

Towards Light Quark Mass Effects in Higgs Production and Decay at Next-to-Next-to-Leading Order

Von der Fakultät für Mathematik, Informatik und
Naturwissenschaften der RWTH Aachen University zur
Erlangung des akademischen Grades eines Doktors der
Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Physiker
Mario Prausa

aus Heilbronn

Berichter: Prof. Dr.rer.nat. Robert Harlander
Prof. Dr.rer.nat. Michał Czakon

Tag der mündlichen Prüfung: 25.04.2018

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek verfügbar.

Zusammenfassung

Obwohl das Standardmodell der Teilchenphysik bislang den genauesten Tests stand gehalten hat, hofft man besonders im Higgs-Sektor Abweichungen zwischen experimentellen Daten und Hochpräzisions-Rechnungen zu finden. Derartige Abweichungen wären wegweisend für theoretische Arbeiten die auf eine phänomenologisch vollständigere Quantenfeldtheorie hinarbeiten, die z.B. auch Dunkle Materie oder die Gravitation beschreibt.

Wir betrachten zwei hochrelevante Prozesse des Higgs-Sektors am Large Hadron Collider am CERN: Die Higgs-Produktion über die Fusion zweier Gluonen und den Zerfall eines Higgs-Bosons in zwei Photonen. Bereits in führender Ordnung benötigen beide Prozesse eine Einschleifenrechnung. Dennoch ist es gelungen, den Wirkungsquerschnitt für diesen Higgs-Produktionskanal sowie die partielle Zerfallsrate in sehr hoher Genauigkeit zu bestimmen. Allerdings wurde diese Rechnung im Grenzfall schwerer Top-Quarks durchgeführt, wobei die Effekte leichter Quarks vernachlässigt wurden. Die Vernachlässigung dieser Effekte erhöht die Unsicherheit bei der Vorhersage des Gluon-Fusions-Wirkungsquerschnitts um etwa einen Prozent.

Für beide Prozesse konnte die exakte Massenabhängigkeit bisher nur zur nächstführenden Ordnung bestimmt werden. In dieser Arbeit stellen wir Techniken vor, die auch die Vorhersage zur übernächsten Ordnung verbessern.

Wir zeigen, dass ein Standardsetup für Mehrschleifenrechnungen einen guten Ansatzpunkt liefert um die Beiträge der Effekte leichter Quarks zu bestimmen. Zwar ist es nicht kompliziert, die Feynman-Amplitude durch skalare Integrale auszudrücken; dennoch erwies sich die Reduktion dieser Integrale zu einem kleinen Satz sogenannter Masterintegrale als nicht trivial.

Den größten Teil dieser Arbeit nimmt jedoch die Berechnung der Masterintegrale ein. Jüngste Fortschritte bei der Berechnungsmethode mittels Differentialgleichungen haben uns dazu motiviert, ein Computerprogramm zu entwickeln, um eine kanonische Basis für Masterintegrale zu finden. In dieser neuen Basis ist das Lösen der Masterintegrale dann trivial. Mittels dieser Technik konnten wir ungefähr die Hälfte aller Masterintegrale für den Higgs-Zerfall in zwei Photonen berechnen.

Für die übrigen Masterintegrale verfolgen wir einen anderen Ansatz. Hierfür scheint eine Berechnung der exakten Quarkmassenabhängigkeit im Moment nicht durchführbar, weshalb wir eine Entwicklung in einer kleinen Quarkmasse verglichen mit der Higgs-Masse anstreben. Diese Näherung beruht auf einer Kombination aus der Differentialgleichungsmethode und Mellin-Barnes-Techniken. Hierbei fanden wir eine neue Methode um Mellin-Barnes-Darstellungen zu konstruieren, welche bereits bestehende Ansätze gut ergänzt.

Es ist unsere Hoffnung, dass die Techniken, die in dieser Arbeit diskutiert werden, ausreichend sind, um für beide Prozesse die Effekte leichter Quarks in der übernächsten Ordnung berechnen zu können. Vollständige Ergebnisse hierfür stehen allerdings noch aus. Dennoch können die ausgearbeiteten Techniken auch für andere Probleme von Nutzen sein.

Abstract

Even though the standard model of particle physics withstands the most precise tests so far, one hopes to find inconsistencies of experimental data from high precision calculations especially in the Higgs sector. Such deviations would guide the direction of theoretical efforts towards a quantum field theory that is phenomenologically more complete, including e.g. dark matter or gravity.

We consider two highly relevant processes of the Higgs physics measured at the Large Hadron Collider at CERN, i.e. Higgs production via gluon fusion and the decay of a Higgs boson into two photons. Already at leading-order the contributions to both processes require a one-loop calculation. Nevertheless, the cross section of this Higgs production mode as well as the partial Higgs decay rate were already calculated to very high precision. However this calculation was done in the limit of a heavy top-quark and neglects effects from lighter quarks. This approximation increases the uncertainty of the gluon fusion cross section by about one percent.

For both processes, a calculation with an exact quark-mass dependence is only known up to next-to-leading order. In this thesis, we discuss techniques to improve the next-to-next-to-leading order predictions.

We show that a standard setup for multiloop calculations can provide a useful starting point for the computation of light quark mass effects. While expressing the Feynman amplitudes in terms of scalar integrals is straightforward, the reduction of those integrals to a small set of so-called master integrals is highly non-trivial.

The major part of this thesis deals with calculation methods for master integrals. Recent progress in the method of differential equations motivated us to develop a tool for the construction of a so-called canonical basis of master integrals. Within this basis, the calculation of the master integrals becomes trivial. Using this technique, we were able to solve about half of the master integrals for the Higgs decay into two photons.

For the remaining integrals, we use a different approach. Here, a calculation of the exact quark-mass dependence is not feasible at the moment. Therefore, we aim at an expansion in a small quark mass compared to the Higgs-boson mass. This expansion requires a combination of the method of differential equations and Mellin-Barnes techniques. In this process, we found a novel method for the construction of Mellin-Barnes representations, which complements existing approaches.

We hope that the techniques discussed in this thesis are sufficient to obtain results for the light quark effects at the next-to-next-to-leading order of both processes. A complete calculation is still work in progress. Nevertheless, the techniques developed during the thesis can still be useful also for other problems.

List of Publications

- [1] M. Prausa, *epsilon: A tool to find a canonical basis of master integrals*, *Comput. Phys. Commun.* **219** (2017) 361–376, [arXiv:1701.00725 \[hep-ph\]](#).
- [2] M. Prausa, *Mellin-Barnes meets Method of Brackets: a novel approach to Mellin-Barnes representations of Feynman integrals*, *Eur. Phys. J.* **C77** no. 9, (2017) 594, [arXiv:1706.09852 \[hep-ph\]](#).

Contents

| | | |
|----------|--|-----------|
| 1 | The Higgs Boson in the Standard Model | 1 |
| 1.1 | Higgs Production via Gluon Fusion | 2 |
| 1.2 | Higgs Decay into Two Photons | 4 |
| 1.3 | Outline of the Thesis | 5 |
| 2 | From Feynman Diagrams To Master Integrals | 7 |
| 2.1 | Feynman Diagrams | 7 |
| 2.2 | Tensor Reduction | 9 |
| 2.3 | Two-Scale Approximation | 10 |
| 2.4 | Topologies | 11 |
| 2.5 | The Kinematic Variable | 13 |
| 2.6 | Reduction to Master Integrals | 15 |
| 2.6.1 | Integration-by-Parts and Lorentz-invariance Identities | 15 |
| 2.6.2 | Sector Relations | 16 |
| 2.6.3 | The Reduction Database | 17 |
| 3 | Calculation Methods for Master Integrals | 19 |
| 3.1 | Goncharov Polylogarithms | 19 |
| 3.1.1 | A Minimal Basis of GPLs | 20 |
| 3.1.2 | Series Expansion | 22 |
| 3.1.3 | Analytic Continuation | 27 |
| 3.2 | The Method of Differential Equations | 28 |
| 3.3 | A Canonical Basis for Master Integrals | 30 |
| 3.3.1 | Definitions | 30 |
| 3.3.2 | Utilizing the Explicit x -Dependence | 31 |
| 3.3.3 | Overview of Lee's algorithm | 33 |
| 3.3.4 | Solving Canonical Master Integrals | 38 |
| 3.4 | Expansion by Subgraphs | 39 |
| 3.5 | Feynman Integral Parametrizations | 42 |
| 3.5.1 | Dealing with Numerators | 44 |
| 3.6 | Evaluation by Mellin-Barnes Representations | 45 |
| 3.6.1 | Resolving Singularities | 47 |
| 3.6.2 | Asymptotic Expansion | 51 |
| 3.7 | Mellin-Barnes meets Method of Brackets | 54 |
| 3.7.1 | The Modified Method of Brackets | 54 |
| 3.7.2 | Optimization Procedure | 58 |
| 3.7.3 | Example | 58 |

| | | |
|----------|--|------------|
| 4 | Master Integrals | 61 |
| 4.1 | Master Integrals and Goncharov Polylogarithms | 61 |
| 4.1.1 | Canonical Diagonal Blocks | 61 |
| 4.1.2 | Boundary Conditions | 64 |
| 4.2 | Differential Equation Expansion | 64 |
| 4.2.1 | Expansion of the System of Differential Equations in Small x | 65 |
| 4.2.2 | An Ansatz for Feynman Integrals | 67 |
| 4.2.3 | Relevant Scaling Regions | 67 |
| 4.2.4 | Reduction | 68 |
| 4.2.5 | Master Coefficients | 69 |
| 5 | Some Results | 71 |
| 6 | Conclusion and Outlook | 79 |
| | Appendices | 81 |
| A | Notation and Conventions | 81 |
| A.1 | Diagrammatic Representation of Feynman Integrals | 82 |
| A.2 | Pseudo-Code Conventions | 83 |
| B | Feynman Rules | 85 |
| C | One-Scale Master Integrals | 87 |
| C.1 | The Simple Ones | 87 |
| C.2 | Three-Loop Bubble with Four Massive Lines | 88 |
| C.3 | Non-Planar Vertex Diagram | 89 |
| D | The tool epsilon | 91 |
| D.1 | Usage | 91 |
| D.1.1 | Installation Guide on Linux Systems | 91 |
| D.1.2 | Input/Output Format | 92 |
| D.1.3 | Usage of <code>epsilon-prepare</code> | 92 |
| D.1.4 | Usage of <code>epsilon</code> | 93 |
| D.1.5 | Using Field Extensions | 95 |
| D.1.6 | Usage of <code>EpsilonTools.m</code> | 96 |
| D.2 | Transformations | 97 |
| D.2.1 | Balances | 97 |
| D.2.2 | Fuchsification for Off-Diagonal Blocks | 100 |
| D.2.3 | ϵ -Factorization | 101 |
| E | Common Subexpressions | 103 |
| | Bibliography | 109 |

The Higgs Boson in the Standard Model

1

The standard model of particle physics is one of the greatest success stories of theoretical physics. For example, one of its great accomplishments is the theoretical prediction of the W and Z bosons by Glashow, Weinberg and Salam in the 1960s [3–5]. These bosons were experimentally confirmed at the Super Proton Synchrotron at CERN in 1983 [6–9]. In fact, until today all measurements agree with the predictions of the standard model up to such great accuracy, that it has become increasingly difficult to find the limits of the standard model. This is a dilemma since the standard model itself is phenomenological in nature and leaves many problems unaddressed, e.g. the matter-antimatter asymmetry or the hierarchy problem.

If a flaw in the predictions of the standard model would be found, this could be a valuable hint towards a more complete physical picture beyond the standard model. However, to identify such deviations, one first needs to make more exact theoretical predictions. This requires the evaluation of higher order corrections for the processes described by the standard model (SM). The SM consists of two parts that describe three of the fundamental forces of nature: the strong force, the electromagnetic force, and the weak force.

The part of the SM that deals with the strong force is known as quantum chromodynamics (QCD). QCD is a non-abelian gauge theory with an $SU(3)$ gauge group. As a first step towards this theory, Gell-Mann and Ne’eman suggested in 1961 that recently discovered hadrons (four Δ resonances, six hyperons and four K mesons) are members of specific representations of $SU(3)$ [10, 11]. Shortly after, in 1964, Gell-Mann and Zweig proposed that the hadrons and mesons are in fact bound states of particles, which form an $SU(3)$ triplet [12–14]. These new fundamental particles were dubbed “quarks” as an homage to James Joyce’s book “Finnegans Wake”. This quark model could explain all hadrons and mesons known at the time as compound states of three quark flavors, the up-, down- and strange-quark. Since the quarks all have different masses, the $SU(3)$ flavor symmetry has to be broken.

A major problem of this early quark model was the Ω^- baryon discovered in 1964 [15]. This baryon is made of three strange-quarks. Since its ground state wave function should be symmetrical if two quarks are interchanged, it violates the Pauli principle. This contradiction was resolved in 1972 by Fritzsche and Gell-Mann who introduced a new conserved quantum number called color [16]. This quantum number originates from an unbroken $SU(3)$ gauge symmetry. While quarks carry a color index of the fundamental representation, the gluons, the gauge bosons of this theory, live in the adjoint representation of $SU(3)$. Hence, the Pauli principle is recovered by considering the wave functions of baryons to be antisymmetric in the fundamental color indices. The modern version of the quark model includes not only three, but six different quark flavors, the up-, down-, strange-, charm-, top- and bottom-quarks.

In 1973, Gross and Wilczek [17] and independently Politzer [18] discovered another fascinating property of this theory, which was rewarded with a Nobel prize in 2004: Asymptotic freedom. The strength of the interaction between particles actually decreases if one goes to higher energies, i.e. the particles seem to be free at infinite energy. In contrast, at low energies (or large distances), the QCD interaction becomes strong, leading to an effect called confinement, which is responsible for the fact that colored particles cannot be isolated.

The other piece of the standard model is the theory of electroweak interaction.

In 1961, Glashow discovered a way to unify quantum electrodynamics with the weak interaction, the force that is responsible, among other things, for radioactive decay [3]. Even though Glashow's model, an explicitly broken $SU(2) \times U(1)$ gauge theory, already explained the weak interaction by massive vector bosons mediating the force, it missed a descriptions where these masses stem from.

Spontaneous symmetry breaking provides a mechanism to turn gauge bosons massive without explicitly breaking gauge invariance. This mechanism was discovered 1964 independently by Englert and Brout [19], Higgs [20], and Guralnik, Hagen and Kibble [21]. It was incorporated into Glashow's theory in 1967 by Weinberg [4] and Salam [5], casting the theory into what we nowadays call the Glashow-Weinberg-Salam model. This theory has been proven to be renormalizable in 1971 by 't Hooft [22].

Besides providing mass terms for some vector bosons (and also for fermions) of the SM, the mechanism of spontaneous gauge symmetry breaking also predicted a massive scalar boson we call Higgs boson. However, the experimental discovery of this particular boson proved to be a long ordeal. In 2012, a new particle with a mass of about 125 GeV was discovered by the two major experiments ATLAS and CMS of the Large Hadron Collider (LHC) at CERN [23, 24]. The properties of this new particles are consistent with the best existing predictions of SM Higgs physics.

Nonetheless, to ensure that this is indeed the famed Higgs boson, not only the experimental side has to obtain more and more precise measurements. It is also the responsibility of the theorists to improve the theoretical predictions and reduce the theoretical errors. This, in turn, might help to uncover inconsistencies between measurements and predictions, which may lead to a better understanding of physics that are not covered by the SM. In particular, the SM does not incorporate a theory of gravity and lacks an explanation for dark matter deduced from cosmological observations. It is conceivable that dark matter particles obtain their masses from the same mechanism behind the SM mass terms. If this is true, it is likely that measurements in the Higgs sector will differ from SM expectations.

In this thesis, we develop the technical tools to improve the precision of the predictions for two very important processes with similar kinematics, the fusion of two gluons to a Higgs boson and the decay of a Higgs boson into two photons.

1.1 Higgs Production via Gluon Fusion

At proton-proton colliders there are four main production modes for a Higgs boson within the SM. These are gluon fusion, top associated Higgs production, Higgs strahlung and vector boson fusion, for which representative leading order Feynman diagrams are depicted in fig. 1.1. Fig. 1.2 shows calculations for cross sections of the most important Higgs production channels. In the considered range for the center-of-mass energy, gluon fusion exceeds the cross-section value of any other process by at least one order of magnitude. Although even the leading order contribution to the gluon fusion process $gg \rightarrow H$ contains a fermionic loop (see fig. 1.1(a)), it is the dominant production channel for Higgs bosons at particle colliders with a high gluon luminosity like the LHC at CERN.

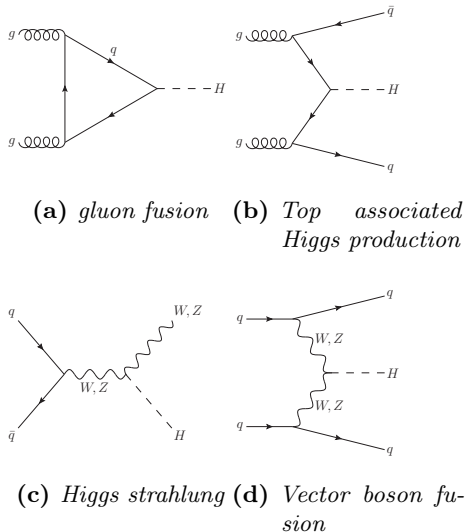


Fig. 1.1: Higgs boson production modes.

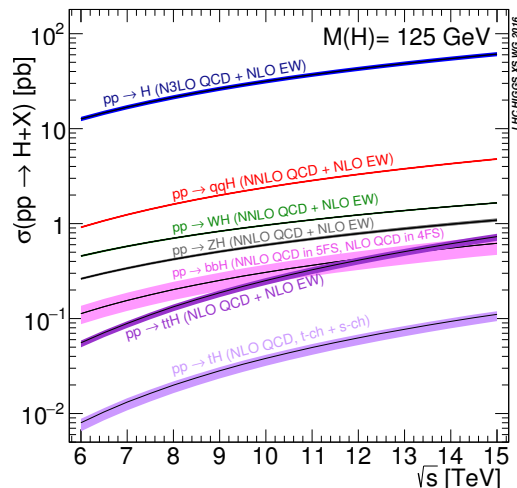


Fig. 1.2: Higgs production cross sections as function of the center of mass energy \sqrt{s} [25].

The leading order (LO) contributions to the process $gg \rightarrow H$ were first calculated in 1978 by Georgi, Glashow, Machacek, and Nanopoulos [26] in the heavy top limit. A result with a full quark-mass dependence followed 1980 by Rizzo [27]. The next-to-leading order (NLO) QCD corrections appeared 1991 in two independent papers, one by Djouadi, Spira, and Zerwas [28] and the other by Dawson [29], where again the mass of the top-quark was considered infinite. The first two-loop calculation with a full quark-mass dependence was carried out in 1993 by Spira, Djouadi, Graudenz and Zerwas [30, 31], where the result was given in terms of one-dimensional integrals. These one-dimensional integrals could be evaluated analytically in 2002 by Harlander and Kant [32] via a technique called “Expansion and Inversion”. A different approach was taken by Anastasiou, Beerli, Bucherer, Daleo and Kunszt [33, 34] who calculated the NLO Feynman diagrams via the method of differential equations in 2006. A further computation was performed in 2007 by Aglietti, Bonciani, Degrandi and Vicini [35]. All these calculations are in perfect agreement. The NLO corrections increase the cross section almost by a factor of two, which stresses the importance of higher order corrections for this process. The next-to-next-to-leading order (NNLO) corrections have been calculated in 2002 and 2003, but only in the framework of an effective field theory (EFT), where the top quark is considered to be infinitely heavy [36–38]. The precision of the NNLO corrections was further improved in 2009 when finite top-quark mass effects were included in terms of a power series in the inverse quark mass. In 2015, even the next-to-next-to-next-to-leading order (N³LO) QCD corrections were calculated in this EFT, albeit in a threshold expansion up to order 37 [39].

All these building blocks were recently combined in order to obtain the so far most precise value of

$$\sigma = 48.58 \text{ pb}^{+2.22 \text{ pb}(+4.56\%)}_{-3.27 \text{ pb}(-6.72\%)} (\text{theory}) \pm 1.56 \text{ pb}(3.20\%) (\text{PDF}+\alpha_s) \quad (1.1)$$

for the Higgs boson cross-section via gluon-fusion [40], where a Higgs mass of 125 GeV and a center of mass energy of 13 TeV was used. The first uncertainty contains all theoretical error estimates¹. The parton distribution functions and the strong coupling constant α_s also come with an uncertainty of their own, i.e. the second error term in (1.1). The influence of light quark mass effects at LO and NLO to this cross section is also given in Ref. [40]. Tab. 1.1 shows that a non-zero bottom-quark

¹The theoretical uncertainty is the linear sum of different sources of error. A further uncertainty of $\pm 2.2 \text{ pb}(4.5\%)$ is given in Ref. [25] for which all sources of error were considered as one-sigma ranges.

| | LO | NLO |
|------------------------------------|----------|----------|
| σ_{EFT} | 15.05 pb | 34.66 pb |
| $R_{\text{LO}}\sigma_{\text{EFT}}$ | 16.00 pb | 36.84 pb |
| $\sigma_{ex;t}$ | 16.00 pb | 36.60 pb |
| $\sigma_{ex;t+b}$ | 14.94 pb | 34.96 pb |
| $\sigma_{ex;t+b+c}$ | 14.83 pb | 34.77 pb |

Tab. 1.1: *Quark-mass effects to the Higgs production cross section for the gluon fusion channel [40]. σ_{EFT} denotes the cross section in an effective field theory, where the top-quark is considered infinitely heavy. $\sigma_{ex;t}$ denotes the cross section for a finite top-quark mass, but where all other quarks are considered massless. $\sigma_{ex;t+b}$ includes finite quark masses for the bottom quark and $\sigma_{ex;t+b+c}$ also for the charm quark. R_{LO} is the ratio $\sigma_{\text{EFT}}^{\text{LO}}/\sigma_{ex;t}^{\text{LO}}$ of the LO cross sections.*

| $\delta(\text{scale})$ | $\delta(\text{trunc})$ | $\delta(\text{PDF-TH})$ | $\delta(\text{EW})$ | $\delta(t, b, c)$ | $\delta(1/m_t)$ |
|------------------------|------------------------|-------------------------|---------------------|-------------------|-----------------|
| +0.10 pb -1.15 pb | ± 0.18 pb | ± 0.56 pb | ± 0.49 pb | ± 0.40 pb | ± 0.49 pb |
| +0.21% -2.37% | $\pm 0.37\%$ | $\pm 1.16\%$ | $\pm 1\%$ | $\pm 0.83\%$ | $\pm 1\%$ |

Tab. 1.2: *Contributions to the theoretical error in (1.1) [40]. $\delta(\text{scale})$ is the uncertainty due to scale variations and $\delta(\text{trunc})$ due to the truncation of the threshold expansion at N^3LO . $\delta(\text{PDF-TH})$ estimates the error made by using only NNLO parton distribution functions. $\delta(\text{EW})$ takes the uncertainty due to missing mixed QCD-electroweak corrections into account and $\delta(t, b, c)$ estimates the uncertainty because of the missing quark-mass effects at NNLO. $\delta(1/m_t)$ denotes the error made by treating the top-quark mass effects at NNLO by an expansion in $1/m_t$.*

mass decreases the NLO total cross section by 4.5%. Of course, at NNLO, where light quark mass effects are totally unknown, these corrections are expected to be much smaller.

However, tab. 1.2, which provides the individual contributions to the theoretical error estimate for (1.1), lists an uncertainty of $\pm 0.83\%$ because of this missing light quark mass effects. Since this uncertainty is of the same order of magnitude as other sources of error, trying to minimize this contribution is a plausible way to reduce the overall error.

The main motivation for this thesis is to develop new techniques to compute the missing light quark mass effects to the NNLO virtual QCD corrections and thus reduce the corresponding uncertainty.

1.2 Higgs Decay into Two Photons

The interactions of the Higgs boson with all massive particles in the SM lead to many different decay channels. These decay channels have the branching ratios depicted in fig. 1.3. While the dominant decay mode in fig. 1.3 clearly is the decay into a bottom-quark-antiquark pair, the process $H \rightarrow \gamma\gamma$, the decay into two photons, is very important as well. Even though it is mediated by a closed fermion loop or a massive vector boson loop and therefore very rare, it is nevertheless highly relevant due to the high precision to which photons can be measured in a particle detector [41].

The partial decay width for $H \rightarrow \gamma\gamma$ at LO was discussed very early in the literature (see e.g. Ref. [42]). The calculation of the closed fermion loop is almost identical to the cross section calculation of $gg \rightarrow H$.

The QCD corrections at NLO were first calculated in the heavy top limit independently by three groups, by Zheng and Wu [43] and by Djouadi, Spira, van der Bij and Zerwas [44], both in 1990,

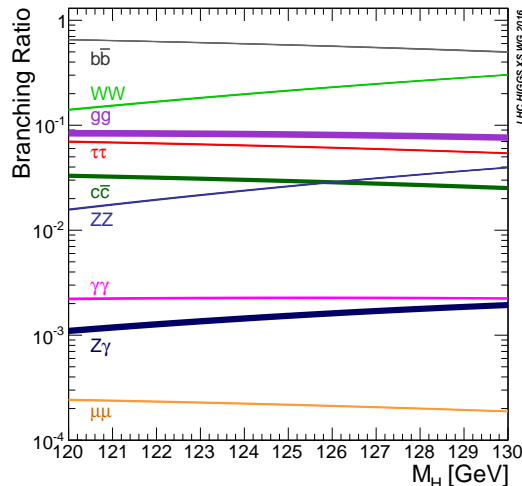


Fig. 1.3: Branching ratios for Higgs decay as a function of the Higgs mass M_H [25].

and by Dawson and Kauffman [45] in 1993. In 1995, the full quark-mass dependence was evaluated by Spira, Djouadi, Graudenz and Zerwas [31] in terms of one-dimensional integrals, which can be considered as by-product of their Higgs production via gluon-fusion calculation. Analytic results were obtained in 2004 by Fleischer, O.V. Tarasov and V.O. Tarasov [46], in 2005 by Harlander and Kant [32] (who evaluated the one-dimensional integrals of Ref. [31]) and in 2007 by Aglietti, Bonciani, Degrossi and Vicini [35].

A first, albeit incomplete, calculation of the NNLO QCD corrections was carried out by Steinhauser already in 1996 [47]. He evaluated most of the three loop diagrams in the heavy top limit (including subleading terms in the inverse top-mass up to order four), but neglected the so-called singlet diagrams, in which the Higgs boson and the photons couple to different closed quark loops. It took 16 years until Maierhöfer and Marquard completed the calculation in 2012 [48]. The authors of Ref. [48] also recalculated and extended the findings of Ref. [47] in order to provide a result for the decay width up to the tenth order in the inverse top-mass.

The Feynman diagrams for the process $H \rightarrow \gamma\gamma$ and $gg \rightarrow H$ have the same topologies. Since both photons and gluons are massless vector bosons, the kinematics between those two processes also coincide. The main difference between the two processes stems from the absent self-couplings of the external vector bosons in $H \rightarrow \gamma\gamma$, which drastically reduces the number of contributing Feynman diagrams. Therefore, a NNLO calculation of the light quark mass effects for the partial decay rate of the Higgs decay into two photons can be considered as a first step towards a result for the Higgs production cross section via gluon fusion.

1.3 Outline of the Thesis

In this thesis, we present techniques to improve the predictions for the $gg \rightarrow H$ cross section and the $H \rightarrow \gamma\gamma$ partial decay rate with a focus on the effects of light quarks. The virtual QCD corrections to these processes with an exact dependence on the mediating quark mass are only known up to next-to-leading order in the strong coupling constant α_s . Starting at next-to-next-to-leading order, the only results are expansions in the inverse quark mass, which cover the top-quark mass effects very well but break down for lighter quarks. We present the first steps to fill this gap by providing tools to perform the next-to-next-to-leading order calculations in the opposite limit, where the

mediating quark is considered light compared to the Higgs boson. We will also discuss why a calculation at this order with an exact quark mass dependence is not feasible at the moment.

Chapter 2 primarily describes the individual steps for both three-loop calculations. In Chapter 3, we introduce all techniques required for the calculation of the master integrals, while Chapter 4 explains their application to the integrals appearing in the considered Feynman amplitudes. In Chapter 5, we present first results for some master integrals. The final Chapter 6 gives an outlook to the still missing pieces of the project.

From Feynman Diagrams To Master Integrals 2

In general, a multiloop calculation always begins with the generation of the Feynman diagrams associated with the process. The higher the order of the calculation, the higher is the amount of diagrams that are necessary to describe the desired process. Also, the diagrams tend to become more complicated and require more and more elaborate computational methods.

In this chapter, we present the procedure that is used in this thesis for the three-loop virtual corrections to $gg \rightarrow H$ and $H \rightarrow \gamma\gamma$, in order to express the Feynman diagrams in terms of a rather small number of so-called master integrals.

In Section 2.1, we describe the generation of Feynman diagrams and the insertion of Feynman rules. Since the external states of the processes under consideration include vector bosons, a tensor reduction has to be performed. This is content of Section 2.2. Section 2.3 discusses an approximation to reduce the number of scales of the problem in order to simplify the calculation significantly. In Section 2.4, we describe the next step, the mapping to topologies, where we also explain the automatic generation of topology dependent FORM-code to express the Feynman integrals in a uniform way. Section 2.5 introduces a useful choice for a kinematic variable in which to express the results. At last, Section 2.6 deals with the reduction to master integrals via integration-by-parts and similar relations.

2.1 Feynman Diagrams

In perturbative quantum field theory, the amplitude of any process can be expressed as a sum of Feynman diagrams. This sum contains all possible diagrams that have the correct external legs, i.e. the right initial/final states, and obey the Feynman rules of the theory. The Feynman rules also provide mathematical expressions for every line (propagator) and every vertex in the diagram. At every vertex in the diagram, momentum and energy are conserved. Therefore, if a diagram contains loops, not all four-momenta through the propagators are fixed and one has to integrate over the free momenta. In QCD, increasing the number of loops in a diagram is equivalent to raising the power of the coupling constant α_s . In the perturbative regime this means that higher order terms should have a reduced contribution to the final result. Therefore, breaking of the infinite series of Feynman diagrams after a specific order should lead to a reasonable, if less precise, result.

The Feynman rules of QCD, extended by the Yukawa-coupling of the Higgs boson to a massive quark, are listed in Appendix B.

The generation of all Feynman diagrams to a given loop order is a graph theoretical problem. Algorithms for this problem are implemented in the program QGRAF [49]. The tool QGRAF not only is able to generate all Feynman diagrams in an adequate time even for high loop orders, but also

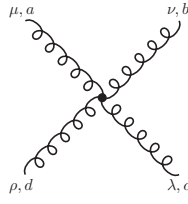
| | 1 loop | 2 loops | 3 loops |
|------------------------------|--------|---------|---------|
| $gg \rightarrow H$ | 2 | 22 | 623 |
| $H \rightarrow \gamma\gamma$ | 2 | 12 | 206 |

Tab. 2.1: Number of Feynman diagrams generated by *QGRAF* up to three loops for the processes $gg \rightarrow H$ and $H \rightarrow \gamma\gamma$.

calculates the required symmetry factors of the diagrams and the signs associated with closed fermion loops.

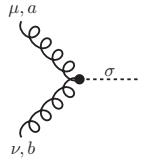
Next, the Feynman rules have to be inserted into the diagrams. This task is done by the tool *q2e* [50, 51]. The output of *q2e* consists of two files. One of them is a *FORM*-code [52] containing the mathematical expressions for the Feynman diagrams. The other one contains the topology of the Feynman diagram in a format that can be subsequently processed by the tool *exp* [50, 51] (see Section 2.4).

The tool *q2e* splits the expression for a Feynman diagram into a QCD-color factor and a momentum factor. This factorization is of course impossible if the diagram contains a four-gluon vertex



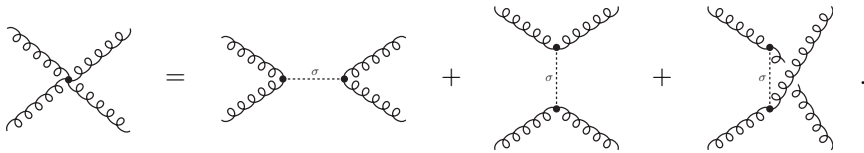
$$= -ig^2 [f^{abe} f^{cde} (g^{\mu\lambda} g^{\nu\rho} - g^{\mu\rho} g^{\nu\lambda}) + f^{ace} f^{bde} (g^{\mu\nu} g^{\lambda\rho} - g^{\mu\rho} g^{\nu\lambda}) + f^{ade} f^{cbe} (g^{\mu\lambda} g^{\nu\rho} - g^{\mu\nu} g^{\rho\lambda})].$$

Therefore, a new auxiliary particle σ has to be introduced with the Feynman rules

$$\alpha, \beta, a \cdots \sigma \cdots \gamma, \delta, b = i\delta^{ab} g_{\alpha\delta} g_{\beta\gamma},$$


$$= \frac{g}{\sqrt{2}} f^{abc} [g^{\mu\alpha} g^{\nu\beta} - g^{\mu\beta} g^{\nu\alpha}].$$

In these Feynman rules the color factor now factorizes. The Feynman rule of the four-gluon vertex can then be constructed as a sum of three diagrams featuring the σ -particle:



Obviously, this technique increases the number of Feynman diagrams.

The number of Feynman diagrams generated by *QGRAF* is given in tab. 2.1 for both processes $gg \rightarrow H$ and $H \rightarrow \gamma\gamma$. A few representative three-loop diagrams for $gg \rightarrow H$ are depicted in fig. 2.1(a)–(e) and for $H \rightarrow \gamma\gamma$ in fig. 2.1(f)–(j). Note that diagrams like the ones shown in fig. 2.1(d)–(e), where the external vector bosons do not directly couple to a quark line, are not present for the process $H \rightarrow \gamma\gamma$. This is the reason for the large difference in the number of Feynman diagrams between the two processes at three-loop order.

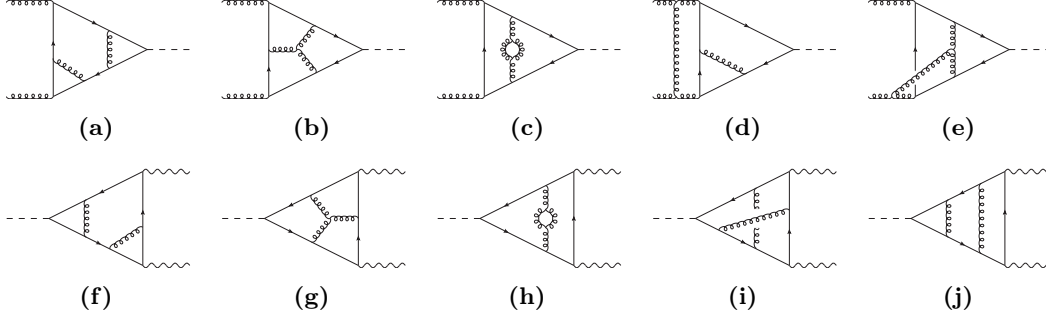


Fig. 2.1: Sample diagrams for the processes $gg \rightarrow H$ (fig. (a)–(e)) and $H \rightarrow \gamma\gamma$ (fig. (f)–(j)).

2.2 Tensor Reduction

Both processes, $gg \rightarrow H$ and $H \rightarrow \gamma\gamma$, have two vector bosons as external particles with momenta q_1 and q_2 . At first, we focus on the Lorentz structure, i.e. we ignore the color indices of the external gluons of the process $gg \rightarrow H$. The amplitudes can therefore be written as

$$\mathcal{M} = \varepsilon_{i,\mu}(q_1)\varepsilon_{j,\nu}(q_2)\mathcal{M}^{\mu\nu},$$

where the $\varepsilon_{i,\mu}(q)$ denote polarization vectors with $\varepsilon_i(q) \cdot q = 0$.

The external vector bosons are interchangeable. This is why we expect a symmetry $\mathcal{M}^{\mu\nu} = \mathcal{M}^{\nu\mu}|_{q_1 \leftrightarrow q_2}$. The most general Lorentz structure with two free Lorentz indices obeying this symmetry is given by

$$\mathcal{M}^{\mu\nu} = (q_1^\mu q_1^\nu + q_2^\mu q_2^\nu)A + q_1^\mu q_2^\nu B + q_2^\mu q_1^\nu C + (q_1 \cdot q_2)g^{\mu\nu}D,$$

where A, B, C, D are scalar form factors. For gluons in the external states, the (non-abelian) Ward identity $q_{1\mu}\varepsilon_{j,\nu}(q_2)\mathcal{M}^{\mu\nu} = 0$ can be used in order to constrain these coefficients (see e.g. Ref. [53]). This identity yields the relation $D = -C$, leading to the simplification

$$\mathcal{M}^{\mu\nu} = (q_1^\mu q_1^\nu + q_2^\mu q_2^\nu)A + q_1^\mu q_2^\nu B + [q_2^\mu q_1^\nu - (q_1 \cdot q_2)g^{\mu\nu}]C. \quad (2.1)$$

For external photons, the even stronger (abelian) Ward identity $q_{1\mu}\mathcal{M}^{\mu\nu} = 0$ leads in addition to a vanishing form factor A .

Cross sections and decay rates only depend on the polarization summed squared amplitude

$$\sum_{i,j} |\mathcal{M}|^2 = \sum_i \varepsilon_{i,\mu}(q_1)\varepsilon_{i,\rho}^*(q_1) \sum_j \varepsilon_{j,\nu}(q_2)\varepsilon_{j,\sigma}^*(q_2) \mathcal{M}^{\mu\nu} \mathcal{M}^{*\rho\sigma}. \quad (2.2)$$

The polarization sums are given by

$$\sum_i \varepsilon_{i,\mu}(q_{1,2})\varepsilon_{i,\nu}^*(q_{1,2}) = -g_{\mu\nu} + \frac{q_{1\mu}q_{2\nu} + q_{2\mu}q_{1\nu}}{q_1 \cdot q_2}, \quad (2.3)$$

where we have chosen q_2 as reference vector to $\varepsilon_i(q_1)$ and q_1 as reference vector to $\varepsilon_i(q_2)$. Plugging (2.1) and (2.3) into (2.2) yields

$$\sum_{i,j} |\mathcal{M}|^2 = (d-2)(q_1 \cdot q_2) |C|^2. \quad (2.4)$$

Hence, only the form factor C has to be calculated. The coefficient C can be projected out of (2.1) by

$$C = \frac{1}{(q_1 \cdot q_2)(2-d)} \left[g_{\mu\nu} - \frac{q_{2\mu}q_{1\nu}}{(q_1 \cdot q_2)} \right] \mathcal{M}^{\mu\nu}. \quad (2.5)$$

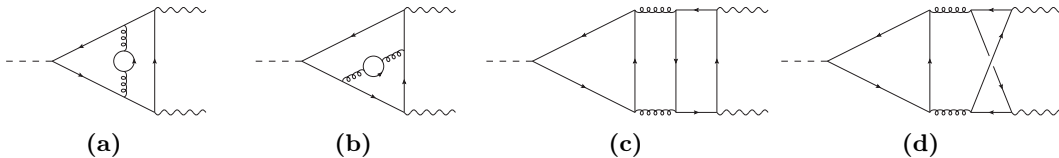


Fig. 2.2: Sample Feynman diagrams to the process $H \rightarrow \gamma\gamma$ involving more than two scales.

For the process $gg \rightarrow H$, one also has to consider the color structure of the amplitude. The only possible color tensor with two indices in the adjoint configuration is the Kronecker delta δ^{ab} . Thus, we can write

$$\mathcal{M}^{ab} = \delta^{ab} \mathcal{M}.$$

The factor \mathcal{M} can easily be projected out by

$$\mathcal{M} = \frac{\delta^{ab}}{N_A} \mathcal{M}^{ab}, \quad (2.6)$$

where $N_A = N_C^2 - 1$ is the number of color-states in the adjoint representation.

The projectors (2.5) and (2.6) have been implemented in FORM-code [52].

2.3 Two-Scale Approximation

The complexity of a multiloop calculation strongly depends on the number of scales involved in the problem. Up to two-loops, the processes we discuss in this thesis, $gg \rightarrow H$ and $H \rightarrow \gamma\gamma$, depend on only two scales, i.e. the mass of the intermediate quark m and the mass of the Higgs boson M .

Unfortunately, two scales are not sufficient at the three-loop level. Fig. 2.2 shows sample diagrams involving more than two scales. These diagrams contain two closed quark loops, where both quarks are in general of a different type.

In order to simplify the computation, we only keep the mass m of the quark that is coupled to the Higgs boson. For the second quark with mass m_2 , we have to distinguish three cases:

1. $m_2 < m$: The second quark is lighter than the first quark,
2. $m_2 > m$: The second quark is heavier than the first quark,
3. $m_2 = m$: Both quarks are of the same type.

The case 3 can be treated easily by setting $m_2 = m$ in the calculation.

For the case 1, we treat all quark masses less than m as zero. Hence, we calculate the Feynman diagrams with $m_2 = 0$. Since not only one lighter quark contributes, the result has then to be added multiple times to the amplitude. For the process $gg \rightarrow H$ this is very simple, since we only have to multiply the contribution by a factor n_l , counting the lighter quarks¹. For the process $H \rightarrow \gamma\gamma$ one has to be careful and treat the singlet diagrams (fig. 2.2(c)–(d)) correctly. Here the external photons couple with a different coupling strength to up- and down-type quarks.

The case 2 was not considered in this thesis, but can be treated with an expansion by subgraphs [54–57] (see also Section 3.4).

¹Since we are mainly interested in the bottom-quark contributions, this would be $n_l = 4$.

2.4 Topologies

As a next step, we map all Feynman diagrams onto a small number of topologies. This task is accomplished by the tool `exp` [50, 51]. The 23 required topologies for a three-loop calculation are depicted in fig. 2.3, where mirrored versions of unsymmetric graphs are not shown. Note that we do not distinguish different mass distributions at this point.

The output of `exp` consists of FORM-files, which contain mathematical expressions for the Feynman diagram and the associated topology name. The momenta in the mathematical expression are labeled in the same way as the corresponding lines in the topology, i.e. the momentum through line N in one of the topologies in fig. 2.3 is called p_N .

The output files can then be processed by a FORM-code common for all Feynman diagrams with the primary function to evaluate traces of Dirac matrices that stem from the Feynman rules. This FORM-code originated from the MINCER [58] and MATAD [59] project. Furthermore, the FORM-package `color` [60] is called to calculate the QCD color factors.

Afterwards, the four-momentum conservation at every vertex has to be inserted and the scalar Feynman integrals in the expressions have to be rewritten in a form suitable for further processing. This is clearly dependent on the topology of the diagram. Hence, a specific FORM-code has to be written for all 23 topologies.

We chose an automatic approach to generate these FORM-files. But before we discuss the generation of these codes, we have to specify the desired output. We would like to express all integrals over loop-momenta in terms of scalar functions of the form

$$F(\text{ID}; a_1, \dots, a_N) = \int_{l_1} \dots \int_{l_K} \frac{1}{D_1^{a_1} \dots D_N^{a_N}}, \quad (2.7)$$

where K is the number of loops and ID is a number that exactly specifies the denominators D_1, \dots, D_N . Numerators in the integrand of (2.7) are represented by negative indices a_j . The notation used for the integral measure is defined in Appendix A. The ID should contain the topology as well as the mass distribution of the Feynman integral. The set of all integrals with the same ID is denoted as integral family. Because of the approximation described in Section 2.3, we only deal with massless propagators or massive propagators with one specific mass m at this point. This enables us to encode the mass distribution as a number, where every bit in its binary representation stands for the mass of one propagator. For a massive propagator, we set the corresponding bit to 1 and for a massless propagator to 0. To get the ID, the number of the topology is multiplied by 10000 and added to this binary representation of the mass distribution.

To be able to express any possible numerator structure in terms of functions F , it is a necessary condition that the D_i are chosen such that every possible scalar product involving a loop momentum can be written as a linear combination of D_i . Further, we do not permit subsets of the D_i to be linearly dependent. Therefore, the number of indices N is fixed to the number of possible scalar products involving at least one loop-momentum. This number is given by

$$N = \frac{K(K+1)}{2} + KX,$$

where X is the number of independent external momenta. For the three loop calculations of this thesis, we have $K = 3$ and $X = 2$, which leads to twelve indices. Nine of the twelve denominators are determined by the momentum flow through the lines in the topology (cf. fig. 2.3) and the mass distribution. The remaining three denominators have to be chosen linearly independent with respect to the already fixed ones. We choose them to be simple bilinear forms (scalar products).

The `Mathematica`-script `topologies.m` generates a dedicated FORM-code for every topology using the same input file that already specified the topologies for the tool `exp`.

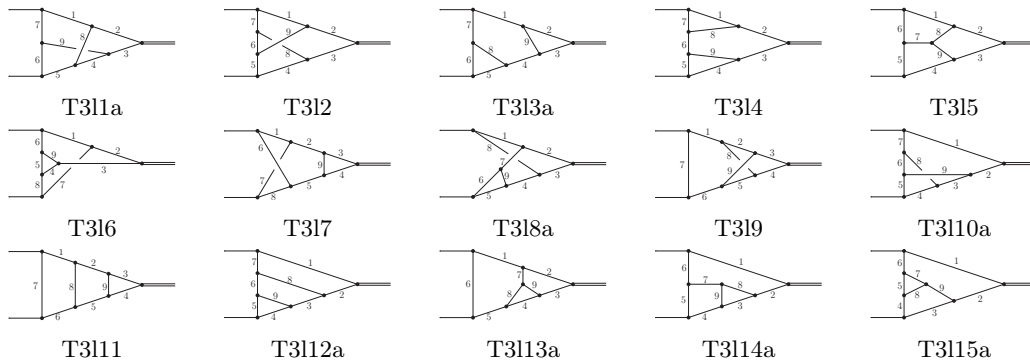


Fig. 2.3: 15 topologies required for $gg \rightarrow H$ and $H \rightarrow \gamma\gamma$. The upper incoming line carries momentum q_1 and the lower incoming line q_2 . The topologies with the suffix ‘a’ are not mirror symmetric w.r.t. the horizontal axis, this is why a second mirrored topology has to be added (denoted by a suffix ‘b’).

For each topology contained in the input file, `topologies.m` first assigns a momentum flow through each line by constructing one equation for every vertex. Such an equation represents the momentum conservation at each vertex as a set of momenta that sum up to zero. The resulting system of equations is then solved for the momenta through the individual lines of the topology using the `Solve` function of `Mathematica`. Since this system is under-determined, some momenta remain in the solution, which are then chosen as the loop-momenta. The solution of this system fixes the nine denominators that correspond to lines of the topology.

The three extra denominators D_{10}, D_{11}, D_{12} are chosen from the list of all twelve scalar products involving at least one loop-momentum. A simple algorithm tries out these scalar products one-by-one to find those independent of the already fixed denominators. The choice of these extra denominators is clearly independent of the mass distribution. For this reason, we are able to choose the same scalar products for all families of the same topology.

After fixing all denominators for a given topology, the `FORM`-code that rewrites everything in terms of functions F as defined in (2.7) can finally be generated by `topologies.m`. This `FORM`-code will later be used on all Feynman diagrams that correspond to this specific topology.

Every term of the expression that enters this generated `FORM`-code contains a product of propagators

$$\frac{1}{[p_i^2 - m_i^2]^a} \quad ; \quad a > 0,$$

where p_i is the momentum through line i of the topology and m_i is either zero or m . Products of these propagators can be directly rewritten as a function G that can be understood as another variant of F in (2.7). Instead of encoding it in the `ID` argument, the function G represents the mass distribution as another set of indices. If two propagators with the same momentum p_i but different m_i appear, a partial fraction decomposition

$$\frac{1}{[p_i^2][p_i^2 + m^2]} = -\frac{1}{m^2} \frac{1}{[p_i^2 + m^2]} + \frac{1}{m^2} \frac{1}{[p_i^2]}$$

has to be performed first.

Afterwards, the generated `FORM`-code has to deal with possible numerators. For this purpose, `topologies.m` builds a set of rules expressing all possible twelve scalar products involving loop-momenta as linear combinations of the denominators D_i . Obviously, the dependence of the denominators D_i on the mass distribution is passed over to these relations. The denominators D_i on the right-hand side of these rules can be thought of as ladder operators decreasing the indices of

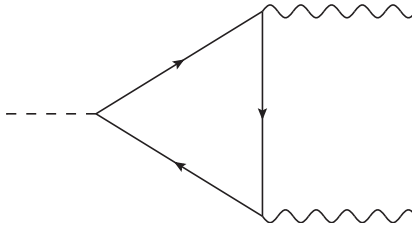


Fig. 2.4: One-loop contribution to the decay $H \rightarrow \gamma\gamma$.

the function G . Since the function G contains the mass distribution in a simple way, a FORM-identity can easily be generated for every rule. It incorporates the scalar product on the left-hand side into the function G by applying the ladder operators on the right-hand side.

As a last step, the FORM-code has to rewrite the function G into a function F , as defined in (2.7), i.e. it has to change the way the mass distribution is represented. In addition to the generated FORM-code, `topologies.m` also stores the denominators for the integral families in an extra file suitable for further use.

