Reduze 2, matroids, symbols

and their application to

Top quark pairs at two loops

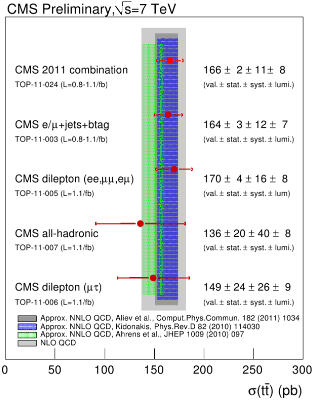
Andreas v. Manteuffel

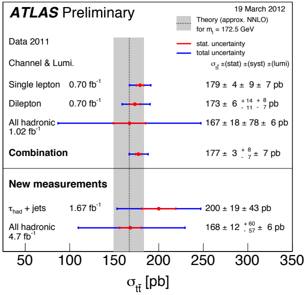
Johannes Gutenberg Universität Mainz

PRiSMA

Theory Seminar
DESY Zeuthen
15 November 2012

Top pair production at the LHC





- **LHC precision** below NLO accuracy already
- **full** *WWbb* at NLO
- recent **threshold/resummations**, **approx. NNLO**: Beneke, Falgari, Klein, Piclum, Schwinn, Ubiali, Yan; Moch, Uwer, Vogt; Cacciari, Czakon, Mangano, Mitov, Nason; Ahrens, Ferroglia, Neubert, Pecjak, Yang; Kidonakis
- full NNLO for $q\bar{q}$: Bärnreuther, Czakon, Mitov '12

Gauge invariant subsets in two-loop contributions

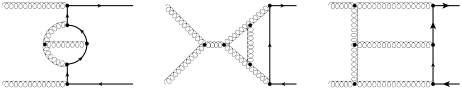
gg **channel**: 789 two-loop diagrams (+ ghost init.) contrib. to 16 coeff.:

$$2\,\mathrm{Re}\left\langle \mathcal{M}^{(0)} | \mathcal{M}^{(2)} \right\rangle = 2C_F N_c \Big(N_c^3 \mathbf{A} + N_c \mathbf{B} + \frac{1}{N_c}\mathbf{C} + \frac{1}{N_c^3}\mathbf{D}$$
$$+ N_c^2 n_l \mathbf{E_l} + n_l \mathbf{F_l} + \frac{n_l}{N_c^2}\mathbf{G_l} + N_c n_l^2 \mathbf{H_l} + \frac{n_l^2}{N_c}\mathbf{I_l}$$
$$+ N_c^2 n_h \mathbf{E_h} + n_h \mathbf{F_h} + \frac{n_h}{N_c^2}\mathbf{G_h} + N_c n_h^2 \mathbf{H_h} + \frac{n_h^2}{N_c}\mathbf{I_h}$$
$$+ N_c n_l n_h \mathbf{H_{lh}} + \frac{n_l n_h}{N_c}\mathbf{I_{lh}} \Big)$$

$q\bar{q}$ **channel**: 218 two-loop diagrams, 10 coefficients

example: for **leading N_c coefficient** A we need:

- 300 two-loop diagrams (+ ghost initiated), e.g.:



- two independent ratios of scales
- up to: 4-point, 7 propagators, 4 loop momenta in numerator

Steps toward complete NNLO calculation

for $2\,Re\,\langle \mathcal{M}^{(0)} | \mathcal{M}^{(2)} \rangle$:

small mass expansion:

- Czakon, Mitov, Moch (2006) for $q\bar{q}$, gg

IR poles:

- Ferroglia, Neubert, Pecjak, Yang (2009) for $q\bar{q}$, gg

$q\bar{q}$ with **full dependence** on s, t, m_t, μ:

- numerical result for all contributions:
  Czakon (2008)
- analytical result for fermionic:
  Bonciani, Ferroglia, Gehrmann, Maitre, Studerus (2008)
- analytical result for leading N_c:
  Bonciani, Ferroglia, Gehrmann, Studerus (2009)

gg with **full dependence** on s, t, m_t, μ:

- analytical result for leading N_c:
  Bonciani, Ferroglia, Gehrmann, A.v.M., Studerus (2010)
- analytical result for light fermionic:
  Bonciani, Ferroglia, Gehrmann, A.v.M., Studerus (in preparation)
- numerical result for all contributions:
  Czakon, Bärnreuther (in preparation)

Organisation of our calculation

Recipe

1. generate **Feynman diagrams** with `QGRAF` by Nogueira
2. build **interference** terms
3. **reduce** scalar integrals to masters with **parallel Laporta**
4. **solve masters** with differential equations
5. **renormalize**: $\overline{MS}$, pole mass

⇒ **analytical result** in terms of generalized polylogarithms, allows fast **numerical evaluation**, **expansions**, ...

various tasks automatized in computer program `Reduze 2`

```
________      ________
___  __ \_________  /___  _________
__  /_/ /  _ \  __  /_  / / /__  /  _ \
_  _, _//  __/ /_/ / / /_/ /__  /_  __/
/_/ |_| \___/\__,_/  \__,_/ _____\___/
```

Reduze 2 - Distributed Feynman Integral Reduction

A.v.M., Studerus

arXiv:1201.4330

http://projects.hepforge.org/reduze

Reduze 2 and external software

- **key features**
  - fully parallized reductions, resumable
  - topological analysis of diagrams and sectors
- **open source** C++
- **libraries / programs** used (no proprietary mandatory):
  - GiNaC by Bauer, Frink, Kreckel
  - yaml-cpp
  - optional: MPI
  - optional: Berkeley DB
  - optional: Fermat CAS by Lewis (closed source)
- **interfaces**
  - input: QGRAF, YAML
  - output: FORM, Mathematica, Maple

