UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia Galileo Galilei

UNIVERSITÀ DEGLI STUDI DI PADOVA

Head office: Università degli Studi di Padova
Department: Dipartimento di Fisica e Astronomia "Galileo Galilei"

Ph.D. course in Physics
Cycle XXXVI

Evaluating Feynman Integrals Using $\mathcal{D}$-modules and Tropical Geometry

Coordinator:

Prof. Giulio Monaco

Supervisor:

Prof. Pierpaolo Mastrolia

Ph.D. candidate:

Henrik Jessen Munch

Academic Year
2022/2023

Abstract

Feynman integrals play a central role in the modern scattering amplitudes research program. Advancing our methods for evaluating Feynman integrals will, therefore, strengthen our ability to compare theoretical predictions with data from particle accelerators such as the Large Hadron Collider. Motivated by this, the present manuscript purports to study mathematical concepts related to Feynman integrals. In particular, we present both numerical and analytical algorithms for the evaluation of Feynman integrals.

The content is divided into three parts.

Part I focuses on the method of DEQs for evaluating Feynman integrals. An otherwise daunting integral expression is thereby traded for the comparatively simpler task of solving a system of DEQs. We use this technique to evaluate a family of two-loop Feynman integrals of relevance for dark matter detection.

Part II situates the study of DEQs for Feynman integrals within the framework of *$\mathcal{D}$-modules*, a natural language for studying PDEs algebraically. Special emphasis is put on a particular $\mathcal{D}$-module called the *GKZ system*, a set of higher-order PDEs that annihilate a generalized version of a Feynman integral. In the course of matching the generalized integral to a Feynman integral proper, we discover an algorithm for evaluating the latter in terms of logarithmic series.

Part III develops a numerical integration algorithm. It combines Monte Carlo sampling with *tropical geometry*, a particular offspring of algebraic geometry that studies "piecewise-linear" polynomials. Feynman's $i\varepsilon$-prescription is incorporated into the algorithm via contour deformation. We present a new open-source program named FEYNTROP that implements this algorithm, and use it to numerically evaluate Feynman integrals between 1-5 loops and 0-5 legs in physical regions of phase space.

Author's Declaration

I hereby declare that this thesis is solely authored by me. The work presented herein was carried out by the author during his Ph.D. studies at Università degli Studi di Padova, Dipartimento di Fisica e Astronomia "Galileo Galilei". The following manuscripts were co-authored by me during my Ph.D.

Articles

1. Raghuveer Garani, Federico Gasparotto, Pierpaolo Mastrolia, Henrik J. Munch, Sergio Palomares-Ruiz and Amedeo Primo, *Two-photon exchange in leptophilic dark matter scenarios*, JHEP 12 (2021), DOI: 10.1007/JHEP12(2021)212, arXiv: 2105.12116 [hep-ph].

2. Vsevolod Chestnov, Federico Gasparotto, Manoj K. Mandal, Pierpaolo Mastrolia, Saiei J. Matsubara-Heo, Henrik J. Munch and Nobuki Takayama, *Macaulay Matrix for Feynman Integrals: Linear Relations and Intersection Numbers*, JHEP 09 (2022), DOI: 10.1007/JHEP09(2022)187, arXiv: 2204.12983 [hep-th].

3. Michael Borinsky, Henrik J. Munch and Felix Tellander, *Tropical Feynman integration in the Minkowski regime*, COMPHY 292 (2023), DOI: https://doi.org/10.1016/j.cpc.2023.108874, arXiv: 2302.08955 [hep-ph].

4. Vsevolod Chestnov, Saiei J. Matsubara-Heo, Henrik J. Munch and Nobuki Takayama *Restrictions of Pfaffian Systems for Feynman Integrals*, arXiv: 2305.01585 [hep-th].

Conference proceedings

1. Henrik J. Munch, *Feynman Integral Relations from GKZ Hypergeometric Systems*, PoS, LL2022 (2022), DOI: 10.22323/1.416.0042, arXiV: 2207.09780 [hep-th].

2. E. Bernardini, M. Carli, M. Y. Elkhashab, A. Ferroglia, M. Fiolhais, L. Gabelli, H. J. Munch, D. Krym, P. Mastrolia, G. Ossola, O. Pantano, J. Postiglione, J. S. Poveda Correa, C. Sirignano and F. Soramel, *Using Smartphones to Innovate Laboratories in Introductory Physics Courses*, GIREP 2022: Effective Learning in Physics from Contemporary Physics to Remote Settings, Ljubljana, Slovenia.

Acknowledgements

It takes a village to raise a child.

African proverb

Let me here express my sincerest gratitude to all those who have so wonderfully shaped my Ph.D. journey over the past three years.

I first and foremost thank my supervisor Pierpaolo Mastrolia for passionately introducing me to the subject of scattering amplitudes, and for allowing me to pursue my research goals.

It is not an exaggeration to say that most of what I learnt throughout my Ph.D. came from interactions with Federico Gasparotto and Seva Chestnov. I thoroughly enjoyed collaborating with you, and I cannot thank you enough for sharing your knowledge with me.

Let me give further thanks to all the people associated to the Padova Amplitudes group, with whom I had many stimulating discussions on topics in physics, namely Manoj Kumar Mandal, Luca Mattiazzi, Giulio Crisanti, Giacomo Brunello, Sid Smith, and Hjalte Frellesvig; I thank Hjalte in particular for hosting me at the Niels Bohr Institute during the Amplitudes 2021 conference. I also appreciate all the good times spent with Yousry, Marah, Prajwal and Marco here in Padova.

I feel extremely fortunate to have worked with Felix Tellander and Michael Borinsky, and I am so proud of the work we achieved together. I am grateful for the hospitality shown to us three by Nima Arkani-Hamed, Sebastian Mizera and Aaron Hillman during our visit to the Institute for Advanced Study in Princeton.

Though I was officially enrolled in a physics Ph.D. program, my interests lead me into more mathematical areas of research. In this regard, I had the pleasure of collaborating with and learning from two brilliant mathematicians, Saiei-Jaeyeong Matsubara-Heo and Nobuki Takayama. These interactions had a tremendous impact on my intellectual development, and instilled in me a great appreciation for the beauty of mathematical structures.

Speaking of mathematics, I want to thank everyone with whom I engaged with at the Max Planck Institute for Mathematics in the Sciences in Leipzig. I am especially grateful to Bernd Sturmfels for his hospitality, and for creating such a unique and engaging environment at the institute. I thank my collaborators Saiei and Claudia Fevola for all

the joyous times spent there diving into the mathematics of hyperplane arrangements and more.

I thank Jonah Stalknecht, Robert William Moerman and David Damgaard for co-organizing the Geometry and Scattering Amplitudes Journal Club with me, and I appreciate all the speakers and audience members for sharing their discoveries within the field of scattering amplitudes.

Kæmpe tak til mine drenge Skov, Topic og Jeppe for en sindssyg fed tur rundt i Italien. Det bragte mange gode minder.

Finally, I wish to thank my mother Ljudmila and my partner Laura for their love and support throughout this whole period.

Contents

List of Acronyms

(**CAS**):	Computer algebra system
(**DM**):	Dark matter
(**DR**):	Dimensional regularization
(**FI**):	Feynman integral
(**GFI**):	Generalized Feynman integral
(**GKZ**):	Gel'fand Kapranov Zelevinsky
(**GP**):	Generalized permutahedron
(**GPL**):	Generalized polylogarithm
(**HPL**):	Harmonic polylogarithm
(**MC**):	Monte Carlo
(**ISP**):	Irreducible scalar product
(**LPr**):	Lee-Pomeransky representation
(**MI**):	Master integral
(**SM**):	Standard model
(**QCD**):	Quantum chromodynamics
(**QED**):	Quantum electrodynamics
(**QFT**):	Quantum field theory

Chapter 1

Introduction

Eugene Wigner famously pondered about the *unreasonable effectiveness of mathematics in the natural sciences* [1]. How can it be that esoteric mathematical structures such as group theory, calculus and geometry - often developed for entirely independent reasons - are able to describe the "laws of Nature" so powerfully? A prime example is that of quantum field theory (QFT). This theory combines two pillars of our modern understanding of Nature: quantum mechanics and special relativity. Although some intuition can be gained from describing QFT in plain English, it is only in terms of mathematics that we can properly (insofar as possible) *define* the theory and *do* something with it!

The standard model (SM) is the crown jewel of the modern QFT program. It describes three out of four known forces of Nature: electromagnetism, the weak and the strong nuclear forces. Although it does not incorporate gravity (the last, known force) at the quantum level, the mathematical techniques that were developed to probe the SM are now also being used to study gravity in the weak field limit [2].

Physicists test the SM in the same way Rutherford first investigated the sub-atomic world a century ago: by scattering particles together and measuring their cross sections (roughly speaking the effective range of interaction of the scattered particles). Figure (1.1) compares a host of experimentally determined total cross sections, from ATLAS data, with corresponding theoretical predictions made within the SM. The level of agreement is striking.

A QFT, such as the SM, is typically defined in the abstract language of Lagrangians, i.e. polynomials in quantum fields and derivatives thereof. How can this be connected to concrete, observable quantities such as the aforementioned cross sections? The connection is made through the calculation of *scattering amplitudes*. These are complex-valued functions (depending on data such the masses, spins and momenta of the particles involved in a given scattering experiment) which measure the quantum mechanical probability for turning an initial state into a final state. Given such an amplitude $\mathcal{A}$, one obtains a total

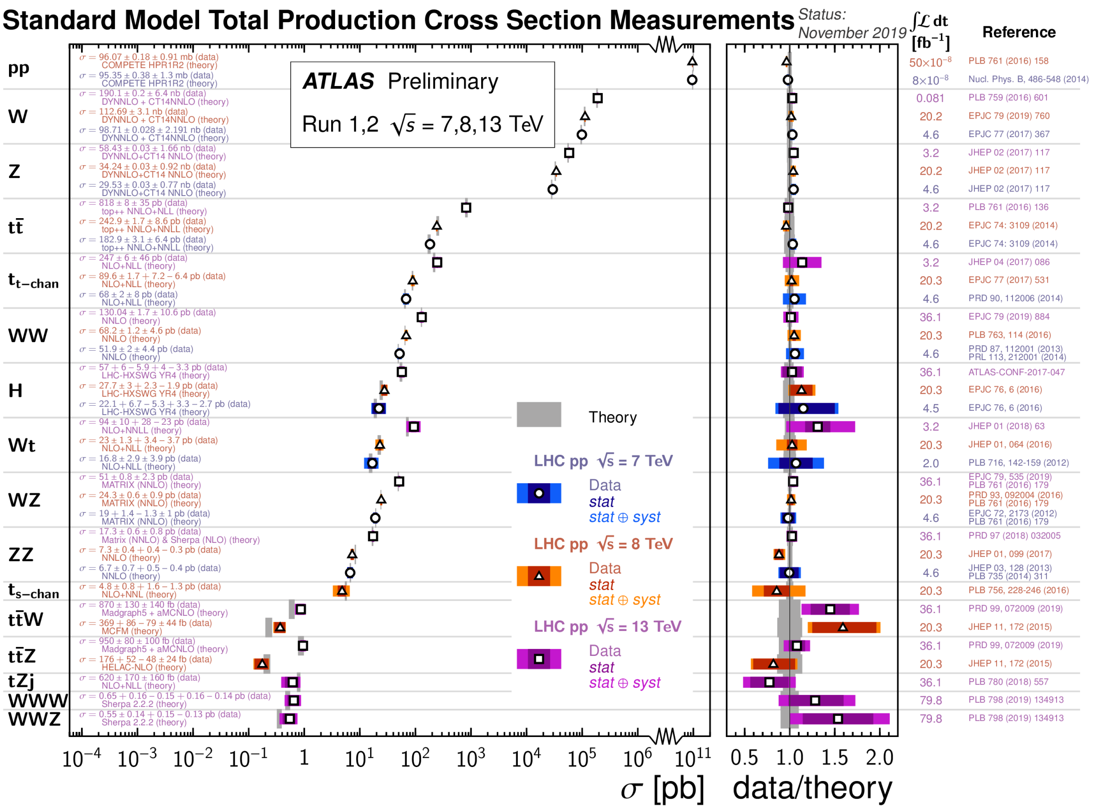
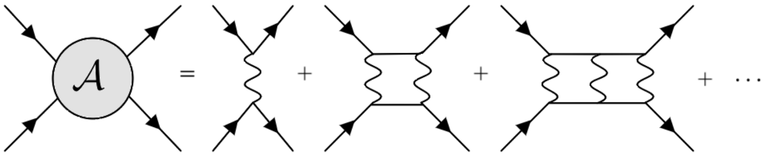


Figure 1.1: Summary plot anno 2019 comparing experimental vs. theoretical values for Standard Model cross sections [4].

cross section by integrating $|\mathcal{A}|^2$ over a relevant region of phase space.

Because the SM Lagrangian contains several small parameters in the form of coupling constants, it is natural to compute scattering amplitudes as *perturbative* expansions in said constants. The numerical accuracy of a given amplitude is then determined by how many orders of this expansion we are able to compute. Ever since the work of Feynman [3], physicists have organized these expansions in terms of *Feynman diagrams*. Schematically, the perturbative expansion of an amplitude $\mathcal{A}$ is written as



(1.1)

The *Feynman rules*, derived from the Lagrangian of a given theory, give instructions for how to calculate each diagram. The leading-order contributions, written as sums of *tree* diagrams, are, arguably, now under control, and so will not be a focus of this thesis. The subleading-order contributions are written as sums of *loop* diagrams. In these cases, the Feynman rules instruct us to compute integrals over the momenta of virtual particles running inside the loops. These *Feynman integrals* (FIs) are the source of nearly all the computational complexity that goes into the calculation of perturbative scattering amplitudes beyond leading order. If we want to probe the SM at increased accuracy in the hopes of detecting signs of physics *beyond* the SM (which we know must exist, due to unexplained phenomena such as neutrino masses, dark matter and the supposedly "unnatural" value of the Higgs mass), it thus behooves us to strengthen our mathematical capabilities for computing FIs. Indeed, following Wigner, we should use the unreasonable effectiveness of mathematics to our benefit!

The context of this thesis

There is a very long road from writing down a Lagrangian in, say, QCD to actually arriving at a cross section at some given order in perturbation theory (this includes writing down all the relevant Feynman diagrams, taking care of tensor structure, renormalizing and regularizing UV and IR divergences respectively, feeding in non-perturbative information via parton distribution functions, choosing proper values for factorization/renormalization scales, and much much more). The modern scattering amplitudes program is, in a word, all about finding potential *simplicity* hiding within this highly complex process in order to lessen the calculational burden [5][1]. More narrowly, for the purpose of this thesis we may ponder the question: *do FIs also exhibit any simple mathematical structure?*

IBPs and DEQs. The first sign of "simplicity" was arguably the discovery of the *linear relations* among FIs called *integration-by-parts* identities (IBPs) [6, 7, 8]. To mention a concrete application of these relations, take the recent 3-loop QCD amplitude computed in [9]. The original expression for the amplitude contains of order $\mathcal{O}(10^5)$ FIs to be evaluated. However, only $\mathcal{O}(300)$ of these integrals are actually linearly independent! In other words, the original $\mathcal{O}(10^5)$ integrals can all be written as linear combinations of $\mathcal{O}(300)$ *master integrals* (MIs), where the coefficients in front of the MIs are rational functions in the kinematic variables and the dimension of spacetime. When such linear relations can be generated systematically and efficiently, it constitutes a significant *simplification* of a given

[1] In fact, it is now quite common in this field to dispense with Lagrangians entirely, as they carry so much physically irrelevant (i.e. off-shell) information!

calculation. There are now many public codes dedicated to this purpose being used in virtually every current amplitude calculation beyond leading order [10, 11, 12, 13, 14, 15]. Indeed, we shall make frequent use of them in this thesis.

Suppose the many FIs in a given calculation have been reduced to a small set of MIs. The question is then how to evaluate the MIs themselves. IBPs once again lend a helping hand, as they can be used to derive *differential equations* for MIs [16, 17, 18, 19]. The second example of "simplicity" that we highlight was discovered by Henn [20]: there is a freedom in choosing which FIs that are promoted to MIs, and if this choice is made judiciously, then their DEQs simplify considerably. Concretely, the simplification stems from the decoupling, or factorization, of a parameter in the DEQs called ϵ, which appears because it is related to the dimension D of spacetime (as mentioned above, D is brought into the game due to IBPs). Such a basis is dubbed to be *canonical*. When the DEQs are brought into this canonical form, they are *much* easier to solve.

$\mathcal{D}$-modules. These two notions, IBPs and DEQs, thus appear to be central to the (analytical, or symbolic) evaluation of FIs. To learn more about the mathematical structure of these integrals, we argue that it is interesting to reformulate these two notions in novel mathematical frameworks[2], as new perspectives often tend to bring new insights. One such mathematical framework, which will occupy a large portion of this thesis, is that of *$\mathcal{D}$-modules*. $\mathcal{D}$-modules study DEQs from an *algebraic* point of view, as a supplement to the usual tools from complex analysis. Its mathematical development goes back to the Japanese school of the 70s and 80s spearheaded by M. Sato and M. Kashiwara. (See [22] for a historical account and the textbooks [23, 24] for technical details. Early studies on the relationship between FIs and $\mathcal{D}$-modules include [25, 26] in the 1970s, and later [27, 28, 29] in the 2010s.)

One of the goals of this thesis is thus to re-interpret IBPs and DEQs in this language. To foreshadow the contents of future chapters, we shall interpret IBPs in terms of so-called *Macaulay matrices*, and solutions to DEQs via *"restrictions" of $\mathcal{D}$-modules*.

Numerical integration. Alongside the aforementioned developments in the analytical evaluation of FIs, there is also a tradition for the *numerical* evaluation thereof (see [30, Section 3.3] for a historical account). However, because of limited computer resources in the previous century, this tradition only goes back roughly two decades (some of the earliest developments include [31, 32]). But why go down this route? Although

[2] A prime example hereof is the recent reformulation of IBPs within the framework of *twisted cohomology* [21].

analytic control over FIs is arguably the most favorable situation, sometimes the symbolic expressions are just too complicated even with current tools! Nevertheless, blind numerical evaluation is rarely the most efficient way to proceed. It is better to perform some kind of symbolic *preprocessing*, so as to reveal an expression that is simple to numerically evaluate. For instance, one could still use IBPs to derive DEQs for FIs, but now with every parameter except one fixed to a number in order to simplify the IBPs. The DEQs are then written w.r.t. this single, unfixed parameter, and are solved through numerical series expansions rather than the analytic method of canonical bases. This approach, called *auxiliary mass flow*, has proven to be very fruitful in recent years [33, 34].

Tropical geometry. With the goal of numerical evaluation in mind, we shall carry out such a symbolic preprocessing by stepping into the world of *tropical geometry.* The mathematical disciple of *algebraic geometry* is perhaps already familiar to the reader. The latter studies the "smooth" geometries associated to the solutions of polynomial equations. Tropical geometry, instead, studies the solutions to equations involving "piecewise-linear" polynomials. This "piecewise" nature then translates into more "chunky" geometric figures, more precisely *polytopes*, which are comparatively easier to handle. (See the textbook [35] for a pedagogical exposition on tropical geometry.) The connection to FIs comes from representing them as integrals over certain polynomials raised to rational powers. By studying the simpler, tropical nature of these integrand polynomials, one can learn about the UV/IR divergent behaviour of FIs [36], as well as the connection between QFT and string theory amplitudes [37, 38].

For our purposes, the central notion will be that of a *tropical approximation* to a polynomial, envisioned by Panzer [39], which essentially just replaces ”+” with ”max”. We obtain tropical versions of FIs by applying this operation to the integrands. Despite their *simplicity* (or perhaps because of it!), tropical versions of FIs appear to retain an uncanny amount of information about the original integral [39, Section 5.1]. We shall employ this particular symbolic preprocessing, i.e. the tropicalization of the Feynman integrand, to simplify the numerical evaluation of FIs. This idea was originally put forward by Borinsky [40].

The content of this thesis

The above discussion hopefully motivates the main topic of this thesis, namely the evaluation of FIs. We have divided the presentation into three parts.

Part I. The first part of the thesis focuses on the evaluation of FIs through the method of DEQs.

In Chapter (2), starting from a single FI, we show how to associate to it a whole family of integrals $I_{\nu_1 \dots \nu_n}$, where each $\nu_i \in \mathbb{Z}$. There exist linear relations among integrals with different values of $(\nu_1, \dots, \nu_n)$, namely the IBPs mentioned above. We shall describe how to derive these relations, how to identify a set of MIs $\vec{I}$, and lastly how set up a system of first-order DEQs

$$\frac{\partial}{\partial z_i} \vec{I}(z) = P_i(z) \cdot \vec{I}(z) \quad \text{for} \quad i = 1, \dots, N\,. \tag{1.2}$$

$z = (z_1, \dots, z_N)$ is a collection of kinematic variables, and $P_i(z)$ are rational matrices. This is called a *Pfaffian system* of DEQs. Direct integration of FIs, often being too hard, is thereby sidestepped in favor of the comparatively easier task of solving Pfaffian systems.

In Chapter (3), we use the method of IBPs and Pfaffian systems to calculate a collection of 2-loop 3-point FIs with internally massive propagators. These integrals are then fed into the form factors for a scattering amplitude that is of relevance to dark matter detection.

Part II. The second part of this thesis is much more mathematically formal (though hopefully still accessible to a physicist audience), as it dives into the theory of $\mathcal{D}$*-modules*.

In Chapter (4), we begin by describing those concepts in the theory of $\mathcal{D}$-modules that are needed for developments in future chapters. This includes the notion of a *holonomic* $\mathcal{D}$-module, which carries the structure of a finite-dimensional vector space. This mimics the vector space structure of families of FIs due to IBPs.

In Chapter (5), we present a novel algorithm whose output is a Pfaffian system for a holonomic $\mathcal{D}$-module. It takes as input a collection of differential operators that annihilate a given function as well a basis of operators for the associated vector space. The key concept of the algorithm is that of a *Macaulay matrix*, which encodes relations among differential operators. This matrix is frequently used in the setting of commutative polynomial rings to avoid Gröbner basis computations, as they tend to scale poorly with the complexity of the problem. Indeed, one feature of the algorithm presented here is that it too avoids the need for Gröbner bases, now in a non-commutative ring of differential operators.

In Chapter (6), we draw our attention to a specific holonomic $\mathcal{D}$-module called the *GKZ system*, named after Gel'fand, Kapranov and Zelevinsky [41, 42]. It is built from a collection of differential operators that can immediately be written down given the integral representation of an *Euler integral.* The benefit from employing the GKZ system is two-fold: 1) a complete set of annihilating operators is known, and 2) an explicit vector space basis can be computed very fast.

In Chapter (7), the connection between Euler integrals and FIs is established by generalizing the parameters present in the *Lee-Pomeransky representation* [43] of a FI. The GKZ system can consequently be used to study this *generalized* FI. In particular, we use the GKZ system as input for the Macaulay matrix algorithm in order to derive Pfaffian systems for a couple of simple FIs.

In Chapter (8), we address the discrepancy between Euler integrals and FIs: an Euler integral often has too many parameters compared to the number of kinematic variables in a FI. To land on the $\mathcal{D}$-module for the FI proper, one must take a kind of "limit" of the GKZ system, called a *restriction* in the language of $\mathcal{D}$-modules. We develop two different methods for computing restrictions whose output is the Pfaffian system for the restricted $\mathcal{D}$-module. The first method is based on manipulating Pfaffian systems through series expansions and gauge transformations. The second method simply takes the limits directly at the level of the Macaulay matrix. The first method turns out to be applicable to the computation of logarithmic series solutions to Pfaffian systems. The second method turns out to allow for restrictions onto hyper*surfaces* in addition to just hyper*planes*, which, to the best of our knowledge, has so far not been achieved within the theory of $\mathcal{D}$-modules.

In Chapter (9), we showcase both restriction algorithms through several FI examples. The examples concerning logarithmic solutions to Pfaffian systems unfortunately only contain partial results, and so will have to be completed in future work.

Part III. The third and final part of this thesis shifts focus from analytical to numerical methods for evaluating FIs. The mathematical framework is that of *tropical geometry.*

In Chapter (10), we begin by describing the notion of a *tropical approximation* to a polynomial. For instance, given

$$p(x) = 2x_1x_2 + 3x_1^2 - 8x_2^2\,, \tag{1.3}$$

then its tropical approximation is

$$p^{\text{tr}}(x) = \max\{x_1x_2,\, x_1^2,\, x_2^2\}\,. \tag{1.4}$$

We thereafter present a striking observation made by Borinsky [40]: for an integral of a polynomial raised to some power $\int p(x)^a \mathrm{d}^n x$ (over a certain integration contour and modulo some technical conditions), one can build a *probability measure* from $p^{\mathrm{tr}}(x)$. By sampling points from this measure, it is thereby possible to perform numerical Monte Carlo integration. This can be used to evaluate FIs in parametric representation. We combine this *tropical integration* algorithm with the ϵ-expansion of FIs in the dimensional regularization scheme. Feynman's $i\varepsilon$-prescription is incorporated via a certain contour deformation $x \to xe^{if(x)}$, which maps the Schwinger integration variables x from real line into a path in the complex plane.

In Chapter (11), the program FEYNTROP is introduced. It is an implementation of the tropical Monte Carlo integration scheme, and includes both the ϵ-expansion and the $i\varepsilon$-prescription. The program can therefore numerically evaluate dimensionally regulated FIs for physical kinematics in Minkowski space (rather than just Euclidean kinematics). After a tutorial of the program, we then apply it to several non-trivial examples between 1-5 loops and 0-5 legs.

In Chapter (12), we present the conclusions reached throughout this thesis and speculate on developments for the future.

Part I

Multi-loop Technology

Chapter 2

Feynman Integrals

This chapter is a summary of basic concepts pertaining to Feynman integrals. It is far from an exhaustive review (for such a text, consult e.g. [44] and [45]).

Section 2.1

Definition

The main object of study in this thesis is the *Feynman integral* (FI) associated to an L-loop Feynman diagram G:

$$I_G = \lim_{\varepsilon \to 0^+} \int_{(\mathbb{R}^{1,\mathrm{D}-1})^L} \mathrm{d}^{L \cdot \mathrm{D}} \ell \prod_{e=1}^{E_{\text{int}}} \frac{1}{[-q_e(p,\ell)^2 + m_e^2 - i\varepsilon]^{\nu_e}} \,. \tag{2.1}$$

We often also write $I_{\nu_1 \nu_2 \ldots} = I(\nu_1, \nu_2, \ldots)$. Let us go through the notation used above.

2.1.1 Feynman diagram

A (scalar) *Feynman diagram* $G = (E, V)$ is defined by a set of edges $E = \{e_1, \ldots, e_{|E|}\}$ and vertices $V = \{v_1, \ldots, v_{|V|}\}$. Given the number of edges and vertices, the number of loops is given by $L = |E| - |V| + 1$. To each edge e, we associate a D-dimensional momentum vector flowing along the orientation of e. Momentum conservation is imposed at each vertex v, meaning that all edge momenta flowing into v equal those flowing out of v (in analogy with Kirchhoff's law for currents in electrical circuits).

2.1.2 The integrand

Let us partition the set of edges into *external* and *internal* ones, i.e. $E = E_{\text{ext}} \sqcup E_{\text{int}}$. The external edges have one end which is not connected to any other edge, while the internal edges are connected to at least one other edge on each end.

Consider an edge e. If $e \in E_{\text{ext}}$ is external, then we label its associated momentum vector by

$$p_e = \left(p_e^0, \ldots, p_e^{\mathrm{D}-1}\right) \in \mathbb{R}^{1,\mathrm{D}-1}\,, \tag{2.2}$$

where $\mathbb{R}^{1,\mathrm{D}-1}$ denotes D-dimensional Minkowski space equipped with the inner product

$$a \cdot b = a^0 b^0 - a^1 b^1 - \ldots - a^{\mathrm{D}-1} b^{\mathrm{D}-1}\,. \tag{2.3}$$

On the other hand, if $e \in E_{\text{int}}$ is internal, then we associate to it the vector $q_e \in \mathbb{R}^{1,\mathrm{D}-1}$ written as a sum of external momenta p_i and *loop momenta* ℓ_j:

$$q_e(p,\ell) = \sum_i \pm p_i \;\; + \;\; \sum_j \pm \ell_j\,. \tag{2.4}$$

The signs and summation ranges for (i,j) depend on the chosen *momentum routing* for the graph G. The L loop momenta $\{\ell_1, \ldots, \ell_L\}$ can be viewed as the leftover degrees of freedom after imposing momentum conservation at each vertex. In (2.1), these loop momenta are integrated against the measure

$$\mathrm{d}^{L\cdot\mathrm{D}}\ell := \mathrm{d}\ell_1^{(0)} \wedge \cdots \wedge \mathrm{d}\ell_1^{(\mathrm{D}-1)} \wedge \cdots \wedge \mathrm{d}\ell_L^{(0)} \wedge \cdots \wedge \mathrm{d}\ell_L^{(\mathrm{D}-1)} \tag{2.5}$$

over the whole of Minkowski space $\mathbb{R}^{1,\mathrm{D}-1}$.

To any edge e, we additionally assign a scalar mass $m_e \in \mathbb{R}_{\geq 0}$. For an external momentum, we say that it is *on-shell* if it satisfies Einstein's energy momentum relation: $p_e^2 = m_e^2$. An internal momentum is generally not on-shell, that is $q_e^2 \neq m_e^2$ (apart from special points within the integration range).

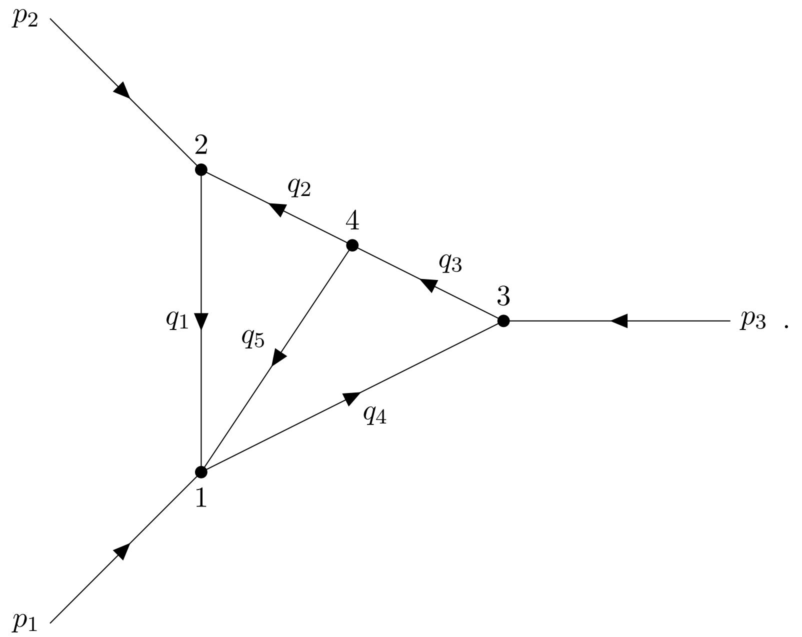
The integrand of (2.1) is constructed by taking a product of *Feynman propagators*

$$\frac{1}{[-q_e(p,\ell)^2 + m_e^2 - i\varepsilon]^{\nu_e}} \tag{2.6}$$

for each internal edge. The integers $\nu_e \in \mathbb{Z}$ are called *edge weights* or simply *propagator powers*. The infinitesimal imaginary part $i\varepsilon$ is introduced to set a particular integration contour. We shall have more to say about this in Section (10.2.1), but until then we can safely leave the $i\varepsilon$ as well as the limit $\varepsilon \to 0^+$ in front of (2.1) as implicit. In gauge theories such as QED and QCD, Feynman propagators become dressed with numerators having some theory-dependent tensor structure. This additional structure is often factored out in front of scalar FIs (cf. Section (3.2)).

Example 2.1. Consider the Feynman diagram

$G \quad =$



(2.7)

It has $L = 2$ loops by inspection. As a sanity check, we get the same value by writing $L = |E_{\text{int}}| - |V_{\text{int}}| + 1 = 5 - 4 + 1 = 2$, where we just used the internal edges and vertices for counting (the external lines add three edges but technically also three vertices, so these cancel in the formula for L).

We have chosen an arbitrary momentum routing by drawing arrows on each edge. The external momenta are denoted by $\{p_1, p_2, p_3\}$, and the internal ones are $\{q_1, \ldots, q_5\}$. Imposing momentum conservation at each vertex, we have

$$\begin{aligned} \text{vertex 1:} \quad & p_1 + q_1 + q_5 = q_4 \\ \text{vertex 2:} \quad & p_2 + q_2 = q_1 \\ \text{vertex 3:} \quad & p_3 + q_4 = q_3 \\ \text{vertex 4:} \quad & q_3 = q_2 + q_5 \,. \end{aligned} \tag{2.8}$$

There are not enough relations here to completely fix all of the momenta q_e. Given that $L = 2$, there will be two leftover degrees of freedom ℓ_1 and ℓ_2. Arbitrarily choosing to call $q_1 = \ell_1$ and $q_5 = \ell_2$, then a solution to the linear system above is

$$\begin{aligned} q_1 &= \ell_1 \\ q_2 &= \ell_1 - p_2 \\ q_3 &= \ell_1 + \ell_2 - p_2 \\ q_4 &= \ell_1 + \ell_2 - p_2 - p_3 \\ q_5 &= \ell_2 \,. \end{aligned} \tag{2.9}$$

Suppose the internal masses are given by $\{m_1, m_2, m_3, m_4, m_5\} = \{m, 0, 0, 0, m\}$, and that the edge weights are $\{\nu_1, \nu_2, \nu_3, \nu_4, \nu_5\} = \{1, 2, 1, 1, 3\}$.

The FI associated to this data is then

$$I_G = \int_{\left(\mathbb{R}^{(1,\mathrm{D}-1)}\right)^2} \frac{\mathrm{d}^{2\cdot\mathrm{D}}\ell}{\left[-q_1^2 + m^2\right]\left[-q_2^2\right]^2\left[-q_3^2\right]\left[-q_4^2\right]\left[-q_5^2 + m^2\right]^3}\,, \tag{2.10}$$

where the q_e are given by (2.9). ■

2.1.3 General features

Having defined our object of study, let us now describe some general features of FIs. The first thing to mention is that they are often *divergent* when the loop momenta are small and/or large, i.e. when $\ell \to 0$ and/or $|\ell| \to \infty$. This should not deter us, since we can introduce a regularization scheme to make these integrals well-defined. The most common scheme, and the one we adopt throughout this thesis, is that of *dimensional regularization* (DR) [46]. Although (2.1) will generically diverge in, say, $\mathrm{D} = 4$ spacetime dimensions, we can promote D to an indeterminate parameter, thereby considering $I_G = I_G(\mathrm{D})$ as a function of D. Divergences then show up as poles in $(\mathrm{D} - 4)$, leading us to interpret the FI a Laurent series in $(\mathrm{D} - 4)$. It is customary to write

$$\mathrm{D} = \mathrm{D}_0 - 2\epsilon \tag{2.11}$$

with $\mathrm{D}_0 \in \mathbb{Z}_{>0}$ and $|\epsilon| \ll 1$, such that

$$I_G(\mathrm{D}) = \sum_{i=i_{\min}}^{\infty} I_G^{(i)} \epsilon^i\,, \quad i_{\min} \in \mathbb{Z}\,. \tag{2.12}$$

The poles in ϵ happily cancel during the computations of physical observables. There are two mechanisms for these cancellations: UV renormalization and IR subtraction, but we shall not dive into these deep topics here. Note that the mass dimension of the integral measure $\left[\mathrm{d}^{\mathrm{D}\cdot L}\ell\right] = \mathrm{D}\cdot L$ changes in DR. To counterbalance this, one introduces another arbitrary parameter μ with mass dimension $[\mu] = 1$, leading to a new measure $(\mu^2)^{\epsilon L}\, \mathrm{d}^{\mathrm{D}\cdot L}\ell$. We will often be sloppy with factors of μ throughout this thesis, since they are easy to reinstate.

The end goal of a FI computation is to obtain the coefficient functions $I_G^{(i)}$. What do they look like, qualitatively speaking? Inspecting (2.1), observe that $I_G^{(i)}$ will be a functions of masses m_e and $(|E_{\text{ext}}|-1)/2$ scalar products $p_e \cdot p_d$. The factor of $1/2$ is due to $p_e \cdot p_d = p_d \cdot p_e$, and the factor $(|E_{\text{ext}}|-1)$ comes from momentum conservation

$$p_{|\text{Ext}|} = -\sum_{e=1}^{|E_{\text{ext}}|-1} p_e\,, \tag{2.13}$$

with the convention that all external momenta are in-coming. In practical calculations, it is useful to trade the dependence on scalar products for a dependence on *Mandelstam variables* $s_{ij} = (p_i + p_j)^2$.

So the $I_G^{(i)}$ are functions of masses and external momenta, but what kind of functions? The answer to this question is not yet settled in full generality, though much is known at low loop-order [47]. Part of the answer depends on what properties we would *like* a space of functions to have. Take, for instance, the famous Γ-function

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} \mathrm{d}x\,, \quad \mathrm{Re}(z) > 0\,. \tag{2.14}$$

Because Γ is *defined* by its integral representation, we don't think of that integral as being "leftover work" to be done, as opposed to e.g. $\int_0^z x^2\, \mathrm{d}x$. The integral representation of Γ is perfectly fine because

1. It allows for the derivation of many identities, such as the famous reflection formula $\Gamma(1-z)\Gamma(z) = \frac{\pi}{\sin(\pi z)}$, which can simplify calculations involving large expressions. For instance, there could be two terms like $4\Gamma(1-z)\Gamma(z) - 4\frac{\pi}{\sin(\pi z)}$ in some result, but it would be wasteful to numerically evaluate this expression since it gives 0.

2. It opens the door to other representations which allow for accurate numerical evaluations, such as the Stirling series $\Gamma(1+z) = \sqrt{2\pi z}\,(z/e)^z \exp\left(\sum_{i=1}^{\infty} \frac{B_{2i}}{2i(2i-1)z^{2i-1}}\right)$ written in terms of Bernoulli numbers B_i.

3. It is a starting point for determining the analytic continuation of $\Gamma(z)$ to larger regions of the complex plane.

Based on this analogy, we claim that "calculating" the ith ϵ-coefficient function $I_G^{(i)}$

is equivalent to rewriting it in terms of functions with "nice" numerical and analytic properties. A natural choice for such "nice" functions turns out to be the *Chen iterated integrals* which we define in Section (2.4).

SECTION 2.2

Integration-by-parts identities

In state-of-the-art computations at 2- or 3-loop orders, one typically encounters $\mathcal{O}(10^4)$ to $\mathcal{O}(10^6)$ different FIs. Fortunately, only a select few of these integrals, typically of order $\mathcal{O}(10^2)$, need to be calculated. This huge decrease in workload is a consequence of linear relations among FIs called *integration-by-parts identities* (IBPs) [6, 7, 8]. These identities express the many integrals appearing in a QFT computation in terms of a small basis of *master integrals* (MIs). The MIs can then be computed via the method of differential equations [16, 17, 18, 19], as outlined in Section (2.3).

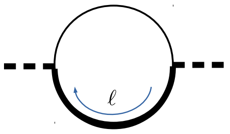
IBPs are born from the fact that a total derivative vanishes in DR [44, Section 6.1]:

$$0 = \int \mathrm{d}^{\mathrm{D}\cdot L}\ell \, \frac{\partial}{\partial \ell^\mu} \left[v^\mu \cdot f(\ell) \right] , \tag{2.15}$$

where v^μ is an arbitrary vector and $f(\ell)$ is some product of propagators. Let us go through an example to see why (2.15) leads to linear relations among FIs.

Example 2.1. Consider the following 1-loop bubble integral family[1]

$$I(\nu_1, \nu_2) \;=$$



$$= \int \frac{\mathrm{d}^{\mathrm{D}}\ell}{[-\ell^2 + m^2]^{\nu_1} \, [-(\ell + p)^2]^{\nu_2}} \, . \tag{2.16}$$

Say we choose $v^\mu = p^\mu$ in (2.15). Setting $D_1 = -\ell^2 + m^2$ and $D_2 = -(\ell + p)^2$, then the chain and product rules yield

$$\frac{\partial}{\partial \ell^\mu} \frac{p^\mu}{D_1^{\nu_1} D_2^{\nu_2}} = \frac{p^\mu}{D_2^{\nu_2}} \frac{\partial}{\partial \ell^\mu} \frac{1}{D_1^{\nu_1}} \;+\; \frac{p^\mu}{D_1^{\nu_1}} \frac{\partial}{\partial \ell^\mu} \frac{1}{D_2^{\nu_2}} \tag{2.17}$$

$$= \frac{p^\mu}{D_2^{\nu_2}} \frac{2\nu_1 \ell_\mu}{D_1^{\nu_1+1}} \;+\; \frac{p^\mu}{D_1^{\nu_1}} \frac{2\nu_2(\ell_\mu + p_\mu)}{D_2^{\nu_2+1}} \, . \tag{2.18}$$

The combination $2p^\mu \ell_\mu = 2p \cdot \ell$ appears in both terms. It can be rewritten in terms of

[1] The solid, thick line has mass m, the solid, thin line is massless, and the dashed lines have squared momentum p^2.

the two propagators D_1 and D_2 as follows:

$$\begin{aligned} D_2 &= -\ell^2 - p^2 - 2p\cdot\ell = D_1 - m^2 - p^2 - 2p\cdot\ell \implies \\ 2p\cdot\ell &= D_1 - D_2 - m^2 - p^2\,. \end{aligned} \tag{2.19}$$

Inserting this into (2.18) gives

$$\nu_1 \frac{D_1 - D_2 - m^2 - p^2}{D_2^{\nu_2}\, D_1^{\nu_1+1}} \;+\; \nu_2 \frac{D_1 - D_2 - m^2 - p^2 + 2p^2}{D_1^{\nu_1}\, D_2^{\nu_2+1}}\,. \tag{2.20}$$

Simplifying, reinstating the integral sign, and equating to zero, we find the family of IBPs

$$\begin{aligned} 0 &= (\nu_2 - \nu_1) I(\nu_1,\nu_2) \\ &+\;\; \nu_1\big[(m^2+p^2)I(\nu_1+1,\nu_2) + I(\nu_1+1,\nu_2-1)\big] \\ &+\;\; \nu_2\big[(m^2-p^2)I(\nu_1,\nu_2+1) - I(\nu_1-1,\nu_2+1)\big]\,. \end{aligned} \tag{2.21}$$

Choosing instead $v^\mu = \ell^\mu$, then a similar calculation yields

$$\begin{aligned} 0 &= (\mathrm{D} - 2\nu_1 - \nu_2) I(\nu_1,\nu_2) \\ &+\;\; 2m^2\nu_1 I(\nu_1+1,\nu_2) \\ &+\;\; \nu_2\big[(m^2-p^2)I(\nu_1,\nu_2+1) - I(\nu_1-1,\nu_2+1)\big]\,. \end{aligned} \tag{2.22}$$

The latter derivation requires the additional identity

$$\frac{\partial}{\partial\ell^\mu}\ell^\mu = (\partial_0, -\partial_i)^\mu\, \eta_{\mu\nu}\, (\ell_0,\ell_i)^\nu = \mathrm{D}. \tag{2.23}$$

Let's now showcase an *IBP reduction*. Setting $\nu_1 = \nu_1 = 1$ in (2.21), we have

$$0 = (m^2+p^2)I(2,1) + I(2,0) + (m^2-p^2)I(1,2)\,. \tag{2.24}$$

Here we set $I(0,2) = 0$. This is an example of a *scaleless* integral, i.e. it does not depend on any variables with mass dimension. It can be shown that scaleless integrals vanish in DR [44, Page 40][2].

Setting also $\nu_1 = \nu_2 = 1$ in (2.22), we get

$$0 = (\mathrm{D}-3)I(1,1) + 2m^2 I(2,1) + (m^2-p^2)I(1,2)\,. \tag{2.25}$$

[2] $I(0,2)$ naively does appear to depend on the scale p. But we can shift $\ell \to \ell - p$ without altering the integral since we are integrating over the whole space $\mathbb{R}^{1,\mathrm{D}-1}$, thereby removing the p-dependence.

The last equation we will need is (2.22) for $\nu_1 = 1$, $\nu_2 = 0$:

$$0 = (\mathrm{D}-2)I(1,0) + 2m^2 I(2,0)\,. \tag{2.26}$$

Combining the three equations (2.24), (2.25) and (2.26), a few lines of algebra grant

$$I(2,1) = \frac{\mathrm{D}-2}{2m^2(p^2-m^2)} I(1,0) + \frac{\mathrm{D}-3}{p^2-m^2} I(1,1)\,. \tag{2.27}$$

This is an example of an IBP reduction: we have expressed an integral $I(2,1)$ with "large" denominator powers in terms of two simpler integrals, $I(1,0)$ and $I(1,1)$. In fact, the IBPs (2.21) and (2.22) allow us to express *any* integral $I(\nu_1,\nu_2)$ in terms of $I(1,0)$ and $I(1,1)$. These two integrals are thereby dubbed the *master integrals* (MIs) for the integral family $I(\nu_1,\nu_2)$. These MIs are of course not unique. The MI $I(1,1)$ could for instance be swapped for $I(2,1)$ via (2.27). ■

IBP reduction has been automated in many public codes such has KIRA [10], FIRE [11], REDUZE [12], NEATIBP [13], BLADE [14], and LITERED [15].

2.2.1 Irreducible scalar products

There is an important detail to deriving IBPs which did not feature in Example (2.1), namely that of *irreducible scalar products* (ISPs). Recall (2.19), where we expressed the scalar product $2p\cdot\ell$ in terms of the propagators D_1 and D_2. This is always possible at 1-loop order because there is a one-to-one correspondence between scalar products and propagators. At 2-loop order or higher, this is no longer the case, as there will generally be more scalar products than propagators. These "extra" scalar products are called *irreducible.*

Example 2.2. Take for instance a 2-loop integral written in terms 3 propagators:

$$\begin{aligned} I(\nu_1,\nu_2,\nu_2) &= \int \frac{\mathrm{d}^{2\cdot\mathrm{D}}\ell}{D_1^{\nu_1}\,D_2^{\nu_2}\,D_3^{\nu_3}} \quad \text{where} \\ D_1 = -\ell_1^2\,, &\quad D_2 = -\ell_2^2\,, \quad D_3 = -(\ell_1+\ell_2+p)^2\,, \end{aligned} \tag{2.28}$$

with p being the only external momentum. It is possible to form $5 > 3$ scalar products

$$\ell_1^2\,, \quad \ell_1^2\,, \quad \ell_1\cdot\ell_2\,, \quad \ell_1\cdot p\,, \quad \ell_2\cdot p\,. \tag{2.29}$$

Two out of the last three scalar products are irreducible; say we choose $\ell_1\cdot\ell_2$ and $\ell_1\cdot p$.

The trick is simply to enlarge the original integral family to include two new integrand factors $D_4 = \ell_1 \cdot \ell_2$ and $D_5 = \ell_1 \cdot p$, leading to

$$I(\nu_1, \ldots, \nu_5) = \int \frac{\mathrm{d}^{2 \cdot \mathrm{D}} \ell}{D_1^{\nu_1} \cdots D_5^{\nu_5}} \,. \tag{2.30}$$

In intermediate steps of an IBP reduction, D_4 and D_5 will generally appear in numerators. This is the reason for letting the edge weights $\nu_e \in \mathbb{Z}$ be integral rather than just non-negative. ■

Section 2.3

The method of differential equations

So IBPs allow us to write a given FI in terms of MIs. The next question is how to compute the MIs themselves. Let $I(\nu) = I(\nu_1, \nu_2, \ldots)$ denote a family of FIs. This family depends on some kinematic variables, call them $z = (z_1, z_2, \ldots)$, such as masses m_e^2 and Mandelstam variables s_{ij}. Suppose a vector of MIs $\vec{I}$ has been revealed after IBP reduction. The *method of differential equations* for computing the MIs consists of the following steps:

1. Take derivatives $\partial_i \vec{I}$, w.r.t. the kinematic variables z_i. These derivatives lead to shifts in the edge weights ν_e.

2. IBP reduce the integrals with shifted ν_e down to MIs $\vec{I}$. The reduction can be factored as $P_i(z) \cdot \vec{I}$ for some matrices P_i whose entries are rational functions of z and D. The resulting system of DEQs is

$$\partial_i \vec{I} = P_i(z) \cdot \vec{I} \,. \tag{2.31}$$

   This is called a *Pfaffian system*, and will be a major topic of interest for this thesis. The matrices P_i are called *Pfaffian matrices.*

3. Solve the Pfaffian system in terms of a *path-ordered exponential.*

Let us illustrate the first three steps by an example. The fourth step will be discussed in the next section.

Example 2.1. This is a continuation of Example (2.1), wherein we found two MIs

$$\vec{I} = \begin{bmatrix} I(1,0) \\ I(1,1) \end{bmatrix} \tag{2.32}$$

for a 1-loop integral family.

There are two kinematic variables $(z_1, z_2) = (m^2, p^2)$ w.r.t. which we can take derivatives. The derivative w.r.t. z_1 works out to

$$\begin{aligned}
\partial_1 \vec{I} &= \begin{bmatrix} -I(2,0) \\ -I(2,1) \end{bmatrix} && (2.33)\\
&\overset{\text{IBP}}{=} \begin{bmatrix} \frac{\mathrm{D}-2}{2z_1} I(1,0) \\ \frac{\mathrm{D}-2}{2z_1(z_1-z_2)} I(1,0) + \frac{\mathrm{D}-3}{z_1-z_2} I(1,1) \end{bmatrix} && (2.34)\\
&= \begin{bmatrix} \frac{\mathrm{D}-2}{2z_1} & 0 \\ \frac{\mathrm{D}-2}{2z_1(z_1-z_2)} & \frac{\mathrm{D}-3}{z_1-z_2} \end{bmatrix} \cdot \begin{bmatrix} I(1,0) \\ I(1,1) \end{bmatrix} && (2.35)\\
&= P_1 \cdot \vec{I}\,. && (2.36)
\end{aligned}$$

A similar computation can be carried out for z_2. The result is

$$\partial_2 \vec{I} = \begin{bmatrix} 0 & 0 \\ \frac{\mathrm{D}-2}{2z_2(z_1-z_2)} & \frac{(2-\mathrm{D})z_1+(4-\mathrm{D})z_2}{2z_2(z_1-z_2)} \end{bmatrix} \cdot \vec{I} = P_2 \cdot \vec{I}\,. \tag{2.37}$$

Pfaffian matrices such as P_1 and P_2 satisfy interesting identities. The *Euler homogeneity condition*

$$z_1 P_1 + z_2 P_2 = \begin{bmatrix} \frac{\mathrm{D}-2}{2} & 0 \\ 0 & \frac{\mathrm{D}-4}{2} \end{bmatrix} \tag{2.38}$$

reflects the fact that the MIs are certain homogeneous functions of z. Furthermore, the *integrability condition*

$$\partial_1 P_2 - \partial_2 P_1 = [P_1, P_2] \tag{2.39}$$

follows from the fact that the MIs are C^2 (twice continuously differentiable). ■

2.3.1 Canonical bases

Obtaining the Pfaffian system for a basis of MIs is only part of the work, the next step is to calculate a solution. In practice, we are only interested in computing solutions up a specific order in the ϵ-expansion (2.12). As discovered by Henn [20], it is particularly simple to find this expansion if we first write the Pfaffian system in *canonical form.*

The starting point is the Pfaffian system $\partial_i \vec{I} = P_i(z|\epsilon) \cdot \vec{I}$ for N kinematic variables $z = (z_1, \ldots, z_N)$. Note that we highlight both the z- and ϵ-dependence of the matrix P_i. It is more clean to write this system in terms of the total derivative $\mathrm{d}_z = \mathrm{d}z_1 \frac{\partial}{\partial z_1} + \ldots + \mathrm{d}z_N \frac{\partial}{\partial z_N}$ as

$$\mathrm{d}_z \vec{I} = P(z|\epsilon) \cdot \vec{I} \quad \text{where} \quad P = P_1(z|\epsilon)\, \mathrm{d}z_1 + \ldots + P_N(z|\epsilon)\, \mathrm{d}z_N \,. \tag{2.40}$$

Suppose we would like to change to a new basis of MIs $\vec{J}$. Due to IBP identities, there must necessarily exist an invertible matrix $G = G(z|\epsilon)$ such that $\vec{I} = G \cdot \vec{J}$. Inserting this expression for $\vec{I}$ into (2.40), it follows from the chain rule that

$$\mathrm{d}_z \vec{J} = Q(z|\epsilon) \cdot \vec{J} \quad \text{where} \quad Q_i(z|\epsilon) = G^{-1} \cdot \big[P_i \cdot G - \partial_i G \big] \,. \tag{2.41}$$

We say that the new Pfaffian matrices Q_i are obtained from a *gauge transformation* of P_i by G. Now, assume that the basis $\vec{J}$ has the following special properties:

1. The ϵ-dependence *factorizes* in its associated Pfaffian matrices, that is

$$Q(z|\epsilon) = \epsilon\, Q(z) \,. \tag{2.42}$$

2. The poles coming from the denominators of the Pfaffian matrices all have order 1. In particular, we can write

$$\epsilon\, Q(z) = \epsilon \sum_{\eta \in A} Q_i \, \mathrm{d} \log \eta(z) \,, \tag{2.43}$$

where

$$\mathrm{d} \log \eta(z) = \frac{1}{\eta(z)} \frac{\partial \eta(z)}{\partial z_1} \mathrm{d}z_1 + \ldots + \frac{1}{\eta(z)} \frac{\partial \eta(z)}{\partial z_N} \mathrm{d}z_N \,. \tag{2.44}$$

Every matrix entry $(Q_i)_{ab} \in \mathbb{Q}$ is here a rational number, and the (possibly algebraic or even transcendental) functions $\eta(z)$ are called *letters.* The full set of letters, denoted here by A, is called the *alphabet.*

By virtue of these two properties, $\vec{J}$ is called a *canonical basis.* Many calculations in the literature have shown that when condition 1 is satisfied, then condition 2 follows automatically - though, to the best of our knowledge, this has yet to be proven rigorously. In practice, the effort in searching for canonical bases is hence put into ϵ-factorization.

The ϵ-expansion of the canonical solution vector $\vec{J}$ is now particularly simple:

$$\vec{J} = \mathcal{P}\exp\left(\epsilon\int_\Gamma Q(z)\right)\cdot\vec{J_0}(\epsilon)\,, \tag{2.45}$$

where the *path-ordered exponential* is given by

$$\mathcal{P}\exp\left(\epsilon\int_\Gamma Q(z)\right) = \mathbf{1} + \epsilon\int_0^t Q(\Gamma(t_1)) + \epsilon^2\int_0^t Q(\Gamma(t_1))\int_0^{t_1} Q(\Gamma(t_2)) + \mathcal{O}(\epsilon^3) \tag{2.46}$$

for some chosen integration contour Γ parametrized by $t \in \mathbb{R}$. The vector of boundary constants also has an ϵ-expansion:

$$\vec{J_0}(\epsilon) = \vec{J_0}^{(0)} + \epsilon\vec{J_0}^{(1)} + \epsilon^2\vec{J_0}^{(2)} + \mathcal{O}(\epsilon^3)\,, \tag{2.47}$$

with each vector $\vec{J_0}^{(n)}$ consisting solely of real numbers. These boundary vectors are often determined by considering special limits of the MIs; concrete examples are given in Section (3.3).

The solution above is written somewhat abstractly in terms of a matrix-valued one-form $Q(z)$ (given in (2.43)) and some unspecified contour Γ. In Section (2.4), we shall rewrite the iterated integrals (2.46) in form that is more suitable for computer implementations.

2.3.2 Magnus expansion

With the discussion above in mind, the key question is thus the following: given a Pfaffian system $\partial_i\vec{I} = P_i\cdot\vec{I}$, is it possible to algorithmically construct a gauge transformation G such that the vector $\vec{J} = G^{-1}\cdot\vec{I}$ satisfies a new system in canonical form? Such an algorithm should preferably terminate in reasonable time, and be applicable to a large category of FIs. This question has been, and is, subject to intense study (see e.g. [44, Chapter 7] and the references therein). In this section, we present a method for that calculating canonical forms that is at least effective for the class of functions called *generalized polylogarithms* (GPLs), to be defined in Section (2.4). (There are now also many methods that output canonical forms for FIs even when they cannot be evaluated in terms of GPLs [48, 49, 50, 51, 52, 53, 54].)

The approach described here is based on the *Magnus expansion* for solving systems of

DEQs [55]. 60 years after Magnus' work, Argeri et al. [56] discovered that his expansion can be used to obtain canonical forms. The starting point is a Pfaffian system in *pre-canonical form*:

$$\mathrm{d}_z\vec{I} = \left[P^{(0)}(z) + \epsilon P^{(1)}(z)\right] \cdot \vec{I}\,. \tag{2.48}$$

I.e. a system linear in ϵ (the algorithm can be generalized to a polynomial dependence on ϵ too [57, Section 5.2.4]). By choosing judicial edge weights and simple ϵ-dependent prefactors for the MIs $\vec{I}$, it is often not too difficult to find a pre-canonical form.

The clever observation is now this: suppose that there exists a matrix $G = G(z)$, depending only on z, subject to the matrix DEQ

$$\mathrm{d}_z G = P^{(0)}(z) \cdot G\,. \tag{2.49}$$

If we change basis to $\vec{I} = G \cdot \vec{J}$, then the gauge transformation (2.41) simplifies to

$$\begin{aligned} Q(z|\epsilon) &= G^{-1} \cdot \left[\left(P^{(0)} + \epsilon P^{(1)}\right) \cdot G - \mathrm{d}_z G\right] && (2.50)\\ &= G^{-1} \cdot \left[\left(P^{(0)} + \epsilon P^{(1)}\right) \cdot G - P^{(0)} \cdot G\right] && (2.51)\\ &= \epsilon\, G^{-1} \cdot P^{(1)} \cdot G\,, && (2.52) \end{aligned}$$

which is indeed ϵ-factorized.

The question is thus how to find the solution G to the matrix DEQ (2.49). It turns out that the solution takes the form of a matrix exponential:

$$G = e^{\Omega[P^{(0)}]} \cdot G(z=0)\,. \tag{2.53}$$

For our purposes, we can simply take the boundary condition to be the identity matrix, $G(z=0) = \mathbf{1}$. The Ω matrix in the exponent enjoys a *Magnus expansion*

$$\Omega\left[P^{(0)}\right] = \sum_{n=1}^{\infty} \Omega_n\left[P^{(0)}\right], \tag{2.54}$$

with each term computed by integrating iterated commutators:

$$\begin{aligned} \Omega_1\left[P^{(0)}\right] &= \int_\bullet^z \mathrm{d}z_1\, P^{(0)}(z_1) \\ \Omega_2\left[P^{(0)}\right] &= \frac{1}{2}\int_\bullet^z \mathrm{d}z_1 \int_\bullet^{z_1} \mathrm{d}z_2\, \left[P^{(0)}(z_1),\, P^{(0)}(z_2)\right] && (2.55) \end{aligned}$$

$$\Omega_3\big[P^{(0)}\big] = \frac{1}{6}\int_\bullet^z \mathrm{d}z_1 \int_\bullet^{z_1} \mathrm{d}z_2 \int_\bullet^{z_2} \mathrm{d}z_3 \Big\{ \Big[P^{(0)}(z_1),\, \Big[P^{(0)}(z_2),\, P^{(0)}(z_3)\Big]\Big] + \Big[P^{(0)}(z_3),\, \Big[P^{(0)}(z_2),\, P^{(0)}(z_1)\Big]\Big] \Big\}.$$

Clean formulas for the nth term of this expansion can be found in [58]. Here $P^{(0)}$ was assumed to depend on just a single variable z. For multivariate problems, one can simply apply these formulas one variable at a time [59, Page 108].

It turns out that the Magnus expansion truncates for FIs that can be evaluated in terms of GPLs, i.e. $\Omega_{n_{\max}} = 0$ for some $n_{\max} \in \mathbb{Z}_{>0}$. In such cases, the matrix $P^{(0)}$ is apparently sparse enough to become nilpotent, meaning that the iterated commutators eventually evaluate to zero.

Example 2.2. Here we apply the Magnus expansion to the bubble integral from Examples (2.1) and (2.1). Consider a basis with squared propagators:

$$\vec{I} = \begin{bmatrix} I(2,0) \\ I(2,1) \end{bmatrix}. \tag{2.56}$$

Using IBPs as in Example (2.1), it is possible to obtain two Pfaffian matrices P_1 and P_2 in the variables $z_1 = m^2$ and $z_2 = p^2$. Working in units such that $z_1 = 1$, then P_1 can be discarded. Call $z_2 = z$. We thence get a pre-canonical system

$$\frac{\partial}{\partial z}\vec{I} = \Big[P^{(0)} + \epsilon\, P^{(1)}\Big] \cdot \vec{I}, \tag{2.57}$$

where the constant and linear matrices w.r.t. ϵ are

$$P^{(0)} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{z} \end{bmatrix} \quad \text{and} \quad P^{(1)} = \begin{bmatrix} 0 & 0 \\ \frac{1}{(1-z)z} & \frac{1+z}{(1-z)z} \end{bmatrix}. \tag{2.58}$$

The Magnus expansion truncates at the first term, giving

$$\Omega\big[P^{(0)}\big] = \Omega_1\big[P^{(0)}\big] = \int_\bullet^z \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{z'} \end{bmatrix} \mathrm{d}z' = \begin{bmatrix} 0 & 0 \\ 0 & -\log(z) \end{bmatrix}, \tag{2.59}$$

wherefore

$$G = e^{\Omega[P^{(0)}]} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{z} \end{bmatrix}. \tag{2.60}$$

Gauge transforming with G via (2.52) yields a new Pfaffian matrix

$$Q = \epsilon \begin{bmatrix} 0 & 0 \\ \frac{1}{1-z} & \frac{2}{1-z} + \frac{1}{z} \end{bmatrix}, \tag{2.61}$$

which is indeed canonical. ■

Example 2.3. Every physicist's favorite equation is the DEQ for the harmonic oscillator:

$$\frac{\mathrm{d}^2 f(t)}{\mathrm{d}t^2} + \omega^2 f(t) = 0\,. \tag{2.62}$$

The angular frequency ω is taken to be constant. Let us solve this equation using the Magnus expansion. The initial step is to rewrite the DEQ as a first-order matrix equation. Setting $f_1(t) = f(t)$ and $f_2(t) = \frac{\mathrm{d}f(t)}{\mathrm{d}t}$, then the DEQ is equivalent to

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \cdot \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix}. \tag{2.63}$$

In addition to solving *matrix* DEQs such as (2.49), the Magnus expansion also grants solution *vectors* in the form

$$\begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix} = e^{\Omega(t)} \cdot \begin{bmatrix} f_1(0) \\ f_2(0) \end{bmatrix}. \tag{2.64}$$

The Magnus expansion truncates at the first term because ω is constant. Thus,

$$e^{\Omega(t)} = \exp\left(\int_0^t \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \mathrm{d}t' \right) \tag{2.65}$$

$$= \begin{bmatrix} \cosh\left(\sqrt{-\omega^2} t\right) & \frac{\sinh\left(\sqrt{-\omega^2} t\right)}{\sqrt{-\omega^2}} \\ \sqrt{-\omega^2} \sinh\left(\sqrt{-\omega^2} t\right) & \cosh\left(\sqrt{-\omega^2} t\right) \end{bmatrix} \tag{2.66}$$

$$= \begin{bmatrix} \cos(\omega t) & \frac{\sin(\omega t)}{\omega} \\ -\omega \sin(\omega t) & \cos(\omega t) \end{bmatrix}. \tag{2.67}$$

We obtain oscillating functions as expected. Further, this solution satisfies $f_2(t) = \dot{f}_1(t)$. ■

Section 2.4

Generalized polylogarithms

The path-ordered exponential (2.46) naturally leads to a class of functions called *Chen iterated integrals* [60]. Let us first say what we mean by an integration contour for these integrals. Suppose that there are N kinematic variables $z = (z_1, \ldots, z_N)$ which parametrize an N-dimensional (smooth) complex manifold Z. We write write a path in this space as

$$\Gamma : [a, b] \to Z\,, \tag{2.68}$$

where $\Gamma(a) = z_a \in Z$ and $\Gamma(b) = z_b \in Z$ are the start and end points. One can think of $t \in [a, b] \subset \mathbb{R}$ as being a "time" coordinate.