2.5 The Kinematic Variable

The approximation introduced in Section 2.3 reduces the number of relevant scales for the considered problems to only two, namely the mass of the Higgs boson M and the mass of the intermediate quark m . It is often convenient to parametrize a dependence on these kinematic scales through a dimensionless variable. For our purpose, a practical choice is

$$\tau = \frac{M^2}{4m^2}.$$

In the upcoming chapters, we deal with Feynman integrals whose internal mass m we usually set to unity. For the external momenta, this leads to

$$q_1^2 = q_2^2 = 0, \quad q_1 \cdot q_2 = 2\tau.$$

Considering the well-known contribution of the diagram in fig. 2.4 to the $H \rightarrow \gamma\gamma$ amplitude [42, 61]

$$\mathcal{M} \sim \frac{2}{\tau^2} [\tau + (\tau - 1)f(\tau)], \quad f(\tau) = \begin{cases} -\frac{1}{4} \ln^2 \left(\frac{\sqrt{1-1/\tau}-1}{\sqrt{1-1/\tau}+1} \right) & ; \tau \leq 1 \\ -\frac{1}{4} \left[\ln \left(\frac{1+\sqrt{1-1/\tau}}{1-\sqrt{1-1/\tau}} \right) - i\pi \right]^2 & ; \tau > 1, \end{cases}$$

we see that the variable τ seems not to be a good choice. A better variable would be

$$x = \frac{\sqrt{1-1/\tau}-1}{\sqrt{1-1/\tau}+1} + i0. \tag{2.8}$$

related to τ by a conformal mapping (see e.g. [62]). This relation can be inverted via

$$\tau = -\frac{(1-x)^2}{4x}, \tag{2.9}$$

in order to express τ as a function of x . This mapping is illustrated in fig. 2.5.

The purpose of this variable is not only aesthetic. The algorithm proposed in Ref. [63] and explained in great detail in Section 3.3 to find a so-called canonical basis for master integrals requires such a

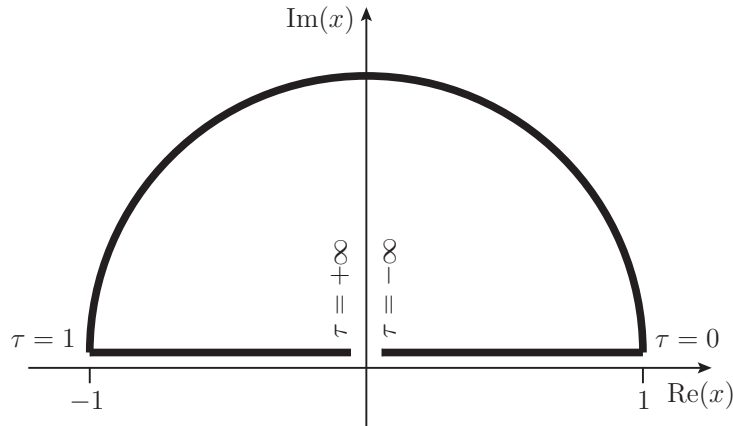


Fig. 2.5: Mapping from the variable τ to the variable x . For $\tau < 0$ and $\tau > 1$, x has an infinitesimal positive imaginary part. The region $0 < x < 1$ corresponds to the Euclidean domain ($\tau < 0$), the interval $-1 < x < 0$ corresponds to the region above threshold ($\tau > 1$), and the unit circle $x = e^{i\theta}$ for $0 < \theta < \pi$ corresponds to a value of τ below threshold $0 < \tau < 1$.



Fig. 2.6: Master integrals that can be expressed in terms of Goncharov polylogarithms but require a different kinematic variable.

choice of variable. For an inconvenient variable, this algorithm would not be able to find a rational transformation matrix to a basis, where the master integrals can be expressed purely in terms of Goncharov polylogarithms (see Section 3.1).

At the three loop level, a new feature emerges related to the choice of variable. Because of the two-scale approximation (see Section 2.3), new master integrals like the ones depicted in fig. 2.6 are required. The variable x in (2.8) is not a good choice for these master integrals, since even though these integrals can be expressed in terms of Goncharov polylogarithms, no rational transformation to a canonical basis can be found. The integral in fig. 2.6(a) factorizes into a one-loop massive tadpole and a two-loop integral. The two-loop integral can be found in Ref. [64], where a new variable \bar{x} was used, defined by

$$\bar{x} = \frac{1 - \sqrt{1 + 1/\tau}}{1 + \sqrt{1 + 1/\tau}} + i0, \quad \tau = -\frac{(1 + \bar{x})^2}{4\bar{x}}.$$

However, difficulties arise when one wants to solve, for example, the master integral shown in fig. 2.6(b). The differential equation fulfilled by this integral depends on master integrals that are not expressible in terms of Goncharov polylogarithms of x , as well as on those not expressible in terms of Goncharov polylogarithms of \bar{x} . Hence, a choice of variable is required that is convenient for both classes of master integrals. A recent paper by Lee and Pomeransky [65] presents a systematic approach to find such a kinematic variable. Unfortunately, the variable y related to τ by

$$\tau = \frac{2y^2}{1 + y^4},$$

which is found this way, leads to singularities in the differential equations at complex points that are not zeros of quadratic polynomials. Such complex numbers are unsuitable for the **Fermat** computer algebra system [66], on which our implementation of Lee’s algorithm [63] is based. Therefore, we treat these problematic integrals in the same way we deal with the integrals that cannot be expressed at all in terms of Goncharov polylogarithms: by an expansion in the variable x (see Section 4.2).

2.6 Reduction to Master Integrals

The setup discussed in the previous sections leads to expressions for the amplitudes in terms of a huge number of scalar Feynman integrals of the form (2.7). For the $gg \rightarrow H$ amplitude, 10371 scalar integrals are needed, while for the process $H \rightarrow \gamma\gamma$ this number is reduced to 8814 integrals. Fortunately, not all of these integrals are linearly independent. Hence, they can be reduced to a smaller number of so-called master integrals.

In Section 2.6.1, we consider the integration-by-parts and Lorentz-invariance identities required for the reduction of a single Feynman integral family to master integrals. In Section 2.6.2, another type of relations is discussed, which can be used to identify equal integrals among different integral families, i.e. the sector relations. Section 2.6.3 focuses on how to handle the huge amount of data produced by the reduction procedure via a single-file database.

2.6.1 Integration-by-Parts and Lorentz-invariance Identities

The integration-by-parts (IBP) identities [67, 68] are the most important relations between scalar multiloop Feynman integrals of the same family. They can be used to generate a large system of linear equations, which is then solvable with a Gauß algorithm.

IBP identities are generated by the simple observation that for dimensionally regularized scalar Feynman integrals, the equation

$$0 = \int_{l_1} \cdots \int_{l_K} \frac{\partial}{\partial l_i^\mu} \frac{p^\mu}{D_1^{a_1} \cdots D_N^{a_N}} \quad (2.10)$$

holds true. Here, l_i is one of the loop momenta l_1, \dots, l_K and p is either a loop momentum or an external momentum. This formula can be written as

$$0 = \int_{l_1} \cdots \int_{l_K} \left(\frac{\partial p^\mu}{\partial l_i^\mu} \right) \frac{1}{D_1^{a_1} \cdots D_N^{a_N}} - \sum_{j=1}^N a_j \int_{l_1} \cdots \int_{l_K} \frac{p^\mu \frac{\partial D_j}{\partial l_i^\mu}}{D_1^{a_1} \cdots D_{j-1}^{a_{j-1}} D_j^{a_j+1} D_{j+1}^{a_{j+1}} \cdots D_N^{a_N}}. \quad (2.11)$$

For the IBP identities with $p = l_i$, the derivative in the first term of (2.11) yields d , i.e. the number of space-time dimensions. For all other identities, the first term vanishes. The numerators in the second term are sums of scalar products of loop momenta and external momenta.

For a given set of denominators D_i of an integral family specified by an ID as in (2.7), the right-hand side of (2.11) can be expressed as a linear combination of the scalar functions $F(\text{ID}; \dots)$. The indices of these functions F are related to the indices a_1, \dots, a_N by integer shifts. Note that we defined an integral family in such a way that every scalar product involving a loop momentum can be expressed as linear combinations of the denominators D_i .

For a given integral family, this leads to a total of $K(K + X)$ IBP identities, where K is the number of loops and X the number of independent external momenta.

In addition to the IBP identities one can also use Lorentz-invariance (LI) identities [69] with similar properties. LI identities are obtained by contracting

$$0 = \sum_{i=1}^X \left(q_i^\nu \frac{\partial}{\partial q_{i\mu}} - q_i^\mu \frac{\partial}{\partial q_{i\nu}} \right) \int_{l_1} \cdots \int_{l_K} \frac{1}{D_1^{a_1} \cdots D_N^{a_N}} \quad (2.12)$$

with all antisymmetric combinations of the form

$$q_{j\mu} q_{k\nu} - q_{k\mu} q_{j\nu},$$

where q_j and q_k are two independent external momenta. These equations can be treated in the same way as discussed for (2.10), leading to vanishing linear combinations of scalar Feynman integrals with shifted indices. Although it is true that these LI identities can always be represented as linear combinations of IBP relations [70], their inclusion speeds up at least some reduction algorithms.

There is no unique definition of the term master integral. First, a reduction algorithm has to specify an ordering for all Feynman integrals of a given family. Then one is able to define the master integrals as the set of lowest linearly independent integrals with respect to this ordering. This ordering is usually chosen in such a way that a lower integral is easier to evaluate. The number of linearly independent integrals required to express all Feynman integrals of a given family is finite [71]. Thus, one can think of a scalar Feynman integral as a vector in a finite dimensional vector space with the master integrals as a basis.

Essentially, two different strategies are known to utilize IBP and LI identities for a reduction to master integrals. One method constructs a set of rules by combining the identities with *symbolic* indices in a very specific way. When applied to a Feynman integral with *numeric* indices, these rules always lead to a linear combination of lower integrals with respect to the chosen ordering. A recursive application of this set of rules then reduces all integrals to master integrals. This approach is used in the FORM packages MINCER [58] and MATAD [59] as well as in the recently published program FORCER [72]. While these programs are restricted to certain types of Feynman integrals, the Mathematica-package LiteRed, presented in Ref. [73, 74], can be used to construct such rules for arbitrary integrals. Unfortunately, the construction of the rules is highly non-trivial and the underlying algorithms in LiteRed tend to become slow for complicated cases. Further, even if the rules can be found on a reasonable timescale, due to their complexity, their application is often not fast enough to obtain a full reduction in a feasible time.

The second strategy is based on the generation of a large linear system of equations for Feynman integrals with *numeric* indices. Here, one first generates all combinations of numerical indices up to certain boundaries. These indices are then inserted into the IBP and LI identities, in order to generate a set of equations. This system is then solved for the higher (more complicated) integrals with a Gaussian elimination algorithm.

The second approach was first proposed by Laporta [75]. Today, various public implementations [76–82] and many more private ones exist, which often incorporate some improvements over the original algorithm. The development of one of these packages, the C++-program Kira [82], was strongly influenced by the reduction problems presented in this thesis. Further, at the time of writing, it was the only public tool able to perform the required reductions for the $H \rightarrow \gamma\gamma$ process in a feasible time. One notable feature of Kira is the heavy use of modular arithmetic to remove linearly dependent equations at an early stage of the reduction.

2.6.2 Sector Relations

In the previous section, we discussed relations between Feynman integrals within the same family. However, different families of Feynman integrals may also be connected to each other through so called sector relations.

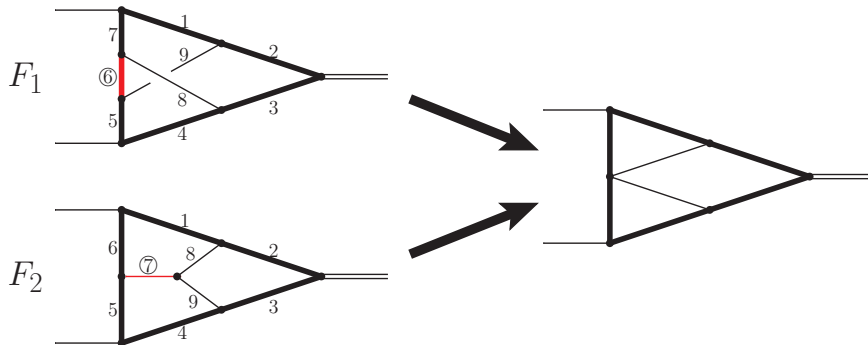


Fig. 2.7: Two integral families F_1 and F_2 with subsectors related by sector relations. The red lines (labeled with circled numbers) denote propagators not present in the subsectors under consideration.

The sector S of a scalar Feynman integral F , given in the form (2.7), is represented by the set $\{i|a_i > 0\}$, i.e. the set of all i for which the corresponding indices a_i are positive. With this in mind, a subset of this set represents a different sector, which is called a subsector of S . If a sector can be represented by a Feynman graph, the graph of one of its subsectors is obtained by contracting the lines corresponding to the missing indices to a point.

Now we consider the full sectors of the two integral families shown on the left-hand side of fig. 2.7, where all lines of the corresponding topology are present. As shown in fig. 2.7, the sector $S_1 = \{1, 2, 3, 4, 5, 7, 8, 9\}$ of the upper family F_1 and the sector $S_2 = \{1, 2, 3, 4, 5, 6, 8, 9\}$ of the lower family F_2 have isomorphic graphs. Hence, an integral belonging to sector S_1 of family F_1 can always be expressed as a linear combination of integrals belonging to sector S_2 of family F_2 and subsectors of S_2 (and vice versa). These types of identities are called *sector relations*.

Since equal or related master integrals can appear in different integral families, sector relations help to further reduce the number of master integrals. However, these relations are also useful in order to decrease the number of integrals that have to be reduced in the first place. In particular, Feynman integrals belonging to mirrored topologies are identified with each other.

The tool `Reduze 2` [81] offers the possibility to automatically detect sector relations². These relations can subsequently be exported as `Mathematica` rules and included into the reduction tables.

2.6.3 The Reduction Database

The reduction tables provided by `Kira` are often huge in size. They contain rules for all integrals up to certain boundaries $r \leq r_{\max}$ and $s \leq s_{\max}$, where r is the sum of all positive indices of a Feynman integral and s is the absolute value of the sum of all negative indices.

In order to handle this huge amount of data, one could for example only export the rules for those integrals needed in the amplitudes in a suitable format (e.g. `Mathematica` rules or a `FORM` tablebase). Unfortunately, this naïve strategy has the disadvantage of being not flexible enough, when one wants to include the sector relations discussed in Section 2.6.2. A sector relation often involves a permutation of indices when mapping from one family to another. Since the integral ordering in most Laporta implementations only depends on the sequence of indices, a master integral in a certain integral family is not necessarily a master integral in another family after the mapping. Hence, after the application of sector relations, a second run of the reduction procedure is required.

²The published version of `Kira` also offers options to include sector relations. This feature was not yet available in the preliminary version used for this thesis.

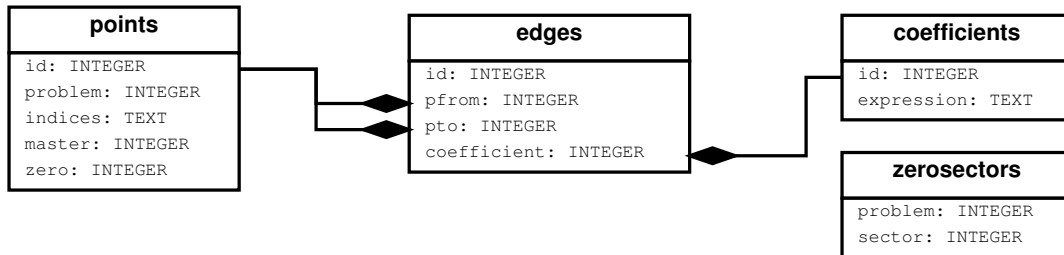


Fig. 2.8: SQL-tables for the reduction database. The lines $A \blacklozenge B$ can be read as “A refers to B”.

Since we prefer to perform all reductions required for a specific family in one run, we use a more sophisticated approach, where we store all reduction data in a file-based database system. The choice fell on the system `SQLite` [83], which offers access using the standardized `Structured Query Language` (SQL) and can also be integrated in `Mathematica` [84].

The database is set up with the SQL-tables shown in fig. 2.8. The field `id` in three of the four tables is used to identify a certain row when it is referenced by a row of another table.

The table `points` holds one row for every Feynman integral appearing on any side of a reduction rule. The family ID is stored in `problem` and the indices as a comma-separated string in the field `indices`. The fields `master` and `zero` are actually Boolean values denoting if the integral is a master integral and if it is zero.

In the table `edges`, multiple rows contain the same value in the field `pfrom`. All these rows form a reduction rule. The integer value in `pfrom` refers to an integral in the table `points`. This integral is the left-hand side of the rule, i.e. the integral to be reduced. The value in `pto` refers to an integral in the linear combination on the right-hand side of the rule. The value in the field `coefficient` points to a row in the table `coefficients`, which holds the mathematical expression of the corresponding coefficient as character string in the field `expression`.

The table `zerosectors` contains one row for every *trivial sector*. A trivial sector is a sector in which every integral is zero (e.g. if the sector contains a massless tadpole). The field `problem` stores the ID of the family and the field `sector` encodes the sector as an integer value. This integer value is given by $\sum_{i \in S} 2^{i-1}$, where S is a set that defines the sector as in Section 2.6.2.

Using this format both the tables from the Laporta reduction and the sector relations can be inserted. Already at this step, we perform the change of variable (2.9) such that the coefficients in the database are expressed in terms of x . A `Mathematica`-package was written to access the rules from within `Mathematica`. This is reasonably fast because of the use of SQL-indexes wherever it was appropriate.

At this stage, the database does not ensure that the reduction leads to master integrals after applying the rules a single time, because of the different sources of the reduction rules. In order to optimize the database in that sense, a parallelized C++-program was written to perform a back-substitution at the database level. Of course, this C++-code has to handle the multiplication and summation of rational functions, which is performed by the `Fermat` computer algebra system [66]. After the back-substitution stage, it is guaranteed that every rule in the database contains only master integrals from a minimal basis on the right-hand side.

Calculation Methods for Master Integrals 3

The evaluation of master integrals is an essential part of a multiloop calculation. Many techniques have been developed over the years. For an overview see e.g. Ref. [85]. Among the most successful ones are the method of differential equations [86–88], Mellin-Barnes integral representations [89, 90], and, for numerical evaluations, the method of sector decomposition [91–93].

In this chapter, we discuss the various methods used in this thesis for solving the master integrals of $gg \rightarrow H$ and $H \rightarrow \gamma\gamma$. Most of the methods can be formulated in a more general way for multiscale problems. Since, for this thesis, only master integrals with two scales are required, we refrain from giving a full discussion of this very involved topic.

In Section 3.1, we introduce Goncharov Polylogarithms, which are basic building blocks for a large class of Feynman integrals. In particular for this class the method of differential equations proves to be very powerful. This method is discussed in Section 3.2 and Section 3.3. In the latter, recent developments involving a so-called canonical basis of master integrals are described and an implementation of an algorithm to find such a basis is presented. Section 3.4 describes a strategy to calculate boundary conditions, also needed for the method of differential equations. Section 3.5 deals with some standard parametrizations of Feynman integrals. Another impressive method, the evaluation by Mellin-Barnes representations, is covered in Section 3.6. A novel method to find these Mellin-Barnes representations is presented in Section 3.7.

3.1 Goncharov Polylogarithms

The Goncharov polylogarithms¹ (GPLs) are the basic building blocks of many multiloop Feynman integrals. As a generalization of classical polylogarithms they were studied very early by Kummer [94] and Poincaré [95] and later by Chen in his famous work on iterated integrals [96]. The work of Goncharov [97] in 1998 about this class of functions then caught the attention of the physics community (see also Ref. [98]).

The Goncharov polylogarithms are defined by

$$G(a_1, \dots, a_n | x) = \int_0^x \frac{dz}{z - a_1} G(a_2, \dots, a_n | z), \quad (3.1)$$
$$G(\underbrace{0, \dots, 0}_n | x) = \frac{1}{n!} \ln^n(x), \quad G(|x) = 1,$$

¹Other names in the literature include generalized polylogarithms, multiple polylogarithms, and hyperlogarithms.

where the number of indices n is called weight and $G(\cdot|x)$ is a weight-zero GPL. The GPLs are generalizations of the harmonic polylogarithms (HPLs) [99]. The HPLs are recovered if the indices a_i are restricted to $-1, 0, 1$.

The GPLs fulfill a number of identities. In this thesis, we make heavy use of the shuffle identity [100]

$$G(a_1, \dots, a_n|x)G(b_1, \dots, b_m|x) = \sum_{\vec{c} \in \vec{a} \sqcup \vec{b}} G(c_1, \dots, c_{n+m}|x) \quad (3.2)$$

and the scaling identity

$$G(\vec{a}, b|x) = G(k\vec{a}, kb|kx) \quad ; \quad b \neq 0 \wedge k \neq 0 \quad (3.3)$$

The right-hand side of (3.2) sums over all arrangements of the indices $a_1, \dots, a_n, b_1, \dots, b_m$, where the order of the elements inside the two lists a_1, \dots, a_n and b_1, \dots, b_m is kept individually.

In this section, we describe three important algorithms to deal with GPLs. Section 3.1.1 describes an algorithm for the reduction of GPLs to a minimal set of basis GPLs. In Section 3.1.2, we present a method for an expansion of GPLs around the arguments $x = 0$ and $x = 1$. Section 3.1.3 explains the analytic continuation of GPLs to negative arguments.

3.1.1 A Minimal Basis of GPLs

The shuffle relation (3.2) indicates many possible relations between different GPLs, i.e. not all GPLs of a given weight are linearly independent. Hereby, a GPL of a given weight can be identified with a sum of GPLs of the same weight with permuted indices and products of lower-weight GPLs. One can use these identities to reduce all GPLs of a given weight to a minimal set of linearly independent ones. The algorithm presented in this section is a generalization of the method described in Ref. [99] for HPLs.

While the indices of HPLs might only be $-1, 0, 1$, the generalization to GPLs has to allow all complex numbers. Therefore, as a first step, one has to define an ordering for complex numbers a and b by

$$a < b \Leftrightarrow \text{Re}(a) < \text{Re}(b) \vee (\text{Re}(a) = \text{Re}(b) \wedge \text{Im}(a) < \text{Im}(b)). \quad (3.4)$$

Using this definition, we define an ordering for GPLs via

$$G(\vec{a}|x) \prec G(\vec{b}|x) \quad \text{if } \text{GPLLESS}(\vec{a}, \vec{b}) \text{ returns TRUE,} \quad (3.5)$$

where the function GPLLESS is given in alg. 3.1. The special treatment of the indices 0 and 1 is necessary in order to ensure compatibility with the minimal basis defined for HPLs in Ref. [99] and is also useful to split of singularities at $x = 0$ and $x = 1$. In the lexicographically ordering at line 17 the indices 0 and 1 are always considered as less than all other indices (and $0 < 1$).

The minimal basis does not depend on the numerical values of the indices but only on their relative order. Thus, one can first compute reduction tables for symbolic indices $a < b < c < \dots$ and store them in a database. Because of the special treatment of the indices 0 and 1 in alg. 3.1, these values are not represented by the symbolic indices but kept numeric. The reduction rules for a given set of GPLs can later be obtained from this database by inserting numerical values in the correct order into a, b, c, \dots .

The right-hand side of the shuffle relation (3.2) consists of GPLs with different permutations of the *same* indices. Therefore, the algorithm has to find individual reduction rules for every possible set of (symbolic) indices, where in particular different multiplicities have to be taken into account. For example, a weight-five GPL with one 0, no 1 and where the second lowest index (not 0 and not 1) occurs twice is found by the algorithm when run with the set of indices ‘0abbc’².

²We will make use of this character string notation for lists of indices throughout this section.

Algorithm 3.1 Ordering of two lists of indices a and b

```

1: function GPLLESS( $a, b$ )
2:   in:   $a, b$ : two lists of indices.
3:   out: True, if  $G(a|x) \prec G(b|x)$  and False, otherwise.
4:   if  $|a| \neq |b|$  then
5:     return  $|a| < |b|$ 
6:   end if
7:    $N_a \leftarrow$  number of 1's at the beginning of  $a$ 
8:    $N_b \leftarrow$  number of 1's at the beginning of  $b$ 
9:   if  $N_a \neq N_b$  then
10:    return  $N_a > N_b$ .
11:  end if
12:   $N_a \leftarrow$  number of 0's at the end of  $a$ 
13:   $N_b \leftarrow$  number of 0's at the end of  $b$ 
14:  if  $N_a \neq N_b$  then
15:    return  $N_a > N_b$ .
16:  end if
17:  if  $a$  is lexicographically less than  $b$  then
18:    return FALSE
19:  else
20:    return TRUE
21:  end if
22: end function

```

All required multisets³ of indices for a given weight can be determined by the function INDEXSETS in alg. 3.2. For $N = 3$ this function will generate the 15 lists of indices

'000', '001', '00a', '011', '01a'
 '0aa', '0ab', '111', '11a', '1aa'
 '1ab', 'aaa', 'aab', 'abb', 'abc'.

For all lists of indices generated by alg. 3.2 the function REDUCTION in alg. 3.3 can be started afterwards. This can of course be done in parallel. The function CUT, which is called from line 7 of alg. 3.3, cuts the list of indices s in all possible ways and returns a set of distinct multicuts. A multicut is represented as an ordered list of lists of indices, where the ordering is defined in the sense of alg. 3.1. If all indices in s are distinct, the result of CUT contains $2^{|s|-1} - 1$ elements. If some indices in s are repeated, different multicuts can lead to the same element in the result after the sorting. The result for CUT('a0ab') would be

{('b', 'a0a'), ('a', '0ab'), ('ab', 'a0'), ('b', 'a', 'a0'),
 ('b', 'a', '0a'), ('a', '0', 'ab'), ('b', 'a', 'a', '0')}.

In line 13 of alg. 3.3 it is expected that the reduction algorithm has already been completed for lower-weight GPLs. For GPLs of weight one, the reduction table is empty.

For example, a reduction rule for GPLs of weight three looks like

$$G(b, 0, a|x) = -G(0, a, b|x) - G(0, b, a|x) + G(0, a|x)G(b|x).$$

³See Appendix A.2 for the definitions of the data structures used in the thesis.

Algorithm 3.2 Algorithm to generate all multisets of indices for a given weight

```

1: function INDEXSETS( $N, x \leftarrow '0'$ )
2:   in:    $N$ : number of elements in the resulting set of indices
            $x$ : a symbolic index of the list  $'0', '1', 'a', 'b', 'c', \dots$ 
3:   out: a set of distinct sets of indices.
4:   if  $N = 0$  then
5:     return  $\{\}$ 
6:   end if
7:   if  $x = '0'$  then
8:      $S \leftarrow \{'0', '1', 'a'\}$ 
9:   else if  $x = '1'$  then
10:     $S \leftarrow \{'1', 'a'\}$ 
11:  else
12:     $S \leftarrow \{x, x + 1\}$   $\triangleright x + 1$  denotes the symbol after  $x$  in the list  $'a', 'b', 'c', \dots$ 
13:  end if
14:   $R \leftarrow \{\}$ 
15:  for all  $y \in S$  do
16:     $R' \leftarrow \text{INDEXSETS}(N - 1, y)$ 
17:    prepend  $y$  to all elements in  $R'$ 
18:     $R \leftarrow R \cup R'$ 
19:  end for
20:  return  $R$ 
21: end function

```

Using the ordering in (3.4) for complex numbers, one can apply such a rule to GPLs with numeric indices:

$$\begin{aligned}
G\left(\frac{1+i\sqrt{3}}{2}, 0, \frac{1-i\sqrt{3}}{2} \middle| x\right) &= -G\left(0, \frac{1-i\sqrt{3}}{2}, \frac{1+i\sqrt{3}}{2} \middle| x\right) - G\left(0, \frac{1+i\sqrt{3}}{2}, \frac{1-i\sqrt{3}}{2} \middle| x\right) \\
&\quad + G\left(0, \frac{1-i\sqrt{3}}{2} \middle| x\right) G\left(\frac{1+i\sqrt{3}}{2} \middle| x\right).
\end{aligned}$$

We implemented the algorithm described in this section as a C++ code and generated reduction tables up to weight eight. The reduction tables are stored in a single SQLite [83] database with a final size of 11 GB. The database can be accessed from within Mathematica [84] and FORM [52].

3.1.2 Series Expansion

In this section, we present a method for the expansion of GPLs around the arguments $x = 0$ and $x = 1$. Even though the method is inspired by the algorithms described in Ref. [99] for HPLs, it is more than just a generalization to GPLs. In contrast to Ref. [99], our method for the expansion around $x = 0$ can be applied to GPLs which still exhibit a singular behavior for small x . It is not required to extract this behavior beforehand. The proper expansion of such GPLs is therefore not a Taylor expansion but an expansion in x and $\ln(x)$. Further, we developed a new algorithm for an expansion around $x = 1$ that is more general than the one implemented in the Mathematica-package HPL [101]. There, relations between HPLs at argument x and $1 - x$ are exploited. Unfortunately, these identities exist only in certain cases, i.e. if all indices of the HPLs are either 0 or 1. Our method does not have this limitation.

Algorithm 3.3 Reduction algorithm

```

1: function REDUCTION( $a$ )
2:   in:  $a$ : a set of indices generated by INDEXSETS
3:   out: reduction tables for all GPLs, where the indices are permutations of  $a$ .
4:    $S \leftarrow$  all distinct permutations of  $a$ 
5:    $P \leftarrow \{\}$ 
6:   for all  $s \in S$  do
7:      $P \leftarrow P \cup \text{CUT}(s)$ 
8:   end for
9:    $L \leftarrow \{\}$ 
10:  for all  $p \in P$  do  $\triangleright p$  is a list of lists of indices
11:     $r \leftarrow \prod_{b \in p} G(b|x)$ 
12:     $l \leftarrow$  right-hand side of the shuffle identity (3.2) applied to  $r$ 
13:    apply lower weight reduction tables to  $r$ 
14:     $L \leftarrow L \cup \{l = r\}$ 
15:  end for
16:  return the solution of  $L$  for the highest GPLs of weight  $|a|$  with respect to the ordering
    defined by (3.5)
17: end function
    
```

Expansion around $x = 0$

A GPL allows an expansion around $x = 0$ of the form

$$G(\vec{a}|x) = \sum_{n=0}^{\infty} \sum_{m=0}^{\hat{m}} C_{nm}(\vec{a}) x^n \ln^m(x), \quad (3.6)$$

where \hat{m} denotes the highest possible power of $\ln(x)$ and is given by the number of trailing zeros in \vec{a} . Plugging this expansion into (3.1) yields

$$G(b, \vec{a}|x) = \sum_{n=0}^{\infty} \sum_{m=0}^{\hat{m}} C_{nm}(\vec{a}) \int_0^x dt \frac{t^n}{t-b} \ln^m(t). \quad (3.7)$$

For $b = 0$, (3.7) simplifies to

$$\begin{aligned} G(0, \vec{a}|x) &= \sum_{n=0}^{\infty} \sum_{m=0}^{\hat{m}} C_{nm}(\vec{a}) \int_0^x dt t^{n-1} \ln^m(t) \\ &= - \sum_{n=1}^{\infty} \sum_{j=0}^{\hat{m}} x^n \ln^j(x) \sum_{m=j}^{\hat{m}} C_{nm}(\vec{a}) \frac{m!}{j!} (-n)^{j-m-1}, \end{aligned} \quad (3.8)$$

where we used

$$\int_0^x dt t^n \ln^m(t) = -m! x^{n+1} \sum_{j=0}^m \frac{(-1-n)^{j-m-1}}{j!} \ln^j(x) \quad ; \quad n \geq 0. \quad (3.9)$$

Equation (3.8) is of course only valid if $C_{0m}(\vec{a}) = 0$ for all m . This condition is fulfilled if $\vec{a} \neq \vec{0}$. Comparing (3.8) to (3.6) yields the recurrence relations

$$C_{0m}(0, \vec{a}) = 0 \quad ; \quad \vec{a} \neq \vec{0} \quad (3.10)$$

$$C_{nj}(0, \vec{a}) = - \sum_{m=j}^{\hat{m}} C_{nm}(\vec{a}) \frac{m!}{j!} (-n)^{j-m-1} \quad ; \quad \vec{a} \neq \vec{0}, n > 0. \quad (3.11)$$

For the case $b \neq 0$ in (3.7), we first expand

$$\frac{1}{t-b} = - \sum_{i=0}^{\infty} \frac{t^i}{b^{i+1}}.$$

This yields

$$\begin{aligned} G(b, \vec{a}|x) &= - \sum_{n=0}^{\infty} \sum_{i=0}^{\infty} \sum_{m=0}^{\hat{m}} \frac{C_{nm}(\vec{a})}{b^{i+1}} \int_0^x dt t^{i+n} \ln^m(t) \\ &= \sum_{i=1}^{\infty} \sum_{j=0}^{\hat{m}} x^i \ln^j(x) \sum_{n=0}^{i-1} \sum_{m=j}^{\hat{m}} \frac{C_{nm}(\vec{a}) m!}{b^{i-n} j!} (-i)^{j-m-1}, \end{aligned}$$

where we again applied (3.9). Thus, we gain the recurrence relations

$$C_{0j}(b, \vec{a}) = 0 \quad ; \quad b \neq 0 \quad (3.12)$$

$$C_{ij}(b, \vec{a}) = \sum_{n=0}^{i-1} \sum_{m=j}^{\hat{m}} \frac{C_{nm}(\vec{a}) m!}{b^{i-n} j!} (-i)^{j-m-1} \quad ; \quad b \neq 0, i > 0. \quad (3.13)$$

The relations (3.10) and (3.12) can now be used to express all coefficients $C_{ij}(\vec{a})$ in terms of coefficients where either all indices are 0, or only a single index is left. There is no difficulty in obtaining these initial values from

$$G(\underbrace{0, \dots, 0}_n | x) = \frac{1}{n!} \ln^n(x), \quad G(b|x) = \ln\left(1 - \frac{x}{b}\right) = - \sum_{n=1}^{\infty} \frac{x^n}{nb^n} \quad ; \quad b \neq 0.$$

This yields

$$\begin{aligned} C_{0m}(0, \underbrace{\dots, 0}_n) &= \frac{1}{m!} \delta_{mn}, \\ C_{n0}(b) &= - \frac{1}{nb^n} \quad ; \quad b \neq 0, n > 0, \end{aligned} \quad (3.14)$$

where all coefficients not explicitly given here vanish.

The recurrence relations given in (3.10) and (3.12) as well as the initial values in (3.14) can easily be implemented in computer algebra systems like `Mathematica` [84] or `FORM` [52].

Expansion around $x = 1$

For an expansion around $x = 1$, we proceed similarly to the expansion method for $x = 0$. The starting point is

$$G(\vec{a}|1-y) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} C_{nm}(\vec{a}) y^n \ln^m(y) \quad ; \quad y \ll 1. \quad (3.15)$$

The maximum index m for which $C_{nm}(\vec{a}) \neq 0$ may occur is determined by counting the indices of value 1 in \vec{a} . Thus, the upper bound of the summation over m in (3.15) is in fact finite. From the definition (3.1), we gain

$$G(b, \vec{a}|1-y) = \int_0^{1-y} \frac{dz}{z-b} G(\vec{a}|z) = \int_0^1 \frac{dz}{z-b} G(\vec{a}|z) + \int_1^{1-y} \frac{dz}{z-b} G(\vec{a}|z),$$

where we require $b \neq 1$ for the moment. The first term is just a GPL with 1 as the argument. For the second term, we substitute $z = 1 - t$. This leads to

$$G(b, \vec{a}|1-y) = G(b, \vec{a}|1) - \int_0^y \frac{dt}{1-b-t} G(\vec{a}|1-t).$$

At this point, the expansion (3.15) can be inserted, which yields

$$G(b, \vec{a}|1-y) = G(b, \vec{a}|1) - \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} C_{nm}(\vec{a}) \int_0^y dt \frac{t^n}{1-b-t} \ln^m(t).$$

Using the geometric series

$$\frac{1}{1-b-t} = \sum_{i=0}^{\infty} \frac{t^i}{(1-b)^{i+1}}$$

leads to

$$\begin{aligned} G(b, \vec{a}|1-y) &= G(b, \vec{a}|1) - \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \sum_{i=0}^{\infty} \frac{C_{nm}(\vec{a})}{(1-b)^{i+1}} \int_0^y dt t^{n+i} \ln^m(t) \\ &= G(b, \vec{a}|1) - \sum_{n=1}^{\infty} \sum_{j=0}^{\infty} \sum_{i=0}^{n-1} \sum_{m=0}^{\infty} \frac{(-1)^m (m+j)!}{(1-b)^{i+1} n^{m+1} j!} C_{n-i-1; m+j}(\vec{a}) y^n \ln^j(y), \end{aligned}$$

where again the integral (3.9) has been used. From this equation, we obtain the relations

$$\begin{aligned} C_{00}(b, \vec{a}) &= G(b, \vec{a}|1) \quad ; \quad b \neq 1, \\ C_{nj}(b, \vec{a}) &= - \sum_{i=0}^{n-1} \sum_{m=0}^{\infty} \frac{(-1)^m (m+j)!}{(1-b)^{i+1} n^{m+1} j!} C_{n-i-1; m+j}(\vec{a}) \quad ; \quad b \neq 1, n > 0. \end{aligned} \tag{3.16}$$

The case $b = 1$ is a bit more involved. Here we start with

$$G(1, \vec{a}|1-y) = \int_0^{1-y} \frac{dz}{z-1} G(\vec{a}|z) = \int_1^y \frac{dt}{t} G(\vec{a}|1-t),$$

where the substitution $z = 1 - t$ has been performed. Now, (3.15) can be inserted, which yields

$$G(1, \vec{a}|1-y) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} C_{nm}(\vec{a}) \int_1^y dt t^{n-1} \ln^m(t).$$

Using

$$\begin{aligned} \int_1^y dt t^n \ln^m(t) &= -m! y^{n+1} \sum_{j=0}^m \frac{(-1-n)^{j-m-1}}{j!} \ln^j(y) + m! (-1-n)^{-m-1} \quad ; \quad n \geq 0, \\ \int_1^y dt \frac{\ln^m(t)}{t} &= \frac{\ln^{m+1}(y)}{(m+1)!}, \end{aligned}$$

we obtain

$$\begin{aligned}
 G(1, \vec{a}|1-y) &= - \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \frac{(-1)^m m!}{n^{m+1}} C_{nm}(\vec{a}) + \sum_{m=1}^{\infty} \frac{1}{m} C_{0;m-1}(\vec{a}) \ln^m(y) \\
 &\quad + \sum_{n=1}^{\infty} \sum_{j=0}^{\infty} \sum_{m=0}^{\infty} \frac{(-1)^m (m+j)!}{n^{m+1} j!} C_{n;m+j}(\vec{a}) y^n \ln^j(y).
 \end{aligned} \tag{3.17}$$

Comparing (3.17) to (3.15) yields the recurrence relations

$$\begin{aligned}
 C_{0m}(1, \vec{a}) &= \frac{1}{m} C_{0;m-1}(\vec{a}) \quad ; \quad m > 0, \\
 C_{nj}(1, \vec{a}) &= \sum_{m=0}^{\infty} \frac{(-1)^m (m+j)!}{n^{m+1} j!} C_{n;m+j}(\vec{a}) \quad ; \quad n > 0.
 \end{aligned} \tag{3.18}$$

The constant term of expansion (3.15) has yet to be determined. Unfortunately, the equation

$$C_{00}(1, \vec{a}) = - \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \frac{(-1)^m m!}{n^{m+1}} C_{nm}(\vec{a}),$$

which can be obtained from (3.17) as well, is of no use since it sums over an infinite number of terms. A better way to determine $C_{00}(1, \vec{a})$ is to consider (3.15) near $y = 0$, which can be written as

$$G(\vec{a}|1-y) \approx C_{00}(\vec{a}) + \sum_{m=1}^k C_{0m}(\vec{a}) \ln^m(y). \tag{3.19}$$

Here, k is the number of indices at the beginning of \vec{a} that are 1. The shuffle relation (3.2) indicates that a GPL where the first k indices are 1,

$$G(\underbrace{1, \dots, 1}_k, \vec{a}|1-y),$$

can be expressed in terms of GPLs of the same weight plus a term of the form

$$G(\underbrace{1, \dots, 1}_k | 1-y) G(\vec{a}|1-y) = \frac{\ln^k(y)}{k!} G(\vec{a}|1-y).$$

This extra term is exactly the $\ln^k(y)$ contribution to (3.19). From this observation, one can deduce a strategy to extract $C_{00}(1, \vec{a})$, given in alg. 3.4.

The recurrence relations (3.16) and (3.18) can be used to reduce all coefficients to coefficients for GPLs with only a single index. From

$$G(b|1-y) = \ln\left(\frac{b-1}{b}\right) + \ln\left(1 - \frac{y}{1-b}\right) = G(b|1) - \sum_{n=1}^{\infty} \frac{y^n}{n(1-b)^n} \quad ; \quad b \neq 0 \wedge b \neq 1,$$

$$G(0|1-y) = \ln(1-y) = - \sum_{n=1}^{\infty} \frac{y^n}{n},$$

$$G(1|1-y) = \ln(y),$$

Algorithm 3.4 Extraction of the constant term $C_{00}(1, \dots, 1, \vec{a})$

1. Consider a GPL, where the first k indices are 1.
2. Subtract the $\ln^k(y)$ contribution as discussed in the text via

$$G(\underbrace{1, \dots, 1}_k, \vec{a}|1-y) - G(\underbrace{1, \dots, 1}_k |1-y)G(\vec{a}|1-y). \quad (3.20)$$

3. Apply the shuffle relation (3.2) to the second term of (3.20). The shuffle relation will generate a term

$$-G(\underbrace{1, \dots, 1}_k, \vec{a}|1-y)$$

which cancels the first term of (3.20).

4. The result consists only of GPLs, where less than k indices at the beginning are 1. Proceed with step 1 for all GPLs for which the indices still start with 1.
5. As soon as all GPLs start with an index $\neq 1$, set $y = 0$. This will result in the coefficient

$$C_{00}(\underbrace{1, \dots, 1}_k, \vec{a}).$$

one can extract the initial values

$$\begin{aligned} C_{00}(b) &= G(b|1) \quad ; \quad b \neq 0 \wedge b \neq 1 \\ C_{n0}(b) &= -\frac{1}{n(1-b)^n} \quad ; \quad b \neq 0 \wedge b \neq 1, n > 0, \\ C_{n0}(0) &= -\frac{1}{n} \quad ; \quad n > 0, \\ C_{01}(1) &= 1, \\ C_{nm}(b) &= 0 \quad ; \quad \text{else.} \end{aligned} \quad (3.21)$$

The constant terms $C_{00}(\vec{a})$ are expressed in terms of GPLs at argument 1. If the GPLs are in fact HPLs (i.e. only indices $-1, 0, 1$ appear) they can be evaluated to multiple zeta values using tables from Ref. [99] or Ref. [101].

The recurrence relations given in (3.16) and (3.18), the initial values in (3.21) and the algorithm alg. 3.4 can easily be implemented in computer algebra systems like *Mathematica* [84] or *FORM* [52].

3.1.3 Analytic Continuation

Even though the definition (3.1) is only valid for $\text{Re}(x) > 0$, physical problems often require GPLs for $\text{Re}(x) < 0$. Luckily, the analytic continuation to negative arguments is quite simple. We define

$$x = -t + i0,$$

where $t > 0$. The GPL

$$G(\vec{a}, b|x) = G(\vec{a}, b| -t + i0) \quad ; \quad b \neq 0$$

can now be transformed to a GPL at argument t using the scaling identity (3.3) with $k = -1$:

$$G(\vec{a}, b|x) = G(-\vec{a}, -b|t). \quad (3.22)$$

For GPLs with trailing zeros, one first has to use the shuffle relation (3.2) (or the algorithm proposed in Section 3.1.1) to extract the singular behavior at $x = 0$. This leads to an expression in terms of GPLs which either do not have trailing zeros or are of the form $G(0, \dots, 0|x)$. For the GPLs without trailing zeros, (3.22) can be applied. If we consider GPLs of the form $G(0, \dots, 0|x)$, we can use the definition (3.1)

$$G(\underbrace{0, \dots, 0}_n|x) = \frac{1}{n!} \ln^n(x) = \frac{1}{n!} \ln^n(-t + i0) = \frac{1}{n!} [\ln(t) + i\pi]^n,$$

where we had to take the branch cuts of the complex logarithm into account.

3.2 The Method of Differential Equations

The reduction to master integrals based on IBP and LI identities as well as on sector relations, discussed in Section 2.6, can provide more than just a minimization of the number of Feynman integrals to be evaluated. It also contains useful information for the actual calculation of master integrals. This information can be utilized with the method of differential equations [86–88].

A scalar Feynman integral $F(\text{ID}; a_1, \dots, a_N)$ defined by (2.7) only depends on the external momenta q_1 and q_2 through the kinematic variable $\tau = (q_1 \cdot q_2)/2$. The differential operator with respect to the external momentum q_1 acting on a function of τ can be written as⁴

$$\frac{\partial}{\partial q_1^\mu} \equiv \frac{\partial \tau}{\partial q_1^\mu} \frac{\partial}{\partial \tau} = \frac{q_{2\mu}}{2} \frac{\partial}{\partial \tau}.$$

Hence, the derivative with respect to τ is given by

$$\frac{\partial}{\partial \tau} \equiv \frac{q_1^\mu}{\tau} \frac{\partial}{\partial q_1^\mu}.$$

The derivative of $F(\text{ID}; a_1, \dots, a_N)$ can now be evaluated by commuting the differential operator with the integral signs, i.e.

$$\frac{\partial}{\partial \tau} F(\text{ID}; a_1, \dots, a_N) = \frac{1}{\tau} \int_{l_1} \dots \int_{l_K} q_1^\mu \frac{\partial}{\partial q_1^\mu} \frac{1}{D_1^{a_1} \dots D_N^{a_N}}.$$

This is equivalent to

$$\frac{\partial}{\partial \tau} F(\text{ID}; a_1, \dots, a_N) = -\frac{1}{\tau} \sum_{i=1}^N a_i \int_{l_1} \dots \int_{l_K} \frac{q_1^\mu \frac{\partial D_i}{\partial q_1^\mu}}{D_1^{a_1} \dots D_{j-1}^{a_{j-1}} D_j^{a_j+1} D_{j+1}^{a_{j+1}} \dots D_N^{a_N}}, \quad (3.23)$$

where the numerator of the integrand is a sum of scalar products. Note that every scalar product involving a loop-momentum can be expressed as linear combination of the D_j . Hence, the right-hand side of (3.23) is a linear combination of functions F with the same ID as on the left-hand side but with shifted indices. These scalar Feynman integrals can now be reduced to master integrals, using the reduction procedure (or more accurately the reduction database) described in Section 2.6.

⁴Of course, the same considerations can also be carried out for q_2 .

Now, let

$$\vec{f} \equiv \begin{pmatrix} I_1 \\ \vdots \\ I_N \end{pmatrix}$$

be an N -dimensional vector, where the components I_1, \dots, I_N are scalar Feynman integrals building a basis of master integrals. Applying the reduction tables to $\partial \vec{f} / \partial \tau$ leads to a linear combination of the same master integrals I_1, \dots, I_N for every component. Hence, this derivative can be written as

$$\frac{\partial \vec{f}}{\partial \tau} = \mathbb{K}(\tau, \epsilon) \vec{f},$$

where $\mathbb{K}(\tau, \epsilon)$ is an $N \times N$ -matrix of rational functions in τ and $\epsilon = (4 - d)/2$, where d is the number of space-time dimensions.

A change of variable

$$\frac{\partial}{\partial x} \equiv \frac{\partial \tau}{\partial x} \frac{\partial}{\partial \tau} = \frac{1 - x^2}{4x^2} \frac{\partial}{\partial \tau}$$

leads to a coupled system of differential equations with respect to the variable x , defined in (2.8),

$$\frac{\partial \vec{f}}{\partial x} = \mathbb{M}(x, \epsilon) \vec{f}, \tag{3.24}$$

where

$$\mathbb{M}(x, \epsilon) = \frac{1 - x^2}{4x^2} \mathbb{K} \left(-\frac{(1 - x)^2}{4x}, \epsilon \right)$$

is a matrix of rational functions in x and ϵ .

A change of basis

$$\vec{g} = \mathbb{T}^{-1}(x, \epsilon) \vec{f}$$

modifies the system (3.24) to

$$\frac{\partial \vec{g}}{\partial x} = \widetilde{\mathbb{M}}(x, \epsilon) \vec{g},$$

with

$$\widetilde{\mathbb{M}}(x, \epsilon) = \mathbb{T}^{-1}(x, \epsilon) \mathbb{M}(x, \epsilon) \mathbb{T}(x, \epsilon) - \mathbb{T}^{-1}(x, \epsilon) \frac{\partial}{\partial x} \mathbb{T}(x, \epsilon). \tag{3.25}$$

In many applications, it is useful to order the master integrals in \vec{f} in such a way that the matrix $\mathbb{M}(x, \epsilon)$ becomes block-triangular. It is not difficult to find this ordering. The right-hand side of (3.23) will only contain scalar Feynman integrals of the same family and sector as the Feynman integral on the left-hand side and subsectors of this sector. Also, the reduction to master integrals will only lead to integrals of the same sector or sectors with a smaller number of positive indices a_j . It is therefore easy to understand that the desired structure is reached if we order the master integrals by the number of positive indices and keep integrals of the same sector together.

