Reduze 2 and top quark pair production at two loops

Andreas v. Manteuffel

Universität Zürich[UZH]

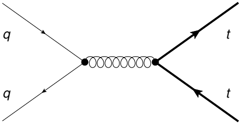
and Cedric Studerus (Uni Bielefeld)

10th International Symposium on Radiative Corrections
RADCOR 2011
26-30 September, Mamallapuram, India

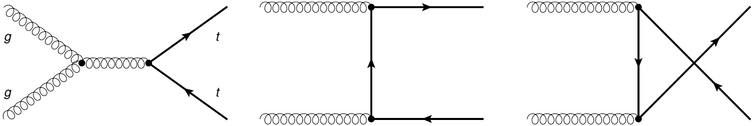
Top pair production at hadron colliders

- Top quark pair production at tree level:

  $q\bar{q} \to t\bar{t}$ channel (dominant at Tevatron $\sim$ 85%):



  $gg \to t\bar{t}$ channel (dominant at LHC $\sim$ 90%):



- **LHC** precision below NLO accuracy already: see talks by J. Katzy, M. Pieri
- **resummations**: see talks by M. Neubert, P. Falgari, M. Stahlhofen and refs. therein
- **full** *WWbb* at NLO: see talks by A. van Hameren, S. Pozzorini

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; Gehrmann, Monni '11; Bierenbaum, Czakon, Mitov '11

consider $2 \to 2$ 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

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

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

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

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

Gauge invariant subsets in two-loop contributions

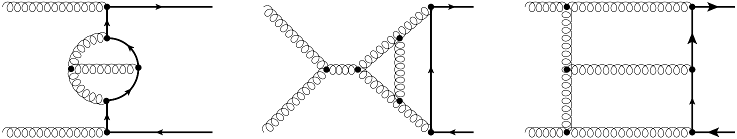
gg **channel**: 789 two-loop diagrams (+ ghost init.) contrib. to 16 coeff.:

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

$q\bar{q}$ **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 $2\,Re\left\langle \mathcal{M}^{(0)} | \mathcal{M}^{(2)} \right\rangle$:

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 the 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`
A.v.M., Studerus (in preparation)

Reduze 2

Reduze 2:

- computer program to perform reductions of Feynman integrals to master integrals
- written in C++, open source, by A.v.M., C. Studerus (in preparation)
- successor of Reduze 1 by C. Studerus

new features:

- fully parallelized reductions, resumable
- topological analysis of diagrams and sectors
- generation of differential equations for masters
- computation of QCD diagram interferences up to masters

Reduze 2 and external software

- libraries / programs used:
  - 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

Laporta 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
- 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

public implementations:

- Anastasiou: `AIR`, Smirnov: `FIRE`, Studerus: `Reduze 1`

Parallelization of Laporta-Algorithm

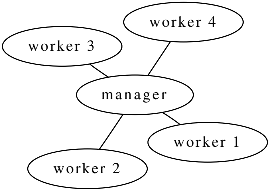
- generate the system of equations
- sort equations in blocks with the **same leading integral**

$$\begin{array}{|l|}\hline I_5 + c_{14} I_4 + c_{13} I_3 = 0 \\ I_5 + c_{24} I_4 + c_{22} I_2 = 0 \\ I_5 + c_{33} I_3 + c_{32} I_2 = 0 \\ \hline \end{array}$$

$$\begin{array}{|l|}\hline I_3 + c_{42} I_2 = 0 \\ I_3 + c_{51} I_1 = 0 \\ \hline \end{array}$$

