BlackHat: Implementation of unitarity cut method for one-loop amplitudes

Daniel Maître

for the BlackHat collaboration

C. Berger, Z. Bern, L. Dixon, F. Febres Cordero, D. Forde, H. Ita, D. Kosower, DM

Stanford Linear Accelerator Center

LHC-D workshop 2008, Zürich, 3 June 2008

NLO corrections

NLO corrections are very important

- NLO corrections are large in particular in QCD
- NLO corrections affect the shape of distributions
- New production channels may open at NLO, affecting both cross sections and distributions
- Reduce the scale uncertainty of tree level cross sections
- The signals can closely resemble the background
- A precise understanding of the background is then mandatory
- Precise studies at the LHC will require NLO corrections for many high multiplicity QCD background processes

Automation of NLO computations

- many processes, many legs $\Rightarrow$ need automation

Tasks:

- Real radiation part
  - Challenge is systematic extraction of the singularities
  - Different methods
    - Subtraction method (dipole, antenna)
    - Sector decomposition
    - Phase space slicing
  - Automation possible [Gleisberg,Krauss;Seymour,Tevlin]
- Virtual part
  - Current bottleneck for the automation of the NLO corrections
  - both for analytic and numerical evaluation

One loop corrections

- Method based on Feynman diagrams work well for $2 \to 3,4$ processes
- scaling of the Feynman diagrams approach is very challenging
- Unitarity method
  - Many new analytic results
  - Numerical results for full amplitudes :
    $e^+e^- \to Z \to 4partons$ [Bern,Dixon,Kosower,Weinzierl]
    included in MCFM [Campbell,Ellis]
    ZZZ,WZZ,WWZ,ZZZ [Binoth,Ossola,Papadopoulos,Pittau]
  - Tools: CutTools [Ossola,Papadopoulos,Pittau]
    + improvements [Mastrolia,Ossola,Papadopoulos,Pittau]
  - Numerical implementations for virtual amplitudes
    - 6 gluons, cut part only [Ellis,Giele,Kunszt]
    - 6 gluons [Giele,Kunszt,Melnikov]
    - up to 20 gluons [Giele,Zanderighi]

The BlackHat Project

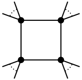
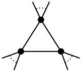
Goal: Automating the computation of one-loop amplitudes using on-shell methods

- C++ code
- To compute one-loop virtual amplitudes we use
  - Unitarity bootstrap
    - generalized unitarity for the cut constructible part
    - recursion relations for the rational part
  - Spinor-helicity formalism
  - Complex momenta

One-Loop Decomposition

A one-loop amplitude can be decomposed into a sum of coefficients multiplying scalar integrals and rational terms.

$$A = R + C$$

$C = \sum_i b_i$  $+ \sum_i c_i$  $+ \sum_i d_i$ 

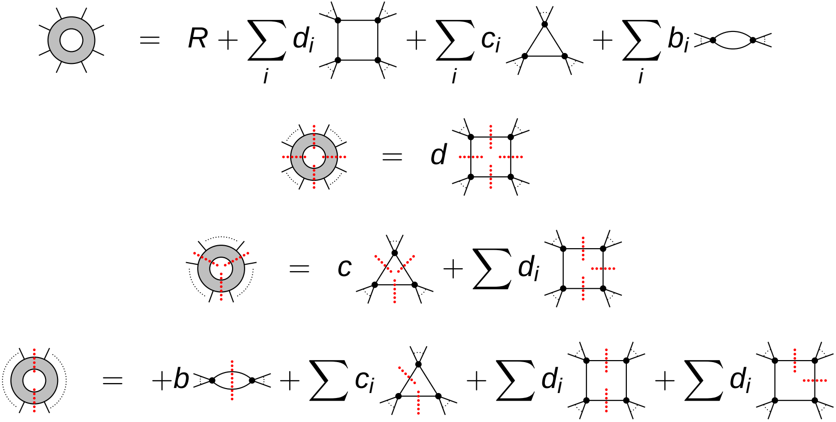
- The task is reduced to determining the coefficients
- The coefficients b_i, c_i, d_i can be computed using generalized unitarity techniques in $d = 4$ dimensions
- The rational part has to be computed separately.
  $\Rightarrow$ use recursion relations

Generalized Unitarity

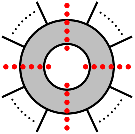
A unitarity cut is the replacement

$$\frac{i}{p^2 - m^2 + i\epsilon} \to 2\pi\delta(p^2 - m^2)$$

Cut can be used to isolate integral coefficient



Quadruple cut



$$= \int d^4 l \delta(l_1^2)\delta(l_2^2)\delta(l_3^2)\delta(l_4^4)\mathcal{A}$$

$$= A_1^{\text{tree}}(l^\pm)A_2^{\text{tree}}(l^\pm)A_3^{\text{tree}}(l^\pm)A_4^{\text{tree}}(l^\pm)$$