The system of differential equations (3.24) contains the full kinematic dependence of the master integrals I_1, \dots, I_N . If the general solution to this system is known, only the boundary conditions have to be fixed to obtain a full result. A possible method to fix the boundary conditions is given in Section 3.4.

3.3 A Canonical Basis for Master Integrals

In 2013, Henn proposed a special basis, the canonical basis, for a certain class of master integrals, which are expressible in terms of iterated integrals [102]. In this basis, the master integrals obey a differential equation of the form (3.24), where $\mathbb{M}(x, \epsilon)$ is proportional to ϵ . This form of a differential equation makes the solution of the master integrals in an ϵ -expansion trivial if the boundary conditions are known.

In 2014, Lee proposed an algorithm to automate finding a canonical basis [63]. A first implementation of this algorithm was presented in Ref. [103, 104].

In this section we present `epsilon`, a further implementation of Lee's algorithm, developed as part of this thesis and published in Ref. [1]. The tool `epsilon` is written in C++ and is based on the computer algebra system `Fermat` [66]. Our implementation utilizes the explicit dependence of the transformations used by Lee's algorithm on the kinematic variable to reduce the number of variables in intermediate steps. Another advantage of our implementation is the support of systems with singularities at complex points using `Fermat`'s `polymod` capability.

In Section 3.3.1, we introduce some definitions. In Section 3.3.2, the utilization of the explicit x -dependence of the transformations is explained. In Section 3.3.3, we give an overview of Lee's algorithm. Section 3.3.4 finally explains how to solve master integrals in a canonical basis.

The installation procedure and the usage of `epsilon` is described in Appendix D.1.

3.3.1 Definitions

We consider a set of N master integrals in \vec{f} fulfilling an ordinary system of differential equations

$$\frac{\partial \vec{f}(x, \epsilon)}{\partial x} = \mathbb{M}(x, \epsilon) \vec{f}(x, \epsilon), \quad (3.26)$$

where x is a kinematic variable, $\mathbb{M}(x, \epsilon)$ is an $N \times N$ -matrix and ϵ is a regulator in $d = 4 - 2\epsilon$ dimensions in the context of dimensional regularization (see Section 3.2). We restrict ourselves to the case

$$\mathbb{M}(x, \epsilon) = \sum_{x_j \in S} \sum_{k \geq 0} \frac{\mathbb{M}_k^{(x_j)}(\epsilon)}{(x - x_j)^{k+1}} + \sum_{k \geq 0} x^k \mathbb{M}_k(\epsilon), \quad (3.27)$$

where S is the set of all finite singularities and $\mathbb{M}_k^{(x_j)}$ and $\mathbb{M}_k(\epsilon)$ are independent of x . In particular, singularities x_j depending on ϵ are forbidden. In many physically relevant cases one can use a trial and error approach to find a basis of master integrals \vec{f} fulfilling the restriction (3.27). The main strategy of our implementation is to keep the system always in the form of (3.27) since here the x -dependence is explicit.

A singularity $|x_j| < \infty$ has Poincaré rank p if $\mathbb{M}_p^{(x_j)} \neq 0$ and $\mathbb{M}_k^{(x_j)} = 0$ for $k > p$. In addition to the finite singularities, the system might also have a singularity at ∞ . The Poincaré rank p of a singularity at ∞ is defined as the Poincaré rank of the singularity at $y = 0$ of the system $\mathbb{M}(1/y, \epsilon)/y^2$.

So (3.27) has Poincaré rank $p > 0$ at ∞ if $\mathbb{M}_{p-1} \neq 0$ and $\mathbb{M}_k = 0$ for $k \geq p$, and Poincaré rank $p = 0$ at ∞ if all $\mathbb{M}_k = 0$ and $\sum_{x_j \in S} \mathbb{M}_0^{(x_j)} \neq 0$. If all $\mathbb{M}_k = 0$ and $\sum_{x_j \in S} \mathbb{M}_0^{(x_j)} = 0$, the system is not singular at ∞ .

Let p be the Poincaré rank of a singularity $x_j < \infty$, then the generalized Poincaré rank (or Moser rank) [105] of this singularity is defined as $(p - 1 + r/n)$, where $r = \text{rank}(\mathbb{M}_p^{(x_j)})$ and n is the dimension of the system.

A system

$$\mathbf{M}(x, \epsilon) = \sum_{x_j \in S} \frac{\mathbf{M}_0^{(x_j)}(\epsilon)}{x - x_j}, \quad (3.28)$$

where all singularities have Poincaré rank zero is called Fuchsian, and a system

$$\mathbf{M}(x, \epsilon) = \epsilon \sum_{x_j \in S} \frac{\widehat{\mathbf{M}}_0^{(x_j)}}{x - x_j}, \quad (3.29)$$

where $\widehat{\mathbf{M}}_0^{(x_j)}$ is no longer a function of ϵ , is said to be in ϵ -form.

We assume the master integrals in \vec{f} to be ordered such that a block-triangular structure of the system is obtained (for details see Section 3.2). We will often make use of this block-triangular structure. Therefore, we write

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & 0 & 0 \\ \mathbf{B} & \mathbf{C} & 0 \\ \mathbf{D} & \mathbf{E} & \mathbf{F} \end{pmatrix}, \quad (3.30)$$

and use the same indices as in (3.27) for the matrices $\mathbf{A}, \dots, \mathbf{F}$ (e.g. $\mathbf{C}_k^{(x_j)}(\epsilon)$). The block \mathbf{C} is called the active block as we apply Lee's algorithm to this block. As \mathbf{A} to \mathbf{F} are matrices as well, the definition of what we call the active block is more or less arbitrary as long as a block-triangular structure is obtained. But from a computational point of view a small dimension of the active block is preferable since this reduces the complexity of the resulting operations. In the following, the matrices $\mathbf{A}_k^{(x_j)}, \dots, \mathbf{F}_k^{(x_j)}$ and $\mathbf{A}_k, \dots, \mathbf{F}_k$ will be referred to as coefficient matrices.

3.3.2 Utilizing the Explicit x -Dependence

Lee's algorithm uses three types of transformations: balances, off-diagonal reductions and x -independent transformations.

We define balances as

$$\mathcal{B}(\mathbf{P}, x_1, x_2) = \overline{\mathbf{P}} + \frac{x - x_2}{x - x_1} \mathbf{P}, \quad (3.31a)$$

$$\mathcal{B}(\mathbf{P}, x_1, \infty) = \overline{\mathbf{P}} + \frac{1}{x - x_1} \mathbf{P}, \quad (3.31b)$$

$$\mathcal{B}(\mathbf{P}, \infty, x_2) = \overline{\mathbf{P}} + (x - x_2) \mathbf{P}, \quad (3.31c)$$

where \mathbf{P} is a projector to be specified below, depending only on ϵ , $\overline{\mathbf{P}} = \mathbf{1} - \mathbf{P}$ and $x_1, x_2 < \infty$. In Lee's algorithm balances are applied to the active block \mathbf{C} in order to reduce the generalized Poincaré rank of singular points and to normalize eigenvalues of Fuchsian singularities.

Off-diagonal reductions are used to reduce the block \mathbf{B} to Fuchsian form after the blocks \mathbf{A} and \mathbf{C} were already reduced to ϵ -form. They are defined by

$$\mathcal{L}(x_1, k, \mathbf{G}) = \mathbf{1} + \frac{1}{(x - x_1)^k} \mathbf{G}, \quad (3.32a)$$

$$\mathcal{L}(\infty, k, \mathbf{G}) = \mathbf{1} + x^k \mathbf{G}, \quad (3.32b)$$

where

$$\mathbf{G} = \begin{pmatrix} 0 & 0 & 0 \\ \widehat{\mathbf{G}} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (3.32c)$$

The block $\widehat{\mathbb{G}}$ has the same boundaries in \mathbb{G} as block \mathbb{B} in (3.30). Note that $\mathbb{G}^2 = 0$.

In both types of transformations the x -dependence is explicit. Another type of transformation, which is independent of x , is used in the last step of Lee's algorithm to factor out ϵ .

Our goal is to use those three types in the transformation rule (3.25) without spoiling the form (3.27) or the block-triangular structure (3.30) of the system.

As a pedagogical example we consider the transformation of block \mathbb{B} under a balance between two singularities x_1 and x_2 , i.e.

$$\mathbb{T} = \mathcal{B}(\mathbb{P}, x_1, x_2) = \overline{\mathbb{P}} + \frac{x - x_2}{x - x_1} \mathbb{P}, \quad \mathbb{T}^{-1} = \mathcal{B}(\mathbb{P}, x_2, x_1) = \overline{\mathbb{P}} + \frac{x - x_1}{x - x_2} \mathbb{P}. \quad (3.33)$$

Since we want to apply Lee's algorithm to the active block, we can restrict the form of the projector \mathbb{P} to

$$\mathbb{P} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \mathbb{Q} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (3.34)$$

where \mathbb{Q} is a projector with the dimensions of the active block. Inserting (3.33) into (3.25) we obtain

$$\begin{aligned} \widetilde{\mathbb{M}}(x, \epsilon) &= \left[\overline{\mathbb{P}} + \frac{x - x_1}{x - x_2} \mathbb{P} \right] \mathbb{M}(x, \epsilon) \left[\overline{\mathbb{P}} + \frac{x - x_2}{x - x_1} \mathbb{P} \right] + \frac{x_2 - x_1}{(x - x_1)(x - x_2)} \mathbb{P} \\ &= \mathbb{M}(x, \epsilon) - \mathbb{P} \mathbb{M}(x, \epsilon) \overline{\mathbb{P}} - \overline{\mathbb{P}} \mathbb{M}(x, \epsilon) \mathbb{P} + \frac{x - x_2}{x - x_1} \overline{\mathbb{P}} \mathbb{M}(x, \epsilon) \mathbb{P} + \frac{x - x_1}{x - x_2} \mathbb{P} \mathbb{M}(x, \epsilon) \overline{\mathbb{P}} \\ &\quad + \frac{x_2 - x_1}{(x - x_1)(x - x_2)} \mathbb{P}. \end{aligned}$$

So block \mathbb{B} in (3.30) transforms as

$$\widetilde{\mathbb{B}}(x, \epsilon) = \mathbb{B}(x, \epsilon) - \mathbb{Q} \mathbb{B}(x, \epsilon) + \frac{x - x_1}{x - x_2} \mathbb{Q} \mathbb{B}(x, \epsilon).$$

Inserting the form (3.27) of $\mathbb{B}(x, \epsilon)$ leads to

$$\widetilde{\mathbb{B}}(x, \epsilon) = \mathbb{B}(x, \epsilon) - \mathbb{Q} \mathbb{B}(x, \epsilon) + \sum_{x_j \in S} \sum_{k \geq 0} \frac{(x - x_1) \mathbb{Q} \mathbb{B}_k^{(x_j)}(\epsilon)}{(x - x_2)(x - x_j)^{k+1}} + \sum_{k \geq 0} \frac{(x - x_1) x^k}{x - x_2} \mathbb{Q} \mathbb{B}_k(\epsilon). \quad (3.35)$$

Using a partial fraction decomposition and the incomplete geometric series, we can show that

$$\begin{aligned} \sum_{k \geq 0} \frac{a_k}{(x - x_2)(x - x_j)^{k+1}} &= \frac{1}{x - x_2} \sum_{n \geq 0} \frac{a_n}{(x_2 - x_j)^{n+1}} \\ &\quad - \sum_{k \geq 0} \frac{1}{(x - x_j)^{k+1}} \sum_{n \geq 0} \frac{a_{n+k}}{(x_2 - x_j)^{n+1}}, \end{aligned} \quad (3.36a)$$

$$\sum_{k \geq 0} \frac{x^k}{x - x_2} a_k = \sum_{k \geq 0} x^k \sum_{n \geq 0} x_2^n a_{k+n+1} + \frac{1}{x - x_2} \sum_{n \geq 0} x_2^n a_n, \quad (3.36b)$$

where identity (3.36a) only holds for $x_j \neq x_2$. Combining (3.35) and (3.36) yields

$$\begin{aligned} \widetilde{\mathbb{B}}(x, \epsilon) &= \mathbb{B}(x, \epsilon) + \frac{x_2 - x_1}{x - x_2} \sum_{x_j \in S \setminus \{x_2\}} \sum_{n \geq 0} \frac{\mathbb{Q} \mathbb{B}_n^{(x_j)}}{(x_2 - x_j)^{n+1}} + \frac{x_2 - x_1}{x - x_2} \sum_{n \geq 0} x_2^n \mathbb{Q} \mathbb{B}_n \\ &\quad + \sum_{k \geq 1} \frac{(x_2 - x_1) \mathbb{Q} \mathbb{B}_{k-1}^{(x_2)}(\epsilon)}{(x - x_2)^{k+1}} + \sum_{x_j \in S \setminus \{x_2\}} \sum_{k \geq 0} \frac{x_1 - x_2}{(x - x_j)^{k+1}} \sum_{n \geq 0} \frac{\mathbb{Q} \mathbb{B}_{n+k}^{(x_j)}(\epsilon)}{(x_2 - x_j)^{n+1}} \\ &\quad + (x_2 - x_1) \sum_{k \geq 0} x^k \sum_{n \geq 0} x_2^n \mathbb{Q} \mathbb{B}_{k+n+1}(\epsilon). \end{aligned} \quad (3.37)$$

Hence, the transformation laws for the coefficient matrices can be found by comparing (3.37) with the structure of (3.27):

$$\begin{aligned}\tilde{\mathbb{B}}_0^{(x_2)}(\epsilon) &= \mathbb{B}_0^{(x_2)}(\epsilon) + \sum_{x_j \in S \setminus \{x_2\}} \sum_{n \geq 0} \frac{x_2 - x_1}{(x_2 - x_j)^{n+1}} \mathbb{Q}\mathbb{B}_n^{(x_j)}(\epsilon) + (x_2 - x_1) \sum_{n \geq 0} x_2^n \mathbb{Q}\mathbb{B}_n(\epsilon), \\ \tilde{\mathbb{B}}_{k>0}^{(x_2)}(\epsilon) &= \mathbb{B}_k^{(x_2)}(\epsilon) + (x_2 - x_1) \mathbb{Q}\mathbb{B}_{k-1}^{(x_2)}(\epsilon), \\ \tilde{\mathbb{B}}_k^{(x_j \neq x_2)}(\epsilon) &= \mathbb{B}_k^{(x_j)}(\epsilon) + \sum_{n \geq 0} \frac{x_1 - x_2}{(x_2 - x_j)^{n+1}} \mathbb{Q}\mathbb{B}_{n+k}^{(x_j)}(\epsilon), \\ \tilde{\mathbb{B}}_k(\epsilon) &= (x_2 - x_1) \sum_{n \geq 0} x_2^n \mathbb{Q}\mathbb{B}_{k+n+1}(\epsilon).\end{aligned}$$

An advantage of this form over the original transformation (3.33) is that now all operations are independent of x . Therefore, the underlying computer algebra system has to deal with rational functions of one variable less. The form (3.27) remains unspoiled, i.e. it is not necessary to perform a partial fraction decomposition after the transformation.

All transformations in terms of the coefficient matrices are listed in Appendix D.2.

3.3.3 Overview of Lee's algorithm

Three basic steps allow Lee's algorithm [63] to transform an ordinary system of differential equations into an ϵ -form (3.29) if they are applied to the whole system:

1. transformation of a system into Fuchsian form,
2. normalization of the eigenvalues of all matrix residues,
3. factorization of ϵ .

In order to make use of the block-triangular structure of the system (3.30), these three steps are applied only to the active block followed by a fourth step to transform the off-diagonal block \mathbb{B} into Fuchsian form.

In this section, we briefly discuss all four steps. More details can be found in the original paper by Lee [63].

Fuchsification

The basic building blocks for the first part of Lee's algorithm, the transformation of the system to Fuchsian form (3.28), are the balances defined in (3.31). With the right choice of a projector \mathbb{P} , it is possible to perform a so-called Moser reduction to strictly lower the generalized Poincaré rank of the singularity at x_1 [105]. In this discussion, we restrict ourselves to the case where $|x_1| < \infty$. A more general treatment is given in Ref. [63].

Let p be the Poincaré rank of the singularity at x_1 of the active block \mathbb{C} and $\{u_k^\alpha\}$ with $\alpha = 0, \dots, n_k$ be the set of $n_k + 1$ right generalized eigenvectors of $\mathbb{C}_p^{(x_1)}$ belonging to a Jordan block k in the Jordan decomposition of $\mathbb{C}_p^{(x_1)}$. The Jordan blocks are ordered by their size so that $n_i \geq n_{i+1}$. If $p > 0$, we assume all eigenvalues of $\mathbb{C}_p^{(x_1)}$ to be zero, or else no transformation to lower the generalized Poincaré rank exists. The right generalized eigenvectors fulfill

$$\mathbb{C}_p^{(x_1)} u_k^{(0)} = 0, \quad \mathbb{C}_p^{(x_1)} u_k^{(\alpha+1)} = u_k^{(\alpha)}.$$

These relations are invariant under the transformation

$$u_k^{(\alpha)} \rightarrow u_k^{(\alpha)} + c u_l^{(\alpha)}, \tag{3.38}$$

where $\alpha = 0, \dots, n_k$ and $k > l$. The left generalized eigenvectors $v_k^{(\alpha)}$ are related to the right generalized eigenvectors by

$$\left(v_1^{(n_1)}, \dots, v_1^{(0)}, v_2^{(n_2)}, \dots, v_2^{(0)}, \dots \right) = \left[\left(u_1^{(0)}, \dots, u_1^{(n_1)}, u_2^{(0)}, \dots, u_2^{(n_2)}, \dots \right)^{-1} \right]^\dagger, \quad (3.39)$$

and fulfill

$$v_k^{(0)\dagger} \mathbb{C}_p^{(x_1)} = 0, \quad v_k^{(\alpha+1)\dagger} \mathbb{C}_p^{(x_1)} = v_k^{(\alpha)\dagger}.$$

The transformation (3.38) allows us to find a basis of eigenvectors that satisfies

$$v_j^{(0)\dagger} \mathbb{C}_{p-1}^{(x_1)} u_k^{(0)} = 0, \quad (3.40)$$

for $j \notin R$ and $k \in R \cup \{k_0\}$, where R is some set of trivial Jordan blocks ($n_i = 0$ for $i \in R$) and k_0 is a non-trivial Jordan block ($n_{k_0} > 0$). An algorithm to find these generalized eigenvectors together with the set R and k_0 is given in [63].

If in the definition of \mathbb{P} (3.34) we use

$$\mathbb{Q} = \sum_{k \in R \cup \{k_0\}} u_k^{(0)} w_k^\dagger, \quad (3.41)$$

with $w_j^\dagger u_k^{(0)} = \delta_{jk}$ and then apply the balance (3.31a), the generalized Poincaré rank of the singularity at x_1 is decreased. To see this, we introduce the projector

$$\mathbb{Q}_1 = \sum_{k \in R \cup \{k_0\}} u_k^{(0)} v_k^{(n_k)\dagger}. \quad (3.42)$$

Since $w_j^\dagger u_k^{(0)} = \delta_{jk}$, the projector \mathbb{Q}_1 also is of the form (3.41). From the definitions of \mathbb{Q} and \mathbb{Q}_1 follows

$$\mathbb{Q}_1 \mathbb{Q} = \mathbb{Q}, \quad \mathbb{Q} \mathbb{Q}_1 = \mathbb{Q}_1, \quad \mathbb{C}_p^{(x_1)} \mathbb{Q} = \mathbb{C}_p^{(x_1)} \mathbb{Q}_1 = 0.$$

Evaluating the transformation (D.1b) at $k = p$ gives the transformation of $\mathbb{C}_p^{(x_1)}$ for a balance between x_1 and x_2 . This yields

$$\begin{aligned} \tilde{\mathbb{C}}_p^{(x_1)} &= \overline{\mathbb{Q}} \mathbb{C}_p^{(x_1)} + (x_1 - x_2) \overline{\mathbb{Q}} \mathbb{C}_{p-1}^{(x_1)} \mathbb{Q} \\ &= (\overline{\mathbb{Q}} + \mathbb{Q}_1) \left[\overline{\mathbb{Q}}_1 \mathbb{C}_p^{(x_1)} + (x_1 - x_2) \overline{\mathbb{Q}}_1 \mathbb{C}_{p-1}^{(x_1)} \mathbb{Q}_1 \right] (\overline{\mathbb{Q}}_1 + \mathbb{Q}) \end{aligned}$$

Since $(\overline{\mathbb{Q}} + \mathbb{Q}_1) = (\overline{\mathbb{Q}}_1 + \mathbb{Q})^{-1}$, the matrix rank of $\tilde{\mathbb{C}}_p^{(x_1)}$ is given by

$$\text{rank } \tilde{\mathbb{C}}_p^{(x_1)} = \text{rank } \widehat{\mathbb{C}}_p^{(x_1)}, \quad \widehat{\mathbb{C}}_p^{(x_1)} = \overline{\mathbb{Q}}_1 \mathbb{C}_p^{(x_1)} + (x_1 - x_2) \overline{\mathbb{Q}}_1 \mathbb{C}_{p-1}^{(x_1)} \mathbb{Q}_1.$$

The argument that the matrix rank (and therefore the generalized Poincaré rank) of $\tilde{\mathbb{C}}_p^{(x_1)}$ is lower than the matrix rank of $\mathbb{C}_p^{(x_1)}$ is as follows:

- All left eigenvectors $v_j^{(0)}$ of $\mathbb{C}_p^{(x_1)}$ with $j \in R$ are left eigenvectors of $\widehat{\mathbb{C}}_p^{(x_1)}$ since $v_j^{(0)\dagger} \overline{\mathbb{Q}}_1 = 0$.
- All left eigenvectors $v_j^{(0)}$ of $\mathbb{C}_p^{(x_1)}$ with $j \notin R$ are left eigenvectors of $\widehat{\mathbb{C}}_p^{(x_1)}$ since $v_j^{(0)\dagger} \overline{\mathbb{Q}}_1 = v_j^{(0)\dagger}$. Because of (3.40), we obtain

$$v_j^{(0)\dagger} \widehat{\mathbb{C}}_p^{(x_1)} = (x_1 - x_2) \sum_{k \in R \cup \{k_0\}} v_j^{(0)\dagger} \mathbb{C}_{p-1}^{(x_1)} u_k^{(0)} v_k^{(n_k)\dagger} = 0.$$

- The vector $v_{k_0}^{(n_{k_0})}$ which is not a left eigenvector of $\mathbb{C}_p^{(x_1)}$ is an additional left eigenvector of $\widehat{\mathbb{C}}_p^{(x_1)}$ since $v_{k_0}^{(n_{k_0})\dagger}\overline{\mathbb{Q}}_1 = 0$.

So $\widehat{\mathbb{C}}_p^{(x_1)}$ has one eigenvector more than $\mathbb{C}_p^{(x_1)}$ and therefore has a lower matrix rank.

Unfortunately, the balance (3.31a) might also increase the Poincaré rank at x_2 . Therefore, we have a closer look at (D.1d) evaluated at $k = q + 1$, where q is the Poincaré rank at the singularity x_2 :

$$\widetilde{\mathbb{C}}_{q+1}^{(x_2)} = (x_2 - x_1)\mathbb{Q}\mathbb{C}_q^{(x_2)}\overline{\mathbb{Q}}.$$

If this expression vanishes, the Poincaré rank at x_2 is not increased. This is indeed the case if the vectors w_k^\dagger in (3.41) span a left-invariant space of $\mathbb{C}_q^{(x_2)}$.

In Lee's algorithm, the above steps are used to decrease the Poincaré rank of a singularity x_1 . This can be done as long as it is possible to find a projector (3.41) with w_k^\dagger spanning a left-invariant space of $\mathbb{C}_q^{(x_2)}$ for some other singular point $x_2 \neq x_1$. If no such projector exists, a regular point y is chosen and the projector \mathbb{Q}_1 , defined by (3.42), is used in the balance. This of course creates a new (apparent) Fuchsian singularity at y . This way it is possible to reduce the system to an equivalent one with all singularities having Poincaré rank zero. Fortunately, the next step, the normalization of the eigenvalues, removes all apparent singularities introduced in this step.

Normalization

The second step of Lee's algorithm is the normalization of the eigenvalues of the matrix residues. A normalized eigenvalue is an eigenvalue proportional to ϵ . Again, the main ingredients are the balances defined in (3.31).

We assume all singularities to be Fuchsian now and the eigenvalues of the matrix residues should have the form $a + b\epsilon$ with $a \in \mathbb{Z}$. This is a necessary condition for the existence of a transformation normalizing all eigenvalues. If this condition cannot be fulfilled a redefinition of the kinematic variable might be helpful (cf. Ref. [65]).

The matrix residue at ∞ is equal to $-\sum_{x_j \in S} \mathbb{C}_0^{(x_j)}$. Therefore, the sum of all matrix residues (including the residue at ∞) vanishes and hence the sum of all eigenvalues of all matrix residues must vanish as well.

For brevity's sake, we will restrict our discussion to finite singularities $|x_{1,2}| < \infty$; the generalized treatment can again be found in Ref. [63].

Let $\{u_k^{(\alpha)}\}$ with $\alpha = 0, \dots, n_k$ be the set of $n_k + 1$ right generalized eigenvectors of $\mathbb{C}_0^{(x_1)}$, which belong to the Jordan block k :

$$\mathbb{C}_0^{(x_1)} u_k^{(0)} = \lambda_k u_k^{(0)}, \quad \mathbb{C}_0^{(x_1)} u_k^{(\alpha+1)} = \lambda_k u_k^{(\alpha+1)} + u_k^{(\alpha)}.$$

In analogy to the Fuchsification step, we define left generalized eigenvectors $v_k^{(\alpha)}$ via (3.39), which satisfy

$$v_k^{(0)\dagger} \mathbb{C}_0^{(x_1)} = \lambda_k v_k^{(0)\dagger}, \quad v_k^{(\alpha+1)\dagger} \mathbb{C}_0^{(x_1)} = \lambda_k v_k^{(\alpha+1)\dagger} + v_k^{(\alpha)\dagger}.$$

A balance (3.31a) with a right choice of a projector \mathbb{P} can now be used to shift one eigenvalue of the matrix residue at x_1 up by one and/or one eigenvalue of the matrix residue at x_2 down by one. Let us consider a projector \mathbb{P} defined by (3.34) for which we set

$$\mathbb{Q} = u_k^{(0)} w^\dagger, \tag{3.43}$$

where $u_k^{(0)}$ is the eigenvector to the eigenvalue we want to shift and w is some vector fulfilling $w^\dagger u_k^{(0)} = 1$. Also an additional projector

$$\mathbb{Q}_1 = u_k^{(0)} v_k^{(n_k)}$$

is useful for the discussion. The following relations involving these two projectors hold:

$$\mathbb{Q}\mathbb{Q}_1 = \mathbb{Q}_1, \quad \mathbb{Q}_1\mathbb{Q} = \mathbb{Q}, \quad \mathbb{C}_0^{(x_1)}\mathbb{Q} = \lambda_k\mathbb{Q}, \quad \mathbb{C}_0^{(x_1)}\mathbb{Q}_1 = \lambda_k\mathbb{Q}_1.$$

Let us now use (3.43) for the transformation of $\mathbb{C}_0^{(x_1)}$ under a balance (3.31a) as given in (D.1a), which yields

$$\begin{aligned} \tilde{\mathbb{C}}_0^{(x_1)} &= \mathbb{C}_0^{(x_1)} - \mathbb{Q}\mathbb{C}_0^{(x_1)}\bar{\mathbb{Q}} + \sum_{x_j \in S \setminus \{x_1\}} \frac{x_1 - x_2}{x_1 - x_j} \bar{\mathbb{Q}}\mathbb{C}_0^{(x_j)}\mathbb{Q} + \mathbb{Q} \\ &= (\bar{\mathbb{Q}} + \mathbb{Q}_1)\hat{\mathbb{C}}_0^{(x_1)}(\bar{\mathbb{Q}}_1 + \mathbb{Q}), \end{aligned}$$

where

$$\hat{\mathbb{C}}_0^{(x_1)} = \mathbb{C}_0^{(x_1)} - \mathbb{Q}_1\mathbb{C}_0^{(x_1)}\bar{\mathbb{Q}}_1 + \sum_{x_j \in S \setminus \{x_1\}} \frac{x_1 - x_2}{x_1 - x_j} \bar{\mathbb{Q}}_1\mathbb{C}_0^{(x_j)}\mathbb{Q}_1 + \mathbb{Q}_1. \quad (3.44)$$

Because of $(\bar{\mathbb{Q}} + \mathbb{Q}_1) = (\bar{\mathbb{Q}}_1 + \mathbb{Q})^{-1}$, $\tilde{\mathbb{C}}_0^{(x_1)}$ and $\hat{\mathbb{C}}_0^{(x_1)}$ are related by a similarity transformation and thus have the same eigenvalues. In order to evaluate the eigenvalues of $\tilde{\mathbb{C}}_0^{(x_1)}$ it is therefore sufficient to analyze the eigenvalues of $\hat{\mathbb{C}}_0^{(x_1)}$, which are much simpler to determine. We consider $\hat{\mathbb{C}}_0^{(x_1)}$ in the basis of the generalized eigenvectors of $\mathbb{C}_0^{(x_1)}$. In this basis, $\mathbb{C}_0^{(x_1)}$ is in Jordan normal form. The second term of (3.44) removes all elements from the row corresponding to $u_k^{(0)}$ except the diagonal one containing the eigenvalue to $u_k^{(0)}$. The last term of (3.44) ($+\mathbb{Q}_1$) increases this diagonal element by one. The sum in (3.44) contributes to the off-diagonal elements of the column corresponding to $u_k^{(0)}$ (or $v_k^{(n_k)\dagger}$). Hence, the transformations can be summarized as

$$\left(\begin{array}{ccccccc} \ddots & & & & & & \\ & \lambda_{k-1} & & & & & \\ & & \lambda_k & 1 & & & \\ & & & \lambda_k & 1 & & \\ & & & & & \ddots & \\ & & & & & & \lambda_k & 1 \\ & & & & & & & \lambda_k \\ & & & & & & & \lambda_{k+1} & 1 \\ & & & & & & & & \ddots \end{array} \right) \rightarrow \left(\begin{array}{ccccccc} \ddots & & & & & & \\ & \lambda_{k-1} & & * & & & \\ & & \lambda_k + 1 & 0 & & & \\ & & & * & \lambda_k & 1 & \\ & & & & & \ddots & \\ & & & & & & \lambda_k & 1 \\ & & & & & & & \lambda_k \\ & & & & & & & \lambda_{k+1} & 1 \\ & & & & & & & & \ddots \end{array} \right),$$

where $*$ stands for contributions from the $\bar{\mathbb{Q}}_1\mathbb{C}_0^{(x_j)}\mathbb{Q}_1$ terms. Calculating the characteristic polynomial by means of a Laplace expansion along the row corresponding to $u_k^{(0)}$ leads to the conclusion that all eigenvalues but one stay the same; only one eigenvalue λ_k is changed to $\lambda_k + 1$.

In the same way, a projector

$$\mathbb{Q} = wv_k^{(0)\dagger} \quad (3.45)$$

with a left eigenvector $v_k^{(0)\dagger}$ of $\mathbb{C}_0^{(x_2)}$ and some vector w which fulfills $v_k^{(0)\dagger}w = 1$ shifts one eigenvalue λ_k of $\mathbb{C}_0^{(x_2)}$ down by one.

A balance with a projector (3.43) could spoil the Fuchsian form of the system if it increases the Poincaré rank at any singularity. This is in principle possible at x_2 . Evaluating the corresponding transformation (D.1d) at $k = 1$ leads to

$$\tilde{\mathbb{C}}_1^{(x_2)} = (x_2 - x_1)\mathbb{Q}\mathbb{C}_0^{(x_2)}\overline{\mathbb{Q}}.$$

This vanishes if w^\dagger is a left eigenvector of $\mathbb{C}_0^{(x_2)}$. In that case, not only is the Fuchsian form of the system preserved, but we also arrive at a projector which is both of the form (3.43) and of the form (3.45). Thus, the projector of choice is

$$\mathbb{Q} = u_k^{(0)}v_l^{(0)\dagger}, \quad (3.46)$$

where $u_k^{(0)}$ is a right eigenvector of $\mathbb{C}_0^{(x_1)}$, $v_l^{(0)\dagger}$ is a left eigenvector of $\mathbb{C}_0^{(x_2)}$ and $v_l^{(0)\dagger}u_k^{(0)} = 1$. This projector, used in a balance (3.31a), increases one eigenvalue λ_k of $\mathbb{C}_0^{(x_1)}$ by one and decreases one eigenvalue μ_l of $\mathbb{C}_0^{(x_2)}$ by one while conserving the Fuchsian form of the system.

In order to utilize the considerations above in an algorithmic approach, one first selects a singularity x_0 as ‘fallback’. Next, the balances with projectors of the form (3.46) are used to ‘mutually balance’ eigenvalues between two singularities. Certainly, the eigenvalue to be increased should be negative and the eigenvalue to be decreased should be positive (for $\epsilon = 0$).

If no such balance exists, the eigenvalues will be balanced with the ‘fallback singularity’ x_0 regardless of the sign of the eigenvalue at x_0 . This normalizes the eigenvalues at all singularities except x_0 . In order to normalize even the eigenvalues at x_0 , we balance one unnormalized eigenvalue with some regular point creating a new apparent singularity there and restart the algorithm. Hopefully, the unnormalized eigenvalue at this new apparent singularity can now be mutually balanced with another unnormalized eigenvalue at x_0 . If this fails, this is often a sign that a canonical basis does not exist.

ϵ -Factorization

In the next step, an x -independent transformation $\mathbb{T}(\epsilon)$ has to be found to factor out ϵ . For an x -independent transformation matrix \mathbb{T} , (3.25) becomes a similarity transformation and does not change the eigenvalues of the system. This makes it necessary for the eigenvalues to be proportional to ϵ , i.e. the normalization step must have been carried out successfully. We use the fact that $\mathbb{T}^{-1}(\epsilon)[\mathbb{M}_0^{(x_j)}(\epsilon)/\epsilon]\mathbb{T}(\epsilon)$ should be independent of ϵ , so that the equation

$$\mathbb{T}^{-1}(\epsilon)\frac{\mathbb{M}_0^{(x_j)}(\epsilon)}{\epsilon}\mathbb{T}(\epsilon) = \mathbb{T}^{-1}(\mu)\frac{\mathbb{M}_0^{(x_j)}(\mu)}{\mu}\mathbb{T}(\mu)$$

holds for all $x_j \in S$. Multiplying this equation from the left by $\mathbb{T}(\epsilon)$ and from the right by $\mathbb{T}^{-1}(\mu)$, leads to

$$\frac{\mathbb{M}_0^{(x_j)}(\epsilon)}{\epsilon}\mathbb{T}(\epsilon, \mu) = \mathbb{T}(\epsilon, \mu)\frac{\mathbb{M}_0^{(x_j)}(\mu)}{\mu}, \quad (3.47)$$

where $\mathbb{T}(\epsilon, \mu) = \mathbb{T}(\epsilon)\mathbb{T}^{-1}(\mu)$. This linear system can be solved e.g. with Gaußian elimination and the constants should be fixed so that $\mathbb{T}(\epsilon, \mu_0)$ is an invertible matrix, where μ_0 is some arbitrary number. The transformation $\mathbb{T}(\epsilon, \mu_0)$ can now be used to factor out ϵ .

Fuchsification for off-diagonal blocks

Since the definition of the active block is somewhat arbitrary, we should in principle be able to transform all diagonal blocks to ϵ -form by redefining the active block and applying the three steps

described above. However, we still need to reduce the off-diagonal block \mathbb{B} to Fuchsian form. Again we will restrict the discussion to finite singularities $|x_1| < \infty$ and assume the block \mathbb{A} and \mathbb{C} to be already in ϵ -form.

Let p be the Poincaré rank of the off-diagonal block \mathbb{B} at the singularity x_1 , i.e. $\mathbb{B}_p^{(x_1)} \neq 0$ and $\mathbb{B}_k^{(x_1)} = 0$ for $k > p$. The behavior of $\mathbb{B}_p^{(x_1)}$ under a transformation (3.32a) with $k = p$ is given by (D.3a):

$$\widetilde{\mathbb{B}}_p^{(x_1)} = \mathbb{B}_p^{(x_1)} + \mathbb{C}_0^{(x_1)} \widehat{\mathbb{G}} - \widehat{\mathbb{G}} \mathbb{A}_0^{(x_1)} + p \widehat{\mathbb{G}}.$$

In order to decrease the Poincaré rank of \mathbb{B} at x_1 , $\widehat{\mathbb{G}}$ has to be determined such that $\widetilde{\mathbb{B}}_p^{(x_1)}$ vanishes. This linear system of equations can be solved with Gaußian elimination.

Hence, with transformations of the form (3.32) it is possible to reduce all singularities of the off-diagonal block \mathbb{B} to Fuchsian form.

3.3.4 Solving Canonical Master Integrals

Solving master integrals becomes trivial if they are in a canonical basis. We start our discussion with a more general form of a canonical basis

$$\frac{\partial \vec{f}(x, \epsilon)}{\partial x} = \epsilon \widehat{\mathbb{M}}(x) \vec{f}(x, \epsilon). \quad (3.48)$$

We impose as a condition only that the right-hand side is already proportional to ϵ , whereas the x dependence of $\widehat{\mathbb{M}}(x)$ is not restricted to the form (3.28). A suitable ansatz to solve (3.48) is

$$\vec{f}(x, \epsilon) = U(x, x_0; \epsilon) \vec{f}(x_0, \epsilon), \quad (3.49)$$

with a Dyson-like evolution operator fulfilling

$$U(x_0, x_0; \epsilon) = \mathbb{1}. \quad (3.50)$$

The operator $U(x, x_0; \epsilon)$ evolves the set of master integrals at a certain kinematic point x_0 to another point x . Plugging (3.49) into (3.48) yields a differential equation for the evolution operator

$$\frac{\partial U(x, x_0; \epsilon)}{\partial x} = \epsilon \widehat{\mathbb{M}}(x) U(x, x_0; \epsilon). \quad (3.51)$$

The operator U can now be expanded as a power series in ϵ

$$U(x, x_0; \epsilon) = \sum_{n=0}^{\infty} \epsilon^n U_n(x, x_0). \quad (3.52)$$

Inserting (3.52) into (3.51) leads to

$$\frac{\partial U_0(x, x_0)}{\partial x} = 0, \quad (3.53a)$$

$$\frac{\partial U_n(x, x_0)}{\partial x} = \widehat{\mathbb{M}}(x) U_{n-1}(x, x_0) \quad ; \quad n > 0, \quad (3.53b)$$

where the conditions

$$U_0(x, x_0) = \mathbb{1}, \quad U_n(x_0, x_0) = 0 \quad ; \quad n > 0, \quad (3.54)$$

are required so that (3.50) and (3.53a) are satisfied. Integrating (3.53b) gives

$$U_n(x, x_0) = \int_{x_0}^x dx' \widehat{\mathbb{M}}(x') U_{n-1}(x', x_0) \quad ; \quad n > 0, \quad (3.55)$$

where the limits of integration are chosen to fulfill (3.54). Equation (3.55) can be used to solve (3.51) iteratively up to the desired order in ϵ in terms of iterated integrals.

In the framework of Lee's algorithm we obtained a canonical basis of the form (3.29). This coincides with a special case of (3.48) when one chooses $\widehat{\mathbb{M}}(x)$ in (3.55) as

$$\widehat{\mathbb{M}}(x) = \sum_{x_j \in S} \frac{\widehat{\mathbb{M}}_0^{(x_j)}}{x - x_j},$$

which yields

$$U_n(x, x_0) = \sum_{x_j \in S} \widehat{\mathbb{M}}_0^{(x_j)} \int_{x_0}^x \frac{dx'}{x' - x_j} U_{n-1}(x', x_0). \quad (3.56)$$

The integrals in this expression can be identified with the GPLs defined in (3.1). The set of singularities S is also the set of possible indices in the GPLs required to express the solutions.

A solution of the master integrals $\vec{f}(x, \epsilon)$ also requires the boundary conditions $\vec{f}(x_0, \epsilon)$ at some suitable kinematic point x_0 , where an asymptotic expansion can be easily obtained. There are various methods to compute asymptotic expansions of Feynman integrals. In Section 3.4, we give a brief overview of the strategy used in this thesis to calculate the boundary conditions. In general neither $U(x, x_0; \epsilon)$ nor $\vec{f}(x_0, \epsilon)$ will be well defined for a specific point x_0 . Hence, equation (3.49) should be understood as

$$\vec{f}(x, \epsilon) = \lim_{x' \rightarrow x_0} U(x, x'; \epsilon) \vec{f}(x', \epsilon).$$

A further discussion of this problem can be found in Section 4.1.

3.4 Expansion by Subgraphs

The boundary conditions for the differential equation technique described in Section 3.2 and Section 3.3 require an expansion of all master integrals in a suitable limit. To perform an asymptotic expansion of Feynman integrals, various strategies are available. One strategy is the expansion by regions [57, 106–108]. As another possibility, an asymptotic expansion may be obtained using Mellin-Barnes techniques as demonstrated in Section 3.6.2. A third option is a more diagrammatic approach, i.e. the expansion by subgraphs [54–57]. While the first two strategies are more general, the expansion by subgraphs is often simpler. Unfortunately, the expansion by subgraphs is only valid in the Euclidean limits.

In this section, we will focus on a hard mass expansion via an expansion by subgraphs while we restrict ourselves to a description of the method without providing any proofs. For a rigorous discussion of asymptotic expansion methods in both momenta and masses, see the excellent textbook by Smirnov, Ref. [57].

In the framework of an expansion by subgraphs, a Feynman integral I_Γ with graph Γ can be expressed as power series in $1/m$ for a large mass m by (see e.g. Ref. [56])

$$I_\Gamma(\underline{q}, m, \underline{\mu}) \stackrel{m \rightarrow \infty}{\approx} \sum_{\gamma} I_{\Gamma \setminus \gamma}(\underline{q}, \underline{\mu}) \circ \mathcal{T}_{\underline{q}^\gamma, \underline{\mu}} I_\gamma(\underline{q}^\gamma, m, \underline{\mu}), \quad (3.57)$$

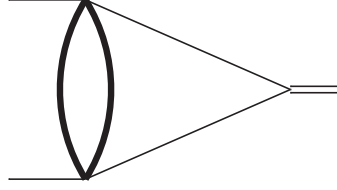


Fig. 3.1: Two loop triangle diagram.

where \underline{q} denotes the set of all external momenta and $\underline{\mu}$ the set of all small masses $\mu_i \ll m$. γ is a subgraph of Γ and \underline{q}^γ is the set of all momenta external to this subgraph. The co-graph $\Gamma \setminus \gamma$ is obtained from Γ by contracting the subgraph γ to a point. $\mathcal{T}_{\underline{q}^\gamma, \underline{\mu}}$ denotes a Taylor expansion in \underline{q}^γ and $\underline{\mu}$ to the desired order. This Taylor expansion is then reinserted into $I_{\Gamma \setminus \gamma}$ at the point where the subgraph γ was once located. The sum runs over all subgraphs γ of Γ , which contain all lines with the large mass m and are also one-particle irreducible with respect to lines carrying the small masses $\underline{\mu}$ or are massless.

Since all lines with mass m are included into γ , the integral $I_{\Gamma \setminus \gamma}$ is independent of this scale. The Taylor coefficients of I_γ do not depend on external momenta anymore and are therefore only tadpole integrals. Hence, both contributions are Feynman integrals with less scales than the original integral I_Γ .

As an example, we consider the two loop integral depicted in fig. 3.1 in the limit of a large internal mass compared to the mass of the outgoing particle. We choose a momentum flow as in

$$I = \iint_{l, k} \frac{1}{(l + q_1)^2 (l - q_2)^2 [k^2 - 1] [(k - l)^2 - 1]},$$

where we set the internal mass to one as usual.

The variable $\tau = q_1 \cdot q_2 / 2$ is considered small with respect to the internal mass ($\tau \ll 1$). Formula (3.57) can be written in diagrammatic notation as

$$\text{Diagram} \approx \text{Diagram} \circ \mathcal{T}_\tau \text{Diagram} + \text{Diagram} \circ \mathcal{T}_l \text{Diagram}. \quad (3.58)$$

The first term of (3.58) is particularly simple since it involves a Taylor expansion of the original diagram in the external momenta and a co-graph equal to one. The Taylor expansion is given by [62]

$$\begin{aligned} \mathcal{T}_\tau \iint_{l, k} \frac{1}{(l + q_1)^2 (l - q_2)^2 [k^2 - 1] [(k - l)^2 - 1]} &= \iint_{l, k} \frac{1}{[l^2]^2 [k^2 - 1] [(k - l)^2 - 1]} \\ &+ \frac{2\tau}{d} \frac{\partial^2}{\partial q_{1\mu} \partial q_2^\mu} \iint_{l, k} \frac{1}{(l + q_1)^2 (l - q_2)^2 [k^2 - 1] [(k - l)^2 - 1]} \Bigg|_{q_1=q_2=0} + \mathcal{O}(\tau^2), \end{aligned}$$

which yields in diagrammatic notation

$$\mathcal{T}_\tau \text{Diagram} = \text{Diagram} - \frac{8\tau}{d} \text{Diagram} + \mathcal{O}(\tau^2).$$

For the second term of (3.58), we need the Taylor expansion of the subgraph [62]

$$\begin{aligned} \tau_l \int_k \frac{1}{[k^2 - 1][(k - l)^2 - 1]} &= \int_k \frac{1}{[k^2 - 1]^2} \\ &+ \frac{l^2}{2d} \frac{\partial^2}{\partial l_\mu \partial l^\mu} \int_k \frac{1}{[k^2 - 1][(k - l)^2 - 1]} \Bigg|_{l=0} + \mathcal{O}(l^4), \end{aligned}$$

where the loop momentum l of the co-graph is treated as an external momentum. In diagrammatic notation, we obtain

$$\tau_l \left(\text{tadpole} \right) = \left(\text{circle} \right) + l^2 \left[\frac{4-d}{d} \left(\text{circle with 2 dots} \right) + \frac{4}{d} \left(\text{circle with 4 dots} \right) \right] + \mathcal{O}(l^4).$$

The next-to-leading order term leads to an additional numerator l^2 in the co-graph. Shifting the loop momentum yields

$$\int_l \frac{l^2}{(l + q_1)^2 (l - q_2)^2} = \int_l \frac{l^2 + 2l \cdot q_2}{(l + q_1 + q_2)^2 l^2} = 2q_{2\mu} \int_l \frac{l^\mu}{(l + q_1 + q_2)^2 l^2},$$

where we dropped a scale-less tadpole integral. The form factor proportional to $(q_1 + q_2)^\mu$ of the tensor integral can now be projected out, which leads to

$$\int_l \frac{l^2}{(l + q_1)^2 (l - q_2)^2} = -2\tau \int_l \frac{1}{(l + q_1 + q_2)^2 l^2}.$$

Using these ingredients, the asymptotic expansion can be composed of one-scale diagrams via

$$\begin{aligned} I &= I_1 + I_2, \\ I_1 &= \left(\text{circle with 1 dot} \right) - \frac{8\tau}{d} \left(\text{circle with 2 dots} \right) + \mathcal{O}(\tau^2), \\ I_2 &= \left\{ \left(\text{circle with 2 dots} \right) - 2\tau \left[\frac{4-d}{d} \left(\text{circle with 2 dots} \right) + \frac{4}{d} \left(\text{circle with 4 dots} \right) \right] \right\} \cdot \left(\text{tadpole} \right) + \mathcal{O}(\tau^2). \end{aligned}$$

The one-scale diagrams can be solved in terms of gamma functions, given by

$$\begin{aligned} \left(\text{circle with } n-1 \text{ dots} \right) &= \frac{(-1)^n (\epsilon - 1)^2 \Gamma(2 - \epsilon - n) \Gamma^2(-1 + \epsilon + n) \Gamma(-2 + 2\epsilon + n)}{\Gamma(2 - \epsilon) \Gamma^2(1 + \epsilon) \Gamma(-2 + 2\epsilon + 2n)}, \\ \left(\text{circle with } n \text{ dots} \right) &= \frac{(-1)^n (1 - \epsilon) \Gamma(-2 + \epsilon + n)}{(n - 1)! \Gamma(1 + \epsilon)}, \\ \left(\text{tadpole} \right) &= -\frac{(-4\tau)^{-\epsilon} \Gamma^2(2 - \epsilon) \Gamma(\epsilon - 1)}{\Gamma(2 - 2\epsilon) \Gamma(1 + \epsilon)}. \end{aligned}$$

This yields

$$\begin{aligned}
 I &= I_1 + I_2, \\
 I_1 &= \frac{\epsilon - 1}{2\epsilon^2(1 + 2\epsilon)} + (-4\tau) \frac{\epsilon - 1}{2\epsilon(6 + 13\epsilon - 4\epsilon^3)} + \mathcal{O}(\tau^2), \\
 I_2 &= (-4\tau)^{-\epsilon} \frac{\Gamma^2(2 - \epsilon)}{\epsilon^2 \Gamma(2 - 2\epsilon)} + (-4\tau)^{1-\epsilon} \frac{\Gamma^2(2 - \epsilon)}{12\epsilon \Gamma(2 - 2\epsilon)} + \mathcal{O}(\tau^2).
 \end{aligned} \tag{3.59}$$

We will later, in Section 3.6.2, consider the same example with a completely different expansion technique based on Mellin-Barnes representations.

