LUND UNIVERSITY

TeV4LHC Workshop
CERN
29 April 2005

(1) New PDG Particle Codes
(2) PYTHIA 6.3 Parton Showering*
(3) PYTHIA 8 Progress Report

Torbjörn Sjöstrand

CERN/PH and

Department of Theoretical Physics, Lund University

---

* reporting for P. Skands, T. Plehn, D. Rainwater

New PDG codes for Nuclei

$\boxed{\texttt{id = 10LZZZAAAI}}$

For a (hyper)nucleus consisting of n_p protons, n_n neutrons and n_Λ Λ^0's:

$A = n_p + n_n + n_\Lambda$ gives the total baryon number,

$Z = n_p$ the total charge, and

$L = n_\Lambda$ the total number of strange quarks.

I gives the isomer level, with $I = 0$ corresponding to the ground state and $I > 0$ to excitations.

Examples:

deuteron	1000010020
α	1000020040
^{235}U	1000922350

Warning: single hadrons, like p, n or Λ^0, are not changed.

(has been discussed & circulated, almost "cast in stone")

New PDG codes for R-hadrons

Prompted by split-SUSY interest, but intended more generically for long-lived colour triplets and octets (leptoquarks, extra dimensions, ...) which hadronize to give $\tilde{g}g$, $\tilde{g}q\overline{q}$, $\tilde{g}qqq$, $\tilde{q}\overline{q}$, $\tilde{q}qq$

Main principles:

- Put in the 1,000,000 and 2,000,000 normal SUSY series
- Enumerate the flavour content about as for normal mesons/baryons
- Let the squark/gluino flavour be the first one given
- For squark-mesons, use sign $+$ for squarks and $-$ for antisquarks
- Represent gluinos by a 9, like for gluons in glueballs
- The $2s+1$ digit is based only on spin of the light degrees of freedom (since the heavy spin decouples for $M \to \infty$)

Examples:

gluino-hadrons		squark-hadrons	
$\tilde{g}g$	1000993	$\tilde{u}_L\overline{s}$	1000232
$\tilde{g}u\overline{d}$	1009213	$\tilde{u}_R\overline{s}$	2000232
$\tilde{g}\overline{u}d$	-1009213	$\tilde{u}^*_R s$	-2000232
$\tilde{g}uud$	-1092212	$\tilde{c}_L uu$	1004221

(informal agreement)

PYTHIA 6.3 Showering in Transverse Momentum

1) *Define*
$$p_{\perp\mathrm{evol}}^2 = z(1-z)Q^2 = z(1-z)M^2 \text{ for FSR}$$
$$p_{\perp\mathrm{evol}}^2 = (1-z)Q^2 = (1-z)(-M^2) \text{ for ISR}$$

2) *Evolve* all partons *downwards* in $p_{\perp\mathrm{evol}}$ from common $p_{\perp\mathrm{max}}$

$$\mathrm{d}\mathcal{P}_a = \frac{\mathrm{d}p_{\perp\mathrm{evol}}^2}{p_{\perp\mathrm{evol}}^2} \frac{\alpha_s(p_{\perp\mathrm{evol}}^2)}{2\pi} P_{a\to bc}(z)\,\mathrm{d}z\,\exp\left(-\int_{p_{\perp\mathrm{evol}}^2}^{p_{\perp\mathrm{max}}^2} \cdots\right)$$

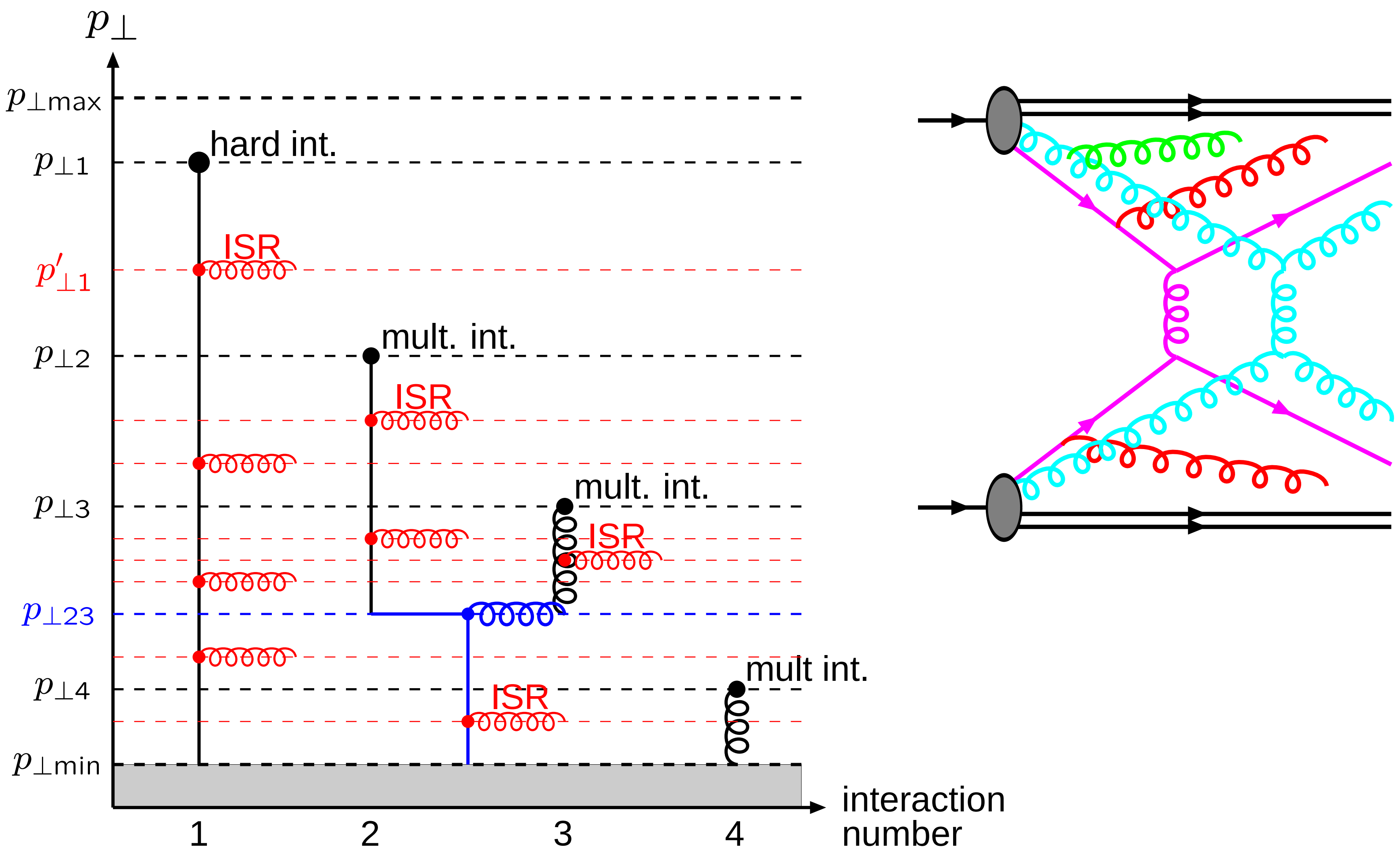
$$\mathrm{d}\mathcal{P}_b = \frac{\mathrm{d}p_{\perp\mathrm{evol}}^2}{p_{\perp\mathrm{evol}}^2} \frac{\alpha_s(p_{\perp\mathrm{evol}}^2)}{2\pi} \frac{x' f_a(x', p_{\perp\mathrm{evol}}^2)}{x f_b(x, p_{\perp\mathrm{evol}}^2)} P_{a\to bc}(z)\,\mathrm{d}z\,\exp\left(-\cdots\right)$$

