

Exercise #9 - Analysis of Large Data Sets

Computational Physics Lab

Instructor: Prof. Sean Dobbs

March 31, 2020

Due by 11 PM, April 24th

Submission

For each problem, you will write one or more python programs. These programs should follow the Python coding and formatting conventions outlined for the course. Many problems will have additional questions to answer, or will ask for a record of the terminal showing the output when the program is run with various arguments. These answers can be given either in block comments in the prologue of the corresponding assignment, or in clearly labeled text files. The terminal output should contain some information on the username and machine you are running on (this will often be in the command prompt).

You will be evaluated based on the files contained in your remote GitHub repository at the due date. You should and commit all the files you created to your local repository on a regular basis. You should do this by adding any new or modified files using "git add", and then finalizing changes using "git commit". Remember that "git status" will give you information on which files in your repository have been changed. Remember to add short, but useful comments when performing a commit. When you are finished with the exercise, push the current status of your local repository to the remote repository on GitHub using the command "git push -u". You are encouraged to push your local files to the remote repository periodically before you are done, and certainly well before the deadline, if possible.

1. Analysis of Large Data Sets

Studies of hadronic spectra via strong decays of hadrons provide insight regarding the Strong Nuclear Force (i.e. Quantum Chromodynamics, or QCD) at the confinement scale. These studies have led to phenomenological models for QCD such as the constituent quark model. Data from studies of $\gamma p \rightarrow \pi^+ \pi^- \pi^+ n$ are located in a file named n3pi.dat. This can be accessed at

<http://hadron.physics.fsu.edu/~sdobbs/phyz4151/n3pi.dat> (this can be downloaded on the command line by a command like `wget http://hadron.physics.fsu.edu/~sdobbs/phyz4151/n3pi.dat`), or it can be accessed in the HPC at `/gpfs/home/sdobbs/phz_4151-scripts/pyroot/n3pi.data`. This file contains over 1 million events of the reaction depicted above. The information for each reaction is given in a simple ascii format. An example of one such event is given below:

```
4
1 0 0.000000e+00 0.000000e+00 5.000000e+00 5.000000e+00
8 1 -7.692400e-02 1.210350e-01 3.835767e+00 3.841000e+00
9 -1 -1.182520e-01 -1.387120e-01 3.485660e-01 4.175200e-01
8 1 2.804500e-02 -3.515470e-01 4.627850e-01 5.984500e-01
```

For this exercise, you are expected to use the ROOT framework, and are strongly recommended to work on the HPC, where a common installation of ROOT will be provided for you. Configuration directions can be found on the course website or the provided lecture slides. Installing ROOT on your local computer is often non-trivial.

The first line of the event corresponds to the number of observed event particles which follow. After that, the next line corresponds to the first particle which is the photon beam. Following the beam are three charged pion mesons (two positively and 1 negatively charged). The format of each particle line is: {particle id} {charge} {p.x} {p.y} {p.z} {E}. The charge is given in units of the electron charge (+1,-1, or 0), and the momentum and energy are given in GeV/c and GeV, respectively. Particle id=1 represents the photon beam whereas id=8 is a π^+ and id=9 is a π^- . The two baryon particles (i.e. the target proton and the recoil neutron) in the event are not given. The 2 proton target interacts at rest with a rest mass $m_p = 0.938 \text{ GeV}/c$. Conservation of four-momentum (energy and momentum) can be readily employed to provide the neutron four-momentum information.

1) Choose an event from the data file and calculate by hand (or calculator) the magnitude of the four-vector for each particle listed in the event. Given your event and the fact the target proton four-momentum (p.x, p.y, p.z, E) is (0,0,0,0.938), calculate the magnitude of the missing particle four-momentum. Recall that the missing_4vec = beam_4vec + target_4vec - (piplus1_4vec + piplus2_4vec + piminus_4vec). Note, this does not require the use of a computer.

2) Your repository also contains a copy of the program `dumpData.py`, which reads the `n3pi.dat` data file. Use this program to create a new program which uses the ROOT `TLorentzVector` objects to represent the four-momentum vectors of the particles in an interaction event. For each event, calculate the four-momentum of the missing neutron. For the first 5 events print the invariant mass of the missing neutron and check that the mass is consistent with the rest mass of a neutron ($937 - 941 \text{ MeV}/c^2$). After the first 5 events, your program should continue and count the number of events in the data file, and finally print out this total number of events.

3) Use your program to make a new program that uses either the ROOT `TH1F` histograms or a ROOT `TNtuple` data container. In either case use the ROOT `TFile` object to save your ROOT Objects to a ROOT file with the filename extension `".root"`. For each particle in the event, calculate the invariant four-momenta mass for the following 8 four-momenta combinations: $n\pi_1^+\pi_2^+\pi^-$, $\pi_1^+\pi_2^+\pi^-$, $\pi_1^+\pi^-$, $\pi_2^+\pi^-$, $\pi_1^+\pi_2^+$, $n\pi_1^+$, $n\pi_2^+$, $n\pi^-$. Fill histograms or a `TNtuple` with the 8 invariant mass values over all events in the data file. After processing all events, draw histograms or `TNtuple` projections of each invariant mass quantity using 8 plots on one ROOT `TCanvas` in a 2x4 division. Save the histograms or `TNtuple` to a ROOT `TFile`. Using the results, describe and explain the features in the invariant mass spectra.

For full credit in your programs and a print out of the discussion from part 3.