Automation of NNLO Amplitude Construction in OpenLoops

Natalie Schär
June 1, 2021

in collaboration with:
S. Pozzorini and M. F. Zoller
Monte Carlo Simulation contains:

- Hard Scattering Amplitudes → OpenLoops
- PDFs, Parton Shower, Hadronisation, Underlying Events

OpenLoops constructs Amplitudes from Feynman Diagrams.

\[ \sigma_{\text{part}} \sim \left( \begin{array}{c}
\includegraphics{feynman_diagram1.png}
\end{array} \right) \sim \alpha_s^2 \quad \text{LO} \]

\[ + \left( \begin{array}{c}
\includegraphics{feynman_diagram2.png}
\end{array} \right) + \left( \begin{array}{c}
\includegraphics{feynman_diagram3.png}
\end{array} \right) \sim \alpha_s^3 \quad \text{NLO} \]

\[ + \left( \begin{array}{c}
\includegraphics{feynman_diagram4.png}
\end{array} \right) + \left( \begin{array}{c}
\includegraphics{feynman_diagram5.png}
\end{array} \right) + \left( \begin{array}{c}
\includegraphics{feynman_diagram6.png}
\end{array} \right) + \left( \begin{array}{c}
\includegraphics{feynman_diagram7.png}
\end{array} \right) \sim \alpha_s^4 \quad \text{NNLO} \]

NNLO required for LHC.
Some components to NNLO are available in the public version of OpenLoops:

- Tree components are available.
- Virtual components are available.
- Real components are available.
- Double virtual components are new.
- Double real components are available.
- Loop squared components are available.

Double virtual required for NNLO.
Components to NNLO

Distinguish three types of double virtual diagrams.

- **Tree**
- **Virtual**
- **Real**
- **Real Virtual**
- **Double Virtual**
- **Double Real**
- **Loop Squared**

The diagrams represent different components of the NNLO (Next-to-Next-to-Leading Order) calculations in particle physics, specifically in the context of quantum field theory.
Components to NNLO

For one diagram $\Gamma$:
renormalized Amplitude (D dimensions) = $\mathcal{M}_{2,\Gamma}(4d$ numerator, D-dim denominator)
+ rational and UV counterterms (in lower loop diagrams)

$$\mathcal{M}_{2,\Gamma} = C_{2,\Gamma} \sum_{color} \sum_{r_1=0}^{R_1} \sum_{r_2=0}^{R_2} N_{\mu_1 \cdots \mu_{r_1} \nu_1 \cdots \nu_{r_2}} \int d\vec{q}_1 \int d\vec{q}_2 \frac{q_1^{\mu_1} \cdots q_1^{\mu_{r_1}} q_2^{\nu_1} \cdots q_2^{\nu_{r_2}}}{D(1)(\vec{q}_1) D(2)(\vec{q}_2) D(3)(\vec{q}_3)} |q_3 \to -(q_1+q_2)|$$

In OpenLoops:
- tensor coefficients constructed numerically → in 4 dimensions
- restore coefficients to D dimensions by rational counterterms
- denominators kept analytical
Components to NNLO Calculation

\[ M_{2, \Gamma} = c_{2, \Gamma} \sum_{r_1=0}^{R_1} \sum_{r_2=0}^{R_2} N_{\mu_1 \ldots \mu_{r_1} \nu_1 \ldots \nu_{r_2}} \int d\bar{q}_1 \int d\bar{q}_2 \frac{q_1^{\mu_1} \ldots q_1^{\mu_{r_1}} q_2^{\nu_1} \ldots q_2^{\nu_{r_2}}}{D^{(1)}(\bar{q}_1) D^{(2)}(\bar{q}_2) D^{(3)}(\bar{q}_3)} \bigg| q_3 \to -(q_1 + q_2) \]

tensor integral

tensor coefficient

- In this talk: numerical construction of \( N \) from universal Feynman rules (dressing) in 4d, 2-loop irreducible (ID1, ID2), reducible (RED) diagrams

- further tasks:
  - Renormalization, Rational Terms
  - Reduction (reduction to scalar master integrals, scalar integral evaluation/library)
  - Treatment of IR divergences.
Outline

