RADCOR 2019, Avignon, September 12, 2019

A refined machinery to calculate large moments from coupled systems of linear differential equations

Carsten Schneider

(in cooperation with J. Blümlein, P. Marquard and N. Rana)

Research Institute for Symbolic Computation (RISC)
Johannes Kepler University Linz

JOHANNES KEPLER UNIVERSITY LINZ

Der Wissenschaftsfonds.

Recurrence solving

A recurrence solver (Sigma.m)

GIVEN a recurrence

$a_0(N), \ldots, a_\delta(N)$: polynomials in N
$h(N)$: expression in indefinite nested sums
defined over hypergeometric products.

$$a_0(N)F(N) + \cdots + a_\delta(N)F(N+\delta) = h(N);$$

together with initial values $F(0), \ldots, F(\delta-1) \in \mathbb{K}$

A recurrence solver (Sigma.m)

GIVEN a recurrence

$a_0(N), \dots, a_\delta(N)$: polynomials in N
$h(N)$: expression in indefinite nested sums defined over hypergeometric products.

$$a_0(N)F(N) + \cdots + a_\delta(N)F(N+\delta) = h(N);$$

together with initial values $F(0), \dots, F(\delta - 1) \in \mathbb{K}$

DECIDE constructively if $F(N)$ can be expressed in terms indefinite nested sums defined over hypergeometric products.

A recurrence solver (Sigma.m)

GIVEN a recurrence

$a_0(N), \dots, a_\delta(N)$: polynomials in N
$h(N)$: expression in indefinite nested sums defined over hypergeometric products.

$$a_0(N)F(N) + \dots + a_\delta(N)F(N+\delta) = h(N);$$

together with initial values $F(0), \dots, F(\delta-1) \in \mathbb{K}$

DECIDE constructively if $F(N)$ can be expressed in terms indefinite nested sums defined over hypergeometric products.

Special cases of indefinite nested sums over hypergeometric products:

$$S_{2,1}(N) = \sum_{i=1}^{n} \frac{1}{i^2} \sum_{j=1}^{i} \frac{1}{j} \qquad \text{(harmonic sums)}$$

J. Blümlein and S. Kurth, Phys. Rev. D **60** (1999) 014018 [arXiv:hep-ph/9810241];
J.A.M. Vermaseren, Int. J. Mod. Phys. A **14** (1999) 2037 [arXiv:hep-ph/9806280].

A recurrence solver (Sigma.m)

GIVEN a recurrence

$a_0(N), \dots, a_\delta(N)$: polynomials in N
$h(N)$: expression in indefinite nested sums defined over hypergeometric products.

$$a_0(N)F(N) + \dots + a_\delta(N)F(N+\delta) = h(N);$$

together with initial values $F(0), \dots, F(\delta-1) \in \mathbb{K}$

DECIDE constructively if $F(N)$ can be expressed in terms indefinite nested sums defined over hypergeometric products.

Special cases of indefinite nested sums over hypergeometric products:

$$\sum_{k=1}^{n} \frac{2^k}{k} \sum_{i=1}^{k} \frac{2^{-i}}{i} \sum_{j=1}^{i} \frac{S_1(j)}{j} \qquad \text{(generalized harmonic sums)}$$

S. Moch, P. Uwer and S. Weinzierl, J. Math. Phys. **43** (2002) 3363 [hep-ph/0110083];
J. Ablinger, J. Blümlein and C. Schneider, J. Math. Phys. **54** (2013) 082301 [arXiv:1302.0378].

A recurrence solver (Sigma.m)

GIVEN a recurrence

$a_0(N), \dots, a_\delta(N)$: polynomials in N
$h(N)$: expression in indefinite nested sums defined over hypergeometric products.

$$a_0(N)F(N) + \dots + a_\delta(N)F(N+\delta) = h(N);$$

together with initial values $F(0), \dots, F(\delta-1) \in \mathbb{K}$

DECIDE constructively if $F(N)$ can be expressed in terms indefinite nested sums defined over hypergeometric products.

Special cases of indefinite nested sums over hypergeometric products:

$$\sum_{k=1}^{N} \frac{1}{(1+2k)^2} \sum_{j=1}^{k} \frac{1}{j^2} \sum_{i=1}^{j} \frac{1}{1+2i}$$ (cyclotomic harmonic sums)

J. Ablinger, J. Blümlein and C. Schneider, J. Math. Phys. **52** (2011) 102301 [arXiv:1105.6063].

A recurrence solver (Sigma.m)

GIVEN a recurrence

$a_0(N), \dots, a_\delta(N)$: polynomials in N
$h(N)$: expression in indefinite nested sums defined over hypergeometric products.

$$a_0(N)F(N) + \cdots + a_\delta(N)F(N+\delta) = h(N);$$

together with initial values $F(0), \dots, F(\delta - 1) \in \mathbb{K}$

DECIDE constructively if $F(N)$ can be expressed in terms indefinite nested sums defined over hypergeometric products.

Special cases of indefinite nested sums over hypergeometric products:

$$\sum_{j=1}^{N} \frac{4^j S_1(j-1)}{\binom{2j}{j} j^2} \qquad \text{(binomial sums)}$$

J. Ablinger, J. Blümlein, C. G. Raab and C. Schneider, J. Math. Phys. **55** (2014) 112301 [arXiv:1407.1822].

A recurrence solver (Sigma.m)

GIVEN a recurrence

$a_0(N), \dots, a_\delta(N)$: polynomials in N
$h(N)$: expression in indefinite nested sums defined over hypergeometric products.

$$a_0(N)F(N) + \dots + a_\delta(N)F(N+\delta) = h(N);$$

together with initial values $F(0), \dots, F(\delta - 1) \in \mathbb{K}$

DECIDE constructively if $F(N)$ can be expressed in terms indefinite nested sums defined over hypergeometric products.

Special cases of indefinite nested sums over hypergeometric products:

$$\sum_{h=1}^{N} 2^{-2h}(1-\eta)^h \binom{2h}{h} \sum_{k=1}^{h} \frac{2^{2k}}{k^2 \binom{2k}{k}}$$ (generalized binomial sums)

see talks of J. Ablinger and K. Schönwald

Sigma.m is based on difference ring/field theory

1. M. Karr. Summation in finite terms. *J. ACM*, 28:305–350, 1981.
2. P. Paule. Greatest factorial factorization and symbolic summation. J. Symbolic Comput. **20**(3), 235–268 (1995)
3. M. Petkovšek, H. S. Wilf, and D. Zeilberger. $A = B$. A. K. Peters, Wellesley, MA, 1996.
4. P. A. Hendriks and M. F. Singer. Solving difference equations in finite terms. *J. Symbolic Comput.*, 27(3):239–259, 1999.
5. M. Bronstein. On solutions of linear ordinary difference equations in their coefficient field. *J. Symbolic Comput.*, 29(6):841–877, 2000.
6. CS. Symbolic summation in difference fields. J. Kepler University, May 2001. PhD Thesis.
7. CS. A collection of denominator bounds to solve parameterized linear difference equations in $\Pi\Sigma$-extensions. *An. Univ. Timişoara Ser. Mat.-Inform.*, 42(2):163–179, 2004.
8. CS. Symbolic summation with single-nested sum extensions. In J. Gutierrez, editor, *Proc. ISSAC'04*, pages 282–289. ACM Press, 2004.
9. CS. Degree bounds to find polynomial solutions of parameterized linear difference equations in $\Pi\Sigma$-fields. *Appl. Algebra Engrg. Comm. Comput.*, 16(1):1–32, 2005.
10. CS. Product representations in $\Pi\Sigma$-fields. *Ann. Comb.*, 9(1):75–99, 2005.
11. CS. Solving parameterized linear difference equations in terms of indefinite nested sums and products. *J. Differ. Equations Appl.*, 11(9):799–821, 2005.
12. CS. Finding telescopers with minimal depth for indefinite nested sums and product expressions. In *Proc. ISSAC'05*, pages 285–292. ACM Press, 2005.
13. CS. Simplifying Sums in $\Pi\Sigma$-Extensions. *J. Algebra Appl.*, 6(3):415–441, 2007.
14. CS. A refined difference field theory for symbolic summation. *J. Symbolic Comput.*, 43(9):611–644, 2008. [arXiv:0808.2543v1].
15. CS. A Symbolic Summation Approach to Find Optimal Nested Sum Representations. In A. Carey, D. Ellwood, S. Paycha, and S. Rosenberg, editors, *Motives, Quantum Field Theory, and Pseudodifferential Operators*, pages 285–308. 2010.
16. CS. Parameterized Telescoping Proves Algebraic Independence of Sums. *Ann. Comb.*, 14(4):533–552, 2010. [arXiv:0808.2596].
17. CS. Structural Theorems for Symbolic Summation. *Appl. Algebra Engrg. Comm. Comput.*, 21(1):1–32, 2010.
18. C. Schneider. Simplifying Multiple Sums in Difference Fields. In: Computer Algebra in Quantum Field Theory: Integration, Summation and Special Functions, J. Blümlein, C. Schneider (ed.), Texts and Monographs in Symbolic Computation, pp. 325-360. Springer, 2013.
19. CS. Fast Algorithms for Refined Parameterized Telescoping in Difference Fields. In *Computer Algebra and Polynomials*, Lecture Notes in Computer Science (LNCS), Springer, 2014. arXiv:1307.7887 [cs.SC].
20. CS. A Difference Ring Theory for Symbolic Summation. J. Symb. Comput. 72, pp. 82-127. 2016.
21. CS. Summation Theory II: Characterizations of $R\Pi\Sigma$-extensions and algorithmic aspects. J. Symb. Comput. 80(3), pp. 616-664. 2017.
22. E.D. Ocansey, CS. Representing (q-)hypergeometric products and mixed versions in difference rings. In: Advances in Computer Algebra,, C. Schneider, E. Zima (ed.), Springer Proceedings in Mathematics & Statistics 226. 2018.
23. S.A. Abramov, M. Bronstein, M. Petkovšek, CS, in preparation

