PGS

John Conway
University of California, Davis

Fast Simulator Workshop
CERN, June 11, 2012

Contributors

PGS is the work of many people!

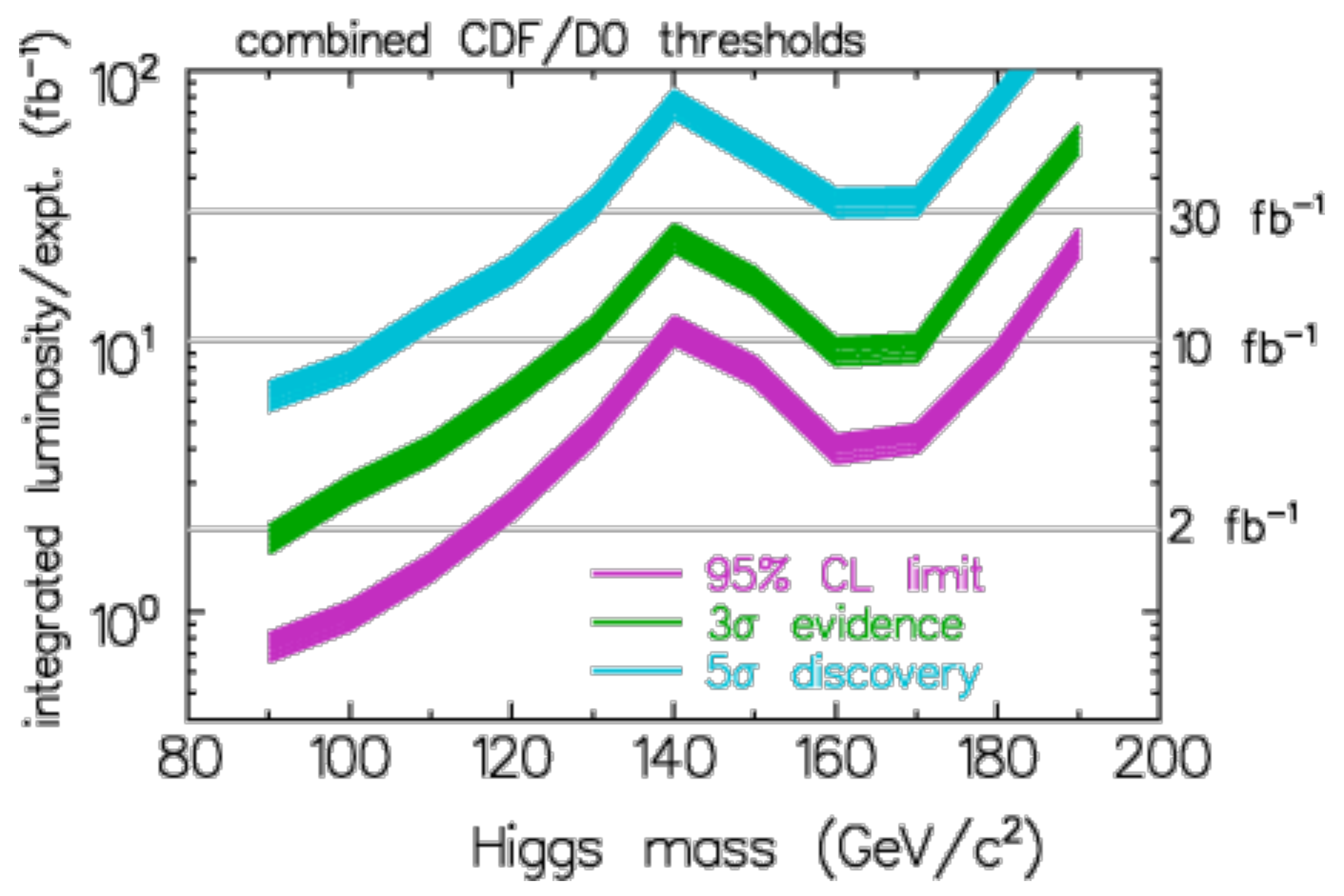
John Conway (UC Davis), Ray Culbertson (FNAL), Regina Demina (U. Rochester), Ben Kilminster (Ohio State), Mark Kruse (Duke), Steve Mrenna (FNAL), Jason Nielsen (LBNL), Aaron Pierce and Jesse Thaler (Harvard), Natalia Toro (Harvard), Chris Tully (Princeton), Jay Wacker (SLAC), Anson Hook (SLAC)

Special thanks to Matt Strassler, Matt Bowen, Nima Arkani-Hamed and Liantao Wang for furthering the use and development of PGS.

Origin of PGS

- March 1998: kickoff of the Tevatron Run 2 SUSY/Higgs Workshop
- no Run 2 CDF/D0 simulations available then
- developed "SHW" simulation as average of CDF/D0
- published SHW Higgs report: hep-ph/0010338 →
- fairly accurate for Tevatron Higgs reach!
- SHW -> PGS for Snowmass 2001
- has been used for VLHC, LHC, LC, Tevatron comparisons, especially by theorists

Tevatron SM Higgs: SHW



Famous result from the 1998 Tevatron Run 2 Susy/Higgs Workshop: from SHW simulation!

Origin of PGS

- March 1998: kickoff of the Tevatron Run 2 SUSY/Higgs Workshop
- no Run 2 CDF/D0 simulations available then
- developed "SHW" simulation as average of CDF/D0
- published SHW Higgs report: hep-ph/0010338
- fairly accurate for Tevatron Higgs reach!
- SHW -> PGS for Snowmass 2001
- has been used for VLHC, LHC, LC, Tevatron comparisons, especially by theorists

PGS Design Goals

- interface to standard physics process generators (PYTHIA, HERWIG, ISAJET, ALPGEN, ...)
- perform very basic detector simulation with
    - tracks
    - calorimeter deposits
    - muon ID
- reconstruct physics “objects”: γ, e, μ, τ, jet (b), MET from tracks and calorimeter clusters
- parametrize where needed
- BE FAST!

PGS Simulation Features

	PGS?
• detector acceptance	yes
• detector efficiency	yes
• detector resolution	yes
• secondary interactions	
- nuclear interactions	no
- brehmsstrahlung	no
- pair production	no
- multiple scattering	no
• multiple interactions (pileup)	no
• event reconstruction effects	yes

Flow of PGS

event generation (PYTHIA, HERWIG, ...)

