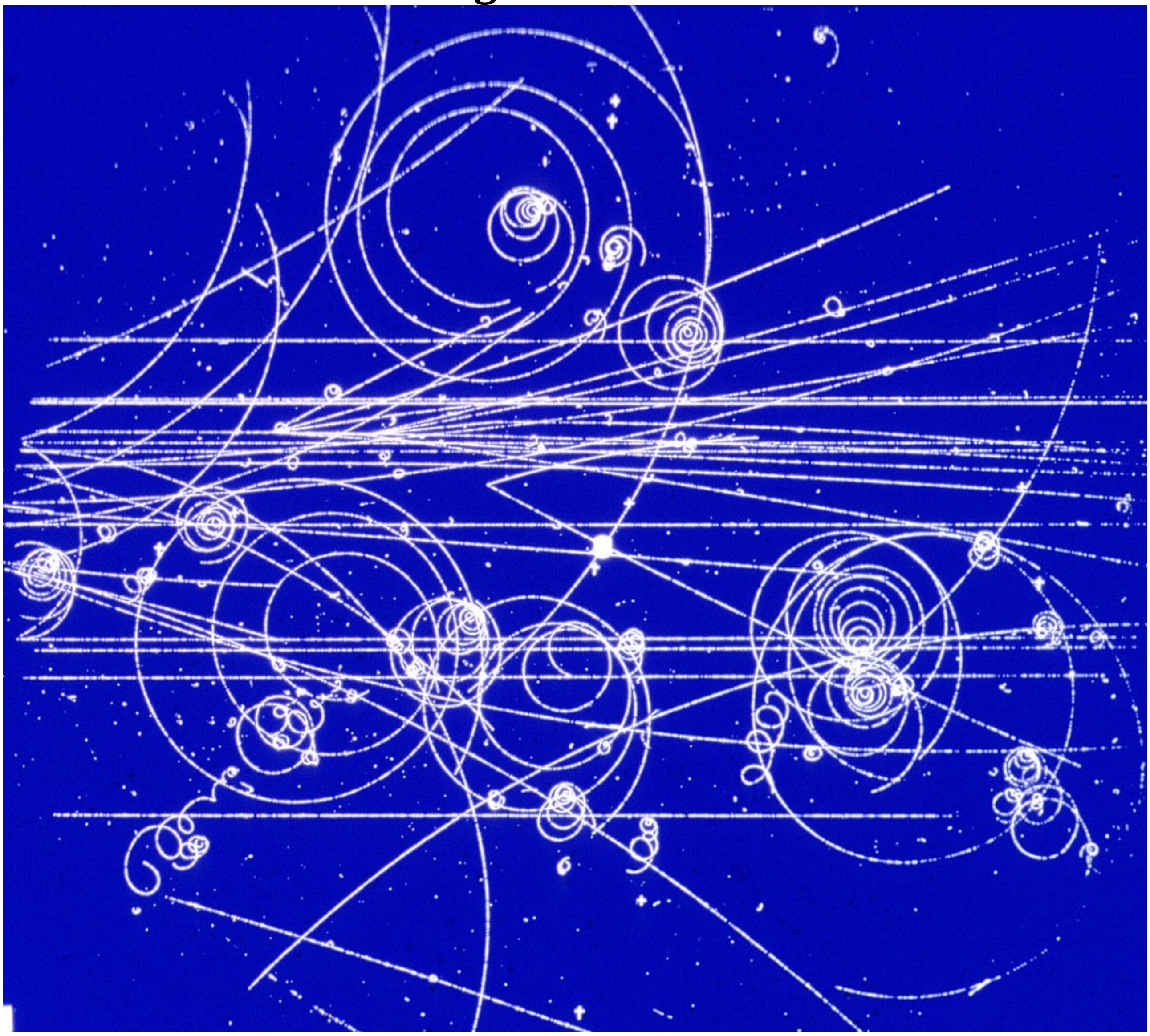
KVPY SUMMER PROJECT

TOPIC: Scattering And Simulation Softwares



Tathagata Karmakar
Physics, IIT Kanpur
KVPY Reg.No- SB1312043

This project was done under Professor Pankaj Jain, Department of Physics, IIT Kanpur

Table of Contents

Coverpageimage source:
https://www.google.co.in/search?q=LHC&espv=2&biw=1366&bih=681&source=lnms&tbm=isch&sa=X&ei=mXiZVeilFo-9uAT30oLIBg&ved=0CAcQ_AUoAg#tbm=isch&q=particle+collision+wallpaper&imgrc=N7NkHDGUQLQjjM%3A

KVPY SUMMER PROJECT

::

Aim: Learning basics of quantum scattering and knowing how to use various softwares for High Energy Physics.

Motivation: As a first year undergraduate student my primary motivation was exploring what is there in expeimental high energy physics. I wanted to gather knowledge about what happens in the collider and how we predict various properties of particles.

:::

Project work:-

Learning some theory: I was not exposed to Quantum Mechanics before. So, I began to learn basics of it .

<QM>

I learnt

- Frank and Hertz experiment, Young's Double-Slit experiment[1].
- Heisenberg's uncertainty principle, principle of superposition[2].
- Schrodinger's equation[3].
- Dirac notations ('Bra' and 'ket') [4].
- Basics of Probability theory[5] and Fourier series[6].

<Particle Physics>

As it was needed in the project, I studied standard model[7] and historical account of reaching the model[8]. I studied in brief some famous groundbreaking experiments done to discover new particles.

- Discovery of pion and muon from experiments with cosmic rays[8].
- Discovery of first antiparticle positron by Anderson followed by other antiparticle discoveries[8].
- Experiment by Cowan and Reines at Savannah River nuclear reactor in South Carolina (1950) and confirmation of neutrino's existence[8].
- Discovery of strange particles[8].

I also studied

- Klein-Gordon equation[7].
- Dirac sea and how this picture (sea of electrons having negative energy states) were used as theoritical prediction of antiparticles[8,9].
- Quantum scattering, differential cross section, total cross section and its

physical significance[10].
- Rapidity and pseudo rapidity[11,12].
- How to use Feynman diagrams to represent various processes[8,13].
- How to calculate invariant amplitude for scattering of two spinless particles from it[8,9].
- How Fermi's golden rule gives us decay rates and scattering cross sections[8].

Overview of Work: Along with reading theory, I learnt c++[14]. I learnt basics of data analysis framework root developed by Cern[15]. In ROOT, I learnt

- How to draw graphs for various functions.
- How to draw histograms with and without error bars.
- How to create and use tree[16].
- How to deal with files in root.
- Use of class TVector3 and TVorentzVector.
- How to compile and execute .C files in root (using .x and .L command).

After learning root I was given a file data_log_udp_n0017.dat to

- Create histograms for the data in each column and save them in a .root file.
- Reading histograms from .root files and save mean, rms, number of entries of each histograms in a .txt file.
- Read columns from the .dat file and add them as branches of a tree and save the tree as another .root file.
- Create histograms after reading the tree.

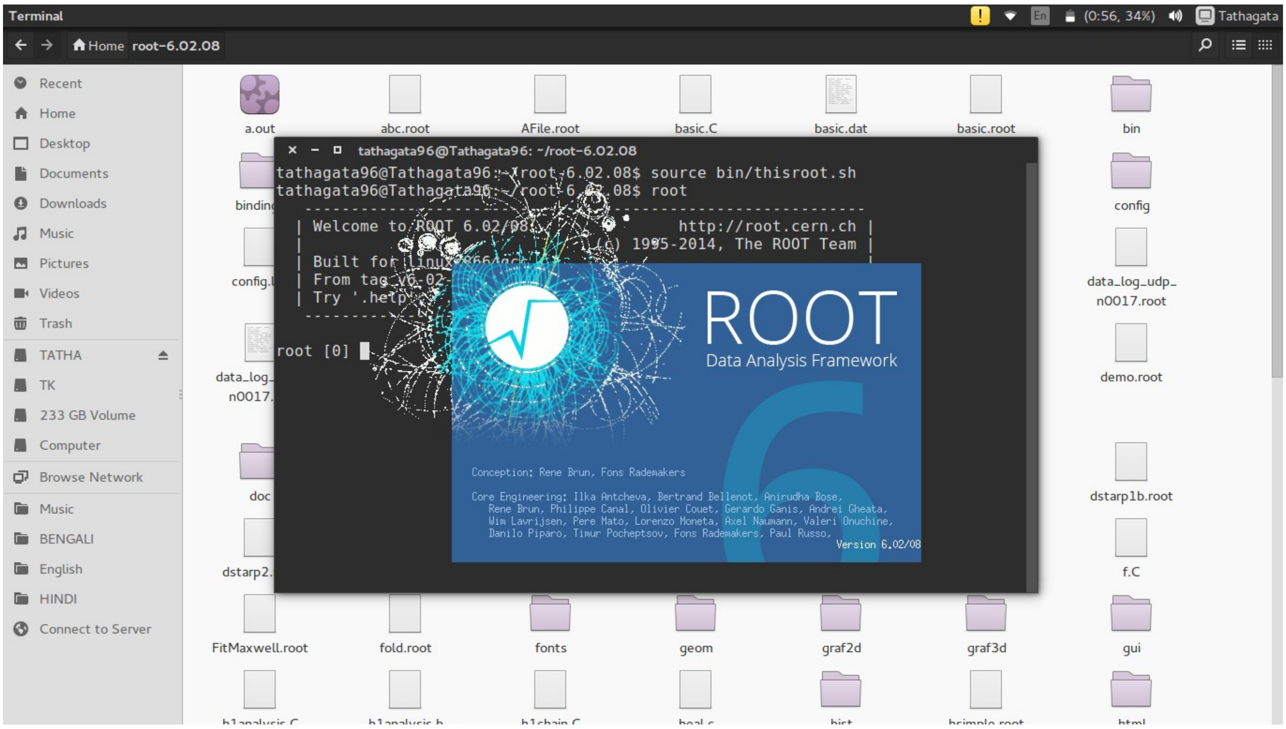
Then I learnt basics of MCFM(Monte Carlo for FeMtobarn processes) and used it to get cross-section of scattering of two protons.

