A beginner's guide to Reduze 2

Andreas v. Manteuffel

Advances in QCD and Applications to Hadron Colliders
Argonne, October 26-28, 2016

A master recipe to compute amplitudes

1. generate Feynman diagrams and apply Feynman rules
2. handle tensors and Dirac algebra, reduce to scalar loop integrals I_k:

$$\Rightarrow \qquad A = \sum_k c_k(p_{ij}^2) I_k(p_{ij}^2)$$

   (integrals indexed by families with tuples k of propagator powers)
3. reduce scalar loop integrals to master integrals M_k

$$\Rightarrow \qquad A = \sum_l d_l(p_{ij}^2) M_l(p_{ij}^2)$$

   (only a small number of masters)
4. compute master integrals (or look up in the literature)

$$\Rightarrow \qquad A = \sum_m f_m(p_{ij}^2) G_m(p_{ij}^2)$$

   (some fundamental functions G_k, e.g. multiple polylogarithms)

note1: potentially also for phase space integrals

note2: alternatives to avoid factorial growth of complexity for many external legs:

- recursion relations + on-shell-methods (a.k.a. "generalized unitarity"))
- direct integration of loop momenta

Feynman integrals form a linear vector space

$$I = \int \mathrm{d}^d k_1 \cdots \mathrm{d}^d k_L \frac{1}{D_1^{a_1} \cdots D_N^{a_N}} \qquad a_i \in \mathbb{Z}, \quad D_1 = k_1^2 - m_1^2 \text{ etc.}$$

family of loop integrals:

- fulfill linear relations: integration-by-parts (IBP) identities
- systematic reduction to master integrals possible
- think of it as linear vector space with some finite basis
- specific basis choices:
  - canonical basis for method of differential equations [Henn]
  - basis of finite integrals for direct integration (analyt., numeric.) [Panzer; Panzer, AvM, Schabinger]

reductions are technical challenge:

1. often a bottleneck of the computation
2. various ideas + implementations

Integration-by-parts (IBP) identities

in dimensional regularisation, integral over total derivative vanishes:

$$0 = \int \mathrm{d}^d k_1 \cdots \mathrm{d}^d k_L \frac{\partial}{\partial k_i^\mu} \left(k_j^\mu \frac{1}{D_1^{a_1} \cdots D_N^{a_N}} \right)$$

$$0 = \int \mathrm{d}^d k_1 \cdots \mathrm{d}^d k_L \frac{\partial}{\partial k_i^\mu} \left(p_j^\mu \frac{1}{D_1^{a_1} \cdots D_N^{a_N}} \right)$$

where p_j are external momenta, $a_i \in \mathbb{Z}$, $D_1 = k_1^2 - m_1^2$ etc.

integral reduction:

- express arbitrary integral for given problem via few basis integrals
- integration-by-parts (IBP) reductions [Chetyrkin, Tkachov '81]
- public codes: `Air` [Anastasiou], `Fire` [Smirnov], `Reduze 1` [Studerus], `Reduze 2` [AvM, Studerus], `LiteRed` [Lee]
- possible: exploit structure at algebra level
- here: Laporta's approach

Simple example

example: massive 1-loop tadpole

$$\int d^d k \frac{1}{(k^2 - m^2)^a} \qquad \text{with } a \in \mathbb{Z}$$

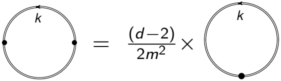
calculating IBP identity

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

gives directly reduction of integral with additional numerator

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

diagrams for $a = 1$:

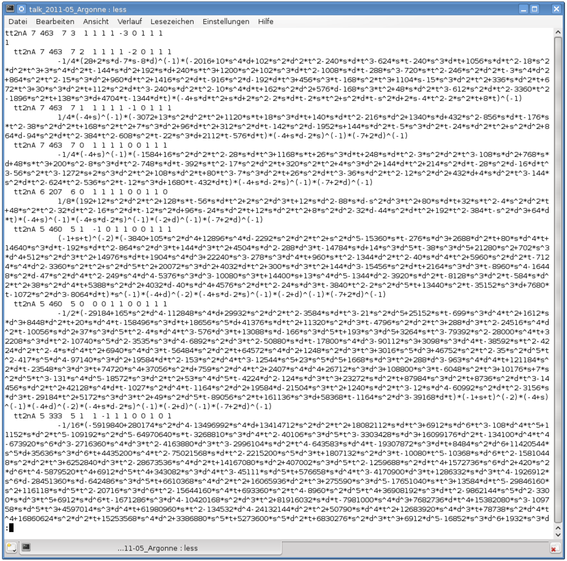


Laporta's algorithm:

1. index integrals by propagator exponents: $I(a_1, \ldots, a_N)$
2. define ordering (e.g. fewer denominators means simpler)
3. generate IBPs for explicit values $a_1, \ldots, a_N$
4. results in linear system of equations
5. solve linear system of equations

Realistic example

in reality we might be after something like this



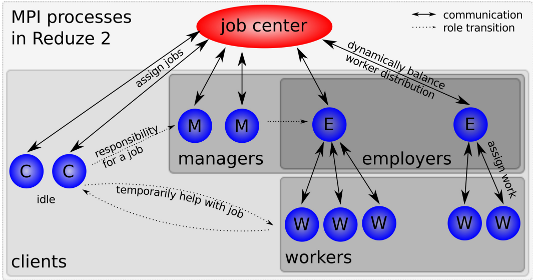
⇒ need some computer program

Reduze 2 [AvM, C. Studerus]
arXiv:1201.4330, HepForge

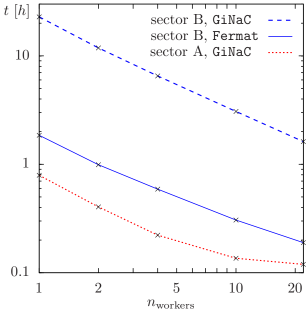
based on Reduze 1 [Studerus]
uses GiNaC [Bauer, Frink, Kreckel]
and Fermat [Lewis]

- open source C++ program
- distributed Feynman integral reduction
- advanced shift finders
- *brand-new version 2.1* with many new features
  - linear propagators
    3-loop heavy flavour Wilson coefficients in DIS [with Blümlein ea '13-'16]
  - phase space integrals
    soft-virtual N^3LO Higgs and DY [Li,AvM,Schabinger,Zhu '14,'14]
    $e^+e^- \to t\bar{t}$ 2-loop soft function [AvM,Schabinger,Zhu '14]
  - finite integral finder
    new type of singularity resolution [AvM,Schabinger,Panzer '14]
  - dimensional recurrences
  - works also on Mac (thanks to Stephen Jones)
  - ...

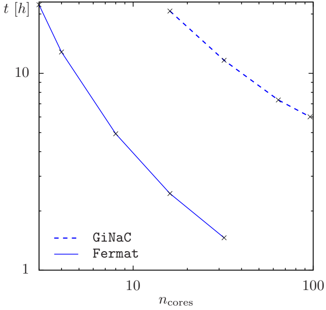
Reduze 2: distributed Laporta algorithm



reduction of single sector:



reduction of sector selection:



Shift relations

- assigning loop momenta to propagators not unique:

$$\int \mathrm{d}^d k \frac{1}{(k^2)^{a_1}((k+p)^2)^{a_2}} \xrightarrow{k \to -k+p} \int \mathrm{d}^d k \frac{1}{((k+p)^2)^{a_1}(k^2)^{a_2}}$$

- shift induces relations between indexed integrals of same or different sectors:

$$D_1 := k^2$$
$$D_2 := (k+p)^2$$
$$I_{a_1,a_2} := \int \mathrm{d}^d k \frac{1}{D_1^{a_1} D_2^{a_2}}$$

shift relation:

$$\Rightarrow \quad I_{a_1,a_2} = I_{a_2,a_1}$$

- also needed to map diagrams to sectors of families

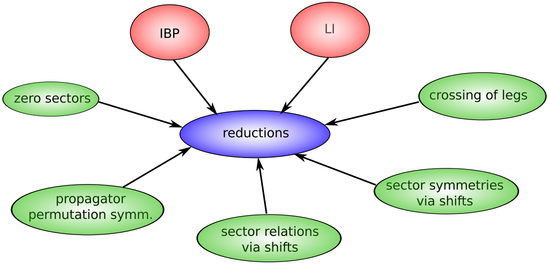
Shift relations

- consider trafo of loop momenta

$$k_i \to \sum_{j=1}^{l} M_{ij} k_j + \sum_{j=1}^{m} N_{ij} p_j$$

with $|\det M| = 1$

- systematic shift finding:
  - pure combinatoric
  - graph/matroid based
- Reduze 2 handles 3 types of shifts:
  - permutation symmetries: for all sectors of a family, very efficient
  - sector relations: eliminate complete sectors
  - sector symmetries: relates integrals of same sector (sometimes important !)

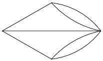
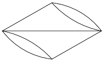


integral A (propagators) ⟵ ∃shift ? ⟶ integral B (propagators)

↓ *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

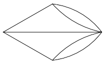
based on: first Symanzik polynomials $\mathcal{U}$ isomorphic ⇔ graphs isomorphic up to twists

theorem by [Bogner, Weinzierl (2010)], proof based on [Whitney's theorem] for isomorphisms of graph matroids

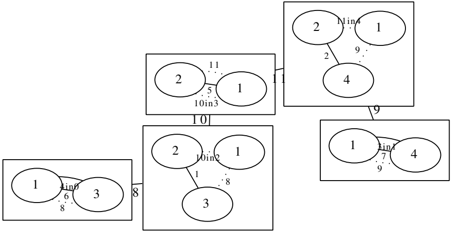
Algorithm: shift finder

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:



$\longrightarrow$

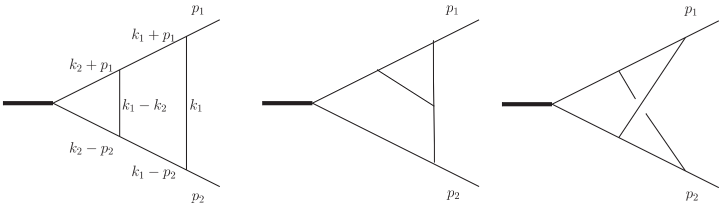


tree of triconnected components
(dashed "virtual edges" mark positions for [Tutte] twists)

employed by Reduze for sector relations, shift identities, diagram matching etc.

Example 1: setup, reductions, finite integrals

massless two-loop form factors:

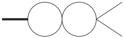


on-line demo of Reduze:

- setup of input files
- setup sector mappings
- reductions
- change to basis of finite integrals
- other jobs

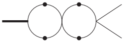
Form factors @ 2-loops: to finite basis [AvM, Schabinger, Panzer '15]

(4−2ε)



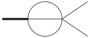
$$= \frac{1}{\epsilon^2}\,\frac{1}{(1-\epsilon)^2}$$

(6−2ε)



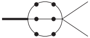
,

(4−2ε)



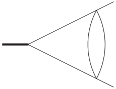
$$= \frac{1}{\epsilon}\,\frac{-4}{(2-\epsilon)^2(1-\epsilon)^2(1-2\epsilon)}$$

(8−2ε)



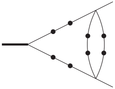
,

(4−2ε)



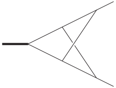
$$= \frac{1}{\epsilon^2}\,\frac{16(3-2\epsilon)(2-3\epsilon)}{(3-\epsilon)^2(2-\epsilon)^2(1-\epsilon)^3(1+2\epsilon)}$$

(10−2ε)



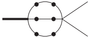
,

(4−2ε)



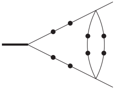
$$= \frac{1}{\epsilon^4}\,\frac{-4(2-3\epsilon)(14-81\epsilon+115\epsilon^2+14\epsilon^3-132\epsilon^4+72\epsilon^5)}{(2-\epsilon)^2(1-\epsilon)^2(1-2\epsilon)^2(2-\epsilon-2\epsilon^2)}$$

(8−2ε)