STDHEP common blocks

event simulation, object reconstruction

user analysis

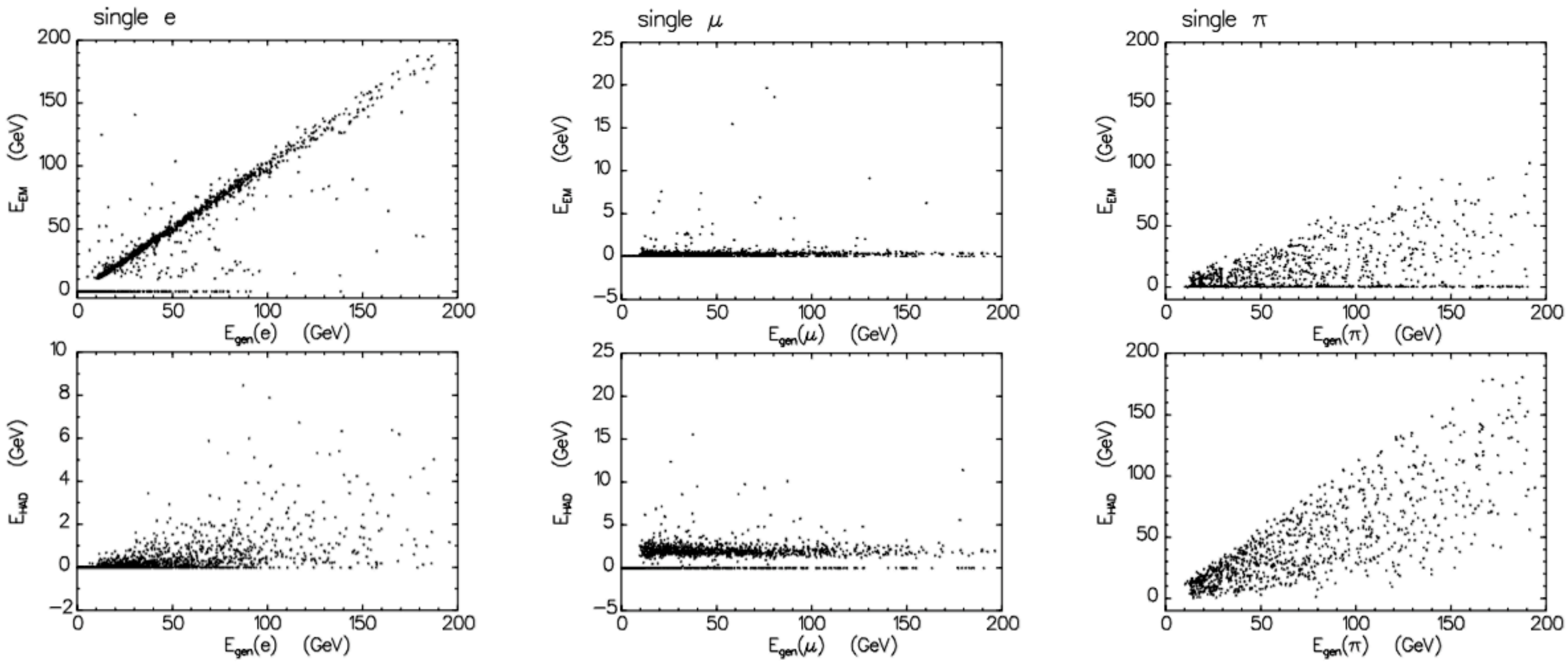
user output

PGS Detector Simulation

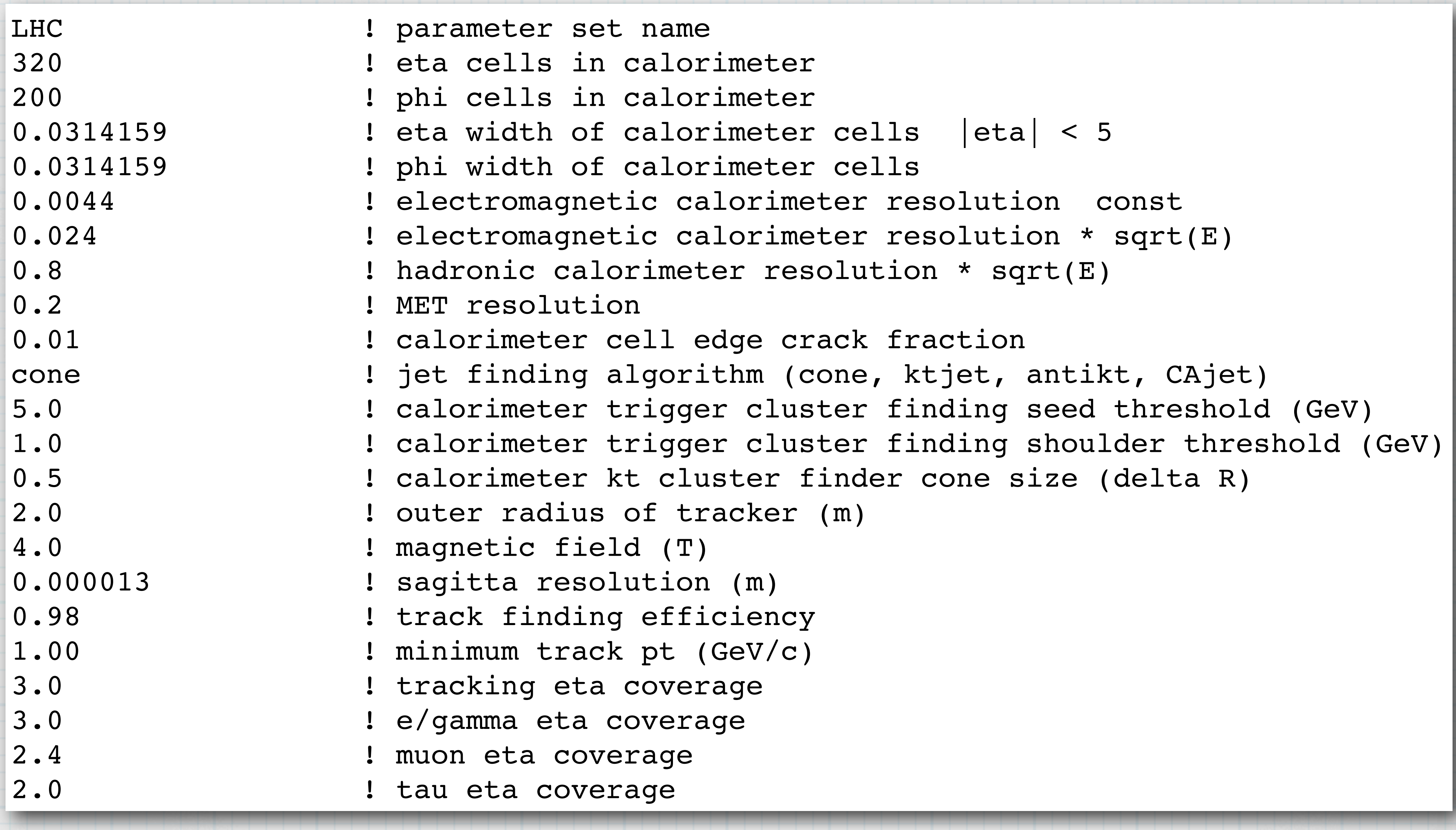
- loop through all final-state particles
- if charged, make charged track (straight...)
- calorimeter deposits:
    - gamma/electron: mostly electromagnetic
    - hadron: mostly hadronic
    - muon: minimum ionizing
- calorimeter is idealized, segmented in eta/phi
- resolutions are controllable parameters

PGS Event Simulation

- plots of electromagnetic, hadronic, muonic energy deposits as implemented in PGS:



PGS Parameters



```
LHC                    ! parameter set name
320                    ! eta cells in calorimeter
200                    ! phi cells in calorimeter
0.0314159              ! eta width of calorimeter cells  |eta| < 5
0.0314159              ! phi width of calorimeter cells
0.0044                 ! electromagnetic calorimeter resolution  const
0.024                  ! electromagnetic calorimeter resolution * sqrt(E)
0.8                    ! hadronic calorimeter resolution * sqrt(E)
0.2                    ! MET resolution
0.01                   ! calorimeter cell edge crack fraction
cone                   ! jet finding algorithm (cone, ktjet, antikt, CAjet)
5.0                    ! calorimeter trigger cluster finding seed threshold (GeV)
1.0                    ! calorimeter trigger cluster finding shoulder threshold (GeV)
0.5                    ! calorimeter kt cluster finder cone size (delta R)
2.0                    ! outer radius of tracker (m)
4.0                    ! magnetic field (T)
0.000013               ! sagitta resolution (m)
0.98                   ! track finding efficiency
1.00                   ! minimum track pt (GeV/c)
3.0                    ! tracking eta coverage
3.0                    ! e/gamma eta coverage
2.4                    ! muon eta coverage
2.0                    ! tau eta coverage
```

User is free to change these...at his or her own risk!

