A question

When re-using phase space calculation results is enabled in Powheg, does it still check intelligently if they need to be recalculated?

The `nFinal` parameter in `main31`

= number of outgoing particles in the Born level matrix element.
Used internally to check if Powheg event was radiative or not.

nFinal

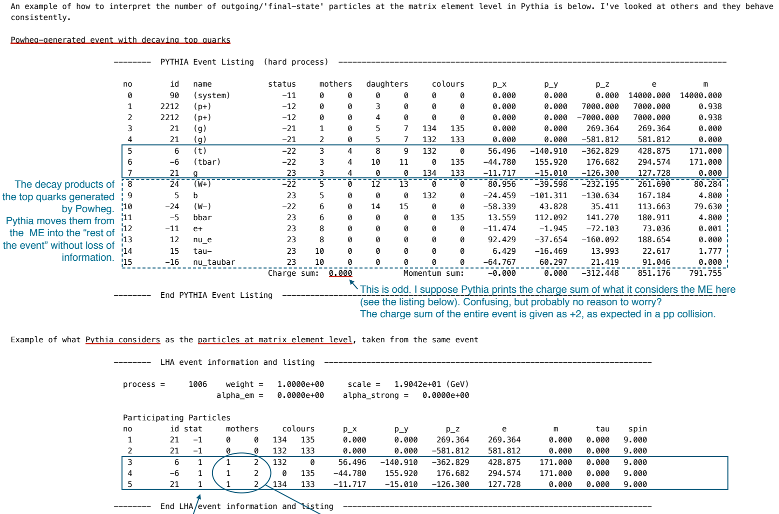
In a 2 → 2 process *at the Born level* (such as $\mathrm{t\bar{t}}$), `nFinal` should always be set to 2.

This does not depend on whether the Powheg simulates the top quark decay or not.

It also does not depend on whether the event is radiative or not.

See slide 4 for more information.

nFinal — some Pythia print-out with comments



An example of how to interpret the number of outgoing/'final-state' particles at the matrix element level in Pythia is below. I've looked at others and they behave consistently.

Powheg-generated event with decaying top quarks

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

    no        id   name            status     mothers   daughters     colours      p_x        p_y        p_z         e          m 
     0        90   (system)           -11     0     0     0     0     0     0      0.000      0.000      0.000  14000.000  14000.000
     1      2212   (p+)               -12     0     0     3     0     0     0      0.000      0.000   7000.000   7000.000      0.938
     2      2212   (p+)               -12     0     0     4     0     0     0      0.000      0.000  -7000.000   7000.000      0.938
     3        21   (g)                -21     1     0     5     7   134   135      0.000      0.000    269.364    269.364      0.000
     4        21   (g)                -21     2     0     5     7   132   133      0.000      0.000   -581.812    581.812      0.000
     5         6   (t)                -22     3     4     8     9   132     0     56.496   -140.910   -362.829    428.875    171.000
     6        -6   (tbar)             -22     3     4    10    11     0   135    -44.780    155.920    176.682    294.574    171.000
     7        21   g                   23     3     4     0     0   134   133    -11.717    -15.010   -126.300    127.728      0.000
     8        24   (W+)               -22     5     0    12    13     0     0     80.956    -39.598   -232.195    261.690     80.284
     9         5   b                   23     5     0     0     0   132     0    -24.459   -101.311   -130.634    167.184      4.800
    10       -24   (W-)               -22     6     0    14    15     0     0    -58.339     43.828     35.411    113.663     79.630
    11        -5   bbar                23     6     0     0     0     0   135     13.559    112.092    141.270    180.911      4.800
    12       -11   e+                  23     8     0     0     0     0     0    -11.474     -1.945    -72.103     73.036      0.001
    13        12   nu_e                23     8     0     0     0     0     0     92.429    -37.654   -160.092    188.654      0.000
    14        15   tau-                23    10     0     0     0     0     0      6.429    -16.469     13.993     22.617      1.777
    15       -16   nu_taubar           23    10     0     0     0     0     0    -64.767     60.297     21.419     91.046      0.000
                                   Charge sum:  0.000           Momentum sum:     -0.000      0.000   -312.448    851.176    791.755

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

The decay products of the top quarks generated by Powheg. Pythia moves them from the ME into the "rest of the event" without loss of information.

