



NEXT-TO-LEADING ORDER

LECTURE II

Rikkert Frederix
University of Zurich

PERTURBATIVE EXPANSION

$\hat{\sigma}_{ab \rightarrow X}(\hat{S}, \mu_F, \mu_R)$ Parton-level cross section

- ✱ The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter

$$\hat{\sigma} = \sigma^{\text{Born}} \left(1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \left(\frac{\alpha_s}{2\pi} \right)^2 \sigma^{(2)} + \left(\frac{\alpha_s}{2\pi} \right)^3 \sigma^{(3)} + \dots \right)$$

LO
predictions

NLO
corrections

NNLO
corrections

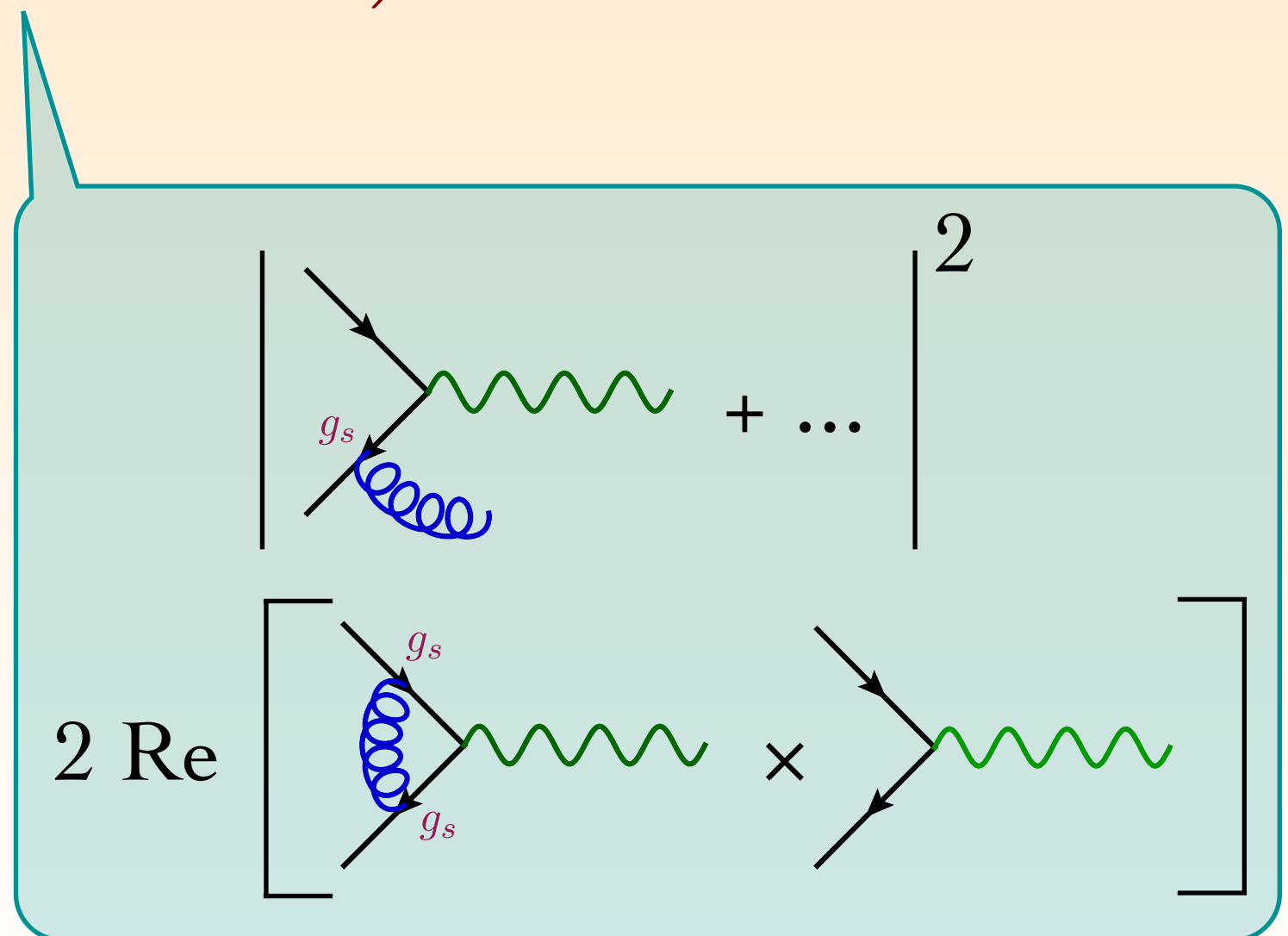
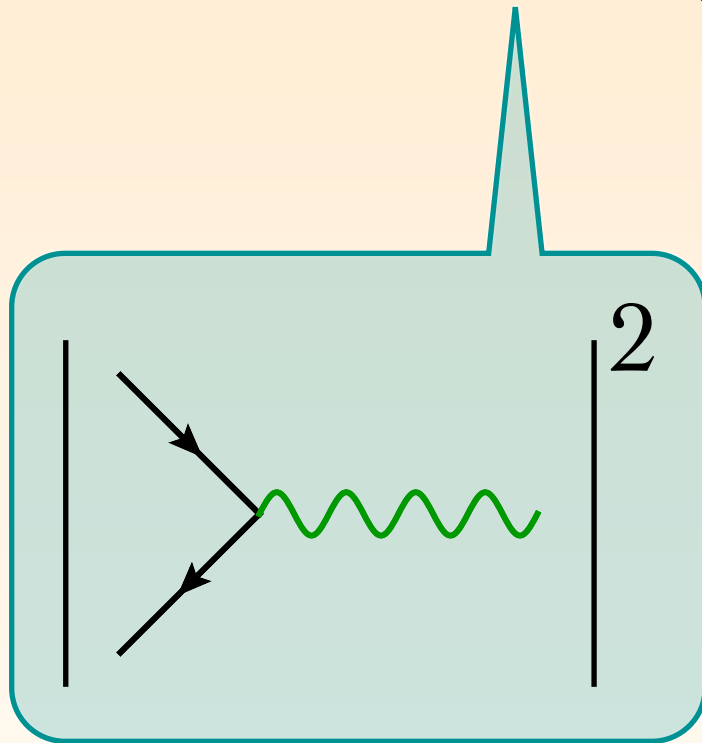
NNNLO
corrections

