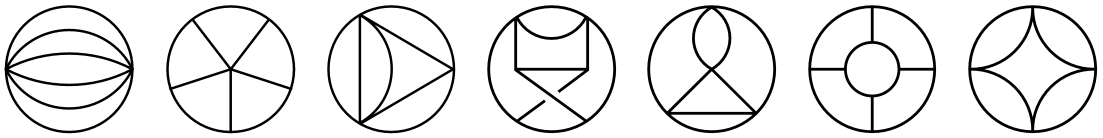
Fully massive vacuum integrals at 5 loops



Dissertation

submitted to the

Faculty of Physics,

Bielefeld University

by

Thomas Luthe

August 2015

1st examiner: Prof. Dr. Nicolas Borghini

Supervisor & 2nd examiner: Prof. Dr. York Schröder

Abstract

Massive vacuum integrals with a single mass scale are a class of Feynman integrals that appear in many precision calculations within the Standard Model of particle physics and have been calculated to the 4-loop level. In this thesis I start pushing this limit to 5 loops by considering the subclass of fully massive vacuum integrals, which can be used to determine the β-function of Quantum Chromodynamics (QCD). To this end I employ a method devised by Laporta for the evaluation of multi-loop Feynman integrals based on difference equations and factorial series. Significant improvements to this method are introduced to account for the great increase in complexity when going from 4 to 5 loops. An implementation of the improved approach in C++ is then used to obtain high-precision numerical results for the integrals needed for the 5-loop correction to the QCD β-function.

Contents

1 Introduction and motivation

Quantum Field Theory (QFT) and the Standard Model (SM) in particular have proven to be powerful tools in the description of particle physics and theoretical predictions are in good agreement with experiments for many observables. Since the accuracy of measurements is steadily increasing, it is necessary to improve the precision of calculations to obtain more stringent tests on the theory and possibly identify new physics. Within QFT these calculations often rely on expansions in a small coupling parameter and precision is increased by including more orders in this parameter. Higher orders in the expansions contain Feynman integrals with an increasing number of loop momenta which need to be integrated over. Each additional loop significantly increases the complexity and number of Feynman integrals that appear, making it necessary to automate their calculation.

In this thesis I refine and implement a technique for multiloop calculations first introduced by Laporta [1], which is based on difference equations and factorial series. I apply this method to the class of fully massive 5-loop Feynman integrals with no external momenta and a single mass scale. Integrals without external momenta are referred to as *vacuum* or *tadpole* integrals and appear in many calculations in QFT, but the main motivation for this thesis is their appearance in the β-function of Quantum Chromodynamics (QCD). This function governs the dependence of the coupling parameter of the strong interaction on the energy scale and can be computed using only fully massive tadpoles with a single mass scale. The β-function has been known up to the 4-loop level [2, 3] for some time and the integrals I calculate in this thesis can be used to push this limit to 5 loops.

Fully massive tadpoles with a single mass scale have been calculated at 4 loops using Laporta's method in both four [4] and three [5] dimensions, while at 5 loops only a handful of integrals can be found in the literature. Attempts have been made to apply Laporta's method to the 5-loop integrals [6, 7], but progress has been limited by the fact that there is a steep increase in complexity compared to 4 loops. In this thesis I introduce a number of improvements to the method that improve its speed by several orders of magnitude for difficult integrals, thus making it a viable option for 5-loop calculations. The fully massive tadpoles considered here are only the first step in the evaluation of 5-loop tadpoles. The improvements made to Laporta's method are not limited to this class of integrals and could e.g. also be applied to obtain the tadpole integrals with some massless propagators, a class of integrals which appears in many additional calculations.

The thesis is organised as follows. In the remainder of this section I motivate the calculation of massive tadpoles by briefly presenting how they can be used to obtain the β-function of QCD and other results in QFT. In section 2 I review some commonly used techniques that reduce the number of Feynman integrals that need to be evaluated to a small number

of master integrals. Many different techniques have been developed to solve these master integrals and section 3 provides a brief introduction to some of them. The main part of the thesis starts with section 4, where I examine the original version of Laporta's method and discuss its applicability to the fully massive 5-loop tadpoles. My own improvements to the method are presented in detail in sections 5 through 7. Subsequently I outline in section 8 some of the technical details of my implementation of the method in C++ . Results for the 5-loop fully massive tadpoles are presented in section 9, before I make some concluding remarks and list ideas for further improvements and applications of Laporta's method in section 10.

1.1 The β-function of QCD

The Lagrangian for a Yang-Mills theory [8] with fermions is

$$\begin{aligned}
\mathcal{L} &= \sum_{f=1}^{N_f} \sum_{i,j=1}^{N_c} \bar{\Psi}_{f,i} \left[i\gamma^\mu D_\mu - m_f \right]_{ij} \Psi_{f,j} - \frac{1}{4} F^a_{\mu\nu} F^{a\mu\nu} + \mathcal{L}_{\text{gf}} + \mathcal{L}_{\text{ghosts}} \,, \\
\mathcal{L}_{\text{gf}} &= -\frac{1}{2\xi} \left(\partial^\mu A^a_\mu \right)^2 \,, \\
\mathcal{L}_{\text{ghosts}} &= -\bar{c}^a \partial^\mu \left(g f^{abc} A^b_\mu + \delta^{ac} \partial_\mu \right) c^b \,, \\
[D_\mu]_{ij} &= \delta_{ij} \partial_\mu - i g A^a_\mu T^a_{ij} \,, \\
F^a_{\mu\nu} &= \partial_\mu A^a_\nu - \partial_\nu A^a_\mu + g f^{abc} A^b_\mu A^c_\nu \,,
\end{aligned} \tag{1.1}$$

where a gauge fixing term $\mathcal{L}_{\text{gf}}$ has already been introduced using the Faddeev-Popov method [9] and $\mathcal{L}_{\text{ghosts}}$ is the corresponding term with ghost fields c^a. The matrices T^a are the generators of the underlying Lie group $SU(N_c)$ and satisfy the algebra

$$\left[T^a, T^b \right] = i f^{abc} T^c \tag{1.2}$$

with structure constants f^{abc}. For QCD Ψ_f are the quark fields with masses m_f and N_f flavours, A^a_μ the gluon fields, the number of colours N_c is 3 and g is the coupling parameter of the strong interaction.

Ultraviolet (UV) divergences that arise in QFT are treated in this thesis by dimensional regularisation [10], in which the space-time dimension $d = 4 - 2\epsilon$ is expanded around 4 and divergences appear as poles in the expansion parameter ϵ. QCD can then be renormalised [11] to remove UV divergences by rescaling the quantities appearing in the Lagrangian as

$$\begin{aligned}
X_B &= Z_X X_R \,, \quad X \in \{\Psi, A, c, m, \xi\} \,, \\
g_B &= Z_g g_R \mu^\epsilon \,,
\end{aligned} \tag{1.3}$$

where the subscripts B and R denote bare and renormalised quantities, respectively, and μ is the unit of mass introduced by 't Hooft to keep the coupling parameter g_R dimensionless. In the minimal subtraction (MS) scheme [10] the renormalisation constants Z_X take the form

$$Z_X = 1 + \delta Z_X = 1 + \sum_{l=1}^{L} \sum_{k=1}^{l} Z_X^{(l,k)} \frac{h^l}{\epsilon^k} \,, \tag{1.4}$$

where $h \equiv \frac{g_R^2}{16\pi^2}$ and L is the number of loops up to which UV divergences are removed. The β-function is defined in terms of h as

$$\beta(h) = \mu^2 \frac{\partial h}{\partial \mu^2} \tag{1.5}$$

and describes the dependence of the coupling on the energy scale μ. With the definition $Z_h \equiv Z_g^2$ and using the fact that the bare coupling does not depend on μ one finds

$$\begin{aligned} \beta(h) &= \mu^2 \frac{\partial}{\partial \mu^2} \frac{g_B^2}{16\pi^2} \mu^{-2\epsilon} Z_h^{-1} \\ &= -\epsilon h + h Z_h \mu^2 \frac{\partial Z_h^{-1}}{\partial \mu^2} \\ &= -\epsilon h - h\beta(h) \frac{\partial \ln Z_h}{\partial h} \\ &= \frac{-\epsilon h}{1 + h \frac{\partial \ln(Z_h)}{\partial h}} \,. \end{aligned} \tag{1.6}$$

Inserting equation (1.4) and expanding in h yields

$$\begin{aligned} \beta(h) &= -\epsilon h \sum_{i=0}^{\infty} (-1)^i \left(h \frac{\partial}{\partial h} \sum_{j=1}^{\infty} \frac{(-1)^{j+1}}{j} \left[\delta Z_h\right]^j \right)^i \\ &= -\epsilon h \sum_{i=0}^{\infty} (-1)^i \left(h \frac{\partial}{\partial h} \sum_{j=1}^{\infty} \frac{(-1)^{j+1}}{j} \left[\sum_{l=1}^{\infty} \sum_{k=1}^{l} Z_h^{(l,k)} \frac{h^l}{\epsilon^k} \right]^j \right)^i \,. \end{aligned} \tag{1.7}$$

Demanding that the limit $\epsilon \to 0$ exists fixes all $Z_h^{(l,k)}$ with $k > 1$ in terms of the $\frac{1}{\epsilon}$ poles. The only terms that remain are those with $i = j = k = 1$ and the β-function then reads

$$\beta(h) = h^2 \frac{\partial}{\partial h} \sum_{l=1}^{\infty} Z_{h,l1} h^l = \sum_{l=0}^{\infty} h^{l+2} (l+1) Z_h^{(l+1,1)} \equiv -\sum_{l=0}^{\infty} \beta_l h^{l+2} \,, \tag{1.8}$$

For the coefficients $\beta_l = -(l+1) Z_h^{(l+1,1)}$ the renormalisation constant Z_h is needed to $L = l + 1$ loops. The calculation of the leading order

$$\beta_0 = \frac{11}{3} N_c - \frac{2}{3} N_f \tag{1.9}$$

was done in the 1970s by 't Hooft [12], Gross and Wilczek [13] and Politzer [14]. The minus sign in this result for the beta function for $N_c = 3$ and small N_f leads to asymptotic freedom for the quarks at high energies as the coupling strength decreases. This discovery played an important role in establishing QCD as the theory describing the strong interaction and was recognised with the Nobel prize in 2004. Since the leading order was found, the 2-loop [15, 16, 17], 3-loop [18, 19] and 4-loop [2, 3] corrections have also been obtained. In general $\beta(h)$ depends on the renormalisation scheme, but in the commonly used schemes MS and $\overline{\text{MS}}$ [20] it is identical, since Z_h has no explicit dependence on μ [2].

The β-function is closely related to the more general concept of anomalous dimensions $\gamma_X(h)$ which are defined as

$$\gamma_X(h) = -\mu^2 \frac{\mathrm{d} \log Z_X(H)}{\mathrm{d}\mu^2} = \sum_{l=0}^{\infty} \gamma_{X,l} h^{l+1} \,, \tag{1.10}$$

where Z_X is the renormalisation constant of a quantity X. In fact, up to normalisation, $\beta(h)$ is simply the anomalous dimension of the coupling parameter h. There are then several possible choices to calculate the β-function in terms of other anomalous dimensions or equivalently Z_h in terms of other renormalisation constants. It can be obtained e.g. by calculating the quark-gluon-vertex and the quark- and gluon-propagators, or alternatively the quark-ghost-vertex and quark- and ghost-propagators. It is even possible to obtain Z_h from only the gluon-propagator by using the background field method [21], albeit at the cost of more complicated Feynman rules. Some of the QCD anomalous dimensions at the 5-loop level have already been computed [22, 23] by reducing the calculation to the evaluation of 4-loop massless propagator type integrals with the so-called R^*-operation [24], but a result for the β-function is still missing.

The crucial point about the calculation of the β-function and anomalous dimensions from the point of view of this thesis is the fact that it only depends on UV-divergent, but not finite or infrared-divergent terms. This allows for significant simplifications compared to the calculation of full Green's functions. One option for such a simplification is offered by the R^* operation [24] used in the above mentioned calculations of 5-loop anomalous dimensions [22, 23]. This operation allows to extract the UV counterterms of integrals with L loops in terms of divergent and finite parts of $L-1$ loop massless propagators, but at the cost of a rather complicated procedure. A simpler approach is given in [25, 2] and has been used e.g. in the 4-loop calculations of the QCD β-function [2, 3]. In this approach the UV divergences of integrals are completely extracted in terms of fully massive tadpole integrals with a single artificially introduced mass m. This can be achieved by applying

the exact decomposition

$$\frac{1}{(k+p)^2+M^2} = \frac{1}{k^2+m^2} - \frac{2k\cdot p + p^2 + M^2 - m^2}{k^2+m^2}\frac{1}{(k+p)^2+M^2} \tag{1.11}$$

to all propagators, where k is a loop momentum, p an external momentum and M the original mass of the propagator, which can also be 0. The first term in this sum does not depend on any external momenta and does not lead to infrared divergences. The second term goes as $1/k^3$ for large loop momenta and thus has a lower superficial degree of (UV) divergence. Since the original propagator appears in the second term, the decomposition can be applied recursively until p and M only appear in UV-finite integrals or numerators. The UV-finite integrals can then be dropped and the numerator dependence on p and M can be extracted from the integrals. In the end one is left with the class of integrals treated in this thesis, fully massive vacuum integrals with a single mass scale. Since the decomposition in equation (1.11) is exact, the final result cannot depend on the artificial mass m. A minor inconvenience of this method is the necessity to introduce a "gluon mass" counterterm $\frac{m^2}{2} Z_{gm} A^a_\mu A^{a\mu}$ [25].

1.2 Pressure in hot QCD

At high temperature T hadronic matter ungergoes a phase transition to a quark-gluon-plasma. Studying properties like the pressure of this plasma can yield a better understanding of heavy ion collision experiments or help determine the expansion rate of the early universe. In a perturbative approach one is faced with the problem that the coupling strength g depends strongly on the momentum scale, but there are three different important momentum scales that play a role in the plasma. In decreasing order of magnitude these are the hard scale T of a typical particle in the plasma, the soft scale gT associated with the screening of colour-electric forces, and the ultrasoft scale g^2T associated with colour-magnetic screening [26]. To separate the treatment of these scales a set of effective theories is introduced by the process of *dimensional reduction* [27, 28]. In the following I briefly summarise this approach following the notation of [29] in order to show how massive tadpoles emerge in the calculation.

Using a Euclidean metric, the pressure of full massless QCD is defined as

$$p_{\mathrm{QCD}}(T) \equiv \lim_{V\to\infty} \frac{T}{V} \ln \int \mathcal{D}A^a_\mu \, \mathcal{D}\psi \, \mathcal{D}\bar{\psi} \exp\Big(-S_{\mathrm{QCD}}\Big), \tag{1.12}$$

where V is the volume of the d-dimensional coordinate space and the action and Lagrangian

are given by

$$S_{\mathrm{QCD}} = \int_0^\beta \mathrm{d}\tau \int \mathrm{d}^d x \, \mathcal{L}_{\mathrm{QCD}} \, , \tag{1.13}$$

$$\mathcal{L}_{\mathrm{QCD}} = \frac{1}{4} F^a_{\mu\nu} F^a_{\mu\nu} + \bar{\psi} \gamma_\mu D_\mu \psi \tag{1.14}$$

with $\beta = 1/T$ and $F^a_{\mu\nu}$ and D_μ defined as in eq. (1.1). After the first separation of scales with dimensional reduction the pressure is given in the resulting 3-dimensional (i.e. here $d = 3 - 2\epsilon$) effective field theory called electrostatic QCD (EQCD) as

$$p_{\mathrm{QCD}}(T) \equiv p_{\mathrm{E}}(T) + \lim_{V\to\infty} \frac{T}{V} \ln \int \mathcal{D}A^a_k \, \mathcal{D}A^a_0 \exp\Big(-S_{\mathrm{E}}\Big), \tag{1.15}$$

$$S_{\mathrm{E}} = \int \mathrm{d}^d x \, \mathcal{L}_{\mathrm{E}}, \tag{1.16}$$

$$\mathcal{L}_{\mathrm{E}} = \frac{1}{2} \mathrm{Tr}\, F^2_{kl} + \mathrm{Tr}\,[D_k, A_0]^2 + m^2_{\mathrm{E}} \mathrm{Tr}\, A^2_0 + \lambda^{(1)}_{\mathrm{E}} (\mathrm{Tr}\, A^2_0)^2 + \lambda^{(2)}_{\mathrm{E}} \mathrm{Tr}\, A^4_0 + ... \tag{1.17}$$

with $k = 1, \ldots, d$ and g replaced by a coupling g_E in the definitions of F_{kl} and D_k. In this effective theory $p_{\mathrm{E}}(T)$ contains the contributions of the hard scale, while the remaining partition function includes contributions of the soft and ultrasoft scales. The parameters of EQCD with subscript E are not a priori known, but have to be determined by matching calculations of the effective theory with calculations of the full theory.

In a second step the soft and ultrasoft scales are separated by integrating out the field A_0 to arrive at another effective theory called magnetostatic QCD (MQCD) in which the pressure is given by

$$p_{\mathrm{QCD}}(T) \equiv p_{\mathrm{E}}(T) + p_{\mathrm{M}}(T) + p_{\mathrm{G}}(T) \, , \tag{1.18}$$

$$p_{\mathrm{G}}(T) = \lim_{V\to\infty} \frac{T}{V} \ln \int \mathcal{D}A^a_k \exp\Big(-S_{\mathrm{M}}\Big), \tag{1.19}$$

$$S_{\mathrm{M}} = \int \mathrm{d}^d x \, \mathcal{L}_{\mathrm{M}}, \tag{1.20}$$

$$\mathcal{L}_{\mathrm{M}} = \frac{1}{2} \mathrm{Tr}\, F^2_{kl} + ... \, , \tag{1.21}$$

where the hard, soft and ultrasoft scales are now contained in p_{E}, p_{M} and p_{G}, respectively. MQCD contains the new parameters p_M and g_M (in F_{kl}) which have to be matched to the parameters of EQCD by appropriate calculations in both theories. Because $\mathcal{L}_{\mathrm{M}}$ defines a confining theory, p_{G} can only be computed non-perturbatively by methods such as lattice gauge theory.

Massive tadpole integrals enter this calculation in the theory of EQCD (eq. (1.17)), which consists of the massless gauge fields A_k and the so-called adjoint Higgs field A_0 with mass m_E. In the expression for the pressure one therefore encounters vacuum integrals in

$d = 3 - 2\epsilon$ dimensions with both massive and massless lines, but only a single mass scale. The calculations for EQCD have been done to the 4-loop level [29], but for full QCD there are still unsolved 4-loop sum integrals.

Fully massive vacuum integrals can be found if the method of separating scales with dimensional reduction is used on hot massless ϕ^4 theory (see e.g. [30, 31]). The Lagrangian of the resulting effective 3-dimensional theory reads [31]

$$\mathcal{L}_{\text{eff}} = \frac{1}{2}(\nabla\phi)^2 + \frac{m^2}{2}\phi^2 + \frac{g_3^2}{4!}\phi^4 + \dots \,, \tag{1.22}$$

where m and g_3 are parameters that have to be matched to the full theory with sum integrals in 4 dimensions. The pressure in this theory has been calculated up to 5-loop contributions [31], including some of the leading ϵ-orders of three of the fully massive vacuum integrals considered in this thesis, which provides a check on my own results (see sect. 9.3).

1.3 Other applications

There are many further applications for massive tadpole integrals, listed e.g. in [32, 33] or more recently in [34]. One such application is the ρ parameter of the electroweak interaction, which is defined as the ratio of the strengths of charged and neutral currents [35]. At leading order ρ is simply one, but at higher orders it is sensitive to the full Standard Model. The corrections include contributions from QCD which require the evaluation of massive tadpoles and have been calculated to the 4-loop level [36, 33]. Higher orders of the ρ parameter can also be used to improve the theoretical predictions for electroweak observables such as the mass of the W boson (see e.g. [37, 38]) or the weak mixing angle (see e.g. [38, 39]).

Another application of massive tadpoles is given by decoupling functions. By integrating out a heavy quark with mass m to avoid large logarithms in MS-like renormalisation schemes one arrives at an effective theory with only $N_l = N_f - 1$ quark flavours [34]. The strong coupling parameter $g^{(N_l)}$ of the effective theory differs from the coupling $g^{(N_f)}$ in the full theory as

$$g^{(N_l)}(\mu) = \zeta_g\left(\mu, g^{(N_f)}(\mu), m\right) g^{(N_f)}(\mu) \,, \tag{1.23}$$

where $\zeta_g(\mu, g^{(N_f)}(\mu), m)$ is called the decoupling function. The evaluation of ζ_g involves gluon and ghost vacuum polarisation functions as well as the gluon-ghost vertex at zero momenta and thus requires massive tadpoles [34]. This calculation has been performed up to 4 loops [40, 41]. ζ_g plays an important role in Higgs boson production via gluon fusion, where it is needed to obtain the coupling of gluons to the Higgs in the effective

theory [42]. In an analogous fashion to ζ_g one can define the photon decoupling function, which has also been obtained to 4 loops [43] and is needed e.g. for the amplitude of Higgs to photon decay in the effective theory [34].

In some calculations it is sufficient to only consider some low moments in external momenta. Each term of the expansion in these momenta then consists of tadpole integrals with increasingly higher powers of propagators and irreducible scalar products in the numerator. This can be used e.g. in the determination of charm and bottom quark masses [34, 44], where this technique has been applied up to the 4-loop level [45, 46]. 5-loop results from anomalous dimensions [22, 23] have been applied in parts, but the contribution of the β-function is still missing [34].

The above examples also depend on tadpole integrals where only some of the propagators are massive. This class of integrals is not studied in this thesis, but is closely related to the fully massive integrals considered here. Laporta's method of difference equations and its improvements presented in later sections of the thesis are not limited to the fully massive case and thus the applications given here serve as further motivation to push the technical limits of tadpole evaluation to the 5-loop level.

2 Reduction to master integrals

In this chapter I will fix the notation and classification of Feynman integrals used in this thesis. I will also describe the steps that are commonly used to express the Feynman integrals one may encounter in calculations in terms of a relatively small set of integrals. This reduction to so-called *master integrals* is often necessary due to the vast number of integrals that appear, e.g. already at 4 loops of the order of two million integrals were evaluated during the calculation of the QCD β-function [3].

2.1 Notation

All momentum vectors in this thesis use a Euclidean metric unless otherwise stated. Integrals are regularised with dimensional regularisation [10], where the space-time dimension d is expanded around an integer value in a small parameter ϵ. I use

$$\int_{\{k\}} \equiv \int_{k_1,\ldots,k_L} \equiv \int \frac{\mathrm{d}^d k_1}{\pi^{d/2}} \ldots \int \frac{\mathrm{d}^d k_L}{\pi^{d/2}} \tag{2.1}$$

as a shorthand for the integration measure, where L is the number of loops of the associated Feynman diagram and k_i are the loop momenta. A generic Feynman integral with n_E external momenta p_i can then be written as

$$F^{\mu_1,\ldots,\mu_n}(p_1,\ldots,p_{n_E},\{m_i\},d) \equiv \int_{\{k\}} \frac{N^{\mu_1,\ldots,\mu_n}}{D_1^{z_1}\ldots D_t^{z_t}} \,. \tag{2.2}$$

The D_i are the denominators of propagators[1] and have the form $D_i = q_i^2 + m_i^2$, where m_i denotes the corresponding mass and the propagator momenta q_i are linear combinations of both loop and external momenta with coefficients -1,0 or 1. The t propagators have exponents z_i, which are typically small integers. The numerator $N^{\mu_1,\ldots,\mu_n}$ depends on polynomials of momentum dot products and can in general also carry a Lorentz-structure.

In this thesis I will restrict myself to fully massive vacuum integrals with a single mass scale, which means $n_E = 0$ and $m_i = m \equiv 1$. Furthermore, since tensor integrals can always be expressed as linear combinations of scalar integrals where the Lorentz-structure is independent of the loop momenta (see e.g. [47, 48]), I will only consider scalar numerators N for all integrals. N may then only depend on the $w = L(L+1)/2$ dot products of loop momenta. Since the q_i^2 depend on the same dot products, the notation of numerator and

[1] I will usually refer to the D_i simply as *propagators* when there is no chance of confusion.

denominator can be unified. By choosing a set of propagator momenta

$$A_L = \left\{ q_i | i = 1, \ldots, w; \{q_i^2\} \text{ linearly independent} \right\} \tag{2.3}$$

any numerator N can be expressed as a linear combination of the D_i and $m^2 = 1$. The remaining integrals

$$I(z_1, \ldots, z_w) = \int_{\{k\}} \frac{1}{D_1^{z_1} \ldots D_w^{z_w}} \tag{2.4}$$

are then uniquely identified by their integer exponents z_i, which may now also be negative.

It should be noted that the choice of A_L is not completely arbitrary, since the set of integrals which it defines through the possible choices of the z_i is not necessarily a superset of the set of L-loop integrals which may appear in quantum field theory calculations. The latter consists of integrals whose denominator momenta can be mapped onto an L-loop graph to form a Feynman diagram with momentum conservation at each vertex. While in practice some of those graphs may not yield a contribution to a particular calculation, all of them can do so in principle and should therefore have associated subsets of A_L which can be mapped onto them. It is trivial to show that not all choices of A_L cover all possible graphs: Suppose someone would choose at the 2-loop level $A_2 = \{q_1 = k_1, q_2 = k_1 - k_2, q_3 = k_1 + k_2\}$. Even though the previous conditions (coefficients $\in \{-1, 0, 1\}$, q_i^2 linearly independent) are satisfied, there is no way to map these momenta onto the 2-loop sunset graph in figure 2.1. Ideally, one should therefore additionally require that each L-loop graph has at least one associated subset of A_L. In the 2-loop sunset example a better choice would be e.g. $A_2 = \{q_1 = k_1, q_2 = k_2, q_3 = k_1 - k_2\}$. If it is not possible to find a set A_L which covers all possible graphs, multiple sets $A_L^{(1)}, A_L^{(2)}, \ldots$ are needed and integrals will often have to be converted between them. Fortunately this is not the case for tadpole integrals up to 5 loops and the momenta chosen for $A_1, \ldots, A_5$ in this thesis can be found in table A.1.

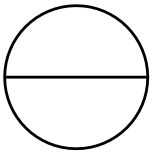


Figure 2.1: The 2-loop sunset graph.

The subsets of A_L also offer a natural starting point for the classification of the integrals $I(\{z_i\})$. Every subset $S \subset A_L$ defines an infinite set of integrals

$$\left\{ I(z_1, \ldots, z_w) \middle| \begin{array}{ll} z_i \geq 1, & \text{if } q_i \in S \\ z_i \leq 0, & \text{if } q_i \notin S \end{array} \right\} \tag{2.5}$$

which I will call a *sector*. To refer to the 2^w different possible sectors, each of them is assigned an integer *sector ID*. Along with a few other useful definitions for an integral

$I(\{z_i\})$, it is given by

$$\begin{aligned} sID(I) &= \sum_i \Theta(z_i) 2^{w-i} \ , && \text{sector ID,} \\ r(I) &= \sum_i \Theta(z_i)(z_i - 1) \ , && \text{denominator powers beyond 1 (\textit{dots}),} \\ s(I) &= \sum_i -\Theta(-z_i) z_i \ , && \text{numerator powers,} \\ t(I) &= \sum_i \Theta(z_i) \ , && \text{number of denominators,} \end{aligned} \tag{2.6}$$

where $\Theta(> 0) = 1$, $\Theta(\leq 0) = 0$. The number of denominators t is identical for all integrals of the same sector and, if the sector has an associated graph, also denotes the number of lines of that graph. The additional powers r of the denominators beyond 1 are often simply referred to as *dots*, because in a Feynman diagram they are symbolised by dots on the line carrying their respective propagator momentum. For the numerator powers s there is no good pictorial representation, since their momenta do not have any associated lines. r, s and t play an important role in establishing an ordering relation between integrals, which will be defined in section 2.4. As an example for the parameters defined here and their connection to the diagram consider the 3-loop integral

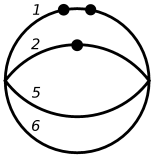


Figure 2.2: A 3-loop diagram of sector 51.

$$I(3, 2, 0, -2, 1, 1) = \int_{\{k\}} \frac{D_4^2}{D_1^3 D_2^2 D_5 D_6} \ , \tag{2.7}$$

which is part of sector 51 and has $t = 4$, $r = 3$ and $s = 2$. The corresponding diagram using the set of propagator momenta $A_3 = \{k_1, k_2, k_3, k_1 - k_2, k_1 - k_3, k_2 - k_3\}$ is given in figure 2.2 with the lines numbered by the index of their associated propagators.

If one looks at the maximum number of lines an L-loop vacuum graph can have, given by

$$t_{max}(L) = \begin{cases} 1 & \text{if } L = 1 \\ 3(L-1) & \text{if } L > 1 \end{cases} \ , \tag{2.8}$$

one finds that $t_{max}(L) < w(L) = L(L+1)/2$ for $L \geq 4$. This means that there are inevitably sectors with $t > t_{max}$ which have no associated diagram and can thus never appear in QFT-calculations. It therefore becomes useful to divide the sectors into different categories. I will refer to a sector with subset $S \subset A_L$ as

- a *physical sector*, if it contains non-vanishing integrals and the momenta in S can be mapped onto an L-loop graph with momentum conservation,

- a *zero-sector*, if all its integrals vanish in dimensional regularisation,
- an *anti-sector*, if the momenta in S cannot be mapped onto any L-loop graph.

Anti-sectors are not limited to $t > t_{max}$, as can be seen from the 2-loop example above with the unfortunate choice $A_2 = \{k_1, k_1 - k_2, k_1 + k_2\}$, where the sector with $t = t_{max} = 3$ does not have an associated graph. Zero-sectors appear when one or more of the loop momenta are not present in the denominator to begin with or can be removed from it by a momentum shift (see section 2.2). This condition is trivially satisfied for $t < L$, but can still be met for as many as $t = w(L-1) = L(L-1)/2$ propagators. In the end one only has to consider the physical sectors, many of which will still turn out to be equivalent using momentum shifts.

2.2 Momentum shifts

A great number of relations between integrals can be found by simply performing the transformation

$$k_i \to R_{ij} k_j \tag{2.9}$$

for the loop momenta, where R is an invertible $L \times L$ matrix. To avoid introducing a dependence on the dimension d in the exponent of the Jacobian determinant, the transformations should be limited to $\det R = \pm 1$. While it is possible to construct situations where a shift with $|\det R| \neq 1$ might be called for, these do not occur in practice for reasonably chosen momenta [7]. Transformations of the form (2.9) are first needed to fit any integral that might appear in a given calculation into the chosen classification scheme by shifting the momenta k_i in such a way that all resulting propagator momenta q_i (or alternatively $-q_i$) in the denominator are elements of A_L. Once this is accomplished, one only needs to consider how transformations R act on the propagators D_i. Since the q_i^2 form a basis in the space of momentum dot-products, R will map any propagator D_i to a linear combination of propagators and a pure mass term:

$$D_i \xrightarrow{R} D_i' = \sum_{n=1}^{w} c_n D_n + c \,, \quad c_n, c = \text{const.} \tag{2.10}$$

For D_i in the numerator a linear combination poses no problems, since one can simply split the integral into a sum of integrals, but in the denominator it would move an integral outside the classification scheme of A_L. For an integral $I(\{z_i\})$ the transformations R thus

have to be restricted to those which satisfy

$$D_i \xrightarrow{R} D_i' = D_j \quad \text{if } z_i > 0. \tag{2.11}$$

In the following I distinguish between two kinds of these transformations. If, for a given sector S, P is the set of denominator D_i and P' the set of the corresponding D_i', I call the transformation R

- a *symmetry shift*, if $P = P'$ and R thus transforms integrals in S to other integrals in S, or
- a *sector shift*, if $P \neq P'$ and R thus transfers integrals in S to a different sector.

It is obvious that sector shifts can only exist between sectors which are associated to the same graph, since the transformation is simply a relabelling of the loop momenta and does not change the structure of the graph. It also follows that all sectors associated to the same graph are connected by sector shifts. One can therefore choose a representative sector for each graph and shift all integrals that can be mapped onto the graph to that sector. For higher loop orders, this drastically reduces the number of sectors that need to be considered. In this work I always pick the sector with the greatest sector ID as the representative. Once a minimal set of sectors is identified, the symmetry shifts within these sectors offer a convenient option to further reduce the number of integrals that need to be evaluated. To this end each symmetry relation will be solved for its most difficult integral according to the ordering relation I define in section 2.4 and used to eliminate that integral whenever it occurs in an equation. More details on the implementation of momentum shifts are given in section 8.1.1.

If an integral subjected to either a sector shift or a symmetry shift contains negative exponents z_i, the corresponding propagators in the numerator will in general be shifted to a linear combination of propagators and a mass term (see eq. (2.10)). The classification scheme then requires that the numerator be expanded and the integral split up into a sum of integrals. It can happen that in some of these terms the numerator contains a factor D_i which cancels the corresponding propagator in the denominator. If an originally positive exponent z_i goes to zero or becomes negative in this way, the number of propagators t is reduced. In the associated Feynman graph of the original integral this corresponds to contracting the i-th line to a point. If the original integral is in a sector S with t propagators, the new integral will be in a different sector S' with $t-1$ propagators. In the following I will refer to S' as a *subsector* of S in this situation. More generally, I will use the term subsector (of S) for all sectors which can be arrived at by removing propagators from S and also all of their equivalent sectors, which can be reached via sector shifts.

2.3 Linear relations between integrals

While the momentum shifts described in section 2.2 already considerably reduce the number of integrals which need to be evaluated, there are further linear relations between integrals to be found. These relations can be used to reduce any integral to a linear combination of a small set of master integrals. The algorithm typically applied for this reduction is explained in section 2.4, while in this section I describe several methods to generate the needed equations.

Since every sector contains an infinite number of integrals, one has to set constraints which define the finite set of integrals that should be reduced. I will in the following consider all integrals with $r(I) \leq r_{\max}$, $s(I) \leq s_{\max}$ for some (typically small) integers $r_{\max}$, $s_{\max}$. The choice of these constraints is crucial, since it can have a huge impact on the time and memory consumption of the reduction. The integrals appearing in the expression which one actually wants to solve provide lower bounds for $r_{\max}$ and $s_{\max}$, but those might not always be sufficient. With low constraints one risks not finding sufficient information to determine the minimal number of master integrals, since the number of equations which can be generated containing only integrals within the constraints is limited. If a relation between integrals is missing from the system, the equations can contain a spurious zero in the form of the missing relation times an arbitrary coefficient. Over the course of the reduction such coefficients often grow rapidly up to the point where they considerably slow down the whole calculation or even make it infeasible altogether.

When increasing $r_{\max}$ or $s_{\max}$ on the other hand, the computation is quickly bound by combinatorics, since the number of integrals (before symmetrisation) with fixed r and s in a sector with t propagators is given by

$$N(w,t,r,s) = \underbrace{\binom{t+r-1}{r}}_{\substack{\text{distribution}\\ \text{of dots}}} \cdot \underbrace{\binom{w-t+s-1}{s}}_{\substack{\text{distribution of}\\ \text{negative powers}}} . \tag{2.12}$$

The rapid growth of the number of integrals can be a challenge to limited memory and CPU capacities. To give an example, for a 5-loop sector with 8 propagators, choosing $r_{\max} = s_{\max} = 5$, one would have to consider more than 10^6 integrals for that sector alone.

2.3.1 Integration by parts

The most common way to generate relations between integrals is by taking a *seed integral* $I(z_1, \ldots, z_w)$ and inserting into the integrand a differential operator $O_{ij} = \frac{\partial}{\partial k_i^\mu} k_j^\mu$ such that the integral vanishes in dimensional regularisation [49]:

$$0 = \int_{\{k\}} \frac{\partial}{\partial k_i^\mu} k_j^\mu \frac{1}{D_1^{z_1} \ldots D_w^{z_w}} , \quad i, j = 1, \ldots, L. \tag{2.13}$$

After explicitly performing the derivative, the result can be expressed as a linear combination of integrals in the chosen classification scheme with exponents possibly differing from the seed integral, thus giving a non-trivial relation between integrals. For non-vacuum integrals, k_j^μ may also be replaced with external momenta, yielding a total of $L(L + n_E)$ equations per seed integral. This means that the system of equations generated from integration-by-parts (IBP) is typically vastly overdetermined. In reference [50] Lee showed that the operators $O_{ij} = \frac{\partial}{\partial k_i^\mu} k_j^\mu$ form a Lie-algebra and in principle it is sufficient to work with a smaller subset of the IBP-relations which only needs $L + n_E + 1$ equations per seed integral. In practice however this turns out to be of little use, since for finite $r_{\max}$ and $s_{\max}$ only a small fraction of all IBP-relations can be safely skipped without risk of losing information [7]. An efficient method of dealing with redundant equations is presented in section 8.1.4.

2.3.2 Syzygies

For a seed integral with parameters r and s the IBP-relations in general contain integrals for which both parameters can be increased by 1. This increase can pose a problem in the reduction, since it introduces more complicated integrals (see section 2.4). To partially avoid this problem, Gluza, Kajda and Kosower [51] proposed to replace the IBP-operators O_{ij} with linear combinations $\sum_{ij} O_{ij} \alpha_{ij}$ which do not raise the number of dots r. This can be achieved by requiring that each raised propagator is cancelled immediately, using the condition

$$\sum_{i=1}^{L} \sum_{j=1}^{L+n_E} \alpha_{ij} \frac{\partial D_c}{\partial k_i^\mu} K_j^\mu \;\propto\; D_c \quad \forall c \text{ with } z_c > 0, \tag{2.14}$$

where K_j runs over both loop and external momenta. The α_{ij} can be polynomials in scalar products of momenta (or equivalently polynomials in the inverse propagators D_i), so the drawback of not raising r is that the number of powers in the numerator s might be increased by more than 1. The solutions α to eq. (2.14) are called *syzygies* and can be constructed using Gröbner bases or basic linear algebra [52]. By using a modified and less strict version of condition (2.14), it is possible to also allow solutions α where

raised propagators are eliminated due to symmetries or integrals vanishing in dimensional regularisation [7].

Equations generated from syzygies are by design simply linear combinations of IBP-relations and can thus not introduce information which cannot also be obtained via IBP. Their advantage is that the linear combinations are chosen such that integrals with raised propagator exponents never have to be considered, independent of which seed integral is used. This can reduce the overall number of integrals, since fewer integrals are introduced to the system in this way. The linear combination of IBP-operators does however also lead to equations which typically have more terms than pure IBP-relations.

2.3.3 Mass derivatives and Lorentz invariance

If all propagators are expressed as $D_i = q_i^2 + x_i m^2$ in terms of a single mass scale m by introducing dimensionless parameters x_i where needed, the mass dimension of an integral can be made explicit by pulling it outside the integral:

$$I(z_1, \ldots, z_w) = m^{Ld-2\sum_i z_i} X \ . \tag{2.15}$$

Here X does not depend on m and acting on both sides with $\frac{\partial}{\partial m^2}$ yields

$$-\sum_i z_i x_i I(z_1, \ldots, z_i + 1, \ldots, z_w) = \left(\frac{Ld}{2} - \sum_i z_i\right) m^{-2} I(z_1, \ldots, z_w) \ . \tag{2.16}$$

This turns out to simply be the IBP-relation for the operator $\sum_i O_{ii}$ and thus does not offer any additional information. Nevertheless this particular form can be useful, since unlike most IBP-relations it does not contain integrals where the parameter s is increased compared to the seed integral $I(z_1, \ldots, z_w)$.

If the integrals in question depend on external momenta, one can also derive linear equations from the fact that they are Lorentz scalars [47]. While the equations have been shown to be contained in the IBP-relations [50] and only vacuum integrals are considered in this thesis, I still show the derivation for the sake of completeness. From the Lorentz-invariance of an integral $J(p_1, \ldots, p_{n_E})$ with external momenta p_i follows that J is independent under the infinitesimal transformation

$$p^\mu \to p^\mu + \delta p^\mu = p^\mu + \delta \epsilon^\mu{}_\nu p^\nu \ , \quad \epsilon^\mu{}_\nu = -\epsilon^\nu{}_\mu \tag{2.17}$$

and thus

$$J(\{p\}) = J(\{p+\delta p\}) = J(\{p\}) + \delta\epsilon^{\mu}{}_{\nu}\left(\sum_{i=1}^{n_E} p_i^{\nu}\frac{\partial}{\partial p_i^{\mu}}\right) J(\{p\}) + \mathcal{O}\left(\delta^2\right) . \tag{2.18}$$

Since the resulting $\epsilon^{\mu}{}_{\nu}\left(\sum_{i=1}^{n_E} p_i^{\nu}\frac{\partial}{\partial p_i^{\mu}}\right) J(\{p\}) = 0$ is valid for any $\epsilon^{\mu}{}_{\nu} = -\epsilon^{\nu}{}_{\mu}$, the $\epsilon^{\mu\nu}$ can be replaced by $p_a^{\mu}p_b^{\nu} - p_a^{\nu}p_b^{\mu}$ for arbitrary a, b and one obtains the relations

$$(p_a^{\mu}p_b^{\nu} - p_a^{\nu}p_b^{\mu})\left(\sum_{i=1}^{n_E} p_{i,\nu}\frac{\partial}{\partial p_i^{\mu}}\right) J(\{p\}) = 0. \tag{2.19}$$

For n_E independent external momenta there are $n_E(n_E-1)/2$ non-trivial antisymmetric combinations, each of which gives rise to one equation per seed integral.

2.3.4 Relations between different dimensions

All of the above methods to generate linear relations between integrals leave the space-time dimension of those integrals unchanged, but it is also possible to find relations between integrals with different space-time dimensions [53]. They are not needed in this thesis, but can be very useful if one wants to choose a basis of (quasi-)finite master integrals [54], which are generally more accessible to certain methods of evaluation (see sect. 3). In the following I will show Tarasov's method [53] of generating equations for integrals with different dimensions, which relies upon the Schwinger parametrisation [55]

$$\frac{1}{(D_j + i\epsilon)^{z_j}} = \frac{i^{-z_j}}{\Gamma(z_j)}\int_0^{\infty} \mathrm{d}x_j x_j^{z_j-1} e^{ix_j(D_j+i\epsilon)} . \tag{2.20}$$

Applying this parametrisation to all propagators of an integral $J^{(d)}(\{s_i\},\{m_i\})$ with dimension d, where $\{s_i\}$ is the set of scalar products of external momenta, transforms the integrations over loop momenta into Gaussian integrals. After performing these integrals one is left with

$$J^{(d)}\left(\{s_i\},\{m_i\}\right) = i^L\left(\frac{\pi}{i}\right)^{\frac{Ld}{2}}\prod_j \frac{i^{-z_j}}{\Gamma\left(z_j\right)}\int_0^{\infty}\frac{\mathrm{d}x_j\ x_j^{z_j-1}}{\left(\mathcal{U}(x)\right)^{d/2}} e^{\ i\left(\frac{\mathcal{F}(\{s_i\},x)}{\mathcal{U}(x)}+\sum\limits_l x_l(m_l^2+i\epsilon)\right)} , \tag{2.21}$$

where $\mathcal{U}$ and $\mathcal{F}$ are the first and second graph polynomials or Symanzik polynomials (see e.g. [56] or sect. 3.1) of the graph which can be drawn using the propagator momenta in J. The important thing to note at this point is that the dimension d only enters (non-trivially) as an exponent in the denominator. Assuming for a moment that all masses are

different, one constructs the differential operator

$$\mathcal{U}(\partial) = \mathcal{U}(x)|_{x_i \to \frac{\partial}{\partial m_i^2}} \tag{2.22}$$

and notes that

$$\mathcal{U}(\partial) e^{i \sum_l x_l m_l^2} = \mathcal{U}(x) i^L e^{i \sum_l x_l m_l^2} \ , \tag{2.23}$$

since all terms in $\mathcal{U}$ are of degree L. Acting upon the integral one finds

$$\mathcal{U}(\partial) J^{(d)}\left(\{s_i\}, \{m_i\}\right) = \pi^L J^{(d-2)}\left(\{s_i\}, \{m_i\}\right) \ . \tag{2.24}$$

The derivatives on the left hand side can be evaluated explicitly to yield a non-trivial equation for integrals with different dimensions, which holds for any choice of the masses m_i. Equations of the form (2.24) can be used to remove all propagators in the numerator, but at the cost of introducing dimensions with higher dimensions [53].

2.4 Reduction

Given a set of integrals and a system of linear relations between them, generated by the methods described in section 2.3, one wants to reduce the system to a form where all integrals are expressed in terms of a few master integrals. In other words, an algorithm is needed for the task

$$\left\{ \sum_j p_j I_j = 0 \right\} \overset{\text{reduction}}{\longrightarrow} \left\{ I_k = \sum_{j \in \text{ masters}} q_{kj} I_j \quad \forall k \right\}, \tag{2.25}$$

where the index j was introduced to label the different integrals and p_j and q_{kj} are in general rational functions of d, masses and kinematic variables. This should in principle be trivial to do, since it only involves Gaussian elimination. In practice however, it is observed that the run time and memory consumption of such a reduction often crucially depend on the implementation details of the algorithm used to solve the system. This is mainly due to the fact that the coefficients of the integrals can grow very large during the reduction, which slows down the rational algebra considerably. The growth of the coefficients is particularly bad when many steps[2] are performed on a set of equations where information is "missing", i.e. a particular linear combination X of integrals appearing in the equations is zero, but the information $X = 0$ is not contained in the set. In this case the spurious zero X can appear with arbitrary prefactors, which will typically grow much faster with every step than the rest of the coefficients, for which many cancellations

[2] In the context of reductions, one step is adding one equation to another in order to cancel a particular term.

occur. It is therefore of particular importance to choose the order in which integrals are eliminated and the order in which equations are considered in a way that minimises the amount of steps and missing information for intermediate stages of the reduction, where only part of the input equations have been treated.

An approach that works fairly well in many cases is the *Laporta algorithm*[3], which was first described in reference [1]. It has since been implemented in several public [57, 58, 59, 60] and private codes. Laporta proposed an ordering relation for the integrals based primarily on the parameters defined in (2.6). I will in the following use a very similar ordering, which I refer to as an ordering by difficulty. For two different integrals the higher difficulty is assigned to, in descending order of priority,

1. the one with greater t (number of denominators),
2. the one with lower sector ID,
3. the one with greater r (number of dots),
4. the one with greater s (numerator powers),
5. the one with greater $\max(z_i)$,
6. the one with lower $\min(z_i)$,
7. the one with greater z_1, z_2, ...

