Multi-leg scattering amplitudes for LHC phenomenology: modern tools and methods

Tiziano Peraro

Higgs Centre for Theoretical Physics
University of Edinburgh

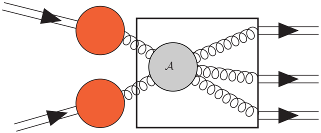
QCD@LHC, Zurich
25 August 2016

Introduction and motivation

Introduction and motivation

- Experiments Large Hadron Collider (LHC)
  - high-accuracy experimental data (up to % level)
  - high c.o.m. energy ⇒ multi-particle final states
  - large SM background (could hide new/interesting physics)

We need scattering amplitudes for theoretical predictions with
- high accuracy
- multi-particle interactions



Scattering amplitudes
- LO not reliable ⇒ need at least NLO
- NNLO needed for high precision

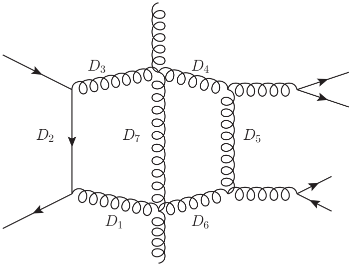
⇒ need to compute loop integrals

Loop amplitudes

- The integrand of a generic ℓ-loop integral:
  - is a rational function in the components of the loop momenta k_i
  - polynomial numerator $\mathcal{N}_{i_1\cdots i_n}$

$$\mathcal{M}_n = \int d^d k_1 \cdots d^d k_\ell \;\; \mathcal{I}_{i_1\cdots i_n}, \qquad \mathcal{I}_{i_1\cdots i_n} \equiv \frac{\mathcal{N}_{i_1\cdots i_n}}{D_{i_1}\cdots D_{i_n}}$$

  - quadratic polynomial denominators D_i
    - they correspond to Feynman loop propagators



$$D_i = \Big(\sum_j (-)^{s_{ij}} k_j + p_i\Big)^2 - m_i^2$$

Loop amplitudes

- Loop amplitudes can be written as linear combinations of Master Integrals (MIs)

$$\mathcal{A}^{(L)} = \sum_i c_i \, I_i$$

- the integrals I_i are special functions of the kinematic invariants
  - at one-loop only logarithms and dilogarithms
  - at higher loops multiple polylogarithms, elliptic functions, etc...
- the coefficients c_i are rational functions of kinematic invariants
  - ...but their computation can be more complex than the MIs, especially for high-multiplicity processes

Loop amplitudes

- Loop amplitudes can be written as linear combinations of Master Integrals (MIs)

$$\mathcal{A}^{(L)} = \sum_i c_i \, I_i$$

- the integrals I_i are special functions of the kinematic invariants
  - at one-loop only logarithms and dilogarithms
  - at higher loops multiple polylogarithms, elliptic functions, etc. . .
- the coefficients c_i are rational functions of kinematic invariants
  - . . . but their computation can be more complex than the MIs, especially for high-multiplicity processes

In this talk I will mostly focus on the coefficients

$\Rightarrow$ **see also K. Papadopoulos's talk on the calculation of MIs**

One-loop integrand reduction and automated tools

The Integrand reduction of one-loop amplitudes

- Every one-loop integrand, can be decomposed as
  [Ossola, Papadopoulos, Pittau (2007); Ellis, Giele, Kunszt, Melnikov (2008)]

$$\mathcal{I}_n = \frac{\mathcal{N}}{D_1 \cdots D_n} = \sum_{j_1 \cdots j_5} \frac{\Delta_{j_1 j_2 j_3 j_4 j_5}}{D_{j_1} D_{j_2} D_{j_3} D_{j_4} D_{j_5}} + \sum_{j_1 j_2 j_3 j_4} \frac{\Delta_{j_1 j_2 j_3 j_4}}{D_{j_1} D_{j_2} D_{j_3} D_{j_4}} + \sum_{j_1 j_2 j_3} \frac{\Delta_{j_1 j_2 j_3}}{D_{j_1} D_{j_2} D_{j_3}} + \sum_{j_1 j_2} \frac{\Delta_{j_1 j_2}}{D_{j_1} D_{j_2}} + \sum_{j_1} \frac{\Delta_{j_1}}{D_{j_1}}$$

