Pythia tutorial

Joni Laulainen (University of Jyväskylä)

25th November 2025

Who am I? Why am I here?

- 4th year PhD student at University of Jyväskylä, Finland.
- My projects aim to improve DIS in Pythia (matching, merging, tuning).
- Supervisor I. Helenius is an author in Pythia collaboration.

Who am I? Why am I here?

- 4th year PhD student at University of Jyväskylä, Finland.
- My projects aim to improve DIS in Pythia (matching, merging, tuning).
- Supervisor I. Helenius is an author in Pythia collaboration.
- He[1] made me do it[2].

[1] well, it was another Pythia author, but Ilkka was *very* encouraging

[2] "It's good practice! You get to meet experts! The subjects at the school are important!"

Who am I? Why am I here?

- 4th year PhD student at University of Jyväskylä, Finland.
- My projects aim to improve DIS in Pythia (matching, merging, tuning).
- Supervisor I. Helenius is an author in Pythia collaboration.

- I'm going to use a silly car analogy:

Who am I? Why am I here?

- 4th year PhD student at University of Jyväskylä, Finland.
- My projects aim to improve DIS in Pythia (matching, merging, tuning).
- Supervisor I. Helenius is an author in Pythia collaboration.

- I'm going to use a silly car analogy:
  - **Basic knowledge:** how to drive the car, what kind of fuel it needs, how and when to use fog lights etc. (How to run Pythia, what input and settings to use)

Who am I? Why am I here?

- 4th year PhD student at University of Jyväskylä, Finland.
- My projects aim to improve DIS in Pythia (matching, merging, tuning).
- Supervisor I. Helenius is an author in Pythia collaboration.

- I'm going to use a silly car analogy:
  - **Basic knowledge:** how to drive the car, what kind of fuel it needs, how and when to use fog lights etc. (How to run Pythia, what input and settings to use)
  - **Intermediate level:** How does a combustion engine work? (How does Pythia generate events)

Who am I? Why am I here?

- 4th year PhD student at University of Jyväskylä, Finland.
- My projects aim to improve DIS in Pythia (matching, merging, tuning).
- Supervisor I. Helenius is an author in Pythia collaboration.

- I'm going to use a silly car analogy:
  - **Basic knowledge:** how to drive the car, what kind of fuel it needs, how and when to use fog lights etc. (How to run Pythia, what input and settings to use)
  - **Intermediate level:** How does a combustion engine work? (How does Pythia generate events)
  - **Expert level:** How to build it from scratch? (Code structure, physics models, mathematical tools and their implementation)
- My goal today is to teach you how to drive the car (run the code).

Setup

- Install Docker Desktop, test with

```
$ docker --version
> Docker version 27.4.0, build bde2b89
```

- For Linux systems, some steps might be required: post-installation steps, re-launch terminal and Docker Desktop.
- Download the image for Pythia8 + Rivet4

```
$ docker pull hepstore/rivet-pythia
```

- Start the container, setting up a shared "host" directory between your local machine and the container

```
$ mkdir PythiaTutorial && cd PythiaTutorial
$ docker run -v $PWD:/host -it --rm hepstore/rivet-pythia
```

Introduction

- What is Pythia?

Introduction

- Pythia was the oracle at the Temple of Apollo in ancient Greece. She would inhale volcanic fumes and ramble and shriek incoherently.

Introduction

- Pythia was the oracle at the Temple of Apollo in ancient Greece. She would inhale volcanic fumes and ramble and shriek incoherently.
- The priests would interpret these as predictions of the future, writing them as poems.

Introduction

- Pythia was the oracle at the Temple of Apollo in ancient Greece. She would inhale volcanic fumes and ramble and shriek incoherently.
- The priests would interpret these as predictions of the future, writing them as poems.
- Now in this analogy, Pythia (the MCEG) is Pythia (the oracle), and **you** are the priests.

Introduction

- Pythia was the oracle at the Temple of Apollo in ancient Greece. She would inhale volcanic fumes and ramble and shriek incoherently.
- The priests would interpret these as predictions of the future, writing them as poems.
- Now in this analogy, Pythia (the MCEG) is Pythia (the oracle), and **you** are the priests.
- The traffic rules for Pythia, or any MCEG, is to use critical thinking when interpreting the results.

Introduction

- What is Pythia?

Introduction

- What is Pythia?

  A general-purpose Monte Carlo event generator

Introduction

- What is Pythia?

  A general-purpose Monte Carlo event generator
- What is a general-purpose Monte Carlo event generator?

general-purpose

QCD processes

- Partonic hard scattering ($2 \rightarrow 2$, some $2 \rightarrow 3$)
- Heavy quark production

Electroweak processes

- Prompt photon production
- EW boson production and exchange
- Deep inelastic scattering
- Photon collisions

Onia production

- Charmonium, Bottomonium with different cpin states

Top production

- $t\bar{t}$ pairs and single top

Heavy-ion collisions and cosmic rays

- ^{2}H, ^{16}O, ^{208}Pb, ...

