DiffExp and Feynman parameter integrals

Martijn Hidding

Uppsala University

@ “High Precision for Hard Processes (HP2 2022)”

Based on work in collaboration with J. Usovitsch (arXiv:2206.14790)

21 September 2022

Introduction

Introduction

[Kotikov, 1991], [Remiddi, 1997]
[Gehrmann, Remiddi, 2000]

- Differential equations are a powerful approach to computing master integrals.
- The effectiveness of differential equation methods is especially striking when applied to polylogarithmic integral families that (often) admit an ϵ-factorized (canonical) basis.

[Henn, 2013]

- Numerical approaches to solving differential equations can be efficient, precise, and may extend to cases beyond multiple polylogarithms or elliptic generalizations thereof.

e.g.: [Lee, Smirnov, Smirnov, ’18], [Mandal, Zhao, ’19], [Moriello, ’19],
[Bonciani, Del Duca, Frellesvig, Henn, MH, Maestri, Moriello, Salvatori, Smirnov, ’19],
[MH ’20], [Abreu, Ita, Moriello, Page, Tschernow, Zeng ’20], [Liu, Ma, ’21]

- In this talk, I will review the iterative series expansion method for solving differential equations and present some recent developments.

Differential equations

- We consider a family of scalar Feynman integrals:

$$I_{a_1,\ldots,a_{n+m}} = \int \left(\prod_{i=1}^{l} d^d k_i \right) \frac{\prod_{i=n+1}^{n+m} N_i^{-a_i}}{\prod_{i=1}^{n} D_i^{a_i}}$$

$$d = d_{\text{int}} - 2\epsilon$$
$$D_i = -q_i^2 + m_i^2 - i\delta$$

and a basis of master integrals $\vec{I}$. Taking derivatives on kinematic invariants and masses, denoted x_j, and performing IBP reductions, we obtain:

[Kotikov, 1991], [Remiddi, 1997]
[Gehrmann, Remiddi, 2000]

$$\boxed{\partial_{x_i} \vec{I} = \mathbf{M}_{x_i}(\{x_j\}, \epsilon)\, \vec{I}}$$

- We aim to solve these differential equations. Since they are of Fuchsian type, they admit convergent (generalized) power series solutions

(See e.g. [1212.4389], [1411.0911]
[1702.04279])

Canonical differential equations

- In many cases the differential equations can be brought into a canonical form:

$$\frac{\partial \vec{B}}{\partial x_i} = \epsilon \frac{\partial \tilde{\mathbf{A}}}{\partial x_i} \vec{B}, \quad d\vec{B} = \epsilon d\tilde{\mathbf{A}} \vec{B}$$

[Henn, 2013]
See also:
[Lee, 1411.0911]
[Prausa, 1701.00725]
[Gituliar, Magerya, 1701.04269]
[Meyer, 1705.06252]
[Dlapa, Henn, Yan, 2002.02340]

- Consider a line: $$\begin{aligned} \gamma : [0,1] &\to \mathbb{C}^{|S|} \\ x &\mapsto (\gamma_{x_1}(x), \ldots, \gamma_{x_\kappa}(x)) \end{aligned}$$

- Then order-by-order we have: $$\vec{B}(x,\epsilon) = \sum_{j=-N}^{\infty} \vec{B}^{(j)}(x)\,\epsilon^j$$

The boundary conditions must still be determined in some way.

$$\boxed{\vec{B}^{(i)}(1) = \int_0^1 \mathbf{A}_x \vec{B}^{(i-1)} dx + \boxed{\vec{B}^{(i)}(0)}}$$

Series expansion methods

Series expansions (canonical basis)

- Let us expand the matrix as a power series: $\mathbf{A}_x = x^r \left[\sum_{p=0}^{n} \mathbf{C}_p x^p + \mathcal{O}\left(x^{n+1}\right) \right]$
- Using integration-by-parts, we can write:

$$\int \frac{1}{x} \log(x)^n = \frac{1}{n+1} \log(x)^{n+1} \qquad \int x^m \log(x)^n = x^{m+1} \sum_{j=1}^{n} c_j \log(x)^j \quad (\text{for } m \neq -1)$$

- Thus, all the integrations can be performed in terms of (generalized) series expansions:

$$B_j^{(k)}(x) = x^r \sum_{n=0}^{\infty} \sum_{m=0}^{k} c_{mn} x^n \log(x)^m, \quad c_{mn} \in \mathbb{C}, \quad 0 \geq r \in \mathbb{Q}$$

- We may similarly integrate non-canonical systems in terms of series expansions (but we leave out the details here.)

Series expansion method

- Set up a linear system of differential equations.
- Reduce multi-scale problems to single-scale by integrating along a one-dimensional contour.
- Split up the contour into multiple segments such that series expansions converge on each segment.
- Find series solutions of the integrals along each segment, and fix boundary conditions by matching neighboring segments.
- Cross thresholds by assigning $\pm i\delta$ to logarithms and algebraic roots in the solutions.

(History) Series expansions

- This strategy was demonstrated in [F. Moriello, 1907.13234] for the computation of planar integrals relevant to H+j production in QCD at NLO

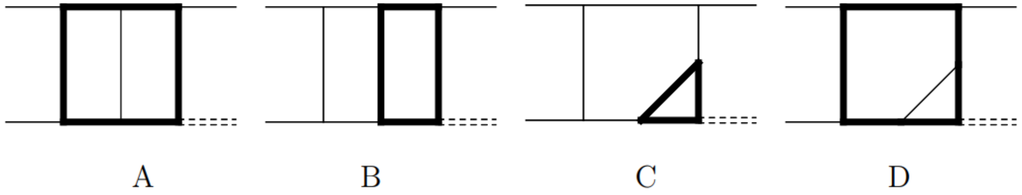
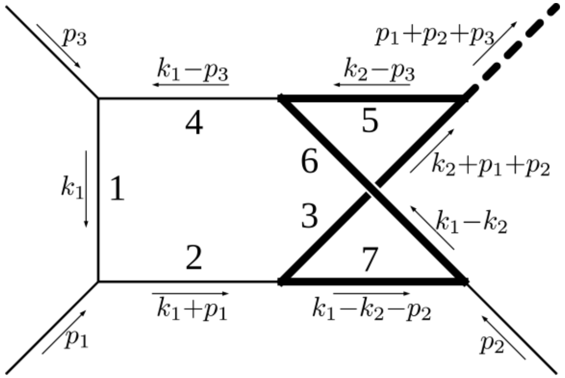


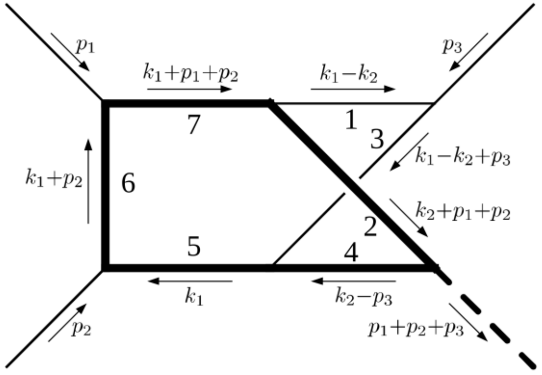
Figure 1: The four planar integral families contributing to two-loop $H+j$-production in QCD.

- Simultaneously, in a larger collaboration, we applied these methods to the computation of non-planar H+j integrals:

[R. Bonciani, V. Del Duca, H. Frellesvig, J. M. Henn, MH, L. Maestri, F. Moriello, G. Salvatori, V. A. Smirnov]

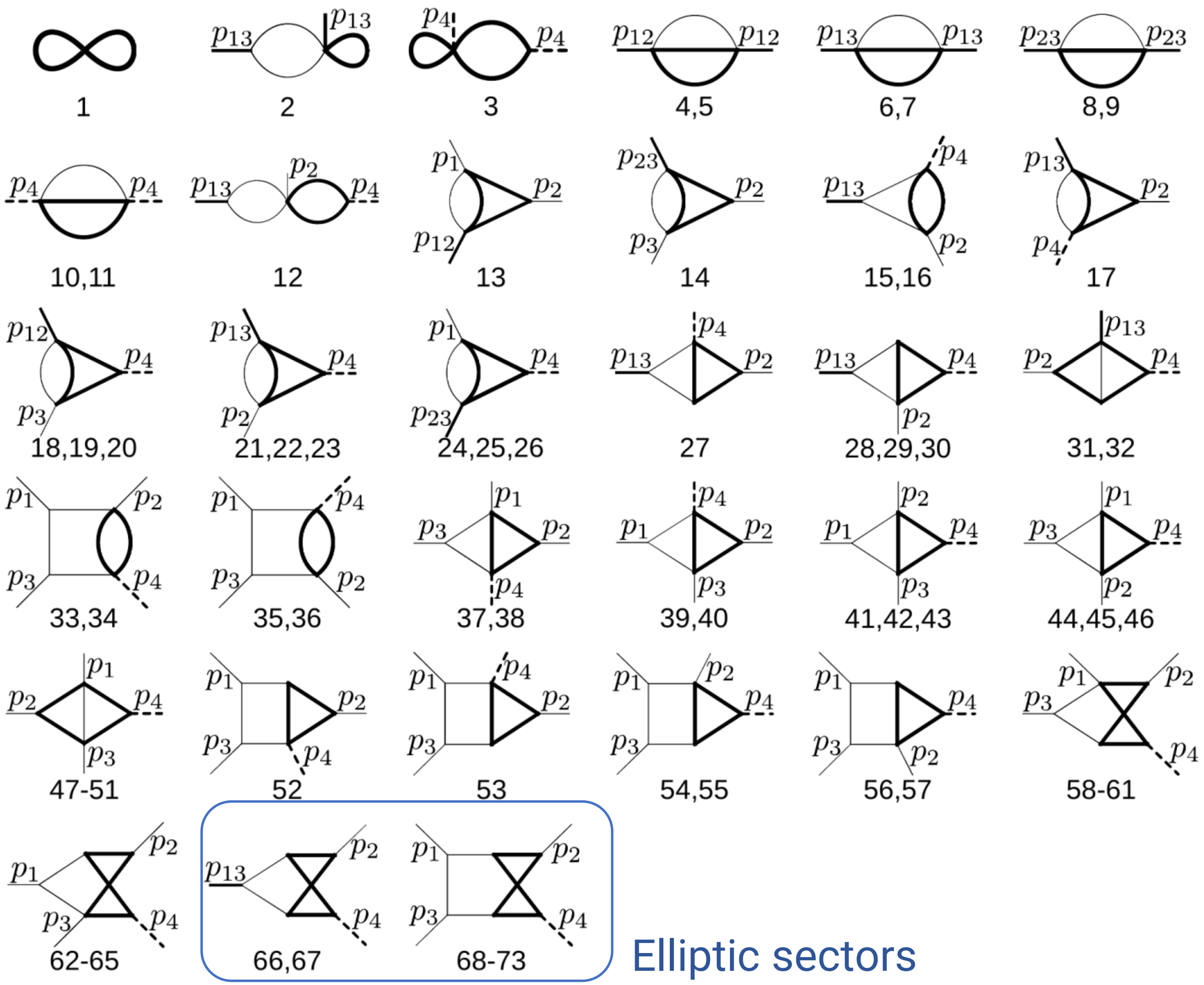


[Bonciani et al, 1907.13156]



[Frellesvig et al, 1911.06308]

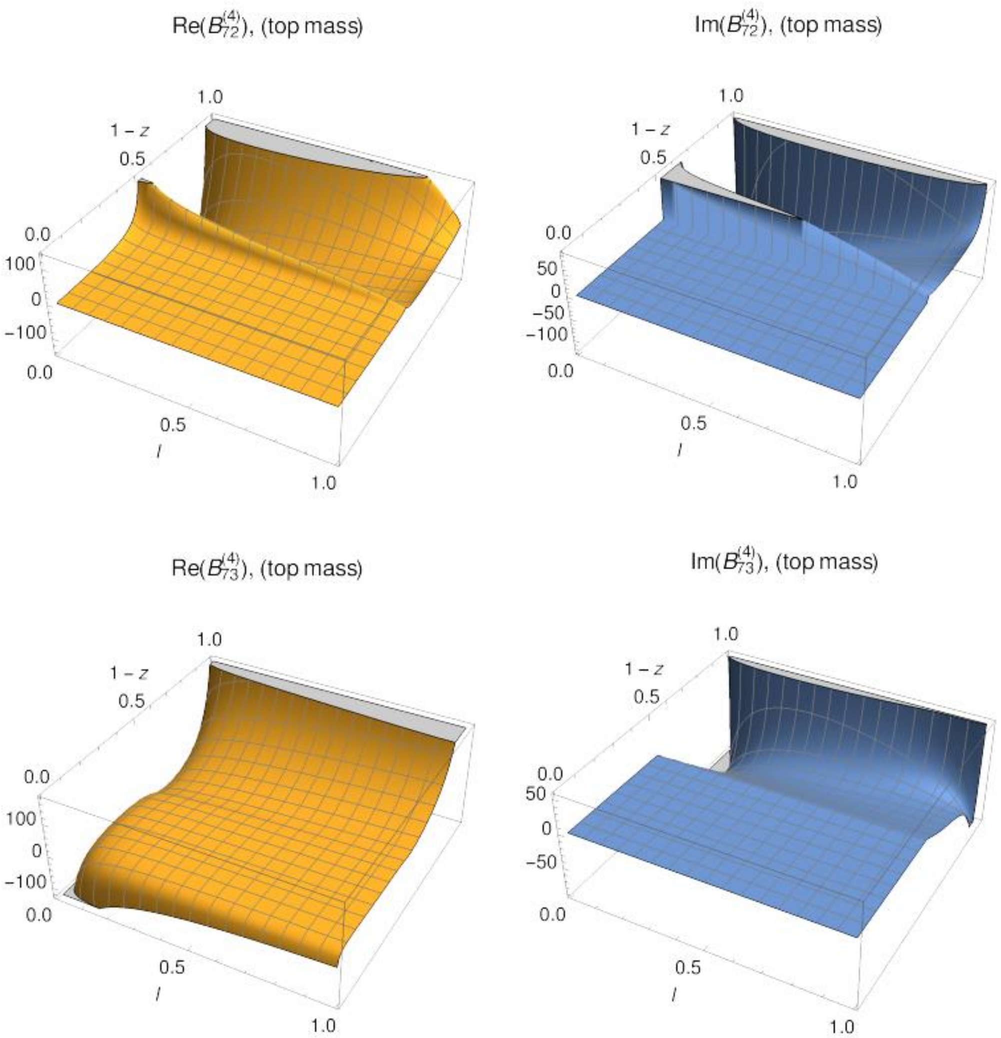
Example: H+j integrals (family F)



Elliptic sectors

Plots for family F

The real part of the integrals is in blue, the imaginary part is orange.



