Last lecture on triggering and data acquisition … next …

# Part II of Course: Data treatment

Today and Friday … Overview of how to make an 'arf decent measurement

Today … - Basic Definitions
- Luminosity
- Monte Carlo techniques

# Data Treatment and Analysis: the Task

- A typical analysis may have many millions of events ☺, but it's rarely enough just to count them!

- Only raw data is obtained from detector (digitised drift chamber, calorimeter signals etc)

- Reconstruct events … list of tracks, primary & secondary vertices, jets, some identified particles (e, $\gamma$, $\mu$, $\nu$)

- Detectors are not perfect! Reconstructed information needs to be treated (corrected) in many ways before it can be compared with theory or another imperfect detector!

- Data analysis is about extracting meaningful, transferable results from the reconstructed detector information

# The role of the Data Analyser

The job of the data analyser is to ……

- Correct the data for detector effects
- Minimise and control systematic effects
- Quantify statistical and systematic uncertainties
- Compare data with theory and test agreement
- Present the data in a <u>digestible, transferable</u> form

… in as unbiased a way as possible!

`Digestible, transferable results' can take many forms …

Simple numbers (model parameters, masses …)
Ratios or asymmetries between 2 numbers
Cross sections or differential cross sections

Next 2 lectures will look at measuring a differential (binned) cross section … if you can do that, you can do most things!

# Cross Sections … Definitions

- Probabilitiy that interaction occurs between 2 particles can be thought of as the effective area that one of the particles presents  to the other. 'Cross section' is therefore an area:

  1 barn=$10^{-28}$ $m^2$   (1b) ~ size of a Uranium nucleus ("it's as big as a barn" said a physicist scattering neutrons off Uranium whilst developing the atomic bomb in WWII!)

  … but these days we deal in much smaller units pb or fb
- Formally, cross section for process ab->X is the rate of the process per incident flux of particle a per target particle b
- i.e. $\sigma$=(reactions per unit time)/(number of incident particles per unit time per unit area x number of target particles)

… How does this help?…

# Luminosity … Definitions

- <u>Instantaneous Luminosity</u> is a property of an accelerator, defined as …

  $L_{inst}$ = # of events produced per unit x-sec per unit time.
  (e.g. LHC design luminosity is $10^{34}$ $cm^{-2}$ $s^{-1}$)

$L_{inst}$ depends on beam currents and overlap area ($L \propto I_a I_b / A$)

- Integrated over the time taken to collect our event sample, luminosity is a property of the sample, telling us how many events to expect per unit cross section

- i.e. we can extract cross sections using <u>**$\sigma$=N/L**</u> where …

  N = number of events counted

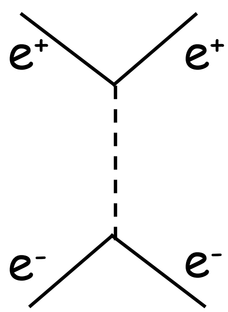
  L= <u>integrate</u>d luminosity of sample (units $barn^{-1}$)

- We just need to count events N and somehow measure the integrated luminosity (over time) L of our sample.

# Measuring Luminosity

- In practice, beam currents can be easily measured, but their overlap area at interaction point cannot!
- Luminosity is determined by measuring the number of events for a process with a `well known' cross section… …ideally a high rate process with well known cross section, calculable from QED. OK for $e^+e^-$, ep machines. Hard for pp!