Let $\{\omega_1, \ldots, \omega_M\}$ denote a collection of differential 1-forms on Z. For example, they could be the dlog forms from (2.43). These 1-forms are supposed to be integrated over the contour Γ. So instead of parametrizing the ω_i in terms of z, we would like to parametrize them using $t \in [a, b]$. This is called a *pullback* to the interval $[a, b]$, written as

$$\Gamma^* \omega_i = f_i(t)\mathrm{d}t\,. \tag{2.69}$$

This gives a concrete expression for the integral of 1-form along Γ:

$$\int_\Gamma \omega_i = \int_a^b f_i(t)\mathrm{d}t\,. \tag{2.70}$$

The question is how to compute the function $f_i(t)$. Writing

$$\omega_i = \sum_{j=1}^{N} \omega_{ij}(z)\mathrm{d}z_j \quad \text{and} \quad \Gamma(t) = \begin{bmatrix} \Gamma_1(t) \\ \vdots \\ \Gamma_N(t) \end{bmatrix}, \tag{2.71}$$

then the formula for $f_i(t)$ follows from the chain rule:

$$\Gamma^* \omega_i = \sum_{j=1}^{N} \omega_{ij}\big(\Gamma(t)\big)\frac{\mathrm{d}\Gamma_j(t)}{\mathrm{d}t}\mathrm{d}t \quad \Longrightarrow \quad f_i(t) = \sum_{j=1}^{N} \omega_{ij}\big(\Gamma(t)\big)\frac{\mathrm{d}\Gamma_j(t)}{\mathrm{d}t}\,. \tag{2.72}$$

Example 2.1. To demystify the construction above, let us compute

$$\int_\Gamma \omega \tag{2.73}$$

with

$$\omega = \frac{2}{z_2 - 2}\mathrm{d}z_1 + \frac{3}{z_1+4}\mathrm{d}z_2\,, \quad \text{and} \quad \Gamma : [0,1] \to \mathbb{C}^2 \quad \text{given by} \quad \Gamma(t) = \begin{bmatrix} t^2 \\ 1-t \end{bmatrix}.$$

Inserting this data into (2.72) gives

$$\begin{aligned} f(t) &= \omega_1\big(\Gamma(t)\big) \times \frac{\mathrm{d}\Gamma_1(t)}{\mathrm{d}t} \;+\; \omega_2\big(\Gamma(t)\big) \times \frac{\mathrm{d}\Gamma_2(t)}{\mathrm{d}t} && (2.74)\\ &= \frac{2}{(1-t)-2} \times 2t \;+\; \frac{3}{t^2+4} \times (-1) && (2.75)\\ &= -\frac{4t^3+19t+3}{(1+t)(4+t^2)}\,. && (2.76) \end{aligned}$$

Using any CAS, one can then easily integrate

$$\int_\Gamma \omega \;=\; \int_0^1 f(t)\mathrm{d}t \;\approx\; -1.923\,. \tag{2.77}$$

■

Given a $t \in [a,b]$, the *Chen iterated integral* can now be defined as

$$\begin{aligned} I_\Gamma(\omega_1,\dots,\omega_M|t) &= \int_\Gamma \omega_1 \wedge \cdots \wedge \omega_M && (2.78)\\ &= \int_a^t f_1(t_1)\mathrm{d}t \int_a^{t_1} f_2(t_2)\mathrm{d}t_2 \cdots \int_a^{t_{M-1}} f_M(t_M)\mathrm{d}t_M\,. && (2.79) \end{aligned}$$

An equivalent, recursive definition is

$$\begin{aligned} I_\Gamma(\,|t) &= 1 && (2.80)\\ I_\Gamma(\omega_1,\dots,\omega_M|t) &= \int_a^t f_1(t_1) I_\Gamma(\omega_2,\dots,\omega_M|t_1)\mathrm{d}t_1\,. && (2.81) \end{aligned}$$

Chen proved that I_Γ is independent of Γ, provided that Γ does not hit any singularities nor cross any branch points of the integrand for $a < t < b$. If the integrand has singularities at the endpoints $t = a$ or $t = b$, these are assumed to be integrable singularities (caveat: endpoint singularities can also be regularized using so-called shuffle identities [61]).

Consider now a special class of 1-forms with a simple pole at $t = z_i$:

$$\omega_{\text{GPL}}(z_i) = \frac{\mathrm{d}t}{t - z_i} = \mathrm{d}\log(t - z_i)\,. \tag{2.82}$$

The complex number z_i is treated as a parameter. Let Γ be a path along the non-negative real axis such that $\Gamma(a = 0) = 0$ and $\Gamma(t) = y \in \mathbb{R}_{\geq 0}$. The special class of Chen iterated integrals called *generalized multiple polylogarithms* (GPLs) [62] is defined by

$$G(z_1, \ldots, z_M|y) = I_\Gamma\left(\omega_{\text{GPL}}(z_1), \ldots, \omega_{\text{GPL}}(z_M)|y\right)\,. \tag{2.83}$$

Explicitly,

$$G(\underbrace{0, \ldots, 0}_{M \text{ times}}|y) = \frac{1}{M!}\log^M(y) \tag{2.84}$$

$$G(z_1, \ldots, z_M|y) = \int_0^y \frac{\mathrm{d}y_1}{y_1 - z_1} G(z_2, \ldots, z_M|y_1)\,. \tag{2.85}$$

The z-parameters are often called *weights*. The case $z = 0$ above naively constitutes a divergence of the integral representation, so the first formula above is more of a stipulation than a derivation.

There is a huge literature on these functions, see [61] [44, Chapters 6,8] and references therein. There are also plenty of powerful codes for symbolically manipulating and numerically evaluating these functions: the MAPLE package [63], HYPERINT [64], HPLOG [65], HPL [66], POLYLOGTOOLS [67], GINAC [68], CHAPLIN [69], HANDYG [70], and FASTGPL [71].

Now harken back to the path-ordered exponential (2.46). If the functions $\eta(z)$ from (2.43) are *rational* functions of the kinematic variables z, then the solution for the FI in question can be written in terms of GPLs. This is indeed a favorable form for the solution because so much is known about the identities among GPLs, how to analytically continue them, and how to numerically evaluate them.

One may encounter letters η that depend on z in terms of algebraic roots. In those cases, one can sometimes find a change of a variables $w = f(z)$ that turns the roots into rational functions. The package RATIONALIZEROOTS [72] automates the procedure for finding such an f.

Chapter 3

Two-loop Feynman Integrals for Dark Matter Detection

We now apply the multi-loop technology from the previous chapter to compute form factors for a *leptophillic dark matter* (DM) model. Form factors are certain building blocks of scattering amplitudes. For phenomenological implications of these results, we refer to the paper [73] on which this chapter is based.

Section 3.1

The model

Consider a DM particle χ that scatters elastically off a nucleon A: $\chi A \to \chi A$. This is called a *direct DM search*, and is probed by remarkable experiments having large tanks filled with e.g. oxygen, helium or xenon nucleons.

Assuming that DM does not couple *directly* to SM particles, this scattering can only occur via a *mediator* particle ϕ that *does* couple to the SM. Here we consider a *leptophillic* DM model, wherein χ couples to ϕ, which in turn only couples to SM leptons; the interaction between leptons and the quarks inside the nucleon A is finally mediated via photons. The mediator ϕ is taken to be a singlet scalar (called ϕ_S) or a pseudo-scalar (called ϕ_P). The Lagrangian interaction terms are thus of the form

$$-\mathcal{L}_S \supseteq g_S\, \phi_S\, \bar{\ell}\, \ell \,+\, g_\chi\, \phi_S\, \Gamma_\chi \quad \text{or} \quad -\mathcal{L}_P \supseteq i\, g_P\, \phi_P\, \bar{\ell}\, \gamma_5\, \ell \,+\, g_\chi\, \phi_P\, \Gamma_\chi\,, \tag{3.1}$$

where $\Gamma_\chi = \chi^\dagger \chi$ for scalar DM and $\Gamma_\chi = \{\bar{\chi}\, \chi,\, \bar{\chi}\, \gamma_5\, \chi\}$ for fermionic DM, $\{g_S, g_P, g_\chi\}$ are coupling constants, and $\{\ell, \bar{\ell}\}$ denote a lepton and an anti-lepton.

Since this model assumes no tree-level couplings to quarks, the χA scattering cross section is purely loop-induced. It turns out that the leading contributions come from

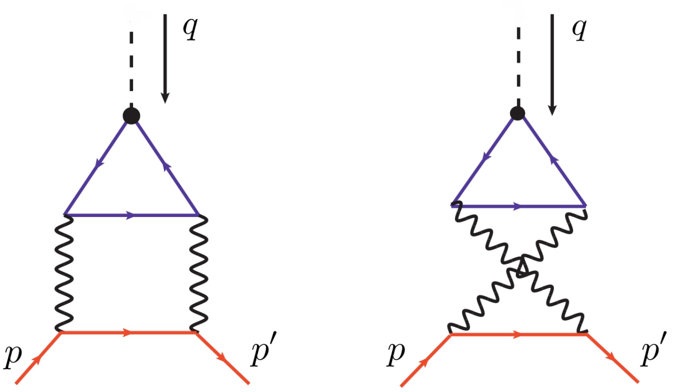


Figure 3.1: Leading contributions to the scattering amplitude for the leptophillic DM model.

the diagrams[1] in Figure (3.1). This is a t-channel process, with an incoming mediator $\phi(q)$ (dashed black lines) transferring momentum to a nucleon $A(p)$ (leftmost red lines), leading to a final state nucleon $A(p')$ (rightmost red lines). Momentum conservation dictates that $q = p' - p$. The blue triangle loops represent a lepton which couples to (the quarks inside) the nucleon via photons (wiggly black lines).

Let m_A denote the nucleon mass, meaning that it can either be the proton or neutron mass. The kinematic invariants for this process are then defined by

$$t = q^2 = (p' - p)^2, \quad m_A^2 = p^2 = p'^2 \,. \tag{3.2}$$

Section 3.2

The scattering amplitude

According to the Feynman rules dictated by the Lagrangian terms (3.1), the desired scattering amplitude takes the form

$$\mathcal{A}_{S,P}(t, m_\ell, m_A) = i\, g_{S,P}\, Q_A^2 \sum_{\ell = e,\, \mu,\, \tau} Q_\ell^2 \Big[\bar{u}_A(p') \cdot \Gamma_{S,P}(t, m_\ell, m_A) \cdot u_A(p) \Big] \,. \tag{3.3}$$

Let us parse this expression from left to right. The amplitude $\mathcal{A}$ has an index S or P, depending on whether the mediator $\phi_{S,P}$ is taken to be scalar or pseudo-scalar. $\mathcal{A}$ has kinematic dependence on the momentum transfer variable t, the lepton masses

[1] These are only the so-called one-body diagrams. The virtual photons can also scatter off two nucleons, yielding additional one-loop two-body diagrams. See [73] for more details.

$m_\ell = \{m_e, m_\mu, m_\tau\}$ and the nucleon mass m_A. The prefactor in front of the sum contains the coupling constants $g_{S,P}$ from (3.1) and the electric charge of the nucleon, where $Q_A = 1$ for the proton and $Q_A = 0$ for the neutron[2]. The lepton electric charge inside the sum is $Q_\ell = 1$. The sum over leptons contains an operator $\Gamma_{S,P}(t, m_\ell, m_A)$ sandwiched between spinors $\bar{u}(p')$ and $u(p)$ for the final and initial state nucleons respectively. This operator stems from the sum of the two diagrams above. Its analytic expression is

$$\Gamma_{S,P}(t, m_\ell, m_A) = -32\pi^2 \alpha_{\text{em}} \int \frac{\mathrm{d}^{\mathrm{D}}\ell_1}{(2\pi)^{\mathrm{D}}} \frac{\mathrm{d}^{\mathrm{D}}\ell_2}{(2\pi)^{\mathrm{D}}} \frac{g_{\mu\rho}\ \gamma^\rho\ (\not{\ell}_2 - m_A)\ g_{\nu\sigma}\ \gamma^\sigma\ \mathrm{Tr}^{\mu\nu}_{S,P}}{D_1 D_2 D_3 D_4 D_5 D_6}\,, \tag{3.4}$$

where the electromagnetic coupling is $\alpha_{\text{em}} = e^2/(4\pi)$, and the Dirac trace over spinor indices is given by

$$\mathrm{Tr}^{\mu\nu}_{S,P} = \mathrm{Tr}\left[(\not{\ell}_1 + \not{q} + m_\ell) \cdot \Lambda_{S,P} \cdot (\not{\ell}_1 + m_\ell) \cdot \gamma^\mu \cdot (\not{\ell}_1 + \not{\ell}_2 + \not{p}' + m_\ell) \cdot \gamma^\nu\right]\,, \tag{3.5}$$

with $\Lambda_S = \mathbf{1}$ and $\Lambda_P = \gamma_5$. The denominators of the momentum space integral (3.4) are

$$\begin{aligned} &D_1 = \ell_1^2 - m_\ell^2, \quad D_2 = (\ell_1 + q) - m_\ell^2, \quad D_3 = \ell_2^2 - m_A^2 \\ &D_4 = (\ell_2 + p)^2, \quad D_5 = (\ell_2 + p')^2, \quad\quad D_6 = (\ell_1 + k_2 + p')^2 - m_\ell^2\,. \end{aligned} \tag{3.6}$$

It is standard to write such an amplitude containing tensor structure as

$$\mathcal{A}_{S,P}(t, m_\ell, m_A) \propto \sum_{\ell = e,\mu,\tau} \mathcal{F}_{S,P} \tag{3.7}$$

where the *form factors* $\mathcal{F}_{S,P}$ are related to $\Gamma_{S,P}$ via projection:

$$\mathcal{F}_{S,P}(t, m_\ell, m_A) = \frac{1}{2(p' \pm p)^2} \mathrm{Tr}\left[\Lambda_{S,P} \cdot (\not{p} \pm m_A) \cdot \Gamma_{S,P} \cdot (\not{p}' \pm m_A)\right]\,. \tag{3.8}$$

The signs are $(+1)$ and (-1) for the scalar (S) and pseudo-scalar (P) cases respectively. These form factors can in turn be expanded as sums over scalar FIs:

$$\mathcal{F}_{S,P} = \sum_{\vec{\nu}} c^{\vec{\nu}}_{S,P}(t, m_\ell, m_A|\epsilon) \times I_{\vec{\nu}}(t, m_\ell, m_A|\epsilon), \quad \vec{\nu} \in \mathbb{N}^7\,. \tag{3.9}$$

[2] The expression (3.3) therefore vanishes when A is the neutron. This is an approximation which holds only in the non-relativistic limit, which is the regime being probed by DM experiments. See [73] for more details.

The integral family $I_{\vec{\nu}}$ is given by

$$I_{\nu_1,\ldots,\nu_7}(t, m_\ell, m_A|\epsilon) = \int \frac{\mathrm{d}^{\mathrm{D}}\ell_1}{(2\pi)^{\mathrm{D}}} \frac{\mathrm{d}^{\mathrm{D}}\ell_2}{(2\pi)^{\mathrm{D}}} \frac{D_7^{\nu_7}}{D_1^{\nu_1}\cdots D_6^{\nu_6}}\,, \tag{3.10}$$

with denominators defined in (3.6) and an ISP chosen to be

$$D_7 = (\ell_1 - p)^2\,. \tag{3.11}$$

The coefficients $c^{\vec{\nu}}_{S,P}$ are rational functions of the kinematic variables and the DR parameter ϵ, and stem from carrying out the Dirac algebra[3].

It turns out that $\mathcal{F}_S$ is a sum of 34 integrals, and $\mathcal{F}_P$ a sum of 5 integrals. The 34 integrals are not all linearly independent due to IBP relations. The goal for the rest of this chapter is to compute a linearly independent set of MIs, which in turn leads to expressions for the form factors $\mathcal{F}_{S,P}$ via (3.9).

Section 3.3

Computing the Feynman integrals

Although the diagrams in Figure (3.1) are finite in $\mathrm{D} = 4$ dimensions, we treat them in DR in order to benefit from IBPs and DEQs. The limit $\epsilon \to 0$ in (3.9) then recovers the desired form factor results.

For the purpose of verification, we compute MIs for both the two-mass ($m_A \neq m_\ell$) and equal-mass ($m_A = m_\ell$) cases. The *ab initio* equal-mass calculation can then be compared with the limit $m_A \to m_\ell$ taken on the two-mass form factors.

3.3.1 Two-mass case

We begin by computing MIs for the family of FIs associated to Figure (3.2). The integral family was defined in the formulas (3.6), (3.10) and (3.11). As before, the kinematics are $p^2 = p'^2 = m_A^2$ and $t = (p - p')^2$. MIs for this topology have already been computed in [75] in the context of $H \to \bar{b}\,b$ decay; here we re-compute them independently.

IBP reduction and differentiation of FIs is performed with the codes Reduze2 [12] and LiteRed [15]. Using the Magnus expansion from Section (2.3.2), the following set of canonical MIs are identified:

[3] The prescription for treating γ_5 in $\mathrm{D} \neq 4$ dimensions is taken from [74].

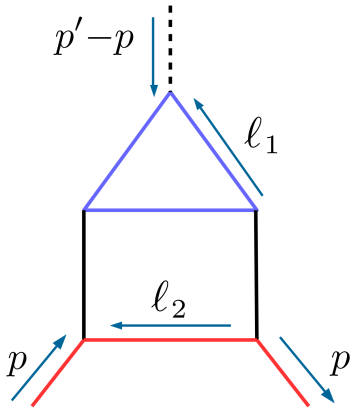


Figure 3.2: Two-mass Feynman diagram.

$$\begin{array}{lllllll}
I_1 & = & \epsilon^2 J_1 & I_2 & = & \epsilon^2 J_2 \\
I_3 & = & \epsilon^2 \lambda_\ell J_3 & I_4 & = & -\epsilon^2 t J_4 \\
I_5 & = & \epsilon^2 \left[\frac{\lambda_\ell - t}{2} J_4 + \lambda_\ell J_4\right] & I_6 & = & -\epsilon^2 t J_6 \\
I_7 & = & \frac{\epsilon^2 m_A^2 \rho_A (t+\lambda_\ell)}{\rho_\ell (t+\lambda_A)} [J_7 + 2J_8] & I_8 & = & \epsilon^2 m_A^2 J_8 \\
I_9 & = & \epsilon^2 \lambda_\ell J_9 & I_{10} & = & -\epsilon^2 t \lambda_\ell J_{10} \\
I_{11} & = & \epsilon^3 \lambda_A J_{11} & I_{12} & = & \frac{\epsilon^2 \lambda_\ell}{4t} \left[(t-\lambda_A)(J_4 + 2J_5) - 4m_A^2 \lambda_A J_{12}\right] \\
I_{13} & = & \epsilon^3 \lambda_A J_{13} & I_{14} & = & \epsilon^3 (2\epsilon - 1) t J_{14} \\
I_{15} & = & \epsilon^3 \lambda_\ell \lambda_A J_{15} & I_{16} & = & \epsilon^3 \lambda_A J_{16} \\
I_{17} & = & \epsilon^3 \lambda_A J_{17} & I_{18} & = & (3.13) \\
I_{19} & = & (3.14) & I_{20} & = & -\epsilon^4 t \lambda_A J_{20} \,,
\end{array} \tag{3.12}$$

where

$$I_{18} = \frac{\epsilon^2}{t} \left[\lambda_\ell (\lambda_A - t) J_9 + \epsilon \lambda_A (t - \lambda_\ell) J_{17} + (2\epsilon - 1) \lambda_\ell \lambda_A J_{18}\right] \tag{3.13}$$

and

$$\begin{aligned} I_{19} = \frac{\epsilon^2}{2t} \Big[& t(\lambda_\ell - t) J_3 - 2t m_\ell^2 (J_7 + 2J_8) + \big(4t m_\ell^2 + \lambda_\ell (\lambda_A - t)\big) J_9 + \frac{4t^2 m_A^2}{\lambda_A + t} J_{16} \\ & + \epsilon \big(\lambda_A (t - \lambda_\ell) 4t m_\ell^2\big) J_{17} + (2\epsilon - 1)(4t m_\ell^2 + \lambda_\ell \lambda_A - t^2) J_{18} + 2t^2 (m_\ell^2 - m_A^2) J_{19} \Big] \,. \end{aligned} \tag{3.14}$$

The integrals J_i are graphically represented in Figure (3.3).

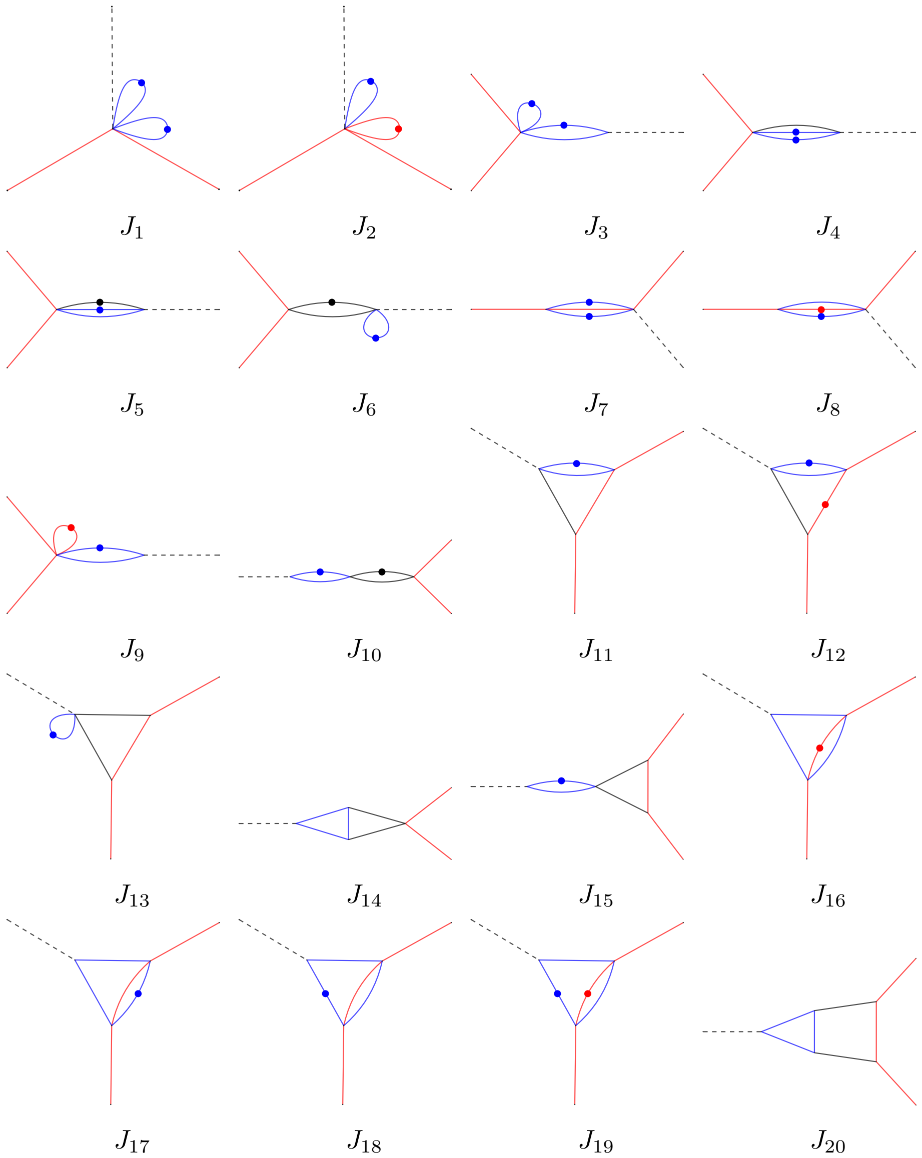


Figure 3.3: The MIs J_i appearing in the canonical basis (3.12). Squared propagators are represented by dots.

The canonical integrals contain square root prefactors λ_i and ρ_i defined by

$$\lambda_i = \sqrt{-t}\sqrt{4m_i^2 - t}\,, \quad \rho_i = \sqrt{\frac{2m_i^2 - t - \lambda_i}{m_i^2}}\,, \quad i = \ell, A\,. \tag{3.15}$$

These roots are simultaneously rationalized by the change of variables

$$t = -m_\ell^2 \frac{(1-x)^2}{x^2}\,, \quad m_A^2 = m_\ell^2 \frac{(1-x^2)^2 y^2}{(1-y^2)^2 x^2}\,, \tag{3.16}$$

with a corresponding inverse map

$$x = \frac{\sqrt{4-\sigma_\ell} - \sqrt{-\sigma_\ell}}{2}\,, \quad y = \frac{\sqrt{4-\sigma_A} - \sqrt{-\sigma_A}}{2}\,, \quad \sigma_i = \frac{t}{m_i^2}\,. \tag{3.17}$$

The Euclidean kinematic regime[4]

$$\big\{t < 0\big\} \quad \bigcap \quad \big\{0 < m_A^2 < m_\ell^2 \quad \text{or} \quad 0 < m_\ell^2 < m_A^2\big\} \tag{3.18}$$

is parametrized by

$$\big\{0 < x < 1\big\} \quad \bigcap \quad \big\{0 < y < x \quad \text{or} \quad 0 < x < y\big\} \tag{3.19}$$

in the (x, y) coordinates.

The basis (3.12) satisfies a canonical Pfaffian system

$$\mathrm{d}\vec{I}(x,y|\epsilon) = \epsilon\, P(x,y) \cdot \vec{I}(x,y|\epsilon)\,, \quad P(x,y) = \sum_{i=1}^{12} P_i \,\mathrm{d}\log \eta_i(x,y)\,, \quad P_i \in \mathbb{Q}^{20\times 20} \tag{3.20}$$

with letters

$$\begin{aligned} \eta_1 &= x\,, & \eta_2 &= 1+x\,, & \eta_3 &= 1-x\,, \\ \eta_4 &= 1+x^2\,, & \eta_5 &= y\,, & \eta_6 &= 1+y\,, \\ \eta_7 &= 1-y\,, & \eta_8 &= 1+y^2\,, & \eta_9 &= x+y\,, \\ \eta_{10} &= x-y\,, & \eta_{11} &= 1+xy\,, & \eta_{12} &= 1-xy\,. \end{aligned} \tag{3.21}$$

The general solution to this Pfaffian system is given by the path-ordered exponential (2.46). The result takes the form $\vec{I}(x,y|\epsilon) = \left[\sum_{i=0}^{4} M_i(x,y)\,\epsilon^i + O(\epsilon^5)\right] \cdot \vec{I}_0(\epsilon)\,,$ where the matrices $M_i(x,y)$ contain GPLs built from the letters η_i, and $\vec{I}_0(\epsilon)$ is a vector of boundary constants. The latter is fixed as follows:

[4] It is standard to solve Pfaffian systems for Euclidean kinematics and then analytically continue the resulting GPLs to physical kinematic regimes.

- By normalizing the integration measure as

$$\left(\frac{m_\ell^2}{\mu^2}\right)^\epsilon \int \frac{\mathrm{d}^{\mathrm{D}}\ell_i}{i\pi^{\mathrm{D}/2}\Gamma(1+\epsilon)}\,, \quad i=1,2\,, \tag{3.22}$$

  then direct integration (via Feynman parametrization) of the tadpoles $\{I_1,\, I_2\}$ as well as the factorized integral I_6 gives

$$\begin{aligned} I_1 &= 1 \\ I_2 &= \left(\frac{(1-y^2)x^2}{(1-x^2)y^2}\right)^\epsilon \\ I_6 &= \left(\frac{x^2}{(1-x^2)^2}\right)^\epsilon \left[1-\zeta_2\epsilon^2-2\zeta_3\epsilon^3-\frac{9}{4}\zeta_4\epsilon^4+O(\epsilon^5)\right]\,, \end{aligned} \tag{3.23}$$

  where ζ_n is the nth Riemann zeta value. These external inputs are fed into the DEQ to give relations among the remaining boundary constants.

- Boundary constants for the set of integrals $\{I_3, I_4, I_5, I_9, I_{10}, I_{11}, I_{12}, I_{14}, I_{16}, I_{17}, I_{18}, I_{19}\}$ are fixed by requiring regularity at the pseudo-threshold limit $t \to 0$. In particular for $\{I_3, I_4, I_5, I_9, I_{10}, I_{14}, I_{16}, I_{17}\}$, the prefactors in (3.12) make them vanish in this limit[5]. Further, $\{I_{11}, I_{12}, I_{18}, I_{19}\}$ are seen to vanish as $t \to 0$ by analyzing their DEQs. Finally, the factorized double-bubble integral I_{10} vanishes in the limit because the *massive* bubble factor (drawn in blue in Figure (3.3)) goes to 0 at a faster rate than the divergence of the *massless* bubble factor (drawn in black).

- Boundary constants for $\{I_7, I_8\}$ are determined by regularity at $m_A^2 \to 0$. Due to the prefactors in (3.12), both integrals vanish in this limit.

- Regularity at the limit $t \to 4m_A^2$ fixes boundary constants for the final three integrals $\{I_{13}, I_{15}, I_{20}\}$. Thanks to the prefactors of (3.12), they all vanish in this limit.

- In some of the cases listed above, the equations for the boundary constants only led to solutions at very high precision, as opposed to analytic expressions in terms of transcendental constants. Fortunately, using the implementation of the `PSLQ` algorithm [77] in PolyLogTools [67], we can reconstruct the analytic constants by fitting to an ansatz.

[5] The vanishing of I_{14} as $t \to 0$ follows from the results of [76].

3.3.2 Equal-mass case

The equal-mass topology is shown in Figure (3.4). Its associated FI family is given by (3.10) upon setting $m_A^2 = m_\ell^2 = m^2$ in (3.6) and (3.11).

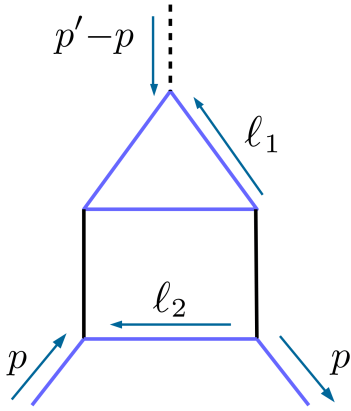


Figure 3.4: Equal-mass Feynman diagram.

IBP reduction reveals 15 MIs. The Magnus expansion then yields the canonical basis

$$\begin{array}{lcl lcl}
I_1 & = & \epsilon^2 J_1 & I_2 & = & \epsilon^2 \lambda_m J_2 \\
I_3 & = & -\epsilon^2 t J_3 & I_4 & = & \epsilon^2 \left[\frac{\lambda_m + t}{2} J_3 + \lambda_m J_4\right] \\
I_5 & = & -\epsilon^2 t J_5 & I_6 & = & \epsilon^2 m^2 J_6 \\
I_7 & = & -\epsilon^2 t \lambda_m J_7 & I_8 & = & \epsilon^3 \lambda_m J_8 \\
I_9 & = & \epsilon^2 \left[\frac{4m^2 - \lambda_m - t}{4}(J_3 + 2J_4) + \ m^2(4m^2 - t) J_9\right] & I_{10} & = & \epsilon^3 \lambda_m J_{10} \\
I_{11} & = & \epsilon^3 (1 - 2\epsilon) t J_{11} & I_{12} & = & \epsilon^3 t (t - 4m^2) J_{12} \\
I_{13} & = & \epsilon^3 \lambda_m J_{13} & I_{14} & = & (3.25) \\
I_{15} & = & -\epsilon^4 \lambda_m t J_{15}\,, & & &
\end{array} \tag{3.24}$$

with

$$I_{14} = \epsilon^2 \left[(4m^2 - \lambda_m - t)(J_2 - \epsilon J_{13}) + (2\epsilon - 1)(4m^2 - t) J_{14}\right] . \tag{3.25}$$

Diagrams for the integrals J_i can be seen in Figure (3.5).

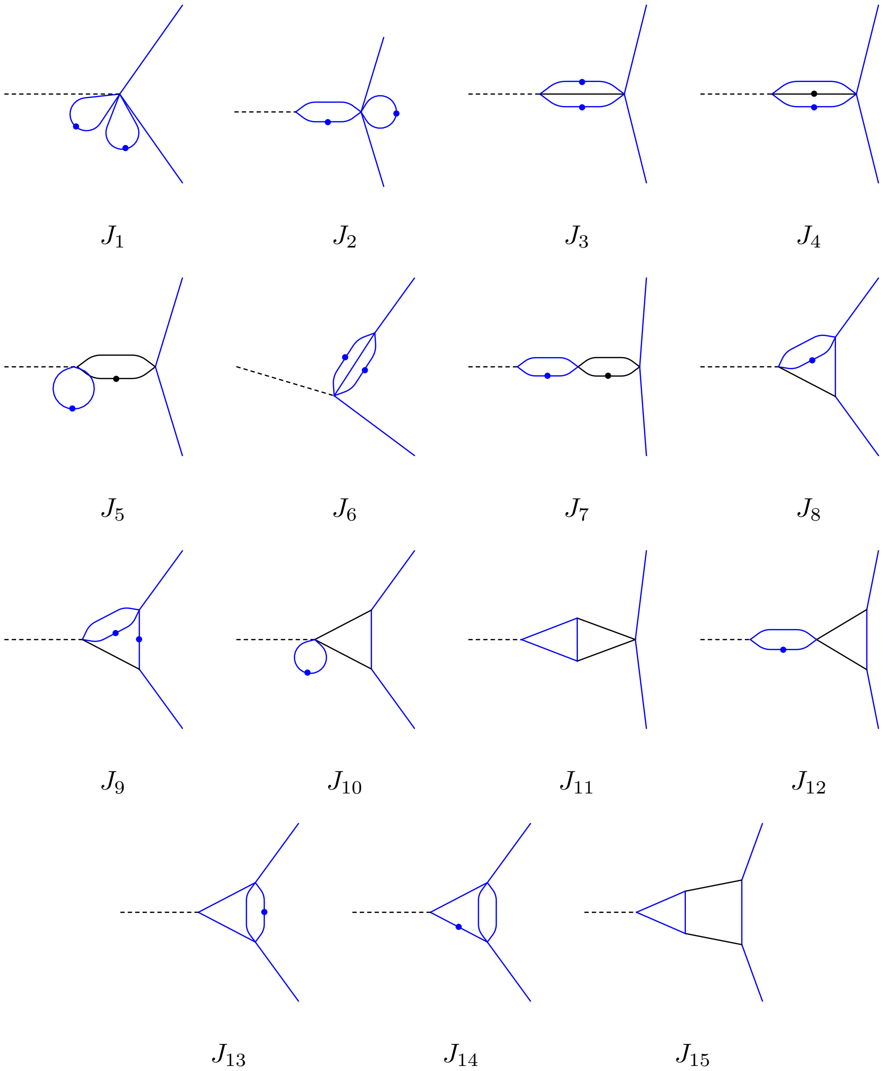


Figure 3.5: The MIs J_i appearing in the canonical basis (3.24). Squared propagators are represented by dots.

The canonical integrals contain an algebraic function λ_m given by

$$\lambda_m = \sqrt{-t}\sqrt{4m^2 - t}\,. \tag{3.26}$$

Both of these roots are simultaneously rationalized by the coordinate change

$$t = -m^2 \frac{(1-w)^2}{w}\,, \tag{3.27}$$

whose inverse map is given by

$$w = \frac{\sqrt{4m^2 - t} - \sqrt{-t}}{\sqrt{4m^2 - t} + \sqrt{-t}} \,. \tag{3.28}$$

The Euclidean kinematic regime

$$\{t < 0\} \quad \bigcap \quad \{m^2 > 0\} \tag{3.29}$$

is then given by

$$0 < w < 1 \,. \tag{3.30}$$

The canonical Pfaffian system for the basis (3.24) takes the form

$$\mathrm{d}\vec{I}(w,\epsilon) = \epsilon\, P(w) \cdot \vec{I}(w,\epsilon) \,, \quad P(w) = \sum_{i=1}^{3} P_i \, \mathrm{d}\log \eta_i(w) \,, \quad P_i \in \mathbb{Q}^{15\times 15} \,, \tag{3.31}$$

where the three letters are

$$\eta_1 = w \,, \quad \eta_2 = 1 + w \,, \quad \eta_3 = 1 - w \,. \tag{3.32}$$

This means that the Pfaffian system has a solution in terms *harmonic polylogarithms* (HPLs) denoted by $H(\vec{z}|w)$, i.e. GPLs with weights $z_i \in \{-1, 0, 1\}$ [78].

Boundary constants are determined as follows:

- The integrals $\{I_1, I_5, I_6\}$ can be directly integrated, and are provided as external input. The first integral is normalized to unity by the measure, and the second integral is swiftly computed with Feynman parameters, giving

$$I_1 = 1 \tag{3.33}$$

$$I_5 = \left(\frac{w}{(1-w)^2}\right)^{\epsilon} \left[1 - \zeta_2 \epsilon^2 - 2\zeta_3 \epsilon^3 - \frac{9}{4}\zeta_4 \epsilon^4 + O(\epsilon^5)\right] . \tag{3.34}$$

  The equal-mass sunrise integral I_6 is computed by following the strategy of [79]. Since the square of its external momentum equals its internal masses, $p^2 = m^2$, this integral evaluates to a number rather than a function, wherefore it cannot immediately be computed from a Pfaffian system. The trick is to change two of the propagator masses to M^2 rather than m^2, and then solve a Pfaffian system in the auxiliary variable $x = m^2/M^2$. Boundary constants are fixed by regularity at

$x = 0$. The desired integral is obtained by setting $x = 1$, giving the result

$$I_6 = -\frac{\zeta_2}{2}\epsilon^2 + \left[3\zeta_2\log(2) - \frac{7}{4}\zeta_3\right]\epsilon^3 + \left[\frac{31}{4}\zeta_4 - 6\zeta_2\log^2(2) - \frac{\log^4(2)}{2} - 12\mathrm{Li}_4(\tfrac{1}{2})\right]\epsilon^4 + O(\epsilon^5)\,. \tag{3.35}$$

- Boundary values for the integrals $\{I_2, I_3, I_4, I_7, I_8, I_9, I_{11}, I_{13}, I_{14}\}$ are fixed by regularity at the pseudo-threshold $t \to 0$. In particular, due to the prefactors in (3.24), the integrals $\{I_2, I_3, I_4, I_8, I_{11}, I_{13}\}$ go to 0 in this limit. By analyzing the Pfaffian system as $t \to 0$, we obtain the relations $I_9|_{t\to 0} = 3I_6$ and $I_{14}|_{t\to 0} = 6I_6$, which fixes boundary values for I_9 and I_{14} because I_6 is known by the formula (3.35). The integral I_7 vanishes because the *massive* canonical bubble factor goes to zero faster than the divergence of *massless* bubble factor.

- For I_{10}, we use regularity at the pseudo-threshold $t \to 4m^2$. Thanks to the prefactor in (3.24), the integral vanishes in this limit.

- Boundary constants for the integral I_{12} follow from comparing the left- and right-hand sides of $I_{12} = I_2 \times I_{10}$, since I_2 and I_{10} are known. This relation can be gleaned from Figure (3.24) by recalling that the double-tadpole I_1 is normalized to unity due to the integration measure.

- We are able to determine boundary constants for the top sector integral I_{15} by comparing our result with [80].

The final results for the canonical MIs are quite compact. For instance, the top sector integral reads

$$I_{15} = \epsilon^4\Big[2\zeta_3 H(0|w) + 4\zeta_2 H(0,0|w) + 8H(0,0,-1,0|w) + H(0,0,0,0|w) + 6H(0,0,0,1|w) + 4H(0,1,0,0|w) + 3\zeta_4\Big] + O(\epsilon^5)\,. \tag{3.36}$$

3.3.3 Soft limit

The soft limit $t \to 0$ of the form factors in Section (3.4.3) is of phenomenological interest. For this reason, we now compute the two-mass MIs in this limit. The original two-mass topology simplifies in this limit, as depicted in Figure (3.6).

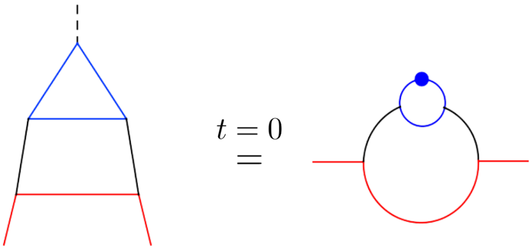


Figure 3.6: Diagrammatic representation of the soft limit.

The reason for this simplification is as follows. Recall that $t = q^2$, where q^μ is the momentum of the mediator particle. Setting $q^\mu = 0$ in the integral family (3.10) implies that $D_2 \to D_1$ and $D_5 \to D_4$, wherefore we only need to consider IBPs and DEQs for the simplified integral family

$$I_{\nu_1\nu_3\nu_4\nu_6\nu_7} = \int \frac{\mathrm{d}^{\mathrm{D}}\ell_1}{(2\pi)^{\mathrm{D}}} \frac{\mathrm{d}^{\mathrm{D}}\ell_2}{(2\pi)^{\mathrm{D}}} \frac{D_7^{\nu_7}}{D_1^{\nu_1} D_3^{\nu_3} D_4^{\nu_4} D_6^{\nu_6}} \tag{3.37}$$

having four denominators

$$\begin{aligned} D_1 &= \ell_1^2 - m_\ell^2\,, & D_3 &= \ell_2^2 - m_A^2\,, \\ D_4 &= (\ell_2 + p)^2\,, & D_6 &= (\ell_1 + \ell_2 + p)^2 - m_\ell^2\,, \end{aligned} \tag{3.38}$$

and an ISP $D_7 = (\ell_1 - p)^2$. Note that p' has been set equal to p due to momentum conservation.

IBP reduction for the family (3.37) reveals 4 MIs. By the Magnus exponential, we obtain the canonical basis

$$\begin{aligned} I_1 &= \epsilon^2 J_1\,, & I_2 &= \epsilon^2 J_2\,, \\ I_3 &= \epsilon^2 m_\ell m_A \big[J_3 + 2J_4\big]\,, & I_4 &= \epsilon^2 m_A^2 J_4\,, \end{aligned} \tag{3.39}$$

in terms of the integrals $\{J_1, J_2, J_3, J_4\}$ shown in Figure (3.7).

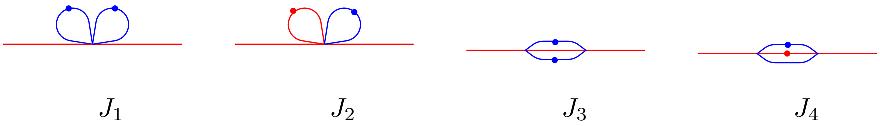


Figure 3.7: The integrals J_1 appearing in the canonical basis (3.39). Squared propagators are depicted by dots.

Defining the mass-ratio variable

$$z = \frac{m_A}{m_\ell}\,, \tag{3.40}$$

the canonical Pfaffian system takes the form

$$\mathrm{d}\vec{I}(z|\epsilon) = \epsilon\, P(z)\cdot\vec{I}(z|\epsilon)\,, \quad P(z) = \sum_{i=1}^{3} P_i\,\mathrm{d}\log\eta_i(z)\,, \quad P_i \in \mathbb{Q}^{4\times 4}\,, \tag{3.41}$$

where the η_i are HPL letters

$$\eta_1 = z\,, \quad \eta_2 = 1+z\,, \quad \eta_3 = 1-z\,. \tag{3.42}$$

Boundary constants for the solution vector are fixed as follows:

- The two double-tadpoles $\{I_1, I_2\}$ can be directly integrated to

$$I_1 = 1 \quad \text{and} \quad I_2 = z^{-2\epsilon}\,. \tag{3.43}$$

- Boundary values for the two sunrises $\{I_3, I_4\}$ are fixed by regularity at $m_A \to 0$ as in [79]. In particular, due to the prefactors of (3.39), both integrals vanish in this limit.

Section 3.4

Form factor results

We now present form factor results for the following cases:

- Two masses ($m_A \neq m_\ell$): scalar and pseudo-scalar.
- Equal-mass ($m_A = m_\ell$): scalar and pseudo-scalar.
- Soft limit ($t \to 0$): scalar and pseudo-scalar.

Let us emphasize again that we only include results for one-body interactions. The analytic form factors in this section are all rescaled as $\mathcal{F}_{S,P} \to \frac{\pi^2}{\alpha_{\text{em}}}\mathcal{F}_{S,P}$ for the sake of readability.

3.4.1 Two-mass case

The analytic expressions for the two-mass form factors $\mathcal{F}_{S,P}(t, m_\ell, m_A)$ would span several pages, so we omit them here. Instead, we showcase the results as numerical plots in Figures (3.8a) and (3.8b).

3.4.2 Equal-mass case

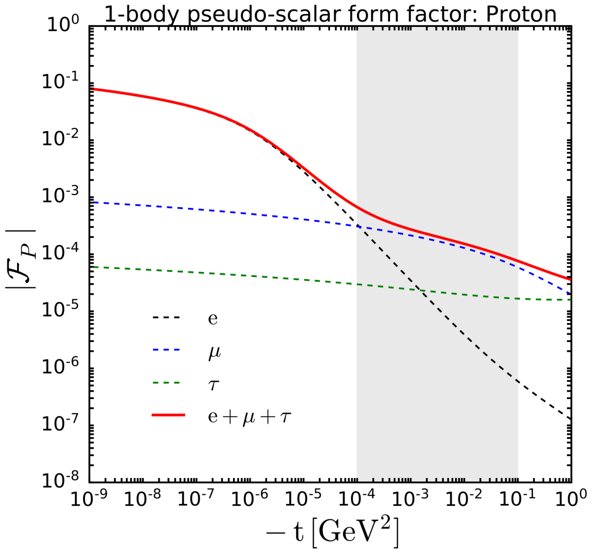
Define the shorthand notation $G(\vec{z}|w) := G_{\vec{z}}(w)$.

Scalar form factor

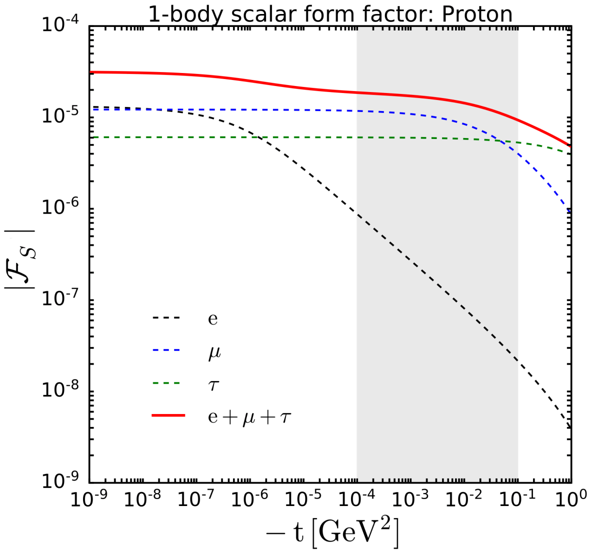
The analytic expression for the equal-mass form factor with a scalar mediator particle is

$$
\begin{aligned}
&\mathcal{F}_S(w) = -\frac{(20\,w^2+4\,w)}{(w-1)\,(w+1)^2}\,G_0(w) + \frac{w}{(w-1)\,(w+1)}\,\left[24\,G_{-1,0}(w)+4\,G_{1,0}(w)\right]\\
&+\frac{w}{(w-1)^2\,(w+1)^3}[-12\,w^3-38\,w^2+8\,w+2]\,G_{0,0}(w) - \frac{w}{\pi^2\,(w-1)^2}\,\left[2\,G_{0,0,0}(w)+4\,\zeta_3\right]\\
&+\frac{w\,\zeta_3}{(w-1)^3\,(w+1)}\,[5\,w^2-6\,w+5]\,G_0(w) + \frac{w}{\pi^2\,(w+1)^2}\,\left[12\,G_{0,0,1}(w)-\frac{4\,\pi^2}{3}\,G_0(w)+16\,G_1(w)\right]\\
&+\frac{w}{(w-1)^2\,(w+1)^2}\,\left[(-16\,w^2+64\,w-16)\,G_{0,-1,0}(w)+(8\,w^2-48\,w+8)\,G_{1,0,0}(w)+16\,w\,G_{0,1,0}(w)\right]\\
&+\frac{w}{(w-1)\,(w+1)^3}\left[(-14\,w^2+4\,w-14)\,G_{0,1}(w)+\left(\frac{w^2}{3}+2\,w+\frac{1}{3}\right)\,G_{1,0}(w)\,\pi^2\right.\\
&\left.+(2\,w^2+12\,w+2)\,\left(G_{1,0,0,0}(w)-G_{0,0,1,0}(w)\right)+\left(2\,w^2-\frac{20\,w}{3}+2\right)\,\pi^2\right]\\
&+\frac{w}{(w-1)^3\,(w+1)^3}\left[\left(\frac{5\,w^4}{6}-2\,w^3+\frac{23\,w^2}{3}-2\,w+\frac{5}{6}\right)\,G_{0,0}(w)\,\pi^2\right.\\
&+\left(8\,w^4-32\,w^3+112\,w^2-32\,w+8\right)\,G_{0,0,-1,0}(w)+\left(\frac{3\,w^4}{2}-2\,w^3+9\,w^2-2\,w+\frac{3}{2}\right)\,G_{0,0,0,0}(w)\\
&+\left(-9\,w^4+12\,w^3-54\,w^2+12\,w-9\right)\,G_{0,0,0,1}(w)+\left(-2\,w^4+24\,w^3-76\,w^2+24\,w-2\right)\,G_{0,1,0,0}(w)\\
&\left.+\left(\frac{w^4}{18}-\frac{2\,w^3}{45}+\frac{11\,w^2}{45}-\frac{2\,w}{45}+\frac{1}{18}\right)\,\pi^4\right].
\end{aligned}
\tag{3.44}
$$

This expression can be turned into an s-channel form factor by crossing symmetry. Then it is related to a form factor for Higgs decay into two heavy quarks [81].



(a) Two-mass **pseudoscalar** form factor as a function of momentum transfer t. The nucleon mass m_A is set equal to the proton mass. The dashed lines represent individual contributions from the electron (e), the muon (μ), and the tau lepton (τ). The solid red line shows the sum of all contributions. The grey shaded band indicates the range of t that is expected to be relevant for DM particles arriving from our local galaxy.



(b) Two-mass **scalar** form factor as a function of momentum transfer t. See the caption of Figure (3.8a) for additional details.

Pseudo-scalar form factor

The equal-mass form factor for the pseudoscalar reads

$$\begin{aligned}\mathcal{F}_P(w) = &-\frac{w^2}{(w-1)^3\,(w+1)}\left[\frac{2\pi^2}{3}\,G_0(w)+4\,G_{0,0,0}(w)\right]\\ &+\frac{w}{(w-1)^2}\left[G_{0,0}(w)-4\,G_{0,0,1}(w)-4\,G_{0,1,0}(w)+8\,G_{1,0,0}(w)+12\,\zeta_3+\frac{\pi^2}{3}\right]\\ &+\frac{w}{(w-1)\,(w+1)}\left[-3\,\zeta_3\,G_0(w)-\frac{\pi^2}{6}\,G_{0,0}(w)-\frac{\pi^2}{3}\,G_{1,0}(w)-\frac{1}{2}\,G_{0,0,0,0}(w)\right.\\ &\qquad\left.+3\,G_{0,0,0,1}(w)+2\,G_{0,0,1,0}(w)-2\,G_{0,1,0,0}(w)-2\,G_{1,0,0,0}(w)-\frac{\pi^4}{45}\right].\end{aligned} \tag{3.45}$$

Using crossing symmetry, this form factor is related to the s-channel QCD form factor dubbed $A_R(s,m,m)$ in [74].

3.4.3 Soft limit

The soft limit $t \to 0$ is important for phenomenological applications, as the DM particles are expected to scatter elastically. It is therefore useful to have simple expressions for the form factors in this limit that can be easily numerically evaluated.

Scalar form factor

The scalar form factor is finite at $t = 0$, so the result $\mathcal{F}_S(t = 0, m_\ell, m_A)$ will only depend on the mass ratio $z = m_A/m_\ell$. Writing the form factor as an expansion in terms of FIs as in (3.9), we then IBP reduce those integrals to the soft limit MIs from Section (3.3.3). The smooth 4-dimensional limit $\epsilon \to 0$ finally yields the result

$$\begin{aligned}\mathcal{F}_S(t=0,m_\ell,m_A) &= -\frac{2}{z}\left[1-\frac{\log(z)}{2}+f_S(z)+f_S(-z)\right] \\ \text{with}\quad f_S(z) &= \frac{4+3z+z^3}{4z^2}\big[\log|z|\log(1+z)+\mathrm{Li}_2(-z)\big]\,.\end{aligned} \tag{3.46}$$

Pseudo-scalar form factor

The pseudo-scalar form factor is logarithmically divergent as $t \to 0$, so we compute an asymptotic expansion in terms of the variables (x, y) from (3.16). Keeping $m_{A,\ell}$ fixed, the soft limit then corresponds to $(x, y) \to (1, 1)$.

Our strategy for computing the asymptotic expansion is as follows:

1. Staring with the full expression for the form factor $\mathcal{F}_P(t, m_\ell, m_A)$ in terms of GPLs, we use PolyLogTools to rewrite every $G(\vec{z}\,|\,v)$ such that $\vec{z} \in \mathbb{Z}^M$ and $v = x$ or $v = y$. The weight vectors $\vec{z}$ will thereby not contain any x's or y's, which makes it easier to expand the GPLs.

2. PolyLogTools has a built-in function for asymptotically expanding GPLs. However, it only supports an expansion around $v \sim 0$, not $v \sim 1$. Introducing a dummy variable v' by $v = 1 - v'$, we hence seek to rewrite every $G(\vec{z}\,|\,v)$ in terms of $G(\vec{z}\,|\,v')$. This is done by employing the well-known recursion

$$\begin{aligned} G(z_1, \ldots, z_M|v) &= G(z_1, \ldots, z_M|1 - v') \\ &= G(z_1, \ldots, z_M|1) + \int_0^{v'} \frac{\mathrm{d}t}{t - (1 - z_1)} G(z_2, \ldots, z_M|1 - t)\,, \end{aligned} \tag{3.47}$$

   which produces GPLs having argument v' due to the upper integration bound in the second term. Experimentally, we find that this recursion admits a closed-form solution

$$\begin{aligned} G(z_1, \ldots, z_M|1 - v') = \sum_{i=0}^{M} & G(1 - z_1, \ldots, 1 - z_i|v') \\ & \times G(z_{i+1}, \ldots, z_M|1)\,\big|_{\text{shuffle reg.}}\,, \end{aligned} \tag{3.48}$$

   where we emphasized the need for *shuffle regularization* to deal with divergent GPLs [61, Section 5.3]. We use the convention $G(1 - z_1, \ldots, 1 - z_i|v') = 1$ when $i = 0$.

   As an aside, although it is not a deep result, we have not been able to locate this closed-form solution elsewhere in the literature.

3. PolyLogTools can now expand the terms $G(1 - z_1, \ldots, 1 - z_i|v')$ from (3.48) around $v' \sim 0$ up the desired order $\mathcal{O}\big((1-x)^n, (1-y)^m\big)$. The latter notation means that we first expand in $(1 - x)$ up to nth order, and then in $(1 - y)$ up to mth order. The reversed expansion order is written as $\mathcal{O}\big((1 - y)^n, (1 - x)^m\big)$. The maximal powers (n, m) are chosen so as to give sufficiently accurate numerical results in comparison with the full form factor $\mathcal{F}_P(t, m_\ell, m_A)$ when the latter is evaluated for small t.

A double asymptotic expansion in (x, y) will depend on the orderings $x < y$ or $y < x$, so we treat these two cases separately. The orderings correspond to the mass hierarchies $m_\ell < m_A$ and $m_A < m_\ell$ respectively. By the steps outlined above, we obtain the following results for the pseudoscalar form factor in the soft limit:

- Mass hierarchy $m_\ell < m_A\,(x < y)$:

$$\begin{aligned}\mathcal{F}_P(t \sim 0,\, m_\ell < m_A) = {}& -2\zeta_2 + \frac{(1-x)}{(1-y)}\big[-2\log(1-x) - 2\log(2) + 3\big] \\ & + \frac{(1-x)^2}{(1-y)}\left[-\log(1-x) - \log(2) + \frac{1}{2}\right] \\ & + (1-x)^2\left[\frac{\log(1-x)}{2} + \frac{\log(2)}{2} + \frac{37\zeta_2}{60} - \frac{1}{4}\right] \\ & + \frac{6(1-x)}{\zeta_2}\left[\log(1-x) + \log(2) + \frac{3\zeta_2}{2} - \frac{3}{2}\right] \\ & + \mathcal{O}\big((1-y),\, (1-x)^3\big)\,. \end{aligned} \tag{3.49}$$

- Mass hierarchy $m_A < m_\ell\,(y < x)$:

$$\begin{aligned}\mathcal{F}_P(t \sim 0,\, m_A < m_\ell) = {}& \frac{(1-x)}{(1-y)}\left[-\frac{3\log(1-x)}{2} - \frac{(1-y)}{2} - 2\log(2) + \frac{17}{8}\right] \\ & + (1-x)\left[\frac{3\log(1-x)}{4} + \frac{\log(1-y)}{4} + \log(2) + \frac{3\zeta_2}{2} - \frac{21}{16}\right] \\ & + \mathcal{O}\big((1-x)^2,\, (1-y)\big)\,. \end{aligned} \tag{3.50}$$

These formulas differ by roughly 1% in comparison with the full analytic results evaluated at $t = -10^{-7}\mathrm{GeV}^2$.

Part II

$\mathcal{D}$-modules

Chapter 4

Introduction to $\mathcal{D}$-modules

The preceding chapters of this thesis dealt with tried and tested methods for evaluating FIs. We now venture into somewhat uncharted territory from the perspective of the physics literature.

Based on the papers [82, 83], the next six chapters of this thesis study FIs within the framework of *$\mathcal{D}$-modules*. A $\mathcal{D}$-module can, informally, be thought of as a vector space of linear differential operators that annihilate a given function. As a supplement to the tools coming from complex *analysis*, $\mathcal{D}$-modules provide *algebraic* machinery to uncover properties of PDEs. For instance, one can learn the dimension of the solution space, construct bases of differential operators, obtain relations between basis elements, and more. We will only scratch the surface of this vast topic, so we refer to the classic textbooks [23] and [24] for exhaustive discussions on $\mathcal{D}$-modules.

Our overarching goal in **Part II** of this thesis is to build a bridge between these two independently well-studied topics, namely FIs and $\mathcal{D}$-modules. But, as it so often happens in blue skies research, practical applications will spring out as well.

Section 4.1

What is a $\mathcal{D}$-module?

How might one represent a function? Take $\sin(x)$ as an example. It can be represented in several ways, each one having different advantages. For instance, it can be analytically represented as a power series, or geometrically via the unit circle. In the philosophy of $\mathcal{D}$-modules, $\sin(x)$ is represented by the set of linear PDEs which annihilate the function; in this case simply $\{\partial^2 + 1\}$.

We note that this philosophy would not work for non-linear PDEs, since a differential operator L cannot be "pulled out" in the form $L \bullet f = 0$ if there is a term like $f(x)^2$.

Example 4.1. A striking example of this philosophy comes from calculating roots of an Nth order polynomial

$$z_0 + z_1 x + z_2 x^2 + \cdots + z_{N-1} x^{N-1} + z_N x^N = (x - r_1) \cdots (x - r_N) \, . \tag{4.1}$$

Is there an equation for the roots $r_k = r_k(z_1, \ldots, z_N)$ in terms of the indeterminate polynomial coefficients z_k? There is, for instance, a well-known formula for $r_{1,2}(z_0, z_1, z_2)$ in the quadratic case $N = 2$.

Alas, Galois theory teaches us that no such formula exists in terms of radicals when $N \geq 5$. But if we settle for an infinite series representation of r_k, then such a formula *does* exist!

Fixing k, it turns out that r_k is annihilated, indeed *represented*, by the following set of partial differential operators:

$$\begin{aligned} &\frac{\partial^2}{\partial z_i \partial z_j} - \frac{\partial^2}{\partial z_k \partial z_l} \quad \text{whenever} \quad i + j = k + l \, , \\ &\sum_{i=0}^{N} i \, z_i \frac{\partial}{\partial z_i} + 1 \qquad \text{and} \qquad \sum_{i=0}^{N} z_i \frac{\partial}{\partial z_i} \, . \end{aligned} \tag{4.2}$$

Many features of polynomial roots can then be learned from studying these differential operators. For one, it is possible to construct power series solutions of the form $r_k = \sum_{i_0 \ldots i_N}^{\infty} f_k(i_0, \ldots, i_N) z_0^{i_0} \cdots z_N^{i_N}$, where the f_k's are various combinations of factorials [84]. ■

So, what is a $\mathcal{D}$-module? It is a module over a ring of $\mathcal{D}$ifferential operators. Let us spell this out more formally.

Let R be a ring with a unit "1", and M an abelian group w.r.t. addition " + ". Given $r \in R$ and $m \in M$, we assume the existence of a map

$$\begin{aligned} \psi &: R \times M \to M \\ \psi(r, m) &= r \bullet m \, , \end{aligned} \tag{4.3}$$

where $r \bullet m$ denotes the *action* of r on m. For instance, if $R = \mathbb{C}$ and $M = \mathbb{C}^n$, then ψ could represent the scalar multiplication of a complex number r on a complex vector m.

Definition 4.1. An abelian group M becomes a *left*[1] *R-module* when the following three axioms are fulfilled[2]:

[1] *Right* R-modules are defined by the same axioms with r's acting from the right, i.e. $m \bullet r$.

[2] We omit " + " subscripts of kind $(r_1 +_R r_2) \bullet m = r_1 \bullet m +_M r_2 \bullet m$ for notational cleanliness.

1. For any $r \in R$ and $m_i \in M$, we have $r \bullet (m_1 + m_2) = r \bullet m_1 + r \bullet m_2$.
2. For any $r_i \in R$ and $m \in M$, we have $(r_1 r_2) \bullet m = r_1 \bullet (r_2 \bullet m)$ and $(r_1 + r_2) \bullet m = r_1 \bullet m + r_2 \bullet m$.
3. For any $m \in M$, we have $1 \bullet m = m$.

If R is a commutative field such as $\mathbb{C}$, then a left R-module is nothing but an R-vector space.

The Nth *Weyl algebra*

$$\mathcal{D}_N = \mathbb{C}[z_1, \ldots, z_N]\langle \partial_1, \ldots, \partial_N \rangle \tag{4.4}$$

is generated by symbols z_i and ∂_i subject to the commutation relations

$$[z_i, z_j] = 0\,, \quad [\partial_i, \partial_j] = 0\,, \quad [\partial_i, z_j] = \delta_{ij}\,. \tag{4.5}$$

We shall often suppress the subscript in $\mathcal{D}_N$ when the number of variables is understood. The square brackets $[\ldots]$ in (4.4) mean that the z_i are thought of as coefficients of the generators ∂_i. Note that the last commutator of (4.5) encodes Leibniz' rule of differentiation when we insert a dummy function $f(z)$: $\frac{\partial}{\partial z}[zf(z)] - z\frac{\partial}{\partial z}f(z) = 1$. We therefore interpret $\mathcal{D}$ as the collection of all differential operators having polynomial coefficients. For instance, $(z_1^2 + z_3 + 1)\partial_1^2 \in \mathcal{D}_3$. We emphasize that $\mathcal{D}$ has the structure of a ring.

We also have the Nth *rational* Weyl algebra

$$\mathcal{R}_N = \mathbb{C}(z_1, \ldots, z_N)\langle \partial_1, \ldots, \partial_N \rangle\,, \tag{4.6}$$

wherein the coefficients are promoted to rational functions. We shall often shorten $\mathcal{R}_N$ to $\mathcal{R}$. In this case, the commutators are extended to

$$[z_i, z_j] = 0\,, \quad [\partial_i, \partial_j] = 0\,, \quad [\partial_i, r(z)] = \frac{\partial r(z)}{\partial z_i}\,, \tag{4.7}$$

for a rational function $r(z) \in \mathbb{C}(z_1, \ldots, z_N)$. The rational Weyl algebra is also endowed with a ring structure.

Definition 4.2. A $\mathcal{D}$-module is a left R-module (see Definition (4.1)) with the specific choice of ring $R = \mathcal{D}$. An $\mathcal{R}$-module is defined similarly for $R = \mathcal{R}$.

We shall be careful in distinguishing between ring *multiplication* and ring *action* " $\bullet$ ", because there is a difference between e.g.

$$\partial_i z_i = z_i \partial_i + 1 \quad \text{and} \quad \partial_i \bullet z_i = 1 \,. \tag{4.8}$$

Definition (4.2) can be instantiated in many different ways because it only specifies the ring R. *Particular* $\mathcal{D}$- and $\mathcal{R}$-modules follow from further specifying the abelian group M and the action map ψ. Let us list some examples.

Example 4.2. The Weyl algebras $\mathcal{D}$ and $\mathcal{R}$ are themselves $\mathcal{D}$- and $\mathcal{R}$-modules respectively. The abelian group M is formed from the addition of differential operators. The algebras acts on themselves according to (4.5) in the case of $\mathcal{D}$ and (4.7) in the case of $\mathcal{R}$, which specifies ψ. ■

Example 4.3. Many function spaces are equipped with the structure of a $\mathcal{D}$-module. Take M to be some suitable space of holomorphic functions on a domain of $\mathbb{C}^n$. Assume that M is closed under differentiation, meaning that if $f(z) \in M$ then also $\frac{\partial f}{\partial z_i} \in M$. The addition of such functions indeed forms an abelian group. The action map $\psi : R \times M \to M$ is defined by

$$\psi(\partial_i, f) = \partial_i \bullet f := \frac{\partial f}{\partial z_i} \,, \quad \psi(z_i, f) = z_i \bullet f := z_i \cdot f \,, \tag{4.9}$$

where " $\cdot$ " in the last equation stands for ordinary scalar multiplication. ■

Example 4.4. The most important example of a $\mathcal{D}$-module (or $\mathcal{R}$-module) for the study of PDEs is arguably that of the *quotient* ring $\mathcal{D}/\mathcal{I}$ (or $\mathcal{R}/\mathcal{I}$), where $\mathcal{I}$ is a so-called *left ideal* to be defined shortly. The abelian group M is given by the addition of operators, and the action is that of the Weyl algebra.

To motivate the notion of a left ideal $\mathcal{I}$, consider a collection of differential operators $\{L_1, L_2, \ldots\}$ that annihilate some function f: $L_1 \bullet f = L_2 \bullet f = \ldots = 0$. Note that we still get zero if we act with any other operator $L \in \mathcal{D}$ from the left: $LL_1 \bullet f = LL_2 \bullet f = \ldots = 0$. The definition[3] of a left ideal $\mathcal{I}$ in the Weyl algebra encapsulates this fact:

$$\mathcal{I} = \langle L_1, L_2, \ldots \rangle = \{LL_1,\, LL_2, \ldots \;|\; \forall\; L \in \mathcal{D}\} \,. \tag{4.10}$$

The notation $\langle L_1, L_2, \ldots \rangle$ means that the L's *generate* $\mathcal{I}$.