- ✱ Including higher corrections improves predictions and reduces theoretical uncertainties

NLO PREDICTIONS

✦ As an example, consider Drell-Yan production

$$\hat{\sigma} = \sigma^{\text{Born}} \left(1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \dots \right)$$



STANDARD APPROACH

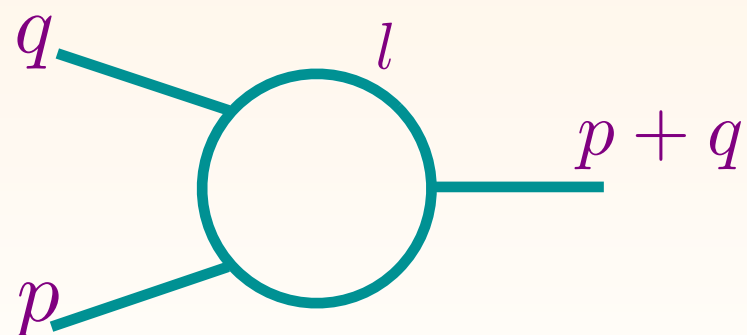
✱ Passarino-Veltman reduction:

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

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

✱ Let’s do an example:

Suppose we want to calculate this triangle integral



$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)}$$

BASIS OF SCALAR INTEGRALS

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

✱ 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], QCDDLoop [Ellis, Zanderighi], OneLOop [v. Hameren])

DIVERGENCES

$$\begin{aligned}
 \mathcal{M}^{1\text{-loop}} = & \sum_{i_0 < i_1 < i_2 < i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} & D_i = (l + p_i)^2 - m_i^2 \\
 & + \sum_{i_0 < i_1 < i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2} & \text{Tadpole}_{i_0} = \int d^d l \frac{1}{D_{i_0}} \\
 & + \sum_{i_0 < i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1} & \text{Bubble}_{i_0 i_1} = \int d^d l \frac{1}{D_{i_0} D_{i_1}} \\
 & + \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} & \text{Triangle}_{i_0 i_1 i_2} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2}} \\
 & + R + \mathcal{O}(\epsilon) & \text{Box}_{i_0 i_1 i_2 i_3} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}
 \end{aligned}$$

- ✱ The coefficients **d**, **c**, **b** and **a** are finite and do not contain poles in $1/\epsilon$ and are finite
- ✱ 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)

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

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 $2 \rightarrow 3$ process); simplifications require hard work and are difficult to do in a general way
- ☀ Does only work when the integrals are known analytically



THE “NLO REVOLUTION”

One indicator of NLO progress

$pp \rightarrow W + 0 \text{ jet}$	1978	Altarelli, Ellis, Martinelli
$pp \rightarrow W + 1 \text{ jet}$	1989	Arnold, Ellis, Reno
$pp \rightarrow W + 2 \text{ jets}$	2002	Campbell, Ellis
$pp \rightarrow W + 3 \text{ jets}$	2009	BH+Sherpa Ellis, Melnikov, Zanderighi
$pp \rightarrow W + 4 \text{ jets}$	2010	BH+Sherpa

Slide from Lance Dixon

NEW LOOP TECHNIQUES

- ✿ The “loop revolution”: new techniques for computing one-loop matrix elements are now established:
 - ✿ Generalized unitarity (e.g. BlackHat, Rocket, ...)
[Bern, Dixon, Dunbar, Kosower, 1994...; Ellis Giele Kunst 2007 + Melnikov 2008;...]
 - ✿ Integrand reduction (e.g. CutTools, Sumarai)
[Ossola, Papadopoulos, Pittau 2006; del Aguila, Pittau 2004; Mastrolia, Ossola, Reiter, Tramontano 2010;...]
 - ✿ Tensor reduction (e.g. Golem)
[Passarino, Veltman 1979; Denner, Dittmaier 2005; Binoth Guillet, Heinrich, Pilon, Reiter 2008]

