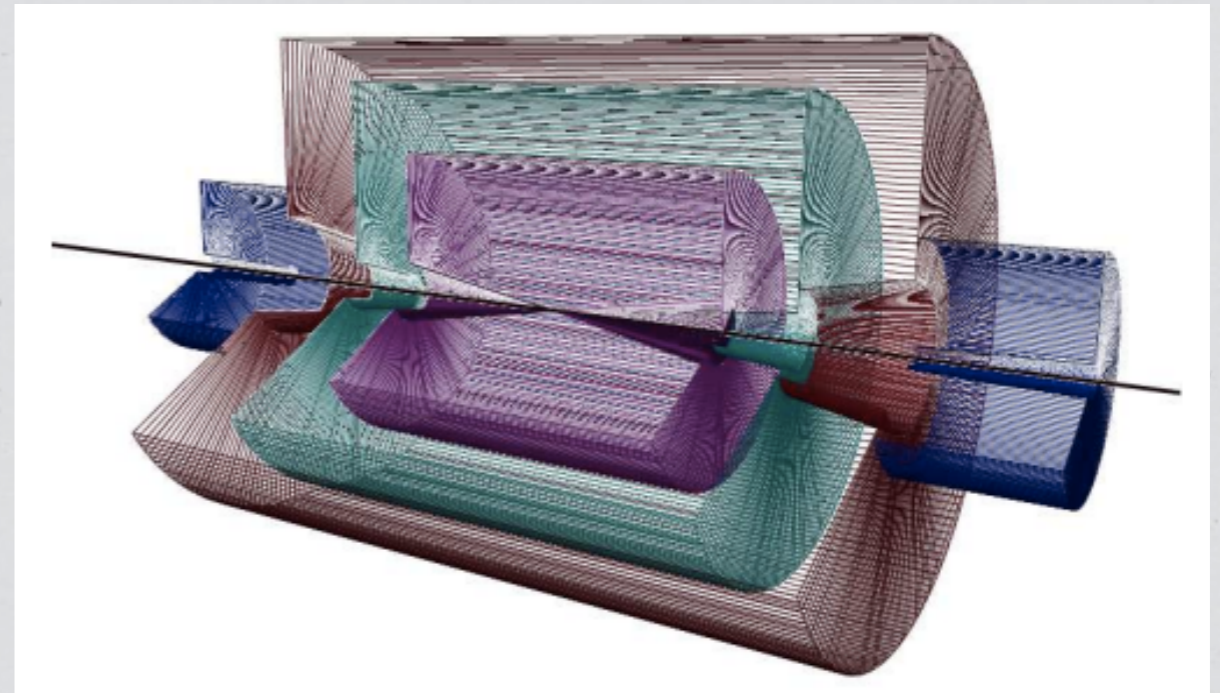
Sonra bırakılan enerji beklenen çözünürlüğe uygun bir şekilde yayılıyor.

Fortran eğreti-koda benzediği için PGS4'den aldık.



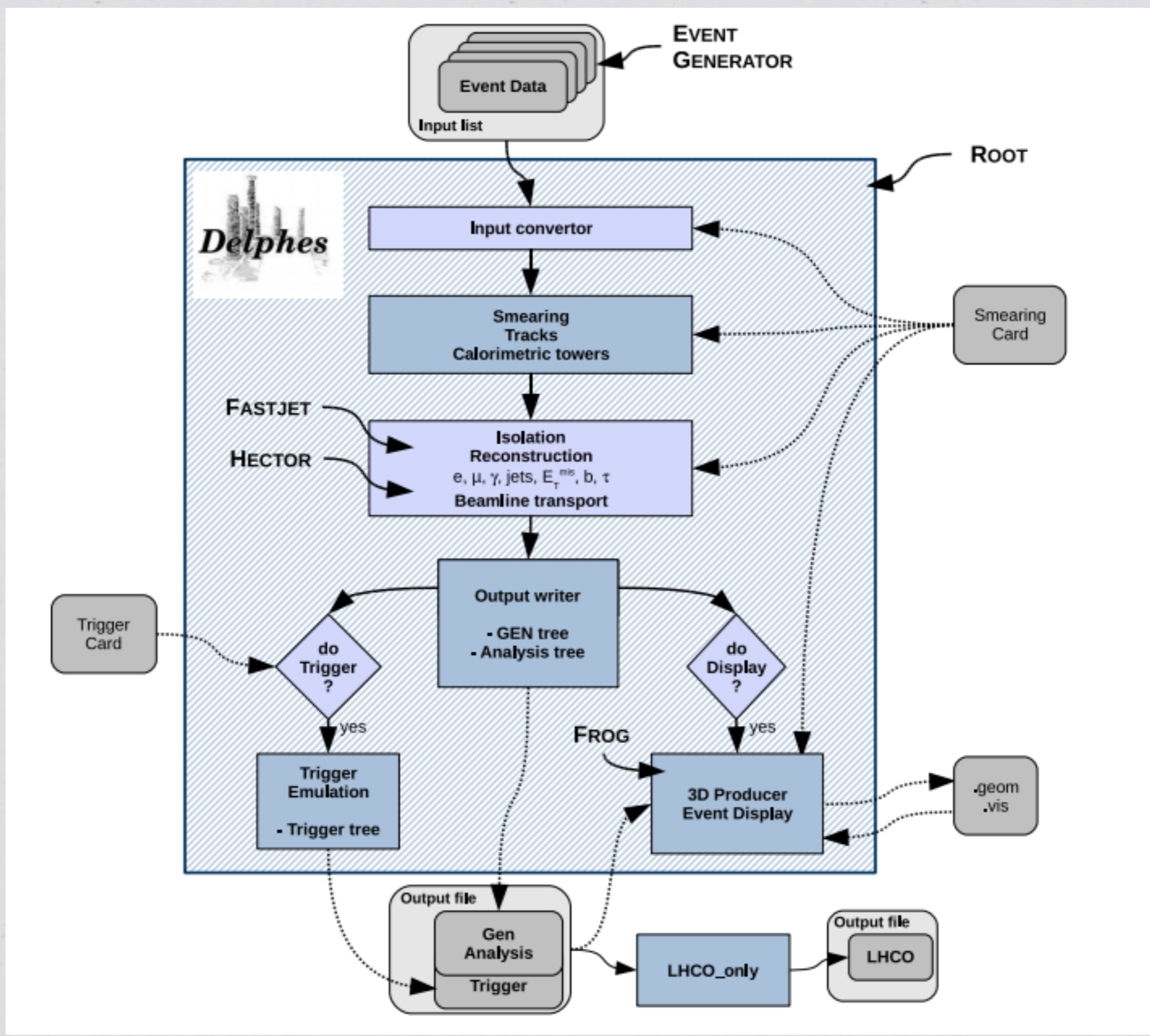
# Delphes

- \* S. Oryn, X. Rouby, V. Lemaitre, [arxiv.org/abs/0903.2225](https://arxiv.org/abs/0903.2225) (Delphes 2)
- \* C++ ile yazılmış. ROOT kullanıyor.
- \* Birçok açıdan önceki nesil HBlerden üstün.
  - \* İleri ve sıfır-derece kalorimetreler
  - \* Enerji çözünürlüklerinin matematiksel modelleri ve kalorimetre hücrelerinin dağılımları daha detaylı.
  - \* Siscone ve midpoint jet algoritmaları eklenmiş.
  - \* ATLAS ve CMS için sonuçları test edilmiş.
  - \* STDHEP ve LHE dosyalarını girdi olarak kabul ediyor. LHCO çıktısı dışında daha detaylı root dosyaları veriyor.
  - \* Düzenli olarak güncelleniyor. Geçen yıl Delphes 3 çıktı.