Pick the one with *largest* $p_{\perp\mathrm{evol}}$ to undergo branching; also gives z.

3) Kinematics: *Derive* $Q^2 = \pm M^2$ by inversion of 1), but then interpret z as *energy fraction* (not lightcone) in “dipole” rest frame, so that *Lorentz invariant* and matched to matrix elements.
Assume yet unbranched partons on-shell and shuffle $(E, \mathbf{p})$ inside dipole.

4) *Iterate* $\Rightarrow$ combined sequence $p_{\perp\mathrm{max}} > p_{\perp 1} > p_{\perp 2} > \ldots > p_{\perp\mathrm{min}}$.

One Objective: Interleaved Multiple Interactions



Matrix Elements and Parton Showers

Complementary strengths:

- ME's good for well separated jets
- PS's good for structure inside jets

Marriage desirable! But how? Many problems!

Much work ongoing $\Longrightarrow$ no established orthodoxy

Three main areas, in ascending order of complication:

1) Match to lowest-order nontrivial process — merging

2) Combine leading-order multiparton process — vetoed parton showers

3) Match to next-to-leading order process — MC@NLO

. . . but let's not forget the "original" approach:

0) Improve the shower algorithm itself

(if doable then it gives fast results for "all" processes)

Shower Issues

1) Is $p_\perp^2$ a better evolution variable than $Q^2 = |M^2|$?

Sudakovs different even if phase space the same;

evolution in $p_\perp$ favours larger $p_\perp$'s.

2) What is appropriate maximum scale of evolution?

Conventional wisdom: evolve below "characteristic scale" of process,

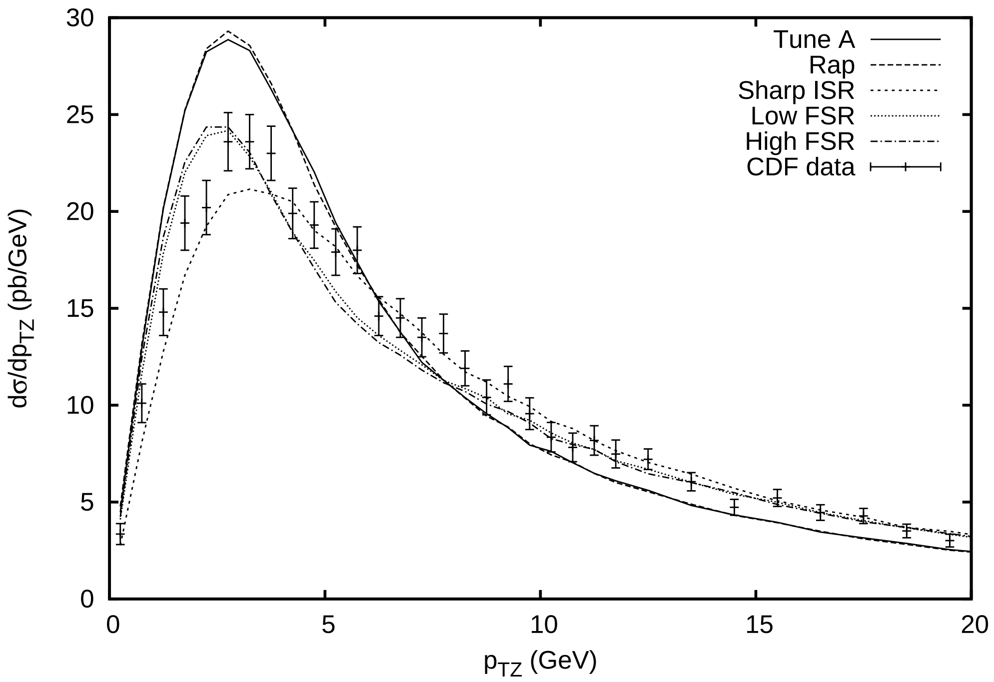
$\sim$ as for PDF scale choice, $\rightarrow f(\hat{s}, \hat{t}, m_i^2)$,