INTEGRAND REDUCTION

- ✱ Any one-loop integral can be decomposed in scalar integrals
- ✱ The task is to find these coefficients efficiently (analytically or numerically)
- ✱ The integrand (or OPP [Ossola, Papadopoulos, Pittau 2006]) reduction method is a method that has been automated in the **CutTools** program to find these coefficients in an automated way
- ✱ The integrand reduction technique is what we have adopted to use in MadGraph to compute the loop diagrams

AT THE INTEGRAND LEVEL

✱ The 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

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

AT THE INTEGRAND LEVEL

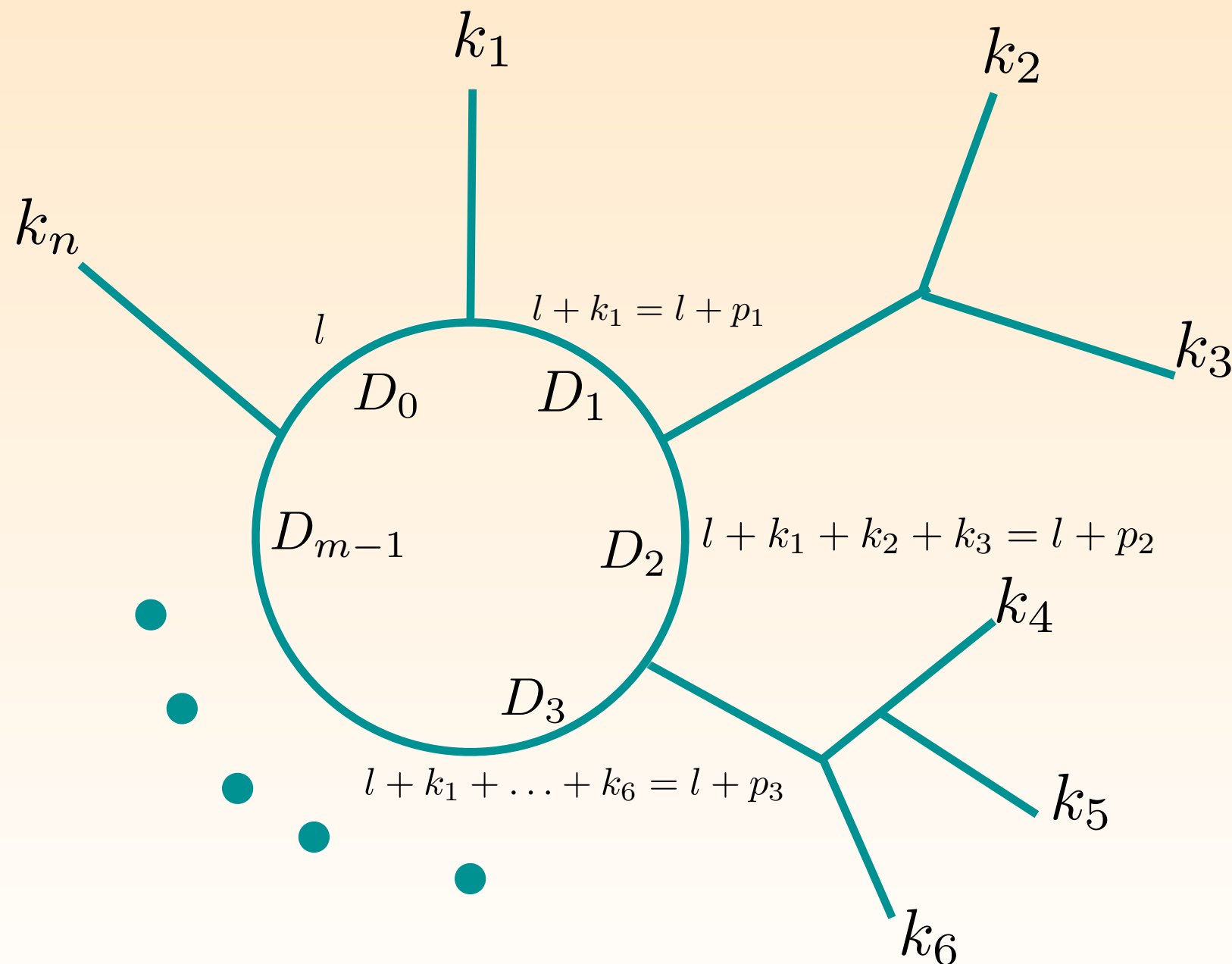
- ✱ Consider, e.g., the Box coefficient:

$$\begin{aligned}d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} &= d_{i_0 i_1 i_2 i_3} \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}} \\ &= \int d^d l \frac{d_{i_0 i_1 i_2 i_3}}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}} \\ &= \int d^d l \frac{d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}\end{aligned}$$

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

- ✱ 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

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

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

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

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

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 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 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 p_i is the sum of the momentum that has entered the loop so far, so we always have $p_0 = 0$)

- ✱ 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...

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

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.