- The residues or on-shell integrands

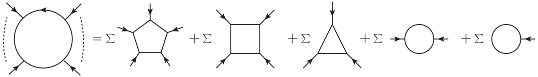
$$\Delta_{i_1 \cdots i_k} = \sum_i \underbrace{c_i^{(i_1 \cdots i_k)}}_{\text{process dep.}} \qquad \underbrace{\mathbf{m}_i^{(i_1 \cdots i_k)}(k)}_{\substack{\text{universal basis} \\ \text{polynomials in the loop } k^\mu}}$$

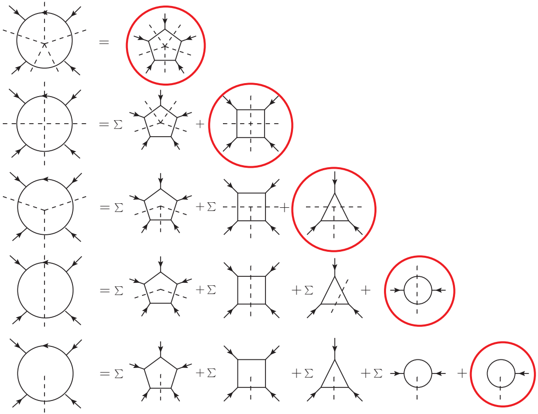
  - form a known, universal integrand basis
  - unknown, process-dependent coefficients $c_i \Rightarrow$ polynomial fit
- All the integrals of the integrand basis $\mathbf{m}_i^{(i_1 \cdots i_k)}$ are known at one loop

Fit-on-the-cut at one-loop

[Ossola, Papadopoulos, Pittau (2007)]

Integrand decomposition:





Fit-on-the cut

- fit m-point residues on m-ple cuts
- Cutting a loop propagator means

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

i.e. putting it on-shell

One-loop integrand reduction: implementations

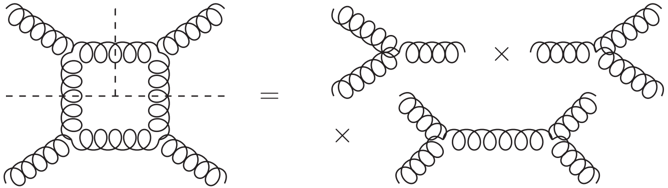
General-purpose implementations of one-loop integrand reduction:

- CUTTOOLS [Ossola, Papadopoulos, Pittau (2007)]
  - four-dimensional integrand reduction
  - extra-dimensional contributions in dim. regularization computed via process-independent (but theory-dependent) Feynman rules
- SAMURAI [Mastrolia, Ossola, Reiter, Tramontano (2010)]
  - d-dimensional integrand reduction
  - works with d dimensional integrands for any theory
- NINJA [T.P. (2014)]
  - semi-numerical integrand reduction via Laurent expansion
    Forde (2007), Badger (2008), P. Mastrolia, E. Mirabella, T.P. (2012)
  - faster and more stable integrand-reduction algorithm
  - used by GOSAM and MADLOOP (MADGRAPH5_AMC@NLO)

Generalized unitarity: loops from trees

Britto, Cachazo, Feng (2004), Giele, Kunszt, Melnikov (2008)

- Evaluating loop integrands on multiple cuts
  - the cut loop propagators are put on-shell
  - the integrand factorizes as a product of tree-level amplitudes



Loops from trees

We can compute the coefficients of loop amplitudes from products of tree-level amplitudes

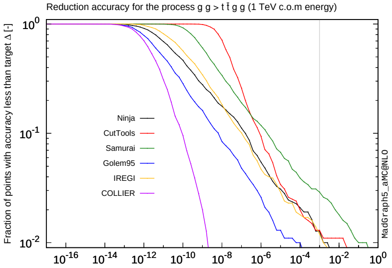
- implemented in BLACKHAT, NJET and several private codes