[3] This is more of a "physicist's definition". Formally, a left ideal $\mathcal{I}$ of a ring R satisfies 1) it is a subgroup under addition, and 2) for $r \in R$ and $L \in \mathcal{I}$ we have $r\,L \in \mathcal{I}$.

The quotient ring $\mathcal{D}/\mathcal{I}$ is then given by all operators $L \in \mathcal{D}$ modulo the equivalence relation $\sim$ determined by

$$L \sim K \iff L = K + K_{\mathcal{I}} \quad \text{where} \quad K \in \mathcal{D}\,, \; K_{\mathcal{I}} \in \mathcal{I}\,. \tag{4.11}$$

Take, for instance, the left ideal generated by $\mathcal{I} = \langle \partial_1 \partial_2 \rangle$. Define two operators by

$$\begin{aligned} L &= \partial_2 \quad \text{and} \\ K &= \partial_1 z_1 \partial_2 \overset{(4.5)}{=} (1 + z_1 \partial_1)\partial_2 = \partial_2 + z_1 \partial_1 \partial_2\,. \end{aligned} \tag{4.12}$$

L and K represent *the same* element inside $\mathcal{D}_2/\mathcal{I}$ because they only differ by a term $z_1\partial_1\partial_2 \in \mathcal{I}$. We say that L and K lie in the same *equivalence class*, written as $[L] = [K]$.

We shall see that the algebraic structure of the $\mathcal{D}$-module $\mathcal{D}/\mathcal{I}$ carries a wealth of information about the functions that are annihilated by the ideal $\mathcal{I}$. This $\mathcal{D}$-module hence embodies the intuition, argued for in the beginning of this section, that a function can be represented by the differential operators that annihilate it. ■

SECTION 4.2

Holonomic $\mathcal{D}$-modules

A $\mathcal{D}$-module $\mathcal{D}/\mathcal{I}$ has particularly nice features when it is *holonomic.* Holonomic $\mathcal{D}$-modules describe FIs, trigonometric functions, various probability distributions, hypergeometric functions, Bessel functions, and GPLs. It is necessary to introduce a few concepts before we can define the notion of holonomicity (we refer to the seminal text [85] for many more details).

- Using the Weyl algebra commutation relations, any operator $L \in \mathcal{D}$ can be uniquely expressed in the *normally ordered form*

$$L = \sum_{(p,q)\,\in\,\mathrm{supp}(L)} c_{p,q}\, z^p\, \partial^q\,, \quad c_{p,q} \in \mathbb{C}\,. \tag{4.13}$$

  The summation range supp (L) denotes a finite set of non-zero multi-indices $(p, q) = (p_1, \ldots, p_N, q_1, \ldots, q_N)$, and we introduced the multi-index exponent notation

$$z^p = z_1^{p_1} \cdots z_N^{p_N} \quad \text{and} \quad \partial^q = \partial_1^{q_1} \cdots \partial_N^{q_N}\,, \tag{4.14}$$

  which will be used frequently throughout this text.

- For a given a vector $(u,v) \in \mathbb{R}^{2N}$, the (u,v)-*weight* of $z^p\,\partial^q$ is defined by $u\cdot p + v \cdot q$.
- The differential operator $\mathrm{in}_{(u,v)}(L)$ is constructed by only keeping the largest (u,v)-weight terms in L.
- The *initial form* is defined by $\mathrm{in}_{(u,v)}(L)$ for the specific choice $u = (0,\ldots,0) = \mathbf{0}$ and $v = (1,\ldots,1) = \mathbf{1}$. I.e. one assigns weight 0 to all z's, and weight 1 to all ∂'s. The standard notation for the initial form is

$$\mathrm{in}_{(\mathbf{0},\mathbf{1})}(L) = \sum_{\substack{(p,q)\,\in\,\mathrm{supp}(L),\\ |q| \text{ is max}}} c_{p,q}\, z^p\, \xi^q\,, \tag{4.15}$$

where $|q| = q_1 + \ldots + q_N$. The ∂'s have here been replaced by commuting ξ variables, but this replacement may only be done *after* writing L in normally ordered form. As an example, take the normally ordered operator

$$L = z_1^2 + 2\partial_1^2\partial_2 + z_2^3\partial_1\partial_2 + z_1 z_2 \partial_1^3. \tag{4.16}$$

From left to right, the terms have weights $(0,3,2,3)$. So we only keep the 2nd and 4th terms, giving

$$\mathrm{in}_{(\mathbf{0},\mathbf{1})}(L) = 2\xi_1^2\xi_2 + z_1 z_2 \xi_1^3\,. \tag{4.17}$$

- Given a left ideal $\mathcal{I}$ in $\mathcal{D}_N$, we set

$$\mathrm{in}_{(\mathbf{0},\mathbf{1})}(\mathcal{I}) = \mathbb{C}\cdot\{\mathrm{in}_{(\mathbf{0},\mathbf{1})}(L) \mid L \in \mathcal{I}\}\,, \tag{4.18}$$

i.e. the ideal generated by the highest $(\mathbf{0},\mathbf{1})$-weight terms in $\mathcal{I}$. It is a *polynomial* ideal since it is written in terms of commuting variables (z,ξ).

- The *characteristic variety* of an ideal is defined as

$$\mathrm{char}(\mathcal{I}) = \big\{(z,\xi) \mid p(z,\xi) = 0 \;\text{ for all }\; p \in \mathrm{in}_{(\mathbf{0},\mathbf{1})}(\mathcal{I})\big\} \;\subset\; \mathbb{C}^{2N}\,. \tag{4.19}$$

Namely, it the zero-set of all the polynomials in the ideal $\mathrm{in}_{(\mathbf{0},\mathbf{1})}(\mathcal{I})$. The computation of $\mathrm{char}(\mathcal{I})$ typically requires knowing a *Gröbner basis* for $\mathcal{I}$ (to be defined in the next section).

With all of these notions in place, we can finally state

Definition 4.3. An ideal $\mathcal{I}$ is *holonomic* when its characteristic variety $\mathrm{char}(\mathcal{I})$ has dimension[4] N. A $\mathcal{D}$-module $\mathcal{D}/\mathcal{I}$ is holonomic if and only if $\mathcal{I}$ is holonomic[5]. The *holonomic rank* is defined as the following vector space dimension over $\mathbb{C}(z)$:

$$\mathrm{rank}(\mathcal{I}) = \dim_{\mathbb{C}(z)} \big(\mathbb{C}(z)[\xi] \,/\, \mathbb{C}(z)[\xi] \cdot \mathrm{char}(\mathcal{I})\,\big)\,.$$

The above definition of $\mathrm{rank}(\mathcal{I})$ relies on the following important fact: a holonomic $\mathcal{D}$-module carries the structure of a *finite-dimensional* vector space [85, 86]. It follows that there exists a *basis* of differential operators for the vector space $\mathcal{D}/\mathcal{I}$ having exactly $\mathrm{rank}(\mathcal{I})$ many elements. Quite remarkably, $\mathrm{rank}(\mathcal{I})$ also counts the number of independent holomorphic solutions to the system of PDEs defined by $\mathcal{I}$. When $\mathcal{I}$ is the annihilating ideal for a FI, then $\mathrm{rank}(\mathcal{I})$ is indeed the number of MIs!

4.2.1 Regular holonomic

Throughout the rest of this thesis, when we say an ideal $\mathcal{I}$ is holonomic we shall always mean *regular holonomic*. The extra adjective "regular" signifies, loosely speaking, that the solutions to the PDEs dictated by $\mathcal{I}$ only have mild singularities.

Suppose that $z = 0$ is a singularity of a solution $f(z)$ in the univariate case. It is a *regular singularity* if there exists a $k \in \mathbb{Z}_{>0}$ such that $|z|^k f(z) \to 0$ as $z \to 0$ at a fixed angle $\arg(z)$ in the complex plane.

The definition of regular holonomic is much more complicated in the multivariate case. We refer to [85, Section 2.4] [23, Chapter 6] for details. A working definition for our purposes is to say that the solutions to $\mathcal{I}$ admit logarithmic series expansions near the singular hypersurfaces. Equivalently, the Pfaffian system born from $\mathcal{I}$, to be discussed in Section (4.4), will have at most simple poles (higher-order poles can appear, but they are "fake" in the sense they can be turned into simple poles via a gauge transformation).

[4] According to a result called Bernstein's inequality, all connected components of $\mathrm{char}(\mathcal{I})$ have dimension at least N. An ideal is hence holonomic when its characteristic variety has the smallest possible dimension (the maximal dimension would be $2N$).

[5] There is actually an independent definition of holonomicity for $\mathcal{D}$-modules, depending on so-called Bernstein filtrations. The fact that holonomic $\mathcal{D}$-modules coincide with holonomic ideals is then a non-trivial theorem. See [86, Chapter 6] for details.

Section 4.3

Gröbner bases

How does one compute a basis for a holonomic $\mathcal{D}$- or $\mathcal{R}$-module? The answer, as with almost any computational question in commutative and non-commutative algebra, is via *Gröbner bases.* A couple of extra concepts are needed before the definition can be stated.

- Given two monomials ∂^p and ∂^q, a *term order* $\prec$ discerns which of them is the largest one. We refer to [86] for the technical definition, and content ourselves with giving an example, namely the *graded reverse lexicographic order*[6]. Setting $|p| = p_1 + \cdots + p_N$, then $\partial^p \prec \partial^q$ iff

  $$|p| < |q| \quad \text{or} \quad |p| = |q| \text{ and the } \textit{last} \text{ non-zero entry of } q - p \in \mathbb{Z}^N \text{ is } \textit{negative.}$$

  So the monomial with the largest degree wins, unless there is a tie, in which case the one with the smallest variable index wins. In the case of $N = 3$ variables and degree less than 3, we have

  $$1 \prec \partial_3 \prec \partial_2 \prec \partial_1 \prec \partial_3^2 \prec \partial_2\partial_3 \prec \partial_1\partial_3 \prec \partial_2^2 \prec \partial_1\partial_2 \prec \partial_1^2 \,. \tag{4.20}$$

- Consider an operator $L = \sum_{q\in\mathrm{supp}(L)} c_q(z)\, \partial^q \in \mathcal{R}$ in normally ordered form. Fixing a term order, the *initial monomial*[7] $\mathrm{in}_\prec(L)$ is defined as the largest monomial ∂^q (stripped of its rational function $c_q(z) \in \mathbb{C}(z)$) w.r.t. $\prec$.

- The *initial ideal* of $\mathcal{I}$ is the ideal generated by its initial monomials:

  $$\mathrm{in}_\prec(\mathcal{I}) = \mathbb{C} \cdot \big\{\mathrm{in}_\prec(L) \mid L \in \mathcal{I}\big\} \ \subset\ \mathbb{C}[\partial_1, \ldots, \partial_N] \,. \tag{4.21}$$

Definition 4.4. A finite subset $G = \{g_1, \ldots, g_M\} \in \mathcal{R}$ is called a *Gröbner basis* for the ideal $\mathcal{I}$ w.r.t. the term order $\prec$ iff

1. G generates $\mathcal{I}$, i.e. $\langle g_1, \ldots, g_M\rangle = \mathcal{I}$.

2. $\mathrm{in}_\prec(\mathcal{I}) = \langle \mathrm{in}_\prec(g_1), \ldots, \mathrm{in}_\prec(g_M)\rangle$.

[6] The speed with which Gröbner bases are calculated can heavily depend on the chosen term order (of which there are many). The graded reverse lexicographic order tends to be the fastest.

[7] The similarity between this notation and $\mathrm{in}_{(u,v)}$ from (4.15) is not coincidental. There is indeed a strong connection between comparing monomials according to weights $(u, v) \in \mathbb{R}^{2N}$ and according to term orders $\prec$ [85, Chapter 1].

A Gröbner basis containing the minimal number of generators g_i is called *reduced*, and is unique for a given term order. A monomial ∂^q is called a *standard monomial* w.r.t. G when it does *not* belong to $\mathrm{in}_\prec(\mathcal{I})$.

Absent any other methods, the standard monomials only become known at the end of a Gröbner basis computation.

We can now answer the question posed in the very beginning of this section: there are exactly $\mathrm{rank}(\mathcal{I})$ many standard monomials, and they form an operator basis for the $\mathcal{D}$-module $\mathcal{D}/\mathcal{I}$!

Gröbner basis computations are typically performed using Buchberger's algorithm[8] [86]. The intensity of such calculations grows significantly with the number of variables. For this reason, the main theme of the next chapter will be to develop a more efficient algorithm that avoids Gröbner bases.

Section 4.4

Pfaffian systems

Let $\mathcal{I}$ be a holonomic ideal born from a system of PDEs, and suppose $f = f(z)$ is a solution to this system. Assume a basis of standard monomials is known for the $\mathcal{R}$-module $\mathcal{R}/\mathcal{I}$. It is here denoted by

$$\mathrm{Std} = \{\partial^{q_1}, \partial^{q_2}, \dots, \partial^{q_R}\}\,, \tag{4.22}$$

where $R := \mathrm{rank}(\mathcal{I})$ and each $q_i \in \mathbb{Z}^N_{\geq 0}$. Without loss of generality, we may set $\mathrm{Std}_1 = 1$. A basis of *functions* for the PDE solution space is then given by the action of each standard monomial on f:

$$e = \begin{bmatrix} \mathrm{Std}_1 \bullet f = f \\ \mathrm{Std}_2 \bullet f \\ \vdots \\ \mathrm{Std}_R \bullet f \end{bmatrix}\,. \tag{4.23}$$

We are interested in finding a set of first-order PDEs satisfied by e. To this end, consider the Weyl algebra element $\partial_i \mathrm{Std}_j$. Since the standard monomials form a basis for $\mathcal{R}/\mathcal{I}$, it

[8] Bruno Buchberger was in fact himself the inventor of Gröbner bases, but he named them after his doctoral supervisor Wolfgang Gröbner as a sign of respect.

must be possible to write

$$\partial_i \mathrm{Std}_j = \sum_{k=1}^{R} p_{ij}^k(z)\,\mathrm{Std}_k \mod \mathcal{I}\,, \quad p_{ij}^k(z) \in \mathbb{C}(z)\,. \tag{4.24}$$

The notation "mod $\mathcal{I}$" means that this equation holds modulo the ideal $\mathcal{I}$, in the sense of (4.11). Most of the time we leave the "mod $\mathcal{I}$" as implicit.

Combining (4.23) with (4.24), we then have

$$\partial_i \bullet e = \begin{bmatrix} (\partial_i \mathrm{Std}_1) \bullet f \\ (\partial_i \mathrm{Std}_2) \bullet f \\ \vdots \\ (\partial_i \mathrm{Std}_R) \bullet f \end{bmatrix} = P_i(z) \cdot e \mod \mathcal{I}\,, \quad i = 1, \ldots, N\,, \tag{4.25}$$

where $P_i(z)$ is an $R \times R$ matrix whose (j,k)th entry equals the rational function $p_{ij}^k(z)$. The set of first-order PDEs (4.25) is dubbed a *Pfaffian system*. By manipulating the standard monomials, it can be shown [86] that the *Pfaffian matrices* P_i satisfy the *integrability condition*

$$\partial_i \bullet P_j + P_i \cdot P_j = \partial_j \bullet P_i + P_j \cdot P_i\,. \tag{4.26}$$

The next chapter will be dedicated to deriving Pfaffian systems from the point of view of $\mathcal{D}$-modules. Such a calculation requires two steps:

1. Find a set of standard monomials Std.

2. Derive the decomposition (4.24).

Both steps are possible when a Gröbner basis is known (the second step is then performed via an algorithm analogous to that of polynomial division). However, as mentioned earlier, we shall seek to avoid Gröbner bases as their computation often requires huge resources. As an aside, if a Pfaffian system is known by other means, then one can immediately construct a Gröbner basis too [82, Appendix B].

We end this section with the following observation: if f were a single FI, then (4.25) is exactly the kind of Pfaffian systems that we studied in **Part I** of this thesis! The decomposition of $\partial_i \mathrm{Std}_j$ in (4.24) is then analogous to an IBP reduction.

Chapter 5

Macaulay Matrices

Based on [82], this chapter presents a novel algorithm for the computation of Pfaffian systems. It avoids the use of Gröbner bases in favor of linear algebra methods, making it quite efficient.

Throughout this chapter we assume that

$$\text{a set of standard monomials Std} = \{\partial^{q_i}\}_{i=1,\ldots,R} \text{ is given,} \tag{5.1}$$

where the cardinality $|\mathrm{Std}| = R$ equals the holonomic rank of a given left ideal $\mathcal{I}$. While this is a strong assumption, it will actually be fulfilled for the so-called GKZ system that we study in Chapter (6).

Section 5.1

From Pfaffian to Macaulay matrix

Let $\mathcal{I} = \langle d_1, \ldots, d_D \rangle$ be a holonomic ideal generated by a collection of D operators $d_i \in \mathcal{R}$. In this section we suppose to know the Pfaffian matrices P_i associated to $\mathcal{I}$. It holds that

$$\partial_i \mathrm{Std}_j = \sum_{k=1}^{R} (P_i)_{jk} \, \mathrm{Std}_k \quad \text{in} \quad \mathcal{R}/\mathcal{I}\,. \tag{5.2}$$

This expression can be lifted from the quotient ring $\mathcal{R}/\mathcal{I}$ to the whole ring $\mathcal{R}$ by writing

$$\partial_i \mathrm{Std}_j = \sum_{k=1}^{R} (P_i)_{jk} \, \mathrm{Std}_k \;+\; \sum_{k=1}^{D} \Delta_{jk} \, d_k \quad \text{in} \quad \mathcal{R}\,, \tag{5.3}$$

where each $\Delta_{jk} \in \mathcal{R}$. Our short-term goal is to aptly rewrite the second sum above, which will culminate in (5.7). We warn that this will require some index gymnastics, but the final result does simplify. To ease the notation in intermediate expressions, we shall keep the index i fixed (otherwise we would have to write Δ_{ijk} etc.). The index i will be reinstated at the very end.

To start, we bring the operator Δ_{jk} into normally ordered form:

$$\Delta_{jk} \quad = \sum_{q \,\in\, \mathrm{supp(Der)}} \delta_{jkq}(z)\, \partial^q\,, \quad \partial^q \in \mathrm{Der}\,, \tag{5.4}$$

where $\delta_{jkq} \in \mathbb{C}(z)^{R\times D\times|\mathrm{Der}|}$ is a newly introduced rank-3 tensor of rational functions. The set Der contains all the derivative monomials appearing on the RHS.

If we substitute the operators Δ_{jk} from (5.4) into the second sum of (5.3), then the monomials $\partial^q \in \mathrm{Der}$ will act on the generators d_i (which potentially contain rational functions in z as prefactors). In normally ordered form, this action is written as

$$\partial^q\, d_k = \sum_{p \,\in\, \mathrm{supp(Mons)}} M_{qkp}(z)\, \partial^p\,, \quad \partial^p \in \mathrm{Mons}\,. \tag{5.5}$$

Here we introduced a rank-3 tensor of rational functions $M_{qkp} \in \mathbb{C}(z)^{|\mathrm{Der}|\times D\times|\mathrm{Mons}|}$ and defined a set Mons whose elements consists of all the monomials ∂^p appearing on the RHS. Upon concatenation of the first two indices q and k, the rank-3 tensor M_{qkp} turns into the *Macaulay matrix* [87]

$$M_{op}(z) := M_{(qk)p}(z) \in \mathbb{C}^{(|\mathrm{Der}|\cdot D)\times|\mathrm{Mons}|}\,. \tag{5.6}$$

The combined index $o = (qk) = (kq)$ runs over the Cartesian product of sets supp (Der) $\times$ $\{1, \ldots, D\}$.

Plugging equations (5.4), (5.5) and (5.6) into the second sum of (5.3), this term now takes the form of a triple matrix product:

$$\sum_{k=1}^{D} \Delta_{jk}\, d_k \quad = \quad \sum_{o,\,p} \delta_{jo}(z)\, M_{op}(z)\, \partial^p \quad = \quad \big(C \cdot M \cdot \mathrm{Mons}\big)_j\,. \tag{5.7}$$

$C \in \mathbb{C}(z)^{R\times(|\mathrm{Der}|\cdot D)}$ stands for the coefficient matrix $C_{jo} = \delta_{j(kq)}(z)$, when Der is regarded as a column vector in (5.4). The set Mons is here also regarded as a column vector.

Some of the monomials in Mons may be standard, but all need not be. The non-standard monomials in Mons shall henceforth be called *exterior monomials*:

$$\mathrm{Ext} = \mathrm{Mons} \setminus \mathrm{Std}\,. \tag{5.8}$$

The disjoint union

$$\mathrm{Mons} = \mathrm{Ext} \ \sqcup\ \mathrm{Std} \tag{5.9}$$

then induces a partition on the columns of the Macaulay matrix:

$$M = \left[\ M_{\mathrm{Ext}}\ \middle|\ M_{\mathrm{Std}}\ \right], \quad \text{with} \tag{5.10}$$

$$M_{\mathrm{Ext}} \in \mathbb{C}(z)^{(|\mathrm{Der}|\cdot D)\times|\mathrm{Ext}|} \qquad \text{and} \qquad M_{\mathrm{Std}} \in \mathbb{C}(z)^{(|\mathrm{Der}|\cdot D)\times|\mathrm{Std}|}$$

Thinking of the set Std as a column vector, we further decompose the LHS of (5.3) as

$$\partial_i \mathrm{Std} \quad = \quad \left[\ C_{\mathrm{Ext}}\ \middle|\ C_{\mathrm{Std}}\ \right] \cdot \mathrm{Mons} \quad = \quad C_{\mathrm{Ext}} \cdot \mathrm{Ext} \ + \ C_{\mathrm{Std}} \cdot \mathrm{Std}\,, \quad \text{with} \tag{5.11}$$

$$C_{\mathrm{Ext}} \in \{0,1\}^{|\mathrm{Std}|\times|\mathrm{Ext}|} \qquad \text{and} \qquad C_{\mathrm{Std}} \in \{0,1\}^{|\mathrm{Std}|\times|\mathrm{Std}|}\,.$$

In other words, C_{Ext} and C_{Std} are two binary matrices that encode whether the monomials in ∂_iStd are exterior or standard.

Finally, inserting equations (5.7), (5.10) and (5.11) into the Pfaffian equation (5.3), we arrive at

$$(C_{\mathrm{Ext}} - C \cdot M_{\mathrm{Ext}}) \ + \ (C_{\mathrm{Std}} - C \cdot M_{\mathrm{Std}}) \quad = \quad P_i \cdot \mathrm{Std}\,. \tag{5.12}$$

But, because Ext and Std are linearly independent in $\mathcal{R}$, this is equivalent to the system

$$\boxed{\begin{aligned} C^{(i)}_{\mathrm{Ext}} - C^{(i)} \cdot M_{\mathrm{Ext}} &= 0 && (5.13)\\ C^{(i)}_{\mathrm{Std}} - C^{(i)} \cdot M_{\mathrm{Std}} &= P_i\,. && (5.14)\end{aligned}}$$

Here we also reinstated the dependence on the index i, as promised earlier in this section. These two matrix equations are central to the algorithm for computing Pfaffian matrices that we propose in the next section.

Section 5.2

From Macaulay to Pfaffian matrix

Now we reverse the logic in the derivation leading to equations (5.13) and (5.14). Although we started by assuming to know the Pfaffian system (5.2), we now regard it as the goal.

Supposing that $C_\bullet^{(i)}$ and $M_\bullet$ in (5.13) and (5.14) are easily calculable ($\bullet$ = Ext, Std), then $C^{(i)}$ and P_i are the only unknowns. The purported strategy is this:

solve (5.13) for the rational matrix $C^{(i)} \implies$ insert $C^{(i)}$ into (5.14) to obtain P_i.

Let us formalize this strategy into a proper algorithm. To begin, define the set of all derivative monomials bounded by a degree $Q \in \mathbb{Z}_{\geq 0}$:

$$\mathrm{Der}_Q = \{\partial^q\}_{|q| \leq Q} \, . \tag{5.15}$$

In analogy with Mons from (5.5), we let Mons_Q denote the derivative monomials appearing in the normally order form of the set $\big\{\partial^q\, d_k\big\}_{\partial^q \,\in\, \mathrm{Der}_Q}^{k=1,\ldots,D}$. The Macaulay matrix M_Q of degree Q is defined as in (5.6), but with Der and Mons replaced with Der_Q and Mons_Q. It thus holds that

$$\begin{aligned} \Big\{\partial^q\, d_k\Big\}_{\partial^q \,\in\, \mathrm{Der}_Q}^{k=1,\ldots,D} &= \sum_{p \,\in\, \mathrm{supp}\big(\mathrm{Mons}_Q\big)} (M_Q)_{(qk)p}\, \partial^p \qquad (5.16) \\ &= M_Q \cdot \mathrm{Mons}_Q \, . \qquad (5.17) \end{aligned}$$

The set on the LHS of (5.16) is here viewed as a column vector of operators with rational function coefficients, and $M_Q = M_Q(z)$ is the coefficient matrix of the monomial derivatives Mons_Q. The set Der_Q can be thought of as "seeding" the Macaulay matrix.

The idea is now to adjust the degree Q until (5.13) admits a solution for $C^{(i)}$:

Algorithm 1 : Pfaffian matrix by the Macaulay matrix method

Input: Standard monomials Std, generators $\{d_1, \ldots d_D\}$ of a holonomic ideal $\mathcal{I}$, direction i.

Output: Pfaffian matrix P_i.

```
$Q = 0$
$M = M_0 = \big[\ M_{\mathrm{Ext}}\ \big|\ M_{\mathrm{Std}}\ \big]$
Mons = $\mathrm{Mons}_0$
while $C_{\mathrm{Ext}}^{(i)} = C^{(i)} \cdot M_{\mathrm{Ext}}$ is not solvable w.r.t. $C^{(i)}$ do
    $Q{+}{+}$
    $M \leftarrow M_Q$
    Mons $\leftarrow \mathrm{Mons}_Q$
end while
Solve $C_{\mathrm{Ext}}^{(i)} = C^{(i)} \cdot M_{\mathrm{Ext}}$ for $C^{(i)}$
return $P_i = C_{\mathrm{Std}}^{(i)} - C^{(i)} \cdot M_{\mathrm{Std}}$
```

There are a couple of immediate optimizations to this algorithm. First, notice that (5.13) has a solution when the rows of $C^{(i)}_{\text{Ext}}$ lie in the row space of M_{Ext}. The condition in the **while** statement in line 4 of Algorithm (1) then translates into the requirement

$$\operatorname{rank}\left[\frac{M_{\text{Ext}}}{C^{(i)}_{\text{Ext}}}\right] = \operatorname{rank}\big[M_{\text{Ext}}\big]\,. \tag{5.18}$$

It is quick to check this equality by setting every parameter in the problem to a number in $\mathbb{Q}$ or a finite field $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ where p is prime.

Second, to solve the system (5.13), it often suffices to pick only a subset of the rows in M_{Ext}. The reason is that some rows may be linearly dependent. Such a subset can be identified by replacing parameters with numbers and row reducing.

Lastly, since every matrix in the problem is defined over the rational function field $\mathbb{C}(z)$, it is vastly more efficient to employ *rational reconstruction* algorithms rather than Gaussian elimination. These algorithms reconstruct analytic expressions for rational functions by repeated evaluations over $\mathbb{F}_p$. One need not even reconstruct the intermediate "dummy matrix" $C^{(i)}$, as we are ultimately only interested in the Pfaffian matrix P_i. I.e., rational reconstruction can be postponed until all the matrix operations have been carried out in (5.14). This thesis employs the rational reconstruction software FiniteFlow [88], because it allows one to easily chain together several operations (such as matrix multiplication and addition of matrices), with reconstruction postponed until the final step. See [89, 90, 91, 92] for alternative implementations.

Example 5.1. Consider the holonomic ideal

$$\mathcal{I} = \langle z\partial - \beta \rangle\,, \tag{5.19}$$

where $z_1 = z$, $\partial_1 = \partial$ and $\beta \in \mathbb{C}$ is a parameter. Although this example is too simple for an interesting application of the Macaulay matrix method, we include it to showcase the many definitions in the above text.

To build the Macaulay matrix, we must first choose the seeds Der_Q defined in (5.16). It turns out that $Q = 0$ is a sufficient degree, but for the sake of illustration let us pick $Q = 1$. Then $\text{Der}_Q = \{1, \partial\}$. Following (5.16), we apply the monomials in Der_Q to the generator of $\mathcal{I}$:

$$1(z\partial - \beta) = z\partial - \beta \tag{5.20}$$

$$\partial(z\partial - \beta) = \partial z\partial - \beta\partial = (1 + z\partial)\partial - \beta\partial = z\partial^2 + (1 - \beta)\partial\,. \tag{5.21}$$

The set Mons_Q is formed from all the derivative monomials appearing above:

$$\mathrm{Mons}_Q = \{\partial^2, \partial, 1\} = \mathrm{Ext} \ \sqcup \ \mathrm{Std} = \{\partial^2, \partial\} \ \sqcup \ \{1\} \,. \tag{5.22}$$

The Macaulay matrix can now be constructed as

$$\begin{bmatrix} 1(z\partial - \beta) \\ \partial(z\partial - \beta) \end{bmatrix} = \left[\begin{array}{c|c} M_{\mathrm{Ext}} & M_{\mathrm{Std}} \end{array} \right] \cdot \mathrm{Mons}_Q \tag{5.23}$$

$$= \left[\begin{array}{cc|c} 0 & z & -\beta \\ z & 1-\beta & 0 \end{array} \right] \cdot \begin{bmatrix} \partial^2 \\ \partial \\ 1 \end{bmatrix} \,. \tag{5.24}$$

Next we build the two matrices C_{Ext} and C_{Std} as per (5.11):

$$\partial \mathrm{Std} = \partial [1] = [\partial] \tag{5.25}$$

$$= \left[\begin{array}{c|c} C_{\mathrm{Ext}} & C_{\mathrm{Std}} \end{array} \right] \cdot \mathrm{Mons}_Q \tag{5.26}$$

$$= C_{\mathrm{Ext}} \cdot \mathrm{Ext} \ + \ C_{\mathrm{Std}} \cdot \mathrm{Std} \tag{5.27}$$

$$= \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \partial^2 \\ \partial \end{bmatrix} \ + \ [0] \cdot [1] \,. \tag{5.28}$$

At this point we have all ingredients present in (5.13) and (5.14). The first equation is

$$C_{\mathrm{Ext}} - C \cdot M_{\mathrm{Ext}} = \begin{bmatrix} 0 & 1 \end{bmatrix} - \begin{bmatrix} C_{11} & C_{12} \end{bmatrix} \cdot \begin{bmatrix} 0 & z \\ z & 1-\beta \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix} \,. \tag{5.29}$$

Its solution is $C = \begin{bmatrix} \frac{1}{z} & 0 \end{bmatrix}$. Inserting this into (5.14), we get

$$C_{\mathrm{Std}} - C \cdot M_{\mathrm{Std}} = [0] - \begin{bmatrix} \frac{1}{z} & 0 \end{bmatrix} \cdot \begin{bmatrix} -\beta \\ 0 \end{bmatrix} = \left[\frac{\beta}{z} \right] = P \,, \tag{5.30}$$

where P is the Pfaffian matrix (albeit just a scalar in this case). In other words, we have derived that

$$\partial \mathrm{Std} = P \cdot \mathrm{Std} \implies [\partial] = \left[\frac{\beta}{z} \right] \,, \tag{5.31}$$

which is exactly the relation dictated by the ideal (5.19)! So we have obtained a trivial result, but we can at least be satisfied that everything is self-consistent. ■

Example 5.2. Let $\mathcal{I} = \langle d_1, d_2, d_3, d_4 \rangle$ be the ideal generated by

$$\begin{aligned} d_1 &= z_1\partial_1 + z_2\partial_2 + z_3\partial_3 + z_4\partial_4 - \beta_1 \\ d_2 &= z_2\partial_2 + z_4\partial_4 - \beta_2 \\ d_3 &= z_3\partial_3 + z_4\partial_4 - \beta_3 \\ d_4 &= \partial_1\partial_4 - \partial_2\partial_3\,. \end{aligned} \tag{5.32}$$

In Section (6.5), we show that this is the annihilating ideal for Gauss' ${}_2F_1$ hypergeometric function. The $\mathcal{D}$-module $\mathcal{D}/\mathcal{I}$ turns out to have a basis of standard monomials given by $\mathrm{Std} = \{\partial_4, 1\}$ (later on, in Section (6.4), we explain how to easily find such a basis). The goal of this example is to find the Pfaffian system in direction z_4:

$$\partial_4 \mathrm{Std} = \partial_4 \begin{bmatrix} \partial_4 \\ 1 \end{bmatrix} = \begin{bmatrix} \partial_4^2 \\ \partial_4 \end{bmatrix} = P_4 \cdot \begin{bmatrix} \partial_4 \\ 1 \end{bmatrix}\,. \tag{5.33}$$

To compute P_4, we need to resolve the higher-order monomial ∂_4^2 in terms of ∂_4 and 1. But ∂_4^2 does not appear in the generators d_i from (5.32), so the seeding set $\mathrm{Der}_{Q=0} = \{1\}$ is insufficient. It turns out that the degree $Q = 1$ monomials $\mathrm{Der}_1 = \{1, \partial_1, \partial_2, \partial_3, \partial_4\}$ lead to a solvable system.

It is rather tedious to compute the Macaulay matrix by hand given this seed. Instead, we use the newly developed package MT_MM written by Nobuki Takayama and Saiei J. Matsubara-Heo in the CAS called ASIR. ASIR can be built from source by cloning the GIT repository https://github.com/openxm-org/OpenXM. Source code and documentation for MT_MM can be found in the directories

```
OpenXM/src/asir-contrib/packages/src/mt_mm
OpenXM/doc/asir-contrib/en.
```

For the case at hand, the Macaulay matrix is obtained by entering the following commands into ASIR[1]:

```
D1 = z1*dz1 + z2*dz2 + z3*dz3 + z4*dz4 - b1;
D2 = z2*dz2 + z4*dz4 - b2;
D3 = z3*dz3 + z4*dz4 - b3;
D4 = dz1*dz4 - dz2*dz3;
Ideal = [D1, D2, D3, D4];
Std = [dz4, 1];
mt_mm.find_macaulay(Ideal, Std, [z1, z2, z3, z4]);
```

The output is a list $\{M_{\mathrm{Ext}}, M_{\mathrm{Std}}, \mathrm{Ext}, \mathrm{Std}\}$. Explicitly, we obtain the following data:

[1] We call the generators `Di` rather than `di` since most lower-case objects are name-protected in ASIR.

$$
M_{\mathrm{Ext}} = \left[\begin{array}{c||ccccccccccccccccccccc}
\mathrm{Ext} \rightarrow & \partial_1\partial_2\partial_3 & \partial_2^2\partial_3 & \partial_2\partial_3^2 & \partial_1^2\partial_4 & \partial_1\partial_2\partial_4 & \partial_1\partial_3\partial_4 & \partial_2\partial_3\partial_4 & \partial_1\partial_4^2 & \partial_1^2 & \partial_1\partial_2 & \partial_2^2 & \partial_1\partial_3 & \partial_2\partial_3 & \partial_3^2 & \partial_1\partial_4 & \partial_2\partial_4 & \partial_3\partial_4 & \partial_4^2 & \partial_1 & \partial_2 & \partial_3 \\
\hline\hline
1d_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_1 & z_2 & z_3 \\
\hline
1d_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_2 & \cdot \\
\hline
1d_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_3 \\
\hline
1d_4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\hline\hline
\partial_1 d_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_1 & z_2 & z_3 & z_4 & \cdot & \cdot & \cdot \\
\hline
\partial_1 d_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_2 & \cdot & z_4 & \cdot & \cdot & \cdot \\
\hline
\partial_1 d_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_3 & z_4 & \cdot & \cdot & \cdot \\
\hline
\partial_1 d_4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\hline\hline
\partial_2 d_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_1 & z_2 & z_3 & \cdot & \cdot & z_4 & \cdot & \cdot & \cdot & 1-\beta_1 \\
\hline
\partial_2 d_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_2 & \cdot & \cdot & \cdot & z_4 & \cdot & \cdot & \cdot & -\beta_3 \\
\hline
\partial_2 d_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_3 & \cdot & \cdot & z_4 & \cdot & \cdot & \cdot & 1-\beta_3 \\
\hline
\partial_2 d_4 & \cdot & \cdot & 1 & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\hline\hline
\partial_3 d_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_1 & z_2 & \cdot & z_3 & \cdot & \cdot & z_4 & \cdot & \cdot & \cdot & 1-\beta_1 & \cdot \\
\hline
\partial_3 d_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_2 & \cdot & \cdot & \cdot & \cdot & z_4 & \cdot & \cdot & \cdot & 1-\beta_3 & \cdot \\
\hline
\partial_3 d_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_3 & \cdot & \cdot & z_4 & \cdot & \cdot & \cdot & -\beta_3 & \cdot \\
\hline
\partial_3 d_4 & \cdot & 1 & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\hline\hline
\partial_4 d_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_1 & z_2 & \cdot & z_3 & \cdot & \cdot & z_4 & \cdot & \cdot & \cdot & 1-\beta_1 & \cdot & \cdot \\
\hline
\partial_4 d_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_2 & \cdot & \cdot & \cdot & \cdot & z_4 & \cdot & \cdot & \cdot & -\beta_3 & \cdot & \cdot \\
\hline
\partial_4 d_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & z_3 & \cdot & \cdot & z_4 & \cdot & \cdot & \cdot & -\beta_3 & \cdot & \cdot \\
\hline
\partial_4 d_4 & 1 & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{array}\right]
$$

$$M_{\mathrm{Std}} = \left[\begin{array}{c||cc}
\mathrm{Std} \to & \partial_4 & 1 \\ \hline\hline
1d_1 & z_4 & -\beta_1 \\ \hline
1d_2 & z_4 & -\beta_3 \\ \hline
1d_3 & z_4 & -\beta_3 \\ \hline
1d_4 & \cdot & \cdot \\ \hline\hline
\partial_1 d_1 & 1-\beta_1 & \cdot \\ \hline
\partial_1 d_2 & 1-\beta_3 & \cdot \\ \hline
\partial_1 d_3 & 1-\beta_3 & \cdot \\ \hline
\partial_1 d_4 & \cdot & \cdot \\ \hline\hline
\partial_2 d_1 & \cdot & \cdot \\ \hline
\partial_2 d_2 & \cdot & \cdot \\ \hline
\partial_2 d_3 & \cdot & \cdot \\ \hline
\partial_2 d_4 & \cdot & \cdot \\ \hline\hline
\partial_3 d_1 & \cdot & \cdot \\ \hline
\partial_3 d_2 & \cdot & \cdot \\ \hline
\partial_3 d_3 & \cdot & \cdot \\ \hline
\partial_3 d_4 & \cdot & \cdot \\ \hline\hline
\partial_4 d_1 & \cdot & \cdot \\ \hline
\partial_4 d_2 & \cdot & \cdot \\ \hline
\partial_4 d_3 & \cdot & \cdot \\ \hline
\partial_4 d_4 & \cdot & \cdot
\end{array}\right].$$

The 21 columns in M_{Ext} are labeled by the exterior monomials Ext. The two columns in M_{Std} are labeled by Std. The 20 rows in M_{Ext} and M_{Std} are labeled by the seeds Der_1 acting on the generators d_i.

Rows of the matrix M_{Ext} dictate relations among the external monomials. One notices, however, that some of the rows are linearly dependent on each other (this stems from an overzealous choice of the set $\{\partial^q d_k\}$ from (5.16), as only a subset of the seeds would suffice). Row reduction of M_{Ext} reveals that the 17 rows numbered by $\{1, \ldots, 14, 16, 17, 20\}$ are independent. Let M'_{Ext} be the 17×21 matrix built from these rows. The 17×2 matrix M'_{Std} is defined in a similar manner. This row reduction can naturally be performed by fixing parameters to numbers for more complicated examples.

Next we construct the two matrices $C^{(4)}_{\text{Ext}}$ and $C^{(4)}_{\text{Std}}$ according to (5.11), finding that

$$C^{(4)}_{\text{Ext}} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \tag{5.34}$$

$$C^{(4)}_{\text{Std}} = \begin{bmatrix} \cdot & \cdot \\ 1 & \cdot \end{bmatrix} . \tag{5.35}$$

Now we can employ the central equations (5.13) and (5.14): The two-step process is

1. Define an unknown matrix $C^{(4)}$ of size $R \times S$ where $R = |\text{Std}| = 2$ and $S = \#\{\text{rows in } M'_{\text{Ext}}\} = 17$. This matrix is determined by solving

   $$C^{(4)}_{\text{Ext}} - C^{(4)} \cdot M'_{\text{Ext}} = 0\,, \tag{5.36}$$

   which can be done instantaneously with e.g. Mathematica.

2. The solution for $C^{(4)}$ is inserted into

   $$C^{(4)}_{\text{Std}} - C^{(4)} \cdot M'_{\text{Std}} = P_4 \tag{5.37}$$

   to obtain the Pfaffian matrix P_4.

The resulting Pfaffian system in direction z_4 works out to

$$\partial_4 \begin{bmatrix} \partial_4 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{-\beta_1+\beta_3+\beta_3-1}{z_4} + \frac{\beta_1 z_1}{z_1 z_4 - z_2 z_3} & \frac{\beta_3 \beta_3 z_1}{z_4(z_2 z_3 - z_1 z_4)} \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \partial_4 \\ 1 \end{bmatrix} . \tag{5.38}$$

In particular, the first row of this Pfaffian equation says that

$$\partial_4^2 = \frac{z_2 z_3(\beta_1 - \beta_3 - \beta_3 + 1) + z_1 z_4(\beta_3 + \beta_3 - 1)}{z_4(z_1 z_4 - z_2 z_3)} \partial_4 - \frac{\beta_3 \beta_3 z_1}{z_4(z_1 z_4 - z_2 z_3)} 1 \quad \text{mod } \mathcal{I}\,,$$

which, in physicist terms, could be thought of as an "IBP reduction" of ∂_4^2 in terms of the "MIs" ∂_4 and 1.

Pfaffian matrices in the directions $\{z_1, z_2, z_3\}$ are found in a similar manner. ■

Chapter 6

GKZ Systems

Among the Euler type integrals ... there are the integrals $\int \Pi P_i(t_1, \ldots, t_n)^{\alpha_i} \times t_1^{\beta_1} \ldots t_n^{\beta_n} \, \mathrm{d}t_1 \ldots \mathrm{d}t_n$*, where* P_i *are polynomials, i.e. practically all integrals which arise in quantum field theory.*

Gel'fand, Kapranov, Zelevinsky ('89) [41]

A GKZ system is a particular $\mathcal{D}$-module $\mathcal{D}/H_A(\beta)$, where $H_A(\beta)$ is an annihilating ideal built from an integer matrix A and a collection of complex parameters β_i. We have secretly already shown examples of generators for GKZ systems in equations (4.2), (5.19) and (5.32).

Why study this specific $\mathcal{D}$-module? As indicated by the quote above, GKZ systems can be related to FIs when they are written in parametric representations rather than in momentum space (the connection between GKZ systems and FIs will be explained in the next chapter). What's more, it is easy to determine the annihilating ideal $H_A(\beta)$ as well as a basis of standard monomials, which is computationally advantageous.

Section 6.1

Euler integrals

To define the GKZ system, let us begin by considering the class of functions annihilated by the generators of $H_A(\beta)$. These functions are called *Euler integrals* [93], written as[1]

$$f_\beta(z) = \int_\Gamma g(z|x)^{\beta_0} x_1^{-\beta_1} \cdots x_n^{-\beta_n} \frac{\mathrm{d}x}{x} \quad \text{with} \quad \frac{\mathrm{d}x}{x} := \frac{\mathrm{d}x_1 \wedge \cdots \wedge \mathrm{d}x_n}{x_1 \cdots x_n} \,. \tag{6.1}$$

Let us dwell on the building blocks of this integral.

[1] The most general definition contains several polynomials $\{g_1, \ldots, g_l\}$ in the integrand, but for our inquiry a single polynomial $g(z|x)$ will suffice.

1. Γ is some chosen integration cycle in $\mathbb{C}^n$. For the purpose of this thesis we may pick $\Gamma = (0, \infty)^n$ (though to be rigorous one should also specify a Riemann sheet data, as the integrand is multivalued; see [93, Section 3.1]).

2. $g(z|x)$ is a Laurent polynomial in the integration variables $x = (x_1, \ldots, x_n)$ with monomial coefficients $z = (z_1, \ldots, z_N)$. In multivariate exponent notation

$$g(z|x) = \sum_{i=1}^{N} z_i \, x^{\alpha_i} \,, \quad \alpha_i \in \mathbb{Z}^n \,. \tag{6.2}$$

   Crucially, each monomial coefficient z_i is regarded is an *independent* variable of the function $f_\beta(z)$.

3. The exponents $\beta = (\beta_0, \ldots, \beta_n) \in \mathbb{C}^{n+1}$ are complex parameters. We generally assume that they are *generic* real numbers, meaning that they should not be integral. This will be formulated more precisely as a *non-resonance* condition later on.

By appending[2] a "1" in front of each multi-index α_i from (6.2), we define a collection of N column vectors $a_i = \left[1, \alpha_i\right]^T \in \mathbb{Z}^{n+1}$. These vectors are assumed to span $\mathbb{Z}^{n+1}$. Further, they give rise to an $(n+1) \times N$ dimensional matrix

$$A = \left[\; a_1 \; \middle| \; \ldots \; \middle| \; a_N \; \right] \,. \tag{6.3}$$

The polynomial g in the integrand of (6.1) thus specifies the integer matrix A, and vice versa. The (left) kernel of A is defined as

$$\ker(A) = \left\{ (u_1, \, \ldots, \, u_N) \in \mathbb{Z}^N \; \middle| \; \sum_{i=1}^{N} u_i \, a_i = \mathbf{0} \right\} \,. \tag{6.4}$$

This can alternatively be viewed as a linear space spanned by finitely many integer vectors u satisfying $A \cdot u = \mathbf{0}$. Given a vector $u \in \ker(A)$, we shall decompose it into positive and negative components by writing $u = u^+ - u^-$ for $u^+, \, u^- \in \mathbb{Z}^N_{\geq 0}$.

[2] This is called the *Cayley trick* or the *Cayley configuration*.

SECTION 6.2

GKZ $\mathcal{D}$-module

The main characters of this chapter are the following operators parametrized by the integer matrix A and the complex parameters β:

$$E = A \cdot \theta - \beta \tag{6.5}$$

$$\Box_u = \partial^{u^+} - \partial^{u^-} \,, \quad \forall\, u \in \ker(A) \,. \tag{6.6}$$

The vector $\theta = \big[\theta_1, \ldots, \theta_N\big]^T$ is a collection of *Euler operators* $\theta_i = z_i\, \partial_i$. The $n+1$ operators E_i are often called the *homogeneity operators* (cf. Section (6.5)). The ideal generated by the operators $\Box_u$ is called the *toric ideal*[3].

Quite remarkably, E and $\Box_u$ can be shown [42] to be the complete set of annihilating operators for the Euler integral (6.1):

$$E_i \bullet f_\beta(z) = 0 \,, \quad i = 0, \ldots, n \tag{6.7}$$

$$\Box_u \bullet f_\beta(z) = 0 \,, \quad \forall\, u \in \ker(A) \,. \tag{6.8}$$

The *GKZ system* [41, 94] associated to an Euler integral is defined as the $\mathcal{D}$-module $\mathcal{D}/H_A(\beta)$, where $H_A(\beta)$ is the annihilating ideal generated by E and $\Box_u$:

$$H_A(\beta) = \sum_{i=0}^{n} \mathcal{D}\, E_i + \sum_{u \,\in\, \ker(A)} \mathcal{D}\, \Box_u \,. \tag{6.9}$$

Given some complicated integral, it might well be very hard to find the complete set of annihilating operators. But for the case of Euler integrals having indeterminate z variables, we can apparently immediately write down all of those operators! In the $\mathcal{D}$-module philosophy, we may thus study the GKZ system as a proxy for studying the Euler integral itself.

The GKZ system is holonomic [95], so it is endowed with the structure of a finite-dimensional vector space. This means that we can use the Macaulay matrix method to construct first-order Pfaffian systems for Euler integrals given the higher-order operators that generate $H_A(\beta)$. This will be topic of Chapter (7).

[3] The name stems from its relation to *toric geometry*, a field of algebraic geometry that studies varieties containing a torus as an open dense subset [85].

Example 6.1. Consider the Euler integral

$$f_\beta(z) = \int_\Gamma \left(z_1x_1 + z_2x_2 + z_3x_3 + z_4x_1x_2 + z_5x_3^2\right)^{\beta_0} x_1^{-\beta_1} x_2^{-\beta_3} x_3^{-\beta_3} \frac{\mathrm{d}x}{x} \,. \tag{6.10}$$

We seek to construct its GKZ system. The first step is to input the exponent multi-indices (with 1's appended) into the columns of the A-matrix (6.3):

$$A = \left[\begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 2 \end{array}\right] . \tag{6.11}$$

The 4th column, for instance, represents the monomial $z_4x^{(1,1,0)} = z_4x_1x_2$. The operators E_i from (6.5) can now be written as

$$E = A \cdot \theta - \beta = \begin{bmatrix} \theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 - \beta_0 \\ \theta_1 + \theta_4 - \beta_1 \\ \theta_2 + \theta_4 - \beta_2 \\ \theta_3 + 2\theta_5 - \beta_3 \end{bmatrix} . \tag{6.12}$$

The toric part (6.6) of the GKZ system requires knowing $\ker(A)$. This can be found by solving a linear system, or by using one's favorite CAS (e.g. the command `NullSpace` in MATHEMATICA). One finds that $\ker(A) = \mathrm{span}(u)$, where u is a vector that we decompose as

$$u = \begin{bmatrix} 1 \\ 1 \\ -2 \\ -1 \\ 1 \end{bmatrix} = u^+ - u^- = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 2 \\ 1 \\ 0 \end{bmatrix} . \tag{6.13}$$

The toric operator is therefore

$$\Box_u = \partial^{u^+} - \partial^{u^-} = \partial_1\partial_2\partial_5 - \partial_3^2\partial_4 \,. \tag{6.14}$$

In summary, the annihilating ideal for the Euler integral (6.10) is generated as $H_A(\beta) = \langle E, \Box_u \rangle$, where E and $\Box_u$ are given by (6.12) and (6.14). ■

Section 6.3

Twisted de Rham cohomology

The GKZ system can be related to another useful vector space for studying Euler integrals, namely the *twisted de Rham cohomology group*. The elements of this group are, loosely speaking, Euler inte*grands*. By defining this space we will discover concrete formulas for representing the Euler integral (6.1) as a differential operator inside a $\mathcal{D}$-module.

Twisted de Rham cohomology has recently garnered a lot of attention in the study of FIs because it offers a novel and potentially more powerful framework for IBP reduction. See e.g. [21, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108] and references therein.

6.3.1 "Traditional" de Rham cohomology

Before discussing the "twisted" de Rham cohomology group, it is instructive to first summarize the "traditional" theory. The story begins with the celebrated theorem by Stokes:

$$\int_\Gamma \mathrm{d}\omega = \int_{\partial\Gamma} \omega\,. \tag{6.15}$$

$\omega = h(x)\mathrm{d}^k x$ is a differential k-form, where $h(x)$ is some rational function, and d is the total derivative on integration variables acting as

$$\mathrm{d}\omega = \frac{\partial h(x)}{\partial x_1}\mathrm{d}x_1 \wedge \mathrm{d}^k x + \cdots + \frac{\partial h(x)}{\partial x_n}\mathrm{d}x_n \wedge \mathrm{d}^k x\,. \tag{6.16}$$

Note that d sends a k-form to a $(k+1)$-form.

The notation $\partial\Gamma$ denotes the boundary of the k-dimensional integration contour Γ (see the classical textbook [109] for a rigorous definition). Note that the boundary is $(k-1)$-dimensional.

We focus on a particular class of integrals $\int_\Gamma \omega$ for which

$$\begin{array}{ccccc} \Gamma \text{ is a } \textit{cycle} & \Longleftrightarrow & \partial\Gamma & = & 0 \\ \omega \text{ is a } \textit{co-cycle} & \Longleftrightarrow & \mathrm{d}\omega & = & 0\,. \end{array}$$

Consider now the following two scenarios:

1. Fix ω and shift Γ by the boundary of another contour, $\Gamma \to \Gamma + \partial\gamma$. It follows from

Stokes' theorem that

$$\int_{\Gamma+\partial\gamma}\omega=\int_\Gamma\omega+\int_{\partial\gamma}\omega=\int_\Gamma\omega+\cancelto{0}{\int_\gamma \mathrm{d}\omega}=\int_\Gamma\omega\,. \tag{6.17}$$

So, the value of the integral does not depend on any particular contour, but rather an *equivalence class* $[\Gamma]$ of contours, where $\Gamma\sim\Gamma'$ precisely when $\Gamma=\Gamma'+\partial\gamma$.

2. Fix Γ and shift ω by the total derivative of another differential form, $\omega\to\omega+\mathrm{d}\psi$. Stokes' theorem implies that

$$\int_\Gamma(\omega+\mathrm{d}\psi)=\int_\Gamma\omega+\int_\Gamma\mathrm{d}\psi=\int_\Gamma\omega+\cancelto{0}{\int_{\partial\Gamma}\psi}=\int_\Gamma\omega\,. \tag{6.18}$$

The numerical value of the integral is thus determined by an *equivalence class* $[\omega]$ of differential forms, where $\omega\sim\omega'$ iff $\omega=\omega'+\mathrm{d}\psi$.

It is therefore natural to study contours up to boundary terms, and differential forms up to total derivatives. Let's formalize these notions.

Define

$$C_k=\big\{\text{formal sums of } k\text{-dimensional cycles } \Gamma\big\}\,. \tag{6.19}$$

The boundary map $\partial_k : C_k\to C_{k-1}$ sends k-cycles to $(k-1)$-cycles. An important result is that "the boundary of a boundary is zero":

$$\partial_k\circ\partial_{k+1}=0 \qquad\Longrightarrow\qquad \text{Image}(\partial_{k+1})\ \subset\ \text{Kernel}(\partial_k)\,. \tag{6.20}$$

It thus makes sense to define the kth *homology group* as the quotient space

$$H_k=\frac{\text{Kernel}(\partial_k)}{\text{Image}(\partial_{k+1})}\,. \tag{6.21}$$

The numerator indicates that we focus on k-cycles Γ for which $\partial_k\Gamma=0$, and the denominator means that we mod out by boundary terms of the form $\partial_{k+1}\gamma$. So this space indeed consists of the equivalence classes $[\Gamma]$ discussed above.

Furthermore, let

$$\Omega^k=\big\{\text{formal sums of co-cycle } k\text{-forms } \omega\big\}\,. \tag{6.22}$$

The total derivative $\mathrm{d}_k : \Omega^k \to \Omega^{k+1}$ sends k-forms to $(k+1)$-forms. It holds that

$$\mathrm{d}_k \circ \mathrm{d}_{k-1} = 0 \quad \Longrightarrow \quad \mathrm{Image}(\mathrm{d}_{k-1}) \ \subset \ \mathrm{Kernel}(\mathrm{d}_k)\,. \tag{6.23}$$

The kth *de Rham cohomology group* is then defined by the quotient

$$H^k = \frac{\mathrm{Kernel}(\mathrm{d}_k)}{\mathrm{Image}(\mathrm{d}_{k-1})}\,. \tag{6.24}$$

The numerator consists of co-cycles satisfying $\mathrm{d}_k\omega = 0$, and the denominator mods out by total derivatives $\mathrm{d}_{k-1}\psi$. This space therefore consists of the equivalence classes $[\omega]$ mentioned earlier.

6.3.2 Twisted de Rham cohomology

Now consider integrals of the form

$$\int_\Gamma U\,\omega_q \quad \text{where} \quad U = g(z|x)^{\beta_0}\,x^{-\beta'} \quad \text{and} \quad \omega_q = g(z|x)^{-q_0} x^{q'}\,\frac{\mathrm{d}x}{x}\,, \tag{6.25}$$

with $x^{\beta'} = x_1^{\beta_1} \cdots x_n^{\beta_n}$. We collect the integer exponents of ω_q into a vector

$$q = (q_0,\, q') \in \mathbb{Z} \times \mathbb{Z}^n\,. \tag{6.26}$$

In "traditional" de Rham cohomology, we restricted ourselves to integrals of differential forms with rational coefficient functions. Here there is a new ingredient: the *twist* U. This function is multivalued because the exponents $\beta = (\beta_0, \beta')$ are non-integer by stipulation.

In this setting, Stokes' theorem (6.15) says

$$\int_\Gamma \mathrm{d}\big[U\,\omega_q\big] = \int_{\partial_U \Gamma} U\,\omega_q\,. \tag{6.27}$$

We have upgraded the usual boundary operator to ∂_U due to the multivaluedness of U. ∂_U keeps track of the branch of U because Γ might cross some branch cuts in its path; we refer to [93, Chapter 3.1] for a proper definition.

On the LHS, we can expand the integrand to get

$$\mathrm{d}\big[U\,\omega_q\big] = U\mathrm{d}\omega_q + \mathrm{d}U \wedge \omega_q \tag{6.28}$$

$$= U \underbrace{\left[\mathrm{d} + \frac{\mathrm{d}U}{U}\wedge\right]}_{\nabla} \omega_q\,. \tag{6.29}$$

In the last line we defined the *covariant derivative* ∇. Inserting the twist U from (6.25), it can be represented as

$$\nabla = \mathrm{d} + \beta_0 \frac{\mathrm{d}g(z|x)}{g(z|x)} \wedge - \sum_{i=1}^{n} \beta_i \frac{\mathrm{d}x_i}{x_i} \wedge \,. \tag{6.30}$$

We choose to focus on integrals $\int_\Gamma U\omega_q$ such that

$$\begin{array}{rcrcl} \Gamma \text{ is a } \textit{twisted cycle} & \iff & \partial_U \Gamma & = & 0 \\ \omega_q \text{ is a } \textit{twisted co-cycle} & \iff & \nabla\omega_q & = & 0\,. \end{array}$$

Following the same logic as before, we infer that the integral in question only depends on the equivalence classes $[\Gamma]$ and $[\omega_q]$, where $\Gamma \sim \Gamma' \iff \Gamma = \Gamma' + \partial_U \gamma$ and $\omega_q \sim \omega_q' \iff \omega_q = \omega_q' + \nabla\psi$.

We won't need the twisted homology group in what follows, so we skip that definition. To define the *twisted de Rham cohomology group*, we need to be careful in distinguishing between the x- and z-variables appearing in the integrand. Let

$$X = \left\{(z,x) \in \mathbb{A}^N \times (\mathbb{C}^*)^n \;\middle|\; g(z|x) \neq 0\right\} \quad \text{and} \quad Y = \mathbb{A}^N\,, \tag{6.31}$$

where $\mathbb{A}$ is the complex affine line (essentially $\mathbb{C}$ without a distinguished choice of origin), and $\mathbb{C}^*$ is the complex torus (which equals $\mathbb{C}\setminus\{0\}$ as a set). The space of holomorphic functions on X (technically a *sheaf*) is written as

$$\mathcal{O}(X) = \mathbb{C}\left[z_1,\ldots,z_N,x_1^{\pm 1},\ldots,x_n^{\pm 1},\frac{1}{g}\right]. \tag{6.32}$$

For $1 \leq k \leq n$, ω_q from (6.25) should be thought of as a differential k-form in the x-variables (i.e. we allow for $\mathrm{d}^k x$ but not $\mathrm{d}^k z$), with a coefficient function that lives in $\mathcal{O}(X)$. These objects are called *relative k-forms* in the literature [23, Section 1.5] [110, Section 4.1], and they live in the space[4]

$$\Omega^k_{X/Y} \;=\; \bigoplus_{\substack{K\subset\{1,\ldots,n\}\\ |K|=k}} \mathcal{O}(X)\,\mathrm{d}x^K\,. \tag{6.33}$$

The covariant derivative $\nabla_k : \Omega^k_{X/Y} \to \Omega^{k+1}_{X/Y}$ thus sends a relative k-form to a relative

[4] The subscript X/Y does not denote a quotient. This is merely the standard notation for the space of relative k-forms.

$(k+1)$-form. A calculation shows that

$$\nabla_k \circ \nabla_{k-1} = 0 \qquad \Longrightarrow \qquad \text{Image}(\nabla_{k-1}) \;\subset\; \text{Kernel}(\nabla_k)\,. \tag{6.34}$$

The kth *twisted de Rham cohomology group* can at last be defined as

$$\mathbb{H}^k_A(\beta) = \frac{\text{Kernel}(\nabla_k)}{\text{Image}(\nabla_{k-1})}\,. \tag{6.35}$$

We have emphasized the dependence on the A-matrix and the β-parameter vector, as they are used to build the integrand (6.25).

$\mathbb{H}^k_A(\beta)$ is endowed with the structure of a finite-dimensional vector space [111]. When β is generic enough (loosely speaking, it should not be integral), then the dimension of $\mathbb{H}^k_A(\beta)$ equals number of roots of the *likelihood equations* [112, 113]

$$\frac{\beta_0}{g(z|x)}\frac{\partial g(z|x)}{\partial x_i} - \frac{\beta_i}{x_i} = 0 \quad \text{for} \quad i = 1, \ldots, n\,. \tag{6.36}$$

The integer obtained in this fashion in fact equals the *Euler characteristic* χ of the space $(\mathbb{C}^*)^n \setminus \{x_1 \cdots x_n \cdot g(z|x) = 0\}$. For FIs, this integer even corresponds to the number of MIs [114][5]! What is more, this counting works whether or not the z's take on special values. In the GKZ case, these variables are required to be independent and indeterminate. But one is often interested in situations where some of the z's are equal to each other, and others equal unity. The dimension of $\mathbb{H}^k_A(\beta)$ will generally change when the z's are fixed to special values, but the counting via (6.36) will still be valid.

$\mathbb{H}^k_A(\beta)$ can also be endowed with the structure of a $\mathcal{D}$-module. For generic z and β, it is in fact isomorphic to the GKZ system as a $\mathcal{D}$-module! That is,

$$\mathcal{D}_N / H_A(\beta) \quad \cong \quad \mathbb{H}^k_A(\beta)\,. \tag{6.37}$$

Appendix (A.4) presents a rough sketch for the proof of this claim. Here we simply take this fact as given, and in the next subsection we leverage it to map elements from the RHS of the isomorphism to elements on the LHS.

[5] Albeit without taking symmetry relations into account, which arise when different momentum routings result in the same FI.

6.3.3 Representing Euler integrals inside $\mathcal{D}$-modules

The isomorphism (6.37) sends the equivalence class of operators $[1] \in \mathcal{D}/H_A(\beta)$ to the equivalence class of differential forms $\left[\frac{\mathrm{d}x}{x}\right] \in \mathbb{H}^k_A(\beta)$. Suppose we are given some other equivalence class $[\omega(z)]$. In this section we shall construct an operator d with the property

$$d \bullet \left[\frac{\mathrm{d}x}{x}\right] = [\omega(z)]\,. \tag{6.38}$$

Given a vector $q = (q_0, q') \in \mathbb{Z} \times \mathbb{Z}^n$, recall the expression for ω_q from (6.25). For some fixed Γ, the Euler integral associated to this differential form is denoted by

$$\langle \omega_q \rangle = \int_\Gamma U \omega_q\,. \tag{6.39}$$

By differentiating w.r.t. z_i, then q gets shifted by the ith column vector of the A-matrix:

$$\partial_i \bullet \langle \omega_q \rangle = (\beta_0 - q_0)\langle \omega_{q+a_i} \rangle\,. \tag{6.40}$$

There exists an operator which shifts q in the opposite direction as well, i.e. by $-a_i$. To set this up, the authors of [115] construct a so-called *creation operator* C_i and a *b-function* $b_i(\beta)$ such that

$$C_i\,\partial_i - b_i(\beta) = 0 \mod H_A(\beta)\,. \tag{6.41}$$

Using that $\langle \omega_q \rangle$ is a solution to $H_A(\beta - q)$, one can then derive that

$$C_i \bullet \langle \omega_q \rangle = b_i(\beta - a_i)\langle \omega_{q-a_i} \rangle\,. \tag{6.42}$$

Suppose that an integral $\langle \omega_q \rangle$ is given. Since the vectors a_i span $\mathbb{Z}^{n+1}$ by stipulation, it is possible to write the q-vector for $\langle \omega_q \rangle$ as

$$q = \sum_{i=1}^N r_i\,a_i\,, \quad r_i \in \mathbb{Z}\,. \tag{6.43}$$

The integers r_i allow us to balance shifts by $\pm a_i$ in the q-vector of $\langle \omega_q \rangle$ via (6.40) and (6.42). It thereby becomes possible to erect the operator

$$d_q = \prod_{r_i<0} C_i^{-r_i} \prod_{r_i>0} \frac{1}{B(\beta)B'(\beta)} \partial^{r_i} \tag{6.44}$$

satisfying[6]

$$d_q \bullet \langle \omega_0 \rangle = \langle \omega_q \rangle \,, \quad \omega_0 = \frac{\mathrm{d}x}{x} \,. \tag{6.45}$$

The two functions $B(\beta)$ and $B'(\beta)$ are built from the prefactors of (6.40) and (6.42). Their explicit formulas are somewhat lengthy, so we refer to [116] for details. The formula (6.44) has been implemented in the ASIR package MT_GKZ; see Example (6.1) below.

Note that the vector $r = [r_1, \dots, r_N]^T \in \mathbb{Z}^N$ from (6.43) is not unique for fixed q. Given a collection of vectors $u_i \in \mathbb{Z}^N$ which lie in $\ker(A)$, then it is the case that $q = A \cdot \left(r + \sum_i c_i\, u_i\right)$ for any choice of $c_i \in \mathbb{Z}$. This freedom can be exploited to simplify the operator d_q from (6.44). In particular, the creation operators C_i tend to have a large degrees in ∂_i, so for practical purposes it is useful to adjust the coefficients c_i such that $r_i > 0$, in which case the first product of (6.44) drops out.