Example: A master integral from Ladder and V-topologies

[arXiv:1509.08324]

$$F(\varepsilon,N)=\int_0^1 dx\int_0^1 dy\int_0^1 dz x^{\varepsilon/2}y^{\varepsilon/2}(1-z)^{-\frac{3\varepsilon}{2}-2}z^{\frac{\varepsilon}{2}+N+1}$$
$$(1-xz)^{\varepsilon/2}\times(1-yz)^{\varepsilon/2}(x+y-1)^N$$

Ablinger's MultiIntegrate.m $\big\downarrow$ (9 hours)

$$a_0(\varepsilon,N)F(\varepsilon,N)+a_1(\varepsilon,N)F(\varepsilon,N+1)+\cdots+a_5(\varepsilon,N)F(\varepsilon,N+5)=0$$

Example: A master integral from Ladder and V-topologies

[arXiv:1509.08324]

$$F(\varepsilon,N)=\int_0^1 dx\int_0^1 dy\int_0^1 dz x^{\varepsilon/2}y^{\varepsilon/2}(1-z)^{-\frac{3\varepsilon}{2}-2}z^{\frac{\varepsilon}{2}+N+1}$$
$$(1-xz)^{\varepsilon/2}\times(1-yz)^{\varepsilon/2}(x+y-1)^N$$

Ablinger's MultiIntegrate.m $\downarrow$ (9 hours)

$$a_0(\varepsilon,N)F(\varepsilon,N)+a_1(\varepsilon,N)F(\varepsilon,N+1)+\cdots+a_5(\varepsilon,N)F(\varepsilon,N+5)=0$$

recurrence solver $\downarrow$

$F(\varepsilon,N)=$ expression in terms of special functions

A refined recurrence solver (Sigma.m)

GIVEN a recurrence

$a_0(\varepsilon, N), \dots, a_\delta(\varepsilon, N)$: polynomials in ε, N
$h_l(N), h_{l+1}(N), \dots, h_\lambda(N)$:
expressions in indefinite nested sums
defined over hypergeometric products.

$$a_0(\varepsilon, N)F(\varepsilon, N) + \dots + a_\delta(\varepsilon, N)F(\varepsilon, N+\delta)$$
$$= h_l(N)\varepsilon^l + h_{l+1}(N)\varepsilon^{l+1} + \dots h_\lambda(N)\varepsilon^r + O(\varepsilon^{r+1});$$

together with ε-expansions of $F(0), \dots, F(\delta - 1)$ up to a certain order.

A refined recurrence solver (Sigma.m)

GIVEN a recurrence

$a_0(\varepsilon,N),\dots,a_\delta(\varepsilon,N)$: polynomials in ε, N
$h_l(N), h_{l+1}(N),\dots,h_\lambda(N)$:
expressions in indefinite nested sums
defined over hypergeometric products.

$$a_0(\varepsilon,N)F(\varepsilon,N)+\dots+a_\delta(\varepsilon,N)F(\varepsilon,N+\delta)$$
$$=h_l(N)\varepsilon^l+h_{l+1}(N)\varepsilon^{l+1}+\dots h_\lambda(N)\varepsilon^r+O(\varepsilon^{r+1});$$

together with ε-expansions of $F(0),\dots,F(\delta-1)$ up to a certain order.

DECIDE constructively if the coefficients $F_i(N)$ of

$$F(N)=F_l(N)\,\varepsilon^l+F_{l+1}(N)\,\varepsilon^{l+1}+\dots+F_\lambda(N)\,\varepsilon^r+O(\varepsilon^{r+1})$$

can be given in terms of indefinite nested sums defined over hypergeometric products.

Blümlein, Klein, CS, Stan, J. Symbol. Comput. 2012; arXiv:1011.2656[cs.SC]
Ablinger, Blümlein, Round, CS, LL2012, arXiv:1210.1685 [cs.SC]

Ansatz (for power series)

$$
\begin{aligned}
&a_0(\varepsilon, N)\Big[F(N)\Big] \\
&+a_1(\varepsilon, N)\Big[F(N+1)\Big] \\
&+ \\
&\vdots \\
&+a_\delta(\varepsilon, N)\Big[F(N+\delta)\Big]
\end{aligned}
$$

$$= h_0(N) + h_1(N)\varepsilon + h_1(N)\varepsilon^2 + \dots$$

given (in terms of indefinite nested sums and products)

Ansatz (for power series)

$$
\begin{aligned}
& a_0(\varepsilon,N)\Big[F_0(N)+F_1(N)\varepsilon+F_2(N)\varepsilon^2+\dots\Big] \\
+& a_1(\varepsilon,N)\Big[F(N+1)\Big] \\
+& \\
\vdots & \\
+& a_\delta(\varepsilon,N)\Big[F(N+\delta)\Big]
\end{aligned}
$$

$$= h_0(N)+h_1(N)\varepsilon+h_1(N)\varepsilon^2+\dots$$

given (in terms of indefinite nested sums and products)

Ansatz (for power series)

$$
\begin{aligned}
&a_0(\varepsilon,N)\Big[F_0(N)+F_1(N)\varepsilon+F_2(N)\varepsilon^2+\dots\Big]\\
+&a_1(\varepsilon,N)\Big[F_0(N+1)+F_1(N+1)\varepsilon+F_2(N+1)\varepsilon^2+\dots\Big]\\
+&\\
&\vdots\\
+&a_\delta(\varepsilon,N)\Big[F(N+\delta)\Big]\\
&\qquad\qquad = h_0(N)+h_1(N)\varepsilon+h_1(N)\varepsilon^2+\dots
\end{aligned}
$$

given (in terms of indefinite nested sums and products)

Ansatz (for power series)

$$
\begin{aligned}
&a_0(\varepsilon,N)\Big[F_0(N)+F_1(N)\varepsilon+F_2(N)\varepsilon^2+\dots\Big]\\
&+a_1(\varepsilon,N)\Big[F_0(N+1)+F_1(N+1)\varepsilon+F_2(N+1)\varepsilon^2+\dots\Big]\\
&+\\
&\ \vdots\\
&+a_\delta(\varepsilon,N)\Big[F_0(N+\delta)+F_1(N+\delta)\varepsilon+F_2(N+\delta)\varepsilon^2+\dots\Big]\\
&\qquad\qquad\qquad\qquad\qquad\qquad=h_0(N)+h_1(N)\varepsilon+h_1(N)\varepsilon^2+\dots
\end{aligned}
$$

given (in terms of indefinite nested sums and products)

Ansatz (for power series)

$$
\begin{aligned}
&a_0(\varepsilon,N)\Big[F_0(N)+F_1(N)\varepsilon+F_2(N)\varepsilon^2+\dots\Big]\\
&+a_1(\varepsilon,N)\Big[F_0(N+1)+F_1(N+1)\varepsilon+F_2(N+1)\varepsilon^2+\dots\Big]\\
&+\\
&\ \vdots\\
&+a_\delta(\varepsilon,N)\Big[F_0(N+\delta)+F_1(N+\delta)\varepsilon+F_2(N+\delta)\varepsilon^2+\dots\Big]\\
&\qquad\qquad\qquad\qquad=h_0(N)+h_1(N)\varepsilon+h_1(N)\varepsilon^2+\dots
\end{aligned}
$$

$\Downarrow$ constant terms must agree

$$
\boxed{a_0(0,N)F_0(N)+a_1(0,N)F_0(N+1)+\dots+a_\delta(0,N)F_0(N+\delta)=h_0(N)}
$$

Ansatz (for power series)

$$
\begin{aligned}
&a_0(\varepsilon,N)\Big[F_0(N)+F_1(N)\varepsilon+F_2(N)\varepsilon^2+\dots\Big]\\
&+a_1(\varepsilon,N)\Big[F_0(N+1)+F_1(N+1)\varepsilon+F_2(N+1)\varepsilon^2+\dots\Big]\\
&+\\
&\ \vdots\\
&+a_\delta(\varepsilon,N)\Big[F_0(N+\delta)+F_1(N+\delta)\varepsilon+F_2(N+\delta)\varepsilon^2+\dots\Big]\\
&\qquad\qquad\qquad\qquad\qquad=h_0(N)+h_1(N)\varepsilon+h_1(N)\varepsilon^2+\dots
\end{aligned}
$$

$\Downarrow$ constant terms must agree

$$
\boxed{a_0(0,N)F_0(N)+a_1(0,N)F_0(N+1)+\dots+a_\delta(0,N)F_0(N+\delta)=h_0(N)}
$$

REC solver: Given the initial values $F_0(1), F_0(2), \dots, F_0(\delta)$, **decide** if $F_0(N)$ can be written in terms of indefinite nested sums and products.

Ansatz (for power series)

$$
\begin{aligned}
& a_0(\varepsilon,N)\Big[F_0(N)+F_1(N)\varepsilon+F_2(N)\varepsilon^2+\dots\Big]\\
+& a_1(\varepsilon,N)\Big[F_0(N+1)+F_1(N+1)\varepsilon+F_2(N+1)\varepsilon^2+\dots\Big]\\
+&\\
&\vdots\\
+& a_\delta(\varepsilon,N)\Big[F_0(N+\delta)+F_1(N+\delta)\varepsilon+F_2(N+\delta)\varepsilon^2+\dots\Big]\\
&\qquad = h_0(N)+h_1(N)\varepsilon+h_1(N)\varepsilon^2+\dots
\end{aligned}
$$

$\Downarrow$ constant terms must agree

$$
\boxed{a_0(0,N)F_0(N)+a_1(0,N)F_0(N+1)+\dots+a_\delta(0,N)F_0(N+\delta)=h_0(N)}
$$

Ansatz (for power series)

$$
\begin{aligned}
&a_0(\varepsilon,N)\Big[F_0(N)+F_1(N)\varepsilon+F_2(N)\varepsilon^2+\dots\Big]\\
+&a_1(\varepsilon,N)\Big[F_0(N+1)+F_1(N+1)\varepsilon+F_2(N+1)\varepsilon^2+\dots\Big]\\
+&\\
\vdots&\\
+&a_\delta(\varepsilon,N)\Big[F_0(N+\delta)+F_1(N+\delta)\varepsilon+F_2(N+\delta)\varepsilon^2+\dots\Big]\\
&\qquad\qquad\qquad\qquad = h_0(N)+h_1(N)\varepsilon+h_1(N)\varepsilon^2+\dots
\end{aligned}
$$