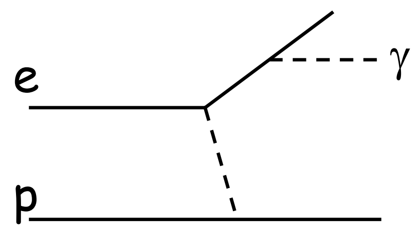
LEP




Bhabba Scattering

$\sigma(L)/L \approx 0.1\%$

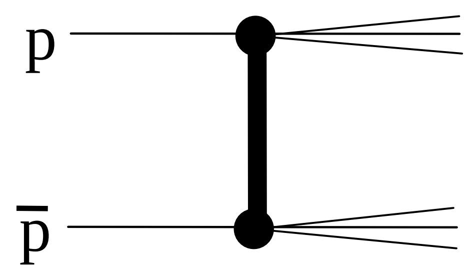
HERA




Bethe-Heitler

$\sigma(L)/L \approx 1\%$

Tevatron / LHC

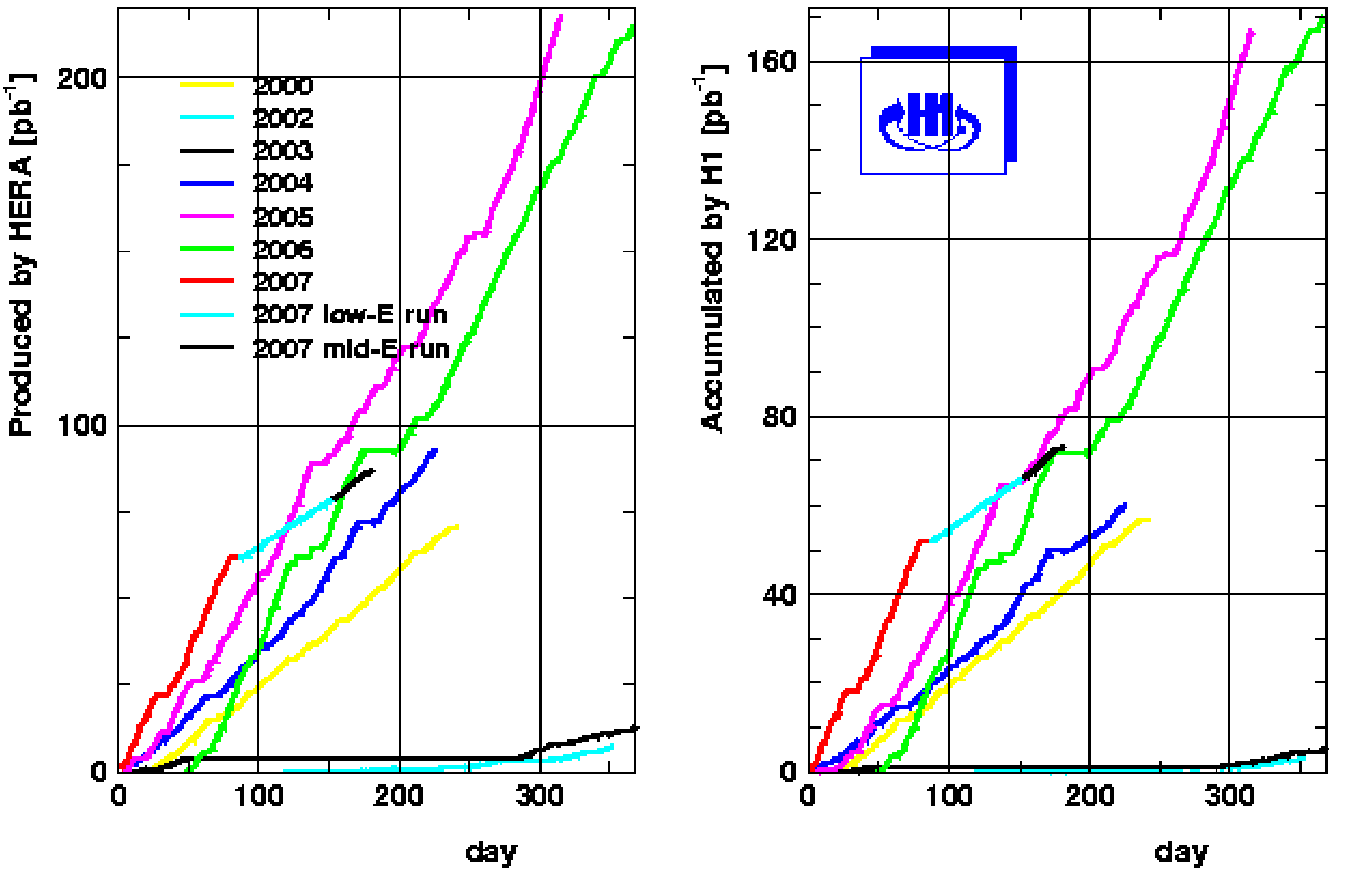



e.g. total or elastic cross section, not well known

Teavtron $\sigma(L)/L \approx 7\%$

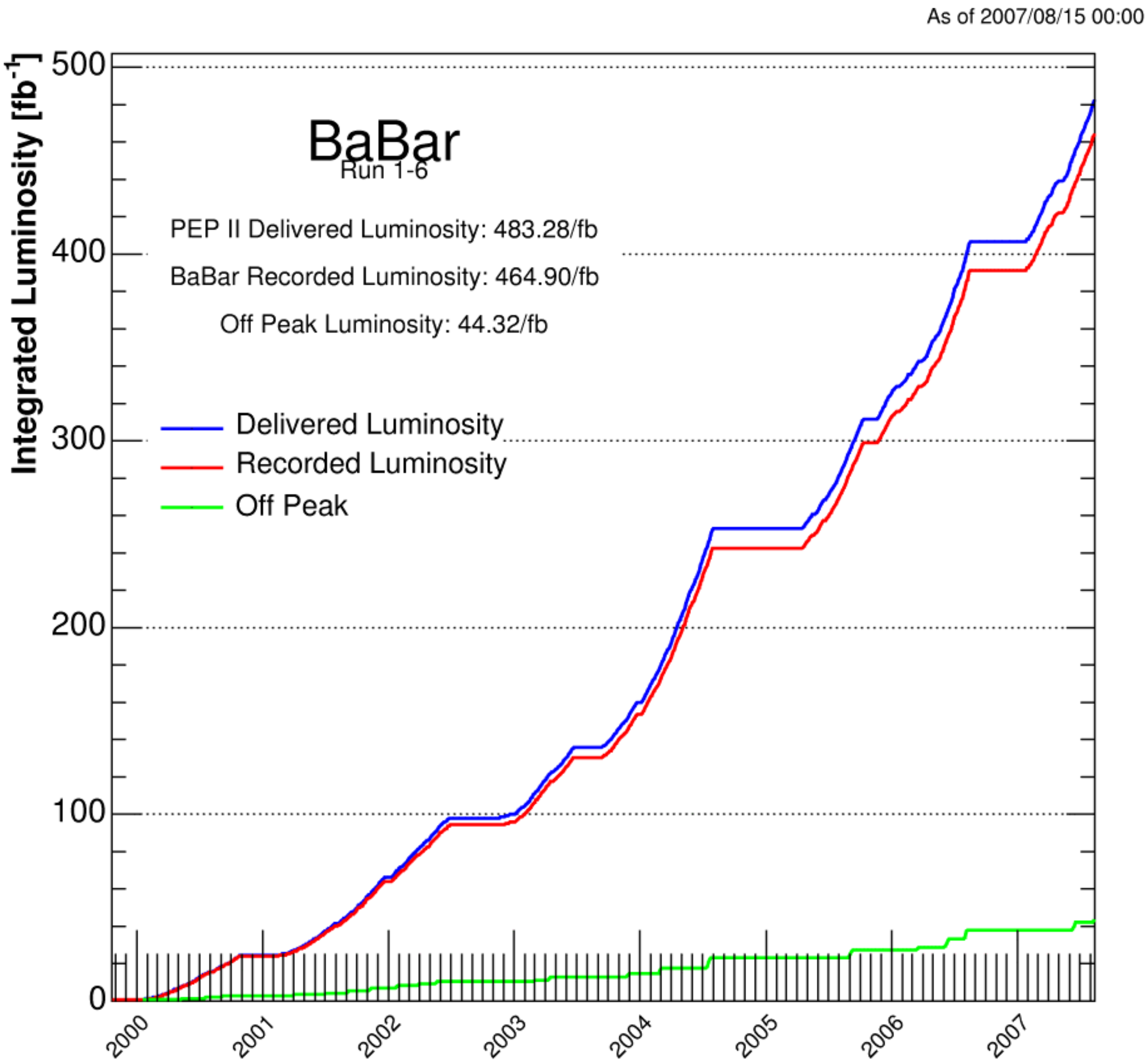
Still not fully resolved for LHC!

# Example … H1 Integrated Luminosity



# Example from BaBar

Current world records held by b factories…




1 fb$^{-1}$ per day achieved!
(same as LHC design aim …
a few years away)!

Grand total delivered
is 480 fb$^{-1}$
(slightly more at Belle)

# How to deal with an Imperfect Detector

Need to correct number of observed events to number of expected events given a perfect detector:

- no holes / cracks
- perfect efficiency for measuring particles
- perfect resolution and no misalignments / miscalibrations
- no background events … etc!!!