<https://server06.fynu.ucl.ac.be/projects/delphes>



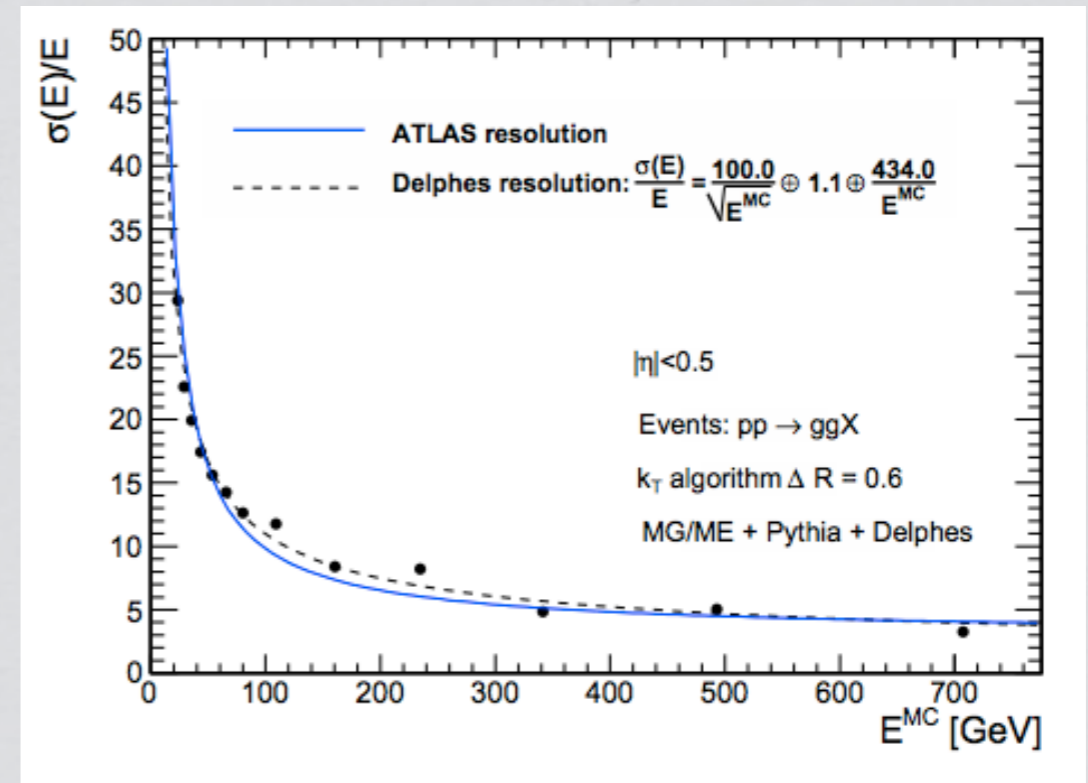
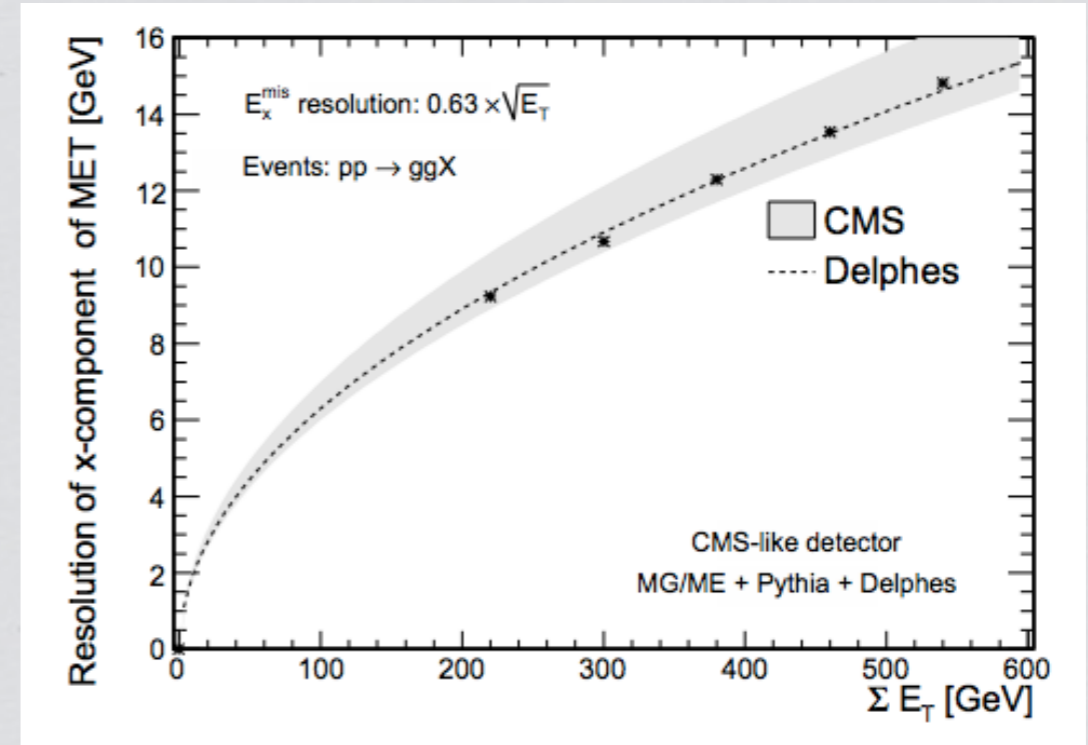




