# D-modules and integral reduction with CALICO

**Gaia Fontana (University of Zürich)**

In collaboration with Giuseppe Bertolini & Tiziano Peraro


**Loops & Geometry —22/04/2026**

**JHEP 10 (2025) 018**

# Roadmap of the talk

- Introduction & notation
- Parametric annihilators
- How do we compute them?
- A similar technique for differential operators
- Applications
- Conclusions & Outlook

# Introduction

# Loop Integrals

- LEGO® blocks of perturbative QFT beyond tree level
- Key ingredients for theoretical studies and phenomenological predictions in particle physics and gravity
- Rich and interesting mathematical structures


**Thousands of loop integrals appear when studying perturbative predictions!**

- **Crucial**: Finding relations between them
- Loop integrals admit various integral representations with different tradeoffs and mathematical properties

# This work

- Study & elaborate on the method of **parametric annihilators** for finding integral identities,
  - Focus on **parametric representations** of loop integrals
  - Extend applications to **different** representations
  - Similar technique for finding **differential equations**
- Provide an **implementation** of annihilators & differential operators based on modern linear solvers relying on cutting edge **finite-fields** techniques
- Implementation in the public Mathematica package: **CALICO**

**C**omputing **A**nnihilators from **L**inear **I**dentities
**C**onstraining (differential) **O**perators

# Some useful definitions

A list of variables: $\mathbf{z} = (z_1, \ldots, z_n)$

A list of exponents: $\alpha = (\alpha_1, \ldots, \alpha_n)$

$$\left.\begin{array}{l}\text{A list of variables: } \mathbf{z} = (z_1, \ldots, z_n) \\ \text{A list of exponents: } \alpha = (\alpha_1, \ldots, \alpha_n)\end{array}\right\} \longrightarrow \quad \mathbf{z}^\alpha = \prod_j z_j^{\alpha_j} \qquad |\alpha| = \sum_j \alpha_j$$

Monomials — total degree

We are interested in **integral families** of the form

These include **many parametrizations** of loop integrals

$$I_\alpha = \int \mathrm{d}^n \mathbf{z}\, \varphi_\alpha(\mathbf{z})\, u(\mathbf{z})$$

**Rational factors raised to integer powers**

**Twist**

Multivalued

$$\begin{cases} u(\mathbf{z}) = \prod_j B_j(\mathbf{z})^{\gamma_j} \\ u(\mathbf{z}) = \exp F(\mathbf{z}) \prod_j B_j(\mathbf{z})^{\gamma_j} \end{cases}$$

**Assumption:**

$\varphi_\alpha(\mathbf{z})$ closed under monomial multiplication & differentiation

# What do we want to do?

- Finding and solving **linear relations** satisfied by integrals having the form of $I_\alpha$
- Express integrals within a family as a linear combination of a set of independent **master integrals (MIs)**

$$I_\alpha = \sum_{\beta \in \mathrm{MIs}} c_{\alpha\beta} I_\beta$$

Crucial ingredient are **Integration-By-Parts identities (IBP):**

$$\int \mathrm{d}^n \mathbf{z}\, \partial_j \Big( \varphi_\alpha(\mathbf{z})\, u(\mathbf{z}) \Big) = 0$$

(Regulated integrals vanish at the integration boundary)

$$\int \mathrm{d}^n\mathbf{z}\,\partial_j\Big(\varphi_\alpha(\mathbf{z})\,u(\mathbf{z})\Big) = 0$$

By expanding the LHS of the IBP relation we get :

$$\int \mathrm{d}^n\mathbf{z}\,\partial_j\Big(\varphi_\alpha(\mathbf{z})\,u(\mathbf{z})\Big) = \int \mathrm{d}^n\mathbf{z}\;\left(\partial_j\varphi_\alpha(\mathbf{z})\right)\;u(\mathbf{z}) + \int \mathrm{d}^n\mathbf{z}\,\varphi_\alpha(\mathbf{z})\;\left(\partial_j \log u(\mathbf{z})\right)\;u(\mathbf{z})$$


**We get:**

- Non-trivial identities among the integrals
- $\partial_j \log u(\mathbf{z})$ usually creates terms that cannot be absorbed in $\varphi_\alpha(\mathbf{z})$

**not a relation within the original integral family!**

# Parametric annihilators


For Annihilators of Feynman integrals: See Seva's talk!

