# Two-loop amplitude reduction in the HELAC framework

Based on work with Costas Papadopoulos, Giussepe Bevilacqua, Dhimiter Canko and Nikos Dokmetzoglou

Aris Spourdalakis

NCSR Demokritos

UoT THEP Seminar 29/4/2024

Monday April 29th

Hellenic Foundation for Research & Innovation

# Outline

- HELAC
- Amplitude reduction and construction at 1-loop
- 2-loop reduction: 4 vs D-dimensions
- Amplitude construction in D-dimension?
- Outlook

# HELAC

Costas Papadopoulos and Aggeliki Kanaki introduced HELAC [Kanaki and Papadopoulos, 2000]
This was an automated way to compute amplitudes using the Dyson-Swinger Equations i.e. recursively expressing n-point Green's Functions in terms of lesser point Green's Functions.

# What do we need for HELAC 2-loop?

3 steps:

- 1. Amplitude Construction
- **2. Amplitude Reduction at 2-loops (Focus of my talk today)**
- 3. Do the integrals

# Dyson Schwinger Equations

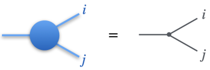



Figura: First Step of the recursion

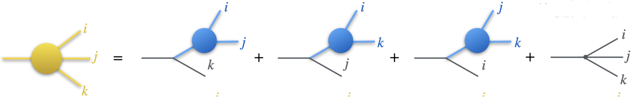



Figura: Second Step of the recursion

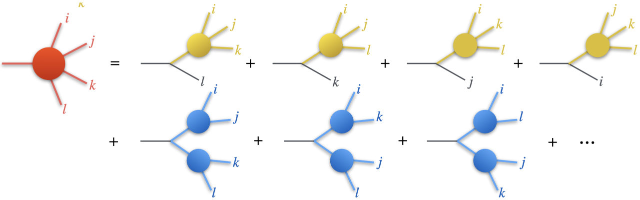



Figura: Third Step of the recursion

# Simple Example $f\bar{f} \to f\bar{f}b$

Binary ID assigned to each particle in powers of 2, so we can write our building blocks as $\psi(1), \bar{\psi}(2), \bar{\psi}(4), \psi(8), A(16)$.
Use this to build all the level 2 sub-amplitudes

- $A_\mu(12) = (ig)\Pi_\mu^\nu \bar{\psi}(4)\gamma_\nu\psi(8)$
- $\bar{\psi}(18) = (ig)\bar{\psi}(2)\not{A}(16)\mathcal{P}$
- $\bar{\psi}(20) = (ig)\bar{\psi}(4)\not{A}(16)\mathcal{P}$
- $\psi(24) = (ig)\mathcal{P}\not{A}(16)\psi(8)$

where $\Pi_{\mu\nu}$ is the boson propagator and $\mathcal{P}$ is the fermion propagator.

# Simple Example $f\bar{f} \to f\bar{f}b$

Continue by building the 3-level subamplitudes

- $A_\mu(12) = \Pi_\mu^\nu \bar{\psi}(4)\gamma_\nu\psi(8)$
- $A_\mu(28) = (ig)\Pi_\mu^\nu \left(\bar{\psi}(20)\gamma_\nu\psi(8) + \bar{\psi}(4)\gamma_\nu\psi(24)\right)$

4-level subamplitudes

- $\bar{\psi}_0(30) = (ig)\left(\bar{\psi}(2)\not{A}(28) + \bar{\psi}(14)\not{A}(16) + \bar{\psi}(18)\not{\psi}(12)\right)$

