

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 :

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

(integrals indexed by families with tuples k of propagator powers)

- 3 reduce scalar loop integrals to master integrals M_l

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

(only a small number of masters)

- 4 compute master integrals (or look up in the literature)

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

(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

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

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:

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

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

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

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

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 $a = 1$:

$$\text{tadpole} = \frac{(d-2)}{2m^2} \times \text{bubble}$$

Laporta's algorithm:

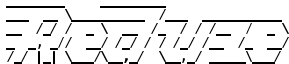
- 1 index integrals by propagator exponents: $I(a_1, \dots, a_N)$
- 2 define **ordering** (e.g. fewer denominators means simpler)
- 3 generate IBPs for explicit values a_1, \dots, a_N
- 4 results in **linear system** of equations
- 5 solve **linear system** of equations

REALISTIC EXAMPLE

in reality we might be after something like this

```
File Bearbeiten Ansicht Verlauf Lesezeichen Einstellungen Hilfe
tt2hA 7 463 7 3 1 1 1 1 -3 0 1 1 1
1
tt2hA 7 463 7 2 1 1 1 1 -2 0 1 1 1
-/4+128+2*ns+d.7*ns.8*nd*(.|-)|(.|2016+10*ns+4*d+102*ns+2*d+2*t+2.240*ns+d+t*3.624*ns+tt.240*ns+3*d+tt+1056*ns+d+t*2.18*ns+2
+d+2*t+3*ns+4*d+2*t.144*ns+d+2.152*ns+d+240*ns+tt+3.1200*ns+2.102*ns+3*d+tt+2.1008*ns+d+tt.288*ns+3.720*ns+tt+2.240*ns+2*d+2*t.3*ns+4*d+2
+624*ns+2*t+2.15*ns+3*d+560*ns+tt+2+1410*ns+2*d+tt.910*ns+2*d.152*d+tt+3*ns+560*ns+3*tt.168*ns+2*tt+3+1104*ns.15*ns+3*d+2*tt+2+230*ns+d+2*t+2
72*tt+3*ns+3*d+2*tt+112*ns+2*d+tt*3.240*ns+2*d+tt+2.10*ns+4*d+tt+162*ns+2*d+576*d.168*ns+3*tt+2+48*ns+d+2*tt*3.612*ns+d+tt+2.3360*tt+2
-186*ns+2*tt+138*ns+3*d+4704*tt+1344*d+tt*(|-4*ns+d+tt+2*ns+d+2*ns+2*ns+d+tt+2*ns+tt+2*ns+2*d+2*ns.4*tt+2.2*ns+2*tt+8*tt*(|-|)
tt2hA 7 463 7 1 1 1 1 1 1 -1 0 1 1 1
1/4*(|-4*ns)*(|-|)|(.|3072+13*ns+2*d+2*tt+2+1120*ns+tt+18*ns+3*d+tt+140*ns+d+tt+2.216*ns+d+2+1340*ns+d+432*ns+2.856*ns+d+tt.176*ns+
2*tt+2.384*ns+2*tt+2.168*ns+2*tt+2.22*ns+2*tt+2.22*ns+2*tt+2.127*ns+2*tt+192*ns+144*ns+d+tt.142*ns+3*ns+2*tt+2+2*ns+2*d+2*ns