Reductions using integration by parts (IBP) identities

example: massive 1-loop tadpole

$$\int d^d k \frac{1}{k^2 - m^2}$$

calculating **IBP** identity

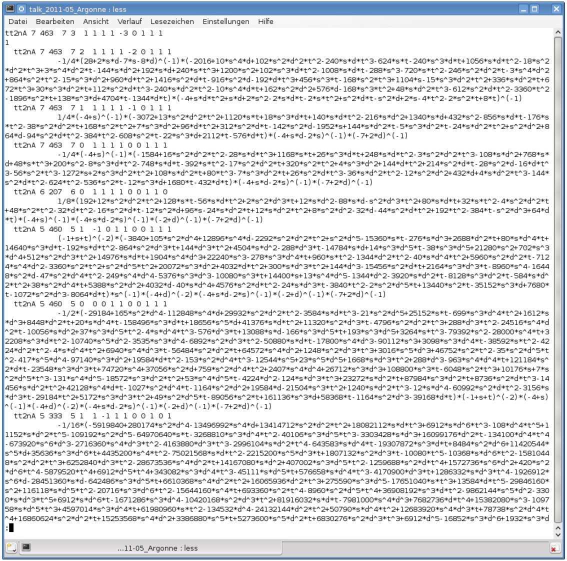
$$\begin{aligned}
0 = & \int d^d k \frac{\partial}{\partial k_\mu} \left(k_\mu \frac{1}{k^2 - m^2} \right) \\
= & \int d^d k \left(\frac{d}{k^2 - m^2} - \frac{2k^2}{(k^2 - m^2)^2} \right) \\
= & \int d^d k \left(\frac{d}{k^2 - m^2} - \frac{2(k^2 - m^2 + m^2)}{(k^2 - m^2)^2} \right) \\
= & (d-2) \int d^d k \frac{1}{k^2 - m^2} - 2m^2 \int d^d k \frac{1}{(k^2 - m^2)^2}
\end{aligned}$$

gives directly **reduction** of integral with additional numerator

$$\int d^d k \frac{1}{(k^2 - m^2)^2} = \frac{(d-2)}{2m^2} \int d^d k \frac{1}{k^2 - m^2}$$

Example of reduction

in reality we are after something like this



⇒ need some computer program

Laporta's algorithm

index integrals:

- define **integral family**: set of propagators $\{1/D_1, \ldots, 1/D_N\}$ such that: all scalar products with loop momenta are linear combinations of D_i
- **sector** defined by subset of propagators (denominators)
- counting propagator exponents indexes integrals:

$$\text{Feynman integrals of some topologies} \quad \rightarrow \quad \mathbb{Z}^N$$

$$\int d^d k_1 \cdots d^d k_L \frac{1}{D_1^{n_1} \cdots D_N^{n_N}} \mapsto \{n_1, \ldots, n_N\} \quad \text{with } n_i \in \mathbb{Z}$$

Laporta algorithm

1. define **ordering** for integrals $I(n_1, \ldots, n_N)$
2. generate **integration by parts identities** (IBPs): **sparse system** of equations
3. **solve linear system** of equations

based on:

- Chetyrkin, Tkachov '81

public implementations:

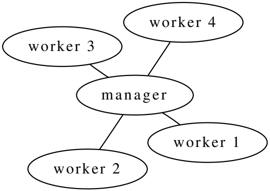
- Anastasiou: `AIR`, Smirnov: `FIRE`, Studerus: `Reduze 1`
- parallel variant: AvM, Studerus: `Reduze 2`

Reduze 2: parallelization of Laporta-Algorithm

- generate the system of equations
- sort equations in blocks with the **same leading integral**

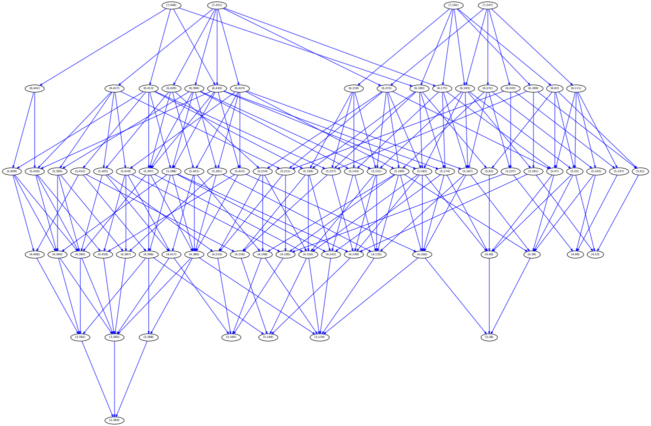
$$
\begin{array}{|l|}
\hline
\mathbf{I}_5 + c_{14}\mathbf{I}_4 + c_{13}\mathbf{I}_3 = 0 \\
\mathbf{I}_5 + c_{24}\mathbf{I}_4 + c_{22}\mathbf{I}_2 = 0 \\
\mathbf{I}_5 + c_{33}\mathbf{I}_3 + c_{32}\mathbf{I}_2 = 0 \\
\hline
\mathbf{I}_3 + c_{42}\mathbf{I}_2 = 0 \\
\mathbf{I}_3 + c_{51}\mathbf{I}_1 = 0 \\
\hline
\mathbf{I}_2 + c_{61}\mathbf{I}_1 = 0 \\
\hline
\end{array}
$$

- send blocks to workers



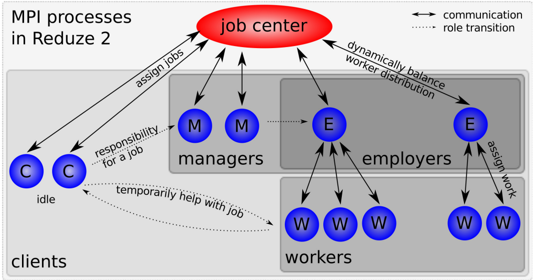
Reduze 2: parallel execution of jobs

consider reduction of multiple sectors:

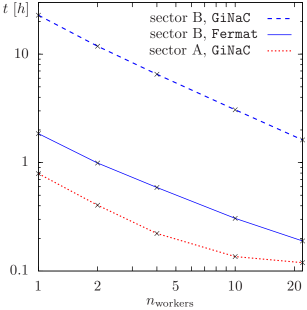


⇒ parallel reduction of several sectors ("jobs"), balance workers between them

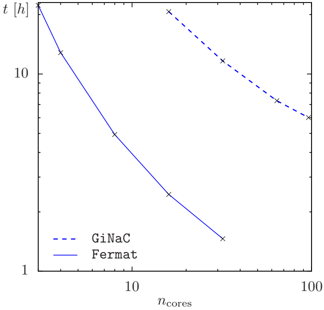
Reduze 2: distributed Laporta algorithm



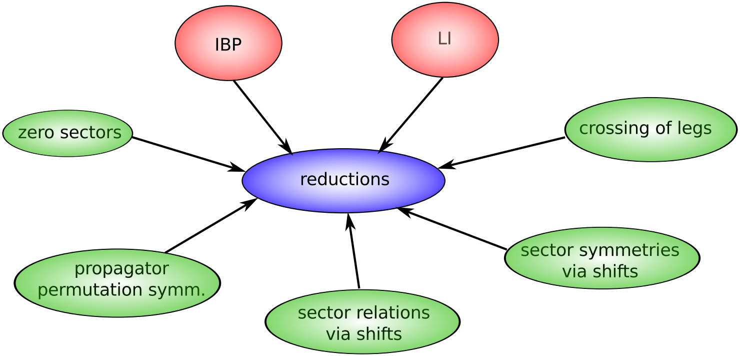
reduction of single sector:



reduction of sector selection:



Removing ambiguities for integrals



Reduze 2: sectors, graphs and matroids

integral A (propagators) ⟵ ∃shift ? ⟶ **integral B** (propagators)

↓ *construct* ↓ *construct*

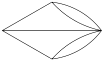
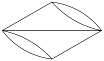
graph A ⟵ isomorphic ? ⟶ **graph B**

integral A (propagators) ⟵ ∃shift ? ⟶ **integral B** (propagators)

↓ *construct* ↓ *construct*

graph A ⟵ isomorphic ? ⟶ **graph B**

- **problem**: graphs not unique !

e.g.:  ⟵ same propagators but not isomorphic ⟶ 

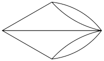
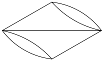
Reduze 2: sectors, graphs and matroids

integral A (propagators) ⟵ ∃shift ? ⟶ **integral B** (propagators)

↓ *construct* ↓ *construct*

graph A ⟵ isomorphic ? ⟶ **graph B**

- **problem**: graphs not unique !

e.g.:  ⟵ same propagators but not isomorphic ⟶ 

- **solution**: select unique representative by allowing for **twists**

idea based on **theorem** by Bogner, Weinzierl (2010):

first Symanzik polynomials $\mathcal{U}$ isomorphic $\Leftrightarrow$ graphs isomorphic **up to twists**

proof based on Whitney's theorem for isomorphisms of graph matroids

Definition of matroid

A matroid is a pair $(E, \mathcal{I})$ where E finite ground set, $\mathcal{I}$ collection of subsets of E, the **"independent sets"**, and

- $\emptyset \in \mathcal{I}$
- if $I \in \mathcal{I}$ and $I' \subset I$ then also $I' \in \mathcal{I}$
- if $I_1, I_2 \in \mathcal{I}$, $|I_1| < |I_2|$ then $\exists e \in I_2 - I_1$ with $I_1 \cup \{e\} \in \mathcal{I}$

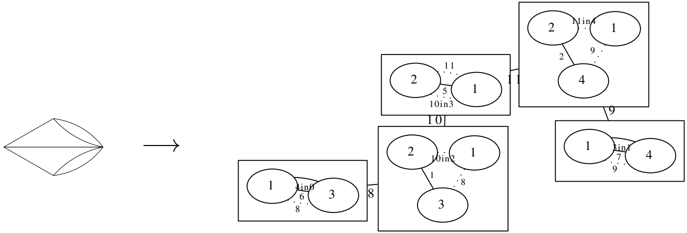
- generalizes notion of **linear dependency**
- graph matroid: dependencies of **edges**
- application to Feynman graph: propagators relevant, no reference to vertices

propagators of two vacuum diagrams related by shift $\Leftrightarrow$ graph matroids isomorphic

Algorithm: shift finder

1. generate graph for sector
2. colour edges according to masses
3. connect external legs with a new vertex
4. decompose into triconnected components (Hopcroft, Tarjan '73; Gutwenger, Mutzel '01)
5. minimize graph by twists
6. check for graph isomorphism (McKay '81)

example:



tree of triconnected components
(dashed "virtual edges" mark positions for Tutte twists)

