Simplification of expressions containing classical polylogarithms and Chen's iterated integrals.

Lee Roman
BLTP, JINR, Dubna, October 23, 2024

Budker Institute of Nuclear Physics

- Modern and planned high-energy physics experiments promise to provide a lot of high-precision experimental data. The high precision is especially important in the context of searches of deviations from Standard Model predictions — the New Physics.
- Consequently, the theoretical predictions should also have high precision, which in practice means going beyond NLO approximation. Fortunately, the multiloop calculational methods have evolved enough to provide this precision (with some reservations). Among the most important calculations are those of NNLO corrections to differential cross sections of processes involving massive particles.
- However, already at NNLO level, the final results often have a very cumbersome form, which may complicate their practical use in experimental data processing. As the complexity explosively grows with increasing of the number of loops, the problem of simplification should not be underestimated.

Iterated integrals, why do they appear in multiloop calculations.

Standard calculation path

1. Diagram generation ✓

Generate diagrams contributing to the chosen order of perturbation theory.

Tools: `qgraf` [Nogueira, 1993], `FeynArts` [Hahn, 2001], `tapir` [Gerlach et al., 2022],...

2. IBP reduction

Setup IBP reduction, derive differential system for master integrals.

Tools: `FIRE6` [Smirnov and Chuharev, 2020], `Kira2` [Klappert et al., 2021], `LiteRed` [RL, 2012], `NeatIBP` [Wu et al., 2024], ...

3. DE Solution

Reduce the system to ϵ-form, write down solution in terms of polylogarithms. Fix boundary conditions by auxiliary methods.

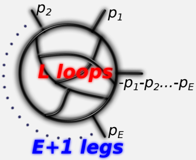
Tools: `Fuchsia` [Gituliar and Magerya, 2017], `epsilon` [Prausa, 2017], `Libra` [RL, 2021]

IBP identities

Given a Feynman diagram, consider a family

$$j(\boldsymbol{n}) = j(n_1, \ldots, n_N) = \int d\mu_L \boldsymbol{D}^{-\boldsymbol{n}} = \int \prod_{i=1}^{L} d^d l_i \prod_{k=1}^{N} D_k^{-n_k},$$

$D_1, \ldots, D_M$ — denominators of the diagram, $D_{M+1}, \ldots, D_N$ — irreducible numerators.

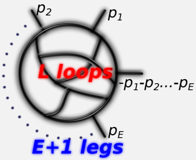


IBP identities

Given a Feynman diagram, consider a family

$$j(\boldsymbol{n}) = j(n_1, \ldots, n_N) = \int d\mu_L \boldsymbol{D}^{-\boldsymbol{n}} = \int \prod_{i=1}^{L} d^d l_i \prod_{k=1}^{N} D_k^{-n_k},$$

$D_1, \ldots, D_M$ — denominators of the diagram,
$D_{M+1}, \ldots, D_N$ — irreducible numerators.



IBP identities [Chetyrkin and Tkachov, 1981]

In dim. reg. the integral of divergence is zero (no surface terms):

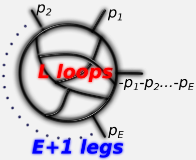
$$0 = \int d\mu_L \frac{\partial}{\partial l_i} \cdot q_j \boldsymbol{D}^{-\boldsymbol{n}} = \sum_s c_s(\boldsymbol{n}) j(\boldsymbol{n} + \delta_s).$$

Explicitly differentiating, we obtain relations between integrals.

Given a Feynman diagram, consider a family

$$j(\boldsymbol{n}) = j(n_1,\ldots,n_N) = \int d\mu_L \boldsymbol{D}^{-\boldsymbol{n}} = \int \prod_{i=1}^{L} d^d l_i \prod_{k=1}^{N} D_k^{-n_k},$$

$D_1,\ldots,D_M$ — denominators of the diagram, $D_{M+1},\ldots,D_N$ — irreducible numerators.



IBP identities [Chetyrkin and Tkachov, 1981]

In dim. reg. the integral of divergence is zero (no surface terms):

$$0 = \int d\mu_L \frac{\partial}{\partial l_i} \cdot q_j \boldsymbol{D}^{-\boldsymbol{n}} = \sum_s c_s(\boldsymbol{n}) j(\boldsymbol{n}+\delta_s).$$

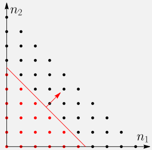
Explicitly differentiating, we obtain relations between integrals.

More recent ideas [RL, 2014; Yang Zhang, 2014]

IBP identities in Lee-Pomeransky and Baikov representations : approach based on calculating syzygies. NB: parametric IBPs work also for non-standard setup.

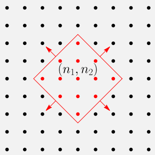
Laporta algorithm (FIRE, Kira, Reduze, NeatIBP ...)