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

- 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 amplitude, diagram or in between. As long as we provide the corresponding numerator function. In MadGraph we decompose diagram by diagram, but we are considering improvements
- ✱ For a given phase-space point, we have to compute the numerator function several times (~ 50 or so for a $2 \rightarrow 3$ process)

A CLASSICAL EXAMPLE

✱ Suppose we want to compute this integral $\int d^d l \frac{1}{D_0 D_1 D_2 D_3 D_4 D_5 D_6}$

✱ So we that the numerator is $N(l) = 1$ $D_i = (l + p_i)^2 - m_i^2$

✱ We know that we need only Box, Triangle, Bubble (and Tadpole) contributions. Let's find the first Box integral coefficient.

✱ Take the two solutions of

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

✱ And use the relation we found before and we directly have

$$d_{0123} = \frac{1}{2} \left[\frac{1}{D_4(l^+) D_5(l^+) D_6(l^+)} + \frac{1}{D_4(l^-) D_5(l^-) D_6(l^-)} \right]$$

...

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

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

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

\nearrow
d dim

\nearrow
4 dim

\nearrow
epsilon dim

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

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

IMPLICATIONS

$$\begin{aligned}
 & \sum_{0 \leq i_0 < i_1 < i_2 < i_3}^{m-1} d(i_0 i_1 i_2 i_3) \int d^d \bar{\ell} \frac{1}{\bar{D}_{i_0} \bar{D}_{i_1} \bar{D}_{i_2} \bar{D}_{i_3}} \\
 & + \sum_{0 \leq i_0 < i_1 < i_2}^{m-1} c(i_0 i_1 i_2) \int d^d \bar{\ell} \frac{1}{\bar{D}_{i_0} \bar{D}_{i_1} \bar{D}_{i_2}} \\
 & + \sum_{0 \leq i_0 < i_1}^{m-1} b(i_0 i_1) \int d^d \bar{\ell} \frac{1}{\bar{D}_{i_0} \bar{D}_{i_1}} \\
 & + \sum_{i_0=0}^{m-1} a(i_0) \int d^d \bar{\ell} \frac{1}{\bar{D}_{i_0}} \\
 & + R.
 \end{aligned}$$

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

RATIONAL TERMS

- ✿ The main difference is that 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 and is given by the CutTools program

R₁

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

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

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

- ✱ They give contributions proportional to

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

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

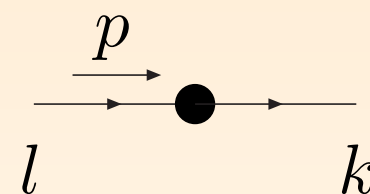
$$\int d^d \bar{l} \frac{\tilde{l}^4}{\bar{D}_i \bar{D}_j \bar{D}_k \bar{D}_l} = -\frac{i\pi^2}{6} + \mathcal{O}(\epsilon)$$

R₂

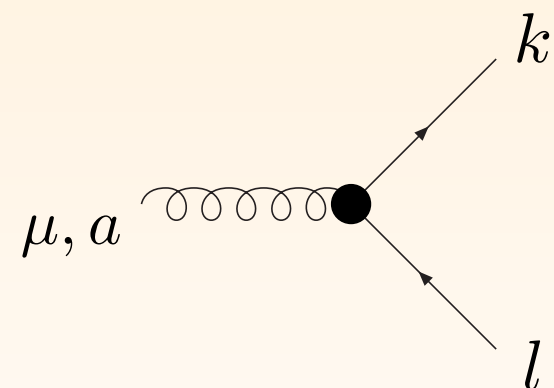
- ✱ The other origin of rational terms is the numerator itself. For integrals with rank > 2 we can have dependence in the numerator that is proportional to \vec{l}^2
- ✱ Unfortunately, this dependence can be quite hidden; maybe it is only explicitly there after doing the Clifford algebra
- ✱ Because 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 “R₂ 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.



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



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

[Draggiotis, Garzelli, Papadopoulos, Pittau]

- Unfortunately these Feynman rules are model dependent.
 \Rightarrow Maybe we can use FeynRules+FeynArts to compute them for any model?

IMPROVEMENT OVER 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 😊
 - ✱ Choosing l such that internal propagators go on-shell, enormously simplifies the resulting system 😊
 - ✱ OPP reduction is implemented in CutTools (publicly available). Given the integrand, CutTools provides all the coefficients in front of the scalar integrals and the R_1 term 😊
 - ✱ Analytic information is needed for the R_2 term, but can be compute once and for all for a given model 😞 😊