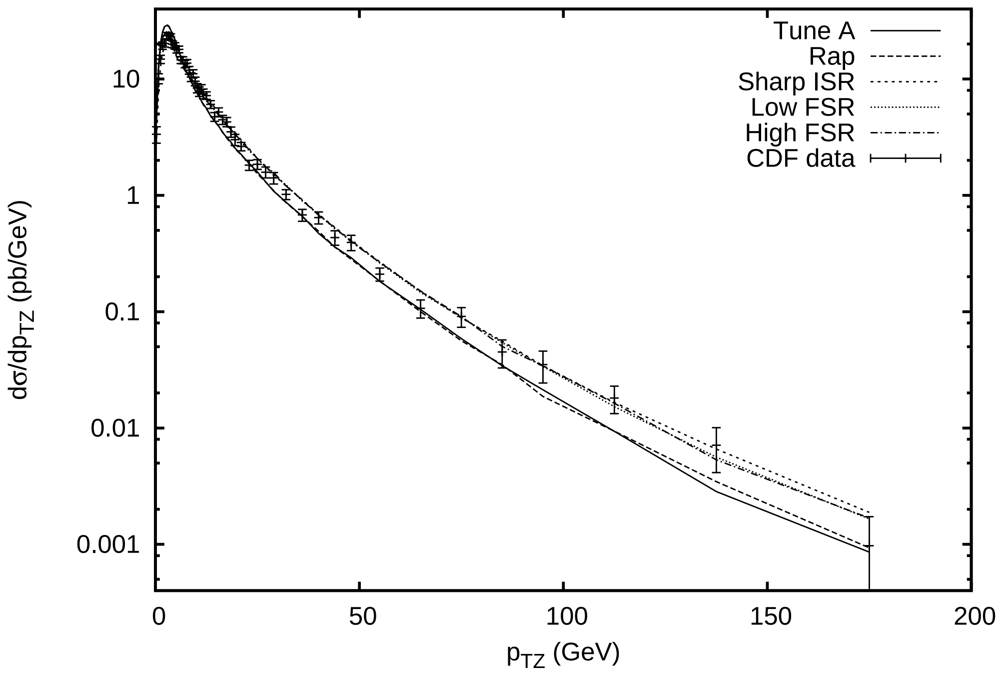
e.g. $Q^2_{\mathrm{max}} \approx \hat{s} \approx m_\mathrm{Z}^2$ for s-channel process like Z^0 production,

or $Q^2_{\mathrm{max}} \approx m_\perp^2 = m_\mathrm{t}^2 + p_\perp^2$ for $\mathrm{t\bar{t}}$ production.

But Z^0 experience: $Q^2_{\mathrm{max}} = s$ surprisingly good,

i.e. let shower populate *whole* phase space.





Test of Z^0 production at the Tevatron

max scale = s

Tune A = old
$Q^2 = |M^2|$ ordering

The others = new
$p_\perp^2$-ordering,
various variants of
MI/ISR/FSR matching
(see TS & P. Skands,
EPJ C39 (2005) 129)

Conclusion:
$p_\perp$ gives improvement,
but details matter

Shower Issues

1) Is $p_\perp^2$ a better evolution variable than $Q^2 = |M^2|$?

Sudakovs different even if phase space the same;

evolution in $p_\perp$ favours larger $p_\perp$'s.

2) What is appropriate maximum scale of evolution?

Conventional wisdom: evolve below "characteristic scale" of process,

$\sim$ as for PDF scale choice, $\to f(\hat{s}, \hat{t}, m_i^2)$,

e.g. $Q^2_{\mathrm{max}} \approx \hat{s} \approx m_{\mathrm{Z}}^2$ for s-channel process like Z^0 production,

or $Q^2_{\mathrm{max}} \approx m_\perp^2 = m_{\mathrm{t}}^2 + p_\perp^2$ for $\mathrm{t\bar{t}}$ production.

But Z^0 experience: $Q^2_{\mathrm{max}} = s$ surprisingly good,

i.e. let shower populate *whole* phase space.

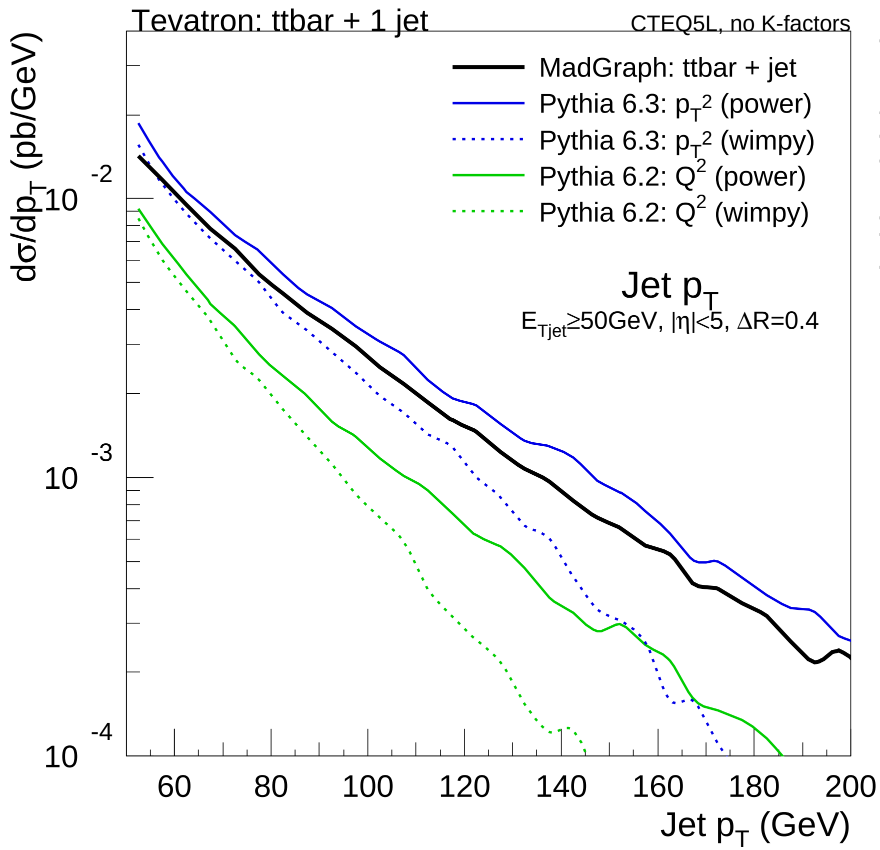
So study $\mathrm{t\bar{t}}$ production and compare

