Recent developments in Open Loops

Philipp Maierhöfer

Institute for Theoretical Physics
University of Zürich

HP2: High Precision for Hard Processes
München, 4 September 2012

In Collaboration with
F. Cascioli, S. Pozzorini,
F. Krauss, F. Siegert and J. Thompson

Outline

1. Basics
   - Feynman Diagrams and Colour factorisation
   - Tensor Integrals and OPP Reduction

2. The Open Loops Algorithm
   - Recursive Construction of Loop Diagram Numerators
   - Implementation
   - Performance and Numerical Stability
   - Checks Against Independent Code

3. Towards LHC Phenomenology with Open Loops
   - Automation of NLO Calculations
   - Open Loops + Sherpa

State of the Art @ NLO

The list of accomplished NLO calculations is steadily growing . . .

$pp \to W^+W^-b\bar{b}$	[Denner, Dittmaier, Kallweit, Pozzorini ‘11]
	[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]
$pp \to t\bar{t}t\bar{t}$	[Bevilacqua, Worek ‘12]
$pp \to W^\pm W^\pm + 2j$	[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, ‘10]
$pp \to Z/W^\pm + 4j$	[Berger, Bern, Dixon, Febres Cordero, Forde, Gleisberg, Ita, Kosower, Maître ‘10, ‘11]
$pp \to 4j$	[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]

Various techniques have been used to perform the calculations.
Each approach has its specific advantages and disadvantages.
What we want is high speed, numerical stability and wide applicability with minimal human interaction.

Diagrammatic approach 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

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

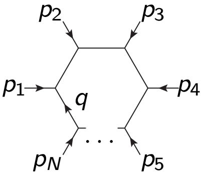
Diagrams factorise in colour factors and colour stripped amplitudes

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

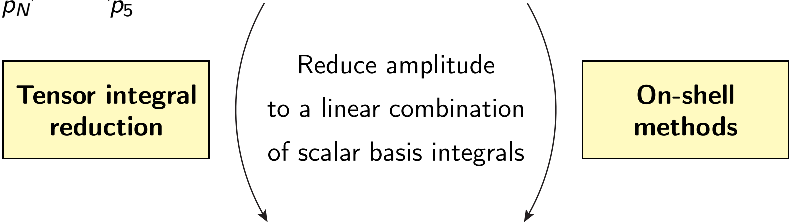
Algebraic colour reduction and summation only once per process

Feynman Diagrams ⇒ colour sums at zero cost

From Loop Amplitudes to Scalar Integrals



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



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 on-shell methods in n-gluon scattering with up to 10 gluons. [van Hameren '09]

Tensor Integral Reduction

Separate tensor coefficients from tensor integrals.

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

- 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; Binoth et al.; Fleischer, Riemann; & many others]
- Numerically unstable in pathological phase space regions, but the instabilities are understood and can be cured (e. g. expansion in small Gram determinants [Denner, Dittmaier])
- We use `Collier`, a private library by Denner and Dittmaier with an interface by Hofer to build a tensor component representation.

"Traditional" approach: construct $\mathcal{N}_r^{\mu_1 \cdots \mu_r}$ analytically in $d = 4 - 2\epsilon$.
Huge expressions & expensive algebraic simplifications limit applicability.

Open Loops: **Recursive numerical construction of $\mathcal{N}_r^{\mu_1 \cdots \mu_r}$ in $d = 4$**

OPP reduction & Tree Generator

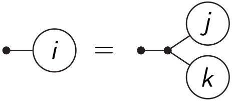
OPP directly extracts coefficients of the scalar basis integrals.

- 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.

Numerical instabilities are yet to be understood → quadruple precision

---

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, starting from external legs.



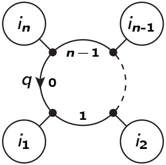
$$w^\beta(i) = \frac{X^\beta_{\gamma\delta}}{p_i^2 - m_i^2}\, w^\gamma(j)\, w^\delta(k)$$

external lines are not depicted

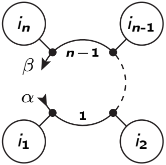
$X^\beta_{\gamma\delta}$ describes the interaction of i, j, k

From Tree Structures to Open Loops

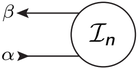
A one-loop diagram is an ordered set of sub-trees $\mathcal{I}_n = \{i_1, \ldots, i_n\}$



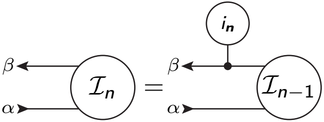
$$\xrightarrow{\text{cut } D_0} \quad \mathcal{N}^\beta_\alpha(\mathcal{I}_n; q) =$$



$$\equiv$$



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 . . .