With one massless leg in one corner:

$$l_1^{\pm\mu} = \frac{\langle 1^\mp | \not{K}_2 \not{K}_3 \not{K}_4 \gamma^\mu | 1^\pm \rangle}{2\langle 1^\mp | \not{K}_2 \not{K}_4 | 1^\pm \rangle}, \quad l_2^{\pm\mu} = -\frac{\langle 1^\mp | \gamma^\mu \not{K}_2 \not{K}_3 \not{K}_4 | 1^\pm \rangle}{2\langle 1^\mp | \not{K}_2 \not{K}_4 | 1^\pm \rangle},$$

$$l_3^{\pm\mu} = \frac{\langle 1^\mp | \not{K}_2 \gamma^\mu \not{K}_3 \not{K}_4 | 1^\pm \rangle}{2\langle 1^\mp | \not{K}_2 \not{K}_4 | 1^\pm \rangle}, \quad l_4^{\pm\mu} = -\frac{\langle 1^\mp | \not{K}_2 \not{K}_3 \gamma^\mu \not{K}_4 | 1^\pm \rangle}{2\langle 1^\mp | \not{K}_2 \not{K}_4 | 1^\pm \rangle}.$$

Gram determinant for $K_1^2 = 0$

$$\Delta_4 = -2\langle 1^- | \not{K}_2 \not{K}_4 | 1^+ \rangle \langle 1^+ | \not{K}_2 \not{K}_4 | 1^- \rangle$$

Three-particle cut

Momentum parametrization with two massless four vectors [del Aguila,Ossola,Papadopoulos,Pittau;Forde]

$$l^\mu = \alpha_1 k_1^\mu + \alpha_2 k_2^\mu + \frac{\alpha_3}{2}\langle k_1|\gamma^\mu|k_2] + \frac{\alpha_4}{2}\langle k_2|\gamma^\mu|k_1]$$

The three delta functions fix three of the coefficients α_i

$$l^\mu = \tilde{K}_1^\mu + \tilde{K}_2^\mu + \frac{t}{2}\left\langle \tilde{K}_1|\gamma^\mu|\tilde{K}_2\right] + \frac{1}{2t}\left\langle \tilde{K}_2|\gamma^\mu|\tilde{K}_1\right]$$

Triple cut is a function of t

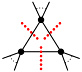
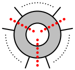
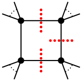
$$C(t) = \frac{c_{-3}}{t^3} + \frac{c_{-2}}{t^2} + \frac{c_{-1}}{t} + c_0 + c_1 t + c_2 t^2 + c_3 t^3 + \sum_{poles} \frac{d_i}{\xi_i (t - t_i)}$$

Poles in t originate from additional propagators going on-shell

$$(l_i - K)^2 \to (t - t_i)\xi_i$$

Three-particle cut

$$T(t) \equiv C(t) - \sum \frac{d_i}{\xi_i (t - t_i)}$$

c  $=$  $- \sum d_i$ 

Subtracted triple cut is a function of t [OPP]

$$T(t) = \frac{c_{-3}}{t^3} + \frac{c_{-2}}{t^2} + \frac{c_{-1}}{t} + c_0 + c_1 t + c_2 t^2 + c_3 t^3$$

$$c_0 = \frac{1}{2p+1} \sum_{j=-p}^{p} T_3(t_0 e^{2\pi i j/(2p+1)})$$

- Discrete Fourier projection avoids numerically unstable matrix invertions.
- Extraction of coefficients using DFP is also used in other methods [Britto,Feng,Mastrolia;Mastrolia,Ossola,Papadopoulos,Pittau]

Two-particle cut

Momentum parametrization with two massless four vectors

$$l^\mu = \alpha_1 k_1^\mu + \alpha_2 k_2^\mu + \frac{\alpha_3}{2}\langle k_1|\gamma^\mu|k_2] + \frac{\alpha_4}{2}\langle k_2|\gamma^\mu|k_1]$$

The two delta functions leave two free parameters α_i

$$l^\mu = y\tilde{K}_1^\mu + (1-y)\tilde{\chi}^\mu + \frac{t}{2}\left\langle \tilde{K}_1|\gamma^\mu|\chi\right] + \frac{y(1-y)}{2t}\left\langle \chi|\gamma^\mu|\tilde{K}_1\right]$$