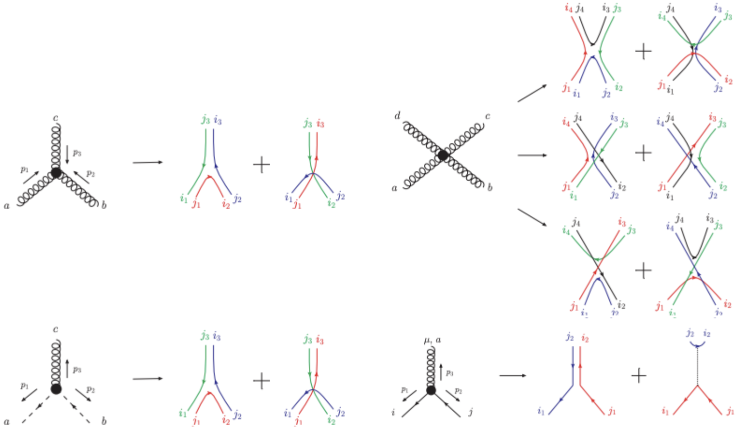
And so the amplitude is finally given by

$$\mathcal{A} = \bar{\psi}_0(30)\psi(1)$$

The recursion is made to always end and the Binary ID 1.
This is how HELAC calculates amplitudes!

# Color Flow representation

How do we deal with QCD?



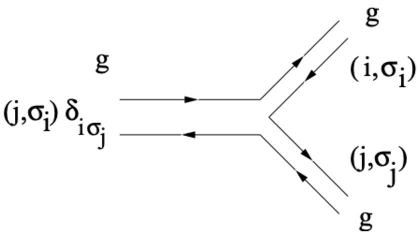
Each colored line corresponds to a Kronecker delta.

# Color Flow Example

How do we deal with QCD? Color connection representation: gluons are represented by a pair of color/anti-color indices $(i, j)$ and the quarks (anti-quarks) by a single color $(i, 0)$ (anti-color $(0, j)$) index, with $i, j \in (1, ..., N_C)$. All the other particles that do not carry color have

$$(0, 0)$$

. First term in the 3-gluon color factor:




or

$$\delta_{1\sigma_2}\delta_{2\sigma_3}\delta_{3\sigma_1}$$

and so on.

# Color Flow representation

Therefore, the color factor can in total be written as

$$\mathcal{F} = \delta_{1\sigma_I(1)}\delta_{2\sigma_I(2)}\dots\delta_{n\sigma_I(n)}$$

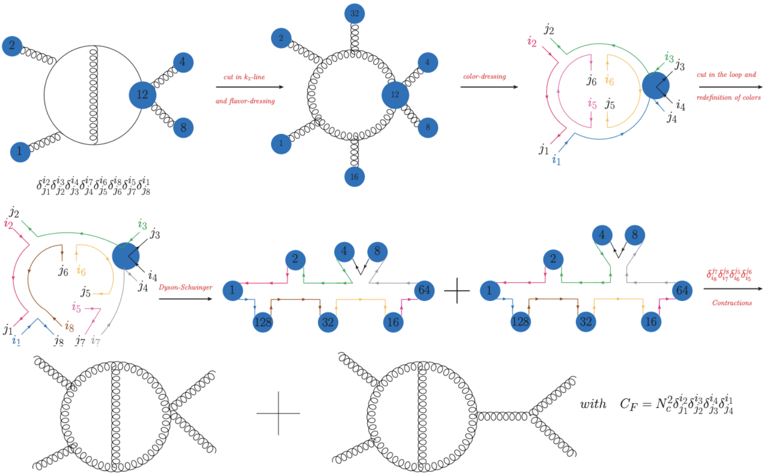
where $\sigma$ is some permutation of $1\dots n$ and $I = 1,\dots n!$
We can therefore write a general amplitude in the form

$$\mathcal{M} = \sum_{\sigma} \mathcal{F}\mathcal{A}_{\sigma}$$

where the color striped amplitude is properly calculated using appropriate Feynman rules.
**Benefits**:

- Computational complexity is polynomial
- Amplitude squared form is simply a product of Kroenecker deltas leading to a simple form.

