Fine-tuning the Laporta approach

Thomas Luthe

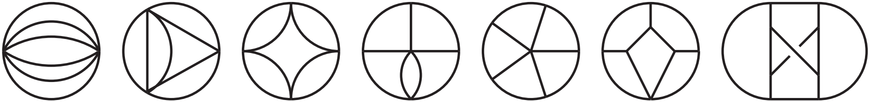
Bielefeld University

in collaboration with York Schröder

Radcor-Loopfest 2015

Motivation

- Previous talk: many applications for tadpole integrals
- Push computational limits to 5 loops, starting with fully massive tadpoles
- Low maintenance approach, one method for a complete set of integrals with little human input needed



Outline

- Short review of difference equation and factorial series
- Improvements
- Results

Difference equations and factorial series [Laporta '01]

- $I(x) = \int \frac{1}{D_1^x D_2^{b_2} \cdots D_n^{b_n}}$, here all D_i massive with $m = 1$
- $\sum_{k=0}^{R} q_k(x) I(x+k) = \sum_i \sum_{k=0}^{R_i - 1} p_{ik}(x) J_i(x+k)$, $J_i \in$ subsectors
- $I(x) = \sum_{s=0}^{\infty} \frac{\Gamma(x+1)}{\Gamma(x+s+d/2+1)} a_s$
- $\sum_{k=0}^{R'} q'_k(s) a_{s+k} = \sum_i \sum_{k=0}^{R'_i - 1} p'_{ik}(s) a_{i,s+k}$
- a_0 from large-x behaviour in terms of lower loop integrals
- $a_0 \xrightarrow{\text{rec. rel.}} a_{s_{\max}}, \sum_{s=0}^{s_{\max}} \frac{\Gamma(x_{\max}+1)}{\Gamma(x_{\max}+s+d/2+1)} a_s, \ I(x_{\max}) \xrightarrow{\text{diff. eq.}} I(1)$

Difference equations and factorial series [Laporta '01]

Advantages

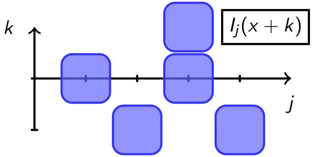
- Everything can be automated
- Works well also with divergent integrals and does not depend on a special class of functions
- High precision results for arbitrarily many orders in ϵ
- Can expand around any dimension
- Cross-checks by putting x on different propagators

Typical problems and limitations

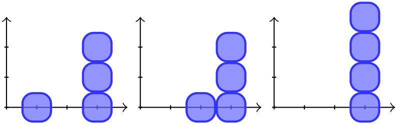
- Usually only numeric results $\rightarrow$ limited use for integrals with multiple scales
- Complexity of the coefficients in high order equations
- High orders of the recurrence relation
- Divergence of the factorial series in numerical evaluation

Coupled vs decoupled equations

Typically generate equations via IBP: $0 =$
[Chetyrkin, Tkachov '81]

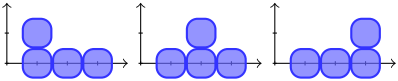


Decoupled equations



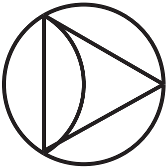
- simple solve algorithm
- need to solve only one integral numerically
- coeffs. grow large very quickly with R

Coupled equations



- more involved solve algorithm
- need to solve R integrals simultaneously
- coeffs. grow less quickly with R
- can choose master integral basis

Coupled vs decoupled equations

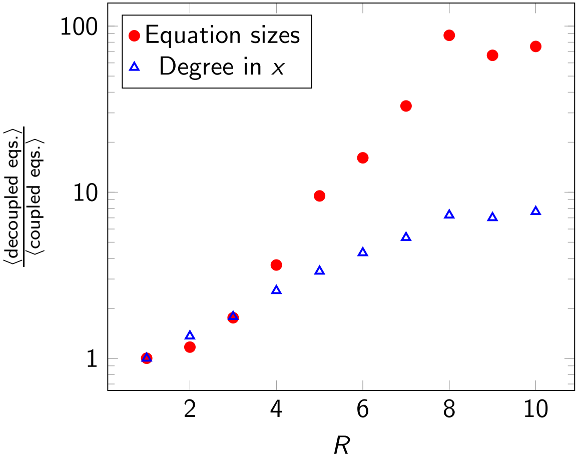


Example: Difference eq. 29703#3
Propagators: 7
Order: 8
Integrals: 1396 + sub-topologies
Input equations: 1400

	decoupled equations	coupled equations	coupled eqs. + opt. basis
$\langle \deg_x \rangle$ / coeff.	33.54	5.63	4.65
$\langle \deg_d \rangle$ / coeff.	31.59	6.51	4.51
$\langle$size as string$\rangle$ / eq.	$5.3 \cdot 10^5$	$7.0 \cdot 10^3$	$3.9 \cdot 10^3$
$\langle \#$ coefficients$\rangle$ / eq.	17.97	17.44	16.06
# steps to solve	$6.1 \cdot 10^4$	$5.9 \cdot 10^4$	$5.6 \cdot 10^4$
time to solve	$\sim$2d	68s	48s