Usage

Input files:

```
# kinematics.yaml

kinematics:
  incoming_momenta: [p1, p2]
  outgoing_momenta: [p3, p4]
  momentum_conservation: [p4, p1 + p2 - p3]
  kinematic_invariants:
    - [mt, 1]
    - [s,  2]
    - [t,  2]
  scalarproduct_rules:
    - [[p1,p1],  0]
    - [[p2,p2],  0]
    - [[p3,p3],  mt^2]
    - [[p1+p2, p1+p2], s]
    - [[p1-p3, p1-p3], t]
    - [[p2-p3, p2-p3], -s-t+2*mt^2] # == u
  symbol_to_replace_by_one: mt
```

```
# integralfamilies.yaml

integralfamilies:
  - name: planarbox
    loop_momenta: [k1, k2]
    propagators:
      - [ k1, 0 ]
      - [ k2, 0 ]
      - [ k1-k2, 0 ]
      - [ k1-p1, 0 ]
      - [ k2-p1, 0 ]
      - [ k1-p1-p2, 0 ]
      - [ k2-p1-p2, 0 ]
      - [ k1-p3, "mt^2" ]
      - [ k2-p3, "mt^2" ]
    permutation_symmetries:
      - [ [ 1, 6 ], [ 2, 7 ] ]  # ki<->-ki+p1+p2, p1<->p2, p3<->p4
      - [ [ 1, 2 ], [ 4, 5 ], [ 6, 7 ], [ 8, 9] ]  # k1<->k2
```

automatic determination of sector properties:

- zero sectors
- unphysical sectors
- generation of graphs for physical sectors
- symmetry shifts of sectors
- shifts which identifies different sectors (also between different integral families)
- handles crossings of external momenta
- upcoming: generation of maximally symmetric family for given graph

Application example

auto-generated shifts for non-planar double box family:

figures : less

Datei Bearbeiten Ansicht Verlauf Lesezeichen Einstellungen Hilfe


```
---
sectormappings:
  name: box2n
  zero_sectors:
    t=0: [0]
    t=1: [1, 4, 16, 64, 256]
    t=2: [3, 5, 6, 12, 17, 18, 20, 24, 48, 65, 66, 68, 72, 80, 96, 192, 257, 260, 272, 320]
    [ ... truncated ]
  sectors_without_graph:
    t=3: [22, 28, 52, 82, 84, 88, 104, 112, 276]
    t=4: [23, 29, 30, 53, 54, 60, 83, 85, 86, 89, 90, 92, 105, 106, 108, 113, 114, 116, 120, 21
    t=5: [31, 55, 61, 87, 91, 93, 94, 107, 109, 110, 115, 117, 118, 121, 122, 124, 211, 213, 21
    [ ... truncated ]
  sector_relations:
    3: [[box2p, 3], [[k1, k1], [k2, k2]]]
    6: [[box2p, 3], [[k1, -p1+k2], [k2, k1]]]
    7: [[box2p, 11], [[k1, k1-p1], [k2, k2]]]
    12: [[box2p, 3], [[k1, k1-p1], [k2, k2-p2]]]
    13: [[box2p, 11], [[k1, k1-p1], [k2, k2-p2]]]
    15: [[box2p, 58], [[k1, -p1+k2], [k2, k1-p1-p2]]]
    18: [[box2p, 3], [[k1, k2+p3], [k2, k1]]]
    19: [[box2px13x24, 11], [[k1, k1+p3], [k2, k2]]]
    24: [[box2p, 3], [[k1, k2+p3], [k2, k1-p2]]]
    25: [[box2px13x24, 11], [[k1, k1+p3], [k2, k2-p2]]]
    26: [[box2px12x34, 11], [[k1, k2+p3], [k2, k1-p2]]]
    27: [[box2px13, 58], [[k1, k2+p3], [k2, k1-p2+p3]]]
    [ ... truncated ]
  crossed_sector_relations:
    x12:
      207: [[box2n, 207], [[k1, -k2-p2], [k2, -k1-p1]]]
      335: [[box2n, 335], [[k1, -k2-p2], [k2, -k1-p1]]]
      463: [[box2n, 463], [[k1, -k2-p2], [k2, -k1-p1]]]
    x123:
      335: [[box2n, 335], [[k1, -k2-p2], [k2, k1+p1-k2]]]
    x1234:
      335: [[box2nx14x23, 335], [[k1, -k1+p1+k2-p3]]]
    x124:
      335: [[box2nx14x23, 335], [[k1, -k1+p1+k2-p3], [k2, -k1+p1+p2-p3]]]
    x1243:
      335: [[box2nx12x34, 335], [[k2, k1-k2+p2]]]
  sector_symmetries:
    207:
      - [[k1, -k1-p1], [k2, -k2-p2]]
      - [[k1, -k2-p2], [p1, p2], [k2, -k1-p1], [p2, p1], [p3, p1+p2-p3]]
      - [[k1, k2], [p1, p2], [k2, k1], [p2, p1], [p3, p1+p2-p3]]
    463:
      - [[k1, k2], [p1, p2], [k2, k1], [p2, p1], [p3, p1+p2-p3]]
(END)
```


figures : less

Available jobs in Reduze 2

andreas : bash

Datei Bearbeiten Ansicht Verlauf Lesezeichen Einstellungen Hilfe

```
andreas@chili:~$ reduze -h jobs

List of available job types:

apply_crossings:                  Generates reduction results for crossed sectors.
cat_files:                        Concatenates files.
collect_integrals:                Collects all integrals appearing in the input file.
compute_diagram_interferences:    Computes interferences of diagrams.
compute_differential_equations:   Computes derivatives of integrals wrt invariants.
export:                           Exports to FORM, Mathematica or Maple format.
find_diagram_shifts:              Matches diagrams to sectors via graphs.
find_diagram_shifts_alt:          Matches diagrams to sectors via combinatorics.
generate_identities:              Generates identities like IBPs for given seeds.
generate_seeds:                   Generates integrals from a sector.
insert_reductions:                Inserts reductions in expressions.
normalize:                        Simplifies linear combinations and equations.
print_reduction_info_file:        Analyzes reductions in a file.
print_reduction_info_sectors:     Analyzes reductions available for sectors.
print_sector_info:                Prints diagrams and other information for sectors.
reduce_files:                     Reduces identities in given files.
reduce_sectors:                   Reduces integrals from a selection of sectors.
run_reduction:                    Low-level job to run a reduction.
select_reductions:                Selects reductions for integrals.
setup_sector_mappings:            Finds shifts between sectors via graphs.
setup_sector_mappings_alt:        Finds shifts between sectors via combinatorics.
sum_terms:                        Sums terms.
test:                             Empty job template.
verify_same_terms:                Verifies two files contain the same terms.

andreas@chili:~$
```