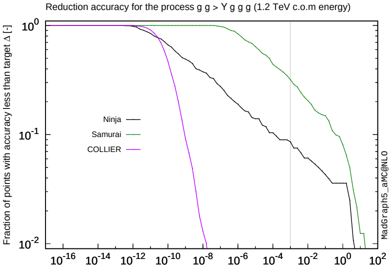
One-loop tools

- Master Integrals
  - FF [van Oldenborg (1990)]
  - LoopTools [Hahn et al. (1998)]
  - QCDLoop [Ellis, Zanderighi (2007), Carrazza, Ellis, Zanderighi (2016)]
  - OneLOop [van Hameren (2010)]
  - ...
- Reduction
  - integrand reduction (CutTools, Samurai, Ninja)
  - tensor reduction
    - Collier [Denner, Dittmaier (since 2003), Denner, Dittmaier, Hofer (2016)]
    - Golem95 [T. Binoth, J.-P. Guillet, G. Heinrich, E. Pilon, T. Reiter (2009), J.P. Guillet, G. Heinrich, J. von Soden-Fraunhofen (2014)]
    - Iregi (part of MadLoop)
    - ...

One-loop tools (reduction tools)

Testing reduction tools with MadLoop (courtesy of V. Hirschi)





	Pure reduction time* (x ≡ relative to Ninja)					
	$gg \to t\bar{t} + n\,g$			$gg \to Y + (n+1)\,g$		
Tool	$n=0$	$n=1$	$n=2$	$n=0$	$n=1$	$n=2$
Ninja	0.4 ms	5.3 ms	78 ms	2.2 ms	33 ms	1.4 s
CutTools	2.6 x	2.5 x	2.8 x	N/A	N/A	N/A
Samurai	5.0 x	3.9 x	4.3 x	4.1 x	4.3 x	6.3 x
Golem95	12 x	20 x	40 x	8.9 x	25 x	N/A
Iregi	14 x	51 x	150 x	25 x	175 x	N/A
Collier	2.1 x	2.6 x	2.8 x	1.3 x	2.9 x	5.6 x

*all tools but Collier require performing the reduction twice for estimating the numerical accuracy.

One-loop tools (cont.)

- One-loop packages
  - HELAC-NLO: numerical recursion + OPP reduction
  - FORMCALC: analytic generation + PV or integrand reduction
  - OPENLOOPS: recursive numerical generation of tensor integrands
    - reduction via COLLIER, CUTTOOLS, SAMURAI
  - MADLOOP (MADGRAPH5_AMC@NLO) alt. OpenLoops
    - reduction via NINJA, GOLEM95, IREGI, CUTTOOLS, SAMURAI, ...
  - GOSAM: analytic generation (with a two-loop extension)
    - reduction via NINJA, SAMURAI, GOLEM95
  - RECOLA: recursion relations + reduction via COLLIER
  - BLACKHAT and NJET: generalized unitarity
- Montecarlo tools (Born, real+subtraction, phase-space,...)
  - SHERPA, AMC@NLO, MADEVENT, POWHEG, HERWIG, PYTHIA,...

Integrand reduction and generalized unitarity at higher loops

Progress in integrand reduction at higher loop

- Integrand decomposition found with techniques of algebraic geometry (e.g. multivariate polynomial division)
  Y. Zhang (2012), P. Mastrolia, E. Mirabella, G. Ossola, T.P. (2012)
- It can be combined with generalized unitarity, diagrammatic approaches and purely algebraic techniques
  S. Badger, H. Frellesvig, P. Mastrolia, E. Mirabella, G. Ossola, A. Primo, Y. Zhang, T.P. (2011—now)
- First two-loop 5-point amplitude recently computed
  Badger, Frellesvig, Zhang (2013), Badger, Mogull, Ochirov, O'Connell (2015), Gehrmann, Henn, Lo Presti (2015)
- First two-loop 5-point Master Integrals have been computed
  Gehrmann, Henn, Lo Presti (2015), Papadopoulos, Tommasini, Wever (2015)
  $\Rightarrow$ **see K. Papadopoulos's talk**
- First two-loop 6-point amplitude recently computed
  Dunbar, Perkins, Warren (2016), Badger, Mogull, T.P. (2016)
- Functional reconstruction for 2-loop generalized unitarity T.P. (2016)

