Transforming multi-loop Feynman integrals to a canonical basis

based on [C. M. '16]

CHRISTOPH MEYER (HU BERLIN)

PHENOMENOLOGY OF ELEMENTARY PARTICLE PHYSICS GROUP

PARTICLE THEORY SEMINAR ZURICH

DECEMBER 20, 2016

Motivation: Demand for multi-loop calculations

- Experimental precision @ LHC pushing below 5%

e.g. $t\bar{t}$ @ 13TeV

Measurement:

$$\sigma_{t\bar{t}} = 818.0 \pm 8.0 \pm 35.0\,\mathrm{pb}$$ [ATLAS Collaboration '16]

Theory NNLO + NNLL:

$$\sigma_{t\bar{t}} = 832.0^{+40}_{-46}\,\mathrm{pb}$$ [M. Czakon, A. Mitov '13]

⇨ Theory and experiment at $\mathcal{O}(5\%)$ precision

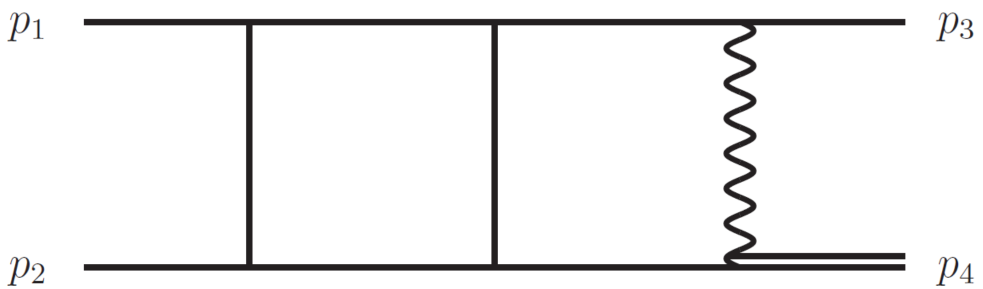
- Increasing integrated luminosity ⇨ decrease statistical uncertainties

⇨ Demand for precise theoretical predictions will grow

Introduction

Goal: Compute scalar multi-loop Feynman integrals

- **Example:** Single top-quark production @ NNLO QCD



Kinematics:

$$p_1^2 = 0, \quad p_2^2 = 0, \quad p_3^2 = 0, \quad p_4^2 = m_t^2$$

$$s = (p_1 + p_2)^2, \quad t = (p_2 - p_3)^2$$

- Consider whole family of integrals:

$$\int \frac{\mathrm{d}^d l_1}{\mathrm{i}\pi^{d/2}} \frac{\mathrm{d}^d l_2}{\mathrm{i}\pi^{d/2}} \frac{[(l_1 - p_2)^2]^{-a_8}[(l_2 + p_3 + p_1)^2]^{-a_9}}{[l_2^2]^{a_1}[l_1^2 - m_w^2]^{a_2}[(l_1 + p_3)^2]^{a_3}[(l_2 + p_2)^2]^{a_4}[(l_1 - p_4)^2]^{a_5}[(l_2 - p_1)^2]^{a_6}[(l_1 + l_2 - p_1 + p_3)^2]^{a_7}}$$

with integer powers $a_i \in \mathbb{Z}$

Introduction: Integration by parts relations

- Infinite number of scalar integrals in one family

$$I[a_1, \ldots, a_n] = \int \frac{\mathrm{d}^d l_1}{\mathrm{i}\pi^{d/2}} \cdots \frac{\mathrm{d}^d l_L}{\mathrm{i}\pi^{d/2}} \frac{P_{t+1}^{-a_{t+1}} \cdots P_n^{-a_n}}{P_1^{a_1} \cdots P_t^{a_t}}$$

Only a finite number is independent!

- Related by: Infinite number of Integration by parts relations: [K. G. Chetyrkin, F. V. Tkachov ’81]

$$\int \frac{\mathrm{d}^d l_1}{\mathrm{i}\pi^{d/2}} \cdots \frac{\mathrm{d}^d l_L}{\mathrm{i}\pi^{d/2}} \frac{\partial}{\partial l_j^\mu} v^\mu \frac{P_{t+1}^{-a_{t+1}} \cdots P_n^{-a_n}}{P_1^{a_1} \cdots P_t^{a_t}} = 0$$

any loop momentum

any external or loop momentum

- Relations can be applied systematically (Laporta algorithm) [S. Laporta ’01]

⇨ finite basis of independent master integrals

Introduction: Method of differential equations

[A. V. Kotikov '91; E. Remiddi '97; T. Gehrmann, E. Remiddi '00]

- Notation: vector of master integrals

$$\vec{f}(\epsilon, \{x_i\}) = \begin{pmatrix} f_1(\epsilon, \{x_i\}) \\ \vdots \\ f_m(\epsilon, \{x_i\}) \end{pmatrix}$$

dimensional regulator $\epsilon = (d-4)/2$

dimensionless invariants

- Derivative of Feynman integral ➡ Linear combination of Feynman integrals

Idea: Express derivative of $\vec{f}$ in terms of $\vec{f}$ again:

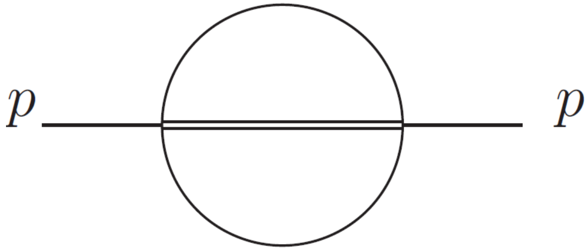
$$\mathrm{d}\vec{f}(\epsilon, \{x_i\}) = \sum_{j=1}^{M} \frac{\partial \vec{f}}{\partial x_j} \mathrm{d}x_j = a(\epsilon, \{x_i\}) \vec{f}(\epsilon, \{x_i\})$$

w.r.t. the invariants

$$a(\epsilon, \{x_i\}) = \sum_{j=1}^{M} a_j(\epsilon, \{x_i\}) \mathrm{d}x_j$$

- $a_j(\epsilon, \{x_i\})$ are $m \times m$ matrices of rational functions

- Two loop sunrise:



$$p^2 = s$$

$$I[a_1,a_2,a_3,a_4,a_5] = \int \frac{\mathrm{d}^d l_1}{\mathrm{i}\pi^{d/2}} \frac{\mathrm{d}^d l_2}{\mathrm{i}\pi^{d/2}} \frac{[(l_1-p)^2]^{-a_4}(l_2-p)^2]^{-a_5}}{[l_1^2]^{a_1}[l_2^2]^{a_2}[(l_1+l_2-p)^2-m^2]^{a_3}}$$

- Family contains two master integrals: e.g. $\{I[1,1,1,0,0],\ I[1,1,2,0,0]\}$

- Rescale for dimensionless master integrals:

$$f_1 = (m^2)^{2\epsilon-1} I[1,1,1,0,0] \qquad f_2 = (m^2)^{2\epsilon} I[1,1,2,0,0]$$

- Can only depend on dimensionless variable: $x = \dfrac{p^2}{m^2}$

Introduction: Example for differential equations

- Take derivative of master integral:

$$\frac{\partial I[1,1,1,0,0]}{\partial s} = \frac{-1}{2s}\left(I[1,1,2,0,-1]+I[1,1,2,-1,0]\right)$$
$$= \frac{1}{s}\left((1-2\epsilon)I[1,1,1,0,0]-m^2 I[1,1,2,0,0]\right)$$

IPBs

$$\Rightarrow \quad \frac{\partial f_1(\epsilon,x)}{\partial x} = \frac{1-2\epsilon}{x} f_1(\epsilon,x) - \frac{1}{x} f_2(\epsilon,x)$$

$$\begin{pmatrix} \partial_x f_1(\epsilon,x) \\ \partial_x f_2(\epsilon,x) \end{pmatrix} = \begin{pmatrix} \frac{1-2\epsilon}{x} & -\frac{1}{x} \\ \frac{(-1+2\epsilon)(-2+3\epsilon)}{(-1+x)x} & \frac{-2+3\epsilon+\epsilon x}{(-1+x)x} \end{pmatrix} \begin{pmatrix} f_1(\epsilon,x) \\ f_2(\epsilon,x) \end{pmatrix}$$

- Differential equation determines the master integrals up to integration constants

- Solve differential equation for $\vec{f}(\epsilon,\{x_i\})$ $\Rightarrow$ in general very hard!

[J.M. Henn '13]

- Turn to a different basis: $\vec{f} = T(\epsilon, \{x_i\})\vec{f'}$

$$\mathrm{d}\vec{f'} = a'\vec{f'}$$

transformation law: $a' = T^{-1}aT - T^{-1}\mathrm{d}T$

- Idea: Use a basis such that the differential equation is in ϵ - *form*:

$$a(\epsilon, \{x_i\}) = \epsilon\, \mathrm{d}\tilde{A} \qquad \text{with} \qquad \tilde{A} = \sum_{l=1}^{N} \tilde{A}_l \log(L_l(\{x_i\}))$$

constant $m \times m$ matrices

- Call set of letters *alphabet:* $\mathcal{A} = \{L_1(\{x_i\}), \ldots, L_N(\{x_i\})\}$

$$\boxed{\mathrm{d}\vec{f}(\epsilon, \{x_i\}) = \epsilon\, \mathrm{d}\tilde{A}\vec{f}(\epsilon, \{x\})}$$

canonical form / ϵ - form

⇨ Solution order by order in ϵ

Current status

- New method has been very successfull: e.g.

