Simulating pp collisions with Pythia

By
Vishu Saini, MSc Physics IIT Bombay

Nov. - December 2021

1 Simulations with Pythia

1.1 Pythia

The Pythia program is a standard tool for the generation of high- energy physics collisions, by the help of pythia, events are generated . Events are the sets of outgoing particles produced in the interactions between two incoming particles.

1.2 pp Collisions at 13 TeV

This program will take two incoming proton beams at a center of mass energy of 13TeV,and the outcomes are given in histograms. The Code for Pythia is given below:

```
/*Author: Vishu Saini, MSc Physics, IIT Bombay */
/*Date: December 2021*/


/* proton -proton collisions*/
/* Using trees to store data and plotting them using root. Trees can also be used to feed data to GEANT4 */
#include <iostream>

#include "TFile.h"
#include "TTree.h"

#include "Pythia8/Pythia.h"

int main()
{
  TFile *output = new TFile("tutorial3.root", "recreate");

  TTree *tree1 = new TTree("tree1","tree1");   // charged particle tree

  TTree *tree2 = new TTree("tree2","tree2");  // neutral particle tree

  // Saving the variables as branches

  int id, event, size, no;
  double pT_1, eta_1, y_1;          //declare the variables first
                                   // I = integer, D = double
  tree1->Branch("pT_1", &pT_1, "pT_1/D");
  tree1->Branch("eta_1", &eta_1, "eta_1/D");
  tree1->Branch("y_1", &y_1, "y_1/D");

  double pT_2, eta_2, y_2, phi_2;
  tree2->Branch("pT_2", &pT_2, "pT_2/D");
  tree2->Branch("eta_2", &eta_2, "eta_2/D");
  tree2->Branch("y_2", &y_2, "y_2/D");



  int nevents = 1e4;    // number of events

  Pythia8::Pythia pythia ;

  pythia.readString("Beams:idA = 2212");     // proton beam A
  pythia.readString("Beams:idB = 2212");     //proton beam B
  pythia.readString("Beams:eCM = 14.e3");    //CM energy
  pythia.readString("SoftQCD:all = on");     // turn on QCD physics
  pythia.readString("HardQCD:all = on");    //turnn on QCD physics

  pythia.init() ;

  for (int i = 0; i < nevents; i++)
  {

    if (!pythia.next()) continue ;

    int n_charged_particle = 0, n_neutral_particle = 0, n_pion_particle = 0,
    n_kaon_particle = 0;
    int entries = pythia.event.size();

    std:: cout << "Event: " << i << std:: endl;
    //std:: cout << "Event size: " << enteries << std:: endl;

    event = i;
    size = entries;
    for (int j = 0; j < entries ; j++)
    {
      if (pythia.event[j].isFinal()) {
        if (pythia.event[j].isCharged()) {
          pT_1 = pythia.event[j].pT();
          eta_1 = pythia.event[j].eta();
          y_1 = pythia.event[j].y();
          n_charged_particle ++;
          tree1->Fill();}
        }

      if (pythia.event[j].isFinal()) {
        if (pythia.event[j].isNeutral()) {
          pT_2 = pythia.event[j].pT();
          eta_2 = pythia.event[j].eta();
          y_2 = pythia.event[j].y();
          phi_2 = pythia.event[j].phi();
          n_neutral_particle ++ ;
          tree2->Fill();}
          }

    }

  }

  output->Write();
  output->Close();

  return 0;

}
```

