A recursive one-loop algorithm for many-particle amplitudes

Philipp Maierhöfer

Institute for Theoretical Physics
University of Zürich

RADCOR 2011
Mamallapuram, 27 September 2011

In Collaboration with
Fabio Cascioli and Stefano Pozzorini

Outline

State of the Art @ NLO

$pp \to W^+W^-b\bar{b}$	[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]
$pp \to W^\pm + 3j$	[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]
$pp \to b\bar{b}b\bar{b}$	[Binoth, Greiner, Guffanti, Guillet, Reiter, Reuter '09]
$pp \to W\gamma\gamma j$	[Campanario, Englert, Rauch, Zeppenfeld '11]

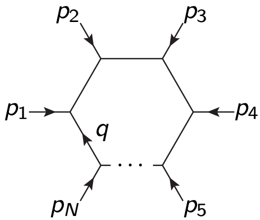
On-shell methods

Tensor integral reduction
- analytical coefficients
- numerical coefficients

numerical integration

$\longrightarrow$ Tensor integral reduction can compete with on-shell methods up to 10 external legs. [van Hameren '09]

It's all about tensor reduction



$$= \int d^d q \, \frac{G_0 + G_1^{\mu_1} q_{\mu_1} + \; \ldots \; + G_R^{\mu_1 \ldots \mu_R} q_{\mu_1} \ldots q_{\mu_R}}{D_0 \; D_1 \; \ldots \; D_{N-1}}$$

Tensor integral reduction

Reduce amplitude

to a linear combination

of scalar integrals

On-shell methods

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

$$D_i = \left(q + \sum_{\ell=0}^{i} p_\ell \right)^2 - m_i^2 + i\delta$$

Tensor integral reduction

- Generate Feynman diagrams and insert Feynman rules.
- Separate tensor coefficients from tensor integrals.

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

- 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.

- Construction of explicit tensor components

coefficient calculated . . .

analytically in d dimensions

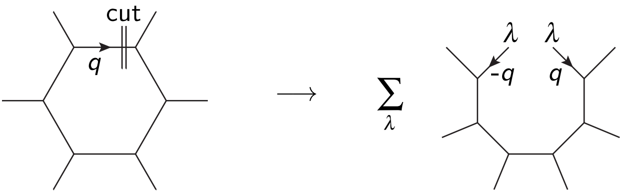
numerically in 4 dimensions (need rational terms R_2)

OPP method from the users' point of view

- Public implementations available:
  - CutTools [Ossola, Papadopoulos, Pittau]
  - Samurai [Mastrolia, Ossola, Reiter, Tramontano]
- Provide a Fortran subroutine which evaluates the numerator $N(q)$ numerically for given (complex) q.
- $N(q)$ can be calculated by a generator for tree-level amplitudes.
- OPP routines extract coefficients of the scalar basis integrals.
- Coefficient extraction can be numerically unstable.
- Calculate scalar integrals (several libraries available).

Construction of the OPP numerator function

To construct the numerator function, “cut” one of the loop propagators.



- Two additional external legs with momenta q and $-q$.
- Remove denominators from loop propagators.
- Step-by-step attach vertices and propagators to build tree wave functions.

Diagram by diagram, sharing common structures between diagrams [like MadGraph]	Current recursion using Dyson-Schwinger equations [like HELAC-Phegas]

Tensor coefficients can be used with tensor integrals and OPP

If the tensor coefficients $G_r^{\mu_1\ldots\mu_r}$ are known, the numerator function $N(q)$ can be easily calculated by contracting the coefficients with direct products of the loop momentum.

$$N(q) = \sum_{r=0}^{R} G_r^{\mu_1\ldots\mu_r} \, q_{\mu_1} \, \ldots \, q_{\mu_r}$$

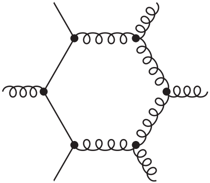
$\Rightarrow$ If one has an efficient way to calculate $G_r^{\mu_1\ldots\mu_r}$,
one can use it for both, contraction with tensor integrals, or as input for the OPP method.

Reconstructing G from $N(q)$ for several q is always possible, but the performance is only acceptable to cure numerical instabilities in exceptional phase space points. [Heinrich, Ossola, Reiter, Tramontano]

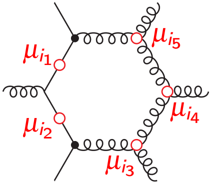
But: $\boxed{\textbf{Coefficients can be constructed in a tree-like way!}}$

Tree-like construction of tensor coefficients

Separate coefficient G from the tensor integrals



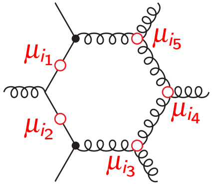
$$= \sum_{\substack{\text{subsets of} \\ \{i_1,\ldots,i_R\} \\ = \{1,\ldots,r\}}}$$



$$\int d^d q \; \frac{q_{\mu_1} \ldots q_{\mu_r}}{D_0 \; D_1 \; \ldots \; D_{N-1}}$$