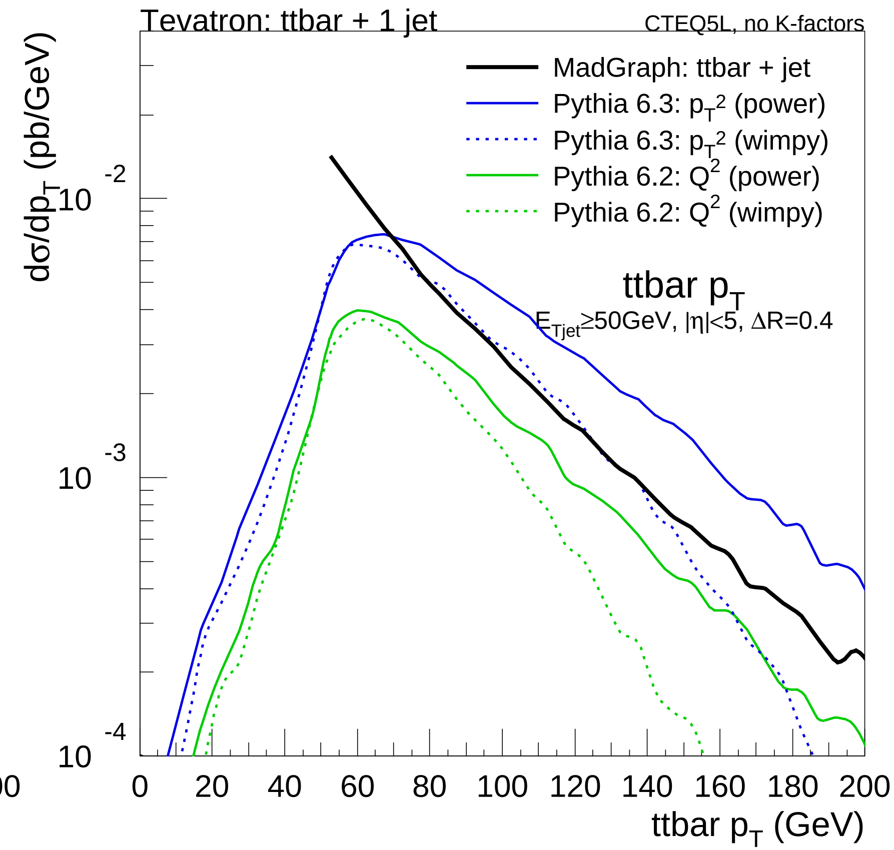
- PYTHIA old Q^2-ordered, $Q^2_{\mathrm{max}} = m_\perp^2$

  (PYTHIA default is $Q^2_{\mathrm{max}} = 4m_\perp^2$)
- PYTHIA old Q^2-ordered, $Q^2_{\mathrm{max}} = s$
- PYTHIA new $p_\perp^2$-ordered, $Q^2_{\mathrm{max}} = m_\perp^2$
- PYTHIA new $p_\perp^2$-ordered, $Q^2_{\mathrm{max}} = s$