  Two-loop non-leptonic B decays [G. Bell, T. Huber '14]
  Two-loop Bhabha scattering [J. M. Henn, A. V. Smirnov, V. A. Smirnov '14]
  Two-loop VV production [J. M. Henn, K. Melnikov, V. A. Smirnov '14; F. Caola, J. M. Henn, K. Melnikov, V. A. Smirnov '14; T. Gehrmann, A. v. Manteuffel, L. Tancredi, E. Weihs '14]
  Three-loop ladder boxes [J. M. Henn, V. A. Smirnov '13]
  Three-loop $gg \to H$ [M. Höschele, J. Hoff, T. Ueda '14]
  Two-loop $H \to Z\gamma$ [R. Bonciani, V. Del Duca, H. Frellesvig, J. M. Henn, F. Moriello, V. A. Smirnov '15; T. Gehrmann, S. Guns, D. Kara '15]
  many more...

How to find such a basis?

Several methods:

- Integrals with constant leading singularities [F. Cachazo '08; J.M. Henn '13]
- Algorithms available for case of one variable [R. Lee '14; J. M. Henn '14; J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. Von Manteuffel, C. Schneider '15]
- $a(\epsilon, \{x_i\})$ linear in ϵ: Magnus/Dyson series approach [M. Argeri, S. Di Vita, P. Mastrolia, E. Mirabella, J. Schlenk, U. Schubert, L. Tancredi '14]
- Diagonal blocks of $a(\epsilon, \{x_i\})$ linear in ϵ [T. Gehrmann, A. von Manteuffel, L. Tancredi, E. Weihs '14]

Public implementatios only for single variable case:

- Implementation of Lees algorithm [O. Gituliar, V. Magerya '16]

Expansion of the transformation law

- Problem description:

Given differential equation with rational $a(\epsilon, \{x_i\})$

Find invertible rational $T(\epsilon, \{x_i\})$ such that:

$$\epsilon \mathrm{d}\tilde{A} = T^{-1} a T - T^{-1} \mathrm{d}T$$

if it exists

- Equivalently:

$$\mathrm{d}T - aT + \epsilon T \mathrm{d}\tilde{A} = 0$$

free of T^{-1}

Idea: Expand in ϵ and Solve for T order by order

Note: $\mathrm{d}\tilde{A}$ is unknown as well!

Expansion of the transformation law

- Transformation law:

$$\boxed{\mathrm{d}T - aT + \epsilon T \mathrm{d}\tilde{A} = 0}$$

- Invariant under: $T \to T g(\epsilon)$

- Choice of $g(\epsilon)$ ⇨ expansion of T starts with $T^{(0)} \neq 0$:

$$T = \sum_{n=0}^{n_{\max}} \epsilon^n T^{(n)}$$

- For simplicity assume finite expansion: $a = \sum_{k=0}^{k_{\max}} \epsilon^k a^{(k)}$

General case works too

Expansion of the transformation law

- Expand transformation law:

$$\mathrm{d}T^{(0)} - a^{(0)}T^{(0)} = 0$$

$$\mathrm{d}T^{(n)} - \sum_{k=0}^{\min(k_{\max},n)} a^{(k)}T^{(n-k)} + T^{(n-1)}\mathrm{d}\tilde{A} = 0$$

⟹ Equation at order n contains only $T^{(k)}$ with $k \leq n$ ⟹ solve order by order

- Highest order $n_{\max}$ unknown ⟹ Check at each order n whether $n = n_{\max}$

Need to find rational solution ⟹ solve with rational Ansatz

Linear ansatz with rational functions

- Transformation law is linear in $T^{(n)}$

  Idea: Ansatz linear in parameters ➡ equations linear in parameters

Ansatz:

$$T^{(n)} = \sum_{k=1}^{R} \tau_k^{(n)} r_k(\{x_i\})$$

$m \times m$ matrix of parameters

rational functions

What kind of rational functions should be used?

- Linearly independent over $\mathbb{C}$ ➡ no apparent redundancies
- As *simple* as possible ➡ cheap computations
- Any rational function $=$ $\mathbb{C}$ - linear combination of *simple* rational functions

  ➡ solution for T can be represented as well

Linear ansatz with rational functions

- Univariate case:

  polynomial division and partial fractioning

  ⇒ $$r(x) = \sum_{j=0}^{J} a_j x^j + \sum_{k=1}^{K} \sum_{n=1}^{D_k} \frac{c_{kn}}{(x - a_k)^n}$$

  rational

  ⇒ Linear combination of monomials and rational functions with one pole

- Multivariate case:

  upon choice of monomial ordering

  - polynomial division generalizes straightforwardly
  - generalization of partial fractioning:

  [E. K. Leinartas '78, A. Raichev '12]

  ⇒ Leinartas decomposition

Two steps: Nullstellensatz decomposition and algebraic independence decomposition

Leinartas decomposition step 1: Nullstellensatz decomposition

Finite set of polynomials $\{f_1, \dots f_m\}$ with no common zero

(weak) Nullstellensatz $\Downarrow$

Exist polynomials $\{h_1, \dots h_m\}$ such that $\sum_{i=1}^{m} h_i f_i = 1$

Example: $\{x,\, 1+xy\}$ has no common zero $\Longrightarrow$ $h_1 = -y,\, h_2 = 1$

