Next-to-Leading Order Scattering Amplitudes with Open Loops

Philipp Maierhöfer

Institute for Theoretical Physics
University of Zürich

LHCphenonet Annual Meeting
Lumley Castle, 20 March 2012

In Collaboration with
Fabio Cascioli and Stefano Pozzorini

Outline

State of the Art @ NLO

In the last few years, new theoretical developments lead to tremendous progress in the field of NLO calculations.

$pp \to W^+W^- b\bar{b}$	[Denner, Dittmaier, Kallweit, Pozzorini '10]
	[Bevilacqua, Czakon, van Hameren, Papadopoulos, Worek '11]
$pp \to t\bar{t}b\bar{b}$	[Bredenstein, Denner, Dittmaier, Pozzorini '08, '09, '10]
	[Bevilacqua, Czakon, Papadopoulos, Pittau, Worek '09]
$pp \to t\bar{t}jj$	[Bevilacqua, Czakon, Papadopoulos, Pittau, Worek '10]
$pp \to W^\pm W^\pm + 2j$	[Melia, Melnikov, Rontsch, Zanderighi '10]
	[Greiner, Heinrich, Mastrolia, Ossola, Reiter, Tramontano '12]
$pp \to W^\pm + 3j$	[Ellis, Melnikov, Zanderighi '09]
$pp \to \gamma^*/Z/W^\pm + 3j$	[Berger, Bern, Dixon, Febres Cordero, Forde, Gleisberg, Ita, Kosower, Maître '09]
$pp \to Z/W^\pm + 4j$	[—"— '09, '10]
$pp \to 4j$	[Bern, Diana, Dixon, Febres Cordero, Hoeche, Kosower, Ita, Maître, Ozeren '11]
$pp \to b\bar{b}b\bar{b}$	[Greiner, Guffanti, Reiter, Reuter '11]
$pp \to W\gamma\gamma j$	[Campanario, Englert, Rauch, Zeppenfeld '11]
$e^+e^- \to 7j$	[Becker, Goetz, Reuschle, Schwan, Weinzierl '11]

However, there's a trade-off between automation, flexibility, and speed.

Can the advantages of the different approaches be combined in a single algorithm?

Feynman Diagrams and Colour factorisation

Tree and one-loop amplitudes are handled as sums of Feynman diagrams

$$\mathcal{M} = \sum_{d} \mathcal{M}^{(d)}, \qquad \delta\mathcal{M} = \sum_{d'} \delta\mathcal{M}^{(d')}$$

Colour and helicity summed scattering probability densities

$$\mathcal{W} = \sum_{\mathrm{hel,col}} |\mathcal{M}|^2, \qquad \delta\mathcal{W} = \sum_{\mathrm{hel,col}} 2\,\mathrm{Re}(\mathcal{M}^*\delta\mathcal{M})$$

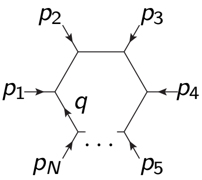
Diagrams factorise in colour factors and colour stripped amplitudes

$$\mathcal{M}^{(d)} = \mathcal{C}^{(d)}\mathcal{A}^{(d)}, \qquad \delta\mathcal{M}^{(d')} = \mathcal{C}^{(d')}\delta\mathcal{A}^{(d')}$$

Algebraic colour reduction and summation once per process

colour sums at zero cost

From amplitudes to scalar integrals



$$= \int d^d q \frac{\mathcal{N}(q)}{D_0 \; D_1 \; \dots \; D_{N-1}}, \qquad D_i = \left(q + \sum_{\ell=0}^{i} p_\ell\right)^2 - m_i^2$$

Tensor integral reduction

Reduce amplitude to a linear combination of scalar basis integrals

On-shell methods

$$\int d^d q \left[\sum_{i_1} \frac{a_{i_1}}{D_{i_1}} + \sum_{i_1, i_2} \frac{b_{i_1 i_2}}{D_{i_1} D_{i_2}} + \sum_{i_1, i_2, i_3} \frac{c_{i_1 i_2 i_3}}{D_{i_1} D_{i_2} D_{i_3}} + \sum_{i_1, i_2, i_3, i_4} \frac{d_{i_1 i_2 i_3 i_4}}{D_{i_1} D_{i_2} D_{i_3} D_{i_4}} \right]$$

Tensor integral reduction combined with off-shell current recursion can compete with OPP in *n*-gluon scattering with up to 10 gluons.