Tree Algorithm

One Loop Algorithm

Two Loop Algorithm
  Timings and Accuracy

Conclusion
Tree Algorithm in OpenLoops

\[
\begin{align*}
&\text{tree} \\
&+ \text{virtual} \\
&+ \text{real} \\
&+ \text{real virtual} \\
&+ \text{double virtual} \\
&+ \text{double real} \\
&+ \text{loop squared}
\end{align*}
\]
start with external wavefunctions $ex_1, ex_2, ex_3, ex_4, ex_5$
Combine $e_4$, $e_5$ into subtree $w_1$:

$$w_1 = \text{vert}_{QV_A}(e_4, e_5)$$

($Q=$fermion, $A=$anti-fermion, $V=$boson)
Add propagator to $wf_1$:

$wf_1 = vert_{QV_A}(ex_4, ex_5)$

$wf_2 = prop_{Q_A}(wf_1)$

(Q=fermion, A=anti-fermion, V=boson)
Add next external leg:

\[
\begin{align*}
wf_1 &= \text{vert}_\text{QV}_\text{A}(\text{ex}4, \text{ex}5) \\
wf_2 &= \text{prop}_\text{Q}_\text{A}(wf_1) \\
wf_3 &= \text{vert}_\text{QA}_\text{V}(wf_2, \text{ex}3)
\end{align*}
\]

(Q=fermion, A=anti-fermion, V=boson)
same on the other side:

\( \text{wf1=vert}_{-}QV_{-}A(\text{ex4,ex5}) \)
\( \text{wf2=prop}_{-}Q_{-}A(\text{wf1}) \)
\( \text{wf3=vert}_{-}QA_{-}V(\text{wf2,ex3}) \)
\( \text{wf4=vert}_{-}VV_{-}V(\text{ex1,ex2}) \)

(Q=fermion, A=anti-fermion, V=boson)
contract into full diagram, multiply denominator:

$\text{wf1} = \text{vert}_{QV_A}(\text{ex4}, \text{ex5})$

$\text{wf2} = \text{prop}_{Q_A}(\text{wf1})$

$\text{wf3} = \text{vert}_{QA_V}(\text{wf2}, \text{ex3})$

$\text{wf4} = \text{vert}_{VV_V}(\text{ex1}, \text{ex2})$

$A_1 = \text{cont}_{VV}(\text{wf3}, \text{wf4}) \times \text{den}$

(Q=fermion, A=anti-fermion, V=boson)
Tree Level Algorithm: Generalized

Recursively construct subtrees (=vertex+propagator):

\[ w_{a}^{\sigma_{a}}(k_{a}, h_{a}) = \delta_{\sigma_{a} \sigma_{b}} \sigma_{b} \tilde{w}_{b}^{\sigma_{b}}(k_{b}, h_{b}) \]

Then contract into full diagram:

\[ M_{0,\Gamma}(h) = C_{0,\Gamma} \cdot w_{a}^{\sigma_{a}}(k_{a}, h_{a}) \delta_{\sigma_{a} \sigma_{b}} \tilde{w}_{b}^{\sigma_{b}}(k_{b}, h_{b}) \]

- diagrams constructed using universal feynman rules
- subtrees appearing in multiple diagrams are recycled
One Loop Algorithm in OpenLoops

\[
\begin{align*}
\text{tree} & \quad \left( \begin{array}{c}
\text{virtual} \\
\text{real}
\end{array} \right) \\
+ & \quad \left( \begin{array}{c}
\text{real virtual} \\
\text{double virtual}
\end{array} \right) \\
+ & \quad \left( \begin{array}{c}
\text{double real} \\
\text{loop squared}
\end{array} \right)
\end{align*}
\]
external subtrees constructed in tree level algorithm (in combination with tree level diagrams):

$w_2, w_3 \rightarrow w_6$
Open Loop:
Diagram factorizes into chain of segments: $\mathcal{N} = S_1 \cdots S_N$
Dress first segment 
(=vertex+propagator+subtree) $S_1$
attaching the external wavefunction $w_1$.

