A New h-> ng Parton Shower

M. E. Peskin
November 2008

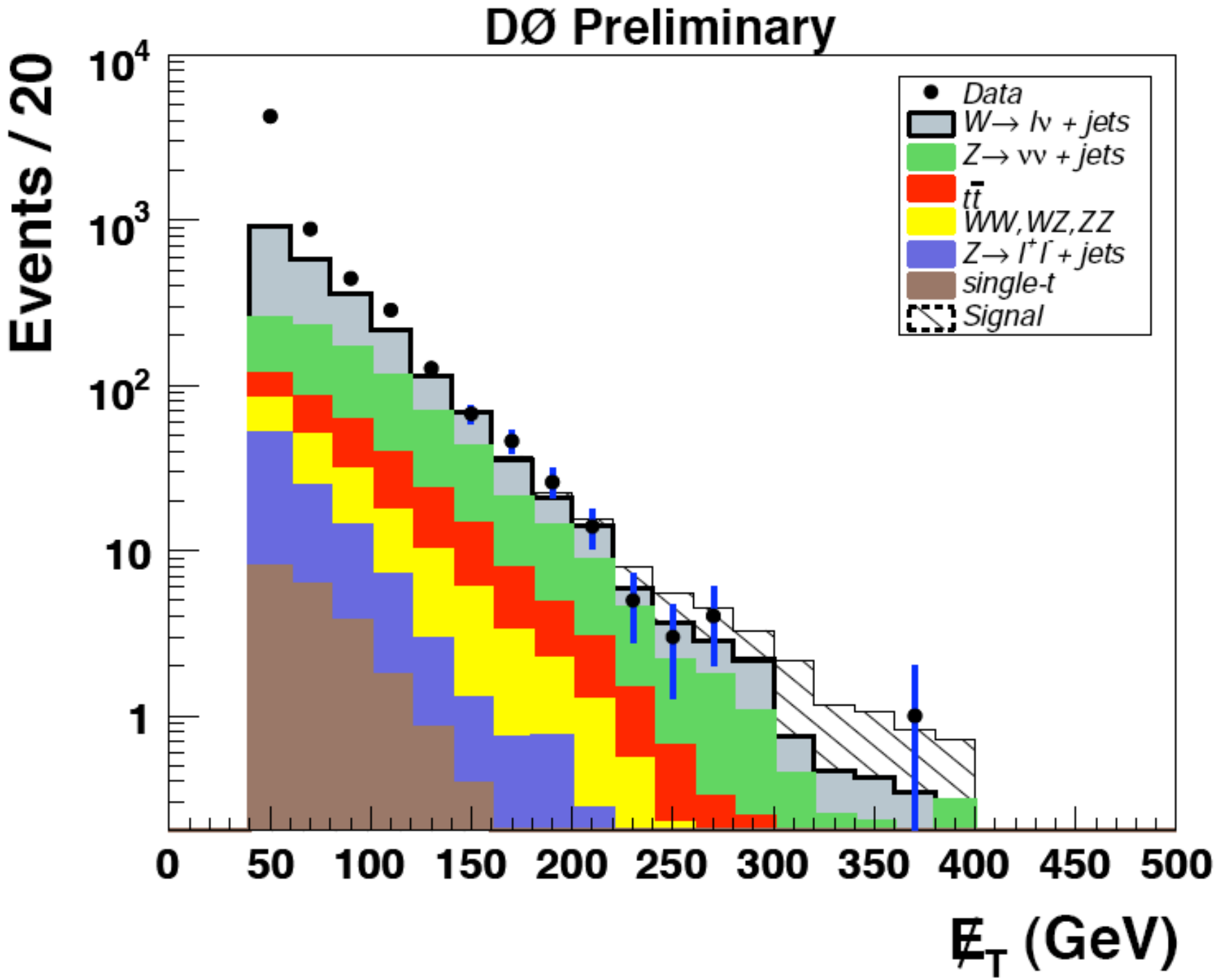
Today, many people are interesting in methods for merging partons showers at low pT with exact QCD calculations for the hardest jets.

Much of this development is driven by the problems of the Tevatron and the LHC. Our ability to understand physics at hadron colliders is limited by our understanding of the shapes of Standard Model backgrounds.

New physics appears at large values of

missing energy, HT, number of jets, ...

These are precisely the regions treated poorly by standard parton shower algorithms.


DØ Preliminary
Events / 20
10^4
10^3
10^2
10
1
0
50
100
150
200
250
300
350
400
450
500
$\not\!\!E_T$ (GeV)
Data
$W\rightarrow l\nu$ + jets
$Z\rightarrow \nu\nu$ + jets
$t\bar{t}$
WW,WZ,ZZ
$Z\rightarrow l^+l^-$ + jets
single-t
Signal

There are many approaches to this problem with somewhat different ambitions. All are versions of "matrix element/ parton shower" matching:

correction of high-order matrix element calculations for consistency with subsequent parton showering

ALPGEN, MADEVENT, SHERPA, HELAC
Catani-Krauss-Kuhn-Webber

correction of parton showers to incorporate exact 1-loop calculations

MC@NLO Frixione, Webber, Nason

improvement of parton shower algorithms using QCD resummation in the soft and collinear regions