Later on when we discuss applications to FIs, the q-vector will be used to specify a particular integral within a family. In the Lee-Pomeransky representation of a FI (shown in the future equation (7.3)), q_0 is related to the spacetime dimension, and q' contains the propagator powers.

Example 6.1. Consider the GKZ system for the 4-fold Euler integral

$$f_\beta(z) = \int_\Gamma \left(z_1 x_1 + z_2 x_2 + z_3 x_3 + z_4 x_4 + z_5 x_1 x_3 + z_6 x_2 x_4\right)^{\beta_0} x^{-\beta'} \frac{\mathrm{d}x}{x} \,. \tag{6.46}$$

The goal of this example is to build the operator d_q corresponding to the differential form

$$\omega_q = \left[\frac{x_1 x_2}{g^2} \frac{\mathrm{d}x}{x}\right] \quad \text{with} \quad q = (2, 1, 1, 0, 0) \,. \tag{6.47}$$

We shall gain more insight from deriving d_q by hand rather using the closed formula (6.44). The idea is to take suitable derivatives of $f_\beta(z)$ w.r.t. z_i. Since ω_q has a g^2 in the denominator, we ought to take two derivatives to decrease $\beta_0 \to \beta_0 - 2$. The powers of x_1 and x_2 in ω_q are obtained by specifically choosing derivatives w.r.t. z_1 and z_2. So we have

$$\partial_1 \partial_2 \bullet f_\beta(z) = \beta_0(\beta_0 - 1) \int_\Gamma g(z|x)^{\beta_0 - 2} x^{-\beta'} x_1 x_2 \frac{\mathrm{d}x}{x} \,. \tag{6.48}$$

[6] Contrary to what was promised in the early remarks of this section, this equation is formulated as an action on a integral $\langle \omega_q \rangle$ rather than on a differential form $[\omega(z)]$. The action of a differential operator on $[\omega(z)]$ follows from (A.42), but the formula for d_q remains the same in that case.

Rearranging factors,

$$\frac{\partial_1\partial_2}{\beta_0(\beta_0-1)} \bullet f_\beta(z) = \int_\Gamma g(z|x)^{\beta_0} x^{-\beta'} \left[\frac{x_1 x_2}{g^2}\frac{\mathrm{d}x}{x}\right] . \tag{6.49}$$

The $\mathcal{D}$-module representative for the differential form ω_q is therefore $d_q = \frac{\partial_1\partial_2}{\beta_0(\beta_0-1)}$.

The same computation can be performed in ASIR via the following script:

```
import("mt_gkz.rr");

A = [
        [1,1,1,1,1,1]
        [1,1,0,0,0,0]
        [0,0,1,1,0,0]
        [1,0,0,0,1,0]
        [0,0,1,0,0,1]
];
Ap = mt_gkz.make_Ap(A, 1);
Ap = vtol(Ap);
Beta = [b0,b1,b2,b3,b4];

Rvec = [1,1,0,0,0,0];
mt_gkz.ff(Rvec, A, Ap, Beta);
```

The first row of `Ap` consists of 1's, and the remaining rows contain only 0's[7]. The r-vector `Rvec` satisfies $q = A \cdot r$. The final command `mt_gkz.ff(...)` outputs the result `(dx1*dx2)/(b0^2-b0)`, as expected. ■

To summarize, we now have three ways of representing *the same* object:

1. The Euler integral $\langle\omega_q\rangle = \int_\Gamma U\omega_q$, written in the notation of (6.25). This lives in some suitable space of functions.

2. The equivalence class of differential forms $[\omega_q]$. This lives in the twisted cohomology group (6.35).

3. The equivalence class of differential operators $[d_q]$ defined by formula (6.44). It lives inside the $\mathcal{D}$-module $\mathcal{D}_N/H_A(\beta)$.

[7] `Ap` is built from the vectors called $\mathbf{a}'_i$ in [116]. See the documentation of MT_GKZ for additional details.

Section 6.4

Basis of standard monomials

As the GKZ $\mathcal{D}$-module is holonomic, it must have a finite collection of standard monomials serving as a basis. There is a simple formula for the size of this basis given a genericity condition on the parameter vector $\beta = (\beta_0, \ldots, \beta_n)$. We require some definitions from polyhedral geometry to state the formula.

- Interpret the column vectors a_i of the A-matrix as points in $\mathbb{Z}^{n+1}$. The positive real *cone* generated by these points is defined as

$$\mathbf{C}[A] = \left\{ \sum_{i=1}^{N} \lambda_i \, a_i \;\middle|\; \lambda_i \geq 0 \right\} . \tag{6.50}$$

- We say that β is *non-resonant* if any integer shift $\beta + \mathbb{Z}^{n+1}$ contains no points on the *facets* of $\mathbf{C}[A]$. Facets are defined as codimension-1 faces of $\mathbf{C}[A]$.

  Alternatively, we can say that β is resonant when some integer translate of the facet hyperplanes for $\mathbf{C}[A]$ can be made to "hit" β.

  While this definition is somewhat technical, in practice it just means that β should be "generic enough", in the sense of not containing integer values.

- The *Newton polytope* associated to A is

$$\mathbf{N}[A] = \left\{ \sum_{i=1}^{N} \lambda_i \, a_i \;\middle|\; \lambda_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{N} \lambda_i = 1 \right\} . \tag{6.51}$$

  In other words, $\mathbf{N}[A]$ is the convex hull of A's column vectors $a_i \in \mathbb{Z}^{n+1}$.

When β is non-resonant, it was proved in [95] that the holonomic rank of the GKZ system, and hence also the number of standard monomials, is given by

$$R = n! \, \times \, \mathrm{vol}(\mathbf{N}[A]) \, , \tag{6.52}$$

where vol($\mathbf{N}[A]$) is the volume of the Newton polytope (this can be efficiently computed with software such as Polymake [117]). This formula is quite remarkable. Recall that R counts the number of independent solutions to the PDEs (6.7)-(6.8). The latter is statement of complex analysis, while the formula (6.52) is purely combinatorial.

For GKZ systems, not only is the number of basis elements known a priori, there is even a fast algorithm to determine *explicit* standard monomials due to Hibi, Nishiyama and Takayama [118]. The idea of the algorithm is to compute standard monomials for a simpler, auxiliary ideal. The construction can be summarized as follows. Given the toric ideal of a GKZ system

$$\mathcal{I}_\square = \left\langle \square_u \right\rangle_{\forall u \in \ker(A)} , \tag{6.53}$$

one must first build its ideal of initial terms $\mathrm{in}_\prec(\mathcal{I}_\square)$ (recall (4.21)). Any generator d of $\mathrm{in}_\prec(\mathcal{I}_\square)$ can be written in the form $d = \prod_{i=1}^N \partial^{q_i}$ for $q_i \in \mathbb{Z}_{\geq 0}$ The *distraction* of d is defined by

$$\mathrm{dist}(d) = \prod_{i=1}^{N} \partial_i(\partial_i - 1) \cdots (\partial_i - q_i - 1) \,. \tag{6.54}$$

The authors of [118] then consider the ideal $\mathcal{I}'$ generated as

$$\mathcal{I}' \quad = \quad \sum_{i=0}^{n} \mathcal{D}\, E_i\big|_{z_1 = \cdots = z_N = 1} \;+ \sum_{\forall d \,\in\, \mathrm{in}_\prec(\mathcal{I}_\square)} \mathcal{D}\,\mathrm{dist}(d)\,, \tag{6.55}$$

where E_i are the homogeneity operators from (6.5). Surprisingly, the set of standard monomials for $\mathcal{D}/\mathcal{I}'$ also constitutes a basis for the GKZ system $\mathcal{D}/H_A(\beta)$. Hibi et al. thus obtain standard monomials for the latter by a Gröbner basis computation in the former. Crucially, $\mathcal{I}'$ is an ideal in the *commutative* subring $\mathbb{C}[\partial_1, \ldots, \partial_N]$, which makes the computation much more efficient than the execution of Buchberger's algorithm in a non-commutative Weyl algebra.

Example 6.1. To illustrate the efficiency of this algorithm, consider the following A-matrix formed from a polynomial with $N = 26$ terms:

$$A = \begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&1&1&1&1&1&\cdot&\cdot&\cdot&\cdot&\cdot\\
1&\cdot&\cdot&\cdot&1&1&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&1&\cdot&\cdot&\cdot&\cdot&1&1&1&1&\cdot\\
\cdot&1&\cdot&\cdot&1&\cdot&\cdot&1&1&1&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&1&\cdot&\cdot&1&1&\cdot&\cdot&1\\
\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&1&\cdot&\cdot&\cdot&1&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&1\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&1&\cdot&\cdot&\cdot&\cdot&1&1&1&\cdot&1&\cdot&1&1&1&\cdot&\cdot&1&\cdot\\
\cdot&\cdot&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&1&\cdot&1&\cdot&\cdot&1&\cdot&1&1&\cdot&\cdot&1&1&1&\cdot\\
\cdot&\cdot&\cdot&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&1&\cdot&1&\cdot&\cdot&\cdot&\cdot&\cdot&1&\cdot&\cdot&\cdot&\cdot&1
\end{bmatrix} .$$

The algorithm of [118] has been implemented by Nobuki Takayama in ASIR via the command

```
mt_gkz.cbase_by_euler(A);
```

In less than a minute on a laptop, this command gives a list of $R = 238$ standard monomials for the GKZ system. The first few elements are

$$\mathrm{Std} = \left\{\partial_{26}\partial_{25}^4,\, \partial_{26}^4\partial_{14},\, \partial_{25}^2\partial_{24}^2,\, \partial_{25}^3\partial_8,\, \partial_{25}^3\partial_{12},\, \dots\right\}. \tag{6.56}$$

■

SECTION 6.5

Integrand rescaling

By cleverly rescaling the z-variables of the Euler integral (6.1), we show here how to reduce the number of variables from N to $N-(n+1)$. This reduction is useful for practical computations.

The construction follows from a homogeneity property satisfied by Euler integrals:

$$f_\beta(t^{a_1}z_1, \dots, t^{a_N}z_N) = t_0^{\beta_0}\, t_1^{\beta_1} \cdots t_n^{\beta_n} f_\beta(z)\,, \tag{6.57}$$

where the vectors $a_i \in \mathbb{Z}^{n+1}$ come from the columns of the A-matrix. By differentiating w.r.t. t and using the chain rule, it can be shown that this property is equivalent to the collection of PDEs $E_i \bullet f_\beta(z) = 0$ from (6.5). This is the reason for calling E_i the "homogeneity operators".

We can freeze $(n+1)$ of the z-variables to unity by judiciously choosing the rescaling parameters $t \in \mathbb{C}^{n+1}$. This is best illustrated by an example.

Example 6.1. This is a continuation of Example (5.2). After relabeling $\beta_i \to \beta_{i-1}$ in the generators called d_i in (5.32), we now see that this is the GKZ system built from A-matrix

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{6.58}$$

The associated Euler integral is

$$f_\beta(z) = \int_\Gamma \left(z_1 + z_2x_1 + z_3x_2 + z_4x_1x_2\right)^{\beta_0} x_1^{-\beta_1} x_2^{-\beta_2} \frac{\mathrm{d}x}{x}\,. \tag{6.59}$$

In this case, the homogeneity property (6.57) reads

$$f_\beta(t_0 z_1,\, t_0 t_1 z_2,\, t_0 t_2 z_3,\, t_0 t_1 t_2 z_4) = t_0^{\beta_0} t_1^{\beta_1} t_2^{\beta_2} f_\beta(z_1, z_2, z_3, z_4)\,. \tag{6.60}$$

Choosing

$$t_0 = \frac{1}{z_1}\,, \quad t_1 = \frac{z_1}{z_2}\,, \quad t_2 = \frac{z_1}{z_3}\,, \tag{6.61}$$

then (6.60) becomes

$$f_\beta(1,\, 1,\, 1, w) = z_1^{-\beta_0+\beta_1+\beta_2} z_2^{-\beta_1} z_3^{-\beta_2} f_\beta(z_1, z_2, z_3, z_4) \quad \text{with} \quad w = \frac{z_1\, z_4}{z_2\, z_3}\,. \tag{6.62}$$

A 4-variable problem has thereby been reduced to one that depends on a single cross-ratio variable w.

The three homogeneity operators $\{E_0, E_1, E_2\}$ have now been "gauge fixed" away by this rescaling. The GKZ system then only depends on a single toric operator:

$$\Box_u \bullet f_\beta(z) = (\partial_1\partial_4 - \partial_2\partial_3) \bullet f_\beta(z) = 0\,. \tag{6.63}$$

Inserting the representation of f_β from the RHS of (6.62), a short calculation shows that the two terms in $\Box_u$ get mapped to

$$\begin{aligned} \partial_1\partial_4 &\;\to\; -w^{-1}(\beta_0 - \beta_1 - \beta_2 - w\partial_w)(w\partial_w) \\ \partial_2\partial_3 &\;\to\; (\beta_1 + w\partial_w)(\beta_2 + w\partial_w)\,. \end{aligned} \tag{6.64}$$

The toric PDE (6.63) thus turns into the second-order ODE

$$\begin{aligned} &L \bullet f_\beta(1, 1, 1, w) = 0 \quad \text{with} \\ &L = w(1-w)\partial_w^2 + \big[1 - \beta_0 + \beta_1 + \beta_2 - (1 + \beta_1 + \beta_2)w\big]\partial_w - \beta_1\beta_2\,, \end{aligned} \tag{6.65}$$

which is recognized to be the DEQ for Gauss' hypergeometric ${}_2F_1$ function. ■

The rescaling parameters (6.61) were here pulled out of a hat. In Appendix (A.3), we present a general formula for how to choose these t's.

Section 6.6

Recurrence relations

This section describes a method, relying solely on products of Pfaffian matrices, to determine recurrence relations among Euler integrals with different values of β. (We remark that these matrix products can be swiftly calculated using rational reconstruction.) One can think of these recurrence relations as being a counterpart to IBP for FIs, though in this case the IBPs come first and the Pfaffian systems last. That logic is reversed by the method presented here, as we obtain Pfaffian matrices first (via the Macaulay matrix method), and then use those Pfaffians to derive recurrence relations/IBPs.

6.6.1 Shifting β by matrix factorials

For convenience, we shall work with a rescaled version of the Euler integral:

$$f_\beta(z) \to f(\beta) = \frac{1}{\Gamma(\beta_0+1)} \int_\Gamma g(z|x)^{\beta_0} x^{-\beta'} \frac{\mathrm{d}x}{x}\,, \tag{6.66}$$

where we chose to only emphasize the β-dependence of $f(\beta)$ since z will stay fixed throughout the discussion. The Γ-function prefactor provides a clean shift relation when a derivative is taken w.r.t. z_i:

$$\partial_i \bullet f(\beta) = f(\beta - a_i)\,. \tag{6.67}$$

Let Std be a basis of standard monomials for a GKZ system $\mathcal{D}/H_A(\beta)$. We convert this $\mathcal{D}$-module basis into a basis of functions by writing

$$F(\beta) = \begin{bmatrix} \mathrm{Std}_1 \bullet f(\beta) \\ \vdots \\ \mathrm{Std}_R \bullet f(\beta) \end{bmatrix}. \tag{6.68}$$

Suppose we have computed the Pfaffian system for $F(\beta)$ via the Macaulay matrix method. Combining the relation $\partial_i \mathrm{Std}_j = \mathrm{Std}_j \partial_i$ with (6.67), we deduce that

$$\partial_i \bullet F(\beta) = P_i(\beta) \cdot F(\beta) = F(\beta - a_i)\,. \tag{6.69}$$

In other words, matrix multiplication by $P_i(\beta)$ induces a difference equation for $F(\beta)$.

It is possible to shift β in the opposite direction as well. Assuming β is non-resonant, then $P_i(\beta + a_i)$ is invertible. Setting

$$Q_i(\beta) = P_i(\beta + a_i)^{-1}\,, \tag{6.70}$$

we thereby obtain an "inverse derivative operator" ∂_i^{-1} acting as

$$\partial_i^{-1} \bullet F(\beta) = Q_i(\beta) \cdot F(\beta) = F(\beta + a_i)\,. \tag{6.71}$$

Let $b \in \mathbb{Z}_{\geq 0}$. If $b > 0$, we write the *falling matrix factorial* as

$$\big[P_i(\beta)\big]_b = P_i\big(\beta - (b-1)a_i\big) \cdot P_i\big(\beta - (b-2)a_i\big) \cdots P_i(\beta - a_i) \cdot P_i(\beta)\,, \tag{6.72}$$

and the *rising matrix factorial* as

$$\big(Q_i(\beta)\big)_b = Q_i\big(\beta + (b-1)a_i\big) \cdot Q_i\big(\beta + (b-2)a_i\big) \cdots Q_i(\beta + a_i) \cdot Q_i(\beta)\,. \tag{6.73}$$

If $b = 0$, then $\big[P_i(\beta)\big]_0 = \big(Q_i(\beta)\big)_0 = \mathbf{1}$.

When the subscript b is promoted to an integer vector $v \in \mathbb{Z}^N_{\geq 0}$, these definitions are extended to

$$\big[P(\beta)\big]_v = \prod_{i=1}^{N} \left[P_i\left(\beta - \textstyle\sum_{j=i+1}^{N} v_j\, a_j\right)\right]_{v_i} \tag{6.74}$$

$$\big(Q(\beta)\big)_v = \prod_{i=1}^{N} \left[Q_i\left(\beta - \textstyle\sum_{j=i+1}^{N} v_j\, a_j\right)\right]_{v_i}\,, \tag{6.75}$$

with the convention that $\sum_{j=i+1}^{N} v_j a_j = 0$ when $i = N$.

Now, by induction over $b > 0$, it is short proof (see [82, Lemma 6.3]) to show that

$$\begin{array}{rclcl} \partial_i^{b} \bullet F(\beta) & = & \big[P_i(\beta)\big]_b \cdot F(\beta) & = & F(\beta - b a_i) \\ \partial_i^{-b} \bullet F(\beta) & = & \big(Q_i(\beta)\big)_b \cdot F(\beta) & = & F(\beta + b a_i)\,. \end{array} \tag{6.76}$$

Repeatedly applying these relations for each v_i in a given vector $v \in \mathbb{Z}^N_{\geq 0}$, we obtain the general shift relations

$$\begin{array}{rclcl} \partial_i^{v} \bullet F(\beta) & = & \big[P_i(\beta)\big]_v \cdot F(\beta) & = & F\left(\beta - \sum_{i=1}^{N} v_i\, a_i\right) \\ \partial_i^{-v} \bullet F(\beta) & = & \big(Q_i(\beta)\big)_v \cdot F(\beta) & = & F\left(\beta + \sum_{i=1}^{N} v_i\, a_i\right)\,. \end{array} \tag{6.77}$$

Note that these matrix factorial relations are not uniquely represented because

$$\begin{array}{rcl} F\big((\beta - a_i) - a_j\big) & = & P_i(\beta - a_j) \cdot P_j(\beta) \cdot F(\beta) \\ & \text{and} & \\ F\big((\beta - a_j) - a_i\big) & = & P_j(\beta - a_i) \cdot P_i(\beta) \cdot F(\beta) \end{array} \tag{6.78}$$

imply the commutation relation

$$P_i(\beta - a_i) \cdot P_j(\beta) = P_j(\beta - a_i) \cdot P_i(\beta)\,, \tag{6.79}$$

and likewise for Q_i.

6.6.2 General β shift algorithm

Here we employ the formulas derived in the previous subsection to state a general algorithm for expressing a given Euler integral as a sum of basis integrals. To begin, write the standard monomials in the form $\mathrm{Std}_i = \partial^{q_i}$ for some $q_i \in \mathbb{Z}^N_{\geq 0}$. Recalling that the ith component of $F(\beta)$ is of the form $\mathrm{Std}_i \bullet f(\beta)$, then (6.67) implies that

$$F(\beta) = \begin{bmatrix} f(\beta - A \cdot q_1) \\ \vdots \\ f(\beta - A \cdot q_R) \end{bmatrix}. \tag{6.80}$$

Given an integer vector $q_0 \in \mathbb{Z}^N$, we seek a recurrence relation for $f(\beta - A \cdot q_0)$ of the form

$$f(\beta - A \cdot q_0) = \sum_{i=1}^{R} c_i(\beta, z)\, f(\beta - A \cdot q_i) \quad \text{with} \quad c_i \in \mathbb{Q}(\beta, z)\,. \tag{6.81}$$

Observe that $f(\beta - A \cdot q_0)$ is the first element of the vector $F\big(\beta - A \cdot (q_0 - q_1)\big)$. Combining this fact with the shift relations (6.77), we are lead to the following algorithm.

```
Algorithm 2 : Recurrence relations for Euler integrals
Input:
  • Vector $q_0 \in \mathbb{Z}^N$.
  • Indeterminate vector $\beta$.
  • Monomial basis $\mathrm{Std} = \{\partial^{q_1}, \dots, \partial^{q_R}\}$ for $q_i \in \mathbb{Z}^N_{\geq 0}$.
Output: The recurrence relation (6.81).
1: Calculate $P_i(\beta)$, $i = 1, \dots, N$, w.r.t. the basis Std by calling Algorithm (1).
2: Decompose
       $$q_0 - q_1 = q^+ - q^- = q$$
   for $q^\pm \in \mathbb{Z}^N_{\geq 0}$.
3: Compute the matrix factorial
       $$\Big[P\Big(\beta - \textstyle\sum_{q_j<0} q_j\, a_j\Big)\Big]_{q^+} \;\cdot\; \big(Q(\beta)\big)_{q^-} \;\cdot\; F(\beta)\,.$$
4: return the first element of step 3.
```

Since this algorithm avoids the use of derivatives, it is possible to fix the z-variables to special values in the Pfaffian matrices. We give a simple example of how to use this algorithm in Section (7.4).

Chapter 7

Generalized Feynman Integrals

In the physics literature, Tullio Regge observed already in 1968 that FIs often evaluate to *hypergeometric functions* [119] (for instance the ${}_2F_1$ function that we saw in the previous chapter). In the mathematics literature, although a large zoo of hypergeometric[1] functions (carrying famous names such as Gauss, Kummer, Mellin, Barnes, Appell, Kampé de Fériet, Lauricella, Horn and Meijer) were known in the last half of the previous century, there was arguably no overarching framework that tied them all together. The work of Gel'fand, Kapranov and Zelevinsky in the 1990s did exactly that: it gave a general construction which subsumed the whole zoo of hypergeometric functions as special cases.

Following in the footsteps of Regge, it is then natural to wonder whether GKZ systems can say something about FIs. The first proper study in this direction appears to have been by Nasrollahpoursamami in 2016 [28, 120]. There has since then been a flurry of works studying this topic [82, 83, 113, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143]. There is now even a practical program, named FeynGKZ [123], that computes series expansions for FIs based on the GKZ framework.

The connection between FIs and GKZ systems is not so immediate in the momentum space representation. The connection becomes more clear in parametric representations such as the *Lee-Pomeransky representation* (LPr) [43]. The FI then looks more like the Euler integral defined in (6.1), which is a solution to the GKZ system. The difference is that the z-variables take on special values in the LPr, so the class of Euler integrals is too general. Though the proper function space for FIs is unknown (cf. Section (2.1.3)), the Euler integral can thus be thought of as an "upper bound" on the complexity of this space.

In what follows, we use the technology developed in the preceding chapters to study some simple FIs in the GKZ framework. We especially focus on Pfaffian systems for MIs.

[1] Let $F(z) = \sum_{n=0}^{\infty} c_n z^n$. We say that F is *hypergeometric* if c_{n+1}/c_n is a rational function of n. The generalization to multivariate and logarithmic series is immediate.

Section 7.1

Lee-Pomeransky representation as a GKZ system

Fix a Feynman graph G. There are two special *Symanzik polynomials* associated to G, given by

$$\mathcal{U}(x) = \sum_{T} \prod_{e \notin T} x_e \tag{7.1}$$

$$\mathcal{F}(x) = -\sum_{F} p(F)^2 \prod_{e \notin F} + \mathcal{U}(x) \sum_{i=1}^{n} m_i^2 x_i \,. \tag{7.2}$$

T and F are sets of so-called *spanning trees* and *spanning forests* respectively, and $p(F)^2$ is the Minkowski square of the momentum flowing into certain forest components. We shall not dwell on these definitions here, as there are many exhaustive expositions already, see e.g. [44, Chapter 3]. The parameters m_i denote the internal masses associated to the n internal edges of G.

Starting from the momentum space representation of a FI, it is possible to derive an equivalent parametric representation called the *Lee-Pomeransky representation* (LPr) [43] (see also [44, Section 2.5]). We denote it by

$$I(\mathrm{D}_0|\nu) = \frac{\Gamma\left(\frac{\mathrm{D}}{2}\right)}{\Gamma\left(\frac{\mathrm{D}}{2} - \omega\right)\Gamma(\nu)} \int_0^\infty \mathcal{G}(x)^{-\frac{\mathrm{D}}{2}}\, x^{\nu}\, \frac{\mathrm{d}x}{x} \,, \tag{7.3}$$

where

- $\mathcal{G}$ is the *Lee-Pomeransky* (LP) *polynomial*

$$\mathcal{G}(x) = \mathcal{U}(x) + \mathcal{F}(x) \,. \tag{7.4}$$

- ω is the *superficial degree of divergence*

$$\omega = \sum_{i=1}^{n} \nu_e - \mathrm{D}L/2 \tag{7.5}$$

  written in terms of edge weights $\nu = (\nu_1, \ldots, \nu_n) \in \mathbb{Z}^n$, the spacetime dimension $\mathrm{D} = \mathrm{D}_0 - 2\epsilon$, and the number of loops L. We assume $\mathrm{D}_0 \in 2 \cdot \mathbb{Z}_{>0}$ is even for simplicity.

- $\Gamma(\nu)$ is shorthand for $\prod_{i=1}^{n} \Gamma(\nu_i)$.

7.1.1 Generalized Feynman integral

Let $0 < \delta \ll 1$. The two small parameters (ϵ, δ) correspond to analytic regulators in the works of Speer [144, 145]. Fix the exponents β of the Euler integral $f_\beta(z)$ from (6.1) to[2]

$$\beta = (\epsilon,\, -\epsilon\delta,\, \ldots, -\epsilon\delta) \;\; - \;\; (\mathrm{D}_0/2,\, \nu_1, \ldots,\, \nu_n)\,. \tag{7.6}$$

The *generalized FI* (GFI) is now defined to be

$$I(\mathrm{D}_0|\nu) = c(\mathrm{D}_0|\nu) \times f_\beta(z)\,. \tag{7.7}$$

- Inserting the special choice of β from (7.6), the Euler integral reads

  $$f_\beta(z) = \int_{(0,\infty)^n} \mathcal{G}(z|x)^{\epsilon - \mathrm{D}_0/2}\, x_1^{\nu_1 + \epsilon\delta} \cdots x_n^{\nu_n + \epsilon\delta}\, \frac{\mathrm{d}x}{x}\,, \tag{7.8}$$

  where the integration contour has been fixed to $\Gamma = (0, \infty)^n$.

- Rewriting the prefactor in (7.3) with this β, we have

  $$c(\mathrm{D}_0|\nu) = \frac{\Gamma\big(\mathrm{D}_0/2 - \epsilon\big)}{\Gamma\Big((L+1)(\mathrm{D}_0/2 - \epsilon) - |\nu| - n\epsilon\delta\Big) \prod_{i=1}^n \Gamma\big(\nu_i + \epsilon\delta\big)}\,, \tag{7.9}$$

  with $|\nu| = \nu_1 + \ldots + \nu_n$.

- The monomials x^{α_i} in $\mathcal{G}$ come from the LP polynomial (7.3):

  $$\mathcal{G}(z|x) = \mathcal{U}(z|x) + \mathcal{F}(z|x) = \sum_{i=1}^N z_i\, x^{\alpha_i}\,. \tag{7.10}$$

  But, in contrast to the original LP polynomial, each monomial coefficient z_i is here taken to be indeterminate.

There is a GKZ system $\mathcal{D}/H_A(\beta)$ associated to such a GFI. According to the isomorphism (6.37), $I(\mathrm{D}_0|\nu)$ is also represented by a twisted cohomology group $\mathbb{H}_A^k(\beta)$. In the notation of Section (6.3.3), given a twist $U = \mathcal{G}(z|x)^\epsilon\, x^{\epsilon\delta}$ and a differential form

$$[\omega_{\mathrm{D}_0/2,\nu}] = c(\mathrm{D}_0|\nu) \times \mathcal{G}^{-\mathrm{D}_0/2}\, x^\nu\, \frac{\mathrm{d}x}{x}\,, \tag{7.11}$$

[2] Writing $-\epsilon\delta$ instead of, say, δ is merely a matter of convention. This convention was chosen so as to more easily compare results from Section (7.2) with [101].

we then have that $I(\mathrm{D}_0|\nu) = \langle \omega_{\mathrm{D}_0/2,\nu} \rangle$. The r-vector from (6.43) therefore satisfies $A \cdot r = \big(\mathrm{D}_0/2, \nu\big)$. The columns $a_i = [1, \alpha_i]^T$ of this A-matrix come from the multi-indices α_i in (7.10).

Now, making the identifications

- $\delta \to 0$
- $z_i \to \mathbb{Z}_{>0}$ or $\big\{$kinematic variables $(m_i^2, p_i^2, p_i \cdot p_j)\big\}$

then a GFI agrees exactly with the LPr. The $\delta \to 0$ limit is not so troublesome (this parameter is inserted to ensure non-resonance of the GKZ system). The more pressing issue is how to deal with each z_i being indeterminate in the GFI. By the homogeneity rescaling described in Section (6.5), we can at least rescale $(n+1)$ of the z-variables to unity in order to match with some or all of the monomial coefficients in the $\mathcal{U}$-polynomial. In doing so, it is possible to exactly match the GFI with a large class of 1-loop graphs [82, Section 5.1] - in particular those which are fully massless, or contain off-shell legs whose squared momenta are independent, $p_i^2 \neq p_j^2 \neq 0$.

In general, however, there are *more* GKZ variables than there are independent monomial coefficients in the proper LP polynomial. This is especially true at $L = 2$ loops and beyond. In those cases one must take a kind of "limit", called a *restriction*, of the GKZ system. We study such restrictions in detail in Chapters (8) and (9). For the remaining part of this chapter though, we only study examples where the number of GKZ variables can be made to match with the proper FI.

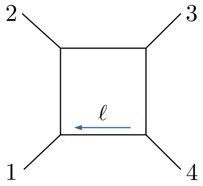
Section 7.2

Macaulay matrix example: 1-loop massless box

Here we compute the Pfaffian system for the 1-loop massless box diagram using the Macaulay matrix method from Chapter (5). This example was studied in the LPr using twisted cohomology in [101].

The kinematic configuration is

$$\sum_{i=1}^{4} p_i = 0\,, \quad p_1^2 = \cdots = p_4^2 = 0\,, \quad s = 2p_1 \cdot p_2\,, \quad t = 2p_2 \cdot p_3\,.$$



In momentum space, one has the following four inverse propagators:

$$\begin{aligned} D_1 &= -\ell^2\,, & D_2 &= -(\ell-p_1)^2\,, \\ D_3 &= -(\ell-p_1-p_2)^2\,, & D_4 &= -(\ell-p_1-p_2-p_3)^2\,. \end{aligned} \tag{7.12}$$

The corresponding GFI reads

$$I(\mathrm{D}_0|\nu) = c(\mathrm{D}_0|\nu)\int_{(0,\infty)^4} \mathcal{G}(z|x)^{\epsilon-\mathrm{D}_0/2}\, x_1^{\nu_1+\epsilon\delta}\cdots x_4^{\nu_4+\epsilon\delta}\,\frac{\mathrm{d}x}{x}\,, \tag{7.13}$$

where

$$\mathcal{G}(z|x) = \sum_{i=1}^{4} z_i x_i + z_5 x_1 x_3 + z_6 x_2 x_4\,. \tag{7.14}$$

Upon rescaling the integration variables in the proper LPr as $x \to x/(-s)$, then its monomial coefficients become

$$z_1 = \ldots = z_5 = 1\,, \quad z_6 = \frac{t}{s} = z\,. \tag{7.15}$$

So the proper LP has one independent coefficient z. On the other hand, the GKZ system has $N = 6$ variables. Using homogeneity, we can rescale $n+1 = 4+1 = 5$ of those variables to unity. To match with (7.15), we choose to rescale $\{z_1,\ldots,z_5\}$. Including an overall s-dependent prefactor because of the x-rescaling above,

$$I(\mathrm{D}_0|\nu) \to (-s)^{\mathrm{D}_0/2-\epsilon-4\epsilon\delta-|\nu|} \times I(\mathrm{D}_0|\nu)\,, \tag{7.16}$$

then our GFI matches exactly with the LPr in the limit $\delta \to 0$.

Having "gauged away" the Euler homogeneity operators, the GKZ system now only contains the toric part. Using (A.34) or the ASIR command `mt_gkz.gkz_b`, the single toric operator works out to

$$\begin{aligned} \mathcal{I}_\square = z^3[1+z]\partial^3 \;&+\; z^2\big[3(1+z)+\epsilon(2+z+6\delta(1+z))\big]\partial^2 \\ +\; z\big[(1+\epsilon+3\delta\epsilon)^2 + z(1+\epsilon+6\delta\epsilon+\epsilon^2\delta(2+9\delta))\big]\partial \;&+\; z\epsilon^3\delta^2[1+4\delta]\,, \end{aligned} \tag{7.17}$$

where $\partial = \partial_z$.

7.2.1 Basis

Since the number of GKZ variables matches with the proper LPr, the formula $R = n! \times \mathrm{vol}(\mathbf{N}[A])$ ought to correctly count the number of MIs. In polymake, this volume can be computed as

```
$P = new Polytope(POINTS =>
        [[1,1,0,0,0],
         [1,0,1,0,0],
         [1,0,0,1,0],
         [1,0,0,0,1],
         [1,1,0,1,0],
         [1,0,1,0,1]]
);
print $P -> VOLUME;
```

The matrix above is the transpose of the A-matrix associated to the LP polynomial (7.14), due to the conventions of the software. The script outputs the volume 1/8, so we expect $R = 4! \times 1/8 = 3$ MIs. This indeed holds true, as can be verified by an IBP calculation.

Taking inspiration from [146], we choose the following basis of canonical integrals near 4 dimensions:

$$e^{(I)} = (-s)^{\epsilon+1} \begin{bmatrix} z\, I(4|0102) \\ I(4|1020) \\ \epsilon\, z\, (-s)\, I(4|1111) \end{bmatrix} . \tag{7.18}$$

Using (7.11), this corresponds to the basis of differential forms

$$e^{(\mathrm{dR})} = \mathrm{c} \cdot \begin{bmatrix} \frac{x_4}{x_1\, x_2\, \mathcal{G}^2} \mathrm{d}x \\ \frac{x_3}{x_2\, x_4\, \mathcal{G}^2} \mathrm{d}x \\ \frac{1}{\mathcal{G}^2} \mathrm{d}x \end{bmatrix} , \tag{7.19}$$

where c is a 3×3 diagonal matrix containing the Γ-function constants $c(\mathrm{D}_0|\nu)$ multiplied by the prefactors from (7.18). The formulas (6.44) and (A.34) turn these differential forms into a $\mathcal{D}$-module basis

$$e^{(\mathcal{D})} = \mathrm{c} \cdot \begin{bmatrix} \frac{\epsilon\delta-1}{\epsilon(\epsilon-1)} \partial - \frac{z}{\epsilon(\epsilon-1)} \partial^2 \\ \frac{\epsilon(1-3\delta)(4\delta-1)}{\epsilon-1} + \frac{z(7\epsilon\delta-2\epsilon-1)}{\epsilon(\epsilon-1)} \partial - \frac{z^2}{\epsilon(\epsilon-1)} \partial^2 \\ \frac{4\epsilon\delta-\epsilon-1}{\epsilon(\epsilon-1)} \partial - \frac{z}{\epsilon(\epsilon-1)} \partial^3 \end{bmatrix} . \tag{7.20}$$

7.2.2 Macaulay matrix

Our goal is to obtain the Pfaffian system $\partial e^{(\mathcal{D})} = P \cdot e^{(\mathcal{D})}$. Using the Macaulay matrix method, we first construct the Pfaffian system in the basis of standard monomials,

$$\partial \mathrm{Std} = \partial \begin{bmatrix} \partial^2 \\ \partial \\ 1 \end{bmatrix} = \begin{bmatrix} \partial^3 \\ \partial^2 \\ \partial \end{bmatrix} = P^{(\mathrm{Std})} \cdot \begin{bmatrix} \partial^2 \\ \partial \\ 1 \end{bmatrix}, \tag{7.21}$$

whereafter we gauge transform to the $e^{(\mathcal{D})}$ system. With the toric operator (7.17) and the basis Std as input, the Macaulay data works out to

$$\begin{aligned} M_{\mathrm{Ext}} &= \begin{bmatrix} z^2(z+1) \end{bmatrix} \\ M_{\mathrm{Std}} &= \begin{bmatrix} z\big[\epsilon(6\delta+2) + z(6\epsilon\delta+\epsilon+3)+3\big] \\ \big(3\epsilon\delta+\epsilon+1\big)^2 + z\big[\epsilon^2\delta(9\delta+2)+6\epsilon\delta+\epsilon+1\big] \\ \epsilon^3\delta^2(4\delta+1) \end{bmatrix}^T \\ C_{\mathrm{Ext}} &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad C_{\mathrm{Std}} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\ \mathrm{Mons} &= \mathrm{Ext} \sqcup \mathrm{Std} = \{\partial^3\} \sqcup \{\partial^2, \partial, 1\}. \end{aligned} \tag{7.22}$$

Letting $C = \big[c_{11}\ c_{12}\ c_{13}\big]$ be an unknown matrix, we then solve $C_{\mathrm{Ext}} - C \cdot M_{\mathrm{Ext}} = 0$ to obtain

$$C = \begin{bmatrix} \frac{1}{z^2(z+1)} & 0 & 0 \end{bmatrix}. \tag{7.23}$$

The Pfaffian matrix in the basis of standard monomials is then

$$P^{(\mathrm{Std})} = C_{\mathrm{Std}} - C \cdot M_{\mathrm{Std}} \tag{7.24}$$

$$= \begin{bmatrix} P^{(\mathrm{Std})}_{11} & P^{(\mathrm{Std})}_{12} & P^{(\mathrm{Std})}_{13} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \tag{7.25}$$

with

$$
\begin{aligned}
P_{11}^{(\mathrm{Std})} &= -\frac{\epsilon\big[6\delta(z+1)+z+2\big]+3(z+1)}{z(z+1)} \\
P_{12}^{(\mathrm{Std})} &= -\frac{\big(3\epsilon\delta+\epsilon+1\big)^2+z\big[\epsilon^2\delta(9\delta+2)+6\epsilon\delta+\epsilon+1\big]}{z^2(z+1)} \\
P_{13}^{(\mathrm{Std})} &= -\frac{\epsilon^3\delta^2(4\delta+1)}{z^2(z+1)}\,.
\end{aligned}
\tag{7.26}
$$

Finally, using the basis change method described in Appendix (A.1) we build a matrix G such that $e^{(\mathcal{D})} = G \cdot e^{(\mathrm{Std})}$, leading to the gauge transformation

$$
P = \Big(\partial \bullet G + G \cdot P^{(\mathrm{Std})}\Big) \cdot G^{-1} \tag{7.27}
$$

$$
= \epsilon \begin{bmatrix} -\frac{1}{z} & 0 & 0 \\ 0 & 0 & 0 \\ -\frac{2}{z(z+1)} & \frac{2}{z+1} & -\frac{1}{z(z+1)} \end{bmatrix}. \tag{7.28}
$$

The limit $\delta \to 0$ has already been taken here. This matrix is canonical as expected, and is in agreement with an independent LiteRed computation.

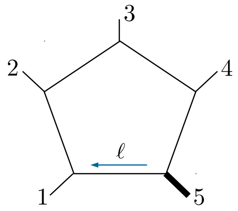
Section 7.3

Macaulay matrix example: 1-loop pentagon with off-shell leg

Here we compute a Pfaffian system for the 1-loop massless pentagon integral with one massive leg. The kinematics are

$$
\sum_{i=1}^{5} p_i = 0\,, \quad p_1^2 = \cdots = p_4^2 = 0\,, \quad p_5^2 = p^2
$$

$$
\text{and} \quad s_{ij} = 2p_i \cdot p_j\,.
$$



Note that the identity $(p_1+p_2+p_3+p_4)^2 = (-p_5)^2 = p^2$ can be used to eliminate one of the Mandelstam variables s_{ij}.

The inverse propagators are given by

$$
\begin{aligned}
D_1 &= -\ell^2\,, \quad D_2 = -(\ell-p_1)^2\,, & D_3 &= -(\ell-p_1-p_2)^2 \\
D_4 &= -(\ell-p_1-p_2-p_3)^2\,, & D_5 &= -(\ell-p_1-p_2-p_3-p_4)^2\,.
\end{aligned}
\tag{7.29}
$$

The associated GFI is

$$I(\mathrm{D}_0|\nu) = c(\mathrm{D}_0|\nu)(-s_{12})^{\mathrm{D}_0/2-\epsilon-|\nu|-5\epsilon\delta} \int_{(0,\infty)^5} \mathcal{G}(z|x)^{\epsilon-\mathrm{D}_0/2}\, x_1^{\nu_1+\epsilon\delta} \cdots x_5^{\nu_5+\epsilon\delta}\, \frac{\mathrm{d}x}{x}\,, \quad (7.30)$$

with LP monomials given by

$$\mathcal{G}(z|x) = \sum_{i=1}^{5} z_i\, x_i + z_7\, x_1\, x_4 + z_8\, x_1\, x_5 + z_9\, x_2\, x_4 + z_{10}\, x_1\, x_5 + z_{11}\, x_3\, x_5\,, \quad (7.31)$$

$$\text{corresponding to} \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

The s_{12}-dependent prefactor in (7.30) results from rescaling integration variables by $x_i \to x_i/(-s_{12})$.

The proper LP polynomial has the monomial coefficients

$$\begin{aligned} z_1 &= \ldots = z_6 = 1 \\ z_7 &= 1 + y_2 + y_4 \\ z_8 &= y_1 \\ z_9 &= y_4 \\ z_{10} &= -1 + y_1 - y_2 - y_3 \\ z_{11} &= -1 + y_1 - y_2 - y_3 - y_4 - y_5 \end{aligned} \quad (7.32)$$

in terms of ratios

$$y_1 = \frac{p^2}{s_{12}}\,, \; y_2 = \frac{s_{13}}{s_{12}}\,, \; y_3 = \frac{s_{14}}{s_{12}}\,, \; y_4 = \frac{s_{23}}{s_{12}}\,, \; y_5 = \frac{s_{24}}{s_{12}}\,. \quad (7.33)$$

There are 5 monomial coefficients that differ from unity. We can exactly match them with GKZ variables using homogeneity. More precisely, the GKZ system originally has $N = 11$ variables, but $n + 1 = 5 + 1 = 6$ variables can be rescaled as $z_1 = \ldots = z_6 = 1$, leaving us with 5 variables $\{z_7, \ldots, z_{11}\}$.

7.3.1 Basis

Computing the volume of the Newton polytope with polymake, we expect $R = 5! \times 13/120 = 13$ MIs. With the asir command `mt_gkz.cbase_by_euler`, we instantly identify a basis of 13 standard monomials

$$e^{(\mathrm{Std})} = \begin{bmatrix} \partial_9\partial_{11}^2 \\ \partial_9^2 \\ \partial_{10}^2 \\ \partial_8\partial_{11} \\ \partial_9\partial_{11} \\ \partial_{10}\partial_{11} \\ \partial_{11}^2 \\ \partial_7 \\ \partial_8 \\ \partial_9 \\ \partial_{10} \\ \partial_{11} \\ 1 \end{bmatrix} . \tag{7.34}$$

For the sake of verifying our results against an independent IBP calculation, we also determine an integral basis using LiteRed:

$$e^{(I)} = (-s_{12})^{\epsilon} \begin{bmatrix} I(2|0,0,1,0,1) \\ I(2|0,1,0,0,1) \\ I(2|0,1,0,1,0) \\ I(2|1,0,0,0,1) \\ I(2|1,0,0,1,0) \\ I(2|1,0,1,0,0) \\ \epsilon I(4|1,0,1,0,1) \\ \epsilon(-s_{12})I(4|0,1,1,1,1) \\ \epsilon(2\epsilon-1)I(6|1,0,1,1,1) \\ \epsilon(-s_{12})I(4|1,1,0,1,1) \\ \epsilon(-s_{12})I(4|1,1,1,0,1) \\ \epsilon(-s_{12})I(4|1,1,1,1,0) \\ \epsilon^2(-s_{12})I(6|1,1,1,1,1) \end{bmatrix} . \tag{7.35}$$

This basis has been slightly dressed up with prefactors to simplify its Pfaffian system. Moreover, the dimension D_0 of each $I(\mathrm{D}_0|\nu)$ is chosen to ensure that its associated r-vector only contains non-negative integers, as per the remark above Example (6.1). The $\mathcal{D}$-module basis corresponding to (7.35) is computed by inserting these r-vectors into the asir command `mt_gkz.rvec_red2`, with the result

$$e^{(\mathcal{D})} = \mathrm{c} \cdot \begin{bmatrix} \partial_{11} \\ \partial_{10} \\ \partial_9 \\ \partial_8 \\ \partial_7 \\ \epsilon(5\delta+1) + z_7\partial_7 + z_8\partial_8 + z_9\partial_9 + z_{10}\partial_{10} + z_{11}\partial_{11} \\ (4\epsilon\delta+\epsilon+1)\partial_{11} + z_{11}\partial_{11}^2 + z_9\partial_9\partial_{11} + z_{10}\partial_{10}\partial_{11} \\ \partial_9\partial_{11} \\ e_9^{(\mathcal{D})} \\ \partial_7\partial_{10} \\ (5\epsilon\delta+\epsilon+1)\partial_{10} + z_{10}\partial_{10}^2 + z_7\partial_7\partial_{10} + z_8\partial_8\partial_{10} + z_9\partial_9\partial_{10} + z_{11}\partial_{11}\partial_{10} \\ (5\epsilon\delta+\epsilon+1)\partial_9 + z_9\partial_9^2 + z_7\partial_7\partial_9 + z_8\partial_8\partial_9 + z_{10}\partial_{10}\partial_9 + z_{11}\partial_{11}\partial_9 \\ (4\epsilon\delta+\epsilon+2)\partial_{11}\partial_9 + z_9\partial_{11}\partial_9^2 + z_{10}\partial_{10}\partial_{11}\partial_9 + z_{11}\partial_{11}^2\partial_9 \end{bmatrix}, \tag{7.36}$$

where

$$\begin{aligned} e_9^{(\mathcal{D})} = {} & \epsilon\delta\,(4\epsilon\delta+1)\,\partial_{11} + z_{11}\epsilon\delta\partial_{11}^2 + z_7\,(4\epsilon\delta+\epsilon+1)\,\partial_7\partial_{11} + z_9\,(5\epsilon\delta+\epsilon+2)\,\partial_9\partial_{11} \\ & + z_{10}\epsilon\delta\partial_{10}\partial_{11} + z_7z_{11}\partial_7\partial_{11}^2 + z_9z_{11}\partial_9\partial_{11}^2 + z_9^2\partial_9^2\partial_{11} \\ & + z_7z_9\partial_7\partial_9\partial_{11} + z_7z_{10}\partial_7\partial_{10}\partial_{11} + z_9z_{10}\partial_9\partial_{10}\partial_{11}\,, \end{aligned} \tag{7.37}$$

and c is a 13 × 13 diagonal matrix containing prefactors (it can be downloaded from a mathematica notebook in [147]).

7.3.2 Macaulay matrix

The Macaulay matrix data is obtained in less than a second on a laptop via the asir command `mt_mm.find_macaulay`. The matrix M_{Ext} has dimensions 189 × 113, meaning that there are 113 exterior monomials. After fixing every parameter in M_{Ext} to a number, row reduction reveals that only 133 out of the 189 rows are independent. The matrix containing these independent rows is denoted by M'_{Ext}. Starting from the 189×13 matrix M_{Std}, we similarly define the 133×13 matrix M'_{Std}.

For each $i \in \{7, \ldots, 11\}$ we proceed to solve the equation

$$C^{(i)}_{\text{Ext}} - C^{(i)} \cdot M'_{\text{Ext}} = 0 \tag{7.38}$$

for the unknown matrices $C^{(i)}$. This only takes a few minutes on a laptop with FiniteFlow. The Pfaffian matrix in direction i is finally obtained from

$$C^{(i)}_{\text{Std}} - C^{(i)} \cdot M'_{\text{Std}} = P^{(\text{Std})}_i \,. \tag{7.39}$$

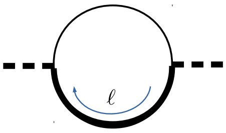
To compare it with LiteRed, we perform a gauge transformation from the basis $e^{(\text{Std})}$ in (7.34) to $e^{(\mathcal{D})}$ in (7.36). After sending $\delta \to 0$, we find perfect agreement.

Section 7.4

Recurrence relation example: 1-loop bubble

Here we showcase Algorithm (2) by computing an IBP relation for a simple bubble diagram. The inverse propagators are

$$D_1 = -\ell^2 + m^2 \,, \quad D_2 = -(\ell + p)^2 \,,$$



where p is an incoming external momentum. The GFI takes the form

$$I(\mathrm{D}_0|\nu_1, \nu_2) = c(\mathrm{D}_0|\nu_1, \nu_2) \int_{(0,\infty)^2} \mathcal{G}(z|x)^{\epsilon - \mathrm{D}_0/2} \, x_1^{\nu_1 + \epsilon\delta} \, x_2^{\nu_2 + \epsilon\delta} \, \frac{\mathrm{d}x}{x} \,, \tag{7.40}$$

with the LP monomials

$$\mathcal{G}(z|x) = z_1 \, x_1 + z_2 \, x_2 + z_3 \, x_1 \, x_2 + z_4 \, x_1^2 \,, \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 0 \end{bmatrix} . \tag{7.41}$$

We avoid the use of homogeneity in order to follow the steps of Algorithm (2) as closely as possible[3]. Instead, we simply specify

$$z_1 = z_2 = 1\,, \quad z_3 = m^2 - p^2, \quad z_4 = m^2 \tag{7.42}$$

at the end of the computation to match with the physical LPr.

Fixing a basis of MIs

$$e = \begin{bmatrix} I(4|1,1) \\ I(4|2,0) \end{bmatrix}, \tag{7.43}$$

we now proceed to find $c_{1,2}$ in the decomposition[4]

$$e_0 = I(4|1,2) = c_1 e_1 + c_2 e_2 \tag{7.44}$$

using Algorithm (2).

Step 0: Input. Set $\beta = \{\epsilon, -\epsilon\delta, -\epsilon\delta\} \in \mathbb{C}^3 \setminus \mathbb{Z}^3$. According to the formula (6.81), Algorithm (2) pertains to Euler integrals of the form

$$f(\beta - A\cdot q) = I(\mathrm{D}_0|\nu)/c(\mathrm{D}_0|\nu)\,. \tag{7.45}$$

By the definition of $I(\mathrm{D}_0|\nu)$ from (7.7), we thus require that the argument of f be

$$\beta - A\cdot q = \beta - \big(\mathrm{D}_0/2, \nu\big)\,. \tag{7.46}$$

(Such a q-vector plays a similar role as the r-vector from (6.43).) For the integrals in the example at hand, we have

	q	$A\cdot q$	$\mathrm{Std}_q = \partial^q$
$I(4\|1,2)$	$q_0 = (0,1,1,0)$	$(2,1,2)$	$\partial_2\partial_3$
$I(4\|1,1)$	$q_1 = (1,1,0,0)$	$(2,1,1)$	$\partial_1\partial_2$
$I(4\|2,0)$	$q_2 = (2,0,0,0)$	$(2,2,0)$	∂_1^2

(7.47)

In the final column, we provided the $\mathcal{D}$-module representatives of the integrals (up to prefactors).

[3] Algorithm (2) assumes that the input Std consists of single monomials. Rewriting the basis in rescaled variables using (A.34) would generally lead to operators with several terms, slightly modifying the algorithm.

[4] Here we pick an example where all the ν_i are non-negative. For negative ν_i, one has to slightly tweak the algorithm with the shift vector α from [82, Equation 6.28].

Step 1. Calling Algorithm (1), we obtain the 2×2 Pfaffian matrices $\{P_1, P_2, P_3, P_4\}$ in the basis $\text{Std} = \{\partial^{q_1}, \partial^{q_2}\}$.

Step 2. We decompose

$$\begin{aligned} q_0 - q_1 &= (-1, 0, 1, 0) \\ &= (0, 0, 1, 0) - (1, 0, 0, 0) \\ &= q^+ - q^- \\ &= q\,. \end{aligned} \tag{7.48}$$

Step 3. This step instructs us to compute the matrix factorial corresponding to the operator $\partial_3^{+1}\,\partial_1^{-1}$, where ∂_3 stems from q^+ and ∂_1 stems from q^-. Explicitly, we compute

$$P_3(\beta + a_1) \cdot Q_1(\beta) \cdot e\,. \tag{7.49}$$

Step 4: Output. The sought after recurrence relation is encoded in the first element of (7.49). After sending $\delta \to 0$ and replacing the z_i variables as in (7.42), we obtain the result

$$I(4|1,2) = \frac{(1-2\epsilon)(p^2+m^2)}{(p^2-m^2)^2} I(4|1,1) - \frac{2m^2}{(p^2-m^2)^2} I(4|2,0)\,, \tag{7.50}$$

which is in agreement with LiteRed.

Chapter 8

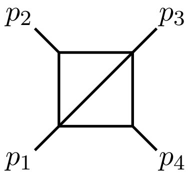
Restriction of $\mathcal{D}$-modules

As emphasized in the previous chapter, the GKZ system does not perfectly capture the nature of FIs in the LPr. The former requires all monomial coefficients z_i of the integrand polynomial $\mathcal{G}(z|x)$ to be indeterminate, but the latter has some of the z_i fixed to special values. The question is then whether there exists a kind of *limit* of the GKZ system, at the level of $\mathcal{D}$-modules, which will exactly identify it with the annihilating ideal of a proper FI.

To motivate the need for a restriction procedure, let us consider an example.

Example 8.1. Let

$$I(\mathrm{D}_0|\nu) \propto \int_{(0,\infty)^5} \mathcal{G}(z|x)^{\epsilon-\mathrm{D}_0/2}\, x_1^{\nu_1+\epsilon\delta} \cdots x_5^{\nu_5+\epsilon\delta}\, \frac{\mathrm{d}x}{x}$$



denote the GFI associated to the 2-loop massless N-box diagram. The kinematics are

$$p_1^2 = \cdots = p_4^2 = 0 \quad s = 2p_1 \cdot p_2 \quad t = 2p_2 \cdot p_3\,, \tag{8.1}$$

and the LP monomials read

$$\begin{aligned}\mathcal{G}(z|x) = {} & z_1x_1x_2 + z_2x_1x_4 + z_3x_1x_5 + z_4x_2x_3 + z_5x_2x_4 + z_6x_2x_5 \\ & + z_7x_3x_4 + z_8x_3x_5 + z_9x_1x_2x_4 + z_{10}x_2x_3x_5\,.\end{aligned} \tag{8.2}$$

The holonomic rank of the GKZ system is $R = 9$, as dictated by the formula (6.52), and it stays fixed when six of the z_i variables are rescaled to unity via homogeneity.

After rescaling integration variables by $x \to x/s$, the proper LPr has monomial coefficients confined to the space

$$Y' = \left\{(z_1, \ldots, z_{10}) \in \mathbb{A}^{10} \,\middle|\, z_1 = \ldots = z_9 = 1\right\}, \tag{8.3}$$

with $z_{10} = t/s$ being generic. Counting the dimension of the twisted cohomology group $\mathbb{H}^k_A(\beta)$ for $z \in Y'$ via the likelihood equation (6.36), one finds that $\dim(\mathbb{H}^k_A(\beta)) = 3$. This is the correct number of MIs, as as verified by an IBP calculation.

As $R = 9 > 3$, we infer that the GKZ system is "too big". We therefore seek some way to recover the space $\mathbb{H}^k_A(\beta)$, having special values for the z's, via some limiting procedure of the $\mathcal{D}/H_A(\beta)$. ■

"Limits" of $\mathcal{D}$-module are known under the name of *restrictions* [85, Chapter 5]. Standard techniques for computing restrictions require huge computational resources, and can generally not be applied to FIs of interest. In this chapter, we shall develop two new methods for computing restrictions that are more efficient. The first one takes the Pfaffian system associated to a basis of Euler integrals as input. The second one works at the level of the Macaulay matrix. These two methods have different strengths and weaknesses, so it depends on the context whether to apply one or the other.

Section 8.1

Preliminaries

We begin with some extra $\mathcal{D}$-module formalism. More details and examples besides what is presented here can be found in [83, Appendix A].

8.1.1 Tensor products

Tensor products of vector spaces are familiar to physicists from the study of quantum mechanics. It is possible to define tensor products of $\mathcal{D}$-modules as well. Let $\mathcal{M}$ and $\mathcal{N}$ be two $\mathcal{D}$-modules. We proceed now to define the tensor product $\mathcal{N} \otimes_{\mathcal{D}} \mathcal{M}$.

Let $n \otimes m$ denote the tuple (n, m) where $n \in \mathcal{N}$ and $m \in \mathcal{M}$, and define the space $S(\mathcal{N} \otimes \mathcal{M}) = \left\{\text{finite, formal sums } \sum_i n_i \otimes m_i\right\}$. We specify an equivalence relation on S:

1. Linearity in the left slot:

$$n_1 \otimes m + n_2 \otimes m \quad \sim \quad (n_1 + n_2) \otimes m\,. \tag{8.4}$$

2. Linearity in the right slot:

$$n \otimes m_1 + n \otimes m_2 \quad \sim \quad n \otimes (m_1 + m_2)\,. \tag{8.5}$$

3. Moving factors "across" the $\otimes$ sign:

$$(n\,d) \otimes m \quad \sim \quad n \otimes (d\,m)\,, \tag{8.6}$$

   for $d \in \mathcal{D}$. This is the reason for the subscript in $\otimes_{\mathcal{D}}$.

The $\mathcal{D}$-module tensor product is then defined by the space $S(\mathcal{N} \otimes \mathcal{M})$ modulo this equivalence relation, that is $\mathcal{N} \otimes_{\mathcal{D}} \mathcal{M} := S(\mathcal{N} \otimes \mathcal{M})/\sim$.

8.1.2 Definition of restriction

The variables $z = (z_1, \ldots, z_N)$ parametrize an affine space $Y = \mathbb{A}^N$. We write $\mathcal{D}_Y, \mathcal{R}_Y$ etc. for spaces of operators defined on Y. Consider an affine hyperplane $Y' \subset Y$ of lower dimension - this will, later on, represent the proper space of kinematic variables. By an affine linear change of coordinates, we can define Y' as the vanishing locus

$$Y' = \{z \in Y \,|\, z_1 = \cdots = z_{N'} = 0\}\,. \tag{8.7}$$

Definition 8.1. Denote by $\iota : Y' \hookrightarrow Y$ the natural inclusion map. Let $\mathcal{M}$ be a $\mathcal{D}_Y$-module. The *restriction module* of $\mathcal{M}$ is defined as the $\mathcal{D}_{Y'}$-module

$$\iota^* \mathcal{M} := \frac{\mathcal{M}}{z_1 \mathcal{M} + \ldots + z_{N'} \mathcal{M}}\,. \tag{8.8}$$

The operation ι^* can be thought of as a pullback from Y to Y'. See [85, Chapter 5] [86, Chapter 6] for details.

A typical use case for the restriction module is the following. Suppose the $\mathcal{D}_Y$-module $\mathcal{M} = \mathcal{D}_Y/\mathcal{I}$ is constructed from the annihilating ideal $\mathcal{I}$ of some function $f(z)$ for $z \in Y$. By computing $\iota^*\mathcal{M} = \mathcal{D}_{Y'}/\mathcal{I}'$ for some $\mathcal{I}'$, then $\mathcal{I}'$ is the annihilating ideal for $f(z)$ where now $z \in Y'$. Naturally, it is not necessary that $\mathcal{M}$ be given by a quotient of operator spaces. As in Example (4.3), the $\mathcal{D}_Y$-module $\mathcal{M}$ can also be equal, as a set, to a suitable space of functions.

Example 8.1. Consider the GKZ system associated to the Euler integral

$$f_\beta(z_1,z_2,z_3) = \int_{(0,\infty)^2} (z_1x_1 + z_2x_2 + z_3x_1x_2)^{\beta_0} x_1^{-\beta_1} x_2^{-\beta_2} \frac{\mathrm{d}x}{x} \,. \tag{8.9}$$

The annihilating ideal $\mathcal{I} = H_A(\beta)$ is solely generated by the homogeneity operators

$$\begin{aligned} E_0 &= z_1\partial_1 + z_2\partial_2 + z_3\partial_3 - \beta_0 \\ E_1 &= z_1\partial_1 + z_3\partial_3 - \beta_1 \\ E_2 &= z_2\partial_2 + z_3\partial_3 - \beta_2 \,. \end{aligned} \tag{8.10}$$

We are interested in the annihilating ideal for the function $f_\beta(1,1,z_3)$. By an affine linear coordinate change $\{z_1 \to z_1 + 1\,, z_2 \to z_2 + 1\,, z_3 \to z_3\}$, we hence have the affine subspace given by $Y' = \{z_1 = z_2 = 0\}$.

A variant of the restriction algorithm presented in Oaku's seminal work [148] has been implemented in the ASIR package NK_RESTRICTION [149]. Inputting the data above, one finds the annihilating ideal

$$\mathcal{I}' = \langle -z_3\partial_3 - \beta_0 + \beta_1 + \beta_3 \rangle \,. \tag{8.11}$$

One can check that this indeed annihilates the function

$$f_\beta(1,1,z_3) = \frac{\Gamma(-\beta_0+\beta_1)\Gamma(-\beta_0+\beta_2)\Gamma(\beta_0-\beta_1-\beta_2)}{\Gamma(-\beta_0)} z_3^{-\beta_0+\beta_1+\beta_2} \,. \tag{8.12}$$

■

The discussion above was directed at $\mathcal{D}$-modules, i.e. when the action is given by operators with polynomial coefficient functions. We are, more broadly, interested in $\mathcal{R}$-modules having rational function coefficients, because this is what appears in Pfaffian systems. An $\mathcal{R}$-module naturally arises from a given $\mathcal{D}$-module $\mathcal{M}$ by taking a tensor product: $\mathcal{R} \otimes_{\mathcal{D}_Y} \mathcal{M}$.

Definition 8.2. The *rational restriction* of a $\mathcal{D}_Y$-module $\mathcal{M}$ onto Y' is defined by

$$\mathcal{R}_{Y'} \otimes_{\mathcal{D}_{Y'}} \iota^*\mathcal{M} \,, \tag{8.13}$$

where $\iota^*\mathcal{M}$ is the restriction module (8.8).

The restriction algorithms presented in this chapter will output Pfaffian systems that live in the space (8.13).

8.1.3 Normal form

Let $\partial_i \bullet e(z) = P_i \cdot e(z)$ be a Pfaffian system that is *regular singular* (cf. Section (4.2.1)). Loosely speaking, this means that the solutions vector $e(z)$ is free of essential singularities such $\exp(1/z)$ at $z = 0$. It is expected that FIs fall into this category, as the poles in ϵ-factorized Pfaffian systems always appear to be simple.

We focus on a single pole of the system, which can be written as $Y' = \{z_1 = 0\}$ by a change of coordinates. Theorems by Deligne [150, 151] ensure that the Pfaffian system can be brought into *normal form* w.r.t. Y'. Namely, there exists another set of Pfaffian matrices $\widetilde{P}_i(z)$ such that

1. $\widetilde{P}_1$ has a simple pole on Y' and $\{\widetilde{P}_2, \ldots, \widetilde{P}_N\}$ are finite on Y':

$$\widetilde{P}_1(z) = \sum_{n=-1}^{\infty} \widetilde{P}_{1,n}(z')z_1^n\,, \quad \text{and} \tag{8.14}$$

$$\widetilde{P}_i(z) = \sum_{n=0}^{\infty} \widetilde{P}_{i,n}(z')z_1^n\,, \qquad \text{for} \quad i = 2, \ldots, N\,, \tag{8.15}$$

   where the truncated list of variables is

$$z' = (z_2, \ldots, z_N)\,. \tag{8.16}$$

   Note that the expansion of $\widetilde{P}_1$ starts at $n = -1$, and the expansions for $\widetilde{P}_{i=2,\ldots,N}$ start at $n = 0$.

2. The spectrum of the *residue matrix* $\widetilde{P}_{1,-1}(z')$ is *non-resonant*, meaning that 1) no two distinct eigenvalues of $\widetilde{P}_{1,-1}(z')$ have an integral difference, and 2) the only integral eigenvalue, if it exists, is 0.

The matrices $\widetilde{P}_i(z)$ are related to the original ones $P_i(z)$ via a gauge transformation. By the work of Moser [152], and later Barkatou et al. [153, 154], there exist explicit *Moser reduction* algorithms for constructing such a gauge transformation (see [83, Appendix B] for a review). Moser reduction has been implemented in software such as Fuchsia [155] and Isolde [156].