PGS Resolutions

- tracking (B field, radius, sagitta)
  - ✓ calculate sagitta, smear, get p_T
  - ✓ includes possibility of charge confusion
- em calorimetry

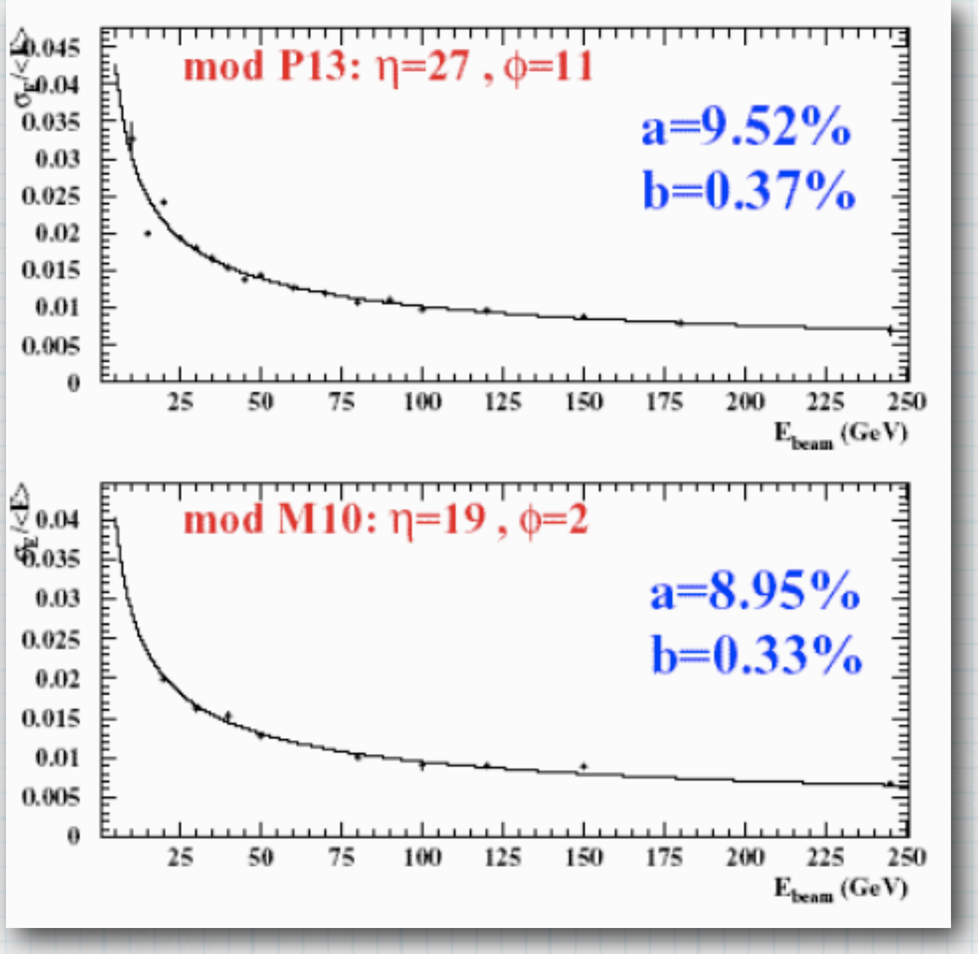
$$\Delta E/E = a + b/\sqrt{E}$$

- hadron calorimetry

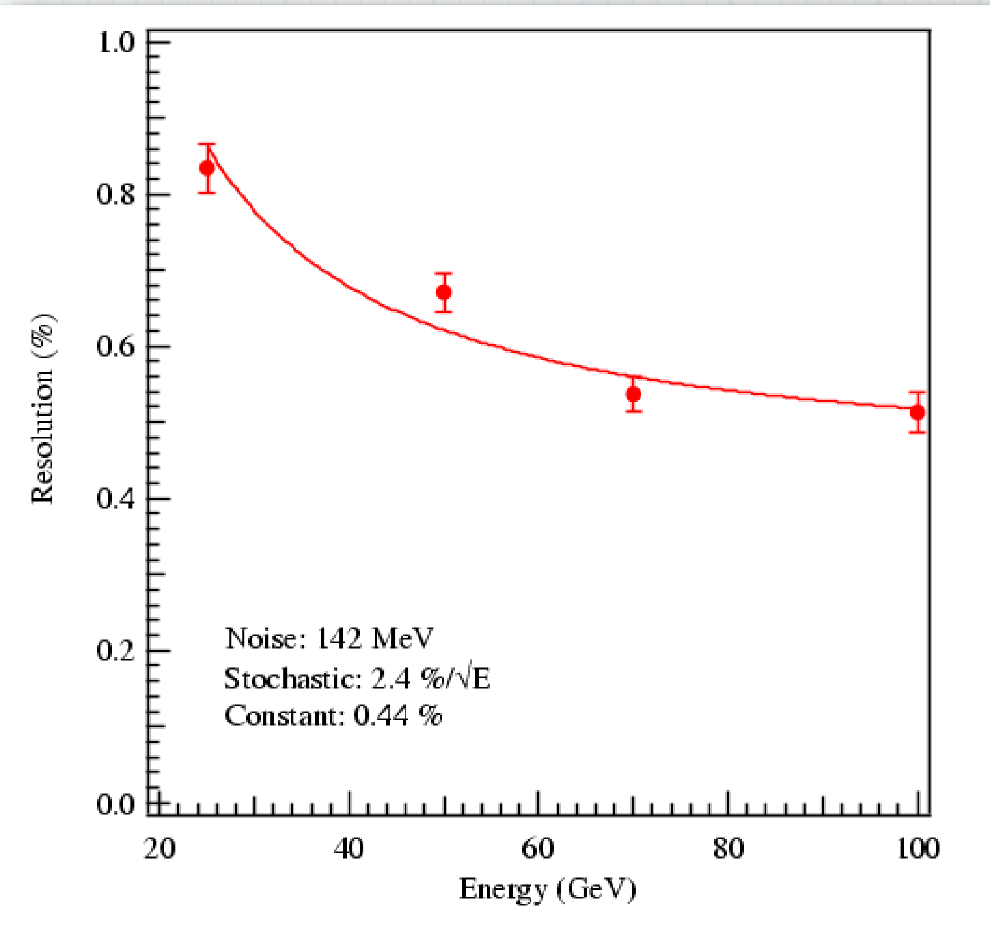
$$\Delta E/E = b/\sqrt{E}$$

ATLAS/CMS Calorimetry

ATLAS



CMS



This is from test beams - does not tell the whole story!