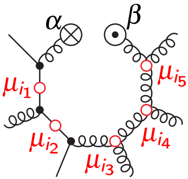
$$G \;=\; \{G_0,\; G_1^{\mu_1},\; \ldots,\; G_R^{\mu_1 \cdots \mu_R}\}$$

Open the loop and choose build direction



G

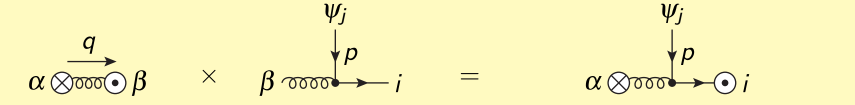
$$= \; \delta_\beta{}^\alpha$$



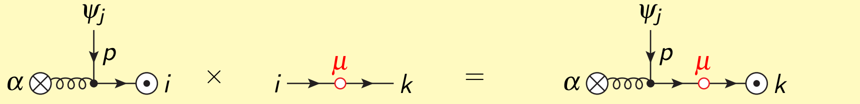
$G_\alpha{}^\beta$

Step-by-step construction of tensor coefficients

Just like in a tree generator, connection rules for the vertices and propagators of the theory are the universal building blocks for loop structures.



$G_\alpha{}^\beta = \delta_\alpha{}^\beta$ $\qquad -ig_s(\gamma_\beta)_{ij}$ $\qquad G_\alpha{}^i = -ig_s(\gamma_\alpha)_{ij}\psi_j$

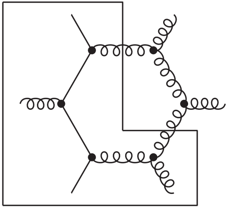


$G_\alpha{}^i = -ig_s(\gamma_\alpha)_{ij}\psi_j$ $\qquad (\gamma_\mu)_{ki}q^\mu + (\not p + m)_{ki}$ $\qquad G_\alpha{}^k = \{G_\alpha{}^i(\not p + m)_{ki},\ G_\alpha{}^i(\gamma_\mu)_{ki}\}$

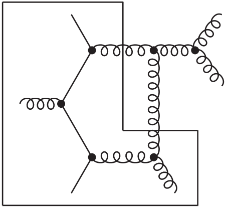
Vertices and propagators which contain the loop momentum q raise the rank of the coefficient, i. e. they add an open Lorentz index, but they also contribute to the equal rank part.

Sharing loop structures between diagrams

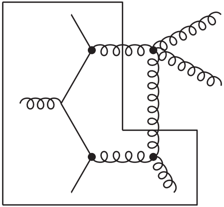
Loop structures can be shared between diagrams, if the loop momentum is chosen in the same way in the corresponding diagrams.



parent



child 1



child 2

⇒ Exploit the freedom of putting the cut and choosing the direction to maximise recyclability.

- In QCD cutting a gluon line assures that all child diagrams exist.
- Choose a cutting algorithm which assures that merging the last two vertices which are connected to the loop does not change the cut position and direction.

Pseudo-tree consistency check

Instead of closing the loop by taking the trace $G = G_\alpha{}^\alpha$ one can attach external legs $\varepsilon_{1,2}$ to the chain and contract with a fixed "loop" momentum

$$A \;=\; \varepsilon_1^\alpha(q)\left[G_\alpha{}^\beta \cdot Q\right]\varepsilon_{2\beta}(-q), \quad \text{where} \quad Q = \{1,\; q^{\mu_1},\; \ldots,\; q^{\mu_1}\ldots q^{\mu_R}\}.$$

A is the amplitude of the tree-level diagram obtained by cutting the loop diagram and can be compared to the same diagram calculated by a tree generator.

$\Rightarrow$ Checks the implementation of vertices and propagators for the loop structures.

Remarks

The tree generator has been thoroughly checked against MadGraph.

One can easily switch between tensor integrals and OPP for consistency checks, numerical stability performance studies.

Since the tensor integrals are totally symmetric, only the symmetric part of the coefficient has to be constructed. For rank up to R the number of components is $\binom{R+4}{4}$ instead of $\frac{1}{3}(4^{R+1}-1)$.

max. rank	0	1	2	3	4	5	6
components	1	5	15	35	70	126	210

Each component is a 4×4 matrix for the spinor/Lorentz index of the incoming and outgoing particles of the cut loop (except for scalars/ghosts).

Increasing tensor rank decreases the effectivness of sharing loop structures, because the highest rank (and therefore most expensive) structures cannot be reused.

