Project Caesar, a blueprint for numerical evaluation of Feynman integrals

(common work with: Ayres Freitas, Janusz Gluza, Krzysztof Grzanka, Martijn Hidding, Ievgen Dubovik)

Theory QCD coffee seminar

Johann Usovitsch

20. October 2021

Outline

1. Introduction
2. Caesar: blueprint for numerical evaluation of Feynman integrals
3. Benchmark
   - v3t181
   - Bhabha
   - BoxPentagon
4. Summary

Introduction

Many formal successfull studies are available on the market.

- Loop tree duality [Capatti, Hirschi, Pelloni, Ruijl, 2021]
- Unitarity cut techniques [Abreu, Ita, Page,Tschernow,2021]
- pySecDec approach [Long Chen, Heinrich, Jones, Kerner, Klappert, Schlenk, 2021]
- Auxiliary mass flow [Brønnum-Hansen, Melnikov, Quarroz, Chen-YuWang, 2021]
- Solving a system of differential equations numerically [Lee, Smirnov, Smirnov, 2018], [Mandal, Zhao, 2019], [Moriello, 2019], [Bonciani, Del Duca, Frellesvig, Henn, Hidding, Maestri, Moriello, Salvatori, Smirnov, 2019], [Hidding, 2020], [Abreu, Ita, Moriello, Page, Tschernow, Zeng 2020]
- For the full automation some glue is missing.

We demonstrate one possible full automation of the differential equations method approach.

Electroweak Precision Physics

	Experiment	Theory uncertainty	Main source
$M_W\,[\mathrm{MeV}]$	80385 ± 15	4	$N_f^2\alpha^3$, $N_f\alpha^2\alpha_\mathrm{s}$
$\sin^2\theta^\mathrm{l}_\mathrm{eff}\,[10^{-5}]$	23153 ± 16	4.5	$N_f^2\alpha^3$, $N_f\alpha^2\alpha_\mathrm{s}$
$\Gamma_Z\,[\mathrm{MeV}]$	2495.2 ± 2.3	0.4	$N_f^2\alpha^3$, $N_f\alpha^2\alpha_\mathrm{s}$, $\alpha\alpha_\mathrm{s}^2$
$\sigma^0_\mathrm{had}\,[\mathrm{pb}]$	41540 ± 37	6	$N_f^2\alpha^3$, $N_f\alpha^2\alpha_\mathrm{s}$
$R_b = \Gamma_Z^b/\Gamma_Z^\mathrm{had}\,[10^{-5}]$	21629 ± 66	15	$N_f^2\alpha^3$, $N_f\alpha^2\alpha_\mathrm{s}$

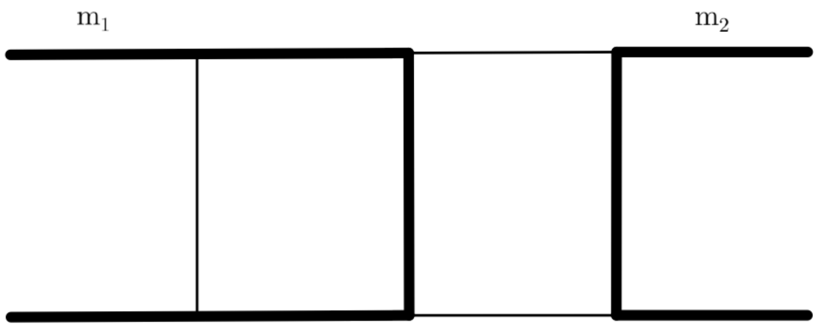
- The number of Z-bonos collected at LEP is 1.7×10^7
- Many pseudo observables are determined with high precision
- Present theoretical predictions are accurate enough to fullfill experimental demands