DiffExp

- A general implementation of these methods is implemented in the Mathematica package DiffExp, introduced in arXiv:2006.05510, (available at https://gitlab.com/hiddingm/diffexp)

- DiffExp accepts a system of differential equations of the form

$$\frac{\partial}{\partial s}\vec{f}(\{S\},\epsilon) = \mathbf{A}_s\vec{f}(\{S\},\epsilon) \qquad \mathbf{A}_x(x,\epsilon) = \sum_{k=0}^{\infty}\mathbf{A}_x^{(k)}(x)\epsilon^k$$

  for which the matrix entries are combinations of rational and algebraic functions

- It enables one to numerically integrate various multi-scale Feynman integrals at arbitrary points in phase-space, and at precisions of tens of digits (or higher)

- Various new packages are also showing up, e.g. SeaSyde and AMFlow, implementing new ideas and techniques.

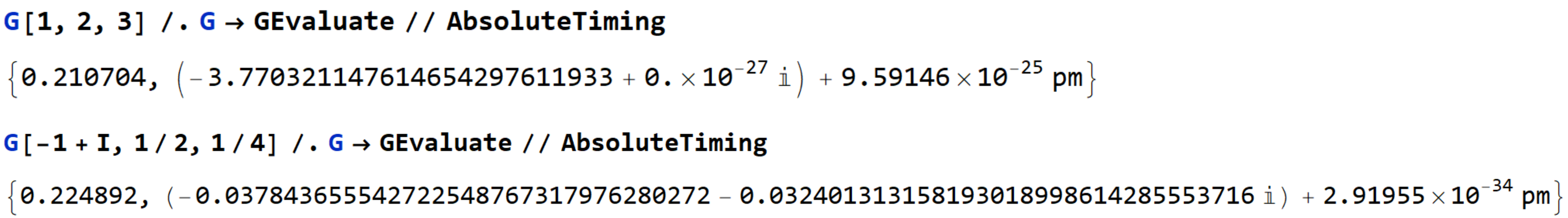
Special functions

$$\underbrace{G(a_1,\ldots,a_n;z)}_{\text{weight } n} = \int_0^z \frac{dt}{t-a_1} G(a_2,\ldots,a_n;t) \quad \text{for} \quad a_i, z \in \mathbb{C}$$

- Let us see how we can use DiffExp for evaluating MPLs. Note that:

$$\partial_z \begin{pmatrix} G(1,2;z) \\ G(2;z) \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{t-1} & 0 \\ 0 & 0 & \frac{1}{t-2} \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} G(1,2;z) \\ G(2;z) \\ 1 \end{pmatrix}$$

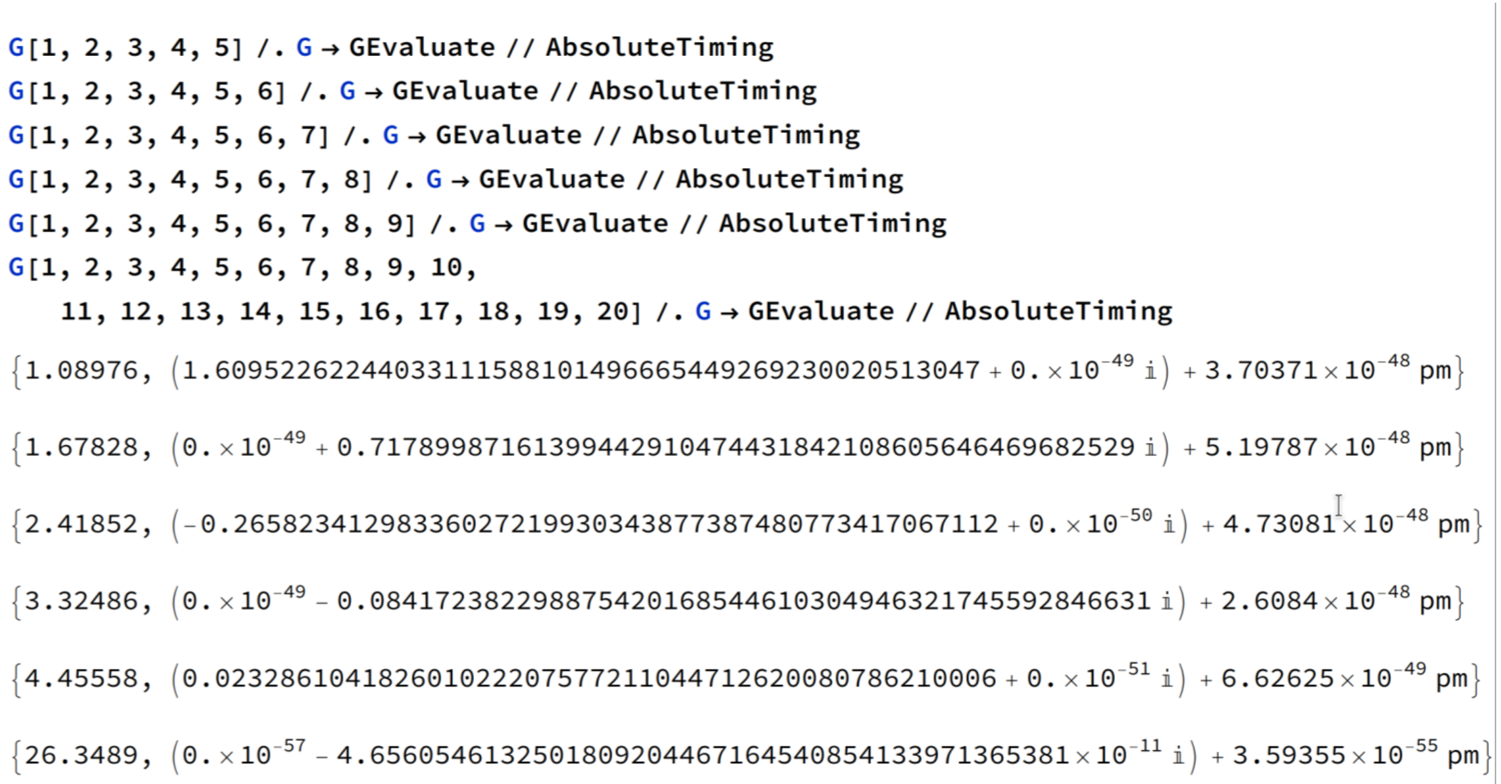
- For which the boundary conditions are $(0{,}0{,}1)$ at $z = 0$.
- After building a wrapper function, we can evaluate any MPL:



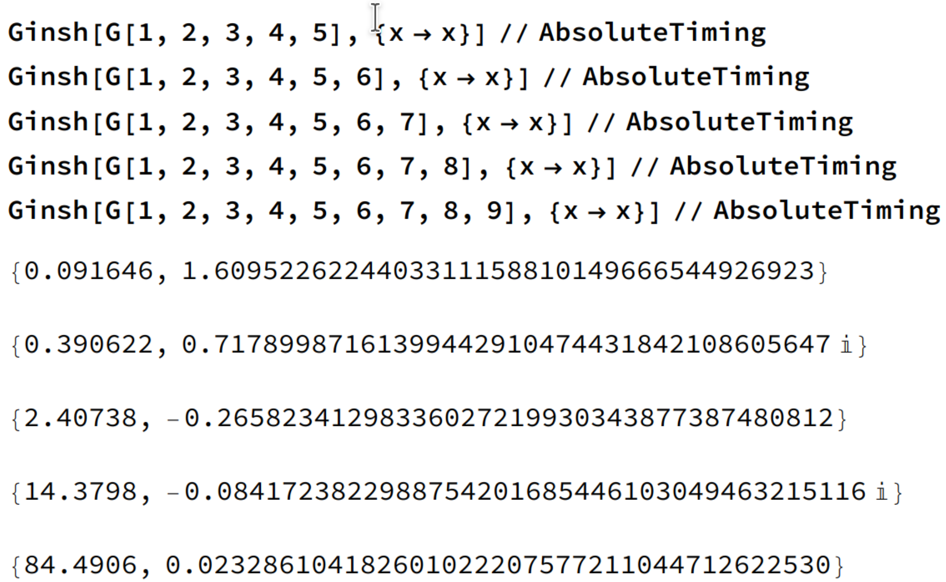
```
G[1, 2, 3] /. G → GEvaluate // AbsoluteTiming
{0.210704, (-3.77032114761465429761193 3 + 0. × 10^-27 ⅈ) + 9.59146 × 10^-25 pm}
G[-1 + I, 1 / 2, 1 / 4] /. G → GEvaluate // AbsoluteTiming
{0.224892, (-0.0378436555427225487673179762802 72 - 0.0324013131581930189986142855537 16 ⅈ) + 2.91955 × 10^-34 pm}
```

Special functions

- Under normal circumstances, the timing lacks behind *GiNaC*.
- But, in edge cases, we can beat *GiNaC*:



```
G[1, 2, 3, 4, 5] /. G → GEvaluate // AbsoluteTiming
G[1, 2, 3, 4, 5, 6] /. G → GEvaluate // AbsoluteTiming
G[1, 2, 3, 4, 5, 6, 7] /. G → GEvaluate // AbsoluteTiming
G[1, 2, 3, 4, 5, 6, 7, 8] /. G → GEvaluate // AbsoluteTiming
G[1, 2, 3, 4, 5, 6, 7, 8, 9] /. G → GEvaluate // AbsoluteTiming
G[1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
   11, 12, 13, 14, 15, 16, 17, 18, 19, 20] /. G → GEvaluate // AbsoluteTiming
{1.08976, (1.60952262244033111588101496665449269230020513047 + 0. × 10^-49 i) + 3.70371 × 10^-48 pm}

{1.67828, (0. × 10^-49 + 0.71789987161399442910474431842108605646469682529 i) + 5.19787 × 10^-48 pm}

{2.41852, (-0.26582341298336027219930343877387480773417067112 + 0. × 10^-50 i) + 4.73081 × 10^-48 pm}

{3.32486, (0. × 10^-49 - 0.084172382298875420168544610304946321745592846631 i) + 2.6084 × 10^-48 pm}

{4.45558, (0.023286104182601022207577211044712620080786210006 + 0. × 10^-51 i) + 6.62625 × 10^-49 pm}

{26.3489, (0. × 10^-57 - 4.6560546132501809204467164540854133971365381 × 10^-11 i) + 3.59355 × 10^-55 pm}
```



```
Ginsh[G[1, 2, 3, 4, 5], {x → x}] // AbsoluteTiming
Ginsh[G[1, 2, 3, 4, 5, 6], {x → x}] // AbsoluteTiming
Ginsh[G[1, 2, 3, 4, 5, 6, 7], {x → x}] // AbsoluteTiming
Ginsh[G[1, 2, 3, 4, 5, 6, 7, 8], {x → x}] // AbsoluteTiming
Ginsh[G[1, 2, 3, 4, 5, 6, 7, 8, 9], {x → x}] // AbsoluteTiming
{0.091646, 1.6095226224403311158810149666544926923}

{0.390622, 0.71789987161399442910474431842108605647 i}

{2.40738, -0.26582341298336027219930343877387480812}

{14.3798, -0.0841723822988754201685446103049463215116 i}

{84.4906, 0.023286104182601022207577211044712622530}
```

Special functions

- We can also evaluate generalized hypergeometric functions, such as the Appell functions. For example, we have with $F_1(x,y) \equiv F_1(a,b_1,b_2,c;x,y)$

$$x(1-x)\frac{\partial^2 F_1(x,y)}{\partial x^2} + y(1-x)\frac{\partial^2 F_1(x,y)}{\partial x \partial y} + [c-(a+b_1+1)x]\frac{\partial F_1(x,y)}{\partial x} - b_1 y\frac{\partial F_1(x,y)}{\partial y} - ab_1 F_1(x,y) = 0$$

$$y(1-y)\frac{\partial^2 F_1(x,y)}{\partial y^2} + x(1-y)\frac{\partial^2 F_1(x,y)}{\partial x \partial y} + [c-(a+b_2+1)y]\frac{\partial F_1(x,y)}{\partial y} - b_2 x\frac{\partial F_1(x,y)}{\partial x} - ab_2 F_1(x,y) = 0$$