The strategy *expansion by subgraphs* is implemented in the C++-tool `exp` [50, 51] accompanied by a variety of FORM-codes to perform the Taylor expansions.

3.5 Feynman Integral Parametrizations

In this section, we discuss two important parametrizations of Feynman integrals, Schwinger parametrization and Feynman parametrization. Both representations are often used as starting points for analytical and numerical evaluations.

We define a Minkowskian scalar K -loop Feynman integral by⁵

$$I^{(d)}(a_1, \dots, a_N) = \int \frac{d^d l_1}{i\pi^{d/2}} \cdots \int \frac{d^d l_K}{i\pi^{d/2}} \frac{1}{D_1^{a_1} \cdots D_N^{a_N}}, \tag{3.60}$$

where we restrict the discussion for the moment to integrals without numerators, i.e. $a_i > 0$. The denominators D_i are usually of the form

$$D_i = p_i^2 - m_i^2 + i0, \tag{3.61}$$

where p_i is a linear combination of external and loop momenta.

For Minkowskian integrals like the one in (3.60), one can introduce Schwinger parameters using the well-known formula

$$\frac{1}{D_i^{a_i}} = \frac{(-i)^{a_i}}{\Gamma(a_i)} \int_0^\infty dx_i x_i^{a_i-1} e^{ix_i D_i}, \tag{3.62}$$

which is valid for $\text{Im}(D_i) > 0$. After using (3.62) for all D_i in (3.60), the momentum integrals can be performed loop-by-loop via

$$\int \frac{d^d l}{i\pi^{d/2}} e^{i(\alpha l^2 + 2 q \cdot l)} = (-i)^{d/2} \alpha^{-d/2} e^{-iq^2/\alpha}. \tag{3.63}$$

Together, this leads to the Schwinger parametrization of the Feynman integral (3.60)

$$I^{(d)}(a_1, \dots, a_N) = \frac{(-i)^{A+Kd/2}}{\Gamma(a_1) \cdots \Gamma(a_N)} \int_0^\infty dx_1 x_1^{a_1-1} \cdots \int_0^\infty dx_N x_N^{a_N-1} \frac{e^{-iF/U - i \sum_i x_i m_i^2}}{U^{d/2}}, \tag{3.64}$$

where $A = a_1 + \cdots + a_N$. The Symanzik polynomials U and F depend on the Schwinger parameters x_i and can be read off directly from the Feynman graph. Nevertheless, in this thesis, we usually

⁵Note that we are using a different integral measure as given in Appendix A.

compute them with the formulas given in this section. To get an overview of the properties of these graph polynomials, see Ref. [109].

An Euclidean Feynman integral is defined by

$$\tilde{I}^{(d)}(a_1, \dots, a_N) = \int \frac{d^d L_1}{\pi^{d/2}} \cdots \int \frac{d^d L_K}{\pi^{d/2}} \frac{1}{\tilde{D}_1^{a_1} \cdots \tilde{D}_N^{a_N}}, \quad (3.65)$$

where

$$\tilde{D}_i = P_i^2 + m_i^2.$$

Here, it is advisable to use a different type of Schwinger parameters, namely

$$\frac{1}{\tilde{D}_i^{a_i}} = \frac{1}{\Gamma(a_i)} \int_0^\infty dx_i x_i^{a_i-1} e^{-x_i \tilde{D}_i}, \quad (3.66)$$

which is valid for $\tilde{D}_i > 0$. The Gaussian loop-integrations are then performed via

$$\int \frac{d^d L}{\pi^{d/2}} e^{-\alpha L^2 + 2Q \cdot L} = \alpha^{-d/2} e^{Q^2/\alpha}, \quad (3.67)$$

and the final Schwinger parametrized Euclidean Feynman integral is given by

$$\tilde{I}^{(d)}(a_1, \dots, a_N) = \frac{1}{\Gamma(a_1) \cdots \Gamma(a_N)} \int_0^\infty dx_1 x_1^{a_1-1} \cdots \int_0^\infty dx_N x_N^{a_N-1} \frac{e^{-F/U - \sum_i x_i m_i^2}}{U^{d/2}}. \quad (3.68)$$

Note that the form (3.68) can also be useful for Minkowskian Feynman integrals since one can usually Wick-rotate first [110].

The Feynman parametrization is closely related to the Schwinger parametrization. We start with the Schwinger-parametrized Feynman integral (3.64) and make a change of variables to

$$\begin{aligned} x'_i &= \frac{x_i}{\sum_j x_j} \quad ; \quad i = 1, \dots, N-1, \\ \eta &= \sum_j x_j. \end{aligned}$$

The Schwinger-parameters have to be replaced by $x_i = \eta x'_i$, where we define $x'_N = 1 - \sum_{i=1}^{N-1} x'_i$. The region of integration is bounded by $0 < \sum_{i=1}^{N-1} x'_i < 1$, $0 < \eta < \infty$ and the Jacobian determinant equals η^{N-1} . This leads to

$$\begin{aligned} I^{(d)}(a_1, \dots, a_N) &= \frac{(-i)^{A+Kd/2}}{\Gamma(a_1) \cdots \Gamma(a_N)} \int dx'_1 \cdots dx'_{N-1} x_1^{a_1-1} \cdots x_N^{a_N-1} \\ &\quad \int_0^\infty d\eta \eta^{A-Kd/2-1} \frac{e^{-i\eta(F'/U' + \sum_i x'_i m_i^2)}}{U'^{d/2}}. \end{aligned}$$

We used the property of the Symanzik polynomials to be homogeneous in the Schwinger parameters x_i with degrees $\deg(U) = K$ and $\deg(F) = K+1$. The functions U' and F' denote the polynomials U and F , where the parameters x_i are replaced by x'_i . The η -integral can be solved via

$$\int_0^\infty d\eta \eta^{\alpha-1} e^{-i\eta A} = \Gamma(\alpha) (iA)^{-\alpha},$$

leading to a Feynman-parameter integral

$$\begin{aligned}
 I^{(d)}(a_1, \dots, a_N) &= (-1)^A \frac{\Gamma(A - Kd/2)}{\Gamma(a_1) \cdots \Gamma(a_N)} \\
 &\int_0^\infty dx_1 x_1^{a_1-1} \cdots \int_0^\infty dx_N x_N^{a_N-1} \delta\left(1 - \sum_{i=1}^N x_i\right) \frac{U^{A-(K+1)d/2}}{(F + U \sum_i x_i m_i^2)^{A-Kd/2}},
 \end{aligned} \tag{3.69}$$

where we dropped the primes and introduced an additional integral over x_N compensated by a delta function. The delta functions allowed us to relax the limits on the x_i integrals to the interval $0 < x_i < \infty$.

3.5.1 Dealing with Numerators

Until now, we assumed that all indices a_i of (3.60) are positive. For numerators represented by negative integers a_i , (3.62) cannot be used to introduce Schwinger parameters. Instead, one can make use of

$$D_i^{|a_i|} = \left(-i \frac{\partial}{\partial x_i}\right)^{|a_i|} e^{ix_i D_i} \Big|_{x_i=0}.$$

The parameter x_i can then be treated like an ordinary Schwinger-parameter in the calculation of the Symanzik-polynomials U and F . In (3.64), only the replacement

$$\frac{1}{\Gamma(a_i)} \int_0^\infty dx_i x_i^{a_i-1} \rightarrow \left(-\frac{\partial}{\partial x_i}\right)^{|a_i|} \Big|_{x_i=0} \tag{3.70}$$

has to be made for the negative indices.

However, this substitution is not always helpful. Especially, when the Schwinger parametrization is used as starting point for the construction of a Mellin-Barnes representation (cf. Section 3.7), the Schwinger integral is essential and cannot be replaced by a derivative.

In order to deal with this problem, one can treat the indices a_i corresponding to the numerators as positive in the computation of the Schwinger- or Feynman-parametrization and only in the result (i.e. in the Mellin-Barnes representation) perform an analytic continuation to negative indices. Unfortunately, this approach suffers from some subtleties that limit its practicality.

The method described in Section 3.7 is based on the Euclidean Schwinger parametrization (3.68). Therefore, bilinear forms as numerators are not supported as they are not strictly positive after a Wick rotation. Furthermore, the Mellin-Barnes representations obtained for Feynman integrals with artificially introduced additional propagators tend to have a higher dimensionality than the representations without these propagators. In order to perform the final analytic continuation, we also need to introduce additional analytic regulators. This, in turn, will complicate all subsequent evaluations of the Feynman Integral.

A different approach is based on dimensional shifts of the Feynman integral (see e.g. Ref. [85]). For this, we consider (3.64) and apply (3.70) to some of the indices:

$$\begin{aligned}
 I^{(d)}(a_1, \dots, a_N; c_1, \dots, c_J) &= \frac{(-i)^{A-C+Kd/2}}{\Gamma(a_1) \cdots \Gamma(a_N)} \int_0^\infty dx_1 x_1^{a_1-1} \cdots \int_0^\infty dx_N x_N^{a_N-1} \\
 &\quad \left(\frac{\partial}{\partial x_{N+1}} \right)^{-c_1} \cdots \left(\frac{\partial}{\partial x_{N+J}} \right)^{-c_J} \frac{e^{-iF/U - i \sum_i x_i m_i^2}}{U^{d/2}} \Bigg|_{x_{N+1}=\dots=x_{N+J}=0},
 \end{aligned} \tag{3.71}$$

where $c_i < 0$ correspond to the numerators and $C = c_1 + \dots + c_J$.

First, let us examine the case with only a single derivative of order one (i.e. $J = 1$, $c_1 = -1$), for which we have

$$\frac{\partial}{\partial x_{N+1}} \frac{e^{-iF/U - i \sum_i x_i m_i^2}}{U^{d/2}} \Bigg|_{x_{N+1}=0} = \left[-\frac{d}{2} \frac{1}{U} - i \frac{F'}{U} + i \frac{FU'}{U^2} - im_{N+1}^2 \right] \frac{e^{-iF/U - i \sum_i x_i m_i^2}}{U^{d/2}} \Bigg|_{x_{N+1}=0},$$

where $F' = \partial F / \partial x_{N+1}$ and $U' = \partial U / \partial x_{N+1}$. The first thing to note is that the replacement rule $x_{N+1} = 0$ turns U and F to the Symanzik polynomials for the graph without the artificially introduced propagator. Furthermore, in the square bracket on the right-hand side the denominator contains only powers of U , whereas the numerator consists of polynomials in the Schwinger parameters. These statements remain true even in the more general case of more and/or higher order numerators. Thus, a single term in the square bracket always has the form

$$\xi \frac{x_1^{s_1} \cdots x_N^{s_N}}{U^r},$$

where ξ is a prefactor independent of the Schwinger parameters x_i . Such a term leads in (3.71) to

$$\xi \frac{(-i)^{A-C+Kd/2}}{\Gamma(a_1) \cdots \Gamma(a_N)} \int_0^\infty dx_1 x_1^{a_1+s_1-1} \cdots \int_0^\infty dx_N x_N^{a_N+s_N-1} \frac{e^{-iF/U - i \sum_i x_i m_i^2}}{U^{(d+2r)/2}} \Bigg|_{x_{N+1}=\dots=x_{N+J}=0}. \tag{3.72}$$

This integral is proportional to $I^{(d+2r)}(a_1 + s_1, \dots, a_N + s_N; 0, \dots, 0)$. Therefore, it belongs to the same sector as the original integral but without numerators and an increased number of space-time dimensions. Comparing (3.72) to (3.71) helps to determine the proportionality factor and to deduce the replacement rule

$$\frac{x_1^{s_1} \cdots x_N^{s_N}}{U^r} \rightarrow i^{S+C+Kdr} (a_1)_{s_1} \cdots (a_N)_{s_N} I^{(d+2r)}(a_1 + s_1, \dots, a_N + s_N; 0, \dots, 0),$$

where $S = s_1 + \dots + s_N$, and $(a)_s = \Gamma(a + s) / \Gamma(a)$ denotes a Pochhammer symbol.

So we can conclude that every scalar Feynman integral with numerators can be expressed as linear combination of Feynman integrals of the same sector without numerators but with shifted indices and shifted space-time dimensions. The drawback of this method is that instead of only one, a sometimes large number of Feynman integrals has to be evaluated.

3.6 Evaluation by Mellin-Barnes Representations

The evaluation by Mellin-Barnes (MB) representations [89, 90] is one of the most successful techniques to compute Feynman integrals. A MB representation is a (multidimensional) contour integral along

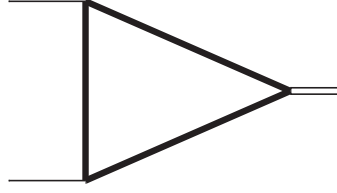


Fig. 3.2: Massive one-loop triangle diagram.

the imaginary axis involving products of gamma functions. This representation can be constructed with various methods, which are based on the Cahen-Mellin integral

$$e^{-A} = \int_{c-i\infty}^{c+i\infty} \frac{dz}{2\pi i} A^{-z} \Gamma(z) \quad ; \quad c > 0, \operatorname{Re}(A) > 0, \quad (3.73)$$

or (generalizations of) the corollary⁶

$$\frac{1}{(A+B)^a} = \frac{1}{\Gamma(a)} \int_{c-i\infty}^{c+i\infty} \frac{dz}{2\pi i} \frac{B^z}{A^{a+z}} \Gamma(a+z) \Gamma(-z) \quad ; \quad -a \notin \mathbb{N}_0, \quad (3.74)$$

where the contour in (3.74) is chosen such that the poles of $\Gamma(a+z)$ are to the left, and the poles of $\Gamma(-z)$ are to the right of the contour. The simplest way to make use of (3.74) is to apply it directly to the massive propagators of a Feynman integral

$$\frac{1}{(p^2 - m^2)^a} = \frac{1}{\Gamma(a)} \int_{c-i\infty}^{c+i\infty} \frac{dz}{2\pi i} \frac{(-m^2)^z}{(p^2)^{a+z}} \Gamma(a+z) \Gamma(-z)$$

to transform them effectively to massless propagators until the Feynman integral is solvable in terms of gamma functions.

This method is applicable mainly for one-loop integrals. As an example, we consider the integral in fig. 3.2, given by

$$I(a_1, a_2, a_3) = \int \frac{d^d l}{i\pi^{d/2}} \frac{1}{[l^2 - m^2]^{a_1} [(l+q_1)^2 - m^2]^{a_2} [(l-q_2)^2 - m^2]^{a_3}},$$

where $q_1^2 = q_2^2 = 0$ and $q_1 \cdot q_2 = 2\tau$. Using the above method, this can be expressed as

$$\begin{aligned} I(a_1, a_2, a_3) &= \int_{c_1-i\infty}^{c_1+i\infty} \frac{dz_1}{2\pi i} \int_{c_2-i\infty}^{c_2+i\infty} \frac{dz_2}{2\pi i} \int_{c_3-i\infty}^{c_3+i\infty} \frac{dz_3}{2\pi i} (-m^2)^{z_{123}} \\ &\quad \cdot \frac{\Gamma(-z_1) \Gamma(-z_2) \Gamma(-z_3) \Gamma(a_1+z_1) \Gamma(a_2+z_2) \Gamma(a_3+z_3)}{\Gamma(a_1) \Gamma(a_2) \Gamma(a_3)} \\ &\quad \cdot \int \frac{d^d l}{i\pi^{d/2}} \frac{1}{[l^2]^{z_1+a_1} [(l+q_1)^2]^{z_2+a_2} [(l-q_2)^2]^{z_3+a_3}}. \end{aligned}$$

The remaining massless Feynman integral can now be solved in terms of gamma functions by

$$\begin{aligned} &\int \frac{d^d l}{i\pi^{d/2}} \frac{1}{[l^2]^{c_1} [(l+q_1)^2]^{c_2} [(l-q_2)^2]^{c_3}} \\ &= e^{-i\pi d/2} (4\tau)^{-c_{123}+d/2} \frac{\Gamma(c_{123}-d/2) \Gamma(-c_{12}+d/2) \Gamma(-c_{13}+d/2)}{\Gamma(c_2) \Gamma(c_3) \Gamma(-c_{123}+d)}. \end{aligned}$$

⁶(3.74) can be derived by applying a Schwinger parametrization (3.66) to the left-hand side and then using (3.73) for e^{-xA} and e^{-xB} . One contour integral can subsequently be solved by means of the Mellin inversion theorem.

This leads to a MB representation

$$\begin{aligned}
 I(a_1, a_2, a_3) = & e^{-i\pi d/2} (4\tau)^{-a_{123}+d/2} \int_{c_1-i\infty}^{c_1+i\infty} \frac{dz_1}{2\pi i} \int_{c_2-i\infty}^{c_2+i\infty} \frac{dz_2}{2\pi i} \int_{c_3-i\infty}^{c_3+i\infty} \frac{dz_3}{2\pi i} \left(-\frac{m^2}{4\tau}\right)^{z_{123}} \\
 & \cdot \frac{\Gamma(-z_1)\Gamma(-z_2)\Gamma(-z_3)\Gamma(a_1+z_1)\Gamma(a_{123}+z_{123}-d/2)}{\Gamma(a_1)\Gamma(a_2)\Gamma(a_3)} \\
 & \cdot \frac{\Gamma(-a_{12}-z_{12}+d/2)\Gamma(-a_{13}-z_{13}+d/2)}{\Gamma(-a_{123}-z_{123}+d)}.
 \end{aligned}$$

The contours of a MB representation are always chosen such that the poles of the gamma functions $\Gamma(\dots + z_i)$ are to the left, and the poles of the gamma functions $\Gamma(\dots - z_i)$ are to the right of the z_i -contour in the complex plane. Note that these contours do not necessarily need to be straight lines parallel to the imaginary axis.

For multiloop integrals or more complicated one-loop integrals, more advanced techniques are necessary for the construction of MB representations. Two approaches are implemented in the Mathematica package `AMBRE` [111–114]. A novel method, which is particularly powerful for non-planar integrals with not too many masses, was developed as part of this thesis. This technique is inspired by the Method of Brackets [115–117] and published in Ref. [2]. See Section 3.7 for details.

The focus of this section lies on the individual steps required to evaluate a Feynman integral after a suitable MB representation has been found. In Section 3.6.1, the resolution of singularities when approaching the limit of $d = 4$ space-time dimensions is discussed. In Section 3.6.2, we describe a method for an asymptotic expansion based on MB representations and the residue theorem.

3.6.1 Resolving Singularities

In general, Feynman integrals are divergent in four space-time dimensions. In the context of dimensional regularization, they are therefore calculated in $d = 4 - 2\epsilon$ dimensions. In many cases results with an exact d (or ϵ) dependence cannot be obtained, which is why one has to look for methods to directly obtain Laurent expansions in ϵ .

MB representations allow carrying out the Laurent expansion before the contour integrals are performed. The individual terms in the Laurent expansion are then expressed as sums of MB integrals independent of ϵ .

There are two main strategies to resolve singularities of a MB representation. In the literature these two methods are usually referred to as Strategy A [89] and Strategy B [90]. The latter is implemented in the public Mathematica-package `MB` [118]. Throughout this thesis we use a modified and more algorithmic version of Strategy A, implemented in the Mathematica-package `MBresolve` [119]. For practical reasons, we also reimplemented this method in a private C++-code.

At first, we discuss the origin of ϵ -poles in a MB representation. As a very basic example for the use of the modified Strategy A, we consider the positions of the poles of

$$\Gamma(-2 - \epsilon + z)\Gamma(1 - z)$$

in the complex z -plane, shown in fig. 3.3. The solid line \mathcal{C}_1 in fig. 3.3 depicts a possible contour separating the poles of $\Gamma(-2 - \epsilon + z)$ from the poles of $\Gamma(1 - z)$. Note that \mathcal{C}_1 is pinched between some poles coming from the two gamma functions which are separated only by ϵ . For $\epsilon = 0$, such a contour would be impossible. This pinching of contours is the origin of ϵ -poles in the result.

Now, we consider the integral

$$I = \int_{\mathcal{C}_1} \frac{dz}{2\pi i} \Gamma(-2 - \epsilon + z)\Gamma(1 - z)f(z).$$

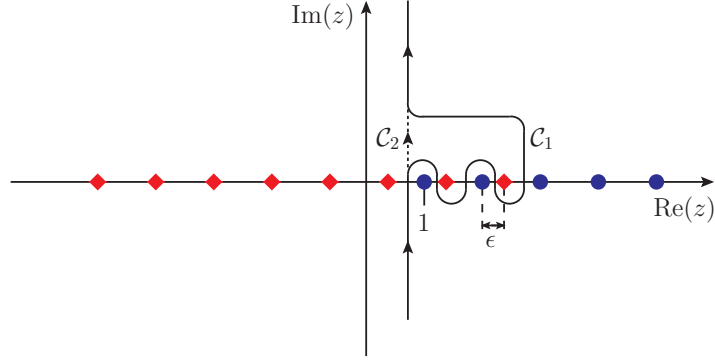


Fig. 3.3: Possible contour C_1 (solid line) in the complex z -plane separating the poles of $\Gamma(-2 + \epsilon + z)$ (red diamonds) from the poles of $\Gamma(1 - z)$ (blue circles). The dotted line shows contour C_2 .

This integral can be expressed via an integral over the contour C_2 , depicted as dotted line in fig. 3.3, by

$$I = \int_{C_2} \frac{dz}{2\pi i} \Gamma(-2 - \epsilon + z) \Gamma(1 - z) f(z) + \oint_{C_1 - C_2} \frac{dz}{2\pi i} \Gamma(-2 - \epsilon + z) \Gamma(1 - z) f(z).$$

The integral over the closed contour $C_1 - C_2$ circles the poles at $z = 1 + \epsilon$ and $z = 2 + \epsilon$ counter-clockwise and can be treated by the residue theorem⁷,

$$\begin{aligned} I &= \int_{C_2} \frac{dz}{2\pi i} \Gamma(-2 - \epsilon + z) \Gamma(1 - z) f(z) + \sum_{n=1,2} \text{Res}_{z=n+\epsilon} \Gamma(-2 - \epsilon + z) \Gamma(1 - z) f(z) \\ &= \int_{C_2} \frac{dz}{2\pi i} \Gamma(-2 - \epsilon + z) \Gamma(1 - z) f(z) - \Gamma(-\epsilon) f(1 + \epsilon) + \Gamma(-1 - \epsilon) f(2 + \epsilon). \end{aligned}$$

The integral over C_2 is finite in the limit $\epsilon = 0$ and the ϵ -pole has been made explicit through $\Gamma(-\epsilon)$ and $\Gamma(-1 - \epsilon)$, which stem from the residues. The integrand can now be Taylor expanded in ϵ prior to the integration.

In the original Strategy A, the resulting contours were not restricted to straight lines in the complex plane parallel to the imaginary axis. However, the modified Strategy A imposes this restriction, which allows the contours to be parametrized more easily.

For a general N -dimensional MB integral, we introduce the notation

$$R = \int_{C_1} \frac{dz_1}{2\pi i} \cdots \int_{C_N} \frac{dz_N}{2\pi i} x_1^{C_1} \cdots x_M^{C_M} \frac{\Gamma(A_1|n_1) \cdots \Gamma(A_J|n_J) \psi^{(k_1)}(A_{J+1}|n_{j+1}) \cdots \psi^{(k_K)}(A_K|n_K)}{\Gamma(B_1) \cdots \Gamma(B_L)}, \quad (3.75)$$

where $\psi^{(k)}(x)$ denotes a polygamma function, x_i denotes kinematic variables (e.g. ratios of squared masses), and A_i, B_i, C_i are linear combinations of the z_n and d . The extra arguments n_i to the gamma and polygamma functions in the numerator (after the vertical bar) define the contours of the integration variables z_n . If $n_i = 0$, the gamma (or polygamma) function is treated in the standard way. This means that if the coefficient of an integration variable z_j in the argument A_i of the (poly)gamma function is positive (negative), all poles should lie to the left (right) of the

⁷We assume that $f(z)$ does not have any poles within the closed contour $C_1 - C_2$.

\mathcal{C}_j -contour. The case that n_i is a positive integer implies a choice of the contours, where n_i poles are on the “wrong” side. An initial MB representation does not contain polygamma functions and is always defined with standard contours, i.e. all extra arguments are zero. The polygamma functions first appear after residues in some integration variables have been taken.

Straight contours parallel to the imaginary axis with real parts given by the point $\vec{c} \equiv (c_1, \dots, c_N)^T$ will in general violate some contour prescriptions given by the extra arguments. Therefore, we define a function

$$\sigma(R, \vec{c}) = \sum_{i=1}^K \theta(A_i, n_i, \vec{c})$$

to quantify how much the contour prescriptions are violated, where

$$\theta(A, n, \vec{c}) = \begin{cases} \lfloor -A|_{\vec{z}=\vec{c}, \epsilon=0} + 1 \rfloor & ; n = 0 \wedge A|_{\vec{z}=\vec{c}, \epsilon=0} \geq 0, \\ 0 & ; n = 0 \wedge A|_{\vec{z}=\vec{c}, \epsilon=0} < 0, \\ \lfloor -A|_{\vec{z}=\vec{c}, \epsilon=0} - n + 1 \rfloor & ; n > 0 \wedge A|_{\vec{z}=\vec{c}, \epsilon=0} \geq -n + 1, \\ \infty & ; n > 0 \wedge A|_{\vec{z}=\vec{c}, \epsilon=0} < -n + 1. \end{cases}$$

The bracket $\lfloor x \rfloor$ gives the smallest integer $\geq x$ and the notation $A|_{\vec{z}=\vec{c}, \epsilon=0}$ stands for A with the integration variables z_1, \dots, z_N substituted by c_1, \dots, c_N and the regulator ϵ (and in general also analytic regulators) set to zero.

The function $\sigma(R, \vec{c})$ will give a vanishing result if all contour prescriptions are fulfilled. A positive number denotes how many poles are on the “wrong” side. An infinite value indicates a point \vec{c} useless for the algorithm.

In alg. 3.5, we show the modified Strategy A in a similar way as given in Ref. [119], but more closely to our private C++ implementation. Here, the function RESOLVE takes a MB representation R in the form (3.75), where the A_i and n_i used in the pseudo-code correspond to R . The optimization problem in line 9 can be solved with a differential evolution algorithm [120, 121]. In line 15, the contours in R are set to straight lines parallel to the imaginary axis with real parts given by \vec{p} . If the first recursion level of RESOLVE returns **Failed**, the singularities of the MB representation cannot be resolved. In that case, additional analytic regulators might be helpful.

The result of RESOLVE is a sum of contour integrals over straight lines parallel to the imaginary axis, where all singularities have been resolved. Thus, the integrands can be Taylor expanded in ϵ up to the desired order.

Algorithm 3.5 Modified Strategy A to resolve singularities in a given MB representation.

```

1: function RESOLVE( $R, \vec{p} \leftarrow \text{null}$ )
2:   if  $\vec{p} = \text{null}$  then
3:     if  $\exists(i, j) : (-A_i - n_i - A_j - n_j) \in \mathbb{N}_0$  then
4:       return Failed
5:     end if
6:     if  $\exists(i, j) : A_i = A_j \wedge n_i \neq n_j$  then
7:       return Failed
8:     end if
9:      $\vec{p} \leftarrow$  point where  $\sigma(R, \vec{p})$  is minimal
10:  end if
11:  if  $\sigma(R, \vec{p}) = \infty$  then
12:    return Failed
13:  end if
14:  if  $\sigma(R, \vec{p}) = 0$  then
15:    return  $R|_{\vec{c}=\vec{p}}$ 
16:  end if
17:  for all distinct  $(A_i, n_i)$ , where  $\theta(A_i, n_i) \neq 0$  do
18:     $R' \leftarrow R$ 
19:    increment all contour prescriptions in  $R'$  corresponding to (poly)gamma functions
    with argument  $A_i$ 
20:     $t_1 \leftarrow \text{RESOLVE}(R', \vec{p})$ 
21:    if  $t_1 \neq \text{Failed}$  then
22:       $z \leftarrow$  one of the integration variables  $z_n$  in  $A_i$ 
23:       $q \leftarrow \frac{dA_i}{dz}$ 
24:       $z_0 \leftarrow z - \frac{A_i + n_i}{q}$ 
25:       $R'' \leftarrow \text{Res}_{z=z_0} \tilde{R}$ , where  $\tilde{R}$  is the integrand of  $R$  w.r.t. the variable  $z$ 
26:       $t_2 \leftarrow \text{RESOLVE}(R'')$ 
27:      if  $t_2 \neq \text{Failed}$  then
28:        return  $t_1 - \text{sgn}(q) t_2$ 
29:      end if
30:    end if
31:  end for
32:  return Failed
33: end function

```

3.6.2 Asymptotic Expansion

MB representations are helpful instruments to find asymptotic expansions of Feynman integrals through the residue theorem. This can be seen quite easily by the same example we already discussed in Section 3.4, given in fig. 3.1. A one dimensional MB representation for this diagram can be obtained via the method described in Section 3.7 and is given by

$$I = \frac{1}{\epsilon^2 \Gamma^2(\epsilon - 1)} \int_{\mathcal{C}} \frac{dz}{2\pi i} (-4\tau)^{z-\epsilon} \frac{\Gamma(\epsilon - z)\Gamma(-z)\Gamma^2(1+z)\Gamma^2(1-\epsilon+z)\Gamma(\epsilon+z)}{\Gamma(2-2\epsilon+z)\Gamma(2+2z)},$$

where the contour \mathcal{C} fulfills the standard requirements by separating the poles of the gamma functions $\Gamma(\dots + z)$ from the poles of the gamma functions $\Gamma(\dots - z)$. We are aiming at a next-to-leading order expansion in the small kinematic variable τ . For small τ the integrand vanishes in the limits $z \rightarrow \infty$ and $z \rightarrow \pm i\infty$. Thus, the integration contour can be closed to the right. The closed contour circles the poles at $z = n + \epsilon$ and $z = n$ for $n \in \mathbb{N}_0$ clockwise. An asymptotic expansion is now obtained by summing up all residues up to the desired order in τ . For a next-to-leading order expansion, these are the residues at $z = \epsilon, 1 + \epsilon, 0, 1$, leading to

$$\begin{aligned} I &= I_1 + I_2, \\ I_1 &= \frac{\epsilon - 1}{2\epsilon^2(1 + 2\epsilon)} + (-4\tau) \frac{\epsilon - 1}{2\epsilon(6 + 13\epsilon - 4\epsilon^3)} + \mathcal{O}(\tau^2), \\ I_2 &= (-4\tau)^{-\epsilon} \frac{\Gamma^2(2 - \epsilon)}{\epsilon^2 \Gamma(2 - 2\epsilon)} + (-4\tau)^{1-\epsilon} \frac{\Gamma^2(2 - \epsilon)}{12\epsilon \Gamma(2 - 2\epsilon)} + \mathcal{O}(\tau^2). \end{aligned}$$

This result agrees with (3.59) obtained via an expansion by subgraphs. The two contributions I_1 and I_2 stem from the poles of different gamma functions and have a different scaling behavior with respect to τ . These contributions correspond to different regions in the context of an expansion by regions [57, 106–108] or to different subgraphs as discussed in Section 3.4. An ϵ -expansion of I_2 would lead to logarithms of τ in the result. Note that this is not the only source for logarithms in the asymptotic expansion since residues at higher order poles in the integrand contain logarithms in τ even before the ϵ -expansion is performed. This is of particular importance for the method described in Section 4.2.

The asymptotic expansion of multidimensional MB representations is more involved. The algorithm given in alg. 3.6 is taken from the public `Mathematica`-package `MBasymptotics` [122] by Czakon. It is limited to MB integrals with straight contours parallel to the imaginary axis. Therefore, one has to apply the modified Strategy A described in Section 3.6.1 (or Strategy B) first. As part of this thesis, we also reimplemented this algorithm in a private `C++`-code.

As first argument, the function `ASYMPTOTICS` expects an N -dimensional MB integral I of the form

$$I = \int_{c_1 - i\infty}^{c_1 + i\infty} \frac{dz_1}{2\pi i} \cdots \int_{c_N - i\infty}^{c_N + i\infty} \frac{dz_N}{2\pi i} x_1^{C_1} \cdots x_M^{C_M} \frac{\Gamma(A_1) \cdots \Gamma(A_J) \psi^{(k_{J+1})}(A_{J+1}) \cdots \psi^{(k_K)}(A_K)}{\Gamma(B_1) \cdots \Gamma(B_L)}, \quad (3.76)$$

where A_i , B_i and C_i are linear combinations of the integration variables z_1, \dots, z_N and $\epsilon = (4-d)/2$. The objects $\vec{z} = (z_1, \dots, z_N)^T$, $\vec{c} = (c_1, \dots, c_N)^T$ and A_i used in the algorithm correspond to (3.76). Line 5 of alg. 3.6 checks if closing the contours to obtain an asymptotic expansion for small x yields any terms of order less than or equal to n . If this is not the case, zero is returned. If the exponent C of x does not depend on any integration variable, the expansion in x will only contain one term. Since we already checked that the exponent of x for this single term is less than or equal to n , line 9 simply returns the input I . This is the termination condition of the recursive algorithm. In line 11, the algorithm selects one of the integration variables z_i , C depends on. This particular integration variable is denoted as z in the following. The loop in lines 15–27 scans all arguments

of the (poly)gamma functions in I for poles contributing to an expansion up to order n . These poles lie in the interval $[v_0, v_1]$ for which the boundaries have been calculated in the lines 16–22. In line 17, at first z is fixed to z_0 and then the remaining z_i are set to c_i (and the regulator ϵ to zero). After compiling a set of poles contributing to the expansion, lines 29–34 collect the residues at these poles. The residues are MB integrals with one dimension less. These MB integrals may still have dependences on the remaining integration variables in the exponent of x . This is why line 32 recursively calls ASYMPTOTICS again for all terms in the residues.

Algorithm 3.6 Asymptotic expansion of a given MB integral.

```

1: function ASYMPOTICS( $I, x, n$ )
2:   in:    $I$ : a MB integral
            $x$ : a variable to expand in
            $n$ : the order of the expansion
3:   out: the asymptotic expansion of  $I$  in  $x$  to order  $n$ 
4:    $C \leftarrow$  the exponent of  $x$  in the integrand of  $I$ 
5:   if  $C|_{\vec{z}=\vec{c}, \epsilon=0} > n$  then
6:     return 0
7:   end if
8:   if  $C$  is independent of any  $z_i$  then
9:     return  $I$ 
10:  end if
11:   $z \leftarrow$  one of the  $z_i$  in  $C$ 
12:   $z_0 \leftarrow$  the solution for  $z$  of the equation  $C = n$ , where  $\vec{z}$  is set to  $\vec{c}$  in the result
13:   $\sigma \leftarrow \operatorname{sgn} \frac{\partial C}{\partial z}$ 
14:   $P \leftarrow \{\}$ 
15:  for all distinct  $A_i$  depending on  $z$  do
16:     $v_0 \leftarrow A_i|_{\vec{z}=\vec{c}, \epsilon=0}$ 
17:     $v_1 \leftarrow (A_i|_{z=z_0})|_{\vec{z}=\vec{c}, \epsilon=0}$ 
18:    if  $v_0 > v_1$  then
19:      swap  $v_0 \leftrightarrow v_1$ 
20:    end if
21:     $v_0 = \min(\lceil v_0 \rceil, 1)$ 
22:     $v_1 = \min(\lfloor v_1 \rfloor, 0)$ 
23:    for  $v = v_0, \dots, v_1$  do
24:       $p \leftarrow$  the solution for  $z$  of the equation  $A_i = v$ 
25:       $P \leftarrow P \cup \{p\}$ 
26:    end for
27:  end for
28:   $R \leftarrow 0$ 
29:  for all  $p \in P$  do
30:     $J \leftarrow \operatorname{Res}_{z=p} \tilde{I}$ , where  $\tilde{I}$  is the integrand of  $I$  w.r.t. the variable  $z$ 
31:    for all terms  $t$  in  $J$  do
32:       $R \leftarrow R + \text{ASYMPOTICS}(t, x, n)$ 
33:    end for
34:  end for
35:  return  $-\sigma R$ 
36: end function

```

3.7 Mellin-Barnes meets Method of Brackets

For the evaluation of Feynman integrals using the MB technique, MB representations with a low number of MB integrations (dimensions) are always preferable. Therefore, we denote a MB representation as better if it requires a lower number of dimensions. The dimensionality of a MB representation strongly depends on the method used for its construction.

Two widely used techniques for the construction are the loop-by-loop approach [111, 112] and the global approach [113, 114], both implemented in the public Mathematica package `AMBRE` [111–114]. The loop-by-loop approach seems to be very successful in order to construct low-dimensional MB representations. Unfortunately, this method is limited to planar Feynman integrals. The global approach is applicable for both planar and non-planar integrals. However, it often produces very high dimensional representations that are not suitable for the subsequent evaluation.

As part of this thesis, we developed a new algorithm inspired by the Method of Brackets [115–117] to construct low-dimensional MB representations, which was published in Ref. [2]. For most cases, our method leads to better MB representations than the global-approach. This is why we prefer this method especially for non-planar Feynman integrals. For planar integrals, the loop-by-loop approach often is the better choice.

The Method of Brackets is a rather unknown technique for the evaluation of Feynman integrals. It is an improvement on an older method called Negative Dimension Integration [123].

This method defines a small set of simple rules, which, when applied to a Schwinger parametrized Feynman integral, yields a set of multifold sums. Unfortunately, in many cases not all of these sums contribute to the final result and it is sometimes hard to tell which sum does contribute and which sum should be neglected.

In this section, we modify the Method of Brackets in such a way that it leads to a set of multidimensional MB integrals instead of a set of multifold sums. From this set of solutions, a single multidimensional MB integral contains the full result of the Feynman integral. The ambiguity of the original method is therefore not present.

Ref. [115] describes an optimization procedure for the Symanzik polynomials that appear in the Schwinger parametrization of Feynman integrals. This optimization reduces the multiplicity of the resulting multifold sums in the context of the original Method of Brackets. In our adapted version the same optimization helps to minimize the number of MB integrations of the constructed representation.

In Section 3.7.1, we derive a set of rules for the adapted Method of Brackets in analogy to the rules defined in Ref. [117]. Section 3.7.2 discusses the optimization of Symanzik polynomials. At last, we present a detailed example of the method in Section 3.7.3.

3.7.1 The Modified Method of Brackets

The original Method of Brackets is based on Ramanujan’s master theorem [124] which states that if a function $g(x)$ admits a Taylor expansion

$$g(x) = \sum_{n=0}^{\infty} G(n) \frac{(-x)^n}{n!}, \quad (3.77a)$$

the integral over the parameter x is given by

$$\int_0^{\infty} dx x^{\alpha-1} g(x) = \Gamma(\alpha) G(-\alpha). \quad (3.77b)$$

The similarity of this relation to the well-known Mellin-transform

$$f(x) = \int_{c-i\infty}^{c+i\infty} \frac{dz}{2\pi i} x^z F(z), \quad (3.78a)$$

with

$$\int_0^\infty dx x^{\alpha-1} f(x) = F(-\alpha) \quad (3.78b)$$

allows reformulating the Method of Brackets in a way that leads to MB representations instead of multifold sums.

Utilizing (3.77), the original Method of Brackets formulates a set of simple rules to rewrite a Schwinger parametrized Feynman integral (3.68) into a so-called presolution of the diagram – a multifold sum over gamma functions and newly introduced objects called brackets [117]. The brackets in the presolution can then be eliminated using only linear algebra.

In this section, we present a similar set of rules, but our presolution will be a multidimensional MB integral instead of a multifold sum.

3.7.1.1 The Bracket

The central object of the technique is the bracket, which is defined as

$$\langle \alpha \rangle \equiv \int_0^\infty dx x^{\alpha-1}. \quad (3.79)$$

Of course, this object by itself is not well-defined as the integral on the right-hand side is divergent for all α . However, it makes sense inside a MB integral

$$\int_{c-i\infty}^{c+i\infty} \frac{dz}{2\pi i} \langle \alpha + z \rangle F(z) = \int_0^\infty dx \int_{c-i\infty}^{c+i\infty} \frac{dz}{2\pi i} x^{\alpha+z-1} F(z) = F(-\alpha), \quad (3.80)$$

where in the last step, we used equation (3.78). In contrast, the original Method of Brackets interprets this object inside a multifold sum using Ramanujan's master theorem (3.77).

3.7.1.2 The Rules

The rules provided in this section have to be applied successively to a Schwinger parametrized Feynman integral (3.68). During this process, Rule B has to be used multiple times if the Symanzik polynomials are given in optimized form (see Section 3.7.2 for details).

Rule A: Exponential functions. The exponential function in (3.68) is first split into factors using $e^{-\sum_i A_i} = \prod_i e^{-A_i}$ so that every exponent A_i consists only of a monomial or a monomial divided by U . Afterwards the exponential functions are rewritten into contour integrals using the Cahen-Mellin formula

$$e^{-A_i} = \int_{c_i-i\infty}^{c_i+i\infty} \frac{dz_i}{2\pi i} A_i^{z_i} \Gamma(-z_i). \quad (3.81)$$

The contour is chosen such that all singularities coming from $\Gamma(-z_i)$ are to the right of the contour (i.e. $c_i < 0$). The validity of this equation can be checked by closing the contour at $|z_i| \rightarrow \infty$ to the right and using the residue theorem.

The factor $A_i^{z_i}$ on the right-hand side of (3.81) should then be expanded to a product of powers, where the base is a single Schwinger parameter, the polynomial U , or one of the symbols introduced by the optimization procedure described in Section 3.7.2. After this, all powers of a common base have to be combined, e.g.

$$U^{-d/2} \left(\frac{x_1 x_3}{U} \right)^{z_1} \left(\frac{x_1 x_4}{U} \right)^{z_2} = U^{-d/2 - z_1 - z_2} x_1^{z_1 + z_2} x_3^{z_1} x_4^{z_2}.$$

This corresponds to Rule I in Ref. [117].

Rule B: Multinomials. Powers of multinomials occur after the insertion of the Symanzik polynomial U or the resubstitution of the symbols introduced by the optimization procedure in Section 3.7.2. These powers are then rewritten in terms of MB integrals using the formula

$$\begin{aligned} & (A_1 + \dots + A_J)^\alpha \\ &= \frac{1}{\Gamma(-\alpha)} \int_{c_1 - i\infty}^{c_1 + i\infty} \frac{dz_1}{2\pi i} \dots \int_{c_J - i\infty}^{c_J + i\infty} \frac{dz_J}{2\pi i} \langle z_1 + \dots + z_J - \alpha \rangle A_1^{z_1} \dots A_J^{z_J} \Gamma(-z_1) \dots \Gamma(-z_J). \end{aligned} \quad (3.82)$$

The formula can be derived by first applying Schwinger parametrization to the left-hand side of (3.82) and then applying Rule A and the definition of the bracket (3.79).

Afterwards, the factors $A_1^{z_1}, \dots, A_J^{z_J}$ on the right-hand side of (3.82) are expanded in the same manner as described at the end of Rule A.

This corresponds to Rule III in Ref. [117].

Rule C: Schwinger parameters. After the application of Rule A and B, the Schwinger integrals should all be of the form

$$\int_0^\infty dx_i x_i^{L(a_1, \dots; z_1, \dots) - 1},$$

where $L(a_1, \dots; z_1, \dots)$ is a linear combination of the indices a_j and the Mellin-Barnes variables z_j . The integrals can now be rewritten as brackets using the definition (3.79):

$$\int_0^\infty dx_i x_i^{L(a_1, \dots; z_2, \dots) - 1} = \langle L(a_1, \dots; z_1, \dots) \rangle.$$

This corresponds to Rule II in Ref. [117].

Rule D: Eliminating the brackets. Applying the Rules A, B and C to a Schwinger parametrized Feynman integral results in a presolution of the form

$$P = \int_{c_1 - i\infty}^{c_1 + i\infty} \frac{dz_1}{2\pi i} \dots \int_{c_J - i\infty}^{c_J + i\infty} \frac{dz_J}{2\pi i} \langle \beta_1 + \vec{\alpha}_1 \cdot \vec{z} \rangle \dots \langle \beta_K + \vec{\alpha}_K \cdot \vec{z} \rangle f(\vec{z}),$$

where $J \geq K$ and $\vec{z} = (z_1, \dots, z_J)^T$.

We first consider the case $J = K$ and define a $K \times K$ -matrix A by

$$A = \begin{pmatrix} \vec{\alpha}_1^T \\ \vdots \\ \vec{\alpha}_K^T \end{pmatrix},$$

where we assume for now its invertibility. A change of basis $\vec{z} = -A^{-1}\vec{s}$ leads to

$$P = \frac{1}{|\det A|} \int_{d_1-i\infty}^{d_1+i\infty} \frac{ds_1}{2\pi i} \cdots \int_{d_K-i\infty}^{d_K+i\infty} \frac{ds_K}{2\pi i} \langle \beta_1 - s_1 \rangle \cdots \langle \beta_K - s_K \rangle f(-A^{-1}\vec{s}).$$

Note the change in the integration contour to $(d_1, \dots, d_K)^T = -A(c_1, \dots, c_K)^T$. Now all MB integrations can be solved one-by-one using (3.80), which yields

$$P = \frac{1}{|\det A|} f(-A^{-1}\vec{\beta}),$$

where $\vec{\beta} = (\beta_1, \dots, \beta_K)^T$.

In the case $J > K$, this formula can be used to solve K out of the J MB integrations. The result will be a $(J - K)$ -dimensional MB integral. Without loss of generality, we solve the MB integrals over z_1, \dots, z_K using the K brackets while the $J - K$ integrals over z_{K+1}, \dots, z_J should remain. Therefore, we arrange the first K integration variables into a vector $\vec{z}_1 = (z_1, \dots, z_K)^T$ and the variables of the remaining integrals into a vector $\vec{z}_2 = (z_{K+1}, \dots, z_J)^T$. Now, we can write down our last rule:

$$\begin{aligned} & \int_{c_1-i\infty}^{c_1+i\infty} \frac{dz_1}{2\pi i} \cdots \int_{c_J-i\infty}^{c_J+i\infty} \frac{dz_J}{2\pi i} \langle \beta_1 + \vec{\alpha}_1 \cdot \vec{z}_1 + \vec{\gamma}_1 \cdot \vec{z}_2 \rangle \cdots \langle \beta_K + \vec{\alpha}_K \cdot \vec{z}_1 + \vec{\gamma}_K \cdot \vec{z}_2 \rangle f(\vec{z}_1, \vec{z}_2) \\ &= \frac{1}{|\det A|} \int_{c_{K+1}-i\infty}^{c_{K+1}+i\infty} \frac{dz_{K+1}}{2\pi i} \cdots \int_{c_J-i\infty}^{c_J+i\infty} \frac{dz_J}{2\pi i} f(-A^{-1}\vec{\beta} - A^{-1}C \vec{z}_2, \vec{z}_2), \end{aligned}$$

where the $K \times (J - K)$ -matrix C is given by $C = (\vec{\gamma}_1, \dots, \vec{\gamma}_K)^T$ and the $\vec{\gamma}_i$ contain the coefficients of \vec{z}_2 in the brackets. The vector $\vec{\beta}$ and the matrix A are again defined as K -dimensional quantities in the same way as before.

The choice which integrals should be solved by this formula and which integrals should remain is somewhat arbitrary. Some of the $\binom{J}{K}$ possibilities lead to a singular matrix A and yield no solution. All other choices lead to a possible MB representation for the full result⁸, which implies that we only have to consider one of them.

This is a major advantage over the original Method of Brackets, where the individual sum only gives a partial result for the Feynman integral and various choices (but not all) have to be considered to obtain a full result.

We let the question unanswered, whether some of the obtained MB representations are in some sense better than others. More studies are necessary to tackle this problem.

This corresponds to Rule IV in Ref. [117].

⁸Unfortunately, we cannot present a proof that a choice with $\det A \neq 0$ always exists.

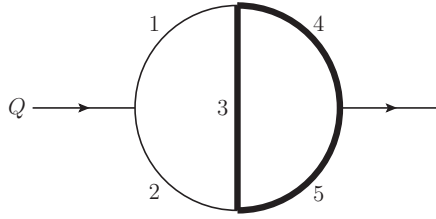


Fig. 3.4: Two-loop propagator diagram.

3.7.2 Optimization Procedure

Rules A to D applied to (3.68) are sufficient to obtain a MB representation for a given Feynman integral. However, the naïve application of these rules often leads to a huge number of MB integrations in the result.

Better MB representations can be achieved by first analyzing the Symanzik polynomials U and F as well as the polynomial $\sum_i x_i m_i^2$ for subexpressions (polynomials of Schwinger parameters) that appear multiple times. These common subexpressions are then substituted by new variables which are later treated as Schwinger parameters for Rule A and Rule B. This can be done recursively.

