Automated computation of scattering amplitudes

Giovanni Ossola

New York City College of Technology
City University of New York (CUNY)

ACAT 2013

15th International Workshop on Advanced Computing and Analysis Techniques in Physics Research

Institute of High Energy Physics, Beijing, May 16 - 21, 2013

Introduction

Scattering Amplitudes turn our theories into predictions:

- The Evaluation of **Scattering Amplitudes** is necessary to test our theoretical models and **compare their prediction with the experiments**
- There is a **long tradition** and **extensive literature**
- The **understanding of the structure** of Scattering Amplitudes goes in parallel with the **development of tools** for computations

Introduction

Scattering Amplitudes turn our theories into predictions:

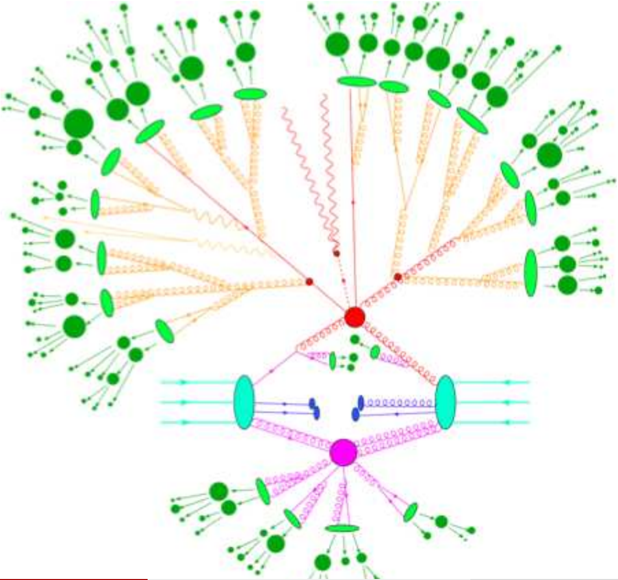
- The Evaluation of **Scattering Amplitudes** is necessary to test our theoretical models and **compare their prediction with the experiments**
- There is a **long tradition** and **extensive literature**
- The **understanding of the structure** of Scattering Amplitudes goes in parallel with the **development of tools** for computations

Outline of this talk:

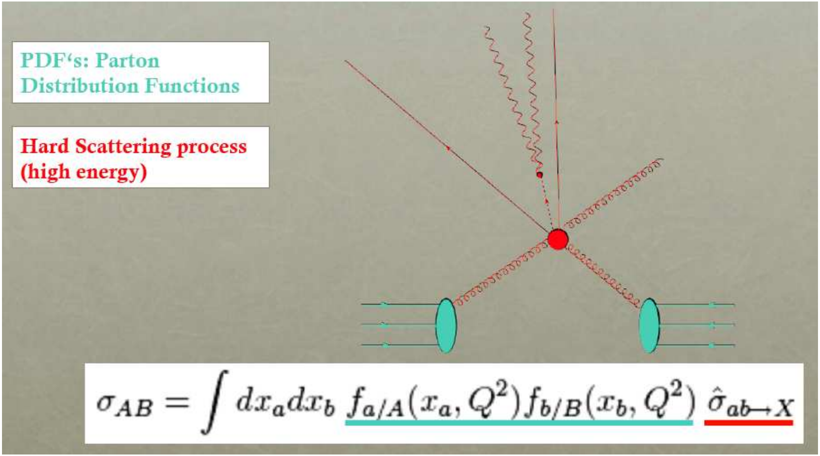
- Motivation for higher order calculations
- An overview of techniques employed for virtual one-loop amplitudes
- Can we extend what we learned at one-loop to study a more general problem?

For more details $\rightarrow$ several talks in **Track 3**

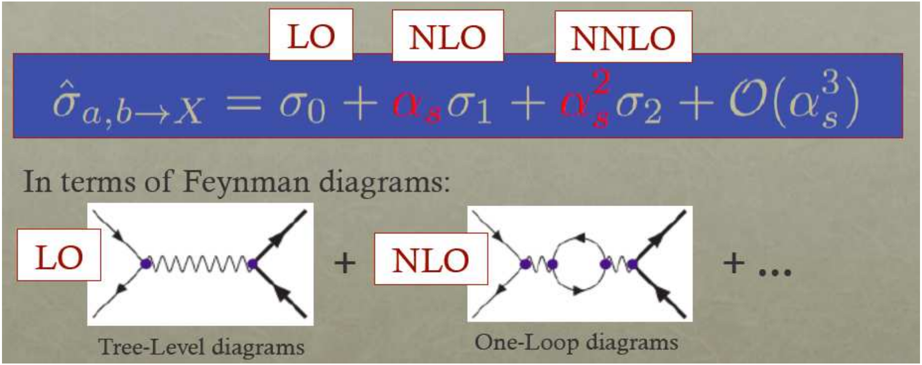
THE "BIG PICTURE"