$\mathcal{N}_0 = 1$
$\mathcal{N}_1 = \mathcal{N}_0 \cdot S_1(w_1)$
Dress second segment attaching the subtree $w_6$.

\[ \mathcal{N}_0 = 1 \]
\[ \mathcal{N}_1 = \mathcal{N}_0 \cdot S_1(w_1) \]
\[ \mathcal{N}_2 = \mathcal{N}_1 \cdot S_2(w_6) \]
Dress third segment.

\[
\begin{align*}
\mathcal{N}_0 &= 1 \\
\mathcal{N}_1 &= \mathcal{N}_0 \cdot S_1(w_1) \\
\mathcal{N}_2 &= \mathcal{N}_1 \cdot S_2(w_6) \\
\mathcal{N}_3 &= \mathcal{N}_2 \cdot S_3(w_4)
\end{align*}
\]
One Loop Algorithm: Example

Dress last segment.

\[ \mathcal{N}_0 = 1 \]
\[ \mathcal{N}_1 = \mathcal{N}_0 \cdot S_1(w_1) \]
\[ \mathcal{N}_2 = \mathcal{N}_1 \cdot S_2(w_6) \]
\[ \mathcal{N}_3 = \mathcal{N}_2 \cdot S_3(w_4) \]
\[ \mathcal{N}_4 = \mathcal{N}_3 \cdot S_4(w_5) \]
One Loop Algorithm: Example

Close the loop (contract open Lorentz/spinor indices).

\[ \mathcal{N}_0 = 1 \]
\[ \mathcal{N}_1 = \mathcal{N}_0 \cdot S_1(w_1) \]
\[ \mathcal{N}_2 = \mathcal{N}_1 \cdot S_2(w_6) \]
\[ \mathcal{N}_3 = \mathcal{N}_2 \cdot S_3(w_4) \]
\[ \mathcal{N}_4 = \mathcal{N}_3 \cdot S_4(w_5) = \mathcal{N}_4^{\beta N} \]

\[ \mathcal{N} = Tr(\mathcal{N}_4^{\beta N}) \]
Segments (vertex + propagator + subtree(s)) can always be written as:

\[
\left[ S_i(q_1, h_i) \right]^{\beta_i}_{\beta_{i-1}} = \frac{\beta_i}{\beta_{i-1}} = \left\{ \left[ Y_i^{\sigma_i} \right]^{\beta_i}_{\beta_{i-1}} + \left[ Z_j^{\nu_i; \sigma_i} \right]^{\beta_i}_{\beta_{i-1}} \right\} w_i^{\sigma_i(k_i, h_i)}
\]

Partially constructed chain (open loop):

\[
\mathcal{N}_n(q_1, \hat{h}_k^{(1)}) = \prod_{i=1}^{k} S_i(q_1, h_i) = w_1 \cdot w_2 \cdot \ldots \cdot w_k \cdot w_{k+1} \cdot w_{k+1} \cdot \ldots \cdot w_N
\]

Recursion step: \( \mathcal{N}_n = \mathcal{N}_{n-1} \cdot S_n \)

- Diagrams factorize into segments
- Universal Feynman Rules (encoded in Y, Z)
Helicities and Rank

- Final result: scattering probability density $\sim \sum_h |M|^2$
- Born-Loop interference required (for virtual, real virtual etc.)
- Multiplication with Born and color factor in the beginning of construction possible $\rightarrow$ start with maximal helicities of any diagrams
- $U_0(h) = 2 \sum_{col} M_0^* C$

Helicities may be summed after each dressing step (exploiting factorization):

$$\sum_h U_0 \text{Tr}(N(h)) = \sum_h \left[ \sum_{h_2} \cdots \sum_{h_N} \left[ \sum_{h_1} U_0(h) S_1(h_1) \right] S_2(h_2) \cdots \right] S_N(h_N)$$

- (in renormalizable theories) each segment:
  - increases rank by 1 (or 0)
  - decreases total helicities by a factor of # helicities of wavefunction in the segment
- minimal helicities with maximal rank $\rightarrow$ efficient, complexity is kept low in final recursion steps
Helicities and Rank: Example