# Integral identities via parametric annihilators

## Parametric annihilator of order $o$ of $u(\mathbf{z})$

[Baikov (1996); Lee (2014); Bitoun, Bogner, Klausen, Panzer (2017)]

$$\hat{A} = c_0(\mathbf{z}) + \sum_j c_j(\mathbf{z})\,\partial_j + \sum_{j_1 \le j_2} c_{j_1 j_2}(\mathbf{z})\,\partial_{j_1}\partial_{j_2}$$

$$+ \cdots + \sum_{j_1 \le \cdots \le j_o} c_{j_1 \cdots j_o}(\mathbf{z})\,\partial_{j_1} \cdots \partial_{j_o}$$

$c_{j_1 j_2 \ldots}(\mathbf{z})$ polynomials in $\mathbf{z}$

**Such that** $\hat{A}\,u(\mathbf{z}) = 0$

$$\hat{A}\, u(\mathbf{z}) = 0$$

For any annihilator $\hat{A}$, we have infinitely many integral identities

$$\int \mathrm{d}^n \mathbf{z}\, \varphi_\alpha(\mathbf{z})\, \hat{A}\, u(\mathbf{z}) = 0, \;\; \forall \alpha$$

**symbolic $\alpha$**

Using IBPs on derivatives, we get a **template identity** for symbolic $\alpha$

$$\int u\,(\varphi_\alpha c_0) - \sum_j \int u\,(\partial_j c_j \varphi_\alpha) + \cdots + (-1)^o \sum_{j_1 \le \cdots \le j_o} \int u\, \partial_{j_1} \cdots \partial_{j_o} (c_{j_1 \cdots j_o} \varphi_\alpha) = 0$$

**All the integrals belong to the family $I_\alpha$**

# Laporta algorithm

[Chetyrkin, Tkachov (1981); Laporta (2000)]

$$\int u\,(\varphi_\alpha c_0) - \sum_j \int u\,(\partial_j c_j \varphi_\alpha) + \cdots + (-1)^o \sum_{j_1 \le \cdots \le j_o} \int u\,\partial_{j_1} \cdots \partial_{j_o} (c_{j_1 \cdots j_o} \varphi_\alpha) = 0$$

**Seeding** the template eq.s : replacing symbolic $\alpha$ with integer numbers

- Applying each template identity to a large number of seed integrals: obtain a **linear system** of equations
- Choice of an **ordering**: express **complex** integrals as a function of **simple** ones
- solving it: reduction to **master integrals**★

$$I_\alpha = \sum_{\beta \in \mathrm{MIs}} c_{\alpha\beta} I_\beta$$

**Computational bottleneck in state-of-the-art calculations**

See Sid's talk!

★ **Additional relations may exist, such as symmetry relations**

# Properties of parametric annihilators

$\hat{A}$ is an annihilator $\Rightarrow$ $\mathbf{z}^\beta \hat{A}$ and $\partial_j \hat{A}$ are annihilators

**Set of annihilators of $u(\mathbf{z})$ is a D-module**

$$\int \mathrm{d}^n\mathbf{z}\, \varphi_\alpha(\mathbf{z}) \left(\mathbf{z}^\beta \hat{A}\, u(\mathbf{z})\right) = \int \mathrm{d}^n\mathbf{z} \left(\mathbf{z}^\beta \varphi_\alpha(\mathbf{z})\right) \hat{A}\, u(\mathbf{z})$$

$$\int \mathrm{d}^n\mathbf{z}\, \varphi_\alpha(\mathbf{z}) \left(\partial_j \hat{A}\, u(\mathbf{z})\right) = -\int \mathrm{d}^n\mathbf{z} \left(\partial_j \varphi_\alpha(\mathbf{z})\right) \hat{A}\, u(\mathbf{z})$$

**$\varphi_\alpha(\mathbf{z})$ closed under monomial multiplication & differentiation**

interested in a **minimal** set of **generators**:

set of annihilators $\hat{A}$ independent modulo linear combinations of $\mathbf{z}^\beta \hat{A}$ and $\partial_j \hat{A}$

# How to compute parametric annihilators

# Computing annihilators via linear constraints

- Computing parametric annihilators up to a certain order and polynomial degree
- implemented in the **CALICO** package

## Step 1 : From annihilators to syzygies

$$\mathbf{f}(\mathbf{z}) \cdot \mathbf{g}(\mathbf{z}) = 0$$ Syzygy equation

known polynomials

$$\mathbf{f}(\mathbf{z}) = \{f_1(\mathbf{z}), \ldots, f_n(\mathbf{z})\}$$

unknown polynomials

$$\mathbf{g}(\mathbf{z}) = \{g_1(\mathbf{z}), \ldots, g_n(\mathbf{z})\}$$

**Syzygy sol.s form a module:**

$\mathbf{g}^{(k)}(\mathbf{z})$ set of solutions $\Rightarrow \mathbf{g}(\mathbf{z}) = \sum_k p_j(\mathbf{z})\mathbf{g}^{(k)}(\mathbf{z})$ is also a solution

$p_j(\mathbf{z})$ arbitrary polynomials

# Step 1 : From annihilators to syzygies

Start from $\hat{A}\,u(\mathbf{z}) = 0 \;\Rightarrow$ Identify $c_{j_1,\ldots,j_k}(\mathbf{z})$ as the unknown polynomials $\mathbf{g}(\mathbf{z})$ of a syzygy $\mathbf{f}(\mathbf{z}) \cdot \mathbf{g}(\mathbf{z}) = 0$

$$\frac{1}{u(\mathbf{z})}\,\hat{A}\,u(\mathbf{z}) = 0$$

- Insert the expression for $\hat{A} = c_0(\mathbf{z}) + \sum c_j(\mathbf{z})\partial_j + \ldots$
- Collect LHS under a common denominator
- Imposing vanishing of the numerator ⇒ obtain a syzygy equation for $c_{j_1,\ldots,j_k}(\mathbf{z})$

