COLLIER
a Complex One-Loop Library in Extended Regularizations

Lars Hofer
IFAE Barcelona

in collaboration with
A. Denner and S. Dittmaier

Los Angeles, June 2015
Motivation

- **no new particles** beyond the SM found so far at the LHC
  - new physics might show up only as **small deviations** from SM predictions
  - entering **precision era of LHC**

- perform **precise measurements** of particle couplings (e.g. couplings of the Higgs boson)

- comparison with precise SM predictions
  - need SM predictions at **NNLO QCD** and at **NLO electroweak**

- **one-loop amplitudes** needed for **NLO virtual** and **NNLO real-virtual** contributions
One-loop amplitudes

general structure of one-loop amplitudes:

\[
\frac{N(q)}{D_0 \cdots D_{N-1}} = \sum_r c_{\mu_1 \cdots \mu_r} \int d^D q \frac{q^{\mu_1} \cdots q^{\mu_r}}{D_0 \cdots D_{N-1}}
\]

with \( D_i = (q + p_i)^2 - m_i^2 \)
One-loop amplitudes

general structure of one-loop amplitudes:

\[
\int d^D q \frac{N(q)}{D_0 \cdots D_{N-1}} = \sum_{r} c_{\mu_1 \cdots \mu_r} \int d^D q \frac{q^{\mu_1} \cdots q^{\mu_r}}{D_0 \cdots D_{N-1}}
\]

with \( D_i = (q + p_i)^2 - m_i^2 \)

can be decomposed in terms of scalar integrals:

\[
\sum_{l} d_l D_0(l) + \sum_{k} c_k C_0(k) + \sum_{j} b_j B_0(j) + \sum_{i} a_i A_0(i) + R
\]
One-loop amplitudes

general structure of one-loop amplitudes:

\[
\mathcal{A} = \int d^D q \frac{N(q)}{D_0 \cdots D_{N-1}} = \sum_{r} c_{\mu_1 \cdots \mu_r} \int d^D q \frac{q^{\mu_1} \cdots q^{\mu_r}}{D_0 \cdots D_{N-1}}
\]

with \( D_i = (q + p_i)^2 - m_i^2 \)

can be decomposed in terms of scalar integrals:

\[
\mathcal{A} = \sum_{l} d_l D_0(l) + \sum_{k} c_k C_0(k) + \sum_{j} b_j B_0(j) + \sum_{i} a_i A_0(i) + R
\]

different approaches for calculation:

- conventional method (Feynman diagrams) \( \rightarrow \) TI’s needed
- generalised unitarity [Ossola, Papadopoulos, Pittau ’07, Bern, Dixon, Kosower, Britto, Cachazo, Feng, Ellis, Giele, Melnikov, . . .]
- recursive methods using tensor integrals \( \rightarrow \) TI’s needed [van Hameren’09; Cascioli, Maierhöfer, Pozzorini’11; Actis, Denner, LH, Scharf, Uccirati’12]
Many tools for NLO calculations, e.g.
FeynCalc/FormCalc, Blackhat, NGLuon, aMC@NLO, HELAC-NLO, GoSam, CutTools, HELAC-1LOOP, Samurai, Madloop, OpenLoops, Recola, ...

Libraries for scalar and tensor integrals, e.g.
FF [van Oldenborgh], LoopTools [Hahn, Perez-Victoria], QCDLoop [R.K. Ellis, Zanderighi], OneLOop [van Hameren], Golem95C [Cullen, Guillet, Heinrich, Kleinschmidt, Pilon, ...], PJFry [Fleischer, Riemann]

This talk:

COLLIER = Complex one loop library in extended regularizations
fortran-library for fast and stable numerical evaluation of tensor integrals [Denner, Dittmaier, LH → publication in preparation]
Collier: Applications