- Helicities = 32, Rank = 0

- Increases rank by 1
- Decreases total helicities by a factor of the number of helicities of the wavefunction in the segment
Helicities and Rank: Example

each segment:
- increases rank by 1
- decreases total helicities by a factor of \( \# \) helicities of wavefunction in the segment

helicities=16,
rank=1

\[ w_5 \]
\[ w_6 \]
\[ w_4 \]
\[ w_1 \]
Helicities and Rank: Example

each segment:
- increases rank by 1
- decreases total helicities by a factor of \( \# \) helicities of wavefunction in the segment

helicities=4, rank=2
Helicities and Rank: Example

each segment:
- increases rank by 1
- decreases total helicities by a factor of \# helicities of wavefunction in the segment

helicities=2, rank=3
Helicities and Rank: Example

Each segment:
- increases rank by 1
- decreases total helicities by a factor of \# helicities of wavefunction in the segment

Helicities = 1, rank = 4
Example:

- After one dressing step subsequent dressing steps are identical.
- Topology (scalar propagators) is identical for both diagrams.
- Diagrams can be merged.

For diagrams A, B with identical segments after n dressing steps (exploit factorization):

\[
U_{A,B} = U_0 \text{Tr}(N_{A,B}) = \text{numerator} \times \text{Born} \times \text{color}
\]

\[
U_A + U_B = (U_{n,A} \cdot S_{n+1} \cdots S_N) + (U_{n,B} \cdot S_{n+1} \cdots S_N)
\]

\[
= (U_{n,A} + U_{n,B}) \cdot S_{n+1} \cdots S_N
\]

Only perform dressing steps n+1 to N once.

Highly efficient way of dressing a large number of diagrams for complicated processes.
Two Loop Algorithm in OpenLoops

\[
\begin{align*}
\text{tree} & \quad + \\
\text{virtual} & \quad + \\
\text{real virtual} & \quad + \\
\text{double virtual} & \quad + \\
\text{double real} & \quad + \\
\text{loop squared} &
\end{align*}
\]
Two Loop Algorithm: Components

- chain 1 = longest chain
- chain 2 = middle chain
- chain 3 = shortest chain
- $V_0, V_1$ = vertices connecting chains
- $q_1, q_2, q_3$ = loop momenta $q_3 = -q_1 - q_2$

Diagram factorizes into 3 chains and 2 vertices (matrix multiplications, indices suppressed):

$$\mathcal{N}(q_1, q_2) = \left[ \mathcal{N}^{(2)}(q_2) \right] \left[ \mathcal{N}^{(3)}(q_3) \right] \left[ V_0(q_1, q_2) \right] \left[ V_1(q_1, q_2) \right] \bigg|_{q_3 \rightarrow -(q_1 + q_2)}$$

Each chain in factorizes into segments

$$\mathcal{N}^{(i)}(q_i) = S_0^{(i)}(q_i) S_1^{(i)}(q_i) \cdots S_{N_i-1}^{(i)}(q_i)$$

Factorization results in freedom of choice for dressing algorithm.
Two Loop Algorithm: Naive Approach

1. dress chains $\mathcal{N}_1(q_1), \mathcal{N}_2(q_2), \mathcal{N}_3(q_3)$

\[
\begin{array}{c}
\left[ \mathcal{N}_1^{(1)}(q_1) \right]^{\beta_{N_1}^{(1)}}_{\beta_{0}^{(1)}} \left[ \mathcal{N}_2^{(2)}(q_2) \right]^{\beta_{N_2}^{(2)}}_{\beta_{0}^{(2)}} \left[ \mathcal{N}_3^{(3)}(q_3) \right]^{\beta_{N_3}^{(3)}}_{\beta_{0}^{(3)}}
\end{array}
\]
Two Loop Algorithm: Naive Approach

1. dress chains $\mathcal{N}^{(1)}(q_1), \mathcal{N}^{(2)}(q_2), \mathcal{N}^{(3)}(q_3)$

2. combine with vertex $V_1$, closing indices $\beta^{(1)}_{N_1}, \beta^{(2)}_{N_2}, \beta^{(3)}_{N_3}$

$$
\left[ \mathcal{N}^{(1)}(q_1) \right]_{\beta^{(1)}_0}^{\beta^{(1)}_{N_1}} \left[ \mathcal{N}^{(2)}(q_2) \right]_{\beta^{(2)}_0}^{\beta^{(2)}_{N_2}} \left[ \mathcal{N}^{(3)}(q_3) \right]_{\beta^{(3)}_0}^{\beta^{(3)}_{N_3}} \left[ V_1(q_1, q_2) \right]_{\beta^{(1)}_{N_1}}^{\beta^{(2)}_{N_2}}^{\beta^{(3)}_{N_3}}
$$
Two Loop Algorithm: Naive Approach

