NLO QCD Calculations II

Rikkert Frederix
CERN theory group

MCNet and Terascale Alliance Monte Carlo School 2013, Goettingen, Germany, August 5-9, 2013

Master equation for hadron colliders

$$d\sigma = \sum_{a,b} \int dx_1 dx_2 \; f_a(x_1, \mu_F) f_b(x_2, \mu_F) \, d\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$$

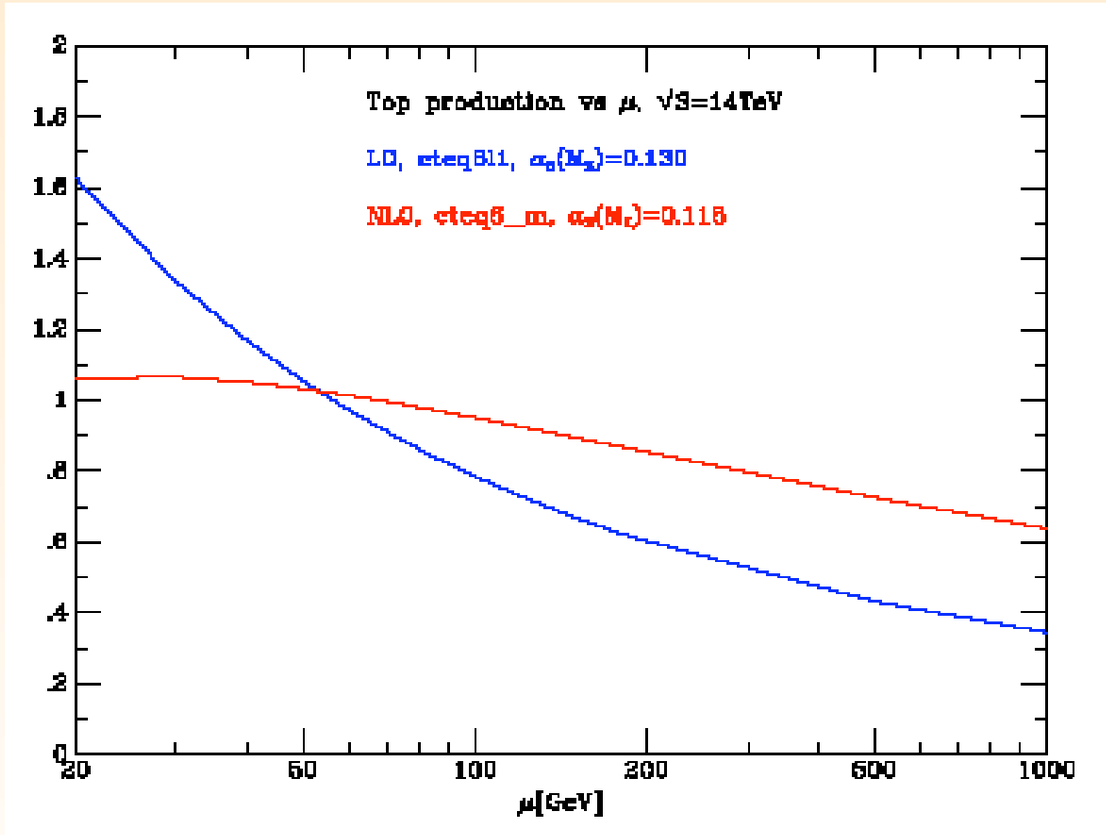
Parton density functions

Parton-level (differential) cross section

- Parton-level cross section from matrix elements: model and process dependent
- Parton density (or distribution) functions: process independent
- Differences between colliders given by parton luminosities

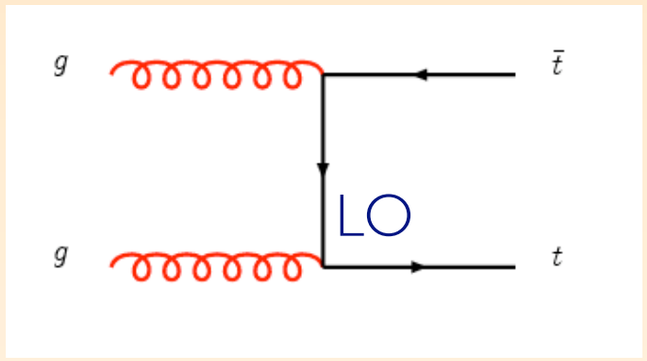
Going NLO

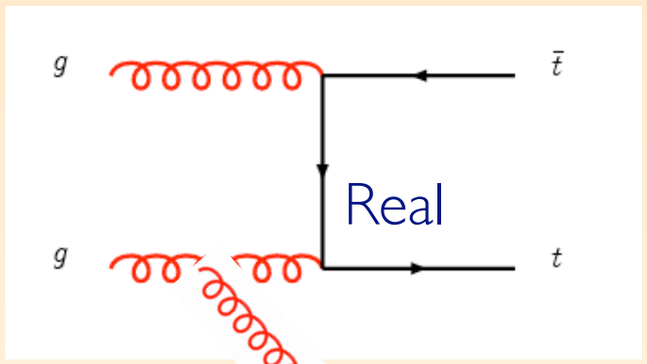
- At NLO the dependence on the renormalization and factorization scales is reduced
  - First order where scale dependence in the running coupling and the PDFs is compensated for via the loop corrections: **first reliable estimate of the total cross section**
  - Better description of final state: impact of extra radiation included (e.g. jets can have substructure)
  - Opening of additional initial state partonic channels

