Pythia 8.3 Python Worksheet

Written by:

- Torbjörn Sjöstrand (Department of Astronomy and Theoretical Physics, Lund University)
- Peter Skands (School of Physics, Monash University)
- Stefan Prestel
- Philip Ilten (School of Physics, University of Cincinnati)
- Leif Gellerson (Department of Astronomy and Theoretical Physics, Lund University)

The Pythia 8.3 program is a standard tool for the generation of high-energy collisions (specifically, it focuses on centre-of-mass energies greater than about 10 GeV), comprising a coherent set of physics models for the evolution from a few-body high-energy ("hard") scattering process to a complex multihadronic final state. The particles are produced in vacuum. Simulation of the interaction of the produced particles with detector material is not included in Pythia but can, if needed, be done by interfacing to external detector-simulation codes.

The Pythia 8.3 code package contains a library of hard interactions and models for initial- and final-state parton showers, multiple parton-parton interactions, beam remnants, string fragmentation and particle decays. It also has a set of utilities and interfaces to external programs.

The objective of this exercise is to teach you the basics of how to use the Pythia 8.3 event generator to study various physics aspects. As you become more familiar you will better understand the tools at your disposal, and can develop your own style to use them. Within this first exercise it is not possible to describe the physics models used in the program; for this we refer to the Pythia 8.3 manual, and to all the further references found in them. For practical usage, the Pythia 8.3 HTML manual provides full documentation on running and configuring Pythia. A Pythia 8.3 Doxygen manual is also available, although the HTML manual should always be used as the primary documentation.

Requirements

Before running this notebook, we need to set up our environment. First, we install and import the `wurlitzer` module. This allows programs that have C-like backends to write their output to the

Python console. In short, this allows the output of Pythia to be displayed in this notebook.

```
# Redirect the C output of Pythia to the notebook.
!pip install wurlitzer
from wurlitzer import sys_pipes_forever

sys_pipes_forever()
```

```
Requirement already satisfied: wurlitzer in /usr/local/lib/python3.11/dist-packages
```

Next, we need to install the Pythia module.

```
# Install and import the Pythia module.
!pip install pythia8mc
import pythia8mc as pythia8
```

```
Collecting pythia8mc
  Downloading pythia8mc-8.315.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86
Downloading pythia8mc-8.315.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_6
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 29.9/29.9 MB 23.6 MB/s eta 0:00:00
Installing collected packages: pythia8mc
Successfully installed pythia8mc-8.315.0
```

Introduction

Pythia 8 is, by today's standards, a small package. It is completely self-contained, and is therefore easy to install for standalone usage, e.g. if you want to have it on your own laptop, or if you want to explore physics or debug code without any danger of destructive interference between different libraries. This worksheet uses Pythia's Python interface, using an installation included in the Docker container. If you would like to run Pythia 8.3 natively using C++, please take a look at the analogous standalone worksheet, which includes installation instructions. You can download the latest version at pythia.org.

When you use Pythia you are expected to write the main program yourself, for maximal flexibility and power. Several examples of such main programs are included with the code, to illustrate common tasks and help getting started. The section A Hello World Program gives you a simple step-by-step recipe how to write a minimal main program, that can then gradually be expanded in different directions, e.g. as in A first realistic analysis.

In the section Input files you will see how the parameters of a run can be read in from a file, so that the main program can be kept fixed. Many of the provided C++ main programs therefore allow you to create executables that can be used for different physics studies without recompilation, but potentially at the cost of some flexibility. For Python usage, this is not an issue.

The final three sections provide suggestions for optional further studies, and can be addressed in any order. CKKW-L merging deals with the important topic of merging of external matrix-element input of different orders, introducing the CKKW-L scheme as a suitable starting point. Some studies of Higgs production describes how you can explore various physics aspects of the Standard Model Higgs production and decay. Further studies, finally, collects suggestions for a few diverse studies.

While Pythia can be run standalone, it can also be interfaced with a set of other libraries. One example is `HepMC`, which is the standard format used by experimentalists to store generated events. Further main programs included with the Pythia code provide examples of linking, e.g., to `AlpGen`, `MadGraph`, `PowHeg`, `FastJet`, `ROOT`, and the Les Houches Accords LHEF, LHAPDF and SLHA.

The Event Record contains a brief summary of the event-record structure.

Before running this notebook, we need to set up our environment. First, we install and import the `wurlitzer` module. This allows programs that have C-like backends to write their output to the Python console. In short, this allows the output of Pythia to be displayed in this notebook.

A Hello World Program

We will now generate a single $gg \to t\bar{t}$ event at the LHC, using Pythia's Python interface. The following example contains explanatory comments.

```
# Import the Pythia module.
import pythia8mc as pythia8

# Create a Pythia object.
pythia = pythia8.Pythia()
```

```
 *-------------------------------------------------------------------------
 |
 |  *----------------------------------------------------------------------
 |  |
 |  |
 |  |   PPP   Y   Y  TTTTT  H   H  III    A      Welcome to the Lund Monte Carlo!
 |  |   P  P   Y Y     T    H   H   I    A A     This is PYTHIA version 8.315
 |  |   PPP     Y      T    HHHHH   I   AAAAA    Last date of change: 27 May 2025
 |  |   P       Y      T    H   H   I   A   A
 |  |   P       Y      T    H   H  III  A   A    Now is 03 Jul 2025 at 17:36:11
 |  |
 |  |   Program documentation and an archive of historic versions is found on:
 |  |
 |  |                               https://pythia.org/
 |  |
 |  |   PYTHIA is authored by a collaboration consisting of:
 |  |
 |  |   Javira Altmann, Christian Bierlich, Naomi Cooke, Nishita Desai,
 |  |   Ilkka Helenius, Philip Ilten, Leif Lonnblad, Stephen Mrenna,
 |  |   Christian Preuss, Torbjorn Sjostrand, and Peter Skands.
 |  |
 |  |   The complete list of authors, including contact information and
```

```
 |  |   affiliations, can be found on https://pythia.org/.
 |  |   Problems or bugs should be reported on email at authors@pythia.org.
 |  |
 |  |   The main program reference is C. Bierlich et al,
 |  |   'A comprehensive guide to the physics and usage of Pythia 8.3',
 |  |   SciPost Phys. Codebases 8-r8.3 (2022) [arXiv:2203.11601 [hep-ph]]
 |  |
 |  |   PYTHIA is released under the GNU General Public Licence version 2
 |  |   or later. Please respect the MCnet Guidelines for Generator Authors
 |  |   and Users.
 |  |
 |  |   Disclaimer: this program comes without any guarantees.
 |  |   Beware of errors and use common sense when interpreting results.
 |  |
 |  |   Copyright (C) 2025 Torbjorn Sjostrand
 |  |
 |  |
 |  *--------------------------------------------------------------------------------
 |
 *----------------------------------------------------------------------------------
```

```
# Configure the Pythia object.
pythia.readString("Top:gg2ttbar = on")  # Switch on process.
pythia.readString("Beams:eCM = 14000.")  # Set 14 TeV CM energy.
```

```
True
```

```
# Initialize, incoming pp beams are default.
pythia.init()
```

```
 *-------  PYTHIA Process Initialization  --------------------------*
 |                                                                  |
 | We collide p+ with p+ at a CM energy of 1.400e+04 GeV            |
 |                                                                  |
 |------------------------------------------------------------------|
 |                                                    |             |
 | Subprocess                                    Code |   Estimated |
 |                                                    |    max (mb) |
 |                                                    |             |
 |------------------------------------------------------------------|
 |                                                    |             |
 | g g -> t tbar                                  601 |   9.089e-06 |
 |                                                                  |
 *-------  End PYTHIA Process Initialization -----------------------*

 *-------  PYTHIA Multiparton Interactions Initialization  ---------*
 |                                                                  |
 |                   sigmaNonDiffractive =    57.17 mb              |
 |                                                                  |
 |    pT0 =  2.65 gives sigmaInteraction =   328.84 mb: accepted    |
 |                                                                  |
 *-------  End PYTHIA Multiparton Interactions Initialization  -----*

 *-------  PYTHIA Flag + Mode + Parm + Word + FVec + MVec + PVec + WVec Settings (c
```

```
 |
 | Name                                              |                        Now |
 |                                                   |                            |
 | StringZ:useOldAExtra                              |                         on |
 | Top:gg2ttbar                                      |                         on |
 |
 *-------  End PYTHIA Flag + Mode + Parm + Word + FVec + MVec + PVec + WVec Setting

 --------  PYTHIA Particle Data Table (changed only)  -----------------------------

      id   name            antiName         spn chg col      m0        mWidth
             no onMode   bRatio   meMode     products

 no particle data has been changed from its default value

 --------  End PYTHIA Particle Data Table  -----------------------------------------

True
```

```
# Generate an(other) event. Fill event record.
pythia.next()
```

```
 --------  PYTHIA Info Listing  ----------------------------------------

 Beam A: id =   2212, pz =  7.000e+03, e =  7.000e+03, m =  9.383e-01.
 Beam B: id =   2212, pz = -7.000e+03, e =  7.000e+03, m =  9.383e-01.

 In 1: id =   21, x =  3.574e-03, pdf =  1.690e+01 at Q2 =  3.680e+04.
 In 2: id =   21, x =  2.582e-01, pdf =  1.538e-01 at same Q2.

 Subprocess g g -> t tbar with code 601 is 2 -> 2.
 It has sHat =  1.809e+05,    tHat = -2.327e+04,    uHat = -9.623e+04,
       pTHat =  8.471e+01,   m3Hat =  1.782e+02,   m4Hat =  1.721e+02,
    thetaHat =  7.791e-01,  phiHat =  3.709e+00.
     alphaEM =  7.885e-03,  alphaS =  1.162e-01    at Q2 =  3.786e+04.

 Impact parameter b =  9.121e-01 gives enhancement factor =  8.273e-01.
 Max pT scale for MPI =  1.400e+04, ISR =  1.400e+04, FSR =  1.400e+04.
 Number of MPI =     6, ISR =    15, FSRproc =    49, FSRreson =    24.

 --------  End PYTHIA Info Listing  ------------------------------------

 --------  PYTHIA Event Listing  (hard process)  -----------------------------------

    no         id  name            status     mothers   daughters     colours
     0         90  (system)           -11     0     0     0     0     0     0
     1       2212  (p+)               -12     0     0     3     0     0     0
     2       2212  (p+)               -12     0     0     4     0     0     0
     3         21  (g)                -21     1     0     5     6   101   102
     4         21  (g)                -21     2     0     5     6   102   103
     5          6  (t)                -22     3     4     7     8   101     0    -7
     6         -6  (tbar)             -22     3     4     9    10     0   103     7
     7         24  (W+)               -22     5     0    11    12     0     0    -2
     8          5  b                   23     5     0     0     0   101     0    -4
     9        -24  (W-)               -22     6     0    13    14     0     0     5
    10         -5  bbar                23     6     0     0     0     0   103     1
```

```
    11        -15  tau+                23     7     0     0     0     0     0    -4
    12         16  nu_tau              23     7     0     0     0     0     0     2
    13          3  s                   23     9     0     0     0   104     0     3
    14         -4  cbar                23     9     0     0     0     0   104     2
                                  Charge sum:  0.000          Momentum sum:       -

 --------  End PYTHIA Event Listing  -----------------------------------------

 --------  PYTHIA Event Listing  (complete event)  ---------------------------

    no         id  name            status     mothers   daughters     colours
     0         90  (system)           -11     0     0     0     0     0     0
     1       2212  (p+)               -12     0     0   303     0     0     0
     2       2212  (p+)               -12     0     0   304     0     0     0
     3         21  (g)                -21     7     0     5     6   101   102
     4         21  (g)                -21     8     8     5     6   102   103
     5          6  (t)                -22     3     4     9     9   101     0    -7
     6         -6  (tbar)             -22     3     4    10    10     0   103     7
     7         21  (g)                -41    12    12    11     3   101   105
     8         21  (g)                -42    13     0     4     4   102   103     -
     9          6  (t)                -44     5     5    14    14   101     0    -2
    10         -6  (tbar)             -44     6     6    15    15     0   103     8
    11         21  (g)                -43     7     0    16    16   102   105    -6
```

Now you can study the output, especially the example of a complete event record (preceded by initialization information, and by kinematical-variable and hard-process listing for the same event). At this point you need to turn to the [The Event Record](#) for a brief overview of the information stored in the event record. Documentation of the event record is can also be found under [The Event Record page](#) of the HTML manual. In Python, it is also possible to access documentation by the standard Python `help` method. Note, however, that this does not provide detailed documentation like the HTML manual does.

```
# Print documentation for the Event class in Pythia.
help(pythia.event)
```

```
Help on Event in module pythia8mc object:

class Event(pybind11_builtins.pybind11_object)
 |  Method resolution order:
 |      Event
 |      pybind11_builtins.pybind11_object
 |      builtins.object
 |
 |  Methods defined here:
 |
 |  REtaPhi(...)
 |      REtaPhi(self: pythia8mc.Event, i1: int, i2: int) -> float
 |
 |      C++: Pythia8::Event::REtaPhi(int, int) const --> double
 |
 |  RRapPhi(...)
 |      RRapPhi(self: pythia8mc.Event, i1: int, i2: int) -> float
 |
```

```
 |      C++: Pythia8::Event::RRapPhi(int, int) const --> double
 |
 |  __getitem__(...)
 |      __getitem__(self: pythia8mc.Event, i: int) -> pythia8mc.Particle
 |
 |      C++: Pythia8::Event::operator[](int) --> class Pythia8::Particle &
 |
 |  __iadd__(...)
 |      __iadd__(self: pythia8mc.Event, addEvent: pythia8mc.Event) -> pythia8mc.Eve
 |
 |      C++: Pythia8::Event::operator+=(const class Pythia8::Event &) --> class Pyt
 |
 |  __init__(...)
 |      __init__(*args, **kwargs)
 |      Overloaded function.
 |
 |      1. __init__(self: pythia8mc.Event) -> None
 |
 |      doc
 |
 |      2. __init__(self: pythia8mc.Event, capacity: int) -> None
 |
 |      3. __init__(self: pythia8mc.Event, arg0: pythia8mc.Event) -> None
 |
 |  append(...)
 |      append(*args, **kwargs)
 |      Overloaded function.
 |
 |      1. append(self: pythia8mc.Event, entryIn: pythia8mc.Particle) -> int
 |
 |      C++: Pythia8::Event::append(class Pythia8::Particle) --> int
 |
 |      2. append(self: pythia8mc.Event, id: int, status: int, mother1: int, mother
 |
 |      3. append(self: pythia8mc.Event, id: int, status: int, mother1: int, mother
 |
 |      4. append(self: pythia8mc.Event, id: int, status: int, mother1: int, mother
 |
 |      5. append(self: pythia8mc.Event, id: int, status: int, mother1: int, mother
 |
```

An important part of the event record is that many copies of the same particle may exist, but only those with a positive status code are still present in the final state. To exemplify, consider a top quark produced in the hard interaction, initially with positive status code. When later a shower branching $\mathrm{t} \to \mathrm{tg}$ occurs, the new t and g are added at the bottom of the then-current event record, but the old t is not removed. It is marked as decayed, however, by negating its status code. At any stage of the shower there is thus only one "current" copy of the top. After the shower, when the final top decays, $\mathrm{t} \to \mathrm{bW}^+$, also that copy receives a negative status code. When you understand the basic principles, 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. Also note how the mother/daughter indices tie together the various copies.

A first realistic analysis

We will now gradually expand the skeleton program from above, towards what would be needed for a more realistic analysis setup. In the following input cell, we have consolidated all the necessary code for ease of editing and use. For each suggestion below, simply copy the relevant code here and rerun the input cell.

```
# Import the Pythia module.
import pythia8mc as pythia8

# Create a Pythia object.
pythia = pythia8.Pythia()

# Configure the Pythia object.
pythia.readString("Top:gg2ttbar = on")  # Switch on process.
pythia.readString("Beams:eCM = 14000.")  # Set 14 TeV CM energy.

# Initialize, incoming pp beams are default.
pythia.init()

# Generate an(other) event. Fill event record.
pythia.next()
```

```
 *-------------------------------------------------------------------------------
 |
 |  *----------------------------------------------------------------------------
 |  |
 |  |
 |  |   PPP   Y   Y  TTTTT  H   H  III    A      Welcome to the Lund Monte Carlo!
 |  |   P  P   Y Y     T    H   H   I    A A     This is PYTHIA version 8.315
 |  |   PPP     Y      T    HHHHH   I   AAAAA    Last date of change: 27 May 2025
 |  |   P       Y      T    H   H   I   A   A
 |  |   P       Y      T    H   H  III  A   A    Now is 03 Jul 2025 at 17:36:12
 |  |
 |  |   Program documentation and an archive of historic versions is found on:
 |  |
 |  |                               https://pythia.org/
 |  |
 |  |   PYTHIA is authored by a collaboration consisting of:
 |  |
 |  |   Javira Altmann, Christian Bierlich, Naomi Cooke, Nishita Desai,
 |  |   Ilkka Helenius, Philip Ilten, Leif Lonnblad, Stephen Mrenna,
 |  |   Christian Preuss, Torbjorn Sjostrand, and Peter Skands.
 |  |
 |  |   The complete list of authors, including contact information and
 |  |   affiliations, can be found on https://pythia.org/.
 |  |   Problems or bugs should be reported on email at authors@pythia.org.
 |  |
 |  |   The main program reference is C. Bierlich et al,
 |  |   'A comprehensive guide to the physics and usage of Pythia 8.3',
 |  |   SciPost Phys. Codebases 8-r8.3 (2022) [arXiv:2203.11601 [hep-ph]]
 |  |
 |  |   PYTHIA is released under the GNU General Public Licence version 2
```

```
 |  |   or later. Please respect the MCnet Guidelines for Generator Authors
 |  |   and Users.
 |  |
 |  |   Disclaimer: this program comes without any guarantees.
 |  |   Beware of errors and use common sense when interpreting results.
 |  |
 |  |   Copyright (C) 2025 Torbjorn Sjostrand
 |  |
 |  |
 |  *------------------------------------------------------------------------------
 |
 *--------------------------------------------------------------------------------
```

```
 *-------  PYTHIA Process Initialization  --------------------------*
 |                                                                  |
 | We collide p+ with p+ at a CM energy of 1.400e+04 GeV            |
 |                                                                  |
 |------------------------------------------------------------------|
 |                                                    |             |
 | Subprocess                                    Code |   Estimated |
 |                                                    |    max (mb) |
 |                                                    |             |
 |------------------------------------------------------------------|
 |                                                    |             |
 | g g -> t tbar                                  601 |   9.089e-06 |
 |                                                                  |
```

- Often, we wish to mix several processes together. To add the process $\mathrm{q\bar{q}} \to \mathrm{t\bar{t}}$ to the above example, just include a third `pythia.readString` call

  ```
  pythia.readString("Top:qqbar2ttbar = on")
  ```

  Note that the `Pythia` object must be reinitialized whenever a new configuration is added.

- Now we wish to generate more than one event. To do this, introduce a loop around `pythia.next()`, so the code now reads as the following input cell. If you have already created and initialized the `Pythia` object, you can simply run this cell to see the output.

```
for iEvent in range(5):
    pythia.next()
```

> Hereafter, we will call this the event loop. The program will now generate 5 events; each call to `pythia.next()` resets the event record and fills it with a new event. To list more of the events, you also need to add
>
> ```
> pythia.readString("Next:numberShowEvent = 5")
> ```

> along with the other `pythia.readString` commands, or simply add `pythia.event.list()` after `pythia.next()`.

- To obtain statistics on the number of events generated of the different kinds, and the estimated cross sections, add the following, just before the end of the program. If you have already generated a few events, you can just run the input cell below.

```
pythia.stat()
```

```
 *-------  PYTHIA Event and Cross Section Statistics  ----------------------------
 |
 | Subprocess                                     Code |            Number of events
 |                                                     |       Tried   Selected   Ac
 |                                                     |
 |---------------------------------------------------------------------------------
 |                                                     |
 | g g -> t tbar                                   601 |          75          6
 |                                                     |
 | sum                                                 |          75          6
 |
 *-------  End PYTHIA Event and Cross Section Statistics -------------------------

 *-------  PYTHIA Error and Warning Messages Statistics  -------------------------
 |
 |  times   message
 |
 |      0   no errors or warnings to report
 |
 *-------  End PYTHIA Error and Warning Messages Statistics  ----------------------
```

- During the run you may receive problem messages. These come in three kinds:
  - a *warning* is a minor problem that is automatically fixed by the program, at least approximately;
  - an *error* is a bigger problem, that is normally still automatically fixed by the program, by backing up and trying again;
  - an *abort* is such a major problem that the current event could not be completed; in such a rare case `pythia.next()` is false and the event should be skipped. Thus the user need only be on the lookout for aborts. During event generation, a problem message is printed only the first time it occurs (except for a few special cases). The above-mentioned `pythia.stat()` will then tell you how many times each problem was encountered over the entire run.