$\Downarrow$ constant terms must agree

$$
\boxed{a_0(0,N)F_0(N)+a_1(0,N)F_0(N+1)+\dots+a_\delta(0,N)F_0(N+\delta)=h_0(N)}
$$

$$
\begin{aligned}
&a_0(\varepsilon,N)\Big[F_1(N)\varepsilon + F_2(N)\varepsilon^2 + \dots\Big]\\
&+a_1(\varepsilon,N)\Big[F_1(N+1)\varepsilon + F_2(N+1)\varepsilon^2 + \dots\Big]\\
&+\\
&\ \vdots\\
&+a_\delta(\varepsilon,N)\Big[F_1(N+\delta)\varepsilon + F_2(N+\delta)\varepsilon^2 + \dots\Big]\\
&\qquad\qquad\qquad\qquad = h'_0(N) + h'_1(N)\varepsilon + h'_2(N)\varepsilon^2 + \dots
\end{aligned}
$$

$$
\begin{aligned}
&a_0(\varepsilon,N)\Big[F_1(N)\varepsilon + F_2(N)\varepsilon^2 + \dots\Big]\\
&+a_1(\varepsilon,N)\Big[F_1(N+1)\varepsilon + F_2(N+1)\varepsilon^2 + \dots\Big]\\
&+\\
&\vdots\\
&+a_\delta(\varepsilon,N)\Big[F_1(N+\delta)\varepsilon + F_2(N+\delta)\varepsilon^2 + \dots\Big]\\
&\qquad\qquad\qquad = \underbrace{h_0'(N)}_{=0} + h_1'(N)\varepsilon + h_2'(N)\varepsilon^2 + \dots
\end{aligned}
$$

Devide by ε

$$
\begin{aligned}
&a_0(\varepsilon,N)\Big[F_1(N)+F_2(N)\varepsilon+\dots\Big]\\
+&a_1(\varepsilon,N)\Big[F_1(N+1)+F_2(N+1)\varepsilon+\dots\Big]\\
+&\\
&\vdots\\
+&a_\delta(\varepsilon,N)\Big[F_1(N+\delta)+F_2(N+\delta)\varepsilon+\dots\Big]=h_1'(N)+h_2'(N)\varepsilon+\dots
\end{aligned}
$$

Repeat to get $F_1(N), F_2(N), \dots$

Remark: Works the same for Laurent series.

Blümlein, Klein, CS, Stan, J. Symbol. Comput. 2012; arXiv:1011.2656[cs.SC]
Ablinger, Blümlein, Round, CS, LL2012, arXiv:1210.1685 [cs.SC]

$$F(N) = \int_0^1 dx \int_0^1 dy \int_0^1 dz x^{\varepsilon/2} y^{\varepsilon/2} (1-z)^{-\frac{3\varepsilon}{2}-2} z^{\frac{\varepsilon}{2}+N+1} (1-xz)^{\varepsilon/2} \times (1-yz)^{\varepsilon/2} (x+y-1)^N$$

↓ (package `MultiIntegrate.m`)

$$a_0(\varepsilon, N)F(N) + a_1(\varepsilon, N)F(N+1) + \cdots + a_5(\varepsilon, N)F(N+5) = 0$$

$$F(2) = \frac{20}{27\varepsilon^3} - \frac{40}{27\varepsilon^2} + \frac{1}{\varepsilon}\left(\frac{1393}{486} + \frac{5\zeta_2}{18}\right) + \dots$$

$$\vdots$$

$$F(6) = \frac{22}{147\varepsilon^3} - \frac{535}{2058\varepsilon^2} + \frac{1}{\varepsilon}\left(\frac{630043}{1234800} + \frac{11\zeta_2}{196}\right) + \dots$$

$$F(N)=\int_0^1 dx\int_0^1 dy\int_0^1 dz x^{\varepsilon/2}y^{\varepsilon/2}(1-z)^{-\frac{3\varepsilon}{2}-2}z^{\frac{\varepsilon}{2}+N+1}(1-xz)^{\varepsilon/2}\times(1-yz)^{\varepsilon/2}(x+y-1)^N$$

↓ (package `MultiIntegrate.m`)

$$a_0(\varepsilon,N)F(N)+a_1(\varepsilon,N)F(N+1)+\cdots+a_5(\varepsilon,N)F(N+5)=0$$

$$F(2)=\frac{20}{27\varepsilon^3}-\frac{40}{27\varepsilon^2}+\frac{1}{\varepsilon}\left(\frac{1393}{486}+\frac{5\zeta_2}{18}\right)+\dots$$

$$\vdots$$

$$F(6)=\frac{22}{147\varepsilon^3}-\frac{535}{2058\varepsilon^2}+\frac{1}{\varepsilon}\left(\frac{630043}{1234800}+\frac{11\zeta_2}{196}\right)+\dots$$

↓ (summation package `Sigma.m`)

$$\boxed{F(N)=F_{-3}(N)\,\varepsilon^{-3}+F_{-2}(N)\,\varepsilon^{-2}+F_{-1}(N)\,\varepsilon^{-1}+\dots}$$

We get
$F_{-3}(N) = \frac{8(-1)^N}{3(N+1)(N+2)} + \frac{8(2N+3)}{3(N+1)^2(N+2)}$

We get

$$F_{-3}(N) = \frac{8(-1)^N}{3(N+1)(N+2)} + \frac{8(2N+3)}{3(N+1)^2(N+2)}$$

$$F_{-2}(N) = -\frac{4(-1)^N\left(3N^3+18N^2+31N+18\right)}{3(N+1)^3(N+2)^2} - \frac{4\left(6N^3+32N^2+51N+26\right)}{3(N+1)^3(N+2)^2}$$

We get

$$F_{-3}(N) = \frac{8(-1)^N}{3(N+1)(N+2)} + \frac{8(2N+3)}{3(N+1)^2(N+2)}$$

$$F_{-2}(N) = -\frac{4(-1)^N\big(3N^3+18N^2+31N+18\big)}{3(N+1)^3(N+2)^2} - \frac{4\big(6N^3+32N^2+51N+26\big)}{3(N+1)^3(N+2)^2}$$

$$\begin{aligned}F_{-1}(N) = (-1)^N \Bigg(& \frac{2\big(9N^5+81N^4+295N^3+533N^2+500N+204\big)}{3(N+1)^4(N+2)^3} + \frac{\zeta_2}{(N+1)(N+2)} \\ & + \frac{2\big(18N^5+150N^4+490N^3+755N^2+536N+132\big)}{3(N+1)^4(N+2)^3} + \frac{(2N+3)\zeta_2}{(N+1)^2(N+2)} \\ & + \left(-\frac{4}{(N+1)^2(N+2)} + \frac{4(-1)^N}{(N+1)(N+2)}\right) S_2 \\ & + \left(\frac{4(-1)^N}{3(N+1)(N+2)} - \frac{4(N+9)}{3(N+1)^2(N+2)}\right) S_{-2}\end{aligned}$$

Solving coupled systems

[coming, e.g., from IBP methods]

Given invert. $A(x) \in \mathbb{K}(x)^{\lambda\times\lambda}$ and $\hat{R}_1(x), \dots, \hat{R}_\lambda(x)$ (in terms of special functions)
Determine $\hat{I}_1(x), \dots, \hat{I}_\lambda(x)$ (for given initial values) s.t.

$$D_x \begin{pmatrix} \hat{I}_1(x) \\ \dots \\ \hat{I}_\lambda(x) \end{pmatrix} = A(x) \begin{pmatrix} \hat{I}_1(x) \\ \dots \\ \hat{I}_\lambda(x) \end{pmatrix} + \begin{pmatrix} \hat{R}_1(x) \\ \dots \\ \hat{R}_\lambda(x) \end{pmatrix}$$

given

Given invert. $A(x) \in \mathbb{K}(x)^{\lambda\times\lambda}$ and $\hat{R}_1(x), \ldots, \hat{R}_\lambda(x)$ (in terms of special functions)
Determine $\hat{I}_1(x), \ldots, \hat{I}_\lambda(x)$ (for given initial values) s.t.

$$D_x \begin{pmatrix} \hat{I}_1(x) \\ \ldots \\ \hat{I}_\lambda(x) \end{pmatrix} = A(x) \begin{pmatrix} \hat{I}_1(x) \\ \ldots \\ \hat{I}_\lambda(x) \end{pmatrix} + \begin{pmatrix} \hat{R}_1(x) \\ \ldots \\ \hat{R}_\lambda(x) \end{pmatrix}$$

$\downarrow$ uncoupling algorithms
(Zürcher,Abramov/Zima,Gauss,...)

1. $\hat{I}_1(x)$ is a solution of

$$d_0(x)\hat{I}_1(x) + d_1(x)D_x\hat{I}_1(x) + \cdots + d_\lambda(x)D_x^\lambda\hat{I}_1(x) = \hat{r}(x)$$

Given invert. $A(x) \in \mathbb{K}(x)^{\lambda\times\lambda}$ and $\hat{R}_1(x), \dots, \hat{R}_\lambda(x)$ (in terms of special functions)
Determine $\hat{I}_1(x), \dots, \hat{I}_\lambda(x)$ (for given initial values) s.t.

$$D_x \begin{pmatrix} \hat{I}_1(x) \\ \dots \\ \hat{I}_\lambda(x) \end{pmatrix} = A(x) \begin{pmatrix} \hat{I}_1(x) \\ \dots \\ \hat{I}_\lambda(x) \end{pmatrix} + \begin{pmatrix} \hat{R}_1(x) \\ \dots \\ \hat{R}_\lambda(x) \end{pmatrix}$$

$\downarrow$ uncoupling algorithms
(Zürcher,Abramov/Zima,Gauss,...)