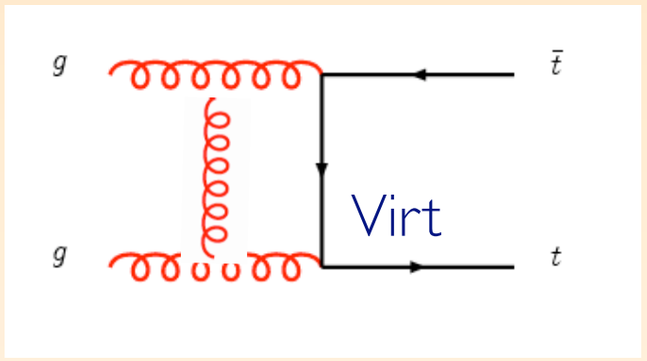


NLO…?

- Are all (IR-safe) observables that we can compute using a NLO code correctly described at NLO? Suppose we have a NLO code for pp → ttbar

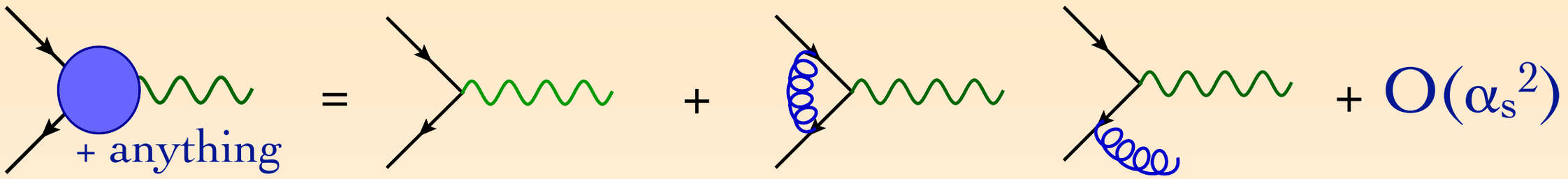






	NLO?
Total cross section	✔
Transverse momentum of the top quark	✔
Transverse momentum of the top-antitop pair	✘
Transverse momentum of the jet	✘
Top-antitop invariant mass	✔
Azimuthal distance between the top and anti-top	✘

