8.882 LHC Physics

Experimental Methods and Measurements

Efficiency and Acceptance

[Lecture 15, April 1, 2009]

Organization

Project 1

- completed – in process of reading and correcting, few comments so far == very good

Project 2

- due April 9 (the following week Thursday)

Project 3

- instructions are complete
- due May 2

Project 4 and Conference Session

- they are considered the final, contents to be defined

Organization

Our little conference

- one student one presentation

Proposed rough program

- “Overview – The LHC Project and Status”
- “Interesting Physics at the LHC”
- “A Charge Multiplicity Measurement”
- “Measurement of the Upsilon Cross Section”
- “Measurement of the *B* lifetime”
- “Standard Model Higgs Search: $H \rightarrow ZZ^*$”
- “Standard Model Higgs Search: $H \rightarrow WW^*$”
- “Standard Model Higgs Search: $H \rightarrow \tau\tau$”
- “Standard Model Higgs Search: $H \rightarrow \gamma\gamma$”
- ....

Physics Colloquium Series

'09 Spring

A Physics and Chemistry joint Colloquium

Thursday, April 2 at ***4:15 pm*** *in room* ***10-250***

George Whitesides

Harvard University

"Problems at the Interface between Physics, Chemistry, and Energy"

For a full listing of this semester's colloquia, please visit our website at web.mit.edu/physics

Lecture Outline

Efficiency and Acceptance

- introduction
- details about the Upsilon data
  - how where they triggered?
  - is all data good? goodrun lists!
- details of the Upsilon Monte Carlo sample
  - rough generator description
  - decaying Upsilons according to phase space
  - how to derive a relative and absolute efficiency?
- some systematic uncertainties

Introduction

Acceptance

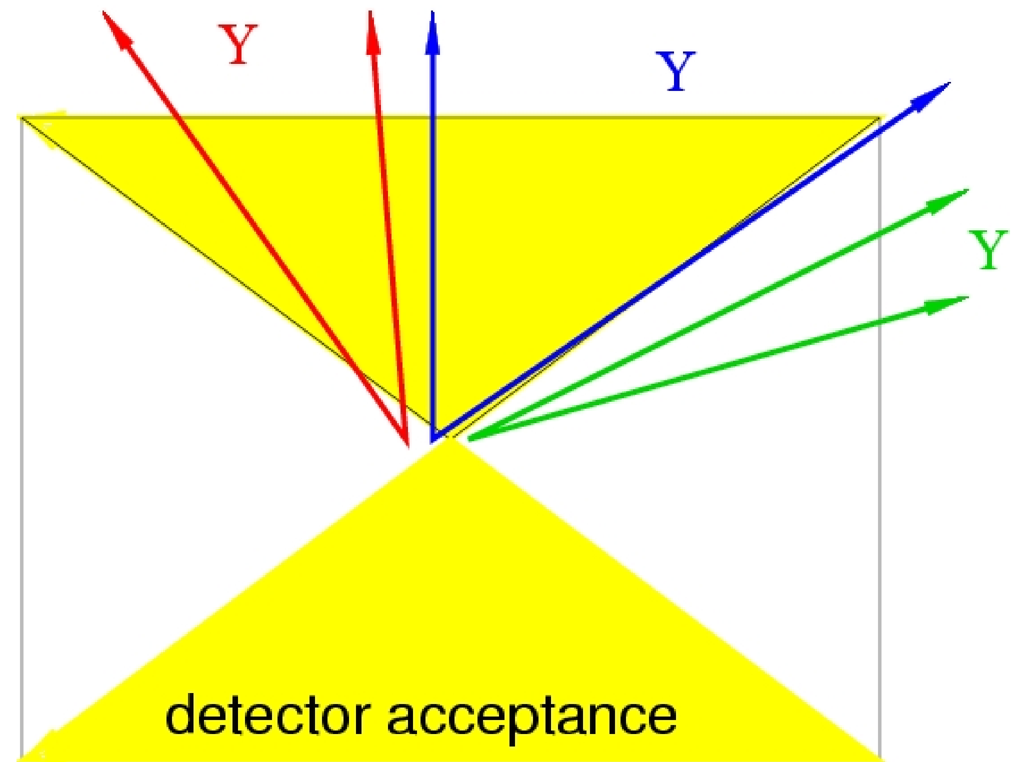
- refers to purely geometric fiducial volume of the detector