[van Hameren '09]

Tensor integral reduction

Separate tensor coefficients from tensor integrals.

$$\mathcal{A} = \sum_{r=0}^{R} \mathcal{N}_r^{\mu_1\cdots\mu_r} \cdot \int d^d q \, \frac{q_{\mu_1} \cdots q_{\mu_r}}{D_0 \; D_1 \; \ldots \; D_{N-1}}$$

- Covariant decomposition in tensor monomials built from $g^{\mu\nu}$ and p_i^μ.
- Reduce tensor integrals to scalar basis integrals.
  [Melrose], [Passarino, Veltman], [Denner, Dittmaier]
- Can be implemented in a numerically stable way (e. g. expansion in small Gram determinants):
  We use `Collier`, a private library by Denner and Dittmaier.

"Traditional" approach: construct $\mathcal{N}_r^{\mu_1\cdots\mu_r}$ analytically

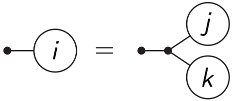
huge expressions & expensive algebraic simplifications limit applicability

OPP & Tree Generator

OPP directly extracts coefficients of the scalar basis integrals.

- Need multiple numerical evaluations of $\mathcal{N}(q)$ for complex q.
- Public implementations: `CutTools` [Ossola, Papadopoulos, Pittau]
  `Samurai` [Mastrolia, Ossola, Reiter, Tramontano]
- $\mathcal{N}(q)$ can be calculated by a generator for tree-level amplitudes.
  (diagrammatic like `MadLoop`, or current recursion like `HELAC-NLO`)

Wave functions w^α of "sub-trees" are 4-tuples
(for the spinor/Lorentz index) which are built by
recursivly connecting lower sub-trees with vertices and propagators.



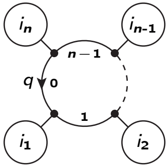
$$w^\beta(i) = \frac{X^\beta_{\gamma\delta}}{p_i^2 - m_i^2}\, w^\gamma(j)\, w^\delta(k)$$

external lines are not depicted

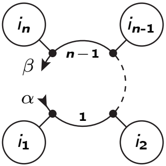
$X^\beta_{\gamma\delta}$ describes the interaction of i, j, k

Recursive Numerator Construction

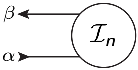
A one-loop diagram is an ordered set of sub-trees $\mathcal{I}_n = \{i_1, \ldots, i_n\}$



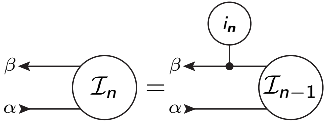
$\xrightarrow{\text{cut } D_0}$ $\mathcal{N}^{\beta}_{\alpha}(\mathcal{I}_n; q) =$



$\equiv$



Connect sub-trees along the loop to build the numerator $\mathcal{N} = \mathcal{N}^{\alpha}_{\alpha}$



$$\mathcal{N}^{\beta}_{\alpha}(\mathcal{I}_n; q) = X^{\beta}_{\gamma\delta}\, \mathcal{N}^{\gamma}_{\alpha}(\mathcal{I}_{n-1}; q)\; w^{\delta}(i_n)$$

Separation of the loop momentum q ...

$$\sum_{r=0}^{n} \mathcal{N}^{\beta}_{\mu_1\ldots\mu_r;\alpha}(\mathcal{I}_n)\, q^{\mu_1} \ldots q^{\mu_r} = (Y^{\beta}_{\gamma\delta} + q^{\nu} Z^{\beta}_{\nu;\gamma\delta}) \sum_{r=0}^{n-1} \mathcal{N}^{\gamma}_{\mu_1\ldots\mu_r;\alpha}(\mathcal{I}_{n-1})\, q^{\mu_1} \ldots q^{\mu_r}\, w^{\delta}(i_n)$$

... leads to a recursion formula for the coefficients $\mathcal{N}^{\beta}_{\mu_1\ldots\mu_r;\alpha}$.