- successfully used in many calculations of
  - **NLO QCD corrections**, e.g.
    - $pp \rightarrow t\bar{t}j$ [Dittmaier, Uwer, Weinzierl '07]
    - $pp \rightarrow t\bar{t}b\bar{b}$ [Bredenstein, Denner, Dittmaier, Pozzorini '09]
    - $pp \rightarrow WWb\bar{b}$ [Denner, Dittmaier, Kallweit, Pozzorini '11]
    - $pp \rightarrow WWb\bar{b}H$ [Denner, Feger in prep.] (talk by A.Denner)
  - **NLO EW corrections**, e.g.
    - $e^+e^- \rightarrow 4 \text{ fermions}$ [Denner, Dittmaier, Roth, Wieders '05]
    - $pp \rightarrow Hjj \text{ via VBF}$ [Ciccolini, Denner, Dittmaier '07]
    - $pp \rightarrow H^+\text{ dilepton}$ [Denner, Dittmaier, Kallweit, Mück '11]
    - $pp \rightarrow l^+l^-jj$ [Denner, LH, Scharf, Uccirati '14] (talk by S.Uccirati)
    - $pp \rightarrow \mu^+\mu^-e^+e^-$ [Biedermann et al. in prep.] (talk by B.Biedermann)

- integrated in **automated NLO generators**
  - **OpenLoops** [Cascioli, Maierhöfer, Pozzorini] (talks by J.Lindert and P.Maierhöfer)
  - **Recola** [Actis, Denner, LH, Scharf, Uccirati] (talk by S.Uccirati)
Reduction of tensor integrals

Methods implemented in Collier: applied method depends on number $N$ of propagators

- $N = 1, 2$: explicit analytical expressions
- $N = 3, 4$: exploit Lorentz-covariance standard PV reduction [Passarino, Veltman '79] + stable expansions in exceptional phase-space regions [Denner, Dittmaier '05]
- $N \geq 5$: exploit 4-dimensionality of space-time [Melrose '65; Denner, Dittmaier '02, '05; Binoth et al. '05]

Basic scalar integrals from analytic expressions
[’t Hooft, Veltman’79; Beenaker, Denner’90; Denner, Nierste, Scharf’91; Ellis, Zanderighi’08; Denner, Dittmaier’11]

⇒ fast and stable numerical reduction algorithm
**$N = 3, 4$: PV reduction**

- $T^\mu_1...^\mu_r = \int d^Dq \frac{q^\mu_1...q^\mu_r}{D_0...D_{N-1}}$, \hspace{1cm} D_i = (q + p_i)^2 - m_i^2$

contractions:

- $p_i^\mu q_\mu = -f_i + D_i - D_0$, \hspace{1cm} $g^{\mu\nu} q_\mu q_\nu = m_0^2 + D_0$

→ reduction to **lower-rank** and **lower-point** integrals
$N = 3, 4$: PV reduction

- $T^{\mu_1 \cdots \mu_r} = \int d^D q \frac{q^{\mu_1} \cdots q^{\mu_r}}{D_0 \cdots D_{N-1}}$,  \quad D_i = (q + p_i)^2 - m_i^2$

contractions:
$p_i^\mu q_\mu = -f_i + D_i - D_0$,  \quad $g^{\mu \nu} q_\mu q_\nu = m_0^2 + D_0$

$\rightarrow$ reduction to lower-rank and lower-point integrals

- covariant decomposition of tensors:

\[(T^N)_{\mu_1 \cdots \mu_P} = \sum_k \sum_{i_1, \ldots, i_k} T_{0 \cdots 0}^{N, P} \{ g \cdots g \ p_{i_1} \cdots p_{i_k} \}^{\mu_1 \cdots \mu_P}_{(P-k)/2} \]
\( N = 3, 4: \text{PV reduction} \)

\[ T^{\mu_1 \ldots \mu_r} = \int d^D q \frac{g_{\mu_1} \ldots g_{\mu_r}}{D_0 \ldots D_{N-1}}, \quad D_i = (q + p_i)^2 - m_i^2 \]

contractions:
\[ p_i^\mu q_\mu = -f_i + D_i - D_0, \quad g^{\mu\nu} q_\mu q_\nu = m_0^2 + D_0 \]