This is odd. I suppose Pythia prints the charge sum of what it considers the ME here (see the listing below). Confusing, but probably no reason to worry? The charge sum of the entire event is given as +2, as expected in a pp collision.

Example of what Pythia considers as the particles at matrix element level, taken from the same event

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

    process =     1006    weight =    1.0000e+00     scale =    1.9042e+01 (GeV)
                        alpha_em =    0.0000e+00     alpha_strong =    0.0000e+00

    Participating Particles 
    no        id stat      mothers     colours      p_x        p_y        p_z         e          m        tau    spin 
     1        21   -1     0     0   134   135      0.000      0.000    269.364    269.364      0.000    0.000   9.000
     2        21   -1     0     0   132   133      0.000      0.000   -581.812    581.812      0.000    0.000   9.000
     3         6    1     1     2   132     0     56.496   -140.910   -362.829    428.875    171.000    0.000   9.000
     4        -6    1     1     2     0   135    -44.780    155.920    176.682    294.574    171.000    0.000   9.000
     5        21    1     1     2   134   133    -11.717    -15.010   -126.300    127.728      0.000    0.000   9.000

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

Positive status —> method isFinal() returns true

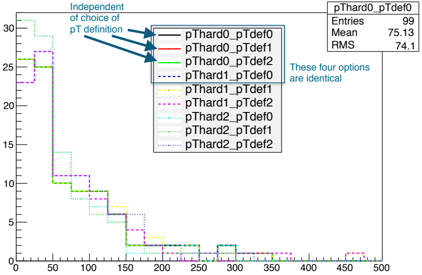
Mothers are the incoming partons at ME level

The key point: Pythia ME content is independent of whether Powheg lets top quarks decay or not! (This is good!) In $t\bar{t}$: always nFinal = 2

Hard scale in `main31`

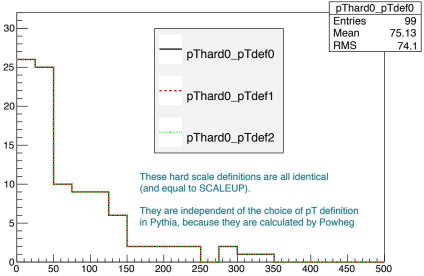
= scale above which emissions can be vetoed.
Internally called `pThard`.

Hard scale in `main31` (Pythia default Powheg LHE sample, 100 evts, $\sqrt{s}$ = 14 TeV, inclusive dec.)



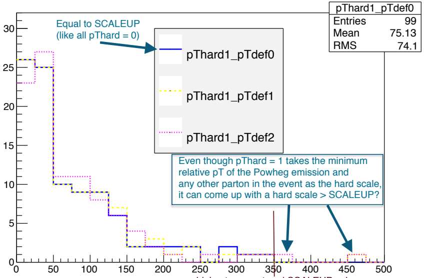
Independent of choice of pT definition

These four options are identical



These hard scale definitions are all identical (and equal to SCALEUP).

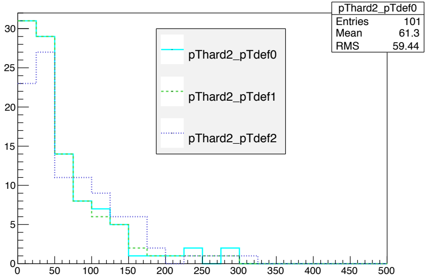
They are independent of the choice of pT definition in Pythia, because they are calculated by Powheg