64*d.94*ns+2*d+tt+2.384*tt+2.608*ns+2*tt.22*ns+3*d+2112*tt.576*d+tt*(|-4*ns+d-2*ns)*(|-|)|(.|7+2*d)*(|-|)
tt2hA 7 463 7 0 1 1 1 1 0 0 1 1 1
-1/4*(|-4*ns)*(|-|)|(.|1584+16*ns+2*d+2*tt+2.28*ns+d+tt+3+1168*ns+tt+26*ns+3*d+tt+248*ns+d+tt+2.3*ns+2*d+2*tt+3.108*ns+d+2+768*ns+
d+48*ns+tt+200*ns+2.8*ns+3*d+tt+2.748*ns+d+tt.392*ns+tt+2.17*ns+2*d+2*tt+320*ns+2*tt+2+4*ns+3*d+2+144*d+tt+2+214*ns+2*d+tt.2*ns+2*d.16*d+tt
3.56*ns+2*tt+3.1272*ns+2*ns+3*d+2*tt+2+108*ns+d+2*tt+80*tt+3.7*ns+3*d+2*tt+26*ns+2*d+tt*3.36*ns+d+2*tt+2.12*ns+2*d+2+432*d+ns+d+2*tt*3.144*
*ns+2*tt+2.624*tt+2.536*ns+2*tt.12*ns+3*d+1680*tt.432*d+tt*(|-4*ns+d-2*ns)*(|-|)|(.|7+2*d)*(|-|)
tt2hA 6 207 6 0 1 1 1 1 0 0 1 1 0
1/8*(192+12*ns+2*d+2*tt+2+128*ns+tt.56*ns+d+tt+2*ns+2*d+3*tt+128*d+2.88*ns+d.8*2*d+3*tt+280*ns+d+tt+324*ns+tt+2.4*ns+2*d+2*tt
+48*ns+2*tt+2.32*d+tt+2.16*ns+2*d+tt.12*ns+2*d+96*ns.24*ns+d+2*tt+12*ns+d+2*tt+2+8*ns+2*d+2.32*d.44*ns+2*d+d+tt+192*tt+2.384*tt+ns+2*ns+3*64*d
*tt*(|-4*ns)*(|-|)|(.|4*ns+d-2*ns)*(|-|)|(.|2*d)*(|-|)|(.|7+2*d)*(|-|)
tt2hA 5 460 5 1 -1 0 1 0 0 1 1 1
(|+ns+*)*(2)|(.|3840+102*ns+2*d+4+12896*ns+4*d.2292*ns+2*d+2*tt+2*ns+2*d*5.15360*ns+tt.276*ns+3*ns+2688*d+2*tt+80*ns+d+4*tt
+1440*ns+3*d+tt.192*ns+d+tt+2.864*ns+2*d+3*tt+144*d+3*tt+2+4504*ns+d+2.288*d+3*tt.14784*ns+d+14*ns+3*d+5*tt.38*ns+3*ns+51280*ns+2+702*ns+3
*d+41512*ns+2*d+3*tt+2+16976*ns+d+tt+1904*ns+4*d+3+22240*ns+3.278*ns+3*d+4*tt+960*ns+tt+2.1344*d+2*tt+2.40*ns+d+4*tt+2+5600*ns+2*d+2*tt.712
4*ns+4*d+2.3360*ns+2*tt+2*ns+2*d+5*tt+2+20072*ns+3*d+2+4032*d+tt+2+300*ns+d+3*tt+144*d+3.15456*ns+2*d+tt+2164*ns+3*tt.8160*ns+4.1644
8*ns+2*d.47*ns+2*d+4*tt+2.240*ns+4*d+4.5376*ns+3*d+3.10080*ns+3*tt+14400*ns+13*ns+d+5.1344*d+2.3600*ns+d+2*tt.8128*ns+3*tt+2.584*ns+d
+2*tt+38*ns+2*d+2*ns+2*d+2+4032*d.40*ns+d+4+5176*ns+2*d+2+2.40*d+3*ns.3840*tt.2*ns+2*d+5*tt+13440*ns+2*tt.35152*ns+3*tt+7880*
+1072*ns+2*d+3.8654*d+tt*(|-|)|(.|4*d)*(2)|(.|4*ns+d-2*ns)*(|-|)|(.|2*d)*(|-|)|(.|7+2*d)*(|-|)
tt2hA 5 460 5 0 0 1 1 0 0 1 1 1
-1/2*(|-39184+165*ns+2*d+4+112848*ns+4*d+29932*ns+2*d+2*tt+2.3584*ns+d+tt*3.21*ns+2*d+5+25152*ns+tt.699*ns+3*d+4*tt+2+1612*ns
+d+3+8448*d+2*tt+20*ns+d+4*tt.15846*ns+3*tt+18656*ns+d+tt+41376*ns+d+tt+11320*ns+2*d+3*tt.4796*ns+2*d+2*tt+3*ns+2*tt+2.24516*ns+4*d
+2*tt.10056*ns+d+2+37*ns+3*d+5*tt+2.4*ns+d+4*tt+3.576*d+3*tt+13088*d+tt.166*ns+3*d+5*tt+193*ns+3*d+5+3264*ns+tt+2.28000*ns+4*tt+3
2208*ns+3*tt+2.10740*ns+2*tt.3625*ns+3*tt.6892*ns+2*d+3*tt+2.50880*ns+d+tt.17800*ns+4*d+3.90112*ns+3+3058*ns+3*d+4*tt.38552*ns+2.42