- Studying the event listing for a few events at the beginning of each run is useful to make sure you are generating the right kind of events, at the right energies, etc. For real analyses, however, you need automated access to the event record. The Pythia event record provides many utilities to need automated access to the event record. The Pythia event record provides many utilities to make this as simple and efficient as possible. To access all the particles in the event record, use the following loop after calling `pythia.next()` (typically within the event loop).

```
for prt in pythia.event:
    print(prt.id())
```

```
90
2212
2212
21
21
6
-6
21
21
6
-6
21
21
21
6
-6
21
21
21
21
21
21
21
21
21
21
21
21
21
21
21
21
21
21
21
21
21
21
21
21
1
2
1
2
1
2
1
2
21
21
```

```
21
21
6
-6
21
21
21
21
1
```

> This is called the `particle loop`. Inside this loop, you can access the properties of each particle `prt`. For instance, the method `id()` returns the PDG identity code of a particle (see [Identity codes]). The `print` statement, therefore, will give a list of the PDG code of every particle in the event record. Note, it is also possible to access particles via index, e.g.
>
> ```
> for i in range(0, pythia.event.size()):
>     print(pythia.event[i])
> ```
>
> The entries of the `Event` class in Pythia are particles of type `Particle`. The `Particle` class within Pythia provides information on the particle, including fundemental properties as well as kinematics. The [Particle Properties page] of the HTML manual gives full details on all the available information. Below, you can see the available methods using `help`.

```
# Print documentation for the Particle class in Pythia.
help(pythia8.Particle)
```

```
Help on class Particle in module pythia8mc:

class Particle(pybind11_builtins.pybind11_object)
 |  Method resolution order:
 |      Particle
 |      pybind11_builtins.pybind11_object
 |      builtins.object
 |
 |  Methods defined here:
 |
 |  __init__(...)
 |      __init__(*args, **kwargs)
 |      Overloaded function.
 |
 |      1. __init__(self: pythia8mc.Particle) -> None
 |
 |      2. __init__(self: pythia8mc.Particle, arg0: int) -> None
 |
 |      doc
 |
 |      3. __init__(self: pythia8mc.Particle, arg0: int, arg1: int) -> None
 |
```

```
|       doc
|
|       4. __init__(self: pythia8mc.Particle, arg0: int, arg1: int, arg2: int) -> N
|
|       doc
|
|       5. __init__(self: pythia8mc.Particle, arg0: int, arg1: int, arg2: int, arg3
|
|       doc
|
|       6. __init__(self: pythia8mc.Particle, arg0: int, arg1: int, arg2: int, arg3
|
|       doc
|
|       7. __init__(self: pythia8mc.Particle, arg0: int, arg1: int, arg2: int, arg3
|
|       doc
|
|       8. __init__(self: pythia8mc.Particle, arg0: int, arg1: int, arg2: int, arg3
|
|       doc
|
|       9. __init__(self: pythia8mc.Particle, arg0: int, arg1: int, arg2: int, arg3
|
|       doc
|
|       10. __init__(self: pythia8mc.Particle, arg0: int, arg1: int, arg2: int, arg
|
|       doc
|
|       11. __init__(self: pythia8mc.Particle, arg0: int, arg1: int, arg2: int, arg
|
|       doc
|
|       12. __init__(self: pythia8mc.Particle, arg0: int, arg1: int, arg2: int, arg
|
```

- As mentioned above, the event listing contains all partons and particles, traced through a number of intermediate steps. Eventually, the top will decay ($\mathrm{t} \to \mathrm{Wb}$), and by implication it is the last top copy in the event record that defines the definitive top production kinematics, just before the decay. You can obtain the location of this final top e.g. by inserting a line just before the particle loop

  ```
  top = None
  ```

  and a line inside the particle loop

  ```
  if prt.id() == 6: top = prt
  ```

  The value of `top` will be set every time a top is found in the event record. When the particle loop is complete, `top` will now point to the final top in the event record. The index of this particle in the event record can be retrieved by calling `top.index()`. Executing the cell below will find the

the event record can be retrieved by calling `top.index()`. Executing the cell below will find the final t of the last generated event.

```
top = None
for prt in pythia.event:
    if prt.id() == 6:
        top = prt
```

- In addition to the particle properties in the event listing, there are also methods that return many derived quantities for a particle, such as transverse momentum, `top.pT()`, and pseudorapidity, `top.eta()`. Use these methods to print out the values for the final top found above.

```
top.pT(), top.eta()
```

```
(8.790018419245058, 1.2715189035643584)
```

- We now want to generate more events, say 1000, to view the shape of these distributions. Inside Pythia is a very simple histogramming class, see Histograms, that can be used for rapid check/debug purposes. To book a histogram (typically before the event loop), use the following, where the last three arguments are the number of bins, the lower edge and the upper edge of the histogram, respectively.

```
pT = pythia8.Hist("top transverse momentum", 100, 0.0, 200.0)
eta = pythia8.Hist("top pseudorapidity", 100, -5.0, 5.0)
```

> Now we want to fill the histograms in each event, which can be done as follows.

```
pT.fill(top.pT())
eta.fill(top.eta())
```

> Finally, to write out the histograms, after the event loop we can write the following, typically at the end of the event loop.

```
print(pT)
```

```
 2025-07-03 17:36       top transverse momentum

        1.00*10^ 0      X
        0.96*10^ 0      X
        0.92*10^ 0      X
        0.88*10^ 0      X
        0.84*10^ 0      X
```

```
              0.84*10^ 0      X
              0.80*10^ 0      X
              0.76*10^ 0      X
              0.72*10^ 0      X
              0.68*10^ 0      X
              0.64*10^ 0      X
              0.60*10^ 0      X
              0.56*10^ 0      X
              0.52*10^ 0      X
              0.48*10^ 0      X
              0.44*10^ 0      X
              0.40*10^ 0      X
              0.36*10^ 0      X
              0.32*10^ 0      X
              0.28*10^ 0      X
              0.24*10^ 0      X
              0.20*10^ 0      X
              0.16*10^ 0      X
              0.12*10^ 0      X
              0.08*10^ 0      X
              0.04*10^ 0      X

                Contents
                  *10^ 0  000010000000000000000000000000000000000000000000000000000000000
                  *10^-1  000000000000000000000000000000000000000000000000000000000000000
                  *10^-2  000000000000000000000000000000000000000000000000000000000000000
                  *10^-3  000000000000000000000000000000000000000000000000000000000000000

                Low edge
                  *10^ 2  000000000000000000000000000000000000000000000000001111111111111
                  *10^ 1  000001111122222333334444455555666667777788888999990000011111222
                  *10^ 0  024680246802468024680246802468024680246802468024680246802468024

       Entries  =          1   xMin =     0.000   Underflow =    0.0000   Mean   =     9
       SumW(in) =    1.0000   xMax =   200.000   Overflow  =    0.0000   Median =     9
```

> In Python, it is also simple to plot this histogram with `Matplotlib`. First, we need to create our `x` and `y` vectors to plot from the histogram.

```
x = [(x0 + x1) / 2 for x0, x1 in zip(pT.getBinEdges(), pT.getBinEdges()[1:])]
y = pT.getBinContents()
```

> Next, we can use `Matplotlib` as normal.

```
%matplotlib inline
from matplotlib import pyplot

pyplot.plot(x, y)
pyplot.show()
```

> Do you understand why the η distribution looks the way it does? Propose and study a related but alternative measure and compare.

- As a final standalone exercise, consider plotting the charged multiplicity of events. You then need to have a counter set to zero for each new event. Inside the particle loop this counter should be incremented whenever the particle `isCharged()` and `isFinal()`. For the histogram, note that it can be treacherous to have bin limits at integers, where roundoff errors decide whichever way they go. In this particular case only even numbers are possible, so 100 bins from -1 to 399 would still be acceptable.

- Now, to make it easier to examine the event record and discuss the physics content, please execute the following command.

```
commands = """import pythia8mc as pythia8; pythia = pythia8.Pythia(); pythia.readString

import os

os.system(f"python -c '{commands}' > ./evt_record.txt")
```

```
0
```

- Please navigate to the "Home Page" tab in your browser and open the file "evt_record.txt" in a different window.

Input files

When running structure developed above via `C++` code, it is necessary to recompile the main program for each minor change, e.g. if you want to rerun with more statistics. This is not time-consuming for a simple standalone run, but may become so for more realistic applications. Therefore, parameters can be put in special input "card" files that are read by the main program. For Python analyses, like the examples here, compilation is no longer an issue, but it can be cleaner to factorize the code from the settings that are passed to Pythia. In this way, it is easy to track settings changes vs. analysis changes.

We will now create such a file, with the same settings used in the example program above.

```
with open("mymain01.cmnd", "w") as file:
    file.write(
        """
# t tbar production at the LHC
```

```
Beams:idA = 2212          # first incoming beam is a 2212, i.e. a proton.
Beams:idB = 2212          # second beam is also a proton.
Beams:eCM = 8000.         # the cm energy of collisions.
Top:gg2ttbar = on         # switch on the process g g -> t tbar.
Top:qqbar2ttbar = on      # switch on the process q qbar -> t tbar.
Main:numberOfEvents = 100 # set the number of events to generate.
"""
    )
```

The `mymain01.cmnd` file can contain one command per line, of the type

```
variable = value
```

All variable names are case-insensitive (the mixing of cases has been chosen purely to improve readability) and non-alphanumeric characters (such as !, # or $) will be interpreted as the start of a comment. All valid variables are listed in the HTML manual. Cut-and-paste of variable names can be used to avoid spelling mistakes.

The final step is to modify our program to use this input file. The name of this input file can be hardcoded in the main program, but for more flexibility, it can also be provided as a command-line argument when running a Python file as a script. To do this in Python, we recommend using `argparse`.

```
# Set up the arguments.
import argparse

parser = argparse.ArgumentParser(description="Description of what the program does.")
parser.add_argument(
    "-c",
    "--cmnd",
    help="Command file.",
    required=True,
    type=str,
    default="mymain01.cmnd",
)
args = vars(parser.parse_args())

# Read in the command file.
pythia.readFile(args.cmnd)
```

Here, we can just create a Pythia instance, read in the file, and initialize.

```
import pythia8mc as pythia8

pythia = pythia8.Pythia()
pythia.readFile("mymain01.cmnd")
pythia.init()
```

```
 *-------------------------------------------------------------------------------
 |
 |  *----------------------------------------------------------------------------
 |  |
 |  |
 |  |   PPP   Y   Y  TTTTT  H   H  III    A      Welcome to the Lund Monte Carlo!
 |  |   P  P   Y Y     T    H   H   I    A A     This is PYTHIA version 8.315
 |  |   PPP     Y      T    HHHHH   I   AAAAA    Last date of change: 27 May 2025
 |  |   P       Y      T    H   H   I   A   A
 |  |   P       Y      T    H   H  III  A   A    Now is 03 Jul 2025 at 17:36:18
 |  |
 |  |   Program documentation and an archive of historic versions is found on:
 |  |
 |  |                               https://pythia.org/
 |  |
 |  |   PYTHIA is authored by a collaboration consisting of:
 |  |
 |  |   Javira Altmann, Christian Bierlich, Naomi Cooke, Nishita Desai,
 |  |   Ilkka Helenius, Philip Ilten, Leif Lonnblad, Stephen Mrenna,
 |  |   Christian Preuss, Torbjorn Sjostrand, and Peter Skands.
 |  |
 |  |   The complete list of authors, including contact information and
 |  |   affiliations, can be found on https://pythia.org/.
 |  |   Problems or bugs should be reported on email at authors@pythia.org.
 |  |
 |  |   The main program reference is C. Bierlich et al,
 |  |   'A comprehensive guide to the physics and usage of Pythia 8.3',
 |  |   SciPost Phys. Codebases 8-r8.3 (2022) [arXiv:2203.11601 [hep-ph]]
 |  |
 |  |   PYTHIA is released under the GNU General Public Licence version 2
 |  |   or later. Please respect the MCnet Guidelines for Generator Authors
 |  |   and Users.
 |  |
 |  |   Disclaimer: this program comes without any guarantees.
 |  |   Beware of errors and use common sense when interpreting results.
 |  |
 |  |   Copyright (C) 2025 Torbjorn Sjostrand
 |  |
 |  |
 |  *----------------------------------------------------------------------------
 |
 *-------------------------------------------------------------------------------


 *-------  PYTHIA Process Initialization  --------------------------*
 |                                                                  |
 | We collide p+ with p+ at a CM energy of 8.000e+03 GeV            |
 |                                                                  |
 |------------------------------------------------------------------|
 |                                                    |             |
```

```
| Subprocess                                              Code |   Estimated |
|                                                              |    max (mb) |
|                                                              |             |
|------------------------------------------------------------------------------|
|                                                              |             |
| g g -> t tbar                                            601 |   2.241e-06 |
| q qbar -> t tbar                                         602 |   4.599e-07 |
```

In addition to all the internal `Pythia` variables there exist a few defined in the database but not actually used. These are intended to be useful in the main program, and thus begin with `Main:`. The most basic of those is `Main:numberOfEvents`, which you can use to specify how many events you want to generate. To make this have any effect, you need to read it in the main program, after the `pythia.readFile(...)` command, by a line like the following.

```
nEvent = pythia.mode("Main:numberOfEvents")
for iEvent in range(nEvent):
    pythia.next()
```

```
 --------  PYTHIA Info Listing  ----------------------------------------

 Beam A: id =   2212, pz =  4.000e+03, e =  4.000e+03, m =  9.383e-01.
 Beam B: id =   2212, pz = -4.000e+03, e =  4.000e+03, m =  9.383e-01.

 In 1: id =   21, x =  1.660e-02, pdf =  5.415e+00 at Q2 =  3.087e+04.
 In 2: id =   21, x =  1.299e-01, pdf =  5.694e-01 at same Q2.

 Subprocess g g -> t tbar with code 601 is 2 -> 2.
 It has sHat =  1.380e+05,    tHat = -6.132e+04,    uHat = -1.666e+04,
       pTHat =  2.985e+01,   m3Hat =  1.732e+02,   m4Hat =  1.731e+02,
    thetaHat =  2.681e+00,  phiHat =  2.829e+00.
     alphaEM =  7.876e-03,  alphaS =  1.178e-01    at Q2 =  3.087e+04.

 Impact parameter b =  4.236e-01 gives enhancement factor =  2.758e+00.
 Max pT scale for MPI =  8.000e+03, ISR =  8.000e+03, FSR =  8.000e+03.
 Number of MPI =     7, ISR =    18, FSRproc =    59, FSRreson =    18.

 --------  End PYTHIA Info Listing  ------------------------------------

 --------  PYTHIA Event Listing  (hard process)  -----------------------------------

    no         id  name            status     mothers   daughters     colours
     0         90  (system)           -11     0     0     0     0     0     0
     1       2212  (p+)               -12     0     0     3     0     0     0
     2       2212  (p+)               -12     0     0     4     0     0     0
     3         21  (g)                -21     1     0     5     6   101   102
     4         21  (g)                -21     2     0     5     6   103   101
     5          6  (t)                -22     3     4     7     8   103     0    -2
     6         -6  (tbar)             -22     3     4     9    10     0   102     2
     7         24  (W+)               -22     5     0    11    12     0     0    -8
     8          5  b                   23     5     0     0     0   103     0     5
     9        -24  (W-)               -22     6     0    13    14     0     0     7
    10         -5  bbar                23     6     0     0     0     0   102    -4
    11         -1  dbar                23     7     0     0     0     0   104    -8
```

```
   12          2  u                   23     7     0     0     0   104     0     -
   13         15  tau-                23     9     0     0     0     0     0     2
   14        -16  nu_taubar           23     9     0     0     0     0     0     4
                                  Charge sum:  0.000          Momentum sum:     -

 --------  End PYTHIA Event Listing  -----------------------------------------

 --------  PYTHIA Event Listing  (complete event)  ---------------------------

   no         id  name            status     mothers   daughters     colours
    0         90  (system)           -11     0     0     0     0     0     0
    1       2212  (p+)               -12     0     0   377     0     0     0
    2       2212  (p+)               -12     0     0   378     0     0     0
    3         21  (g)                -21     7     7     5     6   101   102
    4         21  (g)                -21     8     0     5     6   103   101
    5          6  (t)                -22     3     4     9     9   103     0    -2
    6         -6  (tbar)             -22     3     4    10    10     0   102     2
    7         21  (g)                -42    12     0     3     3   101   102     -
    8         21  (g)                -41    13    13    11     4   105   101
    9          6  (t)                -44     5     5    14    14   103     0    -2
   10         -6  (tbar)             -44     6     6    15    15     0   102     2
   11         21  (g)                -43     8     0    16    16   105   103     -
```

You are now free to play with further options in the input file, such as:

- change the top mass, which by default is 171 GeV.

```
6:m0 = 175
```

- switch off final-state radiation.

```
PartonLevel:FSR = off
```

- switch off initial-state radiation.

```
PartonLevel:ISR = off
```

- switch off multiparton interactions.

```
PartonLevel:MPI = off
```

- different combined tunes, in particular to radiation and multiparton interactions parameters. In part this reflects that no generator is perfect, and also not all data is perfect, so different emphasis will result in different optima.

```
Tune:pp = 3 # (or other values between 1 and 17)
```

- all runs by default use the same random-number sequence, for reproducibility, but you can pick any number between 1 and 900,000,000 to obtain a unique sequence.

```
Random:setSeed = on
Random:seed = 123456789
```

For instance, check the importance of FSR, ISR and MPI on the charged multiplicity of events by switching off one component at a time. The possibility to use command-line input files is further illustrated e.g. in `main132.cc` and `main231.cc`.

You have now completed the core part of the worksheet - congratulations! From now on you should be able to take off in different directions, depending on your interests. The following three sections contain examples of further possible studies, and can be addressed in any order.

CKKW-L merging

The main programs we have constructed and studied in the previous sections have one common drawback: all start from the Pythia 8 internal library of lowest-order processes, and then add higher-order corrections entirely by the internal parton-shower machinery. This will give reliable results for soft and collinear configurations, but less so for multiple hard, well-separated jets. To model the latter similarly well, we need to include external input from higher-order calculations at tree level and, where feasible, at one-loop level. A number of different external programs can provide such input, using the LHA/LHEF standard format to transfer information, usually as LHE files (see Generic User Process Interface for Event Generators, A standard format for Les Houches Event Files, and Les Houches 2013: Physics at TeV Colliders: Standard Model Working Group Report). The hard-process events stored in these files will be accepted or rejected in such a way that doublecounting between different parton multiplicities is removed, resulting in a smooth transition between the multiplicities, and between the external input and the internal handling of parton showers. These two tasks usually go hand in hand.

Many different schemes have been proposed for matrix element + parton shower merging (MEPS), and a comprehensive selection of such schemes is available with the Pythia 8 distribution, including

- tree-level merging: *MLM* jet matching (see Matching matrix elements and shower evolution for top-quark production in hadronic collisions, *MadGraph*- or *AlpGen*-style), *CKKW-L* merging (see Matching Tree-Level Matrix Elements with Interleaved Showers), and unitarised *ME*+*PS* merging (*UMEPS*, see Unitarising Matrix Element + Parton Shower merging); and
- next-to-leading order merging: NL^3 merging and unitarised NLO+PS merging (*UNLOPS*, see Merging Multi-leg NLO Matrix Elements with Parton Showers).

The setup of such merging schemes is documented in the Matching and Merging page of the HTML manual manual with further subpages, and is illustrated in several of the example main programs.

