Semi- and Fully-Inclusive Phase-Space Integrals at Four Loops

Dissertation
zur Erlangung des Doktorgrades
an der Fakultät für Mathematik, Informatik und Naturwissenschaften,
Fachbereich Physik
der Universität Hamburg

vorgelegt von
Vitalii Maheria

Hamburg
2022

Dedicated to
Magerya Lubov Vitalievna
(1954–2019),
who always wanted her son to be scientist,
and through a lifetime of work and guidance
has made his work possible.

Eidesstattliche Versicherung / Declaration on oath

Hiermit versichere ich an Eides statt, die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt zu haben.

Hamburg, 2022

(Vitalii Maheria)

Gutachter/innen der Dissertation:	Prof. Dr. Sven-Olaf Moch
	Prof. Dr. Bernd Andreas Kniehl
Zusammensetzung der Prüfungskommission:	Dr. Markus Diehl
	Dr. Judith Katzy
	Prof. Dr. Bernd Andreas Kniehl
	Prof. Dr. Sven-Olaf Moch
	Prof. Dr. Peter Schmelcher
Vorsitzende/r der Prüfungskommission:	Prof. Dr. Peter Schmelcher
Datum der Disputation:	01.07.2022
Vorsitzender Fach-Promotionsausschusses PHYSIK:	Prof. Dr. Wolfgang Parak
Leiter des Fachbereichs PHYSIK:	Prof. Dr. Günter H. W. Sigl
Dekan der Fakultät MIN:	Prof. Dr. Heinrich Graener

Abstract

In this thesis we present the analytic calculation of all master integrals for 3-, 4-, and 5-particle cuts of massless 4-loop propagators, as well as 3-, 4-, and 5-particle semi-inclusive cuts of massless 4-loop propagators—all of which are novel results, obtained using multiple techniques: differential equations, dimensional recurrence relations, and direct phase-space integration. These integrals are needed for the direct calculation of next-to-next leading order (NNLO) time-like splitting functions (that is, scale evolution kernels of the fragmentation functions; a required ingredient for the analysis of semi-inclusive single hadron production at Large Hadron Collider and other colliders), and N^{3}LO photonic coefficient functions (a required ingredient for the analysis of e^+e^- annihilation at future colliders).

Additionally we present computer programs enabling this calculation: FUCHSIA, a tool for reducing differential equations to an ε-form, and FEYNSON, a tool for resolving Feynman integral symmetries. Both tools are of general usefulness outside of this work, with FUCHSIA in particular already having found use in multiple independent calculations.

Zusammenfassung

Diese Arbeit präsentiert die analytische Berechnung aller Masterintegrale mit 3-, 4-, und 5-Teilchen-Schnitten der masselosen 4-Schleifen-Propagatoren, sowie der semi-inklusiven 3-, 4-, und 5-Teilchen-Schnitte der masselosen 4-Schleifen-Propagatoren. Dies alles sind komplett neue Ergebnisse, welche mit den folgenden Methoden erhalten wurden: Differentialgleichungen, dimensionale Rekursionsgleichungen und direkte Phasenraumintegration. Diese Integrale werden benötigt für die direkte Berechnung der nächst-nächst führenden Ordnung (NNLO) der zeitartigen Splitting-Funktionen (diese sind Kerne der Differentialgleichung für die Skalenevolution der Fragmentationsfunktionen und notwendige Bestandteile der Analyse der semi-inklusiven Produktion einzelner Hadronen am Large Hadron Collider und an anderen Streuexperimenten), sowie der N^{3}LO Photonenkoeffizientenfunktionen (diese sind notwendige Bestandteile der Analysen von e^+e^- Annihilationen an zukünftigen Streuexperimenten).

Zusätzlich präsentieren wir die folgenden Computerprogramme: FUCHSIA, ein Programm für die Reduktion von Differentialgleichungen auf eine ε-Form, und FEYNSON, ein Programm zur Auflösung von Symmetrien zwischen Feynmanintegralen. Beide Programme sind von universeller Nützlichkeit auch über den Rahmen dieser Arbeit hinaus. Insbesondere hat FUCHSIA bereits Anwendung in mehreren unabhängigen Berechnungen gefunden.

Contents

1. Introduction

In the years 1665-1666 an outbreak of the bubonic plague in London forced the rich to flee, the poor to isolate, and Isaac Newton to study *natural philosophy* in near total solitude. 355 years later, the tradition remains the same; we now call it *physics*.

1.1. The general motivation

The goal of physics, same as any natural science, is to further our shared understanding of nature. This understanding has many frontiers; as far as fundamental laws of nature are concerned, the frontier lies within two areas: the Standard Model of particle physics ("SM"), which describes strong, electroweak, and Higgs' interactions of matter, and the general theory of relativity ("GR"), which describes the gravitational interaction.[1] Both theories are wildly successful and precisely tested, and yet both are incomplete. We know this because:

1. Standard Model and general relativity are incompatible with each other, in the sense that a naïve construction of quantum gravity is unrenormalizable, and thus can only be viewed as a low-energy effective theory of some more general theory. The work to construct such theories is ongoing, but so far the experimental evidence to support any is lacking.

   Just from the dimensional analysis it is expected that both quantum and gravitational effects would be relevant simultaneously at energies of the order of the Planck scale, $\sqrt{\hbar c^5/G} \approx 10^{28}$eV, but such energies are unobtainable in a lab, with the largest particle accelerator today, the Large Hadron Collider, achieving only $\approx 10^{13}$eV. Thus, the definitive resolution of this problem is experimentally out of reach.

2. The discovery of neutrino oscillations [1, 2] imply that neutrinos have masses. These need to be incorporated into the SM, and there are multiple ways to do so. Experiments are underway to measure the properties of neutrinos more precisely [3, 4].

3. The existence of the so-called dark matter in the universe: the masses of many galaxies determined from the luminosities of their stars appears to be smaller than the same masses determined from their gravitational properties (e.g. the motion of the stars, gravitational lensing, etc). For some galaxies the difference can be quite drastic, for others practically zero. Thus, aside from the luminous matter (stars and interstellar gas) there appears to be a varying amount of dark matter present in each galaxy. The average dark matter mass proportion in the observed universe is estimated to be 85%. The nature of this matter is presently unknown, and the working conjecture is that it consists of a new class of stable particles, characterized by very weak interaction with photons and ordinary matter in general. Candidate SM extensions include axions [5], weakly interacting massive particles ("WIMPs"), and multiple others.

[1] Conventionally, Higgs' interaction is not classified as one of the "fundamental forces of nature". In our opinion this is only because its experimental discovery in 2012 is still fairly recent, and the conventions have not yet caught up—otherwise it is by all means a fundamental force of nature.

4. There is an asymmetry between the amount of matter and antimatter in the universe. The observable universe (i.e. the stars and interstellar gas) consists of ordinary matter, with anti-matter being absent. For these conditions to arise dynamically from a symmetric starting point during the Big Bang, several conditions must be met [6]: baryon number non-conservation, charge-parity (CP) symmetry violation, and departure from thermal equilibrium. The weak interactions in SM do violate the CP symmetry, but calculations show that a much stronger effect is required to explain the observed matter-antimatter asymmetry, and multiple SM extensions are put forward as potential solutions [7, 8].

The takeaway here is that the SM is incomplete, multiple extensions are proposed, and trying to describe and observe the class of particles that will solve problems #3 and #4 is a big part of what preoccupies the field of particle physics at the moment.

There are three general avenues available for these efforts: observation of cosmic phenomena which can imply limits on the properties of new unknown classes of particles, direct detection searches (see e.g. [9]), and general searches at particle colliders. Out of the three the last one has been (and remains) the most productive, with quarks [10, 11], gluons [12, 13], $W^{\pm}$ and Z^{0} bosons [14, 15], and Higgs particles [16, 17] all discovered at different colliders.

The largest collider to date is the Large Hadron Collider (LHC), achieving 13 TeV collision energy. In 2012 two of its experimental groups, ATLAS and CMS, reported the first detection of events attributed to the Higgs' particle. Since then efforts are ongoing to measure the interaction of Higgs with quarks (mainly the top quarks, being the heaviest) and vector bosons, with the LHC High-Luminosity (LHC-HL) upgrade [18, 19] promising much more events (at possibly slightly higher energies), greatly increasing the experimental precision, possibly allowing the precise study of other interactions (e.g. Higgs' self-interactions).

Those efforts aside, the general public is most interested in one question: "What's next?" At present there are no strong signals indicating *direct* production of new unknown particles, and with the LHC-HL collision energy not being significantly higher than the LHC one, the hope for such signals appearing is low—not until a higher-energy collider like FCC [20], ILC [21], CLIC [22], or CEPC [23] is constructed. For the time being the most promising strategy is to search for evidence of new particles not through direct production, but rather by deviations of the experimental measurements from theoretical prediction—however small—where such particles participate as virtual corrections. This is what the high-luminosity effort is aiming at: increasing the statistical accuracy so that even small deviations could become visible.

Such a goal however poses a problem on the theoretical side: already with the currently available LHC data theoretical uncertainties on key measurements of differential cross-sections are comparable to the experimental uncertainties, so to make use of the data from LHC-HL or the future colliders theoretical predictions must to be upgraded too. One of the key limiting factors here is the depth of the perturbative expansion of the analytic results, and so systematic next-to-next-to-leading order (NNLO) or even N^{3}LO results are needed for all observables to match the experimental precision.

These are the considerations that bring us to the calculation of higher loop integrals relevant to NNLO terms of the perturbative expansion of scattering amplitudes in general.

1.2. The goal of this thesis

The specific topic of this thesis is the calculation of NNLO QCD corrections to time-like splitting functions and the closely related photonic decay coefficient functions. These are relevant to the analysis of hadron production at any collider, and to electron-positron annihilation at $e^{+}e^{-}$ colliders, both past (i.e. Large Electron-Positron Collider, LEP) and future: CLIC, ILC, and CEPC are all $e^{+}e^{-}$ machines, and FCC contains provisions for both e^+e^- and protons. The splitting functions in question have already been calculated previously almost fully in [24, 25, 26], but the method employed—analytic continuation from space-like splitting functions—has left the answer incomplete. Specifically, one of the terms in quark-gluon and gluon-quark NNLO terms was left undetermined, a gap that was closed only recently by results from soft-collinear effective theory [27]. The corresponding coefficient functions are still not known (for the same reason).

We aim at calculating NNLO photonic coefficient functions, and recalculating NNLO time-like splitting functions from the ground up using the direct approach advocated for in [28]. To this end we shall calculate the full set of four-loop Feynman master integrals with cuts: both inclusive and semi-inclusive, all of which are needed for the calculation of semi-inclusive decay cross-section at NNLO, from which the splitting and coefficient functions can be directly extracted. Along the way we shall also develop techniques and software that will make these calculations possible. Specifically, among the code that powers our calculations two parts were factored out as standalone tools useful outside of it: the program FUCHSIA that helps to solve differential equations for master integrals by transforming them into an ε-form [29, 30], and the program FEYNSON which finds symmetry relations between Feynman integrals and is needed as the step preceding IBP reduction.[2] FUCHSIA is of special note here, having been already used in calculations such as [31, 32, 33, 34].

Our calculation of fully inclusive cut integrals was previously presented in [35, 36]; the previous version of FUCHSIA was described in [37, 38]. The semi-inclusive cut integrals (Chapter 10), the new version of FUCHSIA (Chapter 8), and FEYNSON (Chapter 4) are described here for the first time.

This thesis is organized as follows. In Chapter 2 we will back up a bit and continue from the general motivation, zooming in on how splitting and coefficient functions enter in the greater picture of analyzing the experimental results from particle colliders, and establish which Feynman integrals we will need to calculate. In Chapter 3 we will review the basic techniques used to calculate Feynman integrals analytically, focusing on the methods we will use later on. In Chapter 4 we will describe FEYNSON, a computer implementation of a subset of these techniques covering the discovery of symmetries between Feynman integrals. In Chapter 5 and Chapter 6 we will specifically focus on calculating Feynman integrals using dimensional recurrence relations, and differential equations respectively. In Chapter 7 we will cover the class of function arising from differential equations—multiple polylogarithms. In Chapter 8 we will describe FUCHSIA, a computer program for reducing the differential equations to an ε-form, which is the central step of solving them. Finally, in Chapter 9 we will present the calculation of the full set of master integrals for fully inclusive cuts of 4-loop propagators, following up with the full set of semi-inclusive ones in Chapter 10. We shall summarize the results in Chapter 11.

[2] A third part of this code, "the amplitude library" ALIBRARY, is also public (github.com/magv/alibrary), but it is not described here.

2. Theoretical background

The principal method of study of elementary particle interactions available today is the analysis of scattering processes happening at particle colliders. On such a machine a beam of accelerated particles—often protons, electrons, or positrons—is directed to intersect with an opposing beam (or a fixed target), and the stream of reaction products is then recorded, analyzed by extracting various observable quantities (statistically aggregated variables), and compared with the theoretical predictions. The higher the energy of the beams, the shorter the scale of the interactions that can be probed, and the finer the structure of particles that can be resolved.

Out of observable quantities of interest the most straightforward is the total scattering cross section: σ, the proportion of scattering events per unit of incoming particles times the beam cross-section area (filtered by the types of particles detected in the final state), and differential cross sections: $\sigma(x) \equiv \mathrm{d}\sigma/\mathrm{d}x$, the distributions of σ in variables x such as the scattering angles, energies, or transverse momenta of the scattering products. Up to normalization both correspond to the probability of scattering (square of the scattering amplitude), fully or partially integrated over the final state parameters.

Of particular interest is the measurement of the distribution of energies of kaons (K^+ and K^-) and pions (π^+ , π^-) that result from the scattering. These are relatively long-lived hadronic particles (i.e. bound states of quarks) that can be identified precisely in the detectors by their mass and electric charge. In this thesis we will be mostly concerned with integrals needed for the calculation of such differential cross sections.

2.1. Fragmentation, splitting, and coefficient functions

A convenient theoretical picture of the formation of hadrons in the final state is the parton fragmentation model:[1] first the particles from the accelerated beams interact at high energies producing a multitude of partons (quarks and gluons); then as partons drift apart they fragment and lose energy, and eventually due to their effective interaction strength increasing at lower energies (a phenomenon known as "confinement") they merge into combinations of zero total color charge, forming hadrons ("hadronization"). See Figure 2.1.1 for a schematic illustration. The first part—the high-energy interaction of incoming beam particles—can be conveniently described by a perturbative expansion, while the second—hadronization of the partons—cannot, so instead it is modeled and fitted to data. The reason is that hadronization happens at lower energies where the perturbative expansion does not converge, and we have no good tools of solving quantum field theory calculations non-perturbatively (aside from lattice calculations, the applications of which are currently limited).

One model of parameterizing hadronization is to assume that each parton hadronizes independently, and so the process can be described by a single *fragmentation function* $D_{p\to H}(x)$, which gives the probability density of a parton $p \in \{q, \bar{q}, g\}$ of 4-momentum q hadronizing into a hadron $H \in \left\{K^\pm, \pi^\pm, \dots\right\}$ of energy $x\sqrt{q^2}$, where $0 \le x \le 1$ is just the hadron's fraction of the energy. The fragmentation function $D_{p\to H}$ is

[1] We recommend [39, 40] for a readable description of this topic.

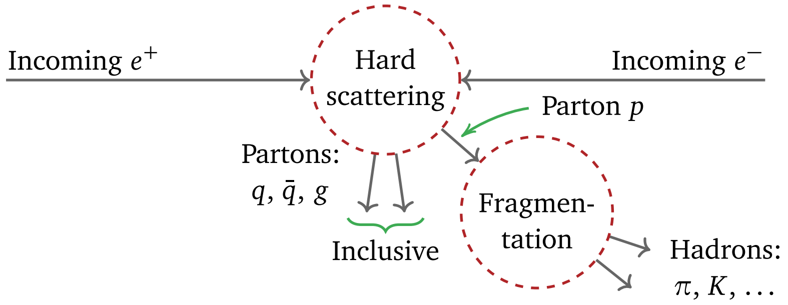


Figure 2.1.1.: Hadron production on a e^+e^- collider.

process-independent: it can be tabulated from one experiment and then reused for all the others. The principal complication here is that in perturbative quantum chromodynamics (QCD) $D_{p\to H}$ acquires a dependence on the renormalization scale μ^2; this dependence however is governed by the well known renormalization group equations of the DGLAP type [41, 42, 43], and so can be described perturbatively.

To see how $D_{p\to H}(x,\mu^2)$ depends on μ^2 let us take a look at an annihilation of an electron-positron pair that results in a hadron H (along with other particles which we will integrate out). From Figure 2.1.1, if parton p is produced with energy fraction z, and hadron H takes a fraction z' of that, then the semi-inclusive hadronic cross section (that is, cross section of the hadron differential only in its overall energy fraction $x = zz'$) is connected to the semi-inclusive partonic cross section as

$$\underbrace{\sigma_{\text{hadron } H}(\text{energy fraction } x)}_{\text{Semi-inclusive handronic cross-section}} = \sum_{p\in\{q,\bar{q},g\}} \int \overbrace{\hat{\sigma}_{\text{parton } p}(z,\hat{\alpha}_s)}^{\text{Semi-inclusive partonic cross-section}} \hat{D}_{p\to H}(z')\,\delta(x-zz')\,\mathrm{d}z\,\mathrm{d}z', \tag{2.1.1}$$

where μ^2 is the energy of the parton p, the fragmentation scale, and the renormalization scale: for convenience these are taken to be the same here to prevent the proliferation of scale ratios.[2]

Note that here and throughout this chapter we mark all *bare* (unrenormalized) quantities like $\hat{\sigma}$ with a "hat" sign to distinguish them from their renormalized counterparts which we shall soon introduce.

At this point we can already simplify the notation by introducing the *convolution operation* as

$$(f\otimes g)(x) \equiv \int_0^1 \mathrm{d}z \int_0^1 \mathrm{d}z'\, f(z)\, g(z')\,\delta(x-zz'), \tag{2.1.2}$$

so that eq. (2.1.1) would shorten to just

$$\sigma_H = \sum_p \hat{\sigma}_p \otimes \hat{D}_{p\to H}. \tag{2.1.3}$$

Note that

$$f\otimes g = g\otimes f, \qquad \text{and} \qquad f\otimes\delta(1-x) = f, \tag{2.1.4}$$

so we can treat $\otimes$ as multiplication, and $\delta(1-x)$ as a multiplicative identity.

Because $\sigma_H(x)$ can be measured experimentally (by counting observed hadrons in each energy fraction range), it is necessarily finite; its partonic counterpart $\hat{\sigma}_p(z,\hat{\alpha}_s)$ on the other hand is not experimentally

[2] For a general overview of fragmentation functions and related topics see [44], or [40] upon which it is based. The method of calculation we use in this section was previously used in [45, 46, 47, 48], with [46] giving a particularly detailed description in modern notation.

observable (because neither free quarks nor free gluons exist), and calculations of it starting from the QCD Lagrangian (as given in e.g. Appendix B) as a perturbative expansion in $\hat{a}_s$,

$$\hat{\sigma}_p(z,\hat{a}_s) \equiv \sum_{k=0}^{\infty} \hat{\sigma}_p^{(k)}(z)\,\hat{a}_s^k, \qquad \text{where} \qquad \hat{a}_s \equiv \frac{\hat{\alpha}_s}{4\pi} \qquad \text{and} \qquad \hat{\alpha}_s \equiv \frac{\hat{g}^2}{4\pi}, \tag{2.1.5}$$

results in divergent Feynman integrals for $\hat{\sigma}_p^{(1)}$ and onward. There are two types of divergences present here: those stemming from integration in the regions of large loop momenta (ultraviolet, or UV, divergences), and those stemming from from regions of loop momenta collinear to the external moment (collinear, or "mass", divergences).

The solution is three-fold.

2.1.1. Regularization

First we must *regularize* the divergences by introducing an artificial regularization parameter in a way that will make them diverge only in its limit.

To this end we shall use *dimensional regularization* [49]: a convenient and widely used regularization scheme for QCD, in which both the loop and the external momenta in all the integrals are taken to reside not in 4 space-time dimensions, but in $d = 4 - 2\varepsilon$ instead, so that ε becomes the regularizing parameter, the limit $\varepsilon \to 0$ restores the original physical picture, and integral divergences manifest as $1/\varepsilon$ poles (and higher).

Because dimensional regularization changes the dimensionality of the momenta, the strong coupling parameter g_s appearing in the QCD Lagrangian ceases to be dimensionless. For this reason we need to introduce an additional arbitrary mass scale μ, and write the strong coupling parameter as $\mu^\varepsilon \hat{g}_s$, where $\hat{g}_s$ is kept dimensionless. Correspondingly, the d-dimensional strong coupling constant must be written as $\mu^{2\varepsilon}\hat{\alpha}_s$, rather than just $\hat{\alpha}_s$.

2.1.2. UV renormalization

Next we *renormalize* the bare $\hat{\sigma}_p(z,\hat{\alpha}_s)$ into the UV-renormalized $\sigma_p(z,\alpha_s)$ by absorbing its UV divergences into the definition of the bare coupling constant $\hat{\alpha}_s$.[3]

For this we exploit the fact that $\hat{\alpha}_s$ is not an observable parameter, so it can be given poles in ε that would counteract the UV poles of $\hat{\sigma}_p$:

$$\hat{\alpha}_s = \underbrace{Z_{\alpha_s}}_{\text{divergent}} \overbrace{\alpha_s,}^{\text{finite}} \qquad \text{and} \qquad \underbrace{\sigma_p(x,\alpha_s)}_{\text{renormalized}} = \underbrace{\hat{\sigma}_p\big(x, Z_{\alpha_s}\alpha_s\big),}_{\text{bare}} \tag{2.1.6}$$

where Z_{α_s} is the α_s renormalization constant: a divergent quantity (i.e. with poles in ε), chosen such that $\hat{\sigma}_p\big(x, Z_{\alpha_s}\alpha_s\big)$ becomes finite.

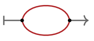
There are multiple equally valid schemes to choose renormalization constants (after all, we require σ_p only to be finite, without imposing constraints on its value); we shall adopt the *modified minimal subtraction scheme* [51, 52], a.k.a. $\overline{\text{MS}}$, in which the regularization scale $\bar{\mu}$ is used instead of μ, with

$$\mu^{2\varepsilon} \equiv \bar{\mu}^{2\varepsilon} S_\varepsilon, \qquad \text{and} \qquad S_\varepsilon \equiv \left(\frac{e^{\gamma_E}}{4\pi}\right)^{\varepsilon}, \tag{2.1.7}$$

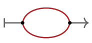
[3] See [50] for a general overview of renormalization.

effectively multiplying each loop integration by S_ε, and the renormalization of bare quantities is performed by subtracting all of their poles in ε, but leaving the finite $\mathcal{O}(\varepsilon^0)$ parts intact.

Here γ_E stands for the Euler constant; the factor of S_ε is introduced to remove ubiquitous terms of the form $\gamma_E - \log(4\pi)$ from the integrals' values when presented as series in ε. For example one of the simplest loop integrals we shall consider in Chapter 3 is the bubble integral given by eq. (3.2.10), and its value simplifies when multiplied by S_ε from


$$= \left(-q^2 - i0\right)^{-\varepsilon} \frac{i}{16\pi^2} \left(\frac{1}{\varepsilon} + (2 + \log(4\pi) - \gamma_E) + \mathcal{O}(\varepsilon)\right) \tag{2.1.8}$$

to

S_ε 
$$= \left(-q^2 - i0\right)^{-\varepsilon} \frac{i}{16\pi^2} \left(\frac{1}{\varepsilon} + 2 + \mathcal{O}(\varepsilon)\right). \tag{2.1.9}$$

The value of Z_{a_s} in $\overline{\text{MS}}$ is known from [53] up to $\mathcal{O}(a_s^4)$ and from [54, 55, 56] up to $\mathcal{O}(a_s^5)$, and can be expressed in terms of the coefficients of the β-function given in those works,

$$\beta(a_s) \equiv \frac{\mathrm{d}a_s}{\mathrm{d}\ln\mu^2} \equiv -\varepsilon a_s - \beta_0 a_s^2 - \beta_1 a_s^3 - \beta_2 a_s^4 - \beta_3 a_s^5 + \mathcal{O}(a_s^6), \tag{2.1.10}$$

if one demands that the bare strong coupling $\mu^{2\varepsilon}\hat{a}_s = \mu^{2\varepsilon} Z_{a_s} a_s$ does not depend on the renormalization scale:

$$\frac{\mathrm{d}}{\mathrm{d}\ln\mu^2}\left(\mu^{2\varepsilon} Z_{a_s} a_s\right) = 0 \qquad \Rightarrow \qquad \beta(a_s) = -\frac{\varepsilon Z_{a_s} a_s}{a_s \frac{\partial}{\partial a_s} Z_{a_s} + Z_{a_s}}. \tag{2.1.11}$$

This works out into

$$\begin{aligned} Z_{a_s} =& 1 + \left(-\frac{\beta_0}{\varepsilon}\right) a_s + \left(\frac{\beta_0^2}{\varepsilon^2} - \frac{\beta_1}{2\varepsilon}\right) a_s^2 + \left(-\frac{\beta_0^3}{\varepsilon^3} + \frac{7\beta_0\beta_1}{6\varepsilon^2} - \frac{\beta_2}{3\varepsilon}\right) a_s^3 + \\ &+ \left(\frac{\beta_0^4}{\varepsilon^4} - \frac{23\beta_0^2\beta_1}{12\varepsilon^3} + \frac{9\beta_1^2 + 20\beta_0\beta_2}{24\varepsilon^2} - \frac{\beta_3}{4\varepsilon}\right) a_s^4 + \mathcal{O}(a_s^5). \end{aligned} \tag{2.1.12}$$

2.1.3. Collinear factorization

As the last step we require that the remaining collinear divergences of $\sigma_p(z, a_s)$ would factorize so we could absorb them into $\hat{D}_{p\to H}(x, \mu^2)$, making both finite:[4]

$$\sigma_H = \sum_p \underbrace{\sigma_p \otimes \hat{D}_{p\to H}}_{\text{collinearly divergent}} = \sum_p \underbrace{C_p \otimes D_{p\to H}}_{\text{fully finite}}, \tag{2.1.13}$$

where C_p are the finite *coefficient functions*, and $D_{p\to H}$ are the finite fragmentation functions, no longer "bare". This transformation can be achieved by requiring that the collinear divergences of σ_p are factorized via some transition functions $\Gamma_{pp'}$ as

$$\sigma_p = \Gamma_{pp'} \otimes C_{p'}, \qquad \text{and correspondingly} \qquad D_{p\to H} = \hat{D}_{p'\to H} \otimes \Gamma_{p'p}. \tag{2.1.14}$$

To determine $\Gamma_{pp'}$ we only need to require that the bare fragmentation functions do not depend on μ^2,

$$\frac{\mathrm{d}}{\mathrm{d}\ln\mu^2}\hat{D}_{p\to H} = 0 \qquad \Rightarrow \qquad \frac{\mathrm{d}}{\mathrm{d}\ln\mu^2} D_{p\to H} = D_{p''\to H} \otimes \Gamma^{-1}_{p''p'} \otimes \frac{\mathrm{d}\Gamma_{p'p}}{\mathrm{d}a_s}\beta, \tag{2.1.15}$$

[4] See [57] for a review of proofs of this factorization.

where $\Gamma^{-1}_{pi} \otimes \Gamma_{ip'} = \delta(1-x)\,\delta_{pp'}$ so that $\hat{D}_p = D_{p'} \otimes \Gamma^{-1}_{p'p}$, and the combinations

$$P_{pp'} \equiv \Gamma^{-1}_{pi} \otimes \frac{\mathrm{d}\Gamma_{ip'}}{\mathrm{d}a_s}\beta \tag{2.1.16}$$

are called the *time-like splitting functions*. These are often denoted as $P^T_{pp'}$ to distinguish them from the *space-like splitting functions* which play the same role for parton density functions (PDFs) as $P^T_{pp'}$ play for fragmentation functions. The "time-like" versus "space-like" refers to the fact that $q^2 > 0$ in e^+e^- annihilation, but $q^2 < 0$ in deep inelastic scattering.

If one expands $P_{pp'}$ in a_s as

$$P_{pp'} \equiv P^{(0)}_{pp'}a_s + P^{(1)}_{pp'}a_s^2 + P^{(2)}_{pp'}a_s^3 + \mathcal{O}\left(a_s^4\right), \tag{2.1.17}$$

then eq. (2.1.16) can be solved for $\Gamma_{pp'}$ in terms of $P_{pp'}$ as[5]

$$\begin{aligned}\Gamma_{pp'} =& \delta_{pp'}\delta(1-x) + \left(\frac{-P^{(0)}_{pp'}}{\varepsilon}\right)a_s + \left(\frac{-P^{(1)}_{pp'}}{2\varepsilon} + \frac{P^{(0)}_{pi}\otimes P^{(0)}_{ip'} + \beta_0 P^{(0)}_{pp'}}{2\varepsilon^2}\right)a_s^2 + \\ &+ \left(\frac{-P^{(2)}_{pp'}}{3\varepsilon} + \frac{P^{(1)}_{pi}\otimes P^{(0)}_{ip'} + 2P^{(0)}_{pi}\otimes P^{(1)}_{ip'} + 2\beta_0 P^{(1)}_{pp'} + 2\beta_1 P^{(0)}_{pp'}}{6\varepsilon^2} + \right. \\ &+ \left. \frac{-P^{(0)}_{pi}\otimes P^{(0)}_{ij}\otimes P^{(0)}_{jp'} - 3\beta_0 P^{(0)}_{pi}\otimes P^{(0)}_{ip'} - 2\beta_0^2 P^{(0)}_{pp'}}{6\varepsilon^3}\right)a_s^3 + \mathcal{O}\left(a_s^4\right),\end{aligned} \tag{2.1.18}$$

and its inverse as

$$\begin{aligned}\Gamma^{-1}_{pp'} =& \delta_{pp'}\delta(1-x) + \left(\frac{P^{(0)}_{pp'}}{\varepsilon}\right)a_s + \left(\frac{P^{(1)}_{pp'}}{2\varepsilon} + \frac{P^{(0)}_{pi}\otimes P^{(0)}_{ip'} - \beta_0 P^{(0)}_{pp'}}{2\varepsilon^2}\right)a_s^2 + \\ &+ \left(\frac{P^{(2)}_{pp'}}{3\varepsilon} + \frac{2P^{(1)}_{pi}\otimes P^{(0)}_{ip'} + P^{(0)}_{pi}\otimes P^{(1)}_{ip'} - 2\beta_0 P^{(1)}_{pp'} - 2\beta_1 P^{(0)}_{pp'}}{6\varepsilon^2} + \right. \\ &+ \left. \frac{P^{(0)}_{pi}\otimes P^{(0)}_{ij}\otimes P^{(0)}_{jp'} - 3\beta_0 P^{(0)}_{pi}\otimes P^{(0)}_{ip'} + 2\beta_0^2 P^{(0)}_{pp'}}{6\varepsilon^3}\right)a_s^3 + \mathcal{O}\left(a_s^4\right).\end{aligned} \tag{2.1.19}$$

2.1.4. Summary

Combined together, we have went from eq. (2.1.1) with divergent bare cross-sections $\hat{\sigma}_p$ and bare fragmentation functions $\hat{D}_{p\to H}$ to the finite coefficient functions C_p and finite fragmentation functions $D_{p\to H}$, with the hadronic cross-section becoming

$$\sigma_H(x) = \sum_p C_p\left(x, \alpha_s\left(\mu^2\right)\right) \otimes D_{p\to H}\left(x, \mu^2\right). \tag{2.1.20}$$

Further, we have established the dependence of $D_{p\to H}$ on μ^2 as

$$\frac{\mathrm{d}}{\mathrm{d}\ln\mu^2}D_{p\to H} = D_{p'\to H} \otimes P_{p'p}, \tag{2.1.21}$$

[5] We have used the NCAlgebra package (github.com/NCAlgebra/NC) to derive this solution.

where both the splitting functions $P_{p'p}$ and the coefficient functions C_p can be extracted from σ_p through the combination of eq. (2.1.14) and eq. (2.1.18). Specifically, if we represent C_p as

$$C_p \equiv C_p^{(0)} + \left(C_p^{(1)} + \varepsilon A_p^{(1)} + \varepsilon^2 B_p^{(1)} + \mathcal{O}\left(\varepsilon^3\right)\right) a_s + \left(C_p^{(2)} + \varepsilon A_p^{(2)} + \mathcal{O}\left(\varepsilon^2\right)\right) a_s^2 + \left(C_p^{(3)} + \mathcal{O}(\varepsilon)\right) a_s^3 + \mathcal{O}\left(a_s^4\right), \tag{2.1.22}$$

then the UV-renormalized $\sigma_p(x, \alpha_s) \equiv \hat{\sigma}_p\left(x, Z_{\alpha_s} \hat{\alpha}_s\right)$ must have the form[6]

$$\begin{aligned}
\sigma_p =& C_p^{(0)} + \\
&+\left(-\frac{1}{\varepsilon} P_{pp'}^{(0)} \otimes C_{p'}^{(0)} + C_p^{(1)} + \varepsilon A_p^{(1)} + \varepsilon^2 B_p^{(1)} \right) a_s + \\
&+\left(\frac{1}{\varepsilon^2} \left(\frac{1}{2} P_{pi}^{(0)} \otimes P_{ip'}^{(0)} + \frac{\beta_0}{2} P_{pp'}^{(0)} \right) \otimes C_{p'}^{(0)} + \right. \\
&\quad + \frac{1}{\varepsilon} \left\{ -\frac{1}{2} P_{pp'}^{(1)} \otimes C_{p'}^{(0)} - P_{pp'}^{(0)} \otimes C_{p'}^{(1)} \right\} + \left(C_p^{(2)} - P_{pp'}^{(0)} A_{p'}^{(1)} \right) + \varepsilon \left(A_p^{(2)} - P_{pp'}^{(0)} B_{p'}^{(1)} \right) \left. \right) a_s^2 + \\
&+\left(\frac{1}{\varepsilon^3} \left(-\frac{\beta_0^2}{3} P_{pp'}^{(0)} - \frac{\beta_0}{2} P_{pi}^{(0)} \otimes P_{ip'}^{(0)} - \frac{1}{6} P_{pi}^{(0)} \otimes P_{ij}^{(0)} \otimes P_{jp'}^{(0)} \right) \otimes C_{p'}^{(0)} + \right. \\
&\quad + \frac{1}{\varepsilon^2} \left\{ \left(\frac{\beta_0}{2} P_{pp'}^{(0)} + \frac{1}{2} P_{pi}^{(0)} \otimes P_{ip'}^{(0)} \right) \otimes C_{p'}^{(1)} + \left(\frac{\beta_1}{3} P_{pp'}^{(0)} + \frac{\beta_0}{3} P_{pp'}^{(1)} + \frac{1}{3} P_{pi}^{(0)} \otimes P_{ip'}^{(1)} + \frac{1}{6} P_{pi}^{(1)} \otimes P_{ip'}^{(0)} \right) \otimes C_{p'}^{(0)} \right\} + \\
&\quad + \frac{1}{\varepsilon} \left\{ -\frac{1}{3} P_{pp'}^{(2)} \otimes C_{p'}^{(0)} - \frac{1}{2} P_{pp'}^{(1)} \otimes C_{p'}^{(1)} - P_{pp'}^{(0)} \otimes C_{p'}^{(2)} + \left(\frac{\beta_0}{2} P_{pp'}^{(0)} + \frac{1}{2} P_{pi}^{(0)} \otimes P_{ip'}^{(0)} \right) \otimes A_{p'}^{(1)} \right\} + \\
&\quad + \left\{ C_p^{(3)} - P_{pp'}^{(0)} \otimes A_{p'}^{(2)} - \frac{1}{2} P_{pp'}^{(1)} \otimes A_{p'}^{(1)} + \left(\frac{\beta_0}{2} P_{pp'}^{(0)} + \frac{1}{2} P_{pi}^{(0)} \otimes P_{ip'}^{(0)} \right) \otimes B_{p'}^{(1)} \right\} \left. \right) a_s^3 + \mathcal{O}\left(a_s^4\right).
\end{aligned} \tag{2.1.23}$$

From this we see that $P^{(0)}$ first enters at $\mathcal{O}(1/\varepsilon, a_s)$, $P^{(1)}$ at $\mathcal{O}\left(1/\varepsilon, a_s^2\right)$, and $P^{(2)}$ at $\mathcal{O}\left(1/\varepsilon, a_s^3\right)$. Similarly, $C^{(1)}$ enters at $\mathcal{O}\left(\varepsilon^0, a_s\right)$, $C^{(2)}$ at $\mathcal{O}\left(\varepsilon^0, a_s^2\right)$, and $C^{(3)}$ at $\mathcal{O}\left(\varepsilon^0, a_s^3\right)$.

Here we would like to stress that even though the explanation above was for e^+e^- annihilation, only the coefficient functions are specific to this process; the fragmentation functions govern the hadronization of partons and are process-independent.

2.2. State of the art

QCD splitting functions are fully known at the leading [58, 59] and next-to-leading orders (NLO) [60, 61]. At the next-to-next-to-leading order (NNLO) the space-like fragmentation functions were calculated in [62, 63] by calculating the cross section of a deep-inelastic scattering (DIS) process via the optical theorem, and the time-like were derived from them by an analytic continuation procedure in [24, 25, 26]. The procedure employed there has an inherent ambiguity, which has left some of the terms of $P_{q\to g}^{(2)T}$ and $P_{g\to q}^{(2)T}$ undetermined. Recently, this gap in the time-like splitting functions was closed in [27] using results for space-like transverse-momentum-dependent (TMD) splitting functions, for which a similar analytic continuation procedure is unambiguous. Currently work is ongoing to calculate the 4-th order (N^{3}LO) corrections, with the leading-color limit for both space-like and time-like presented in [64, 65].

The coefficient functions for e^+e^- annihilation are known at α_s^1 [66, 48] and α_s^2 [67, 47, 68] orders.

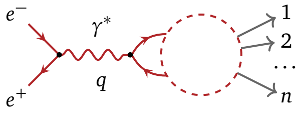
[6] Note that our eq. (2.1.23) reproduces eq.(2.6), eq.(2.7), and eq.(2.8) from [28] up to a typo in eq.(2.8), where the combination $\frac{1}{6}P_{pi}^{(0)}P_{iq}^{(1)} + \frac{1}{3}P_{pi}^{(1)}P_{iq}^{(0)}$ next to $1/\varepsilon^2$ should read $\frac{1}{6}P_{pi}^{(1)}P_{iq}^{(0)} + \frac{1}{3}P_{pi}^{(0)}P_{iq}^{(1)}$ instead. We have confirmed this typo with the authors.

This brings us to the goal of this thesis: as can be seen from eq. (2.1.23) (and previously advocated by [28, 69]) if one knows $\sigma_q(x)$ and $\sigma_g(x)$ up to $\mathcal{O}\left(a_s^3\right)$, then the recently calculated $P^{(2)T}_{q\to g}$ and $P^{(2)T}_{g\to q}$ can be extracted from $\mathcal{O}\left(1/\varepsilon, a_s^3\right)$ of these results, and the currently unknown $C^{(3)}_q$ with $C^{(3)}_g$ can be extracted from $\mathcal{O}\left(\varepsilon^0, a_s^3\right)$. Calculating semi-inclusive cross-sections up to α_s^3 however is not a trivial exercise because the integrals appearing at this order are not yet known, so in the rest of this thesis we shall concentrate on completing the calculation of all integrals required for $\sigma_p(x)$ at $\mathcal{O}\left(a_s^3\right)$.

2.3. Calculating the cross sections

The calculation of $\sigma_p(x)$ in perturbation theory involves the calculation scattering amplitudes, and thus Feynman diagrams. In general we have

$$\sigma_p(x) \propto \sum_n \int \mathrm{dPS}_n(x) \Bigg| \sum_{\text{diagrams}}$$



$$\Bigg|^2, \qquad (2.3.1)$$

where dPS(x) is the semi-inclusive phase-space volume element,

$$\mathrm{dPS}_n(x) = \left(\prod_{i=1}^{n} \frac{\mathrm{d}^d p_i}{(2\pi)^{d-1}} \delta^+\left(p_i^2\right)\right) (2\pi)^d\, \delta^{(d)}\left(q - \sum_{i=1}^{n} p_i\right) \delta\left(x - 2\frac{q\cdot p_1}{q^2}\right), \qquad (2.3.2)$$

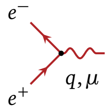
and we have assumed that the momentum p_1 corresponds to the parton of type p that we are after.

To further simplify eq. (2.3.1) we can factorize the diagrams into leptonic (L) and hadronic (H) tensors as

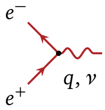
$$\sigma_p(x) \propto L_{\mu\nu} \frac{1}{q^4} H_p^{\mu\nu}(x), \qquad (2.3.3)$$

where

$$L_{\mu\nu} \equiv$$



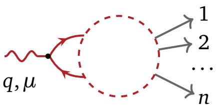
$$\Bigg($$



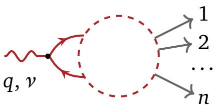
$$\Bigg)^*, \qquad (2.3.4)$$

and

$$H_p^{\mu\nu}(x) \equiv \sum_n \int \mathrm{dPS}_n(x) \Bigg(\sum_{\text{diagrams}}$$



$$\Bigg) \Bigg(\sum_{\text{diagrams}}$$



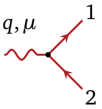
$$\Bigg)^*. \qquad (2.3.5)$$

The reason for this split is that all the QCD corrections along with all α_s dependence are in $H^{\mu\nu}$, while $L_{\mu\nu}$ is just an overall factor. Let us then expand $H^{\mu\nu}$ in α_s as

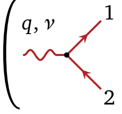
$$H_{p,\mu\nu} \equiv H^{(0)}_{p,\mu\nu} + H^{(1)}_{p,\mu\nu} a_s + H^{(2)}_{p,\mu\nu} a_s^2 + H^{(3)}_{p,\mu\nu} a_s^3 + \mathcal{O}\left(a_s^4\right). \qquad (2.3.6)$$

At the leading order we have

$$H^{(0)}_{q,\mu\nu} = \int \mathrm{dPS}_2(x)$$

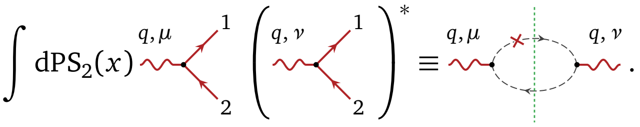


$$\Bigg($$



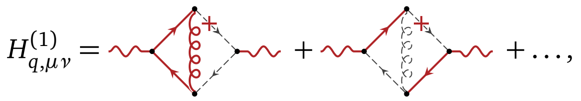
$$\Bigg)^*. \qquad (2.3.7)$$

At this point it is useful to introduce a shortcut diagrammatic notation: we shall denote the final state particles over which the phase-space integration is performed via dashed lines ("cut lines"), optionally with a cutting line drawn through them, and put a cross mark on the line corresponding to p_1, and thus, to $\delta\left(x-2\frac{q\cdot p_1}{q^2}\right)$. In this notation



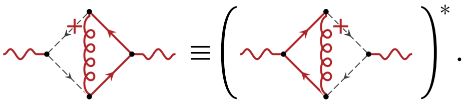
(2.3.8)

At the next order we have



(2.3.9)

and so on. Note that we always depict the complex-conjugated part of the amplitude to the right of the cut line, so that



(2.3.10)

Of course, calculating tensor integrals is less convenient than scalar ones, so we should project $H^{\mu\nu}$ onto some set of scalars. Because $H^{\mu\nu}$ is a Lorentz tensor and the only other Lorentz-covariant objects it can depend on are $g_{\mu\nu}$, q_μ, $p_{1,\mu}$, and the antisymmetric tensor $\epsilon_{\mu\nu\rho\sigma}$, we should expect

$$H^{\mu\nu}=F_1g^{\mu\nu}+F_2q^\mu q^\nu+F_3p_1^\mu p_1^\nu+F_4p_1^\mu q^\nu+F_5q^\mu p_1^\nu+F_6\epsilon_{\mu\nu\rho\sigma}q^\rho p_1^\sigma, \tag{2.3.11}$$

where F_i are some scalar quantities. Due to Ward identities,

$$q_\mu H^{\mu\nu}=q_\nu H^{\mu\nu}=0, \tag{2.3.12}$$

this ansatz can be simplified, and is conventionally written as

$$\begin{aligned}H^{\mu\nu}(x)=&\left(\frac{p_1^\mu q^\nu+q^\mu p_1^\nu}{p_1\cdot q}-g^{\mu\nu}-\frac{q^2}{(p_1\cdot q)^2}p_1^\mu p_1^\nu\right)\frac{q^2}{2\,(p_1\cdot q)}F_T(x)+\\&+\left(p^\mu-\frac{p_1\cdot q}{q^2}q^\mu\right)\left(p_1^\nu-\frac{p_1\cdot q}{q^2}q^\nu\right)\frac{q^4}{p_1\cdot q}F_L(x)+\\&+i\epsilon_{\mu\nu\rho\sigma}p_1^\rho q^\sigma\frac{q^2}{2\,(p_1\cdot q)}F_A(x),\end{aligned} \tag{2.3.13}$$

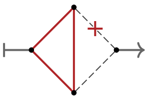
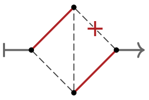
where F_T, F_L, and F_A are the *transverse*, *longitudinal*, and *asymmetric structure functions*, which can be expressed through $H^{\mu\nu}$ using the following projectors:

$$F_T=\frac{2}{d-2}\left(-\frac{p_1\cdot q}{q^2}g^{\mu\nu}-\frac{p_1^\mu p_1^\nu}{p_1\cdot q}\right)H_{\mu\nu}, \tag{2.3.14}$$

$$F_L=\frac{p_1^\mu p_1^\nu}{p_1\cdot q}H_{\mu\nu}, \tag{2.3.15}$$

$$F_A=-\frac{1}{q^2}\frac{2}{(d-2)(d-3)}i\epsilon_{\mu\nu\rho\sigma}p_1^\rho q^\sigma H^{\mu\nu}. \tag{2.3.16}$$

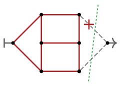
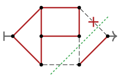
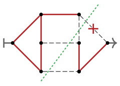
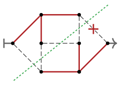
So if one takes the hadronic tensor in terms of Feynman diagrams as given in eq. (2.3.5) or eq. (2.3.9), inserts Feynman rules, and applies these projectors, then F_T, F_L, and F_A would be reduced to a weighted sum of scalar Feynman integrals, e.g.:[7]

$F_T^{(1)} = \mathscr{C}_1$  $+ \mathscr{C}_2$  $+ \ldots.$ (2.3.17)

Unfortunately among these Feynman integrals there are many which values are not known (or, rather, have not been known until this work).

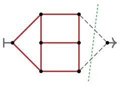
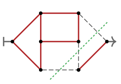
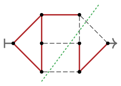
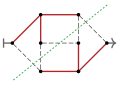
2.4. Which integrals are needed?

To calculate σ_p up to α_s^3 we need $F_T^{(3)}$. The corresponding integrals are semi-inclusive cuts of massless 4-loop 2-point functions. Several kinds of them are needed:

- integrals with 2 cut lines and 3 loop integrations such as ,
- 3 cut lines and 2 loop integrations like ,
- 4 cut lines and 1 loop integration like ,
- and 5 cut lines with no loop integration like .

The current state of the art is that semi-inclusive cuts of 3-loop propagators are known from [69], and semi-inclusive integrals with 2 cut lines can be taken from the 3-loop form factor integrals calculated in [70, 71, 72]; 3-, 4-, and 5-particle cut integrals we need to complete by ourselves (this will be done in Chapter 10).

To calculate these integrals we follow the suggestion of [28]: it is possible to obtain them via the differential equations method, if one knows the values of their fully inclusive versions, i.e. fully inclusive cuts of 4-loop propagators, to fix the boundary conditions. In other words, integrals like

, , , and . (2.4.1)

Unfortunately, the full set of values of these integrals is also something we need to derive ourselves—and we have, in [35, 36] (this calculation will be described in detail in Chapter 9).

Before we can proceed with the calculation of the inclusive and then semi-inclusive cut integrals, we must start from the beginning and review the techniques of calculation of Feynman integrals in general, concentrating on the methods applicable to our calculation.

[7] We do not describe these steps in detail here; a computer implementation is available in ALIBRARY.

3. Feynman integrals

A *Feynman integral* in d space-time dimensions (without cuts) has the general form

$$I_{\nu_1\dots\nu_N} = \int \frac{\mathrm{d}^d l_1}{(2\pi)^d}\cdots\frac{\mathrm{d}^d l_L}{(2\pi)^d}\frac{1}{D_1^{\nu_1}\cdots D_N^{\nu_N}}, \tag{3.0.1}$$

where l_i are the L loop momenta, D_i are the N denominators, and ν_i are the corresponding denominator powers (also called "the indices"). The denominators in general are quadratic polynomials of the loop momenta, normally having the form

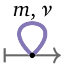
$$D_k \equiv \left(a_i^{(k)} l_i + b_i^{(k)} p_i\right)^2 - m_k^2 + i0, \tag{3.0.2}$$

where p_i are the E external momenta, m_k^2 are the mass terms, and $a_i^{(k)}$ and $b_i^{(k)}$ are some constants. The infinitesimal positive imaginary term $+i0$ is here to unambiguously define integration over poles of the integrand (zeros of D_k).