$$\mathcal{N}^\beta_\alpha(\mathcal{I}_n; q) = \sum_{r=0}^{n} \mathcal{N}^\beta_{\mu_1\ldots\mu_r;\alpha}(\mathcal{I}_n)\, q^{\mu_1} \ldots q^{\mu_r}, \qquad X^\beta_{\gamma\delta} = Y^\beta_{\gamma\delta} + q^\nu Z^\beta_{\nu;\gamma\delta}$$

. . . 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

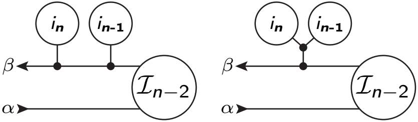
$$\mathcal{N}^{\beta}_{\mu_1\ldots\mu_r;\alpha}(\mathcal{I}_n) = \left[Y^{\beta}_{\gamma\delta}\, \mathcal{N}^{\gamma}_{\mu_1\ldots\mu_r;\alpha}(\mathcal{I}_{n-1}) + Z^{\beta}_{\mu_1;\gamma\delta}\, \mathcal{N}^{\gamma}_{\mu_2\ldots\mu_r;\alpha}(\mathcal{I}_{n-1}) \right] w^{\delta}(i_n)$$

⇒ **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. ⇒ 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.

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**.

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

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

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

minimises the number of OPP calls and leads to

very efficient helicity sums.

Implementation

User input: process definition file

- FeynArts [Hahn] 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 $\mathcal{N}^{\beta}_{\mu_{\mathbf{1}}\ldots\mu_{\mathbf{r}};\alpha}$ keeps the number of components manageable.
- Rational terms R_2 are calculated using the tree generator. [Draggiotis, Garzelli, Malamos, Papadopoulos, Pittau ‘09, ‘10; Shao, Zhang, Chao ‘11]
- No user interaction required: process definition → compiled library.

Consistency checks

- UV/IR cancellations and Ward identities
- 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).

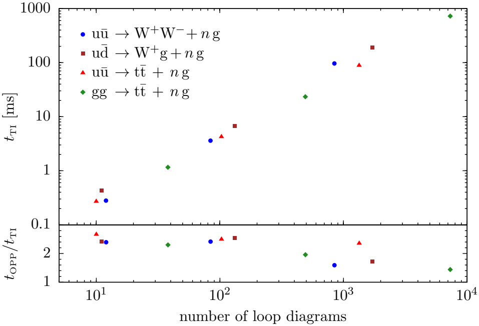
t_{code} = code generation & compilation (lots of room for improvement here);

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	1.2
$u\bar{u} \to t\bar{t}g$	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
$u\bar{u} \to t\bar{t}gg$	236	3.6	88
$u\bar{u} \to W^+W^-gg$	382	2.5	96
$u\bar{d} \to W^+ggg$	366	4.2	191
$gg \to t\bar{t}gg$	3005	16	725



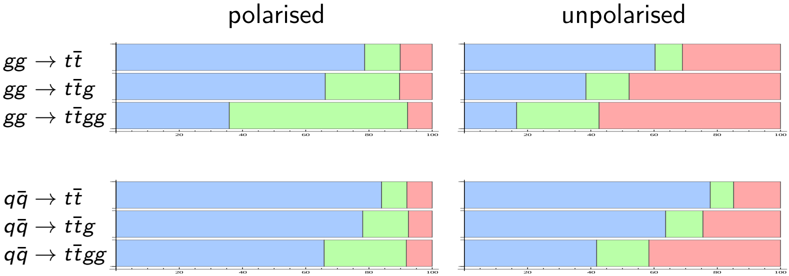
Timing studies

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 (here: instead of 16)
- 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:

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

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

Sample of 10^6 homogeniously distributed phase space points;

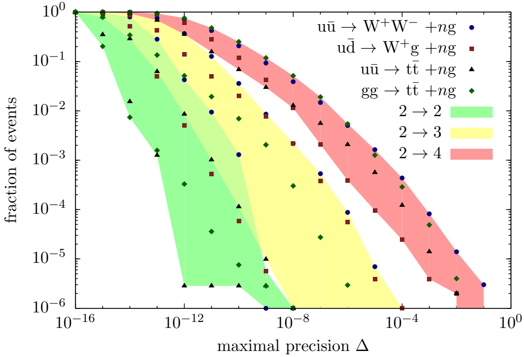
$\sqrt{s} = 1\,\text{TeV}$

$p_T > 50\,\text{GeV}$

$\Delta R_{ij} > 0.5$

using tensor integrals,

double precision



Checks Against Independent Code

In addition to internal consistency checks we perfom comparisons against an independent in-house generator for loop amplitudes.

