Advances in computational methods for Quantum Field Theory calculations

Ben Ruijl

ISBN: 978-94-6233-746-6
Printed by: Gildeprint, Enschede
Cover: thesisexpert.nl

SIKS Dissertation Series No. 2017-31. The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.
This work is supported by the ERC Advanced Grant no. 320651, “HEPGAME”.

Advances in computational methods for Quantum Field Theory calculations

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof. mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op donderdag 2 november 2017
klokke 15.00 uur

door

Ben Jacobus Gertruda Ruijl

geboren te Geleen
in 1989

Promotores:	Prof. dr. H.J. van den Herik
	Prof. dr. A. Plaat
Co-promotor:	dr. J.A.M. Vermaseren (Nikhef)
Promotiecommissie:	Prof. dr. J.N. Kok
	Prof. dr. A. Achúcarro
	Prof. dr. H.H. Hoos
	Prof. dr. J.W. van Holten
	Prof. dr. C. Anastasiou (ETH Zürich)
	Prof. dr. E.L.P.M. Laenen (Universiteit van Amsterdam)
	Prof. dr. A. Vogt (University of Liverpool)
	dr. G. Heinrich (MPP München)

Isn't it a noble, an enlightened way of spending our brief time in the sun, to work at understanding the universe and how we have come to wake up in it?

Richard Dawkins

PREFACE

This PhD thesis is the result of work I performed as part of the HEPGAME (High Energy Physics and Games) project. The HEPGAME project is supported by the ERC Advanced Grant no. 320651 and its primary goal is to apply methods from artificial intelligence to solve problems in high energy physics. The symbolic manipulation toolkit Form plays a key role in the investigation.

During the PhD I worked in three different locations. The first half year I mostly worked at Tilburg University. The next few years I have worked at Leiden University and Nikhef. At Leiden University, I taught a course, supervised a student, and worked on the computer science aspect of this thesis with my close colleagues Jaap van den Herik, Aske Plaat, and Ali Mirsoleimani. This mainly involved expression simplification, which turned into chapter 2.

At Nikhef, I focused on physics, and worked on designing computer programs with my close colleagues Jos Vermaseren and Takahiro Ueda. This resulted in the Forcer program, described in chapter 3. Later, Andreas Vogt joined to compute new physical quantities with Forcer. The results formed the basis of chapter 4.

Together with Franz Herzog we decided to aim for something we had not thought possible: computing the five-loop beta function. After a year of puzzling, we understood the R^*-method which could help us achieve our goal. The method is explained in chapter 5.

Finally, after combining the Forcer program and the computer code for the R^*-method, we were able to compute the five-loop beta function for Yang-Mills theory with fermions. This formed the basis for chapter 6.

I have had the pleasure to work with experts in the fields of Artificial Intelligence, Computer Science, and Theoretical Physics. This enriching experience has helped shape this thesis, for which I am grateful.

LIST OF ABBREVIATIONS

CSEE	Common Subexpression Elimination 11–15, 17, 35, 38
HEP	High Energy Physics 15, 16, 18–21, 24, 26–36
IBP	Integration by Parts 8, 39, 40, 42, 43, 45, 46, 48, 49, 52–55, 59, 60, 103, 128, 147, 161, 190
IR	Infrared 85, 103–110, 113, 118–127, 129–135, 181, 187, 188
IRR	Infrared Rearrangement 104, 118
LHC	Large Hadron Collider 4–6, 9, 39
LTVG	Logarithmic Tensor Vacuum Graph 105, 120, 123, 125, 127, 128, 133, 135
MCTS	Monte Carlo Tree Search 11, 12, 16–22, 35–37, 61, 161, 189, 190
QCD	Quantum Chromodynamics 3, 7–9, 39, 41, 80–82, 85–87, 90, 95, 97, 101, 104, 105, 114, 115, 129, 130, 135, 137, 138, 146, 150, 151, 153–157, 159, 160, 162, 190, 191, 195
QED	Quantum Electrodynamics 3, 7, 87, 137, 150–152, 159, 191, 195
QFT	Quantum Field Theory 3, 7–9, 39, 95, 103, 104, 129, 162, 189, 192
SA	Simulated Annealing 22, 25–27, 32–34
SA-UCT	Simulated Annealing - UCT 11, 12, 16, 18, 21–24, 36
SDD	Superficial Degree of Divergence 106, 116, 118, 120–123, 126, 129, 133
SHC	Stochastic Hill Climbing 22, 25–28, 30–37, 190
UCT	Upper Confidence bounds applied to Trees 11, 16, 17, 20–24, 36, 189
UV	Ultraviolet 103–109, 111, 113, 115, 118–135, 139, 185, 187, 188

CONTENTS

The material presented in this thesis is based on the following works.

1. B. Ruijl, T. Ueda and J.A.M. Vermaseren, *Forcer, a FORM program for the parametric reduction of four-loop massless propagator diagrams*, arXiv:1704.06650
2. B. Ruijl, T. Ueda, J.A.M. Vermaseren and A. Vogt, *Four-loop QCD propagators and vertices with one vanishing external momentum*, *JHEP* **2017** (2017) 1, http://dx.doi.org/10.1007/JHEP06(2017)040
3. F. Herzog and B. Ruijl, *The R^*-operation for Feynman graphs with generic numerators*, *JHEP* **2017** (2017) 37, http://dx.doi.org/10.1007/JHEP05(2017)037
4. F. Herzog, B. Ruijl, T. Ueda, J.A.M. Vermaseren and A. Vogt, *The five-loop beta function of Yang-Mills theory with fermions*, *JHEP* **02** (2017) 090 [arXiv:1701.01404]
5. J. Davies, A. Vogt, B. Ruijl, T. Ueda and J.A.M. Vermaseren, *Large-n_f contributions to the four-loop splitting functions in QCD*, *Nucl. Phys.* **B915** (2017) 335
6. F. Herzog, B. Ruijl, T. Ueda, J.A.M. Vermaseren and A. Vogt, *FORM, Diagrams and Topologies*, *PoS* **LL2016** (2016) 073 [arXiv:1608.01834]
7. T. Ueda, B. Ruijl and J.A.M. Vermaseren, *Forcer: a FORM program for 4-loop massless propagators*, *PoS* **LL2016** (2016) 070 [arXiv:1607.07318]
8. B. Ruijl, T. Ueda, J.A.M. Vermaseren, J. Davies and A. Vogt, *First Forcer results on deep-inelastic scattering and related quantities*, *PoS* **LL2016** (2016) 071, https://arxiv.org/abs/1605.08408
9. T. Ueda, B. Ruijl and J.A.M. Vermaseren, *Calculating four-loop massless propagators with Forcer*, *J. Phys. Conf. Ser.* **762** (2016) 012060 [arXiv:1604.08767]
10. B. Ruijl, T. Ueda and J. Vermaseren, *The diamond rule for multi-loop Feynman diagrams*, *Phys. Lett.* **B746** (2015) 347 [arXiv:1504.08258]
11. B. Ruijl, J. Vermaseren, A. Plaat and H.J. van den Herik, *Why Local Search Excels in Expression Simplification*, http://arxiv.org/abs/1409.5223, 2014, arXiv:1409.5223
12. B. Ruijl, J. Vermaseren, A. Plaat and J. van den Herik, *Problems and New Solutions in the Boolean Domain*, ch. Simplification of Extremely Large Expressions, p. 76. Cambridge Scholars Publishing, 2016
13. B. Ruijl, J. Vermaseren, A. Plaat and H.J. van den Herik, *HEPGAME and the Simplification of Expressions*, *Proceedings of the 11th International Workshop on Boolean Problems* (2014), http://arxiv.org/abs/1405.6369
14. B. Ruijl, J. Vermaseren, A. Plaat and H.J. van den Herik, *Combining Simulated Annealing and Monte Carlo Tree Search for Expression Simplification*, *Proceedings of ICAART Conference 2014* **1** (2014) 724 [arXiv:1312.0841], http://arxiv.org/abs/1312.0841

1 INTRODUCTION

Understanding how Nature works on a fundamental level is one of the key goals of physics. Consequently, physicists try to identify what the smallest building blocks of our universe are and how they interact. The ideas about these fundamental building blocks have undergone many revolutions, each radically changing the way we describe Nature.

The ancients reasoned that all objects were composed of the elements fire, earth, water, and air. As technology progressed, scientists discovered cells, molecules, and atoms. Atoms only have a radius of about 30 trillionths of a meter. For a while it was believed that the atom was the smallest component (the Greek name means *indivisible*). This idea lasted until the early 20th century with the discoveries of the electron and proton. Six decades later, it turned out that even protons and neutrons were not fundamental, but consisted of quarks [15, 16].

From the invention of quantum mechanics in the early 1920s, it became clear that these small particles behave differently from everyday experience: particles could be in two places at once, act like waves, or spontaneously emerge from the vacuum, and quickly disappear again [17]. The fact that the fundamental building blocks had both particle-like and wave-like features was later unified by the framework of Quantum Field Theory (QFT) [18]. The new quantum field theoretic description of the electromagnetic interaction, Quantum Electrodynamics (QED), was hugely successful and is still used to this day.

Below we provide a brief introduction to the world of QFT. We describe the Standard Model in section 1.1, the aim for precise predictions in section 1.2, computer methods in section 1.3, and Feynman diagrams in section 1.4. Then we formulate our Problem Statement (PS) in section 1.5, and the Research Questions (RQs) in section 1.6. In section 1.7 we list our contributions and in section 1.8 we outline the structure of the thesis.

1.1 THE STANDARD MODEL

In the early 1960s, the first version of the Standard Model was constructed [19]. The goal was to capture all fundamental particles and interactions in this model. The first version contained several particles, such as quarks and electrons, and the electromagnetic and weak forces. The electromagnetic force governs the interactions of photons (light) with charged particles. The weak forces govern nuclear decay and are mediated by the W and Z bosons. The Higgs boson, which is responsible for giving elementary particles mass, was added to the model shortly after, in 1964 [20–22]. Finally, the strong force, mediated by gluons, was added in 1973 [23, 24]. The theory for the strong interaction, Quantum Chromodynamics (QCD), explained why quarks of opposite charge can stay in a stable configuration in the nucleus of an atom. In figure 1 the modern-day Standard Model is displayed [25]. It contains six quarks, six leptons, and four force carriers. The shading signifies which particles are susceptible to which force. All non-fundamental particles in the universe are composed of these quarks. For example, the proton, which is part of the core of an atom, consists of two up quarks and a down quark.

The theories behind the Standard Model describe many properties of the particles, such as their charge and spin, and whether they have mass. However, the actual mass is not predicted by the Standard Model and has to be experimentally determined.

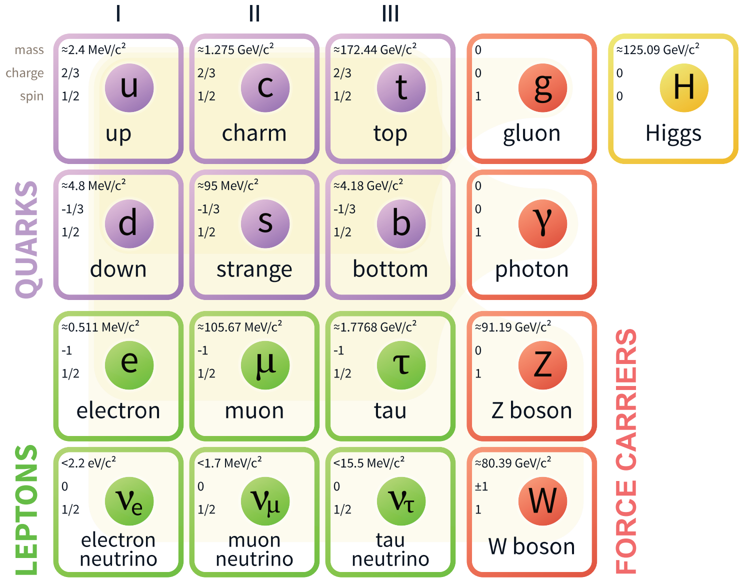


Figure 1: The Standard Model of elementary particles [25]. It contains 6 quarks and 6 leptons. The gluon is the force carrier of the strong force, the photon of the electromagnetic force, and Z/W bosons of the weak force. The shading signifies which particles are involved with which force. The Higgs particle is responsible for the mass of all massive particles.

Most features of the Standard Model are measured in particle accelerators, such as the Large Hadron Collider (LHC) [26]. At the LHC clusters of protons are accelerated in opposite directions in a 27 km long circular tunnel to almost the speed of light and are made to collide. From the energy of the collisions, new particles may be created, which are measured by detectors.

Over the last decades, many measurements have been performed that confirm the successful match of the Standard Model with Nature. The most famous measurement is the discovery of the Higgs boson in 2012 at the LHC [27, 28]. A second impressive measurement is the value of the fine-structure constant [29, 30] which governs the strength of the electromagnetic interaction. Theory and experiment are in agreement to within ten parts per billion. This staggering precision constitutes one of the best predictions of mankind.

1.2 Precise predictions

The Standard Model is extremely successful and has remained virtually unmodified for the last thirty years. So far, there have been no measurements that conclusively contradict the Standard Model. However, we know that it cannot be the final model, since it for instance does not include gravity or explain dark matter. The question then arises: how can we find physics beyond the Standard Model if we do not find discrepancies?

One of the reasons why experiments at the LHC may not find discrepancies is that the effects of new physics may only change the results of high-energy scattering experiments by a tiny fraction. For gravity this scenario is likely, since quantum gravity effects are believed to occur at the Planck scale, which is 10^{15} times higher than the energy that the LHC can produce. For dark matter, some candidates derived from Supersymmetry could be measured by the LHC, and those would lead to small discrepancies.

Therefore, extremely precise predictions are imperative to detect minuscule differences between what is expected and what is observed. Obtaining precise predictions will be the subject of the thesis.

1.3 Computer methods

Precise predictions are hard to compute by hand, since performing the algebra is a laborious exercise. With the rise of computers in the 1960s, software started to be developed to perform calculations for quantum field theories.