- generate identities for many numeric $\boldsymbol{n} \in \mathbb{Z}^N$.
- use Gauss elimination and collect reduction rules to database.
- twist: mapping to finite fields $\mathbb{F}_p$ + reconstruction. $\Longleftarrow$ naturally parallelizable



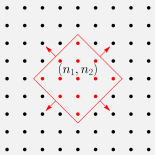
Heuristic search (`LiteRed`)

1. Generate identities for shifts around $\boldsymbol{n}$ with *symbolic* entries.
2. Use Gauss elimination until proper rule is found.
3. **Solve Diophantine equations to derive applicability condition.**



Heuristic search (LiteRed)

1. Generate identities for shifts around $\boldsymbol{n}$ with *symbolic* entries.
2. Use Gauss elimination until proper rule is found.
3. **Solve Diophantine equations to derive applicability condition.**

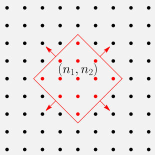


As a result of IBP reduction we express amplitudes via a finite set of master integrals $\boxed{\boldsymbol{j} = (j_1, \ldots, j_K)^\intercal}$.

IBP reduction and differential equations

Heuristic search (LiteRed)

1. Generate identities for shifts around $\boldsymbol{n}$ with *symbolic* entries.
2. Use Gauss elimination until proper rule is found.
3. **Solve Diophantine equations to derive applicability condition.**



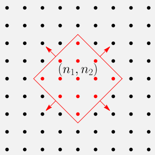
As a result of IBP reduction we express amplitudes via a finite set of master integrals $\boxed{\boldsymbol{j} = (j_1, \ldots, j_K)^\intercal}$.

What is even more important, using IBP reduction we can obtain differential equations for the master integrals:

$$\partial_x \boldsymbol{j} = M(x, d)\boldsymbol{j}$$

Heuristic search (`LiteRed`)

1. Generate identities for shifts around $\boldsymbol{n}$ with *symbolic* entries.
2. Use Gauss elimination until proper rule is found.
3. **Solve Diophantine equations to derive applicability condition.**



As a result of IBP reduction we express amplitudes via a finite set of master integrals $\boxed{\boldsymbol{j} = (j_1, \ldots, j_K)^\intercal}$.

What is even more important, using IBP reduction we can obtain differential equations for the master integrals:

$$\partial_x \boldsymbol{j} = M(x, d)\boldsymbol{j}$$

It is often easier to solve these equations rather than to use direct methods for calculation of the master integrals.

- For several kinematic variables we have the corresponding number of differential systems:

$$\frac{\partial}{\partial x_i} \boldsymbol{j} = M_i(\boldsymbol{x}, d) \boldsymbol{j}$$

Here $M_i(\boldsymbol{x}, d)$ are matrices of rational functions of $\boldsymbol{x}$ and d.

Differential equations for several kinematic variables

- For several kinematic variables we have the corresponding number of differential systems:

$$\frac{\partial}{\partial x_i}\boldsymbol{j} = M_i(\boldsymbol{x}, d)\boldsymbol{j}$$

  Here $M_i(\boldsymbol{x}, d)$ are matrices of rational functions of $\boldsymbol{x}$ and d.

- Integrability conditions — flatness of the connection $\nabla_i = \frac{\partial}{\partial x_i} - M_i$:

$$[\nabla_i, \nabla_j] = 0.$$

  NB: Introducing differential 1-form $\mathcal{M} = M_i dx_i$ we can write the integrability condition as $\boxed{d\mathcal{M} - \mathcal{M} \wedge \mathcal{M} = 0}$.

Differential equations for several kinematic variables

- For several kinematic variables we have the corresponding number of differential systems:

$$\frac{\partial}{\partial x_i}\boldsymbol{j} = M_i(\boldsymbol{x}, d)\boldsymbol{j}$$

  Here $M_i(\boldsymbol{x}, d)$ are matrices of rational functions of $\boldsymbol{x}$ and d.

- Integrability conditions — flatness of the connection $\nabla_i = \frac{\partial}{\partial x_i} - M_i$:

$$[\nabla_i, \nabla_j] = 0.$$

  NB: Introducing differential 1-form $\mathcal{M} = M_i dx_i$ we can write the integrability condition as $\boxed{d\mathcal{M} - \mathcal{M} \wedge \mathcal{M} = 0}$.

- The general solution (or evolution operator) is expressed as path-ordered exponent

$$U(\boldsymbol{x}, \boldsymbol{x}_0) = \mathrm{Pexp}\left[\int_C d\boldsymbol{x}' \cdot \boldsymbol{M}(\boldsymbol{x}', d)\right] = \mathrm{Pexp}\left[\int_C \mathcal{M}(\boldsymbol{x}', d)\right],$$

  where $C = C(\boldsymbol{x}_0, \boldsymbol{x})$ denotes a path connecting $\boldsymbol{x}_0$ and $\boldsymbol{x}$.

- For several kinematic variables we have the corresponding number of differential systems:

$$\frac{\partial}{\partial x_i}\boldsymbol{j} = M_i(\boldsymbol{x}, d)\boldsymbol{j}$$

  Here $M_i(\boldsymbol{x}, d)$ are matrices of rational functions of $\boldsymbol{x}$ and d.

- Integrability conditions — flatness of the connection $\nabla_i = \frac{\partial}{\partial x_i} - M_i$:

$$[\nabla_i, \nabla_j] = 0.$$

  NB: Introducing differential 1-form $\mathcal{M} = M_i dx_i$ we can write the integrability condition as $\boxed{d\mathcal{M} - \mathcal{M} \wedge \mathcal{M} = 0}$.