Organisation of the calculation

FeynArts
- diagram generation

User input
- Process definition file

Fortran templates

Code generator (Mathematica)

Tree generator
- tree-level process
- tree structures connected to loops
- real corrections
- UV counter terms
- rational terms of type 2
- colour and helicity correlations for Catani-Seymour dipoles
- full Standard Model
- unitary gauge and Feynman gauge
- complex masses and momenta
- identification of common structures

Loop generator
- QCD corrections for Standard Model processes
- complex masses
- dipole I-operators
- optimised identification of common structures

Fortran 90 code

CutTools

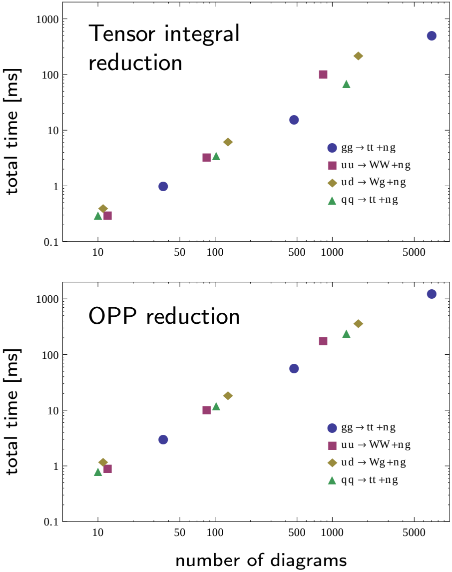
COLI (private tensor integral library by Denner & Dittmaier)

Runtime vs. number of diagrams

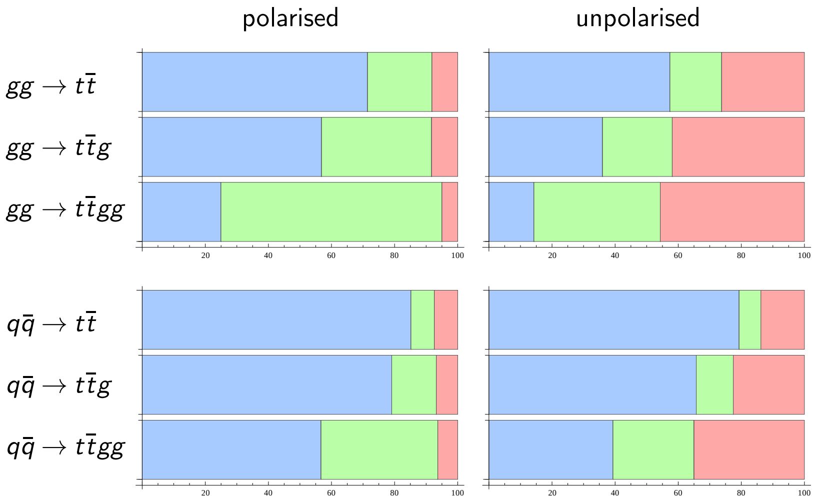
CPU cost for virtual correction;
polarised;
full colour sums;
on Intel Core i5-750;
ifort 10.1 -O2;
preliminary

Process	TIR	OPP
$gg \to t\bar{t}$	1.0	3.0
$gg \to t\bar{t}g$	15.5	57.0
$gg \to t\bar{t}gg$	508.8	1244.0
$u\bar{u} \to W^+W^-$	0.3	0.9
$u\bar{u} \to W^+W^-g$	3.3	10.3
$u\bar{u} \to W^+W^-gg$	102.3	176.0
$u\bar{d} \to W^+g$	0.4	1.2
$u\bar{d} \to W^+gg$	6.3	18.7
$u\bar{d} \to W^+ggg$	220.0	370.0
$q\bar{q} \to t\bar{t}$	0.3	0.8
$q\bar{q} \to t\bar{t}g$	3.5	12.0
$q\bar{q} \to t\bar{t}gg$	69.2	243.0

<1s for 10^4 diagrams, all processes from the Les Houches wishlist included.



Benchmark results using tensor intgerals



fractions of total runtime for scalar integrals, tensor reduction, coefficients;
helicity sums contain one state per W/top-quark
(decay into left-handed massless fermions)

Remarks and Outlook

- Code generation is fast: $\mathscr{O}(1\,\text{min})$
- Executables are small: $\mathscr{O}(10\,\text{MB})$

$\Big\}$ for a $2 \to 4$ process

- Diagrammatic approach allows for colour factorisation and therefore colour summing at low CPU cost.
- Helicity sampling is possible to further reduce CPU cost.
- UV counter terms, IR subtractions and rational terms only partially implemented and tested.
- Optimise construction of tensor integral components, maybe without first constructing a covariant decomposition. [Hofer]
- Detailed numerical stability studies needed.
- Extend loop generator to full Standard Model.
- Interface with an event generator.

Conclusions

- We implemented a numerical algorithm to construct tensor integral coefficients for NLO calculations.
- Coefficients are constructed in a tree-like way.
- Easily extensible by implementing new vertices and propagators.
- Can be used with tensor integral reduction and OPP reduction.
- Common loop structures are shared between diagrams.
- Performance studies for $2 \to 4$ processes look extremly promising and there is still potential for optimisation.
- CPU time is dominated by tensor integral rsp. OPP reduction.
- There is still a lot to do, before we can produce first physical results.