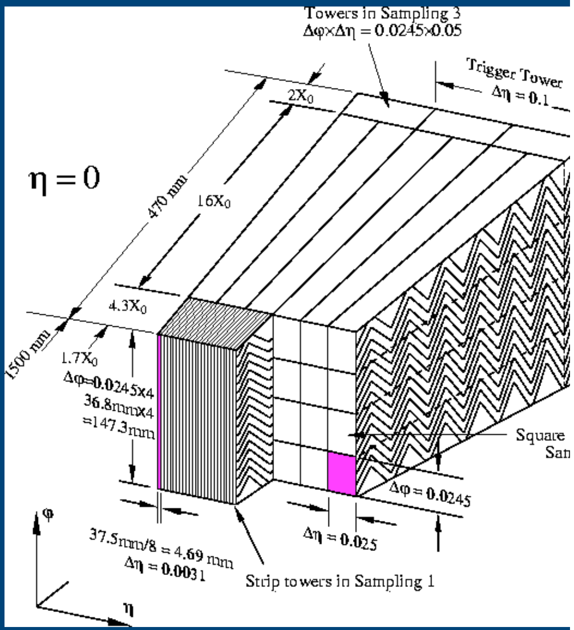
Electrons in the AOD

Nicolas Kerschen (Sheffield)

- Outline:
  - The ATLAS EM calorimeter
    - Showers/cells/clusters & calibration
      (see my talk from yesterday)
  - Electrons in real Life
    - Identification of electrons in our detector.
    - What observables to use?
  - Electrons in the AOD
    - What variables to use for identification
    - How to access cluster/track information or other information one might need for analysis

Electron identification in real Life

- How to identify electrons (in very simple words):
  - shower in the EM calorimeter
  - energy deposited in cells/compartments of the calorimeter
  - Cells/compartments form clusters
  - position/energy of cluster gives information about position and energy of shower
  - check if cluster is matching with a track (separate electrons and photons).
  - do some cuts on calorimeter/track observables to reject jets.



- Discriminating variables from the calorimeters
  - Hadronic leakage
    - Electromagnetic showers deposit a small amount of energy in the hadronic calorimeter, typically less t 2% for electrons.
  - Use of the second compartment of the EM calorimeter
    - Electromagnetic showers deposit most of their energy in the second sampling of the electromagnetic calorimeter. Variables used:
      - The lateral shower shape, $R\eta(37)$, is given by the ratio of the energy reconstructed in a 3× 7 clu the energy in a 7× 7 cluster. This variable shows a peak near one for electrons because of the ve small lateral leakage; large tails at lower values of $R\eta(37)$ for the jets are expected.
      - The lateral width is calculated with a window of 3× 5 cells using the energy weighted sum over a cells, which depends on the particle impact point inside the cell

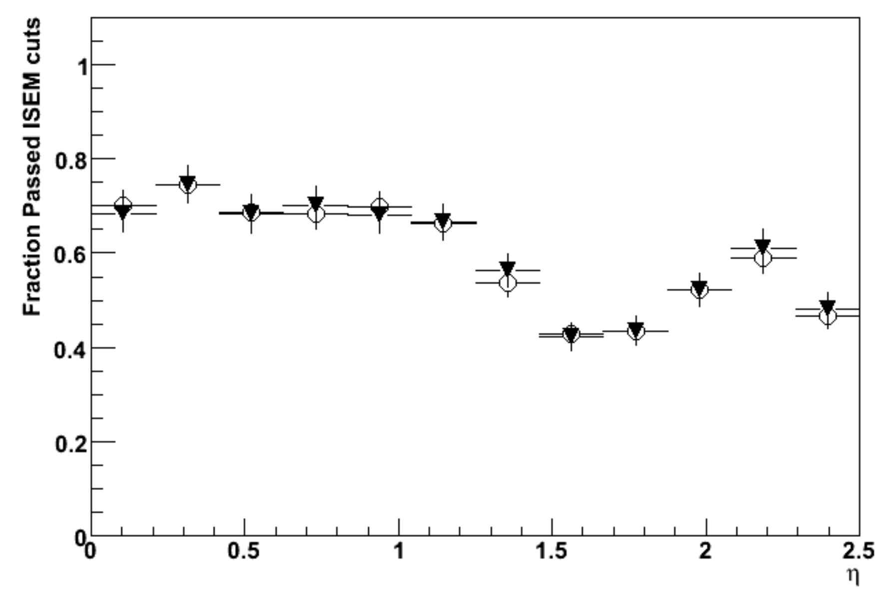
- Discriminating variables from the calorimeters (2)
  - Use of the first compartment of the EM calorimeter
    - The first compartment with its very fine granularity in rapidity can be used to dete substructures within a shower and thus isolated π_0 discriminated against efficientl Variables used:
      - $\Delta E = E_{max2} - E_{min}$
      - $\Delta E_{max2} = E_{max2}/(1+9(5)\times 10\text{-}3\ E_T)$, with E_T the transverse energy of the clust
      - $\omega tot1\sim=\sim\surd(\Sigma\ E_i \times (\ i-i_{max})2\ /\ \Sigma\ E_i)$, where i is the strip number and i_{max} the strip number of the first local maximum.
      - $Fside = [\ E(\pm 3) - E(\pm 1)]/E(\pm 1)$, where $E(\pm n)$ is the energy in ± n strips around the strip with highest energy.
      - $\omega 3\ strips = \surd\{\Sigma\ E_i \times (\ i-i_{max})2\ /\ \Sigma\ E_i\}$, where i is the number of the strip and the strip number of the most energetic one.