\( \rightarrow \) reduction to lower-rank and lower-point integrals

\( \text{covariant decomposition of tensors:} \)
\[
(T^N)_{\mu_1 \ldots \mu_P} = \sum_k \sum_{i_1, \ldots, i_k} T^{N,P}_{0 \ldots 0 i_1 \ldots i_k} \left\{ g \cdots g \right\} p_{i_1} \cdots p_{i_k} \}
\]

\( \text{system of linear equations for coefficients:} \)
\( \rightarrow \) invert for \( T^{N,P} \)'s \( \Rightarrow \) recursive numerical calculation

\[ \Delta T^{N,P} = \left[ T^{N,P-1}, T^{N,P-2}, T^{N-1} \right] \]

Gram determinant: \( \Delta = \det(Z) \) with \( Z_{ij} = 2p_ip_j \)
Small Gram determinants

\[(PV) \quad \Delta T^{N,P} = [T^{N,P-1}, T^{N,P-2}, T^{N-1}]\]

small Gram determinant: \(\Delta \rightarrow 0\)

\(T^{N,P-1}, T^{N,P-2}, T^{N-1}\) become linearly dependent
Small Gram determinants

\[
\Delta T^{N,P} = \left[ T^{N,P-1}, T^{N,P-2}, T^{N-1} \right]
\]

small Gram determinant: \( \Delta \to 0 \)

- \( T^{N,P-1}, T^{N,P-2}, T^{N-1} \) become linearly dependent
- \( T^{N,P} \) as sum of \( 1/\Delta \)-singular terms
  - spurious singularities cancel to give \( O(\Delta)/\Delta \)-result
  - numerical determination of \( T^{N,P} \) becomes unstable
Small Gram determinants

(PV) \[ \Delta T^{N,P} = [T^{N,P-1}, T^{N,P-2}, T^{N-1}] \]

small Gram determinant: \( \Delta \to 0 \)

- \( T^{N,P-1}, T^{N,P-2}, T^{N-1} \) become linearly dependent
- \( T^{N,P} \) as sum of \( 1/\Delta \)-singular terms
  - spurious singularities cancel to give \( O(\Delta)/\Delta \)-result
  - numerical determination of \( T^{N,P} \) becomes unstable
- scalar integrals \( D_0, C_0, B_0, A_0 \) become linearly dependent
  \( \Rightarrow O(\Delta)/\Delta \)-instabilities intrinsic to all methods relying on the full set of basis integrals \( D_0, C_0, B_0, A_0 \)
- solution: choose appropriate set of base functions depending on phase-space point
Expansion in Gram determinant

\[ \Delta T^{N,P} = [T^{N,P-1}, T^{N,P-2}, T^{N-1}] \]
Expansion in Gram determinant

\[ \Delta T^{N,P+1} = [T^{N,P}, T^{N,P-1}, T^{N-1}] \]

- exploit linear dependence of \( T^{N,P}, T^{N,P-1}, T^{N-1} \) for \( \Delta = 0 \) to determine \( T^{N,P} \) up to terms of \( \mathcal{O}(\Delta) \)
Expansion in Gram determinant

\[ \Delta T^{N,P+1} = [T^{N,P}, T^{N,P-1}, T^{N-1}] \]

\[ \Delta T^{N,P+2} = [T^{N,P+1}, T^{N,P}, T^{N-1}] \]

- exploit linear dependence of \( T^{N,P}, T^{N,P-1}, T^N \) for \( \Delta = 0 \) to determine \( T^{N,P} \) up to terms of \( \mathcal{O}(\Delta) \)

- calculate \( T^{N,P+1} \) in the same way
Expansion in Gram determinant

\[ \Delta T^{N,P+1} = [T^{N,P}, T^{N,P-1}, T^{N-1}] \]

\[ \Delta T^{N,P+2} = [T^{N,P+1}, T^{N,P}, T^{N-1}] \]