Becher + Schwartz

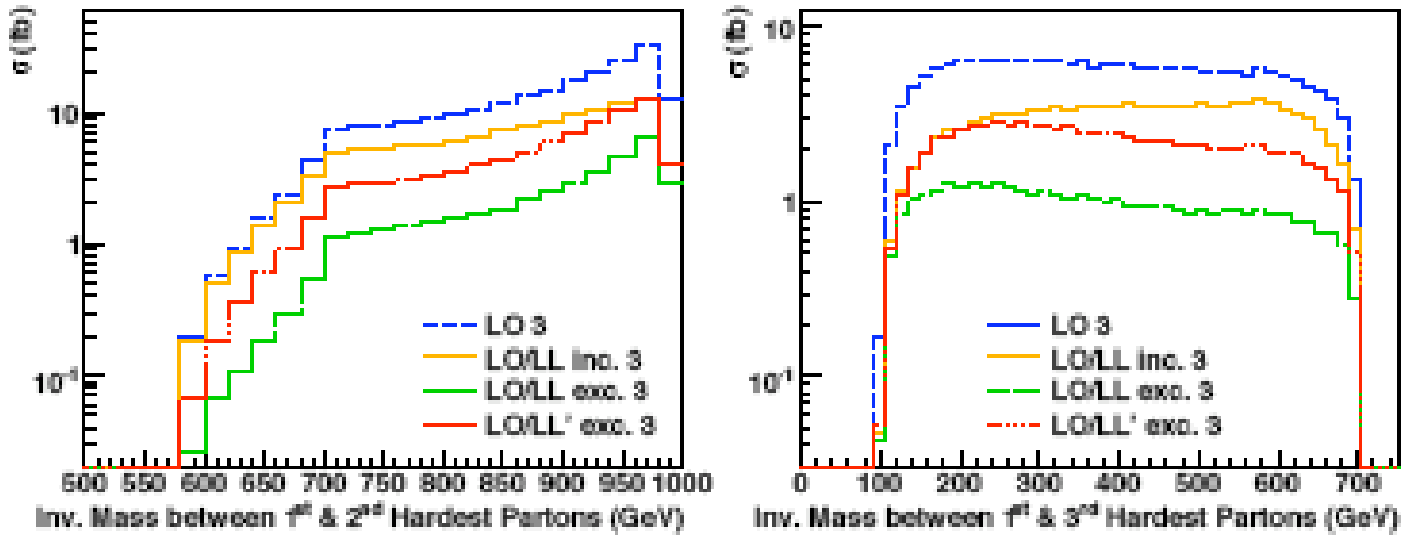
This is the version of the matching problem that I will be concerned with here:

Can one write a parton shower algorithm that can systematically incorporate QCD tree amplitudes,

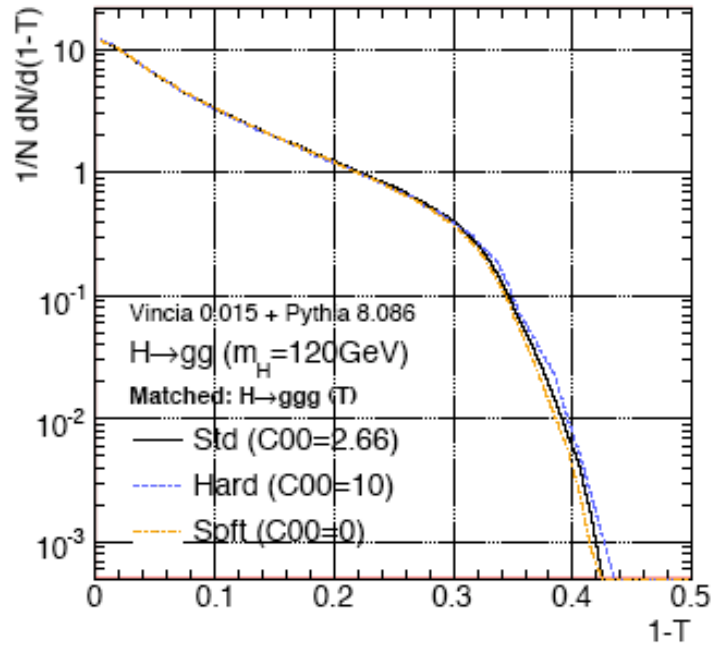
so that n-jet emission automatically has the shape that those amplitudes predict?

I will not attempt to get the normalization of cross sections right. That requires loop calculations. I will try to efficiently generate weight-1 events. For speed of computation, my "exact" amplitudes will be leading-color only.

Bauer, Tackmann, and Thaler have attacked this same problem, and are much further along: (GenEVa)



Giele, Kosower, and Skands also have a new, more systematic approach to parton showers (VINCIA).



My approach is built on the answers to three questions:

1. How can we rapidly produce exact QCD amplitudes ?

2. These amplitudes refer to points in n-particle phase space. How do we parametrize exact phase space so that it looks like a parton shower ?

3. The use of these amplitudes requires reweighting and acceptance/rejection in a parton shower. How is this done ?

In this talk, I will discuss the shower in $h^0 \longrightarrow ng$.

To generate tree amplitudes, I use the Britto-Cachazo-Feng recursion formula for on-shell amplitudes:

$$i\mathcal{M}(1\cdots n) \quad = \quad \sum_{splits} i\mathcal{M}(b+1\cdots\hat{i}\cdots a-1\,-\hat{Q}) \cdot \frac{1}{s_{a\cdots b}} \cdot i\mathcal{M}(a\cdots\hat{j}\cdots b\,\hat{Q})$$

I am content with amplitudes at the leading order in Nc.

The BCF formula recursively breaks amplitudes down (numerically, on the fly) to the MHV result (Dixon, Glover, Khoze)

$$i\mathcal{M}(h^0 \to g_1^+ g_2^+ \cdots g_n^+) = ig^{n-2} C \frac{m_h^4}{\langle 12\rangle\langle 23\rangle\cdots\langle N1\rangle}$$

and the conjugate, with all g(-).

Duhr, Hoche, and Maltoni and Dinsdale, Ternick, and Weinzierl (DTW) have investigated how to do this. The latter group has written an especially fast code for multigluon amplitudes. Both groups conclude that, if you are sophisticated, BCF recursion has no advantage over the more venerable Berends-Giele recursion. However, BCF recursion can be implemented to give fast computations with a very simple code.