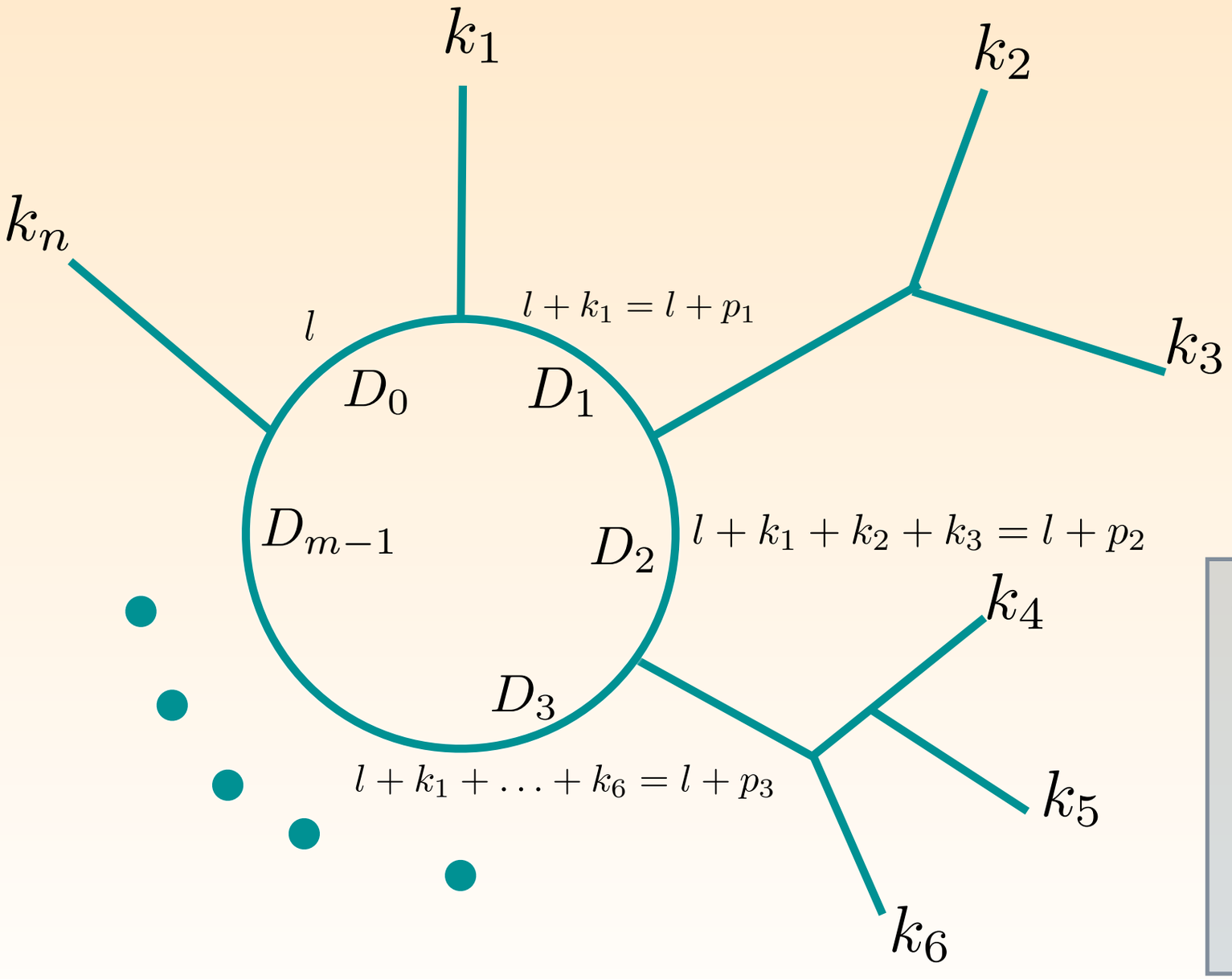
Obstacles



- Let us focus on NLO… there are already enough steps to be taken:
  - Virtual amplitudes: how to compute the loops automatically in a reasonable amount of time
  - How to deal with infra-red divergences: virtual corrections and real-emission corrections are separately divergent and only their sum is finite (for IR-safe observables) according to the KLN theorem
  - How to match these processes to a parton shower without double counting

VIRTUAL CORRECTIONS

ONE-LOOP INTEGRAL



- Consider this m-point loop diagram with n external momenta

- The integral to compute is

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}}$$

$$D_i = (l + p_i)^2 - m_i^2$$

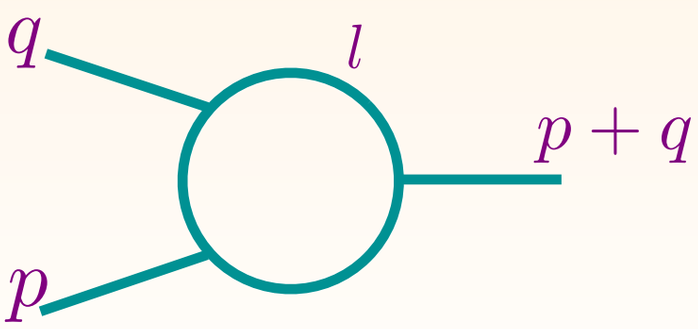
Standard Approach

- Passarino-Veltman reduction:

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}} \rightarrow \sum_i \text{coeff}_i \int d^d l \, \frac{1}{D_0 D_1 \cdots}$$

  - Reduce a general integral to “scalar integrals” by “completing the square”

- Let’s do an example:
  Suppose we want to calculate this triangle integral



$$\int \frac{d^d l}{(2\pi)^d} \frac{l^\mu}{l^2 \, (l+p)^2 \, (l+q)^2}$$

$$\int \frac{d^d l}{(2\pi)^d} \frac{l^\mu}{l^2\,(l+p)^2\,(l+q)^2}$$

- The only independent four vectors are p^μ and q^μ . Therefore, the integral must be proportional to those. We can set-up a system of linear equations and try to solve for C_1 and C_2

$$\int \frac{d^d l}{(2\pi)^d} \frac{l^\mu}{l^2\,(l+p)^2\,(l+q)^2} = \left(\begin{array}{cc} p^\mu & q^\mu \end{array} \right) \left(\begin{array}{c} C_1 \\ C_2 \end{array} \right)$$

- We can solve for C_1 and C_2 by contracting with p and q

$$\left(\begin{array}{c} R_1 \\ R_2 \end{array} \right) = \left(\begin{array}{c} [2l\cdot p] \\ [2l\cdot q] \end{array} \right) = G \left(\begin{array}{c} C_1 \\ C_2 \end{array} \right) \equiv \left(\begin{array}{cc} 2p\cdot p & 2p\cdot q \\ 2p\cdot q & 2q\cdot q \end{array} \right) \left(\begin{array}{c} C_1 \\ C_2 \end{array} \right)$$

where $[2l\cdot p] = \int \frac{d^d l}{(2\pi)^d} \frac{2l\cdot p}{l^2\,(l+p)^2\,(l+q)^2}$ (For simplicity, the masses are neglected here)

- By expressing $2l.p$ and $2l.q$ as a sum of denominators we can express R_1 and R_2 as a sum of simpler integrals, e.g.