Listing 1: Pythia Code

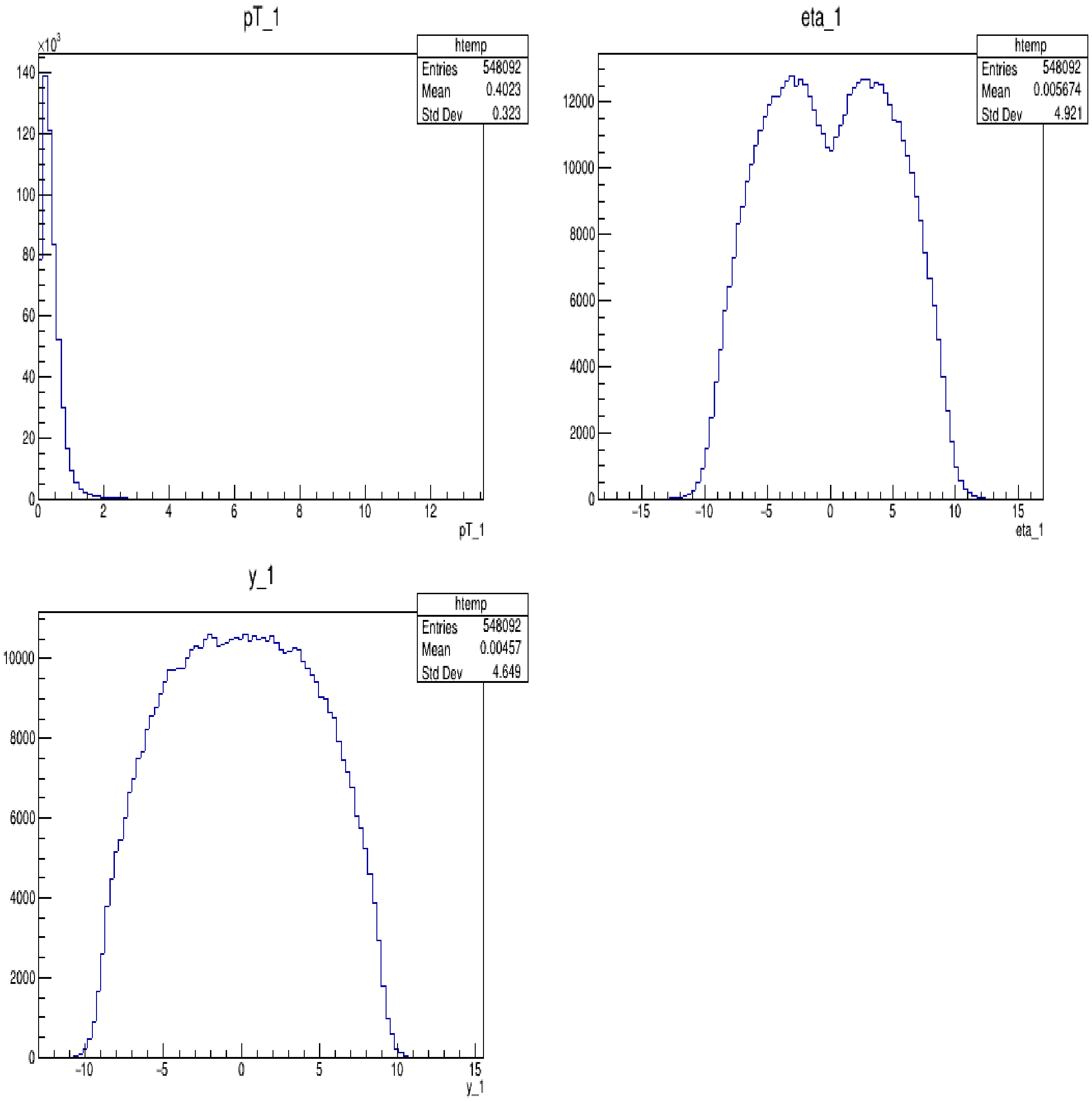

Figure 1: Charged particle distributions

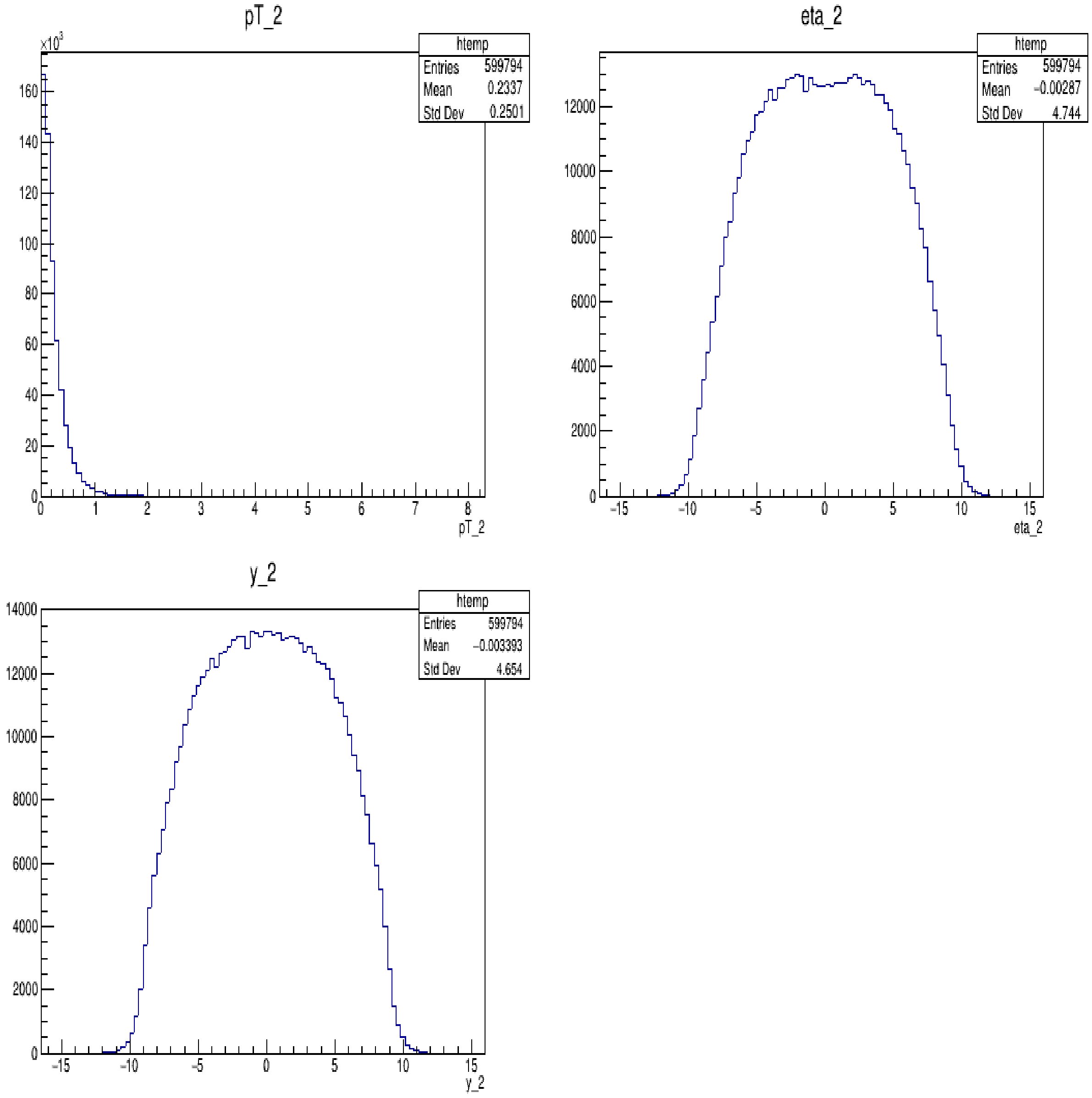

Figure 2: Neutral particle distributions

References

[1] Cheuk-Yin Wong. *Introduction to High-Energy Heavy-Ion Collisons*. World Scientific., Singapore, 2016.

[2] S.C Garg, R.M Bansal, C.K Ghosh. *Pythia Documentation*. [*https://pythia.org/documentation/*].

[3] R.K Puri, V.K Babbar. *ROOT Manual*. [*https://root.cern/manual/*].