Detailed Work:

<Using ROOT>

ROOT is a data analysis framework which provides us with various tools particularly useful for physics(like drawing graphs, histograms, vector transformations, geometry).

After successful installation (see https://root.cern.ch/drupal/content/installing-root-source) root can be opened through terminal



Running root

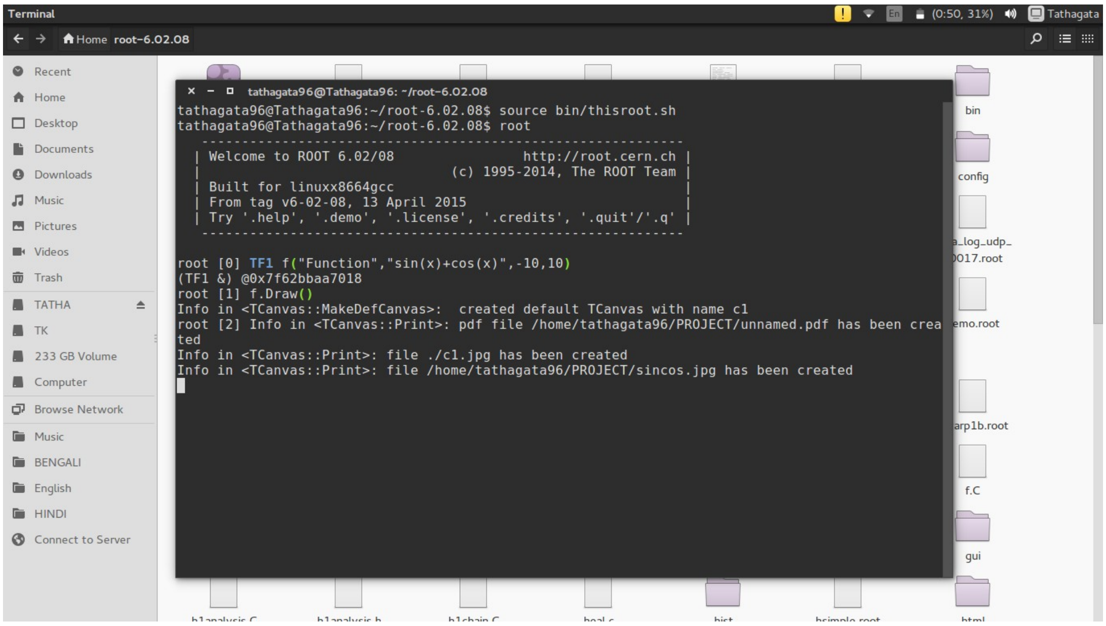
In ROOT , I learnt

1) Drawing graph of a function

This can be done by typing command

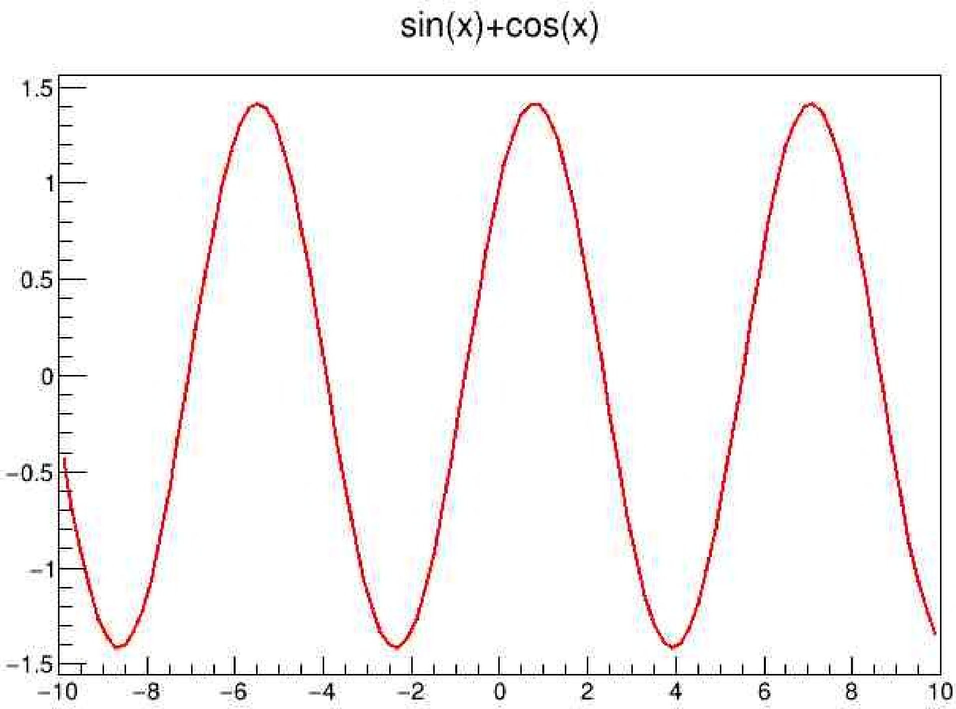
TF1 f(“Function_name”, “f(x)”, xmin, xmax)

and then with the command f.Draw() as shown



Drawing a graph: root prompt command

This produces the graph below



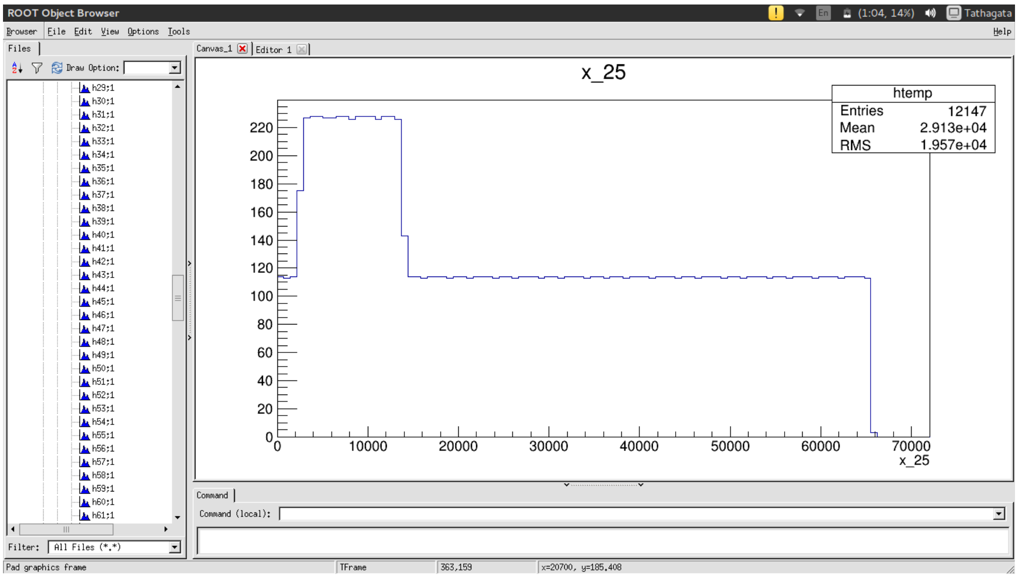
Graph of sin(x)+cos(x) in range [-10,10] as produced through Root

There are various modes of drawing graph. Polar graphs can be drawn (using TGraphPolar) , 2D and 3D graphs can be drawn.

2) Drawing histograms in root

I learnt to draw 1D, 2D and 3D histograms in root using TH1, TH2, TH3 classes.

This can be done by constructor
TH1F h("histogram_name", "Title", number of bins, xmax, xmin)
Histograms can be filled by h.Fill(x,weight) method. It can be drawn on canvas by the method h.Draw()
Below one such histogram is shown.