24*d+2*tt+2.4*ns+d+4*tt+2+6940*ns+4*d+3*tt.5648*ns+2*d+2*tt+64572*ns+4*d+2+1248*ns+2*d+3*tt+30116*ns+5*tt+3+46752*ns+d+2*tt.35*ns+2*d+5*tt
+2.417*ns+4*d+4.97140*ns+3*d+2+19584*d+tt+2.1534*ns+2*d+4*tt+3.12544*ns+5*23*ns+5*d+5*tt+1668*ns+d+3*tt+2+288*d+3.963*ns+4*d+4*tt+2.108*ns+2
*d+2*tt.23548*ns+3*d+3*tt+74720*ns+4+37056*ns+2*d+759*ns+2*d+4*tt+2+2407*ns+4*d+4+26712*ns+3*d+3+108800*ns+3*tt.6048*ns+3*tt+3+10176*ns+7*ns
+2*d+5*tt+3.131*ns+4*d+5.18572*ns+3*d+2*tt+2+53*ns+4*d+5*tt.4224*d+2.124*ns+d+3*tt+3+23272*ns+d+2*tt+67964*ns+3*tt+2+8736*ns+3*tt+3.14
450*ns+d+2*tt+2+24218*ns+4*tt+1027*ns+2*d+4*tt.1164*ns+2*d+2+19584*d.21504*ns+3*tt+2+1240*ns+d+2*tt+3.12*ns+d+4.60929*ns+2*tt+2*tt+2.3156*ns
+d+3*tt.29184*ns+2+1172*ns+3*d+3*tt+2+40*ns+2*d+ns.80956*ns+2*tt+161136*ns+3*d+58368*tt.1164*ns+2*d+3+30168*d+tt*(|-+ns+tt*(|-2)|(.|4*ns)
*tt*(|-|)|(.|4*d)*(2)|(.|4*ns+d-2*ns)*(|-|)|(.|2*d)*(|-|)|(.|7+2*d)*(|-|)
tt2hA 5 393 5 1 -1 1 1 0 0 1 0 1
-1/16*(|-5919840+260174*ns+2*d+4+13486992*ns+4*d+13414712*ns+2*d+2*tt+2+18082112*ns+d+tt+36912*ns+d+6*tt+3.108*d+4*tt+5+1
1152*ns+d+2*tt+5.109192*ns+2*d+5.6467604*ns+tt.326880*ns+3*d+4*tt+2.40106*ns+3*d+5*tt+3.3303428*ns+3*ns+16099176*ns+d+2*tt.134100*d+4*tt+4
+679320*ns+d+3.2716360*ns+3*tt.2716360*d+3*tt+2.4163880*d+3*tt+2.2956104*ns+d+2*tt+2.6435896*d+tt.19307872*ns+3*d+tt+888*ns+2*d+6*tt+1420544
*ns+4+26336*ns+3*d+6*tt+4+35200*ns+4*tt+2.7502156*ns+d+tt+2.2215200*ns+5*d+3*tt+1807132*d+2*d+3*tt.10080*ns+5.10388*ns+d+6*tt+2.158104
8*ns+2*d+2*tt+3*ns+25840*d+3*tt+2.28973536*ns+4*d+2*tt+14167080*ns+d+2+407002*ns+3*d+5*tt+2.1256988*ns+2*d+4*tt+1572736*ns+6*d+2+420*ns+2
*d+6*tt+4.5879520*tt+4+6912*d+4*tt+3+343082*ns+3*d+4*tt+3.45111*ns+d+5*tt+576658*ns+d+3*tt+4.4170900*d+3*tt+1286332*ns+3*tt+4.1926912*
*ns+d.28451360*ns+d.64248*ns+3*d+5*tt+6610368*ns+4*d+2*tt+2+16065936*d+2*tt+3+275500*ns+3*d+5.17651040*ns+3+13584*d+5.20846160*
ns+2+116118*ns+d+5*tt+2.20716*ns+3*d+6*tt+2.15644160*ns+4*tt+693360*ns+2*tt+4.8960*ns+2*d+4+36908192*ns+3*d+tt+2.9962144*ns+5*tt+2.330
0*ns+3*tt+5.69124*d+6*tt.167128*ns+4.10420168*ns+2*d+3*tt+2.41910168*ns+d+2*tt.7981000*ns+3*tt+4+7662780*ns+3*tt+3.1087
58*ns+d+5*tt+3+4507014*ns+3*d+4*tt+61980960*ns+2.134532*d+4.24132144*d+2*tt+2+50700*ns+d+4*tt+2+1268320*d+4*tt+3*tt+78738*ns+2*d+4*tt
+4+1686024*ns+2*d+2*tt+15253568*ns+4*d+2+386880*ns+5*tt+5273600*ns+5*d+2*tt+6830276*ns+2*d+3*tt+3+6912*d+5.16052*ns+3*ns+6*tt+1932*ns+3*d
```

