(Semi-)automated methods for solving Feynman integrals through differential equations

Martijn Hidding

Uppsala University

Based on work in collaboration with:

Ievgen Dubovyk, Krzysztof Grzanka, Johann Usovitsch

RADCOR/Loopfest 2021

Introduction

- In recent years, the method of diffential equations has proven to be an exceptionally powerful way of computing Feynman integrals.

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

- The effectiveness of the differential equations method is especially striking when it is applied to polylogarithmic integral families that admit an ϵ-factorized (canonical) basis.

  [Henn, 2013]

- Furthermore, numerical approaches to solving the 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]

- Although many individual steps have been automated, some "glue" is still missing. In this talk we will consider some steps towards a full automatization.

Outline of the talk

- The method of differential equations
- Solutions through iterated series expansions
- Overview of an automated computational strategy
- The DiffExp Mathematica package & the Caesar toolbox
- Applications to a 3-loop vertex topology

Differential equations

- We consider a family of Feynman integrals:

$$I_{a_1,\ldots,a_{n+m}} = \int \left(\prod_{i=1}^{l} \frac{d^d k_i}{i\pi^{d/2}} \right) \frac{\prod_{i=n+1}^{n+m} N_i^{-a_i}}{\prod_{i=1}^{n} D_i^{a_i}}, \qquad \begin{array}{l} d = d_{\text{int}} - 2\epsilon \\ D_i = -q_i^2 + m_i^2 - i\delta \end{array}$$

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

$$\partial_{s_j} \vec{I} = \mathbf{M}_{s_j}(\{s_i\}, \epsilon)\vec{I}$$

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

- We will proceed by solving these equations iteratively in terms of one-dimensional series expansions, which will allow us to obtain numerical results everywhere in phase-space.

Differential equations

- Let us briefly consider the special case of a canonical basis. Under a change of variables $\vec{B} = \mathbf{T}\vec{I}$, we have that:

$$\frac{\partial}{\partial s_i}\vec{B} = \left[(\partial_{s_i}\mathbf{T})\,\mathbf{T}^{-1} + \mathbf{T}\mathbf{M}_{s_i}\mathbf{T}^{-1}\right]\vec{B}\,.$$

[Henn, 2013]

- For polylogarithmic families, it is conjectured that a $\mathbf{T}$ exists, such that:

$$\frac{\partial \vec{B}}{\partial s_i} = \epsilon \frac{\partial \tilde{\mathbf{A}}}{\partial s_i}\vec{B}, \qquad d\vec{B} = \epsilon d\tilde{\mathbf{A}}\vec{B}$$

where $\tilde{\boldsymbol{A}}$ does not depends on ϵ, and such that

$$\tilde{\mathbf{A}} = \sum_{i\in\mathcal{A}} \mathbf{C}_i \log(l_i)$$

decomposes as a $\mathbb{Q}$-linear combination of logarithms of rat./algebraic functions.

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

Differential equations

- Let us parametrize the differential equations along a one-dimensional path. In other words, we consider: $\gamma : [0,1] \to \mathbb{C}^{|S|}$

$$x \mapsto \left(\gamma_{s_1}(x), \ldots, \gamma_{s_{|S|}}(x)\right)$$

- Then we have that: $\partial_x \vec{B} = \varepsilon \frac{\partial \tilde{\mathbf{A}}(\gamma(x))}{\partial x} \vec{B}$

$$\partial_x \vec{B} \equiv \varepsilon \mathbf{A}_x \vec{B}$$

- Upon expanding in ϵ, the equations can be solved order-by-order:

$$\vec{B} = \sum_{i \geq 0} \vec{B}^{(i)} \varepsilon^i \qquad \vec{B}^{(i)}(x) = \int_0^x \mathbf{A}_{x'} \vec{B}^{(i-1)}(x') dx' + \vec{B}^{(i)}(x=0)$$

Differential equations

- Let us expand the matrix $\mathbf{A}_x$ in the line parameter. Then we have:

$$\mathbf{A}_x = x^r \left[\sum_{p=0}^{n} \mathbf{c}_p x^p + \mathcal{O}\left(x^{n+1}\right) \right]$$

- Using integration-by-parts, we find can write for each rational m and integer n:

$$\int x^m \log(x)^n = x^{m+1} \sum_{j=1}^{n} c_j \log(x)^j$$

- Thus, we may perform all the integrations in terms of (generalized) series expansions $B_j^{(k)}(x) = x^r \sum_{n=0}^{\infty} \sum_{m=0}^{k} c_{mn}\, x^n \log(x)^m, \qquad c_{mn} \in \mathbb{C}, \qquad 0 \geq r \in \mathbb{Q}$

- Although each series solution has a limited range of convergence, we may concatenate such solutions to reach any point in phase-space.

Differential equations

- More generally, consider an unsimplified or partially simplified basis $\vec{f}$, satisfying:

$$\frac{\partial}{\partial x}\vec{f}(x,\epsilon) = \mathbf{A}_x(x,\epsilon)\vec{f}(x,\epsilon)$$

See e.g.: [Moriello, '19], [R. Bonciani, V. Del Duca, H. Frellesvig, J. M. Henn, MH, L. Maestri, F. Moriello, G. Salvatori, V. A. Smirnov, '19] [MH, `20]

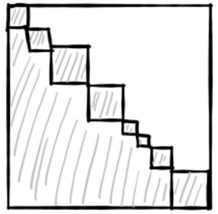
- We will assume that $\mathbf{A}_x$ is finite as ϵ goes to zero, which gives

$$\partial_x \vec{f}^{(k)} = \mathbf{A}_x^{(0)} \vec{f}^{(k)} + \sum_{j=0}^{k-1} \mathbf{A}_x^{(k-j)} f^{(j)}$$

- This can typically be achieved by rescalings of the form:

$$f_i \to \varepsilon^{\rho_i} f_i, \quad \rho_i \in \mathbb{Z}$$

- Lastly, upon ordering the integrals sector-wise, we obtain a "block-triangular" form:

$\mathbf{A}_x^{(0)} \sim$  , which allows us to decompose into differential equations of the form: $\partial_x \vec{g} = \mathbf{M}\vec{g} + \vec{b}$

DiffExp

- DiffExp is a Mathematica package for solving linear systems of differential equations in terms of one-dimensional series expansions. [MH, ` 2006.05510]
- Capable of computing “coupled” systems of more than two integrals
- Takes in (any) system of differential equations of the form

$$\frac{\partial}{\partial s_i}\vec{f}(\{s_j\},\epsilon) = \mathbf{A}_{s_i}\vec{f}(\{s_j\},\epsilon) \qquad \mathbf{A}_{s_i}(\{s_j\},\epsilon) = \sum_{k=0}^{\infty}\mathbf{A}_{s_i}^{(k)}(\{s_j\})\epsilon^k$$

- Uses: compute Feynman integrals numerically at high precision. Analytically continue results across thresholds. Transporting boundary conditions from one special point to another.

