Towards five loop calculations in QCD

Ben Ruijl*, Franz Herzog, Takahiro Ueda, Jos Vermaseren, Andreas Vogt

*LCDS Leiden and Nikhef Amsterdam

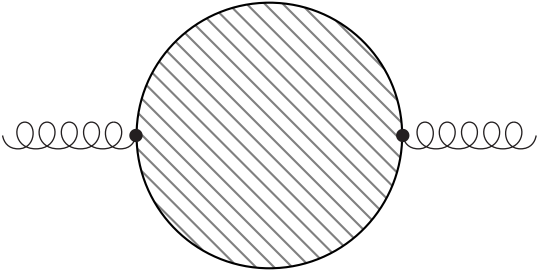
Dec 06, 2016

Universiteit Leiden
Leiden Centre of Data Science

Nikhef

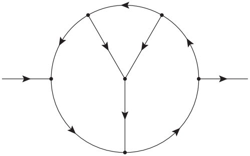
Forcer package

- We have built a program that calculates four-loop massless propagator diagrams
- Express diagrams as linear combinations of diagrams with fewer propagators



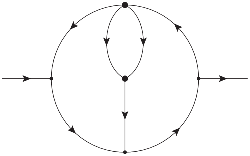
IBP identities

Through integration by parts (IBP) identities we find rules to remove lines:



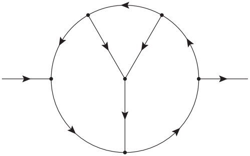
IBP identities

Through integration by parts (IBP) identities we find rules to remove lines:



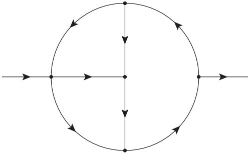
IBP identities

Through integration by parts (IBP) identities we find rules to remove lines:



IBP identities

Through integration by parts (IBP) identities we find rules to remove lines:



Identify substructures

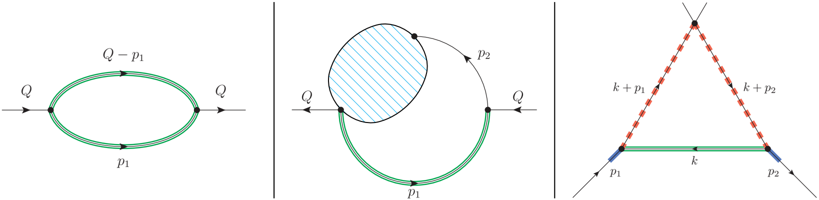


Figure: Guaranteed to remove a green or blue line

Triangle generalizations: diamonds [Ruijl,Ueda,Vermaseren '15]