Analytic multi-leg calculations: kinematic variables

Hodges (2009), Badger, Frellesvig, Zhang (2013), Badger (2016)

- rational parametrization of the n-point phase-space and the spinor components using $3n-10$ momentum-twistor variables
- 5-point example $\to$ 5 variables $\{x_1,\ldots,x_5\}$

$$|1\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \qquad |1] = \begin{pmatrix}1\\ \frac{x_4-x_5}{x_4}\end{pmatrix}, \qquad x_k = x_k(s_{ij}, \mathrm{tr}(\gamma_5\, 1\,2\,3\,4))$$

$$|2\rangle = \begin{pmatrix}0\\1\end{pmatrix}, \qquad |2] = \begin{pmatrix}0\\x_1\end{pmatrix}, \qquad p_i^\mu = \frac{\langle i|\,\sigma^\mu\,|i]}{2}$$

$$|3\rangle = \begin{pmatrix}\frac{1}{x_1}\\1\end{pmatrix}, \qquad |3] = \begin{pmatrix}x_1\,x_4\\-x_1\end{pmatrix},$$

$$|4\rangle = \begin{pmatrix}\frac{1}{x_1}+\frac{1}{x_1\,x_2}\\1\end{pmatrix}, \qquad |4] = \begin{pmatrix}x_1\,(x_2\,x_3 - x_3\,x_4 - x_4)\\-\frac{x_1\,x_2\,x_3\,x_5}{x_4}\end{pmatrix},$$

$$|5\rangle = \begin{pmatrix}\frac{1}{x_1}+\frac{1}{x_1\,x_2}+\frac{1}{x_1\,x_2\,x_3}\\1\end{pmatrix}, \qquad |5] = \begin{pmatrix}x_1\,x_3\,(x_4-x_2)\\\frac{x_1\,x_2\,x_3\,x_5}{x_4}\end{pmatrix}.$$

Choosing an integrand basis

Badger, Mogull, T.P. (2016)

- Choosing an integrand basis:
  - the problem of finding an integrand basis is solved at any loop
  - the choice is however not unique
  - the complexity of the results can heavily depend of the choice
- Local integrands for 5- and 6-point 2-loop all-plus amplitudes
  - free of spurious singularities
  - smooth soft limits to lower-point integrands
  - infrared properties manifest at the integrand level

  $\Rightarrow$ simpler results

  ✗ …but no general algorithm for building one (yet)

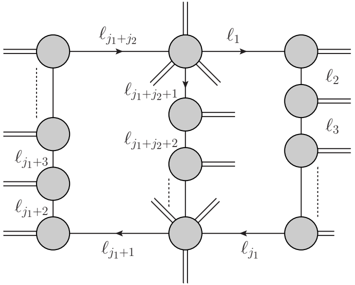
Two-loop unitarity cuts in d dimensions

Badger, Frellesvig, Zhang (2013)

- d-dim. dependence of loops $k_i^\mu \Rightarrow$ embed k_i^μ in $\mathcal{D}$ dimensions $(\mathcal{D} > 4)$
- unitarity cuts $\ell_i^2 = 0 \Rightarrow$ explicit $\mathcal{D}$-dim. representation of loop components
- describe internal on-shell states with $\mathcal{D}$-dim. spinor-helicity formalism
- additional gluon states as $d_s - \mathcal{D}$ scalars ($d_s = 4, d$ in FDH, tHV)

$\mathcal{D} = 6$ sufficient up to two loops

also useful for functional reconstruction



Finite fields and functional reconstruction techniques

Finite fields and functional reconstruction

- Calculation of multi-leg amplitudes
  - several independent invariants
  - large intermediate expressions
- Functional reconstruction from numerical evaluation
  - sidesteps issue of large intermediate expressions
  - evaluation over finite fields $\mathcal{Z}_p = \{1, \ldots, p-1\}$ (p prime)
    - fast but exact
    - first proposed for IBPs [von Manteuffel, Schabinger (2014)]