- Example : **first-order annihilator**

$$\hat{A} = c_0(\mathbf{z}) + \sum_j c_j(\mathbf{z})\,\partial_j \qquad\qquad u(\mathbf{z}) = \prod_j B_j(\mathbf{z})^{\gamma_j}$$

$$c_0(\mathbf{z}) \prod_k B_k(\mathbf{z}) + \sum_{j=1}^{n} c_j(\mathbf{z}) \sum_k \gamma_k \left(\partial_j B_k(\mathbf{z})\right) \prod_{l\neq k} B_l(\mathbf{z}) = 0$$

**Form of a syzygy equation**

$$\mathbf{f}(\mathbf{z}) \cdot \mathbf{g}(\mathbf{z}) = 0$$

# Step 2: Solving syzygy equations via linear constraints

[Schabinger (2015)]

- Ansatz for the solution
  - ★ Also with user-defined filters!

$$\mathbf{g}(\mathbf{z}) = \sum_j \sum_\alpha c_{j\alpha}\, \mathbf{z}^\alpha\, \hat{e}_j$$

$|\alpha| \leq d$ for some max degree $d$

$\hat{e}_j$ unit vector in the $j$-th direction

- Plug it into $\mathbf{f}(\mathbf{z}) \cdot \mathbf{g}(\mathbf{z}) = 0$
- Impose coeff.s of each monomial $\mathbf{z}$ vanish

$\Rightarrow$ **Linear system for $c_{j\alpha}$**

CALICO uses the efficient linear solver of FiniteFlow, based on finite-field methods

[Peraro (2019)]

# Differential operators

# Differential equations

[Barucchi, Ponzano (1973); Kotikov (1991); Bern, Dixon, Kosower (1994); Gehrmann, Remiddi (2000)]

- Integrals in the form of $I_\alpha$ also depend on additional **free parameters** $x$ (e.g. kinematic invariants)
- Studying of **analytic structure** & their numerical or analytical **evaluation**
- Reducing the derivative of MIs with respect to $x$ to MIs, write a **system of differential equations** satisfied by the MIs themselves

$$\partial_x I_\alpha = \sum_{\beta \in \mathrm{MIs}} M_{\alpha\beta} I_\beta, \qquad \text{for } \alpha \in \mathrm{MIs}.$$

$x$ **free parameter of the integrals**

# Differential equations via differential operators

- Derive an operator $\hat{O}_x$ that realizes differentiation with respect to $x$

$$\hat{O}_x = c_0^{(x)}(\mathbf{z}) + \sum_j c_j^{(x)}(\mathbf{z})\,\partial_j + \sum_{j_1 \le j_2} c_{j_1 j_2}^{(x)}(\mathbf{z})\,\partial_{j_1}\partial_{j_2}$$

$$+ \cdots + \sum_{j_1 \le \cdots \le j_o} c_{j_1 \cdots j_o}^{(x)}(\mathbf{z})\,\partial_{j_1} \cdots \partial_{j_o}$$

$c_{j_1 j_2 \cdots}^{(x)}$ polynomials

**Such that** $\hat{O}_x\, u(\mathbf{z}) = \partial_x\, u(\mathbf{z})$

# Differential equations via differential operators

- Explicitly, integrating by parts all derivatives in $\hat{O}_x$

$$\partial_x I_\alpha = \int (\hat{O}_x u)\, \varphi_\alpha + \int u\, (\partial_x \varphi_\alpha)$$

$$= \int u\, (\varphi_\alpha c_0^{(x)}) - \sum_j \int u\, (\partial_j c_j^{(x)} \varphi_\alpha) + \cdots$$

$$+ (-1)^o \sum_{j_1 \leq \cdots \leq j_o} \int u\, \partial_{j_1} \cdots \partial_{j_o} (c_{j_1 \cdots j_o}^{(x)} \varphi_\alpha) + \int u\, (\partial_x \varphi_\alpha)$$

- Our preferred way to compute derivatives
  - Alternatively: recurrence relations

# Applications

# Hypergeometric ${}_2F_1$

$$I_\alpha = \int_0^1 \mathrm{d}z\, \varphi_\alpha(z)\, u(z)$$

$$\varphi_\alpha(z) = z^\alpha, \qquad u(z) = z^{b_2-1}\,(1-z)^{b_3-b_2-1}\,(1-x\,z)^{-b_1}$$

Related to the hypergeometric ${}_2F_1$

$$I_\alpha = \frac{\Gamma(b_2+\alpha)\Gamma(b_3-b_2)}{\Gamma(b_3+\alpha)}\,{}_2F_1(b_1,\, b_2+\alpha,\, b_3+\alpha;\, x)$$