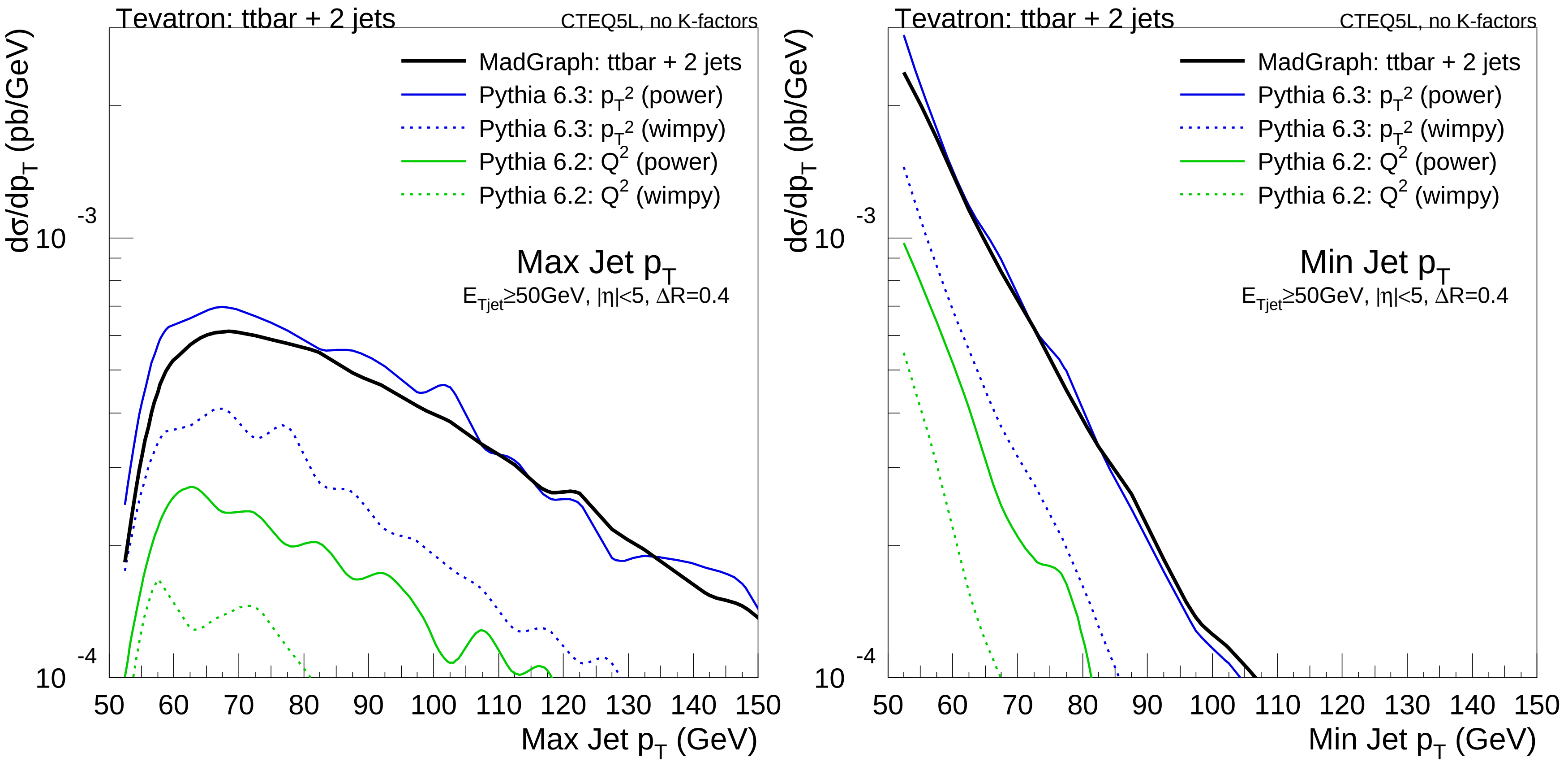
Current test: $t\bar{t}$ production at the Tevatron

Plots and shower studies by P. Skands
MadGraph ME calculations by T. Plehn & D. Rainwater
(publication in preparation)

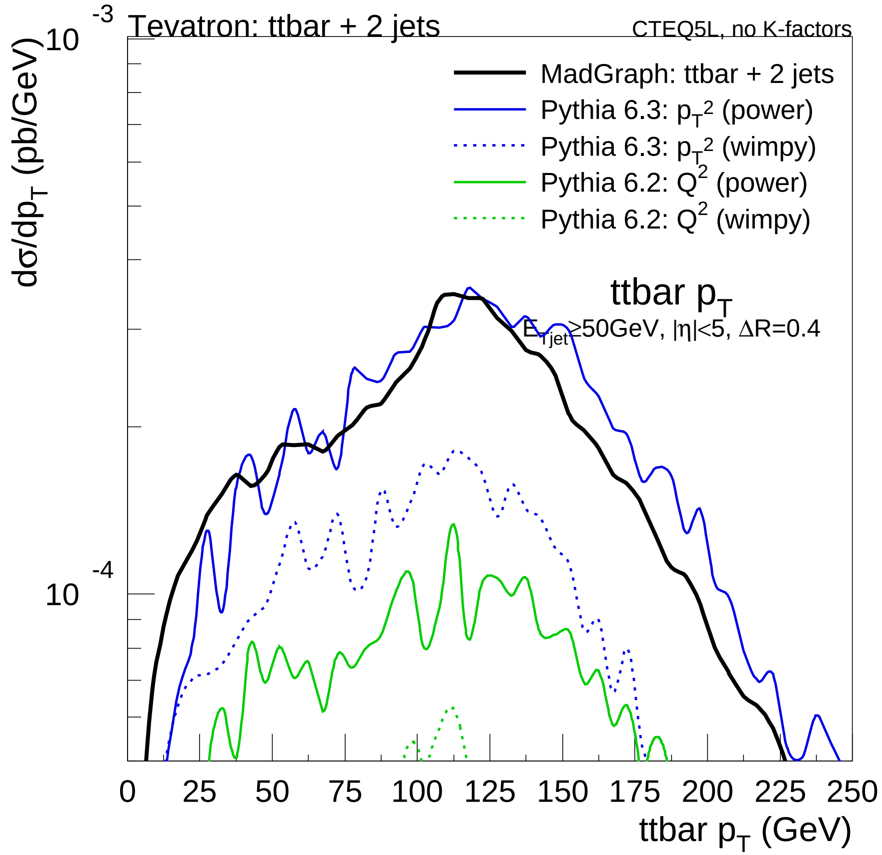




Parton-level only, no underlying event, no top decays
$E_{\perp \text{jet}} > 50$ GeV, $\Delta R = 0.4$



Tevatron: ttbar + 2 jets
CTEQ5L, no K-factors
MadGraph: ttbar + 2 jets
Pythia 6.3: p_T^2 (power)
Pythia 6.3: p_T^2 (wimpy)
Pythia 6.2: Q^2 (power)
Pythia 6.2: Q^2 (wimpy)
Max Jet p_T
$E_{Tjet} \geq 50$GeV, $|\eta|<5$, $\Delta R=0.4$
$d\sigma/dp_T$ (pb/GeV)
Max Jet p_T (GeV)
Min Jet p_T
Min Jet p_T (GeV)



Conclusions:

Transverse-momentum-ordered showers
with maximal starting scale does amazingly well!