Graphically Feynman integrals can be represented as *Feynman graphs*: directed graphs with each edge carrying some momenta k and corresponding to a denominator $D = k^2 - m^2 + i0$, and each vertex implying conservation of momenta, with the total momenta flowing into any edge being zero. See eq. (3.1.1), eq. (3.2.7), and eq. (3.5.4) for some examples.

3.1. Direct integration: massive vacuum bubble

The simplest non-trivial Feynman integral is the massive one-loop vacuum bubble (here given with arbitrary denominator power ν):



$$\equiv \int \frac{\mathrm{d}^d l}{(2\pi)^d}\frac{1}{(l^2 - m^2 + i0)^\nu}. \tag{3.1.1}$$

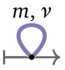
The integrand here is highly symmetric: it remains invariant under any Lorentz rotation of l. To exploit this fact we can introduce the components of l as $l \equiv (E, \vec{p})$ and rewrite $\mathrm{d}^d l$ in spherical coordinates using eq. (A.0.2) as

$$\mathrm{d}^d l = \mathrm{d}E\,\mathrm{d}p\,p^{d-2}\Omega_{d-2}, \tag{3.1.2}$$

where $p \equiv |\vec{p}|$, and Ω_k is the surface area of a k-sphere as given by eq. (A.0.3). The resulting integral,

$$\int \frac{\Omega_{d-2}}{(2\pi)^d}\frac{\mathrm{d}E\,\mathrm{d}p\,p^{d-2}}{(E^2 - p^2 - m^2 + i0)^\nu}, \tag{3.1.3}$$

can then be taken routinely.[1] Its value is



$$\equiv (-1 - i0)^\nu \frac{i\pi^{\frac{d}{2}}}{(2\pi)^d}\frac{\Gamma\left(\nu - \frac{d}{2}\right)}{\Gamma(\nu)}\left(m^2 - i0\right)^{\frac{d}{2}-\nu}, \tag{3.1.4}$$

[1] That is, with MATHEMATICA.

where the factor $(-1-i0)^{\nu}$ can also be written as $e^{-i\pi\nu}$. Note that this result remains correct and unambiguous even if m^2 is negative, or ν is not an integer—this is the whole point of the $\pm i0$ prescriptions.

Because the integrals we will be interested in are less symmetric than this one, calculating them will require applying more involved techniques. Let us review some of them.

3.2. Feynman parameterization

A Feynman integral of the form eq. (3.0.1) can be rewritten into the *Feynman parameterization* as

$$I=\left((-1-i0)^{\frac{d}{2}}\frac{i\pi^{\frac{d}{2}}}{(2\pi)^{d}}\right)^{L}\frac{\Gamma\left(\nu-L\frac{d}{2}\right)}{\Gamma(\nu_1)\cdots\Gamma(\nu_N)}\int\left(\prod_{i=1}^{N}\mathrm{d}x_i x_i^{\nu_i-1}\right)\delta\left(1-\sum_{i=1}^{N}x_i\right)\mathcal{U}(x)^{\nu-(L+1)\frac{d}{2}}\,\mathcal{F}(p,x)^{L\frac{d}{2}-\nu},\tag{3.2.1}$$

where $\nu\equiv\nu_1+\cdots+\nu_N$, x_i are the Feynman parameters, and $\mathcal{U}(x)$ and $\mathcal{F}(p,x)$ are homogeneous polynomials of degrees L and $L+1$ in x_i, with $\mathcal{F}$ additionally depending on the external kinematic invariants $p_i\cdot p_j$.

The fact that the space-time dimension d enters this parameterization as just another variable is, mathematically speaking, what allows us to extend the domain of d from integers to general real or even complex values; in that sense this parameterization can be viewed as the definition of Feynman integrals in d dimensions.

This parameterization will allow us to calculate some simpler integrals directly, as well as to recognize integral symmetries, and detect scaleless integrals.

3.2.1. The Feynman trick

The first step to obtain the form eq. (3.2.1)i s to rewrite the denominators of eq. (3.0.1) using the *Feynman trick*:

$$\frac{1}{D_1D_2}=\int_0^1\mathrm{d}x\frac{1}{(D_1x+D_2\,(1-x))}=\int_0^1\mathrm{d}x\int_0^1\mathrm{d}y\frac{\delta(1-x-y)}{(D_1x+D_2y)}.\tag{3.2.2}$$

Differentiating both sides of this formula by D_1 and D_2 enough times, one obtains the same relation for arbitrary powers,

$$\frac{1}{D_1^nD_2^m}=\frac{(n+m-1)!}{(n-1)!\,(m-1)!}\int_0^1\mathrm{d}x\,x^{n-1}\int_0^1\mathrm{d}y\,y^{m-1}\frac{\delta(1-x-y)}{(D_1x+D_2y)^{n+m}},\tag{3.2.3}$$

and applying it to a sequence of denominators, we have the general Feynman trick:

$$\frac{1}{D_1^{\nu_1}\cdots D_N^{\nu_N}}=\frac{\Gamma(\nu)}{\Gamma(\nu_1)\cdots\Gamma(\nu_N)}\int_0^1\left(\prod_{i=1}^{N}\mathrm{d}x_i\,x_i^{\nu_i-1}\right)\frac{\delta(1-x_1-\cdots-x_N)}{(D_1x_1+\cdots+D_Nx_N)^{\nu}},\tag{3.2.4}$$

where $\nu\equiv\nu_1+\cdots+\nu_N$.

Integration region

Note that although the integration in eq. (3.2.4) ostensibly has the limits 0 and 1, the actual integration region is defined by the δ-function: it is a simplex, with each x_i bounded from below by $x_i \geq 0$, and from above by $x_1 + \cdots + x_N = 1$.

Also keep in mind that

$$\int_0^1 \mathrm{d}x_N \, \delta(1 - x_1 - \cdots - x_N) = \theta(1 - x_1 - \cdots - x_{N-1}) \, \theta(x_1 + \cdots + x_{N-1}), \tag{3.2.5}$$

and thus, even after one of the parameters, e.g. x_N , is integrated out, the integration region does not become a cube, because the condition $0 \leq x_1 + \cdots + x_{N-1} \leq 1$ remains.

3.2.2. Resolving the loop integration

After the Feynman trick is applied to eq. (3.0.1), we can proceed to resolve all the l_i integration, one by one. To this end, for every l_i one should:

1. Express the denominator as $A l_i^2 + l_i{\cdot}B + C$.
2. Complete the squares, rewriting that as $A\left(l_i + \frac{B}{2A}\right)^2 + C - \frac{B^2}{4A}$; and shift l_i by $-2A$ so that denominator would turn into just $A l_i^2 + C - \frac{B}{4A}$.
3. Observe that the integral has now taken the form of the massive vacuum bubble integral from eq. (3.1.1) with $-\frac{C}{A} + \frac{B}{4A^2}$ playing the role of m^2, and so eq. (3.1.4) can be used to evaluate it.
4. In summary, we have

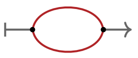
$$\int \frac{\mathrm{d}^d l_i}{(2\pi)^d} \frac{1}{\left(A l_i^2 + l_i{\cdot}B + C + i0\right)^\nu} = (-1 - i0)^{\frac{d}{2}} \frac{i\pi^{\frac{d}{2}}}{(2\pi)^d} \frac{\Gamma\left(\nu - \frac{d}{2}\right)}{\Gamma(\nu)} \frac{(A + i0)^{\nu - d}}{\left(CA - \frac{1}{4}B^2 + i0 \, \mathrm{sgn} A\right)^{\nu - \frac{d}{2}}}, \tag{3.2.6}$$

   where we have supplied the $i0$ prescription for A for completeness only; A is always non-negative in practice.

Interestingly, in the second and further iterations of this procedure the denominator of eq. (3.2.6) will be divisible by the numerator of the previous iteration, canceling it. After the final iteration, the result will take the form of eq. (3.2.1), with the numerator becoming $\mathscr{U}(x)$ and the denominator becoming $\mathscr{F}(p,x)$.

3.2.3. Example: one loop self-energy

As an example, let us calculate the one loop self-energy integral ("the bubble"),



$$\equiv \int \frac{\mathrm{d}^d l}{(2\pi)^d} \frac{1}{\left((q-l)^2 + i0\right)(l^2 + i0)}. \tag{3.2.7}$$

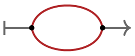
First, using the Feynman trick we have

$$\int \frac{\mathrm{d}^d l}{(2\pi)^d} \int_0^1 \mathrm{d}x_1 \int_0^1 \mathrm{d}x_2 \frac{\delta(1 - x_1 - x_2)}{((q^2 - 2q{\cdot}l + l^2)x_1 + l^2 x_2 + i0)^2}. \tag{3.2.8}$$

Next, integrating out l we get

$$\int_0^1 \mathrm{d}x_1 \int_0^1 \mathrm{d}x_2 \, (-1-i0)^{\frac{d}{2}} \frac{i\pi^{\frac{d}{2}}}{(2\pi)^d} \frac{\Gamma\big(2-\frac{d}{2}\big)}{\Gamma(2)} \frac{(x_1+x_2)^{2-d}}{(x_1x_2q^2+i0)^{2-\frac{d}{2}}} \delta(1-x_1-x_2)\,. \tag{3.2.9}$$

The integration over x_2 resolves the δ-function, and the remaining integral over x_1 can be taken via the hypergeometric function ${}_2F_1$. Assuming $d>2$ the result reads



$$= \frac{i\pi^{\frac{d}{2}}}{(2\pi)^d} \frac{\Gamma^2\big(\frac{d}{2}-1\big)\Gamma\big(2-\frac{d}{2}\big)}{\Gamma(d-2)} \big(-q^2-i0\big)^{\frac{d}{2}-2}. \tag{3.2.10}$$

3.2.4. The Lee-Pomeransky representation

A useful variation of the Feynman representation from eq. (3.2.1) is introduced in [73]: it transforms the parameterization so that the $\mathscr{U}$ and $\mathscr{F}$ polynomials only enter it as a single expression,

$$\mathscr{G}(p,x) \equiv \mathscr{U}(x) + \mathscr{F}(p,x)\,. \tag{3.2.11}$$

This is the *Lee-Pomeransky polynomial*. To obtain the corresponding representation we first need to transform eq. (3.2.1) by applying the hypergeometric integral,

$$\mathscr{U}^{\alpha+k}\mathscr{F}^{-\alpha} = \frac{\Gamma(-k)}{\Gamma(\alpha)\Gamma(-k-\alpha)} \int_0^\infty \mathrm{d}s\, s^{\alpha-1} (\mathscr{U}+s\,\mathscr{F})^k\,, \tag{3.2.12}$$

and then to remove the s variable from inside the parentheses by noting that because $\mathscr{U}$ and $\mathscr{F}$ are homogeneous polynomials of orders L and $L+1$ in x_i , they must scale as

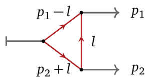
$$\mathscr{U}\Big(\frac{x}{s}\Big) = s^{-L}\mathscr{U}(x) \qquad \text{and} \qquad \mathscr{F}\Big(\frac{x}{s}\Big) = s^{-L-1}\mathscr{F}(x)\,, \tag{3.2.13}$$

so making a change of variables $x \to x/s$ transforms $(\mathscr{U}+s\,\mathscr{F})^k$ into $s^{-Lk}(U+F)^k$. Moreover, under this change of variables the δ-function in eq. (3.2.1) becomes $\delta\big(1-\sum_i x_i/s\big)$, which allows one to immediately integrate out s. The result then reads

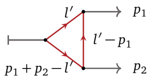
$$I = \left((-1-i0)^{\frac{d}{2}} \frac{i\pi^{\frac{d}{2}}}{(2\pi)^d}\right)^L \frac{\Gamma\big(\frac{d}{2}\big)}{\Gamma\big((L+1)\frac{d}{2}-\nu\big)} \frac{1}{\Gamma(\nu_1)\cdots\Gamma(\nu_N)} \left(\prod_{i=1}^N \int_0^1 \mathrm{d}x_i\, x_i^{\nu_i-1}\cdot\right) \mathscr{G}(p,x)^{-\frac{d}{2}}\,. \tag{3.2.14}$$

3.3. Integral symmetries

Many Feynman integrals that have different integrands when written down in form eq. (3.0.1) are in fact identical, and can be made expressly so by a change in loop momenta variables. For example consider the following two integrals:[2]



$$\equiv \int \frac{\mathrm{d}^d l}{(2\pi)^d} \frac{1}{(l)^2(p_1-l)^2(p_2+l)^2}, \tag{3.3.1}$$

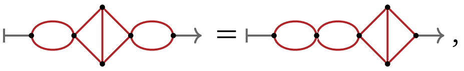


$$\equiv \int \frac{\mathrm{d}^d l}{(2\pi)^d} \frac{1}{(l')^2(p_1-l')^2(p_1+p_2-l')^2}. \tag{3.3.2}$$

[2] The $+i0$ prescription for the denominators is implied here; we have omitted it for brevity.

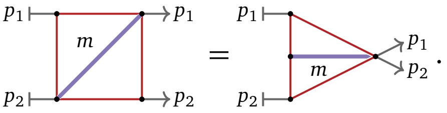
Just from Feynman graphs it can be seen that these integrals should be identical: they only differ in how the loop momenta is assigned to the edges. Indeed, the change of variables $l' \to p_1 - l$ makes the integrands expressly identical.

While in this example the symmetry can be recognized by looking at the graph equivalence of Feynman graphs corresponding to the integral, this is not sufficient in general: there are cases when integrals with different graphs are symmetric too. The simplest and most common such example corresponds to the bubble insertion reordering:



(3.3.3)

a more complex example is given in [74]:



(3.3.4)

To systematically recognize these symmetries we can turn to analyzing the Feynman parameterization: eq. (3.2.1) is invariant under the permutation of the Feynman parameters x_i, so two integrals are symmetric if their $\mathscr{U}$ and $\mathscr{F}$ polynomials can be made identical by any such permutation. Practically, instead of analyzing $\mathscr{U}$ and $\mathscr{F}$ separately, the symmetry of just the Lee-Pomeransky polynomials $\mathscr{G} = \mathscr{U} + \mathscr{F}$ can be considered.

To continue the example of the integrals above, assuming $p_1^2 = p_2^2 = 0$, their $\mathscr{G}$ polynomials are

$$x_1 + x_2 + x_3 + 2p_1{\cdot}p_2 x_2 x_3 \qquad \text{and} \qquad x_1' + x_2' + x_3' + 2p_1{\cdot}p_2 x_1' x_3'. \tag{3.3.5}$$

One can see that these polynomials become identical by substituting

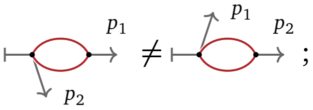
$$\{x_1, x_2, x_3\} \leftrightarrow \{x_2', x_1', x_3'\}. \tag{3.3.6}$$

This proves that the integrals are identical. Moreover, the found substitution also helps to figure out which momenta mapping is needed to make the integrands identical: it implies that the first denominator of the first integral corresponds to the second denominator of the second integral, and so on. Overall the substitution gives us the following equations:

$$\begin{aligned} (l)^2 &= \left(p_1 - l'\right)^2, \\ (p_1 - l)^2 &= \left(l'\right)^2, \\ (p_2 + l)^2 &= \left(p_1 + p_2 - l'\right)^2. \end{aligned} \tag{3.3.7}$$

This is a quadratic equation system with only one solution: $l' = p_1 - l$, which is exactly the momenta mapping we expected. This solution can be found by noting that the system reduces to a linear system with unknown signs (i.e. $\pm l = p_1 - l', \ldots$), so one can try to take the equations one by one recursively, setting the sign of each to first + and then −, and backtrack if the currently assembled linear system has no solutions. In practical problems this works sufficiently fast.

Note however that sometimes it may happen that a system like eq. (3.3.7) has no solutions. This means that although a symmetry was identified in the Feynman parameter space, it has no equivalent in the momenta space. Such a situation can happen when the kinematics of the problem provides avenues for symmetries not present in the general kinematics. For example the following two integrals appear in the subsectors of the triangle from eq. (3.3.1), and are generally speaking different:



(3.3.8)

if, however, $p_1^2 = p_2^2$, they become identical, and there is no loop momenta shift that would make their integrands identical (if there was, it would have been valid when $p_1^2 \neq p_2^2$ too). In such cases one can still use the relation in the Feynman parameter space eq. (3.3.6) to establish the equivalence.

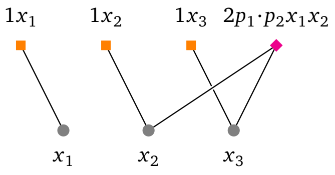
The question then lies in how to find substitutions like eq. (3.3.6): there are $N!$ of them in total, where N is the number of the denominators, so trying all permutations quickly becomes impractical. In fact, finding a permutation of x_i under which two polynomials become the same is equivalent to finding graph isomorphisms, a problem for which no polynomial-time solution is known. Still, methods of finding graph isomorphisms are well-studied and software exists that performs well on variety of graphs: Nauty and Traces [75], Saucy [76], and Bliss [77] are some of the commonly used libraries that given a graph will produce its canonical labeling. So a practical strategy can be just that:

1. Construct such a graph that relabelings that leave it unchanged are isomorphic to x_i permutations that leave $\mathcal{G}$ unchanged.
2. Find the canonical labeling of that graph.
3. Turn the labeling into the canonical x_i permutation, and thus, canonical $\mathcal{G}$ polynomial.

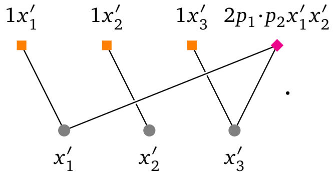
The construction of a graph corresponding to the polynomial is simple:

1. For each x_i add a vertex.
2. For each term $C x_1^{k_1} x_2^{k_2} \ldots$ in $\mathcal{G}$ add a vertex with a different color for each unique C.
3. For each $x_i^{k_i}$ in each term $C x_1^{k_1} x_2^{k_2} \cdots$ add an edge of color k_i between the vertex corresponding to x_i and the vertex corresponding to the whole term.

To illustrate, the polynomials from eq. (3.3.5) correspond to the following graphs:

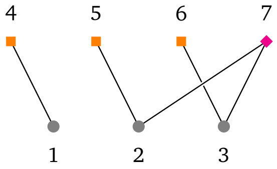


and

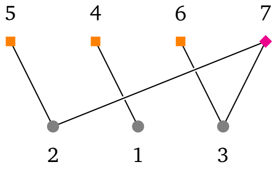


. (3.3.9)

A canonical labeling procedure might label these two graphs as[3]



and



. (3.3.10)

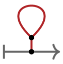
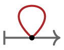
Under this labeling both graphs have the exact same set of edges: $\{1 \longleftrightarrow 4,\ 2 \longleftrightarrow 5,\ \ldots\}$, and thus are identical. After verifying the equality, we can use the labeling to immediately deduce the mapping between x_i and x_i' from eq. (3.3.6), and then solve the corresponding system of equations like eq. (3.3.7) to find the momenta mapping.

We have implemented this strategy in the Feynson tool, see Chapter 4 for its description.

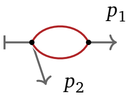
[3] Note that for example Nauty and Traces do not support graphs with colored edges, only colored vertices; this however is not a problem because a colored graph can be transformed into an equivalent uncolored one. The suggested construction is to make multiple layers (copies) of all vertices with each layer having a distinct set of vertex colors; connect all copies of a given vertices across the layers by a sequence of edges; and finally replicate an edge of color $k = 1, 2, \ldots$ in layer $i = 1, 2, \ldots$ if $\lfloor k/2^{i-1} \rfloor = 1 \bmod 2$. For a graph with n distinct edge colors, the total of $\log \lceil \log_2(n+1) \rceil$ layers are needed for this construction.

3.4. Scaleless integrals

A Feynman integral is *scaleless* if it depends on no external mass scale. Such integrals are zero in dimensional regularization. Examples of these are massless tadpoles and snails,

 $= 0,$  $= 0,$ (3.4.1)

also propagators (2-point functions) with zero momentum flowing through them,



$$, \qquad \text{if } p_1^2 = 0. \tag{3.4.2}$$

Naturally, any integral corresponding to a diagram that has one of these as a subdiagram is also zero. A practical condition for detecting zero integrals is given in [78]: one can start from the Feynman parameterization of eq. (3.2.1), rescale

$$x_i \to x_i + \alpha k_i x_i, \tag{3.4.3}$$

where α is an infinitesimal parameter and k_i are some constants; then if such k_i exist that $\mathcal{G}$ scales as

$$\mathcal{G}(x_i + \alpha k_i x_i) = (1+\alpha)\,\mathcal{G}(x_i), \tag{3.4.4}$$

then the whole integral will scale as

$$I \to \left(1 + \alpha\left(\sum_i k_i - \frac{d}{2}\right)\right) I, \tag{3.4.5}$$

and therefore will be scaleless. After rewriting eq. (3.4.4) as

$$\sum_i k_i x_i \frac{\partial}{\partial x_i} \mathcal{G}(x) = \mathcal{G}(x), \tag{3.4.6}$$

we have the scaleless integral criteria: if there exist such k_i that eq. (3.4.6) becomes an identity, then the corresponding integral is zero. To check this one only needs to collect the coefficients next to all x_i combinations in eq. (3.4.6), and demanding that each become zero; this results in a linear system of equations in k_i, which can be solved via standard methods.

The FEYNSON tool from Chapter 4 contains an implementation of this condition for zero integral detection.

3.5. Integration-by-parts relations

Let us take a look at a family of Feynman integrals with L loop momenta l_i, and E external momenta p_i. Let us define it by

$$I_{\nu_1,\ldots,\nu_N} \equiv \int \frac{\mathrm{d}^d l_1}{(2\pi)^d} \cdots \frac{\mathrm{d}^d l_L}{(2\pi)^d} \frac{1}{D_1^{\nu_1} \cdots D_N^{\nu_N}}, \tag{3.5.1}$$

where $D_i \equiv a_{jk}^{(i)} l_j \cdot l_k + b_{jk}^{(i)} l_j \cdot p_k - m_i^2 + i0$ are the denominators, and ν_i are their powers, also called "indices".

The integrals in such a family are not all linearly independent: there are linear relations between these integrals, and in fact the linear basis for them is known to be finite.

One class of such relations are the so-called *integration-by-parts relations* (IBP) [79]. We can derive them succinctly by noting that the integration volume for l_i is unbounded, so changing it by a finite amount should not change the integral (if the integral converges at all). Let us then shift one of the loop momenta, l_k, by αv, where v is some arbitrary vector, and require invariance. In the limit $\alpha \to 0$ the invariance condition then becomes just a derivative:[4]

$$\lim_{\alpha\to 0}\frac{\mathrm{d}}{\mathrm{d}\alpha} I(l_k \to l_k + \alpha v) = \int \frac{\mathrm{d}^d l_1}{(2\pi)^d}\cdots\frac{\mathrm{d}^d l_L}{(2\pi)^d}\frac{\partial}{\partial l_k^\mu}\frac{v^\mu}{D_1^{\nu_1}\cdots D_N^{\nu_N}} = 0. \tag{3.5.2}$$

The derivative acting on the denominators produces two kinds of structures: denominators D_i raised to powers different than the original ν_i, and additional scalar product terms in the numerator. The trick is to express these scalar products in terms of the denominators D_i: if this is possible, then the whole eq. (3.5.2) will become a linear combination of integrals in the same family but with different indices,

$$\sum_k C_k I_{\nu_1+n_1^{(k)},\dots\nu_N+n_N^{(k)}} = 0, \tag{3.5.3}$$

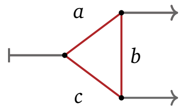
where C_i are some coefficients that may depend on the external kinematic invariants $p_i \cdot p_j$ and d, and $n_i^{(k)} \in \{0, \pm 1\}$.

The relation eq. (3.5.2) holds for all k, and the vector v can be chosen arbitrary: any of l_i and p_i can be chosen; in total this gives us $L\,(L+E)$ relations. These relations can be solved. In fact even though there is an infinite number of integrals $I_{\nu_1,\dots,\nu_N}$, these relations allow linearly expressing all of them through a number of basis integrals called *the master integrals*. This basis is known to always be finite [80].

3.5.1. Example: the triangle topology

As an example let us use the following family of integrals:

$$I_{a,b,c} =$$



$$\equiv \int \frac{\mathrm{d}^d l}{(2\pi)^d}\frac{1}{\underbrace{l^{2a}}_{D_1^a}\,\underbrace{(p_1-l)^{2b}}_{D_2^b}\,\underbrace{(p_1+p_2-l)^{2c}}_{D_3^c}}, \tag{3.5.4}$$

where p_1 and p_2 are external momenta, both on-shell ($p_1^2 = p_2^2 = 0$), and l is the loop momentum. Putting this definition into eq. (3.5.2), and using

$$v^\mu \frac{\partial}{\partial l^\mu}\frac{1}{(k^2)^n} = -n\frac{1}{(k^2)^{n+1}}2v^\mu\frac{\partial k^\nu}{\partial l^\mu}k_\nu, \tag{3.5.5}$$

the IBP relation for the general v becomes

$$0 = \int \frac{\mathrm{d}^d l}{(2\pi)^d}\frac{1}{l^{2a}(p_1-l)^{2b}(p_1+p_2-l)^{2c}}\left(\frac{\partial k^\mu}{\partial l^\mu} - 2a\frac{v\cdot l}{l^2} - 2b\frac{-v\cdot(p_1-l)}{(p_1-l)^2} - 2c\frac{-v\cdot(p_1+p_2-l)}{(p_1+p_2-l)^2}\right). \tag{3.5.6}$$

[4] Note that l_k itself can be chosen for v, which corresponds to rescaling of the loop momenta. This case is special in that $\mathrm{d}^d l_k$ changes too, and eq. (3.5.2) is written in a way to handle this case correctly: v^μ is inside the differentiation operation for this reason.

Next, we would like to substitute p_1, p_2, and l for v into this equation, express the scalar products appearing in the numerators via the denominators D_i, and finally obtain the relations between $I_{a,b,c}$.

All the possible scalar products involving the loop momentum l can be expressed via the denominators as follows:

$$p_1{\cdot}l = -\frac{1}{2}(p_1-l)^2+\frac{1}{2}l^2 = -\frac{1}{2}D_2+\frac{1}{2}D_1, \tag{3.5.7}$$

$$p_2{\cdot}l = -\frac{1}{2}(p_1+p_2-l)^2+\frac{1}{2}(p_1-l)^2+p_1{\cdot}p_2 = -\frac{1}{2}D_3+\frac{1}{2}D_2+p_1{\cdot}p_2, \tag{3.5.8}$$

$$l{\cdot}l = l^2 = D_1. \tag{3.5.9}$$

Using these relations, and substituting p_1 for v we get the first IBP relation:

$$\begin{aligned} 0 = &-2a\left(-\frac{1}{2}I_{a+1,b-1,c}+\frac{1}{2}I_{a,b,c}\right)- \\ &-2b\left(-\frac{1}{2}I_{a,b,c}+\frac{1}{2}I_{a-1,b+1,c}\right)- \\ &-2c\left(-p_1{\cdot}p_2 I_{a,b,c+1}-\frac{1}{2}I_{a,b-1,c+1}+\frac{1}{2}I_{a-1,b,c+1}\right). \end{aligned} \tag{3.5.10}$$

Substituting p_2 for v we get the second one:

$$\begin{aligned} 0 = &-2a\left(-\frac{1}{2}I_{a+1,b,c-1}+\frac{1}{2}I_{a+1,b-1,c}+p_1{\cdot}p_2 I_{a+1,b,c}\right)- \\ &-2b\left(-\frac{1}{2}I_{a,b+1,c-1}+\frac{1}{2}I_{a,b,c}\right)- \\ &-2c\left(-\frac{1}{2}I_{a,b,c}+\frac{1}{2}I_{a,b-1,c+1}\right). \end{aligned} \tag{3.5.11}$$

And, substituting l for v we get the third one:

$$\begin{aligned} 0 = &+dI_{a,b,c}-2aI_{a,b,c}- \\ &-2b\left(\frac{1}{2}I_{a,b,c}+\frac{1}{2}I_{a-1,b+1,c}\right)- \\ &-2c\left(\frac{1}{2}I_{a,b,c}-p_1{\cdot}p_2 I_{a,b,c+1}+\frac{1}{2}I_{a-1,b,c+1}\right). \end{aligned} \tag{3.5.12}$$

3.5.2. Lorentz invariance relations

A related class of relation are the *Lorentz invariance relations* [81]; these follow the same idea as IBP, except that instead of translations the integrals are required to be invariant under arbitrary Lorenz boosts of the external momenta:

$$\lim_{\alpha\to 0}\frac{\mathrm{d}}{\mathrm{d}\alpha}I\big(p_i^\mu \to p_i^\mu+\alpha\omega^\mu_\nu p_i^\nu\big) = \omega^\mu_\nu\left(\sum_i p_i^\mu\frac{\partial}{\partial p_i^\nu}\right)I = 0, \tag{3.5.13}$$

where ω^μ_ν is an arbitrary antisymmetric tensor. Such tensors can be constructed from pairs of momenta p_i and p_j as

$$\omega^\mu_\nu = p_i^\mu p_{j\nu} - p_i^\nu p_{j\mu}, \tag{3.5.14}$$

with i and j being arbitrary, giving a total of $E\,(E-1)\,/2$ combinations of ω^{μ}_{ν}, and the total of $E^2\,(E-1)\,/2$ of Lorentz invariance relations.

This class of relations is still used in practice, even though it is known to be linearly dependent on the IBP relations [82].

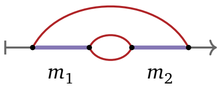
3.5.3. IBP integral family construction

To derive concrete IBP relations like eq. (3.5.10) from the general eq. (3.5.2) we must be able to uniquely express every scalar product involving the loop momenta l_i in terms of the denominators D_i. This means that:

1. Denominators of an IBP integral family must be linearly independent.
2. There must as many of them as there are $l_i{\cdot}l_j$ and $l_1{\cdot}p_j$ combinations: $L\,(L+1)\,/2+LE$ in total.

Neither of these conditions is automatically satisfied by the integrals coming directly from Feynman diagrams. For the second condition, note that a diagram with L loops and $E+1$ legs will have up to $E-2+3L$ denominators (and exactly that many if all vertices have three edges); if that is less than $L\,(L+1)\,/2+LE$, auxiliary denominators must be added to complete the integral family. There is a considerable freedom in choosing how these look like. Because the total time needed to solve IBP relations depends on this choice, it may be profitable to try different options; a heuristic such as “select the set that minimizes the number of terms in IBP relations” can be used too.

For the first condition, keep in mind that some Feynman diagrams naturally result in integrals with linearly dependent denominators. Diagrams with masses are one such case; for example:

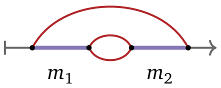


$$= \int \frac{1}{l^2-m_1^2}\frac{1}{l^2-m_2^2}\cdots, \qquad (3.5.15)$$

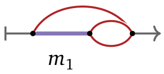
where the denominators $l^2-m_1^2$ and $l^2-m_2^2$ are linearly dependent. This problem is solved by performing *partial fraction decomposition* on the integrands. In the example above it is as simple as substituting

$$\frac{1}{l^2-m_1^2}\frac{1}{l^2-m_2^2} = \frac{1}{m_1^2-m_2^2}\frac{1}{l^2-m_1^2} + \frac{1}{m_2^2-m_1^2}\frac{1}{l^2-m_2^2}, \qquad (3.5.16)$$

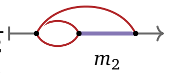
which in diagram form means



$$= \frac{1}{m_1^2-m_2^2}$$



$$+ \frac{1}{m_2^2-m_1^2}$$



$$. \qquad (3.5.17)$$

In other words, a single integral with linearly dependent denominators must be represented as an equivalent combination of multiple integrals from different integral families for IBP relations to be derived.

3.5.3.1. Partial fraction decomposition in the general case

In the most general case when multiple denominators of an integral are linearly dependent, partial fraction decomposition can be done via Leinartas' algorithm [83, 84]:

1. For each integrand term of the form $\mathscr{C} D_1^{-\nu_1} \cdots D_N^{-\nu_N}$, check if there is a linear dependence among the denominators, such that for some vector $\mathbf{A}$ and constant B, $\mathbf{A}_i D_i + B = 0$.

   To do this, decompose D_i into the scalar products $D_i = N_{ij} s_j + M_i$, where $s = \{l_i \cdot l_j, l_i \cdot p_j\}$, then every $\mathbf{A} \in \text{nullspace}\, \mathbb{N}^T$ and $B = -A_i M_i$ will satisfy the dependence condition.

2. If there is a dependence, and $B \neq 0$, multiply the term by a factor of

   $$1 = -\frac{1}{B} \sum_i A_i D_i; \tag{3.5.18}$$

   this way the original term will split into several new terms, each with one of the dependent denominators canceled (at least partially).

3. If there is a dependence, and $B = 0$, choose any denominator D_k such that $A_k \neq 0$, and multiply the term by

   $$1 = -\frac{1}{A_k D_k} \sum_{i \neq k} A_i D_i. \tag{3.5.19}$$

   Note that the index of D_k will necessarily increase in this case.

4. Repeat until no term has linearly dependent denominators.

Note that this algorithm may produce different final expressions depending on the choice of denominators in step 3; sometimes this may be undesirable, and algorithms based on Gröbner bases have been developed that guarantee uniqueness of the final answer [85].

3.5.4. Solving IBP relations: the Laporta algorithm

In some simpler cases the IBP relations can be solved by hand for general indices. For larger problems the best choice is the *Laporta algorithm* [86]:

1. Substitute integer values for the indices ν_i into the IBP relations, obtaining a large linear system with many different $I_{\nu_1 \ldots \nu_N}$.

2. Define an ordering of $I_{\nu_1 \ldots \nu_N}$ from the most "simple" to the most "complex".

   The order is usually defined using the number of the denominators $t \equiv \sum \theta\left(\nu_i - \frac{1}{2}\right)$, the sum of the positive indices $r \equiv \sum \nu_i \theta\left(\nu_i - \frac{1}{2}\right)$, the sum of the negative indices $s \equiv \sum -\nu_i \theta\left(\frac{1}{2} - \nu_i\right)$, etc.

3. Perform Gaussian elimination on the linear system, eliminating the most complex integrals first.

4. A small number of integrals will survive the elimination. These are the *master integrals*. All the others will be expressed as their linear combination.

A lot of effort has been put into software implementations of variations of this algorithm. At the time of writing the fastest publicly available implementations are FIRE6 [87] and KIRA [88]. The biggest difference usage-wise is that FIRE6 constructs IBP relations on the fly until all requested integrals are encountered, while KIRA constructs the whole set of equations up to the specified limit (given in terms of the sum of indices of the integrals), and then solves the whole set.

In our experience FIRE6 turned out to be the faster choice for our problems; it also has kept the memory usage minimal by storing most intermediate results on disk. The downside of FIRE6 it is that on a few occasions it has identified more master integrals than it needed to (compared to KIRA). KIRA on the other hand produces smaller master set sizes and automatically resolves symmetries between integral families (something FIRE6 cannot do). The biggest downside of KIRA is its high memory usage (e.g. 100-200GB or more for a 4-loop semi-inclusive problem); combined with the fact that to obtain the minimal set of master integrals one must increase the set of relations it looks at (by increasing r and s limits) high memory usage prevented us from using it on 4-loop problems.[5]

Other notable IBP software include LITERED [78], which uses heuristics to try finding a solution for general indices without relying on Laporta-style reduction. While this approach does not scale to more complex problems, LITERED contains functions for integral differentiation, Feynman parameterization, and dimensional recurrence construction; the symmetry tables that it constructs are a requirement for FIRE6 usage. Another notable IBP solver is FORCER [90], which contains special hand-crafted solutions for massless 2-point functions up to 4 loops, and for that particular problem is unrivaled in performance.

At present there are three major avenues to improve the runtime performance of the Laporta reduction:

1. The first is to perform the reduction using finite field arithmetic (i.e. modulo a prime number) with all variables replaced by numbers—something that can be done much faster than the normal reduction—and then restore the dependence on the variables via the Chinese remainder theorem. FIRE6 can operate in this mode, although it lacks facilities to reconstruct multivariate expression. KIRA starting with version 2.0 [88] also supports this method via the library FIREFLY [91], including full multivariate support. FINITEFLOW [92] provides another implementation of this technique.

   This mode of operation allows for much greater parallelizability of the computation because it requires solving the relations many times with different numbers substituted for the variables, and each such run can be performed completely independently from one another. On the other hand practice shows that the speedup this method achieves depends on the problem, and there are reductions that finish faster using the classical approach.

2. The second is to select a master integral basis such that the coefficients of the final IBP relations become smaller in size, and thus are faster to operate on. In particular, it is possible to select a basis such that the dependence on the dimensional variable d in the denominators is factorized—a *d-factorizing basis* [93, 94].

   The selection of the master integral basis in general is very important: some tests show that a reduction that takes 30 minutes with one basis can take just 11 seconds with a different one.

3. The third is to partially solve the IBP relations (by e.g. eliminating relations that raise indices) using algebraic geometry methods before running the Laporta algorithm [95, 96, 97, 98, 99].

3.5.5. IBP for phase space integrals

So far we have been considering integrals of the form eq. (3.0.1). Following [100] it is possible to apply IBP reduction to phase-space integrals with minimal changes to the procedure: the idea is that the δ function defining the phase-space is equivalent to a denominator:

$$\delta\big(p^2\big) = \frac{1}{2\pi i}\left(\frac{1}{p^2 - i0} - \frac{1}{p^2 + i0}\right), \tag{3.5.20}$$

so the IBP relations for cut integrals are the same as for the uncut ones. Note that this relation is just one of the representations of the δ function, unrelated to Feynman integrals as such. A denominator standing in place of a δ function is called a *cut denominator*.

[5] This was out experience with Kira 1.2 [89]; we did not test KIRA 2.0 with FIREFLY, which promises significant memory usage reduction.

Solving IBP relations with cut denominators is the same as with normal ones, with two differences: first, because $p^2\delta\big(p^2\big) = 0$, any integral with cut denominator to a positive power can be immediately put to zero. Second, when integral symmetries are calculated care must be taken not to treat cut and normal denominators as interchangeable.[6]

[6] Practically, this difference can be tracked by adding a unique fictitious mass term to cut propagators; this way the symmetry finding algorithm from Section 3.3 can be reused without modifications.

4. Feynson: a tool for zeros & symmetries

A typical workflow for calculating loop corrections to a physical quantity involves generating Feynman diagrams using tools like QGRAF [101], converting diagrams into integrals through Feynman rules (with custom code), sorting the integrals into IBP families (custom code again), running Laporta-style IBP reductions on them, and then finally inserting the values of master integrals (calculated separately). When sorting the integrals into IBP topologies one wants to obtain the minimal amount of topologies covering all the integrals. To do this it is essential to recognize when a loop momentum substitution exists that makes one family of integrals identical to another family, or a subset of it. IBP programs like KIRA and LITERED do provide tools for this: both can find symmetries between families of integrals, but neither is optimized to handle a large number of them: figuring out symmetries between one thousand of 4-loop integral families takes weeks with KIRA 1.2. So, we have developed a separate tool for this task: FEYNSON.

FEYNSON is a command-line tool that finds symmetries between integral families, helping in the process of mapping integrals into IBP topologies. It does so by figuring out the loop momentum substitutions that make the integrands of different families identical if the families are symmetric, and subsets of each other if there is a subset relation. The intended usage is this: for every integral in the problem calculate the set of denominators; pass this set to FEYNSON, and it will output a loop momentum map for each set; after this momentum map is applied symmetric integrals will have identical denominator sets, and the remaining unique denominator sets will define the minimal set of IBP topologies covering the integrals. The technique for this is described in Section 3.3.

As a related function, FEYNSON can also determine which integral families correspond to scaleless integrals, so that these can be removed from further consideration early. The algorithm from Section 3.4 is used here.

FEYNSON is designed to handle a large amount of integral families, so that there would be no need to preprocess data before passing into it. When possible it performs calculations in parallel, and is generally able to handle thousands of families with no difficulty.

FEYNSON is written in C++ using GINAC [102] and NAUTY AND TRACES [75]; it should work on any Unix-like system. One can find its source code at github.com/magv/feynson.

4.1. Command-line usage summary

The syntax for FEYNSON invocation on the command line is:

feynson `[options]` **command** *args* ...

Commands

The **`command`** argument gives the sub-command name. Available sub-commands are:

feynson `[-d]` **`symmetrize`** *spec-file*

Print a list of momenta substitutions that make symmetries between a list of integral families explicit.

These momenta substitutions will make the set of denominators of two integral families exactly match (up to a reordering) if the families are isomorphic, and will make one a subset of the other if one family is isomorphic to a subsector of another family.

The input specification file should be a list of three elements:

1. a list of all integral families, with each family being a list of propagators (e.g. "`(l1+l2)^2`";
2. a list of all loop momenta;
3. a list of external invariant substitution rules, each rule being a list of two elements: a scalar product and its substitution (e.g. "`{q^2, 1} or {p1*p2, s12}`").

For example: "`{ {{(q-l)^2, l^2}, {(q+l)^2, l^2}}, {l}, {} }`".

Each family that can be mapped to (a subsector of) another is guaranteed to be mapped to the first possible family, preferring families that are larger or listed earlier. If the `-d` flag is given, earlier families are preferred irrespective of their size.

Note that if non-trivial invariant substitution rules are supplied, it becomes possible that two families are identical, but no loop momentum substitution exists to map them onto each other. For example, a 1-loop propagator with momentum p_1 is equal to a 1-loop propagator with p_2, but only if $p_1^2 = p_2^2$, in which case no loop momentum substitution can make the integrands identical.

For this reason, it is best to use symmetrize with the invariant substitution rules set to "`{}`", and to fall back to **`mapping-rules`** otherwise.

feynson `[-d]` **`mapping-rules`** *spec-file*

Do the same as **`symmetrize`**, but instead of printing the loop momenta substitutions, produce explicit rules of mapping between families: for each family that is symmetric to another, print "`{fam, {n1, n2, ...}}`", meaning that any integral in this family with indices "`{i_1, i_2, ...}`" is equal to an integral in the family number "`fam`" with indices "`{i_n1, i_n2, ...}`". For unique families, print "`{}`". The families are numbered starting at 1. If a given family is symmetric to a subfamily, some of the n indices will be 0: the convention is that $i_0 = 0$.

feynson `[-s]` **`zero-sectors`** *spec-file*

Print a list of all zero sectors of a given integral family.

The input specification file should be a list of four elements:

1. a list of all propagator momenta (e.g. "`(l1-q)^2`");
2. a list of cut flags, "`0`" for normal propagators, "`1`" for cut propagators;
3. a list of all loop momenta (e.g. "`l1`");
4. and a list of external invariant substitutions (e.g. "`{q^2, 1}`").

For example: "`{ {(q-l)^2, l^2}, {0, 0}, {l}, {{q^2,1}} }`".

The output will be a list of zero sectors, each denoted by an integer $s = 2^{i_1-1} + ... + 2^{i_n-1}$, where i_k are the indices of denominators that belong to this sector (counting from 1).

If the `-s` flag is given, the output will be shortened by only listing the topmost zero sectors: all the remaining zero sectors are their subsectors.

Every sector that is missing a cut propagator of its supersectors will be reported as zero.

feynson `ufx` *spec-file*

Print the Feynman parametrization ($\mathcal{U}$, $\mathcal{F}$, $\{x_i\}$) of an integral defined by a set of propagators.

The input specification file should be a list of three elements:

1. a list of all propagators, e.g. "`(l1-q)^2`";
2. a list of all loop momenta, e.g. "`l1`";
3. and a list of external invariant substitutions, e.g. "`{q^2, 1}`".

For example: "`{ {(q-l)^2, l^2}, {l}, {{q^2,1}} }`".

The output will be a list of three items: the $\mathcal{U}$ polynomial, the $\mathcal{F}$ polynomial, and the list of Feynman parameter variables.

Options

`-j` *jobs* Parallelize calculations using at most this many workers.

`-d` Prioritize families in the definition order, irrespective of size.

`-s` Shorten the output (depending on the command).

`-q` Print a more quiet log.

`-h` Show a help message.

`-V` Print version information.

Arguments

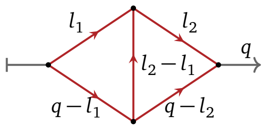
spec-file Filename of the input file, with "`-`" meaning the standard input.

Environment variables

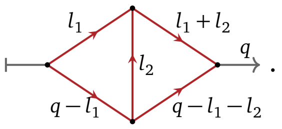
`TMPDIR` Temporary files will be created here.

4.2. Usage example: integral symmetry

As an example, consider the following two equivalent Feynman integral families:



and



(4.2.1)

The first one has propagators $\{l_1, l_2, q-l_1, l_2-l_1, q-l_2\}$, while the second has $\{l_1, q-l_1-l_2, q-l_1, l_2, l_1+l_2\}$. To figure out that these two are equivalent, one can use FEYNSON as follows:[1]

[1] We assume that the reader is familiar with the basics of Unix shell usage.

```
$ cat >families.in <<EOF
{
 {{l1^2, l2^2, (q-l1)^2, (l2-l1)^2, (q-l2)^2},
  {l1^2, (q-l1-l2)^2, (q-l1)^2, l2^2, (l1+l2)^2}},
 {l1, l2},
 {}
}
EOF
$ feynson symmetrize families.in 2>/dev/null
{
 {},
 {{l2,-l1+l2}}
}
```

So the suggestion is to replace l_2 with $l_2 - l_1$ in the second integral family. This will map its set of propagators to $\{l_1, q-l_2, q-l_1, l_2-l_1, l_2\}$, which is exactly the set of propagators of the first family—up to a reorder and a change of sign in $l_2 - l_1$. Thus, both families are equivalent.

Note that FEYNSON can be conveniently used directly from within MATHEMATICA via the `RunThrough` function as follows:

```
In[1]:= RunThrough["feynson symmetrize -", {
 {{l1^2, l2^2, (q-l1)^2, (l2-l1)^2, (q-l2)^2},
  {l1^2, (q-l1-l2)^2, (q-l1)^2, l2^2, (l1+l2)^2}},
 {l1, l2},
 {}
}]
Out[1]= {{}, {{l2,-l1+l2}}}
```

As an alternative to finding the momentum mapping, we can also extract explicit mapping rules between integral families:

```
$ feynson mapping-rules families.in 2>/dev/null
{
 {},
 {1, {1,5,3,4,2}}
}
```

If we will denote the integral family corresponding to the first diagram of eq. (4.2.1) as $A_{\nu_1,\dots,\nu_5}$ and the second as $B_{\nu_1,\dots,\nu_5}$, then this output indicates that the first family (A) is unique, and the second family (B) is symmetric to the first one as

$$B_{\nu_1\nu_2\nu_3\nu_4\nu_5} = A_{\nu_1\nu_5\nu_3\nu_4\nu_2}. \tag{4.2.2}$$

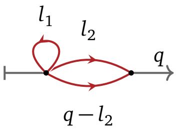
4.3. Usage example: zero sectors

To figure out zero sectors in the same family, we can do this:

```
$ cat >family.in <<EOF
{
 {l1^2, l2^2, (q-l1)^2, (l2-l1)^2, (q-l2)^2},
 {0, 0, 0, 0, 0},
 {l1, l2},
 {}
}
EOF
$ feynson zero-sectors -s family.in 2>/dev/null
```

```
{
 7,
 11,
 13,
 19,
 21,
 22,
 26,
 28
}
```

In binary notation these sector numbers read 00111_2, 01011_2, 01101_2, 10011_2, 10101_2, 10110_2, 11010_2, and 11100_2. As an example, sector 10011_2 (19) corresponds to the 3-rd and 4-th propagators removed, leaving only the set $\{l_1, l_2, q - l_2\}$; this sector can be depicted as



. (4.3.1)

Note the massless self-loop in the diagram: this is the scaleless factor that makes the whole family zero.

5. Dimensional recurrence relations

In [103] it was noted that the operation of shifting the space-time dimension d by ±2 in a parametric representation of Feynman integral like eq. (3.2.1) is expressible in terms of derivatives by the mass terms; this allows expressing an integral in $d+2$ dimensions as a combination of integrals in d dimensions. If one applies this idea to a set of IBP master integrals $\mathbf{I} = \{I_i\}$,, it is further possible to express the resulting integrals as a linear combination of the same set of mater integrals (via IBP relations), resulting in *dimensional recurrence relations* (DRR) of the general form

$$\mathbf{I}(d+2) = \mathbb{M}(d, \mathbf{p})\,\mathbf{I}(d), \tag{5.0.1}$$

where $\mathbb{M}$ is the "lowering" DRR matrix; it depends on d as well as external invariants $p_i \cdot p_j$ and/or masses. A general method of constructing these relations is described in [104] and implemented in LiteRed [78] (see the `LoweringDRR` routine).

The inverse of the "lowering" DRR from eq. (5.0.1) are the "raising" DRR:

$$\mathbf{I}(d-2) = \mathbb{M}(d, \mathbf{p})\,\mathbf{I}(d), \tag{5.0.2}$$

which can also be constructed by LiteRed (via the `RaisingDRR` routine), and are significantly more compact, and thus faster to reduce via IBP.