Example reduction

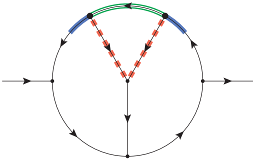


Figure: Reduction of the Benz diagram

Example reduction

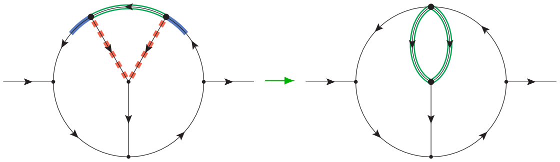


Figure: Reduction of the Benz diagram

Example reduction

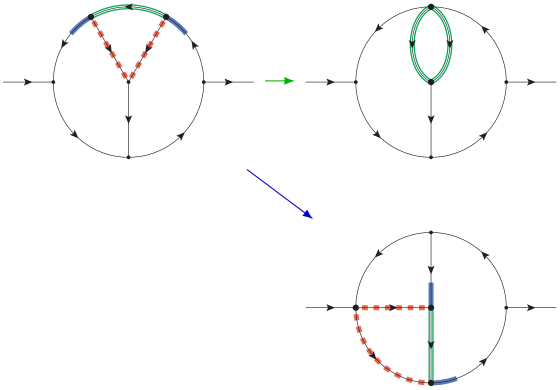


Figure: Reduction of the Benz diagram

Example reduction

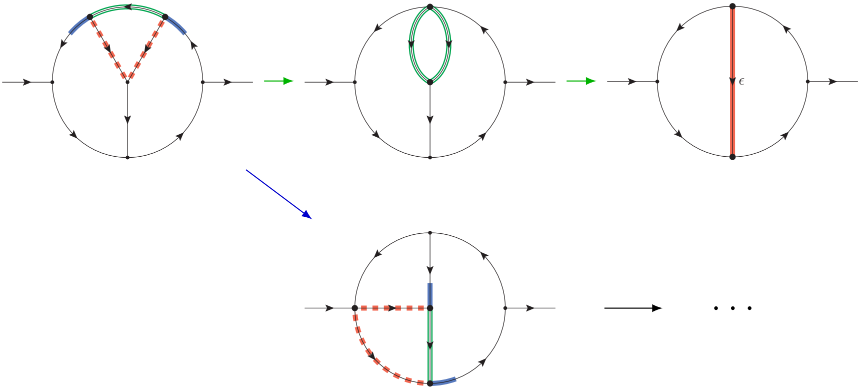


Figure: Reduction of the Benz diagram

3-loop reduction graph

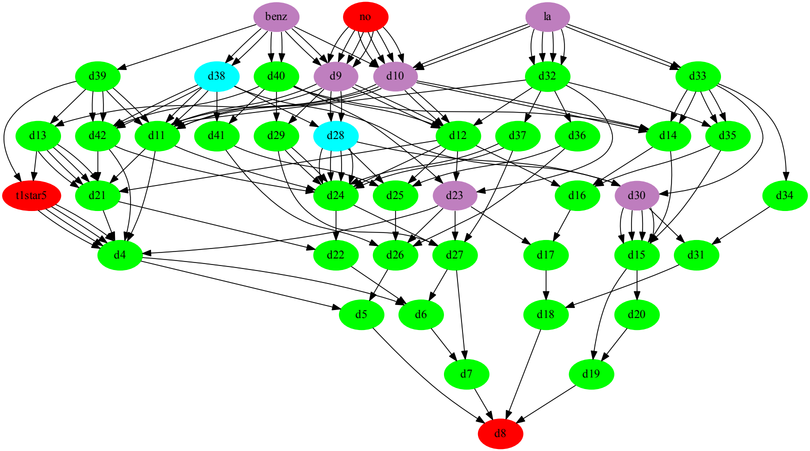


Figure: Reduction graph for 3 loop diagrams

4-loop reduction graph

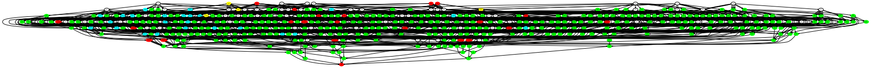


Figure: Reduction graph for 4 loop diagrams

4-loop reduction graph

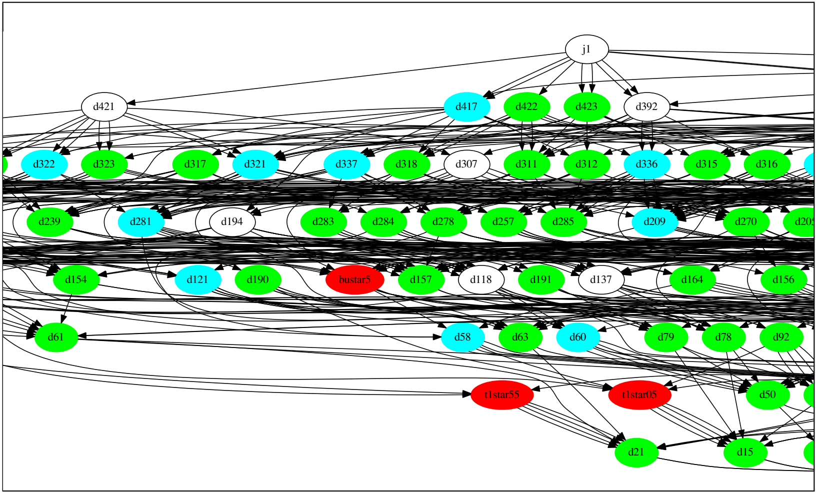
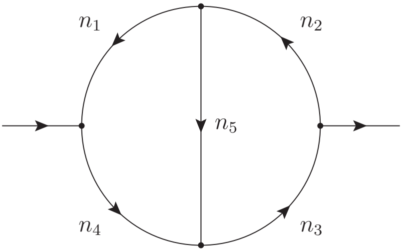


Figure: Part of reduction graph for 4 loops

Manual solutions

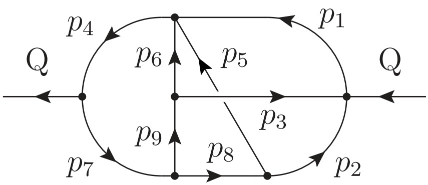
- 21 topologies do not have substructures
- Solve recursion relations parametrically
- Much faster than Laporta methods



$$I(n_1,n_2,n_3,n_4,n_5) = n_1 I(n_1-1,n_2,n_3,n_4,n_5) + n_2 I(n_1,n_2+1,n_3,n_4,n_5) + \ldots$$

$$I(n_1,n_2,n_3,n_4,n_5) = n_3 I(n_1,n_2,n_3-1,n_4,n_5) + n_3 I(n_1,n_2+1,n_3,n_4,n_5) + \ldots$$

Public enemy #1



- 9 propagators that should go to 1
- 5 irreducible dot products that should go to 0
- Took 3 months to find solution!

