INTERPLAY OF COLOUR-KINEMATICS DUALITY AND ANALYTIC CALCULATION OF MULTI-LOOP SCATTERING AMPLITUDES: ONE- AND TWO-LOOPS

William J. Torres Bobadilla
Institut de Física Corpuscular
Universitat de València - CSIC

In collaboration with:
J. Llanes, G. Rodrigo;
P. Mastrolia, G. Ossola, M. Passera, T. Peraro, A. Primo and U. Schubert

RADCOR 2017
28-09-2017, Hotel Hollweger, St. Gilgen, Austria.
Outline

- Colour-Kinematics duality
  - C/K duality @ tree-level in $d$
  - Integral relations @1L
- Calculation of multi-loop scattering amplitudes
  - Adaptive integrand decomposition
  - Improved reduction @1- and 2-loops
  - Automated 2-loop reduction for any generic processes
- Results
- Conclusions/Outlook
Colour-kinematics duality
**Colour-Kinematics duality**

- Relations between kinematic numerators
- Provides symmetries among amplitudes
- Strong Connection between gravity and Yang-Mills amplitudes
- Construction of gravity from knowledge of Yang-Mills amplitudes

Consider Jacobi identity

\[
\begin{align*}
&c \left( \begin{array}{c} 1 \\ 2 \\ 3 \\
\end{array} \right) = -c \left( \begin{array}{c} 1 \\ 2 \\ 3 \\
\end{array} \right) \\
&-c \left( \begin{array}{c} 1 \\ 2 \\ 3 \\
\end{array} \right) - c \left( \begin{array}{c} 1 \\ 2 \\ 3 \\
\end{array} \right) + c \left( \begin{array}{c} 1 \\ 2 \\ 3 \\
\end{array} \right) = 0
\end{align*}
\]

Feynman graphs —> cubic vertices

\[
A_{m}^{\text{tree}}(1, 2, \ldots, m) = \sum_{i=1}^{N} \frac{c_{i} n_{i}}{D_{i}} \quad D_{i} = \prod_{\alpha_{i}} s_{\alpha_{i}}
\]

- Satisfied automatically for 4-point tree amplitudes
  \[-n_{1} - n_{2} + n_{3} = 0 \quad [Zhu (1980)]\]
- For high multiplicity, it is not trivially satisfied

Consider the adjoint and fundamental representation:

\[
\begin{align*}
&- \tilde{f}^{a_{1}a_{2}x} \tilde{f}^{a_{3}a_{4}x} - \tilde{f}^{a_{1}a_{4}x} \tilde{f}^{a_{2}a_{3}x} + \tilde{f}^{a_{1}a_{3}x} \tilde{f}^{a_{2}a_{4}x} = 0 \\
&- \tilde{f}^{a_{1}a_{2}x} T^{x} - T^{a_{1}} T^{a_{2}} + T^{a_{2}} T^{a_{1}} = 0
\end{align*}
\]

[Bern, Carrasco, Johansson (2008), (2010)]
[Johansson, Ochirov (2014), (2015)]
[de la Cruz, Kniss, Weinzierl (2015), (2016)]
[Boels, Isermann (2012)]
[Mastrolia, Primo, Schubert, W.J.T. (2015)]
Colour-Kinematics duality

Construct an off-shell current

[LLanes, Rodrigo, W.J.T. (to appear)]
Colour-Kinematics duality

Construct an off-shell current

\[
J = -n \left( \begin{array}{ccc} 1 & 2 \\ 3 & 4 \end{array} \right) - n \left( \begin{array}{ccc} 1 & 3 \\ 2 & 4 \end{array} \right) + n \left( \begin{array}{ccc} 1 & 4 \\ 2 & 3 \end{array} \right)
\]

\[
p_{1}^{\mu} \left( \begin{array}{ccc} p_{4}, \mu_{4} \\ p_{3}, \mu_{3} \end{array} \right) (2 \leftrightarrow 3)
\]

\[
p_{1}^{\mu} \left( \begin{array}{ccc} p_{4}, \mu_{4} \\ p_{3}, \mu_{3} \end{array} \right) + p_{2} \left( \begin{array}{ccc} p_{3} \\ p_{1, \mu_{1}} \end{array} \right) + \{1234 \rightarrow \{4321}\}
\]

\[
p_{1}^{\mu} \left( \begin{array}{ccc} p_{4}, \mu_{4} \\ p_{3}, \mu_{3} \end{array} \right) \pm \text{cyc. perm.}
\]
Colour-Kinematics duality

Construct an off-shell current

Embed off-shell currents in a richer topology

Remark:
\[ \Pi^\mu_{i\nu_i} (p_i ; q_i) = -g^\mu_{i\nu_i} + \frac{p_i^\mu_i q_i^{\nu_i} + p_i^{\nu_i} q_i^\mu_i}{p_i \cdot q_i} \]

\[ p_i, \mu; \Pi^\mu_{i\nu_i} (p_i ; q_i) = p_i^2 \frac{q_i^\nu}{p_i \cdot q_i} \]

\[ \phi_i \phi_i = p_i^2 \]