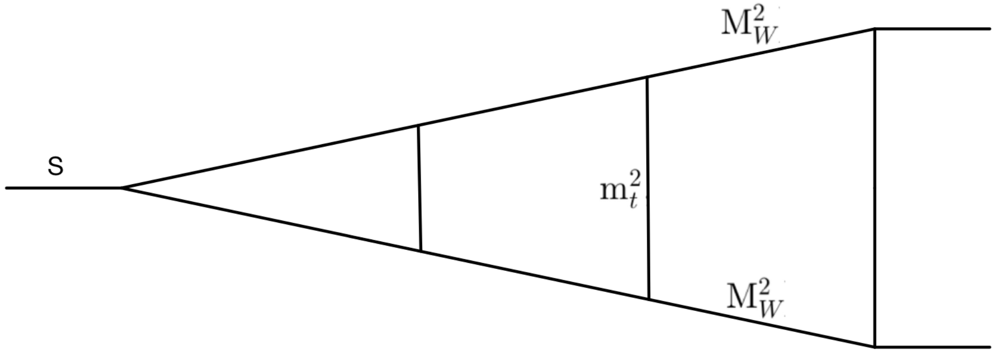
Overview Experiment Future

	Experiment uncertainty			Theory uncertainty	
	ILC	**CEPC**	**FCC-ee**	**Current**	**Future**
M_W [MeV]	3-4	3	1	4	1
$\sin^2\theta^{\mathrm{l}}_{\mathrm{eff}}[10^{-5}]$	1	2.3	0.6	4.5	1.5
Γ_Z [MeV]	0.8	0.5	0.1	0.4	0.2
$R_b[10^{-5}]$	14	17	6	15	7

- The concepts for the new experiments will have new demands to the theoreticle predictions
- **FCC-ee** will generate 5×10^{12} Z-bosons which is 10^5 more than during the LEP times
- The projection to the theory errors in the future assumes that the missing corrections $\alpha\alpha_{\mathrm{s}}^2$, $N_f^2\alpha^3$, $N_f\alpha^2\alpha_{\mathrm{s}}$ will become available

Samples of two-loop and three-loop Feynman integrals





- We project all Feynman integrals to scalar integrals
- We need to compute all Feynman integrals only up to the finite order in $\epsilon = (4 - d)/2$, d the space time dimension
- At the end we want to make sure we are able to compute all three-loop Feynman integrals appearing in e.g. the $Z\bar{b}b$ vertex numerically with at least eight significant digits of accuracy in physical kinematic regions

Grading the difficulty of a computation

- The integrals depend on up to four dimensionless parameters

$$\left\{ \frac{M_H^2}{M_Z^2}, \frac{M_W^2}{M_Z^2}, \frac{m_t^2}{M_Z^2}, \frac{(s+i\delta)}{M_Z^2} \right\} \Big|_{s=M_Z^2} \tag{1}$$

- Many of them contain ultraviolet and infrared singularities, even though the divergences cancel in the final result
- Computations involve $\mathcal{O}(100)$ master integrals

Feynman integral

$$T(a_1,\ldots,a_N) = \int \left(\prod_{i=1}^{L} \mathrm{d}^d \ell_i\right) \frac{1}{P_1^{a_1} P_2^{a_2} \cdots P_N^{a_N}}, \quad N = \frac{L}{2}(L+1) + LE \tag{2}$$

- $P_j = q_j^2 - m_j^2$, $j = 1,\ldots,N$, are the inverse propagators
- The momenta q_j are linear combinations of the loop momenta ℓ_i, $i = 1,\ldots,L$ for an L-loop integral, and external momenta p_k, $k = 1,\ldots,E$ for $E+1$ external legs
- The m_j are the propagator masses
- The a_j are the (integer) propagator powers