- exploit linear dependence of \( T^{N,P}, T^{N,P-1}, T^{N} \) for \( \Delta = 0 \) to determine \( T^{N,P} \) up to terms of \( \mathcal{O}(\Delta) \)
- calculate \( T^{N,P+1} \) in the same way
- use \( T^{N,P+1} \) to compute \( \mathcal{O}(\Delta) \) in \( T^{N,P} \)
Expansion in Gram determinant

\[ \Delta T^{N,P+1} = [T^{N,P}, T^{N,P-1}, T^{N-1}] \]

\[ \Delta T^{N,P+2} = [T^{N,P+1}, T^{N,P}, T^{N-1}] \]

- exploit linear dependence of \( T^{N,P}, T^{N,P-1}, T^{N} \) for \( \Delta = 0 \) to determine \( T^{N,P} \) up to terms of \( O(\Delta) \)
- calculate \( T^{N,P+1} \) in the same way
- use \( T^{N,P+1} \) to compute \( O(\Delta) \) in \( T^{N,P} \)
- higher orders in \( \Delta \) iteratively:
  \( O(\Delta^k) \) of \( T^{N,P} \) requires lower-point \( T^{N-1} \) up to rank \( P + k \)
- basis of scalar integrals effectively reduced (e.g. \( D_0 \) from \( C_0 \)'s)
Coefficients vs. tensors

\[
(T^N)^{\mu_1 \cdots \mu_P} = \sum_k \sum_{i_1, \ldots, i_k} T_{0 \cdots 0}^{N,P}_{i_1 \cdots i_k} \{ g \cdots g \}_{(P-k)/2}^{p_{i_1} \cdots p_{i_k}} \}_{\mu_1 \cdots \mu_P}
\]

# of tensor coefficients (TC) vs. # of tensor elements (TE)

<table>
<thead>
<tr>
<th></th>
<th>r = 0</th>
<th>r = 1</th>
<th>r = 2</th>
<th>r = 3</th>
<th>r = 4</th>
<th>r = 5</th>
<th>r = 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>N = 3</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>13</td>
<td>22</td>
<td>34</td>
<td>50</td>
</tr>
<tr>
<td>N = 4</td>
<td>1</td>
<td>4</td>
<td>11</td>
<td>24</td>
<td>46</td>
<td>80</td>
<td>130</td>
</tr>
<tr>
<td>N = 5</td>
<td>1</td>
<td>5</td>
<td>16</td>
<td>40</td>
<td>86</td>
<td>166</td>
<td>296</td>
</tr>
<tr>
<td>N = 6</td>
<td>1</td>
<td>6</td>
<td>22</td>
<td>62</td>
<td>148</td>
<td>314</td>
<td>610</td>
</tr>
<tr>
<td>N = 7</td>
<td>1</td>
<td>7</td>
<td>29</td>
<td>91</td>
<td>239</td>
<td>553</td>
<td>1163</td>
</tr>
<tr>
<td>tensor</td>
<td>1</td>
<td>5</td>
<td>15</td>
<td>35</td>
<td>70</td>
<td>126</td>
<td>210</td>
</tr>
</tbody>
</table>

#TC < #TE

#TC > #TE
Coefficients vs. tensors

\[(T^N)_{\mu_1 \cdots \mu_P} = \sum_k \sum_{i_1, \ldots, i_k} T_{0 \cdots 0}^{N, P}_{P-k} \{ g \cdots g \ p_{i_1} \cdots p_{i_k} \}^{\mu_1 \cdots \mu_P}
\]