In axial gauge

[Maistrolia, Primo, Schubert, W.J.T. (2015)]

[LLanes, Rodrigo, W.J.T. (to appear)]
Colour-Kinematics duality

Decompose off- into on-shell momenta

\[ p_i^\alpha = r_i^\alpha + \frac{p_i^2}{2q \cdot r_i} q^\alpha \]

Extract full dependence on the off-shell momenta

\[ \sum_{\lambda=1}^{d_s-2} \varepsilon_{\lambda(d_s)}^\alpha (p_i) \varepsilon_{\lambda(d_s)}^{*\beta} (p_i) = \sum_{\lambda=1}^{d_s-2} \varepsilon_i^\alpha \varepsilon_i^{*\beta} + \frac{p_i^2}{(r_i \cdot q)^2} q^\alpha q^{\beta}, \]

\[ \sum_{\lambda=1}^{2(d_s-2)/2} u_{\lambda(d_s)} (p_i) \bar{u}_{\lambda(d_s)} (p_i) = \sum_{\lambda=1}^{2(d_s-2)/2} u_i \bar{u}_i + \frac{p_i^2}{2(r_i \cdot q)} q. \]

Completeness relations

Construct multi-loop numerator

\[ N_g = N_{g\mu_1...\mu_4} X^{\mu_1...\mu_4}, \quad N_{g\mu_1...\mu_4} = J^{[\mu_1...\mu_4} \Pi_{\mu_1\nu_1} (p_1, q) \ldots \Pi_{\mu_4\nu_4} (p_4, q). \]

Residual kinematic dependence

Numerator is decomposed in product of squared momenta

\[ N_{g\nu_1...\nu_4} = \frac{1}{2} \sum_{i,j,k,l=1}^{4} \epsilon_{ijkl} p_i^2 \left( A_{ijkl} \varepsilon_{ij}^{\nu_1 \nu_j} \varepsilon_{kl}^{\nu_k \nu_l} + B_{ijkl} \varepsilon_{ij}^{\nu_1 \nu_k} Q_i^{\nu_1 \nu_l} + C_{ijkl} p_j^2 q_{ij}^{\nu_1 \nu_j} \varepsilon_{kl}^{\nu_k \nu_l} \right), \]

\[ A, B \text{ and } C \text{ are completely independent of } p_i^2 \]

By imposing on-shellness of the four particles

\[ = 0 \]
Colour-Kinematics duality

Decompose off- into on-shell momenta

\[ p_i^\alpha = r_i^\alpha + \frac{p_i^2}{2q \cdot r_i} q^\alpha \]

Extract full dependence on the off-shell momenta

\[ \sum_{\lambda=1}^{d_s-2} \varepsilon_{\lambda(d_s)}^\alpha (p_i) \varepsilon_{\lambda(d_s)}^* (p_i) \]

\[ \sum_{\lambda=1}^{2(d_s-2)/2} u_{\lambda(d_s)} (p_i) \bar{u}_{\lambda(d_s)} (p_i) \]

Completeness relations

Construct multi-loop numerator

\[ N_g = N_{g \mu_1 \ldots \mu_4} X^{\mu_1 \ldots \mu_4} , \quad N_{g \mu_1 \ldots \mu_4} = J_{g}^{\nu_1 \ldots \nu_4} \Pi_{\mu_1 \nu_1} (p_1, q) \ldots \Pi_{\mu_4 \nu_4} (p_4, q) . \]

Residual kinematic dependence

Numerator is decomposed in product of squared momenta

\[ N_g^{\nu_1 \ldots \nu_4} = \frac{1}{2} \sum_{i,j,k,l} \varepsilon_{ijkl} p_i^2 \left( A_{ijkl} \varepsilon_{ij}^{\nu_j} \varepsilon_{kl}^{\nu_k} + B_{ijkl} \varepsilon_{jk}^{\nu_j} \varepsilon_{kl}^{\nu_k} + C_{ijkl} q_{ij} q_{kl} \varepsilon_{ij}^{\nu_j} \varepsilon_{kl}^{\nu_k} \right) , \]

\[ A, B \text{ and } C \text{ are completely independent of } p_i^2 \]

What about \( p_i^2 p_j^2 p_k^2 \) and \( p_i^2 p_j^2 p_k^2 p_l^2 \) contributions?

By imposing on-shellness of the four particles

\[ = 0 \]
**Colour-Kinematics duality**

Decompose off- into on-shell momenta

\[ p_i^\alpha = r_i^\alpha + \frac{p_i^2}{2q \cdot r_i} q^\alpha \]

Extract full dependence on the off-shell momenta

\[
\begin{align*}
\sum_{\lambda=1}^{d_s-2} \varepsilon_{\lambda(d_s)}^\alpha (p_i) \varepsilon_{\lambda(d_s)}^{*\beta} (p_i) &= \sum_{\lambda=1}^{d_s-2} \varepsilon_i^\alpha \varepsilon_i^{*\beta} + \frac{p_i^2}{(r_i \cdot q)^2} q^\alpha q^\beta, \\
\sum_{\lambda=1}^{2(d_s-2)/2} u_{\lambda(d_s)} (p_i) \bar{u}_{\lambda(d_s)} (p_i) &= \sum_{\lambda=1}^{2(d_s-2)/2} u_i \bar{u}_i + \frac{p_i^2}{2(r_i \cdot q)} q.
\end{align*}
\]

