LUND UNIVERSITY

Basics of Event Generators I

Leif Lönnblad

Department of Astronomy
and Theoretical Physics
Lund University

MCnet School on Event Generators
São Paolo 2015.04.27

Outline of Lectures

- Lecture I: Basics of Monte Carlo methods, the event generator strategy, matrix elements, LO/NLO, . . .
- Lecture II: Parton showers, initial/final state, matching/merging, . . .
- Lecture III: Matching/merging (cntd.), underlying events, multiple interactions, minimum bias, pile-up, hadronization, decays, . . .
- Lecture IV: Protons vs. heavy ions, summary, . . .

Buckley et al. (MCnet collaboration), *Phys. Rep.* **504** (2011) 145.

Outline

How do we numerically estimate an integral of an arbitrary function $f(\boldsymbol{x})$?

$$I = \int_{\Omega} d^n\boldsymbol{x}\, f(\boldsymbol{x})$$

Simple discretization (Simpsons rule, Gaussian quadrature) can be extremely inefficient if

- n is large
- Ω is complicated
- $f(\boldsymbol{x})$ has peaks and divergencies.

Importance sampling

Assume we are able to generate random variables $\boldsymbol{X}_i$ such that

$$P\left(x^{(j)} < X_i^{(j)} < x^{(j)} + dx^{(j)}\right) = p_X(\boldsymbol{x})$$

if $p(\boldsymbol{x}) > 0, \forall \boldsymbol{x} \in \Omega$ and zero outside, we can rewrite our integral

$$I = \int_\Omega d^n\boldsymbol{x} \frac{f(\boldsymbol{x})}{p_X(\boldsymbol{x})} p_X(\boldsymbol{x}).$$

Now, for any random variable Y and any function g, we know that

$$\frac{1}{N}\sum_{i=1}^{N} g(Y_i) \approx \langle g(Y) \rangle = \int_{-\infty}^{\infty} dy\, p_Y(y) g(y)$$

Hence

$$\left\langle \frac{f(\boldsymbol{X})}{p_X(\boldsymbol{X})} \right\rangle = \int_\Omega d^n\boldsymbol{x} \frac{f(\boldsymbol{x})}{p_X(\boldsymbol{x})} p_X(\boldsymbol{x}) = I$$

So, we can numerically estimate our integral by generating N points $\boldsymbol{X}_i$ and take the average of $f(\boldsymbol{X})/p_X(\boldsymbol{X})$.

In doing so we will get an error which we can estimate by

$$\delta \approx \sigma\left(\frac{f(\boldsymbol{X})}{p_X(\boldsymbol{X})}\right)/\sqrt{N}$$

where the variance is given by $\sigma^2(Y) = \langle Y^2 \rangle - \langle Y \rangle^2$.

(cf. Simpsons rule $\delta \propto 1/N^{4/d}$)

Clearly if $p_X(\boldsymbol{x}) = C|f(\boldsymbol{x})|$, we get the smallest possible error (if $f(x) > 0$ the error is zero).

However, with a bad choice of p_X, the variance and the error need not even be finite.

Numerically generating points directly according to $p_X(\boldsymbol{x}) = C|f(\boldsymbol{x})|$ is in general difficult, and typically involves analytically solving the integral we want to estimate. But there are some tricks. . .

Normally we only have uniformly distributed (flat) random numbers available on the computer

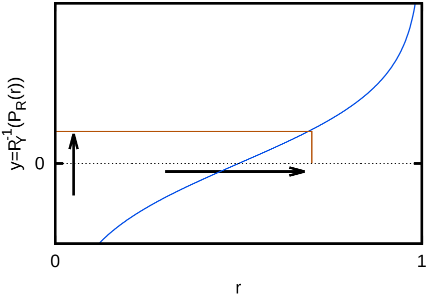
$$p_R(r) = \begin{cases} 1 & 0 < r < 1 \\ 0 & \text{otherwise} \end{cases}$$

We can transform any distribution into any other by the a transformation using the cumulative distributions

$$P_Y(y) = \int_{-\infty}^{y} dt\, p_Y(t) = \int_0^r dt\, p_R(t) = P_R(r) = r$$

as long as $P_Y^{-1}(P_R(r))$ is a monotonically increasing function.

(If $P_Y^{-1}(P_R(r))$ is not monotonous, we can divide up in intervals.)



Think of it as variable substitution:

$$\int_{y_{\min}}^{y_{\max}} p_Y(y) f(y) dy = \left\{ \begin{array}{rcl} P_Y(y) & = & r \\ \frac{dy}{dr} & = & \frac{1}{p_Y(y)} \\ P_Y(y_{\min}) & = & 0 \\ P_Y(y_{\max}) & = & 1 \end{array} \right\} = \int_0^1 f(P_Y^{-1}(r)) dr$$

What if P_Y^{-1} is hard to find ...