- Modified cross-section formula:

$$\sigma = \frac{N_{obs} - N_{b/g}}{L \,.\, \varepsilon \,.\, A_{cc}}$$

  - $N_{obs}$ = number of observed events
  - $N_{b/g}$ = number of background events
  - $A_{cc}$ = Acceptance
  - $\varepsilon$ = efficiency

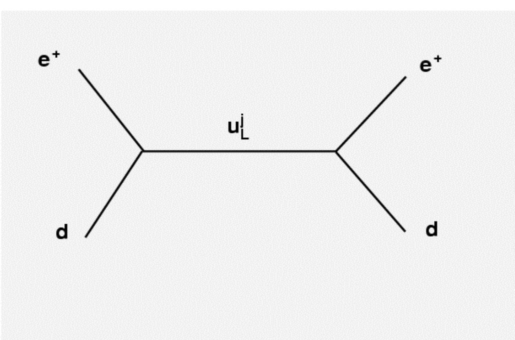
**Acceptance, efficiency, background and most other evils can be calculated using a <u>`Monte Carlo' simulation</u>**

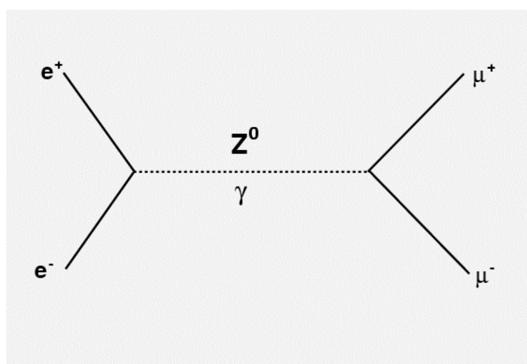
# 3 Steps to a Monte Carlo Simulation

- MC techniques are fundamental to most particle physics results!
- A Monte Carlo simulation begins with a model of the underlying process under study and tells us how we expect it to look in detector … allows us to **understand the effects of our detector and analysis cuts**

## Step 1: An event generator

- A program to calculate the cross-section of a specific physics process in a given kinematic range, using a theoretical model (e.g. Standard Model).
- Given the theoretical distribution, uses random number generator to produce individual events within distribution - hence name Monte Carlo

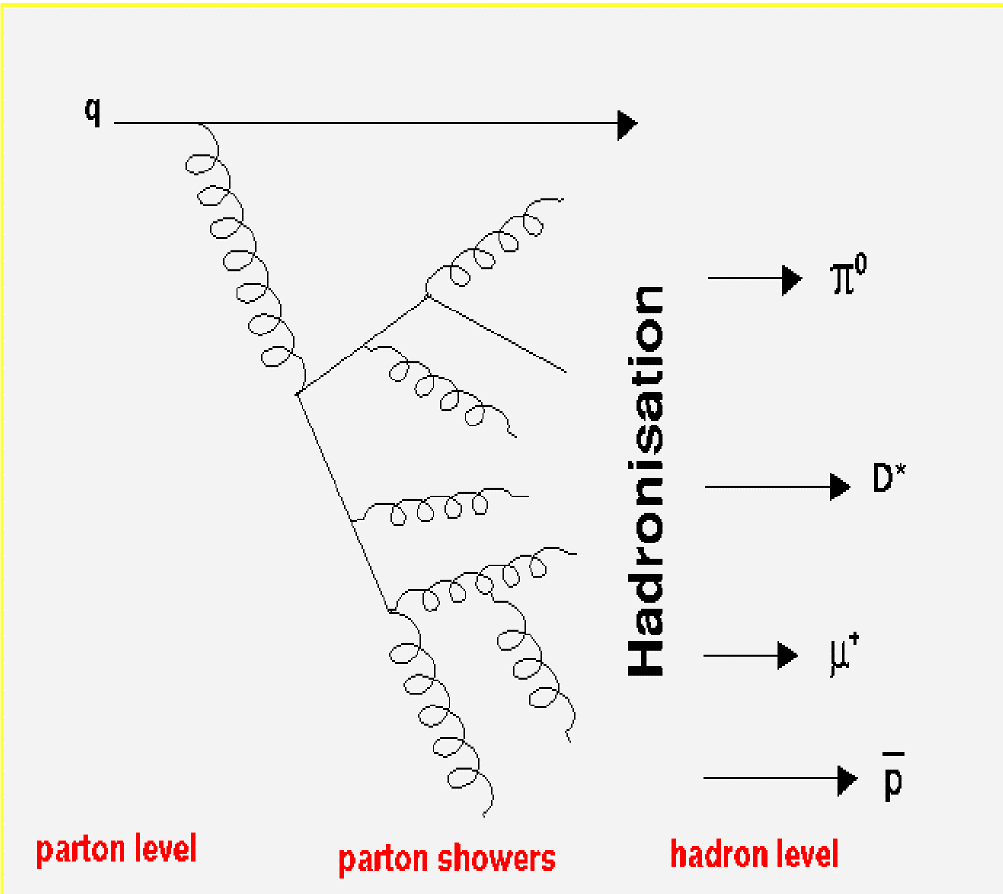



Calculation is usually only done to leading order in QCD (though See MC@NLO for LHC).

# Simulating Higher Order Processes and Hadronisation

- Higher order quark / gluon emissions are simulated statistically using approximations to QCD: usually 'parton showers' (DGLAP evolution)
- High energy particles radiate until all particles have low energy, then forced into hadrons → 'JETSET' (based on the LUND string model)




- Final generator output is a set of "stable" final state particles

… as in this example, but also "fast" decays such as $\pi^0 \rightarrow \gamma\gamma$, $D^{*+} \rightarrow K^-\pi^+\pi^+$

- Complete 4-vector info for every final state particle
- Parton state known as **<u>parton level</u>**
- "Stable" particle state called **<u>hadron level</u>** or **<u>"MC truth"</u>**

# Example of a Generator Steering File

Printed by Eram Rizvi

Feb 13, 01 15:17 — mcsteer.txt — Page 1/2

```
*------------------------------------------------------------------------
GOJG 0 / ! DJANGO 6.2   PARAMETERS
*------------------------------------------------------------------------
*  DJANGO 6.2 parameters and steering variables available to the user
*  are stored in the following commons and can be changed as indicated
*  below. They are explained in the DJANGO and HERACLES manuals.
*
*
*---LEPTO parameter commons
*     COMMON /LEPTOU/ CUT(14),LST(40),PARL(30),X,Y,W2,Q2,U
*
*     The parameters and their defaults are descibed in the LEPTO
*     manual. Please note that LST(2) to LST(19) and PARL(1) to PARL(19)
*     are input parameters, whereas LST(20) to LST(40) and PARL(20) to
*     PARL(30) are output parameters.
*
*  Changing parameters:
*  The order and the number of parameters to be changed are arbitrary.
*  In case of a one-dimensional array only the first index is used, the
*  second index may have any value.
*  The basic rules (for details see the BOS manual) for writing card
*  images are:
*     1) columns 1-4 are free
*     2) items (names, indices, values) without inside blanks
*     3) blank(s) between items
*     4) the array names in PAR-NAM have to be shortened to four
*        characters if necessary
*     5) comment cards start with '*'
*     6) space behind "!" in each line reserved for comment
*
*   PAR-NAM  1.INDX  2.INDX NEW-VALUE ! COMMENT IF NEEDED
*   -------  ------  ------ --------- ------------------------------
*
**************** HERACLES steering cards ******************************
*
*   'EELE'     0      0      26.7     ! (D=30.) Energy of electron
    'EELE'     0      0      27.6     ! (D=30.) Energy of electron
*                                       beam in GeV
*   'POLA'     0      0       1.      ! (D=0.) polarization of lepton
*                                       beam ( -1. ... 1. )
    'LLEP'     0      0      -1       ! (D=-1) Lepton charge
*                                       1=positron, -1=electron
    'EPRO'     0      0      920.     ! (D=820.) Energy of proton beam
*                                       in GeV
    'Q2WG'     0      0      $q2wg     ! (D=0.) Q^2-weighting of events
*                                       the value specifies below which
*                                       the events will be weighted
*   'ICUT'     0      0       1       ! (D=3) definition of cuts applied
*                                       1=cuts in x and lower Q2
*                                       2=cuts in x, lower Q2, W
*                                       3=cuts in x, y, Q2, W
*
*                                       must be 3 for the use of THMI!
    'XMIN'     0      0      $xmin   ! (D=1E-4) lower cut in x
    'XMAX'     0      0      0.999     ! (D=1.0 ) upper cut in x
*   'YMIN'     0      0      0.001    ! (D=0.0 ) lower cut in y
*   'YMAX'     0      0      0.99      ! (D=1.0 ) upper cut in y
    'Q2MI'     0      0      $q2min     ! (D=4.0 ) lower cut in Q**2
    'WMIN'     0      0      5.0      ! (D=5.0 ) lower cut in W (W2 90)
*ER generate electrons only up to 133 Deg
    'THMI'     0      0      $thmin     ! (D=0.0 ) lower cut in Theta_el.
*                                       in rad. wrt. the incoming electron,
```