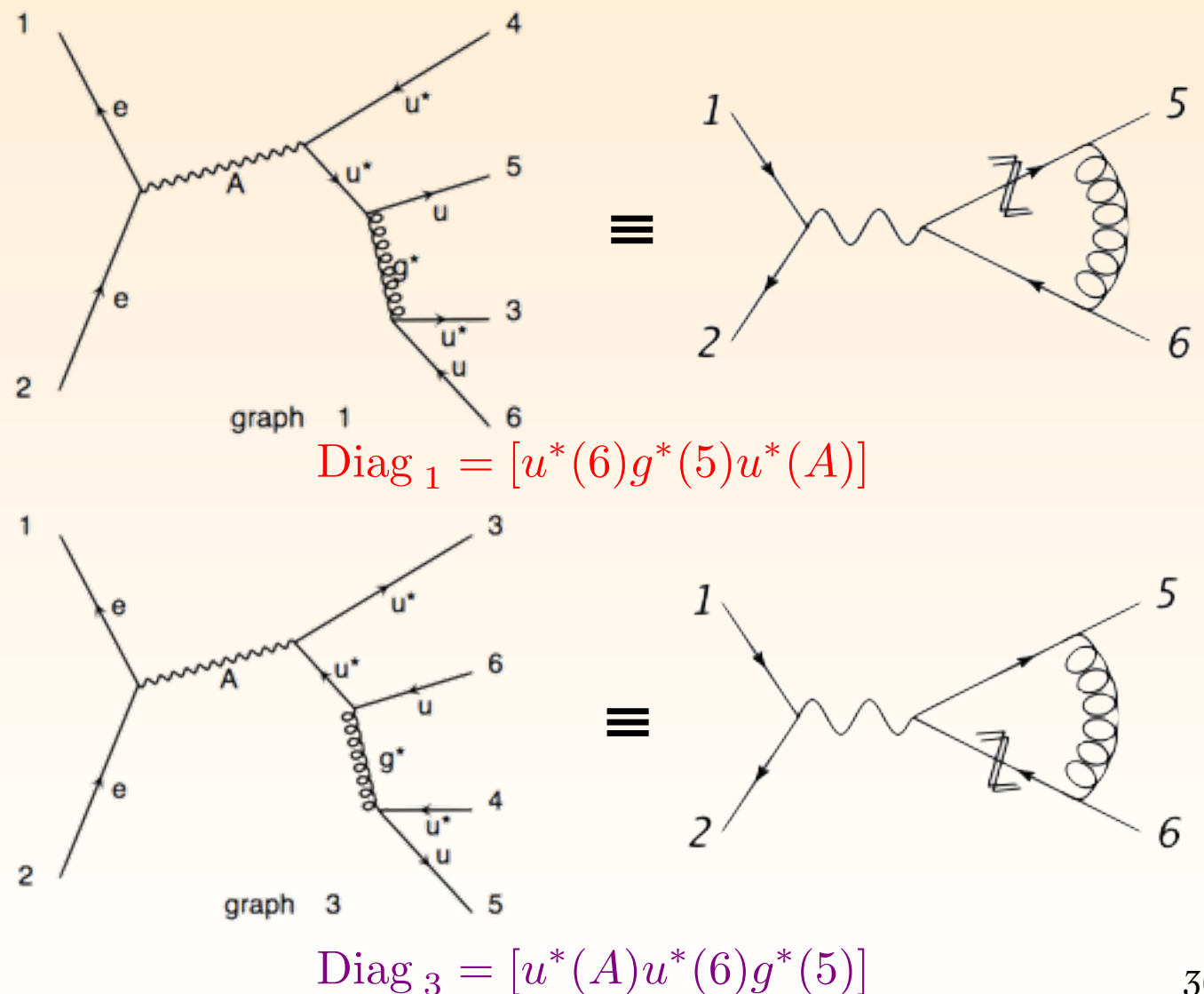
IN MADGRAPH

- ✿ MadGraph is very good at giving numerical expressions for matrix elements. Exactly what is needed by CutTools to get the coefficients of the scalar integrals
- ✿ However, it is only tree-level...
- ✿ Needed to create an add-on to MadGraph to generate loop diagrams: [MadLoop!](#)

MADLOOP

- ✱ Instead of writing a new code to generate loop diagrams, we use the existing, well-tested **MadGraph** code to generate tree-level diagrams
- ✱ A loop diagrams with the loop cut open has to extra external particles. Consider $e^+e^- \rightarrow u^* \text{ubar}^* u \text{ubar}$ (loop particles are denoted with a star). MadGraph will generate 8 L-cut diagrams. Here are two of them:

- ✱ All diagrams with two extra particles are generated and the ones that are needed are **filtered out**
- ✱ Each diagram gets an unique **tag**: any **mirror** and/or **cyclic** permutations of tags of diagrams already in the set are taken out
- ✱ Additional filter to eliminate **tadpoles** and **bubbles** attached to external lines



MADLOOP

- ✱ Several new features needed to be implemented in MadGraph
 - ✱ Recognition of the loop topologies in order to **filter L-cut diagrams**
 - ✱ Structure to deal with **two MadGraph processes simultaneously** (L-cut and Born-like)
 - ✱ Treat the **color** to obtain the correct interference between the Born and the loop diagrams
 - ✱ **Special form of the integrand** for CutTools: no propagator denominators, complex momenta and reconstruction of the missing propagator for sewed particles (e.g., when L-cut particle is a gluon, $\sum \epsilon^\mu(p) \epsilon^\nu(p) \rightarrow g^{\mu\nu}$)
 - ✱ Implementation of QCD **ghosts**
 - ✱ Implementation of the special vertices for the **rational part R_1** and the **UV renormalization**