Our code is not as fast as Weinzierl's, but we can compute h-> 8 g amplitudes in 0.1 msec. That is quite fast enough.

Some fun C++ programming is involved. In our approach, we use as the basic object a C++ class called a bispinor. This holds 8 complex numbers

$$\begin{array}{rcll} p^{\pm} & = & p^0 \pm p^1 & q^{\pm} = p^2 \pm ip^3 \\ p\rangle & = & u_R(p) & p] = u_L(p) \end{array}$$

and implements methods that derive the vector components from the spinor components and vice versa.

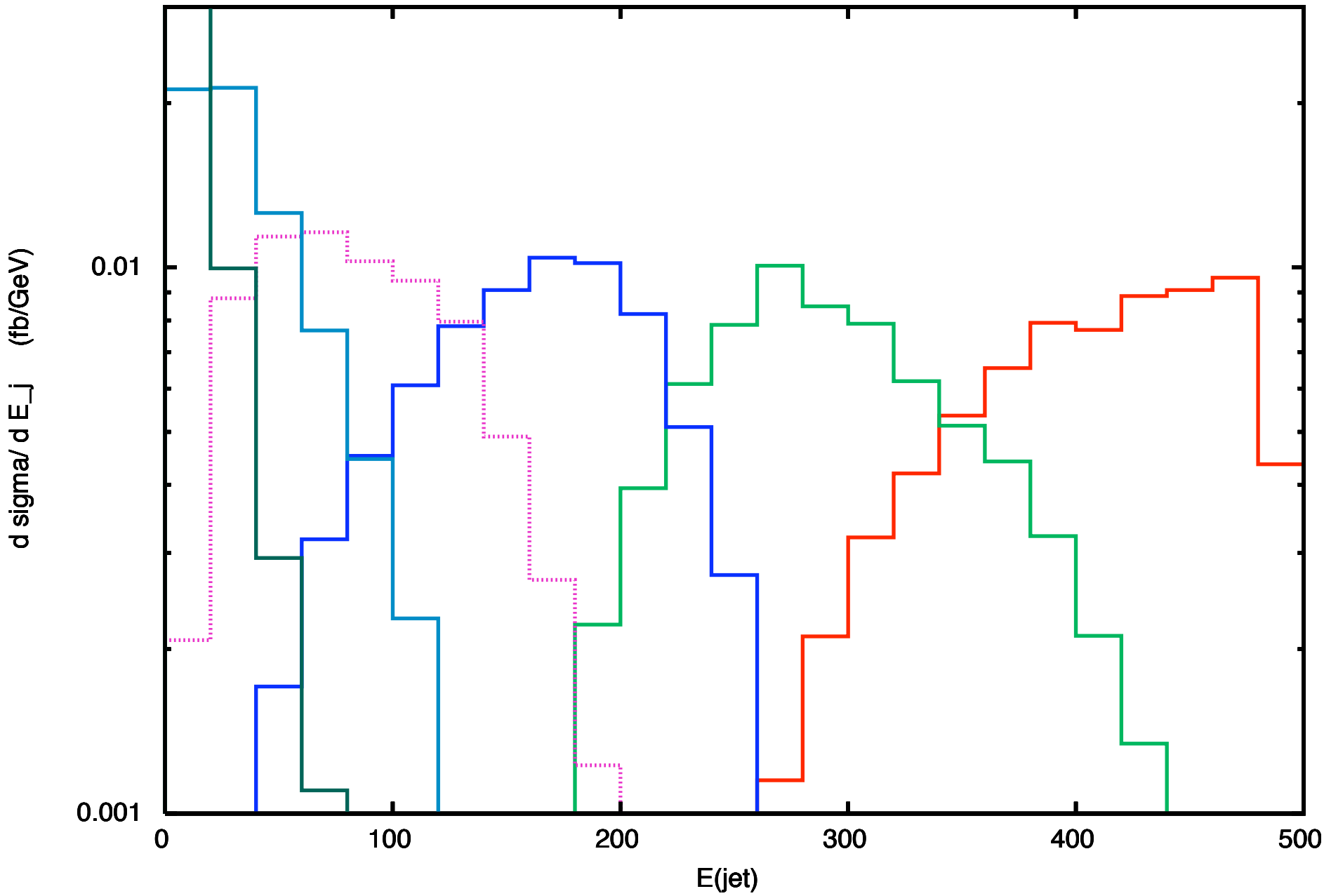
It is important not to lose phase information in converting vectors to bispinors. Having obtained u_R^1 with some phase, we use

$$u_L^1 = \frac{q^-}{u_R^1} \quad u_L^2 = \frac{p^+}{u_R^1} \quad u_R^2 = \frac{q^+}{p^+} u_R^1$$

Then the composite object $p\rangle[p$ preserves the original phases.

DTW emphasize the importance of memory management. So, actually, we replace the bispinor class by a cone class that holds all of the bispinors needed to compute the desired amplitude.

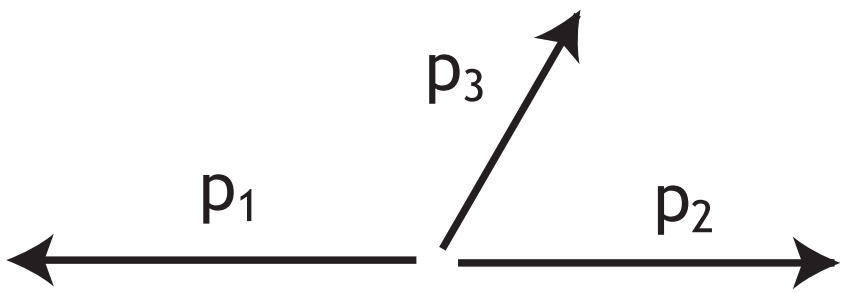
parton energies in $e^+e^- \to q + 4g + \bar{q}$



Once we have the amplitudes, we need to integrate them over phase space. To do this, we need to efficiently generate multiparticle phase space, enhanced in the region where the QCD denominators are large.