1. dress chains $\mathcal{N}^{(1)}(q_1)$, $\mathcal{N}^{(2)}(q_2)$, $\mathcal{N}^{(3)}(q_3)$

2. combine with vertex $\nu_1$, closing indices $\beta^{(1)}_{N_1}, \beta^{(2)}_{N_2}, \beta^{(3)}_{N_3}$

3. combine with vertex $\nu_0$, closing indices $\beta^{(1)}_0, \beta^{(2)}_0, \beta^{(3)}_0$

\[
\begin{align*}
\left[ \mathcal{N}^{(1)}(q_1) \right]_{\beta^{(1)}_{N_1}} & \left[ \mathcal{N}^{(2)}(q_2) \right]_{\beta^{(2)}_{N_2}} & \left[ \mathcal{N}^{(3)}(q_3) \right]_{\beta^{(3)}_{N_3}} & \left[ \nu_1(q_1,q_2) \right]_{\beta^{(1)}_{N_1}} & \left[ \nu_0(q_1,q_2) \right]_{\beta^{(2)}_{N_2}} \beta^{(3)}_{N_3} \\
\end{align*}
\]
Two Loop Algorithm: Naive Approach

1. dress chains $\mathcal{N}^{(1)}(q_1)$, $\mathcal{N}^{(2)}(q_2)$, $\mathcal{N}^{(3)}(q_3)$

2. combine with vertex $\nu_1$, closing indices $\beta_{N_1}^{(1)}, \beta_{N_2}^{(2)}, \beta_{N_3}^{(3)}$

3. combine with vertex $\nu_0$, closing indices $\beta_0^{(1)}, \beta_0^{(2)}, \beta_0^{(3)}$

4. map momenta, loop over helicities

\[
\begin{align*}
\left[ \mathcal{N}^{(1)}(q_1) \right]^{\beta_{N_1}^{(1)}}_{\beta_0^{(1)}} \left[ \mathcal{N}^{(2)}(q_2) \right]^{\beta_{N_2}^{(2)}}_{\beta_0^{(2)}} \left[ \mathcal{N}^{(3)}(q_3) \right]^{\beta_{N_3}^{(3)}}_{\beta_0^{(3)}} \left[ \nu_1(q_1, q_2) \right]^{\beta_{N_1}^{(1)} \beta_{N_2}^{(2)} \beta_{N_3}^{(3)}}_{\beta_0^{(1)} \beta_0^{(2)} \beta_0^{(3)}} \left[ \nu_0(q_1, q_2) \right]^{\beta_{N_1}^{(1)} \beta_{N_2}^{(2)} \beta_{N_3}^{(3)}}_{\beta_0^{(1)} \beta_0^{(2)} \beta_0^{(3)}} & \mid q_3 \rightarrow -(q_1 + q_2) 
\end{align*}
\]
Two Loop Algorithm: Observations and Challenges

\[
\left[ \mathcal{N}^{(1)}(q_1) \right]_{\beta_0^{(1)}} \beta_{N_1}^{(1)} \left[ \mathcal{N}^{(2)}(q_2) \right]_{\beta_0^{(2)}} \beta_{N_2}^{(2)} \left[ \mathcal{N}^{(3)}(q_3) \right]_{\beta_0^{(3)}} \beta_{N_3}^{(3)} \left[ \mathcal{V}_0(q_1, q_2) \right]_{\beta_0^{(1)} \beta_0^{(2)} \beta_0^{(3)}} \left[ \mathcal{V}_1(q_1, q_2) \right]_{\beta_{N_1}^{(1)} \beta_{N_2}^{(2)} \beta_{N_3}^{(3)}} \mid q_3 \rightarrow -(q_1 + q_2)
\]

1. dress chains \( \mathcal{N}^{(1)}(q_1), \mathcal{N}^{(2)}(q_2), \mathcal{N}^{(3)}(q_3) \)
2. combine with vertex \( \mathcal{V}_1 \), closing indices \( \beta_{N_1}^{(1)} \beta_{N_2}^{(2)} \beta_{N_3}^{(3)} \)
3. combine with vertex \( \mathcal{V}_0 \), closing indices \( \beta_0^{(1)}, \beta_0^{(2)}, \beta_0^{(3)} \)
4. map momenta, loop over helicities