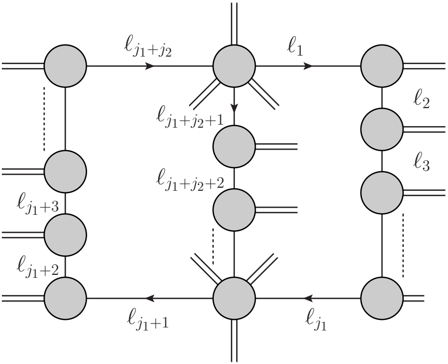
Developed an efficient algorithm for functional reconstruction [T.P. (2016)]

- works on (dense) multivariate polynomials and rational functions
- implemented in C++ code (proof of concept)
- the input is a numerical procedure computing a function
- the output is its analytic expression

Finite fields and functional reconstruction

T.P. (2016)

- Scattering amplitudes over finite fields
  - spinor-helicity
  - tree-level recursion
  - two-loop d-dim. gen. unitarity

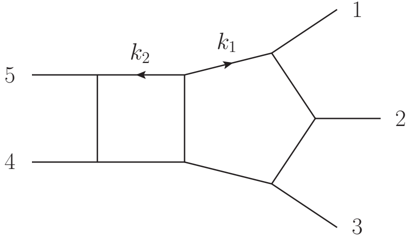


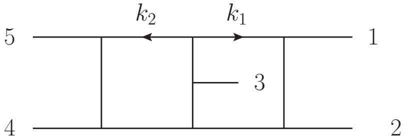
use efficient numerical techniques for analytic calculations

two-loop unitarity cuts from Berends-Giele off-shell currents

Finite fields and functional reconstruction: examples

- five-gluon on-shell integrands of maximal cuts ($\equiv$ top-level topology) for





(for a complete set of helicities)

Finite fields and functional reconstruction

- penta-box

Helicity	Non-vanishing coeff.	Max. terms	Max. degree	Avg. non-zero terms
$(1^+,2^+,3^+,4^+,5^+)$	14	19	8	15.00
$(1^-,2^+,3^+,4^+,5^+)$	27	443	19	152.96
$(1^+,2^-,3^+,4^+,5^+)$	37	1977	24	674.97
$(1^+,2^+,3^+,4^-,5^+)$	61	474	18	184.05
$(1^-,2^-,3^+,4^+,5^+)$	35	1511	24	278.77
$(1^-,2^+,3^+,4^+,5^-)$	79	7027	34	1112.82
$(1^+,2^+,3^+,4^-,5^-)$	18	19	8	15.00
$(1^-,2^+,3^-,4^+,5^+)$	41	2412	22	368.41
$(1^+,2^-,3^+,4^-,5^+)$	85	18960	42	3934.96
$(1^-,2^+,3^+,4^-,5^+)$	85	10386	37	1803.52

- double-pentagon

Helicity	Non-vanishing coeff.	Max. terms	Max. degree	Avg. non-zero terms
$(1^+,2^+,3^+,4^+,5^+)$	104	1937	26	626.39
$(1^-,2^+,3^+,4^+,5^+)$	104	1449	27	601.43
$(1^+,2^+,3^-,4^+,5^+)$	104	1554	23	642.90
$(1^-,2^-,3^+,4^+,5^+)$	99	1751	26	739.05
$(1^+,2^-,3^-,4^+,5^+)$	104	2524	24	923.71
$(1^-,2^+,3^+,4^+,5^-)$	104	1838	27	823.00
$(1^-,2^+,3^+,4^-,5^+)$	104	1307	24	630.48

Summary & Outlook

Summary and Outlook

Summary

- One-loop multi-leg calculations
  - are automated by many public and private tools
  - current focus is performance, numerical stability and reliability
- High-multiplicity ($2 \to 3$ or higher) processes at two-loops
  - first MIs available
  - first amplitudes using integrand reduction and generalized unitarity
  - use of functional-reconstruction and finite-field techniques

Outlook

- complete two-loop five-point amplitudes for arbitrary helicities
- broader application of multivariate functional reconstruction (good integrand-basis, IBPs, diagrammatic techniques, . . .)

THANKS!