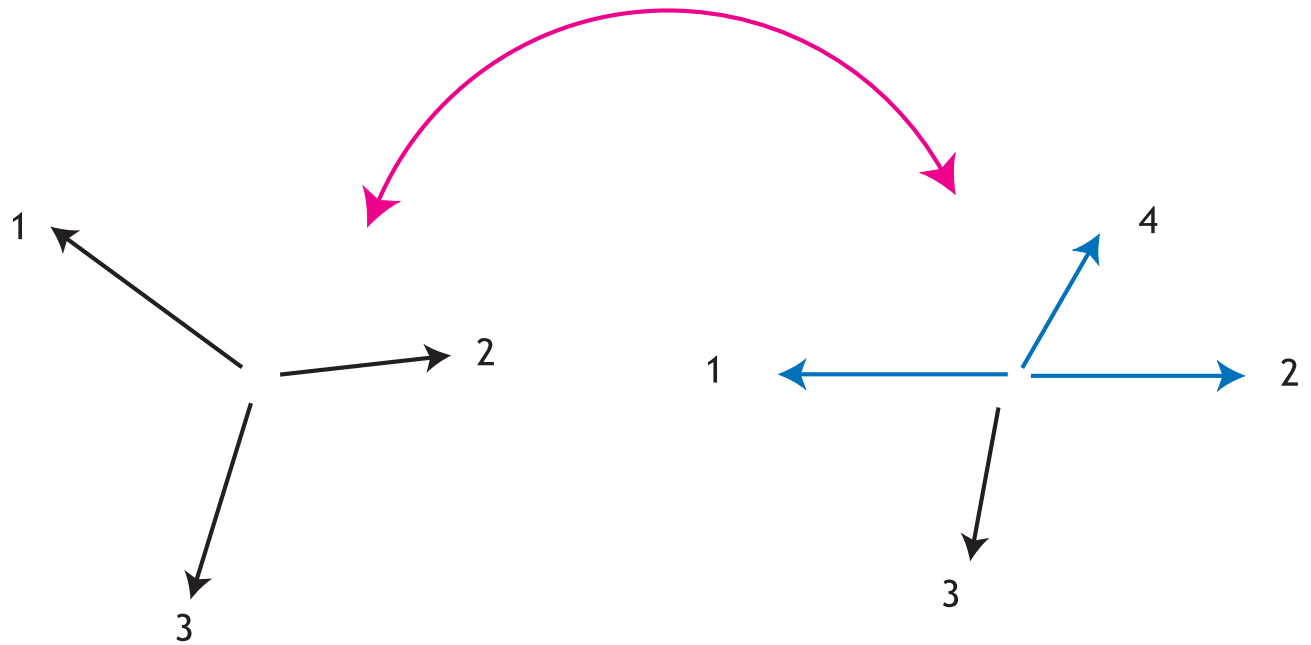
An effective trick has been introduced by Draggiotis, van Hameren, and Kleiss as the basis of their SARGE algorithm

Start with two back-to-back lightlike vectors. Add a third lightlike vector $p_3 = \xi_1 p_1 + \xi_2 p_2 + p_\perp$



Then boost and rescale to the original CM frame and energy.

To add the fourth vector, pick two neighbors, boost these back-to-back, add a vector as before, and then boost the entire system back to the CM frame.



Effectively, the entire event recoils when a new vector is added.

The logarthmic integral over the parameters reproduces massless phase space

$$\int \frac{d^3p_3}{(2\pi)^2 2p_3} \frac{2p_1\cdot p_2}{2p_1\cdot p_3\, 2p_3\cdot p_2} = \frac{1}{(4\pi)^2}\int\frac{d\xi_1}{\xi_1}\int\frac{d\xi_2}{\xi_2}\int\frac{d\phi}{2\pi}$$

Applying this operation repeatedly, we build up phase space with all of the QCD denominators of the color-ordered amplitude for emission of final-state radiation.

$$\int d\Pi_n \frac{1}{2p_1\cdot p_2\, 2p_2\cdot p_3\, \cdots 2p_n\cdot p_1} = \frac{1}{8\pi Q^4}\prod_i\left[\frac{1}{(4\pi)^2}\int\frac{d\xi_{1i}}{\xi_{1i}}\int\frac{d\xi_{2i}}{\xi_{2i}}\int\frac{d\phi}{2\pi}\right]$$

This is an exact formula for massless phase space with QCD denominators, but only if we integrate over every point in phase space exactly once.

Draggiotis, van Hameren, and Kleiss suggested adding the vectors 1, 2, 3 in fixed (color) order. This requires very large values for the ξ_i to reproduce some phase space configurations.

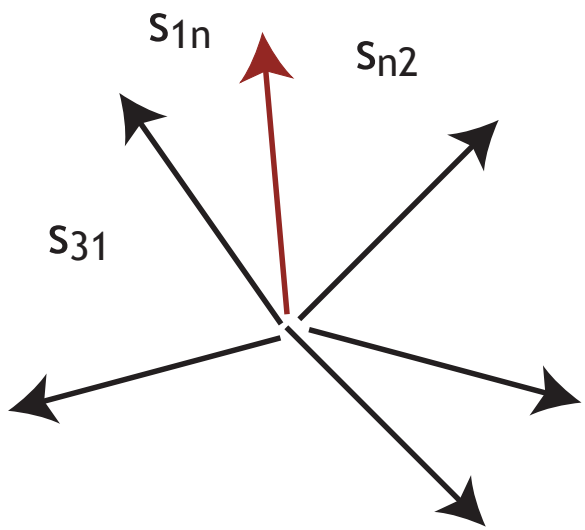
An alternative approach is to choose arbitrarily at each step one interval in which to insert a new vector. We call the set of such choices a chamber. It is then necessary to define the limits of each chamber so that the full set of chambers tiles phase space.