1. $\hat{I}_1(x)$ is a solution of

$$d_0(x)\hat{I}_1(x) + d_1(x)D_x\hat{I}_1(x) + \dots + d_\lambda(x)D_x^\lambda\hat{I}_1(x) = \hat{r}(x)$$

2. For $i = 2, \dots, r$ we get

$$\hat{I}_i(x) = \mathsf{LinCom}(\hat{I}_1(x), \dots, D_x^{\lambda-1}\hat{I}_1(x)) + \mathsf{LinCom}(..., D^i\hat{R}_i(x), \dots)$$

Given invert. $A(x) \in \mathbb{K}(x)^{\lambda\times\lambda}$ and $\hat{R}_1(x),\dots,\hat{R}_\lambda(x)$ (in terms of special functions)
Determine $\hat{I}_1(x),\dots,\hat{I}_\lambda(x)$ (for given initial values) s.t.

$$D_x \begin{pmatrix}\hat{I}_1(x)\\ \dots \\ \hat{I}_\lambda(x)\end{pmatrix} = A(x) \begin{pmatrix}\hat{I}_1(x)\\ \dots \\ \hat{I}_\lambda(x)\end{pmatrix} + \begin{pmatrix}\hat{R}_1(x)\\ \dots \\ \hat{R}_\lambda(x)\end{pmatrix}$$

↓ uncoupling algorithms
(Zürcher,Abramov/Zima,Gauss,...)

1. $\hat{I}_1(x)$ is a solution of

$$d_0(x)\hat{I}_1(x) + d_1(x)D_x\hat{I}_1(x) + \dots + d_\lambda(x)D_x^\lambda\hat{I}_1(x) = \hat{r}(x)$$

DE-solver
(see, e.g., [arXiv:1810.12261])

Given invert. $A(x) \in \mathbb{K}(x)^{\lambda\times\lambda}$ and $\hat{R}_1(x), \dots, \hat{R}_\lambda(x)$ (in terms of special functions)
Determine $\hat{I}_1(x), \dots, \hat{I}_\lambda(x)$ (for given initial values) s.t.

$$D_x \begin{pmatrix} \hat{I}_1(x) \\ \dots \\ \hat{I}_\lambda(x) \end{pmatrix} = A(x) \begin{pmatrix} \hat{I}_1(x) \\ \dots \\ \hat{I}_\lambda(x) \end{pmatrix} + \begin{pmatrix} \hat{R}_1(x) \\ \dots \\ \hat{R}_\lambda(x) \end{pmatrix}$$

↓ uncoupling algorithms
(Zürcher,Abramov/Zima,Gauss,...)

1. $\hat{I}_1(x)$ is a solution of

$$d_0(x)\hat{I}_1(x) + d_1(x)D_x\hat{I}_1(x) + \dots + d_\lambda(x)D_x^\lambda \hat{I}_1(x) = \hat{r}(x)$$

DE-solver
(see, e.g., [arXiv:1810.12261])

REC-solver

The DE-REC approach

DE system

$$D_x \hat{I}(x) = A\,\hat{I}(x) + \hat{R}(x)$$

The DE-REC approach

DE system	OreSys package (S. Gerhold) → uncoupling algorithm	uncoupled DE system
$D_x \hat{I}(x) = A\,\hat{I}(x) + \hat{R}(x)$		$\sum_i d_i(x) D_x^i \hat{I}_1(x) = \hat{r}(x)$ $\hat{I}_k(x) = \mathsf{expr}_k(D_x^i \hat{I}_1(x)), k > 1$

The DE-REC approach

DE system	OreSys package (S. Gerhold) → uncoupling algorithm	uncoupled DE system
$D_x \hat{I}(x) = A\,\hat{I}(x) + \hat{R}(x)$		$\sum_i d_i(x) D_x^i \hat{I}_1(x) = \hat{r}(x)$ $\hat{I}_k(x) = \text{expr}_k(D_x^i \hat{I}_1(x)), k > 1$

$$\hat{I}_1(x) = \sum_{N=0}^{\infty} I_1(N) x^N$$

The DE-REC approach

DE system

$$D_x\hat{I}(x) = A\,\hat{I}(x) + \hat{R}(x)$$

→ OreSys package (S. Gerhold) — uncoupling algorithm →

uncoupled DE system

$$\sum_i d_i(x) D_x^i \hat{I}_1(x) = \hat{r}(x)$$

$$\hat{I}_k(x) = \text{expr}_k(D_x^i \hat{I}_1(x)), k > 1$$

$$\hat{I}_1(x) = \sum_{N=0}^{\infty} I_1(N) x^N$$

↓ coeff. comparison w.r.t. x^N ↓

linear recurrence

$$\sum_i a_i(N) I_1(N+i) = r(N)$$

The DE-REC approach

DE system

$$D_x \hat{I}(x) = A\,\hat{I}(x) + \hat{R}(x)$$

→ OreSys package (S. Gerhold) — uncoupling algorithm →

uncoupled DE system

$$\sum_i d_i(x) D_x^i \hat{I}_1(x) = \hat{r}(x)$$

$$\hat{I}_k(x) = \mathsf{expr}_k(D_x^i \hat{I}_1(x)), k > 1$$

$$\hat{I}_1(x) = \sum_{N=0}^{\infty} I_1(N) x^N$$

↓ coeff. comparison w.r.t. x^N ↓

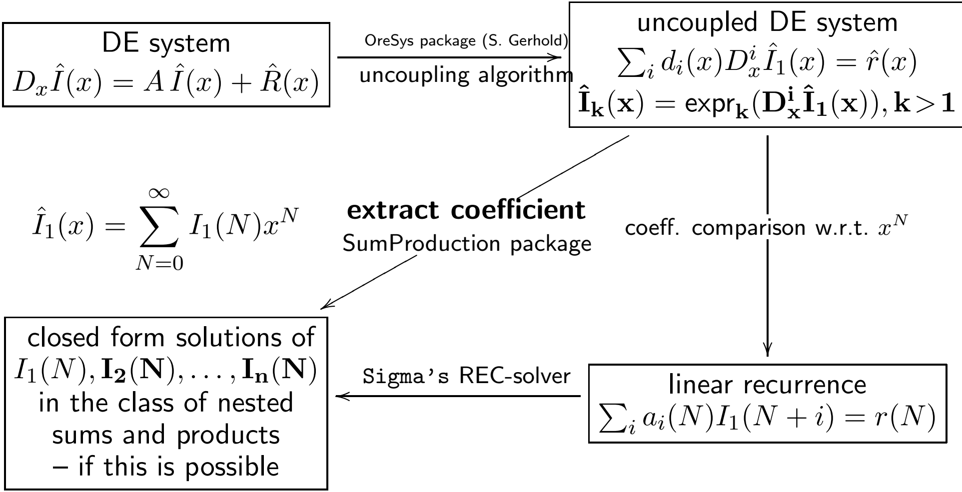
linear recurrence

$$\sum_i a_i(N) I_1(N+i) = r(N)$$

← `Sigma's` REC-solver ←

closed form solutions of $I_1(N)$ in the class of nested sums and products – if this is possible

The DE-REC approach (SolveCoupledSystem package)



J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel, CS. Comput. Phys. Comm. 202, pp. 33-112.

General strategy:

$$\Big\downarrow \text{IBP methods}$$

- Recursively defined coupled DE systems for unknown MIs $\hat{I}_i(x)$
- $\hat{D}(x) = \mathsf{LinComb}(\hat{I}_1(x), \ldots, \hat{I}_u(x))$

General strategy:

$\downarrow$ IBP methods

- Recursively defined coupled DE systems for unknown MIs $\hat{I}_i(x)$
- $\hat{D}(x) = \text{LinComb}(\hat{I}_1(x), \ldots, \hat{I}_u(x))$

$\downarrow$ solver for $\hat{I}_i(x) = \sum_{N=0}^{\infty} I_i(N) x^N$

$$I_i(N) = \varepsilon^{-3} F_{-3}(N) + \varepsilon^{-2} F_{-2}(N) + \varepsilon^{-1} F_{-1}(N) + \ldots$$

General strategy:

$$\Big\downarrow \text{IBP methods}$$

- Recursively defined coupled DE systems for unknown MIs $\hat{I}_i(x)$
- $\hat{D}(x) = \text{LinComb}(\hat{I}_1(x), \ldots, \hat{I}_u(x))$

$$\Big\downarrow \text{solver for } \hat{I}_i(x) = \sum_{N=0}^{\infty} I_i(N) x^N$$

$$I_i(N) = \varepsilon^{-3} F_{-3}(N) + \varepsilon^{-2} F_{-2}(N) + \varepsilon^{-1} F_{-1}(N) + \ldots$$

$$\Big\downarrow \text{plug into } \hat{D}(x) = \sum_{N=0}^{\infty} D(N) x^N$$

$$D(N) = \varepsilon^{-3} D_{-3}(N) + \varepsilon^{-2} D_{-2}(N) + \varepsilon^{-1} D_{-1}(N) + \varepsilon^{0} D_0(N) + \ldots$$

Concrete calculations:

- J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. Hasselhuhn, A. von Manteuffel, M. Round, CS, F. Wissbrock. The 3-Loop Non-Singlet Heavy Flavor Contributions and Anomalous Dimensions for the Structure Function $F_2(x, Q^2)$ and Transversity. Nuclear Physics B 886, 2014. arXiv:1406.4654 [hep-ph].
- J. Ablinger, J. Blümlein, A. De Freitas, A. Hasselhuhn, A. von Manteuffel, M. Round, CS. The $O(\alpha_s^3 T_F^2)$ Contributions to the Gluonic Operator Matrix Element. Nuclear Physics B 885, 2014. arXiv:1405.4259 [hep-ph].
- J. Ablinger, J. Blümlein, A. De Freitas A. Hasselhuhn, A. von Manteuffel, M. Round, CS, F. Wissbrock. The Transition Matrix Element $A_{gq}(N)$ of the Variable Flavor Number Scheme at $O(\alpha_s^3)$. Nuclear Physics B 882, 2014. arXiv:1402.0359 [hep-ph].
- A. Behring, J. Blümlein, A. De Freitas, A. Hasselhuhn, A. von Manteuffel, CS. The $O(\alpha_s^3)$ Heavy Flavor Contributions to the Charged Current Structure Function $xF_3(x, Q^2)$ at Large Momentum Transfer. Physical Review D 92(114005), 2015. arXiv:1508.01449 [hep-ph].
- A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel, CS. The 3-Loop Non-Singlet Heavy Flavor Contributions to the Structure Function $g_1(x, Q^2)$ at Large Momentum Transfer. Nucl. Phys. B 897, 2015. arXiv:1504.08217 [hep-ph].
- J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel, CS. The 3-Loop Pure Singlet Heavy Flavor Contributions to the Structure Function $F_2(x, Q^2)$ and the Anomalous Dimension. Nuclear Physics B 890, 2015. arXiv:1409.1135.
- A. Behring, J. Blümlein, G. Falcioni, A. De Freitas, A. von Manteuffel, CS. The Asymptotic 3-Loop Heavy Flavor Corrections to the Charged Current Structure Functions $F_L^{W^+-W^-}(x, Q^2)$ and $F_2^{W^+-W^-}(x, Q^2)$. Physical Review D 94(11), 2016. arXiv:1609.06255 [hep-ph].
- J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel, CS. Calculating Three Loop Ladder and V-Topologies for Massive Operator Matrix Elements by Computer Algebra. Comput. Phys. Comm. 202, 2016. arXiv:1509.08324 [hep-ph].
- J. Ablinger, J. Blümlein, P. Marquard, N. Rana, CS. Heavy Quark Form Factors at Three Loops in the Planar Limit, 2018. arXiv:1804.07313 [hep-ph].
- J. Ablinger, A. Behring, J. Blümlein, G. Falcioni, A. De Freitas, P. Marquard, N. Rana, CS. The Heavy Quark Form Factors at Two Loops. Physical Review D, 2018. arXiv:1712.09889.