Bodes well for "blind" application of the new showers, either

- to processes not yet studied with more "sophisticated" methods
- to *further* emissions when hardest given by matrix elements

On To C++

Currently HERWIG and PYTHIA are successfully being used,
also in new LHC environments, using C++ wrappers

Q: Why rewrite?
A1: Need to clean up!
A2: Fortran 77 is limiting **Fortran 90**
A3: Young experimentalists will expect C++

PYTHIA7 project $\Longrightarrow$ **ThePEG**
Toolkit for High Energy Physics Event Generation
(L. Lönnblad; S. Gieseke, A. Ribon, P. Richardson)

HERWIG++: complete reimplementation
(B.R. Webber; S. Gieseke, A. Ribon, P. Richardson, M. Seymour, P. Stephens, 3 new)

ARIADNE/LDC: to do ISR/FSR showers, multiple interactions
(L. Lönnblad; N. Lavesson)

SHERPA: in C++ from start, partly wrappers to PYTHIA Fortran
(F. Krauss; T. Gleisberg, S. Hoeche, A. Schaelicke, S. Schumann, J. Winter)

PYTHIA8: A fresh start

Problem: PYTHIA7 stalled, no other manpower

Solution?: take a sabbatical and work “full-time”!

($\Rightarrow$ baseline model, S. Mrenna & P. Skands join later ?)

Tentative schedule:

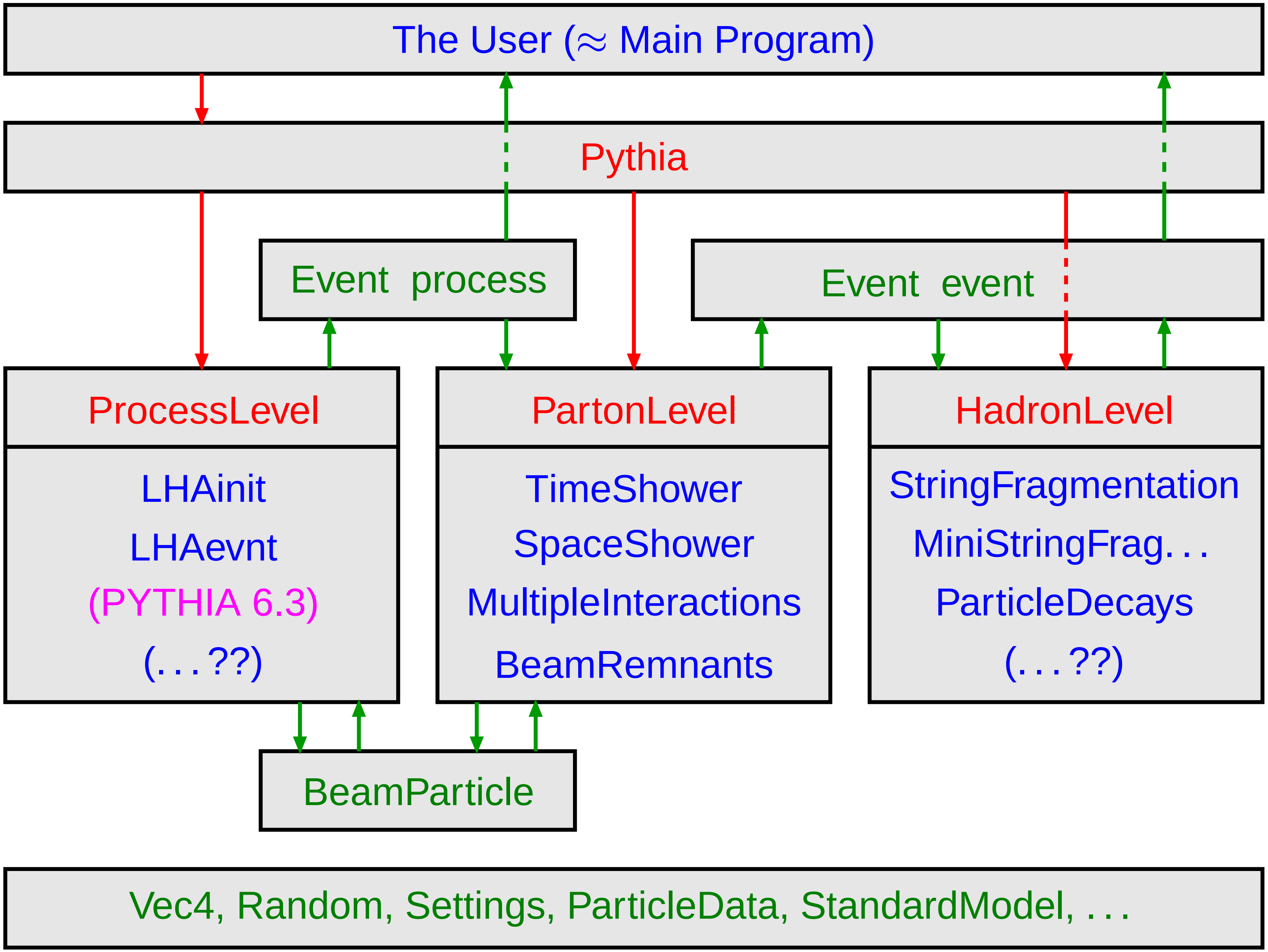
time	date	processes	final states
0 =	1 Sept. 2004	—	—
1 =	1 Sept. 2005	LHA-style input	incomplete draft
2 =	1 Sept. 2006	a few processes	complete, buggy(?)
3 =	1 Sept. 2007	more processes	stable, debugged

. . . but don’t forget Murphy’s law

Objectives:

- clean up, keep the most recent models
- Les Houches Accord style input central
- independent of ThePEG (or anything else), but
- interface to ThePEG later written by Leif (?)