Higgs production

- Standard Model Higgs, also in association with other particles
- Beyond-the-Standard-Model Higgs

Beyond the Standard Model

- Supersymmetry
- Dark matter
- Hidden Valley
- Leptoquarks

Monte Carlo event generator

- Uses random number generation to produce events, owing to the probabilistic nature of Quantum Mechanics.
- Acts as a "transfer function" between theory and experiment: physical theory as input and outgoing on-shell particles as output.
- Also goes the other way: experiments drive the development of MCEGs and constraints of theory

Introduction

- What is Pythia?

  A general-purpose Monte Carlo event generator
- What is a general-purpose Monte Carlo event generator?

  Some transfer function between theory and experiment

Introduction

- What is Pythia?

  A general-purpose Monte Carlo event generator
- What is a general-purpose Monte Carlo event generator?

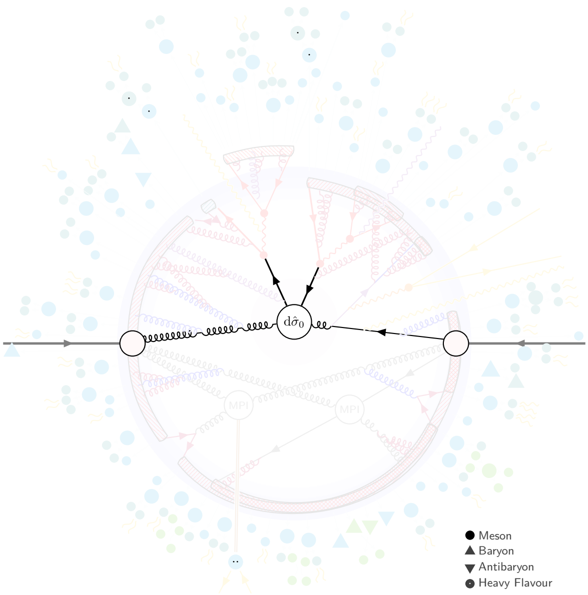
  Some transfer function between theory and experiment
- How does it work?

Event generation steps

Factorization theorems allow the separation in terms of "hardness"

1. Hard process

Image credit: P. Skands

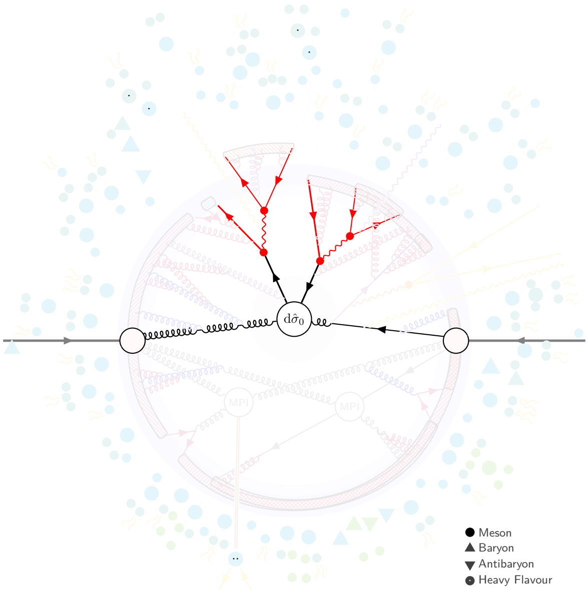


Event generation steps

Factorization theorems allow the separation in terms of "hardness"

1. Hard process
2. Resonance decays

Image credit: P. Skands

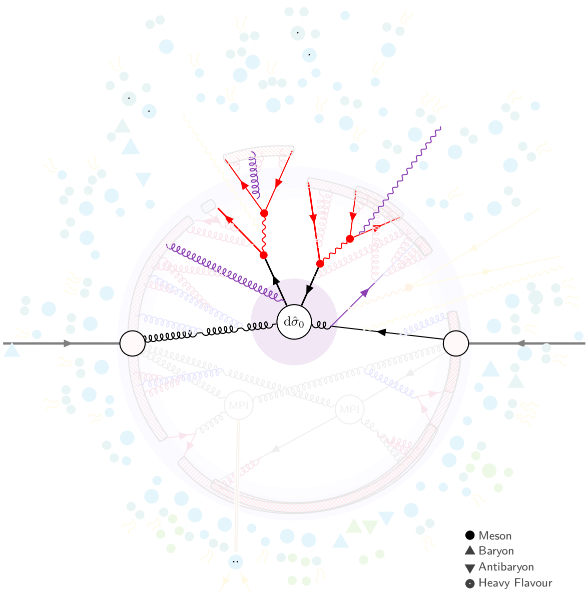


Event generation steps

Image credit: P. Skands

Factorization theorems allow the separation in terms of "hardness"

1. Hard process
2. Resonance decays
3. Matching, merging and matrix-element corrections

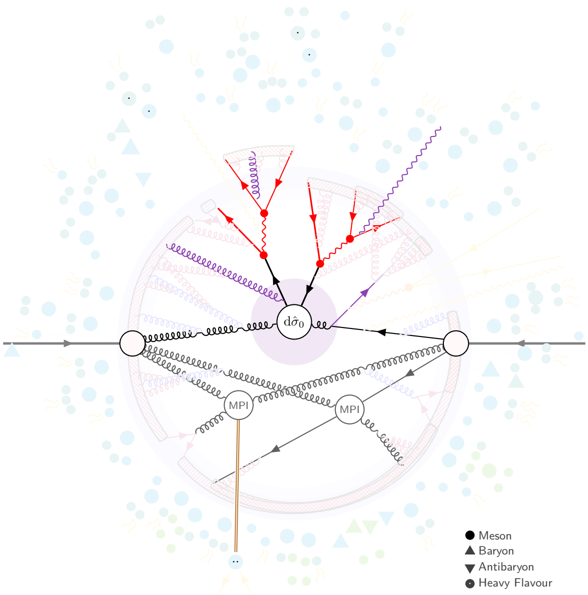


Event generation steps

Factorization theorems allow the separation in terms of "hardness"

