MC School 2017

Zoltán Nagy
DESY-HH

Born Level Cross Section

It is an integral of the

$$\sigma^B[O_J] = \int \overbrace{d\{p,f\}_m}^{\text{Phase space measure}} \underbrace{f_{a/A}(\eta_{\rm a},\mu_F^2) f_{b/B}(\eta_{\rm b},\mu_F^2)}_{\text{PDF functions}} \frac{1}{2\eta_{\rm a}\eta_{\rm b}\, s} \overbrace{\langle |M^{(0)}(\{p,f\}_m)|^2\rangle}^{\text{Spin averaged matrix element square}} \underbrace{O_J(\{p\}_m)}_{\text{Jet measurement function}}$$

$$\{p,f\}_m \equiv \{\eta_{\rm a} p_A, a; \eta_{\rm b} p_B, b, ; p_1, f_1, ; \ldots; p_m, f_m\} \quad .$$

Let's start with phase space integral

$$\int d\{p,f\}_m \, g(\{p,f\}_m) \equiv \prod_{i=1}^{m} \left\{ \sum_{f_i} \int \frac{d^d p_i}{(2\pi)^d} \, 2\pi\delta_+(p_i^2) \right\} \sum_a \int_0^1 d\eta_{\rm a} \sum_b \int_0^1 d\eta_{\rm b}$$

$$\times (2\pi)^d \delta\left(p_{\rm a} + p_{\rm b} - \sum_{i=1}^{m} p_i \right) \, g(\{p,f\}_m) \quad .$$

Monte Carlo Integral

An integral over a unit hypercube of any function can be calculated as an average of the function over a uniformly distributed random sample:

$$\int_0^1 d\rho_1 \cdots \int_0^1 d\rho_m \, g(\{\rho\}_m) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} g(\{\rho_i\}_m)$$

We want to do phase space integral and that is more complicated than this, so we have to translate the the integral over momenta to a unit hypercube, so there is an transformation as

$$R(\{\rho\}_n) = \{p\}_m$$

and it has an inverse

$$\rho_i = \rho_i(\{p\}_m)$$

$$1 = \int_0^1 d\rho_1 \cdots \int_0^1 d\rho_n \prod_{i=1}^{n} \delta(\rho_i - \rho_i(\{p\}_m))$$

Plug this 1 into the phase space integral and we have

Monte Carlo Integral

$$\int d\{p,f\}_m \, g(\{p,f\}_m) \equiv \prod_{i=1}^{m} \left\{ \sum_{f_i} \int \frac{d^d p_i}{(2\pi)^d} \, 2\pi\delta_+(p_i^2) \right\} \sum_a \int_0^1 d\eta_{\mathrm{a}} \sum_b \int_0^1 d\eta_{\mathrm{b}}$$
$$\times \, (2\pi)^d \delta\Big(p_{\mathrm{a}} + p_{\mathrm{b}} - \sum_{i=1}^{m} p_i \Big) \int_0^1 d\rho_1 \cdots \int_0^1 d\rho_n \prod_{i=1}^{n} \delta(\rho_i - \rho_i(\{p\}_m)) \; g(\{p,f\}_m) \quad .$$

Rearranging this expression, we have

$$\int d\{p,f\}_m \, g(\{p,f\}_m) \equiv \sum_{\{f\}_m} \int_0^1 d\rho_1 \cdots \int_0^1 d\rho_n \, g\big(\{f\}_m, R(\{\rho\}_n)\big)$$
$$\times \prod_{i=1}^{m} \left\{ \int \frac{d^d p_i}{(2\pi)^d} \, 2\pi\delta_+(p_i^2) \right\} \int_0^1 d\eta_{\mathrm{a}} \int_0^1 d\eta_{\mathrm{b}}$$
$$\times \, (2\pi)^d \delta\Big(\eta_{\mathrm{a}} p_A + \eta_{\mathrm{b}} p_B - \sum_{i=1}^{m} p_i \Big) \prod_{i=1}^{n} \delta(\rho_i - \rho_i(\{p\}_m)) \quad .$$

Now we can perform the integrals over the momenta in the last two lines and that gives the Jacobian of the transformation.