Efficiency

- refers to purely detector effectiveness in finding objects which have passed through the detector

In practice: ambiguous

- inside acceptance
- one leg mostly out of acceptance: efficiency will matter
- both legs outside of acceptance



Introduction

Cross section analysis

- cross section is given by

$$\sigma = \frac{N_{occurred}}{\mathcal{L}} = \frac{N_{observed}}{a\varepsilon\mathcal{L}}$$

Ingredients of the analysis

- L – integrated luminosity (provided to you)
- $N_{observed}$ – various methods exist (usually straight forward)
  - simple sideband subtraction, binned χ^2 or unbinned likelihood fits
- a – acceptance from the Monte Carlo
  - not clear how to get this without storing every event
  - also must be able to carefully calculate fiducial volume per muon
- ε – efficiency again from Monte Carlo
  - only possible to quote separately if acceptance known
- it makes sense to combine a and ε into one number
- often people refer to efficiency as the product: $a\ \varepsilon$

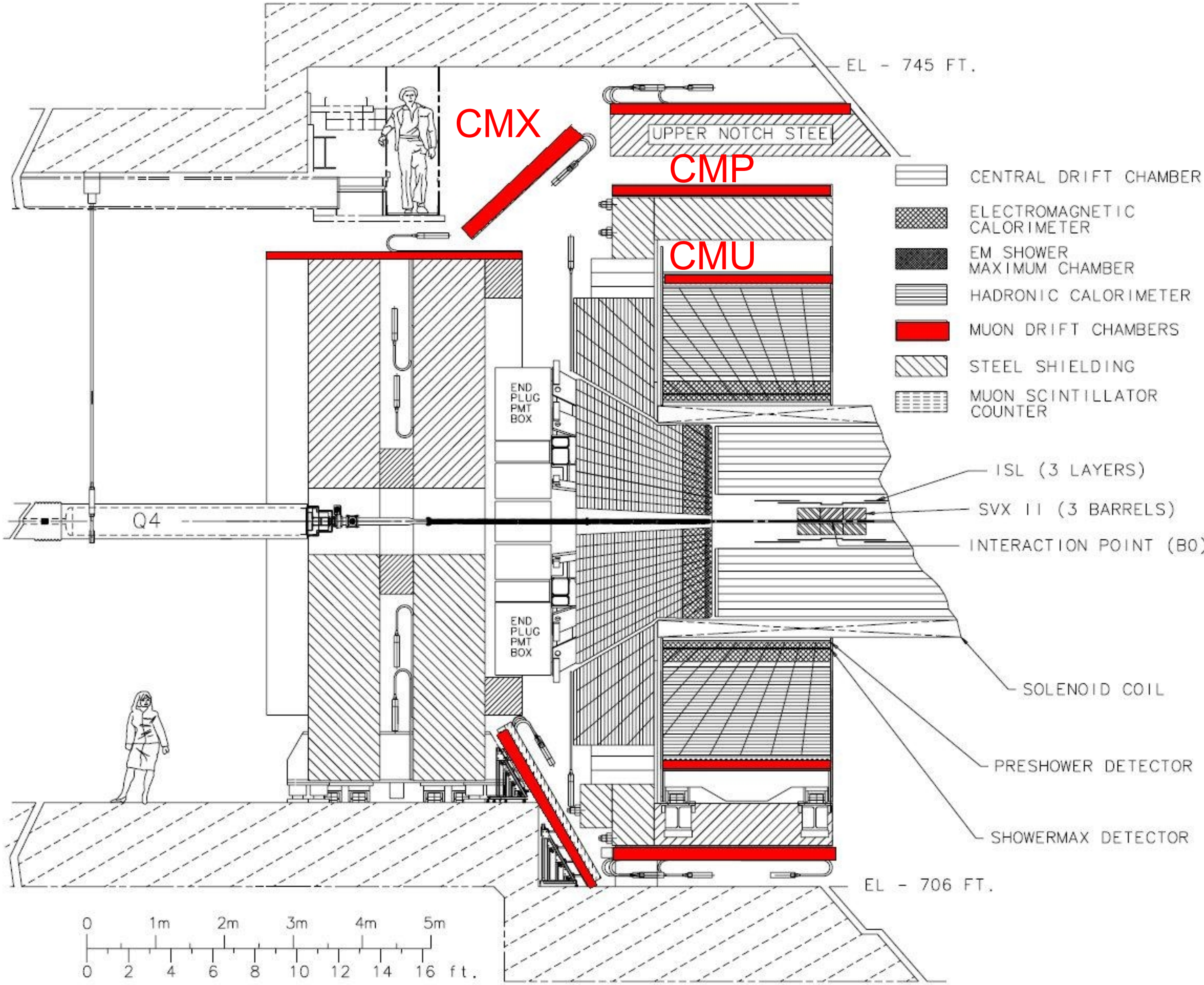
Introduction

Our efficiency ($a\ \varepsilon$) can be subdivided into

- trigger efficiency
  - level1
  - level2
  - level3
- ntuple
  - reconstruction
  - pre-selection efficiency
- your analysis
  - reconstruction and final selection efficiency
- deal with efficiencies of trigger and your analysis only
- ntuple related efficiency is implicitly taken care of: apply harder analysis requirements

CDF Muon Detection System

Muon detection starts at the muon chambers



CMU

- on HCAL
- $|\eta| < 0.6$

CMP

- add steel
- $|\eta| < 0.6$

CMX

- $0.6<|\eta|<1.0$

IMU

- $1.0<|\eta|<1.5$
- no trigger

Trigger Essentials

Trigger tables

- every event has to follow one or more exactly defined sequences through the level-1/level-2/level-3 system
- avoids all volunteers

Volunteer (*ex.* our upsilon sample)

- level1 requires CMUP muon
- level3 requires CMUP muon, higher quality data
- some CMUP muon identified at level3 but not at level1
- exact defined path avoids events without level1 CMUP
- if other level1 triggers (*ex.* track trigger) are considered additional events can show up, efficiency for those events is very difficult to determine

Trigger Essentials

Deadtime

- full detector read out takes a finite amount of time
- this time is larger then time between beam crossings