Differential Equations

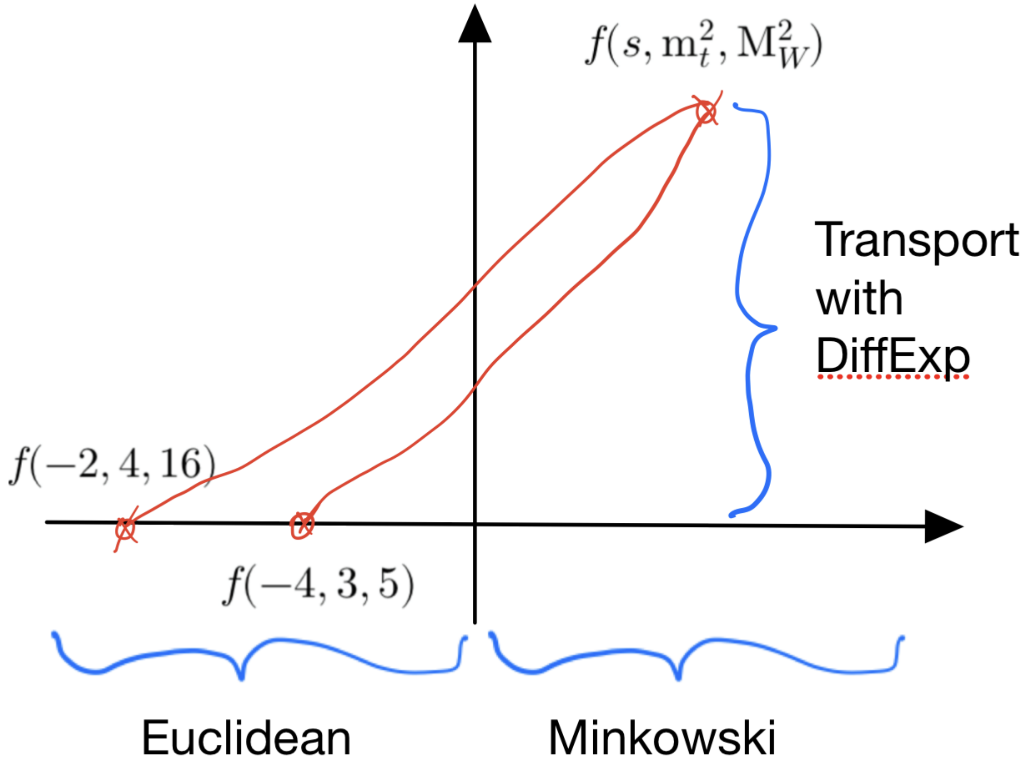
- Each family of Feynman integrals $T(a_1, \ldots, a_N)$ may be charcterized through a system of differential equations [Kotikov, 1991], [Remiddi, 1997][Gehrmann, Remiddi, 2000]
$$\partial_{s_i} \vec{f} = M_{s_i}(s_i, \epsilon) \vec{f} \quad (3)$$
and a set of master integrals $\vec{f}$
- We take derivatives in kinematic invariants and masses denoted as s_i in $\vec{f}$
- We express these derivatives again as a linear combination in terms of the same master integrals with the help of integration-by-parts identities [Chetyrkin, Tkachov, 1981]

The difficult part is to cast a physics problem in the form of Eq. (3). If this is done successfully, we have powerful tools to solve the physics problem.

Caesar: blueprint for numerical evaluation of Feynman integrals

- Developers team: Martijn Hidding and me.
- Basic idea: Caesar has an interface to Kira, Reduze 2 [Von Manteuffel, Studerus, 2012], pySecDec [Borowka et al., 2018] and DiffExp [Martijn Hidding, 2021].
- Kira - the backbone / major bottleneck of the Caesar project - solves linear system of equations
- Reduze 2 - finds candidates for a finite basis of master integrals
- pySecDec - computes these master integrals in Euclidean regions - boundary terms for the system of differential equations
- DiffExp - transports the Euclidean point to an arbitrary physical point
- Error estimate: repeat the chain of tools for different Euclidean point

One Possible Application of Caesar



- All master integrals $f_i(\ldots)$ are finite integrals (Reduze)
- Master integrals $f_i(\ldots)$ are evaluated numerically in Euclidean regions (pySecDec)
- System of differential equations is generated with (Kira)
- Use series expansion of the system of differential equations to transport from the Eucliden points to Minkowskien physical regions (DiffExp)

Benefits of the blueprint Caesar

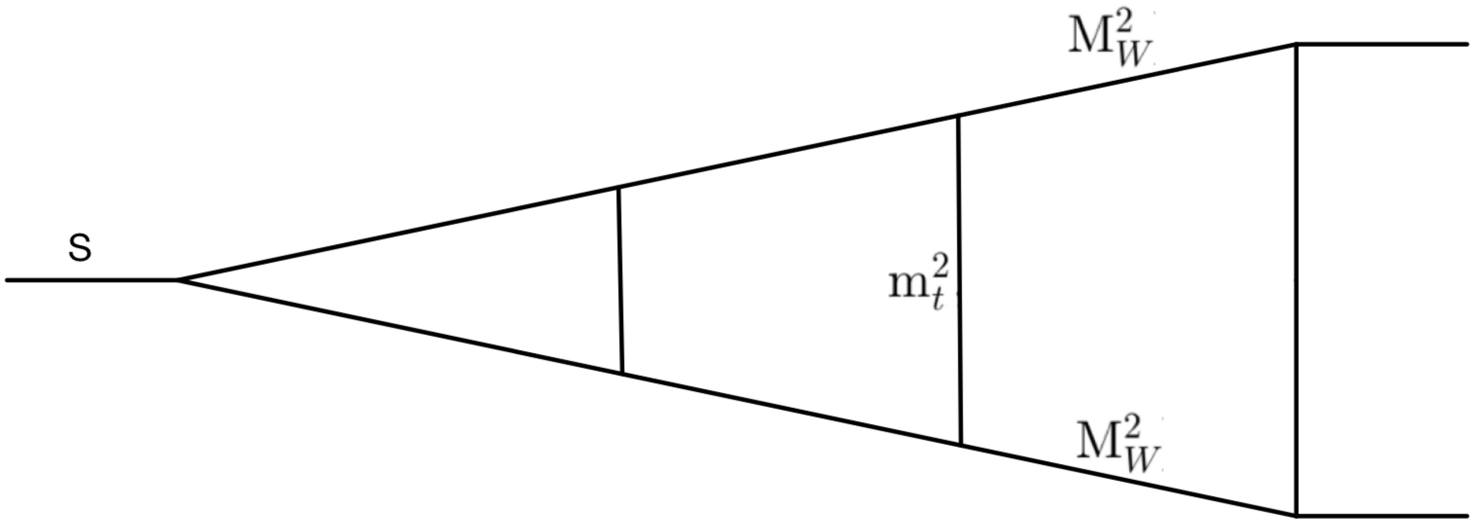
- Proof of concept available in other projects [Frellesvig, Hidding, Maestri, Moriello, Salvatori, 2020], [Faela, Lange, Sch ̀lönwalda, Steinhauser, 2021], [Canko,Gasparotto,Mattiazzi,Papadopoulos,Syrrakos,2021]
- We may set all masses to physical values — reductions with Kira simplify enormously
- Finite integrals in Euclidean regions — avoid the contour deformation and the tedious resolving of the UV or IR divergences