Here is a useful definition of a chamber:

Let the nth vector be inserted between 1 and 2. Then allow all values of ξ_1, ξ_2, ϕ such that

s_{1n} is the smallest invariant mass of two neighbors, and $s_{n2} < s_{13}$

Reversing the inequality defines a second chamber in which n is radiated on the left side of 1.



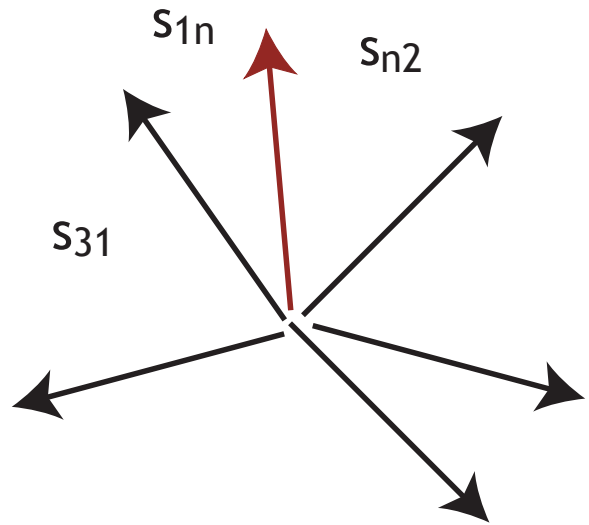
These prescriptions put reasonable upper limits on the ξ_{1j} integrals.

The ordering of virtualities s_{ij} is similar to the ordering in a parton shower. In fact, we can identify s_{ij} with the evolution variable of a parton shower.

We look at the emission in the chamber

between 1 and 2, on the side of 1

as an emission from the gluon 1
in the antenna (in the sense of VINCIA)
of gluons 1 and 2.



At each stage in the shower, we choose an antenna and an emission side at random.

The correspondence to Altarelli-Parisi is

$$(1-z) = \frac{1}{(1+\xi_1+\xi_2)}$$

and

$$\int \frac{d\xi_2}{\xi_2} \int \frac{d\xi_1}{\xi_1} \approx \int \frac{dQ^2}{Q^2} \int \frac{dz}{z(1-z)}$$

We also choose definite values of the gluon helicities.

We can exactly solve for emissions with the measure

$$\int \frac{d\xi_2}{\xi_2} \int \frac{d\xi_1}{\xi_1} \int \frac{d\phi}{2\pi} \cdot \frac{3\alpha_s(\xi_2 s_{12})}{2\pi}$$

with the leading-log formula for $\alpha_s(Q^2)$

The gluon splitting functions are:

$$\begin{aligned} P(g_+ \to g_+ g_+) &= \frac{1}{z(1-z)} \\ P(g_+ \to g_- g_+) &= \frac{(1-z)^4}{z(1-z)} \\ P(g_+ \to g_+ g_-) &= \frac{z^4}{z(1-z)} \end{aligned}$$

We implement the numerators as a weight applied to each emission.

More precisely, each emission then receives a weight $w(x)$, equal to the numerator of the splitting function times zero if the chosen kinematics violates the chamber conditions.

In most Monte Carlo programs, we generate weight-1 events by hit-or-miss. Arrange the weights so that, always, $w(x) < 1$. For each generated event, choose a random number R such that $0 < R < 1$. Accept the event if $w > R$.

In a parton shower, this method must be modified. It is obviously not correct to reject the whole event if $w < R$ in one parton emission.

The PYTHIA manual gives a simple solution to this problem.

Let $t = \log(m_h^2/s_{ij})$ be the evolution variable. Let $g(t)$ be the emission probability for one parton in a shower that we can model explicitly (e.g. all numerators = 1, no chamber inequalities), and let $f(t)$ be the emission probability for the true shower. In general, f, g depend on all of the kinematic variables of the emission. The weight is

$$w = f/g < 1$$

The correct probability for emitting the next parton obeys

$$\frac{dP}{dt} = f(t) \cdot P(t)$$

for which the solution is

$$P(t) = e^{-S(t)} \qquad S(t) = \int_0^t dt' f(t')$$

$S(t)$ is called the Sudakov factor. To choose the next t, we would solve: $log(1/R) = S(t)$ - but typically this is too hard to do.

Choose $g(t)$ so that we can solve exactly the corresponding equation for its Sudakov factor. Then, proceed as follows:

Work forward in t. Starting from $t = 0$, choose the next emission point t_1 using the probability $g(t)$.

Compute $w = f/g$ and accept the emission if $w > R$.

If this fails, use t_1 as the starting point and go farther forward, choosing the emission point t_2 using the probability $g(t)$. Accept this emission with probability $w = f/g$.

Continue as needed, stopping when t reaches $t_{\max}$ (minimum virtuality).

Add up all possible sequences of emissions and acceptance or rejections. The sum is equal to $P(t) = \exp[-S(t)]$!

For a simple parton shower, we choose for the numerators of the splitting functions:

$$1\ , \quad \frac{1}{(1+\xi_1+\xi_2)^4}\ , \quad \frac{\xi_2^4}{(1+\xi_1+\xi_2)^4}$$

times zero if the chosen point violates the chamber inequalities.