# Amplitude Construction at 2-Loops



# Amplitude Construction at 2-Loops: Skeleton

The information for each amplitude is incorporated in the Skeleton

```
              INFO ==========================================
              INFO COLOR            9 out of         24
              INFO number of nums          0
              INFO ==========================================
              INFO COLOR           10 out of         24
              INFO number of nums        332
              INFO NUM            1  of         332           8
 INFO NUM        110  of         332           7
 INFO ==============================
INFO    4  80  35   9   1   1  16  35   5  64  35   7   0   0   0   0   1   2
INFO    4  12  35  10   1   1   4  35   3   8  35   4   0   0   0   0   1   1
INFO    4  92  35  11   1   2  12  35  10  80  35   9   0   0   0   0   1   1
INFO    5  92  35  11   2   2   4  35   3   8  35   4  80  35   9   0   1   5
INFO    4 124  35  12   1   1  32  35   6  92  35  11   0   0   0   0   1   2
INFO    4 126  35  13   1   1   2  35   2 124  35  12   0   0   0   0   1   1
INFO    4 254  35  14   1   1 128  35   8 126  35  13   0   0   0   0   1   2
INFO    6   1  12   1   2  12  35  35  35  35  35  35   0   0   0   0   5   9
```

From the first line to the second-to-last line, there is a sequence of sub-amplitudes accompanied by instructions for their computation.

# Amplitude Construction: Performance

Performace of amplitude generation for various processes

| Process | Loops | Loop-Flavors | Color | Skeleton Size | Timing | Numerators |
|---|---|---|---|---|---|---|
| $gg \to gg$ | 2 | $\{g, c, \bar{c}\}$ | Leading | 8.9 MB | 15.017s | 4560 |
| $gg \to gg$ | 2 | $\{g, q, \bar{q}, c, \bar{c}\}$ | Full | 110.6 MB | 6m 54.574s | 89392 |
| $gg \to q\bar{q}$ | 2 | $\{g, q, \bar{q}, c, \bar{c}\}$ | Full | 16.1 MB | 3m 14.509s | 13856 |
| $gg \to ggg$ | 2 | $\{g, c, \bar{c}\}$ | Leading | 300.0 MB | 21m 42.609s | 81480 |
| $gg \to gg$ | 1 | $\{g, q, \bar{q}, c, \bar{c}\}$ | Full | 537.8 kB | 2.386s | 768 |
| $gg \to ggg$ | 1 | $\{g, q, \bar{q}, c, \bar{c}\}$ | Full | 15.1 MB | 8m 53.349s | 11496 |
| $gg \to gggg$ | 1 | $\{g, c, \bar{c}\}$ | Leading | 394.0 MB | 104m 14.95s | 19680 |

Results obtained running 1-core in a personal laptop (i7 processor, 8-core, 25GB RAM).

So far, everything is done in 4 dimensions But, this is not necessary! The skeleton doesn't know about the number of dimensions!
*We will come back to this!*

# What do we need for HELAC 2-loop?

3 steps:

- 1. Amplitude Construction $\longrightarrow$**Done!** (At least in 4d)
- **2. Amplitude Reduction at 2-loops** (Focus of my talk today)
- 3. Do the integrals

# Amplitude reduction OPP

Reduction is done at the *integrand* level. OPP [Ossola et al., 2007] master formula for the 4-d numerator at 1 loop:

$$
\begin{aligned}
N(q) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} d\left(i_0 i_1 i_2 i_3\right) + \tilde{d}\left(q; i_0 i_1 i_2 i_3\right) \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} c\left(i_0 i_1 i_2\right) + \tilde{c}\left(q; i_0 i_1 i_2\right) \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} b\left(i_0 i_1\right) + \tilde{b}\left(q; i_0 i_1\right) \prod_{i\neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} a\left(i_0\right) + \tilde{a}\left(q; i_0\right) \prod_{i\neq i_0}^{m-1} D_i
\end{aligned}
$$

where $\tilde{d}$, $\tilde{c}$, $\tilde{b}$, $\tilde{a}$ are terms which vanish upon integration.