**Observations:**

- step 2. is performed for 6 open spinor/Lorentz indices
- step 3. is performed for 3 open spinor/Lorentz indices
- in step 2,3 we have maximal ranks, as all chains have been fully dressed
- the mapping in step 4 is performed for maximal ranks
- all dressing steps are performed for all helicities

**This is very inefficient.**
Cost Simulation for Two Loop Algorithm

- factorization: freedom of order in combining chains and vertices
- full algorithm: $N$ recursion steps with partially dressed numerators
  $$\mathcal{N}_n = \mathcal{N}_{n-1} X_n,$$
  with building blocks $X_n \in \{ S_k^{(i)}, V_j, \mathcal{N}^{(i)}, M_0^* C \}$
- CPU cost $\sim \#$ multiplications
- → cost simulation tracking $\#$ components and multiplications
- test different variants to determine most efficient algorithm for two loop diagrams
0. Sort chains by length: $N_1 \geq N_2 \geq N_3$, choose order of $\nu_0, \nu_1$ by vertex type
0. Sort chains by length: \( N_1 \geq N_2 \geq N_3 \), choose order of \( \nu_0, \nu_1 \) by vertex type

1a. Initial Condition for chain 1 (longest chain): Born \( \times \) color factor. Start with maximal \( \# \) helicities.

\[
\mathcal{U}_0^{(1)} = 2 \sum_{col} C\mathcal{M}_0^* 
\]
0. Sort chains by length: $N_1 \geq N_2 \geq N_3$, choose order of $\nu_0, \nu_1$ by vertex type
1a. Initial Condition for chain 1 (longest chain): Born $\times$ color factor. Start with maximal $\#$ helicities.
1b. Dress $(\mathcal{N}^{(1)}(q_1) \times \text{Born} \times \text{color})$ summing helicities at each vertex (as at one loop).

\[ u_n^{(1)} = u_{n-1}^{(1)} s_n^{(1)}, \quad u_0^{(1)} = 2 \sum_{\text{col}} C M_0^* \]
0. Sort chains by length: $N_1 \geq N_2 \geq N_3$, choose order of $\nu_0$, $\nu_1$ by vertex type

1a. Initial Condition for chain 1 (longest chain): Born $\times$ color factor. Start with maximal # helicities.

1b. Dress $(\mathcal{N}^{(1)}(q_1) \times $ Born $\times$ color) summing helicities at each vertex (as at one loop).

2. Dress $\mathcal{N}^{(3)}(q_3)$, start with no helicities, new helicities enter at each vertex.

$$\mathcal{N}_n^{(3)}(q_3) = \mathcal{N}_{n-1}^{(3)} S_n^{(3)}, \quad \mathcal{N}_0^{(3)} = 1,$$
Two Loop Algorithm in OpenLoops

0. Sort chains by length: \( N_1 \geq N_2 \geq N_3 \), choose order of \( \nu_0, \nu_1 \) by vertex type

1a. Initial Condition for chain 1 (longest chain): Born \( \times \) color factor. Start with maximal \# helicities.

1b. Dress \( (\mathcal{N}^{(1)}(q_1) \times \text{Born} \times \text{color}) \) summing helicities at each vertex (as at one loop).

2. Dress \( \mathcal{N}^{(3)}(q_3) \), start with no helicities, new helicities enter at each vertex.

3. Attach \( \mathcal{N}^{(1)}(q_1), \mathcal{N}^{(3)}(q_3) \) to \( \nu_0 \) and \( \nu_1 \), map \( q_3 \rightarrow -q_1 - q_2 \), sum helcs of \( \mathcal{N}^{(3)}(q_3), \nu_1, \nu_0 \).

\[
[U^{(13)}]^{\beta_2(2)}_{\beta_0(2)} = [U^{(1)}]^{\beta_1(1)}_{\beta_0(1)} [\mathcal{N}^{(3)}]^{\beta_3(3)}_{\beta_0(3)} \left[ \nu_0(q_1, q_3) \right]^{\beta_0(1)} \left[ \nu_1(q_1, q_3) \right]^{\beta_1(1)} \left[ \nu_1(q_1, q_3) \right]^{\beta_3(3)} \left| q_3 \rightarrow -(q_1 + q_2) \right.
\]
Two Loop Algorithm in OpenLoops

0. Sort chains by length: \( N_1 \geq N_2 \geq N_3 \), choose order of \( \nu_0, \nu_1 \) by vertex type

1a. Initial Condition for chain 1 (longest chain): Born \( \times \) color factor. Start with maximal \# helicities.