In 1963 the computer program Schoonschip was created by Veltman for the symbolic computation of the early Standard Model [31]. The program had to process an expression with 50 000 terms, which at the time could only be done by storing intermediate results on tape.

The development of the symbolic manipulation toolkit Form was started in 1984. It was designed to efficiently handle large expressions. A notable package was Mincer, originally developed for Schoonschip [32] and later ported to Form [33]. Using Mincer and Form, the evolution of unpolarized combinations of quark densities has been computed to next-to-next-to-leading order [34, 35].

Computer methods were also developed for numerical integration, which for complicated integrals is almost impossible to do by hand. Monte Carlo integrators, such as the popular Vegas [36] program, became the preferred way to compute complicated scattering processes.

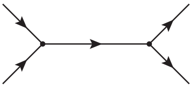
The fields of computer algebra and particle physics have developed hand in hand. Even though new computational methods have been designed and the hardware has had spectacular improvements, combinatorial and algorithmic challenges have not disappeared due to the need for more precision. In the 1960s a calculation involving proton interactions was considered a success if the order of magnitude agreed with the experiment. Nowadays, the goal is to achieve 1% accuracy. As a result, we will encounter expressions with billions of terms, taking up more than a terabyte of memory.

We are going to study some of these computational challenges. First, we will have a slightly more detailed look at the objects that we wish to compute.

1.4 FEYNMAN DIAGRAMS

In particle colliders such as the LHC, fast moving protons collide with each other. The new particles that are created are measured by detectors. Quantum Chromodynamics is the theory that describes these collisions, in the same way that Newton's equations describe how two macroscopic objects collide [37].

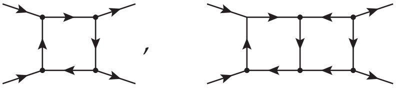
In Quantum Chromodynamics, a scattering process can be described in pictures called Feynman diagrams [38]. An example of a Feynman diagram is



, (1)

which signifies a process where two initial-state particles annihilate, creating a new particle, which then splits up into two new particles. Each particle is represented by a line, and each interaction by a vertex (a point where lines meet). The arrows represent whether the particle is moving to or from a vertex. The two final-state particles are measured by a detector.

One could imagine many more diagrams with two particles in the initial state and two in the final state. For example:



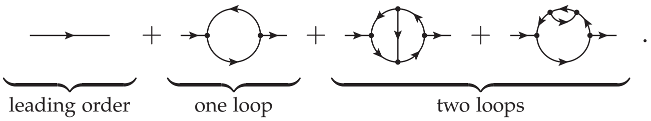
, , ... (2)

Each diagram with more particles (edges), more interactions (vertices), and more loops describes a more intricate process. From the rules of quantum mechanics, we know that in order to describe a scattering process, we need to take into account *all* the interactions that could happen in between [39]. This implies that we need to compute an infinite series of Feynman diagrams.

Computing the infinite series up to a certain number of loops leads to a reasonable estimate (i.e., the problem is suitable for perturbation theory). This is due to the fact that every interaction is suppressed by the interaction strength, called the 'coupling constant', which is smaller than one. Every diagram with the same number of loops has the same suppression factor. Contrary to what its name suggests, the coupling constant is actually dependent on the energy of the collision. For low-energy QED, the coupling constant is about $1/137$. For the experimentally viable energies of QCD, the coupling constant varies from 0.1 to 0.3. Consequently, for a two-loop process the relative suppression of QED over QCD is 10^4, which means that for QCD more diagrams have to be computed to obtain the same accuracy as for QED.

Below we consider a single particle moving in space-time (1 in and 1 out) expanded up to two loops:

$1 \to 1 =$



(3)

The number of possible diagrams increases exponentially with the number of loops. By simply constructing all possible graphs with vertices of degree three (where we enforce that a graph remains connected when one edge is removed), we see that at two loops we have two diagrams and at four loops 95 diagrams. If we also allow vertices with degree four, the number of diagrams at four loops is 1536.

For one-to-one reactions in QCD, the state-of-the-art is currently at four and five loop accuracy [2, 4, 40, 41]. For processes in QCD, each particle in the diagram could be of a different type (quark, gluon, etc.), which makes the number of diagrams much higher. Additionally, each individual diagram represents many fundamental integrals (also called scalar integrals). For example, at five loop accuracy one difficult diagram of the gluon propagator (with 160 500 diagrams) generates twelve million scalar integrals. This is an enormous number, which demands carefully constructed algorithms to be able to get an answer without running out of memory or time.

Having thus described the field of my research, we are now ready to state the overall problem statement of this thesis.

1.5 PROBLEM STATEMENT

Achieving higher precision, and thus going to a higher number of loops, creates at least three computational challenges: (1) billions of terms may be created, (2) the terms themselves may get large coefficients (fractions with more than a hundred digits), and (3) for some Feynman diagrams it is unknown how to compute them within a few months (or at all).

As we have reasoned in section 1.2, precise predictions are necessary to advance our understanding of Nature, which means these challenges have to be overcome. We now formulate our problem statement.

> **Problem statement:** *In what way can we improve the performance of QFT calculations with respect to obtaining more precise predictions?*

In this research we focus on three computational and combinatorial problems of QFT calculations that we deemed the most urgent ones.

Problem 1 Slowness of Monte Carlo integrations.

Monte Carlo integrators often take months on modern supercomputers to obtain high-quality results. This is caused by the large size of the input expression.

Problem 2 Slowness in the computation of massless propagator integrals.

Current methods based on Laporta-style algorithms [42] are notoriously slow. Often computations run for months or years on clusters with more than forty workstations.

Problem 3 Slowness when computing the singular parts of Feynman diagrams.

Computing the singular parts (poles) of diagrams is generally easier than computing the finite pieces. There exists a complicated R^* method that is not used often, in part due to combinatorial blow-up.

Below we derive three research questions, one for each problem.

1.6 THREE RESEARCH QUESTIONS

PROBLEM 1 Monte Carlo methods are frequently used in QFT calculations, since some integrals cannot be computed analytically. After the integrals are rewritten to a suitable polynomial representation, they are sampled millions of times. Some of these polynomials will have more than twenty variables and hundreds of thousands of terms, making evaluation very slow. Simplification of these polynomials will speed up integration. Thus, the first research question is as follows.

> **Research question 1 (RQ1):** *To what extent can the number of arithmetic operations of large multivariate polynomials be reduced?*

PROBLEM 2 Most integrals that can be computed analytically, are calculated by using Integration by Parts (IBP) identities to express integrals into simpler ones [43]. This method is generally quite slow and often requires months of computation time on a cluster. The Mincer program for three-loop massless propagator diagrams solves the IBP systems in a parametric way, which is more difficult to construct, but yields faster reductions [32, 33]. A four-loop equivalent of Mincer would mean that computations that currently take months, could be done in hours. Hence, our second research question reads as follows.

> **Research question 2 (RQ2):** *How can we construct a program that can compute four-loop massless propagator integrals more efficiently?*

PROBLEM 3 For five-loop calculations in QCD, so far only the poles of integrals have been computed, as the finite pieces are too difficult [40, 44]. There are several methods involving infrared rearrangement that allows one to compute the poles of five-loop integrals using only four-loop integral computations. One of these is the

R^*-operation [45], which is complicated and quite slow. Hence, we formulate our third research question as follows.

> **Research question 3 (RQ3):** *To what extent can we compute the poles of five-loop diagrams using only four-loop diagrams more efficiently?*

1.7 RESEARCH METHODOLOGY AND FOUR CONTRIBUTIONS

The four main contributions of this thesis are listed below. After each bullet, we summarise the research methodology.

- The improvement of expression simplification for large expressions.

  We provide an algorithm that simplifies large expressions faster than the state-of-the-art and with slightly improved quality. The simplifications result in a speed-up of Monte Carlo integrations [11].

- The construction of Forcer, a program to compute massless four-loop propagators.

  The Forcer program has already been used in many large calculations, including the computation of four-loop Mellin moments of structure functions. These quantities are important ingredients for processes such as Higgs production at the LHC. So far, the calculations of these objects could not be performed in a reasonable amount of time by any other existing program [1].

- The generalisation of the R^*-operation to integrals with generic numerator structure.

  The R^*-operation can be used to compute the poles of Feynman diagrams using an infrared-rearranged version of the diagram. We have extended the R^*-operation to include Feynman diagrams with arbitrary numerator structure, which makes the method more suitable for practical use. We have written a computer code that can compute the poles of five-loop massless propagator diagrams efficiently [3].

- The computation of the five-loop beta function for Yang-Mills theory with fermions.

  The five-loop beta function of QCD describes how the strength of the strong coupling evolves with the energy scale. Precise determination is relevant for many processes, as well as for studying theoretical properties of QFTs. We have verified the QCD beta function presented in [40] and generalized the result to an arbitrary colour group. Our more general computation took only six days on a single 32-core machine using the methods developed in this thesis [4]. The computation in [40] took 1.5 years on 20 computers with 8 cores each.

1.8 STRUCTURE OF THE THESIS

In chapter 1 we formulated the problem statement, stated the three research questions, and alluded to the contributions of this thesis. In chapter 2 we will address **RQ1**. Chapter 3 is devoted to **RQ2**. Next, chapter 4 shows new four-loop results, obtained by answering **RQ2**. Chapter 5 addresses **RQ3**. Finally, in chapter 6 we compute the five-loop beta function for Yang-Mills theory with fermions, which was only possible by using the methods developed to answer **RQ2** and **RQ3**.

2 EXPRESSION SIMPLIFICATION

Monte Carlo integration is the preferred way to compute complicated cross sections of scattering processes [46–49]. The disadvantage of this method is that it converges slowly and thus requires a large number of samples. Sampling large multivariate polynomials is very time consuming, which makes Monte Carlo methods slow. In this chapter we investigate these challenges by answering

> **RQ1**: *To what extent can the number of arithmetic operations of large multivariate polynomials be reduced?*

The expressions that arise from Quantum Chromodynamics (QCD) are polynomials in many variables. The number of variables could range from a few to several hundreds. Analogously, the number of terms range from ten thousand terms to millions of terms [50]. In some extreme cases, the executable code that performs the evaluation of the expression could take up a few gigabytes. If we are able to reduce the number of operations required to evaluate these expressions, sampling becomes faster. As a result, Monte Carlo integrators can obtain precise results much faster.

We describe two methods to reduce the number of operations. The first is Horner's rule for multivariate polynomials, which is extracting variables outside brackets [51] (sec. 2.1.1). For multivariate expressions the order of these variables is called a *Horner scheme*. The second is called Common Subexpression Elimination (CSEE) [52], which is performed after the Horner scheme has been applied (sec. 2.1.2). We will investigate four methods (named H1 to H4) of finding a Horner scheme that yields a near-minimal number of operations after (1) the Horner scheme and (2) CSEE have been applied. The first three are based on tree search algorithms, and the fourth is based on local search algorithms.

H1. Following recent successes in expression simplification, we investigate tree search methods such as Monte Carlo Tree Search (MCTS), using Upper Confidence bounds applied to Trees (UCT) as best-child criterion [53] (sec. 2.3). However, UCT is not straightforward, as (1) it introduces an exploration-exploitation constant C_p that must be tuned, and (2) it does little exploration at the bottom of the tree.

H2. The second method is Nested Monte Carlo Search (NMCS) [54], described in sec. 2.3.2. NMCS does not not have the two issues of MCTS+UCT. However, since our evaluation function is quite expensive (3 seconds for one of our benchmark polynomials), NMCS performs (too) many evaluations to find a path in the tree, rendering it unsuitable for our simplification task.

H3. We make a modification to UCT (sec. 2.3.3), which we call Simulated Annealing - UCT (SA-UCT). SA-UCT introduces a dynamic exploration-exploitation parameter $T(i)$ that decreases linearly with the iteration number i. SA-UCT causes a gradual shift from exploration at the start of the simulation to exploitation at the end. As a consequence, the final iterations will be used for exploitation, improving their solution quality. Additionally, more branches reach the final states, resulting in more exploration at the bottom of the tree. Moreover, we show that the tuning of C_p has become easier, since the region with appropriate values for C_p has increased by at least a tenfold [14]. The main contribution of SA-UCT is that this simplification of tuning allows for the results of our MCTS approach to be obtained much faster.

H4. We study local search methods (sec. 2.4). We find that the state space of Horner schemes is ideally suited for local search methods, since it is relatively flat and contains few local minima. This allows us to simplify expressions without expensive tuning of parameters. Our final algorithm is a variant of Stochastic Hill Climbing, which requires about ten times fewer samples than SA-UCT for similar results.

Our final Stochastic Hill Climbing algorithm is able to reduce the computation time of numerical integration from weeks to days or even hours. The methods have been implemented in Form and are used by at least one other research group.

The remainder of this chapter is structured as follows. First, we discuss methods for expression simplification in section 2.1. In section 2.2 we discuss the experimental setup. Next, we discuss tree search methods (method H1, H2, H3) in section 2.3. In section 2.4 we examine local search methods (method H4). We discuss performance and results in section 2.5. Finally, we present the chapter conclusion in section 2.6.

2.1 horner schemes and common subexpression elimination

Expression simplification is a widely studied problem. Some examples are Horner schemes [52], common subexpression elimination (CSEE) [55], partial syntactic factorisation [56] and Breuer's growth algorithm [57]. Much research is put into simplifications using more algebraic properties, such as factorisation, especially because of its interest for cryptographic research [58, 59]. Simplification methods that depend on factorisation have the major problem of being notoriously slow. Horner schemes and CSEE do not require sophisticated mathematics: only the commutative and associative properties of the operators are used. The expressions we are considering often have more than twenty variables and more than a hundred thousand terms [50]. In this regime, computationally expensive methods are infeasible. Therefore, we consider using basic methods such as Horner schemes and CSEE.