It can be computationally expensive to enforce a non-resonant spectrum on a Pfaffian system, as this step of the algorithm requires the calculation of Jordan decompositions[1]. However, if we only seek to perform Moser reduction w.r.t. a single pole $z_1 = 0$ rather

[1] In particular, one is required to know the sizes of the Jordan cells [83, Appendix B.2].

than many, then the algorithm can readily be applied to rather large systems. Moreover, since the Jordan decomposed matrix does not depend on the z-variables [157, Theorem 12.1], it is possible speed up the computation by fixing the variables to generic numbers.

Section 8.2

Restriction at the level of Pfaffian systems

In this section, we present a restriction algorithm which takes a Pfaffian system on $Y = \mathbb{A}^N$ as input and outputs a new Pfaffian system that holds true on an affine subspace $Y' \subset Y$. As has been mentioned, the original motivation was to calculate restrictions of GKZ systems. However, the algorithm turns out to work for any regular holonomic $\mathcal{D}$-module. So it can, rather serendipitously, also be applied to Pfaffian systems for FIs obtained by conventional means, such as IBPs in momentum space.

As it will turn out, the algorithm can also be used to compute logarithmic series solutions to Pfaffian systems. This is helpful for calculating asymptotic expansions for FIs, such as threshold and small-mass expansions.

8.2.1 Restriction of a logarithmic connection

Suppose we are given a Pfaffian system on the "big" space Y. We want to restrict it onto a "smaller" space $Y' = \{z_1 = 0\}$. Further, $z_1 = 0$ should constitute a pole in the Pfaffian system. There might poles elsewhere on Y, but we imagine working "locally" around this single pole[2]. Assume that the Pfaffian system is in normal form w.r.t. Y' (cf. Section (8.1.3)). Then it is possible to write it in a peculiar way:

$$z_1\partial_1 \bullet e(z) = P_1 \cdot e(z) \tag{8.17}$$

$$\partial_i \bullet e(z) = P_i \cdot e(z)\,, \quad i = 2, \ldots, N\,, \tag{8.18}$$

where $e(z)$ is an R-dimensional vector of functions. We would like to use this Pfaffian system to construct a certain $\mathcal{D}$-module $\mathcal{M}$ consisting of *regular* functions. The action of the derivative ∂_i should be well-defined on such a space, in the sense of not creating unwanted singularities. We have therefore guarded ourselves in two ways to ensure well-definedness: 1) by working locally around Y' we can thereby ignore singularities elsewhere, and 2) by moving $1/z_1$ from the RHS of (8.17) to get $z_1\partial_1$ on the LHS. We

[2] The notion of "local" is here meant w.r.t. the Zariski topology. So to be rigorous, we ought to formulate everything in terms of Zariski open subsets Y_0 of Y as in [83, Section 3]. Many such mathematical details are neglected here in order to take the shortest path towards defining the practical algorithms.

are thence motivated to define a subring of $\mathcal{D}_Y$ by

$$\mathcal{D}_Y(\log Y') = \mathbb{C}\langle z_1, \ldots, z_N, z_1\partial_1, \partial_2, \ldots, \partial_N \rangle \,, \tag{8.19}$$

and the abelian group

$$\mathcal{M} = \big\{ R\text{-dimensional vectors of functions on } Y \text{ that are regular near } Y' \big\} \,. \tag{8.20}$$

The action of the ring $\mathcal{D}_Y(\log Y')$ onto $\mathcal{M}$ is well-defined. The generators z_i act by scalar multiplication, and the generators $\{z_1\partial_1, \partial_2, \ldots, \partial_N\}$ act according to the RHSs of the Pfaffian system (8.17)-(8.17). This turns $\mathcal{M}$ into a $\mathcal{D}_Y(\log Y')$-module. In technical jargon, such an $\mathcal{M}$ is called a *logarithmic connection* [110, Section 11.4] [23, Section 5.2.2]. We now proceed to study the rational restriction $\mathcal{R}_{Y'} \otimes_{\mathcal{D}_{Y'}} \iota^*\mathcal{M}$ of the logarithmic connection $\mathcal{M}$.

To start, let $P_{1,-1}(z')$ be the residue matrix (cf. (8.14)) associated to $P_1(z)$ defined in (8.17), where $z' = (z_2, \ldots, z_N)$. The entries of $P_{1,-1}(z')$ live in the field $\mathcal{K}_{Y'}$ consisting of rational functions on Y'. The R rows of $P_{1,-1}(z')$ generate a subspace of $\mathcal{K}_{Y'}^R$ which we call $\mathcal{N}'$. The following quotient space is important for our construction of the rational restriction:

$$\mathcal{M}' = \mathcal{K}_{Y'}^R \,/\, \mathcal{N}' \,. \tag{8.21}$$

As a set, this consists of R-dimensional vectors of rational functions on Y', and we identify two vectors if they differ by another vector lying in the row space of $P_{1,-1}(z')$. $\mathcal{M}'$ can actually be turned into an $\mathcal{R}_{Y'}$-module. The action of the derivatives $\{\partial_2, \ldots, \partial_N\}$ on an R-dimensional equivalence class $[m] \in \mathcal{M}'$ is

$$\partial_i \bullet [m] = P_{i,0}(z') \cdot [m] \quad \text{for} \quad i = 2, \ldots, N \,, \tag{8.22}$$

where the matrices $P_{i,0}(z')$ come from the leading terms in the z_1-expansion (8.15).

With all of this build-up, we can finally state the following crucial isomorphism of $\mathcal{R}_{Y'}$-modules[3]:

$$\mathcal{R}_{Y'} \otimes_{\mathcal{D}_{Y'}} \iota^*\mathcal{M} \;\simeq\; \mathcal{M}' \,. \tag{8.23}$$

[3] The formula (8.23) shows the rational restriction of $\mathcal{M}$ when it equals a space of functions as a set. Recalling the discussion from Section (4.4), this isomorphism can also be employed when $\mathcal{M}$ is of the form $\mathcal{D}_Y/\mathcal{I}$ for some annihilating ideal of differential operators $\mathcal{I}$.

We refer to [83, Proposition 3.4] for many more mathematical details as well as a proof.

The takeaway message up until now is this. We are given some Pfaffian matrices $\{P_1(z), \dots, P_N(z)\}$ on Y which define $\mathcal{M}$. We would like to know the Pfaffian matrices $\{Q_2(z'), \dots, Q_N(z')\}$ on the smaller space Y'. The matrices $Q_i(z')$ are associated to the rational restriction module of $\mathcal{M}$. Because of the isomorphism (8.23), the matrices $Q_i(z')$ are equally well associated to the quotient space $\mathcal{M}'$. Fortunately, we have a lot of information about $\mathcal{M}'$: it is completely specified by the residue matrix $P_{1,-1}(z')$! In the following subsection, we use this information to provide an explicit formula for the Pfaffian matrices $Q_i(z')$.

8.2.2 Constructing the restricted Pfaffian system

Let R' denote the holonomic rank of $\mathcal{M}'$. In general $R' < R$, i.e. the holonomic rank drops when we fix z-variables to special values. Given an element $m \in \mathcal{K}^R_{Y'}$, we write $[m]$ for its equivalence class in $\mathcal{M}'$.

We define three sets of Pfaffian systems:

1. We suppose the existence of a *special* basis $f = f(z') = \{f_1, \dots, f_R\}$ obeying the Pfaffian system
$$\partial_i \bullet f = \widetilde{Q}_i(z') \cdot f \quad \text{for} \quad i = 2, \dots, N \,. \tag{8.24}$$
We do not yet know f nor $\widetilde{Q}_i(z')$.

2. The basis f is special for the following reason: we assume that $\{f_{R'+1}, \dots, f_R\}$ is a basis of $\mathcal{N}'$. The set $[f] = \{[f_1], \dots, [f_{R'}]\}$ is then a basis for $\mathcal{M}'$, with an associated Pfaffian system
$$\partial_i \bullet [f] = Q_i(z') \cdot [f] \quad \text{for} \quad i = 2, \dots, N \,. \tag{8.25}$$
By stipulation, the first $R' \times R'$ block of $\widetilde{Q}_i(z')$ equals $Q_i(z')$. We do not yet know $[f]$ nor $Q_i(z')$.

3. Let $e = e(z') = \{e_1, \dots, e_R\}$ be the basis associated to the Pfaffian system
$$\partial_i \bullet e = P_{i,0}(z') \cdot e \quad \text{for} \quad i = 2, \dots, N \,. \tag{8.26}$$
The matrices $P_{i,0}(z')$ *are* assumed to be known via the formula (8.15). In contrast to $\widetilde{Q}_i(z')$, the matrices $P_{i,0}(z')$ have no nice block structure.

We are hence in the following situation: we know $P_{i,0}(z')$ (for the basis e), but we do not know $\widetilde{Q}_i(z')$ (for the special basis f). We would like to know $\widetilde{Q}_i(z')$, because then we would automatically get the Pfaffian matrices $Q_i(z')$ for the rational restriction module $\mathcal{M}'$. We shall achieve this goal by building a gauge transformation mapping e to the special basis f. This is done by through three new matrices $\{\mathsf{B}, \mathsf{R}, \mathsf{M}\}$, which we define now.

- Recall the basis $[f] = \{[f_1], \ldots, [f_{R'}]\}$ of $\mathcal{M}'$. Each representative of these equivalence classes can be expanded in the basis e via some $R' \times R$ matrix B having entries in $\mathcal{K}_{Y'}$:

$$f_i = \sum_{j=1}^{R} \mathsf{B}_{ij}\, e_j \quad \text{for} \quad i = 1, \ldots, R' \,. \tag{8.27}$$

  The entries of B are not known at this point.

- By the definition of $\mathcal{M}'$, the residue matrix $P_{1,-1}(z')$ has exactly $R - R'$ independent rows. Define an $(R - R') \times R$ matrix by

$$\mathsf{R} = \texttt{RowReduce}\big[P_{1,-1}(z')\big] \,, \tag{8.28}$$

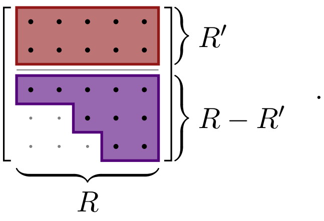
  where `RowReduce` includes the deletion of zero-rows. Then

$$f_i = \sum_{j=1}^{R} \mathsf{R}_{ij}\, e_j \quad \text{for} \quad i = R' + 1, \ldots, R \tag{8.29}$$

  forms a basis of $\mathcal{N}'$.

- Aligning the row of B and R produces an invertible $R \times R$ matrix M:

$$\mathsf{M} = \left[\frac{\mathsf{B}}{\mathsf{R}}\right] = \tag{8.30}$$



  Note that R is known by stipulation, and the matrix B can be anything as long as M becomes invertible. We shall give a simple recipe for generating B later on.

In vector notation, we now have that $f = \mathsf{M} \cdot e$. This means that the unknown matrices $\widetilde{Q}_i(z')$ (associated to f) are obtained from the known matrices $P_{i,0}(z')$ (associated to e) by gauge transforming with M^{-1}. Because the sought after matrix $Q_i(z')$ is given by the first $R' \times R'$ block of $\widetilde{Q}_i(z')$, we can finally write the key formula

$$\boxed{\widetilde{Q}_i(z') \quad = \quad \big[\partial_i \bullet \mathsf{M} \quad + \quad \mathsf{M} \cdot P_{i,0}(z')\big] \cdot \mathsf{M}^{-1} \quad = \quad \left[\begin{array}{c|c} Q_i(z') & \star \\ \hline \mathbf{0} & \star \end{array}\right].} \tag{8.31}$$

See [83, Appendix C.2.1] for an explanation of the zero block in the lower left corner. Obtaining this block is a useful check for practical computations.

Before closing this section, let us observe that the integer N does not have to be the "N" of the GKZ system. In fact, we did not use any notions related to the GKZ system to derive (8.31), so the preceding discussion holds true for any (regular singular) Pfaffian system!

8.2.3 Holomorphic restriction of solutions

The previous section focused on the matrices associated to Pfaffian systems. In the following two subsections, we draw our attention to the *solution vectors* of Pfaffian systems, in particular how to obtain actual series representations using restriction. This topic can be viewed as singular boundary value problem (see [157] [158] and references therein).

To emphasize the connection to MIs, let us use the notation $\vec{I}(z)$ for the solution vector of a Pfaffian system

$$\partial_i \bullet \vec{I}(z) = P_i(z) \cdot \vec{I}(z)\,, \quad i = 1, \ldots, N\,. \tag{8.32}$$

Suppose that we are interested in solutions that are *holomorphic* on $Y' = \{z_1 = 0\}$. This means that $\vec{I}(z)$ enjoys a Taylor expansion

$$\vec{I}(z) = \sum_{n=0}^{\infty} \vec{I}^{\,(n)}(z')\, z_1^n\,, \tag{8.33}$$

where as usual $z' = (z_2, \ldots, z_N)$. The issue at hand is how compute the expansion coefficient functions $\vec{I}^{\,(n)}$ up to some appropriate order $\mathcal{O}(z_1^k)$.

Assume that the matrices $P_i(z)$ are in normal form. Insert the z_1-expansions of $P_i(z)$ and $\vec{I}(z)$, given by (8.14)-(8.15) and (8.33), into the Pfaffian system (8.32). Relations among the $\vec{I}^{\,(n)}$ are then obtained by comparing powers of z_1^n on both sides of the equality

[157, Chapter 4] [83, Appendix D.6]. Comparing the leading-order terms in z_1, the following relations are found:

$$\begin{aligned} \partial_i \bullet \vec{I}^{\,(0)}(z') &= P_{i,0}(z') \cdot \vec{I}^{\,(0)}(z') \,, \quad i = 2, \ldots, N \\ \text{Rank jump constraint:} \quad P_{1,-1}(z') \cdot \vec{I}^{\,(0)}(z') &= 0 \,. \end{aligned} \tag{8.34}$$

The higher-order terms in z_1 yield the recursion relation[4]

$$\vec{I}^{\,(n)}(z') = \left[n\mathbf{1} - P_{1,-1}(z')\right]^{-1} \cdot \sum_{i=0}^{n-1} P_{1,n-i-1} \cdot \vec{I}^{\,(i)}(z') \,, \quad n > 0 \,. \tag{8.35}$$

The set of PDEs (8.34) is dubbed the *holomorphic restriction* of the Pfaffian system (8.32). The first equation of (8.34) shows that $\vec{I}^{\,(0)}$ plays the role of the basis e in (8.26). The second equation of (8.34) shows why the residue matrix $P_{1,-1}(z')$ played such a central role in the construction of Section (8.2.1), wherein we modded out by the rows of $P_{1,-1}(z')$ to obtain the rational restriction module.

At the level of solutions for Pfaffian systems, we now have a clear interpretation for the *rank jump* from R to R': the equation $P_{1,-1}(z') \cdot \vec{I}^{\,(0)}(z') = 0$ imposes constraints among the entries of $\vec{I}^{\,(0)}(z')$ in the limit $z_1 \to 0$, meaning that some of them become linearly dependent on each other. For FIs, these relations turn out to be the new IBP relations among MIs which arise when the limit $z_1 \to 0$ is taken at the *integrand* level. An example of this phenomenon is shown in Section (9.2).

Observe that once $\vec{I}^{\,(0)}$ is determined from the PDEs (8.34), then (8.35) fixes all higher-order terms $\vec{I}^{\,(n)}$ via simple matrix multiplication (one can even use finite field reconstruction in this step for efficiency). The PDEs (8.34) are, however, written in a redundant way due to the residue matrix constraint. We thus seek to "mod out" by this constraint, in order to write the system in a minimal basis. To set this up, begin by writing $\vec{I}^{\,(0)}$ in terms of some dummy symbols, say

$$\vec{I}^{\,(0)} = \begin{bmatrix} I_1 \\ \vdots \\ I_R \end{bmatrix} \,. \tag{8.36}$$

Recall the definition of the R-matrix from (8.28). Because of the second equation in

[4] The matrix in front of the sum is indeed invertible because the spectrum of $P_{1,-1}(z')$ is non-resonant. More precisely, the only potential integral eigenvalue of $P_{1,-1}(z')$ is 0, and the diagonal is shifted by some $n > 0$, wherefore the determinant of the full expression is non-zero.

(8.34), we have that

$$\mathsf{R} \cdot \vec{I}^{\,(0)} = 0. \tag{8.37}$$

This equation can be solved to determine which of the I_i symbols are linearly dependent on each other. Removing those linearly dependent entries from $\vec{I}^{\,(0)}$, the remaining ones are collected into a new R'-dimensional basis

$$\vec{J} \quad = \quad \begin{bmatrix} I_{i_1} \\ \vdots \\ I_{i_{R'}} \end{bmatrix} \quad = \quad \mathsf{B} \cdot \vec{I}^{\,(0)} \,. \tag{8.38}$$

The newly introduced B-matrix has dimension $R' \times R$ and entries $\mathsf{B}_{ij} \in \{0, 1\}$. In fact, this is precisely the B-matrix from (8.27). Therefore, both blocks of the M-matrix (8.30) are now known. By construction, it holds that

$$\mathsf{M} \cdot \vec{I}^{\,(0)} = \left[\begin{array}{c} \mathsf{B} \\ \hline \mathsf{R} \end{array}\right] \cdot \vec{I}^{\,(0)} = \left[\begin{array}{c} \vec{J} \\ \hline \mathbf{0} \end{array}\right] . \tag{8.39}$$

The Pfaffian matrices $Q_i(z')$ for the basis $\vec{J}$ now immediately follow from (8.31), giving the desired, restricted Pfaffian system

$$\partial_i \bullet \vec{J}(z') = Q_i(z') \cdot \vec{J}(z') \quad \text{for} \quad i = 2, \ldots, N \,. \tag{8.40}$$

This system is *simpler* to solve than the original Pfaffian system (8.32) for $\vec{I}(z)$ because 1) it has a smaller rank $R' < R$, and 2) it contains one less variable. Once $\vec{J}$ is solved for, then $\vec{I}^{\,(0)}$ is determined by multiplying equation (8.39) with M^{-1}.

Algorithm (3) summarizes the preceding discussion. Note that an initial Moser reduction step is required if the input is not in normal form.

Algorithm 3 : Holomorphic series expansion

```
Input:
  • Pfaffian system in normal form of rank $R$: $\partial_i \bullet \vec{I}(z) = P_i(z) \cdot \vec{I}(z)$ for $i = 1, \ldots, N$.
  • Integer $k > 0$.
Output: The first $k$ coefficient functions of $\vec{I}(z) = \sum_{n=0}^{\infty} \vec{I}^{(n)}(z')\, z_1^n$.
Set $\mathsf{R} = \texttt{RowReduce}[P_{1,-1}]$ and $R' = R - \mathrm{rank}(\mathsf{R})$.
Write $\vec{I}^{(0)} = \left[I_1, \ldots, I_R\right]^T$. Eliminate linearly dependent dummy symbols $I_i$ by solving $\mathsf{R} \cdot \vec{I}^{(0)} = 0$. Collect linearly independent ones into $\vec{J}$. Fix $\mathsf{B}$ via $\vec{J} = \mathsf{B} \cdot \vec{I}^{(0)}$.
Set $\mathsf{M} = \left[\frac{\mathsf{B}}{\mathsf{R}}\right]$.
For each $i = 2, \ldots, N$, compute $\left(\partial_i \bullet \mathsf{M} + \mathsf{M} \cdot P_{i,0}\right) \cdot \mathsf{M}^{-1}$ and save the upper left $R' \times R'$ block matrices dubbed $Q_i$.
Solve $\partial_i \bullet \vec{J} = Q_i \cdot \vec{J}$ for $\vec{J}$ and set $\vec{I}^{(0)} = \mathsf{M}^{-1} \cdot \left[\frac{\vec{J}}{\mathbf{0}}\right]$.
For each $n = 1, \ldots, k$, calculate $\vec{I}^{(n)}$ using the recursion (8.35).
return $\{\vec{I}^{(0)}, \ldots, \vec{I}^{(k)}\}$.
```

A geometric interpretation of the holomorphic restriction is illustrated in Figure (8.1). There are three axes, one for the variable z_2 and two for the vector components of $\vec{I}^{(0)}(z') = [I_1(0, z_2),\, I_2(0, z_2)]^T$, where $I_1(z_1, z_2)$ and $I_2(z_1, z_2)$ represent a basis before restriction.

The red surface depicts all possible solutions to the rank jump constraint, i.e. the nullspace of the residue matrix $P_{1,-1}(z_2)$. For fixed z_2, this nullspace spans a 1-dimensional vector space shown as the red lines. As z_2 varies, the lines trace out the whole red surface[5]. The solutions living on the surface constitute a basis, determined by the row space of the B-matrix (8.27), for the restricted Pfaffian system. Since the nullspace is 1-dimensional, the size of this basis is $R' = R - 1 = 1$.

For fixed z_2, each purple line represents the 1-dimensional span of the row vectors in $P_{1,-1}(z_2)$. Varying z_2, then these lines trace out the purple surface. This surface is thus a depiction of the R-matrix (8.28).

[5] Such a family of vector spaces is naturally formalized in terms of vector bundles.

Since the red and purple surfaces are orthogonal to each other, together they span a 2-dimensional vector space. This implies that the M-matrix (8.30) is full rank and therefore invertible.

The yellow curve illustrates a *particular* solution vector $\vec{I}^{(0)}(z_2)$, given some initial condition $\vec{I}^{(0)}(0)$. Observe that this solution is constrained to lie on the red surface - this is analogous to the situation in classical mechanics where one finds the solution to some equations of motion given a holonomic constraint. According to (8.34), the "flow" of the solution vector is governed by the Pfaffian matrix $P_{2,0}(z_2)$: given some point $\vec{I}^{(0)}(\overset{\circ}{z}_2)$ on the curve, then the solution is evolved a small step in z_2 by

$$\begin{aligned} \vec{I}^{(0)}(\overset{\circ}{z}_2 + \Delta z_2) \;&\simeq\; \vec{I}^{(0)}(\overset{\circ}{z}_2) \;+\; \Delta z_2 \Big[\partial_2 \bullet \vec{I}^{(0)}(z_2) \Big] \Big|_{z_2 = \overset{\circ}{z}_2} \\ &=\; \vec{I}^{(0)}(\overset{\circ}{z}_2) \;+\; \Delta z_2 \Big[P_{2,0}(\overset{\circ}{z}_2) \cdot \vec{I}^{(0)}(\overset{\circ}{z}_2) \Big] \,. \end{aligned} \tag{8.41}$$

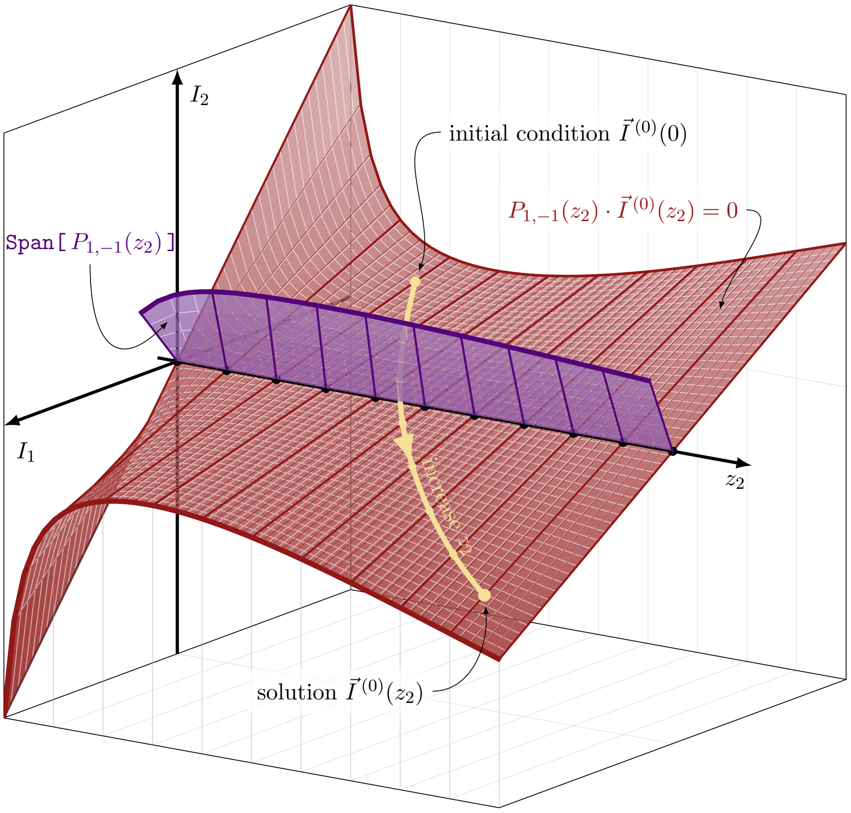

I_2
initial condition $\vec{I}^{(0)}(0)$
$P_{1,-1}(z_2) \cdot \vec{I}^{(0)}(z_2) = 0$
Span[$P_{1,-1}(z_2)$]
I_1
z_2
increase z_2
solution $\vec{I}^{(0)}(z_2)$

Figure 8.1: The holomorphic restriction of a rank $R = 2$ solution vector $\vec{I}(z_1, z_2) = \begin{bmatrix} I_1(z_1,z_2) \\ I_2(z_1,z_2) \end{bmatrix}$ as $z_1 \to 0$. The purple and red surfaces depict R and B respectively.

8.2.4 Logarithmic restriction of solutions

Now suppose that the solution vector $\vec{I}(z)$ to the Pfaffian system (8.32) is logarithmically singular rather than holomorphic on $Y' = \{z_1 = 0\}$. Our goal is to state an algorithm for calculating the logarithmic series expansion of $\vec{I}(z)$.

The first step is to write down the most general ansatz to a Pfaffian system with regular singularities. When the Pfaffian system is in normal form, it is known that the ansatz decomposes into a sum over unique eigenvalues of the residue matrix [157, Chapter 4]. Write the spectrum of the residue matrix as

$$\mathrm{spec}\left(P_{1,-1}(z')\right) = \Big\{ \underbrace{\lambda_1, \ldots, \lambda_1}_{\Lambda_1}\,,\ \underbrace{\lambda_2, \ldots, \lambda_2}_{\Lambda_2}\,, \ldots \Big\}\,, \tag{8.42}$$

where Λ_i is the multiplicity of eigenvalue λ_i. Even though $P_{1,-1}(z')$ depends on z', the integrability relations (4.26) actually dictate that the eigenvalues λ_i are independent of z' [157, Theorem 12.1]. For FIs in DR, it means that λ_i can only depend on ϵ. In all examples known to us, the dependence is even linear in ϵ:

$$\lambda_i = a_i + b_i\epsilon \quad \text{for} \quad a_i, b_i \in \mathbb{Q}\,. \tag{8.43}$$

We are not aware of a proof of the above statement.

The ansatz for a logarithmically divergent solution vector can now be represented as

$$\vec{I}(z) \quad = \quad \sum_{\lambda\ \in!\ \mathrm{spec}(P_{1,-1})} z_1^{\lambda} \times \sum_{n=0}^{\infty} z_1^{n} \times \sum_{m=0}^{M_\lambda} \vec{I}^{(\lambda,n,m)}(z') \times \log^m(z_1)\,, \tag{8.44}$$

where "∈!" means "unique element of". In DR, $\vec{I}^{(\lambda,n,m)}$ can be further expanded in ϵ. Note that there are two sources of logarithms in that case: from the ϵ-expansion of $z_1^{\lambda_i} = z_1^{a_i+b_i\epsilon}$ and from the finite sum over m.

The integer $M_\lambda \in \mathbb{Z}_{\geq 0}$ denotes the maximal power of $\log^m(z_1)$. Its value depends on the Jordan form of the residue matrix. Decomposing this matrix into blocks labeled by unique eigenvalues

$$\texttt{JordanDecomposition}[P_{1,-1}] = \bigoplus_{\lambda\ \in!\ \mathrm{spec}(P_{1,-1})} \mathsf{J}^{(\lambda)}\,, \tag{8.45}$$

then[6]

$$M_\lambda = \left[\text{size of the largest Jordan block of } \mathsf{J}^{(\lambda)}\right] - 1\,. \tag{8.46}$$

The ansatz (8.44) can now be inserted into the Pfaffian system (8.32) together with the z_1-expansions of $P_i(z)$ from (8.14)-(8.15). One thereby obtains relations among the coefficient functions $\vec{I}^{(\lambda,n,m)}$ by comparing the coefficients of $z_1^n \log^m(z_1)$ on both sides of the equality [157, Chapter 4]. For fixed λ, among these relations one finds

$$\begin{aligned} &\partial_i \bullet \vec{I}^{(\lambda,0,0)}(z') \;=\; P_{i,0}(z') \;\cdot\; \vec{I}^{(\lambda,0,0)}(z') \qquad \text{for} \qquad i = 2,\dots,N \\ &\text{Rank jump constraint :} \quad \left[P_{1,-1}(z') - \lambda\mathbf{1}\right]^{M_\lambda+1} \cdot\; \vec{I}^{(\lambda,0,0)}(z') \;=\; 0\,. \end{aligned} \tag{8.47}$$

We call this the *logarithmic restriction* of the solution vector. If $\lambda = 0$, which is the unique integral eigenvalue for a system in normal form, then (8.47) reduces to the holomorphic restriction (8.34).

The second equation of (8.47) represents a constraint on the vector $\vec{I}^{(\lambda,0,0)}$; it must be a generalized eigenvector for $P_{1,-1}(z')$. Recall that we modded out by the rows of $P_{1,-1}(z')$ in the holomorphic case because of the constraint $P_{1,-1}(z') \cdot \vec{I}^{(0)} = 0$ (cf. (8.21)). For fixed λ, in the logarithmic case we instead mod out the rows of $\left[P_{1,-1}(z') - \lambda\mathbf{1}\right]^{M_\lambda+1}$ (this would therefore supplant the $\mathcal{N}'$ in (8.21)).

The vector $\vec{I}^{(\lambda,0,0)}$ hence plays the role of the basis e in (8.26), and we can repeat the construction of Section (8.2.1) by replacing R with

$$\mathsf{R}^{(\lambda)} = \texttt{RowReduce}\left[\left(P_{1,-1}(z') - \lambda\mathbf{1}\right)^{M_\lambda+1}\right]\,. \tag{8.48}$$

This matrix can be inserted into the formula (8.31) together with some basis matrix $\mathsf{B}^{(\lambda)}$. The result is a restricted Pfaffian system of size $R'^{(\lambda)}$ for a basis $\vec{J}^{(\lambda)}(z')$:

$$\partial_i \bullet \vec{J}^{(\lambda)}(z') \;=\; Q_i^{(\lambda)}(z') \;\cdot\; \vec{J}^{(\lambda)}(z') \quad \text{for} \quad i = 2,\dots,N\,. \tag{8.49}$$

The rank of this system is precisely the eigenvalue multiplicity of λ, which also equals the size of the block matrix $\mathsf{J}^{(\lambda)}$ in (8.45). That is,

$$R'^{(\lambda)} = \Lambda\,. \tag{8.50}$$

The higher-order terms $\vec{I}^{(\lambda,n,m)}$ follow from recursion relations given $\vec{I}^{(\lambda,0,0)}$ as input.

[6] Recall that a Jordan block has the eigenvalue λ on the diagonal, and 1's on the superdiagonal. 0's on the superdiagonal correspond to Jordan blocks of size 1.

- Shifting powers of $\log^m(z_1)$. Set

$$\Phi^{(\lambda,m)} = \frac{P_{1,-1} - \lambda\mathbf{1}}{m}\,. \tag{8.51}$$

Then

$$\vec{I}^{(\lambda,0,m)} \quad = \quad \Phi^{(\lambda,m)} \ \cdot \ \vec{I}^{(\lambda,0,m-1)} \quad \text{for} \quad 0 < m \leq M_\lambda\,. \tag{A}$$

The terminating condition is $\vec{I}^{(\lambda,0,M_\lambda+1)} = 0$.

- Shifting powers of z_1^n. Define

$$\Psi^{(\lambda,n)} = \big[(\lambda+n)\mathbf{1} - P_{1,-1}\big]^{-1}\,. \tag{8.52}$$

It holds that

$$\vec{I}^{(\lambda,n,m)} \quad = \quad \Psi^{(\lambda,n)} \ \cdot \ \left(\sum_{\nu=0}^{n-1} P_{1,n-\nu-1} \ \cdot \ \vec{I}^{(\lambda,\nu,m)} \quad - \quad (m+1)\vec{I}^{(\lambda,n,m+1)}\right) \ . \tag{B}$$

The last term drops out when $m = M_\lambda$, leading to[7]

$$\vec{I}^{(\lambda,n,M_\lambda)} \quad = \quad \Psi^{(\lambda,n)} \ \cdot \ \sum_{\nu=0}^{n-1} P_{1,n-\nu-1} \ \cdot \ \vec{I}^{(\lambda,\nu,M_\lambda)}\,. \tag{C}$$

The index n runs up to ∞, so this recursion ought to be truncated based on a numerical accuracy condition for the full solution vector $\vec{I}(z)$.

Starting from $\vec{I}^{(\lambda,0,0)}$, we present a flowchart showing how to use the recursion relations (A), (B) and (C) to compute $\vec{I}^{(\lambda,n,m)}$ for all $0 \leq n \leq N_\lambda$ and $0 \leq m \leq M_\lambda$. The flowchart

[7] (B) in fact also reduces to (A) when $n = 0$.

is adapted from [157, Page 42], and the tuple (n, m) represents $\vec{I}^{(\lambda,n,m)}$.

$$\begin{tikzcd}
(0,0) \arrow[r, "\text{(A)}"] & (0,1) \arrow[r, "\text{(A)}"] & \cdots \arrow[r, "\text{(A)}"] & (0,M_\lambda-1) \arrow[r, "\text{(A)}"] & (0,M_\lambda) \arrow[d, "\text{(C)}"] \\
(1,0) & (1,1) \arrow[l, "\text{(B)}"'] & \cdots \arrow[l, "\text{(B)}"'] & (1,M_\lambda-1) \arrow[l, "\text{(B)}"'] & (1,M_\lambda) \arrow[l, "\text{(B)}"'] \arrow[d, "\text{(C)}"] \\
 & & & & \vdots \arrow[d, "\text{(C)}"] \\
(N_\lambda,0) & (N_\lambda,1) \arrow[l, "\text{(B)}"'] & \cdots \arrow[l, "\text{(B)}"'] & (N_\lambda,M_\lambda-1) \arrow[l, "\text{(B)}"'] & (N_\lambda,M_\lambda) \arrow[l, "\text{(B)}"']
\end{tikzcd} \tag{8.53}$$

Example 8.1. Fix λ and drop the λ-index from $\vec{I}^{(\lambda,n,m)}$, $\Phi^{(\lambda,m)}$ and $\Psi^{(\lambda,n)}$. We showcase the recursion for $0 \leq n \leq 2$ and $0 \leq m \leq 2$.

1. $n = 0$: Using (A) twice,

$$\vec{I}^{(0,1)} = \Phi^{(1)} \cdot \vec{I}^{(0,0)} \tag{8.54}$$

$$\vec{I}^{(0,2)} = \Phi^{(2)} \cdot \vec{I}^{(0,1)}\,. \tag{8.55}$$

2. $n = 1$: Using (C) once,

$$\vec{I}^{(1,2)} = \Psi^{(1)} \cdot P_{1,0} \cdot \vec{I}^{(0,2)}\,. \tag{8.56}$$

   Using (B) twice,

$$\vec{I}^{(1,1)} = \Psi^{(1)} \cdot \left(P_{1,0} \cdot \vec{I}^{(0,1)} - 2\vec{I}^{(1,2)}\right) \tag{8.57}$$

$$\vec{I}^{(1,0)} = \Psi^{(1)} \cdot \left(P_{1,0} \cdot \vec{I}^{(0,0)} - 1\vec{I}^{(1,1)}\right). \tag{8.58}$$

3. $n = 2$: Using (C) once,

$$\vec{I}^{(2,2)} = \Psi^{(2)} \cdot \left(P_{1,1} \cdot \vec{I}^{(0,2)} + P_{1,0} \cdot \vec{I}^{(1,2)}\right). \tag{8.59}$$

Using (B) twice,

$$\vec{I}^{(2,1)} = \Psi^{(2)} \cdot \left(P_{1,1} \cdot \vec{I}^{(0,1)} + P_{1,0} \cdot \vec{I}^{(1,1)} - 2\vec{I}^{(2,2)}\right) \tag{8.60}$$

$$\vec{I}^{(2,0)} = \Psi^{(2)} \cdot \left(P_{1,1} \cdot \vec{I}^{(0,0)} + P_{1,0} \cdot \vec{I}^{(1,0)} - 1\vec{I}^{(2,1)}\right). \tag{8.61}$$

Note that this recursion can be performed over a finite field for the sake of efficiency. ■

The discussion above is summarized in the following algorithm.

Algorithm 4 : Logarithmic series expansion

Input:

- Pfaffian system in normal form: $\partial_i \bullet \vec{I}(z) = P_i(z) \cdot \vec{I}(z)$ for $i = 1, \ldots, N$.
- Integer $k > 0$.

Output: The series expansion coefficients $\vec{I}^{(\lambda,n,m)}$ from (8.44) up to order $\mathcal{O}(z_1^k)$.

```
Compute JordanDecomposition$[P_{1,-1}(z')]$ with $z'$ fixed to generic numbers. Save
    • unique eigenvalues $\{\lambda_1, \lambda_2, \ldots\}$
    • eigenvalue multiplicities $\{\Lambda_1, \Lambda_2, \ldots\}$
    • integers $\{M_{\lambda_1}, M_{\lambda_2}, \ldots\}$ from (8.46).
for each unique $\lambda$ do
    Repeat steps 1. to 5. in Algorithm (3) with the replacements
        • $\mathsf{R} \leftarrow \mathsf{R}^{(\lambda)} = \texttt{RowReduce}\left[\left(P_{1,-1}(z') - \lambda\mathbf{1}\right)^{M_\lambda+1}\right]$
        • $R' \leftarrow \Lambda$
        • $\vec{I}^{(0)} \leftarrow \vec{I}^{(\lambda,0,0)}$
    and save the result for $\vec{I}^{(\lambda,0,0)}$.
    for $0 \leq n \leq k$ and $0 \leq m \leq M_\lambda$ do
        Input $\vec{I}^{(\lambda,0,0)}$ into the recursion relations (A), (B), (C) to compute $\vec{I}^{(\lambda,n,m)}$.
    end for
end for
return The coefficient functions $\bigcup_\lambda \bigcup_{0\leq n\leq k} \bigcup_{0\leq m\leq M_\lambda} \vec{I}^{(\lambda,n,m)}$
```

We expect this to be an efficient method for computing series expansions of MIs. Its efficiency stems from trading a "big" Pfaffian system (satisfied by the full solution vector

$\vec{I}(z)$) for several "small" Pfaffian systems (satisfied by the restricted bases $\vec{J}^{(\lambda)}(z')$), each one containing one less variable. For instance, if the "big" system includes unrationalizable square roots and elliptic functions, then it is likely that the "small" systems only contain simple GPLs.

The catch is that the algorithm does not produce results valid in the entire phase space; there must exist a variable $z_1 \ll 1$ that can be used for the expansion. Luckily, there are many such situations in high-energy physics: z_1 could be a small mass, a threshold variable (e.g. $s - 4m^2$), an angle between two collinear momenta (e.g. in Regge theory), a small or large transverse momentum ($z_1 = p_T$ or $z_1 = 1/p_T$), a soft momentum variable in Post-Newtonian/Minkowskian gravity, and so forth.

The *method of regions* is the classic approach to asymptotic expansions [159, 160, 161]. More recent approaches include [162, 163, 164, 165, 166]. It would be interesting to import some of the wisdom from these studies in order to enhance the logarithmic restriction approach presented here.

Section 8.3

Restriction at the level of Macaulay matrices

While the protocol of the previous section is applicable to any regularly singular Pfaffian system, it does require an unrestricted system as input, which might be hard to obtain. It should also be mentioned that this method is only conjectured to yield the rational restriction of the twisted cohomology group (6.35) when it is viewed as a $\mathcal{D}$-module [83, Remark 3.5].

Building on the Macaulay matrix method introduced in Chapter (5), this section presents a holomorphic restriction protocol which works entirely at the level of $\mathcal{D}$-modules. Theorems from [83, Section 4] ensure that the method yields correct results. Some steps of the construction are technical, so we refer to the previous reference for details.

8.3.1 Restriction to a hyperplane

Let $\mathcal{I}$ be a holonomic ideal in $\mathcal{D}_Y$. This could e.g. be the annihilating ideal for the GKZ system. The goal is to compute a Pfaffian system for the rational restriction module onto the hyperplane given by $Y' = \{z_1 = \cdots = z_{N'} = 0\}$.

The first question to answer is that of a basis for the restriction module. In [83, Algorithm 4.7], a systematic search algorithm is formulated which succeeds when the dimension of the basis matches the expected holonomic rank R' of the restriction module.

The algorithm requires R' as input, but this is easily obtained for the case of Euler integrals by computing the Euler characteristic χ via the likelihood equations (6.36). Moreover, this algorithm relies entirely on numerical linear algebra, with all parameters fixed to generic numbers, meaning that it will terminate with modest computing power even for relatively large Pfaffian systems.

The resulting standard monomials for the restriction module, call them RStd, now serve as input for the Macaulay matrix M in Algorithm (1). Recall that M is built by acting with "seed" monomials on the generators of $\mathcal{I}$. Since $\mathcal{I}$ contains unrestricted variables defined on Y, so will M. The key observation is this: all the entries of M are polynomial in $z \in Y$, so we can simply fix $z_1 = \ldots = z_{N'} = 0$ in M, and then proceed with the usual Macaulay matrix algorithm.

Algorithm 5 : Rationally restricted Pfaffian system by the Macaulay matrix

Input:

- The restricted basis RStd.
- Generators $\{d_1, \ldots, d_D\}$ of a holonomic ideal $\mathcal{I}$ in $\mathcal{D}_Y$.
- Direction i.

Output: Pfaffian matrix $Q_i(z')$ for the restriction

$$\mathcal{R}_{Y'} \otimes_{\mathcal{D}_{Y'}} \mathcal{D}_Y / \big(\mathcal{I} + z_1\mathcal{D}_Y + \ldots + z_{N'}\mathcal{D}_Y\big)\,. \tag{8.62}$$

```
Call Algorithm (1) with the replacements
    • Std ← RStd
    • $M \leftarrow \left[\ M_{\text{Ext}} \ \middle|\ M_{\text{RStd}}\ \right]\Big|_{z_1=\ldots=z_{N'}=0}$
return $Q_i(z') = C^{(i)}_{\text{RStd}} - C^{(i)} \cdot M_{\text{RStd}}$
```

Let us remark that the basis RStd can also be guessed, assuming R' is known, by executing Algorithm (5) several times with all parameters set to numbers until it succeeds.

Algorithm (5) has been implemented in the ASIR package MT_MM by Nobuki Takayama, as showcased in the following example.

Example 8.1. Let $\mathcal{I} = \langle d_1, d_2 \rangle$ be the ideal generated by

$$d_1 = z_1(1-z_1)\partial_1^2 - z_1 z_2 \partial_1 \partial_2 + (c_1 - z_1(a + b_1 + 1))\partial_1 - b_1 z_2 \partial_2 - a b_1 \tag{8.63}$$

$$d_2 = z_2(1-z_2)\partial_2^2 - z_1 z_2 \partial_1 \partial_2 + (c_2 - z_2(a + b_2 + 1))\partial_2 - b_2 z_1 \partial_1 - a b_2\,, \tag{8.64}$$

which annihilates Appell's F_2 function

$$F_2(a,b_1,b_2,c_1,c_2|z_1,z_2) = \frac{\Gamma(c_1)\Gamma(c_2)}{\Gamma(b_1)\Gamma(b_2)\Gamma(c_1-b_1)\Gamma(c_2-b_2)} \times \int_{(0,1)^2} (1-x_1)^{c_1-b_1-1}(1-x_2)^{c_2-b_2-1}(1-z_1x_1-z_2x_2)^{-a}x_1^{b_1}x_2^{b_2}\frac{\mathrm{d}x}{x}\,. \tag{8.65}$$

This function appears often in the FI calculus [167, 168].

The following Asir script computes the holomorphic restriction to $Y' = \{z_1 = 0\}$.

```
import("mt_mm.rr")$

Vars = [z1, z2]$
DVars = [dz1, dz2]$

D1 = z1*(1-z1)*dz1^2 - z1*z2*dz1*dz2 + (c1-z1*(a+b1+1))*dz1 - b1*z2*dz2 - a*b1;
D2 = z2*(1-z2)*dz2^2 - z1*z2*dz1*dz2 + (c2-z2*(a+b2+1))*dz2 - b2*z1*dz1 - a*b2;
Ideal = [D1, D2];

NumericRule = [[z2,z2+1/3], [a,1/2], [b1,1/3], [b2,1/5], [c1,1/7], [c2,1/11]];
IdealNumeric = base_replace(Ideal, NumericRule);

RStd = mt_mm.restriction_to_pt_(IdealNumeric, Gamma = 2, KK = 4, Vars | p = 10^8);
RStd = reverse(map(dp_ptod, RStd[0], DVars));

Ideal = map(dp_ptod, Ideal, DVars);
MMData = mt_mm.find_macaulay(Ideal, RStd, Vars | restriction_var = [z1]);

Q2 = mt_mm.find_pfaffian(MMData, Vars, 2 | use_orig = 1);
```

The inputs of `mt_mm.restriction_to_pt_()` merit explanation. The list `IdealNumeric` contains generators with z_2 and exponent parameters fixed to generic numbers. The integers `Gamma` and `KK` refer to the γ and k of the basis search algorithm [83, Algorithm 4.7]. The value `p = 10^8` means that the function is executed over a finite field of order $p = 100000007$, which is more efficient than a symbolic calculation.

The result of this script is the Pfaffian system

$$\partial_2 \, \mathrm{RStd} = Q_2 \cdot \mathrm{RStd} \tag{8.66}$$

given by

$$\partial_2 \begin{bmatrix} 1 \\ \partial_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-ab_2}{z_2(z_2-1)} & \frac{c_2-(a+b_2+1)}{z_2(z_2-1)} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \partial_2 \end{bmatrix} . \tag{8.67}$$

■

8.3.2 Restriction to a hypersurface

This chapter has, up until now, only dealt with restrictions onto hyperplanes. Namely, we have assumed that $Y' = \{z_1 = \ldots = z_{N'} = 0\}$ is defined by an affine *linear* coordinate change. To date, there is no known algorithm for restrictions of $\mathcal{D}$-modules onto general hypersurfaces. Surprisingly, the Macaulay matrix restriction can be generalized to this scenario.

Suppose that $Y' = \{f = 0\}$ is the vanishing locus of an irreducible polynomial $f = f(z) \in \mathbb{C}[z_1, \ldots, z_N]$. The idea is, essentially, just to solve the Macaulay matrix equation (5.13) for $C^{(i)}$ modulo f:

$$C^{(i)}_{\mathrm{Ext}} \; - \; C^{(i)} \cdot M_{\mathrm{Ext}} \equiv 0 \mod f\,. \tag{8.68}$$

We give one example below to illustrate this procedure. See [169] for further results on restrictions onto hypersurface singularities of Horn's hypergeometric functions $\{H_1, \ldots, H_7\}$.

Example 8.2. We compute the restriction of Appell's F_4 function

$$\begin{gathered} F_4(a,b,c_1,c_2|z_1,z_2) = \frac{\Gamma(c_1)\Gamma(c_2)}{\Gamma(a)\Gamma(b)\Gamma(c_1-a)\Gamma(c_2-b)} \times \\ \int_{(0,1)^2} (1-x_1)^{c_1-a-1}(1-x_2)^{c_2-b-1}(1-z_1x_1)^{a-c_1-c_2+1} \times \\ (1-z_2x_2)^{b-c_1-c_2+1}(1-z_1x_1-z_2x_1)^{c_1+c_2-a-b-1}x_1^a x_2^b \frac{\mathrm{d}x}{x} \end{gathered} \tag{8.69}$$

onto its singular locus defined by

$$L = z_1 z_2 f = 0 \quad \text{with} \quad f = (z_1 - z_2)^2 - 2(z_1 + z_2) + 1\,. \tag{8.70}$$

In terms of Euler operators $\theta_i = z_i\,\partial_i$, the annihilating ideal $\mathcal{I} = \langle d_1, d_2 \rangle$ is generated by

$$d_1 = \theta_1(\theta_1 + c_1 - 1) - z_1(\theta_1 + \theta_2 + a)(\theta_1 + \theta_2 + b) \tag{8.71}$$

$$d_2 = \theta_2(\theta_2 + c_2 - 1) - z_2(\theta_1 + \theta_2 + a)(\theta_1 + \theta_2 + b)\,. \tag{8.72}$$

The holonomic rank outside the singular locus is $R = 4$. On $Y' = \{f = 0\}$, the rank drops to $R' = 3$. Both of these ranks assume that the exponent parameters (a, b, c_1, c_2) are generic.

We search for a restricted basis by executing Algorithm (5) with different guesses for RStd until it succeeds. The parameters are set to generic numbers in this search. The result is

$$\mathrm{RStd} = \left[1, \partial_1, \partial_2\right]^T . \tag{8.73}$$

Next we build a 6×10 Macaulay matrix $M_1 = \left[M_{\mathrm{Ext}} \middle| M_{\mathrm{RStd}}\right]$ of degree 1. The left block contains 7 columns labeled by $\mathrm{Ext} = \left[\partial_1^3,\, \partial_1^2\partial_2,\, \partial_1\partial_2^2,\, \partial_2^3,\, \partial_1^2,\, \partial_1\,\partial_2,\, \partial_2^2\right]^T$.

The goal is now to solve (8.68) modulo f. Since f is quadratic in z_1, we can eliminate all instances of z_1^p for $p \geq 2$ via polynomial division by f. It is therefore enough to consider the ansatz

$$C^{(i)} = C_0^{(i)} + C_1^{(i)} z_1 , \tag{8.74}$$

where $C_{1,2}^{(i)}$ are independent of z_1. The reduction modulo f can be performed via a sequence of replacements $z_1^p \to g_p(z_2) + h_p(z_2) z_1$. For instance, $f = 0$ immediately gives

$$z_1^2 \to (-1 + 2z_2 - z_2^2) + 2(1 + z_2) z_1 . \tag{8.75}$$

Inserting this into $z_1 f = 0$, we further have

$$z_1^3 \to 2(-1 + z_2 + z_2^2 - z_2^3) + (3 + 10z_2 + 3z_2^2) z_1 , \tag{8.76}$$

and so forth.

After solving for $C_{1,2}^{(i)}$, the Pfaffian matrices are computed as usual by

$$Q_i = C_{\mathrm{RStd}}^{(i)} - C^{(i)} \cdot M_{\mathrm{RStd}} . \tag{8.77}$$

Setting $(a, b, c_1, c_2) = (-2/3,\ 1/3,\ 1/3,\ 1/3)$ to simplify the output, we finally get

$$\begin{aligned} Q_1 &= \begin{bmatrix} 0 & 1 & 0 \\ -\frac{5(3z_1z_2+z_1-5z_2(z_2+2)-1)}{162(z_2-1)^3 z_2} & \frac{z_1(41z_2-5)-z_2(87z_2+62)+5}{108(z_2-1)^2 z_2} & \frac{5(3z_1z_2+z_1-5z_2(z_2+2)-1)}{108(z_2-1)^3} \\ \frac{4(-z_1+2z_2+2)}{81(z_2-1)^2 z_2} & -\frac{2}{27z_2} & \frac{2(z_1-2(z_2+1))}{27(z_2-1)^2} \end{bmatrix} \\ Q_2 &= \begin{bmatrix} 0 & 0 & 1 \\ \frac{4(-z_1+2z_2+2)}{81(z_2-1)^2 z_2} & -\frac{2}{27z_2} & \frac{2(z_1-2(z_2+1))}{27(z_2-1)^2} \\ \frac{5(-z_1+3z_2+1)}{162(z_2-1)z_2^2} & -\frac{5(z_1+z_2-1)}{108z_2^2} & \frac{5z_1-51z_2+31}{108(z_2-1)z_2} \end{bmatrix} \end{aligned} \tag{8.78}$$

See the script [169, `2023-01-26-F4-on-sing.py`] for comparisons of known analytic results with numerical solutions to this Pfaffian system. ■

Chapter 9

Restriction Examples

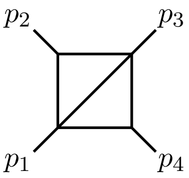
This chapter gives examples of how to compute Pfaffian systems for restriction modules related to FIs. Sections (9.2) and (9.3) also include partial results for logarithmic series expansions related to 1- and 2-loop Bhabha scattering.

Section 9.1

Holomorphic restriction: 2-loop N-box

This is a continuation of Example (8.1). The goal is to apply the holomorphic restriction protocol of Section (8.2.3) to that GKZ system. Schematically, we restrict

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ 1 & \cdot & \cdot & 1 & 1 & 1 & \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 1 & 1 & \cdot & 1 \\ \cdot & 1 & \cdot & \cdot & 1 & \cdot & 1 & \cdot & 1 & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & 1 & \cdot & 1 & \cdot & 1 \end{bmatrix} \xrightarrow[i\neq 10]{z_i=1}$$



$$. \qquad (9.1)$$

We begin with a rank $R = 9$ Pfaffian system, obtained via the Macaulay matrix method, in terms of variables $z \in Y = \mathbb{A}^{10}$. After a coordinate shift $(z_1, \ldots, z_9, z_{10}) \to (z_1, \ldots, z_9, z_{10}) + (1, \ldots, 1, 0)$, the restriction surface is $Y' = \{z_1 = \ldots = z_9 = 0\}$. Six variables can immediately be restricted using homogeneity (cf. Section (6.5)), say $\{z_1, z_2, z_3, z_4, z_5, z_9\}$. So the restriction protocol need only be applied to $\{z_6, z_7, z_8\}$. By calculating the Euler characteristic on Y', we expect a rank $R' = 3$ Pfaffian system in the end.

9.1.1 Basis

The method of [118] (implemented in the ASIR command `mt_gkz.cbase_by_euler(A)`) gives a basis of 9 standard monomials

$$\mathrm{Std} = \begin{bmatrix} \partial_8^2 \\ \partial_6\partial_{10} \\ \partial_7\partial_{10} \\ \partial_{10}^2 \\ \partial_6 \\ \partial_7 \\ \partial_8 \\ \partial_{10} \\ 1 \end{bmatrix} . \tag{9.2}$$

The Macaulay matrix method then outputs a Pfaffian system

$$\partial_i \, \mathrm{Std} = P_i \cdot \mathrm{Std} \quad \text{for} \quad i = 6, 7, 8, 10 \,. \tag{9.3}$$

For the sake of verifying our results, an independent IBP calculation is performed for the following MIs in D = 4 − 2ϵ dimensions:

$$\begin{bmatrix} I_{02202} \\ I_{22020} \\ I_{11111} \end{bmatrix} = G_0 \cdot e^{(I)} \,. \tag{9.4}$$

The subscripts on the integrals label powers of the inverse propagators

$$D_1 = \ell_1^2 \,, \; D_2 = \ell_2^2 \,, \; D_3 = (p_1 - \ell_1)^2 \,, \; D_4 = (p_{12} - \ell_{12})^2 \,, \; D_5 = (-p_{123} + \ell_{12})^2 \,, \tag{9.5}$$

and G_0 is a matrix containing powers of $s = 2p_1 \cdot p_2$ that render $e^{(I)}$ dimensionless.

We would like to land on the basis (9.4) after computing the restriction of (9.3), in order to compare our Pfaffian matrices with those coming from IBPs. By the formulas of Section (6.3.3), we therefore translate the integrals (9.3) into $\mathcal{D}$-module elements:

$$e^{(I)} \to e^{(\mathcal{D})} = \mathrm{c} \cdot \begin{bmatrix} \partial_{10}^2 \\ \epsilon(2 + 5\delta)(1 + 2\epsilon + 5\epsilon\delta) + 2z_{10}(1 + 2\epsilon + 5\epsilon\delta)\partial_{10} + z_{10}^2\partial_{10} \\ (1 + \epsilon + 2\epsilon\delta)\partial_{10} + z_{10}\partial_{10}^2 - z_7\partial_7\partial_{10} + z_6\partial_6\partial_{10} \end{bmatrix} . \tag{9.6}$$

The 3×3 diagonal matrix c contains prefactors depending on $\{\epsilon, \delta, s\}$. Thus, there is a 3×9 B-matrix (8.27) with rows and columns labeled by $e^{(\mathcal{D})}$ and Std respectively:

$$\mathsf{B} = \begin{bmatrix} \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & z_{10}^2 & \cdot & \cdot & \cdot & 2z_{10}(1+2\epsilon+5\epsilon\delta) & \epsilon(2+5\delta)(1+2\epsilon+5\epsilon\delta) \\ \cdot & z_6 & -z_7 & z_{10} & \cdot & \cdot & \cdot & 1+\epsilon+2\epsilon\delta & \cdot \end{bmatrix} . \tag{9.7}$$

This matrix satisfies[1] $e^{(\mathcal{D})} = \mathrm{c} \cdot \mathsf{B} \cdot \mathrm{Std}$.

9.1.2 Normal form

The Pfaffian system (9.3) must be brought to normal form before the restriction procedure can be applied (cf. Section (8.1.3)). The system turns out to be in normal form w.r.t. the variables z_6 and z_7, but P_8 has a second-order pole in z_8:

$$P_8\big|_{z_6,\, z_7 \to 1} = (z_8 - 1)^2\, P_{8,-2}\big|_{z_6,\, z_7 \to 1} + \dots . \tag{9.8}$$

The second-order pole can be cured by Moser reduction, namely a gauge transformation by

$$G_1 = \texttt{Diagonal}\big[(z_8-1)^{-1},\, 1, \dots, 1\big] . \tag{9.9}$$

The notation

$$G[P_i] = G^{-1} \cdot (P_i \cdot G - \partial_i \bullet G) \tag{9.10}$$

is used throughout this chapter to denote gauge transformations. In addition to curing the second-order pole, the new residue matrices $G[P_i]_{-1}, i \in \{6, 7, 8\}$, also obtain non-resonant spectra. Consequently, the system is in normal form. To simplify notation, let us

$$\text{replace } G_1[P_i] \;\to\; P_i \text{ in the rest of this example.} \tag{9.11}$$

[1] One may employ (A.10) in cases where $e^{(\mathcal{D})}$ contains some monomials that are not contained in Std.

9.1.3 Restriction

The Pfaffian-level restriction protocol implores us to build the matrix

$$\mathsf{M} = \left[\begin{array}{c} \mathsf{B} \\ \hline \mathsf{R} \end{array}\right] \tag{9.12}$$

coming from (8.30). The B-matrix was already found in (9.7). The R-matrix is constructed from the independent rows of all the (gauged transformed) residue matrices:

$$\mathsf{R} = \texttt{RowReduce}\left[\begin{array}{c} P_{6,-1} \\ \hline P_{7,-1} \\ \hline P_{8,-1} \end{array}\right] . \tag{9.13}$$

We remind that the `RowReduce`[...] operation includes the deletion of zero-rows.

The M-matrix can now be inserted into (8.31), giving

$$\left(\partial_{10} \bullet \mathsf{M} + \mathsf{M} \cdot P_{10,0}\right) \cdot \mathsf{M}^{-1} = \left[\begin{array}{c|c} Q_{10} & \star \\ \hline \mathbf{0}_{6\times 3} & \star \end{array}\right] , \tag{9.14}$$

where

$$P_{10,0} = P_{10}\big|_{z_6,\, z_7,\, z_8 \to 0} , \tag{9.15}$$

which is well-defined because the system is in normal form. Sending the regulator $\delta \to 0$ in Q_{10}, the final result is

$$Q_{10} = \begin{bmatrix} -\frac{2(1+\epsilon)}{z_{10}} & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ -\frac{z_{10}\epsilon}{6(1+z_{10})} & \frac{\epsilon}{6z_{10}(1+z_{10})} & -\frac{z_{10}+2\epsilon}{z_{10}(1+z_{10})} \end{bmatrix} . \tag{9.16}$$

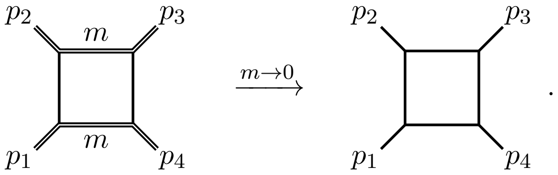
This Pfaffian matrix agrees with an independent IBP calculation for the basis $e^{(I)}$ in (9.4), where $z_{10} = t/s$ is a ratio of Mandelstam variables.

In conclusion, the $\mathcal{D}$-module associated to the massless 2-loop N-box diagram is isomorphic to the rational restriction of a GKZ system.

Section 9.2

Logarithmic restriction: 1-loop Bhabha scattering

This example uses Algorithm (4) to compute a logarithmic restriction. Our goal is the small-mass expansion of the 1-loop box topology for Bhabha scattering:



The momentum space integral family in $\mathrm{D} = 4-2\epsilon$ dimensions is

$$I_{\nu_1\ldots\nu_5}(m) = \int \frac{\mathrm{d}^{\mathrm{D}}\ell}{[\ell^2-m^2]^{\nu_1}\,[(\ell+p_1)^2]^{\nu_2}\,[(\ell+p_{12})^2-m^2]^{\nu_3}\,[(\ell+p_{123})^2]^{\nu_4}}\,, \tag{9.17}$$

and the kinematic variables are

$$p_1^2 = p_2^2 = p_3^2 = p_4^2 = m^2, \quad s = p_{12}^2\,, \quad t = p_{23}^2\,. \tag{9.18}$$

IBP reduction reveals a basis of 5 MIs:

$$\begin{bmatrix} I_{0010}(m) \\ I_{0101}(m) \\ I_{1010}(m) \\ I_{0111}(m) \\ I_{1111}(m) \end{bmatrix} = G_0 \cdot \vec{I}\,, \quad G_0 = (-s)^{-\epsilon}\,\texttt{Diagonal}\big[(-s),\,1,\,1,\,(-s)^{-1},\,(-s)^{-2}\big]\,. \tag{9.19}$$

The purpose of G_0 is to factor out the mass dimension of the integrals. Using IBPs, the MIs $\vec{I}$ are found to obey a Pfaffian system

$$\partial_i \bullet \vec{I} = P_i \cdot \vec{I} \quad \text{with} \quad z_1 = \frac{m^2}{-s} \quad \text{and} \quad z_2 = \frac{t}{s}\,. \tag{9.20}$$

Using the method of Section (8.2.4), we now compute the logarithmic restriction of solutions onto $Y' = \{z_1 = 0\}$, i.e. for $m \sim 0$.

9.2.1 Normal form

The matrices $P_{1,2}$ in (9.20) are not in normal form w.r.t. Y'. In particular, P_1 has a second-order pole there, and P_2 a first-order pole. This is cured via Moser reduction:

$$\vec{I} \to G_1 \cdot \vec{I} \quad \text{with} \quad G_1 = \texttt{Diagonal}[z_1,\, 1,\, 1,\, 1,\, 1]\ . \tag{9.21}$$

To simplify notation, we

$$\text{replace } G_1[P_i] \ \to\ P_i \text{ in this example.} \tag{9.22}$$

The gauge transformed Pfaffian matrices now have the desired pole structure:

$$\begin{aligned} P_1 &= z_1^{-1} &\times\ & P_{1,0}(z_2) &+\ & \mathcal{O}(z_1^0) \\ P_2 &= z_1^0 &\times\ & P_{2,0}(z_2) &+\ & \mathcal{O}(z_1)\,. \end{aligned} \tag{9.23}$$

In addition, the spectrum of the residue matrix $P_{1,-1}$ is non-resonant, meaning that the system is now in normal form.

From

$$\texttt{JordanDecomposition}[P_{1,-1}] = \begin{bmatrix} 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & -\epsilon & 1 \\ \cdot & \cdot & \cdot & \cdot & -\epsilon \end{bmatrix} \tag{9.24}$$

we read off the data

$$\begin{aligned} \{\lambda_1,\, \lambda_2\} &= \{0,\, -\epsilon\} \\ \{\Lambda_1,\, \Lambda_2\} &= \{3,\, 2\} \\ \{M_{\lambda_1},\, M_{\lambda_2}\} &= \{0,\, 1\}\,, \end{aligned} \tag{9.25}$$

in the notation of step 1. of Algorithm (4). The ansatz for the logarithmic expansion (8.44) thus takes the form

$$\begin{aligned} \vec{I} \quad &= \quad z_1^{\lambda_1} \sum_{n=0}^{\infty} \vec{I}^{\,(\lambda_1,n)}(z_2) z_1^n \\ &+ \quad z_1^{\lambda_2} \sum_{n=0}^{\infty} \left[\vec{I}^{\,(\lambda_2,n,0)}(z_2) \ + \ \vec{I}^{\,(\lambda_2,n,1)}(z_2) \times \log(z_1)\right] z_1^n\,. \end{aligned} \tag{9.26}$$

The logarithm appears due to the "1" in the λ_2-eigenvalue block of (9.24). Recall that the vector $\vec{I}^{(\lambda_2,0,1)}$ is immediately obtained from $\vec{I}^{(\lambda_2,0,0)}$ via the recursion relation (A). Higher-order terms in z_1^n are obtained from the recursions (B) and (C). If what follows, we determine $\vec{I}^{(\lambda_1,0)}$ from the solution to a Pfaffian system of rank $\Lambda_1 = 3$, and $\vec{I}^{(\lambda_2,0,0)}$ from the solution to a system of rank $\Lambda_2 = 2$. Both systems are univariate.

9.2.2 Eigenvalue $\lambda_1 = 0$

There is no logarithm for the eigenvalue λ_1, so this case can in fact be treated with the holomorphic restriction of Section (8.2.3).