Here we will experiment with the *CKKW-L* scheme, which was the first merging scheme available in

Pythia 8, and also is among the simpler to work with. We will take the `main162` example main program as a starting point for our studies, reproduced inline below. In its general structure it closely resembles the main program(s) we already constructed step by step, so we will only need to comment on aspects that are new for the merging game. The process $\mathrm{W}^{+} + \leq 2$ jets will be taken as an example. It uses the LHE files:

- `w+_production_lhc_0.lhe` for $\mathrm{W}^{+} + 0$ partons,
- `w+_production_lhc_1.lhe` for $\mathrm{W}^{+} + 1$ parton,
- `w+_production_lhc_2.lhe` for $\mathrm{W}^{+} + 2$ partons, in the `examples` directory to produce a result that simultaneously describes $\mathrm{W}^{+} + 0, 1, 2$ jet observables with leading-order matrix elements, while also including arbitrarily many shower emissions. Jets are here defined by a clustering procedure on the partons thus generated. (We omit other effects from consideration, such as MPIs or hadronization.)

```
# First, we need to download Pythia to access the examples.
!git clone --depth 1 https://gitlab.com/Pythia8/releases.git

# Next, we need to change to the examples directory.
%cd releases/examples
```

```
Cloning into 'releases'...
remote: Enumerating objects: 1470, done.
remote: Counting objects: 100% (1470/1470), done.
remote: Compressing objects: 100% (1177/1177), done.
remote: Total 1470 (delta 359), reused 685 (delta 285), pack-reused 0 (from 0)
Receiving objects: 100% (1470/1470), 28.75 MiB | 20.63 MiB/s, done.
Resolving deltas: 100% (359/359), done.
/content/releases/examples
```

```
# Create a little plotting tool for the Pythia histogram class.
%matplotlib inline
from matplotlib import pyplot


def plot(hist, xlabel=None, ylabel="entries"):
    x = [(x0 + x1) / 2 for x0, x1 in zip(hist.getBinEdges(), hist.getBinEdges()[1:])]
    y = hist.getBinContents()
    pyplot.plot(x, y)
    pyplot.xlabel(xlabel)
    pyplot.ylabel(ylabel)
    pyplot.show()


# Initialize the Pythia object.
import pythia8mc as pythia8

# Generator. Input parameters.
pythia = pythia8.Pythia()
pythia.readFile("main162ckkwl.cmnd")
```

```
pythia.readString("Merging:includeWeightInXsection = off")

# Extract number of events and max number of jets in merging.
nEvent = pythia.mode("Main:numberOfEvents")
nMerge = pythia.mode("Merging:nJetMax")

# Histograms combined over all jet multiplicities.
pTWsum = pythia8.Hist("pT of W, summed over all subruns", 100, 0.0, 200.0)

# Merged total cross section, summed over subruns.
sigmaTotal = 0.0
```

```
 *-------------------------------------------------------------------------------
 |
 |  *----------------------------------------------------------------------------
 |  |
 |  |
 |  |   PPP   Y   Y  TTTTT  H   H  III    A      Welcome to the Lund Monte Carlo!
 |  |   P  P   Y Y     T    H   H   I    A A     This is PYTHIA version 8.315
 |  |   PPP     Y      T    HHHHH   I   AAAAA    Last date of change: 27 May 2025
 |  |   P       Y      T    H   H   I   A   A
 |  |   P       Y      T    H   H  III  A   A    Now is 03 Jul 2025 at 17:36:23
 |  |
 |  |   Program documentation and an archive of historic versions is found on:
 |  |
 |  |                               https://pythia.org/
 |  |
 |  |   PYTHIA is authored by a collaboration consisting of:
 |  |
 |  |   Javira Altmann, Christian Bierlich, Naomi Cooke, Nishita Desai,
 |  |   Ilkka Helenius, Philip Ilten, Leif Lonnblad, Stephen Mrenna,
 |  |   Christian Preuss, Torbjorn Sjostrand, and Peter Skands.
 |  |
 |  |   The complete list of authors, including contact information and
 |  |   affiliations, can be found on https://pythia.org/.
 |  |   Problems or bugs should be reported on email at authors@pythia.org.
 |  |
 |  |   The main program reference is C. Bierlich et al,
 |  |   'A comprehensive guide to the physics and usage of Pythia 8.3',
 |  |   SciPost Phys. Codebases 8-r8.3 (2022) [arXiv:2203.11601 [hep-ph]]
 |  |
 |  |   PYTHIA is released under the GNU General Public Licence version 2
 |  |   or later. Please respect the MCnet Guidelines for Generator Authors
 |  |   and Users.
 |  |
 |  |   Disclaimer: this program comes without any guarantees.
 |  |   Beware of errors and use common sense when interpreting results.
 |  |
 |  |   Copyright (C) 2025 Torbjorn Sjostrand
 |  |
 |  |
 |  *----------------------------------------------------------------------------
 |
 *-------------------------------------------------------------------------------
```

```
# Loop over subruns with varying number of jets.
for iMerge in range(1, nMerge + 1):
    sigmaSample = 0.0

    # Read in name of LHE file for current subrun and initialize.
    pythia.readFile("main162ckkwl.cmnd", iMerge)
    pythia.init()

    # Histograms for current jet multiplicity.
    weightNow = pythia8.Hist("event weights, current subrun", 100, 0.0, 2.5)
    pTWnow = pythia8.Hist("pT of W, current subrun", 100, 0.0, 200.0)

    # Start event generation loop.
    for iEvent in range(0, nEvent):
        # Generate next event. Break out of event loop if at end of LHE file.
        if not pythia.next():
            if pythia.infoPython().atEndOfFile():
                break
            else:
                continue

        # Get CKKWL weight of current event. Histogram and accumulate it.
        evtweight = pythia.infoPython().weight()
        weight = pythia.infoPython().mergingWeight()
        weightNow.fill(weight)
        weight *= evtweight
        sigmaSample += weight

        # Do nothing for vanishing weight (event record might not be filled).
        if weight == 0:
            continue

        # Find the final copy of the W+, which is after the full shower.
        prtW = None
        for prt in pythia.event:
            if prt.id() == 24:
                prtW = prt

        # Fill the pT of the W histogram, with CKKWL weight.
        if prtW:
            pTWnow.fill(prtW.pT(), weight)

    # Normalize pTW histogram, convert mb -> pb, and correct for bin width.
    pTWnow *= (
        1e9 * pythia.infoPython().sigmaGen() / (2.0 * pythia.infoPython().nAccepted())
    )

    # Print cross section and histograms for current subrun.
    # Alternatively, use print(hist) for Pythia style histogram printing.
    pythia.stat()
    plot(weightNow, f"event weights, subrun {iMerge}")
    plot(pTWnow, f"$p_T(W^+)$, subrun {iMerge}")
```

```
# Sum up merged cross section of current run.
sigmaSample *= pythia.infoPython().sigmaGen() / float(
    pythia.infoPython().nAccepted()
)
sigmaTotal += sigmaSample

# Add current histogram to the combined one. End of subrun loop.
pTWsum += pTWnow
```

```
 *-----------------------  SusyLesHouches SUSY/BSM Interface  ---------------------
 | Last Change 12 Apr 2017 - P. Skands
 | Parsing: w_production_tree_2.lhe
 | (SLHA::readFile) line 159 - storing non-SLHA(2) block: yukawa
 *--------------------------------------------------------------------------------
 PYTHIA Info from SLHAinterface::initSLHA: No MODSEL found, keeping internal SUSY s
 PYTHIA Warning in SLHAinterface::initSLHA: ignoring MASS entries for id = {1,2,3,4
 PYTHIA Warning in SLHAinterface::initSLHA: ignoring DECAY tables for id = {23,24,1
 PYTHIA Warning in SLHAinterface::initSLHA: ignoring empty DECAY tables for id = {6

 *------------------ MEPS Merging Initialization  ------------------*
 |                                                                   |
 | CKKW-L merge                                                      |
 |             pp>LEPTONS,NEUTRINOS  with up to  2 additional jets |
 | Merging scale is defined by Lund pT, with value tMS =   15.0 GeV |
 |                                                                   |
 *-------------- END MEPS Merging Initialization  ------------------*


 *-------  PYTHIA Process Initialization  --------------------------*
 |                                                                  |
 | We collide p+ with p+ at a CM energy of 7.000e+03 GeV            |
 |                                                                  |
 |------------------------------------------------------------------|
 |                                                    |             |
 | Subprocess                                    Code |   Estimated |
 |                                                    |    max (mb) |
 |                                                    |             |
 |------------------------------------------------------------------|
 |                                                    |             |
 | Les Houches User Process(es)                  9999 |   9.085e-07 |
 |                                                                  |
 *-------  End PYTHIA Process Initialization ----------------------*

 *-------  PYTHIA Multiparton Interactions Initialization  ---------*
 |                                                                  |
 |                   sigmaNonDiffractive =    50.58 mb              |
 |                                                                  |
 |    pT0 =  2.28 gives sigmaInteraction =   243.78 mb: accepted    |
 |                                                                  |
 *-------  End PYTHIA Multiparton Interactions Initialization  -----*

 *-------  PYTHIA Multiparton Interactions Initialization  ---------*
 |                                                                  |
 |                   sigmaNonDiffractive =    50.58 mb              |
 |                                                                  |
 |    pT0 =  2.28 gives sigmaInteraction =   242.07 mb: accepted    |
```

```
 |                                                                  |
 *-------  End PYTHIA Multiparton Interactions Initialization  -----*
 PYTHIA Warning in MultipartonInteractions::init: maximum increased by factor 1.075

 *-------  PYTHIA Flag + Mode + Parm + Word + FVec + MVec + PVec + WVec Settings (c
 |
 | Name                                              |                      Now |
 |                                                   |                          |
 | Beams:frameType                                   |                        4 |
 | Beams:LHEF                                        |  w_production_tree_2.lhe |
 | Main:numberOfSubruns                              |                        3 |
 | Main:subrun                                       |                        1 |
 | Merging:doPTLundMerging                           |                       on |
 | Merging:nJetMax                                   |                        2 |
 | Merging:Process                                   |    pp>LEPTONS,NEUTRINOS |
 | Merging:TMS                                       |                15.00000 |
 | SpaceShower:rapidityOrder                         |                      off |
 | StringZ:useOldAExtra                              |                       on |
 | UncertaintyBands:doVariations                     |                       on |
 | UncertaintyBands:List                             |                          | alpha
 |                                                   |                          |   alp
 |                                                   |                          |   har
 |                                                   |                          |  hard
 |
 *-------  End PYTHIA Flag + Mode + Parm + Word + FVec + MVec + PVec + WVec Setting

 --------  PYTHIA Particle Data Table (changed only)  ----------------------------

      id   name            antiName         spn chg col      m0        mWidth
             no onMode   bRatio   meMode     products

       6  t                tbar               2   2   1  173.00000    1.35678  165.
            0     1   0.0000664    0       24        1
            1     1   0.0017361    0       24        3
            2     1   0.9981975    0       24        5
            3     0   0.0000000    0       37        5

      25  h0                                  1   0   0  125.00000    0.00408  124.
            0     1   0.0000009    0        1       -1
            1     1   0.0000002    0        2       -2
            2     1   0.0002163    0        3       -3
            3     1   0.0288493    0        4       -4
            4     1   0.5768873    0        5       -5
            5     1   0.0000000    0        6       -6
            6     1   0.0000000    0       11      -11
            7     1   0.0002185    0       13      -13
            8     1   0.0628569    0       15      -15
            9     1   0.0850383    0       21       21
           10     1   0.0022965    0       22       22
           11     1   0.0016155    0       22       23
           12     1   0.0261758    0       23       23
           13     1   0.2158445    0       24      -24
           14     1   0.0000000  103  1000022  1000022
           15     1   0.0000000  103  1000023  1000022
           16     1   0.0000000  103  1000023  1000023
           17     1   0.0000000  103  1000025  1000022
           18     1   0.0000000  103  1000025  1000023
           19     1   0.0000000  103  1000025  1000025
```

20	1	0.0000000	103	1000035	1000022
21	1	0.0000000	103	1000035	1000023
22	1	0.0000000	103	1000035	1000025
23	1	0.0000000	103	1000035	1000035
24	1	0.0000000	103	1000024	-1000024
25	1	0.0000000	103	1000024	-1000037
26	1	0.0000000	103	1000037	-1000024
27	1	0.0000000	103	1000037	-1000037
28	1	0.0000000	103	1000001	-1000001
29	1	0.0000000	103	2000001	-2000001
30	1	0.0000000	103	1000001	-2000001
31	1	0.0000000	103	-1000001	2000001
32	1	0.0000000	103	1000002	-1000002
33	1	0.0000000	103	2000002	-2000002
34	1	0.0000000	103	1000002	-2000002
35	1	0.0000000	103	-1000002	2000002
36	1	0.0000000	103	1000003	-1000003
37	1	0.0000000	103	2000003	-2000003
38	1	0.0000000	103	1000003	-2000003
39	1	0.0000000	103	-1000003	2000003
40	1	0.0000000	103	1000004	-1000004
41	1	0.0000000	103	2000004	-2000004
42	1	0.0000000	103	1000004	-2000004
43	1	0.0000000	103	-1000004	2000004
44	1	0.0000000	103	1000005	-1000005
45	1	0.0000000	103	2000005	-2000005
46	1	0.0000000	103	1000005	-2000005
47	1	0.0000000	103	-1000005	2000005
48	1	0.0000000	103	1000006	-1000006
49	1	0.0000000	103	2000006	-2000006
50	1	0.0000000	103	1000006	-2000006
51	1	0.0000000	103	-1000006	2000006
52	1	0.0000000	103	1000011	-1000011
53	1	0.0000000	103	2000011	-2000011
54	1	0.0000000	103	1000011	-2000011
55	1	0.0000000	103	-1000011	2000011
56	1	0.0000000	103	1000012	-1000012
57	1	0.0000000	103	2000012	-2000012
58	1	0.0000000	103	1000012	-2000012
59	1	0.0000000	103	-1000012	2000012
60	1	0.0000000	103	1000013	-1000013
61	1	0.0000000	103	2000013	-2000013
62	1	0.0000000	103	1000013	-2000013
63	1	0.0000000	103	-1000013	2000013
64	1	0.0000000	103	1000014	-1000014
65	1	0.0000000	103	2000014	-2000014
66	1	0.0000000	103	1000014	-2000014
67	1	0.0000000	103	-1000014	2000014
68	1	0.0000000	103	1000015	-1000015
69	1	0.0000000	103	2000015	-2000015
70	1	0.0000000	103	1000015	-2000015
71	1	0.0000000	103	-1000015	2000015
72	1	0.0000000	103	1000016	-1000016
73	1	0.0000000	103	2000016	-2000016
74	1	0.0000000	103	1000016	-2000016
75	1	0.0000000	103	-1000016	2000016

-------- End PYTHIA Particle Data Table --

```
 --------  LHA initialization information  ------------

  beam    kind      energy  pdfgrp  pdfset
     A    2212    3500.000       0   10042
     B    2212    3500.000       0   10042

  Event weighting strategy =  3

  Processes, with strategy-dependent cross section info
  number      xsec (pb)      xerr (pb)      xmax (pb)
       1     5.6146e+02     3.5741e+00     9.1728e-01
       2     3.4701e+02     2.2090e+00     9.1728e-01

 --------  End LHA initialization information  --------

 --------  LHA event information and listing  -------------------------------------

    process =        2    weight =   9.1728e-01     scale =   8.0400e+01 (GeV)
                        alpha_em =   7.9577e-02    alpha_strong =   1.2978e-01

    Participating Particles
    no        id stat     mothers     colours      p_x        p_y        p_z
     1        21   -1     0     0   501   502      0.000      0.000    188.882    1
     2         1   -1     0     0   503     0      0.000      0.000   -170.899    1
     3       -24    2     1     2     0     0    -85.479     24.325    -47.618    1
     4        11    1     3     3     0     0    -43.339     44.847    -54.895
     5       -12    1     3     3     0     0    -42.140    -20.523      7.277
     6        21    1     1     2   503   502     23.974     75.209     78.777    1
     7         2    1     1     2   501     0     61.505    -99.534    -13.176    1

 --------  End LHA event information and listing  ---------------------------------

 --------  PYTHIA Info Listing  ----------------------------------------

 Beam A: id =   2212, pz =  3.500e+03, e =  3.500e+03, m =  9.383e-01.
 Beam B: id =   2212, pz = -3.500e+03, e =  3.500e+03, m =  9.383e-01.

 In 1: id =   21, x =  5.397e-02, pdf =  0.000e+00 at Q2 =  6.464e+03.
 In 2: id =    1, x =  4.883e-02, pdf =  0.000e+00 at same Q2.

 Process Les Houches User Process(es) with code 9999 is 2 -> 3.
 Subprocess user process 2 with code 2 is 2 -> 3.
 It has sHat =  1.291e+05, <pTHat> =  9.494e+01.
     alphaEM =  7.958e-02,  alphaS =  1.298e-01    at Q2 =  6.464e+03.

 Impact parameter b =  9.622e-01 gives enhancement factor =  7.255e-01.
 Max pT scale for MPI =  8.612e+01, ISR =  8.040e+01, FSR =  8.040e+01.
 Number of MPI =     5, ISR =    10, FSRproc =    36, FSRreson =     0.

 --------  End PYTHIA Info Listing  ------------------------------------

 --------  PYTHIA Event Listing  (hard process)  -----------------------------------

    no         id  name            status     mothers   daughters     colours
     0         90  (system)           -11     0     0     0     0     0     0
     1       2212  (p+)               -12     0     0     3     0     0     0
     2       2212  (p+)               -12     0     0     4     0     0     0
```

```
    2       2212  (p+)              -12     0     0     4     0     0     0
    3         21  (g)               -21     1     0     5     7   501   502
    4          1  (d)               -21     2     0     5     7   503     0
    5        -24  (W-)              -22     3     4     8     9     0     0    -8
    6         21  g                  23     3     4     0     0   503   502     2
    7          2  u                  23     3     4     0     0   501     0     6
    8         11  e-                 23     5     0     0     0     0     0    -4
    9        -12  nu_ebar            23     5     0     0     0     0     0    -4
                                 Charge sum: -0.333           Momentum sum:

 --------  End PYTHIA Event Listing  -----------------------------------------------

 --------  PYTHIA Event Listing  (complete event)  ---------------------------------

   no         id  name            status     mothers   daughters     colours
    0         90  (system)          -11     0     0     0     0     0     0
    1       2212  (p+)              -12     0     0   315     0     0     0
    2       2212  (p+)              -12     0     0   316     0     0     0
    3         21  (g)               -21     8     0     5     7   501   502
    4          1  (d)               -21     9     9     5     7   503     0
    5        -24  (W-)              -22     3     4    10    10     0     0    -8
    6         21  (g)               -23     3     4    11    11   503   502     2
    7          2  (u)               -23     3     4    12    12   501     0     6
    8         21  (g)               -41    14     0    13     3   513   502     -
    9          1  (d)               -42    15    15     4     4   503     0
   10        -24  (W-)              -44     5     5    16    16     0     0    -8
   11         21  (g)               -44     6     6    17    17   503   502     2
   12          2  (u)               -44     7     7    18    18   501     0     5
   13         21  (g)               -43     8     0    19    19   513   501
   14         21  (g)               -41    26    26    20     8   520   502     -
   15          1  (d)               -42    28     0     9     9   503     0     -
   16        -24  (W-)              -44    10    10    29    29     0     0    -8
   17         21  (g)               -44    11    11    30    30   503   502     2
   18          2  (u)               -44    12    12    21    22   501     0     5
   19         21  (g)               -44    13    13    23    23   513   501
   20         21  (g)               -43    14     0    24    25   520   513
   21          2  (u)               -51    18     0    31    31   522     0     5
   22         21  (g)               -51    18     0    34    34   501   522
   23         21  (g)               -52    19    19    32    32   513   501
   24         -5  (bbar)            -51    20     0    33    33     0   513
   25          5  (b)               -51    20     0    35    35   520     0     -
   26         21  (g)               -53    27    27    14    14   520   502     -
   27         21  (g)               -42    37     0    26    26   520   502     -
   28          1  (d)               -41    38    38    36    15   526     0
   29        -24  (W-)              -44    16    16    39    39     0     0    -8
   30         21  (g)               -44    17    17    40    40   503   502     2
   31          2  (u)               -44    21    21    41    41   522     0     5
   32         21  (g)               -44    23    23    42    42   513   501
   33         -5  (bbar)            -44    24    24    43    43     0   513
   34         21  (g)               -44    22    22    44    44   501   522
   35          5  (b)               -44    25    25    45    45   520     0     -
   36         21  (g)               -43    28     0    46    46   526   503
   37         21  (g)               -41    57    57    47    27   529   502
   38          1  (d)               -42    58     0    28    28   526     0
   39        -24  (W-)              -44    29    29    59    59     0     0    -8
   40         21  (g)               -44    30    30    53    53   503   502     1
   41          2  (u)               -44    31    31    48    49   522     0     5
   42         21  (g)               -44    32    32    62    62   513   501
   43         -5  (bbar)            -44    33    33    63    63     0   513
```

43	-5	(bbar)	-44	33	33	63	63	0	513	
44	21	(g)	-44	34	34	50	50	501	522	
45	5	(b)	-44	35	35	65	65	520	0	-
46	21	(g)	-44	36	36	51	52	526	503	
47	21	(g)	-43	37	0	67	67	529	520	
48	2	(u)	-51	41	0	61	61	530	0	1
49	21	(g)	-51	41	0	68	68	522	530	3
50	21	(g)	-52	44	44	64	64	501	522	
51	21	(g)	-51	46	0	66	66	526	533	
52	21	(g)	-51	46	0	54	55	533	503	-
53	21	(g)	-52	40	40	56	56	503	502	1
54	2	(u)	-51	52	0	69	69	533	0	
55	-2	(ubar)	-51	52	0	70	70	0	503	-
56	21	(g)	-52	53	53	60	60	503	502	1
57	21	(g)	-42	72	0	37	37	529	502	
58	1	(d)	-41	73	73	71	38	538	0	-
59	-24	(W-)	-44	39	39	74	74	0	0	-8
60	21	(g)	-44	56	56	75	75	503	502	1
61	2	(u)	-44	48	48	76	76	530	0	1
62	21	(g)	-44	42	42	77	77	513	501	
63	-5	(bbar)	-44	43	43	78	78	0	513	
64	21	(g)	-44	50	50	79	79	501	522	
65	5	(b)	-44	45	45	80	80	520	0	-
66	21	(g)	-44	51	51	81	81	526	533	
67	21	(g)	-44	47	47	82	82	529	520	
68	21	(g)	-44	49	49	83	83	522	530	3
69	2	(u)	-44	54	54	84	84	533	0	
70	-2	(ubar)	-44	55	55	85	85	0	503	-
71	21	(g)	-43	58	0	86	86	538	526	
72	21	(g)	-41	114	0	87	57	539	502	
73	1	(d)	-42	115	115	58	58	538	0	
74	-24	(W-)	-44	59	59	116	116	0	0	-8
75	21	(g)	-44	60	60	117	117	503	502	1
76	2	(u)	-44	61	61	118	118	530	0	1
77	21	(g)	-44	62	62	90	90	513	501	
78	-5	(bbar)	-44	63	63	120	120	0	513	
79	21	(g)	-44	64	64	88	89	501	522	
80	5	(b)	-44	65	65	122	122	520	0	-
81	21	(g)	-44	66	66	96	96	526	533	
82	21	(g)	-44	67	67	97	98	529	520	
83	21	(g)	-44	68	68	91	92	522	530	3
84	2	(u)	-44	69	69	126	126	533	0	
85	-2	(ubar)	-44	70	70	127	127	0	503	-
86	21	(g)	-44	71	71	94	95	538	526	
87	21	(g)	-43	72	0	99	99	539	529	
88	-1	(dbar)	-51	79	0	93	93	0	522	-
89	1	(d)	-51	79	0	110	110	501	0	
90	21	(g)	-52	77	77	108	109	513	501	
91	21	(g)	-51	83	0	125	125	556	530	3
92	21	(g)	-51	83	0	131	131	522	556	
93	-1	(dbar)	-52	88	88	121	121	0	522	-
94	21	(g)	-51	86	0	128	128	538	561	
95	21	(g)	-51	86	0	132	132	561	526	-
96	21	(g)	-52	81	81	123	123	526	533	
97	21	(g)	-51	82	0	111	112	566	520	
98	21	(g)	-51	82	0	113	113	529	566	
99	21	(g)	-52	87	87	129	129	539	529	
100	-1	(dbar)	-31	204	204	102	103	0	569	
101	21	(g)	-31	207	207	102	103	571	570	