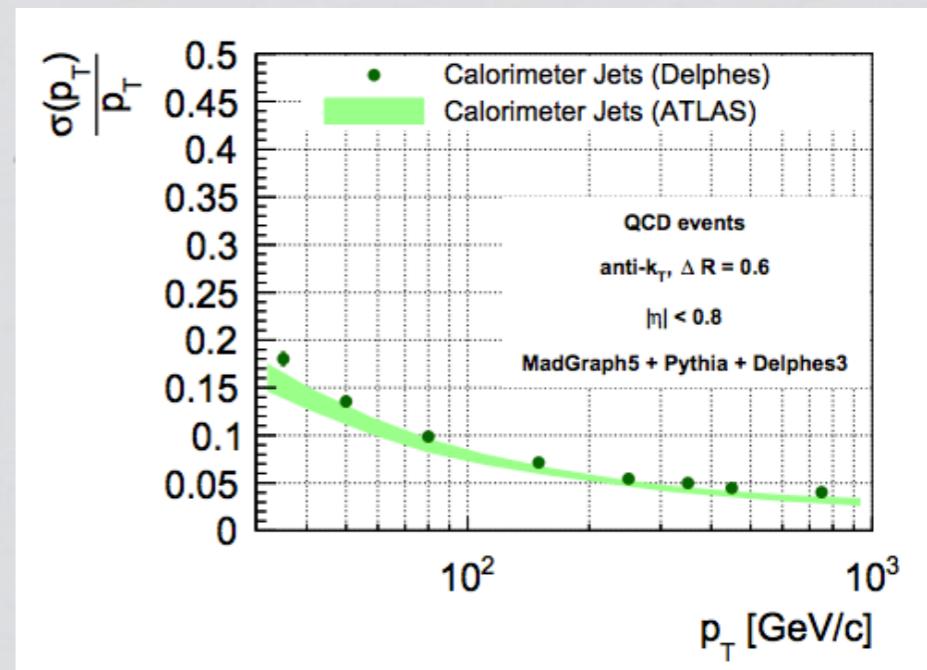
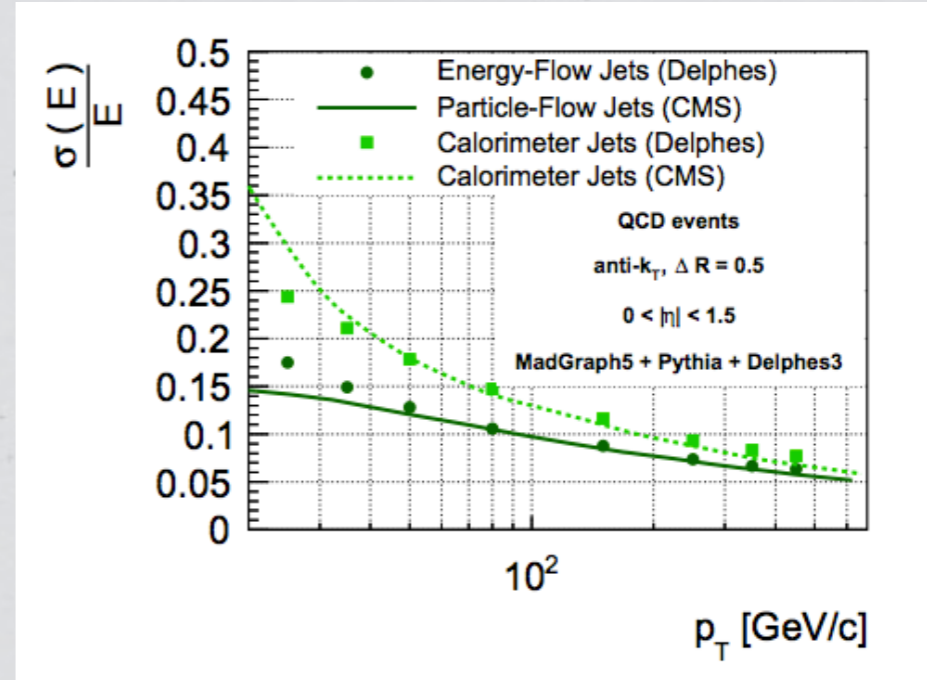
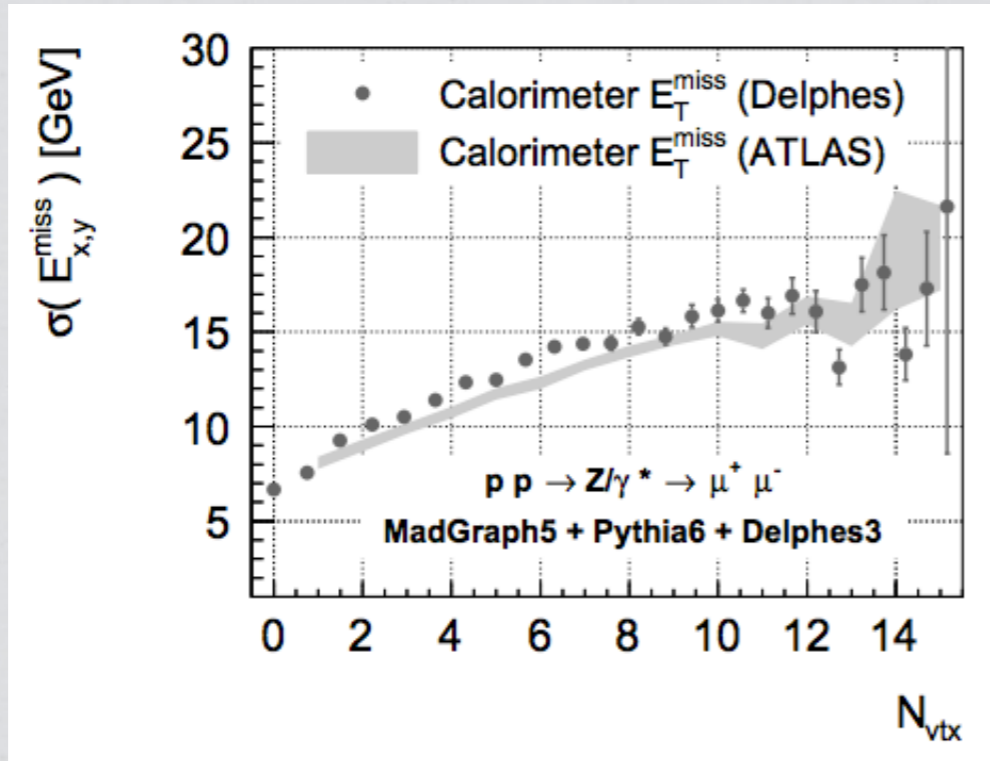
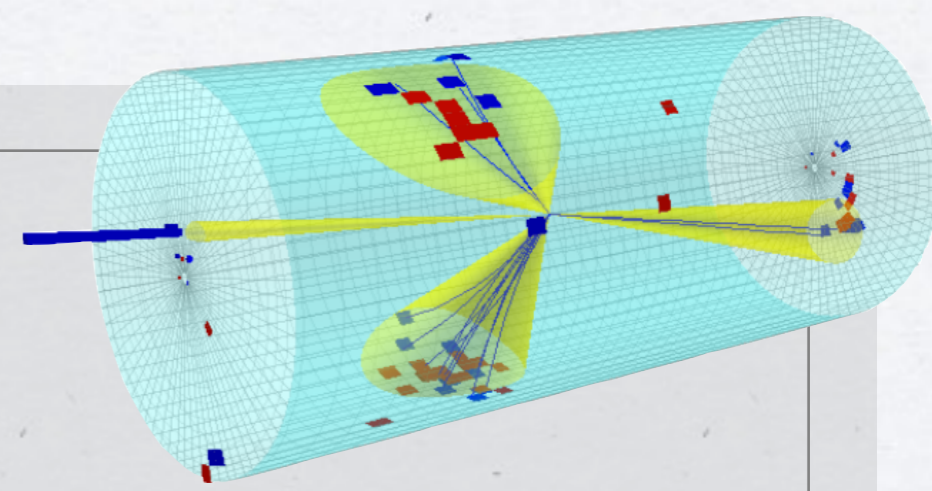
# Delphes2 Testleri

	CMS	Delphes
$Z \rightarrow \tau^+\tau^-$	38.2%	$32.4 \pm 1.8\%$
$H(140) \rightarrow \tau^+\tau^-$	36.3%	$39.9 \pm 1.6\%$
$H(300) \rightarrow \tau^+\tau^-$	47.3%	$49.7 \pm 1.5\%$

- \* Delphes'in alt benzetim adımları sonucunda çıkardığı jet enerjisi, kayıp enerji, tau jet bulma işlemlerinin tam Geant4 benzetimi + gerçek olay yapılandırması sonuçlarıyla karşılaştırılması...
- \* Çıkarılan enerji çözünürlükleri ve tau jet bulma etkinlikleri ATLAS ve CMS deneylerinin performansını ile son derece tutarlı.



# Delphes3



- \* Yenilenmiş, modüler Delphes ...
- \* Olay görüntüleme.
- \* Parçacık-akışı (particle-flow) jetleri.
- \* Yığılma (pile-up) testleri.



# Pythia8 + Delphes ile Bir Örnek

\* Halihazırda mevcut olan örnek programlardan biri ile başlayalım.

```
> cd ~/hepWork/pythia8/examples  
> cp -p main20.cc z+jet.cc
```

\* Kodun nasıl derlenmesi gerektiğini görelim:

```
> make main20  
g++ -O2 -ansi -pedantic -W -Wall -Wshadow -fbounds-check -I../include  
main20.cc -o bin/main20.exe -L../lib/archive -lpythia8 -lhapdfdummy
```

\* Biz kendi dosyamızı derleyelim:

```
> g++ -O2 -ansi -pedantic -W -Wall -Wshadow -fbounds-check -I../include  
z+jet.cc -o bin/z+jet.exe -L../lib/archive -lpythia8 -lhapdfdummy
```

# Pythia8 php Arayüzü

home.thep.lu.se/~torbjorn/pythia81php/Welcome.php



## PYTHIA 8

### Welcome to PYTHIA - The Lund Monte Carlo!

PYTHIA 8 is the successor to PYTHIA 6, rewritten from scratch in C++. With the release of PYTHIA 8.1 it now has reached the same level of reliability as the older one. This testing will only happen if people begin to work with it, so we encourage a gradual transition to the new version, starting now. There are some new physics features in PYTHIA 8.1 but also some topics still missing, where 6.4 would have to be used. Further, many obsolete features will not be supported in PYTHIA 8.1. For studies again 6.4 would be the choice.

### Documentation

On these webpages you will find the up-to-date manual for PYTHIA 8.1. Use the left-hand index to navigate through all possible program settings. All parameters are provided with sensible default values, however, so you need to study, such as choice of beams, processes and phase space cuts. The pages also contain a fairly extensive summary of the produced events. What is lacking on these webpages is an overview, on the one hand, and an in-depth

The overview can be found in the attached PDF file

#### **A Brief Introduction to PYTHIA 8.1**

T. Sjöstrand, S. Mrenna and P. Skands, *Comput. Phys. Comm.* 178 (2008) 852 [arXiv:0710.3820].

You are strongly recommended to read this summary when you start out to learn how to use PYTHIA 8.1. Note the version described there.