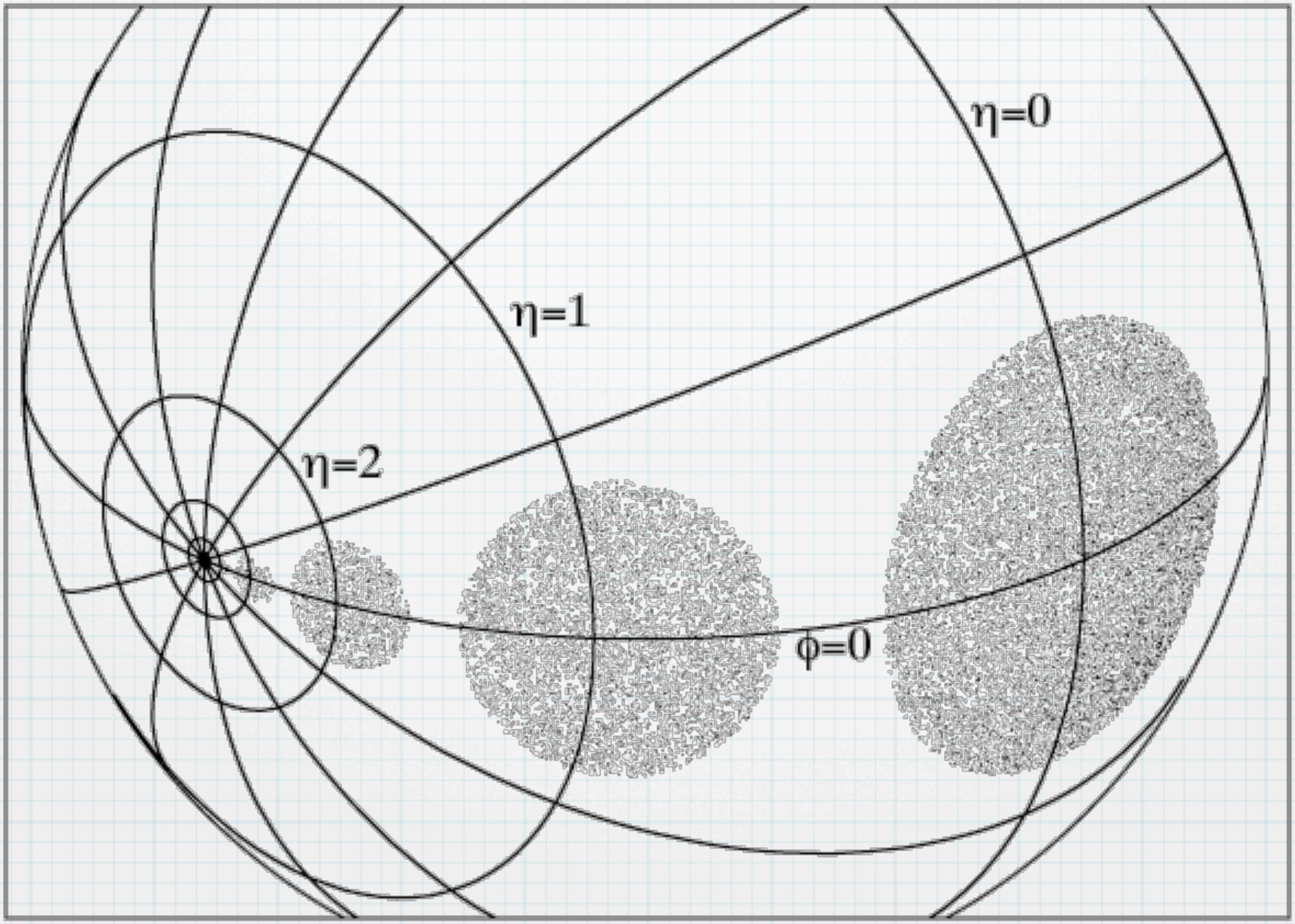
PGS Jet Finding

- after second LHC Olympics, request was made to use kt jet algorithm rather than the "JETCLU"-like cone algorithm formerly used
- both were available: top-down cone jets used for trigger objects, and bottom-up kt jets used for physics jet objects
- kt/anti-kt/CAjet jet finding done with FastJet-like algorithm (N lnN scaling)
- BUG in 090401 release of PGS, corrected as of April 2012 for version 120404 (thanks to Jay Wacker and Anson Hook for finding this!)

PGS Jet Finding

- only calorimeter tower energies used to reconstruct PGS jets! (no track info used)
- jet algorithms differ in the tails of various distributions
- funny-shaped jets (e.g. with g radiation) will always be a difficulty
- is ΔR even the right measure of separation?
- ΔR is z-boost invariant but the solid angle subtended by cones of constant ΔR varies dramatically with pseudorapidity

We plot here random points lying within ΔR of 0.4 from several reference points:



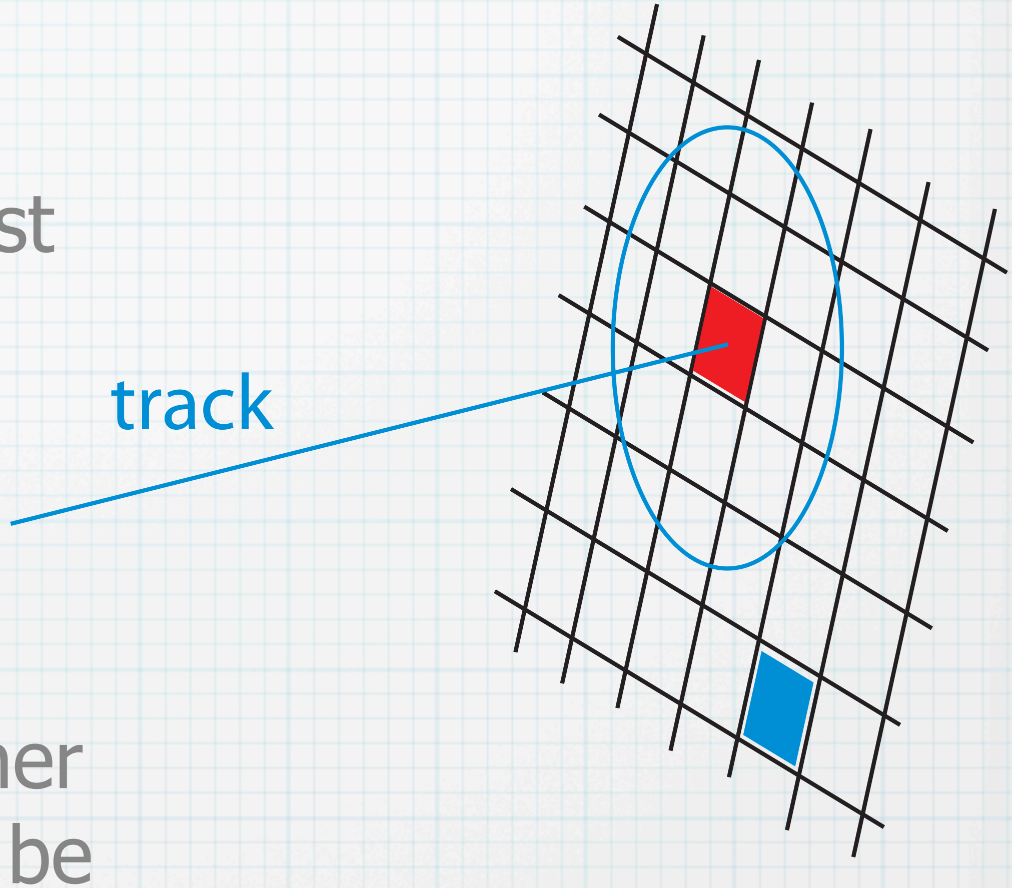
ΔR used for jet finding/merging, isolation, ... is it what we want in all cases?