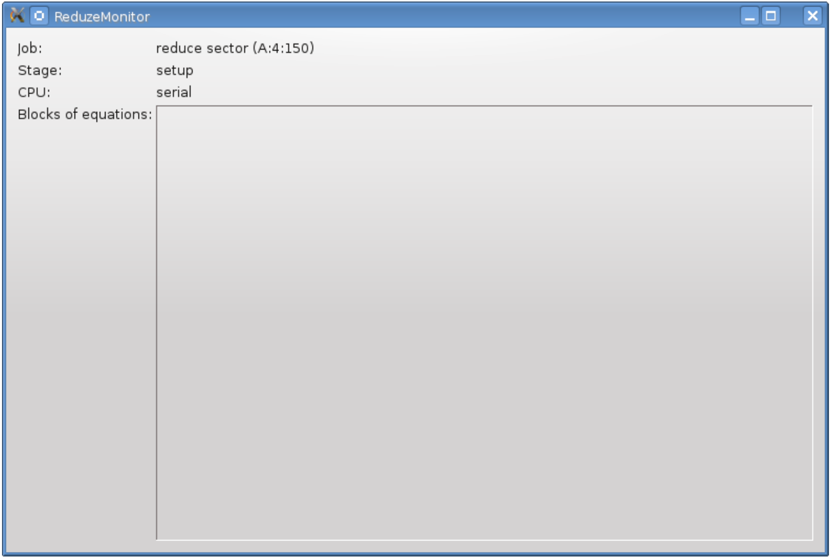
$$\begin{array}{|l|}\hline I_2 + c_{61} I_1 = 0 \\ \hline \end{array}$$

- send blocks to workers



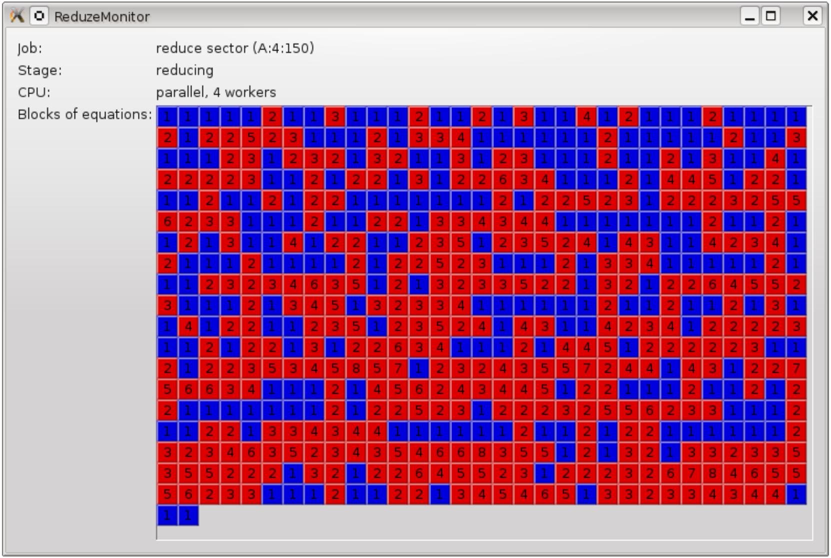
Example: distributed reduction of one sector

visualisation of reduction:



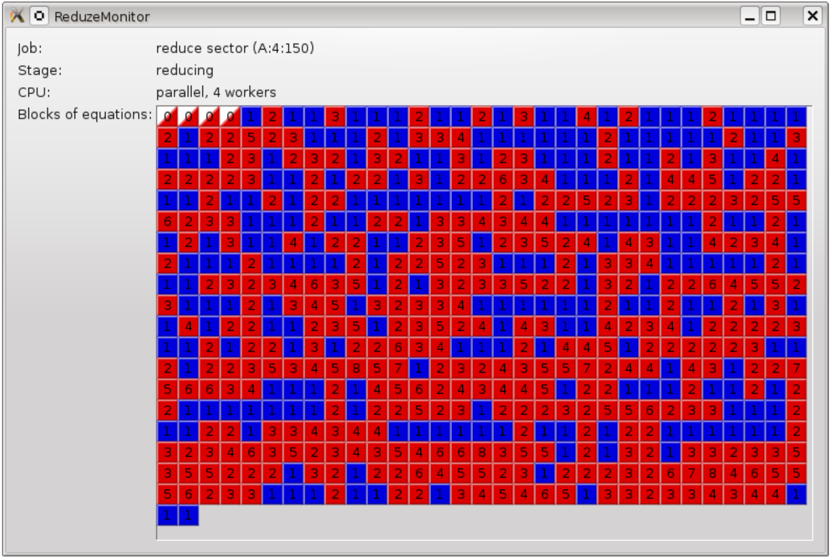
Example: distributed reduction of one sector

visualisation of reduction:



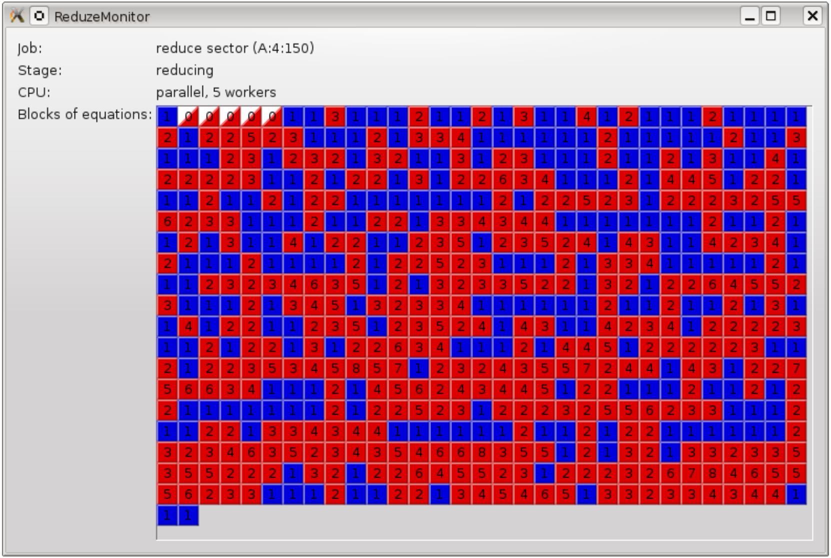
Example: distributed reduction of one sector

visualisation of reduction:



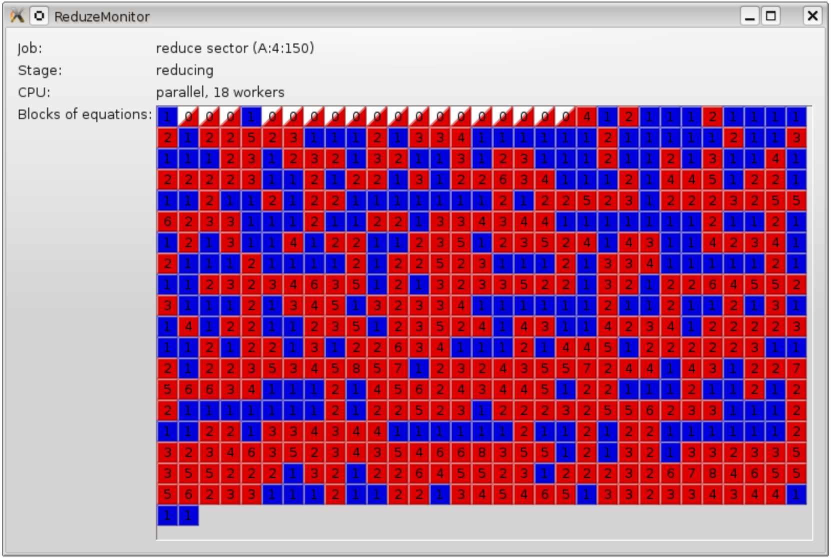
EXAMPLE: DISTRIBUTED REDUCTION OF ONE SECTOR

visualisation of reduction:



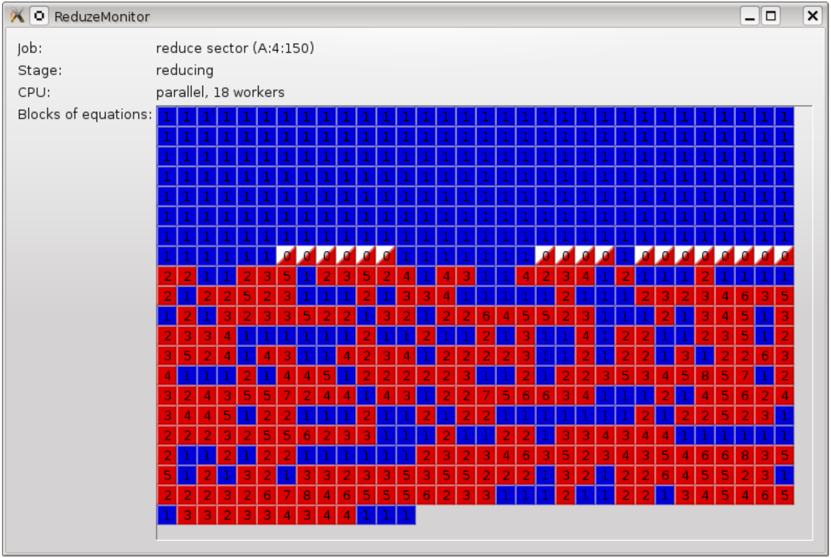
Example: distributed reduction of one sector

visualisation of reduction:



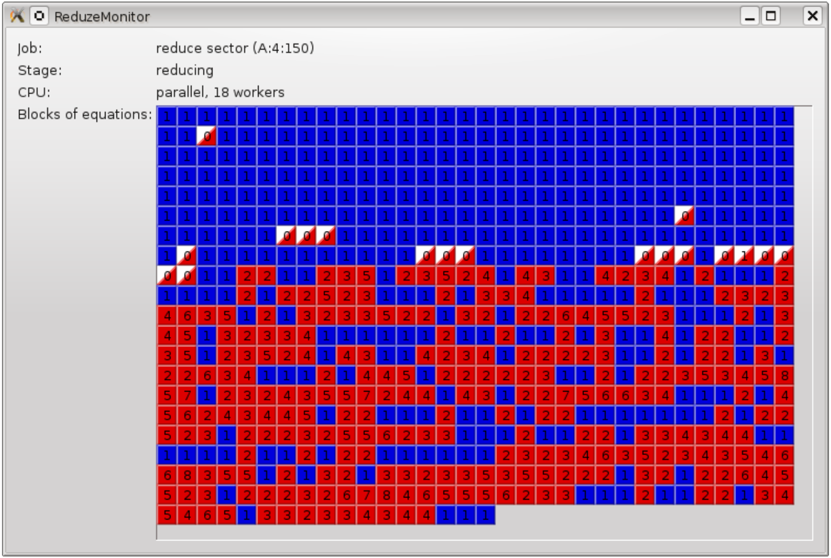
Example: distributed reduction of one sector

visualisation of reduction:



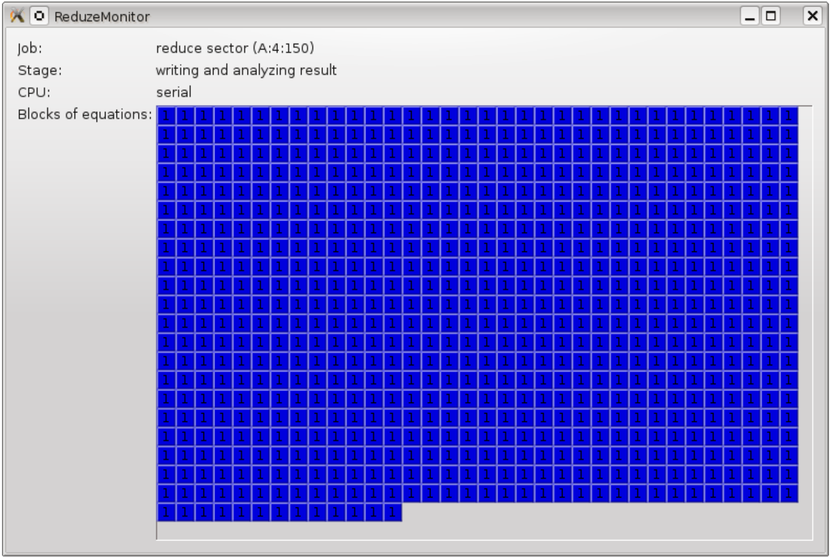
Example: distributed reduction of one sector

visualisation of reduction:



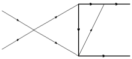
Example: distributed reduction of one sector

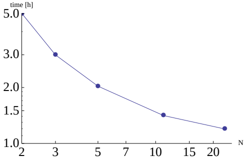
visualisation of reduction:

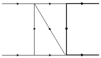


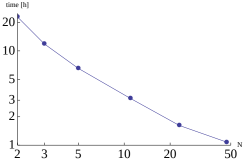
Performance: single sectors

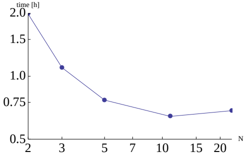
GiNaC | Fermat

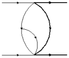


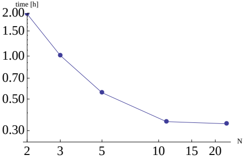








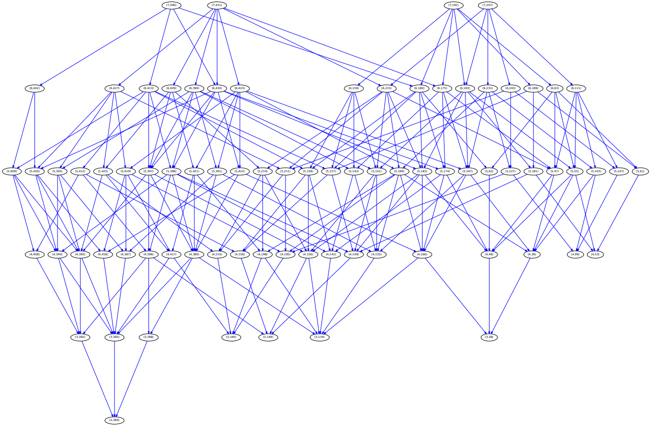




- Fermat very fast GCD (speedup often $>$ 10x)
- scaling problem specific
- single sector: $t\bar{t}$ typically benefits from up to 10-20 cores (speedup 5x-10x)

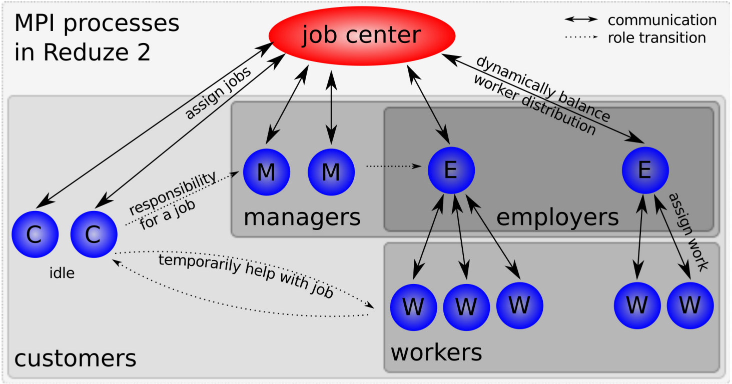
Parallel execution of jobs

consider reduction of multiple sectors:



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

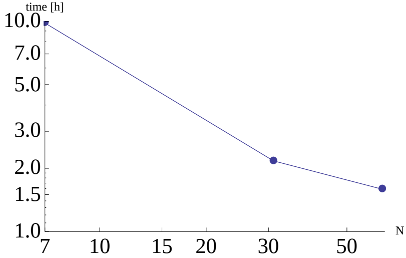
MPI processes in Reduze 2



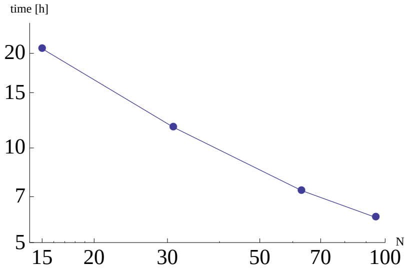
Performance: multiple sectors

using GiNaC, database (not fastest):