⇒ need some computer program



Reduze 2 [AvM, C. Studerus]

arXiv:1201.4330, HepForge

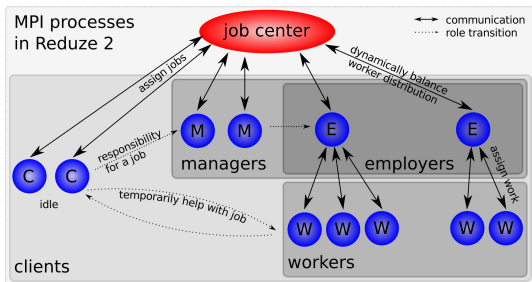
based on Reduze 1 [Studerus]

uses GiNaC [Bauer, Frink, Kreckel]

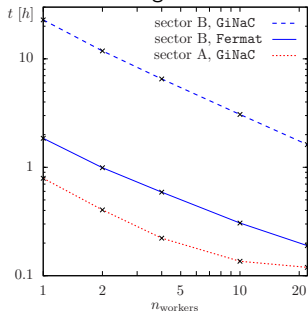
and Fermat [Lewis]

- 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^3 LO Higgs and DY [Li,AvM,Schabinger,Zhu '14,'14]
 $e^+e^- \rightarrow t\bar{t}$ 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)
 - ▶ ...

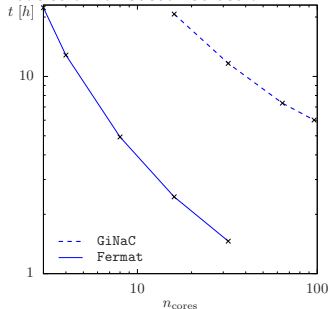
REDUCE 2: DISTRIBUTED LAPORTA ALGORITHM



reduction of single sector:



reduction of sector selection:



SHIFT RELATIONS

- assigning loop momenta to propagators **not unique**:

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

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

$$D_1 := k^2$$

$$D_2 := (k+p)^2$$

$$I_{a_1, a_2} := \int d^d k \frac{1}{D_1^{a_1} D_2^{a_2}}$$

shift relation:

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

- also needed to map diagrams to sectors of families

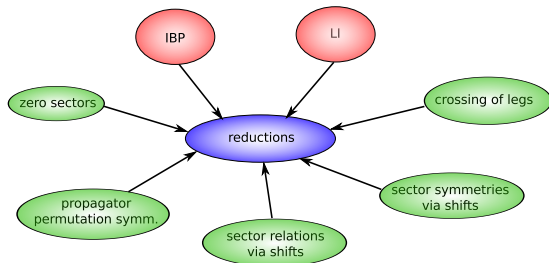
SHIFT RELATIONS

- consider trafo of loop momenta

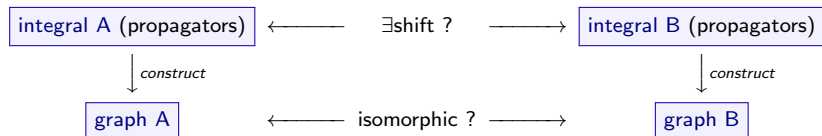
$$k_i \rightarrow \sum_{j=1}^l M_{ij} k_j + \sum_{j=1}^m N_{ij} p_j$$

with $|\det M| = 1$

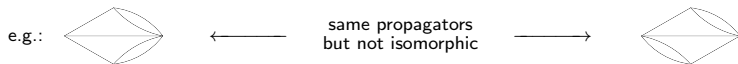
- systematic **shift finding**:
 - ▶ pure combinatoric
 - ▶ graph/matroid based
- Reduze 2 handles 3 types of shifts:
 - ▶ **permutation symmetries**: for all sectors of a family, very efficient
 - ▶ **sector relations**: eliminate complete sectors
 - ▶ **sector symmetries**: relates integrals of same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



REDUZE 2: SECTORS, GRAPHS AND MATROIDS



- **problem:** graphs not unique !



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

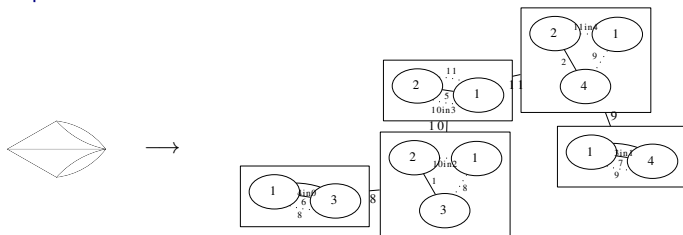
based on: **first Symanzik polynomials \mathcal{U} isomorphic \Leftrightarrow graphs isomorphic up to twists**

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:



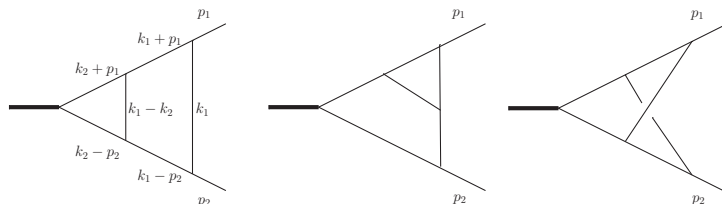
tree of triconnected components

(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:



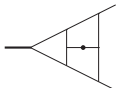
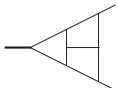
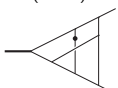
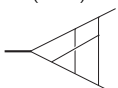
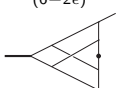
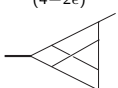
on-line demo of Reduze:

- 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]

$$\begin{aligned}
 & \text{Diagram } (4-2\epsilon) = \frac{1}{\epsilon^2} \frac{1}{(1-\epsilon)^2} \text{Diagram } (6-2\epsilon), \\
 & \text{Diagram } (4-2\epsilon) = \frac{1}{\epsilon} \frac{-4}{(2-\epsilon)^2(1-\epsilon)^2(1-2\epsilon)} \text{Diagram } (8-2\epsilon), \\
 & \text{Diagram } (4-2\epsilon) = \frac{1}{\epsilon^2} \frac{16(3-2\epsilon)(2-3\epsilon)}{(3-\epsilon)^2(2-\epsilon)^2(1-\epsilon)^3(1+2\epsilon)} \text{Diagram } (10-2\epsilon), \\
 & \text{Diagram } (4-2\epsilon) = \frac{1}{\epsilon^4} \frac{-4(2-3\epsilon)(14-81\epsilon+115\epsilon^2+14\epsilon^3-132\epsilon^4+72\epsilon^5)}{(2-\epsilon)^2(1-\epsilon)^2(1-2\epsilon)^2(2-\epsilon-2\epsilon^2)} \text{Diagram } (8-2\epsilon) \\
 & \quad + \frac{1}{\epsilon^4} \frac{-16(1+\epsilon)(3-2\epsilon)(2-3\epsilon)(10-61\epsilon+102\epsilon^2-44\epsilon^3-8\epsilon^4)}{(3-\epsilon)^2(2-\epsilon)^2(1-\epsilon)^3(1-2\epsilon)(1+2\epsilon)(2-\epsilon-2\epsilon^2)} \text{Diagram } (10-2\epsilon) \\
 & \quad + \frac{1}{\epsilon} \frac{4(3-4\epsilon)(1-4\epsilon)}{(2-\epsilon)(1-\epsilon)(1-2\epsilon)(2-\epsilon-2\epsilon^2)} \text{Diagram } (8-2\epsilon)
 \end{aligned}$$

improvement wrt conventional basis:

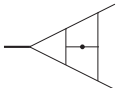
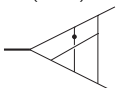
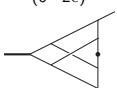
finite	time	rel. err.	conventional	time	rel. err.
$(6-2\epsilon)$ 	128 s	5.12×10^{-6}	$(4-2\epsilon)$ 	39094 s	9.91×10^{-4}
$(6-2\epsilon)$ 	192 s	2.68×10^{-6}	$(4-2\epsilon)$ 	19025 s	9.38×10^{-5}
$(6-2\epsilon)$ 	127 s	2.26×10^{-6}	$(4-2\epsilon)$ 	19586 s	1.07×10^{-4}

timings with Fiesta 4, ϵ expansion through to weight 6

NUMERICAL PERFORMANCE

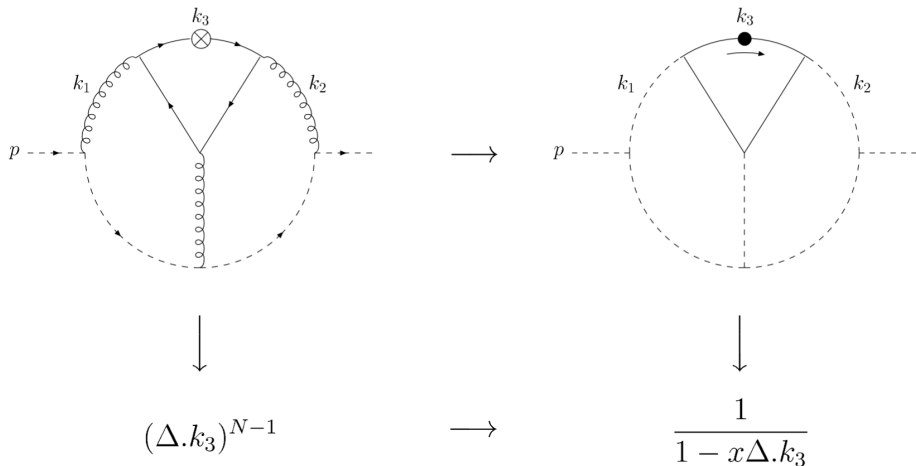
[AvM, Schabinger (in prep)]