Partons and Protons



Perturbative expansion

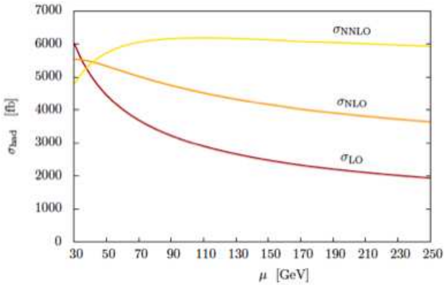


Theoretical predictions should match the needs of the experiments

For many analyses, Leading-Order (LO) predictions are not sufficient $\rightarrow$ we need higher orders in the perturbative expansion

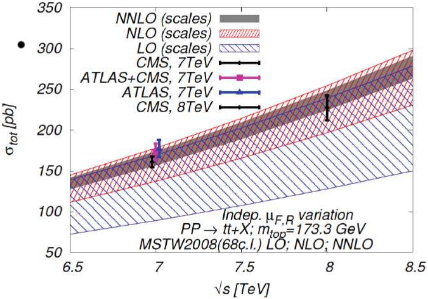
Why Higher Orders?

- Reduce the **Theoretical Error**
  - Parametric Error (input parameters: PDF, masses, couplings...)
  - Truncation Error (missing Higher Orders)
  - Estimated by looking at the Scale-Dependence



Higgs + jet at NNLO

Boughezal at al. (2013)



Top-pair production at NNLO

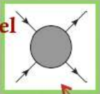
Czakon, Fiedler, Mitov (2013)

Why Higher Orders?

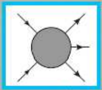
- Reduce the **Theoretical Error**
  - Parametric Error (input parameters: PDF, masses, couplings...)
  - Truncation Error (missing Higher Orders)
  - Estimated by looking at the Scale-Dependence
- Effects of Higher Order Corrections (NLO, NNLO, etc)
  - Large effects in the cross-section (in particular in QCD)
  - Changes in the shape of distributions
  - Loop-induced effects
- At the LHC:
  - Distinguish the Signal from Backgrounds
  - Provide the "theory answers" to anomalies/new physics signals

From LO to NLO

Tree-Level (Born)



Virtual Part (+1 loop)



Real Emission (Born +1 leg)

$$\sigma_{NLO} = \int_n \left(d\sigma^B + d\sigma^V + \int_1 d\sigma^A\right) + \int_{n+1} \left(d\sigma^R - d\sigma^A\right)$$

Virtual Part and Real Emission contributions are IR divergent

Subtraction terms

Needed to cancel infrared singularities numerically.
Idea: Add zero in suitable way to cancel infrared singularities from real and virtual parts.

Several **Automated** Implementations for **Tree-Level** and **Subtraction Terms**

Automated tools for Real & Subtraction

- ❑ Born & Real radiation: tree level matrix element

✓ Many tools available on the market, too many to mention all of them:

GRACE	SHERPA
HELAS/MADGRAPH/MADEVENT	HELAC/PHEGAS
WHIZARD	CompHep

....

- ❑ Subtractions: automated in several programs:

✓ MadDipole [Frederix, Greiner, Gehrmann]

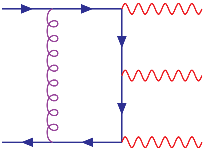
✓ AutoDipole (SuperAutoDipole) [Hasegawa, Moch, Uwer]

✓ Dipoles in SHERPA [Gleisberg, Krauss]

✓ HELAC-DIPOLES [Czakon, Papadopoulos, Worek]

In the following: we focus on the computation of the Virtual Part of NLO calculations

One-loop – Notations



Any m-point one-loop amplitude can be written, before integration, as

$$A(\bar{q}) = \frac{N(\bar{q})}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{m-1}}$$

where

$$\bar{D}_i = (\bar{q} + p_i)^2 - m_i^2 \quad , \quad \bar{q}^2 = q^2 - \mu^2 \quad , \quad \bar{D}_i = D_i - \mu^2$$

Our task is to calculate, for each phase space point:

$$\mathcal{M} = \int d^n\bar{q}\ A(\bar{q}) = \int d^n\bar{q} \frac{N(\bar{q})}{\bar{D}_0 \bar{D}_1 \ldots \bar{D}_{m-1}}$$