# of tensor coefficients (TC) vs. # of tensor elements (TE)

<table>
<thead>
<tr>
<th>r</th>
<th>N = 3</th>
<th>N = 4</th>
<th>N = 5</th>
<th>N = 6</th>
<th>N = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TC</td>
<td>TC</td>
<td>TC</td>
<td>TC</td>
<td>TC</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>11</td>
<td>16</td>
<td>22</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>24</td>
<td>40</td>
<td>62</td>
<td>91</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>46</td>
<td>86</td>
<td>148</td>
<td>239</td>
</tr>
<tr>
<td>5</td>
<td>34</td>
<td>80</td>
<td>166</td>
<td>314</td>
<td>553</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>130</td>
<td>296</td>
<td>610</td>
<td>1163</td>
</tr>
</tbody>
</table>

NLO generators OpenLoops and Recola:
parametrisation of one-loop amplitude in terms of tensor integrals:
calculated by OpenLoops/Recola

\[\mathcal{M} = \sum_j C_{\mu_1 \cdots \mu_{n_j}}^{(j)} T_{(j)}^{\mu_1 \cdots \mu_{n_j}} \]

⇒ need full tensors!
From coefficients to tensors

$$(T^N)_{\mu_1\cdots\mu_P} = \sum_k \sum_{i_1,\ldots,i_k} T^{N,P}_{0\cdots0_{i_1\cdots i_k}} \left\{ g \cdots g p_{i_1} \cdots p_{i_k} \right\}_{(P-k)/2}^{\mu_1\cdots\mu_P}$$

In Collier:

- output: coefficients $T^N_{0\cdots0_{i_1\cdots i_k}}$ or tensors $(T^N)_{\mu_1\cdots\mu_P}$
- efficient algorithm to construct tensors from invariant coefficients for arbitrary $N, P$ via recursive calculation of tensor structures
- for $N \geq 6$: Direct reduction at tensor level
Features of Collier

- complete set of one-loop scalar integrals
- implementation of tensor integrals for (in principle) arbitrary number of external momenta $N$ (tested in many physical processes up to $N = 6$)
- various expansion methods implemented for exceptional phase-space points (to arbitrary order in expansion parameter)
- mass- and dimensional regularisation supported for IR-singularities
- complex masses supported (unstable particles)
- cache-system to avoid recalculation of identical integrals
- output: coefficients $T_{0\ldots0i_1\ldots i_k}^N$ or tensors $(T^N_{\mu_1\ldots\mu_P})$
- two independent implementations: COLI+DD
Latest developments

- validation of 7-point functions in $pp \rightarrow WWb\bar{b}H$
  (talk by A.Denner)

- improvement in selection of expansion method
  $\Rightarrow$ increased stability

- error estimates are performed and returned with the results for the integrals

- improved user friendliness:
  - different in- and output formats supported for calls of tensor integrals
  - various options for output and error handling
  - demo programs illustrating the usage of the library
Structure of Collier

Collier

- set/get parameters in Coli and DD
- \( N \)-point coefficients
- \( N \)-point tensors

Coli
- scalar integrals
- 2-point coefficients
- 3-, 4- point reduction
- \( PV + \) expansions
- \( N \geq 5 \)-point reduction

DD
- scalar integrals
- 2-point coefficients
- 3-, 4- point reduction
- \( PV + \) expansions
- \( 5-, 6\)-point reduction

Cache system

- construction of \( N \)-point tensors from coefficients
- direct reduction for \( N \geq 6 \)-point tensors
three different modes to run Collier:

1: Use Coli implementation
2: Use DD implementation
3: Use Coli and DD and compare results

mode=3:
allows to set parameter check precision:
arguments and results of function calls are reported to an output file if agreement between Coli and DD is worse than check precision
Output of Collier

Structure UV- or IR-singular integrals in $D = 4 - 2\epsilon$ dimensions

$$T^N = T^N_{\text{fin}}(\mu^2_{\text{UV}}, \mu^2_{\text{IR}}) + a^\text{UV} \Delta_{\text{UV}} + a^{\text{IR}}_2 \left( \Delta^{(2)}_{\text{IR}} + \Delta^{(1)}_{\text{IR}} \ln \mu^2_{\text{IR}} \right) + a^{\text{IR}}_1 \Delta^{(1)}_{\text{IR}}$$
Output of Collier