The Accept/Reject Method

Assume we want to generate random variables, Y_i, according to some difficult distribution $p_Y(y)$. We already know how to generate according to some other distribution, $p_{Y'}(y)$ such that $Cp_{Y'}(y) \geq p_Y(y)$ everywhere.

1. Generate Y' according to $p_{Y'}(y)$
2. Generate R according to a flat distribution
3.
    - If $\frac{p_Y(Y')}{Cp_{Y'}(Y')} > R$ then accept $Y = Y'$
    - otherwise reject Y' and goto 1

The accepted Y will be distributed according to $p_Y(y)$.

We need $2C$ random numbers to get one Y.

Multi-channel

Sometimes it is difficult to find an overestimate. But there are many tricks!
Assume $p(x) \leq g(x) = \sum_i g_i(x)$ where we know how to generate random variables according to each g_i.

1. select i with relative probability $A_i = \int g_i(x)dx$
2. select x according to $g_i(x)$
3. throw away x and i with probability $f(x)/\sum_i g_i(x)$

$$\int f(x)dx = \int \frac{f(x)}{g(x)} \sum_i g_i(x) = \sum A_i \int \frac{g_i(x)dx}{A_i} \frac{f(x)}{g(x)}$$

Alternatively we can divide up the integration region into sub-regions, where we can find a suitable overestimate. Again we first choose region according to the integral of the overestimate, and then generate in there.

How do we get random numbers?

There are ways of getting truly random numbers, but we will use (and actually prefer) pseudo-random numbers.

There are many algorithms around for producing pseudo-random numbers.

The simplest one is called Linear congruential:

- Pick integers a, b, m, and a seed R_0
- generate random numbers according to

$$R_i = aR_{i-1} + b \qquad \pmod m$$

DON'T USE THIS

The Marsaglia Effect

Take successive d-tuplets from a congruential generator with t bits ($m = 2^t$).
Interpret them as point coordinates in a d-dimensional hypercube.

Then they all fall on at most $(d!2^t)^{1/d}$ parallel hyperplanes.

t	$d = 3$	$d = 4$	$d = 6$	$d = 10$
16	73	35	19	13
32	2 953	566	120	41
48	119 086	9 065	766	126
64	4 801 280	145 055	4 866	382

Disastrous for any repetitive application.

Has lead to explosion of new tests and new generators.

Don't worry, there are several good pseudo random generators out there:

`http://en.wikipedia.org/wiki/List_of_random_number_generators`

Predicting an Observable

To calculate the expectation value of an observable, $\mathcal{O}$, in a $pp \rightarrow X$ collision we need to evaluate an integral looking like

$$\langle\mathcal{O}\rangle = \sum_{n}\sum_{\boldsymbol{Q}}\int d^{4n}\boldsymbol{p}\;|\mathcal{M}_n(\boldsymbol{Q},\boldsymbol{p})|^2\,\mathcal{O}_n(\boldsymbol{Q},\boldsymbol{p})\,\boldsymbol{\Phi}_n(\boldsymbol{p})$$

- $\boldsymbol{p}$ are the momenta of the n particles
- $\boldsymbol{Q}$ are their quantum numbers
- $\mathcal{M}$ is the matrix element
- $\boldsymbol{\Phi}_n$ is the phase space density etc.

So now, all we need to do is to find a probability distribution $p(n, \boldsymbol{Q}, \boldsymbol{p})$ such that

$$C\, p(n, \boldsymbol{Q}, \boldsymbol{p}) = |\mathcal{M}_n(\boldsymbol{Q}, \boldsymbol{p})|^2 \ \boldsymbol{\Phi}_n(\boldsymbol{p})$$

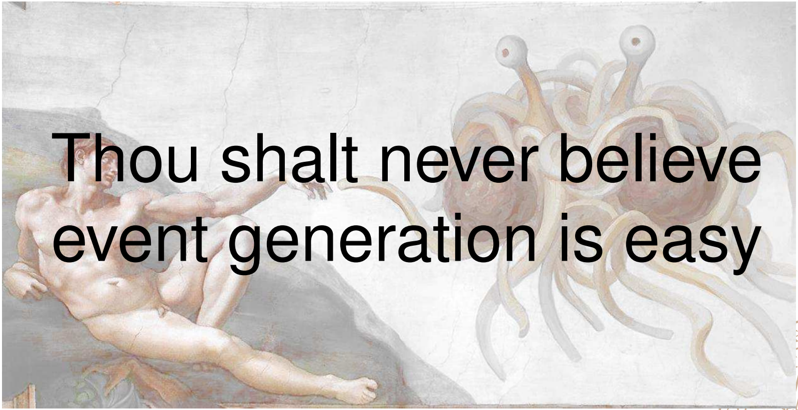
Then we generate N points, $(n_i, \boldsymbol{Q}_i, \boldsymbol{p}_i)$ according to this and get

$$\langle \mathcal{O} \rangle = \frac{C}{N} \sum_i^N \mathcal{O}_n(\boldsymbol{Q}_i, \boldsymbol{p}_i)$$

In the same way as we do when measuring the observable experimentally.