Reduction rule

```
id,ifmatch->bubu1,
    Z(n1?pos_,n2?pos_,n3?pos_,n4?pos_,n5?pos_,n6?pos_,n7?pos_,
    n8?pos_,n9?pos_,n10?neg0_,n11?neg0_,n12?neg0_,n13?neg0_,n14?neg_)
       = -rat(1,-2*ep-2*n1-n3-n6-n12-n14+4)*(
+Z(-1+n1,-1+n2,n3,n4,1+n5,n6,n7,n8,n9,n10,n11,n12,n13,1+n14)*rat(-n5,1)
+Z(-1+n1,1+n2,n3,n4,-1+n5,n6,n7,n8,n9,n10,n11,n12,n13,1+n14)*rat(n2,1)
+Z(-1+n1,1+n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,n12,n13,1+n14)*rat(-n2,1)
+Z(-1+n1,n2,1+n3,n4,n5,n6,n7,n8,-1+n9,n10,n11,n12,n13,1+n14)*rat(-n3,1)
+Z(-1+n1,n2,n3,n4,-1+n5,n6,n7,n8,n9,n10,n11,1+n12,n13,1+n14)*rat(-n12,1)
+Z(-1+n1,n2,n3,n4,1+n5,n6,n7,-1+n8,n9,n10,n11,n12,n13,1+n14)*rat(n5,1)
+Z(-1+n1,n2,n3,n4,n5,n6,-1+n7,n8,n9,n10,n11,n12,1+n13,1+n14)*rat(-n13,1)
+Z(-1+n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,1+n12,n13,1+n14)*rat(2*n12,1)
+Z(-1+n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,n12,1+n13,1+n14)*rat(n13,1)
+Z(-1+n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,n12,n13,2+n14)*rat(2*n14+2,1)
+Z(-1+n1,n2,n3,n4,n5,n6,n7,n8,-1+n9,n10,n11,n12,1+n13,1+n14)*rat(n13,1)
+Z(-1+n1,n2,n3,n4,n5,n6,n7,n8,n9,1+n10,-1+n11,n12,n13,1+n14)*rat(-n10,1)
+Z(-1+n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,1+n12,n13,n14)*rat(-n12,1)
+Z(-1+n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,1+n14)*rat(-n2+n5,1)
+Z(n1,-1+n2,-1+n3,n4,n5,n6,n7,n8,n9,n10,n11,1+n12,n13,1+n14)*rat(n12,1)
+Z(n1,-1+n2,-1+n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,2+n14)*rat(1+n14,1)
+Z(n1,-1+n2,1+n3,n4,n5,n6,n7,n8,-1+n9,n10,n11,n12,n13,1+n14)*rat(n3,1)
```

```
+Z(n1,-1+n2,n3,-1+n4,n5,n6,n7,n8,n9,n10,1+n11,n12,n13,1+n14)*rat(n11,1)
+Z(n1,-1+n2,n3,n4,-1+n5,n6,n7,n8,n9,n10,n11,1+n12,n13,1+n14)*rat(n12,1)
+Z(n1,-1+n2,n3,n4,n5,-1+n6,n7,n8,n9,n10,n11,n12,n13,2+n14)*rat(-n14-1,1)
+Z(n1,-1+n2,n3,n4,n5,n6,-1+n7,1+n8,n9,n10,n11,n12,n13,1+n14)*rat(2*n8,1)
+Z(n1,-1+n2,n3,n4,n5,n6,-1+n7,n8,n9,n10,1+n11,n12,n13,1+n14)*rat(-n11,1)
+Z(n1,-1+n2,n3,n4,n5,n6,-1+n7,n8,n9,n10,n11,n12,1+n13,1+n14)*rat(-n13,1)
+Z(n1,-1+n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,1+n12,n13,1+n14)*rat(-2*n12,1)
+Z(n1,-1+n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,n12,1+n13,1+n14)*rat(n13,1)
+Z(n1,-1+n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,n12,n13,2+n14)*rat(-2*n14-2,1)
+Z(n1,-1+n2,n3,n4,n5,n6,n7,1+n8,-1+n9,n10,n11,n12,n13,1+n14)*rat(-n8,1)
+Z(n1,-1+n2,n3,n4,n5,n6,n7,n8,-1+n9,n10,n11,1+n12,n13,1+n14)*rat(-2*n12,1)
+Z(n1,-1+n2,n3,n4,n5,n6,n7,n8,n9,1+n10,-1+n11,n12,n13,1+n14)*rat(n10,1)
+Z(n1,-1+n2,n3,n4,n5,n6,n7,n8,n9,1+n10,n11,n12,n13,1+n14)*rat(-2*n10,1)
+Z(n1,-1+n2,n3,n4,n5,n6,n7,n8,n9,n10,1+n11,n12,n13,1+n14)*rat(-n11,1)
+Z(n1,-1+n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,1+n12,n13,n14)*rat(-n12,1)
+Z(n1,1+n2,n3,n4,-1+n5,n6,n7,n8,n9,-1+n10,n11,n12,n13,1+n14)*rat(-n2,1)
+Z(n1,1+n2,n3,n4,-1+n5,n6,n7,n8,n9,n10,n11,n12,n13,1+n14)*rat(-n2,1)
+Z(n1,1+n2,n3,n4,n5,n6,n7,-1+n8,n9,-1+n10,n11,n12,n13,1+n14)*rat(2*n2,1)
+Z(n1,1+n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,n12,n13,1+n14)*rat(n2,1)
+Z(n1,n2,-1+n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,1+n12,n13,1+n14)*rat(-n12,1)
+Z(n1,n2,-1+n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,n12,n13,2+n14)*rat(-n14-1,1)
+Z(n1,n2,1+n3,n4,n5,n6,n7,n8,-1+n9,-1+n10,n11,n12,n13,1+n14)*rat(n3,1)
```

```
+Z(n1,n2,n3,n4,-1+n5,n6,-1+n7,n8,n9,n10,n11,n12,1+n13,1+n14)*rat(n13,1)
+Z(n1,n2,n3,n4,-1+n5,n6,n7,n8,-1+n9,n10,n11,n12,1+n13,1+n14)*rat(-n13,1)
+Z(n1,n2,n3,n4,-1+n5,n6,n7,n8,-1+n9,n10,n11,n12,n13,2+n14)*rat(1+n14,1)
+Z(n1,n2,n3,n4,-1+n5,n6,n7,n8,n9,-1+n10,n11,1+n12,n13,1+n14)*rat(n12,1)
+Z(n1,n2,n3,n4,-1+n5,n6,n7,n8,n9,n10,-1+n11,n12,1+n13,1+n14)*rat(n13,1)
+Z(n1,n2,n3,n4,-1+n5,n6,n7,n8,n9,n10,n11,1+n12,n13,1+n14)*rat(n12,1)
+Z(n1,n2,n3,n4,-1+n5,n6,n7,n8,n9,n10,n11,n12,1+n13,1+n14)*rat(n13,1)
+Z(n1,n2,n3,n4,-1+n5,n6,n7,n8,n9,n10,n11,n12,n13,1+n14)*rat(-n2+n8-n13,1)
+Z(n1,n2,n3,n4,n5,-1+n6,n7,-1+n8,n9,n10,n11,n12,n13,2+n14)*rat(1+n14,1)
+Z(n1,n2,n3,n4,n5,n6,-1+n7,-1+n8,n9,n10,1+n11,n12,n13,1+n14)*rat(n11,1)
+Z(n1,n2,n3,n4,n5,n6,-1+n7,-1+n8,n9,n10,n11,n12,1+n13,1+n14)*rat(-2*n13,1)
+Z(n1,n2,n3,n4,n5,n6,-1+n7,n8,n9,-1+n10,1+n11,n12,n13,1+n14)*rat(n11,1)
+Z(n1,n2,n3,n4,n5,n6,-1+n7,n8,n9,n10,n11,-1+n12,n13,2+n14)*rat(1+n14,1)
+Z(n1,n2,n3,n4,n5,n6,-1+n7,n8,n9,n10,n11,n12,1+n13,1+n14)*rat(n13,1)
+Z(n1,n2,n3,n4,n5,n6,-1+n7,n8,n9,n10,n11,n12,n13,1+n14)*rat(-2*ep-2*n4-1,1)
+Z(n1,n2,n3,n4,n5,n6,1+n7,-1+n8,n9,n10,n11,n12,n13,1+n14)*rat(-n7,1)
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,-1+n9,n10,n11,n12,1+n13,1+n14)*rat(n13,1)
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,-1+n9,n10,n11,n12,n13,2+n14)*rat(-2*n14-2,1)
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,-1+n10,1+n11,n12,n13,1+n14)*rat(-n11,1)
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,-1+n10,n11,1+n12,n13,1+n14)*rat(-2*n12,1)
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,-1+n10,n11,n12,n13,2+n14)*rat(-2*n14-2,1)
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,1+n10,n11,n12,n13,1+n14)*rat(2*n10,1)
```

```
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,-1+n11,n12,1+n13,1+n14)*rat(-2*n13,1)
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,1+n11,n12,n13,1+n14)*rat(n11,1)
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,-1+n12,n13,2+n14)*rat(-n14-1,1)
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,1+n12,n13,1+n14)*rat(-2*n12,1)
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,n12,1+n13,1+n14)*rat(-3*n13,1)
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,n12,n13,1+n14)*rat(10*ep+2*n1,1)
+Z(n1,n2,n3,n4,n5,n6,n7,-1+n8,n9,n10,n11,n12,n13,2+n14)*rat(-2*n14-2,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,-1+n9,-1+n10,n11,1+n12,n13,1+n14)*rat(-n12,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,-1+n9,-1+n10,n11,n12,1+n13,1+n14)*rat(-n13,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,-1+n9,n10,-1+n11,n12,n13,2+n14)*rat(-n14-1,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,-1+n9,n10,n11,n12,1+n13,1+n14)*rat(-n13,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,-1+n9,n10,n11,n12,n13,1+n14)*rat(-4*ep-2*n1-n3,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,n9,-1+n10,n11,n12,n13,1+n14)*rat(-n5+1,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,n9,1+n10,-1+n11,n12,n13,1+n14)*rat(n10,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,-1+n11,n12,n13,1+n14)*rat(2*ep+n5+2*n8
    +n9+n10+n11-5,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,1+n12,n13,n14)*rat(n12,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,-1+n13,1+n14)*rat(-2*ep-n5-2*n8
    -n9-n11-n14+3,1)
+Z(n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,1+n14)*rat(2*ep+n2+n7+
    2*n8+n9+n11+n14-4,1)
);
```

First results and benchmarks

- Reproduced 4-loop QCD β-function [Ritbergen, Vermaseren, Larin '97; Czakon '04]
- All ξ and ϵ 4-loop solution for QCD propagators and vertices [new!]