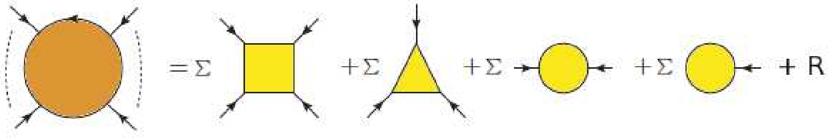
The traditional one-loop "master" formula

Passarino, Veltman (1979)

$$
\begin{aligned}
\int d^n\bar{q}\, \frac{N(\bar{q})}{\bar{D}_{i_0}\bar{D}_{i_1}\ldots\bar{D}_{m-1}} &= \sum_{i_0<i_1<i_2<i_3}^{m-1} d(i_0 i_1 i_2 i_3) \int d^n\bar{q}\, \frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}\bar{D}_{i_3}} \\
&+ \sum_{i_0<i_1<i_2}^{m-1} c(i_0 i_1 i_2) \int d^n\bar{q}\, \frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}} \\
&+ \sum_{i_0<i_1}^{m-1} b(i_0 i_1) \int d^n\bar{q}\, \frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}} \\
&+ \sum_{i_0}^{m-1} a(i_0) \int d^n\bar{q}\, \frac{1}{\bar{D}_{i_0}} \\
&+ \text{rational terms}
\end{aligned}
$$

$$\int d^n\bar{q}\,\frac{N(\bar{q})}{\bar{D}_{i_0}\bar{D}_{i_1}\ldots\bar{D}_{m-1}} = \sum_i d_i\,\mathrm{Box}_i + \sum_i c_i\,\mathrm{Triangle}_i + \sum_i b_i\,\mathrm{Bubble}_i + \sum_i a_i\,\mathrm{Tadpole}_i + \mathrm{R}\,,$$

1) **Generation**: Compute the unintegrated amplitudes for all diagrams
2) **Reduction**: Extract all coefficients and rational terms
3) **Master Integrals**: Calculate the Master Integrals (scalar integrals) and combine with the coefficients



One-Loop as a 3 step process

$$
\begin{aligned}
\int d^n\bar{q}\,\frac{N(\bar{q})}{\bar{D}_{i_0}\bar{D}_{i_1}\ldots\bar{D}_{m-1}} &= \sum_i d_i\ \mathrm{Box}_i + \sum_i c_i\ \mathrm{Triangle}_i \\
&+ \sum_i b_i\ \mathrm{Bubble}_i + \sum_i a_i\ \mathrm{Tadpole}_i + \mathrm{R}\,,
\end{aligned}
$$

1) **Generation**: Compute the unintegrated amplitudes for all diagrams
2) **Reduction**: Extract all coefficients and rational terms
3) **Master Integrals**: Calculate the Master Integrals (scalar integrals) and combine with the coefficients

There are several techniques available for Generation+Reduction and available codes to compute the one-loop Scalar Integrals

One-Loop Master Integrals: **Ellis, Zanderighi; van Oldenborgh; van Hameren; Binoth et al.; Hahn et al.**

Tensorial Reduction

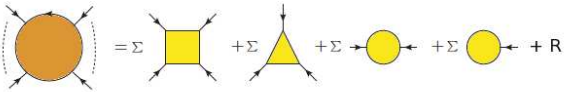
- The numerator function of any amplitude is a **polynomial** in the integration momentum q

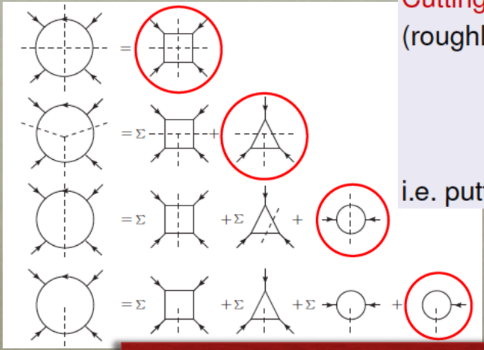
$$\mathcal{N}(\bar{q}) = \sum_{r=0}^{R} C_{\mu_1 \ldots \mu_r} \bar{q}^{\mu_1} \ldots \bar{q}^{\mu_r}$$

- Each amplitude can be decomposed in a linear combination of **kinematic factors**, which are **q-independent**, multiplied by **tensors of various ranks in q** sitting on sets of Denominators
- Tensor integrals can be written in terms of scalar integrals (or not!)
- PROS: **control over spurious singularities** (Gram); fully **algebraic**.
- Needs **efficient computer algebra** and smart book-keeping/caching

Denner, Dittmaier; Binoth et al.;
Hahn et al.; Fleisher, Riemann, Yundin.