2.1.1 *Horner Schemes*

Horner's rule reduces the number of multiplications in an expression by lifting variables outside brackets [51, 52, 60]. For multivariate expressions Horner's rule can be applied sequentially, once for each variable. The order of this sequence is called the *Horner scheme*. Take for example:

$$x^2z + x^3y + x^3yz \to x^2(z + x(y(1 + z))) \; . \tag{4}$$

Here, first the variable x is extracted (i.e., x^2 and x) and second, y. The number of multiplications is now reduced from 8 to 4. However, the order x, y is chosen arbitrarily. One could also try the order y, x:

$$x^2z + x^3y + x^3yz \to x^2z + y(x^3(1 + z)) \; , \tag{5}$$

for which the number of multiplications is 6. Evidently, this is a sub-optimal Horner scheme. There are $n!$ orders of extracting variables, where n is the number of variables, and it turns out that the problem of selecting an optimal ordering is NP-hard [60].

A heuristic that works reasonably well is selecting variables according to how frequently a term with such a variable occurs ("occurrence order") [53]. A counter-example that shows that occurrence order is not always optimal is

$$x^{50}y + x^{40} + y + yz \; , \tag{6}$$

where extracting the most occurring variable y first causes the x^{50} and x^{40} to end up in different subparts of the polynomial, preventing their common terms from being extracted. We note that ordering the variables according to its highest power or to the sum of its powers in all the terms leads to other counter-examples.

2.1.2 *Common subexpression elimination*

The number of operations can be reduced further by applying common subexpression elimination (CSEE). This method is well known from the fields of compiler construction [55] and computer chess [61], where it is applied to much smaller expressions or subtrees than what we are considering here. Figure 2 shows an example of a common subexpression in a tree representation of an expression. The shaded expression $b(a + e)$ appears twice, and its removal means removing one superfluous addition and one multiplication.

CSEE is able to reduce both the number of multiplications and the number of additions, whereas Horner schemes are only able to reduce the number of multiplications.

2.1.3 *The evaluation function*

Writing an efficient evaluation function is important, since this function gets called many times. It consists of two parts: (1) applying the Horner scheme to an expression, and (2) removing common subexpressions. The Horner scheme is applied to the expression in Form's internal format, which is a linear representation of a polynomial.

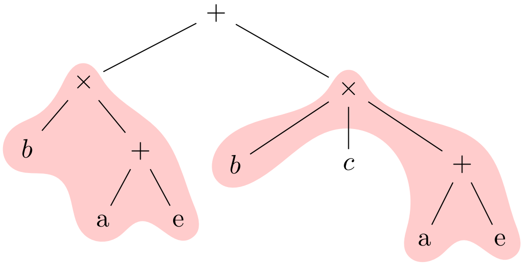


Figure 2: A common subexpression (shaded) in an associative and commutative tree representation.

In order to compute subexpressions efficiently, we transform the expression from a linear internal format to a tree. While building the expression tree (similar to figure 2), we store a hash of the branch that starts at the current node. Since the tree is built from the bottom up, we combine the hashes of the two subnodes. When the tree is completely built, we try to find common subexpressions. We keep track of each subtree we have come across in a hashtable. If a node is found, we skip exploring the subnodes, since it is a common subexpression. If we have not found the node before, we add the appropriate number of operations, add the node to the hashtable (the hash for the entire subtree is easily and quickly retrieved from the node itself), and continue to its children. A top-down approach has the benefit that entire subtrees can be easily identified as common subexpressions and double searches can be prevented. This process outweighs the time it takes to build the tree[1].

We apply two improvements to increase the number of common subexpressions: (1) when we extract the Horner scheme variable, we also extract a greatest common divisor (gcd). From our measurements, we have seen that this exposes more common subexpressions. (2) We store common subexpressions for exponentiations as well:

$$x^{20} = (x^{10})^2 = ((x^5)^2)^2 = x(((x^2)^2)^2)^2 \,. \tag{7}$$

2.1.4 *Interplay*

We note that there is an interplay between Horner schemes and CSEE: a certain “optimal” Horner scheme may reduce the number of multiplications the most, but may expose fewer common subexpressions than a “mediocre” Horner scheme. Thus, we need to find a way to obtain a Horner scheme that reduces the number of operations the most after both Horner and CSEE have been applied.

Finding appropriate Horner schemes is not a trivial task, for at least three reasons. First, there are no known local heuristics. For the Travelling Salesman Problem (TSP), the distance between two cities can be used as a heuristic [62], and more specialised

[1] The tree can be allocated at once, since a close upper bound to the number of nodes is known.

	variables	terms	operations	eval. time (s)
res(7,4)	13	2561	29 163	0.001
res(7,5)	14	11 379	142 711	0.03
res(7,6)	15	43 165	587 880	0.13
res(9,8)	19	4 793 296	83 778 591	25.0
HEP(σ)	15	5716	47 424	0.008
HEP(F_{13})	24	105 058	1 068 153	0.4
HEP(F_{24})	31	836 009	7 722 027	3.0
HEP(b)	107	193 767	1 817 520	2.0

Table 1: The number of variables, terms, operations, and the evaluation time of applying a single Horner scheme and CSEE in seconds, for our eight (unoptimised) benchmark expressions. The time measurement is performed on a 2.4 GHz Xeon computer. All expressions fit in memory (192 GB).

heuristics are able to solve symmetric TSP instances with thousands of cities (a historic example is a TSP with 7397 cities [63, 64]). Second, the Horner scheme is applied to an expression. This means that the scheme has a particular context: the nth entry applies to the subexpressions that are created after the first $n-1$ entries in the Horner scheme have been applied to the expression. Third, the evaluation of a Horner scheme and CSEE is slow: for some benchmark expressions the evaluation took multiple seconds on a 2.4 GHz computer (see table 1). Since the evaluation is so slow, we have to find an optimisation algorithm that performs well with only a limited number of samples. Our attempted parallelisation of the evaluation function was unsuccessful, since the Horner scheme evaluation function is too fine-grained.

The time it takes to apply a Horner scheme is directly related to the number of variables and the number of terms in the expression. The common subexpression elimination time scales linearly with the number of operations. The difficulty of finding a good Horner scheme is related to (1) the size of the permutation space, i.e., related to the number of variables, but also to (2) the distribution of the variables in the terms. The composition of the variables affects the flatness of the state space and the occurrence of saddle points and local minima, as we shall see in section 2.4.3.

2.2 EXPERIMENTAL SETUP

We use eight large benchmark expressions, four from mathematics and four from real-world High Energy Physics (HEP) calculations. In table 1 statistics for the expressions are displayed. We show the number of variables, terms, operations, and the evaluation time of applying a Horner scheme and CSEE.

The expressions called res(7,4), res(7,5), res(7,6), and res(9,8) are resolvents and are defined by $\text{res}(m,n) = \text{res}_x(\sum_{i=0}^{m} a_i x^i, \sum_{i=0}^{n} b_i x^i)$, as described in [56]. The number of variables is $m+n+2$. The polynomial res(9,8) is the largest polynomial we have tested and has been included to test the boundaries of our hardware.

The High Energy Physics expressions represent scattering processes for the future International Linear Collider, a likely successor to the Large Hadron Collider [50]. A standard method of calculating the probability of certain collision events is by using perturbation theory. As a result, for each order of perturbations, additional expressions are calculated as corrections to previous orders of precision. The HEP polynomials of table 1 are second-order corrections to various processes.

HEP(σ) describes parts of the process $e^+e^- \to \mu^+\mu^-\gamma$, namely the collision of an electron and positron that creates a muon, an anti-muon, and a photon.

HEP(F_{13}), HEP(F_{24}), and HEP(b) are obtained from the process $e^+e^- \to \mu^+\mu^-u\bar{u}$, namely the collision of an electron and positron that creates a muon, anti-muon, an up-quark, and an up-antiquark. The results can be used to obtain next-generation precision measurements for electron-positron scattering [50].

These four HEP polynomials represent classes of polynomials with approximately the same behaviour.

2.3 TREE SEARCH METHODS

In this section we investigate three tree search methods. First, we review MCTS combined with UCT and discuss some issues in section 2.3.1. Second, we consider Nested Monte Carlo Search (NMCS) in section 2.3.2. Third, we construct a new best-child criterion called SA-UCT in section 2.3.3.

2.3.1 *Monte Carlo Tree Search*

Recently, Monte Carlo Tree Search has been shown to yield good quality Horner schemes [53]. We will describe its characteristics, so that we can see if we can improve its performance.

Monte Carlo Tree Search (MCTS) is a tree search method that has been successful in games such as Go, Hex, and other applications with a large state space [65, 66]. It works by selectively building a tree, expanding only branches it deems worthwhile to explore. MCTS consists of four steps, which are displayed in figure 3. The first step (3a) is the selection step, where a leaf or a not fully expanded node is selected according to some criterion (see below). Our choice is node z. In the expansion step (3b), a random unexplored child of the selected node is added to the tree (node y). In the simulation step (3(c)), the rest of the path to a final node is completed using random child selection. Finally a score Δ is obtained that signifies the score of the chosen path through the state space. In the backpropagation step (3d), this value is propagated back through the tree, which affects the average score (winrate) of a node (see below). The tree is built iteratively by repeating the four steps.

In the game of Go, each node represents a player move and in the expansion phase the game is played out, in basic implementations, by random moves. In the best performing implementations heuristics and pattern knowledge are used to complement a random playout [65]. The final score is 1 if the game is won, and 0 if the game is lost. The entire tree is built, ultimately, to select the best first move.

For our purposes, we need to build a complete Horner scheme, variable by variable. As such, each node will represent a variable and the depth of a node in the tree represents the position in the Horner scheme. Thus, in figure 3(c) the partial Horner scheme is x,z,y and the rest of the scheme is filled in randomly with unused variables. The score of a path in our case, is the improvement of the path on the number of operations: the original number of operations divided by the number of operations after the Horner scheme and CSEE have been applied. We note that for our purposes the *entire* Horner scheme is important and not just the first variable.

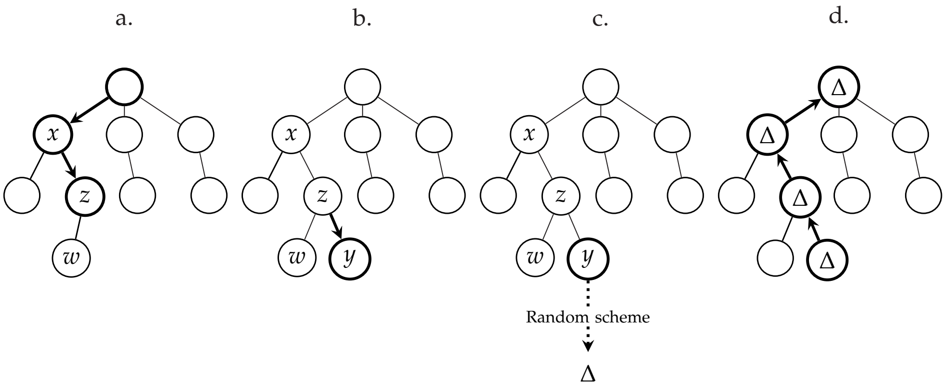


Figure 3: An overview of the four phases of MCTS: selection (a), expansion (b), simulation (c), and backpropagation (d). The selection of a not fully expanded node is done using the best child criterion (in our case UCT). Δ is the number of operations left in the final expression, after the Horner scheme and CSEE have been applied. See also [67].

In many MCTS implementations UCT (formula 8) is chosen as the selection criterion [67, 68]: [2]

$$\underset{\text{children } c \text{ of } s}{\operatorname{argmax}} \; \bar{x}(c) + 2C_p\sqrt{\frac{2\ln n(s)}{n(c)}}\,, \tag{8}$$

where c is a child node of node s, $\bar{x}(c)$ the average score of node c, $n(c)$ the number of times the node c has been visited, C_p the exploration-exploitation constant, and argmax the function that selects the child with the maximum value. This formula balances exploitation, i.e., picking terms with a high average score, and exploration, i.e., selecting nodes where the child has not been visited often compared to the parent. The C_p constant determines how strong the exploration term is: for high C_p the focus will be on exploration, and for low C_p the focus will be on exploitation.

There are two issues with the current form of the MCTS algorithm. The first was already mentioned and involves the tuning of the C_p parameter. Sometimes

[2] The factor two outside and inside the square root of the UCT formula are a convention.

the region of C_p that yields good values is small, thus it may be computationally expensive to find an appropriate C_p. We return to this issue in section 2.3.3. Now we focus on the second issue: due to the natural asymmetry of trees, there is more exploration at nodes close to the root compared to the nodes deeper in the tree. Moreover, only a few branches are fully expanded to the bottom. Consequently, the final variables in the scheme will be filled out with the variables of a random playout. No optimisation is done at the end of these branches. As a result, if a very specific order of moves at the end of the tree gives a better outcome, this solution will not be found by MCTS. The issue can be partially reduced by adding a new parameter that specifies whether the Horner scheme should be constructed in reverse, so that the variables selected near the root of the tree are actually the last to be extracted [14, 53].

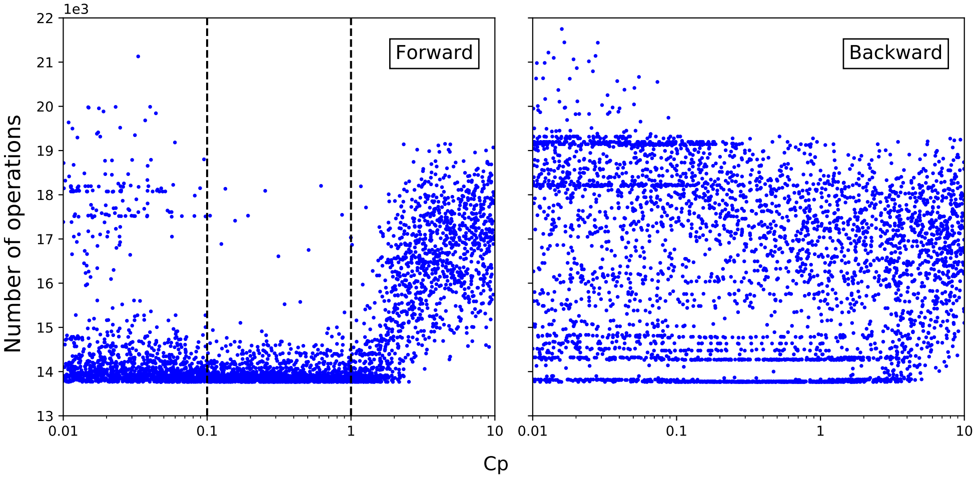


Figure 4: res(7,5): differences between forward (left) and backward (right) Horner schemes, at $N = 1000$ tree updates with SA-UCT. Forward Horner schemes generate a region of C_p where the number of operations is near the global minimum, whereas backward schemes have multiple high-density local minima and a diffuse region.