The Schwinger parametrized Feynman integral (3.68) can now be treated with the set of rules given in Section 3.7.1.2 as before. However, after the first application of Rule B (to the power of base U), the intermediate result contains powers of the variables introduced by the optimization, such that Rule C cannot yet be applied. After the corresponding subexpressions are resubstituted, Rule B has to be used on these powers as well. Only after all optimization variables have been eliminated, one can continue with Rule C.

This procedure reduces the number of MB integrals in the result significantly. If a polynomial ξ with N terms appears J times in U , F and $\sum_i x_i m_i^2$, the substitution of ξ decreases the number of terms in these polynomials by $J(N - 1)$. After Rule A and the first application of Rule B, the number of MB integrals is therefore reduced by $J(N - 1)$ as well. However, the second application of Rule B, after ξ is reinserted, produces N additional MB integrals and one additional bracket. In the end, this optimization leads therefore to a reduction of the number of MB integrals in the final result (i.e. after Rule D) by $(J - 1)(N - 1)$.

This optimization approach was first proposed in Ref. [115] in the context of the original Method of Brackets. An algorithm to find common subexpressions in a list of polynomials is given in Appendix E.

3.7.3 Example

As an example, we consider the two-loop propagator diagram in fig. 3.4. The corresponding Euclidean Feynman integral is given by

$$I(a_1, \dots, a_5) = \int \frac{d^d L_1}{\pi^{d/2}} \int \frac{d^d L_2}{\pi^{d/2}} \frac{1}{[L_1^2]^{a_1} [(L_1 - Q)^2]^{a_2} [(L_1 - L_2)^2 + m^2]^{a_3} [L_2^2 + m^2]^{a_4} [(L_2 - Q)^2 + m^2]^{a_5}},$$

and the Schwinger parametrization by

$$I(a_1, \dots, a_5) = \frac{1}{\Gamma(a_1) \dots \Gamma(a_5)} \int_0^\infty dx_1 x_1^{a_1-1} \dots \int_0^\infty dx_5 x_5^{a_5-1} \frac{e^{-F/U - m^2(x_3+x_4+x_5)}}{U^{d/2}},$$

with

$$U = x_2x_5 + x_1x_3 + x_1x_5 + x_3x_4 + x_1x_4 + x_2x_3 + x_3x_5 + x_2x_4,$$

$$F = Q^2(x_1x_2x_5 + x_1x_2x_3 + x_1x_3x_5 + x_1x_4x_5 + x_1x_2x_4 + x_2x_4x_5 + x_2x_3x_4 + x_3x_4x_5).$$

A naïve application of Rules A to D without optimization would lead to a 13-fold MB representation.

In order to reduce this number, we first identify common subexpressions in U , F and $\sum_i x_i m_i^2 = m^2(x_3 + x_4 + x_5)$ and replace them by new variables r_i :

$$\begin{aligned} r_1 &= x_3 + x_4, \\ r_2 &= r_1 + x_5, \\ r_3 &= r_2x_1 + x_3x_4, \\ U &= r_2x_2 + r_3 + x_3x_5, \\ F &= q^2(x_2x_4x_5 + r_1x_1x_5 + r_3x_2 + x_3x_4x_5), \\ \sum_i x_i m_i^2 &= m^2 r_2. \end{aligned}$$

Rule A then leads to

$$I(a_1, \dots, a_5) = \int_0^\infty dx_1 \cdots \int_0^\infty dx_5 \int_{c_1-i\infty}^{c_1+i\infty} \frac{dz_1}{2\pi i} \cdots \int_{c_5-i\infty}^{c_5+i\infty} \frac{dz_5}{2\pi i} (Q^2)^{z_{1234}} (m^2)^{z_5} \frac{\Gamma(-z_1) \cdots \Gamma(-z_5)}{\Gamma(a_1) \cdots \Gamma(a_5)} \\ \cdot U^{-d/2 - z_{1234}} r_1^{z_2} r_2^{z_5} r_3^{z_3} x_1^{a_1+z_2-1} x_2^{a_2+z_{13}-1} x_3^{a_3+z_4-1} x_4^{a_4+z_{14}-1} x_5^{a_5+z_{124}-1},$$

where the index notation is explained in Appendix A. Note that we have combined all powers of a common base. As a next step, the Symanzik polynomial U in optimized form is reinserted and Rule B applied:

$$I(a_1, \dots, a_5) = \int_0^\infty dx_1 \cdots \int_0^\infty dx_5 \int_{c_1-i\infty}^{c_1+i\infty} \frac{dz_1}{2\pi i} \cdots \int_{c_8-i\infty}^{c_8+i\infty} \frac{dz_8}{2\pi i} (Q^2)^{z_{1234}} (m^2)^{z_5} \langle d/2 + z_{1234678} \rangle \\ \cdot \frac{\Gamma(-z_1) \cdots \Gamma(-z_8)}{\Gamma(a_1) \cdots \Gamma(a_5) \Gamma(d/2 + z_{1234})} \\ \cdot r_1^{z_2} r_2^{z_5} r_3^{z_3} x_1^{a_1+z_2-1} x_2^{a_2+z_{136}-1} x_3^{a_3+z_{48}-1} x_4^{a_4+z_{14}-1} x_5^{a_5+z_{1248}-1},$$

Now, Rule B must be used again three times for r_3 , r_2 and r_1 in that order, which leads to

$$I(a_1, \dots, a_5) = \int_0^\infty dx_1 \cdots \int_0^\infty dx_5 \int_{c_1-i\infty}^{c_1+i\infty} \frac{dz_1}{2\pi i} \cdots \int_{c_e-i\infty}^{c_e+i\infty} \frac{dz_e}{2\pi i} (Q^2)^{z_{1234}} (m^2)^{z_5} \\ \cdot \langle d/2 + z_{1234678} \rangle \langle z_{9a} - z_{37} \rangle \langle z_{bc} - z_{569} \rangle \langle z_{de} - z_{2b} \rangle \\ \cdot \frac{\Gamma(-z_1) \cdots \Gamma(-z_e)}{\Gamma(a_1) \cdots \Gamma(a_5) \Gamma(d/2 + z_{1234}) \Gamma(-z_{37}) \Gamma(-z_{569}) \Gamma(-z_{2b})} \\ \cdot x_1^{a_1+z_{29}-1} x_2^{a_2+z_{136}-1} x_3^{a_3+z_{48ad}-1} x_4^{a_4+z_{14ae}-1} x_5^{a_5+z_{1248c}-1}.$$

Now we can apply Rule C to obtain the presolution

$$I(a_1, \dots, a_5) = \int_{c_1-i\infty}^{c_1+i\infty} \frac{dz_1}{2\pi i} \cdots \int_{c_e-i\infty}^{c_e+i\infty} \frac{dz_e}{2\pi i} (Q^2)^{z_{1234}} (m^2)^{z_5} \langle a_1 + z_{29} \rangle \langle a_2 + z_{136} \rangle \langle a_3 + z_{48ad} \rangle \\ \cdot \langle a_4 + z_{14ae} \rangle \langle a_5 + z_{1248c} \rangle \langle d/2 + z_{1234678} \rangle \langle z_{9a} - z_{37} \rangle \langle z_{bc} - z_{569} \rangle \langle z_{de} - z_{2b} \rangle \\ \cdot \frac{\Gamma(-z_1) \cdots \Gamma(-z_e)}{\Gamma(a_1) \cdots \Gamma(a_5) \Gamma(d/2 + z_{1234}) \Gamma(-z_{37}) \Gamma(-z_{569}) \Gamma(-z_{2b})}, \quad (3.83)$$

with 14 MB integrals and nine brackets, which leads to only *five* MB integrations at the end (in contrast to the unoptimized 13).

From the $\binom{14}{9} = 2002$ possibilities only 957 lead to a non-singular matrix A . We choose for the example the MB integrals over z_1, z_2, z_3, z_4, z_7 to remain. The vectors \vec{z}_1, \vec{z}_2 and $\vec{\beta}$ and the matrices A and C defined in Rule D read

$$\vec{z}_1 = \begin{pmatrix} z_5 \\ z_6 \\ z_8 \\ z_9 \\ z_a \\ z_b \\ z_c \\ z_d \\ z_e \end{pmatrix}, \quad \vec{z}_2 = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_7 \end{pmatrix}, \quad \vec{\beta} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ \frac{d}{2} \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{pmatrix}.$$

Using the formulas of Rule D, we have to substitute

$$\begin{aligned} z_5 &\rightarrow d - a_{12345} - z_{1234}, & z_6 &\rightarrow -a_2 - z_{13}, \\ z_8 &\rightarrow -\frac{d}{2} + a_2 - z_{247}, & z_9 &\rightarrow -a_1 - z_2, \\ z_a &\rightarrow a_1 + z_{237}, & z_b &\rightarrow \frac{d}{2} - 2a_1 - a_{234} - z_{147} - 2z_{23}, \\ z_c &\rightarrow \frac{d}{2} - a_{25} - z_{17}, & z_d &\rightarrow \frac{d}{2} - a_{123} - z_3, \\ z_e &\rightarrow -a_{14} - z_{12347} \end{aligned}$$

in (3.83), which yields the final five-dimensional MB representation

$$\begin{aligned} &I(a_1, \dots, a_5) \\ &= \int_{c_1-i\infty}^{c_1+i\infty} \frac{dz_1}{2\pi i} \cdots \int_{c_4-i\infty}^{c_4+i\infty} \frac{dz_4}{2\pi i} \int_{c_7-i\infty}^{c_7+i\infty} \frac{dz_7}{2\pi i} (m^2)^{d-a_{12345}-z_{1234}} (Q^2)^{z_{1234}} \\ &\quad \cdot \frac{\Gamma(-z_1) \cdots \Gamma(-z_4) \Gamma(-z_7) \Gamma(-d + a_{12345} + z_{1234}) \Gamma(a_1 + z_2) \Gamma(-d/2 + a_{123} + z_3)}{\Gamma(a_1) \cdots \Gamma(a_5)} \\ &\quad \cdot \frac{\Gamma(a_2 + z_{13}) \Gamma(-d/2 + a_{25} + z_1 - z_7) \Gamma(-a_1 - z_{237}) \Gamma(d/2 - a_2 + z_{247})}{\Gamma(d/2 + z_{1234}) \Gamma(-d + 2a_{12} + a_{345} + 2z_{1234})} \\ &\quad \cdot \frac{\Gamma(a_{14} + z_{12347}) \Gamma(-d/2 + 2a_1 + a_{234} + z_{147} + 2z_{23})}{\Gamma(-z_{37}) \Gamma(-d/2 + 2a_1 + a_{234} + z_{1247} + 2z_3)}. \end{aligned}$$

Master Integrals 4

In the previous chapter, we described various calculation methods for master integrals in a rather general way. In this chapter, we present methods that apply directly to the calculation of the 328 master integrals in the $H \rightarrow \gamma\gamma$ decay.

Section 4.1 covers one huge class of master integrals, i.e. integrals that can be expressed in terms of GPLs. For the remaining master integrals, we introduce a promising technique in Section 4.2 with which we hope to provide an expansion for small x (i.e. small quark masses).

Note that the methods described in this chapter are all work in progress, but have been proven to be successful in many test cases.

4.1 Master Integrals and Goncharov Polylogarithms

In the calculation of the $H \rightarrow \gamma\gamma$ amplitude, 151 master integrals appear that can be expressed in terms of GPLs and rational functions in x . These master integrals either belong to the 47 sectors depicted in fig. 4.1, are tadpole integrals, or factorize into lower loop integrals. The number in parentheses indicates the number of master integrals in this sector.

Due to limitations of Lee's algorithm the strategy of Section 3.3 was not directly applicable. Hence, we present a slightly modified version in Section 4.1.1. Section 4.1.2 describes the setup of an expansion by subgraphs of the master integrals in the large quark mass limit, which can be used as boundary conditions.

4.1.1 Canonical Diagonal Blocks

The concept of canonical bases was introduced in Section 3.3. The implementation `epsilon` of a powerful algorithm by Lee [63] was also presented in that section. Unfortunately, it is a well known issue that Lee's algorithm does not always succeed in the last step, the factorization of ϵ (see, e.g. Ref. [125]). This problem, which affects the global transformation of the system of differential equations to ϵ -form, also appeared for our set of master integrals. Luckily, a global ϵ -form is not required to solve the master integrals in terms of GPLs as demonstrated below.

In order to derive a solution for all master integrals, we use the fact that a convenient ordering of the master integrals leads to a block triangular form of the system of differential equations (cf. Section 3.2). The diagonal blocks can then be transformed into ϵ -form (3.29) and the off-diagonal blocks into Fuchsian form (3.28) using the tool `epsilon`.

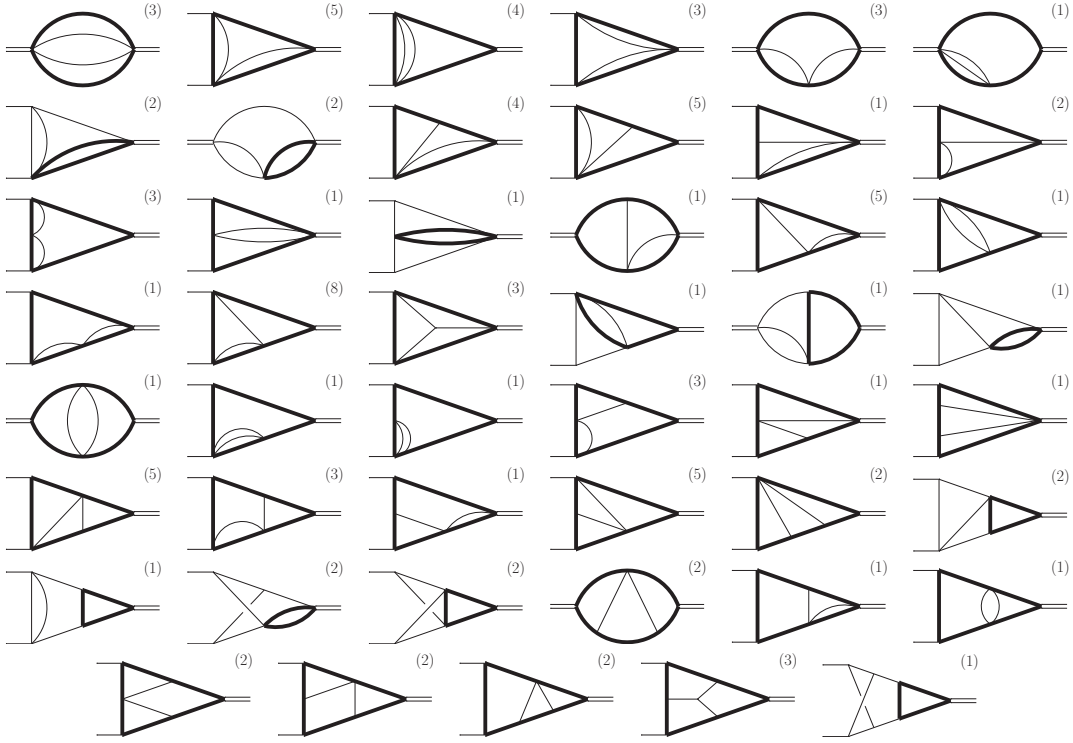


Fig. 4.1: Sectors of master integrals expressible through GPLs. The sectors of tadpole integrals and factorizing integrals are not shown. The number in parentheses indicates the number of master integrals in the sector.

We now consider a system with two diagonal blocks fulfilling

$$\frac{\partial}{\partial x} \vec{F}(x, \epsilon) = \mathbb{M}(x, \epsilon) \vec{F}(x, \epsilon),$$

where

$$\mathbb{M}(x, \epsilon) = \begin{pmatrix} \mathbb{A}(x, \epsilon) & 0 \\ \mathbb{B}(x, \epsilon) & \mathbb{C}(x, \epsilon) \end{pmatrix}, \quad \vec{F}(x, \epsilon) = \begin{pmatrix} \vec{g}(x, \epsilon) \\ \vec{f}(x, \epsilon) \end{pmatrix}, \quad (4.1)$$

and $\mathbb{C}(x, \epsilon)$ is in ϵ -form and $\mathbb{B}(x, \epsilon)$ in Fuchsian form, i.e.

$$\mathbb{C}(x, \epsilon) = \epsilon \sum_{x_j \in S} \frac{\widehat{\mathbb{C}}_0^{(x_j)}}{x - x_j}, \quad (4.2a)$$

$$\mathbb{B}(x, \epsilon) = \sum_{x_j \in S} \frac{\mathbb{B}_0^{(x_j)}(\epsilon)}{x - x_j}. \quad (4.2b)$$

The number of components in $\vec{g}(x, \epsilon)$ and $\vec{f}(x, \epsilon)$ coincides with the dimensions of the square matrices $\mathbb{A}(x, \epsilon)$ and $\mathbb{C}(x, \epsilon)$, respectively. Let the master integrals in $\vec{g}(x, \epsilon)$ be already known as a linear combination of GPLs with argument x . For example, if $\mathbb{A}(x, \epsilon)$ is given in ϵ -form the solution of $\vec{g}(x, \epsilon)$ can be obtained via the strategy of Section 3.3.4.

The master integrals in $\vec{f}(x, \epsilon)$ fulfill the inhomogeneous system of differential equations

$$\frac{\partial}{\partial x} \vec{f}(x, \epsilon) = \mathbb{C}(x, \epsilon) \vec{f}(x, \epsilon) + \mathbb{B}(x, \epsilon) \vec{g}(x, \epsilon). \quad (4.3)$$

As a first step, we solve the homogeneous part using an evolution operator $U(x, x_0; \epsilon)$, which fulfills

$$\frac{\partial}{\partial x} U(x, x_0; \epsilon) = \mathbb{C}(x, \epsilon) U(x, x_0; \epsilon) \quad ; \quad U(x_0, x_0; \epsilon) = \mathbb{1}. \quad (4.4)$$

Since $\mathbb{C}(x, \epsilon)$ is in ϵ -form, this can be done in terms of a power series in ϵ as described in Section 3.3.4. Inserting the ansatz

$$\vec{f}(x, \epsilon) = U(x, x_0; \epsilon) \vec{h}(x, \epsilon) \quad (4.5)$$

into (4.3) yields

$$\left(\frac{\partial}{\partial x} U(x, x_0; \epsilon) \right) \vec{h}(x, \epsilon) + U(x, x_0; \epsilon) \frac{\partial}{\partial x} \vec{h}(x, \epsilon) = \mathbb{C}(x, \epsilon) U(x, x_0; \epsilon) \vec{h}(x, \epsilon) + \mathbb{B}(x, \epsilon) \vec{g}(x, \epsilon).$$

Using (4.4), this equation simplifies to

$$U(x, x_0; \epsilon) \frac{\partial}{\partial x} \vec{h}(x, \epsilon) = \mathbb{B}(x, \epsilon) \vec{g}(x, \epsilon).$$

Multiplying with $U^{-1}(x, x_0; \epsilon) = U(x_0, x; \epsilon)$ from the left and integrating leads to

$$\vec{h}(x, \epsilon) = \int_{x_0}^x dx' U(x_0, x'; \epsilon) \mathbb{B}(x', \epsilon) \vec{g}(x', \epsilon) + \vec{h}_0(x_0, \epsilon).$$

We have chosen the lower limit such that the integral vanishes at $x = x_0$. Inserting $\vec{h}(x, \epsilon)$ into the ansatz (4.5) yields

$$\vec{f}(x, \epsilon) = \int_{x_0}^x dx' U(x, x'; \epsilon) \mathbb{B}(x', \epsilon) \vec{g}(x', \epsilon) + U(x, x_0; \epsilon) \vec{h}_0(x_0, \epsilon), \quad (4.6)$$

where we used $U(x, x_0; \epsilon) U(x_0, x'; \epsilon) = U(x, x'; \epsilon)$. At $x = x_0$, (4.6) evaluates to

$$\vec{f}(x_0, \epsilon) = \vec{h}_0(x_0, \epsilon),$$

which is why $\vec{h}_0(x_0, \epsilon)$ can be matched to the boundary conditions at $x = x_0$.

The integrand of (4.6) consists of three ingredients: the evolution operator $U(x, x'; \epsilon)$, the master integrals $\vec{g}(x', \epsilon)$ and the off-diagonal block $\mathbb{B}(x', \epsilon)$ in (4.1). Note that every term of $U(x, x'; \epsilon)$ and $\vec{g}(x', \epsilon)$ is a product of GPLs with argument x' and $\mathbb{B}(x', \epsilon)$ is given in Fuchsian form (4.2b).

The shuffle identity of GPLs (3.2) allows expressing the products of GPLs with argument x' as linear combinations of single GPLs. Since $\mathbb{B}(x', \epsilon)$ is Fuchsian, all the resulting integrals are of the form

$$\int_{x_0}^x \frac{dx'}{x' - x_j} G(a_1, \dots, a_n | x') = G(x_j, a_1, \dots, a_n | x) - G(x_j, a_1, \dots, a_n | x_0),$$

where we used the definition of the GPLs (3.1).

Even though the system (4.1) was not in global ϵ -form, we were able to solve $\vec{f}(x, \epsilon)$ solely in terms of GPLs. Therefore, we can continue with a possible next block of the system, where the master integrals $\vec{F}(x, \epsilon)$ play the role of $\vec{g}(x, \epsilon)$ in (4.1). This consideration allows solving all master integrals on a block-by-block basis.

For our master integrals, we implemented this strategy as code for the FORM computer algebra system [52], where we made use of the `shuffle` command, which implements the shuffle algebra of GPLs.

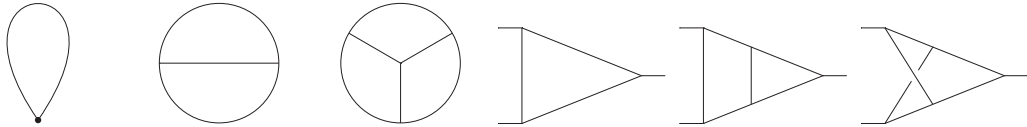


Fig. 4.2: One-scale topologies. At least one line of the tadpole diagrams has to be massive. The lines of the vertex diagrams are all massless.

4.1.2 Boundary Conditions

The boundary conditions are calculated around $x = x_0 = 1$ via an expansion by subgraphs, as described in Section 3.4. The limit $x \rightarrow 1$ corresponds to the limit of a small value τ or a large quark mass. The asymptotic expansions are calculated in the standard basis, chosen by the integral ordering of the reduction algorithm, and have to be transformed to the semi-canonical basis of Section 4.1.1 afterwards. This is necessary since the expansion method requires a diagrammatic representation of the master integrals, which is not given in the basis found by Lee’s algorithm.

For the expansion by subgraphs, we use the public C++ program `exp` [50, 51]. This tool requires the integrand of the Feynman integral as FORM-code as well as information about the corresponding graph. Therefore, we reconstruct the graph from the list of propagators using a private `Mathematica` [84] implementation of an algorithm described in Ref. [81], which is originally from the `Reduze`-package. The tool `exp` performs the identification of subgraphs as described in Section 3.4 and maps the resulting one-scale integrals onto the topologies in fig. 4.2. For convenience, we remap these (products of) one-scale topologies in a later step to the topologies in fig. 2.3, where the topologies T316, T317 and T3111 are sufficient to express all possible three-loop products of the topologies in fig. 4.2. This remapping enabled us to reuse some of the code described in Chapter 2. Furthermore, a direct treatment of the one-scale topologies would lead to tensor integrals, which require extensive tensor reduction routines. Since the order of these tensor integrals increases with the order of the asymptotic expansion, this becomes unfeasible very quickly.

With our approach, the Feynman integrals to be calculated can again be expressed in terms of the scalar function F defined in (2.7). These functions can then be reduced to master integrals, where we used a combination of `LiteRed` [73, 74] and `FIRE` [77–79] this time. The resulting master integrals are, as expected, products of one-scale integrals. The one-scale integrals to be calculated are depicted in fig. 4.3. The steps required for their evaluation as well as their solutions are given in Appendix C.

The asymptotic expansions can afterwards be expressed as a series in $(1 - x_0)$ and $\ln(1 - x_0)$. After transforming these series to the semi-canonical basis of Section 4.1.1, they can be inserted as $\vec{h}_0(x_0, \epsilon)$ into (4.6). In general, naively setting $x_0 = 1$ in (4.6) will fail since $\vec{h}_0(x_0, \epsilon)$ contains poles at $x_0 = 1$ and both $U(x, x_0; \epsilon)$ and $\vec{h}_0(x_0, \epsilon)$ contain logarithms of $(1 - x_0)$. Therefore, we expand the GPLs with argument x_0 in $U(x, x_0; \epsilon)$ around $x_0 = 1$ using the algorithm of Section 3.1.2, which cancels all poles and dangerous logarithms. Subsequently, one can safely set $x_0 = 1$ to obtain the final results for the master integrals.

4.2 Differential Equation Expansion

Unfortunately, the class of GPLs is not sufficient to express all Feynman integrals. The simplest example of a Feynman integral that cannot be expressed in terms of GPLs is the two-loop massive sunrise diagram in fig. 4.4. The theoretical understanding of this diagram is founded on the works of Ref. [126–128]. In Ref. [129] it is shown that an analytic expression requires a complete elliptic integral of the first kind. As expected, this problematic feature becomes worse at three-loop [130].

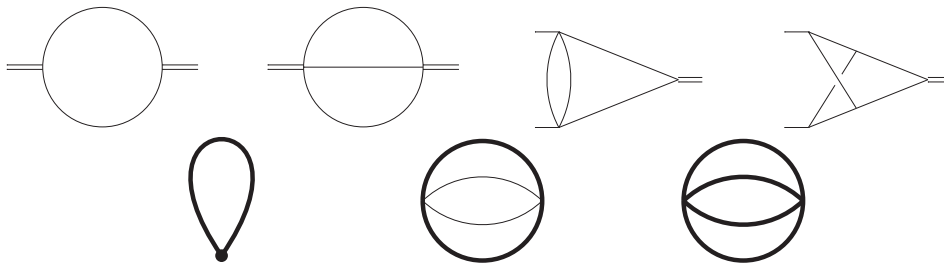


Fig. 4.3: *One-scale master integrals*

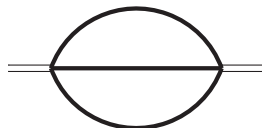


Fig. 4.4: *Two-loop massive sunrise-diagram.*

Despite recent progress towards a solution strategy for Feynman integrals involving an “elliptic sector” [131–134], we do not pursue this direction any further. Instead, we aim for an expansion in small quark masses for the affected master integrals.

For a total number of 177 master integrals of the process $H \rightarrow \gamma\gamma$, GPLs of argument x are insufficient to express the results¹. The corresponding sectors are shown in fig. 4.5. We solve these master integrals for small x via an expansion of their system of differential equation in a Laurent series. This expanded differential equation can then be solved using an appropriate ansatz for the master integrals. Such an approach is not new, see e.g. Ref. [135, 136]. The strategy presented in this section can, of course, also be applied to those master integrals that are indeed expressible through GPLs. This provides a crucial crosscheck for both the results of the method in Section 4.1 and the expansion procedure.

Section 4.2.1 explains technical details for such an expansion. In Section 4.2.2, an ansatz for the master integrals as a series in x and $\ln(x)$ is suggested. A way to determine the relevant scaling regions is presented in Section 4.2.3. If the relevant scaling regions are known, the coefficients of the ansatz can be reduced to a minimal set of master coefficients. This is done via a Laporta-like algorithm, described in Section 4.2.4. The most complicated part of this approach, the computation of the master coefficients, is covered in Section 4.2.5.

4.2.1 Expansion of the System of Differential Equations in Small x

The starting point for our expansion procedure is a Laurent series around $x = 0$ for the system of differential equation.

We choose an initial basis of master integrals $\vec{f}(x, \epsilon)$ that obeys

$$\frac{\partial}{\partial x} \vec{f}(x, \epsilon) = \mathbf{M}(x, \epsilon) \vec{f}(x, \epsilon) \quad (4.7)$$

with

$$\mathbf{M}(x, \epsilon) = \sum_i \sum_{k=1}^{\hat{k}} \sum_{j=0}^{\hat{j}_i} \frac{x^j \mathbf{M}_i^{(k,j)}(\epsilon)}{p_i^{k+1}(x)} + \sum_{i=0}^{\infty} x^i \mathbf{M}_i(\epsilon). \quad (4.8)$$

¹This number includes the master integrals that can in principle be expressed in terms of GPLs but with a kinematic variable different from x (see the discussion in Section 2.5).

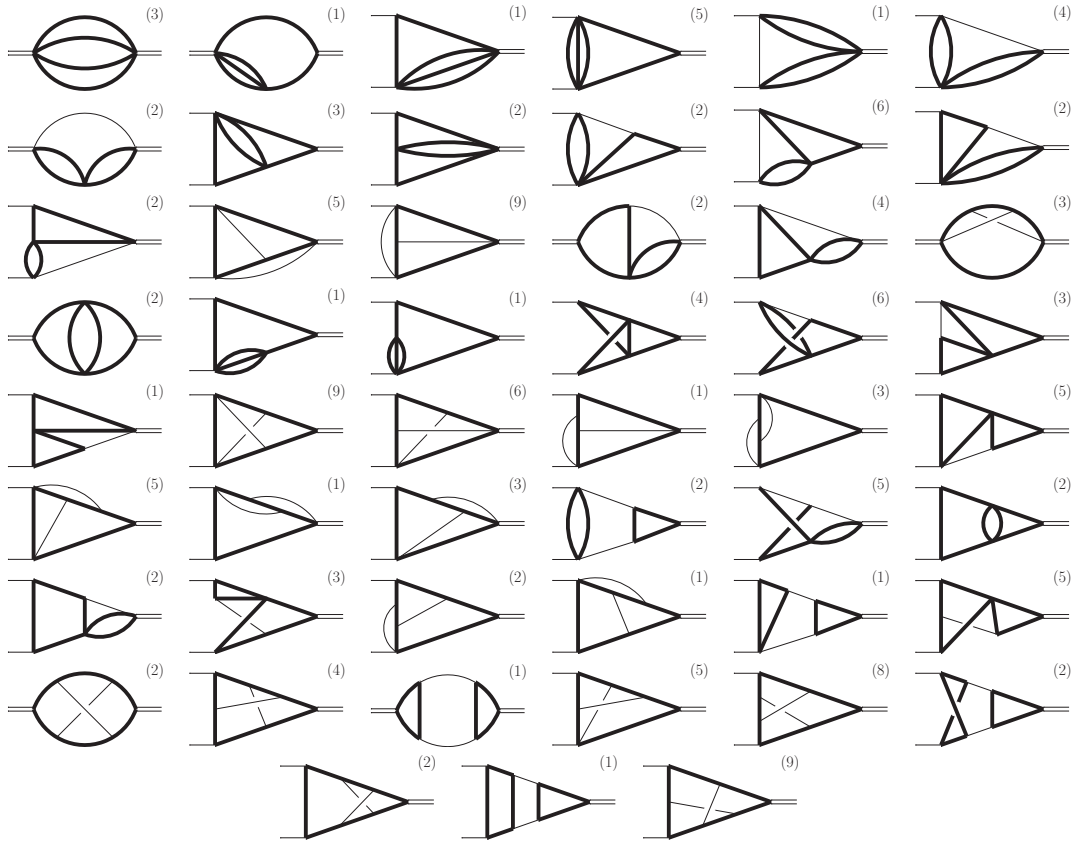


Fig. 4.5: Sectors of master integrals not expressible through GPLs. Sectors of factorizing integrals are not shown. The number in parentheses indicates the number of master integrals in the sector.

The $p_i(x)$ are first or second degree monic polynomials in x and do not depend on ϵ . The upper limit of the j -sum is given by $\hat{j}_i = \deg p_i(x) - 1$, i.e. it is zero or one. This form can be obtained by a partial fraction decomposition over the reals. If the $p_i(x)$ are not independent of ϵ , another basis has to be chosen by trial and error.

Since we restrict ourselves to first and second degree polynomials in the denominator, only two formulas are required to expand (4.8) in x . For monic first degree polynomials, we have

$$\frac{1}{(x-a)^{k+1}} = -(-1)^k \sum_{n=0}^{\infty} \binom{k+n}{n} a^{-k-n-1} x^n \quad ; \quad a \neq 0 \quad (4.9a)$$

and for monic second degree polynomials

$$\frac{x^j}{(x^2+px+q)^{k+1}} = \sum_{m=0}^{\infty} x^{j+m} \sum_{n=0}^{\lfloor \frac{m}{2} \rfloor} (-1)^{m-n} \binom{k+m-n}{m-n} \binom{m-n}{m-2n} p^{m-2n} q^{n-k-m-1}. \quad (4.9b)$$

The case of a monic first degree polynomial with $a = 0$ in the denominator is trivial, since $1/x^{k+1}$ does not need to be expanded. The same is true for the second term in (4.8).

For the Laurent expansion of (4.8), we introduce the notation

$$\mathbf{M}(x, \epsilon) = \sum_{l=l_0}^{\infty} x^l \mathbf{M}_l(\epsilon). \quad (4.10)$$

The coefficient matrices $\mathbf{M}_l(\epsilon)$ can be directly obtained from (4.8) using the formulas (4.9).

4.2.2 An Ansatz for Feynman Integrals

A well-known property of a Feynman integral is that it exhibits logarithmic singularities in the limit of vanishing internal masses (see e.g. Ref. [53, 57]). Hence, the proper expansion near $x = 0$ is expected to be a series in x and $\ln(x)$. Therefore, we consider the ansatz

$$\vec{f}(x, \epsilon) = \sum_k \sum_{j \geq j_0} \sum_{n=0}^K \vec{c}_{kjn}(\epsilon) x^{j+k\epsilon} \ln^n(x). \quad (4.11)$$

Already in the discussion of two asymptotic expansion methods, the expansion by subgraphs (see Section 3.4) and the expansion by Mellin-Barnes representation (see Section 3.6.2), it became clear that various terms with different scaling behaviors may contribute to the expansion of a given Feynman integral. These scaling regions are represented by the index k in (4.11). The powers of the logarithms $\ln(x)$ are limited by the number of loops K . This value is a matter of experience and can only be proven for simple cases [57].

One might ask, whether the logarithmic behavior of a Feynman integral has perhaps already been covered by the term $x^{j+k\epsilon}$ when expanded in ϵ . In other kinematic limits (e.g. Euclidean limits, where the method of subgraphs can be applied) this is indeed true. Unfortunately, in the limit under consideration, i.e. the limit of vanishing quark masses, additional logarithms appear. This can be understood in the context of an expansion by Mellin-Barnes representations, where higher order poles in the integration variables generally lead to residues containing logarithms of the kinematic variables. In an expansion by regions [57, 106–108], a further expansion method, the emergence of these logarithms is made even clearer. Here, contributions appear that cannot be dimensionally regularized so that additional analytic regulators are required. The exponents of x then also depend on these extra regulators in a way that is similar to their dependence on ϵ . A subsequent Laurent expansion in the analytic regulators leads to the additional logarithms.

Plugging (4.11) and $\mathbb{M}(x, \epsilon)$ in the expanded form of (4.10) into (4.7) yields the recurrence relations

$$0 = (j + k\epsilon)\vec{c}_{kjn}(\epsilon) + (n + 1)\vec{c}_{kj;n+1}(\epsilon) - \sum_{l=l_0}^{j-j_0-1} \mathbb{M}_l(\epsilon) \vec{c}_{k;j-l-1;n}(\epsilon) \quad (4.12)$$

for the coefficients $\vec{c}_{jkn}(\epsilon)$, where $\vec{c}_{jkn} = 0$ for $j < j_0$ or $n > K$. The recurrence relations (4.12) are helpful to express the coefficients of (4.11) up to a desired order in terms of a small number of “master coefficients” (see Section 4.2.4). Experience shows that the number of master coefficients remains constant even when the order of the expansion is increased.

4.2.3 Relevant Scaling Regions

An important feature of (4.12) is that it does not couple coefficients with different indices k . Therefore, one has to determine the relevant scaling regions (i.e. the relevant k -values) before (4.12) can be used for a reduction to master coefficients.

First, we consider a system of differential equations for $\vec{f}(x, \epsilon)$, where the singularity at $x = 0$ is Fuchsian. For the leading-order Laurent expansion of this system, we write

$$\frac{\partial}{\partial x} \vec{f}(x, \epsilon) \approx \frac{\mathbb{M}(\epsilon)}{x} \vec{f}(x, \epsilon). \quad (4.13)$$

The solution is given by

$$\vec{f}(x, \epsilon) = e^{\mathbb{M}(\epsilon) \ln(x)} \vec{f}_0(\epsilon), \quad (4.14)$$

which is of course approximate due to the neglected terms in (4.13). The matrix exponential can be treated in the standard way by transforming $\mathbf{M}(\epsilon)$ to Jordan normal form

$$J(\epsilon) = S^{-1}(\epsilon)\mathbf{M}(\epsilon)S(\epsilon) = \begin{pmatrix} J_1(\epsilon) & & \\ & \ddots & \\ & & J_N(\epsilon) \end{pmatrix},$$

where the Jordan blocks $J_i(\epsilon)$ are square matrices of the form

$$J_i(\epsilon) = \begin{pmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i & 1 \end{pmatrix}.$$

The λ_i are the eigenvalues of $\mathbf{M}(\epsilon)$. Plugging $\mathbf{M}(\epsilon) = S(\epsilon)J(\epsilon)S^{-1}(\epsilon)$ into (4.14) yields

$$\vec{f}(x, \epsilon) = S(\epsilon)e^{J(\epsilon)\ln(x)}S^{-1}(\epsilon)\vec{f}_0(\epsilon)$$

with

$$e^{J(\epsilon)\ln(x)} = \begin{pmatrix} e^{J_1(\epsilon)\ln(x)} & & \\ & \ddots & \\ & & e^{J_N(\epsilon)\ln(x)} \end{pmatrix}.$$

The diagonal blocks are given by

$$e^{J_i(\epsilon)\ln(x)} = x^{\lambda_i} \begin{pmatrix} 1 & \frac{\ln^1(x)}{1!} & \frac{\ln^2(x)}{2!} & \cdots & \frac{\ln^{n-1}(x)}{(n-1)!} \\ & 1 & \frac{\ln^1(x)}{1!} & \cdots & \frac{\ln^{n-2}(x)}{(n-2)!} \\ & & \ddots & \ddots & \vdots \\ & & & 1 & \frac{\ln^1(x)}{1!} \\ & & & & 1 \end{pmatrix}.$$

The systems of differential equations fulfilled by Feynman integrals have eigenvalues of the form $\lambda_i = j_i + k_i\epsilon$. Hence, The solution (4.14) is in the correct form (4.11) and the k -values can be directly read-off from the eigenvalues λ_i of $\mathbf{M}(\epsilon)$. The inclusion of higher order terms in (4.13) does not spoil these scaling behaviors.

So far, the consideration was limited to systems where the singularity at $x = 0$ is Fuchsian. In order to find a basis in which this condition is fulfilled, the tool `epsilon` can be used with the option `--fuchsify-at` to reduce a single singularity to Fuchsian form². The matrix elements of the transformation matrix are rational functions in x and ϵ and thus will not change the scaling behavior of the master integrals.

4.2.4 Reduction

The recurrence relations (4.12) can be used to reduce all coefficients in (4.11) up to a desired order to a small set of master coefficients. This reduction procedure is similar to an IBP-reduction of Feynman integrals discussed in Section 2.6. The reduction of Feynman integrals to master integrals was performed with the well-known Laporta algorithm [75]. We will use a similar algorithm for the reduction to master coefficients.

²This feature was not included in the version described in Ref. [1].

The first step in order to use a Laporta-like algorithm is to define an ordering of the coefficients. Hence, we introduce the notation

$$\vec{c}_{kjn}(\epsilon) = \begin{pmatrix} c_{1kjn}(\epsilon) \\ \vdots \\ c_{Nkjn}(\epsilon) \end{pmatrix}.$$

The ordering of the coefficients $c_{ikjn}(\epsilon)$ is very simple. Since the reduction has to be run for all relevant k -values independently, the index k can be ignored. The coefficients are then ordered lexicographically by the remaining indices i , j , and n . This means that a coefficient of the master integral at position i in $\vec{f}(x, \epsilon)$ is considered to be simpler than a coefficient of an integral at position i' if $i < i'$. This is reasonable, since we usually order the master integrals in a way to obtain a block-triangular structure of the system of differential equations. Consequently, a master integral with a lower number of propagators appears first in $\vec{f}(x, \epsilon)$ (cf. Section 3.2). If two coefficients have the same index i , they belong to the same master integral. We order these coefficients first by the exponent of x in (4.11) and then by the exponent of $\ln(x)$, so that lower terms in the asymptotic expansion are considered to be simpler.

In a next step seeds have to be generated. Here we simply generate all combinations of indices up to a given maximum value of j . The upper limit of the index n is given by the number of loops K . The index i takes all values $1, \dots, N$, where N is the number of master integrals in $\vec{f}(x, \epsilon)$. Note that the maximum value of j should be slightly larger than the desired order \hat{j} of the expansion to obtain a full reduction of the coefficients at order \hat{j} .

We implemented the formulas (4.9) to obtain the matrices $M_l(\epsilon)$ of (4.10) from a system of differential equations of the form (4.8) in a C++-tool. Afterwards, this program generates seeds, as described above, and inserts them into the recurrence relations (4.12). The linear system of equations obtained this way is then written to disk in a format understood by the Laporta implementation `Kira` [82], which is used to perform the actual reduction.

4.2.5 Master Coefficients

In this section, we turn to the most complicated part of the differential equation expansion method: the evaluation of the master coefficients. These coefficients cannot be obtained from the differential equation and have to be determined with a different method. One possible strategy is to perform an expansion by regions for every master integral up to an order that covers the required master coefficients. In this thesis, we will discuss a different approach based on MB representations.

As shown in Section 3.6.2, MB representations allow us to extract certain coefficients of an asymptotic expansion directly in terms of lower dimensional MB integrals. These lower dimensional MB integrals do not depend on the kinematic variable anymore and can be evaluated either numerically or, in some cases, even analytically using the residue theorem.

We obtain the MB representations for the master integrals in question using a novel technique described in Section 3.7 and published in Ref. [2]. However, this technique is only applicable for Feynman integrals without numerators. Hence, we express those integrals involving numerators as linear combination of Feynman integrals of the same sector without numerators but with shifted indices and a shifted number of space-time dimensions, as discussed in Section 3.5.1. Since the MB representation obtained by our method is valid for arbitrary indices and an arbitrary number of space-time dimensions, the same MB representation can be used for every term in this linear combination.

Our method yields a MB representation expressed in terms of the kinematic variable τ and of the form

$$I = \int_{c_1} \frac{dz_1}{2\pi i} \cdots \int_{c_N} \frac{dz_N}{2\pi i} (-4\tau)^C \frac{\Gamma(A_1) \cdots \Gamma(A_J)}{\Gamma(B_1) \cdots \Gamma(B_L)}, \quad (4.15)$$

where A_i , B_i and C are linear combinations of the integration variables z_1, \dots, z_N and $\epsilon = (4-d)/2$. The integration contours fulfill the standard prescription of MB representations. In order to directly extract the correct coefficient of an asymptotic expansion in x , we first have to change the kinematic variable in the MB representation. Therefore, we insert (2.9) into $(-4\tau)^C$ and apply formula (3.74) in order to obtain

$$(-4\tau)^C = \frac{1}{\Gamma(-2C)} \int_c \frac{dw}{2\pi i} e^{-i\pi w} x^{w-C} \Gamma(-2C+w)\Gamma(-w). \quad (4.16)$$

Unfortunately, a naïve application of (4.16) often leads to MB representations, where singularities in ϵ cannot be resolved with the strategy explained in (3.6.1). This is particularly true if the constant part of C is a positive integer. Thus, we write

$$C = n + \tilde{C}, \quad n \in \mathbb{N}_0, \quad \tilde{C} \Big|_{z_i=0, \epsilon=0} \leq 0,$$

such that a factor $(-4\tau)^n$ can be pulled out of the integral in (4.15). After that, (4.16) is only applied to $(-4\tau)^{\tilde{C}}$. The factor $(-4\tau)^n$ can be expressed as a Laurent polynomial in x using (2.9). In the end, this leads to an $(N+1)$ -dimensional MB representation,

$$I = \frac{(1-x)^{2n}}{x^n} \int_{c_1} \frac{dz_1}{2\pi i} \cdots \int_{c_{N+1}} \frac{dz_{N+1}}{2\pi i} e^{-i\pi z_{N+1}} x^{z_{N+1}-\tilde{C}} \cdot \frac{\Gamma(-2\tilde{C}+z_{N+1})\Gamma(-z_{N+1})\Gamma(A_1) \cdots \Gamma(A_J)}{\Gamma(-2\tilde{C})\Gamma(B_1) \cdots \Gamma(B_L)}, \quad (4.17)$$

in the kinematic variable x .

The singularities in ϵ of the MB representation (4.17) can be resolved in a further step via Strategy A, described in Section 3.6.1. The individual terms of the result are all of the form (3.76), so that alg. 3.6 can be used to perform an asymptotic expansion up to an order which covers all necessary master coefficients of the integral. The master coefficient c_{ikjn} is then given by the coefficient of $x^{j+k\epsilon} \ln^n(x)$ in terms of MB integrals independent of x . These MB integrals have a lower dimensionality than the original unexpanded integral.

These x independent MB integrals still depend on the dimensional regulator ϵ . Singularities in ϵ have already been resolved before the asymptotic expansion was performed. Hence, the integrands of the MB integrals can immediately be expanded in ϵ . The series coefficients of these ϵ -expansions are MB integrals independent of both x and ϵ , which evaluate to irrational numbers.

A straightforward way to evaluate these integrals numerically is to parametrize the contours and to use the `Cuba`-library [137] to perform a numeric integration. For the checks we performed for this method so far, we restricted ourselves to this simple approach, which is implemented in the function `MBIntegrate` of the `Mathematica`-package `MB` [118]. Nevertheless, we believe that a better accuracy could be reached by more sophisticated integration methods, e.g. those presented in Ref. [114, 138].

In many cases even analytic results can be obtained from the MB integrals. This is done by closing the integration contours either to the left or to the right such that the residue theorem can be applied. The resulting multiple sums can then, hopefully, be summed up via tools like `XSummer` [139] or the algorithms by the Linz group [140–142]. The `Mathematica`-package `MBsums` is able to extract these multiple sums from MB integrals, automatically [143].

Some Results 5

In this thesis, we found a method to compute a huge subset of the master integrals for $H \rightarrow \gamma\gamma$, i.e. those expressible in GPLs of argument x (cf. Section 4.1). We refrain from giving all results we obtained, since the results are usually very long in printed form.

Instead, we present the results of a few select examples, which are nonetheless highly non-trivial. Even though there is no fundamental problem in obtaining higher order terms, once more for brevity, we truncate the ϵ -expansion after the zeroth order.

Since the method for master integrals that cannot be expressed in terms of GPLs is still work in progress, we are not able to give any results for this type at the moment.

The master integrals for which we present results at the end of this chapter are depicted in fig. 5.1. To the best of our knowledge, these results do not appear elsewhere in the literature. The expressions contain GPLs with indices $-1, 0, 1, r, r^*, \varphi^2, \varphi^{-2}$, where

$$r = (-1)^{1/3} = \frac{1 + i\sqrt{3}}{2},$$

$$\varphi = \frac{1 + \sqrt{5}}{2}.$$

Note that r and r^* both are sixth roots of unity and φ is the golden ratio. Since we evaluated the boundary conditions around the point $x_0 = 1$, the results contain GPLs at argument 1. The special case of GPLs with indices from the set $-1, 0, 1$ (i.e. harmonic polylogarithms) evaluates at argument 1 to multiple zeta values. For these GPLs, we used the tables of Ref. [99]. GPLs at argument 1 with other indices are left unevaluated¹.

We checked the results presented in this chapter against independent numerical results obtained by the sector decomposition tool FIESTA [145] and found perfect agreement. The checks were performed at an Euclidean kinematic point $x = 1/3$, which translates to $\tau = -1/3$ via (2.9). For physical kinematic points, the GPLs in the result have to be analytically continued as described in Section 3.1.3.

¹In principle, it is also possible to analytically evaluate GPLs at argument 1 of which the indices are sixth roots of unity or zero (see Ref. [144]).

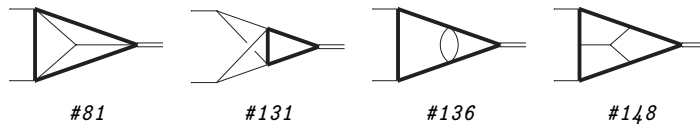


Fig. 5.1: Sample master integrals. The number is used internally to identify the master integral.