## 1. Reduction to MIs

$$\hat{A} = c_0(z) + c_1(z)\,\partial_z$$

**CALICO finds a first-order annihilator**

$$\begin{cases} c_0(z) = 1 - b_2 + (b_3 - 2 + (b_2 - b_1 - 1)\,x)\,z + (2 + b_1 - b_3)\,x\,z^2 \\ c_1(z) = z - (1+x)\,z^2 + x\,z^3 \end{cases}$$

## Template equation

$$-(\alpha+b_2)I_\alpha+(\alpha+b_3+(1+\alpha-b_1+b_2)x)I_{\alpha+1}+(b_1-b_3-1-\alpha)\,x\,I_{\alpha+2}=0$$

$$\alpha=0 \longrightarrow I_2=\frac{b_2}{(b_1-b_3-1)\,x}I_0+\frac{(b_1-b_2-1)\,x-b_3}{(b_1-b_3-1)\,x}I_1$$

two MIs, which we can choose as $I_0$ and $I_1$

## 2. Differential equations

$$\hat{O}_x=c_0^{(x)}(z)+c_1^{(x)}(z)\,\partial_z$$

First order differential operator

$$\begin{cases} c_0^{(x)}(z)=\dfrac{(b_2-1)+(2+b_1-b_3)\,z}{1-x} \\ c_1^{(x)}(z)=\dfrac{z^2-z}{1-x} \end{cases}$$

$$\hat{O}_x\,u(\mathbf{z})=\partial_x\,u(\mathbf{z})$$

# Template differential equation

$$\partial_x I_\alpha = \frac{\alpha + b_2}{1-x} I_\alpha + \frac{b_1 - b_3 - \alpha}{1-x} I_{\alpha+1}$$

Applied to $\alpha = 0,1$ and using the reduction of $I_2$ to the MIs $\{I_0, I_1\}$

$$\partial_x \begin{pmatrix} I_1 \\ I_0 \end{pmatrix} = \begin{pmatrix} \frac{b_1\, x - b_3}{(1-x)\, x} & \frac{b_2}{(1-x)\, x} \\ \frac{b_1 - b_3}{1-x} & \frac{b_2}{1-x} \end{pmatrix} \cdot \begin{pmatrix} I_1 \\ I_0 \end{pmatrix}$$

$$I_2 = \frac{b_2}{(b_1 - b_3 - 1)\, x} I_0 + \frac{(b_1 - b_2 - 1)\, x - b_3}{(b_1 - b_3 - 1)\, x} I_1$$

# Hypergeometric ${}_{n+1}F_n$

$$\varphi_\alpha(\mathbf{z}) = \mathbf{z}^\alpha,$$

$$u(\mathbf{z}) = \left(1 - x\prod_{j=1}^{n} z_j\right)^{-a_{n+1}} \prod_{j=1}^{n}\left[z_j^{a_j-1}(1-z_j)^{b_j-a_j-1}\right]$$

$$I_\alpha = \prod_{j=1}^{n}\left[\frac{\Gamma(a_j+\alpha_j)\Gamma(b_j-a_j)}{\Gamma(b_j+\alpha_j)}\right]$$

$$\times\ {}_{n+1}F_n(a_1+\alpha_1,\ldots,a_n+\alpha_n,a_{n+1};b_1+\alpha_1,\ldots,b_n+\alpha_n;x)$$

- $n+1$ independent master integrals
- differential equations for the MIs

**Checked against numerical evaluations of the hypergeometric functions**

# Loop integrals

**Momentum-space representation**

$$J_\alpha = J_{\alpha_1 \cdots \alpha_n} = \int \prod_{i=1}^{\ell} \frac{\mathrm{d}^d k_i}{i\pi^{d/2}} \frac{1}{D_1^{\alpha_1} \dots D_n^{\alpha_n}}$$

$D_j s$ are generalised denominators

- **Proper denominators:** $D_j$ such that $\alpha_j > 0$
- Irreducible scalar products (ISPs): $D_j$ such that $\alpha_j \leq 0$

$$D_{F,j} = l_j \cdot v_j - m_j^2$$

$$D_{F,j} = l_j^2 - m_j^2$$

$l_j$ linear combination of $k_j$,
$v_j$ linear combination of $p_j$

**IBPs in momentum space**

[Tkachov (1981), Chetyrkin, Tkachov (1981)]

$$\int \prod_{i=1}^{\ell} \frac{\mathrm{d}^d k_i}{i\pi^{d/2}} \frac{\partial}{\partial k_j^\mu} \frac{v^\mu}{D_1^{\alpha_1} \dots D_n^{\alpha_n}} = 0, \quad \text{with } v^\mu = k_i^\mu, p_i^\mu$$

# Parametric representations of loop integrals

**Baikov**

$$I_\alpha = \int \mathrm{d}^n\mathbf{z}\, \frac{1}{\mathbf{z}^\alpha}\, B(\mathbf{z})^\gamma$$