- this is also true for a pipelined trigger system which is called 'deadtimeless'
- during this time no additional events can be accepted
- this time is called deadtime
- if accept rate too high deadtime can seriously affect data taking: every event receives the same deadtime
- rule of thumb: deadtime should be kept well below 10%

Trigger Essentials

Prescales in CDF jargon

- too avoid too high accept rates certain triggers get prescaled: this means accepted events get rejected, at a given scale: the prescale
- prescale of 2 means: only every second event passing all trigger conditions gets accepted
- can be applied at all trigger level (usually level1, level2)
- simple prescale (PS): a fixed scale is applied throughout the data taking period to reject events
- dynamic prescale (DPS): the value of the prescale gets dynamically adjusted throughout the data taking period
  - inst. luminosity decreases more bandwidth is available
  - on a macroscopic timescale bandwidth is saturated
  - fully reproducible because average prescale per run can be calculated

Trigger Essentials

Prescales in CDF jargon, *continued*

- űber prescale (UPS): saturates the bandwidth at a microscopic level
  - in CDF this is done at the first trigger level
  - level2 trigger has four buffers
  - on average they are mostly full when running at a given rate with a given trigger table
  - at the microscopic level (396 ns, beam crossing) there must be instances where more then 1 buffer is free, even up to 4 can be free
  - űber prescale monitors activity in the buffers and will fill the buffers if there are free slots
  - problem: it is not possible to determine the effective scale anymore
  - trigger path with UPS needs to be separated so some analysis can ignore these events (*ex.* cross section analysis cannot use UPS)

Trigger Essentials

Access to trigger data and Monte Carlo in ntuple

- module: TPrereqFast
- specify names with exact or wild card matching
- each trigger level can be specified separately
- careful the Monte Carlo does not include a level3 trigger
- SetPrintLevel(-3) little output, for debugging go up to 1

Example

```
// Prerequisite module (default stuff)
gPrereq = new TPrereqFast();
gAna->AddModule(gPrereq,TStnModule::kFilter);

// Add the trigger name to consider (specifying level3 condition)
gPrereq->AddL3Name    ("UPSILON");
gPrereq->SetExactMatch(false);
```