Yields decomposition:

multiplication by one $\sum_{i=1}^{m} h_i f_i = 1$

$$\frac{1}{x(1+xy)} = \frac{(-y)(x)+(1)(1+xy)}{x(1+xy)} = \frac{-y}{1+xy} + \frac{1}{x}$$

Leinartas decomposition step 2: algebraic independence

Finite set of algebraically dependent polynomials $\{f_1, \dots f_m\}$

$$\Updownarrow$$

Exists polynomial κ in m variables $\kappa(f_1, \dots, f_m) = 0$

Example:

$\{x,\, y,\, x+y\}$ is algebraically dependent $\Longrightarrow \kappa(Y_1, Y_2, Y_3) = Y_1 + Y_2 - Y_3$

$$\kappa(x, y, x+y) = 0 \quad \Longrightarrow \quad (x) + (y) - (x+y) = 0$$

$$\Longrightarrow \quad \frac{(y) - (x+y)}{-(x)} = 1$$

Yields decomposition:

$$\frac{1}{xy(x+y)} = \frac{(y) - (x+y)}{-x} \frac{1}{xy(x+y)} = -\frac{1}{x^2(x+y)} + \frac{1}{x^2 y}$$

Leinartas decomposition: summary

Nice property:

Maximal number of algebraically independent polynomials given by number of variables

For n variables ⇨ summands with at most n distinct factors

⇨ Reduces number of summands in ansatz

Summary Leinartas decomposition:

Apply step 1 and step 2 repeatedly $\Longrightarrow$ Linear combination of summands

with algebraically independent denominator polynomials and no common zero

Call these summands to be in *Leinartas form*

Ansatz for the $T^{(n)}$ ⇨ rational functions in *Leinartas* form

Constraining the ansatz: Information from the trace

Recall:

$$\mathrm{d}T^{(n)} - \sum_{k=0}^{\min(k_{\max},n)} a^{(k)} T^{(n-k)} + T^{(n-1)} \mathrm{d}\tilde{A} = 0$$

➡ $\mathrm{d}\tilde{A}$ is unknown

But constrained by dlog-form: $\tilde{A} = \sum_{l=1}^{N} \alpha_l \log(L_l(\{x_i\}))$

- denominator factors of $a(\epsilon, \{x_i\})$ ➡ set of polynomials $L_l(\{x_i\})$
- $m \times m$ matrices α_l ➡ unknown parameters

Additional information :

$$\det(T) = C(\epsilon) \exp\left(\int_\gamma \mathrm{Tr}\left[a^{(0)} \right] \right)$$

$$\mathrm{Tr}[\mathrm{d}\tilde{A}] = \mathrm{Tr}[a^{(1)}]$$

- Fully determines transformation of 1-dim. sectors
- Formulas also hold for higher dimensional sectors
- Provide useful information for the ansatz

Algorithm Example: Sunrise reloaded

Input:

Differential form: $$a = \begin{pmatrix} \frac{1-2\epsilon}{x} & \frac{2-3\epsilon}{x} \\ \frac{(-1+2\epsilon)}{(-1+x)x} & \frac{2-\epsilon(3+x)}{(-1+x)x} \end{pmatrix} \mathrm{d}x$$

- Alphabet: $\mathcal{A} = \{-1+x, x\}$ ⇒ $\tilde{A} = \alpha_1 \log(-1+x) + \alpha_2 \log(x)$
- Determinant: $\mathrm{Tr}[a^{(0)}] = \left(\frac{2}{-1+x} - \frac{1}{x}\right) \mathrm{d}x$ ⇒ $\det(T) \propto \frac{(-1+x)^2}{x}$
- Traces: $\mathrm{Tr}[a^{(1)}] = \left(\frac{-4}{-1+x} + \frac{1}{x}\right) \mathrm{d}x$ ⇒ $\mathrm{Tr}[\alpha_1] = -4, \quad \mathrm{Tr}[\alpha_2] = 1$
- Ansatz: $T^{(n)} = \tau_1^{(n)} + \tau_2^{(n)} x + \tau_3^{(n)} \frac{1}{x}$

Algorithm Example: first order

- Insert Ansatz into equation of order ϵ^0: $\mathrm{d}T^{(0)} - a^{(0)}T^{(0)} = 0$

- Each component ⇒ an equation of the type:

$$\frac{-(\tau_1^{(0)})_{11} - 2(\tau_1^{(0)})_{21}}{x} - 2(\tau_2^{(0)})_{21} - 2\frac{(\tau_3^{(0)})_{11} + (\tau_3^{(0)})_{21}}{x^2} = 0$$

- Require to hold for all values $x \neq 0$ ⇒ equations in the parameters:

$$(\tau_1^{(0)})_{11} + 2(\tau_1^{(0)})_{21} = 0, \quad (\tau_2^{(0)})_{21} = 0, \quad (\tau_3^{(0)})_{11} + (\tau_3^{(0)})_{21} = 0$$