DiffExp

- Typical usage of the package:
  - Set configuration options using the method `LoadConfiguration[opts_]`
  - Prepare a list of boundary conditions using `PrepareBoundaryConditions[bcs_, line_]`
  - Then we can find series solutions along a line using the function:

```
IntegrateSystem[bcsprepared_, line_]
```

  - Or one can transport the boundary conditions to a new point using:

```
TransportTo[bcsprepared_, point_]
```

Example: 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.
```

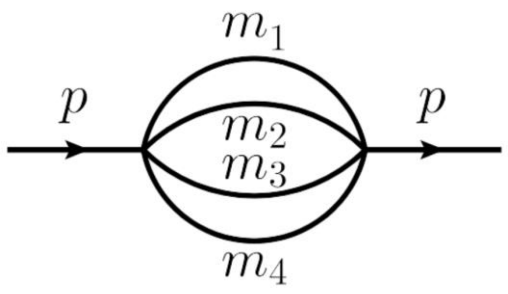


Figure 1: The three-loop unequal mass banana diagram.

Equal-mass case:

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

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

$$D_1 = -k_1^2 + m^2, \quad D_2 = -k_2^2 + m^2,$$

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

3-loop banana graph

- Prepare the boundary conditions along an asymptotic limit:

```
EqualMassBoundaryConditions = {
    "?",
    "?",
```

$$\epsilon\,(1+3\epsilon)\,(1+4\epsilon)\left(-\frac{4\,e^{3\,\text{EulerGamma}\,\epsilon}\,\text{Gamma}[\epsilon]^3}{t}+\frac{6\,e^{3\,\text{EulerGamma}\,\epsilon}\left(-\frac{1}{t}\right)^{1+\epsilon}\epsilon\,\text{Gamma}[-\epsilon]^2\,\text{Gamma}[\epsilon]^3}{\text{Gamma}[-2\epsilon]}+\frac{8\,e^{3\,\text{EulerGamma}\,\epsilon}\left(-\frac{1}{t}\right)^{1+2\epsilon}\epsilon\,\text{Gamma}[-\epsilon]^3\,\text{Gamma}[\epsilon]\,\text{Gamma}[2\epsilon]}{\text{Gamma}[-3\epsilon]}+\frac{3\,e^{3\,\text{EulerGamma}\,\epsilon}\left(-\frac{1}{t}\right)^{1+3\epsilon}\epsilon\,\text{Gamma}[-\epsilon]^4\,\text{Gamma}[3\epsilon]}{\text{Gamma}[-4\epsilon]}\right),$$

$$e^{3\,\text{EulerGamma}\,\epsilon}\,\epsilon^3\,\text{Gamma}[\epsilon]^3$$

```
} // PrepareBoundaryConditions[#, <|t → -1/x|>] &;
```

```
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:

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

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 $\langle | t \to -\frac{1.}{x} | \rangle$ 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: $\langle | t \to \frac{8.\,(-1. + 1.\,x)}{x} | \rangle$.

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}

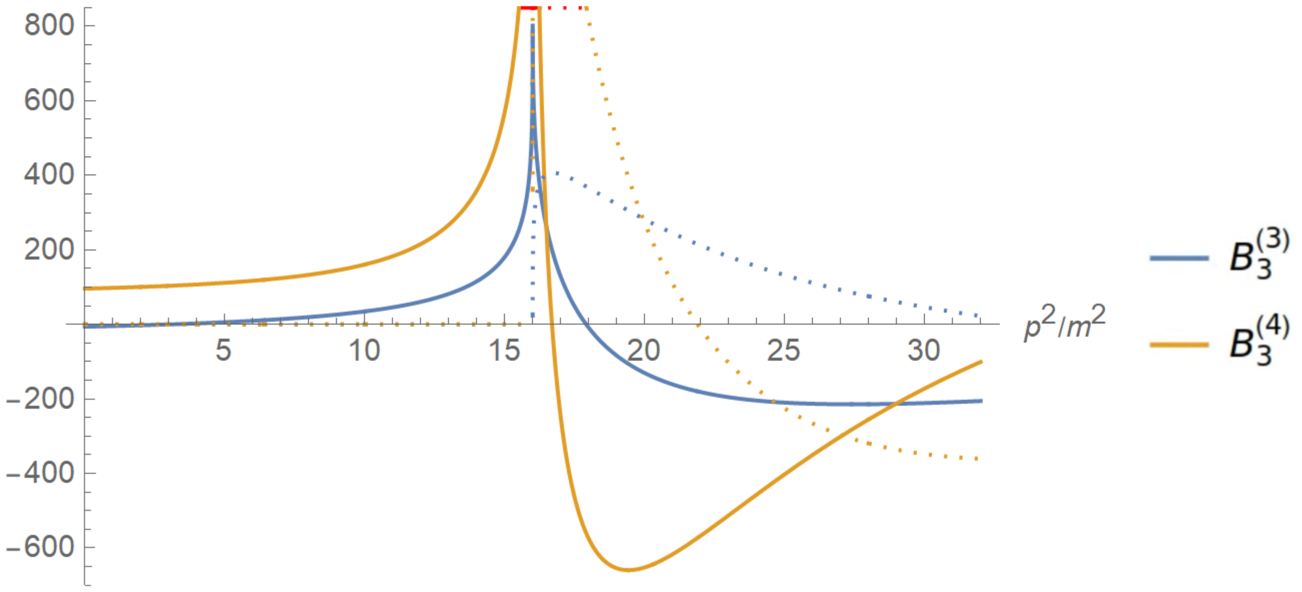
DiffExp: Total error estimate: 5.14483×10^{-31}

DiffExp: Integrating segment: $\langle | t \to -1. + 1.\,x$ ...