β_3 no gauge	10 minutes
β_3 1 gauge	38 minutes
β_3 all gauge	8.5 hours
no1(2,2,2,2,2,2,2,2,2,2,2,-1,-1,-1)	42 minutes

Table: Benchmark on 24 core machine

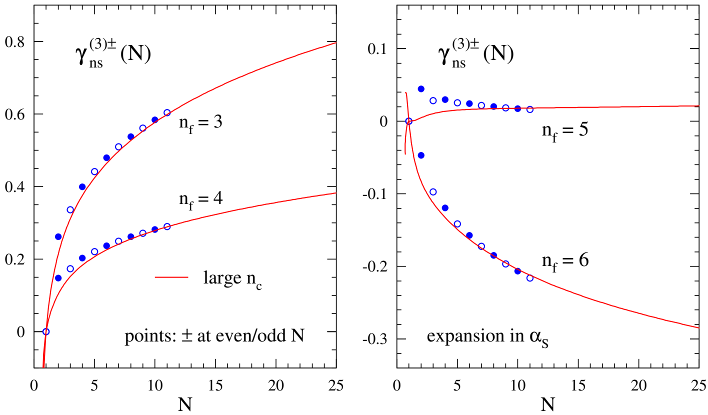
4-loop splitting functions

- Can we calculate the 4-loop corrections to the PDF evolution?
- Calculate Mellin moments N for as many N as possible
- Each $N \to N+2$ increases weight by 4
- 4-loop time: more than 1000 times 3-loop time

$$\gamma_{ij} = -P_{ij} = \sum_n \left(\frac{\alpha_s}{4\pi}\right)^n P_{ij}^{(n-1)} \quad i,j = q,g$$

Non-singlet (NS): combinations of γ_{qq}

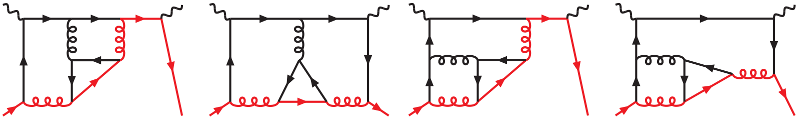
N3LO corrections to γ_{NS}



- $N = 2, 3, 4$ in agreement with [Baikov,Chetyrkin '06, Velizhanin '13,'14]
- New results from $N = 5$ to $N = 11$ [new; Moch,RUVV]

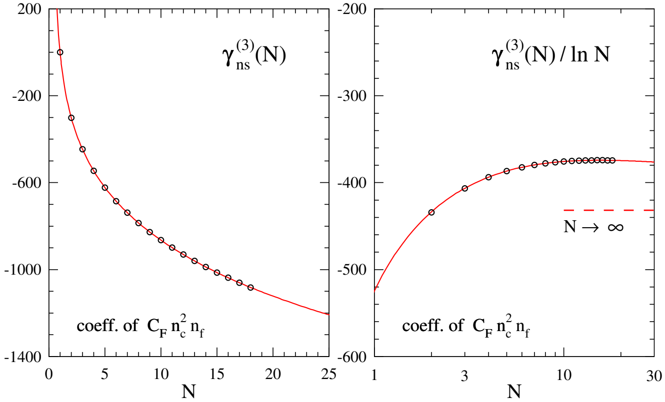
All-N results for n_f^3 to n_f^0

All 3-loop graphs (with insertion on gluon line) for n_f^2 in γ_{qq}:



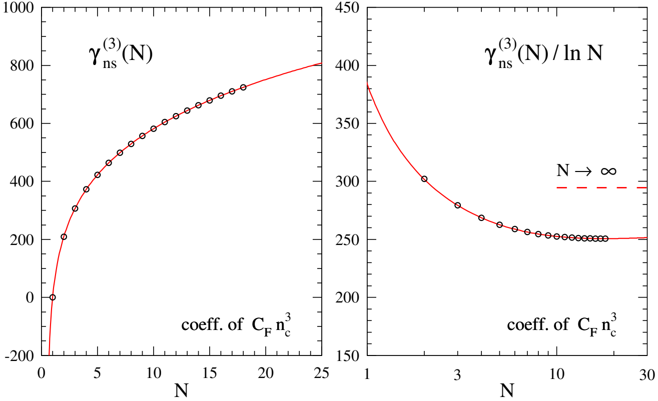
- Simple topologies for forcer $\rightarrow$ high moments
- Analytic form in N reconstructed with LLL-method [Velizhanin '12; Moch,Vermaseren,Vogt '14]
- Singlet splitting function matrix is now known at leading n_f [new; Davies,RUVV]
- n_f^1 and n_f^0 now known for γ_{NS} in large n_c limit using OPE [new; Moch,RUVV]
- The hardest topologies drop out in large n_c limit

n_f^1 coefficient in large n_c limit



- Reconstruction using 18 moments, checked with $N = 19$
- Confirmed result ($N \to \infty$) from [Henn,Steinhauser,Smirnov², '15]
- Agrees with predictions from small-x resummation

n_f^0 coefficient in large n_c limit



- Reconstruction using 18 moments, checked with $N = 19$
- Agrees with predictions of structural relations
- New term in cusp anomalous dimension!