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}

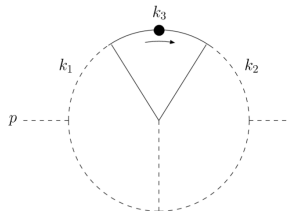
timings with Fiesta 4

EXAMPLE 2: LINEAR PROPAGATORS



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

EXAMPLE 2: LINEAR PROPAGATORS



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

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

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:

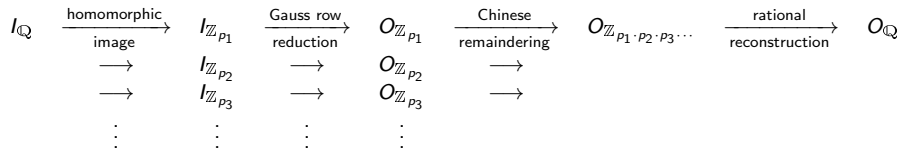
- no intermediate expression swell by construction
- early discard of redundant and auxiliary quantities
- big potential for parallelisation

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



A FAST UNIVARIATE SOLVER

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

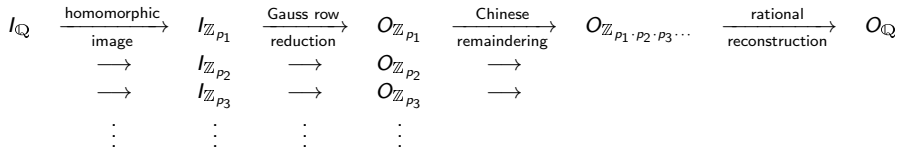
$$\begin{array}{ccccccc}
 I_{\mathbb{Q}} & \xrightarrow[\text{image}]{\text{homomorphic}} & I_{\mathbb{Z}_{p_1}} & \xrightarrow[\text{reduction}]{\text{Gauss row}} & O_{\mathbb{Z}_{p_1}} & \xrightarrow[\text{remaindering}]{\text{Chinese}} & O_{\mathbb{Z}_{p_1 \cdot p_2 \cdot p_3 \cdots}} & \xrightarrow[\text{reconstruction}]{\text{rational}}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

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

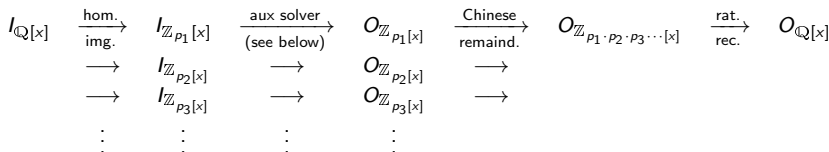
$$\begin{array}{ccccccc}
 I_{\mathbb{Q}[x]} & \xrightarrow[\text{img.}]{\text{hom.}} & I_{\mathbb{Z}_{p_1}[x]} & \xrightarrow[\text{(see below)}]{\text{aux solver}} & O_{\mathbb{Z}_{p_1}[x]} & \xrightarrow[\text{remaind.}]{\text{Chinese}} & O_{\mathbb{Z}_{p_1 \cdot p_2 \cdot p_3 \cdots}[x]} & \xrightarrow[\text{rec.}]{\text{rat.}}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

A FAST UNIVARIATE SOLVER

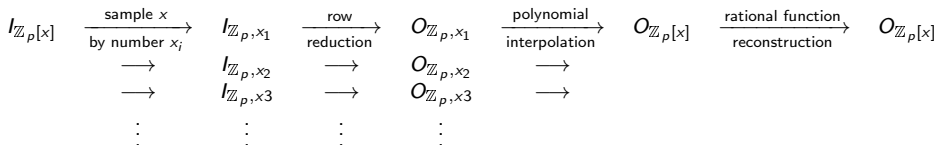
rational solver: reduce matrix $I_{\mathbb{Q}}$ of rational numbers



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



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



note: massively parallelisable

```
(  )(  )(  (  \(  \(  )(  \(  
)  )  )(  /    /  )  /  )  )  )  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

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

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