Trigger Essentials

More complex examples with TPrereqFast

- first module: level1 trigger names to select
- second module: level3 trigger names to reject

```
// Prerequisite module to select
gPrereq = new TPrereqFast();
gAna->AddModule(gPrereq,TStnModule::kFilter);
// Add the trigger names to consider
gPrereq->AddL1Name    ("L1_TWO_CMU1.5");
gPrereq->AddL1Name    ("L1_CMU1.5_PT1.5_&_CMX1.5_PT2");
gPrereq->SetExactMatch(false);
// Prerequisite module to reject
gPrereqRej = new TPrereqFast();
gAna->AddModule(gPrereqRej,TStnModule::kVeto);
// Add the minimum trigger
gPrereqRej->AddL3Name    ("UPSILON_CMUP_CMU_DPS");
gPrereqRej->AddL3Name    ("UPSILON_CMUP_CMX_DPS");
gPrereqRej->SetExactMatch(false);
```

Upsilon Data Trigger - Early Data

Data are based on a dimuon trigger (run 138425)

- in CMU/CMP: UPSILON_CMUP_CMU:1
  - level1: L1_TWO_CMU1.5_PT1.5
  - level2: L2_AUTO_L1_TWO_CMU1.5_PT1.5
  - level3: L3_UPSILON_CMUPCMU
- in CMU/CMP/CMX: UPSILON_CMUP_CMX:1
  - level1: L1_CMU1.5_PT1.5_&_CMX1.5_PT2_PS1
  - level2: L2_AUTO_L1_CMU1.5_PT1.5_&_CMX1.5_PT2
  - level3: L3_UPSILON_CMUPCMX

Trigger summary

- no level1 pre-scale, auto accept level2
- level3 cuts on analysis quantities, should be fine after careful selection is applied

Upsilon Data Trigger – Later Data

Data are based on a di-muon trigger (run 238794)

- in CMU/CMP: UPSILON_CMUP_CMU_DPS:3
  - level1: L1_TWO_CMU1.5_PT1.5
  - level2: L2_CMUP1.5_PT3_&_CMU1.5_PT1.5_DPS
  - level3: L3_UPSILON_CMUPCMU
- in CMU/CMP/CMX: UPSILON_CMUP_CMX_DPS:3
  - level1: L1_CMU1.5_PT1.5_&_CMX1.5_PT2_CSX
  - level2: L2_CMUP1.5_PT3_&_CMX1.5_PT2_CSX
  - level3: L3_UPSILON_CMUPCMX

Trigger: level2 DPS – dynamic prescale

Trigger changed with time.... very careful here!

- play safe: only data with no (dynamic) prescales (1/3)
- alternatively: properly include (dynamic) prescales ....

Detector Status

Modern detectors are complex devices

- CDF has almost 1 million readout channels
- CMS one orders of magnitude more (mostly tracker)
- each channel needs power, cooling, safety systems *etc.*
- with many components involved, some failure is likely
- cannot effort to stop data taking with some part of the detector not 100% working

How do we know the detector worked?

- detector status is carefully monitored in the database
- *ex.* power of tracker is stored per power module *etc.*
- not completely automatic: shift crew classifies run status
- they might have realized something was wrong
- good run lists have to be “published”

Detector Status

Essential components in our analysis

- luminosity measurement
- muon trigger system
- muon detectors are essential
- tracker (COT very important, silicon better as well)

Goodrun list

- a data quality monitoring (DQM) group determines lists which are used by the entire experiment
- a lot of information has to be combined
  - database information
  - shifter information
  - dedicated offline analyses which test many different aspects of the functioning of the detector

Detector Status

Goodrun list for Upsilon analysis

- details to show amount of effort needed to determine the detector status
- this is the final SQL for the database query

```
SELECT RUNNUMBER, sum(LUM_INTEGRAL_OFFLINE),  sum(LUM_INTEGRAL_ONLINE)
FROM Run_Status, FILECATALOG.CDF2_RUNSECTIONS
WHERE
Run_Status.RUNNUMBER = FILECATALOG.CDF2_RUNSECTIONS.RUN_NUMBER
-- ------------------------------
-- online bits: trigger good run
-- ------------------------------
AND Run_Status.RUNCONTROL_STATUS = 1
AND Run_Status.SHIFTCREW_STATUS = 1
AND Run_Status.CLC_STATUS = 1
AND Run_Status.L1T_STATUS = 1
AND Run_Status.L2T_STATUS = 1
AND Run_Status.L3T_STATUS = 1
AND Run_Status.COT_OFFLINE = 1
AND Run_Status.COT_ONLINE = 1
```