Rank jump

Recall the constraint from the second equation of (8.34):

$$\begin{aligned}
\mathsf{R}^{(\lambda_1)} \cdot \vec{I}^{(\lambda_1,0)} &= \texttt{RowReduce}[P_{1,-1}] \cdot \vec{I}^{(\lambda_1,0)} && (9.27)\\
&= \begin{bmatrix} 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \frac{z_2\epsilon}{1-2\epsilon} & \cdot \end{bmatrix} \cdot \vec{I}^{(\lambda_1,0)} && (9.28)\\
&= 0\,. && (9.29)
\end{aligned}$$

Two constraints are here imposed on the entries of the 5-dimensional vector $\vec{I}^{(\lambda_1,0)}$. This is the reason why the rank drops to $5-2=3$.

The entries of $\vec{I}^{(\lambda_1,0)}$ apparently correspond to the MIs (9.19) with the limit $z_1 \to 0$ taken at the *integrand* level[2], i.e. setting $m^2 = 0$ inside the propagators of (9.17). The interpretation of the two constraints from (9.28) is now clear:

1. The first constraint $I_1^{(\lambda_1,0)} = 0$ means that the massless tadpole $I_{0010}(m=0)$ should vanish, which is indeed true in DR.

2. The second constraint $I_2^{(\lambda_1,0)} + \frac{z_2\epsilon}{1-2\epsilon} I_4^{(\lambda_1,0)} = 0$ is an IBP relation between the massless t-channel bubble $I_{0101}(m=0)$ and the massless triangle $I_{0111}(m=0)$. We have verified this relation using LiteRed.

Basis

Following step 2. of Algorithm (3), a basis for the restricted Pfaffian system can be found by simply eliminating linearly dependent entries of $\vec{I}^{(\lambda_1,0)}$. We should eliminate two

[2] This is called the *hard region* in the method of regions [159, 160].

integrals according to the above discussion, so we have[3]

$$\vec{J}^{(\lambda_1)} = \begin{bmatrix} I_2^{(\lambda_1,0)} \\ I_3^{(\lambda_1,0)} \\ I_5^{(\lambda_1,0)} \end{bmatrix} = (-s)^{\epsilon} \begin{bmatrix} I_{0101}(0) \\ I_{1010}(0) \\ (-s)^2 I_{1111}(0) \end{bmatrix} . \tag{9.30}$$

The factors of $(-s)$ stem from the matrix G_0 in (9.19). The B-matrix (8.27) is therefore

$$\mathsf{B}^{(\lambda_1)} = \begin{bmatrix} \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 \end{bmatrix} \quad \text{such that} \quad \vec{J}^{(\lambda_1)} = \mathsf{B}^{(\lambda_1)} \cdot \vec{I}^{(\lambda_1,0)} . \tag{9.31}$$

Restriction

The R- and B-matrices from (9.28) and (9.31) combine to

$$\mathsf{M}^{(\lambda_1)} = \left[\begin{array}{c} \mathsf{B}^{(\lambda_1)} \\ \hline \mathsf{R}^{(\lambda_1)} \end{array}\right] \quad \text{such that} \quad \mathsf{M}^{(\lambda_1)} \cdot \vec{I}^{(\lambda_1,0)} = \left[\begin{array}{c} \vec{J}^{(\lambda_1)} \\ \hline 0 \\ 0 \end{array}\right] . \tag{9.32}$$

The 3×3 Pfaffian matrix $Q_2^{(\lambda_1)}$ associated to $\vec{J}^{(\lambda_1)}$ is now computed with (8.31). Namely, by gauge transforming $P_{2,0}$ from (9.23) with $\left(\mathsf{M}^{(\lambda_1)}\right)^{-1}$, we get

$$\left(\partial_2 \bullet \mathsf{M}^{(\lambda_1)} + \mathsf{M}^{(\lambda_1)} \cdot P_{2,0}\right) \cdot \left(\mathsf{M}^{(\lambda_1)}\right)^{-1} = \left[\begin{array}{c|c} Q_2^{(\lambda_1)} & \star \\ \hline \mathbf{0}_{2\times 3} & \star \end{array}\right] . \tag{9.33}$$

Explicitly, we have derived the Pfaffian system

$$\partial_2 \bullet \vec{J}^{(\lambda_1)} = Q_2^{(\lambda_1)} \cdot \vec{J}^{(\lambda_1)} \quad \text{with} \quad Q_2^{(\lambda_1)} = \begin{bmatrix} -\frac{\epsilon}{z_2} & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \frac{2(1-2\epsilon)}{z_2^2(z_2+1)} & \frac{2(2\epsilon-1)}{z_2(z_2+1)} & \frac{-z_2-\epsilon-1}{z_2(z_2+1)} \end{bmatrix} . \tag{9.34}$$

This system agrees with an independent IBP calculation performed directly on the basis of massless integrals (9.30). See [146] for a pedagogical exposition on how to solve these equations by passing to a canonical basis.

[3] We choose the bubble instead of the triangle for the first basis element since it is preferred to have basis elements with the fewest number of distinct propagators.

9.2.3 Eigenvalue $\lambda_2 = -\epsilon$

The Pfaffian system associated to the eigenvalue $\lambda_2 = -\epsilon$ turns out to be even simpler than the one for $\lambda_1 = 0$. Let's build it.

Rank jump

The rank jump constraint from (8.47) given $M_{\lambda_2} = 1$ reads

$$\begin{aligned}\mathsf{R}^{(\lambda_2)} \cdot \vec{I}^{(\lambda_2,0,0)} &= \texttt{RowReduce}\left[(P_{1,-1} - \lambda_2 \mathbf{1})^2\right] \cdot \vec{I}^{(\lambda_2,0,0)} &(9.35)\\ &= \begin{bmatrix} 1 & \cdot & \cdot & \frac{z_2\epsilon}{1-\epsilon} & \frac{z_2\epsilon}{2(1-\epsilon)} \\ \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot \end{bmatrix} \cdot \vec{I}^{(\lambda_2,0,0)} &(9.36)\\ &= 0\,. &(9.37)\end{aligned}$$

There are 3 equations imposed on the entries of the 5-dimensional vector $\vec{I}^{(\lambda_2,0,0)}$, wherefore the rank drops to 2.

In this case, it is less clear what the relations (9.36) mean! The present author does not know what the momentum space representations looks like for other eigenvalues besides $\lambda = 0$. This is worth clarifying, since it would help with the determination of boundary constants via direct integration.

Basis

The constraints (9.36) fix $I_2^{(\lambda_2,0,0)} = I_3^{(\lambda_2,0,0)} = 0$, and $I_5^{(\lambda_2,0,0)}$ becomes a linear combination of $I_1^{(\lambda_2,0,0)}$ and $I_4^{(\lambda_2,0,0)}$. So

$$\vec{J}^{(\lambda_2)} = \begin{bmatrix} I_1^{(\lambda_2,0,0)} \\ I_4^{(\lambda_2,0,0)} \end{bmatrix} \tag{9.38}$$

is a linearly independent basis. The B-matrix is therefore given by

$$\mathsf{B}^{(\lambda_2)} = \begin{bmatrix} 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot \end{bmatrix} \quad \text{such that} \quad \vec{J}^{(\lambda_2)} = \mathsf{B}^{(\lambda_2)} \cdot \vec{I}^{(\lambda_2,0,0)}\,. \tag{9.39}$$

Restriction

Stacking the R- and B-matrices from (9.36) and (9.39) on top of each other, we get

$$\mathsf{M}^{(\lambda_2)} = \left[\begin{array}{c} \mathsf{B}^{(\lambda_2)} \\ \hline \mathsf{R}^{(\lambda_2)} \end{array}\right] \quad \text{such that} \quad \mathsf{M}^{(\lambda_2)} \cdot \vec{I}^{(\lambda_2,0,0)} = \left[\begin{array}{c} \vec{J}^{(\lambda_2)} \\ \hline 0 \\ 0 \\ 0 \end{array}\right] . \tag{9.40}$$

The Pfaffian system for $\vec{J}^{(\lambda_2)}$ now follows from (8.31):

$$\left(\partial_2 \bullet \mathsf{M}^{(\lambda_2)} + \mathsf{M}^{(\lambda_2)} \cdot P_{2,0}\right) \cdot \left(\mathsf{M}^{(\lambda_2)}\right)^{-1} = \left[\begin{array}{c|c} Q_2^{(\lambda_2)} & \star \\ \hline \mathbf{0}_{3\times 2} & \star \end{array}\right] . \tag{9.41}$$

Explicitly, we have that

$$\partial_2 \bullet \vec{J}^{(\lambda_2)} = Q_2^{(\lambda_2)} \cdot \vec{J}^{(\lambda_2)} \quad \text{with} \quad Q_2^{(\lambda_2)} = \begin{bmatrix} 0 & 0 \\ \frac{\epsilon-1}{z_2^2} & \frac{-1}{z_2} \end{bmatrix} . \tag{9.42}$$

The general solution is swiftly found to be

$$\vec{J}^{(\lambda_2)} = \begin{bmatrix} c_1(\epsilon) \\ \frac{c_1(\epsilon)(1-\epsilon)\log(z_2)}{z_2} + \frac{c_2(\epsilon)}{z_2} \end{bmatrix} , \tag{9.43}$$

where $c_{1,2}(\epsilon)$ are two boundary constants.

9.2.4 Result for the logarithmic series

The solution vectors in (9.26) at 0th order in z_1 take the form

$$\vec{I}^{(\lambda_1,0)} = \left(\mathsf{M}^{(\lambda_1)}\right)^{-1} \cdot \left[\begin{array}{c} \vec{J}^{(\lambda_1)} \\ \hline 0 \\ 0 \end{array}\right] \tag{9.44}$$

$$\vec{I}^{(\lambda_2,0,0)} = \left(\mathsf{M}^{(\lambda_2)}\right)^{-1} \cdot \left[\begin{array}{c} \vec{J}^{(\lambda_2)} \\ \hline 0 \\ 0 \\ 0 \end{array}\right] \tag{9.45}$$

$$\vec{I}^{(\lambda_2,0,1)} = \left(P_{1,-1} - \lambda_2 \mathbf{1}\right) \cdot \vec{I}^{(\lambda_2,0,0)} , \tag{9.46}$$

where the last equation comes from the recursion (A). The vector $\vec{J}^{(\lambda_1)}$ comes from solving the massless 1-loop box system (9.34), and $\vec{J}^{(\lambda_2)}$ was given in (9.43). We do not include higher-order terms in z_1^n (coming from the recursions (B) and (C)) in order to show a compact, final result.

Inserting the solution vectors above into (9.26) we get[4]

$$\begin{bmatrix} I_{0010}(m) \\ I_{0101}(m) \\ I_{1010}(m) \\ I_{0111}(m) \\ I_{1111}(m) \end{bmatrix} \overset{m\to 0}{=} \begin{bmatrix} 0 \\ (-s)^{-\epsilon} I_{0101}(0) \\ (-s)^{-\epsilon} I_{1010}(0) \\ \frac{(-s)^{1-\epsilon}(1-2\epsilon)}{t\epsilon} I_{0101}(0) \\ (-s)^{-2-\epsilon} I_{1111}(0) \end{bmatrix} + \begin{bmatrix} \frac{c_1}{\epsilon-1}(m^2)^{1-\epsilon} \\ 0 \\ 0 \\ \frac{-(m^2)^{-\epsilon}}{t}\left[c_1 \log(-t/m^2) + c_2\right] \\ \frac{-2(m^2)^{-\epsilon}}{st\epsilon}\left[c_1(\epsilon \log(-t/m^2) - 1) + \epsilon c_2\right] \end{bmatrix} \tag{9.47}$$

The first and second terms on the LHS come from λ_1 and λ_2 respectively. Here we reinstated all prefactors, wrote $z_1 = m^2/(-s)$ and $z_2 = t/s$, and lastly rescaled $c_1 \to \frac{c_1}{\epsilon-1}$ to simplify the expression. Note that only the triangle and the box integrals receive logarithmic corrections. The bubbles would naturally receive power corrections in z_1, which we are neglecting.

The two boundary constants are determined as follows.

- c_1 is fixed by comparing with the exact expression $I_{0010}(m) = -\Gamma(1-\epsilon)(m^2)^{1-\epsilon}$ for the tadpole on the LHS. This gives $c_1 = (1-\epsilon)\Gamma(\epsilon-1)$.

- $c_2 = c_2(\epsilon)$ can be fixed order-by-order in ϵ by numerically matching with an integral on the LHS, say the 5th element $I_{1111}(m)$. On the LHS we insert the ansatz $c_2 = \sum_{i=-2}^{2} c_2^{(i)} \epsilon^i$, which starts at $i = -2$ because the massless box $I_{1111}(0)$ has a double pole in ϵ. On the RHS we numerically evaluate $I_{1111}(m)$ at the point $(s, t, m^2) = (-1, -9, 10^{-3})$ with AMFlow [34]. By comparing the LHS and RHS we get

  $$\left[c_2^{(-2)}, c_2^{(-1)}, c_2^{(0)}, c_2^{(1)}, c_2^{(2)}\right] = \left[1, -0.588997, -7.12155, -4.69031, -20.4068\right].$$

  AMFlow is easily able to compute $\mathcal{O}(10^2)$ digits for this topology. Analytical expressions for these constants could therefore, if needed, be found with `PSLQ` [77].

Evaluating at other phase space points with small m^2, we find that the LHS of (9.47) is a fair approximation to the ϵ-expansions of $I_{0111}(m)$ and $I_{1111}(m)$ up to $\mathcal{O}(\epsilon^3)$. For

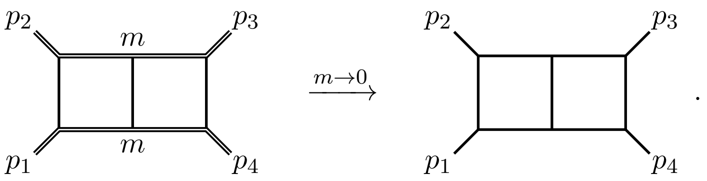
[4] We are neglecting the DR scale μ in factors such as $(-s/\mu^2)^{-\epsilon}$.

instance, at $(s,t,m^2)=(-1,-8,10^{-3})$ we find agreement within 0.01% accuracy for the triangle, and within 0.001% accuracy for the box. The non-leading terms in ϵ for the bubbles $I_{0101}(m)$ and $I_{1010}(m)$ give errors between $5-20\%$ because we have neglected power corrections.

Section 9.3

Logarithmic restriction: 2-loop Bhabha scattering

This is another example of logarithmic restriction using Algorithm (4). We provide partial results on the small-mass logarithmic expansion of the 2-loop Bhabha double-box topology:



(9.48)

The momentum space integral family in $\mathrm{D}=4-2\epsilon$ dimensions reads

$$I_{\nu_1\ldots\nu_9}(m)=\int\frac{\mathrm{d}^{\mathrm{D}}\ell_1\mathrm{d}^{\mathrm{D}}\ell_2}{D_1^{\nu_1}\cdots D_9^{\nu_9}}\,, \tag{9.49}$$

with

$$\begin{array}{lllllllll} D_1 &=& \ell_1^2-m^2 & D_2 &=& (\ell_1+p_1+p_2)^2-m^2 & D_3 &=& \ell_2^2-m^2\\ D_4 &=& (\ell_2+p_1+p_2)^2-m^2 & D_5 &=& (\ell_1+p_1)^2 & D_6 &=& (\ell_1-\ell_2)^2\\ D_7 &=& (\ell_2-p_3)^2 & D_8 &=& (\ell_2+p_1)^2 & D_9 &=& (\ell_1-p_3)^2\,, \end{array}$$

where D_8 and D_9 denote ISPs. Solutions for this integral family were found in [170] (see also [171]).

Inspired by [170], we choose the following basis for the unrestricted Pfaffian system:

$$\begin{array}{rllllllll} G_0\cdot\vec{I}=\big[& I_{202000000}, & I_{000022100}, & I_{001022000}, & I_{122000000}, & I_{012002000}, & I_{022001000},\\ & I_{001112000}, & I_{012011000}, & I_{100011200}, & I_{121200000}, & I_{011011100}, & I_{011012100},\\ & I_{021011100}, & I_{022011100}, & I_{101111000}, & I_{101211000}, & I_{102011100}, & I_{101011200},\\ & I_{110011200}, & I_{111011100}, & I_{1110111-10}, & I_{111111100}, & I_{1111111-10} & \big]^T. \end{array} \tag{9.50}$$

The 23×23 diagonal matrix G_0 factors out the mass dimension of each integral via powers of $(-s)$. IBP reduction with KIRA [10] then provides a Pfaffian system $\partial_i \bullet \vec{I} = P_i \cdot \vec{I}$ in terms of the kinematic variables

$$z_1 = \frac{m^2}{-s} \quad \text{and} \quad z_2 = \frac{t}{s} \,. \tag{9.51}$$

Our goal is to restrict onto the surface $Y' = \{z_1 = 0\}$.

9.3.1 Normal form

The Pfaffian system associated to the basis (9.50) is not in normal form. The first issue to resolve is the higher-order singularities on Y':

$$\begin{aligned} P_1 &= z_1^{-3} \times P_{1,-3}(z_2) + \mathcal{O}(z_1^{-2}) \\ P_2 &= z_1^{-2} \times P_{2,-2}(z_2) + \mathcal{O}(z_1^{-1})\,. \end{aligned} \tag{9.52}$$

Moser reduction suggests to gauge transform with

$$G_1 = \texttt{Diagonal}\big[\underbrace{z_1, \ldots, z_1}_{8\text{ times}}, 1, \ldots, 1\big] \quad \text{and} \quad G_2 = \texttt{Diagonal}\big[\underbrace{z_1, \ldots, z_1}_{12\text{ times}}, 1, \ldots, 1\big]\,.$$

Introducing the shorthand notation $G_{i\ldots j} = G_i \cdot \ldots \cdot G_j$, then the new pole structure becomes

$$\begin{aligned} G_{21}[P_1] &= z_1^{-1} \times G_{21}[P_1]_{-1} + \mathcal{O}(z_1^0) \\ G_{21}[P_2] &= z_1^{-1} \times G_{21}[P_2]_{-1} + \mathcal{O}(z_1^0)\,. \end{aligned} \tag{9.53}$$

So we have cured the higher-order poles in P_1. But P_2 is still not finite on Y' because the residue matrix $G_{21}[P_1]_{-1}$ has a resonant spectrum[5]. In [83, Appendix B], it is shown how to construct a sequence of gauge transformations (involving the Jordan decomposition of the residue matrix) to cure the spectrum[6]. It takes a matter of seconds on a laptop to compute these gauge transformations, even without fixing z-variables to generic numbers (recall that the Jordan decomposition is independent of z). Writing G_3 for the product of said gauge matrices, we get

$$G_{321}[P_2] = G_{321}[P_2]_0 + \mathcal{O}(z_1)\,, \tag{9.54}$$

[5] See [83, Appendix B] for a proof of why non-resonance of the residue matrix $P_{1,-1}$ implies finiteness of the remaining Pfaffian matrices $\{P_2, \ldots P_N\}$ on $\{z_1 = 0\}$.

[6] The MATHEMATICA scripts in [169] contain the gauge transformations for this particular example.

wherefore the system is now in normal form. For brevity, we

$$\text{replace } G_{321}[P_i] \;\to\; P_i \text{ in the rest of this example.} \tag{9.55}$$

$P_{1,-1}$ denotes the residue matrix in this newfound basis. Its Jordan form enjoys an eigenvalue decomposition

$$\texttt{JordanDecomposition}[P_{1,-1}] = \mathsf{J}^{(\lambda_1)} \oplus \mathsf{J}^{(\lambda_2)} \oplus \mathsf{J}^{(\lambda_3)} \oplus \mathsf{J}^{(\lambda_4)} \tag{9.56}$$

in terms of block matrices

$$\begin{aligned}
\mathsf{J}^{(\lambda_1)} &= \mathbf{0}_{8\times 8}\,, \quad \mathsf{J}^{(\lambda_2)} = \begin{bmatrix} \lambda_2 \end{bmatrix} \\
\mathsf{J}^{(\lambda_3)} &= \begin{bmatrix}
\lambda_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \lambda_3 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \lambda_3 & 1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \lambda_3 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \lambda_3 & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \lambda_3 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \lambda_3
\end{bmatrix} \\
\mathsf{J}^{(\lambda_4)} &= \begin{bmatrix}
\lambda_4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \lambda_4 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \lambda_4 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \lambda_4 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \lambda_4 & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \lambda_4 & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \lambda_4
\end{bmatrix},
\end{aligned} \tag{9.57}$$

where the eigenvalues are

$$\{\lambda_1,\, \lambda_2,\, \lambda_3,\, \lambda_4\} = \left\{0,\, 2(\mathrm{D}-5)\,, \frac{\mathrm{D}-4}{2},\, \mathrm{D}-4\right\}. \tag{9.58}$$

Their associated multiplicities

$$\{\Lambda_1,\, \Lambda_2,\, \Lambda_3,\, \Lambda_4\} = \{8,\, 1,\, 7,\, 7\} \tag{9.59}$$

count the sizes of the four different Pfaffian systems that will be obtained via logarithmic restriction. The Jordan blocks above specify the maximum logarithm powers

$$\{M_{\lambda_1},\, M_{\lambda_2},\, M_{\lambda_3},\, M_{\lambda_3}\} = \{0,\, 0,\, 1,\, 2\} \tag{9.60}$$

for the series ansatz (8.44). The vector of MIs $\vec{I}$ can therefore be represented as

$$\begin{aligned} G_{321}\cdot\vec{I}(z) \quad &= \quad z_1^{\lambda_1}\vec{I}^{\,(\lambda_1,0)}(z) \quad + \quad z_1^{\lambda_2}\vec{I}^{\,(\lambda_2,0)}(z) \\ &+ \quad z_1^{\lambda_3}\big[\vec{I}^{\,(\lambda_3,0)}(z) \quad + \quad \vec{I}^{\,(\lambda_3,1)}(z)\log(z_1)\big] \\ &+ \quad z_1^{\lambda_4}\big[\vec{I}^{\,(\lambda_4,0)}(z) \quad + \quad \vec{I}^{\,(\lambda_4,1)}(z)\log(z_1) \quad + \quad \vec{I}^{\,(\lambda_4,2)}(z)\log^2(z_1)\big]\,, \end{aligned} \tag{9.61}$$

where $z=(z_1,z_2)$. The logarithm-free terms above all have holomorphic expansions in z_1 of the form

$$\vec{I}^{\,(\lambda,0)}(z) = \sum_{n=0}^{\infty}\vec{I}^{\,(\lambda,n,0)}(z_2)\, z_1^n\,. \tag{9.62}$$

Recall that all higher-order terms $\vec{I}^{\,(\lambda,n,m)}$ in $z_1^n\log^m(z_1)$ are uniquely specified given $\vec{I}^{\,(\lambda,0,0)}$ as input via the recursion relations (A), (B) and (C).

9.3.2 Restriction

We proceed to compute the $\Lambda\times\Lambda$ Pfaffian matrices $Q_2^{(\lambda)}(z_2)$ associated to each $\vec{I}^{\,(\lambda,0,0)}$. The full expression (9.61) has not yet been mounted because the boundary constants are missing, so this step is left for future study.

Eigenvalue λ_1

The R-matrix for $\lambda_1=0$ splits into a 0-block and an identity matrix block:

$$\mathsf{R}^{(\lambda_1)} = \texttt{RowReduce}[P_{1,-1}] = \left[\begin{array}{c|c} \mathbf{0}_{15\times 8} & \mathbf{1}_{15} \end{array}\right]. \tag{9.63}$$

Since $\mathsf{R}^{(\lambda_1)}\cdot\vec{I}^{\,(\lambda_1,0,0)}=0$, this means that $I_9^{(\lambda_1,0,0)}=\ldots=I_{23}^{(\lambda_1,0,0)}=0$. The 8 remaining entries of $\vec{I}^{\,(\lambda_1,0,0)}$ thus make up a basis for the restricted Pfaffian system:

$$\vec{J}^{\,(\lambda_1)} = \begin{bmatrix} I_1^{(\lambda_1,0,0)} \\ \vdots \\ I_8^{(\lambda_1,0,0)} \end{bmatrix}. \tag{9.64}$$

This corresponds to the B-matrix

$$\mathsf{B}^{(\lambda_1)} = \left[\begin{array}{c|c} \mathbf{I}_8 & \mathbf{0}_{8\times 15} \end{array}\right] \quad \text{such that} \quad \vec{J}^{(\lambda_1)} = \mathsf{B}^{(\lambda_1)} \cdot \vec{I}^{(\lambda_1,0,0)}\,, \tag{9.65}$$

where $\mathbf{I}_8$ represents the "reversed identity matrix", i.e. the 8×8 matrix with 1's on the anti-diagonal. Setting

$$\mathsf{M}^{(\lambda_1)} = \left[\begin{array}{c} \mathsf{B}^{(\lambda_1)} \\ \hline \mathsf{R}^{(\lambda_1)} \end{array}\right], \tag{9.66}$$

then the Pfaffian matrix associated to $\vec{J}^{(\lambda_1)}$ is

$$\left(\partial_2 \bullet \mathsf{M}^{(\lambda_1)} + \mathsf{M}^{(\lambda_1)} \cdot P_{2,0}\right) \cdot \left(\mathsf{M}^{(\lambda_1)}\right)^{-1} = \left[\begin{array}{c|c} Q_2^{(\lambda_1)} & \star \\ \hline \mathbf{0}_{15\times 8} & \star \end{array}\right]. \tag{9.67}$$

We find that

$$\partial_2 \bullet \vec{J}^{(\lambda_1)} = Q_2^{(\lambda_1)} \cdot \vec{J}^{(\lambda_1)} \tag{9.68}$$

is indeed a Pfaffian system for the massless double-box topology, as verified by an independent IBP computation. Each entry of $\vec{J}^{(\lambda_1)}$ is a linear combination[7] of integrals in the family (9.49) with $m^2 = 0$ at the integrand level. The matrix $Q_2^{(\lambda_1)}$ has the form $\frac{Q_a(\epsilon)}{z_1^2} + \frac{Q_b(\epsilon)}{z_1} + \frac{Q_c(\epsilon)}{z_1+1}$ where $Q_{a,b,c}(\epsilon)$ contain rational functions in ϵ with maximal degree ϵ^2. Using the software CANONICA [172], a canonical form for $Q_2^{(\lambda_1)}$ is found within seconds.

Eigenvalue λ_2

The system for the eigenvalue $\lambda_2 = 2(\mathrm{D} - 5) = -2 - 4\epsilon$ is peculiar. The rank jump constraint

$$\mathsf{R}^{(\lambda_2)} \cdot \vec{I}^{(\lambda_2,0,0)} = \texttt{RowReduce}[P_{1,-1} - \lambda_2 \mathbf{1}] \cdot \vec{I}^{(\lambda_2,0,0)} = 0 \tag{9.69}$$

sets every entry of $\vec{I}^{(\lambda_2,0,0)}$ to zero except for the 15th entry $J^{(\lambda_2)} = I_{15}^{(\lambda_2,0,0)}$. The Pfaffian system for this single, non-trivial function works out to

$$\partial_2 \bullet J^{(\lambda_2)} = -\frac{J^{(\lambda_2)}}{z_2} \quad \Longrightarrow \quad J^{(\lambda_2)} = \frac{c^{(\lambda_2)}(\epsilon)}{z_2}\,. \tag{9.70}$$

[7] It is a linear combination because the gauge transformation matrix G_3 from (9.54) is non-diagonal.

Eigenvalue λ_3

The system associated to eigenvalue $\lambda_3 = \frac{\mathrm{D}-4}{2} = -\epsilon$ has rank $\Lambda_3 = 7$. Working out the R- and B-matrices as usual, one finds a basis

$$\vec{J}^{(l_3)} = \begin{bmatrix} I_{16}^{(\lambda_3,0,0)} \\ \vdots \\ I_{22}^{(\lambda_3,0,0)} \end{bmatrix} . \tag{9.71}$$

The associated Pfaffian matrix $Q_2^{(\lambda_3)}$ has the form $\frac{Q_a(\epsilon)}{z_2^2} + \frac{Q_b(\epsilon)}{z_2} + \frac{Q_c(\epsilon)}{(z_2+1)^2} + \frac{Q_d(\epsilon)}{z_2-1}$, where $Q_{a,b,c,d}(\epsilon)$ contain rational functions in ϵ of maximal degree ϵ^3. This system is swiftly ϵ-factorized with canonica.

Eigenvalue λ_4

The last eigenvalue $\lambda_4 = \mathrm{D} - 4 = -2\epsilon$ also has rank $\Lambda_4 = 7$. Its associated rank jump constraint $\mathsf{R}^{(\lambda_4)} \cdot \vec{I}^{(\lambda_4,0,0)} = 0$ contains linear relations with rather complicated coefficients. This is in contrast to the rank jump constraints for the other eigenvalues, which were all of the form $I_i^{(\lambda,0,0)} = 0$. We do not know why this is the case.

Regardless, we are easily able to calculate the Pfaffian system associated to the basis

$$\vec{J}^{(\lambda_4)} = \begin{bmatrix} I_{5}^{(\lambda_4,0,0)} \\ I_{17}^{(\lambda_4,0,0)} \\ I_{19}^{(\lambda_4,0,0)} \\ I_{20}^{(\lambda_4,0,0)} \\ I_{21}^{(\lambda_4,0,0)} \\ I_{22}^{(\lambda_4,0,0)} \\ I_{23}^{(\lambda_4,0,0)} \end{bmatrix} . \tag{9.72}$$

The Pfaffian matrix has the singularity structure $Q_2^{(\lambda_3)} = \frac{Q_a(\epsilon)}{z_2^2} + \frac{Q_b(\epsilon)}{z_2} + \frac{Q_c(\epsilon)}{(z_2+1)^2} + \frac{Q_d(\epsilon)}{z_2-1}$, where $Q_{a,b,c,d}(\epsilon)$ have maximal degree ϵ^3. Canonica brings this system into canonical form with ease.

9.3.3 Discussion

General solutions to the Pfaffian systems above are easily obtained in terms of HPLs in the variable $z_2 = t/s$. What is more, every step in the construction of those systems can be automatized.

It is worth comparing the present method with previous studies. The seminal calculation carried out in [170] was complicated by an unrationalizable square root. For another 2-loop Bhabha scattering topology [171], the authors had to deal with elliptic functions. Because our restriction protocol involves smaller, univariate Pfaffian systems, such complications are more easily avoided. The cost is that a smaller part of phase space is covered, since m^2 is assumed to be small (it could e.g. be the electron mass $m_e \sim 0.511$ MeV).

As mentioned, the final expression for the logarithmic series (9.61) still lacks the determination of boundary constants. These could be fixed by matching with the full set of two-variable MIs $\vec{I}$ from (9.50) after imposing physically motivated constraints. For instance, one could impose real-valuedness in the Euclidean region, or regularity at certain points in phase space. Another way is just to numerically evaluate the two-variable integrals $\vec{I}$ and match this with the logarithmic series. Both of these approaches are, however, unsatisfactory from the point of view of the restriction protocol. It would be preferable to somehow fix the boundary constants solely within each univariate system $\partial_2 \bullet \vec{J}^{(\lambda)} = Q_2^{(\lambda)} \cdot \vec{J}^{(\lambda)}$, rather than comparing with the two-variate integrals. This is indeed possible for the eigenvalue $\lambda = 0$, since in this case we know the momentum space representation of $\vec{J}^{(\lambda)}$. A speculative suggestion is to perform an *additional* restriction on these univariate systems, in which case the rank jump constraints might provide relations among boundary constants.

Section 9.4

Macaulay-level restriction: 2-loop diagonal box

This example concerns the rational restriction of the GKZ system associated to a 2-loop massless diagonal-box diagram. We employ Algorithm (5), which is based on the Macaulay matrix.

9.4.1 Setup

The LP monomials are given by

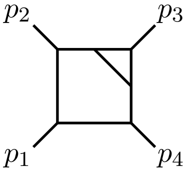
$$\begin{aligned}\mathcal{G}(z|x) = {}& z_1x_1x_4 + z_2x_1x_5 + z_3x_1x_6 + z_4x_2x_4 + z_5x_2x_5 + z_6x_2x_6 + \\ & z_7x_3x_4 + z_8x_3x_5 + z_9x_3x_6 + z_{10}x_4x_5 + z_{11}x_4x_6 + z_{12}x_4x_7 + \\ & z_{13}x_5x_7 + z_{14}x_6x_7 + z_{15}x_1x_3x_4 + z_{16}x_1x_3x_5 + z_{17}x_1x_3x_6 + \\ & z_{18}x_1x_4x_5 + z_{19}x_2x_4x_6 + z_{20}x_2x_4x_7 + z_{21}x_2x_5x_7 + z_{22}x_2x_6x_7\,.\end{aligned} \tag{9.73}$$

Recall that this enters the integrand of the GFI (7.7). $\mathcal{G}(z|x)$ gives rise to to a GKZ system with A-matrix

$$A = \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&1&1&1&\cdot&\cdot&\cdot&\cdot \\ \cdot&\cdot&\cdot&1&1&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&1&1&1 \\ \cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&1&1&\cdot&\cdot&\cdot&\cdot&\cdot&1&1&1&\cdot&\cdot&\cdot&\cdot&\cdot \\ 1&\cdot&\cdot&1&\cdot&\cdot&1&\cdot&\cdot&1&1&1&\cdot&\cdot&1&\cdot&\cdot&1&1&1&\cdot&\cdot \\ \cdot&1&\cdot&\cdot&1&\cdot&\cdot&1&\cdot&1&\cdot&\cdot&1&\cdot&\cdot&1&\cdot&1&\cdot&\cdot&1&\cdot \\ \cdot&\cdot&1&\cdot&\cdot&1&\cdot&\cdot&1&\cdot&1&\cdot&\cdot&1&\cdot&\cdot&1&\cdot&1&\cdot&\cdot&1 \\ \cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1&1&1&\cdot&\cdot&\cdot&\cdot&\cdot&1&1&1 \end{bmatrix}.$$

The restriction we compute is

$$A \xrightarrow[1\leq i\leq 18 \text{ and } 19\leq j\leq 22]{z_i=1 \text{ and } z_j=t/s}$$



, (9.74)

where s and t stand for Mandelstam variables. To be clear, after a linear shift of coordinates, this is a restriction onto the affine hyperplane

$$Y' = \left\{(z_1,\ldots,z_{22}) \in \mathbb{A}^{22} \,\middle|\, z_1 = \ldots = z_{18} = 0 \text{ and } z_{19} = z_{20} = z_{21} = z_{22} = z \text{ is generic}\right\}.$$

Using homogeneity (cf. Section (6.5)) we can immediately restrict $z_i = 0$ for $i \in \{1,2,5,7,9,11,13,18\}$. This means that Algorithm (5) should be applied to restrict $z_i = 0$ for $i \in \{3,4,6,8,10,12,14,15,16,17\}$ and $z_i = z = t/s$ for $i \in \{19,20,21,22\}$.

9.4.2 Restriction

For generic values of the analytic regulators ϵ and δ, the GKZ system is of rank $R = 115$. Computing the Euler characteristic by counting the number of critical points of (6.36) on Y', we find a rank drop to $R' = 7$. This number is verified by counting the number of MIs using IBPs.

We must find a basis of the restriction module before calling Algorithm (5). Executing [83, Algorithm 4.7] as in Example (8.1) with parameters $\gamma = 3$, $k = 3$, $p = 100000007$ and $z = 1/7$, we obtain the 7-dimensional basis[8]

$$\text{RStd} = \begin{bmatrix} \partial_{21}\partial_{22} \\ \partial_{22}^2 \\ \partial_6 \\ \partial_{17} \\ \partial_{21} \\ \partial_{22} \\ 1 \end{bmatrix} . \tag{9.75}$$

A valid Macaulay matrix $M_1 = \left[\begin{array}{c|c} M_{\text{Ext}} & M_{\text{RStd}} \end{array}\right]$ is then found[9] for degree $Q = 1$, meaning that the equation (5.13) given by $C_{\text{Ext}} - C \cdot M_{\text{Ext}} = 0$ has a solution for the matrix C. The matrix M_{Ext} has dimension 1305×2763, so there are 2763 exterior monomials Ext. The 1305 rows are not all independent; by row reducing M_{Ext} with (ϵ, δ, z) fixed to generic numbers, we identify 912 linearly independent equations. It then takes less than 3 minutes on a laptop to determine C via FiniteFlow. The 7×7 Pfaffian matrix P_z finally results from $P_z = C_{\text{RStd}} - C \cdot M_{\text{RStd}}$. This becomes a Pfaffian matrix for the proper 2-loop diagonal-box topology after multiplying by the LP prefactors from (7.7) and sending $\delta \to 0$.

As an aside, note how large the discrepancy is between the ranks $R = 115$ and $R' = 7$ for the GKZ system and the proper FI. So even though we benefit from being able to immediately write down an annihilating ideal and a basis in the GKZ setting, the additional complexity brought about by unspecified z-variables may quickly become too much to handle.

[8] This took 40.56s on a machine with the following specs: `Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz, 8G memory`.

[9] It took 65.76s on the same machine to compute this Macaulay matrix.

Part III

Numerical Integration

Chapter 10

Tropical Integration

This thesis has, so far, predominantly been an *analytic* study of Euler and Feynman integrals. In particular, we have focused on the DEQs obeyed by said integrals. The ideal situation is that 1) these DEQs can be built with reasonable computer resources, and 2) they can be solved symbolically in terms of special functions. For sufficiently difficult problems - e.g. the evaluation of 2- or 3-loop diagrams with several kinematic scales - either or both of these steps become insurmountable with the presently available analytical methods. These difficulties are in many cases unavoidable if the accuracy of theoretical predictions is made to compete with the experimental precision at the LHC. For instance, the slow convergence rate of the QCD perturbative series (stemming from the coupling constant being $\alpha_s(M_Z) \sim 0.1$ near the weak scale) often requires 3-loop results to get cross sections at %-level accuracy. Furthermore, the study of Higgs phenomenology typically requires the calculation of diagrams with many internal masses that do not evaluate to GPLs, which complicates the determination of canonical bases in the method of DEQs for FIs.

In cases where analytical methods fall short, *numerical* methods may instead offer a way forward. The past few years have indeed seen a surge in the development of public codes tailored to the numerical evaluation of FIs. Using series of expansions of Pfaffian systems, the codes AMFlow [34], DiffExp [173, 174] and SeaSyde [175] are able to to achieve phenomenal accuracy. They derive Pfaffian systems by fixing parameters to numbers *before* solving the IBPs, which yields a significant speed boost. The pySecDec codebase [176], employing *Monte Carlo* integration in combination with *sector decomposition*, has been a staple of high-energy physics calculations for many years, and was recently upgraded in [177]. Sophisticated manipulations of momentum space integrals via the *Loop-tree duality* led to efficient and stable numerical integration schemes in [178, 179]. The authors of [180] presented a high-precision numerical algorithm by interpreting FIs as *period integrals.* Further promising studies were carried out in

[181, 182, 183], though so far without publicly available codes.

Part III of this thesis is a contribution to this flourishing research program. The present chapter describes a Monte Carlo integration scheme called *tropical integration*. It was established by Borinsky in [40] (see also [184]), and subsequently generalized by Borinsky, Tellander and the present author in [185]. A public code named Feyntrop was developed alongside the aforementioned article. Feyntrop is able to swiftly numerically evaluate FIs at high loop-order with many mass scales at ‰-level accuracy given modest computer resources. The next chapter gives a tutorial on how to use the code, and showcases several state-of-the-art examples. The present chapter explicates the theoretical background.

Section 10.1

Tropical Monte Carlo sampling, in brief

Tropical integration is a special instance of Monte Carlo (MC) integration. MC methods deal with integrals of the form

$$I_f = \int_\Gamma f(x)\mu\,, \tag{10.1}$$

where $f(x)$ is a square-integrable function integrated over an n-dimensional domain Γ, and μ is a probability measure that is positive on Γ and normalized to unity, $\int_\Gamma \mu = 1$. By sampling many independent random points $\{x^{(1)}, \ldots, x^{(N)}\}$ from μ, it follows from the central limit theorem that

$$I_f \simeq I_f^{(N)} \quad \text{where} \quad I_f^{(N)} = \frac{1}{N}\sum_{i=1}^{N} f\big(x^{(i)}\big)\,. \tag{10.2}$$

Here $I_f^{(N)}$ is simply the average of the integrand f evaluated at the random points $x^{(i)}$. For sufficiently large N, the expected error of this approximation is

$$\sigma_f = \sqrt{\frac{I_{f^2} - I_f^2}{N}} \quad \text{where} \quad I_{f^2} = \int_\Gamma f(x)^2\mu\,. \tag{10.3}$$

This error can itself be estimated by

$$\sigma_f \simeq \sigma_f^{(N)} \quad \text{where} \quad \sigma_f^{(N)} = \sqrt{\frac{I_{f^2}^{(N)} - \left[I_f^{(N)}\right]^2}{N-1}} \quad \text{and} \quad I_{f^2}^{(N)} = \frac{1}{N}\sum_{i=1}^{N} f\big(x^{(i)}\big)^2\,. \tag{10.4}$$

A common choice for μ is just the uniform measure $\mu = \mathrm{d}^n x$ on the n-dimensional unit cube $\Gamma = [0,1]^n$. Tropical integration, instead, makes use of polytopal and tropical geometry (see [35, 186] for seminal textbooks on these topics) to build a more sophisticated *tropical measure* μ^{tr}. This measure is tailored to estimating integrals of functions $f(x) = \frac{g(x)^a}{h(x)^b}$ given by homogeneous polynomials $g, h \in \mathbb{C}[x]$ and rational numbers $a, b \in \mathbb{Q}$. A good example hereof is the evaluation of the integral $\int_0^1 \int_0^1 \frac{\mathrm{d}x \wedge \mathrm{d}y}{x+y} = 2\log 2$. "Ordinary" MC fails because the variance of $\frac{1}{x+y}$ is infinite w.r.t. the uniform measure. It is, however, easily evaluated with the tropical measure. See [187, Section 3.1.1] for a detailed discussion.

10.1.1 Tropical approximation

As we are ultimately interested in integrating homogeneous polynomials, let us now define the notion of *real projective space* $\mathbb{RP}^{n-1}$; this which will serve as an integration domain[1] later on. $\mathbb{RP}^{n-1}$ is defined by the set of points in $\mathbb{R}^n \setminus \{0\}$ modulo the equivalence relation

$$\begin{array}{rcl} (x_0, \dots, x_{n-1}) & \sim & (y_0, \dots, y_{n-1}) \\ & \Updownarrow & \\ \exists \lambda \in \mathbb{R} \setminus \{0\} \, : \, (x_0, \dots, x_{n-1}) & = & (\lambda y_0, \dots, \lambda y_{n-1}) \, . \end{array} \tag{10.5}$$

We denote the coordinates on $\mathbb{RP}^{n-1}$ by $[x_0 : \ldots : x_{n-1}]$. Two points such as $[1:0:2]$ and $[\frac{1}{2}:0:1]$ would be equivalent in $\mathbb{RP}^2$ because they are related by $(1,\ 0,\ 2) = (2\cdot\frac{1}{2},\ 2\cdot 0,\ 2\cdot 1)$. *Positive* projective space is further given by

$$\mathbb{P}_+^{n-1} = \{x = [x_0 : \cdots : x_{n-1}] \in \mathbb{RP}^{n-1} \,|\, x_i > 0\} \, . \tag{10.6}$$

Above we made the unusual choice of starting the index at 0 rather than 1. This convention allows for seamless interoperability with the PYTHON programming language, to be used in the next chapter.

Let p be a homogeneous polynomial. In multi-index notation,

$$p(x) = \sum_{k \in \mathrm{supp}(p)} c_k \, x^k \ , \tag{10.7}$$

where $x^k = x_0^{k_0} \cdots x_{n-1}^{k_{n-1}}$ and $|k| = k_0 + \ldots + k_{n-1}$ is equal for all multi-indices k. The *tropical approximation* [39] to p is the central concept in tropical integration:

$$p^{\mathrm{tr}}(x) = \max_{k \in \mathrm{supp}(p)} x^k \, . \tag{10.8}$$

[1] In practice, this domain can be parametrized as a simplex polytope.

If, for instance, $p(x) = x_0 x_1 x_2^2 - 2x_1^4 + 42i x_0^2 x_1^2$, then $p^{\mathrm{tr}}(x) = \max\left\{x_0 x_1 x_2^2,\, x_1^4,\, x_0^2 x_1^2\right\}$. In other words, the tropical approximation neglects the monomial coefficients, and only cares about the largest monomial. The set of monomial exponents $\operatorname{supp}(p)$ is hence the essential data for p^{tr}. Interpreting each multi-index in $\operatorname{supp}(p)$ as a vector in $\mathbb{Z}^n_{\geq 0}$ and taking the convex hull, we see that p^{tr} is simply a function avatar of the *Newton polytope* $\mathbf{N}[p]$ from (6.51).

Suppose that $\mathbf{N}[p]$ lives in $\mathbb{R}^n$ as parametrized by x. By virtue of being a polytope, $\mathbf{N}[p]$ is given by a finite intersection of *half-spaces*, each of the form $\{v \in \mathbb{R}^n \,|\, v \cdot x \leq c\}$ for some scalar $c \in \mathbb{R}$. A subset $F \subset \mathbf{N}[p]$ is called a *face* of $\mathbf{N}[p]$ iff

1. There exist $v \in \mathbb{R}^n$ and $c \in \mathbb{R}$ such that $\mathbf{N}[p]$ is contained in $\{v \in \mathbb{R}^n \,|\, v \cdot x \leq c\}$.
2. F is the intersection of $\mathbf{N}[p]$ with $\{v \in \mathbb{R}^n \,|\, v \cdot x = c\}$.

The polynomial p truncated to a face F is given by

$$p_F(x) = \sum_{k \,\in\, F \,\cap\, \operatorname{supp}(p)} c_k\, x^k\,. \tag{10.9}$$

p is called *completely non-vanishing* on a domain X if for each face F of the Newton polytope we have $p_F(x) \neq 0$, $x \in X$.

Example 10.1. Consider the polynomial

$$p(x) = x_0 + x_1 + 2x_0^2 + x_1^2 + 3x_0 x_1 + 2x_0^2 x_1 + x_0 x_1^2 + x_0^3\,. \tag{10.10}$$

It is non-homogeneous, but that is inconsequential for the purpose of this example. Its associated Newton polytope $\mathbf{N}[p]$ is shown in Figure (10.1). As an intersection of half-spaces, the Newton polytope is represented by

$$\mathbf{N}[p] = \left\{x_0 \geq 0\right\} \cap \left\{x_1 \geq 0\right\} \cap \left\{x_1 \leq 2\right\} \cap \left\{x_0 + x_1 \geq 1\right\} \cap \left\{x_0 + x_1 \leq 3\right\}. \tag{10.11}$$

Consider a face F of $\mathbf{N}[p]$, for example the bottom line segment touching the x_0-axis given by $F = \{(x_0, x_1) \,|\, 1 \leq x_0 \leq 3 \text{ and } x_1 = 0\}$. F is indeed a face since 1) $\mathbf{N}[p]$ is contained in the half-space $\{x_1 \geq 0\}$, and 2) F is given by an intersection $F = \mathbf{N}[p] \cap \{x_1 = 0\}$. The polynomial truncated to this face is given by

$$p_F(x) = x_0 + 2x_0^2 + x_0^3\,. \tag{10.12}$$

Because each monomial coefficient of p is positive, we have that p is completely non-vanishing on, say, $X = \mathbb{R}^2_{>0}$. ■

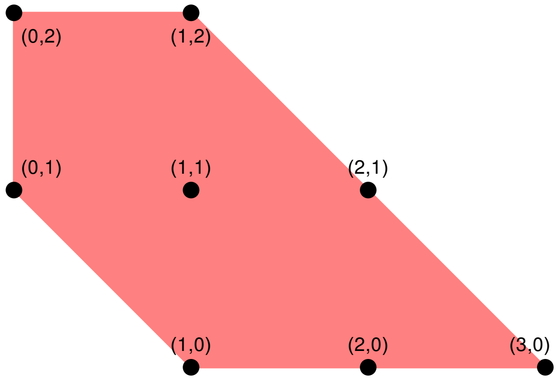


Figure 10.1: Newton polytope for the polynomial $p(x)$ in (10.10). The black dots represent the monomial exponents of $p(x)$, and the pink area is their convex hull in $\mathbb{R}^2$ as parametrized by (x_0, x_1).

Borinsky proved that there is really a sense in which $p^{\mathrm{tr}}(x)$ *approximates* the values of $p(x)$; this is the backbone behind tropical integration. Take a homogeneous polynomial $p \in \mathbb{C}[x]$ that is completely non-vanishing on positive projective space $\mathbb{P}_+^{n-1}$. Then there exist two constants $C_1, C_2 > 0$ such that [40, Theorem 8]

$$C_1 \leq \frac{|p(x)|}{p^{\mathrm{tr}}(x)} \leq C_2 \quad \text{for all} \quad x \in \mathbb{P}_+^{n-1}. \tag{10.13}$$

In other words, the tropical approximation of p bounds the original polynomial from above and from below.

We can now describe the intuition behind tropical integration (a precise algorithm is given in Section (10.3.2)). Consider a convergent integral

$$I = \int_{\mathbb{P}_+^{n-1}} \frac{g(x)^a}{h(x)^b}\,\Omega \tag{10.14}$$

given by two homogeneous polynomials g and h that are completely non-vanishing on the integration domain, and $a, b \in \mathbb{Q}$. We insert a clever way of writing the number "1" into the integrand:

$$\begin{aligned} I &= \int_{\mathbb{P}_+^{n-1}} \frac{g(x)^a}{h(x)^b}\,\Omega \quad \times \quad \left(\frac{g^{\mathrm{tr}}(x)}{g^{\mathrm{tr}}(x)}\right)^a \left(\frac{h^{\mathrm{tr}}(x)}{h^{\mathrm{tr}}(x)}\right)^b \\ &= \underbrace{\int_{\mathbb{P}_+^{n-1}} \frac{g^{\mathrm{tr}}(x)^a}{h^{\mathrm{tr}}(x)^b}\,\Omega}_{\propto \text{ tropical measure } \mu^{\mathrm{tr}}} \quad \times \quad \underbrace{\left(\frac{g(x)}{g^{\mathrm{tr}}(x)}\right)^a \left(\frac{h^{\mathrm{tr}}(x)}{h(x)}\right)^b}_{\text{bounded due to (10.13)}}. \end{aligned} \tag{10.15}$$

The integrand has been factored into something resembling a probability measure (after being suitably normalized) multiplied by a function that is square-integrable as a consequence of being bounded. This is now a suitable setup for MC integration: we sample a large number of random points $x^{(i)}$ from μ^{tr}, and then take the average of the function $(g/g^{\mathrm{tr}})^a\,(h^{\mathrm{tr}}/h)^b$ evaluated at these points.

Section 10.2

Projective Feynman integrals

As we will present a numerical code later on, let us be explicit about our conventions for FIs. Take G to be a one-particle irreducible Feynman diagram with edge set E and vertex set V. With our index convention, we have $E = \{0, 1, \ldots, |E| - 1\}$ and $V = \{0, 1, \ldots, |V| - 1\}$. Every edge $e \in E$ is equipped with a mass m_e and an edge weight ν_e. Every vertex $v \in V$ comes with an incoming momentum vector p_v. Internal vertices are distinguished by having vanishing incoming momenta; i.e., writing $V = V_{\mathrm{ext}} \sqcup V_{\mathrm{int}}$, then $p_v = 0$ if $v \in V_{\mathrm{int}}$.

The *incidence matrix* $\mathcal{E}$ of G is defined by choosing an arbitrary orientation for the edges, and setting $\mathcal{E}_{v,e} = \pm 1$ if edge e points to/from vertex v and $\mathcal{E}_{v,e} = 0$ if e is not incident to v.

In terms of this data, we write the FI associated to G as

$$I_G = \int \prod_{e\in E} \frac{\mathrm{d}^{\mathrm{D}}\ell_e}{i\pi^{\mathrm{D}/2}} \left(\frac{-1}{\ell_e^2 - m_e^2 + i\varepsilon}\right)^{\nu_e} \prod_{v\in V\setminus\{v_0\}} i\pi^{\mathrm{D}/2}\delta^{(\mathrm{D})}\Big(p_v + \sum_{e\in E}\mathcal{E}_{v,e}\,\ell_e\Big)\,. \tag{10.16}$$

The loop momenta ℓ_e are integrated over D-dimensional Minkowski space with mostly-minus signature, $\ell_e^2 = \left(\ell_e^0\right)^2 - \left(\ell_e^1\right)^2 - \left(\ell_e^2\right)^2 - \ldots$, and an overall momentum-conserving δ-function $\delta(p_{v_0} + \ldots + p_{v_{|V|-1}})$ has been extracted by removing a vertex $v_0 \in V$.

To evaluate I_G numerically, we use an equivalent parametric representation (see [44, Section 2.5] for a derivation starting from momentum space):

$$I_G = \Gamma(\omega)\int_{\mathbb{P}_+^{|E|-1}} \phi \qquad \text{with} \tag{10.17}$$

$$\phi = \left(\prod_{e\in E}\frac{x_e^{\nu_e}}{\Gamma(\nu_e)}\right)\frac{1}{\mathcal{U}(x)^{\mathrm{D}/2}}\left(\frac{1}{\mathcal{V}(x) - i\varepsilon\sum_{e\in E}x_e}\right)^{\omega}\Omega\,.$$

This is an integral over positive projective space $\mathbb{P}_+^{|E|-1}$, defined in (10.6), w.r.t. the volume form

$$\Omega = \sum_{e=0}^{|E|-1} (-1)^{|E|-e-1} \frac{\mathrm{d}x_0 \wedge \cdots \wedge \widehat{\mathrm{d}x_e} \wedge \cdots \wedge \mathrm{d}x_{|E|-1}}{x_1 \cdots \widehat{x}_e \cdots x_{|E|-1}}\,, \tag{10.18}$$

where hats denote omission. The superficial degree of divergence of the graph G is given by $\omega = \sum_{e\in E} \nu_e - \mathrm{D}\cdot L/2$ where $L = |E| - |V| + 1$ is the number of loops.

$\mathcal{U}(x)$ is the first Symanzik polynomial, previously defined in (7.1). Recalling also the second Symanzik polynomial $\mathcal{F}(x)$ from (7.2), we set

$$\mathcal{V}(x) = \frac{\mathcal{F}(x)}{\mathcal{U}(x)}\,. \tag{10.19}$$

Since $\mathcal{U}(\lambda x) = \lambda^L \mathcal{U}(x)$ and $\mathcal{F}(\lambda x) = \lambda^{L+1}\mathcal{F}(x)$, one can quickly verify that the integrand ϕ is invariant under the coordinate rescaling $x \to \lambda x$, wherefore it is appropriate to call (10.17) a *projective integral representation.*

The graph-theoretic formulas (7.1) and (7.2) for $\mathcal{U}$ and $\mathcal{F}$ are in fact not so efficient for numerically evaluating millions of points during MC integration. The reason is that the number of monomials grows quickly as a function of $|E|$ and L, making it slow to add up all terms during each numerical evaluation. Let us therefore present more efficient formulas in terms of matrices that scale linearly with the complexity of G. To start, define the $(|V|-1)\times(|V|-1)$ *reduced Laplacian matrix* component-wise by

$$\mathcal{L}_{uv} = \sum_{e\in E} \frac{\mathcal{E}_{u,e}\,\mathcal{E}_{v,e}}{x_e} \quad \text{for} \quad u,v \in V\setminus\{v_0\}\,. \tag{10.20}$$

As this matrix is symmetric, it is only necessary to explicitly calculate the diagonal and the upper triangular block. The lower triangular block can then be copied from the upper one. It turns out that $\mathcal{L}$ is positive definite.

Let $\mathcal{P}_{uv} = p_u \cdot p_v$ be the $(|V|-1)\times(|V|-1)$ matrix containing Minkowski scalar products between all independent momenta.

We then have the identities

$$\mathcal{U}(x) = \det\left[\mathcal{L}(x)\right] \prod_{e\in E} x_e \tag{10.21}$$

$$\mathcal{F}(x) = \mathcal{U}(x)\left[-\sum_{u,v\,\in\, V\setminus\{v_0\}} \left(\mathcal{L}^{-1}(x)\right)_{uv} \mathcal{P}_{uv} + \sum_{e\in E} m_e^2\, x_e\right]. \tag{10.22}$$

Note that the first sum in $\mathcal{F}$ can be expressed as tr $\left[\mathcal{L}^{-1}\cdot\mathcal{P}\right]$. Determinants and traces are computationally cheap, but the matrix inverse $\mathcal{L}^{-1}$ appears to be expensive at first glance. However, notice that we are only interested in the combination $\mathsf{X}=\mathcal{L}^{-1}\cdot\mathcal{P}$, not $\mathcal{L}^{-1}$ itself. Owing to the fact that $\mathcal{L}$ is positive definite, then X can be swiftly evaluated using *Cholesky decomposition*. More precisely, in $O(|V|^3)$ time, it is possible to decompose $\mathcal{L}=\mathsf{L}\cdot\mathsf{D}\cdot\mathsf{L}^T$ in terms of a lower triangular matrix L and a diagonal matrix D. Afterwards, the solution X to the linear system $\mathcal{L}\cdot\mathsf{X}=\mathcal{P}$ is immediately obtained by solving

$$\mathsf{L}\cdot\mathsf{Y}_1=\mathcal{P}\,,\quad \mathsf{D}\cdot\mathsf{Y}_2=\mathsf{Y}_1\,,\quad \mathsf{L}^T\cdot\mathsf{X}=\mathsf{Y}_2 \tag{10.23}$$

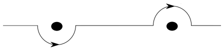
in cascade.

Later on, we will actually require $\mathcal{L}$ to take on complex values rather than real ones. Then $\mathcal{L}$ is no longer Hermitian, wherefore Cholesky decomposition fails. In this case, one must use the less efficient LU (lower-upper) decomposition.

10.2.1 $i\varepsilon$-prescription via contour deformation

For the purposes of this thesis, say that a FI computation is in the *Euclidean regime* if $\mathcal{F}(x)\geq 0$ for all $x\in\mathbb{P}_+^{|E|-1}$ (a more refined notion of the Euclidean regime is given in [185, Section 2.2]). Then it is only the case that $\mathcal{F}(x)=0$ on the boundary $x\to 0$ of the integration domain, which is assumed to be an integrable singularity. In our convention, being in the Euclidean regime is analogous to having a FI with scalar products computed in the Euclidean all-minus metric. Life is simple in the Euclidean regime because one avoids the singularities in (10.16) coming from setting the time-components equal to $\ell^0=\pm\sqrt{\vec{\ell}\cdot\vec{\ell}+m^2}$ in the propagators. Consequently, Feynman's $i\varepsilon$-prescription can be dropped, and the resulting FI will be purely real.

Alas, life is not so simple in Minkowski space. Let us say that we are in the *Minkowski regime* when we are not in the Euclidean regime. In this case, the monomial coefficients of $\mathcal{F}(x)$ can be both positive and negative. The $i\varepsilon$ regulates singularities coming from having $\mathcal{F}(x)=0$ inside the integration domain. In momentum space, the $i\varepsilon$ regulates the ℓ^0-singularities by pushing them slightly away from the real axis. An alternative description is to deform the integration contour given by $\ell^0\in(-\infty,\infty)$ to one that goes around the poles[2].

[2] With two poles there are four possible choices for how to go around them, but only one is consistent with the notion of causality. This contour corresponds to a propagator $\ell^2-m^2+i\varepsilon$. Two non-casual examples are $(\ell^0+i\varepsilon)^2-\vec{\ell}\cdot\vec{\ell}-m^2$ and $\ell^2-m^2-i\varepsilon$. The correct contour goes below the singularity at $-\sqrt{\vec{\ell}\cdot\vec{\ell}+m^2}$ and above the one at $+\sqrt{\vec{\ell}\cdot\vec{\ell}+m^2}$. I.e.  in the complex ℓ^0-plane.

The $i\varepsilon$-prescription only dictates on which side of the poles we integrate. But for the purpose of numerical integration we must make an *explicit* choice for the contour, preferably with decent numerical stability. Moreover, the deformation ought to preserve the projective symmetry of the integral in order to play well with tropical MC sampling. We find that the following deformation into the complex plane proposed in [188, 189] fulfils both of these criteria:

$$\boxed{X_e(x) = x_e \exp\left[-i\lambda \frac{\partial \mathcal{V}(x)}{\partial x_e}\right] \quad \text{for} \quad \lambda \in \mathbb{R}_{>0}\,.} \tag{10.24}$$

Because $\mathcal{U}$ and $\mathcal{F}$ are homogeneous polynomials of degrees L and $L+1$ respectively, then $\mathcal{V} = \mathcal{F}/\mathcal{U}$ is a rational function of degree 1, wherefore $\partial_e \mathcal{V} = \partial \mathcal{V}(x)/\partial x_e$ has homogeneous degree 0. Thus, (10.24) preserves projective symmetry.

The value of parameter λ is finite but small. This is in contrast to the infinitesimal ε of the $i\varepsilon$-prescription. As long as λ is is small enough that no poles of the integrand ϕ are hit, then Cauchy's theorem guarantees that the deformation is valid. Note that λ must have a mass dimension in order to cancel that of $\partial_e \mathcal{V}$ in the exponential. In practice, we shall choose λ to be of order $\mathcal{O}(1/\Lambda^2)$, where Λ^2 is the largest kinematic scale in $\mathcal{V}$.

To see why (10.24) is consistent with the $i\varepsilon$-prescription, let us consider the *Schwinger representation* of a FI [44, Section 2.5.2]:

$$I = \int_0^\infty \ldots \exp\left[i\left(-\mathcal{V}(x) + i\varepsilon \sum_{e\in E} x_e\right)\right]. \tag{10.25}$$

Here we ignored factors in the integrand which are irrelevant for the present argument. Observe that the $i\varepsilon$ term cures the divergence at $x_e \to \infty$ by gifting an exponential damping $\sim \exp\left[-\varepsilon x_e\right]$. Notice that a similar damping is achieved by adding a *positive* imaginary part to the first term $-\mathcal{V}(x)$. In that case the $i\varepsilon$ could be dropped. Indeed, changing coordinates according to (10.24), then a Taylor expansion in λ shows that

$$-\mathcal{V}(X) = -\mathcal{V}(x) + i\lambda \sum_{e\in E} x_e \left[\frac{\partial \mathcal{V}(x)}{\partial x_e}\right]^2 + \mathcal{O}(\lambda^2)\,. \tag{10.26}$$

Assuming that 1) there are no solutions to the *Landau equations*

$$x_e \frac{\partial \mathcal{V}(x)}{\partial x_e} = 0 \quad \text{for each} \quad e \in E \quad \text{and for any} \quad x \in \mathbb{P}_+^{|E|-1}\,, \tag{10.27}$$

and 2) λ is small enough such that the $\mathcal{O}(\lambda^2)$ terms to not change the sign of the $\mathcal{O}(\lambda)$

term, then (10.26) has a positive imaginary part, as desired. For the remaining part of this thesis, it is assumed that the Landau equations have no solutions.

A short calculation shows that the $|E| \times |E|$ Jacobian matrix associated to (10.24) is

$$(\mathcal{J}_\lambda(x))_{eh} = \delta_{eh} - i\lambda x_e \frac{\partial^2 \mathcal{V}(x)}{\partial x_e \partial x_h} \quad \text{for} \quad e, h \in E\,. \tag{10.28}$$

Coming back to the projective FI (10.17), its deformed version now reads

$$I_G = \Gamma(\omega) \int_{\mathbb{P}_+^{|E|-1}} \left(\prod_{e\in E} \frac{X_e^{\nu_e}}{\Gamma(\nu_e)} \right) \frac{\det \mathcal{J}_\lambda(x)}{\mathcal{U}(X)^{\mathrm{D}/2} \; \times \; \mathcal{V}(X)^{\omega}} \, \Omega\,, \tag{10.29}$$

where $X = (X_0, \dots, X_{|E|-1})$ with X_e given by (10.24).

10.2.2 ϵ-expansion for quasi-finite integrals

We have so far not made any assumptions regarding the finiteness of the FI (10.29) as the DR parameter ϵ tends to zero. Recall that ϵ appears inside $\mathrm{D} = \mathrm{D}_0 - 2\epsilon$ and $\omega = \sum_{e\in E} \nu_e - \mathrm{D}L/2$ in this representation.

We shall restrict our attention to the category of integrals dubbed as *quasi-finite* in [190]. This means that I_G in (10.29) can have overall ϵ-poles coming from the prefactor $\Gamma(\omega)$, but the integral $\int_{\mathbb{P}_+^{|E|-1}} (\dots)$ is itself finite as $\epsilon \to 0$. Note that this is less restrictive than requiring the whole FI to be finite.

Given this, the ϵ-expansion of I_G is derived by merely Taylor expanding the integrand. Writing $\omega_0 = \sum_{e\in E} \nu_e - \mathrm{D}_0 L/2$, the expansion becomes

$$\begin{aligned} I_G = \Gamma(\omega_0 + \epsilon L) \sum_{k=0}^{\infty} \frac{\epsilon^k}{k!} \times \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad & \\ \int_{\mathbb{P}_+^{|E|-1}} \left(\prod_{e\in E} \frac{X_e^{\nu_e}}{\Gamma(\nu_e)} \right) \frac{\det \mathcal{J}_\lambda(x)}{\mathcal{U}(X)^{\mathrm{D}_0/2} \; \times \; \mathcal{V}(X)^{\omega_0}} \log^k \left[\frac{\mathcal{U}(X)}{\mathcal{V}(X)^L} \right] \Omega\,. \end{aligned} \tag{10.30}$$

If the integral is finite for $k = 0$, then the $\log^k$ factors cannot spoil the convergence for higher orders in k.