Dimensional recurrence relations can be solved, and have proven to be a powerful tool in calculating single-scale integrals (those that depend on a single external invariant). A particularly simple case of them—and the one applicable to our integrals of interest—is when $\mathbb{M}$ is triangular (that is, $\mathbb{M}_{ij} = 0$ if $i < j$). DRR matrices necessarily have this structure if there is at most one master integral in an IBP sector;[1] when there are more they can form coupled blocks, making $\mathbb{M}$ block-triangular instead.[2] In the triangular case eq. (5.0.1) for each integral immediately splits into a homogeneous and inhomogeneous part:

$$I_i(d+2) = \mathbb{M}_{ii}(d)\,I_i(d) + \sum_{j<i} \mathbb{M}_{ij}(d)\,I_j(d); \tag{5.0.3}$$

the general solution will then have the form

$$I_i(d) = H_i(d)\,\omega_i(d) + R_i(d), \tag{5.0.4}$$

where,

- $H_i(d)$ is a homogeneous solution, satisfying $H(d+2) = \mathbb{M}_{ii}(d)H_i(d)$;
- $R_i(d)$ is a particular solution, determined by integrals from the lower sectors, $I_{j<i}$;
- $\omega_i(d)$ is an arbitrary periodic function, such that $\omega_i(d+2) = \omega_i(d)$; this function cannot be determined from the DRR relations alone, and needs to be fixed separately.

[1] To be precise, depending on the ordering of $\mathbf{I}$ the DRR matrix might be only similar to a triangular matrix, requiring a similarity transformation before it is properly triangular. In practice if one orders the integrals by their denominator numbers, then the lower sector integrals will necessarily come first, and the triangular form of $\mathbb{M}$ is guaranteed.

[2] For methods for dealing with non-triangular DRR matrices see [105, 106].

If one factorizes the diagonal elements of $\mathbb{M}$ as

$$\mathbb{M}_{ii} = \mathscr{C} \prod_k \left(\frac{d}{2} - a_k \right)^{n_k}, \tag{5.0.5}$$

then the homogeneous solution can be immediately constructed as

$$H_i(d) = \mathscr{C}^{\frac{d}{2}} \prod_k \left\{ \Gamma^{n_k}\left(\frac{d}{2} - a_k\right) \quad \text{or} \quad (-1)^{\frac{d}{2}n_k}\, \Gamma^{-n_k}\left(a_k - \frac{d}{2} + 1\right) \right\}, \tag{5.0.6}$$

where both forms of the factors are equally acceptable; they differ only in the positions of the poles and zeros in d: the first one has poles (or zeros if $n_k < 0$) at $d = 2a_k - 2n$, while the second one has zeros (or poles if $n_k < 0$) at $d = 2\,(a_k + 1) + 2n$. If it is convenient to avoid all poles at higher d then the first form should be chosen; otherwise—the second.

The particular solution $R_i(d)$ can be constructed as an infinite sum,

$$R_i(d) = H_i(d) \left\{ -\sum_{k=0}^{\infty} \quad \text{or} \quad \sum_{k=-\infty}^{-1} \right\} H_i^{-1}(d + 2k + 2) \sum_{j<i} \mathbb{M}_{ij}(d + 2k)\, \mathbf{I}_j(d + 2k), \tag{5.0.7}$$

where the direction of the summation is chosen depending on which one converges. The essential step in solving DRR equations is the evaluation of this sum: a specialized package DREAM [105] exists specifically to evaluate these sums numerically as series in ε with arbitrary precision—as long as the integrals from lower sectors $I_{j<i}(d)$ are known numerically too.[3] Normally this is a quickly converging geometric sum, and a precision of thousands of digits can be easily achieved.

With the $H(d)$ and $R(d)$ constructions being understood, the most difficult part of solving DRR is finding the periodic function $\omega(d)$, which plays the same role as integration constants play in the solution of differential equations.

5.1. Fixing the periodic function

One way of finding $\omega_i(d)$ is the *dimensional recurrence and analyticity method* [107]. Its insight is that if one treats d as a complex variable, by studying the poles of $\omega_i(d)$ and its asymptotic behavior at $\mathrm{Im} d \to \pm\infty$ the possible forms that it is allowed to take can be reduced to a small ansatz, which then can be solved for with just a few bits of additional information.

To see how this is done, let us first rewrite eq. (5.0.3) as

$$\omega_i(d) = H_i^{-1}(d)\, I_i(d) - H_i^{-1}(d) R_i(d). \tag{5.1.1}$$

The poles and asymptotic behavior of $\omega_i(d)$ can then be decided by looking at that of $H_i^{-1}(d)$, $I_i(d)$, and $R_i(d)$. The overall method to calculate $\omega_i(d)$ then has the following steps:

1. Solve all integrals in subsectors, $I_{j<i}$, if any.
2. Choose a semi-open stripe of width 2 in the complex plane, such that $\mathrm{Re}\, d \in (d_0, d_0 + 2]$ or $\mathrm{Re}\, d \in [d_0, d_0 + 2)$, and restrict the analysis to this stripe.

   Because $\omega_i(d)$ is periodic, its behavior must be the same in any such stripe. H_i^{-1}, I_i and R_i on the other hand are not periodic, so it is best to choose such a stripe to minimize the total number of poles these functions have on it.

[3] This is one of the reasons why the method works best for single-scale integrals: in the multi-scale case $R(d)$ becomes a non-trivial function of external kinematic parameters, so that a numerical evaluation—as it is done in DREAM—becomes impossible.

3. Construct the homogeneous solution $H_i(d)$ via eq. (5.0.6), utilizing the freedom of choosing the summation direction in that equation to minimize the set of poles on the chosen stripe.
4. Construct the particular solution $R_i(d)$ via eq. (5.0.7) using DREAM.

   A particularly useful trick here is to note that although DREAM only produces numerical values, it is normally possible to convert them into analytic expressions: if one conjectures that the coefficients of the $R_i(d)$ expansion in ε consist of linear combinations of multiple zeta values (MZVs, see Section 7.2.5),

   $$R_i(d_k-2\varepsilon)=\sum_k \varepsilon^k R_i^{(k)}(d_k)=\sum_k \varepsilon^k \sum_i c_i^{(k)}\zeta_{(i)}, \tag{5.1.2}$$

   where $\zeta_{(i)}$ are linearly independent MZV combinations (listed in e.g. [108]), and $c_i^{(k)}$ are rational, constants, then $c_i^{(k)}$ can be restored from $R_i^{(k)}$ using e.g. the PSLQ algorithm [109],[4] as long as it is known with enough digits of precision.

   "Enough" here is a probabilistic measure: if one wants to restore some value in terms of MZVs of weight 8 and less, and assumes that $c_i^{(k)}$ are rational numbers of less than 100 digits (numerator and denominator combined), then because there are 15 elements in the linear MZV basis of weight 8 and less, at least $100*(15+1)=1600$ digits of precision are needed; if at 2000 digits PSLQ does not change the result, then it can be argued that we get 10^{-400} probability of finding a false relation.
5. Determine the full set of poles $\{d_k\}$and their multiplicities $\{p_k\}$ for $H_i^{-1}(d)$, $I_i(d)$, and $R_i(d)$ on the chosen stripe.
   - For $H_i^{-1}(d)$ this follows directly from eq. (5.0.6).
   - For $R_i(d)$ this can be done semi-analytically: by evaluating the sum eq. (5.0.7) numerically with DREAM for many d, plotting it, and noting any regions where it diverges. Experience tells us that the poles will be located at d being small rational numbers, and the number of them will vary from zero (for simpler integrals) up to a dozen.
   - For $I_i(d)$ an alternative method of (numerical) evaluation is needed. Direct Monte-Carlo integration can be used for simpler integrals, sector decomposition with FIESTA [110] and/or PYSECDEC [111] can be used too. Because these methods of evaluation take more time than DREAM, it is best to choose the stripe to avoid $I_i(d)$ poles in particular. A useful observation for this is noting that infrared poles (those that correspond to soft and collinear singularities) disappear completely at sufficiently high d values, so choosing $d_0=6$ or $d_0=8$ might remove all of them. Conversely, at low enough d the UV poles disappear, so choosing negative d_0 may be advantageous.
6. Determine the asymptotic behavior of $|\omega_i(d)|$ at $\operatorname{Im} d\to\pm\infty$ from the ingredients in eq. (5.1.1).
   - The behavior of $I_i(d)$ can be obtained via a parametric representation. Feynman parameterization of eq. (3.2.1) works for integrals without cuts. To use it note that d enters that equation in only two kinds of structures: the Gamma functions $\Gamma(\alpha+\beta d)$ and powers $\alpha^{\beta d}$. The asymptotic behavior of $\Gamma(x)$ follows from the Stirling formula; for $\operatorname{Im} x\to\infty$ we have

     $$|\Gamma(x)|\approx\sqrt{2\pi}e^{-\frac{\pi}{2}|\operatorname{Im} x|}\,|\operatorname{Im} x|^{\operatorname{Re} x-\frac{1}{2}}. \tag{5.1.3}$$

     For powers a^x, as long as the base of this power is positive—which it is in our case—its modulus does not depend on $\operatorname{Im} x$ at all,

     $$|a^x|=a^{\operatorname{Re} x}, \qquad \text{if } a\geq 0. \tag{5.1.4}$$

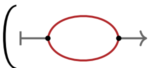
[4] A convenient implementation of PSLQ is available in MATHEMATICA as `FindIntegerNullVector`.

So, as long as $\mathscr{F}(p,x) \geq 0$, an upper bound on $|I_i(d)|$ at $\mathrm{Im}\,d \to \infty$ can be determined by the $\Gamma\big(\nu - L\frac{d}{2}\big)$ prefactor only,

$$|I_i(d)| \leq \mathscr{C}\,|\mathrm{Im}\,d|\,e^{-\frac{\pi L}{4}|\mathrm{Im}\,d|}\,|\mathrm{Im}\,d|^{-\frac{L}{2}\mathrm{Re}\,d}\,|\mathrm{Im}\,d|^{\nu-\frac{1}{2}}\,. \tag{5.1.5}$$

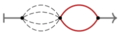
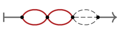
For cut integrals we shall later introduce the phase-space parameterization of eq. (9.2.2); under it an upper bound on $I_i(d)$ at $\mathrm{Im}\,d \to \infty$ can be determined by just the Ω_{d-k} factors, which only depend on the number of cuts.

Note that in both cases the bounds are the same for all integrals of equal loop and cut number, up to possibly a factor of $|\mathrm{Im}\,d|^{\alpha}$. It is useful to exploit this fact by rescaling all integrals to get rid of the common parts. Indeed, when working with integrals with L loops and K cuts, we can introduce

$$J_i \equiv I_i \Big/ \Bigg($$  $$^{L}\,\mathrm{PS}_K\Bigg), \tag{5.1.6}$$

where PS_K is the full K-particle phase-space given by eq. (9.2.2). Then for some $\mathscr{C}$ and α, have the asymptotic behavior as

$$|J_i(d)| \lesssim \mathscr{C}\,|\mathrm{Im}\,d|^{\alpha}\,, \qquad \text{when } \mathrm{Im}\,d \to \pm\infty. \tag{5.1.7}$$

In practice we always do this, and only consider DRR for J_i, not for I_i. This normalizes one of the master integrals (i.e. , , etc) along with the corresponding DRR matrix entry, to just 1.

- The behavior of $H_i^{-1}(d)$ can be directly deduced from eq. (5.0.6) using eq. (5.1.3) and eq. (5.1.4). When we are working with the rescaled integrals J_i, in all the cases we have considered we find that there are such $\mathscr{C}$ and β that

$$|H_i(d)| \approx \mathscr{C}\,|\mathrm{Im}\,d|^{\beta}\,, \qquad \text{when } \mathrm{Im}\,d \to \infty. \tag{5.1.8}$$

- The behavior of $R_i(d)$ directly follows from its construction, eq. (5.0.7). The bounds on $H_i(d)$ and $J_i(d)$ tell us that the modulus of each term in that series is asymptotically bounded by $\mathscr{C}\,|\mathrm{Im}\,d|^{\gamma}$, for some values of $\mathscr{C}$ and γ. To claim that the modulus of the whole series is bounded similarly too, it is enough to show that a series composed of these bounds converges. This is in fact the case, because the dependence of the series terms on $\mathrm{Re}\,d$ approaches an exponential, irrespective of the value of $\mathrm{Im}\,d$, so the whole series converges geometrically. Thus, there are such δ and $\mathscr{C}$ that

$$|R_i(d)| \lesssim |\mathrm{Im}\,d|^{\delta}\,, \qquad \text{when } \mathrm{Im}\,d \to \pm\infty. \tag{5.1.9}$$

If we have established all three bounds: eq. (5.1.7), eq. (5.1.8), and eq. (5.1.9), then from eq. (5.1.1) it follows that $\omega_i(d)$ must be bounded similarly,

$$|\omega_i(d)| \leq \mathscr{C}\,|\mathrm{Im}\,d|^{\alpha}\,, \qquad \text{when } \mathrm{Im}\,d \to \pm\infty. \tag{5.1.10}$$

7. Construct the ansatz for $\omega_i(d)$. Remember that $\omega_i(d)$ is periodic with period 2, so it can instead be viewed as a function of

$$e^{i\pi d} \equiv z. \tag{5.1.11}$$

Now, assuming the bound of eq. (5.1.10) was established, in terms of z it looks like

$$|\omega_i(d)| \leq \mathscr{C}\,|\ln z|^{\alpha}\,, \qquad \text{when } z \to 0 \text{ or } \infty. \tag{5.1.12}$$

Let us first look at the simple case when $\omega_i(d)$ was previously found to have no poles in d. In this case $\omega_i(z)$, when viewed as a function on the Riemann sphere, can only have poles at 0 and ∞ (corresponding to $\operatorname{Im} d \to \pm\infty$), but because $|\ln z|$ grows slower than any non-zero power of z, we can represent $\omega_i(z)$ as its Taylor series and immediately conclude that only one power in it is allowed: zero. In this simple case $\omega_i(z)$ can only be a constant.

A more complex case is when $\omega_i(d)$ was found to have non-empty set of poles at $\{d_k\}$ with multiplicities $\{p_k\}$. In this case observe that if we can subtract from $\omega_i(d)$ some function that would cancel its poles while not altering the form of its asymptotic behavior, then we can reuse the same arguments and conclude that this new expression can only be a constant. One way to construct such a function is to choose some kernel $K(d)$ that has a single pole at $d = 0$ and an appropriate asymptotic behavior, and then add these kernels for each pole d_k, making the full ansatz

$$\omega_i(d) = a_0 + \sum_i \sum_{n>p_i} a_{i,n} K^n(d - d_i), \tag{5.1.13}$$

where a_0 is the constant term, and $a_{i,n}$ are the constants needed to make sure this ansatz correctly reproduces all the poles of $\omega_i(d)$.

As for the $K(d)$ kernel, a common choice is the cotangent function,

$$K(d) = \cot\frac{\pi d}{2}, \tag{5.1.14}$$

which in terms of z is just

$$K(z) = i\frac{z+1}{z-1}. \tag{5.1.15}$$

8. Determine the ansatz constants a_0 and $a_{i,n}$.

   For the poles of $\omega_i(d)$ that originate as $H_i(d)$ and $R_i(d)$ poles, $a_{i,n}$ can be determined simply by expanding eq. (5.1.1) around $d = d_k - 2\varepsilon$: the coefficients around ε^{-1}, ε^{-2}, etc. will only come from $H_i(d)$ (as given by eq. (5.0.6)) or $R_i(d)$ (as given by the series in eq. (5.0.7) and then analytically restored in terms of MZVs via PSLQ), and thus are known. No knowledge of the $I_i(d)$ expansion is needed here. For the poles of $\omega_i(d)$ that originate as $I_i(d)$ poles no such nice procedure exist, because the expansion of $I_i(d)$ is not known (not with sufficient precision, at least). The constants $a_{i,n}$ that correspond to these must be determined from some separate considerations. Same for a_0.

   We will revisit the additional considerations for fixing a_0 and $a_{i,n}$ in Chapter 9.

6. Differential equations method

If a Feynman integral as given by eq. (3.0.1) depends only on a single mass scale (e.g. q), this dependence is trivial: $\mathscr{C} q^n$, where $\mathscr{C}$ is a constant, and n follows directly from the integral's dimensionality. This is why solving dimensional recurrence relations semi-numerically (as in Chapter 5) works so well: only one constant $\mathscr{C}$ needs to be determined. For multi-scale integrals $\mathscr{C}$ additionally depends on one or multiple scaleless parameters (e.g. ratios of scales). For such integrals different methods are needed, and the most powerful of them is the differential equations method [112, 113, 114]. The overall procedure is simple:

1. Start with some Feynman integral family $I_{\nu_1,\dots,\nu_N}$ (defined as in eq. (3.0.1)).
2. Solve IBP relations for this family and identify the set of master integrals I_i.
3. Choose one of the scaleless parameters, x, and differentiate each I_i by it.

   Differentiation by a mass term (i.e. $x = m^2/q^2$) is easy to perform using

   $$\frac{\partial}{\partial m^2} I_{\nu_1,\dots,\nu_N} = \int \frac{\mathrm{d}^d l_1}{(2\pi)^d} \cdots \frac{\mathrm{d}^d l_L}{(2\pi)^d} \frac{\partial}{\partial m^2} \left(\frac{1}{D_1^{\nu_1} \cdots D_N^{\nu_N}} \right). \tag{6.0.1}$$

   Differentiation by a scalar product of external momenta (i.e. $x = p^2/q^2$) can be done via

   $$\frac{\partial}{\partial p_a \!\cdot\! p_b} I_{\nu_1,\dots,\nu_N} = \int \frac{\mathrm{d}^d l_1}{(2\pi)^d} \cdots \frac{\mathrm{d}^d l_L}{(2\pi)^d} \sum_i \left(\mathbb{G}^{-1}\right)_{ia} p_i^\mu \frac{\partial}{\partial p_b^\mu} \left(\frac{1}{D_1^{\nu_1} \cdots D_N^{\nu_N}} \right), \tag{6.0.2}$$

   and

   $$\frac{\partial}{\partial p_a \!\cdot\! p_a} I_{\nu_1,\dots,\nu_N} = \int \frac{\mathrm{d}^d l_1}{(2\pi)^d} \cdots \frac{\mathrm{d}^d l_L}{(2\pi)^d} \sum_i \left(\mathbb{G}^{-1}\right)_{ia} \frac{1}{2} p_i^\mu \frac{\partial}{\partial p_a^\mu} \left(\frac{1}{D_1^{\nu_1} \cdots D_N^{\nu_N}} \right), \tag{6.0.3}$$

   where $\mathbb{G}$ is an $E \times E$ Gram matrix made out of the external momenta:[1]

   $$\mathbb{G}_{ij} \equiv p_i \!\cdot\! p_j. \tag{6.0.4}$$

4. Express the derivatives as linear combinations of integrals in the same family,

   $$\frac{\partial}{\partial x} I_i = \sum_k C_{ik} I_{\nu_1^{(k)},\dots,\nu_N^{(k)}}. \tag{6.0.5}$$

5. Use the IBP relations to reduce $I_{\nu_1^{(k)},\dots,\nu_N^{(k)}}$ back to the master integrals I_i,

   $$I_{\nu_1^{(k)},\dots,\nu_N^{(k)}} \overset{\text{IBP}}{=} \sum_j K_{kj} I_j \qquad \Rightarrow \qquad \frac{\partial}{\partial x} I_i = \sum_j \underbrace{\sum_k C_{ik} K_{kj}}_{\equiv M_{ij}} I_j, \tag{6.0.6}$$

   thus obtaining a linear differential equation system for I_i, which in short reads

   $$\partial_x I_i = \sum_j M_{ij} I_j. \tag{6.0.7}$$

[1] Practically, LITERED [78] conveniently implements this as the `Dinv` function.

The idea is then to search for $I_i(x)$ as the solution of this system of differential equations, instead of performing the loop integration directly.

6.1. Solving differential equations

Suppose we have a set of n master integrals $I_i(d,x)$, and we have used the IBP relations to construct a differential equation system,[2]

$$\partial_x \mathbf{I}(d,x) = \mathbb{M}(d,x)\,\mathbf{I}(d,x), \tag{6.1.1}$$

where $\mathbf{I}(d,x)$ is a column vector consisting of all I_i, and $\mathbb{M}(d,x)$ is an $n\times n$ matrix defining the differential equation. In general, because both the IBP relations and the differentiation operation have rational coefficients, $\mathbb{M}(d,x)$ will be rational in d and x. Let us represent it using the following ansatz:

$$\mathbb{M}(d,x) = \sum_i \frac{\mathbb{M}_i(d)}{(x-x_i)^{k_i}}, \tag{6.1.2}$$

where x_i are the location of poles, and k_i are integer powers, possibly negative (in which case x_i is zero by convention).

Solving eq. (6.1.1) directly can be hard. Instead, it is possible to transform it into a form that allows for an easier solution. There are several transformations that can be applied to this end.

One possible transformation is a *change of a variable* from x to y such that

$$x = f(y). \tag{6.1.3}$$

Under a change of a variable the differential equation is transformed into

$$\partial_y \mathbf{I}(d,x) = \underbrace{f'(y)\,\mathbb{M}(d,f(y))}_{\mathbb{M}'}\mathbf{I}(d,f(y)). \tag{6.1.4}$$

Another possible transformation is a linear change of basis from $\mathbf{I}$ to $\mathbf{J}$ such that

$$\mathbf{I}(d,x) = \mathbb{T}(d,x)\,\mathbf{J}(d,x), \tag{6.1.5}$$

where $\mathbb{T}$ is an $n \times n$ transformation matrix. Under a change of basis the differential equation becomes

$$\partial_x \mathbf{J} = \underbrace{\mathbb{T}^{-1}\left(\mathbb{M}\mathbb{T} - \partial_x \mathbb{T}\right)}_{\mathbb{M}'}\mathbf{J}. \tag{6.1.6}$$

To make use of these transformations, we turn to an observation made in [29]: it is often possible to find such a transformation that factorizes the dependence of $\mathbb{M}$ on d, giving us $\mathbb{M}'(d,x) = \varepsilon\,\mathbb{S}(x)$, where $\varepsilon = 2 - d/2$, and thus turning the system into an *ε-form* (also called the *canonical form*):

$$\partial_x \mathbf{J}(d,x) = \varepsilon\,S(x)\,\mathbf{J}(d,x). \tag{6.1.7}$$

Then, constructing the solution $\mathbf{J}$ as a series in ε becomes trivial. Indeed, by expanding $\mathbf{J}$ in ε starting at some arbitrary order k_0,

$$J(d,x) = \varepsilon^{k_0} \sum_{k=0}^{\infty} \mathbf{J}^{(k)}(x)\,\varepsilon^k, \tag{6.1.8}$$

[2] Here and further we shall use matrix notation, with column vectors denoted as $\mathbf{I} \equiv \{I_i\}$, and matrices as $\mathbb{M} \equiv M_{ij}$.

and inserting this expansion into eq. (6.1.7), we immediately get

$$\mathbf{J}^{(k)}(d,x) = \int_{x_0}^{x} \mathbb{S}(y)\,\mathbf{J}^{(k-1)}(y)\,\mathrm{d}y + \mathbf{C}^{(k)}, \tag{6.1.9}$$

where x_0 is an arbitrary integration origin, and $\mathbf{C}^{(k)}$ are vectors of integration constants, with the combined integration constant,

$$\mathbf{C}(d) \equiv \varepsilon^{k_0} \sum_{k=0}^{\infty} \mathbf{C}^{(k)} \varepsilon^{k}, \tag{6.1.10}$$

having the meaning of $\mathbf{J}(d, x_0)$, if such limit exists. Alternatively the integrals in eq. (6.1.9) can be indefinite, in which case $\mathbf{C}(d)$ has no meaning.

The recursion in eq. (6.1.9) terminates by $\mathbf{J}^{(-1)} = 0$, and the leading order of the ε expansion is defined by k_0 in eq. (6.1.8).

We shall discuss how to find the transformation that achieves an ε-form in Section 6.3. There we will also see that the $\mathbb{S}(x)$ matrices we will construct will come in *Fuchsian form*:

$$\mathbb{S}(x) = \sum_i \frac{\mathbb{S}_i}{x - x_i}, \tag{6.1.11}$$

so the integral in eq. (6.1.9) will split into terms of the general form

$$\int^{x} \frac{\mathrm{d}y_1}{y_1 - w_1} \int^{y_1} \frac{\mathrm{d}y_1}{y_1 - w_1} \cdots \int^{y_n} \frac{\mathrm{d}y_n}{y_n - w_n}. \tag{6.1.12}$$

The simplest of these (corresponding to the first integration) is just $\ln(x) = \int_1^x \mathrm{d}y/y$, the second integration would already require dilogarithms Li_2, the third—polylogarithms, and so on. The most general class of functions covering these integrals are multiple polylogarithms. There is much to be said about them; please refer to Chapter 7 for their precise definition and properties.

6.1.1. Finding the integration constants

After the solution as a series for I_i is constructed, what remains is to determine k_0 and the integration constants $C_i^{(k)}$. This cannot be done from the differential equations alone: additional constraints on the integral values are required. These constraints can come in different forms, and depend on the integrals being considered. Finding them is the essential difficulty of the method. Here are some of the ways.

- By evaluating the integrals in a limit where they simplify.

  For example a massive integral may have a massless equivalent that is easier to calculate—comparing the massless limit of the series for I_i with the known value will determine $C_i^{(k)}$ (in the next section we shall see an example of this). The large-mass limit can be similarly useful. See [115] for an in-depth treatment of both of these limits. Kinematic limits (e.g. zero external momenta, expansion near threshold) can be used in some cases.

- By using the knowledge of the analytic properties of the integrals.

  Just demanding the solution to be regular in particular kinematic limits already constrains the solution greatly, and the more scales there are, the more constraints one obtains. This works because the solutions for multi-scale integrals often contain discontinuities in the limits where they should be absent on physical grounds. For example, massless integrals should only have discontinuities when external momenta invariants cross the mass threshold (that is, zero). In [116] the master integrals for massless two-loop three-point functions were calculated from these constraints alone.[3]

- From partial knowledge of the integral's values, such as its integral over x, or a Mellin moment.

  This is relevant for the semi-inclusive integrals which we are after: an integral over x from a semi-inclusive integral makes it a fully inclusive one. So, with the knowledge of the fully inclusive integral values, the integration constants for semi-inclusive ones can be determined (as proposed in [69]).

6.1.2. Fundamental solution

A different way of constructing the solution to eq. (6.1.7) is to determine first the *fundamental solution* for $\mathbf{J}(d,x)$, $\mathbb{W}(d,x)$, which is an $n\times n$ matrix of independent solutions with the useful property that any solution can be expressed as a linear combination of its columns:

$$\mathbf{J}(d,x)=\mathbb{W}(d,x)\,\mathbf{C}(d)\,, \tag{6.1.13}$$

with $\mathbf{C}(d)$ being a vector of some constants. The constants are arbitrary; it is often useful to represent them as series in ε, the same way as done in eq. (6.1.10).

The fundamental solution for eq. (6.1.7) can be constructed as an expansion in ε analogously to eq. (6.1.9):

$$\mathbb{W}(d,x)=\mathrm{P}\exp\left(\varepsilon\int_{x_0}^{x}\mathbb{S}(y)\,\mathrm{d}y\right)\equiv\mathbb{1}+\varepsilon\int_{x_0}^{x}\mathrm{d}y_1\mathbb{S}(y_1)+\varepsilon^2\int_{x_0}^{x}\mathrm{d}y_1\mathbb{S}(y_1)\int_{x_0}^{y_1}\mathrm{d}y_2\mathbb{S}(y_2)+\ldots, \tag{6.1.14}$$

where x_0 is an arbitrary starting point of integration (usually taken to be 0), and $\mathrm{P}\exp$ is a path-ordered exponent. When $\mathbb{W}(d,x)$ is constructed this way, the constant vector $\mathbf{C}(d)$ acquires the meaning of $\mathbf{J}(d,x_0)$.

This is an alternative way to write down a solution for $\mathbf{J}$, with the benefit of being immediately extendable to the multivariate case.

6.1.3. Multivariate differential equations

So far we have considered differential equations in a single variable—this was for clarity only, and the same arguments apply if the integrals in question depend on several dimensionless variables $x_1,...,x_n$. The solution in this case can, in principle, be computed via differential equations in a stepwise fashion: by first solving the system in x_1, then in x_2, etc, matching the results with each other. A more convenient way is to consider the differential equation system in all x_i at once, and reduce it to a combined ε-form,

$$\frac{\partial}{\partial x_i}\mathbf{J}(d,\mathbf{x})=\varepsilon\,\mathbb{S}_i(\mathbf{x})\,\mathbf{J}(d,\mathbf{x})\,. \tag{6.1.15}$$

Once this is achieved, writing down the solution can be made easy by:

1. Choosing an integration contour along the axes x_i in an some order, for example from $(0,0,\ldots)$ to $(x_1,0,\ldots)$, then to $(x_1,x_2,0,...)$, and so on.

[3] In [35] we have recalculated those integrals to higher transcendental weight using the same idea (as an intermediate step of the overall calculation).

2. Writing down the fundamental solutions along each segment, $\mathbb{W}_i$. Because segments are chosen such that only x_i changes along i-th segment, $\mathbb{W}_i$ can be calculated the same way as in the single-variate case, via eq. (6.1.14) with x_i instead of x. Note that $\mathbb{W}_i = \mathbb{W}_i(d, x_1, \dots, x_i)$.
3. The general solution for $\mathbf{J}$ is then

$$\mathbf{J}(d, \mathbf{x}) = \mathbb{W}_n(d, x_1, \dots, x_n) \cdots \mathbb{W}_1(d, x_1)\, \mathbf{C}(d)\,. \tag{6.1.16}$$

Note that with this integration contour construction the solution for $\mathbb{W}_i$ will consist of multiple polylogarithm terms of the form

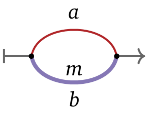
$$\begin{aligned} &G(\{\text{constants}\}; x_1) \times \\ &\quad \times G(\{\text{parameters depending on } x_1\}; x_2) \times \cdots \times \\ &\quad \times G(\{\text{parameters depending on } x_1, \dots, x_{n-1}\}; x_n)\,. \end{aligned} \tag{6.1.17}$$

This form is important because as discussed in Chapter 7 all such combinations of multiple polylogarithms are linearly independent, so when matching two expressions involving them it is enough to demand that the coefficients at each unique product of this form are identical. This would not be the case if e.g. x_2 would enter the parameter list of $G(\dots; x_1)$; such a product would need to be additionally decomposed in terms of a linearly independent basis such as this one.

6.2. Example: self-energy with a mass

As an example let us consider the following family of integrals:

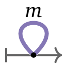
$$I_{a,b} \equiv$$



$$\equiv \int \frac{\mathrm{d}^d l}{(2\pi)^d} \frac{1}{(q-l)^{2a}\,(l^2-m^2)^b}. \tag{6.2.1}$$

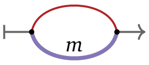
Solving IBP relations for it gives us two master integrals; these can be chosen as

$$I_1 = I_{0,1} \equiv$$



and

$$I_2 = I_{1,1} \equiv$$



$$. \tag{6.2.2}$$

This family has two mass scales: the incoming energy $\sqrt{q^2}$ and the mass m. Let us define the scaleless variable as

$$x = m^2/q^2. \tag{6.2.3}$$

Differentiating $I_{a,b}$ by x can be straightforwardly done by applying the derivative to the integrand of eq. (6.2.1):

$$\frac{\partial}{\partial x} I_{a,b} = q^2 \frac{\partial}{\partial m^2} I_{a,b} = q^2 b I_{a,b+1}. \tag{6.2.4}$$

Note that in practical calculations it is most convenient not to take the q^2 factors along, and set $q^2 = 1$, restoring the proper factors at the end of the calculation from dimensionality. We shall retain these factors here though, for greater clarity.

Next, via IBP relations we can reduce $\partial_x I_1$ and $\partial_x I_2$ back to I_1 and I_2, obtaining the differential equation system:

$$\frac{\partial}{\partial x} \begin{pmatrix} I_1 \\ I_2 \end{pmatrix} = \begin{pmatrix} q^2 I_{0,2} \\ q^2 I_{1,2} \end{pmatrix} \overset{\text{IBP}}{=} \underbrace{\begin{pmatrix} \frac{d-2}{x} & 0 \\ \frac{2-d}{2x(x-1)} \frac{1}{q^2} & \frac{d-3}{x-1} \end{pmatrix}}_{\mathbb{M}} \begin{pmatrix} I_1 \\ I_2 \end{pmatrix}. \tag{6.2.5}$$

6.2.1. Solving the differential equations

Once the differential equation system of eq. (6.2.5) is constructed the next step is to reduce it to an ε-form. The transformation required for this can be obtained using the methods of the following section (specifically, by running Fuchsia from Chapter 8); the transformation is

$$\mathbf{I} = \begin{pmatrix} (2\varepsilon-1)\,xq^2 & 0 \\ \varepsilon-1 & (1-\varepsilon)(1-x) \end{pmatrix} \mathbf{J}. \tag{6.2.6}$$

The differential equation system for $\mathbf{J}$ can then be found via eq. (6.1.6):

$$\partial_x \mathbf{J} = \varepsilon \begin{pmatrix} -\frac{1}{x} & 0 \\ -\frac{1}{x} + \frac{1}{x-1} & -\frac{2}{x-1} \end{pmatrix} \mathbf{J}. \tag{6.2.7}$$

The solution for $\mathbf{J}$ can then be constructed via eq. (6.1.9):[4]

$$\begin{aligned} J_1/\varepsilon^{k_0} &= C_1^{(0)} + \left(-\ln(x)\,C_1^{(0)} + C_1^{(1)}\right)\varepsilon + \dots, \\ J_2/\varepsilon^{k_0} &= C_2^{(0)} + \left(-\ln(x)\,C_1^{(0)} + \ln(1-x)\left(C_1^{(0)} - 2C_2^{(0)}\right) + C_2^{(1)}\right)\varepsilon + \dots, \end{aligned} \tag{6.2.8}$$

and the same for $\mathbf{I}$ then follows from eq. (6.2.6):

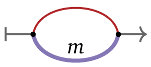
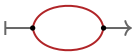
$$\begin{aligned} I_1/\varepsilon^{k_0} &= -q^2 x C_1^{(0)} + \left(x\ln(x)\,C_1^{(0)} + x\left(2C_1^{(0)} - C_1^{(1)}\right)\right)\varepsilon + \dots, \\ I_2/\varepsilon^{k_0} &= \left((1-x)\,C_2^{(0)} - C_1^{(0)}\right) + \\ &\quad + \left(x\ln(x)\,C_1^{(0)} + (1-x)\ln(1-x)\left(C_1^{(0)} - 2C_2^{(0)}\right) + C_1^{(0)} - C_1^{(1)} + (1-x)\left(C_2^{(1)} - C_2^{(0)}\right)\right)\varepsilon + \dots. \end{aligned} \tag{6.2.9}$$

What remains is to fix the integration constants $C_i^{(k)}$.

6.2.2. Integration constants

To obtain $C_1^{(k)}$ and $C_2^{(k)}$ we can use two observations:

1. We have already calculated I_1, the first master integral, previously: its value is given by eq. (3.1.4). By expanding that value into a series in ε and demanding it being equal to eq. (6.2.9) we can obtain all $C_1^{(k)}$ values.
2. The second master integral I_2 is just a massive version of the 1-loop self-energy bubble from eq. (3.2.7), so in the limit of $m \to 0$ (i.e. $x \to 0$) their values must match (unless I_2 diverges):

   $\lim_{m\to 0}$  $=$  . (6.2.10)

   So, expanding the value of the massless bubble given by eq. (3.2.10) and matching it with $\lim_{x\to 0} I_2$, we can obtain all $C_2^{(k)}$. Fortunately for us, as one can see from eq. (6.2.9), $\lim_{x\to 0} I_2$ exists and the expression for it is easy to derive.

[4] Note that the first level of integrals here were taken in terms of logarithms—this is done for demonstration purposes only: high enough ε expansion orders will require using multiple polylogarithms (Chapter 7), so in practice it is best to use them directly from the start.

Before we do that, note that the $i0$ prescriptions for the massive vacuum bubble in eq. (3.1.4) and the massless one in eq. (3.2.10) have the opposite signs. This makes sense because massless integrals are always above the mass threshold, and the vacuum bubbles are always below. So, for greater simplicity, let us drop the $i0$ prescriptions by restricting ourselves to the kinematic region

$$q^2 > m^2 > 0. \tag{6.2.11}$$

Also note that a direct expansion of eq. (3.1.4) will have terms like $\gamma_E - \ln(4\pi)$ and $\ln\big(q^2\big)$ cluttering it; it is more convenient to factor out these terms into a prefactor common among all integrals, and keep it unexpanded. Let us introduce this prefactor as

$$\mathbf{J} = \frac{i e^{-\gamma_E \varepsilon}}{(4\pi)^{2-\varepsilon}} \big(q^2\big)^{-\varepsilon} \, \mathbf{J}'; \tag{6.2.12}$$

the solution for $\mathbf{J}'$ is then the same as in eq. (6.2.8), except the integration constants are different—let us denote them $C_i'^{(k)}$.

Finally, matching I_1 with eq. (3.1.4) gives us

$$C_1'^{(0)} = -1, \qquad C_1'^{(1)} = -3, \qquad C_1'^{(2)} = -7 - \frac{\pi^2}{12}, \tag{6.2.13}$$

and because the expansion of eq. (3.1.4) starts with a $1/\varepsilon$ pole, we must start the solution series from $1/\varepsilon$ too, setting

$$k_0 = -1. \tag{6.2.14}$$

Matching $\lim_{x\to 0} I_2$ with eq. (3.2.10) gives us

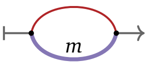
$$C_2'^{(0)} = 0, \qquad C_2'^{(1)} = i\pi, \qquad C_2'^{(2)} = 3i\pi - \frac{2}{3}\pi^2. \tag{6.2.15}$$

Inserting these C' values into eq. (6.2.9), the result for I_2 becomes

$$I_2(x) = \frac{i e^{-\gamma_E \varepsilon}}{(4\pi)^{2-\varepsilon}} \big(q^2\big)^{-\varepsilon} \big[\varepsilon^{-1} + (2 + i\pi(1-x) - x\ln(x) - (1-x)\ln(1-x))\,\varepsilon^0 + \ldots\big]. \tag{6.2.16}$$

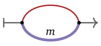
6.2.3. Cross-check

The second master integral I_2 can in fact be taken directly using Feynman parameterization (see Section 3.2), yielding the result[5]



$$= \frac{i\pi^{\frac{d}{2}}}{(2\pi)^d} \big(-q^2 - i0\big)^{\frac{d}{2}-2} \Gamma\Big(2 - \frac{d}{2}\Big) \Bigg[\Big(1 - \frac{m^2 - i0}{q^2}\Big)^{d-3} \frac{\Gamma^2\big(\frac{d}{2}-1\big)}{\Gamma(d-2)} + \\ + \frac{\Big(-\frac{m^2-i0}{q^2}\Big)^{\frac{d}{2}-1}}{1-\frac{d}{2}} {}_2F_1\Big(1, 2-\frac{d}{2}; \frac{d}{2}; \frac{m^2 - i0}{q^2}\Big)\Bigg]. \tag{6.2.17}$$

[5] Interestingly, if one inverts the argument of ${}_2F_1$, the formula becomes simpler:



$$= -\frac{i\pi^{\frac{d}{2}}}{(2\pi)^d} \Gamma\Big(1 - \frac{d}{2}\Big) \big(m^2 - i0\big)^{2-\frac{d}{2}} {}_2F_1\Big(1, 2 - \frac{d}{2}; \frac{d}{2}; \frac{q^2}{m^2 - i0}\Big).$$

Both expressions are, of course, equivalent, and are analytic continuations of each other between the regions $m^2/q^2 < 1$ and $m^2/q^2 > 1$. The $\pm i0$ prescriptions provided here make them valid in all regions, so using one or the other is a matter of convenience.

6.3. Reduction to epsilon form

6.3.1. Algorithm by Moser

In [117] it was shown that one can transform the system eq. (6.1.1), with $\mathbb{M}$ having the form

$$\mathbb{M} = \frac{\mathbb{A}_0}{x^k} + \frac{\mathbb{A}_1}{x^{k-1}} + \ldots, \qquad \text{where } k > 1, \tag{6.3.1}$$

using a sequence of simple basis transformations of the general form

$$\mathbb{T} = (\mathbb{P}_0 + x\mathbb{P})\,\mathrm{diag}(1,\ldots,1,x,\ldots,x), \tag{6.3.2}$$

with each step simplifying the matrix $\mathbb{M}$ slightly by reducing the rank of $\mathbb{A}_0$, until the rank reaches 0, and $\mathbb{A}_0$ itself becomes zero. This way all the poles $1/x^k$ can be removed, until only $1/x$ remains. It was also proven that one step of this reduction is possible if and only if

$$x^{\mathrm{rank}\mathbb{A}_0} \det\left(\lambda\mathbb{1} + \frac{\mathbb{A}_0}{x} + A_1\right)\bigg|_{x=0} = 0 \qquad \text{for all } \lambda, \tag{6.3.3}$$

otherwise no rational transformation can improve $\mathbb{A}_0$. This is *Moser's reducibility criterium*. It is additionally important because it demonstrates that we can restrict ourselves to only considering one simple transformation at a time, and not lose generality in doing so.

Note that not only $1/x^k$, but all the $1/(x-x_i)^{k_i}$ poles are handled by the same method too, because one can always transform x in eq. (6.1.1) using e.g. the Möbius transformation,

$$x = \frac{ax' + b}{cx' + d}, \tag{6.3.4}$$

such that any three chosen points in x are mapped onto any other three points in x_0, applying the algorithm, and reversing the transformation.[6] In other words, $x - x_i$ can be mapped to x', and the method will still hold. Also note that the point at infinity can also be handled this way, with eq. (6.3.4) for it becoming just $x = 1/x'$.

The limitation of the method of [117] (and its later iterations like [118, 119])[7] is that while the parts of $\mathbb{M}$ of the form $1/(x-x_i)^{k_i}$ are reduced to $1/(x-x_i)$, this is done at the expense of introducing possibly large polynomial parts (proportional to x^k, $k \geq 0$), spoiling the behavior at $x = \infty$.

Alternatively, through eq. (6.3.4) one can have the same method spoil the behavior at some point other than ∞. For example after $x = 1/x'$ the polynomial part x^k becomes $1/x'^{k+2}$ (see eq. (6.1.4)), and the same algorithm that reduces the pole $1/x'^{k+2}$ at the expense of the polynomial part x'^k does the reverse in terms of x: reduces the polynomial part at the expense of the $1/x$ pole.

Even more, one can choose a different point to spoil at each step, and have the algorithm introduce multiple new $1/(x-x_i)$ poles while removing higher poles. This will reduce the whole $\mathbb{M}$ to a *Fuchsian form*,

$$\mathbb{M}(d) = \sum_i \frac{\mathbb{M}_i(d)}{x - x_i}, \tag{6.3.5}$$

but at the price of introducing multiple spurious $1/(x-x_i)$ poles.

[6] Similar transformations with higher powers of x may introduce roots when reversed; this is why the Möbius transformation is special.

[7] An implementation of these algorithms is available in the standard MAPLE package DETOOLS as `moser_reduce` and `super_reduce` routines.

6.3.2. Algorithm by Lee

In [30] Moser's algorithm was extended to prevent the spoiling of the $x = \infty$ point (and all other points, if possible), so that the final $\mathbb{M}$ has purely Fuchsian form with no (or at least few) new poles introduced. This subsequently allows for further reduction that achieves an ε-form of eq. (6.1.7), factorizing the $\mathbb{M}_i(d)$ dependence on d as

$$\mathbb{M}(d) = \varepsilon \sum_i \frac{\mathbb{S}_i}{x - x_i}. \tag{6.3.6}$$

The elementary basis transformation matrix to simplify the system considered in [30] has the form

$$\mathbb{B}(\mathbb{P}, x_1, x_2; x) = \overline{\mathbb{P}} + c\frac{x - x_2}{x - x_1}\mathbb{P}, \tag{6.3.7}$$

where $\mathbb{P}$ is some projection matrix ($\mathbb{P}^2 = \mathbb{P}$) constant in x, $\overline{\mathbb{P}} \equiv \mathbb{1} - \mathbb{P}$, and c is there to define the special cases of $x_1 = \infty$ and of $x_2 = \infty$:

$$c \equiv \begin{cases} x_1 & \text{if } x_1 = \infty, \\ 1/x^2 & \text{if } x_2 = \infty, \\ 1 & \text{otherwise.} \end{cases} \tag{6.3.8}$$

These are chosen so that

$$\mathbb{B}(\mathbb{P}, x_1, x_2; x)\mathbb{B}(\mathbb{P}, x_2, x_1; x) = \mathbb{1}. \tag{6.3.9}$$

In fact eq. (6.3.2) is related to $\mathbb{B}(\mathbb{P}, \infty, 0; x)$.

So, how does eq. (6.1.1) change when the transformation $\mathbb{B}(\mathbb{P}, x_1, x_2; x)$ is applied to

$$\mathbb{M} = \frac{\mathbb{A}_0}{(x - x_1)^{k_1}} + \frac{\mathbb{A}_1}{(x - x_1)^{k_1 - 1}} + \dots + \frac{\mathbb{B}_0}{(x - x_2)^{k_2}} + \dots ? \tag{6.3.10}$$

We have the general answer in eq. (6.1.6), and there are two cases to consider: the case of Fuchsian poles (i.e. k_1 being 1), and the case of higher poles. They differ because the $\partial_x \mathbb{T}$ term generates an additional $1/x$ pole that only affects $\mathbb{A}_0'$ if $k_1 = 1$. They also differ because we can eliminate the higher poles, but not the Fuchsian ones.

6.3.3. Eliminating higher poles

If $k_1 > 1$ and $k_2 \geq 1$ in eq. (6.3.10), then applying the basis transformation $\mathbf{I} = \mathbb{B}(\mathbb{P}, x_1, x_2; x)\mathbf{I}'$ results in a differential equation system for $\mathbf{I}'$ defined by the matrix[8]

$$\mathbb{M}' = \frac{(x_2 - x_1)\mathbb{P}\mathbb{B}_0\overline{\mathbb{P}}}{(x - x_2)^{k_2 + 1}} + \frac{(x_1 - x_2)\overline{\mathbb{P}}\mathbb{A}_0\mathbb{P}}{(x - x_1)^{k_1 + 1}} + \underbrace{\frac{\overline{\mathbb{P}}\mathbb{A}_0 + \mathbb{P}\mathbb{A}_0\mathbb{P} + (x_1 - x_2)\overline{\mathbb{P}}\mathbb{A}_1\mathbb{P}}{(x - x_1)^{k_1}}}_{\equiv \mathbb{A}_0'/(x - x_1)^{k_1}} + \dots. \tag{6.3.11}$$

Now, we want this transformation to simplify $\mathbb{M}$. This means that the first and the second terms that introduce higher poles at x_2 and x_1 must vanish, and the third term must be "simpler" than what the original matrix had (which is $\mathbb{A}_0$). By "simpler" we shall mean "having lower rank".

[8] Here we assume for simplicity that neither x_1 nor x_2 are ∞. A practical implementation must however treat the point $x = \infty$ just as it would any other point. The relations for this special case can be derived via Möbius transformations. These work out to replacing $1/(x - x_i)^k$ with x^k and $\mathbb{A}_i$ with $-\mathbb{A}_i$.

To cancel the first two terms we must constrain the co-image (row span) of $\mathbb{P}$ with *condition I*:[9]

$$\mathbb{P}\mathbb{B}_0\overline{\mathbb{P}} = 0, \qquad \text{or equivalently,} \qquad \mathbb{B}_0^T \operatorname{image}\left(\mathbb{P}^T\right) \subseteq \operatorname{image}\left(\mathbb{P}^T\right), \tag{6.3.12}$$

and its image (column span) with *condition II*:

$$\overline{\mathbb{P}}\mathbb{A}_0\mathbb{P} = 0, \qquad \text{or equivalently,} \qquad \mathbb{A}_0 \operatorname{image}(\mathbb{P}) \subseteq \operatorname{image}(\mathbb{P}). \tag{6.3.13}$$

A sufficient (and as we shall see, required) set of conditions to make $\operatorname{rank}\mathbb{A}_0' < \operatorname{rank}\mathbb{A}_0$ is given by *condition III*:

$$\mathbb{A}_0\mathbb{P} = 0, \tag{6.3.14}$$

condition IV:

$$\mathbb{A}_1 \operatorname{image}(\mathbb{P}) \subseteq \operatorname{image}(\mathbb{A}_0) \cup \operatorname{image}(\mathbb{P}), \tag{6.3.15}$$

and *condition V*:

$$\operatorname{image}(\mathbb{P}) \cap \operatorname{image}(\mathbb{A}_0) \neq \{0\}. \tag{6.3.16}$$

Indeed, because of III from eq. (6.3.11) we have[10]

$$\mathbb{A}_0' = \overline{\mathbb{P}}\mathbb{A}_0 + (x_1 - x_2)\overline{\mathbb{P}}\mathbb{A}_1\mathbb{P}; \tag{6.3.17}$$

because of IV and the fact that $\overline{\mathbb{P}}\mathbb{P} = 0$, we have

$$\operatorname{image}\left(\mathbb{A}_0'\right) \subseteq \overline{\mathbb{P}} \operatorname{image}(\mathbb{A}_0) \cup \overline{\mathbb{P}} \operatorname{image}(\mathbb{A}_1\mathbb{P}) \subseteq \overline{\mathbb{P}} \operatorname{image}(\mathbb{A}_0), \tag{6.3.18}$$

and V requires that there is at least one vector in $\operatorname{image}(\mathbb{A}_0)$ that $\overline{\mathbb{P}}$ annihilates, so that

$$\underbrace{\dim \operatorname{image}}_{\equiv \text{rank}}\left(\mathbb{A}_0'\right) < \dim \operatorname{image}(\mathbb{A}_0). \tag{6.3.19}$$

If we manage to construct $\mathbb{P}$ that satisfies these conditions, then applying the basis transformation $\mathbb{B}(\mathbb{P}, x_2, x_1; x)$ will reduce the rank of $\mathbb{A}_0$. The same procedure repeated multiple times (with possibly different x_2) will reduce $\mathbb{A}_0$ to 0, and thus reduce the highest pole power at x_1 by 1. Repeated multiple times to each x_i we can eliminate all the higher poles, achieving the Fuchsian form, as in eq. (6.3.5).