There are many possible variations of the above criteria, some of which perform slightly better for some reductions, but worse for others. The choice given here works sufficiently well for the tadpole integrals considered in this thesis. For a given reduction, the ordering relation will be used to assign an index to all integrals that appear in the reduction, where a lower index indicates higher difficulty.

To fix the order in which equations are considered, Laporta suggests to generate IBP-relations by starting with the simplest integral as seed integral, adding the information of the resulting equations to the system in the manner described below and then working up to more difficult integrals. Instead of following this approach, I will first generate all equations needed for the reduction without reducing them directly. The equations are then reduced in order of difficulty, starting from the simplest ones, where the difficulty of an equation is defined by the most difficult integral it contains. The resulting order of equations is similar to the one Laporta describes, but usually performs slightly better, since it only introduces new integrals to the system after all equations containing only simpler integrals have already been reduced.

[3] Laporta's algorithm is not to be confused with what I refer to as Laporta's method. The latter is a way of solving master integrals described in section 4.

Once the orders of integrals and equations are set, the Laporta algorithm can be described by a few simple steps:

1. Consider the next equation. If there are none left, stop.
2. Insert all solutions for integrals which have already been found.
3. If the equation is trivial ($0 = 0$), go to step 1.
4. Solve for the most difficult integral remaining and add the solution to the system of equations.
5. Use the newly found solution to eliminate the integral from all previously found solutions. Go back to step 1.

If one has generated sufficiently many equations, the above steps express most (if not all) integrals in the reduction in terms of a handful of master integrals. The exact number of masters depends on the class of integrals, but it can be proven that their number is always finite [61].

If not all integrals can be solved in terms of master integrals, information is typically missing on the edges of the (r, s)-plane of the reduction, where integrals have either $r = r_{\max}$ or $s = s_{\max}$. This is due mostly to the fact that IBP-relations generally contain integrals with both parameters r and s increased by 1 compared to their seed integral (see figure 2.3(a)). Since this means that integrals on the edge of the reduction cannot be used as seed integrals for IBP-relations, the equation density per integral in this area of the (r, s)-plane is lower and thus there is a higher probability for missing information. This can be remedied to some degree by also including equations generated via syzygies, which do not increase r (fig. 2.3(b)), and via mass derivatives, which do not increase s (fig. 2.3(c)). Even more important is the inclusion of all identities that can be found from momentum shifts, since they increase neither r nor s and have simpler coefficients due to the absence of the dimension d. I will always apply all symmetry and sector shift information to all equations before applying Laporta's algorithm, but it is also possible to simply consider symmetry shifts as additional input equations to the algorithm. The latter strategy can be advantageous when integrals with large parameters s are considered, as in this case momentum shifts can produce many terms due the expansion of sums in the numerator. Since IBP-equations reproduce some of the information contained in the symmetry relations and can in these situations be much smaller, enforcing the use of the latter before the former can be detrimental. For the fully massive tadpoles however I found that the benefits of simpler coefficients in the symmetry relations and the highly symmetric structure of the integrals due to the absence of external momenta outweigh this problem for all values of s needed in the reductions.

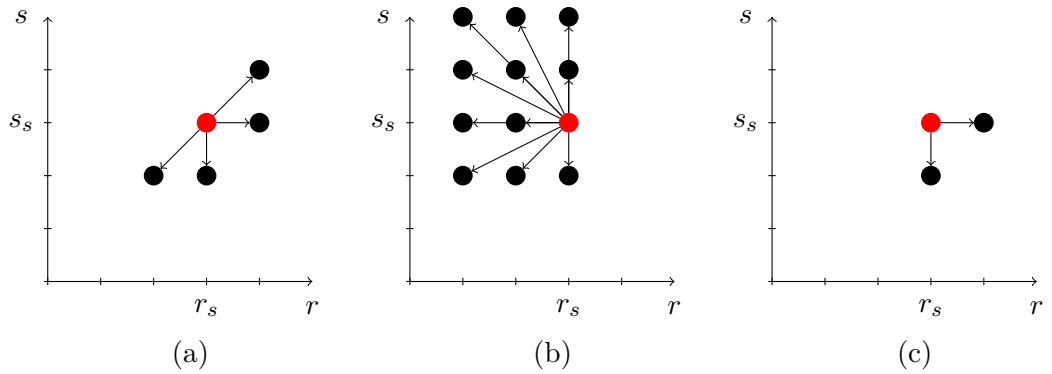


Figure 2.3: Diagrams of linear equations in the (r, s)-plane. The dots represent possible (r, s)-values for equations generated via IBP (a), syzygies (b) or mass derivatives (c), where r_s and s_s are the parameters of the seed integral.

One of the weaknesses of Laporta's algorithm is that despite the carefully chosen orders of integrals and equations large rational coefficients can still appear in intermediate stages of the reduction and slow down the computation. A new approach was recently suggested by von Manteuffel and Schabinger [62] to circumvent this problem. They propose replacing all variables with numbers and reducing the system over a finite (machine-sized) prime field. If one performs enough runs with different replacements and different fields, the full rational function content of the reduced system can be reconstructed from the results of those runs. This avoids the expression swell that can happen in normal reductions, since all coefficients in intermediate stages are simply machine-sized integers. A public implementation of this idea does not exist as of yet. A different application of numerical replacements for variables and finite fields is presented in section 8.1.4, where it is used to minimise the number of redundant equations (also see [63]). This can be useful to speed up the reduction, since $L(L + n_E)$ IBP-relations per seed integral plus equations generated by other methods often lead to systems where less than 10% of all equations actually contribute to the final result.

2.5 Example: 4-loop sunset

In order to demonstrate how most of the steps outlined in this section work in practice, I will show some examples for integrals belonging to the 4-loop sunset diagram in figure 2.4 with all masses equal to 1. For simplicity I do not start from general tensor integrals, but assume that the projection to scalar integrals has already been performed and all integrals are expressed in a classification scheme as defined in section 2.1. Since for 4 loops and no external momenta there are 10 dot products of momenta, there are $2^{10} = 1024$ different sectors. Choosing the set of propagator momenta as (compare table A.1)

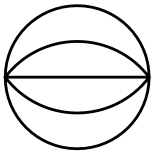


Figure 2.4: The 4-loop sunset graph.

$$A_4 = \{k_1, k_2, k_3, k_4, k_1 - k_4, k_2 - k_4, k_3 - k_4, k_1 - k_2, k_1 - k_3, k_1 - k_2 - k_3\}, \qquad (2.26)$$

63 of the 1024 are anti-sectors without an associated graph and another 281 are zero-sectors for which all integrals vanish, which leaves 680 physical sectors. 15 of these 680 sectors have momenta which can be mapped onto the sunset diagram. For a reduction up to $r_{\max} = s_{\max} = 4$ this means one would have to consider 220500 integrals for the sunset. Using sector momentum shifts to map all 15 sectors to the one with the highest ID (sector 841) reduces this number to 14700. An example of such a mapping would be the momentum transformation $(k_1, k_2, k_3, k_4) \to (k_3, k_1, -k_2, k_4)$ which leads to

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}
$\to$	D_3	D_1	D_2	D_4	D_7	D_5	$2D_2 + 2D_4$ $-D_6 - 2$	D_9	$D_1 + D_2 + D_3 - D_8$ $-D_9 + D_{10} - 1$	D_{10}

and thus maps sector 481 (which has lines 2,3,4,5,10) to sector 841 (lines 1,2,4,7,10). Out of the propagators D_i which can have negative exponents for sector 481, D_7 and D_9 transform non-trivially and thus for $z_7, z_9 < 0$ the shift would map a single integral to a sum of integrals, e.g.

$$\begin{aligned} I(0,1,1,1,1,0,-1,0,0,1) = 2{*}I(1,0,0,1,0,0,1,0,0,1) + 2 * I(1,1,0,0,0,0,1,0,0,1) \\ -I(1,1,0,1,0,-1,1,0,0,1) - 2 * I(1,1,0,1,0,0,1,0,0,1)\ . \end{aligned}$$

It is obvious that for negative exponents with large absolute value shifts like the above can produce a great number of terms, since all products of sums in the numerator have to be expanded to stay within the classification scheme. This is a problem for which no solution is known as of yet. In the example given here one also finds two integrals with $t < 5$, which belong to subsectors and have to be shifted once more to map them to their respective representative sectors. In general this can lead to further proliferation of terms,

but in this simple case no further sums appear and the mapping simplifies to

$$I(0,1,1,1,1,0,-1,0,0,1) = 4*I(1,1,1,1,0,0,0,0,0,0) \\ -I(1,1,0,1,0,-1,1,0,0,1) - 2*I(1,1,0,1,0,0,1,0,0,1) \ .$$

As a next step the symmetries of sector 841 can be applied, which reduces the number of unsolved integrals for the sunset diagram in the example from 14700 to a mere 467. As it turns out, the integral $I(1,1,0,1,0,-1,1,0,0,1)$ on the right hand side above can be shifted to simpler integrals using symmetries in such a way that the mapping for the original integral reduces to

$$I(0,1,1,1,1,0,-1,0,0,1) = I(1,1,0,1,0,0,1,0,-1,1) \ ,$$

where the right hand side cannot be further simplified by any symmetry relation. This identity could have also been found directly from the momentum shift $(k_1,k_2,k_3,k_4) \to (k_1-k_2,k_3-k_4,k_1-k_2-k_3,-k_2)$, but in practice it is not feasible to try all combinations of sector shifts and symmetries for all integrals to always find the best shift immediately.

As a last step, the reduction for sector 841 reveals that the sector contains two master integrals and expresses all others in terms of these two and a single master integral of the subsector 960. The integral I started out with is then finally given by

$$I(0,1,1,1,1,0,-1,0,0,1) = -\frac{1}{2}I(1,1,0,1,0,0,1,0,0,1) + \frac{3}{2}I(1,1,1,1,0,0,0,0,0,0) \ ,$$

where only one of the master integrals of sector 841 appears. Even though it is unlikely that all or even most of the initial 220500 integrals of the sunset diagram would appear in any actual calculation, the fact that all of them can be expressed as linear combinations of just three integrals shows the importance of the reduction, since it reduces the number of integrals that need to be explicitly solved by several orders of magnitude.

3 Solving master integrals

Once the reduction has expressed all needed integrals in terms of master integrals, the latter still need to be solved. A great number of strategies for this task has been developed over the years, each with different advantages and ranges of applicability. In this section I give a brief overview of some commonly used methods, but the list presented here is by no means exhaustive. The overview does not contain Laporta's method of difference equations and factorial series, as it is discussed separately and in much greater detail in the following sections.

For some of the simplest Feynman integrals the integration can be carried out directly. This includes the 1-loop integrals of the class of massive vacuum integrals considered in this thesis, which are given by

$$I(n) = \int_{k_1} \frac{1}{(k_1^2+1)^n} , \quad n \in \mathbb{N} . \tag{3.1}$$

By using e.g. the IBP-relation with operator $\frac{\partial}{\partial k_1^\mu} k_1^\mu$ one finds

$$I(n+1) = \frac{2n-d}{2n} I(n) \tag{3.2}$$

and thus all 1-loop massive tadpoles can be reduced to the master integral $J \equiv I(1)$. In this simple example, however, the reduction is not even needed, as the angular and radial parts of the integration in eq. (3.1) are trivially separated and can be performed in a straightforward manner to obtain the result in terms of Γ-functions as

$$\begin{aligned} I(n) &= \frac{\Gamma(n-\frac{d}{2})}{\Gamma(n)} , \\ J \equiv I(1) &= \Gamma(1-d/2) . \end{aligned} \tag{3.3}$$

The master integral J will be used in later sections to normalise higher-loop results in order to remove the dependence on the convention for the integration measure. For more complicated integrals the straightforward integration no longer works and more involved strategies are needed.

3.1 Feynman parameters

A scalar integral I with L loops in the classification introduced in section 2.1 has the form

$$I \equiv I(z_1, \ldots, z_w) = \int_{k_1,\ldots,k_L} \frac{1}{D_1^{z_1} D_2^{z_2} \ldots D_w^{z_w}} , \tag{3.4}$$

where performing the momentum integrations is generally difficult, because the loop momenta appear in more than one propagator. A nice trick to consolidate the different propagators was suggested by Feynman [55], who noted that

$$\frac{1}{D_1^{z_1} D_2^{z_2} \dots D_w^{z_w}} = \underbrace{\int_0^1 \mathrm{d}x_1 \dots \mathrm{d}x_w \delta\left(1 - \sum x_i\right)}_{\equiv \int_x} \frac{\prod x_i^{z_i - 1}}{(\sum x_i D_i)^z} \frac{\Gamma(z)}{\prod \Gamma(z_i)} \,, \tag{3.5}$$

where $z = \sum_{i=1}^{w} z_i$. When applied to I, the denominator takes the form

$$\sum x_i D_i = k^T M k + 2P \cdot k + Q \,, \tag{3.6}$$

where k is the L-vector of loop momenta, M is an $L \times L$ matrix, P an L-vector of external momenta and Q a scalar depending on both external momenta and masses. In this form the integration over the loop momenta can be performed and one obtains

$$I = \frac{\Gamma(z - \frac{Ld}{2})}{\prod \Gamma(z_i)} \int_x \frac{[\mathcal{U}(x)]^{z-(L+1)\frac{d}{2}}}{[\mathcal{F}(x)]^{z-L\frac{d}{2}}} \prod x_i^{z_i - 1} \,, \tag{3.7}$$

where $\mathcal{U}$ and $\mathcal{F}$ are the graph polynomials already encountered in section 2.3.4 in the closely related Schwinger parametrisation. Here they can be expressed as

$$\begin{aligned} \mathcal{U}(x) &= \det(M) \,, \\ \mathcal{F}(x) &= \det(M)(P^T M^{-1} P + Q) \,. \end{aligned} \tag{3.8}$$

The sum over all parameters x_i in the δ-function of $\int_x$ can also be performed over an arbitrary, non-empty subset λ of the parameters, if the integration range over the remaining parameters is extended to infinity [64], such that

$$\int_x = \int_0^\infty \mathrm{d}x_1 \dots \mathrm{d}x_w \delta\left(1 - \sum_{x_i \in \lambda} x_i\right) \,, \quad \lambda \subset \{x_1, \dots, x_w\} \,, |\lambda| > 0 \,. \tag{3.9}$$

For the fully massive tadpole integrals with a single mass $m = 1$ one finds $P = 0$ and $Q = \sum_i x_i$ and thus $\mathcal{F} = \mathcal{U}$ if λ is chosen as the full set of parameters.

In some cases the remaining integrals over the Feynman parameters x_i can be performed analytically, but one of the main advantages of this parametrisation is that it is more accessible to numerical integration than the loop momentum representation. If the integral is finite, this integration can be carried out with methods such as Monte-Carlo, but the precision that can be achieved in this way is rather limited for higher loop integrals, as the number of integration variables is given by the number of propagators. For divergent integrals one first has to extract the poles in ϵ, e.g. via sector decomposition.

3.2 Sector decomposition

Sector decomposition is a strategy to extract divergences from the Feynman parametric representation of integrals introduced by Binoth and Heinrich [65, 66], which has since been implemented in the publicly available codes `sector_decomposition` [67], SecDec [68, 69, 70] and FIESTA [71, 72, 73]. The best way to illustrate the idea behind it is an example taken from [66], in which it is applied to the integral

$$I = \int_0^1 \mathrm{d}x \int_0^1 \mathrm{d}y\, x^{-1-a\epsilon} y^{-b\epsilon} \left(x + (1-x)y\right)^{-1} \ . \tag{3.10}$$

The main problem here is the term $x+(1-x)y$, which leads to a divergence when x and y go to zero simultaneously. The goal of sector decomposition is to factorise the divergences into singular factors of just one integration variable. To this end the integration over the w-dimensional hypercube (where in this case $w = 2$) is split into different sectors[4] by introducing into the integrand unity composed of step functions. This is done in a way such that for each integration variable there is one sector where this variable is always greater than all others. In the above example one would thus split the integral into two sectors by inserting $1 = \Theta(x-y) + \Theta(y-x)$. The integration ranges can be remapped to the hypercube by rescaling $y \to xy$ in the first sector and $x \to yx$ in the second sector, after which the integral reads

$$\begin{aligned} I = &\int_0^1 \mathrm{d}x\, x^{-1-(a+b)\epsilon} \int_0^1 \mathrm{d}y\, y^{-b\epsilon} \left(1 + (1-x)y\right)^{-1} && (\equiv I_1) \\ &+ \int_0^1 \mathrm{d}y\, y^{-1-(a+b)\epsilon} \int_0^1 \mathrm{d}x\, x^{-1-a\epsilon} \left(1 + (1-y)x\right)^{-1} && (\equiv I_2) \ . \end{aligned} \tag{3.11}$$

Looking e.g. at the first integral I_1, the term $1+(1-x)y$ no longer leads to any divergences, as these now only originate from the singular factor $x^{-1-(a+b)\epsilon}$. This can be used to separate the $1/\epsilon$ pole by extracting the leading order in the Taylor expansion in x of the remaining integral, such that I_1 is given by

$$\begin{aligned} I_1 &= \int_0^1 \mathrm{d}x\, x^{-1-(a+b)\epsilon} \underbrace{\int_0^1 \mathrm{d}y\, y^{-b\epsilon} \left(1 + (1-x)y\right)^{-1}}_{\equiv f(x)} \\ &= \int_0^1 \mathrm{d}x\, x^{-1-(a+b)\epsilon} f(0) + \int_0^1 \mathrm{d}x\, x^{-1-(a+b)\epsilon} \left(f(x) - f(0)\right) \\ &= -\frac{f(0)}{(a+b)\epsilon} + \int_0^1 \mathrm{d}x\, x^{-1-(a+b)\epsilon} \left(f(x) - f(0)\right) \ . \end{aligned} \tag{3.12}$$

The remaining integral is now finite and can be computed numerically.

[4] The word sector refers to a part of the hypercube here and is not to be confused with the concept of a sector in the integral classification scheme.

The main advantages of sector decomposition are its applicability to many classes of integrals, the high degree of automation and the ease of use through the publicly available codes. However, in more complicated cases than the above example the decomposition into multiple sectors has to be iterated several times, which can lead to an exponential proliferation of terms and slow down the computation considerably, especially for integrals with many propagators. In combination with the limited precision from the numerical evaluation this makes sector decomposition a less favourable choice for the 5-loop tadpoles than Laporta's method, which can easily achieve precisions of thousands of digits for many integrals (see sections 4 and 7).

3.3 Parametric integration and linear reducibility

A large number of Feynman integrals can be expressed in terms of multiple polylogarithms, which are defined as

$$\mathrm{Li}_{n_1,\ldots,n_r}(y_1,\ldots,y_r) = \sum_{0<k_1<\cdots<k_r} \frac{y_1^{k_1}\ldots y_r^{k_r}}{k_1^{n_1}\ldots k_r^{n_r}} \,. \tag{3.13}$$

Special values of multiple polylogarithms include e.g. the (multiple) ζ-functions [74]

$$\zeta_{n_1,\ldots,n_r} = \mathrm{Li}_{n_1,\ldots,n_r}(1,\ldots,1) \tag{3.14}$$

and harmonic polylogarithms [75]

$$H_{n_1,\ldots,n_r}(y) = \mathrm{Li}_{n_1,\ldots,n_r}(y,1,\ldots,1) \,. \tag{3.15}$$

Another way of expressing multiple polylogarithms are the so-called Goncharov polylogarithms (GPLs, [76]) $G_{n_1,\ldots,n_r}(y_1,\ldots,y_r;y)$, which describe the same class of functions and are connected by

$$\mathrm{Li}_{n_1,\ldots,n_r}(y_1,\ldots,y_r) = (-1)^r G_{n_1,\ldots,n_r}\left(\frac{1}{y_1},\frac{1}{y_1y_2},\ldots,\frac{1}{y_1\ldots y_r};1\right) \,. \tag{3.16}$$

The GPLs are defined by iterated integrals as

$$\begin{aligned} G_{n_1,\ldots,n_r}(y_1,\ldots,y_r;y) &= G(\underbrace{0,\ldots,0}_{n_1-1},y_1,\ldots,y_{r-1},\underbrace{0,\ldots,0}_{n_r-1},y_r;y) \,, \\ G(y_1,\ldots,y_r;y) &= \int_0^y \frac{\mathrm{d}x_r}{x_r-y_r}\int_0^{x_r}\frac{\mathrm{d}x_{r-1}}{x_{r-1}-y_{r-1}}\ldots\int_0^{x_2}\frac{\mathrm{d}x_1}{x_1-y_1} \\ &= \int_0^x \frac{\mathrm{d}x_r}{x_r-y_r} G(y_1,\ldots,y_{r-1};x_r) \,. \end{aligned} \tag{3.17}$$

In the Feynman parametric representation, the GPLs can be used to solve many finite integrals analytically [77, 78, 79]. After choosing the set λ in eq. (3.9) as a single parameter x_λ, this parameter is set to one by the δ-function and the integrations over the remaining parameters x_i range from zero to infinity. Expanding in ϵ yields integrands which are rational functions in x_i, $\mathcal{U}$, $\mathcal{F}$ and logarithms thereof. The strategy for each order in ϵ is to successively perform the integrals over the parameters x_i by repeatedly bringing the integrand into the form of the last line in eq. (3.17) for one of the parameters. As long as the polynomials that appear can be factorised into linear functions of the next integration parameter x_i (as is the case for the initial polynomials $\mathcal{U}$ and $\mathcal{F}$), this can be done by partial fraction decomposition. The integration over x_i then yields another GPL, which has to be evaluated at zero and infinity, before once again bringing the integrand into the required form for the next parameter. The procedure can be repeated until all integrations have been performed or non-linear factors in the next integration variable appear. Whether all integrations can be performed often depends on the choices for λ and the order of integration over the x_i. In some cases a different parametrisation can also help to avoid non-linear factors [78]. If some parametrisation and ordering of the integration variables exists such that all factors throughout the process are linear in the next integration variable, the integral is called *linearly reducible*. In this case all finite integrals associated with the same graph are linearly reducible, as different exponents z_i in the loop momentum representation only influence the exponents of the polynomials in the representation, but not the (linear) factors of the polynomials. A public implementation in Maple$^{\text{TM}}$ [80] has been presented in [81] that automates the check for linear reducibility and performs the integration where possible.

Parametric integration of linearly reducible integrals via GPLs is a powerful method and has been used to obtain many results [77, 78, 79, 81], but it is limited to integrals that can be expressed as multiple polylogarithms. This makes it infeasible for the 5-loop massive tadpoles considered here, as it is expected from lower loop results that their ϵ-expansions contain at least elliptic integrals [4, 82], which are not contained in the class of multiple polylogarithms. The restriction to finite integrals is not a major problem, as it is always possible to choose a basis of (quasi-)finite master integrals [54], though this may introduce integrals with higher dimensions.

3.4 Mellin-Barnes representations

For a detailed discussion of Mellin-Barnes representations I refer to [83], which most of the following summary is based on. The key ingredient of this method is a decomposition which transforms a sum $A + B$ raised to some power to a Mellin-Barnes integral [84] over a contour $\mathcal{C}$ in the complex plane, in which A and B appear as separate factors in the integrand:

$$\frac{1}{(A+B)^\lambda} = \frac{1}{\Gamma(\lambda)} \frac{1}{2\pi i} \int_{-i\infty}^{+i\infty} \mathrm{d}z\, \Gamma(\lambda+z)\Gamma(-z) \frac{B^z}{A^{\lambda+z}} \,. \tag{3.18}$$

The contour $\mathcal{C}$ is chosen such that it divides the poles of the Γ-functions appearing in this kind of integral into two groups. Poles of $\Gamma(\cdots + z)$ are on the left side of $\mathcal{C}$ (*left poles*), while poles of $\Gamma(\cdots - z)$ are on the right side of $\mathcal{C}$ (*right poles*). The above decomposition of a sum can be applied (multiple times) to Feynman integrals e.g. to separate mass terms from propagators in the loop momentum representation or to split the graph polynomials in the Feynman parametric representation into monomials. The integrals over loop momenta or parameters x_i can then often be evaluated in terms of more Γ-functions.

A simple example is given in [83], where the decomposition is applied to the massive propagator of the 1-loop 2-point integral

$$F(q^2, m^2) = \int_k \frac{1}{(k^2+m^2)^2 (k-q)^2} \,. \tag{3.19}$$

The integration over k can then be performed in $d = 4$ dimensions and yields

$$F(q^2, m^2)|_{d=4} = \frac{1}{q^2} \frac{1}{2\pi i} \int_{\mathcal{C}} \mathrm{d}z \left(\frac{m^2}{q^2}\right)^z \frac{\Gamma(1+z)\Gamma(-z)^2}{\Gamma(1-z)} \,. \tag{3.20}$$

The left poles of this integral are at $z \in \{-1, -2, \dots\}$ and the right poles at $z \in \{0, 1, \dots\}$ and thus the contour $\mathcal{C}$ can be chosen to be parallel to the imaginary axis with $-1 < \mathrm{Re}(z) < 0$. After closing the contour on the right, the Mellin-Barnes integral can be evaluated by Cauchy's theorem in terms of the residues of the right poles. They give the contributions

$$\begin{array}{ll} \dfrac{\ln\left(q^2/m^2\right)}{q^2} & \text{from } z = 0 \text{ and} \\ \dfrac{1}{q^2} \displaystyle\sum_{n=1}^{\infty} \frac{1}{n} \left(-\frac{m^2}{q^2}\right)^n & \text{from } z = 1, 2, \dots \end{array} \tag{3.21}$$

and thus one finds

$$F(q^2, m^2)|_{d=4} = -\frac{\ln(1 + q^2/m^2)}{q^2} \,. \tag{3.22}$$

For divergent integrals the Mellin-Barnes representation will have left and right poles which are separated by ϵ, e.g. in the term $\Gamma(\epsilon + z)\Gamma(-z)$ for $z = 0$ and $z = -\epsilon$. In the

limit $\epsilon \to 0$ there is no space between these poles for the integration contour. This can be resolved by moving the contour to the left of the pole at $-\epsilon$, thus making it a right pole, and including its residue in the result. This residue then contains the divergence and the remaining integrand can be safely expanded in ϵ without merging left and right poles.

For more complicated integrals than in the above example one usually arrives at multi-dimensional Mellin-Barnes integrals, for which choosing the contour becomes more difficult. The evaluation in terms of residues as above generally leads to multiple infinite sums, which are not always solvable analytically. In some cases it is also possible to perform the integration directly via Barnes' lemma [84]. A third option is to evaluate the integrals numerically. A great number of tools such as `MB` [85], `MBresolve` [86] or `AMBRE` [87, 88, 89] have been published to automate many tasks related to Mellin-Barnes integrals, as well as for the summation methods needed for the multiple infinite sums (see. e.g. [90]). These tools and Mellin-Barnes representations in general have been successfully used to solve many Feynman integrals analytically (see e.g. [83, 91] and references therein). However, for complicated integrals the dimensionality (and thus complexity) of the Mellin-Barnes integrals is often much higher than the dimensionality of the Feynman parameter integral (i.e. the number of lines of the corresponding graph) [92] and non-planar massive integrals in particular pose a problem [91], which means the method is not ideally suited for the 5-loop fully massive tadpoles considered in this thesis.

3.5 Differential equations

Differential equations have been widely used as a tool for solving many different Feynman integrals (see e.g. [93, 94, 95, 47], also [96] for a recent review). The idea relies on the fact that for a given class of integrals a reduction to a finite set of master integrals is available. Taking the derivative of these master integrals with respect to a kinematic invariant will result in a linear combination of integrals of the same class, which can once again be expressed in terms of the master integrals. One finds

$$\partial_i f(x,\epsilon) = A^{(i)}(x,\epsilon) f(x,\epsilon) \,, \quad \partial_i = \frac{\partial}{\partial x_i} \,, \quad i = 1,\ldots,m \,, \tag{3.23}$$

where $f = (f_1,\ldots,f_n)$ is the vector of master integrals, $x = \{x_1,\ldots,x_m\}$ is the set of kinematic invariants and $A^{(i)}$ is an $n \times n$-matrix which is rational in the x_i and the dimensional regularisation parameter ϵ. The master integrals thus satisfy a set of partial first order differential equations.

In the following I will for simplicity consider differential equations in only a single kinematic invariant y, but the generalisation to multiple variables is straightforward [96]. Recently, Henn [97] proposed that the differential equations could often be simplified by choosing

an appropriate basis of master integrals. By changing to a new basis

$$g = Tf \tag{3.24}$$

with an invertible matrix T the differential equations can be written as

$$\partial_y g(y,\epsilon) = B(y,\epsilon) g(y,\epsilon) \ , \tag{3.25}$$

where

$$B = T^{-1} A T - T^{-1} \partial_y T \ . \tag{3.26}$$

This freedom can be used to simplify the new matrix B compared to A. In particular one would like to find a so-called *canonical basis*, in which a differential form of eq. (3.25) reads

$$\begin{aligned} \mathrm{d}g(y,\epsilon) &= \epsilon \left(\mathrm{d}B'\right) g(y,\epsilon) \ , \\ B' &= \sum_k B_k \log \alpha_k(y) \end{aligned} \tag{3.27}$$

where B_k are matrices independent of y and ϵ and $\alpha_k(y)$ are algebraic functions of y. If the transformation T is rational, $\alpha_k(y) = y - y_k$, where the y_k are the singularities of the differential equation, which then reads

$$\partial_y g(y,\epsilon) = \epsilon \sum_k \frac{B_k}{y - y_k} g(y,\epsilon) \ . \tag{3.28}$$

The big advantage of this form is the simple dependence on ϵ, which allows for successive rather than simultaneous evaluation of the ϵ-orders. In the rational case the integrals can be solved to all orders in ϵ in terms of Goncharov polylogarithms (see eq. (3.17)), in the general case in terms of Chen iterated integrals [98]. Several strategies have been developed to arrive at a canonical basis [99, 100, 101, 102, 103, 96], but this is not always possible, as e.g. for elliptic integrals the ϵ^0 term in the matrix B cannot be removed and thus ϵ cannot be factored out [96].

Regardless of whether a canonical basis is used or not, the solution of the differential equations still requires boundary conditions to fix the integration constants. This may in part be done by using prior knowledge of the singularities of the integrals [96], or by evaluating them at a fixed value of y, which may still be non-trivial. For single scale integrals like the tadpole integrals considered in this thesis the scale dependence is trivial and thus differentiating with respect to that scale does not yield new information. In this case one can introduce an additional artificial scale in such a way that the original integrals are reproduced in some limit.

3.6 Dimensional recurrence relations

The method of using recurrence relations with the space-time dimension d as their variable has been developed in [53, 104, 105] and has since been applied to various single-scale integrals (see e.g. [106, 107, 108, 109]). It relies on equations between integrals with different dimensions (see sect. 2.3.4) and the reduction to master integrals. By combining the two one can find a recurrence relation

$$f^{(d-2)} = C(d) f^{(d)} + R(d) \,, \tag{3.29}$$

where $f^{(d)} = \{f_1, \ldots, f_n\}$ is the vector of master integrals of a given sector (in d dimensions), $C(d)$ is an $n \times n$-matrix with rational coefficients in d and $R(d)$ depends on master integrals of subsectors, which are assumed to be known analytically. For simplicity I consider in the following the approach outlined in [105] for the case that the sector only has a single master integral I, such that $f^{(d)} = I^{(d)}$ and C and R are known scalar functions. C can be written as

$$C(d) = c \frac{\prod_{i=1}^{m_a} (a_i - d/2)}{\prod_{j=1}^{m_b} (b_j - d/2)} \tag{3.30}$$

where the a_i, b_j and c are constant. After decomposing the subsector solutions into

$$R(d) = R_+(d-2) + R_-(d) \,, \tag{3.31}$$

such that

$$\begin{aligned} \left| \lim_{d \to +\infty} \frac{C(d+2) R_+(d+2)}{R_+(d)} \right| < 1 \,, \\ \left| \lim_{d \to -\infty} \frac{R_-(d-2)}{C(d-2) R_+(d)} \right| < 1 \,, \end{aligned} \tag{3.32}$$

the general solution of the recurrence relation can be written as [105]

$$\begin{aligned} I^{(d)} &= \omega(d) c^{-d/2} \frac{\prod_{i=1}^{m_a} \Gamma(a_i - d/2)}{\prod_{j=1}^{m_b} \Gamma(b_j - d/2)} + \sum_{k=0}^{\infty} \left[s_+(d,k) - s_-(d,k) \right] \,, \\ s_+(d,k) &= (-1)^{k(m_a+m_b)} c^k \frac{\prod_{i=1}^{m_a} (d/2 + 1 - a_i)^{\overline{k}}}{\prod_{j=1}^{m_b} (d/2 + 1 - b_j)^{\overline{k}}} R_+(d+2k) \,, \\ s_-(d,k) &= c^{-k-1} \frac{\prod_{j=1}^{m_b} (b_j - d/2)^{\overline{k+1}}}{\prod_{i=1}^{m_a} (a_i - d/2)^{\overline{k+1}}} R_-(d-2k) \,, \end{aligned} \tag{3.33}$$

where $(x)^{\overline{k}} = x(x+1)\ldots(x+k-1)$ is the Pochhammer symbol. $\omega(d) = \omega(d+2)$ is an arbitrary periodic function of d and is the only remaining unknown quantity in the solution at this point. It can be determined in two steps [105]. By analysing analytical proper-

ties of $I^{(d)}$ and the known functions in the solutions, such as poles in an arbitrary stripe $S = \{d | d_0 < \mathrm{Re}(d) \leq d_0 + 2\}$ of the complex plane of d or the behaviour for $\mathrm{Im}(d) \to \pm\infty$, $\omega(d)$ can be fixed up to some constants. These constants are then determined by performing the integral at some convenient dimension d.

The infinite sums in the solution converge exponentially, which is convenient for numerical evaluation, however the results need to be known analytically if the master integrals appear in the inhomogeneous part of recurrence relations for sectors with more propagators. Very recently the Mathematica [110] package `SummerTime` [111] has been published, which performs these sums numerically with high precision. If a basis of functions and numbers (e.g. multiple polylogarithms) that may appear in the result is known[5], the algorithm PSLQ [112, 113] may be used to obtain the analytical result from the high precision numerical one.

[5]This is not yet the case for the 5-loop massive tadpoles.

4 Laporta's method

In this section I begin discussing the method I use for solving the fully massive tadpole master integrals introduced in section 2.1. It is based on Laporta's approach built on difference equations and factorial series [1] (inspired by [114]), which I have improved in several key aspects to make it feasible for 5-loop calculations. In the following I shortly review the main ideas of Laporta's method as well as some of its advantages and limitations compared with other methods, before presenting in detail the individual steps and my improvements in sections 5 through 7.

Henceforth I assume that the reduction of integrals described in section 2.4 is already under control and the master integrals are known (but not solved). Laporta's approach then is a hierarchical one, solving first the master integrals of the simplest sectors with the fewest propagators and then working upwards by increasing the number of propagators, where the solution of the master integrals requires that the subsector master integrals are already known. As a starting point, all integrals are generalised by replacing one of their positive propagator exponents by a variable x, while all other exponents remain integers, e.g.

$$I(1,1,0,1,0,0,1,0,0,1) \to I(x,1,0,1,0,0,1,0,0,1) = \int_{\{k\}} \frac{1}{D_1^x D_2 D_4 D_7 D_{10}} \tag{4.1}$$

for one of the master integrals of the 4-loop sunset sector 841 discussed in section 2.5. In general it can make a difference which propagator the x is put on (see sect. 5), but because the sunset diagram is fully symmetric, all choices are equivalent in this case.

Since Laporta's method allows for solving the master integrals one by one, I will in the following simply write $I(x)$ for the integral that needs to be solved and $J_i(x)$, $i = 1, 2, \ldots$ for subsector integrals which are already solved. To obtain the solution, one must first find a *difference equation* for $I(x)$ of the form

$$\sum_{k=0}^{R} p_k(x) I(x+k) = \sum_i \sum_{k=0}^{R_i} p_{ik}(x) J_i(x+k) \; . \tag{4.2}$$

Here the p_k and p_{ik} are rational functions of x and d, but for more general integrals they would also depend on masses and external momenta. R is called the order of the equation and I will refer to the k in $I(x+k)$ as the *offset* of the integral. Details on how to obtain difference equations will be given in section 5.

Laporta then proposes to expand the integrals as factorial series of the first kind [114],

such that

$$I^{(m)}(x) = \sum_{s=0}^{\infty} \frac{\Gamma(x+1)}{\Gamma(x-K_m+s+1)} a_s^{(m)} \tag{4.3}$$
$$= \frac{\Gamma(x+1)}{\Gamma(x-K_m+1)} \left(a_0^{(m)} + \frac{a_1^{(m)}}{x-K_m+1} + \frac{a_2^{(m)}}{(x-K_m+1)(x-K_m+2)} + \dots \right),$$

where m labels different solutions of the difference equation (4.2) and its homogeneous version (without the J_i). The general solution can then be expressed as a sum over the homogeneous solutions and the inhomogeneous solution as

$$I(x) = \sum_m \mu_m^x \omega_m(x) I^{(m)}(x) \,, \tag{4.4}$$

which for the integrals in this thesis can still be simplified significantly. The $\omega_m(x)$ are periodic functions with period 1, which for a truly general solution of $I(x)$ would have to be fixed over one whole period. Since one is typically only interested in integer values of x, these functions can be chosen constant and are absorbed into the normalisation of the $a_s^{(m)}$. It turns out (see sect. 7.3) that for the tadpole integrals of interest here it suffices to look at solutions with $\mu_m = 1$ and $K_m = -d/2$. The expansion for the integral then reduces to

$$I(x) = \sum_{s=0}^{\infty} \frac{\Gamma(x+1)}{\Gamma(x+d/2+s+1)} a_s \,, \tag{4.5}$$

where only the coefficients a_s of the factorial series remain unknown.

By inserting the factorial series into the difference equation (4.2), the latter can be translated into a recurrence relation

$$\sum_{k=0}^{R'} g_k(s) a_{s+k} = \sum_i \sum_{k=0}^{R'_i} g_{ik}(s) a_{i,s+k} \tag{4.6}$$

for the coefficients a_s. The translation process will be explained in more detail in section 6.1. On the right hand side the $a_{i,s}$ are the coefficients of the factorial series expansions of the subsector integrals $J_i(x)$, which are already known. As in the difference equation, the g_k and g_{ik} are rational functions, but now depend on s and d. To avoid confusion due to the similar structures of eqs. (4.2) and (4.6), I will always refer to equations for integrals as difference equations and to equations for the factorial series coefficients as recurrence relations, even though in the literature these terms are often used interchangeably[6].

[6]Technically speaking eq. (4.2) is a recurrence relation, but not a difference equation. Since any recurrence relation can be expressed as a difference equation, the term has already been established in this function in the literature and to avoid confusion, I will stick with calling it a difference equation.

For further progress one needs to fix some of the a_s as initial conditions. Laporta describes how to do this for various kinds of integrals in reference [1] by examining the large-x behaviour of the integrals. In this way one can obtain values for the first few coefficients $a_0, a_1, \ldots, a_{R'-1}$ of the series, which for the massive tadpoles always take on the form of linear combinations of integrals with one loop less. Since these are assumed to be known, the recurrence relation (4.6) can then be used to iteratively evaluate the next terms $a_R, a_{R+1}, \ldots$ of the series. At this point it is usually impractical to continue working analytically, since a closed form for the solution of an arbitrary a_s is generally very difficult to find and each iteration of the recurrence relation increases the degree in d of the rational functions found in the exact expression for the a_s. The expression swell this would result in can be avoided by expanding everything around the dimensional regularisation parameter ϵ up to a finite order and switching to numerical values instead of rational numbers with ever-growing integer contents. For numerical calculation it is then sufficient to evaluate the factorial series coefficients a_s up to some finite maximum value $s = s_{\max}$.

Performing the sum over the factorial series requires a choice for the exponent x. Typically one is interested in $x = 1$, since in most cases $I(1)$ is the master integral one wants to solve. It is obvious from eq. (4.3) however that the series has a much better convergence behaviour for large x. While some factorial series might actually converge even for $x = 1$ (see sect. 7.4 for details on convergence), choosing a larger value for x in the very least allows for a lower $s_{\max}$ and thus fewer iterations of the recurrence relation to reach the same accuracy due to the faster convergence. After picking a maximum value $x_{\max}$ and evaluating the sum over the factorial series to obtain $I(x)$ for $x \in \{x_{\max} - R + 1, \ldots, x_{\max} - 1, x_{\max}\}$, the difference equation can be used iteratively to push down the argument of $I(x)$ from $x_{\max}$ to 1, at which point the solution for the master integral $I(1)$ is recovered. A summary of the steps in Laporta's method can be found in figure 4.1.

In most sectors it is possible to express all master integrals in the reduction in terms of $I(x = 1), I(x = 2), \ldots$. In this case a single difference equation suffices to calculate all master integrals and thus all integrals in that sector. If it is not possible to express the master integrals in this way, the sector will have more than one difference equation (see sect. 5) which will solve the additional master integrals analogously.

There are several reasons to pick Laporta's above described approach as the method to solve the fully massive 5-loop tadpoles. The most compelling one is the fact that it has already been used to solve the same class of integrals at 4 loops for both dimension $d = 4 - 2\epsilon$ [4] and $d = 3 - 2\epsilon$ [5], as well as the 4-loop tadpoles with some massless lines [115, 33]. Obtaining results in different dimensions is rather simple in this approach, since both difference equations and recurrence relations are valid for arbitrary d and only for the numerical evaluation does one have to pick an integer around which to expand in ϵ.

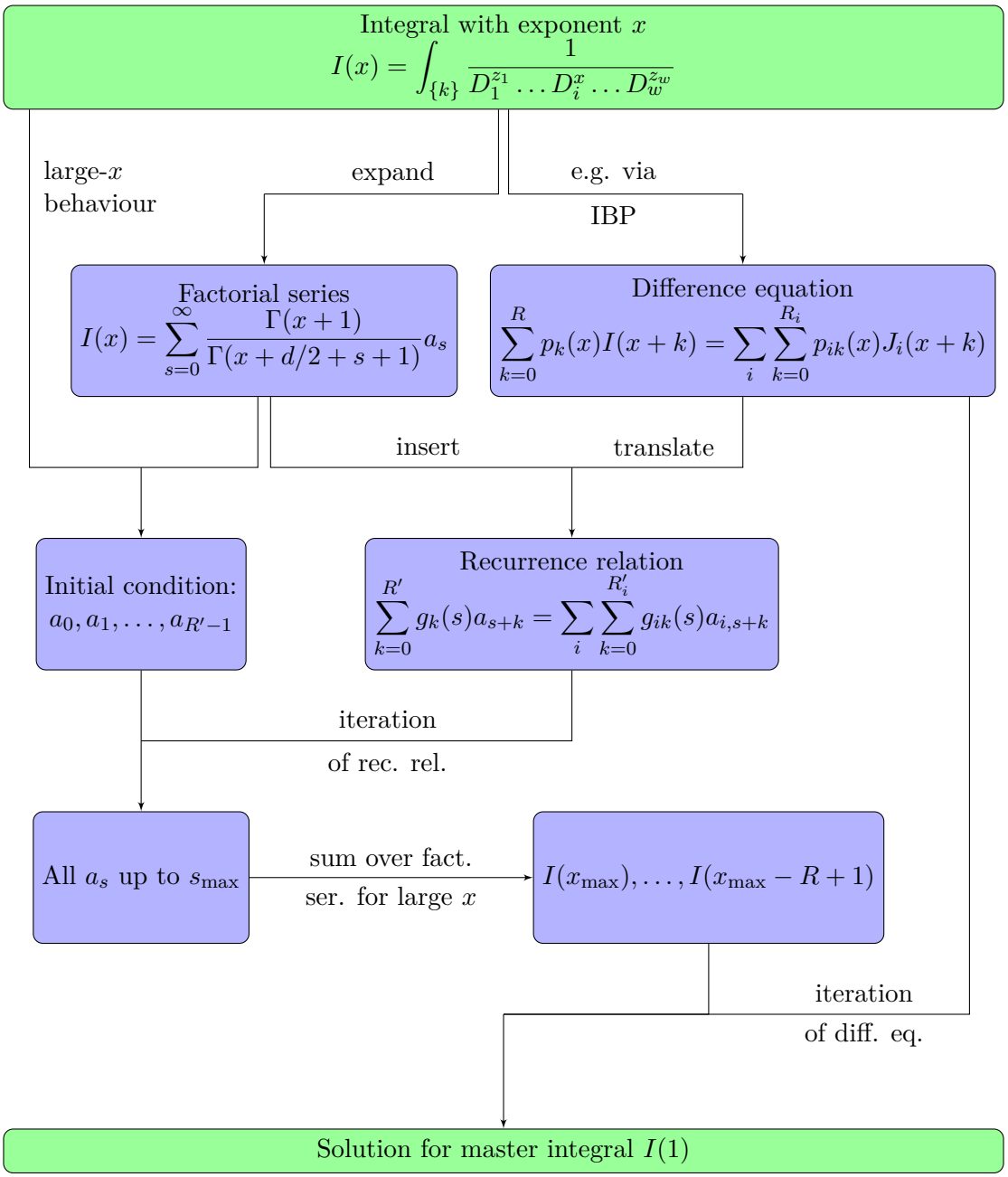


Figure 4.1: Flow chart of Laporta's method [1] of difference equations and factorial series.

Even though in most cases only numerical results can be obtained, it is possible to achieve extremely high precisions (see e.g. sect. 9.2 or [4]), which can be used to recover the analytical result via an algorithm like PSLQ [112, 113], if one knows the numbers and special functions which may appear. In the case of the 5-loop tadpoles the numerical nature of the approach can then even be an advantage, since it is known from lower loop results [4, 82] that elliptic integrals appear in the ϵ-expansions. This makes the 5-loop tadpole integrals inaccessible to some powerful analytic methods like symbolic integration based on linear reducibility [81, 79] or first order differential equations using a canonical basis [97], which are restricted to multiple polylogarithms [76].

While it would certainly be nice to directly obtain analytical solutions for the recurrence relations or even the difference equation, to the best of my knowledge no strategy is known that can accomplish this in the general case. Some algorithms exist that can express the solutions in terms of indefinite nested sums and products (see e.g. [116, 117, 118]). They do however rely on the fact that the recurrence relation factorises into linear parts[7], which is in general not the case for the massive tadpoles.