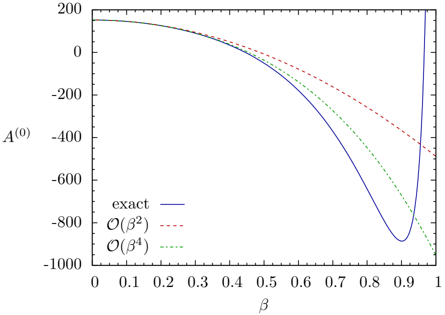
andreas : bash

detailed description: see builtin on-line help

Top quark pair production at two loops: gluon channel results

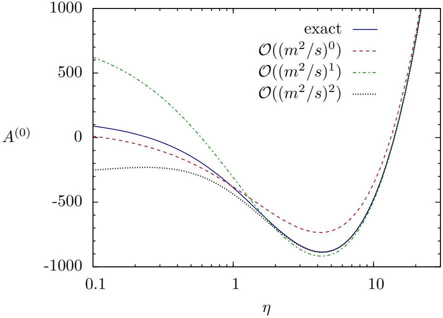
Results for gg leading N_c: expansions

threshold expansion:



$\beta = \sqrt{1 - 4m_t^2/s}$,
c.m. scatt. angle $= \pi/2$

small mass expansion:



$\eta = \frac{s}{4m_t^2} - 1$,
c.m. scatt. angle $= \pi/2$

(careful with phase space for small m expansion:
don't introduce forw.-backw. asymmetry, $\phi = -(t - m_t^2)/s = const$)

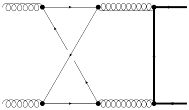
Bonciani, Ferroglia, Gehrmann, A.v.M., Studerus (2010)

LIGHT FERMION LOOPS IN $gg \to t\bar{t}$

- **analytical** result for **light fermion** loops in $gg \to t\bar{t}$:
  Bonciani, Ferroglia, Gehrmann, A.v.M., Studerus (in preparation)

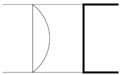
- required calculation of **11 new master integrals**
  - differential equations + Mellin-Barnes for const.
  - used: `MB.m` by Czakon '05, planar: `AMBRE` by Gluza, Kajda, Riemann '07 + Yundin '10

- most difficult integrals: **massive non-planar double box**

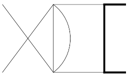


  - 3 master integrals (+ subtopo MIs)
  - analytical solution for massless case: Tausk 1999
  - 3 + 1 scale problem: s, t, u and m appear naturally

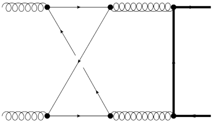
subtopo with s,t:



subtopo with s,u:



Analytical solution for massive non-planar double box



solved with differential equations:
A.v.M, Studerus (in preparation)

- all masters up to and including finite parts
- determined integration constants from
  - regularity conditions
  - symmetry conditions
  - Mellin-Barnes (cmp. with SecDec by Carter, Heinrich '10) in kinematical limits (finite const. up to 3-fold scaleless M.B.)

found **result** in terms of (x, y):

- linear combinations of GPLs with rational prefactors
- transcendality up to 4
- 805 GPLs total, 166 two-dim.
- GPLs with argument y: weights $\{0, -1, -x, -1/x, -1/x - x, -1/x - x + 1\}$
- GPLs with argument x: weights $\{0, \pm 1, \pm i, (1 \pm i\sqrt{3})/2\}$
  - for univariate (cyclotomic) cmp. Ablinger, Blümlein, Schneider '11

Generalised polylogarithms

Remiddi, Gehrmann; Goncharov:

Definition of generalised polylogarithms (GPLs)

$$G(\vec{0}_n; x) = \frac{1}{n!}\log^n(x),$$

$$G(a; x) = \int_0^x dt\ f_a(t), \qquad \text{for } a \neq 0$$

$$G(a, \vec{b}; x) = \int_0^x dt\ f_a(t) G(\vec{b}; t), \qquad \text{for } a \neq 0$$

with weight functions for (complex) weight a:

$$f_a(x) = \frac{1}{x - a}$$

- $G(a \neq 0; x) = \log\left(\frac{a-x}{a}\right)$, $G(\vec{0}_{n-1}, 1; x) = Li_n(x)$, $G(\vec{0}_n, \vec{1}_p; x) = S_{n,p}(x)$
- for weights $0, 1, -1$, GPLs specialize to harmonic polylogarithms (HPLs), Remiddi, Vermaseren (1999)
- shuffle algebra, **symbols** (see $N = 4$ remainder func), ...