6.3.3.1. Constructing the projector matrix

Note that condition I restricts the co-image of $\mathbb{P}$, while II-V restrict its image. These two can be constructed separately and then combined into the full $\mathbb{P}$.

Constructing $\operatorname{image}\left(\mathbb{P}^T\right)$ that satisfies condition I is easy: any eigenspace of $\mathbb{B}_0^T$ (or their union) fits.

To construct $\operatorname{image}(\mathbb{P})$ that satisfies conditions II-V, let us return to Moser's reducibility criterium of eq. (6.3.3). An equivalent form of it was introduced in [120, ch E.8] (building upon [118]) as:

$$\dim \operatorname{nullspace}\underbrace{\begin{pmatrix} \mathbb{A}_0 & \mathbb{A}_1 + \mathbb{1}\lambda \\ 0 & \mathbb{A}_0 \end{pmatrix}}_{\equiv \mathbb{A}_\lambda} > \dim \operatorname{nullspace}(\mathbb{A}_0). \tag{6.3.20}$$

[9] Here we define $\operatorname{image}(\mathbb{P})$ as the span of columns of $\mathbb{P}$ (i.e. the set of $\mathbb{P}\mathbf{v}$ for all vectors $\mathbf{v}$), and $\operatorname{nullspace}(\mathbb{P})$ as the set of all $\mathbf{v}$ such that $\mathbb{P}\mathbf{v} = 0$. Note that because $\mathbb{P}\overline{\mathbb{P}} = \overline{\mathbb{P}}\mathbb{P} = 0$, $\operatorname{nullspace}\left(\overline{\mathbb{P}}\right) = \operatorname{image}(\mathbb{P})$ and $\operatorname{image}\left(\overline{\mathbb{P}}\right) = \operatorname{nullspace}(\mathbb{P})$.

[10] Note that this equation is also given in [30] as eq. (3.22), but it has a mistake there: the $(x_1 - x_2)$ factor is missing. This omission however does not affect any further results in that paper.

Note that for any vector $\mathbf{v}$ belonging to nullspace($\mathbb{A}_0$) there is a corresponding vector $\begin{pmatrix} \mathbf{v} \\ 0 \end{pmatrix}$ belonging to $\mathbb{A}_\lambda$, and vice versa. The criterium then requires nullspace($\mathbb{A}_\lambda$) to contain at least one vector $\begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix}$ with $\mathbf{u} \neq 0$.

Now, computing a null space of a matrix is a straightforward operation: performing Gaussian elimination on $\mathbb{A}_\lambda$ immediately leads to a basis of solutions to $\mathbb{A}_\lambda \mathbf{N} = 0$; the null space is then the span of this basis.

Suppose we have constructed this basis and have found some $\mathbf{N}(\lambda) = \begin{pmatrix} \mathbf{v}(\lambda) \\ \mathbf{u}(\lambda) \end{pmatrix}$ with $\mathbf{u}(\lambda) \neq 0$. In general $\mathbf{u}$ will be a rational function of λ, but we can rescale $\mathbf{N}$ to remove its overall denominator and make $\mathbf{u}$ just a polynomial in λ,

$$\mathbf{u}(\lambda) = \sum_k \mathbf{u}^{(k)} \lambda^k \qquad \text{and} \qquad \mathbf{v}(\lambda) = \sum_k \mathbf{v}^{(k)} \lambda^k. \tag{6.3.21}$$

The span of all $\mathbf{u}^{(k)}$ then satisfies all the conditions for image($\mathbb{P}$):

- conditions II and III because of the second row of eq. (6.3.20): $\mathbb{A}_0 \mathbf{u} = 0$;
- condition IV because of the first row of eq. (6.3.20):

$$\mathbb{A}_0 \mathbf{v}^{(k)} + \mathbb{A}_1 \mathbf{u}^{(k)} + \mathbf{u}^{(k-1)} = 0 \qquad \Rightarrow \qquad \mathbb{A}_1 \mathbf{u}^{(k)} \in \operatorname{image}(\mathbb{A}_0) \cup \operatorname{span}\big(\mathbf{u}^{(k-1)}\big); \tag{6.3.22}$$

- condition V because the coefficient next to the highest power of λ in the same row reads

$$\mathbb{A}_0 \mathbf{v}^{(k+1)} + \mathbf{u}^{(k)} = 0. \tag{6.3.23}$$

6.3.3.2. Combining image and co-image

Once we have constructed image($\mathbb{P}$) and image($\mathbb{P}^\mathsf{T}$), constructing the projector $\mathbb{P}$ is easy: let us select some matrix $\mathbb{U}$, such that its columns span image($\mathbb{P}$), and some matrix $\mathbb{V}$, such that its columns span image($\mathbb{P}^\mathsf{T}$); then,

$$\mathbb{P} = \mathbb{U}\,(\mathbb{V}\mathbb{U})^{-1}\,\mathbb{V}. \tag{6.3.24}$$

Note that it is possible that $\mathbb{V}\mathbb{U}$ is not invertible for any choice of $\mathbb{U}$ and $\mathbb{V}$, and no such $\mathbb{P}$ can be constructed. In such a case we can just give up on trying to satisfy the restriction on image($\mathbb{P}^\mathsf{T}$), and just set $\mathbb{P} = \mathbb{U}$: this restriction only exists to cancel the first term of eq. (6.3.11), so that the power of the pole at x_2 is not raised—but if all we are interested in is removing the higher poles, then instead of choosing x_2 as some existing pole and demanding that its power is not increased, we can choose x_2 as any regular point, and allow the transformation to introduce a new (spurious) pole of power 1 at it—this spurious pole will be eliminated during normalization anyway.

6.3.4. Normalizing Fuchsian poles

If $k_1 = 1$ and $k_2 = 1$ in eq. (6.3.10), then the corresponding poles cannot be entirely eliminated. Instead we note that in the differential equations for Feynman master integrals it is often observed that that eigenvalues of the residue matrices like $\mathbb{A}_0$ have the form $n + k\varepsilon$. Coincidentally it may be possible to shift these eigenvalues by ±1 via the same transformation eq. (6.3.7); then, once all eigenvalues are of the form $k\varepsilon$, i.e. proportional to ε, it may be possible to transform the whole matrix to also be proportional to ε.

To see how the shifting of eigenvalues can be done, let us start by applying the basis transformation $\mathbf{I} = \mathbb{B}(\mathbb{P}, x_1, x_2; x)\,\mathbf{I}'$. The system of differential equations for $\mathbf{I}'$ then will be given by the matrix

$$\begin{aligned} \mathbb{M}' = {}& \frac{(x_2 - x_1)\,\mathbb{P}\mathbb{B}_0\overline{\mathbb{P}}}{(x-x_2)^2} + \frac{(x_1 - x_2)\,\overline{\mathbb{P}}\mathbb{A}_0\mathbb{P}}{(x-x_1)^2} + \\ &+ \frac{\mathbb{P} + \mathbb{A}_0\mathbb{P} - \overline{\mathbb{P}}\mathbb{A}_0\mathbb{P} + \overline{\mathbb{P}}\mathbb{A}_0 + \overline{\mathbb{P}}\mathbb{B}_0\mathbb{P}}{x - x_1} + \frac{-\mathbb{P} + \mathbb{B}_0\overline{\mathbb{P}} - \mathbb{P}\mathbb{B}_0\overline{\mathbb{P}} + \mathbb{P}\mathbb{B}_0 + \mathbb{P}\mathbb{A}_0\overline{\mathbb{P}}}{x - x_2} + \ldots \end{aligned} \tag{6.3.25}$$

As in the previous section we want the first two terms to disappear, so we must constrain $\mathbb{P}$ by

$$\mathbb{P}\mathbb{B}_0\overline{\mathbb{P}} = 0 \qquad \text{and} \qquad \overline{\mathbb{P}}\mathbb{A}_0\mathbb{P} = 0, \tag{6.3.26}$$

which leaves us with

$$\mathbb{A}_0' = \mathbb{P} + \mathbb{A}_0\mathbb{P} + \overline{\mathbb{P}}\mathbb{A}_0 + \overline{\mathbb{P}}\mathbb{B}_0\mathbb{P}. \tag{6.3.27}$$

Following [30] let us select a right eigenvector of $\mathbb{A}_0$ as $\mathbf{u}$, and a left eigenvector of $\mathbb{B}_0$ as $\mathbf{v}$,

$$\mathbb{A}_0\mathbf{u} = \lambda_1\mathbf{u} \qquad \text{and} \qquad \mathbf{v}^{\mathsf{T}}\mathbb{B}_0 = \lambda_2\mathbf{v}^{\mathsf{T}}; \tag{6.3.28}$$

then if we choose the projector $\mathbb{P}$ as

$$\mathbb{P} = \frac{\mathbf{u}\mathbf{v}^{\mathsf{T}}}{\mathbf{v}^{\mathsf{T}}\mathbf{u}}, \tag{6.3.29}$$

then the conditions of eq. (6.3.26) will be satisfied, and the transformation $\mathbb{B}(\mathbb{P}, x_1, x_2; x)$ will shift λ_1 to $\lambda_1 + 1$ and λ_2 to $\lambda_2 - 1$. To see the shift, consider eq. (6.3.27) if $\mathbb{A}$ was in its Jordan normal form, and $\mathbf{u}$ was its first eigenvector:

$$\mathbb{A}_0 = \begin{pmatrix} \lambda_1 & \# & 0 & \cdots \\ 0 & \# & \# & \cdots \\ 0 & 0 & \# & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \qquad \mathbb{P} = \begin{pmatrix} 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \tag{6.3.30}$$

where # denotes some possibly non-zero values. In this case

$$\mathbb{A}_0\mathbb{P} = \begin{pmatrix} \lambda_1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \qquad \overline{\mathbb{P}}\mathbb{A}_0 = \begin{pmatrix} 0 & 0 & 0 & \cdots \\ 0 & \# & \# & \cdots \\ 0 & 0 & \# & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \qquad \overline{\mathbb{P}}\mathbb{B}\mathbb{P} = \begin{pmatrix} 0 & 0 & 0 & \cdots \\ \# & 0 & 0 & \cdots \\ \# & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \tag{6.3.31}$$

and so the transformed value would look like

$$\mathbb{A}_0' = \begin{pmatrix} \lambda_1 + 1 & 0 & 0 & \cdots \\ \# & \# & \# & \cdots \\ \# & 0 & \# & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \tag{6.3.32}$$

The same transformation in forms other than the Jordan normal form differ only by a similarity transformation; the shift of the eigenvalue remains the same.

Because the same transformation changes $\mathbb{B}$ into

$$\mathbb{B}_0' = -\mathbb{P} + \mathbb{B}_0\overline{\mathbb{P}} + \mathbb{P}\mathbb{B}_0 + \mathbb{P}\mathbb{A}_0\overline{\mathbb{P}}, \tag{6.3.33}$$

we can similarly conclude that it also shifts its eigenvalue λ_2 into $\lambda_2 - 1$.

So the strategy is then to inspect the eigenvalues of all the residue matrices (having the form $n + k\varepsilon$), choose such x_1 and x_2 that there are eignevalues with negative n at x_1 and with positive n at x_2, and apply $\mathbb{B}(\mathbb{P}, x_1, x_2; x)$ that will simultaneously bring both closer to zero ("improve"), with the goal of eventually making all of them proportional to ε.

This overall strategy has several potential issues.

1. Generally it may not possible to construct the projector in eq. (6.3.29) for any arbitrary pair of eigenvectors, because $\mathbf{v}^\mathsf{T}\mathbf{u} \equiv \mathbf{v}\cdot\mathbf{u}$ may be zero. Typically only some subset of eigenvector pairs works.
2. Even if there are many possible pairs of eigenvectors, it might happen that no pair can simultaneously improve eigenvalues at both x_1 and x_2: sometimes only balances between eigenvalues of the same sign of n are available, in which case one of them will be improved, while the other will be worsened. The only solution in this case is to try, and hope that in the next iteration better balances will become possible. No foolproof stopping condition exists here.
3. Sometimes the eigenvalues of the residue matrices do not have the form $n + k\varepsilon$, instead they come as $n/2 + k\varepsilon$. In this case shifting them by ±1 can obviously never reduce them to just $k\varepsilon$. The solution in this case is presented in [121]:
   a) If there are only two different x_i with half-integer eigenvalues (say, x_1 and x_2), then a change of variable is possible that will make the eigenvalues at both points integer:
   $$x = \text{möbius}\big(x_1, t, x_2; y^2\big), \qquad \text{for any } t, \tag{6.3.34}$$
   where
   $$\text{möbius}(a, b, c; y) \equiv \frac{(c-b)a + (b-a)cy}{(c-b) + (b-a)y} \tag{6.3.35}$$
   is a Möbius transformation that maps the point a to $y = 0$, b to $y = 1$, and c to $y = \infty$.
   b) If there are only three different x_i with half-integer eigenvalues (say, x_1, x_2, and x_3), then a change of variable that will make the eigenvalues at all points integer is
   $$x = \text{möbius}\left(x_1, x_2, x_3; \frac{y^2+1}{2y}\right). \tag{6.3.36}$$
   c) If there are four or more different x_i with half-integer eigenvalues, then no change of variables can fix all the points.

6.3.5. Factorizing epsilon dependence

Once $\mathbb{M}$ is normalized, such that it is Fuchsian and all of its eigenvalues are proportional to ε, factorizing its dependence on ε by a transformation constant in x becomes possible. Let us perform a basis transformation with a matrix $\mathbb{T}(d)$, and require that $\mathbb{M}'(d, x) = \varepsilon\mathbb{S}(x)$. From eq. (6.1.6) we have

$$\mathbb{M}'(d,x) = \mathbb{T}^{-1}(d)\,\mathbb{M}(d,x)\,\mathbb{T}(d) = \mathbb{T}^{-1}(d)\sum_i \frac{\mathbb{M}_i(d)}{x - x_i}\mathbb{T}(d) = \varepsilon \sum_i \frac{\mathbb{S}_i}{x - x_i}, \tag{6.3.37}$$

or

$$\mathbb{S}_i = \frac{1}{\varepsilon}\mathbb{T}^{-1}(d)\,\mathbb{M}_i(d)\,\mathbb{T}(d)\,. \tag{6.3.38}$$

Because $\mathbb{S}_i$ does not depend on d, no matter its value, we can demand that

$$\mathbb{S}_i = \frac{1}{\varepsilon}\mathbb{T}^{-1}(4-2\varepsilon)\,\mathbb{M}_i(4-2\varepsilon)\,\mathbb{T}(4-2\varepsilon) = \frac{1}{\mu}\mathbb{T}^{-1}(4-2\mu)\,\mathbb{M}_i(4-2\mu)\,\mathbb{T}(4-2\mu)\,. \tag{6.3.39}$$

Defining

$$\mathbb{T}(\varepsilon,\mu) \equiv \mathbb{T}(4-2\varepsilon)\,\mathbb{T}^{-1}(4-2\mu)\,, \tag{6.3.40}$$

we get a system of linear equations for entries of $\mathbb{T}(\varepsilon,\mu)$:

$$\frac{1}{\varepsilon}\mathbb{M}_i(4-2\varepsilon)\,\mathbb{T}(\varepsilon,\mu) = \frac{1}{\mu}\mathbb{T}(\varepsilon,\mu)\,\mathbb{M}_i(4-2\mu)\,, \qquad \text{for all } i. \tag{6.3.41}$$

This system can be solved by inserting the entries of $\mathbb{T}(\varepsilon,\mu)$ as independent variables and solving for them. The initial transformation $\mathbb{T}(4-2\varepsilon)$ can then be obtained as $\mathbb{T}(4-2\varepsilon) = \mathbb{T}(\varepsilon,\mu_0)$, where the value of μ_0 can be chosen arbitrary, as long as $\mathbb{T}(4-2\varepsilon)$ will come out invertible. In general, the solution for $\mathbb{T}(\varepsilon,\mu)$ can have multiple degrees of freedom, if eq. (6.3.41) is underdetermined; these can also be chosen arbitrary, as long as $\mathbb{T}(4-2\varepsilon)$ is invertible. In both cases it is best to select these values as to simplify the obtained $\mathbb{T}$.

6.3.6. Reducing off-diagonal blocks

Differential equations for Feynman integrals naturally have block-triangular structure: because a differential of a Feynman integral by a kinematic invariant does not introduce new propagators, it must be a combination of only the integrals with the same set of propagators or a subset of them. Then, simply grouping and ordering integrals by their propagator set (from smallest to largest) must make the differential equation matrix $\mathbb{M}$ lower block-triangular, i.e.

$$\mathbb{M} = \begin{pmatrix} \mathbb{M}^{(11)} & 0 & 0 & 0 \\ \mathbb{M}^{(21)} & \mathbb{M}^{(22)} & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ \mathbb{M}^{(n1)} & \mathbb{M}^{(n2)} & & \mathbb{M}^{(nn)} \end{pmatrix}. \tag{6.3.42}$$

Each diagonal block will then correspond to a set of master integrals with the same propagator set but different indices—otherwise they could not have been coupled. Luckily, these blocks are normally very constrained in size, and exploiting the block structure can simplify the reduction process significantly.

Indeed, suppose all the diagonal blocks have been reduced to an ε-form separately,

$$\mathbb{M}^{(ii)}(d,x) = \varepsilon\,\mathbb{S}^{(ii)}(x)\,. \tag{6.3.43}$$

Note that since the characteristic polynomial of a block-triangular matrix is a product of characteristic polynomials of its diagonal blocks, this necessarily means that the whole matrix $\mathbb{M}$ is normalized (its eigenvalues are proportional to ε), even if not fully Fuchsian yet. To transform it to a Fuchsian form, let us consider some off-diagonal block $\mathbb{M}^{(ab)}$. The parts of the differential equation system relevant to it are

$$\begin{cases} \partial_x \mathbf{I}^{(b)} = \mathbb{M}^{(bb)}\mathbf{I}^{(b)} + \ldots, \\ \partial_x \mathbf{I}^{(a)} = \mathbb{M}^{(aa)}\mathbf{I}^{(a)} + \mathbb{M}^{(ab)}\mathbf{I}^{(b)} + \ldots. \end{cases} \tag{6.3.44}$$

If $\mathbb{M}^{(ab)}(d,x)$ has a higher pole at $x=x_i$, then we can try to reduce it by applying a basis transformation

$$\mathbf{I}^{(a)} = \mathbf{I}'^{(a)} + \frac{1}{(x-x_i)^r}\mathbb{D}\,\mathbf{I}^{(b)}, \tag{6.3.45}$$

where $\mathbb{D}$ is some constant matrix. This transformation changes $\mathbb{M}^{(ab)}$ into

$$\mathbb{M}_i'^{(ab)} = \mathbb{M}_i^{(ab)} + r\mathbb{D} + \mathbb{M}_i^{(aa)}\mathbb{D} - \mathbb{D}\,\mathbb{M}_i^{(bb)}. \tag{6.3.46}$$

If $\mathbb{M}^{(aa)}$ and $\mathbb{M}^{(bb)}$ are both in ε-form, then there is always such $\mathbb{D}$ that will make $\mathbb{M}_i'^{(aa)}$ zero. It can be found by substituting $\mathbb{D}$ components and performing a Gaussian elimination. Applying the basis transformation from eq. (6.3.45) then will remove the highest pole of $\mathbb{M}^{(ab)}$ at x_i. We can proceed this way with all the poles; in the end $\mathbb{M}^{(ab)}$ will become purely Fuchsian.

Note that the transformation of eq. (6.3.45) only affects $\mathbb{M}^{(ai)}$ for $i \le a$, and $\mathbb{M}^{(ib)}$ for $i > b$; this means that we can use it to sequentially transform all off-diagonal blocks, and if the blocks are chosen in the order from the first row to the last, and from the last column to the first, then the already transformed blocks will remain so after subsequent transformations. This way all off-diagonal blocks can be made Fuchsian (while keeping the matrix normalized).

Once the whole matrix is Fuchsian and normalized, the ε dependence of the off-diagonal blocks can be factorized the same way the as for the diagonal blocks. The slight simplification is that $\mathbb{T}(\varepsilon,\mu)$ can be taken to have the same block structure as $\mathbb{M}$, reducing the number of variables that need to be solved for.

6.3.7. The case of multiple variables

The algorithm that reduces single-variable differential equations x to an ε-form can be reused in full for the case of multiple variables,

$$\frac{\partial}{\partial x_i}\mathbf{I}(d,\mathbf{x}) = \mathbb{M}(d,\mathbf{x})\;\mathbf{I}(d,\mathbf{x}), \tag{6.3.47}$$

to achieve the combined ε-form, as in eq. (6.1.15). To this end:

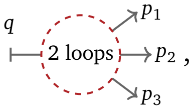
1. Consider only the differential equation system in x_1, and construct a transformation $\mathbf{I} = \mathbb{T}^{(1)}\mathbf{J}^{(1)}$ that reduces it to an ε-form. Treat all the other x_i as free parameters during this reduction.
2. Write down the differential equation system in the second variable, x_2, for the new basis $\mathbf{J}^{(1)}$, and reduce that into an ε-form as well via $\mathbf{J}^{(1)} = \mathbb{T}^{(2)}\mathbf{J}^{(2)}$.
3. If $\mathbb{T}^{(2)}$ is independent of x_1 and d—and this is the case in practice—then it will not spoil the ε-form in x_1, so that the combined transformation $\mathbf{I} = \mathbb{T}^{(1)}\mathbb{T}^{(2)}\mathbf{J}^{(2)}$ will result in an ε-form in both x_1 and x_2.

   Note that because $\mathbb{T}^{(2)}$ is independent of x_1, it is possible to set x_1 to any value (e.g., 0) in step 2, speeding it up significantly.
4. Iterate from step 2 for each of the remaining variables, if any.

6.4. Example: 2-loop 1-to-3 amplitudes

As a useful example, let us consider the 2-loop massless integrals with one incoming off-shell leg and three outgoing on-shell legs ("1 → 3"):

$$I =$$



$$, \qquad p_1^2 = p_2^2 = p_3^2 = 0. \tag{6.4.1}$$

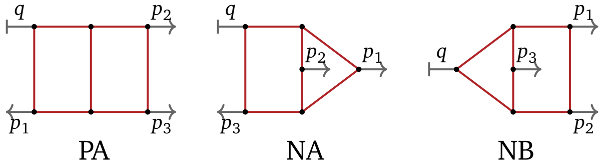


Table 6.4.1.: Topologies for the 1 → 3 master integrals at two loops.

These integrals depend on three independent external invariants: $p_1 \cdot p_2$, $p_1 \cdot p_3$, and $p_2 \cdot p_3$, with the incoming energy q^2 being twice the sum of them (because $q = p_1 + p_2 + p_3$). This allows us to consider the integrals as functions of one of the invariants and two dimensionless variables; for example

$$I(p_1, p_2, p_3) = \left(q^2\right)^k f(y, z), \tag{6.4.2}$$

where k is determined from dimensionality ($+d/2$ for each loop, -1 for each denominator), and we define the dimensionless variables as

$$x \equiv s_{12}, \qquad y \equiv s_{13}, \qquad \text{and } z \equiv s_{23}, \qquad \text{with } s_{ij} \equiv \left(p_i + p_j\right)^2 / q^2. \tag{6.4.3}$$

The master integrals for these integrals were first calculated in [122, 116].[11] In those works the answers are provided in terms of "2d HPLs" (a subclass of multiple polylogarithms of two variables, see Chapter 7), up to transcendental weight 4. Later in Section 9.3.4 we will find it useful to know these integrals at least up to weight 7; for this reason we have recomputed them up to weight 8 in [35]. Let us see how this is done.

To obtain differential equations for the master integrals, we first we need to define the IBP topologies. There are three of them: PA, NA, and NB, all listed in Table 6.4.1. Then, we can use e.g. LiteRed to find a list of master integrals and compute differential equations in both y and z for each topology. The master integrals for PA obtained this way are listed in Table 6.4.2, for NA in Table 6.4.3, and for NB in Table 6.4.4. Note that there is a lot of duplication between the integrals in different topologies; the ones with the highest number of denominators are all different though.

Once the differential equations are ready, then for each set of master integrals $\mathbf{I}$ we can proceed as described in the previous section to transform them into a combined ε-form (or, rather use Fuchsia from Chapter 8 to do this automatically),

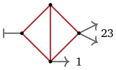
$$\begin{cases} \partial_y \mathbf{J}(d, y, z) = \varepsilon \mathbb{S}^{(y)}(y, z)\, \mathbf{J}(d, y, z), \\ \partial_z \mathbf{J}(d, y, z) = \varepsilon \mathbb{S}^{(z)}(y, z)\, \mathbf{J}(d, y, z), \\ \mathbf{I}(d, y, z) = \mathbb{T}(d, y, z)\, \mathbf{J}(d, y, z), \end{cases} \tag{6.4.4}$$

where the $\mathbb{S}$ matrices have the general form of

$$\mathbb{S}(y, z) = \sum_i \frac{\mathbb{S}_i}{D_i}, \qquad \text{with } D_i \in \{y, z, \underbrace{1 - y - z}_{x}, 1 - y, 1 - z, \underbrace{y + z}_{1-x}\}. \tag{6.4.5}$$

Then, we can use eq. (6.1.16) to write down the solution for each integral I_i in the form

$$I_i = \left(q^2\right)^k N \sum C\, G(\ldots, a_i, \ldots; z)\, G(\ldots, b_i, \ldots; y), \tag{6.4.6}$$

[11] Note that in [81] the same authors provide a subset of these integrals in arbitrary d in terms of hypergeometric functions. We have found that at least eq. (5.24) from that work, listing the value of , is wrong. In [116] the same integral is given as a series in ε in eq. (4.17)—that value is correct, and matches our calculations in this section. We have confirmed this error with one of the authors.

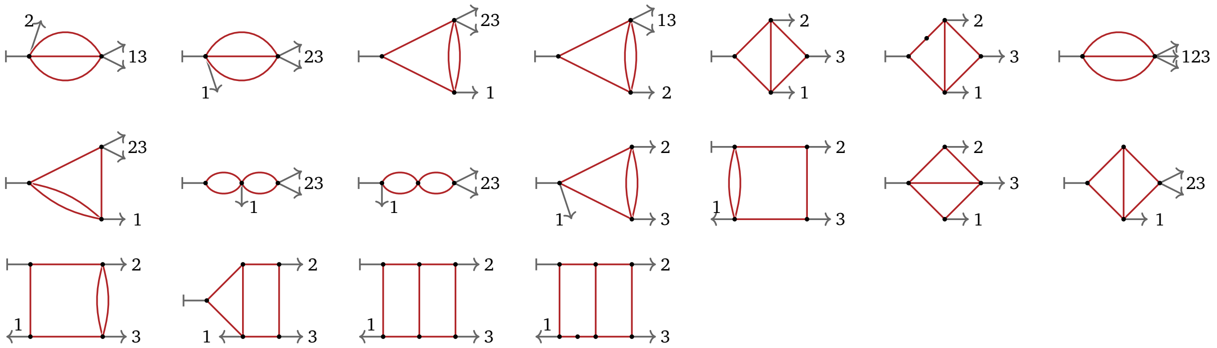


Table 6.4.2.: All the 18 master integrals from the PA topology.

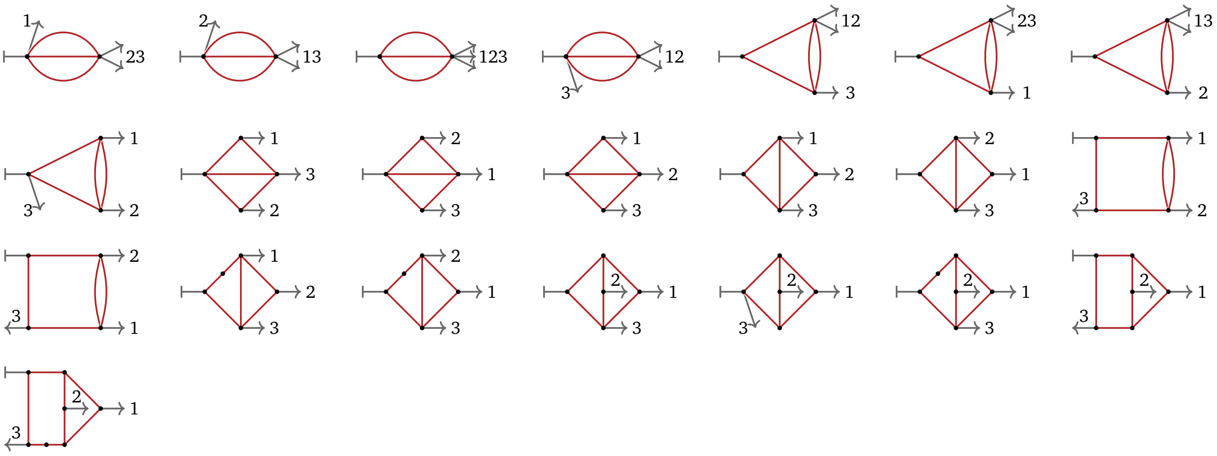


Table 6.4.3.: All the 22 master integrals from the NA topology.

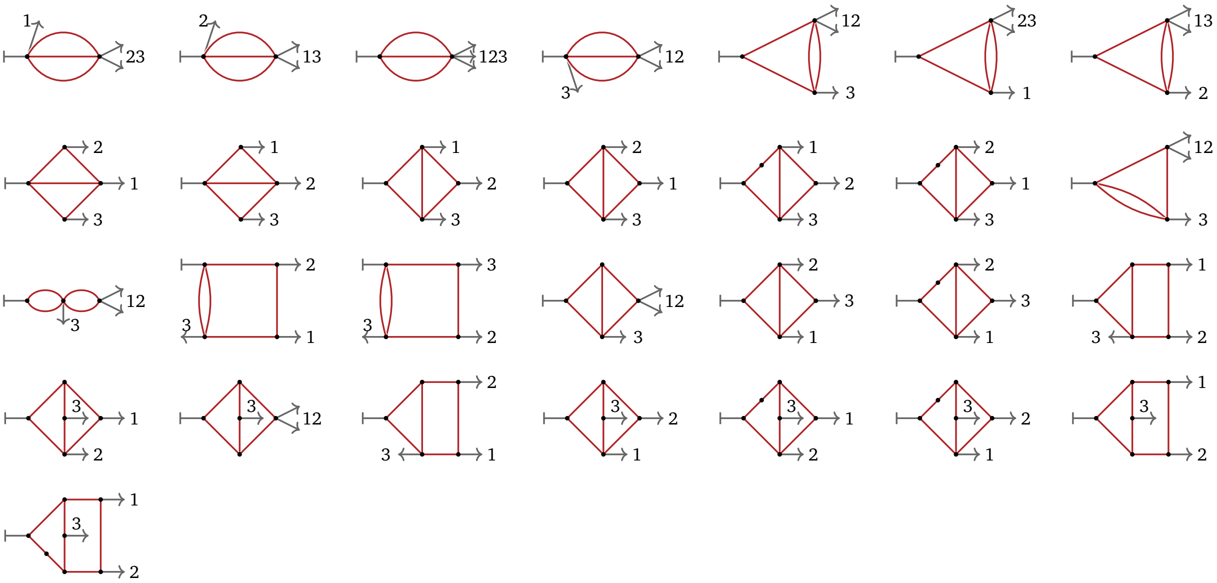


Table 6.4.4.: All the 29 master integrals from the NB topology.

where

$$a_i \in \{0, 1, 1-y, -y\}, \qquad \text{and} \qquad b_j \in \{0, 1\}, \tag{6.4.7}$$

k follows from dimensionality, and N is a common normalization prefactor,

$$N \equiv (-1-i0)^{-2\varepsilon} \left((4\pi)^{\varepsilon-2} \frac{\Gamma(1+\varepsilon)\,\Gamma^2(1-\varepsilon)}{\Gamma(1-2\varepsilon)} \right)^2, \tag{6.4.8}$$

which we have factored out this way to match the prefactors in [122, 116]: in their notation N is just $(-1-i0)^{-2\varepsilon}\left(S_\varepsilon/\left(16\pi^2\right)\right)^2$.

Finally, to fix the integration constants $\mathbf{C}(d)$ (expanded in ε as in eq. (6.1.10)), we can use the following observations:

1. A number of simpler master integrals only depend on one or two scales; all of these are known for arbitrary d in terms of Gamma functions Γ, and hypergeometric functions ${}_2F_1$and ${}_3F_2$. For these integrals the series obtained via eq. (6.1.16) can be compared with the known series, and all the $C_i^{(k)}$ that define these integrals can be obtained from this comparison.
2. All of our integrals are massless, and therefore must not have any discontinuities other than from limits $\left(p_i+p_j\right)^2 \to 0$. On the other hand, the differential equations we are solving also have poles at $\left(p_i+p_j\right)^2 \to q^2$, as the list of denominators in eq. (6.4.5) demonstrates. Thus, requiring that the apparent discontinuities of the general solution at $y, z, 1-y-z \to 1$ vanish will generate nontrivial identities between the integration constants. This requirement can be written down by separating the terms proportional to $\ln(1-y)$, $\ln(1-z)$, or $\ln(y+z)$, and enforcing that the coefficient in front of each vanishes in the limit.
3. The planar integrals only have discontinuities at limits where adjacent momenta go to zero. For the PA topology this means that it should be regular at $y+z \to 1$ (i.e. $(p_1+p_2)^2 \to 0$), as long as $q^2 \neq 0$. Similarly for the planar integrals from other topologies. Here again we are looking at the logarithmic terms like $\ln(1-y-z)$, enforcing the cancellation of the coefficients in front of them in the limit.

To apply these regularity conditions one needs to separate the terms proportional to $\ln^k\left(1-s_{ij}\right)$, and require that the coefficient of each is exactly zero in the limit $s_{ij} \to 1$. For the limit $y \to 1$, to separate the divergent logarithms, it is enough to employ the shuffle relations to rewrite every $G\left(1,\ldots,1,w_y;y\right)$ in eq. (6.4.6) into a product of the divergent factor $G(1,\ldots,1;y)$ and the part finite at $y \to 1$. For the limit $z \to 1$ the same cannot be done directly on eq. (6.4.6), because z appears in the parameter list of $G(\ldots;y)$. Instead, we can rewrite it into the reverse form,

$$I_i = \left(q^2\right)^k N \sum C\, G\left(\ldots,a_i',\ldots;z\right) G\left(\ldots,b_i',\ldots;y\right), \tag{6.4.9}$$

where

$$a_i \in \{0, 1\}, \qquad \text{and} \qquad b_j \in \{0, 1, 1-z, -z\}, \tag{6.4.10}$$

and factor our logarithmic terms from that. The general procedure for this transformation is described in Section 7.10; up to transcendental weight 9 we have found the `fibrationBasis` routine from HyperInt [123] to be a convenient way to do this; at weight 10 the results contain too many terms for HyperInt to handle with ease, so we had to implement this transformation manually in C++ (with the help of GiNaC [102]) instead. The need to operate on weight 10 expressions comes from the fact that applying the regularity conditions to the weight-10 expansion of I_i only fixes the constants $C_i^{(k)}$ up to weight 8.

The weight-8 results for these integrals weight megabytes of text, and we have provided them in auxiliary files attached to [35].

7. Multiple polylogarithms

If the matrix $\mathbb{S}$ defining the ε-form differential equation system in eq. (6.1.7) has the form

$$\mathbb{S}(x) = \sum_i \frac{\mathbb{S}_i}{x - x_i}, \tag{7.0.1}$$

then the integrals that appear as the result of the ε-form solution given by eq. (6.1.9) will be iterated integrals of the form

$$G(w_1, w_2, \dots, w_n; x) \equiv \int_0^x \frac{\mathrm{d}t_1}{t_1 - w_1} \int_0^{t_1} \frac{\mathrm{d}t_2}{t_2 - w_2} \cdots \int_0^{t_n} \frac{\mathrm{d}t_n}{t_n - w_n}. \tag{7.0.2}$$

Together with a special definition for the otherwise divergent case of all $w_i = 0$,

$$G(0, \dots, 0; x) \equiv \frac{\ln^n(x)}{n!}, \tag{7.0.3}$$

these are *multiple polylogarithms* [124], also called *Goncharov polylogarithms* or *hyperlogarithms*.

An alternative representation for G as an infinite sum is given by

$$\mathrm{Li}_{m_1,\dots,m_n}(x_1, \dots, x_n) \equiv \sum_{i_1 > \dots > i_n > 0} \frac{x_1^{i_1}}{i_1^{m_1}} \cdots \frac{x_n^{i_n}}{i_n^{m_n}} = (-1)^n\, G\Big(\underbrace{0, \dots, 0}_{m_1 - 1}, \frac{1}{x_1}, \underbrace{0, \dots, 0}_{m_2 - 1}, \frac{1}{x_1 x_2}, \dots, \frac{1}{x_1 \cdots x_n}; 1\Big). \tag{7.0.4}$$

with the case of trailing 0 defined through shuffling relations (see Section 7.4).

Multiple polylogarithms is a well studied and convenient class of functions. There exist publicly available libraries that can transform them: HyperInt [123] and PolyLogTools [125], and evaluate them numerically with arbitrary precision: GiNaC [126, 102].

Let us review the main properties of multiple polylogarithms, as they are relevant to our calculations.

7.1. Transcendental weight and grading

Transcendental weight, or simply *weight*, of a multiple polylogarithm is the number of w_i parameters it has. A sum of terms of identical weight has the same weight as its terms. A product has weight equal to the sum of weights of its factors, unless the whole product is zero.

Conjecturally, multiple polylogarithms form an algebra graded by weight, which means that no linear combination of Gs of a uniform weight with coefficients and arguments being rational functions is ever equal to a combination of a different weight, unless both are zero. This conjecture is related to the multiple zeta value grading conjecture [127].

Irrational constants related to multiple polylogarithms are also weighted. The weight of rational numbers is zero. Because $G(0, 1; 1) = \pi^2/6$, the weight of π is 1. The weight of $\ln(2) = G(0; 2)$ is also 1.

7.2. Related classes of functions

Multiple polylogarithms subsume a number of well known (and useful in their own right) classes of functions. These include the following.

7.2.1. Natural logarithms

Natural logarithms are equivalent to multiple polylogarithms of weight 1 and vice versa:

$$\ln(x) = G(0;x) \qquad \text{and} \qquad G(a;x) = \ln\Big(\frac{a-x}{a}\Big). \tag{7.2.1}$$

7.2.2. Dilogarithms and (classical) polylogarithms

Polylogarithms [128] are defined in terms of an infinite sum,

$$\mathrm{Li}_n(x) \equiv \sum_{k=1}^{\infty} \frac{x^k}{k^n}, \tag{7.2.2}$$

and dilogarithms are just Li_2. When $n > 0$ these allow for a recursive integral definition:

$$\mathrm{Li}_n(x) = \int_0^x \frac{\mathrm{Li}_{n-1}(t)}{t} \mathrm{d}t, \tag{7.2.3}$$

with the base case of

$$\mathrm{Li}_1(x) = -\ln(1-x). \tag{7.2.4}$$

Polylogarithms are a special case of weight-n multiple polylogarithms:

$$\mathrm{Li}_n(x) = -G(\underbrace{0,\ldots,0}_{n-1},1;x). \tag{7.2.5}$$

Polylogarithms with $n \le 0$ are rational functions.

7.2.3. Harmonic polylogarithms

Harmonic polylogarithms [129] are defined as iterated integrals of the form

$$\mathrm{H}(w_1,w_2,\ldots,w_n;x) = \int_0^x h(w_1;t)\,\mathrm{H}(w_2,\ldots,w_n;t)\,\mathrm{d}t, \tag{7.2.6}$$

where the parameters $w_i \in \{-1,0,1\}$, and the integration kernels are

$$h(w;x) \equiv \begin{cases} \frac{1}{1-x} & \text{if } w=1, \\ \frac{1}{x} & \text{if } w=0, \\ \frac{1}{1+x} & \text{if } w=-1. \end{cases} \tag{7.2.7}$$

These are a subset of multiple polylogarithms with the parameters w_i restricted to $\{-1,0,1\}$, and differ only by an overall sign:

$$\mathrm{H}(w_1,\ldots,w_n;x) = (-1)^{\delta_{w_1,1}+\cdots+\delta_{w_n,1}}\,G(w_1,\ldots,w_n;x). \tag{7.2.8}$$

Note that there is a commonly adopted shorthand notation for the parameters of H where a sequence of $n-1$ zeros followed by a 1 is denoted as just n, and a sequence of $n-1$ zeros followed by a -1 is denoted $-n$:

$$\mathrm{H}(\ldots,\pm n,\ldots;x)\equiv\mathrm{H}(\ldots,\underbrace{0,\ldots,0}_{n-1},\pm 1,\ldots;x). \tag{7.2.9}$$

The HPL library [130, 131] can be useful to transform and evaluate these functions.

7.2.4. Nielsen's generalized polylogarithms

Nielsen's generalized polylogarithms [132],

$$S_{n,p}(x)\equiv\frac{(-1)^{n+p-1}}{(n-1)!p!}\int_0^1\frac{\ln^{n-1}(t)\ln^p(1-xt)}{t}\mathrm{d}t, \tag{7.2.10}$$

can be expressed as a particular case of harmonic polylogarithms, and thus multiple polylogarithms:

$$S_{n,p}(x)=\mathrm{H}(\underbrace{0,\ldots,0}_{n},\underbrace{1,\ldots,1}_{p};x)=(-1)^p\,G(\underbrace{0,\ldots,0}_{n},\underbrace{1,\ldots,1}_{p};x). \tag{7.2.11}$$

7.2.5. Multiple zeta values and alternating sums

Multiple zeta values (MZV) and alternating sums [108] are a class of irrational constants (rather than functions) closely related to harmonic polylogarithms:

$$\zeta_{a,b,\ldots}\equiv\sum_{n_1>n_2>\cdots>0}\frac{\mathrm{sgn}^{n_1}(a)}{n_1^{|a|}}\frac{\mathrm{sgn}^{n_2}(b)}{n_2^{|b|}}\cdots. \tag{7.2.12}$$

With all non-negative parameters these are multiple zeta values proper, and they are equivalent to harmonic polylogarithms (and thus, multiple polylogarithms) of argument 1:

$$\zeta_{a,b,\ldots}=\mathrm{H}(a,b,\ldots;1)\qquad\text{if } a,b,\cdots\geq 0, \tag{7.2.13}$$

where the shorthand notation of eq. (7.2.9) is assumed.

With some parameters being negative these are called alternating sums rather than MZVs proper, and their relation to harmonic polylogarithms is also straightforward:

$$\zeta_{w_1,\ldots,w_n}=\left(\prod_{i=1}^n s_i\right)\mathrm{H}(s_1\,|w_1|,\ldots,s_n\,|w_n|;1),\qquad\text{where } s_i=\prod_{j=1}^i\mathrm{sign}\big(w_j\big). \tag{7.2.14}$$

Note that eq. (7.2.12) is a common "physicist" notation for multiple zeta values and alternating sums; it makes their relation to harmonic polylogarithms straightforward. On the other hand, the same notation but with the order of parameters reversed (the "mathematician" notation) is also occasionally used in the literature and software. Notably, [124] and HyperInt [123] use the latter, while GiNaC [126], HPL [131], `SpecialFunctions'MultipleZetaValue[{a,b,...}]` from Mathematica, and the MZV Data Mine [108] use the former.

7.3. Linear independence

All multiple polylogarithms $G(\mathbf{w}; x)$ with different parameters $\mathbf{w}$ but identical arguments x are linearly independent (assuming the parameters do not depend on the argument), meaning that no linear combination of them with rational coefficients is zero.

Multiple polylogarithms of unequal weights in general are linearly independent (with coefficients being rational functions) by the grading conjecture.

7.4. Shuffling relations

Multiple polylogarithms obey the shuffling relations:

$$G(\mathbf{p}; x)\, G(\mathbf{q}; x) = \sum_{\mathbf{w} \in \mathbf{p}//\mathbf{q}} G(\mathbf{w}; x)\,, \tag{7.4.1}$$

where $\mathbf{p}//\mathbf{q}$ denotes the set of shuffles of sequences $\mathbf{p}$ and $\mathbf{q}$, i.e. such permutations of the combined set of their elements that preserve the relative order among elements of $\mathbf{p}$ and of $\mathbf{q}$ separately. For example:

$$(1,2)//(x,y) = \{(1,2,x,y), (1,x,2,y), (1,x,y,2), (x,1,2,y), (x,1,y,2), (x,y,1,2)\}\,. \tag{7.4.2}$$

The shuffle operation can be defined recursively as

$$(x,\mathbf{p})//(y,\mathbf{q}) = x\cdot(\mathbf{p}//(y,\mathbf{q})) \cup y\cdot((x,\mathbf{p})//\mathbf{q})\,, \tag{7.4.3}$$

where “·” denotes sequence concatenation.

7.5. Divergences

$G(w_1, \ldots, w_n; x)$ is finite if $x \neq w_1$, otherwise it may diverge. Its precise behavior when $x \to w_1$ can be extracted via the reverse application of the shuffling relations (see Section 7.4). As a general rule this divergence is logarithmic:

$$G(\underbrace{a, \ldots, a}_{n}, \mathbf{w}; x) = G(\underbrace{a, \ldots, a}_{n}; x) G(\mathbf{w}; x) + \mathcal{O}(G(\underbrace{a, \ldots, a}_{n-1}; x)), \tag{7.5.1}$$

where

$$G(\underbrace{a, \ldots, a}_{n}; x) = \begin{cases} \frac{1}{n!} \ln\left(\frac{a-x}{x}\right) & \text{if } a \neq 0, \\ \frac{1}{n!} \ln(x) & \text{if } a = 0. \end{cases} \tag{7.5.2}$$

An important corollary is that even though $G(\mathbf{w}; x)$ may diverge, its integral over any finite interval is always finite, because the divergence is at most logarithmic.

7.6. Scaling

If the last parameter $w_n \neq 0$, from the integral representation of eq. (7.0.2) it can be seen that rescaling the parameters and the argument of G does not change its value:

$$G(\alpha \mathbf{w}; \alpha x) = G(\mathbf{w}; x)\,. \tag{7.6.1}$$

If there are trailing zeros in the parameter list ($w_n \neq 0$), this scaling is not observed, but then the shuffling relations can be first applied to split the multiple polylogarithms into $G(0, \ldots, 0; x)$ (the logarithms) and $G(\ldots, w_n; x)$ with $w_n \neq 0$. Once this is done, the scaling relation can be used again.

7.7. Series expansion around zero

It is sometimes useful to know (and easy to calculate) the behavior of multiple polylogarithms around $x = 0$. If $w_n \neq 0$, the expansion can be done directly through the sum representation of eq. (7.0.4):

$$G(\underbrace{0,\dots,0}_{m_1-1},w_1,\underbrace{0,\dots,0}_{m_2-1},w_2,\dots,w_n;x) = (-1)^n \sum_{i_1>\dots>i_n>0} \frac{(x/w_1)^{i_1}}{i_1^{m_1}} \frac{(w_1/w_2)^{i_2}}{i_1^{m_2}} \cdots \frac{(w_{n-1}/w_n)^{i_m}}{i_n^{m_n}}. \tag{7.7.1}$$

In particular, if all parameters of G are non-zero, then the expansion starts with the following term:

$$G(w_1,\dots,w_n;x) = \frac{(-1)^n}{n!}\frac{x^n}{w_1\cdots w_n} + \mathcal{O}\big(x^{n+1}\big), \qquad \text{if } w_i \neq 0. \tag{7.7.2}$$

In the case when w_n is zero, multiple polylogarithms diverge logarithmically at $x \to 0$, and the full series expansion can be obtained by first using the shuffling relations to make the logarithms explicit and thus eliminate the trailing zero parameters, and then safely using eq. (7.7.1) on the remaining multiple polylogarithms.