Completeness relations

Construct multi-loop numerator

\[ N_g = N_{g \mu_1 \ldots \mu_4} X^{\mu_1 \ldots \mu_4}, \quad N_{g \mu_1 \ldots \mu_4} = J_{g}^{\nu_1 \ldots \nu_4} \Pi_{\mu_1 \nu_1} (p_1 (\overline{q})) \ldots \Pi_{\mu_4 \nu_4} (p_4 (\overline{q})). \]

Residual kinematic dependence

Numerator is decomposed in product of squared momenta

\[ N_{g \nu_1 \ldots \nu_4} = \frac{1}{2} \sum_{i,j,k,l=1}^{4} \epsilon_{ijkl} p_i^2 \left( A_{ijkl} \varepsilon_{ij}^{\nu_j} \varepsilon_{kl}^{\nu_k} + B_{ijkl} \varepsilon_{jk}^{\nu_j} Q_{kl}^{\nu_k \nu_l} + C_{ijkl} p_j^2 q^{\nu_j \nu_l} \varepsilon_{kl}^{\nu_k} \right), \]

\[ A, B \text{ and } C \text{ are completely independent of } p_i^2 \]

What about \( p_j^2 p_k^2 \) and \( p_i^2 p_j^2 p_k^2 \) contributions?

Same reference momentum \( q \) for all internal gluons!

By imposing on-shellness of the four particles

\[
\begin{align*}
\sum_{i,j=1}^{4} r_i^2 &= 0, \\
\sum_{i,j,k=1}^{4} r_i^2 r_j r_k &= 0, \\
\sum_{i,j,k,l=1}^{4} r_i^2 r_j r_k r_l &= 0.
\end{align*}
\]
Colour-Kinematics duality

One-loop example

\[\int \frac{d^d \ell}{(2\pi)^d} \frac{1}{D_0 D_1 D_2} \left( \Phi_1 \right) = \int \frac{d^d \ell}{(2\pi)^d} \frac{1}{D_0 D_1 D_2} \left( \Phi_2 \right) = 0 \]

\[\int \frac{d^d \ell}{(2\pi)^d} \frac{1}{D_0 D_1} \left( \Phi_3 \right) = \int \frac{d^d \ell}{(2\pi)^d} \frac{1}{D_0 D_1} \left( \Phi_4 \right) = 0 \]

From string theory

\[\int \frac{d^d \ell}{(2\pi)^d} \left( \Phi_5 \right) = 0 \]

\[\int \frac{d^d \ell}{(2\pi)^d} \left( \Phi_6 \right) = 0 \]

- Satisfied automatically for 4-point one-loop amplitudes
- Off-shell decomposition eliminates redundant terms
- Interesting integral relations at one-loop level
- Straightforward application with Loop-Tree duality formalism

[LLanes, Rodrigo, W.J.T. (to appear)]
[LLanes, Rodrigo, W.J.T. (work in progress)]

[Tourkine, Vanhove (2016)]
[Ochirov, Tourkine, Vanhove (2017)]

Work in progress coming soon!
Multi-loop Scattering amplitudes
One-loop scattering amplitudes

Deal with with integrals of the form

\[ I_{i_1 \cdots i_k} \left[ \mathcal{N}(\bar{l}, p_i) \right] = \int d^d \bar{l} \frac{\mathcal{N}_{i_1 \cdots i_k}(\bar{l}, p_i)}{D_{i_1} \cdots D_{i_k}} \]

Numerator and denominators are polynomials in the integration variable

Tensor reduction

\[ A_{n}^{(1), D=4}(\{p_i\}) = \sum_{K_2} C_{4,K4}^{[0]} + \sum_{K_3} C_{3,K3}^{[0]} + \sum_{K_2} C_{2,K2}^{[0]} + \sum_{K_1} C_{1,K1}^{[0]} \]

[Passarino - Veltman (1979)]

- ✔ Cut-constructible amplitude -> determined by its branch cuts
- ✔ All one-loop amplitudes are cut-constructible in dimensional regularisation.
- ✔ Master integrals are known
One-loop scattering amplitudes

Deal with with integrals of the form

\[ I_{i_1 \cdots i_k} \left[ N(\bar{l}, p_i) \right] = \int d^d\bar{l} \frac{N_{i_1 \cdots i_k} (\bar{l}, p_i)}{D_{i_1} \cdots D_{i_k}} \]

Numerator and denominators are polynomials in the integration variable

Tensor reduction

Unitarity based methods

\[ 2\pi\delta^{(+)} (p^2 - m^2) \rightarrow \frac{i}{p^2 - m^2 - i\epsilon} \]

**cut-4 ::** Britto Cachazo Feng

**cut-3 ::** Forde

Bjerrum-Bohr, Dunbar, Ita, Perkins

Mastrolia

**cut-2 ::** Bern, Dixon, Dunbar, Kosower.

Britto, Buchbinder, Cachazo, Feng.