For the physics description we refer to the complete

#### **PYTHIA 6.4 Physics and Manual**

T. Sjöstrand, S. Mrenna and P. Skands, *JHEP05 (2006) 026*,

which in detail describes the physics (largely) implemented also in PYTHIA 8, and also provides a more extensive

When you use PYTHIA 8.1, you should therefore cite both, e.g. like

**T. Sjöstrand, S. Mrenna and P. Skands, *JHEP05 (2006) 026, *Comput. Phys. Comm.* 178 (2008) 852.***

### PYTHIA 8 Index

#### Program Overview

- Frontpage
- Program Flow
- Settings Scheme
- Particle Data Scheme
- Program Files
- Program Classes
- Program Methods
- Sample Main Programs

#### Setup Run Tasks

- Save Settings
- Main-Program Settings
- Beam Parameters
- Random-Number Seed
- PDF Selection
- Master Switches
- Process Selection
  - QCD
  - Electroweak
  - Onia
  - Top
  - Fourth Generation

\* Process Selection -> Electroweak Processes -> 241, 242



# Pythia8 php Arayüzü

home.thep.lu.se/~torbjorn/pythia81.php/Welcome.php



## PYTHIA 8

Boson and parton

### PYTHIA 8 Index

#### Program Overview

- Frontpage
- Program Flow
- Settings Scheme
- Particle Data Scheme
- Program Files
- Program Classes
- Program Methods
- Sample Main Programs

#### Setup Run Tasks

##### Save Settings

- Main-Program Settings
- Beam Parameters
- Random-Number Seed
- PDF Selection
- Master Switches
- Process Selection
  - QCD
  - Electroweak
  - Onia
  - Top
  - Fourth Generation

**WeakBosonAndParton:all**  On  Off (default = off)

Common switch for the group of production of a single electroweak gauge boson, i.e. a  $\gamma^*/Z^0$  or a  $W^{+-}$ , in association with a parton, i.e. a quark, gluon, photon or lepton. These processes give first-order corrections to the ones in the **WeakSingleBoson** class, and both sets cannot be used simultaneously without unphysical doublecounting. The current class should only be used to study the high- $p_T$  tail of the gauge-boson production processes (for LHC applications at least  $p_T > 20$  GeV), while the ones in **WeakSingleBoson** should be used for inclusive production.

**WeakBosonAndParton:qqbar2gmZg**  On  Off (default = off)

Scattering  $q \bar{q} \rightarrow \gamma^*/Z^0 g$ . Code 241.

**WeakBosonAndParton:qg2gmZq**  On  Off (default = off)

Scattering  $q g \rightarrow \gamma^*/Z^0 q$ . Code 242.

**WeakBosonAndParton:ffbar2gmZgm**  On  Off (default = off)

Scattering  $f \bar{f} \rightarrow \gamma^*/Z^0 g$ . Code 243.

**WeakBosonAndParton:fgm2gmZf**  On  Off (default = off)

Scattering  $f g \rightarrow \gamma^*/Z^0 f$ . Code 244.

**WeakBosonAndParton:qqbar2Wg**  On  Off (default = off)

Scattering  $q \bar{q} \rightarrow W^{+-} g$ . Code 251.

\* Process Selection -> Electroweak Processes -> 241, 242

# Pythia8 php Arayüzü



## PYTHIA 8 Index

### Program Overview

- Frontpage
- Program Flow
- Settings Scheme
- Particle Data Scheme
- Program Files
- Program Classes
- Program Methods
- Sample Main Programs

### Setup Run Tasks

- Save Settings**
- Main-Program Settings
- Beam Parameters
- Random-Number Seed

## Save Settings

The information on this webpage is only valid if you access the PHP dynamic webpages via a web browser, and does not apply to the static HTML equivalents. With PHP, all of the settings in the PYTHIA program are represented by radio buttons or fill-in boxes, that makes it easy for you to construct a file with your desired changes. This file can then be read into PYTHIA by your main program to steer the whole run.

## Basic instructions

The functionality of the PHP option is described in the following.

Filename chosen: **temp**

[Finish File](#) [RESET](#)

- To begin with, you must specify a **(temporary) file name** in the box above. If the filename already exists on the server, you will be requested to pick a new name.
- Once you have **Submitted** your filename, you can browse through the pages and make your selections. The values currently selected when you load the page are the default values.
- When you have finished making your changes to a part of the page. This will write the changes to your temporary file again.

- **Right Click to Save Target: temp**

[RESET](#)

Bottom  
for that



# Les Houches Olay Çıktısı

## PYTHIA 8 output to an LHEF

The above methods could be used by any program to write an LHEF. For PYTHIA 8 to do this, a derived class already exists, `LHAupFromPYTHIA8`. In order for it to do its job, it must gain access to the information produced by PYTHIA, specifically the `process` event record and the generic information stored in `info`. Therefore, if you are working with an instance `pythia` of the `Pythia` class, you have to instantiate `LHAupFromPYTHIA8` with pointers to the `process` and `info` objects of `pythia`:

```
LHAupFromPYTHIA8 myLHA(&pythia.process, &pythia.info);
```

The method `setInit()` should be called to store the `pythia` initialization information in the LHA object, and `setEvent()` to store event information. Furthermore, `updateSigma()` can be used at the end of the run to update cross-section information, cf. `closeLHEF(true)` above. An example how the generation, translation and writing methods should be ordered is found in `main20.cc`.

Currently there are some limitations, that could be overcome if necessary. Firstly, you may mix many processes in the same run, but the cross-section information stored in `info` only refers to the sum of them all, and therefore they are all classified as a common process 9999. Secondly, you should generate your events in the CM frame of the collision, since this is the assumed frame of stored Les Houches events, and no boosts have been implemented for the case that `Pythia::process` is not in this frame.

The LHEF standard is the agreed format to store the particles of a hard process, as input to generators, whereas output of final states is normally handled using the `HepMC` standard. It is possible to use LHEF also here, however. It requires that the above initialization is replaced by

```
LHAupFromPYTHIA8 myLHA(&pythia.event, &pythia.info);
```

i.e. that `process` is replaced by `event`. In addition, the `PartonLevel:all = off` command found in `main20.cc` obviously must be removed if one wants to obtain complete events.

```

#include "Pythia.h"
using namespace Pythia8;
int main() {

    // Generator.
    Pythia pythia;

    // Process selection. Minimal masses for gamma*/Z.
    pythia.readString("WeakBosonAndParton:qqbar2gmZg = on");
    pythia.readString("WeakBosonAndParton:qg2gmZq = on");
    pythia.readString("23:mMin = 50.");

    // Z'nin tum kanallarini kapat, sonra sadece muonlari ac (main06.cc'den)
    pythia.readString("23:onMode = off");
    pythia.readString("23:onIfAny = 13");

    // Create an LHAup object that can access relevant information in pythia.
    LHAupFromPYTHIA8 myLHA(&pythia.event, &pythia.info);

    // Open a file on which LHEF events should be stored, and write header.
    myLHA.openLHEF("z+jet.lhe");

    // LHC 8 TeV initialization.
    pythia.readString("Beams:eCM = 8000.");
    pythia.init();

    // Store initialization info in the LHAup object.
    myLHA.setInit();

    // Write out this initialization info on the file.
    myLHA.initLHEF();

    // Loop over events.
    for (int i = 0; i < 100; ++i) {