Feb 13, 01 15:17 — mcsteer.txt — Page 2/2

```
*                                      ICUT must be 3, if THMI is used!
*                                      extremely small THMI-cuts (mrad) must
*                                      be avoided!
*  'PTMI'     0       0       0.      ! (D=0.0 ) lower cut on the transverse
*                                      momentum of the scattered electron
*  'EGMI'     0       0       0.5     ! (D=0.0 ) cutoff energy for
*                                      bremsstrahlung photons
*                                      0 =radiative and non-rad. ev.
*                                      >0=only hard photon bremsstr.
   'INT2'     1       0       1       ! (D=0) integration of the non-
*                                      radiative contribution (NC)
*                                      0=off, 1=on
*  'INT2'     2       0       1       ! (D=0) integration of the non-
*                                      radiative contribution (CC)
*                                      0=off, 1=on
   'INT3'     1       0       5       ! (D=0) integration of initial   (radcor)
*                                      state bremsstr. contr. NC
*                                      number of iterations for VEGAS
*                                      <100:VEGAS, >100:VEGAS1/2
   'INT3'     2       0       5       ! (D=0) as INT3(1) but for final (radcor)
*                                      state bremsstr. NC
   'INT3'     3       0       5       ! (D=0) as INT3(1) but for Compton (radcor
)
*
*                                      contribution NC
*  'INT3'     7       0       5       ! (D=0) integration of initial
*                                      state bremsstr. contr. CC
*                                      number of iterations for VEGAS
*                                      <100:VEGAS, >100:VEGAS1/2
   'NPVE'     0       0   10000       ! (D=1000) number of integration
*                                      points used for VEGAS
*  'LPIN'     1       0       1       ! (D=2) EW-parameters
*                                      1=from Mw,Mz; 2=from Mz, G
*  'LPIN'     2       0       0       ! (D=1) only Born x-section term   (radcor
)
*
*                                      or 1=including corrections
*  'LPIN'     4       0       0       ! (D=1) leptonic QED corrections
*                                      1=included, 0=excluded
*  'LPIN'     5       0       1       ! (D=0) quarkonic QED corrections
*                                      1=included, 0=excluded
*  'LPIN'     6       0       1       ! (D=0) lepton-quark interference
*                                      1=included, 0=excluded
*  'LPIN'     7       0       0       ! (D=2) fermionic contr. to
*                                      the photon self energy
*                                      0=no 1=with q-mass 2=param.
*  'LPIN'     8       0       1       ! (D=0) fermionic contribution
*                                      to the gamma-Z mixing
*  'LPIN'     9       0       1       ! (D=0) fermionic contr. to the
*                                      self energy of Z-boson
*  'LPIN'    10       0       1       ! (D=0) fermionic contr. to the
*                                      self energy of W-boson
*  'LPIN'    11       0       1       ! (D=0) purely weak contr. to self
*                                      energies, boxes, vertex corr.
*  'M111'     0       0       2       ! (D=1) =0 fixed alpha_s (P111)
*                                      =1 running alpha_s to 1st order
*                                      =2 running alpha_s to 2nd order
*  'M115'     0       0       2       ! (D=0) =0 diverge at q2-->0
*                                      =1 softened diverge at q2-->0
*                                      =2 fixed below q2=4
*  'P111'     0       0       0.2     ! (D=0.2) fixed alpha_s value
   'P112'     0       0       0.235   ! (D=0.25) Lambda used for alpha_s
```

Tuesday February 13, 2001 — mcsteer.txt — 1/1

Typically, MC production needs several pages of steering parameters just for event generator step!

- MC steering parameters obtained from detailed tuning to match (previous) data

# 3 Steps to a Monte Carlo Simulation

**Step 2: A simulation of "stable" particles in the detector**

- Each experiment creates a detailed detector simulation: e.g. description of all materials where particles could interact
- A program (usually GEANT) takes generator output (particle 4-vectors) and simulates detector response to those particles
  - particle ionisation in trackers
  - energy deposition in calorimeters
  - intermediate particle decays/radiation
  - detector noise …
- Final output is raw detector data just like real data at read-out
  - charges measured on each tracker wire
  - Signals read out of each calorimeter channel
- MC events now in same format as real raw data!

# 3 Steps to a Monte Carlo Simulation

## Step 3: Reconstruction, cuts etc as for Real Data

- Pass MC "raw data" through same reconstruction chain as data.
- **`Reconstructed Level'** output now in form useable for analysis:
  - particle track trajectories, identified particles etc
- The MC now looks exactly like the data, but crucially,

  we still have complete info about **parton** and **hadron** levels!

  … Can answer questions essential to our measurement. e.g.