Our ultimate goal is to numerically evaluate the integrals appearing in the ϵ-expansion above. To this end, it is important that the tropical approximation property (10.13) holds true, but this is now complicated by the fact that X depends on λ. So we make the following *assumption*: there exist λ-dependent constants $C_1(\lambda), C_2(\lambda) > 0$ such that for small enough $\lambda > 0$,

$$C_1(\lambda) \;\leq\; \left| \left(\frac{\mathcal{U}^{\mathrm{tr}}(x)}{\mathcal{U}(X)}\right)^{\mathrm{D}_0/2} \left(\frac{\mathcal{V}^{\mathrm{tr}}(x)}{\mathcal{V}(X)}\right)^{\omega_0} \right| \;\leq\; C_2(\lambda) \quad \text{for all} \quad x \in \mathbb{P}_+^{|E|-1}\,. \tag{10.31}$$

We have checked this assumption through extensive numerical testing, but lack a mathematical argument for its validity.

The statement (10.31) suggest an optimal value of λ: choose it such that the bounds are as tight as possible. This is an interesting research direction, as it would constitute the first *canonical* contour deformation without any free parameters.

10.2.3 Convergence of Euler integrals

It was proven by Berkesch, Forsgård and Passare that a quasi-finite representation can always be found [191, Theorem 2.4]. The proof of this theorem gives a concrete prescription for factoring out ϵ-poles in front of finite integrals, but we have not yet implemented this protocol[3].

Assuming a quasi-finite basis of MIs is known, one can alternatively employ a combination of dimensional recurrence relations and IBPs as in [190] to write a given FI as linear combination of said basis.

To find a quasi-finite basis, one ought to have test for finiteness of Euler integrals (in contrast to the GKZ setting studied earlier in this thesis, we are now allowing the monomial coefficients of the Euler integrands to be fixed to special values). Convergence theorems for Euler integrals were first discovered in the mathematics literature by Berkesch, Forsgård, Nilsson and Passare [192, Theorem 1] [191, Theorem 2.2]. In the physics literature, convergence criteria were later established by Panzer, Schabinger and von Manteuffel in [190], followed by works of Arkani-Hamed et al. [193, Claim 1] and Borinsky [40, Theorem 3].

The most general, proven statement is roughly as follows. Suppose $\{f_1(x), \ldots, f_l(x)\}$ are completely non-vanishing polynomials on $\mathbb{R}^n_{>0}$ (an equivalent statement can be formulated in projective space). Let $s = (s_1, \ldots, s_l) \in \mathbb{C}^l$ and $\nu = (\nu_1, \ldots, \nu_n) \in \mathbb{C}^n$ be

[3] Their method requires expanding out derivatives of $\mathcal{U}$ and $\mathcal{V}$, which can lead to a huge proliferation of terms if done symbolically. This problem is naturally avoided in the setting of numerical MC integration. Indeed, we shall provide efficient formulas in Section (10.3.3) for derivatives of $\mathcal{U}$ and $\mathcal{V}$ that could be used in a future implementation.

complex parameters. Assume $\mathrm{Re}(s_i) > 0$. The integral

$$I = \int_{\mathbb{R}^n_{>0}} \frac{x_1^{\nu_1} \cdots x_n^{\nu_n}}{f_1^{s_1} \cdots f_l^{s_l}} \frac{\mathrm{d}x}{x} \tag{10.32}$$

is convergent iff[4]

$$\begin{bmatrix} \mathrm{Re}(\nu_1) \\ \vdots \\ \mathrm{Re}(\nu_n) \end{bmatrix} \in \ \mathrm{int}(\mathbf{N}(s)) \quad \text{where} \tag{10.33}$$

$$\mathbf{N}(s) = \mathrm{Re}(s_1) \cdot \mathbf{N}[f_1] + \cdots + \mathrm{Re}(s_l) \cdot \mathbf{N}[f_l]\,.$$

- The notation $s \cdot \mathbf{N}[f]$ means that the Newton polytope associated to f is rescaled by a constant $s \in \mathbb{R}_{\geq 0}$. When $\mathbf{N}[f]$ is determined by a collection half-spaces $\{v_i \cdot x \leq c_i\}$, then $s \cdot \mathbf{N}[f]$ has the effect of rescaling the c's, that is $\{v_i \cdot x \leq s\, c_i\}$.
- The *Minkowski sum* of two polytopes $\mathbf{N}[f_A]$ and $\mathbf{N}[f_B]$ is given by $\mathbf{N}[f_A] + \mathbf{N}[f_B] = \big\{a + b \,\big|\, a \in \mathbf{N}[f_A] \text{ and } b \in \mathbf{N}[f_B]\big\}$.
- A point x is in the *interior* of $\mathbf{N}(s)$, $x \in \mathrm{int}(\mathbf{N}(s))$, if there exists a small ball (w.r.t. the Euclidean metric) centered at x that is completely contained in $\mathbf{N}(s)$.

The condition (10.33) can be checked in practice as follows. First compute the half-space representation of $\mathbf{N}(s)$ (for example with SAGE or POLYMAKE). This gives a set of inequalities of the form $\{v \cdot x \leq c\}$ for the hyperplanes that cut out $\mathbf{N}(s)$. For FIs we have $s_1 = \mathrm{D}/2$ and $s_2 = \omega$, so these inequalities would depend parametrically on the DR parameter ϵ. A vector $v = [\mathrm{Re}(\nu_1), \ldots, \mathrm{Re}(\nu_n)]^T$ is then in the interior of $\mathbf{N}(s)$ if the inequalities are strict, namely $\{v \cdot x < c\}$.

This suggests the following heuristic algorithm for determining a quasi-finite basis: check whether the criterion (10.33) is fulfilled during a bottom-up search in the space (ν, D) of propagator powers and spacetime dimensions[5].

We remark that the alternative quasi-finite basis search algorithm from [190] has been automated in REDUZE2 [12] via the command `find_finite_masters`.

[4] An additional, weak assumption is that the Minkowski sum of Newton polytopes $\mathbf{N}[f_1] + \ldots + \mathbf{N}[f_l]$ should have dimension n.

[5] The search space for ν is cut down drastically in case *any* other basis is known, so as to pinpoint a list of relevant integral sectors.

Section 10.3

Tropical Monte Carlo sampling, in detail

We have already briefly explained the idea behind tropical MC sampling in Section (10.1). This algorithm is here described in detail. The goal is to tropically integrate the terms in the ϵ-expansion (10.30), so let us factorize the integrals in a way analogous to (10.15):

$$I_G = I^{\mathrm{tr}} \frac{\Gamma(\omega_0 + \epsilon L)}{\prod_{e\in E} \Gamma(\nu_e)} \sum_{k=0}^{\infty} \frac{\epsilon^k}{k!} I^{(k)} \,, \tag{10.34}$$

where the kth order integral reads as

$$I^{(k)} = \int_{\mathbb{P}_+^{|E|-1}} \frac{\det \mathcal{J}_\lambda(x) \;\times\; \prod_{e\in E}(X_e/x_e)^{\nu_e}}{\big(\mathcal{U}(X)/\mathcal{U}^{\mathrm{tr}}(x)\big)^{\mathrm{D}_0/2} \;\times\; \big(\mathcal{V}(X)/\mathcal{V}^{\mathrm{tr}}(x)\big)^{\omega_0}} \log^k \left[\frac{\mathcal{U}(X)}{\mathcal{V}(X)^L}\right] \;\times\; \mu^{\mathrm{tr}} \,, \tag{10.35}$$

and the *tropical probability measure* is

$$\mu^{\mathrm{tr}} = \frac{1}{I^{\mathrm{tr}}} \frac{\prod_{e\in E} x_e^{\nu_e}}{\mathcal{U}^{\mathrm{tr}}(x)^{\mathrm{D}_0/2} \;\times\; \mathcal{V}^{\mathrm{tr}}(x)^{\omega_0}}\, \Omega \,. \tag{10.36}$$

The normalization factor I^{tr} ensures that $\int_{\mathbb{P}_+^{|E|-1}} \mu^{\mathrm{tr}} = 1$ (a formula for I^{tr} is given further down in (10.46)).

The idea of tropical MC is to sample N points $\{x^{(1)}, \ldots, x^{(N)}\}$ from the measure μ^{tr}, then evaluate the integrand of (10.35) at those points, and finally take the average of these evaluations. Borinsky introduced two different algorithms that sample from μ^{tr} in [40]. The first algorithm [40, Section 5] works for any projective integral over rational functions. A key step in the algorithm requires the triangulation of a certain *normal fans*[6] associated to the Newton polytopes of $\mathcal{U}$ and $\mathcal{F}$. The computation of such triangulations unfortunately becomes a bottleneck when a Feynman diagram has many edges.

The second algorithm [40, Section 6] takes special features of $\mathcal{U}$ and $\mathcal{F}$ into account, thereby avoiding the triangulation step. Namely, the Newton polytope of $\mathcal{U}$ is always a so-called *generalized permutahedron*, and the same holds true for the Newton polytope of $\mathcal{F}$ given sufficiently generic kinematics. The triangulation step of the previous algorithm gets traded for a much simpler preprocessing step, involving the calculation of a certain recursive function called $J(\gamma)$ below.

[6] A normal fan is essentially a collection of cones that can be associated to the normal directions of each face of a given polytope [186].

10.3.1 Generalized permutahedra

Let E be any finite set. Consider a function[7] $z : 2^E \to \mathbb{R}$ that assigns a number to each subset γ of E. Given such a z, the *base polytope* $\mathbf{P}[z] \subset \mathbb{R}^{|E|}$ is defined by the set of points $(a_0, \dots, a_{|E|-1}) \in \mathbb{R}^{|E|}$ satisfying

$$\begin{aligned} \sum_{e \in E} a_e &= z(E) \quad \text{and} \\ \sum_{e \in \gamma} a_e &\geq z(\gamma) \quad \text{for all} \quad \gamma \subsetneq E \,. \end{aligned} \tag{10.37}$$

The function z is called *supermodular* if it satisfies the inequalities

$$z(\gamma) + z(\delta) \ \leq \ z(\gamma \cap \delta) + z(\gamma \cup \delta) \quad \text{for all pairs of subsets} \quad \gamma, \delta \subset E \,. \tag{10.38}$$

Given a supermodular function z, Aguiar and Ardila proved that the base polytope $\mathbf{P}[z]$ is always a *generalized permutahedron* (GP) [194, Theorem 12.3] (as a shorthand, we shall often say that a polytope "has the GP property", or simply "is GP"). This is a wonderful class of polytopes defined by Postnikov in [195]. For the purposes of this discussion, we shall simply *define* a GP to a supermodular base polytope. To build some intuition for these polytopes, let us nevertheless give a geometric description of Postnikov's original construction. To start, define the *standard permutahedron* by

$$\Pi_n = \texttt{ConvexHull}[\text{permutations of } (1, \dots, n)] \subset \mathbb{R}^n \,. \tag{10.39}$$

In words, we take the convex hull in $\mathbb{R}^n$ of the vector $\mathrm{v} = (1, \dots, n)$ and all its $n!$ permutations $\mathrm{v}_\sigma = (\sigma(1), \dots, \sigma(n))$. Equivalently, each permutation of v corresponds to a vertex. The dimension of Π_n is $n-1$ rather than n because its points are all constrained to lie on the hyperplane $1 + 2 + \ldots + n = n(n+1)/2$. Postnikov then defines a GP as a special deformation of a standard permutahedron: one is allowed to move any facet along its normal ray as long as it does not pass through any vertices. Equivalently, one can move any vertex as long as the directions are preserved. Examples of generalized permutahedra are shown in Figure (10.2).

GP property of $\mathcal{U}$

Let us now see how this connects to FIs. Take the set E to be the edge set of a Feynman graph G. A subset of E corresponds to a subgraph γ of G, so z can be thought of as

[7] The notation 2^E stands for the power set of E.

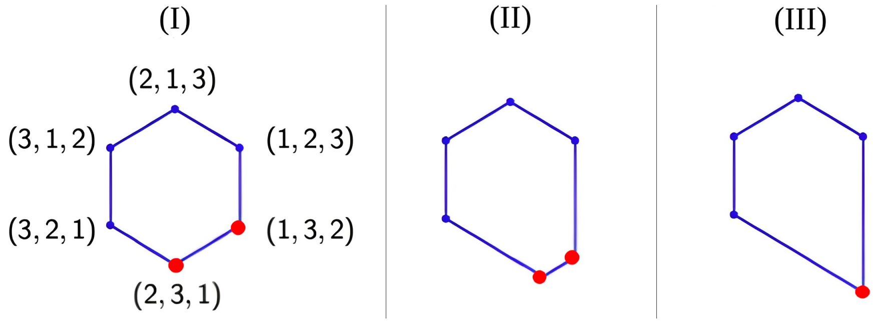


Figure 10.2: **(I)** The standard permutahedron Π_3. Each vertex is labeled by a permutation of $(1, 2, 3)$. Since the vector components (x_1, x_2, x_3) of each vertex satisfies $x_1+x_2+x_3 = 6$, the dimension of the polytope drops from 3 to 2. **(II)** Deforming the two red vertices while preserving edge direction gives a GP. **(III)** Continuing to deform the vertices creates another GP. A facet has now degenerated to a point.

a collection of numbers for each subgraph of G. The integer L_γ denotes the number of loops of γ. It was shown by Schultka [196] that the function

$$z_{\mathcal{U}}(\gamma) = L_\gamma \tag{10.40}$$

is supermodular, and Newton polytope $\mathbf{N}[\mathcal{U}]$ of the $\mathcal{U}$-polynomial equals the base polytope $\mathbf{P}[z_{\mathcal{U}}]$. According to the theorem by Aguiar and Ardila, this means that $\mathbf{N}[\mathcal{U}]$ is GP.

GP property of $\mathcal{F}$

The situation is much more complicated for the $\mathcal{F}$-polynomial. Following Brown [197], call a subgraph $\gamma \subset E$ *mass-momentum spanning* if the $\mathcal{F}$-polynomial restricted to the cograph $G \setminus \gamma$ vanishes identically, $\mathcal{F}_{G\setminus\gamma} = 0$. Since the $\mathcal{F}$-polynomial contains all the kinematic scales $(s, t, m^2$ etc.) of a given FI, an equivalent definition is to say that $\mathcal{F}_{G\setminus\gamma}$ is scaleless.

We further define the notion of *generic kinematics*. Consider the $\mathcal{F}$-polynomial of a 1-loop bubble graph with two different internal masses: $\mathcal{F}_{\text{bubble}}(x) = (p^2 - m_0^2 - m_1^2)x_0x_1 + m_0^2x_0^2 + m_1^2x_1^2$. Notice that if the kinematics are tuned to enforce $p^2 - m_0^2 - m_1^2 = 0$, then the first monomial drops out. Special choices of kinematics can therefore change the Newton polytope $\mathbf{N}[\mathcal{F}]$, as its vertices stem from the monomial exponents of $\mathcal{F}$. The question is now how to avoid this degeneration of $\mathbf{N}[\mathcal{F}]$ by setting a criterion which prevents cancellation among momentum and mass terms in $\mathcal{F}$. To this end, begin by recalling the scalar product matrix $\mathcal{P}_{uv} = p_u \cdot p_v$, where $u, v \in V$ are vertices of G. Call a vertex v *internal* if $\mathcal{P}_{uv} = 0$ for all $u \in V$ and *external* otherwise. We then say that a kinematic configuration is *generic* if

$$\sum_{u,v\,\in\,V'} \mathcal{P}_{uv} \neq \sum_{e\,\in\,E'} m_e^2 \tag{10.41}$$

for each proper subset of external vertices $V' \subsetneq V_{\text{ext}}$ and each non-empty subset of edges $E' \subset E$. When the kinematics are not generic, they are called *exceptional*. For example, kinematics are always generic in the Euclidean regime if 1) $m_e > 0$ for all $e \in E$, or 2) $\sum_{u,v\,\in\,V'} \mathcal{P}_{uv} < 0$ for all $V' \subsetneq V_{\text{ext}}$. See [185, Section 2.2] for additional details.

Now, assuming generic kinematics, it was shown by Schultka [196][8] that the function

$$z_{\mathcal{F}} = \begin{cases} L_\gamma + 1 & \text{if } \gamma \text{ is mass-momentum spanning} \\ L_\gamma & \text{else}\,. \end{cases} \tag{10.42}$$

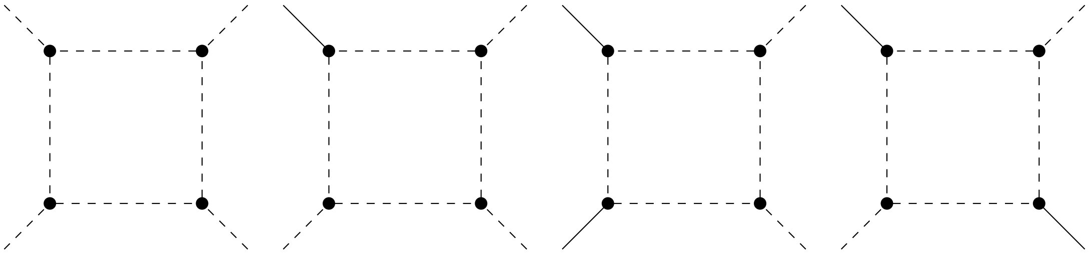
is supermodular, and the Newton polytope $\mathbf{N}[\mathcal{F}]$ equals the base polytope $\mathbf{P}[z_{\mathcal{F}}]$. Therefore, for generic kinematics, the Newton polytope $\mathbf{N}[\mathcal{F}]$ is GP. There are reasons to expect that $\mathbf{N}[\mathcal{F}]$ is also GP even for exceptional kinematics in the Euclidean regime [185, Conjecture 3.7].

It should be emphasized that $\mathbf{N}[\mathcal{F}]$ is generally *not* GP outside of the Euclidean regime. Indeed, the polytopes for three box diagrams in Figures (10.3a), (10.3b) and (10.3c) are *not* GP. The GP property *is* fulfilled for the polytope associated to the diagram (10.3d) though.

The story is further complicated by an interesting observation of Arkani-Hamed, Hillman and Mizera [36, Section IV]: even for exceptional kinematics, the Newton polytope of $\mathcal{F}$ is often equal to the base polytope $\mathbf{P}[z_{\mathcal{F}}]$ with $z_{\mathcal{F}}$ given by (10.42)! In fact, for all four examples in Figure (10.3) we have that $\mathbf{N}[\mathcal{F}] = \mathbf{P}[z_{\mathcal{F}}]$. This is expected for the polytope of the last graph (10.3d) since it is GP, but how is this consistent with the fact that the first three cases (10.3a), (10.3b) and (10.3c) are not GP? The answer is that $z_{\mathcal{F}}$ is not supermodular for those graphs. So the function $z_{\mathcal{F}}$ does *not* fulfil the inequalities (10.38) for the first three boxes, but it *does* so for the last box.

While $\mathbf{N}[\mathcal{F}] = \mathbf{P}[z_{\mathcal{F}}]$ for generic kinematics, we are not aware of a set of necessary and sufficient conditions that specify when $\mathbf{N}[\mathcal{F}]$ equals $\mathbf{P}[z_{\mathcal{F}}]$ for exceptional kinematics. It is at least the case that $\mathbf{N}[\mathcal{F}] \subset \mathbf{P}[z_{\mathcal{F}}]$, because we can only lose monomials in the

[8] Schultka technically proved this theorem with kinematics in the Euclidean regime. The same result also holds in the Minkowski regime because the Newton polytopes coincide in both regimes given the genericity assumption.



(a) On-shell box. Not GP.

(b) One off-shell leg. Not GP.

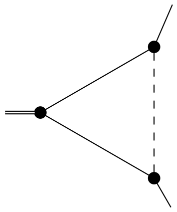
(c) Two off-shell legs (adjacent). Not GP.

(d) Two off-shell legs (crossed). Is GP.

Figure 10.3: 1-loop box diagrams with massless internal propagators and different combinations of on- or off-shell external legs. On-shell ($p^2 = 0$) legs are depicted by dashed lines, and off-shell ($p^2 \neq 0$) legs are drawn as solid lines. In all four cases we have $\mathbf{N}[\mathcal{F}] = \mathbf{P}[z_{\mathcal{F}}]$, but only (10.3d) is GP.

exceptional case. A counter example[9] to equality is given by the following 1-loop triangle:

$$\mathcal{F}_{\text{triangle}} = \mathcal{F}\Bigg($$



$$\Bigg) = m^2(x_0^2 + x_1^2) + (2m^2 - p^2)x_0x_1\,. \tag{10.43}$$

The dashed line is massless, the double line has off-shell momentum $p^2 \neq 0$, and every single solid line has mass m. This is an exceptional kinematic configuration. The Newton polytope of $\mathcal{F}_{\text{triangle}}$ is the convex hull of the points $(2,0), (0,2)$ and $(1,1)$. Since they all lie on a line, the dimension of $\mathbf{N}[\mathcal{F}_{\text{triangle}}]$ equals 1. However, the base polytope $\mathbf{P}[z_{\mathcal{F}_{\text{triangle}}}]$ can be shown to have dimension 2. Hence $\mathbf{P}[z_{\mathcal{F}_{\text{triangle}}}] \neq \mathbf{N}[\mathcal{F}_{\text{triangle}}]$.

Why is this discussion relevant for tropical MC sampling? The answer is that the sampling algorithm of Feyntrop assumes that $\mathbf{N}[\mathcal{F}]$ is GP. Curiously, although we have highlighted the complications with this assumption above, it appears not to matter that much in practice. Indeed, by evaluating the examples from this section with Feyntrop (i.e. quasi-finite versions of the four boxes in Figure (10.3) and the triangle from (10.43)), we find agreement with other numerical codes in all cases. The catch is that numerical convergence for Feyntrop becomes slower in the non-GP cases. We suspect that Feyntrop will fail for more complicated diagrams with exceptional kinematics, but we did not yet find such an example.

[9] We thank Erik Panzer for sharing this example with us.

	Euclidean	Minkowski
Generic	no deformation / always GP	deformation / always GP
Exceptional	no deformation / always GP	deformation / not always GP

Table 10.1: Summary of whether contour deformation is required and whether the GP property of $\mathbf{N}[\mathcal{F}]$ is fulfilled for each kinematic regime.

10.3.2 Tropical sampling algorithm

FEYNTROP employs a modification of [40, Algorithm 4] to sample from the tropical measure μ^{tr} given in (10.36), under the assumption that $\mathbf{N}[\mathcal{F}]$ is GP. The algorithm includes a preprocessing step, to be performed once, followed by a sampling step. The preprocessing step scales as $\mathcal{O}(n2^n)$, where $n = |E|$ is the number of edges in a Feynman diagram. This scaling is determined from the preparation of a certain table of rational numbers $J(\gamma)$ for each subgraph $\gamma \subset E$. The sampling step scales as $\mathcal{O}\left(n^2\delta^{-2}\right)$, where δ is the demanded relative accuracy of the result. This is just ordinary MC scaling.

Because of the preprocessing step, the time complexity of this algorithm is not fully polynomial. It is a very interesting question whether a polynomial time algorithm exists for the numerical evaluation of FIs.

Preprocessing

The first task is to determine the kinematic regime, namely Euclidean vs. Minkowski and generic vs. exceptional.

The regime is Euclidean if the scalar product matrix P_{uv} is negative semi-definite, and Minkowski otherwise. If the kinematics are in the Minkowski regime, then contour deformation needs to be switched on.

If the kinematic point is exceptional and Minkowski, then $\mathbf{N}[\mathcal{F}]$ might not be GP, and it might not even be equal to $\mathbf{P}[z_{\mathcal{F}}]$. In this case, FEYNTROP prints a warning saying that the integration might give the wrong result (though, as mentioned, we have not yet witnessed erroneous numerics). The program then bravely continues under the assumption that $\mathbf{N}[\mathcal{F}] = \mathbf{P}[z_{\mathcal{F}}]$. For generic kinematics, regardless of whether we are in the Euclidean or Minkowski regimes, the sampling algorithm is guaranteed to converge to the correct result [40, Proposition 31].

A summary of the preceding paragraphs is given in Table (10.1). For brevity, we have left out a technicality in that table regarding the Euclidean regime: there is in fact a larger regime, containing the Euclidean one, called the *pseudo-Euclidean regime*; it does not require contour deformation, but might witness a failure of the GP property for exceptional kinematics. See [185, Section 2.2] for details.

The next task is to prepare data associated to each of the $2^{|E|}$ subgraphs $\gamma \subset E$:

1. We calculate the loop number $L_\gamma = |\gamma| - |V_\gamma| + 1$, where V_γ is the vertex set associated to γ.

2. We check whether or not γ is mass-momentum spanning, i.e. whether $\mathcal{F}_{G\backslash\gamma} = 0$. By definition, $\delta_\gamma^{\text{m.m}} = 1$ if γ is mass-momentum spanning and $\delta_\gamma^{\text{m.m}} = 0$ otherwise.

3. We calculate the *generalized degree of divergence* [40, Section 7.2]
$$\omega(\gamma) = \sum_{e\in\gamma} \nu_e - \mathrm{D}_0 L_\gamma/2 - \omega_0\, \delta_\gamma^{\text{m.m}}\,, \tag{10.44}$$
where D_0 is the integer part of the spacetime dimension and $\omega_0 = \sum_{e\in E}\nu_e - \mathrm{D}_0 L/2$. If $\omega(\gamma) \leq 0$ for any proper subgraph (i.e. not the whole graph $\gamma = E$), then G has a subdivergence. It implies that every term of the ϵ-expansion (10.30) is divergent, wherefore Feyntrop prints an error and terminates. In this case, one has to shift the integrand exponents (ν, D) to obtain a quasi-finite integral (cf. Section (10.2.3)).

4. If $\omega(\gamma) > 0$ for all $\gamma \subset E$, then we *recursively* compute a table of numbers $J(\gamma)$ via the formulas
$$\begin{aligned} J(\emptyset) &= \omega(\emptyset) = 1 \\ J(\gamma) &= \sum_{e\in\gamma} \frac{J(\gamma\setminus e)}{\omega(\gamma\setminus e)} \quad \text{for all} \quad \gamma \subset E\,, \end{aligned} \tag{10.45}$$
where $\gamma \setminus e$ means that we delete the edge e from γ. The last step of the recursion is for $\gamma = E$. In [40, Proposition 29], Borinsky proved the surprising identity
$$J(E) = I^{\text{tr}}\,, \tag{10.46}$$
where I^{tr} is the tropical normalization factor from (10.36). The rational numbers $J(\gamma)$ are intimately related to the values $z_{\mathcal{F}}(\gamma)$ [40, Section 6.1]. Indeed, this is the step during preprocessing wherein the GP assumption is employed.

The data $(L_\gamma, \delta_\gamma^{\text{m.m}})$ feeds into the computation of $z_{\mathcal{F}}(\gamma)$ via (10.42). For an exceptional kinematic point in the Minkowski regime, Feyntrop checks whether $z_{\mathcal{F}}$ is supermodular by testing a (more efficient) variant of the inequalities (10.38), see [185, Equation (26)].

FEYNTROP prints a message if the inequalities are satisfied to notify that $\mathbf{P}[z_{\mathcal{F}}]$ is GP, and a warning otherwise.

The output of this preprocessing is the data $\bigcup_{\gamma \subset E} \left\{ L_\gamma,\, \delta_\gamma^{\mathrm{m.m}},\, \omega(\gamma),\, J(\gamma) \right\} \bigcup \left\{ I^{\mathrm{tr}} \right\}$, which is stored in computer memory.

MC sampling

The validity of this tropical sampling algorithm was proven in [40, Proposition 31].

Algorithm 6 Generating a sample x distributed as μ^{tr} from (10.36)

```
Initialize the variables $\gamma = E$ and $\kappa = U = 1$.
while $\gamma \neq \emptyset$ do
    Pick a random edge $e \in \gamma$ with probability
        $$p_e^\gamma = \frac{1}{J(\gamma)} \frac{J(\gamma \setminus e)}{\omega(\gamma \setminus e)}.$$
    Set $x_e = \kappa$.
    If $\gamma$ is mass-momentum spanning but $\gamma \setminus e$ is not, set $V = x_e$.
    If $L_{\gamma \setminus e} < L_\gamma$, multiply $U$ with $x_e$ and store the result in $U$, i.e. set $U \leftarrow x_e \cdot U$.
    Remove the edge $e$ from $\gamma$, i.e. set $\gamma \leftarrow \gamma \setminus e$.
    Pick a uniformly distributed random number $\xi \in [0, 1]$.
    Multiply $\kappa$ with $\xi^{1/\omega(\gamma)}$ and store the result in $\kappa$, i.e. set $\kappa \leftarrow \kappa\, \xi^{1/\omega(\gamma)}$.
end while
return $x = [x_0, \ldots, x_{|E|-1}] \in \mathbb{P}_+^{|E|-1}$, $\mathcal{U}^{\mathrm{tr}}(x) = U$ and $\mathcal{V}^{\mathrm{tr}}(x) = V$.
```

The main idea of this algorithm is to interpret p_e^γ as a probability distribution over the edges e of a given subgraph γ. Indeed, it follows from the definition of $J(\gamma)$ in (10.45) that $p_e^\gamma \geq 0$ and $\sum_{e \in \gamma} p_e^\gamma = 1$. Starting from $\gamma = E$, the algorithm proceeds to *cut* an edge e of the graph G with probability p_e^γ. This yields a new graph $\gamma \setminus e$ with the edge removed. The cutting process repeats until all edges are gone.

We refer to [40, Section 6.1] for explanations on why Algorithm (6) yields correct values for the triple $\left\{ x,\, \mathcal{U}^{\mathrm{tr}}(x),\, \mathcal{V}^{\mathrm{tr}}(x) \right\}$.

10.3.3 Evaluating deformed Feynman integrands

Algorithm (6) explains how to efficiently sample points x from μ^{tr}, but it does not take care of all the extra factors in the integrand of (10.30) that arose due to contour deformation. In particular, we would like to efficiently evaluate the factors $X_e = x_e \exp\left[-i\lambda\partial_e\mathcal{V}(x)\right], \mathcal{U}(X)$, $\mathcal{V}(X)$ and $\mathcal{J}_\lambda(x)$ for any $x \in \mathbb{P}_+^{|E|-1}$.

The non-trivial step in evaluating X_e is the derivative $\partial_e\mathcal{V}(x)$. Moreover, the Jacobian $\mathcal{J}_\lambda(x)$ includes the second-order derivative $\partial_e\partial_h\mathcal{V}(x)$ according to (10.28). For the sake of efficiency, it would be nice to have formulas for these derivatives based only on linear algebra. To this end, begin by defining the $(|V|-1)\times(|V|-1)$ matrix

$$\mathcal{M}(x) = \mathcal{L}^{-1}(x)\cdot\mathcal{P}\cdot\mathcal{L}^{-1}(x)\,, \tag{10.47}$$

where the Laplacian matrix $\mathcal{L}(x)$ and the scalar product matrix $\mathcal{P}$ come from (10.22). To avoid confusion, we clarify that $\mathcal{L}^{-1}_{uv}$ means the (u,v)th component of the inverse matrix $\mathcal{L}^{-1}$. Let e and h be two edges that respectively connect the vertices (u_e, v_e) and (u_h, v_h). Furthermore, define two matrices component-wise by

$$\begin{aligned}
\mathcal{A}_{eh}(x) &= \frac{1}{x_e x_h}\left[\mathcal{M}_{u_e\,u_h}(x) + \mathcal{M}_{v_e\,v_h}(x) - \mathcal{M}_{u_e\,v_h}(x) - \mathcal{M}_{v_e\,u_h}(x)\right]\\
\mathcal{B}_{eh}(x) &= \frac{1}{x_e x_h}\left[\mathcal{L}^{-1}_{u_e\,u_h}(x) + \mathcal{L}^{-1}_{v_e\,v_h}(x) - \mathcal{L}^{-1}_{u_e\,v_h}(x) - \mathcal{L}^{-1}_{v_e\,u_h}(x)\right],
\end{aligned}$$

with the convention that $\mathcal{L}^{-1}_{uv}(x) = \mathcal{M}_{uv}(x) = 0$ if either u or v equal the vertex v_0 that was removed in the FI (10.16) by momentum conservation. Using the identity

$$\frac{\partial}{\partial x_e}\mathcal{L}^{-1}_{uv}(x) = \left(\mathcal{L}^{-1}(x)\cdot\frac{\partial\mathcal{L}(x)}{\partial x_e}\cdot\mathcal{L}^{-1}(x)\right)_{uv}\,, \tag{10.48}$$

a tedious but straightforward calculation gives the sought after derivative formulas

$$\frac{\partial\mathcal{V}(x)}{\partial x_e} = -\mathcal{A}_{ee}(x) + m_e^2 \tag{10.49}$$

$$\frac{\partial^2\mathcal{V}(x)}{\partial x_e\partial x_h} = \frac{2\delta_{eh}\mathcal{A}_{ee}(x)}{x_e} - 2\big(\mathcal{A}(x)\circ\mathcal{B}(x)\big)_{eh}\,, \tag{10.50}$$

where the last term denotes the element-wise matrix product (also called the Hadamard product): $\big(\mathcal{A}(x)\circ\mathcal{B}(x)\big)_{eh} = \mathcal{A}(x)_{eh}\times\mathcal{B}(x)_{eh}$.

Summary

Let us summarize all the non-trivial steps that must be carried out to evaluate the integrand of (10.30) at a single point. Cholesky decomposition is used to evaluate determinants and inverses whenever possible, that is when the matrix in question is Hermitian and positive semi-definite (see the discussion around equation (10.23)). When those two conditions are invalid, the less efficient LU decomposition is opted for instead.

1. Use Algorithm (6) to output the triple $\big\{x, \mathcal{U}^{\mathrm{tr}}(x), \mathcal{V}^{\mathrm{tr}}(x)\big\}$.
2. Compute the Laplacian matrix $\mathcal{L}(x)$ via (10.20).
3. Compute the inverse $\mathcal{L}^{-1}(x)$ via Cholesky decomposition.
4. Use $\mathcal{L}(x)$ and $\mathcal{L}^{-1}(x)$ to evaluate derivatives of $\mathcal{V}(x)$ using (10.49) and (10.50).
5. Compute the deformed X parameters by $X_e = x_e \exp\left[-i\lambda\partial_e\mathcal{V}(x)\right]$.
6. Compute the Jacobian $\mathcal{J}_\lambda(x)$ using (10.28).
7. Evaluate $\det[\mathcal{J}_\lambda(x)]$ using LU decomposition.
8. Compute the deformed Laplacian $\mathcal{L}(X)$.
9. Compute $\mathcal{L}^{-1}(X)$ and $\det[\mathcal{L}(X)]$ using LU decomposition.
10. Employ the formulas (10.22) to evaluate $\mathcal{U}(X)$, $\mathcal{F}(X)$ and $\mathcal{V}(X) = \mathcal{F}(X)/\mathcal{U}(X)$.

For a kinematic point in the Euclidean regime we are allowed to set $\lambda = 0$ implying $X = x$, in which case Feyntrop only executes step 1.

Chapter 11

The Program FEYNTROP

We have implemented the contour-deformed tropical integration algorithm described in the previous chapter in a C++ module named Feyntrop. The module is equipped with a python interface. The C++ codebase is an upgrade to the one developed by Borinsky in [40]. Feyntrop relies on Eigen3 [198] for optimized linear algebra routines, on OpenMP [199] for parallelization of MC sampling, and on xoshiro256+ [200] for pseudorandom number generation.

Feyntrop has been verified against AMFlow [34] and pySecDec [176] for roughly 15 different 2- to 5-point Feynman diagrams between 1 and 3 loops at varying kinematic points, both in the Euclidean and Minkowski regimes. Numerical agreement was found in all cases within the given uncertainty bounds. In the Euclidean regime, the code was checked up to $\mathcal{O}(10)$ loop-orders in [40] against analytic results for conformal 4-point functions from φ^4-theory.

In this chapter we explain how to use the code, and thereafter present state-of-the-art examples of numerical evaluations of FIs.

Section 11.1

Basic usage of Feyntrop

Source code for Feyntrop is available in the github repository

https://github.com/michibo/feyntrop .

To download and install it, the following commands can be run in a Linux terminal:

```
git clone https://github.com/michibo/feyntrop.git
cd feyntrop
make clean && make
```

(It should also work in a macOS environment, although C++ compiler issues may ensue. See the README.txt document on github).

Feyntrop is loaded in a python environment by importing the file py_feyntrop.py located in the top directory. To test whether the installation was successful, the commands

```
cd tests
python test_suite.py
```

will compute six examples with 1-2 loops and 2-5 legs and compare the results with precomputed values.

The python interface serves three purposes:

1. It simplifies the specification of vertices V and edges E of a Feynman diagram, in comparison with the more verbose data structures of C++.

2. It allows for self-chosen momentum variables given by a list of replacement rules, rather than having to manually input the full matrix of scalar products $\mathcal{P}_{uv}$ from (10.22).

3. The numerical results for the ϵ-expansion are printed in an easily readable format.

It is also possible to directly interface with the C++ module via .json files. See the README.txt document in the github repository for more details on this.

11.1.1 Tutorial

The basic workflow of Feyntrop is illustrated here by means of an example. The code can be inspected with jupyter [201] by calling

```
jupyter notebook tutorial_2L_3pt.ipynb
```

in the top directory of the Feyntrop package. We will evaluate the 2-loop 3-point function shown in Figure (11.1) in $\mathrm{D} = 2 - 2\epsilon$ dimensions. The dashed lines represent massless on-shell external momenta: $p_0^2 = p_1^2 = 0$. The solid, internal lines each have mass m. The double line denotes an off-shell external momentum $p_2^2 \neq 0$. The four vertices $V = V_{\text{ext}} \sqcup V_{\text{int}} = \{0, 1, 2\} \sqcup \{3\}$ and four edges $E = \{0, 1, 2, 3\}$ have been explicitly labeled. The convention of Feyntrop requires one to label external vertices *before* internal ones.

By the conventions of (10.16), we have the following momentum space representation with edge weights $\nu_0 = \ldots = \nu_4 = 1$:

$$\pi^{-2+2\epsilon} \int \frac{\mathrm{d}^{2-2\epsilon}\ell_0 \, \mathrm{d}^{2-2\epsilon}\ell_1}{(q_0^2 - m^2 + i\varepsilon)(q_1^2 - m^2 + i\varepsilon)(q_2^2 - m^2 + i\varepsilon)(q_3^2 - m^2 + i\varepsilon)(q_4^2 - m^2 + i\varepsilon)} \, ,$$

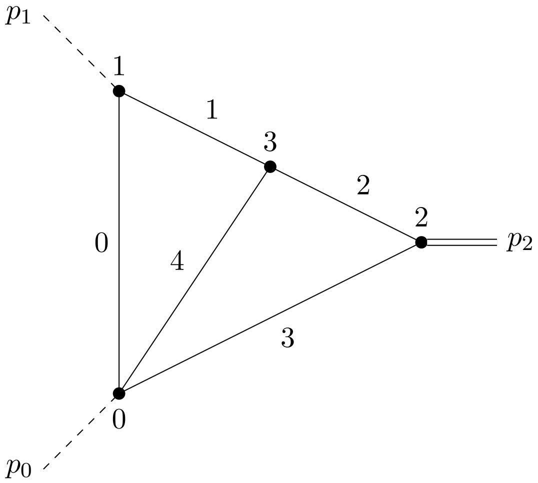


Figure 11.1: 2-loop 3-point function in the Feyntrop tutorial.

where the $\delta^{(D)}$-functions of (10.16) have been integrated out by setting

$$q_0 = \ell_0\,, \quad q_1 = \ell_0 + p_1\,, \quad q_2 = \ell_0 + \ell_1 + p_1\,, \quad q_3 = p_0 - \ell_0 - \ell_1\,, \quad q_4 = \ell_1\,. \tag{11.1}$$

We evaluate the integral at the phase space point

$$m^2 = 0.2\,, \quad p_0^2 = p_1^2 = 0\,, \quad p_2^2 = 1\,, \tag{11.2}$$

which is in the Minkowski regime given that $p_2^2 > 0$.

The Feyntrop computation begins by importing `py_feyntrop.py` in a python environment:

```
from py_feyntrop import *
```

Here it is assumed that the files `feyntrop.so` and `py_feyntrop.py` are both in the working directory. The graph in Figure (11.1) is defined by providing a list of edges, edge weights ν_e and squared masses m_e^2 in the format

$$\Big[\big((u_0, v_0)\,,\,\nu_0\,,\,m_0^2\big)\,,\,\ldots\,,\,\big((u_{|E|-1}, v_{|E|-1})\,,\,\nu_{|E|-1}\,,\,m_{|E|-1}^2\big)\Big]\,. \tag{11.3}$$

The notation $(u_e,\, v_e)$ means that there is an edge e connecting the vertices u_e and v_e. The vertex ordering $(u_e,\, v_e)$ is insignificant. For the example at hand, we thus write

```
edges =  [((0,1), 1, 'mm'), ((1,3), 1, 'mm'), ((2,3), 1, 'mm'),
          ((2,0), 1, 'mm'), ((0,3), 1, 'mm')]
```

The edge weights were here set to $\nu_e = 1$ for all e. The chosen symbol `mm` stands for m^2, and will be replaced by its value 0.2 later on[1].

The next step is to fix the momentum variables. By the zero-indexing convention, we have external momenta $\{p_0, \dots, p_{|V_\text{ext}|-1}\}$. But the last momentum $p_{|V_\text{ext}|-1}$ is automatically inferred by Feyntrop using momentum conservation, leaving $\{p_0, \dots, p_{|V_\text{ext}|-2}\}$ to be fixed by the user. A momentum configuration is then specified by providing a collection of scalar products

$$p_u \cdot p_v \quad \text{for all} \quad 0 \leq u \leq v \leq |V_\text{ext}| - 2\,. \tag{11.4}$$

In the code, we must give a replacement rule for each scalar product in terms of some variables of choice. In the present example there are $|V_\text{ext}| = 3$ external vertices, so we ought to provide replacement rules for $\{p_0^2, p_1^2, p_0 \cdot p_1\}$. We have $p_0^2 = p_1^2 = 0$, and by momentum conservation,

$$p_2^2 = (-p_0 - p_1)^2 = 2p_0 \cdot p_1 \implies p_0 \cdot p_1 = p_2^2/2\,. \tag{11.5}$$

In the syntax of Feyntrop we then write

```
replacement_rules = [(sp[0,0], '0'), (sp[1,1], '0'), (sp[0,1], 'pp2/2')]
```

The scalar product between p_u and p_v is represented by the symbol `sp[u,v]`, and `pp2` corresponds to p_2^2. Since we will eventually substitute values for the two parameters `pp2` and `mm`, we further define

```
phase_space_point = [('mm', 0.2), ('pp2', 1)]
```

which fixes $m^2 = 0.2$ and $p_2^2 = 1$.

The final data to be provided are

```
D0 = 2
eps_order = 5
Lambda = 7.6
N = int(1e7)
```

`D0` is the integer part of $\mathrm{D} = \mathrm{D}_0 - 2\epsilon$. We expand up to, but not including, `eps_order`. `Lambda` is the contour deformation parameter in (10.24). `N` denotes the number of MC sampling points, so increasing this value will yield better numerical accuracy (the error on MC sampling scales as $\mathcal{O}(1/\sqrt{\mathtt{N}})$ according to (10.4)).

Tropical Monte Carlo integration is finally performed by calling

[1] It is also allowed to input numerical values for the masses already in the `edges` list. The first entry, for instance, would then be `((0,1), 1, '0.2')`.

```
trop_res, Itr = tropical_integration(
        N,
        D0,
        Lambda,
        eps_order,
        edges,
        replacement_rules,
        phase_space_point)
```

If no error is printed, then `trop_res` will contain the ϵ-expansion (10.30) *without* the prefactor $\Gamma(\omega)/\big(\Gamma(\nu_0)\cdots\Gamma(\nu_{|E|-1})\big) = \Gamma(2\epsilon+3)$. `Itr` is the value of the tropical normalization factor from (10.46). Running this code on a laptop gives the output

```
Prefactor: gamma(2*eps + 3).
(Effective) kinematic regime: Minkowski (generic).
Generalized permutahedron property: fulfilled.
Analytic continuation: activated. Lambda = 7.6
Started integrating using 8 threads and N = 1e+07 points.
Finished in 6.00369 seconds = 0.00166769 hours.

-- eps^0: [-46.59  +/- 0.13] + i * [ 87.19  +/- 0.12]
-- eps^1: [-274.46 +/- 0.55] + i * [111.26  +/- 0.55]
-- eps^2: [-435.06 +/- 1.30] + i * [-174.47 +/- 1.33]
-- eps^3: [-191.72 +/- 2.15] + i * [-494.69 +/- 2.14]
-- eps^4: [219.15  +/- 2.68] + i * [-431.96 +/- 2.67]
```

`trop_res` contains the printed ϵ-expansion coefficients in the format

$$\Big[\,\big((\mathrm{re}_0,\,\sigma_0^{\mathrm{re}})\,,\,(\mathrm{im}_0,\,\sigma_0^{\mathrm{im}})\big)\,,\,\dots\,,\,\big((\mathrm{re}_4,\,\sigma_4^{\mathrm{re}})\,,\,(\mathrm{im}_4,\,\sigma_4^{\mathrm{im}})\big)\,\Big]\,,$$

where $\mathrm{re}_0 \pm \sigma_0^{\mathrm{re}}$ is the real part of the 0th order term, and so forth. To obtain the ϵ-expansion of the integral with the prefactor included, one may call

```
eps_expansion(trop_res, edges, D0)
```

which gives

```
174.3842115*i - 93.17486662 + eps*(-720.8731714 + 544.3677186*i) +
eps**2*(-2115.45025 + 496.490128*i) + eps**3*(-3571.990969 - 677.5254794*i) +
eps**4*(-3872.475723 - 2726.965026*i) + O(eps**5)
```

On the choice of λ

A few comments are in order regarding the choice of the deformation parameter λ (what was called `Lambda` in the code). The analytic expression for the integral (10.29) is independent of λ (so long as the deformed contour doesn't hit any poles), but the numerical error coming from MC sampling depends rather sensitively on its value. Moreover, the optimal value for λ will generally vary depending on the phase space point. It is currently up to the user to find a suitable value by trial and error, e.g. by performing several integrations with low sampling numbers N. A heuristic starting value is $\mathcal{O}(1/\Lambda^2)$, where Λ^2 is the largest kinematic scale in the $\mathcal{F}$-polynomial (for instance some mass m^2).

In the future, it would be beneficial to automate the choice of λ either by 1) minimizing the sampling variance σ_f, given in (10.4), by solving for λ in

$$\partial_\lambda \sigma_f = 0\,, \tag{11.6}$$

or 2) by tightening the tropical approximation bounds in (10.31).

Section 11.2

Examples of Feynman integral evaluations

In this section, we use Feyntrop to numerically evaluate many FIs of interest. Each script is available in the `/examples` folder in the gitub repository. When possible, the results have been verified against AMFlow and/or pySecDec[2]. The exception is the example from Section (11.2.1), as we were not been able to re-compute these values with any other software in reasonable time - though it might be possible now with the new release of pySecDec [177].

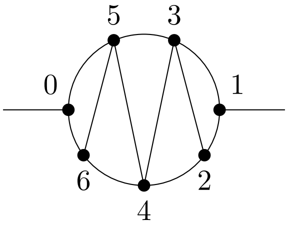
Each example can be computed on a laptop within a few minutes. Though to properly showcase the potential of Feyntrop, we shall only include results obtained from a high-performance machine: a single `AMD EPYC 7H12` 64-core processor using all cores. With $N = 10^8$ MC sampling points each example only took a few seconds to finish. The relative accuracies of the results fall between 10^{-2} to 10^{-4}.

The phase space points are chosen to be away from any kinematic thresholds to ensure numerical convergence. In all but the last example, the ϵ-expansion goes up to and including $\mathcal{O}(\epsilon^{2L})$.

[2] We thank Vitaly Magerya for help with verifying results using pySecDec.

11.2.1 5-loop 2-point zigzag diagram

Here we evaluate a 5-loop 2-point function with all masses different in D = $3-2\epsilon$ dimensions:



The edge set is

```
edges = [((0,6), 1, ’1’) , ((0,5), 1, ’2’), ((5,6), 1, ’3’),
         ((6,4), 1, ’4’) , ((5,3), 1, ’5’), ((5,4), 1, ’6’),
         ((4,3), 1, ’7’) , ((4,2), 1, ’8’), ((3,2), 1, ’9’),
         ((3,1), 1, ’10’), ((2,1), 1, ’11’)]
```

Above we already input the chosen mass values

$$m_0^2 = 1\,, \quad m_1^2 = 2\,, \quad \ldots, \quad m_{10}^2 = 11\,. \tag{11.7}$$

The remaining kinematic variable to be specified is the incoming momentum. We set $p_0^2 = 100$ in Feyntrop by writing

```
replacement_rules = [(sp[0,0], ’pp0’)]
phase_space_point = [(’pp0’, 100)]
```

The value $\lambda = 0.02$ turns out to yield small errors. This is of order $\mathcal{O}(1/p_0^2)$, in accordance with the heuristic described near the end of the previous section. With $N = 10^8$ MC sampling points, the command `tropical_integration` in Feyntrop gives

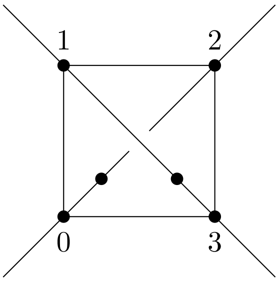
```
Prefactor: gamma(5*eps + 7/2).
(Effective) kinematic regime: Minkowski (generic).
Finished in 9.62 seconds.
-- eps^0: [0.0001976 +/- 0.0000016]  +  i * [0.0001415 +/- 0.0000018]
-- eps^1: [-0.004961 +/- 0.000023 ]  +  i * [-0.000802 +/- 0.000024 ]
-- eps^2: [ 0.04943  +/-  0.00017 ]  +  i * [-0.01552  +/-  0.00017 ]
-- eps^3: [-0.25468  +/-  0.00083 ]  +  i * [ 0.24778  +/-  0.00093 ]
-- eps^4: [ 0.5909   +/-  0.0033  ]  +  i * [ -1.7261  +/-  0.0038  ]
-- eps^5: [  1.048   +/-   0.012  ]  +  i * [   7.410  +/-   0.013  ]
-- eps^6: [ -14.652  +/-   0.037  ]  +  i * [ -20.933  +/-   0.038  ]
-- eps^7: [  65.87   +/-   0.10   ]  +  i * [  35.25   +/-   0.11   ]
-- eps^8: [ -190.90  +/-   0.27   ]  +  i * [  -4.91   +/-   0.26   ]
-- eps^9: [ 393.08   +/-   0.70   ]  +  i * [ -182.56  +/-   0.59   ]
-- eps^10:[ -558.01  +/-   1.64   ]  +  i * [ 685.62   +/-   1.29   ]
```

For fun, the same computation with $N = 10^{12}$ samples results in

```
Finished in 20 hours.
-- eps^0: [0.000196885 +/- 0.000000032]  +  i * [0.000140824 +/- 0.000000034]
-- eps^1: [-0.00493791 +/- 0.00000040 ]  +  i * [-0.00079691 +/- 0.00000038 ]
-- eps^2: [ 0.0491933  +/-  0.0000025 ]  +  i * [-0.0154647  +/-  0.0000025 ]
-- eps^3: [ -0.253458  +/-  0.000012  ]  +  i * [ 0.246827   +/-  0.000012  ]
-- eps^4: [ 0.587258   +/-  0.000046  ]  +  i * [ -1.720213  +/-  0.000046  ]
-- eps^5: [  1.05452   +/-   0.00015  ]  +  i * [  7.38725   +/-   0.00015  ]
-- eps^6: [ -14.66144  +/-   0.00047  ]  +  i * [ -20.86779  +/-   0.00046  ]
-- eps^7: [  65.8924   +/-   0.0013   ]  +  i * [  35.0793   +/-   0.0013   ]
-- eps^8: [ -190.9702  +/-   0.0036   ]  +  i * [  -4.4620   +/-   0.0034   ]
-- eps^9: [ 393.2522   +/-   0.0092   ]  +  i * [ -183.7431  +/-   0.0087   ]
-- eps^10:[ -558.202   +/-    0.023   ]  +  i * [  688.556   +/-    0.021   ]
```

11.2.2 3-loop 4-point envelope diagram

This example is a non-planar, 3-loop 4-point diagram in D $= 4 - 2\epsilon$ dimensions:



The dots on the crossed lines represent squared propagators rather than vertices; this choice of edge weights makes the diagram quasi-finite. The edge data are

```
edges = [((0,1), 1, 'mm0'), ((1,2), 1, 'mm1'), ((2,3), 1, 'mm2'),
         ((3,0), 1, 'mm3'), ((0,2), 2, 'mm4'), ((1,3), 2, 'mm5')]
```

Note that the edge weights ν_4 and ν_5 equal 2.

We define two-index Mandelstam variables by $s_{ij} = (p_i + p_j)^2$. They are input into replacement rules for Feyntrop by writing `(sp[i,j], '(sij - ppi - ppj)/2')` with $0 \leq i \leq j \leq 2$. The chosen phase space point is

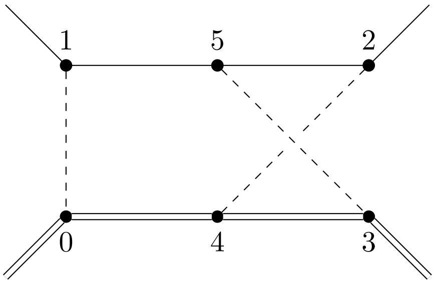
$$\begin{array}{rclrclrcl} p_0^2 &=& 1.1\,, & p_1^2 &=& 1.2\,, & p_2^2 &=& 1.3\,, \\ s_{01} &=& 2.1\,, & s_{02} &=& 2.2\,, & s_{12} &=& 2.3\,, \\ m_0^2 &=& 0.05\,, & m_1^2 &=& 0.06\,, & m_2^2 &=& 0.07\,, \\ m_3^2 &=& 0.08\,, & m_4^2 &=& 0.09\,, & m_5^2 &=& 0.1\,. \end{array} \tag{11.8}$$

Setting $\lambda = 1.24$ and $N = 10^8$, we get the result

```
Prefactor: gamma(3*eps + 2).
(Effective) kinematic regime: Minkowski (generic).
Finished in 5.12 seconds.
-- eps^0: [-10.8335 +/- 0.0084]  +  i * [-12.7145 +/- 0.0083]
-- eps^1: [ 47.971  +/- 0.059 ]  +  i * [-105.057 +/- 0.059 ]
-- eps^2: [ 413.05  +/-  0.23 ]  +  i * [  7.29   +/-  0.23 ]
-- eps^3: [ 372.07  +/-  0.65 ]  +  i * [ 947.82  +/-  0.65 ]
-- eps^4: [-1412.36 +/-  1.45 ]  +  i * [1325.74  +/-  1.45 ]
-- eps^5: [-2726.00 +/-  2.67 ]  +  i * [-1295.36 +/-  2.69 ]
-- eps^6: [ 287.25  +/-  4.28 ]  +  i * [-3982.04 +/-  4.30 ]
```

11.2.3 2-loop 4-point μe-scattering diagram

We evaluate a non-planar, 2-loop 4-point diagram that contributes to muon-electron scattering. It is quasi-finite in $\mathrm{D} = 6 - 2\epsilon$ dimensions. The diagram was evaluated for vanishing electron mass in [202], but here we keep the electron massive.



The dashed lines represent photons. The solid lines are electrons with mass m. The double lines denote muons with mass $M \sim 200m$. The set of edges is

```
edges = [((0,1), 1, ’0’), ((0,4), 1, ’MM’), ((1,5), 1, ’mm’), ((5,2), 1, ’mm’),
         ((5,3), 1, ’0’), ((4,3), 1, ’MM’), ((4,2), 1, ’0’)]
```

where `mm` and `MM` stand for m^2 and M^2 respectively. We choose a phase space point similar to that of [202, Section 4.1.2]:

$$p_0^2 = M^2 = 1\,, \quad p_1^2 = p_2^2 = m^2 = 1/200\,, \quad s_{01} = -1/7\,, \quad s_{12} = -1/3\,. \tag{11.9}$$

By momentum conservation, this fixes

$$s_{02} = 2M^2 - 2m^2 - s_{01} - s_{12} = 2.49\,. \tag{11.10}$$

With additional settings $\lambda = 1.29$ and $N = 10^8$, the result comes out to

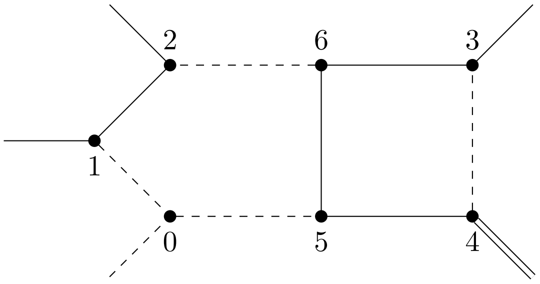
```
Prefactor: gamma(2*eps + 1).
(Effective) kinematic regime: Minkowski (exceptional).
Finished in 6.53 seconds.
-- eps^0: [1.16483 +/- 0.00083]  +  i * [0.24155 +/- 0.00074]
-- eps^1: [5.5387  +/- 0.0086 ]  +  i * [2.2818  +/- 0.0093 ]
-- eps^2: [15.171  +/-  0.058 ]  +  i * [10.079  +/-  0.064 ]
-- eps^3: [ 28.02  +/-  0.32  ]  +  i * [ 28.17  +/-  0.28  ]
-- eps^4: [ 38.20  +/-  1.42  ]  +  i * [ 56.94  +/-  0.85  ]
```

Because the momentum configuration is exceptional, we cannot be sure that the GP property holds (cf. Section (10.3.1)). In spite of that, we are able to confirm the values above with both AMFlow and pySecDec.

The leading-order term in ϵ differs from [202, Equation (4.20)] by roughly 10% because we have included the electron mass. If we set $m^2 = 0$ in Feyntrop, then we are able to reproduce the value from that reference (even though this is an even more exceptional configuration!).

11.2.4 2-loop 5-point QCD-like diagram

Here we evaluate a QCD-like, $(6-2\epsilon)$-dimensional graph with 2 loops and 5 legs:



The dashed lines represent on-shell gluons. The solid lines are quarks each having mass m. The double line denotes some off-shell leg $p_4^2 \neq 0$ fixed by momentum conservation.

The edge list is

```
edges = [((0,1), 1, '0'), ((1,2), 1, 'mm'), ((2,6), 1, '0'), ((6,3), 1, 'mm'),
         ((3,4), 1, '0'), ((4,5), 1, 'mm'), ((5,0), 1, '0'), ((5,6), 1, 'mm')]
```

where `mm` is the symbol standing for m^2. Our chosen phase space point is

$$\begin{aligned} &s_{01} = 2.2\,, \quad s_{02} = 2.3\,, \quad s_{03} = 2.4\,, \quad s_{12} = 2.5\,, \\ &s_{13} = 2.6\,, \quad s_{23} = 2.7\,, \quad p_0^2 \;=\; 0\,, \quad p_1^2 = p_2^2 = p_3^2 = m^2 = 1/2\,, \end{aligned} \tag{11.11}$$

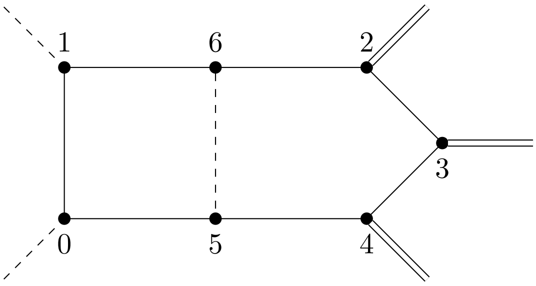
with $s_{ij} = (p_i + p_j)^2$ as before. With settings $\lambda = 0.28$ and $N = 10^8$, Feyntrop gives the result

```
Prefactor: gamma(2*eps + 2).
(Effective) kinematic regime: Minkowski (exceptional).
Finished in 8.20 seconds.
-- eps^0: [0.06480 +/- 0.00078]  +  i * [-0.08150 +/- 0.00098]
-- eps^1: [0.4036  +/- 0.0045 ]  +  i * [ 0.3257  +/- 0.0035 ]
-- eps^2: [-0.7889 +/- 0.0060 ]  +  i * [ 0.957   +/-  0.016 ]
-- eps^3: [-1.373  +/-  0.030 ]  +  i * [ -1.181  +/-  0.034 ]
-- eps^4: [ 1.258  +/-  0.088 ]  +  i * [ -1.205  +/-  0.036 ]
```

The kinematic configuration is once again exceptional. The result is nevertheless verified with pySecDec.

11.2.5 2-loop 5-point Higgs production diagram

This example contributes to triple Higgs production via gluon fusion, $gg \to HHH$, in $\mathrm{D} = 4 - 2\epsilon$ dimensions:



The dashed lines represent massless, on-shell gluons. The single solid lines are propagators containing the top quark mass. The three external double lines are put on-shell to the Higgs mass.

The edge data are

```
edges = [((0,1), 1, ’mm_top’), ((1,6), 1, ’mm_top’), ((5,6), 1, ’0’),
         ((6,2), 1, ’mm_top’), ((2,3), 1, ’mm_top’), ((3,4), 1, ’mm_top’),
         ((4,5), 1, ’mm_top’), ((5,0), 1, ’mm_top’)]
```

with `mm_top` standing for the square of top quark mass, m_t^2. The kinematic setup is given by

$$\begin{aligned} &p_0^2 = p_1^2 = 0\,, \quad p_2^2 = p_3^2 = p_4^2 = m_H^2\,, \\ &s_{01} = 5m_H^2 - s_{02} - s_{03} - s_{12} - s_{13} - s_{23}\,, \end{aligned} \tag{11.12}$$

with the last relation being a consequence of momentum conservation. So the space of kinematic variables is parametrized by $(s_{02}, s_{03}, s_{12}, s_{23}, m_t^2, m_H^2)$, and we choose the point

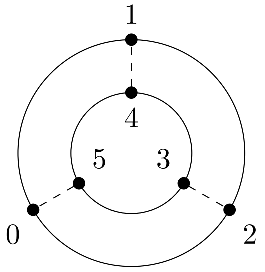
$$\begin{aligned} &m_t^2 = 1.8995\,, \quad m_H^2 = 1\,, \qquad s_{02} = -4.4\,, \quad s_{03} = -0.5\,, \\ &s_{12} = -0.6\,, \qquad s_{13} = -0.7\,, \quad s_{23} = 1.8\,. \end{aligned} \tag{11.13}$$

This point lies in the physical region, and it contains the physically relevant mass ratio $m_t^2/m_H^2 = 1.8995$. For $\lambda = 0.64$ and $N = 10^8$, the FI then evaluates to

```
Prefactor: gamma(2*eps + 4).
(Effective) kinematic regime: Minkowski (generic).
Finished in 8.12 seconds.
-- eps^0: [-0.0114757 +/- 0.0000082]  +  i * [0.0035991 +/- 0.0000068]
-- eps^1: [ 0.003250  +/- 0.000031 ]  +  i * [-0.035808 +/- 0.000041 ]
-- eps^2: [ 0.046575  +/- 0.000098 ]  +  i * [0.016143  +/- 0.000088 ]
-- eps^3: [ -0.01637  +/-  0.00017 ]  +  i * [ 0.03969  +/-  0.00016 ]
-- eps^4: [ -0.02831  +/-  0.00023 ]  +  i * [-0.00823  +/-  0.00024 ]
```

11.2.6 4-loop 0-point QED-like diagram

Next we evaluate a QED-like, 4-loop vacuum diagram in D $= 4 - 2\epsilon$ dimensions:



The dashed lines are photons, and the solid lines are fermions with mass m. Since this diagram has no external momenta, it is not necessary to perform any contour deformation. This means that we can set $\lambda = 0$, and result is expected to be purely real. In the Feyntrop script, we write `replacement_rules = []` to specify that no external momenta are present.

The edge set is given by

```
edges = [((0,1), 1, ’mm’), ((1,2), 1, ’mm’), ((2,0), 1, ’mm’),
         ((0,5), 1, ’0’ ), ((1,4), 1, ’0’ ), ((2,3), 1, ’0’ ),
         ((3,4), 1, ’mm’), ((4,5), 1, ’mm’), ((5,3), 1, ’mm’)]
```

with `mm` standing for m^2. Setting this mass to unity by

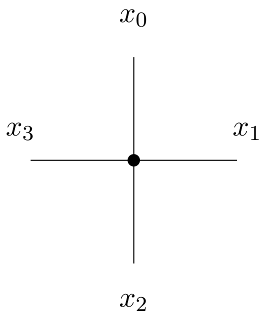
```
phase_space_point = [(’mm’, 1)]
```

the result for $N = 10^8$ sampling points becomes

```
Prefactor: gamma(4*eps + 1).
(Effective) kinematic regime: Euclidean (generic).
Finished in 3.58 seconds.
-- eps^0: [3.01913 +/- 0.00047]  +  i * [0.0 +/- 0.0]
-- eps^1: [-7.0679 +/- 0.0021 ]  +  i * [0.0 +/- 0.0]
-- eps^2: [20.5399 +/- 0.0074 ]  +  i * [0.0 +/- 0.0]
-- eps^3: [-27.895 +/-  0.024 ]  +  i * [0.0 +/- 0.0]
-- eps^4: [62.043  +/-  0.074 ]  +  i * [0.0 +/- 0.0]
-- eps^5: [-59.46  +/-  0.23  ]  +  i * [0.0 +/- 0.0]
-- eps^6: [155.27  +/-  0.73  ]  +  i * [0.0 +/- 0.0]
-- eps^7: [-90.81  +/-  2.26  ]  +  i * [0.0 +/- 0.0]
-- eps^8: [403.78  +/-  6.71  ]  +  i * [0.0 +/- 0.0]
```

11.2.7 An elliptic, conformal, 4-point integral

The final example is a 1-loop 4-point conformally invariant integral [203]. The analytic result for this diagram with edge weights $\nu_0 = \nu_1 = \nu_2 = \nu_3 = 1/2$ and dimension $\mathrm{D} = 2$ was calculated in [204, Section 7.2]:



$$= \frac{4}{\sqrt{-p_2^2}} \big[K(z)K(1-\bar{z}) + K(\bar{z})K(1-z)\big] \,. \tag{11.14}$$

This result features the elliptic integral K given by

$$K(z) = \int_0^1 \frac{\mathrm{d}x}{\sqrt{(1-x^2)(1-z^2x^2)}} \tag{11.15}$$

and kinematic variables

$$z\bar{z} = \frac{p_0^2}{p_2^2} \,, \quad (1-z)(1-\bar{z}) = \frac{p_1^2}{p_2^2} \,. \tag{11.16}$$

The denominator of (11.14) differs from [204, Equation (7.6)] because we have used conformal symmetry to send $x_3 \to \infty$, where x_3 is a dual momentum variable (defined by $p_i = x_i - x_{i+1}$). This reduces the kinematic space to that of a 3-point function.