MADLOOP: EXCEPTIONAL PS POINTS

- ✱ There are (almost) always phase-space points for which the numerical reduction to determine the coefficients in front of the scalar integrals does not work due to **numerical instabilities**
- ✱ CutTools has **build-in routines** to determine if a phase-space point is **exceptional or not**
 - ✱ CT can ask MadLoop to **evaluate the integrand at a given loop momentum** and check if the result is close enough to the one from the reconstructed integrand
 - ✱ By sending $m_i^2 \rightarrow m_i^2 + M^2$ CT has an **independent reconstruction of the numerator** and can check if both match
- ✱ Using **quadruple precision** numerics in the reduction helps, but not always

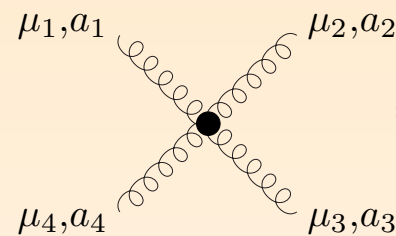
MADLOOP: EXCEPTIONAL PS POINTS

- ✱ When CutTools assigns a phase-space point to be **unstable**, MadLoop tries to cure it
 - ✱ Check if the **Ward Identity** holds at a satisfactory level
 - ✱ Shift the phase-space point by **rescaling** one of the components of the **3-momenta** (for all particles), e.g. $k_i^3 = (1 + \lambda_{\pm})k_i^3$, and adjusting the energy components to keep the point on-shell
 - ✱ Provide an **estimate of the virtual** of the original phase-space point (**with uncertainty**) $V_{\lambda=0}^{FIN} = |\mathcal{A}_{\lambda=0}^{born}|^2 (c \pm \Delta)$ where

$$c = \frac{1}{2} \left(v_{\lambda_+}^{FIN} + v_{\lambda_-}^{FIN} \right) \quad \Delta = \left| v_{\lambda_+}^{FIN} - v_{\lambda_-}^{FIN} \right| \quad v_{\lambda_{\pm}}^{FIN} = \frac{V_{\lambda_{\pm}}^{FIN}}{|\mathcal{A}_{\lambda=0}^{born}|^2}$$
- ✱ If all shifts fail (very rarely) use the **median** of the results of the last 100 stable points and the **median absolute deviation** (MAD (!)) to determine the associated uncertainty

MADLOOP: LIMITATIONS

- Of course, there are some **limitations** on what the code cannot do yet...
- No **four-gluon vertex** at the Born level: the special vertex to compute the remainder is too complicated to implement in **MadGraph v4**



$$\begin{aligned}
 &= -\frac{ig^4 N_{col}}{96\pi^2} \sum_{P(234)} \left\{ \left[\frac{\delta_{a_1 a_2} \delta_{a_3 a_4} + \delta_{a_1 a_3} \delta_{a_4 a_2} + \delta_{a_1 a_4} \delta_{a_2 a_3}}{N_{col}} \right. \right. \\
 &\quad \left. \left. + 4 \text{Tr}(t^{a_1} t^{a_3} t^{a_2} t^{a_4} + t^{a_1} t^{a_4} t^{a_2} t^{a_3}) (3 + \lambda_{HV}) \right. \right. \\
 &\quad \left. \left. - \text{Tr}(\{t^{a_1} t^{a_2}\} \{t^{a_3} t^{a_4}\}) (5 + 2\lambda_{HV}) \right] g_{\mu_1 \mu_2} g_{\mu_3 \mu_4} \right. \\
 &\quad \left. + 12 \frac{N_f}{N_{col}} \text{Tr}(t^{a_1} t^{a_2} t^{a_3} t^{a_4}) \left(\frac{5}{3} g_{\mu_1 \mu_3} g_{\mu_2 \mu_4} - g_{\mu_1 \mu_2} g_{\mu_3 \mu_4} - g_{\mu_2 \mu_3} g_{\mu_1 \mu_4} \right) \right\}
 \end{aligned}$$

- If **EW bosons appear in the loops**, the reduction by CutTools might not work because we use gauge a physical gauge (rank of diagrams can become too large)
- No **finite-width effects** for massive particles also appearing in the loops
- All Born contributions must **factorize the same power of all coupling orders**



ON-GOING IMPROVEMENTS

- ✱ The MadLoop code is being rewritten in [MadGraph v5](#). This will:
 - ✱ **remove the limitations** presented on the previous slide
 - ✱ make it faster:
 - ✱ **Recycling of tree-structures** attached to the loops
 - ✱ Identify **identical contributions** (e.g. massless fermion loops of different flavors)
 - ✱ Call CutTools not per diagram, but per **set of diagrams with the same loop kinematics**
 - ✱ Use **recursion** relations (will mostly help the real-emission corrections)
 - ✱ allow for the automatic generation of **UV renormalization** and **remainder** vertices using **FeynRules** [*Christensen, Duhr et al.*] for BSM physics

LOCAL CHECKS

$u\bar{u} \rightarrow W^+W^-b\bar{b}$	MADLOOP	Ref. [33]
a_0	2.338047209268890E-008	2.338047130649064E-008
c_{-2}	-2.493920703542680E-007	-2.493916939359002E-007
c_{-1}	-4.885901939046758E-007	-4.885901774740355E-007
c_0	-2.775800623041098E-007	-2.775787767591390E-007