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 → {"B3^(3)", "B3^(4)"}, AxesLabel → {"p^2/m^2"}, PlotRange → {-700, 850},
  MaxRecursion → 15, WorkingPrecision → 100]]
```



3-loop banana graph

- Timing:
  - Moving from $p^2 = -\infty$ to $p^2 = 30$ at a precision of 25 digits takes about 90 sec, where we computed the top sector integrals up to and including order ϵ^3.
  - Moving from $p^2 = -\infty$ to $p^2 = 30$ at a precision of 100 digits takes a bit under 20 min, where we computed the top sector integrals up to and including order ϵ^3.
  - Obtaining 100+ digits at $p^2 = -100$ up to and including order ϵ^3 takes about 2.5 min.

- $B_3^{(k)}$:

```
0
4.082413202704059607801991461045097339855501253774222434496563798314848283907330199489603248642178129
-0.77131509152278575462585596925436762983509391519807746079082772367699344909736120048660363407870260 38
-15.522685324165188555766965480194336177309375782260392074283020085862627674041835486196067437962390 99
78.12509728148001692986790482079302619114776011817121195506011258285334682242128391076363566162968586
```

3-Loop banana graph

- We may also compute the fully unequal mass case. We choose the basis:

$$\vec{B}^{\text{banana}} = \left\{ \begin{array}{l} \epsilon I^{\text{banana}}_{1122},\ \epsilon I^{\text{banana}}_{1212},\ \epsilon I^{\text{banana}}_{1221},\ \epsilon I^{\text{banana}}_{2112},\ \epsilon I^{\text{banana}}_{2121},\ \epsilon I^{\text{banana}}_{2211}, \\ \epsilon(1+3\epsilon) I^{\text{banana}}_{1112},\ \epsilon(1+3\epsilon) I^{\text{banana}}_{1121},\ \epsilon(1+3\epsilon) I^{\text{banana}}_{1211}, \\ \epsilon(1+3\epsilon) I^{\text{banana}}_{2111},\ \epsilon(1+3\epsilon)(1+4\epsilon) I^{\text{banana}}_{1111}, \\ \epsilon^3 I^{\text{banana}}_{0111},\ \epsilon^3 I^{\text{banana}}_{1011},\ \epsilon^3 I^{\text{banana}}_{1101},\ \epsilon^3 I^{\text{banana}}_{1110} \end{array} \right\}$$

- We provide 55 digits of basis integral B_{11} below, in the point

$$(p^2 = 50, m_1^2 = 2, m_2^2 = 3/2, m_3^2 = 4/3, m_4^2 = 1)$$

$$\begin{aligned} B_{11}^{(0)} &= 0 \\ B_{11}^{(1)} &= 5.197252113696504317012957853856365240561893912238907864 5 \\ &\quad + i\, 6.875516953539020750137068564553890229955902455183095659 4 \\ B_{11}^{(2)} &= -17.95801081120940608995233616989284789487806870538990757 33 \\ &\quad + i\, 31.74367036336930909084029322990119719135089506494942310 47 \\ B_{11}^{(3)} &= -121.5101152068177565203392807541216084962880772908306370 668 \\ &\quad - i\, 40.76907623602027664537759999171722265374282585291457547 46 \\ B_{11}^{(4)} &= 125.6113388023605534745593764004798958232118632681257073 923 \\ &\quad - i\, 229.9200257172388589952062757571215176834471783495112755 027 \end{aligned}$$

These results were obtained in about 20 minutes on a single CPU-core

Further automatization

- In the previous example, the boundary conditions were provided as closed-form expressions in ϵ. In general, this requires a manual case-by-case analysis using expansion by regions in the parametric representation.

[See works by Beneke and Smirov] & [Jantzen, Smirnov, Smirnov, 1206.0546] for the asy.m package

- Furthermore, the basis was chosen such that the differential equations are finite (and also in precanonical form $\mathbf{A}_0 + \epsilon\mathbf{A}_1$.)
- More generally, we would like to derive the basis, differential equations and boundary terms in an automated way.

An automated computational strategy

- Find a basis of (quasi-)finite Feynman integrals.
- Derive a closed linear system of differential equations for the basis.
- Rescale integrals by powers of ϵ to make the differential equations finite in ϵ.
- Compute boundary conditions in a Euclidean point by numerical integration.
- Obtain points in the physical region (and analytically continue) by numerically solving the differential equations using iterated series expansions.
- (Optional) upgrade the boundary conditions to a higher precision by analyzing behavior near thresholds and pseudo-thresholds.

Caesar package