Britto, Feng, Mastrolia.

[Passarino - Veltman (1979)]

>> Dunbar’s talk

>> Kosower’s talk
One-loop integrand decomposition

Recall

\[ \int \frac{\mathcal{N}(l)}{D_1 \cdots D_n} \, d^4\ell \left( \sum_{i \ll m} c_{ijkm} \int \frac{1}{D_i D_j D_k D_m} + \sum_{i \ll k} c_{ijk} \int \frac{1}{D_i D_j D_k} + \sum_{i < j} c_{ij} \int \frac{1}{D_i D_j} + \sum_i \int d^4\ell \frac{1}{D_i} \right) \]

Find an identity between integrands. Moreover,

\[ \frac{\mathcal{N}(l)}{D_1 \cdots D_n} \neq \sum_{i \ll m} c_{ijkm} \frac{1}{D_i D_j D_k D_m} + \sum_{i \ll k} c_{ijk} \frac{1}{D_i D_j D_k} + \sum_{i < j} c_{ij} \frac{1}{D_i D_j} + \sum_i \frac{1}{D_i} \]

Suppose a multipole decomposition

\[ \frac{\mathcal{N}(l)}{D_1 \cdots D_n} = \sum_{i \ll m} \tilde{c}_{ijkm} \frac{\Delta_{ijkm}(l)}{D_i D_j D_k D_m} + \sum_{i \ll k} \tilde{c}_{ijk} \frac{\Delta_{ijk}(l)}{D_i D_j D_k} + \sum_{i < j} \tilde{c}_{ij} \frac{\Delta_{ij}(l)}{D_i D_j} + \sum_i \tilde{c}_i \frac{\Delta_i(l)}{D_i} \]

- Residues $\Delta$ are made of Irreducible Scalar Products
- Can we find parametric expressions for $\Delta$’s in 4- or d-dimensions?
- Parametric expressions
One-loop integrand decomposition

Recall

\[ \int d^4\ell \frac{\mathcal{N}(l)}{D_1 \cdots D_n} = \sum_{i \ll m} c_{ijk} \int d^4\ell \frac{1}{D_i D_j D_k D_m} + \sum_{i \ll k} c_{ij} \int d^4\ell \frac{1}{D_i D_j D_m} + \sum_{i < j} c_{i} \int d^4\ell \frac{1}{D_i D_j} \]

Find an identity between integrands. Moreover,

\[ \frac{\mathcal{N}(l)}{D_1 \cdots D_n} \neq \sum_{i \ll m} c_{ijk} \frac{1}{D_i D_j D_k D_m} + \sum_{i \ll k} c_{ij} \frac{1}{D_i D_j D_k} + \sum_{i < j} c_{i} \frac{1}{D_i D_j} \]

Suppose a multipole decomposition

\[ \frac{\mathcal{N}(l)}{D_1 \cdots D_n} = \sum_{i \ll m} \tilde{c}_{ijk} \frac{\Delta_{ijk m}(l)}{D_i D_j D_k D_m} + \sum_{i \ll k} \tilde{c}_{ij} \frac{\Delta_{ijk}(l)}{D_i D_j D_k} + \sum_{i < j} \tilde{c}_{ij} \frac{\Delta_i(l)}{D_i D_j} + \sum_i \tilde{c}_i \frac{\Delta_i(l)}{D_i} \]

- Residues \( \Delta \) are made of **Irreducible Scalar Products**
- Can we find parametric expressions for \( \Delta \)'s in 4- or d-dimensions?
- Parametric expressions **Yes.** General way → Use multivariate polynomial division coefficients are fixed by sampling the numerators on the cut

>> Papadopoulos’ talk

[Ossola, Papadopoulos, Pittau (2006)]
[Ellis, Giele, Kunszt, Melnikov (2007)]
[Mastrolia, Ossola, Papadopoulos, Pittau (2008)]
One-loop integrand decomposition

Loop parametrisation

\[
l_i^\alpha = p_i^\alpha + x_1 e_1^\alpha + x_2 e_2^\alpha + x_3 e_3^\alpha + x_4 e_4^\alpha
\]

\[
\mathcal{N} (\vec{l}) = \mathcal{N} (l, \mu^2) = \mathcal{N} (z) \quad z = \{x_1, x_2, x_3, x_4, \mu^2\}
\]

Multivariate polynomial division

Write the numerator in terms of Irreducible polynomials

\[
\mathcal{I} \equiv \frac{\mathcal{N}}{D_0 \cdots D_{n-1}} = \sum_{k=1}^{5} \sum_{\{i_1, \ldots, i_k\}} \frac{\Delta_{i_1 \cdots i_k}}{D_{i_1} \cdots D_{i_k}}
\]

sum of integrands with five or less denominators

\[\Delta_{i_1 \cdots i_k}\] Made of Irreducible Scalar Products
Cannot be expressed in terms of denominators