Histogram in ROOT

Besides these, I learnt
3) Creating a tree using TTree t("name","title") constructor.

A tree is like a data file which is more user friendly & less memory taking. We can store values of a variable in a branch of tree. Once we have the tree we can extract all the information, display the histos ,plot graphs between two or three variables and many more things.

Branch can be added to a tree using
t.Branch("Variable_name", variable address, "Variable/Type")
method. Filling the tree using t.Fill() method. As in tree file size gets compressed (compared to .dat) and it autometically generates histograms, it is advantageous to use tree.

4) Using classes TVector3 and TlorentzVector to do calculations(such as components in a rotated coordinate) concerning three vectors and four vectors easily .

5) Compilation of c++ code using root:
The c++ code should be saved as filename.C
Its main function should be named filename()

Then we can execute the program using command .x Filename.C and we can compile it using .L filename.C

6)In root files can be saved with extension .root
Files can be created using constructor TFile f("filename.root", "recreate");
Files can be opened using TFile f("filename.root") constructor.

After learning basics of Root , I was given a file data_log_udp_n0017.dat (filled with several columns of data) to do the following works with it

(a) Importing the data file

(b) Printing all the columns/rows of this file into the terminal

(c) Creating Histograms for each column with proper labels(axes, names, etc)

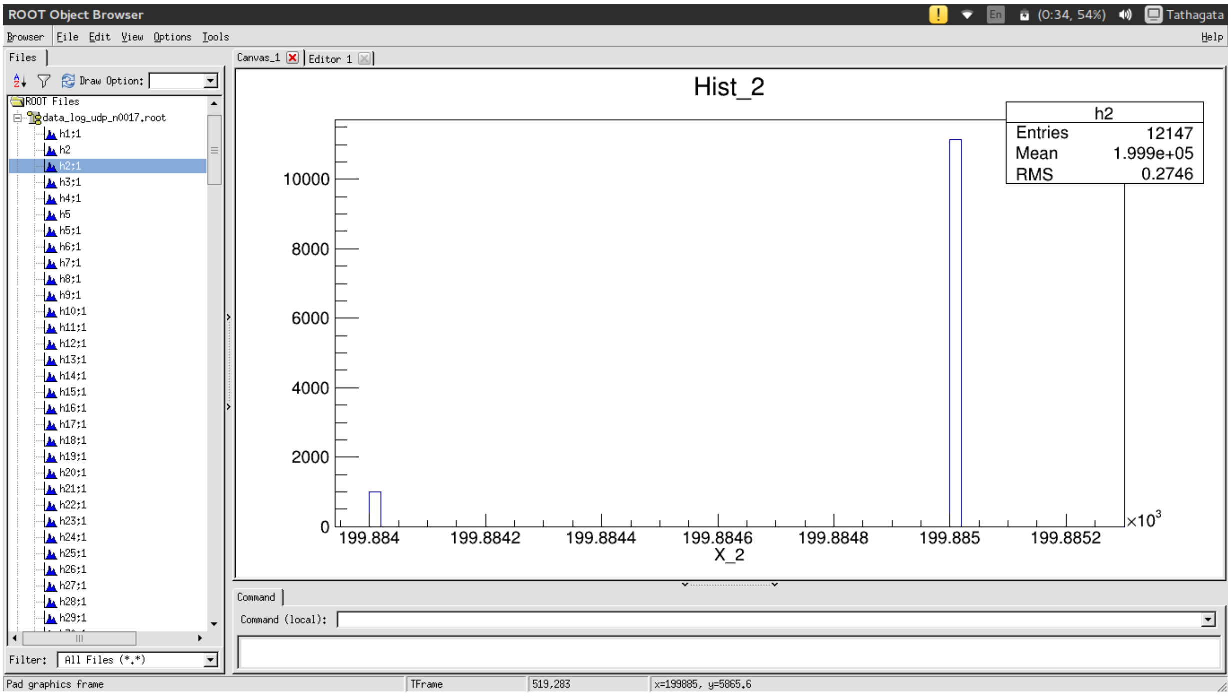
(d) Storing these Histograms in a .root file

I determined the number of columns in the files using no_columns.c program and found it to be 62. Then I wrote the program data_log_udp_n0017.C which does all the above four tasks. The histograms were saved in file data_log_udp_n0017.root

All these files can be found in the link
https://drive.google.com/folderview?id=0B3yQpROHCwaRfm1RY0NUSVUzUzhnYkdwNTJUOFFMZERYZDItVF9CNkFjb0ttTl9MQzg0Z1U&usp=drive_web

Files can be found in the folder named Data_File.

One of the 62 histograms is shown on the next page.



Histogram

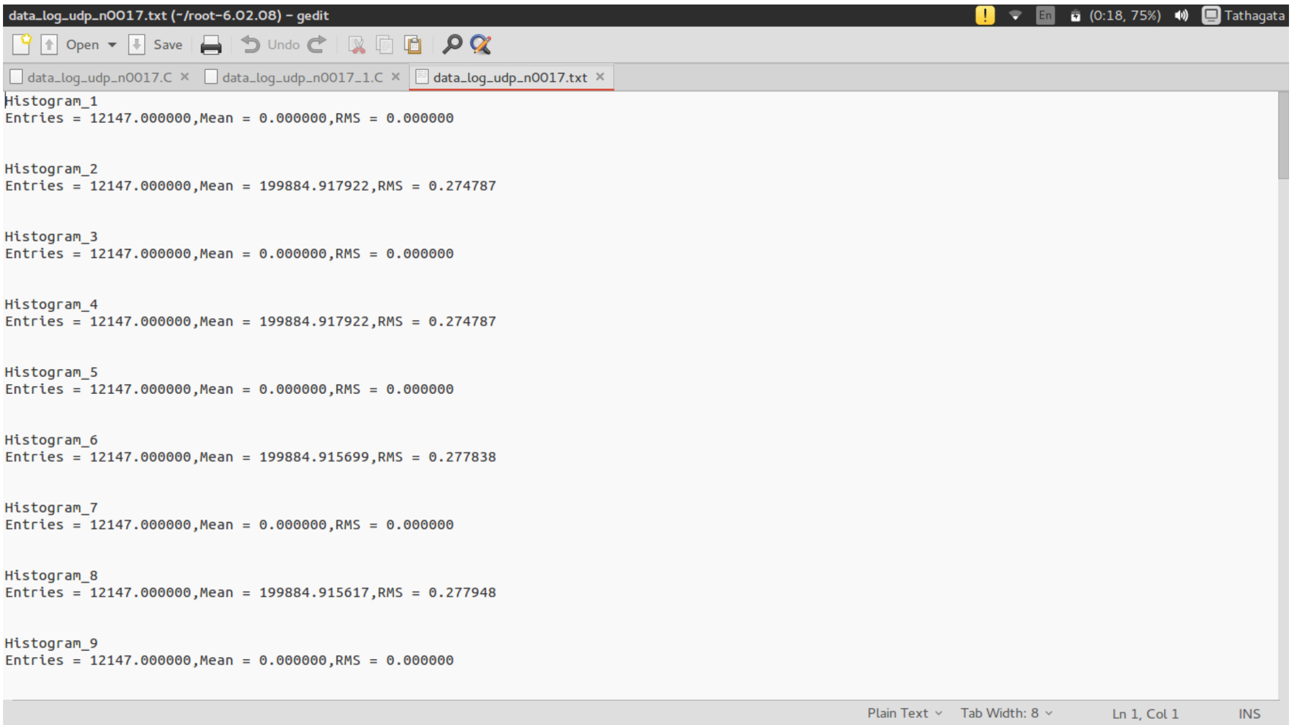
Next steps were:

(a) Importing the previously created .root file

(b) Reading all the histos and their statistics(mean,rms,entry)

(c) Putting these statistics in a .txt file for later use.

The code written to achieve these is data_log_udp_n0017_1.C and again it is shared in the folder mentioned above. New file data_log_udp_n0017.txt was created.