Open-Loops recursion

“Open loops” polynomials in q can be built recursively

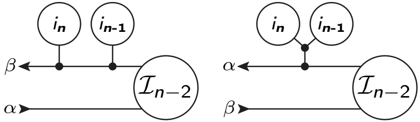
$$\mathcal{N}^{\beta}_{\mu_1\ldots\mu_r;\alpha}(\mathcal{I}_n) = \left[Y^{\beta}_{\gamma\delta}\, \mathcal{N}^{\gamma}_{\mu_1\ldots\mu_r;\alpha}(\mathcal{I}_{n-1}) + Z^{\beta}_{\mu_1;\gamma\delta}\, \mathcal{N}^{\gamma}_{\mu_2\ldots\mu_r;\alpha}(\mathcal{I}_{n-1}) \right] w^{\delta}(i_n)$$

$\Rightarrow$ **retains full loop momentum dependence.**

Once the polynomials are known, multiple evaluations of $\mathcal{N}(q) = \sum_{r=0}^{n} \mathcal{N}^{\alpha}_{\mu_1\ldots\mu_r;\alpha} q^{\mu_1}\ldots q^{\mu_r}$ are very fast. $\Rightarrow$ boosts OPP

On the other hand $\mathcal{N}^{\alpha}_{\mu_1\ldots\mu_r;\alpha}$ are the coefficients of the tensor integrals.

Open loops can be interfaced with both tensor integrals and OPP in a straight forward way.



Recycling:
Lower-point open-loops can be shared between diagrams if the cut it put appropriately.

Optimising OPP calls: efficient helicity summation

Once a tensor integral has been calculated, it can be reused in all diagrams with the same set of denominators.

The same degree of optimisation can be achieved for OPP reduction: Perform **interference** with the Born amplitude $\mathcal{M}$, **colour and helicity sums** and the sum over the set of diagrams Δ with identical denominator structure on the level of open-loop coefficients ***before* OPP reduction**.

$$\delta\mathcal{W}^{\Delta} = \sum_{\mathrm{hel,col}} 2\,\mathrm{Re}\left[\mathcal{M}^*\Big(\sum_{d'\in\Delta}\delta\mathcal{M}^{(d')}\Big)\right]$$

$$\delta\mathcal{W}^{\Delta}_{\mu_1\dots\mu_R} = \sum_{\mathrm{hel,col}} 2\times\left[\mathcal{M}^*\Big(\sum_{d'\in\Delta}\mathcal{C}^{(d')}\mathcal{N}^{(d')}_{\mu_1\dots\mu_R}\Big)\right]$$

Unpolarised, colour summed numerator $\mathcal{N}^{\Delta}(q) = \delta\mathcal{W}^{\Delta}_{\mu_1\dots\mu_R} q^{\mu_1}\dots q^{\mu_R}$

minimises the number of OPP calls
and leads to very efficient helicity sums.

Implementation

User input: process definition file

- `FeynArts` generates Feynman diagrams.
- `Mathematica` organises recursion and recycling, reduces colour factors and generates `Fortran 90` code.
- Numerical routines for QCD corrections to Standard Model processes implemented in `Fortran 90`.
- Symmetrising tensors keeps the number of components manageable.
- Rational terms R_2 are calculated using the tree generator. [Draggiotis, Garzelli, Papadopoulos, Pittau]
- Executable linked to `Collier`, `CutTools` and `Samurai`.

Consistency checks

- UV and IR cancellations
- Tensor integrals vs. OPP reduction
- "pseudo-tree": fix loop momentum and compare to tree generator

Speed and Flexibility

Performance studies for all non-trivial processes of the Les Houches priority list (Intel i5-750, single thread, compiled with ifort 10.1).