In figure 4 the difference between a forward and a backward MCTS search with 1000 updates is shown for the polynomial res(7,5) in scatter plot. For the forward construction, we see that there is a region in C_p where the results are good: the lowest measured value is found often. However, the backward scheme does not have a similar range. For other polynomials, it may be better to use the backward scheme, as is the case for HEP(σ) and HEP(F_{13}). Currently, there is no known way to predict whether forward or backward construction should be used. Thus, this introduces an extra parameter to our algorithm.

Even though the scheme direction parameter reduces the problem somewhat, the underlying problem that there is little exploration at the end of the tree still remains.

To overcome the issues of tuning C_p and the lack of exploration, we study a related tree search method called Nested Monte Carlo Search in the next section.

2.3.2 *Nested Monte Carlo Search*

Nested Monte Carlo Search (NMCS) addresses the issue of the exploration bias towards the top of the tree by sampling all children at every level of the tree [54]. In its simplest form, called a level-1 search, a random playout is performed for each child of a node. Next, the child with the best score is selected, and the process is repeated until one of the end states is reached. This method can be generalized to a level k search, where the above process is nested: a level k search chooses the best node from a level $k-1$ search performed on its children. Thus, if the NMCS level is increased, the top of the tree is optimised with greater detail. Even though NMCS makes use of random playouts, it does so at every depth of the tree as the search progresses. Consequently, there is always exploration near the end of the tree.

In figure **??** the results for level 2 NMCS are shown for HEP(σ). The average number of operations is 4189 ± 43 (a standard deviation of 43). To compare the performance of NMCS to that of MCTS, we study the run-time. Since more than 90% of the run-time is spent on the evaluation function, we may compare the number of evaluations instead. A level-2 search for HEP(σ) takes 8500 evaluations. In order to be on a par with MCTS, the score should have been between MCTS 1000 and MCTS 10 000 iterations. However, we see that the score is higher than MCTS with 1000 iterations and thus we may conclude that the performance of NMCS is inferior to MCTS for HEP(σ).

We have performed similar experiments with NMCS on other polynomials, but the resulting average number of operations were always greater than MCTS's. The reason is likely because we select a low level k: a level-1 search selects the best child using one sample per child, a process which is highly influenced by chance. However, there are some performance issues with using a higher k. To analyse these, we investigate the number of evaluations that a level n search requires.

The form of our tree is known, since every variable should appear only once. This means that there are n children at the root, $n-1$ children of children, and $n-d$ children at depth d. Thus a level-1 search takes $n + (n-1) + (n-2) + \ldots + 1 = n(n+1)/2$ evaluations. It can be shown that a level k search takes S_n^{k+n}, where S is the Stirling Number of the First Kind. This number grows rapidly: if $k = 1$ and $n = 15$, the number of evaluations is 120, and for level $k = 2$, it takes 8500 evaluations. For an expression with 100 variables, a level-1 search takes 5050 evaluations, and a level-2 search takes 13 092 125 evaluations.

The evaluation function is expensive for our polynomials: HEP(F_{13}) takes about 0.4 second per evaluation and HEP(F_{24}) takes 3.0 seconds. We have experimented with paralellising the evaluation function, but due to the fine-grained nature of the evaluation function, this was unsuccessful. For HEP(F_{24}) a million iterations will be slow, hence for practical reasons we have only experimented with a level-1 search.

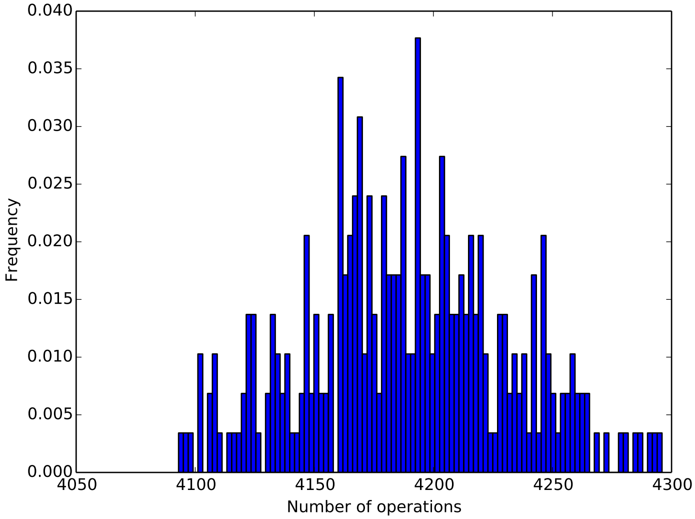


Figure 5: NMCS level 2 for HEP(σ), taking 8500 evaluations. This is comparable in CPU time to MCTS with 8500 runs. The number of operations is 4189 ± 43, averaged over 292 samples.

The domains in which NMCS performed well, such as Morpion Solitaire and SameGame, have a cheap evaluation function relative to the tree construction [54]. If the evaluation function is expensive, even the level-1 search takes a long time.

Based on the remarks above, we may conclude that for polynomials with a large number of variables, NMCS becomes infeasible.

2.3.3 *SA-UCT*

Since NMCS is unsuitable for simplifying large expressions, we return our focus to MCTS, but this time on the UCT best child criterion. We now consider the role of the exploration-exploitation constant C_p. We notice that in the first iterations of the simulation there is as much exploration as there is in the final iterations, since C_p remains constant throughout the search. For example, the final 100 iterations of a 1000 iterations MCTS run are used to explore new branches even though we know in advance that there is likely not enough time to reach the final nodes. Thus we would like to modify the C_p to change during the simulation to emphasise exploration early in the search and emphasise exploitation towards the end.

We introduce a new, dynamic exploration-exploitation parameter T that decreases linearly with the iteration number:

$$T(i) = C_p \frac{N - i}{N} \,, \tag{9}$$

where i is the current iteration number, N the preset maximum number of iterations, and C_p the initial exploration-exploitation constant at $i = 0$.

We modify the UCT formula to become:

$$\underset{\text{children } c \text{ of } s}{\text{argmax}} \;\; \bar{x}(c) + 2T(i)\sqrt{\frac{2\ln n(s)}{n(c)}}\;, \tag{10}$$

where c is a child of node s, $\bar{x}(c)$ is the average score of child c, $n(c)$ the number of visits at node c, and $T(i)$ the dynamic exploration-exploitation parameter of formula (9).

The role of T is similar to the role of the temperature in Simulated Annealing [69]: in the beginning of the simulation there is much emphasis on exploration, the analogue of allowing transitions to energetically unfavourable states. During the simulation the focus gradually shifts to exploitation, analogous to annealing. Hence, we call our new UCT formula "Simulated Annealing UCT (SA-UCT)".

In the past related changes have been proposed. For example, Discounted UCB [70] and Accelerated UCT [71] both modify the average score of a node to discount old wins over new ones. The difference between our method and past work is that the previous modifications alter the importance of exploring based on the history and do not guarantee that the focus shifts from exploration to exploitation. In contrast, this work focuses on the exploration-exploitation constant C_p and on the role of exploration during the simulation.

We implemented four improvements over UCT. (1) The final iterations are used effectively. (2) There is more exploration in the middle and at the bottom of the tree. This is due to more nodes being expanded at lower levels, because the T is lowered. As a consequence, we see that (3) more branches reach the end states. As a result, (4) there is exploration near the bottom, where there was none for the random playouts.

In order to analyse the effect of SA-UCT on the fine-tuning of C_p (the initial temperature), we perform a sensitivity analysis on C_p and N [14]. In figure 6 the results for the res(7,5) polynomial with 14 variables are displayed. Horizontally, we have C_p, and vertically we have the number of operations (where less is better). A total of 4000 MCTS runs (dots) are performed for a C_p between 0.001 and 10. On the left we show the results for UCT and on the right for SA-UCT. We identify a region with local minima for low C_p, a diffuse region for high C_p and an intermediate region in C_p where good results are obtained. This region becomes wider if the number of iterations N increases, for both SA-UCT and UCT.

However, we notice that the intermediate region is wider for SA-UCT, compared to UCT. For $N = 1000$, the region is $[0.1, 1.0]$ for SA-UCT, whereas it is $[0.07, 0.15]$ for UCT. Thus, SA-UCT makes the region of interest about 11 times larger for res(7,5). This stretching is not just an overall rescaling of C_p: the uninteresting region of low C_p did not grow significantly. For $N = 3000$, the difference in width of the region of interest is even larger.

In figure 7, we show a similar sensitivity analysis for HEP(σ) with 15 variables. We identify the same three regions and see that the region of interest is $[0.5, 0.7]$ for

UCT and $[0.8, 5.0]$ for SA-UCT at $N = 1000$. This means that the region is about 20 times larger relative to the uninteresting region of low C_p, which grew from 0.5 to 0.8. We have performed experiments on six other expressions, and we obtain similar results [14].

On the basis that SA-UCT compared to UCT (1) decreases the sensitivity to C_p by an order of magnitude, and (2) produces Horner schemes of the same quality, we may conclude that SA-UCT reduces the fine-tuning problem without overhead.

2.4 STOCHASTIC LOCAL SEARCH

In the preceding sections we have concluded that MCTS with SA-UCT is able to find Horner schemes that yield a smaller number of operations than the naive occurrence order schemes. For some polynomials, MCTS yields reductions of more than a factor 24. However, in section 2.3.1 we have seen that there are some intrinsic shortcomings to using a tree representation, especially if the depth of the search tree becomes (too) large. We noticed that many branches do not reach the bottom when there are more than 20 variables (we remind the reader that the problem depth is equivalent to the number of variables) as is the case with many of our expressions. MCTS determines the scores of a branch by performing a random play-out. If the branch is not constructed all the way to the bottom, the final nodes are therefore random (no optimisation). For Horner schemes, the *entire* scheme is important, so sub-optimal selection of variables at the end of the scheme can have a significant impact.

The issue of poor optimisation at the bottom of the tree motivated us to look for a method that is symmetric in its optimisation: both the beginning and the end should be optimised equally well. In this section we (re)consider which class of algorithms is best suited for the Horner scheme problem. The Horner scheme problem belongs to the class of permutation problems. Many algorithms for optimising permutation problems have been suggested in the literature, such as Stochastic Hill Climbing (SHC) [72], Simulated Annealing (SA) [69], Tabu Search [62], Ant Colony Optimisation [73], and Evolutionary Algorithms [74]. Since measurements take weeks per algorithm, we limit ourselves to two. In [11] we provide qualitative motivations for focusing on SHC and SA. In summary, the absence of heuristics and the high cost of sampling to tune parameters make the other options less interesting.

The structure of section 2.4 is as follows. In section 2.4.1 we consider the differences between SHC and SA, in section 2.4.2 we find an appropriate neighbourhood structure, and in 2.4.3 we study the state space properties for given neighbourhood structures.

2.4.1 *SHC versus SA*

A Stochastic Hill Climbing procedure, also known as iterative improvement local search, has two parameters: (1) the number of iterations N, and (2) the neighbourhood structure, which defines the transition function [75]. We consider the

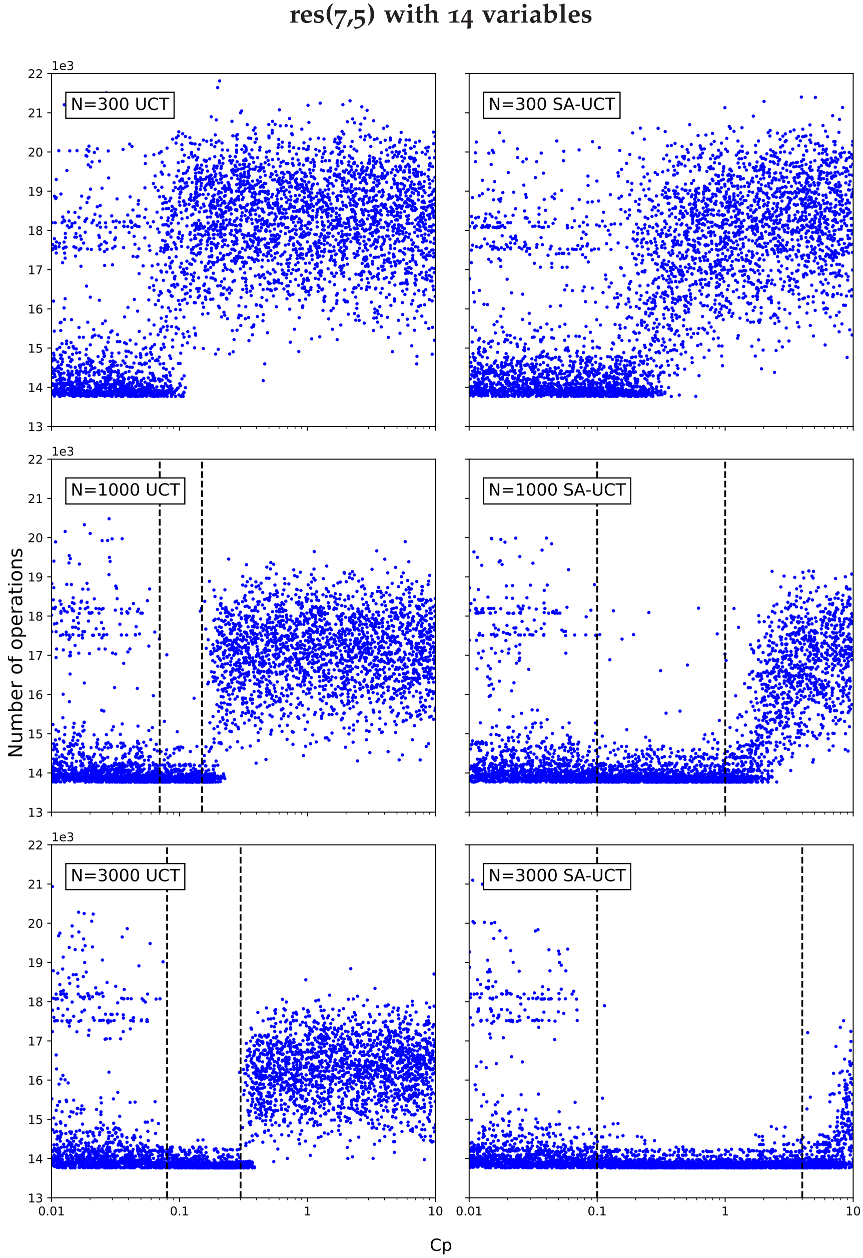


Figure 6: res(7,5) polynomial with 14 variables: on the x-axis we show C_p and on the y-axis the number of operations. A lower number of operations is better. On the left, we show UCT with constant C_p and on the right we show SA-UCT where C_p is the starting value of T. Vertically, we increase the tree updates N from 300, to 1000, to 3000. As indicated by the dashed lines, an area with an operation count close to the global minimum appears, as soon as there are sufficient tree updates N. This area is wider for SA-UCT than for UCT.