Detector Status

```
-- ------------------------------
-- comment(--) the following lines
-- if you do not want Silicon
-- ------------------------------
AND  (Run_Status.SVX_OFFLINE != 0 OR
         ((Run_Status.SVX_OFFLINE Is Null) AND Run_Status.SVX_STATUS = 1))
         AND Run_Status.SVX_ONLINE = 1
-- ------------------------------
-- comment(--) the following lines
-- if you do not want muons
-- ------------------------------
AND (Run_Status.CMU_OFFLINE = 1 OR
         ((Run_Status.CMU_OFFLINE Is Null) AND Run_Status.CMU_STATUS = 1))
AND (Run_Status.CMP_OFFLINE = 1 OR
         ((Run_Status.CMP_OFFLINE Is Null) AND Run_Status.CMP_STATUS = 1))
AND (Run_Status.CMX_OFFLINE = 1 OR
         ((Run_Status.CMX_OFFLINE Is Null) AND Run_Status.CMX_STATUS = 1))
AND  (RUNNUMBER>150145) AND (RUNNUMBER<152636 OR
          RUNNUMBER>152945)
AND Run_Status.SVT_ONLINE = 1
AND Run_Status.SVT_OFFLINE != 0
AND Run_Status.CAL_OFFLINE = 1
AND Run_Status.CAL_ONLINE = 1
```

Detector Status

```
-- ---------------------------------------------
-- Specify run periods here
-- ---------------------------------------------
-- COT comprimised
AND (RUNNUMBER<=179056
        OR RUNNUMBER>=182843
        OR (RUNNUMBER>=180954 AND
              RUNNUMBER<=181190))
-- COT recovery
AND (RUNNUMBER<184062
       OR RUNNUMBER>184208)
GROUP BY RUNNUMBER
ORDER BY RUNNUMBER ASC
/
QUIT
```

Goodrun List in Your Analysis

Goodrun list

- attached to the TWiki description of the analysis
- download and put into directory where you run root (¼ of the runs are good)

Application

- need to run the TGoodRunFilter module
- see example

```
gGoodRunFilter = new TGoodRunFilter();
gAna->AddModule(gGoodRunFilter,TStnModule::kFilter);
gGoodRunFilter->SetPrintLevel(-4);
gGoodRunFilter->ApplyGoodRun (true);
gGoodRunFilter->DumpEvents   (false);
gGoodRunFilter->SetGoodRunFile("./goodrun.list");
```

Some Analysis Essentials

Cross section analysis

- is about counting and making sure your Monte Carlo really describes the data
- make sure all data analyzed which is included in lumi
- and the luminosity of course but that is given to you

Data Monte Carlo comparisons

- momentum and pseudorapidity distribution*
- opening angles*, *Δφ, Δη: Upsilon*
- track quantities: hits, momenta
- vertexing quantities: probability, L_{xy} *etc.*
- decay angle distribution (Upsilon might be polarized)

* cannot do this with our Monte Carlo

Some Analysis Essentials

Sanity checks

- cross section per run and per larger periods (or smaller units)
- check that *Upsilon* mass and width is stable (time wise)
- ....

Monte Carlo for Upsilons

Only one sample, *Upsilon(1S)*

- sample generated with
  - flat transverse momentum (0-200 GeV) and rapidity (-2,2)
  - generated total of 2 million events
- this implies that efficiencies have to be calculated per (transverse momentum,pseudorapidity) bin
- many Monte Carlo comparisons have to be done with some care
- Monte Carlo is mapped to the data in terms of the runs
  - good run list also has to be applied to the data
  - check that MC is complete
  - check that run numbers really match up
- no level3 trigger in MC

Conclusion

Acceptance and efficiency

- geometric detector fiduciality defined as acceptance, a
- detector efficiency, ε, for particle passing through
- mostly use 'efficiency' as, $a\ \varepsilon$

Upsilon analysis

- cross section, technically, a simple analysis
- the uncertainty is dominated by the 6% luminosity uncertainty
- requires a lot of checking/bookkeeping because missed or duplicated events immediately cause an error
- special Monte Carlo flat generation needs some thought to be properly applied

Next Lecture

High energy physics overview

- B physics
- Standard Model physics
  - QCD, electroweak, top, SM Higgs
- Beyond the Standard Model
  - SUSY: Higgses and all the other new particles: neutralinos, charginos, squarks, sleptons, winos, zinos .....
  - little Higgses
  - extra dimensions
  - technicolor
  - exotic stuff: heavy leptons, monopoles, ...