- Together with J. Usovitsch, I am working on a Mathematica toolbox, Caesar, which automates all steps. It works by interfacing with various programs that are already on the market.

Kira 2.0:
[J. Klappert, F. Lange, P. Maierhöfer, J. Usovitsch, 2008.06494]

  - A finite basis is derived in an automated fashion by using Reduze to obtain candidate integrals and using Kira to select an independent set.

Reduze 2:
[A. von Manteuffel, C. Studerus, 2008.06494]

LiteRed 1.4:
[R.N. Lee, 1310.1145]

  - The differential equations are computed using inbuilt code, while the dimensional reduction relations are generated using LiteRed.

pySecDec:
[S. Borowka, G. Heinrich, S. Jahn, S.P. Jones, M. Kerner, J. Schlenk, T. Zirke, 1703.09692]

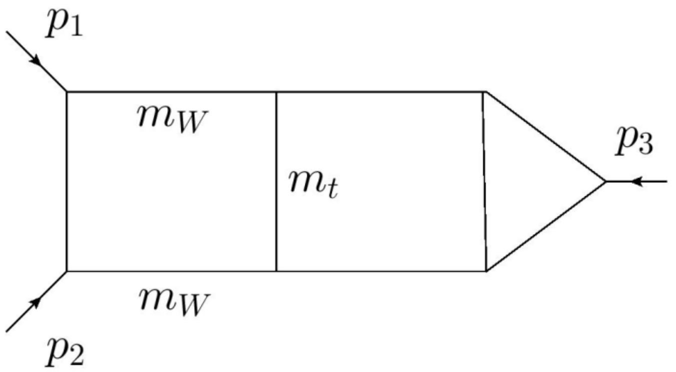
  - pySecDec is used to obtain numerical boundary conditions in the Euclidean region
  - DiffExp is used to obtain results everywhere else.

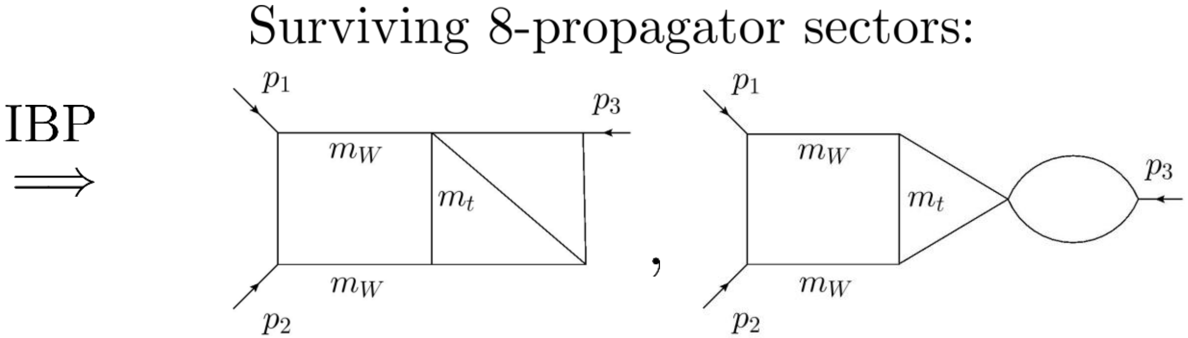
DiffExp:
[MH, 2006.05510]

Application: 3-loop vertex topology (relevant for EW pseudo-observables at Z-boson resonance)

In collaboration with:
[Ievgen Dubovyk, Ayres Freitas, Janusz Gluza, Krzysztof Grzanka, MH, Johann Usovitsch]

- We consider the 3-loop topology pictured below:





in the kinematic configuration: $p_1^2 = 0,\ p_2^2 = 0,\ p_1 \cdot p_2 = s/2$. We choose the following propagators:

$$\begin{aligned}
&D_1 = m_W^2 - k_3^2 && D_2 = -k_2^2 && D_3 = -k_1^2 && D_4 = -(k_1 - p_1 - p_2)^2 \\
&D_5 = -(k_2 - p_1 - p_2)^2 && D_6 = m_W^2 - (k_3 - p_1 - p_2)^2 && D_7 = -(k_3 - p_1)^2 && D_8 = m_t^2 - (k_3 - k_2)^2 \\
&D_9 = -(k_2 - k_1)^2 && N_{10} = -(k_1 - k_3)^2 && N_{11} = -(k_1 - p_2)^2 && N_{12} = -(k_2 - p_2)^2
\end{aligned}$$

- After IBP-reduction, the top sector collapses. The highest sectors remaining after IBP reduction have 8 propagators and are pictured in the top-right.

Example: 3-loop topology

- The (finite) basis consists of 77 integrals in total. We choose 19 integrals in $d = 4$, 53 integrals in $d = 6$, and 5 integrals in $d = 8$.

- We rescale the integrals by powers of ϵ in order to make the differential equations finite as $\epsilon \to 0$. The largest power we rescale by is ϵ^{-5}.

- We set up the system of differential equations, making use of IBP identities and dimensional recurrence relations. The differential equations are ~ 12 MB before expanding in ϵ.

Basis integrals

$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{4,2,2,2,2,0,0,0,0,0,0,0}$	$\left(\frac{1}{\epsilon}\right) I^{d=6-2\epsilon}_{3,2,2,2,2,1,0,0,0,0,0,0}$	$\left(\frac{1}{\epsilon}\right) I^{d=6-2\epsilon}_{2,2,2,2,2,1,1,0,0,0,0,0}$	$\left(\frac{1}{\epsilon^4}\right) I^{d=6-2\epsilon}_{4,0,2,2,0,0,0,4,0,0,0,0}$	$\left(\frac{1}{\epsilon^3}\right) I^{d=6-2\epsilon}_{3,2,2,2,0,0,0,2,0,0,0,0}$
$\left(\frac{1}{\epsilon^3}\right) I^{d=6-2\epsilon}_{3,0,2,2,2,0,0,2,0,0,0,0}$	$\left(\frac{1}{\epsilon^3}\right) I^{d=6-2\epsilon}_{3,0,2,2,1,0,0,3,0,0,0,0}$	$\left(\frac{1}{\epsilon^3}\right) I^{d=6-2\epsilon}_{2,0,2,2,2,0,0,3,0,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{0,2,2,2,2,0,0,4,0,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{3,0,2,2,0,1,0,4,0,0,0,0}$
$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{2,2,2,2,0,1,0,2,0,0,0,0}$	$I^{d=6-2\epsilon}_{2,2,2,2,1,1,0,1,0,0,0,0}$	$\left(\frac{1}{\epsilon^4}\right) I^{d=6-2\epsilon}_{0,2,2,2,0,0,2,3,0,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{2,0,2,2,2,0,1,2,0,0,0,0}$	$I^{d=6-2\epsilon}_{0,2,2,2,1,0,2,2,0,0,0,0}$
$I^{d=6-2\epsilon}_{0,2,2,2,1,0,1,3,0,0,0,0}$	$I^{d=6-2\epsilon}_{2,2,2,2,1,0,1,1,0,0,0,0}$	$I^{d=6-2\epsilon}_{2,1,2,2,2,0,1,1,0,0,0,0}$	$I^{d=6-2\epsilon}_{2,1,2,2,1,0,1,2,0,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{2,0,2,2,0,1,1,4,0,0,0,0}$
$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{1,2,2,2,0,1,1,2,0,0,0,0}$	$I^{d=6-2\epsilon}_{1,2,2,2,1,1,1,1,0,0,0,0}$	$\left(\frac{1}{\epsilon^4}\right) I^{d=8-2\epsilon}_{5,3,0,3,0,0,0,0,3,0,0,0}$	$\left(\frac{1}{\epsilon^3}\right) I^{d=8-2\epsilon}_{4,3,0,3,0,1,0,0,3,0,0,0}$	$\left(\frac{1}{\epsilon^3}\right) I^{d=8-2\epsilon}_{3,3,0,3,0,1,1,0,3,0,0,0}$