```



# Derle ve çalıştır

```
----- End PYTHIA Event Listing -----
-----
PYTHIA Error in StringFragmentation::fragment: stuck in joining
PYTHIA Error in Pythia::next: hadronLevel failed; try again
PYTHIA Warning in MultipartonInteractions::pTnext: weight above unity

*----- PYTHIA Event and Cross Section Statistics -----*
| Subprocess                Code |           Number of events           |           sigma +- delta           |
|                            |           Tried  Selected  Accepted |           (estimated) (mb)         | | |
|---|---|---|---|---|
| q qbar -> gamma*/Z0 g     241 |           2445      57      57 |           6.265e-07  3.707e-08 |
| q g-> gamma*/Z0 q         242 |           2309      43      43 |           5.513e-07  3.556e-08 |
| sum                       |           4754      100     100 |           1.178e-06  5.137e-08 |
|-----|-----|-----|-----|-----|
*----- End PYTHIA Event and Cross Section Statistics -----*

*----- PYTHIA Error and Warning Messages Statistics -----*
| times  message
|-----|-----
|      1  Error in Pythia::next: hadronLevel failed; try again
|      1  Error in StringFragmentation::fragment: stuck in joining
|      1  Warning in MultipartonInteractions::pTnext: weight above unity
|      3  Warning in ParticleDataEntry::initBlwmass: switching off width
|-----|-----
*----- End PYTHIA Error and Warning Messages Statistics -----*
[fizikci@hpfbu2012 examples]$
```

```
> g++ -O2 -ansi -pedantic -W -Wall -Wshadow -fbounds-check -I../include z+jet.cc -o bin/z+jet.exe -L../lib/archive -lpythia8 -llhapdfdummy
> bin/z+jet.exe
```

# Delphes3'ün Çalıştırılması

## \* Delphes3'ün kullanımı:

```
> cd ~/hepWork/Delphes*/
```

```
> ./DelphesLHEF --help
```

Usage: DelphesLHEF 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 LHEF format,

with no input\_file, or when input\_file is -, read standard input.

## \* Delphes'in LHE kütüklerini kabul eden programını çalıştıralım:

```
> ./DelphesLHEF examples/delphes_card_ATLAS.tcl z+jet.root ~/hepWork/  
pythia8/examples/z+jet.lhe
```



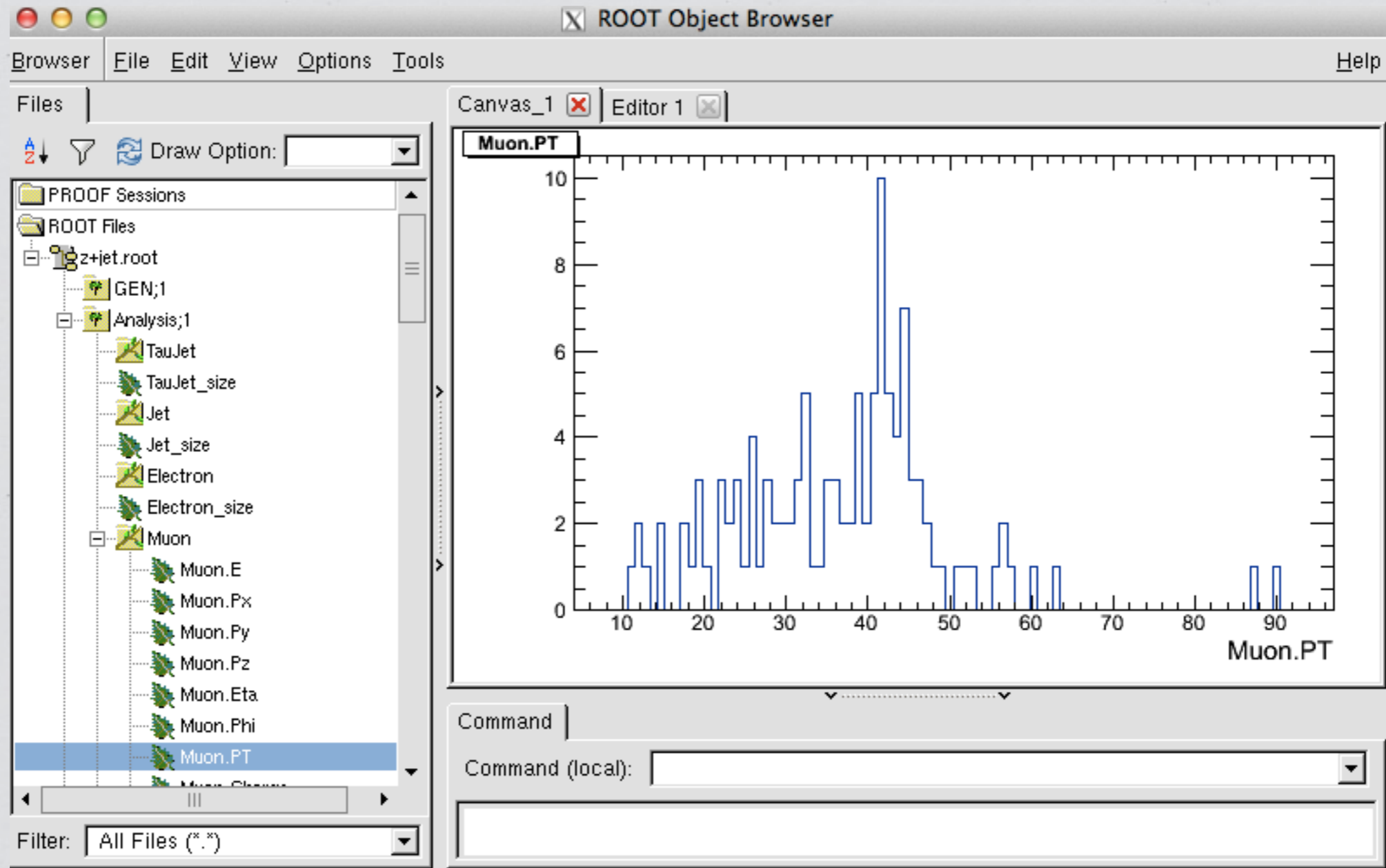
# Delphes Girdi Kartları

```
#####  
# Momentum resolution for muons  
#####  
  
module MomentumSmearing MuonMomentumSmearing {  
  set InputArray MuonTrackingEfficiency/muons  
  set OutputArray muons  
  
  # set ResolutionFormula {resolution formula as a function of eta and pt}  
  
  # resolution formula for muons  
  set ResolutionFormula {  
    (abs(eta) <= 1.5) * (pt > 0.1 && pt <= 1.0) * (0.03) + \  
    (abs(eta) <= 1.5) * (pt > 1.0 && pt <= 5.0e1) * (0.03) + \  
    (abs(eta) <= 1.5) * (pt > 5.0e1 && pt <= 1.0e2) * (0.04) + \  
    (abs(eta) <= 1.5) * (pt > 1.0e2) * (0.07) + \  
    (abs(eta) > 1.5 && abs(eta) <= 2.5) * (pt > 0.1 && pt <= 1.0) * (0.04) + \  
    (abs(eta) > 1.5 && abs(eta) <= 2.5) * (pt > 1.0 && pt <= 5.0e1) * (0.04) + \  
    (abs(eta) > 1.5 && abs(eta) <= 2.5) * (pt > 5.0e1 && pt <= 1.0e2) * (0.05) + \  
    (abs(eta) > 1.5 && abs(eta) <= 2.5) * (pt > 1.0e2) * (0.10)}  
}
```

```
#####  
# Jet finder  
#####  
  
module FastJetFinder FastJetFinder {  
  set InputArray Calorimeter/towers  
  
  set OutputArray jets  
  
  # algorithm: 1 CDFJetClu, 2 MidPoint, 3 SIScone, 4 kt, 5 Camb  
  set JetAlgorithm 6  
  set ParameterR 0.6  
  
  set JetPTMin 20.0  
}
```

\* TCL ile yazılmış. Oldukça sade, okunması kolay.

# Delphes ROOT kütüğü





# LHCO Biçimi

\* PGS'in önce LHC Olimpiyatları için hazırlanmış, düz ASCII dizgesindeki, neredeyse standart olan çıktı biçimi.

#	tür	eta	phi	pt	jkt1	#iz	btag	had/em	dum1	dum2
0		39	1043							
1	0	-1.350	3.341	26.11	0.00	0.0	0.0	0.00	0.0	0.0
2	1	-0.663	5.233	164.40	0.00	1.0	0.0	0.02	0.0	0.0
3	2	-0.589	4.675	147.62	0.11	-1.0	4.0	95.99	0.0	0.0
4	4	-0.629	4.998	308.94	85.25	12.0	2.0	0.33	0.0	0.0
5	4	-2.061	1.571	455.01	156.56	32.0	0.0	1.19	0.0	0.0
6	4	-1.954	2.699	24.07	2.38	32.0	0.0	11.60	0.0	0.0
7	4	1.149	5.756	6.23	1.93	2.0	0.0	3.73	0.0	0.0
8	6	0.000	3.521	28.86	0.00	0.0	0.0	0.00	0.0	0.0

# LHCO Biçimi

\* PGS'in önce LHC Olimpiyatları için hazırlanmış, düz A dizgesindeki, neredeyse standart olan çıktı biçimi.