On-Shell Methods / Unitarity Cuts





Cutting a loop propagator (roughly) means

$$\frac{1}{D_i} \rightarrow \delta(D_i)$$

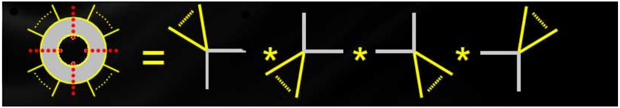
i.e. putting it on-shell

In the Loop-Integral = Generalized Unitarity Methods

In the Integrand = OPP integrand-level reduction

Generalized Unitarity

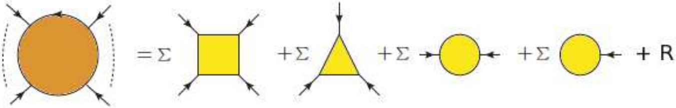
- Generate **loop momenta configurations that satisfy the cut conditions** (complex momenta)
- For each configuration, **compute and multiply the trees** at the corner of the cut diagram
- Combine the results appropriately to **get all the coefficients** of the scalar integrals



- More subtleties for the triple cut (leakage from higher point functions)
- Effectively **reduces a loop computation to tree computation** ("fusing tree amplitudes into loop amplitudes")

Bern, Dixon, Dunbar, Kosower;
Britto, Cachazo, Feng; Ellis, Giele, Kunszt, Melnikov

Rational Term **R**



Cut-Constructible part vs **Rational Term**

R is "automatic" in the tensorial reduction (algebra in dimension d)

On-shell methods offer different options for calculation of **R**

- Higher integer dimensions Giele, Kunszt, Melnikov
- Loop-level on shell recursions Berger, Bern, Dixon, Forde and Kosower
- Mass continuation method Badger
- Tree-level like Feynman Rules G.O., Papadopoulos, Pittau; Draggiotis, Garzelli, Malamos, Pittau; Shao, Zhang, Chao

Integral-level vs Integrand-level

Description in terms of Master Integrals I_i $\rightarrow$ Integral Level

$$\int d^n\bar{q}\ A(\bar{q}) = \int d^n\bar{q} \frac{N(\bar{q})}{\bar{D}_0\bar{D}_1\ldots\bar{D}_{m-1}} = c_0 I_0 + c_1 I_1 + \ldots + C_n I_n$$

Integral-level vs Integrand-level

Description in terms of Master Integrals I_i $\rightarrow$ Integral Level

$$\int d^n\bar{q}\; A(\bar{q}) = \int d^n\bar{q} \frac{N(\bar{q})}{\bar{D}_0\bar{D}_1\ldots\bar{D}_{m-1}} = c_0 I_0 + c_1 I_1 + \ldots + C_n I_n$$

Integrand Level $\rightarrow$ The $N = N$ identity

$$N(\bar{q}) = ???$$

Challenge: write a complete expression at the l.h.s. for $N(\bar{q})$

- powers of q and μ^2
- scalar products
- reconstructed denominators $\bar{D}_i$

Integrand-Level Approach

Integrand level decomposition:

$$
\begin{aligned}
N(\bar{q}) &= \sum_{i<<m}^{n-1} \Delta_{ijk\ell m}(\bar{q}) \prod_{h\neq i,j,k,\ell,m}^{n-1} \bar{D}_h + \sum_{i<<\ell}^{n-1} \Delta_{ijk\ell}(\bar{q}) \prod_{h\neq i,j,k,\ell}^{n-1} \bar{D}_h + \\
&+ \sum_{i<<k}^{n-1} \Delta_{ijk}(\bar{q}) \prod_{h\neq i,j,k}^{n-1} \bar{D}_h + \sum_{i<j}^{n-1} \Delta_{ij}(\bar{q}) \prod_{h\neq i,j}^{n-1} \bar{D}_h + \sum_{i}^{n-1} \Delta_{i}(\bar{q}) \prod_{h\neq i}^{n-1} \bar{D}_h
\end{aligned}
$$