$\left(\frac{1}{\epsilon^5}\right) I^{d=6-2\epsilon}_{3,0,2,0,0,0,0,3,2,0,0,0}$	$\left(\frac{1}{\epsilon^5}\right) I^{d=6-2\epsilon}_{4,0,2,0,0,0,0,3,2,0,0,0}$	$\left(\frac{1}{\epsilon^5}\right) I^{d=6-2\epsilon}_{3,0,0,2,0,0,0,3,2,0,0,0}$	$\left(\frac{1}{\epsilon^5}\right) I^{d=6-2\epsilon}_{4,0,0,2,0,0,0,3,2,0,0,0}$	$\left(\frac{1}{\epsilon^5}\right) I^{d=6-2\epsilon}_{3,0,0,2,0,0,0,4,2,0,0,0}$
$\left(\frac{1}{\epsilon^5}\right) I^{d=6-2\epsilon}_{5,0,0,2,0,0,0,3,2,0,0,0}$	$\left(\frac{1}{\epsilon^4}\right) I^{d=8-2\epsilon}_{0,3,0,3,0,0,0,5,3,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{3,2,0,2,0,0,0,1,2,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{3,0,2,0,2,0,0,1,2,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{3,0,2,0,1,0,0,2,2,0,0,0}$
$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{2,0,2,0,2,0,0,2,2,0,0,0}$	$\left(\frac{1}{\epsilon^3}\right) I^{d=6-2\epsilon}_{2,0,2,0,0,1,0,3,2,0,0,0}$	$\left(\frac{1}{\epsilon^3}\right) I^{d=6-2\epsilon}_{1,0,2,0,0,2,0,3,2,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{2,2,0,2,0,1,0,1,2,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{2,1,0,2,0,1,0,2,2,0,0,0}$
$\left(\frac{1}{\epsilon}\right) I^{d=4-2\epsilon}_{1,0,1,1,0,1,0,2,1,0,0,0}$	$\left(\frac{1}{\epsilon}\right) I^{d=4-2\epsilon}_{2,0,1,1,0,1,0,2,1,0,0,0}$	$\left(\frac{1}{\epsilon}\right) I^{d=4-2\epsilon}_{1,0,1,1,0,1,0,3,1,0,0,0}$	$I^{d=4-2\epsilon}_{1,1,1,1,0,1,0,1,1,0,0,0}$	$\left(\frac{1}{\epsilon^5}\right) I^{d=8-2\epsilon}_{0,0,3,0,0,0,3,4,3,0,0,0}$
$\left(\frac{1}{\epsilon^3}\right) I^{d=6-2\epsilon}_{2,0,0,2,0,0,1,3,2,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{0,2,0,2,0,0,2,2,2,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{0,2,0,2,0,0,1,3,2,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{2,2,0,2,0,0,1,1,2,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{2,1,0,2,0,0,1,2,2,0,0,0}$
$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{1,2,0,2,0,0,2,1,2,0,0,0}$	$\left(\frac{1}{\epsilon^3}\right) I^{d=6-2\epsilon}_{0,0,2,1,0,0,2,3,2,0,0,0}$	$\left(\frac{1}{\epsilon^3}\right) I^{d=6-2\epsilon}_{0,0,2,2,0,0,2,3,2,0,0,0}$	$\left(\frac{1}{\epsilon^3}\right) I^{d=6-2\epsilon}_{0,0,2,1,0,0,2,4,2,0,0,0}$	$I^{d=4-2\epsilon}_{1,0,1,1,0,0,1,2,1,0,0,0}$
$\left(\frac{1}{\epsilon}\right) I^{d=4-2\epsilon}_{2,0,1,1,0,0,1,2,1,0,0,0}$	$\left(\frac{1}{\epsilon}\right) I^{d=4-2\epsilon}_{1,0,1,1,0,0,1,3,1,0,0,0}$	$\left(\frac{1}{\epsilon}\right) I^{d=4-2\epsilon}_{3,0,1,1,0,0,1,2,1,0,0,0}$	$\left(\frac{1}{\epsilon}\right) I^{d=4-2\epsilon}_{2,0,1,1,0,0,1,3,1,0,0,0}$	$I^{d=4-2\epsilon}_{0,1,1,1,0,0,1,2,1,0,0,0}$
$I^{d=4-2\epsilon}_{1,1,1,1,0,0,1,1,1,0,0,0}$	$I^{d=4-2\epsilon}_{2,1,1,1,0,0,1,1,1,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{2,0,2,0,2,0,1,1,2,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{2,0,2,0,1,0,1,2,2,0,0,0}$	$\left(\frac{1}{\epsilon}\right) I^{d=6-2\epsilon}_{1,0,2,0,2,0,2,1,2,0,0,0}$
$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{1,0,2,0,2,0,1,2,2,0,0,0}$	$I^{d=4-2\epsilon}_{1,0,1,1,1,0,1,1,1,0,0,0}$	$I^{d=4-2\epsilon}_{2,0,1,1,1,0,1,1,1,0,0,0}$	$I^{d=4-2\epsilon}_{1,0,1,1,1,0,1,2,1,0,0,0}$	$\left(\frac{1}{\epsilon^3}\right) I^{d=6-2\epsilon}_{1,0,2,0,0,1,1,3,2,0,0,0}$
$\left(\frac{1}{\epsilon^3}\right) I^{d=6-2\epsilon}_{2,0,2,0,0,1,1,3,2,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{1,2,0,2,0,1,1,1,2,0,0,0}$	$\left(\frac{1}{\epsilon^2}\right) I^{d=6-2\epsilon}_{1,1,0,2,0,1,1,2,2,0,0,0}$	$\left(\frac{1}{\epsilon}\right) I^{d=4-2\epsilon}_{1,0,1,1,0,1,1,2,1,0,0,0}$	$\left(\frac{1}{\epsilon}\right) I^{d=4-2\epsilon}_{2,0,1,1,0,1,1,2,1,0,0,0}$
$\left(\frac{1}{\epsilon}\right) I^{d=4-2\epsilon}_{1,0,1,1,0,1,1,3,1,0,0,0}$	$I^{d=4-2\epsilon}_{1,1,1,1,0,1,1,1,1,0,0,0}$			

Numerical boundary conditions using pySecDec

- When all basis integrals are finite, their numerical integration using pySecDec is sped up considerably.
- We compute all basis integrals in the Euclidean region in the point $s = -2, m_W^2 = 4, m_t^2 = 16$, using the Qmc integrator configured with:

```
lib.use_Qmc(minn=10**7, maxeval=10**9, transform='korobov3', epsabs=1e-12, cputhreads=16)
```

- The computation took between 1/2-1 day on a Ryzen Threadripper Pro 3955WX.
- We find for example: $\mathrm{I}_{1,1,1,1,0,1,1,1,1} = 0.13395266651743990 - 0.13899149646580500\,\epsilon + O(\epsilon^2)$
  $\pm\left(2.\times 10^{-10} + 7.\times 10^{-10}\,\epsilon\right)$

Results in the physical region, using DiffExp

- Using DiffExp we may transport from the Euclidean point to any other (real) point in phase-space.

- Transporting from $(s, m_W^2, m_t^2) = (-2,4,16)$ to $(s, m_W^2, m_t^2) = \left(1, \left(\frac{401925}{455938}\right)^2, \left(\frac{433000}{227969}\right)^2\right)$, we obtain:

$$I^{d=6-2\epsilon}_{1,1,0,2,0,1,1,2,2,0,0,0} = (0.125019 + 0.0127438\, i) - (0.334035 - 0.0731341\, i)\,\epsilon + (1.81433 + 0.208055\, i)\,\epsilon^2 - (6.08263 - 0.389921\, i)\,\epsilon^3 + O(\epsilon^4)$$

$$I^{d=4-2\epsilon}_{1,0,1,1,0,1,1,2,1,0,0,0} = (1.17171 + 1.03298\, i) - (3.13434 - 1.43328\, i)\,\epsilon + (5.9312 + 3.04346\, i)\,\epsilon^2 + O(\epsilon^3)$$

$$I^{d=4-2\epsilon}_{2,0,1,1,0,1,1,2,1,0,0,0} = (0.912403 + 0.837335\, i) - (1.66844 - 1.83869\, i)\,\epsilon + (2.25671 + 3.31779\, i)\,\epsilon^2 + O(\epsilon^3)$$

$$I^{d=4-2\epsilon}_{1,0,1,1,0,1,1,3,1,0,0,0} = (0.102616 + 0.123891\, i) - (0.137177 - 0.313638\, i)\,\epsilon - (0.0575107 - 0.560502\, i)\,\epsilon^2 + O(\epsilon^3)$$

$$I^{d=4-2\epsilon}_{1,1,1,1,0,1,1,1,1,0,0,0} = (1.30731 + 3.42323\, i) - (10.0551 - 8.533\, i)\,\epsilon + O(\epsilon^2)$$

- The computation involved 16 line segments and took 45 minutes on a single CPU core. The precision of the expansions was 10^{-17}, exceeding the precision of the boundary conditions.

Results in the physical region, using DiffExp

- We find that the numerical error of the boundary conditions approximately carries over after transporting from the Euclidean to the physical point.

- For example, at $(s, m_W^2, m_t^2) = (-2,4,16)$ we have:

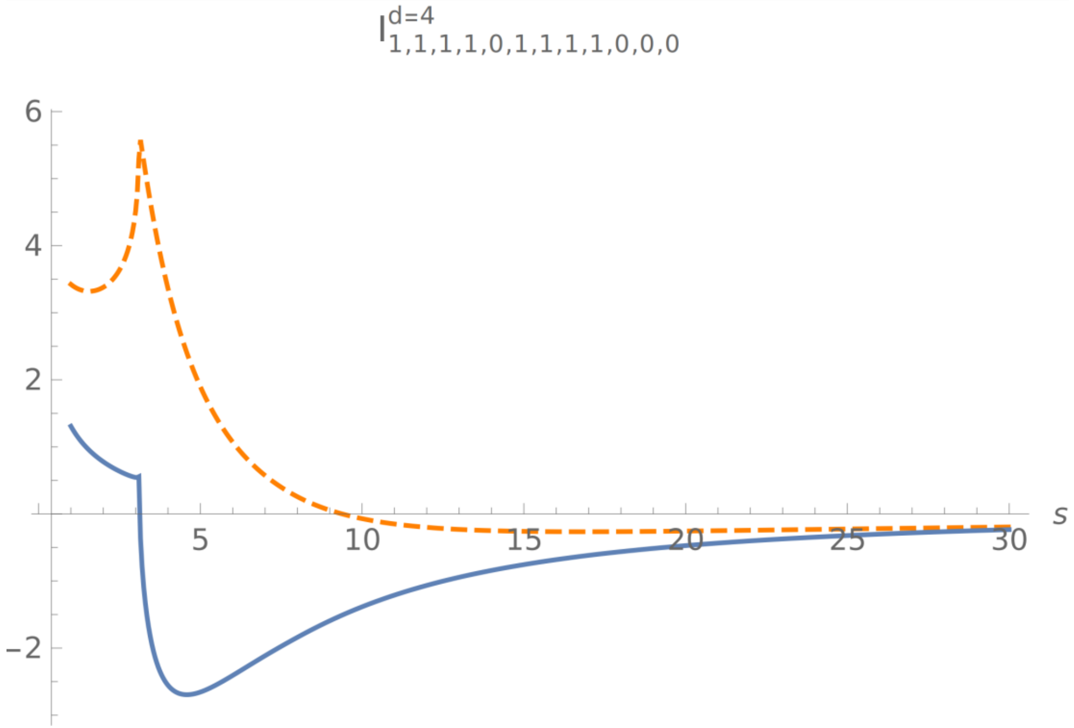
$$\mathrm{I}^{\mathrm{d}=4-2\epsilon}_{1,1,1,1,0,1,1,1,1,0,0,0} = +0.133952666651744 \pm 2 \times 10^{-10}$$

- While at $(s, m_W^2, m_t^2) = \left(1, \left(\frac{401925}{455938}\right)^2, \left(\frac{433000}{227969}\right)^2\right)$ we have:

$$\mathrm{I}^{\mathrm{d}=4-2\epsilon}_{1,1,1,1,0,1,1,1,1,0,0,0} = (1.30730596404577 + 3.42322623988039i) \pm (3 \times 10^{-11} + 2 \times 10^{-9} i)$$

Results in the physical region, using DiffExp

- By concatenating series expansions along line segments, we can plot the results along a line. For example:



- It took about 2 hour and 15 minutes to obtain the results along this line, at a precision of $\sim 10^{-13}$.
- Afterwards, evaluating an integral anywhere along the line takes about 0.01 seconds.

Optional: upgrading the boundary conditions

- Suppose we want to go beyond the precision that pySecDec can provide in the Euclidean region. It turns out that we can lift the boundary conditions to a higher precision by looking at the scaling of the integrals near (pseudo-)thresholds.

- We don't have to use expansion by regions. Instead, we take the numerical boundary conditions, move around in phase-space and record at which locations there are branch-points or singularities.

See also:
[D. Chicherin, T. Gehrmann, J. M. Henn, N. A. Lo Presti, V. Mitev, P. Wasser, 1809.06240]
[Abreu, Ita, Moriello, Page, Tschernow, Zeng, 2005.04195]

- In particular, for each line segment we record presence or absence of terms of the form of x^{-n}, $x^{-n/2}$ and $\log(x)^m$, where we let $n \leq 0$.

- Because the boundary conditions are of finite precision, such terms may carry coefficients of the form 10^{-10} which we will interpret to be 0 exactly.

Optional: upgrading the boundary conditions

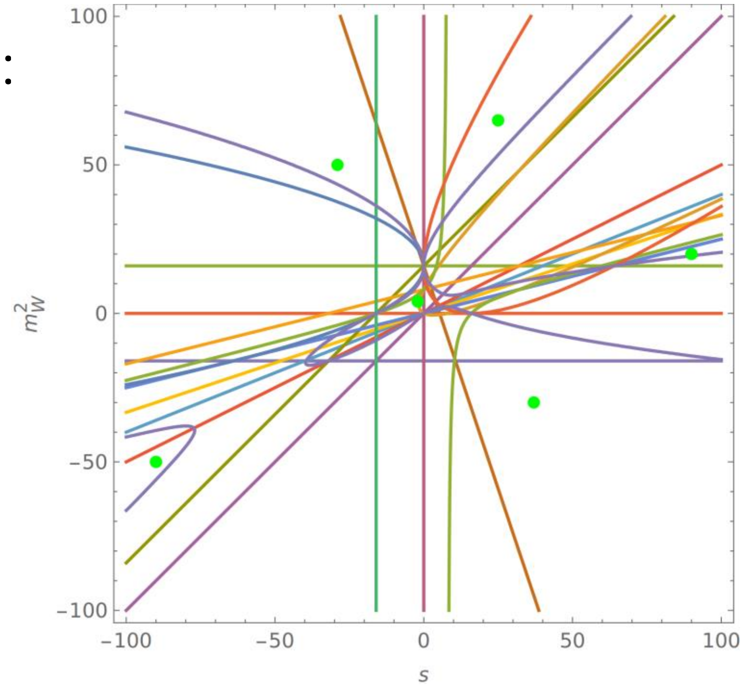
- We get a feeling for which directions to move towards, by looking at the poles of the differential equations. The differential equations have the following poles:

```
mt,                                   mw,                                      s,                        mt + mw,
mt + s,                               mt - mw,                                 mw - s,                   mt - mw + s,
5*mw - 2*s,                           2*mw - s,                                3*mw - s,                 4*mw - s,
mt - mw - 3*s,                        2*mt - 4*mw + s,                         mt^2 - 2*mt*mw + mw^2 + mt*s,   mt^2 - 2*mt*mw + 4*mw*s - s^2,
mt^2 - 2*mt*mw + mw^2 - 2*mt*s - 2*mw*s + s^2,   2*mt*mw - 4*mw^2 + mt*s + 6*mw*s - 2*s^2,
```

$2\,mt^4\,mw - 8\,mt^3\,mw^2 + 12\,mt^2\,mw^3 - 8\,mt\,mw^4 + 2\,mw^5 - mt^4\,s + 5\,mt^3\,mw\,s - 9\,mt^2\,mw^2\,s + 7\,mt\,mw^3\,s - 2\,mw^4\,s - mt^2\,mw\,s^2 - 3\,mt\,mw^2\,s^2 + mt^2\,s^3$

- For example, with $m_t^2 = 16$, we obtain the following contour plot:
- The green dots represents points between which

we transport. In particular, we consider lines from the Euclidean point $(s, m_W^2, m_t^2) = (-2,4,16)$, towards the outer green points. The points have been chosen in order to cross as many of the poles as possible.



Optional: upgrading the boundary conditions

- Adding two additional points that cross $m_t = 0$ as well, we end up with the following 8 points to which we transport from $(s, m_W^2, m_t^2) = (-2,4,16)$:

$$\{s \to -29,\ m_W^2 \to 50,\ m_t^2 \to 17\} \qquad \{s \to -90,\ m_W^2 \to -50,\ m_t^2 \to 17\}$$
$$\{s \to 25,\ m_W^2 \to 65,\ m_t^2 \to 17\} \qquad \{s \to 90,\ m_W^2 \to 20,\ m_t^2 \to 17\}$$
$$\{s \to 37,\ m_W^2 \to -30,\ m_t^2 \to 17\} \qquad \{s \to -26,\ m_W^2 \to -18,\ m_t^2 \to -10\}$$
$$\{s \to -22,\ m_W^2 \to 20,\ m_t^2 \to -10\} \qquad \{s \to 70,\ m_W^2 \to 40,\ m_t^2 \to -10\}$$

Optional: upgrading the boundary conditions

- Next, we repeat the computation with a set of unfixed boundary conditions:

0	0	0	$c_{1,4}$	$c_{1,5}$	$c_{1,6}$	$c_{1,7}$
0	0	0	0	$c_{2,5}$	$c_{2,6}$	$c_{2,7}$
0	0	0	0	$c_{3,5}$	$c_{3,6}$	$c_{3,7}$
0	$c_{4,2}$	$c_{4,3}$	$c_{4,4}$	$c_{4,5}$	$c_{4,6}$	$c_{4,7}$
0	0	$c_{5,3}$	$c_{5,4}$	$c_{5,5}$	$c_{5,6}$	$c_{5,7}$
0	0	$c_{6,3}$	$c_{6,4}$	$c_{6,5}$	$c_{6,6}$	$c_{6,7}$
0	0	$c_{7,3}$	$c_{7,4}$	$c_{7,5}$	$c_{7,6}$	$c_{7,7}$
0	0	$c_{8,3}$	$c_{8,4}$	$c_{8,5}$	$c_{8,6}$	$c_{8,7}$
0	0	0	$c_{9,4}$	$c_{9,5}$	$c_{9,6}$	$c_{9,7}$
0	0	0	$c_{10,4}$	$c_{10,5}$	$c_{10,6}$	$c_{10,7}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	0	0	$c_{68,6}$	$c_{68,7}$
0	0	0	0	0	$c_{69,6}$	$c_{69,7}$
0	0	$c_{70,3}$	$c_{70,4}$	$c_{70,5}$	$c_{70,6}$	$c_{70,7}$
0	0	$c_{71,3}$	$c_{71,4}$	$c_{71,5}$	$c_{71,6}$	$c_{71,7}$
0	0	0	$c_{72,4}$	$c_{72,5}$	$c_{72,6}$	$c_{72,7}$
0	0	0	$c_{73,4}$	$c_{73,5}$	$c_{73,6}$	$c_{73,7}$
0	0	0	0	$c_{74,5}$	$c_{74,6}$	$c_{74,7}$
0	0	0	0	$c_{75,5}$	$c_{75,6}$	$c_{75,7}$
0	0	0	0	$c_{76,5}$	$c_{76,6}$	$c_{76,7}$
0	0	0	0	0	$c_{77,6}$	$c_{77,7}$

- Lastly, we impose the same behavior around the singular points, which fixes the coefficients:

0.	0.	0.	0.0104167	−0.00525246	0.0344235	−0.023964
0.	0.	0.	0.	0.00970283	−0.0055748	0.0324391
0.	0.	0.	0.	0.0148169	−0.0007125	0.0491314
0.	0.000217014	−0.000994722	0.00289226	−0.00649465	0.012447	$c_{4,7}$
0.	0.	0.00883742	−0.0437098	0.155112	−0.438363	1.10301
0.	0.	0.00861711	−0.0431707	0.153748	−0.436032	1.09904
0.	0.	0.00713239	−0.0443696	0.173696	−0.53395	1.4213
0.	0.	0.00547568	−0.0308585	0.118034	−0.356945	0.945135
0.	0.	0.	0.00260417	−0.00492326	0.0129286	−0.0203394
0.	0.	0.	0.000202142	−0.000940769	0.00275984	−0.00624196
⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.	0.	0.	0.	0.	0.138799	−0.384399
0.	0.	0.	0.	0.	0.0413123	−0.0991391
0.	0.	0.0171007	−0.172654	1.06597	−5.11074	$c_{70,7}$
0.	0.	0.000711127	−0.00496931	0.0221132	−0.0760124	$0.414657 - 0.00910467\, c_{70,7}$
0.	0.	0.	0.0668526	−0.323007	1.56549	$-1.\, c_{49,7} - 0.790243\, c_{70,7} + 4.6316$
0.	0.	0.	0.0211336	−0.170294	0.931839	$0.0474158\, c_{70,7} - 5.06544$
0.	0.	0.	0.	0.0544231	−0.289769	1.13232
0.	0.	0.	0.	0.00711423	−0.0309455	0.110314
0.	0.	0.	0.	0.00118868	−0.00406291	0.0123885
0.	0.	0.	0.	0.	0.133953	$c_{77,7}$

Optional: upgrading the boundary conditions

- We see that order ϵ^6 has not been fully determined, and we would need to expand up to order ϵ^7 in order to fully fix this order.

- Furthermore, we manually added high precision results for the basis integrals 1, 4, 23 and 26:

$$\left(\frac{1}{\epsilon^2}\right) \mathrm{I}^{d=6-2\epsilon}_{4,2,2,2,2,0,0,0,0,0,0,0}, \left(\frac{1}{\epsilon^4}\right) \mathrm{I}^{d=6-2\epsilon}_{4,0,2,2,0,0,0,4,0,0,0,0}, \left(\frac{1}{\epsilon^4}\right) \mathrm{I}^{d=8-2\epsilon}_{5,3,0,3,0,0,0,0,3,0,0,0}, \left(\frac{1}{\epsilon^5}\right) \mathrm{I}^{d=6-2\epsilon}_{3,0,2,0,0,0,0,3,2,0,0,0}$$

  which were obtained by integrating the Feynman parametrization analytically.

- We performed the lifting procedure twice by transporting along different lines, in order to check consistency of the results. We obtain the following (preliminary) results at $(s, m_W^2, m_t^2) = (-2,4,16)$:

$$\mathrm{I}^{d=4}_{1,1,1,1,0,1,1,1,1,0,0,0} = 0.1339526664441601839027498 12$$

  at an expected precision of about 10^{-25}.

Conclusions

- Without spending significant effort on simplification of the basis, we can numerically solve the differential equations of non-trivial 3-loop Feynman integrals.
- By choosing the basis representatives to be finite integrals, we can obtain precise numerical boundary conditions in the Euclidean region using pySecDec.
- We find that the precision of the boundary conditions in the Euclidean region carries over to the physical region.
- We can upgrade the boundary conditions to a higher precision by reading of the scaling behavior of the integrals around singular points.
- The process can be almost fully automated.

Thank you for listening!