General strategy:

$\downarrow$ IBP methods

- Recursively defined coupled DE systems for unknown MIs $\hat{I}_i(x)$
- $\hat{D}(x) = \text{LinComb}(\hat{I}_1(x), \ldots, \hat{I}_u(x))$

$\downarrow$ solver for $\hat{I}_i(x) = \sum_{N=0}^{\infty} I_i(N) x^N$

$$I_i(N) = \varepsilon^{-3} F_{-3}(N) + \varepsilon^{-2} F_{-2}(N) + \varepsilon^{-1} F_{-1}(N) \underbrace{+\varepsilon^0 F_0(N) + \cdots + \varepsilon^6 F_6(N)}_{\text{expensive calculation}}$$

$\downarrow$ plug into $\hat{D}(x) = \sum_{N=0}^{\infty} D(N) x^N$

$$D(N) = \varepsilon^{-3} D_{-3}(N) + \varepsilon^{-2} D_{-2}(N) + \varepsilon^{-1} D_{-1}(N) + \varepsilon^0 D_0(N) + \ldots$$

General strategy:

$\downarrow$ IBP methods

- Recursively defined coupled DE systems for unknown MIs $\hat{I}_i(x)$
- $\hat{D}(x) = \text{LinComb}(\hat{I}_1(x), \ldots, \hat{I}_u(x))$

$\downarrow$ solver for $\hat{I}_i(x) = \sum_{N=0}^{\infty} I_i(N) x^N$

$$I_i(N) = \varepsilon^{-3} F_{-3}(N) + \varepsilon^{-2} F_{-2}(N) + \varepsilon^{-1} F_{-1}(N) \underbrace{+\varepsilon^{0} F_0(N) + \cdots + \varepsilon^{6} F_6(N)}_{\text{often unknown functions}}$$

$\downarrow$ plug into $\hat{D}(x) = \sum_{N=0}^{\infty} D(N) x^N$

$$D(N) = \varepsilon^{-3} D_{-3}(N) + \varepsilon^{-2} D_{-2}(N) + \varepsilon^{-1} D_{-1}(N) + \varepsilon^{0} D_0(N) + \ldots$$

General strategy:

$\downarrow$ IBP methods

- Recursively defined coupled DE systems for unknown MIs $\hat{I}_i(x)$
- $\hat{D}(x) = \text{LinComb}(\hat{I}_1(x), \ldots, \hat{I}_u(x))$

$\downarrow$ solver for $\hat{I}_i(x) = \sum_{N=0}^{\infty} I_i(N) x^N$

$$I_i(N) = \varepsilon^{-3} F_{-3}(N) + \varepsilon^{-2} F_{-2}(N) + \varepsilon^{-1} F_{-1}(N) \underbrace{+ \varepsilon^0 F_0(N) + \cdots + \varepsilon^6 F_6(N)}_{\text{often unknown functions}}$$

$\downarrow$ plug into $\hat{D}(x) = \sum_{N=0}^{\infty} D(N) x^N$

$$D(N) = \underbrace{\varepsilon^{-3} D_{-3}(N) + \varepsilon^{-2} D_{-2}(N) + \varepsilon^{-1} D_{-1}(N)}_{\text{often nice}} + \underbrace{\varepsilon^0 D_0(N)}_{\text{partially nice}} + \ldots$$

Computing large moments and guessing recurrences

The DE-REC approach (SolveCoupledSystem package)

DE system
$$D_x \hat{I}(x) = A\,\hat{I}(x) + \hat{R}(x)$$

OreSys package (S. Gerhold)

uncoupling algorithm

uncoupled DE system
$$\sum_i d_i(x) D_x^i \hat{I}_1(x) = \hat{r}(x)$$
$$\mathbf{\hat{I}_k(x)} = \text{expr}_{\mathbf{k}}(\mathbf{D_x^i \hat{I}_1(x)}), \mathbf{k > 1}$$

$$\hat{I}_1(x) = \sum_{N=0}^{\infty} I_1(N) x^N$$

extract coefficient

SumProduction package

coeff. comparison w.r.t. x^N

linear recurrence
$$\sum_i a_i(N) I_1(N+i) = r(N)$$

`Sigma's` REC-solver

closed form solutions of $I_1(N), \mathbf{I_2(N)}, \dots, \mathbf{I_n(N)}$ in the class of nested sums and products – if this is possible

J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel, CS. Comput. Phys. Comm. 202, pp. 33-112.

The method of large moments (SolveCoupledSystem)

DE system

$$D_x \hat{I}(x) = A\,\hat{I}(x) + \hat{R}(x)$$

OreSys package (S. Gerhold)

uncoupling algorithm →

uncoupled DE system

$$\sum_i d_i(x) D_x^i \hat{I}_1(x) = \hat{r}(x)$$

$$\hat{I}_k(x) = \text{expr}_k(D_x^i \hat{I}_1(x)), k > 1$$

$$\hat{I}_1(x) = \sum_{N=0}^{\infty} I_1(N) x^N$$

extract coefficient

SumProduction package

coeff. comparison w.r.t. x^N

linear recurrence

$$\sum_i a_i(N) I_1(N+i) = r(N)$$

`Sigma`'s REC-solver ←

closed form solutions of $I_1(N)$ in the class of nested sums and products

– if th

Given $r(0), r(1), \ldots, r(8000)$,
compute $I_1(0), I_1(1), \ldots, I_1(8000)$

J. Blümlein, CS, Physics Letters B 771, pp. 31-36., 2017; arXiv:1701.04614 [hep-ph].

General strategy:

$$\Big\downarrow \text{IBP methods}$$

- Recursively defined coupled DE systems for unknown MIs $\hat{I}_i(x)$
- $\hat{D}(x) = \text{LinComb}(\hat{I}_1(x), \dots, \hat{I}_u(x))$

$$\Big\downarrow \text{solver for } \hat{I}_i(x) = \sum_{N=0}^{\infty} I_i(N) x^N$$

$$I_i(N) = \varepsilon^{-3} F_{-3}(N) + \varepsilon^{-2} F_{-2}(N) + \varepsilon^{-1} F_{-1}(N) \underbrace{+\varepsilon^0 F_0(N) + \dots + \varepsilon^6 F_6(N)}_{\text{only numbers}}$$

$N = 0, 1, \dots, 8000$

$$\Big\downarrow \text{plug into } \hat{D}(x) = \sum_{N=0}^{\infty} D(N) x^N$$

$$D(N) = \underbrace{\varepsilon^{-3} D_{-3}(N) + \varepsilon^{-2} D_{-2}(N) + \varepsilon^{-1} D_{-1}(N)}_{\text{numbers}} + \underbrace{\varepsilon^0 D_0(N)}_{\text{numbers}} + \dots$$