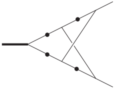
$$+ \frac{1}{\epsilon^4}\,\frac{-16(1+\epsilon)(3-2\epsilon)(2-3\epsilon)(10-61\epsilon+102\epsilon^2-44\epsilon^3-8\epsilon^4)}{(3-\epsilon)^2(2-\epsilon)^2(1-\epsilon)^3(1-2\epsilon)(1+2\epsilon)(2-\epsilon-2\epsilon^2)}$$

(10−2ε)



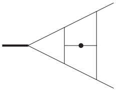
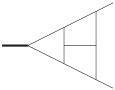
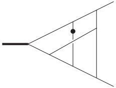
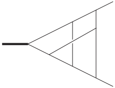
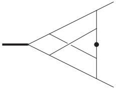
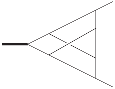
$$+ \frac{1}{\epsilon}\,\frac{4(3-4\epsilon)(1-4\epsilon)}{(2-\epsilon)(1-\epsilon)(1-2\epsilon)(2-\epsilon-2\epsilon^2)}$$

(8−2ε)



Numerical performance [AvM, Schabinger (in prep)]

improvement wrt conventional basis:

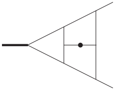
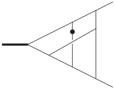
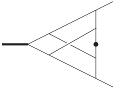
finite	time	rel. err.	conventional	time	rel. err.
$(6-2\epsilon)$ 	128 s	5.12×10^{-6}	$(4-2\epsilon)$ 	39094 s	9.91×10^{-4}
$(6-2\epsilon)$ 	192 s	2.68×10^{-6}	$(4-2\epsilon)$ 	19025 s	9.38×10^{-5}
$(6-2\epsilon)$ 	127 s	2.26×10^{-6}	$(4-2\epsilon)$ 	19586 s	1.07×10^{-4}

timings with Fiesta 4, ϵ expansion through to weight 6

Numerical performance

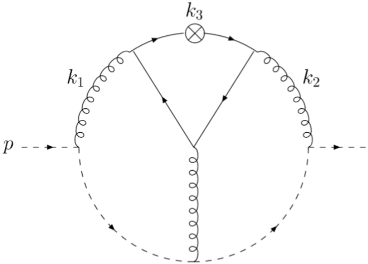
[AvM, Schabinger (in prep)]

ϵ expansions to high weights feasible:

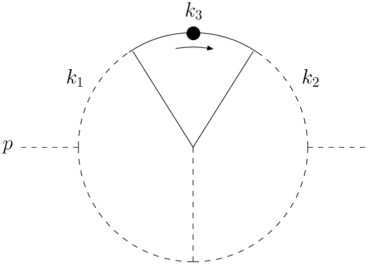
	weight 6		weight 8	
	time	rel. err.	time	rel. err.
$(6-2\epsilon)$ 	128 s	5.12×10^{-6}	491 s	2.22×10^{-5}
$(6-2\epsilon)$ 	192 s	2.68×10^{-6}	761 s	5.84×10^{-6}
$(6-2\epsilon)$ 	127 s	2.26×10^{-6}	485 s	8.45×10^{-6}

timings with Fiesta 4

Example 2: linear propagators



$\longrightarrow$

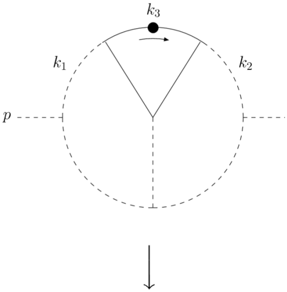


$$\downarrow \qquad\qquad \downarrow$$

$$(\Delta.k_3)^{N-1} \longrightarrow \frac{1}{1 - x\Delta.k_3}$$

[Ablinger, Behring, Blümlein, De Freitas, Hasselhuhn, AvM, Round, Schneider, Wißbrock '13-'16]