Also **Loop-by-Loop Baikov** & **Duals of loop integrals**

**Lee-Pomeransky**

$$I_\alpha = \int \mathrm{d}^n\mathbf{z} \left( \prod_{j=1}^{n} \frac{1}{\Gamma(\alpha_j)} \right) \mathbf{z}^{\alpha-1} G(\mathbf{z})^{-d/2}$$

$$G(\mathbf{z}) = \mathcal{U}(\mathbf{z}) + \mathcal{F}(\mathbf{z})$$

**Schwinger**

$$I_\alpha = \int \mathrm{d}^n\mathbf{z} \left( \prod_{j=1}^{n} \frac{1}{\Gamma(\alpha_j)} \right) \mathbf{z}^{\alpha-1} \exp\left[ -\mathcal{F}(\mathbf{z})/\mathcal{U}(\mathbf{z}) \right] \mathcal{U}(\mathbf{z})^{-d/2}$$

# Laporta algorithm: toy example

**Momentum representation** of 1L bubble integral

$$I_{\alpha_1\alpha_2} = \int \frac{\mathrm{d}^D k}{(2\pi)^D} \frac{1}{D_1^{\alpha_1} D_2^{\alpha_2}}$$

$$s = p^2 \quad \begin{cases} D_1 = k^2 - m^2 \\ D_2 = (k+p)^2 - m^2 \end{cases}$$

**IBPs**

$$\begin{cases} \displaystyle\int \frac{\partial}{\partial k_\mu} \frac{k^\mu}{D_1^{\alpha_1} D_2^{\alpha_2}} = 0 \\ \displaystyle\int \frac{\partial}{\partial p_\mu} \frac{k^\mu}{D_1^{\alpha_1} D_2^{\alpha_2}} = 0 \end{cases}$$

**Template equations**

$$\begin{cases} -\alpha_2\, I[\alpha_1 - 1, \alpha_2 + 1] + (D - \alpha_2 - 2\alpha_1)\, I[\alpha_1, \alpha_2] + \alpha_2 (2m^2 - s)\, I[\alpha_1, \alpha_2 + 1] - 2\alpha_1 m^2\, I[\alpha_1 + 1, \alpha_2] = 0 \\ \alpha_2\, I[\alpha_1 - 1, \alpha_2 + 1] + (\alpha_1 - \alpha_2)\, I[\alpha_1, \alpha_2] + \alpha_2 s\, I[\alpha_1, \alpha_2 + 1] - \alpha_1\, I[\alpha_1 + 1, \alpha_2 - 1] + \alpha_1 s\, I[\alpha_1 + 1, \alpha_2] = 0 \end{cases}$$

- Seeding ($\alpha \to$ integer values)
- Solution of the linear system

# Laporta algorithm: toy example

**Schwinger representation** of 1L bubble integral

$$I[\alpha_1,\alpha_2] = \int \mathrm{d}^2\mathbf{z}\;\; u(\mathbf{z})\,\varphi(\alpha_1,\alpha_2,\mathbf{z})$$

$$\begin{cases} u(\mathbf{z}) = \exp^{-\frac{F(\mathbf{z})}{U(\mathbf{z})}}\; U(\mathbf{z})^{-D/2} \\ \varphi(\alpha_1,\alpha_2,\mathbf{z}) = \dfrac{1}{z_1^{\alpha_1} z_1^{\alpha_2}} \end{cases}$$

$$\begin{cases} F(\mathbf{z}) = m^2 z_1^2 + 2m^2 z_1 z_2 - s z_1 z_2 + m^2 z_2^2 \\ U(\mathbf{z}) = z_1 + z_2 \end{cases}$$

First and second Symanzik polys

**Annihilators:**

$$\hat{A}_1 = (D - s z_2 + 2m^2(z_1+z_2)) + z_2\partial_{z_2} + (2z_1+z_2)\partial_{z_1}$$
$$\hat{A}_2 = (D + s(-2z_1+z_2) + 2m^2(z_1+z_2)) + (2z_1+3z_2)\partial_{z_2} - z_2\partial_{z_1}$$

## Template equations

$$\begin{cases} -2\alpha_1 m^2\, I[\alpha_1+1,\alpha_2] + (D-2\alpha_1-\alpha_2)\, I[\alpha_1,\alpha_2] - \alpha_2 I[\alpha_1-1,\alpha_2+1] - (2m^2-s)\alpha_2\, I[\alpha_1,\alpha_2+1] = 0 \\ -2\alpha_1\, I[\alpha_1+1,\alpha_2-1] - 2(m^2-s)\alpha_1\, I[\alpha_1+1,\alpha_2] - (D-3\alpha_2)\, I[\alpha_1,\alpha_2] + \alpha_2\, I[\alpha_1-1,\alpha_2+1] - (2m^2+s)\alpha_2\, I[\alpha_1,\alpha_2+1] = 0 \end{cases}$$