39 numaralı olay;  
tetikleyici sözcüğü  
= 1043

#	tür	eta	phi	pt	jkt1	#iz	btag	had/em	dum1	dum2
0		39	1043							
1	0	-1.350	3.341	26.11	0.00	0.0	0.0	0.00	0.0	0.0
2	1	-0.663	5.233	164.40	0.00	1.0	0.0	0.02	0.0	0.0
3	2	-0.589	4.675	147.62	0.11	-1.0	4.0	95.99	0.0	0.0
4	4	-0.629	4.998	308.94	85.25	12.0	2.0	0.33	0.0	0.0
5	4	-2.061	1.571	455.01	156.56	32.0	0.0	1.19	0.0	0.0
6	4	-1.954	2.699	24.07	2.38	32.0	0.0	11.60	0.0	0.0
7	4	1.149	5.756	6.23	1.93	2.0	0.0	3.73	0.0	0.0
8	6	0.000	3.521	28.86	0.00	0.0	0.0	0.00	0.0	0.0



# LHCO Biçimi

\* PGS'in önce LHC Olimpiyatları için hazırlanmış, düz ASCII dizgesindeki, neredeyse standart olan çıktı biçimi.

0 = foton  
26.11 GeV  
 $\eta = -1.350$   
 $\varphi = 3.341$

#	tür	eta	phi	pt	jkt1	#iz	btag	had/em	dum1	dum2
0		39	1043							
1	0	-1.350	3.341	26.11	0.00	0.0	0.0	0.00	0.0	0.0
2	1	-0.663	5.233	164.40	0.00	1.0	0.0	0.02	0.0	0.0
3	2	-0.589	4.675	147.62	0.11	-1.0	4.0	95.99	0.0	0.0
4	4	-0.629	4.998	308.94	85.25	12.0	2.0	0.33	0.0	0.0
5	4	-2.061	1.571	455.01	156.56	32.0	0.0	1.19	0.0	0.0
6	4	-1.954	2.699	24.07	2.38	32.0	0.0	11.60	0.0	0.0
7	4	1.149	5.756	6.23	1.93	2.0	0.0	3.73	0.0	0.0
8	6	0.000	3.521	28.86	0.00	0.0	0.0	0.00	0.0	0.0

# LHCO Biçimi

\* PGS'in önce LHC Olimpiyatları için hazırlanmış, düz ASCII dizgesindeki, neredeyse standart olan çıktı biçimi.

tür = 1  
#iz = 1  
pozitron  
164.4 GeV

#	tür	eta	phi	pt	jkt1	#iz	btag	had/em	dum1	dum2
0		39	1043							
1	0	-1.350	3.341	26.11	0.00	0.0	0.0	0.00	0.0	0.0
2	1	-0.663	5.233	164.40	0.00	1.0	0.0	0.02	0.0	0.0
3	2	-0.589	4.675	147.62	0.11	-1.0	4.0	95.99	0.0	0.0
4	4	-0.629	4.998	308.94	85.25	12.0	2.0	0.33	0.0	0.0
5	4	-2.061	1.571	455.01	156.56	32.0	0.0	1.19	0.0	0.0
6	4	-1.954	2.699	24.07	2.38	32.0	0.0	11.60	0.0	0.0
7	4	1.149	5.756	6.23	1.93	2.0	0.0	3.73	0.0	0.0
8	6	0.000	3.521	28.86	0.00	0.0	0.0	0.00	0.0	0.0



# LHCO Biçimi

\* PGS'in önce LHC Olimpiyatları için hazırlanmış, düz ASCII dizgesindeki, neredeyse standart olan çıktı biçimi.

#	tür	eta	phi	pt	jkt1	#iz	btag	had/em	dum1	du
0		39	1043							
1	0	-1.350	3.341	26.11	0.00	0.0	0.0	0.00	0.0	0.0
2	1	-0.663	5.233	164.40	0.00	1.0	0.0	0.02	0.0	0.0
3	2	-0.589	4.675	147.62	0.11	-1.0	4.0	95.99	0.0	0.0
4	4	-0.629	4.998	308.94	85.25	12.0	2.0	0.33	0.0	0.0
5	4	-2.061	1.571	455.01	156.56	32.0	0.0	1.19	0.0	0.0
6	4	-1.954	2.699	24.07	2.38	32.0	0.0	11.60	0.0	0.0
7	4	1.149	5.756	6.23	1.93	2.0	0.0	3.73	0.0	0.0
8	6	0.000	3.521	28.86	0.00	0.0	0.0	0.00	0.0	0.0

$\mu^{-4}$  numaralı  
jete yakın;  
çevresinde 95  
GeV'lik enerji

# LHCO Biçimi

\* PGS'in önce LHC Olimpiyatları için hazırlanmış, düz ASCII dizgesindeki, neredeyse standart olan çıktı biçimi.

#	tür	eta	phi	pt	jkt1	#iz	btag	had/em	dum1	du
0		39	1043							
1	0	-1.350	3.341	26.11	0.00	0.0	0.0	0.00	0.0	0.0
2	1	-0.663	5.233	164.40	0.00	1.0	0.0	0.02	0.0	0.0
3	2	-0.589	4.675	147.62	0.11	-1.0	4.0	95.99	0.0	0.0
4	4	-0.629	4.998	308.94	85.25	12.0	2.0	0.33	0.0	0.0
5	4	-2.061	1.571	455.01	156.56	32.0	0.0	1.19	0.0	0.0
6	4	-1.954	2.699	24.07	2.38	32.0	0.0	11.60	0.0	0.0
7	4	1.149	5.756	6.23	1.93	2.0	0.0	3.73	0.0	0.0
8	6	0.000	3.521	28.86	0.00	0.0	0.0	0.00	0.0	0.0

jet  
b-işaretlenmiş  
m = 85.25 GeV  
"içinde" 12 iz



# LHCO Biçimi

\* PGS'in önce LHC Olimpiyatları için hazırlanmış, düz ASCII dizgesindeki, neredeyse standart olan çıktı biçimi.

#	tür	eta	phi	pt	jkt1	#iz	btag	had/em	dum1	dum2
0		39	1043							
1	0	-1.350	3.341	26.11	0.00	0.0	0.0	0.00	0.0	0.0
2	1	-0.663	5.233	164.40	0.00	1.0	0.0	0.02	0.0	0.0
3	2	-0.589	4.675	147.62	0.11	-1.0	4.0	95.99	0.0	0.0
4	4	-0.629	4.998	308.94	85.25	12.0	2.0	0.33	0.0	0.0
5	4	-2.061	1.571	455.01	156.56	32.0	0.0	1.19	0.0	0.0
6	4	-1.954	2.699	24.07	2.38	32.0	0.0	11.60	0.0	0.0
7	4	1.149	5.756	6.23	1.93	2.0	0.0	3.73	0.0	0.0
8	6	0.000	3.521	28.86	0.00	0.0	0.0	0.00	0.0	0.0

3 jet.  
 $e^{\text{Hadronik}} > e^{\text{EM}}$

# LHCO Biçimi

\* PGS'in önce LHC Olimpiyatları için hazırlanmış, düz ASCII dizgesindeki, neredeyse standart olan çıktı biçimi.