$N = 0, 1, \dots, 8000$

coupled systems for $D(N)$

`SolveCoupledSystem.m` ↓

large no. of moments

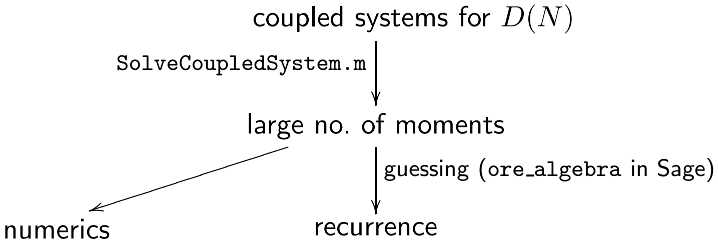
coupled systems for $D(N)$

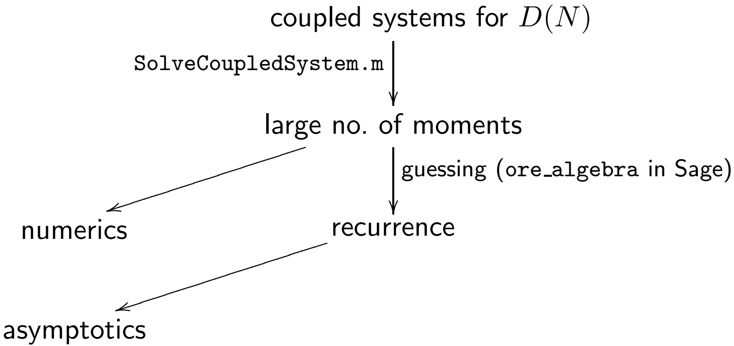
↓ `SolveCoupledSystem.m`

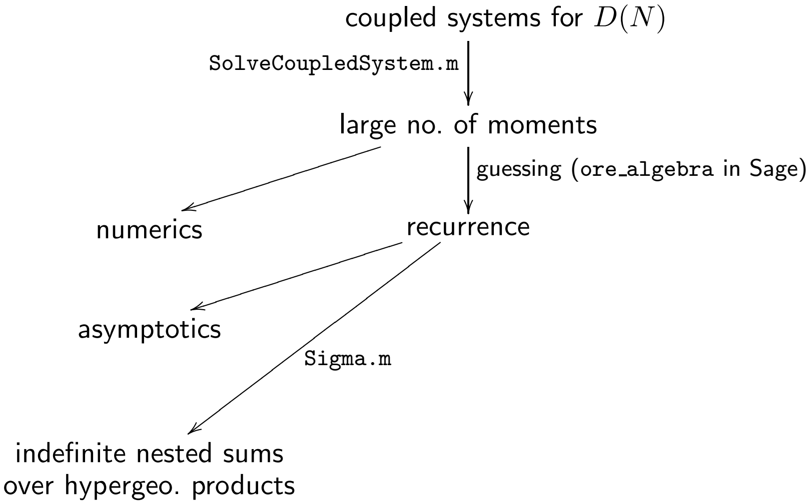
large no. of moments

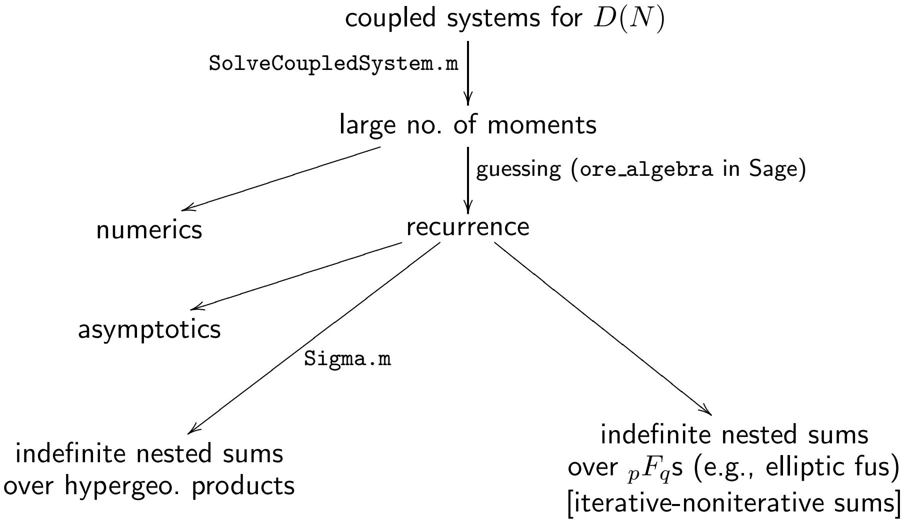
↙

numerics


coupled systems for $D(N)$
SolveCoupledSystem.m
large no. of moments
guessing (ore_algebra in Sage)
numerics
recurrence


coupled systems for $D(N)$
SolveCoupledSystem.m
large no. of moments
guessing (ore_algebra in Sage)
numerics
recurrence
asymptotics


coupled systems for $D(N)$
SolveCoupledSystem.m
large no. of moments
guessing (ore_algebra in Sage)
numerics
recurrence
asymptotics
Sigma.m
indefinite nested sums
over hypergeo. products


coupled systems for $D(N)$
SolveCoupledSystem.m
large no. of moments
guessing (ore_algebra in Sage)
numerics
recurrence
asymptotics
Sigma.m
indefinite nested sums
over hypergeo. products
indefinite nested sums
over ${}_pF_q$s (e.g., elliptic fus)
[iterative-noniterative sums]

coupled systems for $D(N)$

↓ `SolveCoupledSystem.m`

large no. of moments

↓ guessing (`ore_algebra` in Sage)

recurrence

↓ `Sigma.m`

indefinite nested sums
over hypergeo. products

General strategy:

$$\Big\downarrow \text{IBP methods}$$

- Recursively defined coupled DE systems for unknown MIs $\hat{I}_i(x)$
- $\hat{D}(x) = \text{LinComb}(\hat{I}_1(x), \dots, \hat{I}_u(x))$

$$\Big\downarrow \text{solver for } \hat{I}_i(x) = \sum_{N=0}^{\infty} I_i(N) x^N$$

$$I_i(N) = \varepsilon^{-3} F_{-3}(N) + \varepsilon^{-2} F_{-2}(N) + \varepsilon^{-1} F_{-1}(N) \underbrace{+\varepsilon^0 F_0(N) + \dots + \varepsilon^6 F_6(N)}_{\text{only numbers}}$$

$N = 0, 1, \dots, 8000$

$$\Big\downarrow \text{plug into } \hat{D}(x) = \sum_{N=0}^{\infty} D(N) x^N$$

$$D(N) = \underbrace{\varepsilon^{-3} D_{-3}(N) + \varepsilon^{-2} D_{-2}(N) + \varepsilon^{-1} D_{-1}(N)}_{\text{numbers}} + \underbrace{\varepsilon^0 D_0(N)}_{\text{numbers}} + \dots$$

$N = 0, 1, \dots, 8000$

General strategy:

$\downarrow$ IBP methods

- Recursively defined coupled DE systems for unknown MIs $\hat{I}_i(x)$
- $\hat{D}(x) = \text{LinComb}(\hat{I}_1(x), \dots, \hat{I}_u(x))$

$\downarrow$ solver for $\hat{I}_i(x) = \sum_{N=0}^{\infty} I_i(N) x^N$

$$I_i(N) = \varepsilon^{-3} F_{-3}(N) + \varepsilon^{-2} F_{-2}(N) + \varepsilon^{-1} F_{-1}(N) \underbrace{+ \varepsilon^0 F_0(N) + \dots + \varepsilon^6 F_6(N)}_{\text{only numbers}}$$

$\downarrow$ plug into $\hat{D}(x) = \sum_{N=0}^{\infty} D(N) x^N$

$$D(N) = \underbrace{\varepsilon^{-3} D_{-3}(N) + \varepsilon^{-2} D_{-2}(N) + \varepsilon^{-1} D_{-1}(N)}_{\text{nice}} + \underbrace{\varepsilon^0 D_0(N)}_{\text{partially nice}} + \dots$$

all N solution

Concrete calculations of large moments:

- The three-loop splitting functions $P_{qg}^{(2)}$ and $P_{gg}^{(2,\mathrm{N_F})}$ [Nucl.Phys.B./2017]
  1. computed ~ 2400 moments
  2. guessed all recurrences
  3. solved all recurrences in terms of harmonic sums.

Concrete calculations of large moments:

- The three-loop splitting functions $P_{qg}^{(2)}$ and $P_{gg}^{(2,\mathrm{N_F})}$ [Nucl.Phys.B./2017]
  1. computed ~ 2400 moments
  2. guessed all recurrences
  3. solved all recurrences in terms of harmonic sums.

- The massive Wilson coefficient A_{Qg}:
  1. computed 2000 moments
  2. guessed and solved some recurrences.

Concrete calculations of large moments:

- The three-loop splitting functions $P_{qg}^{(2)}$ and $P_{gg}^{(2,\mathrm{N_F})}$ [Nucl.Phys.B./2017]
  1. computed ~ 2400 moments
  2. guessed all recurrences
  3. solved all recurrences in terms of harmonic sums.

- The massive Wilson coefficient A_{Qg}:
  1. computed 2000 moments
  2. guessed and solved some recurrences.

- The second calculation of the polarized three–loop splitting functions in the M–scheme (Moch/Vermaseren/Vogt 2014)
  1. computed 6000 moments
  2. guessed all recurrences and solved them

  $\to$ Johannes Blümlein's talk (this afternoon)

Concrete calculations of large moments:

- The three-loop splitting functions $P_{qg}^{(2)}$ and $P_{gg}^{(2,\mathrm{N_F})}$ [Nucl.Phys.B./2017]
  1. computed ~ 2400 moments
  2. guessed all recurrences
  3. solved all recurrences in terms of harmonic sums.

- The massive Wilson coefficient A_{Qg}:
  1. computed 2000 moments
  2. guessed and solved some recurrences.

- The second calculation of the polarized three–loop splitting functions in the M–scheme (Moch/Vermaseren/Vogt 2014)
  1. computed 6000 moments
  2. guessed all recurrences and solved them

  $\rightarrow$ Johannes Blümlein's talk (this afternoon)

- The heavy fermion contributions to the massive three loop form factors

  $\rightarrow$ Peter Marquard's talk (yesterday)

  Requires a refined large moment machinery

Refinements of the large moment method

Basic approach

$$\Big\downarrow \text{uncoupling}$$

$$d_0(x,\varepsilon)\hat{I}_1(x,\varepsilon) + d_1(x,\varepsilon)D_x\hat{I}_1(x,\varepsilon) + \cdots + d_\lambda(x,\varepsilon)D_x^\lambda\hat{I}_1(x,\varepsilon) = r(x,\varepsilon)$$

$$\Big\downarrow$$

$$a_0(N,\varepsilon)I_1(\varepsilon,N)+a_1(x,\varepsilon)I_1(\varepsilon,N+1)+\cdots+a_\delta(\varepsilon,N)I_1(\varepsilon,N+\delta) = h(\varepsilon,N)$$

$$\Big\downarrow \delta \text{ initial values}$$

$$I_1(\varepsilon,N) = F_{-3}(N)\,\varepsilon^{-3} + F_{-2}(N)\,\varepsilon^{-2} + F_{-1}(N)\,\varepsilon^{-1} + \dots$$

for $N = 0, 1, \dots, 8000$

Basic approach

$$\Big\downarrow \text{uncoupling}$$

$$d_0(x,\varepsilon)\hat{I}_1(x,\varepsilon) + d_1(x,\varepsilon)D_x\hat{I}_1(x,\varepsilon) + \dots + d_\lambda(x,\varepsilon)D_x^\lambda\hat{I}_1(x,\varepsilon) = r(x,\varepsilon)$$

$$\Big\downarrow$$

$$a_0(N,\varepsilon)I_1(\varepsilon,N)+a_1(x,\varepsilon)I_1(\varepsilon,N+1)+\dots+a_\delta(\varepsilon,N)I_1(\varepsilon,N+\delta) = h(\varepsilon,N)$$