$$R_1 = \int \frac{d^d l}{(2\pi)^d} \frac{2l\cdot p}{l^2(l+p)^2(l+q)^2} = \int \frac{d^d l}{(2\pi)^d} \frac{(l+p)^2 - l^2 - p^2}{l^2(l+p)^2(l+q)^2}$$

$$= \int \frac{d^d l}{(2\pi)^d} \frac{1}{l^2(l+q)^2} - \int \frac{d^d l}{(2\pi)^d} \frac{1}{(l+p)^2(l+q)^2} - p^2 \int \frac{d^d l}{(2\pi)^d} \frac{1}{l^2(l+p)^2(l+q)^2}$$

- And similarly for R_2

$$R_2 = \int \frac{d^d l}{(2\pi)^d} \frac{2l \cdot q}{l^2 (l+p)^2 (l+q)^2} = \int \frac{d^d l}{(2\pi)^d} \frac{(l+q)^2 - l^2 - q^2}{l^2 (l+p)^2 (l+q)^2}$$
$$= \int \frac{d^d l}{(2\pi)^d} \frac{1}{l^2 (l+p)^2} - \int \frac{d^d l}{(2\pi)^d} \frac{1}{(l+p)^2 (l+q)^2} - q^2 \int \frac{d^d l}{(2\pi)^d} \frac{1}{l^2 (l+p)^2 (l+q)^2}$$

- Now we can solve the equation

$$\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} [2l \cdot p] \\ [2l \cdot q] \end{pmatrix} = G \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \equiv \begin{pmatrix} 2p \cdot p & 2p \cdot q \\ 2p \cdot q & 2q \cdot q \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

by inverting the "Gram" matrix G

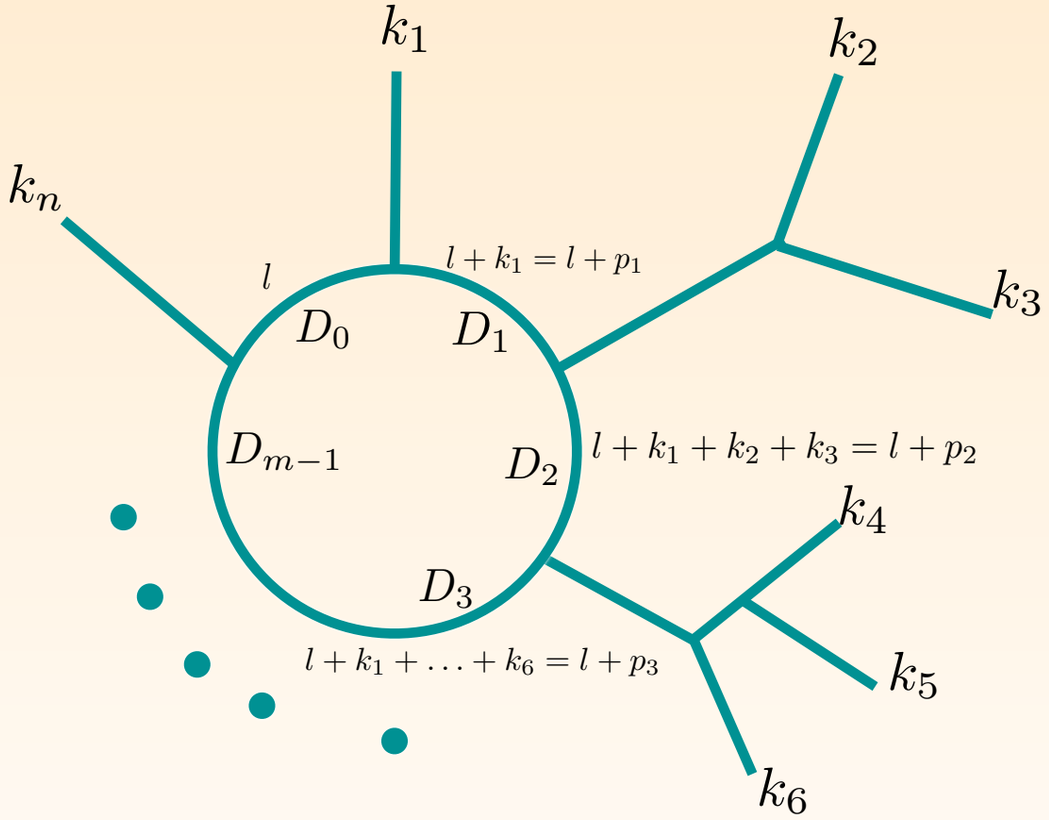
$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = G^{-1} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$$

and we have expressed our original integral

$$\int \frac{d^d l}{(2\pi)^d} \frac{l^\mu}{l^2 \, (l+p)^2 \, (l+q)^2} = \begin{pmatrix} p^\mu & q^\mu \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

in terms of known, simpler integrals and we are done!