Drawbacks of the project Caesar

What if ...

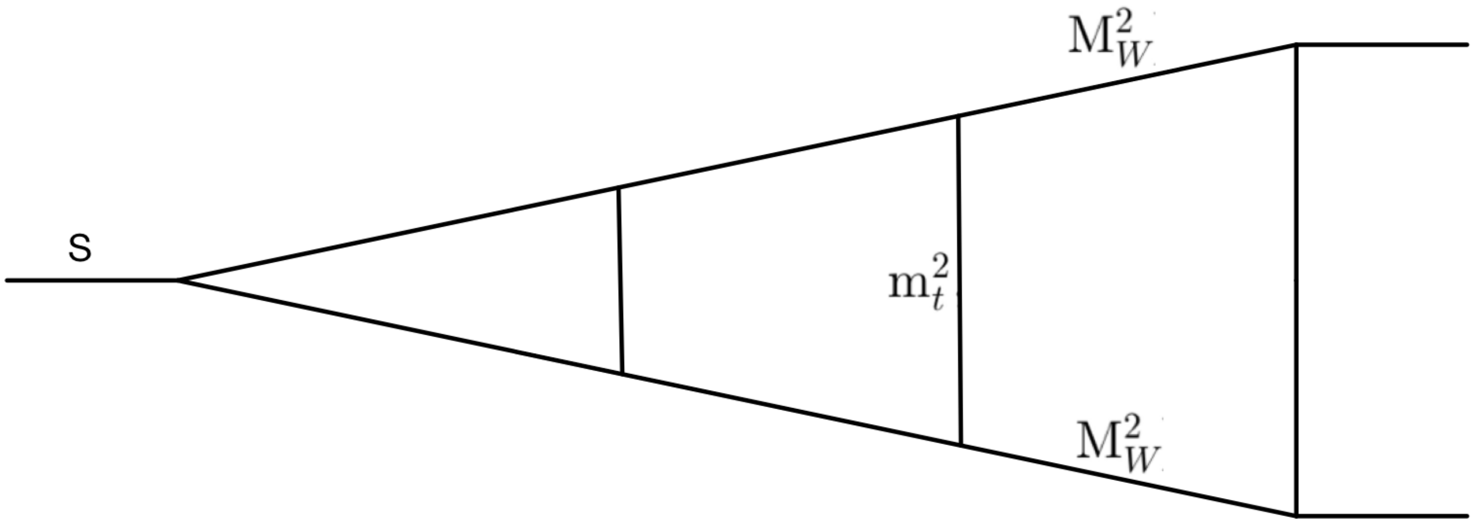
- no Euclidean region exist or no immediate comfortable kinematic point
- we encounter system of differential equations involving $\mathcal{O}(1000)$ of master integrals
-

Caesar: Integralfamily v3t181



- In Euclidean regions (s, M_W^2, m_t^2)=(-2,4,16)
  -> v3t181$^{d=4-2\epsilon}[1,1,1,1,0,1,1,1,1,0,0,0] =$
  $0.1339526664441601839027498 12$ with 25 significant digits
- At the present stage of the project this high accuracy was achieved semi-automatic :)