1. Hard process
2. Resonance decays
3. Matching, merging and matrix-element corrections
4. Multiparton interactions

Image credit: P. Skands

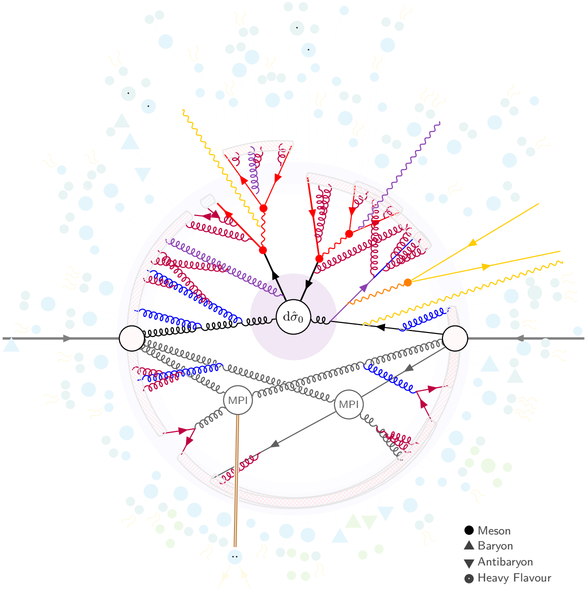


Event generation steps

Factorization theorems allow the separation in terms of "hardness"

1. Hard process
2. Resonance decays
3. Matching, merging and matrix-element corrections
4. Multiparton interactions
5. Parton showers: ISR, FSR, QED, weak

Image credit: P. Skands

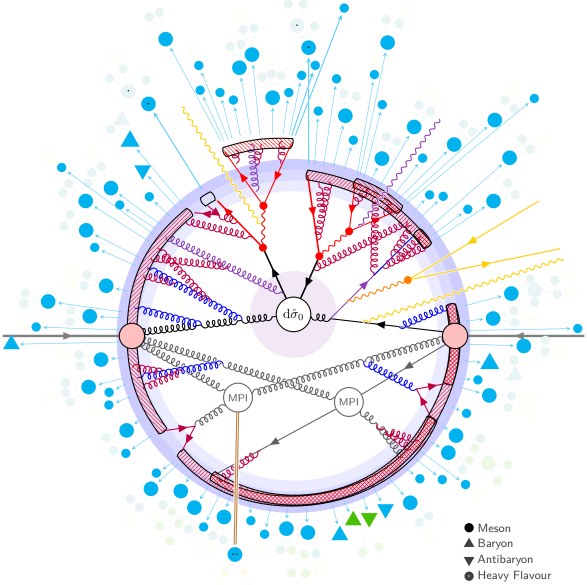


Event generation steps

Factorization theorems allow the separation in terms of "hardness"

1. Hard process
2. Resonance decays
3. Matching, merging and matrix-element corrections
4. Multiparton interactions
5. Parton showers: ISR, FSR, QED, weak
6. Hadronization, beam Remnants

Image credit: P. Skands

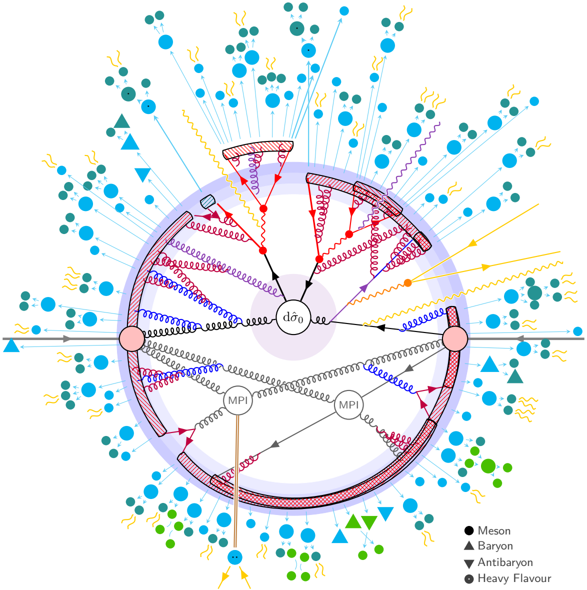


Event generation steps

Factorization theorems allow the separation in terms of "hardness"

1. Hard process
2. Resonance decays
3. Matching, merging and matrix-element corrections
4. Multiparton interactions
5. Parton showers: ISR, FSR, QED, weak
6. Hadronization, beam Remnants
7. Decays,rescattering

Image credit: P. Skands



Introduction

- What is Pythia?

  A general-purpose Monte Carlo event generator
- What is a general-purpose Monte Carlo event generator?