101	21	(g)	-31	207	207	102	103	571	570	
102	-1	(dbar)	-33	100	101	223	224	0	570	
103	21	(g)	-33	100	101	202	203	571	569	-
104	2	(u)	-31	370	370	106	107	572	0	
105	1	(d)	-31	371	371	106	107	573	0	
106	2	(u)	-33	104	105	372	372	573	0	
107	1	(d)	-33	104	105	373	373	572	0	-
108	21	(g)	-51	90	0	119	119	513	581	
109	21	(g)	-51	90	0	134	134	581	501	-
110	1	(d)	-52	89	89	130	130	501	0	
111	21	(g)	-51	97	0	124	124	590	520	
112	21	(g)	-51	97	0	135	135	566	590	-
113	21	(g)	-52	98	98	133	133	529	566	
114	21	(g)	-41	137	0	136	72	591	502	
115	1	(d)	-42	138	138	73	73	538	0	-
116	-24	(W-)	-44	74	74	139	139	0	0	-8
117	21	(g)	-44	75	75	140	140	503	502	1
118	2	(u)	-44	76	76	141	141	530	0	1
119	21	(g)	-44	108	108	142	142	513	581	
120	-5	(bbar)	-44	78	78	143	143	0	513	
121	-1	(dbar)	-44	93	93	144	144	0	522	-
122	5	(b)	-44	80	80	145	145	520	0	-
123	21	(g)	-44	96	96	146	146	526	533	
124	21	(g)	-44	111	111	147	147	590	520	
125	21	(g)	-44	91	91	148	148	556	530	3
126	2	(u)	-44	84	84	149	149	533	0	
127	-2	(ubar)	-44	85	85	150	150	0	503	-
128	21	(g)	-44	94	94	151	151	538	561	
129	21	(g)	-44	99	99	152	152	539	529	
130	1	(d)	-44	110	110	153	153	501	0	
131	21	(g)	-44	92	92	154	154	522	556	
132	21	(g)	-44	95	95	155	155	561	526	-
133	21	(g)	-44	113	113	156	156	529	566	
134	21	(g)	-44	109	109	157	157	581	501	-
135	21	(g)	-44	112	112	158	158	566	590	-
136	21	(g)	-43	114	0	159	159	591	539	
137	21	(g)	-41	164	0	160	114	591	598	
138	1	(d)	-42	165	165	115	115	538	0	-
139	-24	(W-)	-44	116	116	166	166	0	0	-8
140	21	(g)	-44	117	117	167	167	503	502	1
141	2	(u)	-44	118	118	168	168	530	0	1
142	21	(g)	-44	119	119	169	169	513	581	
143	-5	(bbar)	-44	120	120	170	170	0	513	
144	-1	(dbar)	-44	121	121	171	171	0	522	-
145	5	(b)	-44	122	122	172	172	520	0	-
146	21	(g)	-44	123	123	173	173	526	533	
147	21	(g)	-44	124	124	174	174	590	520	
148	21	(g)	-44	125	125	175	175	556	530	3
149	2	(u)	-44	126	126	176	176	533	0	
150	-2	(ubar)	-44	127	127	177	177	0	503	-
151	21	(g)	-44	128	128	178	178	538	561	
152	21	(g)	-44	129	129	163	163	539	529	
153	1	(d)	-44	130	130	180	180	501	0	
154	21	(g)	-44	131	131	181	181	522	556	
155	21	(g)	-44	132	132	182	182	561	526	-
156	21	(g)	-44	133	133	183	183	529	566	
157	21	(g)	-44	134	134	184	184	581	501	-
158	21	(g)	-44	135	135	185	185	566	590	-
159	21	(g)	-44	136	136	161	162	591	539	

159	21	(g)	-44	136	136	161	162	591	539	
160	21	(g)	-43	137	0	187	187	502	598	
161	21	(g)	-51	159	0	186	186	591	605	
162	21	(g)	-51	159	0	188	188	605	539	-
163	21	(g)	-52	152	152	179	179	539	529	
164	21	(g)	-41	260	260	189	137	591	609	
165	1	(d)	-42	261	0	138	138	538	0	
166	-24	(W-)	-44	139	139	262	262	0	0	-8
167	21	(g)	-44	140	140	243	243	503	502	1
168	2	(u)	-44	141	141	195	195	530	0	1
169	21	(g)	-44	142	142	235	236	513	581	
170	-5	(bbar)	-44	143	143	237	237	0	513	
171	-1	(dbar)	-44	144	144	190	191	0	522	-
172	5	(b)	-44	145	145	198	198	520	0	-
173	21	(g)	-44	146	146	199	200	526	533	
174	21	(g)	-44	147	147	196	197	590	520	
175	21	(g)	-44	148	148	193	194	556	530	3
176	2	(u)	-44	149	149	201	201	533	0	
177	-2	(ubar)	-44	150	150	208	209	0	503	-
178	21	(g)	-44	151	151	274	274	538	561	
179	21	(g)	-44	163	163	247	248	539	529	
180	1	(d)	-44	153	153	276	276	501	0	
181	21	(g)	-44	154	154	192	192	522	556	
182	21	(g)	-44	155	155	226	227	561	526	-
183	21	(g)	-44	156	156	234	234	529	566	
184	21	(g)	-44	157	157	280	280	581	501	-
185	21	(g)	-44	158	158	232	233	566	590	-
186	21	(g)	-44	161	161	282	282	591	605	
187	21	(g)	-44	160	160	283	283	502	598	
188	21	(g)	-44	162	162	249	249	605	539	-
189	21	(g)	-43	164	0	285	285	598	609	-
190	-1	(dbar)	-51	171	0	238	239	0	612	-
191	21	(g)	-51	171	0	240	240	612	522	-
192	21	(g)	-52	181	181	219	219	522	556	
193	21	(g)	-51	175	0	217	218	556	613	1
194	21	(g)	-51	175	0	244	245	613	530	2
195	2	(u)	-52	168	168	246	246	530	0	1
196	21	(g)	-51	174	0	270	270	590	616	
197	21	(g)	-51	174	0	288	288	616	520	
198	5	(b)	-52	172	172	268	268	520	0	-
199	21	(g)	-51	173	0	213	213	526	623	
200	21	(g)	-51	173	0	211	212	623	533	
201	2	(u)	-52	176	176	210	210	533	0	
202	21	(g)	-51	103	0	205	206	571	624	
203	21	(g)	-51	103	0	214	215	624	569	-
204	-1	(dbar)	-53	216	216	100	100	0	569	
205	21	(g)	-51	202	0	365	365	631	624	-
206	21	(g)	-51	202	0	367	367	571	631	
207	21	(g)	-53	225	225	101	101	571	570	
208	-2	(ubar)	-51	177	0	241	242	0	503	-
209	22	(gamma)	-51	177	0	290	290	0	0	-
210	2	(u)	-52	201	201	231	231	533	0	
211	21	(g)	-51	200	0	220	221	650	533	
212	21	(g)	-51	200	0	222	222	623	650	-
213	21	(g)	-52	199	199	228	228	526	623	
214	2	(u)	-51	203	0	366	366	624	0	-
215	-2	(ubar)	-51	203	0	368	368	0	569	-
216	-1	(dbar)	-53	362	362	204	204	0	569	
217	21	(g)	-51	193	0	271	271	657	613	1

217	21	(g)	-51	193	0	271	271	657	613	1
218	21	(g)	-51	193	0	292	292	556	657	
219	21	(g)	-52	192	192	259	259	522	556	
220	21	(g)	-51	211	0	229	230	666	533	
221	21	(g)	-51	211	0	293	293	650	666	
222	21	(g)	-52	212	212	291	291	623	650	-
223	-1	(dbar)	-51	102	0	364	364	0	671	
224	21	(g)	-51	102	0	369	369	671	570	
225	21	(g)	-53	363	363	207	207	571	570	
226	21	(g)	-51	182	0	278	278	561	673	-
227	21	(g)	-51	182	0	294	294	673	526	
228	21	(g)	-52	213	213	269	269	526	623	
229	21	(g)	-51	220	0	289	289	666	678	
230	21	(g)	-51	220	0	295	295	678	533	
231	2	(u)	-52	210	210	272	272	533	0	
232	21	(g)	-51	185	0	281	281	682	590	-
233	21	(g)	-51	185	0	296	296	566	682	
234	21	(g)	-52	183	183	279	279	529	566	
235	21	(g)	-51	169	0	265	265	684	581	
236	21	(g)	-51	169	0	297	297	513	684	
237	-5	(bbar)	-52	170	170	266	266	0	513	
238	-1	(dbar)	-51	190	0	267	267	0	685	-
239	21	(g)	-51	190	0	298	298	685	612	-
240	21	(g)	-52	191	191	257	258	612	522	-
241	-2	(ubar)	-51	208	0	273	273	0	687	-
242	21	(g)	-51	208	0	256	256	687	503	-
243	21	(g)	-52	167	167	254	255	503	502	1
244	21	(g)	-51	194	0	287	287	613	693	
245	21	(g)	-51	194	0	300	300	693	530	2
246	2	(u)	-52	195	195	264	264	530	0	1
247	21	(g)	-51	179	0	275	275	698	529	
248	21	(g)	-51	179	0	301	301	539	698	
249	21	(g)	-52	188	188	284	284	605	539	-
250	-1	(dbar)	-31	374	374	252	253	0	702	
251	21	(g)	-31	375	375	252	253	704	703	
252	-1	(dbar)	-33	250	251	376	376	0	703	-
253	21	(g)	-33	250	251	377	377	704	702	
254	21	(g)	-51	243	0	263	263	705	502	1
255	21	(g)	-51	243	0	302	302	503	705	
256	21	(g)	-52	242	242	299	299	687	503	-
257	21	(g)	-51	240	0	286	286	612	719	-
258	21	(g)	-51	240	0	303	303	719	522	
259	21	(g)	-52	219	219	277	277	522	556	
260	21	(g)	-42	315	315	164	164	591	609	-
261	1	(d)	-41	316	316	304	165	729	0	
262	-24	(W-)	-44	166	166	312	313	0	0	-8
263	21	(g)	-44	254	254	318	318	705	502	1
264	2	(u)	-44	246	246	314	314	530	0	1
265	21	(g)	-44	235	235	320	320	684	581	
266	-5	(bbar)	-44	237	237	321	321	0	513	
267	-1	(dbar)	-44	238	238	322	322	0	685	-
268	5	(b)	-44	198	198	323	323	520	0	-
269	21	(g)	-44	228	228	324	324	526	623	
270	21	(g)	-44	196	196	325	325	590	616	
271	21	(g)	-44	217	217	326	326	657	613	1
272	2	(u)	-44	231	231	327	327	533	0	
273	-2	(ubar)	-44	241	241	328	328	0	687	-
274	21	(g)	-44	178	178	307	307	538	561	
275	21	(g)	-44	247	247	330	330	698	529	

275	21	(g)	-44	247	247	330	330	698	529	
276	1	(d)	-44	180	180	331	331	501	0	
277	21	(g)	-44	259	259	332	332	522	556	
278	21	(g)	-44	226	226	305	306	561	673	-
279	21	(g)	-44	234	234	334	334	529	566	
280	21	(g)	-44	184	184	335	335	581	501	-
281	21	(g)	-44	232	232	336	336	682	590	-
282	21	(g)	-44	186	186	337	337	591	605	
283	21	(g)	-44	187	187	338	338	502	598	
284	21	(g)	-44	249	249	339	339	605	539	-
285	21	(g)	-44	189	189	340	340	598	609	-
286	21	(g)	-44	257	257	341	341	612	719	-
287	21	(g)	-44	244	244	342	342	613	693	
288	21	(g)	-44	197	197	343	343	616	520	
289	21	(g)	-44	229	229	344	344	666	678	
290	22	(gamma)	-44	209	209	345	345	0	0	-
291	21	(g)	-44	222	222	346	346	623	650	-
292	21	(g)	-44	218	218	347	347	556	657	
293	21	(g)	-44	221	221	348	348	650	666	
294	21	(g)	-44	227	227	349	349	673	526	
295	21	(g)	-44	230	230	350	350	678	533	
296	21	(g)	-44	233	233	351	351	566	682	
297	21	(g)	-44	236	236	352	352	513	684	
298	21	(g)	-44	239	239	353	353	685	612	-
299	21	(g)	-44	256	256	354	354	687	503	-
300	21	(g)	-44	245	245	355	355	693	530	2
301	21	(g)	-44	248	248	356	356	539	698	
302	21	(g)	-44	255	255	357	357	503	705	
303	21	(g)	-44	258	258	358	358	719	522	
304	21	(g)	-43	261	0	359	359	729	538	-
305	21	(g)	-51	278	0	333	333	735	673	-
306	21	(g)	-51	278	0	360	360	561	735	
307	21	(g)	-52	274	274	329	329	538	561	
308	21	(g)	-31	378	378	310	311	737	736	
309	21	(g)	-31	379	379	310	311	739	738	
310	21	(g)	-33	308	309	380	380	739	736	-
311	21	(g)	-33	308	309	381	381	737	738	
312	-24	(W-)	-51	262	0	317	317	0	0	-8
313	22	(gamma)	-51	262	0	361	361	0	0	-
314	2	(u)	-52	264	264	319	319	530	0	1
315	21	(g)	-61	1	0	260	260	569	609	
316	1	(d)	-61	2	0	261	261	729	0	
317	-24	(W-)	-62	312	312	387	388	0	0	-8
318	21	(g)	-62	263	263	535	535	705	502	1
319	2	(u)	-62	314	314	396	396	530	0	1
320	21	(g)	-62	265	265	426	426	684	581	
321	-5	(bbar)	-62	266	266	428	428	0	513	
322	-1	(dbar)	-62	267	267	405	405	0	685	-
323	5	(b)	-62	268	268	490	490	520	0	-
324	21	(g)	-62	269	269	462	462	526	623	
325	21	(g)	-62	270	270	492	492	590	616	
326	21	(g)	-62	271	271	399	399	657	613	1
327	2	(u)	-62	272	272	457	457	533	0	
328	-2	(ubar)	-62	273	273	538	538	0	687	-
329	21	(g)	-62	307	307	466	466	538	561	
330	21	(g)	-62	275	275	496	496	698	529	
331	1	(d)	-62	276	276	424	424	501	0	
332	21	(g)	-62	277	277	401	401	522	556	
333	21	(g)	-62	305	305	464	464	735	673	-

333	21	(g)	-62	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
334	21	(g)	-62	279	279	495	495	529	566	
335	21	(g)	-62	280	280	425	425	581	501	-
336	21	(g)	-62	281	281	493	493	682	590	-
337	21	(g)	-62	282	282	499	499	569	605	
338	21	(g)	-62	283	283	534	534	502	598	
339	21	(g)	-62	284	284	498	498	605	539	-
340	21	(g)	-62	285	285	533	533	598	609	-
341	21	(g)	-62	286	286	403	403	612	719	-
342	21	(g)	-62	287	287	398	398	613	693	
343	21	(g)	-62	288	288	491	491	616	520	
344	21	(g)	-62	289	289	459	459	666	678	
345	22	gamma	62	290	290	0	0	0	0	-
346	21	(g)	-62	291	291	461	461	623	650	-
347	21	(g)	-62	292	292	400	400	556	657	
348	21	(g)	-62	293	293	460	460	650	666	
349	21	(g)	-62	294	294	463	463	673	526	
350	21	(g)	-62	295	295	458	458	678	533	
351	21	(g)	-62	296	296	494	494	566	682	
352	21	(g)	-62	297	297	427	427	513	684	
353	21	(g)	-62	298	298	404	404	685	612	-
354	21	(g)	-62	299	299	537	537	687	503	-
355	21	(g)	-62	300	300	397	397	693	530	2
356	21	(g)	-62	301	301	497	497	539	698	
357	21	(g)	-62	302	302	536	536	503	705	
358	21	(g)	-62	303	303	402	402	719	522	
359	21	(g)	-62	304	304	389	0	729	538	-
360	21	(g)	-62	306	306	465	465	561	735	
361	22	gamma	62	313	313	0	0	0	0	-
362	-1	(dbar)	-61	1	0	216	216	0	569	
363	21	(g)	-61	2	0	225	225	703	570	
364	-1	(dbar)	-62	223	223	584	584	0	671	
365	21	(g)	-62	205	205	439	439	631	624	-
366	2	(u)	-62	214	214	438	438	624	0	-
367	21	(g)	-62	206	206	440	440	571	631	
368	-2	(ubar)	-62	215	215	501	501	0	704	-
369	21	(g)	-62	224	224	583	583	671	739	
370	2	(u)	-61	1	0	104	104	572	0	
371	1	(d)	-61	2	0	105	105	570	0	
372	2	(u)	-62	106	106	581	581	570	0	
373	1	(d)	-62	107	107	563	563	572	0	-
374	-1	(dbar)	-61	1	0	250	250	0	572	
375	21	(g)	-61	2	0	251	251	572	703	-
376	-1	(dbar)	-62	252	252	442	442	0	703	-
377	21	(g)	-62	253	253	500	500	704	569	-
378	21	(g)	-61	1	0	308	308	738	738	-
379	21	(g)	-61	2	0	309	309	736	736	-
380	21	(g)	-62	310	310	582	582	739	570	-
381	21	(g)	-62	311	311	441	441	703	571	
382	2101	(ud_0)	-63	1	0	391	391	0	740	-
383	1	(d)	-63	1	0	390	390	740	0	-
384	1	(d)	-63	1	0	532	532	609	0	-
385	2203	(uu_1)	-63	2	0	564	564	0	572	
386	-1	(dbar)	-63	2	0	389	0	0	729	-
387	11	e-	23	317	0	0	0	0	0	-4
388	-12	nu_ebar	23	317	0	0	0	0	0	-4
389	-1	(dbar)	-73	386	359	467	467	0	538	-
390	1	(d)	-71	383	383	392	395	740	0	-
391	2101	(ud_0)	-71	382	382	392	395	0	740	-

391	2101	(ud_0)	-71	382	382	392	393	0	740	
392	-213	(rho-)	-83	390	391	613	614	0	0	
393	213	(rho+)	-83	390	391	615	616	0	0	-
394	-211	pi-	84	390	391	0	0	0	0	-
395	2212	p+	84	390	391	0	0	0	0	-
396	2	(u)	-71	319	319	406	423	530	0	1
397	21	(g)	-71	355	355	406	423	693	530	2
398	21	(g)	-71	342	342	406	423	613	693	
399	21	(g)	-71	326	326	406	423	657	613	1
400	21	(g)	-71	347	347	406	423	556	657	
401	21	(g)	-71	332	332	406	423	522	556	
402	21	(g)	-71	358	358	406	423	719	522	
403	21	(g)	-71	341	341	406	423	612	719	-
404	21	(g)	-71	353	353	406	423	685	612	-
405	-1	(dbar)	-71	322	322	406	423	0	685	-
406	211	pi+	83	396	405	0	0	0	0	
407	-213	(rho-)	-83	396	405	617	618	0	0	
408	223	(omega)	-83	396	405	720	722	0	0	
409	223	(omega)	-83	396	405	723	725	0	0	
410	221	(eta)	-83	396	405	726	727	0	0	
411	111	(pi0)	-83	396	405	728	729	0	0	
412	213	(rho+)	-83	396	405	619	620	0	0	
413	223	(omega)	-83	396	405	730	732	0	0	
414	221	(eta)	-83	396	405	733	735	0	0	
415	111	(pi0)	-84	396	405	736	737	0	0	
416	-211	pi-	84	396	405	0	0	0	0	
417	111	(pi0)	-84	396	405	738	739	0	0	-
418	213	(rho+)	-84	396	405	621	622	0	0	
419	-213	(rho-)	-84	396	405	623	624	0	0	-
420	211	pi+	84	396	405	0	0	0	0	-
421	-211	pi-	84	396	405	0	0	0	0	
422	211	pi+	84	396	405	0	0	0	0	
423	113	(rho0)	-84	396	405	625	626	0	0	-
424	1	(d)	-71	331	331	429	437	501	0	
425	21	(g)	-71	335	335	429	437	581	501	-
426	21	(g)	-71	320	320	429	437	684	581	
427	21	(g)	-71	352	352	429	437	513	684	
428	-5	(bbar)	-71	321	321	429	437	0	513	
429	311	(K0)	-83	424	428	627	627	0	0	
430	-321	K-	83	424	428	0	0	0	0	
431	113	(rho0)	-83	424	428	628	629	0	0	
432	211	pi+	83	424	428	0	0	0	0	
433	-211	pi-	83	424	428	0	0	0	0	-
434	223	(omega)	-83	424	428	740	742	0	0	-
435	2212	p+	83	424	428	0	0	0	0	
436	-2112	nbar0	84	424	428	0	0	0	0	
437	513	(B*0)	-84	424	428	743	744	0	0	
438	2	(u)	-71	366	366	443	456	624	0	-
439	21	(g)	-71	365	365	443	456	631	624	-
440	21	(g)	-71	367	367	443	456	571	631	
441	21	(g)	-71	381	381	443	456	703	571	
442	-1	(dbar)	-71	376	376	443	456	0	703	-
443	111	(pi0)	-83	438	442	745	746	0	0	-
444	213	(rho+)	-83	438	442	630	631	0	0	
445	-211	pi-	83	438	442	0	0	0	0	-
446	213	(rho+)	-83	438	442	632	633	0	0	-
447	-211	pi-	83	438	442	0	0	0	0	
448	331	(eta')	-84	438	442	747	749	0	0	
449	211	pi+	84	438	442	0	0	0	0	-