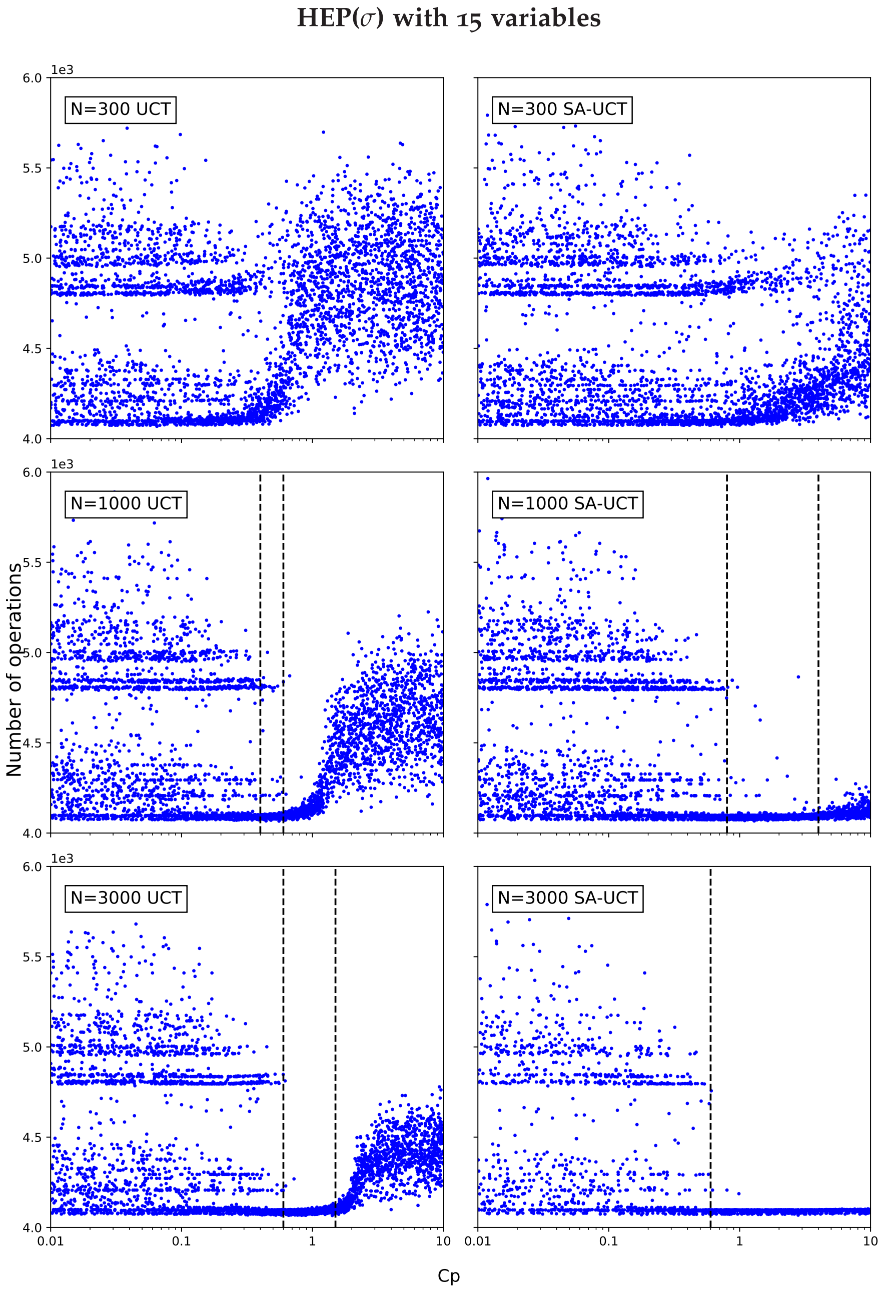


Figure 7: HEP(σ) with 15 variables: on the x-axis we show C_p and on the y-axis the number of operations. A lower number of operations is better. On the left, we show UCT with constant C_p and on the right we show SA-UCT where C_p is the starting value of T. Vertically, we increase the tree updates N from 300, to 1000, to 3000. As indicated by the dashed lines, an area with an operation count close to the global minimum appears, as soon as there are sufficient tree updates N. This area is wider for SA-UCT than for UCT.

neighbourhood structure the most important parameter, since it is tunable. A Stochastic Hill Climbing procedure only moves to a neighbour if the evaluation score (number of operations) is improved. As a consequence, SHC could get stuck in local minima. Therefore, we consider to use Simulated Annealing instead of SHC, since SA has the ability to escape from local minima.

Simulated Annealing (SA) is a popular generalisation of SHC. It has four additional parameters with respect to SHC, namely (3) the initial temperature T_i, (4) the final temperature T_f, (5) the acceptance scheme, and (6) the cooling scheme [69]. The temperature governs the probability of accepting transitions with an energy higher than the energy of the current state. The cooling scheme governs how fast and in what way the temperature is decreased during the simulation (linearly, exponentially, etc.). Exponential cooling is frequently used. The acceptance scheme is most often the Boltzmann probability $\exp(\Delta E/T)$, that defines the probability of selecting a transition to an inferior state, given the difference in evaluation score ΔE. The number of iterations (1) of SA is not independent of the other parameters, as it can be computed from the initial temperature, final temperature, and cooling scheme. Most often, the search is started from a random position in the state space. In our application, we start from a random permutation of the variables. A basic SA algorithm is displayed in Algorithm 1.

```
$s \leftarrow$ random state, best $\leftarrow s$, $T \leftarrow T_i$;
while $T > T_f$ do
    $s' \leftarrow$ random neighbour(s);
    if $e^{(E_s - E_{s'})/T} > rand(0,1)$ then
        $s \leftarrow s'$;                                /* Accept new state */
        if $E_s < E_{best}$ then
            best $\leftarrow s$;
        end
    end
    $T \leftarrow \alpha T$;                                    /* Annealing */
end
return best;
```

Algorithm 1: A basic SA implementation, with a neighbourhood function (2), initial temperature T_i (3), final temperature T_f (4), a Boltzmann acceptance scheme (5), and exponential cooling with cooling parameter α (6). The number of iterations (1) is $\log_\alpha(T_f/T_i)$.

For SA we consider the following. If the initial temperature is high, then transitions to inferior states are permitted, allowing an escape from local minima. In order to determine the effect of the initial temperature T_i on the results, we will perform a sensitivity analysis. We use Boltzmann probability as the acceptance scheme, exponential cooling, a final temperature of 0.01, $N = 1000$ iterations, and a swap neighbourhood structure (for a visualisation, see figure 9).

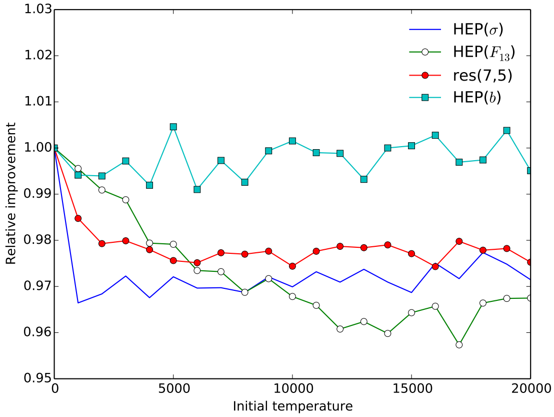


Figure 8: The relative improvement (lower is better) of the number of operations for a given initial temperature T_i, compared to $T_i = 0$. Each data point is the average of more than 100 SA runs with 1000 iterations, and a swap neighbourhood structure. We show the expressions HEP(σ), HEP(F_{13}), res(7,5), and HEP(b). The number of operations is only slightly influenced by the initial temperature, since the best improvement over $T_i = 0$ is smaller than 5%.

In figure 8 we show the relative improvement (lower is better) of the number of iterations for a given initial temperature T_i compared to $T_i = 0$ for the expressions HEP(σ), HEP(F_{13}), res(7,5), and HEP(b). Naturally, for $T_i = 0$, the relative improvement to itself is 1. For all expressions except HEP(b), we see a region where the improvement is largest: for HEP(σ) it is approximately $[1000, 7000]$, for HEP(F_{13}) it is $[12\,000, 17\,000]$ and for res(7,5) it is $[5000, 20\,000]$. This improvement is less than 5%. For higher T, too many transitions to inferior states are accepted to obtain good results. HEP(b) seems to be independent of the initial temperature. The fluctuations of 1% are statistical fluctuations.

The difference between the best results for all the expressions in figure 8 and the result at $T = 0$ is less than 5%. Since a $T_i = 0$ SA search is effectively an SHC search, this means that an almost parameterless Stochastic Hill Climbing (SHC) is able to obtain results that are only slightly inferior. This is surprising, since the way SHC traverses the state space is different from the way by SA. We here reiterate once more, SHC can get stuck in local minima, whereas SA has the possibility to escape. Furthermore, if a saddle point is reached, SA is able to climb over the hill, whereas SHC has to walk around the hill in order to escape. In subsection 2.4.3 we will show that local minima are uncommon, and that most of them are actually saddle points (i.e., local “minima” with a way to escape). Consequently, SA performs slightly better not because it can escape from local minima, but because, for some polynomials, walking over a saddle point (SA) is slightly faster to find better states than trying to circumvent the saddle point (SHC).

The reason why we prefer SHC over SA is that (1) the fundamental algorithmic improvement of SA – the ability to escape from local minima – is not used in practice, as we will see in subsection 2.4.3, and (2) tuning the SA parameters is expensive. Several methods have been suggested to tune the initial temperature, such as [76] and [77], but they often take several hundred iterations to obtain reliable values (which is quite expensive in our case). The small benefit of SA can be obtained by three other ways. First, by performing SHC runs in parallel (see section 2.4.2), second, by increasing the number of iterations, and third by selecting an initial temperature based on previous information such as figure 8. Using these three optimisations, a small improvement is obtained without increasing the run time.

2.4.2 *Neighbourhood structure*

The main parameter of SHC is the neighbourhood structure. Choosing an appropriate neighbourhood structure is crucial, since it determines the shape of the search space and thus influences the search performance. In [78] it is observed that the neighbourhood structure can have a significant impact on the quality of the solutions for the Travelling Salesman Problem, the Quadratic Assignment Problem, and the Flow-shop Scheduling Problem.

There are many neighbourhood structures for permutation problems such as Horner schemes. For example, a transition could swap two variables in the Horner scheme or move a variable in the scheme. However, there are also neighbourhood structures that involve changing larger structures. Figure 9 gives an overview of four basic transitions from which others can be constructed. From top to bottom, it shows (a) a single swap of two variables in the scheme, (b) a shift of a variable, (c) a shift of a sublist, and (d) a mirroring of a sublist. At each iteration of SHC, a transition to a randomly chosen neighbour is proposed. For the single swap transition, this involves the selection of two random variables in the scheme.

To examine which neighbourhood structure performs best for Horner schemes, we investigate seven (combinations of) neighbourhood structures, viz. (1) a single swap, (2) two consecutive swaps, (3) three consecutive swaps, (4) a shift of a single variable, (5) mirroring of a sublist, (6) a sublist shift (which we call 'many shift'), and (7) mirroring and/or shifting with an equal probability (which we call 'mirror shift'). Swapping multiple times in succession allows for faster traversal of the state space, but also runs the risk to miss states. Moreover, we have tested hybrid transitions. For instance, we performed two consecutive swaps in the first half of the simulation and resorted to single swaps for the latter half. However, we found that these combinations did not perform better. In order to present clear plots, we have omitted the plots resulting from these combinations.

Below we discuss (A) the measuring quality, (B) detailed results for res(7,6) and HEP(σ), (C) detailed results for HEP(F_{13}) and HEP(b), and (D) combined results.

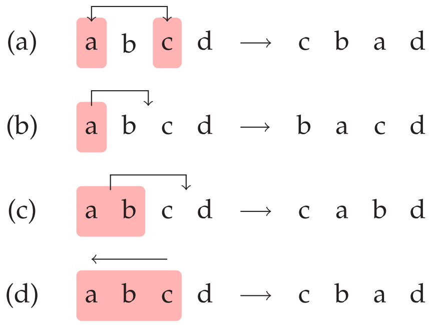


Figure 9: The elementary neighbourhood structures we use. From top to bottom: (a) a single swap, (b) a single shift, (c) shift of a sublist, and (d) mirroring of a sublist.

(A) Measuring quality

We now start investigating two methods of measuring the quality of a neighbourhood structure by: (A1) the average number of operations obtained by using a neighbourhood structure, and (A2) the lowest number of operations after performing *several* runs.

A1 Figure 10 shows the distribution of the number of operations of the expression HEP(F_{13}) after 10 000 SHC runs with the neighbourhood structure that exchanges two random variables. The average of this distribution is somewhere in the middle, but the actual values that one will measure will be either near 51 000 or near 62 000. Thus, the average is not an appropriate measure.