7.8. Differentiation

In the simple case when the parameters w_i do not depend on x, the derivative by x directly follows from the definition eq. (7.0.2):

$$\frac{\mathrm{d}}{\mathrm{d}x}G(w_1,w_2,\dots,w_n;x) = \frac{1}{x-w_1}G(w_2,\dots,w_n;x), \qquad \text{if } \frac{\mathrm{d}}{\mathrm{d}x}w_i = 0. \tag{7.8.1}$$

The most general case can be compactly given as

$$\frac{\mathrm{d}}{\mathrm{d}x}G(w_1(x),\dots,w_n(x);y(x)) = \sum_{i=1}^{n} G(w_1,\dots,\cancel{w_i},\dots,w_n;y)\,\frac{\mathrm{d}}{\mathrm{d}x}\ln\!\left(\frac{w_i-w_{i-1}}{w_i-w_{i+1}}\right), \tag{7.8.2}$$

where $\cancel{w_i}$ denotes the operation of removing the specified item from the sequence, and we have introduced auxiliary values $w_0 \equiv y$ and $w_{n+1} \equiv 0$.

7.9. Integration

The simplest case of integration is already given by the definition in eq. (7.0.2),

$$\int \mathrm{d}x\,\frac{1}{x-w_1}G(w_2,\dots,d_n;x) = G(w_1,w_2,\dots,w_n;x) + C, \qquad \text{if } \frac{\mathrm{d}}{\mathrm{d}x}w_i = 0. \tag{7.9.1}$$

More generally, if w_i do depend on x, then as a first step one should transform G to eliminate this dependency. This is always possible, and the procedure for this is described in Section 7.10.

If a product of multiple G functions is to be integrated, one should first use the shuffling relations to expand this product into a sum.

Finally, let us consider kernels of integration other than $1/(x-w)$. To integrate a multiple polylogarithm multiplied by an arbitrary rational function one must first perform a partial fraction decomposition on this function. Then the most general remaining expression will be of the form

$$\int \frac{1}{(x-a)^k} G(w_1,\ldots,w_n;x)\,\mathrm{d}x \qquad \text{or} \qquad \int x^p G(w_1,\ldots,w_n;x)\,\mathrm{d}x, \tag{7.9.2}$$

where $k > 0$ and $p \geq 0$. Any integral of the first form can be taken by a recursive application of the following rules:

$$\int \frac{1}{(x-a)^{p+1}} G(u,\mathbf{w};x)\,\mathrm{d}x = -\frac{1}{p}\frac{1}{(x-a)^p} G(u,\mathbf{w};x) + \frac{1}{p}\int \frac{1}{(x-a)^p}\frac{1}{x-u} G(\mathbf{w};x)\,\mathrm{d}x, \tag{7.9.3}$$

$$\frac{1}{(x-a)^p}\frac{1}{x-b} = \frac{1}{a-b}\frac{1}{(x-a)^p} + \frac{1}{b-a}\frac{1}{(x-a)^{p-1}}\frac{1}{x-b}, \tag{7.9.4}$$

with the base case of the recursion given by eq. (7.9.1) and

$$\int \frac{1}{(x-a)^{p+1}} G(;x)\,\mathrm{d}x = -\frac{1}{p}\frac{1}{(x-a)^p} + C \qquad \text{if } p > 0. \tag{7.9.5}$$

The second form of eq. (7.9.2) can be taken by recursively applying the following rules:

$$\int x^{k-1} G(u,\mathbf{w};x)\,\mathrm{d}x = \frac{x^k}{k} G(u,\mathbf{w};x) - \frac{1}{k}\int \frac{x^k}{x-u} G(\mathbf{w};x)\,\mathrm{d}x, \tag{7.9.6}$$

$$\int \frac{x^k}{x-a} G(\mathbf{w};x)\,\mathrm{d}x = \int x^{k-1} G(\mathbf{w};x)\,\mathrm{d}x + a\int \frac{x^{k-1}}{x-a} G(\mathbf{w};x)\,\mathrm{d}x, \tag{7.9.7}$$

$$\int G(u,\mathbf{w};x)\,\mathrm{d}x = (x-u)\,G(u,\mathbf{w};x) - \int G(\mathbf{w};x)\,\mathrm{d}x, \tag{7.9.8}$$

with the base case given by eq. (7.9.1) and

$$\int x^{k-1} G(;x)\,\mathrm{d}x = \frac{x^k}{k} + C. \tag{7.9.9}$$

Together these rules are enough to integrate any product of rational functions and multiple polylogarithms.

7.10. Parameter transformations

If one has a multiple polylogarithm $G(\mathbf{w}(x);y(x))$, with both its parameters $\mathbf{w}$ and the argument y depending on a given variable x, it is often needed to rewrite it in terms of multiple polylogarithms of argument x, $G(\mathbf{w}';x)$, such that x enters nowhere else, $\mathbf{w}' \neq \mathbf{w}'(x)$. This is needed for example if one wants to integrate $G(\mathbf{w}(x);y(x))$ over x, or if one is interested in mapping a complicated expression onto a linearly independent basis.

As advocated in [133] a general way to achieve this is to recursively differentiate by x and then integrate back:

$$G(\mathbf{w}(x);y(x)) = \int \mathrm{d}x \frac{\partial}{\partial x} G(\mathbf{w}(x);y(x)) + C. \tag{7.10.1}$$

This works because differentiation will reduce the weight of $G(\mathbf{w};y)$ by 1, and applied recursively it will reach zero at some point; then, integrating the expression back will always give the form $G\big(\mathbf{w}';x\big)$.

As a simple example, let us take a look at $G(1-x;y)$. Applying eq. (7.10.1) to it we get

$$G(1-x;y)=\int\left(\frac{1}{x+y-1}-\frac{1}{x-1}\right)\mathrm{d}x+C=G(1-y;x)-G(1;x)+C, \tag{7.10.2}$$

where C can be determined from the limit of $x\to 0$. This limit is trivial, and we have

$$C=\lim_{x\to 0}(G(1-x;y)-G(1-y;x)+G(1;x))=G(0;y)\,. \tag{7.10.3}$$

In many cases such limits are not trivial, and finding the integration constants C is the main difficulty of the method. For example, consider

$$G(x;y)=\int\left(\frac{1}{x-y}-\frac{1}{x}\right)\mathrm{d}x+C=G(y;x)+G(0;x)+C, \tag{7.10.4}$$

where C can also be determined from the limit of $x\to 0$,[1]

$$C=\lim_{x\to 0}(G(x;y)-G(y;x)+G(0;x))\,, \tag{7.10.5}$$

but the complication here is that

$$\lim_{x\to 0}G(0;x)=\infty, \qquad \text{and} \qquad \lim_{x\to 0}G(x;y)\neq G(0;y)\,. \tag{7.10.6}$$

The first of these means that the limits can only be taken separately after regularizing the divergences, for example by expanding the terms into a series in $\ln(x)$, and keeping the logs. The second is because $G(0;y)$ is the exceptional case in the G definition of eq. (7.0.2),

$$G(0;y)\equiv\ln(y)=\int_\delta^y\frac{\mathrm{d}t}{t}+\ln(\delta)\neq\int_0^y\frac{\mathrm{d}t}{t}. \tag{7.10.7}$$

Additionally, keep in mind that $\lim_{x\to 0}G(x;y)$ assumes that $x<y$, which means that the integration contour crosses a singularity at $y=x$. To properly define this function we can give a small imaginary part to x, $\pm i\delta_x$, so that the integration contour is shifted to avoid the singularity. The next step is to split the integration contour and take each piece separately

$$\begin{aligned}\lim_{x\to 0}G(x;y)&=\lim_{x,\delta\to 0}\left(\int_0^{x-\delta}\frac{\mathrm{d}t}{t-x}+\int_{x-\delta}^{x+\delta}\frac{\mathrm{d}t}{t-x}+\int_{x+\delta}^{y}\frac{\mathrm{d}t}{t-x}\right)=\\&=\ln\delta-\ln x\pm i\pi+\ln y-\ln\delta=\\&=G(0;x)\pm i\pi+G(0;y)\,,\end{aligned} \tag{7.10.8}$$

where the sign of ± is the same as in $\pm i\delta_x$. The whole limit then becomes

$$C=G(0;y)\pm i\pi. \tag{7.10.9}$$

For higher weights the same procedure needs to be applied recursively.

[1] Of course the derivation here would have been easier if we were to immediately rewrite G in terms of ln, but this is only possible at weight 1, and we want to illustrate the general case.

A general algorithm for finding these integration constants is described in [123] and available as a MAPLE package HYPERINT. That method is based on taking limits of both sides of eq. (7.10.1) at $x \to \infty$, regularized by throwing away all terms proportional to $\ln x$.

A different approach is pursued by POLYLOGTOOLS [125]: the constants are found by evaluating both the left- and the right-hand sides of eq. (7.10.1) numerically at some fixed value of x using GINAC [126, 102] up to hundreds of digits of precision, and then by reconstructing the analytical expression for C from its numerical value using integer reconstruction algorithm PSLQ [109] in the basis of multiple zeta values of appropriate weight. The biggest drawback of this method is performance at higher weights: at each weight the size of the linear basis of multiple zeta values grows exponentially, the number of digits of precision one needs to use is linear in the size of the basis, and GINAC takes time approximately quadratic in the requested number of digits.

8. Fuchsia: a tool for epsilon form construction

We have implemented the algorithms of reduction to an ε-form described so far in FUCHSIA. The first version of it was announced in [38] and fully presented in [37]. That version was written in Python using the free computer algebra system SAGEMATH [134]; it is available at github.com/gituliar/fuchsia. Over time we have experienced several major problems with the way FUCHSIA was made:

1. Its performance was limited on larger problems (i.e. matrices with deep poles or with sizes larger than 20×20).
2. Simply installing SAGEMATH turned out to be a constant source of problems. It is fairly big (more than 2GB archived), and only comes in a prebuilt form for a limited set of Linux distributions; building it from source takes a long time and often fails.
3. Algebraic extensions in the matrices (that is, complex numbers and roots)—although recognized—would often result in expression blowup, so that the computation would never finish. The reason is that SAGEMATH (or, more precisely, MAXIMA [135], which it uses under the hood) does not simplify ratios of such numbers to a normal form, and factors that should have been canceled linger on. This is one of the reasons FUCHSIA acquired the ability to use MAPLE to help its computations.
4. No support for matrices with spurious poles that depend on ε. These poles can be completely eliminated as a preliminary step, but in practice they often come in the form of polynomials with higher powers of x, so either introduction of roots is needed (see issue 3), or possibly an approach that deals with higher-order polynomials explicitly like [136] can be used.

   An alternative is to always start with a d-factorizing master integral basis, in which this problem does not exist. See [94, 93] for ways to do this.

To address points 1 and 2, FUCHSIA was rewritten in C++ using GINAC [102]. The source code and the prebuilt binaries of this version are available at github.com/magv/fuchsia.cpp. The 3-rd point is still unaddressed; the 4-th is only partially addressed: the new FUCHSIA is able to reduce matrices with ε-dependent poles under limited conditions (in fact we use this property in Chapter 10), but this functionality is far from complete, and we are yet to guarantee correctness in all cases—instead we advocate using d-factorizing bases for this problem, because they also improve the performance of the IBP reduction, which is often a major bottleneck, so most calculations would benefit—and some even require—such choice anyway.

8.1. How does it work?

We have discussed the general method of reducing a differential equation system to an ε-form in Section 6.3, but it allows for considerable freedom: transformations that reduce higher poles and those that normalize Fuchsian poles can be constructed in different orders, and each transformation has several possible $\mathbb{P}$ matrices that can be used.

The strategy FUCHSIA uses for univariate differential equation systems is this:

1. The initial matrix $\mathbb{M}$ is normalized into partial fraction form in x, as in eq. (6.1.2). Following the advice of [137], all the subsequent transformations are applied directly to this form, maintaining it.

2. The matrix is shuffled into a block-triangular form (if not already so) using the algorithm of [138].

3. For each block on the diagonal:

   a) All higher poles are eliminated, transforming $\mathbb{M}$ into a Fuchsian form. This step is called *fuchsification*.

      A major practical concern in this step—as well as others—is the intermediate expression swell. If one chooses the points to reduce randomly, the size of $\mathbb{M}$ will tend to grow, slowing down all operations, potentially preventing practical application completely. To combat this, at each step FUCHSIA constructs all $\mathbb{B}(\mathbb{P}, x_1, x_2; x)$ for all possible x_i, x_j, and $\mathbb{P}$; then the one that produces the "simplest" overall $\mathbb{M}$ is applied.[1] If the matrix is Moser-reducible but no $\mathbb{B}$ can be constructed because none of the $\mathbb{V}\mathbb{U}$ combinations from eq. (6.3.24) are invertible, then a fresh x is chosen and a spurious pole is introduced there (such cases are rare in practice).

      To guarantee progress, we only reduce from points with higher pole power to the ones with lower or equal. In the latter case, we also make sure the sum of ranks of $\mathbb{A}_0$ and $\mathbb{B}_0$ is decreasing after the transformation (this does not always happen because the rank of $\mathbb{B}_0$ is often increased by the transformation).

      This does mean that $\mathcal{O}\big(\text{number of poles}^2\big)$ of $\mathbb{B}$ matrices need to be constructed and applied at every step—but as long as the expression swell for the bigger matrices is prevented, this is worth it. (For smaller matrices this effort is mostly wasted, but we are not concerned about these too much).

   b) All Fuchsian poles are normalized. This step is called *normalization*.

      The approach is similar here: construct all possible $\mathbb{B}(\mathbb{P}, x_1, x_2; x)$, sort them by how close is the resulting $\mathbb{M}$ to being fully normalized and how big it is, and apply the best one. The difference is that in the case of fuchsification Moser's reducibility criterium guarantees progress; there is no such guarantee here, and often there are no transformations available that make $\mathbb{M}$ more normalized. To handle such cases FUCHSIA considers the transformations that denormalize $\mathbb{M}$ too, and keeps a priority queue of all $\mathbb{M}$ states constructed, so if one sequence of reduction steps would lead to a dead end, the search could continue with a different path.

   c) The dependence of $\mathbb{M}$ on ε is factorized ("*factorization*").

      When constructing the factorization matrix $\mathbb{T}(d)$, there are multiple degrees of freedom that can be chosen arbitrary (as discussed in Section 6.3.5); FUCHSIA tries to set all of them to random small integers, preferably zeros, and once a few versions of $\mathbb{T}(d)$ that are invertible are identified, the "simplest" of them is applied.

4. After each diagonal block is in an ε-forms, each off-diagonal block is reduced to it too, one by one.

5. The resulting matrix is simplified to reduce the numerical coefficients. There are two simplifications applied here:

[1] We measure how "complex" or "simple" a matrix is by a weighted sum of the number of arithmetic operations it contains and the number of bits needed to represent all of its coefficients. This is roughly proportional to the length of the textual representation of the matrix. A very crude metric, but it works because the difference between the most simple and the most complex matrices that appear on any given step are rather large.

a) For each off-diagonal block a constant transformation is chosen such that it would cancel at least one of the terms in the matrix, at the expense of potentially introducing other terms. If more terms are canceled than introduced, the transformation is accepted.

b) Each integral is rescaled by a numeric factor that minimizes the numeric content of the resulting matrix.

These transformations are heuristics and do not guarantee the smallest possible matrix in the end, but they do lead to practical improvements.

For multivariate differential equation systems the same algorithm is applied in a stepwise fashion, as described in Section 6.3.7.

8.2. Command-line usage summary

The syntax for FUCHSIA invocation on the command line is:

fuchsia `[options]` **command** *args* ...

Commands

The **`command`** argument gives the sub-command name. The main command is **`reduce`**. Available sub-commands are:

fuchsia `reduce` [`-x` *name*] [`-e` *name*] [`-m` *path*] [`-t` *path*] [`-i` *path*] *matrix*

Find an epsilon form of the given matrix. Internally this is a combination of **`reduce-diagonal-blocks`**, **`fuchsify-off-diagonal-blocks`**, **`factorize`**, and **`simplify`**.

fuchsia `reduce` [`-x` *name*] ... [`-e` *name*] [`-m` *path*] ... [`-t` *path*] [`-i` *path*] *matrix* ...

Find an epsilon form of a given multivariate differential equation system. A matching number of *matrix* arguments, `-x`, and `-m` flags is required.

The matrices are reduced one by one, and a single transformation is computed that simultaneously transforms all of them into an epsilon form. It may be best to list the simplest matrix first.

fuchsia `show` [`-x` *name*] [`-e` *name*] *matrix*

Show a human-readable description of a given matrix, listing residues and eigenvalues.

fuchsia `reduce-diagonal-blocks` [`-x` *name*] [`-e` *name*] [`-m` *path*] [`-t` *path*] [`-i` *path*] *matrix*

Transform the matrix into a block-triangular form and reduce the diagonal blocks into an epsilon form.

fuchsia `fuchsify-off-diagonal-blocks` [`-x` *name*] [`-m` *path*] [`-t` *path*] [`-i` *path*] *matrix*

Transform the off-diagonal blocks of a block-triangular matrix into a Fuchsian form, assuming the diagonal blocks are already in an epsilon form, thus making the whole matrix normalized Fuchsian.

fuchsia `factorize` [`-x` *name*] [`-e` *name*] [`-m` *path*] [`-t` *path*] [`-i` *path*] *matrix*

Find a transformation that will make a given normalized Fuchsian matrix proportional to the infinitesimal parameter.

fuchsia `fuchsify` [`-x` *name*] [`-m` *path*] [`-t` *path*] [`-i` *path*] *matrix*

Find a transformation that will transform a given matrix into a Fuchsian form. This is less efficient than block-based commands, because it effectively treats the whole matrix as one big block.

fuchsia `normalize` [`-x` *name*] [`-e` *name*] [`-m` *path*] [`-t` *path*] [`-i` *path*] *matrix*

Find a transformation that will transform a given Fuchsian matrix into a normalized form. This is less efficient than block-based commands, because it effectively treats the whole matrix as one big block.

fuchsia `sort` [`-m` *path*] [`-t` *path*] [`-i` *path*] *matrix*

Find a block-triangular form of the given matrix by shuffling.

fuchsia `transform` [`-x` *name*] [`-m` *path*] *matrix transform* ...

Transform a given matrix using a given transformation.

fuchsia `changevar` [`-x` *name*] [`-y` *name*] [`-m` *path*] *matrix expr*

Perform a change of variable from x to y, such that $x = \text{expr}(y)$.

fuchsia `suggest-changevar` [`-x` *name*] [`-y` *name*] *matrix*

Suggest a rational change of variable that will transform residue eigenvalues of the form $n/2 + k\varepsilon$ into $n + k\varepsilon$, thus making it possible to find an epsilon form of the matrix.

Note that some bad eigenvalues disappear when the matrix is fuchsified, so this routine is best used after **`fuchsify`**.

fuchsia `simplify` [`-x` *name*] ... [`-m` *path*] ... [`-t` *path*] [`-i` *path*] *matrix* ...

Try to find a transformation that makes a given matrix (or a set of matrices) simpler, for some definition of “simple”. This command tries to reduce the size of numerical coefficients in the matrix.

Options

`-x` *name* Use this name for the free variable (default: `x`).

`-0` *expr* Set this value for x during multivariate reduction (default: `0`).

`-y` *name* Use this name for the new free variable (default: `y`).

`-e` *name* Use this name for the infinitesimal parameter (default: `eps`).

`-m` *path* Save the resulting matrix into this file.

`-t` *path* Save the resulting transformation into this file.

`-i` *path* Save the inverse transformation into this file.

`-C` Force colored output even if the standard output is not a TTY.

`-P` Paranoid mode: spend more time checking internal invariants.

`-q` Print a more quiet log.

`-h` Show a help message.

`-V` Print version information.

Arguments

matrix Read the input matrix from this file.

transform Read the transformation matrix from this file.

expr Arbitrary expression.

8.3. Usage example

Let us obtain an ε-form for one of the previous examples: eq. (6.2.6). First we need to prepare the file with the matrix $\mathbb{M}$:

```
$ cat >example.m <<EOF
{{(2-2*ep)/(2*x), 0}, {(2*ep-2)/(2*q2*(x-1)*x), (1-2*ep)/(x-1)}}
EOF
```

Optionally we can review its content:

```
$ fuchsia show -q -e ep example.m
Matrix size: 2x2
Matrix shape:
  #.
  ##
Matrix complexity: 364
Matrix expansion:
  pole of power -1 at x=0
    complexity: 181
    e-value^1: 0
    e-value^1: 1-ep
    shape:
      #.
      #.
  pole of power -1 at x=1
    complexity: 173
    e-value^1: 0
    e-value^1: 1-2*ep
    shape:
      ..
      ##
  effective pole of power -1 at x=Infinity
    complexity: 119
    e-value^1: -1+ep
    e-value^1: -1+2*ep
    shape:
      #.
      .#
```

Next, let us reduce it to an ε-form:

```
$ fuchsia reduce -q -e ep example.m -t example.ep.t -m example.ep
[inf 0.0009s +0.0009s] Reducing a diff. eq. system in x
[inf 0.0022s +0.0013s **] Reducing 1x1 diagonal block at offset 0
[inf 0.0030s +0.0008s ****] Reduction between Infinity (eval=-1+ep) and 0
                            (eval=1-ep) can give complexity 54, distance 0
[inf 0.0032s +0.0002s ****] Reduction between Infinity (eval=-1+ep) and Infinity
                            (eval=-1+ep) can give complexity 118, distance 2
[inf 0.0035s +0.0002s ****] Reduction between 0 (eval=1-ep) and 0
```

```
                            (eval=1-ep) can give complexity 79, distance 2
[inf 0.0045s +0.0010s **] Reducing 1x1 diagonal block at offset 1
[inf 0.0055s +0.0010s ****] Reduction between Infinity (eval=-1+2*ep) and 1
                            (eval=1-2*ep) can give complexity 65, distance 0
[inf 0.0058s +0.0003s ****] Reduction between Infinity (eval=-1+2*ep) and Infinity
                            (eval=-1+2*ep) can give complexity 129, distance 2
[inf 0.0061s +0.0003s ****] Reduction between 1 (eval=1-2*ep) and 1
                            (eval=1-2*ep) can give complexity 90, distance 2
[inf 0.0082s +0.0022s **] Reducing 1x1 block at 1:0, at x=1, k=-2
[inf 0.0088s +0.0006s **] Use off-diagonal transformation, p=1, k=-1, D=
                          {{0,0}, {(2*ep*q2-q2)^(-1)*(-1+ep),0}}
[inf 0.0159s +0.0071s **] Use constant transformation of complexity 144:
                          {{1-2*ep,0}, {0,1-ep}}
[inf 0.0190s +0.0031s *] Shuffle into block-diagonal form with: {{1,0}, {0,1}}
[inf 0.0195s +0.0004s] Saving a diff. eq. matrix to example.ep
[inf 0.0198s +0.0003s] Saving the (unsimplified) transformation to example.ep.t
[inf 0.0199s +0.0002s] Saving the simplified transformation to example.ep.t
```

After 0.02 seconds FUCHSIA is done, and we can examine the transformation matrix (compare with eq. (6.2.6)):

```
$ cat example.ep.t
{{x*(-1+2*ep),0},
{(-1+ep)*q2^(-1),(-1+x)*(-1+ep)}}
```

Also the resulting differential equation matrix (compare with eq. (6.2.7)):

```
$ cat example.ep
{{-ep*x^(-1),0},
{q2^(-1)*ep*(-1+x)^(-1)-q2^(-1)*ep*x^(-1),-2*ep*(-1+x)^(-1)}}
```

9. Fully-inclusive phase-space integrals

A massless fully-inclusive phase-space Feynman integral in d dimensions with L loop momenta, P denominators, and n cut momenta has the form

$$I = \int \frac{\mathrm{d}^d l_1}{(2\pi)^d} \cdots \frac{\mathrm{d}^d l_L}{(2\pi)^d} \frac{1}{D_1^{\nu_1} \cdots D_P^{\nu_P}} \mathrm{dPS}_n(q), \tag{9.0.1}$$

where l_i are the loop momenta, $\mathrm{dPS}_n(q)$ is the phase-space volume element given by eq. (9.1.2), D_i are the denominators having the form

$$D_i = k_i^2 \pm i0, \tag{9.0.2}$$

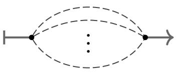
where k_i are some linear combinations of the loop and external momenta, and the sign of the $i0$ prescription is $+1$ for the denominators corresponding to lines in the Feynman diagram “to the left” of the cut, and -1 for the ones “to the right” (this just means that the right part of the diagram is complex-conjugated).

As explained in Section 2.4, to get at semi-inclusive phase-space integrals we need to know the fully-inclusive ones. Previously, master integrals for 4-particle cuts of 3-loop propagators were calculated in [139]; to get corrections of order α_s^3 we however need the cuts of 4-loop propagators. That calculation was started in [36] with 5-particle cuts, and completed in [35] with the rest of them. Because these are all single-scale integrals, the method of choice for them is the dimensional recurrence relations (discussed in Chapter 5). Let us explain how this calculation is performed.

9.1. Total inclusive phase space

Before proceeding further, let us work out the total fully inclusive phase-space volume of n particles in d dimensions,

$$\mathrm{PS}_n(q) \equiv$$



$$\equiv \int \mathrm{dPS}_n(q), \tag{9.1.1}$$

where the phase-space volume element is given by

$$\mathrm{dPS}_n(q) \equiv \left(\prod_{i=1}^{n} \frac{\mathrm{d}^d p_i}{(2\pi)^d} 2\pi \delta^+\!\left(p_i^2\right) \right) (2\pi)^d \, \delta^d\!\left(q - \sum_{j=1}^{n} p_j \right), \tag{9.1.2}$$

and δ^+ stands for positive-energy Dirac delta function,

$$\delta^+\!\left(p^2\right) \equiv \delta\!\left(p^2\right) \theta(p_0). \tag{9.1.3}$$

Note that if one would introduce the components if p_i explicitly as $p_i = (E_i, \vec{p}_i)$, then E_i can be immediately integrated out from this definition using the observation that

$$\delta^+\!\left(p_i^2\right) = \theta(E_i)\, \delta\!\left(E_i^2 - p_i^2\right) = \frac{1}{2E_i} \delta(E_i - |\vec{p}_i|), \tag{9.1.4}$$

which leads to an equivalent definition of the phase-space volume:

$$\mathrm{dPS}_n(q) \equiv \left(\prod_{i=1}^{n} \frac{\mathrm{d}^{d-1}\vec{p}_i}{(2\pi)^{d-1}} \frac{1}{2\,|\vec{p}_i|} \right) (2\pi)^d \, \delta^d\!\left(q - \sum_{j=1}^{n} p_j \right), \tag{9.1.5}$$

where p_i is assumed to have the components $(|\vec{p}_i|, \vec{p}_i)$.

9.1.1. Recurrence relation for the phase space

To calculate $\mathrm{PS}_n(q)$, first note that just from dimensional analysis and the requirement of Lorentz invariance, the full phase space must have the form of

$$\mathrm{PS}_n(q) = \Phi_n \times \left(q^2\right)^{n\left(\frac{d}{2}-1\right)-\frac{d}{2}}, \tag{9.1.6}$$

where Φ_n is a dimensionless constant,

$$\Phi_n = \mathrm{PS}_n(q)\big|_{q^2=1}. \tag{9.1.7}$$

Next, observe that we can factor out the last momentum from eq. (9.1.5), and the remaining part will be exactly $\mathrm{dPS}_{n-1}(q-p_n)$. This allows us to set up a recursion relation of the form

$$\begin{aligned} \mathrm{PS}_n(q) &= \int \frac{\mathrm{d}^{d-1}\vec{p}_n}{(2\pi)^{d-1}} \frac{1}{2\,|\vec{p}_n|} \mathrm{PS}_{n-1}(q-p_n) = \\ &= \int \frac{\mathrm{d}^{d-1}\vec{p}_n}{(2\pi)^{d-1}} \frac{1}{2\,|\vec{p}_n|} \Phi_{n-1} \left(q^2 - 2q\cdot p_n\right)^{(n-1)\left(\frac{d}{2}-1\right)-\frac{d}{2}}. \end{aligned} \tag{9.1.8}$$

The integrand here only depends on $q\cdot p_n$. Let us again exploit Lorentz invariance, and change into a frame of reference where $q = \big(q, \vec{0}\big)$. Then, $q\cdot p_n$ becomes just $q\,|\vec{p}_n|$. Because the integrand is invariant under rotations of $\vec{p}_n$, we can rewrite $\mathrm{d}^{d-1}\vec{p}_n$ in spherical coordinates, immediately integrating out the angular degrees of freedom,

$$\mathrm{d}^{d-1}\vec{p}_n = \Omega_{d-2}\,|\vec{p}_n|^{d-2}\,\mathrm{d}\,|\vec{p}_n|, \tag{9.1.9}$$

where Ω_k is the surface area of a unit k-sphere in $(k+1)$-dimensional Euclidean space, or the solid angle, as given by eq. (A.0.3). Then, the integral becomes

$$\mathrm{PS}_n(q) = \frac{\Phi_{n-1}\Omega_{d-2}}{2\,(2\pi)^{d-1}} \int \mathrm{d}\,|\vec{p}_n|\,|\vec{p}_n|^{d-3} \left(q^2 - 2q\,|\vec{p}_n|\right)^{(n-1)\left(\frac{d}{2}-1\right)-\frac{d}{2}}. \tag{9.1.10}$$

This is a hypergeometric integral, which can be taken routinely. With the integration range going from 0 to $q/2$, we have the resulting recurrence relation,

$$\Phi_n = \frac{2}{(16\pi)^{\frac{d-1}{2}}} \frac{\Gamma(d-2)\,\Gamma\!\left(\left(\frac{d}{2}-1\right)(n-2)\right)}{\Gamma\!\left(\frac{d-1}{2}\right)\Gamma\!\left(\left(\frac{d}{2}-1\right)n\right)} \Phi_{n-1}. \tag{9.1.11}$$

9.1.2. Two-particle phase space as the base case

For a solution to the recurrence of eq. (9.1.11) we will need a base case, Φ_2. Fortunately it is easy to calculate. In the reference frame where $q = \big(q, \vec{0}\big)$ we have

$$\mathrm{PS}_2(q) = \int \frac{\mathrm{d}^{d-1}\vec{p}_1}{(2\pi)^{d-1}} \frac{1}{2\,|\vec{p}_1|} \frac{\mathrm{d}^{d-1}\vec{p}_2}{(2\pi)^{d-1}} \frac{1}{2\,|\vec{p}_2|} (2\pi)^d \, \delta^{d-1}(\vec{p}_1 + \vec{p}_2)\, \delta(q - |\vec{p}_1| - |\vec{p}_2|). \tag{9.1.12}$$

Using the first δ to integrate out $\vec{p}_2$ sets it to $-\vec{p}_1$. Integration over the angular degrees of freedom of $\vec{p}_1$ just gives us $|\vec{p}_1|^{d-2}\,\Omega_{d-2}$. Then the second δ can be used to integrate out $|\vec{p}_1|$, setting it to $q/2$. In the end, we are left with just

$$\mathrm{PS}_2(q) = \frac{1}{8}\frac{\Omega_{d-2}}{(2\pi)^{d-2}}\left(\frac{q}{2}\right)^{d-4} = \frac{1}{4}\frac{1}{(16\pi)^{\frac{d-3}{2}}\,\Gamma\!\left(\frac{d-1}{2}\right)}\left(q^2\right)^{\frac{d}{2}-2}. \qquad (9.1.13)$$

9.1.3. Solving the recurrence

With the base case given by eq. (9.1.13), we can solve eq. (9.1.11) for any fixed $n > 2$. Moreover, observing several such solutions, it is easy to recognize the general expression for arbitrary n. The final result then reads

$$\mathrm{PS}_n(q) = \left(q^2\right)^{n\left(\frac{d}{2}-1\right)-\frac{d}{2}}\,\frac{2\pi}{(4\pi)^{\frac{d}{2}(n-1)}}\,\frac{\Gamma^n\!\left(\frac{d}{2}-1\right)}{\Gamma\!\left(\left(\frac{d}{2}-1\right)(n-1)\right)\Gamma\!\left(\left(\frac{d}{2}-1\right)n\right)}. \qquad (9.1.14)$$

At $d = 4$ this reproduces the formulas from the classical works of e.g. [140, 141].

9.2. Phase-space parameterization

Direct integration in momentum components quickly becomes unfeasible for integrals more complex than the full phase space. We can improve upon this by noting that an n-particle phase space has $n\,(d-1)-d$ degrees of freedom (components of the phase-space momenta p_i with one on-shell condition for each and one energy conservation condition), but Lorentz invariance requires the integrand of a phase-space integral to only depend on the scalar products of p_i,

$$s_{ij} \equiv \frac{\left(p_i + p_j\right)^2}{q^2}, \qquad (9.2.1)$$

of which there are $n\,(n-1)\,/2$ (remember that in the massless case $s_{ii} = 4p_i^2/q^2 = 0$). So, with high enough d there are extra degrees of freedom (of rotational nature) that the integrand does not depend on, and can be integrated out.

To that end, we can rewrite the massless phase-space volume element of eq. (9.1.2) in terms of $\mathrm{d}s_{ij}$ as

$$\mathrm{dPS}_n(q) = \frac{\left(q^2\right)^{n\left(\frac{d}{2}-1\right)-\frac{d}{2}}}{(2\pi)^{n(d-1)-d}}\left(\prod_{k=2}^{n}\frac{\Omega_{d-k}}{2}\right)\left(\prod_{i<j}\frac{\mathrm{d}s_{ij}}{2}\right)\Delta_n^{\frac{d-n-1}{2}}\,\theta(\Delta_n)\,\delta\!\left(1-\sum_{i<j}s_{ij}\right), \qquad (9.2.2)$$

where Ω_k is the surface area of a unit k-sphere given by eq. (A.0.3) (this corresponds to integrated out angular degrees of freedom), and Δ_n is the Gram determinant, which we define as

$$\Delta_n \equiv (-1)^{n+1}\det\left\{\frac{s_{ij}}{2}\right\}. \qquad (9.2.3)$$

9.2.1. Two-particle phase-space element

Explicitly, for the 2-particle phase-space element we have

$$\mathrm{dPS}_2(q) = \left(q^2\right)^{\frac{d}{2}-2} \frac{2^{4-2d}\pi^{\frac{3-d}{2}}}{\Gamma\left(\frac{d-1}{2}\right)} \mathrm{d}s_{12}\delta(1-s_{12}). \tag{9.2.4}$$

This of course is in agreement with the PS_2 value given in eq. (9.1.13). Note that the 2-particle phase-space is very restricted: s_{12} can only be 1, and

$$\mathrm{dPS}_2(q) = \mathrm{PS}_2(q)\,\delta(1-s_{12})\,\mathrm{d}s_{12}. \tag{9.2.5}$$

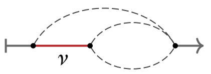
9.2.2. Three-particle phase-space element

For the 3-particle phase-space element we have

$$\mathrm{dPS}_3 = \left(q^2\right)^{d-3} \frac{2^{1-2d}\pi^{1-d}}{\Gamma(d-2)} \left(s_{12}s_{13}\left(1-s_{12}-s_{13}\right)\right)^{\frac{d}{2}-2} \theta(1-s_{12}-s_{13})\,\mathrm{d}s_{12}\mathrm{d}s_{13}. \tag{9.2.6}$$

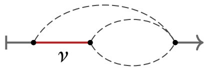
By direct integration this value matches the full PS_3 given by eq. (9.1.14), of course.

This parameterization allows us to take some 3-particle phase-space integrals in full generality (arbitrary d). For example, for an integral with one denominator with arbitrary power ν we have



$$\equiv \int \frac{\mathrm{dPS}_3(q)}{(p_1+p_2)^{2\nu}} = \frac{\mathrm{PS}_3(q)}{(q^2)^{\nu}} \frac{\Gamma\left(\frac{3d}{2}-3\right)}{\Gamma^3\left(\frac{d}{2}-1\right)} \int_0^1 \mathrm{d}s_{12} \int_0^{1-s_{12}} \mathrm{d}s_{13} \frac{\left(s_{12}s_{13}\left(1-s_{12}-s_{13}\right)\right)^{\frac{d}{2}-2}}{s_{12}^{\nu}}. \tag{9.2.7}$$

This works out to



$$= \frac{\Gamma\left(\frac{3d}{2}-3\right)}{\Gamma\left(\frac{3d}{2}-3-\nu\right)} \frac{\Gamma\left(\frac{d}{2}-1-\nu\right)}{\Gamma\left(\frac{d}{2}-1\right)} \frac{\mathrm{PS}_3(q)}{(q^2)^{\nu}}. \tag{9.2.8}$$

The same parameterization also helps in a more general case: if the loop part of the integral eq. (9.0.1) is known in terms of multiple polylogarithms, then we can parameterize dPS_3 as above, expand it in ε using

$$\left(s_{12}s_{13}\left(1-s_{12}-s_{13}\right)\right)^{\frac{d}{2}-2} = 1 + \left(-\ln s_{12} - \ln s_{13} - \ln(1-s_{12}-s_{13})\right) + \ldots, \tag{9.2.9}$$

multiply the loop part by this series, expand the products of multiple polylogarithms and logarithms using the shuffling relations (see Section 7.4), and integrate the results order by order (using the relations of Section 7.9). In this regard the 3-particle phase space is simple.

9.2.3. Four-particle phase-space element

The four-particle phase-space element is unfortunately more complex,

$$\mathrm{dPS}_4(q) = \left(q^2\right)^{\frac{3}{2}d-4} \frac{2^{4-4d}\pi^{\frac{1-3d}{2}}}{\Gamma(d-3)\,\Gamma\left(\frac{d-1}{2}\right)} \delta\left(1-\sum_{i<j} s_{ij}\right) \Delta^{\frac{d-5}{2}}\theta\left(\Delta_4\right) \prod_{i<j} \mathrm{d}s_{ij}, \tag{9.2.10}$$

where

$$\Delta_4 = \frac{1}{16}\left(2s_{12}s_{13}s_{24}s_{34} + 2s_{12}s_{14}s_{34}s_{23} + 2s_{13}s_{14}s_{24}s_{23} - s_{12}^2s_{34}^2 - s_{13}^2s_{24}^2 - s_{14}^2s_{23}^2\right). \tag{9.2.11}$$

The complexity lies in the $\theta(\Delta_4)$ factor: the integration volume is determined by an unfactorizable quadratic polynomial. So unlike the 3-particle case we cannot easily integrate dPS_4 after expanding it into a series in ε, because the limits will be given as square roots.

Now, it is possible to perform a change of variables that will simplify the integration volume. For example the *tripole parameterization* introduced in [139] is achieved with the following change of variables:

$$\begin{cases} s_{12} \to (1-t)(1-y)(1-z), \\ s_{13} \to \left(1-t-v-tv+tvz-2(1-2\xi)\sqrt{(1-t)(1-v)tvz}\right)y, \\ s_{14} \to (1-z)vy, \\ s_{23} \to (1-y)z, \\ s_{24} \to (1-y)(1-z)t, \\ s_{34} \to \left(t-tv+vz-tvz+2(1-2\xi)\sqrt{(1-t)(1-v)tvz}\right)y, \end{cases} \tag{9.2.12}$$

and with this substitution the integration region (in t, v, y, z, ξ) becomes a unit hypercube. The phase-space volume element under it becomes

$$\mathrm{dPS}_4(q) = \frac{\left(q^2\right)^{\frac{3d}{4}-4} 2^{-3d} \pi^{1-\frac{3d}{2}}}{\Gamma\left(\frac{d-1}{2}\right)\Gamma\left(\frac{d-2}{2}\right)\Gamma\left(\frac{d-3}{2}\right)} (tvz(1-t)(1-v))^{\frac{d}{2}-2} (y(1-y)(1-z))^{d-3} (\xi(1-\xi))^{\frac{d-5}{2}} \,\mathrm{d}t\,\mathrm{d}v\,\mathrm{d}y\,\mathrm{d}z\,\mathrm{d}\xi. \tag{9.2.13}$$

The simplicity of the integration volume here is gained at the expense of introducing roots into the integrand, so explicit analytic integration is still not possible in the general case. Note though that if the integrand does not depend on s_{13} and s_{34}, then no roots will be introduced. This allows for integration of a subset of 4-particle phase-space integrals. For example the total phase-space volume can be easily taken, and the result matches eq. (9.1.14).[1]

Phase spaces of five and higher number of particles are similarly complex, and do not allow for direct integration of arbitrary phase-space integrals.

9.3. Inclusive cuts of four-loop propagators

For the α_s^3 corrections we need all sets of cuts of 4-loop propagators: 2-particle cuts, 3-particle, 4-particle, and 5-particle cuts. Among these, the 2-particle cuts correspond to 3-loop form-factors, and have been completed in [70, 71, 72]. The 5-particle cuts are purely phase-space integrals (no loop part); we have presented their calculation in [36]. A subset of 3- and 4-particle cuts have been calculated in [142], but the majority where unknown until we have completed their calculation in [35].

9.3.1. Identifying the master integrals

To calculate the cut master integrals we first need to identify them. This turns out to be particularly easy if one starts with the master integrals for the propagators themselves (without cuts): there are 28 such master integrals, and their values are known from [143, 144]. See Table 9.3.1 for their list.

[1] Other examples of using using the tripole parameterization include a number of master integrals for 4-particle cuts of 3-loop propagators calculated in [139], and a few 4-particle cuts of 4-loop propagators that were taken in [142].

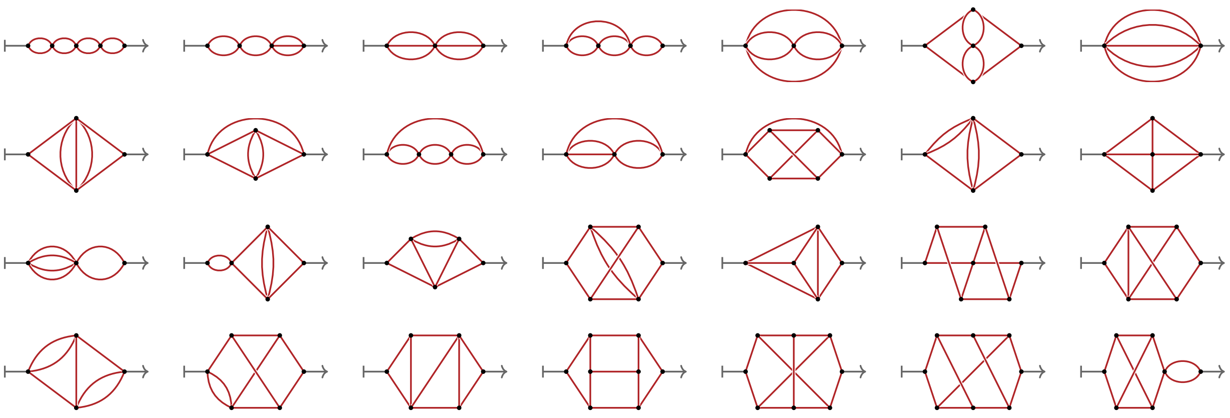
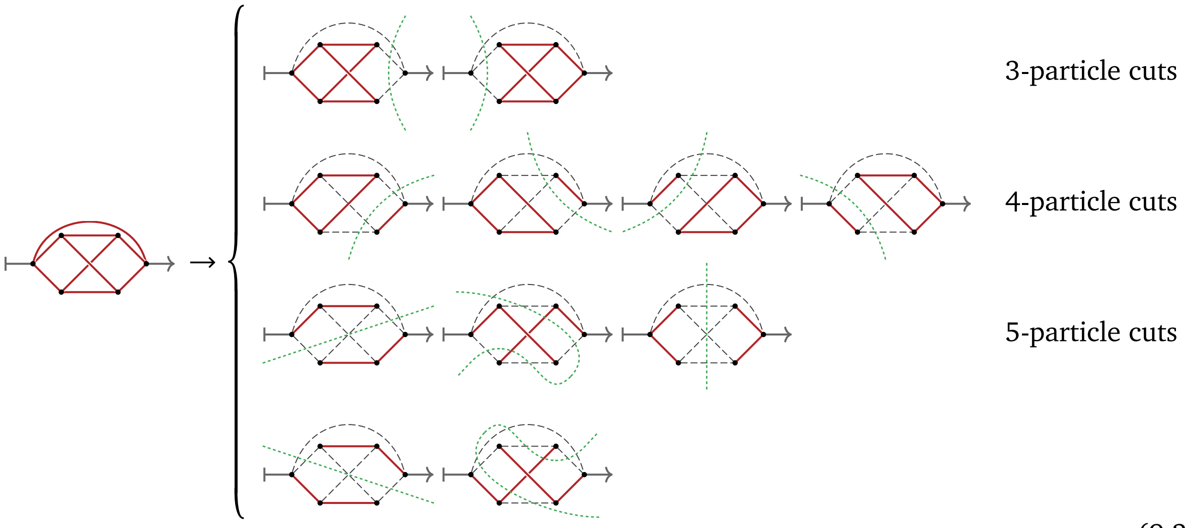


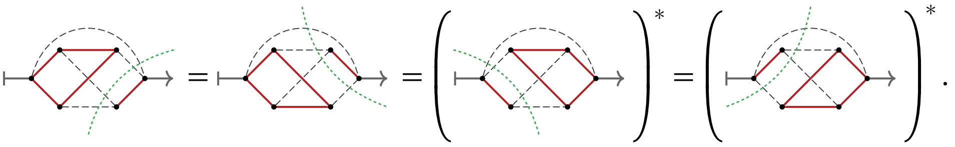
Table 9.3.1.: All 28 master integrals for 4-loop propagators.

To identify the cut master integrals, first, for each of the 28 propagator master integrals one needs to construct all of its possible cuts. For example:[2]



(9.3.1)

Second, all symmetric duplicate integrals should be removed from the resulting set of cuts. For example, all of the following four integrals are identical up to a complex conjugation:



(9.3.2)

And this is all, the resulting set will be the final answer. There are neither additional IBP relations between the remaining integrals, nor are there any additional master integrals (we have verified this explicitly using FIRE6 with LITERED).

In total the identified set of cut master integrals include 31 5-particle cuts (see Table 9.3.2), 35 4-particle cuts (Table 9.3.3), 27 3-particle cuts (Table 9.3.4), and 22 2-particle cuts (Table 9.3.5).

[2] The dotted cutting line is included in these diagrams for greater clarity. The best way to think of it is as a separator of the vertices into the "left" and the "right" parts; all edges between the left and the right sets of vertices are cut.

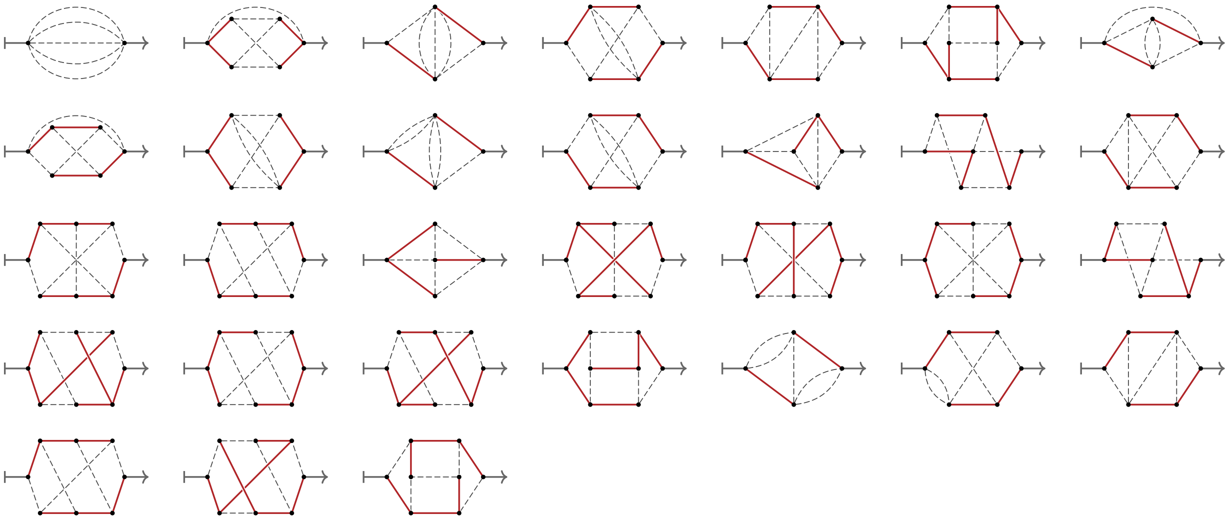


Table 9.3.2.: All 31 master integrals for 5-particle cuts of 4-loop propagators.