Another nice feature of Laporta's method is the fact that it deals well with divergent integrals. Poles in ϵ may arise during the iterations of either difference equations or recurrence relations and are naturally incorporated into the ϵ-expansions of integrals and factorial series coefficients. This quality might however lose some relevance in the future, since a recent proposal [54] shows how to choose a basis of (quasi-)finite master integrals for a given space-time dimension d. As a trade-off this basis generally also contains higher dimensional integrals, but all divergences are made explicit in the form of ϵ-poles of the coefficients in front of the chosen master integrals.

Every step in Laporta's method can be readily automated and I will describe my own implementation in section 8. Once this is done, very little manual input is needed to obtain solutions for the master integrals. This reduces the probability of human error and is very convenient, since there are more than 100 master integrals for the fully massive 5-loop tadpoles [6]. Furthermore the quality and accuracy of the results can be estimated from a cross-check that is built into the method (see sect. 5).

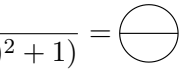
Despite its qualities Laporta's method as described above does not work for the fully massive 5-loop tadpole integrals. Several of its steps scale very poorly with the complexity of the integrals and cannot be performed at 5 loops within a reasonable amount of time. In addition, many of the 5-loop tadpoles exhibit divergences during the evaluation of the factorial series. In sections 5 through 7 I will describe these problems in detail and demonstrate how the method can be changed to solve or at least attenuate them.

[7] A simple example for such a factorisation is given in section 7.1

4.1 Example: 2-loop sunset

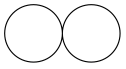
To illustrate Laporta's method, I will show some of the steps in its application to solve the 2-loop sunset master integral

$$I(1,1,1) = \int_{k_1,k_2} \frac{1}{(k_1^2+1)(k_2^2+1)((k_1-k_2)^2+1)} = \tag{4.7}$$



with the assumption that the subsector master integral

$$I(1,1,0) = \int_{k_1,k_2} \frac{1}{(k_1^2+1)(k_2^2+1)} = \tag{4.8}$$



is already solved. The integrals are first generalised to $I_1(x) \equiv I(x,1,1)$ and $I_2(x) \equiv I(x,1,0)$ by replacing the exponent on the first propagator with the variable x. By combining several IBP-relations for these integrals one finds

$$\begin{aligned} 6x(x+1)I_1(x+2) + 2x(-2x+d-3)I_1(x+1) + 2x(-x+d-2)I_1(x) \\ +((-2d+4)x + d^2 - 2d)I_2(x) = 0 \end{aligned} \tag{4.9}$$

as the difference equation for $I_1(x)$. Inserting the factorial series

$$I_i(x) = \sum_{s=0}^{\infty} \frac{a_{i,s}\Gamma(x+1)}{\Gamma(x+d/2+s+1)} \tag{4.10}$$

for I_1 and I_2 and going through the translation process described in section 6.1 yields the recurrence relation[8]

$$\begin{aligned} & 128(-2+s)(-1+s)s(1+2s)a_{1,s} \\ & -16(-2+s)(-1+s)s(-2+d+2s)(-10+5d+22s)a_{1,s-1} \\ & +16(-2+s)(-1+s)s(-4+d+2s)(-2+d+2s)(-13+4d+10s)a_{1,s-2} \\ & -12(-2+s)(-1+s)s(-6+d+2s)(-4+d+2s)^2(-2+d+2s)a_{1,s-3} \\ & +(-2+d)^2d(-6+d+2s)(-4+d+2s)^2(-2+d+2s)a_{2,s-3} = 0\ . \end{aligned} \tag{4.11}$$

Since the pivot coefficient in the first line vanishes for $s = 0, 1, 2$, the first terms $a_{1,0}$ to $a_{1,2}$ of the factorial series have to be obtained as initial conditions[9]. All three can be expressed

[8] Here I have already used the recurrence relation for $a_{2,s}$ to reduce the number of terms.

[9] In this case the recurrence relation could actually be written in an alternative form which is then also able to determine $a_{1,0}$ to $a_{1,2}$ in terms of the $a_{2,s}$, but in general this is not always possible.

(see sect. 7.3) in terms of the known 1-loop master integral $J = \int_{k_1} (k_1^2 + 1)^{-1}$ as

$$\begin{aligned} a_{1,0} &= -\frac{d-2}{2} J \\ a_{1,1} &= -\frac{(-2+d)d(8+7d)}{48} J \\ a_{1,2} &= -\frac{(-2+d)d(2+d)(104+d(170+41d))}{1920} J \,. \end{aligned} \tag{4.12}$$

After expanding these values around $d = 4$ in $\epsilon = 2 - d/2$ up to a chosen order, the recurrence relation can be used to iteratively obtain numerical ϵ-expansions for the remaining series coefficients $a_{1,4}, a_{1,5}, \ldots, a_{1,s_{\max}}$. The $a_{2,s}$ needed to do so are already known from the solution for $I_2(x)$. Performing the sum over the factorial series for $x = x_{\max} = 100$ up to $s_{\max} = 500$ yields

$$\begin{aligned} \frac{I_1(100)}{J^2} =& 1.0307153164\ldots \cdot 10^{-4}\epsilon + 3.2505119301\ldots \cdot 10^{-4}\epsilon^2 \\ &+ 3.2541271033\ldots \cdot 10^{-4}\epsilon^3 - 7.8779658176\ldots \cdot 10^{-4}\epsilon^4 + \ldots \,, \end{aligned} \tag{4.13}$$

where the normalisation with the 1-loop tadpole $J = \Gamma(1 - d/2)$ from eq. (3.3) is introduced to prevent a dependence on the convention for the integration measure. A similar expansion can be obtained for $I_1(99)$ and since the $I_2(x)$ are known, the difference equation (4.9) can be used to iteratively compute $I_1(98), I_1(97), \ldots, I_1(1)$. The final result is

$$\frac{I_1(1)}{J^2} = -1.499999999 - 1.499999999\epsilon + 0.515860858\epsilon^2 - 8.540503339\epsilon^3 + \ldots \,. \tag{4.14}$$

This agrees with the analytical solution, which is known for arbitrary d in terms of hypergeometric functions as [119, 115]

$$\frac{I_1(1)}{J^2} = -\frac{3(d-2)}{4(d-3)} \left[{}_2F_1\left(\frac{4-d}{2}, 1; \frac{5-d}{2}; \frac{3}{4}\right) - 3^{\frac{d-5}{2}} \frac{2\pi\Gamma(5-d)}{\Gamma\left(\frac{4-d}{2}\right)\Gamma\left(\frac{6-d}{2}\right)} \right] \,. \tag{4.15}$$

Both $I_1(100)$ and $I_1(1)$ actually contain divergences, which are masked here by the normalisation due to the $1/\epsilon$ pole in J. The additional pole of $I_1(1)$ compared to $I_1(100)$ is picked up without any special treatment at $x = 2$, where the coefficient of $I_1(x)$ in the difference equation (4.9) starts at order ϵ^1.

5 Difference equations

Finding difference equations of the form (4.2) is a task very similar to the reduction to master integrals described in section 2, with the key difference being the introduction of the symbolic exponent x for one of the propagators. In this section I first adapt and extend upon some of the concepts introduced for the reduction to account for this change. I then review Laporta's algorithm for obtaining difference equations (sect. 5.1) before presenting my own algorithm (sect. 5.2), which produces an alternative form of the difference equations. A comparison of the performance of the two algorithms for the 5-loop tadpoles is given in section 5.3.

In eq. (2.6) I defined the parameters r, s, t and the sector ID, which are needed for the ordering relation of integrals given in section 2.4. To adjust these definitions to integrals of the form

$$I(z_1, \ldots, x+k, \ldots, z_w) = \int_{\{k\}} \frac{1}{D_1^{z_1} \ldots D_v^{x+k} \ldots D_w^{z_w}} \quad z_i, k \in \mathbb{Z} \,, \tag{5.1}$$

I will assume that $x + k$ is positive for any offset k, so that $\Theta(x + k) = 1$ and the definition of the sector and number of propagators t is consistent with those of the master integral one is trying to compute, where $x + k$ is replaced by (typically) 1. The number of propagator powers s in the numerator is then also unchanged, but the number of dots r will be redefined to ignore the exponent containing x completely to ensure it remains a non-negative integer. Furthermore, the offset k will be ignored in the ordering relation, since integrals only differing in k are related by a simple shift in x. This means that after assigning indices to the integrals, each index j now represents the set of integrals $\{I_j(x + k)|k \in \mathbb{Z}\}$ instead of a single integral.

The variable x can be used as the exponent for any propagator in the denominator, and different positions v in general produce different difference equations. Due to symmetries of the associated graph some choices for the position v might be related by momentum shifts and are therefore equivalent. In this case I will by convention only consider integrals with the smallest v out of every set of equivalent positions. This choice is incorporated into the ordering relation by adding it after the comparison of sector IDs. For convenience, I repeat here the ordering relation for (now sets of) integrals introduced in section 2.4 with this addition, using the parameters defined in eq. (2.6) with the above described changes. For two different (sets of) integrals of the form given in eq. (5.1) the higher difficulty will be assigned to, in descending order of priority,

1. the one with greater t (number of denominators),
2. the one with lower sector ID,
3. the one with greater position v for the symbolic exponent x,
4. the one with greater r (number of dots, ignoring exponent at position v),
5. the one with greater s (numerator powers),
6. the one with greater $\max(z_i)$,
7. the one with lower $\min(z_i)$,
8. the one with greater z_1, z_2, ...

After assigning indices j based on the ordering relation, a general equation for the integrals has the form

$$\sum_j \sum_{k=k_b^{(j)}}^{k_t^{(j)}} p_{jk}(x) I_j(x+k) = 0 \,, \tag{5.2}$$

where the p_{jk} are rational functions of x and d. In the following I refer to the difference $k_t^{(j)} - k_b^{(j)}$ of the top and bottom offsets of I_j as the *span* of that integral in the equation. The span of the whole equation is then defined as $k_t - k_b$, where $k_t = \max(k_t^{(j)})$ and $k_b = \min(k_b^{(j)})$ need not necessarily appear on the same integral. This is in contrast to the term *order* of the equation, which I would like to reserve for the span of the pivot integral, in accordance with eq. (4.2). In Laporta's algorithm the pivot integral is always the most difficult integral in the equation, but in my alternative algorithm the choice of pivot integral will be more involved.

A system of equations of the form (5.2) can be generated by the same methods of section 2.3 which are used for the normal reduction. Reducing this system to difference equations also follows the same basic idea of starting from simple equations and always eliminating the most difficult integrals from as many equations as possible. More details on this reduction will be given in sections 5.1 and 5.2. When generating equations from a seed integral in sector S with x at position v, all integrals in the resulting equations also start out with x at v. The only way the position of x can be changed is by sector shifts or symmetry shifts. Since for all seed integrals S is always the representative sector of its associated graph and v is always the representative of its equivalent positions[10], such shifts can only occur for integrals in subsectors of S. As a consequence, the difference equations for the same sector S but different (non-equivalent) positions for x can be obtained independently.

In the following I call the combination of a sector S and a position v a *zone* and denote it as $sID(S)\#v$, e.g. 841#1 for the 4-loop sunset integral $I(x,1,0,1,0,0,1,0,0,1)$. A *subzone*

[10] Choosing seed integrals from other sectors or with different positions v is possible, but would only give redundant information.

is then defined as any zone that can be reached by removing denominator propagators and applying sector or symmetry shifts to the result. The set of subzones of a zone Z is a subset of, but not necessarily the full set of all zones of subsectors of the sector of Z, e.g. 4-loop zone 1010#2 has subzones 993#2 and 993#4, but not 993#1. To solve a single master integral, it is in principle sufficient to only find a difference equation in one zone of its sector, but since most zones will be needed as subzones in difference equations for integrals with more propagators, one usually has to reduce all zones of a given sector.

Obtaining the difference equations in all (inequivalent) zones of a given sector has the additional benefit that it provides a cross-check for the final results. Inequivalent zones in general have different difference equations and recurrence relations, which do not even have to share the same orders. The final result $I(1)$ however should not depend on the position v the exponent x occupied. Since very few sectors are fully symmetric and therefore have only one set of equivalent zones, one almost always obtains multiple results for $I(1)$ from different zones. Even though the equations in different zones might not be fully independent, since they can depend on the same subzones, this still provides a non-trivial check for the results and can be used to estimate their numerical accuracy.

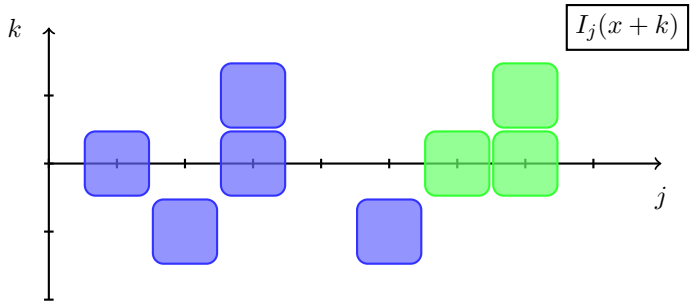


Figure 5.1: Pictorial representation of a typical IBP-relation between integrals $I_j(x)$. Each box represents one integral times a coefficient, the sum over which is zero. Blue boxes are integrals in the most difficult zone (the one of the seed integral), green boxes are integrals in subzones.

5.1 Laporta's algorithm for difference equations

The algorithm Laporta described in [1] to obtain difference equations is essentially the algorithm used for the reduction to master integrals with the above described changes to the ordering relation to account for the introduction of the symbolic exponent x. As in the normal reduction, the system of equations is hierarchical and can be reduced zone by zone, starting with the lowest number of propagators and then working upwards, where equations in a zone Z may also depend on integrals in subzones of Z. Equations are generated mostly via IBP, but all other methods described in section 2.3 are also applicable. The typical

structure of an IBP-relation between integrals $I_j(x)$ is depicted in figure 5.1. To reduce a given zone Z, Laporta suggests to generate all equations for integrals $I_j(x+k) \in Z$ with $r \leq r_{\max}$, $s \leq s_{\max}$ and offsets $k \leq k_{\max}$. With the order of equations defined as in section 2.4, the reduction algorithm is then given by:

1. Consider the next equation. If there are none left, stop.
2. Insert solutions for all integrals which have already been found.
3. If the equation is trivial ($0 = 0$), go to step 1.
4. Out of the most difficult set of integrals $\{I_j(x+k)|k \in \mathbb{Z}\}$ remaining (the one with the lowest index j), pick as pivot integral the one with
   - the lowest offset k in the equation, if I_j appears with negative offsets, or
   - the highest offset k in the equation otherwise.

   Solve the equation for the pivot integral.
5. Use the newly found solution to eliminate the pivot integral from all previously found solutions. Go back to step 1.

If the system of equations contains sufficient information for the reduction, this algorithm produces equations of the form depicted in figure 5.2. In most cases there is one difference equation of the form in eq. (4.2), containing only one integral in zone Z with different offsets (and possibly several integrals in subzones of Z). In analogy with the normal reduction, this integral will be called a master integral. For more difficult integrals in the ordering relation the algorithm generates equations of order 0, which fully determine these integrals in terms of the master integral and subzone master integrals. These equations are not actually needed for the solution via factorial series, but have to be kept to eliminate their pivot integrals in reductions of superzones of Z. In some zones it also happens that there is more than one equation with order $R > 0$, none of which can be reduced further. In this case I refer to all their pivot integrals $I_{j_1}(x), I_{j_2}(x), \dots$ as master integrals and to the set $M_Z \equiv \{I_{j_1}(x), I_{j_2}(x), \dots\}$ as the *master basis* of zone Z. I further define the *order* of a zone to be the sum of all the orders of difference equations for integrals in that zone. A master basis containing multiple integrals does not spoil the hierarchical nature of the approach, since their respective difference equations only depend on their pivot integrals and simpler integrals, so they can still be solved one by one.

There is one obvious optimisation to the algorithm described above, which I assume Laporta also makes use of, although it is never stated explicitly in reference [1]. Since any equation of the form (5.2) is valid for all x, one can perform shifts in x, which is a freedom that Laporta's description of his algorithm does not mention. This means that it is not

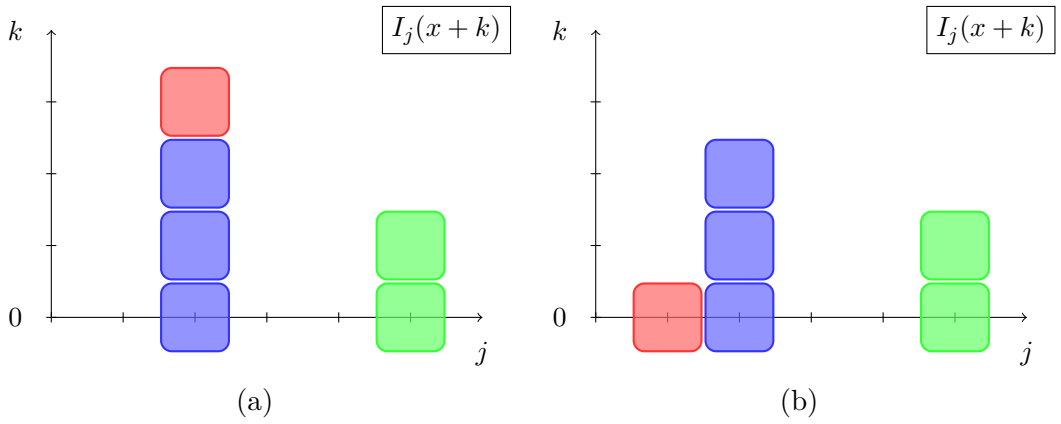


Figure 5.2: Pictorial representation of fully reduced equations after Laporta's algorithm in the style of figure 5.1. Red boxes represent the pivot integral and coefficient of an equation. (a) is a difference equation of the form in eq. (4.2) with order 3, while (b) has order 0 and the pivot integral is thus fully determined by simpler integrals.

necessary during the generation of equations to use seed integrals $I(x+k)$ that have offset $k \neq 0$, since the information gained from those equations can also be recovered if the reduction algorithm makes use of such shifts. This is a significant improvement, since it decreases the number of equations that have to be considered and avoids repeating the same reduction steps for different offsets. A restatement of the above algorithm including x-shifts and only using seed integrals with offset 0 is provided in algorithm 1. For the remainder of this thesis, when I mention Laporta's algorithm for difference equations, this is the version I am referring to, since it is the one I use in my implementation (see sect. 8). It produces the same results (fig. 5.2), but is more efficient than the version without x-shifts.

For tadpole integrals with a single mass and up to 4 loops, Laporta's algorithm works well to obtain all difference equations (see e.g. [4, 5, 115]). Attempts have been made to find the difference equations for the fully massive 5-loop tadpoles [6, 7], but progress has been limited so far, since the calculations at 5 loops are significantly more difficult than at 4 loops. This is partly due to the increased numbers of diagrams, propagators, zones and integrals per zone that need to be considered. The main problem however is a sharp increase of complexity of the rational coefficients in the equations. The size of these coefficients scales very badly with the span of the equations, as I will show in section 5.3. This increase in complexity was already noted by Laporta in [1] in the context of decoupling equations. At 4 loops, the maximum order and therefore also maximum span of the difference equations is 5, while at the 5-loop level the orders reach up to 20 [7]. As a result, the rational algebra during the execution of algorithm 1 slows down to the point where obtaining the difference equations in this way becomes altogether infeasible.

Algorithm 1 Laporta's algorithm [1] using x-shifts

Require:

- An ordering relation for integrals $I(x)$.
- A set of equations of the form (5.2), generated from seed integrals in zone Z.
- A list P of pivot equations, where $P[j]$ is the equation with I_j as a pivot integral (if one exists). P is initially filled with equations from previous runs, e.g. for subzones.

1. If no new equations remain, stop. Otherwise pick the simplest of the remaining equations and call it Q. Out of two equations, the more difficult one is defined to be, in decreasing order of priority,
   - the one with the most difficult integral,
   - the one with the greatest *leading span* (the span of its most difficult integral),
   - the one with the most terms,
   - either one.
2. For $j = 1, 2, \ldots$ do:
   (a) If Q does not depend on I_j or $P[j]$ does not exist yet, continue with next j.
   (b) Add x-shifted versions of $P[j]$ with appropriate factors to Q to eliminate all integrals $I_j(x + k)$ from Q where $k < 0$.
   (c) Let s_P be the leading span of $P[j]$. Add x-shifted versions of $P[j]$ with appropriate factors to Q to eliminate all integrals $I_j(x + k)$ from Q where $k \geq s_P$.
3. If Q is trivial, go to step 1.
4. Let I_m be the most difficult integral in Q. Divide Q by the coefficient of $I_m(x+k_t^{(m)})$, where $k_t^{(m)}$ is the top offset of I_m. Shift Q such that the bottom offset $k_b^{(m)}$ of I_m is zero. If this introduced any offsets < 0 for other integrals, perform step 2b again for $j = m + 1, m + 2, \ldots$.
5. Let s_Q be the leading span of Q. Use Q to eliminate all integrals $I_m(x + k)$ in $P[1], P[2], \ldots, P[m - 1]$ where $k < 0$ or $k \geq s_Q$.
6. Set $P[m] = Q$. If this replaces a previous pivot equation $P[m]$, use that equation as the next Q and go to step 2, else go to step 1.

5.2 Minimal order reduction algorithm

In this section I present an alternative algorithm for the task of reducing general equations of the form (5.2) to difference equations. Since the main reason Laporta's algorithm falls short at the 5-loop level is the complexity of the coefficients in equations with high orders, my algorithm is built to minimise the spans /orders of equations. This means that instead of decoupling equations in a given zone Z as much as possible to find a difference equation of order R for a single integral, it produces a set of R coupled equations with order 1 for R different integrals of the zone. Consequently I will refer to the algorithm as the Minimal Order Reduction Algorithm (MORA), which is listed below as algorithm 2. The difference equations are always generated in two versions, *top* and *bottom* equations, with the respective forms given by

$$\begin{aligned} I_i(x+1) + \sum_{\substack{j \\ I_j \in M_Z}} p_{ij}^{(t)}(x) I_j(x) + \sum_{\substack{j \\ I_j \in sub(M_Z)}} p_{ij}^{(t)} I_j(x) = 0\,, \quad I_i \in M_Z, \\ I_i(x-1) + \sum_{\substack{j \\ I_j \in M_Z}} p_{ij}^{(b)}(x) I_j(x) + \sum_{\substack{j \\ I_j \in sub(M_Z)}} p_{ij}^{(b)} I_j(x) = 0\,, \quad I_i \in M_Z, \end{aligned} \tag{5.3}$$

where M_Z is the master basis of Z (with $|M_Z| = R$) and $sub(M_Z)$ is the union of master bases of subzones of Z. The structure of these equations is also depicted in figure 5.3. The sets of top and bottom equations contain equivalent information, but during the reduction it is convenient to have both forms to be able to easily solve for integrals with both positive and negative offsets of x.

Coupled difference equations of the form (5.3) are produced by the MORA using the same kind of input equations as Laporta's algorithm, but the former is more elaborate than the latter. This difference in complexity is mostly due to the fact that in the MORA the choice of pivot integral for a given equation is more involved than simply picking the most difficult integral according to the ordering relation. The basic ideas of the two algorithms can be briefly summarised in terms of the pictorial representations of equations introduced in figures 5.1, 5.2 and 5.3 as:

- Algorithm 1 (Laporta's algorithm): In each equation, move as many boxes as possible as far to the right as possible, even if it increases the number of rows. Only decrease the number of rows in a column when it does not influence columns to the left.
  $\Rightarrow$ For equations of integrals $I_j(x+k)$, reduce first in j, then in k.
- Algorithm 2 (MORA): In each equation, try to move as many boxes as possible to as few adjacent rows as possible, even if it introduces new columns to the left. Only move boxes from left to right when it does not increase the number of rows.
  $\Rightarrow$ For equations of integrals $I_j(x+k)$, reduce first in k, then in j.

Starting from the same input, the coupled equations approach of the MORA often requires slightly more steps to arrive at a fully reduced system than Laporta's algorithm, but the complexity of the coefficients scales significantly better, as I will show in section 5.3.

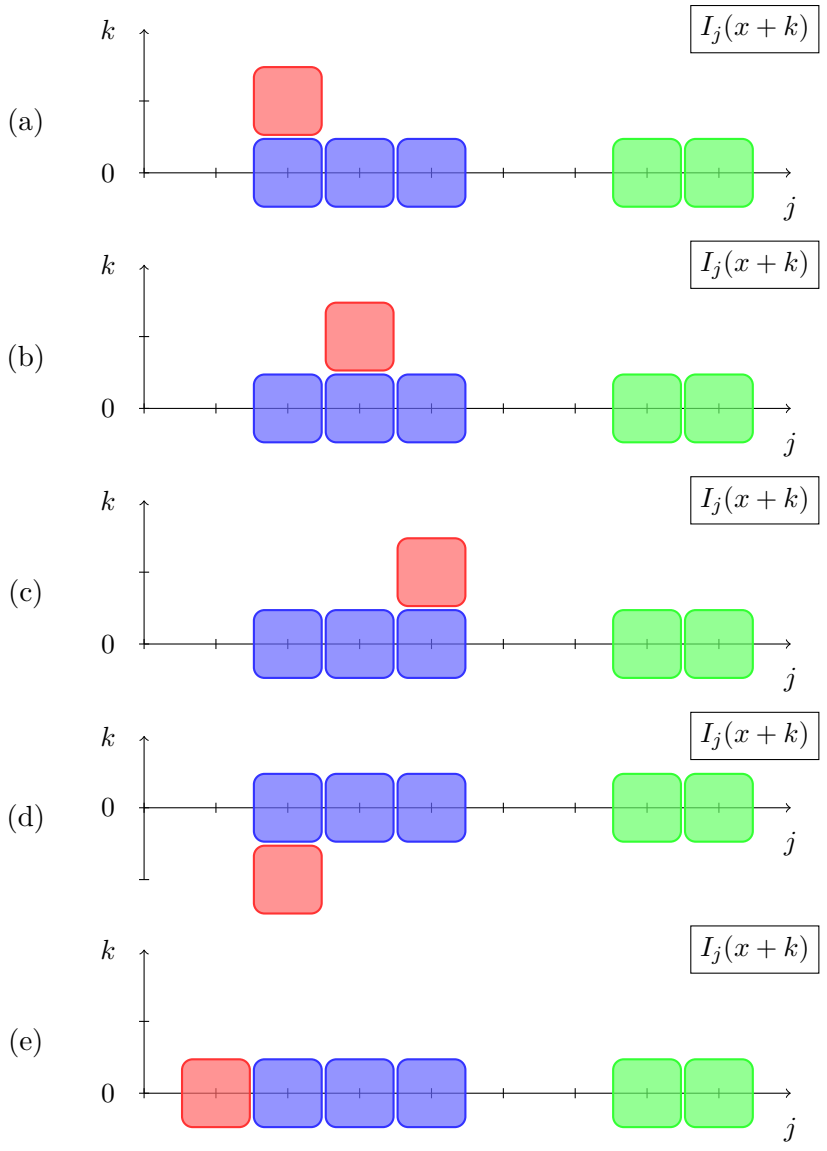


Figure 5.3: Pictorial representation of fully reduced equations in the coupled form in the style of figure 5.1. The decoupled version of these equations is depicted in figure 5.2. Pivot integrals are indicated by a red box. (a)-(c) show 3 coupled top difference equations of order 1, while (d) is one of the corresponding bottom equations. Equation (e) has order (and span) 0 and the pivot integral is fully determined by simpler integrals.

Switching to coupled difference equations also opens up an additional degree of freedom which can be used to decrease the complexity of the rational coefficients further. By modifying the ordering relation for integrals inside a zone Z, the master basis M_Z can be changed, which leads to a different, but equivalent set of difference equations. Since the total number of integrals considered in a zone is typically much larger than the size of the master basis, there are many possible choices for M_Z, which can differ significantly in complexity of coefficients. Due to the potentially vast number of possibilities, in practice only a subset of bases are tested by reducing a system where the dimension d is replaced by a number for simplicity and the quality of a basis is gauged by the degrees in x of the coefficients (for details see section 8.1.5). While the master basis could of course also be changed for decoupled equations, there is little room for improvement in exploiting this, since here the basis usually consists of only a single integral.

In principle it would also be possible to consider arbitrary linear combinations of integrals I_j as elements of the master basis. One might even find something akin to the canonical basis proposed by Henn [97] for differential equations, where the coefficients of the equations have a very simple structure in the dimensional regularisation parameter ϵ. This has not been pursued in the work of this thesis, because choosing good candidates for the linear combinations in an automated way is a highly complex problem and it was not needed for the integrals considered here, but it is an interesting avenue for future improvement of the approach.

An unfortunate consequence of working with coupled equations is the necessity to solve several integrals simultaneously. While different zones can still be considered sequentially, all integrals in a master basis M_Z have to be expanded as factorial series at the same time. This is not a fundamental problem, but it means that some additional work is required after translating the difference equations to recurrence relations to make the latter suitable for the iterative evaluation of the factorial series coefficients (see sect. 6). Furthermore, the fact that the number of integrals that need to be solved in this approach increases by a factor $|M_Z|$ slows down the numerical evaluation of the integrals. In order to avoid these negative aspects of coupled equations, one might be tempted to first reduce the full system with the MORA to coupled equations and then only decouple the difference equations for the master basis. Since the master basis is usually only a small part of all integrals that need to be reduced, this would avoid the large coefficients in the decoupled equations for most of the reduction. For simpler problems this strategy might prove effective, but for the 5-loop tadpoles I have found that even when trying to only decouple the equations for the master basis the rational algebra of the coefficients turns out to be too costly in terms of computation time. An additional reason for choosing coupled equations over decoupled ones is the ability to also reduce recurrence relations, which will be described in section 6.

Algorithm 2 Minimal Order Reduction Algorithm (MORA)

Require:

- An ordering relation for integrals $I(x)$.
- A set U of previously unused equations of the form (5.2), generated from seed integrals in zone Z.
- Tables for keeping track of top and bottom pivot equations (P_t, P_b) and their respective spans (S_t, S_b) for each index j assigned by the ordering relation, i.e. $P_t[j]$ would be the top pivot equation for integral I_j and $S_t[j]$ the span of this equation.
- Sets of currently "benched" equations, one for top (B_t), one for bottom (B_b), initially empty.
- Pivot equations from previous runs, e.g. for subzones.

1. If U is empty, stop. Otherwise pick the simplest equation $Q \in U$ as the next equation to consider and remove Q from U. Out of two equations, the more difficult one is defined to be, in decreasing order of priority,
   (a) the one with the most difficult integral,
   (b) the one with the greatest *leading span* (the span of its most difficult integral),
   (c) the one with the most terms,
   (d) either one.
2. Using $P_t[j]$, remove all integrals $I_j(x+k)$ from Q for indices j with span $S_t[j] = 0$.
3. Determine the top offset k_t and the span R of Q.
4. For indices $j = 1, 2, \ldots,$ do:
   (a) If Q has a non-trivial coefficient of $I_j(x+k_t)$, $P_t[j]$ exists and $S_t[j] \leq R$, use an appropriately x-shifted version of $P_t[j]$ to eliminate $I_j(x+k_t)$ from Q.
5. If Q is trivial ($0 = 0$), go to step 1. If Q is non-trivial, but the span of Q has decreased in the last step, go to step 3.
6. If $R > 0$:
   (a) Make a copy Q_b of Q which will serve as a bottom equation, while Q will be a top equation.
   (b) `reduce_eq`(Q_b, bottom) (see subr. 1)
   (c) `make_pivot`(Q_b, bottom) (see subr. 2)
7. `reduce_eq`(Q, top) (see subr. 1)
8. `make_pivot`(Q, top) (see subr. 2)
9. While B_t or B_b contain equations:
   (a) Pick and remove an equation Q' from either B_t or B_b.
   (b) If Q' came from B_t:
      i. `reduce_eq`(Q', top)
      ii. If Q' is non-trivial: `make_pivot`(Q', top)

      Else:
      i. `reduce_eq`(Q', bottom)
      ii. If Q' is non-trivial: `make_pivot`(Q', bottom)
10. `reduce_system` (see subr. 3)
11. Go to step 1.

Subroutine 1 `reduce_eq`(*equation* Q, *mode* m)

Require: Same as in algorithm 2 plus one input equation Q and a mode m, which can be top (t) or bottom (b). In the following, top mode is assumed, but bottom mode is analogous with the obvious exchanges of high/low, top/bottom... .

1. Determine the top offset k_t and the span R of Q.
2. Determine the pivot index c as the smallest index with a non-zero coefficient of $I_c(x + k_t)$.
3. If $P_t[c]$ exists, $P_t[c] \neq Q$ and $S_t[c] \leq R$:
   (a) Use an appropriately x-shifted version of $P_t[c]$ to eliminate $I_c(x + k_t)$ from Q.
   (b) Go to step 1.

   Otherwise $I_c(x + k_t)$ becomes the pivot integral of Q.
4. Determine the bottom offset k_b and the span R of Q.
5. Check if it is possible to remove (using the bottom pivot equations P_b) <u>all</u> integrals $I_j(x + k_b)$ with bottom offset from Q without
   - introducing integrals $I_j(x + k_t)$ with offset $k > k_t$,
   - introducing integrals $I_j(x + k_t)$ with top offset and $j < c$,
   - removing the pivot integral $I_c(x + k_t)$ from Q.

   If this is possible, perform the necessary steps to do so and go back to step 4.
6. For index $j = c + 1, c + 2, \ldots,$ do:
   (a) If Q has a non-zero coefficient for $I_j(x + k_t)$, $P_t[j]$ exists and $S_t[j] \leq R$, use $P_t[j]$ to eliminate $I_j(x + k_t)$ from Q.
7. For $k = k_b, k_b + 1, \ldots, k_t - 1$ (outer loop) and $j = 1, 2, \ldots$ (inner loop), do:
   (a) If k is the lowest offset of I_j appearing in Q, $P_b[j]$ exists and $S_b[j] < k_t - k$, use $P_b[t]$ to eliminate $I_j(x + k)$ from Q.

Subroutine 2 `make_pivot`(*equation* Q, *mode* m)

Require: Same as in algorithm 2 plus one input equation Q and a mode m, which can be top (t) or bottom (b). Let $\bar{m}$ then denote the opposite mode.

1. Determine the highest (if $m = t$) or lowest ($m = b$) offset k_m and span R of Q.
2. Determine the pivot index c as the smallest index with a non-zero coefficient of $I_c(x + k_m)$.
3. Divide Q by the coefficient of $I_c(x + k_m)$ and shift Q in x such that the average of the lowest and highest offset in Q is either 0 or $\pm 1/2$ (+ for $m = t$, − for $m = b$).
4. If a pivot equation $P_m[c]$ exists and $P_m[c] \neq Q$, add $P_m[c]$ to the set B_m.
5. Set $P_m[c] = Q$ and $S_m[c] = R$.
6. If $R = 0$, also perform the previous two steps for $P_{\bar{m}}[c]$, $S_{\bar{m}}[c]$ and $B_{\bar{m}}$.

Subroutine 3 `reduce_system`

Require: Same as in algorithm 2.

1. For index $j = 1, 2, \ldots,$ do:
   (a) If $P_t[j]$ exists, `reduce_eq`($P_t[j]$, top) (see subr. 1).
   (b) If $P_t[j]$ was changed in (a), `make_pivot`($P_t[j]$, top) (see subr. 2).
2. Perform the equivalent steps for the bottom pivot equations P_b.
3. If any pivot equations were changed in this iteration, go back to step 1.

5.3 Comparison of reduction algorithms

The key differences of the MORA compared to Laporta's algorithm are:

- The algorithm itself is more involved and often needs slightly more steps to reduce a system of equations.
- The master basis M_Z of a zone is larger and thus more integrals need to be expanded as factorial series. The integrals in M_Z now need to be solved simultaneously.
- Each integral in M_Z now has two associated difference equations (top and bottom), which hold redundant information.
- The rational coefficients in the coupled equations produced by the MORA are typically simpler than in the decoupled equations, especially in zones with high orders.
- By changing the master basis M_Z the rational coefficients can be further simplified.

In the following I will quantify some of these points with examples and statistics of the fully massive 5-loop tadpoles integrals. Table 5.1 shows some statistics for the reduction to difference equations of zone 30214#1 using the different algorithms. Having 7 propagators, this zone is still fairly easy to reduce, since it only has 3 subzones. The order of the zone is 7, concentrated in a single difference equation in the output of Laporta's algorithm and spread over 7 equations of order 1 in the output of the MORA. Due to the top and bottom versions of equations with span > 0 the latter algorithm thus produces 7 additional equations. The table furthermore shows that the complexity of the coefficients, measured in size as a string and degrees in x and d, is significantly lower for coupled equations than for decoupled equations, and even more so if an optimised master basis is used. Size as a string is used as a measure because all rational algebra in my implementation of the algorithms is performed by the external program Fermat [120] and internally all coefficients are only stored as strings. Besides being important for memory consumption this size also takes into account the integer contents of the coefficients in addition to the degrees in its variables, which is an important factor for the speed of the rational algebra.

Another effect that can be seen from table 5.1 is that "accidental" zeroes are more likely to happen for coupled equations, which reduces the average number of coefficients per equation compared to the maximum 8 in the zone ($8 = R + 1$) plus another 7 in subzones ($7 =$ sum of orders of those zones) which can appear in a fully reduced equation. Even though the coupled equations tend to have fewer coefficients, the MORA needs more steps to reduce zone 30214#1 than Laporta's algorithm. This is due to additional work to produce both top and bottom equations and situations like the one depicted in figure 5.4, where reducing a single integral with offset 2 takes more steps with coupled than with decoupled equations. Despite often needing more steps for the reduction, the MORA takes

algorithm	including subzones	#eqs.	⟨#coeffs.⟩	⟨size⟩	⟨deg$_x$⟩	⟨deg$_d$⟩	steps	runtime
LA	no	214	7.94	354	5.02	5.50	15628	36s
MORA	no	221	7.51	82	1.16	3.60	21825	18s
MORA+B	no	221	7.10	52	1.54	2.13	20432	10s
LA	yes	214	14.90	702	7.75	7.39	34085	84s
MORA	yes	221	13.99	229	3.13	5.77	39605	30s
MORA+B	yes	221	12.81	65	1.89	2.50	36548	17s

Table 5.1: Statistics of the reduction to difference equations for zone 30214#1 ($t = 7$, order 7) using different algorithms: Laporta's algorithm (LA), MORA and MORA with an optimised master basis (MORA+B). Except for the last two columns, all numbers are for the final reduced system including all integrals up to $r \leq r_{\max} = 2$ and $s \leq s_{\max} = 2$, where denominators have been cleared to have polynomial coefficients. Shown here are the number of equations in the zone, the average number of coefficients per equation, the average size of coefficients written as a string and the average degrees of the coefficients in x and d. The number of steps and runtime are for the reduction of the larger system with $r_{\max} = 2$, $s_{\max} = 3$.

up less computation time than Laporta's algorithm. For this rather simple reduction it is difficult to judge whether this is due mainly to the less complex rational algebra or the fact that I have spent more time optimising the implementation of the former algorithm than the one of the latter, even though many of the optimisations are used in both versions.

Table 5.2 lists the same kind of statistics as table 5.1 for several other zones with different orders. It is obvious from these statistics that the performance benefit of coupled equations over decoupled equations grows with the difficulty of the reduction. While for zone 30858#2 with order 2 Laporta's algorithm actually produces better coefficients and runs faster, its coefficients in zone 29703#3 with order 8 are roughly 100 times larger than those of the MORA. As a result, the reduction using Laporta's algorithm takes almost 2 days[11], while the reduction to coupled difference equations only needs about a minute starting from the same input, where most of that minute is even spent on organising the reduction rather than the rational algebra. These extreme differences in computation time can also show up in zones like 31246#6, which do not have a particularly high order themselves, but may have subzones with higher orders. While the orders of zones for the 5-loop tadpoles reach up to 20, the highest order listed in table 5.2 is 8. This is due to the fact that I was not able to reduce zones with higher orders including subzones via Laporta's algorithm for comparison in a reasonable amount of time.

[11] See section 8 for hardware configuration.

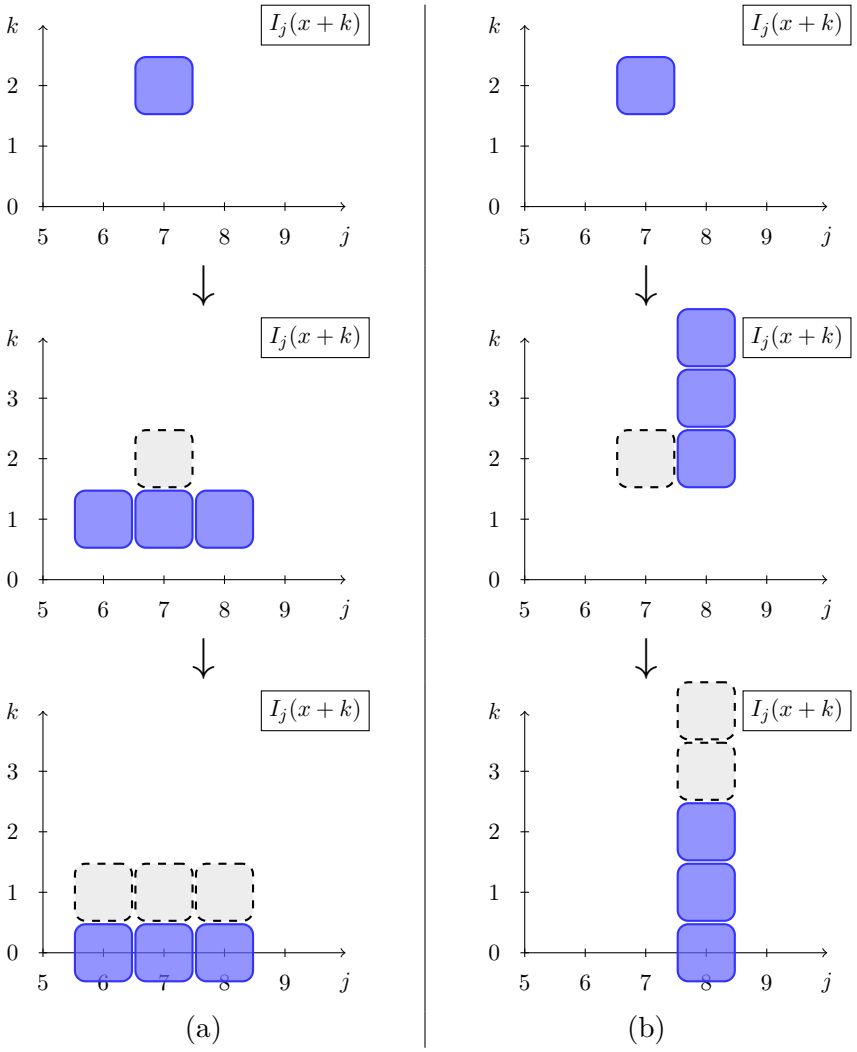


Figure 5.4: Pictorial representation of the reduction of a single integral $I_7(x+2)$ in an equation (which might have additional integrals not shown here). (a) shows the reduction using 3 coupled difference equations of order 1 for integrals I_6, I_7, I_8 (compare with fig. 5.3 (a)-(c)), while (b) shows the reduction in a system with one decoupled difference equation of order 3 for integral I_8 (compare with fig. 5.2). Dashed boxes represent integrals which were eliminated from the previous picture, which means 4 elimination steps are necessary to reduce the initial integral in the coupled system, while only 3 are needed in the decoupled system. This difference in number of steps would be more pronounced for orders higher than 3 or offsets greater than 2.

algorithm	#eqs.	⟨#coeffs.⟩	⟨size⟩	⟨$\deg_x$⟩	⟨$\deg_d$⟩	steps	runtime
Zone 28686#1: $t = 6$, $R = 4$, 1 subzone							
LA	108	5.84	314	6.04	5.45	9131	9s
MORA	112	5.89	175	4.25	4.67	10421	7s
MORA+B	112	5.71	92	2.92	3.18	10256	4s
Zone 30858#1: $t = 7$, $R = 4$, 2 subzones							
LA	131	7.89	483	7.44	6.81	18743	22s
MORA	135	7.81	241	4.39	5.10	19804	14s
MORA+B	135	7.54	194	3.90	4.58	18671	9s
Zone 30858#2: $t = 7$, $R = 2$, 4 subzones							
LA	399	10.53	319	5.12	5.90	70541	43s
MORA	401	10.80	843	8.15	9.71	78348	68s
MORA+B	401	10.70	584	6.70	8.15	75370	43s
Zone 29703#1: $t = 7$, $R = 7$, 3 subzones							
LA	335	14	2347	11.18	13.74	40737	833s
MORA	342	13.42	218	4.29	5.28	48540	31s
MORA+B	342	13.02	84	2.53	3.04	46253	25s
Zone 29703#3: $t = 7$, $R = 8$, 4 subzones							
LA	396	17.97	29351	33.54	31.59	61139	2.06d
MORA	404	17.44	401	5.63	6.51	59364	68s
MORA+B	404	16.06	243	4.65	4.51	56277	48s
Zone 31246#6: $t = 8$, $R = 5$, 7 subzones							
LA	359	30	5529	14.38	19.63	128828	0.97d
MORA	364	29.55	822	5.34	10.12	149340	133s
MORA+B	364	29.18	302	4.38	5.42	143794	78s

Table 5.2: Statistics of the reduction to difference equations for several zones using different algorithms: Laporta's algorithm (LA), MORA and MORA with an optimised master basis (MORA+B). Except for the last two columns, all numbers are for the final reduced system including all integrals up to $r \leq r_{\max} = 2$ and $s \leq s_{\max} = 2$ where denominators have been cleared to have polynomial coefficients. Shown here are the number of equations in the zone, the average number of coefficients per equations, the average size of coefficients written as a string and the average degrees of the coefficients in x and d. The number of steps and runtime are for the reduction of the larger system with $r_{\max} = 2$, $s_{\max} = 3$.