- All components together ⇒ linear system of equations for the τ

Algorithm Example: first order

- General solution: $\tau_2^{(0)} = \tau_3^{(0)} = 0, \quad \tau_1^{(0)} = \begin{pmatrix} \lambda_1 & \lambda_2 \\ -\frac{1}{2}\lambda_1 & -\frac{1}{2}\lambda_2 \end{pmatrix}, \quad \lambda_1, \lambda_2 \in \mathbb{R}$

$$\Rightarrow \quad T^{(0)} = \begin{pmatrix} \lambda_1 & \lambda_2 \\ -\frac{1}{2}\lambda_1 & -\frac{1}{2}\lambda_2 \end{pmatrix}$$

- ϵ - form only unique up to constant transformation $\Rightarrow$ redundancy expected

$\Rightarrow$ Fix $\lambda_1, \ \lambda_2$ such that $\mathrm{rank}(T^{(0)})$ is preserved, e.g. $\lambda_1 = 1, \lambda_2 = 0$

- Check series abortion:

$$\det(T^{(0)}) = 0 \quad \Rightarrow \quad n_{\max} \neq 0$$

$\Rightarrow$ Proceed to next order

Algorithm Example: second order

- Equation of order ϵ^1 :

$$\mathrm{d}T^{(1)} - a^{(0)}T^{(1)} - a^{(1)}T^{(0)} + T^{(0)}\mathrm{d}\tilde{A} = 0$$

⇒ Insert ansatz for $T^{(1)}$ and $\tilde{A} = \alpha_1 \log(-1+x) + \alpha_2 \log(x)$

- $T^{(0)}$ is completely fixed ⇒ get linear system of equations for the $\tau_i^{(1)}$ and α_1, α_2

Additionally: $\mathrm{Tr}[\alpha_1] = -4, \quad \mathrm{Tr}[\alpha_2] = 1$

⇒ Solution determines $T^{(1)}$ up to 4 parameters $T^{(1)} = T^{(1)}(\lambda_3, \lambda_4, \lambda_5, \lambda_6)$

- Check series abortion: highest order of $a(\epsilon, x)$ is ϵ^1

⇒ Sufficient to check only the equation of the next order:

$$-a^{(1)}T^{(1)} - T^{(1)}\mathrm{d}\tilde{A} = 0$$

⇒ Solution exists and fixes two of the four parameters

Algorithm Example: second order

- $T^{(1)} = T^{(1)}(\lambda_5, \lambda_6)$ ➡ attains ϵ- form for all λ_5, λ_6 with $\det(T) \neq 0$

➡ remaining two degrees of freedom in the choice of constant transformation

(could not be exploited for $T^{(0)}$ since it did not have full rank)

Output:

Full transformation: $T = \begin{pmatrix} 1 & 0 \\ -\frac{1}{2} & 0 \end{pmatrix} + \epsilon \begin{pmatrix} -\frac{7+x}{2} & \frac{-1+x^2}{2x} \\ 2 & -\frac{-1+x}{2x} \end{pmatrix}$

- ϵ - form: $a' = \begin{pmatrix} 0 & -\frac{\epsilon}{x} \\ \frac{2\epsilon}{(-1+x)} & -\frac{4\epsilon}{(-1+x)} + \frac{\epsilon}{x} \end{pmatrix} \mathrm{d}x$

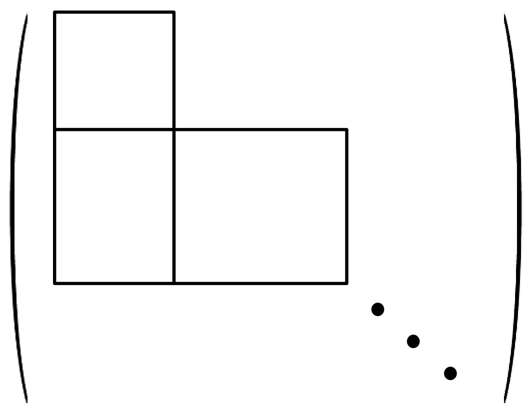
- Determinant as predicted: $\det(T) = -\frac{\epsilon(-1+3\epsilon)(-1+x)^2}{4x}$

Block triangular structure

[S. Caron-Huot, J. M. Henn ’14; T. Gehrmann, A. von Manteuffel, L. Tancredi, E. Weihs ’14; R. Lee ’14]

- Recusion over subsectors ⇒ huge gain in performance
- Derivative of Masterintegral $=$ Sum of Masterintegrals of the same or lower sectors

⇒ Differential Equation in block-triangular form:

$a =$ 

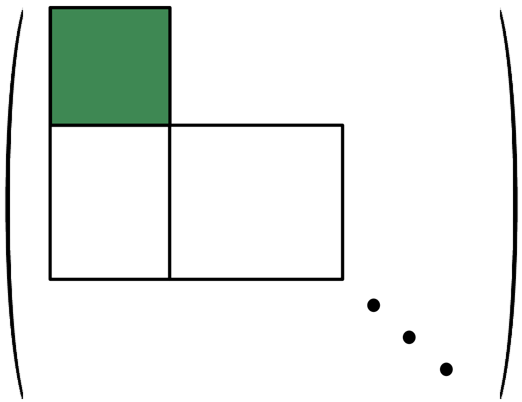
⇒ Compute transformation recursively:

Block triangular structure

[S. Caron-Huot, J. M. Henn ’14; T. Gehrmann, A. von Manteuffel, L. Tancredi, E. Weihs ’14; R. Lee ’14]

- Recusion over subsectors ➡ huge gain in performance
- Derivative of Masterintegral $=$ Sum of Masterintegrals of the same or lower sectors

➡ Differential Equation in block-triangular form:

$a' =$ 

ϵ - form

➡ Compute transformation recursively:

$$\begin{pmatrix} T_1 & 0 \\ 0 & \mathbb{I} \end{pmatrix}$$

algorithm just presented

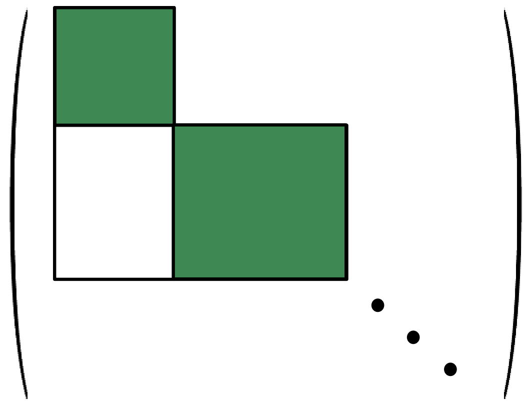
Block triangular structure

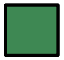
[S. Caron-Huot, J. M. Henn ’14; T. Gehrmann, A. von Manteuffel, L. Tancredi, E. Weihs ’14; R. Lee ’14]

- Recusion over subsectors ⇒ huge gain in performance
- Derivative of Masterintegral $=$ Sum of Masterintegrals of the same or lower sectors

⇒ Differential Equation in block-triangular form:

$$a'' =$$



 ϵ - form

⇒ Compute transformation recursively:

$$\begin{pmatrix} T_1 & 0 \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & T_2 \end{pmatrix}$$

algorithm just presented

Block triangular structure

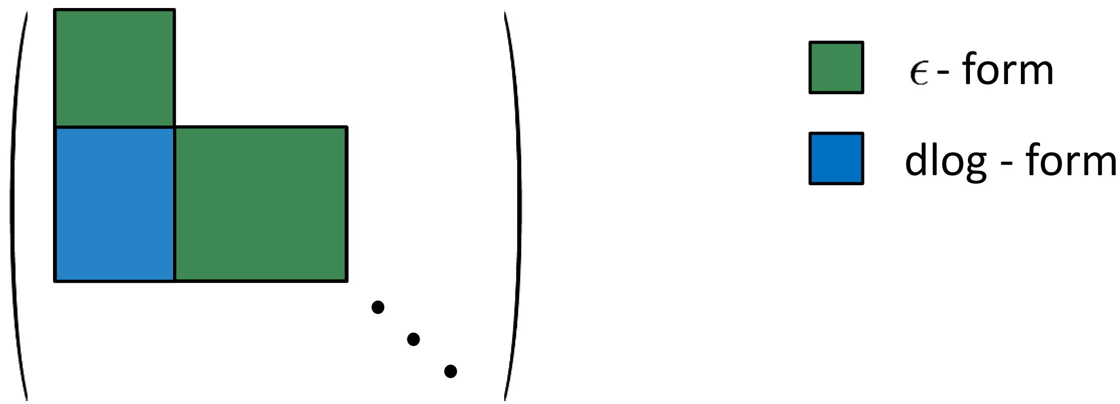
[S. Caron-Huot, J. M. Henn ’14; T. Gehrmann, A. von Manteuffel, L. Tancredi, E. Weihs ’14; R. Lee ’14]

- Recusion over subsectors ➡ huge gain in performance

- Derivative of Masterintegral $=$ Sum of Masterintegrals of the same or lower sectors

➡ Differential Equation in block-triangular form:

$$a''' =$$



➡ Compute transformation recursively:

$$\begin{pmatrix} T_1 & 0 \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & T_2 \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ D & \mathbb{I} \end{pmatrix}$$