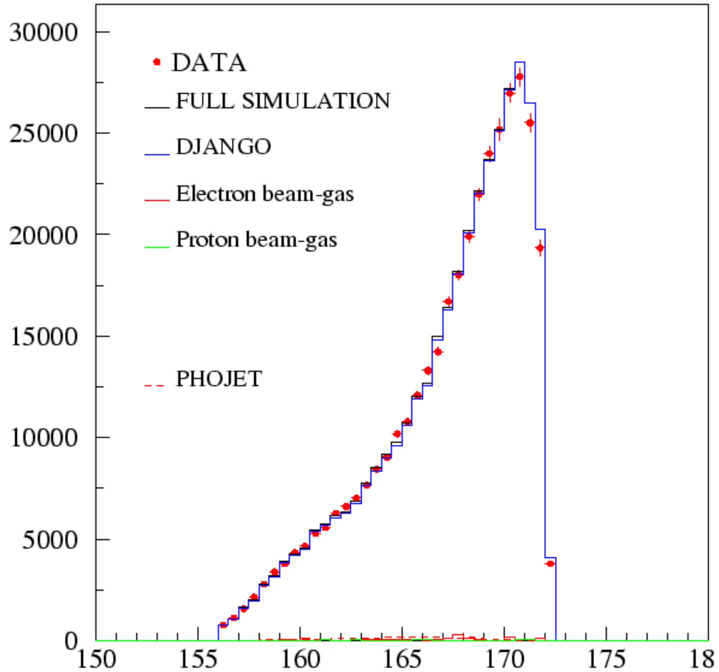
  - What was my acceptance and efficiency?

  - What fraction of my identified `electrons' were really photons?
- To first approximation, these answers are not greatly affected if our underlying generator is a poor model of the physics.
- But for precision measurements, the simulation has to be a good model of the data. Test with `control distributions'

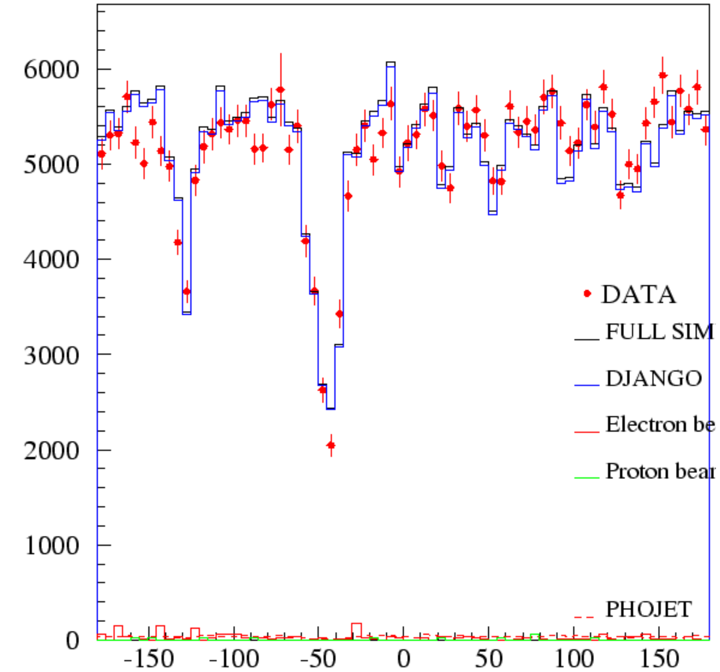
# e.g. Control Distributions of reconstructed data (H1)