The results up to order ϵ^0 are given by:

$$\begin{aligned} \triangle &= \iiint_{l_1 l_2 l_3} \frac{1}{[l_1^2 - 1][(l_2 - q_1 - q_2)^2 - 1][(l_3 + q_2)^2 - 1]} \\ &\quad \cdot \frac{1}{[(l_1 - l_3 - q_1 - q_2)^2][(l_1 - l_2)^2][(l_2 - l_3 - q_1 - q_2)^2]} \\ &= \sum_{n=-1}^0 \epsilon^n F_n^{81} + \mathcal{O}(\epsilon), \end{aligned}$$

where

$$\begin{aligned} F_{-1}^{81} &= 2\zeta(3), \\ F_0^{81} &= -\frac{1-10x-x^2}{30-60x+30x^2}\pi^4 + \frac{6-10x+10x^2}{1-2x+x^2}\zeta(3)G(0|x) - \frac{4-18x+6x^2}{1-2x+x^2} \\ &\quad G^2(0|x)G(0,1|x) - \frac{x+10x^2}{24-48x+24x^2}G^4(0|x) - \frac{8-82x+44x^2}{1-2x+x^2}G(0,0,0,1|x) \\ &\quad + \frac{8-58x+24x^2}{1-2x+x^2}G(0,0,1|x)G(0|x) + \frac{16x}{1-x} \left[-G(0,0,-1|x)G(0|x) \right. \\ &\quad \left. + 3G(0,0,0,-1|x) \right] + \frac{1}{30} \left[-3\pi^4 - 60\pi^2 G(0|x)G(1|x) + 60\pi^2 G(0,1|x) \right. \\ &\quad \left. - 480G(0,-1|x)G(0|x)G(1|x) + 240G(0,1|x)G(0|x)G(1|x) + 360G(0,0,1,1|x) \right. \\ &\quad \left. + 480G(0,-1,1|x)G(0|x) + 960G(0,0,-1|x)G(1|x) - 360G(0,0,1|x)G(1|x) \right. \\ &\quad \left. + 480G(0,1,-1|x)G(0|x) - 360G(0,1,1|x)G(0|x) - 960G(0,0,-1,1|x) \right. \\ &\quad \left. - 960G(0,0,1,-1|x) - 480G(0,1,0,-1|x) + 40G^3(0|x)G(1|x) + 60G^2(0,1|x) \right. \\ &\quad \left. - 30G^2(0|x)G^2(1|x) + 360\zeta(3) - 1080\zeta(3)G(1|x) \right] + \frac{1-3x+x^2}{6-12x+6x^2} \left[\right. \\ &\quad \left. - 60G(0,1|x)G(0|x)G(\varphi^2|x) - 60G(0,1|x)G(0|x)G(\varphi^{-2}|x) + 12\zeta(3)G(\varphi^2|1) \right. \\ &\quad \left. + 60G(0,0,1|x)G(\varphi^2|x) + 60G(0,0,1|x)G(\varphi^{-2}|x) - 30G(0,0,\varphi^2|x)G(0|x) \right. \\ &\quad \left. - 30G(0,0,\varphi^{-2}|x)G(0|x) + 60G(0,1,\varphi^2|x)G(0|x) + 60G(0,1,\varphi^{-2}|x)G(0|x) \right. \\ &\quad \left. - 30G(0,0,0,\varphi^2|1) + 30G(0,0,0,\varphi^2|x) - 30G(0,0,0,\varphi^{-2}|1) + 48\zeta(3)G(\varphi^2|x) \right. \\ &\quad \left. + 30G(0,0,0,\varphi^{-2}|x) + 60G(0,0,1,\varphi^2|1) - 60G(0,0,1,\varphi^2|x) + 12\zeta(3)G(\varphi^{-2}|1) \right. \\ &\quad \left. + 60G(0,0,1,\varphi^{-2}|1) - 60G(0,0,1,\varphi^{-2}|x) + 15G^2(0|x)G(0,\varphi^2|x) \right. \\ &\quad \left. + 15G^2(0|x)G(0,\varphi^{-2}|x) + 30G^2(0|x)G(\varphi^2,1|x) + 30G^2(0|x)G(\varphi^{-2},1|x) \right. \\ &\quad \left. - 5G^3(0|x)G(\varphi^2|x) - 5G^3(0|x)G(\varphi^{-2}|x) + 48\zeta(3)G(\varphi^{-2}|x) \right] + \frac{1-x+x^2}{1-2x+x^2} \left[\right. \\ &\quad \left. \pi^2 G(0|x)G(r|x) + \pi^2 G(0|x)G(r^*|x) - \pi^2 G(0,r|x) - 6G(0,1|x)G(0,r|x) \right. \\ &\quad \left. - \pi^2 G(0,r^*|x) - 6G(0,1|x)G(0,r^*|x) + 6G(0,0,1|x)G(r|x) + 6G(0,0,0,r|1) \right. \\ &\quad \left. + 6G(0,0,1|x)G(r^*|x) + 6G(0,0,r|x)G(0|x) + 6G(0,0,r^*|x)G(0|x) \right. \\ &\quad \left. + 6G(0|x)G(0,r,1|x) + 6G(0|x)G(0,r^*,1|x) - 6G(0,0,0,r|x) + 6G(0,0,0,r^*|1) \right. \\ &\quad \left. - 6G(0,0,0,r^*|x) - 6G(0,0,1,r|1) + 6G(0,0,1,r|x) - 6G(0,0,1,r^*|1) \right. \\ &\quad \left. + 6G(0,0,1,r^*|x) - 6G(0,1,0,r|1) + 6G(0,1,0,r|x) - 6G(0,1,0,r^*|1) \right. \\ &\quad \left. + 6G(0,1,0,r^*|x) - 3G^2(0|x)G(0,r|x) - 3G^2(0|x)G(0,r^*|x) + G^3(0|x)G(r|x) \right. \\ &\quad \left. - 3G^2(0|x)G(r,1|x) - 3G^2(0|x)G(r^*,1|x) + G^3(0|x)G(r^*|x) + 4\zeta(3)G(r|1) \right. \\ &\quad \left. + 2\zeta(3)G(r|x) + 4\zeta(3)G(r^*|1) + 2\zeta(3)G(r^*|x) \right] + \frac{x}{120-240x+120x^2} \left[G^5(0|x) \right. \\ &\quad \left. - 32\pi^4 G(0|x) + 80\pi^4 G(1|x) + 240\pi^2 G(0|x)G(1|x)G(r|x) - 240\pi^2 G(0,0,1|x) \right. \end{aligned}$$

$$\begin{aligned}
& + 240\pi^2 G(0|x) G(1|x) G(r^*|x) + 240\pi^2 G(0, 1|x) G(0|x) - 120\pi^2 G(0|x) G(0, r|x) \\
& + 2400G(0, 1|x) G(0|x) G(1|x) G(\varphi^2|x) - 1200G(0, 1|x) G(0|x) G(0, \varphi^2|x) \\
& + 2400G(0, 1|x) G(0|x) G(1|x) G(\varphi^{-2}|x) - 1200G(0, 1|x) G(0|x) G(0, \varphi^{-2}|x) \\
& - 240\pi^2 G(0, r|x) G(1|x) - 1440G(0, 1|x) G(0, r|x) G(1|x) - 240\pi^2 G(0, 0, r|1) \\
& - 120\pi^2 G(0|x) G(0, r^*|x) - 240\pi^2 G(0, r^*|x) G(1|x) - 240\pi^2 G(0|x) G(r, 1|x) \\
& - 1440G(0, 1|x) G(0, r^*|x) G(1|x) - 2400G(0, 1|x) G(0|x) G(\varphi^2, 1|x) + 3840\zeta(5) \\
& - 2400G(0, 1|x) G(0|x) G(\varphi^{-2}, 1|x) - 240\pi^2 G(0|x) G(r^*, 1|x) - 60\pi^2 G^2(0|x) \\
& - 6240G(0, 0, 1|x) G(0|x) G(1|x) - 2400G(0, 0, 1|x) G(1|x) G(\varphi^2|x) \\
& - 2400G(0, 0, 1|x) G(1|x) G(\varphi^{-2}|x) + 1440G(0, 0, 1|x) G(1|x) G(r|x) \\
& + 1440G(0, 0, 1|x) G(1|x) G(r^*|x) - 960G(0, 0, 1|x) G(0, 1|x) + 240\pi^2 G(0, 0, r|x) \\
& + 1200G(0, 0, 1|x) G(0, \varphi^2|x) + 1200G(0, 0, 1|x) G(0, \varphi^{-2}|x) + 240\pi^2 G(0, r, 1|x) \\
& - 720G(0, 0, 1|x) G(0, r|x) - 720G(0, 0, 1|x) G(0, r^*|x) - 240\pi^2 G(0, 0, r^*|1) \\
& + 2400G(0, 0, 1|x) G(\varphi^2, 1|x) + 2400G(0, 0, 1|x) G(\varphi^{-2}, 1|x) + 20\pi^2 G^3(0|x) \\
& - 1440G(0, 0, 1|x) G(r, 1|x) - 1440G(0, 0, 1|x) G(r^*, 1|x) + 240\pi^2 G(0, 0, r^*|x) \\
& + 1200G(0, 0, \varphi^2|x) G(0|x) G(1|x) + 1200G(0, 0, \varphi^{-2}|x) G(0|x) G(1|x) \\
& + 1440G(0, 0, r|x) G(0|x) G(1|x) + 1440G(0, 0, r^*|x) G(0|x) G(1|x) - 1520\pi^2 \zeta(3) \\
& - 2400G(0, 1, \varphi^2|x) G(0|x) G(1|x) - 2400G(0, 1, \varphi^{-2}|x) G(0|x) G(1|x) \\
& + 1440G(0|x) G(0, r, 1|x) G(1|x) + 1440G(0, 1|x) G(0, r, 1|x) - 480\zeta(3)G(0, r|1) \\
& + 240\pi^2 G(0, r^*, 1|x) + 1440G(0|x) G(0, r^*, 1|x) G(1|x) - 7680G(0, 0, 0, -1|x) \\
& + 1440G(0, 1|x) G(0, r^*, 1|x) + 1920G(0, 0, 0, -1|x) G(0|x) - 5760G(0, 0, 0, r, 1|1) \\
& + 10080G(0, 0, 0, 1|x) G(0|x) + 7200G(0, 0, 0, 1|x) G(1|x) - 18000G(0, 0, 0, 0, 1|x) \\
& - 600G(0, 0, 0, \varphi^2|1) G(0|x) + 1200G(0, 0, 0, \varphi^2|1) G(1|x) - 2880G(0, 0, 0, 0, r|1) \\
& - 1800G(0, 0, 0, \varphi^2|x) G(0|x) - 1200G(0, 0, 0, \varphi^2|x) G(1|x) - 480G(0, 0, 1, 0, 1|x) \\
& - 600G(0, 0, 0, \varphi^{-2}|1) G(0|x) + 1200G(0, 0, 0, \varphi^{-2}|1) G(1|x) - 240\zeta(3)G(0, r|x) \\
& - 1800G(0, 0, 0, \varphi^{-2}|x) G(0|x) - 1200G(0, 0, 0, \varphi^{-2}|x) G(1|x) - 120\zeta(3)G^2(0|x) \\
& - 720G(0, 0, 0, r|1) G(0|x) + 1440G(0, 0, 0, r|1) G(1|x) + 720G(0, 0, 1, r|1) G(0|x) \\
& - 2160G(0, 0, 0, r|x) G(0|x) - 1440G(0, 0, 0, r|x) G(1|x) - 2400G(0, 0, 0, \varphi^2|1) \\
& - 720G(0, 0, 0, r^*|1) G(0|x) + 1440G(0, 0, 0, r^*|1) G(1|x) + 2400G(0, 0, 0, 0, \varphi^2|x) \\
& - 2160G(0, 0, 0, r^*|x) G(0|x) - 1440G(0, 0, 0, r^*|x) G(1|x) + 2880G(0, 0, 0, 0, r|x) \\
& + 7680G(0, 0, 1, 1|x) G(0|x) + 1200G(0, 0, 1, \varphi^2|1) G(0|x) - 2880G(0, 0, 0, 0, r^*|1) \\
& - 2400G(0, 0, 1, \varphi^2|1) G(1|x) + 2400G(0, 0, 1, \varphi^2|x) G(0|x) - 720G(0, 0, 1, 0, r|1) \\
& + 2400G(0, 0, 1, \varphi^2|x) G(1|x) + 1200G(0, 0, 1, \varphi^{-2}|1) G(0|x) - 1440\zeta(3)G(0, 1|x) \\
& - 2400G(0, 0, 1, \varphi^{-2}|1) G(1|x) + 2400G(0, 0, 1, \varphi^{-2}|x) G(0|x) - 960\zeta(3)G(r, 1|1) \\
& + 2400G(0, 0, 1, \varphi^{-2}|x) G(1|x) - 1440G(0, 0, 1, r|1) G(1|x) + 720G(0, 0, 1, 0, r|x) \\
& + 1440G(0, 0, 1, r|x) G(1|x) + 720G(0, 0, 1, r^*|1) G(0|x) + 2880G(0, 0, 0, 0, r^*|x) \\
& - 1440G(0, 0, 1, r^*|1) G(1|x) + 1440G(0, 0, 1, r^*|x) G(1|x) - 8640G(0, 0, 0, 1, 1|x) \\
& - 1200G(0, 0, \varphi^2, 1|x) G(0|x) - 1200G(0, 0, \varphi^{-2}, 1|x) G(0|x) - 480\zeta(3)G(0, r^*|1) \\
& - 2880G(0, 0, r, 1|x) G(0|x) - 2880G(0, 0, r^*, 1|x) G(0|x) + 3600G(0, 0, 0, 1, \varphi^2|1) \\
& + 1200G(0, 1, 0, \varphi^2|x) G(0|x) + 1200G(0, 1, 0, \varphi^{-2}|x) G(0|x) - 480\zeta(3)G(r, 1|x)
\end{aligned}$$

$$\begin{aligned}
& + 720G(0, 1, 0, r|1)G(0|x) - 1440G(0, 1, 0, r|1)G(1|x) - 2400G(0, 0, 0, 0, \varphi^{-2}|1) \\
& + 1440G(0, 1, 0, r|x)G(1|x) + 720G(0, 1, 0, r^*|1)G(0|x) - 3600G(0, 0, 0, 1, \varphi^2|x) \\
& - 1440G(0, 1, 0, r^*|1)G(1|x) + 1440G(0, 1, 0, r^*|x)G(1|x) - 2160G(0, 0, 0, 1, r|1) \\
& + 2400G(0, 1, \varphi^2, 1|x)G(0|x) + 2400G(0, 1, \varphi^{-2}, 1|x)G(0|x) - 960\zeta(3)G(r^*, 1|1) \\
& - 2880G(0|x)G(0, r, 1, 1|x) - 2880G(0|x)G(0, r^*, 1, 1|x) + 2160G(0, 0, 0, 1, r|x) \\
& + 2400G(0, 0, 0, 0, \varphi^{-2}|x) + 3600G(0, 0, 0, 1, \varphi^{-2}|1) - 3600G(0, 0, 0, 1, \varphi^{-2}|x) \\
& - 2160G(0, 0, 0, 1, r^*|1) + 2160G(0, 0, 0, 1, r^*|x) - 1200G(0, 0, 0, \varphi^2, 1|1) \\
& + 1200G(0, 0, 0, \varphi^2, 1|x) - 1200G(0, 0, 0, \varphi^{-2}, 1|1) + 1200G(0, 0, 0, \varphi^{-2}, 1|x) \\
& + 5760G(0, 0, 0, r, 1|x) - 5760G(0, 0, 0, r^*, 1|1) + 5760G(0, 0, 0, r^*, 1|x) \\
& + 1200G(0, 0, 1, 0, \varphi^2|1) - 1200G(0, 0, 1, 0, \varphi^2|x) + 1200G(0, 0, 1, 0, \varphi^{-2}|1) \\
& - 1200G(0, 0, 1, 0, \varphi^{-2}|x) - 720G(0, 0, 1, 0, r^*|1) + 720G(0, 0, 1, 0, r^*|x) \\
& + 2400G(0, 0, 1, \varphi^2, 1|1) - 2400G(0, 0, 1, \varphi^2, 1|x) + 2400G(0, 0, 1, \varphi^{-2}, 1|1) \\
& - 2400G(0, 0, 1, \varphi^{-2}, 1|x) + 1440G(0, 0, 1, r, 1|1) - 1440G(0, 0, 1, r, 1|x) \\
& + 1440G(0, 0, 1, r^*, 1|1) - 1440G(0, 0, 1, r^*, 1|x) - 1440G(0, 0, r, 0, 1|1) \\
& + 1440G(0, 0, r, 0, 1|x) - 1440G(0, 0, r^*, 0, 1|1) + 1440G(0, 0, r^*, 0, 1|x) \\
& + 1440G(0, 1, 0, r, 1|1) - 1440G(0, 1, 0, r, 1|x) + 1440G(0, 1, 0, r^*, 1|1) \\
& - 1440G(0, 1, 0, r^*, 1|x) - 120\pi^2G^2(0|x)G(1|x) + 1920G^2(0|x)G(0, 1|x)G(1|x) \\
& - 600G^2(0|x)G(0, \varphi^2|x)G(1|x) - 600G^2(0|x)G(0, \varphi^{-2}|x)G(1|x) \\
& - 720G^2(0|x)G(0, r|x)G(1|x) - 720G^2(0|x)G(0, r^*|x)G(1|x) \\
& - 1200G^2(0|x)G(1|x)G(\varphi^2, 1|x) - 1200G^2(0|x)G(1|x)G(\varphi^{-2}, 1|x) \\
& - 720G^2(0|x)G(1|x)G(r, 1|x) - 720G^2(0|x)G(1|x)G(r^*, 1|x) \\
& - 2520G^2(0|x)G(0, 0, 1|x) + 600G^2(0|x)G(0, 0, \varphi^2|x) + 720G^2(0|x)G(0, 0, r|x) \\
& + 600G^2(0|x)G(0, 0, \varphi^{-2}|x) + 720G^2(0|x)G(0, 0, r^*|x) + 440G^3(0|x)G(0, 1|x) \\
& - 3840G^2(0|x)G(0, 1, 1|x) + 1200G^2(0|x)G(0, \varphi^2, 1|x) - 100G^3(0|x)G(0, \varphi^2|x) \\
& + 1200G^2(0|x)G(0, \varphi^{-2}, 1|x) + 1080G^2(0|x)G(0, r, 1|x) - 120G^3(0|x)G(0, r|x) \\
& + 1080G^2(0|x)G(0, r^*, 1|x) + 2400G^2(0|x)G(\varphi^2, 1, 1|x) - 120G^3(0|x)G(0, r^*|x) \\
& + 2400G^2(0|x)G(\varphi^{-2}, 1, 1|x) + 1440G^2(0|x)G(r, 1, 1|x) - 240G^3(0|x)G(r, 1|x) \\
& + 1440G^2(0|x)G(r^*, 1, 1|x) + 200G^3(0|x)G(1|x)G(\varphi^2|x) - 110G^4(0|x)G(1|x) \\
& + 200G^3(0|x)G(1|x)G(\varphi^{-2}|x) + 240G^3(0|x)G(1|x)G(r|x) - 240\zeta(3)G(0, r^*|x) \\
& + 240G^3(0|x)G(1|x)G(r^*|x) - 100G^3(0|x)G(0, \varphi^{-2}|x) - 240G^3(0|x)G(r^*, 1|x) \\
& - 200G^3(0|x)G(\varphi^2, 1|x) - 200G^3(0|x)G(\varphi^{-2}, 1|x) + 1200G^2(0, 1|x)G(0|x) \\
& + 1440\zeta(3)G(0|x)G(1|x) + 240\zeta(3)G(0|x)G(\varphi^2|1) - 480\zeta(3)G(1|x)G(\varphi^2|1) \\
& - 1920\zeta(3)G(1|x)G(\varphi^2|x) + 240\zeta(3)G(0|x)G(\varphi^{-2}|1) - 480\zeta(3)G(0|x)G(r|1) \\
& - 480\zeta(3)G(1|x)G(\varphi^{-2}|1) - 1920\zeta(3)G(1|x)G(\varphi^{-2}|x) + 960\zeta(3)G(1|x)G(r|1) \\
& + 480\zeta(3)G(1|x)G(r|x) - 480\zeta(3)G(0|x)G(r^*|1) + 960\zeta(3)G(1|x)G(r^*|1) \\
& + 480\zeta(3)G(1|x)G(r^*|x) + 240\zeta(3)G(0, \varphi^2|1) + 960\zeta(3)G(0, \varphi^2|x) \\
& + 240\zeta(3)G(0, \varphi^{-2}|1) + 960\zeta(3)G(0, \varphi^{-2}|x) + 480\zeta(3)G(\varphi^2, 1|1) \\
& + 1920\zeta(3)G(\varphi^2, 1|x) + 480\zeta(3)G(\varphi^{-2}, 1|1) + 1920\zeta(3)G(\varphi^{-2}, 1|x) \\
& - 480\zeta(3)G(r^*, 1|x) \Big].
\end{aligned}$$

$$\begin{aligned}
\text{Diagram} &= \iiint_{l_1 l_2 l_3} \frac{1}{[l_1^2][l_3^2 - 1][(l_3 - q_1 - q_2)^2 - 1][(l_1 - q_1)^2]} \\
&\quad \cdot \frac{1}{[(l_1 - l_2)^2][(l_1 - l_2 + q_2)^2][(l_2 - l_3)^2 - 1]} \\
&= \sum_{n=-2}^0 \epsilon^n F_n^{131} + \mathcal{O}(\epsilon),
\end{aligned}$$

where

$$\begin{aligned}
F_{-2}^{131} &= -\frac{x^2}{1-3x+3x^2-x^3} G^3(0|x) + \frac{3x}{1-2x+x^2} \left[2G(0,1|x)G(0|x) - 2G(0,0,1|x) \right. \\
&\quad \left. - G^2(0|x)G(1|x) - 2\zeta(3) \right], \\
F_{-1}^{131} &= -\frac{11x-41x^2}{90-270x+270x^2-90x^3} \pi^4 - \frac{18x-56x^2}{1-3x+3x^2-x^3} \zeta(3)G(0|x) \\
&\quad - \frac{6x-110x^2}{1-3x+3x^2-x^3} G(0,0,1|x)G(0|x) - \frac{12x+168x^2}{1-3x+3x^2-x^3} G(0,0,0,1|x) \\
&\quad - \frac{6x+234x^2}{1-3x+3x^2-x^3} G(0,0,0,-1|x) + \frac{12x-34x^2}{1-3x+3x^2-x^3} G^2(0|x)G(0,1|x) \\
&\quad + \frac{6x+82x^2}{1-3x+3x^2-x^3} G(0,0,-1|x)G(0|x) + \frac{x+x^2}{1-3x+3x^2-x^3} \left[G^3(0|x)G(-1|x) \right. \\
&\quad \left. - 3G^2(0|x)G(0,-1|x) \right] + \frac{x^2}{12-36x+36x^2-12x^3} \left[22\pi^2 G^2(0|x) - 36G^3(0|x) \right. \\
&\quad \left. - 15G^4(0|x) \right] + \frac{x}{3-6x+3x^2} \left[11\pi^2 G(0|x)G(1|x) - 11\pi^2 G(0,1|x) - 54\zeta(3) \right. \\
&\quad + 228G(0,-1|x)G(0|x)G(1|x) + 54G(0,1|x)G(0|x) - 228G(0,-1,1|x)G(0|x) \\
&\quad + 72G(0,1|x)G(0|x)G(1|x) - 456G(0,0,-1|x)G(1|x) - 54G(0,0,1|x) \\
&\quad - 120G(0,0,1|x)G(1|x) - 228G(0,1,-1|x)G(0|x) - 120G(0,1,1|x)G(0|x) \\
&\quad + 456G(0,0,-1,1|x) + 456G(0,0,1,-1|x) + 120G(0,0,1,1|x) + 24G^2(0,1|x) \\
&\quad + 228G(0,1,0,-1|x) - 27G^2(0|x)G(1|x) - 21G^3(0|x)G(1|x) + 222\zeta(3)G(1|x) \\
&\quad \left. - 6G^2(0|x)G^2(1|x) \right], \\
F_0^{131} &= -\frac{46x-100x^2}{1-3x+3x^2-x^3} \zeta(5) - \frac{11x-41x^2}{30-90x+90x^2-30x^3} \pi^4 + \frac{15x-34x^2}{3-9x+9x^2-3x^3} \pi^2 \zeta(3) \\
&\quad - \frac{66x-289x^2}{180-540x+540x^2-180x^3} \pi^4 G(0|x) - \frac{54x-168x^2}{1-3x+3x^2-x^3} \zeta(3)G(0|x) \\
&\quad - \frac{146x-278x^2}{1-3x+3x^2-x^3} \zeta(3)G(0,1|x) - \frac{27x-119x^2}{1-3x+3x^2-x^3} \zeta(3)G^2(0|x) \\
&\quad - \frac{38x-56x^2}{3-9x+9x^2-3x^3} \pi^2 G(0,1|x)G(0|x) - \frac{11x+65x^2}{3-9x+9x^2-3x^3} \pi^2 G(0,0,-1|x) \\
&\quad + \frac{5x-81x^2}{3-9x+9x^2-3x^3} \pi^2 G(0,0,1|x) - \frac{18x-330x^2}{1-3x+3x^2-x^3} G(0,0,1|x)G(0|x) \\
&\quad - \frac{36x+504x^2}{1-3x+3x^2-x^3} G(0,0,0,1|x) - \frac{18x+702x^2}{1-3x+3x^2-x^3} G(0,0,0,-1|x) \\
&\quad + \frac{36x-102x^2}{1-3x+3x^2-x^3} G^2(0|x)G(0,1|x) + \frac{18x+246x^2}{1-3x+3x^2-x^3} G(0,0,-1|x)G(0|x)
\end{aligned}$$

$$\begin{aligned}
& - \frac{220x - 2380x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, 0, 1, 1|x) - \frac{42x - 1518x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, 0, 0, -1|x) \\
& - \frac{64x - 1192x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, 1, 0, -1|x) - \frac{180x - 1188x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, -1, 0, 1|x) \\
& - \frac{44x - 764x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, 1, 0, 1|x) - \frac{3x - 291x^2}{1 - 3x + 3x^2 - x^3} G^2(0|x) G(0, 0, -1|x) \\
& - \frac{186x - 130x^2}{1 - 3x + 3x^2 - x^3} G^2(0|x) G(0, 1, 1|x) - \frac{2x + 6x^2}{1 - 3x + 3x^2 - x^3} G^3(0|x) G(0, -1|x) \\
& - \frac{16x + 368x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, 1|x) G(0, 1|x) - \frac{42x + 950x^2}{1 - 3x + 3x^2 - x^3} \\
& G(0, 0, 0, 1|x) G(0|x) + \frac{24x - 1224x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, 0, -1|x) G(0|x) \\
& + \frac{196x - 884x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, -1|x) G(0, 1|x) + \frac{12x - 780x^2}{1 - 3x + 3x^2 - x^3} \\
& G(0, 0, -1, -1|x) G(0|x) + \frac{252x - 588x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, 1, 1|x) G(0|x) \\
& + \frac{24x - 116x^2}{3 - 9x + 9x^2 - 3x^3} G^3(0|x) G(0, 1|x) + \frac{12x + 266x^2}{1 - 3x + 3x^2 - x^3} G^2(0|x) G(0, 0, 1|x) \\
& + \frac{228x + 1020x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, -1, 0, -1|x) + \frac{6x + 1362x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, 0, 0, 1|x) \\
& + \frac{444x + 2820x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, 0, -1, -1|x) + \frac{372x - 3396x^2}{1 - 3x + 3x^2 - x^3} \left[-G(0, 0, 0, -1, 1|x) \right. \\
& \left. - G(0, 0, 0, 1, -1|x) \right] + \frac{468x - 788x^2}{1 - 3x + 3x^2 - x^3} \left[G(0, 0, -1, 1|x) G(0|x) \right. \\
& \left. + G(0, 0, 1, -1|x) G(0|x) \right] + \frac{292x - 248x^2}{1 - 3x + 3x^2 - x^3} \left[-G^2(0|x) G(0, -1, 1|x) \right. \\
& \left. - G^2(0|x) G(0, 1, -1|x) \right] + \frac{x^2}{180 - 540x + 540x^2 - 180x^3} \left[990\pi^2 G^2(0|x) \right. \\
& \left. - 3780G^3(0|x) + 490\pi^2 G^3(0|x) - 675G^4(0|x) - 171G^5(0|x) \right] \\
& + \frac{x + x^2}{12 - 36x + 36x^2 - 12x^3} \left[-4\pi^4 G(-1|x) + 44\pi^2 G(0, -1|x) G(0|x) \right. \\
& + 720G(0, -1|x) G(0, 1|x) G(0|x) - 1056G(0, 0, -1|x) G(0|x) G(-1|x) \\
& - 1824G(0, 0, -1|x) G(0, -1|x) - 1248G(0, 0, 1|x) G(0|x) G(-1|x) \\
& - 912G(0, 0, 1|x) G(0, -1|x) + 2880G(0, 0, 0, -1|x) G(-1|x) \\
& + 2160G(0, 0, 0, 1|x) G(-1|x) + 1344G(0, 1, 0, -1|x) G(0|x) \\
& - 22\pi^2 G^2(0|x) G(-1|x) - 108G^2(0|x) G(0, -1|x) - 72G^2(0|x) G(0, -1, -1|x) \\
& + 72G^2(0|x) G(0, -1|x) G(-1|x) + 264G^2(0|x) G(0, 1|x) G(-1|x) \\
& + 36G^3(0|x) G(-1|x) - 12G^2(-1|x) G^3(0|x) + 15G^4(0|x) G(-1|x) \\
& + 456G^2(0, -1|x) G(0|x) + 744G^2(0, 1|x) G(0|x) - 456\zeta(3)G(0|x) G(-1|x) \\
& \left. + 456\zeta(3)G(0, -1|x) \right] + \frac{x}{36 - 72x + 36x^2} \left[95\pi^4 G(1|x) + 396\pi^2 G(0|x) G(1|x) \right. \\
& - 648\pi^2 G(0, -1|x) G(1|x) + 8208G(0, -1|x) G(0|x) G(1|x) - 396\pi^2 G(0, 1|x) \\
& + 4536G(0, 1|x) G(0|x) + 96\pi^2 G(0, 1|x) G(1|x) + 2592G(0, 1|x) G(0|x) G(1|x) \\
& + 10368G(0, -1|x) G(0, 1|x) G(1|x) - 28512G(0, -1, -1|x) G(0|x) G(1|x) \\
& \left. + 648\pi^2 G(0, -1, 1|x) - 8208G(0, -1, 1|x) G(0|x) - 10368G(0, -1, 1|x) G(0, 1|x) \right]
\end{aligned}$$

$$\begin{aligned}
& - 6336G(0, -1, 1|x)G(0|x)G(1|x) - 28512G(0, -1, 1|x)G(0, -1|x) - 4536\zeta(3) \\
& - 16416G(0, 0, -1|x)G(1|x) - 17424G(0, 0, -1|x)G(0|x)G(1|x) \\
& - 4536G(0, 0, 1|x) - 4320G(0, 0, 1|x)G(1|x) - 6912G(0, 0, 1|x)G(0|x)G(1|x) \\
& + 648\pi^2G(0, 1, -1|x) - 8208G(0, 1, -1|x)G(0|x) - 10368G(0, 1, -1|x)G(0, 1|x) \\
& - 6336G(0, 1, -1|x)G(0|x)G(1|x) + 384\pi^2G(0, 1, 1|x) + 16416G(0, 0, -1, 1|x) \\
& - 4320G(0, 1, 1|x)G(0|x) - 7488G(0, 1, 1|x)G(0|x)G(1|x) + 4320G(0, 0, 1, 1|x) \\
& - 6048G(0, 1, 1|x)G(0, 1|x) + 28512G(0, -1, -1, 1|x)G(0|x) + 864G^2(0, 1|x) \\
& + 28512G(0, -1, 1, -1|x)G(0|x) + 16704G(0, -1, 1, 1|x)G(0|x) \\
& - 8064G(0, 0, -1, 1|x)G(1|x) - 6048G(0, 0, 0, -1|x)G(1|x) + 7992\zeta(3)G(1|x) \\
& + 3168G(0, 0, 0, 1|x)G(1|x) + 16416G(0, 0, 1, -1|x) + 5184G(0, 0, 1, 1|x)G(1|x) \\
& - 8064G(0, 0, 1, -1|x)G(1|x) + 28512G(0, 1, -1, -1|x)G(0|x) \\
& + 16704G(0, 1, -1, 1|x)G(0|x) + 8208G(0, 1, 0, -1|x) + 8064G(0, 0, -1, 1, 1|x) \\
& - 4032G(0, 1, 0, -1|x)G(1|x) + 16704G(0, 1, 1, -1|x)G(0|x) - 5616\zeta(3)G^2(1|x) \\
& + 13536G(0, 1, 1, 1|x)G(0|x) + 28512G(0, -1, 0, -1, 1|x) + 8352G(0, 1, 0, 1, 1|x) \\
& + 57024G(0, 0, -1, -1, 1|x) - 57024G(0, 0, 1, -1, -1|x) + 8064G(0, 0, 1, -1, 1|x) \\
& + 8064G(0, 0, 1, 1, -1|x) + 11520G(0, 0, 1, 1, 1|x) - 28512G(0, 1, -1, 0, -1|x) \\
& - 28512G(0, 1, 0, -1, -1|x) + 4032G(0, 1, 0, -1, 1|x) + 4032G(0, 1, 0, 1, -1|x) \\
& - 16704G(0, 1, 1, 0, -1|x) - 2268G^2(0|x)G(1|x) + 426\pi^2G^2(0|x)G(1|x) \\
& + 9720G^2(0|x)G(0, -1|x)G(1|x) + 2520G^2(0|x)G(0, 1|x)G(1|x) \\
& - 756G^3(0|x)G(1|x) - 261G^4(0|x)G(1|x) - 288\pi^2G^2(1|x)G(0|x) \\
& - 2016G^2(1|x)G(0, -1|x)G(0|x) + 1872G^2(1|x)G(0, 1|x)G(0|x) \\
& + 4032G^2(1|x)G(0, 0, -1|x) - 2592G^2(1|x)G(0, 0, 1|x) - 216G^2(0|x)G^2(1|x) \\
& + 204G^2(1|x)G^3(0|x) - 192G^2(0|x)G^3(1|x) + 14256G^2(0, -1|x)G(1|x) \\
& + 1872G^2(0, 1|x)G(1|x) + 13248\zeta(3)G(0|x)G(1|x) \Big].
\end{aligned}$$

$$\begin{aligned}
\triangle &= \iiint_{l_1 l_2 l_3} \frac{1}{[l_1^2 - 1][l_3^2 - 1][(l_3 - q_1 - q_2)^2 - 1][(l_1 - q_1 - q_2)^2 - 1]} \\
&\quad \cdot \frac{1}{[(l_1 - q_1)^2 - 1][(l_1 - l_2)^2][(l_2 - l_3)^2]} \\
&= \sum_{n=-2}^0 \epsilon^n F_n^{136} + \mathcal{O}(\epsilon),
\end{aligned}$$

where

$$\begin{aligned}
F_{-2}^{136} &= -\frac{x}{4 - 8x + 4x^2}G^2(0|x), \\
F_{-1}^{136} &= -\frac{7x + 5x^2}{12 - 36x + 36x^2 - 12x^3}G^3(0|x) + \frac{x}{12 - 24x + 12x^2} \left[\pi^2G(0|x) - 24G(0, 0, -1|x) \right. \\
&\quad \left. + 12G(0, -1|x)G(0|x) - 6G^2(0|x) + 18\zeta(3) \right], \\
F_0^{136} &= -\frac{23x + 25x^2}{720 - 2160x + 2160x^2 - 720x^3}\pi^4 + \frac{2x^2}{1 - 3x + 3x^2 - x^3}\zeta(3)G(0|x) \\
&\quad + \frac{5x - x^2}{24 - 72x + 72x^2 - 24x^3}\pi^2G^2(0|x) - \frac{7x + 5x^2}{6 - 18x + 18x^2 - 6x^3}G^3(0|x)
\end{aligned}$$

$$\begin{aligned}
& - \frac{12x + 16x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, 1|x) G(0|x) - \frac{21x + 19x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, -1|x) G(0|x) \\
& - \frac{33x + 43x^2}{48 - 144x + 144x^2 - 48x^3} G^4(0|x) + \frac{9x + 7x^2}{2 - 6x + 6x^2 - 2x^3} G^2(0|x) G(0, 1|x) \\
& + \frac{11x + 9x^2}{2 - 6x + 6x^2 - 2x^3} G^2(0|x) G(0, -1|x) + \frac{14x + 22x^2}{1 - 3x + 3x^2 - x^3} G(0, 0, 0, 1|x) \\
& + \frac{x + x^2}{1 - 3x + 3x^2 - x^3} \left[24G(0, 0, 0, -1|x) + G^3(0|x) G(-1|x) \right] + \frac{x}{12 - 24x + 12x^2} \left[\right. \\
& 2\pi^2 G(0|x) - 4\pi^2 G(0|x) G(1|x) - 2\pi^2 G(0, -1|x) + 24G(0, -1|x) G(0|x) \\
& + 4\pi^2 G(0, 1|x) - 72G(0, 1|x) G(0|x) G(1|x) - 24G(0, -1, -1|x) G(0|x) + 36\zeta(3) \\
& - 48G(0, 0, -1|x) + 48G(0, 0, 1|x) G(1|x) + 48G(0, 1, 1|x) G(0|x) - 21G^2(0|x) \\
& - 48G(0, 0, 1, 1|x) - 10G^3(0|x) G(1|x) + 24G^2(0|x) G^2(1|x) + 12G^2(0, -1|x) \\
& \left. + 12G^2(0, 1|x) + 48\zeta(3)G(1|x) \right].
\end{aligned}$$

$$\begin{aligned}
\triangleleft &= \iiint_{l_1 l_2 l_3} \frac{1}{[l_1^2 - 1][l_2^2 - 1][(l_2 - q_1 - q_2)^2 - 1][l_3^2 - 1][(l_3 + q_2)^2 - 1]} \\
& \cdot \frac{1}{[(l_1 - q_1)^2 - 1][(l_1 - l_3 - q_1 - q_2)^2][(l_1 - l_2)^2][(l_2 - l_3 - q_1 - q_2)^2]} \\
&= F_0^{148} + \mathcal{O}(\epsilon),
\end{aligned}$$

where

$$\begin{aligned}
F_0^{148} &= \frac{4x^2}{45 - 180x + 270x^2 - 180x^3 + 45x^4} \left[2\pi^4 G(0|x) + 2160G(0, 0, 0, -1|x) G(0|x) \right. \\
& + 1620G(0, 0, 0, 1|x) G(0|x) - 3600G(0, 0, 0, 0, -1|x) - 2700G(0, 0, 0, 0, 1|x) \\
& - 540G^2(0|x) G(0, 0, -1|x) - 405G^2(0|x) G(0, 0, 1|x) + 60G^3(0|x) G(0, -1|x) \\
& \left. + 45G^3(0|x) G(0, 1|x) + 675\zeta(5) \right].
\end{aligned}$$

Conclusion and Outlook 6

The ambitious goal of this thesis was the computation of the light quark mass effects to the NNLO virtual QCD corrections for two similar processes, the Higgs production via gluon-fusion ($gg \rightarrow H$) and the Higgs decay into two photons ($H \rightarrow \gamma\gamma$). Although good progress has been made, this goal has not yet been fully accomplished.

We designed a setup for the computations with standard tools for multiloop calculations and managed the reduction of all Feynman integrals of the $H \rightarrow \gamma\gamma$ amplitude to scalar master integrals. Many different techniques were investigated for their applicability to those master integrals.

For a large subset of master integrals, recent developments in the method of differential equations proved useful. Within a special basis, the master integrals in d space-time dimensions fulfill a system of differential equations with the right-hand side proportional to $\epsilon = (4 - d)/2$. This system can be solved order-by-order in ϵ . We developed the tool `epsilon`, published in Ref. [1], in order to find such a basis. This tool implements an algorithm suggested by Lee in Ref. [63]. It enabled us to successfully find a canonical bases for all master integrals of $H \rightarrow \gamma\gamma$ that can be expressed in Goncharov polylogarithms of the kinematic variable x . Even though we obtained a solution for all these master integrals, the computation of the Feynman amplitude requires even higher orders in ϵ than our current results. In principle, there should be no major obstacles in obtaining these higher orders in the framework of our setup other than the availability of sufficient computing resources.

The remaining master integrals require different techniques. Here, we aim for an expansion in the mass of the mediating quark, which is considered small compared to the Higgs mass. We developed a strategy to expand the system of differential equations around a small quark mass (or, equivalently, around small x) and to solve this expanded version with a suitable ansatz. This leads to an expression of the expanded master integrals in terms of a rather small number of “master coefficients”. The evaluation of these master coefficients requires sophisticated techniques to obtain leading-order terms in the asymptotic expansions of Feynman integrals. Since this is an expansion in a non-Euclidean limit, the strategy of subgraphs cannot be used here. We found the Mellin-Barnes representation of Feynman integrals most useful to extract the master coefficients. Unfortunately, finding low-dimensional Mellin-Barnes representations is a highly non-trivial task. A technique inspired by the “Method of Brackets” [115–117] was developed in this thesis and published in Ref. [2]. In combination with the public `Mathematica` package `AMBRE` [111–114], we hope to find the best Mellin-Barnes representations for all our master integrals, but this step is still work in progress. If the Mellin-Barnes approach fails for some master integrals, it is also possible to obtain the leading-order terms of the asymptotic expansions via different strategies, e.g. the expansion of regions [57, 106–108].

Even though the computation of the amplitude for $gg \rightarrow H$ was the main motivation of this thesis, it is yet in a very early stage. Here, we still require the reduction tables to master integrals. Hopefully, after the reduction is done, the presented techniques can be applied analogously to this

more involved computation. Besides the virtual corrections, also real gluon radiation effects have to be taken into account to obtain finite results after renormalization (see Ref. [136] for first results).

Despite the fact that we did not succeed in completing the computation for two particular processes, we gained significant insights about what techniques are required on the deepest level of such a calculation. However, since Feynman integrals are the basic building blocks of any multiloop calculation, we are confident that the developed methods have a broader applicability than what we used them for so far.

Notation and Conventions

A

Throughout this thesis, we work in natural units ($\hbar = c = 1$). Consequently, masses, momenta and energies are measured in electronvolts (eV) and distances and time in eV^{-1} .

The metric tensor of a four-dimensional Minkowski space-time is defined by

$$g = \text{diag}(1, -1, -1, -1),$$

i.e. scalar products of four-dimensional Minkowski vectors are given by

$$a \cdot b \equiv a_\mu b^\mu \equiv a_0 b_0 - \mathbf{a} \cdot \mathbf{b}.$$

Here, we used a bold font to indicate three-dimensional Euclidean vectors. For four-dimensional Euclidean vectors, we use capital letters, for which scalar products are given by

$$A \cdot B \equiv A_0 B_0 + \mathbf{A} \cdot \mathbf{B}.$$

Feynman integrals are regularized with dimensional regularization, where the number of space-time dimensions is given by $d = 4 - 2\epsilon$. We introduce the notation

$$\int_l \equiv \frac{d-2}{2\Gamma(3-d/2)} \int \frac{d^d l}{i\pi^{d/2}} \quad (\text{A.1})$$

for loop-integrations with a Minkowski space-time metric. Likewise, we define an Euclidean loop integral by

$$\int_L \equiv \frac{d-2}{2\Gamma(3-d/2)} \int \frac{d^d L}{\pi^{d/2}}. \quad (\text{A.2})$$

With these somewhat unorthodox definitions, tadpole integrals are particularly simple. For example the three-loop tadpole integral would be

$$\iiint_{l_1 l_2 l_3} \frac{1}{[l_1^2 - 1 + i0][l_2^2 - 1 + i0][l_3^2 - 1 + i0]} = \left\{ \int_l \frac{1}{l^2 - 1 + i0} \right\}^3 = \frac{1}{\epsilon^3}$$

The notation $\dots + i0$ describes an infinitesimal positive imaginary part. This shift is dropped where it is non-essential.

We always denote the two momenta of the incoming particles of the process $gg \rightarrow H$ and of the two outgoing particles of the process $H \rightarrow \gamma\gamma$ as q_1 and q_2 . The Higgs boson carries mass M and

the quark running in a closed loop carries mass m . The dimensionless kinematic variable τ (cf. Section 2.5) is given by

$$\tau = \frac{M^2}{4m^2}.$$

It is useful to rescale the quark mass m to unity. In doing so, all momenta become dimensionless, such that

$$q_1 \cdot q_2 = 2\tau.$$

For many purposes, it is preferable (and sometimes even required) to choose a different kinematic variable x by a conformal mapping

$$x = \frac{\sqrt{1 - 1/\tau} - 1}{\sqrt{1 - 1/\tau} + 1} + i0.$$

More details on this variable transformation is given in Section 2.5.

A Goncharov polylogarithm of weight n with argument x is denoted by $G(a_1, \dots, a_n|x)$. In this thesis, we often make use of the short-hand notation $G(\vec{a}|x) = G(a_1, \dots, a_n|x)$ if the weight is non-essential or obvious from the context. Similarly, we use $G(b, \vec{a}|x) = G(b, a_1, \dots, a_n|x)$ and $G(\vec{a}, b|x) = G(a_1, \dots, a_n, b|x)$ to prepend or append an index.

Mellin-Barnes representations often depend on sums of many integration variables z_i for which we write for brevity

$$z_{ijk\dots} = z_i + z_j + z_k + \dots. \tag{A.3}$$

The same short-hand notation is also used, if appropriate, for other types of variables. If more than nine variables are involved, we sort the variables as in

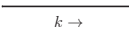
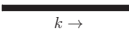
$$z_1, \dots, z_9, z_a, z_b, \dots$$

in order to avoid confusion when the notation (A.3) is used.

A.1 Diagrammatic Representation of Feynman Integrals

For Feynman integrals, a diagrammatic representation similar to Feynman graphs is often useful. Unless specified otherwise, we use a consistent notation for all Feynman integrals in the thesis.

Internal *thin* lines are used for *massless* propagators and internal *bold* lines for *massive* propagators with mass 1, i.e.

| | | |
|----------------------|---|--------------------------|
| massless line |  | $\frac{1}{k^2 + i0}$ |
| massive line |  | $\frac{1}{k^2 - 1 + i0}$ |

External *thin* lines denote incoming or outgoing *massless* particles. External *double* lines are used for incoming or outgoing *massive* particles with a squared mass of 4τ . A dot in an internal line increases the corresponding propagator power by one.

For the loop integrals, we usually imply the definition (A.1).

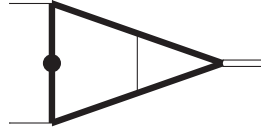


Fig. A.1: Example Feynman integral in diagrammatic notation.

As an example, we consider the two-loop Feynman integral in fig. A.1, which is given by

$$\iint_{l_1 l_2} \frac{1}{[l_1^2 - 1]^2 [(l_1 + q_1)^2 - 1] [(l_1 - l_2 + q_1)^2 - 1] [(l_1 - l_2 - q_2)^2 - 1] [(l_1 - q_2)^2 - 1] [l_2^2]},$$

where $q_1^2 = q_2^2 = 0$ and $q_1 \cdot q_2 = 2\tau$.

A.2 Pseudo-Code Conventions

In this thesis, we discuss many different algorithms, which are usually given in pseudo-code. Our pseudo-code conventions follow a procedural imperative programming paradigm, which is close to many real programming languages (e.g. C++).

The code is structured in blocks (**function**, **if**, etc.), which are always closed by an **end** directive (**end function**, **end if**, etc.). The blocks **function** and **procedure** both contain subroutines of the algorithm. The difference is that procedures do not have return values.

The directives **function** and **procedure** are followed by a list of arguments in parentheses. The parameters are explained via an **in:** statement in the line following the **function/procedure** directive. In the same way, an **out:** statement specifies the return value.

Functions with more than one return value define a structure with named symbols. This is declared at the end of the line of the **function** statement with a right arrow “ \rightarrow ” and a list of symbols in parenthesis (see e.g. alg. E.1). The meaning of the symbols are then explained in the **out:** statement. In that case, the **return** statement gets a list of all return values in the same order as in the defined structure.

We use the data structures *lists*, *sets* and *multisets*. A *list* is an enumeration of elements without a specific order. A literal list is given in curly braces “ $\{b, a, c, b, d, \dots\}$ ”. A *set* is an ordered enumeration of distinct elements and is given in parentheses “ (a, b, d, e, \dots) ”. By *multiset* we mean an ordered enumeration of elements, which are not necessarily distinct. Literal multisets are also given in parentheses. The distinction of sets from multisets should be clear from the context. The notation $|A|$ is used for the number of elements in the list, set or multiset A .

The assignment of variables is denoted by a left arrow “ \leftarrow ”.

We often make use of loops of the form

for all (distinct) $e \in S$.

Such a loop iterates over all elements in S . The order is specified by the type of S , which can be a list, a set or a multiset. If S is a list or a multiset, an extra keyword **distinct** will discard multiple occurrences of equal elements. In the body of the for-loop, e contains the element of the corresponding loop run.

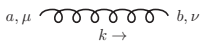
The conditional block starting with

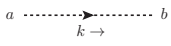
if $\exists i : condition$ then

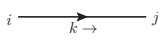
is executed if an i exists for which *condition* is true. Depending on the target programming language, this can often not be implemented within a simple **if** statement.