algorithm just presented

next slide

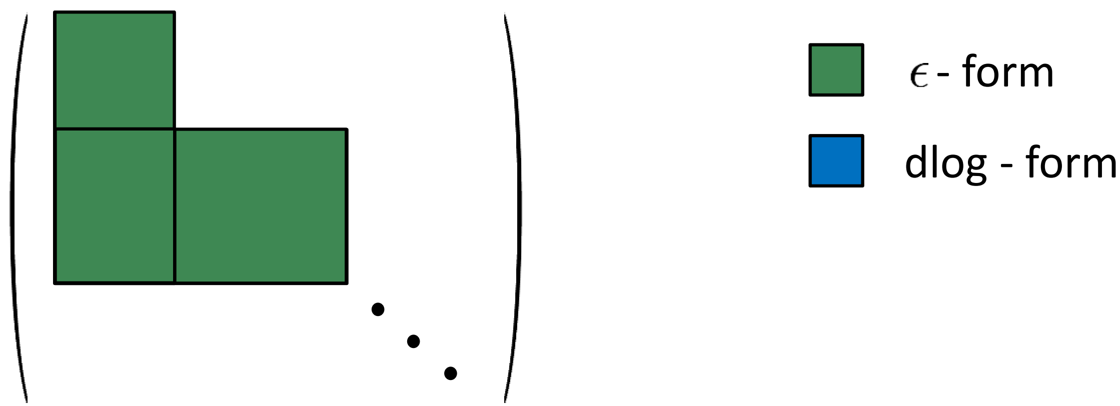
Block triangular structure

[S. Caron-Huot, J. M. Henn '14; T. Gehrmann, A. von Manteuffel, L. Tancredi, E. Weihs '14; R. Lee '14]

- Recusion over subsectors ⇒ huge gain in performance
- Derivative of Masterintegral $=$ Sum of Masterintegrals of the same or lower sectors

⇒ Differential Equation in block-triangular form:

$$a'''' =$$



⇒ Compute transformation recursively:

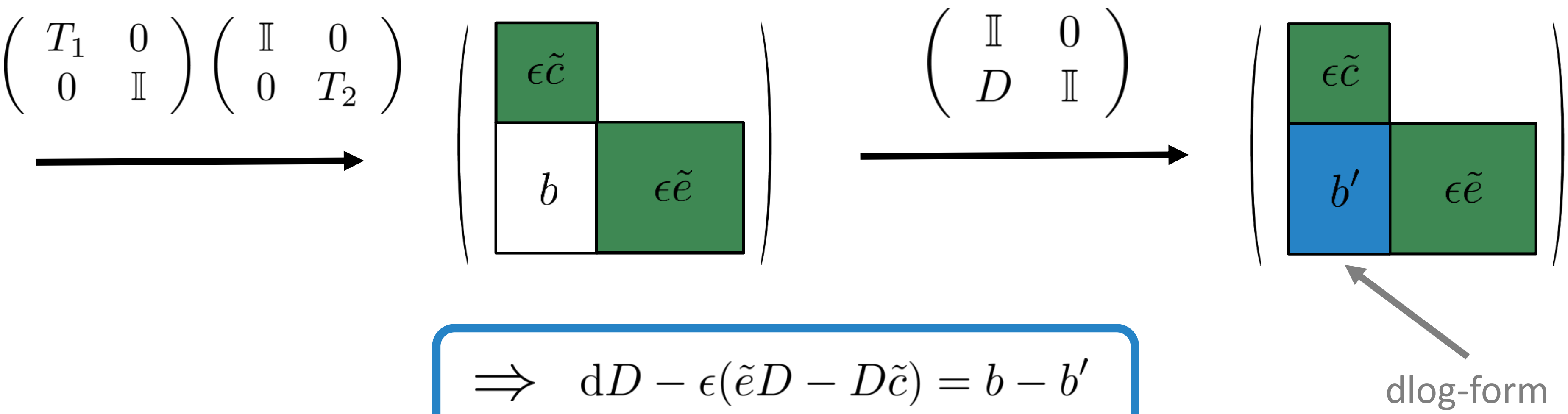
$$\begin{pmatrix} T_1 & 0 \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & T_2 \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ D & \mathbb{I} \end{pmatrix} \begin{pmatrix} K(\epsilon) & 0 \\ 0 & \mathbb{I} \end{pmatrix}$$

algorithm just presented

next slide

[R. Lee '14]

Off – diagonal blocks



$$\boxed{\Rightarrow \quad \mathrm{d}D - \epsilon(\tilde{e}D - D\tilde{c}) = b - b'}$$

Analogous strategy:

- Extract rational ansatz for D from b
- b' in dlog-form ➡ only alphabet needed for ansatz

Implementation in Mathematica

Mathematica package *CANONICA*

- Implementation of the presented algorithm [C.M. '16]

Flexibility:

- High level functions: Input: differential equation
  Output: Transformation to ϵ- form
- Low level functions: Perform particular steps of the algorithm

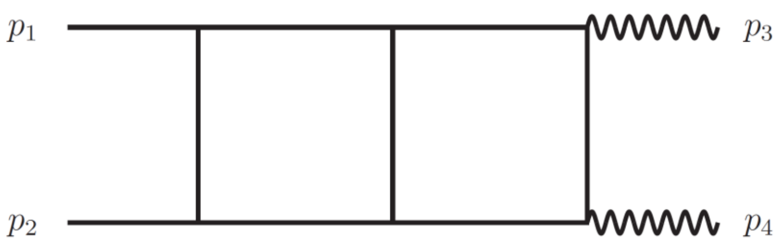
- Successfully applied to many non-trivial examples

soon to be published...