- The general solution (or evolution operator) is expressed as path-ordered exponent

$$U(\boldsymbol{x}, \boldsymbol{x}_0) = \mathrm{Pexp}\left[\int_C d\boldsymbol{x}' \cdot \boldsymbol{M}(\boldsymbol{x}', d)\right] = \mathrm{Pexp}\left[\int_C \mathcal{M}(\boldsymbol{x}', d)\right],$$

  where $C = C(\boldsymbol{x}_0, \boldsymbol{x})$ denotes a path connecting $\boldsymbol{x}_0$ and $\boldsymbol{x}$.

> Note that $U(\boldsymbol{x}, \boldsymbol{x}_0)$ is **path-independent**, i.e., does not change upon deformations of the path $C(\boldsymbol{x}_0, \boldsymbol{x})$ provided they retain the end points $\boldsymbol{x}_0$ and $\boldsymbol{x}$ and do not cross singularities of $\boldsymbol{M}(\boldsymbol{x}, d)$.

- [Henn, 2013]: It is often possible to find a canonical basis $\boldsymbol{J} = T^{-1}\boldsymbol{j}$ such that

$$\partial_i \boldsymbol{J} = \epsilon S_i(\boldsymbol{x})\boldsymbol{J}$$

  Here $\epsilon = 2 - d/2$ is the parameter of dimensional regularization, $\boldsymbol{S}(\boldsymbol{x})$ is Fuchsian, i.e., has no multiple poles and falls of at infinity. [RL, 2015]: the algorithm of finding the transformation to ϵ-form for a given differential system.

- [Henn, 2013]: It is often possible to find a canonical basis $\boldsymbol{J} = T^{-1}\boldsymbol{j}$ such that

$$\partial_i \boldsymbol{J} = \epsilon S_i(\boldsymbol{x})\boldsymbol{J}$$

  Here $\epsilon = 2 - d/2$ is the parameter of dimensional regularization, $\boldsymbol{S}(\boldsymbol{x})$ is Fuchsian, i.e., has no multiple poles and falls of at infinity. [RL, 2015]: the algorithm of finding the transformation to ϵ-form for a given differential system.

- The path-ordered exponent can be expanded in perturbative series in ϵ:

$$U(\boldsymbol{x}, \boldsymbol{x}_0) = \mathrm{Pexp}\left[\epsilon \int_C \mathcal{S}(\boldsymbol{x}')\right] = \sum_n \epsilon^n \int\!\!\cdots\!\!\int_{\boldsymbol{x} \underset{C}{>} \boldsymbol{x}_n \underset{C}{>} \ldots \underset{C}{>} \boldsymbol{x}_0} \mathcal{S}(\boldsymbol{x}_n) \ldots \mathcal{S}(\boldsymbol{x}_1), \qquad \boxed{\mathcal{S}(\boldsymbol{x}) = d\boldsymbol{x} \cdot \boldsymbol{S}(x)}$$

- [Henn, 2013]: It is often possible to find a canonical basis $\boldsymbol{J} = T^{-1}\boldsymbol{j}$ such that

$$\partial_i \boldsymbol{J} = \epsilon S_i(\boldsymbol{x})\boldsymbol{J}$$

  Here $\epsilon = 2 - d/2$ is the parameter of dimensional regularization, $\boldsymbol{S}(\boldsymbol{x})$ is Fuchsian, i.e., has no multiple poles and falls of at infinity. [RL, 2015]: the algorithm of finding the transformation to ϵ-form for a given differential system.

- The path-ordered exponent can be expanded in perturbative series in ϵ:

$$U(\boldsymbol{x}, \boldsymbol{x}_0) = \mathrm{Pexp}\left[\epsilon \int_C \mathcal{S}(\boldsymbol{x}')\right] = \sum_n \epsilon^n \int\limits_{\boldsymbol{x} \underset{C}{>} \boldsymbol{x}_n \underset{C}{>} \ldots \underset{C}{>} \boldsymbol{x}_0} \!\!\cdots\!\! \int \mathcal{S}(\boldsymbol{x}_n) \ldots \mathcal{S}(\boldsymbol{x}_1), \qquad \boxed{\mathcal{S}(\boldsymbol{x}) = d\boldsymbol{x} \cdot \boldsymbol{S}(x)}$$

Chen's iterated path integrals

$$\mathcal{I}_C(\omega_n, \ldots, \omega_1) = \int\limits_{\boldsymbol{x} \underset{C}{>} \boldsymbol{x}_n \underset{C}{>} \ldots \underset{C}{>} \boldsymbol{x}_0} \!\!\cdots\!\! \int \omega_n(\boldsymbol{x}_n) \ldots \omega_1(\boldsymbol{x}_1)$$

where $\omega_k(\boldsymbol{x}_k) = d\boldsymbol{x}_k \cdot \boldsymbol{f}_k(\boldsymbol{x}_k)$ are some differential 1-forms. Note that the integrability condition now implies $d\mathcal{S} = 0$ and therefore we have that $d\omega_k = 0$.