We are generating events. And we can measure several observables in one go. Life is simple!

The First Commandment of Event Generation



Thou shalt never believe event generation is easy

There are no free lunches

- $\mathcal{M}$ can typically only be calculated perturbatively to leading and maybe next-to-leading order for a small number of particles.
- $\mathbf{\Phi}_n$ is not trivial
- finding $p(n, \boldsymbol{Q}, \boldsymbol{p})$ may be **very** difficult

Weighted vs. Unweighted Events

We can, of course, use any probability distribution and get

$$\langle \mathcal{O} \rangle = \frac{C}{N} \sum_{i}^{N} \frac{|\mathcal{M}_n(\boldsymbol{Q}_i, \boldsymbol{p}_i)|^2 \; \boldsymbol{\Phi}_n(\boldsymbol{p}_i)}{p(n_i, \boldsymbol{Q}_i, \boldsymbol{p}_i)} \mathcal{O}_n(\boldsymbol{Q}_i, \boldsymbol{p}_i)$$

which means we get weighted events.

This is OK as long as the variance is not too big.

Factorization - Divide and conquer!

$$
\begin{aligned}
\langle\mathcal{O}\rangle = & \sum_{n_q,\boldsymbol{Q}_q}\int d^{4n_q}\boldsymbol{q}\,\left|\mathcal{M}_{n_q}(\boldsymbol{Q}_q,\boldsymbol{q})\right|^2\boldsymbol{\Phi}_{n_q}(\boldsymbol{q})\times \\
& \left[\sum_{n_k,\boldsymbol{Q}_k}\int d^{4n_k}\boldsymbol{k}\,PS(\boldsymbol{Q}_q,\boldsymbol{q};\boldsymbol{Q}_k,\boldsymbol{k})\times\right. \\
& \left.\left\{\sum_{n_p,\boldsymbol{Q}_p}\int d^{4n_p}\boldsymbol{p}\,H(\boldsymbol{Q}_k,\boldsymbol{k};\boldsymbol{Q}_p,\boldsymbol{p})\mathcal{O}_{n_p}(\boldsymbol{Q}_p,\boldsymbol{p})\right\}\right]
\end{aligned}
$$

- $\mathcal{M}$ now only gives a few partons
- *PS* is a parton shower giving more partons with unit probability
- *H* is hadronization and decays giving final state hadrons with unit probability

Factorization

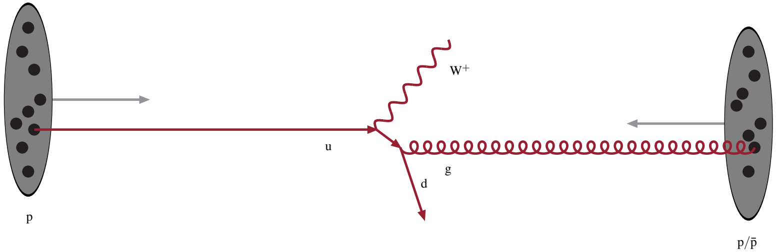
Relies on the factorization ansatz.

The cross section and main structure of the event is determined by the hard partonic sub process.

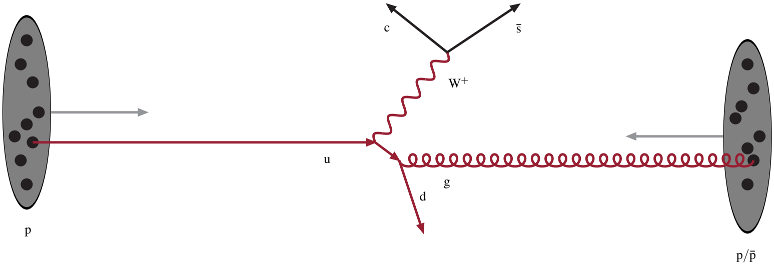
Parton showers and hadronization happens at lower (softer) scales and *dresses* the events without influencing the cross section.