$gg \rightarrow W^+W^-b\bar{b}$	MADLOOP	Ref. [33]
a_0	1.549795815702494E-008	1.549794572435312E-008
c_{-2}	-2.686312747217639E-007	-2.686310592221201E-007
c_{-1}	-6.078687041491385E-007	-6.078682316434646E-007
c_0	-5.519004042667462E-007	-5.519004727276688E-007

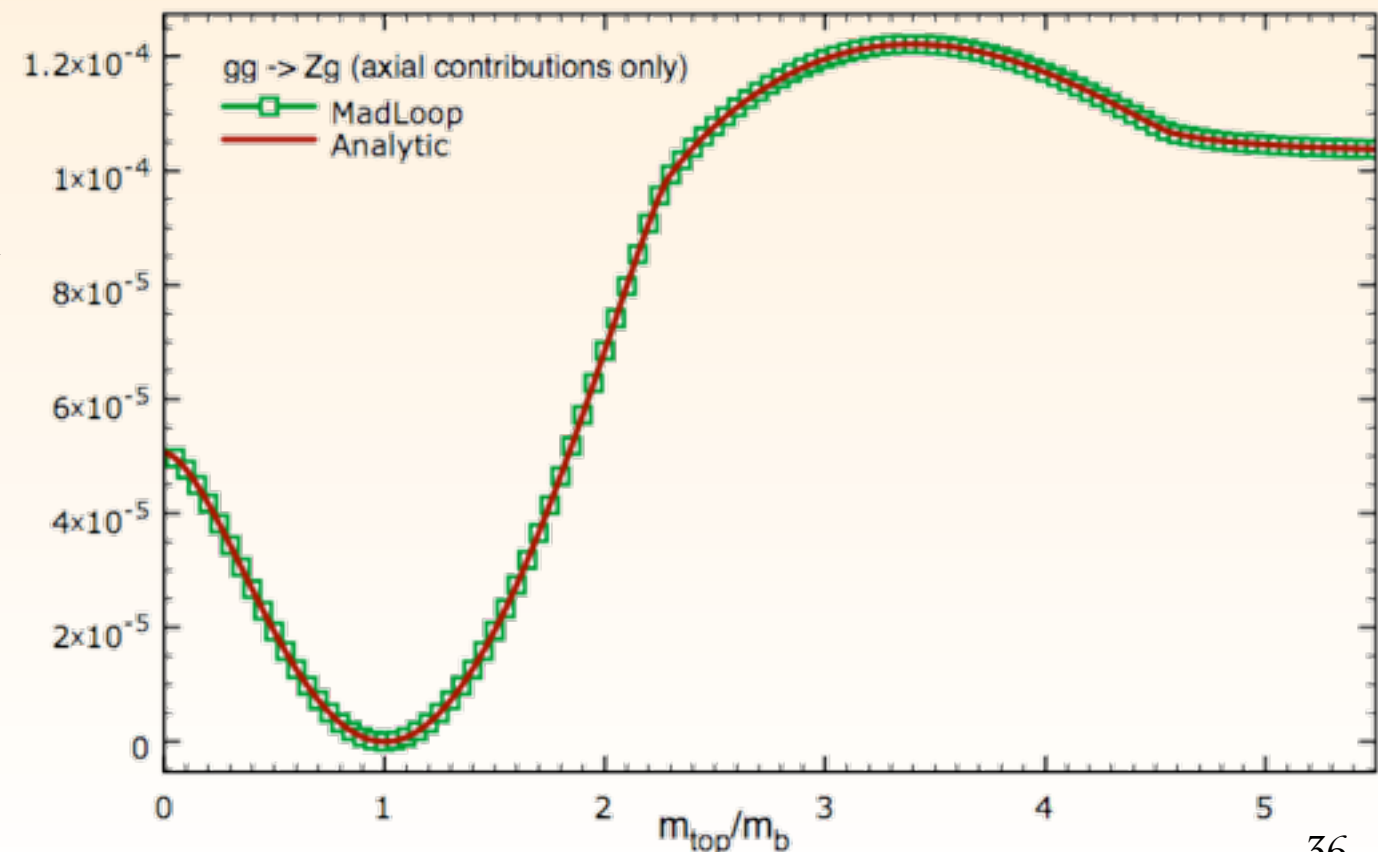
Ref. [33]: *A. van Hameren et al. arXiv:0903.4665*

The numerics are **pin-point** on analytical data, even with **several mass scales**.