# Amplitude reduction OPP: Fitting

The system is solved by iteratively:

- Evaluate the numerator on values of $k$ for which $D_i(k) = 0$, starting from the first line of the previous equation, where 4 propagators are put on shell
- This condition by default sets to zero all the rest of the terms, allowing us to calculate $d$ and $\tilde{d}$
- Repeat for $c$, $\tilde{c}$ and so on until we fit all the coefficients

# Rational Terms

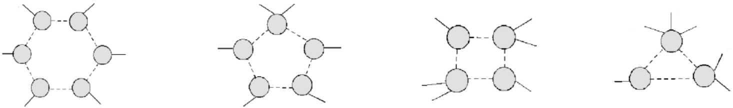
When doing this in $d = 4 - 2\epsilon$, we need to account for the mismatch of dimension between the left and right hand sides $\longrightarrow$ Rational Terms. Categorized into two terms [Ossola et al., 2008b]:

- R1, arising from the mismatch between the cancellation of D- and 4-dimensional propagators
- R2, arising from the extra dimensional parts of the loop momentum, metric tensor and gamma matrices

# 1 Loop Example

Example of a 6 point amplitude

$$\mathcal{A}(q) = \sum \underbrace{\frac{N_i^{(6)}(q)}{\bar{D}_{i_0}\bar{D}_{i_1}\cdots\bar{D}_{i_5}}} + \underbrace{\frac{N_i^{(5)}(q)}{\bar{D}_{i_0}\bar{D}_{i_1}\cdots\bar{D}_{i_4}}} + \underbrace{\frac{N_i^{(4)}(q)}{\bar{D}_{i_0}\bar{D}_{i_1}\cdots\bar{D}_{i_3}}} + \underbrace{\frac{N_i^{(3)}(q)}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}}} + \cdots$$

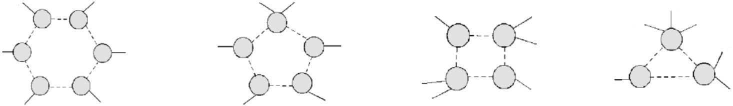


- HELAC numerically computes the numerators taking into account all flavors and colors that are consistent with the given process
- Loops are evaluated as $n+2$, like a tree level process with 2 extra legs from the cut
- The value of the loop momentum q is given by CutTools [Ossola et al., 2008a]

# 1 Loop Example

Example of a 6 point amplitude

$$\mathcal{A}(q) = \sum \underbrace{\frac{N_i^{(6)}(q)}{\bar{D}_{i_0}\bar{D}_{i_1}\cdots\bar{D}_{i_5}}} + \underbrace{\frac{N_i^{(5)}(q)}{\bar{D}_{i_0}\bar{D}_{i_1}\cdots\bar{D}_{i_4}}} + \underbrace{\frac{N_i^{(4)}(q)}{\bar{D}_{i_0}\bar{D}_{i_1}\cdots\bar{D}_{i_3}}} + \underbrace{\frac{N_i^{(3)}(q)}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}}} + \cdots$$



Now we can do the reduction:

- Calculate the coefficients of the OPP formula for each numerator
- Use Integration by Parts relations to get a Master Integral Basis
- Compute the rational terms

Has beed done! $\longrightarrow$ **HELAC One Loop!**[Bevilacqua et al., 2013]

# 2-Loop Amplitude: Reduction

Reduction is done at the *integrand* level. **The fit will be more complex!**
A generic 2-loop integrand can be written using the following scalar product set:

$$\{p_i \cdot p_j, k_i \cdot k_j, k_i \cdot p_j, k_i \cdot \eta_j\}$$

as well as any masses inside the loops. For the rest of the discussion we will ignore the case of massive loops.
The integrand therefore has the general form

$$\mathcal{R} = \frac{\mathcal{N}}{\mathcal{D}} = \frac{\sum_a c_a (z_1^{(a)})^{\beta_1} \ldots (z_{n_a}^{(a)})^{\beta_N}}{D_1 \ldots D_{N_p}} \tag{1}$$

where the $z_i$ are any of the scalar products in the set.