The structure of a proton collision

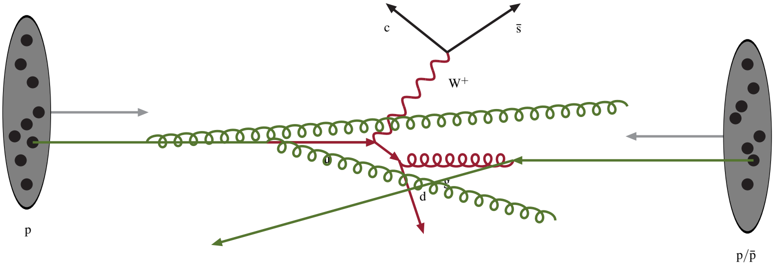
The hard/primary scattering



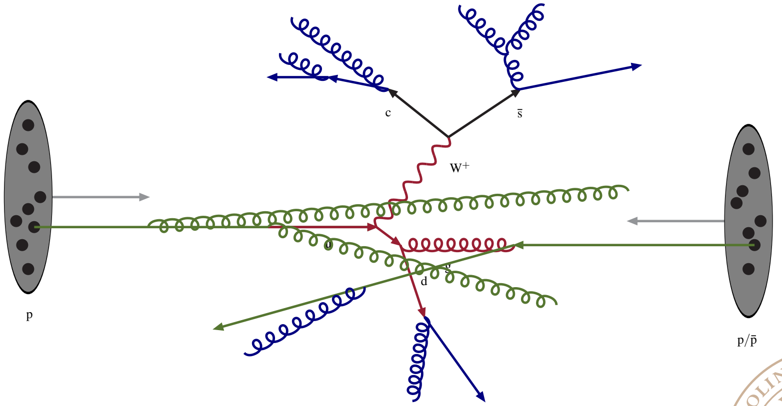
Immediate decay of unstable elementary particles



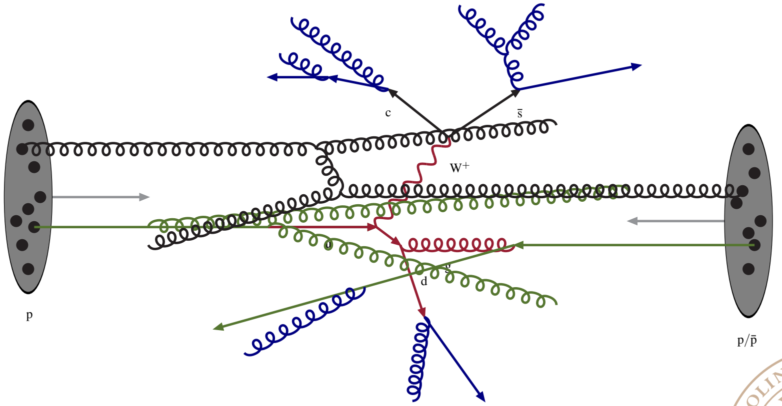
Radiation from particles before primary interaction