Double cut integrand is a function of t and y

$$\begin{aligned} C_2(y,t) &= A_1(t,y)A_2(t,y) \\ B_2(y,t) &\equiv C_2(y,t) - \sum box - \sum triangle \end{aligned}$$

$$b_0 = \frac{1}{20}\sum_{j=0}^{4}\left[B_2\left(0, t_0\, e^{2\pi i j/5}\right) + 3B_2\left(2/3, t_0\, e^{2\pi i j/5}\right)\right].$$

Recursion for Rational Terms

Use the analytic properties of the one-loop amplitude to construct the rational term

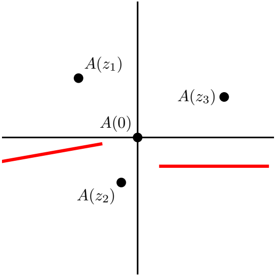
- use a complex shift on the full amplitude

$$p_1 \rightarrow p_1(z), p_2 \rightarrow p_2(z)$$

$$A \rightarrow A(z)$$

Consider the complex function $A(z)$

- poles, $s_{i\ldots j}(z) \rightarrow 0$
- cuts $\log(s_{i\ldots j}(z))$



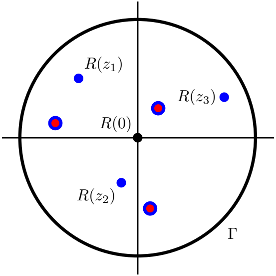
Rational term

Consider $R(z)$

$$\frac{1}{2\pi i}\int_\Gamma \frac{R(z)}{z} = R_\infty$$

- The value R_∞ of the contour integral at ∞ can be constructed using an auxiliary recursion.

$$\begin{aligned} R(0) &= R_\infty - \sum_{\text{poles}\,\alpha} \operatorname*{Res}_{z=z_\alpha} \frac{R(z)}{z} \\ &= R_\infty - \sum_{\text{phys}} \operatorname*{Res}_{z_p} \frac{R(z)}{z} - \sum_{\text{spur}} \operatorname*{Res}_{z_s} \frac{R(z)}{z} \end{aligned}$$



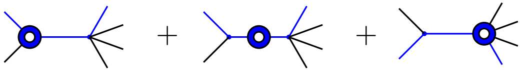
- Two different types of poles
  - physical poles, $s_{i\dots j}(z) \to 0$
  - spurious poles

Rational term

$$R(0) = R_\infty - \sum_{\text{phys}} \operatorname*{Res}_{z_p} \frac{R(z)}{z} - \sum_{\text{spur}} \operatorname*{Res}_{z_s} \frac{R(z)}{z}$$

- $R(z)$ factorizes at the physical pole locations, so that we can use recursion relations. [Bern,Dixon,Kosower]

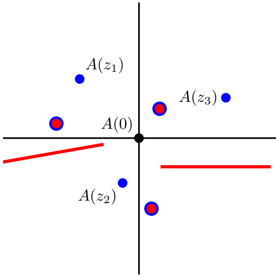
$$\operatorname*{Res}_{z_p} \frac{R(z_p)}{z_p} =$$



$$R_D = -\sum_{z_p} \operatorname*{Res}_{z_p} \frac{R(z_p)}{z_p}$$

Recursion for Rational Terms

$$\frac{A(z)}{z} = \frac{R(z)}{z} + \frac{C(z)}{z}$$



- Spurious poles z_s appear in $C(z)$ and $R(z)$ due to Gram determinants
- The residues of $R(z)/z$ and $C(z)/z$ at the unphysical poles have to cancel since $A(z)$ has no spurious poles.

$$Res_{z_s}\frac{R_S(z_s)}{z_s} = -Res_{z_s}\frac{C(z_s)}{z_s}$$

Numerical extraction of the spurious poles

We compute $\sum_{\text{spur}} \operatorname{Res}_{z_s} \frac{C(z_s)}{z_s}$ numerically

Numerical spurious extraction is tricky, but possible because

- Precise cut part input
- Location of the spurious poles is known a priori
- Only need to evaluate a small part of $C(z)$ around the pole.