… compare data before any corrections with rec level MC …

Scattered electron θ

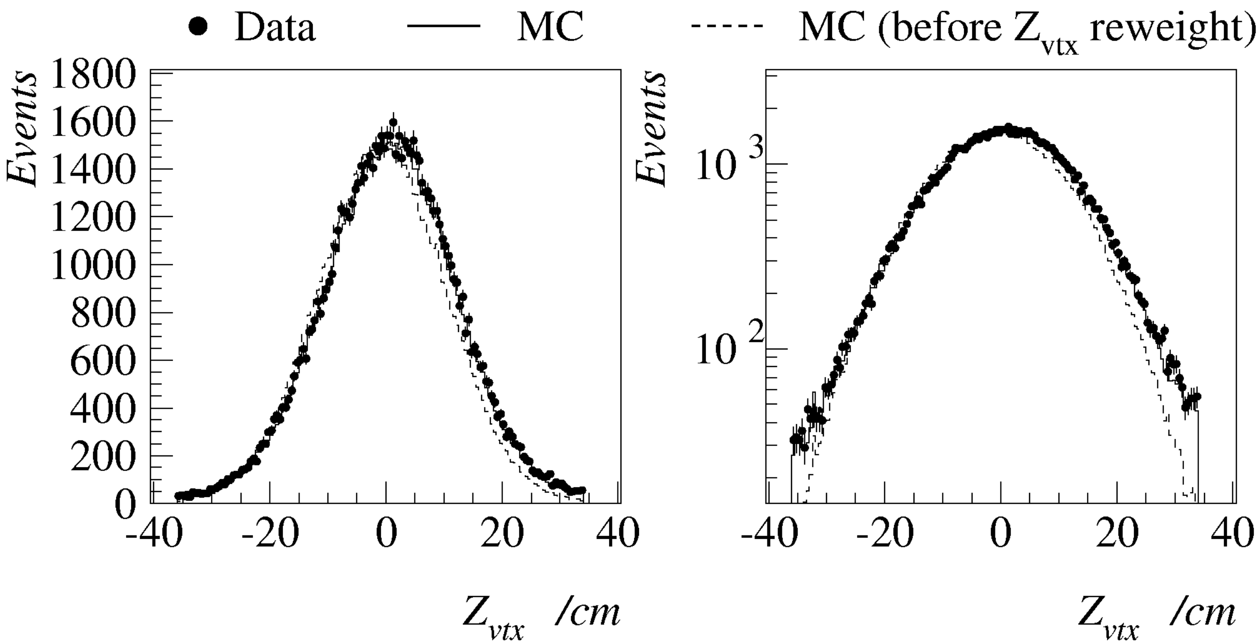



Scattered electron ϕ

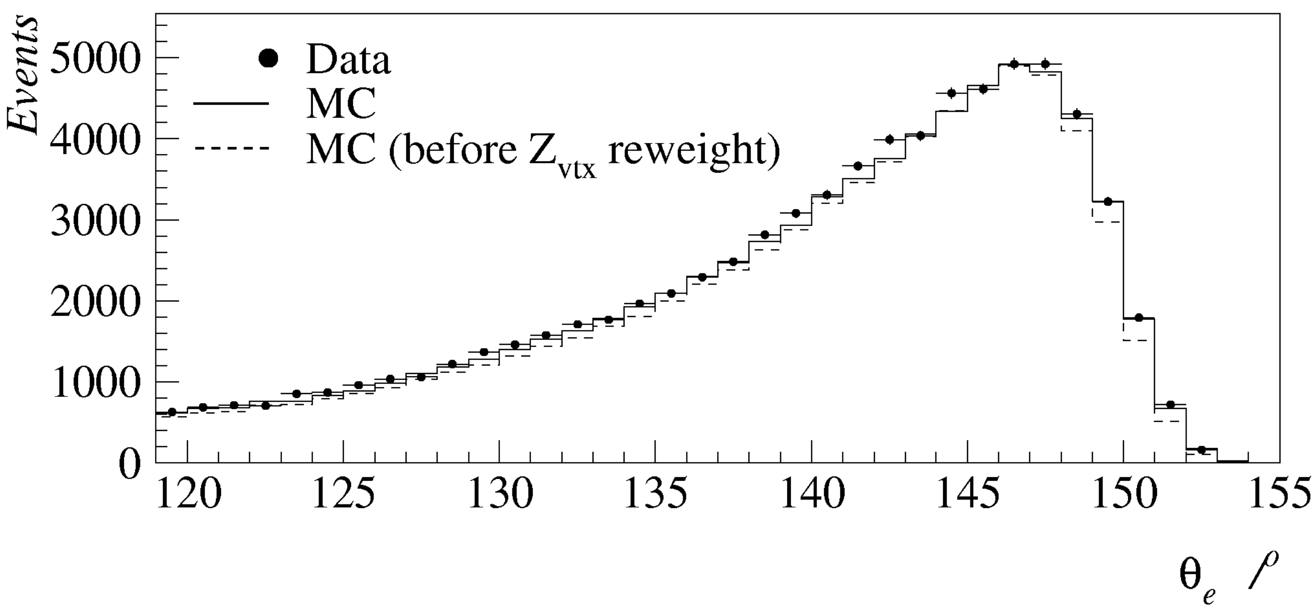



- Good description by DJANGO DIS Monte Carlo simulation ☺
- Big holes in ϕ due to dead bits of detector – but described by simulation
- Background Monte Carlo (PHOJET) gives tiny contribution
- Reassuring results – happy with model - use to correct for inefficiencies such as the holes!

# Bad Control Distributions and their Influence




In this H1 example, distribution of simulated z position of interaction vertex is slightly wrong due to a mistake in steering parameters




Fixing vertex distribution results in much better description of an important phsysics distribution – electron scattering angle.

# What is a Differential Cross Section

Often we are interested in the dependence of a cross section on a variable (e.g. transverse energy of a jet, pseudorapidity, momentum transferred ...)

Work in terms of `differential cross sections e.g. $d\sigma / dE_t$

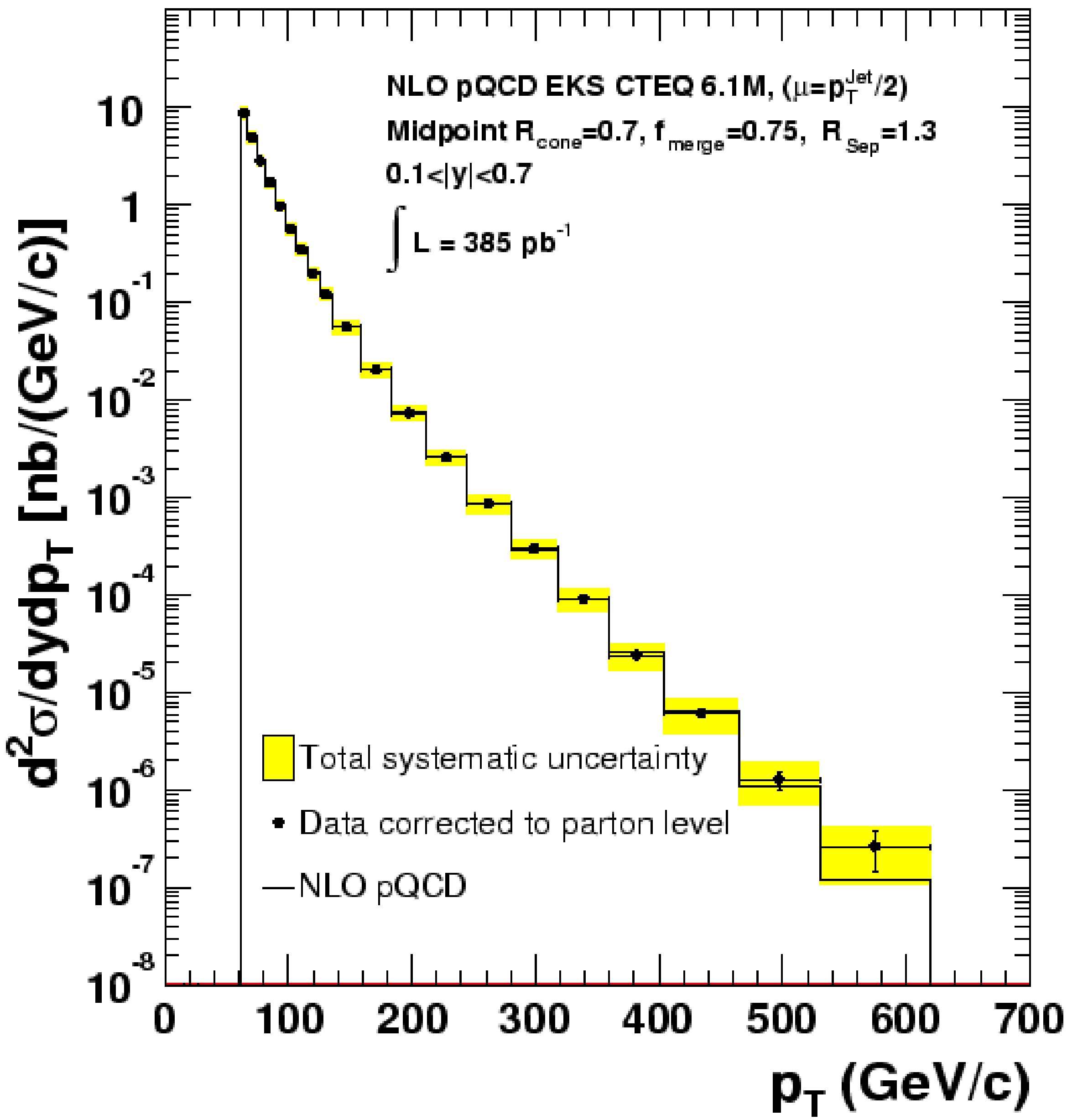
Definition: $$\frac{d\sigma}{dx} = \lim_{\Delta x \to 0} \frac{\Delta\sigma}{\Delta x}$$ such that $$\int_{-\infty}^{\infty} \frac{d\sigma}{dx}\, dx = \sigma_{tot}$$

... measure a differential cross section at a single fixed point in x by creating a bin of width $\Delta x$ about that point ...
... modification to cross section formula ...

$$\frac{d\sigma}{dx} = \frac{N_{obs} - N_{b/g}}{L \,.\, A_{cc} \,.\, \Delta x}$$

... where $\Delta x$ is the width of our bin in x
(still have to worry about exactly which x value we measured at)

# An Example Differential Cross Section




In this example, a jet cross section is measured differentially in the jet transverse momentum $p_T$ and the jet rapidity y

… note the horizontal error bars indicating that the data point could correspond to any point in the bin … (pessamistic!!!)

