2017 GSoC: electron-ion collisions in Pythia 8

Phil Ilten (philten@cern.ch) and Mike Williams (mwill@mit.edu)
CERN, HEP Software Foundation, MIT

1 Introduction

The Pythia 8 Monte Carlo event generator provides a comprehensive software tool for simulating high energy particle collisions, designed specifically to describe proton-proton collisions at the Large Hadron Collider (LHC). The primary focus over the next decade of the US nuclear physics program is the development of the Electron-Ion Collider (EIC) which requires dedicated Monte Carlo tools for detector design and phenomenology studies. The focus of this project is to develop Pythia 8 to provide accurate simulations of electron-proton (ion) collisions with the option of arbitrarily polarized beam configurations.

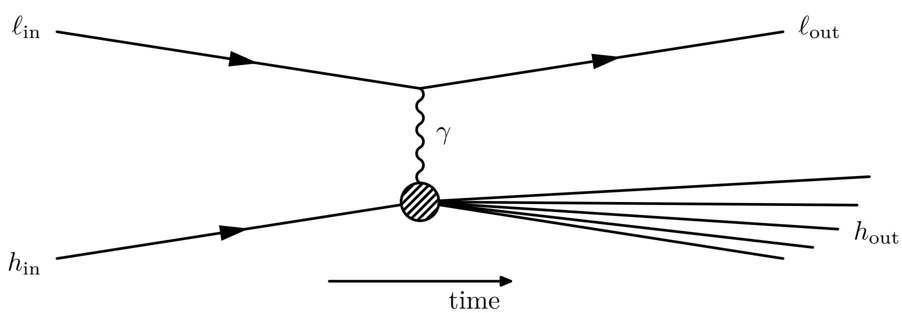
Currently, some electron-proton processes are already available in Pythia 8. However, there are a number of processes missing and a framework for polarized beam configurations is not yet implemented. The goals for this project are:

1. Implement remaining electron-proton processes in Pythia 6 missing from Pythia 8. This will primarily involve including vector meson contributions to photo-production.
2. Create a framework for polarized beams and begin work to propagate this throughout the entirety of the event generation, up to the beginning of the parton shower.
3. Compare electron-proton scattering results from Pythia 8 to data, and tune relevant parameters.

The above goals involve not only significant coding but also some non-negligible physics. However, in Pythia 6 (the predecessor to Pythia 8) the physics of 1 and 2 are already implemented in Fortran. Consequently, extensive physics knowledge is not necessarily needed, but rather the ability to port code from the Fortran of Pythia 6 to the native C++ of Pythia 8. However, a basic understanding of vectors and special relativity is needed. In the following sections a series of exercises are provided to introduce interested students to some details of the project.

2 Transverse Energy Flow

The HERA accelerator collided electrons into protons and used particle detectors to measure the momentum four-vectors of the particles produced in each collision. This process is very similar to the EIC processes we are interested in making theory predictions for. The diagram below shows a schematic of the process.



The incoming electron (ℓ_{in}) exchanges a photon (γ) with the incoming proton (h_{in}). The outgoing electron (ℓ_{out}) is the incoming electron but with a modified momentum four-vector, while the incoming proton dissociates into a spray of particles (h_{out}). The number of h_{out} can be different for every collision. Each incoming and outgoing particle has a known momentum four-vector (p) corresponding to the x, y, and z-components of the particle momentum, and its energy. The azimuthal angle (ϕ), polar angle (θ), and pseudo-rapidity (η) for each particle can be calculated from its momentum four-vector.

Given the incoming and outgoing momentum four-vectors, there are two important physics quantities that can be calculated for each electron-proton collision. First, the momentum four-vector for γ can be calculated by,

$$p(\gamma) = p(\ell_{\text{in}}) - p(\ell_{\text{out}}) \tag{1}$$

which is the momentum four-vector of the outgoing electron subtracted from the momentum four-vector of the incoming electron. Second, the variable W is defined as,

$$W \equiv m\big(p(\gamma) + p(h_{\text{in}})\big) \tag{2}$$

where $m(p)$ is the invariant mass of a momentum four-vector.

Sometimes it is useful to transform the momentum four-vectors into another frame through a series of rotations and boosts. The hadronic center-of-mass system (CMS) is defined by the three following requirements.