Integrand-Level Approach

Integrand level decomposition:

$$
\begin{aligned}
N(\bar{q}) &= \sum_{i<<m}^{n-1} \Delta_{ijk\ell m}(\bar{q}) \prod_{h\neq i,j,k,\ell,m}^{n-1} \bar{D}_h + \sum_{i<<\ell}^{n-1} \Delta_{ijk\ell}(\bar{q}) \prod_{h\neq i,j,k,\ell}^{n-1} \bar{D}_h + \\
&+ \sum_{i<<k}^{n-1} \Delta_{ijk}(\bar{q}) \prod_{h\neq i,j,k}^{n-1} \bar{D}_h + \sum_{i<j}^{n-1} \Delta_{ij}(\bar{q}) \prod_{h\neq i,j}^{n-1} \bar{D}_h + \sum_{i}^{n-1} \Delta_{i}(\bar{q}) \prod_{h\neq i}^{n-1} \bar{D}_h
\end{aligned}
$$

Recombining with the denominators:

$$
\begin{aligned}
A(\bar{q}) &= \sum_{i<<m}^{n-1} \frac{\Delta_{ijk\ell m}(\bar{q})}{\bar{D}_i \bar{D}_j \bar{D}_k \bar{D}_\ell \bar{D}_m} + \sum_{i<<\ell}^{n-1} \frac{\Delta_{ijk\ell}(\bar{q})}{\bar{D}_i \bar{D}_j \bar{D}_k \bar{D}_\ell} + \sum_{i<<k}^{n-1} \frac{\Delta_{ijk}(\bar{q})}{\bar{D}_i \bar{D}_j \bar{D}_k} + \\
&+ \sum_{i<j}^{n-1} \frac{\Delta_{ij}(\bar{q})}{\bar{D}_i \bar{D}_j} + \sum_{i}^{n-1} \frac{\Delta_{i}(\bar{q})}{\bar{D}_i} \; ,
\end{aligned}
$$

the decomposition exposes **the multi-pole nature of the integrand**

Integrand-Level Approach

Integrand level decomposition:

$$\begin{aligned} N(\bar{q}) &= \sum_{i<<m}^{n-1} \Delta_{ijk\ell m}(\bar{q}) \prod_{h\neq i,j,k,\ell,m}^{n-1} \bar{D}_h + \sum_{i<<\ell}^{n-1} \Delta_{ijk\ell}(\bar{q}) \prod_{h\neq i,j,k,\ell}^{n-1} \bar{D}_h + \\ &+ \sum_{i<<k}^{n-1} \Delta_{ijk}(\bar{q}) \prod_{h\neq i,j,k}^{n-1} \bar{D}_h + \sum_{i<j}^{n-1} \Delta_{ij}(\bar{q}) \prod_{h\neq i,j}^{n-1} \bar{D}_h + \sum_{i}^{n-1} \Delta_{i}(\bar{q}) \prod_{h\neq i}^{n-1} \bar{D}_h \end{aligned}$$

- the functional form is **process-independent**, the process-dependent coefficients are contained in the Δ's
- the **Rational Term is automatically included**
- **polynomial fitting replaces the integration** (both sides are polynomial)
- we can **extract the coefficients going on-shell**: we only need the **Numerator Function evaluated on the cuts**

GO, Papadopoulos, Pittau;
Mastrolia, GO, Reiter, Tramontano

Example: 4-particles process (in 4-dim)

$$
\begin{aligned}
N(q) &= d + \tilde{d}(q) + \sum_{i=0}^{3} \left[c(i) + \tilde{c}(q;i) \right] D_i + \sum_{i_0 < i_1}^{3} \left[b(i_0 i_1) + \tilde{b}(q; i_0 i_1) \right] D_{i_0} D_{i_1} \\
&+ \sum_{i_0=0}^{3} \left[a(i_0) + \tilde{a}(q; i_0) \right] D_{i \neq i_0} D_{j \neq i_0} D_{k \neq i_0}
\end{aligned}
$$

We look for a q such that

$$D_0 = D_1 = D_2 = D_3 = 0$$

$\rightarrow$ there are two solutions $q_0^{\pm}$

Example: 4-particles process (in 4-dim)

$$N(q) \;=\; d + \tilde{d}(q)$$

Our "master formula" for $q = q_0^{\pm}$ is:

$$N(q_0^{\pm}) = [d + \tilde{d}\,(w \cdot q_0^{\pm})]$$

$\rightarrow$ solve to extract the coefficients d and $\tilde{d}$

Example: 4-particles process (in 4-dim)

$$N(q) - d - \tilde{d}(q) = \sum_{i=0}^{3} \left[c(i) + \tilde{c}(q;i) \right] D_i + \sum_{i_0<i_1}^{3} \left[b(i_0 i_1) + \tilde{b}(q;i_0 i_1) \right] D_{i_0} D_{i_1}$$
$$+ \sum_{i_0=0}^{3} \left[a(i_0) + \tilde{a}(q;i_0) \right] D_{i\neq i_0} D_{j\neq i_0} D_{k\neq i_0}$$