# `Generalised' or `Smeared' Acceptance

Recall: $$\sigma = \frac{N_{obs} - N_{b/g}}{L\,.\varepsilon\,.\,A_{cc}}$$

Introduce smeared acceptance $A_{cc}$ (calculated from Monte Carlo), which corrects for 'everything' in binned cross section measurements:

1. Finite selection efficiency / acceptance: some events end up in no bin at all.
2. Finite resolution: events end up in wrong bin (migrations).

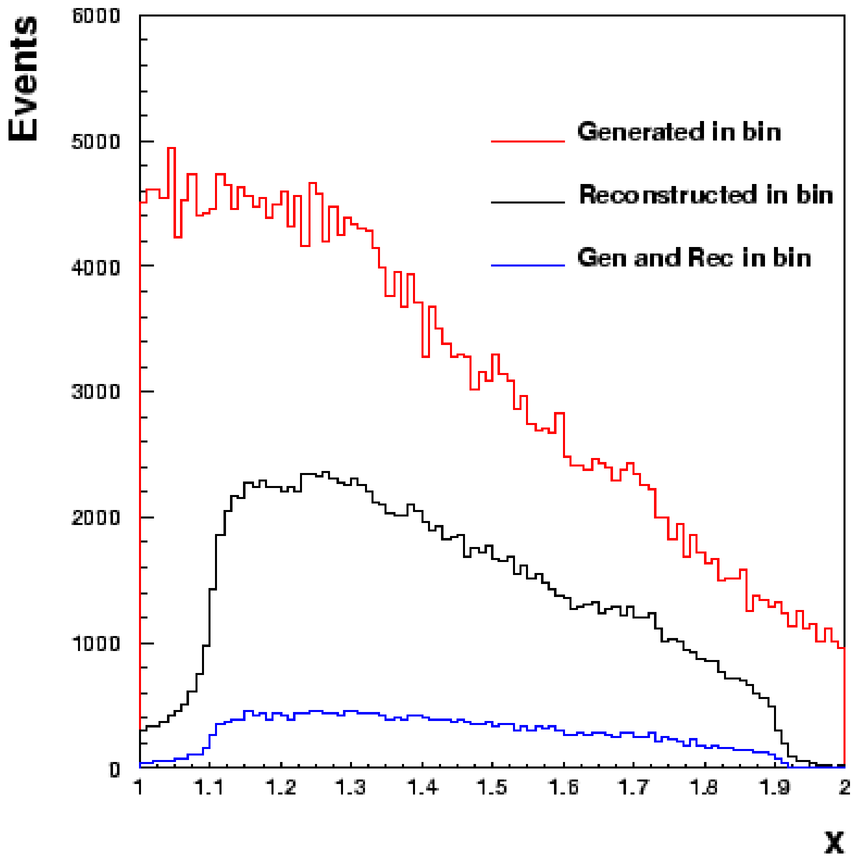
Calculating $A_{cc}$ with an MC can be very easy!..

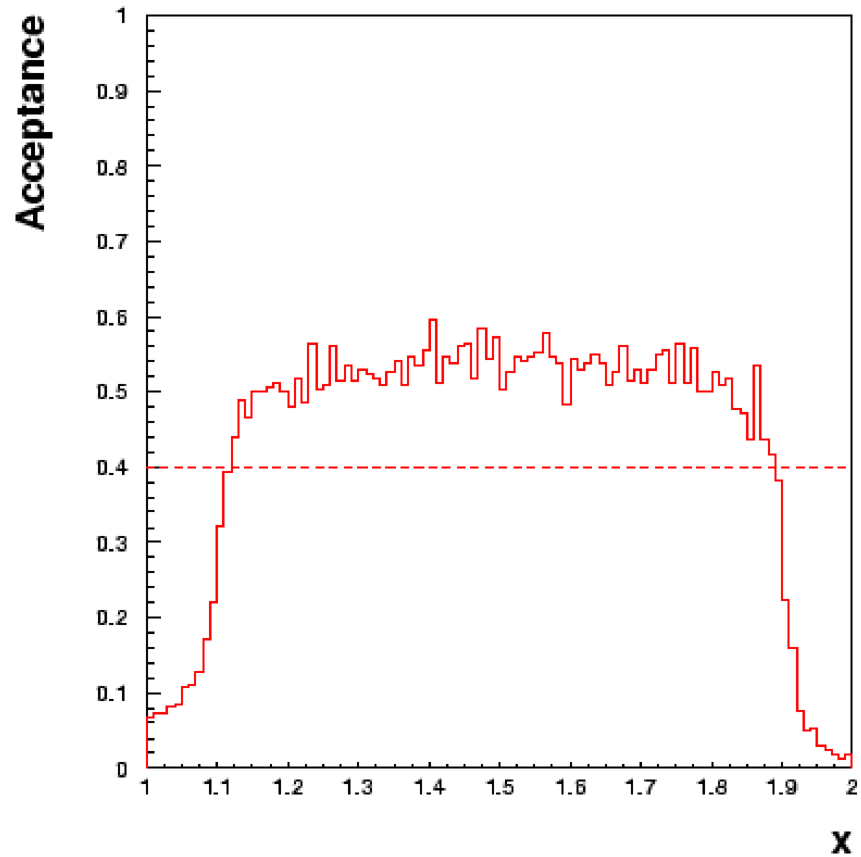
$$\boxed{A_{cc} = N_{rec}/N_{gen}}$$

$N_{rec}$ = # Events reconstructed (found) in a bin (wherever they were generated)
$N_{gen}$ = # Events generated in a bin

This simple treatment is OK if resolutions are well behaved (Gaussian) and bin choices are sensible (migrations not too large) … otherwise more complex 'unfolding' needed.