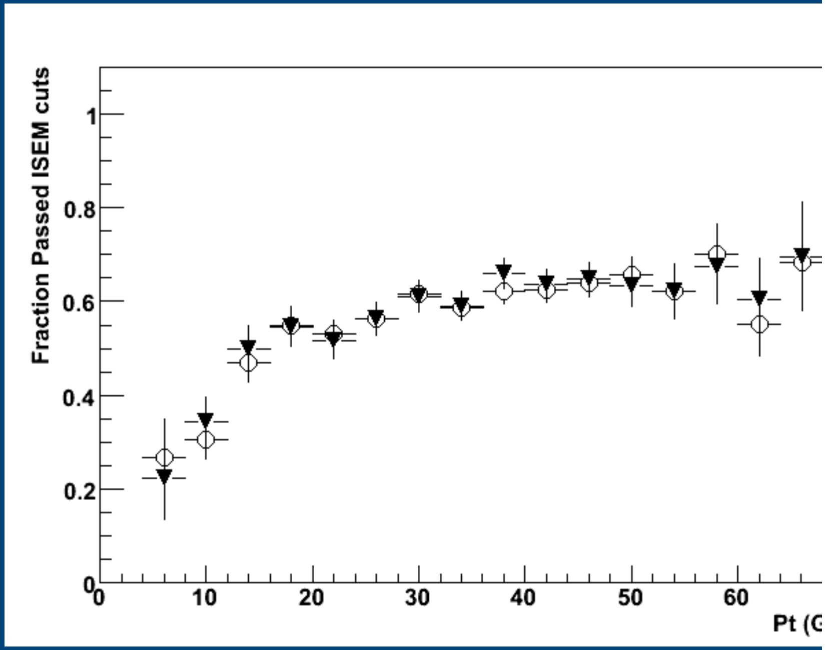
- Discriminating variables from the inner detector
  - Track quality cuts
    - at least nine precision hits (Pixel+SCT)
    - at least two hits in the pixels, one of which being in the b-layer
    - a transverse impact parameter A0 < 0.1 cm
  - Inner detector/calorimeter spatial matching information
    - $\Delta\eta = |\eta_{strips} - \eta_{ID}|$, where η_{strips} is computed in the first sampling of the electromagnetic calorimeter, where the granularity is very fine, and η_{ID} is the pseudo-rapidity of the track extrapolated to the calorimeter.
    - $\Delta\varphi = |\varphi_{middle} - \varphi_{ID}|$, where φ_{middle} is computed in the second compartment o the electromagnetic calorimeter and φ_{ID} is the azimuth of the track extrapolated to the calorimeter.

- Discriminating variables from the inner detector (2)
    - Inner Detector/calorimeter energy matching information
        - The energy E measured in the electromagnetic calorimeter is compared to the momentum p measured in the Inner Detector. In the case of an electron, the momentum should match the energy.
    - Transition radiation in the TRT information
        - A further reduction of the charged hadron contamination is obtained by rejecting tracks having a low fraction of high-threshold hits. The discriminating variable use is the ratio N_{high}/N_{all} between the number of high threshold hits and the total number of TRT hits.

Identification efficiences

- IsEM tight





Electrons in the AOD

- The electron collection in the AOD, the keys to use are:
  - For the collection: ElectronAODCollection
  - For the details (Shower/Trackmatch): egDetailAOD

- ElectronAODcollection
  - If a cluster has a matched track, the egamma object is added to the electron collection

- IsEM flag
  - The candidate has to pass a series of cuts based on the shower shape properties in different compartments of the calorimeter as well as variables combining ID and Calo informations (see previous slides). If a cut is not passed, then a bit is set in the IsEM flag
  - You can use the bit mask to use only certain parts of the ID criteria

- Predefined set of cuts:
  - Loose cuts: electron->isem(ElectronLoose)==0
    - Calorimeter cuts only
  - Medium cuts: electron->isem(ElectronMedium)==0
    - Calorimeter cuts & Track quality cuts (no b-layer)
  - Tight cuts: electron->isem(ElectronTight)==0
    - All cuts (previous & Track matching & TRT cuts)
  - All these cuts can be turned on or off individually using bit masks

```
enum BitDef {
   // Cluster based egamma
   ClusterEtaRange       =  0,
   ClusterHadronicLeakage =  1,
   ClusterMiddleEnergy    =  4,
   ClusterMiddleEratio37  =  5,
   ClusterMiddleEratio33  =  6,
   ClusterMiddleWidth     =  7,
   ClusterStripsEratio    =  8,
   ClusterStripsDeltaEmax2=  9,
   ClusterStripsDeltaE    = 10,
   ClusterStripsWtot      = 11,
   ClusterStripsFracm     = 12,
   ClusterStripsWeta1c    = 13,
   ClusterIsolation       = 14,
   //Track based egamma
   TrackBlayer           = 16,
   TrackPixel            = 17,
   TrackSi               = 18,
   TrackA0               = 19,
   TrackMatchEta         = 20,
   TrackMatchPhi         = 21,
   TrackMatchEoverP      = 22,
   TrackTRThits          = 24,
   TrackTRTratio         = 25,
   TrackTRTratio90       = 26
  };
```