A2 So, we decided to measure the lowest score of several runs (A2), because in practice SHC is run in parallel, and so the results are more in line with those from practical applications. Thus, we are interested in the neighbourhood structure that has the lowest expected value of the minimum of k measurements. Here, we can use the expected value $\mathrm{E}\left[\min\left(X_0, \ldots, X_{k-1}\right)\right]$:

$$V_0 + \sum_{t=0}^{L-2} \left(V_{t+1} - V_t\right)\left(1 - \mathrm{cdf}(D,t)\right)^k \tag{11}$$

where k is the number of measurements, X_n is the score of the nth measurement, t is an index in the discrete distribution, V_t is the number of operations at t, D_t is the probability of outcome V_t, L is the number of possible outcomes, and cdf the cumulative distribution function. We shall denote the expected value of the minimum of k runs by $\mathrm{E}_{\mathrm{min},k}$.

Because the number of measurements k is in the exponent in eq. (11), $\mathrm{E}_{\mathrm{min},k}$ decreases exponentially with k and finally converges to V_0. As a consequence,

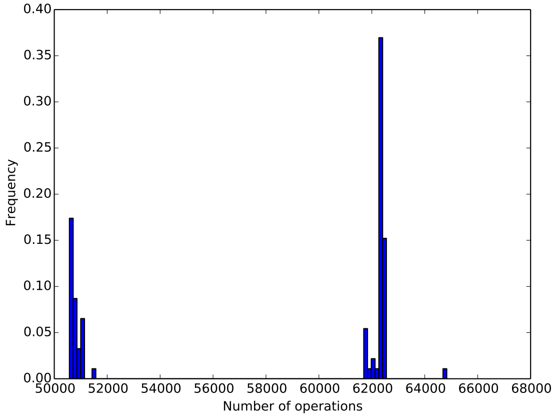


Figure 10: HEP(F_{13}) expression with 10 000 runs and 1 swap as a neighbourhood structure. Typical for our domain is that there are often two or more spikes. If the simulation is run multiple times, the probability of finding a value close to the minimum is high.

neighbourhood structures with a high standard deviation are more likely to achieve better results, since at high k the probability of finding a low value at least once is high. We found that four parallel runs ($k = 4$) yielded good results.

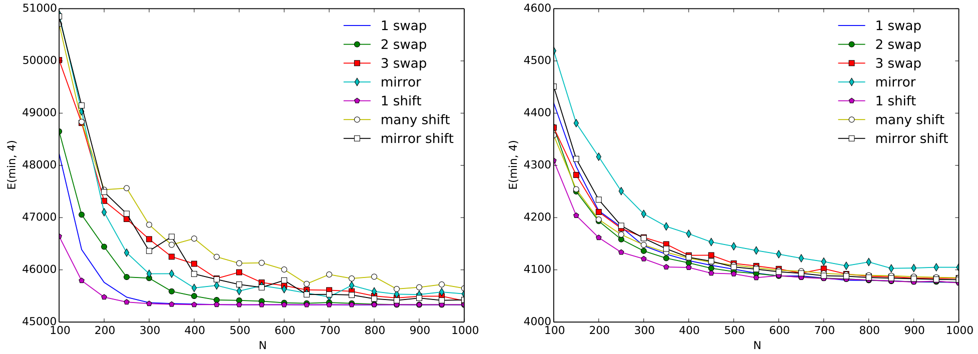
The results for res(7,4), res(7,5), and HEP(F_{24}) are similar, and are omitted for brevity (see [11]). Polynomial res(9,8) is too time consuming for such a detailed analysis (it would take around 35 days to collect all data).

(B) Results for res(7,6) and HEP(σ)

In figure 11a the performance of the neighbourhood structures for the expression res(7,6) is shown. We see that shifting a single variable ('1 shift') has the best performance at a low number of iterations N, followed by 2 consecutive swaps ('2 swap'). At around $N = 900$ all neighbourhood structures have converged. Thus, from $N = 900$ onward it does not matter which structure is chosen.

In figure 11b we show the performance of the neighbourhood structures for HEP(σ). We see that '1 shift' has the best performance at low N. At $N = 600$, all the neighbourhood structures have converged. The characteristics of this plot are similar to those of res(7,6).

We suspect that for a small state space, i.e., a small number of variables, there is not much difference between the neighbourhood structures, since the convergence occurs quite early (below $N = 1000$). Therefore, we look at two expressions with more variables: HEP(F_{13}) with 24 variables, and HEP(b) with 107 variables.



(a) res(7,6)

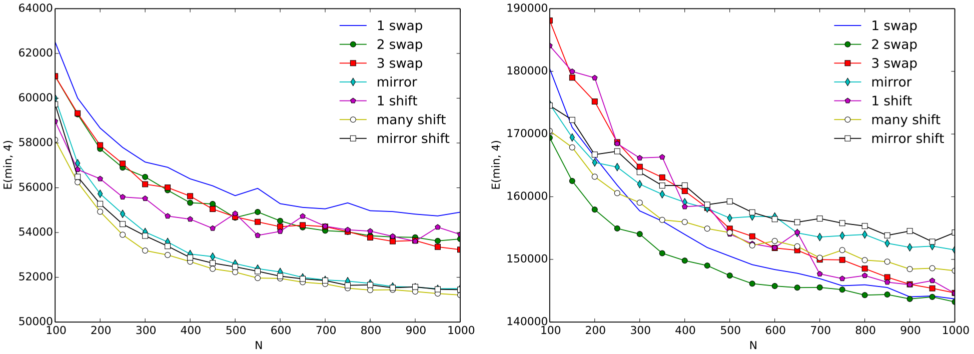
(b) HEP(σ)

Figure 11: The expected number of operations of the minimum of four SHC runs with N iterations for res(7,6) (left), and HEP(σ) (right). For res(7,6) the 1 shift performs best for low N, followed by the 2 swap. All the neighbourhood structures converge at $N = 900$. For HEP(σ), we see that 1 shift has the best performance at low N. At $N = 600$, all the neighbourhood structures have converged. The characteristics of the two plots are similar.

(C) Results for HEP(F_{13}) and HEP(b)

In figure 12a we show the results for HEP(F_{13}). We see that all the neighbourhood structures that involve small changes ('1 swap', '2 swap', '3 swap', and '1 shift') are outperformed by the neighbourhood structures that have larger structural changes ('mirror', 'many shift', and 'mirror shift'). The difference is approximately 8%. Both groups seem to have converged independently to different values. However, for larger N, we expect all neighbourhood structures to converge to the same value. The point of convergence has shifted to higher N compared to res(7,6), and HEP(σ), since the state space has increased in size from 15! to 24!. From this plot, we may conclude that the state space of HEP(F_{13}) is more suited to be traversed with larger changes.

In figure 12b the performance of the neighbourhood structures for HEP(b) is shown. We see that two consecutive swaps perform best at low N and that '1 swap', '2 swap', '3 swap', and '1 shift' converge at $N = 1000$. The neighbourhood structures that involve larger structural changes ('mirror', 'many shift', 'mirror shift') perform worse. These results are different from those of HEP(F_{13}): for HEP(b), with an even larger state space than HEP(F_{13}), smaller moves are better suited. This means that the mere number of variables is not a good indicator for the selection of a neighbourhood scheme.



(a) HEP(F_{13}) (b) HEP(b)

Figure 12: The expected number of operations of the minimum of four SHC runs with N iterations for HEP(F_{13}) (left), and HEP(b) (right). For HEP(F_{13}), mirror, many shift, and mirror shift converge to a lower value than the neighbourhood structures 1 swap, 2 swap, 3 swap, and 1 shift. For HEP(b), the 2 swap performs best for low N. 1 swap, 2 swap, 3 swap, and 1 shift converge at $N = 1000$. The other neighbourhood structures perform worse. Thus, the characteristics of HEP(F_{13}) and HEP(b) are different.

(D) Combined results

For the four benchmark expressions described in (B) and (C), and for the other three benchmark expressions, we observe that the relative improvement of the choice of the best neighbourhood structure compared to the worst neighbourhood structure is never more than 10%. Furthermore, we observe that there are two groups of neighbourhood structures when the state space is sufficiently large. Group 1 makes small changes to the state ('1 swap', '2 swap', '3 swap', '1 shift'). Group 2 makes large structural changes ('mirror', 'many shift' 'mirror shift'). The two groups converge before $N = 1000$ for expressions with small state spaces, such as HEP(σ), but are further apart for expressions with more variables, such as HEP(F_{13}) and HEP(b). The difference in quality in the group itself is often negligible (less than 3%). Thus, as a strategy to apply the appropriate neighbourhood structure, we suggest to distribute the number of parallel runs evenly among the two groups: in the case of four parallel runs, two of the runs can be performed using one neighbourhood structure from the group 1 and two from group 2.

2.4.3 *Two state space properties*

The fact that SHC works so well is surprising. Two well-known obstacles are (1) a Stochastic Hill Climbing can get stuck in local minima which yields inferior results [72], and (2) the Horner scheme problem does not have local heuristics, so

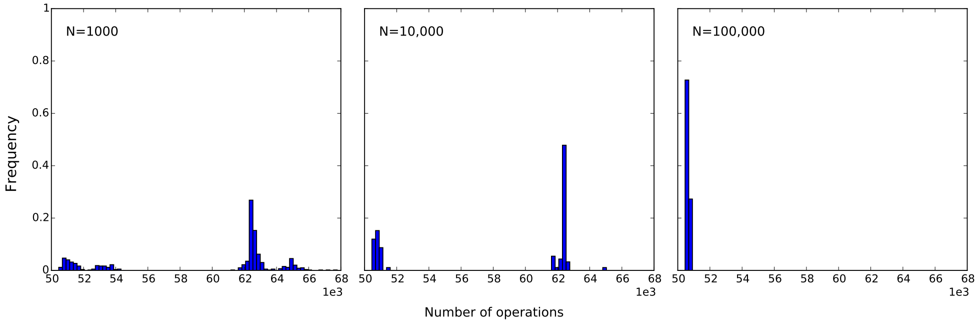


Figure 13: The distribution of the number of operations for the HEP(F_{13}) expression with 1 swaps for 1000 iterations (left), 10 000 iterations (middle) and 100 000 iterations (right) at $T = 0$. There appears to be a local minimum around 62 000. However, as the number of iterations is increased, the local minimum becomes smaller relative to the global minimum region (middle) and completely disappears (right). We may conclude that the apparent local minimum is not a local minimum, but a saddle point, since SHC is able to escape from the 'minimum'.

there is no guidance for any best-first search. Remarkably, SHC only needs 1000 iterations for a 107 variable expression (HEP(b)) to obtain good results, whereas a TSP benchmark problem with a comparable state space size, viz. kroA100 [79] with 100 variables, takes more than a million iterations to converge using a manually tuned SA search.

A thousand iterations is also a small number compared to the size of the state space. The average distance between two arbitrary states is 98 swaps. A thousand iteration SHC search accepts approximately 300 suggested swaps, so at least 33% of all the accepted moves should move towards the global minimum. This scenario would be unlikely if the state space is unsuited for SHC, so perhaps the state space has convenient properties for our purposes. We discuss the following two properties in the subsections below: (A) local minima, and (B) the region of the global minimum.

(A) Local minima and saddle points

To obtain an idea on the number of local minima, we measure how often the simulation gets stuck: if there are many local minima, we expect the simulation to get stuck often. In figure 13, we show the distribution of HEP(F_{13}) for 1000, 10 000, and 100 000 SHC runs respectively. For 1000 and 10 000 runs we see two peaks: one at the global minimum near 51 000 and one at an apparent local minimum near 62 000. As the number of iterations is increased, the weight shifts from the apparent local minimum to the global minimum: at 1000 iterations, there is a probability of 27.5% of arriving in the region of the global minimum, whereas this is 36.3% at

10 000 iterations. Apparently the local minimum is 'leaking': given sufficient time, the search is able to escape. The figure on the right with 100 000 iterations confirms the escaping possibility: the apparent local minimum has completely disappeared. Thus, the local minimum is in reality a saddle point, since for a true local minimum there is no path with a lower score leading away from the minimum. Since SHC requires many iterations to escape from the saddle point, only a few transitions reduce the number of operations.

We observe that apparent local minima disappear for our other benchmark expressions as well. SHC runs with 100 000 iterations approach the global minimum for all of our benchmark expressions. For example, for HEP(F_{13}) mentioned above, the result is 50636 ± 57 and for HEP(σ) the result is 4078 ± 9. The small standard deviations indicate that no runs get stuck in local minima (at least not in local minima significantly higher than the standard deviation).

From these results we may conclude that true local minima, from which a Stochastic Hill Climbing cannot escape, are rare for Horner schemes.

(B) Flatness of the state space

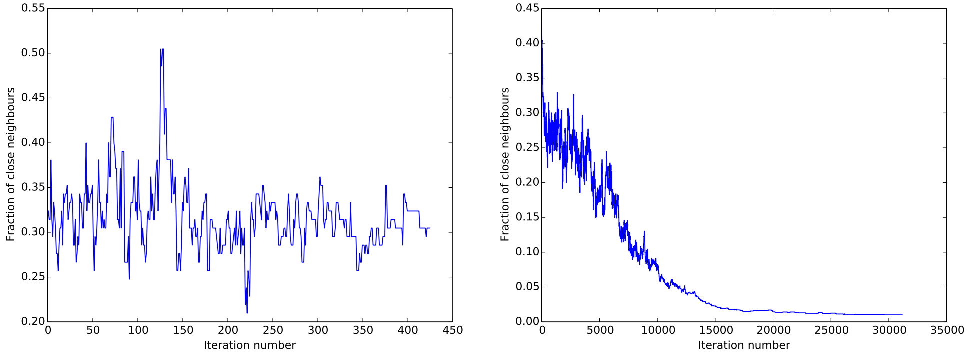
To build an intuition for what the state space looks like, we consider its flatness. We measure how many of the neighbours have a value (number of operations) that does not differ by more than 1%: $\frac{|x_n - x|}{|x|} < 1\%$, where x is the reference state and x_n is a neighbour of x. For brevity, we shall refer to this as 'close'.