450	111	(pi0)	-84	438	442	750	751	0	0	
451	-211	pi-	84	438	442	0	0	0	0	-
452	213	(rho+)	-84	438	442	634	635	0	0	
453	-213	(rho-)	-84	438	442	636	637	0	0	
454	213	(rho+)	-84	438	442	638	639	0	0	
455	111	(pi0)	-84	438	442	752	753	0	0	-
456	113	(rho0)	-84	438	442	640	641	0	0	-
457	2	(u)	-71	327	327	468	489	533	0	
458	21	(g)	-71	350	350	468	489	678	533	
459	21	(g)	-71	344	344	468	489	666	678	
460	21	(g)	-71	348	348	468	489	650	666	
461	21	(g)	-71	346	346	468	489	623	650	-
462	21	(g)	-71	324	324	468	489	526	623	
463	21	(g)	-71	349	349	468	489	673	526	
464	21	(g)	-71	333	333	468	489	735	673	-
465	21	(g)	-71	360	360	468	489	561	735	
466	21	(g)	-71	329	329	468	489	538	561	
467	-1	(dbar)	-71	389	389	468	489	0	538	-
468	323	(K*+)	-83	457	467	642	643	0	0	
469	-321	K-	83	457	467	0	0	0	0	
470	211	pi+	83	457	467	0	0	0	0	
471	111	(pi0)	-83	457	467	754	755	0	0	
472	113	(rho0)	-83	457	467	644	645	0	0	-
473	-211	pi-	83	457	467	0	0	0	0	
474	111	(pi0)	-83	457	467	756	757	0	0	
475	211	pi+	83	457	467	0	0	0	0	
476	111	(pi0)	-83	457	467	758	759	0	0	
477	-211	pi-	83	457	467	0	0	0	0	
478	321	K+	83	457	467	0	0	0	0	-
479	-313	(K*bar0)	-83	457	467	646	647	0	0	
480	111	(pi0)	-83	457	467	760	761	0	0	-
481	223	(omega)	-83	457	467	762	764	0	0	
482	111	(pi0)	-83	457	467	765	766	0	0	
483	223	(omega)	-84	457	467	767	769	0	0	
484	111	(pi0)	-84	457	467	770	771	0	0	
485	-211	pi-	84	457	467	0	0	0	0	-
486	211	pi+	84	457	467	0	0	0	0	
487	223	(omega)	-84	457	467	772	774	0	0	
488	-211	pi-	84	457	467	0	0	0	0	-
489	211	pi+	84	457	467	0	0	0	0	-
490	5	(b)	-71	323	323	502	531	520	0	-
491	21	(g)	-71	343	343	502	531	616	520	
492	21	(g)	-71	325	325	502	531	590	616	
493	21	(g)	-71	336	336	502	531	682	590	-
494	21	(g)	-71	351	351	502	531	566	682	
495	21	(g)	-71	334	334	502	531	529	566	
496	21	(g)	-71	330	330	502	531	698	529	
497	21	(g)	-71	356	356	502	531	539	698	
498	21	(g)	-71	339	339	502	531	605	539	-
499	21	(g)	-71	337	337	502	531	569	605	
500	21	(g)	-71	377	377	502	531	704	569	-
501	-2	(ubar)	-71	368	368	502	531	0	704	-
502	-511	(Bbar0)	-83	490	501	775	777	0	0	-
503	-211	pi-	83	490	501	0	0	0	0	-
504	321	K+	83	490	501	0	0	0	0	
505	-321	K-	83	490	501	0	0	0	0	
506	111	(pi0)	-83	490	501	778	779	0	0	-
507	211	pi+	83	490	501	0	0	0	0	

[illegible]	[illegible]	[illegible]	[illegible]	490	501	0	0	0	0	
508	113	(rho0)	-83	490	501	648	649	0	0	
509	223	(omega)	-83	490	501	780	782	0	0	
510	221	(eta)	-83	490	501	783	785	0	0	
511	-213	(rho-)	-83	490	501	650	651	0	0	-
512	111	(pi0)	-83	490	501	786	787	0	0	
513	113	(rho0)	-83	490	501	652	653	0	0	
514	211	pi+	83	490	501	0	0	0	0	
515	111	(pi0)	-83	490	501	788	789	0	0	
516	113	(rho0)	-83	490	501	654	655	0	0	
517	111	(pi0)	-83	490	501	790	791	0	0	-
518	221	(eta)	-83	490	501	792	794	0	0	
519	111	(pi0)	-83	490	501	795	796	0	0	
520	221	(eta)	-84	490	501	797	799	0	0	
521	313	(K*0)	-84	490	501	656	657	0	0	-
522	-311	(Kbar0)	-84	490	501	658	658	0	0	
523	-213	(rho-)	-84	490	501	659	660	0	0	-
524	211	pi+	84	490	501	0	0	0	0	
525	2112	n0	84	490	501	0	0	0	0	-
526	111	(pi0)	-84	490	501	800	801	0	0	-
527	-2112	nbar0	84	490	501	0	0	0	0	-
528	111	(pi0)	-84	490	501	802	803	0	0	-
529	111	(pi0)	-84	490	501	804	805	0	0	
530	331	(eta')	-84	490	501	806	808	0	0	-
531	-211	pi-	84	490	501	0	0	0	0	-
532	1	(d)	-71	384	384	539	562	609	0	-
533	21	(g)	-71	340	340	539	562	598	609	-
534	21	(g)	-71	338	338	539	562	502	598	
535	21	(g)	-71	318	318	539	562	705	502	1
536	21	(g)	-71	357	357	539	562	503	705	
537	21	(g)	-71	354	354	539	562	687	503	-
538	-2	(ubar)	-71	328	328	539	562	0	687	-
539	-213	(rho-)	-83	532	538	661	662	0	0	-
540	113	(rho0)	-83	532	538	663	664	0	0	-
541	321	K+	83	532	538	0	0	0	0	-
542	-313	(K*bar0)	-83	532	538	665	666	0	0	
543	-211	pi-	83	532	538	0	0	0	0	-
544	213	(rho+)	-83	532	538	667	668	0	0	-
545	111	(pi0)	-83	532	538	809	810	0	0	
546	311	(K0)	-83	532	538	669	669	0	0	-
547	-323	(K*-)	-83	532	538	670	671	0	0	
548	211	pi+	83	532	538	0	0	0	0	-
549	221	(eta)	-83	532	538	811	812	0	0	
550	-211	pi-	83	532	538	0	0	0	0	-
551	211	pi+	84	532	538	0	0	0	0	
552	223	(omega)	-84	532	538	813	815	0	0	
553	-213	(rho-)	-84	532	538	672	673	0	0	
554	113	(rho0)	-84	532	538	674	675	0	0	
555	113	(rho0)	-84	532	538	676	677	0	0	
556	111	(pi0)	-84	532	538	816	817	0	0	
557	111	(pi0)	-84	532	538	818	819	0	0	
558	213	(rho+)	-84	532	538	678	679	0	0	-
559	-211	pi-	84	532	538	0	0	0	0	
560	211	pi+	84	532	538	0	0	0	0	
561	2112	n0	84	532	538	0	0	0	0	-
562	-2212	pbar-	84	532	538	0	0	0	0	-
563	1	(d)	-71	373	373	565	580	572	0	-
564	2203	(uu_1)	-71	385	385	565	580	0	572	
565	-213	(rho-)	-83	563	564	680	681	0	0	-

566	111	(pi0)	-83	563	564	820	821	0	0	-
567	221	(eta)	-83	563	564	822	823	0	0	-
568	213	(rho+)	-83	563	564	682	683	0	0	
569	-211	pi-	83	563	564	0	0	0	0	-
570	211	pi+	84	563	564	0	0	0	0	
571	221	(eta)	-84	563	564	824	826	0	0	-
572	113	(rho0)	-84	563	564	684	685	0	0	-
573	311	(K0)	-84	563	564	686	686	0	0	-
574	-311	(Kbar0)	-84	563	564	687	687	0	0	
575	313	(K*0)	-84	563	564	688	689	0	0	-
576	-311	(Kbar0)	-84	563	564	690	690	0	0	
577	-211	pi-	84	563	564	0	0	0	0	-
578	111	(pi0)	-84	563	564	827	828	0	0	
579	2212	p+	84	563	564	0	0	0	0	-
580	211	pi+	84	563	564	0	0	0	0	
581	2	(u)	-71	372	372	585	612	570	0	
582	21	(g)	-71	380	380	585	612	739	570	-
583	21	(g)	-71	369	369	585	612	671	739	
584	-1	(dbar)	-71	364	364	585	612	0	671	
585	213	(rho+)	-83	581	584	691	692	0	0	
586	113	(rho0)	-83	581	584	693	694	0	0	
587	-211	pi-	83	581	584	0	0	0	0	
588	111	(pi0)	-83	581	584	829	830	0	0	
589	113	(rho0)	-83	581	584	695	696	0	0	-
590	213	(rho+)	-83	581	584	697	698	0	0	
591	113	(rho0)	-83	581	584	699	700	0	0	-
592	113	(rho0)	-83	581	584	701	702	0	0	
593	-211	pi-	83	581	584	0	0	0	0	
594	111	(pi0)	-83	581	584	831	832	0	0	-
595	113	(rho0)	-83	581	584	703	704	0	0	-
596	223	(omega)	-83	581	584	833	835	0	0	
597	111	(pi0)	-83	581	584	836	837	0	0	
598	211	pi+	83	581	584	0	0	0	0	-
599	221	(eta)	-83	581	584	838	839	0	0	
600	-211	pi-	84	581	584	0	0	0	0	
601	211	pi+	84	581	584	0	0	0	0	-
602	-213	(rho-)	-84	581	584	705	706	0	0	
603	211	pi+	84	581	584	0	0	0	0	-
604	-213	(rho-)	-84	581	584	707	708	0	0	-
605	213	(rho+)	-84	581	584	709	710	0	0	-
606	221	(eta)	-84	581	584	840	842	0	0	
607	-211	pi-	84	581	584	0	0	0	0	-
608	211	pi+	84	581	584	0	0	0	0	-
609	-211	pi-	84	581	584	0	0	0	0	
610	213	(rho+)	-84	581	584	711	712	0	0	
611	-213	(rho-)	-84	581	584	713	714	0	0	
612	213	(rho+)	-84	581	584	715	716	0	0	
613	-211	pi-	91	392	0	0	0	0	0	
614	111	(pi0)	-91	392	0	843	844	0	0	-
615	211	pi+	91	393	0	0	0	0	0	-
616	111	(pi0)	-91	393	0	845	846	0	0	-
617	-211	pi-	91	407	0	0	0	0	0	
618	111	(pi0)	-91	407	0	847	848	0	0	
619	211	pi+	91	412	0	0	0	0	0	
620	111	(pi0)	-91	412	0	849	850	0	0	
621	211	pi+	91	418	0	0	0	0	0	-
622	111	(pi0)	-91	418	0	851	852	0	0	
623	-211	pi-	91	419	0	0	0	0	0	-

.

624	111	(pi0)	-91	419	0	853	854	0	0	
625	211	pi+	91	423	0	0	0	0	0	-
626	-211	pi-	91	423	0	0	0	0	0	-
627	310	(K_S0)	-91	429	429	855	856	0	0	
628	211	pi+	91	431	0	0	0	0	0	
629	-211	pi-	91	431	0	0	0	0	0	
630	211	pi+	91	444	0	0	0	0	0	
631	111	(pi0)	-91	444	0	857	858	0	0	-
632	211	pi+	91	446	0	0	0	0	0	-
633	111	(pi0)	-91	446	0	859	860	0	0	
634	211	pi+	91	452	0	0	0	0	0	-
635	111	(pi0)	-91	452	0	861	862	0	0	
636	-211	pi-	91	453	0	0	0	0	0	
637	111	(pi0)	-91	453	0	863	864	0	0	-
638	211	pi+	91	454	0	0	0	0	0	
639	111	(pi0)	-91	454	0	865	866	0	0	-
640	211	pi+	91	456	0	0	0	0	0	
641	-211	pi-	91	456	0	0	0	0	0	-
642	311	(K0)	-91	468	0	717	717	0	0	
643	211	pi+	91	468	0	0	0	0	0	
644	211	pi+	91	472	0	0	0	0	0	
645	-211	pi-	91	472	0	0	0	0	0	-
646	-321	K-	91	479	0	0	0	0	0	-
647	211	pi+	91	479	0	0	0	0	0	
648	211	pi+	91	508	0	0	0	0	0	
649	-211	pi-	91	508	0	0	0	0	0	
650	-211	pi-	91	511	0	0	0	0	0	-
651	111	(pi0)	-91	511	0	867	868	0	0	-
652	211	pi+	91	513	0	0	0	0	0	-
653	-211	pi-	91	513	0	0	0	0	0	
654	211	pi+	91	516	0	0	0	0	0	
655	-211	pi-	91	516	0	0	0	0	0	
656	311	(K0)	-91	521	0	718	718	0	0	-
657	111	(pi0)	-91	521	0	869	870	0	0	-
658	130	K_L0	91	522	522	0	0	0	0	
659	-211	pi-	91	523	0	0	0	0	0	
660	111	(pi0)	-91	523	0	871	872	0	0	-
661	-211	pi-	91	539	0	0	0	0	0	-
662	111	(pi0)	-91	539	0	873	874	0	0	-
663	211	pi+	91	540	0	0	0	0	0	-
664	-211	pi-	91	540	0	0	0	0	0	-
665	-321	K-	91	542	0	0	0	0	0	
666	211	pi+	91	542	0	0	0	0	0	
667	211	pi+	91	544	0	0	0	0	0	-
668	111	(pi0)	-91	544	0	875	876	0	0	
669	130	K_L0	91	546	546	0	0	0	0	-
670	-311	(Kbar0)	-91	547	0	719	719	0	0	
671	-211	pi-	91	547	0	0	0	0	0	
672	-211	pi-	91	553	0	0	0	0	0	
673	111	(pi0)	-91	553	0	877	878	0	0	
674	211	pi+	91	554	0	0	0	0	0	
675	-211	pi-	91	554	0	0	0	0	0	
676	211	pi+	91	555	0	0	0	0	0	
677	-211	pi-	91	555	0	0	0	0	0	
678	211	pi+	91	558	0	0	0	0	0	
679	111	(pi0)	-91	558	0	879	880	0	0	-
680	-211	pi-	91	565	0	0	0	0	0	-
681	111	(pi0)	-91	565	0	881	882	0	0	

682	211	pi+	91	568	0	0	0	0	0	
683	111	(pi0)	-91	568	0	883	884	0	0	-
684	211	pi+	91	572	0	0	0	0	0	-
685	-211	pi-	91	572	0	0	0	0	0	
686	310	(K_S0)	-91	573	573	885	886	0	0	-
687	130	K_L0	91	574	574	0	0	0	0	
688	321	K+	91	575	0	0	0	0	0	-
689	-211	pi-	91	575	0	0	0	0	0	
690	310	(K_S0)	-91	576	576	887	888	0	0	
691	211	pi+	91	585	0	0	0	0	0	
692	111	(pi0)	-91	585	0	889	890	0	0	-
693	211	pi+	91	586	0	0	0	0	0	-
694	-211	pi-	91	586	0	0	0	0	0	
695	211	pi+	91	589	0	0	0	0	0	-
696	-211	pi-	91	589	0	0	0	0	0	-
697	211	pi+	91	590	0	0	0	0	0	
698	111	(pi0)	-91	590	0	891	892	0	0	
699	111	(pi0)	-91	591	0	893	894	0	0	-
700	22	gamma	91	591	0	0	0	0	0	-
701	211	pi+	91	592	0	0	0	0	0	
702	-211	pi-	91	592	0	0	0	0	0	-
703	211	pi+	91	595	0	0	0	0	0	-
704	-211	pi-	91	595	0	0	0	0	0	-
705	-211	pi-	91	602	0	0	0	0	0	
706	111	(pi0)	-91	602	0	895	896	0	0	-
707	-211	pi-	91	604	0	0	0	0	0	-
708	111	(pi0)	-91	604	0	897	898	0	0	
709	211	pi+	91	605	0	0	0	0	0	-
710	111	(pi0)	-91	605	0	899	900	0	0	-
711	211	pi+	91	610	0	0	0	0	0	-
712	111	(pi0)	-91	610	0	901	902	0	0	
713	-211	pi-	91	611	0	0	0	0	0	
714	111	(pi0)	-91	611	0	903	904	0	0	
715	211	pi+	91	612	0	0	0	0	0	
716	111	(pi0)	-91	612	0	905	906	0	0	
717	310	(K_S0)	-91	642	642	907	908	0	0	
718	310	(K_S0)	-91	656	656	909	910	0	0	-
719	130	K_L0	91	670	670	0	0	0	0	
720	211	pi+	91	408	0	0	0	0	0	
721	-211	pi-	91	408	0	0	0	0	0	
722	111	(pi0)	-91	408	0	911	912	0	0	
723	211	pi+	91	409	0	0	0	0	0	
724	-211	pi-	91	409	0	0	0	0	0	
725	111	(pi0)	-91	409	0	913	914	0	0	
726	22	gamma	91	410	0	0	0	0	0	
727	22	gamma	91	410	0	0	0	0	0	
728	22	gamma	91	411	0	0	0	0	0	
729	22	gamma	91	411	0	0	0	0	0	
730	211	pi+	91	413	0	0	0	0	0	
731	-211	pi-	91	413	0	0	0	0	0	
732	111	(pi0)	-91	413	0	915	916	0	0	
733	111	(pi0)	-91	414	0	917	918	0	0	
734	111	(pi0)	-91	414	0	919	920	0	0	
735	111	(pi0)	-91	414	0	921	922	0	0	
736	22	gamma	91	415	0	0	0	0	0	
737	22	gamma	91	415	0	0	0	0	0	
738	22	gamma	91	417	0	0	0	0	0	-
739	22	gamma	91	417	0	0	0	0	0	

740	211	pi+	91	434	0	0	0	0	0	
741	-211	pi-	91	434	0	0	0	0	0	-
742	111	(pi0)	-91	434	0	923	925	0	0	-
743	511	(B0)	-91	437	0	926	927	0	0	
744	22	gamma	91	437	0	0	0	0	0	
745	22	gamma	91	443	0	0	0	0	0	-
746	22	gamma	91	443	0	0	0	0	0	-
747	211	pi+	91	448	0	0	0	0	0	
748	-211	pi-	91	448	0	0	0	0	0	
749	221	(eta)	-91	448	0	928	929	0	0	
750	22	gamma	91	450	0	0	0	0	0	
751	22	gamma	91	450	0	0	0	0	0	
752	22	gamma	91	455	0	0	0	0	0	-
753	22	gamma	91	455	0	0	0	0	0	-
754	22	gamma	91	471	0	0	0	0	0	
755	22	gamma	91	471	0	0	0	0	0	
756	22	gamma	91	474	0	0	0	0	0	
757	22	gamma	91	474	0	0	0	0	0	
758	22	gamma	91	476	0	0	0	0	0	-
759	22	gamma	91	476	0	0	0	0	0	
760	22	gamma	91	480	0	0	0	0	0	
761	22	gamma	91	480	0	0	0	0	0	-
762	211	pi+	91	481	0	0	0	0	0	
763	-211	pi-	91	481	0	0	0	0	0	-
764	111	(pi0)	-91	481	0	930	931	0	0	
765	22	gamma	91	482	0	0	0	0	0	
766	22	gamma	91	482	0	0	0	0	0	
767	211	pi+	91	483	0	0	0	0	0	-
768	-211	pi-	91	483	0	0	0	0	0	
769	111	(pi0)	-91	483	0	932	933	0	0	
770	22	gamma	91	484	0	0	0	0	0	
771	22	gamma	91	484	0	0	0	0	0	
772	211	pi+	91	487	0	0	0	0	0	
773	-211	pi-	91	487	0	0	0	0	0	
774	111	(pi0)	-91	487	0	934	935	0	0	
775	-213	(rho-)	-91	502	0	936	937	0	0	-
776	4212	(Sigma_c+)	-91	502	0	938	939	0	0	-
777	-2114	(Deltabar0)	-91	502	0	940	941	0	0	-
778	22	gamma	91	506	0	0	0	0	0	-
779	22	gamma	91	506	0	0	0	0	0	-
780	211	pi+	91	509	0	0	0	0	0	-
781	-211	pi-	91	509	0	0	0	0	0	
782	111	(pi0)	-91	509	0	942	943	0	0	
783	211	pi+	91	510	0	0	0	0	0	
784	-211	pi-	91	510	0	0	0	0	0	
785	111	(pi0)	-91	510	0	944	945	0	0	
786	22	gamma	91	512	0	0	0	0	0	
787	22	gamma	91	512	0	0	0	0	0	
788	22	gamma	91	515	0	0	0	0	0	
789	22	gamma	91	515	0	0	0	0	0	
790	22	gamma	91	517	0	0	0	0	0	-
791	22	gamma	91	517	0	0	0	0	0	-
792	211	pi+	91	518	0	0	0	0	0	
793	-211	pi-	91	518	0	0	0	0	0	
794	111	(pi0)	-91	518	0	946	947	0	0	
795	22	gamma	91	519	0	0	0	0	0	
796	22	gamma	91	519	0	0	0	0	0	
797	111	(pi0)	-91	520	0	948	949	0	0	