Analytic computation via an implementation of the formulae found in a paper by *J.J. van der Bij e³ N. Glover*

~25 processes checked against known results (24 pages appendix of MadLoop paper, arXiv:1103.0621)

We believe the code is **very robust** - e.g., MadLoop helped **to find mistakes** in published NLO computations implementations ($pp \rightarrow Zjj$, $pp \rightarrow W^+W^+jj$)





INTEGRATED RESULTS

- ✱ Errors are the MC integration uncertainty only
- ✱ Cuts on jets, γ^*/Z decay products and photons, but **no cuts on b quarks** (their mass regulates the IR singularities)
- ✱ Efficient handling of **exceptional phase-space points**: their uncertainty always at least two orders of magnitude smaller than the integration uncertainty
- ✱ Running time: **two weeks on ~150 node cluster** leading to rather small integration uncertainties
- ✱ MadFKS+MadLoop results are fully **differential** in the final states (but only parton-level)

Process	μ	n_{lf}	Cross section (pb)	
			LO	NLO
a.1 $pp \rightarrow t\bar{t}$	m_{top}	5	123.76 ± 0.05	162.08 ± 0.12
a.2 $pp \rightarrow tj$	m_{top}	5	34.78 ± 0.03	41.03 ± 0.07
a.3 $pp \rightarrow tjj$	m_{top}	5	11.851 ± 0.006	13.71 ± 0.02
a.4 $pp \rightarrow t\bar{b}j$	$m_{top}/4$	4	25.62 ± 0.01	30.96 ± 0.06
a.5 $pp \rightarrow t\bar{b}jj$	$m_{top}/4$	4	8.195 ± 0.002	8.91 ± 0.01
b.1 $pp \rightarrow (W^+ \rightarrow)e^+\nu_e$	m_W	5	5072.5 ± 2.9	6146.2 ± 9.8
b.2 $pp \rightarrow (W^+ \rightarrow)e^+\nu_e j$	m_W	5	828.4 ± 0.8	1065.3 ± 1.8
b.3 $pp \rightarrow (W^+ \rightarrow)e^+\nu_e jj$	m_W	5	298.8 ± 0.4	300.3 ± 0.6
b.4 $pp \rightarrow (\gamma^*/Z \rightarrow)e^+e^-$	m_Z	5	1007.0 ± 0.1	1170.0 ± 2.4
b.5 $pp \rightarrow (\gamma^*/Z \rightarrow)e^+e^- j$	m_Z	5	156.11 ± 0.03	203.0 ± 0.2
b.6 $pp \rightarrow (\gamma^*/Z \rightarrow)e^+e^- jj$	m_Z	5	54.24 ± 0.02	56.69 ± 0.07
c.1 $pp \rightarrow (W^+ \rightarrow)e^+\nu_e b\bar{b}$	$m_W + 2m_b$	4	11.557 ± 0.005	22.95 ± 0.07
c.2 $pp \rightarrow (W^+ \rightarrow)e^+\nu_e t\bar{t}$	$m_W + 2m_{top}$	5	0.009415 ± 0.000003	0.01159 ± 0.00001
c.3 $pp \rightarrow (\gamma^*/Z \rightarrow)e^+e^- b\bar{b}$	$m_Z + 2m_b$	4	9.459 ± 0.004	15.31 ± 0.03
c.4 $pp \rightarrow (\gamma^*/Z \rightarrow)e^+e^- t\bar{t}$	$m_Z + 2m_{top}$	5	0.0035131 ± 0.0000004	0.004876 ± 0.000002
c.5 $pp \rightarrow \gamma t\bar{t}$	$2m_{top}$	5	0.2906 ± 0.0001	0.4169 ± 0.0003
d.1 $pp \rightarrow W^+W^-$	$2m_W$	4	29.976 ± 0.004	43.92 ± 0.03
d.2 $pp \rightarrow W^+W^- j$	$2m_W$	4	11.613 ± 0.002	15.174 ± 0.008
d.3 $pp \rightarrow W^+W^+ jj$	$2m_W$	4	0.07048 ± 0.00004	0.1377 ± 0.0005
e.1 $pp \rightarrow HW^+$	$m_W + m_H$	5	0.3428 ± 0.0003	0.4455 ± 0.0003
e.2 $pp \rightarrow HW^+ j$	$m_W + m_H$	5	0.1223 ± 0.0001	0.1501 ± 0.0002
e.3 $pp \rightarrow HZ$	$m_Z + m_H$	5	0.2781 ± 0.0001	0.3659 ± 0.0002
e.4 $pp \rightarrow HZ j$	$m_Z + m_H$	5	0.0988 ± 0.0001	0.1237 ± 0.0001
e.5 $pp \rightarrow Ht\bar{t}$	$m_{top} + m_H$	5	0.08896 ± 0.00001	0.09869 ± 0.00003
e.6 $pp \rightarrow Hb\bar{b}$	$m_b + m_H$	4	0.16510 ± 0.00009	0.2099 ± 0.0006
e.7 $pp \rightarrow Hjj$	m_H	5	1.104 ± 0.002	1.036 ± 0.002

SUMMARY

- ✱ One-loop integrals can be written as coefficients **a**, **b**, **c** and **d** times scalar functions and a rational part ***R***
- ✱ The **traditional approach** for computing one-loop diagrams (Passarino-Veltman reduction) becomes more and more complicated and difficult to automate when the number of external particles increases
- ✱ The **OPP reduction works at the integrand level**: choosing specific values of the loop momentum results in a linear system of equations, which can be solved numerically
- ✱ MadGraph has been extended to compute loops by using the OPP reduction as implemented in the **CutTools computer code**
- ✱ **MadLoop generates loop diagrams by cutting them open**, which results in tree-level diagrams with two extra external particles
- ✱ CutTools provides the values for which the numerator should be computed numerically and solves the resulting system of equations numerically