COLLIER
a fortran-library for one-loop integrals

Lars Hofer
IFAE Barcelona

in collaboration with
A. Denner and S. Dittmaier

Weimar, May 2014

Status of LHC as discovery machine

- spectacular result: discovery of Higgs boson
- but: no new particles beyond the SM found so far

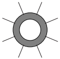
⇒ we have to be prepared for the possibility that new physics might be very subtil and show up only as small deviations from SM predictions

⇒ 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

general structure of one-loop amplitudes:

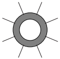


$$= \int d^Dq \, \frac{N(q)}{D_0 \cdots D_{N-1}} = \sum_r c_{\mu_1 \ldots \mu_r} \underbrace{\int d^Dq \, \frac{q^{\mu_1} \cdots q^{\mu_r}}{D_0 \cdots D_{N-1}}}_{\text{tensor integral } T^{\mu_1 \cdots \mu_r}}$$

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

One-loop amplitudes

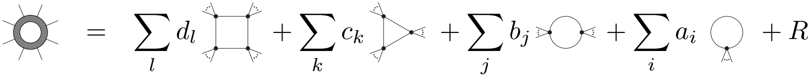
general structure of one-loop amplitudes:



$$= \int d^D q \, \frac{N(q)}{D_0 \cdots D_{N-1}} = \sum_r c_{\mu_1 \ldots \mu_r} \underbrace{\int d^D q \, \frac{q^{\mu_1} \cdots q^{\mu_r}}{D_0 \cdots D_{N-1}}}_{\text{tensor integral } T^{\mu_1 \cdots \mu_r}}$$

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

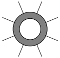
can be decomposed in terms of scalar integrals:



$$= \sum_l d_l D_0(l) + \sum_k c_k C_0(k) + \sum_j b_j B_0(j) + \sum_i a_i A_0(i) + R$$

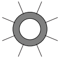
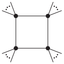
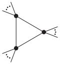
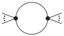
One-loop amplitudes

general structure of one-loop amplitudes:

 $$= \int d^D q \, \frac{N(q)}{D_0 \cdots D_{N-1}} = \sum_r c_{\mu_1 \dots \mu_r} \underbrace{\int d^D q \, \frac{q^{\mu_1} \cdots q^{\mu_r}}{D_0 \cdots D_{N-1}}}_{\text{tensor integral } T^{\mu_1 \cdots \mu_r}}$$

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

can be decomposed in terms of scalar integrals:

 $= \sum_l d_l$  $+ \sum_k c_k$  $+ \sum_j b_j$  $+ \sum_i a_i$  $+ R$

$$= \sum_l d_l D_0(l) + \sum_k c_k C_0(k) + \sum_j b_j B_0(j) + \sum_i a_i A_0(i) + R$$

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]

Tools for NLO

- Many tools for NLO calculatios, 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 $\longrightarrow$ publication in preparation]

COLLIER: Applications

- successfully used in many calculations of
  - NLO QCD corrections, e.g.
    $\mathrm{pp} \to \mathrm{t\bar{t}j}$ [Dittmaier,Uwer,Weinzierl '07]
    $\mathrm{pp} \to \mathrm{t\bar{t}b\bar{b}}$ [Bredenstein,Denner,Dittmaier,Pozzorini '09]
    $\mathrm{pp} \to \mathrm{WWb\bar{b}}$ [Denner,Dittmaier,Kallweit,Pozzorini '11]
  - NLO EW corrections, e.g.
    $\mathrm{e^+e^-} \to$ 4 fermions [Denner,Dittmaier,Roth,Wieders '05]
    $\mathrm{pp} \to \mathrm{Hjj}$ via VBF [Ciccolini,Denner,Dittmaier '07]
    $\mathrm{pp} \to$ dilepton+jet [Denner,Dittmaier,Kasprzik,Mück '11]
    $\mathrm{pp} \to \mathrm{H}+$dilepton [Denner,Dittmaier,Kallweit,Mück '11]
    $\mathrm{pp} \to l^+l^-\mathrm{jj}$ [Denner,LH,Scharf,Uccirati, in prep.](talk by A.Denner)
- integrated in automated NLO generators
  - OpenLoops [Cascioli,Maierhöfer,Pozzorini] (talk by 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]

