road-to-pi

robertej17

June 2020

Chapter 1

Personal To Do

1.1 Introduction

Need to make plan! Make tentative outline of how it will go to get a cross section Define issues, etc Push forward It's a question of the uncertainties What do we need to get cross sections by the end of the year using some subset of the data and some subset of the cross sections

Is it possible to show the same type graphs, but as a function of the t variable. Also, even more interesting would be, a two dimensional graphs showing the distributions of events vs Q2 and t for a given W, e.g. 3 GeV, for both the forward and central detectors.

Run through the 174 inbending runs

How does analysis change when using custom cuts instead of normal event builder cuts?

Make previous plots all broken up on fd and cd, and without using proton momentum

Make same plot on fd on cd
test across all 6 sectors
Show missing mass plots
Plot train info from pid

Plot based for pid says it's a proton at 5 gev, see that pod is still working there, cut shouldnt be too narrow, might have kinematic constraints

Compare with and without proton momentum - tag it, but dont use it to measure t and phi. Different systematics, so compare proton sytematics cd to pi systematics fd

1.2 Particle ID

Using Event Builder Cuts:

1.3 Uncertainties

1.4 Binning

1.5 Luminosity

1.6 Simulation

Chapter 2

Motivation and Background

Chapter 3

CLAS12

Chapter 4

Data Sets

```
inbending: /cache/clas12/rg-a/production/recon/fall2018/torus-1/pass1/v0/dst/train/skim8/
outbending: /volatile/clas12/rg-a/production/recon/fall2018/torus+1/pass1/v0/dst/train/skim4/
Bobby, FX has eppi0 train data, which can significantly reduce code run time at /work/clas12/fxgirod/eppi0/
```

Outbending cooking is ongoing, so there aren't many files yet.

Chapter 5

Particle Identification

Chapter 6

Coding

6.1 Coding notes

Syntatic Sugar

6.2 How to Enable Multi-Threading

6.2.1 1

At ifarm, check if you have /.groovy directory Otherwise create one.

6.2.2 2

Download Andrey's sugar.

```
cd ~/.groovy
wget https://github.com/drewkenjo/dst_monitoring/releases/download/v0/sugar.jar
```

6.2.3 3

The standard way to access coatjava at ifarm is adding following commands at ~/.cshrc or ~/.bashrc depending on which shell you use.

```
source /group/clas12/packages/setup.csh
module load clas12/pro
```

However, this doesn't allow you to edit bin files. Simply copy run-groovy at any directory with your permission, say ~/.groovy, i.e.

```
cp $COATJAVA/bin/run-groovy ~/.groovy/
```

edit ~/.groovy/run-groovy 's line 59 from -Xmx2048m to -Xmx4096m. This will increase memory that can be used in your ifarm.

6.2.4 4

I guess you have your own standalone analysis script that runs as a main file. The idea is to convert this as a class, and call from other main file. This is not necessary for parallel computing, but this helps to run multiple analysis at one time and to maintain scripts tidy.

Actually, groovy supports usage of both class and main likewise python's ' if __name__ == "main" '. You can put a closure with the same name of the class. See here

6.2.5 5

I have standalone analysis script epg.groovy (, which is main itself), here The code's structure is like this:
line 1–17: import libraries
line 18–127: defining histograms
line 128–319: main loop to read files (ignore line 131–154 because nowadays Nick is placing inbending and outbending in separate directories) line 320–384: saving histograms

I converted that to a class file dvcs.groovy, here Here're how-to's.
1) Add package name at line 1
2) Add import java.util.concurrent.ConcurrentHashMap at line 19
3) class name at line 22. class dvcs{ for this case
4) Change histograms to be used for ConcurrentHashMap. Please compare line 24–60 of dvcs.groovy to 18–127 of epg.groovy.
5) add def processEvent(event){ before the loop starts (line 80). The loop contents don't have to differ from your previous main file. 6) Remove all histogram saving steps (e.g. line 320-384 of my epg.groovy)
7) Instead, you can set up your histogram's name and directory directly when you fill the histogram, e.g.) hists.computeIfAbsent("/epg/elec_polar_rate).fill(Math.toDegrees(ele.theta())) This will save histograms at /epg/elec_polar_sec1–/epg/elec_polar_sec6
Andrey's sugar will deal with directory structures so that you don't have to "out.mkdir(dir); out.cd(dir); out.addDataSet(hist)" manually.

By the way, with Andrey's sugar, you can now add or subtract LorentzVector like def VmissG = beam + target - ele - pro at line 109.

6.2.6 6

Now download this
Change line 22 of run.groovy to your own class name.
To run this,

run-groovy sangbaek/run.groovy 'find /cache/clas12/rg-a/production/recon/fall2018/torus-1/pass1/v0/dst/train/skim8/ -name "*.hipo"'

6.2.7 7

As for 3, I forgot to tell you that line 5–7 needs to be changed as

```
CLARA_HOME=$COATJAVA/bin/..
CLAS12DIR=$COATJAVA/bin/..
CLAS12DIR=$COATJAVA/bin/.. ; export CLAS12DIR
.
```

To use your edited run-groovy, there are two ways that I know of.

First way is to include following line at ~/.cshrc.
alias run-groovy /home/sangbaek/.groovy/run-groovy

Second one is to include following shebang line at your groovy scripts like below.

```
#!/home/sangbaek/.groovy/run-groovy\\
```

I find the first one easier.

Chapter 7

Analysis Method

For each kinematic bin the differential cross section can be written as:

$$\sigma = \frac{N_{meas}}{L\epsilon}\frac{1}{\delta} \tag{7.1}$$

Where $\frac{N_{meas}}{L}$ is the number of events from experiment normalized by the integrated luminosity before acceptance and radiatvie corrections. $\epsilon = \frac{N_{rec}^{RAD}}{N_{gen}^{RAD}}$ is the acceptance correction and δ is the radiative correction.

δ can be obtained by using the following:

$$\delta = \frac{N_{gen}^{RAD}}{N_{gen}^{NORAD}} \tag{7.2}$$

luminosity = avogadros constant times the length of the target times teh density of the target (liquid hydrogen), times the ttoal collected charge in the farday cup, divided by the electron charge.

$$L = \frac{N_A l \rho Q_{FCUP}}{e} \tag{7.3}$$

.2 Analytical Structure fu diative corrections

$$\frac{d\sigma(ep \to ep\pi^0)}{dQ^2 dx_B dt d\phi} = \Gamma(Q^2, x_B)\frac{1}{2\pi}[\sigma_T + \epsilon\sigma_L + \epsilon\sigma$$

$$\Gamma(Q^2, x_B) = \frac{\alpha y^2(1$$

$$\epsilon = \frac{1 - y - \gamma}{1 - y + y^2/2}$$

$$y = \frac{Q^2}{2M_p x_B E_{bea}}$$

$$\gamma = \frac{(y x_B M_p)^2}{Q^2}$$

The model:

$$\sigma_T = \sigma^T e^{B_T(x_B)t}/(Q^2 + M^2)^n, B_T(x_B)$$

$$\sigma_L = \sigma^L Q^2 e^{B_L(x_B)t}/(Q^2 + M^2)^n, B_L(x_B$$