- **Goncharov's polylogarithms** are 1-dimensional cousins of $\mathcal{I}_C$. They are conveniently defined recursively:

$$G(a_n, a_{n-1}, \ldots, a_1|x) = \int_0^x \frac{dx_n}{x - a_n} G(a_{n-1}, \ldots, a_1|x) \text{ and } G(\underbrace{0, \ldots, 0}_{n}|x) = \frac{\ln^n x}{n!}$$

  If $a_1 \neq 0$, they are related to 1-dimensional $\mathcal{I}_C$ via

$$\boxed{G(a_n, a_{n-1}, \ldots, a_1|x) = \mathcal{I}_C(d\ln(x - a_n), \ldots, d\ln(x - a_1)) \text{ with } C = C(x, 0).}$$

- Classical polylogarithms Li_n are expressed via G as $Li_n(x) = -G(\underbrace{0, \ldots, 0, 1}_{n}|x)$.

  Moreover, generic G with up to three indices can be expressed via Li_n with $n = 1, 2, 3$.

- NNLO results are often expressible via classical polylogarithms.

Simplification of classical polylogarithms

There is a standard approach to the simplification of the polylogarithmic expressions using symbol map. One might think of symbols as a cleaner way to represent iterated (or path-ordered) integrals with logarithmic weights (with some reservations, though):

$$I = \int\limits_{1>\tau_n>\ldots>\tau_1>0}\!\!\cdots\!\!\int d\ln p_n(\tau_n)\ldots d\ln p_1(\tau_1) \xrightarrow{\quad\mathcal{S}\quad} p_n\otimes\ldots\otimes p_1$$

Formal symbol manipulation rules then easily follow, e.g.

$$d\ln(pq) = d\ln p + d\ln q \qquad \Longrightarrow \qquad (\ldots\otimes pq\otimes\ldots) = (\ldots\otimes p\otimes\ldots) + (\ldots\otimes q\otimes\ldots)$$

Similarly, by ordering the integration variables in the product of integrals, we get $\mathcal{S}(I_1 I_2) = \mathcal{S}(I_1) \sqcup\!\sqcup\, \mathcal{S}(I_2)$, where $\sqcup\!\sqcup$ denotes a shuffle product, e.g.

$$(a\otimes b)\sqcup\!\sqcup(c\otimes d) = a\otimes b\otimes c\otimes d + a\otimes c\otimes b\otimes d + a\otimes c\otimes d\otimes b + c\otimes a\otimes b\otimes d + c\otimes a\otimes d\otimes b + c\otimes d\otimes a\otimes b$$

We have, in particular, symbols for classical polylogarithms

$$\mathcal{S}(\mathrm{Li}_n(x)) = -[\underbrace{x\otimes\ldots\otimes x}_{n-1}\otimes(x-1)]$$

Symbols are good for checking the identities, e.g., using $\mathcal{S}$ it is easy to establish

$$7\mathrm{Li}_2\left(\frac{1+\varepsilon/z}{1-i\varepsilon}\right)-7\mathrm{Li}_2\left(\frac{1+\bar{\varepsilon}/z}{1+i\bar{\varepsilon}}\right)+7\mathrm{Li}_2\left(\frac{z+\bar{\varepsilon}}{\bar{\varepsilon}-i}\right)-7\mathrm{Li}_2\left(\frac{z+\varepsilon}{\varepsilon+i}\right)+11\mathrm{Li}_2\left(\frac{z+\varepsilon}{\varepsilon-i}\right)-11\mathrm{Li}_2\left(\frac{z+\bar{\varepsilon}}{\bar{\varepsilon}+i}\right)$$
$$+4\mathrm{Li}_2(1+z\varepsilon)-4\mathrm{Li}_2(1+z\bar{\varepsilon})+18\mathrm{Li}_2(-iz)-18\mathrm{Li}_2(iz)+11\mathrm{Li}_2\left(\frac{1+\bar{\varepsilon}/z}{1-i\bar{\varepsilon}}\right)-11\mathrm{Li}_2\left(\frac{1+\varepsilon/z}{1+i\varepsilon}\right)$$
$$=\frac{2i\pi^2}{5\sqrt{3}}-\frac{23}{3}i\pi\ln z+6i\pi\ln\left(2-\sqrt{3}\right)-\frac{i\psi'\left(\frac{1}{6}\right)}{5\sqrt{3}}-24iG,$$

where $\varepsilon=1/\bar{\varepsilon}=e^{2\pi i/3}$ and $G=\sum_n\frac{(-1)^n}{(2n+1)^2}$ is Catalan constant.

But how can we construct a basis of Li_n functions which might enter the simplified expression?