Higher point integrals



- For loop integrals with many legs, the reduction to scalar integrals can still be performed
- Only up to 4-point scalar integrals are needed (in 4 dimensions)!
- The proof is beyond the scope of these lectures (it is straight forward by using the Van Neerven-Vermaseren basis for the loop momentum); it is related to the fact that in 4 dimensions only four 4-vectors can be linearly independent

Basis of scalar integrals

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3}\,\text{Box}_{i_0 i_1 i_2 i_3}$$
$$+ \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2}\,\text{Triangle}_{i_0 i_1 i_2}$$
$$+ \sum_{i_0<i_1} b_{i_0 i_1}\,\text{Bubble}_{i_0 i_1}$$
$$+ \sum_{i_0} a_{i_0}\,\text{Tadpole}_{i_0}$$
$$+ R + \mathcal{O}(\epsilon)$$

- The a, b, c, d and R coefficients depend only on external parameters and momenta

$$D_i = (l+p_i)^2 - m_i^2$$

$$\text{Tadpole}_{i_0} = \int d^d l \frac{1}{D_{i_0}}$$
$$\text{Bubble}_{i_0 i_1} = \int d^d l \frac{1}{D_{i_0} D_{i_1}}$$
$$\text{Triangle}_{i_0 i_1 i_2} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2}}$$
$$\text{Box}_{i_0 i_1 i_2 i_3} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

- All these scalar integrals are known and available in computer libraries (FF [v. Oldenborgh], QCDLoop [Ellis, Zanderighi], OneLOop [v. Hameren])

Divergences

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2} + \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1} + \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} + R + \mathcal{O}(\epsilon)$$

$$D_i = (l + p_i)^2 - m_i^2$$

$$\text{Tadpole}_{i_0} = \int d^d l \frac{1}{D_{i_0}}$$

$$\text{Bubble}_{i_0 i_1} = \int d^d l \frac{1}{D_{i_0} D_{i_1}}$$

$$\text{Triangle}_{i_0 i_1 i_2} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2}}$$

$$\text{Box}_{i_0 i_1 i_2 i_3} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

- The coefficients d, c, b and a are finite and do not contain poles in $1/\epsilon$
- The $1/\epsilon$ dependence is in the scalar integrals (and the UV renormalization)
- When we have solved this system (and included the UV renormalization) we have the full dependence on the soft/collinear divergences in terms of coefficients in front of the poles. These divergences should cancel against divergences in the real emission corrections (according to KLN theorem)

$$\text{Virtual} \sim v_0 + \frac{v_1}{\epsilon} + \frac{v_2}{\epsilon^2}$$

About the R term

- In our example the decomposition to scalar integrals was "exact", i.e. there were no left-over terms.
- This is true for most integrals. Only if the rank of the integral is

$$r \geq \max\{(N-1), 2\}$$

there are some extra contributions which are called "Rational terms" that are not proportional to a scalar integral

- They are of UV origin and come from the ϵ (dimensional regulator) dependence of the integral times a scalar integral that is UV divergent

$$\text{Rational terms } \sim \epsilon B_0(p, m_1, m_2)$$

(The Bubble scalar integrals are the only UV divergent scalar integrals)

- When taking the limit $\epsilon \to 0$, only the leading contribution remains, which are independent from the scalar integral itself

Automation

- Advantage:
  - The method above can be straight-forwardly generalized to any one-loop integral (appearing in a renormalizable theory)
- Disadvantage:
  - For relatively simple processes, the number of terms already explodes (several 100 MB of code is no exception for the matrix elements of a $2 \to 3$ process); simplifications require hard work and are difficult to do in a general way

New loop techniques

- The "loop revolution": new techniques for computing one-loop matrix elements are now established:
  - Generalized unitarity (e.g. BlackHat, Njet, ...)
    [Bern, Dixon, Dunbar, Kosower, 1994...; Ellis Giele Kunst 2007 + Melnikov 2008; Badger...]
  - Integrand reduction (OPP method) (e.g. MadLoop, GoSam)
    [Ossola, Papadopoulos, Pittau 2006; del Aguila, Pittau 2004; Mastrolia, Ossola, Reiter, Tramontano 2010;...]
  - Tensor reduction (e.g. Golem, Openloops)
    [Passarino, Veltman 1979; Denner, Dittmaier 2005; Binoth Guillet, Heinrich, Pilon, Reiter 2008; Cascioli, Maierhofer, Pozzorini 2011;...]

Personal bias

- This is a technical, complicated topic
- I've got only 1 hour, which means that I can only explain one of the methods
- Generalized unitarity works extremely well for processes with many massless partons around: e.g. W+4jets
  (also only including leading color approximation simplifies a lot here)
- The integrand reduction method scales worse with including extra light jets, but performs very well when there are massive particles around (top and bottom quarks)
- Integrand reduction is used in packages that aim for automation of NLO corrections to any SM process, which is why I focus on this method only.