Kinematics: [T. Gehrmann, L. Tancredi, E. Weihs ’13; J.M. Henn, K. Melnikov, V.A. Smirnov ’14]

$$p_1^2 = 0, \quad p_2^2 = 0, \quad p_3^2 = m_3^2, \quad p_4^2 = m_4^2 \qquad s = (p_1 + p_2)^2, \quad t = (p_1 - p_3)^2$$

Depend on 3 scales: $(1+x)(1+xy) = \frac{s}{m_3^2}, \quad -xz = \frac{t}{m_3^2}, \quad x^2 y = \frac{m_4^2}{m_3^2}$

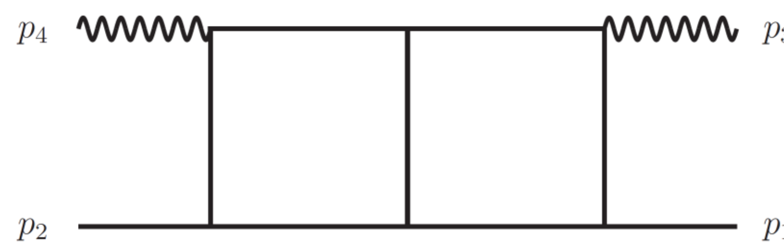


31 master integrals

Alphabet contains 12 letters:

$$\mathcal{A} = \{x,\, 1+x,\, 1-y,\, y,\, 1+xy,\, 1+x(1+y-z),\, 1-z,\, z,\, z-y,\, z+xy,\, 1+xz\, 1+(1+x)y-z\}$$

CANONICA runtime: < 1h



29 master integrals

Alphabet contains 14 letters:

$$\mathcal{A} = \{x,\, 1+x,\, 1-y,\, y,\, 1+xy,\, 1+x(1+y-z),\, 1-z,\, 1+(1+x)y-z\, z,\, z-y,\, z+xy,\, 1+xz,\, z-y+yz+xyz,\, z-xy+xz+xyz)\}$$

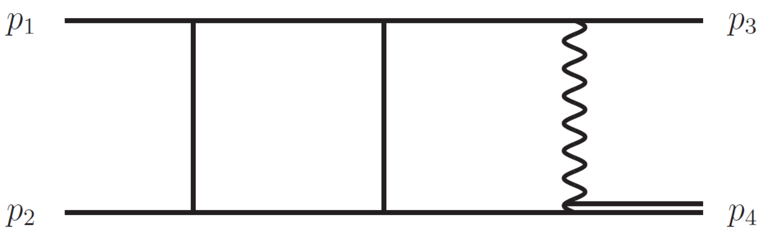
CANONICA runtime: < 1h

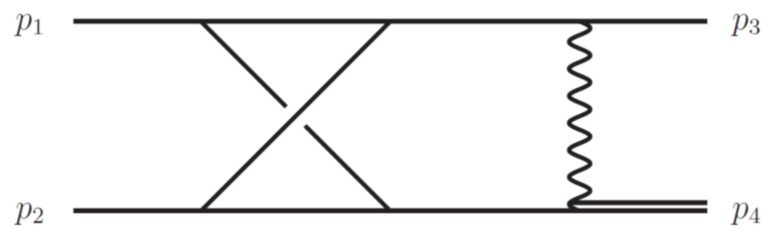
Single Top Quark Production @ NNLO QCD

Kinematics:

$$p_1^2 = 0, \quad p_2^2 = 0, \quad p_3^2 = 0, \quad p_4^2 = m_t^2 \qquad s = (p_1 + p_2)^2, \quad t = (p_2 - p_3)^2$$

Depend on 3 scales: $x = \frac{s}{m_W^2}, \quad y = \frac{t}{m_W^2}, \quad z = \frac{m_t^2}{m_W^2}$





31 master integrals

Alphabet contains 11 letters:

$$\mathcal{A} = \{x,\, y,\, x+y,\, x-z,\, y-z,\, x+y-z,\, 1+x+y-z,\, -1+z,\, z-1-x+z,\, y(-1+z)+(1+x-z)z\}$$

CANONICA runtime: < 1h

35 master integrals

Alphabet contains 13 letters:

$$\mathcal{A} = \{x,\, -1+y,\, y,\, x+y,\, x-z,\, 1+x-z,\, y-z,\, x+y-z,\, 1+x+y-z,\, -1+z,\, z,\, x+y(1-z),\, x(-1+y)+y(y-z)\}$$

CANONICA runtime: a few hours

Other topologies entail irrational letters ➡ requires extension of the algorithm

Conclusions and Outlook

Conclusions:

- New algorithm applicable to the general case of rational $a(\epsilon, \{x_i\})$
- Implemented in Mathematica Package *CANONICA*
- Tested for non-trivial examples

Outlook:

- Study further applications
- Better understanding of how to automatically choose ansatz
- Extension to irrational letters

Thank You!