Generic structure of the residue

\[
\Delta_{i_1 i_2 i_3 i_4 i_5} = c_0 \mu^2,
\]

\[
\Delta_{i_1 i_2 i_3 i_4} = c_0 + c_1 x_{4,v} + \mu^2 \left( c_2 + c_3 x_{4,v} + \mu^2 c_4 \right),
\]

\[
\Delta_{i_1 i_2 i_3} = c_0 + c_1 x_4 + c_2 x_4^2 + c_3 x_4^3 + c_4 x_3 + c_5 x_3^2 + c_6 x_3^3 + \mu^2 \left( c_7 + c_8 x_4 + c_9 x_3 \right),
\]

\[
\Delta_{i_1 i_2} = c_0 + c_1 x_1 + c_2 x_2 + c_3 x_4 + c_4 x_4^2 + c_5 x_3 + c_6 x_3^2 + c_7 x_1 x_4 + c_8 x_1 x_3 + c_9 \mu^2,
\]

\[
\Delta_{i_1} = c_0 + c_1 x_1 + c_2 x_2 + c_3 x_3 + c_4 x_4,
\]
Adaptive integrand decomposition (AID)

- Splits $d=4-2\varepsilon$ into parallel and orthogonal directions
- Nice properties for less than 5 external legs

Loop momenta

$$\vec{l}_i^\alpha = \vec{l}_{\parallel i}^\alpha + \lambda_i^\alpha$$

$$\vec{l}_i^\alpha = \sum_{j=1}^{d_\parallel} x_{ji} e_j^\alpha$$

$$\lambda_i^\alpha = \sum_{j=d_\parallel+1}^{4} x_{ji} e_j^\alpha + \mu_i^\alpha$$

Numerator and denominators depend on different variables

$$d_\parallel = n - 1$$

$$d_\perp = (5 - n) - 2\varepsilon$$

loop integrals

$$\int \prod_i d^{d_\parallel} \vec{l}_{\parallel i}^\alpha \int \prod_{1 \leq i \leq j \leq \ell} d\lambda_{ij} G(\lambda_{ij}) \frac{d_\perp - 1 - \ell}{2} \int d\Theta_\perp \frac{N(\vec{l}_{\parallel i}, \lambda_{ij}\Theta_\perp)}{D_1(l_{\parallel i}, \lambda_{ij}) \cdots D_m(l_{\parallel i}, \lambda_{ij})}$$

Expand in Gegenbauer polynomials

$$\int d\Theta_\perp = \int_{-1}^{1} \prod_{i=1}^{4-d_\parallel} \prod_{j=1}^{\ell} d\cos\theta_{i,j-1} \left(\sin\theta_{i,j-1}\right)^{d_\perp - i - j - 1}$$

$$\int_{-1}^{1} d\cos\theta (\sin\theta)^{2\alpha-1} C_{n}^{(\alpha)}(\cos\theta) C_{m}^{(\alpha)}(\cos\theta) = \delta_{mn} \frac{2^{1-2\alpha} \pi \Gamma(n + 2\alpha)}{n!(n + \alpha) \Gamma^2(\alpha)}$$
Adaptive integrand decomposition (AID)

- Splits $d=4-2\varepsilon$ into parallel and orthogonal directions
- Nice properties for less than 5 external legs

\[ d = d_\parallel + d_\perp \]

\[ d_\parallel = n - 1 \]

\[ d_\perp = (5 - n) - 2\varepsilon \]

Loop momenta

\[ \vec{l}_i^\alpha = \vec{t}_{\parallel i}^\alpha + \lambda_i^\alpha \]

\[ \vec{l}_i^\alpha = \sum_{j=1}^{d_\parallel} x_{ji} e_j^\alpha, \quad \lambda_i^\alpha = \sum_{j=d_\parallel+1}^{4} x_{ji} e_j^\alpha + \mu_i^\alpha, \quad \lambda_{ij} = \sum_{l=d_i+1}^{4} x_{li} x_{lj} + \mu_{ij} \]

Numerator and denominators depend on different variables

\[ \prod_i d^d_{\parallel} \int \prod_{1\leq i \leq j \leq \ell} d\lambda_{ij} G(\lambda_{ij}) \left( \frac{d_\perp - 1}{2} \right)^{\frac{d_\perp - 1 - \ell}{2}} \int d\Theta_\perp \frac{N(\vec{t}_{\parallel i}, \lambda_{ij} \Theta_\perp)}{D_1(\vec{t}_{\parallel i}, \lambda_{ij}) \cdots D_m(\vec{t}_{\parallel i}, \lambda_{ij})} \]

Straightforward integration of transverse components

Expand in Gegenbauer polynomials

\[ \int d\Theta_\perp = \int_{-1}^{1} \prod_{i=1}^{\ell} \prod_{j=1}^{\ell} d\cos \theta_{i+j-1} \cos \theta_{i+j-1} \]

\[ \int_{-1}^{1} d\cos \theta (\sin \theta)^{2\alpha-1} C_n^{(\alpha)}(\cos \theta) C_m^{(\alpha)}(\cos \theta) = \delta_{mn} \frac{2^{1-2\alpha} \pi \Gamma(n+2\alpha)}{n!(n+\alpha)\Gamma^2(\alpha)} \]

and identification of spurious terms
Adaptive integrand decomposition (AID)