1b. Dress \( (\mathcal{N}^{(1)}(q_1) \times \text{Born} \times \text{color}) \) summing helicities at each vertex (as at one loop).

2. Dress \( \mathcal{N}^{(3)}(q_3) \), start with no helicities, new helicities enter at each vertex.

3. Attach \( \mathcal{N}^{(1)}(q_1), \mathcal{N}^{(3)}(q_3) \) to \( \nu_0 \) and \( \nu_1 \), map \( q_3 \rightarrow -q_1-q_2 \), sum hels of \( \mathcal{N}^{(3)}(q_3), \nu_1, \nu_0 \).

4. Attach \( \mathcal{N}^{(2)}(q_2) \) segments to previously constructed object, sum helicities at each vertex.

\[
\mathcal{U}_n^{(123)} = \mathcal{U}_{(n-1)}^{(123)} S_n^{(2)}, \quad \mathcal{U}_0^{(123)} = \mathcal{U}^{(13)} = \mathcal{U}^{(1)}(q_1) \mathcal{N}^{(3)}(q_3) \nu_0(q_1,q_2) \nu_1(q_1,q_2)
\]
Two Loop Algorithm in OpenLoops

0. Sort chains by length: $N_1 \geq N_2 \geq N_3$, choose order of $\nu_0$, $\nu_1$ by vertex type

1a. Initial Condition for chain 1 (longest chain): Born $\times$ color factor. Start with maximal $\neq$ helicities.

1b. Dress $(\mathcal{N}^{(1)}(q_1) \times \text{Born} \times \text{color})$ summing helicities at each vertex (as at one loop).

2. Dress $\mathcal{N}^{(3)}(q_3)$, start with no helicities, new helicities enter at each vertex.

3. Attach $\mathcal{N}^{(1)}(q_1)$, $\mathcal{N}^{(3)}(q_3)$ to $\nu_0$ and $\nu_1$, map $q_3 \rightarrow -q_1 - q_2$, sum helics of $\mathcal{N}^{(3)}(q_3), \nu_1, \nu_0$.

4. Attach $\mathcal{N}^{(2)}(q_2)$ segments to previously constructed object, sum helicities at each vertex.

This algorithm is two orders of magnitude faster than the naive approach.
Pseudotree Test

Test validity and numerical stability of two loop algorithm without computing tensor integrals.

- Insert pseudo wavefunctions $e_1, e_2, e_3, e_4 \rightarrow$ saturate indices
- set $q_1, q_2$ to random (constant) values, contract tensor coefficients $N_{\mu_1 \ldots \mu_r_1 \nu_1 \ldots \nu_r_2}$ with fixed-value tensor integrand $\frac{q_1^{\mu_1} \ldots q_1^{\mu_r_1} q_2^{\nu_1} \ldots q_2^{\nu_r_2}}{D(q_1, q_2)}$
- $\rightarrow$ compare with well tested tree level algorithm
- establish quad precision as benchmark, perfect (16 digit) agreement at quad precision
Accuracy

Two loop algorithm using pseudotree mode for $10^5$ uniform random phase space points. Numerical stability of double (dp) vs quad (qp) precision scattering probability density $\mathcal{W}_{02} = \sum_{\text{hel}} \sum_{\text{col}} 2 \text{Re}[M^*_0 M_2]$:

Process: $gg \rightarrow \bar{t}t$

Process: $d\bar{d} \rightarrow u\bar{u}g$

Relative Error: $\mathcal{A} = \frac{|\mathcal{W}_{02}^{dp} - \mathcal{W}_{02}^{qp}|}{\text{Min}(|\mathcal{W}_{02}^{dp}|, |\mathcal{W}_{02}^{qp}|)}$

