Progress of Project, May 24, 2018

Aidan Lytle

June 11, 2019

1 List of Scripts and Their Progress

We can begin with a list of our scripts:

All3.C generates a set of comparative Graphs of our three candidate fitting functions, and allows us to have a visual representation of pair production (2 particle correlations) as a function of charged multiplicity. It is ostensibly complete, and its output has been displayed in a beamer format.
It was my first complete code, and had many parts pulled from other places.

simpleparticle.C uses the external libraries of ROOT and Pythia to generate a given number of events, and stores their outputs in a TFile. These outputs inlcude, but are not limited to, the pairs produced, the phi distribution, and the eta distibution, along with some other samplings.
This is our main event generation and storage code, and most of its issues came from trying to access the pythia libraries from outside the pythia code. These issues have been resolved.

TFileOpener.C opens and plots the data produced and processed in simple... and is itself mostly complete.

CosinePlotter.C plots the q vectors which are measured from the earlier productions. It also is a first set of data that has been matched with a function from All3.C.
This code is our first synthesis of a purely mathematic basis and the physical situation we are trying to model. It is rather simple, in three sections, which respectively open, fit, and plot the data and its respective graph.

CumulantFromCK.C uses the data produced in the earlier TFile to plot the cn values for 2 through 8, and uses basic mathematic error manipulation to remove the contribution from the lower harmonics to the higher.

RandomPhi.C is a random 'phi' distibutor (that I did in my spare time) which is a prototype for Dylan's work. It is (in the future) going to be equivalent to a mini-pythia, or part of one. it radnomly generates a number between 0 and pi, and uses this number (with a system time and ROOT TRandom compatible random number procedure).

NGrapher.C allows us to plot the particle correlations as a function of N in our mathematical model, namely a plot of N versus $\frac{(N-1)...^{k-1}}{N^{k-1}}$. This is relevant because we need to determine the behavior of the function in this space, so that we can understand the conditions under which it adheres to reality. We are also adding a bit of a loop to extend this to $\frac{N}{k}$ as well.

2 Towards Completion

Our end goal is to produce a solid correlation between the 2, 4, 6, and 8 particle correlations and the charged muliplicity. The combinatorics have been heavily analyzed and produced in All3.C, and the specific 'choose' function has been shown to be a converging approximation of the Factorial function at high values. The intuition for this is that the particles have directional behaviors which are semi-random with respect to the beam axis, allowing us to treat them as a semi-flat distribution (offset by the percent overlap and total momentum transfer/energy density at time of collision). Pythia, however, generates only non-flow events, versus the flow events we are trying to separate out from these.

This leads to our next effort: run many more events, then run the scripts against real data from the RCF (RHIC Computing Facility). This will allow us to test our hypothesis against data from 'true' collisons.

The data from the events generated by pythia needs to be matched, fully, to the given function, and we need a much larger body of data to be certain (our largest plotted data set was around 1,000,000 events). Our macros for opening this data are mostly complete, and we have also been awarded a 1000.00 dollar grant from the UNCG office of Undergraduate Research. This will got towards compensation, costs of computing etc, and motivation for the work.

A nice organized folder is also in order in the near future, though github has been working well.