Then we can move to the extraction of c coefficients using

$$N'(q) = N(q) - d - \tilde{d}(w \cdot q)$$

and setting to zero three denominators (ex: $D_1 = 0$, $D_2 = 0$, $D_3 = 0$)

Example: 4-particles process (in 4-dim)

$$N(q) - d - \tilde{d}(q) \quad = \qquad [c(0) + \tilde{c}(q;0)]\, D_0$$

We have infinite values of q for which

$$D_1 = D_2 = D_3 = 0 \quad \text{and} \quad D_0 \neq 0$$

$\rightarrow$ Here we need 7 of them to determine $c(0)$ and $\tilde{c}(q;0)$

Integrand-Reduction via Laurent Expansion

- The coefficients of the integrand can be extracted by performing a Laurent expansion with respect to one of the free parameters which appear in the solutions of the cut
- corrections at the coefficient level replace the subtractions at the integrand level

Advantages:

- a "lighter" reduction algorithm where fewer coefficients
- 4-cut decoupled from triple-, double-, and single-cut.
- No more "sampling on the cuts"

This method has been implemented in the C++ library **Ninja** and interfaced with GoSam: Preliminary tests show an improvement in the computational performance

Mastrolia, Mirabella, Peraro

see also **Forde; Badger**

Different Multi-Purpose Codes/Tools

Several Strategies = Many Computational Tools:

- ★ BlackHat [Bern, Dixon, Febres Cordero, Hoeche, Ita, **Kosower**, Maitre, Ozeren]
- ★ Feynarts/Formcalc/LoopTools [**Hahn** et al.]
- ★ GoSam [Cullen, Greiner, **Heinrich**, Mastrolia, GO, Reiter, Tramontano, Luisoni]
- ★ Helac-NLO [Bevilacqua, Czakon, van Hameren, Papadopoulos, Pittau, Worek]
- ★ MadLoop [Hirschi, Frederix, Frixione, Garzelli, Maltoni, Pittau]
- ★ NJet [Badger, Biedermann, Uwer, **Yundin**]
- ★ Openloops [Cascioli, **Maierhöfer**, Pozzorini]
- ★ Recola [Actis, Denner, Hofer, Scharf, Uccirati]
- ★ Rocket [Ellis, Giele, Kunszt, Melnikov, Zanderighi]

The "NLO industrial revolution" = NLO Automation

Different Multi-Purpose Codes/Tools

Several Strategies = Many Computational Tools:

- ★ BlackHat → Generalized Unitarity
- ★ Feynarts/Formcalc/LoopTools → Feynman Diagrams + Tens.Red./Integrand-Level
- ★ GoSam → Feynman Diagrams + Tens.Red./Integrand-Level
- ★ Helac-NLO → Tree-level recursion + Integrand-Level
- ★ MadLoop → Tree-level recursion + Integrand-Level
- ★ NJet → Generalized Unitarity
- ★ Openloops → Recursive Tensorial Reconstruction + Tens.Red./Integrand-Level
- ★ Recola → Recursive Tensorial Reconstruction + Tens.Red./Integrand-Level
- ★ Rocket → Generalized Unitarity

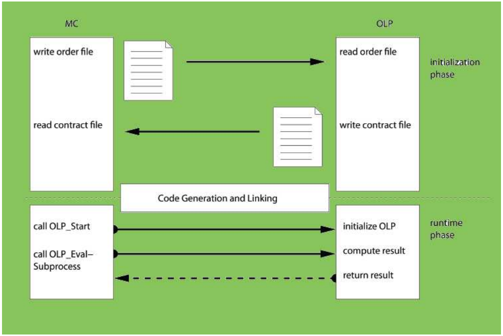
The "NLO industrial revolution" = NLO Automation

Modularity, Automation, BLHA

Automation is crucial for multi-lop NLO calculations:

- Optimization/Self-organization
- Avoid human mistakes
- Process-independent techniques

Different Levels of Automation: MC controls the OLP via BLHA



WHAT ABOUT HIGHER LOOPS?

Problem: at two loops (and higher), we do not have a
Standard Complete Basis of Master Integrals

recent work of Gluza, Kosower, Kajda; Schabinger

The most common (and successful) approach relies on:

- Amplitude generation via **Feynman diagrams**
- Reduction to a minimal set of MIs using **IBPs** (Laporta algorithm)
- Direct Computation of the MIs (analytically or numerically)

WHAT ABOUT HIGHER LOOPS?