$\Rightarrow$ fast and stable numerical reduction algorithm

$N = 3, 4$: PV reduction

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

  contractions:

  $p_i^{\mu} q_{\mu} \; = \; -f_i + D_i - D_0, \qquad g^{\mu\nu} q_\mu q_\nu \; = \; m_0^2 + D_0$

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

$N = 3, 4$: PV reduction

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

  contractions:

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

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

- covariant decomposition of tensors:

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

$N = 3, 4$: PV reduction

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

  contractions:

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

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

- covariant decomposition of tensors:

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

- 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 $Z_{ij} = 2 p_i p_j$

Small Gram determinants

$$\text{(PV)} \qquad \Delta T^{N,P} = \left[T^{N,P-1}, T^{N,P-2}, T^{N-1}\right]$$

small Gram determinant: $\Delta \to 0$

- $T^{N,P-1}, T^{N,P-2}, T^{N-1}$ become linearly dependent

Small Gram determinants

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

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 $\mathcal{O}(\Delta)/\Delta$-result
  - numerical determination of $T^{N,P}$ becomes unstable

Small Gram determinants

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

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 $\mathcal{O}(\Delta)/\Delta$-result
  - numerical determination of $T^{N,P}$ becomes unstable
- scalar integrals D_0, C_0, B_0, A_0 become linearly dependent $\Rightarrow \mathcal{O}(\Delta)/\Delta$-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

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

Expansion in Gram determinant

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

- 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 $\mathcal{O}(\Delta)$

Expansion in Gram determinant

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

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

- 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

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

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

- 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

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

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

- 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 $T^{N,P}$
- higher orders in Δ iteratively:
  $\mathcal{O}(\Delta^k)$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

$$(T^N)^{\mu_1\cdots\mu_P} = \sum_k \sum_{i_1,\ldots,i_k} T^{N,P}_{\underbrace{0\cdots0}_{P-k}\, i_1\cdots i_k} \{ \underbrace{g\cdots g}_{(P-k)/2}\, p_{i_1}\cdots p_{i_k}\}^{\mu_1\cdots\mu_P}$$

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

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

#TC < #TE

#TC > #TE

coefficients vs. tensors

$$(T^N)^{\mu_1\cdots\mu_P} = \sum_k \sum_{i_1,\ldots,i_k} T^{N,P}_{\underbrace{0\cdots0}_{P-k}\, i_1\cdots i_k} \{\underbrace{g\cdots g}_{(P-k)/2}\; p_{i_1}\cdots p_{i_k}\}^{\mu_1\cdots\mu_P}$$

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

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

\#TC $<$ #TE

\#TC $>$ #TE

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

calculated by OpenLoops/Recola

$$\mathcal{M} = \sum_j c^{(j)}_{\mu_1\ldots\mu_{n_j}}\, T^{\mu_1\ldots\mu_{n_j}}_{(j)}$$

Tensor Integrals

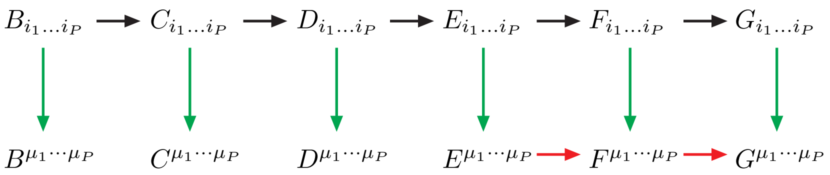
$\Rightarrow$ need full tensors!

From coefficients to tensors

$$(T^N)^{\mu_1\cdots\mu_P} = \sum_k \sum_{i_1,\ldots,i_k} T^{N,P}_{\underbrace{0\cdots0}_{P-k}\,i_1\cdots i_k} \{\underbrace{g\cdots g}_{(P-k)/2}\, p_{i_1}\cdots p_{i_k}\}^{\mu_1\cdots\mu_P}$$

In COLLIER:

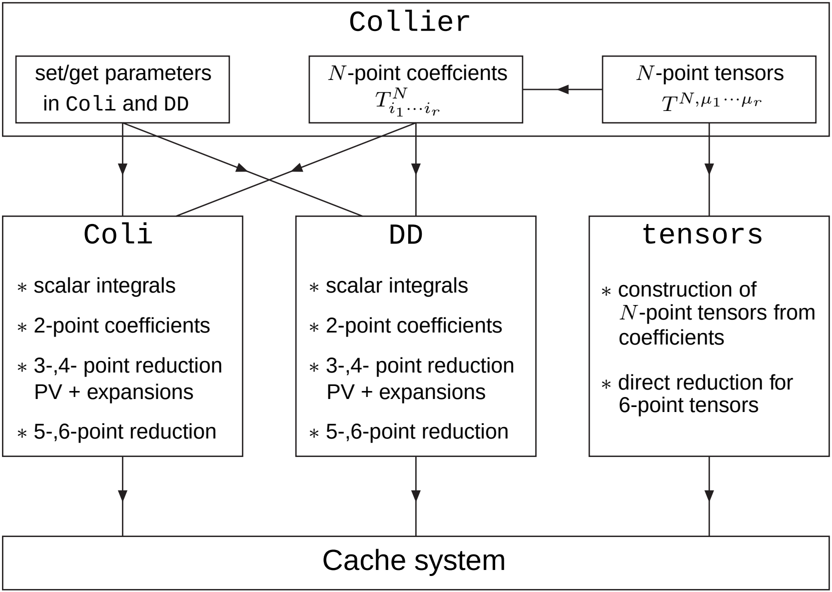
- output: coefficients $T^N_{0\cdots0i_1\cdots i_k}$ or tensors $(T^N)^{\mu_1\cdots\mu_P}$
- efficient algorithm to construct tensors from invariant coefficients for arbitrary N, P via recursive calculation of tensor structures
- for $N \geq 6$: Direct reduction at tensor level



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 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 $T^{N}_{0\cdots 0 i_1 \cdots i_k}$ or tensors $(T^N)^{\mu_1 \cdots \mu_P}$
- two independent implementations: COLI+DD

Structure of COLLIER



Output of Collier

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

$$
\begin{aligned}
T^N \quad = \quad & \Gamma(1+\epsilon)(4\pi)^\epsilon \bigg(T^N_{(\mathrm{fin})} + a^{\mathrm{UV}} \frac{1}{\epsilon_{\mathrm{UV}}} + a_2^{\mathrm{IR}} \frac{1}{\epsilon_{\mathrm{IR}}^2} + a_1^{\mathrm{IR}} \frac{1}{\epsilon_{\mathrm{IR}}} \\
& + b^{\mathrm{UV}} \log(\mu_{\mathrm{UV}}^2) + b^{\mathrm{IR}} \log(\mu_{\mathrm{IR}}^2) \bigg)
\end{aligned}
$$

Output of Collier

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

$$T^N = \Gamma(1+\epsilon)(4\pi)^\epsilon \Bigg(T^N_{(\mathrm{fin})} + a^{\mathrm{UV}} \frac{1}{\epsilon_{\mathrm{UV}}} + a_2^{\mathrm{IR}} \frac{1}{\epsilon_{\mathrm{IR}}^2} + a_1^{\mathrm{IR}} \frac{1}{\epsilon_{\mathrm{IR}}} + b^{\mathrm{UV}} \log(\mu_{\mathrm{UV}}^2) + b^{\mathrm{IR}} \log(\mu_{\mathrm{IR}}^2) \Bigg)$$

- scales $\mu_{\mathrm{UV}}^2, \quad \mu_{\mathrm{IR}}^2$
  and poles $\delta_{\mathrm{UV}} = 1/\epsilon_{\mathrm{UV}}, \quad \delta_{\mathrm{IR},1} = 1/\epsilon_{\mathrm{IR}}, \quad \delta_{\mathrm{IR},2} = 1/\epsilon_{\mathrm{IR}}^2$
  can be set to arbitrary real values
  $\Rightarrow$ output of `Collier`: numerical value for bracket $(\ldots)$