At the integrand level

- Remember we only need to determine the coefficients in front of the scalar integrals
- The PV decomposition to scalar integrals presented before works at the level of the integrals
- If we would know a similar relation at the **integrand** level, we would be able to manipulate the integrands and extract the coefficients without doing the integrals
- This is exactly what the OPP reduction does
  - The decomposition is the same, except that there might be contributions that integrate to zero

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3}$$
$$+ \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2}$$
$$+ \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1}$$
$$+ \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0}$$
$$+ R + \mathcal{O}(\epsilon)$$

At the integrand level

- Consider, e.g., the Box coefficient:

$$d_{i_0 i_1 i_2 i_3}\mathrm{Box}_{i_0 i_1 i_2 i_3} = d_{i_0 i_1 i_2 i_3}\int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

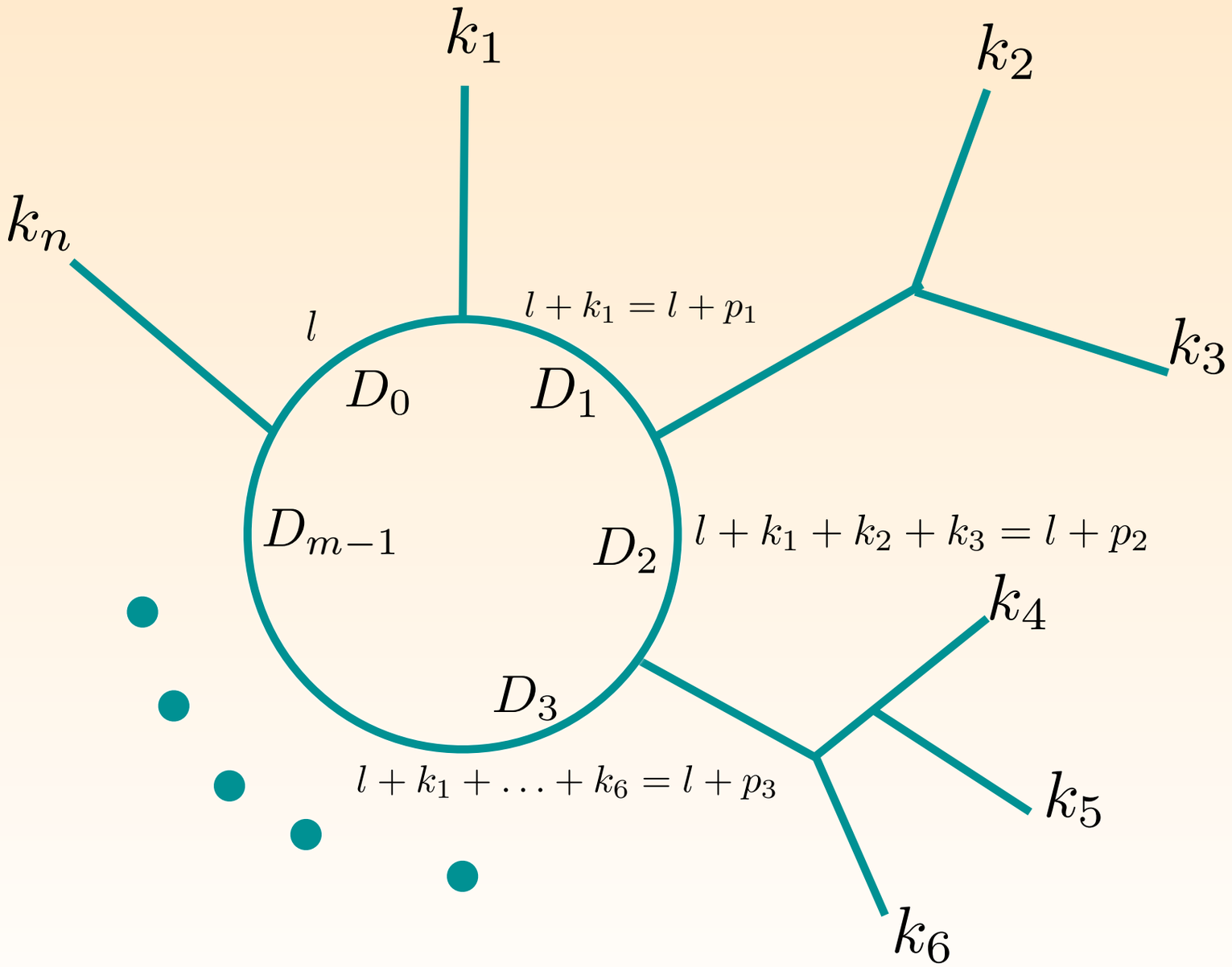
$$= \int d^d l \frac{d_{i_0 i_1 i_2 i_3}}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

$$= \int d^d l \frac{d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

$$\text{where} \quad \int d^d l \frac{\tilde{d}_{i_0 i_1 i_2 i_3}(l)}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}} = 0$$

- And similarly for the c, b, a and R terms
- The contributions that vanish when doing the integral are called "spurious terms"

ONE-LOOP INTEGRAL



- Consider this m-point loop diagram with n external momenta
- The integral to compute is

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}}$$

$$D_i = (l + p_i)^2 - m_i^2$$

OPP DECOMPOSITION

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}}$$

$$D_i = (l + p_i)^2 - m_i^2$$

- For the numerator of any integrand of a one-loop computation we can therefore write

$$\begin{aligned}
N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i \\
& + \tilde{P}(l) \prod_{i}^{m-1} D_i
\end{aligned}$$