Current PYTHIA8 structure



Current PYTHIA8 status

Existing classes			Missing classes/topics
Process Level	LHAinit	$\star$	ThePEG input, alternatively Cross section administration Phase space selection Process matrix elements Parton density libraries Resonance decays
	LHAevnt	$\star\star$	
	(PYTHIA 6.3)	$\star\star\star$	
Parton Level	TimeShower	$\star\star$	MI/ISR/FSR interleaving colour flow models ME/PS matching
	SpaceShower	$\star\star$	
	MultipleInteractions	$\star$	
	BeamRemnants	$\star$	
Hadron Level	StringFragmentation	$\star$	Junction fragmentation Popcorn baryons updated decay tables Bose-Einstein
	MiniStringFrag. . .	$\star$	
	ParticleDecays	$\star$	
—	Event	$\star\star$	event analysis routines
	BeamParticle	$\star\star$	. . . and much, much more
	Vec4, Random	$\star\star\star$	
	Settings	$\star\star$	
	ParticleData	$\star$	

$\Longrightarrow$ Roughly according to three-year plan so far!

First public “proof-of-concept” version by GENSER July meeting (!?)

Event generation structure

1) Initialization step

- select process(es) to study
- modify physics parameters
- set kinematics constraints
- modify generator settings
- initialize generator
- book histograms

```
#include "Pythia.h"
using namespace Pythia8;
Pythia pythia;
pythia.readLine("command");
pythia.readFile("command.file");
pythia.init(idBeamA,idBeamB,eCM);
```

2) Generation loop

- generate one event at a time
- analyze it (or store for later)
- add results to histograms
- print a few events

```
pythia.next();
pythia.process.list();
pythia.event.list();
int id = pythia.event[i].id();
```

3) Finishing step

- print deduced cross-sections
- print/save histograms etc.

```
pythia.statistics();
pythia.settings.listChanged();
```

Sample input cards

```
! This file contains commands to be read in for a Pythia8 run.
! Lines not beginning with a letter are comments.

! 1) Settings that could be used in a main program, if desired.
Main:idBeamA = 2212                  ! first beam, p = 2212, pbar = -2212
Main:idBeamB = 2212                  ! second beam, p = 2212, pbar = -2212
Main:eCM = 14000.                    ! CM energy of collision
Main:numberOfEvents = 1000           ! number of events to generate
Main:numberToPrint = 2               ! number of events to print
Main:numberToShow = 50               ! show how far along run is
Main:showChangedSettings = on        ! print changed flags/modes/parameters
Main:showAllSettings = off           ! print all flags/modes/parameters

! 2) Settings for the hard-process generation.
! Based on an interface to the Fortran Pythia6 program.
#Pythia6:msel = 1                     ! QCD production
#Pythia6:ckin(3) = 100.               ! pTmin cut
Pythia6:msel = 6                     ! t tbar production

! 3) Settings for the event generation process in the Pythia8 library.
#PartonLevel:MI = off                 ! no multiple interactions
#PartonLevel:ISR = off                ! no initial-state radiation
PartonLevel:FSR = off                ! no final-state radiation
#HadronLevel:Hadronize = off          ! no hadronization
SpaceShower:pT0 = 2.0                ! dampening of pT -> 0 divergence
MultipleInteractions:pTmin = 3.0     ! lower pT cutoff for interactions
```