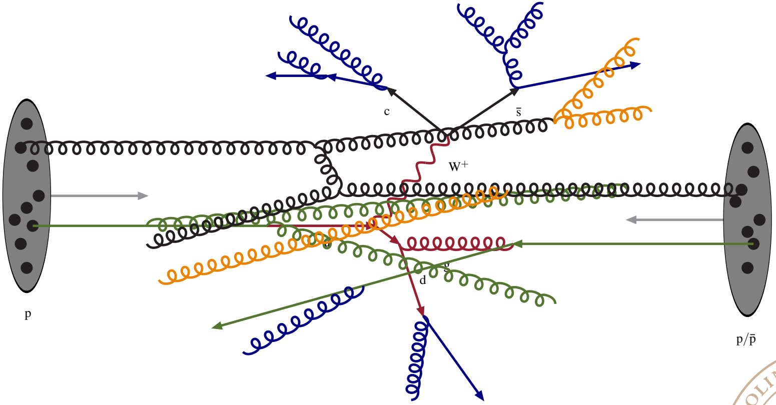
Radiation from produced particles



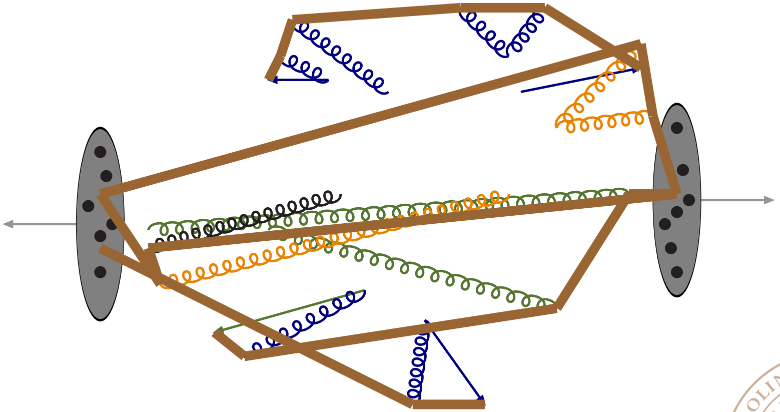
Additional sub-scatterings



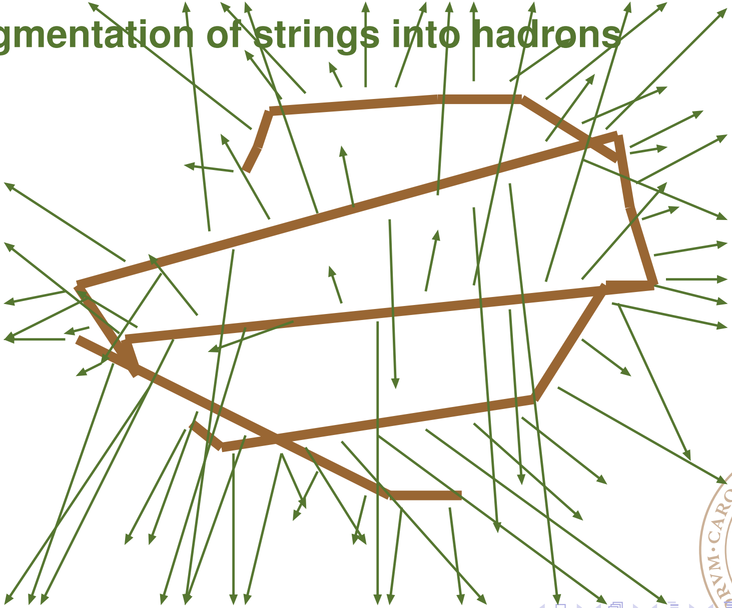
... with accompanying radiation



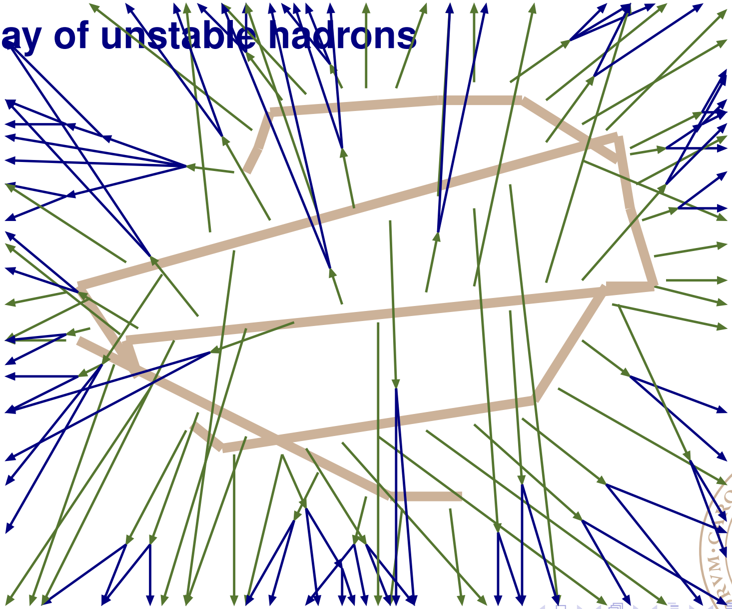
Formation of *colour strings*



Fragmentation of strings into hadrons



Decay of unstable hadrons



Everything at the LHC is QCD

- Any measurement at the LHC requires understanding of QCD
- Electro-weak processes or BSM processes are easy (although sometimes tedious)
- Even golden signals such as $H \to 4\mu$ are influenced by QCD
- Any observable prediction will have QCD corrections $\langle \mathcal{O} \rangle = \sigma_0(1 + \alpha_s C_1 + \alpha_s^2 C_2 + \ldots)$
- Any signal will have a QCD background
- QCD is difficult

Everything at the LHC is QCD

- Any measurement at the LHC requires understanding of QCD
- Electro-weak processes or BSM processes are easy (although sometimes tedious)
- Even golden signals such as $H \to 4\mu$ are influenced by QCD
- Any observable prediction will have QCD corrections $\langle\mathcal{O}\rangle = \sigma_0(1 + \alpha_s C_1 + \alpha_s^2 C_2 + \ldots)$
- Any signal will have a QCD background
- QCD is difficult

Event Generators are all about QCD.

Why is QCD difficult?

- α_s is not very small ($\gtrsim 0.1$)
- The gluon has a self-coupling and we get a lot of gluons
- Even if α_s is small the phase space for emitting gluons is large. In any α_s expansion the coefficients may be large.
- In the end we need hadrons, which are produced in a non-perturbative process.