Simplification via symbols

result for $1/\epsilon$ of scalar master:

- using GPLs with variables (x, y): **62 different GPLs** (23 two-dim.)
- choosing different arguments via **symbols**, this **simplifies to**:

$$
\begin{aligned}
M_1\big|_{1/\epsilon} = & \frac{1}{16 m^2 y_1 z_1 (y_1 + z_1)} \times \\
& \times \left(\mathrm{Li}_3 \left(\frac{y_1 z_1}{y_1 + z_1} \right) \right. \\
& \quad + \mathrm{Li}_2 \left(\frac{y_1 z_1}{y_1 + z_1} \right) \left(\log(-y_1 - z_1) - \log y_1 - \log z_1 \right) \\
& \quad \left. + \text{polynomial in } \log(y_1),\ \log(z_1),\ \log(-y_1 - z_1),\ \log\left(1 - \frac{y_1 z_1}{y_1 + z_1}\right) \right)
\end{aligned}
$$

with $y_1 := -t/m^2 + 1$,
$z_1 := -u/m^2 + 1$

- recent progress with symbols: Duhr, Gangl, Rhodes '11, Duhr '12

Symbol calculus for multiple polylogarithms

Motivation for symbol calculus

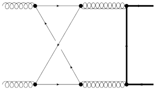
symbol calculus:

- tool to exploit functional relations between polylogarithms in a systematic way
  - works: simplify and transform functions; Zagier ('91), Goncharov ('95), Gangl ('02)
  - wish: perform integrations, see Chavez, Duhr '12 and talks by C. Bogner

first application in theoretical physics:

- $N = 4$ supersymmetric Yang-Mills two-loop six-point remainder function
  Del Duca, Duhr, Smirnov (09): $\mathcal{O}(10^3)$ multi-dimensional polylogs
- simplifies to few-line expression with few Li_4, Li_2 and ln !
  found with symbols by Goncharov, Spradlin, Vergu, Volovich ('10)

symbols also very welcome for **QCD**, e.g.:

- e.g. 
- corner integral alone gives $\mathcal{O}(10^3)$ GPLs, many two-dimensional
- numerical evaluation slow, stability problematic
- switching basis functions, expansions, analytical continuation difficult
- simplifications guided by symbols possible; Ferroglia, AvM, Studerus

crucial progress very recently:

- construction of functions for given symbol; Duhr, Gangl, Rhodes ('11)
  - applied to HPLs; Bühler, Duhr ('11)
- extended symbol calculus from coproduct: **constants**;
  Goncharov ('02), Brown ('11), Duhr ('12)
  demonstrated for QCD amplitude; Duhr ('12)

Definition of symbol map

Let G be a generalized polylogarithm with

$$\mathrm{d}G = \sum_i \hat{G}_i \, \mathrm{d} \ln(R_i)$$

where R_i is a rational function of the polylog arguments. The **symbol map $\mathcal{S}$**

$$\mathcal{S}(G) = \sum_i \mathcal{S}(\hat{G}_i) \otimes R_i \,,$$

associates a tensor with the polylogarithm.

examples:

- $\mathcal{S}(\ln x) = x$
- $\mathcal{S}(\mathrm{Li}_3\, x) = -((1-x) \otimes x \otimes x)$
- $\mathcal{S}(G(1,0,-1,-1,x)) = (1+x) \otimes (1+x) \otimes x \otimes (1-x)$

Rules for symbols

- $R_1 \cdots \otimes (R_a R_b) \otimes \cdots R_k = R_1 \cdots \otimes R_a \otimes \cdots R_k + R_1 \cdots \otimes R_b \otimes \cdots R_k$ (log law)
- $R_1 \cdots \otimes (c R_a) \otimes \cdots R_k = R_1 \cdots \otimes R_a \otimes \cdots R_k$ for constant c
- preserves shuffle product

Example for symbol calculus

goal: derive "simplification formula" for $\mathrm{Li}_2(1/x)$ with $0 < x < 1$, $\mathrm{Im}\, x = \varepsilon$

$$\begin{aligned}\mathcal{S}(\mathrm{Li}_2(1/x)) &= -(-1 + 1/x) \otimes (1/x) \\ &= (1 - x) \otimes x - x \otimes x \\ &= \mathcal{S}(-\mathrm{Li}_2(x) - (1/2)\ln^2 x)\end{aligned}$$

reproduces the highest degree part of the full answer

$$\mathrm{Li}_2(1/x) = -\mathrm{Li}_2(x) - (1/2)\ln^2 x + i\pi \ln x - (2/3)\pi^2$$

note: works at **highest degree** only

- $\mathcal{S}(\ln(-x)) = \mathcal{S}(\ln(x))$: no info on discontinuity
- $\mathcal{S}(\pi) = \mathcal{S}(\zeta_3) = 0$: no constants

"integrating the symbol" $\Rightarrow$ **algorithmic reduction** (structured with shuffle eliminators): Duhr, Gangl, Rhodes ('11)

Systematic reduction to basis function with symbols

$gg \to t\bar{t}$ **at two-loops**, ferm. contrib. (incl. non-planar part)

- pick specific coefficient in finite part: linear combination of polylogs
- pick uniform weight 4 part
- representation not unique (due to crossings etc.), imaginary polylogs
- here: 182 GPLs in total, 84 weight 4 non-Li_k GPLs
- symbol contains slots (weight have unit roots):

$$\{-1+x, x, 1+x, y, 1+y, x+y, 1+xy, 1+x^2+xy, 1-x+x^2+xy\}$$

algorithmic conversion at highest degree to new set of real basis function:

1. partition (4) (no shuffles), $\mathrm{Li}_{22}(R_1, R_2)$, $\mathrm{Li}_4(R_1)$
2. partition (3,1): $\mathrm{Li}_3(R_1)\ln(R_2)$
3. partition (2,2): $\mathrm{Li}_2(R_1)\,\mathrm{Li}_2(R_2)$
4. partition (2,1,1): $\mathrm{Li}_2(R_1)\ln(R_2)\ln(R_3)$
5. partition (1,1,1,1) (rest): $\ln(R_1)\ln(R_2)\ln(R_3)\ln(R_4)$

result:

- 28 Li_{22}, 48 Li_4, 58 Li_3, 18 Li_2, 11 ln
- all real

What about **subleading degree** *terms (const $\times$ polylog) ?*

- accessible by **coproduct**: Goncharov ('02), Brown ('11)
- extended symbol calculus based on coproduct by Duhr ('12) with:

$$\Delta(\pi) = \pi \otimes 1$$
$$\Delta(\zeta_k) = \zeta_k \otimes 1 + 1 \otimes \zeta_k \qquad \text{for } k \text{ (odd)}$$

example:
$$\begin{aligned}\Delta_{1,1}\left(\mathrm{Li}_2(1/x)\right) &= -\ln(1-1/x) \otimes \ln(1/x) \\ &= \ln(1-x) \otimes \ln(x) - \ln(x) \otimes \ln(x) + i\pi \otimes \ln(x) \\ &= \Delta_{1,1}\left(-\mathrm{Li}_2(x) - (1/2)\ln^2(x) + i\pi\ln(x)\right)\end{aligned}$$

reproduces identity up to pure constant, fix by limits or numerical evaluation:

$$\mathrm{Li}_2(1/x) - \left(-\mathrm{Li}_2(x) - (1/2)\ln^2(x) + i\pi\ln(x)\right) = -6.5797362673929\cdots = -(2/3)\pi^2$$

results for $gg \to t\bar{t}$: **systematic reductions** give:

- considerable simplifications, heavy improvement in access wrt traditional techniques
- beyond classical polylogs, need Li_{22}

Conclusions

- $gg \to t\bar{t}$: analytical two-loop corrections
  - leading N_c + light fermionic
  - massive non-planar double box: analytical solution
- Reduze 2: open source tool
  - parallelized reductions of Feynman integrals
  - graph matroid algorithm: determine shifts between sectors or diagrams
- symbol + coproduct refinement:
  - powerful algorithmic treatment of multiple polylogs
  - simplifications of QCD integrals with mass
  - outlook: direct solving of differential equations ?

Supplementary Slides

Ingredients for full NNLO calculation

- **VV**: two-loop ME for $q\bar{q} \to t\bar{t}$ and $gg \to t\bar{t}$
- **RV**: one-loop ME for $t\bar{t}$ + 1 parton
  Dittmaier, Uwer, Weinzierl '07
- **RR**: tree level ME for $t\bar{t}$ + 2 partons
- **subtraction terms**: up to 2 unresolved partons needed
  Gehrmann-De Ridder, Ritzmann '09; Daleo et al. '09; Boughezal et al. '10; Glover, Pires '10; Czakon '10, '11; Anastasiou, Herzog, Lazopoulos '10; Abelof, Gehrmann-De Ridder '11, '12; Gehrmann, Monni '11; Bierenbaum, Czakon, Mitov '11
- **combined $q\bar{q}$**: Bärnreuther, Czakon, Mitov '12

consider **VV** ($2 \to 2$ ingredients):

$$\sum_{\text{spin,colour}} |\mathcal{M}|^2 = 16\pi^2\alpha_s^2 \left[\mathcal{A}_0 + \left(\frac{\alpha_s}{\pi}\right)\mathcal{A}_1 + \left(\frac{\alpha_s}{\pi}\right)^2 \mathcal{A}_2 + \mathcal{O}(\alpha_s^3)\right]$$

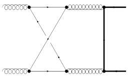
with

$$\mathcal{A}_0 = \left\langle \mathcal{M}^{(0)} | \mathcal{M}^{(0)} \right\rangle$$

$$\mathcal{A}_1 = 2\,Re \left\langle \mathcal{M}^{(0)} | \mathcal{M}^{(1)} \right\rangle$$

$$\mathcal{A}_2 = \left\langle \mathcal{M}^{(1)} | \mathcal{M}^{(1)} \right\rangle + 2\,Re \left\langle \mathcal{M}^{(0)} | \mathcal{M}^{(2)} \right\rangle$$

$\langle \mathcal{M}^{(1)} | \mathcal{M}^{(1)} \rangle$: Kniehl, Körner, Merebashvili, Rogal '05-'08, Anastasiou, Aybat '08



- Mellin-Barnes representation:

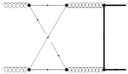
$$I(a_9) = (-1)^{-1+a_9}\pi^d \frac{\Gamma(-2+d/2)^2}{\Gamma(a_9)\Gamma(-4+d)\Gamma(-6-a_9+3d/2)}$$
$$\int_{-i\infty}^{i\infty}\frac{\mathrm{d}z_1}{2\pi i}\int_{-i\infty}^{i\infty}\frac{\mathrm{d}z_2}{2\pi i}\int_{-i\infty}^{i\infty}\frac{\mathrm{d}z_3}{2\pi i}\int_{-i\infty}^{i\infty}\frac{\mathrm{d}z_4}{2\pi i}\int_{-i\infty}^{i\infty}\frac{\mathrm{d}z_5}{2\pi i}$$
$$(-s)^{-6-a_9+d-z_1-z_2-z_5}(-t_1)^{z_1}(-u_1)^{z_2}(m^2)^{z_5}$$
$$\Gamma(-z_1)\Gamma(-z_2)\Gamma(-z_3)\Gamma(-z_4)\Gamma(-z_5)$$
$$\Gamma(1+z_1+z_3)\Gamma(1+z_2+z_3)\Gamma(1+z_1+z_4)\Gamma(1+z_2+z_4)$$
$$\Gamma(4-d/2+z_1+z_2+z_3+z_4)\Gamma(-5-a_9+d-z_1-z_2-z_3-z_5)$$
$$\Gamma(-5-a_9+d-z_1-z_2-z_4-z_5)\Gamma(6+a_9-d+z_1+z_2+z_3+z_4+z_5)$$
$$\Gamma(a_9+z_1+z_2+2z_5))/\Gamma(2+z_1+z_2+z_3+z_4)^2$$