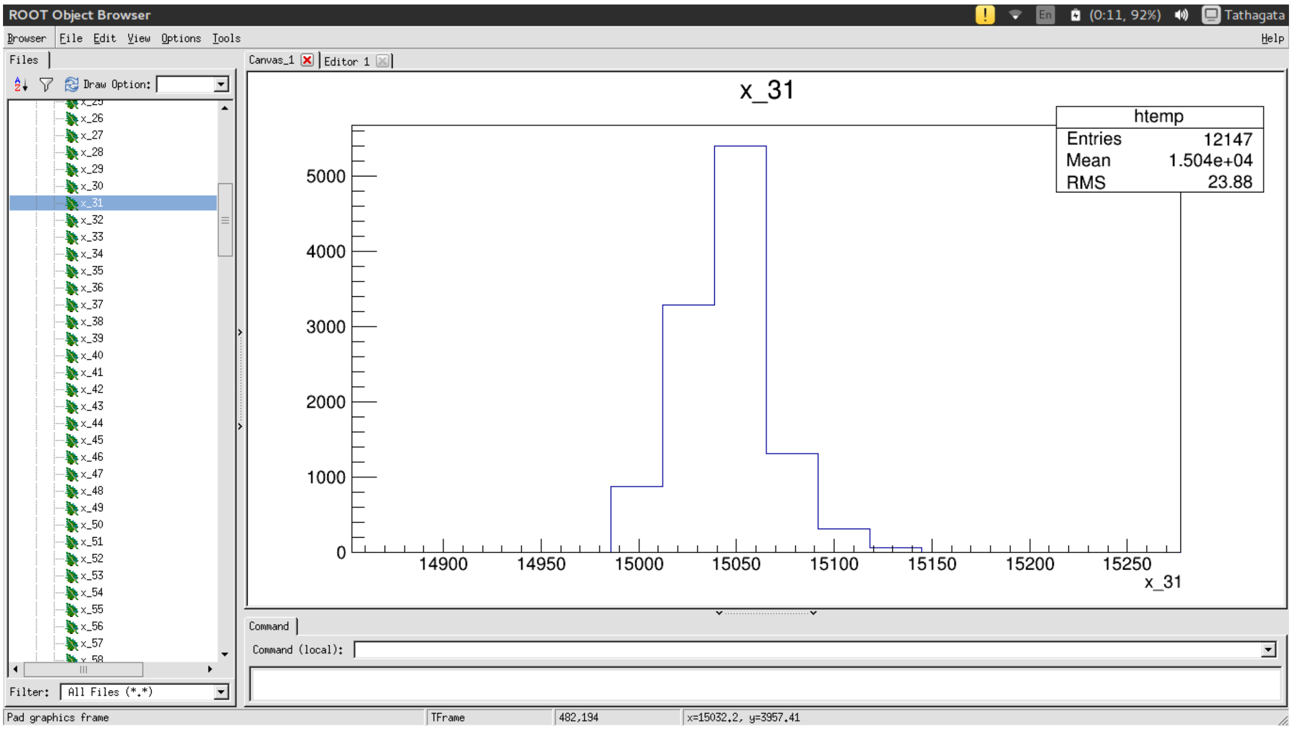

```
Histogram_1
Entries = 12147.000000,Mean = 0.000000,RMS = 0.000000


Histogram_2
Entries = 12147.000000,Mean = 199884.917922,RMS = 0.274787


Histogram_3
Entries = 12147.000000,Mean = 0.000000,RMS = 0.000000


Histogram_4
Entries = 12147.000000,Mean = 199884.917922,RMS = 0.274787


Histogram_5
Entries = 12147.000000,Mean = 0.000000,RMS = 0.000000


Histogram_6
Entries = 12147.000000,Mean = 199884.915699,RMS = 0.277838


Histogram_7
Entries = 12147.000000,Mean = 0.000000,RMS = 0.000000


Histogram_8
Entries = 12147.000000,Mean = 199884.915617,RMS = 0.277948


Histogram_9
Entries = 12147.000000,Mean = 0.000000,RMS = 0.000000
```

The .txt file where statistics of histograms are saved

Then next six steps were:
(A) Creating skeleton of tree
(B) Opening the .dat file and filling the tree
(C) Printing and writing the tree
(D) Importing the file containing tree and reading the tree
(E) Creating skeleton of histograms
(F) Filling histograms from the tree

To do first three steps, data_log_udp_n0017_tree.C was written. For the last three steps data_log_udp_n0017_treeread.C was written. The files are available in the link.

https://drive.google.com/open?id=0B3yQpROHCwaRfmZvNnM0OERselZ0ZXU2aUtqX1hkU1dJUVBaZlFjUGJMZ0NwMHBEM1RXNGM



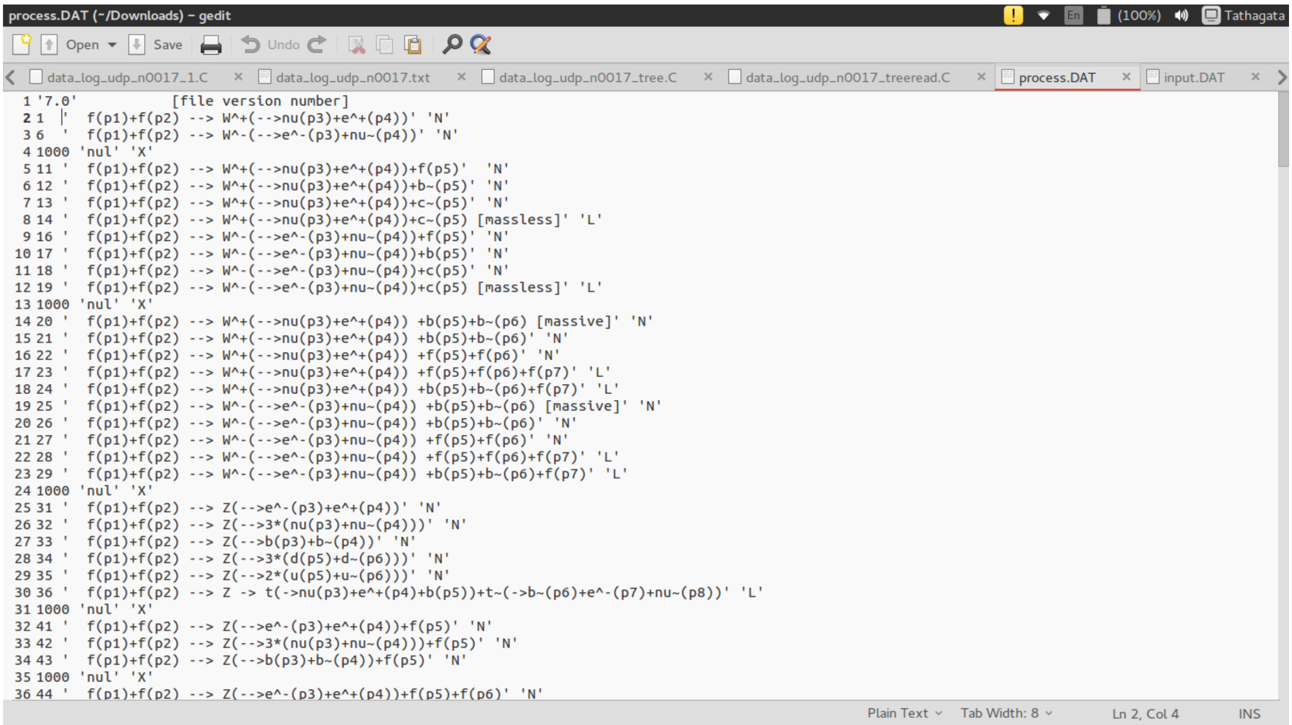
A Histogram drawn from a branch of a Tree in ROOT

<Using MCFM>

After getting used to with root I started to learn MCFM software(Monte Carlo for FeMtobarn Processes). This software was reading input parameters from a file input.dat and it was generating various histograms for a particular scattering process(processes are given in a file process.dat) and it was calculating total cross section. It was giving .C (to be executed in root) , .dat and .top files as output.

After installing MCFM (see http://mcfm.fnal.gov/) the above two .dat files can be found in MCFM/Bin folder.