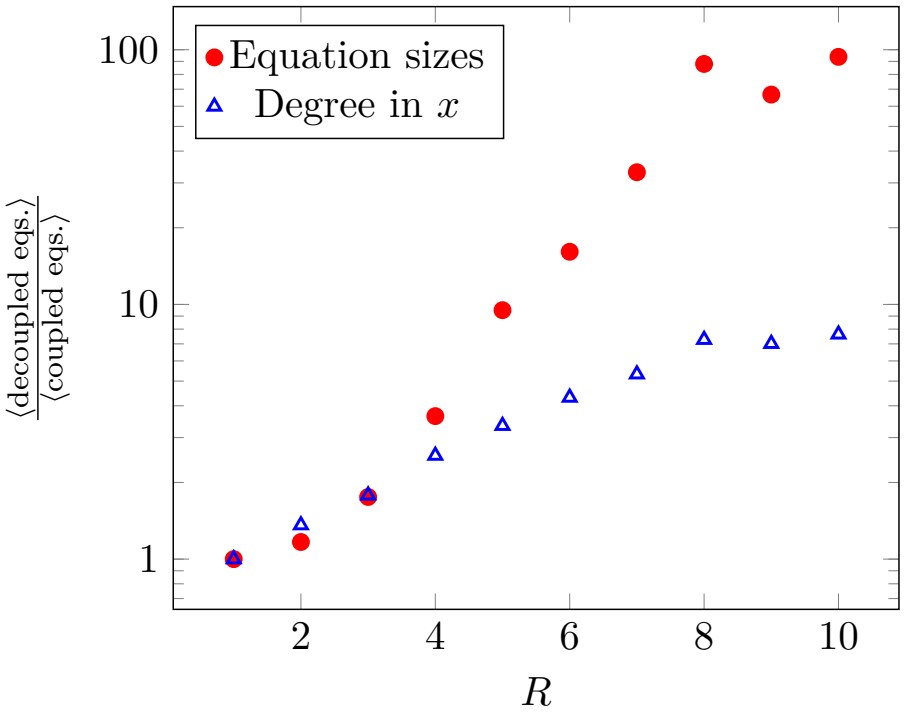


Figure 5.5: Scaling of coupled vs. decoupled equations with the order R of the zone. Shown here are ratios of average sizes as string and average x-degrees for homogeneous equations with all subzone integrals set to zero.

A larger dataset was used for figure 5.5, where I compare the scaling of the coefficients with the zone-order R in the two algorithms for all zones up to $R = 10$ that I was able to reduce using Laporta's algorithm. All subzones were set to zero in these reductions to avoid their orders influencing the result and to be able to include reductions which would have taken too long otherwise. The data shown in figure 5.5 includes all equations for integrals up to parameters $s_{\max} = r_{\max} = 2$. It is obvious from the plot that the size of coefficients in coupled equations scales much better than in decoupled equations. Beyond $R = 8$ this trend appears to be slightly unstable, but this is likely to be a statistical coincidence, since there are very few zones per order for $R = 9, 10$ which I could consider in the plot (see table B.6 for a list of orders by zone).

6 Translation and recurrence relations

As a consequence of switching to coupled difference equations, all integrals in the master basis of a given zone will also have coupled recurrence relations for the coefficients a_s of their respective factorial series expansions. The general form of such a recurrence relation is given by

$$\sum_j \sum_{k=k_b^{(j)}}^{k_t^{(j)}} g_{jk}(s) a_{j,s+k} = 0 \,, \tag{6.1}$$

where j is the index assigned to the integrals and thus the factorial series coefficients by the ordering relation. When translating R coupled difference equations, one obtains R linearly independent recurrence relations of this form, which may also depend on factorial series coefficients of master basis integrals of previously solved subzones. To iteratively evaluate the $a_{j,s}$ using these recurrence relations one would ideally like to have their structure such that for each index j in the master basis, there is one equation (6.1) such that $k_t^{(j)} > k_t^{(i)} \forall i \neq j$. With this condition it is trivial to solve for each $a_{j,s+k_t^{(j)}}$ and thus in one iteration step all $a_{j,s}$ for different j but same s can be obtained independently from each other. In principle it is also possible to have a more general setup where one iteration step calculates series coefficients $a_{j,s+k_j}$ with a different offset k_j for each integral and potentially not simultaneously, but in a fixed order. In either case one might have to build linear combinations of the initial recurrence relations obtained by translation from difference equations in order to, for each j in the master basis, have one equation which is solved for $a_{j,s+k_j}$ without depending on any series coefficients that are not known prior to $a_{j,s+k_j}$. In Laporta's original decoupled setup this was not necessary, since only one factorial series was evaluated at a time.

Building linear combinations such that for each index j there is one equation in which the highest offset is only realised for j is exactly what the MORA does for difference equations. Since the recurrence relations in eq. (6.1) have the same structure as linear equations between integrals (eq. (5.2)), the algorithm can be used as stated in section 5.2 to also reduce the former, with the only differences being that integrals are replaced with factorial series coefficients and the summation parameter s of the series now plays the role of the exponent x. While in theory this should work nicely out of the box, in practice it turns out that the reduction of the recurrence relations, if done naively, will generally be tremendously slow. In section 6.1 I show that the translation process for the reduced difference equations typically produces recurrence relations with very high spans. These go hand in hand with large coefficients (as seen in sect. 5.3), which in turn slow down the rational algebra in the reduction. The solution is to reduce the equations for integrals to a form specifically tailored to yield better recurrence relations after the

translation process, which I present in section 6.2. As a consequence of this step one has to reduce the system of factorial series coefficients of all integrals, not just the ones in the master basis. This reduction of the $a_{j,s}$ is then almost analogous to the one for integrals $I_j(x)$ and is the subject of section 6.3. The approach described here adds several steps to Laporta's initially straightforward setup, but results in recurrence relations with much smaller spans and simpler rational coefficients, which can speed up the iterative evaluation of the factorial series coefficients $a_{j,s}$ considerably.

6.1 Translation process

Recurrence relations between the $a_{j,s}$ of the form (6.1) are obtained by translating difference equations for the $I_j(x)$. Even though the translation process was already well described by Milne-Thomson [114] and Laporta [1], I repeat it here in some detail to make an important observation which is missing in those references. Instead of working with the general solution (4.4) of a difference equation, I already set the parameters $\mu = 1$ and $K = -d/2$ to simplify the discussion of the translation, since these are the only solutions that contribute for the massive tadpoles (see sect. 7.3 or ref. [1]). The generalisation to arbitrary μ and K is straightforward and is treated in section 7.2.

As a starting point of the translation process Milne-Thomson [114] defines the operators $\boldsymbol{\rho}$ and $\boldsymbol{\pi}$ by

$$\boldsymbol{\rho}^m I(x) = \frac{\Gamma(x+1)}{\Gamma(x-m+1)} I(x-m), \tag{6.2}$$

$$\boldsymbol{\pi} I(x) = x(I(x) - I(x-1)). \tag{6.3}$$

He then shows several properties of these operators, including

$$\begin{aligned}
\boldsymbol{\rho}^m \boldsymbol{\rho}^n I(x) &= \boldsymbol{\rho}^{m+n} I(x) \,, \\
\boldsymbol{\rho}^m 1 = \boldsymbol{\rho}^m &= \frac{\Gamma(x+1)}{\Gamma(x-m+1)} \,, \\
\boldsymbol{\pi} 1 &= 0 \,, \\
[\boldsymbol{\pi}, \boldsymbol{\rho}] I(x) &= \boldsymbol{\rho} I(x) \,, \\
p(\boldsymbol{\pi}) \boldsymbol{\rho}^m I(x) &= \boldsymbol{\rho}^m p(\boldsymbol{\pi} + m) I(x) \,, \\
x I(x) &= (\boldsymbol{\rho} + \boldsymbol{\pi}) I(x) \,,
\end{aligned} \tag{6.4}$$

where p is an arbitrary polynomial. Laporta [1] adds to this the convenient generalisation

$$x^n I(x) = \sum_{k=0}^{n} \left(\sum_{j=k}^{n} (-1)^{j-k} \binom{n}{j} S_{jk} \boldsymbol{\pi}^{n-j} \right) \boldsymbol{\rho}^k I(x) \,, \tag{6.5}$$

where the S_{jk} are Stirling's numbers of the second kind. Using the properties of the $\boldsymbol{\rho}$-operator, the factorial series for an integral can be rewritten as

$$I_j(x) = \sum_{s=0}^{\infty} a_{j,s} \frac{\Gamma(x+1)}{\Gamma(x+d/2+s+1)} = \sum_{s=0}^{\infty} a_{j,s} \boldsymbol{\rho}^{-d/2-s} \; . \tag{6.6}$$

Any linear equation for integrals $I_j(x)$ can be shifted in x and multiplied by its denominators to be of the form

$$\sum_j \sum_{k=0}^{R} q_{jk}(x) I_j(x-k) = 0 \; , \tag{6.7}$$

where the q_{jk} are now polynomials in x and d and R is the span of the equation. From the properties in eq. (6.4) it follows that $(x-k+1)\cdots x I(x-k) = \boldsymbol{\rho}^k I(x)$ and thus after multiplying the equation by $(x-R+1)\cdots x$ one obtains

$$\sum_j \sum_{k=0}^{R} q_{jk}(x) \left(\prod_{n=k}^{R-1} (x-n) \right) \boldsymbol{\rho}^k I_j(x) = 0 \; . \tag{6.8}$$

Inserting eq. (6.6) and $x = \boldsymbol{\rho} + \boldsymbol{\pi}$ then yields

$$\sum_j \sum_{k=0}^{R} q_{jk}(\boldsymbol{\rho}+\boldsymbol{\pi}) \left(\prod_{n=k}^{R-1} (\boldsymbol{\rho}+\boldsymbol{\pi}-n) \right) \boldsymbol{\rho}^k \sum_{s=0}^{\infty} a_{j,s} \boldsymbol{\rho}^{-d/2-s} = 0 \; . \tag{6.9}$$

By using the commutator in eq. (6.4), or alternatively eq. (6.5) before replacing x, all occurrences of $\boldsymbol{\pi}$ can be shifted to the left of $\boldsymbol{\rho}$ to obtain

$$\sum_j \sum_{k=0}^{R'} f_{j,k}(\boldsymbol{\pi}) \boldsymbol{\rho}^k \sum_{s=0}^{\infty} a_{j,s} \boldsymbol{\rho}^{-d/2-s} = 0 \; , \tag{6.10}$$

where the $f_{j,k}$ are some polynomials in $\boldsymbol{\pi}$ and d. The parameter R' is bounded by the upper limit

$$R' \leq R + N \; , \tag{6.11}$$

where

$$N \equiv \max_{j,k} (\deg_x (q_{jk}(x))) \tag{6.12}$$

is the maximum degree in x of the coefficients in the input equation. After shifting $s \to s+k$ and once more using the properties of eq. (6.4) to move the polynomials f to the right of $\boldsymbol{\rho}$ one finds

$$\sum_j \sum_{k=0}^{R'} \sum_{s=-k}^{\infty} a_{j,s+k} \boldsymbol{\rho}^{-d/2-s} f_{j,k}(\boldsymbol{\pi} - d/2 - s) = 0 \; . \tag{6.13}$$

Because $\boldsymbol{\pi}$ only acts on unity at this point, it can be dropped to obtain

$$\sum_{s=-R'}^{\infty} \boldsymbol{\rho}^{-d/2-s} \sum_j \sum_{k=0}^{R'} a_{j,s+k} f_{j,k}(-d/2-s) = 0 \tag{6.14}$$

with the additional understanding that $a_{j,s} \equiv 0$ for $s < 0$. Since each power in $\boldsymbol{\rho}$ has to vanish independently for this equation to hold for arbitrary x, one can finally extract

$$\sum_j \sum_{k=0}^{R'} a_{j,s+k} f_{j,k}(-d/2-s) = 0 \tag{6.15}$$

as the recurrence relation for the $a_{j,s}$ corresponding to the input equation (6.7). Every step in the translation process described here is reversible, which means there is a duality between the difference equations and the recurrence relations, but the reverse translation is not needed in this thesis.

The crucial new observation regarding the translation process is that the span R' of the resulting recurrence relation is bounded by

$$N \leq R' \leq N + R\,, \tag{6.16}$$

where N is the maximum degree in x of the input equation. The upper bound was already found in eq. (6.11), so I will in the following prove the lower bound. Without loss of generality I drop the index j everywhere during the proof and consider equations for a single integral $I(x)$ or its factorial series coefficients a_s. The span R' is then defined as the difference of the highest and lowest number k in the sequence $0, 1, \ldots, N+R-1, N+R$ for which the polynomials f_k in eq. (6.15) do not vanish. Since of all the coefficients $q_k(x)$ in the input equation only q_0 contributes to f_0 and $q_0 \neq 0$ by definition, one finds that $f_0 \neq 0$ is always the first non-vanishing polynomial in this list.

To find a bound for the last non-vanishing f_k, it is sufficient to consider their highest order in s. The s-degree of the f_k is limited by

$$\deg_s(f_k(-d/2-s)) \leq N + R - k \tag{6.17}$$

and for each f_k the maximal degree in s only receives contributions from terms of order x^N in eq. (6.7). It therefore suffices to consider a simplified input equation

$$x^N \sum_{k=0}^{R} q_k I(x-k) = 0 \tag{6.18}$$

with constant q_k and, after going through the translation process, find that the coefficient

of the term with the highest possible order in s of $f_k(-d/2 - s)$ is given by

$$\hat{f}_k \equiv \frac{\partial_s^{N+R-k}}{(N+R-k)!} f_k = (-1)^{N+R-k} \sum_{i=0}^{R} \binom{N+R-i}{k-i} q_i \,, \tag{6.19}$$

$$\text{with } \partial_s \equiv \frac{\partial}{\partial s} \text{ and } \binom{a}{b} \equiv 0 \text{ for } b < 0 \,.$$

Since the $(N+R+1) \times (R+1)$-matrix M defined by $M_{ki} = \binom{N+R-i}{k-i}$ has rank $R+1$ for any R and N, at most R of the $\hat{f}_k$ can be made to vanish without setting all q_i to zero and thus rendering the input equation trivial. Hence at least one of the polynomials $f_N, \ldots f_{N+R}$ does not vanish and therefore $R' \geq N$, which completes the proof.

This result is a disaster for the reduction of recurrence relations, since it means that the span of the output equations of the translation process is bounded by the x-degree and not just the span of the input equations. If one were to take an input equation Q and multiply it by x before the translation, the resulting recurrence relation would have a span which is higher by one than that of the direct translation of Q. In this sense every power of x in the input equations leads to a loss of information during the translation, some (but not necessary all) of which might be recovered by linear combinations of the recurrence relations. Since the translation requires all coefficients of the input equations to be polynomial in x, one has to multiply with every denominator that occurs, whether in the zone of the difference equation itself or in subzones. This can lead to rather high degrees in x. Even for coupled difference equations some of the more difficult zones for the 5-loop tadpoles have orders between 10-20 and equations with x-degree $N \approx 40$. Trying to reduce a system of 10 recurrence relations with span 40 is bound to fail due to the complexity of the coefficients, which even before the reduction have up to degree $N+R$ in s (see eq. (6.17)). For decoupled equations with their significantly higher degrees in x the recurrence relations are even more complicated. While they would not need to be reduced, this would still slow down the iterative evaluation of the factorial series coefficients considerably. To avoid these problems, one should clearly try to not only minimise the span R, but also the x-degree N of the input equations to the translation process.

6.2 Optimising input equations

If one wants to perform a full reduction of the recurrence relations including the factorial series coefficients of all integrals, it is crucial to start with a system of recurrence relations where no information is missing. As in the reduction to difference equations and the reduction to (ordinary) master integrals, missing information will manifest itself in the form of spurious zeroes, which can appear with arbitrarily large coefficients. If this happens, reducing the system often becomes infeasible. It is thus important to find ideal input equations for the translation process to recurrence relations, so that no information is lost. As seen in the previous section, this requires minimising both span and x-degree N of the input equations.

A natural first attempt is to simply use the initial equations of the reduction to difference equations as input for the translation process. All equations for integrals $I_j(x)$ generated by the methods described in section 2.3 are at most linear in x and the most important method (IBP) produces equations with span $R \leq 2$. It turns out, however, that $(N \leq 1, R \leq 2)$ is not sufficient to prevent loss of information. If one simply translates the initial integral equations, the resulting system of recurrence relations generally still contains numerous spurious zeroes which prevent a reduction of the system. I have therefore developed a modified version of the MORA, which also tries to minimise the span of all equations, but keeps $N \leq 1$ everywhere at all times during the reduction. Since this restriction renders the algorithm considerably more complicated and lengthy than the MORA, I will not show it here in full detail, but instead present its core ideas, which are listed in algorithm 3.

The output of algorithm 3 consists of many equations with $(R, N) = (0, 0)$ and some with $(R, N) = (0, 1)$, $(1, 0)$ or $(1, 1)$. The $(0, 0)$-equations are particularly valuable, since they do not change at all during the translation process and thus have the same form and coefficients in the spaces of difference equations and recurrence relations. There are two steps which improve the quality of the output of algorithm 3 further. At the moment these are performed after the algorithm, but in future work I aim to incorporate them into the reduction itself to improve performance. The first of the two steps is simply to take the x-dependent equations and eliminate x from as many coefficients as possible. This is meant to deal with situations similar to

$$\begin{aligned} I_1(x) + xI_3(x) &= 0, \\ I_2(x) + xI_3(x) &= 0, \end{aligned} \tag{6.20}$$

where algorithm 3 would not find the x-independent[12] equation $I_1(x) - I_2(x) = 0$, since I_1 and I_2 would be the respective pivot integrals of the two equations. The x-independent

[12] In this context x-independent means $N = 0$.

Algorithm 3 Reduction algorithm with $N \leq 1$ based on MORA.

Require:

- An ordering relation for integrals $I(x)$.
- A set U of previously unused equations of the form (5.2), with all coefficients at most linear in x.
- Tables for keeping track of 6 different kinds of pivot equations:
  - P_t for top pivot equations with all coefficients independent of x.
  - P_t^x for top pivot equations where the pivot coefficient is linear in x.
  - P_t^c for top pivot equations where the pivot coefficient is independent of x, but at least one other coefficient is not.
  - P_b, P_b^x and P_b^c for the equivalent bottom pivot equations.

  Additionally, tables S_t, S_t^x, S_t^c, S_b, S_b^x and S_b^c for keeping track of the spans of the pivot equations. This means that e.g. $P_t^c[j]$ would be a top pivot equation for $I_j(x)$ with span $S_t^c[j]$, which depends on x in some coefficients, but not in the pivot coefficient.
- Sets of currently "benched" equations, one for top (B_t), one for bottom (B_b), initially empty.
- Pivot equations from previous runs, e.g. for subzones.

The algorithm itself is then based on the MORA (algorithm 2), with several modifications:

- Any step which increases the x-degree N of an equation or introduces denominators containing x is forbidden. This also means that x-independent eqs. can be used to remove integrals from x-dependent eqs., but not vice versa.
- All x-independent equations are reduced before the x-dependent equations.
- The pivot integral of an equation is still chosen in the same way for all equations. For P_t, P_t^c, P_b and P_b^c an equation is normalised such that the pivot coefficient is 1, for P_t^x and P_b^x such that the pivot coefficient is $x + const.$
- Whenever an equation is registered as a pivot equation, it is checked whether its x-dependence can be factored out completely and thus be dropped.
- x-dependent equations are first reduced by x-independent equations, then by other x-dependent equations. P_t^x and P_b^x are used to remove the linear parts of coefficients, P_t^c and P_b^c to remove the constant parts.

equations found in this way are then added to the system and reduced by algorithm 3.

The second way to obtain additional x-independent equations is to build linear combinations of x-dependent equations such that all coefficients in an equation are proportional to $x - \alpha$, where α can be a rational function of d. A simple example of this where $\alpha = \pm 1$ would be

$$\begin{aligned} I_1(x) + xI_2(x) + (x+1)I_3(x) &= 0 \,, \\ xI_1(x) + I_2(x) + (x+1)I_3(x) &= 0 \,, \end{aligned} \tag{6.21}$$

which algorithm 3 would not reduce further, but is in fact equivalent to

$$\begin{aligned} I_1(x) + I_3(x) &= 0 \ , \\ I_2(x) + I_3(x) &= 0 \ . \end{aligned} \tag{6.22}$$

In general finding suitable linear combinations for this second step is considerably more difficult than the first step, since a priori the values for α for which this is possible are not known and one would thus have to solve a non-linear system of equations. To express this problem in a simpler way, I rewrite the system of x-dependent equations as a matrix equation $M(x) \cdot I = 0$, where I is a vector of all integrals $I_j(x+k)$ that appear in those equations and each row of the matrix $M(x)$ contains the coefficients of one equation of the system[13]. $M(x)$ will be an $m \times n$ matrix with $m \leq n$ of the form

$$M(x) = M_0 + xM_1 \ , \tag{6.23}$$

where the previous step guarantees that M_1 has rank m. Whether an x-independent equation can be constructed as described above is equivalent to whether non-trivial x-independent vectors u, v exist such that for some α one has

$$M^T(x) \cdot u = (x - \alpha)v \ . \tag{6.24}$$

Rewriting this condition as $M_0^T \cdot u = -\alpha v$, $M_1^T \cdot u = v$ one finds

$$M^T(\alpha) \cdot u = 0 \ . \tag{6.25}$$

The possible values of α are thus those for which $M^T(\alpha)$ has a non-trivial kernel, i.e. $M(\alpha)$ does not have full rank. To find these values, it is sufficient to bring $M(\alpha)$ into row echelon form with pivot entries 1 for each row. Since M_1 has full rank, no row will become trivial in this process and thus the only values α for which $M(\alpha)$ could have less than full rank are the roots of the denominators in the row echelon form. For each of these roots eq. (6.24) turns into a system of linear equations which can be solved to see if a non-trivial x-independent equation arises. Any additional equations found in this way are then also reduced by algorithm 3.

Using algorithm 3 in combination with these two steps yields a system of equations which is well suited for the translation process. On average, equations with $(R, N) = (0, 0)$ are found for more than 90% of all integrals. For the remaining integrals one finds sufficiently many equations with $(R, N) = (1, 0)$, $(0, 1)$ or $(1, 1)$ to be able to reduce the system of recurrence relations after the translation process.

[13] It is sufficient to only consider top equations, since the bottom equations are equivalent. To account for the possibility of x-shifts, equations with span 0 will be entered with both offset 0 and offset 1 to cover the same integrals as top eqs. with span 1.

6.3 Reducing recurrence relations

Once the integral equations with minimised R and N have been translated to recurrence relations, the latter can be reduced with the MORA. In analogy with the reduction to difference equations, for a given zone Z this will yield top and bottom equations of the form

$$\begin{aligned} a_{i,s+1} + \sum_{\substack{j \\ a_j \in M'_Z}} g_{ij}^{(t)}(s) a_{j,s} + \sum_{\substack{j \\ a_j \in sub(M'_Z)}} g_{ij}^{(t)}(s) a_{j,s} = 0 \;, \quad a_i \in M'_Z, \\ a_{i,s-1} + \sum_{\substack{j \\ a_j \in M'_Z}} g_{ij}^{(b)}(s) a_{j,s} + \sum_{\substack{j \\ a_j \in sub(M'_Z)}} g_{ij}^{(b)}(s) a_{j,s} = 0 \;, \quad a_i \in M'_Z, \end{aligned} \tag{6.26}$$

where M'_Z is the master basis of the factorial series coefficients. All $a_i \notin M'_Z$ will be expressed in terms of coefficients in M'_Z and $sub(M'_Z)$ by recurrence relations of order 0. In general M'_Z is not simply the set of factorial series coefficients of the integral master basis M_Z. For many zones the orders $R = |M_Z|$ and $R' = |M'_Z|$ are identical, but the two master bases do not necessarily contain matching sets of integrals and series coefficients. In several other zones one finds $R' > R$, and thus the reduction of recurrence relations is usually more difficult than the reduction of difference equations. For the 5-loop fully massive tadpoles the maximum order of a zone for recurrence relations is 28, compared to order 20 for the difference equations. A full list of orders is given in table B.7. It is also possible to have $R' < R$ as in zone 32266#1, where $R = 6$ and $R' = 5$, but this is the case only for very few zones.

For difference equations the main advantage of a minimal order reduction over a reduction to decoupled equations is that equations with lower spans contain considerably simpler rational coefficients. The same holds true for recurrence relations and one can once again choose the master basis M'_Z in a way that optimises the complexity of these coefficients. There is however also an additional advantage of the new method for recurrence relations beyond the advantages also found for difference equations. Laporta's original method translated a single difference equation in a zone Z with order R to a single recurrence relation with order R'_L. By reducing the full set of recurrence relations as described above one often finds that the zone actually has an order R' for the recurrence relations which is lower than R'_L. The simplest example for this is the 2-loop sunset zone 7#1. This zone has an order 2 difference equation

$$\begin{aligned} I_1(x+2) + \frac{-2x+d-3}{3x+3} I_1(x+1) + \frac{-x+d-2}{3x+3} I_1(x) \\ + \frac{(-2d+4)x + d^2 - 2d}{6x(x+1)} I_2(x) = 0 \;, \end{aligned} \tag{6.27}$$

which was already seen in section 4.1, where $I_1(x) = I(x, 1, 1)$ is the sunset integral and $I_2(x) = I(x, 1, 0)$ is in the 1-loop squared subzone 6#1. Translating this difference equation yields the recurrence relation

$$\begin{aligned} a_{1,s+3} + \frac{-(2s+d+4)(22s+5d+56)}{16s+56} a_{1,s+2} \\ + \frac{(2s+d+2)(2s+d+4)(10s+4d+17)}{16s+56} a_{1,s+1} \\ \frac{-3(2s+d)(2s+d+2)^2(2s+d+4)}{64s+224} a_{1,s} + \text{subzone} = 0 \end{aligned} \tag{6.28}$$

with order 3. By translating integral equations with minimised R and N and reducing the results, one finds that the zone actually only has order 2 for the recurrence relations, which can be decoupled to

$$\begin{aligned} a_{1,s+2} + \frac{-(2s+d+2)(14s+5d+22)}{16s+40} a_{1,s+1} \\ + \frac{3(2s+d)(2s+d+2)^2}{32s+80} a_{1,s} + \text{subzone} = 0 \, . \end{aligned} \tag{6.29}$$

This recurrence relation will of course produce the same values for the $a_{1,s}$ as the one with order 3, but since it has less terms (and simpler coefficients), the numeric evaluation will be faster.

In this example the higher order in Laporta's approach can be traced back to the denominator of the subzone integral in the difference equation (6.27). This denominator carries a factor of x which is not present in the other denominators, but which the whole equation has to be multiplied with to have polynomial coefficients for the translation process. This increases the x-degree of the coefficients in zone 7#1 and thus the order of the resulting recurrence relation. If the subzone integral is set to zero before the translation, one finds the same order 2 recurrence relation as with the new method, but without the subzone information. The difference between R' and R'_L grows larger for more difficult zones, which generally have a lot more subzones that can contain different denominators. While subzone denominators are the main cause of this difference, it can also appear independent of them. An example for this is the 4-loop zone 952#1 which has a difference equation with order 2 that translates into an order $R'_L = 5$ recurrence relation when all subzone integrals are set to zero, while the full reduction of recurrence relations reveals $R' = 4$.

There is one other method to avoid spurious zeroes in the reduction of the recurrence relations besides reducing R and N of the input equations as described in section 6.2. By redefining integrals as $I'_j(x) = I_j(x + k_j)$ for some offsets k_j and expanding the I'_j as factorial series instead of the I_j the series coefficients $a_{j,s}$ and therefore the recurrence relations are changed. The introduction of the offsets k_j can make a difference in the

translation process, since integrals with different offsets are translated differently. As an example, consider an integral equation with span 1. The equation would be multiplied with x during the translation to use $xI(x-1) = \boldsymbol{\rho} I(x)$ to unify the arguments of the integrals. The term $\boldsymbol{\rho} I(x)$ would then in the end contribute to a single term in the recurrence relation since it is of uniform degree in $\boldsymbol{\rho}$. Starting from $xI(x) = (\boldsymbol{\rho} + \boldsymbol{\pi})I(x)$ however would yield contributions to the factors of both a_s and a_{s+1} and might thus increase the span of the recurrence relation for this factorial series coefficient. There are several choices for the offsets k_j for which this difference during the translation leads to significantly fewer spurious zeroes in the recurrence relation when translating IBP-relations directly, without first reducing them using algorithm 3. One such choice is to define the k_j such that all $I'_j(x)$ have the same mass dimension. While this method provides a conceptually simpler way to avoid spurious zeroes than algorithm 3, I have found that, at least for the 5-loop tadpoles, it produces difference equations and recurrence relations with more complicated rational coefficients. Furthermore it is unclear whether this heuristically found decrease in spurious zeroes is guaranteed to occur for all zones. For these reasons I have chosen algorithm 3 over this option.

7 Evaluation via factorial series

The top recurrence relations in eqs. (6.26) are ideally suited for the iterative evaluation of the factorial series coefficients. For each s all $a_{j,s}$ with $a_j \in M'_Z$ can be calculated simultaneously from the $a_{j,s-1}$, which makes it easy to parallelise the computation. Whenever the pivot coefficient of a recurrence relation vanishes for a particular value of s, the corresponding series coefficient $a_{j,s}$ is needed as an initial condition for the iteration (see sect. 7.3). For coupled recurrence relations of order 1 this typically only happens for $s = 0$. The top recurrence relations can then be used to obtain all $a_{j,s}$ from $s = 1$ to $s_{\max}$ for a_j in the master basis M'_Z of a given zone as ϵ-expansions around the chosen dimension. The finite sums up to $s_{\max}$ over the factorial series in eq. (4.5) are then performed for $x = x_{\max}$ and yield the $I_j(x_{\max})$ corresponding to M'_Z. If the set of integrals obtained in this way is not the master basis M_Z, values for the latter can be derived from the former using integral equations with span 0. Using the bottom difference equations in (5.3) the argument x in the integrals of M_Z is then pushed down from $x_{\max}$ to 1 in the same iterative manner as s was increased for the $a_{j,s}$ with the recurrence relations.

In the following I discuss several aspects of the above described process in greater detail. Section 7.2 covers the general solution of difference equations in terms of factorial series and how the allowed values for the parameters μ and K in eqs. (4.3) and (4.4) are obtained. The starting point of the process is then the determination of the initial conditions, which is described in section 7.3. Afterwards I discuss convergence and numerical errors for the iterative application of both recurrence relations and difference equations (sect. 7.4), before making some remarks on possible issues that can arise in practice (sect. 7.5).

7.1 Interlude: Fibonacci numbers

A very simple example of linear recurrence relations defines the well-known sequence of Fibonacci numbers f_n, $n \in \mathbb{N}$. They satisfy the condition

$$f_{n+2} = f_n + f_{n+1} \tag{7.1}$$

with the initial conditions $f_0 = 0$ and $f_1 = 1$. The first few terms in the sequence are thus $0, 1, 1, 2, 3, 5, 8, \ldots$. Equation (7.1) has a much simpler structure than the difference equations for integrals or recurrence relations for factorial series coefficients encountered in this thesis, since it has no inhomogeneous (independent of f_n) terms and the coefficients of the f_{n+k} do not depend on n. Nevertheless it is instructive to study the solution of this simple recurrence relation, as much of the structure of the solution carries over to the more difficult cases.

By defining the operator $\hat{\mu}_+$ via

$$\hat{\mu}_+ f_n = f_{n+1} \tag{7.2}$$

the Fibonacci recurrence relation can be rewritten as

$$p(\hat{\mu}_+) f_n = (\hat{\mu}_+^2 - \hat{\mu}_+ - 1) f_n = (\hat{\mu}_+ - \mu_1)(\hat{\mu}_+ - \mu_2) f_n = 0 \,, \tag{7.3}$$

where $p(\hat{\mu}_+)$ is called the *characteristic polynomial* and its roots are

$$\mu_1 = \frac{1+\sqrt{5}}{2} \,, \quad \mu_2 = \frac{1-\sqrt{5}}{2} \,. \tag{7.4}$$

From the factorised form it is easy to identify the two independent solutions of the recurrence relation as μ_1^n and μ_2^n and thus the general solution is

$$f_n = c_1 \mu_1^n + c_2 \mu_2^n \,. \tag{7.5}$$

The coefficients c_1 and c_2 have to be determined from the initial conditions, which for the Fibonacci numbers yields

$$f_n = \frac{\mu_1^n - \mu_2^n}{\sqrt{5}} \,. \tag{7.6}$$

7.2 The general solution

The general solution of a difference equation of order R for a single integral $I(x)$ in terms of factorial series has the form [1, 114]

$$I(x) = \sum_m \mu_m^x \sum_{s=0}^{\infty} \frac{\Gamma(x+1)}{\Gamma(x+s-K_m+1)} a_s^{(m)} \ . \tag{7.7}$$

This is a sum of the solution of inhomogeneous difference equation and up to R solutions of the homogeneous difference equation where subzones are set to zero. The possible values for μ and K for the homogeneous solutions are determined by the recurrence relations, while the values for the inhomogeneous solutions are simply those which already appear in subzones.

For the fully massive tadpoles with mass $m^2 = 1$ Laporta [1] has shown that only solutions with $\mu = 1/m^2 = 1$ and $K = -d/2$ contribute to the general solution (see also sect. 7.3), which is why those values are already assumed in the description of the translation process in section 6.1. Nevertheless the other possible values for μ in the homogeneous solution play an important role for the numerical error in the determination of the factorial series coefficients and integrals (see sect. 7.4). In the following I therefore describe how the μ_m are determined for a difference equation of a single integral [1, 114] and show that these values do not change if one switches to coupled difference equations and recurrence relations.

Starting from a homogeneous difference equation

$$\sum_{k=0}^{R} q_k(x) I(x-k) = 0 \tag{7.8}$$

the parameter μ is introduced before the translation by replacing

$$I(x) \to \mu^x I(x) \ , \tag{7.9}$$

or equivalently $q_k \to \mu^{R-k} q_k$. The recurrence relation then takes the form

$$\sum_{k=0}^{N+R} f_k(K-s, \mu) a_{s+k} = 0 \ , \tag{7.10}$$

where the polynomials f_k now depend on μ and K, and N is once again the x-degree of the input equation. The terms $\hat{f}_k$ with the highest possible s-degree in the f_k (see eq. (6.19))

are given by

$$\hat{f}_k(\mu) = (-1)^{N+R-k} \sum_{i=0}^{R} \binom{N+R-i}{k-i} \hat{q}_i \mu^{R-i} \,, \tag{7.11}$$

where the $\hat{q}_i$ are the coefficients of x^N in the $q_i(x)$ and the $\hat{f}_k$ do not depend on K. Of particular interest is the *characteristic polynomial* of the difference equation, defined by

$$p(\mu) \equiv f_{N+R}(K-s,\mu) = \hat{f}_{N+R}(\mu) = \sum_{i=0}^{R} \hat{q}_i \mu^{R-i} \,, \tag{7.12}$$

since in the recurrence relation (7.10) $p(\mu)$ is the coefficient of the a_s with the highest offset. Due to $a_{s<0} = 0$ one finds

$$p(\mu)a_0 = 0 \,, \quad p(\mu)a_1 \propto a_0 \,, \ \ldots \tag{7.13}$$

and thus $p(\mu)$ must vanish in order to have non-zero factorial series coefficients. Therefore the roots $\mu_1, \ldots, \mu_n$ of the characteristic polynomial with $n \equiv \deg_\mu(p(\mu)) \leq R$ are the only allowed values for μ in the homogeneous solutions. This is in analogy with the example of the Fibonacci numbers in section 7.1. The only difference is that unlike for the Fibonacci recurrence relation the coefficients for the difference equations of integrals may depend on x and the characteristic polynomial is only determined by the large-x behaviour of the difference equation and not its full coefficients.

Upon choosing $\mu = \mu_m$ the order of the recurrence relations is reduced and a different coefficient $f_{k_m}(K-s,\mu_m)$ becomes the top coefficient. The first iteration of the recurrence relation would then be

$$f_{k_m}(K+k_m-s,\mu_m)|_{s=0} a_0 = 0 \tag{7.14}$$

and thus the roots $K_1^{(m)}, \ldots$ of $f_{k_m}(K+k_m,\mu_m)$ are the only values for K which lead to a non-vanishing homogeneous solution in conjunction with μ_m. If two roots $K_i^{(m)}, K_j^{(m)}$ differ by an integer z, their corresponding factorial series can be merged by picking the greater root as K and requiring both a_0 and a_z as initial conditions, since both will be undetermined from the recurrence relations. The original suggestion of Laporta [1] for this case was to evaluate a factorial series each for both K and to set $a_z = 0$ in the series of the greater root. This gives equivalent results, but it is nicer to avoid evaluating two series instead of one. The only time where it might be beneficial to follow this suggestion is when determining a_z as an initial condition proves to be too difficult for large z (see sect. 7.3).

When switching from a single difference equation to coupled difference equations, it is trivial to show that the allowed values for μ are the same for all integrals in the master basis M_Z. If $I_c(x)$ is the integral for which Laporta's algorithm found a decoupled difference

equation of order R, then the reduction will also contain equations solving the other integrals in the master basis in terms of I_c in the form

$$I_j(x) + \sum_{k=0}^{R-1} p_{jk}(x) I_c(x+k) = 0 \ , \ j \neq c \ , \ I_j \in M_Z \ . \tag{7.15}$$

Since the I_j are simply linear combinations of I_c with different offsets and the coupled and decoupled forms of the system of equations are equivalent, the general solution of the coupled difference equations has the same values for μ as the decoupled difference equation for all integrals. Due to the different offsets in eq. (7.15) the allowed values of K might change by integers, but as seen above this is merely a question of finding different initial conditions.

If the decoupled difference equation is not known, the roots μ_i of the characteristic polynomial can also be obtained directly from the coupled difference equations. As in the decoupled case, for each equation only the terms with the highest x-degree contribute to $p(\mu)$. The R difference equations of span 1 can always be combined such that after dropping all terms except those with the highest degrees, the combinations are still linearly independent and can be written as

$$x^{N_i} \sum_{j=1}^{R} \sum_{k=0}^{1} \hat{q}_{ij,k} I_j(x-k) = 0 \ , \quad i = 1, \ldots, R. \tag{7.16}$$

By decoupling these equations explicitly, one finds that

$$p(\mu) = \text{const} \cdot \det A(\mu) \ , \quad A_{ij}(\mu) \equiv \mu \hat{q}_{ij,0} + \hat{q}_{ij,1} \ . \tag{7.17}$$

Alternatively, eqs. (7.16) can be translated to recurrence relations, where the polynomials with the highest offsets of the equation are then given by

$$f_{ij,N_i+1}(K-s,\mu) = \hat{f}_{ij,N_i+1}(\mu) = \mu \hat{q}_{ij,0} + \hat{q}_{ij,1} = A_{ij}(\mu) \quad \text{(compare eq. (7.12))}. \tag{7.18}$$

The recurrence would thus start at $s = 0$ with

$$\sum_{j=1}^{R} A_{ij}(\mu) a_{j,0} = 0 \quad \forall i \tag{7.19}$$

and one again obtains the condition $\det A(\mu) = 0$ to have non-zero homogeneous solutions of the difference equations.

7.3 Initial conditions

In reference [1] Laporta shows how to obtain the first factorial series coefficients a_0, a_1, ... for several classes of integrals by examining their large-x behaviour. For the fully massive tadpoles considered in this thesis only the Euclidean massive case discussed in section 5.1 of [1] is relevant and I briefly summarise it here.

Given a fully massive Feynman integral $I(x)$, the loop momenta can always be shifted such that the propagator carrying the exponent x has momentum k_1. After separating the radial part of the k_1-integration, $I(x)$ can be written as

$$I(x) = \frac{1}{\Gamma(d/2)} \int_0^\infty \frac{\mathrm{d}k_1^2 (k_1^2)^{d/2-1}}{(k_1^2 + m_1^2)^x} f(k_1^2) \ , \tag{7.20}$$

where $f(k_1^2)$ is the angular mean over k_1 of the $k_2, \ldots, k_L$-integration over the remaining propagators. Since all propagators are massive and there are no external momenta, the integrand has no singularities for $k_1^2 \geq 0$. For large x the integral is thus dominated by the peak of the first propagator at $k_1^2 = 0$. After expanding $f(k_1^2)$ as

$$f(k_1^2) = \sum_{s=0}^{\infty} f_s k_1^{2s} \tag{7.21}$$

and changing variables via $k_1^2 = m_1^2 \frac{u}{1-u}$ with $\mathrm{d}k_1^2 = \mathrm{d}u\, m_1^2 (1-u)^{-2}$ it can be expressed as

$$f(m_1^2 \frac{u}{1-u}) = (1-u)^{d/2+1} \sum_{s=0}^{\infty} b_s u^s \ . \tag{7.22}$$

Inserting this into the integral yields

$$I(x) = \frac{(m_1^2)^{d/2-x}}{\Gamma(d/2)} \int_0^1 \mathrm{d}u\, u^{d/2-1} (1-u)^x \sum_{s=0}^{\infty} b_s u^s = \left(\frac{1}{m_1^2}\right)^x \sum_{s=0}^{\infty} \frac{a_s \Gamma(x+1)}{\Gamma(x+s+d/2+1)}, \tag{7.23}$$

which is a factorial series with coefficients

$$a_s = b_s m_1^d \frac{\Gamma(s+d/2)}{\Gamma(d/2)} \ . \tag{7.24}$$

Comparison with the general solution (7.7) of the difference equation shows that for the fully massive tadpoles only solutions with $\mu = 1/m_1^2 = 1$ and $K = -d/2$ can contribute. The factorial series coefficients a_s can be calculated in terms of the f_s in eq. (7.21) with a_s depending on $f_0, f_1, \ldots, f_s$. The coefficients f_s are in turn obtained by expanding $f(k_1^2)$, which means each f_s will be a sum of integrals independent of x and k_1 and have one loop less than $I(x)$. These integrals have to be moved into a classification scheme for $L-1$ loop

integrals as defined in section 2.1 by applying suitable momentum shifts. Afterwards they can be expressed in terms of the master integrals at that loop level, which are assumed to already be known.

While in theory all terms a_s of the factorial series could be calculated in this way, in practice only the first few terms can be obtained. If $f(k_1^2)$ has a k_1-dependence in the denominator, the number of dots r in the integrals $I_j^{(s)}$ in f_s grows with the expansion parameter[14] s. Since the number of integrals in a reduction increases significantly with the number of dots (see eq. (2.12)), expressing the $I_j^{(s)}$ in terms of master integrals is only feasible for small s. When using decoupled equations, this can potentially cause problems if some a_s is undetermined from the recurrence relations. In this case more than one factorial series might need to be evaluated for the general solution in eq. (7.7) (see sect. 7.2). For all coupled recurrence relations of order 1 used in this thesis it was sufficient to determine only the first coefficient $a_{j,0}$ for each integral as the initial condition.

In principle it should be possible to reduce the number of $a_{j,s}$ that need to be obtained as initial conditions by finding linear relations between them that do not depend on s. This would reduce the $a_{j,s}$ to some master coefficients, much like reducing x-independent integrals to master integrals. I have not tested this idea, since it is not needed for the fully massive tadpoles if using coupled recurrence relations, but it might prove useful for decoupled recurrence relations or more difficult classes of integrals, where obtaining the initial conditions is more involved [1]. To generate non-trivial equations between the $a_{j,s}$, I suggest expanding the factorial series and difference equations in $1/x$. The integrals would then be written as

$$\begin{aligned} I_j(x) &= \sum_{s=0}^{\infty} a_{j,s} \frac{\Gamma(x+1)}{\Gamma(x+s-K+1)} \\ &= \frac{\Gamma(x+1)}{\Gamma(x+K+1)} \sum_{s=0}^{\infty} a_{j,s} \frac{1}{x^s} \prod_{i=1}^{s} \sum_{n=0}^{\infty} \left(\frac{K-i}{x}\right)^n \\ &= \frac{\Gamma(x+1)}{\Gamma(x+K+1)} \sum_{l=0}^{\infty} \frac{h_{jl}(a_{j,0}, a_{j,1}, \dots a_{j,l})}{x^l} \,. \end{aligned} \tag{7.25}$$

Integrals with non-zero offsets can be expanded into the same form. When inserting this into a difference equation, the Γ-functions can be dropped, since they are identical for all integrals. The whole equation can then be expanded in $1/x$ and each order should yield a relation between the factorial series coefficients. In particular, the first order $1/x^0$ gives an equation only involving the $a_{j,0}$. If the equation is not trivial, it clearly contains information which cannot be obtained from the recurrence relations for the master basis M'_Z, since they all have spans greater than 0.

[14] Here s is not to be confused with the number of numerator powers defined in eq. (2.6).

7.4 Precision and convergence

Whether a factorial series

$$\sum_{s=0}^{\infty} \frac{a_s \Gamma(x+1)}{\Gamma(x+s-K+1)} \tag{7.26}$$

converges depends on the large-s behaviour of the a_s. In analogy with the difference equations, the general solution of a recurrence relation has the form

$$a_s = \sum_i \lambda_i a_s^{(i)} \,, \tag{7.27}$$

where the λ_i are the roots of the characteristic polynomial $p'(\lambda)$ of the recurrence relation. Unlike in the case of difference equations, the s-degree of coefficients in the recurrence relations decreases with larger offsets (see eq. (6.17)) and thus for large s the solutions of the recurrence relations grow like a factorial. To study the large-s behaviour in more detail the homogeneous part of a (decoupled) recurrence relation can be written as

$$\sum_{k=0}^{R'} \left(\alpha_k s + \beta_k + \mathcal{O}\left(\frac{1}{s}\right) \right) s^{-k} a_{s+k} = 0 \,. \tag{7.28}$$

The characteristic polynomial is then defined as

$$p'(\lambda) = \sum_{k=0}^{R'} \alpha_k \lambda^k \,. \tag{7.29}$$

For large s the solutions of the homogeneous recurrence relation tend to the form

$$\begin{aligned} a_s^{(i)} &\sim c_i \lambda_i^s \Gamma(s+\sigma_i+1) \sim c_i \lambda_i^s s! s^{\sigma_i} \,, \\ a_{s+k}^{(i)} &\sim \lambda_i^k \left(s^k + \sum_{j=1}^{k} (\sigma_i + j) s^{k-1} + \dots \right) a_s^{(i)} \\ &\sim \lambda_i^k s^{k-1} \left(s + k\sigma_i + \frac{k(k+1)}{2} + \mathcal{O}\left(\frac{1}{s}\right) \right) a_s^{(i)} \,, \end{aligned} \tag{7.30}$$

where c_i is a constant and σ_i can be determined by inserting this ansatz into the recurrence relation, which yields

$$\sigma_i = -\frac{\sum_{k=0}^{R'} \left(\frac{k(k+1)}{2} \alpha_k + \beta_k \right) \lambda_i^k}{\sum_{k=0}^{R'} k \alpha_k \lambda_i^k} \,. \tag{7.31}$$

With eq. (7.30) the generic term of the factorial series tends to

$$\frac{a_s^{(i)} \Gamma(x+1)}{\Gamma(x+s-K+1)} \overset{s\to\infty}{\sim} c_i \Gamma(x+1) \lambda_i^s s^{\sigma_i + K - x} \,. \tag{7.32}$$

For $|\lambda_i| < 1$ the series thus converges exponentially, but if a solution with $|\lambda_i| > 1$ contributes to the integral, the corresponding factorial series diverges and the method breaks down. For the fully massive tadpole integrals only solutions with $\lambda = 1$ can contribute, since no exponential dependence on s appears in the exact solution for the a_s in eq. (7.24). In this case the series shares the convergence behaviour with the Dirichlet series

$$\sum_{s=0}^{\infty} g_s s^{\sigma_i + K - x} \tag{7.33}$$

with bounded g_s. The series thus converges on a complex half-plane bounded on the left by the *line of convergence* defined by $\mathrm{Re}(x) = \eta$, where $\eta \geq \sigma_i + K + 1$ is called the *abscissa of convergence*. The solution of the inhomogeneous recurrence relation might have stronger constraints on the abscissa of convergence, but will still have $\lambda = 1$ as the the inhomogeneous parts of the equation originate in the same class of integrals. The parameter $x_{\max}$ for which the factorial series is evaluated must thus be chosen such that $x_{\max} > \eta$, but larger $x_{\max}$ will accelerate the convergence. As a result of the limitation to $\lambda = 1$, all factorial series for massive tadpoles converge for appropriately chosen $x_{\max}$. I expect that this limitation and thus the convergence of the factorial series also holds for many other classes of integrals, but a general proof would be cumbersome. I do however prove below that $\lambda = 1$ is always a root of the characteristic polynomial of the recurrence relation corresponding to any non-trivial difference equation.