PGS Electrons/Photons

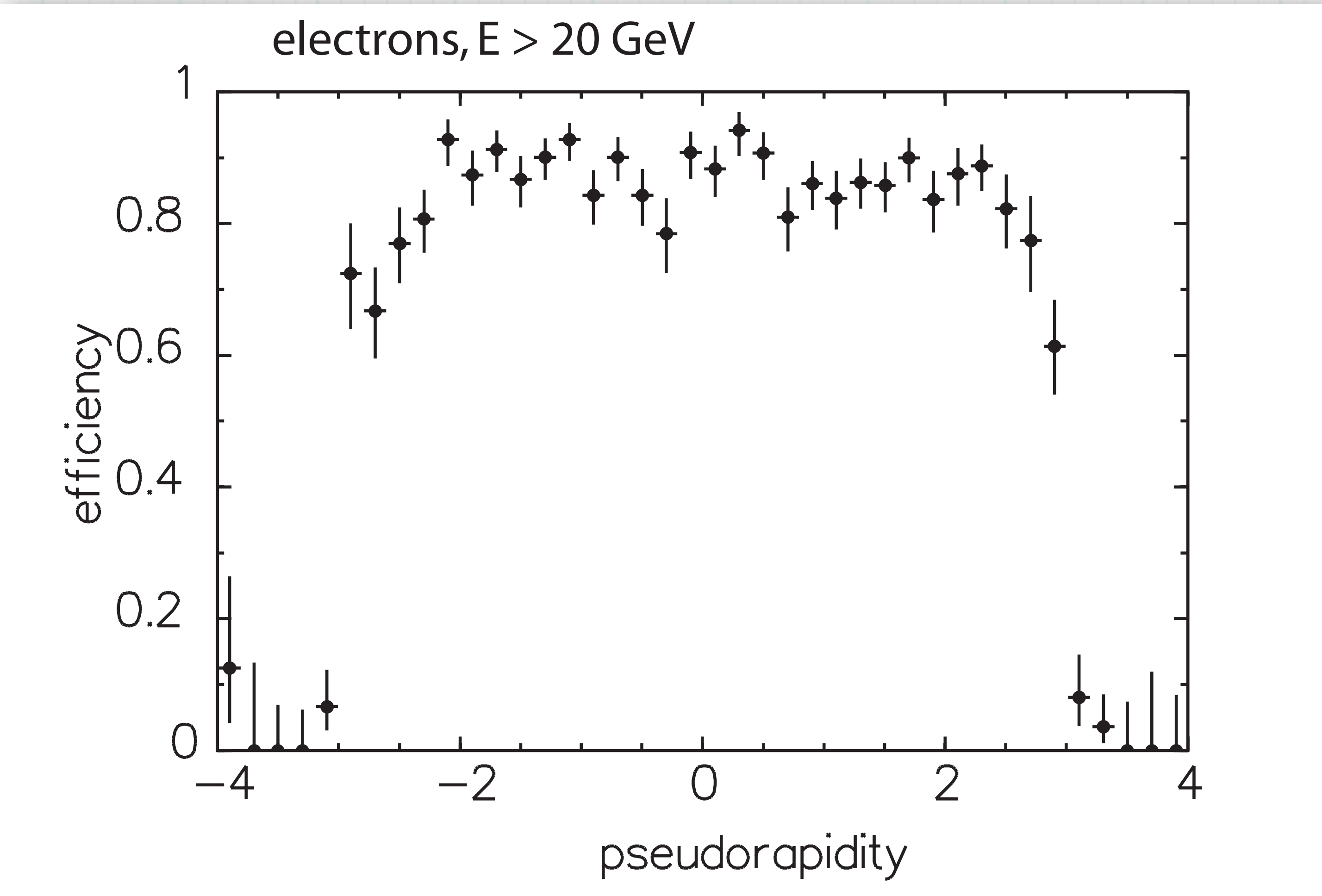
- in real life electromagnetic showers are narrow; hadronic showers are wide
- in PGS there is no lateral shower spread: the energy of each particle goes in one tower
- we simply rely on the fact that the energy is deposited in the em section of the calorimeter
- start with clusters and apply e.m. fraction cuts, match with track
- apply calorimeter isolation cut (3x3 region)

PGS Electrons/Photons

- look at em fraction of cluster (single tower most likely)
- see if there is a track; no track ⇒ photon
- require sum of p_T of other tracks in ΔR cone of 0.4 be less than 5 GeV
- require sum of energy in 3x3 collar region < 0.1 E