process.dat contains lists of several scattering process. There are only two incoming



```
'7.0'            [file version number]
1  '  f(p1)+f(p2) --> W^+(-->nu(p3)+e^+(p4))' 'N'
6  '  f(p1)+f(p2) --> W^-(-->e^-(p3)+nu~(p4))' 'N'
1000 'nul' 'X'
11 '  f(p1)+f(p2) --> W^+(-->nu(p3)+e^+(p4))+f(p5)'  'N'
12 '  f(p1)+f(p2) --> W^+(-->nu(p3)+e^+(p4))+b~(p5)' 'N'
13 '  f(p1)+f(p2) --> W^+(-->nu(p3)+e^+(p4))+c~(p5)' 'N'
14 '  f(p1)+f(p2) --> W^+(-->nu(p3)+e^+(p4))+c~(p5) [massless]' 'L'
16 '  f(p1)+f(p2) --> W^-(-->e^-(p3)+nu~(p4))+f(p5)' 'N'
17 '  f(p1)+f(p2) --> W^-(-->e^-(p3)+nu~(p4))+b(p5)' 'N'
18 '  f(p1)+f(p2) --> W^-(-->e^-(p3)+nu~(p4))+c(p5)' 'N'
19 '  f(p1)+f(p2) --> W^-(-->e^-(p3)+nu~(p4))+c(p5) [massless]' 'L'
1000 'nul' 'X'
20 '  f(p1)+f(p2) --> W^+(-->nu(p3)+e^+(p4)) +b(p5)+b~(p6) [massive]' 'N'
21 '  f(p1)+f(p2) --> W^+(-->nu(p3)+e^+(p4)) +b(p5)+b~(p6)' 'N'
22 '  f(p1)+f(p2) --> W^+(-->nu(p3)+e^+(p4)) +f(p5)+f(p6)' 'N'
23 '  f(p1)+f(p2) --> W^+(-->nu(p3)+e^+(p4)) +f(p5)+f(p6)+f(p7)' 'L'
24 '  f(p1)+f(p2) --> W^+(-->nu(p3)+e^+(p4)) +b(p5)+b~(p6)+f(p7)' 'L'
25 '  f(p1)+f(p2) --> W^-(-->e^-(p3)+nu~(p4)) +b(p5)+b~(p6) [massive]' 'N'
26 '  f(p1)+f(p2) --> W^-(-->e^-(p3)+nu~(p4)) +b(p5)+b~(p6)' 'N'
27 '  f(p1)+f(p2) --> W^-(-->e^-(p3)+nu~(p4)) +f(p5)+f(p6)' 'N'
28 '  f(p1)+f(p2) --> W^-(-->e^-(p3)+nu~(p4)) +f(p5)+f(p6)+f(p7)' 'L'
29 '  f(p1)+f(p2) --> W^-(-->e^-(p3)+nu~(p4)) +b(p5)+b~(p6)+f(p7)' 'L'
1000 'nul' 'X'
31 '  f(p1)+f(p2) --> Z(-->e^-(p3)+e^+(p4))' 'N'
32 '  f(p1)+f(p2) --> Z(-->3*(nu(p3)+nu~(p4)))' 'N'
33 '  f(p1)+f(p2) --> Z(-->b(p3)+b~(p4))' 'N'
34 '  f(p1)+f(p2) --> Z(-->3*(d(p5)+d~(p6)))' 'N'
35 '  f(p1)+f(p2) --> Z(-->2*(u(p5)+u~(p6)))' 'N'
36 '  f(p1)+f(p2) --> Z -> t(->nu(p3)+e^+(p4)+b(p5))+t~(->b~(p6)+e^-(p7)+nu~(p8))' 'L'
1000 'nul' 'X'
41 '  f(p1)+f(p2) --> Z(-->e^-(p3)+e^+(p4))+f(p5)' 'N'
42 '  f(p1)+f(p2) --> Z(-->3*(nu(p3)+nu~(p4)))+f(p5)' 'N'
43 '  f(p1)+f(p2) --> Z(-->b(p3)+b~(p4))+f(p5)' 'N'
1000 'nul' 'X'
44 '  f(p1)+f(p2) --> Z(-->e^-(p3)+e^+(p4))+f(p5)+f(p6)' 'N'
```

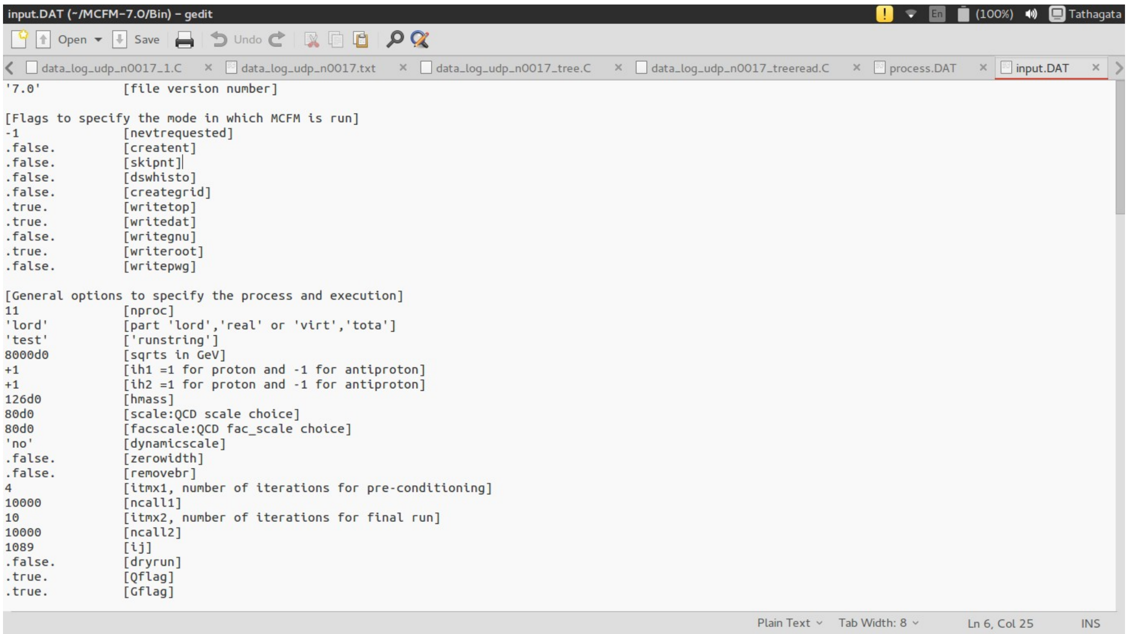
process.DAT

particles. We can chose them to be proton-proton (This collision is done in LHC), proton-antiproton, antiproton-antiproton. These two particles have momentum p1 and p2. The various processes correspond to various possibilities of outcomes. For example process 1 is given as

' f(p1)+f(p2) --> W^+(-->nu(p3)+e^+(p4))' 'N'

Here in this process after collision of two protons **W^+** boson is created. Which decays leptonically into ν_e (nutrino) (momentum p3) and e^+ (positron) (momentum p4). 'N' implies next to leading order process.

Now input parameters can be changed in the file input.dat .




```
'7.0'           [file version number]

[Flags to specify the mode in which MCFM is run]
-1              [nevtrequested]
.false.         [creatent]
.false.         [skipnt]
.false.         [dswhisto]
.false.         [creategrid]
.true.          [writetop]
.true.          [writedat]
.false.         [writegnu]
.true.          [writeroot]
.false.         [writepwg]

[General options to specify the process and execution]
11              [nproc]
'lord'          [part 'lord','real' or 'virt','tota']
'test'          ['runstring']
8000d0          [sqrts in GeV]
+1              [ih1 =1 for proton and -1 for antiproton]
+1              [ih2 =1 for proton and -1 for antiproton]
126d0           [hmass]
80d0            [scale:QCD scale choice]
80d0            [facscale:QCD fac_scale choice]
'no'            [dynamicscale]
.false.         [zerowidth]
.false.         [removebr]
4               [itmx1, number of iterations for pre-conditioning]
10000           [ncall1]
10              [itmx2, number of iterations for final run]
10000           [ncall2]
1089            [ij]
.false.         [dryrun]
.true.          [Qflag]
.true.          [Gflag]
```

input.DAT

For example, chosing the logical write parameters (writetop, writeroot etc) we can select whether particular type of file is to be created (.top, .C etc) or not depending on whether .true. or .false. is selected.
The parameter [nproc] selects particular process (in the picture , it is 11th).
[ih1] and [ih2] is selected for identity of the incoming particles (1=proton and -1=antiproton).

Since process 1 is least complicated I was told to run that process. But a there was a problem:
We(me and Mr. Ravindra Kr Verma, research scholar under my mentor) found that the generated .C file was buggy and it gave compile-time error. For example, it contained multiple declarations of 'int xbin'.

To fix the bug we needed to change some of its source code. As the software was written in fortran, I learnt basics of fortran[17]. As we needed to find keywords inside all .f files in all the folders of the main folder MCFM, I wrote a C program find.c that finds a given string in all files of a folder and its subfolders and if found, it writes the file names in a .txt file. Then we searched for files containing words 'xbin', 'hist' etc. and we found two files : mbook.f and histofin.f. We modified them and the software was producing correct .C files that does not give compilation error.

Problems with the .C file and manual corrections

1. The output C file (*W_only_lord_CT10.00_80__80__test.C*) doesn't have any function, but to compile with ROOT we need 'void filename () ' at the beginning of the code. From the ROOT, to use ' .x filename.C ' we need the filename to be W_only_lord_CT10.00_80__80__test. However C++ does not support '.' in filename (_CT10.00_ should be _CT10_00_) so we need to replace '.' by '_' . We fix this issue by renaming the C file.

2. In the line 164 (*TH1F *hist = new TH1F("id1",.........)* 'hist' corresponds to a particular histogram but in line 166 *(id1 -> GetXaxis()-> Set.......)* 'id1' is called as 'hist'. This gives an error (*error: use of undeclared identifier 'id1'*) when compiled with ROOT. The best way to fix it is to replace 'hist' by 'id1' or 'id1' by 'hist' . We replace 'hist' by 'id1'(otherwise there would be multiple declarations of 'hist').