Structure UV- or IR-singular integrals in $D = 4 - 2\epsilon$ dimensions

$$T^N = T^N_{\text{fin}}(\mu^2_{\text{UV}}, \mu^2_{\text{IR}}) + a^{\text{UV}} \Delta_{\text{UV}} + a^{\text{IR}}_2 \left( \Delta^{(2)}_{\text{IR}} + \Delta^{(1)}_{\text{IR}} \ln \mu^2_{\text{IR}} \right) + a^{\text{IR}}_1 \Delta^{(1)}_{\text{IR}}$$

- scales
- and poles

$$\Delta_{\text{UV}} = \frac{c(\epsilon_{\text{UV}})}{\epsilon_{\text{UV}}}, \quad \Delta_{\text{IR},1} = \frac{c(\epsilon_{\text{IR}})}{\epsilon_{\text{IR}}}, \quad \Delta_{\text{IR},2} = \frac{c(\epsilon_{\text{IR}})}{\epsilon^2_{\text{IR}}}$$

can be set to arbitrary real values

$\Rightarrow$ output of Collier: numerical value for full $T^N$
Output of Collier

Structure UV- or IR-singular integrals in $D = 4 - 2\epsilon$ dimensions

$$T^N = T^N_{\text{fin}}(\mu^2_{\text{UV}}, \mu^2_{\text{IR}}) + a^{\text{UV}} \Delta_{\text{UV}} + a^{\text{IR}}_2 \left( \Delta^{(2)}_{\text{IR}} + \Delta^{(1)}_{\text{IR}} \ln \mu^2_{\text{IR}} \right) + a^{\text{IR}}_1 \Delta^{(1)}_{\text{IR}}$$

- scales
  $$\mu^2_{\text{UV}}, \mu^2_{\text{IR}}$$

- and poles
  $$\Delta_{\text{UV}} = \frac{c(\epsilon_{\text{UV}})}{\epsilon_{\text{UV}}}, \quad \Delta_{\text{IR},1} = \frac{c(\epsilon_{\text{IR}})}{\epsilon_{\text{IR}}}, \quad \Delta_{\text{IR},2} = \frac{c(\epsilon_{\text{IR}})}{\epsilon^2_{\text{IR}}}$$

- can be set to arbitrary real values

$\Rightarrow$ output of Collier: numerical value for full $T^N$

- cancellation of poles can be checked varying $\Delta_{\text{UV}}, \Delta_{\text{IR},1}, \Delta_{\text{IR},2}$

- prefactor $c(\epsilon) = \Gamma(1 + \epsilon)(4\pi)^\epsilon$ effectively factored out
- convention can be changed by shifting $\Delta_{\text{UV}}, \Delta_{\text{IR},1}, \Delta_{\text{IR},2}$ accordingly

- coefficient $a^{\text{UV}}$ of $1/\epsilon_{\text{UV}}$-pole returned also as separate output
default: use dimensional regularization

mass regularization supported for collinear singularities:

- declare array of squared regulator masses:
  \[ \text{minf2} = \{m_1^2, m_2^2, \ldots, m_k^2\} \]
  with complex (not-necessarily small) numerical values

- if a call of a tensor integral involves an element from minf2, the corresponding mass is
  - set to zero in IR finite integrals
  - kept as regulator mass in IR-singular integrals

- In the case of mass regularization the IR-scale \( \mu_{\text{IR}} \) can be interpreted as gluon/photon mass
Error estimates in Coli: (similar in DD)

1. PV-reduction

   - error propagation:
     \[ \delta D_r \sim \max \{ a_r \delta D_0, b_r \delta C_0, c_r \delta C_{r-1} \} \]

     with \( a_r, b_r \sim 1/\Delta^r \), \( c_r \sim 1/\Delta \)