Problem: at two loops (and higher), we do not have a **Standard Complete Basis of Master Integrals**

Can we extend what we learned at one-loop about to develop alternative approaches for multi-loop reduction?

- **Integrand-level Techniques**
  Mastrolia, GO (2011); Badger, Frellersvig, Zhang; Kleiss, Malamos, Papadopoulos, Verheyen; Zhang; Mastrolia, GO, Mirabella, Peraro; Feng, Huang; Huang, Zhang
- **Maximal Unitarity**
  Kosower, Larsen (2011); Johansson, Kosower, Larsen; Larsen, Caron-Huot

Integrand-Level Strategy

Let's consider a two-loop integral with n denominators:

$$\int dq\ dk\ \frac{N(q,k)}{\bar{D}_1 \bar{D}_2 \dots \bar{D}_n}$$

As in the one-loop case, we want to construct an identity for the integrands:

$$N(q,k) = \sum_{i_1 << i_8}^{n} \Delta_{i_1,\dots,i_8}(q,k) \prod_{h \neq i_1,\dots,i_8}^{n} \bar{D}_h + \dots + \sum_{i_1 << i_2}^{n} \Delta_{i_1,i_2}(q,k) \prod_{h \neq i_1,i_2}^{n} \bar{D}_h$$

$$A(q,k) = \sum_{i_1 << i_8}^{n} \frac{\Delta_{i_1,\dots,i_8}(q,k)}{\bar{D}_{i_1} \bar{D}_{i_2} \dots \bar{D}_{i_8}} + \sum_{i_1 << i_7}^{n} \frac{\Delta_{i_1,\dots,i_7}(q,k)}{\bar{D}_{i_1} \bar{D}_{i_2} \dots \bar{D}_{i_7}} + \dots + \sum_{i_1 << i_2}^{n} \frac{\Delta_{i_1,i_2}(q,k)}{\bar{D}_{i_1} \bar{D}_{i_2}}$$

- Which terms appear in the above expressions?
- What is the general form of the residues $\Delta_{i_1,\dots,i_m}$?

Collaboration with **P.Mastrolia, E.Mirabella, T.Peraro, U.Schubert**

Integrand-Level Strategy

Mastrolia, GO, Mirabella, Peraro

Let's look at the on-shell conditions, and impose

$$D_1 = D_2 = \ldots = D_n = 0$$

1) There are no solutions $\rightarrow$ **reducible**

$\rightarrow$ The n-denominator integrand can be written in terms of integrands with $(n-1)$ denominators
$\rightarrow$ it is **fully reducible in terms of lower point functions**
i.e. a six-point function at one-loop

Integrand-Level Strategy

Mastrolia, GO, Mirabella, Peraro

Let's look at the on-shell conditions, and impose

$$D_1 = D_2 = \ldots = D_n = 0$$

1) There are no solutions $\rightarrow$ **reducible**

2) The cut has solutions $\rightarrow$ there is a **residue** Δ

We divide the numerator modulo the Gröbner basis of the n-ple cut (a set of polynomials vanishing on the same on-shell cuts of the denominators).
$\rightarrow$ The *remainder* of the division is the *residue* of the n-ple cut.
$\rightarrow$ The *quotients* generate integrands with $(n-1)$ denominators.

Mastrolia, GO, Mirabella, Peraro

Let's look at the on-shell conditions, and impose

$$D_1 = D_2 = \ldots = D_n = 0$$

1) There are no solutions $\rightarrow$ **reducible**
2) The cut has solutions $\rightarrow$ there is a **residue** Δ
3) Finite number of solutions n_s $\rightarrow$ **Maximum Cut**

$\rightarrow$ "Maximum Cut" i.e. a four-point function at one-loop (in 4-dim)
$\rightarrow$ its residue is a univariate polynomial parametrized by n_s coefficients
$\rightarrow$ the corresponding residue can always be reconstructed at the cut
$\rightarrow$ the residue is determined as in the previous case

"On-shell" in Algebraic Geometry Language

$$\mathcal{I}_{i_1\cdots i_n} = \frac{\mathcal{N}_{i_1\cdots i_n}(z)}{D_{i_1}(z)\cdots D_{i_n}(z)}$$

where z = components of the loop momenta

- Ideal: $\mathcal{J}_{i_1\cdots i_n} = \langle D_{i_1}, \cdots, D_{i_n}\rangle$
- Gröbner basis $\mathcal{G}_{i_1\cdots i_n}$: same zero as the denominators
- Multivariate division of $\mathcal{N}_{i_1\cdots i_n}$ modulo $\mathcal{G}_{i_1\cdots i_n}$

$$\mathcal{N}_{i_1\cdots i_n}(z) = \Gamma_{i_1\cdots i_n} + \Delta_{i_1\cdots i_n}(z)$$