# A Real Acceptance Example

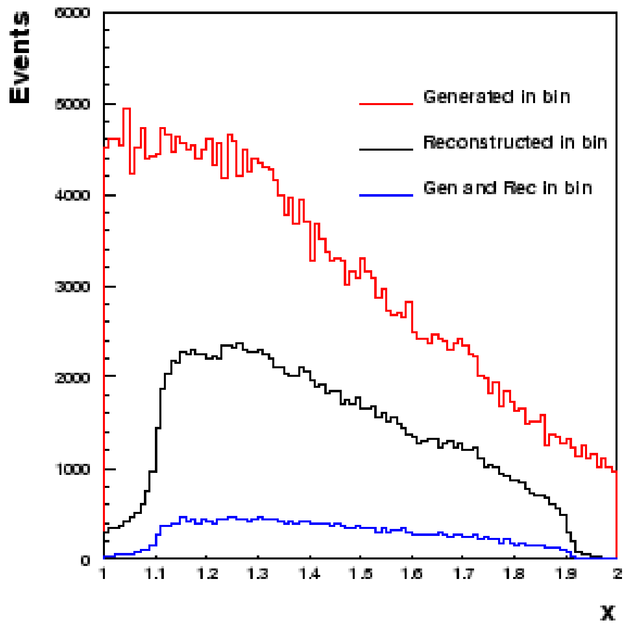


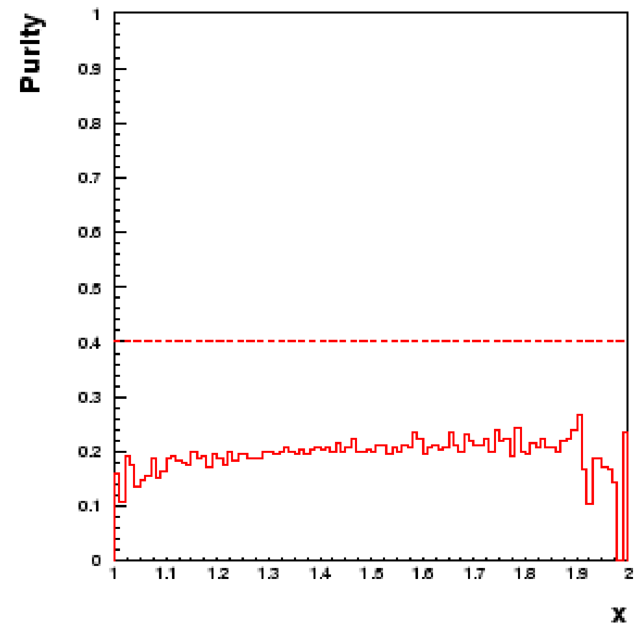



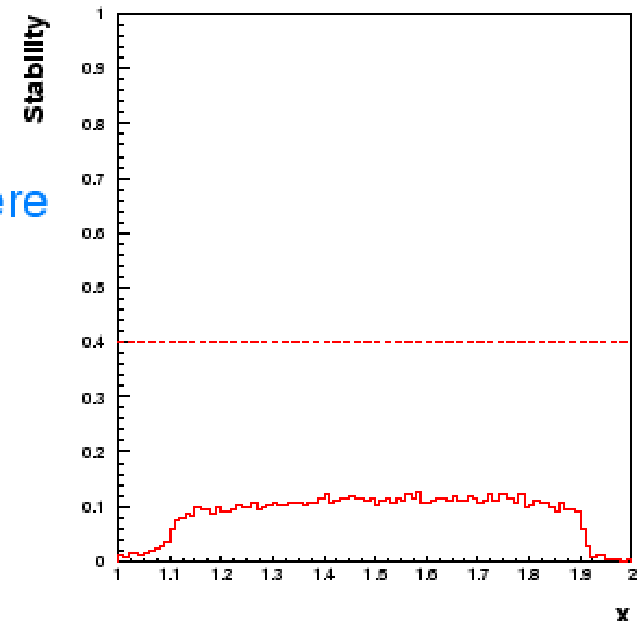

Here acceptance $\sim 50\%$ for $1.1 \lesssim x \lesssim 1.9 \ldots \rightsquigarrow$ can make measurement

. . . but is acceptance the full story? . . . here very few events are reconstructed and generated in same bin . . . we need many fewer bins!

# Bin Purity and Stability



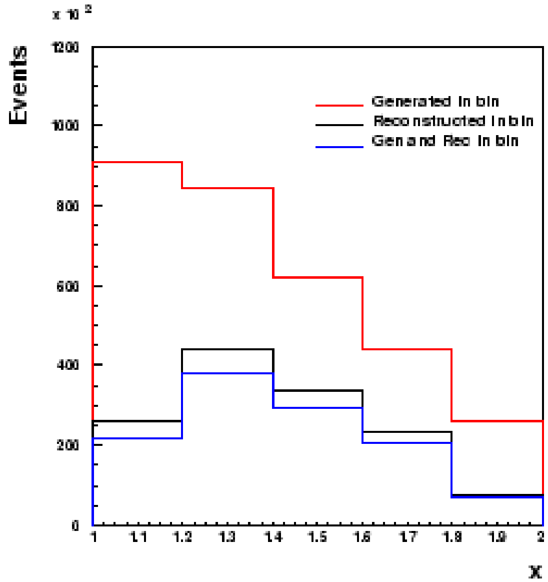

$$\text{Purity} = \frac{\text{gen \& rec in bin}}{\text{rec in bin}}$$ . . . tells us what fraction of events reconstructed in a bin belong there

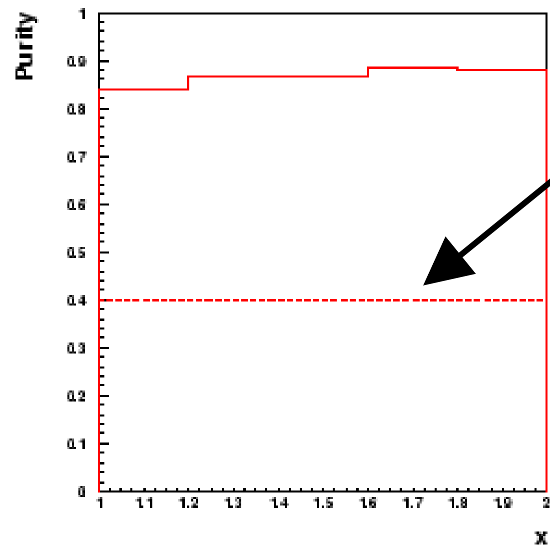
$$\text{Stability} = \frac{\text{gen \& rec in bin}}{\text{gen in bin}}$$ . . . tells us what fraction of events generated in a bin stay there

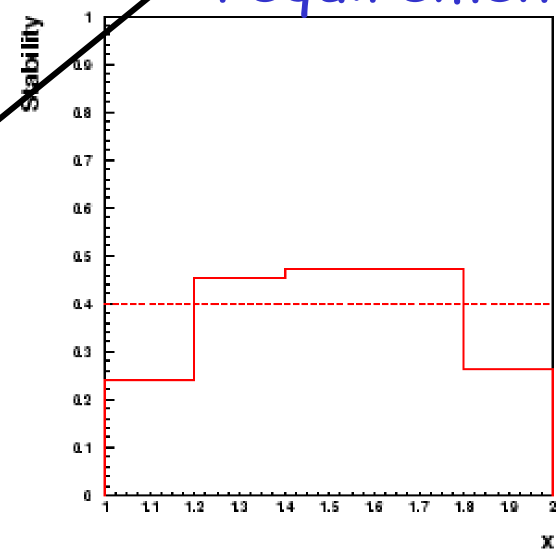
. . . $\text{Acceptance} = \frac{\text{Stability}}{\text{Purity}}$

# . . . Same data, bigger bin sizes

40% is a typical requirement




With bigger bin sizes, purity and stability improve whereas acceptance is unchanged.

Measurement region is determined by acceptance
. . . must be high and stable across bin

Binning choices usually defined by purity
e.g. require purity > 0.68 . . . i.e. bin width > resolution

Exception: if data statistics are limited