3. In the C file the data type 'xbin' is declared every time(line 171 onwards) as 'int xbin' and this gives error (*error: redefinition of 'xbin'*). Once we declare 'int xbin' at the top of function, we can call xbin as many times as we want inside our program.

4. At the end of the file, line 706 (*if (histos -> GetEntries() > 0) then*) 'then' should not be there. To fix this we just remove 'then'.

Steps to get modified output C file automatically

- In MCFM folder go to /src/need
- Open the file named *histofin.f* with any editor
- Go to /src/Integrate
- Open file *mbook.f*

To fix the first problem:

<histofin.f>

- In line 12, where integer variables are declared, add two more variables of names '**i**' and '**n**'. (You can choose any name)

- After line 57 (*the "**end if**" statement*) add the following lines:

```
i=1
n=len_trim(runname)
do
if(i>n) exit
if(runname(i:i)=='.') runname(i:i)='_'
i=i+1
end do
```

This will remove dots from file name.

- At lines 119 and 124 (*previously 113 and 118*) you will see two format statements named as *120* and *121* respectively. Make the *121* format staement as below:

```
121 FORMAT (/1x,
&  'void ',A,'()', /1x,
&  ' {', /1x,
&  ' mcfmhisto = new TFile("', A, '"', "recreate");',/1x,
&  ' mcfmhisto -> cd();',/1x,
&  ' int xbin;', /1X,
&  ' histos = new TobjArray(0);',/1x)
```

This also contains the declaration '**int xbin**'.

- Make line 116 as following:

```
write(96,121) outfileroot(1:nlength),outfileroot(1:nlength+5)
```

To fix second problem:

<mbook.f>

- Make the lines 486-491 as below (*After **IF(BOOK(N).NE.'YES') RETURN***):

```
WRITE(96,131) histoid(1:idlength), histoid(1:idlength),
& TITLE(N)(1:istring), NBIN(N),
& HMIN(N), HMAX(N)

131  FORMAT ( /1X,
& ' mcfmhisto -> cd();', /1X,
& ' TH1F *', A,'= new TH1F( "', A, '"', "', A, '"', ',
```

& I0, ', ', F10.5, ', ', F10.5, ');')

- Change line 522 as below(*where write statement is written*):

  WRITE (96,133) histoid(1:idlength)

- Add a format statement 150 just below of line 522 and before **WRITE (96, *) ' '** as :

  133 FORMAT(/1X, ' histos -> Add(', A,');')

To fix the third problem:

<mbook.f>

- At line 513 where '**int xbin**' is written , delete '**int**' and make it only '**xbin**'. The declaration '**int xbin**' has been made before.

To fix fourth problem:

<histofin.f>

- Go to line 214 and where ***write(96,*) ' if (histos -> GetEntries() > 0) then {'*** is written.
- Delete '**then**' from it and make it **write(96,*) ' if (histos -> GetEntries() >) {'**

- Save files histofin.f and mbook.f

Final executions :

- Now go to folder MCFM and run the '*make*' command again
- From now on, after running ./mcfm, it will generate output .C file with error removed and the C file will contain a function of its own name.
- It can be executed in root using .x filename.C or .L filename.C

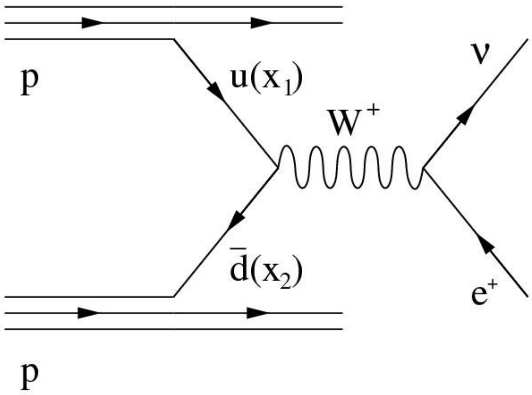
The files find.c, corrected histofin.f and mbook.f can be found in the following link https://drive.google.com/folderview?id=0B3yQpROHCwaRfjVIcVZfMm40T2p5cGxjRWM5X3hVYS1UN1hwOGh0TUNXQV96dUlPZ2hJemM&usp=sharing

Please note that if old .f files are replaced with the new ones provided in the link, delete mbook.o & histofin.o from the object directory, before running 'make'.

After removing bug the software was running correctly.

<Running MCFM for process1>

The Feynman diagram for first process is given :



source: http://inspirehep.net/record/552282/plots

In the above picture x1 and x2 denotes fraction of proton momenta carried by respective partons (for first proton, it is up quark and for second proton it is down antiquark).

Proton is bound state of quarks. When a proton is accelerated at high energy its constituents(quarks) start radiating glouns . These gluons further decay into quark and anti-quark pairs(such type of quarks are callled sea or virtual quarks). This newly generated anti quark(e.g d bar) combine to other quark(e.g u) to produce W+. The W+ further decays into nutrino and positron.

The cross section can be found by running MCFM (through terminal going to folder MCFM/Bin and the typing command ./mcfm_omp) .

For this process getnerated files were:
W_only_lord_CT10_00_80__80__test.C, W_only_lord_CT10_00_80__80__test.dat, W_only_lord_CT10_00_80__80__test.top
The .top file was topdrawer histogram file but I was not supposed to use that. After compiling and executing .C file, three histograms were generated.

RESULTS:

The incident beam direction is taken to be positive z-axis. The component of momentum of a particle in xy plane is called its transverse momentum(P_T).

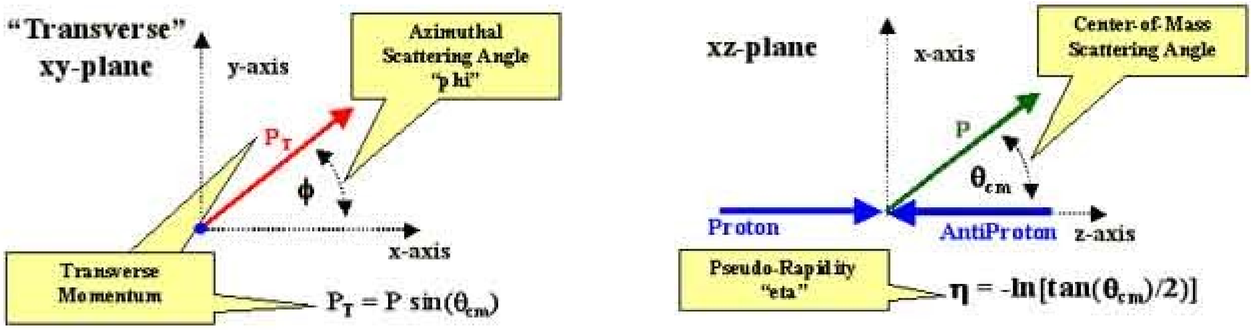
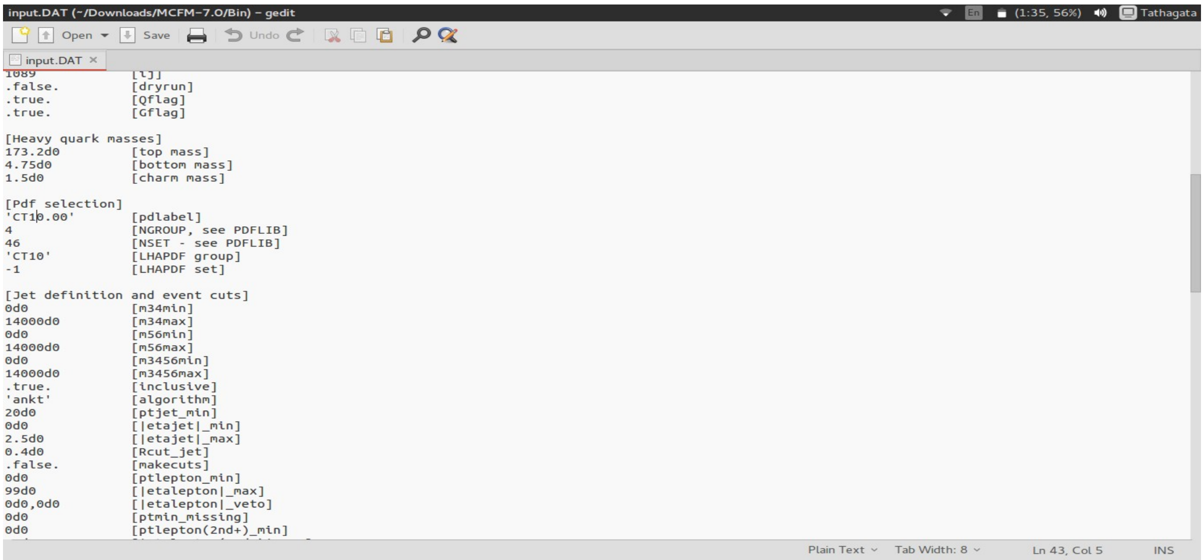


Image Source:http://www.phys.ufl.edu/~rfield/cdf/chgjet/etaphi.html

Rapidity y is defined as $y = \tanh^{-1}(p_z c/E)$ where E is total energy

Definition of P_T and pseudo-rapidity is clear from the picture

Proton(uud) is made of qurks(called partons). These partons share fraction of momenta of proton. The momentum sharing among these partons are given by parton distribution functions(pdf). These pdfs are determined from experiments. There are hundreds of pdfs from various experimental groups. Whenever we study a hadronic process(e.g proton proton collision) we use one of the pdfs. To see whether the pdf we are using is correct we plot the rapidity distribution. Then we match this distribution to the one determined by experiments. If both are matching then the pdf we used is all right otherwise we test other pdfs.