Caesar: Integralfamily v3t181

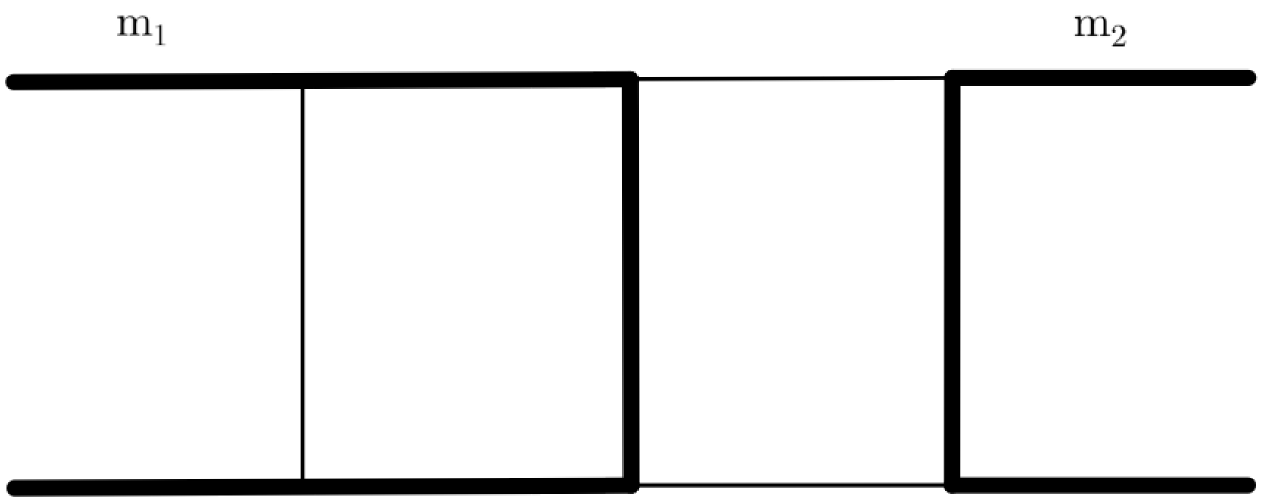


- In physical regions $(s, M_W^2, m_t^2)=(1,(\frac{401925}{4559382})^2, (\frac{433000}{227969})^2)$
  -> v3t181$^{d=4-2\epsilon}$[1, 1, 1, 1, 1, 1, 1, 1, 1, -3, 0, 0] $=$
  $$\frac{1.99999999981 + 8.18*10^{-12}\, i}{\epsilon^3} + \frac{9.87003934692 + 18.84955592198\, i}{\epsilon^2} - \frac{26.50733688118 - 41.19670709595\, i}{\epsilon} + (2.29574696253 + 201.06880202144\, i) + O(\epsilon)$$
- Fully automated following the blueprint Caesar

Few comments about v3t181

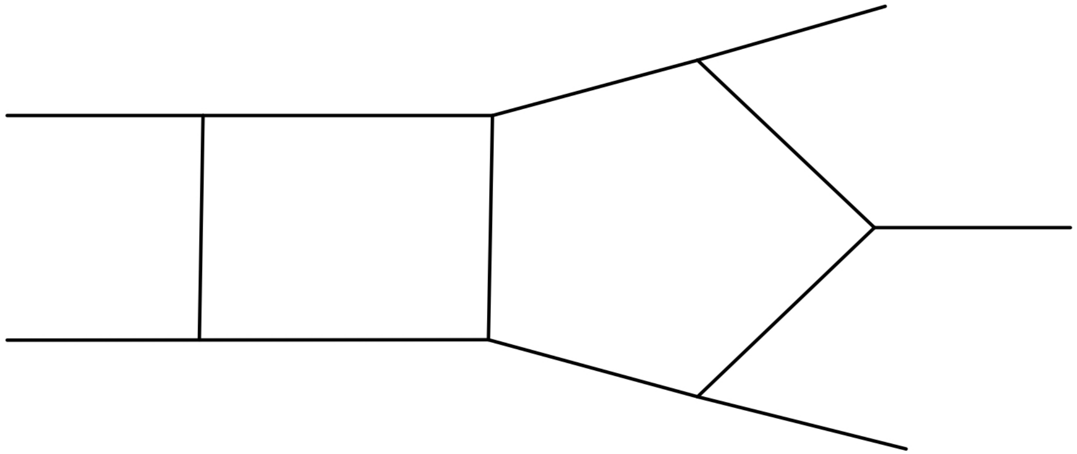
- The integral v3t181 has 77 master integrals all in different dimensions, d=4,6,8.
- Automatic resale of master integrals is implemented to meet the requirement that the matrix of system of differential equations is finite in the ϵ series expansion. The largest power is ϵ^{-5}
- The matrix is $\sim$3 MB big before expanding in ϵ