798	111	(pi0)	-91	520	0	950	951	0	0	
799	111	(pi0)	-91	520	0	952	953	0	0	-
800	22	gamma	91	526	0	0	0	0	0	
801	22	gamma	91	526	0	0	0	0	0	-
802	22	gamma	91	528	0	0	0	0	0	-
803	22	gamma	91	528	0	0	0	0	0	-
804	22	gamma	91	529	0	0	0	0	0	
805	22	gamma	91	529	0	0	0	0	0	
806	111	(pi0)	-91	530	0	954	955	0	0	-
807	111	(pi0)	-91	530	0	956	957	0	0	-
808	221	(eta)	-91	530	0	958	959	0	0	-
809	22	gamma	91	545	0	0	0	0	0	
810	22	gamma	91	545	0	0	0	0	0	
811	22	gamma	91	549	0	0	0	0	0	
812	22	gamma	91	549	0	0	0	0	0	
813	211	pi+	91	552	0	0	0	0	0	
814	-211	pi-	91	552	0	0	0	0	0	
815	111	(pi0)	-91	552	0	960	961	0	0	
816	22	gamma	91	556	0	0	0	0	0	
817	22	gamma	91	556	0	0	0	0	0	
818	22	gamma	91	557	0	0	0	0	0	
819	22	gamma	91	557	0	0	0	0	0	
820	22	gamma	91	566	0	0	0	0	0	-
821	22	gamma	91	566	0	0	0	0	0	-
822	22	gamma	91	567	0	0	0	0	0	-
823	22	gamma	91	567	0	0	0	0	0	
824	211	pi+	91	571	0	0	0	0	0	-
825	-211	pi-	91	571	0	0	0	0	0	-
826	111	(pi0)	-91	571	0	962	963	0	0	
827	22	gamma	91	578	0	0	0	0	0	-
828	22	gamma	91	578	0	0	0	0	0	
829	22	gamma	91	588	0	0	0	0	0	
830	22	gamma	91	588	0	0	0	0	0	-
831	22	gamma	91	594	0	0	0	0	0	
832	22	gamma	91	594	0	0	0	0	0	-
833	211	pi+	91	596	0	0	0	0	0	
834	-211	pi-	91	596	0	0	0	0	0	
835	111	(pi0)	-91	596	0	964	965	0	0	
836	22	gamma	91	597	0	0	0	0	0	
837	22	gamma	91	597	0	0	0	0	0	-
838	22	gamma	91	599	0	0	0	0	0	-
839	22	gamma	91	599	0	0	0	0	0	
840	111	(pi0)	-91	606	0	966	967	0	0	
841	111	(pi0)	-91	606	0	968	969	0	0	
842	111	(pi0)	-91	606	0	970	971	0	0	
843	22	gamma	91	614	0	0	0	0	0	-
844	22	gamma	91	614	0	0	0	0	0	-
845	22	gamma	91	616	0	0	0	0	0	-
846	22	gamma	91	616	0	0	0	0	0	-
847	22	gamma	91	618	0	0	0	0	0	
848	22	gamma	91	618	0	0	0	0	0	
849	22	gamma	91	620	0	0	0	0	0	
850	22	gamma	91	620	0	0	0	0	0	
851	22	gamma	91	622	0	0	0	0	0	
852	22	gamma	91	622	0	0	0	0	0	
853	22	gamma	91	624	0	0	0	0	0	
854	22	gamma	91	624	0	0	0	0	0	
855	111	(pi0)	-91	627	0	972	973	0	0	

856	111	(pi0)	-91	627	0	974	975	0	0	
857	22	gamma	91	631	0	0	0	0	0	-
858	22	gamma	91	631	0	0	0	0	0	-
859	22	gamma	91	633	0	0	0	0	0	
860	22	gamma	91	633	0	0	0	0	0	
861	22	gamma	91	635	0	0	0	0	0	
862	22	gamma	91	635	0	0	0	0	0	
863	22	gamma	91	637	0	0	0	0	0	-
864	22	gamma	91	637	0	0	0	0	0	-
865	22	gamma	91	639	0	0	0	0	0	-
866	22	gamma	91	639	0	0	0	0	0	
867	22	gamma	91	651	0	0	0	0	0	
868	22	gamma	91	651	0	0	0	0	0	-
869	22	gamma	91	657	0	0	0	0	0	-
870	22	gamma	91	657	0	0	0	0	0	-
871	22	gamma	91	660	0	0	0	0	0	-
872	22	gamma	91	660	0	0	0	0	0	-
873	22	gamma	91	662	0	0	0	0	0	-
874	22	gamma	91	662	0	0	0	0	0	-
875	22	gamma	91	668	0	0	0	0	0	-
876	22	gamma	91	668	0	0	0	0	0	
877	22	gamma	91	673	0	0	0	0	0	
878	22	gamma	91	673	0	0	0	0	0	
879	22	gamma	91	679	0	0	0	0	0	-
880	22	gamma	91	679	0	0	0	0	0	-
881	22	gamma	91	681	0	0	0	0	0	
882	22	gamma	91	681	0	0	0	0	0	-
883	22	gamma	91	683	0	0	0	0	0	-
884	22	gamma	91	683	0	0	0	0	0	-
885	211	pi+	91	686	0	0	0	0	0	
886	-211	pi-	91	686	0	0	0	0	0	-
887	211	pi+	91	690	0	0	0	0	0	
888	-211	pi-	91	690	0	0	0	0	0	
889	22	gamma	91	692	0	0	0	0	0	-
890	22	gamma	91	692	0	0	0	0	0	-
891	22	gamma	91	698	0	0	0	0	0	
892	22	gamma	91	698	0	0	0	0	0	
893	22	gamma	91	699	0	0	0	0	0	
894	22	gamma	91	699	0	0	0	0	0	-
895	22	gamma	91	706	0	0	0	0	0	
896	22	gamma	91	706	0	0	0	0	0	-
897	22	gamma	91	708	0	0	0	0	0	
898	22	gamma	91	708	0	0	0	0	0	
899	22	gamma	91	710	0	0	0	0	0	-
900	22	gamma	91	710	0	0	0	0	0	-
901	22	gamma	91	712	0	0	0	0	0	
902	22	gamma	91	712	0	0	0	0	0	
903	22	gamma	91	714	0	0	0	0	0	
904	22	gamma	91	714	0	0	0	0	0	
905	22	gamma	91	716	0	0	0	0	0	
906	22	gamma	91	716	0	0	0	0	0	
907	211	pi+	91	717	0	0	0	0	0	
908	-211	pi-	91	717	0	0	0	0	0	
909	111	(pi0)	-91	718	0	976	977	0	0	
910	111	(pi0)	-91	718	0	978	979	0	0	-
911	22	gamma	91	722	0	0	0	0	0	
912	22	gamma	91	722	0	0	0	0	0	
913	22	gamma	91	725	0	0	0	0	0	

914	22	gamma	91	725	0	0	0	0	0	
915	22	gamma	91	732	0	0	0	0	0	
916	22	gamma	91	732	0	0	0	0	0	
917	22	gamma	91	733	0	0	0	0	0	
918	22	gamma	91	733	0	0	0	0	0	
919	22	gamma	91	734	0	0	0	0	0	
920	22	gamma	91	734	0	0	0	0	0	
921	22	gamma	91	735	0	0	0	0	0	
922	22	gamma	91	735	0	0	0	0	0	
923	22	gamma	91	742	0	0	0	0	0	-
924	11	e-	91	742	0	0	0	0	0	-
925	-11	e+	91	742	0	0	0	0	0	-
926	433	(D*_s+)	-91	743	0	980	981	0	0	
927	-413	(D*-)	-91	743	0	982	983	0	0	
928	22	gamma	91	749	0	0	0	0	0	
929	22	gamma	91	749	0	0	0	0	0	
930	22	gamma	91	764	0	0	0	0	0	
931	22	gamma	91	764	0	0	0	0	0	
932	22	gamma	91	769	0	0	0	0	0	
933	22	gamma	91	769	0	0	0	0	0	
934	22	gamma	91	774	0	0	0	0	0	
935	22	gamma	91	774	0	0	0	0	0	
936	-211	pi-	91	775	0	0	0	0	0	-
937	111	(pi0)	-91	775	0	984	985	0	0	-
938	4122	(Lambda_c+)	-91	776	0	986	987	0	0	-
939	111	(pi0)	-91	776	0	988	989	0	0	-
940	-2112	nbar0	91	777	0	0	0	0	0	-
941	111	(pi0)	-91	777	0	990	991	0	0	
942	22	gamma	91	782	0	0	0	0	0	
943	22	gamma	91	782	0	0	0	0	0	
944	22	gamma	91	785	0	0	0	0	0	
945	22	gamma	91	785	0	0	0	0	0	
946	22	gamma	91	794	0	0	0	0	0	
947	22	gamma	91	794	0	0	0	0	0	-
948	22	gamma	91	797	0	0	0	0	0	
949	22	gamma	91	797	0	0	0	0	0	
950	22	gamma	91	798	0	0	0	0	0	
951	22	gamma	91	798	0	0	0	0	0	
952	22	gamma	91	799	0	0	0	0	0	
953	22	gamma	91	799	0	0	0	0	0	-
954	22	gamma	91	806	0	0	0	0	0	-
955	22	gamma	91	806	0	0	0	0	0	-
956	22	gamma	91	807	0	0	0	0	0	-
957	22	gamma	91	807	0	0	0	0	0	-
958	22	gamma	91	808	0	0	0	0	0	
959	22	gamma	91	808	0	0	0	0	0	-
960	22	gamma	91	815	0	0	0	0	0	
961	22	gamma	91	815	0	0	0	0	0	
962	22	gamma	91	826	0	0	0	0	0	
963	22	gamma	91	826	0	0	0	0	0	
964	22	gamma	91	835	0	0	0	0	0	
965	22	gamma	91	835	0	0	0	0	0	
966	22	gamma	91	840	0	0	0	0	0	
967	22	gamma	91	840	0	0	0	0	0	-
968	22	gamma	91	841	0	0	0	0	0	-
969	22	gamma	91	841	0	0	0	0	0	
970	22	gamma	91	842	0	0	0	0	0	
971	22	gamma	91	842	0	0	0	0	0	

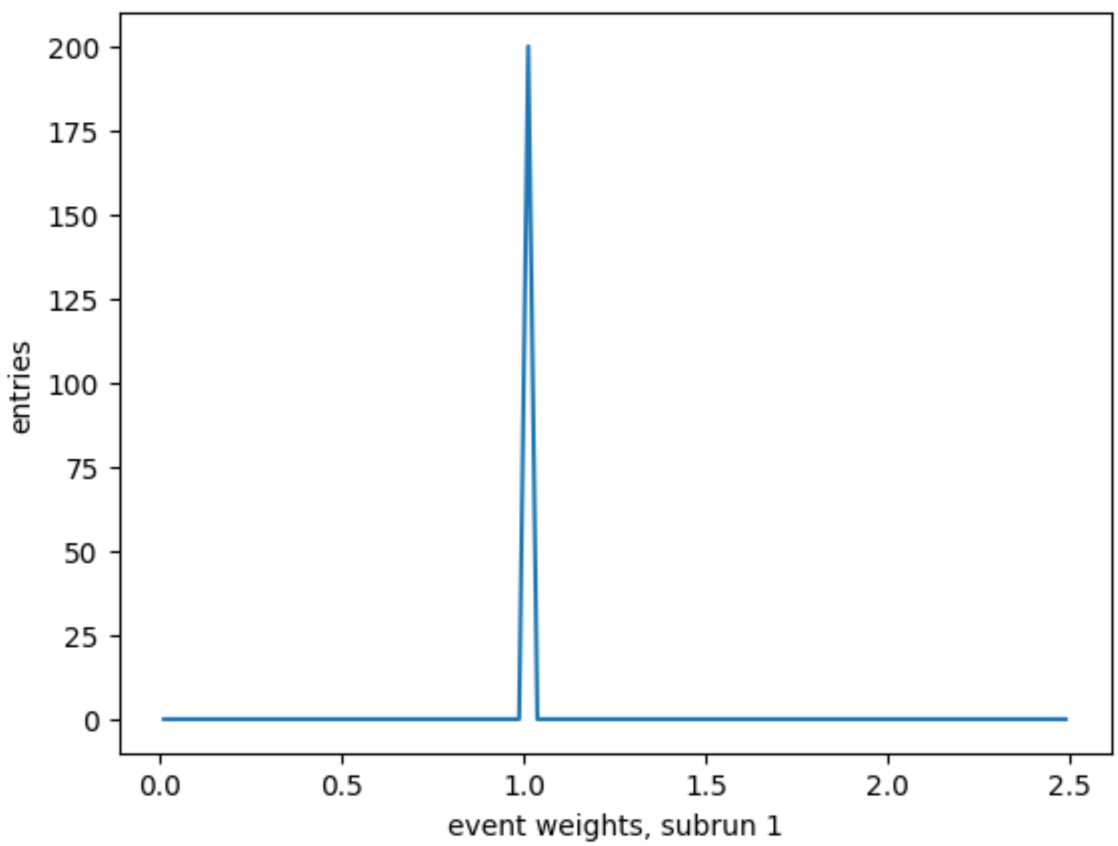
972	22	gamma	91	855	0	0	0	0	0	
973	22	gamma	91	855	0	0	0	0	0	
974	22	gamma	91	856	0	0	0	0	0	
975	22	gamma	91	856	0	0	0	0	0	
976	22	gamma	91	909	0	0	0	0	0	-
977	22	gamma	91	909	0	0	0	0	0	
978	22	gamma	91	910	0	0	0	0	0	-
979	22	gamma	91	910	0	0	0	0	0	-
980	431	(D_s+)	-91	926	0	992	994	0	0	
981	111	(pi0)	-91	926	0	995	996	0	0	
982	-421	(Dbar0)	-91	927	0	997	999	0	0	
983	-211	pi-	91	927	0	0	0	0	0	
984	22	gamma	91	937	0	0	0	0	0	-
985	22	gamma	91	937	0	0	0	0	0	-
986	2212	p+	91	938	0	0	0	0	0	-
987	331	(eta')	-91	938	0	1000	1002	0	0	-
988	22	gamma	91	939	0	0	0	0	0	-
989	22	gamma	91	939	0	0	0	0	0	-
990	22	gamma	91	941	0	0	0	0	0	
991	22	gamma	91	941	0	0	0	0	0	
992	331	(eta')	-91	980	0	1003	1005	0	0	
993	211	pi+	91	980	0	0	0	0	0	
994	111	(pi0)	-91	980	0	1006	1007	0	0	-
995	22	gamma	91	981	0	0	0	0	0	
996	22	gamma	91	981	0	0	0	0	0	
997	321	K+	91	982	0	0	0	0	0	
998	-211	pi-	91	982	0	0	0	0	0	
999	111	(pi0)	-91	982	0	1008	1009	0	0	
1000	211	pi+	91	987	0	0	0	0	0	-
1001	-211	pi-	91	987	0	0	0	0	0	-
1002	221	(eta)	-91	987	0	1010	1012	0	0	-
1003	211	pi+	91	992	0	0	0	0	0	
1004	-211	pi-	91	992	0	0	0	0	0	
1005	221	(eta)	-91	992	0	1013	1015	0	0	
1006	22	gamma	91	994	0	0	0	0	0	-
1007	22	gamma	91	994	0	0	0	0	0	
1008	22	gamma	91	999	0	0	0	0	0	
1009	22	gamma	91	999	0	0	0	0	0	
1010	111	(pi0)	-91	1002	0	1016	1017	0	0	-
1011	111	(pi0)	-91	1002	0	1018	1019	0	0	-
1012	111	(pi0)	-91	1002	0	1020	1021	0	0	
1013	111	(pi0)	-91	1005	0	1022	1023	0	0	-
1014	111	(pi0)	-91	1005	0	1024	1025	0	0	
1015	111	(pi0)	-91	1005	0	1026	1027	0	0	
1016	22	gamma	91	1010	0	0	0	0	0	-
1017	22	gamma	91	1010	0	0	0	0	0	
1018	22	gamma	91	1011	0	0	0	0	0	-
1019	22	gamma	91	1011	0	0	0	0	0	-
1020	22	gamma	91	1012	0	0	0	0	0	
1021	22	gamma	91	1012	0	0	0	0	0	-
1022	22	gamma	91	1013	0	0	0	0	0	-
1023	22	gamma	91	1013	0	0	0	0	0	-
1024	22	gamma	91	1014	0	0	0	0	0	
1025	22	gamma	91	1014	0	0	0	0	0	
1026	22	gamma	91	1015	0	0	0	0	0	
1027	22	gamma	91	1015	0	0	0	0	0	

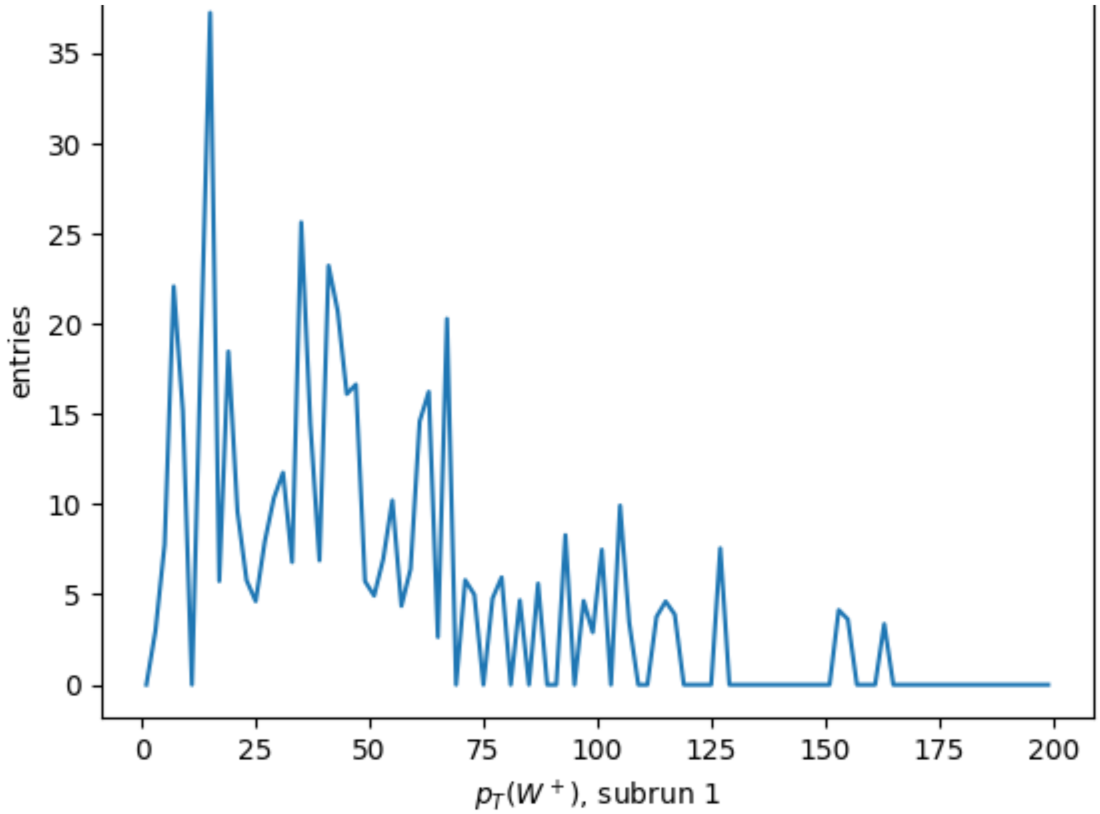
Charge sum: 2.000 Momentum sum:

```
--------  End PYTHIA Event Listing  ------------------------------------------
PYTHIA Warning in History::weightCKKWL: no ordered history found. Using unordered
PYTHIA Abort from Pythia::next: reached end of Les Houches Events File

*-------  PYTHIA Event and Cross Section Statistics  ----------------------------
|
| Subprocess                                              Code |            Number of events
|                                                              |       Tried   Selected   Ac
|                                                              |
|------------------------------------------------------------------------------------
|                                                              |
| Les Houches User Process(es)                            9999 |         200        200
|    ... whereof user classification code                    1 |         132        132
|    ... whereof user classification code                    2 |          68         68
|                                                              |
| sum                                                          |         200        200
|
*-------  End PYTHIA Event and Cross Section Statistics ----------------------------

*-------  PYTHIA Error and Warning Messages Statistics  ----------------------------
|
|  times   message
|
|      1   Abort from Pythia::next: reached end of Les Houches Events File
|     29   Warning in History::weightCKKWL: no ordered history found. Using unorde
|      1   Warning in MultipartonInteractions::init: maximum increased
|      1   Warning in SLHAinterface::initSLHA: ignoring DECAY tables
|      1   Warning in SLHAinterface::initSLHA: ignoring MASS entries
|      1   Warning in SLHAinterface::initSLHA: ignoring empty DECAY tables
|      1   Info from SLHAinterface::initSLHA: No MODSEL found, keeping internal SU
|
*-------  End PYTHIA Error and Warning Messages Statistics  ------------------------
```






```
PYTHIA Warning in Pythia::init: be aware that successive calls to init() do not cl
*-----------------------  SusyLesHouches SUSY/BSM Interface  ---------------------
| Last Change 12 Apr 2017 - P. Skands
| Parsing: w_production_tree_1.lhe
| (SLHA::readFile) line 159 - storing non-SLHA(2) block: yukawa
*--------------------------------------------------------------------------------
PYTHIA Warning in SLHAinterface::initSLHA: ignoring MASS entries for id = {1,2,3,4
PYTHIA Warning in SLHAinterface::initSLHA: ignoring DECAY tables for id = {23,24,1
PYTHIA Warning in SLHAinterface::initSLHA: ignoring empty DECAY tables for id = {6

*------------------ MEPS Merging Initialization  ------------------*
|                                                                  |
| CKKW-L merge                                                     |
|              pp>LEPTONS,NEUTRINOS  with up to  2 additional jets |
| Merging scale is defined by Lund pT, with value tMS =   15.0 GeV |
|                                                                  |
*-------------- END MEPS Merging Initialization  ------------------*


*-------  PYTHIA Process Initialization  --------------------------*
|                                                                  |
| We collide p+ with p+ at a CM energy of 7.000e+03 GeV            |
|                                                                  |
|------------------------------------------------------------------|
|                                                    |             |
| Subprocess                                    Code |   Estimated |
|                                                    |    max (mb) |
|                                                    |             |
|------------------------------------------------------------------|
|                                                    |             |
| Les Houches User Process(es)                  9999 |   2.592e-06 |
|                                                    |             |
*-------  End PYTHIA Process Initialization -----------------------*
```

```
*-------  PYTHIA Multiparton Interactions Initialization  ---------*
|                                                                  |
|                   sigmaNonDiffractive =    50.58 mb               |
|                                                                  |
|    pT0 =  2.28 gives sigmaInteraction =   243.78 mb: accepted    |
|                                                                  |
*-------  End PYTHIA Multiparton Interactions Initialization  -----*

*-------  PYTHIA Multiparton Interactions Initialization  ---------*
|                                                                  |
|                   sigmaNonDiffractive =    50.58 mb               |
|                                                                  |
|    pT0 =  2.28 gives sigmaInteraction =   242.07 mb: accepted    |
|                                                                  |
*-------  End PYTHIA Multiparton Interactions Initialization  -----*

*-------  PYTHIA Flag + Mode + Parm + Word + FVec + MVec + PVec + WVec Settings (c
|
| Name                                          |                      Now |
|                                               |                          |
| Beams:frameType                               |                        4 |
| Beams:LHEF                                    |  w_production_tree_1.lhe |
| Main:numberOfSubruns                          |                        3 |
| Main:subrun                                   |                        2 |
| Merging:doPTLundMerging                       |                       on |
| Merging:nJetMax                               |                        2 |
| Merging:Process                               |     pp>LEPTONS,NEUTRINOS |
| Merging:TMS                                   |                 15.00000 |
| SpaceShower:rapidityOrder                     |                      off |
| StringZ:useOldAExtra                          |                       on |
| UncertaintyBands:doVariations                 |                       on |
| UncertaintyBands:List                         |                          | alpha
|                                               |                          |   alp
|                                               |                          |   har
|                                               |                          |  hard
|
*-------  End PYTHIA Flag + Mode + Parm + Word + FVec + MVec + PVec + WVec Setting

--------  PYTHIA Particle Data Table (changed only)  ----------------------------

     id   name            antiName         spn chg col      m0        mWidth
            no onMode   bRatio   meMode     products

      6  t                tbar               2   2   1  173.00000    1.35678  165.
             0     1   0.0000664    0       24        1
             1     1   0.0017361    0       24        3
             2     1   0.9981975    0       24        5
             3     0   0.0000000    0       37        5

     25  h0                                  1   0   0  125.00000    0.00408  124.
             0     1   0.0000009    0        1       -1
             1     1   0.0000002    0        2       -2
             2     1   0.0002163    0        3       -3
             3     1   0.0288493    0        4       -4
             4     1   0.5768873    0        5       -5
             5     1   0.0000000    0        6       -6
             6     1   0.0000000    0       11      -11
             7     1   0.0002185    0       13      -13
```