Example 2: linear propagators



$$\frac{1}{1 - x\Delta.k_3}$$

```
integralfamilies:

  - name: "B1a" # Benz and Ladder with 3 massive
                # propagators. Insertions 5,6,7.
    loop_momenta: [k1,k2,k3]
    propagators:
      - [ "k1",      0  ]
      - [ "k1-p",    0  ]
      - [ "k2",      0  ]
      - [ "k2-p",    0  ]
      - [ "k3",      m2 ]
      - [ "k1-k3",   m2 ]
      - [ "k2-k3",   m2 ]
      - [ "k1-k2",   0  ]
      - [ "k3-p",    m2 ]
      - { bilinear: [ ["-q", k3-k1], -m2 ] }
      - { bilinear: [ ["-q", k3   ], -m2 ] }
      - { bilinear: [ ["-q", k3-k2], -m2 ] }
    permutation_symmetries:
      - [[1, 3], [2, 4], [6, 7], [10, 12]] # k1<->k2
```

Example 3: phase space integrals, eikonal propagators

- treat phase space integration as loop integrals with additional on-shell constraint

$$\int dPS_2 = i^{-2} \int \frac{d^d p_1}{(2\pi)^d} \frac{d^d p_2}{(2\pi)^d} (2\pi)^d \delta^{(d)}(q - p_1 - p_2) \frac{1}{D_1} \frac{1}{D_2}$$

  where on-shell constraint is incorporated via cut propagators

$$D_j = 2\pi i \theta(p_j^0) \delta(p_j^2) = \frac{1}{p_j^2 - i0 p_j^0} - \frac{1}{p_j^2 + i0}$$

  allows to use standard IBP reductions [Anastasiou, Melnikov '02]; [Anastasiou, Dixon, Melnikov, Petriello '03,'03]; [Gehrmann-de Ridder, Gehrmann, Heirich '03]

- linear propagators encountered e.g. in effective theories

$$\frac{1}{2kp}$$

show setup in Reduze

IBP reductions from finite field samples

A novel approach to IBPs [AvM, Schabinger '14]

1. finite field sampling
   - set variables to integer numbers
   - consider coefficients modulo a prime field $\mathbb{Z}_p$
2. solve finite field system
3. reconstruct rational solution from many such samples

IBP reductions from finite field samples

A novel approach to IBPs [AvM, Schabinger '14]

1. finite field sampling
   - set variables to integer numbers
   - consider coefficients modulo a prime field $\mathbb{Z}_p$
2. solve finite field system
3. reconstruct rational solution from many such samples

finite field techniques:

- no intermediate expression swell by construction
- early discard of redundant and auxiliary quantities
- big potential for parallelisation

established in math literature, becomes popular in physics:

- dense solver: [Kauers]
- filtering: `Ice` [Kant '13]
- supersymmetric integrand construction: [Bern et al]
- tensor reduction: [Heller]
- QCD integrand construction: [Peraro '16]

A fast univariate solver

rational solver: reduce matrix $I_{\mathbb{Q}}$ of rational numbers

$$
\begin{array}{ccccccccc}
I_{\mathbb{Q}} & \xrightarrow[\text{image}]{\text{homomorphic}} & I_{\mathbb{Z}_{p_1}} & \xrightarrow[\text{reduction}]{\text{Gauss row}} & O_{\mathbb{Z}_{p_1}} & \xrightarrow[\text{remaindering}]{\text{Chinese}} & O_{\mathbb{Z}_{p_1 \cdot p_2 \cdot p_3 \cdots}} & \xrightarrow[\text{reconstruction}]{\text{rational}} & O_{\mathbb{Q}} \\
 & \longrightarrow & I_{\mathbb{Z}_{p_2}} & \longrightarrow & O_{\mathbb{Z}_{p_2}} & \longrightarrow & & & \\
 & \longrightarrow & I_{\mathbb{Z}_{p_3}} & \longrightarrow & O_{\mathbb{Z}_{p_3}} & \longrightarrow & & & \\
 & \vdots & \vdots & \vdots & \vdots & & & &
\end{array}
$$