- The quotient $\Gamma_{i_1\cdots i_n}$ can be expressed in terms of denominators

$$\Gamma_{i_1\cdots i_n} = \sum_{\kappa=1}^{n} \mathcal{N}_{i_1\cdots i_{\kappa-1} i_{\kappa+1}\cdots i_n}(\zeta)\bar{D}_{i_\kappa}(\zeta)$$

- Which provides the Recursive Formula

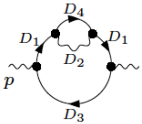
$$\mathcal{I}_{i_1\cdots i_n} = \sum_{\kappa=1}^{n} \mathcal{I}_{i_1\cdots i_{\kappa-1} i_{\kappa+1} i_n} + \frac{\Delta_{i_1\cdots i_n}}{\bar{D}_{i_1}\cdots\bar{D}_{i_n}}$$

Two-loop Photon Self-Energy in QED

Mastrolia, Peraro (2013)

⇝ higher powers of propagators are not problematic

- Example: QED photon self energy @ two loops



$$\mathcal{I} = \frac{\Delta_{11234}}{D_1^2 D_2 D_3 D_4} + \frac{\Delta_{1234}}{D_1 D_2 D_3 D_4} + \frac{\Delta_{1124}}{D_1^2 D_2 D_4} + \frac{\Delta_{234}}{D_2 D_3 D_4} + \frac{\Delta_{124}}{D_1 D_2 D_4}$$

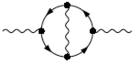
$$\mathcal{N} = 16[\mu_{11}^2 - \mathrm{k}_1^2\,(\mathrm{k}_1 \cdot p)] + \cdots$$

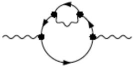
d dimensions: $\bar{\mathrm{k}}_i^\mu = \mathrm{k}_i^\mu + \vec{\mu}_i \qquad \mu_{ij} \equiv \vec{\mu}_i \cdot \vec{\mu}_j$

- Reduction completed after five steps ...

$$\mathcal{I} = \frac{8\mu_{11}(2\mu_{11} - p^2)}{D_1^2 D_2 D_3 D_4} - \frac{8(\mu_{11} + p^2)}{D_1 D_2 D_3 D_4} + \frac{8\mu_{11}}{D_1^2 D_2 D_4} - \frac{8}{D_2 D_3 D_4} + \frac{8}{D_1 D_2 D_4}$$

- ... performed for the full $\mathcal{N}$ ...
- ... and for the other diagrams





Conclusions/Outlook

- Mini-collection of "Conclusions" form the past 5 years:
  "Can we achieve at NLO the same degree of automation of the LO?"
  "This method is potentially a good candidate for NLO automation"
  "A generic NLO calculator seems feasible"

Conclusions/Outlook

- At present, there is a great variety of methods available for NLO scattering amplitudes: Old ideas merged with New Ideas

Tensorial Reduction

Generalized Unitarity

Integrand-level reduction

Techniques for the Rational Terms

Trees "recycled" into loops

Automation of Feynman Diagrams

On-shell tree-amplitudes

Off-shell currents

Recursion Relations

Tensorial Reconstruction

Conclusions/Outlook

- At present, there is a great variety of methods available for NLO scattering amplitudes: Old ideas merged with New Ideas

Tensorial Reduction	Automation of Feynman Diagrams
Generalized Unitarity	On-shell tree-amplitudes
Integrand-level reduction	Off-shell currents
Techniques for the Rational Terms	Recursion Relations
Trees "recycled" into loops	Tensorial Reconstruction

- OLP integrated inside the MC (or via BLHA) to generate theoretical prediction for a large variety of processes: high multiplicities, several scales, effective vertices . . .

Conclusions/Outlook

- At present, there is a great variety of methods available for NLO scattering amplitudes: Old ideas merged with New Ideas

Tensorial Reduction

Generalized Unitarity

Integrand-level reduction

Techniques for the Rational Terms

Trees "recycled" into loops

Automation of Feynman Diagrams

On-shell tree-amplitudes

Off-shell currents

Recursion Relations

Tensorial Reconstruction

- OLP integrated inside the MC (or via BLHA) to generate theoretical prediction for a large variety of processes: high multiplicities, several scales, effective vertices ...

Ideas + Automation = NLO Revolution

"Will we achieve at NNLO the same degree of automation of the NLO?"

(not a serious question, but...)