- $\phi(\gamma) = \theta(\gamma) = 0$: the photon momentum three-vector is oriented along the z-axis
- $\vec{p}(\gamma) = \vec{p}(h_{\text{in}})$: the momentum three-vectors of the photon and the incoming proton are equal
- $\phi(\ell_{\text{out}}) = 0$: the outgoing electron momentum three-vector is in the xz-plane

In special relativity, a sequence of rotations and boosts can be encoded in a 4×4 transformation matrix. The matrix needed to transform from the collision frame to the hadronic CMS ($\mathcal{H}$) can be defined by the following operations.

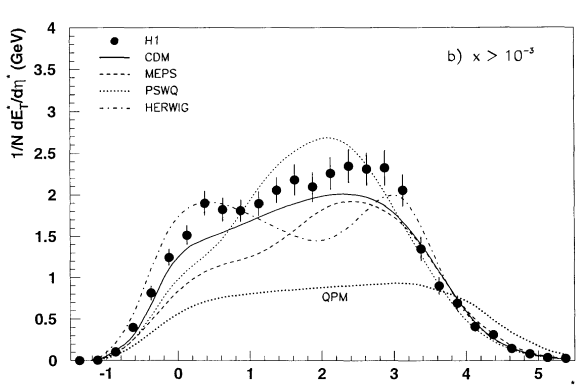
- apply a backwards boost of $p(\gamma) + p(h_{\text{in}})$ to $\mathcal{H}$
- transform γ with $\mathcal{H}$ to γ'
- apply a rotation in ϕ of $-\phi(\gamma')$ to $\mathcal{H}$
- apply a rotation in θ of $-\theta(\gamma')$ to $\mathcal{H}$
- transform ℓ_{out} with $\mathcal{H}$ to ℓ'_{out}
- apply a rotation in ϕ of $-\phi(\ell'_{\text{out}})$ to $\mathcal{H}$

Note that the order of operations is important. In most Monte Carlo generators, including Pythia 8, transformation matrix and four-vector classes are provided to make implementing the process above relatively simple.

In the paper "Energy flow and charged particle spectra in deep inelastic scattering at HERA" from,

`https://cds.cern.ch/record/260613`

the average transverse energy (E_{T}) of all h_{out} per electron-proton collision is measured as a function of η in the hadronic CMS. The E_{T} for a particle can be determined from its momentum four-vector. The data from this measurement is given in the plot below.



The goal of these exercises is to use Pythia 8 to make a theory prediction for this data. To do this, we first need to determine exactly how this measurement was made.

1. What is the energy of the incoming electron and incoming proton?
2. In Monte Carlo, what are the requirements placed on the following for each event?
   - outgoing electron, ℓ_{out}
   - outgoing particles, h_{out}
   - W^2
   - forward energy

   Try looking in the introduction to section 5 of the paper. The forward energy is the sum of the energies from all outgoing particles with $4.4° < \theta(h_{\text{out}}) < 15°$.

3 Introduction to Pythia 8

Pythia 8 is an open source program written in C++. The source for the most recent version can be downloaded from,

`http://home.thep.lu.se/~torbjorn/pythia8/pythia8223.tgz`

as a zipped tarball. Within the source a `README` is provided, as well as an `HTML` manual and extensive example programs. The following resources are also useful when working with Pythia 8.

- online version of the `HTML` manual: `http://home.thep.lu.se/~torbjorn/pythia82html/Welcome.html`
- Doxygen API documentation: `http://home.thep.lu.se/~torbjorn/doxygen/index.html`
- introductory worksheet: `http://home.thep.lu.se/~torbjorn/pdfdoc/worksheet8200.pdf`

Try downloading Pythia 8 and running some of the examples. Take some time exploring the documentation and source code to gather a first impression of how the code works.

1. What is the primary class and what is the primary method through which it is configured?
2. What method is used to generate a collision, and how are the particles from the collision accessed?
3. What class is used to represent momentum four-vectors, and where in the source code is it defined?
4. In section 2 the variables ϕ, θ, η, m, and E_{T} were introduced as a quantities which can be calculated from a momentum four-vector. Find the definition of these variables in the source code as a function of the components of a momentum four-vector, $p = (E, p_x, p_y, p_z)$.
5. In section 2 the transformation matrix $\mathcal{H}$ was introduced. What class is used in Pythia 8 to represent a transformation matrix? What method applies a backwards boost to the matrix? What method applies a rotation in θ or ϕ? How is a transformation matrix applied to a four-vector?