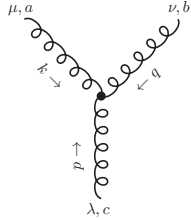
Feynman Rules B

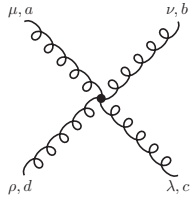
In this appendix, we list all momentum space Feynman rules required in this thesis. These are the Feynman rules of QCD in Feynman gauge, the Yukawa coupling of the Higgs boson to a quark, and the photon-quark-antiquark vertex. Since the Higgs boson and the photon only appear as external particles in our calculation, corresponding propagators are not required. The mass m_q denotes the mass of the quark in the quark propagator and in the Yukawa coupling to the Higgs boson. The electric charge of the quark in units of the elementary charge is denoted by Q_q in the photon-quark-antiquark vertex.

gluon propagator 
$$-\frac{ig_{\mu\nu}\delta^{ab}}{k^2+i0} \quad (\text{B.1})$$

ghost propagator 
$$\frac{i\delta^{ab}}{k^2+i0} \quad (\text{B.2})$$

quark propagator 
$$\frac{i(\not{k}+m_q)\delta^{ij}}{k^2-m_q^2+i0} \quad (\text{B.3})$$

3-gluon vertex 
$$-gf^{abc} [g^{\mu\nu}(k-q)^\lambda + g^{\nu\lambda}(q-p)^\mu + g^{\lambda\mu}(p-k)^\nu] \quad (\text{B.4})$$

4-gluon vertex 
$$-ig^2 [f^{abe}f^{cde}(g^{\mu\lambda}g^{\nu\rho} - g^{\mu\rho}g^{\nu\lambda}) + f^{ace}f^{bde}(g^{\mu\nu}g^{\lambda\rho} - g^{\mu\rho}g^{\nu\lambda}) + f^{ade}f^{cbe}(g^{\mu\lambda}g^{\nu\rho} - g^{\mu\nu}g^{\rho\lambda})] \quad (\text{B.5})$$

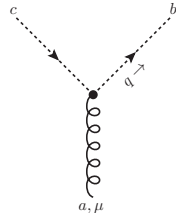
gluon-quark-antiquark
vertex



$$-ig\gamma^\mu T_{ji}^a$$

(B.6)

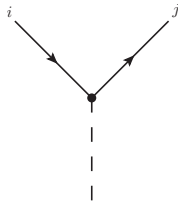
gluon-ghost-antighost
vertex



$$-gf^{abc}q^\mu$$

(B.7)

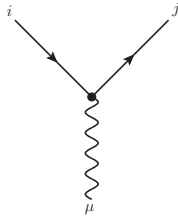
Higgs-quark-antiquark
vertex



$$-\frac{im_q\delta_{ji}}{v}$$

(B.8)

photon-quark-
antiquark vertex



$$-ieQ_q\gamma^\mu\delta_{ji}$$

(B.9)

One-Scale Master Integrals



In this appendix, we calculate the seven one-scale master integrals depicted in fig. 4.3. These integrals are required for the determination of the boundary conditions for the method of differential equations used in Section 4.1.

The one-loop tadpole, the one- and two-loop propagator diagrams, the planar vertex, and the three-loop bubble with two massive lines are covered in Section C.1. The three-loop bubble with four massive lines is calculated in Section C.2 using a configuration space technique. And last but not least, in Section C.3, the non-planar vertex diagram is evaluated via a MB representation.

C.1 The Simple Ones

By far the simplest master integral is the one-loop tadpole, given by

$$\text{tadpole} = \frac{1}{\epsilon},$$

where we implied the Feynman integral measure (A.1). The computational steps to obtain this result are given in many textbooks on quantum field theory (see e.g. [53]).

The one- and two-loop propagator diagrams, the planar vertex diagram, and the three-loop bubble with two massive lines can be expressed in terms of gamma functions. For this purpose, we can use the method described in Section 3.7. The presolutions obtained for these integrals contain just as many brackets as MB integrals. Therefore, all MB integrals can be solved in the last step, Rule D of Section 3.7.1.2. The results read

$$\begin{aligned} \text{bubble} &= \frac{\Gamma(1-\epsilon)\Gamma(2-\epsilon)\Gamma(\epsilon)}{\Gamma(2-2\epsilon)\Gamma(1+\epsilon)} (-4\tau)^{-\epsilon}, \\ \text{propagator} &= \frac{2\Gamma^3(2-\epsilon)\Gamma(-2+2\epsilon)}{\Gamma(3-3\epsilon)\Gamma^2(1+\epsilon)} (-4\tau)^{1-2\epsilon}, \\ \text{planar vertex} &= \frac{\Gamma(1-2\epsilon)\Gamma^2(2-\epsilon)\Gamma(2\epsilon)}{\epsilon^2(1-2\epsilon)\Gamma(2-3\epsilon)\Gamma(\epsilon)} (-4\tau)^{-2\epsilon}, \\ \text{three-loop bubble} &= -\frac{2^{1-2\epsilon}\Gamma(2-\epsilon)\Gamma(-1/2+\epsilon)\Gamma(-2+3\epsilon)}{\epsilon^3\Gamma(-1+\epsilon)\Gamma(-1/2+2\epsilon)}. \end{aligned}$$

This one-dimensional integral can now be evaluated by `Mathematica` [84] in terms of (generalized) hypergeometric functions ${}_pF_q$ to

$$\begin{aligned} \text{Diagram} &= \frac{(1-\epsilon)\Gamma(2-\epsilon)\Gamma^2(-1+2\epsilon)\Gamma(-2+3\epsilon)}{\epsilon\Gamma^2(1+\epsilon)\Gamma(-2+4\epsilon)} {}_2F_1\left(\begin{matrix} -\frac{1}{2}+\epsilon, -2+3\epsilon \\ -\frac{1}{2}+2\epsilon \end{matrix} \middle| 1\right) \\ &+ \frac{2-2\epsilon}{\epsilon^3(1-2\epsilon)} {}_3F_2\left(\begin{matrix} \frac{1}{2}, 1, -1+2\epsilon \\ 2-\epsilon, \frac{1}{2}+\epsilon \end{matrix} \middle| 1\right) + \frac{1-\epsilon}{\epsilon^3} {}_3F_2\left(\begin{matrix} 1, \frac{3}{2}-\epsilon, \epsilon \\ \frac{3}{2}, 3-2\epsilon \end{matrix} \middle| 1\right). \end{aligned}$$

An ϵ -expansion can be obtained by the `Mathematica`-package `HypExp` [146, 147], which yields

$$\begin{aligned} \text{Diagram} &= \frac{2}{\epsilon^3} + \frac{5}{3\epsilon^2} + \frac{1}{2\epsilon} - \frac{103}{12} + \epsilon\left(-\frac{1141}{24} + \frac{112\zeta(3)}{3}\right) \\ &+ \epsilon^2\left(-\frac{9055}{48} - \frac{136\pi^4}{45} - \frac{32\pi^2\ln^2 2}{3} + \frac{32\ln^4 2}{3} + 256\text{Li}_4\left(\frac{1}{2}\right) + 168\zeta(3)\right) \\ &+ \epsilon^3\left(-\frac{63517}{96} - \frac{68\pi^4}{5} + \frac{272\pi^4\ln 2}{15} - 48\pi^2\ln^2 2 + \frac{64\pi^2\ln^3 2}{3} + 48\ln^4 2 \right. \\ &\quad \left. - \frac{64\ln^5 2}{5} + 1152\text{Li}_4\left(\frac{1}{2}\right) + 1536\text{Li}_5\left(\frac{1}{2}\right) + \frac{1876\zeta(3)}{3} - 1240\zeta(5)\right) \\ &+ \epsilon^4\left(-\frac{418903}{192} + 3840s_6 - \frac{2278\pi^4}{45} - \frac{32\pi^6}{5} + \frac{408\pi^4\ln 2}{5} - \frac{536\pi^2\ln^2 2}{3} \right. \\ &\quad \left. - \frac{272\pi^4\ln^2 2}{5} + 96\pi^2\ln^3 2 + \frac{536\ln^4 2}{3} - 32\pi^2\ln^4 2 - \frac{288\ln^5 2}{5} \right. \\ &\quad \left. + \frac{64\ln^6 2}{5} + 4288\text{Li}_4\left(\frac{1}{2}\right) + 6912\text{Li}_5\left(\frac{1}{2}\right) + 9216\text{Li}_6\left(\frac{1}{2}\right) \right. \\ &\quad \left. + \frac{6398\zeta(3)}{3} - \frac{4880\zeta^2(3)}{3} - 5580\zeta(5)\right) + \mathcal{O}(\epsilon^5), \end{aligned}$$

where the constant $s_6 = \sum_{i=1}^{\infty} \sum_{j=1}^i (-1)^{i+j} / (i^5 j) \approx 0.98744142640329971377$ can be found in [101]. We checked the results up to order ϵ^3 against Ref. [148].

C.3 Non-Planar Vertex Diagram

For the non-planar vertex diagram, we used the Mellin-Barnes technique. The method in Section 3.7 leads to a two-dimensional MB representation

$$\begin{aligned} \text{Diagram} &= (-4\tau)^{-2-2\epsilon} \int_{c_1} \frac{dz_1}{2\pi i} \int_{c_2} \frac{dz_2}{2\pi i} \frac{(1-\epsilon)^2 \Gamma^2(-\epsilon) \Gamma^2(-z_1) \Gamma(1+z_1) \Gamma(-2\epsilon+z_1) \Gamma(\epsilon-z_1-z_2)}{\Gamma(-3\epsilon) \Gamma(-2\epsilon) \Gamma^2(1+\epsilon)} \\ &\quad \cdot \frac{\Gamma(2\epsilon-z_1-z_2) \Gamma^2(-z_2) \Gamma(1+z_2) \Gamma(-2\epsilon+z_2)}{\Gamma^2(-z_1-z_2)}. \end{aligned}$$

The singularities in ϵ can be resolved using the `Mathematica`-package `MBResolve` [119] (cf. Section 3.6.1). This yields a sum of MB integrals over straight lines parallel to the imaginary axis, given by

$$\begin{aligned}
 \text{Diagram} &= (-4\tau)^{-2-2\epsilon} \left\{ \frac{(1-\epsilon)^2 \Gamma^4(-2\epsilon) \Gamma^2(-\epsilon) \Gamma^2(1+2\epsilon)}{\Gamma^2(1+\epsilon) \Gamma^2(-4\epsilon)} \right. \\
 &+ \int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \frac{dz}{2\pi i} \frac{(1-\epsilon)^2 \Gamma(-2\epsilon) \Gamma^2(-\epsilon) \Gamma(1+2\epsilon) \Gamma(-\epsilon-z) \Gamma^3(-z) \Gamma(1+z) \Gamma(-2\epsilon+z)}{\Gamma^2(1+\epsilon) \Gamma(-3\epsilon) \Gamma^2(-2\epsilon-z)} \\
 &+ \int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \frac{dz_1}{2\pi i} \int_{-\frac{3}{4}-i\infty}^{-\frac{3}{4}+i\infty} \frac{dz_2}{2\pi i} \frac{(1-\epsilon)^2 \Gamma^2(-\epsilon) \Gamma^2(-z_1) \Gamma(1+z_1) \Gamma(-2\epsilon+z_1) \Gamma(\epsilon-z_1-z_2)}{\Gamma^2(1+\epsilon) \Gamma(-3\epsilon) \Gamma(-2\epsilon)} \\
 &\left. \cdot \frac{\Gamma(2\epsilon-z_1-z_2) \Gamma^2(-z_2) \Gamma(1+z_2) \Gamma(-2\epsilon+z_2)}{\Gamma^2(-z_1-z_2)} \right\}.
 \end{aligned}$$

At this stage, the integrands can be Laurent-expanded in ϵ . The series coefficients are linear combinations of up to two-dimensional MB integrals independent of ϵ and can be expressed as multiple sums with the help of the `MBsums`-package [143] by closing the contours to the left. The multifold sums are then solved in terms of multiple zeta values with the code `XSummer` [139], which leads to

$$\begin{aligned}
 \text{Diagram} &= (-4\tau)^{-2-2\epsilon} \left\{ \frac{1}{\epsilon^4} - \frac{2}{\epsilon^3} + \frac{6-7\pi^2}{6\epsilon^2} + \frac{7\pi^2-81\zeta(3)}{3\epsilon} - \frac{7\pi^2}{6} - \frac{19\pi^4}{60} + 54\zeta(3) \right. \\
 &+ \epsilon \left(\frac{19\pi^4}{30} - 27\zeta(3) + 17\pi^2\zeta(3) - 117\zeta(5) \right) \\
 &+ \epsilon^2 \left(-\frac{19\pi^4}{60} + \frac{85\pi^6}{756} - 34\pi^2\zeta(3) + 267\zeta^2(3) + 234\zeta(5) \right) \\
 &+ \epsilon^3 \left(-\frac{85\pi^6}{378} + \frac{211\pi^4\zeta(3)}{30} - 534\zeta^2(3) - 117\zeta(5) \right. \\
 &\quad \left. + 17\pi^2\zeta(3) + 79\pi^2\zeta(5) + 6\zeta(7) \right) \\
 &+ \epsilon^4 \left(\frac{85\pi^6}{756} + \frac{3169\pi^8}{8400} - \frac{211\pi^4\zeta(3)}{15} + 267\zeta^2(3) - 133\pi^2\zeta^2(3) \right. \\
 &\quad \left. - 158\pi^2\zeta(5) + 2466\zeta(3)\zeta(5) - 264s_{8a} - 12\zeta(7) \right) \\
 &+ \epsilon^5 \left(-\frac{3169\pi^8}{4200} + 528s_{8a} + \frac{211\pi^4\zeta(3)}{30} - \frac{359\pi^6\zeta(3)}{630} + 266\pi^2\zeta^2(3) \right. \\
 &\quad - 1626\zeta^3(3) + 79\pi^2\zeta(5) + \frac{1709\pi^4\zeta(5)}{30} - 4932\zeta(3)\zeta(5) \\
 &\quad \left. + 6\zeta(7) + 201\pi^2\zeta(7) + \frac{20777\zeta(9)}{3} \right) \\
 &\left. + \mathcal{O}(\epsilon^6) \right\},
 \end{aligned}$$

where the constant $s_{8a} = \sum_{i=1}^{\infty} \sum_{j=1}^i 1/(i^5 j^3) \approx 1.0417850291827918834$ is given in [101]. The same result for this master integral can be found up to order ϵ^3 in Ref. [149]¹.

¹This reference has been found via Loopedia [150].

The tool `epsilon` D

In this appendix, we provide more details about the tool `epsilon` developed within this thesis and published in Ref. [1]. A description of the algorithms used by this code can be found in Section 3.3 and in the paper by Lee [63].

Section D.1 gives installation and usage instructions for `epsilon`. Section D.2 lists all transformations used by the algorithm in a form in which they are implemented in the package.

D.1 Usage

D.1.1 Installation Guide on Linux Systems

Ensure that the dependencies

- `Fermat` (≥ 6.0) [66],
- `GiNaC` ($\geq 1.6.2$) [151] (for `epsilon-prepare` only)

are installed.

As a next step, `libFermat` has to be installed. `libFermat`, which was developed in connection with `epsilon`, is a C++ library designed to communicate with `Fermat`. Nevertheless, we decided to publish `libFermat` in a separate repository since it might be useful elsewhere. Internally, the communication is done with `PStreams` [152], which is included in the package. The source code of the most recent version of `libFermat` can be obtained via `GitHub` using

```
git clone https://github.com/mprausa/libFermat.git
```

This will create a directory `libFermat/` and clone the library into that location. Now inside this directory, run

```
cmake -DCMAKE_INSTALL_PREFIX=/path/to/install .
make
make install
```

where `/path/to/install` is your desired installation directory and defaults to `/usr/local` on a typical Linux system. The library is installed into the subdirectory `lib` and the header files into the subdirectory `include` of `/path/to/install`. If your choice is a global directory you will require `root` access for the last step `make install`. Remember to include the subdirectory `lib` of `/path/to/install` into the `LD_LIBRARY_PATH` environment variable if you are using a non-standard directory.

The next step is to install `epsilon` and `epsilon-prepare`. In principle the procedure is the same as for the installation of `libFermat`. First, obtain the most recent version of the source code with

```
git clone https://github.com/mprausa/epsilon.git
```

then change into the newly created directory `epsilon/` and run

```
cmake -DCMAKE_INSTALL_PREFIX=/path/to/install .
make
make install
```

It is recommended to use the same `/path/to/install` as for `libFermat`, else the `cmake` step might require additional options to find `libFermat`. The programs `epsilon` and `epsilon-prepare` are installed into the subdirectory `bin` of the installation path. As before, `make install` might need `root` access depending on the installation prefix. For a non-standard installation prefix, the environment variable `PATH` should be adjusted to include the subdirectory `bin` of `/path/to/install` so that the programs `epsilon` and `epsilon-prepare` can be found by the shell.

It is also possible to build `epsilon` and `epsilon-prepare` individually. This can be done by changing into the corresponding subdirectory and running `cmake` and `make` from within there.

The `epsilon`-repository also offers a `Mathematica` package `EpsilonTools.m` found in the subdirectory `mma/`. Run

```
./install.sh
```

from within this subdirectory to install `EpsilonTools.m` into the `Applications/` directory of your `Mathematica` installation.

D.1.2 Input/Output Format

`epsilon` uses its own file format to represent a system of differential equations of the form (3.27), where every line represents one coefficient matrix. A line starts with either `'A[xj,k]:'` or `'B[k]:'` followed by a matrix. The matrix is stored as a list of rows, where each row is itself a list of matrix elements. Lists are enclosed in curly-braces and list entries are separated by commas. A line starting with `'A[xj,k]:'` (`'B[k]:'`) represents a matrix $M_k^{(x_j)}$ (M_k) in (3.27).

The name of the symbol used to represent ϵ is fixed to `ep`.

An example of an input file for `epsilon` is given in Ref. [1].

D.1.3 Usage of `epsilon-prepare`

The tool `epsilon-prepare` is used to convert a matrix in a `Mathematica` readable format (a list of lists) into the `epsilon` input format (see section D.1.2). If, for example, the file containing a matrix in the `Mathematica` format is called `matrix.m`, then the command

```
epsilon-prepare matrix.m > matrix.dat
```

is used to create a file `matrix.dat` in the `epsilon` format. The matrix elements of the input matrix are expected to be rational functions in ϵ and x (represented by the symbols `ep` and `x`, respectively). The command `epsilon-prepare` performs a partial fraction decomposition over complex numbers in the variable x of the matrix elements and results in an expression of the form of (3.27).

In order to perform a partial fraction decomposition of a rational function, the zeros of its denominator have to be determined. Therefore, `epsilon-prepare` applies `GiNaC`'s polynomial

| symbol | polymod | expression |
|-----------------|-------------------------|-------------------------|
| i | $i^2 + 1$ | i |
| rN | $rN^2 - rN + (1 + N)/4$ | $\frac{1+i\sqrt{N}}{2}$ |
| qN | $qN^2 - qN + (1 - N)/4$ | $\frac{1+\sqrt{N}}{2}$ |
| $\text{sqrt}N$ | $\text{sqrt}N^2 - N$ | \sqrt{N} |
| $\text{isqrt}N$ | $\text{isqrt}N^2 + N$ | $i\sqrt{N}$ |

Tab. D.1: Additional symbols used by *epsilon-prepare* to represent zeros of quadratic polynomials. N is a positive integer.

factorization algorithm to the denominator in order to factorize it into polynomials that are irreducible over the integers. In a second step, the zeros of all factors are found individually.

This step will fail, if the considered irreducible polynomial has a degree larger than two. An error is thrown as well if the system is not in form (3.27), i.e. a zero depends on the parameter ϵ .

The additional symbols listed in tab. D.1 might be introduced by `epsilon-prepare`. See section D.1.5 for how to make `epsilon` accept them.

D.1.4 Usage of `epsilon`

The general command syntax for the tool `epsilon` is

```
epsilon [OPTIONS] JOBS...
```

The path to the `Fermat` binary can be set inside the environment variable `FERMAT`. If this variable is not set explicitly, `epsilon` will look for a binary `fer64` inside the directories defined in the environment variable `PATH`. Options are set once at the start of `epsilon`. Jobs are processed one-by-one in the same order they are defined on the command line. Some jobs will perform transformations to the system. These transformations are stored in a so-called internal transformation queue in RAM and can also be written to an external file.

Options:

- `--verbose:` Enable verbose output.
This option prints out all communication between `libFermat` and `Fermat` during a regular run. This is useful as a debugging tool.
- `--timings:` Enable timings.
This option prints the elapsed real time after every job and the total time of the complete run.
- `--symbols symbols:` Adjoin additional symbols to `Fermat`.
This option adds the specified symbols to `Fermat`. The symbols defined in this option are the very first variables adjoint to `Fermat` followed by the symbol `ep` and further internally used symbols. *symbols* has to be a comma-separated list.
- `--echelon-fermat:` Use the `Redrowech` function in `Fermat` to solve linear systems.

At various points in the code linear systems of equations have to be solved. Use this option to choose the `Redrowech` function over our own implementation of Gaußian elimination.

Jobs:

- `--fermat file:` Execute `Fermat` commands.

This job reads *file* line-by-line and sends all non-empty lines to `Fermat`.

- `--load file start end:` Load system.

This job loads *file* and activates the block $\{start, end\}$. Hereby *file* is expected to be in the format specified in Section D.1.2

- `--write file:` Write system.

This job writes the system of differential equations to *file* using the format specified in Section D.1.2.

- `--queue file:` Set external transformation queue.

This job enables an external transformation queue. An external transformation queue is a file containing all transformations already performed by `epsilon` during a run. This is particularly useful in connection with the options `--load-queue` and `--replay` to restore an aborted run to the state after the last successful transformation.

- `--load-queue filename:` Load transformation queue.

This job loads an external transformation queue from *filename* into an internal transformation queue stored in RAM. This job does *not* apply the transformations stored in the file to the system.

- `--replay:` Apply internal transformation queue.

This job ‘replays’ the internal transformation queue, i.e. the transformations in the queue are applied to the system one-by-one. It should only be used immediately after `--load-queue`.

- `--export file:` Export transformation matrix.

This job computes a transformation matrix out of the transformations inside the internal transformation queue. The matrix is written in a `Mathematica` readable format to *file*.

- `--block start end:` Activate a block.

This job activates the block $\{start, end\}$.

- `--fuchsify:` Transform active block into Fuchsian form.

This job reduces the active block to Fuchsian form (3.28). See Section 3.3.3 for details.

- `--fuchsify-at sing:` Reduce Poincaré rank of the singularity *sing* to zero.

This job reduces the Poincaré rank of the the active block at the singularity *sing* to zero.

- `--normalize:` Normalize eigenvalues.

This job normalizes the eigenvalues of all residue matrices making them proportional to ϵ . See Section 3.3.3 for details.

- `--factorep:` Factor out ϵ (auto detect μ).

This job transforms the active block into ϵ -form (3.29) using the method described in Section 3.3.3. The variable μ in (3.47) is left as an unknown and will be fixed only *after* the

system is solved to ensure that the transformation is invertible.

- `--factorep-at mu`: Factor out ϵ (with predefined μ).

This job transforms the active block into ϵ -form (3.29) using the method described in Section 3.3.3. In this variant, the variable μ in (3.47) is set to *mu* before the system is solved. This is faster than `--factorep` because `epsilon` has to deal with one less variable in the polynomials. Unfortunately, an unlucky choice of *mu* can hit a pole in the matrix elements of the system or one can end up with a non-invertible transformation. In both cases an error is thrown.

- `--left-fuchsify`: Fuchsify off-diagonal block.

This job is used to transform the block to the left of the active block (block \mathbb{B} in (3.30)) to Fuchsian form. See Section 3.3.3 for details.

- `--left-fuchsify-at sing`: Fuchsify off-diagonal block at singularity *sing*.

This job reduces the Poincaré rank of the block to the left of the active block (block \mathbb{B} in (3.30)) at the singularity *sing* to zero.

- `--dyson file order type format`: Generate Dyson operator.

This job writes a Dyson operator $U(x, x_0)$ for the active block up to order *order* in ϵ to *file*. The active block has to be in ϵ -form. The Dyson operator fulfills

$$\frac{\partial}{\partial x} U(x, x_0) = \epsilon \sum_{x_j \in S} \frac{\widehat{\mathbb{C}}_0^{(x_j)}}{x - x_j} U(x, x_0), \quad U(x_0, x_0) = \mathbb{1}.$$

The option *type* specifies the type of multiple polylogarithms in the output and can be set to GPL, HPL or HPLalt for Goncharov polylogarithms [98] or harmonic polylogarithms in the “a”- or “m”-notation [99, 101], respectively. *format* should be `mma` or `form` to specify Mathematica [84] or FORM [52] as output format.

D.1.5 Using Field Extensions

The `Fermat` computer algebra system works with multivariate polynomials over the ground ring \mathbb{Z} and the corresponding quotient field, the rational functions over \mathbb{Z} . Fortunately, `Fermat` offers a way to extend the ground ring by setting so-called polymods.

A polymod $p(\xi)$ is a univariate polynomial in ξ , where ξ is one of the variables adjoint to `Fermat`. This instates a new quotient ring $\mathbb{Z}[\xi]/\langle p(\xi) \rangle$ as the ground ring, forcing the condition $p(\xi) = 0$ onto the variable ξ . In other words every polynomial $q(\xi) \in \mathbb{Z}[\xi]$ encountered by `Fermat` is replaced immediately by the remainder of a polynomial division $q(\xi) \div p(\xi)$. For more details see the `Fermat` manual [66].

The polymod $i^2 + 1$ for example leads to the quotient ring $\mathbb{Z}[i]/\langle i^2 + 1 \rangle$ which is equivalent to the Gaussian integers, a field extension of the integers by a number i with $i^2 = -1$. Also other complex numbers can be represented in `Fermat` using polymods. The complex numbers introduced via `epsilon-prepare` can be represented in `Fermat` using the polymods listed in tab. D.1.

The syntax to set a polymod in `Fermat` is `&(P=polymod,1)`, where the variable of the polymod must be the first symbol adjoint to `Fermat` that does not have a polymod assigned yet. This `Fermat` command should be stored in a text file and can be read in by `epsilon` with the `--fermat` job.

For internal reasons `Fermat` runs very slow if more than one polymod is assigned to it. Therefore, in complicated cases with more than one complex number, it is useful to set a polymod only for the

most frequent variable appearing in the block to work with next. As the order of symbols once set cannot be changed inside `Fermat`, the only way to change the ‘active’ polymod is by saving the system with `--write` and reload it into a new session after the symbols are adjoined in a different order. This strategy could be successfully applied to all master integrals in this thesis that can be expressed in terms of GPLs of argument x . The huge system of differential equations for these master integrals even depends on four complex numbers¹ $r = (-1)^{1/3}$, $\varphi = (1 + \sqrt{5})/2$, $\sqrt{2}$, i , where the latter two turned out to be involved only in apparent singularities.

D.1.6 Usage of `EpsilonTools.m`

The Mathematica package `EpsilonTools.m` provides functions which help to set up `epsilon` and to work with the `epsilon` input/output file format (see Section D.1.2). It is not essential in order to run `epsilon`. After a successful installation, the package can be loaded into a Mathematica session with

```
<<EpsilonTools`
```

`EpsilonTools.m` provides three functions:

- `EpsilonSymRules[expression]`

This function scans *expression* for symbols of the form given in tab. D.1 and compiles a list of rules for these symbols to their corresponding Mathematica expressions, e.g.

```
{r3 -> (1 + I * Sqrt[3])/2, q5 -> (1 + Sqrt[5])/2, i -> I}
```

- `EpsilonRead[file]`

This function reads *file* into Mathematica, where *file* is in the format described in Section D.1.2.

Options:

- `ReplaceSymbols` (default: `True`)

If this option is set to `True`, all symbols introduced by `epsilon-prepare` will be replaced by their corresponding Mathematica expression.

- `CheckFuchsian` (default: `False`)

If this option is set to `True`, the function returns `$Failed` if the system in *file* is not in Fuchsian form.

- `CheckEpsilon` (default: `False`)

If this option is set to `True`, the function returns `$Failed` if the system in *file* is not in ϵ -form.

- `EpsilonBlocks[M]`

This function scans for a block-triangular structure of the matrix M and returns a list of the boundaries of all diagonal blocks. The returned boundaries are in the form `{start, end}`. The values *start* and *end* can be used in the `--block` job of `epsilon`.

¹The numbers r and φ are denoted `r3` and `q5` in tab. D.1, respectively.

D.2 Transformations

In this section, we give the transformations used in Lee's algorithm [63] in a different form, which was more suitable for an implementation in our tool `epsilon`.

D.2.1 Balances

The main building blocks of Lee's algorithm are balances. In this appendix, we describe how they act on a system in the form (3.27), which is also assumed to be in block-triangular form (3.30).

The projector \mathbb{P} is determined by its impact on the active block \mathbb{C} . Hence, it has to be of the form

$$\mathbb{P} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \mathbb{Q} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Naturally, this also modifies the blocks \mathbb{B} and \mathbb{E} . In this appendix, we omit writing down the ϵ -dependencies explicitly.

First, we consider a balance between two singularities $|x_{1,2}| < \infty$ (3.31a):

$$\mathbb{T} = \mathcal{B}(\mathbb{P}, x_1, x_2), \quad \mathbb{T}^{-1} = \mathcal{B}(\mathbb{P}, x_2, x_1).$$

The transformation (3.25) can now be written in terms of the coefficient matrices.

We find for the active block

$$\begin{aligned} \tilde{\mathbb{C}}_0^{(x_1)} &= \mathbb{C}_0^{(x_1)} - \sum_{n \geq 0} \frac{1}{(x_2 - x_1)^n} \mathbb{Q} \mathbb{C}_n^{(x_1)} \bar{\mathbb{Q}} + \sum_{x_j \in S \setminus \{x_1\}} \sum_{n \geq 0} \frac{x_1 - x_2}{(x_1 - x_j)^{n+1}} \bar{\mathbb{Q}} \mathbb{C}_n^{(x_j)} \mathbb{Q} \\ &\quad + (x_1 - x_2) \sum_{n \geq 0} x_1^n \bar{\mathbb{Q}} \mathbb{C}_n \mathbb{Q} + \mathbb{Q}, \end{aligned} \quad (\text{D.1a})$$

$$\tilde{\mathbb{C}}_{k>0}^{(x_1)} = \mathbb{C}_k^{(x_1)} + (x_1 - x_2) \bar{\mathbb{Q}} \mathbb{C}_{k-1}^{(x_1)} \mathbb{Q} - \sum_{n \geq 0} \frac{1}{(x_2 - x_1)^n} \mathbb{Q} \mathbb{C}_{n+k}^{(x_1)} \bar{\mathbb{Q}}, \quad (\text{D.1b})$$

$$\begin{aligned} \tilde{\mathbb{C}}_0^{(x_2)} &= \mathbb{C}_0^{(x_2)} - \sum_{n \geq 0} \frac{1}{(x_1 - x_2)^n} \bar{\mathbb{Q}} \mathbb{C}_n^{(x_2)} \mathbb{Q} + \sum_{x_j \in S \setminus \{x_2\}} \sum_{n \geq 0} \frac{x_2 - x_1}{(x_2 - x_j)^{n+1}} \mathbb{Q} \mathbb{C}_n^{(x_j)} \bar{\mathbb{Q}} \\ &\quad + (x_2 - x_1) \sum_{n \geq 0} x_2^n \mathbb{Q} \mathbb{C}_n \bar{\mathbb{Q}} - \mathbb{Q}, \end{aligned} \quad (\text{D.1c})$$

$$\tilde{\mathbb{C}}_{k>0}^{(x_2)} = \mathbb{C}_k^{(x_2)} + (x_2 - x_1) \mathbb{Q} \mathbb{C}_{k-1}^{(x_2)} \bar{\mathbb{Q}} - \sum_{n \geq 0} \frac{1}{(x_1 - x_2)^n} \bar{\mathbb{Q}} \mathbb{C}_{n+k}^{(x_2)} \mathbb{Q}, \quad (\text{D.1d})$$

$$\tilde{\mathbb{C}}_k^{(x_j \neq x_1, x_2)} = \mathbb{C}_k^{(x_j)} + \sum_{n \geq 0} \frac{x_2 - x_1}{(x_1 - x_j)^{n+1}} \bar{\mathbb{Q}} \mathbb{C}_{n+k}^{(x_j)} \mathbb{Q} + \sum_{n \geq 0} \frac{x_1 - x_2}{(x_2 - x_j)^{n+1}} \mathbb{Q} \mathbb{C}_{n+k}^{(x_j)} \bar{\mathbb{Q}}, \quad (\text{D.1e})$$

$$\tilde{\mathbb{C}}_k = \mathbb{C}_k + (x_1 - x_2) \sum_{n \geq 0} \{ x_1^n \bar{\mathbb{Q}} \mathbb{C}_{k+n+1} \mathbb{Q} - x_2^n \mathbb{Q} \mathbb{C}_{k+n+1} \bar{\mathbb{Q}} \}, \quad (\text{D.1f})$$

and for the off-diagonal blocks

$$\begin{aligned}
 \tilde{\mathbb{B}}_0^{(x_2)} &= \mathbb{B}_0^{(x_2)} + \sum_{x_j \in S \setminus \{x_2\}} \sum_{n \geq 0} \frac{x_2 - x_1}{(x_2 - x_j)^{n+1}} \mathbb{Q} \mathbb{B}_n^{(x_j)} + (x_2 - x_1) \sum_{n \geq 0} x_2^n \mathbb{Q} \mathbb{B}_n, \\
 \tilde{\mathbb{B}}_{k>0}^{(x_2)} &= \mathbb{B}_k^{(x_2)} + (x_2 - x_1) \mathbb{Q} \mathbb{B}_{k-1}^{(x_2)}, \\
 \tilde{\mathbb{B}}_k^{(x_j \neq x_2)} &= \mathbb{B}_k^{(x_j)} + \sum_{n \geq 0} \frac{x_1 - x_2}{(x_2 - x_j)^{n+1}} \mathbb{Q} \mathbb{B}_{n+k}^{(x_j)}, \\
 \tilde{\mathbb{B}}_k &= \mathbb{B}_k + (x_2 - x_1) \sum_{n \geq 0} x_2^n \mathbb{Q} \mathbb{B}_{k+n+1}, \\
 \tilde{\mathbb{E}}_0^{(x_1)} &= \mathbb{E}_0^{(x_1)} + \sum_{x_j \in S \setminus \{x_1\}} \sum_{n \geq 0} \frac{x_1 - x_2}{(x_1 - x_j)^{n+1}} \mathbb{E}_n^{(x_j)} \mathbb{Q} + (x_1 - x_2) \sum_{n \geq 0} x_1^n \mathbb{E}_n \mathbb{Q}, \\
 \tilde{\mathbb{E}}_{k>0}^{(x_1)} &= \mathbb{E}_k^{(x_1)} + (x_1 - x_2) \mathbb{E}_{k-1}^{(x_1)} \mathbb{Q}, \\
 \tilde{\mathbb{E}}_k^{(x_j \neq x_1)} &= \mathbb{E}_k^{(x_j)} + \sum_{n \geq 0} \frac{x_2 - x_1}{(x_1 - x_j)^{n+1}} \mathbb{E}_{n+k}^{(x_j)} \mathbb{Q} \\
 \tilde{\mathbb{E}}_k &= \mathbb{E}_k + (x_1 - x_2) \sum_{n \geq 0} x_1^n \mathbb{E}_{k+n+1} \mathbb{Q}.
 \end{aligned}$$

All other blocks are unaffected.

Next, we consider the case $|x_1| < \infty$, $x_2 = \infty$, i.e.

$$\mathbb{T} = \mathcal{B}(\mathbb{P}, x_1, \infty), \quad \mathbb{T}^{-1} = \mathcal{B}(\mathbb{P}, \infty, x_1).$$

In that case, the balances are given by (3.31b) and (3.31c). Here the active block transforms as

$$\begin{aligned}
 \tilde{\mathbb{C}}_0^{(x_1)} &= \mathbb{C}_0^{(x_1)} - \overline{\mathbb{Q}} \mathbb{C}_0^{(x_1)} \mathbb{Q} - \mathbb{Q} \mathbb{C}_0^{(x_1)} \overline{\mathbb{Q}} + \mathbb{Q} \mathbb{C}_1^{(x_1)} \overline{\mathbb{Q}} \\
 &\quad + \sum_{x_j \in S \setminus \{x_1\}} \sum_{n \geq 0} \frac{1}{(x_1 - x_j)^{n+1}} \overline{\mathbb{Q}} \mathbb{C}_n^{(x_j)} \mathbb{Q} + \sum_{n \geq 0} x_1^n \overline{\mathbb{Q}} \mathbb{C}_n \mathbb{Q} + \mathbb{Q}, \\
 \tilde{\mathbb{C}}_{k>0}^{(x_1)} &= \mathbb{C}_k^{(x_1)} - \overline{\mathbb{Q}} \mathbb{C}_k^{(x_1)} \mathbb{Q} - \mathbb{Q} \mathbb{C}_k^{(x_1)} \overline{\mathbb{Q}} + \mathbb{Q} \mathbb{C}_{k+1}^{(x_1)} \overline{\mathbb{Q}} + \overline{\mathbb{Q}} \mathbb{C}_{k-1}^{(x_1)} \mathbb{Q}, \\
 \tilde{\mathbb{C}}_k^{(x_j \neq x_1)} &= \mathbb{C}_k^{(x_j)} - \overline{\mathbb{Q}} \mathbb{C}_k^{(x_j)} \mathbb{Q} - \mathbb{Q} \mathbb{C}_k^{(x_j)} \overline{\mathbb{Q}} + \mathbb{Q} \mathbb{C}_{k+1}^{(x_j)} \overline{\mathbb{Q}} + (x_j - x_1) \mathbb{Q} \mathbb{C}_k^{(x_j)} \overline{\mathbb{Q}} \\
 &\quad - \sum_{n \geq 0} \frac{1}{(x_1 - x_j)^{n+1}} \overline{\mathbb{Q}} \mathbb{C}_{n+k}^{(x_j)} \mathbb{Q}, \\
 \tilde{\mathbb{C}}_0 &= \mathbb{C}_0 - \overline{\mathbb{Q}} \mathbb{C}_0 \mathbb{Q} - (1 + x_1) \mathbb{Q} \mathbb{C}_0 \overline{\mathbb{Q}} + \sum_{x_j \in S} \mathbb{Q} \mathbb{C}_0^{(x_j)} \overline{\mathbb{Q}} + \sum_{n \geq 0} x_1^n \overline{\mathbb{Q}} \mathbb{C}_{n+1} \mathbb{Q}, \\
 \tilde{\mathbb{C}}_{k>0} &= \mathbb{C}_k - \overline{\mathbb{Q}} \mathbb{C}_k \mathbb{Q} - (1 + x_1) \mathbb{Q} \mathbb{C}_k \overline{\mathbb{Q}} + \mathbb{Q} \mathbb{C}_{k-1} \overline{\mathbb{Q}} + \sum_{n \geq 0} x_1^n \overline{\mathbb{Q}} \mathbb{C}_{k+n+1} \mathbb{Q},
 \end{aligned}$$

and the blocks \mathbb{B} and \mathbb{E} as

$$\begin{aligned}
 \tilde{\mathbb{B}}_k^{(x_j)} &= \mathbb{B}_k^{(x_j)} - \mathbb{Q}\mathbb{B}_k^{(x_j)} + \mathbb{Q}\mathbb{B}_{k+1}^{(x_j)} + (x_j - x_1)\mathbb{Q}\mathbb{B}_k^{(x_j)}, \\
 \tilde{\mathbb{B}}_0 &= \mathbb{B}_0 - (1 + x_1)\mathbb{Q}\mathbb{B}_0 + \sum_{x_j \in S} \mathbb{Q}\mathbb{B}_0^{(x_j)}, \\
 \tilde{\mathbb{B}}_{k>0} &= \mathbb{B}_k - (1 + x_1)\mathbb{Q}\mathbb{B}_k + \mathbb{Q}\mathbb{B}_{k-1}, \\
 \tilde{\mathbb{E}}_0^{(x_1)} &= \mathbb{E}_0^{(x_1)} - \mathbb{E}_0^{(x_1)}\mathbb{Q} + \sum_{x_j \in S \setminus \{x_1\}} \sum_{n \geq 0} \frac{1}{(x_1 - x_j)^{n+1}} \mathbb{E}_n^{(x_j)}\mathbb{Q} + \sum_{n \geq 0} x_1^n \mathbb{E}_n \mathbb{Q}, \\
 \tilde{\mathbb{E}}_{k>0}^{(x_1)} &= \mathbb{E}_k^{(x_1)} - \mathbb{E}_k^{(x_1)}\mathbb{Q} + \mathbb{E}_{k-1}^{(x_1)}\mathbb{Q}, \\
 \tilde{\mathbb{E}}_k^{(x_j \neq x_1)} &= \mathbb{E}_k^{(x_j)} - \mathbb{E}_k^{(x_j)}\mathbb{Q} - \sum_{n \geq 0} \frac{1}{(x_1 - x_j)^{n+1}} \mathbb{E}_{n+k}^{(x_j)}\mathbb{Q}, \\
 \tilde{\mathbb{E}}_k &= \mathbb{E}_k - \mathbb{E}_k\mathbb{Q} + \sum_{n \geq 0} x_1^n \mathbb{E}_{k+n+1}\mathbb{Q}.
 \end{aligned}$$

Finally, we consider the case $x_1 = \infty$, $|x_2| < \infty$, i.e.

$$\mathbb{T} = \mathcal{B}(\mathbb{P}, \infty, x_2), \quad \mathbb{T}^{-1} = \mathcal{B}(\mathbb{P}, x_2, \infty).$$

This is similar to the previous case. The active block transforms as

$$\begin{aligned}
 \tilde{\mathbb{C}}_0^{(x_2)} &= \mathbb{C}_0^{(x_2)} - \overline{\mathbb{Q}}\mathbb{C}_0^{(x_2)}\mathbb{Q} - \mathbb{Q}\mathbb{C}_0^{(x_2)}\overline{\mathbb{Q}} + \overline{\mathbb{Q}}\mathbb{C}_1^{(x_2)}\mathbb{Q} \\
 &\quad + \sum_{x_j \in S \setminus \{x_2\}} \sum_{n \geq 0} \frac{1}{(x_2 - x_j)^{n+1}} \mathbb{Q}\mathbb{C}_n^{(x_j)}\overline{\mathbb{Q}} + \sum_{n \geq 0} x_2^n \mathbb{Q}\mathbb{C}_n \overline{\mathbb{Q}} - \mathbb{Q}, \\
 \tilde{\mathbb{C}}_{k>0}^{(x_2)} &= \mathbb{C}_k^{(x_2)} - \overline{\mathbb{Q}}\mathbb{C}_k^{(x_2)}\mathbb{Q} - \mathbb{Q}\mathbb{C}_k^{(x_2)}\overline{\mathbb{Q}} + \mathbb{Q}\mathbb{C}_{k-1}^{(x_2)}\overline{\mathbb{Q}} + \overline{\mathbb{Q}}\mathbb{C}_{k+1}^{(x_2)}\mathbb{Q}, \\
 \tilde{\mathbb{C}}_k^{(x_j \neq x_2)} &= \mathbb{C}_k^{(x_j)} - \overline{\mathbb{Q}}\mathbb{C}_k^{(x_j)}\mathbb{Q} - \mathbb{Q}\mathbb{C}_k^{(x_j)}\overline{\mathbb{Q}} + \overline{\mathbb{Q}}\mathbb{C}_{k+1}^{(x_j)}\mathbb{Q} + (x_j - x_2)\overline{\mathbb{Q}}\mathbb{C}_k^{(x_j)}\mathbb{Q} \\
 &\quad - \sum_{n \geq 0} \frac{1}{(x_2 - x_j)^{n+1}} \mathbb{Q}\mathbb{C}_{n+k}^{(x_j)}\overline{\mathbb{Q}}, \\
 \tilde{\mathbb{C}}_0 &= \mathbb{C}_0 - \overline{\mathbb{Q}}\mathbb{C}_0\mathbb{Q} - \mathbb{Q}\mathbb{C}_0\overline{\mathbb{Q}} + \sum_{n \geq 0} x_2^n \mathbb{Q}\mathbb{C}_{n+1}\overline{\mathbb{Q}} - x_2 \overline{\mathbb{Q}}\mathbb{C}_0\mathbb{Q} + \sum_{x_j \in S} \overline{\mathbb{Q}}\mathbb{C}_0^{(x_j)}\mathbb{Q}, \\
 \tilde{\mathbb{C}}_{k>0} &= \mathbb{C}_k - \overline{\mathbb{Q}}\mathbb{C}_k\mathbb{Q} - \mathbb{Q}\mathbb{C}_k\overline{\mathbb{Q}} + \sum_{n \geq 0} x_2^n \mathbb{Q}\mathbb{C}_{k+n+1}\overline{\mathbb{Q}} + \overline{\mathbb{Q}}\mathbb{C}_{k-1}\mathbb{Q} - x_2 \overline{\mathbb{Q}}\mathbb{C}_k\mathbb{Q},
 \end{aligned}$$

and the blocks \mathbb{B} and \mathbb{E} transform as

$$\begin{aligned}
 \tilde{\mathbb{B}}_0^{(x_2)} &= \mathbb{B}_0^{(x_2)} - \mathbb{Q}\mathbb{B}_0^{(x_2)} + \sum_{x_j \in S \setminus \{x_2\}} \sum_{n \geq 0} \frac{1}{(x_2 - x_j)^{n+1}} \mathbb{Q}\mathbb{B}_n^{(x_j)} + \sum_{n \geq 0} x_2^n \mathbb{Q}\mathbb{B}_n, \\
 \tilde{\mathbb{B}}_{k>0}^{(x_2)} &= \mathbb{B}_k^{(x_2)} - \mathbb{Q}\mathbb{B}_k^{(x_2)} + \mathbb{Q}\mathbb{B}_{k-1}^{(x_2)}, \\
 \tilde{\mathbb{B}}_k^{(x_j \neq x_2)} &= \mathbb{B}_k^{(x_j)} - \mathbb{Q}\mathbb{B}_k^{(x_j)} - \sum_{n \geq 0} \frac{1}{(x_2 - x_j)^{n+1}} \mathbb{Q}\mathbb{B}_{n+k}^{(x_j)}, \\
 \tilde{\mathbb{B}}_k &= \mathbb{B}_k - \mathbb{Q}\mathbb{B}_k + \sum_{n \geq 0} x_2^n \mathbb{Q}\mathbb{B}_{k+n+1}, \\
 \tilde{\mathbb{E}}_k^{(x_j)} &= \mathbb{E}_k^{(x_j)} - \mathbb{E}_k^{(x_j)}\mathbb{Q} + \mathbb{E}_{k+1}^{(x_j)}\mathbb{Q} + (x_j - x_2)\mathbb{E}_k^{(x_j)}\mathbb{Q}, \\
 \tilde{\mathbb{E}}_0 &= \mathbb{E}_0 - (1 + x_2)\mathbb{E}_0\mathbb{Q} + \sum_{x_j \in S} \mathbb{E}_0^{(x_j)}\mathbb{Q}, \\
 \tilde{\mathbb{E}}_{k>0} &= \mathbb{E}_k - (1 + x_2)\mathbb{E}_k\mathbb{Q} + \mathbb{E}_{k-1}\mathbb{Q}.
 \end{aligned}$$

D.2.2 Fuchsification for Off-Diagonal Blocks

In this appendix, we consider the transformation of the off-diagonal block \mathbb{B} to Fuchsian form. We assume that the diagonal blocks \mathbb{A} and \mathbb{C} are already in ϵ -form. The required transformation has the form

$$\mathbb{T} = \mathcal{L}(x_1, k, \mathbb{G}), \quad \mathbb{T}^{-1} = \mathcal{L}(x_1, k, -\mathbb{G}), \quad (\text{D.2})$$

where we used the definitions (3.32).

In addition to block \mathbb{B} , only block \mathbb{D} is influenced by this transformation, i.e. the blocks \mathbb{A} and \mathbb{C} are unaffected.