We need models for parton showers and hadronization

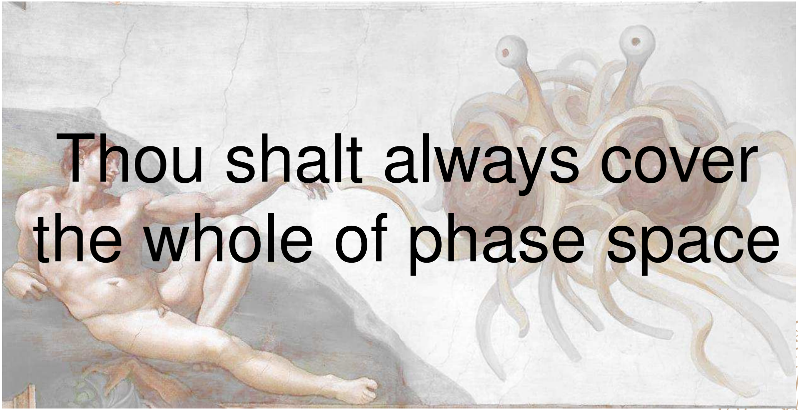
Matrix Element Generation

We always need to start with a $2 \to n$ matrix element. This can in principle be obtained from the standard model (or BSM) Lagrange density in a straight-forward manner.

However,

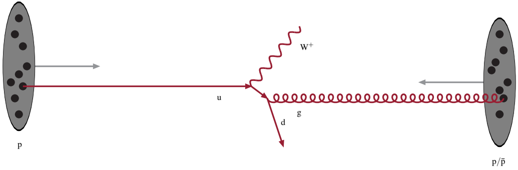
- On tree-level we have divergencies if the scale ($\sim p_\perp$) is small. Soft or collinear partons.
- Beyond leading order we get nasty loops and infinities
- If n is large, the number of diagrams grows factorially
- If n is large, it is difficult to find a suitable probability distribution for the momenta

The Second Commandment of Event Generation



Thou shalt always cover the whole of phase space

Simple $2 \rightarrow 2$ Matrix Elements



Can in principle be written down by hand from relevant Feynman diagrams.

$$\sigma = \int dx_1 dx_2 d\hat{t}\, f_i(x_1, Q^2) f_j(x_2, Q^2) \frac{d\hat{\sigma}_{ij}}{d\hat{t}}$$

With the parton densities sampled at a scale $Q^2 \sim |\hat{t}| \sim p_\perp^2$.

Also fairly easy to generate as the integrand is fairly flat in

$$d\ln(x_1)\, d\ln(x_2)\, d\ln(p_\perp^2)$$

Note however that $\hat{\sigma}$ may be divergent as $\hat{t} \to 0$.

Eg. Standard QCD ME:

$$\frac{\hat{\sigma}_{gg\to gg}}{d\hat{t}} = \frac{9\pi\alpha_s^2}{4\hat{s}^2}\left(\frac{\hat{s}^2}{\hat{t}^2} + 2\frac{\hat{s}}{\hat{t}} + 3 + 2\frac{\hat{t}}{\hat{s}} + \frac{\hat{t}^2}{\hat{s}^2} + \frac{\hat{u}^2}{\hat{s}^2} + 2\frac{\hat{u}}{\hat{s}} + 3 + 2\frac{\hat{s}}{\hat{u}} + \frac{\hat{s}^2}{\hat{u}^2} + \frac{\hat{t}^2}{\hat{u}^2} + 2\frac{\hat{t}}{\hat{u}} + 3 + 2\frac{\hat{u}}{\hat{t}} + \frac{\hat{u}^2}{\hat{t}^2}\right)$$

We clearly need a cutoff.

Typically this is given as a jet resolution scale, which for this simple process typically means a $p_\perp$-cut.

Eg. the $k_\perp$-algorithm:

Find the pair of particles with smallest

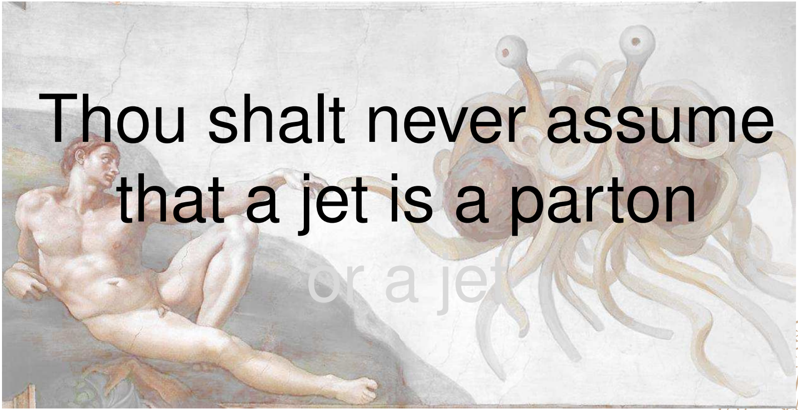
$$k_{\perp ij} = \frac{\min(k_{\perp i}, k_{\perp j})}{R} \sqrt{\Delta\phi_{ij}^2 + \Delta\eta_{ij}^2}$$

and cluster them together into one. Or if any $k_{\perp i}$ is smaller cluster it to the beam.