  Some transfer function between theory and experiment
- How does it work?

Introduction

- What is Pythia?

  A general-purpose Monte Carlo event generator
- What is a general-purpose Monte Carlo event generator?

  Some transfer function between theory and experiment
- How does it work?
- Have you used Pythia or any other MCEG?

Hands-on

Setup

- Install Docker Desktop, test with

```
$ docker --version
> Docker version 27.4.0, build bde2b89
```

- For Linux systems, some steps might be required: post-installation steps, re-launch terminal and Docker Desktop.
- Download the image for Pythia8 + Rivet4

```
$ docker pull hepstore/rivet-pythia
```

- Start the container, setting up a shared "host" directory between your local machine and the container

```
$ mkdir PythiaTutorial && cd PythiaTutorial
$ docker run -v $PWD:/host -it --rm hepstore/rivet-pythia
```

Open Pythia - Latest Online Manual and Pythia 8.3 worksheet, section 3 on page 4.

First run

- Go to the host directory, copy the build configuration files and make a new file called `mymain01.cc`
- You can see and open the files in your local directory

```
$ cd /host
$ cp /usr/local/share/Pythia8/examples/Makefile* .
$ touch mymain01.cc
```

- Fill the example file with the contents on page 4.
- Compile and run the example:

```
$ make mymain01 && ./mymain01
```

You should get a bunch of output from Pythia: a welcome card, initialization and process information, a bunch of particles in the event listing, and the message

```
0 no errors or warnings to report
```

Let's wait until everyone has gotten this far, then we move on to some examples.

mymain01.cc at the start

```
// Headers and Namespaces.
#include "Pythia8/Pythia.h" // Include Pythia headers.
using namespace Pythia8; // Let Pythia8:: be implicit.

int main() { // Begin main program.

    // Set up generation.
    Pythia pythia; // Declare a Pythia object
    pythia.readString("Top:gg2ttbar = on"); // Switch on process.
    pythia.readString("Beams:eCM = 8000."); // 8 TeV CM energy.
    pythia.init(); // Initialize; incoming pp beams is default.

    // Generate event(s).
    pythia.next(); // Generate an(other) event. Fill event record.

    return 0;
} // End main program with error-free return.
```

Exercise 1: Event record

- Let's go through the basics together, starting with the event record.
- Read through section 3 in the worksheet, and try to answer the problem: "see if you can find several copies of the top quarks, and check the status codes to figure out why each new copy has been added."

Exercise 1: Event record

- Let's go through the basics together, starting with the event record.
- Read through section 3 in the worksheet, and try to answer the problem: "see if you can find several copies of the top quarks, and check the status codes to figure out why each new copy has been added."
- If you follow the worksheet, you should have written output in the file `mymain01.log`. Now check this file for the event record.

```
no id name status mothers daughters colours p_x p_y p_z e m
...
 5 6 (t) -22 3 4 9 9 101 0 90.224 -14.092 183.317 269.072 174.515
 9 6 (t) -44 5 5 14 14 103 0 24.487 -44.352 12.944 182.090 174.421
 14 6 (t) -44 9 9 20 20 103 0 29.202 -49.636 14.203 184.230 174.421
 20 6 (t) -44 14 14 30 30 103 0 27.237 -55.269 13.092 185.447 174.421
...
```

Exercise 1: Event record

- Let's go through the basics together, starting with the event record.
- Read through section 3 in the worksheet, and try to answer the problem: "see if you can find several copies of the top quarks, and check the status codes to figure out why each new copy has been added."
- If you follow the worksheet, you should have written output in the file `mymain01.log`. Now check this file for the event record.

```
no id name status mothers daughters colours p_x p_y p_z e m
...
 5 6 (t) -22 3 4 9 9 101 0 90.224 -14.092 183.317 269.072 174.515
 9 6 (t) -44 5 5 14 14 103 0 24.487 -44.352 12.944 182.090 174.421
 14 6 (t) -44 9 9 20 20 103 0 29.202 -49.636 14.203 184.230 174.421
 20 6 (t) -44 14 14 30 30 103 0 27.237 -55.269 13.092 185.447 174.421
...
```

- The status codes are explained in the worksheet appendix A.
- The top quarks come from the hard process and go through initial-state radiation, beam remnant treatment and finally decay into W^+ and b.

Exercise 2: Event loop and particle loop

- Follow the instructions on section 4, halfway through page 6:
  - Include the process $q\bar{q} \to t\bar{t}$.
  - Introduce the event loop and particle loop.
  - Print out the transverse momentum and pseudorapidity of the final top-quark found in the event record.
  - Remember to build with `make mymain01` after changing the code.
- Some information in the worksheet might not work as expected in my Docker container, so please don't hesitate to ask!
- Work together! Share ideas and solutions, compare your results.