New cusp term: n_f^0 in large n_c limit

$$
\begin{aligned}
\gamma^{(3)}_{\text{cusp}} =& + n_c^3 C_F \Big(+ \frac{84278}{81} - \frac{88832}{81}\zeta_2 + \frac{20992}{27}\zeta_3 + 1804\zeta_4 - \frac{352}{3}\zeta_3\zeta_2 \\
& \qquad -352\zeta_5 - 32\zeta_3^2 - 876\zeta_6 \Big) \\
& + n_c^2 C_F n_f \left(-\frac{39883}{81} + \frac{26692}{81}\zeta_2 - \frac{16252}{27}\zeta_3 - \frac{440}{3}\zeta_4 + \frac{256}{3}\zeta_3\zeta_2 + 224\zeta_5 \right) \\
& + n_c C_F n_f^2 \left(\frac{2119}{81} - \frac{608}{81}\zeta_2 + \frac{1280}{27}\zeta_3 - \frac{64}{3}\zeta_4 \right) \\
& + C_F n_f^3 \left(-\frac{32}{81} + \frac{64}{27}\zeta_3 \right) \\
& + \ldots
\end{aligned}
$$

Five loops

- Can we go to five loops?
- Recompute 5-loop QCD β-function [Baikov, Chetyrkin, Kühn '16] with general colour group and gauge? [see also: Luthe,Maier,Marquard,Schröder '16]
- 5-loop Forcer is hard:
    - Unknown master integrals
    - 20 parameter problem
    - More than 200 manual reductions!
    - However: 30% of topologies factorize

Five loops

- We are interested in anomalous dimensions
- Thus, we only need the UV pole part
- Can we IR rearrange to create a *carpet* diagram?

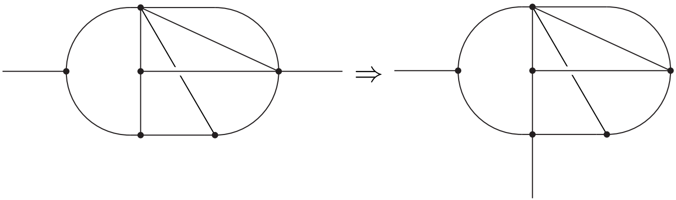


Figure: The graphs have the same UV poles

The R operation

- R is a renormalization operator
- BPHZ renormalization scheme
  [Bogoliubov,Parasiuk,Hepp,Zimmermann]
- R can be expressed as a recursive subtraction operator
  [Chetyrkin,Smirnov,Larin]
- The solution is Zimmermann's forest formula
- We are going to exploit that R can isolate the UV poles

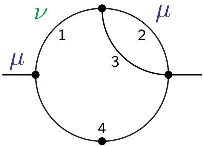
Assumptions

External momenta are massive, as colinear divergences are not supported.

Some applications for R and R^*

- Z,tau $\rightarrow$ Hadrons /Z-decay: QCD at N4LO (Global R^*) [Baikov,Chetyrkin,Kühn,'12]
- Anomalous dimensions
  - QCD
    - 5-loop β function (Global R^*) [Baikov,Chetyrkin,Kühn,'16]
    - 5-loop quark mass and field anomalous dimension (Global R^*) [Baikov,Chetyrkin,Kühn,'14]
  - ϕ^4
    - 6-loop wavefunction (Global & Local R^*) [Batkovich,Chetyrkin,Kompaniets,'16]
    - 6-loop β function (Local R^*) [Kompaniets, Panzer,'16]
- Decay rates: QCD corrections
  - H $\rightarrow$ bb N4LO (Global & Local R^*) [Baikov,Chetyrkin,Kühn,'05]
- g-2: QED corrections
  - 5 loop (Local R^* (?), even numerical) [Kinoshita,Aoyama,Hayakawa,Kinoshita,Nio,'14]
- Automation (for ϕ^4) [Batkovich,Kompaniets,'14]

Superficial UV divergence



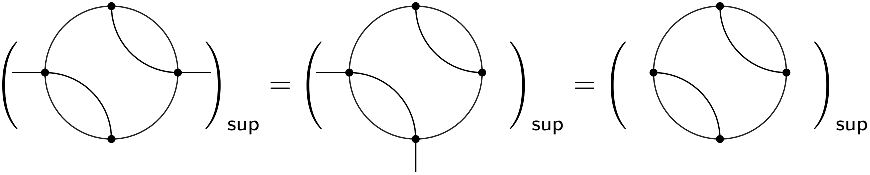
$$= \int d^D p_1 \int d^D p_2 \frac{Q \cdot p_2 p_1^\nu}{{p_1}^2 {p_2}^2 {p_3}^2 {p_4}^4}$$

- Superficial UV divergence: all loop momenta $\to \infty$
- Get degree of divergence through power counting:
  - Each loop contributes $+4$ due to the measure
  - $8 + 1 + 1 - 2 - 2 - 2 - 4 = 0$

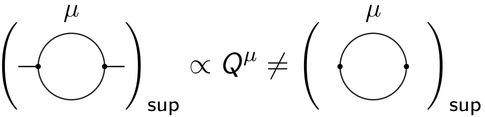
Logarithmically divergent integrals

Infrared rearrangement

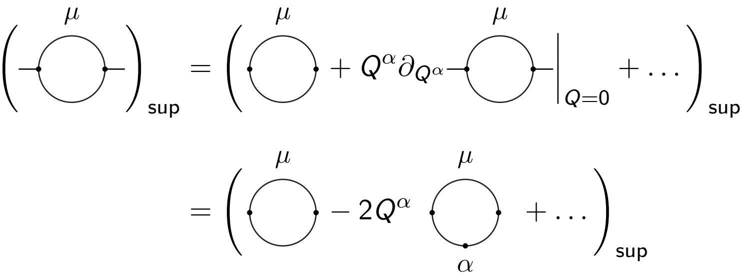
UV pole parts of log integrals do not depend on kinematics or masses



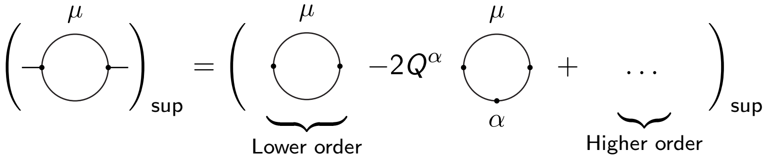
Linearly divergent integrals



Taylor expand in external momenta:

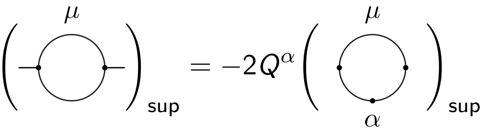


UV Taylor expansion



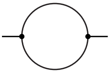
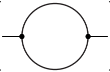
- Higher orders do not have a superficial divergence and are 0
- Lower orders are $>$ log massless tadpoles and are 0 because there is no scale

Thus:

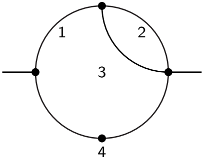


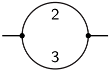
Subdivergences

- Subdivergence: singular behaviour when some loop momenta go to infinity (or zero)
- No subdivergences: superficial divergence is pole part
- Introduce pole operator: $K(\sum_{i=-\infty}^{\infty} c_i \epsilon^i) = \sum_{i=-\infty}^{-1} c_i \epsilon^i$