Challenges

- Evaluation time
- Control of numerical precision for exceptional points
  - Using increased precision (32 digits, 64 digits) when needed for some pieces
  - Automatic diagnosis

Precision diagnosis

The precision of the computed amplitude can be assessed

- Cut part

$$A_n^{\text{oneloop}}|_{1/\epsilon,\,\text{non}-\text{log}} = \frac{1}{\epsilon}\sum_k b_k = -\left[\frac{1}{\epsilon}\left(\frac{11}{3} - \frac{2}{3}\frac{n_f}{N_c}\right)\right] A_n^{\text{tree}}\,,$$

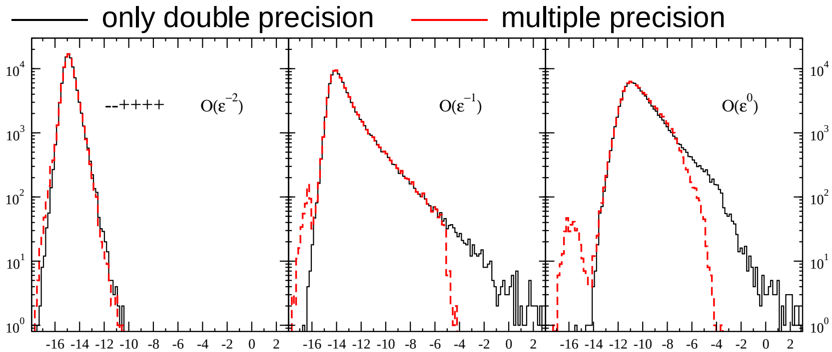
- Spurious poles

$$A_n^{\text{oneloop}}(z_s)|_{1/\epsilon,\,\text{non}-\text{log}} = \frac{1}{\epsilon}\sum_k b_k(z_s) = 0\,,$$

- Big cut/rational cancellations

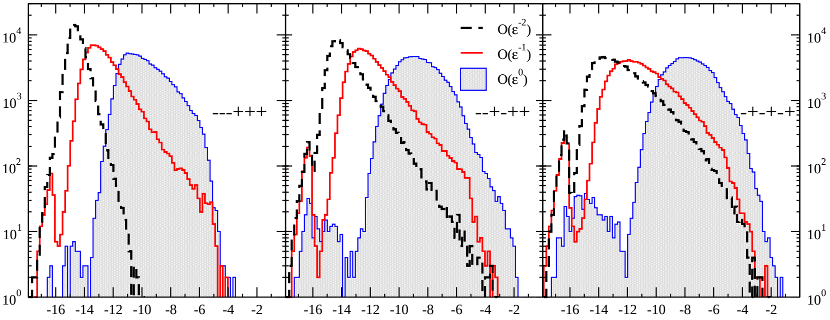
Split MHV results

- Accuracy: $\log_{10}\left(\frac{|A^{\mathrm{num}}-A^{\mathrm{ref}}|}{|A^{\mathrm{ref}}|}\right)$

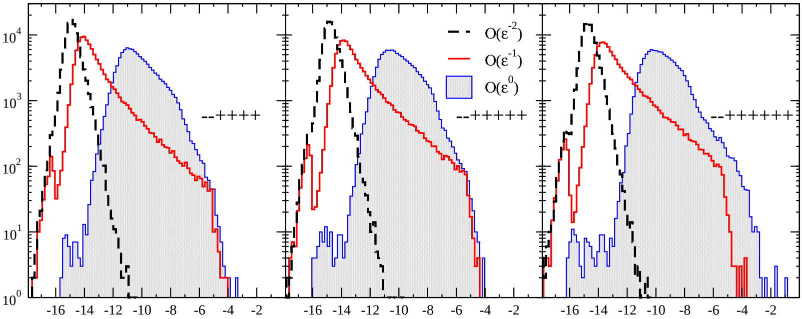


- 100'000 points with cuts
  - $E_T > 0.01\sqrt{s}$
  - Pseudo-rapidity $\eta < 3$
  - $\Delta_R > 0.4 \qquad \Delta_R = \sqrt{\Delta_\eta^2 + \Delta_\phi^2}$

NMHV amplitudes



MHV amplitudes



Timing

- 2.33 GHz Xeon processor

helicity	cut part only	double prec. only	multi-prec.
$--++++$	2.4 ms	6.8 ms	8.8 ms
$--+++++$	3.8 ms	10.5 ms	13 ms
$--++++++$	5.5 ms	27 ms	31 ms
$-+-+++$	2.9 ms	15.5 ms	19 ms
$-++-++$	3.1 ms	55 ms	60 ms
$---+++$	4.3 ms	12 ms	14 ms
$--+-++$	5.7 ms	37 ms	44 ms
$-+-+-+$	6.7 ms	55 ms	67 ms

- Bottleneck is the spurious pole evaluation
- The effect of higher precision on the evaluation time is noticeable but not dominant

Outlook

First tests passed

- Numerical stability is under control
- Evaluation time is reasonable

A lot more to do ...

- Sums (color orderings+ external helicities)
- Fermions (internal+external)
- Vector bosons (external)
- Masses (internal+external)
- Speed and precision improvements
- Combine with real part