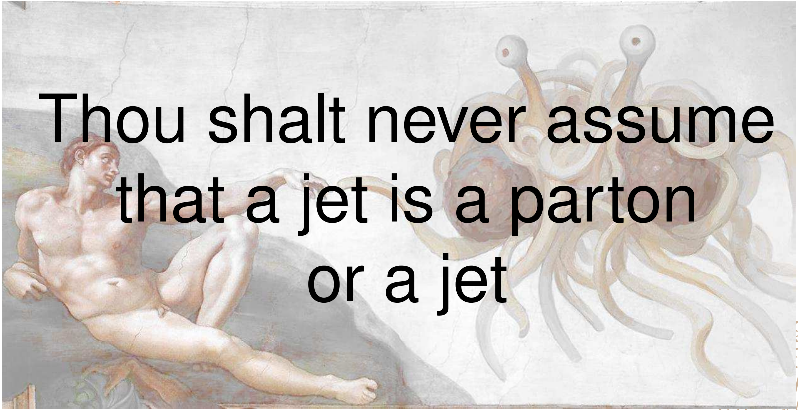
Continue until all *clusters* have $k_{\perp ij}$ and $k_{\perp i}$ above some cut.

These remaining jets are then close to the original partons.

The Third Commandment of Event Generation



The Third Commandment of Event Generation



Thou shalt never assume that a jet is a parton or a jet

Higher Order Tree-Level Matrix Elements

We can go on to higher order $2 \to n$ Matrix Elements. This is in principle straight forward and can even be automated. However the number of diagrams grows $\propto n!$ which makes generation of events forbiddingly slow for $n \gtrsim 8$.

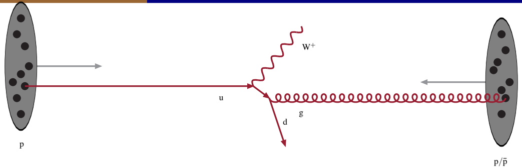
Remember also the difficulty in constructing a reasonable probability distribution for the momenta to sample the phase space, especially since there are divergencies everywhere.

Multi-channel sampling helps:

$$\sigma \propto \left|\sum_i \mathcal{M}_i\right|^2 = \sum_i |\mathcal{M}_i|^2 \frac{\left|\sum_j \mathcal{M}_j\right|^2}{\sum_j |\mathcal{M}_j|^2}$$

Available Tree-Level Generators

- AlpGen
  `http://mlm.web.cern.ch/mlm/alpgen`
- AMEGIC++
  `http://projects.hepforge.org/sherpa`
- CompHep
  `http://comphep.sinp.msu.ru`
- Helac/Phegas
  `http://helac-phegas.web.cern.ch/helac-phegas`
- MadGraph/MadEvent
  `http://madgraph.hep.uiuc.edu`
- ...



We can use a tree-level 2 $\rightarrow$ 2 ME to predict an observable such as the rapidity distribution of a jet in a W-event.

We can try to get a better estimate by going to higher order tree-level MEs

$$\begin{aligned}
\langle\mathcal{O}\rangle_{1j} &= \sigma_{\rightarrow W+1j}(\mu) \otimes \mathcal{O}(W+j) \\
\langle\mathcal{O}\rangle_{2j} &= \sigma_{\rightarrow W+2j}(\mu) \otimes \mathcal{O}(W+j) \\
\langle\mathcal{O}\rangle_{3j} &= \sigma_{\rightarrow W+3j}(\mu) \otimes \mathcal{O}(W+j) \\
&\vdots
\end{aligned}$$

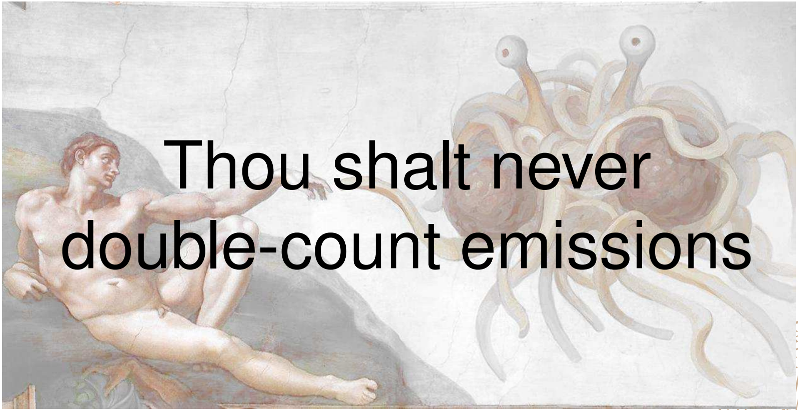
Where we use some jet-resolution scale μ to cut off divergencies.