- Seeding ($\alpha \to$ integer values)
- Solution of the linear system

# Baikov representation

$$I_\alpha = \int \mathrm{d}^n\mathbf{z}\, \varphi_\alpha(\mathbf{z})\, u(\mathbf{z})$$

Reminder

**Baikov**

$$I_\alpha = \int \mathrm{d}^n\mathbf{z}\, \frac{1}{\mathbf{z}^\alpha}\, B(\mathbf{z})^\gamma$$

**Loop-By-Loop Baikov**

[Frellesvig, Papadopoulos (2017)]

$$I_\alpha = \int \mathrm{d}^n\mathbf{z}\, \frac{1}{\mathbf{z}^\alpha} \prod_{j=1}^{2\ell-1} B_j(\mathbf{z})^{\gamma_j}$$

Change of integration vars from loop momenta to generalised denominators

- Require a full set of ISPs
- $\deg B = \min(\ell + e, 2\ell)$
- Fewer variables in LBL (Baikov change of var.s is done one loop at a time)

# Intersection theory in a nutshell

[ Mastrolia, Mizera (2019) ] [Brunello, Cacciatori, Caron-Huot, Chestnov, Crisanti, Duhr, Frellesvig, GF, Gasparotto, Giroux, Laporta, Huang, Ma, Maggio, Mandal, Mastrolia, Matsubara-Heo, Mattiazzi, Mizera, Munch, Peraro, Pokraka, Porkert, Semper, Smith, Sohnle, Stavinski, Takayama, Wang, Yang (2019 – Ongoing)]

Introduction of a scalar product, **intersection number**, between

- **Integrals** $I_\alpha$

$$I_\alpha = \int \mathrm{d}^n \mathbf{z}\, \frac{1}{\mathbf{z}^\alpha}\, B(\mathbf{z})^\gamma$$

- **Dual integrals** $I^\star_\alpha$

$$I^\star_\alpha = \int \mathrm{d}^n \mathbf{z}\, \frac{1}{\mathbf{z}^\alpha}\, B(\mathbf{z})^{-\gamma}$$

- Calculation of intersection numbers $\longrightarrow$ standard procedure: **recursive** algorithm in the integration var.s

- Focus on **dual integrals**
  - Linear relations
  - Differential equations

(necessary step in the intersection numbers calculation)

# Dual integrals & regulators

$$I^\star_\alpha = \int \mathrm{d}^n\mathbf{z}\, \frac{1}{\mathbf{z}^\alpha}\, B(\mathbf{z})^{-\gamma}$$

- $(z_1, \ldots, z_m)$ proper denominators
- $(z_{m+1}, \ldots, z_n)$ ISPs

$\Big\}$ $\alpha_j \leq 0$ for $j > m$

Problem: need of additional regulators $\dfrac{1}{z_j^{\alpha_j}} \to \dfrac{1}{z_j^{\alpha_j - \rho_j}}$ for $j \leq m$

[GF, Peraro (2023)]

$$I^\star_\alpha = \underbrace{\prod_{j=1}^{m} \rho_j^{\Theta(\alpha_j - 1/2)} \int \mathrm{d}^n\mathbf{z}\, \frac{1}{\mathbf{z}^{\alpha-\rho}}}_{\varphi_\alpha(\mathbf{z})}\, \underbrace{B(\mathbf{z})^{-\gamma}}_{u(\mathbf{z})}$$

**work on the leading coefficients** $\rho_j \to 0$

# Reduction of dual integrals

- Method of annihilator can be applied straightforwardly (Baikov, LBL Baikov)
- Tmp ids require the knowledge of $\varphi_\alpha(\mathbf{z})$ under

- Multiplication by monomials

$$z_j^{\beta_j}\,\varphi_\alpha(\mathbf{z}) = \begin{cases} \varphi_{\alpha-\beta_j\hat{e}_j}(\mathbf{z}) & \text{if } \alpha_j > \beta_j \text{ or } \alpha_j \leq 0 \\ 0 & \text{if } 0 < \alpha_j \leq \beta_j. \end{cases}$$

- Differentiation

$$\partial_j\,\varphi_\alpha(\mathbf{z}) = \begin{cases} -(\alpha_j - \delta_{\alpha_j 0})\,\varphi_{\alpha+\hat{e}_j}(\mathbf{z}) & \text{if } j \leq m \\ -\alpha_j\,\varphi_{\alpha+\hat{e}_j}(\mathbf{z}) & \text{if } j > m. \end{cases}$$

- Approach tested for both
  - Reduction of dual integrals
  - Computation of DEQ satisfied by dual integrals in fewer variables
- Successful comparison on all examples of [GF, Peraro (2023)]

# Lee-Pomeransky and Schwinger rep.s

## Lee-Pomeransky

$$I_\alpha = \int \mathrm{d}^n\mathbf{z} \left( \prod_{j=1}^{n} \frac{1}{\Gamma(\alpha_j)} \right) \mathbf{z}^{\alpha-1} G(\mathbf{z})^{-d/2} \qquad G(\mathbf{z}) = \mathcal{U}(\mathbf{z}) + \mathcal{F}(\mathbf{z})$$