# 2-Loop Amplitude reduction

The above can we written in a more reduced form:

$$\mathcal{R} = \frac{\mathcal{N}}{\mathcal{D}} = \sum_{m=0}^{N_p} \sum_{\sigma} \frac{\sum_a \bar{c}_a (\bar{z}_1^{(a)})^{\alpha_1} \dots (\bar{z}_{N_a}^{(a)})^{\alpha_N}}{D_{\sigma_1} \dots D_{\sigma_m}} \tag{2}$$

where now the $\bar{z}_i$ are only the scalar products which cannot be eliminated by being written as linear combinations of $D_i$, know as irreducible scalar products (ISPs) or the transverse $k_i \cdot \eta_j$ and $\sigma$ is any subset of $\{1, \dots, N_p\}$ with m elements.

Write the numerator of the integrand level amplitude as follows:

## Numerator Formula

$$\mathcal{N} = \sum_{m=0}^{N_p} \sum_{\sigma} \sum_{a} \bar{c}_a \prod_{i}^{N_{T+ISP}} (\bar{z}_i)^{\alpha_i} \prod_j D_j \tag{3}$$

where $N_{T+ISP}$ is the number of ISP and transverse scalar products and $j \neq \sigma_i$ for all $i$ [Bevilacqua et al., 2024]

# 2-Loop Amplitude reduction

Our goal now is the calculate the coefficients $\bar{c}_a$ which generically depend on the set of scalar products. For each topology, we

- Identify the maximal set of loop propagators we can set to zero, i.e. the Maximal Cut and solve the equations that put all of them on shell simultaneously, AKA *cut equations*
- Write the equations of the coefficients $\mathbf{M} \cdot \vec{c} = \vec{\mathcal{N}}$ where $\mathbf{M}$ is a matrix of all monomials $\bar{z}_i$ evaluated on different values of cut-solutions, $\vec{c}$ is all the $c_a$ and $\vec{\mathcal{N}}$ is a vector of equal length with values of the numerator evaluated on the cut-solutions
- Solve the system of equations

# 2-Loop Amplitude reduction

- Move to the next cut, where one less propagator is put on shell, AKA a *subtopology*.
- Do this till this for all subtopologies, (up to a 3-cut for massless legs and up to a 2-cut for massive legs)
- The reduction is complete!

# 2-Loop Amplitude: Final Form

$$P = P_{maxcut} + \sum_i P_{maxcut-1} D_i + \sum_{ij} P_{maxcut-2} D_i D_j + \ldots \tag{4}$$

where the Ps are polynomials in the transverse elements and ISPs.

**Final Form**:

$$\mathcal{A} = \sum_i F_i \tag{5}$$

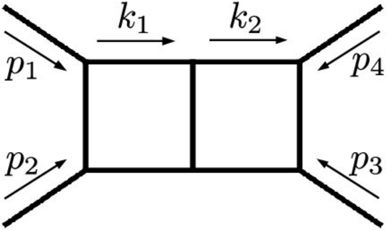
where:

$$F_i \equiv F_{a_1 \ldots a_N} = \int d^d k \frac{\overbrace{(D_{m+1})^{a_{m+1}} \cdots (D_N)^{a_N}}^{ISP}}{\underbrace{(D_1)^{a_1} \cdots (D_m)^{a_m}}_{RSP}}$$

Use Integration by Parts (IBP) to further simplify the integral basis.

# 2-Loop reduction example

Let's look at a specific 2 $\rightarrow$ 2 topology example, part of current work:




Maximal cut: 7 propagators on shell. Question arises: Can/should we fit in 4 dimensions or $D = 4 - 2\epsilon$ dimension?