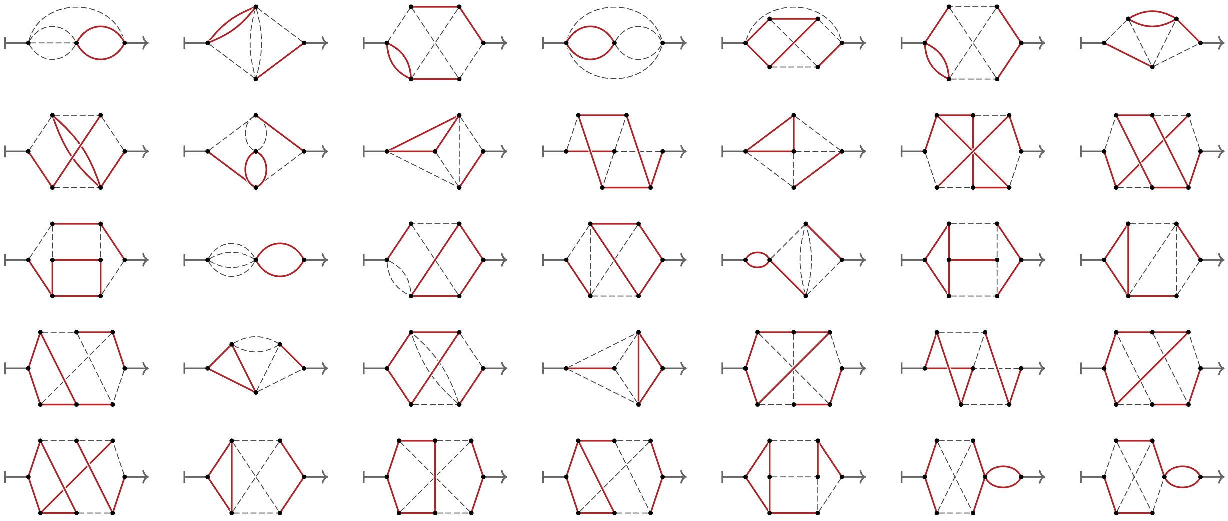


Table 9.3.3.: All 35 master integrals for 4-particle cuts of 4-loop propagators.

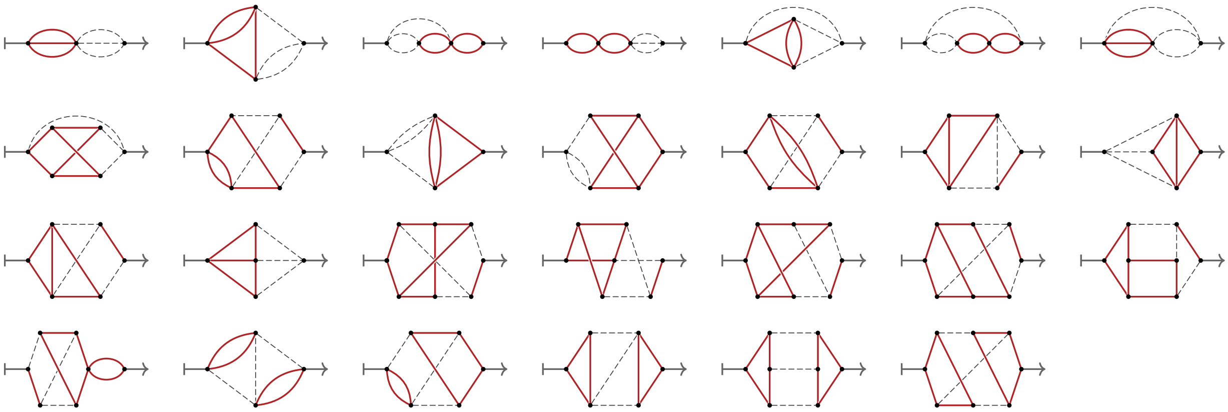


Table 9.3.4.: All 27 master integrals for 3-particle cuts of 4-loop propagators.

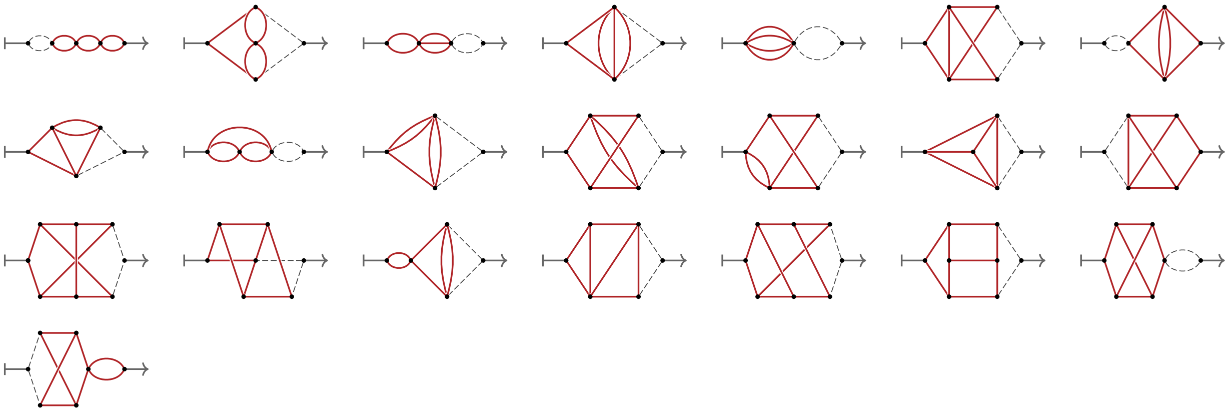
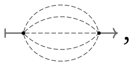


Table 9.3.5.: All 22 master integrals for 2-particle cuts of 4-loop propagators.

9.3.2. Calculating 5-particle cuts

The 5-particle cut master integrals (listed in Table 9.3.2) can be calculated via dimensional recurrence relations. We first derive them using LITERED and FIRE5; the DRR matrix is triangular with a factorizable diagonal, so we proceed to compute homogeneous and particular solutions as described in Chapter 5, but stopping short of using the dimensional recurrence and analyticity method. Instead, to determine $\omega_i(d)$ for 5-particle cut master integrals we consider the limit $d \to \infty$: we can determine the asymptotic behavior of each master $I_i(d)$ in this limit by using the phase-space parameterization of eq. (9.2.2) (in fact, we can easily do this for an arbitrary phase-space integral with no loop momenta), and then compare it to the asymptotic behavior of the homogeneous solution $H_i(d)$ of eq. (5.0.6); it turns out that $H_i(d)$ grows exponentially faster than $I_i(d)$ for all integrals except for the first master, , and through eq. (5.1.1) this means that $\omega_i(d)$ must be zero for those integrals.

Indeed consider an integral of form eq. (9.0.1) with $L = 0$ and $\mathrm{dPS}_n(q)$ parameterized via eq. (9.2.2). If Δ_n has a unique global maximum inside the integration region, $\Delta_n^{\max}$, then we can apply Laplace's method,

$$\int \mathrm{d}^n\mathbf{x}\, e^{Kf(\mathbf{x})} g(x) \overset{K\to\infty}{\approx} \left(\frac{2\pi}{K}\right)^{\frac{n}{2}} \frac{e^{Kf(\mathbf{x}_0)g(\mathbf{x}_0)}}{\sqrt{\det\left\{-\frac{\partial^2}{\partial x_i \partial x_j} f(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_0}\right\}}}, \qquad \text{where } \mathbf{x}_0 = \operatorname{argmax} f(\mathbf{x}), \tag{9.3.3}$$

and obtain the asymptotic behavior of the integral as

$$I(d) = \frac{\left(q^2\right)^{n\left(\frac{d}{2}-1\right)-\frac{d}{2}}}{(2\pi)^{n(d-1)-d}} \left(\prod_{k=2}^{n} \Omega_{d-k}\right) \left(\Delta_n^{\max}\right)^{\frac{d}{2}} \left(\frac{2\pi}{d}\right)^{\frac{1}{2}\left(\frac{n(n-1)}{2}-1\right)} \left(\mathscr{C} + \mathcal{O}\left(\frac{1}{d}\right)\right), \tag{9.3.4}$$

where $\mathscr{C}$ is some constant that does not depend on d.

The global maximum of Δ_n is reached when all s_{ij} $(i \neq j)$ are identical and equal to $n(n-1)$. Geometrically this corresponds to the vectors $\vec{p}_i$ pointing to the vertices of a regular n-hedron embedded into Euclidean space of $(n-1)$ dimensions. Its value is

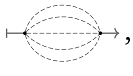
$$\Delta_n^{\max} = \frac{1}{n^n\,(n-1)^{n-1}}. \tag{9.3.5}$$

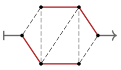
With this value the explicit asymptotic behavior of $I(d)$ becomes

$$I(d) \propto \left(q^2\right)^{2d-5} \left(\frac{e}{2\pi d}\right)^{2d} \left(\frac{1}{4^4 5^5}\right)^{\frac{d}{2}}, \tag{9.3.6}$$

while the asymptotic of $H_i(d)$ depends on the integral, and can be obtained directly from eq. (5.0.6) using the Stirling's formula

$$\Gamma(K) \overset{K\to\infty}{\approx} \sqrt{\frac{2\pi}{K}} \frac{K^K}{e^K}. \tag{9.3.7}$$

Going through the integrals one by one[3] we consistently find that $H(d)$ grows exponentially faster than $I(d)$ for all master integrals other than , meaning that $\omega_i(d)$ for them is zero, and $I(d)$ is then

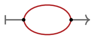
[3] A worked out example for  is presented in [145].

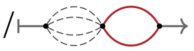
given by just the inhomogeneous solution of eq. (5.0.7), which we evaluate numerically via DREAM and restore the analytic expressions in terms of multiple zeta values via PSLQ.

As a cross-check we have computed all 5-particle cuts numerically at $d = 6$ and $d = 8$ (where they are finite). To this end we have extended the RAMBO phase-space parameterization [141] to work in arbitrary integer d (the original is for $d = 4$ only), obtaining a uniform mapping from a hypercube into p_i components, and then used the Vegas [146] implementation from CUBA [147] to integrate each integral over that hypercube numerically with 0.1% precision.[4] The obtained values match the analytic results within the integration error.

As another cross-check the same logic can be applied to 4-particle cuts of 3-loop propagators. We have done that, and the results match the values known from [139] (the details are given in [36]), with the advantage that our results easily extend to higher orders in the ε expansion.

9.3.3. Calculating 4-particle cuts

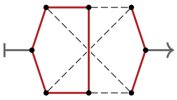
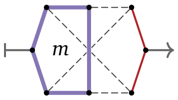
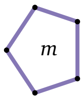
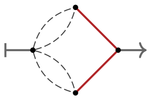
The master integrals for 4-particle cuts (listed in Table 9.3.3) differ from 5-particle cuts in that they contain one loop integration. For this reason we shall not pursue the limit $d \to \infty$, and shall turn to the full method of Chapter 5. A particularly useful property of 1-loop integrals is that they only have surface divergences, all contained in the prefactors of their Feynman parameterization (see eq. (3.2.1)). This means that we can suppress the UV divergences of a 1-loop integral by dividing it by  (or just by $\Gamma\big(2-\frac{d}{2}\big)$, but that is less convenient). The normalization of eq. (5.1.6) does this for us, and

$$J_i(d) \equiv I_i(d)\,/$$  (9.3.8)

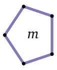
are all free of both IR and UV poles at $d \geq 6$. Once we have obtained DRR for them, we can construct $H_i^{-1}(d)$ and $R_i(d)$: both turn out to be finite in $d \in [6;8]$ too, which means that $\omega_i(d)$ does not have any poles in d. Then, as discussed in Chapter 5, we can proceed to derive bounds on $\omega_i(d)$ at $\operatorname{Im} d \to \infty$ and prove that $\omega_i(d)$ can only be a constant.

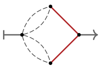
To calculate the constant ω_i, note that this is just one constant per integral, so it is sufficient for example to calculate a single term of the ε-expansion of each integral. Fortunately this can be done via the following observation: for a 4-particle cut integral with with n loop denominators, the superficial degree of divergence of the loop part becomes zero when $d = 2n$ (meaning that the integral begins to diverge logarithmically in the UV region), and importantly no IR divergences are present in this d as well. Changing d to $2n - 2\varepsilon$ regulates the UV divergence via a single ε pole, and being an UV pole, it does not depend on any masses in the diagram. Therefore, one can just as well insert some mass m into the loop without affecting the pole. Then, applying the large mass expansion [115] to the massive diagram factorizes it into a massive one-loop vacuum bubble (equal to the massive loop with external legs amputated) and a 4-particle phase-space integral (equal to the original integral with the loop shrinked into a vertex), while still not changing the pole.

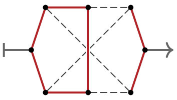
For example,

 $=$  $+\,\mathcal{O}(\varepsilon) =$  $\times$  $+\,\mathcal{O}\!\left(\varepsilon, \dfrac{q^2}{m^2}\right),\qquad \text{in } d = 10-2\varepsilon.$ (9.3.9)

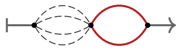
[4] The code for this is available at `https://github.com/magv/rambo`.

The vacuum bubble 

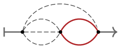
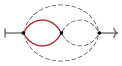
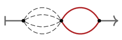
can be evaluated via eq. (3.1.4) (with $\nu = 5$), and  is just a 4-particle cut of a 3-loop propagator—it can be reduced via IBP and DRR to the master integrals known from [139] (also recalculated in the previous section). Substituting the values we have



$$= \frac{-5500}{3} \Big(\quad \Big)^{*} + \mathcal{O}\left(\varepsilon, \frac{q^2}{m^2}\right), \qquad \text{in } d = 10 - 2\varepsilon, \tag{9.3.10}$$



which we can compare to eq. (5.0.3) and figure out the corresponding ω_i.

The same procedure applies to all 4-particle cuts of 4-loop propagators. Interestingly, this way we have found that for all integrals except , , and , ω_i equals to 0.

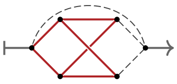
Once all ω_i are known, we can proceed to calculate the analytic expressions for the integrals using DREAM and PSLQ.

As a cross-check we have calculated these integrals numerically using the Feynman parameterization (eq. (3.2.1)) for the loop part of the integrals, and the tripole phase-space parameterization (eq. (9.2.12)) for the phase-space part. Because the integrals in question contain only one loop, their UV divergences manifest only in the prefactor of the Feynman parameterization, which can be factored out. The IR divergences also disappear at $d \geq 6$. For this reason this parameterization can be integrated directly via the standard Monte-Carlo numerical integration methods (we have used the Vegas algorithm from CUBA) in $d = 6 - 2\varepsilon$, and then lowered back to $d = 4 - 2\varepsilon$ via DRR—avoiding the need for methods like sector decomposition. In the end, we have verified the first 3 orders of ε expansion with 1% accuracy. Additionally, we have compared our results with the weight-6 series reported in [142] for 8 out of these integrals, and found them to match fully.

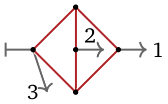
9.3.4. Calculating 3-particle cuts via direct phase-space integration

The master integrals for 3-particle cuts of 4-loop propagators (listed in Table 9.3.4) can be calculated in two ways: via direct phase-space integration (owning to the fact that 3-particle phase space is simple, see eq. (9.2.6)), and via DRR. We have used both, because direct integration only allows us to get results up to multiple zeta values of weight 7, while DRR allows to go to weight 12 easily, but is quite complicated and requires additional knowledge of the integral values to fix all periodic functions—this additional knowledge we obtain via direct integration

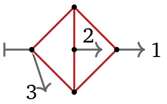
The master integrals for 1 → 3 amplitudes at 2 loops are known from [122, 116]; we have recalculated them in Section 6.4 as series in ε up to terms of transcendental weight 8. In principle we can combine them with the 3-particle phase-space parameterization of eq. (9.2.6) to calculate all the 3-particle cut integrals. For example to calculate



$$\equiv \int \quad \mathrm{dPS}_3 \tag{9.3.11}$$

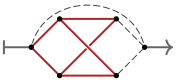


we can take

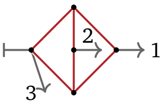


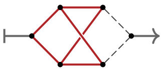
$$= \frac{\left(q^2\right)^{-2-2\varepsilon} N}{s_{12}^2} \left(-\frac{1}{\varepsilon^4} + \frac{2 \ln s_{12}}{\varepsilon^3} + \frac{5\zeta_2 - 2\ln^2 s_{12}}{\varepsilon^2} + \cdots \right), \tag{9.3.12}$$

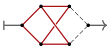
with N being the normalization factor from eq. (6.4.8), expand dPS_3 in ε using eq. (9.2.6), multiply the two, and take the integral in multiple polylogarithms. There is however a complication: integration order by order diverges,

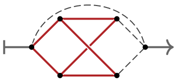

$$\overset{?}{=} \left(q^2\right)^{-1-4\varepsilon} N \frac{2^{-7+4\varepsilon}\pi^{-3+2\varepsilon}}{\Gamma(2-2\varepsilon)} \int_0^1 \left(-\frac{s_{12}^{-2}}{\varepsilon^4}+\dots\right) \mathrm{d}s_{12} = \frac{1}{0}\frac{1}{\varepsilon^4}+\dots. \qquad (9.3.13)$$

On the other hand we know that the integral is convergent: its can be calculated directly from eq. (9.3.11) without expanding in ε, if one notices that

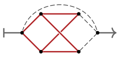


$$= s_{12}^{d-6}$$

$$(9.3.14)$$

where  is a single-scale integral known for arbitrary d from [148] (taken there using Feynman parameterization). This value is


$$\overset{?}{=} \left(q^2\right)^{-1-4\varepsilon} N \left(-\frac{3}{\varepsilon^5}+\frac{1}{\varepsilon^4}+\dots\right). \qquad (9.3.15)$$

The divergence we see in eq. (9.3.13) is just a sign that the integral is not IR-finite (after all, the $1/0$ term comes from $s_{12}=0$ limit of integration). Indeed, note that the series in eq. (9.3.15) starts with $1/\varepsilon^5$, while in eq. (9.3.13) with only $1/\varepsilon^4$: there is no way to obtain the $1/\varepsilon^5$ pole when integrating order by order.

The solution is not to abandon integration order by order, but rather to find a set of master integrals that are IR-finite, integrate them order by order, and then express  and the rest of integrals in this new basis.

For the IR-finite basis we can use the same set of integrals as in Table 9.3.4, but evaluated in $d=6-2\varepsilon$. This works because in higher d IR divergences disappear; this is easy to see from eq. (9.2.6): at high enough d the factor $(s_{12}s_{13}s_{23})^{d/2-2}$ will cancel any singular behavior at $s_{ij}\to 0$. We can easily move to and from this basis via dimensional recurrence relations; so the overall idea is to

1. Use DRR for the 1 → 3 integrals from Section 6.4 to obtain them as series at $d=6-2\varepsilon$.
2. Multiply by dPS_3 from eq. (9.2.6) with the same d.
3. Expand the product of series, integrate in multiple polylogarithms (as explained in Chapter 7). Now we have the values of 3-particle cuts of 4-loop propagators in $d=6-2\varepsilon$.
4. Use DRR to obtain the same integrals in $d=4-2\varepsilon$; these are our original master integrals.

Because we have calculated the 2-loop 1 → 3 integrals up to weight 8, one could expect that this procedure would yield the master integrals to the same weight. In practice there is a cancellation of the higher weight terms happening during DRR for some of the integrals, and the actual maximal weight of the results is 7.[5] This is enough for practical purposes, but can be improved upon through solving the DRR. Before we do that, a cross-check.

[5] Note that this is the reason we have recalculated the 2-loop 1 → 3 integrals in the first place: the answers in [122, 116] are only given up to weight 4, which is not enough for our purpose.

9.3.5. Cutkosky relations

As a test of consistency between the values of the 4-loop propagator masters and the cuts we have calculated, we can take a look at the optical theorem (or rather Cutkosky rules [149, 150]) that relate the cuts to the discontinuity of the propagators.

The general optical theorem comes from the requirement of unitarity of the scattering matrix $\mathbb{S}$,

$$\mathbb{S}^{\dagger}\mathbb{S} = \mathbb{1}. \tag{9.3.16}$$

Introducing the transition matrix $\mathbb{T}$ as

$$\mathbb{S} = \mathbb{1} + i\mathbb{T}, \tag{9.3.17}$$

it follows that

$$i\mathbb{T} + (i\mathbb{T})^{\dagger} = -(i\mathbb{T})^{\dagger}\, i\mathbb{T}. \tag{9.3.18}$$

For a decay of a single particle with momentum q, rewriting this relation in terms of the transition amplitudes gives us

$$2\mathrm{Re}\, \langle q|\, i\mathbb{T}\, |q\rangle = -\langle q|\,(i\mathbb{T})^{\dagger}\left(\sum_x |x\rangle\,\langle x|\right) i\mathbb{T}\,|q\rangle = -\sum_n \int \mathrm{dPS}_n\, |\langle p_1,\ldots,p_n|\, i\mathbb{T}\,|q\rangle|^2. \tag{9.3.19}$$

This is the optical theorem. Cutkosky rules provide a stronger form of this relation that holds not only for the whole transition amplitude $\langle q|\, i\mathbb{T}\,|q\rangle$, but for each individual Feynman diagram F that comprise it too,

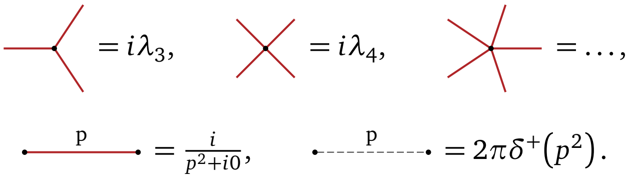
$$F + F^{*} = \sum_i \mathrm{Cut}_i F, \tag{9.3.20}$$

where the sum goes over all possible cuts of the diagram, each cut being a partition into two sides, with the right-hand side complex-conjugated, and the propagators between sides set on shell—exactly as we have defined the cut integrals.

To write down these relations for our master integrals in this convenient form, we shall augment our integrals with Feynman rules stemming from a simple scalar field theory with the Lagrangian of the form

$$\mathscr{L} = \frac{1}{2}(\partial\phi)^2 + \frac{\lambda_3}{3!}\phi^3 + \frac{\lambda_4}{4!}\phi^4 + \ldots. \tag{9.3.21}$$

The momentum-space Feynman rules corresponding to this Lagrangian are:



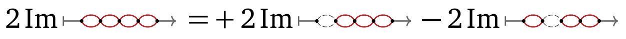
$$= i\lambda_3, \qquad = i\lambda_4, \qquad = \ldots,$$

$$= \frac{i}{p^2+i0}, \qquad = 2\pi\delta^{+}\big(p^2\big). \tag{9.3.22}$$

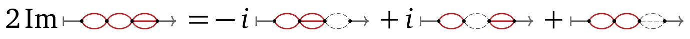
An additional prescription for cut integrals is this: every vertex and propagator to the right side of the cut needs to be complex-conjugated.

Note that the values of λ_i are not important for us here, because a cut of a diagram will have the same overall λ factor as the initial diagram, which will thus factor out from eq. (9.3.20).

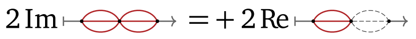
With this in mind, writing down eq. (9.3.20) for each Feynman diagram corresponding to a 4-loop propagator master integral from Table 9.3.1, and mapping the cuts onto our master integrals via symmetries, we obtain the following relations:



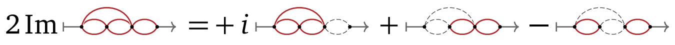
(9.3.23)



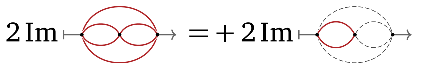
(9.3.24)



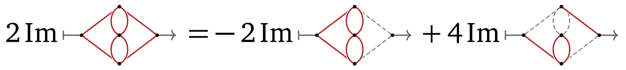
(9.3.25)



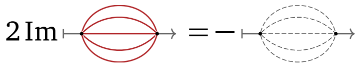
(9.3.26)



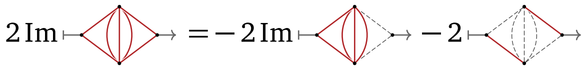
(9.3.27)



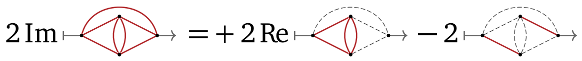
(9.3.28)



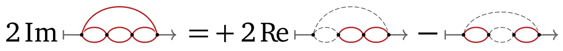
(9.3.29)



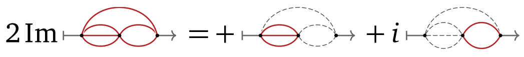
(9.3.30)



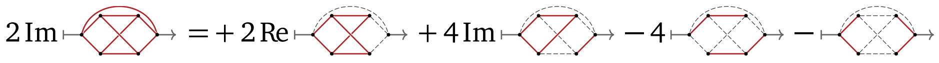
(9.3.31)



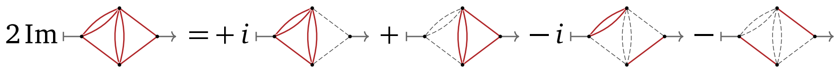
(9.3.32)



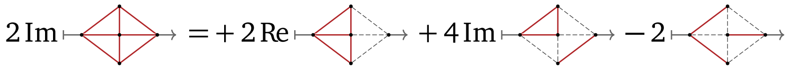
(9.3.33)



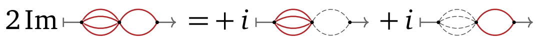
(9.3.34)



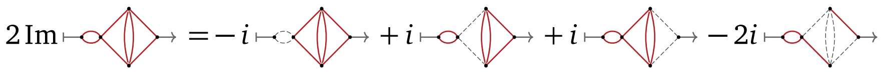
(9.3.35)



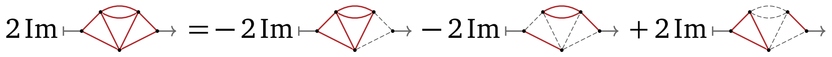
(9.3.36)



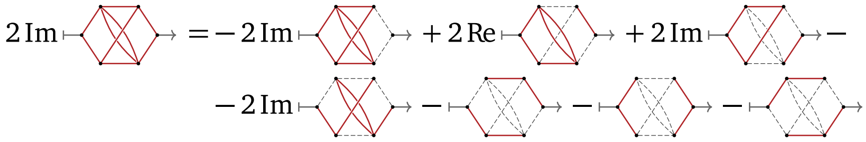
(9.3.37)



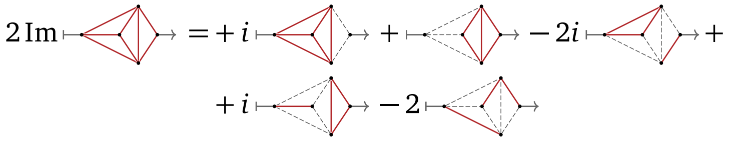
(9.3.38)



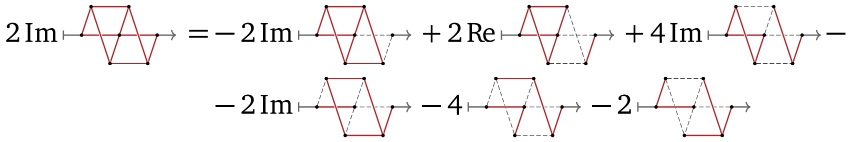
(9.3.39)



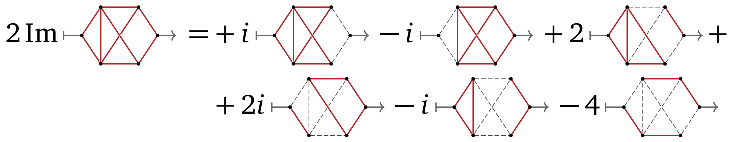
(9.3.40)



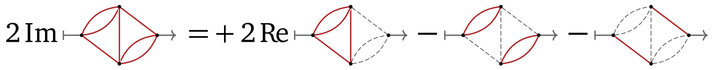
(9.3.41)



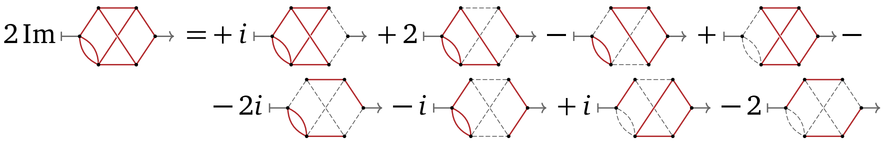
(9.3.42)



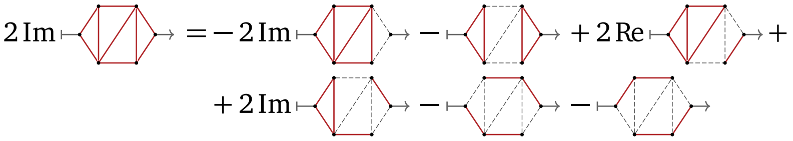
(9.3.43)



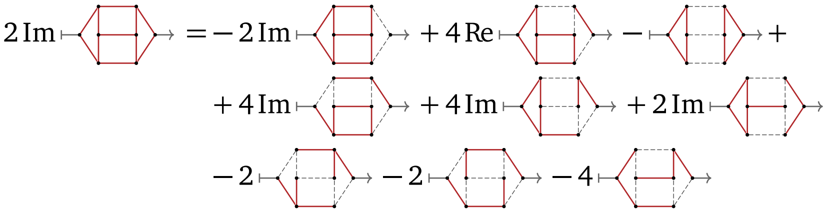
(9.3.44)



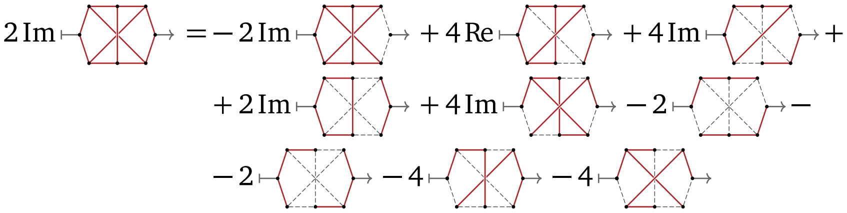
(9.3.45)



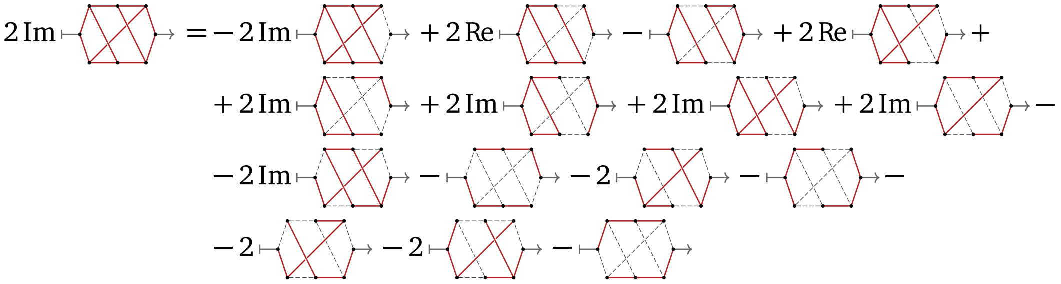
(9.3.46)



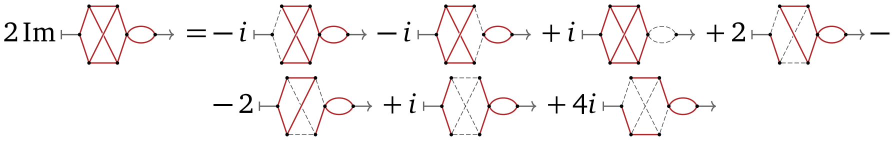
(9.3.47)



(9.3.48)



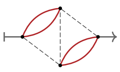
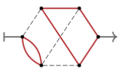
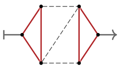
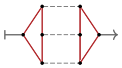
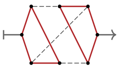
(9.3.49)

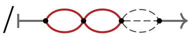


(9.3.50)

Inserting the values of our cut integrals into these relations, we find that they all hold precisely. This concludes our cross-check.

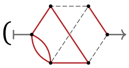
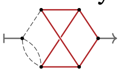
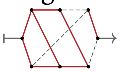
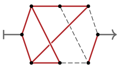
9.3.6. Calculating 3-particle cuts via dimensional recurrence relations

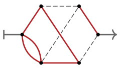
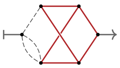
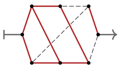
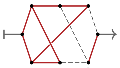
There are two kinds of 3-particle cut integrals: the ones with two loops on the same side of the cut ("VVRR") and the ones with a single loop on each side ("VRRV"). The latter include , , , , and . Due to the difference in the cut structure, these integrals do not mix with the other group in the differential equations or the DRR, and can be considered separately. In fact, VRRV integrals behave more akin to 1-loop integrals than to the 2-loop ones: after we have normalized the 3-particle cuts via

$$J_i \equiv I_i / $$  , (9.3.51)

VRRV integrals become finite in $d \geq 6$, and DRR for them can be solved exactly the way DRR were solved for 4-particle cuts.

Solving DRR for VVRR integrals is more complicated because the normalization of eq. (9.3.51) no longer suppresses all the UV divergences like eq. (9.3.8) did for 4-particle cuts—it only cancels the surface divergences. This means that $J_i(d)$ will have poles in d, and as discussed in Chapter 5 the ansatz for the periodic function $\omega_i(d)$ in eq. (5.1.13) will then contain multiple $a_{i,n}$ that cannot be determined from just expanding eq. (5.1.1) around the poles (because no expansion of $J_i(d)$ will be available). For this reason additional constraints are needed—the more poles of $J_i(d)$, the more constraints.

Luckily, we have two sources of such constraints: the values of the integrals calculated by direct integration in Section 9.3.4, and the Cutkosky relations of eq. (9.3.23) through eq. (9.3.50). Cutkosky relations are quite powerful in this regard: often they contain only a single VVRR master integral—only eq. (9.3.45) and eq. (9.3.49) contain two ( and ,  and  respectively). In these cases the master integral entering the equation can be completely determined by it, if we know the values of propagators and other cut structures. To this end we can use the Cutkosky relations to numerically evaluate the integrals at multiple different d, and determine $a_{i,k}$ from that. We do this instead of just evaluating the equations at $d = 4-2\varepsilon$ to get the final result for two reasons: first we want to check that the DRR solutions we have are consistent with the equations, and second, in the results we want to provide not just the values of the integrals in $d = 4-2\varepsilon$, but also the full solutions to DRR (in terms of SUMMERTIME files described in Appendix C), so that anyone could calculate these integrals in arbitrary d with any precision.

In the case of , , , and  we complete the determination of all $a_{i,k}$ by comparing the DRR solutions with the first few terms of the ε-expansion of the integrals obtained in Section 9.3.4.

10. Semi-inclusive phase-space integrals

A massless semi-inclusive phase-space integral in d dimensions with L loop momenta, P denominators, and n cut momenta has the form

$$I = \int \frac{\mathrm{d}^d l_1}{(2\pi)^d} \cdots \frac{\mathrm{d}^d l_L}{(2\pi)^d} \frac{1}{D_1^{\nu_1} \cdots D_P^{\nu_P}} \underbrace{\mathrm{dPS}_n(q)\, \delta\Big(x - 2\frac{q\cdot p_n}{q^2}\Big)}_{\equiv \mathrm{dPS}_n(q,x)}, \tag{10.0.1}$$

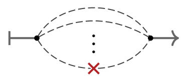
where l_i are the loop momenta, $\mathrm{dPS}_n(q)$ is the phase-space volume element given by eq. (9.1.2), and D_i are the denominators—everything exactly the same as for the fully inclusive integrals in eq. (9.0.1), except for δ-function that makes this integral differential (i.e. not inclusive) in x, hence *semi-inclusive*.

Integrals of this form appear as products of Feynman diagrams integrated over the phase space. We are interested in integrals appearing up to α_s^3, which corresponds to $L + n \leq 5$, or—if semi-inclusive integrals are seen as cuts of propagators—to semi-inclusive cuts of propagators up to 4 loops.

Before we proceed to calculating master integrals for these cuts, let us start by computing several simpler integrals by hand. This will help us to cross-check further calculations.

10.1. Semi-inclusive phase space

The simplest of the semi-inclusive integrals is the no-propagator integral, which is just the semi-inclusive phase space,

$\mathrm{PS}_n(q,x) =$  $= \displaystyle\int \mathrm{dPS}_n(q)\, \delta\Big(x - 2\frac{q\cdot p_n}{q^2}\Big),$ (10.1.1)

where $\mathrm{dPS}_n(q)$ is that of eq. (9.1.2).

This integral is highly symmetric, which will allow us to calculate it. Note that the delta function here does not depend on $p_{i<n}$, so we can factorize the equation as

$$\mathrm{PS}_n(q,x) = \int \mathrm{dPS}_{n-1}(q - p_n) \frac{\mathrm{d}^{d-1}\vec{p}_n}{(2\pi)^{d-1}} \frac{1}{2\,|\vec{p}_n|} \delta\Big(x - 2\frac{q\cdot p_n}{q^2}\Big). \tag{10.1.2}$$

The $(n-1)$-particle phase-space here can be integrated over right away, using the total phase space volume given by eq. (9.1.14). Next, we can move into the frame of reference where $q = \big(q, \vec{0}\big)$, so that $q\cdot p_n$ becomes $q\,|\vec{p}_n|$, and the integrand is manifestly invariant to the rotations of $\vec{p}_n$,

$$\mathrm{PS}_n(q,x) = \Phi_{n-1} \int \frac{\mathrm{d}^{d-1}\vec{p}_n}{(2\pi)^{d-1}} \frac{1}{2\,|\vec{p}_n|} \big(q^2 - 2q\,|\vec{p}_n|\big)^{(n-1)(\frac{d}{2}-1)-\frac{d}{2}} \delta\Big(x - 2\frac{|\vec{p}_n|}{q}\Big). \tag{10.1.3}$$

This allows us to integrate out the angular degrees of freedom of $\mathrm{d}^{d-1}\vec{p}_n$, which results in the surface area of a $(d-2)$-sphere, or $|\vec{p}_n|^{d-2}\,\Omega_{d-2}$ via eq. (A.0.3). Then, we are only left with the radial part of $\mathrm{d}^{d-1}\vec{p}_n$,

$$\mathrm{PS}_n(q,x) = \Phi_{n-1} \int \frac{\mathrm{d}\,|\vec{p}_n|}{(2\pi)^{d-1}} \frac{|\vec{p}_n|^{d-3}\,\Omega_{d-2}}{2} \left(q^2 - 2q\,|\vec{p}_n|\right)^{(n-1)\left(\frac{d}{2}-1\right)-\frac{d}{2}} \underbrace{\delta\!\left(x - 2\frac{|\vec{p}_n|}{q}\right)}_{=\delta\left(|\vec{p}_n|-\frac{1}{2}qx\right)\frac{q}{2}}. \tag{10.1.4}$$

Finally, integrating over $|\vec{p}_n|$ resolves the δ function,

$$\mathrm{PS}_n(q,x) = \Phi_{n-1} \frac{\left(\frac{1}{2}qx\right)^{d-3}\Omega_{d-2}}{2\,(2\pi)^{d-1}} \left(q^2\,(1-x)\right)^{(n-1)\left(\frac{d}{2}-1\right)-\frac{d}{2}} \frac{q}{2}. \tag{10.1.5}$$

To make the relation between the semi-inclusive phase-space $\mathrm{PS}_n(q,x)$ and the fully inclusive phase-space $\mathrm{PS}_n(q)$ obvious, we can factor out the latter, giving us the final answer,

$$\mathrm{PS}_n(q,x) = \mathrm{PS}_n(q)\, \frac{\Gamma\!\left(\left(\frac{d}{2}-1\right)n\right)}{\Gamma(d-2)\,\Gamma\!\left(\left(\frac{d}{2}-1\right)(n-2)\right)} x^{d-3}\,(1-x)^{n\left(\frac{d}{2}-1\right)-d+1}. \tag{10.1.6}$$

A good consistency check for this result is that an explicit integration over all x, $\int_0^1 \mathrm{d}x\,\mathrm{PS}_n(q,x)$, gives precisely the expected $\mathrm{PS}_n(q)$. Note that this integration is only possible if the powers of both x and $(1-x)$ factors are greater than -1,

$$\begin{cases} d-3 > -1 \\ \left(\frac{d}{2}-1\right)n - d + 1 > -1 \end{cases} \iff \begin{cases} d > 2 \\ n > 2 \end{cases}, \tag{10.1.7}$$

otherwise the integration would diverge at the boundaries.

10.1.1. Two-particle semi-inclusive phase space

The restriction on n in eq. (10.1.7) is important to us, because it excludes the physically relevant case of $n=2$. This is easy to understand though, because a massless two-particle phase-space is special in that it is kinematically restricted. If a particle of momentum q decays into two massless particles of momenta p_1 and p_2, then

$$q = p_1 + p_2 \qquad \Rightarrow \qquad q^2 = \underbrace{p_1^2}_{=0} + 2p_1{\cdot}p_2 + \underbrace{p_2^2}_{=0}. \tag{10.1.8}$$

Looking at the definition of the variable x in eq. (10.0.1), we can conclude that only one value of x is possible in this case,

$$x = 1. \tag{10.1.9}$$

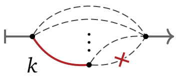
Thus the two-particle semi-inclusive phase space has distributional nature,

$$\mathrm{PS}_2(q,x) = \mathrm{PS}_2(q)\,\delta(x-1). \tag{10.1.10}$$

10.2. One-propagator integral

Another highly symmetric integral that we can calculate is a one-propagator integral with arbitrary propagator power k of the following form:

$$I_1(n;q,x;k) \equiv$$



$$\equiv \int \mathrm{d}\mathrm{PS}_n(q)\,\delta\!\left(x - 2\frac{q{\cdot}p_n}{q^2}\right) \frac{1}{(p_n + p_{n-1})^{2k}}. \tag{10.2.1}$$

The first thing to note is that I_1 is invariant under transformations that leave q, p_n and p_{n-1} unchanged. Because of this, we can factor out the phase space of the first $n-2$ particles, and integrate it out using eq. (9.1.14):

$$I_1 = \frac{\Phi_{n-2}}{(2\pi)^{2d-2}} \int \left((q-p_n-p_{n-1})^2\right)^{(n-2)(\frac{d}{2}-1)-\frac{d}{2}} \frac{\mathrm{d}^{d-1}\vec{p}_{n-1}}{2\,|\vec{p}_{n-1}|} \frac{\mathrm{d}^{d-1}\vec{p}_n}{2\,|\vec{p}_n|} \delta\!\left(x-2\frac{q\cdot p_n}{q^2}\right) \frac{1}{(p_n+p_{n-1})^{2k}}. \tag{10.2.2}$$

In this form the integrand depends on only three non-trivial kinematic invariants: $q\cdot p_n$, $q\cdot p_{n-1}$, and $p_n\cdot p_{n-1}$. The first two are greatly simplified in the rest frame, where $q=\left(q,\vec{0}\right)$, and thus $q\cdot p_i = q\,|\vec{p}_i|$. The third one only depends on the magnitudes of $\vec{p}_{n-1}$ and $\vec{p}_n$, and the angle between them, which we'll call θ. To make use of this observation, we can rewrite the volume elements $\mathrm{d}^{d-1}\vec{p}_{n-1}$ and $\mathrm{d}^{d-1}\vec{p}_n$ using spherical coordinates, constructed in such a way that $\vec{p}_{n-1}$ constitutes the north pole direction for the p_n coordinate system. This will directly expose the integration over the angle θ and the radial components, $|\vec{p}_{n-1}|$ and $|\vec{p}_n|$, while the rest will factor out:

$$\mathrm{d}^{d-1}\vec{p}_{n-1} = |\vec{p}_{n-1}|^{d-2}\,\mathrm{d}\,|\vec{p}_{n-1}|\;\Omega_{d-2}, \tag{10.2.3}$$

$$\mathrm{d}^{d-1}\vec{p}_n = |\vec{p}_n|^{d-2}\,\mathrm{d}\,|\vec{p}_n|\;\sin^{d-3}(\theta)\,\mathrm{d}\theta\,\Omega_{d-3}, \tag{10.2.4}$$

where Ω_k is the total solid angle defined by eq. (A.0.3), and $\sin^{d-3}(\theta)$ comes from the volume element in the spherical coordinates given by eq. (A.0.2).

Next, introducing scaleless quantities y and z such that $|\vec{p}_{n-1}| = qy/2$ and $\cos\theta = 1-2z$, and then integrating out $|\vec{p}_n|$ via $\delta\!\left(x-2\frac{|\vec{p}_n|}{q}\right) = \delta\!\left(|\vec{p}_n| - \frac{1}{2}qx\right)\frac{q}{2}$, we can rewrite parts of I_1 as

$$\frac{\mathrm{d}^{d-1}\vec{p}_{n-1}}{2\,|\vec{p}_{n-1}|} = \frac{1}{2}\left(\frac{qy}{2}\right)^{d-3}\frac{q\,\mathrm{d}y}{2}\Omega_{d-2}, \tag{10.2.5}$$

$$\frac{\mathrm{d}^{d-1}\vec{p}_n}{2\,|\vec{p}_n|} = \frac{1}{2}\left(\frac{qx}{2}\right)^{d-3}\left(1-(1-2z)^2\right)^{\frac{d}{2}-2} 2\,\mathrm{d}z\,\Omega_{d-3}, \tag{10.2.6}$$

after which the whole expression becomes

$$\begin{aligned} I_1 =&\frac{\Phi_{n-2}\Omega_{d-2}\Omega_{d-3}}{2^{d+1}\,(2\pi)^{2d-2}}\left(q^2\right)^{n\left(\frac{d}{2}-1\right)-\frac{d}{2}-k} x^{d-3-k}\times \\ &\times\int_0^1 \mathrm{d}z \int_0^{\frac{1-x}{1-xz}} \mathrm{d}y\,(1-x-y+xyz)^{(n-2)\left(\frac{d}{2}-1\right)-\frac{d}{2}}\,(1-z)^{\frac{d}{2}-2}\,z^{\frac{d}{2}-2-k}\,y^{d-3-k}, \end{aligned} \tag{10.2.7}$$

where the integration limits for y were chosen as to satisfy the requirements of $\theta(E_i)$—we have so far omitted these θ-function factors from the phase-space elements for brevity, but they still are implied.

The resulting integral can be taken routinely, and assuming $d>2$ and $d>2+2k$, the result reads

$$\begin{aligned} I_1(n;q,x;k) =&\frac{\Phi_{n-2}\Omega_{d-2}\Omega_{d-3}}{2^{d+1}\,(2\pi)^{2d-2}}\left(q^2\right)^{n\left(\frac{d}{2}-1\right)-\frac{d}{2}-k} x^{d-3-k}\times \\ &\times(1-x)^{(n-2)\left(\frac{d}{2}-1\right)-1}\,\frac{\Gamma\!\left(\frac{d}{2}-1\right)\Gamma\!\left(\frac{d}{2}-1-k\right)\Gamma\!\left(\left(\frac{d}{2}-1\right)(n-3)\right)}{\Gamma\!\left(\left(\frac{d}{2}-1\right)(n-1)-k\right)}. \end{aligned} \tag{10.2.8}$$

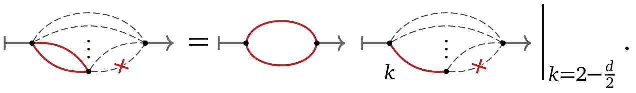
As a consistency check, setting k to 0 in this result and comparing it to eq. (10.1.6), makes it possible to check that $I_1(x;q,x;0)$ is equal to $\mathrm{PS}_n(q,x)$. A different way of writing I_1 to help making this explicit is

$$I_1(n;q,x;k) = \frac{\mathrm{PS}_n(q,x)}{(q^2x)^k}\frac{\Gamma\!\left(\frac{d}{2}-1-k\right)}{\Gamma\!\left(\frac{d}{2}-1\right)}\frac{\Gamma\!\left(\left(\frac{d}{2}-1\right)(n-1)\right)}{\Gamma\!\left(\left(\frac{d}{2}-1\right)(n-1)-k\right)}. \tag{10.2.9}$$

10.2.1. Propagator diagram insertions

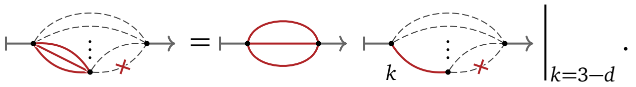
Because we have kept the power of the propagator in eq. (10.2.1) arbitrary, we can now calculate any integral of the same form, with the propagator line replaced by a (massless) propagator-type diagram. This is because such diagrams are necessarily proportional to the momenta squared flowing through them to a fractional power—determined just by dimensionality—and so they all fit the same general form of eq. (10.2.1).

For example, a one-loop bubble integral has the simple form of eq. (3.2.10): it is proportional to $\left(q^2\right)^{\frac{d}{2}-2}$, so we can immediately calculate an integral with a one-loop bubble insertion in place of the propagator as



(10.2.10)

Similarly,



(10.2.11)

The list goes on.

10.3. Semi-inclusive cuts of four-loop propagators

10.3.1. Identifying integral families

First let us start by identifying the integral families covering the integrals in question. This is slightly more involved than with the fully inclusive cuts for two reasons:

- there are more integral families because the x variable can be assigned to any of the cut lines, each assignment giving a different integral;
- because we want to treat the $\delta\left(x-2q{\cdot}p_n/q^2\right)$ factor as a cut denominator (for IBP purposes), the denominator sets of many diagrams turn out to be linearly dependent, so the integrands need to be decomposed into partial fractions (as described in Section 3.5.3).

The procedure we have followed to identify the integral families is as follows:[1]

1. Use QGRAF [101] to generate all Feynman diagrams of decays of off-shell photons, Z-bosons, and Higgs bosons into 2, 3, 4, and 5 particles up to 3 loops. Note that for Higgs we assume an effective theory where the heavy-quark loops are shrinked into a point, giving an effective coupling of Higgs to multiple gluons.
2. Insert Feynman rules into each diagram to get an amplitude. Note that there is no need to expand the tensor sums just yet.
3. Construct all possible products of two amplitudes that are proportional to α_s^3.
4. Perform partial fraction decomposition for each product, transforming each into one or several terms.
5. For each term of the partial fraction determine the set of denominators.
6. Pass the list of denominator sets to FEYNSON (see Chapter 4), and replace the loop momenta in each term as it suggests. This makes symmetries explicit.