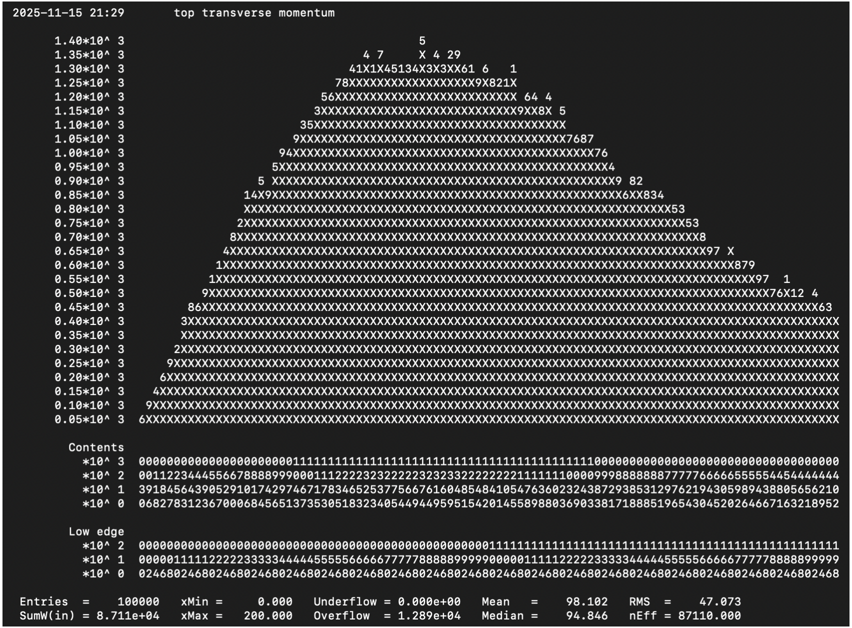
mymain01.cc after Exercise 2

```
// Headers and Namespaces.
#include "Pythia8/Pythia.h" // Include Pythia headers.
using namespace Pythia8; // Let Pythia8:: be implicit.

int main() { // Begin main program.

        // Set up generation.
        Pythia pythia; // Declare a Pythia object
        pythia.readString("Top:gg2ttbar = on"); // Switch on process.
        pythia.readString("Top:qqbar2ttbar = on"); // Another process.
        pythia.readString("Beams:eCM = 8000."); // 8 TeV CM energy.
        pythia.readString("Next:numberShowEvent = 5"); // How many events to show.
        pythia.init(); // Initialize; incoming pp beams is default.

        // Generate events.
        for (int iEvent = 0; iEvent < 5; ++iEvent) {
        pythia.next();
        int iTop = 0; // Integer to find last top quark in process.

                for (int i = 0; i < pythia.event.size(); ++i) {
                        cout << "i = " << i << ", id = " << pythia.event[i].id() << endl;
                        if (pythia.event[i].id() == 6) iTop = i;
                }
        cout << "Top pT  = " << pythia.event[iTop].pT() << endl;
        cout << "Top eta = " << pythia.event[iTop].eta() << endl;
        }

        pythia.stat();

        return 0;
} // End main program with error-free return.
```

Exercise 3: Histograms

- Continue the instructions on section 4.
- Declare and fill histograms for top quark p_{T} and η.
- Write out the histograms.
- Try to produce the histogram for charged multiplicity as well.

Exercise 3: pT histogram, 100K events



Exercise 3: η histogram, 100K events

```
2025-11-15 21:29       top pseudorapidity

      2.00*10^ 3                                       3  3               1    1
      1.92*10^ 3                                      3X18X8              X5685X123
      1.84*10^ 3                                   3  XXXXXX8327 18     88XXXXXXXXX75
      1.76*10^ 3                                  8X59XXXXXXXXXX7XX 4 18XXXXXXXXXXXXXX
      1.68*10^ 3                                 8XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX2
      1.60*10^ 3                                7XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      1.52*10^ 3                               3XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX6 2
      1.44*10^ 3                              3XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX9X
      1.36*10^ 3                             1XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX9
      1.28*10^ 3                             XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX3
      1.20*10^ 3                            9XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      1.12*10^ 3                           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX8
      1.04*10^ 3                          1XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX4
      0.96*10^ 3                         5XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      0.88*10^ 3                        1XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX7
      0.80*10^ 3                       2XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      0.72*10^ 3                       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      0.64*10^ 3                      5XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX6
      0.56*10^ 3                     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX5
      0.48*10^ 3                   3XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX62
      0.40*10^ 3                  3XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX5
      0.32*10^ 3                18XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX65
      0.24*10^ 3              23XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX62
      0.16*10^ 3           539XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX7621
      0.08*10^ 3  32344879XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX97745423

        Contents
          *10^ 3  0000000000000000000000000111111111111111111111111111111111111111111111111110000000000000000000000000
          *10^ 2  0000000001111123344557899112345677778989998778778676678898898988887764443219887654432221110000000000
          *10^ 1  2123365771057840427693017298867648156450402871362817942227908245619618355209752014258707329875534312
          *10^ 0  3763414486713157656386891305273372833515324748495028012089051756478034270462892867079669975429812570

        Low edge  --------------------------------------------------
          *10^ 0  5444444444433333333332222222222111111111100000000000000000001111111111222222222233333333334444444444
          *10^-1  0987654321098765432109876543210987654321098765432101234567890123456789012345678901234567890123456789
          *10^-2  0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

Exercise 4: Input files

- Follow section 5 to run your code using a command file.
- The benefit is that you no longer need to build between different runs, since all changes should be done to command file.
- Try to produce charged multiplicity histograms with 3 setups:
  1. ISR, FSR and MPI turned on.
  2. ISR and FSR turned on, MPI turned off.
  3. ISR, FSR and MPI turned off
- Save results in output log files with