Caesar: Integralfamily Bhabha



- In physical regions (s, t, m_1^2, m_2^2)=(2,5,4,16)
  -> bhabha$^{d=6-2\epsilon}[1,2,1,2,1,1,1,0,0] =$
  $(0.0002973066815 + 0.001542581913\, i)$
  $-(0.002805345908 - 0.003106827180\, i)\, \epsilon + O\left(\epsilon^2\right)$
- Fully automated following the blueprint Caesar

Integralfamily BoxPentagon



- 61 master integrals - that is fine
- Fight for Euclidean regions - trivial automation?
  - Euclidean regions are topology dependent
- We apply expansion by regions for even faster numerical treatment
  - treat all $s_{ij} -> s_{ij}x$ but one s_{12}
- 5 scales - sounds like a lot
  - Apply the blueprint Caesar in an iterative way

Caesar: Iterative Approach

- Generate a system of differential equations for just one variable s_1 and set all other kinematic variables to numeric values $s_i = n_i$(rational number), $i \neq 1$
- The integration-by-parts reductions depend only on s_1 and d
- Evaluate all master integrals in a comfortable point numerically with pySecDec
- Transport the boundary terms with DiffExp to some useful value u_1 in the physical regions for s_1
- Generate a system of differential equations for the next variable s_2 and set all other kinematic variables to numeric values $s_i = n_i$(rational number), $i \neq 1, 2$ and $s_1 = u_1$(physical region)
- We skip the evaluation with pySecDec, since we know the new boundary terms from the last DiffExp call
- Transport the boundary terms with DiffExp to the next useful physical value u_2 for s_2
- Continue this pattern until all s_i are in physical regions

Caesar + Expansion by Regions with pySecDec

- With pySecDec apply expansion by regions
- In boxPentagon we treat all $s_{ij} -> s_{ij} x$ but one s_{12}
- We evaluate the integrals in $x = 10^{-19}$
- Integrals are finite in the general case
- Special case is where the x is an overall scaleless parameter - has to multiply the finite integrals by a $g(x, eps)$ function
- after the numerical computation of the boundary terms we transport the $x = 10^{-19}$ value to $x = 1$ value and continue with the iterative Caesar

Automate search for Euclidean regions

- In boxPentagon Euclidean regions are topology dependent
- e.g. if the propagator a_2 is positive we have Euclidean region if all $p_i p_j < 0$.
- If the propagator is contracted the Euclidean regions have different constraints.
- Apply Caesar blueprint to each different Euclidean region
- Example numerical value: $s_{12} = -4,\ s_{13} = -16,\ s_{14} = -32,\ s_{23} = -36,\ s_{24} = -46,\ x = 10^{-19}$
  boxPentagon$^{d=8-2\epsilon}[2,2,1,1,1,1,1,1,0,0,0] =$
  $0.07406413 + 0.31043095\ \epsilon$
- boxPentagon$^{d=8-2\epsilon}[2,0,1,0,1,2,1,2,0,0,0] =$
  $-3.33671556 - 0.26179939\ i$
  $+(-80.732153 - 11.62026\ i)\ \epsilon$
  $+(-1206.4608 - 258.3949\ i)\ \epsilon^2$

Euclidean regions do exist for boxPentagon (soon new results avaiable!):
$s_{12} = -1,\ s_{13} = -16,\ s_{14} = 101,\ s_{23} = -36,\ s_{24} = -46$

Outlook

- Get a basis where the matrix of the system of differential equations is linear in ϵ
  - -> DiffExp does order of magnitudes faster transport of the boundary terms
- Implement automated search for Euclidean regions
- Iterative application of the blueprint Caesar
- Generation of a whole grid for the function evaluation
- Implement block triangular form in Kira for faster amplitude/ deq evaluation
- Kira supports deformed propagators - have to check the application of Caesar also here

Conclusions

- The first physics goals are already in 6 month reach
- Important is the knowledge transfer and to get people motivated to engineer other methods for practical applications
- Strong computing resources are needed not only for the final product but also for the development of the tools.
- 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.
- The process can be fully automated.
- Possible list of applications possible in physics with Caesar is growing.