Monte Carlo Integrals

The Jacobian is this nice integral

$$J(\{\rho\}_m) = \prod_{i=1}^{m}\left\{\int \frac{d^d p_i}{(2\pi)^d}\,2\pi\delta_+(p_i^2)\right\}\int_0^1 d\eta_\mathrm{a}\int_0^1 d\eta_\mathrm{b}$$
$$\times\,(2\pi)^d\delta\left(\eta_\mathrm{a}p_A+\eta_\mathrm{b}p_B-\sum_{i=1}^{m}p_i\right)\prod_{i=1}^{n}\delta(\rho_i-\rho_i(\{p\}_m)) \quad .$$

Thus our integral is

$$\int d\{p,f\}_m\,g(\{p,f\}_m) = \int_0^1 d\rho_1\cdots\int_0^1 d\rho_n\,J(\{\rho\}_n)\sum_{\{f\}_m} g\big(\{f\}_m,R(\{\rho\}_n)\big)$$

Now, it is simple and defined over a unit hypercube and we can do the usual Monte Carlo estimate for this integral. One can define the new integrand as

$$\tilde{g}(\{\rho\}_n) = J(\{\rho\}_n)\sum_{\{f\}_m} g\big(\{f\}_m,R(\{\rho\}_n)\big)$$

Rambo

```
template<typename _OutputIterator>
double rambo(random_engine_type& rne, double s,
             _OutputIterator first, _OutputIterator last)
{
  /*  generate random momenta   */
  lorentzvector psum;
  unsigned int n = 0U;

  _OutputIterator iter = first;
  while(iter != last) {
    psum += (*iter = rambo_random_momentum(rne));
    n++; iter++;
  }

  /*    parameters of the conform transformation    */
  double x = std::sqrt(s)/psum.mag();
  threevector bVec = -psum.boostVector();

  /*   do the conform transformation    */
  iter = first;
  while(iter != last) {
    iter -> boost(bVec);
    iter -> operator*=(x);
    iter++;
  }

  return rambo_weight(n, s);
}
```

Rambo

```
lorentzvector rambo_random_momentum(random_engine_type& rne)
{
  std::uniform_real_distribution<double> rng(0.0,1.0);

  double E   = -std::log(rng(rne)*rng(rne));
  double pz  = E*(2.0*rng(rne) - 1.0);
  double pt  = std::sqrt(E*E - pz*pz);
  double phi = 2.0*M_PI*rng(rne);

  return {E, pt*std::cos(phi), pt*std::sin(phi), pz};
}
```

Importance Sampling

Let's stick a 1 again into our integral in the following way

$$\int d\{p,f\}_m\, g(\{p,f\}_m) = \int_0^1 d\rho_1 \cdots \int_0^1 d\rho_n\, \frac{w(\{\rho\}_n)}{w(\{\rho\}_n)}\tilde{g}\big(\{\rho\}_n\big)$$
$$= \int_0^1 d\rho_1 \cdots \int_0^1 d\rho_n\, w(\{\rho\}_n)\frac{\tilde{g}\big(\{\rho\}_n\big)}{w(\{\rho\}_n)}$$

Now we do another integral transformation as

$\{\rho\}_n = Q(\{\omega\}_{n'})$ and this transformation has a Jacobian $J_Q(\{\omega\}_{n'})$

$$\int d\{p,f\}_m\, g(\{p,f\}_m) = \int_0^1 d\rho_1 \cdots \int_0^1 d\rho_n\, \frac{w(\{\rho\}_n)}{w(\{\rho\}_n)}\tilde{g}\big(\{\rho\}_n\big)$$
$$= \int_0^1 d\rho_1 \cdots \int_0^1 d\rho_n\, w(\{\rho\}_n)\frac{\tilde{g}\big(\{\rho\}_n\big)}{w(\{\rho\}_n)}$$
$$= \int_0^1 d\omega_1 \cdots \int_0^1 d\omega_{n'}\, J_Q(\{\omega\}_{n'})\frac{\tilde{g}\big(Q(\{\omega\}_{n'})\big)}{w(Q(\{\omega\}_{n'}))}$$

Multi-channel Importance Sampling

What happens when we have a very complicated integrand (as usual in QCD) with lots of peaks in the physical phase space where we want to integrate. We can design a nice w(…) weight function for each peaks but we cannot have a single weight function over the whole phase space.

$$w(\{\rho\}_n) = \sum_{i=1}^{N_p} p_i \underbrace{w_i(\{\rho\}_n)}_{\text{Good approx. around only one singularity}} \qquad \int d\{\rho\}_n \, w(\{\rho\}_n) = \int d\{\rho\}_n \, w_i(\{\rho\}_n) = \sum_{i=1}^{N_p} p_i = 1$$