```
$ ./mymain01 mymain01.cmnd > mymain01_ISR-on_FSR-on_MPI-on.log
$ ./mymain01 mymain01.cmnd > mymain01_ISR-on_FSR-on_MPI-off.log
$ ./mymain01 mymain01.cmnd > mymain01_ISR-off_FSR-off_MPI-off.log
```

Exercise 4: N_{ch}, 100K events

```
2025-11-15 23:59       charged multiplicity

      2.32*10^ 3                                                       1
      2.24*10^ 3                                                   8   X5 4
      2.16*10^ 3                                                 2XX794XX X6692
      2.08*10^ 3                                                4XXXXXXXX9XXXXX41
      2.00*10^ 3                                              XXXXXXXXXXXXXXXXXXX4
      1.92*10^ 3                                           726XXXXXXXXXXXXXXXXXXXX
      1.84*10^ 3                                          4XXXXXXXXXXXXXXXXXXXXXXX98
      1.76*10^ 3                                        52XXXXXXXXXXXXXXXXXXXXXXXXXX46
      1.68*10^ 3                                       1XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      1.60*10^ 3                                    1 9XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX5
      1.52*10^ 3                                    X9XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX833
      1.44*10^ 3                                   8XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      1.36*10^ 3                                  6XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX83
      1.28*10^ 3                                 9XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      1.20*10^ 3                                6XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      1.12*10^ 3                               5XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX3
      1.04*10^ 3                               XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX83
      0.96*10^ 3                             3XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX4
      0.88*10^ 3                             XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX1
      0.80*10^ 3                            8XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX7
      0.72*10^ 3                            XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX4
      0.64*10^ 3                           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX9
      0.56*10^ 3                          7XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX61
      0.48*10^ 3                         6XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      0.40*10^ 3                        7XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX73
      0.32*10^ 3                       7XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX434
      0.24*10^ 3                      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX94
      0.16*10^ 3                   34XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX853511
      0.08*10^ 3              1336XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

        Contents
          *10^ 3  0000000000000000000000000000111111111111111222222222222222222222111111111111110000000000000000000000
          *10^ 2  0000000000000000111223456799012345556767888000012111122011110009887755443310099876655443322221111100
          *10^ 1  0000000000001224701639753480587732309119995800295235140682249315220250664096281076262907476739420288
          *10^ 0  0000000000321247640273214070701212082074354411889535493999499321978465162177403968937019421102126287

        Low edge  -
          *10^ 2  0000000000000000000000000011111111111111111111111112222222222222222222222222333333333333333333333333
          *10^ 1  0001112233344555667778899900111223334455566777889990011122333445556677788999001112233344555667778899
          *10^ 0  1371593715937159371593715937159371593715937159371593715937159371593715937159371593715937159371593715
```

Exercise 4: N_{ch} histogram, MPI off, 100K events

```
2025-11-16 00:21       charged multiplicity

      6.50*10^ 3                             1
      6.25*10^ 3                            5X6
      6.00*10^ 3                            XXX5
      5.75*10^ 3                           6XXXX8
      5.50*10^ 3                          2XXXXXX8
      5.25*10^ 3                          XXXXXXXX
      5.00*10^ 3                          XXXXXXXX1
      4.75*10^ 3                         XXXXXXXXXX
      4.50*10^ 3                         XXXXXXXXXX2
      4.25*10^ 3                        8XXXXXXXXXXX
      4.00*10^ 3                        XXXXXXXXXXXX3
      3.75*10^ 3                        XXXXXXXXXXXXX
      3.50*10^ 3                       XXXXXXXXXXXXXX1
      3.25*10^ 3                       XXXXXXXXXXXXXXX
      3.00*10^ 3                      8XXXXXXXXXXXXXXX
      2.75*10^ 3                      XXXXXXXXXXXXXXXX7
      2.50*10^ 3                     8XXXXXXXXXXXXXXXXX2
      2.25*10^ 3                     XXXXXXXXXXXXXXXXXXX
      2.00*10^ 3                    7XXXXXXXXXXXXXXXXXXX
      1.75*10^ 3                    XXXXXXXXXXXXXXXXXXXXX
      1.50*10^ 3                   8XXXXXXXXXXXXXXXXXXXXXX
      1.25*10^ 3                  2XXXXXXXXXXXXXXXXXXXXXXX9
      1.00*10^ 3                  XXXXXXXXXXXXXXXXXXXXXXXXX6
      0.75*10^ 3                 9XXXXXXXXXXXXXXXXXXXXXXXXXX93
      0.50*10^ 3               2XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX63
      0.25*10^ 3           1257XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX8653322111

        Contents
          *10^ 3  0000000000000000111223445566655544332211100000000000000000000000000000000000000000000000000000000000
          *10^ 2  0000000000011347049494172612186473826274297533111000000000000000000000000000000000000000000000000000
          *10^ 1  0000000002528091441558949428578470377959101792941885431110000000000000000000000000000000000000000000
          *10^ 0  0000000146972283477088621737539026114635607332794421066717433220201100000000000000000000000000000000