Even though for the fully massive tadpoles only the roots $\mu = 1$ and $\lambda = 1$ of the respective characteristic polynomials of difference equation and recurrence relation may contribute to the analytical solution, the remaining roots still play an important role in the numerical evaluation. This can be illustrated nicely in the simple example of the Fibonacci recurrence relation (see sect. 7.1), which has the roots

$$\mu_1 = \frac{1+\sqrt{5}}{2}\,, \quad \mu_2 = \frac{1-\sqrt{5}}{2}\,. \tag{7.34}$$

If the initial conditions for the recurrence relation are chosen as $f_0 = 1$, $f_1 = \mu_2$, the analytical solution to all orders is simply

$$f_n = \mu_2^n\,. \tag{7.35}$$

If however a small numerical error is introduced, e.g. $f_0 = 1$, $f_1 = \mu_2 + 10^{-10}\mu_1$, the μ_1-branch of the general solution will quickly dominate the numerical results, since it grows faster due to $|\mu_1| > |\mu_2|$, and one would find

$$f_n \overset{n\to\infty}{\approx} 10^{-10}\mu_1^n\,. \tag{7.36}$$

The numerical evaluation thus favours the larger root in the limit $n \to \infty$, even though the initial conditions are almost the same as those for which the exact solution only contains the smaller root μ_2.

During the iteration of difference equations and recurrence relations for integrals the same problem occurs. For simplicity I restrict the discussion to fully decoupled equations, but the behaviour is identical in the coupled case. The general solution of a difference equation with order R is given in eq. (7.7) in terms of solutions $\mu^x I^{(m)}(x)$, where each $I^{(m)}(x)$ is expanded as a factorial series. Since any numerical value for a particular solution $I^{(m)}(x)$ always contains a small error, the numerical sequence $\{I^{(m)}(x), I^{(m)}(x+1), \ldots, I^{(m)}(x+R-1)\}$ is a linear combination of the exact sequences $\{I^{(i)}(x), I^{(i)}(x+1), \ldots, I^{(i)}(x+R-1)\}$ in the space of all solutions of the difference equation. The numerical sequence thus in general has (preferably small) contributions from solutions with $i \neq m$. When using the difference equation to calculate $I^{(m)}(x-1)$ from this sequence, the contributions from other solutions will scale with different values of μ than the exact solution for $I^{(m)}(x)$. The numerical error when pushing down the value of x with the difference equation thus grows by a factor

$$F_P^{(m)} = \max_i \left| \frac{\mu_m}{\mu_i} \right| \tag{7.37}$$

in each iteration. In the following I therefore refer to $F_P^{(m)}$ as a *push-down divergence factor*. Analogously, there is also a *recurrence divergence factor*

$$F_R^{(m)} = \max_i |\lambda_i| = \max \left\{ 1, \max_{\substack{i \\ \mu_i \neq \mu_m}} \left| \frac{\mu_m}{\mu_m - \mu_i} \right| \right\} , \tag{7.38}$$

which is the factor by which the numerical error in the factorial series coefficients $a_s^{(m)}$ grows in each iteration of the recurrence relation. Here the λ_i are the roots of the characteristic polynomial of the recurrence relation that is obtained by setting $\mu = \mu_m$ and it is already assumed that only the root $\lambda = 1$ contributes to the exact solution.

Laporta states in [1] that the factorial series corresponding to $\mu = \mu_m$ with coefficients $a_s^{(m)}$ has a finite abscissa of convergence η if none of the roots μ_i satisfy the condition

$$0 < \left| \frac{\mu_i}{\mu_m} - 1 \right| < 1 , \quad \mu_i \neq \mu_m , \tag{7.39}$$

which is equivalent to $F_R^{(m)} = 1$. While this is technically true, it is also misleading. The abscissa of convergence does not depend on $F_R^{(m)}$, which only plays a role for numerical values. Furthermore, the implication that the series does not converge for $F_R^{(m)} > 1$ is incorrect. A more accurate statement is that for $F_R^{(m)} > 1$ the error in a numerical evaluation of the coefficients will diverge and start to dominate the values of $a_s^{(m)}$ for some

$s = s_{\text{div}}$. As long as the upper limit $s_{\max}$ of the sum is chosen such that $s_{\max} \ll s_{\text{div}}$, the numerical evaluation is still viable.

The second expression for the recurrence divergence factor in eq. (7.38) requires the determination of the roots λ_i of the recurrence relation in terms of the roots μ_i of the difference equation. To start, I first rewrite the characteristic polynomial $p(\mu)$ of the difference equation defined in eq. (7.12) in terms of the μ_i as

$$p(\mu) = \hat{f}_{N+R}(\mu) = \sum_{i=0}^{R} \hat{q}_i \mu^{R-i} = c \prod_{i=1}^{n} (\mu - \mu_i) \ , \quad c = \text{const} \ , \tag{7.40}$$

where the $\hat{q}_i$ are the coefficients of x^N in the difference equation and $n \equiv \deg_\mu p(\mu) \leq R$. The characteristic polynomial of the recurrence relation is given in eq. (7.29) in terms of the coefficients α_k, which are identical to the $\hat{f}_k(\mu)$ in section 7.2 and upon setting $\mu = \mu_m$ the characteristic polynomial reads

$$p'_m(\lambda) = \sum_{k=0}^{N+R-1} \hat{f}_k(\mu_m) \lambda^k \ . \tag{7.41}$$

As the form of the coefficients $\hat{f}_k(\mu)$ given in eq. (7.11) does not know about the μ_i, it is useful to rewrite them in terms of $p(\mu)$ as

$$\begin{aligned} \hat{f}_k(\mu) &= (-1)^{N+R-k} \sum_{i=0}^{R} \binom{N+R-i}{k-i} \hat{q}_i \mu^{R-i} \\ &= \frac{(-1)^{N+R-k}}{(N+R-k)!} \mu^{R-k} \left(\frac{\partial}{\partial \mu} \right)^{N+R-k} \mu^N p(\mu) \ . \end{aligned} \tag{7.42}$$

It is then instructive, but not required, to divide[15] $p'_m(\lambda)$ by $\hat{f}_{N+R-1}(\mu_m)$, since the ratios

$$\frac{\hat{f}_{N+R-t}(\mu_m)}{\hat{f}_{N+R-1}(\mu_m)} = (-1)^{t-1} \sum_{s=1}^{t} \binom{N}{t-s} \sum_{\substack{S \subset \{1,\dots m-1, m+1, \dots n\} \\ |S| = s-1}} \prod_{i \in S} \frac{\mu_m}{\mu_m - \mu_i} \tag{7.43}$$

already feature the objects $\lambda_{mi} \equiv \frac{\mu_m}{\mu_m - \mu_i}$ appearing in eq. (7.38). With these ratios the characteristic equation takes the form

$$0 = \sum_{t=1}^{N+R} \lambda^{N+R-t} (-1)^{t-1} \sum_{s=1}^{t} \binom{N}{t-s} \sum_{\substack{S \subset \{1,\dots m-1, m+1, \dots n\} \\ |S| = s-1}} \prod_{i \in S} \lambda_{mi} \ , \tag{7.44}$$

[15] For the division to be possible, μ_m has to be a root of $p(\mu)$ with multiplicity 1, since otherwise $\hat{f}_{N+R-1}(\mu_m) = 0$.

which has the roots

$$\begin{aligned} 1\,, &\quad \text{with multiplicity } N, \\ 0\,, &\quad \text{with multiplicity } R-n, \\ \lambda_{mi}\,, &\quad \text{with multiplicity 1 for each } i \neq m. \end{aligned}$$

The relative numerical error of the $a_s^{(m)}$ due to the other roots of p'_m grows with $\lambda_{mi}/1$ for each iteration of the recurrence relation and thus the value for $F_R^{(m)}$ given in eq. (7.38) is proven. For coupled difference equations the divergence factors $F_P^{(m)}$ and $F_R^{(m)}$ are identical to the ones found for decoupled equations, since both types of equations share the same values μ_i (see sect. 7.2) and as a consequence also the same λ_{mi}.

Combined with the large-s behaviour of the a_s, the divergence factors can be used to estimate in advance the precision of the result $I(1)$ in terms of the starting precision for a_0 and the parameters $s_{\max}$ and $x_{\max}$. If D_{start} is the number of correct digits in a_0, the number of correct digits of $I(x_{\max})$ obtained from summing over the factorial series is approximated by

$$D_x \approx \min \begin{cases} D_{start} - s_{\max} \log_{10}(F_R) \\ \log_{10} \left| \frac{a_0}{\Gamma(x_{\max}+d/2+1)} \frac{\Gamma(x_{\max}+d/2+1+s_{\max})}{a_{s_{\max}}} \right| \approx \log_{10} \binom{x_{\max}+s_{\max}}{s_{\max}} \end{cases} . \tag{7.45}$$

The first case takes into account the divergence of the error, while the second case estimates the precision gained by $a_s/\Gamma(x_{\max}+d/2+1+s)$ converging to zero. Ideally the two cases should be equal, which puts the condition

$$D_{start} \gtrsim \log_{10} \left[\binom{x_{\max}+s_{\max}}{s_{\max}} F_R^{s_{\max}} \right] \tag{7.46}$$

on the precision of a_0. If this condition is satisfied, the number of correct digits D_{end} obtained for $I(1)$ is estimated to be

$$D_{end} \approx D_x - x_{\max} \log_{10}(F_P) \approx \log_{10} \left[\binom{x_{\max}+s_{\max}}{s_{\max}} F_P^{-x_{\max}} \right] . \tag{7.47}$$

Finding good choices for the parameters $s_{\max}$ and $x_{\max}$ therefore crucially depends on the divergence factors F_P and F_R. The highest divergence factors appearing at each loop level for the fully massive tadpoles are listed in table 7.1 along with a set of choices for $s_{\max}$ and $x_{\max}$ which in the end yield a precision of 280 digits for the 5-loop integrals. Up to 4 loops, only a single zone has a recurrence divergence factor greater than 1, which is $F_R = 9/8$ for 4-loop zone 511#2. At 5 loops there is a significant increase in difficulty,

Loops	F_P	F_R	$x_{\max}$	$s_{\max}$	D_{end}
1	1	1	310000	2000000	390000
2	3	1	300000	2000000	240000
3	8	1	150000	2000000	100000
4	15	1.125	21500	1000000	20000
5	26.86	12.93	600	17000	280

Table 7.1: Maximal divergence factors found for the fully massive tadpoles for $L = 1 - 5$ along with an example of choices for the parameters $s_{\max}$ and $x_{\max}$ and the resulting precisions. The exact values at $L = 5$ are $F_P = 13 + 8\sqrt{3}$ (for zone 30239#3) and $F_R = 6 + 4\sqrt{3}$ (zone 30222#2).

because the recurrence divergence factors reach up to 12.93. Since this means that more than one digit is lost per iteration, this puts much stronger limits on $s_{\max}$ than at lower loops. To counter the diverging error, one has to start with a very high accuracy for a_0. Since each loop-order requires the previous one as input for the initial conditions, the lower loop integrals then have to be computed with extremely high precision.

The treatment of the recurrence divergence factor F_R on equal grounds with F_P is a significant improvement of the method over Laporta's original version, which to the best of my knowledge has not yet been considered in the literature. Laporta implied [1] that for $F_R > 1$ the factorial series does not converge for any x. This is certainly not the case for the integrals in this thesis. The recurrence relation for the a_s may have a solution which diverges, but for the massive tadpoles examined here, the initial conditions for the a_s set the coefficient of that solution to zero and only solutions with $\lambda = 1$ remain. What I observe is that even if $F_R > 1$, the terms $a_s/\Gamma(x + s + d/2 + 1)$ converge up to some $s = s_{\text{div}}$, which can be pushed arbitrarily high by increasing the precision. The observed divergence for $s > s_{\text{div}}$ is thus only due to the fact that during numerical evaluation the coefficient of the diverging solution is inevitably not exactly zero but has a finite value. While I cannot prove that the factorial series of all Feynman integrals have a finite abscissa of convergence, $F_R > 1$ is certainly not a sufficient condition for divergence of the factorial series.

The original method proposed by Laporta in reference [1] to deal with divergent factorial series is to use a Laplace transformation to convert the difference equation to a differential equation, which is subsequently solved numerically. This requires a considerable amount of additional work and one does not obtain the factorial series coefficients a_s, which might be needed as input for the inhomogeneous part of recurrence relations in higher zones. This has been done e.g. in [115] for a 3-loop partly massive vacuum integral where the

homogeneous part of the difference equation reads

$$-2(x+1)I(x+2)+3(x+2-\frac{d}{2})I(x+1)-(x+3-d)I(x)=0\,. \tag{7.48}$$

The characteristic polynomial of this equation is given by $p(\mu)=-2\mu^2+3\mu-1$ and has roots 1 and 1/2, which yields the recurrence divergence factor $F_R=2$. I suspect that in this case and others the divergence the authors observed was only present due to numerical errors and the use of Laplace's transformation could have been avoided by increasing the precision in the evaluation of the factorial series. At the moment my own implementation of the method (see sect. 8) is limited to fully massive integrals, but I aim to reproduce the result given in [115] in future work using factorial series.

7.5 Numerical evaluation in practice

One of the nice features of Laporta's original method is that integrals in different zones can be considered sequentially. In theory this is still possible when switching to coupled difference equations, but in combination with the increase in precision made necessary by the recurrence divergence factors, it is often no longer feasible in practice due to increased memory requirements. If zones are evaluated one by one, all factorial series coefficients need to be stored in case they appear in subzones of recurrence relations for more difficult integrals. For the 4-loop massive tadpoles and using the parameters given in table 7.1, this would amount to:

89	Number of integrals in the master bases M'_Z at 4 loops
$\times 10^6$	$s_{\max}$
$\times 30$	Number of terms in the ϵ-expansion
$\times 10^5$	Number of digits for each term
$\times 0.415$B	Bytes per digit if stored ideally
≈ 111TB	Disk space for all $a_{j,s}$

Storing the coefficients $a_{j,s}$ on disk is clearly not a viable option in this case and thus one has to evaluate all factorial series at a given loop level simultaneously. This allows the code to throw away coefficients $a_{j,s}$ once they have been incorporated into the sum for $I(x_{\max})$ and the $a_{j,s+1}$ have been computed, thus cutting down on the memory requirements. Details on the implementation and sample run times are given in section 8.2.

The large-s behaviour of the factorial series coefficients given in eq. (7.30) is problematic for numerical computation, since for large s the values of the $a_{j,s}$ change by several orders of magnitude with each iteration. It is therefore beneficial to combine the $a_{j,s}$ with the

Γ-function in the series to

$$\bar{a}_{j,s} = \frac{a_{j,s}}{\Gamma(x_{\max} + s - K + 1)} \,. \tag{7.49}$$

The $\bar{a}_{j,s}$ then slowly converge to zero. Their dependence on $x_{\max}$ is not a problem, since $x_{\max}$ is typically chosen identical for all integrals of a loop order and for difference equations of order 1 the factorial series only need to be evaluated for a single value of x. The recurrence relations for the $\bar{a}_{j,s}$ pick up different polynomial factors from the Γ-function for each offset, but are otherwise unchanged.

It is necessary to start at lower loop orders with several more orders in ϵ than one wants to obtain for the highest loop order. Loss of ϵ-orders can occur during iterations of either recurrence relations or difference equations. This typically happens for small s or x when the pivot element of the equation starts at order $\epsilon^i, i > 0$ and the leading order(s) of the expansion cancel(s). In some unlucky cases this can happen independently of the value of x or s and one thus loses orders in ϵ in each iteration, which prevents obtaining any result. Whenever this happened for the fully massive tadpoles, it could be remedied by simply changing to a different master basis M_Z or M'_Z for the zone in question. Another possible issue that can be circumvented by changing the master basis M_Z is division by zero for small values of x. This can sometimes occur for some difference equations where the pivot element vanishes for specific values of x (typically 1 or 2) and the argument of the integral can thus not be pushed down further via the difference equation.

8 Implementation details

The core of my work on the massive 5-loop tadpoles is the program *TIDE* (Tadpole Integrals via Difference Equations), which I built to automate the methods and ideas described in sections 2 and 4 through 7. Starting from the scalar integrals defined in section 2.1, it implements all the steps needed to arrive at numerical ϵ-expansions for these integrals, including momentum shifts, reduction to master integrals, the various reduction algorithms for difference equations and recurrence relations, the translation process between these two kinds of equations, initial conditions for the factorial series via lower loop results and the numerical evaluation of the series. For the moment some of these steps are limited to the class of fully massive vacuum integrals with a single mass scale.

TIDE consists of roughly 40,000 lines of code and is written entirely in C++ . The only job that is done externally is the rational algebra during the reductions, which is handled by Fermat [120]. For internal algebraic manipulation I use the library GiNaC [121], and its underlying numerics library CLN [122] is used for the arbitrary precision calculations in the factorial series expansion. Equations for integrals and factorial series coefficients are stored in SQLite-databases [123]. All computations for this thesis have been performed on 5 machines with 24 cores (2.80 GHz, 12MB cache) and 48GB RAM each.

In the remainder of this section I present some of the technical details of the implementation in *TIDE* as well as some optimisation methods which improve performance without really changing the underlying ideas presented in the previous sections. I furthermore list some runtimes for several parts of the program and different reductions to show the influence of these optimisations and the scaling of the complexity of the system with the number of propagators of integrals.

8.1 Reductions

TIDE implements a total of four different reduction algorithms described earlier in this thesis:

- Laporta's algorithm for the reduction to master integrals (see sect. 2.4).
- Laporta's algorithm for the reduction to difference equations (see sect. 5.1).
- The Minimal Order Reduction Algorithm (MORA, see sect. 5.2).
- Algorithm 3, a modified version of the MORA that also minimises the x-degree of equations (see sect. 6.2).

The first algorithm works with ordinary integrals, while the other three reduce either integrals with one symbolic exponent x or factorial series coefficients $a_{j,s}$, where shifting equations in x or s is an additional freedom in the reduction. The implementation of Laporta's algorithm for difference equations in *TIDE* is now only maintained for comparative purposes, for all production runs it has been replaced by the MORA.

Difference equations and recurrence relations are obtained zone by zone, where each zone only requires its subzones to already be reduced. As a consequence all zones with the same number of propagators can be reduced independently and simultaneously, which provides a first trivial layer of parallelisation. The order of steps for a zone Z performed by *TIDE* is given by

1. Generate equations with seed integrals from Z (see sect. 8.1.2 for details).
2. Reduce the generated equations with algorithm 3 to equations with $(R, N) = (0, 0)$, $(1, 0)$, $(0, 1)$ or $(1, 1)$.
3. Reduce the output of algorithm 3 to coupled difference equations with the MORA. It is also possible to start the reduction from the initial IBP-relations, but algorithm 3 has to be run either way and its output is much closer to the final form of the MORA than the IBP-relations.
4. Translate the output of algorithm 3 to recurrence relations as described in section 6.1.
5. Reduce the initial recurrence relations with the MORA. In some cases it is possible to speed up the process and use less memory by first reducing the recurrence relation with algorithm 3 and then with the MORA, as for the difference equations.

In the following I focus on several aspects of the implementation that are needed for reductions or used to improve them. Unless otherwise stated, all methods and optimisations presented are implemented for all four kinds of reduction and valid for integrals with or without a symbolic exponent x.

8.1.1 Momentum shifts

Any time an integral outside the classification scheme is encountered, its loop momenta need to be shifted such that all propagators in the denominator have the form $q_i^2 + 1$ with $q_i \in A_L$. If the original integral has t propagators of the form $d_i^2 + 1$ in the denominator, this is only possible for a set of t momenta q_i that can be mapped to the same graph as the d_i. The brute force approach would be to consider every graph with t lines, take its representative sector with subset $S \subset A_L$ in the classification scheme and try to find a

shift of the loop momenta k_i such that

$$(d_1, \ldots, d_t) \longrightarrow (+q_{i_1}, \pm q_{i_2}, \ldots, \pm q_{i_t}) \quad \text{with } q_{i_j} \in S. \tag{8.1}$$

In the worst case (if the target sector has no symmetries) this would take as many as $t!2^{t-1}$ attempts per sector due to the possible permutations and signs of the q_i. To improve upon these combinatorics, the approach is supplemented with information about the diagrams in *TIDE*. A diagram for the original integral is constructed by first attaching all lines to a single vertex and then splitting vertices repeatedly while maintaining momentum conservation until the diagram has the correct number of loops. An example for such a construction is given in figure 8.1. The diagram is then used to identify the correct sector and constrain the number of permutations of the target q_i by comparing numbers of lines on the vertices and their neighbouring vertices with those of the diagrams of the target sector. In some cases there are distinct diagrams that can be constructed from the same list of momenta (see e.g. figure 8.2), but this can be accounted for by preparing a list of all possible diagrams for each of the representative sectors in advance.

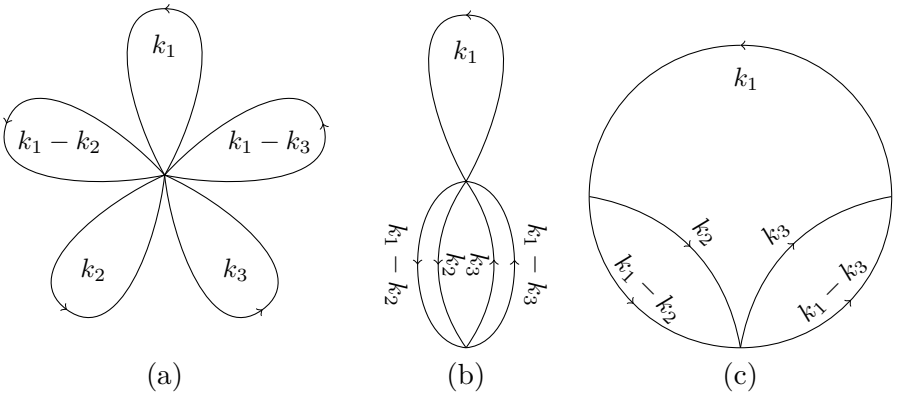


Figure 8.1: Construction of a 3-loop diagram for the momenta k_1, k_2, k_3, $k_1 - k_2$ and $k_1 - k_3$. (a) is the initial diagram with one vertex, (b) after splitting the vertex once, (c) the final diagram where no vertices can be split without breaking momentum conservation.

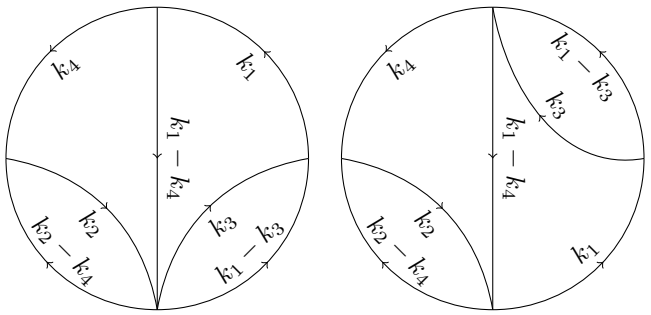


Figure 8.2: Two distinct graphs for the same list of propagator momenta.

Once all integrals are expressed in the classification scheme defined by A_L, momentum transformations are limited to the sector and symmetry shifts introduced in section 2.2. In *TIDE* these are applied during the generation of equations. All integrals that can appear in the equations are identified in a prior step and each one of them is expressed in the simplest integrals possible according to the integral ordering relation. To do so, the program goes through the following steps for each integral I:

1. Check if the solution for this integral is already in the database. If so, stop.
2. If the sector of I is not one of the representative sectors, perform a sector shift to the corresponding representative sector. If there is more than one possible shift for this task (due to symmetries of the target sector), pick the one that will transform the propagator D_i with the highest exponent $|z_i|$ in the numerator to the smallest number of terms (these shifts are predetermined for each sector-D_i combination). Go to step 5.
3. For each possible symmetry shift of the sector, determine the most difficult integral J in the expression that I would be shifted to. This can be done without performing the full shift.
   (a) If J is simpler than I, perform the shift and go to step 5.
   (b) If $J = I$ but the coefficient of J in $I = cJ + \dots$ is $c \neq 1$, perform the shift and solve for I. Go to step 5.
4. The integral cannot be simplified by momentum shifts. Add this information to the database and stop.
5. In general the shift will have produced a sum of integrals. For each integral in the expression, start this process at step 1 and insert the solution when finished.
6. I is now expressed by integrals which cannot be simplified further via momentum shifts. Add this solution to the database.

8.1.2 Generation of equations and the r-s-boundaries

A reduction in *TIDE* requires the parameters $r_{\max}$, $s_{\max}$, r_{gen} and s_{gen}, with $r_{\max} \leq r_{\text{gen}}$ and $s_{\max} \leq s_{\text{gen}}$. Equations generated may then contain integrals with $(r, s) \leq (r_{\text{gen}}, s_{\text{gen}})$, but results will only be stored for integrals with $(r, s) \leq (r_{\max}, s_{\max})$. In the following I will for simplicity refer to the latter group as *wanted integrals* and to integrals with $(r, s) \not\leq (r_{\max}, s_{\max})$ as *unwanted integrals*. The inclusion of unwanted integrals is meant to prevent loss of information at the r-s-boundaries of the reduction, where $r = r_{\max}$ or $s = s_{\max}$. Empirically the system usually contains sufficient information to reduce

all integrals used as seed integrals, but IBP-relations generally increase both r and s of their seed integrals and the resulting integrals may not be fully reduced. To be able to use integrals at the r-s-boundaries as seed integrals, it is thus necessary to allow integrals with higher r and s to be generated. In most cases choosing $r_{\text{gen}} = r_{\text{max}} + 1$ and $s_{\text{gen}} = s_{\text{max}} + 1$ is sufficient to fully reduce all wanted integrals. To obtain the difference equations of the 5-loop tadpole integrals I have chosen $r_{\text{max}} = s_{\text{max}} = 2$ and $r_{\text{gen}} = s_{\text{gen}} = 3$.

To ensure that the unwanted integrals are always extracted first by the reduction algorithms, the ordering relation is changed such that they are considered more difficult than wanted integrals. Equations containing unwanted integrals are not stored, since they are usually not fully reduced, but only used to extract information about the wanted integrals. This extraction of information only requires a relatively small number of equations and steps to actually be considered in the reduction (see sect. 8.1.4).

TIDE uses three types of equations (see sect. 2.3):

- IBP-relations with the operators $O_{ij} = \frac{\partial}{\partial k_i^\mu} k_j^\mu$.
- Linear combinations of IBP-relations which do not increase s. These are generated for each sector and always include the mass derivative relation introduced in section 2.3.3. They are only used on integals with $s = s_{\text{gen}}$, since they are otherwise contained in the ordinary IBP-relations, which usually consist of fewer terms.
- Syzygy relations which do not increase r (see sect. 2.3.2). These are only used on seed integrals with $r = r_{\text{gen}}$, since they are otherwise contained in the ordinary IBP-relations, which usually consist of much fewer terms.

The latter two types can also provide additional information at the r-s-boundaries, but this is usually not sufficient to completely relinquish the introduction of r_{gen} and s_{gen}. For all three types of equations the program first generates the equations for a general seed integral $I(z_1, \ldots, z_w)$ with symbolic exponents z_i. When inserting the integer values z_i for a specific seed integral, all integrals are immediately replaced by their fully symmetrised expressions as described in section 8.1.1. Before the reduction starts, the equations are generated and sorted by their most difficult integral, span and number of terms. If memory usage is a problem, the equations are not kept during this process and generated again one by one during the reduction, otherwise the full system of equations is handed over to the reduction algorithm.

8.1.3 Rational algebra and parallelisation

For all of the more difficult reductions the bulk of the computation time is spent on the algebra of the rational coefficients in the equations. To speed up the process this task is delegated to the external[16] program Fermat [120], since it is much faster for rational algebra than the internally used library GiNaC[17]. Each reduction can make use of multiple Fermat instances which work on single coefficients independently in order to provide a low-level layer of parallelisation.

To communicate with the Fermat executables, the main thread running the reduction algorithm sets up an instance of a class called `fermat_module`. This class spawns a predetermined number of communication threads, each of which starts an instance of the external Fermat executable and sets up a pipeline to it. The main thread can deposit expressions (e.g. sums or products of coefficients of a single integral) that need to be evaluated into a queue which is part of the `fermat_module`. The expressions are then distributed between the communication threads, which send them to the Fermat instances. When the results come back, the communication threads automatically send a query to the queue for the next expression. The structure of this process is illustrated in figure 8.3.

Since the Fermat instances are external executables, the communication cannot be implemented with specialised data-structures for rational functions, but relies on using strings for the expressions. For this purpose *TIDE* has a lightweight wrapper class `ex_string`, which stores expressions as a string, but still allows the reduction thread to perform basic algebraic manipulations on the expressions by implementing them on the string level. During the reduction all coefficients in the system of equations are stored as `ex_string`s. To save memory, the class has a trivial built-in compression function, which stores two consecutive characters of an expression as a single character. This is possible because for a maximum of two variables x and d or s and d the number of allowed combinations of two consecutive symbols in Fermat output is less than the 256 possible configurations for one byte. More sophisticated compression algorithms might yield better compression rates, but this simple setup is certainly the fastest way to compress these particular strings. The decompression and compression of the expressions is performed by the communication threads before and after they are sent to the Fermat instances.

To improve the concurrency of the layer of parallelisation provided by running multiple Fermat instances, expressions are usually sorted by size before they are added to the queue of the `fermat_module`, such that the largest expressions are evaluated first. This is necessary because in many situations during the reductions all expressions in the queue have to

[16] Very recently a library version of Fermat has been published, but it has not yet been implemented into *TIDE*.

[17] See e.g. reference [60] for a comparison of Fermat and GiNaC.

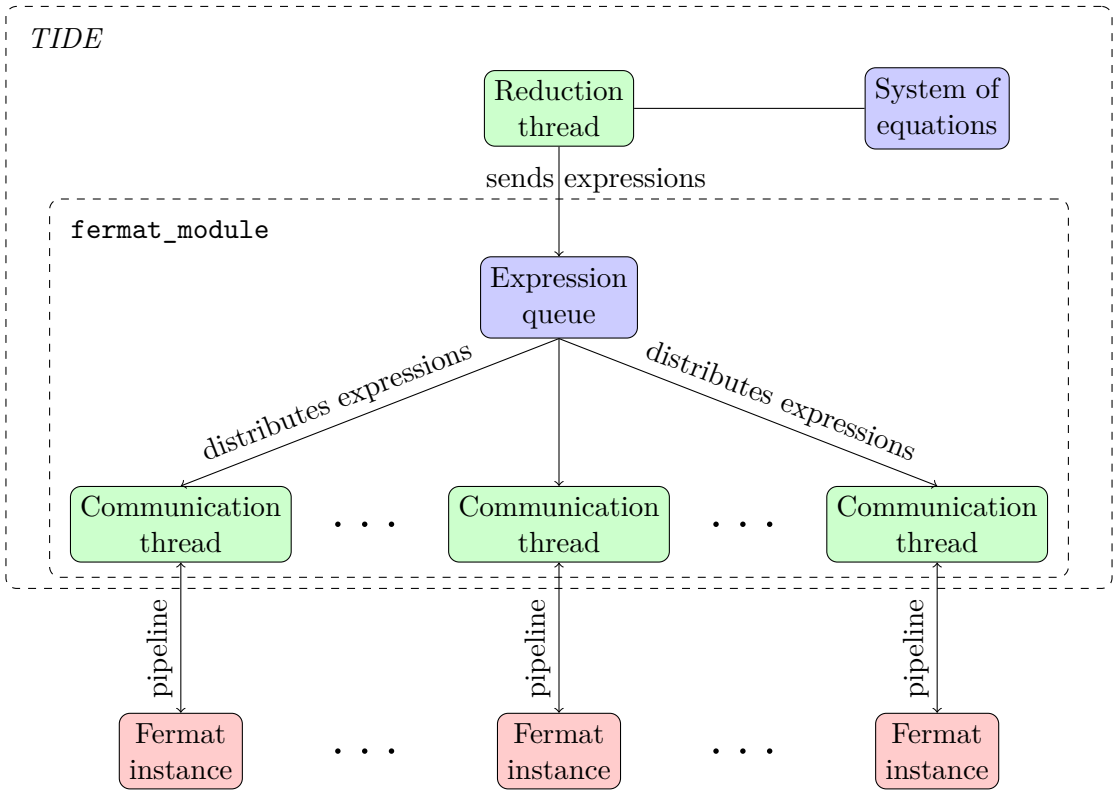


Figure 8.3: Implementation of low-level parallelisation with multiple Fermat instances.

be evaluated before the next ones can be added, e.g. when successively eliminating terms in one equation with several other equations. Within one equation the size of coefficients can vary quite drastically, especially when reducing zones/sectors with many subzones/-subsectors. Starting the evaluation with the most complicated coefficients reduces the amount of time where only a fraction of the Fermat instances is busy and the reduction thread is waiting for the queue to be cleared. The situation can be further improved by adding another layer of parallelisation that allows to perform several linear combinations of equations simultaneously (see sect. 8.1.4).

8.1.4 Numerical runs and delayed subsector execution

For any reduction *TIDE* first executes one or more numerical runs[18], in which the dimension is replaced by a prime number and all calculations are performed over a prime field for the integers in the coefficients, but the system is otherwise reduced the same way as in a real run. The integral exponent x or the factorial series parameter s cannot be replaced, because the reduction algorithms rely on shifts in those variables. A numerical run is usually at least an order of magnitude faster than a full run with all variables, since the rational algebra only contains at most a single variable and integers do not grow beyond the prime chosen for the finite field. Rational algebra over a finite field is already a feature of Fermat, making the implementation in *TIDE* trivial.

In theory it is possible to reconstruct the full rational function content of the final reduced system of equations by performing many numerical runs with different prime numbers [62]. While this has not been implemented in *TIDE* or any public code, even a single numerical run can provide valuable information. The most notable use is the identification of redundant equations [7, 63]. Since the system of IBP-relations is usually vastly overdetermined except around the r-s-boundaries, most equations typically do not contribute any information to the final reduced system. By identifying these equations in a fast numerical run, the full reduction with all variables can be performed starting from a minimal set of equations, which often results in a significant speed-up. There is a small risk that some equations are only thrown away by the numerical run because some coefficients during the reduction are zero for the particular chosen values of d and the finite field prime p, but not in general. This risk of losing information in this way, however, is usually negligible due to the overdeterminedness of the system and can be further reduced by choosing sufficiently large[19] d and p or performing multiple numerical runs.

Distinguishing relevant and redundant equations is especially useful in conjunction with unwanted integrals, which are only introduced to prevent loss of information at the r-s-boundaries. Since equations containing these integrals are not stored anyway, they do not have to be produced in the first place, unless needed to obtain equations for wanted integrals. By keeping track of which of the initial equations contribute to which of the final equations, *TIDE* is able to separate the minimal set of equations needed to reduce all wanted integrals. The full reduction then usually only requires only a small subset of the equations containing unwanted integrals.

For sectors with a high number of propagators, the bulk of the computational effort is not

[18] In this context numerical refers to the fact that the dimension is replaced by a number, not the use of floating point numbers.

[19] Prime numbers with 4 digits were sufficient in all reductions performed for this thesis.

spent on the reduction of the sectors themselves, but on the reduction of subsectors[20]. In order to have more control over this computationally expensive part of the reduction, the execution of the reduction is delayed in *TIDE* for all subsectors. The reduction algorithm is first performed only within the chosen sector S, with all coefficients of subsector integrals being left untouched, but all steps performed being recorded. These steps can then afterwards be performed also for the subsectors, along with the necessary steps to reduce the subsector content of the equations for integrals in S. The advantage of this delayed execution is that once the reduction of the new sector S is complete, all necessary steps for the reduction of the subsector parts of the equations are known in advance, since typically the subsectors are already fully reduced and no new information is found for them. In some rare cases it can happen that the reduction of sector S does indeed provide new information for a subsector S' which cannot be found from using seed integrals of that sector (see e.g. [124, 125]), but this can be checked efficiently with the numerical run.

The steps for the subsector parts of the equations can be divided into two categories; steps performed during the reduction of sector S (main steps) and steps where pivot equations of subsectors are used to eliminate terms from the equations (sub-pivot steps). The fact that the main steps are known in advance and the pivot equations for the sub-pivot steps do not change opens up two avenues for optimisation. The first optimisation is the introduction of another layer of parallelisation. Many steps can be performed simultaneously, which improves the concurrency of the parallelisation introduced by multiple Fermat instances by increasing the number of expressions that are queued for evaluation before the main thread has to wait for the queue to be emptied. This is difficult to achieve if subsectors are treated in the same way as the main sector S, since the reduction algorithm usually cannot tell whether two steps A and B can be performed simultaneously, since A might lead to a change in pivot equations that might affect B.

The second optimisation only works for equations that can be shifted in a variable like x or s. If such shifts occur in the main steps, they can reintroduce integrals in the subzones that have already been removed previously. For the MORA, this includes any integrals or factorial series coefficients with offset $k \neq 0$, since the subzones of any equations with terms in the main zone can always be reduced to terms with offset 0. Any time a shifted version of an equation is added to another equation in the main steps, terms with non-zero offsets are introduced. If the subzones were reduced at the same time as the main zone, these terms would often be removed immediately only to be reintroduced later by a different step. In the delayed execution of the subzone in *TIDE* these redundant steps can be avoided, since all main steps are known in advance.

[20] For integrals with a symbolic exponent x it would be more precise to speak of zones and subzones, but here the more general term sector is used, because it is also valid when reducing ordinary integrals with only numerical exponents.

		Conf. 1	Conf. 2	Conf. 3	Conf. 4
Numerical run		no	yes	no	yes
Delayed subzone		no	no	yes	yes
Numerical run	# equations	-	15630	-	15630
	Steps	-	$2.4 \cdot 10^6$	-	$6.7 \cdot 10^5$
	Time	-	983s	-	111s
Full reduction	# equations	15630	462	15630	462
	Steps	$2.4 \cdot 10^6$	$3.7 \cdot 10^4$	$6.9 \cdot 10^5$	$3.4 \cdot 10^4$
	Time	2671s	46s	276s	17s
Combined time		2671s	1029s	276s	128s

Table 8.1: Reduction of 5-loop zone 31246#6 with the MORA using different configurations of whether or not numerical runs and delayed subzone execution are applied. The parameters of the reduction are $r_{\max} = r_{\text{gen}} = s_{\max} = 2$, $s_{\text{gen}} = 3$.

Table 8.1 displays the effects of a numerical run and the delayed subzone execution for the reduction of zone 31246#6 with the MORA. In this example only 462 of the 15630 generated equations ($\sim 3\%$) are needed to fully reduce all wanted integrals of the zone. The numerical run provides an efficient method of identifying these equations, which speeds up the full reduction with all variables considerably. Zone 31246#6 only has 8 propagators and order 5 and is thus still comparatively easy to reduce, which is why the numerical run in this example takes up more time than the full run with the reduced number of equations. For more difficult zones with larger rational coefficients this is typically not the case any more.

8.1.5 Choosing a master basis

In combination with the MORA, numerical runs are also useful to determine a good master basis M_Z for a given zone Z. Once the system is in a reduced form, the ordering relation of integrals can be changed and upon running the MORA again a new master basis will emerge. In this way many different bases can be tested without needing to start from the initial equations each time. In principle this process could also be performed on the full system including all variables, but then the benefit of smaller coefficients would only be gained in reductions of superzones of Z. Using the reduced system of the numerical run is much faster and also allows to pick an optimised master basis already for the full reduction of Z. Since d is replaced in the numerical run and the integers are capped by the finite field, the only measure for the complexity of the coefficients in the equations for a particular master basis is the x-degree of the coefficients (or the s-degree in the reduction of factorial series coefficients), but experience shows that this reflects the overall complexity of the full coefficients quite well.

In a typical reduction at the 5-loop level there are of the order of a few hundred integrals in a given zone up to $r_{\max} = s_{\max} = 2$ after symmetrisation. If the order of the zone and thus the size of the master basis is greater than 2, testing all possible master bases becomes infeasible. Instead *TIDE* uses a method similar to simulated annealing described by the following steps:

1. Start with a set B of different master bases obtained from pre-defined ordering relations and test their quality by determining the average x-degree of the coefficients in the equations.
2. Pick n bases in B with the smallest x-degrees (where n is a small integer parameter). For each of these bases build all possible bases which differ only by one integral and insert them into a new set B' unless they have already been tested.
3. Test the quality of all bases in B'.
4. If the lowest x-degree for bases in B' is not lower than the one of the best basis in B, stop and use the best basis in B.
5. Set $B = B'$ and go back to step 2.

This process is not guaranteed to find the global minimum, but is still able to significantly decrease the complexity of the coefficients. For zones with orders $R > 10$ the size of the coefficients is often decreased by an order of magnitude compared to those of the master basis of the original ordering relation. The structure of the optimised basis in terms of parameters r or s varies a lot for different zones, making it impossible to formulate a single ordering relation that produces near-optimal master bases for all zones.

8.1.6 Timings

In this section I list some examples for how much time the various steps and algorithms used in this thesis take to obtain difference equations and recurrence relations. The zones for the examples are at the 5-loop level, since *TIDE* is able to obtain all difference equations and recurrence relations for the 4-loop massive tadpole integrals in less than an hour total. The reductions listed here are for difference equations. Reductions for recurrence relations generally behave similarly, but are somewhat slower, since their orders are higher on average.

Zone	t	Numerical run		Full run		Time for $(x-\alpha)$-factorisation	Total time
		# eqs.	Time	# eqs.	Time		
31744#1	5	$2.6 \cdot 10^4$	1s	32	$<$ 1s	$<$ 1s	1s
32256#1	6	$6.1 \cdot 10^4$	6s	93	$<$ 1s	$<$ 1s	6s
28686#1	6	$7.4 \cdot 10^4$	24s	201	1s	$<$ 1s	25s
30858#2	7	$2.0 \cdot 10^5$	36s	789	10s	1s	47s
29703#3	7	$2.6 \cdot 10^5$	51s	934	11s	1s	63s
30876#1	8	$3.5 \cdot 10^5$	61s	244	3s	2s	66s
31246#1	8	$5.6 \cdot 10^5$	343s	2183	266s	58s	667s
32518#1	9	$1.1 \cdot 10^6$	384s	434	8s	3s	395s
30231#1	9	$1.2 \cdot 10^6$	2442s	6162	2637s	370s	5449s
32674#2	10	$1.4 \cdot 10^6$	938s	106	8s	13s	959s
32279#1	10	$2.1 \cdot 10^6$	0.27d	25255	0.27d	5020s	0.59d
32682#1	11	$4.4 \cdot 10^6$	3355s	2604	184s	25s	3564s
32744#1	11	$3.2 \cdot 10^6$	1.26d	50891	1.56d	7988s	2.91d

Table 8.2: Runtimes for the reduction of several 5-loop zones with algorithm 3 and parameters $r_{\max} = s_{\max} = 2$, $r_{\text{gen}} = s_{\text{gen}} = 3$. For each number of propagators t (except 5) one of the most simple and one of the most difficult zones are shown here.

As a first step the 5-loop zones are reduced with algorithm 3. Table 8.2 lists timings for this reduction for several zones with parameter $r_{\max} = s_{\max} = 2$ and $r_{\text{gen}} = s_{\text{gen}} = 3$, where for each number of propagators one of the most simple and one of the most difficult zones were picked. The table shows that each additional propagator can increase the time for the reduction by roughly one order of magnitude, such that each of the more difficult zones takes significantly longer than all 4-loop zones combined. There are also vast differences between zones with the same number of propagators, making it very difficult to predict how much time is needed for a given reduction. No examples with the maximum of 12 propagators at the 5-loop level are shown here, since their reductions could not yet be performed due to RAM-limitations.

The output of algorithm 3 is subsequently reduced with the MORA to obtain the final difference equations. Since the input of the algorithm is already in a reduced form, this

Zone	t	R	Numerical run		Full run			Total time
			# eqs.	Time	# eqs.	Time main zone	Time subzones	
31744#1	5	1	35	$<$ 1s	31	$<$ 1s	$<$ 1s	$<$ 1s
32256#1	6	2	97	$<$ 1s	89	$<$ 1s	$<$ 1s	$<$ 1s
28686#1	6	4	123	$<$ 1s	108	$<$ 1s	$<$ 1s	$<$ 1s
30858#2	7	2	421	1s	400	1s	1s	3s
29703#3	7	8	445	1s	396	3s	1s	5s
30876#1	8	2	200	1s	183	1s	1s	3s
31246#1	8	11	748	6s	670	24s	26s	56s
32518#1	9	2	193	$<$ 1s	184	$<$ 1s	2s	2s
30231#1	9	20	838	30s	729	980s	5594s	6604s
32674#2	10	2	111	1s	98	1s	3s	5s
32279#1	10	19	1294	32s	1155	119s	0.15d	0.15d
32682#1	11	2	111	1s	98	1s	7s	9s
32744#1	11	9	1070	4s	991	9s	0.53d	0.53d

Table 8.3: Runtimes and orders R for the reduction of the 5-loop zones in table 8.2 with the MORA and the output of algorithm 3 as the initial equations. $r_{\max} = s_{\max} = 2$.

is much faster than starting from IBP-relations. Table 8.3 lists the runtimes for these reductions for the same zones as in table 8.2. Due to the preprocessed input the zones with small orders take virtually no time to reduce even for high numbers of propagators, while in zones with higher orders the complexity of the coefficients and the increased number of steps in both the main zone and subzones still have a big influence on the runtime of the reduction.