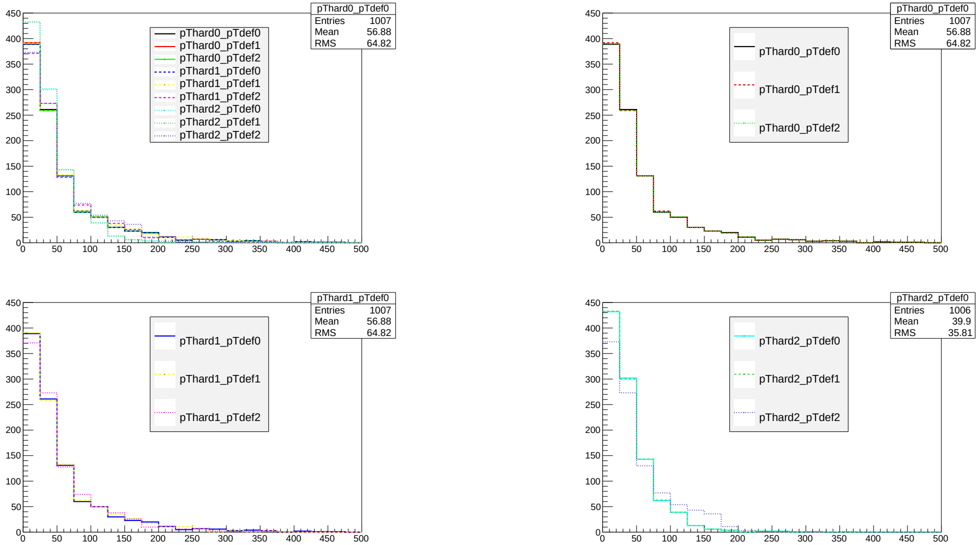
Equal to SCALEUP (like all pThard = 0)

Even though pThard = 1 takes the minimum relative pT of the Powheg emission and any other parton in the event as the hard scale, it can come up with a hard scale > SCALEUP?

highest encountered SCALEUP value



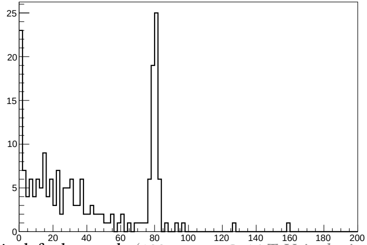
Hard scale in `main31` (My own Powheg LHE sample, 1000 evts, $\sqrt{s} = 7$ TeV, dileptonic)



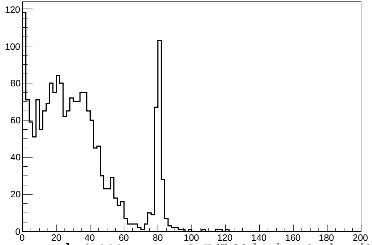
W boson invariant mass & charge in `main31`

Just checking if they take sensible values with and without vetoing.
Also good practice for digging into event records.

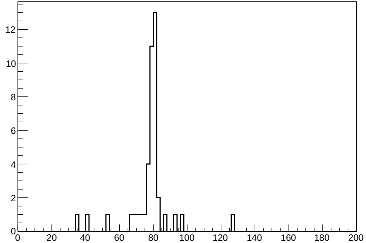
W boson invariant mass — it’s not trivial to find the right W decay products!



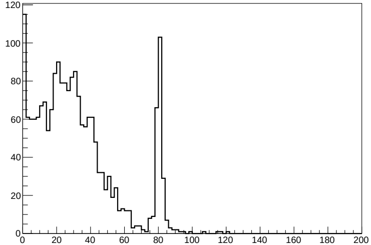
Pythia default sample (100 events @ 14 TeV, inclusive dec.)