## Schwinger

$$I_\alpha = \int \mathrm{d}^n\mathbf{z} \left( \prod_{j=1}^{n} \frac{1}{\Gamma(\alpha_j)} \right) \mathbf{z}^{\alpha-1} \exp\left[ -\mathcal{F}(\mathbf{z})/\mathcal{U}(\mathbf{z}) \right] \mathcal{U}(\mathbf{z})^{-d/2}$$

- $n$ integration variables (# number of proper denominators)
- no need of introducing ISPs (can be added if required)
- $\deg G = \ell + 1$ at $\ell$ loops

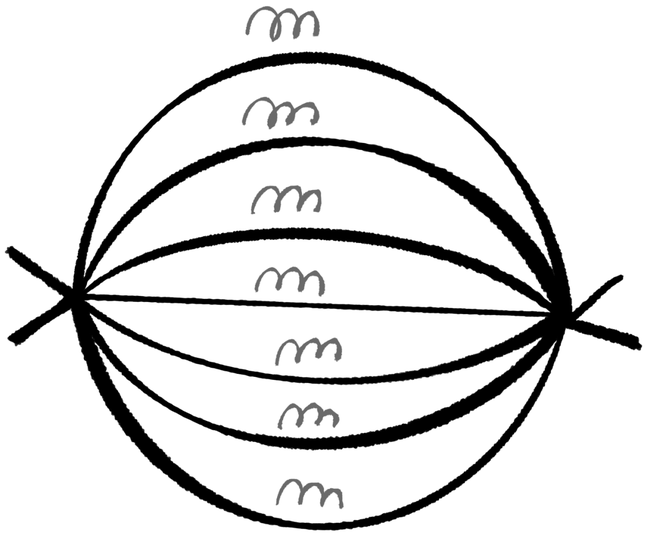
- twist includes exp. factor
- empirically:
  - annihilators have lower degree
  - More frequent use of 2nd order ann.
  - tmp id.s often fewer and simpler

# $L$ loop bananas

$\ell$-loop & one internal mass $m$, defined by the set of $\ell + 1$ proper denominators:

$$D_j = k_j^2 - m^2 \quad \text{for } j = 1,\ldots,\ell$$

$$D_{\ell+1} = (k_1 + \cdots + k_\ell - p)^2 - m^2$$




Momentum space has $(\ell + 2)(\ell - 1)/2$ ISPs $\rightarrow$ for $\ell = 6$, it has $20$ ISPs

Use Lee-Pomeranski or Schwinger representation to

- Reduce to MIs
- Derive DEQs

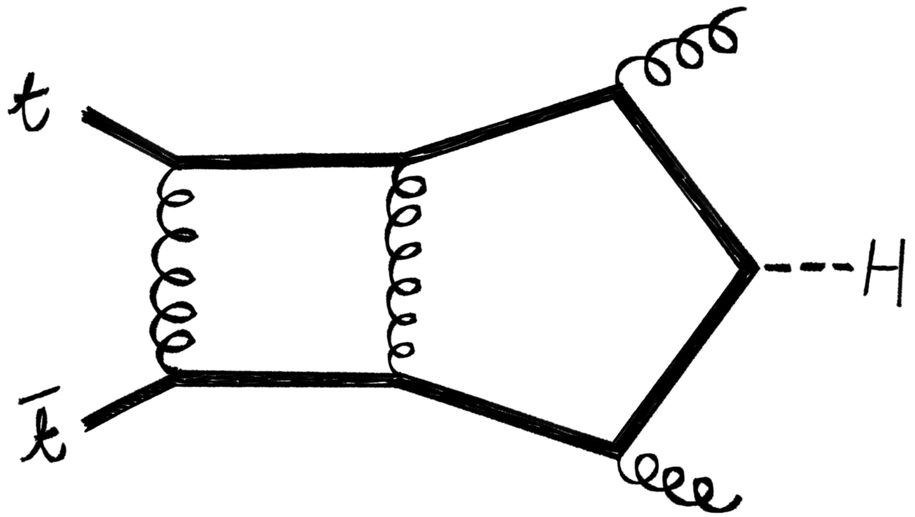
Without the need of additional ISPs!