Simplifications with symbol map

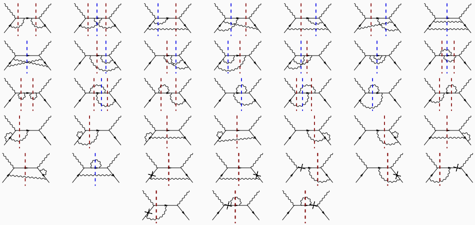
Symbols are good for checking the identities, e.g., using $\mathcal{S}$ it is easy to establish

$$7\mathrm{Li}_2\left(\frac{1+\varepsilon/z}{1-i\varepsilon}\right)-7\mathrm{Li}_2\left(\frac{1+\bar\varepsilon/z}{1+i\bar\varepsilon}\right)+7\mathrm{Li}_2\left(\frac{z+\bar\varepsilon}{\bar\varepsilon-i}\right)-7\mathrm{Li}_2\left(\frac{z+\varepsilon}{\varepsilon+i}\right)+11\mathrm{Li}_2\left(\frac{z+\varepsilon}{\varepsilon-i}\right)-11\mathrm{Li}_2\left(\frac{z+\bar\varepsilon}{\bar\varepsilon+i}\right)$$
$$+4\mathrm{Li}_2(1+z\varepsilon)-4\mathrm{Li}_2(1+z\bar\varepsilon)+18\mathrm{Li}_2(-iz)-18\mathrm{Li}_2(iz)+11\mathrm{Li}_2\left(\frac{1+\bar\varepsilon/z}{1-i\bar\varepsilon}\right)-11\mathrm{Li}_2\left(\frac{1+\varepsilon/z}{1+i\varepsilon}\right)$$
$$=\frac{2i\pi^2}{5\sqrt{3}}-\frac{23}{3}i\pi\ln z+6i\pi\ln\left(2-\sqrt{3}\right)-\frac{i\psi'\left(\frac{1}{6}\right)}{5\sqrt{3}}-24iG,$$

where $\varepsilon=1/\bar\varepsilon=e^{2\pi i/3}$ and $G=\sum_n\frac{(-1)^n}{(2n+1)^2}$ is Catalan constant.

But how can we construct a basis of Li_n functions which might enter the simplified expression?

NB: This identity and more complicated ones involving Li_3 functions was used in real life for the simplification of the total cross section of Compton scattering @NLO [RL et al., 2021].



Suppose that branching points (or, in multivariate setup, branching hypersurfaces) of original expression are determined by polynomial equations[1]

$$\bigvee_{k=1}^{K} p_k(x) = 0, \qquad (*)$$

where $p_k(x)$ are some irreducible polynomials. Then the simplified expression should also have the same set branching points.

[1] For multiloop calculations the polynomials are known in advance: they are the denominators of $S(x)$

Suppose that branching points (or, in multivariate setup, branching hypersurfaces) of original expression are determined by polynomial equations[1]

$$\bigvee_{k=1}^{K} p_k(x) = 0, \qquad (*)$$

where $p_k(x)$ are some irreducible polynomials. Then the simplified expression should also have the same set branching points.

In order to construct all possible arguments of Li_n, we need to recall the position of branching points Li_n function. Those are $\{0, 1, \infty\}$. Then the valid argument of Li_n should be a rational function $N(x)/D(x)$, such that the solutions of any of the three equations

$$N(x)/D(x) = 0, \quad N(x)/D(x) = \infty, \quad N(x)/D(x) = 1$$

belong to the branching set determined by $(*)$.

[1] For multiloop calculations the polynomials are known in advance: they are the denominators of $S(x)$

Suppose that branching points (or, in multivariate setup, branching hypersurfaces) of original expression are determined by polynomial equations[1]

$$\bigvee_{k=1}^{K} p_k(x) = 0, \qquad (*)$$

where $p_k(x)$ are some irreducible polynomials. Then the simplified expression should also have the same set branching points.

In order to construct all possible arguments of Li_n, we need to recall the position of branching points Li_n function. Those are $\{0, 1, \infty\}$. Then the valid argument of Li_n should be a rational function $N(x)/D(x)$, such that the solutions of any of the three equations

$$N(x)/D(x) = 0, \quad N(x)/D(x) = \infty, \quad N(x)/D(x) = 1$$

belong to the branching set determined by $(*)$.
Note that the three equations can be rewritten as

$$N(x) = 0, \quad D(x) = 0, \quad N(x) - D(x) = 0$$

and then our requirement leads to

$$N(x) \propto \prod_{k=1}^{K} p_k^{n_k}, \quad D(x) \propto \prod_{k=1}^{K} p_k^{d_k}, \quad N(x) - D(x) \propto \prod_{k=1}^{K} p_k^{n_k}, \qquad (n_k, d_k, m_k \in \mathbb{Z}_{\geqslant 0})$$

[1] For multiloop calculations the polynomials are known in advance: they are the denominators of $S(x)$

1. Generate polynomials $\{P_0, P_1, \dots P_N\} = \{1, p_1, \dots p_k, p_1^2,\ p_1 p_2, \dots p_1^{i_1} p_2^{i_2} \dots, p_K^{i_K}\}$ up to some sufficiently high power.
2. Search in the above set for linearly dependent triplets $\{P_i, P_j, P_k\}$, such that $a_i P_i + a_j P_j + a_k P_k = 0$, where a_i, a_j, a_k are coefficients independent of x.
3. Then each triplet gives rise to the following 6 possible Li_n functions:

$$\mathrm{Li}_n\left(-\frac{a_i P_i}{a_j P_j}\right), \mathrm{Li}_n\left(-\frac{a_i P_i}{a_k P_k}\right), \mathrm{Li}_n\left(-\frac{a_j P_j}{a_k P_k}\right),$$
$$\mathrm{Li}_n\left(-\frac{a_j P_j}{a_i P_i}\right), \mathrm{Li}_n\left(-\frac{a_k P_k}{a_i P_i}\right), \mathrm{Li}_n\left(-\frac{a_k P_k}{a_j P_j}\right).$$