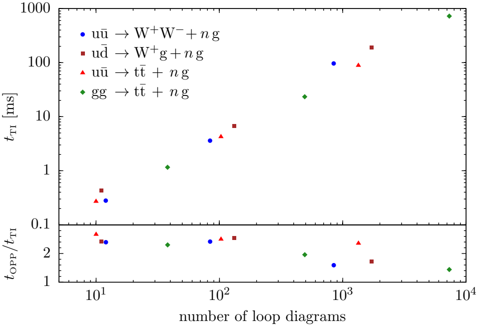
t_{code} = code generation & compilation (lots of room for improvement here);

size of the process library;

t_{TI} = time for a single phase space point using tensor integrals (unpolarised, single helicity for t/W);

t_{OPP} = the same with CutTools

Process	t_{code}/s	size/MB	t_{TI}/ms
$u\bar{u} \to t\bar{t}$	2.2	0.1	0.27
$u\bar{u} \to W^+W^-$	7.2	0.1	0.28
$u\bar{d} \to W^+g$	4.2	0.1	0.43
$gg \to t\bar{t}$	5.4	0.2	1.16
$u\bar{u} \to t\bar{t}g$	13	0.4	4.2
$u\bar{u} \to W^+W^-g$	40	0.4	3.6
$u\bar{d} \to W^+gg$	24	0.5	6.7
$gg \to t\bar{t}g$	53	1.2	23.4
$u\bar{u} \to t\bar{t}gg$	236	3.6	88.4
$u\bar{u} \to W^+W^-gg$	382	2.5	96.4
$u\bar{d} \to W^+ggg$	366	4.2	190.5
$gg \to t\bar{t}gg$	3005	16.0	725.0



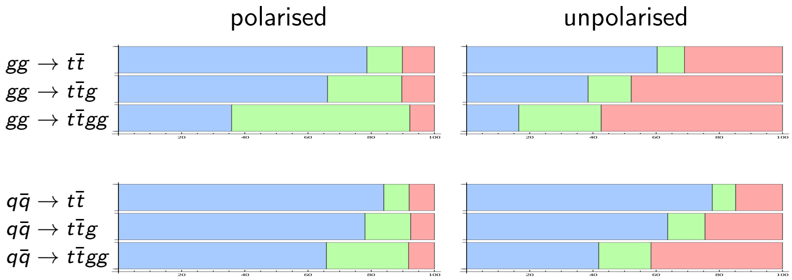
Timimg studies

single helicity: time for tensor reduction $\gg$ time for coefficients

full helicity sum: time for tensor reduction $\approx$ time for coefficients

For $2 \to 4$ processes

- full helicity sums cost only a factor ~ 2
- tensor integral reduction and OPP performance is similar



fractions of total runtime for scalar integrals, tensor reduction, coefficients

Numerical stability

The numerical precision can be estimated by a scaling test:

$$m_i \to \xi m_i,\ p_i^\mu \to \xi p_i^\mu \quad \text{leads to} \quad \delta\mathcal{W} \to \delta\mathcal{W}' = \xi^K \delta\mathcal{W}$$

$$\Rightarrow \text{ precision } \Delta = \left| \frac{\xi^{-K}\delta\mathcal{W}'}{\delta\mathcal{W}} - 1 \right|, \quad \text{rsp. } d = -\log_{10}\Delta \text{ decimal digits.}$$

Sample of 10^6 homogeniously distributed phase space points;

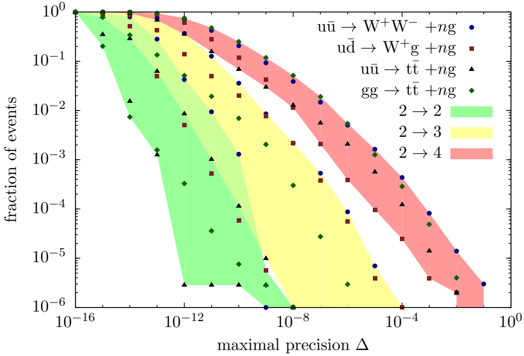
$\sqrt{s} = 1$ TeV

$p_T > 50$ GeV

$\Delta R_{ij} > 0.5$

using tensor integrals,

double precision



Summary

Open loops – A new algorithm for one-loop amplitudes

- Diagrammatic, tree-like recursion for loop momentum polynomials
- Implemented for QCD corrections to SM processes
- Fully automated:
  process definition $\rightarrow$ compact code within seconds/minutes
- Interfaced with tensor integral and OPP reduction
- Very fast: 0.1–1 s/PS-point for colour & helicity summed $2 \rightarrow 4$
- Numerically stable when using tensor integrals