Now we do the same thing but Np times. Around every singular limits we have a transformation as

$$\{\rho\}_n = Q_i(\{\omega\}_{n'}) \qquad \text{with Jacobian} \qquad J_Q^{(i)}(\{\omega\}_{n'})$$

$$\begin{aligned}
\int d\{p,f\}_m \, g(\{p,f\}_m) &= \int_0^1 d\rho_1 \cdots \int_0^1 d\rho_n \, \frac{w(\{\rho\}_n)}{w(\{\rho\}_n)} \tilde{g}\big(\{\rho\}_n\big) \\
&= \sum_{i=1}^{N_p} p_i \int_0^1 d\rho_1 \cdots \int_0^1 d\rho_n \, w_i(\{\rho\}_n) \frac{\tilde{g}\big(\{\rho\}_n\big)}{w(\{\rho\}_n)} \\
&= \sum_{i=1}^{N_p} p_i \int_0^1 d\omega_1 \cdots \int_0^1 d\omega_{n'} \, J_Q^{(i)}(\{\omega\}_{n'}) \frac{\tilde{g}\big(Q_i(\{\omega\}_{n'})\big)}{w(Q_i(\{\omega\}_{n'})}
\end{aligned}$$

Multi-channel Importance Sampling

Now we generate *i with the probability density p_i*. To do that we introduce a new integral variable and

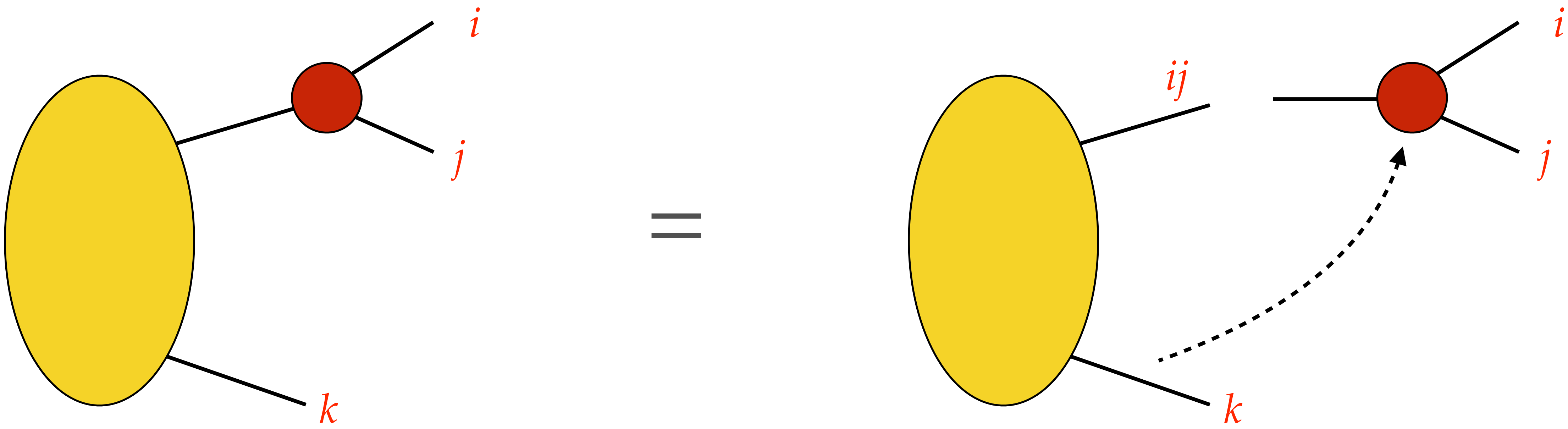
$$i = i(\xi)$$

$$\int d\{p,f\}_m \, g(\{p,f\}_m) = \int_0^1 d\xi \int_0^1 d\omega_1 \cdots \int_0^1 d\omega_{n'} \, J_Q^{(i(\xi))}(\{\omega\}_{n'}) \frac{\tilde{g}\big(Q_{i(\xi)}(\{\omega\}_{n'})\big)}{w(Q_{i(\xi)}(\{\omega\}_{n'}))}$$

Where do we use these?

Phase Space Factorization

The phase space integral is a beautiful object with lots of symmetry and factorization property



$$\int d\{\hat{p},\hat{f}\}_{m+1}\, g(\{\hat{p},\hat{f}\}_{m+1}) = \sum_{\substack{i,j\\ \text{pairs}}} \sum_{k\neq i,j} p_{ij,k} \int d\{p,f\}_m \int d\zeta_{ij,k}\, g(R_{ij,k}(\{p,f\}_m,\zeta_{ij,k}))$$

$$\sum_{\substack{i,j\\ \text{pairs}}} \sum_{k\neq i,j} p_{ij,k} = 1 \qquad \text{and} \qquad p_{ij,k} \geq 0$$