8	1	0.0628569	0	15	-15
9	1	0.0850383	0	21	21
10	1	0.0022965	0	22	22
11	1	0.0016155	0	22	23
12	1	0.0261758	0	23	23
13	1	0.2158445	0	24	-24
14	1	0.0000000	103	1000022	1000022
15	1	0.0000000	103	1000023	1000022
16	1	0.0000000	103	1000023	1000023
17	1	0.0000000	103	1000025	1000022
18	1	0.0000000	103	1000025	1000023
19	1	0.0000000	103	1000025	1000025
20	1	0.0000000	103	1000035	1000022
21	1	0.0000000	103	1000035	1000023
22	1	0.0000000	103	1000035	1000025
23	1	0.0000000	103	1000035	1000035
24	1	0.0000000	103	1000024	-1000024
25	1	0.0000000	103	1000024	-1000037
26	1	0.0000000	103	1000037	-1000024
27	1	0.0000000	103	1000037	-1000037
28	1	0.0000000	103	1000001	-1000001
29	1	0.0000000	103	2000001	-2000001
30	1	0.0000000	103	1000001	-2000001
31	1	0.0000000	103	-1000001	2000001
32	1	0.0000000	103	1000002	-1000002
33	1	0.0000000	103	2000002	-2000002
34	1	0.0000000	103	1000002	-2000002
35	1	0.0000000	103	-1000002	2000002
36	1	0.0000000	103	1000003	-1000003
37	1	0.0000000	103	2000003	-2000003
38	1	0.0000000	103	1000003	-2000003
39	1	0.0000000	103	-1000003	2000003
40	1	0.0000000	103	1000004	-1000004
41	1	0.0000000	103	2000004	-2000004
42	1	0.0000000	103	1000004	-2000004
43	1	0.0000000	103	-1000004	2000004
44	1	0.0000000	103	1000005	-1000005
45	1	0.0000000	103	2000005	-2000005
46	1	0.0000000	103	1000005	-2000005
47	1	0.0000000	103	-1000005	2000005
48	1	0.0000000	103	1000006	-1000006
49	1	0.0000000	103	2000006	-2000006
50	1	0.0000000	103	1000006	-2000006
51	1	0.0000000	103	-1000006	2000006
52	1	0.0000000	103	1000011	-1000011
53	1	0.0000000	103	2000011	-2000011
54	1	0.0000000	103	1000011	-2000011
55	1	0.0000000	103	-1000011	2000011
56	1	0.0000000	103	1000012	-1000012
57	1	0.0000000	103	2000012	-2000012
58	1	0.0000000	103	1000012	-2000012
59	1	0.0000000	103	-1000012	2000012
60	1	0.0000000	103	1000013	-1000013
61	1	0.0000000	103	2000013	-2000013
62	1	0.0000000	103	1000013	-2000013
63	1	0.0000000	103	-1000013	2000013
64	1	0.0000000	103	1000014	-1000014
65	1	0.0000000	103	2000014	-2000014

```
                66     1   0.0000000  103  1000014 -2000014
                67     1   0.0000000  103 -1000014  2000014
                68     1   0.0000000  103  1000015 -1000015
                69     1   0.0000000  103  2000015 -2000015
                70     1   0.0000000  103  1000015 -2000015
                71     1   0.0000000  103 -1000015  2000015
                72     1   0.0000000  103  1000016 -1000016
                73     1   0.0000000  103  2000016 -2000016
                74     1   0.0000000  103  1000016 -2000016
                75     1   0.0000000  103 -1000016  2000016

 --------  End PYTHIA Particle Data Table  ---------------------------------------
```

```
# Print final histograms and info on merged cross section..
plot(pTWsum, r"$\sum p_T(W^+)$")
print(f"The inclusive cross section after merging is: {sigmaTotal:.4e} mb.")
```

Say we want to study a one-jet observable, e.g. the transverse momentum of the jet j in events with *exactly* one jet. In this case, we want to take "hard" jets from the $\mathrm{pp} \to \mathrm{W}j$ matrix element (ME), while "soft" jets should be modelled by parton-shower (PS) emissions off the $\mathrm{pp} \to \mathrm{W}$ states. In order to smoothly merge these two samples, we have to know in which measure "hard" is defined, and which value of this measure separates the hard and soft regions. In `main162ckkwl.cmnd`, these definitions are:

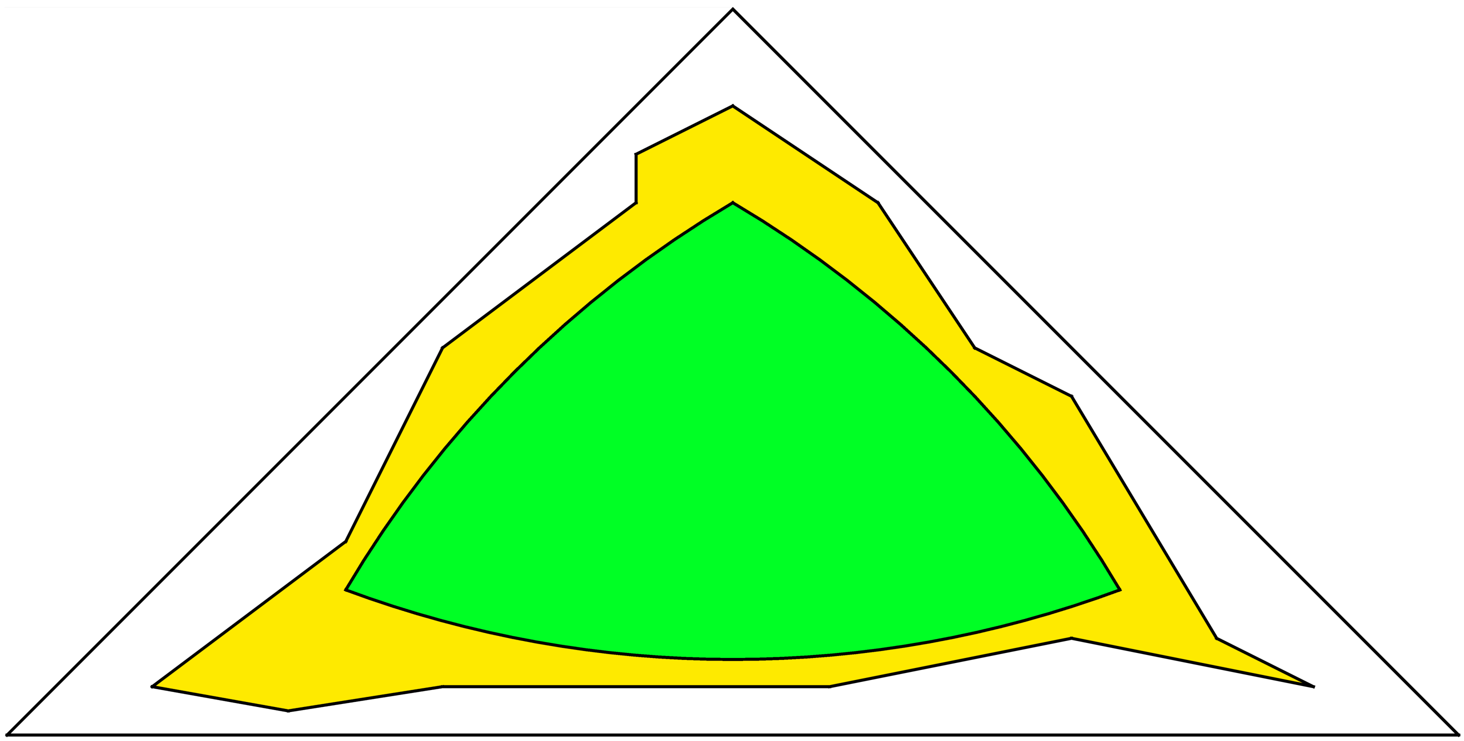
```
Merging:doKTMerging = on
Merging:ktType      = 2
Merging:TMS         = 30.
```

This will enable the merging procedure, with the merging scale defined by the minimal longitudinally invariant $k_\perp$ separation between partons (there are many other possibilities, by `ktType` value or by your own choice of merging procedure), with a merging scale $t_{\text{MS}} = 30$ GeV. Such a definition fixes what we mean when we talk about "hard" and "soft" jets:

type	definition
Hard jets:	$\min\{$any relative $k_\perp$ between sets of partons$\} > t_{\text{MS}}$
Soft jets:	$\min\{$any relative $k_\perp$ between sets of partons$\} < t_{\text{MS}}$

Thus, in order for the merging prescription to work, we need to remove phase space regions with $\min\{\text{any } k_\perp\} < t_{\text{MS}}$ from the $\mathrm{W} + 1$-parton matrix element calculation. Otherwise, there would be an overlap between the "soft jet" and "hard jet" samples.

This requirement means that the merging-scale definition should be implemented as a *cut in the matrix* *element generator*. Alternatively, it is possible to enforce the cut in *Pythia 8* internally, assuming that the ME is calculated with more inclusive (i.e. loose) cuts. This is illustrated below, where the triangle depicts the whole phase space, with soft or collinear divergences located on the edges. The yellow area symbolises the phase-space region used for the generation of the LHEF events, while the green area represents the phase space after *Pythia 8* has enforced the merging-scale cut on the input events. In order to correctly apply the merging-scale cut, the green area has to be fully contained inside the yellow one, i.e. the cut in the ME generator has to be more inclusive than the t_{MS}-cut. For optimal efficiency, the yellow and green areas should be identical. This can be the case in *MadGraph 5* (see MadGraph 5 : Going Beyond), when using the generation cuts `ktdurham` (corresponding to `Merging:doKTMerging = on`) and `ptpythia` (corresponding to `Merging:doPTLundMerging = on`).



After the merging-scale definition, we define the underlying process. To tell *Pythia 8* that we want to merge additional jets in W-boson production, we specify which is the core process, using *MadGraph* notation, where the final state is defined by the W^+ decay products rather than by the W^+ itself:

```
Merging:Process = pp>e+ve
```

in `main162ckkwl.cmnd`. Finally, the setting

```
Merging:nJetMax = 2
```

tells the program to include the pre-generated ME events for up to two additional jets.

In the merging example given above, the input file `main162ckkwl.cmnd` is read early on by the `pythia.readFile(...)` command. This gives access to the number of events to be read from `pythia.readFile(...)` command. This gives access to the number of events to be read from each LHE file, and the number of LHE files to be processed. The *subrun loop* then handles each LHE file, one at a time. Specifically, the

```
pythia.readFile("main162ckkwl.cmnd", iMerge)
```

uses the `iMerge` argument when reading the `main162ckkwl.cmnd` file, so that only those commands following the respective `Main:subrun = iMerge` labels are read. (Plus that everything before the first `Main:subrun` is re-read, but that does not matter since it stays the same.) Thus the proper LHE file is picked up for each jet multiplicity. The

```
Beams:frameType = 4
```

also informs *Pythia* that beam parameters should be read from the header section of the LHE file, and not set by the user.

Then we enter the event loop. The already-discussed difference in phase-space coverage can lead to a fair fraction of all input events being rejected. Thus the number of produced events can be lower than the requested `Main:numberOfEvents` one if the file is not large enough. (When no further events can be read the `pythia.next()` command will return `False`, so that the event loop can be exited at the end of the LHE file.) Those events that survive come with a weight

```
weight = pythia.infoPython().mergingWeight()
```

which contains Sudakov factors (to remove the double counting between samples of different multiplicity), α_{s} ratios (to incorporate the α_{s} running not available in matrix element generators), and ratios of parton distributions (to include variable factorization scales). This weight *must* be used when filling histogram bins, as is e.g. done by

```
pTWnow.fill( pTW, weight)
```

for the $p_{\perp}$ of the W boson. The sum of weights also goes into the calculation of the total generated cross section.

After the event loop, the contribution to the $p_{\perp}$ of the W boson from this particular multiplicity is normalised by

```
pTWnow *= 1e9*pythia.infoPython().sigmaGen()/(2.*pythia.infoPython().nAccepted())
```

where the ratio of the two `pythia.infoPython()` numbers is the weight per event, the `1e9` is for conversion from mb to pb, and the `2.` compensates for the bin width to give cross section per GeV. This number and more detailed statistics are printed to the terminal. As a final step, the contribution of the current subrun is added to the total histogram

```
pTWsum += pTWnow
```

and the subrun loop begins over with the next LHE file. The complete histogram, combining all multiplicities, is printed after the sub-run loop has concluded.

When you run the program above, note that some warning messages are issued routinely as part of the merging machinery, in the steps where a clustering history is found and where it is decided whether an event fails the merging scale cuts. Warnings from the SLHA interface also are irrelevant. So no reason to worry about any of that.

After the first run with the main program as is, you can try different variations.

- Convince yourself that the variation of the "merging weight" is moderate.
- Check in which $p_\perp$ regions which jet multiplicity contributes most.
- Study how the individual contributions and the sum changes when you run with a maximum of 1 or 0 jets, instead of the default 2.
- Compare the $p_\perp$ spectrum of the W with what you get from running the internal *Pythia* production process, by straightforward modifications of your original *Pythia* program.
- A major limitation is the size of the event files that come with the standard *Pythia* distribution, for space reasons. If you have a decent internet connection you can download larger files, with 100,000 events for each multiplicity up to $\mathrm{W}+4$ partons: `wp_tree_0.lhe.gz`, `wp_tree_1.lhe.gz`, `wp_tree_2.lhe.gz`, `wp_tree_3.lhe.gz`, and `wp_tree_4.lhe.gz`. With these files you can repeat the exercise above, and in particular check how much is gained by including the further $\mathrm{W}+3$ and $\mathrm{W}+4$ samples. To run through all events in the files takes a while, so check with a fraction of the sample to begin with, and be prepared to do something else while you wait for the full run to complete. Note that you will need to change `main162ckkwl.cmnd` to update these filenames with the `.gz` extension.
- Alternatively, if you are already a *MadGraph* user, you could generate your own LHE files and merge them. This would take some time, however, in particular for the higher multiplicities, so may not be an option.
- Use a jet finder to analyze the final state, and plot the $p_\perp$ spectra for the first, second, and third hardest jets, combining separate contributions similarly we already did for the W $p_\perp$ spectrum. Instructions how to use the built-in `SlowJet` jet finder can be found in Jet Finding.
- Check the variation of merged predictions with t_{MS}. You can do this by using an "inclusive" event sample, and having *Pythia* enforce a stronger t_{MS} cut. In which phase-space region is the t_{MS} variation most visible?
- Switch between the "wimpy" and "power" options for maximal shower scales by choosing values for `TimeShower:pTmaxMatch` and `SpaceShower:pTmaxMatch`. Are the effects more visible

in merged or non-merged predictions?

Once you have familiarised yourself with the example, you can experiment with more advanced settings in `main162ckkwl.cmnd`, as well as explore the other available `main162` command files: `main162mess.cmnd`, `main162nl3.cmnd`, `main162umeps.cmnd`, `main162unlops.cmnd`. When working directly with *Pythia* via its natice `C++` interface, you can also try `main164.cc` which is intended as a general matching and merging "blackbox" and can be run with several `.cmnd` files for different matching/merging options.

Some studies of Higgs production

The discovery of the Higgs boson has been one of the main accomplishments of the LHC to date. Generators have been part of that story, right from the early days when the ATLAS and CMS detectors were designed in such a way as to permit that discovery. This section offers exercises intended to explore the physics of Higgs production from various aspects, with *Pythia* as guide.

Production kinematics

The dominant production channel is $\mathrm{gg} \to \mathrm{H}^0$. To study the kinematics distribution of the Higgs, the existing top production program could easily be modified. Instead of switching on top production, use `HiggsSM:gg2H = on`. And instead of looking for the last top copy before decay, look for the last Higgs copy `prtH`, ie. the last particle with `id() == 25`. Once found the `prtH` methods can be used to extract and histogram Higgs kinematics distributions, like for the top. In addition to the transverse momentum `pT()` you can compare the distributions for true rapidity `y()` and for pseudorapidity `eta()`.

Production processes

While $\mathrm{gg} \to \mathrm{H}^0$ is the main Higgs production channel, it is not the only one. Do a run with `HiggsSM:all = on` to check which are implemented and their relative importance. Also figure out how you could generate one of the less frequent processes on its own, either from the HTML manual or by making some deductions from the output with all processes.

In order to get decent cross-section statistics faster, you can use `PartonLevel:all = off` to switch off everything except the hard-process generation itself. One price to pay is that the kinematics distributions for the Higgs are not meaningful. If instead complete events are generated you can study how the transverse-momentum distribution varies between processes. What are the reasons behind the significant differences?

Decay channels

Also the decay channels and branching ratios in the Higgs decay are of interest. Here no ready-made statistics routines exist, so you have to do it yourself. You already have the decaying Higgs, `prtH`. Since the standard decay modes are two-body, they can be accessed as:

```
dau1 = pythia.event[prtH.daughter1()]
dau2 = pythia.event[prtH.daughter2()]
```

and from that you can get the identities of the daughters. Introduce counters for all the decay modes you come to think of,that you use to derive and print branching ratios. Print the daughter identities in the leftover decays, where you did not yet have any counters, and iterate until you catch it all.

In this case, `PartonLevel:all = off` cannot be used, since then one does not get to the decays, at least not with the information in `pythia.event`. But you can combine

```
PartonLevel:ISR = off # to switch off initial-state radiation,
PartonLevel:FSR = off # to switch off final-state radiation,
PartonLevel:MPI = off # to switch off multiparton interactions, and
HadronLevel:all = off # to switch off hadronization and decays,
```

to get almost the same net time saving.

Mass distribution

By now the Higgs mass is pretty well pinned down, but you can check what the branching ratios would have been with another mass, e.g. by `25:m0 = 150.` for a 150 GeV Higgs.