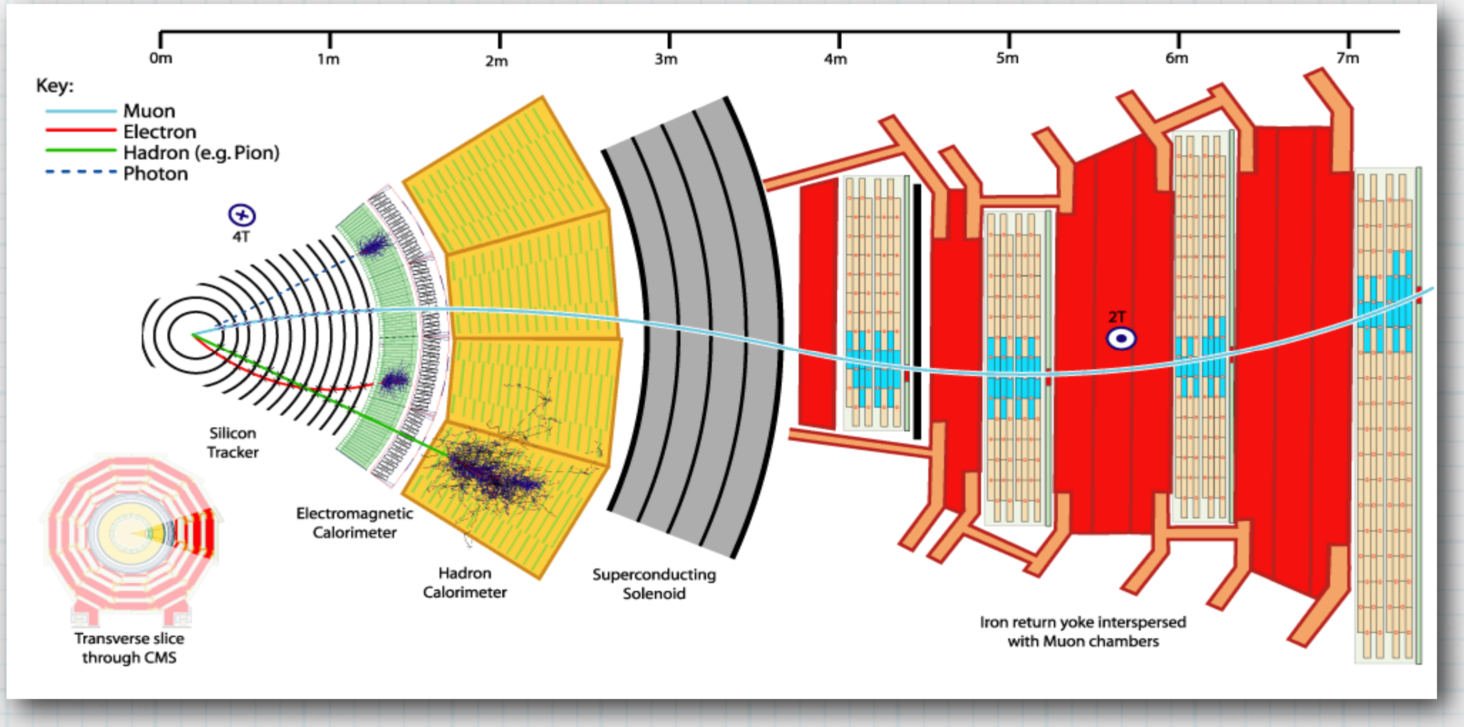
PGS electron efficiency



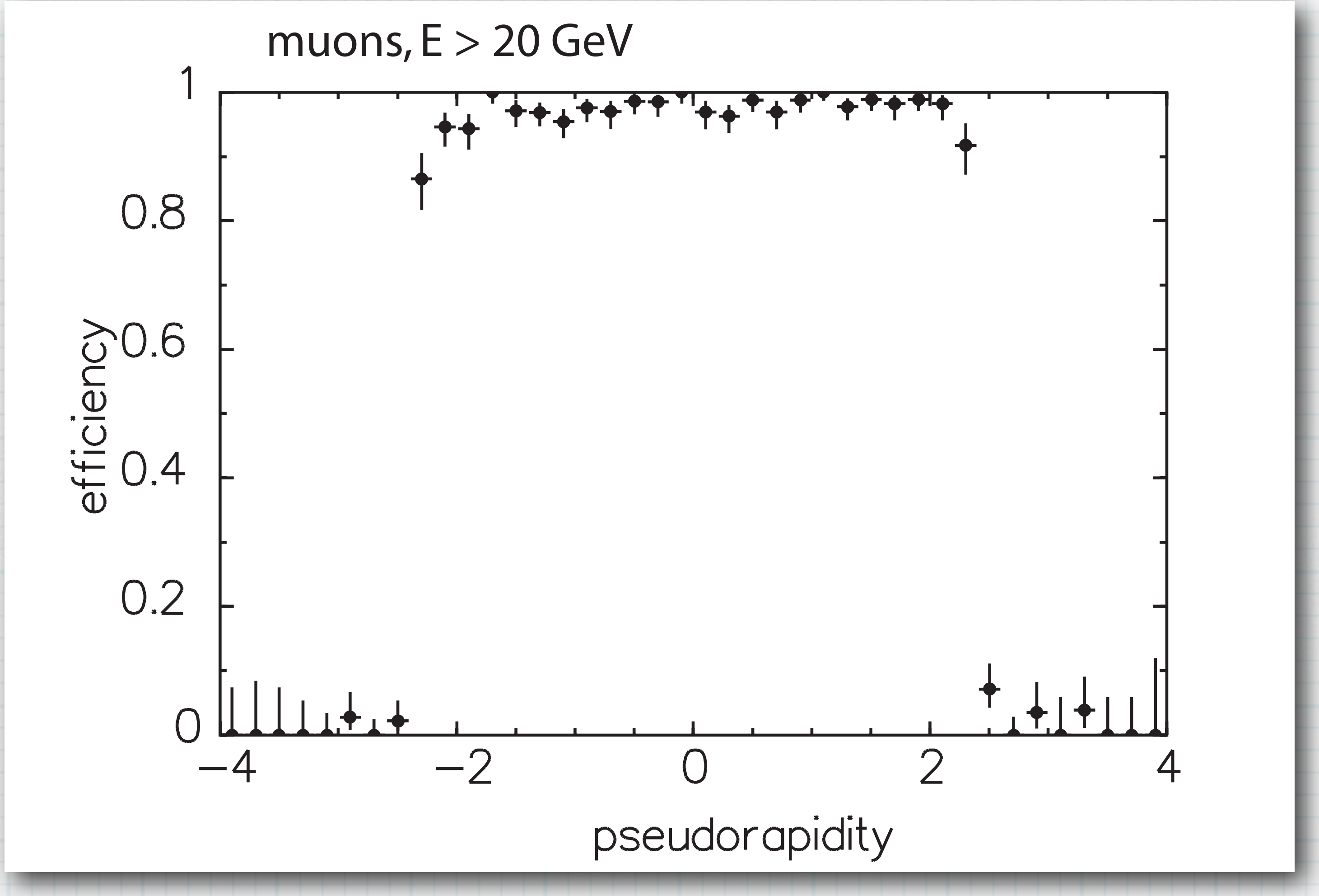
- efficiency about 87% out to $|\eta| = 3$

PGS Muons

- ATLAS/CMS muon systems are highly efficient/redundant!
- We provide a parametrized efficiency function but we do <u>not</u> apply it by default (more relevant to CDF)
- Also, we do not apply a muon isolation cut by default, and leave that to the user (applied in the olympics executable)