$\Big($  $\Big)_{\text{sup}} = K\Big($  $\Big) = \frac{1}{\epsilon}$

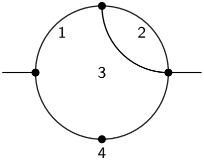
Subdivergences

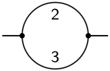


- Has log subdivergence: 

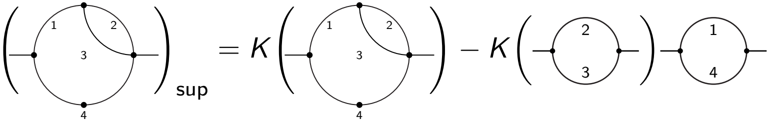
- Superficial divergence is pole part minus subdivergences

Subdivergences

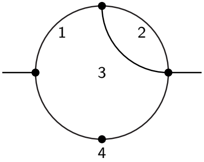


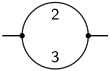
- Has log subdivergence: 

- Superficial divergence is pole part minus subdivergences

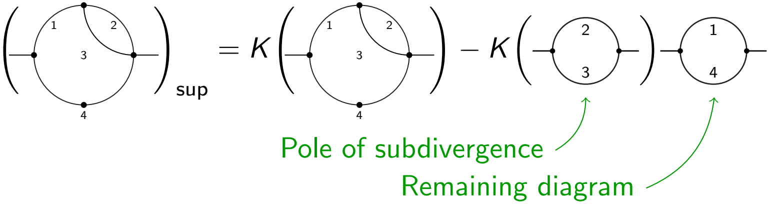


Subdivergences

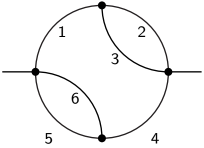


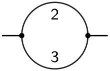
- Has log subdivergence: 

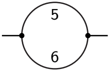
- Superficial divergence is pole part minus subdivergences



Multiple subdivergences

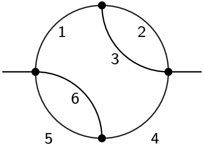


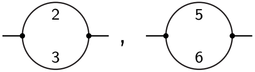
- Has log subdivergences: 

, 

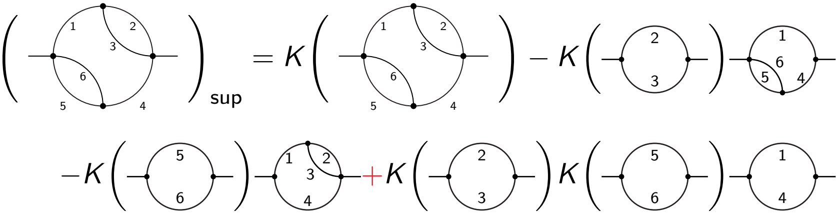
- Loop momenta can independently go to infinity

Multiple subdivergences

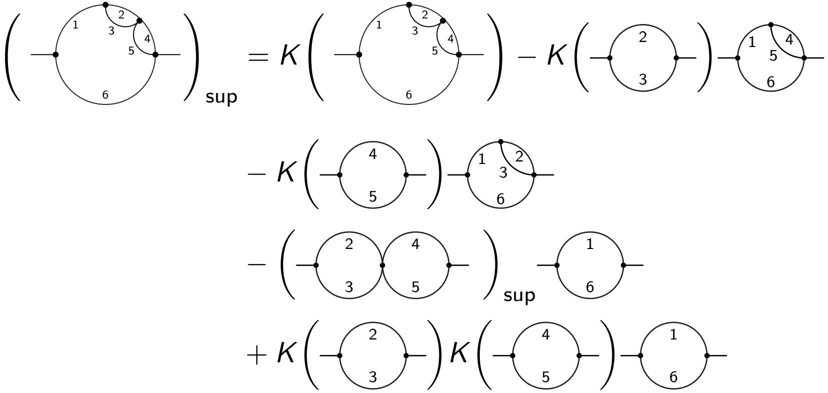


- Has log subdivergences: 

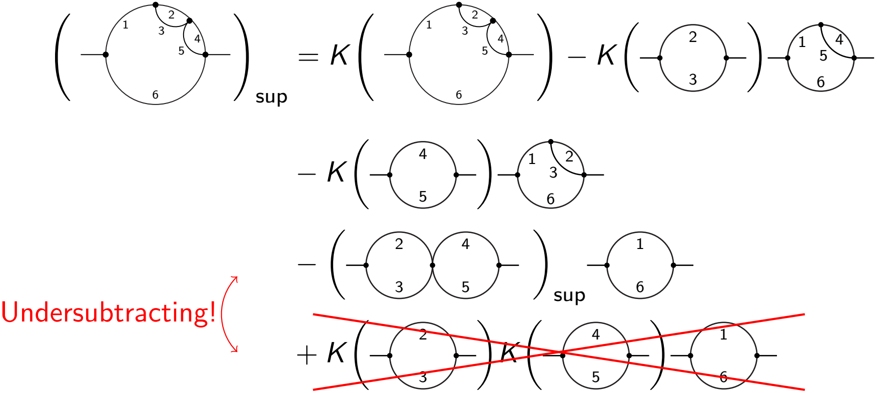
- Loop momenta can independently go to infinity



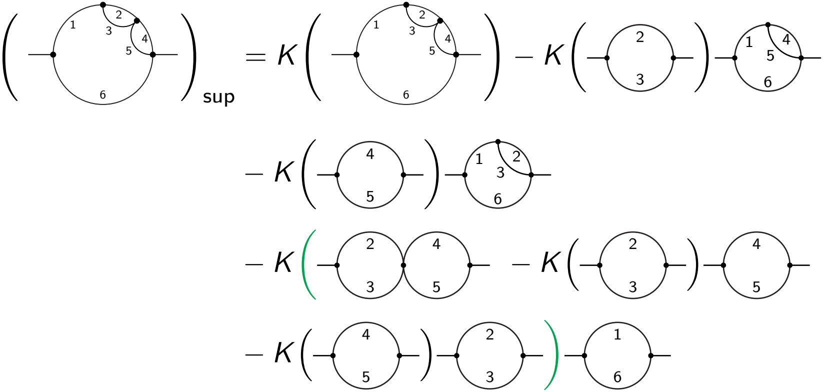
Overlapping subdivergences (I)