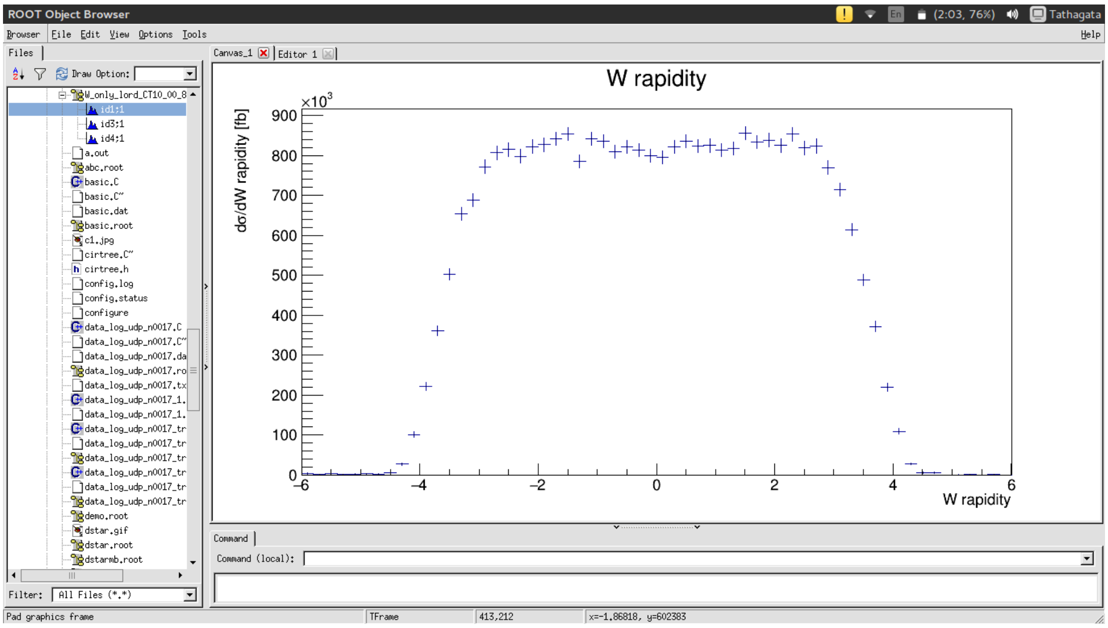


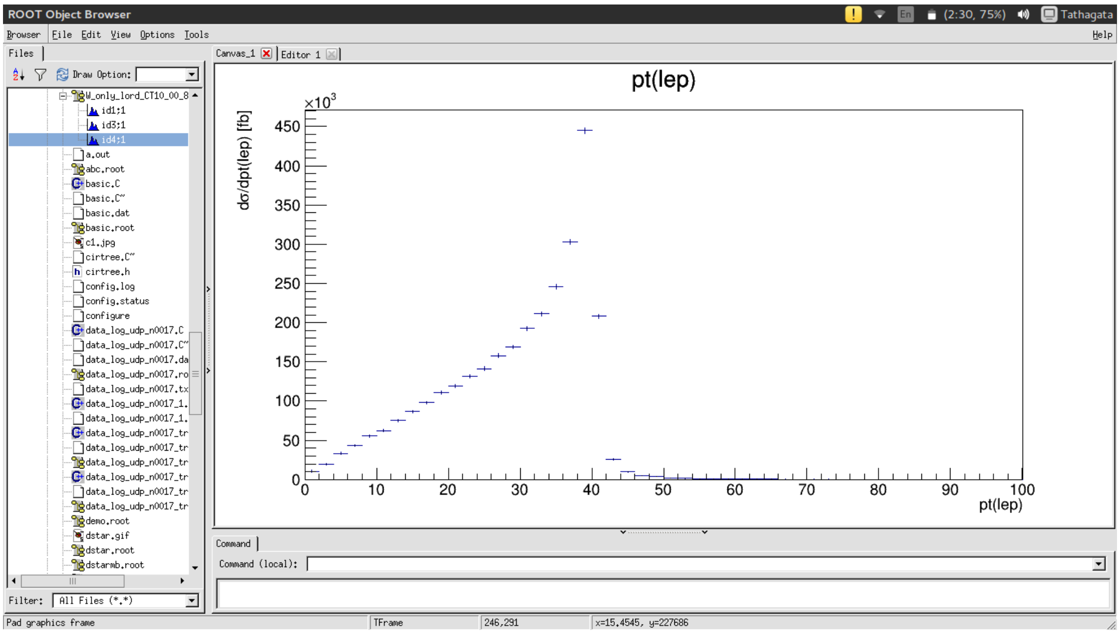
input.DAT (~/Downloads/MCFM-7.0/Bin) - gedit

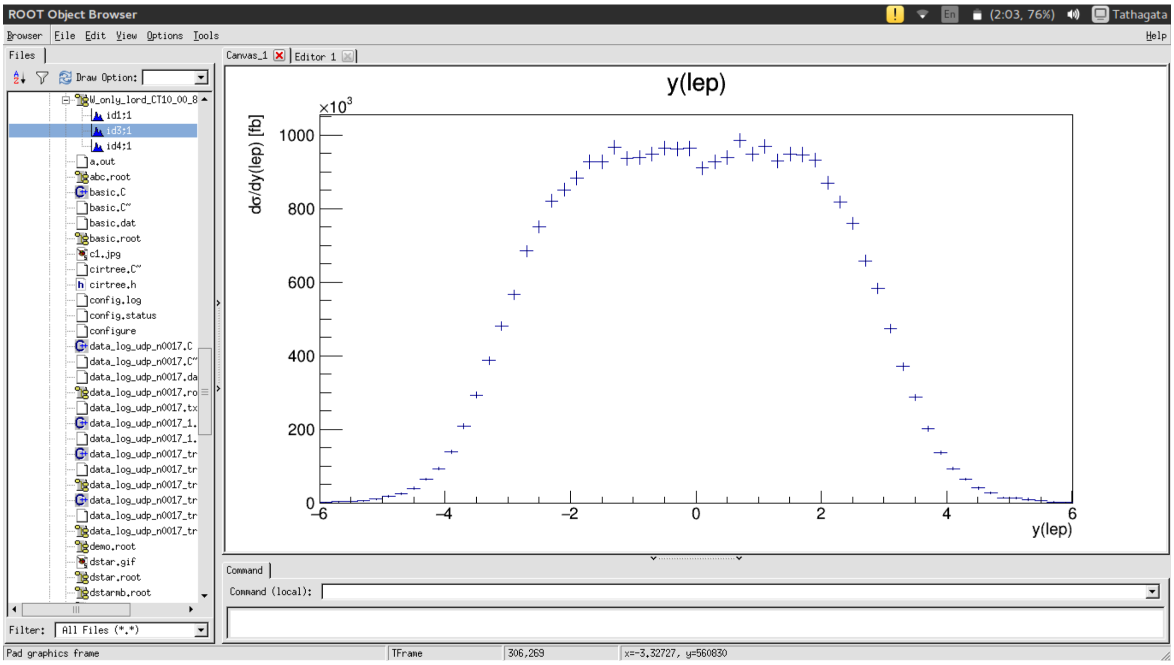
```
.false.         [dryrun]
.true.          [Qflag]
.true.          [Gflag]

[Heavy quark masses]
173.2d0         [top mass]
4.75d0          [bottom mass]
1.5d0           [charm mass]

[Pdf selection]
'CT10.00'       [pdlabel]
4               [NGROUP, see PDFLIB]
46              [NSET - see PDFLIB]
'CT10'          [LHAPDF group]
-1              [LHAPDF set]

[Jet definition and event cuts]
0d0             [m34min]
14000d0         [m34max]
0d0             [m56min]
14000d0         [m56max]
0d0             [m3456min]
14000d0         [m3456max]
.true.          [inclusive]
'ankt'          [algorithm]
20d0            [ptjet_min]
0d0             [|etajet|_min]
2.5d0           [|etajet|_max]
0.4d0           [Rcut_jet]
.false.         [makecuts]
0d0             [ptlepton_min]
99d0            [|etalepton|_max]
0d0,0d0         [|etalepton|_veto]
0d0             [ptmin_missing]
0d0             [ptlepton(2nd+)_min]
```