   Of course, these 6 arguments are related by the group of Moebius transformations stabilizing the $\{0, 1, \infty\}$ set:

$$z \to z,\ 1-z,\ 1/z,\ 1-1/z,\ 1/(1-z),\ z/(z-1).$$

Let us take

$$\{p_1, \ldots, p_5\} = \{x, y, \hat{x}, \hat{y}, \hat{xy}\}, \qquad \text{where } \widehat{a} = 1 - a.$$

Then applying the above algorithm, we find $30 = 6 * 5$ valid arguments of Li_n functions. Using symbol map, we find relation for Li_2 functions:

5-term relation for dilogs

$$f(xy) + f\left(\frac{x\widehat{y}}{\widehat{xy}}\right) + f\left(\frac{y\widehat{x}}{\widehat{xy}}\right) - f(x) - f(y) = 0$$

where

$$f(x) = \mathrm{Li}_2(x) + \frac{1}{2}\ln(1-x)\ln x.$$

This identity was found by W.Spence in 1809.

Let us now take

$$\{p_1, \ldots, p_{10}\} = \{x, y, z, \hat{x}, \hat{y}, \hat{z}, \widehat{xy}, \widehat{xz}, \widehat{yz}, \widehat{xyz}\}$$

Then applying the above algorithm, we find $132 = 6 * 22$ valid arguments of Li_n functions. Using symbol map, we find nontrivial relation for Li_3 functions:

22-term relation for Li_3

$$f(xyz) + 3f\left(\frac{\widehat{x}}{\widehat{xyz}}\right) + 3f\left(\frac{xy\widehat{z}}{\widehat{xyz}}\right) - 3f\left(\frac{-x\widehat{y}\widehat{z}}{\widehat{x}\widehat{xyz}}\right) + 6f\left(\frac{-x\widehat{y}}{\widehat{x}}\right)$$

$$-3f(xy) + 3f(x) + \frac{3}{2}\pi^2 \ln x - 3\zeta_3 + \text{permutations} = 0, \qquad x, y, z \in (0,1)$$

where $\widehat{a} = 1 - a$ and

$$f(x) = \mathrm{Li}_3(x) + \frac{1}{24}\ln(1-x)\ln^2\left(x^2\right) - \frac{\pi^2}{12}\ln\left(x^2\right).$$

This identity is probably equivalent to 22 term relation in [Goncharov, 1991].

Chen’s iterated path integrals via Goncharov’s polylogarithms

Path dependence of $\mathcal{I}_C$

1. **Why it is important to care about the path?** Because we finally want to express $\mathcal{I}_C$ via Goncharov's polylogs — the one-dimensional Chen's iterated integrals with weights $\omega_k = d\log(x - a_k)$. It means that we have to choose path and its parametrization so as to rationalize the weights.

1. **Why it is important to care about the path?** Because we finally want to express $\mathcal{I}_C$ via Goncharov's polylogs — the one-dimensional Chen's iterated integrals with weights $\omega_k = d\log(x - a_k)$. It means that we have to choose path and its parametrization so as to rationalize the weights.
2. For example, if we have global rationalizing variables, we can choose a path consisting of a set of straight line segments on each only one variable is changing.

1. **Why it is important to care about the path?** Because we finally want to express $\mathcal{I}_C$ via Goncharov's polylogs — the one-dimensional Chen's iterated integrals with weights $\omega_k = d\log(x - a_k)$. It means that we have to choose path and its parametrization so as to rationalize the weights.
2. For example, if we have global rationalizing variables, we can choose a path consisting of a set of straight line segments on each only one variable is changing.
3. Path-ordered exponent $U(\boldsymbol{x}, \boldsymbol{x}_0)$ is "path-independent". So is its perturbative expansion. **But does it mean each individual $\mathcal{I}_C$ is also path independent?** No, it does not! Only some specific linear combinations are path-independent.

1. **Why it is important to care about the path?** Because we finally want to express $\mathcal{I}_C$ via Goncharov's polylogs — the one-dimensional Chen's iterated integrals with weights $\omega_k = d\log(x - a_k)$. It means that we have to choose path and its parametrization so as to rationalize the weights.
2. For example, if we have global rationalizing variables, we can choose a path consisting of a set of straight line segments on each only one variable is changing.
3. Path-ordered exponent $U(\boldsymbol{x}, \boldsymbol{x}_0)$ is "path-independent". So is its perturbative expansion. **But does it mean each individual $\mathcal{I}_C$ is also path independent?** No, it does not! Only some specific linear combinations are path-independent.
4. **Why we need path-independent combinations other than those which appear in pert. expansion of $U(\boldsymbol{x}, \boldsymbol{x}_0)$?** Because of the first issue: sometimes we need to choose different paths for different iterated integrals to express them via Goncharov's polylogs.