# 2-Loop reduction example: D-dimensions

In D-dimensions, begin with 11 free parameters: 8 from the components of 2 4-momenta, and 3 $\mu_{11}, \mu_{22}, \mu_{12}$ the $\epsilon$ components of $k_1^2$, $k_2^2$ and $k_1 \cdot k_2$ respectively.

With 7 cut equations, we have a remainder of 4 free parameters.

The right hand side of equation (3) (the Ansatz polynomial) has a total of 70 monomials, i.e. 70 coefficients to be fitted.

Use the 4 free parameters to get 70 sets of solutions in order to solve the system.

Challenge:Get a set of solutions to the cut equations which give an **M** matrix of rank 70.

Success! We have completed a Mathematica simulation of this fit.

How to do this numerically?

-Hopeful progress, currently implementing this in our group

# 2-Loop reduction example: 4-dimensions

In 4-dimensions, we begin with 8 free parameters which we can use to construct solutions to the cut equations, so after imposing the on-shell condition only 1 parameter left to build solutions with.
Problem!Cut solution sets with 1 free parameter cannot generate a matrix of rank of rank 70!
(Partial?) Success In 4-dimensions, we can use Gram determinant relations to reduce the number of coefficient we need to fit [Badger et al., 2012].
We find 28 for the example of the $2 \to 2$ double-box.
Completed a Mathematica simulation for the maximal cut.

# 2-Loop reduction example: 4-dimensions

But [Frellesvig, 2014] mention issues with lower topology cuts:

- A) Some sub-topologies having divergences that require careful subtraction
- B) Other sub-topologies where $\mathrm{Rank}[\mathbf{M}] < \mathrm{Length}[\vec{c}]$, i.e not enough independent equations to fit the polynomial Ansatz.

Both currently under investigation with our approach

# 2-Loop reduction example: 4-dimensions

More specific issues in 4d:

- Some monomials from which we construct the matrix **M** have the same values for different solution sets.
- Computationaly (?) difficult to reduce the numerator using Gram determinant relations when it's written in terms of monomials $\longrightarrow$ write directly in terms of $k_i$ components.

# 2 loop reduction interesting questions

- What goes into to constructing the polynomial Ansatz? Is there some apriori way of determining it for each numerator?
- Especially for 4 d: Can we build the polynomial in some other way, without relying on monomials?
- Structure of the solution space for the cut solutions (Algebraic Geometry question)

# Amplitude Construction in $D = 4 - 2\epsilon$ dimensions?

Remember what we said about Amplitude Construction?
We are motivated to do it in $D = 4 - 2\epsilon$.
Work is on going, but we are optimistic!

# Amplitude Construction in $D = 4 - 2\epsilon$ dimensions

Clarification: Going from 4 to $D$ dimensions is simple to do analytically for example in Mathematica.

| Structure in $d = 4$ | | Extra term |
|---|---|---|
| $q^2\, X$ | $\longrightarrow$ | $\mu\, X$ |
| $\sum_{\lambda} (\varepsilon_{\ell_1,\lambda} \cdot \varepsilon_{\ell_2,\lambda})\, X$ | $\longrightarrow$ | $(d-4)\, X$ |
| $\sum_{\lambda} (q \cdot \varepsilon_{\ell_1,\lambda})(q \cdot \varepsilon_{\ell_2,\lambda})\, X$ | $\longrightarrow$ | $(q^2 + \mu)\, X$ |

How can we do this numerically?

# Amplitude Construction in $D = 4 - 2\epsilon$ dimensions

Keep track of different coefficients of the cut loop current in the recursion. Example for a cut gluon:

$$J_l^\alpha \equiv C_q q^\alpha + C_\varepsilon \varepsilon^\alpha + C^\alpha_{q\cdot\varepsilon}(q\cdot\varepsilon) + R^\alpha$$

Where in the first step of the recursion only $C_\varepsilon = 1$ is non-zero.
Keeping track of these coefficients allows us to calculate the D-dimensional numerator recursively, i.e. without the need for an analytic formula.