Numerical evaluation

- By choosing specific values for the loop momentum l, we end up with a system of linear equations
  - In a renormalizable theory, the rank of the integrand is always smaller (or equal) to the number of particles in the loop (with a conveniently chosen gauge)
  - We can straight-forwardly set the it up by sampling the numerator numerically for various values of the loop momentum l
  - By choosing l smartly, the system greatly reduces
    - In particular when we chose l to be a complex 4-vector

Functional form of the spurious terms

- The functional form of the spurious terms is known (it depends on the rank of the integral and the number of propagators in the loop) [del Aguila, Pittau 2004]
  - for example, a box coefficient from a rank 1 numerator is

$$\tilde{d}_{i_0 i_1 i_2 i_3}(l) = \tilde{d}_{i_0 i_1 i_2 i_3}\, \epsilon^{\mu\nu\rho\sigma}\, l^\mu p_1^\nu p_2^\rho p_3^\sigma$$

(remember that ρ_i is the sum of the momentum that has entered the loop so far, so we always have $\rho_0 = 0$)

- The integral is zero

$$\int d^d l \frac{\tilde{d}_{i_0 i_1 i_2 i_3}(l)}{D_0 D_1 D_2 D_3} = \tilde{d}_{i_0 i_1 i_2 i_3} \int d^d l \frac{\epsilon^{\mu\nu\rho\sigma}\, l^\mu p_1^\nu p_2^\rho p_3^\sigma}{D_0 D_1 D_2 D_3} = 0$$

How it works...

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$
$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$
$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i$$
$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i$$
$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$

To solve the OPP reduction, choosing special values for the loop momenta helps a lot

For example, choosing l such that

$$D_0(l^{\pm}) = D_1(l^{\pm}) = D_2(l^{\pm}) = D_3(l^{\pm}) = 0$$

sets all the terms in this equation to zero except the first line

There are two (complex) solutions to this equation due to the quadratic nature of the propagators

How it works...

$$N(l^{\pm}) = d_{0123} + \tilde{d}_{0123}(l^{\pm}) \prod_{i \neq 0,1,2,3}^{m-1} D_i(l^{\pm})$$

- Two values are enough given the functional form for the spurious term. We can immediately determine the Box coefficient

$$d_{0123} = \frac{1}{2}\left[\frac{N(l^{+})}{\prod_{i\neq 0,1,2,3}^{m-1} D_i(l^{+})} + \frac{N(l^{-})}{\prod_{i\neq 0,1,2,3}^{m-1} D_i(l^{-})}\right]$$

- By choosing other values for l, that set other combinations of 4 "denominators" to zero, we can get all the Box coefficients

How it works...

- Now that we have all the Box coefficients we can start choosing values for *l* that set 3 "denominators" to zero to get the Triangle coefficients. Of course, now both the first and the second lines contribute.

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$
$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$

- We already have solved the coefficients of the first line in the previous iteration, so also here there is only a simple system of equations to solve
- Once we have all the Triangle coefficients, we can continue to determine the Bubble coefficients; and finally the Tadpole coefficients

How it works...

- For each phase-space point we have to solve the system of equations
- Due to the fact that the system reduces when picking special values for the loop momentum, the system greatly reduces
- We can decompose the system at the level of the squared matrix element, amplitude, diagram or anywhere in between. As long as we provide the corresponding numerator function
- For a given phase-space point, we have to compute the numerator function several times (~50 or so for a 4-point loop diagram)

Complications in d dimensions

- In the previous consideration I was very sloppy in considering if we are working in 4 or d dimensions
- In general, external momenta and polarization vectors are in 4 dimensions; only the loop momentum is in d dimensions
- To be more correct, we compute the integral

$$\int d^d l \, \frac{N(l,\tilde{l})}{\bar{D}_0 \bar{D}_1 \bar{D}_2 \cdots \bar{D}_{m-1}}$$

$$\bar{l} = l + \tilde{l}$$

d dim 4 dim epsilon dim

$$\bar{D}_i = (\bar{l} + p_i)^2 - m_i^2 = (l + p_i)^2 - m_i^2 + \tilde{l}^2 = D_i + \tilde{l}^2$$

$$l \cdot \tilde{l} = 0 \qquad \bar{l} \cdot p_i = l \cdot p_i \qquad \bar{l} \cdot \bar{l} = l \cdot l + \tilde{l} \cdot \tilde{l}$$

Implications

$$\sum_{0\le i_0<i_1<i_2<i_3}^{m-1} d(i_0i_1i_2i_3)\int d^d\bar{\ell}\,\frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}\bar{D}_{i_3}}$$

$$+\sum_{0\le i_0<i_1<i_2}^{m-1} c(i_0i_1i_2)\int d^d\bar{\ell}\,\frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}}$$

$$+\sum_{0\le i_0<i_1}^{m-1} b(i_0i_1)\int d^d\bar{\ell}\,\frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}}$$

$$+\sum_{i_0=0}^{m-1} a(i_0)\int d^d\bar{\ell}\,\frac{1}{\bar{D}_{i_0}}$$

$$+\,R\,.$$

- The decomposition in terms of scalar integrals has to be done in d dimensions
- This is why the rational part R is needed