        Low edge  -
          *10^ 2  0000000000000000000000000011111111111111111111111112222222222222222222222222333333333333333333333333
          *10^ 1  0001112233344555667778899900111223334455566777889990011122333445556677788999001112233344555667778899
          *10^ 0  1371593715937159371593715937159371593715937159371593715937159371593715937159371593715937159371593715

 Entries  =     100000   xMin =    -1.000   Underflow = 0.000e+00   Mean   =    113.77   RMS  =     26.35
 SumW(in) = 1.000e+05   xMax =   399.000   Overflow  = 0.000e+00   Median =    112.62   nEff = 1.000e+05
```

Exercise 4: N_{ch} histogram, parton showers off, MPI off, 100K events

```
2025-11-16 00:15       charged multiplicity

      1.30*10^ 4                  3
      1.25*10^ 4                 2X
      1.20*10^ 4                 XX9
      1.15*10^ 4                 XXX
      1.10*10^ 4                 XXX
      1.05*10^ 4                XXXX9
      1.00*10^ 4                XXXXX
      0.95*10^ 4                XXXXX
      0.90*10^ 4                XXXXX
      0.85*10^ 4               9XXXXX
      0.80*10^ 4               XXXXXX5
      0.75*10^ 4               XXXXXXX
      0.70*10^ 4               XXXXXXX
      0.65*10^ 4              5XXXXXXX
      0.60*10^ 4              XXXXXXXX
      0.55*10^ 4              XXXXXXXX7
      0.50*10^ 4              XXXXXXXXX
      0.45*10^ 4             1XXXXXXXXX
      0.40*10^ 4             XXXXXXXXXX
      0.35*10^ 4             XXXXXXXXXX1
      0.30*10^ 4             XXXXXXXXXXX
      0.25*10^ 4            9XXXXXXXXXXX
      0.20*10^ 4            XXXXXXXXXXXX2
      0.15*10^ 4           8XXXXXXXXXXXXX
      0.10*10^ 4          3XXXXXXXXXXXXXX3
      0.05*10^ 4        25XXXXXXXXXXXXXXXX51

        Contents
          *10^ 4  0000000000000011111000000000000000000000000000000000000000000000000000000000000000000000000000000000
          *10^ 3  0000000001246802210753100000000000000000000000000000000000000000000000000000000000000000000000000000
          *10^ 2  0000000263402451694730562000000000000000000000000000000000000000000000000000000000000000000000000000
          *10^ 1  0000018438676610454736857721000000000000000000000000000000000000000000000000000000000000000000000000

        Low edge  -
          *10^ 2  0000000000000000000000000011111111111111111111111112222222222222222222222222333333333333333333333333
          *10^ 1  0001112233344555667778899900111223334455566777889990011122333445556677788999001112233344555667778899
          *10^ 0  1371593715937159371593715937159371593715937159371593715937159371593715937159371593715937159371593715
```

Exercise 5a: Rivet

- Now we know the basics, but a more realistic analysis could use some real-life data.
- Let's take a small detour from the worksheet and try to run some comparisons to experimental data using Rivet.
- Copy and inspect the program main144.cc, and the associated run card main144.cmnd

```
$ cp /usr/local/share/Pythia8/examples/main144.c* .
```

Exercise 5a: Rivet

- Modify the main144.cmnd to run at $\sqrt{s} = 8$ TeV, including hard process and a phase-space cut

```
Beams:eCM = 8000. ! CM energy of collision
SoftQCD:all = off ! All soft QCD processes
HardQCD:all = on ! All hard QCD processes
PhaseSpace:ptHatMin = 50.
```

- Activate Rivet plugin within Pythia, provide an inclusive-jet analysis from ATLAS.

```
Main:runRivet = on
Main:rivetAnalyses = {ATLAS_2017_I1604271}
```

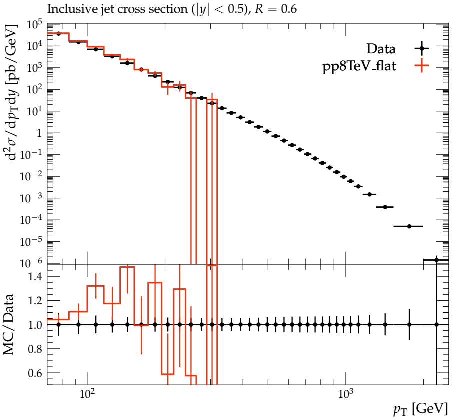
- Run and save Rivet output to file, plot using Rivet

```
$ ./main144 -c main144.cmnd -n 10000 -o pp8TeV_flat # wait until comple
$ rivet-mkhtml pp8TeV_flat.yoda
```

- Inspect by opening the file `rivet-plots/index.html` in a web browser, outside the container

Exercise 5a: Rivet

- The differential cross section scales like
$$\frac{\mathrm{d}\sigma}{\mathrm{d}\hat{p}_T^2} \propto \frac{1}{\hat{p}_T^n},$$
- where $n = 4 - 6$.
- Huge span in magnitude of cross section, need to bias phase-space sampling with $\hat{p}_T$.



Exercise 5a: Rivet

- Next challenge is to use phase-space biasing in the sampling of the hard process. This way, more events will be generated at high-pT, and reweighted to retain the correct cross section.

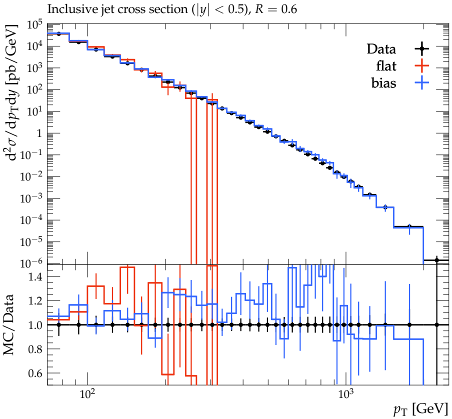
```
PhaseSpace:bias2Selection = on
PhaseSpace:bias2SelectionPow = 4.
PhaseSpace:bias2SelectionRef = 50.
```

- Produce a comparison to the previous run which had flat phase-space sampling in terms of p_{T}.