However, we could also use more complicated weights. The prescription

$$w = \frac{|\mathcal{M}(h \to ng)|^2/|\mathcal{M}(h \to (n-1)g)|^2}{(2p_1\cdot p_2)/(2p_1\cdot p_n)(2p_n\cdot p_2)}$$

reweights the emissions to the probabilities given by exact tree-level matrix elements. We can use this prescription as long as our computer has the strength to compute the matrix elements.

The two prescriptions agree for h-> 3 g, so the simple shower is exact at this level. It is quite accurate at higher levels.

In principle, our code allows the maximum level at which exact matrix elements are used to be set to any value.
The user can choose this value, depending on her patience.

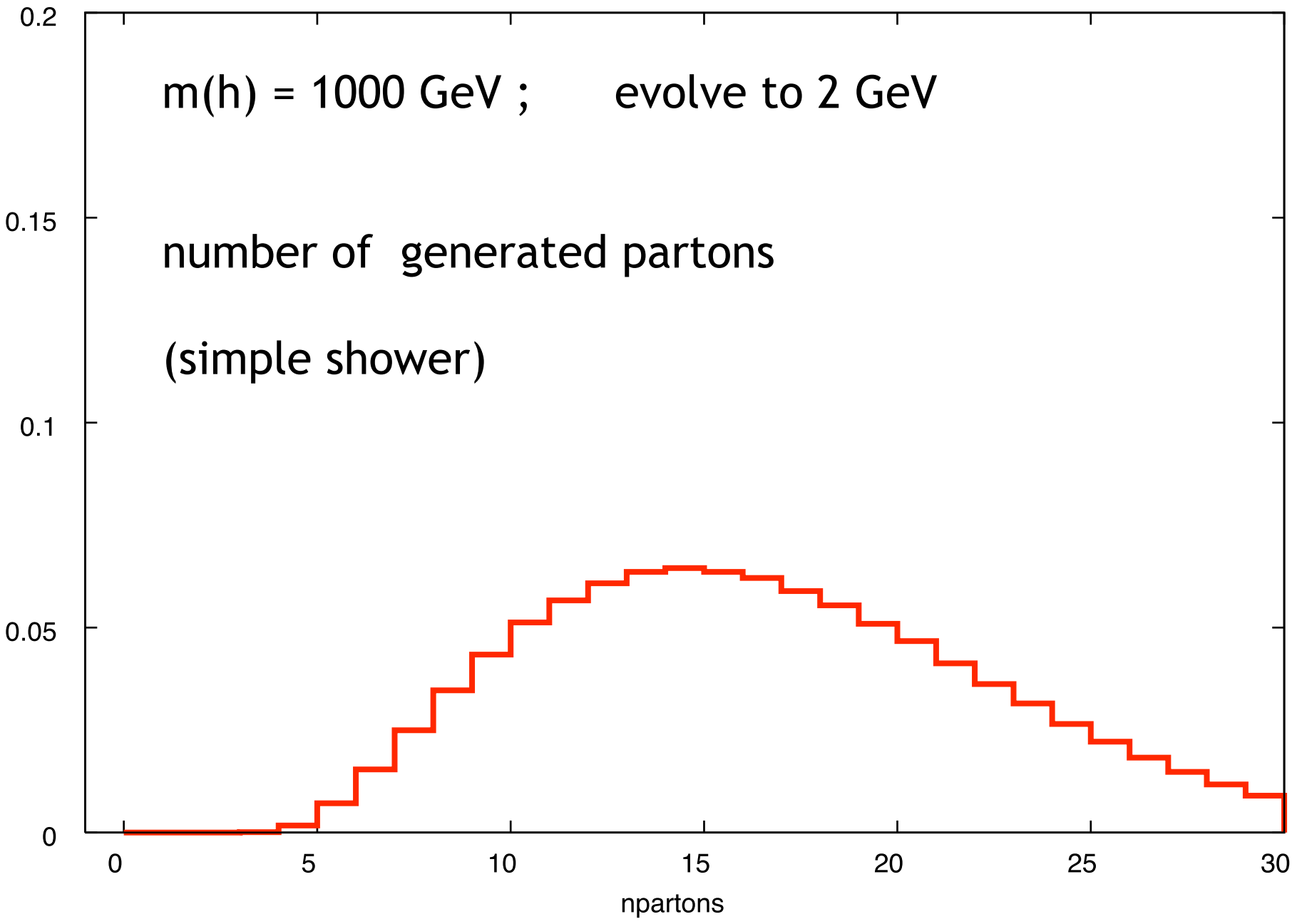
The timing (for the plots I will show) is :