1-dim case

Let us first consider 1-dimensional case

$$\mathcal{I}_C(\omega_n(x), \ldots \omega_1(x))$$

We may vary the path on the complex plane of x. **Are 1-dim $\mathcal{I}_C$ path-independent?**

1-dim case

Let us first consider 1-dimensional case

$$\mathcal{I}_C(\omega_n(x), \ldots \omega_1(x))$$

We may vary the path on the complex plane of x. **Are 1-dim $\mathcal{I}_C$ path-independent?** Yes, they are.

Auxiliary differential system

$$d\boldsymbol{J} = \mathcal{M}\boldsymbol{J}, \qquad \text{with} \qquad \mathcal{M} = \begin{bmatrix} 0 & & & & \\ \omega_1 & \ddots & & & \\ & \omega_2 & \ddots & & \\ & & \ddots & \ddots & \\ & & & \omega_n & 0 \end{bmatrix}$$

Then we have U_{ik} is equal to $\mathcal{I}_C(\omega_{i-1}, \ldots \omega_k|x)$ for $i > k$, to 1 if $i = k$ and to 0 otherwise. In particular, $\boxed{\mathcal{I}_C(\omega_n, \ldots \omega_1) = \boldsymbol{U}_{n+1,1}}$, and we remember that U is path-independent!

Why the same approach does not work for several variables?

Let us first consider 1-dimensional case

$$\mathcal{I}_C(\omega_n(x), \ldots \omega_1(x))$$

We may vary the path on the complex plane of x. **Are 1-dim $\mathcal{I}_C$ path-independent?** Yes, they are.

Auxiliary differential system

$$d\boldsymbol{J} = \mathcal{M}\boldsymbol{J}, \qquad \text{with} \qquad \mathcal{M} = \begin{bmatrix} 0 & & & & \\ \omega_1 & \ddots & & & \\ & \omega_2 & \ddots & & \\ & & \ddots & \ddots & \\ & & & \omega_n & 0 \end{bmatrix}$$

Then we have U_{ik} is equal to $\mathcal{I}_C(\omega_{i-1}, \ldots \omega_k|x)$ for $i > k$, to 1 if $i = k$ and to 0 otherwise. In particular, $\boxed{\mathcal{I}_C(\omega_n, \ldots \omega_1) = U_{n+1,1}}$, and we remember that U is path-independent!

Why the same approach does not work for several variables?

$\boxed{d\mathcal{M} = 0}$ but $\boxed{\mathcal{M} \wedge \mathcal{M} \neq 0}$, so the connection is not flat and Pexp depends on the path.

- Which linear combinations of $\mathcal{I}_C$ are path-independent?
  Note that for one-fold $\mathcal{I}_C(\omega)$ the path-independence is equivalent to the requirement $d\omega = 0$ (which we automatically have for our setup).

[2] Note that despite the similarity this is not the same symbol map that we discussed earlier.

- Which linear combinations of $\mathcal{I}_C$ are path-independent?
  Note that for one-fold $\mathcal{I}_C(\omega)$ the path-independence is equivalent to the requirement $d\omega = 0$ (which we automatically have for our setup).
- Let us relate to each $\mathcal{I}_C$ the "symbol"[2]:

$$\mathcal{I}_C(\omega_n, \ldots, \omega_1) \xrightarrow{\mathcal{S}} \omega_n \otimes \ldots \otimes \omega_1$$

  and linearly extend the definition to linear combinations.

[2] Note that despite the similarity this is not the same symbol map that we discussed earlier.

Path-independent combinations

- Which linear combinations of $\mathcal{I}_C$ are path-independent?
  Note that for one-fold $\mathcal{I}_C(\omega)$ the path-independence is equivalent to the requirement $d\omega = 0$ (which we automatically have for our setup).
- Let us relate to each $\mathcal{I}_C$ the "symbol"[2]:

$$\mathcal{I}_C(\omega_n, \ldots, \omega_1) \xrightarrow{\mathcal{S}} \omega_n \otimes \ldots \otimes \omega_1$$

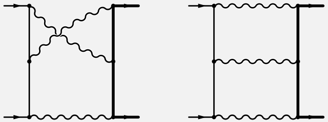
  and linearly extend the definition to linear combinations.
- Let us define the linear operator D (the "differential") acting as

$$D(\omega_n \otimes \ldots \otimes \omega_1) = \sum_{k=1}^{n-1} \omega_n \otimes \ldots \otimes \omega_k \wedge \omega_{k+1} \otimes \ldots \otimes \omega_1 + \sum_{k=1}^{n} \omega_n \otimes \ldots \otimes d\omega_k \otimes \ldots \otimes \omega_1$$

Path-independence criterion

$L = \sum_a c_a \mathcal{I}_C(\omega_a)$ is path-independent $\iff D(\mathcal{S}(L)) = 0$

[2] Note that despite the similarity this is not the same symbol map that we discussed earlier.



Physical variables: β — muon velocity in c.m.s., $c = \cos\theta$ — cosine of scattering angle.

There are 66 master integrals. In order to reduce the differential equation to ϵ-form, one has to pass to new variables ξ, χ via $\beta = \frac{2\xi}{\xi^2+1}$, $c = \frac{(\xi^2+1)\chi}{\xi^2\chi^2+1}$.