Output of Collier

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

$$T^N = \Gamma(1+\epsilon)(4\pi)^\epsilon \Big(T^N_{(\mathrm{fin})} + a^{\mathrm{UV}} \frac{1}{\epsilon_{\mathrm{UV}}} + a_2^{\mathrm{IR}} \frac{1}{\epsilon_{\mathrm{IR}}^2} + a_1^{\mathrm{IR}} \frac{1}{\epsilon_{\mathrm{IR}}} + b^{\mathrm{UV}} \log(\mu_{\mathrm{UV}}^2) + b^{\mathrm{IR}} \log(\mu_{\mathrm{IR}}^2) \Big)$$

- scales $\mu_{\mathrm{UV}}^2, \quad \mu_{\mathrm{IR}}^2$
  and poles $\delta_{\mathrm{UV}} = 1/\epsilon_{\mathrm{UV}}, \quad \delta_{\mathrm{IR},1} = 1/\epsilon_{\mathrm{IR}}, \quad \delta_{\mathrm{IR},2} = 1/\epsilon_{\mathrm{IR}}^2$
  can be set to arbitrary real values
  $\Rightarrow$ output of `Collier`: numerical value for bracket $(\ldots)$
- cancellation of poles can be checked varying $\delta_{\mathrm{UV}}, \ \delta_{\mathrm{IR},1}, \ \delta_{\mathrm{IR},2}$
- convention for $\mathrm{prefactor} = 1 + \mathcal{O}(\epsilon)$ can be changed by shifting $\delta_{\mathrm{UV}}, \ \delta_{\mathrm{IR},1}, \ \delta_{\mathrm{IR},2}$ accordingly
- coefficient a^{UV} of $1/\epsilon_{\mathrm{UV}}$ - pole returned also as separate output

Treatment of IR singularities

default: use dimensional regularization

mass regularization supported for collinear singularities:

- declare array of squared regulator masses:

$$\text{minf2} = \{m_1^2, m_2^2, ..., m_k^2\}$$

  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 μ_{IR} can be interpreted as gluon/photon mass

Choice of reduction scheme in COLI

Strategy for 3-,4-point integrals of rank $r \le r_{\max}$ in `Coli`:

(similar in `DD`)

1 PV reduction:

accuracy for rank $r_{\max}$ better than target precision? — yes → use PV reduction for $r \le r_{\max}$ — done

↓ no

2 Expansions:

do $g = 0, g_{\max}$

  accuracy for rank $r_{\max}$ and expansion up to order g better than target precision? — yes → use expansion up to order g for $r \le r_{\max}$ — done

end do

↓ no

3 No method optimal: → use for $r \le r_{\max}$ method with best accuracy for $r_{\max}$

do $r_0 = r_{\max}, 0$

  is there a method with better accuracy for rank r_0? ↔ use this method for $r \le r_0$

end do

done

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 $a_r, b_r \sim 1/\Delta^r$, $c_r \sim 1/\Delta$

- after calculation: symmetry of coefficients

$$\delta D_r \sim |D_{i_1 i_2 \ldots i_r} - D_{i_2 i_1 \ldots i_r}|, \qquad (0 \neq i_1 \neq i_2 \neq 0)$$

Error estimates in COLI

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

$\boxed{1}$ 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 $\quad a_r, b_r \sim 1/\Delta^r, \qquad c_r \sim 1/\Delta$

- after calculation: symmetry of coefficients

$$\delta D_r \sim |D_{i_1 i_2 \ldots i_r} - D_{i_2 i_1 \ldots i_r}|, \qquad (0 \neq i_1 \neq i_2 \neq 0)$$

$\boxed{2}$ Expansions: $D_r = D_r^{(0)} + \ldots + D_r^{(g)}$

- neglected higher orders + error propagation from C's:

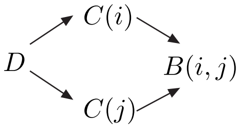
$$\delta D_r = \max\{\, a_{r,g},\, b_r\, \delta C_0,\, c_g\, \delta C_{r+g}\,\}$$

with $\quad a_{r,g}, c_g \sim \Delta^g$

- extrapolation after calculation: $\quad \delta D_r = D_r^{(g)} \times \dfrac{D_r^{(g)}}{D_r^{(g-1)}}$

Cache system

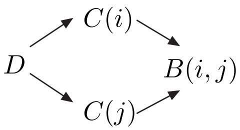
Evaluation of one-loop amplitude leads to multiple calls for the same tensor integral (TI):



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

Cache system

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
- DD: internal cache for 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