How to use the IsEM flag?

```
StoreGateSvc* m_storeGate;
const ElectronContainer* electronColl;
m_storeGate->retrieve(electronColl, "ElectronAODCollection");

ElectronContainer::const_iterator eleItr = electronColl->begin();
ElectronContainer::const_iterator eleEnd = electronColl->end();

for(; eleItr != eleEnd; ++eleItr) {
int isemLoose = (*eleItr)->isem(egammaPID::ElectronLoose);
int isemMedium = (*eleItr)->isem(egammaPID::ElectronMedium);
int isemTight = (*eleItr)->isem(egammaPID::ElectronTight);
if (isemLoose == 0) std::cout  <<   looooooose cut passed  "  << std::endl;
}
```

If you want to apply your own set of cuts you can use the bit masks like this:
For example, let’s say you want to use only the E/p cut and the Hadronic Leakage
(of course you wouldn’t wanna do that)

The bit definition on the previous slide tells us:
E/p cut is bit number 22
Hadronic Leakage is bit number 1
So you do:
`(*electron)[index]-> isem(0x400002);`

WARNING: Bit definition is different in 12.0.X
Closest to loose definition: (isem & 0x7)==0
Closest to medium: (isem & 0x3FF)==0
Closest to tight: isem = 0

- All calorimeter observables used for identification are accessible in the AOD:
  - Shower variables
    - egammaParameters::etcone20 : isolation energy (transverse) in a cone with half-opening angle 0.2
    - egammaParameters::ethad1 : transverse energy in the first sampling of the hadronic calorimeters behind the clu
    - egammaParameters::e233 : uncalibrated energy (sum of cells) of the middle sampling in a rectangle of size 3x
      egammaParameters::e237 : uncalibrated energy (sum of cells) of the middle sampling in a rectangle of size 3x
      egammaParameters::e277 : uncalibrated energy (sum of cells) of the middle sampling in a rectangle of size 7x
      egammaParameters::weta1
    - egammaParameters::weta2
    - egammaParameters::f1
    - egammaParameters::e2tsts1
    - egammaParameters::emins1
    - egammaParameters::wtots1
    - egammaParameters::fracs1
    - egammaParameters::f1core
    - egammaParameters::f3core
    - egammaParameters::pos7
    - egammaParameters::iso
    - egammaParameters::widths2
    - egammaParameters::zvertex
    - egammaParameters::errz
    - egammaParameters::etap : pointing eta reconstructed from the cluster (first and second sampling)
    - egammaParameters::depth

- EMTrackMatch variables
  - egammaParameters::EtaCorrMag
  - egammaParameters::EoverP : ratio of the cluster energy and the track momentum
  - egammaParameters::deltaEta1 : difference between the cluster eta (first sampling) and the eta of the track extrapolated to the first sampling
  - egammaParameters::deltaEta2 :difference between the cluster eta (second sampling) and the eta of the track extrapolated to the second sampling
  - egammaParameters::deltaPhi2 : difference between the cluster phi (second sampling) and the phi of the track extrapolated to the second sampling

- How to access EMShower and EMTrackMatch information

```
const EMShower *shower;
shower = (*eleItr)->detail<EMShower>("egDetailAOD")
float etcone = shower->parameter(egammaParameters::etcone20)

const EMTrackMatch *trackmatch;
trackmatch = (*eleItr)->detail<EMTrackMatch>("egDetailAOD")
float etcone = shower->parameter(egammaParameters ::EoverP )
```

Clusters and Tracks

pointer to CaloCluster : const CaloCluster * egamma::cluster() const

pointer to TrackParticle : const Rec::TrackParticle * egamma::trackParticle() const

```
(*eleItr)->trackParticle()->phi()
(*eleItr)->trackParticle()->eta()
(*eleItr)->trackParticle()->measuredPerigee()->parameters()[Trk::d0]
(*eleItr)->trackParticle()->measuredPerigee()->localErrorMatrix().error(Trk::d0)
(*eleItr)->trackParticle()->trackSummary()->get(Trk::numberOfBLayerHits)
(*eleItr)->trackParticle()->trackSummary()->get(Trk::numberOfPixelHits)

(*eleItr)->cluster() ->phi()
(*eleItr)->cluster() ->eta()
(*eleItr)->cluster() -> energy()
(*eleItr)->cluster() -> energyBE(0)
(*eleItr)->cluster() -> energyBE(1)
(*eleItr)->cluster() -> energyBE(2)
(*eleItr)->cluster() -> energyBE(3)
```