In figure 14a we show the results for HEP(σ) for the current states during a typical single SHC run. We see that throughout the simulation the percentage of close neighbours is approximately 30%. We compare these results to an SA run of the TSP problem kroA100 (displayed in figure 14b). We see that for a random starting state the number of close neighbours is 30% as well, but as the simulation approaches the global minimum (at the right of the graph), the number of close neighbours decreases to 0.9%. As a result, the global minimum for TSP must be very narrow.

These results are a first hint that the state space of Horner is flat and terrace-like, whereas the TSP problem is more trough-like, with steep global/local minima. To investigate the flatness more deeply, we have looked at the distribution of the relative difference $\frac{|x_n - x|}{|x|}$. For the global minimum of the HEP(b) expression, this is depicted in figure 15. We see that about 75% of the neighbours are within 1% and 95% within 5%, which is even higher than for HEP(σ). We observed similar features for the other points in the state space, including hard to escape saddle points.

The property that the state space is flat is not only present in physics expressions, but is found in our other four benchmark expressions as well. Additionally, we have generated test expressions that we know to have interesting mathematical structures, such as powers of expressions. For example, for the expression $(4a + 9b + 12c^2 + 2d + 4e^3 - 2f + 8g^2 - 10h + i - j + 2k^2 - 3j^4 + l - 15m^2)^6$, 43% of the neighbours, 18% of the second neighbours and 7.3% of the third neighbours are close.

The question arises why the number of close neighbours is so high for the HEP(b) expression. For most expressions it is around 30%, but for HEP(b) it is 75%. A closer



(a) HEP(σ) with SHC

(b) kroA100 with SA

Figure 14: The typical number of close neighbours (value difference within 1%) for the current state for HEP(σ) and SHC on the left, and for SA on the the TSP benchmark problem kroA100 with 100 cities on the right. HEP(σ) does not show a decrease in the number of close neighbours, but remains steady around 30%. This is an indication that the state space is flat. For kroA100, the early states have many close neighbours, but as the simulation is converging to a minimum, the number of close neighbours decreases to about 0.9%. This is an indication that the state space has steep (local) minima.

inspection revealed that the HEP(b) expression has the special property that 90 of the 107 variables never appear in the same term: a term that contains variable x does not contain variable y and vice versa. As a result, the partial Horner schemes x, y and y, x yield the same expression. The HEP(b) expression is not the only expression with a high number of close neighbours: it represents a class of problems that often appears in electron-positron scattering processes.

The fact that some variables do not appear together in the same term is caused by a symmetry of the expression, since rearranging these variables in the scheme does nothing if they are direct neighbours in the scheme. The more symmetrical the expression is, the more likely it is that neighbours have the exact same value or a close value (within 5%). In the case of a uniformly random expression where the number of terms is much greater than the number of variables, we expect that practically all swaps are ineffective. The reason is that there is a high probability that each variable appears in an equal number of terms and has equal mixing.

Many, if not all, large expressions exhibit the 'flatness' property of their state space, since in most cases the number of terms is much larger than the number of variables. For the expressions that we have tested, the ratio of the number of terms and the number of variables is always more than a factor 1000. As a consequence, most

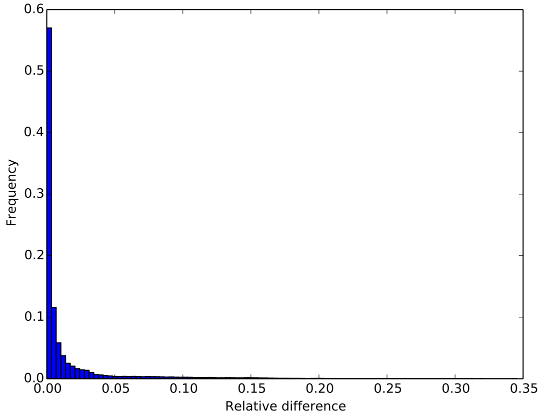


Figure 15: Relative difference of the values of swap neighbours of global minima for the HEP(b) expression, sampled over 37 states. The mean is 0.015 ± 0.034. The region is very flat: we observe that 85% of the neighbours have a value that is within 1% of the current value.

variables will appear in many terms, which in turn increases uniformity, resulting in neighbouring states with small differences in value (less than 5%).

2.5 PERFORMANCE OF SHC VS. MCTS

Below, we compare the results of Stochastic Hill Climbing to the previous best results from MCTS, for our eight benchmark expressions res(7,4), res(7,5), res(7,6), res(9,8), HEP(σ), HEP(F_{13}), HEP(F_{24}) and HEP(b). The results of all the MCTS runs except for res(9,8), and HEP(b) are taken from [53].[3] The results are displayed in table 2.

The results for MCTS with 1000 and 10 000 iterations are obtained after considerable tuning of C_p and after selecting whether the scheme should be constructed forward or in reverse (i.e., the scheme is applied backwards [14]).

For smaller problems, we observe that the averages of SHC are on a par with MCTS. However, we see that the standard deviations of SHC are higher than MCTS. This is because for MCTS the first nodes are fixed rather fast, which limits the variety. Consequently, we expect SHC to outperform MCTS if several runs are performed in parallel. Indeed, this is what we see in the last column of table 2. The standard deviations of MCTS are often an order of magnitude smaller than those of SHC, so the benefits of running MCTS in parallel are smaller. We may conclude that SHC has a better minimal behaviour if run in parallel.

[3] We only consider optimisations by Horner schemes and CSEE. Additional optimisations that are mentioned in [53], such as ‘greedy’ optimisations, can just as well be applied to the results of SHC.

	vars	original	MCTS 1k	MCTS 10k	SHC 1k	SHC 10k	$E_{min,4}$ 1k
res(7,4)	13	29 163	$(3.86 \pm 0.1) \cdot 10^3$	$(3.84 \pm 0.01) \cdot 10^3$	$(3.92 \pm 0.28) \cdot 10^3$	3834 ± 26	3819 ± 9
res(7,5)	14	142 711	$(1.39 \pm 0.01) \cdot 10^4$	13768 ± 28	13841 ± 441	13767 ± 21	13770 ± 5
res(7,6)	15	587 880	$(4.58 \pm 0.05) \cdot 10^4$	$(4.54 \pm 0.01) \cdot 10^4$	46642 ± 3852	$(4.61 \pm 0.25) \cdot 10^4$	$(4.55 \pm 0.16) \cdot 10^4$
res(9,8)	19	83 778 591	$(5.27 \pm 0.25) \cdot 10^6$	$(4.33 \pm 0.31) \cdot 10^6$	$(4.13 \pm 0.34) \cdot 10^6$	$(4.03 \pm 0.17) \cdot 10^6$	$(3.97 \pm 0.18) \cdot 10^6$
HEP(σ)	15	47 424	4114 ± 14	4087 ± 5	4226 ± 257	4082 ± 58	4075 ± 25
HEP(F_{13})	24	1 068 153	$(6.6 \pm 0.2) \cdot 10^4$	$(6.47 \pm 0.08) \cdot 10^4$	$(5.99 \pm 0.51) \cdot 10^4$	$(5.80 \pm 0.55) \cdot 10^4$	$(5.37 \pm 0.40) \cdot 10^4$
HEP(F_{24})	31	7 722 027	$(3.80 \pm 0.06) \cdot 10^5$	$(3.19 \pm 0.04) \cdot 10^5$	$(3.16 \pm 0.23) \cdot 10^5$	$(3.06 \pm 0.23) \cdot 10^5$	$(2.98 \pm 0.09) \cdot 10^5$
HEP(b)	107	1 817 520	$(1.81 \pm 0.04) \cdot 10^5$	$(1.65 \pm 0.08) \cdot 10^5$	$(1.50 \pm 0.08) \cdot 10^5$	$(1.40 \pm 0.06) \cdot 10^5$	$(1.44 \pm 0.04) \cdot 10^5$

Table 2: SHC compared to MCTS. The MCTS results for all expressions except res(9,8) and HEP(b) are from [53]. All the values are statistical averages over at least 100 runs. SHC results have a larger standard deviation, and thus the expected value of the minimum is often lower than these values (see last column).

For our largest expressions, HEP(F_{13}), HEP(F_{24}) and HEP(b), we observe that SHC with 1000 iterations yields better results than MCTS with 10 000 iterations. For HEP(F_{24}), the average of SHC with 1000 iterations is about 20% better than the average for MCTS with 1000 iterations. In fact, the results are slightly better than MCTS with 10 000 iterations. If we take the $E_{min,4}$ into account, the expected value for HEP(F_{24}) is an additional 7% less.

The fact that SHC outperforms MCTS when the number of variables is larger than 23, may be due to the fact that there are not sufficient iterations for the branches to reach the bottom, making the choice of the last variables essentially random (see section 2.4 and [14]). This may also be the reason why for MCTS it is important whether the scheme is constructed forward or in reverse: if most of the performance can be gained by carefully selecting the last variables, building the scheme in reverse will yield better performance.

SHC is 10 times faster (in clock time) than MCTS, since most of the time is spent in the evaluation function. It is able to make reductions up to a factor of 26 for our largest expression.

2.6 CHAPTER CONCLUSION

Monte Carlo Tree Search (MCTS) with UCT has proven to be a good candidate to simplify large expression [53]. A downside of this method is that the constant C_p that governs exploration versus exploitation has to be tuned. The quality of the final scheme largely depends on this constant. We have modified the UCT algorithm so that C_p decreases linearly with the iteration number [14]. As a result, the final iterations are spent on *optimising* the end of the tree, instead of *exploring*. We show that using this modified UCT, the sensitivity to C_p is decreased by at least a factor 10 [13, 14]. Thus, the tuning is simplified.

Tree search methods, even with SA-UCT, have the problem that the beginning of the tree is optimised more than the end. For Horner schemes this does not lead to optimal solutions. Therefore we considered other algorithms that optimise uniformly.

Since sampling is slow for our use case, tuning many parameters is infeasible. For this reason, we preferred straightforward algorithms over sophisticated ones. In the case of Horner schemes, we have found that one of the most basic algorithms, Stochastic Hill Climbing, yields the best results.

2.6.1 *Findings and conclusions*

Stochastic Hill Climbing provides a search method with two parameters: (1) the number of iterations (computation time) and (2) the neighbourhood structure, which is a tunable parameter. We found that running half of the simulations with a neighbourhood structure that makes minor changes to the state (i.e., a single shift of a variable), and running the other half with a neighbourhood structure that involves larger changes (i.e., the mirroring of a random sublist) is a good strategy for all of our benchmark expressions (see subsection 2.4.2). Consequently, only the computation time remains as an actual parameter. From our experimental results we arrive at three subconclusions: (1) SHC obtains similar results to MCTS for expressions with around 15 variables, (2) SHC outperforms MCTS for expressions with 24 or more variables, and (3) SHC requires ten times fewer samples than MCTS to obtain similar results. Therefore we may conclude that SHC is more than 10 times faster [11].

The result that a basic algorithm such as SHC performs well is surprising, since Horner schemes have at least two properties that make the search hard: (1) there are no known local heuristics, and (2) evaluations could take several seconds. In the previous sections we have shown that the performance of SHC is so good, because the state space of Horner schemes is flat and has few local minima.

The number of operations is linearly related to the time it takes to perform numerical evaluations. The difference between the number of operations for the unoptimised and the optimised expression is more than a factor 24 compared. As a consequence, we are able to perform numerical integration (via repeated numerical evaluations) at least 24 times faster.

For High Energy Physics, the contribution is immediate: numerical integration of processes that are currently experimentally verified at CERN can be done significantly faster.

2.6.2 *Future research*

We see two promising options for future research. First, our algorithms assume that the expressions are commutative, but our implementation could be expanded to be applied to generic expressions with non-commuting variables. Especially in physics, where tensors are common objects, this is useful. Horner's rule can only be applied uniquely to commutative variables, but the pulling outside brackets keeps the order of the non-commuting objects intact. Thus, for Horner's rule the only required change is the selection of commutative variables for the scheme. The common subexpression elimination should honour the ordering of the non-commutative objects. For example, in figure 2, the two highlighted parts are not a common subexpression if the variables are non-commutative ($a + e$ cannot be moved to the left of c). To enable non-commutative objects, CSEE should only compare connected subsets.

Second, additional work can be put in finding other methods of reductions. For example, expressing certain variables as linear combinations of other variables may reduce the number of operations even further. Many of these patterns cannot be recognised by common subexpression elimination alone. Determining which variables should be expressed as linear combinations of other variables to yield optimal results is an open problem. Perhaps Stochastic Local Search techniques are applicable to this subject as well.

Our algorithms are implemented in the next release of the open source symbolic manipulation system Form [80] and are used by multiple research groups.

3 FOUR LOOP FEYNMAN DIAGRAM CALCULATIONS

In this chapter we focus on answering

> **RQ2**: *How can we construct a program that can compute four-loop massless propagator integrals more efficiently?*

Over the years particle physics experiments have become more and more precise. This creates the need for more accurate calculations of the underlying processes. Quantum Field Theory (QFT) has proven to be a successful framework for making predictions about scattering experiments in particle accelerators such as the Large Hadron Collider (LHC). Especially Quantum Chromodynamics (QCD), the theory that describes the strong nuclear force mediated by gluons, is essential. One key feature of QCD is that interactions between particles can be described as a perturbative series, where every order improves the accuracy. Since the coupling constant is relatively large, higher order calculations are not negligible. For processes such as the production of Higgs bosons, three-loop calculations have recently been performed [81, 82]. This in turn necessitates the evaluation of the four-loop splitting functions to determine the parton distributions inside the proton. A complete calculation of the four-loop splitting functions is currently out of the question, in part due to the complexity of the integral reductions. The next best solution is to evaluate a number of Mellin moments as was done at the three-loop level over the past 25 years [83–87]. One way to obtain such moments is by converting them to integrals of a massless propagator nature by expansions in terms of the parton momentum. The computer program that could deal with the resulting three-loop integrals is called Mincer [32, 33] and its algorithms are based on Integration by Parts (IBP) identities [43]. To obtain higher moments Mincer has been optimised heavily. This eventually resulted in an $N = 29$ moment calculation for polarised scattering [88].