8.2 Numerical evaluation

The high precision needed during the numerical evaluation to overcome the divergence factors poses a challenge to both memory and CPU resources. As already mentioned in section 7.5, storing the factorial series coefficients on disk is no longer an option and as a consequence all integrals of one loop-order have to be calculated simultaneously. Since the recurrence relations are very interdependent, this should be done on a single machine to avoid slowing down the process by requiring network communication. On the CPU side extensive parallelisation and other optimisations are required to keep computation times at a reasonable size.

The top recurrence relations in equation (6.26) and the bottom difference equations in equation (5.3) are ideally suited for parallelisation, since the $a_{j,s+1}$ or the $I_j(x-1)$ can be obtained simultaneously from the $a_{j,s}$ or $I_j(x)$ for all j. In *TIDE* this is implemented in the same way as the parallelisation for the rational algebra. When all $a_{j,s}$ are known, the computations of the factorial series coefficients $a_{1,s+1}, a_{2,s+1}, \ldots$ are added to a queue. A

number of worker threads then starts computing one coefficient each at a time until the queue is emptied. To improve the concurrency the queue is sorted from long recurrence relations to short recurrence relations. The push-down of the values of x works in the same way with the difference equations, but takes less time since typically $x_{\max} \ll s_{\max}$. In the rare case that the master basis M'_Z of the factorial series coefficients is smaller than the basis M_Z of the integrals, additional series coefficients have to be added to M'_Z to find values for all integrals. The recurrence relations for these additional coefficients originally have order 0, which *TIDE* then converts into equations of order 1 to be able to fit them into the same evaluation pattern.

Beyond parallelisation there is not much that can be done to accelerate the additions, multiplications and divisions of floating point numbers in the ϵ-expansions of integrals and factorial series coefficients. The evaluation of the rational functions of s in the recurrence relations or x in the difference equations, however, can still be improved. These functions have to be evaluated for many values of s or x and depending on the chosen precision and the complexity of the functions this can take as much as, or even more, time than the floating point operations. There are various options for the handling of denominators which influence the size and amount of polynomials that have to be evaluated, ranging from one denominator per term to a common denominator for all terms in an equation. The former option has smaller polynomials due to cancellations of factors, but more denominators and division operations, while the latter has the fewest possible polynomials, but they are larger on average. In *TIDE* I have opted for a middle path in which there is one common denominator per zone. This is faster than the previous two options, since terms in one zone typically share many factors and thus most of the possible cancellations occur, while the number of denominators is still considerably lower than the number of terms in the equation.

To evaluate the polynomials, I have implemented Horner's method [126], in which a polynomial $p(x)$ is written as

$$\begin{aligned} p(x) &= \alpha_0 + \alpha_1 x + \dots \alpha_n x^n \\ &= ((\dots((\alpha_n x + \alpha_{n-1})x + \alpha_{n-2})x + \dots)x + \alpha_1)x + \alpha_0 \ . \end{aligned} \tag{8.2}$$

In the second form p can be evaluated faster for a value of x than in the monomial form, since the number of multiplications is reduced to at most n. Furthermore cancellations of large terms are less likely to happen, which decreases the numerical error if the evaluation is not done analytically[21].

[21] Whether analytical or numerical evaluation of the polynomials is faster depends on the numerical precision and how it compares to the size of the integers that appear during the evaluation. In the calculations performed for this thesis analytical evaluation was usually the faster choice.

With these optimisations the numerical evaluation of most 5-loop fully massive tadpoles up to 11 propagators takes roughly a week running on 24 cores. The parameters for this evaluation are $D_{start} = 20000$, $x_{\max} = 600$, $s_{\max} = 17000$ with ~ 30 orders in ϵ and it yields about 280 digits precision for the results. The lower loop results needed as input have higher precision but fewer integrals and simpler equations (see table 7.1 for the parameters used for $L \leq 4$). The time needed for their evaluation is of the same order of magnitude as for 5 loops. Increasing the precision further might still be possible, but would be expensive in CPU-time, since the time needed for floating point multiplication goes as precision squared and additionally the parameters $s_{\max}$ and $x_{\max}$ would have to be increased.

In some cases it can be necessary to change the master basis M_Z or M'_Z of a zone Z for the numerical evaluation due to losses of ϵ-orders or division by zero for small values of x (see sect. 7.5). An appropriate basis is then found in a similar fashion as during the master basis optimisation after the numerical reduction run (sect. 8.1.5). Elements in the basis are replaced one by one and among those bases which do not have problems with ϵ-orders or division by zero the one with the smallest equations is chosen. To avoid updating all equations in superzones of Z to the new basis, all elements in the old basis which are not elements of the new basis are also included in the evaluation, even if they are no longer needed in the zone Z itself. This slightly increases the number of integrals or factorial series coefficients, but updating superzone equations would typically introduce more complicated coefficients and thus have a similar effect on the runtime.

During the evaluation it is important to identify numerical zeroes whenever they occur. If left untreated, they can quickly decrease the precision of other terms with each iteration of the difference equations or recurrence relations. Since the $\bar{a}_{j,s} = a_{j,s}/\Gamma(x+s+d/2+1)$ cover a large range of orders of magnitude, implementing a hard cutoff for identifying numerical zeroes is difficult. Instead, *TIDE* compares each order in an ϵ-expansion to its neighbouring orders and sets it to zero if it is many orders of magnitude smaller than them (e.g. by a factor $2^{50} \approx 10^{15}$ or more).

9 The fully massive tadpoles up to 5 loops

9.1 Classification

To ensure that all integrals of a loop order are considered in the classification scheme, one first has to generate all possible graphs. For vacuum integrals up to 5 loops this has been done e.g. in [127]. As seen in section 8.1.1, it is possible for some sets of momenta to be mapped to distinct graphs, rendering those graphs equivalent in the sense that they represent the same integrals. The set of equivalent graphs and their associated integrals is referred to as a *topology*. In some topologies the integrals can be factorised into lower loop integrals, which is equivalent to the graph being split into disconnected lower loop graphs that all satisfy momentum conservation. The number of vacuum topologies up to 5 loops including factorised topologies is given by [7]:

- *1-loop*: $1 = 1$ non-factorised topology + 0 factorised topologies
- *2-loop*: $2 = 1$ non-factorised topology + 1 factorised topology (1-loop squared) $\equiv 1\{2\} + 1\{1^2\}$
- *3-loop*: $5 = 3\{3\} + 1\{1^3\} + 1\{1 \cdot 2\}$
- *4-loop*: $16 = 10\{4\} + 1\{1^4\} + 1\{1^2 \cdot 2\} + 1\{2^2\} + 3\{1 \cdot 3\}$
- *5-loop*: $67 = 48\{5\} + 1\{1^5\} + 1\{1^3 \cdot 2\} + 1\{1 \cdot 2^2\} + 3\{1^2 \cdot 3\} + 3\{2 \cdot 3\} + 10\{1 \cdot 4\}$

All of these topologies are covered by the classification schemes defined by the momentum lists A_L given in table A.1 for $L = 1, \ldots, 5$. For each topology a possible graph and the representative sector in the classification scheme are listed in figures A.1 and A.2. At a given loop-order L it is not strictly necessary to obtain values for master integrals in factorised topologies, since these are already known from lower loop-orders. Nevertheless these topologies appear during reductions as sub-topologies (defined in analogy with subsectors) of non-factorised topologies and therefore cannot simply be discarded. The simplest approach is to treat factorised topologies in the same way as non-factorised topologies and re-derive all of their results known from lower loops for L loops. Since lower loops are typically much simpler, the effect of re-deriving the results on the overall computation time is negligible.

L	$w(L) = \frac{L(L+1)}{2}$	Topologies	Physical sectors	Zero-sectors	Anti-sectors
1	1	1	1	1	0
2	3	2	4	4	0
3	6	5	38	26	0
4	10	16	680	281	63
5	15	67	22051	5566	5151

Table 9.1: Numbers of physical, zero- and anti-sectors up to 5 loops.

The $2^{w(L)}$ sectors defined by the list of momenta A_L at each loop level are divided into physical, zero- and anti-sectors and their respective numbers are listed in table 9.1. The representative sectors of the topologies have 1, 2, 5, 19 and 131 master integrals at 1, ..., 5 loops respectively[4, 6], including those of factorised topologies. If factorised topologies are excluded, these numbers change to 1, 1, 3, 13 and 109. In each sector there is one master integral with $r = s = 0$ (meaning $z_i \in \{0, 1\}$), which is in the following called the *corner integral* of the sector. Starting at 4 loops some of the sectors need more than one master integral. For ease of computation with Laporta's method I choose these additional master integrals such that $s = 0$ and all dots are on a single line, but other choices are equally valid. A full list of master integrals can be found in tables B.1, B.2 and B.3.

9.2 Results

Using the program *TIDE* I was able to obtain many new results for the fully massive 5-loop tadpole integrals. This includes all difference equations and almost all recurrence relations and numerical results for integrals up to 11 propagators. For the integrals with the highest possible number of propagators, which is 12, I obtained some of the difference equations, but no recurrence relations yet. Further progress is currently limited by the size of the systems of equations for the reductions of these complicated zones, which no longer fit into the 48GB RAM of the machines I have at my disposal. I feel it is likely that *TIDE* could obtain the remaining equations on machines with more memory or if the program was tweaked to further minimise memory usage. A third, albeit presumably much slower, option would be to (partly) store equations on hard disk during the reduction. I aim to look into these options in future work, but the main goal of this thesis, evaluating the tadpoles needed for the 5-loop β-function, is already met. This is due to the fact that the remaining integrals with 12 propagators are all finite, as can be seen from dimensional arguments and the fact that they do not have subdivergences, and thus do not contribute to anomalous dimensions such as the β-function (see e.g. [3]).

Some of the simpler difference equations for the 5-loop fully massive tadpoles have already

been found in [6] and [7], but already at 8 propagators there are sectors for which all zones are missing in these references. The main reason *TIDE* is now able to obtain these difference equations and many more is the transition to maximally coupled equations and the MORA. Due to the size of the difference equations I do not print them in this thesis, but both they and the recurrence relations can be made available upon request. A list of orders of the 5-loop zones for difference equations and recurrence relations as far as they are known is given in tables B.6 and B.7, respectively, including which of the equations have already been obtained by *TIDE*.

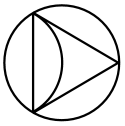
Using the coupled difference equations and recurrence relations I was able to find numerical solutions for the master integrals of all sectors with up to 11 propagators, which include 44 of the 48 non-factorised 5-loop topologies. The solutions have been obtained as ϵ-expansions around both $d = 4 - 2\epsilon$ and $d = 3 - 2\epsilon$ with 15-30 orders in ϵ and precisions of at least 250 digits. Some of these results are listed in appendix C. For compactness only the first few orders of the ϵ-expansions around $d = 4$ of the corner integrals of each sector are included there. The full results including all master integrals in either 3 or 4 dimensions can be made available upon request.

All results in appendix C and the remainder of this section are normalised with $1/J^L$, where J is the 1-loop massive tadpole given by

$$J = \int_{k_1} \frac{1}{k_1^2 + 1} = \Gamma\left(1 - \frac{d}{2}\right) . \tag{9.1}$$

This normalisation makes the results independent of the chosen integration measure and thus simplifies comparison with results from other sources. It also naturally removes the dependence on the Euler-Mascheroni constant γ_E, which otherwise appears in the ϵ-expansions and is removed in many conventions by including $e^{\gamma_E \epsilon}$ in the measure. The downside of the normalisation with J^{-L} is the inconsistent representation of poles in ϵ, since J is finite in odd dimensions, but has a $1/\epsilon$ pole in even dimensions. As a consequence one finds e.g. for the corner integral I_{29703} of sector 29703

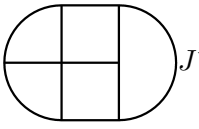
$$I_{29703} = I(1,1,1,0,1,0,0,0,0,0,0,0,1,1,1) = \tag{9.2}$$



$$
\begin{aligned}
I_{29703}J^{-5}|_{d=4-2\epsilon} = \; & 2.20000000000000000000000000000000000000000000000000 \\
& + 4.48333333333333333333333333333333333333333333333333\epsilon \\
& + 4.56666666666666666666666666666666666666666666666666\epsilon^2 \\
& - 9.38870242568726971371952869072060998337697532621 2\epsilon^3 \\
& + 19.31407800749752756112276689558356534059581172145\epsilon^4 \\
& + 307.30415883558477683765391851606087923555677280 66\epsilon^5 \\
& + \dots \, ,
\end{aligned}
$$

$$
\begin{aligned}
I_{29703}J^{-5}|_{d=3-2\epsilon} = \; & 0.12499999999999999999999999999999999999999999999999\epsilon^{-2} \\
& + 1.23212823204218206109850140978021260086015918506 8\epsilon^{-1} \\
& - 1.95981953847570612368495051611554560353147871358 0 \\
& + 138.72455406943621226498824017048445557458913339 07\epsilon \\
& + 184.98518871075293433092186763132022142203796727 24\epsilon^2 \\
& + \dots \, .
\end{aligned}
$$

Here the 3-dimensional result shows the leading ϵ^{-2} pole explicitly, while in 4 dimensions the terms up to ϵ^4 are actually divergent in I_{29703}. Finite integrals such as



$$
\begin{aligned}
J^{-5} = I_{32744}J^{-5} \overset{d=4-2\epsilon}{=} \; & - 2.99823287368040309416876288359009577308 7\epsilon^5 \\
& + 43.75280252615427411169257824938492717855\epsilon^6 \\
& + \dots
\end{aligned}
$$

thus start at ϵ^L for even dimensions.

Ideally one would like to obtain analytical expressions for the integrals or at least the orders in their ϵ-expansions. Many of the divergent orders of the integrals are trivially identified as ζ-functions or rational numbers, but for the finite orders more complicated functions like elliptic integrals are expected to appear [4]. In principle these could be identified by algorithms such as PSLQ [112, 113], which attempt to find integer relations between a given set of numerical expressions. In order to apply this approach one needs a basis of numbers which can appear at a given order in ϵ and sufficient precision for PSLQ to identify beyond doubt the integrals as linear combinations of these basis elements. At the moment such a basis is not known for the 5-loop fully massive tadpoles (or even all lower loop tadpoles) and I expect that even for the first few finite orders in ϵ several thousand digits of precision would be needed.

9.3 Checks

Since *TIDE* implements several new ideas, it is important to perform cross-checks on its results. As a first sanity check I compared the values generated for the 4-loop master integrals to the ones obtained in [4] (4d) and [5] (3d) and found full agreement. For the 5-loop integrals only a handful of results were already known, but a simple internal check is already built into Laporta's method if one compares the values for the corner integral of a sector obtained from its different zones. With the exception of sectors 28686, 30876 and 30526, results for the corner integrals up to 11 propagators have been generated from at least two different zones and are in agreement for 250 digits of precision or more in each case. In sectors 28686 (with $t = 6$ propagators) and 30876 ($t = 8$) all positions for the exponent x are equivalent, while for sector 30526 ($t = 11$) the recurrence relations for 4 of its 5 zones are still missing. For the first two sectors some results are already known in the literature and are listed below.

Potential errors in the results of zones 28686#1 or 30876#1 would have also likely been detected internally. The values obtained in these zones show up in the inhomogeneous parts of their respective superzone equations, which depend on these values in different ways, yet produce consistent results. More generally, the hierarchical structure of the approach and the exponential divergence of numerical errors make it highly unlikely for most mistakes to go unnoticed. Any error usually quickly produces terms which are many orders of magnitude larger than expected due to the divergence factors and these terms then spread to all superzones of the zone where the error first occurred.

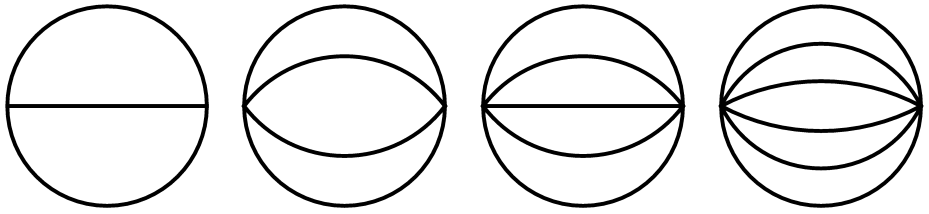


Figure 9.1: The sunset topologies with $L = 2, 3, 4, 5$ loops.

While the internal checks on the 5-loop results already rule out many sources of errors, external checks are still needed to ensure that *TIDE* does not contain systematic errors which might affect all integrals equally in a way that would not be detected by cross-checks between zones of a sector. The simplest candidate for such a check is the 5-loop sunset topology, represented in my classification scheme by sector 28686. The sunset topologies (see fig. 9.1) consist of $L + 1$ lines between two vertices and can be calculated (at least numerically) for any dimension and any number of loops using coordinate-space techniques

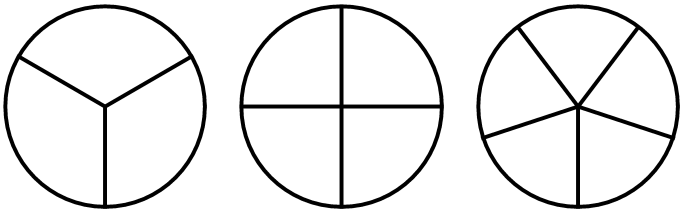
[128, 129]. For the 5-loop sunset corner integral S_5 in four dimensions this yields [129]

$$\begin{aligned} S_5 J^{-5} = &-3 - \frac{3}{2}\epsilon + \frac{13}{24}\epsilon^2 - \frac{1267}{1440}\epsilon^3 - \frac{4193}{3456}\epsilon^4 \\ &+ 135.951\epsilon^5 + 2603.82\epsilon^6 + 17828.\epsilon^7 + 180410.\epsilon^8 + \mathcal{O}\left(\epsilon^9\right) , \end{aligned} \tag{9.3}$$

which *TIDE* reproduces with 250 digits of precision. To also test the program in different dimensions, I have generated the ϵ-expansions for the sunset topology corner integrals for 2-5 loops and integer dimensions between 2 and 8. The expansions agree with known results and the leading orders are listed in table 9.2.

$d+2\epsilon$	$S_2 J^{-2}$	$S_3 J^{-3}$	$S_4 J^{-4}$	$S_5 J^{-5}$
2	$2.34391\epsilon^2$	$8.41439\epsilon^3$	$39.9455\epsilon^4$	$233.508\epsilon^5$
3	$\frac{1}{4\epsilon}$	$\frac{1}{\epsilon}$	$\frac{45}{16\epsilon}$	$\frac{7}{\epsilon}$
4	$-\frac{3}{2}$	-2	$-\frac{5}{2}$	-3
5	$\frac{9}{32\epsilon}$	$\frac{24}{35\epsilon}$	$\frac{52893}{71680\epsilon}$	$-\frac{26118}{25025\epsilon}$
6	$-\frac{5}{2}$	$-\frac{8}{3}$	$-\frac{5}{2}$	-2
7	$\frac{135}{512\epsilon}$	$\frac{480}{1001\epsilon}$	$\frac{408801375}{524812288\epsilon}$	$\frac{147950130}{52055003\epsilon}$
8	$-\frac{63}{20}$	$-\frac{29}{10}$	$-\frac{21}{8}$	$-\frac{27}{10}$

Table 9.2: The leading orders in the ϵ-expansion of the sunset topology corner integrals for 2-5 loops in various dimensions. Results have been obtained numerically, but agree to at least 250 digits precision where rational numbers are given.



W_3 W_4 W_5

Figure 9.2: Diagrams of the wheel with L spokes for $L = 3, 4, 5$.

Another external check is provided by the class of so-called wheel diagrams (see fig. 9.2), for which the leading order in ϵ in four dimensions has been known to all loop orders for some time [130]. For the wheel with L spokes one finds

$$W_L J^{-L} = \frac{(-1)^L}{L}\binom{2L-2}{L-1}\zeta(2L-3)\epsilon^{L-1} + \mathcal{O}\left(\epsilon^L\right) . \tag{9.4}$$

In my classification scheme W_5 is the corner integral of sector 32596 and the full result

obtained by *TIDE* is

$$\begin{aligned} W_5 J^{-5} = &- 14.116889883346919575757165697897154634398089847 91\epsilon^4 \\ &+ 235.077295967834671314543880809504117792393472395 8\epsilon^5 \\ &- 2267.738683293008412296299448058020585548754541373\epsilon^6 \\ &+ 17032.057367335504611145072601547913508995415150 54\epsilon^7 \\ &+ \dots \\ &+ 10998938855208.474461573750012265067520880599420 74\epsilon^{19} \\ &+ \mathcal{O}\left(\epsilon^{20}\right) , \end{aligned}$$

where the leading order agrees with the expected $-14\zeta(7)\epsilon^4$. A similar check is provided by the zig-zag-conjecture of Broadhurst and Kreimer [131], which was subsequently proven by Brown and Schnetz [132]. The conjecture states that the leading order of the L-loop integral Z_L of the series of zig-zag diagrams (see fig. 9.3) is given in 4 dimensions by

$$Z_L J^{-L} = \begin{cases} \frac{(-1)^L 4}{L^2}\binom{2L-2}{L-1}\zeta(2L-3)\epsilon^{L-1} + \mathcal{O}\left(\epsilon^L\right) & \text{if } L \text{ is even,} \\ \frac{(-1)^L(4-4^{3-L})}{L^2}\binom{2L-2}{L-1}\zeta(2L-3)\epsilon^{L-1} + \mathcal{O}\left(\epsilon^L\right) & \text{if } L \text{ is odd.} \end{cases} \tag{9.5}$$

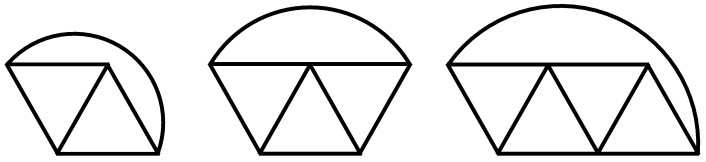


Figure 9.3: Zig-zag diagrams for $L = 3, 4, 5$.

For 3 and 4 loops the zig-zag- and wheel-series are identical, but at 5 loops the integrals differ, with Z_5 being the corner integral of sector 32279. The result produced by *TIDE* is

$$\begin{aligned} Z_5 J^{-5} = &- 11.117050783135699165908767987094009274588495755 23\epsilon^4 \\ &+ 181.782239286123408207907882360186429611987419942 0\epsilon^5 \\ &- 1725.999613740352080595167399244211728860770495754\epsilon^6 \\ &+ \dots , \end{aligned}$$

where the leading order agrees with the exact result $-\frac{441}{40}\zeta(7)\epsilon^4$.

In [31] Anderson et al. have calculated the leading orders of three fully massive 5-loop tadpoles in three dimensions for the pressure of massless ϕ^4-theory. Expressed in my

classification scheme their results are

$$\begin{aligned}
I(1,1,1,1,0,0,0,1,0,0,1,1,1,0,0)J^{-5} =& \frac{9\zeta(3)}{4} - \frac{\pi^2 \ln 2}{2} + \mathcal{O}(\epsilon) \text{ (sector 30876)},\\
I(2,1,1,1,0,0,0,1,0,0,0,1,0,1,0)J^{-5} =& -\frac{1}{32}\epsilon^{-2} + \frac{3+4\ln 2}{16}\epsilon^{-1}\\
&+ \frac{3-\pi^2-108\ln 2+72\ln 3-78\ln^2 2-60\mathrm{Li}_2\frac{1}{4}}{24}\\
&+ \mathcal{O}(\epsilon) \text{ (sector 30858)},\\
I(1,1,1,1,0,0,0,1,0,0,0,1,1,1,0)J^{-5} =& -0.44316 + \mathcal{O}(\epsilon) \text{ (sector 30862)}.
\end{aligned}$$

For sectors 30876 and 30858 *TIDE* reproduces these values, but for sector 30862 it finds

$$\begin{aligned}
I(1,1,1,1,0,0,0,1,0,0,0,1,1,1,0)J^{-5} =& -0.518821725792769087684129841769989253\\
&+ \mathcal{O}(\epsilon)\ .
\end{aligned}$$

The discrepancy originates in a sign error in equation (C.20) in [31], which is taken from [133] and should read

$$C = \frac{p^2+q^2-p\cdot q+4m^2}{m^2}\ , \tag{9.6}$$

where the sign in front of the dot product $p \cdot q$ has been corrected. With this change the calculation in [31] also yields $-0.51882\ldots$ for the leading order of the corner integral of sector 30862 [22].

The above checks cover two out of the three sectors for which *TIDE* has only determined results in a single zone, but for sector 30526 no results are available in the literature. An attempt to obtain the leading order of the corner integral I_{30526} by sector decomposition with Fiesta 3.4 [73] resulted in error notifications after several hours. However, since the integral is finite, a naive `NIntegrate` in Mathematica [110] over the Feynman parametrisation yields a numerical result, albeit with very low precision. The values found are

$$I_{30526}|_{d=4} = \begin{cases} 1.803 \pm 0.189 & \text{(Mathematica)},\\ 1.97039045531\ldots & (\textit{TIDE}), \end{cases} \tag{9.7}$$

$$I_{30526}J^{-5}|_{d=3} = \begin{cases} (-1.671 \pm 0.078)\cdot 10^{-4} & \text{(Mathematica)},\\ -1.69355884089\ldots\cdot 10^{-4} & (\textit{TIDE}). \end{cases} \tag{9.8}$$

As an additional check I ran the numerical evaluation in *TIDE* again with different master bases M_Z and M'_Z for the difference equations and recurrence relations and the results agree with the expected precision of 250 digits.

[22] Y. Schröder, personal communication.

10 Conclusion and outlook

In this thesis I have applied Laporta's method [1] for evaluating multi-loop Feynman integrals via difference equations and factorial series to the class of fully massive vacuum integrals with a single mass scale. To make the method feasible for the evaluation of the 5-loop integrals of this class I have made several improvements over its original version. The main improvements are:

- By using the new Minimal Order Reduction Algorithm (MORA, see sect. 5.2) instead of Laporta's algorithm, difference equations are obtained as R coupled equations of order one rather than one decoupled equation of order R. The coupled equations exhibit much simpler coefficients, especially for zones with high orders. As a result, the time needed for the reduction to difference equations scales significantly better for the MORA and for difficult zones it is many orders of magnitude faster than Laporta's algorithm. This boost in speed is needed due to the much increased complexity of the reductions at 5 loops compared to lower loops, which renders Laporta's algorithm infeasible for obtaining difference equations at 5 loops.
- An analysis of the process that translates difference equations to recurrence relations revealed that the order R' of a recurrence relation is at least the x-degree N of the difference equation it was generated from. Each order in x of the input equations of the translation can thus lead to a loss of information in the space of recurrence relations. To prevent this loss of information and high orders R', I designed algorithm 3 as a modified version of the MORA which reduces both order and x-degree of difference equations. The resulting equations can be used in the translation process without loss of information. This opens up the possibility of reducing the recurrence relations in the same way as the difference equations, which is not done at all in the original version of Laporta's method. The reduction yields much simpler recurrence relations than can be achieved by translating a decoupled difference equation and thus speeds up the numerical evaluation.
- During the iterations of recurrence relations or difference equations the numerical error is increased in each iteration by the divergence factors F_R or F_P, respectively. I have shown that these factors are independent of whether coupled or decoupled equations are used and expressed them in terms of the roots of the characteristic polynomial of the difference equation. Knowledge of the divergence factors can be used to determine ideal values for the parameters $s_{\max}$ and $x_{\max}$ as well as the needed precision for the initial conditions of the factorial series coefficients. By increasing this precision appropriately, the (numerical) divergence of the factorial series can be avoided for all fully massive tadpoles and possibly other classes of integrals. This completely removes the necessity to perform a Laplace transformation on the difference equations and solve the resulting differential equations for these integrals, which was Laporta's original suggestion in cases where $F_R > 1$.

I have implemented all necessary steps for Laporta's method with these optimisations in the C++ -program *TIDE* and have used it to obtain high-precision numerical results for all fully massive 5-loop tadpole master integrals up to 11 propagators in three and four dimensions. This includes all integrals needed for the 5-loop contribution to the QCD β-function which will be determined in future work. The remaining integrals with 12 propagators are within reach.

The results show that with the improvements introduced in this thesis Laporta's method is capable of solving 5-loop integrals with a single scale on current hardware. The natural next step is thus to apply the approach to the partly massive 5-loop tadpoles. This class of integrals has many additional applications described in section 1. Due to the additional freedom of having $m \in \{0, 1\}$ for each propagator the class has more integrals and fewer symmetries than the fully massive tadpoles, but from experiences at the 4-loop level [4, 115] it is expected to have a similar complexity for the difference and recurrence relations. All steps of Laporta's method and my improvements implemented in *TIDE* are applicable to partly massive tadpoles without further adjustments, with the exception of the large-x expansion of the integrals for the initial conditions of the factorial series coefficients. This expansion is slightly more complicated than in the fully massive case due to the additional infrared divergences that may appear, but this case is already described in Laporta's original paper [1]. In principle the method can also be applied to other classes of integrals with a single scale, e.g. massless propagator type integrals or massive propagators with an on-shell external momentum.

While 5-loop calculations with Laporta's method are now feasible, full sets of 6-loop integrals still seem out of reach. I have not attempted to obtain any of the 6-loop fully massive tadpoles, but the jump in complexity from 4 to 5 loops suggest that many further improvements would be needed to successfully do so. Not only is it likely that the orders of difference equations and recurrence relations would be much higher than $R = 20$ and $R' = 28$ already found in this thesis, but the 5-loop results would have to be obtained with much higher precision than the current 250 digits, as they would be needed as initial conditions at 6 loops. Since the recurrence divergence factor $F_R \approx 13$ at 5 loops is the first one that is much larger than one, achieving the needed precision would prove considerably more difficult than at lower loops. It might however be possible to extract subsets of the 6-loop integrals, since the difficulty of sectors and zones varies drastically within one loop-level. One such possible subset could be the integrals needed for ϕ^4 theory. At 5 loops the corresponding zones have low order difference equations and recurrence relations and a small recurrence divergence factor, which means they could be obtained to much higher precision than many other 5-loop integrals. If the same holds true at 6 loops, the pressure of hot ϕ^4 theory (see sect. 1.2) could be calculated to this order.

One of the major challenges of Laporta's method going forward are large values for the divergence factors. This is especially true for the recurrence divergence factor, since typically $s_{\max} \gg x_{\max}$ and F_R jumped from 1.125 at 4 loops to 12.93 at 5 loops. While in principle the numerical divergence of the factorial series due to the exponential growth of the error can be avoided by increasing the precision, this approach is wasteful in the sense that most of the starting precision is lost. A preferable solution would be to somehow project out the divergent solutions in the numerical evaluation. If it is certain that no roots of the characteristic polynomial with absolute value greater than one contribute to the exact solution of a recurrence relation, the numerical error can be approximately identified when the factorial series coefficients start to diverge. This could be used to go back to earlier iterations of the series coefficients and reduce the numerical error there. The backtracking this includes would add additional computational effort, but the added cost is linear in $s_{\max}$, whereas increasing the precision to keep the error small scales the computation time with $\mathcal{O}\left({s_{\max}}^2\right)$ due to multiplications. An even better solution would be to determine the factorial series coefficients analytically in closed form, but to the best of my knowledge no technique exists as of yet which is capable of doing so for arbitrary linear recurrence relations with rational coefficients.

Reductions of difference equations and recurrence relations could also be improved further. An interesting step in that direction would be to allow arbitrary linear combinations of integrals as elements of the master bases M_Z and M'_Z instead of just single integrals. This could have the potential to significantly reduce the complexity of coefficients in the equations further, but a strategy would have to be developed to determine optimised basis elements. Another promising avenue is the reduction over a finite field and subsequent reconstruction of the rational coefficients proposed in [62]. This approach should circumvent the expression swell during intermediate steps of the reduction, but has not yet been implemented in any public codes. It is only partly suited for the Minimal Order Reduction Algorithm, as the algorithms reliance on shifts in x means that coefficients will still be rational functions in x at all times. However, Algorithm 3 would be a good candidate for the finite field method despite also using x-shifts, since all coefficients are at most linear in x at any given time during the reduction.

The improvements to Laporta's method of difference equations and factorial series suggested and implemented in this thesis have already pushed the limits of feasibility of the method to the 5-loop level by increasing its speed by many orders of magnitude. The above ideas show that there is still much potential for further development and the implementation of some of these ideas might make 6-loop calculations with this method a realistic option in the future.

Appendix

A Tadpole classification

	A_1	A_2	A_3	A_4	A_5
$q_1 =$	k_1	k_1	k_1	k_1	k_1
$q_2 =$		k_2	k_2	k_2	k_2
$q_3 =$		$k_1 - k_2$	k_3	k_3	k_3
$q_4 =$			$k_1 - k_2$	k_4	k_4
$q_5 =$			$k_1 - k_3$	$k_1 - k_4$	k_5
$q_6 =$			$k_2 - k_3$	$k_2 - k_4$	$k_1 - k_3$
$q_7 =$				$k_3 - k_4$	$k_1 - k_4$
$q_8 =$				$k_1 - k_2$	$k_1 - k_5$
$q_9 =$				$k_1 - k_3$	$k_2 - k_3$
$q_{10} =$				$k_1 - k_2 - k_3$	$k_2 - k_4$
$q_{11} =$					$k_2 - k_5$
$q_{12} =$					$k_3 - k_5$
$q_{13} =$					$k_4 - k_5$
$q_{14} =$					$k_1 + k_2 - k_4$
$q_{15} =$					$k_3 - k_4$

Table A.1: Sets A_L of propagator momenta q_i chosen for the classification of tadpole integrals up to $L = 5$ loops.

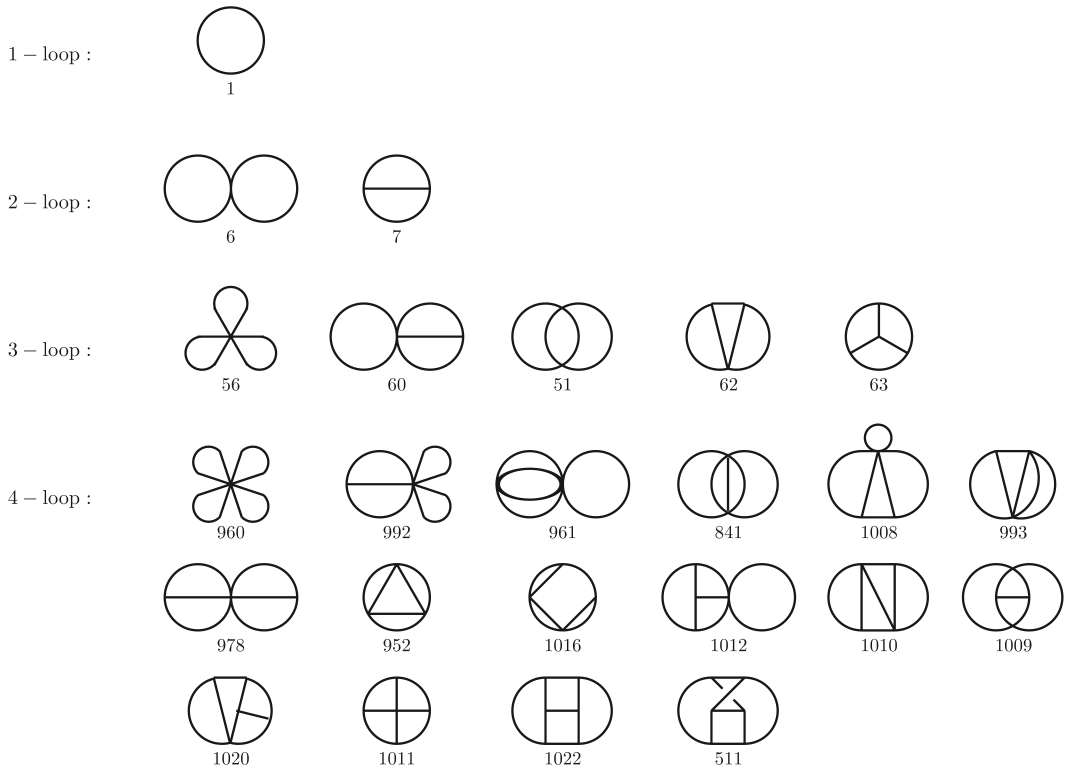


Figure A.1: All topologies for vacuum integrals up to 4 loops with the sector IDs of the respective representative sectors, including factorised topologies [6, 134, 127].

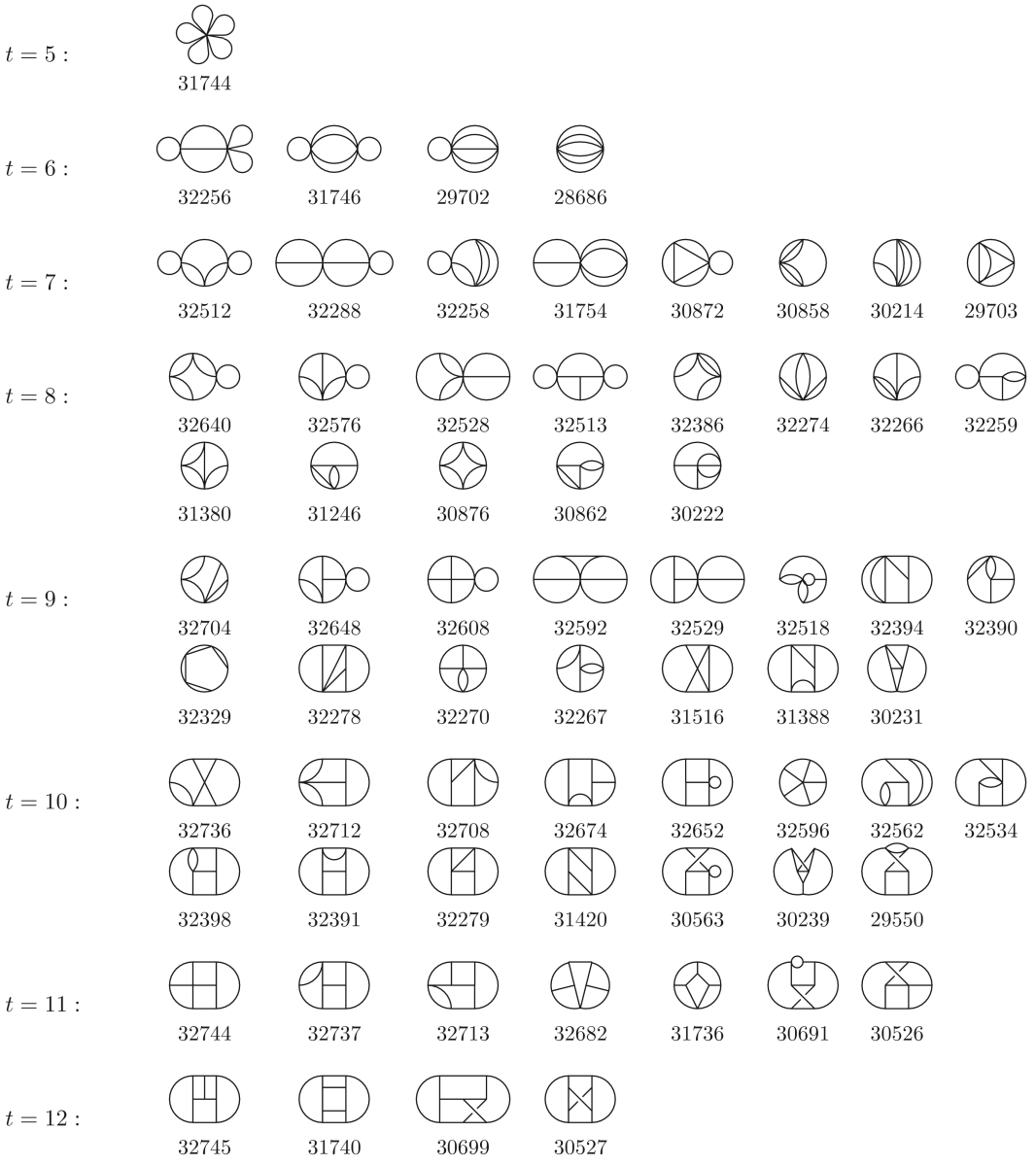


Figure A.2: All 67 topologies for 5-loop vacuum integrals with the sector IDs of the respective representative sectors, including factorised topologies [6, 134, 127].

B Reduction statistics

1-loop			
#	t	sID	Master Integral
1	1	1	I_1

2-loop			
#	t	sID	Master Integral
1	2	6*	$I_{1,1,0}$
2	3	7	$I_{1,1,1}$

3-loop			
#	t	sID	Master Integral
1	3	56*	$I_{1,1,1,0,0,0}$
2	4	60*	$I_{1,1,1,1,0,0}$
3	4	51	$I_{1,1,0,0,1,1}$
4	5	62	$I_{1,1,1,1,1,0}$
5	6	63	$I_{1,1,1,1,1,1}$

4-loop			
#	t	sID	Master Integral
1	4	960*	$I_{1,1,1,1,0,0,0,0,0,0}$
2	5	992*	$I_{1,1,1,1,1,0,0,0,0,0}$
3	5	961*	$I_{1,1,1,1,0,0,0,0,0,1}$
4	5	841	$I_{1,1,0,1,0,0,1,0,0,1}$
5	6	1008*	$I_{1,1,1,1,1,1,0,0,0,0}$
6	6	993	$I_{1,1,1,1,1,0,0,0,0,1}$
7	6	978*	$I_{1,1,1,1,0,1,0,0,1,0}$
8	6	952	$I_{1,1,1,0,1,1,1,0,0,0}$
9	7	1016	$I_{1,1,1,1,1,1,1,0,0,0}$
10	7	1012*	$I_{1,1,1,1,1,1,0,1,0,0}$
11	7	1010	$I_{1,1,1,1,1,1,0,0,1,0}$
12	7	1009	$I_{1,1,1,1,1,1,0,0,0,1}$
13	8	1020	$I_{1,1,1,1,1,1,1,1,0,0}$
14	8	1011	$I_{1,1,1,1,1,1,0,0,1,1}$
15	9	1022	$I_{1,1,1,1,1,1,1,1,1,0}$
16	9	511	$I_{0,1,1,1,1,1,1,1,1,1}$
17	5	841	$I_{\mathbf{3},1,0,1,0,0,1,0,0,1}$
18	7	1009	$I_{\mathbf{2},1,1,1,1,1,0,0,0,1}$
19	8	1011	$I_{\mathbf{2},1,1,1,1,1,0,0,1,1}$

Table B.1: List of all chosen master integrals for 1-4 loops. The notation is $I_{z_1,\ldots,z_w} \equiv I(z_1,\ldots,z_w)$. Sector IDs marked with a star correspond to factorised topologies.