Coupled vs decoupled equations

Scaling of the homogeneous part of equations with order R:



Recurrence relations

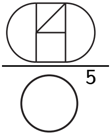
- $\sum_i \sum_{k=0}^{R'} q'_{ik}(s) a_{i,s+k} = 0$
- Can be reduced with the same algorithms as the difference equations!
- Translation from difference equations:
  order R, x-degree $N \rightarrow$ order R' with $N \leq R' \leq N + R$
  $\Rightarrow$ Translation loses information with every order of x
- Need input equations with $R\&N$ minimal.
- IBP: $R \leq 2$, $N \leq 1$, but not good enough.
  $\Rightarrow$ Reduce IBP-equations without multiplying or dividing by x & try to factor out $(x + \alpha)$.

Divergence factors and precision

- The numerical error grows by a factor F_D (F_R) with each iteration of the difference equations (recurrence relations).
- $D_{\text{end}} \approx \log_{10}\left[\binom{x_{\max}+s_{\max}}{s_{\max}} F_D^{-x_{\max}}\right]$ (precision of $I(1)$)
- $D_{\text{start}} \approx \log_{10}\left[\binom{x_{\max}+s_{\max}}{s_{\max}} F_R^{s_{\max}}\right]$ (precision of a_0)

Loops	F_D	F_R	$x_{\max}$	$s_{\max}$	D_{end}
1	1	1	300000	1000000	300000
2	3	1	300000	870000	145000
3	8	1	110000	900000	45000
4	15	1.125	21500	1000000	20000
5	24	12.928	700	18000	300

Results ($d = 4$)



$$
= -\frac{441}{40}\zeta(7)\epsilon^4 \quad \text{[Kazakov '83]}
$$
$$
+ 181.78223928612340820790788236018 6\ldots\epsilon^5
$$
$$
- 1725.9996137403520805951673992442 1\ldots\epsilon^6
$$
$$
+ 12797.999873726824046685551690337 6\ldots\epsilon^7
$$
$$
- 82986.852692581382160559047147390 9\ldots\epsilon^8
$$
$$
+ 496710.27285614832821523150890358 6\ldots\epsilon^9
$$
$$
+ \cdots
$$

See also *zig-zag conjecture* [Broadhurst, Kreimer '95][Brown, Schnetz '12]

Conclusions

- Improvements:
  - Choose coupled over decoupled eqs. to simplify coefficients
  - Reduction of recurrence relations
  - Avoid divergence in factorial series by increased precision
- Everything implemented in C++ , except polynomial algebra (Fermat [Lewis]), all time-critical code parallelised
- At the 5-loop level have produced difference equations up to order 20, recurrence relations up to order 28 + inhomogeneous parts
- Solved all fully massive master integrals for 37 out of 48 vacuum 5-loop diagrams with $\sim$ 300 digits precision, ≥ 10 orders in ϵ around $d = 4 - 2\epsilon, d = 3 - 2\epsilon$

Divergence factors

- $$I^{(hom)}(x) = \sum_{m=1}^{n} \mu_m^x \sum_{s=0}^{\infty} \frac{\Gamma(x+1)}{\Gamma(x+s-K_m+1)} a_{m,s}$$
- μ_m are roots of the characteristic polynomial $p(\mu)$.
- For decoupled eq. $\sum_{i=0}^{N} \sum_{k=0}^{R} p_{ik} x^i I(x+k) = 0$:

$$p(\mu) = \sum_{k=0}^{R} p_{Nk} \mu^k$$

- $$F_P^{(m)} = \max_i \left| \frac{\mu_m}{\mu_i} \right|$$
- $$F_R^{(m)} = \max \left\{ 1, \max_{\substack{i \\ \mu_i \neq \mu_m}} \left| \frac{\mu_m}{\mu_m - \mu_i} \right| \right\}$$