[1] The code to perform each of these steps (and more) is available in ALIBRARY.

7. For each term compute its new denominator set, dropping any set that is a subset of another. The resulting list of denominator sets are the integral families we were after.

In total we have found 256 integral families; they are summarized in Appendix D: 18 with two cuts (Table D.0.1), 34 with three cuts (Table D.0.2), 96 with four cuts (Table D.0.3), and 108 with five cuts (Table D.0.4). Each 2-particle cut family has 11 denominators and an implied $\delta(x-1)$; the rest have 12 denominators, including $\delta\big(x-2q{\cdot}p_n/q^2\big)$.

10.3.2. Identifying master integrals

The next step is to figure out the set of master integrals. We do this by first converting every term of the partial fractions to an explicit $I_{\nu_1,\ldots,\nu_n}$ notation, and then passing the list of these integrals to FIRE6 (used in combination with LITERED)—separately for each integral family, and for the master integral list selection only. Alternatively we could have provided any sufficiently large list of integrals to FIRE6; the only advantage of our way is that we guarantee that no master integral relevant to the decay cross-section computations was missed.

FIRE6 can operate in two modes: the classical symbolic mode, and the modular arithmetic mode, where all of the variables (d and x) are replaced with fixed numerical values (e.g. 101921 and 961748927), and the reduction is performed modulo a large prime number (e.g. 18446744073709551557): the latter is much faster (taking up to several hours per integral family, where the symbolic is taking up to several weeks), and generally produces the same set of master integrals as the symbolic version. We have used three different combinations of d, x, and the modulus values: all report the same set. In the end up to 70 master integrals are identified per integral family. After using FEYNSON to identify symmetric integrals, the total of 693 remain: 22 2-particle cuts (matching Table 9.3.5, as we should expect), 96 3-particle cuts, 277 4-particle cuts, and 298 5-particle cuts. Note that this set is almost certainly overdetermined: even though no two integrals are symmetric, some are linearly dependent, because only removing symmetries is insufficient to undo differences in the master integral selection per family. We do not see this as a problem, however; having results for more integrals than necessary simplifies, not complicates their usage.

10.3.3. Constructing & solving differential equations

Next we shall use differential equations (Chapter 6) to calculate the master integrals. To write them down, for each integral family we differentiate the master integrals by x using

$$\frac{\partial}{\partial x}\delta\Big(x-2\frac{q{\cdot}p_n}{q^2}\Big)=-\frac{\delta\big(x-2q{\cdot}p_n/q^2\big)}{x-2q{\cdot}p_n/q^2},\tag{10.3.1}$$

express that in the $I_{\nu_1,\ldots,\nu_n}$ form, perform IBP reduction for these derivatives, and construct the differential equation matrix $\mathbb{M}(d,x)$ as in eq. (6.1.1).

Overall the reduction takes from several minutes up to 43 hours. In the end we obtain 256 matrices up to 70 × 70 in size, all block-triangular (after sorting the master integrals by their propagator set) with block sizes up to 4 × 4. Unfortunately the differential equation matrices that come directly out of this procedure sometimes have inconvenient spurious poles poles like $1/(2x-1)$, d-dependent poles like $1/(xd-2x-4d+19)$, poles given by unfactorizable polynomials like $1/\big(x^2-3x-2\big)$, or the same but also with d dependence like $1/\big(3x^2d-14x^2+6xd-28x+d-6\big)$. We see such artifacts only when there are more than one master integrals per sector, and ideally we want none of them. For this reason we additionally solve IBP relations for a small subset of integrals equal to the masters but with raised indices; then, in each sector we can modify the master integral selection (by choosing a subset of these) to try to get rid of the inconvenient denominators. Fortunately, for block sizes up to 4 trying even all possible combinations does not take too much time. In all cases we are able to find a set of master integrals that eliminate, or at least significantly reduce, the number of inconvenient denominators in the diagonal blocks: all quadratic polynomials are removed, and only rare spurious poles are left: in particular at $x=\pm 2$, but also in one case a pole of the form $1/(3+8\varepsilon+2x-2\varepsilon x)$ which is later removed during the ε-form construction. The same procedure also improves the off-diagonal blocks, even though those are generally easier to handle, so we are less concerned about them.[2]

Next, we use FUCHSIA to construct basis transformations that reduce the differential equations to ε-forms. This takes from minutes to days per integral family, with the most complex reduction taking 14 days (the integral family in question is a 5-particle cut family with 48 integrals, blocks up to 4 × 4, and the total of 5 different poles, 2 of which are spurious, and the deepest being x^{-9}). During the reduction spurious poles are necessarily removed, and in the ε-form the only remaining poles are located at

$$x=\{0,+1,-1,+2,-2\}. \tag{10.3.2}$$

The last two are a bit surprising: they are absent in the 3-loop case; we would also naively expect to have the answers in terms of harmonic polylogarithms (which correspond to poles at $\{0,\pm 1\}$), but these poles prevent us from that. Interestingly these only remain in the off-diagonal blocks; even then, we have not found a way to remove them completely, at least from the expressions for the master integrals, as we still expect these poles to drop out from the expressions for the physical quantities.

Moving on, we construct the differential equation solution for each integral $I(d,x)$ via eq. (6.1.14), eq. (6.1.13), and eq. (6.1.5).

To fix the integration constants it is sufficient to require that an integral of $I(d,x)$ over x is equal to the corresponding fully inclusive integral (the basis for which we have calculated in Chapter 9):

$$\int_0^1 \mathrm{d}x \underbrace{\int \left(\prod_i \frac{\mathrm{d}^d l_i}{(2\pi)^d}\right) \mathrm{dPS}_n(q) \left(\prod_i \frac{1}{D_i^{\nu_i}}\right) \delta\!\left(x-2\frac{q\cdot p_n}{q^2}\right)}_{I(d,x)} =$$

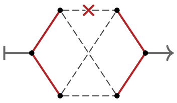
$$= \int \left(\prod_i \frac{\mathrm{d}^d l_i}{(2\pi)^d}\right) \mathrm{dPS}_n(q) \left(\prod_i \frac{1}{D_i^{\nu_i}}\right)\Bigg|_{x=2qp_n/q^2}. \tag{10.3.3}$$

Unfortunately this is not completely straightforward in practice because if we shall insert the series for $I(d,x)$ into the left hand side of this equation and integrate order by order, the integral will likely diverge. This problem is similar to what we have encountered in Section 9.3.4; let us illustrate it a bit.

10.3.3.1. Matching via integration: an illustration

For demonstration purposes let us take a look at the following 4-particle semi-inclusive cut of a 3-loop propagator,

$$I(d,x) = \quad . \tag{10.3.4}$$



[2] For more disciplined methods of removing d-dependent poles (a.k.a. “finding a d-factorizing basis”) see [94, 93].

The differential equation for it is

$$\partial_x I = \left(\frac{1+2\varepsilon}{1-x} - \frac{1+2\varepsilon}{x}\right) I, \tag{10.3.5}$$

and an ε-form can be achieved by changing the basis to such J, that

$$I(d,x) = \frac{1}{x\,(1-x)} J(d,x). \tag{10.3.6}$$

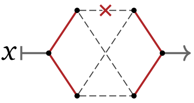
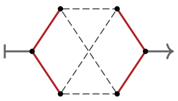
After the change of basis, the ε-form equation reads

$$\partial_x J = \varepsilon\left(\frac{2}{1-x} - \frac{2}{x}\right) J, \tag{10.3.7}$$

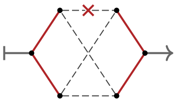
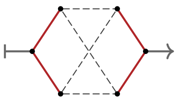
and the solution for it via eq. (6.1.9) is

$$J(d,x) = \varepsilon^{k_0}\left(C_1 + (C_2 - 2C_1 G(0;x) - 2C_1 G(1;x))\,\varepsilon + \mathcal{O}\left(\varepsilon^2\right)\right). \tag{10.3.8}$$

The integration constants C_i in principle can be determined from the condition that

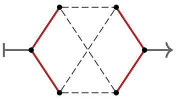
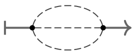
$\int \mathrm{d}x\, I(d,x) \equiv \int \mathrm{d}x$  $=$  $.$ (10.3.9)

Because the transformation eq. (10.3.6) is singular at $x \to 0$ and $x \to 1$, an integral of $I(d,x)$ defined that way will diverge in each order of the series. Instead let us note that the same integral but of $J(d,x)$ will in fact converge, because as one can see from eq. (10.3.8) at the boundaries of $x \to \{0,1\}$ $J(d,x)$ diverges at most logarithmically (any multiple polylogarithm does), and thus can be integrated over safely. This is a property shared by all solutions to differential equations in an ε-form—it is only the basis change of eq. (10.3.6) that introduces the divergence. Thus, to determine C_i we can use

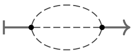
$\int \mathrm{d}x\, J(d,x) = \int \mathrm{d}x\, x\,(1-x)$  $\overset{\text{IBP}}{=} -\frac{\varepsilon}{1-4\varepsilon}$  $.$ (10.3.10)

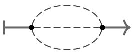
Note that the right-hand side here is one order of ε lower than it was in eq. (10.3.9); this is because the prefactor $x\,(1-x)$ has canceled a divergence—exactly as we wanted.

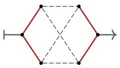
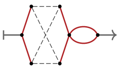
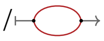
Then, using the known value of

 $= \left(60\varepsilon^{-4} - 59\varepsilon^{-3} + \mathcal{O}\left(\varepsilon^{-2}\right)\right)$  $,$ (10.3.11)

we can fix all the C_i.[3] Inserting them into eq. (10.3.8), the solution becomes

$J(d,x) = \left(-60\varepsilon^{-3} + (120G(0;x) + 120G(1;x) + 590)\,\varepsilon^{-2} + \mathcal{O}\left(\varepsilon^{-1}\right)\right)$  $,$ (10.3.12)

$I(d,x) = \left(\frac{-60}{x\,(1-x)}\varepsilon^{-3} + \frac{120G(0;x) + 120G(1;x) + 590}{x\,(1-x)}\varepsilon^{-2} + \mathcal{O}\left(\varepsilon^{-1}\right)\right)$  $.$ (10.3.13)

[3] Note that  $=$  $/$ , so we can reuse the results for 4-particle cuts of 4-loop propagators to obtain 4-particle cuts of 3-loop propagators: all of the master integrals of the latter are embedded in the former with just a bubble attached (see Table 9.3.4).

10.3.3.2. The general case

To calculate $C_i^{(k)}$ for the semi-inclusive master integrals the procedure could be the same as eq. (10.3.10),

$$\int \mathrm{d}x\,\mathbf{J}(d,x) = \int_0^1 \mathrm{d}x\,\mathbb{T}^{-1}(d,x)\,\mathbf{I}(d,x) = \big(\text{inclusive version of } \mathbb{T}^{-1}\mathbf{I}\big) \overset{\text{IBP}}{=} \ldots, \tag{10.3.14}$$

the only complication is performance: $\mathbb{T}^{-1}(d,x)$ might contain x to a high power, and because x turns into a numerator of the inclusive integral $(2q{\cdot}p_n/q^2)$, this means more work for the IBP reduction. To avoid that and to minimize the numerator power we can instead use the condition

$$\int_0^1 \mathrm{d}x\,x^a\,(1-x)^b\,\mathbf{I}(d,x) = \big(\text{inclusive version of } x^a\,(1-x)^b\,\mathbf{I}\big) \overset{\text{IBP}}{=} \ldots, \tag{10.3.15}$$

where a and b are the smallest powers that compensate the divergence of $\mathbb{T}(d,x)$ at $x \to 0$ and $x \to 1$, so that

$$x^a\mathbb{T}(d,x) = \mathscr{C} + \mathscr{O}(x) \qquad \text{and} \qquad (1-x)^b\,\mathbb{T}(d,x) = \mathscr{C}' + \mathscr{O}(1-x). \tag{10.3.16}$$

These conditions guarantee that the integrand in eq. (10.3.15), being $x^a\,(1-x)^b\,\mathbb{T}\mathbf{J}$, diverges at most logarithmically at the limits (because of $\mathbf{J}$), so the left-hand-side integral converges, while the IBP reduction on the right-hand side needs to deal with the smallest numerator powers.

Note that the idea is still the same as we had in Section 9.3.4: use a basis that does not diverge; the only thing different is the construction of this basis.

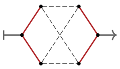
We have performed this matching for each family of semi-inclusive cuts of 4-loop propagators. Solving the IBP for inclusive integrals on average took from an hour to several days, but several exceptional cases required up to 6 weeks of time (this is with FIRE6 running on 16 cores for each family—because of the memory constraints we could not afford to use more cores). The integration of the left-hand-side of eq. (10.3.15) also takes time: when $\mathbf{I}(d,x)$ is expanded via eq. (6.1.14) up to 10 orders in ε the integration takes from several minutes up to 4 days (using our custom code, consisting of a mixture of MATHEMATICA and FORM), depending on the number of master integrals and the number of poles they have. The number of terms in the ansatz for each $\mathbf{I}(d,x)$ is roughly (number of poles)$^{(\text{orders of expansion})}$, which means that there is a practical cutoff on how many orders of expansion can the matching conditions be solved for.

A different problem is that because $x^a\,(1-x)^b\,\mathbb{T}(d,x)$ may have poles at values of x other than $\{0,\pm1\}$, for example at $x = \pm 2$, the left-hand-side of eq. (10.3.15) results in multiple polylogarithms with corresponding poles in the parameters (that is, ±2). This is a complication because if the parameters are restricted to only $\{0,\pm1\}$, then all the integrals are expressible in terms of multiple zeta values, the relations between which are well known from e.g. [108], but with ±2 in the parameter list, the constants that appear are less studied. On the other hand, because these poles are only an artifact of the way we match in eq. (10.3.15), and e.g. eq. (10.3.9) would be free from them, it is expected that the values of $C_i^{(k)}$ should have no traces of these. This is indeed the case, and for all C_i that contain $G(\ldots,\pm2,\ldots;1)$ we try to reduce them to MZVs; to this end, we first evaluate them numerically with GINAC [126, 102], and then use PSLQ to restore them in the linear basis of MZVs. This way we remove the artifacts of the matching procedure. On the other hand, because the ε-form differential equations themselves sometimes contain ±2 (see the list of poles in eq. (10.3.2)), the $G(\ldots,\pm2,\ldots;1)$ constants coming from that are not always expressible in terms of MZVs. In these cases, we leave them as they are without further reduction.

In the end we obtain ε-series for all master integrals of semi-inclusive 2-, 3-, 4-, and 5-particle cuts of 4-loop propagators. The results are quite big, so we provide them in machine-readable form as supplementary material: see Appendix C for the description of them.

In principle this concludes our calculation. There are however technical consideration of using these results in practice that we need to discuss.

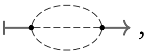
10.3.4. Factorizing boundary behavior

Let us go back to the condition of eq. (10.3.9). Integrating over $I(d,x)$ should give us , a well-defined quantity, and yet integration of $I(d,x)$ defined via eq. (10.3.6) and eq. (10.3.8) order-by-order in ε diverges. How can this be?

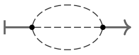
The divergence is a symptom of the fact that the expansion in the ε-series here does not capture the full behavior of I at $x \to 0$ and $x \to 1$, and one needs to sum up all of the infinite series for it to work. This should not be surprising seeing that eq. (10.3.11) is one order higher in ε than eq. (10.3.13), and this order can never be recovered if one integrates order by order. Fortunately the behavior of $J(d,x)$ at $x \to 0$ and $x \to 1$ is fully determined by the differential equation eq. (10.3.7). To allow for the correct integration, we only need to make this behavior explicit. To this end, let us expand eq. (10.3.7) in x around the boundaries:

$$J(d,x \to 0) \approx K_0 x^{-2\varepsilon}, \qquad \text{and} \qquad J(d,x \to 1) \approx K_1 (1-x)^{-2\varepsilon}. \tag{10.3.17}$$

So to fix the integration problem, the form of $J(d,x)$ should become exactly this in the corresponding limits. In other words, we should represent eq. (10.3.12) as

$$J(d,x) = x^{-2\varepsilon}(1-x)^{-2\varepsilon}\left(-60\varepsilon^{-3} + 590\varepsilon^{-2} + \mathcal{O}\left(\varepsilon^{-1}\right)\right)$$  , (10.3.18)

and then I accordingly becomes

$$I(d,x) = x^{-1-2\varepsilon}(1-x)^{-1-2\varepsilon}\left(-60\varepsilon^{-3} + 590\varepsilon^{-2} + \mathcal{O}\left(\varepsilon^{-1}\right)\right)$$  . (10.3.19)

This form can now be conveniently integrated order by order, if one notes that

$$\int_0^1 \mathrm{d}x\, x^{-1-2\varepsilon}(1-x)^{-1-2\varepsilon} = \frac{\Gamma^2(-2\varepsilon)}{\Gamma(-4\varepsilon)} = -\frac{1}{\varepsilon} + 0 + \mathcal{O}(\varepsilon). \tag{10.3.20}$$

Using this relation we can easily see that the integral of eq. (10.3.19) is indeed equal to eq. (10.3.11).

Some notes are due here. This was a simple example, but for more complicated integrals there are additional things to consider:

1. There can be a mix of different powers in the boundary expansion in eq. (10.3.17), both at $x \to 0$ and at $x \to 1$. If this is so, then we shall present the integral as a sum of multiple terms, each with a different prefactor being a combination of x and $1-x$ powers.

2. Once the boundary behavior is separated into an irrational prefactor, the remaining series in ε in eq. (10.3.19) need not be just a constant: it can contain additional x dependence. In such cases integrating the obtained expressions is not as easy as eq. (10.3.20). Let us see how this can be done.

10.3.4.1. Integration with irrational prefactors and the plus-distribution

Let us take a moment to solve an exercise: if there is an expression with an irrational prefactor of the form

$$F(x) = (1-x)^{-n+k\varepsilon} K(x), \tag{10.3.21}$$

where K is some function finite at $x \in \{0,1\}$, how do we integrate it? How to find $\int_0^1 F(x)\,\mathrm{d}x$?

As previously explained, in general we cannot expand $(1-x)^{-n+k\varepsilon}$ into a series in ε, and integrate order by order, because we might get divergences near $x=1$. The solution is to subtract the divergent part of the integrand.

Let us first look at the simple case when $n=1$. Note that integrating the prefactor itself in this case is a simple calculation:

$$\int_0^1 (1-x)^{-1+k\varepsilon}\,\mathrm{d}x = \frac{1}{k\varepsilon}; \tag{10.3.22}$$

the trick is then to separate the divergence at $x=1$ and take it using this formula, leaving the rest of the integral finite so that it could be taken via expansion into series. Here is the method:

$$\begin{aligned}\int_0^1 (1-x)^{-1+k\varepsilon} F(x)\,\mathrm{d}x &= \int_0^1 (1-x)^{-1+k\varepsilon} \left(F(x)-F(1)+F(1)\right)\mathrm{d}x = \\ &= \underbrace{\int_0^1 (1-x)^{-1+k\varepsilon}\left(F(x)-F(1)\right)\mathrm{d}x}_{\text{finite}} + \frac{F(1)}{k\varepsilon}.\end{aligned} \tag{10.3.23}$$

The usual convention is to introduce a special notation for the finite integrand, the *plus distribution*. It is defined as

$$[f(x)]_+ \equiv f(x) - \delta(x-1)\int_0^1 f(y)\,\mathrm{d}y, \tag{10.3.24}$$

which is another way of saying

$$\int_0^1 [f(x)]_+\, g(x)\,\mathrm{d}x \equiv \int_0^1 f(x)\left(g(x)-g(1)\right)\mathrm{d}x. \tag{10.3.25}$$

With this definition we can rewrite the answer as

$$\int_0^1 (1-x)^{-1+k\varepsilon} F(x)\,\mathrm{d}x = \int_0^1 \left[(1-x)^{-1+k\varepsilon}\right]_+ F(x)\,\mathrm{d}x + \frac{F(1)}{k\varepsilon}. \tag{10.3.26}$$

The integral here is finite, so the integrand can be expanded into a series in a straightforward way:

$$\left[(1-x)^{-1+k\varepsilon}\right]_+ = \sum_{i=0}^{\infty} \frac{k^i}{i!}\left[\frac{\ln^i(1-x)}{1-x}\right]_+ \varepsilon^i. \tag{10.3.27}$$

Note that the plus distribution is a notion particular to integration from 0 to 1 with the divergence located at $x=1$. One might imagine needing a similar construction for the divergence at $x=0$, or some other point; so far such extensions have not been required for the splitting function calculation.

In the more general case of

$$\int_0^1 (1-x)^{-n+k\varepsilon} F(x)\,\mathrm{d}x, \tag{10.3.28}$$

the idea is the same, and we can generalize eq. (10.3.23) as

$$\int_0^1 (1-x)^{-n+k\varepsilon} F(x)\,\mathrm{d}x = \underbrace{\int_0^1 (1-x)^{-n+k\varepsilon}\left(F(x)-\sum_{i=0}^{n-1}(x-1)^i\,\frac{F^{(i)}(1)}{i!}\right)\mathrm{d}x}_{\text{finite}} + \sum_{i=0}^{n-1}\frac{(-1)^i}{i+1-n+k\varepsilon}\,\frac{F^{(i)}(1)}{i!}, \tag{10.3.29}$$

where $F^{(i)}(x)$ is the i-th derivative of $F(x)$, which we assume exists. Accordingly, we can introduce the n-th order plus distribution as

$$\int_0^1 [f(x)]_{+^n}\, g(x)\,\mathrm{d}x \equiv \int_0^1 f(x)\left(g(x)-\sum_{i=0}^{n-1}(x-1)^i\,\frac{g^{(i)}(1)}{i!}\right)\mathrm{d}x, \tag{10.3.30}$$

and have

$$\int_0^1 (1-x)^{-n+k\varepsilon} F(x)\,\mathrm{d}x = \int_0^1 \left[(1-x)^{-n+k\varepsilon}\right]_{+^n} F(x)\,\mathrm{d}x + \sum_{i=0}^{n-1}\frac{(-1)^i}{i+1-n+k\varepsilon}\,\frac{F^{(i)}(1)}{i!}. \tag{10.3.31}$$

10.3.4.2. The case of multiple irrational prefactors

The general way to factor out the boundary behavior is to start with the ε-form eq. (6.1.7), and only consider its behavior at $x \to 0$ and $x \to 1$:

$$\partial_x \mathbf{J}(d,x) = \varepsilon\frac{\mathbb{S}_0}{x}\,(1+\mathcal{O}(x))\,\mathbf{J}(d,x) \qquad \text{and} \qquad \partial_x \mathbf{J}(d,x) = \varepsilon\frac{\mathbb{S}_1}{x-1}\,(1+\mathcal{O}(x-1))\,\mathbf{J}(d,x); \tag{10.3.32}$$

these equations can then be solved as

$$\mathbf{J}(d,x) = e^{\varepsilon\mathbb{S}_0 \ln x}\,(1+\mathcal{O}(x)) \qquad \text{and} \qquad J(d,x) = e^{\varepsilon\mathbb{S}_1 \ln(1-x)}\,(1+\mathcal{O}(1-x)). \tag{10.3.33}$$

In the special case of the example of eq. (10.3.7) this gives the results in eq. (10.3.17). In more general cases the matrix exponent needs to be calculated.[4] Knowing this boundary behavior we can then rewrite the ε-form solution given by eq. (6.1.14) and eq. (6.1.13) as

$$\mathbf{J}(d,x) = e^{\varepsilon\mathbb{S}_1 \ln(1-x)}\tilde{\mathbb{W}}(d,x)\,e^{\varepsilon\mathbb{S}_0 \ln x}\,\mathbf{C}(d), \tag{10.3.34}$$

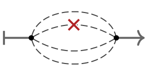
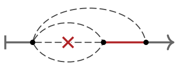
where $\tilde{\mathbb{W}}(d,x)$ is the fundamental solution regularized at $x \to \{0,1\}$,

$$\tilde{\mathbb{W}}(d,x) \equiv e^{-\varepsilon\mathbb{S}_1 \ln(1-x)}\tilde{\mathbb{W}}(d,x)\,e^{-\varepsilon\mathbb{S}_0 \ln x}. \tag{10.3.35}$$

This construction of $\tilde{\mathbb{W}}(d,x)$ ensures that no multiple polylogarithm in its expansion has the form $G(1,\ldots;x)$ or $G(\ldots,0;x)$, because all the leading ones and trailing zeros have been factorized. The idea here is to only expand $\tilde{\mathbb{W}}(d,x)$ into a series in ε, and keep the rest of the factors in eq. (10.3.34) unexpanded, just like in eq. (10.3.19).

10.3.4.3. An example with multiple prefactors

Let us illustrate the construction of eq. (10.3.34) with an example given by the integrals

$$\mathbf{I}(d,x) \equiv \begin{pmatrix} I_1 \\ I_2 \end{pmatrix}, \qquad \text{where} \qquad I_i \equiv$$

$$, \qquad \text{and} \qquad I_2 \equiv$$

$$. \tag{10.3.36}$$

[4] For example using the `MatrixExp` function in Mathematica.

The differential equation system for these two is

$$\partial_x \mathbf{I} = \begin{pmatrix} \frac{1-2\varepsilon}{x} + \frac{-1+2\varepsilon}{1-x} & 0 \\ \frac{-2+3\varepsilon}{x}\frac{1}{q^2} + \frac{-2+3\varepsilon}{1-x}\frac{1}{q^2} & \frac{1-3\varepsilon}{x} \end{pmatrix} \mathbf{I}, \tag{10.3.37}$$

and the ε-form can be achieved with the transformation found by FUCHSIA,

$$\mathbf{I} = \begin{pmatrix} \varepsilon x(1-x) q^2 & 0 \\ 0 & (2-3\varepsilon) x \end{pmatrix} \mathbf{J}, \tag{10.3.38}$$

giving us the ε-form of

$$\partial_x \mathbf{J} = \varepsilon \begin{pmatrix} \frac{2}{1-x} - \frac{2}{x} & 0 \\ -\frac{1}{x} & -\frac{3}{x} \end{pmatrix} \mathbf{J}. \tag{10.3.39}$$

The asymptotic behavior of $\mathbf{J}(d,x)$ in the limit $x \to 0$ can then be found as

$$\mathbf{J}(d,x) = \exp\left(\varepsilon \begin{pmatrix} -2 & 0 \\ -1 & -3 \end{pmatrix} \ln x \right) (\mathbf{C}_0(d) + \mathcal{O}(x)) = \begin{pmatrix} x^{-2\varepsilon} & 0 \\ x^{-3\varepsilon} - x^{-2\varepsilon} & x^{-3\varepsilon} \end{pmatrix} (\mathbf{C}_0(d) + \mathcal{O}(x)), \tag{10.3.40}$$

where $\mathbf{C}_0$ is a vector of integration constants.

Here we can see the main difference from the previous example: instead of having a single irrational prefactor, J_2 (and thus, I_2) behaves as a mixture of $x^{-2\varepsilon}$ and $x^{-3\varepsilon}$ at $x \to 0$.

Similarly, for the $x \to 1$ limit we have

$$J(d,x) = \exp\left(\varepsilon \begin{pmatrix} -2 & 0 \\ 0 & 0 \end{pmatrix} \ln(1-x) \right) (\mathbf{C}_0(d) + \mathcal{O}(1-x)) = \begin{pmatrix} (1-x)^{-2\varepsilon} & 0 \\ 0 & 0 \end{pmatrix} (\mathbf{C}_0(d) + \mathcal{O}(1-x)). \tag{10.3.41}$$

Taking into account the transformation from eq. (10.3.38) we can conclude that

$$I_1 \propto x^{1-2x} (1-x)^{1-2\varepsilon}, \tag{10.3.42}$$

which of course matches with what we expect from the full value of I_1 given by eq. (10.1.6); I_2 on the other hand is a mixture,

$$I_2 \propto A x^{1-3\varepsilon} + B x^{1-2\varepsilon}. \tag{10.3.43}$$

Walking through the same steps of constructing the ε-form solution using eq. (6.1.14), matching with the fully inclusive integrals, and applying eq. (10.3.34), we can obtain the following values of $I(d,x)$:

$$\begin{aligned} I_1 &= x^{1-2\varepsilon} (1-x)^{1-2\varepsilon} \left(6 - 20\varepsilon + (24\zeta_2 - 8)\varepsilon^2 + \dots\right), \\ I_2 &= x^{1-3\varepsilon} \left(\frac{12}{\varepsilon} - 58 + (72\zeta_2 + 44)\varepsilon + \dots \right) + \\ &+ x^{1-2\varepsilon} \left(-\frac{12}{\varepsilon} + 58 + (24 G(0,1;x) - 48\zeta_2 - 44)\varepsilon + \dots \right). \end{aligned} \tag{10.3.44}$$

The same procedure of factorizing the behavior at boundaries can be applied to the results for semi-inclusive cuts of 4-loop propagators. We have done so, and have obtained the results in factorized form similar to eq. (10.3.44) for each master integral. These results are available in machine-readable form, as described in Appendix C.

10.4. Semi-inclusive cuts of three-loop propagators

The three-loop semi-inclusive cut master integrals have been previously computed in [69]. As a cross-check we have re-computed these using the same method as the 4-loop ones, and we confirm those results with the following caveats:

1. The overall normalization factors are not listed in [69], and are assumed to be the same as in [45]. The latter defines its integrals in eq. (A.1) and eq. (A.11) of Appendix A, with "V" integrals defined (in our notation) as

$$V_{i_1,\ldots,i_k}(n) \equiv \frac{e^{3\gamma_E \varepsilon}}{\pi^4} (2\pi)^{3d-3} \int \frac{\mathrm{d}^d l}{(2\pi)^d} \mathrm{dPS}_3 \frac{(2q\cdot p_3)^n}{D_{i_1}\cdots D_{i_k}}, \tag{10.4.1}$$

   where dPS_n is given by eq. (9.1.2), and "R" integrals as

$$R_{i_1,\ldots,i_k}(n) \equiv \frac{e^{3\gamma_E \varepsilon}}{\pi^3} (2\pi)^{3d-4} \int \mathrm{dPS}_4 \frac{\left(2q\cdot p_4\right)^n}{D_{i_1}\cdots D_{i_k}}. \tag{10.4.2}$$

   However, to make the value of $R_5(0)$ reported in eq. (A.21) consistent with eq. (9.1.14), or indeed with that of [139] (which are the same), the prefactor in eq. (10.4.2) should read $\frac{e^{3\gamma_E\varepsilon}}{\pi^{3-3\varepsilon}}$, not $\frac{e^{3\gamma_E\varepsilon}}{\pi^3}$. We have not been able to determine a similarly concise change for the prefactors of V integrals, and we can only report that the values agree with ours up to a global constant factor.

2. Typos: in eq. (A.21) "$x^{-2\varepsilon}$" should read $(1-x)^{-2\varepsilon}$, and "$+2\zeta_2$" should read $-2\zeta_2\mathrm{H}_1$. In eq. (A.26) "$(1-x)^{-1-2\varepsilon}$" should be "$(1+x)^{-1-2\varepsilon}$" instead.[5]

3. The irrational prefactors separated in [69] are not not equivalent to ours from Section 10.3.4, because they don't always fully factorize the logarithms of x and $1-x$. Because of this we have only compared the series-expanded forms of the results.

[5] We have confirmed these errors with the author.

11. Summary and outlook

In this thesis we have presented the analytic calculation of all master integrals for semi-inclusive cuts of 4-loop massless propagators (Chapter 10) required for the extraction of *photonic decay coefficient functions* up to $\mathcal{O}\big(\alpha_s^3\big)$: these enter the analysis of semi-inclusive single-hadron production at e^+e^- colliders (such as the upcoming FCC-ee, ILC, CLIC, and CEPC), and are currently only known up to $\mathcal{O}\big(\alpha_s^2\big)$. The same integrals can be used to extract *time-like splitting functions* up to NNLO precision in α_s: together with the corresponding fragmentation functions these enter the analysis of semi-inclusive hadron production at both e^+e^- and pp colliders; the calculation of the full set of NNLO terms has been a long-standing problem [24, 25, 26] resolved only recently by a calculation in [27] using an independent approach.

Additionally we have presented the calculation of the full set of fully-inclusive cuts of 4-loop propagators (Chapter 9, also [35, 36]), and a recalculation of the 2-loop 1 → 3 integrals first appearing in [122, 116] to a higher transcendental weight (Section 6.4). These sets of integrals were needed to calculate semi-inclusive cuts, but can be useful in their own right too.

Along with the integrals we have described the methods and presented the tools developed for this calculation. The first such tool is FUCHSIA (Chapter 8), a program to construct ε-forms of differential equations for Feynman master integrals, making the differential equation method possible. FUCHSIA is a general tool useful for many systems of master integrals, and has already been used in multiple independent calculations such as [31, 32, 33, 34]. The second tool is FEYNSON (Chapter 4), a program that resolves symmetries between Feynman integrals.

As the direct follow-up to this work we forsee using the calculated integrals to recalculate the NNLO time-like splitting functions, and to calculate for the first time $\mathcal{O}\big(\alpha_s^3\big)$ photonic decay coefficient functions.

Acknowledgements

This work would not exist without Oleksandr Gituliar, who convinced me to rejoin the academia, and Sven Moch, who invited me to work in Hamburg and supported me over the past years. I'm grateful to my co-authors, Oleksandr Gituliar and Andrey Pikelner, with whom a part of this work was completed.

At the risk of sounding sentimental, I'd like to say that scientific work as practiced today would be unbearable without friends that one makes at the work place. I'm grateful to Sergey Alekhin, Benoît Assi, Pia Bredt, Vova Bytev, Bakar Chargeishvili, Seva Chestnov, Constantin Harder, Toni Mäkelä, Andrei Onishchenko, Sam Van Thurenhout, Oleg Veretin, Katharina Voß, and all the other friends I made in Hamburg for making my stay there enjoyable.

Finally, I thank Lisa Biermann for drafting the Zusammenfassung in German, and Sven Moch for suggestions on the structuring of this text, for proofreading and identifying the many mistakes it contained.

Bibliography

[1] T. Kajita. “Nobel Lecture: Discovery of atmospheric neutrino oscillations”. In: *Rev. Mod. Phys.* 88.3 (2016), p. 030501. DOI: 10.1103/RevModPhys.88.030501.

[2] A. B. McDonald. “Nobel Lecture: The Sudbury Neutrino Observatory: Observation of flavor change for solar neutrinos”. In: *Rev. Mod. Phys.* 88.3 (2016), p. 030502. DOI: 10.1103/RevModPhys.88.030502.

[3] R. Acciarri et al. “Long-Baseline Neutrino Facility (LBNF) and Deep Underground Neutrino Experiment (DUNE): Conceptual Design Report, Volume 2: The Physics Program for DUNE at LBNF”. In: (Dec. 2015). arXiv: 1512.06148.

[4] K. Abe et al. “Hyper-Kamiokande Design Report”. In: (May 2018). arXiv: 1805.04163.

[5] P. W. Graham et al. “Experimental Searches for the Axion and Axion-Like Particles”. In: *Ann. Rev. Nucl. Part. Sci.* 65 (2015), pp. 485–514. DOI: 10.1146/annurev-nucl-102014-022120. arXiv: 1602.00039 [hep-ex].

[6] A. D. Sakharov. “Violation of CP Invariance, C asymmetry, and baryon asymmetry of the universe”. In: *Zh. Eksp. Teor. Fiz.* 5 (1967), pp. 32–35. DOI: 10.1070/PU1991v034n05ABEH002497.

[7] D. E. Morrissey and M. J. Ramsey-Musolf. “Electroweak baryogenesis”. In: *New J. Phys.* 14 (2012), p. 125003. DOI: 10.1088/1367-2630/14/12/125003. arXiv: 1206.2942 [hep-ph].

[8] S. Davidson, E. Nardi, and Y. Nir. “Leptogenesis”. In: *Phys. Rept.* 466 (2008), pp. 105–177. DOI: 10.1016/j.physrep.2008.06.002. arXiv: 0802.2962 [hep-ph].

[9] F. Mayet et al. “A review of the discovery reach of directional Dark Matter detection”. In: *Phys. Rept.* 627 (2016), pp. 1–49. DOI: 10.1016/j.physrep.2016.02.007. arXiv: 1602.03781 [astro-ph.CO].

[10] E. D. Bloom et al. “High-Energy Inelastic $e\ p$ Scattering at 6-Degrees and 10-Degrees”. In: *Phys. Rev. Lett.* 23 (1969), pp. 930–934. DOI: 10.1103/PhysRevLett.23.930. URL: https://slac.stanford.edu/pubs/slacpubs/0500/slac-pub-0642.pdf.

[11] M. Breidenbach et al. “Observed behavior of highly inelastic electron-proton scattering”. In: *Phys. Rev. Lett.* 23 (1969), pp. 935–939. DOI: 10.1103/PhysRevLett.23.935. URL: https://slac.stanford.edu/pubs/slacpubs/0500/slac-pub-0650.pdf.

[12] C. Berger et al. “Observation of a Narrow Resonance Formed in e^+e^- Annihilation at 9.46 GeV”. In: *Phys. Lett. B* 76 (1978), pp. 243–245. DOI: 10.1016/0370-2693(78)90287-3.

[13] B. R. Stella and H.-J. Meyer. “Υ(9.46 GeV) and the gluon discovery (a critical recollection of PLUTO results)”. In: *Eur. Phys. J. H* 36 (2011), pp. 203–243. DOI: 10.1140/epjh/e2011-10029-3. arXiv: 1008.1869 [hep-ex].

[14] G. Arnison et al. “Experimental Observation of Isolated Large Transverse Energy Electrons with Associated Missing Energy at $\sqrt{s} = 540$ GeV”. In: *Phys. Lett. B* 122 (1983), pp. 103–116. DOI: 10.1016/0370-2693(83)91177-2.

[15] G. Arnison et al. “Experimental Observation of Lepton Pairs of Invariant Mass Around 95 GeV/c^2 at the CERN SPS Collider”. In: *Phys. Lett. B* 126 (1983), pp. 398–410. DOI: 10.1016/0370-2693(83)90188-0.

[16] G. Aad et al. “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC”. In: *Phys. Lett. B* 716 (2012), pp. 1–29. DOI: 10.1016/j.physletb.2012.08.020. arXiv: 1207.7214 [hep-ex].

[17] S. Chatrchyan et al. “Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC”. In: *Phys. Lett. B* 716 (2012), pp. 30–61. DOI: 10.1016/j.physletb.2012.08.021. arXiv: 1207.7235 [hep-ex].

[18] “High-Luminosity Large Hadron Collider (HL-LHC) : Preliminary Design Report”. In: (Dec. 2015). Ed. by G Apollinari et al. DOI: 10.5170/CERN-2015-005.

[19] ATLAS Collaboration. “Expected performance of the ATLAS detector at the High-Luminosity LHC”. In: (2019). URL: https://cds.cern.ch/record/2655304.

[20] A. Abada et al. “FCC Physics Opportunities: Future Circular Collider Conceptual Design Report Volume 1”. In: *Eur. Phys. J. C* 79.6 (2019), p. 474. DOI: 10.1140/epjc/s10052-019-6904-3.

[21] “The International Linear Collider Technical Design Report - Volume 1: Executive Summary”. In: (June 2013). Ed. by T. Behnke et al. arXiv: 1306.6327 [physics.acc-ph].

[22] T. K. Charles et al. “The Compact Linear Collider (CLIC) - 2018 Summary Report”. In: 2/2018 (Dec. 2018). Ed. by P. N. Burrows et al. DOI: 10.23731/CYRM-2018-002. arXiv: 1812.06018 [physics.acc-ph].

[23] M. Dong et al. “CEPC Conceptual Design Report: Volume 2 - Physics & Detector”. In: (Nov. 2018). Ed. by J. B. Guimarães da Costa et al. arXiv: 1811.10545 [hep-ex].

[24] A. Mitov, S. Moch, and A. Vogt. “Next-to-Next-to-Leading Order Evolution of Non-Singlet Fragmentation Functions”. In: *Phys. Lett. B* 638 (2006), pp. 61–67. DOI: 10.1016/j.physletb.2006.05.005. arXiv: hep-ph/0604053.

[25] S. Moch and A. Vogt. “On third-order timelike splitting functions and top-mediated Higgs decay into hadrons”. In: *Phys. Lett. B* 659 (2008), pp. 290–296. DOI: 10.1016/j.physletb.2007.10.069. arXiv: 0709.3899 [hep-ph].

[26] A. A. Almasy, S. Moch, and A. Vogt. “On the Next-to-Next-to-Leading Order Evolution of Flavour-Singlet Fragmentation Functions”. In: *Nucl. Phys. B* 854 (2012), pp. 133–152. DOI: 10.1016/j.nuclphysb.2011.08.028. arXiv: 1107.2263 [hep-ph].

[27] H. Chen et al. “Analytic Continuation and Reciprocity Relation for Collinear Splitting in QCD”. In: *Chin. Phys. C* 45.4 (2021), p. 043101. DOI: 10.1088/1674-1137/abde2d. arXiv: 2006.10534 [hep-ph].

[28] O. Gituliar and S.-O. Moch. “Towards three-loop QCD corrections to the time-like splitting functions”. In: *Acta Phys. Polon. B* 46.7 (2015), pp. 1279–1289. DOI: 10.5506/APhysPolB.46.1279. arXiv: 1505.02901 [hep-ph].

[29] J. M. Henn. “Multiloop integrals in dimensional regularization made simple”. In: *Phys. Rev. Lett.* 110 (2013), p. 251601. DOI: 10.1103/PhysRevLett.110.251601. arXiv: 1304.1806 [hep-th].

[30] R. N. Lee. “Reducing differential equations for multiloop master integrals”. In: *JHEP* 04 (2015), p. 108. DOI: 10.1007/JHEP04(2015)108. arXiv: 1411.0911 [hep-ph].

[31] M. Diehl, J. R. Gaunt, and P. Plößl. “Two-loop splitting in double parton distributions: the colour non-singlet case”. In: *JHEP* 08 (2021), p. 040. DOI: `10.1007/JHEP08(2021)040`. arXiv: `2105.08425 [hep-ph]`.

[32] B. A. Kniehl et al. “Two-loop diagrams in non-relativistic QCD with elliptics”. In: *Nucl. Phys. B* 948 (2019), p. 114780. DOI: `10.1016/j.nuclphysb.2019.114780`. arXiv: `1907.04638 [hep-ph]`.

[33] K. Melnikov et al. “Triple-real contribution to the quark beam function in QCD at next-to-next-to-next-to-leading order”. In: *JHEP* 06 (2019), p. 033. DOI: `10.1007/JHEP06(2019)033`. arXiv: `1904.02433 [hep-ph]`.

[34] L. J. Dixon et al. “Analytical Computation of Energy-Energy Correlation at Next-to-Leading Order in QCD”. In: *Phys. Rev. Lett.* 120.10 (2018), p. 102001. DOI: `10.1103/PhysRevLett.120.102001`. arXiv: `1801.03219 [hep-ph]`.

[35] V. Magerya and A. Pikelner. “Cutting massless four-loop propagators”. In: *JHEP* 12 (2019), p. 026. DOI: `10.1007/JHEP12(2019)026`. arXiv: `1910.07522 [hep-ph]`.

[36] O. Gituliar, V. Magerya, and A. Pikelner. “Five-Particle Phase-Space Integrals in QCD”. In: *JHEP* 06 (2018), p. 099. DOI: `10.1007/JHEP06(2018)099`. arXiv: `1803.09084 [hep-ph]`.

[37] O. Gituliar and V. Magerya. “Fuchsia: a tool for reducing differential equations for Feynman master integrals to epsilon form”. In: *Comput. Phys. Commun.* 219 (2017), pp. 329–338. DOI: `10.1016/j.cpc.2017.05.004`. arXiv: `1701.04269 [hep-ph]`.

[38] O. Gituliar and V. Magerya. “Fuchsia and master integrals for splitting functions from differential equations in QCD”. In: *PoS* LL2016 (2016), p. 030. DOI: `10.22323/1.260.0030`. arXiv: `1607.00759 [hep-ph]`.

[39] R. D. Field. *Applications of Perturbative QCD*. Frontiers in Physics. Addison-Wesley, 1989. ISBN: 0-201-14295-3.

[40] R. K. Ellis, W. J. Stirling, and B. R. Webber. *QCD and Collider Physics*. Cambridge Monographs on Particle Physics, Nuclear Physics and Cosmology. Cambridge University Press, 1996. DOI: `10.1017/CBO9780511628788`.

[41] Y. L. Dokshitzer. “Calculation of the Structure Functions for Deep Inelastic Scattering and e^+ e^- Annihilation by Perturbation Theory in Quantum Chromodynamics.” In: *Sov. Phys. JETP* 46 (1977), pp. 641–653.

[42] V. N. Gribov and L. N. Lipatov. “Deep inelastic e p scattering in perturbation theory”. In: *Sov. J. Nucl. Phys.* 15 (1972), pp. 438–450.

[43] G. Altarelli and G. Parisi. “Asymptotic Freedom in Parton Language”. In: *Nucl. Phys. B* 126 (1977), pp. 298–318. DOI: `10.1016/0550-3213(77)90384-4`.

[44] P. A. Zyla et al. “Review of Particle Physics”. In: *PTEP* 2020.8 (2020), p. 083C01. DOI: `10.1093/ptep/ptaa104`.

[45] A. Mitov and S.-O. Moch. “QCD Corrections to Semi-Inclusive Hadron Production in Electron-Positron Annihilation at Two Loops”. In: *Nucl. Phys. B* 751 (2006), pp. 18–52. DOI: `10.1016/j.nuclphysb.2006.05.018`. arXiv: `hep-ph/0604160`.

[46] D. A. Kosower and P. Uwer. “Evolution kernels from splitting amplitudes”. In: *Nucl. Phys. B* 674 (2003), pp. 365–400. DOI: `10.1016/j.nuclphysb.2003.09.044`. arXiv: `hep-ph/0307031`.

[47] P. J. Rijken and W. L. van Neerven. “Higher order QCD corrections to the transverse and longitudinal fragmentation functions in electron-positron annihilation”. In: *Nucl. Phys. B* 487 (1997), pp. 233–282. DOI: 10.1016/S0550-3213(96)00669-4. arXiv: hep-ph/9609377.

[48] R. Baier and K. Fey. “Finite corrections to quark fragmentation functions in perturbative QCD”. In: *Z. Phys. C* 2 (1979), pp. 339–349. DOI: 10.1007/BF01545897.

[49] G. ’t Hooft and M. J. G. Veltman. “Regularization and Renormalization of Gauge Fields”. In: *Nucl. Phys. B* 44 (1972), pp. 189–213. DOI: 10.1016/0550-3213(72)90279-9.

[50] J. C. Collins. *Renormalization: An Introduction to Renormalization, the Renormalization Group and the Operator-Product Expansion*. Cambridge Monographs on Mathematical Physics. Cambridge University Press, 1984. DOI: 10.1017/CBO9780511622656.

[51] W. A. Bardeen et al. “Deep Inelastic Scattering Beyond the Leading Order in Asymptotically Free Gauge Theories”. In: *Phys. Rev. D* 18 (1978), p. 3998. DOI: 10.1103/PhysRevD.18.3998.

[52] G. ’t Hooft. “Dimensional regularization and the renormalization group”. In: *Nucl. Phys. B* 61 (1973), pp. 455–468. DOI: 10.1016/0550-3213(73)90376-3.

[53] T. van Ritbergen, J. A. M. Vermaseren, and S. A. Larin. “The Four loop beta function in quantum chromodynamics”. In: *Phys. Lett. B* 400 (1997), pp. 379–384. DOI: 10.1016/S0370-2693(97)00370-5. arXiv: hep-ph/9701390.

[54] P. A. Baikov, K. G. Chetyrkin, and J. H. Kühn. “Five-Loop Running of the QCD coupling constant”. In: *Phys. Rev. Lett.* 118.8 (2017), p. 082002. DOI: 10.1103/PhysRevLett.118.082002. arXiv: 1606.08659 [hep-ph].

[55] F. Herzog et al. “The five-loop beta function of Yang-Mills theory with fermions”. In: *JHEP* 02 (2017), p. 090. DOI: 10.1007/JHEP02(2017)090. arXiv: 1701.01404 [hep-ph].

[56] T. Luthe et al. “The five-loop Beta function for a general gauge group and anomalous dimensions beyond Feynman gauge”. In: *JHEP* 10 (2017), p. 166. DOI: 10.1007/JHEP10(2017)166. arXiv: 1709.07718 [hep-ph].

[57] J. C. Collins and D. E. Soper. “The Theorems of Perturbative QCD”. In: *Ann. Rev. Nucl. Part. Sci.* 37 (1987), pp. 383–409. DOI: 10.1146/annurev.ns.37.120187.002123.

[58] D. J. Gross and F. Wilczek. “Asymptotically Free Gauge Theories. I”. In: *Phys. Rev. D* 8 (1973), pp. 3633–3652. DOI: 10.1103/PhysRevD.8.3633.