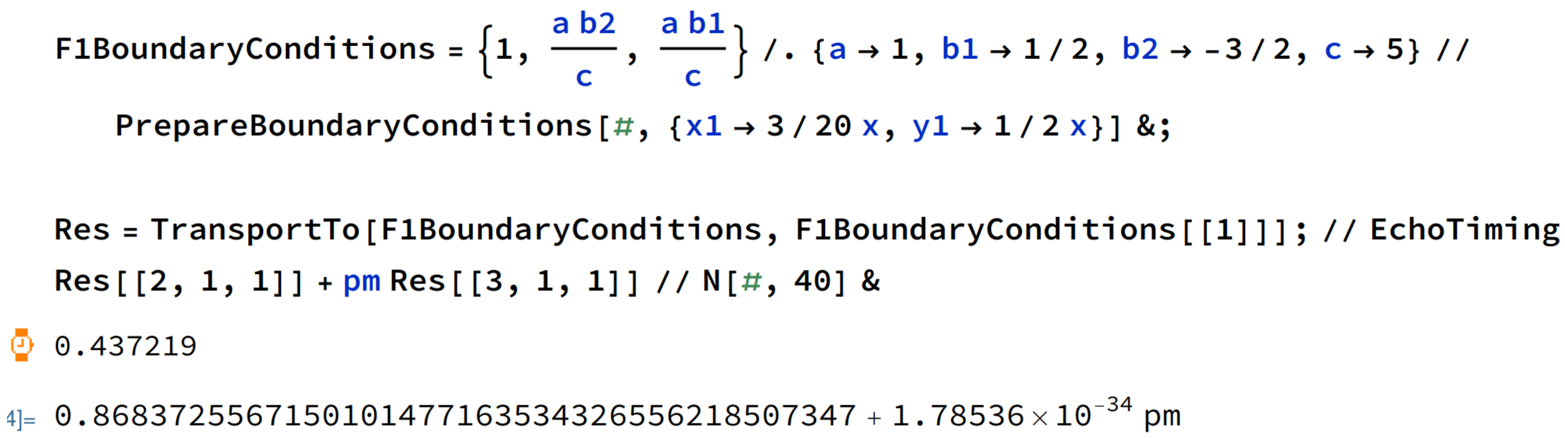
- This can be combined into:

$$\partial_x \begin{pmatrix} F_1(x,y) \\ \partial_y F_1(x,y) \\ \partial_x F_1(x,y) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -\frac{b_1}{x-y} & \frac{b_2}{x-y} \\ \frac{ab_1}{x-x^2} & \frac{(-1+y)yb_1}{(-1+x)x(x-y)} & -\frac{-c+x+ax+xb_1+\frac{(-1+x)yb_2}{x-y}}{(-1+x)x} \end{pmatrix} \begin{pmatrix} F_1(x,y) \\ \partial_y F_1(x,y) \\ \partial_x F_1(x,y) \end{pmatrix}$$

$$\partial_y \begin{pmatrix} F_1(x,y) \\ \partial_y F_1(x,y) \\ \partial_x F_1(x,y) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \frac{ab_2}{y-y^2} & \frac{(x-xy)b_1+(x-y)(-c+y+ay+yb_2)}{(-1+y)y(-x+y)} & \frac{(-1+x)xb_2}{(-1+y)y(-x+y)} \\ 0 & -\frac{b_1}{x-y} & \frac{b_2}{x-y} \end{pmatrix} \begin{pmatrix} F_1(x,y) \\ \partial_y F_1(x,y) \\ \partial_x F_1(x,y) \end{pmatrix}$$

Special functions

- Using the boundary conditions $(1, ab_2/c, ab_1/c)$ at $x = y = 0$, we may use DiffExp to evaluate the Appell F1 for arbitrary (real) x, y.
- For example, $F_1(1, 1/2, -3/2, 5; 3/20, 1/2)$:



```
F1BoundaryConditions = {1, a b2/c, a b1/c} /. {a → 1, b1 → 1/2, b2 → -3/2, c → 5} //
   PrepareBoundaryConditions[#, {x1 → 3/20 x, y1 → 1/2 x}] &;

Res = TransportTo[F1BoundaryConditions, F1BoundaryConditions[[1]]]; // EchoTiming
Res[[2, 1, 1]] + pm Res[[3, 1, 1]] // N[#, 40] &
0.437219
4]= 0.8683725567150101477163534326556218507347 + 1.78536 × 10^-34 pm
```

- Although the timing is not quite competitive with Mathematica’s inbuilt function, this approach is straightforward to generalize to other hypergeometric functions.

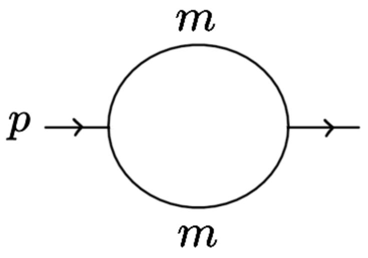
Boundary conditions

Boundary conditions

- In order to solve a system of differential equations, we need to provide boundary conditions at some appropriate kinematic point or limit.
- Various possibilities exist:
  - Analytic results using expansion by regions [See works by Beneke and Smirnov], [Jantzen, Smirnov, Smirnov, 1206.0546]
  - Determine boundary conditions by imposing the absence of pseudo-thresholds [See e.g. works by Henn]
  - Numerical boundary conditions for a finite basis using *pySecDec / FIESTA*
  - The auxiliary mass flow method and *AMFlow* [Liu, Ma, 2107.01864]
  - The "iterative Feynman trick" method discussed in this talk! [MH, J. Usovitsch, 2206.14790]
- Note that asymptotic limits have to be taken carefully in order to get consistent results in dimensional regularization.

Boundary conditions in asymptotic limits

- Typically, we consider a limit where most of the external scales vanish, such that the Feynman integrals simplify as much as possible.
- However, we can not in general commute the limit and the integration.
- Let’s consider the example of the massive bubble:



$$\frac{e^{\gamma_E \epsilon}}{i\pi^{1-\epsilon}} \int d^d k_1 \frac{1}{\left(-k_1^2+m^2\right)\left(-\left(k_1+p\right)^2+m^2\right)} = \frac{2\log\left(\frac{-\sqrt{-p^2}-\sqrt{4m^2-p^2}}{\sqrt{-p^2}-\sqrt{4m^2-p^2}}\right)}{\sqrt{-p^2}\sqrt{4m^2-p^2}} + \mathcal{O}(\epsilon)$$

- In the limit $m^2 = x$, with $x \downarrow 0$, we obtain:

$$\sim -\frac{2\left(\log\left(-p^2\right)-\log(x)\right)}{p^2} + \mathcal{O}(x)$$

Boundary conditions in asymptotic limits

- Suppose we took the limit inside the integrand. This yields:

$$e^{\gamma_E \epsilon}\left(i\pi^{d/2}\right)^{-1}\int d^d k_1 \frac{1}{\left(-k_1^2\right)\left(-\left(k_1+p\right)^2\right)} = \frac{2}{p^2\epsilon} - \frac{2\log\left(-p^2\right)}{p^2} + \mathcal{O}(\epsilon)$$

- The kinematic singularity has been transformed into a dimensionally regulated pole, yielding a different result than before!
- The situation becomes clearer if we consider the limit in closed form in ϵ:

$$\frac{e^{\gamma_E \epsilon}}{i\pi^{1-\epsilon}}\int d^d k_1 \frac{1}{\left(-k_1^2+m^2\right)\left(-\left(k_1+p\right)^2+m^2\right)} \sim -e^{\gamma_E \epsilon}\frac{\Gamma(\epsilon)}{p^2}\left(\epsilon\frac{\left(-p^2\right)^{-\epsilon}\Gamma(-\epsilon)^2}{\Gamma(-2\epsilon)} + 2x^{-\epsilon}\right)$$

- We reproduce either result by taking only the Taylor series part, or also including the term proportional to $x^{-\epsilon}$!

Boundary conditions

- The problem of finding boundary conditions numerically has been significantly advanced by the auxiliary mass flow method.

Auxiliary mass flow:
[Xiao Liu, Yan-Qing Ma, 1801.10523]
[Xiao Liu, Yan-Qing Ma, Wei Tao, Peng Zhang, 2009.07987]

AMFlow package:
[Xiao Liu, Yan-Qing Ma, 2201.11669]

- The central idea is to deform integrals by a complex mass:

$$I_{\vec{\nu}}(\epsilon) = \int \prod_{i=1}^{L} \frac{\mathrm{d}^D \ell_i}{\mathrm{i}\pi^{D/2}} \frac{\mathcal{D}_{K+1}^{-\nu_{K+1}} \cdots \mathcal{D}_N^{-\nu_N}}{(\mathcal{D}_1 + \mathrm{i}0^+)^{\nu_1} \cdots (\mathcal{D}_K + \mathrm{i}0^+)^{\nu_K}}$$

$$\Downarrow$$

$$I_{\vec{\nu}}^{\mathrm{mod}}(\epsilon, \eta) = \int \prod_{i=1}^{L} \frac{\mathrm{d}^D \ell_i}{\mathrm{i}\pi^{D/2}} \frac{\mathcal{D}_{K+1}^{-\nu_{K+1}} \cdots \mathcal{D}_N^{-\nu_N}}{(\mathcal{D}_1 + \lambda_1 \times \mathrm{i}\eta)^{\nu_1} \cdots (\mathcal{D}_K + \lambda_K \times \mathrm{i}\eta)^{\nu_K}}$$

- The original topology is recovered by: $I_{\vec{\nu}}(\epsilon) = \lim_{\eta \to 0^+} I_{\vec{\nu}}^{\mathrm{mod}}(\epsilon, \eta)$.
- And solved via: $\frac{\partial}{\partial \eta} \vec{\mathcal{I}}^{\mathrm{mod}}(\epsilon, \eta) = A(\epsilon, \eta) \vec{\mathcal{I}}^{\mathrm{mod}}(\epsilon, \eta)$

Direct integration

“Direct” integration

- Consider a scalar Feynman integral:

$$I_{\nu_1,\ldots,\nu_n}(\{s_i,m_i\},d) = \int \left(\prod_{j=1}^{l} \frac{d^d k_j}{i\pi^{\frac{d}{2}}}\right) \prod_{j=1}^{n} \frac{1}{D_j^{\nu_j}}$$

Where:

$$d = d_{\text{int}} - 2\epsilon$$

$$D_i = -q_i^2 + m_i^2 - i\delta$$

- A formula by Feynman tells us that:

$$\frac{1}{D_1^{\nu_1}\ldots D_n^{\nu_n}} = \frac{\Gamma(\nu)}{\Gamma(\nu_1)\ldots\Gamma(\nu_n)} \int_0^1 d^n\vec{x}\, \frac{x_1^{\nu_1-1}\ldots x_n^{\nu_n-1}\delta\left(1-\sum_{j=1}^n x_j\right)}{(x_1 D_1 + \ldots + x_n D_n)^{\nu}},$$

- This gives the well-known Feynman parametrization:

$$I_{\nu_1,\ldots,\nu_n} = \frac{\Gamma(\nu - ld/2)}{\prod_{j=1}^n \Gamma(\nu_j)} \int \left(\prod_{j=1}^{n} dx_j x_j^{\nu_j-1}\right) \frac{\mathcal{U}^{\nu-(l+1)d/2}}{\mathcal{F}^{\nu-ld/2}} \delta\left(1 - \sum_{j=1}^{n} x_j\right)$$

Direct integration & differential equations

See also:
[MH, Moriello, 1712.04441], [Papadopoulos, Wever, 1910.06275]

- Alternatively, we may apply the formula recursively to two propagators:

$$\begin{aligned} D_{12} &= x_1 D_1 + (1-x_1) D_2 \\ D_{123} &= x_2 D_{12} + (1-x_2) D_3 \\ &\dots \\ D_{1\dots n} &= x_{n-1} D_{1\dots(n-1)} + (1-x_{n-1}) D_n \end{aligned}$$

- And we define a collection of integral families:

$$I^{(\kappa)}_{\nu_1,\dots,\nu_{n-\kappa}} = \int \left(\prod_{j=1}^{l} \frac{d^d k_j}{i\pi^{\frac{d}{2}}} \right) D^{-\nu_1}_{1\dots(\kappa+1)} \prod_{j=\kappa+2}^{n} D_j^{-\nu_{j-\kappa}} \qquad \text{for } 0 \le \kappa < n-2,$$

$$I^{(n-1)}_{\nu} = \int \left(\prod_{j=1}^{l} \frac{d^d k_j}{i\pi^{\frac{d}{2}}} \right) D^{-\nu}_{1\dots n}.$$

Direct integration & differential equations

- Example: $n = 4$ propagators

$$I^{(0)}_{\nu_1...\nu_4} = I_{\nu_1\cdots\nu_n} = \int \left(\prod_{j=1}^{l} \frac{d^d k_j}{i\pi^{\frac{d}{2}}} \right) D_1^{-\nu_1} D_2^{-\nu_2} D_3^{-\nu_3} D_4^{-\nu_4}$$

$$I^{(1)}_{\nu_1...\nu_4} = \int \left(\prod_{j=1}^{l} \frac{d^d k_j}{i\pi^{\frac{d}{2}}} \right) D_{12}^{-\nu_1} D_3^{-\nu_2} D_4^{-\nu_3} D_1^{-\nu_4}$$

$$I^{(2)}_{\nu_1...\nu_4} = \int \left(\prod_{j=1}^{l} \frac{d^d k_j}{i\pi^{\frac{d}{2}}} \right) D_{123}^{-\nu_1} D_4^{-\nu_2} D_1^{-\nu_3} D_2^{-\nu_4}$$

$$I^{(3)}_{\nu_1...\nu_4} = \int \left(\prod_{j=1}^{l} \frac{d^d k_j}{i\pi^{\frac{d}{2}}} \right) D_{1234}^{-\nu_1} D_1^{-\nu_2} D_2^{-\nu_3} D_3^{-\nu_4}$$

- The orange terms may be added to maintain the same number of propagators / numerators for IBP reductions.