Due to the high performance of both codes the numerical agreement can be easily checked for 10^6 phase space points.
⇒ **Extensive comparisons across the whole phase space**.

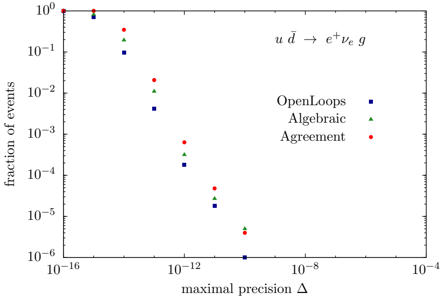
The numerical agreement is checked pointwise and compared to the precision delivered by the individual codes.

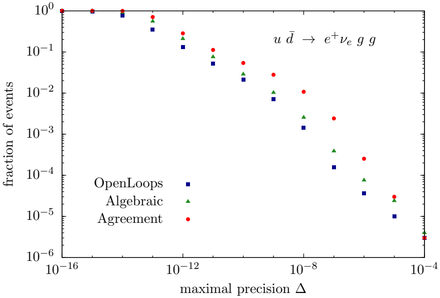
Over 40 processes with 4 or 5 external particles successfully verified.

Numerical Agreement with Independent Code

Stability and agreement of Open Loops and independent code

- Processes $u\bar{d} \rightarrow e^+\bar{\nu}_e\ g$ and $u\bar{d} \rightarrow e^+\bar{\nu}_e\ gg$,
- 10^6 homogeneously distributed phase space points.





- Open Loops appears to be slightly more stable.
- The numerical agreement shows the expected behaviour: within 1 digit of the less stable code.

Automation of NLO Calculations

Combine Open Loops with multi-purpose Monte Carlo programs

- `aMC@NLO`, `POWHEG`, `Sherpa`:
  NLO matching with shower & hadronisation,
  IR subtraction, real emission, phase space integration.
- Open Loops provides an easy to use API to directly access initialisation and matrix element routines.
- Control settings via the user interface of the Monte Carlo program.

Make NLO calculations and analysis as simple as LO calculations.

Open Loops + Sherpa

Interfacing between Sherpa and Open Loops is done

- Use Open Loops to generate and compile process libraries.
- Use standard Sherpa run cards with a few new options.
- Performs some sanity checks and optional consistency checks (e. g. pole cancellations, forced parameter values).
- Validation is in progress.

Libraries for a wide range of processes will be public in the near future.

$e^+\bar{\nu}_e$ and e^+e^- + jets with OpenLoops + Sherpa

Preliminary results for $e^+\bar{\nu}_e + jets$ and $e^+e^- + jets$ total cross sections

Process	σ [pb]	σ_B/σ	σ_R/σ	σ_V/σ	t [min]	t_B/t	t_R/t	t_V/t
$e^+\bar{\nu}_e$	6082 ± 0.8%	92%	5%	4%	0.1	36%	27%	36%
$e^+\bar{\nu}_e j$	1057 ± 1.0%	98%	-5.3%	7%	4.6	9%	84%	7%
$e^+\bar{\nu}_e jj$	294 ± 1.0%	115%	-44%	29%	640	1%	80%	19%

Process	σ [pb]	σ_B/σ	σ_R/σ	σ_V/σ	t [min]	t_B/t	t_R/t	t_V/t
e^+e^-	1190 ± 0.7%	95%	5%	0.3%	0.1	31%	31%	38%
$e^+e^- j$	202 ± 1.1%	100%	-7%	7%	23	5%	85%	10%
$e^+e^- jj$	54.1 ± 1.1%	116%	-46%	30%	2908	1%	66%	33%

(B = Born + I-operator; R = real corrections − dipoles; V = virtual corrections; $\alpha_\mathrm{Nagy} = 1$)

- Nice agreement with `MadLoop` (thanks to R. Frederix for updated results). [Hirschi, Frederix, Frixione, Garzelli, Maltoni, Pittau '11]
- Less than 11 hours for $pp \to Wjj$ (1%) on a single i5-750 CPU core.
- Real corrections (from Sherpa with Amegic) dominates the runtimes. Significant improvements are expected here in the future.

Summary

Open loops is a new algorithm for one-loop amplitudes

- It uses a diagrammatic, tree-like recursion for loop momentum polynomials (instead of fixed loop momenta)
- Process definition → 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 → 4
- Numerically stable when using tensor integrals

OpenLoops+Sherpa: NLO phenomenology for LHC

- Fully automated

NLO predictions become as easy as LO predictions