0.7 msec/event for the simple shower

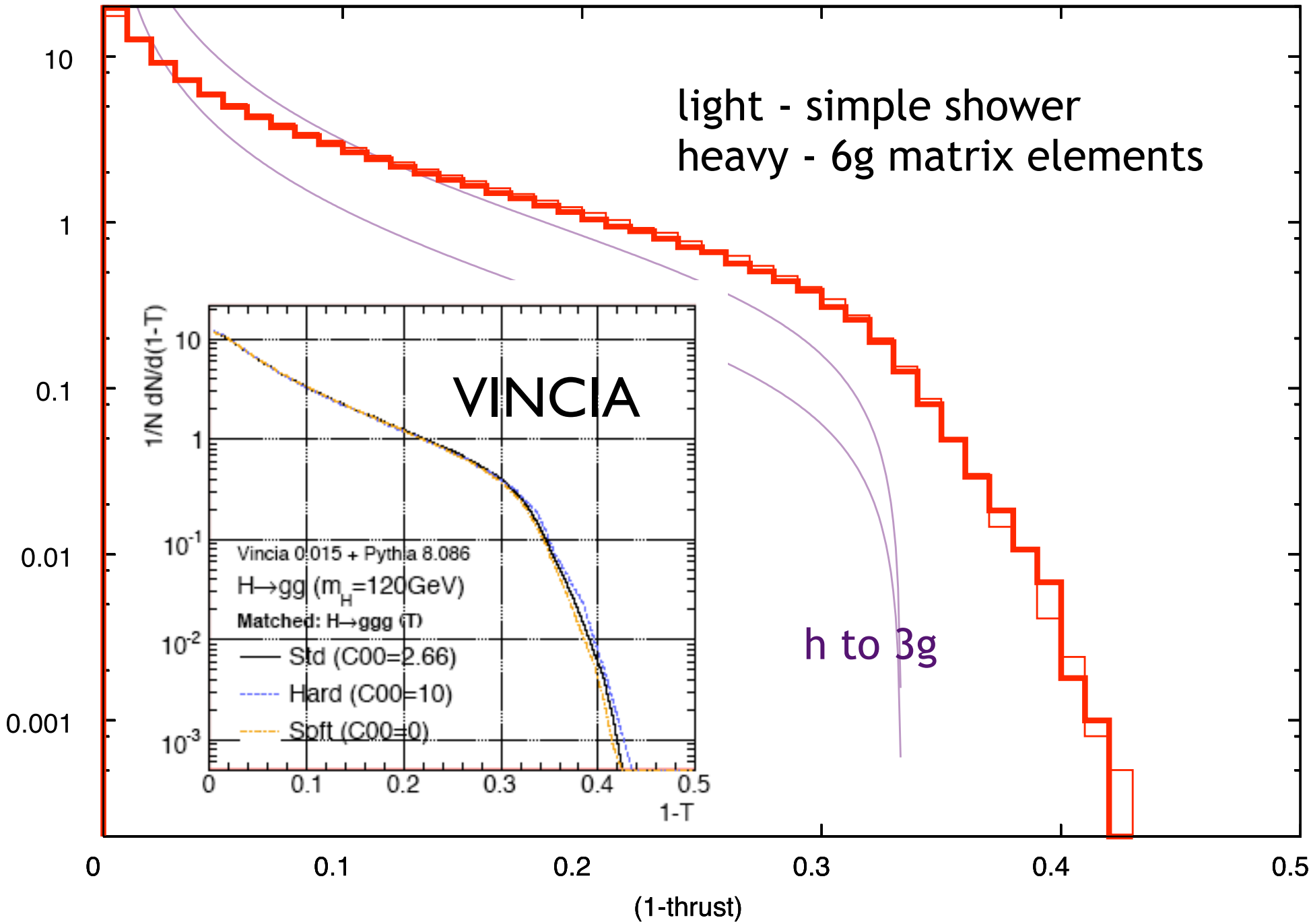
1.7 msec/event using 6 gluon matrix elements

7.6 msec/event using 8 gluon matrix elements

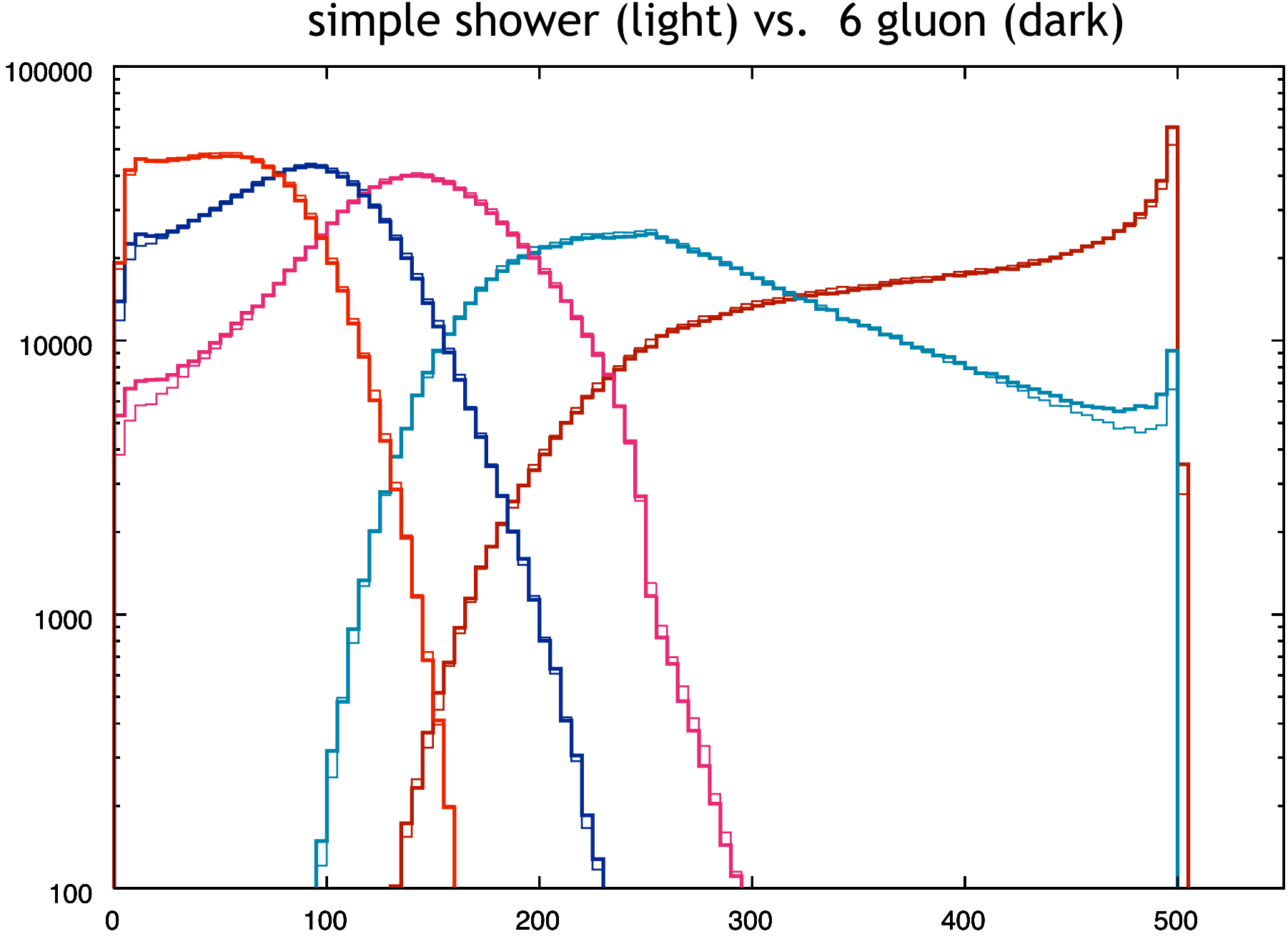
Here are some results from the program:

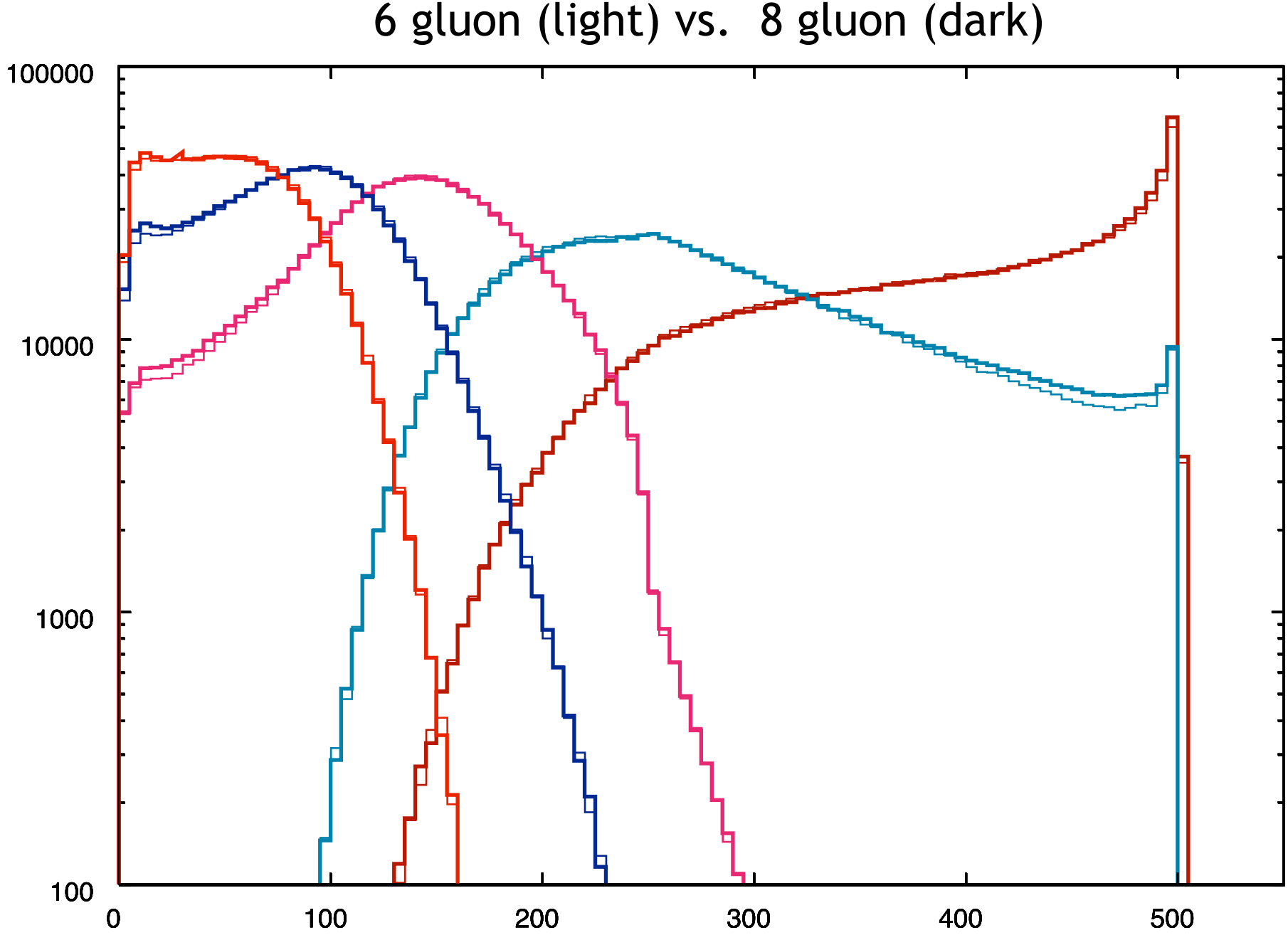

m(h) = 1000 GeV ; evolve to 2 GeV
number of generated partons
(simple shower)
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
npartons

Thrust distribution



To see better what changes with a more exact calculation, cluster the partons with ycut = 0.0001 (cluster mass of 10 GeV), and plot the energies of the 5 highest-energy clusters.


simple shower (light) vs. 6 gluon (dark)
100000
10000
1000
100
0
100
200
300
400
500


6 gluon (light) vs. 8 gluon (dark)
100000
10000
1000
100
0
100
200
300
400
500

Conclusions:

This is a proof of principle for a new way to incorporate exact matrix elements into a parton shower. Only the simplest situation has been implemented so far.

In principle, the method generalizes to processes with massive particles in the final state. I apologize that the shower in

$$e^+e^- \to t\bar{t} + ng$$

is not ready for presentation at this conference.

The generalization to initial-state radiation showers is even more virtual.

Still, there is promise that this might be an interesting tool for modeling multijet QCD processes.