Direct integration & differential equations

- Upon integration we find

$$I_{\nu_1,...,\nu^{(\kappa-1)}_{n-(\kappa-1)}} = \frac{\Gamma(\nu_1+\nu_2)}{\Gamma(\nu_1)\Gamma(\nu_2)} \int_0^1 dx_\kappa x_\kappa^{\nu_1-1}(1-x_\kappa)^{\nu_2-1} I^{(\kappa)}_{\nu_1+\nu_2,\nu_3,...,\nu_{n-\kappa}}$$

assuming that ν_1 and ν_2 are positive.

- For subsectors, it holds that:

$$I^{(\kappa-1)}_{0,0,\nu_3,...,\nu_{n-(\kappa-1)}} = I^{(\kappa)}_{0,\nu_3,...,\nu_{n-\kappa}}$$

$$I^{(\kappa-1)}_{\nu_1,0,\nu_3,...,\nu_{n-(\kappa-1)}} = \lim_{x_\kappa\to 1} I^{(\kappa)}_{\nu_1,\nu_3,...,\nu_{n-\kappa}}$$

$$I^{(\kappa-1)}_{0,\nu_2,\nu_3,...,\nu_{n-(\kappa-1)}} = \lim_{x_\kappa\to 0} I^{(\kappa)}_{\nu_2,\nu_3,...,\nu_{n-\kappa}}$$

For example:

$$\lim_{x_1\to 1} D_{12} = \lim_{x_1\to 1}(x_1D_1+(1-x_1)D_2) = D_1$$

$$\lim_{x_1\to 0} D_{12} = \lim_{x_1\to 0}(x_1D_1+(1-x_1)D_2) = D_2$$

- Thus, all integrals in step $\kappa-1$ are determined from integrals in step κ.

Direct integration & differential equations

- Note that by iterating the integration formula, we find:

$$I_{\nu_1,...,\nu_n} = \frac{\Gamma(\nu)}{\Gamma(\nu_1)\ldots\Gamma(\nu_n)} \left(\prod_{j=1}^{n-1} \int_0^1 dx_j x_j^{\mu_j - 1} (1-x_j)^{\nu_{j+1}-1} \right) I_\nu^{(n-1)}$$

- The recursion ends at a “generalized tadpole” integral:

$$I_\nu^{(n-1)} = \int \left(\prod_{j=1}^{l} \frac{d^d k_j}{i\pi^{\frac{d}{2}}} \right) D_{1\ldots n}^{-\nu} = \frac{\Gamma(\nu - ld/2)}{\Gamma(\nu)} \frac{\tilde{\mathcal{U}}^{\nu-(l+1)d/2}}{\tilde{\mathcal{F}}^{\nu - ld/2}}$$

- Where $\tilde{\mathcal{U}}$ and $\tilde{F}$ are rescaled versions

of the standard Symanzik polynomials:

$$x_1 \to x_1' = \prod_{i=1}^{n-1} x_i$$

$$x_j \to x_j' = (1 - x_{j-1}) \prod_{i=j}^{n-1} x_i \quad \text{for } j = 2, \ldots, n-1$$

$$x_n \to x_n' = (1 - x_{n-1})$$

Direct integration & differential equations

- The recursion allows us to obtain boundary conditions for all families (κ).

1. Set up a system of differential equations:

$$\partial_{x_\kappa} \vec{I}^{(\kappa)} = M_{x_\kappa} \vec{I}^{(\kappa)}$$

2. Transport boundary conditions to obtain a piecewise solution between $0 < x_k < 1$

3. Integrate the expansions according to the recursion formula:

$$I^{(\kappa-1)}_{\nu_1,\ldots,\nu_{n-(\kappa-1)}} = \frac{\Gamma(\nu_1+\nu_2)}{\Gamma(\nu_1)\Gamma(\nu_2)} \int_0^1 dx_\kappa x_\kappa^{\nu_1-1}(1-x_\kappa)^{\nu_2-1} I^{(\kappa)}_{\nu_1+\nu_2,\nu_3,\ldots,\nu_{n-\kappa}}$$

- The first boundary condition is just:

$$I_\nu^{(n-1)} = \int \left(\prod_{j=1}^{l} \frac{d^d k_j}{i\pi^{\frac{d}{2}}}\right) D^{-\nu}_{1\ldots n} = \frac{\Gamma(\nu - ld/2)}{\Gamma(\nu)} \frac{\tilde{\mathcal{U}}^{\nu-(l+1)d/2}}{\tilde{\mathcal{F}}^{\nu-ld/2}}$$

Regularization

- In general, there may be non-integrable singularities at the boundaries $x_\kappa = 0, 1$ as $\epsilon \to 0$!
  - (These are exactly the kinds of singularities that are resolved in the sector decomposition method.)
- Decompose the integrand as follows near $x = x_\kappa = 0$

$$x_\kappa^{\nu_1-1}(1-x_\kappa)^{\nu_2-1} I^{(\kappa)}_{\nu_1+\nu_2,\nu_3,...,\nu_{n-\kappa}} \to \qquad g(x) = g_0(x,\epsilon) + x^{a_1+b_1\varepsilon} g_1(x,\varepsilon) + \ldots + x^{a_k+b_k\varepsilon} g_k(x,\varepsilon)$$

- Then we use the following regularization formula:

$$\int_0^c dx x^{a+b\varepsilon} g_j(x) = \int_0^c dx \frac{x^{a+b\varepsilon+1}}{(1+a+b\varepsilon)} \left(\frac{(2+a+b\varepsilon)}{c} g_j(x) - \left(1 - \frac{x}{c}\right) g_j'(x) \right)$$

Regularization

- Singularities at the boundaries are also a problem for the limit formulas:

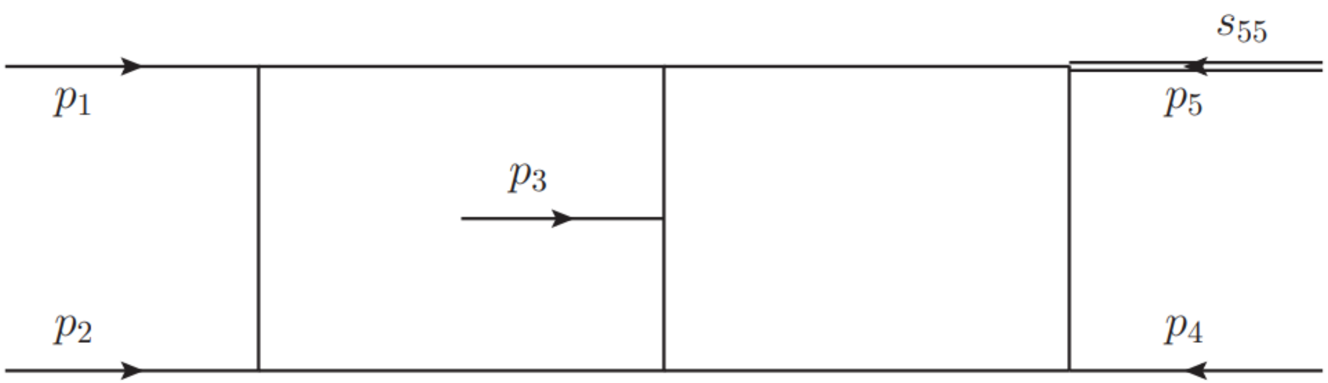
$$
\begin{aligned}
I^{(\kappa-1)}_{0,0,\nu_3,\ldots,\nu_{n-(\kappa-1)}} &= I^{(\kappa)}_{0,\nu_3,\ldots,\nu_{n-\kappa}} \\
I^{(\kappa-1)}_{\nu_1,0,\nu_3,\ldots,\nu_{n-(\kappa-1)}} &= \lim_{x_\kappa \to 1} I^{(\kappa)}_{\nu_1,\nu_3,\ldots,\nu_{n-\kappa}} \\
I^{(\kappa-1)}_{0,\nu_2,\nu_3,\ldots,\nu_{n-(\kappa-1)}} &= \lim_{x_\kappa \to 0} I^{(\kappa)}_{\nu_2,\nu_3,\ldots,\nu_{n-\kappa}}
\end{aligned}
$$

- The resolution is to keep only the Taylor series part

$$
g_0(x,\epsilon) + \cancel{x^{a_1+b_1\varepsilon} g_1(x,\varepsilon) + \ldots + x^{a_k+b_k\varepsilon} g_k(x,\varepsilon)}
$$

and evaluate the limit of $g_0(x,\epsilon)$ at $x = 0$.

5-point 2-loop example:



$$I^{5\text{p}}_{\nu_1,\nu_2,\nu_3,\nu_4,\nu_5,\nu_6,\nu_7,\nu_8,\nu_9,\nu_{10},\nu_{11}} = \int \frac{d^dk_1}{i\pi^{\frac{d}{2}}} \frac{d^dk_2}{i\pi^{\frac{d}{2}}} \frac{D_9^{-\nu_9} D_{10}^{-\nu_{10}} D_{11}^{-\nu_{11}}}{D_1^{\nu_1} D_2^{\nu_2} D_3^{\nu_3} D_4^{\nu_4} D_5^{\nu_5} D_6^{\nu_6} D_7^{\nu_7} D_8^{\nu_8}}$$

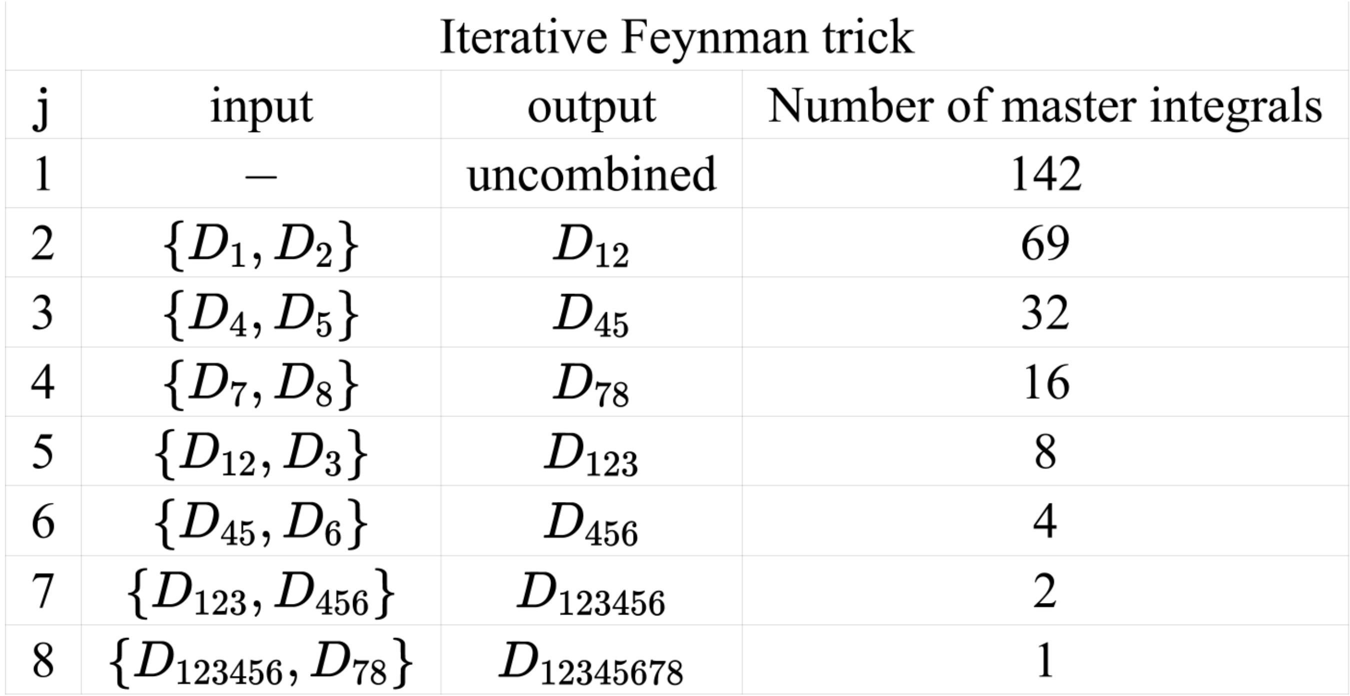
$$\begin{aligned}
D_1 &= (k_2 - p_1 - p_2 - p_3 - p_4)^2 \,, & D_5 &= (k_1 - p_1)^2 \,, & D_9 &= (k_2 - p_1 - p_2)^2 \,, \\
D_2 &= (k_2 - p_1 - p_2 - p_3)^2 \,, & D_6 &= k_1^2 \,, & D_{10} &= (k_1 - p_1 - p_2 - p_3 - p_4)^2 \,, \\
D_3 &= k_2^2 \,, & D_7 &= (k_1 - k_2 + p_3)^2 \,, & D_{11} &= (k_2 - p_1)^2 \,, \\
D_4 &= (k_1 - p_1 - p_2)^2 \,, & D_8 &= (k_1 - k_2)^2 \,. & &
\end{aligned}$$

$$p_1^2 = p_2^2 = p_3^2 = p_4^2 = 0 \,, \qquad p_1 \cdot p_2 = s_{12}/2 \,, \qquad p_1 \cdot p_3 = s_{13}/2 \,, \qquad p_1 \cdot p_4 = s_{14}/2 \,,$$

$$p_2 \cdot p_3 = s_{23}/2 \,, \qquad p_2 \cdot p_4 = -(s_{12} + s_{13} + s_{14} + s_{23} + s_{34} - s_{55})/2 \,, \qquad p_3 \cdot p_4 = s_{34}/2 \,,$$