Rational terms

- The main difference is how we get the rational terms (we already saw them in the Passarino-Veltman reduction)
- In the OPP method, they are split into two contributions, generally called

$$R = R_1 + R_2$$

- Both have their origin in the UV part of the model, but only R_1 can be directly computed in the OPP reduction

R_1

- The origin of R_1 is coming is the denominators of the propagators in the loop

$$\frac{1}{D_i} \to \frac{1}{\bar{D}_i} = \frac{1}{D}\left(1 - \frac{\tilde{l}^2}{D_i}\right)$$

- Of course, the propagator structure is known, so these contributions can be included in the OPP reduction

- They give contributions proportional to

$$\int d^d\bar{l} \frac{\tilde{l}^2}{\bar{D}_i \bar{D}_j} = -\frac{i\pi^2}{2}\left[m_i^2 + m_j^2 - \frac{(p_i - p_j)^2}{3}\right] + \mathcal{O}(\epsilon)$$

$$\int d^d\bar{l} \frac{\tilde{l}^2}{\bar{D}_i \bar{D}_j \bar{D}_k} = -\frac{i\pi^2}{2} + \mathcal{O}(\epsilon)$$

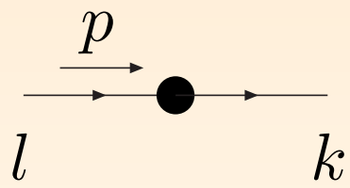
$$\int d^d\bar{l} \frac{\tilde{l}^4}{\bar{D}_i \bar{D}_j \bar{D}_k \bar{D}_l} = -\frac{i\pi^2}{6} + \mathcal{O}(\epsilon)$$

R_2

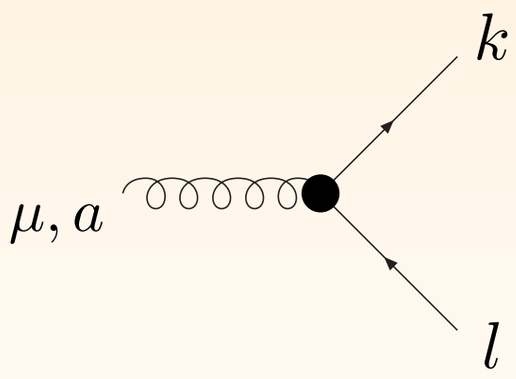
- The other origin of rational terms is the numerator itself. For integrals with rank > 2 we can have dependence in the numerator that is proportional to $\bar{l}^2$
- Unfortunately, this dependence can be quite hidden; maybe it is only explicitly there after doing the Clifford algebra
- Because we want to solve the system without doing this algebra analytically (we want to solve it numerically) we cannot get these contributions directly within the OPP reduction
- Within a given model, there is only a finite number of sources that can give these contributions; They have all been identified within the SM, and can be computed with the "R_2 counter terms"

R_2 Feynman rules

- Given that the R_2 contributions are of UV origin, only up to 4-point functions contribute to it (in a renormalizable theory)
- They can be computed using special Feynman rules, similarly to the UV counter term Feynman rules needed for the UV renormalization, e.g.



$$= \frac{ig^2}{16\pi^2}\frac{N_{col}^2-1}{2N_{col}}\delta_{kl}(-\not{p}+2m_q)\,\lambda_{HV}$$



$$= \frac{ig^3}{16\pi^2}\frac{N_{col}^2-1}{2N_{col}}\,t^a_{kl}\gamma_\mu\,(1+\lambda_{HV})$$

[Draggiotis, Garzelli, Papadopoulos, Pittau]

- Unfortunately these Feynman rules are model dependent, which means the need to be explicitly computed when going to BSM (just like the UV renormalisation)

Summary: Beyond Passarino-Veltman

- In PV reduction, we need analytic expressions for all the integrals. Possible to automate, but in practice too many terms which are difficult to simplify
- In OPP reduction we reduce the system at the **integrand** level.
  - We can solve the system numerically: we only need a numerical function of the (numerator of) integrand. We can set-up a system of linear equations by choosing specific values for the loop momentum l, depending on the kinematics of the event
  - OPP reduction is implemented in CutTools and GoSam (both publicly available). Given the integrand, they provide all the coefficients in front of the scalar integrals and the R_1 term
  - The OPP reduction leads to numerical unstabilities whose origins are related to the inverse Gram determinants.
  - Analytic information is needed for the R_2 term, but can be compute once and for all for any given BSM model

Conclusions

- NLO calculations have been completely automated and are (almost) as easy to run as LO
- Also the matching to the parton shower that allows for event generation at NLO is automated
- This means that at zero extra (human) effort NLO event samples can be generated instead of LO ones
  - No good reason anymore to use only LO in your analysis