#	tür	eta	phi	pt	jkt1	#iz	btag	had/em	dum1	dum2
0		39	1043							
1	0	-1.350	3.341	26.11	0.00	0.0	0.0	0.00	0.0	0.0
2	1	-0.663	5.233	164.40	0.00	1.0	0.0	0.02	0.0	0.0
3	2	-0.589	4.675	147.62	0.11	-1.0	4.0	95.99	0.0	0.0
4	4	-0.629	4.998	308.94	85.25	12.0	2.0	0.33	0.0	0.0
5	4	-2.061	1.571	455.01	156.56	32.0	0.0	1.19	0.0	0.0
6	4	-1.954	2.699	24.07	2.38	32.0	0.0	11.60	0.0	0.0
7	4	1.149	5.756	6.23	1.93	2.0	0.0	3.73	0.0	0.0
8	6	0.000	3.521	28.86	0.00	0.0	0.0	0.00	0.0	0.0

kayıp dikey  
enerji  
 $p_x = -27$  GeV  
 $p_y = -11$  GeV



# LHCO Biçimi

\* PGS'in önce LHC Olimpiyatları için hazırlanmış, düz ASCII dizgesindeki, neredeyse standart olan çıktı biçimi.

#	tür	eta	phi	pt	jkt1	#iz	btag	had/em	dum1	dum2
0		39	1043							
1	0	-1.350	3.341	26.11	0.00	0.0	0.0	0.00	0.0	0.0
2	1	-0.663	5.233	164.40	0.00	1.0	0.0	0.02	0.0	0.0
3	2	-0.589	4.675	147.62	0.11	-1.0	4.0	95.99	0.0	0.0
4	4	-0.629	4.998	308.94	85.25	12.0	2.0	0.33	0.0	0.0
5	4	-2.061	1.571	455.01	156.56	32.0	0.0	1.19	0.0	0.0
6	4	-1.954	2.699	24.07	2.38	32.0	0.0	11.60	0.0	0.0
7	4	1.149	5.756	6.23	1.93	2.0	0.0	3.73	0.0	0.0
8	6	0.000	3.521	28.86	0.00	0.0	0.0	0.00	0.0	0.0

[http://vl.jthaler.net/olympicswiki/doku.php?id=lhc\\_olympics:data\\_file\\_format](http://vl.jthaler.net/olympicswiki/doku.php?id=lhc_olympics:data_file_format)

# LHCO'ya Çevrim

\* Delphes ROOT kütüğünü LHCO'ya çevirebiliriz:

```
> ./root2lhco z+jet.root > z+jets.lhco
```

```
** Reading z+jet.root
```

```
** Input file contains 100 events
```

```
** [#####](100.00%)
```

```
** Exiting...
```

```
[fizikci@hpfbu Delphes-3.0.10]$ tail -15 z+jet.lhco
```

3	4	-2.452	0.383	24.37	8.47	4.0	0.0	3.02	0.0	0.0
4	6	0.000	-2.656	7.64	0.00	0.0	0.0	0.00	0.0	0.0
0		98	0							
1	2	1.423	1.385	41.67	0.11	1.0	0.0	1.01	0.0	0.0
2	6	0.000	1.168	13.51	0.00	0.0	0.0	0.00	0.0	0.0
0		99	0							
1	2	-1.812	2.263	44.45	0.11	1.0	2.0	1.07	0.0	0.0
2	4	-1.977	0.088	26.42	8.39	8.0	0.0	3.41	0.0	0.0
3	6	0.000	-1.133	52.36	0.00	0.0	0.0	0.00	0.0	0.0
0		100	0							
1	2	1.077	-2.849	83.45	0.11	-1.0	4.0	0.00	0.0	0.0
2	2	-0.128	0.838	17.46	0.11	1.0	3.0	0.01	0.0	0.0
3	4	-1.302	0.057	41.74	8.48	7.0	0.0	1.37	0.0	0.0
4	4	2.218	0.344	23.78	6.40	4.0	0.0	2.94	0.0	0.0
5	6	0.000	2.847	6.47	0.00	0.0	0.0	0.00	0.0	0.0

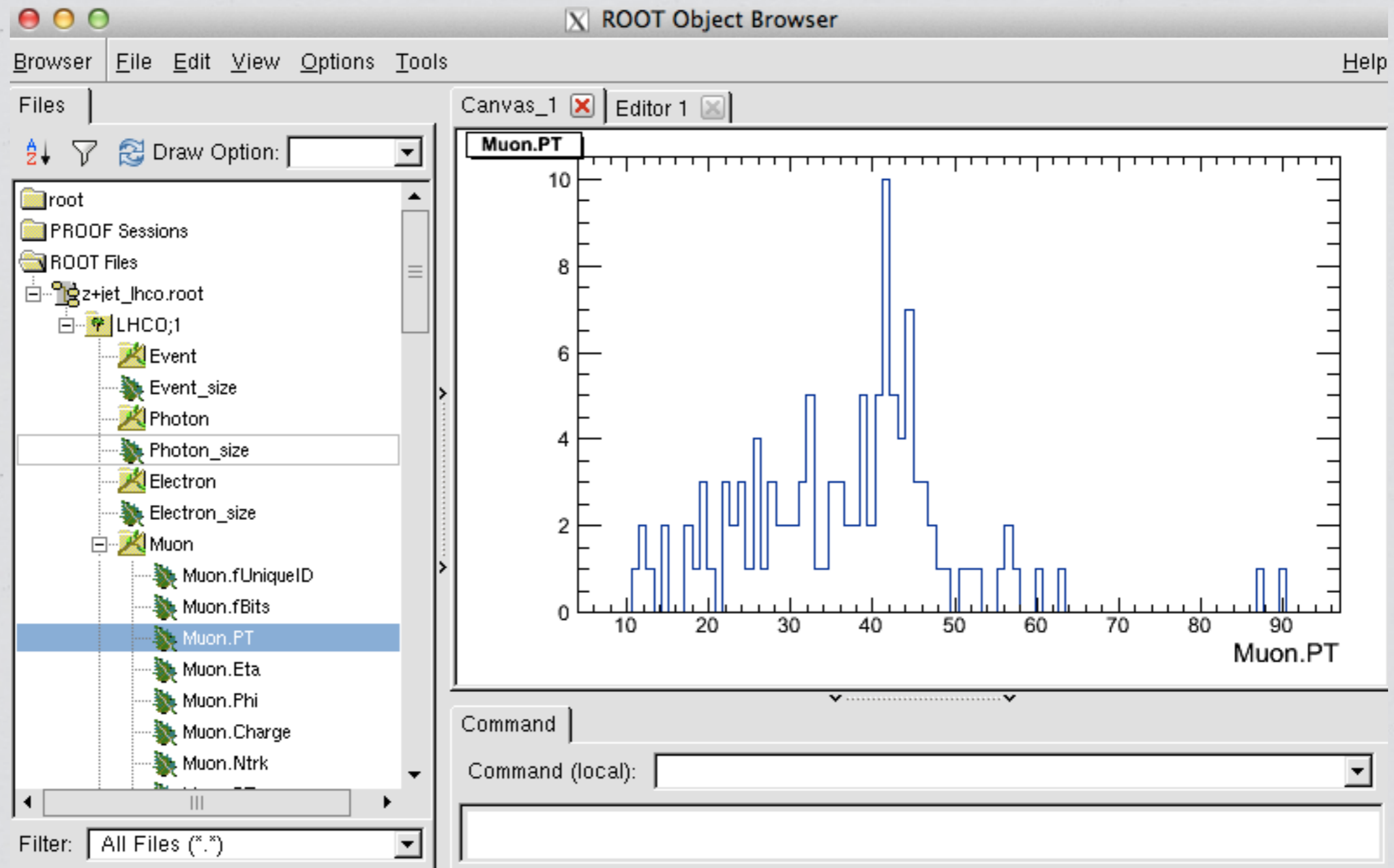
```
[fizikci@hpfbu Delphes-3.0.10]$
```

\* ExROOTAnalysis paketiyle bu LHCO kütüğünü de ROOT kütüğü yapabiliriz:

```
> ~/hepWork/MadGraph5_*/ExRootAnalysis/ExRootLHCOlympicsConverter  
z+jet.lhco z+jet_lhco.root
```