**Done in a couple of minutes on a laptop up to 6 loops (mostly spent computing annihilators and template eqs)**

# Family for $t\bar{t}H$ production

Cutting-edge example

- Many different scales
- Many external legs




Using the integral representations:

- Schwinger (More efficient!)
- Lee-Pomeranski

**Tested: numerical reduction on a laptop with up to 3 extra powers of denominators**

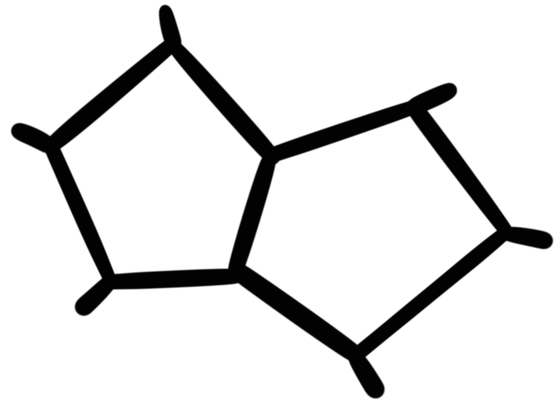
**Finding relations between integrals with constant numerator and higher powers of proper denominators**

Useful for finding integrals with good properties

- Quasi-finite [von Manteuffel, Panzer, Schabinger (2014)]
- Pure functional form [Henn (2013)]


# New applications

# $6$ points $2$ loops planar integrals



Work by X. Liu, A. Matijašić, T. Peraro, Y. Xu, Z. Yang, Y. Zhang
[https://arxiv.org/pdf/2603.16831]

Finding the functions' alphabet:

- For some letter, necessary to reconstruct some matrix elements of the DE
- For one element the reconstruction is **unfeasible** with standard IBP tools

## Setup

- Baikov in $4D$ kinematics with momentum twistors ($11$ generalised den.s)
- octa-cut ($z_j \to 0$) to isolate the desired matrix element
- **CALICO** to generate differential operators on the cut parametrization
- Find identities directly on the cut
- Reconstruct identities on the cut → reconstruct the DE

**~min.s on a laptop!**

# $n_h^2$ contribution to $3L$ virtual corrections to $HH$ production

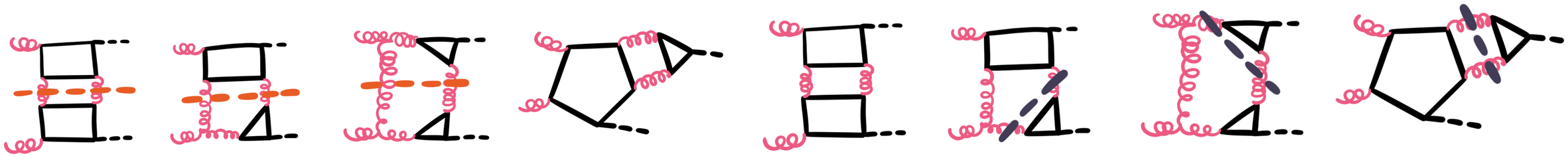
In collaboration with J. Davies, K. Schönwald, M. Steinhauser, M. Vitti

Calculation setup:

- Forward expansion: $m_H^2 \to \lambda m_H^2, t \to \lambda t, \lambda \to 0$

[See M. Vitti talk at Loops&Legs2026]

- Expansions by regions: hard-hard-{hard; soft / ultrasoft; collinear}



$\sim \log(t)$ $\sim \log(m_H^2)$

- Hard region: Taylor expansion
- Collinear region: **nontrivial!** Appearance of non-standard propagators in momentum representation

$$\left[(k_1-k_3)^2-m_t^2\right] \longrightarrow \left[k_1^2-2k_1\cdot q_1\left(\frac{2k_3\cdot q_2}{s}\right)-m_t^2+\mathcal{O}(\lambda)\right]$$

→ use Schwinger parametrisation!

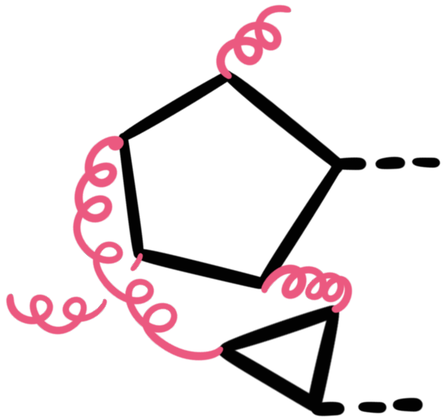
# Calculation of collinear integrals

- Complex direct analytic integration
- Many integrals **(~2500)** for a given topology


**CALICO offers a systematic approach!**

- Write collinear integrals in Schwinger parametrization
- **CALICO:** IBPs in Schwinger representation
  - Reduce to collinear master integrals → small number of less complex integrals
- **CALICO:** DEs via differential operators
  - Analytic solution of the master integrals

e.g.:



- **64k** collinear integrals, up to 24 dots
  ↪**7 MIs**

# Conclusions & Outlook

- Parametric annihilators are a useful tool for finding linear relations between integrals
- Allow to use integral parametrizations tailored to specific problems
  - New applications to LBL Baikov, duals of loop integrals, Schwinger parametrisation
- Introduced similar technique for deriving differential equations
- Implementation released in the public package **CALICO**
- Applications to state-of-the-art calculations

★ **Bonus:** can also solve syzygy equations and polynomial decomposition problems

https://github.com/fontana-g/calico

# Thank you for your attention!