5-point 2-loop example:

- We combine the propagators in the following way:

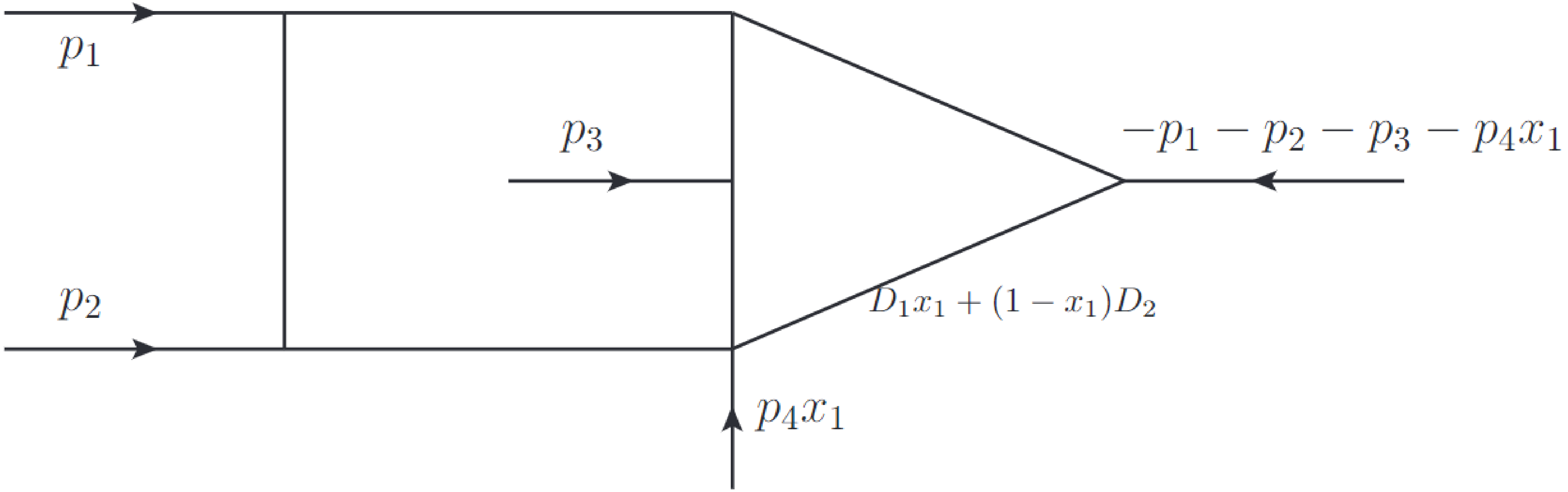


Iterative Feynman trick

j	input	output	Number of master integrals
1	–	uncombined	142
2	$\{D_1, D_2\}$	D_{12}	69
3	$\{D_4, D_5\}$	D_{45}	32
4	$\{D_7, D_8\}$	D_{78}	16
5	$\{D_{12}, D_3\}$	D_{123}	8
6	$\{D_{45}, D_6\}$	D_{456}	4
7	$\{D_{123}, D_{456}\}$	D_{123456}	2
8	$\{D_{123456}, D_{78}\}$	$D_{12345678}$	1

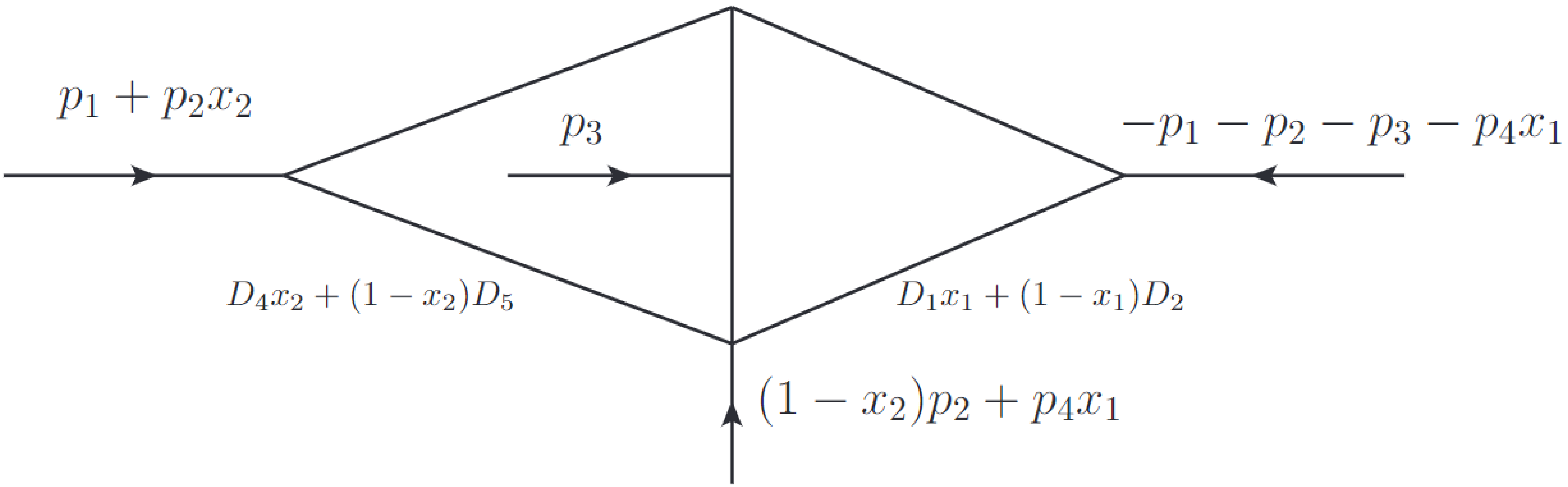
- The choices are motivated by first combining propagators which have the same internal momentum. This leads to simplifications of the graph.

5-point 2-loop example:



- Note that: $D_{12} = (k_2 \underbrace{- p_1 - p_2 - p_3 - p_4 x_1)^2}_{Q} - \underbrace{x_1(1 - x_1)(-p_4^2)}_{M}$

5-point 2-loop example:



5-point 2-loop example:

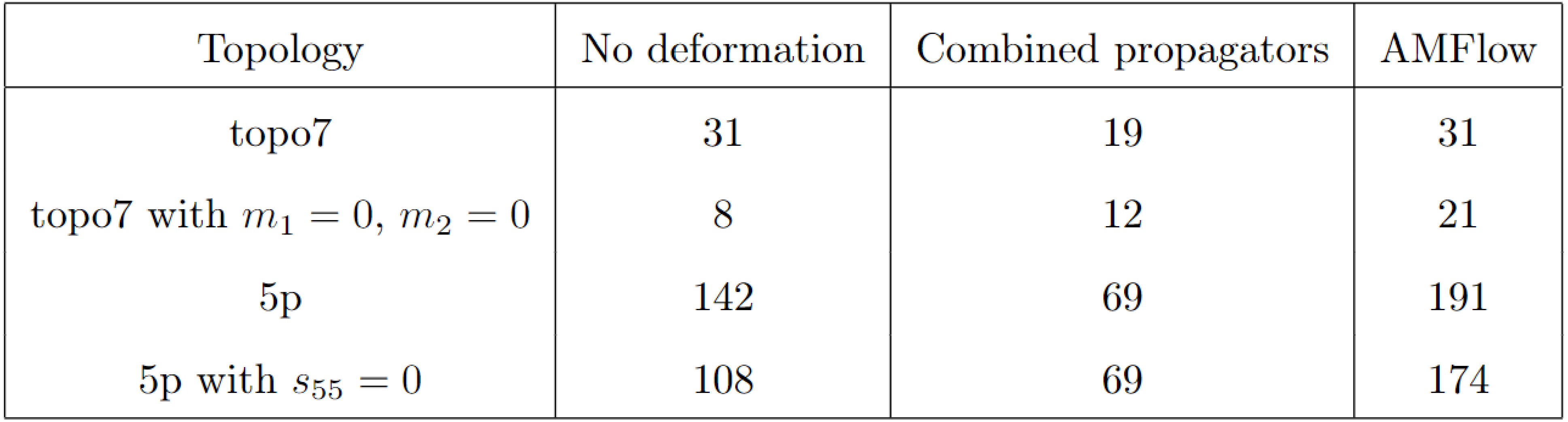
We evaluate one of the most complicated master integrals at the numerical point

$$s_{14} = 3,\ s_{13} = -11/17,\ s_{23} = -13/17,\ s_{12} = -7/17,\ s_{34} = -7/13,\ s_{55} = -1 \text{ in } d = 4 - 2\varepsilon$$

$$\begin{aligned}
I^{5\text{p}}_{13111111000} = {} & \frac{1}{\varepsilon^4}(-80991.446349418328158551349566863301342444 59) \\
& + \frac{1}{\varepsilon^3}(-1176854.14050165085751620090895007182416011 1- \\
& \quad 303701.84533500293424001259182549353163494 29i) \\
& + \frac{1}{\varepsilon^2}(-13432835.8477692962185637394931604891797674- \\
& \quad 4251651.6496598016611477427220153367658058 0i) \\
& + \frac{1}{\varepsilon}(-111346171.6370450328807043552785900423292 1- \\
& \quad 32927342.3956883303000216657885568019681 76i) \\
& + (-763045644.5561305442093867867513427731742- \\
& \quad 183231121.40487741467886614905312052821 19i) \\
& + \varepsilon(-4428755434.16119754697555927652734791719- \\
& \quad 816059490.912195429388068459166197648719i)+ \\
& \varepsilon^2(-23085640630.2598895207779945265376391 99- \\
& \quad 3082908606.7551294811504215473642629605i)+ \\
& \varepsilon^3(-110164352209.709241265245125661094393 8- \\
& \quad 10252510409.4218569155068776615235364 0i)+ \\
& \varepsilon^4(-497649560130.0152092791920986315319 20- \\
& \quad 30796992268.3516086870566559550754104i).
\end{aligned}$$

Computational complexity (IBP):

- Combining two propagators leads to integral families with less master integrals than the deformations from auxiliary mass flow, and in turn faster IBP reductions:



Topology	No deformation	Combined propagators	AMFlow
topo7	31	19	31
topo7 with $m_1 = 0$, $m_2 = 0$	8	12	21
5p	142	69	191
5p with $s_{55} = 0$	108	69	174

- We found that the IBP reductions were 66 times faster compared to auxiliary mass flow for the 5p family. (However, our current implementation is slower on the series solution side.)

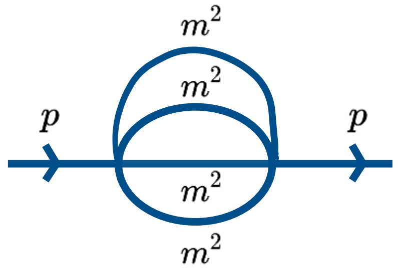
Conclusion

- Series expansion methods allow for obtaining high-precision numerical results for multiloop Feynman integrals with multiple scales.
- The Mathematica package DiffExp can be used for computing user-provided systems of differential equations.
- The “iterative Feynman trick” technique allows us to integrate one Feynman parameter at a time numerically from differential equations.
    - The resulting IBP reductions are less complicated than for the initial topology!
    - The approach can be fully automated.

Thank you for listening!

Backup slides

3-loop banana graph



- First, we consider the equal-mass case:

$$I^{\text{banana}}_{a_1a_2a_3a_4} = \left(\frac{e^{\gamma_E\epsilon}}{i\pi^{d/2}}\right)^3 (m^2)^{a-\frac{3}{2}(2-2\epsilon)} \left(\prod_{i=1}^{4}\int d^dk_i\right) D_1^{-a_1} D_2^{-a_2} D_3^{-a_3} D_4^{-a_4}$$

$$D_1 = -k_1^2 + m^2\,, \quad D_2 = -k_2^2 + m^2\,, \quad D_3 = -k_3^2 + m^2\,, \quad D_4 = -(k_1+k_2+k_3+p_1)^2 + m^2$$

- The differential equations are in precanonical form and given by:

$$\vec{B}^{\text{banana}} = (\epsilon I^{\text{banana}}_{2211}\,, \epsilon(1+3\epsilon)I^{\text{banana}}_{2111}\,, \epsilon(1+3\epsilon)(1+4\epsilon)I^{\text{banana}}_{1111}\,, \epsilon^3 I^{\text{banana}}_{1110}\,)$$

$$\partial_t \vec{B}^{\text{banana}} = \begin{pmatrix} -\frac{64-2t+t^2+(8+t)^2\epsilon}{t(t-16)(t-4)} & \frac{2(t+20)(2\epsilon+1)}{t(t-16)(t-4)} & -\frac{6(2\epsilon+1)}{t(t-16)(t-4)} & -\frac{2\epsilon}{t(t-16)} \\ \frac{3t(3\epsilon+1)}{t(t-4)} & -\frac{2(t+8)\epsilon+t+4}{t(t-4)} & \frac{3\epsilon+1}{t(t-4)} & 0 \\ 0 & \frac{4(4\epsilon+1)}{t} & \frac{-3\epsilon-1}{t} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \vec{B}^{\text{banana}}$$