My own sample (1000 events @ 7 TeV, hadronic dec. [incl. τ])

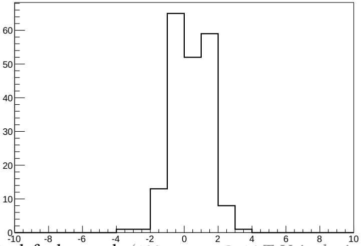


Pythia default sample, leptonic W decays only

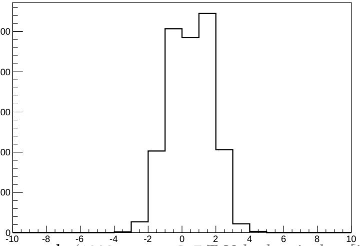


My own sample, no vetoing of emissions

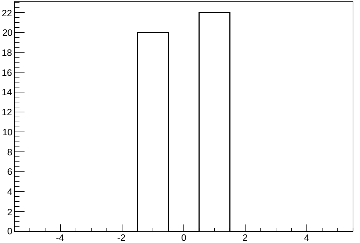
W boson charge — it’s not trivial to find the right W decay products!



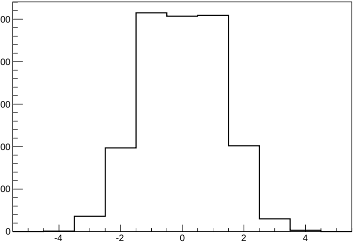
Pythia default sample (100 events @ 14 TeV, inclusive dec.)



My own sample (1000 events @ 7 TeV, hadronic dec. [incl. τ])



Pythia default sample, only leptonic W decays shown



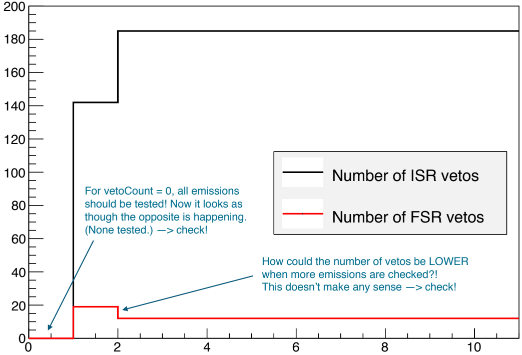
My own sample, no vetoing of emissions

The `vetoCount` parameter

= the number of consecutive accepted emissions
after which vetoing is switched off for the rest of the event.

The name `vetoCount` is confusing, since it does not count vetos but *accepted* emissions in a row!
Should be called “`acceptCount`” or something.

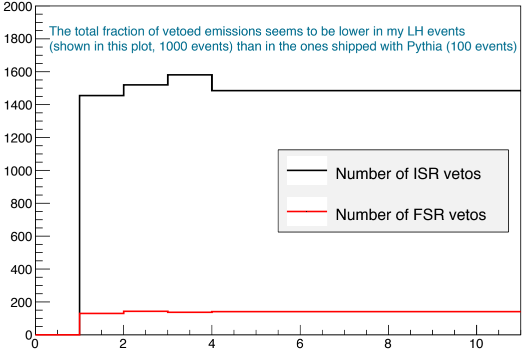
Number of consecutive accepted emissions after which vetoing is stopped



For vetoCount = 0, all emissions should be tested! Now it looks as though the opposite is happening. (None tested.) —> check!

How could the number of vetos be LOWER when more emissions are checked?! This doesn't make any sense —> check!

Pythia default sample (100 events @ 14 TeV, inclusive dec.)



The total fraction of vetoed emissions seems to be lower in my LH events (shown in this plot, 1000 events) than in the ones shipped with Pythia (100 events)

My own sample (1000 events @ 7 TeV, dileptonic decays)

Lower fraction of vetoed emissions in my sample not necessarily alarming: could be due to halved value of $\sqrt{s}$.

Rivet trouble update (~solved!)

I’ve installed Powheg-hvq, Pythia 8 and Rivet 2 on LXPlus.

My own Rivet installation works on LXPlus!

(Haven’t tested Powheg-hvq and Pythia 8 yet)

→ I will use LXPlus for event generation and Rivet analysis.

In the meantime, I’m communicating with Andy Buckley via email to find out what caused the problem on my Mac. Others might have the same issue on theirs, and I want to learn to use Valgrind for much more efficient debugging and tracing.

P.S.: Software versions used

Pythia 8.183

Powheg-hvq (Version? Most recent as of 27/02/2014)

LHAPDF 5.9.1