```
$ ./main144 -c main144.cmnd -n 10000 -o pp8TeV_bias
$ rivet-mkhtml pp8TeV_flat.yoda:"Title=flat" \
pp8TeV_bias.yoda:"Title=bias"
```

Exercise 5a: Rivet

- Whole phase-space filled, events reweighted to preserve physical cross-section.



Exercise 5b: Rivet Z-boson p_T

- The previous exercise was a good example where there's nothing wrong with the events that are generated, but we just didn't fill enough of the phase space.
- This next exercise aims to showcase how how one might generate events that don't describe the physics you're aiming to simulate, emphasizing the importance of ISR for Z-boson transverse momentum.
- Change the beam energy, turn Z-production on and other processes off

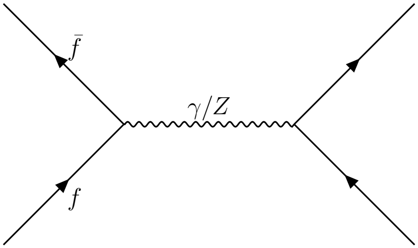
```
Beams:eCM = 7000. ! CM energy of collision
SoftQCD:all = off ! All soft QCD processes
HardQCD:all = off ! All soft QCD processes
WeakSingleBoson:ffbar2gmZ = on  ! Gamma/Z production on
```

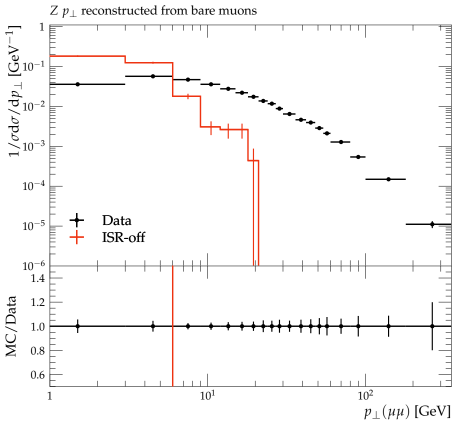
- Turn off ISR, and remove the phase-space biasing

```
PartonLevel:ISR=off
! PhaseSpace:bias2Selection = on
! PhaseSpace:bias2SelectionPow = 4.
! PhaseSpace:bias2SelectionRef = 50.
```

Exercise 5b: Rivet Z-boson p_T

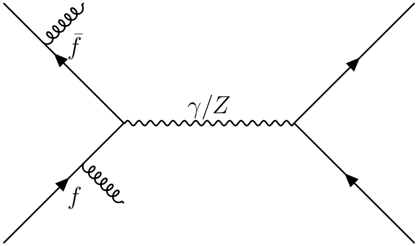
- Add the Z-boson analysis `ATLAS_2011_S9131140`
- The effect is similar: p_T distribution cuts off at some point, but this time the overall shape of the cross section is also wrong.

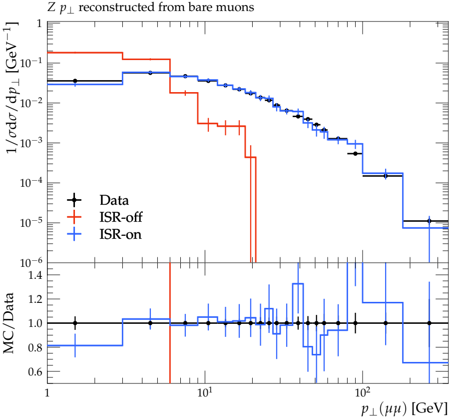




Exercise 5b: Rivet Z-boson p_T

- Without ISR, the Z-boson doesn't acquire much p_T in LO picture. ISR provides p_T kicks through emissions.





Exercises 6-?:

- Congratulations! You now have your drivers license for Pythia, and should be able to tackle the rest of the problems in the worksheet, in any order you wish.
  - Section 6: Jets
  - Section 7: Higgs production
  - Section 8: CKKW-L merging
  - Section 9: Further studies
- Some topics we have covered already which you could try in practice are
  - Factorization and renormalization scale variations (envelope) with SigmaProcess:factorMultFac/renormMultFac
  - Different PDF sets with PDF:pSet
  - Matching and merging: main164.cc and associated .cmnd files

Resources

- pythia.org Online Manual - Very important, especially for basics.
- A comprehensive guide to the physics and usage of PYTHIA 8.3 - More advanced and detailed guide.
- Pythia 8.3 Worksheet - Beginner-friendly tutorial.

thanks