The transformation (3.25), with \mathbb{T} defined as in (D.2) translates to rules for the coefficient matrices. For $|x_1| < \infty$, we find

$$\tilde{\mathbb{B}}_k^{(x_1)} = \mathbb{B}_k^{(x_1)} + \mathbb{C}_0^{(x_1)} \widehat{\mathbb{G}} - \widehat{\mathbb{G}} \mathbb{A}_0^{(x_1)} + k \widehat{\mathbb{G}}, \quad (\text{D.3a})$$

$$\tilde{\mathbb{B}}_{n < k}^{(x_1)} = \mathbb{B}_n^{(x_1)} - \sum_{x_j \in S \setminus \{x_1\}} \frac{\mathbb{C}_0^{(x_j)} \widehat{\mathbb{G}} - \widehat{\mathbb{G}} \mathbb{A}_0^{(x_j)}}{(x_j - x_1)^{k-n}}, \quad (\text{D.3b})$$

$$\tilde{\mathbb{B}}_{n > k}^{(x_1)} = \mathbb{B}_n^{(x_1)}, \quad (\text{D.3c})$$

$$\tilde{\mathbb{B}}_0^{(x_j \neq x_1)} = \mathbb{B}_0^{(x_j)} + \frac{\mathbb{C}_0^{(x_j)} \widehat{\mathbb{G}} - \widehat{\mathbb{G}} \mathbb{A}_0^{(x_j)}}{(x_j - x_1)^k}, \quad (\text{D.3d})$$

$$\tilde{\mathbb{B}}_{n > 0}^{(x_j \neq x_1)} = \mathbb{B}_n^{(x_j)}, \quad (\text{D.3e})$$

$$\tilde{\mathbb{B}}_n = \mathbb{B}_n, \quad (\text{D.3f})$$

and

$$\begin{aligned} \tilde{\mathbb{D}}_{n < k}^{(x_1)} &= \mathbb{D}_n^{(x_1)} - \sum_{x_j \in S \setminus \{x_1\}} \sum_{i \geq 0} (-1)^i \frac{\binom{k+i-n-1}{i}}{(x_j - x_1)^{k+i-n}} \mathbb{E}_i^{(x_j)} \widehat{\mathbb{G}} \\ &\quad + \sum_{i \geq 0} x_1^i \binom{i+k-n-1}{i} \mathbb{E}_{i+k-n-1} \widehat{\mathbb{G}}, \\ \tilde{\mathbb{D}}_{n \geq k}^{(x_1)} &= \mathbb{D}_n^{(x_1)} + \mathbb{E}_{n-k}^{(x_1)} \widehat{\mathbb{G}}, \\ \tilde{\mathbb{D}}_{n \neq 1}^{(x_j)} &= \mathbb{D}_n^{(x_j)} + (-1)^k \sum_{i \geq 0} \frac{\binom{k+i-1}{i}}{(x_1 - x_j)^{k+i}} \mathbb{E}_{n+i}^{(x_j)} \widehat{\mathbb{G}}, \\ \tilde{\mathbb{D}}_n &= \mathbb{D}_n + \sum_{m \geq 0} (-1)^m \sum_{i \geq 0} x_1^{m+i} \binom{n+m}{n} \binom{i+n+m+k}{i} \mathbb{E}_{i+n+m+k} \widehat{\mathbb{G}}. \end{aligned}$$

The case $x_1 = \infty$ leads to

$$\begin{aligned} \tilde{\mathbb{B}}_{k-1} &= \mathbb{B}_{k-1} + \sum_{x_j \in S} \left[\mathbb{C}_0^{(x_j)} \widehat{\mathbb{G}} - \widehat{\mathbb{G}} \mathbb{A}^{(x_j)} \right] - k \widehat{\mathbb{G}}, \\ \tilde{\mathbb{B}}_{n < k-1} &= \mathbb{B}_n + \sum_{x_j \in S} x_j^{k-n-1} \left[\mathbb{C}_0^{(x_j)} \widehat{\mathbb{G}} - \widehat{\mathbb{G}} \mathbb{A}_0^{(x_j)} \right], \\ \tilde{\mathbb{B}}_{n > k-1} &= \mathbb{B}_n, \\ \tilde{\mathbb{B}}_0^{(x_j)} &= \mathbb{B}_0^{(x_j)} + x_j^k \left[\mathbb{C}_0^{(x_j)} \widehat{\mathbb{G}} - \widehat{\mathbb{G}} \mathbb{A}_0^{(x_j)} \right], \\ \tilde{\mathbb{B}}_{n > 0}^{(x_j)} &= \mathbb{B}_n^{(x_j)}, \end{aligned}$$

and

$$\begin{aligned}\tilde{\mathbb{D}}_n^{(x_j)} &= \mathbb{D}_n^{(x_j)} + \sum_{i=0}^k x_j^i \binom{k}{k-i} \mathbb{E}_{n+k-i} \widehat{\mathbb{G}}, \\ \tilde{\mathbb{D}}_{n < k} &= \mathbb{D}_n + \sum_{x_j \in S} \sum_{m=0}^{k-n-1} \sum_{i=0}^m (-1)^{k-n-m-1} x_j^{k-n-i-1} \binom{k-m-1}{n} \binom{k}{k+i-m} \mathbb{E}_i^{(x_j)} \widehat{\mathbb{G}}, \\ \tilde{\mathbb{D}}_{n \geq k} &= \mathbb{D}_n + \mathbb{E}_{n-k} \widehat{\mathbb{G}}.\end{aligned}$$

D.2.3 ϵ -Factorization

The transformation required in the ϵ -factorization is x -independent. Hence, every coefficient matrix in (3.27) transforms the same and the transformation rule (3.25) becomes a similarity transformation

$$\widetilde{\mathbb{M}}(x, \epsilon) = \mathbb{T}^{-1}(\epsilon) \mathbb{M}(x, \epsilon) \mathbb{T}(\epsilon).$$

The matrices $\mathbb{T}(\epsilon)$ and $\mathbb{T}^{-1}(\epsilon)$ have the form

$$\mathbb{T}(\epsilon) = \begin{pmatrix} \mathbb{1} & 0 & 0 \\ 0 & \widehat{\mathbb{T}}(\epsilon) & 0 \\ 0 & 0 & \mathbb{1} \end{pmatrix}, \quad \mathbb{T}^{-1}(\epsilon) = \begin{pmatrix} \mathbb{1} & 0 & 0 \\ 0 & \widehat{\mathbb{T}}^{-1}(\epsilon) & 0 \\ 0 & 0 & \mathbb{1} \end{pmatrix},$$

with $\widehat{\mathbb{T}}(\epsilon)$ and $\widehat{\mathbb{T}}^{-1}(\epsilon)$ corresponding to block \mathbb{C} . Using the block-triangular structure (3.30) yields

$$\begin{pmatrix} \widetilde{\mathbb{A}}(x, \epsilon) & 0 & 0 \\ \widetilde{\mathbb{B}}(x, \epsilon) & \widetilde{\mathbb{C}}(x, \epsilon) & 0 \\ \widetilde{\mathbb{D}}(x, \epsilon) & \widetilde{\mathbb{E}}(x, \epsilon) & \widetilde{\mathbb{F}}(x, \epsilon) \end{pmatrix} = \begin{pmatrix} \mathbb{A}(x, \epsilon) & 0 & 0 \\ \mathbb{T}^{-1}(\epsilon) \mathbb{B}(x, \epsilon) & \mathbb{T}^{-1}(\epsilon) \mathbb{C}(x, \epsilon) \mathbb{T}(\epsilon) & 0 \\ \mathbb{D}(x, \epsilon) & \mathbb{E}(x, \epsilon) \mathbb{T}(\epsilon) & \mathbb{F}(x, \epsilon) \end{pmatrix}.$$

Thus, besides block \mathbb{C} , block \mathbb{B} and block \mathbb{E} are influenced as well.

Common Subexpressions E

In this appendix, we present a possible algorithm to find common subexpressions in a given list of polynomials. Such an algorithm is necessary for the optimization procedure in the modified Method of Brackets, which is described in Section 3.7. The recursive algorithm shown in alg. E.1 is far from being optimal but it proved nevertheless successful for all our tests.

The main function of the algorithm is `COMMONBINOMIALS` starting at line 46. The argument P is an array of polynomials. Polynomials are in turn represented as arrays of terms (monomials). Arrays all start at index one. The function `COMMONBINOMIALS` should be called with $p_0 = t_0 = 1$ and an empty set r . The best optimization found by the algorithm is returned in the global variables \hat{P} and \hat{r} , where \hat{r} is a set of rules that leads back to P , when it is repeatedly applied to the array of polynomials \hat{P} .

The quality of an optimization is measured by the rank ρ calculated by the function `CALCRANK` starting at line 6. The rank ρ minus the number of propagators gives the number of MB integrals in the result. The goal of the algorithm is, therefore, to find an optimization, where ρ is minimal.

The first part (lines 48 to 78) of the algorithm fills an array J with all possible optimizations that can be performed at the current level of the recursion. In this step, we scan for binomials appearing in P more than once. The actual scan starts at term t_0 in polynomial p_0 . These arguments to the function `COMMONBINOMIALS` are used to prevent scans of regions already completed at a lower recursion-level. Lines 55 and 70 find all occurrences of the binomial b in all polynomials in P . These occurrences are stored as triplets (p, t_1, t_2) in the set B , where the first term of the binomial is found at $P[p, t_1]$ and the second term at $P[p, t_2]$, and $t_2 > t_1$.

If the polynomials in P do not have common binomials anymore, J is empty at line 80. In that case the optimization is complete. If the rank ρ of this optimization is the lowest so far, the optimization is stored in the global variables.

If J is not empty, there are still common binomials in P . In principle, we could now try out all optimizations in J one-by-one and then recursively call `COMMONBINOMIALS`. Unfortunately, for large polynomials the algorithm would not terminate in a feasible time. For that reason, we only try out the first N optimizations with the largest number of common binomials. For a finite $N < \infty$, it is therefore not guaranteed that the algorithm finds the best possible optimization. In most test cases even very small values of N , e.g. 3 or 4, were sufficient to find very good optimizations in only a few seconds.

The lines 87 and 91 cause an early exit, if the optimization at the current state is, even in the best-case scenario, not capable of producing a final optimization with a new minimum rank.

The function `COMMONBINOMIALS` only returns rules with *binomials* on the right-hand side. In case, a symbol introduced by the algorithm does not appear in the returned list of optimized polynomials and only once on the right-hand side of one rule, it can be back-substituted without changing the rank of the optimization. This back-substitution leads then to new rules where the right-hand sides have more than two terms.

Algorithm E.1 Algorithm to find common binomials in a list of polynomials

```
1: global variables
2:    $\hat{\rho} \leftarrow \infty$ 
3:    $\hat{P} \leftarrow ()$ 
4:    $\hat{r} \leftarrow \{\}$ 
5: end global variables

6: function CALCRANK( $P, r$ )
7:   in:    $P$ : an array of polynomials
            $r$ : a set of rules
8:   out: an integer rank of the optimization
9:    $\rho \leftarrow |r| - 1$ 
10:  for all  $p \in P$  do
11:     $\rho \leftarrow \rho + |p|$ 
12:  end for
13:  return  $\rho$ 
14: end function

15: function REPLACEBINOMIALS( $P, B$ )  $\rightarrow (P', r, p_0, t_0)$ 
16:  in:    $P$ : an array of polynomials
            $B$ : a set of triplets  $(p, t_1, t_2)$ 
17:  out:  $P'$ : an array of polynomials
            $r$ : a rule
            $p_0$ : an index to a polynomial in  $P'$ 
            $t_0$ : an index to a term in  $P'[p_0]$ 

18:   $(p_0, t_0) \leftarrow \text{undef}$ 
19:   $P' \leftarrow$  an array with  $|P|$  empty elements
20:   $m_1 \leftarrow P[B[1, 1], B[1, 2]]$ 
21:   $m_2 \leftarrow P[B[1, 1], B[1, 3]]$ 
22:   $b \leftarrow \frac{m_1 + m_2}{\text{gcd}(m_1, m_2)}$ 
23:   $\xi \leftarrow$  a new symbol name
24:   $r \leftarrow$  the rule " $\xi \rightarrow b$ "
25:   $D \leftarrow \{\}$ 
26:  for  $p \leftarrow 1$  to  $|P|$  do
27:    for  $t \leftarrow 1$  to  $|P[p]|$  do
28:      if  $(p, t) \notin D$  then
29:        if  $\exists (p, t_1, t_2) \in B : t = t_1 \vee t = t_2$  then
30:           $m_1 \leftarrow P[p, t_1]$ 
31:           $m_2 \leftarrow P[p, t_2]$ 
32:          append  $\xi \cdot \text{gcd}(m_1, m_2)$  to  $P'[p]$ 
33:          if  $(p_0, t_0) = \text{undef}$  then
34:             $(p_0, t_0) \leftarrow (p, |P'[p]|)$ 
35:          end if
36:          end if
37:           $D \leftarrow D \cup \{(p, t_1), (p, t_2)\}$ 
38:        else
39:          append  $P[p, t]$  to  $P'[p]$ 
40:           $D \leftarrow D \cup \{(p, t)\}$ 
41:        end if
42:      end if
43:    end for
```

```

44:   return ( $P', r, p_0, t_0$ )
45: end function

46: procedure COMMONBINOMIALS( $P, r, p_0, t_0, N$ )
47:   in:   $P$ : an array of polynomials
         $r$ : a set of rules
         $p_0$ : an index to a polynomial in  $P$ 
         $t_0$ : an index to a term in  $P[p_0]$ 
         $N$ : an integer
48:    $D \leftarrow \{\}$ 
49:    $J \leftarrow ()$ 
50:   for  $t \leftarrow 1$  to  $t_0 - 1$  do
51:     if  $(p_0, t, t_0) \notin D$  then
52:        $m_1 \leftarrow P[p_0, t]$ 
53:        $m_2 \leftarrow P[p_0, t_0]$ 
54:        $b \leftarrow \frac{m_1 + m_2}{\gcd(m_1, m_2)}$ 
55:        $B \leftarrow$  set of all occurrences of binomial  $b$  in all polynomials in  $P$ 
56:        $D \leftarrow D \cup B$ 
57:       if  $|B| > 1$  then
58:         add  $B$  to  $J$ 
59:       end if
60:     end if
61:   end for
62:   for  $p_1 \leftarrow p_0$  to  $|P|$  do
63:      $t'_0 = \begin{cases} t_0 & \text{if } p_1 = p_0 \\ 1 & \text{else} \end{cases}$ 
64:     for  $t_1 \leftarrow t'_0$  to  $|P[p_1]|$  do
65:       for  $t_2 \leftarrow t_1 + 1$  to  $|P[p_1]|$  do
66:         if  $(p_1, t_1, t_2) \notin D$  then
67:            $m_1 \leftarrow P[p_1, t_1]$ 
68:            $m_2 \leftarrow P[p_1, t_2]$ 
69:            $b \leftarrow \frac{m_1 + m_2}{\gcd(m_1, m_2)}$ 
70:            $B \leftarrow$  set of all occurrences of binomial  $b$  in all polynomials in  $P$ 
71:            $D \leftarrow D \cup B$ 
72:           if  $|B| > 1$  then
73:             add  $B$  to  $J$ 
74:           end if
75:         end if
76:       end for
77:     end for
78:   end for
79:    $\rho \leftarrow \text{CALCRANK}(P, r)$ 
80:   if  $|J| = 0$  then
81:     if  $\rho < \hat{\rho}$  then
82:        $\hat{\rho} \leftarrow \rho$ 
83:        $\hat{P} \leftarrow P$ 
84:        $\hat{r} \leftarrow r$ 
85:     end if
86:   else
87:      $\delta \leftarrow 0$ 
88:     for all  $j \in J$  do

```

```

89:          $\delta \leftarrow \delta + |j| - 1$ 
90:     end for
91:     if  $\rho - \delta < \hat{\rho}$  then
92:         sort  $J$  by the length of the elements. Longest element first.
93:         if  $|J| > N$  then
94:             resize  $J$  to length  $N$ 
95:         end if
96:         for all  $j \in J$  do
97:              $(P', r', p'_0, t'_0) \leftarrow \text{REPLACEBINOMIALS}(P, j)$ 
98:             call  $\text{COMMONBINOMIAL}(P', r \cup \{r'\}, p'_0, t'_0, N)$ 
99:         end for
100:     end if
101: end if
102: end procedure

```

Bibliography

- [1] M. Prausa, *epsilon: A tool to find a canonical basis of master integrals*, *Comput. Phys. Commun.* **219** (2017) 361–376, [arXiv:1701.00725 \[hep-ph\]](#).
- [2] M. Prausa, *Mellin-Barnes meets Method of Brackets: a novel approach to Mellin-Barnes representations of Feynman integrals*, *Eur. Phys. J.* **C77** no. 9, (2017) 594, [arXiv:1706.09852 \[hep-ph\]](#).
- [3] S. L. Glashow, *Partial-symmetries of weak interactions*, *Nucl. Phys.* **22** (1961) 579–588.
- [4] S. Weinberg, *A Model of Leptons*, *Phys. Rev. Lett.* **19** (1967) 1264–1266.
- [5] A. Salam, *Weak and Electromagnetic Interactions*, *Conf. Proc.* **C680519** (1968) 367–377.
- [6] **UA1** Collaboration, G. Arnison *et al.*, *Experimental Observation of Isolated Large Transverse Energy Electrons with Associated Missing Energy at $\sqrt{s} = 540$ GeV*, *Phys. Lett.* **122B** (1983) 103–116.
- [7] **UA1** Collaboration, G. Arnison *et al.*, *Experimental Observation of Lepton Pairs of Invariant Mass Around 95 GeV/c² at the CERN SPS Collider*, *Phys. Lett.* **126B** (1983) 398–410.
- [8] **UA2** Collaboration, M. Banner *et al.*, *Observation of Single Isolated Electrons of High Transverse Momentum in Events with Missing Transverse Energy at the CERN $\bar{p}p$ Collider*, *Phys. Lett.* **122B** (1983) 476–485.
- [9] **UA2** Collaboration, P. Bagnaia *et al.*, *Evidence for $Z^0 \rightarrow e^+e^-$ at the CERN $\bar{p}p$ Collider*, *Phys. Lett.* **129B** (1983) 130–140.
- [10] Y. Ne’eman, *Derivation of Strong Interactions from a Gauge Invariance*, *Nucl. Phys.* **26** (1961) 222–229.
- [11] M. Gell-Mann, *The Eightfold Way: A Theory of strong interaction symmetry*, preprint CTSL-20, unpublished.
- [12] M. Gell-Mann, *A Schematic Model of Baryons and Mesons*, *Phys. Lett.* **8** (1964) 214–215.
- [13] G. Zweig, *An SU(3) model for strong interaction symmetry and its breaking. Version 1*, preprint CERN-TH-401, unpublished.
- [14] G. Zweig, *An SU(3) model for strong interaction symmetry and its breaking. Version 2*, in *DEVELOPMENTS IN THE QUARK THEORY OF HADRONS. VOL. 1. 1964 - 1978*, D. Lichtenberg and S. P. Rosen, eds., pp. 22–101. 1964.

-
- [15] V. E. Barnes *et al.*, *Observation of a Hyperon with Strangeness Minus Three*, *Phys. Rev. Lett.* **12** (1964) 204–206.
- [16] H. Fritzsch and M. Gell-Mann, *Current algebra: Quarks and What Else?*, *eConf C720906V2* (1972) 135–165, [arXiv:hep-ph/0208010](#) [[hep-ph](#)].
- [17] D. J. Gross and F. Wilczek, *Ultraviolet Behavior of Non-Abelian Gauge Theories*, *Phys. Rev. Lett.* **30** (1973) 1343–1346.
- [18] H. D. Politzer, *Reliable Perturbative Results for Strong Interactions?*, *Phys. Rev. Lett.* **30** (1973) 1346–1349.
- [19] F. Englert and R. Brout, *Broken Symmetry and the Mass of Gauge Vector Mesons*, *Phys. Rev. Lett.* **13** (1964) 321–323.
- [20] P. W. Higgs, *Broken Symmetries and the Masses of Gauge Bosons*, *Phys. Rev. Lett.* **13** (1964) 508–509.
- [21] G. S. Guralnik, C. R. Hagen, and T. W. B. Kibble, *Global Conservation Laws and Massless Particles*, *Phys. Rev. Lett.* **13** (1964) 585–587.
- [22] G. 't Hooft, *Renormalizable Lagrangians for Massive Yang-Mills Fields*, *Nucl. Phys.* **B35** (1971) 167–188.
- [23] **ATLAS** Collaboration, G. Aad *et al.*, *Combined search for the Standard Model Higgs boson using up to 4.9 fb^{-1} of pp collision data at $\sqrt{s} = 7 \text{ TeV}$ with the ATLAS detector at the LHC*, *Phys. Lett.* **B710** (2012) 49–66, [arXiv:1202.1408](#) [[hep-ex](#)].
- [24] **CMS** Collaboration, S. Chatrchyan *et al.*, *Combined results of searches for the standard model Higgs boson in pp collisions at $\sqrt{s} = 7 \text{ TeV}$* , *Phys. Lett.* **B710** (2012) 26–48, [arXiv:1202.1488](#) [[hep-ex](#)].
- [25] **LHC Higgs Cross Section Working Group** Collaboration, D. de Florian *et al.*, *Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector*, [arXiv:1610.07922](#) [[hep-ph](#)].
- [26] H. M. Georgi, S. L. Glashow, M. E. Machacek, and D. V. Nanopoulos, *Higgs Bosons from Two Gluon Annihilation in Proton Proton Collisions*, *Phys. Rev. Lett.* **40** (1978) 692.
- [27] T. G. Rizzo, *Gluon final states in Higgs-boson Decay*, *Phys. Rev.* **D22** (1980) 178. [Addendum: *Phys. Rev.* **D22**, 1824 (1980)].
- [28] A. Djouadi, M. Spira, and P. M. Zerwas, *Production of Higgs bosons in proton colliders: QCD corrections*, *Phys. Lett.* **B264** (1991) 440–446.
- [29] S. Dawson, *Radiative corrections to Higgs boson production*, *Nucl. Phys.* **B359** (1991) 283–300.
- [30] D. Graudenz, M. Spira, and P. M. Zerwas, *QCD corrections to Higgs-boson production at proton-proton colliders*, *Phys. Rev. Lett.* **70** (1993) 1372–1375.
- [31] M. Spira, A. Djouadi, D. Graudenz, and P. M. Zerwas, *Higgs boson production at the LHC*, *Nucl. Phys.* **B453** (1995) 17–82, [arXiv:hep-ph/9504378](#) [[hep-ph](#)].
- [32] R. Harlander and P. Kant, *Higgs production and decay: analytic results at next-to-leading order QCD*, *JHEP* **12** (2005) 015, [arXiv:hep-ph/0509189](#) [[hep-ph](#)].
- [33] 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, [arXiv:hep-ph/0611236](#) [[hep-ph](#)].

-
- [34] S. Beerli, *A New Method for Evaluating Two-Loop Feynman Integrals and its Application To Higgs Production*, PhD thesis, Zurich, ETH, 2008.
- [35] U. Aglietti, R. Bonciani, G. Degrossi, and A. Vicini, *Analytic results for virtual QCD corrections to Higgs production and decay*, *JHEP* **01** (2007) 021, [arXiv:hep-ph/0611266 \[hep-ph\]](#).
- [36] C. Anastasiou and K. Melnikov, *Higgs boson production at hadron colliders in NNLO QCD*, *Nucl. Phys.* **B646** (2002) 220–256, [arXiv:hep-ph/0207004 \[hep-ph\]](#).
- [37] R. V. Harlander and W. B. Kilgore, *Next-to-Next-to-Leading Order Higgs Production at Hadron Colliders*, *Phys. Rev. Lett.* **88** (2002) 201801, [arXiv:hep-ph/0201206 \[hep-ph\]](#).
- [38] V. Ravindran, J. Smith, and W. L. van Neerven, *NNLO corrections to the total cross section for Higgs boson production in hadron-hadron collisions*, *Nucl. Phys.* **B665** (2003) 325–366, [arXiv:hep-ph/0302135 \[hep-ph\]](#).
- [39] C. Anastasiou, C. Duhr, F. Dulat, F. Herzog, and B. Mistlberger, *Higgs Boson Gluon-Fusion Production in QCD at Three Loops*, *Phys. Rev. Lett.* **114** (2015) 212001, [arXiv:1503.06056 \[hep-ph\]](#).
- [40] C. Anastasiou, C. Duhr, F. Dulat, E. Furlan, T. Gehrmann, F. Herzog, A. Lazopoulos, and B. Mistlberger, *High precision determination of the gluon fusion Higgs boson cross-section at the LHC*, *JHEP* **05** (2016) 058, [arXiv:1602.00695 \[hep-ph\]](#).
- [41] **Particle Data Group** Collaboration, C. Patrignani *et al.*, *Review of Particle Physics*, *Chin. Phys.* **C40** no. 10, (2016) 100001.
- [42] J. R. Ellis, M. K. Gaillard, and D. V. Nanopoulos, *A phenomenological profile of the Higgs Boson*, *Nucl. Phys.* **B106** (1976) 292.
- [43] H.-Q. Zheng and D.-D. Wu, *First-order QCD corrections to the decay of the Higgs boson into two photons*, *Phys. Rev.* **D42** (1990) 3760–3763.
- [44] A. Djouadi, M. Spira, J. J. van der Bij, and P. M. Zerwas, *QCD corrections to $\gamma\gamma$ decays of Higgs particles in the intermediate mass range*, *Phys. Lett.* **B257** (1991) 187–190.
- [45] S. Dawson and R. P. Kauffman, *QCD corrections to $H \rightarrow \gamma\gamma$* , *Phys. Rev.* **D47** (1993) 1264–1267.
- [46] J. Fleischer, O. V. Tarasov, and V. O. Tarasov, *Analytical result for the two-loop QCD correction to the decay $H \rightarrow 2\gamma$* , *Phys. Lett.* **B584** (2004) 294–297, [arXiv:hep-ph/0401090 \[hep-ph\]](#).
- [47] M. Steinhauser, *Corrections of $\mathcal{O}(\alpha_s^2)$ to the Decay of an Intermediate-Mass Higgs Boson into Two Photons*, in *Tegernsee 1996, The Higgs puzzle*, pp. 177–185. [arXiv:hep-ph/9612395 \[hep-ph\]](#).
- [48] P. Maierhöfer and P. Marquard, *Complete three-loop QCD corrections to the decay $H \rightarrow \gamma\gamma$* , *Phys. Lett.* **B721** (2013) 131–135, [arXiv:1212.6233 \[hep-ph\]](#).
- [49] P. Nogueira, *Automatic Feynman Graph Generation*, *J.Comput.Phys.* **105** (1993) 279–289.
- [50] R. Harlander, T. Seidensticker, and M. Steinhauser, *Corrections of $\mathcal{O}(\alpha\alpha_s)$ to the decay of the Z boson into bottom quarks*, *Phys.Lett.* **B426** (1998) 125–132, [arXiv:hep-ph/9712228 \[hep-ph\]](#).
- [51] T. Seidensticker, *Automatic application of successive asymptotic expansions of Feynman diagrams*, [arXiv:hep-ph/9905298 \[hep-ph\]](#).

-
- [52] J. A. M. Vermaseren, *New features of FORM*, [arXiv:math-ph/0010025](#) [math-ph].
- [53] M. E. Peskin and D. V. Schroeder, *An Introduction to Quantum Field Theory*. Addison-Wesley Publishing Company, 1995.
- [54] S. G. Gorishnii, *Construction of operator expansions and effective theories in the \overline{MS} scheme*, *Nucl. Phys.* **B319** (1989) 633–666.
- [55] V. A. Smirnov, *Asymptotic expansions in limits of large momenta and masses*, *Commun. Math. Phys.* **134** (1990) 109–137.
- [56] V. A. Smirnov, *Asymptotic Expansions in Momenta and Masses and Calculation of Feynman Diagrams*, *Mod. Phys. Lett.* **A10** (1995) 1485–1500, [arXiv:hep-th/9412063](#) [hep-th].
- [57] V. A. Smirnov, *Applied Asymptotic Expansions in Momenta and Masses*, *Springer Tracts Mod. Phys.* **177** (2002) 1–262.
- [58] S. Larin, F. Tkachov, and J. Vermaseren, *The FORM version of MINCER*, *NIKHEF-H/91-18* (1991).
- [59] M. Steinhauser, *MATAD: a program package for the computation of MAssive TADpoles*, *Comput.Phys.Commun.* **134** (2001) 335–364, [arXiv:hep-ph/0009029](#) [hep-ph].
- [60] T. van Ritbergen, A. Schellekens, and J. Vermaseren, *Group theory factors for Feynman diagrams*, *Int.J.Mod.Phys.* **A14** (1999) 41–96, [arXiv:hep-ph/9802376](#) [hep-ph].
- [61] M. A. Shifman, A. I. Vainshtein, M. B. Voloshin, and V. I. Zakharov, *Low-Energy Theorems for Higgs Boson Couplings to Photons*, *Sov. J. Nucl. Phys.* **30** (1979) 711–716. [*Yad. Fiz.*30,1368(1979)].
- [62] J. Fleischer and O. V. Tarasov, *Calculation of Feynman diagrams from their small momentum expansion*, *Z. Phys.* **C64** (1994) 413–426, [arXiv:hep-ph/9403230](#) [hep-ph].
- [63] R. N. Lee, *Reducing differential equations for multiloop master integrals*, *JHEP* **04** (2015) 108, [arXiv:1411.0911](#) [hep-ph].
- [64] R. Bonciani, P. Mastrolia, and E. Remiddi, *Master integrals for the 2-loop QCD virtual corrections to the forward-backward asymmetry*, *Nucl. Phys.* **B690** (2004) 138–176, [arXiv:hep-ph/0311145](#) [hep-ph].
- [65] R. N. Lee and A. A. Pomeransky, *Normalized Fuchsian form on Riemann sphere and differential equations for multiloop integrals*, [arXiv:1707.07856](#) [hep-th].
- [66] R. H. Lewis, “Computer Algebra System Fermat.” <https://home.bway.net/lewis>. Accessed: 2017-09-23.
- [67] F. V. Tkachov, *A theorem on analytical calculability of 4-loop renormalization group functions*, *Phys. Lett.* **100B** (1981) 65–68.
- [68] K. G. Chetyrkin and F. V. Tkachov, *Integration by parts: The algorithm to calculate β -functions in 4 loops*, *Nucl. Phys.* **B192** (1981) 159–204.
- [69] T. Gehrmann and E. Remiddi, *Differential equations for two-loop four-point functions*, *Nucl. Phys.* **B580** (2000) 485–518, [arXiv:hep-ph/9912329](#) [hep-ph].
- [70] R. N. Lee, *Group structure of the integration-by-part identities and its application to the reduction of multiloop integrals*, *JHEP* **07** (2008) 031, [arXiv:0804.3008](#) [hep-ph].
- [71] A. V. Smirnov and A. V. Petukhov, *The Number of Master Integrals is Finite*, *Lett. Math. Phys.* **97** (2011) 37–44, [arXiv:1004.4199](#) [hep-th].

-
- [72] B. Ruijl, T. Ueda, and J. A. M. Vermaseren, *Forcer, a FORM program for the parametric reduction of four-loop massless propagator diagrams*, [arXiv:1704.06650 \[hep-ph\]](#).
- [73] R. N. Lee, *Presenting LiteRed: a tool for the Loop InTEgrals REDuction*, [arXiv:1212.2685 \[hep-ph\]](#).
- [74] R. N. Lee, *LiteRed 1.4: a powerful tool for reduction of multiloop integrals*, *J. Phys. Conf. Ser.* **523** (2014) 012059, [arXiv:1310.1145 \[hep-ph\]](#).
- [75] S. Laporta, *High-precision calculation of multiloop Feynman integrals by difference equations*, *Int. J. Mod. Phys. A* **15** (2000) 5087–5159, [arXiv:hep-ph/0102033 \[hep-ph\]](#).
- [76] C. Anastasiou and A. Lazopoulos, *Automatic integral reduction for higher order perturbative calculations*, *JHEP* **07** (2004) 046, [arXiv:hep-ph/0404258 \[hep-ph\]](#).
- [77] A. V. Smirnov, *Algorithm FIRE – Feynman Integral REDuction*, *JHEP* **10** (2008) 107, [arXiv:0807.3243 \[hep-ph\]](#).
- [78] A. V. Smirnov and V. A. Smirnov, *FIRE4, LiteRed and accompanying tools to solve integration by parts relations*, *Comput. Phys. Commun.* **184** (2013) 2820–2827, [arXiv:1302.5885 \[hep-ph\]](#).
- [79] A. V. Smirnov, *FIRE5: A C++ implementation of Feynman Integral REDuction*, *Comput. Phys. Commun.* **189** (2015) 182–191, [arXiv:1408.2372 \[hep-ph\]](#).
- [80] C. Studerus, *Reduze – Feynman integral reduction in C++*, *Comput. Phys. Commun.* **181** (2010) 1293–1300, [arXiv:0912.2546 \[physics.comp-ph\]](#).
- [81] A. von Manteuffel and C. Studerus, *Reduze 2 - Distributed Feynman Integral Reduction*, [arXiv:1201.4330 \[hep-ph\]](#).
- [82] P. Maierhofer, J. Usovitsch, and P. Uwer, *Kira - A Feynman Integral Reduction Program*, [arXiv:1705.05610 \[hep-ph\]](#).
- [83] “SQLite.” <http://www.sqlite.org>. Accessed: 2017-09-23.
- [84] Wolfram Research, Inc., “Mathematica, Version 10.0.” Champaign, IL, 2014.
- [85] V. A. Smirnov, *Analytic Tools for Feynman Integrals*, *Springer Tracts Mod. Phys.* **250** (2012) 1–296.
- [86] A. V. Kotikov, *Differential equations method. New technique for massive Feynman diagram calculation*, *Phys. Lett.* **B254** (1991) 158–164.
- [87] A. V. Kotikov, *Differential equations method: the Calculation of vertex-type Feynman diagrams*, *Phys. Lett.* **B259** (1991) 314–322.
- [88] A. V. Kotikov, *Differential equation method. The calculation of N-point Feynman diagrams*, *Phys. Lett.* **B267** (1991) 123–127. [Erratum: *Phys. Lett.* **B295** (1992) 409].
- [89] V. A. Smirnov, *Analytical result for dimensionally regularized massless on-shell double box*, *Phys. Lett.* **B460** (1999) 397–404, [arXiv:hep-ph/9905323 \[hep-ph\]](#).
- [90] J. B. Tausk, *Non-planar massless two-loop Feynman diagrams with four on-shell legs*, *Phys. Lett.* **B469** (1999) 225–234, [arXiv:hep-ph/9909506 \[hep-ph\]](#).
- [91] T. Binoth and G. Heinrich, *An automatized algorithm to compute infrared divergent multi-loop integrals*, *Nucl. Phys.* **B585** (2000) 741–759, [arXiv:hep-ph/0004013 \[hep-ph\]](#).

-
- [92] T. Binoth and G. Heinrich, *Numerical evaluation of multi-loop integrals by sector decomposition*, *Nucl. Phys.* **B680** (2004) 375–388, [arXiv:hep-ph/0305234](#) [hep-ph].
- [93] T. Binoth and G. Heinrich, *Numerical evaluation of phase space integrals by sector decomposition*, *Nucl. Phys.* **B693** (2004) 134–148, [arXiv:hep-ph/0402265](#) [hep-ph].
- [94] E. Kummer, *Ueber die Transcendenten, welche aus wiederholten Integrationen rationaler Formeln entstehen.*, *J. Reine Angew. Math.* **21** (1840) 74–90.
- [95] H. Poincaré, *Sur les groupes des équations linéaires*, *Acta Math.* **4** (1884) 201–312.
- [96] K.-T. Chen, *Iterated path integrals*, *Bull. Am. Math. Soc.* **83** (1977) 831–879.
- [97] A. B. Goncharov, *Multiple polylogarithms, cyclotomy and modular complexes*, *Math. Res. Lett.* **5** (1998) 497–516, [arXiv:1105.2076](#) [math.AG].
- [98] A. B. Goncharov, M. Spradlin, C. Vergu, and A. Volovich, *Classical Polylogarithms for Amplitudes and Wilson Loops*, *Phys. Rev. Lett.* **105** (2010) 151605, [arXiv:1006.5703](#) [hep-th].
- [99] E. Remiddi and J. A. M. Vermaseren, *Harmonic Polylogarithms*, *Int. J. Mod. Phys.* **A15** (2000) 725–754, [arXiv:hep-ph/9905237](#) [hep-ph].
- [100] J. M. Borwein, D. M. Bradley, D. J. Broadhurst, and P. Lisonek, *Special values of multiple polylogarithms*, *Trans. Am. Math. Soc.* **353** (2001) 907–941, [arXiv:math/9910045](#) [math-ca].
- [101] D. Maître, *HPL, a Mathematica implementation of the harmonic polylogarithms*, *Comput. Phys. Commun.* **174** (2006) 222–240, [arXiv:hep-ph/0507152](#) [hep-ph].
- [102] J. M. Henn, *Multiloop integrals in dimensional regularization made simple*, *Phys. Rev. Lett.* **110** (2013) 251601, [arXiv:1304.1806](#) [hep-th].
- [103] O. Gituliar and V. Magerya, *Fuchsian and master integrals for splitting functions from differential equations in QCD*, *PoS LL2016* (2016) 030, [arXiv:1607.00759](#) [hep-ph].
- [104] O. Gituliar and V. Magerya, *Fuchsian: A tool for reducing differential equations for Feynman master integrals to epsilon form*, *Comput. Phys. Commun.* **219** (2017) 329–338, [arXiv:1701.04269](#) [hep-ph].
- [105] J. Moser, *The order of a singularity in Fuchs’ Theory*, *Math. Z.* **72** (1959/60) 379–398.
- [106] M. Beneke and V. A. Smirnov, *Asymptotic expansion of Feynman integrals near threshold*, *Nucl. Phys.* **B522** (1998) 321–344, [arXiv:hep-ph/9711391](#) [hep-ph].
- [107] V. A. Smirnov and E. R. Rakhmetov, *The regional strategy in the asymptotic expansion of two-loop vertex Feynman diagrams*, *Theor. Math. Phys.* **120** (1999) 870–875, [arXiv:hep-ph/9812529](#) [hep-ph].
- [108] V. A. Smirnov, *Problems of the strategy of regions*, *Phys. Lett.* **B465** (1999) 226–234, [arXiv:hep-ph/9907471](#) [hep-ph].
- [109] C. Bogner and S. Weinzierl, *Feynman graph polynomials*, *Int. J. Mod. Phys.* **A25** (2010) 2585–2618, [arXiv:1002.3458](#) [hep-ph].
- [110] G. C. Wick, *Properties of Bethe-Salpeter Wave Functions*, *Phys. Rev.* **96** (1954) 1124–1134.
- [111] J. Gluza, K. Kajda, and T. Riemann, *AMBRE – a Mathematica package for the construction of Mellin-Barnes representations for Feynman integrals*, *Comput. Phys. Commun.* **177** (2007) 879–893, [arXiv:0704.2423](#) [hep-ph].

-
- [112] J. Gluza, K. Kajda, T. Riemann, and V. Yundin, *Numerical evaluation of tensor Feynman integrals in Euclidean kinematics*, *Eur. Phys. J.* **C71** (2011) 1516, [arXiv:1010.1667 \[hep-ph\]](#).
- [113] J. Blümlein, I. Dubovyk, J. Gluza, M. Ochman, C. G. Raab, T. Riemann, and C. Schneider, *Non-planar Feynman integrals, Mellin-Barnes representations, multiple sums*, *PoS LL2014* (2014) 052, [arXiv:1407.7832 \[hep-ph\]](#).
- [114] I. Dubovyk, J. Gluza, T. Riemann, and J. Usovitsch, *Numerical integration of massive two-loop Mellin-Barnes integrals in Minkowskian regions*, *PoS LL2016* (2016) 034, [arXiv:1607.07538 \[hep-ph\]](#).
- [115] I. González and I. Schmidt, *Optimized negative dimensional integration method (NDIM) and multiloop Feynman diagram calculation*, *Nucl. Phys.* **B769** (2007) 124–173, [arXiv:hep-th/0702218 \[hep-th\]](#).
- [116] I. González and V. H. Moll, *Definite integrals by the method of brackets. Part 1*, *Adv. Appl. Math.* **45** no. 1, (2010) 50 – 73, [arXiv:0812.3356 \[math-ph\]](#).
- [117] I. González, *Method of Brackets and Feynman diagrams evaluation*, *Nucl. Phys. Proc. Suppl.* **205-206** (2010) 141–146, [arXiv:1008.2148 \[hep-th\]](#).
- [118] M. Czakon, *Automatized analytic continuation of Mellin-Barnes integrals*, *Comput. Phys. Commun.* **175** (2006) 559–571, [arXiv:hep-ph/0511200 \[hep-ph\]](#).
- [119] A. V. Smirnov and V. A. Smirnov, *On the resolution of singularities of multiple Mellin-Barnes Integrals*, *Eur. Phys. J.* **C62** (2009) 445–449, [arXiv:0901.0386 \[hep-ph\]](#).
- [120] R. Storn, *On the Usage of Differential Evolution for Function Optimization*, in *Proceedings of North American Fuzzy Information Processing*, pp. 519–523. Jun, 1996.
- [121] R. Storn and K. Price, *Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces*, *J. Global Optim.* **11** no. 4, (Dec, 1997) 341–359.
- [122] M. Czakon, “MBasymptotics.m.” <https://mbtools.hepforge.org>. Accessed: 2017-09-23.
- [123] I. G. Halliday and R. M. Ricotta, *Negative dimensional integrals. I. Feynman graphs*, *Phys. Lett.* **B193** (1987) 241–246.
- [124] G. Hardy, *Ramanujan: Twelve Lectures on Subjects Suggested by His Life and Work*. AMS Chelsea Publishing Series. AMS Chelsea Pub., 1999.
- [125] R. N. Lee and V. A. Smirnov, *Evaluating the last missing ingredient for the three-loop quark static potential by differential equations*, *JHEP* **10** (2016) 089, [arXiv:1608.02605 \[hep-ph\]](#).
- [126] E. Mendels, *Feynman Diagrams without Feynman Parameters*, *Nuovo Cim.* **A45** (1978) 87–122.
- [127] E. Mendels, *Structure conserving parametrization of Feynman diagrams*, *J. Math. Phys.* **43** (2002) 3011–3035.
- [128] F. A. Berends, M. Böhm, M. Buza, and R. Scharf, *Closed expressions for specific massive multiloop self-energy integrals*, *Z. Phys.* **C63** (1994) 227–234.
- [129] L. Adams, C. Bogner, and S. Weinzierl, *The two-loop sunrise graph with arbitrary masses*, *J. Math. Phys.* **54** (2013) 052303, [arXiv:1302.7004 \[hep-ph\]](#).
- [130] S. Bloch, M. Kerr, and P. Vanhove, *A Feynman integral via higher normal functions*, *Compos. Math.* **151** (2015) 2329–2375, [arXiv:1406.2664 \[hep-th\]](#).

-
- [131] L. Adams, C. Bogner, and S. Weinzierl, *The iterated structure of the all-order result for the two-loop sunrise integral*, *J. Math. Phys.* **57** no. 3, (2016) 032304, [arXiv:1512.05630 \[hep-ph\]](#).
- [132] L. Adams, C. Bogner, A. Schweitzer, and S. Weinzierl, *The kite integral to all orders in terms of elliptic polylogarithms*, *J. Math. Phys.* **57** (2016) 122302, [arXiv:1607.01571 \[hep-ph\]](#).
- [133] L. Adams, E. Chaubey, and S. Weinzierl, *Simplifying Differential Equations for Multiscale Feynman Integrals beyond Multiple Polylogarithms*, *Phys. Rev. Lett.* **118** no. 14, (2017) 141602, [arXiv:1702.04279 \[hep-ph\]](#).
- [134] L. Adams and S. Weinzierl, *Feynman integrals and iterated integrals of modular forms*, [arXiv:1704.08895 \[hep-ph\]](#).
- [135] R. Mueller and D. G. Öztürk, *On the computation of finite bottom-quark mass effects in Higgs boson production*, *JHEP* **08** (2016) 055, [arXiv:1512.08570 \[hep-ph\]](#).
- [136] K. Melnikov, L. Tancredi, and C. Wever, *Two-loop $gg \rightarrow Hg$ amplitude mediated by a nearly massless quark*, *JHEP* **11** (2016) 104, [arXiv:1610.03747 \[hep-ph\]](#).
- [137] T. Hahn, *CUBA – a library for multidimensional numerical integration*, *Comput. Phys. Commun.* **168** (2005) 78–95, [arXiv:hep-ph/0404043 \[hep-ph\]](#).
- [138] I. Dubovyk, J. Gluza, T. Jelinski, T. Riemann, and J. Usovitsch, *New Prospects for the Numerical Calculation of Mellin-Barnes Integrals in Minkowskian Kinematics*, *Acta Phys. Polon.* **B48** (2017) 995, [arXiv:1704.02288 \[hep-ph\]](#).
- [139] S. Moch and P. Uwer, *–XSummer– Transcendental functions and symbolic summation in FORM*, *Comput. Phys. Commun.* **174** (2006) 759–770, [arXiv:math-ph/0508008 \[math-ph\]](#).
- [140] C. Schneider, *Symbolic Summation Assists Combinatorics*, *Sem. Lothar. Combin.* **56** (2007) 1–36. <http://www.mat.univie.ac.at/~slc/>. Article B56b.
- [141] C. Schneider, *Simplifying Multiple Sums in Difference Fields*, in *Computer Algebra in Quantum Field Theory: Integration, Summation and Special Functions*, C. Schneider and J. Blümlein, eds., Texts and Monographs in Symbolic Computation, pp. 325–360. Springer, 2013. [arXiv:1304.4134 \[cs.SC\]](#).
- [142] C. Schneider, *Modern Summation Methods for Loop Integrals in Quantum Field Theory: The Packages Sigma, EvaluateMultiSums and SumProduction*, *J. Phys. Conf. Ser.* **523** (2014) 012037, [arXiv:1310.0160 \[cs.SC\]](#).
- [143] M. Ochman and T. Riemann, *MBsums - a Mathematica package for the Representation of Mellin-Barnes Integrals by Multiple Sums*, *Acta Phys. Polon.* **B46** no. 11, (2015) 2117, [arXiv:1511.01323 \[hep-ph\]](#).
- [144] J. M. Henn, A. V. Smirnov, and V. A. Smirnov, *Evaluating multiple polylogarithm values at sixth roots of unity up to weight six*, *Nucl. Phys.* **B919** (2017) 315–324, [arXiv:1512.08389 \[hep-th\]](#).
- [145] A. V. Smirnov, *FIESTA4: Optimized Feynman integral calculations with GPU support*, *Comput. Phys. Commun.* **204** (2016) 189–199, [arXiv:1511.03614 \[hep-ph\]](#).
- [146] T. Huber and D. Maître, *HypExp, a Mathematica package for expanding hypergeometric functions around integer-valued parameters*, *Comput. Phys. Commun.* **175** (2006) 122–144, [arXiv:hep-ph/0507094 \[hep-ph\]](#).
- [147] T. Huber and D. Maître, *HypExp 2, Expanding hypergeometric functions about half-integer parameters*, *Comput. Phys. Commun.* **178** (2008) 755–776, [arXiv:0708.2443 \[hep-ph\]](#).

-
- [148] Y. Schröder and A. Vuorinen, *High-precision epsilon expansions of single-mass-scale four-loop vacuum bubbles*, *JHEP* **06** (2005) 051, [arXiv:hep-ph/0503209](#) [[hep-ph](#)].
- [149] T. Gehrmann, E. W. N. Glover, T. Huber, N. Ikizlerli, and C. Studerus, *Calculation of the quark and gluon form factors to three loops in QCD*, *JHEP* **06** (2010) 094, [arXiv:1004.3653](#) [[hep-ph](#)].
- [150] C. Bogner, S. Borowka, T. Hahn, G. Heinrich, S. P. Jones, M. Kerner, A. von Manteuffel, M. Michel, E. Panzer, and V. Papara, *Loopedia, a Database for Loop Integrals*, [arXiv:1709.01266](#) [[hep-ph](#)].
- [151] C. W. Bauer, A. Frink, and R. Kreckel, *Introduction to the GiNaC Framework for Symbolic Computation within the C++ Programming Language*, *J. Symb. Comput.* **33** (2000) 1, [arXiv:cs/0004015](#) [[cs-sc](#)].
- [152] “PStreams.” <http://pstreams.sourceforge.net>. Accessed: 2017-09-23.

Danksagung

Abschließend möchte ich mich noch bei den Personen bedanken, die mir bei der Erstellung dieser Dissertation zur Seite standen.

Zuallererst gilt meinem Dank Prof. Robert Harlander für die Projektideen in dieser Arbeit und für die sehr gute Betreuung. Bei Prof. Michał Czakon bedanke ich mich für die sehr kurzfristige Übernahme des Zweitgutachtens.

Besonderen Dank gebührt auch Johann Usovitsch für die Unterstützung bei der Reduktion der Feynman-Integrale. Ohne seine Unterstützung und die von ihm getätigte stetige Weiterentwicklung von *Kira* wäre das Projekt nicht so weit gekommen.

Bei Prof. Robert Lewis bedanke ich mich für die schnelle Reaktion auf Bugreports für das **Fermat** Computer-Algebra-System und für ausführliche Diskussionen über Polymods, die in das Tool **epsilon** Eingang fanden.

Bei der **AMBRE**-Kollaboration bedanke ich mich für hilfreiche Diskussionen über die Konstruktion von Mellin-Barnes-Darstellungen. Insbesondere gilt mein Dank hierbei Ievgen Dubovyk für die Unterstützung beim Vergleich von **AMBRE** mit der eigens entwickelten Technik.

Für das Führen von nützliche Diskussionen, gerade in der letzten Phase der Doktorarbeit, bedanke ich mich auch bei Domenico Bonocore.

Vielen Dank auch an meine beiden Korrekturleser Sonja Fischer und Fabian Lange, die jeden einzelnen Satz dieser Arbeit genau unter die Lupe genommen haben.

Die Arbeitsatmosphäre wäre nicht dieselbe gewesen ohne meine Bürokollegen Tom Zirke, Tobias Neumann, Lennart Oymanns und Jonas Klappert. Auch bei den übrigen Mitarbeitern des Instituts für Theoretische Teilchenphysik und Kosmologie bedanke ich mich für das angenehme Arbeitsumfeld und die herzliche Aufnahme nach unserem Umzug von Wuppertal nach Aachen.

Zu guter Letzt danke ich meinen Eltern für jegliche Unterstützung bereits während meines Studiums aber auch weit darüber hinaus.