$$\boxed{\delta \leq \lambda + \max_{0\leq i\leq\lambda} \deg_x(d_i)}$$

$$\Big\downarrow \delta \text{ initial values}$$

$$I_1(\varepsilon,N) = F_{-3}(N)\,\varepsilon^{-3} + F_{-2}(N)\,\varepsilon^{-2} + F_{-1}(N)\,\varepsilon^{-1} + \dots$$

for $N = 0, 1, \dots, 8000$

Naive improvement

Compute

$$g(x,\varepsilon) = \operatorname*{gcd_x}_{0\le i\le\lambda} d_i(x,\varepsilon)$$

$$d_0(x,\varepsilon)\hat{I}_1(x,\varepsilon) + d_1(x,\varepsilon)D_x\hat{I}_1(x,\varepsilon) + \cdots + d_\lambda(x,\varepsilon)D_x^\lambda\hat{I}_1(x,\varepsilon) = r(x,\varepsilon)$$

$$\downarrow$$

$$a_0(N,\varepsilon)I_1(\varepsilon,N) + a_1(x,\varepsilon)I_1(\varepsilon,N+1) + \cdots + a_\delta(\varepsilon,N)I_1(\varepsilon,N+\delta) = h(\varepsilon,N)$$

$$\boxed{\delta \le \lambda + \max_{0\le i\le\lambda} \deg_x(d_i)}$$

$\downarrow$ δ initial values

$$I_1(\varepsilon,N) = F_{-3}(N)\,\varepsilon^{-3} + F_{-2}(N)\,\varepsilon^{-2} + F_{-1}(N)\,\varepsilon^{-1} + \ldots$$

for $N = 0, 1, \ldots, 8000$

Naive improvement

Compute

$$g(x,\varepsilon) = \underset{0\le i\le\lambda}{\mathrm{gcd}_x}\, d_i(x,\varepsilon)$$

$$\frac{d_0(x,\varepsilon)}{g(x,\varepsilon)}\hat{I}_1(x,\varepsilon) + \frac{d_1(x,\varepsilon)}{g(x,\varepsilon)}D_x\hat{I}_1(x,\varepsilon) + \cdots + \frac{d_\lambda(x,\varepsilon)}{g(x,\varepsilon)}D_x^\lambda\hat{I}_1(x,\varepsilon) = \frac{r(x,\varepsilon)}{g(x,\varepsilon)}$$

$$\downarrow$$

$$a_0(N,\varepsilon)I_1(\varepsilon,N)+a_1(x,\varepsilon)I_1(\varepsilon,N+1)+\cdots+a_\delta(\varepsilon,N)I_1(\varepsilon,N+\delta) = h(\varepsilon,N)$$

$$\boxed{\delta \le \lambda + \max_{0\le i\le\lambda} \deg_x(d_i) - \deg(g)}$$

$$\downarrow \; \delta \text{ initial values}$$

$$I_1(\varepsilon,N) = F_{-3}(N)\,\varepsilon^{-3} + F_{-2}(N)\,\varepsilon^{-2} + F_{-1}(N)\,\varepsilon^{-1} + \ldots$$

for $N = 0,1,\ldots,8000$

Major improvement

Compute

$$g(x) = \gcd_x\limits_{0\le i\le\lambda} d_i(x, 0)$$

$$\frac{d_0(x,\varepsilon)}{g(x)}\hat{I}_1(x,\varepsilon) + \frac{d_1(x,\varepsilon)}{g(x)}D_x\hat{I}_1(x,\varepsilon) + \cdots + \frac{d_\lambda(x,\varepsilon)}{g(x)}D_x^\lambda\hat{I}_1(x,\varepsilon) = \frac{r(x,\varepsilon)}{g(x)}$$

$$\downarrow$$

$$a_0(N)F_{-3}(N) + a_1(x)F_{-3}(N+1) + \cdots + a_\delta(N)F_{-3}(N+\delta) = h(N)$$

$$\boxed{\delta \le \lambda + \max_{0\le i\le\lambda}\deg_x(d_i) - \deg(g)}$$

$\downarrow$ δ initial values

$$I_1(\varepsilon, N) = F_{-3}(N)\,\varepsilon^{-3} + F_{-2}(N)\,\varepsilon^{-2} + F_{-1}(N)\,\varepsilon^{-1} + \ldots$$

for $N = 0, 1, \ldots, 8000$

Ex: $(\lambda = 4)$

$$\delta = 17 \to \delta = 4$$

Major improvement

Compute

$$g(x) = \gcd_{x} \, \underset{0\le i\le\lambda}{} d_i(x, 0)$$

$$\frac{d_0(x,\varepsilon)}{g(x)}\hat{I}_1(x,\varepsilon) + \frac{d_1(x,\varepsilon)}{g(x)}D_x\hat{I}_1(x,\varepsilon) + \cdots + \frac{d_\lambda(x,\varepsilon)}{g(x)}D_x^\lambda\hat{I}_1(x,\varepsilon) = \frac{r(x,\varepsilon)}{g(x)}$$

$$\downarrow$$

$$a_0(N)F_{-3}(N) + a_1(x)F_{-3}(N+1) + \cdots + a_\delta(N)F_{-3}(N+\delta) = h(N)$$

$$\boxed{\delta \le \lambda + \max_{0\le i\le\lambda} \deg_x(d_i) - \deg(g)}$$

$\downarrow$ δ initial values

$$I_1(\varepsilon, N) = F_{-3}(N)\,\varepsilon^{-3} + F_{-2}(N)\,\varepsilon^{-2} + F_{-1}(N)\,\varepsilon^{-1} + \ldots$$

for $N = 0, 1, \ldots, 8000$

Ex: $(\lambda = 4)$

$$\delta = 17 \to \delta = 4$$

Major improvement

Compute

$$g(x) = \gcd_{x\,0\le i\le\lambda} d_i(x,0)$$

$$\frac{d_0(x,\varepsilon)}{g(x)}\hat{I}_1(x,\varepsilon) + \frac{d_1(x,\varepsilon)}{g(x)}D_x\hat{I}_1(x,\varepsilon) + \cdots + \frac{d_\lambda(x,\varepsilon)}{g(x)}D_x^\lambda\hat{I}_1(x,\varepsilon) = \frac{r(x,\varepsilon)}{g(x)}$$

$\downarrow$

$$a_0(N)F_{-3}(N) + a_1(x)F_{-3}(N+1) + \cdots + a_\delta(N)F_{-3}(N+\delta) = h(N)$$

$$\boxed{\delta \le \lambda + \max_{0\le i\le\lambda} \deg_x(d_i) - \deg(g)}$$

$\downarrow$ δ initial values

$$I_1(\varepsilon,N) = F_{-3}(N)\,\varepsilon^{-3} + F_{-2}(N)\,\varepsilon^{-2} + F_{-1}(N)\,\varepsilon^{-1} + \dots$$

for $N = 0, 1, \dots, 8000$

Iterate to get $F_{-2}(N), F_{-1}(N), \dots$

Ex: $(\lambda = 4)$

$$\delta = 17 \to \delta = 4$$

Strategy 1: uncouple with ε:

$$D_x \begin{pmatrix} \hat{I}_1(x,\varepsilon) \\ \dots \\ \hat{I}_\lambda(x,\varepsilon) \end{pmatrix} = A(x,\varepsilon) \begin{pmatrix} \hat{I}_1(x,\varepsilon) \\ \dots \\ \hat{I}_\lambda(x,\varepsilon) \end{pmatrix} + \begin{pmatrix} \hat{R}_1(x,\varepsilon) \\ \dots \\ \hat{R}_\lambda(x,\varepsilon) \end{pmatrix}$$

↓ uncoupling is too hard for $\lambda \geq 5$

$$d_0(x,\varepsilon)\hat{I}_1(x,\varepsilon) + d_1(x,\varepsilon)D_x\hat{I}_1(x,\varepsilon) + \dots + d_\lambda(x,\varepsilon)D_x^\lambda\hat{I}_1(x,\varepsilon) = \hat{r}(x,\varepsilon)$$

↓ compute large moments (see above)

$$I_1(\varepsilon,N) = F_{-3}(N)\,\varepsilon^{-3} + F_{-2}(N)\,\varepsilon^{-2} + F_{-1}(N)\,\varepsilon^{-1} + \dots$$

Strategy 2: uncouple without ε:

$$D_x \begin{pmatrix} \hat{F}_{-3}(x) \\ \dots \\ \hat{F}_{-3}(x) \end{pmatrix} = A(x, 0) \begin{pmatrix} \hat{F}_{-3}(x) \\ \dots \\ \hat{F}_{-3}(x) \end{pmatrix} + \begin{pmatrix} \hat{R}_{-3,1}(x) \\ \dots \\ \hat{R}_{-3,\lambda}(x) \end{pmatrix}$$

$\downarrow$ uncoupling

$$d_0(x)\hat{F}_{-3}(x) + d_1(x)D_x\hat{F}_{-3}(x) + \dots + d_\lambda(x)D_x^\lambda\hat{F}_{-3}(x) = \hat{r}(x)$$

$\downarrow$ compute large moments (see above)

$$I_1(\varepsilon, N) = F_{-3}(N)\,\varepsilon^{-3} + F_{-2}(N)\,\varepsilon^{-2} + F_{-1}(N)\,\varepsilon^{-1} + \dots$$

Strategy 2: uncouple without ε:

$$D_x \begin{pmatrix} \hat{I}_1(x,\varepsilon) \\ \dots \\ \hat{I}_\lambda(x,\varepsilon) \end{pmatrix} = A(x,\varepsilon) \begin{pmatrix} \hat{I}_1(x,\varepsilon) \\ \dots \\ \hat{I}_\lambda(x,\varepsilon) \end{pmatrix} + \begin{pmatrix} \hat{R}_1(x,\varepsilon) \\ \dots \\ \hat{R}_\lambda(x,\varepsilon) \end{pmatrix}$$

↓ uncoupling is too hard for $\lambda \geq 5$

$$d_0(x,\varepsilon)\hat{I}_1(x,\varepsilon) + d_1(x,\varepsilon)D_x\hat{I}_1(x,\varepsilon) + \dots + d_\lambda(x,\varepsilon)D_x^\lambda\hat{I}_1(x,\varepsilon) = \hat{r}(x,\varepsilon)$$