- With $t = p_1^2/m^2$

3-loop banana graph

- We use the method of expansions by regions and `asy.m` to obtain boundary conditions in the limit $\mathtt{t} = x \to -\infty$. They are given by:

$$I_{1111}^{\text{banana}} \overset{x\downarrow 0}{\sim} \frac{6e^{3\gamma\epsilon}\epsilon x^{\epsilon+1}\Gamma(-\epsilon)^2\Gamma(\epsilon)^3}{\Gamma(-2\epsilon)} + \frac{8e^{3\gamma\epsilon}\epsilon x^{2\epsilon+1}\Gamma(-\epsilon)^3\Gamma(\epsilon)\Gamma(2\epsilon)}{\Gamma(-3\epsilon)} + \frac{3e^{3\gamma\epsilon}\epsilon x^{3\epsilon+1}\Gamma(-\epsilon)^4\Gamma(3\epsilon)}{\Gamma(-4\epsilon)}$$
$$+ 4xe^{3\gamma\epsilon}\Gamma(\epsilon)^3 + \mathcal{O}(x^2)\,.$$

$$I_{1110}^{\text{banana}} = e^{3\gamma\epsilon}\Gamma(\epsilon)^3$$

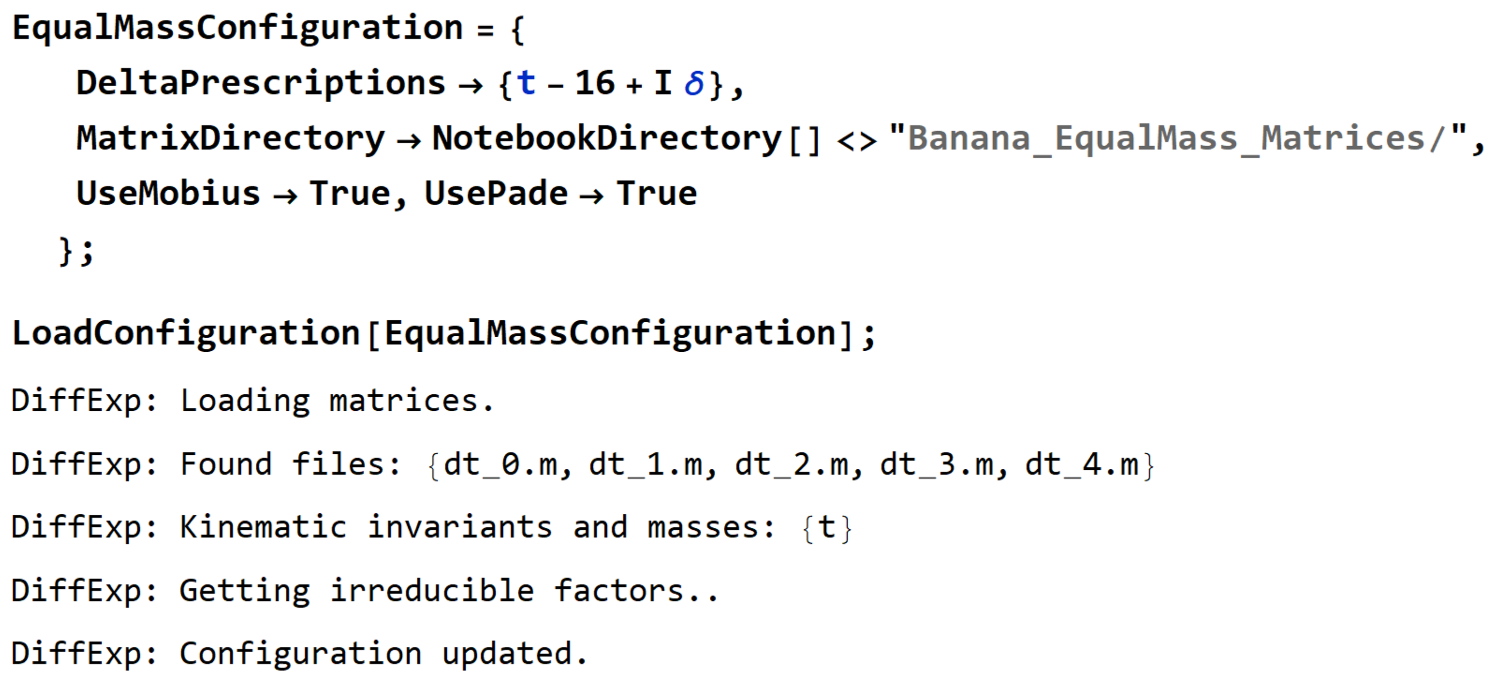
- Next, we show how to obtain results for any values of p^2 using DiffExp

3-loop banana graph

- Load DiffExp:

```
Get[FileNameJoin[{NotebookDirectory[], "..", "DiffExp.m"}]];

Loading DiffExp version 1.0.7

For questions, email: martijn.hidding@physics.uu.se

For the latest version, see: https://gitlab.com/hiddingm/diffexp
```

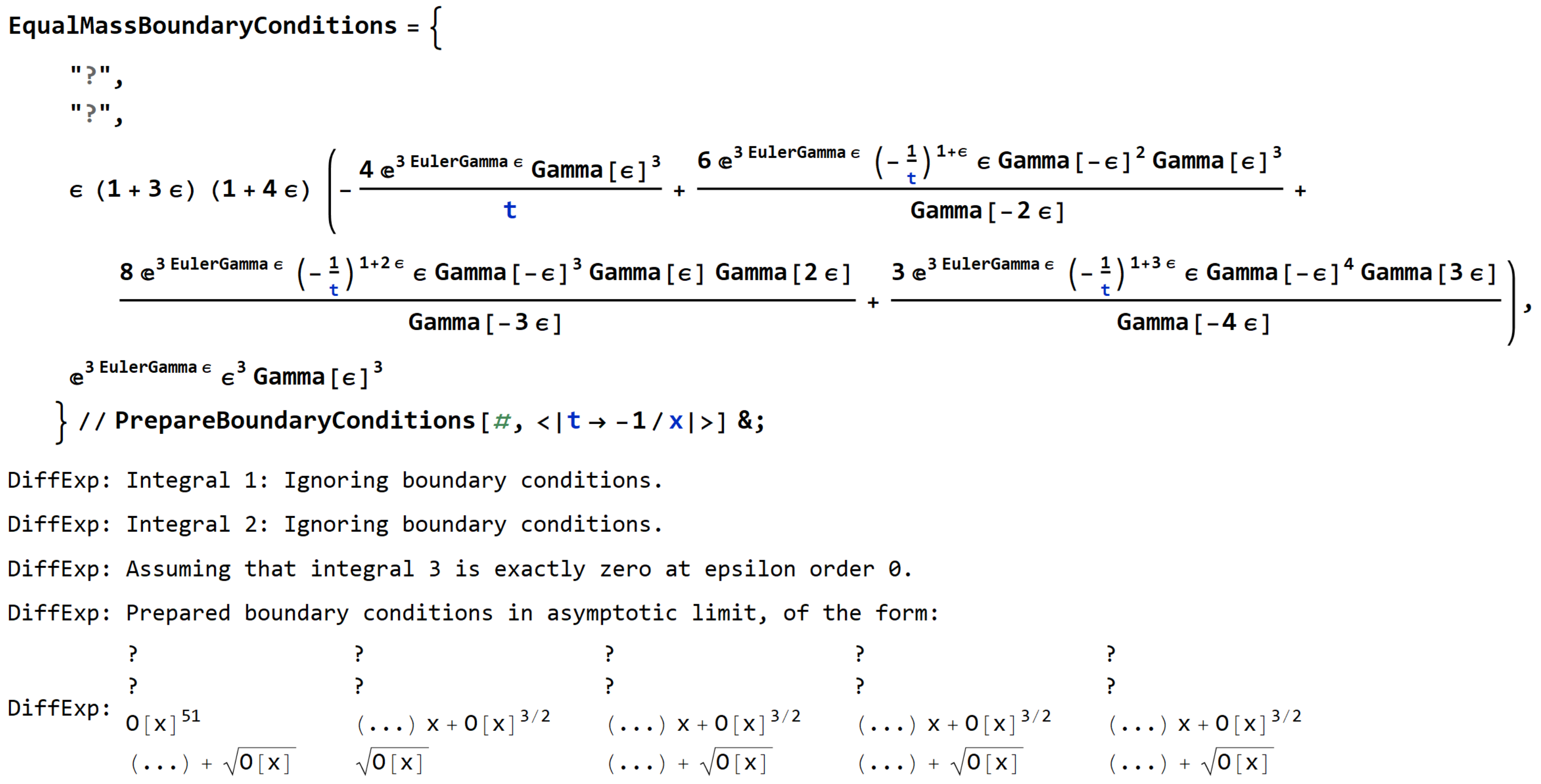
- Set the configuration options and load the matrices



```
EqualMassConfiguration = {
   DeltaPrescriptions → {t - 16 + I δ},
   MatrixDirectory → NotebookDirectory[] <> "Banana_EqualMass_Matrices/",
   UseMobius → True, UsePade → True
  };

LoadConfiguration[EqualMassConfiguration];

DiffExp: Loading matrices.

DiffExp: Found files: {dt_0.m, dt_1.m, dt_2.m, dt_3.m, dt_4.m}

DiffExp: Kinematic invariants and masses: {t}

DiffExp: Getting irreducible factors..

DiffExp: Configuration updated.
```

3-loop banana graph

- Prepare the boundary conditions along an asymptotic limit:



$$
\begin{aligned}
&\texttt{EqualMassBoundaryConditions} = \Big\{\\
&\quad \texttt{"?"},\\
&\quad \texttt{"?"},\\
&\quad \epsilon\,(1+3\epsilon)\,(1+4\epsilon)\left(-\frac{4\,e^{3\,\text{EulerGamma}\,\epsilon}\,\text{Gamma}[\epsilon]^3}{t}+\frac{6\,e^{3\,\text{EulerGamma}\,\epsilon}\left(-\frac{1}{t}\right)^{1+\epsilon}\epsilon\,\text{Gamma}[-\epsilon]^2\,\text{Gamma}[\epsilon]^3}{\text{Gamma}[-2\,\epsilon]}+\right.\\
&\qquad \left.\frac{8\,e^{3\,\text{EulerGamma}\,\epsilon}\left(-\frac{1}{t}\right)^{1+2\,\epsilon}\epsilon\,\text{Gamma}[-\epsilon]^3\,\text{Gamma}[\epsilon]\,\text{Gamma}[2\,\epsilon]}{\text{Gamma}[-3\,\epsilon]}+\frac{3\,e^{3\,\text{EulerGamma}\,\epsilon}\left(-\frac{1}{t}\right)^{1+3\,\epsilon}\epsilon\,\text{Gamma}[-\epsilon]^4\,\text{Gamma}[3\,\epsilon]}{\text{Gamma}[-4\,\epsilon]}\right),\\
&\quad e^{3\,\text{EulerGamma}\,\epsilon}\,\epsilon^3\,\text{Gamma}[\epsilon]^3\\
&\Big\}\;\texttt{//}\;\texttt{PrepareBoundaryConditions}[\#,\ <|t\to -1/x|>]\ \&;
\end{aligned}
$$

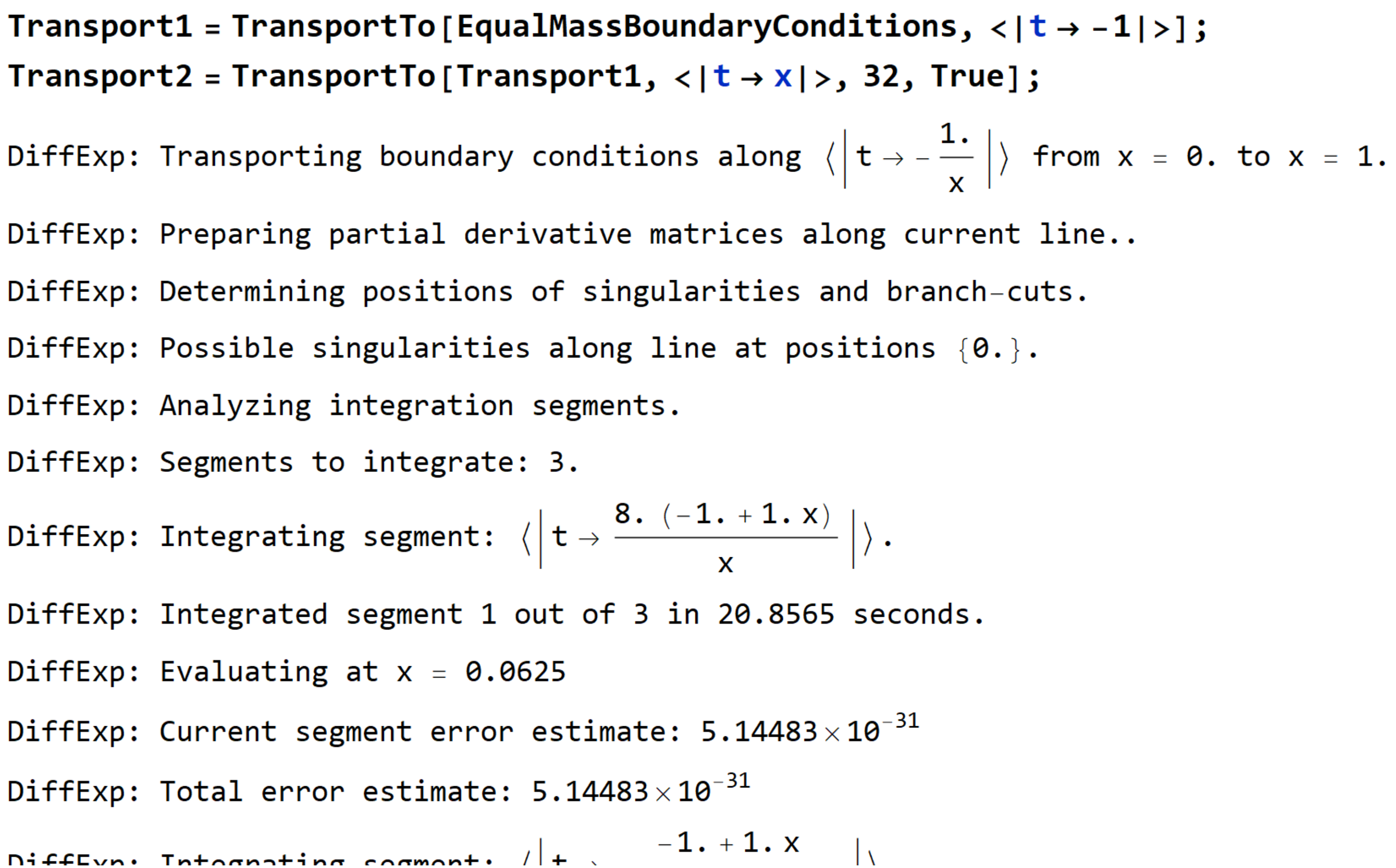
```
DiffExp: Integral 1: Ignoring boundary conditions.

DiffExp: Integral 2: Ignoring boundary conditions.

DiffExp: Assuming that integral 3 is exactly zero at epsilon order 0.

DiffExp: Prepared boundary conditions in asymptotic limit, of the form:
```