Excellent numerical stability. Essential for full calculation (tensor integral reduction will be main source of instabilities).
Timings for Two Loop Tensor Coefficients

QED, QCD and SM (NNLO QCD) processes (single intel i7-6600U, 2.6 GHz, 16GB RAM, 1000 psp)

- 2 → 2 process: 6-100ms/psp
- 2 → 3 process: 60-2500ms/psp

Runtime $\propto$ # diagrams
time/psp/diagram $\sim 150 \mu s$

Constant ratios between NNLO virtual (2l) and real-virtual (1l+g):

\[
\frac{2l \text{ (tensor coefficients)}}{1l+g \text{ (tensor coefficients)}} \sim 9
\]

\[
\frac{2l \text{ (tensor coefficients)}}{1l+g \text{ (full calculation)}} \sim 4
\]

Strong CPU performance, comparable to real-virtual corrections in OpenLoops.
Conclusion

New algorithm for two loop tensor coefficients:

- Excellent numerical stability
- Highly efficient, comparable to real virtual corrections
  - determined most efficient algorithm through cost simulation
  - exploit factorization of two loop diagrams into chains and vertices for ideal order
  - exploit factorization of chains and on the fly helicity summation for efficient treatment of individual building blocks.
  - merging and recycling of dressing steps.
- Fully implemented for NNLO QED and QCD Corrections to SM (reducible and irreducible)
- Fully generic algorithm

next steps

- UV counterterms and rational counterterms
- tensor integrals (reduction and evaluation)
End
Factorization into Segments

\[ \mathcal{N}(q_1, q_2) = \left[ \mathcal{N}(1)(q_1) \right]_{\beta_0(1)}^{\beta_{N_1}(1)} \left[ \mathcal{N}(2)(q_2) \right]_{\beta_0(2)}^{\beta_{N_2}(2)} \left[ \mathcal{N}(3)(q_3) \right]_{\beta_0(3)}^{\beta_{N_3}(3)} \times \left[ \mathcal{V}_0(q_1, q_2) \right]_{\beta_0(1)}^{\beta_0(1)} \left[ \mathcal{V}_1(q_1, q_2) \right]_{\beta_{N_1}(1)\beta_{N_2}(2)\beta_{N_3}(3)} q_3 \rightarrow -(q_1 + q_2) \]

\[ \mathcal{N}(i)(q_i)_{\beta_0(i)}^{\beta_{N_i}(i)} = S_0(i)(q_i)_{\beta_0(i)}^{\beta_1(i)} S_1(i)(q_i)_{\beta_1(i)}^{\beta_2(i)} \cdots S_{N_i-1}(q_i)_{\beta_0(i)}^{\beta_{N_i-1}(i)} \]
Helicities

There are three ways of treating helicities along the three chains of a two-loop 1PI diagram:

- Global helicity loop (like in OpenLoops 1) → this is sure to be the most inefficient.

- “Down” method (represented by downward arrows): Use on-the-fly helicity summation (like in OpenLoops 2), i.e. the number of active helicities is reduced in each step. Requires interference with Born before. After each step we have a helicity array with the d.o.f. of the undressed segments.

- “Up” method (represented by upward arrow): Helicity arrays are constructed for the d.o.f. of the already dressed segments and extended in each dressing step by the d.o.f. of the attached subtree(s).
Before mapping:
Chain 3 (green) has rank 2, V0V1 have rank 0

$$q_3^2 = (-q_1 - q_2)^2 = q_1^2 - 2q_1q_2 + q_2^2$$

rank in $q_1$ is increased by 2 AND rank in $q_2$ is increased by 0
OR
rank in $q_1$ is increased by 0 AND rank in $q_2$ is increased by 2
OR
rank in $q_1$ is increased by 1 AND rank in $q_2$ is increased by 1

maximum ranks in $q_1$ and $q_2$ are not independent, superfluous ranks can be removed
<table>
<thead>
<tr>
<th>r</th>
<th>component</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>q_0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>q_1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>q_2</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>q_3</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>q_0^2</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>q_1 q_2</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>q_1 q_3</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>q_1 q_4</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>q_2^2</td>
<td>10</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>q_3^2</td>
<td>15</td>
</tr>
</tbody>
</table>