#	t	sID	Master integral
1	5	31744*	$I_{1,1,1,1,1,0,0,0,0,0,0,0,0,0}$
2	6	32256*	$I_{1,1,1,1,1,1,0,0,0,0,0,0,0,0}$
3	6	31746*	$I_{1,1,1,1,1,0,0,0,0,0,0,0,0,1,0}$
4	6	29702*	$I_{1,1,1,0,1,0,0,0,0,0,0,0,1,1,0}$
5	6	28686	$I_{1,1,1,0,0,0,0,0,0,0,0,1,1,1,0}$
6	7	32512*	$I_{1,1,1,1,1,1,1,0,0,0,0,0,0,0,0}$
7	7	32288*	$I_{1,1,1,1,1,1,0,0,0,1,0,0,0,0,0}$
8	7	32258*	$I_{1,1,1,1,1,1,0,0,0,0,0,0,0,1,0}$
9	7	31754*	$I_{1,1,1,1,1,0,0,0,0,0,0,1,0,1,0}$
10	7	30872*	$I_{1,1,1,1,0,0,0,1,0,0,1,1,0,0,0}$
11	7	30858	$I_{1,1,1,1,0,0,0,1,0,0,0,1,0,1,0}$
12	7	30214	$I_{1,1,1,0,1,1,0,0,0,0,0,0,1,1,0}$
13	7	29703	$I_{1,1,1,0,1,0,0,0,0,0,0,0,1,1,1}$
14	8	32640*	$I_{1,1,1,1,1,1,1,1,0,0,0,0,0,0,0}$
15	8	32576*	$I_{1,1,1,1,1,1,1,0,1,0,0,0,0,0,0}$
16	8	32528*	$I_{1,1,1,1,1,1,1,0,0,0,1,0,0,0,0}$
17	8	32513*	$I_{1,1,1,1,1,1,1,0,0,0,0,0,0,0,1}$
18	8	32386	$I_{1,1,1,1,1,1,0,1,0,0,0,0,0,1,0}$
19	8	32274	$I_{1,1,1,1,1,1,0,0,0,0,1,0,0,1,0}$
20	8	32266	$I_{1,1,1,1,1,1,0,0,0,0,0,1,0,1,0}$
21	8	32259*	$I_{1,1,1,1,1,1,0,0,0,0,0,0,0,1,1}$
22	8	31380	$I_{1,1,1,1,0,1,0,1,0,0,1,0,1,0,0}$
23	8	31246	$I_{1,1,1,1,0,1,0,0,0,0,0,1,1,1,0}$
24	8	30876	$I_{1,1,1,1,0,0,0,1,0,0,1,1,1,0,0}$
25	8	30862	$I_{1,1,1,1,0,0,0,1,0,0,0,1,1,1,0}$
26	8	30222	$I_{1,1,1,0,1,1,0,0,0,0,0,1,1,1,0}$
27	9	32704	$I_{1,1,1,1,1,1,1,1,1,0,0,0,0,0,0}$
28	9	32648*	$I_{1,1,1,1,1,1,1,1,0,0,0,1,0,0,0}$
29	9	32608*	$I_{1,1,1,1,1,1,1,0,1,1,0,0,0,0,0}$
30	9	32592	$I_{1,1,1,1,1,1,1,0,1,0,1,0,0,0,0}$
31	9	32529*	$I_{1,1,1,1,1,1,1,0,0,0,1,0,0,0,1}$
32	9	32518	$I_{1,1,1,1,1,1,1,0,0,0,0,0,0,1,1,0}$
33	9	32394	$I_{1,1,1,1,1,1,0,1,0,0,0,1,0,1,0}$
34	9	32390	$I_{1,1,1,1,1,1,0,1,0,0,0,0,1,1,0}$
35	9	32329	$I_{1,1,1,1,1,1,1,0,0,1,0,0,1,0,1}$
36	9	32278	$I_{1,1,1,1,1,1,1,0,0,0,0,0,1,1,0}$
37	9	32270	$I_{1,1,1,1,1,1,1,0,0,0,0,0,0,1,1}$
38	9	32267	$I_{1,1,1,1,1,1,1,0,0,0,0,0,0,1,1}$
39	9	31516	$I_{1,1,1,1,0,1,1,0,0,0,0,1,1,1,0}$
40	9	31388	$I_{1,1,1,1,0,1,0,1,0,0,1,1,1,0,0}$
41	9	30231	$I_{1,1,1,0,1,1,0,0,0,0,0,1,0,1,1}$
42	10	32736	$I_{1,1,1,1,1,1,1,1,1,1,1,0,0,0,0}$
43	10	32712	$I_{1,1,1,1,1,1,1,1,1,1,0,0,1,0,0}$
44	10	32708	$I_{1,1,1,1,1,1,1,1,1,1,0,0,0,1,0}$
45	10	32674	$I_{1,1,1,1,1,1,1,1,1,0,1,0,0,0,1}$
46	10	32652*	$I_{1,1,1,1,1,1,1,1,1,0,0,0,1,1,0}$
47	10	32596	$I_{1,1,1,1,1,1,1,1,0,1,0,1,0,1,0}$
48	10	32562	$I_{1,1,1,1,1,1,1,1,0,0,1,1,0,0,1}$
49	10	32534	$I_{1,1,1,1,1,1,1,1,0,0,0,1,0,1,1}$
50	10	32398	$I_{1,1,1,1,1,1,1,0,1,0,0,0,1,1,0}$
51	10	32391	$I_{1,1,1,1,1,1,1,0,1,0,0,0,0,1,1}$
52	10	32279	$I_{1,1,1,1,1,1,1,0,0,0,0,1,0,1,1}$
53	10	31420	$I_{1,1,1,1,0,1,0,1,0,1,1,1,1,0,0}$
54	10	30563*	$I_{1,1,1,0,1,1,1,0,1,1,0,0,0,1,1}$
55	10	30239	$I_{1,1,1,0,1,1,0,0,0,0,1,1,1,1,1}$
56	10	29550	$I_{1,1,1,0,0,1,1,0,1,1,0,1,1,1,0}$
57	11	32744	$I_{1,1,1,1,1,1,1,1,1,1,1,0,1,0,0}$
58	11	32737	$I_{1,1,1,1,1,1,1,1,1,1,0,0,0,0,1}$
59	11	32713	$I_{1,1,1,1,1,1,1,1,1,0,0,1,0,0,1}$
60	11	32682	$I_{1,1,1,1,1,1,1,1,0,1,0,1,0,1,0}$
61	11	31736	$I_{1,1,1,1,0,1,1,1,1,1,1,1,0,0,0}$
62	11	30691	$I_{1,1,1,0,1,1,1,1,1,1,0,0,0,1,1}$
63	11	30526	$I_{1,1,1,0,1,1,1,0,0,1,1,1,1,1,0}$
64	12	32745	$I_{1,1,1,1,1,1,1,1,1,1,1,0,1,0,1}$
65	12	31740	$I_{1,1,1,1,0,1,1,1,1,1,1,1,1,0,0}$
66	12	30699	$I_{1,1,1,0,1,1,1,1,1,1,1,0,1,0,1}$
67	12	30527	$I_{1,1,1,0,1,1,1,0,0,1,1,1,1,1,1}$

Table B.2: List of all chosen master integrals without dots at 5 loops. The notation is $I_{z_1,\ldots,z_w} \equiv I(z_1,\ldots,z_w)$. Master integrals with dots are found in table B.3. Sector IDs marked with a star correspond to factorised topologies.

#	t	sID	Master integral
68	6	29702*	$I_{\mathbf{3},1,1,0,1,0,0,0,0,0,0,0,1,1,0}$
69	6	28686	$I_{\mathbf{3},1,1,0,0,0,0,0,0,0,0,0,1,1,1,0}$
70	7	30214	$I_{\mathbf{2},1,1,0,1,1,0,0,0,0,0,0,1,1,0}$
71	7	30214	$I_{\mathbf{3},1,1,0,1,1,0,0,0,0,0,0,1,1,0}$
72	7	29703	$I_{\mathbf{2},1,1,0,1,0,0,0,0,0,0,0,1,1,1}$
73	7	29703	$I_{1,1,\mathbf{3},0,1,0,0,0,0,0,0,0,1,1,1}$
74	7	29703	$I_{\mathbf{3},1,1,0,1,0,0,0,0,0,0,0,1,1,1}$
75	8	32259*	$I_{\mathbf{2},1,1,1,1,1,0,0,0,0,0,0,0,1,1}$
76	8	31246	$I_{1,\mathbf{2},1,1,0,1,0,0,0,0,0,1,1,1,0}$
77	8	31246	$I_{\mathbf{2},1,1,1,0,1,0,0,0,0,0,1,1,1,0}$
78	8	31246	$I_{\mathbf{3},1,1,1,0,1,0,0,0,0,0,1,1,1,0}$
79	8	30862	$I_{\mathbf{2},1,1,1,0,0,0,1,0,0,0,1,1,1,0}$
80	8	30222	$I_{\mathbf{2},1,1,0,1,1,0,0,0,0,0,1,1,1,0}$
81	8	30222	$I_{1,\mathbf{3},1,0,1,1,0,0,0,0,0,1,1,1,0}$
82	8	30222	$I_{\mathbf{3},1,1,0,1,1,0,0,0,0,0,1,1,1,0}$
83	9	32608*	$I_{\mathbf{2},1,1,1,1,1,1,0,1,1,0,0,0,0,0}$
84	9	32390	$I_{\mathbf{2},1,1,1,1,1,0,1,0,0,0,0,1,1,0}$
85	9	32278	$I_{\mathbf{2},1,1,1,1,1,0,0,0,0,1,0,1,1,0}$
86	9	32270	$I_{1,1,\mathbf{2},1,1,1,0,0,0,0,0,1,1,1,0}$
87	9	32270	$I_{1,\mathbf{2},1,1,1,1,0,0,0,0,0,1,1,1,0}$
88	9	32270	$I_{\mathbf{2},1,1,1,1,1,0,0,0,0,0,1,1,1,0}$
89	9	32270	$I_{1,\mathbf{3},1,1,1,1,0,0,0,0,0,1,1,1,0}$
90	9	32270	$I_{\mathbf{3},1,1,1,1,1,0,0,0,0,0,1,1,1,0}$
91	9	32267	$I_{\mathbf{2},1,1,1,1,1,0,0,0,0,0,1,0,1,1}$
92	9	31516	$I_{1,1,\mathbf{2},1,0,1,1,0,0,0,1,1,1,0,0}$
93	9	31516	$I_{1,\mathbf{2},1,1,0,1,1,0,0,0,1,1,1,0,0}$
94	9	31516	$I_{\mathbf{2},1,1,1,0,1,1,0,0,0,1,1,1,0,0}$
95	9	31388	$I_{\mathbf{2},1,1,1,0,1,0,1,0,0,1,1,1,0,0}$
96	9	30231	$I_{\mathbf{2},1,1,0,1,1,0,0,0,0,1,0,1,1,1}$
97	9	30231	$I_{1,1,1,0,1,1,0,0,0,0,1,0,\mathbf{3},1,1}$
98	9	30231	$I_{1,1,1,0,1,1,0,0,0,0,1,0,\mathbf{4},1,1}$
99	9	30231	$I_{\mathbf{3},1,1,0,1,1,0,0,0,0,1,0,1,1,1}$
100	9	30231	$I_{\mathbf{4},1,1,0,1,1,0,0,0,0,1,0,1,1,1}$
101	10	32736	$I_{\mathbf{2},1,1,1,1,1,1,1,1,1,0,0,0,0,0}$
102	10	32596	$I_{\mathbf{2},1,1,1,1,1,1,0,1,0,1,0,1,0,0}$
103	10	32596	$I_{1,1,1,1,1,\mathbf{3},1,0,1,0,1,0,1,0,0}$
104	10	32596	$I_{\mathbf{4},1,1,1,1,1,1,0,1,0,1,0,1,0,0}$
105	10	32596	$I_{\mathbf{3},1,1,1,1,1,1,0,1,0,1,0,1,0,0}$
106	10	32534	$I_{\mathbf{2},1,1,1,1,1,1,0,0,0,1,0,1,1,0}$
107	10	32398	$I_{\mathbf{2},1,1,1,1,1,0,1,0,0,0,1,1,1,0}$
108	10	32391	$I_{\mathbf{2},1,1,1,1,1,0,1,0,0,0,0,1,1,1}$
109	10	32279	$I_{1,1,1,\mathbf{2},1,1,0,0,0,0,1,0,1,1,1}$
110	10	32279	$I_{\mathbf{2},1,1,1,1,1,0,0,0,0,1,0,1,1,1}$
111	10	32279	$I_{1,1,\mathbf{3},1,1,1,0,0,0,0,1,0,1,1,1}$
112	10	32279	$I_{\mathbf{3},1,1,1,1,1,0,0,0,0,1,0,1,1,1}$
113	10	30239	$I_{\mathbf{2},1,1,0,1,1,0,0,0,0,1,1,1,1,1}$
114	10	30239	$I_{1,1,1,0,1,1,0,0,0,0,1,1,1,\mathbf{3},1}$
115	10	30239	$I_{1,1,\mathbf{3},0,1,1,0,0,0,0,0,1,1,1,1,1}$
116	10	30239	$I_{\mathbf{3},1,1,0,1,1,0,0,0,0,0,1,1,1,1,1}$
117	10	29550	$I_{1,1,1,0,0,\mathbf{2},1,0,1,1,0,1,1,1,0}$
118	10	29550	$I_{\mathbf{2},1,1,0,0,1,1,0,1,1,0,1,1,1,0}$
119	10	29550	$I_{\mathbf{3},1,1,0,0,1,1,0,1,1,0,1,1,1,0}$
120	11	32744	$I_{\mathbf{2},1,1,1,1,1,1,1,1,1,0,1,0,0,0}$
121	11	31736	$I_{\mathbf{2},1,1,1,0,1,1,1,1,1,1,1,0,0,0}$
122	11	30526	$I_{1,1,\mathbf{2},0,1,1,1,0,0,1,1,1,1,1,0}$
123	11	30526	$I_{1,\mathbf{2},1,0,1,1,1,0,0,1,1,1,1,1,0}$
124	11	30526	$I_{\mathbf{2},1,1,0,1,1,1,0,0,1,1,1,1,1,0}$
125	11	30526	$I_{1,\mathbf{3},1,0,1,1,1,0,0,1,1,1,1,1,0}$
126	11	30526	$I_{\mathbf{3},1,1,0,1,1,1,0,0,1,1,1,1,1,0}$
127	12	31740	$I_{\mathbf{3},1,1,1,0,1,1,1,1,1,1,1,1,0,0}$
128	12	30527	$I_{\mathbf{2},1,1,0,1,1,1,0,0,1,1,1,1,1,1}$
129	12	30527	$I_{1,1,1,0,1,\mathbf{3},1,0,0,1,1,1,1,1,1}$
130	12	30527	$I_{\mathbf{3},1,1,0,1,1,1,0,0,1,1,1,1,1,1}$
131	12	30527	$I_{\mathbf{4},1,1,0,1,1,1,0,0,1,1,1,1,1,1}$

Table B.3: List of all chosen master integrals with dots at 5 loops. The notation is $I_{z_1,\ldots,z_w} \equiv I(z_1,\ldots,z_w)$. Master integrals without dots are found in table B.2. Sector IDs marked with a star correspond to factorised topologies.

1-loop				
#	t	sID	Position of x	Order R
1	1	1	1	1

2-loop				
#	t	sID	Position of x	Order R
1	2	6*	1	1
2	3	7	1	2

3-loop				
#	t	sID	Position of x	Order R
1	3	56*	1	1
2	4	60*	1,3	2,1
3	4	51	1	2
4	5	62	1,2	2,2
5	6	63	1	2

4-loop				
#	t	sID	Position of x	Order R
1	4	960*	1	1
2	5	992*	1,2	2,1
3	5	961*	1,4	2,1
4	5	841	1	4
5	6	1008*	1,3,4	2,1,2
6	6	993	1,2,4	5,2,2
7	6	978*	1	2
8	6	952	1	2
9	7	1016	1,4	2,2
10	7	1012*	1,3	2,1
11	7	1010	1,2,5	2,2,2
12	7	1009	1,3,4	3,3,3
13	8	1020	1,3,4,8	2,2,3,2
14	8	1011	1,3	4,3
15	9	1022	1,2	3,2
16	9	511	2	3

Table B.4: Difference equation orders for zones of the fully massive vacuum integrals at 1-4 loops. Sector IDs marked with a star correspond to factorised topologies.

1-loop				
#	t	sID	Position of x	Order R
1	1	1	1	1

2-loop				
#	t	sID	Position of x	Order R
1	2	6*	1	1
2	3	7	1	2

3-loop				
#	t	sID	Position of x	Order R
1	3	56*	1	1
2	4	60*	1,3	2,1
3	4	51	1	3
4	5	62	1,2	2,2
5	6	63	1	2

4-loop				
#	t	sID	Position of x	Order R
1	4	960*	1	1
2	5	992*	1,2	2,1
3	5	961*	1,4	3,1
4	5	841	1	5
5	6	1008*	1,3,4	2,1,2
6	6	993	1,2,4	5,4,2
7	6	978*	1	2
8	6	952	1	4
9	7	1016	1,4	2,2
10	7	1012*	1,3	2,1
11	7	1010	1,2,5	2,2,2
12	7	1009	1,3,4	5,5,3
13	8	1020	1,3,4,8	2,2,3,2
14	8	1011	1,3	4,6
15	9	1022	1,2	3,2
16	9	511	2	4

Table B.5: Recurrence relation orders for zones of the fully massive vacuum integrals at 1-4 loops. Sector IDs marked with a star correspond to factorised topologies.

#	t	sID	Position of x	Order R
1	5	31744*	1	1
2	6	32256*	1,2	2,1
3	6	31746*	1,3	2,1
4	6	29702*	1,3	4,2
5	6	28686	1	4
6	7	32512*	1,2,3	2,1,2
7	7	32288*	1,5	2,1
8	7	32258*	1,2,3,5	5,2,2,1
9	7	31754*	1,3	2,2
10	7	30872*	1,4	2,1
11	7	30858	1,2	4,2
12	7	30214	1,2,3	7,6,6
13	7	29703	1,3	7,8
14	8	32640*	1,2,3	2,1,2
15	8	32576*	1,2,5,6	2,2,1,2
16	8	32528*	1,2,3	2,2,2
17	8	32513*	1,2	2,1
18	8	32386	1,2,3	5,2,2
19	8	32274	1,3,4	7,2,2
20	8	32266	1,2,3,5,6	6,2,2,2,2
21	8	32259*	1,2,3,5	3,3,3,2
22	8	31380	1,2,3,8	7,2,2,2
23	8	31246	1,2,4,6	11,8,8,5
24	8	30876	1	2
25	8	30862	1,2	3,3
26	8	30222	1,2,3	8,7,8
27	9	32704	1,2,3,4,6	2,2,2,2,2
28	9	32648*	1,2,3,4,12	3,1,2,2,2
29	9	32608*	1,5,6	4,2,3
30	9	32592	1,3,4,6	2,2,2,2
31	9	32529*	1,2	2,2
32	9	32518	1,2	2,2
33	9	32394	1,2,3,12	5,2,2,2
34	9	32390	1,2,3,4,5,8,13	7,3,4,3,3,3,3
35	9	32329	1,3	2,2
36	9	32278	1,2,3,5,14	8,3,4,3,3
37	9	32270	1,2,3,6,12	12,10,15,9,7
38	9	32267	1,2,3,5	3,3,4,4
39	9	31516	1,2,3,6,12	6,6,11,5,5
40	9	31388	1,2,6	3,4,3
41	9	30231	1,13	20, 11
42	10	32736	1,2,3,5,6,9	6,4,4,4,3,3
43	10	32712	1,2,5,6,8	3,2,2,2,2
44	10	32708	1,2,3,4,6,13	3,2,2,2,2,2
45	10	32674	1,2,3,4	3,2,2,2
46	10	32652*	1,2,3	3,1,2
47	10	32596	1,6	18,12
48	10	32562	1,3,4	3,2,2
49	10	32534	1,2,3,4,11,13	8,4,4,4,3,3
50	10	32398	1,2,3,4,5,6	10,3,4,3,6,4
51	10	32391	1,2,3,6	6,4,4,4
52	10	32279	1,3,4,6,13,14	19,11,18,9,9,6
53	10	31420	1,6	3,3
54	10	30563*	1,5	3,1
55	10	30239	1,3,14	19,18,6
56	10	29550	1,6,12	10,10,7
57	11	32744	1,2,5,6,7,8,10	9,4,4,13,3,4,3
58	11	32737	1,2,3,5,6,9,15	3,2,3,2,2,2,3
59	11	32713	1,2,3,4,7	3,2,3,2,2
60	11	32682	1,2,3	2,2,2
61	11	31736	1,3,4,7	9,3,4,4
62	11	30691	1,2,3,5	5,3,3,2
63	11	30526	1,2,3,6,7	17,18,11,9,18
64	12	32745	1,2,3,7,8	3,2,3,2,2
65	12	31740	1	5
66	12	30699	1,2,5	7,3,2
67	12	30527	1,6	18,12

Table B.6: Difference equation orders for zones of the fully massive vacuum integrals at 5 loops. Sector IDs marked with a star correspond to factorised topologies. Zones with underlined orders have not been fully reduced yet.

#	t	sID	Position of x	Order R
1	5	31744*	1	1
2	6	32256*	1,2	2,1
3	6	31746*	1,3	3,1
4	6	29702*	1,3	5,2
5	6	28686	1	6
6	7	32512*	1,2,3	2,1,2
7	7	32288*	1,5	2,1
8	7	32258*	1,2,3,5	5,4,2,1
9	7	31754*	1,3	3,2
10	7	30872*	1,4	4,1
11	7	30858	1,2	5,4
12	7	30214	1,2,3	7,8,6
13	7	29703	1,3	9,13
14	8	32640*	1,2,3	2,1,2
15	8	32576*	1,2,5,6	2,2,1,2
16	8	32528*	1,2,3	2,2,2
17	8	32513*	1,2	2,1
18	8	32386	1,2,3	5,4,2
19	8	32274	1,3,4	7,2,4
20	8	32266	1,2,3,5,6	5,4,2,2,2
21	8	32259*	1,2,3,5	5,5,3,2
22	8	31380	1,2,3,8	7,4,2,4
23	8	31246	1,2,4,6	13,14,13,6
24	8	30876	1	4
25	8	30862	1,2	7,6
26	8	30222	1,2,3	10,10,8
27	9	32704	1,2,3,4,6	2,2,2,2,2
28	9	32648*	1,2,3,4,12	3,1,2,2,2
29	9	32608*	1,5,6	4,2,6
30	9	32592	1,3,4,6	2,2,2,2
31	9	32529*	1,2	2,2
32	9	32518	1,2	2,2
33	9	32394	1,2,3,12	5,4,2,2
34	9	32390	1,2,3,4,5,8,13	7,5,4,5,3,5,5

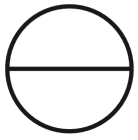
#	t	sID	Position x	Order R
35	9	32329	1,3	2,2
36	9	32278	1,2,3,5,14	8,6,4,3,5
37	9	32270	1,2,3,6,12	18,16,15,14,11
38	9	32267	1,2,3,5	5,5,4,4
39	9	31516	1,2,3,6,12	6,10,17,10,13
40	9	31388	1,2,6	5,7,3
41	9	30231	1,13	27, 19
42	10	32736	1,2,3,5,6,9	6,4,4,4,7,7
43	10	32712	1,2,5,6,8	3,2,2,2,2
44	10	32708	1,2,3,4,6,13	3,2,2,2,2,2
45	10	32674	1,2,3,4	3,2,2,2
46	10	32652*	1,2,3	3,1,2
47	10	32596	1,6	18,17
48	10	32562	1,3,4	3,2,2
49	10	32534	1,2,3,4,11,13	8,4,4,4,6,7
50	10	32398	1,2,3,4,5,6	8,5,4,5,6,4
51	10	32391	1,2,3,6	8,7,4,7
52	10	32279	1,3,4,6,13,14	28,11,20,16,16,16
53	10	31420	1,6	5,3
54	10	30563*	1,5	4,1
55	10	30239	1,3,14	25,19,8
56	10	29550	1,6,12	14,17,11
57	11	32744	1,2,5,6,7,8,10	9,4,4,12,7,?,6
58	11	32737	1,2,3,5,6,9,15	3,2,3,2,2,2,3
59	11	32713	1,2,3,4,7	3,2,3,2,2
60	11	32682	1,2,3	2,2,2
61	11	31736	1,3,4,7	?,5,4,?
62	11	30691	1,2,3,5	6,5,?,2
63	11	30526	1,2,3,6,7	?,?,?,14,?
64	12	32745	1,2,3,7,8	?,?,?,?,?
65	12	31740	1	?
66	12	30699	1,2,5	?,?,?
67	12	30527	1,6	?,?

Table B.7: Recurrence relation orders for zones of the fully massive vacuum integrals at 5 loops. Sector IDs marked with a star correspond to factorised topologies. Question marks denote orders which are not yet determined.

C Numerical results

Listed here are numerical results for the fully massive corner integrals of the representative sectors of non-factorised vacuum topologies in $d = 4 - 2\epsilon$ dimensions. The values given here are normalised by the 1-loop tadpole $J = \int_k (k^2+1)^{-1}$ to the power L and limited to $\sim$50 digits precision and the leading 6 orders of the ϵ-expansion. All results for 2-4 loops as well as a few integrals at 5 loops are already known in the literature (see sect. 9.3), but are listed here for completeness. The full list of results obtained in this work includes all 5-loop master integrals up to 11 propagators in three and four dimensions with 250 digits precision and 15-30 orders in ϵ. This list can be made available upon request.

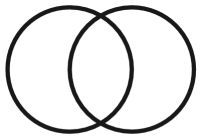
C.1 2-loop



7

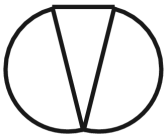
$$
\begin{aligned}
&-1.50000000000000000000000000000000000000000000000000 \\
&-1.50000000000000000000000000000000000000000000000000\ \epsilon \\
&+0.51586085803418833590234343330841560364310451445 3\ \epsilon^2 \\
&-8.54050333961454467179989499779211677236741385177 7\ \epsilon^3 \\
&+1.03920054145134562993799742840243756581445274504 4\ \epsilon^4 \\
&-34.02412109418437876206777042875976448646874597234\ \epsilon^5 \\
&+\dots
\end{aligned}
$$

C.2 3-loop



51

$$
\begin{aligned}
&-2.00000000000000000000000000000000000000000000000000 \\
&-1.66666666666666666666666666666666666666666666666666\ \epsilon \\
&-0.49999999999999999999999999999999999999999999999999\ \epsilon^2 \\
&+8.58333333333333333333333333333333333333333333333333\ \epsilon^3 \\
&+2.66487561537514667840977530357253367810717841928 8\ \epsilon^4 \\
&+196.73537825917304338585630537320304346641599253 26\ \epsilon^5 \\
&+\dots
\end{aligned}
$$



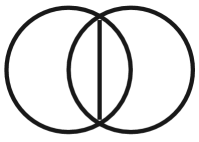
62

$$
\begin{aligned}
&\ 1.00000000000000000000000000000000000000000000000000 \\
&+2.66666666666666666666666666666666666666666666666666\ \epsilon \\
&+1.30161161726495666152864646671650212604712430442 5\ \epsilon^2 \\
&+16.17027687753648029273000749323203572619203425393\ \epsilon^3 \\
&+50.36368751002766464712459970887058692375220280352\ \epsilon^4 \\
&+72.00897461295336034290529765044698186018294035915\ \epsilon^5 \\
&+\dots
\end{aligned}
$$

63

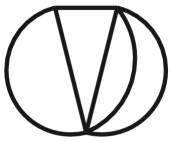
$$
\begin{aligned}
&-2.40411380631918857079947632302289998152997258468 0\,\epsilon^2\\
&+17.24761989872635488431312965422760018324025125004\,\epsilon^3\\
&-73.26296589040362104788617737106101541072605775907\,\epsilon^4\\
&+259.49466712225592469303538065882039393113752331 14\,\epsilon^5\\
&-855.06403242636831826849724618246316409251596833 37\,\epsilon^6\\
&+2715.94677645254438789344399190975665315592963937 2\,\epsilon^7\\
&+\ldots
\end{aligned}
$$

C.3 4-loop



841

$$
\begin{aligned}
&-2.49999999999999999999999999999999999999999999999\\
&-1.66666666666666666666666666666666666666666666666666\,\epsilon\\
&-0.03472222222222222222222222222222222222222222222222\,\epsilon^2\\
&-0.72337962962962962962962962962962962962962962962629\,\epsilon^3\\
&-16.14646921346643369526363642598358965658209440971\,\epsilon^4\\
&-302.68724953271679694373481692172522196605049592 72\,\epsilon^5\\
&+\ldots
\end{aligned}
$$

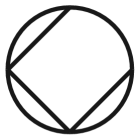


993

$$
\begin{aligned}
&\ 1.62499999999999999999999999999999999999999999999\\
&+3.72916666666666666666666666666666666666666666666666\,\epsilon\\
&+2.29912537961538416281315151670404326120200989 4985\,\epsilon^2\\
&+4.65030393687908158782191320505265565399724013 4599\,\epsilon^3\\
&-30.56788162137333591569301669838011480986789770322\,\epsilon^4\\
&-117.91221911012862079145748806272193006097297604 47\,\epsilon^5\\
&+\ldots
\end{aligned}
$$

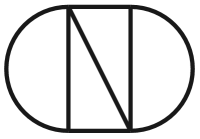
952

$$
\begin{aligned}
&\ 1.49999999999999999999999999999999999999999999999\\
&+3.49999999999999999999999999999999999999999999999\,\epsilon\\
&+4.50000000000000000000000000000000000000000000000\,\epsilon^2\\
&-23.10617070947878285619921448453434997229495887702\,\epsilon^3\\
&-81.84625210730434475325055702968720656910287341851\,\epsilon^4\\
&-665.23777059810474176745476112314598144043587448 43\,\epsilon^5\\
&+\ldots
\end{aligned}
$$



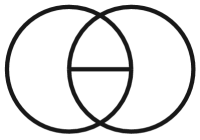
1016

$$
\begin{aligned}
&-0.74999999999999999999999999999999999999999999999\\
&-3.24999999999999999999999999999999999999999999999\,\epsilon\\
&-3.20241742589743499229296970007475318907068645663 8\,\epsilon^2\\
&-23.64446287166479257628968289009838183838632693733\,\epsilon^3\\
&-220.30064253054157442215151674308640540097467863 64\,\epsilon^4\\
&+185.72672725289732916264250527933633691978720619 21\,\epsilon^5\\
&+\ldots
\end{aligned}
$$



1010

$$
\begin{aligned}
&-0.50000000000000000000000000000000000000000000000000 \\
&-2.41666666666666666666666666666666666666666666666666\,\epsilon \\
&-3.80161161726495666152864646671650212604712430442 5\,\epsilon^2 \\
&-18.66035288447634957262454822423808897067280295653\,\epsilon^3 \\
&-117.16243621791132162747140366093341611283526244 09\,\epsilon^4 \\
&-267.78815528995371265658462106510819235671826414 32\,\epsilon^5 \\
&+\ldots
\end{aligned}
$$

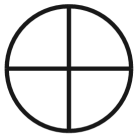


1009

$$
\begin{aligned}
&-0.25000000000000000000000000000000000000000000000000 \\
&-1.33333333333333333333333333333333333333333333333333\,\epsilon \\
&-3.93619604512540594949739472820303438712188177788 6\,\epsilon^2 \\
&-13.23739753045445457230177460694325520434364231558\,\epsilon^3 \\
&-11.96719509053382420804558199010921594173524238764\,\epsilon^4 \\
&-632.61634604430625326073674434891261185214773947 67\,\epsilon^5 \\
&+\ldots
\end{aligned}
$$

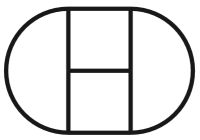
1020

$$
\begin{aligned}
&+1.80308535473939142809960724226717498614747943851 0\,\epsilon^2 \\
&-16.25627696927031709985052518436630112417383502547\,\epsilon^3 \\
&+89.07136923770214012749325727432117027978344859502\,\epsilon^4 \\
&-415.29084991665830539144687139102493726045398479 47\,\epsilon^5 \\
&+1827.15771052792257175437736386490436011266732748 3\,\epsilon^6 \\
&-7810.00582381430338787576792251965352811841799609 1\,\epsilon^7 \\
&+\ldots
\end{aligned}
$$



1011

$$
\begin{aligned}
&+5.18463877571684963165682743228517084028540459750 9\,\epsilon^3 \\
&-59.69766594382494511956968400321266366512646089768\,\epsilon^4 \\
&+403.48510573403908549578035518193673112532004049 72\,\epsilon^5 \\
&-2176.85933463044275515178664032541571848674262148 0\,\epsilon^6 \\
&+10489.71629412778884183467504057023960678789528447\,\epsilon^7 \\
&-47426.26672381895197429844710388040746210832275555\,\epsilon^8 \\
&+\ldots
\end{aligned}
$$



1022

$$
\begin{aligned}
&+1.34894802170970895986445430292139888361832221416 1\,\epsilon^4 \\
&-13.74571732844810597749673020923745948628720600820\,\epsilon^5 \\
&+81.93595470871647214673808535391173195139074515473\,\epsilon^6 \\
&-394.56043290413628585325598520482137638978801413 99\,\epsilon^7 \\
&+1729.01227702733043568489710672149939652554644680 0\,\epsilon^8 \\
&-7243.09884605645431066950222149814208451516498458 7\,\epsilon^9 \\
&+\ldots
\end{aligned}
$$

511

$$
\begin{aligned}
&+0.99767257687426305104909358673611614201186478062 9\,\epsilon^4 \\
&-9.82244751080067924531276547952313861316138791182 9\,\epsilon^5 \\
&+56.87175492287880361818226201426416082730289764036\,\epsilon^6 \\
&-268.46461640254708885043196247723054963189168861 53\,\epsilon^7 \\
&+1162.58695617397693606691730365322106667166318030 5\,\epsilon^8 \\
&-4838.05596926870532621903096183117762996720065982 5\,\epsilon^9 \\
&+\ldots
\end{aligned}
$$

C.4 5-loop

28686

$$
\begin{aligned}
&-3.000000000000000000000000000000000000000000000000 \\
&-1.500000000000000000000000000000000000000000000000\ \epsilon \\
&+0.541666666666666666666666666666666666666666666666\ \epsilon^2 \\
&-0.879861111111111111111111111111111111111111111111\ \epsilon^3 \\
&-1.213252314814814814814814814814814814814814814814\ \epsilon^4 \\
&+135.950728687928714619564927337022185748979929535 8\ \epsilon^5 \\
&+\dots
\end{aligned}
$$

30858

$$
\begin{aligned}
&\ \ 2.400000000000000000000000000000000000000000000000 \\
&+4.850000000000000000000000000000000000000000000000\ \epsilon \\
&+4.575000000000000000000000000000000000000000000000\ \epsilon^2 \\
&-23.65000000000000000000000000000000000000000000000\ \epsilon^3 \\
&-30.277960179458773368562659244050934367654868591 19\ \epsilon^4 \\
&-646.682130156755245399339213018319642021458636355 0\ \epsilon^5 \\
&+\dots
\end{aligned}
$$

30214

$$
\begin{aligned}
&\ \ 2.300000000000000000000000000000000000000000000000 \\
&+4.666666666666666666666666666666666666666666666666\ \epsilon \\
&+1.668278283931623328195313133383168792713790971092\ \epsilon^2 \\
&+17.333469979627949340174443195617457184321799462 06\ \epsilon^3 \\
&+27.482704534857494297098856223951611763317185349 68\ \epsilon^4 \\
&+378.103444504456510184392637978227670846191183262 9\ \epsilon^5 \\
&+\dots
\end{aligned}
$$

29703

$$
\begin{aligned}
&\ \ 2.200000000000000000000000000000000000000000000000 \\
&+4.483333333333333333333333333333333333333333333333\ \epsilon \\
&+4.566666666666666666666666666666666666666666666666\ \epsilon^2 \\
&-9.388702425687269713719528690720609983376975326212\ \epsilon^3 \\
&+19.314078007497527561122766895583565340595811721 45\ \epsilon^4 \\
&+307.304158835584776837653918516060879235556772806 6\ \epsilon^5 \\
&+\dots
\end{aligned}
$$

32386

$$
\begin{aligned}
&-1.250000000000000000000000000000000000000000000000 \\
&-4.791666666666666666666666666666666666666666666666\ \epsilon \\
&-4.989917425897434992292969700074753189070686456638\ \epsilon^2 \\
&-16.966765192625307694105426565510856232994436199 95\ \epsilon^3 \\
&-20.366604984534062133664491474670306795395629107 48\ \epsilon^4 \\
&-283.720687643724097161686508304368579815151739256 6\ \epsilon^5 \\
&+\dots
\end{aligned}
$$

32274

$$
\begin{aligned}
&-1.100000000000000000000000000000000000000000000000 \\
&-4.316666666666666666666666666666666666666666666666\ \epsilon \\
&-6.209944950598289994861979800049835459380457637759\ \epsilon^2 \\
&-12.739202389058147664088072829042155526519305131 28\ \epsilon^3 \\
&+34.949433292010682591871297077453004524900097277 62\ \epsilon^4 \\
&+213.180609119978022699716201102750261963305997452 2\ \epsilon^5 \\
&+\dots
\end{aligned}
$$

32266

$$
\begin{aligned}
&-0.92500000000000000000000000000000000000000000000000\\
&-3.94583333333333333333333333333333333333333333333333\ \epsilon\\
&-5.48743118824786249357747475006229432422557204719 8\ \epsilon^2\\
&-19.46357852559288907006518158822612936218121004990\ \epsilon^3\\
&-51.37125044581005545502000193263395238235621942810\ \epsilon^4\\
&-32.17293665388255170211642453041219246621887663531\ \epsilon^5\\
&+\dots
\end{aligned}
$$

31380

$$
\begin{aligned}
&-0.95000000000000000000000000000000000000000000000000\\
&-3.91666666666666666666666666666666666666666666666666\ \epsilon\\
&-7.25080580863247833076432323335825106302356215 2\ \epsilon^2\\
&+5.13623620639896163505830411807052128155929848766 4\ \epsilon^3\\
&+136.79416616962441615572329946147620508811148104 83\ \epsilon^4\\
&+577.79480854669076695796507163836583698451904274 11\ \epsilon^5\\
&+\dots
\end{aligned}
$$

31246

$$
\begin{aligned}
&-0.65000000000000000000000000000000000000000000000000\\
&-2.86666666666666666666666666666666666666666666666666\ \epsilon\\
&-6.37911799782682042575078042923407772302217852751\ \epsilon^2\\
&-7.98687117068955219410035045695292518237687386855 3\ \epsilon^3\\
&+61.71328604092617393757994432671138847791848732886\ \epsilon^4\\
&+483.79494058360619948868475731144158930670504542 13\ \epsilon^5\\
&+\dots
\end{aligned}
$$

30876

$$
\begin{aligned}
&-1.20000000000000000000000000000000000000000000000000\\
&-4.60000000000000000000000000000000000000000000000000\ \epsilon\\
&-9.10000000000000000000000000000000000000000000000000\ \epsilon^2\\
&+41.58151865769372799823853370446411994828392323710\ \epsilon^3\\
&+307.53798340954805916151025501670467235221283046 91\ \epsilon^4\\
&+1217.59287362512306456961840496644937553474572944 5\ \epsilon^5\\
&+\dots
\end{aligned}
$$

30862

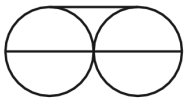
$$
\begin{aligned}
&-0.60000000000000000000000000000000000000000000000000\\
&-2.69999999999999999999999999999999999999999999999999\ \epsilon\\
&-7.97831218919434209498645719587582665927865570053 9\ \epsilon^2\\
&+13.26275192137935340676620540607222513900889972759\ \epsilon^3\\
&+166.60665507266418920492829256696599514724153475 59\ \epsilon^4\\
&+1143.18383075587645994660303038395903232683186058 8\ \epsilon^5\\
&+\dots
\end{aligned}
$$

30222

$$
\begin{aligned}
&-0.60000000000000000000000000000000000000000000000000\\
&-2.86249999999999999999999999999999999999999999999999\ \epsilon\\
&-5.36242909249404656610182269695670545383944941316 3\ \epsilon^2\\
&-20.06150592205425117938426029352451072780542073975\ \epsilon^3\\
&-91.04498558175757706880606535229158160422425519486\ \epsilon^4\\
&+124.65479183312553233003112621467694285689325843 26\ \epsilon^5\\
&+\dots
\end{aligned}
$$

32704

$$
\begin{aligned}
&0.350000000000000000000000000000000000000000000000000000 \\
&+2.350000000000000000000000000000000000000000000000000000\ \epsilon \\
&+5.146324664586893882894531988947030258166089466274\ \epsilon^2 \\
&+22.988171553087973328991060653608932411825485603 64\ \epsilon^3 \\
&+274.008905934322351484098277808479682332855037544 8\ \epsilon^4 \\
&+276.395316232416122349295021766298941620889383145 5\ \epsilon^5 \\
&+\ldots
\end{aligned}
$$

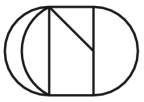


32592

$$
\begin{aligned}
&0.199999999999999999999999999999999999999999999999 \\
&+1.500000000000000000000000000000000000000000000000\ \epsilon \\
&+4.412185522621082218796875422255445861809193980728\ \epsilon^2 \\
&+17.867489800032464788855608125734928852118089207 56\ \epsilon^3 \\
&+138.880019081427326288703370919492323000346582812 5\ \epsilon^4 \\
&+495.621901909635385955347343265215366274535412603 0\ \epsilon^5 \\
&+\ldots
\end{aligned}
$$

32518

$$
\begin{aligned}
&0.299999999999999999999999999999999999999999999999 \\
&+2.100000000000000000000000000000000000000000000000\ \epsilon \\
&+5.268278283931623328195313133383168792713790971092\ \epsilon^2 \\
&+21.617095558082885519185755621910808881820238325 79\ \epsilon^3 \\
&+189.093202559167923345266865434899911266895964084 7\ \epsilon^4 \\
&+928.976884313177695594889407707604998757867839483\ \epsilon^5 \\
&+\ldots
\end{aligned}
$$



32394

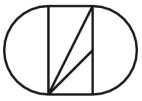
$$
\begin{aligned}
&-0.025000000000000000000000000000000000000000000000\ \epsilon \\
&+2.983053638530904570579293036080914975065462989319\ \epsilon^2 \\
&-24.806311926004153505042133984939376585999196949 40\ \epsilon^3 \\
&+81.539173214618228747048453131743875131542212603 25\ \epsilon^4 \\
&-174.452656637324283057295779258846405841949151367 2\ \epsilon^5 \\
&-1242.325649940329230493812289223380502015253337080\ \epsilon^6 \\
&+\ldots
\end{aligned}
$$

32390

$$
\begin{aligned}
&0.150000000000000000000000000000000000000000000000 \\
&+1.149999999999999999999999999999999999999999999999\ \epsilon \\
&+4.255373283861568235337499463598454390815887260950\ \epsilon^2 \\
&+15.499348521639886526975640297423613481541487897 69\ \epsilon^3 \\
&+53.708154892283932185716370063196569295873817508 74\ \epsilon^4 \\
&+1058.743690637599920074339218910320249047898631214\ \epsilon^5 \\
&+\ldots
\end{aligned}
$$

32329

$$
\begin{aligned}
&0.600000000000000000000000000000000000000000000000 \\
&+3.600000000000000000000000000000000000000000000000\ \epsilon \\
&+5.136556567863246656390626266766337585427581942184\ \epsilon^2 \\
&+30.552444251051603236858999739937720111720796014 26\ \epsilon^3 \\
&+573.960353598399660024218949273853184595391716297 1\ \epsilon^4 \\
&-1310.998766837454065997915351591681359401600370234\ \epsilon^5 \\
&+\ldots
\end{aligned}
$$



32278

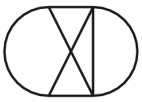
$$
\begin{aligned}
&0.05000000000000000000000000000000000000000000000000\\
&+0.44999999999999999999999999999999999999999999999999\ \epsilon\\
&+2.95989759349890541155898320092416645714078615828 8\ \epsilon^2\\
&+12.14439788313620658714878544428414606056035504366\ \epsilon^3\\
&-0.93160703576480179462526419962139843080808311465 7\ \epsilon^4\\
&+519.27119215716346525900844777782187148315473712 16\ \epsilon^5\\
&+\dots
\end{aligned}
$$

32270

$$
\begin{aligned}
&0.10000000000000000000000000000000000000000000000000\\
&+0.79999999999999999999999999999999999999999999999999\ \epsilon\\
&+2.65609276131054110939843771112772293090459699036 4\ \epsilon^2\\
&+12.49523541032632394470980625904310422947250767949\ \epsilon^3\\
&+79.42288218177492945909736717619494551678126000335\ \epsilon^4\\
&+91.79138259686169597244585173318118920070823092485\ \epsilon^5\\
&+\dots
\end{aligned}
$$

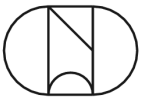
32267

$$
\begin{aligned}
&0.19999999999999999999999999999999999999999999999999\\
&+1.40000000000000000000000000000000000000000000000000\ \epsilon\\
&+4.03341966451683879003671831916231585626818575613 2\ \epsilon^2\\
&+16.39165865854073090802078822602860700600572695094\ \epsilon^3\\
&+138.08147125560144743929574587768179685368169329 27\ \epsilon^4\\
&+435.20265076780389798019775626464967393604251004 63\ \epsilon^5\\
&+\dots
\end{aligned}
$$



31516

$$
\begin{aligned}
&0.04999999999999999999999999999999999999999999999999\\
&+0.45000000000000000000000000000000000000000000000000\ \epsilon\\
&+2.59928052255102712593906175247073145991129027058 6\ \epsilon^2\\
&+11.32709413193374568282983843073178885889590636962\ \epsilon^3\\
&-3.19100545833109441402078550821868768744368402956 0\ \epsilon^4\\
&+568.37131517622121696900123905944775426023272256 06\ \epsilon^5\\
&+\dots
\end{aligned}
$$

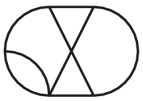


31388

$$
\begin{aligned}
&0.10000000000000000000000000000000000000000000000000\\
&+0.80000000000000000000000000000000000000000000000000\ \epsilon\\
&+4.09856104510205425187812350494146291982258054117 2\ \epsilon^2\\
&+14.47367552703882140757535826292603809988287013862\ \epsilon^3\\
&-29.37326015015386172875499984198288560748216132345\ \epsilon^4\\
&+1530.34331205594879172007282949741249436552273836 9\ \epsilon^5\\
&+\dots
\end{aligned}
$$

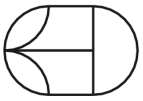
30231

$$
\begin{aligned}
&0.72123414189575657123984289690686999445899177540 4\ \epsilon^2\\
&+9.13351316904043015003740006460624357788146104306 2\ \epsilon^3\\
&+18.95318188466870382986912163074957478331662315353\ \epsilon^4\\
&-531.32391547725635267943444561495368318398901378 43\ \epsilon^5\\
&+7502.67382425933691859121401543210915787133063036 1\ \epsilon^6\\
&-56141.78112476381681554972844331060055708116158225\ \epsilon^7\\
&+\dots
\end{aligned}
$$



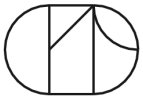
32736

$$
\begin{aligned}
&-4.14771102057347970532546194582813667222832367800 7\ \epsilon^3\\
&+56.76739666338249149041641437964339657915885294410\ \epsilon^4\\
&-462.59915117959131341842175597738647742671116565 64\ \epsilon^5\\
&+3050.20144794936579926852298359493148716924794158 1\ \epsilon^6\\
&-18170.11060050125316561097773107847431297636110621\ \epsilon^7\\
&+102401.97586216856372968452090504832089713377554 34\ \epsilon^8\\
&+\dots
\end{aligned}
$$



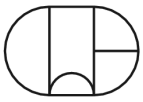
32712

$$
\begin{aligned}
&-1.44246828379151314247968579381373998891798355080 8\ \epsilon^2\\
&+15.66147121159669442917296250244952168873398529073\ \epsilon^3\\
&-103.52393640220653173601706219112234353366231228 10\ \epsilon^4\\
&+580.47410673026355451114418092771879661803718503 95\ \epsilon^5\\
&-3073.87933691816527560638002507859968284344982239 3\ \epsilon^6\\
&+15892.11991655887990072312985187902211863523504677\ \epsilon^7\\
&+\dots
\end{aligned}
$$



32708

$$
\begin{aligned}
&-0.72123414189575657123984289690686999445899177540 4\ \epsilon^2\\
&+7.10950146390259064334663835431789084990800086996 1\ \epsilon^3\\
&-47.13186277613555829347294402151405507389821974085\ \epsilon^4\\
&+296.19056251006069266665125022092315295328546186 68\ \epsilon^5\\
&-1830.15921514195382133217553566702772600119872901 7\ \epsilon^6\\
&+10898.77592636784646364500094992325948383560868844\ \epsilon^7\\
&+\dots
\end{aligned}
$$