Algorithm

- For each integrand, adapt longitudinal and parallel components
- Denominators depend on the minimal set of variables
- Loop components expressed as linear combination of denominators
- Poly division and integration reduced to substitution rules
- Extra dimension variables are always reducible

Recipe in 3 steps

1) Divide and get \( \Delta(\vec{l}_{||i}, \lambda_{ij}, \Theta_\perp) \)
2) Integrate out transverse variables \( \Theta_\perp \)
3) Divide again to get rid of \( \lambda_{ij} \)

Features

- Final decomposition in terms of ISPs
- No need for TID
- Output ready to apply IBPs
- @1L no need of any integral identity
Adaptive integrand decomposition (AID)

Algorithm

For each integrand, adapt longitudinal and parallel components
Denominators depend on the minimal set of variables
Loop components expressed as linear combination of denominators
Poly division and integration reduced to substitution rules
Extra dimension variables are always reducible

Recipe in 3 steps

1) Divide and get $\Delta(\bar{l}_{||i}, \lambda_{ij}, \Theta_\perp)$
2) Integrate out transverse variables $\Theta_\perp$
3) Divide again to get rid of $\lambda_{ij}$

Features

- Final decomposition in terms of ISPs
- No need for TID
- Output ready to apply IBPs
- @1L no need of any integral identity

Algorithm requires automation
**AIDA: a Mathematica implementation**

[Mastrolia, Peraro, Primo, W.J.T. (work in progress)]

**AMPLITUDE GENERATOR**
(FeynArts+FeynCalc, QGRAF...)

**IBPs REDUCTION CODE**
(Reduze, FIRE, Kira, Azurite...)

**AIDA**
(Adaptive Integrand Decomposition Algorithm)

**COMPUTE MIs**

- **Analytically** (Loopedia)
- **Numerically** (SecDec, FIESTA...)

>> Hann’s talk
>> Jahn’ talk

>> Rietkerk’s talk
>> Larsen’s talk
AIDA: a Mathematica implementation

[William J. Torres Bobadilla]

Cut solution parametrisation
- Identify parent topology
- Analyse kinematics for all cuts
- Store adaptive parametrisation
- Organise cuts in Jobs

Read numerators

Divide (apply substitution rules)

Numerator = \Delta^{\text{int}}(\bar{l}_{ij}, \lambda_{ij})

Residues depend on $\Theta_\perp$?
- No
- Yes

Integrate (apply substitution rules)

Analytically (Loopedia)
- Numerically (SecDec, FIESTA...)

AMPLITUDE GENERATOR
(FeynArts+FeynCalc, QGRAF...)

IBPs REDUCTION CODE
(Reduze, FIRE, Kira, Azurite...)

COMPUTE MIs

William J. Torres Bobadilla

[Mastrolia, Peraro, Primo, W.J.T. (work in progress)]

>> Hann’s talk
>> Jahn’s talk
>> Rietkerk’s talk
>> Larsen’s talk
AIDA: a Mathematica implementation

The code is designed for general one- and two-loop numerical and analytical calculations.

**AMPLITUDEN GENERATOR**
(FeynArts+FeynCalc, QGRAF...)

**IBPs REDUCTION CODE**
(Reduze, FIRE, Kira, Azurite...)

**COMPUTE MIs**
(Loopedia, Analytically
(SecDec, FIESTA...), Numerically)

- **Cut solution parametrisation**
  - Identify parent topology
  - Analyse kinematics for all cuts
  - Store adaptive parametrisation
  - Organise cuts in Jobs

- **Read numerators**

- **Divide**
  - (apply substitution rules)

- **Integrate**
  - (apply substitution rules)

**Numerators**
\[
\Delta^\text{int}(I_{i}, \lambda)\]

**Residues depend on \(\Theta\)?**

- **No**
- **Yes**

[Mastrolia, Peraro, Primo, W.J.T. (work in progress)]

William J. Torres Bobadilla

>> Hann’s talk
>> Jahn’s talk
>> Rietkerk’s talk
>> Larsen’s talk
**AIDA for muon-electron scattering**

- Recent proposal for the determination of the hadronic contribution to the muon from the measurement of *muon-electron scattering* $g - 2$
  
  [Carloni Calame, Passera, Trentadue, Venanzoni (2015)]
  [Abbiendi, Carloni Calame, Marconi et al (2017)]

- In the massless electron limit, 4-point process depending on 3 scales

  \[ s = (p_1 + p_2)^2 \quad t = (p_2 + p_3)^2 \]
  \[ m_e^2 \simeq 0 \quad u = -s - t + 2m^2 \]

  \[ e(p_1) + \mu(p_4) \rightarrow e(-p_2) + \mu(-p_3) \]

- NNLO virtual contribution with adaptive integrand decomposition

  [Ossola, Mastrolia, Peraro, Primo, Schubert, W.J.T. (work in progress)]
AIDA for muon-electron scattering

Recent proposal for the determination of the hadronic contribution to the muon from the measurement of muon-electron scattering $g - 2$