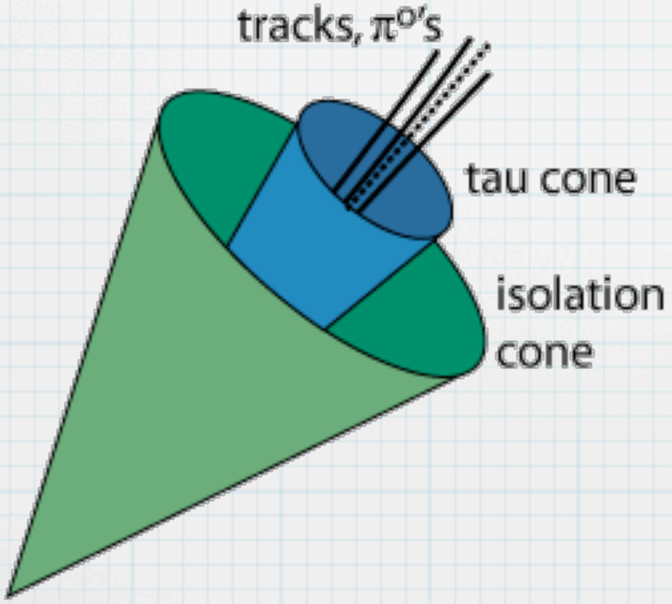
PGS muon efficiency

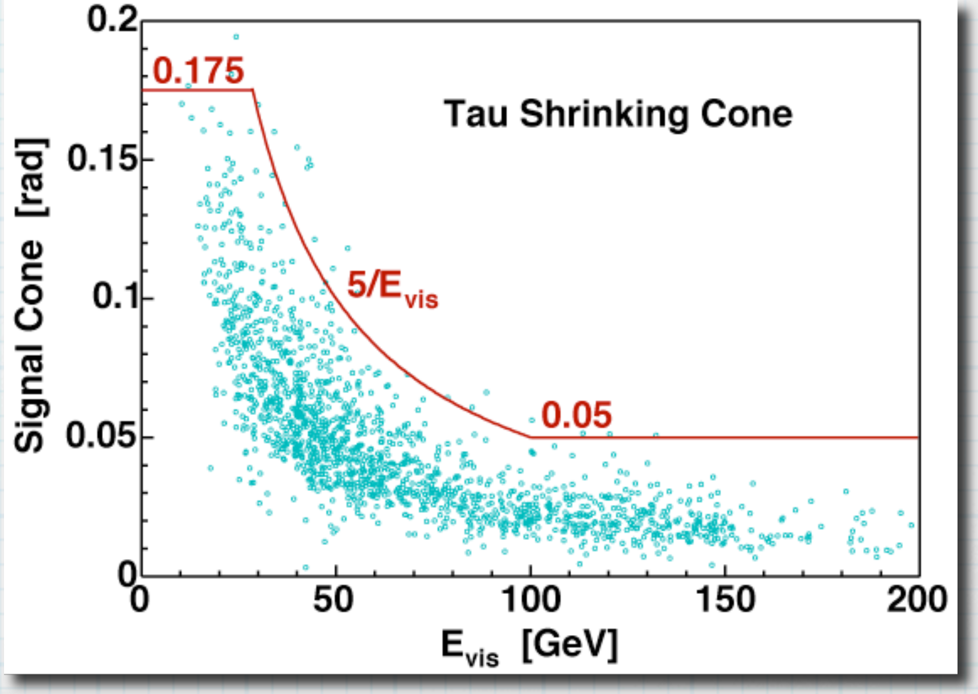


- efficiency about 97% out to |η| = 3

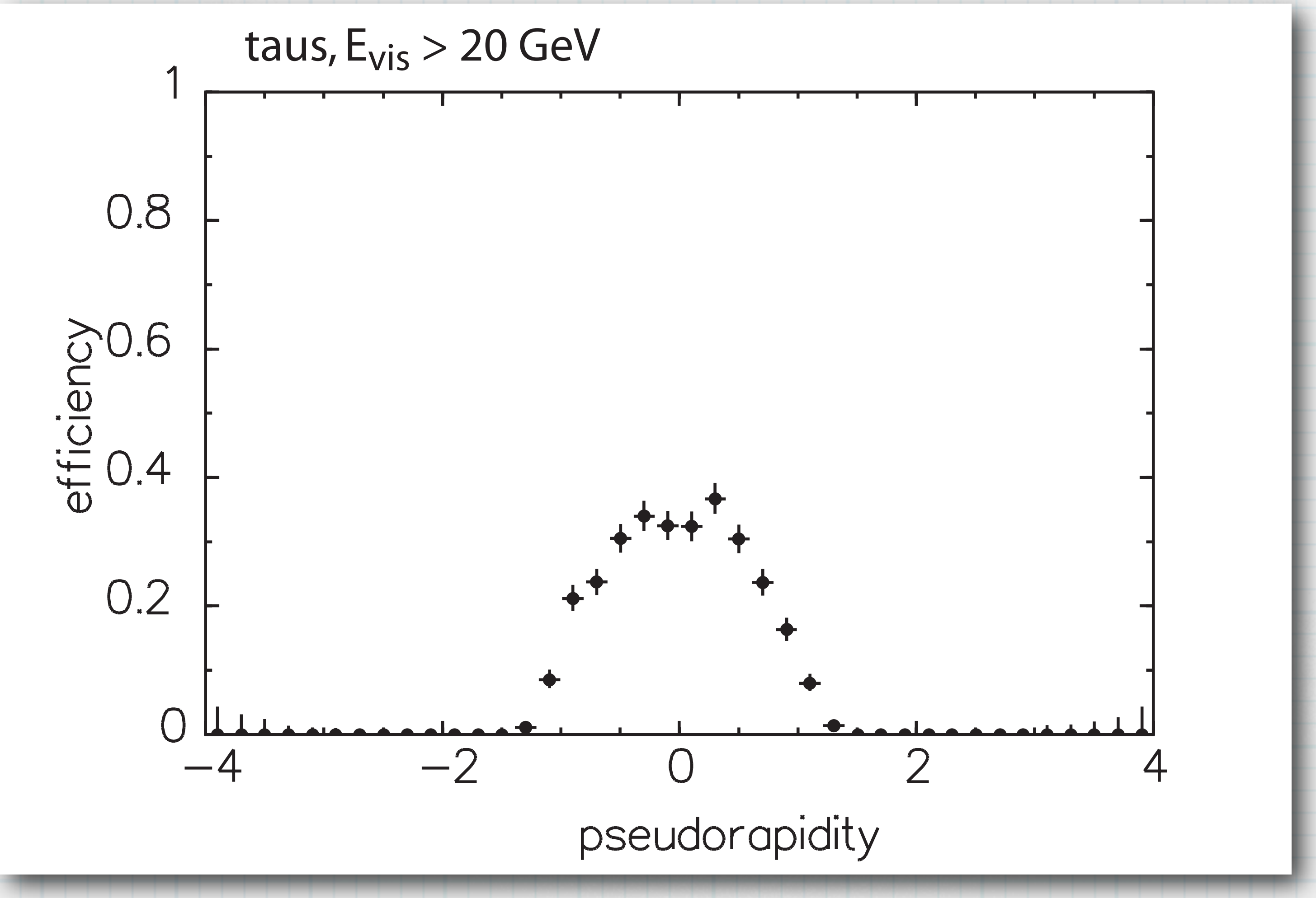
PGS Tau Reconstruction

- traditional standard approach at hadron colliders: cone-based algorithm
- use CDF-style "shrinking cone" surrounding high-p_T seed track
- we "fake" the π^0 reconstruction





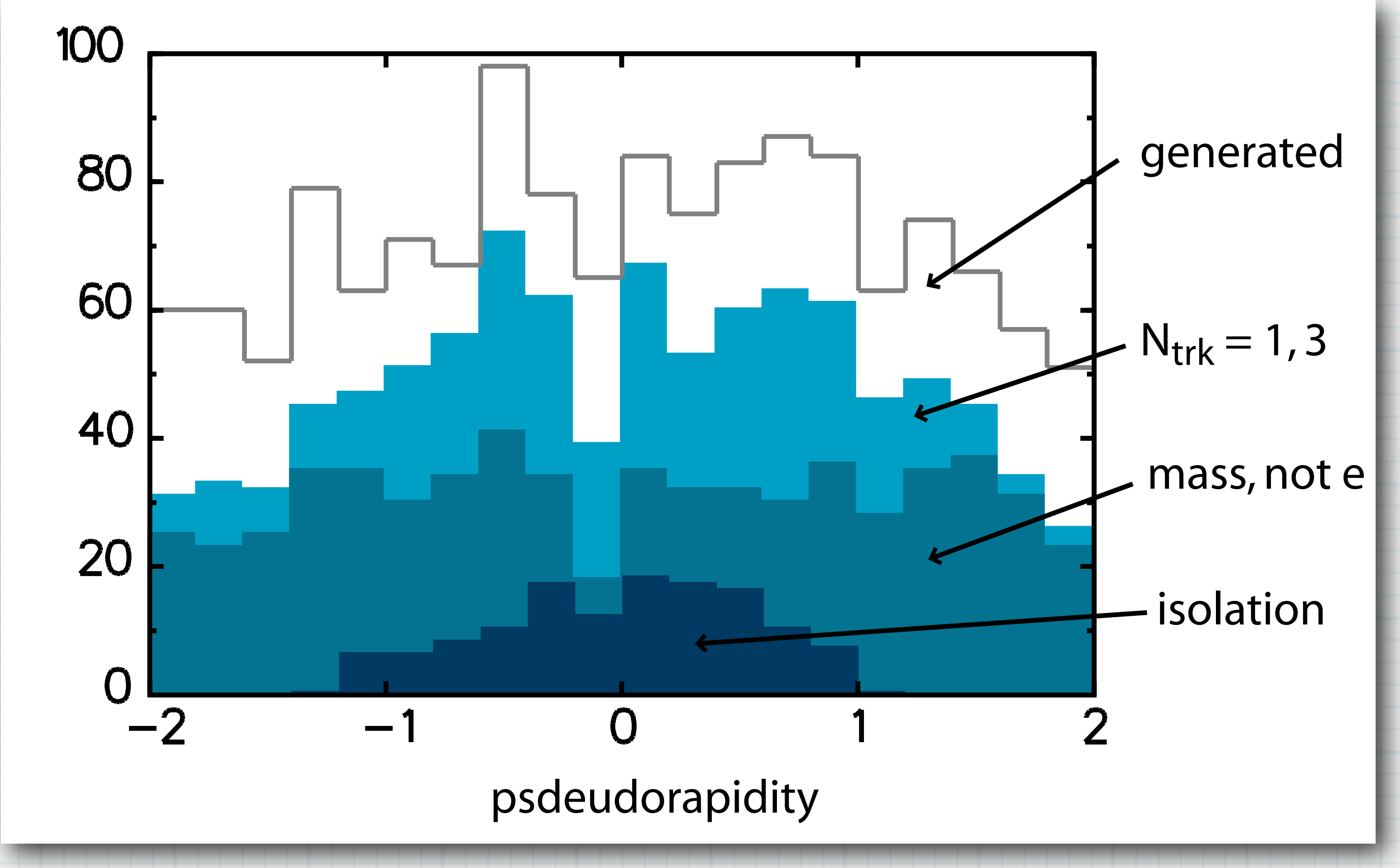
PGS tau efficiency



- efficiency much smaller than electrons, falls of rapidly at high pseudorapidity