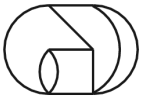


32674

$$
\begin{aligned}
&-1.44246828379151314247968579381373998891798355080 8\ \epsilon^2\\
&+15.66147121159669442917296250244952168873398529073\ \epsilon^3\\
&-105.59779191249327158867979316403641186977647412 00\ \epsilon^4\\
&+618.41488178871017941243346143732902137922442324 58\ \epsilon^5\\
&-3468.94684584826378964734296521301634221664015007 2\ \epsilon^6\\
&+19037.57014187524101245199861004587803837742830381\ \epsilon^7\\
&+\dots
\end{aligned}
$$

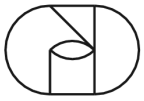
32596

$$
\begin{aligned}
&-14.11688988334691957575716569789715463439808984791\ \epsilon^4\\
&+235.07729596783467131454388080950411779239347239 58\ \epsilon^5\\
&-2267.73868329300841229629944805802058554875454137 3\ \epsilon^6\\
&+17032.05736733550461114507260154791350899541515054\ \epsilon^7\\
&-111545.92934988427964433974836267595922961587302 42\ \epsilon^8\\
&+672718.68651995445291652158970868547547602526534 96\ \epsilon^9\\
&+\dots
\end{aligned}
$$



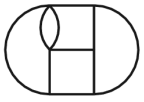
32562

$$
\begin{aligned}
&-1.44246828379151314247968579381373998891798355080 8\ \epsilon^2\\
&+15.66147121159669442917296250244952168873398529073\ \epsilon^3\\
&-102.48700864706316180968569670466530936560523136 15\ \epsilon^4\\
&+561.42550965554118029895153425329926065016306089 99\ \epsilon^5\\
&-2875.16990178617470749733121561052472304468958488 1\ \epsilon^6\\
&+14309.66789055985793699059222565628586547916818050\ \epsilon^7\\
&+\dots
\end{aligned}
$$



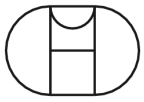
32534

$$
\begin{aligned}
&-4.14771102057347970532546194582813667222832367800 7\ \epsilon^3\\
&+58.50132562150285257111283647421178197177933543101\ \epsilon^4\\
&-494.04982783803847562279179169540491427482027808 55\ \epsilon^5\\
&+3375.25614144190303173453832323782718128267296348 7\ \epsilon^6\\
&-20742.48877945843156784497186190848486909447787042\ \epsilon^7\\
&+119909.98235738689961540078107464482578895995638 46\ \epsilon^8\\
&+\dots
\end{aligned}
$$



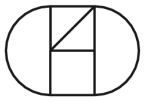
32398

$$
\begin{aligned}
&-0.72123414189575657123984289690686999445899177540 4\ \epsilon^2\\
&+7.10950146390259064334663835431789084990800086996 1\ \epsilon^3\\
&-44.02107951070544851447884756214295256972697698235\ \epsilon^4\\
&+242.02064900465830058277691203646883144579831541 02\ \epsilon^5\\
&-1289.19150744767382848706458165359604270063392178 4\ \epsilon^6\\
&+6731.57486331651755738985125464281955650039174550 8\ \epsilon^7\\
&+\dots
\end{aligned}
$$



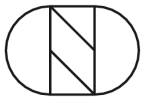
32391

$$
\begin{aligned}
&-4.14771102057347970532546194582813667222832367800 7\ \epsilon^3\\
&+60.23525457962321365180925856878016736439981791793\ \epsilon^4\\
&-524.58456864801302754278455058408513431513105135 25\ \epsilon^5\\
&+3683.53152670172843083958663115012552671865192636 8\ \epsilon^6\\
&-23140.29050618808330097812925263037891370001191410\ \epsilon^7\\
&+136031.07329448496063499378233980894882711056316 76\ \epsilon^8\\
&+\dots
\end{aligned}
$$



32279

$$
\begin{aligned}
&-11.11705078313569916590876798709400927458849575523\ \epsilon^4\\
&+181.78223928612340820790788236018642961198741994 20\ \epsilon^5\\
&-1725.99961374035208059516739924421172886077049575 4\ \epsilon^6\\
&+12797.99987372682404668555169033764331697450364155\ \epsilon^7\\
&-82986.85269258138216055904714739092570657110940340\ \epsilon^8\\
&+496710.27285614832821523150890358612704268599246 41\ \epsilon^9\\
&+\dots
\end{aligned}
$$

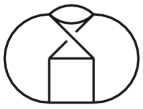


31420

$$
\begin{aligned}
&-10.40357374872216648417853256741031235572289492149\ \epsilon^4\\
&+167.81535305918474061962120601112466233675898298 29\ \epsilon^5\\
&-1572.95815263154633216873701292443012073231185030 4\ \epsilon^6\\
&+11533.80237760276728405588733233544262789155612591\ \epsilon^7\\
&-74112.47851638470253497575693603457890960000148877\ \epsilon^8\\
&+440426.33241866068120544817030131362176285114850 98\ \epsilon^9\\
&+\dots
\end{aligned}
$$

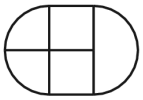
30239

$$
\begin{aligned}
&-10.40357374872216648417853256741031235572289492149\ \epsilon^4\\
&+170.16887918440485844269144771403153445469585733 94\ \epsilon^5\\
&-1617.42761098599460250873651235321159305059428910 9\ \epsilon^6\\
&+12009.95234181003289104385888288726798181425157927\ \epsilon^7\\
&-77990.23000502531150901441543067082347645664379563\ \epsilon^8\\
&+467415.41013486017822671480977953301946684809784 04\ \epsilon^9\\
&+\dots
\end{aligned}
$$



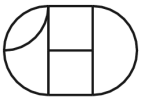
29550

$$
\begin{aligned}
&-4.14771102057347970532546194582813667222832367800 7\ \epsilon^3\\
&+61.96918353774357473250568066334855275702030040485\ \epsilon^4\\
&-554.91071568731937607747903875721289947741058944 29\ \epsilon^5\\
&+3986.99793243934266330848901228905133090644932585 8\ \epsilon^6\\
&-25478.27907133643217288387221762192103140607530609\ \epsilon^7\\
&+151608.39853584839414527203238386216422385066031 35\ \epsilon^8\\
&+\dots
\end{aligned}
$$



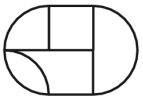
32744

$$
\begin{aligned}
&-2.99823287368040309416876288359009577308740181794 8\ \epsilon^5\\
&+43.75280252615427411169257824938492717855692308423\ \epsilon^6\\
&-368.31966707754814372566473942512877860267835238 99\ \epsilon^7\\
&+2428.02190681555945902174341703492902481384238986 7\ \epsilon^8\\
&-14126.64461336412148378626905087785317127428545699\ \epsilon^9\\
&+76819.46393126078913111863700014264017725700338615\ \epsilon^{10}\\
&+\dots
\end{aligned}
$$



32737

$$
\begin{aligned}
&-1.34894802170970895986445430292139888361832221416 1\ \epsilon^4\\
&+17.40126602800759993452487395539252598288896429420\ \epsilon^5\\
&-132.89211813596259361282516635364924673629411710 60\ \epsilon^6\\
&+820.18123923374054766079619059228853925713203960 83\ \epsilon^7\\
&-4583.65371298775202614666947885279980952666806052 5\ \epsilon^8\\
&+24349.66876563193939110528524660161406231921592191\ \epsilon^9\\
&+\dots
\end{aligned}
$$



32713

$$
\begin{aligned}
&-1.34894802170970895986445430292139888361832221416 1\ \epsilon^4\\
&+16.98146619252512270971010570864997294635856828946\ \epsilon^5\\
&-126.16093959790626286738387435036125454037892305 52\ \epsilon^6\\
&+759.06884383533412802490095216514328475840384679 65\ \epsilon^7\\
&-4157.94976491274209238147845149748316599566641828 7\ \epsilon^8\\
&+21777.73803722606240195514560495004324585202245229\ \epsilon^9\\
&+\dots
\end{aligned}
$$

32682

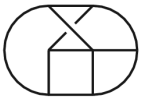
$$
\begin{aligned}
&-3.81655501580294058793897789789716992864455666885 0\ \epsilon^5\\
&+56.69539775835035037600667776934274837412827940746\ \epsilon^6\\
&-484.70750812090395161213924435501594162613926053 99\ \epsilon^7\\
&+3234.06704025609968644136978168797108356482098188 7\ \epsilon^8\\
&-18980.99682611361784720299594740439856094964894219\ \epsilon^9\\
&+103832.33598120469289979873336247916544614611786 92\ \epsilon^{10}\\
&+\dots
\end{aligned}
$$

31736

$$
\begin{aligned}
&-2.21096283270677178249823255964778140410637976127 1\ \epsilon^5\\
&+31.23765127619360951569435211872040110946874586881\ \epsilon^6\\
&-255.31679618584894131301303179714175887905258905 64\ \epsilon^7\\
&+1643.12481525352675372746545276370226743790197787 5\ \epsilon^8\\
&-9390.39400898396900201987557273485184921706568740 4\ \epsilon^9\\
&+50430.86554729172863633478848042785260419782367077\ \epsilon^{10}\\
&+\dots
\end{aligned}
$$

30691

$$
\begin{aligned}
&-0.997672576874263051049093586736116142011864780629\ \epsilon^4\\
&+12.367217044215879225428463265068143700238970396 72\ \epsilon^5\\
&-91.039465930893167081232217903579731374351272093 07\ \epsilon^6\\
&+545.814332034220841233085711288998331819108380005 2\ \epsilon^7\\
&-2988.239057897256094676702060143265114831927288335\ \epsilon^8\\
&+15658.319159938521082878197428263782272944847788 59\ \epsilon^9\\
&+\dots
\end{aligned}
$$



30526

$$
\begin{aligned}
&-1.970390455317890254668792371697895351281107398428\ \epsilon^5\\
&+27.570732867329834761570436624478114396622402177 42\ \epsilon^6\\
&-223.535719989701438879564094282935424122967623044 1\ \epsilon^7\\
&+1430.172138637487321663810941305184259016492117714\ \epsilon^8\\
&-8141.825356850975193558368516068423272809313860785\ \epsilon^9\\
&+43621.107674556821059131280209641361723921727157 99\ \epsilon^{10}\\
&+\dots
\end{aligned}
$$

References

[1] S. Laporta. "High precision calculation of multiloop Feynman integrals by difference equations". In: *Int.J.Mod.Phys.* A15 (2000), pp. 5087–5159. DOI: 10.1016/S0217-751X(00)00215-7. arXiv: hep-ph/0102033 [hep-ph].

[2] T. van Ritbergen, J. A. M. Vermaseren, and S. A. Larin. "The Four loop beta function in quantum chromodynamics". In: *Phys. Lett.* B400 (1997), pp. 379–384. DOI: 10.1016/S0370-2693(97)00370-5. arXiv: hep-ph/9701390 [hep-ph].

[3] M. Czakon. "The Four-loop QCD beta-function and anomalous dimensions". In: *Nucl. Phys.* B710 (2005), pp. 485–498. DOI: 10.1016/j.nuclphysb.2005.01.012. arXiv: hep-ph/0411261 [hep-ph].

[4] S. Laporta. "High precision epsilon expansions of massive four loop vacuum bubbles". In: *Phys.Lett.* B549 (2002), pp. 115–122. DOI: 10.1016/S0370-2693(02)02910-6. arXiv: hep-ph/0210336 [hep-ph].

[5] Y. Schröder and A. Vuorinen. "High precision evaluation of four loop vacuum bubbles in three-dimensions". In: (2003). arXiv: hep-ph/0311323 [hep-ph].

[6] J. Möller. "Fully Massive Tadpoles at 5-loop: Reduction and Difference Equations". PhD thesis. Bielefeld University, 2012. URL: http://pub.uni-bielefeld.de/publication/2509818.

[7] T. Luthe. "Difference Equations for 5-loop Massive Vacuum Integrals Using Syzygies". MA thesis. Bielefeld University, 2012. URL: http://www2.physik.uni-bielefeld.de/fileadmin/Masterarbeit_ThomasLuthe.pdf.

[8] C.-N. Yang and R. L. Mills. "Conservation of Isotopic Spin and Isotopic Gauge Invariance". In: *Phys. Rev.* 96 (1954), pp. 191–195. DOI: 10.1103/PhysRev.96.191.

[9] L. D. Faddeev and V. N. Popov. "Feynman Diagrams for the Yang-Mills Field". In: *Phys. Lett.* B25 (1967), pp. 29–30. DOI: 10.1016/0370-2693(67)90067-6.

[10] G. 't Hooft. "Dimensional regularization and the renormalization group". In: *Nucl.Phys.* B61 (1973), pp. 455–468. DOI: 10.1016/0550-3213(73)90376-3.

[11] G. 't Hooft and M. J. G. Veltman. "Regularization and Renormalization of Gauge Fields". In: *Nucl. Phys.* B44 (1972), pp. 189–213. DOI: 10.1016/0550-3213(72)90279-9.

[12] G. 't Hooft. Report at the Marseille Conference on Yang-Mills Fields. 1972.

[13] D. J. Gross and F. Wilczek. "Ultraviolet Behavior of Nonabelian Gauge Theories". In: *Phys. Rev. Lett.* 30 (1973), pp. 1343–1346. DOI: 10.1103/PhysRevLett.30.1343.

[14] H. D. Politzer. "Reliable Perturbative Results for Strong Interactions?" In: *Phys. Rev. Lett.* 30 (1973), pp. 1346–1349. DOI: 10.1103/PhysRevLett.30.1346.

[15] W. E. Caswell. "Asymptotic Behavior of Nonabelian Gauge Theories to Two Loop Order". In: *Phys. Rev. Lett.* 33 (1974), p. 244. DOI: 10.1103/PhysRevLett.33.244.

[16] D. R. T. Jones. "Two Loop Diagrams in Yang-Mills Theory". In: *Nucl. Phys.* B75 (1974), p. 531. DOI: 10.1016/0550-3213(74)90093-5.

[17] E. Egorian and O. V. Tarasov. "Two Loop Renormalization of the QCD in an Arbitrary Gauge". In: *Teor. Mat. Fiz.* 41 (1979). [Theor. Math. Phys.41,863(1979)], pp. 26–32.

[18] O. V. Tarasov, A. A. Vladimirov, and A. Yu. Zharkov. "The Gell-Mann-Low Function of QCD in the Three Loop Approximation". In: *Phys. Lett.* B93 (1980), pp. 429–432. DOI: 10.1016/0370-2693(80)90358-5.

[19] S. A. Larin and J. A. M. Vermaseren. "The Three loop QCD Beta function and anomalous dimensions". In: *Phys. Lett.* B303 (1993), pp. 334–336. DOI: 10.1016/0370-2693(93)91441-O. arXiv: hep-ph/9302208 [hep-ph].

[20] W. A. Bardeen et al. "Deep Inelastic Scattering Beyond the Leading Order in Asymptotically Free Gauge Theories". In: *Phys. Rev.* D18 (1978), p. 3998. DOI: 10.1103/PhysRevD.18.3998.

[21] L. F. Abbott. "The Background Field Method Beyond One Loop". In: *Nucl. Phys.* B185 (1981), p. 189. DOI: 10.1016/0550-3213(81)90371-0.

[22] K. G. Chetyrkin, P. A. Baikov, and J. H. Kühn. "Towards QCD running in 5 loops: quark mass anomalous dimension". In: *PoS* RADCOR2013 (2013), p. 056. arXiv: 1402.6606 [hep-ph].

[23] P. A. Baikov, K. G. Chetyrkin, and J. H. Kühn. "Quark Mass and Field Anomalous Dimensions to $\mathcal{O}(\alpha\_s^5)$". In: *JHEP* 10 (2014), p. 76. DOI: 10.1007/JHEP10(2014)076. arXiv: 1402.6611 [hep-ph].

[24] K. G. Chetyrkin and V. A. Smirnov. "R* Operation Corrected". In: *Phys. Lett.* B144 (1984), pp. 419–424. DOI: 10.1016/0370-2693(84)91291-7.

[25] K. G. Chetyrkin, M. Misiak, and M. Münz. "Beta functions and anomalous dimensions up to three loops". In: *Nucl. Phys.* B518 (1998), pp. 473–494. DOI: 10.1016/S0550-3213(98)00122-9. arXiv: hep-ph/9711266 [hep-ph].

[26] E. Braaten and A. Nieto. "Free energy of QCD at high temperature". In: *Phys. Rev.* D53 (1996), pp. 3421–3437. DOI: 10.1103/PhysRevD.53.3421. arXiv: hep-ph/9510408 [hep-ph].

[27] D. J. Gross, R. D. Pisarski, and L. G. Yaffe. "QCD and Instantons at Finite Temperature". In: *Rev. Mod. Phys.* 53 (1981), p. 43. DOI: 10.1103/RevModPhys.53.43.

[28] T. Appelquist and R. D. Pisarski. "High-Temperature Yang-Mills Theories and Three-Dimensional Quantum Chromodynamics". In: *Phys. Rev.* D23 (1981), p. 2305. DOI: 10.1103/PhysRevD.23.2305.

[29] K. Kajantie et al. "The Pressure of hot QCD up to g6 ln(1/g)". In: *Phys. Rev.* D67 (2003), p. 105008. DOI: 10.1103/PhysRevD.67.105008. arXiv: hep-ph/0211321 [hep-ph].

[30] A. Gynther et al. "Four-loop pressure of massless O(N) scalar field theory". In: *JHEP* 04 (2007), p. 094. DOI: 10.1088/1126-6708/2007/04/094. arXiv: hep-ph/0703307 [HEP-PH].

[31] J. O. Andersen, L. Kyllingstad, and L. E. Leganger. "Pressure to order g**8 log g of massless phi**4 theory at weak coupling". In: *JHEP* 08 (2009), p. 066. DOI: 10.1088/1126-6708/2009/08/066. arXiv: 0903.4596 [hep-ph].

[32] R. Boughezal, M. Czakon, and T. Schutzmeier. "Four-Loop Tadpoles: Applications in QCD". In: *Nucl. Phys. Proc. Suppl.* 160 (2006). [,160(2006)], pp. 160–164. DOI: 10.1016/j.nuclphysbps.2006.09.041. arXiv: hep-ph/0607141 [hep-ph].

[33] R. Boughezal and M. Czakon. "Single scale tadpoles and O($G_F m_t^2 \alpha_s^3$) corrections to the ρ parameter". In: *Nucl. Phys.* B755 (2006), pp. 221–238. DOI: 10.1016/j.nuclphysb.2006.08.007. arXiv: hep-ph/0606232 [hep-ph].

[34] K. G. Chetyrkin et al. "Massive Tadpoles: Techniques and Applications". In: *Proceedings, Advances in Computational Particle Physics: Final Meeting (SFB-TR-9).* Vol. 261-262. 2015, pp. 19–30. DOI: 10.1016/j.nuclphysbps.2015.03.003. arXiv: 1502.00509 [hep-ph]. URL: http://inspirehep.net/record/1342470/files/arXiv:1502.00509.pdf.

[35] M. J. G. Veltman. "Limit on Mass Differences in the Weinberg Model". In: *Nucl. Phys.* B123 (1977), p. 89. DOI: 10.1016/0550-3213(77)90342-X.

[36] K. G. Chetyrkin et al. "Four-Loop QCD Corrections to the Rho Parameter". In: *Phys. Rev. Lett.* 97 (2006), p. 102003. DOI: 10.1103/PhysRevLett.97.102003. arXiv: hep-ph/0605201 [hep-ph].

[37] M. Awramik et al. "Precise prediction for the W boson mass in the standard model". In: *Phys. Rev.* D69 (2004), p. 053006. DOI: 10.1103/PhysRevD.69.053006. arXiv: hep-ph/0311148 [hep-ph].

[38] S.-Q. Wang et al. "QCD improved electroweak parameter ρ". In: *Phys. Rev.* D89.11 (2014), p. 116001. DOI: 10.1103/PhysRevD.89.116001. arXiv: 1402.0975 [hep-ph].

[39] M. Awramik et al. "Complete two-loop electroweak fermionic corrections to $\sin^2\theta^{\rm lept}_{\rm eff}$ and indirect determination of the Higgs boson mass". In: *Phys. Rev. Lett.* 93 (2004), p. 201805. DOI: 10.1103/PhysRevLett.93.201805. arXiv: hep-ph/0407317 [hep-ph].

[40] K. G. Chetyrkin, Johann H. Kühn, and Christian Sturm. "QCD decoupling at four loops". In: *Nucl. Phys.* B744 (2006), pp. 121–135. DOI: 10.1016/j.nuclphysb.2006.03.020. arXiv: hep-ph/0512060 [hep-ph].

[41] Y. Schröder and M. Steinhauser. "Four-loop decoupling relations for the strong coupling". In: *JHEP* 01 (2006), p. 051. DOI: 10.1088/1126-6708/2006/01/051. arXiv: hep-ph/0512058 [hep-ph].

[42] K. G. Chetyrkin, Bernd A. Kniehl, and M. Steinhauser. "Decoupling relations to O (alpha-s**3) and their connection to low-energy theorems". In: *Nucl. Phys.* B510 (1998), pp. 61–87. DOI: 10.1016/S0550-3213(97)00649-4. arXiv: hep-ph/9708255 [hep-ph].

[43] C. Sturm. "Higher order QCD results for the fermionic contributions of the Higgs-boson decay into two photons and the decoupling function for the $\overline{\rm MS}$ renormalized fine-structure constant". In: *Eur. Phys. J.* C74.8 (2014), p. 2978. DOI: 10.1140/epjc/s10052-014-2978-0. arXiv: 1404.3433 [hep-ph].

[44] V. A. Novikov et al. "Charmonium and Gluons: Basic Experimental Facts and Theoretical Introduction". In: *Phys. Rept.* 41 (1978), pp. 1–133. DOI: 10.1016/0370-1573(78)90120-5.

[45] J. H. Kühn, M. Steinhauser, and C. Sturm. "Heavy Quark Masses from Sum Rules in Four-Loop Approximation". In: *Nucl. Phys.* B778 (2007), pp. 192–215. DOI: 10.1016/j.nuclphysb.2007.04.036. arXiv: hep-ph/0702103 [HEP-PH].

[46] K. G. Chetyrkin et al. "Charm and Bottom Quark Masses: An Update". In: *Phys. Rev.* D80 (2009), p. 074010. DOI: 10.1103/PhysRevD.80.074010. arXiv: 0907.2110 [hep-ph].

[47] T. Gehrmann and E. Remiddi. "Differential equations for two loop four point functions". In: *Nucl. Phys.* B580 (2000), pp. 485–518. DOI: 10.1016/S0550-3213(00)00223-6. arXiv: hep-ph/9912329 [hep-ph].

[48] T. Binoth et al. "Two loop corrections to light by light scattering in supersymmetric QED". In: *JHEP* 05 (2002), p. 060. DOI: 10.1088/1126-6708/2002/05/060. arXiv: hep-ph/0202266 [hep-ph].

[49] K.G. Chetyrkin and F.V. Tkachov. "Integration by Parts: The Algorithm to Calculate beta Functions in 4 Loops". In: *Nucl.Phys.* B192 (1981), pp. 159–204. DOI: 10.1016/0550-3213(81)90199-1.

[50] R.N. Lee. "Group structure of the integration-by-part identities and its application to the reduction of multiloop integrals". In: *JHEP* 0807 (2008), p. 031. DOI: 10.1088/1126-6708/2008/07/031. arXiv: 0804.3008 [hep-ph].

[51] J. Gluza, K. Kajda, and D. A. Kosower. "Towards a Basis for Planar Two-Loop Integrals". In: *Phys.Rev.* D83 (2011), p. 045012. DOI: 10.1103/PhysRevD.83.045012. arXiv: 1009.0472 [hep-th].

[52] R. M. Schabinger. "A New Algorithm For The Generation Of Unitarity-Compatible Integration By Parts Relations". In: *JHEP* 1201 (2012), p. 077. DOI: 10.1007/JHEP01(2012)077. arXiv: 1111.4220 [hep-ph].

[53] O.V. Tarasov. "Connection between Feynman integrals having different values of the space-time dimension". In: *Phys.Rev.* D54 (1996), pp. 6479–6490. DOI: 10.1103/PhysRevD.54.6479. arXiv: hep-th/9606018 [hep-th].

[54] A. von Manteuffel, E. Panzer, and R. M. Schabinger. "A quasi-finite basis for multi-loop Feynman integrals". In: *JHEP* 1502 (2015), p. 120. DOI: 10.1007/JHEP02(2015)120. arXiv: 1411.7392 [hep-ph].

[55] M. E. Peskin and D. V. Schroeder. *An Introduction to quantum field theory*. Westview, 1995.

[56] C. Bogner and S. Weinzierl. "Feynman graph polynomials". In: *Int. J. Mod. Phys.* A25 (2010), pp. 2585–2618. DOI: 10.1142/S0217751X10049438. arXiv: 1002.3458 [hep-ph].

[57] C. Anastasiou and A. Lazopoulos. "Automatic integral reduction for higher order perturbative calculations". In: *JHEP* 0407 (2004), p. 046. DOI: 10.1088/1126-6708/2004/07/046. arXiv: hep-ph/0404258 [hep-ph].

[58] A.V. Smirnov. "Algorithm FIRE – Feynman Integral REduction". In: *JHEP* 0810 (2008), p. 107. DOI: 10.1088/1126-6708/2008/10/107. arXiv: 0807.3243 [hep-ph].

[59] C. Studerus. "Reduze-Feynman Integral Reduction in C++". In: *Comput. Phys. Commun.* 181 (2010), pp. 1293–1300. DOI: 10.1016/j.cpc.2010.03.012. arXiv: 0912.2546 [physics.comp-ph].

[60] A. von Manteuffel and C. Studerus. "Reduze 2 - Distributed Feynman Integral Reduction". In: (2012). arXiv: 1201.4330 [hep-ph].

[61] A.V. Smirnov and A.V. Petukhov. "The Number of Master Integrals is Finite". In: *Lett.Math.Phys.* 97 (2011), pp. 37–44. DOI: 10.1007/s11005-010-0450-0. arXiv: 1004.4199 [hep-th].

[62] A. von Manteuffel and R. M. Schabinger. "A novel approach to integration by parts reduction". In: *Phys.Lett.* B744 (2015), pp. 101–104. DOI: 10.1016/j.physletb.2015.03.029. arXiv: 1406.4513 [hep-ph].

[63] P. Kant. "Finding Linear Dependencies in Integration-By-Parts Equations: A Monte Carlo Approach". In: *Comput.Phys.Commun.* 185 (2014), pp. 1473–1476. DOI: 10.1016/j.cpc.2014.01.017. arXiv: 1309.7287 [hep-ph].

[64] H. Cheng and T. T. Wu. *Expanding Protons: Scattering At High-energies*. 1987.

[65] T. Binoth and G. Heinrich. "An automatized algorithm to compute infrared divergent multiloop integrals". In: *Nucl. Phys.* B585 (2000), pp. 741–759. DOI: 10.1016/S0550-3213(00)00429-6. arXiv: hep-ph/0004013 [hep-ph].

[66] G. Heinrich. "Sector Decomposition". In: *Int.J.Mod.Phys.* A23 (2008), pp. 1457–1486. DOI: 10.1142/S0217751X08040263. arXiv: 0803.4177 [hep-ph].

[67] C. Bogner and S. Weinzierl. "Resolution of singularities for multi-loop integrals". In: *Comput. Phys. Commun.* 178 (2008), pp. 596–610. DOI: 10.1016/j.cpc.2007.11.012. arXiv: 0709.4092 [hep-ph].

[68] J. Carter and G. Heinrich. "SecDec: A general program for sector decomposition". In: *Comput. Phys. Commun.* 182 (2011), pp. 1566–1581. DOI: 10.1016/j.cpc.2011.03.026. arXiv: 1011.5493 [hep-ph].

[69] S. Borowka, J. Carter, and G. Heinrich. "Numerical Evaluation of Multi-Loop Integrals for Arbitrary Kinematics with SecDec 2.0". In: *Comput. Phys. Commun.* 184 (2013), pp. 396–408. DOI: 10.1016/j.cpc.2012.09.020. arXiv: 1204.4152 [hep-ph].

[70] S. Borowka et al. "SecDec-3.0: numerical evaluation of multi-scale integrals beyond one loop". In: (2015). arXiv: 1502.06595 [hep-ph].

[71] A. V. Smirnov and M. N. Tentyukov. "Feynman Integral Evaluation by a Sector decomposiTion Approach (FIESTA)". In: *Comput. Phys. Commun.* 180 (2009), pp. 735–746. DOI: 10.1016/j.cpc.2008.11.006. arXiv: 0807.4129 [hep-ph].

[72] A. V. Smirnov, V. A. Smirnov, and M. Tentyukov. "FIESTA 2: Parallelizeable multiloop numerical calculations". In: *Comput. Phys. Commun.* 182 (2011), pp. 790–803. DOI: 10.1016/j.cpc.2010.11.025. arXiv: 0912.0158 [hep-ph].

[73] A. V. Smirnov. "FIESTA 3: cluster-parallelizable multiloop numerical calculations in physical regions". In: *Comput. Phys. Commun.* 185 (2014), pp. 2090–2100. DOI: 10.1016/j.cpc.2014.03.015. arXiv: 1312.3186 [hep-ph].

[74] J. M. Borwein et al. “Special values of multiple polylogarithms”. In: *Trans. Am. Math. Soc.* 353 (2001), pp. 907–941. DOI: 10.1090/S0002-9947-00-02616-7. arXiv: math/9910045 [math-ca].

[75] E. Remiddi and J. A. M. Vermaseren. “Harmonic polylogarithms”. In: *Int. J. Mod. Phys.* A15 (2000), pp. 725–754. DOI: 10.1142/S0217751X00000367. arXiv: hep-ph/9905237 [hep-ph].

[76] A. B. Goncharov. “Multiple polylogarithms, cyclotomy and modular complexes”. In: *Math.Res.Lett.* 5 (1998), pp. 497–516. DOI: 10.4310/MRL.1998.v5.n4.a7. arXiv: 1105.2076 [math.AG].

[77] F. Brown. “The Massless higher-loop two-point function”. In: *Commun. Math. Phys.* 287 (2009), pp. 925–958. DOI: 10.1007/s00220-009-0740-5. arXiv: 0804.1660 [math.AG].

[78] E. Panzer. “On hyperlogarithms and Feynman integrals with divergences and many scales”. In: *JHEP* 03 (2014), p. 071. DOI: 10.1007/JHEP03(2014)071. arXiv: 1401.4361 [hep-th].

[79] E. Panzer. “Feynman integrals and hyperlogarithms”. PhD thesis. Humboldt U., Berlin, Inst. Math., 2015. arXiv: 1506.07243 [math-ph].

[80] M. B. Monagan et al. *Maple 10 Programming Guide*. Waterloo ON, Canada: Maplesoft, 2005.

[81] E. Panzer. “Algorithms for the symbolic integration of hyperlogarithms with applications to Feynman integrals”. In: *Comput.Phys.Commun.* 188 (2014), pp. 148–166. DOI: 10.1016/j.cpc.2014.10.019. arXiv: 1403.3385 [hep-th].

[82] S. Laporta. “Analytical expressions of 3 and 4-loop sunrise Feynman integrals and 4-dimensional lattice integrals”. In: *Int.J.Mod.Phys.* A23 (2008), pp. 5007–5020. DOI: 10.1142/S0217751X08042869. arXiv: 0803.1007 [hep-ph].

[83] V. A. Smirnov. “Evaluating Feynman integrals”. In: *Springer Tracts Mod. Phys.* 211 (2004), pp. 1–244.

[84] E. W. Barnes. “A New Development of the Theory of the Hypergeometric Functions”. In: *Proceedings of the London Mathematical Society* s2-6.1 (1908), pp. 141–177. DOI: 10.1112/plms/s2-6.1.141. eprint: http://plms.oxfordjournals.org/content/s2-6/1/141.full.pdf+html. URL: http://plms.oxfordjournals.org/content/s2-6/1/141.short.

[85] M. Czakon. “Automatized analytic continuation of Mellin-Barnes integrals”. In: *Comput. Phys. Commun.* 175 (2006), pp. 559–571. DOI: 10.1016/j.cpc.2006.07.002. arXiv: hep-ph/0511200 [hep-ph].

[86] A. V. Smirnov and V. A. Smirnov. “On the Resolution of Singularities of Multiple Mellin-Barnes Integrals”. In: *Eur. Phys. J.* C62 (2009), pp. 445–449. DOI: 10.1140/epjc/s10052-009-1039-6. arXiv: 0901.0386 [hep-ph].

[87] J. Gluza, K. Kajda, and T. Riemann. “AMBRE: A Mathematica package for the construction of Mellin-Barnes representations for Feynman integrals”. In: *Comput. Phys. Commun.* 177 (2007), pp. 879–893. DOI: 10.1016/j.cpc.2007.07.001. arXiv: 0704.2423 [hep-ph].

[88] J. Gluza et al. "Numerical Evaluation of Tensor Feynman Integrals in Euclidean Kinematics". In: *Eur. Phys. J.* C71 (2011), p. 1516. DOI: 10.1140/epjc/s10052-010-1516-y. arXiv: 1010.1667 [hep-ph].

[89] I. Dubovyk, J. Gluza, K. Kajda, T. Rieman. *AMBRE project webpage*. Silesian Univ., Katowice. URL: http://www.us.edu.pl/~gluza/ambre.

[90] C. Schneider. "Modern Summation Methods for Loop Integrals in Quantum Field Theory: The Packages Sigma, EvaluateMultiSums and SumProduction". In: *J. Phys. Conf. Ser.* 523 (2014), p. 012037. DOI: 10.1088/1742-6596/523/1/012037. arXiv: 1310.0160 [cs.SC].

[91] J. Blümlein et al. "Non-planar Feynman integrals, Mellin-Barnes representations, multiple sums". In: *PoS* LL2014 (2014), p. 052. arXiv: 1407.7832 [hep-ph].

[92] I. Dubovyk, J. Gluza, and T. Riemann. "Non-planar Feynman diagrams and Mellin-Barnes representations with *AMBRE* 3.0". In: *J. Phys. Conf. Ser.* 608.1 (2015), p. 012070. DOI: 10.1088/1742-6596/608/1/012070.

[93] A. V. Kotikov. "Differential equations method: New technique for massive Feynman diagrams calculation". In: *Phys. Lett.* B254 (1991), pp. 158–164. DOI: 10.1016/0370-2693(91)90413-K.

[94] A. V. Kotikov. "Differential equation method: The Calculation of N point Feynman diagrams". In: *Phys. Lett.* B267 (1991), pp. 123–127. DOI: 10.1016/0370-2693(91)90536-Y.

[95] Z. Bern, L. J. Dixon, and D. A. Kosower. "Dimensionally regulated pentagon integrals". In: *Nucl. Phys.* B412 (1994), pp. 751–816. DOI: 10.1016/0550-3213(94)90398-0. arXiv: hep-ph/9306240 [hep-ph].

[96] J. M. Henn. "Lectures on differential equations for Feynman integrals". In: *J. Phys.* A48 (2015), p. 153001. DOI: 10.1088/1751-8113/48/15/153001. arXiv: 1412.2296 [hep-ph].

[97] J. M. Henn. "Multiloop integrals in dimensional regularization made simple". In: *Phys.Rev.Lett.* 110.25 (2013), p. 251601. DOI: 10.1103/PhysRevLett.110.251601. arXiv: 1304.1806 [hep-th].

[98] K.-T. Chen. "Iterated path integrals". In: *Bull.Am.Math.Soc.* 83 (1977), pp. 831–879. DOI: 10.1090/S0002-9904-1977-14320-6.

[99] M. Argeri et al. "Magnus and Dyson Series for Master Integrals". In: *JHEP* 03 (2014), p. 082. DOI: 10.1007/JHEP03(2014)082. arXiv: 1401.2979 [hep-ph].

[100] S. Caron-Huot and J. M. Henn. "Iterative structure of finite loop integrals". In: *JHEP* 06 (2014), p. 114. DOI: 10.1007/JHEP06(2014)114. arXiv: 1404.2922 [hep-th].

[101] T. Gehrmann et al. "The two-loop master integrals for $q\bar{q} \to VV$". In: *JHEP* 06 (2014), p. 032. DOI: 10.1007/JHEP06(2014)032. arXiv: 1404.4853 [hep-ph].

[102] M. Höschele, J. Hoff, and T. Ueda. "Adequate bases of phase space master integrals for gg $\to$ h at NNLO and beyond". In: *JHEP* 09 (2014), p. 116. DOI: 10.1007/JHEP09(2014)116. arXiv: 1407.4049 [hep-ph].

[103] R. N. Lee. "Reducing differential equations for multiloop master integrals". In: *JHEP* 04 (2015), p. 108. DOI: 10.1007/JHEP04(2015)108. arXiv: 1411.0911 [hep-ph].

[104] O. V. Tarasov. "Application and explicit solution of recurrence relations with respect to space-time dimension". In: *Nucl. Phys. Proc. Suppl.* 89 (2000). [,237(2000)], pp. 237–245. DOI: 10.1016/S0920-5632(00)00849-5. arXiv: hep-ph/0102271 [hep-ph].

[105] R. N. Lee. "Space-time dimensionality D as complex variable: Calculating loop integrals using dimensional recurrence relation and analytical properties with respect to D". In: *Nucl. Phys.* B830 (2010), pp. 474–492. DOI: 10.1016/j.nuclphysb.2009.12.025. arXiv: 0911.0252 [hep-ph].

[106] R. N. Lee, A. V. Smirnov, and V. A. Smirnov. "Analytic Results for Massless Three-Loop Form Factors". In: *JHEP* 04 (2010), p. 020. DOI: 10.1007/JHEP04(2010)020. arXiv: 1001.2887 [hep-ph].

[107] R. N. Lee and V. A. Smirnov. "Analytic Epsilon Expansions of Master Integrals Corresponding to Massless Three-Loop Form Factors and Three-Loop g-2 up to Four-Loop Transcendentality Weight". In: *JHEP* 02 (2011), p. 102. DOI: 10.1007/JHEP02(2011)102. arXiv: 1010.1334 [hep-ph].

[108] R. N. Lee and I. S. Terekhov. "Application of the DRA method to the calculation of the four-loop QED-type tadpoles". In: *JHEP* 01 (2011), p. 068. DOI: 10.1007/JHEP01(2011)068. arXiv: 1010.6117 [hep-ph].

[109] R. N. Lee, A. V. Smirnov, and V. A. Smirnov. "Master Integrals for Four-Loop Massless Propagators up to Transcendentality Weight Twelve". In: *Nucl. Phys.* B856 (2012), pp. 95–110. DOI: 10.1016/j.nuclphysb.2011.11.005. arXiv: 1108.0732 [hep-th].

[110] Wolfram Research, Inc. *Mathematica 10.0*. 2014. URL: https://www.wolfram.com.

[111] R. N. Lee and K. T. Mingulov. "Introducing SummerTime: a package for high-precision computation of sums appearing in DRA method". In: (2015). arXiv: 1507.04256 [hep-ph].

[112] D. H. Bailey H. R. P. Ferguson and S. Arno. "Analysis of PSLQ, an integer relation finding algorithm". In: *Math. Comp.* 68 (1999), pp. 351–369. DOI: 10.1090/S0025-5718-99-00995-3. eprint: 0803.1007.

[113] D. H. Bailey and D. J. Broadhurst. "Parallel integer relation detection: Techniques and applications". In: *Math.Comput.* 70 (2001), pp. 1719–1736. DOI: 10.1090/S0025-5718-00-01278-3. arXiv: math/9905048 [math-na].

[114] L. M. Milne-Thomson. *The calculus of finite differences*. London: Macmillan, 1933.

[115] Y. Schröder and A. Vuorinen. "High-precision epsilon expansions of single-mass-scale four-loop vacuum bubbles". In: *JHEP* 0506 (2005), p. 051. DOI: 10.1088/1126-6708/2005/06/051. arXiv: hep-ph/0503209 [hep-ph].

[116] S.A. Abramov and M.A. Barkatou. "D'Alembertian series solutions at ordinary points of LODE with polynomial coefficients". In: *Journal of Symbolic Computation* 44.1 (2009), pp. 48–59. DOI: 10.1016/j.jsc.2008.04.004. URL: http://dx.doi.org/10.1016/j.jsc.2008.04.004.

[117] J. Blümlein et al. "Determining the closed forms of the O(a**3(s)) anomalous dimensions and Wilson coefficients from Mellin moments by means of computer algebra". In: *Comput. Phys. Commun.* 180 (2009), pp. 2143–2165. DOI: 10.1016/j.cpc.2009.06.020. arXiv: 0902.4091 [hep-ph].

[118] J. Ablinger et al. "Modern Summation Methods and the Computation of 2- and 3-loop Feynman Diagrams". In: *Nucl. Phys. Proc. Suppl.* 205-206 (2010), pp. 110–115. DOI: 10.1016/j.nuclphysbps.2010.08.028. arXiv: 1006.4797 [math-ph].

[119] A.I. Davydychev and J.B. Tausk. "Two-loop self-energy diagrams with different masses and the momentum expansion". In: *Nuclear Physics B* 397.1-2 (1993), pp. 123–142. DOI: 10.1016/0550-3213(93)90338-p. URL: http://dx.doi.org/10.1016/0550-3213(93)90338-P.

[120] R. H. Lewis. *Computer algebra system Fermat.* URL: http://home.bway.net/lewis/.

[121] C. Bauer, A. Frink, and R. Kreckel. "Introduction to the GiNaC Framework for Symbolic Computation Within the C++ Programming Language". In: *J. Symb. Comput.* 33.1 (Jan. 2002), pp. 1–12. ISSN: 0747-7171. DOI: 10.1006/jsco.2001.0494. URL: http://dx.doi.org/10.1006/jsco.2001.0494.

[122] B. Haible, R. B. Kreckel. *CLN, a class library for numbers.* URL: http://www.ginac.de/CLN/cln.html.

[123] *SQLite, a relational database management system.* URL: https://www.sqlite.org/.

[124] S. Laporta and E. Remiddi. "The electron (g(e)-2) and the value of alpha: A check of QED at 1 ppb". In: *Acta Phys. Polon.* B28 (1997), pp. 959–977.

[125] M. Czakon, M. Awramik, and A. Freitas. "Bosonic corrections to the effective leptonic weak mixing angle at the two-loop level". In: *Nucl. Phys. Proc. Suppl.* 157 (2006). [,58(2006)], pp. 58–62. DOI: 10.1016/j.nuclphysbps.2006.03.036. arXiv: hep-ph/0602029 [hep-ph].

[126] W.G. Horner. In: *Trans. Soc. London* 109 (1819), pp. 308–335.

[127] K. Kajantie, M. Laine, and Y. Schröder. "A Simple way to generate high order vacuum graphs". In: *Phys. Rev.* D65 (2002), p. 045008. DOI: 10.1103/PhysRevD.65.045008. arXiv: hep-ph/0109100 [hep-ph].

[128] S. Groote, J. G. Körner, and A. A. Pivovarov. "On the evaluation of sunset - type Feynman diagrams". In: *Nucl. Phys.* B542 (1999), pp. 515–547. DOI: 10.1016/S0550-3213(98)00812-8. arXiv: hep-ph/9806402 [hep-ph].

[129] S. Groote, J. G. Körner, and A. A. Pivovarov. "On the evaluation of a certain class of Feynman diagrams in x-space: Sunrise-type topologies at any loop order". In: *Annals Phys.* 322 (2007), pp. 2374–2445. DOI: 10.1016/j.aop.2006.11.001. arXiv: hep-ph/0506286 [hep-ph].

[130] D. J. Broadhurst. "Evaluation of a Class of Feynman Diagrams for All Numbers of Loops and Dimensions". In: *Phys. Lett.* B164 (1985), p. 356. DOI: 10.1016/0370-2693(85)90340-5.

[131] D. J. Broadhurst and D. Kreimer. "Knots and numbers in Phi**4 theory to 7 loops and beyond". In: *Int. J. Mod. Phys.* C6 (1995), pp. 519–524. DOI: 10.1142/S012918319500037X. arXiv: hep-ph/9504352 [hep-ph].

[132] F. Brown and O. Schnetz. "Proof of the zig-zag conjecture". In: (2012). arXiv: 1208.1890 [math.NT].

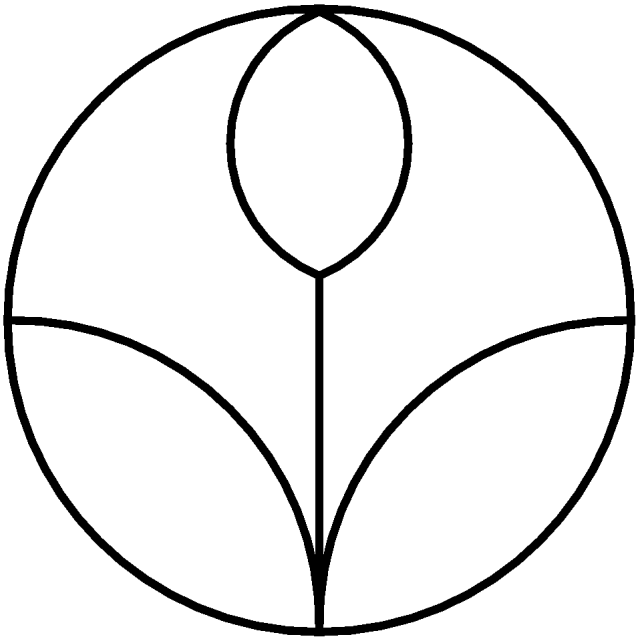
[133] B. G. Nickel. "Evaluation of Simple Feynman Graphs". In: *J. Math. Phys.* 19 (1978), pp. 542–548. DOI: 10.1063/1.523697.

[134] J. A. M. Vermaseren. "Axodraw". In: *Comput. Phys. Commun.* 83 (1994), pp. 45–58. DOI: 10.1016/0010-4655(94)90034-5.

Acknowledgements

I would like to extend my gratitude to the following people:

- York Schröder, for his great supervision of this thesis, his patience with my questions and his hospitality during my visits in Chile,
- Nicolas Borghini, for taking on the role as my official supervisor after York's departure and agreeing to examine this thesis,
- my family, for always being supportive,
- Isabel, Samae, Daniel, Florian and Song, for many lovely lunchtime discussions and distracting me from physics when needed,
- Isabel again, for proofreading this thesis$_{(,)}$ which is most appreciated,
- the people of D6/E6, especially my office mates Johanna, Mandy, Henning and Patric, for a great atmosphere to work in.



Declaration

I hereby affirm that this dissertation represents my own work and has not been previously submitted to any examination office. All resources used have been referenced.

Bielefeld, 13.08.15

Thomas Luthe