## Benefits:

- D-dimensional fit seems more promising, not plagued by the problems mentioned previously
- All parts of the calculation are in $D$ dimensions
  $\longrightarrow$ **No need to calculate $R_1$ or $R_2$ terms!**

# A few words on Integrals

Use the Simplified Differential equations method [Papadopoulos, 2014].

- Parameterise the integrals as a function of a single dimensionless parameter x.
  This is such that $x = 0 \longrightarrow$ massless get and $x = 1 \longrightarrow$ off-shell (massive) leg.
- Vanishing of total derivatives allows us to IBP relations
- Taking derivatives with respect to x and using IBP relations, write down a system of differential equations in the form

$$\partial_x \mathbf{G}(x, \{s_{ij}\}, \epsilon) = \mathbf{H}(x, \{s_{ij}\}, \epsilon)\mathbf{G}(x, \{s_{ij}\}, \epsilon)$$

- Find the boundary condition at $x \longrightarrow 0$ from already know integrals in closed form.
- The $x \longrightarrow 1$ will give us a defined DE system for the same family of integrals with a massive leg for free

# A few words on Integrals

Use the Simplified Differential equations method [Papadopoulos, 2014]. The goal of the method is to express all the results in terms of the iterated integrals known as Goncharov Polylogarithms [Goncharov, 2001]. This is not always staightforwad for all different integral families. Complications are also encoded in the functional form of the **H** in term of the x variable.

# Next Steps/Outlook

Immediate steps

- Determine if a 4-d fit is possible and determine for which cases and why it may not be
- Implement the fully D-dimensional fit at one loop

Outlook

- D-dimensional fit at 2-loops for all $2 \to 2$ topologies
- $2 \to 3$ topology reduction

# Thank you for listening!

The research project was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the 2nd Call for H.F.R.I. Research Projects to support Faculty Members and Researchers (Project Number: 2674).

Badger, S., Frellesvig, H., and Zhang, Y. (2012).
Hepta-cuts of two-loop scattering amplitudes.
*Journal of High Energy Physics*, 2012(4).

Bevilacqua, G., Canko, D., and Papadopoulos, C. (2024).
Two-loop amplitude reduction with helac.

Bevilacqua, G., Czakon, M., Garzelli, M. V., van Hameren, A., Kardos, A., Papadopoulos, C. G., Pittau, R., and Worek, M. (2013).
HELAC-NLO.
*Comput. Phys. Commun.*, 184:986–997.

Frellesvig, H. A. (2014).
*Generalized Unitarity Cuts and Integrand Reduction at Higher Loop Orders*.
PhD thesis, Copenhagen U.

Goncharov, A. B. (2001).
Multiple polylogarithms and mixed tate motives.

Kanaki, A. and Papadopoulos, C. G. (2000).

Helac: A package to compute electroweak helicity amplitudes.
*Computer Physics Communications*, 132(3):306–315.

Ossola, G., Papadopoulos, C. G., and Pittau, R. (2007).
Reducing full one-loop amplitudes to scalar integrals at the integrand level.
*Nucl. Phys. B*, 763:147–169.

Ossola, G., Papadopoulos, C. G., and Pittau, R. (2008a).
CutTools: A Program implementing the OPP reduction method to compute one-loop amplitudes.
*JHEP*, 03:042.

Ossola, G., Papadopoulos, C. G., and Pittau, R. (2008b).
On the rational terms of the one-loop amplitudes.
*Journal of High Energy Physics*, 2008(05):004–004.

Papadopoulos, C. G. (2014).
Simplified differential equations approach for master integrals.
*Journal of High Energy Physics*, 2014(7).