PGS tau efficiency

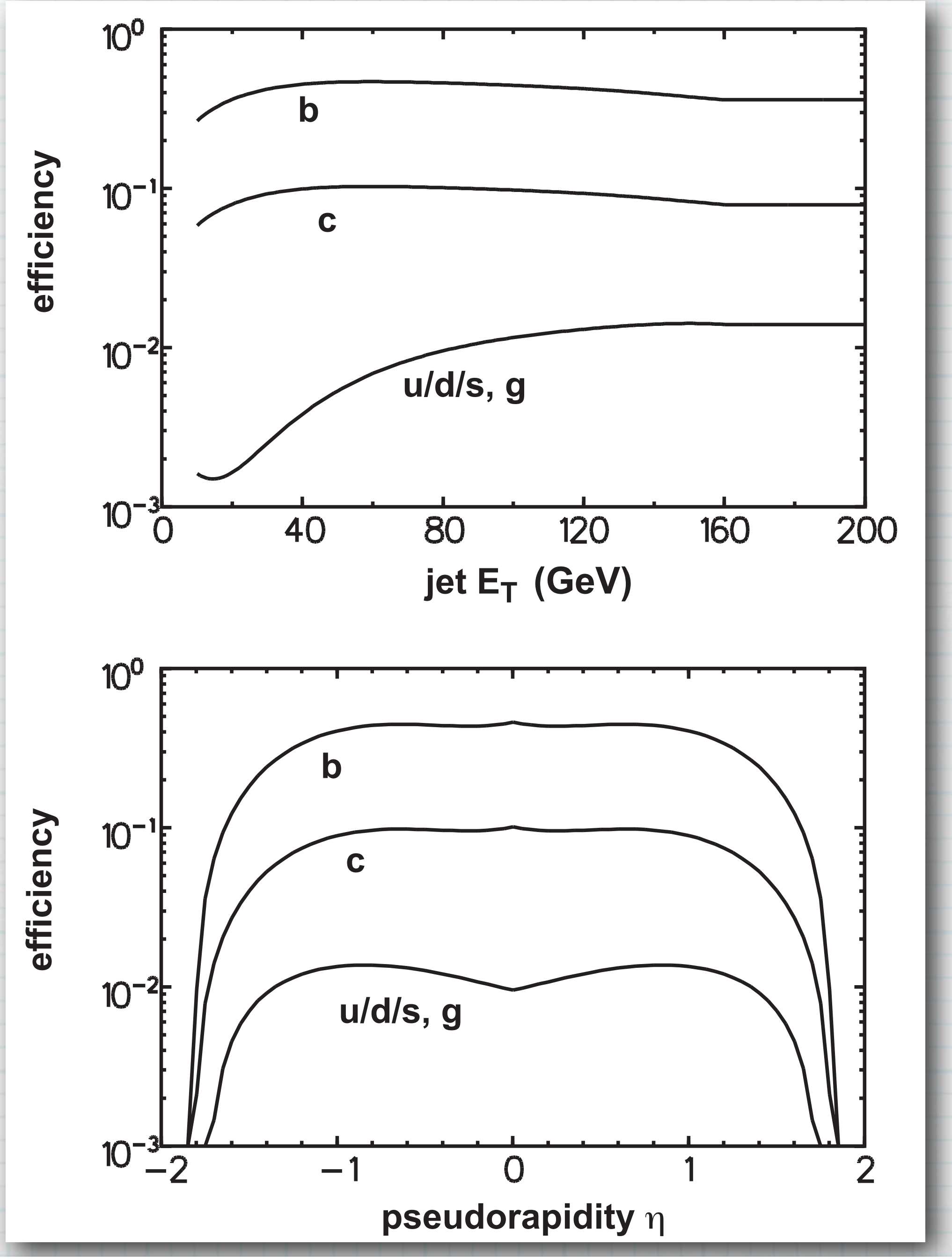
- can we understand which cut is hurting us?



- could be modernized (NN approach)

PGS b-tagging

- parametrize b-tagging efficiency as a function of jet ET, eta
- use MC truth to tell "true jet type"
- this parametrization based on CDF Run 2
- not too far from actual LHC experience...but needs updating



Uniqueness

- a given calorimeter energy (kt jet) cluster can give rise to
  - photon or electron
  - tau
  - jet
- must have algorithm to decide which it is!
- cannot call it two different things!

Uniqueness

- we define physics object precedence:

  γ > e > τ > jet

- if object is already identified as an electron it cannot be a tau or a jet; tau cannot be jet
- jet is "catch-all" class
- muons are all "unique"
- we do this using 3D angle of 10°
- as of PGS 4, provide "unique" flag for each object

PGS Trigger Objects

- PGS provides crude “trigger objects” formed from cone algorithm cluster and tracks:
    - gamma: em deposit, no track
    - electron: em deposit with track
    - muon: straight 98% on all muons that make tracks
    - tau: subset of tau cuts
    - jet: any cluster
- not realistic for LHC

PGS Input/Output

- PGS was designed to call PYTHIA/HERWIG/ISAJET to generate events and simulate on the fly
- PGS can read STDHEP binary files (XDR)
- PGS can read LHE files, run PYTHIA
- Output format is up to the user; Root is not the default, but technically possible
- LHC Olympics: ASCII output format became widely used (LHCO files)

Example Olympics Output

```
 #  typ      eta    phi      pt    jmas  ntrk  btag   had/em  dum1  dum2
                          0            1   3585
 1    4   -1.312  3.143  104.54   21.59  19.0   0.0     1.22   0.0   0.0
 2    4   -1.233  0.957   85.10   15.90  11.0   0.0     5.78   0.0   0.0
 3    4   -2.939  1.139   38.38   26.74  20.0   0.0    63.11   0.0   0.0
 4    4    3.226  5.123   37.37   34.33   8.0   0.0     1.10   0.0   0.0
 5    4   -3.718  4.691   21.52    1.55  17.0   0.0     1.35   0.0   0.0
 6    4    0.211  5.752   12.75   15.57   0.0   0.0     1.03   0.0   0.0
 7    4    1.008  3.038   12.60    4.18   3.0   0.0     1.73   0.0   0.0
 8    4   -2.106  4.275    7.93    2.75  19.0   0.0     3.32   0.0   0.0
 9    6    0.000  6.008   15.64    0.00   0.0   0.0     0.00   0.0   0.0
                          0            2   3599
 1    2   -1.317  3.638    3.36    0.11  -1.0   6.0    11.41   0.0   0.0
 2    2   -1.388  1.845   12.23    0.11   1.0  10.0     0.10   0.0   0.0
 3    4   -0.044  5.646   79.40  335.20   0.0   0.0     1.63   0.0   0.0
 4    4   -0.341  1.677   56.31   32.28   8.0   0.0     5.10   0.0   0.0
 5    4   -3.391  5.279   55.44   30.84  20.0   0.0     1.11   0.0   0.0
 6    4   -1.242  3.464   36.02   34.93   9.0   0.0     2.23   0.0   0.0
 7    4    3.875  2.981   23.08   25.33  12.0   0.0     1.78   0.0   0.0
 8    4   -2.934  0.093   11.33    2.15  21.0   0.0     6.17   0.0   0.0
 9    4   -1.584  4.694   11.12    2.39  18.0   0.0     5.91   0.0   0.0
10    4   -1.716  1.913    9.09    2.20  12.0   0.0     0.90   0.0   0.0
                           0            3   3585
 1    4    0.523  0.059  225.21   48.39  19.0   0.0     3.19   0.0   0.0
 2    4    1.336  3.220  228.44    3.75  10.0   0.0    10.04   0.0   0.0
 3    4    2.918  0.007   62.64  123.09  13.0   0.0     1.53   0.0   0.0
```

Future of PGS

- Should PGS have a future?
  - it’s Fortran
  - no transition to PYTHIA 8
  - not well documented (the code is, though!)

- Work is needed most on b-tagging, tau ID

Why was PGS successful?

- self-contained, simple distribution
- PGS worked out of the box
- supported on Linux, OS X
- final “product” was short list of physics objects that users could easily analyze
- simulation was quite crude but reproduced the main effects: acceptance, resolution, and efficiency
- very fast (100 Hz on ttbar events on Core i7)

Getting PGS

- PGS web page:

http://physics.ucdavis.edu/~conway/research/software/pgs/pgs4-general.htm

- PGS users mailing list:
    - send email to PGS_users@fnal
    - leave subject blank
    - put “subscribe” in first line of message