- The differential system in ϵ-form has the form $dJ = \epsilon \sum_{i=1}^{13} S_i d \ln w_i J$, $w_1, \ldots w_{11}$ are rational functions of β and c. But the last two weights w_{12} and w_{13} only become rational when passing to ξ, χ.
- In principle, we can pass to ξ, χ, but then the weights w_{8-11} become too complicated. E.g.

$$w_8 = \frac{1 - 2\beta c + \beta^2}{(1-\beta)^2(1-\beta c)} = \frac{\xi^6\chi^2 - 4\xi^5\chi + 6\xi^4\chi^2 + \xi^4 - 8\xi^3\chi + \xi^2\chi^2 + 6\xi^2 - 4\xi\chi + 1}{(1-\xi)^4(1-\xi\chi)^2}$$

Example: $e^+e^- \to \mu^+\mu^-$

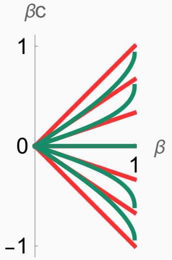
So we really want to stay with β and c where it is possible. The more so that only a few (out of almost 3000) iterated integrals in the final expression involve weights w_{12}, w_{13}:

$$\mathcal{I}_C(w_{12}), \mathcal{I}_C(w_{13}), \mathcal{I}_C(w_{12}, w_{12}, w_5), \mathcal{I}_C(w_{12}, w_{13}, w_5),$$
$$\mathcal{I}_C(w_{12}, w_{12}, w_5, w_5), \mathcal{I}_C(w_{12}, w_{13}, w_5, w_5)$$

Using the above mentioned technique we find, in particular, that

$$\mathcal{I}_C(w_{12}, w_{13}, w_5) - 4\mathcal{I}_C(w_4, w_1, w_2) + 2\mathcal{I}_C(w_6, w_2, w_5) + 2\mathcal{I}_C(w_6, w_5, w_2)$$

is path-independent. So, for this specific combination we can pass to ξ and χ —- note that there are no w_{8-11} weights in this combination.



- Each step towards increasing the # of loops and/or # of scales requires new methods. Those involve both technological advances and new algorithms coming from various fields of mathematics.
- Already at NNLO level the problem of simplification of the results becomes quite important.
- The basis of Li_n functions with a prescribed position of branching points can be found algorithmically.
- Symbol map $\mathcal{S}$ and $D\mathcal{S}$ can help in finding the identities and the path-independent combinations, respectively.
- However, the problem of simplification still remains heuristic to some extent. Maybe AI techniques can help here.

References

K. G. Chetyrkin and F. V. Tkachov. Integration by parts: The algorithm to calculate β-functions in 4 loops. *Nucl. Phys. B*, 192:159, 1981.

Marvin Gerlach, Florian Herren, and Martin Lang. `tapir` a tool for topologies, amplitudes, partial fraction decomposition and input for reductions. 1 2022.

Oleksandr Gituliar and Vitaly Magerya. Fuchsia: a tool for reducing differential equations for Feynman master integrals to epsilon form. *Comput. Phys. Commun.*, 219, 2017.

Alexander B Goncharov. Geometry of configurations, polylogarithms and motivic cohomology. 1991.

Thomas Hahn. Generating Feynman diagrams and amplitudes with FeynArts 3. *Comput. Phys. Commun.*, 140:418–431, 2001. doi: 10.1016/S0010-4655(01)00290-9.

Johannes M. Henn. Multiloop integrals in dimensional regularization made simple. *Phys.Rev.Lett.*, 110(25):251601, 2013. doi: 10.1103/PhysRevLett.110.251601.

Jonas Klappert, Fabian Lange, Philipp Maierhöfer, and Johann Usovitsch. Integral reduction with Kira 2.0 and finite field methods. *Comput. Phys. Commun.*, 266:108024, 2021. doi: 10.1016/j.cpc.2021.108024.

Paulo Nogueira. Automatic Feynman graph generation. *J. Comput. Phys.*, 105:279–289, 1993. doi: 10.1006/jcph.1993.1074.

Mario Prausa. epsilon: A tool to find a canonical basis of master integrals. *Comput. Phys. Commun.*, 219:361–376, 2017. doi: 10.1016/j.cpc.2017.05.026.

RL. Presenting litered: a tool for the loop integrals reduction, 2012.

RL. Modern techniques of multiloop calculations. In Etienne Augé and Jacques Dumarchez, editors, *Proceedings, 49th Rencontres de Moriond on QCD and High Energy Interactions*, pages 297–300, Paris, France, 2014. Moriond, Moriond.

RL. Reducing differential equations for multiloop master integrals. *J. High Energy Phys.*, 1504:108, 2015. doi: 10.1007/JHEP04(2015)108.

RL. Libra: A package for transformation of differential systems for multiloop integrals. *Computer Physics Communications*, 267:108058, 2021. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2021.108058. URL `https://www.sciencedirect.com/science/article/pii/S0010465521001703`.

RL, Schwartz, and Zhang. Compton Scattering Total Cross Section at Next-to-Leading Order. *Phys. Rev. Lett.*, 126(21):211801, 2021. doi: 10.1103/PhysRevLett.126.211801.