$s \in \{0, 1, 2, 3\}$



$s \in \{0, 1, 2, 3, 4\}$



- scaling problem specific
- multiple sectors: may benefit from up to 100 cores (speedup up to $\mathcal{O}(50\text{x})$)

Available jobs in Reduze 2

andreas : bash

Datei Bearbeiten Ansicht Verlauf Lesezeichen Einstellungen Hilfe

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

List of available job types:

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.
extract_database_contents:          Extracts intermediate results from aborted reduction.
find_diagram_shifts:                Matches diagrams to integral families by shifting loop momenta.
find_sector_shifts:                 Finds equivalent sectors.
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 zero sectors.
sum_terms:                          Sums terms.
verify_same_identities:             Verifies two files contain the same identities.

andreas@chili:~$
```

andreas : bash

Usage

Input files:


```
# globals.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 ] ]
      - [ [ 1, 2 ], [ 4, 5 ], [ 6, 7 ], [ 8, 9] ]
```


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

Usage

Job file:

```
# job_file.yaml

jobs:
 - reduce_sectors:
     sector_selection:
       integralfamily: planarbox
       select_recursively: [ 182 ]
     identities:
       ibp:
         - { r: [t, 5], s: [0, 1] }
 - select_reductions:
     input_file: "myintegrals"
     output_file: "myintegrals.sol"
```

and starts the program with

```
mpirun -np 32 reduze job_file.yaml
```

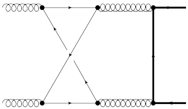
Top quark pair production at two loops: recent progress

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)

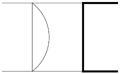
- required calculation of **11 new master integrals**
  - differential equations + Mellin-Barnes for const.
  - used: `MB.m` by Czakon '05, planar: `AMBRE` by Gluza, Kajda, Riemann '07 + Yundin '10

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

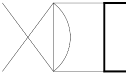


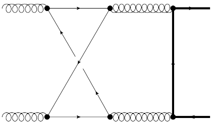
  - 3 master integrals (+ subtopo MIs)
  - analytical solution for massless case: Tausk 1999
  - 3 + 1 scale problem: s, t, u and m appear naturally

subtopo with s,t:



subtopo with s,u:





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 $\{0, \pm 1, \pm i, (1 \pm i\sqrt{3})/2\}$

Outlook: simplification via symbols

symbols for GPLs introduced by Goncharov (2009):
drastic simplification of $N = 4$ rem. func.: Goncharov, Spradlin, Vergu, Volovich (2010)

Definition: Symbol

for a rational function $R(x)$

$$\text{symbol}(\log R(x)) = R(x)$$

for function f(x) with $df = \sum_i g_i(x) d\log(R_i(x))$

$$\text{symbol}(f(x)) = \sum_i (\text{symbol}\, g_i(x)) \otimes (R_i(x))$$

e.g.: $\text{symbol}(G(1, -1, 0, x)) = x \otimes (x + 1) \otimes (x - 1)$

Logarithm law for symbols

$$R_1 \cdot \otimes (R_a R_b) \otimes \cdots R_k = R_1 \cdot \otimes R_a \otimes \cdots R_k + R_1 \cdot \otimes R_b \otimes \cdots R_k$$
$$R_1 \cdot \otimes (c R_a) \otimes \cdots R_k = R_1 \cdot \otimes R_a \otimes \cdots R_k \qquad \text{for constant } c$$

Simplification via symbols: first results

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**:

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

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

Conclusions

- Reduze 2: tool for parallelized reductions of Feynman integrals and more
- progress for analytical $gg \to t\bar{t}$ two-loop corrections: light fermionic
- analytical solution for massive non-planar double box
- choice of polylogs via symbols simplifies QCD integrals with mass