↓ compute large moments (see above)

$$I_1(\varepsilon,N) = F_{-3}(N)\,\varepsilon^{-3} + F_{-2}(N)\,\varepsilon^{-2} + F_{-1}(N)\,\varepsilon^{-1} + \dots$$

Strategy 2: uncouple without ε:

$$D_x \begin{pmatrix} \hat{I}_1(x,\varepsilon) \\ \dots \\ \hat{I}_\lambda(x,\varepsilon) \end{pmatrix} = A(x,\varepsilon) \begin{pmatrix} \hat{I}_1(x,\varepsilon) \\ \dots \\ \hat{I}_\lambda(x,\varepsilon) \end{pmatrix} + \begin{pmatrix} \hat{R}_1(x,\varepsilon) \\ \dots \\ \hat{R}_\lambda(x,\varepsilon) \end{pmatrix}$$

↓ uncoupling is too hard for $\lambda \geq 5$

$$d_0(x,\varepsilon)\hat{I}_1(x,\varepsilon) + d_1(x,\varepsilon)D_x\hat{I}_1(x,\varepsilon) + \dots + d_\lambda(x,\varepsilon)D_x^\lambda \hat{I}_1(x,\varepsilon) = \hat{r}(x,\varepsilon)$$

↓ compute large moments (see above)

$$I_1(\varepsilon, N) = F_{-3}(N)\,\varepsilon^{-3} + F_{-2}(N)\,\varepsilon^{-2} + F_{-1}(N)\,\varepsilon^{-1} + \dots$$

Iterate to get $F_{-2}(N), F_{-1}(N), \dots$

Heavy Fermion Contributions to the Massive Three Loop Form Factors

$\rightarrow$ Peter Marquard's talk [to appear in Nucl. Phys. B]

- 41 systems

number of systems	order	
16	$\lambda = 1$	Strategy 1
15	$\lambda = 2$	
2	$\lambda = 3$	
3	$\lambda = 4$	
3	$\lambda = 5$	Strategy 2
1	$\lambda = 6$	
1	$\lambda = 7$	

Heavy Fermion Contributions to the Massive Three Loop Form Factors

$\rightarrow$ Peter Marquard's talk [to appear in Nucl. Phys. B]

- 41 systems
- code has been parallelized (we used 16 Mathematica kernels)

Heavy Fermion Contributions to the Massive Three Loop Form Factors

$\rightarrow$ Peter Marquard's talk [to appear in Nucl. Phys. B]

- 41 systems
- code has been parallelized (we used 16 Mathematica kernels)
  - 2000 moments: 5 days (30 total CPU days)

    $\rightarrow$ all recurrences ($\leq$ order 17)

    except the ζ_3 and constant-free contributions of ε^0

Heavy Fermion Contributions to the Massive Three Loop Form Factors

→ Peter Marquard's talk [to appear in Nucl. Phys. B]

- 41 systems
- code has been parallelized (we used 16 Mathematica kernels)
  - 2000 moments: 5 days (30 total CPU days)
    → all recurrences ($\leq$ order 17)
    except the ζ_3 and constant-free contributions of ε^0
  - 4000 moments: 8 days (78 total CPU days)
    → recurrences for ζ_3 contributions of ε^0 ($\leq$ order 29)
    but not for the constant-free contributions

Heavy Fermion Contributions to the Massive Three Loop Form Factors

$\rightarrow$ Peter Marquard's talk [to appear in Nucl. Phys. B]

- 41 systems
- code has been parallelized (we used 16 Mathematica kernels)
  - 2000 moments: 5 days (30 total CPU days)

    $\rightarrow$ all recurrences ($\leq$ order 17)
    except the ζ_3 and constant-free contributions of ε^0
  - 4000 moments: 8 days (78 total CPU days)

    $\rightarrow$ recurrences for ζ_3 contributions of ε^0 ($\leq$ order 29)
    but not for the constant-free contributions
  - 6000 moments: 20 days (242 total CPU days)

    $\rightarrow$ no further recurrences

Heavy Fermion Contributions to the Massive Three Loop Form Factors

$\rightarrow$ Peter Marquard's talk [to appear in Nucl. Phys. B]

- 41 systems
- code has been parallelized (we used 16 Mathematica kernels)
  - 2000 moments: 5 days (30 total CPU days)
    $\rightarrow$ all recurrences ($\leq$ order 17)
    except the ζ_3 and constant-free contributions of ε^0
  - 4000 moments: 8 days (78 total CPU days)
    $\rightarrow$ recurrences for ζ_3 contributions of ε^0 ($\leq$ order 29)
    but not for the constant-free contributions
  - 6000 moments: 20 days (242 total CPU days)
    $\rightarrow$ no further recurrences
  - 8000 moments: 43 days (597 total CPU days)
    $\rightarrow$ recurrences for constant-free contributions ($\leq$ order 54)

General strategy:

↓ IBP methods

- Recursively defined coupled DE systems for unknown MIs $\hat{I}_i(x)$
- $\hat{D}(x) = \text{LinComb}(\hat{I}_1(x), \ldots, \hat{I}_u(x))$

↓ solver for $\hat{I}_i(x) = \sum_{N=0}^{\infty} I_i(N) x^N$

$$I_i(N) = \varepsilon^{-3} F_{-3}(N) + \varepsilon^{-2} F_{-2}(N) + \varepsilon^{-1} F_{-1}(N) \underbrace{+ \varepsilon^{0} F_{0}(N) + \cdots + \varepsilon^{6} F_{6}(N)}_{\text{only numbers}}$$

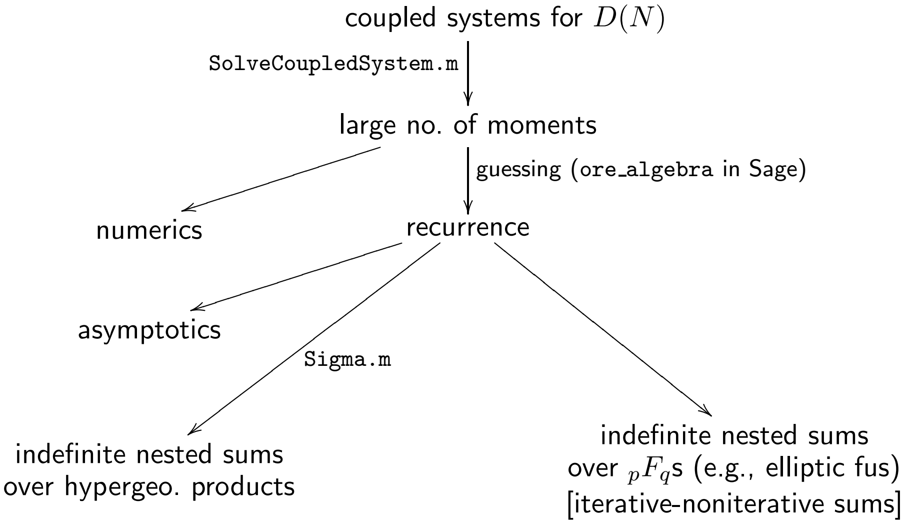
↓ plug into $\hat{D}(x) = \sum_{N=0}^{\infty} D(N) x^N$

$$D(N) = \underbrace{\varepsilon^{-3} D_{-3}(N) + \varepsilon^{-2} D_{-2}(N) + \varepsilon^{-1} D_{-1}(N)}_{\text{nice}} + \underbrace{\varepsilon^{0} D_{0}(N)}_{\text{partially nice}} + \ldots$$

all N solution

Conclusion

1. We presented the Large Moment Methods and its flexibility


coupled systems for $D(N)$
SolveCoupledSystem.m
large no. of moments
guessing (ore_algebra in Sage)
numerics
recurrence
asymptotics
Sigma.m
indefinite nested sums
over hypergeo. products
indefinite nested sums
over ${}_pF_q$s (e.g., elliptic fus)
[iterative-noniterative sums]

Conclusion

1. We presented the Large Moment Methods and its flexibility
2. Suitable for challenging problems

Concrete calculations of large moments:

- The three-loop splitting functions $P_{qg}^{(2)}$ and $P_{gg}^{(2,\mathrm{N_F})}$ [Nucl.Phys.B./2017]
  1. computed ~ 2400 moments
  2. guessed all recurrences
  3. solved all recurrences in terms of harmonic sums.

- The massive Wilson coefficient A_{Qg}:
  1. computed 2000 moments
  2. guessed and solved some recurrences.

- The second calculation of the polarized three–loop splitting functions in the M–scheme (Moch/Vermaseren/Vogt 2014)
  1. computed 6000 moments
  2. guessed all recurrences and solved them

  $\rightarrow$ Johannes Blümlein's talk (this afternoon)

- The heavy fermion contributions to the massive three loop form factors

  $\rightarrow$ Peter Marquard's talk (yesterday)

  Requires a refined large moment machinery

Conclusion

1. We presented the Large Moment Methods and its flexibility
2. Suitable for challenging problems
3. Refined versions (Strategies 1 and 2)

Conclusion

1. We presented the Large Moment Methods and its flexibility
2. Suitable for challenging problems
3. Refined versions (Strategies 1 and 2)
4. Relies on advanced technologies

Used Packages to solve DE systems

In[1]:= << **Sigma.m**

Sigma - A summation package by Carsten Schneider © RISC-Linz

In[2]:= << **HarmonicSums.m**

HarmonicSums by Jakob Ablinger © RISC-Linz

In[3]:= << **EvaluateMultiSums.m**

EvaluateMultiSums by Carsten Schneider © RISC-Linz

In[4]:= << **SumProduction.m**

SumProduction by Carsten Schneider © RISC-Linz

In[5]:= << **OreSys.m**

OreSys by Stefan Gerhold (optimized by C. Schneider) © RISC-Linz

In[6]:= << **SolveCoupledSystem.m**

SolveCoupledSystem by Carsten Schneider © RISC-Linz

Simple integrals: symbolic summation, Matad (M. Steinhauser), literature...
Guessing recurrences: OreAlgebra (M. Kauers)