[Carloni Calame, Passera, Trentadue, Venanzoni (2015)]
[Abbiendi, Carloni Calame, Marconi et al (2017)]

In the massless electron limit, 4-point process depending on 3 scales

\[
s = (p_1 + p_2)^2 \quad t = (p_2 + p_3)^2 \quad m_e^2 \simeq 0 \quad u = -s - t + 2m^2
\]

\[e(p_1) + \mu(p_4) \to e(-p_2) + \mu(-p_3)\]

NNLO virtual contribution with adaptive integrand decomposition

[Ossola, Mastrolia, Peraro, Primo, Schubert, W.J.T. (work in progress)]
**AIDA for muon-electron scattering**

[**Initialisation**]

Identify parent topologies from Feynman graphs

**e.g. 1 Loop**

![Feynman diagrams](image-url)
**Initialisation**

Identify parent topologies from Feynman graphs

- e.g. 1 Loop

Generate all cuts and analyse their kinematics
AIDA for muon-electron scattering

[Mastrolia, Peraro, Primo, W.J.T. (work in progress)]

Initialisation

- Identify parent topologies from Feynman graphs
  
  e.g. 1 Loop

- Generate all cuts and analyse their kinematics

- Define adaptive variables and prepare substitution rules for all cuts

\[
X_{1,(1,2,3)} \rightarrow \frac{s^2d[1] - 2d[2]d[3]}{-4m^2s}
\]

\[
X_{2,(1,2,3)} \rightarrow 2m^2s^22m^2d[1]s \frac{d[1]s[2]d[2]2m^2d[3]}{4m^2s} s
\]

\[
\lambda^2_{(1,2,3)} \rightarrow \frac{m^2s^22m^2s d[1]m^2d[1]^2s^2d[2]s[2]d[2]2m^2d[3]}{s(-4m^2+s)}
\]
AIDA for muon-electron scattering

[Work in progress: Mastrolia, Peraro, Primo, W.J.T.]

**Job structure**

Group diagrams belonging to the same parent topology

*Example: 1 Loop*
**AIDA for muon-electron scattering**

[Mastrolia, Peraro, Primo, W.J.T. (work in progress)]

**Job structure**

- Group diagrams belonging to the same parent topology
  - e.g. 1 Loop

- Organise all cuts of the parent topology in **Jobs**
**AIDA for muon-electron scattering**

**Divide - Integrate - Divide**

- For every Job, **build the numerators** of the corresponding cuts

\[
\begin{align*}
\text{Num} & \left[ \{1, 3\}, \{1, 0, 1, 0\} \right] \\
\text{Num} & \left[ \{1, 3\}, \{1, 0, 1, 0\} \right] \\
\text{Num} & \left[ \{1, 3\}, \{1, 0, 1, 0\} \right] \\
\text{Num} & \left[ \{1, 3\}, \{1, 0, 1, 0\} \right] \\
\text{Num} & \left[ \{1, 3\}, \{1, 0, 1, 0\} \right]
\end{align*}
\]

- **Apply substitution rules** to the numerator

\[
\begin{align*}
X_{1,\{1,3\}} & \rightarrow \frac{s+d[1]-d[3]}{2s} \\
\end{align*}
\]

- **Collect powers of denominators** to read off residue and numerators of lower cuts
- **Integrate** (substitute) transverse vars appearing in the residues
- **Division** again, using as input numerators the residues!
AIDA for muon-electron scattering

[Mastrolia, Peraro, Primo, W.J.T. (work in progress)]

Input numerators
AIDA for muon-electron scattering

[ Mastrolia, Peraro, Primo, W.J.T. (work in progress) ]

Input numerators

\[
\begin{align*}
\beta\left( (1, 2, 3, 4), (1, 1, 1, 1) \right) & = \frac{2}{x^2} \left( 16 \left( s^2 + 2t + 3\right) \right) \\
\beta\left( (2, 3, 4), (0, 1, 1, 1) \right) & = \frac{2}{x^2} \left( d + g \right) \\
\beta\left( (1, 3, 4), (1, 0, 1, 1) \right) & = \frac{2}{x^2} \\
\beta\left( (2, 3), (1, 1, 1, 0) \right) & = \frac{2}{x^2} \left( s^2 - 16 + 7 \right) \\
\beta\left( (3, 4), (0, 0, 1, 1) \right) & = \frac{8}{x^2} \left( 2s^2 + 2t + 4s \right) \\
\beta\left( (2, 4), (1, 1, 1, 0) \right) & = \frac{4}{x^2} \left( 2s^2 + 2t + 4s \right) \\
\beta\left( (2, 3), (0, 1, 1, 0) \right) & = \frac{4}{x^2} \left( 2s^2 + 2t + 4s \right) \\
\beta\left( (1, 4), (1, 0, 1, 0) \right) & = \frac{8}{x^2} \left( 2s^2 + 2t + 4s \right) \\
\beta\left( (1, 3), (1, 0, 1, 0) \right) & = \frac{8}{x^2} \left( 2s^2 + 2t + 4s \right) \\
\beta\left( (1, 2), (1, 1, 1, 0) \right) & = \frac{8}{x^2} \left( 2s^2 + 2t + 4s \right) \\
\beta\left( (1), (0, 0, 1, 0) \right) & = \frac{8}{x^2} \left( 2s^2 + 2t + 4s \right) \\
\beta\left( (1), (0, 1, 1, 0) \right) & = \frac{8}{x^2} \left( 2s^2 + 2t + 4s \right)
\end{align*}
\]
AIDA for muon-electron scattering