But we cannot simply add these together, since each cross section is inclusive

The tree-level $ab \to W + 1j$ matrix element gives the cross section for the production of a W plus at least one jet.

Hence it includes also a part of the tree-level $ab \to W + 2j$ matrix element.

The Fourth Commandment of Event Generation



Thou shalt never double-count emissions

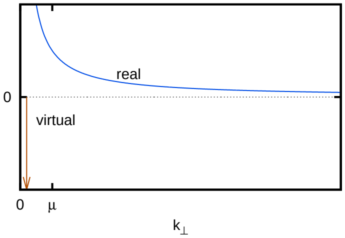
Next-to-Leading Order

To correctly sum $W+1j$ and $W+2j$ contributions to an observable, we need to add virtual contributions to the generated $W+1j$ states. In that way we get a consistent expansion of the observable.

$$\begin{aligned}\langle\mathcal{O}\rangle_{1j} &= \alpha_{\rm s}C_{11}(\mu)+\alpha_{\rm s}^2C_{12}(\mu)\\ \langle\mathcal{O}\rangle_{2j} &= \alpha_{\rm s}^2C_{22}(\mu)\end{aligned}$$

$$\langle\mathcal{O}\rangle_{\rm NLO}=\langle\mathcal{O}\rangle_{1j}+\langle\mathcal{O}\rangle_{2j}$$

Here the jet resolution scale μ is essential, since the virtual corrections are infinite and negative. But if we add together the 1j virtual terms and the unresolved 2j contributions, (the contributions below μ) the sum, $\alpha_s^2 C_{12}(\mu)$ is finite.



Today there are several NLO generators available (cf. next lecture).

They produce few-parton events and you can measure jet observables

Clearly if you have a generator producing $W + 1j$ to NLO, any observable you measure which depends on two jets will only be predicted to leading order.

This can sometimes be tricky. . .

Today there are several NLO generators available (cf. next lecture).

They produce few-parton events and you can measure jet observables (assuming a parton is a jet)

Clearly if you have a generator producing $W + 1j$ to NLO, any observable you measure which depends on two jets will only be predicted to leading order.

This can sometimes be tricky...

Today there are several NLO generators available (cf. next lecture).

They produce few-parton events and you can measure jet observables (assuming a parton is a jet (which it isn't)).

Clearly if you have a generator producing $W + 1j$ to NLO, any observable you measure which depends on two jets will only be predicted to leading order.

This can sometimes be tricky…

- Leading order is the first order in α_s which gives a non-zero result for a given observable.
- If NLO corrections are large, we need NNLO.
- However, chances are that we have a poorly converging series in α_s.
- This means we need to resum.

All-Order Resummation

Rather than calculating a few terms in the α_s expansion exactly, we can try to approximate all terms.

It turns out that if we just consider the leading divergent part of the cross section, everything exponentiates

$$
\begin{aligned}
\sigma_{0j} &= C_{00} + \alpha_s C_{01} + \alpha_s^2 C_{02} + \ldots \approx C_{00} \exp(\alpha_s C'_{01}/C_{00}) \\
\sigma_{1j} &= \alpha_s C_{11} + \alpha_s^2 C_{12} + \alpha_s^3 C_{13} + \ldots \approx \alpha_s C_{11} \exp(\alpha_s C'_{12}/C_{11}) \\
&\vdots
\end{aligned}
$$

Even if the coefficients diverge as $\mu \to 0$ the exponentiation is finite.

The resummation corresponds to obtaining the leading logarithmic contributions to the coefficients

$$\propto \alpha_s^n \ln(\mu)^{2n}$$

This can be done analytically even to next-to-leading log $\propto \alpha_s^n \ln(\mu)^{2n-1}$ and higher.

Or by using parton showers...

Outline of Lectures

- Lecture I: Basics of Monte Carlo methods, the event generator strategy, matrix elements, LO/NLO, . . .
- Lecture II: Parton showers, initial/final state, matching/merging, . . .
- Lecture III: Matching/merging (cntd.), underlying events, multiple interactions, minimum bias, pile-up, hadronization, decays, . . .
- Lecture IV: Protons vs. heavy ions, summary, . . .

Buckley et al. (MCnet collaboration), *Phys. Rep.* **504** (2011) 145.