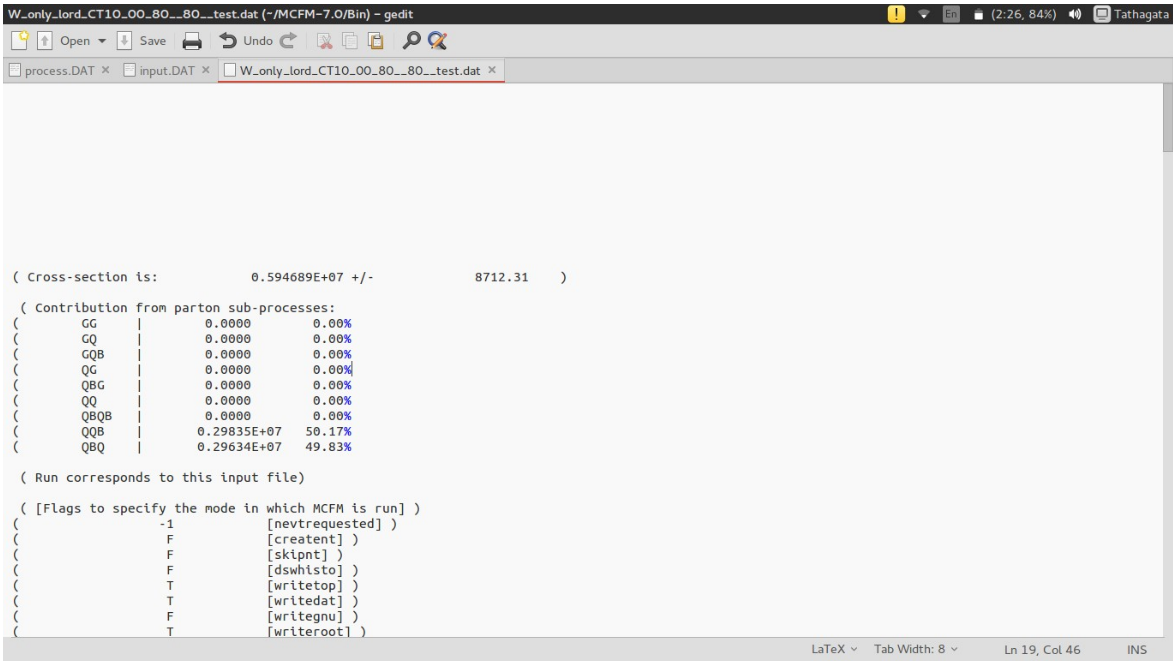
Pdf selected in MCFM can be changed in the file input.DAT file as shown

To get the histograms below we find total cross-section (as function of momentum of incoming particles or any other variable according to our purpose) . Then we differentiate it w.r.t variable in x-axis and plot histograms. These three histograms correspond to rapidity of W^+ boson, rapidity of lepton(y(lep)) and transverse momentum of one of the outgoing leptons(pt(lep)). They represent the result corresponding to the Pdf shown in previous image.









<u>Scattering cross section is</u> 0.594689E+07± 8712.31 femtobarn

CONCLUSION:

Scattering is one of the most important processes used to get details of internal structure of particles. Scattering of two protons (experiment in LHC) have many possible outcomes. Each of the outcomes can be described by many different Pdf's. It is not possible for a human to check all these possibilities. ROOT and MCFM can be used for this purpose. These softwares help us to design and carry out experiments virtually and they can show results in seconds. These softwares will help particle physicists tremendously in their journey of unravelling mystries of nature.

Future Prospects:

As an undergraduate student, I have learnt useful things while doing this project. I was exposed to the procedures of experimental high energy physics to some extent. It has stimulated my interest in high energy physics. For the future, I will study quantum mechanics and particle physics thoroughly. It will help me to understand these processes completely , moreover I will know the complete mathematics behind these. ROOT and MCFM has other aspects (such as geometry package in ROOT) . I am interested to learn them and know their practical use in experiments.

ACKNOWLEDGEMENTS:

I am sincerely grateful to my mentor Prof. Pankaj Jain of our department for giving his valuable time . He has helped me immensely by clearing my doubts from time to time.

I am grateful to Prof. Amey Karkare of CSE department,IITK for helping me with multiple files compilation process.

I sincerely thank Mr. Ravindra Kr Verma for his help in learning MCFM, root and the physics behind the processes. Despite his own busy schedule he managed to find some time for me. I am grateful for that.

Mr. Akshansh Singh helped me with root by supplying various materials and suggestions. I am thankful to him.

Mr. Rishabh Jha helped me from time to time with necessary advices as a responsible senior. I thank him.

Finally, I pay respect to my parents and elder sister for keeping me motivated throughout the project.

REFERENCES:

[1] Quantum Mechanics by Jean-Louis Basdevant and Jean Dalibard, Springer.
[2] Quantum Mechanics Non-Relativistic Theory, L.D. Landau and E.M. Lifshitz, Addision-Wesley Publishing Company, INC.
[3] Introduction To Quantum Mechanics, David J. Griffiths, PRENTICE HALL, 1995.
[4] Quantum Theory, Groups and Representations: An Introduction, Peter Woit(2015) Modern Quantum Mechanics, 2nd Ed, J.J. Sakurai and Jim Napolitano.
[5] An Introduction To Probability Theory and Its Applications, William Feller, First WILEY EASTERN REPRINT, 1972.
[6] Princeton Lectures In Analysis-I (Fourier Analysis An Introduction) , Elias M. Stein and Rami Shakarchi, Princeton University Press.
[7] Introduction To High Energy Physics, 4th Ed, Donald H. Perkins, CAMBRIDGE UNIVERSITY PRESS
[8] Introduction To Elementary Particles, 2nd and Revised Ed, David Griffiths, WILEY-VCH Verlag GmbH & Co. KgaA.
[9] Quarks And Leptons: An Introductory Course in Modern Particle Physics, Francis Halzen, Alan D. Martin, John Wiley & Sons.
[10] Here are some good document about scattering http://www.tcm.phy.cam.ac.uk/~bds10/aqp/lec20-21_compressed.pdf

http://www.itp.phys.ethz.ch/research/qftstrings/archive/12HSQFT1/Chapter09.pdf

[11] This site is very useful to get overview of rapidty and pseudo-rapidity
http://www.hep.shef.ac.uk/edaw/PHY206/Site/2012_course_files/phy206rlec7.pdf

[12] This site contains very good description of a few terms useful in scattering
http://www.phys.ufl.edu/~rfield/cdf/chgjet/etaphi.html
[13] For feynman diagrams
http://www-pnp.physics.ox.ac.uk/~barra/teaching/feynman.pdf

http://www.phy.pmf.unizg.hr/~kkumer/articles/feynman_for_beginners.pdf

[14] Fundamentals of C++ Programming,Richard L. Halterman, School Of Computing, Southern Adventist University,2015
The C++ Programming Language, 3 Ed, Bjarne Stroustrup, Addision-Wesley.

[15] Root user guide is available here
https://root.cern.ch/root/htmldoc/guides/users-guide/ROOTUsersGuideA4.pdf
Tutorials are available here
https://root.cern.ch/drupal/content/tutorials-and-courses

[16]For brief discussion about trees
https://www.linksceem.eu/ls2/images/stories/ROOT_Day4.pdf

[17] These are very useful books of fortran
Modern Fortran Explained, Michael Metcalf, John Reid, Malcolm Cohen, Oxford University Press.
Fortran 95/2003 for Scientists and Engineers, 3 Ed, Stephen J. Chapman, McGraw-Hill Primis.

Submitted by-

Tathagata Karmakar
KVPY Reg. No-SB1312043
BS Physics, 2nd Year
14754, IIT Kanpur

Date......................

Submitted to-

The Mentor
Prof. Pankaj Jain
Dept. of Physics
IIT Kanpur

Guided by-

Mr. Ravindra Kr Verma
Research Scholar
Dept. of Physics
IIT Kanpur

[16]For brief discussion about trees
https://www.linksceem.eu/ls2/images/stories/ROOT_Day4.pdf

[17] These are very useful books of fortran
Modern Fortran Explained, Michael Metcalf, John Reid, Malcolm Cohen, Oxford University Press.
Fortran 95/2003 for Scientists and Engineers, 3 Ed, Stephen J. Chapman, McGraw-Hill Primis.

Submitted by-

Tathagata Karmakar

Tathagata Karmakar
KVPY Reg. No-SB1312043
BS Physics, 2nd Year
14754, IIT Kanpur

Date 06.07.2015

Submitted to-

The Mentor
Prof. Pankaj Jain
Dept. of Physics
IIT Kanpur

Guided by-

Mr. Ravindra Kr Verma
Research Scholar
Dept. of Physics
IIT Kanpur