[59] D. J. Gross and F. Wilczek. “Asymptotically free gauge theories. II”. In: *Phys. Rev. D* 9 (1974), pp. 980–993. DOI: 10.1103/PhysRevD.9.980.

[60] G. Curci, W. Furmanski, and R. Petronzio. “Evolution of Parton Densities Beyond Leading Order: The Nonsinglet Case”. In: *Nucl. Phys. B* 175 (1980), pp. 27–92. DOI: 10.1016/0550-3213(80)90003-6.

[61] W. Furmanski and R. Petronzio. “Singlet Parton Densities Beyond Leading Order”. In: *Phys. Lett. B* 97 (1980), pp. 437–442. DOI: 10.1016/0370-2693(80)90636-X.

[62] S. Moch, J. A. M. Vermaseren, and A. Vogt. “The Three loop splitting functions in QCD: The Nonsinglet case”. In: *Nucl. Phys. B* 688 (2004), pp. 101–134. DOI: 10.1016/j.nuclphysb.2004.03.030. arXiv: hep-ph/0403192.

[63] A. Vogt, S. Moch, and J. A. M. Vermaseren. “The Three-loop splitting functions in QCD: The Singlet case”. In: *Nucl. Phys. B* 691 (2004), pp. 129–181. DOI: 10.1016/j.nuclphysb.2004.04.024. arXiv: hep-ph/0404111.

[64] S. Moch et al. "Four-Loop Non-Singlet Splitting Functions in the Planar Limit and Beyond". In: *JHEP* 10 (2017), p. 041. DOI: 10.1007/JHEP10(2017)041. arXiv: 1707.08315 [hep-ph].

[65] S. Moch et al. "On quartic colour factors in splitting functions and the gluon cusp anomalous dimension". In: *Phys. Lett. B* 782 (2018), pp. 627–632. DOI: 10.1016/j.physletb.2018.06.017. arXiv: 1805.09638 [hep-ph].

[66] G. Altarelli et al. "Processes Involving Fragmentation Functions Beyond the Leading Order in QCD". In: *Nucl. Phys. B* 160 (1979), pp. 301–329. DOI: 10.1016/0550-3213(79)90062-2.

[67] P. J. Rijken and W. L. van Neerven. "$O(\alpha_s^2)$ contributions to the longitudinal fragmentation function in e^+e^- annihilation". In: *Phys. Lett. B* 386 (1996), pp. 422–428. DOI: 10.1016/0370-2693(96)00898-2. arXiv: hep-ph/9604436.

[68] P. J. Rijken and W. L. van Neerven. "$O(\alpha_s^2)$ contributions to the asymmetric fragmentation function in e^+e^- annihilation". In: *Phys. Lett. B* 392 (1997), pp. 207–215. DOI: 10.1016/S0370-2693(96)01529-8. arXiv: hep-ph/9609379.

[69] O. Gituliar. "Master integrals for splitting functions from differential equations in QCD". In: *JHEP* 02 (2016), p. 017. DOI: 10.1007/JHEP02(2016)017. arXiv: 1512.02045 [hep-ph].

[70] G. Heinrich, T. Huber, and D. Maître. "Master integrals for fermionic contributions to massless three-loop form-factors". In: *Phys. Lett. B* 662 (2008), pp. 344–352. DOI: 10.1016/j.physletb.2008.03.028. arXiv: 0711.3590 [hep-ph].

[71] G. Heinrich et al. "Nine-Propagator Master Integrals for Massless Three-Loop Form Factors". In: *Phys. Lett. B* 678 (2009), pp. 359–366. DOI: 10.1016/j.physletb.2009.06.038. arXiv: 0902.3512 [hep-ph].

[72] R. N. Lee, A. V. Smirnov, and V. A. Smirnov. "Analytic Results for Massless Three-Loop Form Factors". In: *JHEP* 04 (2010), p. 020. DOI: 10.1007/JHEP04(2010)020. arXiv: 1001.2887 [hep-ph].

[73] R. N. Lee and A. A. Pomeransky. "Critical points and number of master integrals". In: *JHEP* 11 (2013), p. 165. DOI: 10.1007/JHEP11(2013)165. arXiv: 1308.6676 [hep-ph].

[74] A. Pak. "The Toolbox of modern multi-loop calculations: novel analytic and semi-analytic techniques". In: *J. Phys. Conf. Ser.* 368 (2012). Ed. by L. Teodorescu et al., p. 012049. DOI: 10.1088/1742-6596/368/1/012049. arXiv: 1111.0868 [hep-ph].

[75] B. D. McKay and A. Piperno. "Practical graph isomorphism, II". In: *Journal of Symbolic Computation* 60 (2014), pp. 94–112. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2013.09.003. URL: http://pallini.di.uniroma1.it.

[76] P. T. Darga et al. "Exploiting Structure in Symmetry Detection for CNF". In: *Proceedings of the 41st Annual Design Automation Conference*. DAC '04. 2004, pp. 530–534. ISBN: 1581138288. DOI: 10.1145/996566.996712. URL: https://doi.org/10.1145/996566.996712.

[77] T. Junttila and P. Kaski. "Engineering an Efficient Canonical Labeling Tool for Large and Sparse Graphs". In: *2007 Proceedings of the Workshop on Algorithm Engineering and Experiments (ALENEX)*. 2007, pp. 135–149. DOI: 10.1137/1.9781611972870.13.

[78] R. N. Lee. "LiteRed 1.4: a powerful tool for reduction of multiloop integrals". In: *J. Phys. Conf. Ser.* 523 (2014). Ed. by J. Wang, p. 012059. DOI: 10.1088/1742-6596/523/1/012059. arXiv: 1310.1145 [hep-ph].

[79] K. G. Chetyrkin and F. V. Tkachov. “Integration by Parts: The Algorithm to Calculate beta Functions in 4 Loops”. In: *Nucl. Phys. B* 192 (1981), pp. 159–204. DOI: 10.1016/0550-3213(81)90199-1.

[80] A. V. Smirnov and A. V. Petukhov. “The Number of Master Integrals is Finite”. In: *Lett. Math. Phys.* 97 (2011), pp. 37–44. DOI: 10.1007/s11005-010-0450-0. arXiv: 1004.4199 [hep-th].

[81] T. Gehrmann and E. Remiddi. “Differential equations for two loop four point functions”. In: *Nucl. Phys. B* 580 (2000), pp. 485–518. DOI: 10.1016/S0550-3213(00)00223-6. arXiv: hep-ph/9912329.

[82] R. N. Lee. “Group structure of the integration-by-part identities and its application to the reduction of multiloop integrals”. In: *JHEP* 07 (2008), p. 031. DOI: 10.1088/1126-6708/2008/07/031. arXiv: 0804.3008 [hep-ph].

[83] E. K. Leinartas. “Factorization of rational functions of several variables into partial fractions”. In: *Soviet Math. (Iz. VUZ)* 22 (10 1978), pp. 35–38. URL: http://mi.mathnet.ru/eng/ivm5842.

[84] A. Raichev. “Leinartas’s partial fraction decomposition”. In: (June 2012). arXiv: 1206.4740 [math.AC].

[85] M. Heller and A. von Manteuffel. “MultivariateApart: Generalized Partial Fractions”. In: (Jan. 2021). arXiv: 2101.08283 [cs.SC].

[86] S. Laporta. “High precision calculation of multiloop Feynman integrals by difference equations”. In: *Int. J. Mod. Phys. A* 15 (2000), pp. 5087–5159. DOI: 10.1016/S0217-751X(00)00215-7. arXiv: hep-ph/0102033.

[87] A. V. Smirnov and F. S. Chuharev. “FIRE6: Feynman Integral REduction with Modular Arithmetic”. In: *Comput. Phys. Commun.* 247Â (2020), p. 106877. DOI: 10.1016/j.cpc.2019.106877. arXiv: 1901.07808 [hep-ph].

[88] J. Klappert et al. “Integral reduction with Kira 2.0 and finite field methods”. In: *Comput. Phys. Commun.* 266 (2021), p. 108024. DOI: 10.1016/j.cpc.2021.108024. arXiv: 2008.06494 [hep-ph].

[89] P. Maierhöfer, J. Usovitsch, and P. Uwer. “Kira—A Feynman integral reduction program”. In: *Comput. Phys. Commun.* 230 (2018), pp. 99–112. DOI: 10.1016/j.cpc.2018.04.012. arXiv: 1705.05610 [hep-ph].

[90] B. Ruijl, T. Ueda, and J. A. M. Vermaseren. “Forcer, a FORM program for the parametric reduction of four-loop massless propagator diagrams”. In: *Comput. Phys. Commun.* 253 (2020), p. 107198. DOI: 10.1016/j.cpc.2020.107198. arXiv: 1704.06650 [hep-ph].

[91] J. Klappert, S. Y. Klein, and F. Lange. “Interpolation of dense and sparse rational functions and other improvements in FireFly”. In: *Comput. Phys. Commun.* 264 (2021), p. 107968. DOI: 10.1016/j.cpc.2021.107968. arXiv: 2004.01463 [cs.MS].

[92] T. Peraro. “FiniteFlow: multivariate functional reconstruction using finite fields and dataflow graphs”. In: *JHEP* 07 (2019), p. 031. DOI: 10.1007/JHEP07(2019)031. arXiv: 1905.08019 [hep-ph].

[93] A. V. Smirnov and V. A. Smirnov. “How to choose master integrals”. In: *Nucl. Phys. B* 960 (2020), p. 115213. DOI: 10.1016/j.nuclphysb.2020.115213. arXiv: 2002.08042 [hep-ph].

[94] J. Usovitsch. “Factorization of denominators in integration-by-parts reductions”. In: (Feb. 2020). arXiv: 2002.08173 [hep-ph].

[95] J. Gluza, K. Kajda, and D. A. Kosower. "Towards a Basis for Planar Two-Loop Integrals". In: *Phys. Rev. D* 83 (2011), p. 045012. DOI: 10.1103/PhysRevD.83.045012. arXiv: 1009.0472 [hep-th].

[96] R. M. Schabinger. "A New Algorithm For The Generation Of Unitarity-Compatible Integration By Parts Relations". In: *JHEP* 01 (2012), p. 077. DOI: 10.1007/JHEP01(2012)077. arXiv: 1111.4220 [hep-ph].

[97] H. Ita. "Two-loop Integrand Decomposition into Master Integrals and Surface Terms". In: *Phys. Rev. D* 94.11 (2016), p. 116015. DOI: 10.1103/PhysRevD.94.116015. arXiv: 1510.05626 [hep-th].

[98] J. Böhm et al. "Complete sets of logarithmic vector fields for integration-by-parts identities of Feynman integrals". In: *Phys. Rev. D* 98.2 (2018), p. 025023. DOI: 10.1103/PhysRevD.98.025023. arXiv: 1712.09737 [hep-th].

[99] X. Guan, X. Liu, and Y.-Q. Ma. "Complete reduction of integrals in two-loop five-light-parton scattering amplitudes". In: *Chin. Phys. C* 44.9 (2020), p. 093106. DOI: 10.1088/1674-1137/44/9/093106. arXiv: 1912.09294 [hep-ph].

[100] C. Anastasiou and K. Melnikov. "Higgs boson production at hadron colliders in NNLO QCD". In: *Nucl. Phys. B* 646 (2002), pp. 220–256. DOI: 10.1016/S0550-3213(02)00837-4. arXiv: hep-ph/0207004.

[101] P. Nogueira. "Automatic Feynman graph generation". In: *J. Comput. Phys.* 105 (1993), pp. 279–289. DOI: 10.1006/jcph.1993.1074.

[102] C. W. Bauer, A. Frink, and R. Kreckel. "Introduction to the GiNaC framework for symbolic computation within the C++ programming language". In: *J. Symb. Comput.* 33 (2002), pp. 1–12. DOI: 10.1006/jsco.2001.0494. arXiv: cs/0004015.

[103] O. V. Tarasov. "Connection between Feynman integrals having different values of the space-time dimension". In: *Phys. Rev. D* 54 (1996), pp. 6479–6490. DOI: 10.1103/PhysRevD.54.6479. arXiv: hep-th/9606018.

[104] R. N. Lee. "Calculating multiloop integrals using dimensional recurrence relation and D-analyticity". In: *Nucl. Phys. B Proc. Suppl.* 205-206 (2010). Ed. by J. Blümlein, S.-O. Moch, and T. Riemann, pp. 135–140. DOI: 10.1016/j.nuclphysbps.2010.08.032. arXiv: 1007.2256 [hep-ph].

[105] R. N. Lee and K. T. Mingulov. "DREAM, a program for arbitrary-precision computation of dimensional recurrence relations solutions, and its applications". In: (Dec. 2017). arXiv: 1712.05173 [hep-ph].

[106] R. N. Lee and K. T. Mingulov. "Meromorphic solutions of recurrence relations and DRA method for multicomponent master integrals". In: *JHEP* 04 (2018), p. 061. DOI: 10.1007/JHEP04(2018)061. arXiv: 1712.05166 [hep-ph].

[107] R. N. Lee. "Space-time dimensionality D as complex variable: Calculating loop integrals using dimensional recurrence relation and analytical properties with respect to D". In: *Nucl. Phys. B* 830 (2010), pp. 474–492. DOI: 10.1016/j.nuclphysb.2009.12.025. arXiv: 0911.0252 [hep-ph].

[108] Blümlein, J. and Broadhurst, D. J. and Vermaseren, J. A. M. "The Multiple Zeta Value Data Mine". In: *Comput. Phys. Commun.* 181 (2010), pp. 582–625. DOI: 10.1016/j.cpc.2009.11.007. arXiv: 0907.2557 [math-ph].

[109] H. Ferguson, D. Bailey, and S. Arno. "Analysis of PSLQ, an integer relation finding algorithm". In: *Math. Comp.* 68 (1999), pp. 351–396. ISSN: 1088-6842. DOI: 10.1090/S0025-5718-99-00995-3.

[110] A. V. Smirnov. "FIESTA4: Optimized Feynman integral calculations with GPU support". In: *Comput. Phys. Commun.* 204 (2016), pp. 189–199. DOI: 10.1016/j.cpc.2016.03.013. arXiv: 1511.03614 [hep-ph].

[111] S. Borowka et al. "pySecDec: a toolbox for the numerical evaluation of multi-scale integrals". In: *Comput. Phys. Commun.* 222 (2018), pp. 313–326. DOI: 10.1016/j.cpc.2017.09.015. arXiv: 1703.09692 [hep-ph].

[112] A. V. Kotikov. "Differential equation method: The Calculation of N-point Feynman diagrams". In: *Phys. Lett. B* 267 (1991). [Erratum: Phys.Lett.B 295, 409–409 (1992)], pp. 123–127. DOI: 10.1016/0370-2693(91)90536-Y.

[113] A. V. Kotikov. "Differential equations method: New technique for massive Feynman diagrams calculation". In: *Phys. Lett. B* 254 (1991), pp. 158–164. DOI: 10.1016/0370-2693(91)90413-K.

[114] E. Remiddi. "Differential equations for Feynman graph amplitudes". In: *Nuovo Cim. A* 110 (1997), pp. 1435–1452. arXiv: hep-th/9711188.

[115] V. A. Smirnov. *Applied Asymptotic Expansions in Momenta and Masses*. Springer Tracts Mod. Phys. 2002, p. 262. DOI: 10.1007/3-540-44574-9.

[116] T. Gehrmann and E. Remiddi. "Two loop master integrals for $\gamma^* \to 3$ jets: The Planar topologies". In: *Nucl. Phys. B* 601 (2001), pp. 248–286. DOI: 10.1016/S0550-3213(01)00057-8. arXiv: hep-ph/0008287.

[117] J. Moser. "The order of a singularity in Fuchs' theory". In: *Mathematische Zeitschrift* 72 (1959), pp. 379–398. DOI: 10.1007/BF01162962.

[118] M. A. Barkatou and E. Pflügel. "Computing Super-Irreducible Forms of Systems of Linear Differential Equations via Moser-Reduction: A New Approach". In: *Proceedings of the 2007 International Symposium on Symbolic and Algebraic Computation*. ISSAC '07. 2007, pp. 1–8. ISBN: 9781595937438. DOI: 10.1145/1277548.1277550. URL: https://doi.org/10.1145/1277548.1277550.

[119] M. A. Barkatou and E. Pflügel. "On the Moser- and super-reduction algorithms of systems of linear differential equations and their complexity". In: *Journal of Symbolic Computation* 44.8 (2009), pp. 1017–1036. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2009.01.002. URL: https://www.sciencedirect.com/science/article/pii/S0747717109000388.

[120] A. Blondel et al. "Standard model theory for the FCC-ee Tera-Z stage". In: *Mini Workshop on Precision EW and QCD Calculations for the FCC Studies : Methods and Techniques*. Vol. 3/2019. CERN Yellow Reports: Monographs. Geneva: CERN, Sept. 2018. DOI: 10.23731/CYRM-2019-003. arXiv: 1809.01830 [hep-ph].

[121] R. N. Lee and A. A. Pomeransky. "Normalized Fuchsian form on Riemann sphere and differential equations for multiloop integrals". In: (July 2017). arXiv: 1707.07856 [hep-th].

[122] T. Gehrmann and E. Remiddi. "Two loop master integrals for $\gamma^* \to 3$ jets: The Nonplanar topologies". In: *Nucl. Phys. B* 601 (2001), pp. 287–317. DOI: 10.1016/S0550-3213(01)00074-8. arXiv: hep-ph/0101124.

[123] E. Panzer. “Algorithms for the symbolic integration of hyperlogarithms with applications to Feynman integrals”. In: *Comput. Phys. Commun.* 188 (2015), pp. 148–166. DOI: 10.1016/j.cpc.2014.10.019. arXiv: 1403.3385 [hep-th].

[124] A. B. Goncharov. “Multiple polylogarithms, cyclotomy and modular complexes”. In: *Math. Res. Lett.* 5 (1998), pp. 497–516. DOI: 10.4310/MRL.1998.v5.n4.a7. arXiv: 1105.2076 [math.AG].

[125] C. Duhr and F. Dulat. “PolyLogTools — polylogs for the masses”. In: *JHEP* 08 (2019), p. 135. DOI: 10.1007/JHEP08(2019)135. arXiv: 1904.07279 [hep-th].

[126] J. Vollinga and S. Weinzierl. “Numerical evaluation of multiple polylogarithms”. In: *Comput. Phys. Commun.* 167 (2005), p. 177. DOI: 10.1016/j.cpc.2004.12.009. arXiv: hep-ph/0410259.

[127] A. B. Goncharov. “Multiple ζ-Values, Galois Groups, and Geometry of Modular Varieties”. In: *European Congress of Mathematics*. Birkhäuser Basel, 2001, pp. 361–392. ISBN: 978-3-0348-8268-2. URL: https://link.springer.com/chapter/10.1007/978-3-0348-8268-2_21.

[128] L. Lewin. *Polylogarithms and associated functions*. North-Holland, New York: Addison-Wesley, 1981, p. 376. ISBN: 0-444-00550-1.

[129] E. Remiddi and J. A. M. Vermaseren. “Harmonic polylogarithms”. In: *Int. J. Mod. Phys. A* 15 (2000), pp. 725–754. DOI: 10.1142/S0217751X00000367. arXiv: hep-ph/9905237.

[130] D. Maître. “HPL, a Mathematica implementation of the harmonic polylogarithms”. In: *Comput. Phys. Commun.* 174 (2006), pp. 222–240. DOI: 10.1016/j.cpc.2005.10.008. arXiv: hep-ph/0507152.

[131] D. Maître. “Extension of HPL to complex arguments”. In: *Comput. Phys. Commun.* 183 (2012), p. 846. DOI: 10.1016/j.cpc.2011.11.015. arXiv: hep-ph/0703052.

[132] Kölbig, K. S. and Mignaco, J. A. and Remiddi, E. “On Nielsen’s generalized polylogarithms and their numerical calculation”. In: *BIT Num. Math.* 10 (1970), pp. 38–73. DOI: 10.1007/BF01940890.

[133] F. Brown. “The Massless higher-loop two-point function”. In: *Commun. Math. Phys.* 287 (2009), pp. 925–958. DOI: 10.1007/s00220-009-0740-5. arXiv: 0804.1660 [math.AG].

[134] W. Stein et al. *SageMath*. URL: http://www.sagemath.org.

[135] W. Schelter et al. *Maxima, a Computer Algebra System.* URL: http://maxima.sourceforge.net.

[136] A. Barkatou. “A Rational Version of Moser’s Algorithm”. In: *Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation*. ISSAC ’95. 1995, pp. 297–302. ISBN: 0897916999. DOI: 10.1145/220346.220385. URL: https://doi.org/10.1145/220346.220385.

[137] R. J. Fateman. “Rational Function Computing with Poles and Residues”. In: (2013). URL: https://people.eecs.berkeley.edu/~fateman/papers/polesnew.pdf.

[138] R. Tarjan. “Depth-First Search and Linear Graph Algorithms”. In: *SIAM J. Comput.* 1.2 (1972), pp. 146–160. DOI: 10.1137/0201010.

[139] A. Gehrmann-De Ridder, T. Gehrmann, and G. Heinrich. “Four particle phase space integrals in massless QCD”. In: *Nucl. Phys. B* 682 (2004), pp. 265–288. DOI: 10.1016/j.nuclphysb.2004.01.023. arXiv: hep-ph/0311276.

[140] E. Byckling and K. Kajantie. *Particle Kinematics: (Chapters I-VI, X)*. Jyvaskyla, Finland: University of Jyvaskyla, 1971. URL: `https://lss.fnal.gov/archive/other/Part.Kinmatics-(ChaptersI-VI,X).pdf`.

[141] R. Kleiss, W. J. Stirling, and S. D. Ellis. "A New Monte Carlo Treatment of Multiparticle Phase Space at High-energies". In: *Comput. Phys. Commun.* 40 (1986), p. 359. DOI: `10.1016/0010-4655(86)90119-0`.

[142] M. Czakon et al. "The $(Q_7, Q_{1,2})$ contribution to $\overline{B} \to X_s\gamma$ at $\mathcal{O}\left(\alpha_s^2\right)$". In: *JHEP* 04 (2015), p. 168. DOI: `10.1007/JHEP04(2015)168`. arXiv: `1503.01791 [hep-ph]`.

[143] P. A. Baikov and K. G. Chetyrkin. "Four Loop Massless Propagators: An Algebraic Evaluation of All Master Integrals". In: *Nucl. Phys. B* 837 (2010), pp. 186–220. DOI: `10.1016/j.nuclphysb.2010.05.004`. arXiv: `1004.1153 [hep-ph]`.

[144] R. N. Lee, A. V. Smirnov, and V. A. Smirnov. "Master Integrals for Four-Loop Massless Propagators up to Transcendentality Weight Twelve". In: *Nucl. Phys. B* 856 (2012), pp. 95–110. DOI: `10.1016/j.nuclphysb.2011.11.005`. arXiv: `1108.0732 [hep-th]`.

[145] O. Gituliar, V. Magerya, and A. Pikelner. "Five-Particle Phase-Space Integrals in QCD". In: *PoS* LL2018 (2018), p. 087. DOI: `10.22323/1.303.0087`. arXiv: `1808.05109 [hep-ph]`.

[146] G. P. Lepage. "A New Algorithm for Adaptive Multidimensional Integration". In: *J. Comput. Phys.* 27 (1978), p. 192. DOI: `10.1016/0021-9991(78)90004-9`.

[147] T. Hahn. "CUBA: A Library for multidimensional numerical integration". In: *Comput. Phys. Commun.* 168 (2005), pp. 78–95. DOI: `10.1016/j.cpc.2005.01.010`. arXiv: `hep-ph/0404043`.

[148] T. Gehrmann, T. Huber, and D. Maître. "Two-loop quark and gluon form-factors in dimensional regularisation". In: *Phys. Lett. B* 622 (2005), pp. 295–302. DOI: `10.1016/j.physletb.2005.07.019`. arXiv: `hep-ph/0507061`.

[149] R. E. Cutkosky. "Singularities and discontinuities of Feynman amplitudes". In: *J. Math. Phys.* 1 (1960), pp. 429–433. DOI: `10.1063/1.1703676`.

[150] G. 't Hooft and M. J. G. Veltman. "DIAGRAMMAR". In: *NATO Sci. Ser. B* 4 (1974), pp. 177–322. DOI: `10.1007/978-1-4684-2826-1_5`.

[151] M. E. Peskin and D. V. Schroeder. *An Introduction to quantum field theory*. Reading, USA: Addison-Wesley, 1995. ISBN: 978-0-201-50397-5. URL: `http://www.slac.stanford.edu/~mpeskin/QFT.html`.

[152] J. C. Romao and J. P. Silva. "A resource for signs and Feynman diagrams of the Standard Model". In: *Int. J. Mod. Phys. A* 27 (2012), p. 1230025. DOI: `10.1142/S0217751X12300256`. arXiv: `1209.6213 [hep-ph]`.

[153] N. D. Christensen and C. Duhr. "FeynRules - Feynman rules made easy". In: *Comput. Phys. Commun.* 180 (2009), pp. 1614–1641. DOI: `10.1016/j.cpc.2009.02.018`. arXiv: `0806.4194 [hep-ph]`.

[154] A. I. Davydychev, P. Osland, and O. V. Tarasov. "Two loop three gluon vertex in zero momentum limit". In: *Phys. Rev. D* 58 (1998), p. 036007. DOI: `10.1103/PhysRevD.58.036007`. arXiv: `hep-ph/9801380`.

[155] B. Ruijl et al. "Four-loop QCD propagators and vertices with one vanishing external momentum". In: *JHEP* 06 (2017), p. 040. DOI: `10.1007/JHEP06(2017)040`. arXiv: `1703.08532 [hep-ph]`.

[156] S. A. Larin, F. V. Tkachov, and J. A. M. Vermaseren. “The FORM version of MINCER”. In: (Sept. 1991). URL: `https://www.nikhef.nl/~form/maindir/packages/mincer/`.

A. Spherical coordinate system in *n* dimensions

A spherical coordinate system in n Euclidean dimensions parameterizes a point $\vec{p}$ with a radius r, and angles $\theta_1, \dots, \theta_{n-1}$, such that the Cartesian coordinates corresponding to $\{r, \theta\}$ are

$$(\vec{p})_i = \begin{cases} r\cos\theta_1, & \text{if } i = 1, \\ r\sin\theta_1\cos\theta_2, & \text{if } i = 2, \\ r\sin\theta_1\sin\theta_2\cos\theta_3, & \text{if } i = 3, \\ \dots \\ r\sin\theta_1\sin\theta_2\cdots\cos\theta_{n-1}, & \text{if } i = n-1, \\ r\sin\theta_1\sin\theta_2\cdots\sin\theta_{n-1}, & \text{if } i = n, \end{cases} \tag{A.0.1}$$

where $r \geq 0$, and θ_i are all assumed to lie in the range of $[0; \pi]$, except for the last of them, θ_{n-1}, which lies in $[0; 2\pi)$.

The volume element in spherical coordinates is

$$\mathrm{d}^n\vec{p} = \left|\det\frac{\partial\{\vec{p}\}}{\partial\{r,\theta\}}\right|\mathrm{d}\{r,\theta\} = r^{n-1}\mathrm{d}r\underbrace{\prod_{i=1}^{n-1}\sin^{n-1-i}\theta_i\,\mathrm{d}\theta_i}_{=\mathrm{d}\Omega_{n-1}}. \tag{A.0.2}$$

An integral over the angular component of this volume element is the total solid angle in n dimensions, or the surface area of a unit $(n-1)$-sphere. We shall denote this quantity as Ω_{n-1},

$$\Omega_k \equiv \int\prod_{i=1}^{k}\sin^{k-i}\theta_i\,\mathrm{d}\theta_i = \frac{2\pi^{\frac{k+1}{2}}}{\Gamma\left(\frac{k+1}{2}\right)}, \tag{A.0.3}$$

where we have used

$$\int_0^\pi \sin^n\theta\,\mathrm{d}\theta = \pi^{\frac{1}{2}}\frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{n+2}{2}\right)}. \tag{A.0.4}$$

B. Feynman rules

Although this thesis deals with scalar integrals, we also have in mind a physical model given by the standard QCD Lagrangian with N_c colors and N_f fermion flavors, augmented with the photon field A, and a scalar field H coupled to gluons.

The Lagrangian

For greater clarity (important for both a casual reader and a computer implementation), let us write down this Lagrangian with all indices made explicit:

$$\mathscr{L} = \mathscr{L}_0 + \mathscr{L}_\psi + \mathscr{L}_h + \mathscr{L}_c + \mathscr{L}_\xi, \tag{B.0.1}$$

$$\mathscr{L}_0 \equiv -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} - \frac{1}{4} G^a_{\mu\nu} G^{a,\mu\nu}, \tag{B.0.2}$$

$$\mathscr{L}_\psi \equiv \sum_f \bar{\psi}^s_{f,i}(x)\, i\gamma^{ss'}_\mu\, D^\mu_{f,ij}(x)\, \psi^{s'}_{f,j}(x), \tag{B.0.3}$$

$$\mathscr{L}_h \equiv -\frac{1}{4} g_h H G^a_{\mu\nu} G^{a,\mu\nu}, \tag{B.0.4}$$

$$\mathscr{L}_c \equiv -\bar{c}^a \partial^\mu D^{ab}_\mu c^b, \tag{B.0.5}$$

$$\mathscr{L}_\xi \equiv -\frac{1}{2\xi_A} \left(\partial^\mu A_\mu\right)^2 - \frac{1}{2\xi_g} \left(\partial^\mu G^a_\mu\right)^2, \tag{B.0.6}$$

where

$$F_{\mu\nu}(x) = \partial_\mu A_\nu - \partial_\nu A_\mu, \tag{B.0.7}$$

$$G^a_{\mu\nu}(x) = \partial_\mu G^a_\nu - \partial_\nu G^a_\mu + g_s f^{abc} G^b_\mu G^c_\nu, \tag{B.0.8}$$

$$D^\mu_{f,ij}(x) = \delta_{ij}\partial^\mu + iQ_f g_e A^\mu \delta_{ij} - ig_s G^a_\mu t^a_{ij}, \tag{B.0.9}$$

$$D^{ab}_\mu(x) = \delta^{ab}\partial_\mu - g_s f^{abc} G^c_\mu, \tag{B.0.10}$$

and

- $A^\mu(x)$ is the photon field;
- G^μ_a is the gluon field of color a;
- $\psi^s_{fi}(x)$ is the quark field (Dirac spinors) of flavor f and color i;
- $H(x)$ is a complex scalar field coupled to gluons (effective Higgs);
- $c^a(x)$ is the Fadeev-Popov ghost field of color a;
- γ_μ are the Dirac gamma matrices;
- t^a are the $SU(N)$ generators;
- g_e, g_s, and g_h are the coupling constants;

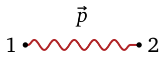
- Q_f is the fractional electric charge of the quark flavor f;
- ξ_A and ξ_g are the photon and the gluon gauge parameters;

and the indices are:

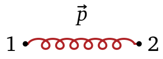
- Lorentz indices $\mu, \nu = 1 \ldots d$, where $d = 4$;
- fundamental color indices $i, j = 1 \ldots N_c$;
- adjoint color indices $a, b, c = 1 \ldots N_c^2 - 1$;
- quark flavor indices $f = 1 \ldots N_f$;
- spinor indices s, the precise dimension of which should not enter the results.

Feynman rules

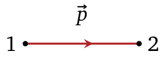
The Feynman rules that correspond to the Lagrangian of eq. (B.0.1) are as follows. For propagators:

Photon: 

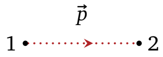
$$= -i\left(\frac{g_{\mu_1\mu_2}}{p^2} - (1-\xi_A)\frac{p_{\mu_1}p_{\mu_2}}{p^4}\right); \tag{B.0.11}$$

Gluon: 

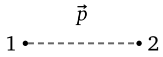
$$= -i\delta_{a_1a_2}\left(\frac{g_{\mu_1\mu_2}}{p^2} - (1-\xi_G)\frac{p_{\mu_1}p_{\mu_2}}{p^4}\right); \tag{B.0.12}$$

Quark: 

$$= i\delta_{f_1f_2}\delta_{i_1i_2}\frac{(p_\nu\gamma^\nu)_{s_2s_1}}{p^2}; \tag{B.0.13}$$

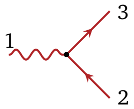
Ghost: 

$$= \frac{i\delta_{a_1a_2}}{p^2}; \tag{B.0.14}$$

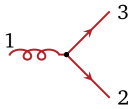
Higgs: 

$$= \frac{i}{p^2}. \tag{B.0.15}$$

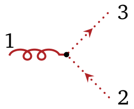
For vertices:



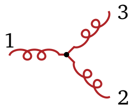
$$= ig_e\delta_{f_2f_3}\delta_{i_2i_3}Q_{f_2}; \tag{B.0.16}$$



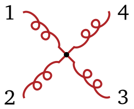
$$= ig_s\delta_{f_2f_3}T^{a_1}_{i_3i_2}(\gamma^{\mu_1})_{s_3s_2}; \tag{B.0.17}$$



$$= g_s f^{a_1a_2a_3} p_3^{\mu_1}; \tag{B.0.18}$$



$$= g_s f^{a_1a_2a_3}\left(g^{\mu_1\mu_2}(p_1-p_2)^{\mu_3} + \{123 \to 231\} + \{123 \to 312\}\right); \tag{B.0.19}$$



$$= -ig_s^2\left(f^{ba_1a_2}f^{ba_3a_4}\left(g^{\mu_1\mu_3}g^{\mu_2\mu_4} - g^{\mu_1\mu_4}g^{\mu_2\mu_3}\right) + \{1234 \to 1342\} + \{1234 \to 1423\}\right). \tag{B.0.20}$$

The $Hggg$ and $Hgggg$ vertices are the same as ggg and $gggg$, but with an additional g_h factor.

These rules are consistent with [151]. We recommend consulting [152] for an overview of different conventions used in the literature. In that paper's notation, the definitions adopted here correspond to the choices of $\eta_G = +1$, $\eta = -1$, and $\eta_s = -1$. The other popular convention of $\eta_s = +1$ differs only in the sign of g_s in eq. (B.0.8), and as far as the Feynman rules are concerned it is identical to ours, but with the overall sign inverted in the quark-gluon and tri-gluon vertices (that is, all vertices proportional to g_s). The values of the diagrams calculated in both conventions are of course identical, because g_s always enters as g_s^2, and its sign does not matter.

The rules presented above have been derived from the Lagrangian of eq. (B.0.1) using FEYNRULES [153], which is a MATHEMATICA package for automatic Feynman rule derivation.[1]

As an essential cross-check we have used these rules to re-calculated the gluon, ghost, and quarks propagators up to two loops: the results match the values reported in e.g. [154, 155], as well as the ones extracted from the sources of MINCER [156].

[1] Aside from the propagators, which FEYNRULES is not able to derive.

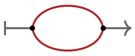
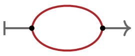
C. The resulting integral tables

Along with this thesis we provide the values of the calculated integrals in machine-readable MATHEMATICA format. These files are available from the author (to be published separately). They consist of the following:

inc4l.w12.m

The values of master integrals for fully-inclusive cuts of 4-loop propagators from Chapter 9. All the 5-, 4-, 3-, and 2-particle cuts are provided here (the ones listed in Table 9.3.2, Table 9.3.3, Table 9.3.4, and Table 9.3.5 correspondingly); for convenience the fully virtual integrals from Table 9.3.1 are also given.

The integrals are defined by eq. (9.0.1). The values here are their normalized versions:

$J = I / \Bigg($  $^{L} \Bigg($  $^{*}\Bigg)^{R} \mathrm{PS}_n \Bigg),$ (C.0.1)

where L is the number of loops to the "left" of the cut, R is the number of cuts to the "right" of it, and n is the number of cut lines. The value of is given by eq. (3.2.10); PS_N is given by eq. (9.1.14).

The file contains a list of items, each defining one integral. Each item is a list of two elements: the product of the denominators defining the integral, and its value as a series in `ep` (which stands for ε). The denominators come in three forms:

- `den[`p`]` , corresponding to a regular denominator, $\frac{1}{p^2+i0}$;
- `cut[`p`]` , corresponding to a cut denominator (part of the phase-space integration), $2\pi\delta^+\big(p^2\big)$;
- `xcut[`p`]` , corresponding to an x-tagged cut denominator, $\delta\big(p^2-(1-x)q^2\big)$.

The values of the integrals are in the standard MATHEMATICA format, with the expansion order indicated via `Ord[ep,`n`]`$\equiv \mathcal{O}(\varepsilon^n)$.

Additionally, `Mzv[`$n,\ldots$`]` stands for multiple zeta values, $\zeta_{n,\ldots}$, as defined by eq. (7.2.12).

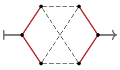
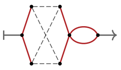
si4l.w8.m

The values of master integrals for semi-inclusive cuts of 4-loop propagators from Chapter 10. The format is the same as with `inc4l.w12.m`. The families these integrals belong to (but not the integrals themselves) are listed in Appendix D.

si4l.factorized.w8.m

The values of master integrals for semi-inclusive cuts of 4-loop propagators from Chapter 10, with endpoint singularities factorized as described in Section 10.3.4. The format is the same as with `inc4l.w12.m`, except the values are sums of expressions multiplied by `Prefactor[]` expressions that wrap the irrational prefactors.

The tables for 2-loop 1 → 3 integrals from Section 6.4 (the ones listed in Table 6.4.2, Table 6.4.3, and Table 6.4.4) are available in the auxiliary files from [35].

The values for cuts of 2- and 3-loop propagators are not included separately, but can be extracted from the provided values: because of the normalization we have used, the normalized value for e.g.  is exactly the same as that of , so all the values for cuts of 2- and 3-loop propagators are contained in the provided tables too.

D. Integral families for semi-inclusive cuts at four loops

We have identified the total of 256 integral families needed to cover semi-inclusive decay at 4 loops. We list these families here: there are 18 2-particle semi-inclusive cut families (Table D.0.1), 34 3-cut families (Table D.0.2), 96 4-cut families (Table D.0.3), and 108 5-cut particles (Table D.0.4).

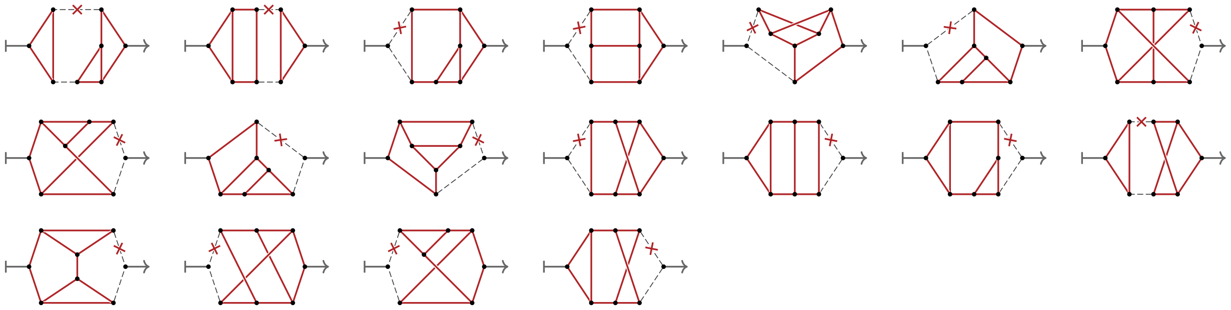


Table D.0.1.: 2-particle-cut semi-inclusive families at 4 loops.

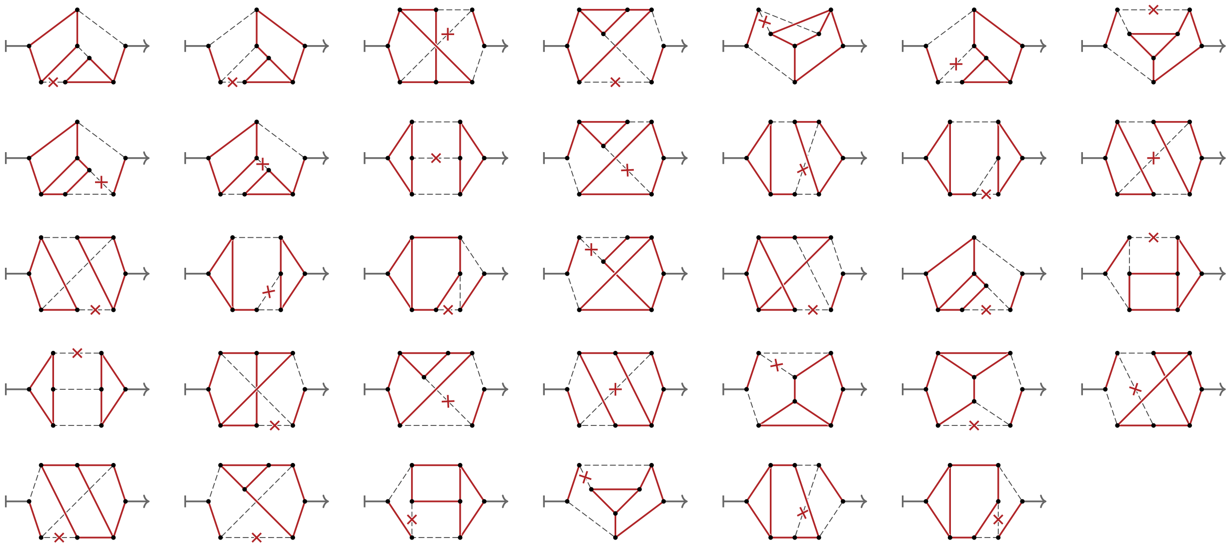


Table D.0.2.: 3-particle-cut semi-inclusive families at 4 loops.

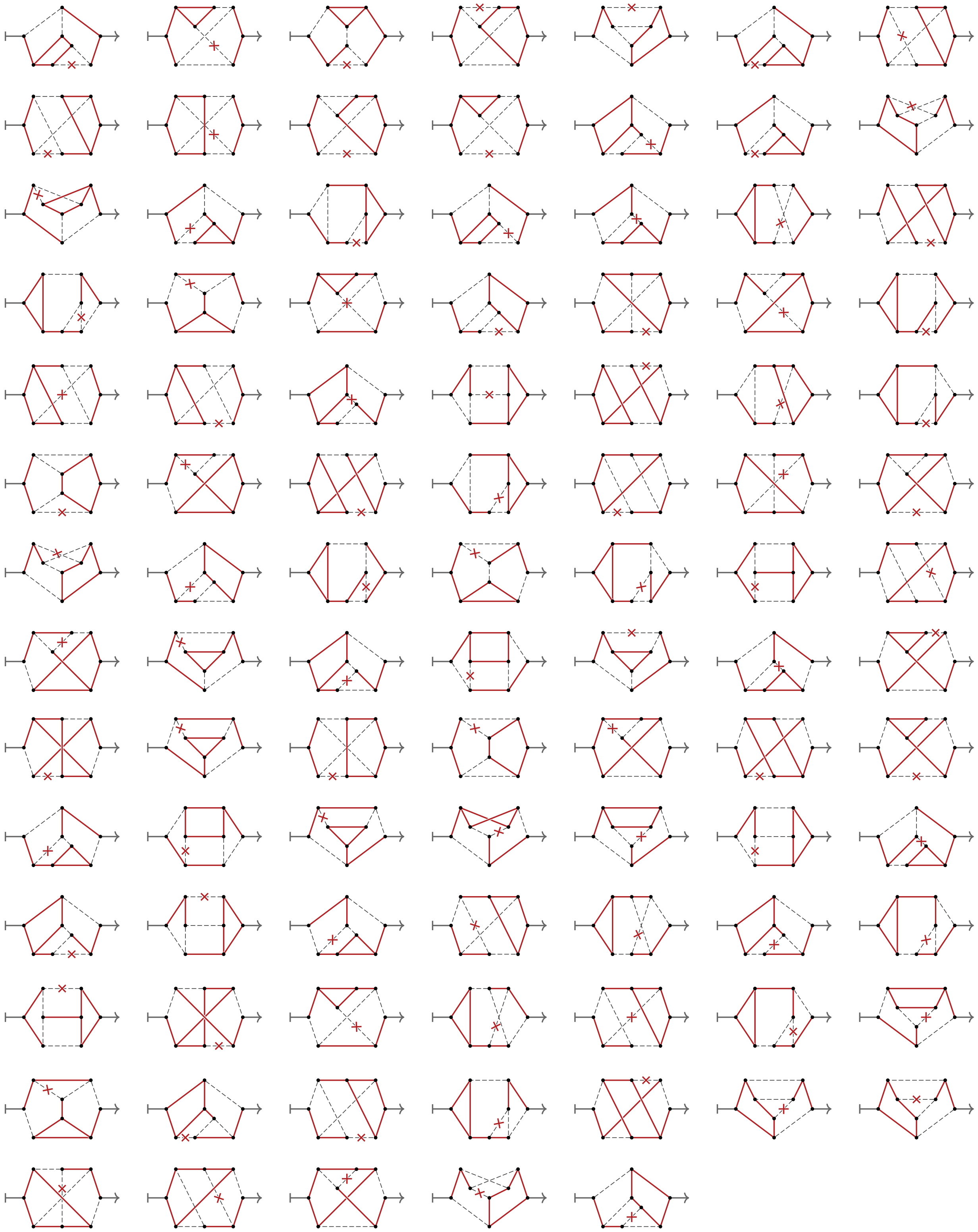


Table D.0.3.: 4-particle-cut semi-inclusive families at 4 loops.

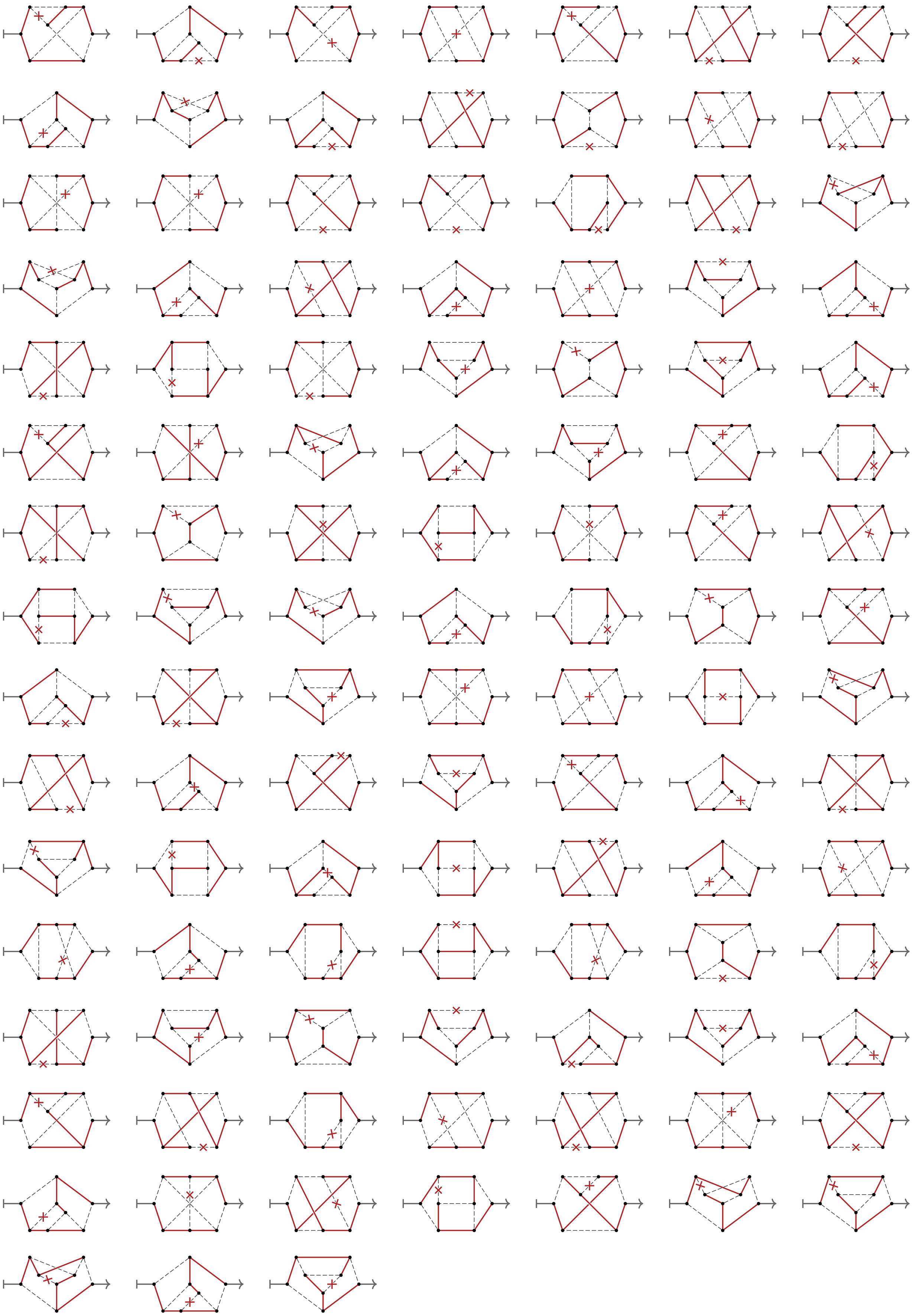


Table D.0.4.: 5-particle-cut semi-inclusive families at 4 loops.