# LHCO ROOT kütüğü



# Algıcımızın Çözünürlüğü

- \* Ürettiğimiz ROOT kütüğünü kullanarak, algıcımızın jet enerji çözünürlüğünü çıkaran bir çözümleme programı yazınız.

```
TLorentzVector mu1,mu2,jet,Z;
TH1F *hMomCoz = new TH1F("hMomCoz", "Dik Momentum Farki",100,-200,200);
TH1F *hDelPhi = new TH1F("hDelPhi", "Phi Farki",100,-4,4);

Long64_t nbytes = 0, nb = 0;
for (Long64_t jentry=0; jentry<nentries;jentry++) {
    Long64_t ientry = LoadTree(jentry);
    if (ientry < 0) break;
    nb = fChain->GetEntry(jentry);    nbytes += nb;
    // if (Cut(ientry) < 0) continue;

    if ( Muon_size < 2 ) continue;

    mu1.SetPtEtaPhiM( Muon_PT[0], Muon_Eta[0], Muon_Phi[1], 0.1057 );
    mu2.SetPtEtaPhiM( Muon_PT[0], Muon_Eta[0], Muon_Phi[1], 0.1057 );
    jet.SetPtEtaPhiM( Jet_PT[0], Jet_Eta[0], Jet_Phi[0], Jet_Mass[0] );

    Z = mu1 + mu2;
    hDelPhi->Fill( jet.Phi() - Z.Phi() );

    // buraya deltaPhi ile ilgili bir kod koyabilir miyiz?
    // veya tam karsidaki jet'i mi aramaliyiz?

    hMomCoz->Fill( jet.Pt() - Z.Pt() );
}

hMomCoz->Draw();
```



# Müon Verimi

- \* Ürettiğimiz ROOT kütüğünü kullanarak, algıcımızın müon verimini işaretle-ve-yakala yöntemiyle çıkaran bir çözümleme programı yazınız.

# Sefa

- \* Bulmaca: LHCO biçimini tanımladığımız sayfadaki örnek olay ne olabilir?
- \*  $z$ +jet çözümlemesini yapınız (beraber ürettiğimiz kütükleri kullanarak).
- \* Kendi hızlı MC benzetiminiz (ikisinden birini seçebilirsiniz):
  - \* PGS'de bir hadronun ve müonun kalorimetrelerde bıraktığı enerjinin fortran kodunu gördük. Bu kodları alıp birer ROOT betiğine dönüştürün. Rastgele hadron ve müonlar kullanarak, kaybedilen enerjinin parçacığın momentumuna göre nasıl değiştiğini gösteren 2 boyutlu histogramlar çizdirin. Eklerdeki grafiklerle sonuçlarınızı karşılaştırın.
  - \* Alternatif ödev: Opera müon spektrometresinin momentum çözünürlüğü ödevini hatırlayın. Bununla ilgili bir hızlı simülasyon programı yazabilir miyiz?



---

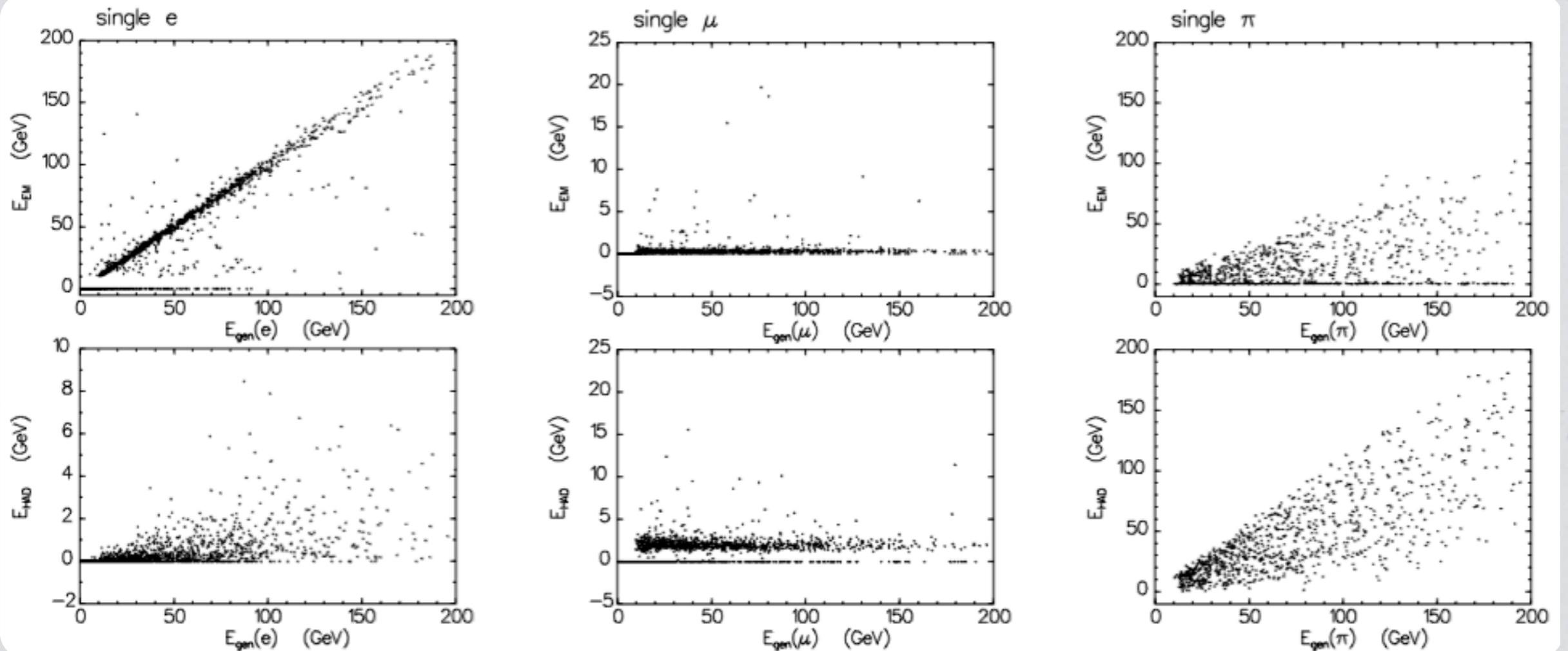
---

# FAZLADAN LEBLEBİ

---

---

# PGS'deki $e$ , $\pi$ ve $\mu$ Enerji Modelleri



[http://online.itp.ucsb.edu/online/lhco\\_c06/conway/](http://online.itp.ucsb.edu/online/lhco_c06/conway/)



# StdHep biçimindeki olay kütüklerini PGS veya Delphes2 ile kullanmak

- \* Bunun en kolay yolu, hali hazırda Madgraph ile sunulan ve onunla kullandığınız pythia-pgs ve Delphes paketlerini kullanmak.
- \* Madgraph'i yüklediğiniz dizine geçin.
- \* Template dizininden yeni bir kopya çıkarın ve oraya geçin: `cp -r Template Leblebi ; cd Leblebi`
- \* PGS/Delphes parametre kartınızı hazırlayın. (Tercih ettiğiniz metin editörü ile Cards/pgs\_card.dat veya Cards/delphes\_card.dat kütüğünü düzeltin.)
- \* Events dizinine geçin: `cd Events`
- \* Stdhep biçimdeki olay kütüğünüzü bulduğunuz dizinine pythia\_events.hep adıyla kopyalayın: `cp kütüğünüzNeredeyseODizin/olayKütüğünüz pythia_events.hep`
- \* PGS veya Delphes'i çalıştırın: `../bin/run_pgs` veya `../bin/run_delphes`
- \* Root kütüğü oluşturmak için (ExRootAnalysis paketini de yüklemişseniz):  
`../.. /ExRootAnalysis/ExRootLHColympicsConverter  
pgs_events.lhco pgs_events.root`

# Delphes2

\* Delphes2'in kullanımını (**eski**, sanal makinada **yok**):

```
> cd <DelphesKlasörü>
```

```
> ./Delphes --help
```

Usage: ./Delphes input\_file output\_file [detector\_card] [trigger\_card]

input\_list - list of files in Ntpl, StdHep, HepMC or 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)

\* Kendi olay kütüğümüzü içeren bir girdi listesi hazırlayalım ve Delphes'i çalıştıralım:

```
> echo ~/hepWork/pythia8/examples/z+jet.lhe > liste.txt
```

```
> ./Delphes liste.txt z+jet.root data/DetectorCard_ATLAS.dat data/TriggerCard_ATLAS.dat
```