4 Predictions with Pythia 8

The goal outlined in section 2 is to make a theory prediction for the E_{T} flow using Pythia 8. Following the steps below as well as the information from sections 2 and 3, try to make an E_{T} flow prediction which can be compared to the data from HERA.

1. Create a `Pythia` object and configure it so beam A is the incoming proton and beam B is the incoming electron. Set the energies for these two beams with the energies from question 1 of section 2. Turn on the physics process `WeakBosonExchange:ff2ff(t:gmZ)` which corresponds to the diagram given in section 2. Set `WeakZ0:gmZmode` to 1.

2. Create a histogram to store the E_T flow. The histogram should have 26 bins from $-1.25 < \eta(h_\mathrm{out}) < 5.5$. This can be done either with the PYTHIA 8 class `Hist` or some external graphics package, *e.g.* ROOT.

3. Generate 100000 events with the `Pythia` object. The following steps should be applied to each generated event.

4. Skip the event if the x value for beam A is less than 10^{-3}. This can be accessed from the `Info` object of the `Pythia` instance; here beam A corresponds to x_1.

5. Determine the event index for the outgoing electron, ℓ_out. The method `iBotCopyId` may be useful. Create momentum four-vectors for the incoming electron, outgoing electron, incoming proton, and the exchanged photon.

6. Calculate W^2. Skip the event if either W^2 or the outgoing electron does not fulfill the requirements from question 2 of section 2.

7. Loop over the particles in the event. Use only the final state particles that are not the outgoing electron and have $4.4^\circ < \theta(h_\mathrm{out}) < 15^\circ$. Skip the event if the forward energy does not meet the requirement from question 2 of section 2.

8. Create the transformation matrix $\mathcal{H}$ using the class of question 5 from section 3 using the sequence of boosts and rotations outlined from section 2.

9. Loop over the particles in the event. For each final state particle that is not the outgoing electron, transform its momentum to the hadronic CMS with $\mathcal{H}$, and fill the histogram with its η, weighted by its E_T.

10. After all events have been generated, scale the E_T flow histogram by,

$$\frac{1}{N_\mathrm{accept}\Delta\eta}$$

where N_accept is the number of events that pass all the requirements of question 2 from section 2 and $\Delta\eta$ is the width of each bin in the histogram. Output the E_T flow histogram.

5 Compare to Data

Now we would like to compare the predictions of section 4 with the actual data from HERA in the plot of section 2. The data from this plot is provided in table 1 of the HERA paper. However, we would rather not transcribe the data from the paper, as this can be both tedious and mistakes can be made. In the particle physics community, much of the data provided by experiments in papers is available online at,

`https://hepdata.net`

in a variety of digital formats.

1. Search the database and find the entry for the HERA paper of section 2.

2. Download the digitized data from table 1 of the HERA paper and compare this to the prediction of section 4.

3. How well does the PYTHIA 8 theory prediction compare to data?

6 Modify Pythia 8

As mentioned in section 1, we want to include more physics details into photo-production, the process we simulated in section 4. With the comparison from section 5 between our prediction and the HERA data, we can now modify Pythia 8 and see whether the agreement becomes better or worse. This modification will just be a toy example, *e.g.* the change is not motivated by any physics reasoning.

1. In what class is the process of `WeakBosonExchange:ff2ff(t:gmZ)` used in section 4 implemented?
2. Where in the source code is the class defined?
3. From what base class(es) does this class derive?
4. The class for this process returns a cross-section ($\hat{\sigma}$) via the method `sigmaHat`. What is the formula for $\hat{\sigma}$ of this process? Remember, `WeakZ0:gmZmode` is 1. The class members x`H` correspond to $\hat{x}$ where x is a Mandelstam variable: s, t, or u.
5. Modify `sigmaHat` for this process by multiplying it by $\hat{s}^2$.
6. Rerun the prediction of section 4. Does anything change?

7 Run Pythia 6

The predecessor to Pythia 8 is Pythia 6 which is written in Fortran. The source can be downloaded from,

`http://home.thep.lu.se/~torbjorn/pythia6/pythia6428.f`

which is just a single source code file. Take some time to browse this code, as much of the physics which we would like to port to Pythia 8 has already been implemented here.

1. What subroutine provides the equivalent $\hat{\sigma}$ calculation of the `WeakBosonExchange:ff2ff(t:gmZ)` process?
2. Optionally, try to repeat the exercise of section 4. This is a rather involved comparison, so only do this if you have time!