[Mastrolia, Peraro, Primo, W.J.T. (work in progress)]
AIDA for muon-electron scattering

[Mastrolia, Peraro, Primo, W.J.T. (work in progress)]

@ 1 Loop:: same result as TID
AIDA for muon-electron scattering

Two-loop features

- Different treatment for Factorised and non factorised topologies

- Independent parametrisation for each loop

- Squared propagators affect Jobs organisation

[Work in progress: Mastrolia, Peraro, Primo, W.J.T.]
AIDA for muon-electron scattering

[69 Feynman diagrams identified]

10 genuine 2 loop 4-point functions appear
Two-loop preliminary results

[Mastrolia, Peraro, Primo, W.J.T. (work in progress)]

AIDA for muon-electron scattering

69 Feynman diagrams identified
10 genuine 2 loop 4-point functions appear
AIDA for muon-electron scattering

[Mastrolia, Peraro, Primo, W.J.T. (work in progress)]

69 Feynman diagrams identified
10 genuine 2 loop 4-point functions appear
AIDA for muon-electron scattering

[Mastrolia, Peraro, Primo, W.J.T. (work in progress)]

Input: rank 4 numerator with 289 monomials
AIDA for muon-electron scattering

[Mastrolia, Peraro, Primo, W.J.T. (work in progress)]

Two-loop preliminary results

AIDA for muon-electron scattering

69 Feynman diagrams identified
10 genuine 2 loop 4-point functions appear

Input:
rank 4 numerator with 289 monomials

Output:
135 contributions

Reduction time ~ 1.5 mins
Preliminary analytic results

\[ \text{gg} \rightarrow \text{H g} \]

Input: rank 5 numerator with 603 monomials
Reduction time ~ 10 mins
Output: 272 contributions

\[ \text{gg} \rightarrow \text{H H} \]

Input: rank 6 numerator with 574 monomials
Reduction time ~ 8 mins
Output: 269 contributions
Towards 2→3 processes

\[ gg \rightarrow H g g \] (Numerical evaluation)

5-point process depending on 7 scales

\[
\begin{align*}
    s_{12} &= (p_1 + p_2)^2 \\
    s_{45} &= (p_4 + p_5)^2 \\
    s_{23} &= (p_2 + p_3)^2 \\
    s_{51} &= (p_5 + p_1)^2 \\
    s_{34} &= (p_3 + p_4)^2 \\
    p_s^2 &= m_H^2 \\
    m_t^2 &= m_t^2
\end{align*}
\]

+ dimension \( d \)

Numerical evaluation for all kin. vars \( \rightarrow \) Retain \( d \)-dependence

Still many things to improve...

Input: rank 6 numerator with 1250 monomials
Reduction time \( \sim \) 1 min
Output: 1008 contributions

Input: rank 6 numerator with 2747 monomials
Reduction time \( \sim \) 2 min
Output: 1169 contributions
Conclusions/Outlook

Colour-kinematics duality

- Further simplifications from Colour-Kinematics duality
- Most compact representation of the Jacobi identity for kinematic numerators
- New integral relations at one-loop level
- Provide integral relations at multi-loop level
- Interplay the use of the C/K and LT duality

Multi-loop scattering amplitudes

- Integrand decomposition methods —@1 and 2 Loops Automated (AIDA)
- Analytic decomposition for all 2→2 processes—Under control
- AIDA’s output — Apply IBPs + evaluation of MIs
- Muon-electron scattering at NNLO is at hand
- Deal with analytic expressions for 2→3,4 processes
- More processes to come in the near future
Conclusions/Outlook

Colour-kinematics duality

- Further simplifications from Colour-Kinematics duality
- Most compact representation of the Jacobi identity for kinematic numerators
- New integral relations at one-loop level
- Provide integral relations at multi-loop level
- Interplay the use of the C/K and LT duality

Multi-loop scattering amplitudes

- Integrand decomposition methods $\rightarrow$ @1 and 2 Loops Automated (AIDA)
- Analytic decomposition for all 2$\rightarrow$2 processes $\rightarrow$ Under control
- AIDA’s output $\rightarrow$ Apply IBPs + evaluation of MIs
- Muon-electron scattering at NNLO is at hand

- Deal with analytic expressions for 2$\rightarrow$3,4 processes
- More processes to come in the near future
Extra slides
Two-loop MIs for muon-electron scattering

Planar integrals :: Family 1 (34 MIs)
[Bonciani, Ferroglia, Gehrmann, Maitre, von Manteuffel, Studerus]

Planar integrals :: Family 2 (42 MIs)
[Mastrolia, Passera, Primo, Schubert (2017)]

Non planar integrals :: Family 3 (44 MIs)
[Mastrolia, Primo, Schubert (work in progress)]