A fast univariate solver

rational solver: reduce matrix $I_{\mathbb{Q}}$ of rational numbers

$$
\begin{array}{ccccccccc}
I_{\mathbb{Q}} & \xrightarrow[\text{image}]{\text{homomorphic}} & I_{\mathbb{Z}_{p_1}} & \xrightarrow[\text{reduction}]{\text{Gauss row}} & O_{\mathbb{Z}_{p_1}} & \xrightarrow[\text{remaindering}]{\text{Chinese}} & O_{\mathbb{Z}_{p_1 \cdot p_2 \cdot p_3 \cdots}} & \xrightarrow[\text{reconstruction}]{\text{rational}} & O_{\mathbb{Q}} \\
 & \longrightarrow & I_{\mathbb{Z}_{p_2}} & \longrightarrow & O_{\mathbb{Z}_{p_2}} & \longrightarrow & & & \\
 & \longrightarrow & I_{\mathbb{Z}_{p_3}} & \longrightarrow & O_{\mathbb{Z}_{p_3}} & \longrightarrow & & & \\
 & \vdots & \vdots & \vdots & \vdots & & & &
\end{array}
$$

univariate solver: reduce matrix $I_{\mathbb{Q}[x]}$ of rational functions in x

$$
\begin{array}{ccccccccc}
I_{\mathbb{Q}[x]} & \xrightarrow[\text{img.}]{\text{hom.}} & I_{\mathbb{Z}_{p_1}}[x] & \xrightarrow[\text{(see below)}]{\text{aux solver}} & O_{\mathbb{Z}_{p_1}[x]} & \xrightarrow[\text{remaind.}]{\text{Chinese}} & O_{\mathbb{Z}_{p_1 \cdot p_2 \cdot p_3 \cdots [x]}} & \xrightarrow[\text{rec.}]{\text{rat.}} & O_{\mathbb{Q}[x]} \\
 & \longrightarrow & I_{\mathbb{Z}_{p_2}[x]} & \longrightarrow & O_{\mathbb{Z}_{p_2}[x]} & \longrightarrow & & & \\
 & \longrightarrow & I_{\mathbb{Z}_{p_3}[x]} & \longrightarrow & O_{\mathbb{Z}_{p_3}[x]} & \longrightarrow & & & \\
 & \vdots & \vdots & \vdots & \vdots & & & &
\end{array}
$$

A fast univariate solver

rational solver: reduce matrix $I_{\mathbb{Q}}$ of rational numbers

$$
\begin{array}{ccccccccc}
I_{\mathbb{Q}} & \xrightarrow[\text{image}]{\text{homomorphic}} & I_{\mathbb{Z}_{p_1}} & \xrightarrow[\text{reduction}]{\text{Gauss row}} & O_{\mathbb{Z}_{p_1}} & \xrightarrow[\text{remaindering}]{\text{Chinese}} & O_{\mathbb{Z}_{p_1 \cdot p_2 \cdot p_3 \cdots}} & \xrightarrow[\text{reconstruction}]{\text{rational}} & O_{\mathbb{Q}} \\
 & \longrightarrow & I_{\mathbb{Z}_{p_2}} & \longrightarrow & O_{\mathbb{Z}_{p_2}} & \longrightarrow & & & \\
 & \longrightarrow & I_{\mathbb{Z}_{p_3}} & \longrightarrow & O_{\mathbb{Z}_{p_3}} & \longrightarrow & & & \\
 & \vdots & \vdots & \vdots & \vdots & & & &
\end{array}
$$

univariate solver: reduce matrix $I_{\mathbb{Q}[x]}$ of rational functions in x

$$
\begin{array}{ccccccccc}
I_{\mathbb{Q}[x]} & \xrightarrow[\text{img.}]{\text{hom.}} & I_{\mathbb{Z}_{p_1}}[x] & \xrightarrow[\text{(see below)}]{\text{aux solver}} & O_{\mathbb{Z}_{p_1}[x]} & \xrightarrow[\text{remaind.}]{\text{Chinese}} & O_{\mathbb{Z}_{p_1 \cdot p_2 \cdot p_3 \cdots [x]}} & \xrightarrow[\text{rec.}]{\text{rat.}} & O_{\mathbb{Q}[x]} \\
 & \longrightarrow & I_{\mathbb{Z}_{p_2[x]}} & \longrightarrow & O_{\mathbb{Z}_{p_2[x]}} & \longrightarrow & & & \\
 & \longrightarrow & I_{\mathbb{Z}_{p_3[x]}} & \longrightarrow & O_{\mathbb{Z}_{p_3[x]}} & \longrightarrow & & & \\
 & \vdots & \vdots & \vdots & \vdots & & & &
\end{array}
$$