On a related note, the Higgs mass is generated according to a Breit-Wigner distribution, convoluted with parton densities. Can you resolve this shape for the default Higgs mass? How does it change had the Higgs been heavier, say at 400 GeV?

Associated jets

Let us return to the different production channels, which give different event characteristics. Well-known is that the $\mathrm{qq} \to \mathrm{qqH}^0$ processes give rise to jets at large rapidities, that can be used for tagging purposes. This can be studied as follows.

The two relevant processes are `HiggsSM:ff2Hff(t:ZZ)` and `HiggsSM:ff2Hff(t:WW)` for $\mathrm{Z}^0\mathrm{Z}^0$ and $\mathrm{W}^+\mathrm{W}^-$ fusion (f here denotes a fermion; the same processes also exist e.g. at $\mathrm{e}^+\mathrm{e}^-$ colliders), that are to be compared with the standard $\mathrm{gg} \to \mathrm{H}^0$ one.

Find jets using the `SlowJet` class, see Jet finding, e.g. using the anti-$k_{\perp}$ algorithm with $R = 0.7$, $p_{\perp\mathrm{min}} = 30$~GeV and $\eta_{\mathrm{max}} = 5$.

One problem is that also the Higgs decay products can give jets, which we are not interested in here. To avoid this, we can switch off Higgs decays by `25:mayDecay = off`. This still leaves the Higgs itself in the event record. A call `prtH.statusNeg()` will negate its status code, and since `slowJet.analyze(...)` only considers the final particles, i.e. those with positive status, the Higgs is thus eliminated.

Now study the $p_{\perp}$ and rapidity spectrum of the hardest jets, and compare those distributions for the two processes. Also study how many jets are produced in the two cases.

Underlying event

Several mechanisms contribute to the overall particle production in Higgs events. This can be studied e.g. by histogramming the charged particle distribution.

You then need to have a counter set to zero for each new event. Inside the particle loop this counter should be incremented whenever the particle `isFinal()` and `isCharged()`. For the histogram, note that it can be treacherous to have bin limits at integers, where roundoff errors decide whichever way they go. In this particular case only even numbers are possible, so 100 bins from -1 to 399 would still be acceptable, for instance.

Once you have the distribution down for normal events, study what happens if you remove ISR, FSR and MPI one by one or all together. Also study the contribution of the Higgs decay itself to the multiplicity, e.g. by setting the Higgs stable. Reflect on why different combinations give the pattern they do, e.g. why ISR on/off makes a bigger difference when MPI is on than off.

Decay properties

The decay mode $\mathrm{H}^0 \to \mathrm{Z}^0\mathrm{Z}^0 \to \ell^+\ell^-\ell'^+\ell'^-$, $\ell, \ell' = \mathrm{e}, \mu$, is called the gold-plated one, since it stands out so well from backgrounds. It also offers angular correlations that probe the spin of a Higgs candidate.

For now consider a simpler, but still interesting, pair of distributions: the mass spectra of the two Z^0 decay products. Plot them for the lighter and the heavier of the two separately, and compare shapes and average values. To improve statistics, you can use `25:onMode = off` to switch off all decay channels, and then `25:onIfMatch = 23 23` to switch back on the decay to $\mathrm{Z}^0\mathrm{Z}^0$ (and nothing else). Further, neither ISR, FSR, MPI nor hadronization affect the mass distributions, so this allows some speedup.

How can one qualitatively understand why the two masses tend to be so far apart, rather than roughly comparable?

Comparison with Z^0 production

One of the key reference processes at hadron colliders is Z^0 production. To lowest order it only involves one process, $\mathrm{q}\overline{\mathrm{q}} \to \gamma^*/\mathrm{Z}^0$, accessible with `WeakSingleBoson:ffbar2gmZ = on`. One complication is that the process involves γ^*/Z^0 interference, and so a significant enhancement at low masses, even if the combined particle always is classified with code 23, however.

Compare the two processes $\mathrm{gg} \to \mathrm{H}^0$ and $\mathrm{q}\overline{\mathrm{q}} \to \gamma^*/\mathrm{Z}^0$, with respect to the $p_\perp$ distribution of the boson and the total charged multiplicity of the events. So as to remove the dependence on the difference in mass, you can set a specific mass range in the γ^*/Z^0 generation with `PhaseSpace:mHatMin = 124` and `PhaseSpace:mHatMax = 126`, to agree with the H^0 mass to $\pm$ 1 GeV.

Can you explain what is driving the differences in the $p_\perp$ and n_{chg} distributions between the two processes?

Further studies

If you have time left, you should take the opportunity to try a few other processes or options. Below are given some examples, but feel free to pick something else that you would be more interested in.

- One popular misconception is that the energy and momentum of a B meson has to be smaller than that of its mother b quark, and similarly for charm. The fallacy is twofold. Firstly, if the b quark is surrounded by nearby colour-connected gluons, the B meson may also pick up some of the momentum of these gluons. Secondly, the concept of smaller momentum is not Lorentz-frame-independent: if the other end of the b colour force field is a parton with a higher momentum (such as a beam remnant) the "drag" of the hadronization process may imply an acceleration in the lab frame (but a deceleration in the beam rest frame). To study this, simulate b production, e.g. the process `HardQCD:gg2bbbar`. Identify B/B^* mesons that come directly from the hadronization, for simplicity those with status code -83 or -84. In the former case the mother b quark is in the `mother1()` position, in the latter in `mother2()` (study a few event listings to see how it works). Plot the ratio of B to b energy to see what it looks like.

- One of the characteristics of multiparton-interactions (MPI) models is that they lead to strong long-range correlations, as observed in data. That is, if many hadrons are produced in one rapidity range of an event, then most likely this is an event where many MPI's occurred (and the impact parameter between the two colliding protons was small), and then one may expect a larger activity also at other rapidities. To study this, select two symmetrically located, one unit wide bins in rapidity (or pseudorapidity), with a variable central separation Δy: $[\Delta y/2, \Delta y/2+1]$ and $[-\Delta y/2-1, -\Delta y/2]$. For each event you may find n_F and n_B, the charged multiplicity in the "forward" and "backward" rapidity bins. Suitable averages over a sample of events then gives the forward-backward correlation coefficient

$$\rho_{FB}(\Delta y) = \frac{\langle n_F\, n_B \rangle - \langle n_F \rangle \langle n_B \rangle}{\sqrt{(\langle n_F^2 \rangle - \langle n_F \rangle^2)(\langle n_B^2 \rangle - \langle n_B \rangle^2)}} = \frac{\langle n_F\, n_B \rangle - \langle n_F \rangle}{}$$

where the last equality holds for symmetric distributions such as in pp and $\mathrm{\overline{p}p}$. Compare how $\rho_{FB}(\Delta y)$ changes for increasing $\Delta y = 0, 1, 2, 3, \ldots$, with and without MPI switched on (`PartonLevel:MPI = on/off`) for minimum-bias events (`SoftQCD:minBias = on`).

- Z^0 production to lowest order only involves one process, which is accessible with `WeakSingleBoson:ffbar2gmZ = on`. The problem here is that the process is $\mathrm{f\bar{f}} \to \gamma^*/\mathrm{Z}^0$ with full γ^*/Z^0 interference and so a signficiant enhancement at low masses. The combined particle is always classified with code `23`, however. So generate events and study the γ^*/Z^0 mass and $p_\perp$ distributions. Then restrict to a more "Z^0-like" mass range with `PhaseSpace:mHatMin = 75 and PhaseSpace:mHatMax = 120`.

- Use a jet clustering algorithm, e.g. one of the `SlowJet` options described in Jet finding, to study the number of jets found in association with the Z^0 above. You can switch off Z^0 decay with `23:mayDecay = no`, and negate its status code by `prtZ.statusNeg()`, so that it will not be included in the jet finding. Here `prtZ` is the last copy of the Z^0, cf. how the last top copy was found above. Again check the importance of FSR/ISR/MPI.

Note that the *Pythia* homepage contains two further tutorials, in addition to older editions of the current one. These share some of the introductory material, but then put the emphasis on two specific areas:

- a merging tutorial, showing the step-by-step construction of a relevant main program, and more details on possible merging approaches than found in the CKKW-L merging section of this notebook; and
- a BSM tutorial, describing how you can input events from Beyond-the-Standard-model scenarios into *Pythia*.

The Event Record

The event record is set up to store every step in the evolution from an initial low-multiplicity partonic process to a final high-multiplicity hadronic state, in the order that new particles are generated. The record is a vector of particles, that expands to fit the needs of the current event (plus some additional pieces of information not discussed here). Thus `event[i]` is the i'th particle of the current event, and you may study its properties by using various `event[i].method()` possibilities.

The `event.list()` listing provides the main properties of each particles, by column:

- `no`, the index number of the particle (`i` above);
- `id`, the PDG particle identity code (method `id()`);
- `name`, a plaintext rendering of the particle name (method `name()`), within brackets for initial or intermediate particles and without for final-state ones;
- `status`, the reason why a new particle was added to the event record (method `status()`);
- `mothers and daughters`, documentation on the event history (methods `mother1()`, `mother2()`, `daughter1()`, and `daughter2()`);
- `colours`, the colour flow of the process (methods `col()` and `acol()`);
- `p_x`, `p_y`, `p_z`, and `e`, the components of the momentum four-vector (p_x , p_y , p_z, E), in units of GeV with $c = 1$ (methods `px()`, `py()`, `pz()`, and `e()`);
- m, the mass, in units as above (method `m()`).

For a complete description of these and other particle properties (such as production and decay vertices, rapidity, $p_\perp$, etc), open the Particle Properties page of the HTML manual. For brief summaries on the less trivial of the ones above, read on.

Identity codes

A complete specification of the PDG codes is found in the Review of Particle Physics. An online listing is available from the PDG MC numbering review. A short summary of the most common id codes would be

ID	particle	ID	particle	ID	particle	ID	particle	ID	particle	ID	particle	ID	particle
1	d	11	e^-	21	g	111	π^0	223	ω	331	η'	2112	n
2	u	12	ν_e	22	γ	113	ρ^0	310	K^0_S	333	ϕ	2212	p
3	s	13	μ^-	23	Z^0	130	K^0_L	311	K^0	411	D^+	3112	Σ^-
4	c	14	ν_μ	24	W^+	211	π^+	313	K^{*0}	421	D^0	3122	Λ^0
5	b	15	τ^-	25	H^0	213	ρ^+	321	K^+	431	D^+_s	3212	Σ^0
6	t	16	ν_τ			221	η	323	K^{*+}	3222	Σ^+		

Antiparticles to the above, where existing as separate entities, are given with a negative sign. Note that simple meson and baryon codes are constructed from the constituent (anti)quark codes, with a final spin-state-counting digit $2s + 1$ (K^0_L and K^0_S being exceptions), and with a set of further rules to make the codes unambiguous.

Status codes

When a new particle is added to the event record, it is assigned a positive status code that describes why it has been added, as follows (see the `Particle::status` section of the Particle Properties page of the HTML manual for the meaning of each specific code):

code range	**explanation**
11 - 19	beam particles
21 - 29	particles of the hardest subprocess
31 - 39	particles of subsequent subprocesses in multiparton interactions
41 - 49	particles produced by initial-state-showers
51 - 59	particles produced by final-state-showers
61 - 69	particles produced by beam-remnant treatment
71 - 79	partons in preparation of hadronization process
81 - 89	primary hadrons produced by hadronization process
91 - 99	particles produced in decay process, or by Bose-Einstein effects

Whenever a particle is allowed to branch or decay further its status code is negated (but it is *never* removed from the event record), such that only particles in the final state remain with positive codes. The `isFinal()` method returns `True/False` for positive/negative status codes.

History information

The two mother and two daughter indices of each particle provide information on the history relationship between the different entries in the event record. The detailed rules depend on the particular physics step being described, as defined by the status code. As an example, in a $2 \to 2$ process $ab \to cd$, the locations of a and b would set the mothers of c and d, with the reverse relationship for daughters. When the two mother or daughter indices are not consecutive they define a range between the first and last entry, such as a string system consisting of several partons fragment into several hadrons.

There are also several special cases. One such is when "the same" particle appears as a second copy, e.g. because its momentum has been shifted by it taking a recoil in the dipole picture of parton showers. Then the original has both daughter indices pointing to the same particle, which in its turn has both mother pointers referring back to the original. Another special case is the description of ISR by backwards evolution, where the mother is constructed at a later stage than the daughter, and therefore appears below it in the event listing.

Colour flow information

The colour flow information is based on the Les Houches Accord convention (see Generic User

Process Interface for Event Generators). In it, the number of colours is assumed infinite, so that each new colour line can be assigned a new separate colour. These colours are given consecutive labels: `101`, `102`, `103`, ... A gluon has both a colour and an anticolour label, an (anti)quark only (anti)colour.

While colours are traced consistently through hard processes and parton showers, the subsequent beam-remnant-handling step often involves a drastic change of colour labels. Firstly, previously unrelated colours and anticolours taken from the beams may at this stage be associated with each other, and be relabelled accordingly. Secondly, it appears that the close space-time overlap of many colour fields leads to reconnections, i.e. a swapping of colour labels, that tends to reduce the total length of field lines.

Some facilities

The *Pythia* package contains some facilities that are not part of the core generation mission, but are useful for standalone running, notably at summer schools. Here we give some brief info on histograms and jet finding.

Histograms

For real-life applications you may want to use sophisticated histogramming programs like `ROOT`, which can take time to install and learn. Within the time at our disposal, we therefore stick with the very primitive `Hist` class. Here is a simple overview of what is involved.

As a first step you need to declare a histogram, with name, title, number of bins and x range (from, to), like

```
import pythia8mc as pythia8

pTH = pythia8.Hist("Higgs transverse momentum", 100, 0.0, 200.0)
```

Once declared, its contents can be added by repeated calls to fill

```
pTH.fill(22.7, 1.0)
```

where the first argument is the x value and the second the weight. Since the weight defaults to 1 the last argument could have been omitted in this case.

A set of overloaded operators have been defined, so that histograms can be added, subtracted, divided or multiplied by each other. Then the contents are modified accordingly bin by bin. Thus the difference between two histograms `data` and `theory` can be found as

```
data = pythia8.Hist("Higgs transverse momentum - data", 100, 0.0, 200.0)
theory = pythia8.Hist("Higgs transverse momentum - theory", 100, 0.0, 200.0)
diff = data - theory
```

assuming that `diff`, `data` and `theory` have been booked with the same number of bins and x range.

Also overloaded operations with double real numbers are available. Again these four operations are defined bin by bin, i.e. the corresponding amount is added to, subtracted from, multiplied by or divided by each bin. The double number can come before or after the histograms, with obvious results. Thus the inverse of a histogram result is given by `1./result`. The two kind of operations can be combined.

A histogram can be printed by making use of `print`, e.g.

```
print(pTH)
```

```
      2025-07-03 17:36       Higgs transverse momentum

            1.00*10^ 0            X
            0.96*10^ 0            X
            0.92*10^ 0            X
            0.88*10^ 0            X
            0.84*10^ 0            X
            0.80*10^ 0            X
            0.76*10^ 0            X
            0.72*10^ 0            X
            0.68*10^ 0            X
            0.64*10^ 0            X
            0.60*10^ 0            X
            0.56*10^ 0            X
            0.52*10^ 0            X
            0.48*10^ 0            X
            0.44*10^ 0            X
            0.40*10^ 0            X
            0.36*10^ 0            X
            0.32*10^ 0            X
            0.28*10^ 0            X
            0.24*10^ 0            X
            0.20*10^ 0            X
            0.16*10^ 0            X
            0.12*10^ 0            X
            0.08*10^ 0            X
            0.04*10^ 0            X

              Contents
                *10^ 0  0000000000010000000000000000000000000000000000000000000000000000
                *10^-1  0000000000000000000000000000000000000000000000000000000000000000
                *10^-2  0000000000000000000000000000000000000000000000000000000000000000
                *10^-3  0000000000000000000000000000000000000000000000000000000000000000
```

```
       Low edge
         *10^ 2  000000000000000000000000000000000000000000000000001111111111111
         *10^ 1  000001111122222333334444455555666667777788888999990000011111222
         *10^ 0  024680246802468024680246802468024680246802468024680246802468024

Entries   =          1   xMin =     0.000   Underflow =    0.0000   Mean   =    23
SumW(in) =    1.0000   xMax =   200.000   Overflow  =    0.0000   Median =    23
```

The printout format is inspired by the old HBOOK one. To understand how to read it, consider the simplified example

```
3.50*10^ 2  9
3.00*10^ 2  X   7
2.50*10^ 2  X  1X
2.00*10^ 2  X6 XX
1.50*10^ 2  XX5XX
1.00*10^ 2  XXXXX
0.50*10^ 2  XXXXX

  Contents
    *10^ 2  31122
    *10^ 1  47208
    *10^ 0  79373

  Low edge  --
    *10^ 1  10001
    *10^ 0  05050
```

The key feature is that the `Contents` and `Low edge` have to be read vertically. For instance, the first bin has the contents $3 \times 10^2 + 4 \times 10^1 + 7 \times 10^0 = 347$. Correspondingly, the other bins have contents 179, 123, 207 and 283. The first bin stretches from $-(1 \times 10^1 + 0 \times 10^0) = -10$ to the beginning of the second bin, at $-(0 \times 10^1 + 5 \times 10^0) = -5$.

The visual representation above the contents give a simple impression of the shape. An `X` means that the contents are filled up to this level, a digit in the topmost row the fraction to which the last level is filled. So the `9` of the first column indicates this bin is filled 9/10 of the way from $3.00 \times 10^2 = 300$ to $3.50 \times 10^2 = 350$, i.e. somewhere close to 345, or more precisely in the range 342.5 to 347.5.

The printout also provides some other information, such as the number of entries, i.e. how many times the histogram has been filled, the total weight inside the histogram, the total weight in underflow and overflow, and the mean value and root-mean-square width (disregarding underflow and overflow). The mean and width assumes that all the contents is in the middle of the respective bin. This is especially relevant when you plot a integer quantity, such as a multiplicity. Then it makes sense to book with limits that are half-integers, e.g.

```
multMPI = pythia8.Hist("number of multiparton interactions", 20, -0.5, 19.5)
```

so that the bins are centered at 0, 1, 2, ..., respectively. This also avoids ambiguities which bin gets to be filled if entries are exactly at the border between two bins. Also note that the `fill(xValue)` method automatically performs a cast to double precision where necessary, i.e. `xValue` can be an integer.

Histogram values can also be output to a file

```
pTH.table("filename")
```

which produces a two-column table, where the first column gives the center of each bin and the second one the corresponding bin content. This may be used for plotting e.g. with Gnuplot.

Jet finding

The `SlowJet` class offer jet finding by the $k_\perp$, Cambridge/Aachen and anti-$k_\perp$ algorithms. By default it is now a front end to the FJcore subset, extracted from the *FastJet* package (see [FastJet user manual]) and distributed as part of the *Pythia* package, and is therefore no longer slow. It is good enough for basic jet studies, but does not allow for jet pruning or other more sophisticated applications. (An interface to the full FastJet package is available for such uses.)

You set up `SlowJet` initially with

```
 slowJet = pythia8.SlowJet(pow, radius, pTjetMin, etaMax)
```

where `pow = -1` for anti-$k_\perp$ (recommended), `pow = 0` for Cambridge/Aachen, `pow = 1` for $k_\perp$, while `radius` is the R parameter, `pTjetMin` the minimum $p_\perp$ of jets, and `etaMax` the maximum pseudorapidity of the detector coverage.

```
import pythia8mc as pythia8

slowJet = pythia8.SlowJet(-1, 0.5, 5.0, 100.0)
```

Inside the event loop, you can analyze an event by a call

```
 slowJet.analyze(pythia.event)
```

The jets found can be listed by `slowJet.list()`, but this is only feasible for a few events. Instead you can use the following methods:

you can use the following methods:

```
slowJet.sizeJet() # gives the number of jets found,
slowJet.pT(i)     # gives the pT for the i'th jet, and
slowJet.y(i)      # gives the rapidity for the i'th jet.
```

The jets are ordered in falling $p_\perp$. Below is a more complete example.

```
# Create a Pythia object.
pythia = pythia8.Pythia("", False)
pythia.readString("HardQCD:hardbbbar = on")
pythia.readString("Print:quiet = on")
pythia.init()

# Generate an event.
pythia.next()

# Build the jets.
slowJet.analyze(pythia.event)

# List the event.
slowJet.list()
```

```
#--------------------------------------------------------------------------
#                      FastJet release 3.4.3 [fjcore]
#                  M. Cacciari, G.P. Salam and G. Soyez
#     A software package for jet finding and analysis at colliders
#                           http://fastjet.fr
#
# Please cite EPJC72(2012)1896 [arXiv:1111.6097] if you use this package
# for scientific work and optionally PLB641(2006)57 [hep-ph/0512210].
#
# FastJet is provided without warranty under the GNU GPL v2 or higher.
# It uses T. Chan's closest pair algorithm, S. Fortune's Voronoi code
# and 3rd party plugin jet algorithms. See COPYING file for details.
#--------------------------------------------------------------------------

 --  PYTHIA SlowJet(fjcore) Listing, p = -1, R = 0.500, pTjetMin =   5.000, etaMax

   no      pTjet      y       phi   mult      p_x        p_y        p_z         e
    0      7.040   -0.200    1.298     9      1.899      6.779     -1.514      7.66

 --------  End PYTHIA SlowJet Listing  --------------------------------------------
```