   - after calculation: symmetry of coefficients
     \[ \delta D_r \sim |D_{i_1i_2...i_r} - D_{i_2i_1...i_r}|, \quad (0 \neq i_1 \neq i_2 \neq 0) \]
Error estimates in COLI

Error estimates in *Coli*: (similar in *DD*)

1. **PV-reduction**
   - error propagation:
     \[
     \delta D_r \sim \max\{ a_r \delta D_0, \ b_r \delta C_0, \ c_r \delta C_{r-1} \}
     \]
     with
     \[
     a_r, b_r \sim 1/\Delta^r, \quad c_r \sim 1/\Delta
     \]
   - after calculation: symmetry of coefficients
     \[
     \delta D_r \sim |D_{i_1i_2\ldots i_r} - D_{i_2i_1\ldots i_r}|, \quad (0 \neq i_1 \neq i_2 \neq 0)
     \]

2. **Expansions:** \( D_r = D_r^{(0)} + \ldots + D_r^{(g)} \)
   - neglected higher orders + error propagation from C’s:
     \[
     \delta D_r = \max\{ a_{r,g}, \ b_r \delta C_0, \ c_g \delta C_{r+g} \}
     \]
     with
     \[
     a_{r,g}, c_g \sim \Delta^g
     \]
   - extrapolation after calculation:
     \[
     \delta D_r = D_r^{(g)} \times \frac{D_r^{(g)}}{D_r^{(g-1)}}
     \]
**Precision handling**

- **max. precision (double precision)**: $O(10^{-16})$
- **target precision**: $O(10^{-8})$
- **critical precision**: $O(1)$

**accuracy flag**

- **target precision**:
  - governs selection of expansion method and expansion depth
  - balancing between precision and run-time
- **critical precision**: arguments and results of function calls are reported to an output file if estimated accuracy is worse than critical precision
- **accuracy flag**: stores status of worst integral within all function calls of the same phase space point (reinitialized for new phase-space point)
Choice of reduction scheme in COLI

Strategy for 3-, 4-point integrals of rank $r \leq r_{\text{max}}$ in COLI:

1. PV reduction:
   - accuracy for rank $r_{\text{max}}$ better than target precision?
     - yes: use PV reduction for $r \leq r_{\text{max}}$
       - done
     - no

2. Expansions:
   - do $g = 0, g_{\text{max}}$
     - accuracy for rank $r_{\text{max}}$ and expansion up to order $g$
       - yes: use expansion up to order $g$ for $r \leq r_{\text{max}}$
         - done
       - no

3. No method optimal:
   - do $r_0 = r_{\text{max}}, 0$
     - is there a method with better accuracy for rank $r_0$?
       - yes: use this method for $r \leq r_0$
       - no: use for $r \leq r_{\text{max}}$ method with best accuracy for $r_{\text{max}}$

(similar in DD)
Evaluation of one-loop amplitude leads to multiple calls for the same tensor integral (TI):

- **within one master-call:**
  - same TI appears several times in reduction tree

- **different master calls** and their reductions lead to same TI
Evaluation of one-loop amplitude leads to multiple calls for the same tensor integral (TI):

- within one master-call: same TI appears several times in reduction tree
- different master calls and their reductions lead to same TI

Cache system in Collier:

- Identify each TI-call via index pair \((N, i)\):
  \(N\) = number of external master call
  \(i\) = binary index for internal calls (propagated in reduction)
- pointers for each pair \((N, i)\) point to same address in cache if arguments of TI’s are identical
  - first call: write cache
  - further calls: read cache
- option: cache only internal calls
Conclusions

- **Collier** = fortran library for numerical calculation of scalar and tensor integrals

- **Numerical stable** results thanks to **expansion methods** for 3-,4-point integrals

- **Dimensional and mass regularization** supported, as well as **complex masses** for unstable particles

- Two independent implementations: **Collier = Coli + DD**

- Used in NLO generators **OpenLoops** and **Recola**

- Publication in preparation