In Feyntrop we thus specify an edge list for a 1-loop graph with 3 external legs:

```
edges = [((0,1), 1/2, ’0’), ((1,2), 1/2, ’0’), ((2,0), 1/2, ’0’)]
```

All internal masses are set to zero, otherwise conformal symmetry would be broken, and every edge weight has been set to 1/2. It is most natural to study conformal integrals in the Euclidean regime, so let us pick

$$p_0^2 = -2\,, \quad p_1^2 = -3\,, \quad p_2^2 = -5\,. \tag{11.17}$$

With $N = 10^8$ sampling points, we obtain

```
(Effective) kinematic regime: Euclidean (generic).
Finished in 1.34 seconds.
-- eps^0: [9.97192 +/- 0.00027]  +  i * [0.0 +/- 0.0]
```

This result agrees with the analytic formula (11.14). We hope that our code can furnish further study into conformal integrals, as they are important for $\mathcal{N} = 4$ SYM and the cosmological bootstrap.

Chapter 12

Conclusion and Outlook

Conclusion

This thesis has been an investigation into the mathematical structure of Feynman integrals. Its overarching goal has been to develop novel techniques for evaluating these integrals.

In **Part I** of this thesis, we studied integration-by-parts identities and the method of Pfaffian DEQs for evaluating Feynman integrals. When the Feynman integrals in question admit a solution in terms of multiple polylogarithms, we showed how the Magnus expansion can bring the Pfaffian system into canonical form, whose solution is readily obtained via the path-ordered exponential.

These techniques were used to compute form factors for a leptophillic dark matter model. The calculation involved a class of 2-loop 3-point Feynman integrals with massive internal propagators, which were solved for in terms of multiple polylogarithms.

In **Part II**, we delved deeper into the mathematics of DEQs by formalizing the notion of a $\mathcal{D}$-module. A $\mathcal{D}$-module encapsulates the algebraic properties of differential operators that annihilate a given function such as a Feynman integral.

We studied a broad class of functions called Euler integrals containing Feynman integrals as a special case. Euler integrals are annihilated by a known collection of higher-order differential operators called the GKZ system. We showed how to connect the GKZ system to the more manageable first-order Pfaffian system through the Macaulay matrix.

In some cases, the Euler integral matched exactly with a Feynman integral. In other cases, the Euler integral had too many unfixed parameters. The latter case was treated by developing two new $\mathcal{D}$-module restriction methods. One of those methods drove us to establish an algorithm for obtaining logarithmic series solutions to Pfaffian systems.

In **Part III**, we formulated a numerical integration scheme for the evaluation of Feynman integrals which combined Monte Carlo integration with tropical geometry. The algorithm relied on the assumption that the Newton polytope of the $\mathcal{F}$ Symanzik polynomial is a generalized permutahedron, and we gave a detailed discussion of when this can be expected to hold. The $i\varepsilon$-prescription was incorporated into the algorithm via contour deformation, which allowed us to evaluate Feynman integrals in physical regions of kinematic space. The algorithm was implemented in a publicly available program named FEYNTROP, whose validity was verified against existing numerical codes. We used FEYNTROP to numerically evaluate Feynman integrals between 1-5 loops and 0-5 legs.

Outlook

There are several avenues of research to pursue based on the results of this thesis.

Applying the Macaulay matrix to other ideals. Algorithm (1) takes a holonomic ideal $\mathcal{I}$ together with a $\mathcal{D}$-module basis as input, feeds it through the Macaulay matrix, and outputs a Pfaffian system. In all examples of this thesis, we worked with the GKZ system $\mathcal{I} = H_A(\beta)$ simply because $H_A(\beta)$ and a basis of standard monomials are easily determined given an Euler integral. However, there might very well be other situations where an annihilating ideal is known. If a basis is not known, it could be guessed by executing Algorithm (1) (with parameters set to numbers) for different choices of standard monomials until it succeeds.

A non-GKZ example is for instance the conformally invariant, massless, 1-loop 3-point function studied in [139]. An annihilating ideal was there determined from the generators of conformal symmetry. It would be interesting to study these operators with the Macaulay matrix, as well as those coming from the larger Yangian symmetry group [205].

Another route would be to employ the algorithms from [180, 206, 207] to compute Picard-Fuchs ideals for Feynman integrals; these algorithms were effective for rather complicated diagrams, e.g. at 2 loops and 7 points. As an aside, it would be interesting to see whether a canonical Feynman integral has any nice structure when it is represented as an operator inside the $\mathcal{D}$-module $\mathcal{D}/\mathcal{I}$, where $\mathcal{I}$ is the Picard-Fuchs ideal.

More complicated examples of logarithmic restriction. An algorithm for the calculation of logarithmic series solutions to Pfaffian systems was presented in Section (8.2.4). We gave results for 1-loop Bhabha scattering in Section (9.2), and partial results for the 2-loop Bhabha double-box in Section (9.3). It would be instructive to first complete the 2-loop calculation; this involves the determination of boundary constants and power corrections $(m^2/s)^n$.

Afterwards, it would be interesting to see how far this approach can go. The complexity of the restricted Pfaffian systems for the 2-loop Bhabha double-box was miniscule compared to today's standards, indicating that more complicated diagrams could potentially be evaluated with this method.

Expansion around singular locus. Recall the form of the logarithmic expansion ansatz (8.44). There we chose a distinguished singularity $z_1 = 0$ to expand around. It would be interesting to reformulate the logarithmic restriction procedure to instead expand around the entire singular locus $L(z)$ of a Pfaffian system, where $L(z)$ is a product of irreducible hypersurfaces. For instance, the singular locus for Appell's F_4 function is $L(z) = z_1 z_2[(z_1 - z_2)^2 - 2(z_1 + z_2) + 1]$. For Feynman integrals, $L(z)$ would correspond to the *Landau variety* [188].

The series ansatz is now roughly represented as (see [157, Equation (12.15)] for details)

$$\vec{I}(z) \quad \sim \quad L(z)^{\lambda} \times \sum_{n=0}^{\infty} \sum_{m=0}^{M} \vec{I}^{(n,m)}(z) \times L(z)^n \times \log[L(z)]^m \,, \tag{12.1}$$

where $\lambda \in \mathbb{R}$ and $M \in \mathbb{Z}_{\geq 0}$. Analytical solutions to Feynman integrals are often hard to numerically evaluate near Landau singularities, but a solution written in this fashion could potentially have better numerical stability.

Applications in algebraic statistics. Looking beyond physics, one finds that GKZ systems and Euler integrals also frequently appear in the course of Bayesian inference in algebraic statistics [208, Section 3].

The article [209] gave an initial application of the Macaulay matrix in this context; it is natural to expect that this research direction can be pushed even further with the more sophisticated Macaulay matrix method developed here. Further, it appears that the Pfaffian-level restriction protocol from Section (8.2) can be used to study contingency tables[1] in statistics [210].

Numerators and ϵ-divergent integrals in FEYNTROP. The program FEYNTROP presented in Chapter (11) has two main shortcomings which would be beneficial to eliminate in future versions:

[1] These tables are a standard tool in statistics to record interrelations between random variables.

1. It only works for scalar Feynman integrals. But in gauge theories such as QED and QCD, one encounters numerators in momentum space representation of the form $(p \cdot \ell)^{\nu}$, where p and ℓ are external and loop momenta respectively.

2. It assumes a quasi-finite integral representation (cf. Section (10.2.2)).

By the formulas of [189, Appendix A], numerators can be written in the Symanzik representation by including an extra (homogeneous) polynomial $\mathcal{N}$ inside the integrand. There is no reason to expect that this polynomial would enjoy the generalized permutahedron property though, so it might not be straightforward to include it in the MC sampling of Algorithm (6).

The quasi-finite condition can be lifted by following the protocol in [191, Page 5], where explicit formulas are given for factoring out ϵ-poles in front of finite integrals. These formulas would involve derivatives of the $\mathcal{U}$ and $\mathcal{F}$ Symanzik polynomials, and we fortunately have efficient expressions for those (cf. Section (10.3.3)).

With these two additions, we envision that FEYNTROP could be used in a wide range of phenomenological calculations. Especially pertaining to scattering amplitudes involving many mass scales, which are the hardest ones to treat analytically with DEQs due to the proliferation of functions beyond the class of multiple polylogarithms[2].

A canonical contour deformation. In Section (10.2.1), we implemented the $i\varepsilon$-prescription via a contour deformation of the Schwinger parameters $x_e \to x_e \exp\left[-i\lambda\partial_e\mathcal{V}\right]$. The small, but finite, number $\lambda \in \mathbb{R}_{>0}$ is a free parameter. An explicit value has to be specified by the user in FEYNTROP, and the accuracy of the result depends quite sensitively on this choice.

It would be a significant development to determine a *canonical* choice for λ, depending on the data of the Feynman integral in question as well as the given phase space point. This would not only improve the user experience in FEYNTROP, but also constitute the first example of a contour deformation for Feynman integrals without any free parameters.

[2] FEYNTROP has already been used to check results for a non-planar 3-loop calculation related to Higgs+jet production [211].

Appendix A

Details on $\mathcal{D}$-modules

Here we collect some additional details on $\mathcal{D}$-modules that are referred to in the main text.

Section A.1

Basis change in a holonomic $\mathcal{D}$-module

The goal in this section is to construct the Pfaffian system for the basis e

$$\partial_i e = P_i^{(e)} \cdot e \tag{A.1}$$

given a known system in the standard monomial basis

$$\partial_i \mathrm{Std} = P_i^{(\mathrm{Std})} \cdot \mathrm{Std}\,. \tag{A.2}$$

$e \in (\mathcal{R}/\mathcal{I})^R$ is here an R-dimensional vector of (equivalence classes of) differential operators.

Observe that, if we could find a matrix G relating the two bases by

$$e = G \cdot \mathrm{Std}\,, \tag{A.3}$$

then the problem would be solved by a gauge transformation[1]

$$P_i^{(e)} = \Big[\partial_i \bullet G \;\;+\;\; G \cdot P_i^{(\mathrm{Std})}\Big] \cdot G^{-1}\,. \tag{A.4}$$

[1] Note that G is invertible under the assumption that e and Std are both bases.

To construct G, we begin by writing the decomposition

$$e = \sum_q G_q^{(1)} \cdot (\partial^q \, \mathrm{Std}) \;, \tag{A.5}$$

where $G_q^{(1)} \in \mathbb{C}(z)^{R\times R}$. In other words, we write each element of e as a sum of rational functions times derivatives ∂^q that act on Std. The idea is now to transform the expression $\partial^q \, \mathrm{Std}$ into the form $G_q^{(2)} \cdot \mathrm{Std}$, for some $G_q^{(2)} \in \mathbb{C}(z)^{R\times R}$, by repeated application of (A.2). This is best illustrated by means of an example.

Example A.1. Let $q = \{1, 1, 0, 0, \ldots, 0\}$. Then $\partial^q = \partial_1 \partial_2$. Applying (A.2) twice, the expression $\partial^q \, \mathrm{Std}$ becomes

$$\begin{aligned} \partial_1 \partial_2 \, \mathrm{Std} &= \partial_1 \left[P_2^{(\mathrm{Std})} \cdot \mathrm{Std} \right] && \text{(A.6)} \\ &= \left[\partial_1 \bullet P_2^{(\mathrm{Std})} \right] \cdot \mathrm{Std} \;+\; P_2^{(\mathrm{Std})} \cdot [\partial_1 \, \mathrm{Std}] && \text{(A.7)} \\ &= \left[\partial_1 \bullet P_2^{(\mathrm{Std})} \right] \cdot \mathrm{Std} \;+\; P_2^{(\mathrm{Std})} \cdot P_1^{(\mathrm{Std})} \cdot \mathrm{Std} \,. && \text{(A.8)} \end{aligned}$$

The matrix

$$G_q^{(2)} = \partial_1 \bullet P_2^{(\mathrm{Std})} \;+\; P_2^{(\mathrm{Std})} \cdot P_1^{(\mathrm{Std})} \,, \tag{A.9}$$

hence satisfies the sought after relation $\partial^q \, \mathrm{Std} = G_q^{(2)} \cdot \mathrm{Std}$. ■

We conclude that the gauge transformation matrix G in (A.3) takes the form

$$G = \sum_q G_q^{(1)} \cdot G_q^{(2)} \tag{A.10}$$

where $G_q^{(2)}$ is built from matrix products between Pfaffian matrices $P_i^{(\mathrm{Std})}$ and derivatives thereof.

Section A.2

Basis change without derivatives

We concluded above that the gauge transformation matrix (A.10) is constructable from derivatives of Pfaffian matrices $P_i(z)$. In the GKZ setting we might, however, like to fix some of the z-variables to numbers. Then we no longer have the luxury of taking derivatives. As a remedy to this, let us derive a formula for $\partial_i \bullet P_j$ which is based solely on matrix multiplication.

Using the same setup as in Section (6.6), we set

$$f(\beta) = \frac{1}{\Gamma(\beta_0+1)} \int_\Gamma g(z|x)^{\beta_0} x^{-\beta'} \frac{\mathrm{d}x}{x} \quad \text{and} \quad F(\beta) = \begin{bmatrix} \mathrm{Std}_1 \bullet f(\beta) \\ \vdots \\ \mathrm{Std}_R \bullet f(\beta) \end{bmatrix}. \tag{A.11}$$

As described in Section (6.6), there are two ways of expressing the derivative of $F(\beta)$:

$$\partial_j \bullet F(\beta) = P_j(\beta) \cdot F(\beta) \tag{A.12}$$

$$\partial_j \bullet F(\beta) = F(\beta - a_j)\,. \tag{A.13}$$

Differentiating (A.12) w.r.t. z_i, we have

$$\partial_i\, \partial_j \bullet F(\beta) = (\partial_j \bullet P_i) \cdot F + P_j \cdot (\partial_i \bullet F) \tag{A.14}$$

$$= (\partial_j \bullet P_i) \cdot F + P_j \cdot P_i \cdot F\,, \tag{A.15}$$

where we omitted the β-dependence for clarity. On the other hand, differentiating (A.13) yields

$$\partial_i\, \partial_j \bullet F(\beta) = \partial_i \bullet F(\beta - a_j) \tag{A.16}$$

$$= F(\beta - a_j - a_i) \tag{A.17}$$

$$= P_j(\beta - a_i) \cdot P_i(\beta) \cdot F(\beta)\,, \tag{A.18}$$

where the identity $P_k(\beta) \cdot F(\beta) = F(\beta - a_k)$ was applied twice in the final step. Equating (A.15) with (A.18) and isolating $(\partial_i \bullet P_j)$, we conclude that

$$(\partial_i \bullet P_j)(\beta) = \big[P_j(\beta - a_i) - P_j(\beta)\big] \cdot P_i(\beta)\,. \tag{A.19}$$

The RHS of this equation can then be inserted into the G-matrix of (A.10).

Section A.3

General formula for homogeneity rescaling

We gave an example in Section (6.5) of how to rescale the GKZ variables $z = (z_1, \ldots, z_N)$ so as to fix $n+1$ of them to unity (cf. equation (6.61)). This section presents a general formula.

To start, given an $(n+1) \times N$ matrix $A = \big[a_1 \mid \dots \mid a_N\big]$ and a list of $n+1$ rescaling parameters $t = \{t_1, \dots, t_{n+1}\}$, we construct a list of N exponentials

$$t^A := \{t^{a_1}, \dots, t^{a_N}\} \quad \text{where} \quad t^{a_i} := \prod_{j=1}^{n+1} t_j^{(a_i)_j} \,. \tag{A.20}$$

This matrix exponentiation notation obeys the rule $(t^A)^B = t^{A \cdot B}$. For two sets r and s, both of length N, the product $r \times s$ is defined to be

$$r \times s := \{r_1 s_1, \dots, r_N s_N\} \,. \tag{A.21}$$

In this language, the homogeneity property (6.57) reads

$$f_\beta(t^A \times z) = t^\beta f_\beta(z) \,. \tag{A.22}$$

Now fix two subsets

$$\sigma \subset \{1, \dots, N\} \quad \text{and} \quad \eta = \{1, \dots, N\} \setminus \sigma \,, \tag{A.23}$$

where σ has size $|\sigma| = n+1$, and η is its complement. Let A_σ denote the submatrix of A whose columns are labeled by σ. We assume that σ is chosen such that A_σ is invertible, i.e. $\det(A_\sigma) \neq 0$.

Returning to (A.22), a judicial choice for t turns out to be

$$t = z_\sigma^{-A_\sigma^{-1}} \,, \tag{A.24}$$

where z_σ picks out the variables of $z = \{z_1, \dots, z_N\}$ labeled by σ. The homogeneity relation then becomes

$$f_\beta\big(z_\sigma^{-A_\sigma^{-1} \cdot A} \times z\big) = z_\sigma^{-A_\sigma^{-1} \cdot \beta} f_\beta(z) \,. \tag{A.25}$$

For this choice of t, all the arguments on the LHS labeled by σ are set to unity:

$$\Big\{ z_\sigma^{-A_\sigma^{-1} \cdot A} \times z \Big\}_i = \begin{cases} 1 \,, & i \in \sigma \\ z_\sigma^{-A_\sigma^{-1} \cdot a_i} \, z_i \,, & i \in \eta \,. \end{cases} \tag{A.26}$$

As desired, the function on the RHS of (A.25)

$$g_\beta(w) := z_\sigma^{-A_\sigma^{-1}\cdot\beta} f_\beta(z) \tag{A.27}$$

now effectively only depends on $N-(n+1)$ cross ratios

$$w := \{w_1, \ldots, w_{N-(n+1)}\} := z_\sigma^{-A_\sigma^{-1}\cdot A_\eta} \times z_\eta\,. \tag{A.28}$$

What's left is to rewrite the GKZ system in terms of the w-variables. To this end, we summarize a result which is proved in [82, Appendix A]. Given two positive integers $a, b \in \mathbb{Z}_{>0}$, we first define the *rising* and *falling factorials* by

$$(a)_b := a(a+1)(a+2)\cdots(a+b+1) \tag{A.29}$$

$$[a]_b := a(a-1)(a-2)\cdots(a-b+1)\,. \tag{A.30}$$

Given an N-dimensional vector u and an integer vector $v \in \mathbb{Z}^N_{\geq 0}$, we extend these definitions to

$$(u)_v := (u_1)_{v_1}\cdots(u_N)_{v_N} \tag{A.31}$$

$$[u]_v := [u_1]_{v_1}\cdots[u_N]_{v_N}\,. \tag{A.32}$$

In what follows, we shall set

$$u = \theta_w = \left[\theta_{w_1},\, \ldots,\, \theta_{w_{N-(n+1)}}\right]^T, \tag{A.33}$$

where $\theta_i = z_i\partial_i$.

To rewrite the GKZ system in terms of w, it suffices to find a relation between $\partial_z^v \bullet f_\beta(z)$ and $g_\beta(w)$ for $v \in \mathbb{Z}^N_{\geq 0}$, The master formula works out to

$$\partial_z^v \bullet f_\beta(z) = \tag{A.34}$$

$$(-1)^{|v_\sigma|} z_\sigma^{-A_\sigma^{-1}\cdot\left(\beta + A\cdot v\right)} \left\{ w^{-v_\eta} \left(A_\sigma^{-1}\cdot\beta + A_\sigma^{-1}\cdot A_\eta\cdot\theta_w\right)_{v_\sigma} [\theta_w]_{v_\eta} \;\bullet\; g_\beta(w)\right\} \Big|_{w=(\text{A.28})},$$

where $|v_\sigma| = \sum_{i\in\sigma} v_i$. Although this expression looks somewhat daunting, it has fortunately been implemented in the ASIR package MT_GKZ.

Example A.1. For the ${}_2F_1$ case of Example (6.1), the following script will output a GKZ system in rescaled variables:

```
A = [
        [1,1,1,1],
        [0,1,0,1],
        [0,0,1,1]
];
Beta = [b1,b2,b3];
Sigma = [1,2,3];
Ideal = mt_gkz.gkz_b(A, Beta, Sigma | partial = 1);
```

The choice of `Sigma` means that we choose to rescale $z_1 = z_2 = z_3 = 1$. The option `partial = 1` indicates that the ideal is to be written in terms of ∂_i rather than θ_i symbols. The list `Ideal` will contain the single operator (6.65) (albeit in variables x_i rather than z_i, as this is the convention for GKZ variables in ASIR). ■

The formula (6.44) for the operator d_q, which maps a differential form $[\omega_q]$ to its corresponding $\mathcal{D}$-module element, has also been implemented with rescaled variables via the command `mt_gkz.rvec_red2(...)`.

Section A.4

GKZ systems are isomorphic to twisted cohomology groups

Section (6.3.2) mentioned an isomorphism between the GKZ system $\mathcal{D}_N/H_A(\beta)$ and the twisted de Rham cohomology group $\mathbb{H}^k_A(\beta)$. This section provides a very rough sketch of the proof behind this theorem. Heavy machinery from category theory is required to prove this in detail, which is outside the scope of this thesis, so we will content ourselves with summarizing the key steps. The idea is to show that the GKZ system and the twisted cohomology group are both isomorphic to another $\mathcal{D}$-module $\int_\pi \mathcal{L}$, to be described shortly, wherefore $\mathcal{D}/H_A(\beta)$ and $\mathbb{H}^k_A(\beta)$ must also be isomorphic.

Begin by recalling the space $\mathcal{O}(X)$ from (6.32). It is equipped with the structure of an abelian group under the addition of functions. As mentioned in Section (4.1), we can turn an abelian group into a $\mathcal{D}$-module by specifying an action of the Weyl algebra $\mathcal{D}$. Given a function $h \in \mathcal{O}(X)$, the action is here

$$\partial_{z_i} \bullet h = \frac{\partial h}{\partial z_i} + \frac{\beta_0}{g(z|x)} \frac{\partial g(z|x)}{\partial z_i} h \tag{A.35}$$

$$\partial_{x_i} \bullet h = \frac{\partial h}{\partial x_i} + \frac{\beta_0}{g(z|x)} \frac{\partial g(z|x)}{\partial x_i} h - \frac{\beta_i}{x_i} h \,. \tag{A.36}$$

The resulting $\mathcal{D}$-module is denoted by the symbol $\mathcal{L}$ instead of $\mathcal{O}(X)$.

Let $\pi : (z, x) \to z$ be a projection map. The *direct image* $\mathcal{D}$-module $\int_\pi \mathcal{L}$, has an abstract categorical definition[2], but for the case of a projection it can be concretely realized as a quotient space that we proceed to build. Write

$$\mathcal{D}_{N+n} = \mathbb{C}\langle z_1, \ldots, z_N, x_1, \ldots, x_n, \partial_{z_1}, \ldots, \partial_{z_N}, \partial_{x_1}, \ldots, \partial_{x_n}\rangle \tag{A.37}$$

for the ring of differential operators in $N+n$ variables. The subring $\mathcal{D}_N$ contains operators in the z-variables only. It is possible to write $\mathcal{L} = \mathcal{D}_{N+n}/\mathcal{I}$, where $\mathcal{I}$ is the annihilating ideal of the Euler inte*grand*

$$U = g(z|x)^{\beta_0} x^{\beta'} . \tag{A.38}$$

By differentiating U w.r.t. (z, x) and recalling the expression for $g = g(z|x)$ from (6.2), a short calculation shows that $\mathcal{I}$ is generated by

$$\mathcal{I} = \sum_{i=1}^{N} \mathcal{D}\big[g\,\partial_{z_i} - \beta_0\, x^{\alpha_i}\big] \;+\; \sum_{i=1}^{n} \mathcal{D}\Big[x_i\, g\, \partial_{x_i} - \beta_0 \sum_{j=1}^{N} (\alpha_j)_i\, z_j\, x^{\alpha_j} + \beta_i\, g\Big] , \tag{A.39}$$

where $(\alpha_j)_i$ denotes the ith component of the multi-index $\alpha_j \in \mathbb{Z}^n$. The direct image $\mathcal{D}$-module then turns out to be represented by the quotient

$$\int_\pi \mathcal{L} = \frac{\mathcal{D}_{N+n}}{\mathcal{I} + \sum_{i=1}^n \partial_{x_i} \mathcal{D}_{N+n}} . \tag{A.40}$$

This is now a $\mathcal{D}_N$-module, as the x-variables have been "projected out" by the map π. The RHS of (A.40) is also called the *integration module* [86, Chapter 6][3]. It is an algebraic counterpart to the familiar integration operation in calculus for the following reason: writing

$$\int_\pi \mathcal{L} = \mathcal{D}_N/\mathcal{J} \tag{A.41}$$

for some $\mathcal{J}$, then the generators of $\mathcal{J}$ can be shown to annihilate the Euler inte*gral* itself [85, Chapter 5]. It is rather surprising that the integration of a function, which is generally a highly transcendental operation, can be captured in purely algebraic terms!

[2] In categorical terms, given a morphism $f : X \to Y$ between two complex manifolds X and Y, the direct image $\int_f$ is a *functor* depending on f. Loosely speaking, it acts as a certain *pushforward* from sheaves on X to sheaves on Y. See [23, Section 1.3] for details.

[3] There are several ASIR packages for computing annihilation ideals of Euler integrands [212]. An algorithm for computing integration modules has been implemented in the package NK_RESTRICTION [149].

Now, it is a classical result that the twisted cohomology group $\mathbb{H}^k_A(\beta)$ is isomorphic to the direct image $\int_\pi \mathcal{L}$ as a $\mathcal{D}$-module [23, Chapter 4] [213]. The addition of differential forms an abelian group, and $\mathbb{H}^k_A(\beta)$ indeed becomes a $\mathcal{D}$-module under the action

$$\partial_{z_i} \bullet [\omega(z)] = \left[\frac{\partial \omega(z)}{\partial z_i} + \beta_0 \frac{x^{\alpha_i}}{g(z|x)} \omega(z) \right] , \tag{A.42}$$

where $[\omega(z)] \in \mathbb{H}^k_A(\beta)$ denotes an equivalence class of differential forms. What is more, Gel'fand, Kapranov and Zelevinsky proved that the annihilating ideal $\mathcal{J}$ defined above precisely equals $H_A(\beta)$ [42]. In other words, $\int_\pi \mathcal{L}$ is isomorphic to the GKZ system.

Since $\int_\pi \mathcal{L}$ is isomorphic to both the GKZ system *and* the twisted cohomology group, we finally infer that

$$\mathcal{D}_N / H_A(\beta) \quad \cong \quad \mathbb{H}^k_A(\beta) \, . \tag{A.43}$$

We emphasize that this isomorphism only holds true when the z-variables are indeterminate and the β-parameters are non-resonant.

Bibliography

[1] E. Wigner, *The unreasonable effectiveness of mathematics in the natural sciences*, Commun. Pure Appl. Math. **13** (1960) 1.

[2] D.A. Kosower, R. Monteiro and D. O'Connell, *The SAGEX review on scattering amplitudes Chapter 14: Classical gravity from scattering amplitudes*, J. Phys. A **55** (2022) 443015 [2203.13025].

[3] R.P. Feynman, *Space-time approach to quantum electrodynamics*, *Physical Review* **76** (1949) 769.

[4] ATLAS, *Standard Model Summary Plots Summer 2019*, *https://cds.cern.ch/record/2682186* (2019) .

[5] G. Travaglini et al., *The SAGEX review on scattering amplitudes*, J. Phys. A **55** (2022) 443001 [2203.13011].

[6] F.V. Tkachov, *A Theorem on Analytical Calculability of Four Loop Renormalization Group Functions*, Phys. Lett. B **100** (1981) 65.

[7] K.G. Chetyrkin and F.V. Tkachov, *Integration by Parts: The Algorithm to Calculate beta Functions in 4 Loops*, Nucl. Phys. B **192** (1981) 159.

[8] S. Laporta, *High precision calculation of multiloop Feynman integrals by difference equations*, Int. J. Mod. Phys. A **15** (2000) 5087 [hep-ph/0102033].

[9] T. Gehrmann, P. Jakubčík, C.C. Mella, N. Syrrakos and L. Tancredi, *Planar three-loop QCD helicity amplitudes for V+jet production at hadron colliders*, 2307.15405.

[10] J. Klappert, F. Lange, P. Maierhöfer and J. Usovitsch, *Integral reduction with Kira 2.0 and finite field methods*, Comput. Phys. Commun. **266** (2021) 108024 [2008.06494].

[11] A.V. Smirnov and F.S. Chuharev, *FIRE6: Feynman Integral REduction with Modular Arithmetic*, *Comput. Phys. Commun.* **247** (2020) 106877 [1901.07808].

[12] A. von Manteuffel and C. Studerus, *Reduze 2 - Distributed Feynman Integral Reduction*, 1201.4330.

[13] Z. Wu, J. Boehm, R. Ma, H. Xu and Y. Zhang, *NeatIBP 1.0, A package generating small-size integration-by-parts relations for Feynman integrals*, 2305.08783.

[14] X. Guan, X. Liu, Y.-Q. Ma and W.-H. Wu, *Blade: A package for block-triangular form improved Feynman integrals decomposition*, *https://gitlab.com/multiloop-pku/blade/* (2023) .

[15] R.N. Lee, *LiteRed 1.4: a powerful tool for reduction of multiloop integrals*, *J. Phys. Conf. Ser.* **523** (2014) 012059 [1310.1145].

[16] G. Barucchi and G. Ponzano, *Differential equations for one-loop generalized Feynman integrals*, *J. Math. Phys.* **14** (1973) 396.

[17] A.V. Kotikov, *Differential equation method: The Calculation of N point Feynman diagrams*, *Phys. Lett. B* **267** (1991) 123.

[18] E. Remiddi, *Differential equations for Feynman graph amplitudes*, *Nuovo Cim. A* **110** (1997) 1435 [hep-th/9711188].

[19] T. Gehrmann and E. Remiddi, *Differential equations for two loop four point functions*, *Nucl. Phys. B* **580** (2000) 485 [hep-ph/9912329].

[20] J.M. Henn, *Multiloop integrals in dimensional regularization made simple*, *Phys. Rev. Lett.* **110** (2013) 251601 [1304.1806].

[21] P. Mastrolia and S. Mizera, *Feynman Integrals and Intersection Theory*, *JHEP* **02** (2019) 139 [1810.03818].

[22] P. Schapira, *Fifty years of mathematics with Masaki Kashiwara*, in *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pp. 97–110, World Scientific, 2018.

[23] R. Hotta, K. Takeuchi and T. Tanisaki, *D-modules, perverse sheaves, and representation theory*, vol. 236 of *Progress in Mathematics*, Birkhäuser Boston, Inc., Boston, MA (2008), 10.1007/978-0-8176-4523-6.

[24] A. Borel, P.-P. Grivel, B. Kaup, A. Haefliger, B. Malgrange and F. Ehlers, *Algebraic D-modules*, vol. 2 of *Perspectives in Mathematics*, Academic Press, Inc., Boston, MA (1987).

[25] M. Kashiwara, T. Kawai and T. Oshima, *A study of Feynman integrals by micro-differential equations*, *Communications in Mathematical Physics* **60** (1978) 97.

[26] M. Kashiwara and T. Kawai, *Holonomic Systems of Linear Differential Equations and Feynman Integrals*, *Publ. Res. Inst. Math. Sci. Kyoto* **12** (1977) 131.

[27] J.G. Fujimoto and T. Kaneko, *Grace and loop integrals*, in *Loops and Legs in Quantum Field Theory: 11th DESY Workshop on Elementary Particle Physics*, 2013, https://api.semanticscholar.org/CorpusID:126023042.

[28] E. Nasrollahpoursamami, *Periods of Feynman Diagrams and GKZ D-modules*, 1605.04970.

[29] T. Bitoun, C. Bogner, R.P. Klausen and E. Panzer, *Feynman integral relations from parametric annihilators*, *Lett. Math. Phys.* **109** (2019) 497 [1712.09215].

[30] G. Heinrich, *Collider Physics at the Precision Frontier*, *Phys. Rept.* **922** (2021) 1 [2009.00516].

[31] G. Passarino and S. Uccirati, *Algebraic numerical evaluation of Feynman diagrams: Two loop selfenergies*, *Nucl. Phys. B* **629** (2002) 97 [hep-ph/0112004].

[32] M. Caffo, H. Czyz and E. Remiddi, *Numerical evaluation of the general massive 2 loop sunrise selfmass master integrals from differential equations*, *Nucl. Phys. B* **634** (2002) 309 [hep-ph/0203256].

[33] X. Liu and Y.-Q. Ma, *Multiloop corrections for collider processes using auxiliary mass flow*, *Phys. Rev. D* **105** (2022) L051503 [2107.01864].

[34] X. Liu and Y.-Q. Ma, *AMFlow: A Mathematica package for Feynman integrals computation via auxiliary mass flow*, *Comput. Phys. Commun.* **283** (2023) 108565 [2201.11669].

[35] D. Maclagan and B. Sturmfels, *Introduction to Tropical Geometry*, vol. 161 of *Graduate Studies in Mathematics*, American Mathematical Society, Providence, RI (2015).

[36] N. Arkani-Hamed, A. Hillman and S. Mizera, *Feynman polytopes and the tropical geometry of UV and IR divergences*, *Phys. Rev. D* **105** (2022) 125013 [2202.12296].

[37] P. Tourkine, *Tropical Amplitudes*, *Annales Henri Poincare* **18** (2017) 2199 [1309.3551].

[38] L. Eberhardt and S. Mizera, *Unitarity cuts of the worldsheet*, *SciPost Phys.* **14** (2023) 015 [2208.12233].

[39] E. Panzer, *Hepp's bound for Feynman graphs and matroids*, 1908.09820.

[40] M. Borinsky, *Tropical Monte Carlo quadrature for Feynman integrals*, 2008.12310.

[41] I.M. Gel'fand, A.V. Zelevinskii and M.M. Kapranov, *Hypergeometric functions and toral manifolds*, *Functional Analysis and Its Applications* **23** (1989) 94.

[42] I. Gelfand, M. Kapranov and A. Zelevinsky, *Generalized Euler integrals and A-hypergeometric functions*, *Advances in Mathematics* **84** (1990) 255.

[43] R.N. Lee and A.A. Pomeransky, *Critical points and number of master integrals*, *JHEP* **11** (2013) 165 [1308.6676].

[44] S. Weinzierl, *Feynman Integrals*, Springer (1, 2022), 10.1007/978-3-030-99558-4, [2201.03593].

[45] S. Badger, J. Henn, J. Plefka and S. Zoia, *Scattering Amplitudes in Quantum Field Theory*, 2306.05976.

[46] G. 't Hooft and M.J.G. Veltman, *Regularization and Renormalization of Gauge Fields*, *Nucl. Phys. B* **44** (1972) 189.

[47] J.L. Bourjaily et al., *Functions Beyond Multiple Polylogarithms for Precision Collider Physics*, in *Snowmass 2021*, 3, 2022 [2203.07088].

[48] S. Pögel, X. Wang and S. Weinzierl, *The three-loop equal-mass banana integral in ε-factorised form with meromorphic modular forms*, *JHEP* **09** (2022) 062 [2207.12893].

[49] H. Frellesvig and S. Weinzierl, *On ε-factorised bases and pure Feynman integrals*, 2301.02264.

[50] C. Dlapa, J.M. Henn and F.J. Wagner, *An algorithmic approach to finding canonical differential equations for elliptic Feynman integrals*, 2211.16357.

[51] X. Jiang, X. Wang, L.L. Yang and J.-B. Zhao, *ε-factorized differential equations for two-loop non-planar triangle Feynman integrals with elliptic curves*, 2305.13951.

[52] L. Görges, C. Nega, L. Tancredi and F.J. Wagner, *On a procedure to derive ϵ-factorised differential equations beyond polylogarithms*, *JHEP* **07** (2023) 206 [2305.14090].

[53] H. Frellesvig, *On epsilon factorized differential equations for elliptic Feynman integrals*, *JHEP* **03** (2022) 079 [2110.07968].

[54] M. Giroux and A. Pokraka, *Loop-by-loop differential equations for dual (elliptic) Feynman integrals*, *JHEP* **03** (2023) 155 [2210.09898].

[55] W. Magnus, *On the exponential solution of differential equations for a linear operator*, *Communications on Pure and Applied Mathematics* **7** (1954) 649 [https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160070404].

[56] M. Argeri, S. Di Vita, P. Mastrolia, E. Mirabella, J. Schlenk, U. Schubert et al., *Magnus and Dyson Series for Master Integrals*, *JHEP* **03** (2014) 082 [1401.2979].

[57] U. Schubert-Mielnik, *Differential Equations and the Magnus Exponential for multi-loop multi-scale Feynman Integrals*, Ph.D. thesis, Munich, Tech. U., 2016.

[58] A. Arnal, F. Casas and C. Chiralt, *A general formula for the Magnus expansion in terms of iterated integrals of right-nested commutators*, *Journal of Physics Communications* **2** (2018) 035024.

[59] A. Primo, *Cutting Feynman Amplitudes: from Adaptive Integrand Decomposition to Differential Equations on Maximal Cut*, Ph.D. thesis, U. Padua (main), Padua U., 2017.

[60] K.-T. Chen, *Iterated path integrals*, *Bulletin of the American Mathematical Society* **83** (1977) 831 .

[61] C. Duhr, *Mathematical aspects of scattering amplitudes*, in *Theoretical Advanced Study Institute in Elementary Particle Physics: Journeys Through the Precision Frontier: Amplitudes for Colliders*, pp. 419–476, 2015, DOI [1411.7538].

[62] A.B. Goncharov, *Multiple polylogarithms and mixed Tate motives*, math/0103059.

[63] H. Frellesvig, *Generalized Polylogarithms in Maple*, 1806.02883.

[64] E. Panzer, *Algorithms for the symbolic integration of hyperlogarithms with applications to Feynman integrals*, *Comput. Phys. Commun.* **188** (2015) 148 [1403.3385].

[65] T. Gehrmann and E. Remiddi, *Numerical evaluation of harmonic polylogarithms*, *Comput. Phys. Commun.* **141** (2001) 296 [hep-ph/0107173].

[66] D. Maitre, *HPL, a mathematica implementation of the harmonic polylogarithms*, *Comput. Phys. Commun.* **174** (2006) 222 [hep-ph/0507152].

[67] C. Duhr and F. Dulat, *PolyLogTools — polylogs for the masses*, *JHEP* **08** (2019) 135 [1904.07279].

[68] J. Vollinga and S. Weinzierl, *Numerical evaluation of multiple polylogarithms*, *Comput. Phys. Commun.* **167** (2005) 177 [hep-ph/0410259].

[69] S. Buehler and C. Duhr, *CHAPLIN - Complex Harmonic Polylogarithms in Fortran*, *Comput. Phys. Commun.* **185** (2014) 2703 [1106.5739].

[70] L. Naterop, A. Signer and Y. Ulrich, *handyG —Rapid numerical evaluation of generalised polylogarithms in Fortran*, *Comput. Phys. Commun.* **253** (2020) 107165 [1909.01656].

[71] Y. Wang, L.L. Yang and B. Zhou, *FastGPL: a C++ library for fast evaluation of generalized polylogarithms*, 2112.04122.

[72] M. Besier, P. Wasser and S. Weinzierl, *RationalizeRoots: Software Package for the Rationalization of Square Roots*, *Comput. Phys. Commun.* **253** (2020) 107197 [1910.13251].

[73] R. Garani, F. Gasparotto, P. Mastrolia, H.J. Munch, S. Palomares-Ruiz and A. Primo, *Two-photon exchange in leptophilic dark matter scenarios*, *JHEP* **12** (2021) 212 [2105.12116].

[74] W. Bernreuther, R. Bonciani, T. Gehrmann, R. Heinesch, T. Leineweber, P. Mastrolia et al., *Two-loop QCD corrections to the heavy quark form-factors: The Vector contributions*, *Nucl. Phys. B* **706** (2005) 245 [hep-ph/0406046].

[75] A. Primo, G. Sasso, G. Somogyi and F. Tramontano, *Exact Top Yukawa corrections to Higgs boson decay into bottom quarks*, *Phys. Rev. D* **99** (2019) 054013 [1812.07811].

[76] C. Anastasiou, S. Beerli, S. Bucherer, A. Daleo and Z. Kunszt, *Two-loop amplitudes and master integrals for the production of a Higgs boson via a massive quark and a scalar-quark loop*, *JHEP* **01** (2007) 082 [hep-ph/0611236].

[77] H.R.P. Ferguson and D.H. Bailey, *A Polynomial Time, Numerically Stable Integer Relation Algorithm*, *RNRTechnical Report RNR-91-032* (1992) .

[78] E. Remiddi and J.A.M. Vermaseren, *Harmonic polylogarithms*, *Int. J. Mod. Phys. A* **15** (2000) 725 [hep-ph/9905237].

[79] M. Argeri, P. Mastrolia and E. Remiddi, *The Analytic value of the sunrise selfmass with two equal masses and the external invariant equal to the third squared mass*, *Nucl. Phys. B* **631** (2002) 388 [hep-ph/0202123].

[80] R. Bonciani, P. Mastrolia and E. Remiddi, *Master integrals for the two loop QCD virtual corrections to the forward backward asymmetry*, *Nucl. Phys. B* **690** (2004) 138 [hep-ph/0311145].

[81] W. Bernreuther, R. Bonciani, T. Gehrmann, R. Heinesch, P. Mastrolia and E. Remiddi, *Decays of scalar and pseudoscalar Higgs bosons into fermions: Two-loop QCD corrections to the Higgs-quark-antiquark amplitude*, *Phys. Rev. D* **72** (2005) 096002 [hep-ph/0508254].

[82] V. Chestnov, F. Gasparotto, M.K. Mandal, P. Mastrolia, S.J. Matsubara-Heo, H.J. Munch et al., *Macaulay matrix for Feynman integrals: linear relations and intersection numbers*, *JHEP* **09** (2022) 187 [2204.12983].

[83] V. Chestnov, S.J. Matsubara-Heo, H.J. Munch and N. Takayama, *Restrictions of Pfaffian Systems for Feynman Integrals*, 2305.01585.

[84] B. Sturmfels, *Solving algebraic equations in terms of A-hypergeometric series*, *Discrete Mathematics* **210** (2000) 171.

[85] M. Saito, B. Sturmfels and N. Takayama, *Gröbner deformations of hypergeometric differential equations*, Algorithms and computation in mathematics, Springer, Berlin, Germany (jul, 2011), 10.1007/978-3-662-04112-3.

[86] T. Hibi et al., *Gröbner bases*, Springer, Tokyo, Japan, 2013 ed. (jan, 2014), 10.1007/978-4-431-54574-3.

[87] K. Ohara and N. Takayama, *Pfaffian Systems of A-Hypergeometric Systems II—Holonomic Gradient Method*, 1505.02947.

[88] T. Peraro, *FiniteFlow: multivariate functional reconstruction using finite fields and dataflow graphs*, *JHEP* **07** (2019) 031 [1905.08019].

[89] J. Klappert and F. Lange, *Reconstructing rational functions with FireFly*, *Comput. Phys. Commun.* **247** (2020) 106951 [1904.00009].

[90] J. Klappert, S.Y. Klein and F. Lange, *Interpolation of dense and sparse rational functions and other improvements in FireFly*, *Comput. Phys. Commun.* **264** (2021) 107968 [2004.01463].

[91] V. Magerya, *Rational Tracer: a Tool for Faster Rational Function Reconstruction*, 2211.03572.

[92] X. Liu, *Reconstruction of rational functions made simple*, 2306.12262.

[93] S.-J. Matsubara-Heo, S. Mizera and S. Telen, *Four Lectures on Euler Integrals*, 2306.13578.

[94] I. Gelfand, M. Kapranov and A. Zelevinsky, *Discriminants, Resultants, and Multidimensional Determinants*, Modern Birkhäuser Classics, Birkhäuser Boston (2009).

[95] A. Adolphson, *Hypergeometric functions and rings generated by monomials*, *Duke Mathematical Journal* **73** (1994) 269 .

[96] S. Mizera, *Scattering Amplitudes from Intersection Theory*, *Phys. Rev. Lett.* **120** (2018) 141602 [1711.00469].

[97] H. Frellesvig, F. Gasparotto, S. Laporta, M.K. Mandal, P. Mastrolia, L. Mattiazzi et al., *Decomposition of Feynman Integrals on the Maximal Cut by Intersection Numbers*, *JHEP* **05** (2019) 153 [1901.11510].

[98] H. Frellesvig, F. Gasparotto, S. Laporta, M.K. Mandal, P. Mastrolia, L. Mattiazzi et al., *Decomposition of Feynman Integrals by Multivariate Intersection Numbers*, *JHEP* **03** (2021) 027 [2008.04823].

[99] H.A. Frellesvig and L. Mattiazzi, *On the Application of Intersection Theory to Feynman Integrals: the univariate case*, *PoS* **MA2019** (2022) 017 [2102.01576].

[100] S. Mizera, *Status of Intersection Theory and Feynman Integrals*, 2002.10476.

[101] S. Mizera and A. Pokraka, *From Infinity to Four Dimensions: Higher Residue Pairings and Feynman Integrals*, *JHEP* **02** (2020) 159 [1910.11852].

[102] S. Caron-Huot and A. Pokraka, *Duals of Feynman integrals. Part I. Differential equations*, *JHEP* **12** (2021) 045 [2104.06898].

[103] S. Caron-Huot and A. Pokraka, *Duals of Feynman Integrals. Part II. Generalized unitarity*, *JHEP* **04** (2022) 078 [2112.00055].

[104] V. Chestnov, H. Frellesvig, F. Gasparotto, M.K. Mandal and P. Mastrolia, *Intersection numbers from higher-order partial differential equations*, *JHEP* **06** (2023) 131 [2209.01997].

[105] F. Gasparotto, A. Rapakoulias and S. Weinzierl, *Nonperturbative computation of lattice correlation functions by differential equations*, *Phys. Rev. D* **107** (2023) 014502 [2210.16052].

[106] F. Gasparotto, S. Weinzierl and X. Xu, *Real time lattice correlation functions from differential equations*, *JHEP* **06** (2023) 128 [2305.05447].

[107] S. De and A. Pokraka, *Cosmology meets cohomology*, 2308.03753.

[108] F. Gasparotto, *Co-Homology and Intersection Theory for Feynman Integrals*, Ph.D. thesis, U. Padua (main), 2023.

[109] J. Lee, *Introduction to Smooth Manifolds*, Graduate Texts in Mathematics, Springer (2003).

[110] Y. André, F. Baldassarri and M. Cailotto, *De Rham Cohomology of Differential Modules on Algebraic Varieties*, Progress in Mathematics, Springer International Publishing (2020).

[111] K. Aomoto and M. Kita, *Theory of Hypergeometric Functions*, Springer (01, 2011), 10.1007/978-4-431-53938-4.

[112] J. Huh, *The maximum likelihood degree of a very affine variety*, *Compositio Mathematica* **149** (2013) 1245.

[113] S.-J. Matsubara-Heo and S. Telen, *Twisted cohomology and likelihood ideals*, 2301.13579.

[114] T. Bitoun, C. Bogner, R.P. Klausen and E. Panzer, *The number of master integrals as Euler characteristic*, *PoS* **LL2018** (2018) 065 [1809.03399].

[115] M. Saito, B. Sturmfels and N. Takayama, *Hypergeometric polynomials and integer programming*, *Compositio Mathematica* **115** (1999) 231–240.

[116] S.-J. Matsubara-Heo and N. Takayama, *Algorithms for Pfaffian systems and cohomology intersection numbers of hypergeometric integrals, Errata in* *http://www.math.kobe-u.ac.jp/OpenXM/Math/intersection2/*, in *Lecture Notes in Computer Science*, Lecture notes in computer science, pp. 73–84, Springer International Publishing (2020).

[117] M. Kaluba, B. Lorenz and S. Timme, *Polymake. jl: A new interface to polymake*, in *International Congress on Mathematical Software*, pp. 377–385, Springer, 2020.

[118] T. Hibi, K. Nishiyama and N. Takayama, *Pfaffian systems of A-hypergeometric equations I: Bases of twisted cohomology groups*, *Advances in Mathematics* **306** (2017) 303.

[119] T. Regge, *Algebraic Topology Methods in the Theory of Feynman Relativistic Amplitudes*, in *Battelle Rencontres*, pp. 433–458, 1968.

[120] E. Nasrollahpoursamami, *Periods of Feynman diagrams*, Ph.D. thesis, Caltech, 2017. 10.7907/Z9GX48MR.

[121] A. Pal and K. Ray, *Conformal integrals in four dimensions*, *JHEP* **10** (2022) 087 [2109.09379].

[122] A. Pal and K. Ray, *Conformal integrals in various dimensions and Clifford groups*, 2303.17326.

[123] B. Ananthanarayan, S. Banik, S. Bera and S. Datta, *FeynGKZ: A Mathematica package for solving Feynman integrals using GKZ hypergeometric systems*, *Comput. Phys. Commun.* **287** (2023) 108699 [2211.01285].

[124] T.-F. Feng, H.-B. Zhang and C.-H. Chang, *Feynman integrals of Grassmannians*, *Phys. Rev. D* **106** (2022) 116025 [2206.04224].

[125] T.-F. Feng, H.-B. Zhang, Y.-Q. Dong and Y. Zhou, *GKZ-system of the 2-loop self energy with 4 propagators*, 2209.15194.

[126] H.-B. Zhang and T.-F. Feng, *GKZ hypergeometric systems of the three-loop vacuum Feynman integrals*, 2303.02795.

[127] U. Walther, *On Feynman graphs, matroids, and GKZ-systems*, *Letters in Mathematical Physics* **112** (2022) 120.

[128] C. Dlapa, M. Helmer, G. Papathanasiou and F. Tellander, *Symbol Alphabets from the Landau Singular Locus*, 2304.02629.

[129] H.J. Munch, *Feynman Integral Relations from GKZ Hypergeometric Systems*, *PoS* **LL2022** (2022) 042 [2207.09780].

[130] A. Klemm, C. Nega and R. Safari, *The l-loop Banana Amplitude from GKZ Systems and relative Calabi-Yau Periods*, *JHEP* **04** (2020) 088 [1912.06201].

[131] K. Bönisch, F. Fischbach, A. Klemm, C. Nega and R. Safari, *Analytic structure of all loop banana integrals*, *JHEP* **05** (2021) 066 [2008.10574].

[132] K. Bönisch, C. Duhr, F. Fischbach, A. Klemm and C. Nega, *Feynman integrals in dimensional regularization and extensions of Calabi-Yau motives*, *JHEP* **09** (2022) 156 [2108.05310].

[133] P. Vanhove, *Feynman integrals, toric geometry and mirror symmetry*, in *KMPB Conference: Elliptic Integrals, Elliptic Functions and Modular Forms in Quantum Field Theory*, pp. 415–458, 2019, DOI [1807.11466].

[134] P. Vanhove, *Differential Equations for Feynman Integrals*, in *International Symposium on Symbolic and Algebraic Computation*, 7, 2021, DOI.

[135] F. Tellander and M. Helmer, *Cohen-Macaulay Property of Feynman Integrals*, *Commun. Math. Phys.* **399** (2023) 1021 [2108.01410].

[136] R.P. Klausen, *Hypergeometric Series Representations of Feynman Integrals by GKZ Hypergeometric Systems*, *JHEP* **04** (2020) 121 [1910.08651].

[137] R.P. Klausen, *Kinematic singularities of Feynman integrals and principal A-determinants*, *JHEP* **02** (2022) 004 [2109.07584].

[138] R.P. Klausen, *Hypergeometric Feynman integrals*, Ph.D. thesis, Mainz U., 2023. 2302.13184. 10.25358/openscience-8527.

[139] J. Henn, E. Pratt, A.-L. Sattelberger and S. Zoia, *D-Module Techniques for Solving Differential Equations in the Context of Feynman Integrals*, 2303.11105.

[140] T.-F. Feng, C.-H. Chang, J.-B. Chen and H.-B. Zhang, *GKZ-hypergeometric systems for Feynman integrals*, *Nucl. Phys. B* **953** (2020) 114952 [1912.01726].

[141] L. de la Cruz, *Feynman integrals as A-hypergeometric functions*, *JHEP* **12** (2019) 123 [1907.00507].

[142] D. Agostini, C. Fevola, A.-L. Sattelberger and S. Telen, *Vector Spaces of Generalized Euler Integrals*, 2208.08967.

[143] S. Srednyak and V. Khachatryan, *Vanishing cycles and analysis of singularities of Feynman diagrams*, 2307.11302.

[144] E.R. Speer, *Generalized Feynman Amplitudes*, Princeton University Press (1969).

[145] E.R. Speer and M.J. Westwater, *Generic Feynman amplitudes*, *Annales de l'I.H.P. Physique théorique* **14** (1971) 1.

[146] J.M. Henn, *Lectures on differential equations for Feynman integrals*, *J. Phys. A* **48** (2015) 153001 [1412.2296].

[147] V. Chestnov, S.J. Matsubara-Heo, H.J. Munch and N. Takayama, *Macaulay Matrix for Feynman Integrals: Programs and Data*, *http://www.math.kobe-u.ac.jp/OpenXM/Math/amp-MM* (2023) .

[148] T. Oaku, *Algorithms for the b-function and D-modules associated with a polynomial*, *Journal of Pure and Applied Algebra* **117-118** (1997) 495.

[149] H. Nakayama and K. Nishiyama, *An algorithm of computing inhomogeneous differential equations for definite integrals*, in *Mathematical Software–ICMS 2010: Third International Congress on Mathematical Software, Kobe, Japan, September 13-17, 2010. Proceedings 3*, pp. 221–232, Springer, 2010.

[150] P. Deligne, *Equations différentielles à points singuliers réguliers*, Lecture notes in mathematics, Springer-Verlag (1970).

[151] T. Hosgood, *Regular singular differential equations*, English translation of *Equations différentielles à points singuliers réguliers* by P. Deligne, *https://labs.thosgood.com/translations/978-3-540-05190-9.pdf* (2023) .

[152] J. Moser, *The order of a singularity in Fuchs' theory*, *Mathematische Zeitschrift* **71** (1959) 379.

[153] M.A. Barkatou, *An algorithm to compute the exponential part of a formal fundamental matrix solution of a linear differential system*, *Appl. Algebra Eng., Commun. Comput.* **8** (1997) 1–23.

[154] M.A. Barkatou, M. Jaroschek and S.S. Maddah, *Formal solutions of completely integrable Pfaffian systems with normal crossings*, Journal of Symbolic Computation **81** (2017) 41.

[155] O. Gituliar and V. Magerya, *Fuchsia: a tool for reducing differential equations for Feynman master integrals to epsilon form*, *Comput. Phys. Commun.* **219** (2017) 329 [1701.04269].

[156] M.A. Barkatou, E. Pflügel and F. Stan, *ISOLDE: a Maple package for systems of linear functional equations*, *ACM Communications in Computer Algebra* **46** (2013) 157.

[157] Y. Haraoka, *Linear Differential Equations in the Complex Domain: From Classical Theory to Forefront*, Lecture Notes in Mathematics, Springer International Publishing (2020).

[158] N. Takayama, *Propagation of singularities of solutions of the Euler-Darboux equation and a global structure of the space of holonomic solutions I*, *Funkcialaj Ekvacioj* **35** (1992) 343.

[159] M. Beneke and V.A. Smirnov, *Asymptotic expansion of Feynman integrals near threshold*, *Nucl. Phys. B* **522** (1998) 321 [hep-ph/9711391].

[160] V.A. Smirnov, *Applied asymptotic expansions in momenta and masses*, *Springer Tracts Mod. Phys.* **177** (2002) 1.

[161] B. Jantzen, *Foundation and generalization of the expansion by regions*, *JHEP* **12** (2011) 076 [1111.2589].

[162] F. Moriello, *Generalised power series expansions for the elliptic planar families of Higgs + jet production at two loops*, *JHEP* **01** (2020) 150 [1907.13234].

[163] M. Egner, M. Fael, K. Schönwald and M. Steinhauser, *Revisiting semileptonic B meson decays at next-to-next-to-leading order*, 2308.01346.

[164] R.N. Lee, A.V. Smirnov and V.A. Smirnov, *Solving differential equations for Feynman integrals by expansions near singular points*, *JHEP* **03** (2018) 008 [1709.07525].

[165] R.N. Lee, A.V. Smirnov and V.A. Smirnov, *Evaluating 'elliptic' master integrals at special kinematic values: using differential equations and their solutions via expansions near singular points*, *JHEP* **07** (2018) 102 [1805.00227].

[166] K. Kudashkin, K. Melnikov and C. Wever, *Two-loop amplitudes for processes $gg \to Hg, qg \to Hq$ and $q\bar{q} \to Hg$ at large Higgs transverse momentum*, JHEP **02** (2018) 135 [1712.06549].

[167] B. Ananthanarayan, S. Bera, S. Friot, O. Marichev and T. Pathak, *On the evaluation of the Appell F2 double hypergeometric function*, Comput. Phys. Commun. **284** (2023) 108589 [2111.05798].

[168] B. Ananthanarayan, S. Bera, S. Friot and T. Pathak, *Olsson.wl : a Mathematica package for the computation of linear transformations of multivariable hypergeometric functions*, 2201.01189.

[169] V. Chestnov, S.J. Matsubara-Heo, H.J. Munch and N. Takayama, *Restrictions of Pfaffian Systems for Feynman Integrals — Data and Programs*, *http://www.math.kobe-u.ac.jp/OpenXM/Math/amp-Restriction* (2022) .

[170] J.M. Henn and V.A. Smirnov, *Analytic results for two-loop master integrals for Bhabha scattering I*, JHEP **11** (2013) 041 [1307.4083].

[171] C. Duhr, V.A. Smirnov and L. Tancredi, *Analytic results for two-loop planar master integrals for Bhabha scattering*, JHEP **09** (2021) 120 [2108.03828].

[172] C. Meyer, *Algorithmic transformation of multi-loop master integrals to a canonical basis with CANONICA*, Comput. Phys. Commun. **222** (2018) 295 [1705.06252].

[173] M. Hidding, *DiffExp, a Mathematica package for computing Feynman integrals in terms of one-dimensional series expansions*, Comput. Phys. Commun. **269** (2021) 108125 [2006.05510].

[174] I. Dubovyk, A. Freitas, J. Gluza, K. Grzanka, M. Hidding and J. Usovitsch, *Evaluation of multiloop multiscale Feynman integrals for precision physics*, Phys. Rev. D **106** (2022) L111301 [2201.02576].

[175] T. Armadillo, R. Bonciani, S. Devoto, N. Rana and A. Vicini, *Evaluation of Feynman integrals with arbitrary complex masses via series expansions*, Comput. Phys. Commun. **282** (2023) 108545 [2205.03345].

[176] S. Borowka, G. Heinrich, S. Jahn, S.P. Jones, M. Kerner, J. Schlenk et al., *pySecDec: a toolbox for the numerical evaluation of multi-scale integrals*, Comput. Phys. Commun. **222** (2018) 313 [1703.09692].

[177] G. Heinrich, S.P. Jones, M. Kerner, V. Magerya, A. Olsson and J. Schlenk, *Numerical Scattering Amplitudes with pySecDec*, 2305.19768.

[178] Z. Capatti, V. Hirschi, D. Kermanschah and B. Ruijl, *Loop-Tree Duality for Multiloop Numerical Integration*, *Phys. Rev. Lett.* **123** (2019) 151602 [1906.06138].

[179] Z. Capatti, V. Hirschi, D. Kermanschah, A. Pelloni and B. Ruijl, *Manifestly Causal Loop-Tree Duality*, 2009.05509.

[180] P. Lairez, E. Pichon-Pharabod and P. Vanhove, *Effective homology and periods of complex projective hypersurfaces*, 2306.05263.

[181] R. Pittau and B. Webber, *Direct numerical evaluation of multi-loop integrals without contour deformation*, *Eur. Phys. J. C* **82** (2022) 55 [2110.12885].

[182] M. Zeng, *Feynman Integrals from Positivity Constraints*, 2303.15624.

[183] R. Jinno, G. Kälin, Z. Liu and H. Rubira, *Machine learning Post-Minkowskian integrals*, *JHEP* **07** (2023) 181 [2209.01091].

[184] M. Borinsky, A.-L. Sattelberger, B. Sturmfels and S. Telen, *Bayesian Integrals on Toric Varieties*, *SIAM Journal on Applied Algebra and Geometry* **7** (2023) 77 [2204.06414].

[185] M. Borinsky, H.J. Munch and F. Tellander, *Tropical Feynman integration in the Minkowski regime*, *Computer Physics Communications* **292** (2023) 108874.

[186] G. Ziegler, *Lectures on Polytopes*, Graduate Texts in Mathematics, Springer New York (2012).

[187] A. Favorito, *Tropical Feynman Period Integration*, Master's thesis, ETH Zürich, 2023.

[188] S. Mizera and S. Telen, *Landau discriminants*, *JHEP* **08** (2022) 200 [2109.08036].

[189] H.S. Hannesdottir and S. Mizera, *What is the $i\varepsilon$ for the S-matrix?*, SpringerBriefs in Physics, Springer (1, 2023), 10.1007/978-3-031-18258-7, [2204.02988].

[190] A. von Manteuffel, E. Panzer and R.M. Schabinger, *A quasi-finite basis for multi-loop Feynman integrals*, *JHEP* **02** (2015) 120 [1411.7392].

[191] C. Berkesch, J. Forsgård and M. Passare, *Euler-Mellin integrals and A-hypergeometric functions*, *Michigan Mathematical Journal* **63** (2014) 101.

[192] L. Nilsson and M. Passare, *Mellin transforms of multivariate rational functions*, *Journal of Geometric Analysis* **23** (2013) 24.

[193] N. Arkani-Hamed, S. He and T. Lam, *Stringy canonical forms*, *JHEP* **02** (2021) 069 [1912.08707].

[194] M. Aguiar and F. Ardila, *Hopf monoids and generalized permutahedra*, 1709.07504.

[195] A. Postnikov, *Permutohedra, associahedra, and beyond*, *International Mathematics Research Notices* **2009** (2009) 1026.

[196] K. Schultka, *Toric geometry and regularization of Feynman integrals*, 1806.01086.

[197] F. Brown, *Feynman amplitudes, coaction principle, and cosmic Galois group*, *Commun. Num. Theor. Phys.* **11** (2017) 453 [1512.06409].

[198] G. Guennebaud, B. Jacob et al., "Eigen v3." `http://eigen.tuxfamily.org`, 2010.

[199] R. Chandra, L. Dagum, D. Kohr, R. Menon, D. Maydan and J. McDonald, *Parallel programming in OpenMP*, Morgan Kaufmann (2001).

[200] D. Blackman and S. Vigna, *Scrambled linear pseudorandom number generators*, *ACM Trans. Math. Software* **47** (2021) Art. 36, 32.

[201] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic et al., *Jupyter notebooks – a publishing format for reproducible computational workflows*, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, eds., pp. 87 – 90, IOS Press, 2016.

[202] S. Di Vita, S. Laporta, P. Mastrolia, A. Primo and U. Schubert, *Master integrals for the NNLO virtual corrections to μe scattering in QED: the non-planar graphs*, *JHEP* **09** (2018) 016 [1806.08241].

[203] J.M. Henn, *Duality between Wilson loops and gluon amplitudes*, *Fortsch. Phys.* **57** (2009) 729 [0903.0522].

[204] L. Corcoran, F. Loebbert and J. Miczajka, *Yangian Ward identities for fishnet four-point integrals*, *JHEP* **04** (2022) 131 [2112.06928].

[205] F. Loebbert, *Integrability for Feynman Integrals*, in *34th International Colloquium on Group Theoretical Methods in Physics Strasbourg*, 2022 [2212.09636].

[206] P. Lairez and P. Vanhove, *Algorithms for minimal Picard–Fuchs operators of Feynman integrals*, *Lett. Math. Phys.* **113** (2023) 37 [2209.10962].

[207] C.F. Doran, A. Harder, E. Pichon-Pharabod and P. Vanhove, *Motivic geometry of two-loop Feynman integrals*, 2302.14840.

[208] A.-L. Sattelberger and B. Sturmfels, *D-modules and holonomic functions*, 1910.01395.

[209] S. Mano and N. Takayama, *Algebraic algorithm for direct sampling from toric models*, 2110.14992.

[210] S.J. Matsubara-Heo, *Seminar talk: Restrictions of Pfaffian system and hypergeometric system of contingency table*, *https://www.math.uni-leipzig.de/cms2/en/cal/detail/2846* (2022) .

[211] J.M. Henn, J. Lim and W.J. Torres Bobadilla, *First look at the evaluation of three-loop non-planar Feynman diagrams for Higgs plus jet production*, *JHEP* **05** (2023) 026 [2302.12776].

[212] Risa/Asir, *Annihilation ideal software*, *https://www.rs.tus.ac.jp/ nabeshima/softwares.html* (2022) .

[213] A. Adolphson and S. Sperber, *A-hypergeometric systems that come from geometry*, *Proceedings of the American Mathematical Society* **140** (2012) 2033.