with

$$a_9 = \text{power of massive propagator}$$
$$t_1 = t+m^2$$
$$u_1 = u+m^2$$

A non-planar massive double box

Analytical solution



- e.g. leading poles of scalar integral very simple:

$$\frac{1}{N(\epsilon)} I(1) = \frac{1}{\epsilon^4} \cdot \frac{y_1 + z_1}{32 x_s^2 y_1 z_1}$$
$$+ \frac{1}{\epsilon^3} \cdot \frac{7(y_1 + z_1) - 6 * (y_1 + z_1) * \log x_s + 3(y_1 - z_1)(\log y_1 - \log z_1)}{96 x_s^2 y_1 z_1}$$
$$+ \frac{1}{\epsilon^2} \cdot \left(c_1 \log x_s + c_2 \log^2 x_s + c_3 \log y_1 + c_4 \log^2 y_1 + c_5 \log z_1 + c_6 \log^2 z_1\right.$$
$$\left. + c_7 \log x_s \log y_1 + c_8 \log x_s \log z_1 + c_9 \log y_1 \log z_1\right)$$

- however: differential equations work with 2 + 1 on-shell variables ($s + t + u = 2m^2$)
- GPL conversions involved, e.g.:

$$G(-1, 0, -1; z) = G\left(-1, 0, -1; -\frac{1}{x} - x - y\right)$$
$$= G\left(1 - \frac{1}{x} - x, -\frac{1}{x} - x, 1 - \frac{1}{x} - x; y\right) + 42 \text{ more GPLs}$$
$$= G\left(\frac{1 - y + \sqrt{-3 - 2y + y^2}}{2}, \frac{-y + \sqrt{-4 + y^2}}{2}, \frac{1 - y + \sqrt{-3 - 2y + y^2}}{2}; x\right) + 39 \text{ more GPLs}$$

- solved up to few integration constants

Definition of the coproduct

For a multiple polylogarithm

$$I(a_0; a_1, \ldots, a_n; a_{n+1}) = \int_{a_0}^{a_{n+1}} \frac{\mathrm{d}t}{t - a_n} I(a_0; a_1, \ldots, a_{n-1}; t)$$

the coproduct Δ is defined according to Goncharov ('02):

$$\Delta\left(I(a_0; a_1, \ldots, a_n; a_{n+1})\right) = \sum_{0=i_1<\ldots<i_{k+1}=n} I(a_0; a_{i_1}, \ldots, a_{i_k}; a_{n+1}) \otimes \prod_{p=0}^{k} I(a_{i_p}; a_{i_p+1}, \ldots, a_{i_{p+1}-1}$$

examples:

- $\Delta(\ln(x)) = 1 \otimes \ln(x) + \ln(x) \otimes 1$
- $\Delta(\mathrm{Li}_2(x)) = 1 \otimes \mathrm{Li}_2(x) - \ln(1-x) \otimes \ln(x) + \mathrm{Li}_2(x) \otimes 1$
- $\Delta(\ln(x)\ln(y)) = 1 \otimes (\ln(x)\ln(y)) + \ln(x) \otimes \ln(y) + \ln(y) \otimes \ln(x) + (\ln(x)\ln(y)) \otimes 1$

Rules for the coproduct

- coassociativity $(\mathrm{id} \otimes \Delta)\,\Delta = (\Delta \otimes \mathrm{id})\,\Delta$
- compatible with product: $\Delta(a \cdot b) = \Delta(a) \cdot \Delta(b)$ where $(a_1 \otimes a_2) \cdot (b_1 \otimes b_2) \equiv (a_1 \cdot b_1) \otimes (a_2 \cdot b_2)$

note: coproduct means "decomposition"

Graded decomposition with the coproduct

- Hopf algebra of multiple polylogs **graded by weight**:

$$\mathcal{H} = \bigoplus_{n=0}^{\infty} \mathcal{H}_n$$

since coproduct preserves weight we may decompose

$$\mathcal{H}_n \xrightarrow{\Delta} \bigoplus_{p+q=n} \mathcal{H}_p \otimes \mathcal{H}_q$$

and define $\Delta_{p,q}$ to be the part with values in $\mathcal{H}_p \otimes \mathcal{H}_q$

- iterated coproduct:

$$\mathcal{H} \xrightarrow{\Delta} \mathcal{H} \otimes \mathcal{H} \xrightarrow{\Delta \otimes \mathrm{id}} \mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H} \xrightarrow{\Delta \otimes \mathrm{id} \otimes \mathrm{id}} \mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H}$$

and corresponding parts $\Delta_{p,q,\ldots,r}$

- symbol $\mathcal{S}$ = maximally iterated coproduct $\Delta_{1,\ldots,1}$ mod π

systematic procedure: back to $gg \to t\bar{t}$ **at two-loops** application:

1. fix maximum degree 4 part with symbol, substract from original expression
2. take $\Delta_{1,1,1,1}$, use unique set of real ln, gives

$$i\pi \otimes \Delta_{1,1,1}(\text{something})$$

   match "something" at degree 3 to basis functions using symbols
3. take $\Delta_{2,1,1}$, use unique set of real ln, Li_2, gives

$$\pi^2 \otimes \Delta_{1,1}(\text{something})$$

   match "something" at degree 2 to basis functions using symbols
4. take $\Delta_{3,1}$, use unique set of real ln, Li_2, Li_3, gives

$$\zeta(3) \otimes \text{something1} + i\pi^3 \otimes \text{something2}$$

   read off "somethingi" (just logs) at degree 1
5. numerical evaluation, match against

$$i\pi\zeta_3, \quad \pi^4$$

(requires matching of degree 3 and lower, obtained by recursion)

results for $gg \to t\bar{t}$:

- considerable simplifications for poles
- finite parts: heavy improvement in access wrt traditional techniques