DiffExp:

$$
\begin{matrix}
? & ? & ? & ? & ?\\
? & ? & ? & ? & ?\\
O[x]^{51} & (\ldots)\,x+O[x]^{3/2} & (\ldots)\,x+O[x]^{3/2} & (\ldots)\,x+O[x]^{3/2} & (\ldots)\,x+O[x]^{3/2}\\
(\ldots)+\sqrt{O[x]} & \sqrt{O[x]} & (\ldots)+\sqrt{O[x]} & (\ldots)+\sqrt{O[x]} & (\ldots)+\sqrt{O[x]}
\end{matrix}
$$

3-loop banana graph

- Next, we transport the boundary conditions:



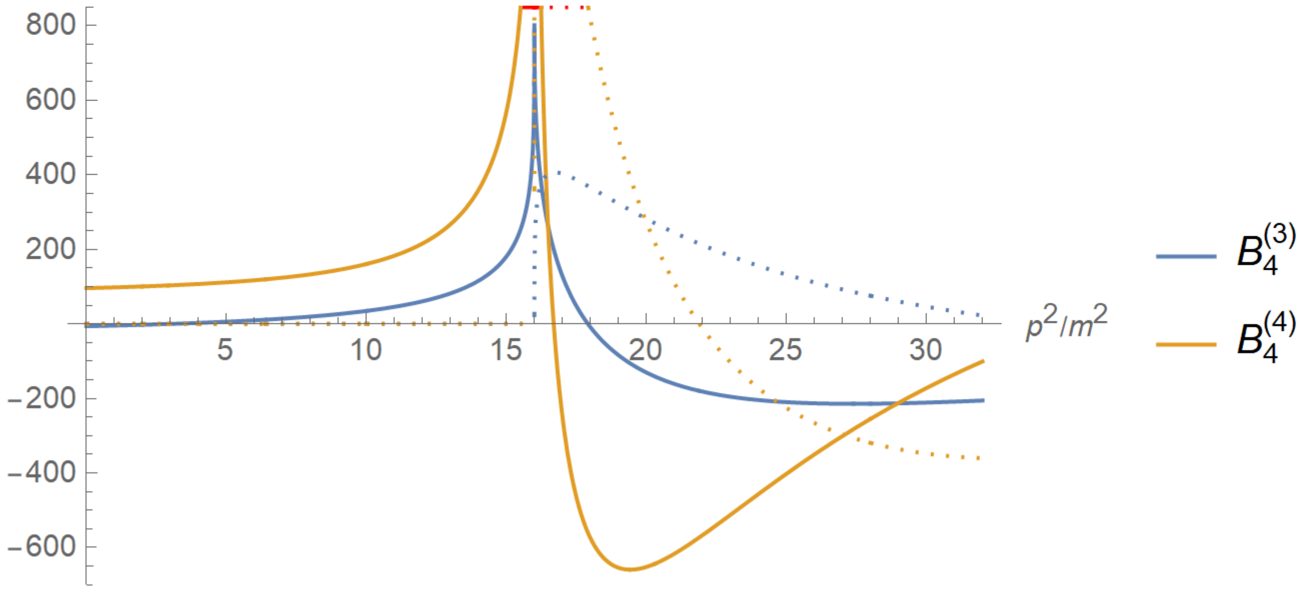
```
Transport1 = TransportTo[EqualMassBoundaryConditions, <|t → -1|>];
Transport2 = TransportTo[Transport1, <|t → x|>, 32, True];
```

```
DiffExp: Transporting boundary conditions along ⟨|t → -1./x|⟩ from x = 0. to x = 1.

DiffExp: Preparing partial derivative matrices along current line..

DiffExp: Determining positions of singularities and branch-cuts.

DiffExp: Possible singularities along line at positions {0.}.

DiffExp: Analyzing integration segments.

DiffExp: Segments to integrate: 3.

DiffExp: Integrating segment: ⟨|t → 8. (-1. + 1. x)/x|⟩.

DiffExp: Integrated segment 1 out of 3 in 20.8565 seconds.

DiffExp: Evaluating at x = 0.0625

DiffExp: Current segment error estimate: 5.14483×10^-31

DiffExp: Total error estimate: 5.14483×10^-31

DiffExp: Integrating segment: ⟨|t → -1. + 1. x
```

3-loop banana graph

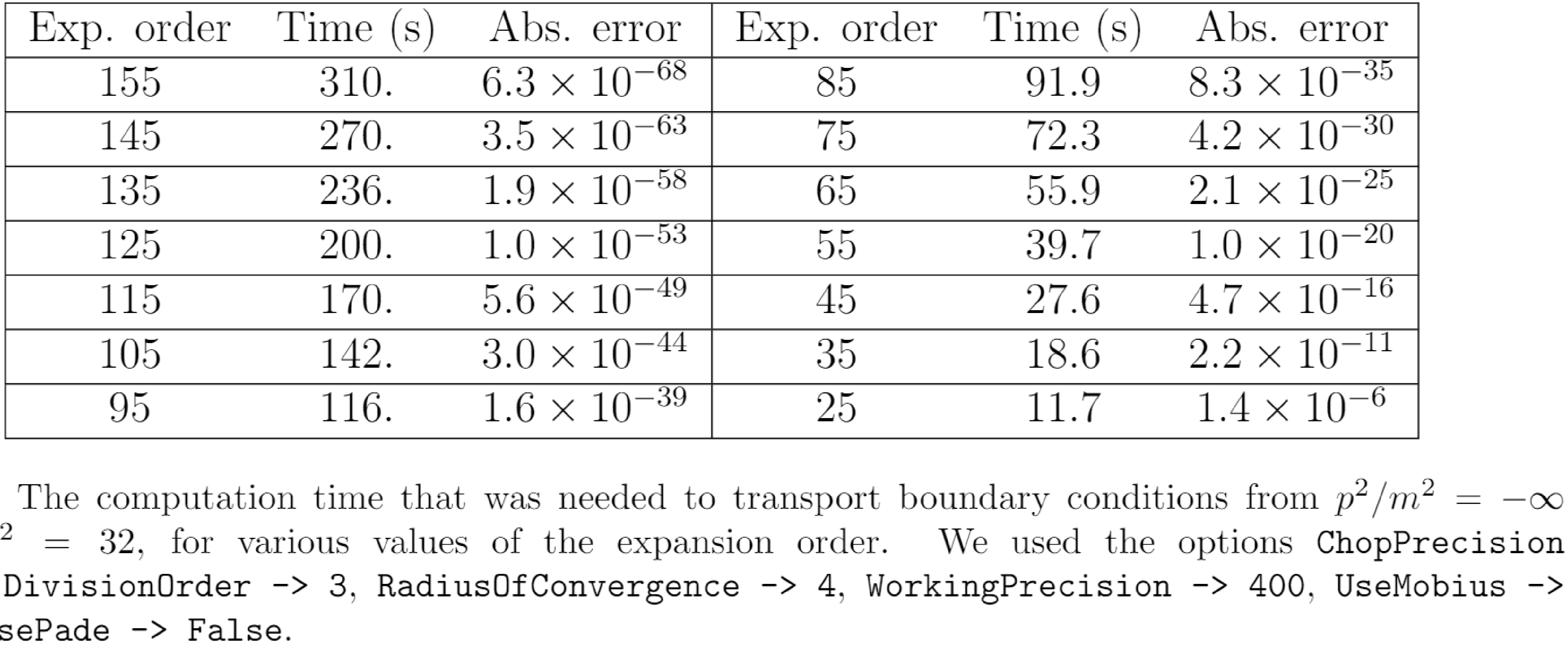
- Lastly, we plot the result:

```
ResultsForPlotting = ToPiecewise[Transport2];
Quiet[ReImPlot[{ResultsForPlotting[[3, 4]][x], ResultsForPlotting[[3, 5]][x]}, {x, 0, 32},
  ClippingStyle → Red, PlotLegends → {"B_4^(3)", "B_4^(4)"}, AxesLabel → {"p^2/m^2"}, PlotRange → {-700, 850},
  MaxRecursion → 15, WorkingPrecision → 100]]
```



3-loop banana graph

- Computation time typically scales quadratically with expansion order:



Exp. order	Time (s)	Abs. error	Exp. order	Time (s)	Abs. error
155	310.	6.3×10^{-68}	85	91.9	8.3×10^{-35}
145	270.	3.5×10^{-63}	75	72.3	4.2×10^{-30}
135	236.	1.9×10^{-58}	65	55.9	2.1×10^{-25}
125	200.	1.0×10^{-53}	55	39.7	1.0×10^{-20}
115	170.	5.6×10^{-49}	45	27.6	4.7×10^{-16}
105	142.	3.0×10^{-44}	35	18.6	2.2×10^{-11}
95	116.	1.6×10^{-39}	25	11.7	1.4×10^{-6}

Table 1: The computation time that was needed to transport boundary conditions from $p^2/m^2 = -\infty$ to $p^2/m^2 = 32$, for various values of the expansion order. We used the options `ChopPrecision -> 225`, `DivisionOrder -> 3`, `RadiusOfConvergence -> 4`, `WorkingPrecision -> 400`, `UseMobius -> False`, `UsePade -> False`.

Expansion by regions

Kinematic invariants and masses

- Suppose we are interested in a kinematic limit $\boldsymbol{s}_i \to \boldsymbol{s}_i' = x^{\gamma_i}\boldsymbol{s}_i$ for $i = 1, \ldots, |S|$
- Then there exists a set of regions $\{R_i\}$, where $R_i = (r_{i1}, \ldots, r_{im})$ is a vector of rational numbers.
- For each region R_i we rescale the Feynman parametrized integral in the following manner: $\underbrace{\alpha_j \to x^{R_{ij}}\alpha_j\,, \quad d\alpha_j \to x^{R_{ij}}d\alpha_j\,,}_{\text{Each Feynman parameter scales according to the given region}} \quad \underbrace{s_j \to x^{\gamma_j}s_j}_{\text{In addition, we take our desired kinematic limit}}$
- The asymptotic limit is then given by summing over the contributions of each region, expanding on x, and integrating.

Expansion by regions

- Let’s have another look at the massive bubble. The Feynman parametrization is:

$$\frac{e^{\gamma_E \epsilon}\Gamma(\epsilon+1)}{i\pi^{1-\epsilon}} \int_\Delta d\alpha_1 d\alpha_2\, (\alpha_1+\alpha_2)^{2\epsilon}\left(\alpha_1^2 m^2+\alpha_2^2 m^2+2\alpha_1\alpha_2 m^2-\alpha_1\alpha_2 p^2\right)^{-1-\epsilon}$$

- We feed `asy.m` the $\mathcal{U}$ and $\mathcal{F}$ polynomials, and obtain the regions:

$$R_1=\{0,0\},\quad R_2=\{0,-1\},\quad R_3=\{0,1\}$$

- Leading to: $$\frac{e^{\gamma_E \epsilon}\Gamma(\epsilon+1)}{i\pi^{1-\epsilon}} \int_\Delta d\alpha_1 d\alpha_2 \Big(x^{-\epsilon}(x\alpha_1+\alpha_2)^{2\epsilon}\left(x^2\alpha_1^2-p^2\alpha_1\alpha_2+2x\alpha_1\alpha_2+\alpha_2^2\right)^{-1-\epsilon} \\ +(\alpha_1+\alpha_2)^{2\epsilon}\left(x\alpha_1^2-p^2\alpha_1\alpha_2+2x\alpha_1\alpha_2+x\alpha_2^2\right)^{-1-\epsilon} \\ +x^{-\epsilon}(\alpha_1+x\alpha_2)^{2\epsilon}\left(\alpha_1^2-p^2\alpha_1\alpha_2+2x\alpha_1\alpha_2+x^2\alpha_2^2\right)^{-1-\epsilon}\Big)$$

- For the purpose of computing boundary conditions, we often only need the leading term of the expansion with respect to the line parameter

Expansion by regions

- At leading order in x, we obtain:

$$\frac{e^{\gamma_E \epsilon}\Gamma(\epsilon+1)}{i\pi^{1-\epsilon}} \int_\Delta d\alpha_1 d\alpha_2 \left(x^{-\epsilon}\alpha_2^{-1+\epsilon}\left(-p^2\alpha_1+m^2\alpha_2\right)^{-1-\epsilon}+\right.$$
$$\left.+\alpha_1^{-\epsilon-1}\alpha_2^{-\epsilon-1}\left(\alpha_1+\alpha_2\right)^{2\epsilon}\left(-p^2\right)^{-1-\epsilon}+x^{-\epsilon}\alpha_1^{\epsilon-1}\left(\alpha_1 m^2-\alpha_2 p^2\right)^{-\epsilon-1}\right)$$