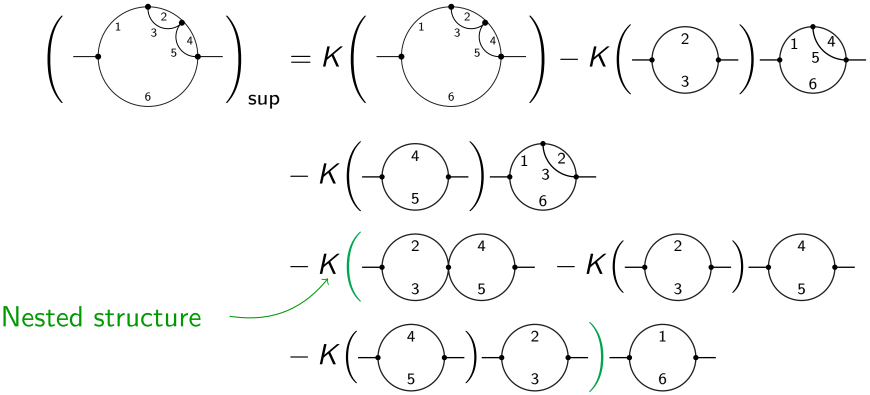
Overlapping subdivergences (I)



Overlapping subdivergences (II)



Overlapping subdivergences (II)



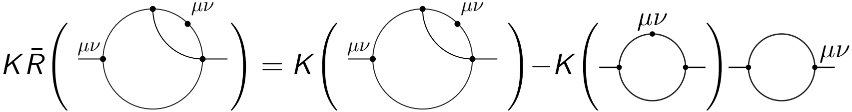
$\bar{R}$ rule

- Overlapping divergences give rise to nested/recursive structures
- Enter the $\bar{R}$ operation:

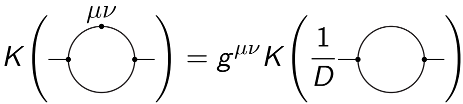
$$G_{\text{sup}} \equiv K\bar{R}G = K \sum_{S \in \bar{W}(G)} \left[\prod_{\gamma \in S} -K\bar{R}\gamma \right] * G/S$$

  - $W(G)$ is the set of all sets of non-overlapping UV subdiagrams
  - G/S is the remaining (quotient) diagram
  - γ are symmetrized massless vacuum diagrams [new]

Tensor reduction

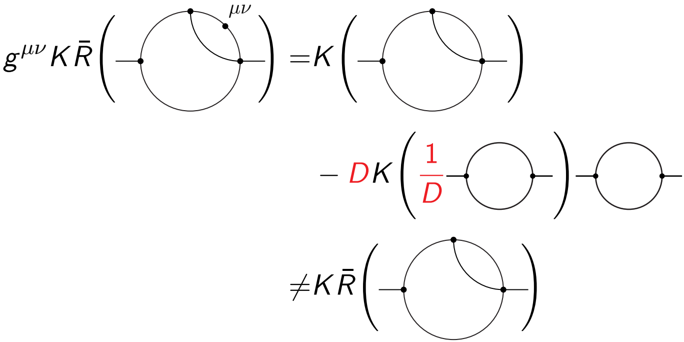


Tensor reduce:



Commutativity issues

- Contractions do not commute with R
- ϵ and thus D does not commute with K
- $\rightarrow$ Feynman rules do not commute with K nor R

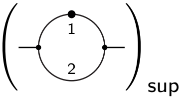


IR divergences

- What if loop momenta go to 0?
- All line combinations that go soft could be IR
- Not all lines with loop momentum are involved

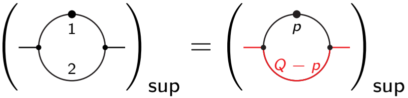
IR divergences

- What if loop momenta go to 0?
- All line combinations that go soft could be IR
- Not all lines with loop momentum are involved



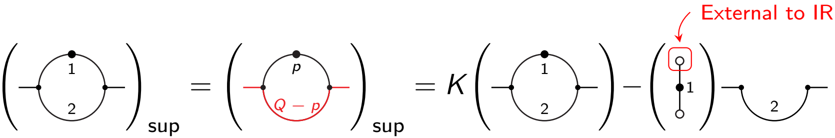
IR divergences

- What if loop momenta go to 0?
- All line combinations that go soft could be IR
- Not all lines with loop momentum are involved



IR divergences

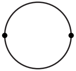
- What if loop momenta go to 0?
- All line combinations that go soft could be IR
- Not all lines with loop momentum are involved

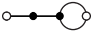
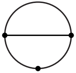


IR divergences

Recipe:

- Remove IR lines from diagram
- Set the IR momenta to 0 in remaining diagram
- Actual IR graph has external vertices shrunk
- Degree of IR divergence: minus UV divergence of IR graph

 $= \int \frac{d^D p}{p^4} =$ 

 $= \int \frac{d^D p_1 d^D p_2}{{p_1}^4 {p_2}^2 (p_1 - p_2)^2} =$ 

$\bar{R}^*$ rule

- $\bar{R}^*$ combines UV and IR subtractions [Chetyrkin,Smirnov]

$$K\bar{R}^*G = K \sum_{\substack{S\in\bar{W}(G)\\ S'\in\bar{W}'(G)\\ S\cap S'=\emptyset}} \left[\prod_{\gamma'\in S'} -K\underline{\mathrm{R}}^*\gamma'\right] * \left[\prod_{\gamma\in S} -K\bar{R}^*\gamma\right] * G/S\backslash S'$$

  - $W'(G)$ is the set of all non-overlapping IR subdiagrams
  - $K\underline{\mathrm{R}}^*$ is the superficial IR divergence [new]

Superficial IR

- How to compute the superficial IR divergence?

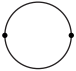
$$K(G) = K\bar{R}^*(G) + K\underline{\mathrm{R}}^*(G) + CT(G)$$

- $CT(G)$ are the subdivergence counter terms of G
- $K(G) = 0$ since IR graphs are massless tadpoles

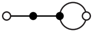
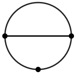
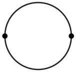
$$K\underline{\mathrm{R}}^*(G) = -K\bar{R}^*(G) - CT(G)$$

Superficial IR: examples

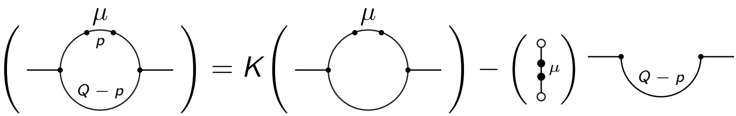
Without subdivergences:

$K\underline{\mathrm{R}}^*\Big($  $\Big) = -K\bar{R}^*\Big($  $\Big) = -\frac{1}{\epsilon}$

With subdivergence:

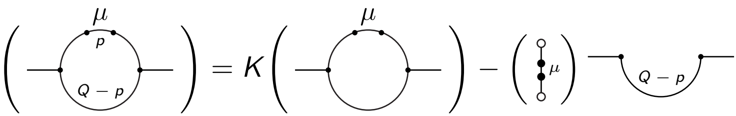
$K\underline{\mathrm{R}}^*\Big($  $\Big) = -K\bar{R}^*\Big($  $\Big) - K\underline{\mathrm{R}}^*\Big($  $\Big)K\Big($  $\Big)$

Linear IR divergences

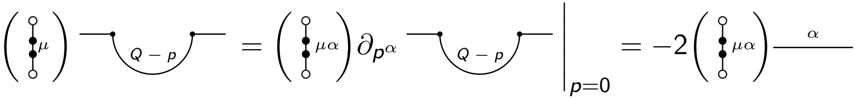


- Taylor expand the remaining diagram around $p = 0$
- Again, only the linear term survives

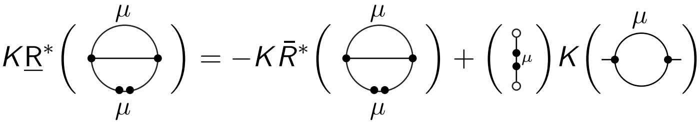
Linear IR divergences



- Taylor expand the remaining diagram around $p = 0$
- Again, only the linear term survives

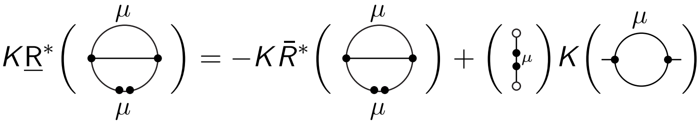


Tricky interactions

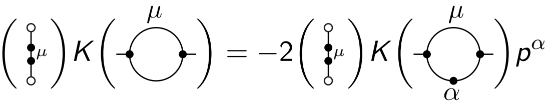


- Higher order UV may create vectors in remaining diagram
- IR Taylor should be sensitive to those!

Tricky interactions

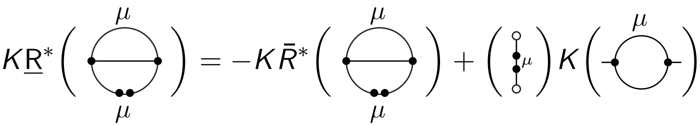


- Higher order UV may create vectors in remaining diagram
- IR Taylor should be sensitive to those!

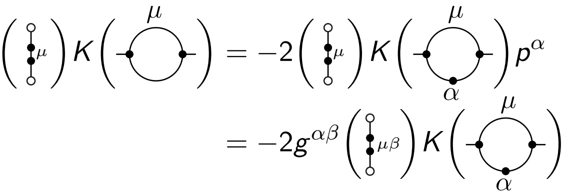


(UV Taylor)

Tricky interactions



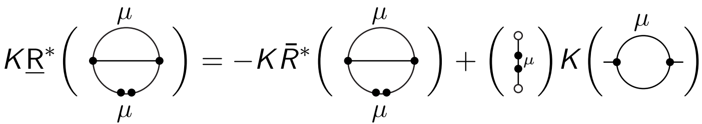
- Higher order UV may create vectors in remaining diagram
- IR Taylor should be sensitive to those!



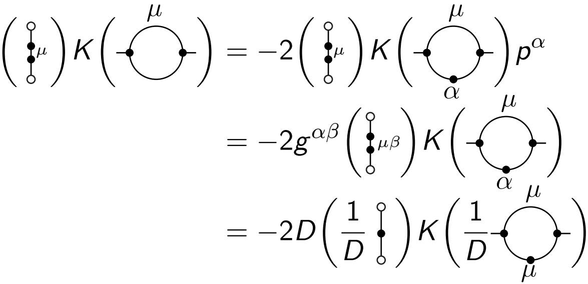
(UV Taylor)

(IR Taylor)

Tricky interactions



- Higher order UV may create vectors in remaining diagram
- IR Taylor should be sensitive to those!

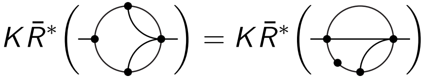


(UV Taylor)

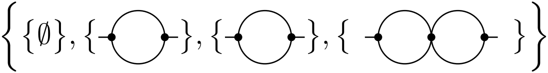
(IR Taylor)

(Projection)

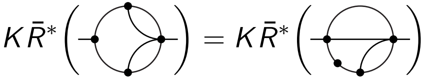
Complete example (I)



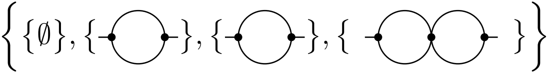
UV subdiagrams:



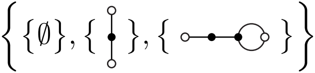
Complete example (I)



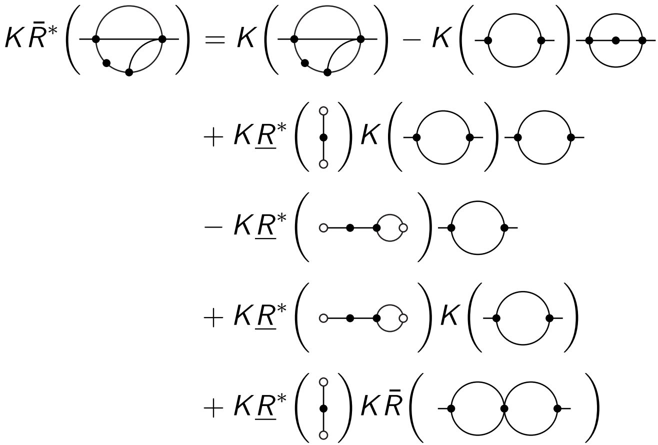
UV subdiagrams:



IR subdiagrams:



Complete example (II)



Results

Current status

Four loop beta function recomputed with R^* in 4 hours

Goals:

- 5 loop beta function with generic colour group
- Moments of 4-loop splitting functions from 3-loop integrals
- Generalise R^* to massless external states (?)

Conclusion

- Calculated up to $N = 11$ for γ_{NS}
- Calculated n_f^0 in large n_c limit of $\gamma_{NS} \rightarrow$ new term in cusp anomalous dimension
- Constructed an R^* method that works for QCD
- Written efficient R^*+Forcer package that in principle could compute superficial divergences of any 5-loop diagram with massive external lines

Acknowledgements

This work is supported by the ERC Advanced Grant no. 320651, "HEPGAME"