Sample output from run

```
 --------  Pythia Flag + Mode + Parameter Settings (changes only)  ----------------

   Kind  Name                                      Now    Default        Min        Max
 double  Main:eCM                             1.40e+04 2000.0000     0.0000   1.00e+05
 double  MultipleInteractions:pTmin             3.0000     2.0000     0.5000    10.0000
   bool  PartonLevel:FSR                           off         on
 double  SpaceShower:pT0                        2.0000     0.5000     0.0000    10.0000

 --------  End Pythia Flag + Mode + Parameter Settings  ---------------------------

 --------  Pythia Event Listing  (hard process)  ----------------------------------

    no        id   name          status     mothers   daughters     colours        p_x
     0        90   (system)         -11     0     0     0     0     0     0      0.000
     1      2212   (p+)             -12     0     0     3     0     0     0      0.000
     2      2212   (p+)             -12     0     0     4     0     0     0      0.000
     3        21   (g)              -21     1     0     5     6   101   102      0.000
     4        21   (g)              -21     2     0     5     6   103   101      0.000
     5        -6   (tbar)           -22     3     4     7     8     0   102   -107.572
     6         6   (t)              -22     3     4     9    10   103     0    107.572
     7       -24   (W-)             -22     5     0    11    12     0     0    -71.772
     8        -5   bbar              23     5     0     0     0     0   102    -35.799
     9        24   (W+)             -22     6     0    13    14     0     0    113.539
    10         5   b                 23     6     0     0     0   103     0     -5.968
    11        11   e-                23     7     0     0     0     0     0    -38.516
    12       -12   nu_ebar           23     7     0     0     0     0     0    -33.256
    13        -1   dbar              23     9     0     0     0     0   104     24.321
    14         2   u                 23     9     0     0     0   104     0     89.218
                                                                     Sum:     -0.000
 --------  End Pythia Event Listing  ----------------------------------------------
```

```
--------  Pythia Event Listing  (complete event)  ---------------------------------------------------------------------------------

   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   187     0     0     0      0.000      0.000   7000.000   7000.000      0.938
    2      2212   (p+)               -12     0     0   188     0     0     0      0.000      0.000  -7000.000   7000.000      0.938
    3        21   (g)                -21     7     0     5     6   101   102      0.000      0.000     53.792     53.792      0.000
    4        21   (g)                -21     8     8     5     6   103   101      0.000      0.000   -829.022    829.022      0.000
    5        -6   (tbar)             -22     3     4     9     9     0   102   -107.572    -45.614   -345.827    404.638    174.595
    6         6   (t)                -22     3     4    10    10   103     0    107.572     45.614   -429.402    478.176    174.969
    7        21   (g)                -41    12    12    11     3   105   102     -0.000     -0.000     76.351     76.351      0.000
    8        21   (g)                -42    13     0     4     4   103   101     -0.000      0.000   -829.022    829.022      0.000
    9        -6   (tbar)             -44     5     5    14    14     0   102   -127.853    -17.612   -332.165    396.829    174.595
   10         6   (t)                -44     6     6    15    15   103     0     90.752     68.837   -379.579    433.208    174.969
   11        21   (g)                -43     7     0    16    16   105   101     37.101    -51.226    -40.927     75.336      0.000
  (skipped)
   63        21   (g)                -31   111     0    65    66   112   111      0.000      0.000      0.070      0.070      0.000
   64        -4   (cbar)             -31   112   112    65    66     0   110      0.000      0.000   -926.957    926.957      0.000
   65        21   (g)                -33    63    64   113   113   112   110      5.011     -0.788   -104.687    104.810      0.000
   66        -4   (cbar)             -33    63    64   114   114     0   111     -5.011      0.788   -822.200    822.217      1.500
  (skipped)
  237      2101   (ud_0)             -63     1     0     0     0     0   137      0.240     -0.007   3177.306   3177.306      0.579
  238        -1   (dbar)             -63     1     0     0     0     0   124      1.153     -0.432    839.002    839.003      0.330
  239      2101   (ud_0)             -63     2     0     0     0     0   142     -1.091      0.128  -2613.733   2613.733      0.579
  240         4   (c)                -63     2     0     0     0   142     0     -0.557      1.321   -174.031    174.043      1.500
  (skipped)
  241       -24   (W-)               -22   195     0   245   245     0     0   -102.292    -46.372   -349.729    376.307     81.747
  242        -5   (bbar)             -23   195     0   243   244     0   102    -39.504     23.812     -8.300     47.111      4.800
  243        -5   (bbar)             -51   242     0   248   248     0   144    -26.921     15.510     -8.835     32.656      4.800
  244        21   (g)                -51   242     0   246   247   144   102    -12.740      8.184     -0.143     15.143      0.000
  245       -24   (W-)               -52   241   241   263   264     0     0   -102.135    -46.255   -349.051    375.619     81.747
  (skipped)
  263        11   (e-)               -23   245     0   265   266     0     0    -49.476     20.517   -126.258    137.149      0.001
  264       -12   (nu_ebar)          -23   245     0   267   267     0     0    -52.659    -66.772   -222.793    238.470      0.000
  265        11   e-                  51   263     0     0     0     0     0    -48.966     20.308   -124.957    135.736      0.001
  266        22   gamma               51   263     0     0     0     0     0     -0.510      0.210     -1.301      1.413      0.000
  267       -12   nu_ebar             52   264   264     0     0     0     0    -52.659    -66.772   -222.793    238.470      0.000
  (skipped)
  285       323   K*+                 73   247     0     0     0     0     0     -8.774      4.484     -1.202      9.966      0.892
  286       533   B*_s0               73   248     0     0     0     0     0    -24.787     14.045     -6.657     29.754      5.416
  287       423   D*0                 73   240     0     0     0     0     0     -0.604      1.434   -307.590    307.600      2.007
  288       223   omega               73   240     0     0     0     0     0     -0.097     -0.243   -316.742    316.743      0.782
  289       113   rho0                73   239     0     0     0     0     0     -0.424     -0.021   -525.177    525.178      0.768
  290      2212   p+                  73   239     0     0     0     0     0     -0.522      0.279  -1638.254   1638.254      0.938
  (skipped)
  490       223   omega               73   237     0     0     0     0     0      0.481     -0.049    154.560    154.563      0.782
  491      2212   p+                  73   237     0     0     0     0     0     -0.269     -0.100   2588.971   2588.972      0.938
                                                                   Sum:    -0.000     -0.000     -0.000  14000.000  14000.000
--------  End Pythia Event Listing  -----------------------------------------------------------------------------------------------
```

Sample run with Les Houches input

```
#include "Pythia.h"
using namespace Pythia8;
int main() {

  int nPrint = 2;                                 // Number of events to print.
  Pythia pythia;                                  // Generator.
  pythia.readLine("PartonLevel:MI = off");        // No multiple interactions.
  pythia.readLine("SpaceShower:pTmin = 1.0");     // Change pTmin cutoff of ISR.
  LHAinitPythia6 lhaInit("sample.init");          // Les Houches initialization object.
  LHAevntPythia6 lhaEvnt("sample.evnt");          // Les Houches event object.
  pythia.init(&lhaInit, &lhaEvnt);                // Initialize with pointers.
  cout << lhaInit;                                // List initialization information.
  Hist nFinal("final particle multiplicity",100,-0.5,499.5);        // Histogram.

  int iEvent = 0;                                 // Begin event loop
  while (pythia.next()) {                         // Generate event until none left.
    if (iEvent++ < nPrint) {                      // List first few events.
      cout << lhaEvnt;                            // List Les Houches input event.
      pythia.process.list();                      // List Pythia hard-process event.
      pythia.event.list();                        // List Pythia complete event.
    }                                             // End listing.
    int nFin = 0;                                 // Sum up final multiplicity
    for (int i = 0; i < pythia.event.size(); ++i)
      if (pythia.event[i].remains()) nFin++;
    nFinal.fill(nFin);                            // Fill histogram.
  }                                               // End of event loop.

  cout << nFinal;                                 // Print histogram.
  return 0;                                       // Done.
}
```