- Although we have a sum of terms, each piece is simpler to integrate than the Feynman parametrization of the massive bubble. Performing the integrations yields:

$$\frac{\epsilon\left(-p^2\right)^{-\epsilon-1}\Gamma(-\epsilon)^2\Gamma(\epsilon)}{\Gamma(-2\epsilon)}-\frac{2x^{-\epsilon}\Gamma(\epsilon)}{p^2}=-\frac{2\left(\log\left(-p^2\right)-\log(x)\right)}{p^2}+\mathcal{O}(\epsilon)$$

- Which agrees with the result we found before!
- Note as well that the boundary conditions are just ratios of gamma functions

DiffExp

- A general implementation of these methods was made into the Mathematica package DiffExp, introduced in arXiv:2006.05510, (available at https://gitlab.com/hiddingm/diffexp)
- DiffExp accepts (any) system of differential equations of the form

$$\frac{\partial}{\partial s}\vec{f}(\{S\},\epsilon) = \mathbf{A}_s\vec{f}(\{S\},\epsilon) \qquad \mathbf{A}_x(x,\epsilon) = \sum_{k=0}^{\infty}\mathbf{A}_x^{(k)}(x)\epsilon^k$$

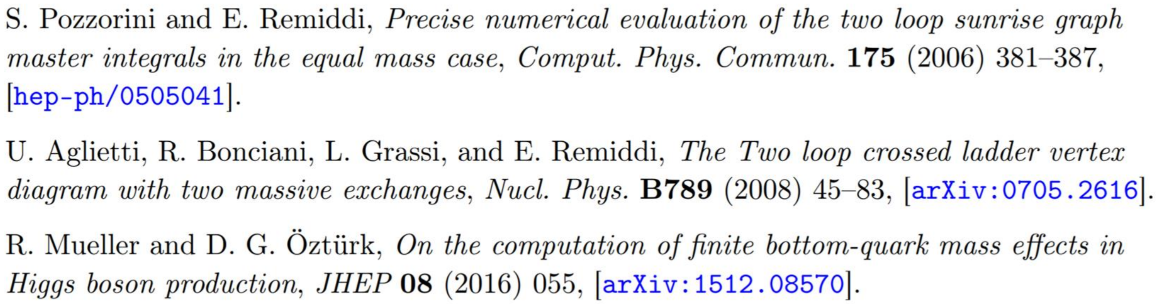
  for which the matrix entries are combinations of rational and algebraic functions

- It enables one to numerically integrate various multi-scale Feynman integrals at arbitrary points in phase-space, and at precisions of tens of digits (or higher)
- The Feynman integrals do not have to be in canonical form and may also be of “elliptic”-type or associated with more complicated geometries.

Series expansions

- Series expansions have been featured various times in the past literature.

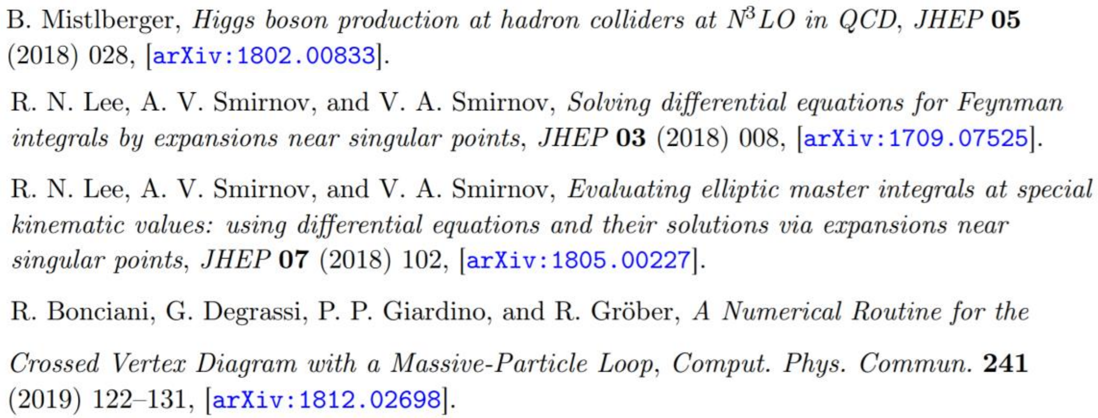
- For single-scale problems, see e.g:



S. Pozzorini and E. Remiddi, *Precise numerical evaluation of the two loop sunrise graph master integrals in the equal mass case*, *Comput. Phys. Commun.* **175** (2006) 381–387, [hep-ph/0505041].

U. Aglietti, R. Bonciani, L. Grassi, and E. Remiddi, *The Two loop crossed ladder vertex diagram with two massive exchanges*, *Nucl. Phys.* **B789** (2008) 45–83, [arXiv:0705.2616].

R. Mueller and D. G. Öztürk, *On the computation of finite bottom-quark mass effects in Higgs boson production*, *JHEP* **08** (2016) 055, [arXiv:1512.08570].



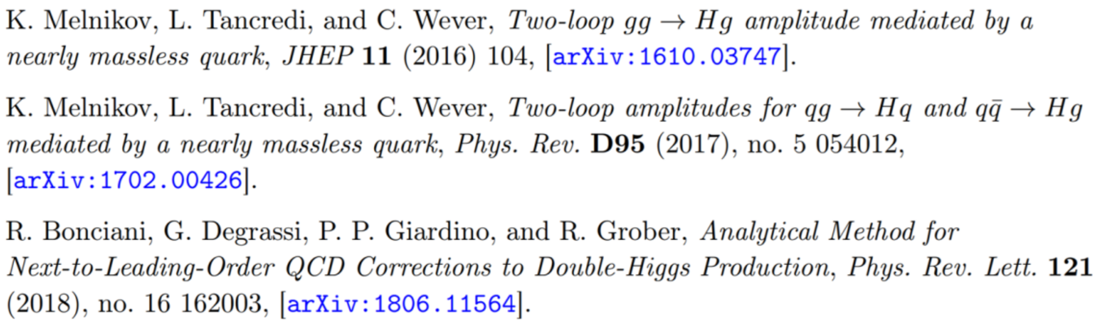
B. Mistlberger, *Higgs boson production at hadron colliders at N^3LO in QCD*, *JHEP* **05** (2018) 028, [arXiv:1802.00833].

R. N. Lee, A. V. Smirnov, and V. A. Smirnov, *Solving differential equations for Feynman integrals by expansions near singular points*, *JHEP* **03** (2018) 008, [arXiv:1709.07525].

R. N. Lee, A. V. Smirnov, and V. A. Smirnov, *Evaluating elliptic master integrals at special kinematic values: using differential equations and their solutions via expansions near singular points*, *JHEP* **07** (2018) 102, [arXiv:1805.00227].

R. Bonciani, G. Degrassi, P. P. Giardino, and R. Gröber, *A Numerical Routine for the Crossed Vertex Diagram with a Massive-Particle Loop*, *Comput. Phys. Commun.* **241** (2019) 122–131, [arXiv:1812.02698].

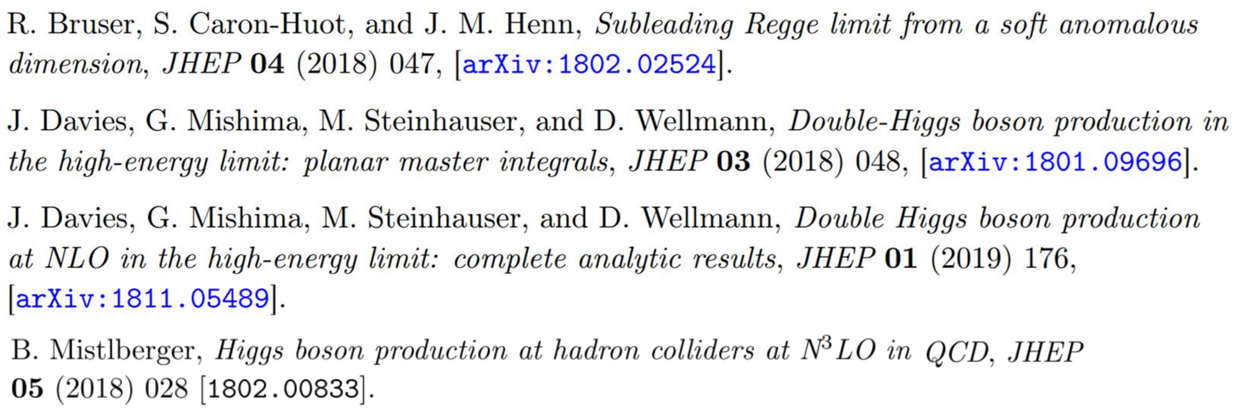
- For multi-scale problems, see for example:



K. Melnikov, L. Tancredi, and C. Wever, *Two-loop $gg \to Hg$ amplitude mediated by a nearly massless quark*, *JHEP* **11** (2016) 104, [arXiv:1610.03747].

K. Melnikov, L. Tancredi, and C. Wever, *Two-loop amplitudes for $qg \to Hq$ and $q\bar{q} \to Hg$ mediated by a nearly massless quark*, *Phys. Rev.* **D95** (2017), no. 5 054012, [arXiv:1702.00426].

R. Bonciani, G. Degrassi, P. P. Giardino, and R. Grober, *Analytical Method for Next-to-Leading-Order QCD Corrections to Double-Higgs Production*, *Phys. Rev. Lett.* **121** (2018), no. 16 162003, [arXiv:1806.11564].



R. Bruser, S. Caron-Huot, and J. M. Henn, *Subleading Regge limit from a soft anomalous dimension*, *JHEP* **04** (2018) 047, [arXiv:1802.02524].

J. Davies, G. Mishima, M. Steinhauser, and D. Wellmann, *Double-Higgs boson production in the high-energy limit: planar master integrals*, *JHEP* **03** (2018) 048, [arXiv:1801.09696].

J. Davies, G. Mishima, M. Steinhauser, and D. Wellmann, *Double Higgs boson production at NLO in the high-energy limit: complete analytic results*, *JHEP* **01** (2019) 176, [arXiv:1811.05489].

B. Mistlberger, *Higgs boson production at hadron colliders at N^3LO in QCD*, *JHEP* **05** (2018) 028 [1802.00833].

Additional literature

Samuel Abreu, Harald Ita, Francesco Moriello, Ben Page, Wladimir Tschernow, and Mao Zeng. Two-Loop Integrals for Planar Five-Point One-Mass Processes. 2020.

Xiao Liu, Yan-Qing Ma, Wei Tao, and Peng Zhang. Calculation of Feynman loop integration and phase-space integration via auxiliary mass flow. *Chin. Phys. C*, 45(1):013115, 2021.

Samuel Abreu, Harald Ita, Ben Page, and Wladimir Tschernow. Two-Loop Hexa-Box Integrals for Non-Planar Five-Point One-Mass Processes. 7 2021.

Matteo Fael, Fabian Lange, Kay Schönwald, and Matthias Steinhauser. A semi-analytic method to compute Feynman integrals applied to four-loop corrections to the $\overline{\mathrm{MS}}$-pole quark mass relation. *JHEP*, 09:152, 2021.

Matteo Becchetti, Francesco Moriello, and Armin Schweitzer. Two-loop amplitude for mixed QCD-EW corrections to $gg \to Hg$. 12 2021.

Tommaso Armadillo, Roberto Bonciani, Simone Devoto, Narayan Rana, and Alessandro Vicini. Two-loop mixed QCD-EW corrections to neutral current Drell-Yan. 1 2022.

Xiao Liu and Yan-Qing Ma. Multiloop corrections for collider processes using auxiliary mass flow. 7 2021.

Xiao Liu and Yan-Qing Ma. AMFlow: a Mathematica Package for Feynman integrals computation via Auxiliary Mass Flow. 1 2022.

Zhi-Feng Liu and Yan-Qing Ma. Automatic computation of Feynman integrals containing linear propagators via auxiliary mass flow. 1 2022.

Additional literature

Matteo Fael, Fabian Lange, Kay Schönwald, and Matthias Steinhauser. Massive Vector Form Factors to Three Loops. *Phys. Rev. Lett.*, 128(17):172003, 2022.

Simon Badger, Heribertus Bayu Hartanto, and Simone Zoia. Two-Loop QCD Corrections to Wbb⁻ Production at Hadron Colliders. *Phys. Rev. Lett.*, 127(1):012001, 2021.

Simon Badger, Heribertus Bayu Hartanto, Jakub Kryś, and Simone Zoia. Two-loop leading-colour QCD helicity amplitudes for Higgs boson production in association with a bottom-quark pair at the LHC. *JHEP*, 11:012, 2021.

Simon Badger, Heribertus Bayu Hartanto, Jakub Kryś, and Simone Zoia. Two-loop leading colour helicity amplitudes for $W^{\pm}\gamma + j$ production at the LHC. 1 2022.

Simon Badger, Matteo Becchetti, Ekta Chaubey, Robin Marzucca, and Francesco Sarandrea. One-loop QCD helicity amplitudes for $pp \to t\bar{t}j$ to $O(\epsilon^2)$. 1 2022.

Samuel Abreu, Matteo Becchetti, Claude Duhr, and Melih A. Ozcelik. Two-loop master integrals for pseudo-scalar quarkonium and leptonium production and decay. 6 2022.

Tommaso Armadillo, Roberto Bonciani, Simone Devoto, Narayan Rana, and Alessandro Vicini. Evaluation of Feynman integrals with arbitrary complex masses via series expansions. 5 2022.