aux solver: reduce matrix $I_{\mathbb{Z}_p[x]}$ of polynomials in x with finite field coefficients

$$
\begin{array}{ccccccccc}
I_{\mathbb{Z}_p[x]} & \xrightarrow[\text{by number } x_i]{\text{sample } x} & I_{\mathbb{Z}_p, x_1} & \xrightarrow[\text{reduction}]{\text{row}} & O_{\mathbb{Z}_p, x_1} & \xrightarrow[\text{interpolation}]{\text{polynomial}} & O_{\mathbb{Z}_p[x]} & \xrightarrow[\text{reconstruction}]{\text{rational function}} & O_{\mathbb{Z}_p[x]} \\
 & \longrightarrow & I_{\mathbb{Z}_p, x_2} & \longrightarrow & O_{\mathbb{Z}_p, x_2} & \longrightarrow & & & \\
 & \longrightarrow & I_{\mathbb{Z}_p, x3} & \longrightarrow & O_{\mathbb{Z}_p, x3} & \longrightarrow & & & \\
 & \vdots & \vdots & \vdots & \vdots & & & &
\end{array}
$$

note: massively parallisable

```
 ____  __  __ _  ____  ____  ____
(  __)(  )(  ( \(  _ \(  __)(    \
 ) _)  )( /    / )   / ) _)  ) D (
(__)  (__)\_)__)(__\_)(____)(____/

Package: finred
Author:  Andreas v. Manteuffel
```

features:

- C++11 implementation for univariate sparse matrices
- employs `flint` library
- parallelisation: SIMD, threads, MPI, batch
- equation filtering: eliminate redundant rows
- plus lots of IBP specific features
- much faster than Reduze 2

First QCD result @ 4-loops for gluons

[AvM, Schabinger (in prep.)]

Bare gluon form factor

$$\mathcal{F}_4^g|_{N_f^3} = C_F\left[-\frac{2}{3\epsilon^3}+\frac{1}{\epsilon^2}\left(\frac{32}{3}\zeta_3-\frac{145}{9}\right)+\frac{1}{\epsilon}\left(\frac{352}{45}\zeta_2^2+\frac{1040}{9}\zeta_3+\frac{68}{9}\zeta_2-\frac{10003}{54}\right)\right.$$
$$\left.+\frac{4288}{27}\zeta_5-64\zeta_3\zeta_2+\frac{2288}{27}\zeta_2^2+\frac{24812}{27}\zeta_3+\frac{3074}{27}\zeta_2-\frac{508069}{324}+\mathcal{O}(\epsilon)\right]$$
$$+C_A\left[\frac{1}{27\epsilon^5}+\frac{5}{27\epsilon^4}+\frac{1}{\epsilon^3}\left(-\frac{14}{27}\zeta_2-\frac{55}{81}\right)+\frac{1}{\epsilon^2}\left(-\frac{586}{81}\zeta_3-\frac{70}{27}\zeta_2-\frac{24167}{1458}\right)\right.$$
$$+\frac{1}{\epsilon}\left(-\frac{802}{135}\zeta_2^2-\frac{5450}{81}\zeta_3-\frac{262}{81}\zeta_2-\frac{465631}{2916}\right)-\frac{14474}{135}\zeta_5+\frac{4556}{81}\zeta_3\zeta_2$$
$$\left.-\frac{1418}{27}\zeta_2^2-\frac{99890}{243}\zeta_3+\frac{38489}{729}\zeta_2-\frac{20832641}{17496}+\mathcal{O}(\epsilon)\right]$$

gluon cusp anomalous dimension:

$$\Gamma_4^g|_{N_f^3} = C_A\left[\frac{64}{27}\zeta_3-\frac{32}{81}\right]$$

- respects Casimir scaling
- non-planar C_F pieces do not contribute to $\Gamma_4^g|_{N_f^3}$

Conclusions

Reduze 2

- open source IBP reduction program
- advanced shift finders
- new features in Reduze 2.1:
  - bilinear propagators
  - cut propagators
  - basis of finite integrals
  - dimensional recurrences

Finred:

- finite field sampling + rational reconstruction
- much faster than Reduze