The construction of a similar program for four-loop propagator integrals is a far more formidable task. The attempts to solve the problem have led to the exploration of different techniques, such as the $1/D$ expansions of Baikov [89–91]. Instead of solving the systems of IBP equations parametrically as was done in Mincer, Laporta developed a method to solve the system by substituting values for its parameters [42]. This method has been used to create generic programs that can handle integrals in a flexible way [92–96]. The drawback of these programs is that it is in essence a brute-force Gaussian reduction, that needs to reduce many subsystems that will drop out of the final answer. An extra complication is the fact that the system is riddled with 'spurious poles' which are powers in $1/\epsilon = 2/(4 - D)$ that only cancel by the time all contributions to the coefficient of a master integral have been added. If it is not known in advance how many of these spurious poles will occur, one cannot safely perform a fixed expansions in ϵ. In the three-loop Mincer program the spurious poles could be avoided thanks to the resolved triangle formula by Tkachov [97], but for the all-N calculation these spurious poles caused significant issues [34, 35, 98]. In general, spurious pole avoidance is considered too complicated and is resorted to the very slow but exact arithmetic of rational polynomials.

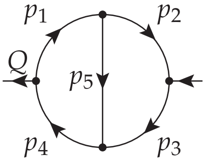
A method capable for a parametric reduction of massless four-loop propagator integrals has been developed by Roman Lee in a series of papers [99, 100]. It resulted in the LiteRed program, which is a Mathematica package that constructs reduction programs (also in Mathematica code). Although it is extremely elegant and as a method powerful, the resulting four-loop propagator programs are too slow for most practical applications (reductions take months or years).

In this chapter we describe Forcer, a Form [101, 102] program that is a hybrid between various approaches. We discuss the construction of a precomputed reduction graph that describes how to reduce each topology and how to map topologies with missing propagators into each other. Most topologies have straightforward reduction rules due to known reducible substructures, such as triangles or insertions. However, 21 topologies require a custom reduction rule (see section 3.6). Our research methodology consists of three steps. (1) We construct the diamond rule, which is able to efficiently reduce more topologies. Next, (2) we provide several heuristics for custom reductions. Finally, (3) we construct a program that applies the reduction rules to every topology.

The remainder of this chapter is structured as follows. In section 3.1 we define our Feynman diagrams. We discuss the reductions of simple substructures in sections 3.2 (the insertion rule) and 3.3 (the carpet rule). Next, we introduce Integration by Parts identities (IBPs) and construct the diamond rule in section 3.4 (step 1). In section 3.5 we show how to solve parametric IBPs by using heuristics (step 2). Section 3.6 lists all 21 topologies that are considered to be irreducible by lacking simple substructures. These involve the master integrals and a few more topologies that cannot be resolved in a single reduction step. In section 3.7 the superstructure of the program and its derivation are described (step 3). The usage of the program is discussed in section 3.8. In section 3.9 we show how to use expansions with Forcer. We describe how to transform physical diagrams to input for Forcer in section 3.10. In section 3.11 we show examples and we study the performance. Finally, section 3.12 provides the chapter conclusion.

3.1 generalised feynman diagrams

We start by describing the objects we would like to compute. Each scattering process requires the computation of Feynman integrals, which have an intuitive graph representation. The order of perturbations is captured by the number of fundamental cycles ('loops') in the graph. An example of a two-loop Feynman diagram with its corresponding integral is



$$= \int d^D p_1 d^D p_2 \frac{1}{p_1^2 p_2^2 (Q+p_2)^2 (Q+p_1)^2 (p_1-p_2)^2} \,, \tag{12}$$

where D is the dimension. Each propagator is rewritten in terms of the momentum basis $\{Q, p_1, p_2\}$. For physical calculations, the integrals should be computed in four dimensions, but sadly many integrals are divergent in $D = 4$. To regulate the infinities, which we should subtract at some point, we shift the space-time dimension to $D = 4 - 2\varepsilon$, where ε is a small positive number.

The diagram in eq. (12) has two loops and is scalar. In QCD, each propagator (line) and vertex are functions, defined by Feynman rules, that can be rewritten into a sum of scalar diagrams. For now we assume that the Feynman rules have been applied, so that we are now concerned with computing diagrams of the form:

$$F = \int d^D l_1 \cdots d^D l_m \frac{(p_{r_1} \cdot p_{r_2})^{n_{r_{12}}} \cdots (p_{r_3} \cdot p_{r_4})^{n_{r_{34}}}}{(p_1 \cdot p_1)^{n_1} \cdots (p_k \cdot p_k)^{n_k}} \,, \tag{13}$$

where D is the dimension, l_i are loop momenta, and p_i are momenta of propagators. Potentially, the powers of the propagators n_i contain powers of ε. These diagrams with arbitrary powers on the numerator are not physical, but can be obtained from them by rewriting momenta. Thus we consider a class of generalized Feynman diagrams.

All dot products can always be rewritten into a basis. For an L-loop diagram with K external momenta, the basis size is $(L+1)L/2 + LK$. If the number of edges is E, then there are $(L+1)L/2 + LK - E$ irreducible numerators for that topology.

In order to compute these diagrams, we shall rewrite them into a sum over simpler integrals. A simpler integral is an integral with fewer number of lines or loops, or with fewer numerators. Especially a reduction in the number of loops makes the integral easier, as the degrees of freedom are lowered.

An example of a simplification of the numerator structure is given below:

$$\begin{aligned}
\int d^D p \frac{p \cdot Q}{p^2 (p+Q)^2} &= \int d^D p \frac{1}{2} \frac{(p+Q)^2 - p^2 - Q^2}{p^2 (p+Q)^2} \\
&= \int d^D p \frac{1}{2} \frac{1}{p^2} + \int d^D p \frac{1}{2} \frac{1}{(p+Q)^2} + \int d^D p \frac{1}{2} \frac{-Q^2}{p^2 (p+Q)^2} \\
&= \frac{-Q^2}{2} \int d^D p \frac{1}{p^2 (p+Q)^2} \,,
\end{aligned} \tag{14}$$

where the first two terms are dropped since they are massless tadpoles. The last term has no numerator structure and is thus considered simpler.

In the following sections we describe simplifications of integrals that can always be performed if the integral has a certain substructure.

3.2 INTEGRATION OF ONE-LOOP TWO-POINT FUNCTIONS

If the Feynman graph has a massless one-loop two-point function, as shown in figure 16, the bubble can be integrated. This structure is also referred to as an *insertion*.

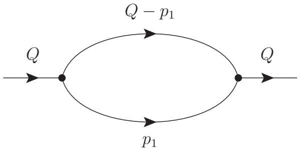


Figure 16: A massless one-loop two-point function, also referred to as an insertion.

An efficient formula that works for an insertion with an arbitrary numerator structure and arbitrary (including non-integer) powers for the propagators is the following formula [103]:

$$\int \frac{d^D P}{(2\pi)^D} \frac{\mathcal{P}_n(P)}{P^{2\alpha}(P-Q)^{2\beta}} =$$
$$\frac{1}{(4\pi)^2}(Q^2)^{D/2-\alpha-\beta} \sum_{\sigma \geq 0}^{[n/2]} G(\alpha,\beta,n,\sigma) Q^{2\sigma} \left\{ \frac{1}{\sigma!} \left(\frac{\Box}{4}\right)^{\sigma} \mathcal{P}_n(P) \right\}_{P=Q}, \tag{15}$$

where

$$\mathcal{P}_n(P) = P_{\mu_1} P_{\mu_2} \cdots P_{\mu_n}, \qquad \Box = \partial^2/\partial P_\mu \partial P_\mu\ , \tag{16}$$

and G can be expressed in terms of Γ-functions:

$$G(\alpha,\beta,n,\sigma) = (4\pi)^\epsilon \frac{\Gamma(\alpha+\beta-\sigma-D/2)\Gamma(D/2-\alpha+n-\sigma)\Gamma(D/2-\beta+\sigma)}{\Gamma(\alpha)\Gamma(\beta)\Gamma(D-\alpha-\beta+n)}. \tag{17}$$

The function G is normalised to a function G in which (1) the powers of the denominators are one plus potentially a multiple of ϵ and (2) there are no numerators. The difference is a number of Pochhammer symbols in ϵ which can either be expressed as rational polynomials or can be expanded in terms of ϵ, depending on what is needed. When finite expansions are used it is easy to generate tables of these Pochhammer symbols. The remaining function G is basically part of the master integral and kept for the end of the program when the master integrals are substituted.

In the presence of powers of the loop momentum P in the numerator, it is much faster to apply the above formula then to use IBP identities to lower the numerator. This holds in particular when one works with the FORM system, because its instruction set allows the evaluation of powers of the d'Alembertians with perfect efficiency.

It means that each term will be generated with the proper combinatoric factor and hence never gets generated more than once. The latter is owing to the combinatoric functions distrib_ and dd_. The code is displayed in listing 3.1.

Listing 3.1: Form code for one-loop insertion

```
    Tensor Ptensor,del;
    Vector P,Q,p1,p2,p3,p4;
    Symbols dAlembertian,j;
    Local F = dAlembertian^15*P.p1^15*P.p2^15*P.p3^15*P.p4^15;
    ToTensor,P,Ptensor;
    id dAlembertian^j?*Ptensor(?a) = distrib_(1,2*j,del,Ptensor,?a);
    ToVector,Ptensor,Q;
    id del(?a) = dd_(?a);
    Print +f +s;
    .end

  Time = 3.09 sec Generated terms = 1133616
               F Terms in output = 1133616
                         Bytes used = 140937744

     F =
         + 30924700750944000000*Q.p1*Q.p2*Q.p3^13*Q.p4^15*p1.p1*
        p1.p2^10*p1.p3^2*p2.p2^2
              .....
         + 145104494304820050000*Q.p1^7*Q.p2^10*Q.p3^13*p1.p1^3*
        p1.p4^2*p2.p2*p2.p3^2*p2.p4*p4.p4^6
          etc.
```

When computing Mellin moments of structure functions (see 4.3), the efficient combinatoric functions described above are essential, since they are used both in the one-loop integrals and the harmonic projections.

Two drawbacks of the approach is that (1) one may need to rewrite the dot products or invariants in the numerator to such a form that they are usable for the above formula. (2) The result of integrating out the insertion is that the remaining graph edge will have a non-integer power. Since the two-point function was embedded in a larger diagram, this edge may be internal. Consequently, this line can never be reduced to zero again by IBP identities.

3.3 carpet rule

For integrals where a subgraph is embedded in an outer one-loop graph, scaling and Lorenz invariance arguments [43] allow us to integrate out the outer one first. In figure 17, we show this structure.

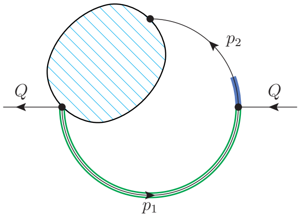


Figure 17: An subgraph (blob) embedded in a one-loop graph. The green triple line can be integrated out, regardless of the content of the shaded blob.

The generalized rule that works for these topologies with a generic numerator structure, we call the 'carpet' rule:

$$
\begin{aligned}
&\int \frac{d^D p}{(2\pi)^D} \frac{1}{(p^2)^\alpha \left[(p-q)^2\right]^\beta} \left[\prod_{i=1}^{L_\text{sub}} \int \frac{d^D l_i}{(2\pi)^D}\right] \left[\prod_{i=1}^{N_\text{sub}} \frac{1}{(p_i^2)^{a_i}}\right] \mathcal{P}_n(\{p_i\}, q) \\
&\quad = \frac{1}{(4\pi)^2} (q^2)^{D/2-\alpha-\beta} \sum_{\sigma=0}^{\lfloor n/2 \rfloor} \left(\frac{D}{2} + n - \sigma\right)_{-\sigma} \\
&\qquad \times G\left(\alpha + \sum_{i=1}^{N_\text{sub}} a_i - \frac{D}{2} L_\text{sub} - \sigma, \beta, n - 2\sigma, 0\right) \qquad (18) \\
&\qquad \times \sum_{j=0}^{\lfloor n/2-\sigma \rfloor} (-1)^j \left(\frac{D}{2} + n - 2\sigma - 1\right)_{-j} (q^2)^{\sigma+j} \\
&\qquad \times \left[\prod_{i=1}^{L_\text{sub}} \int \frac{d^D l_i}{(2\pi)^D}\right] \left[\prod_{i=1}^{N_\text{sub}} \frac{1}{(p_i^2)^{a_i}}\right] \left[\frac{1}{\sigma! j!} \left(\frac{\Box_q}{4}\right)^{\sigma+j} \mathcal{P}_n(\{p_i\}, q)\right]_{p=q} .
\end{aligned}
$$

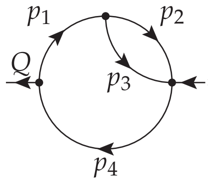
Here L_sub is the number of loops in the embedded subgraph. The integrand of the subgraph consists of two parts: a product of $1/(p_i^2)^{a_i}$ and $\mathcal{P}_n(\{p_i\}, q)$. Each p_i^2 indicates not only a squared propagator in the subgraph but also any quadratic Lorentz scalar that becomes p^2 after the integrations in the subgraph, e.g., $p_i \cdot p_j$ and $p_i \cdot p$. If $\mathcal{P}_n(\{p_i\}, q) = 1$ (and $n = 0$), the formula just describes that the knowledge of the dimension of the subgraph is sufficient to write down the result of the outer loop integral. In general, $\mathcal{P}_n(\{p_i\}, q)$ is a polynomial with degree n both in p_i and q, which are taken as dot products of $p_i \cdot q$ in Forcer. In the right-hand side of the formula, $(x)_n = \Gamma(x+n)/\Gamma(x)$ is the Pochhammer symbol and the function G is given by eq. (17). The d'Alembertian $\Box_q = \partial^2/\partial q_\mu \partial q^\mu$ can be efficiently implemented by `distrib_` and `dd_` functions in Form as explained in sec. 3.2.

A whole class of simplification identities can be derived from the following rule stemming from partial integration of any Feynman integral F [43]:

$$\frac{\partial}{\partial p_i} p_j \circ F = 0\,, \tag{19}$$

where the operator $\circ$ means that the preceding factors should be applied to the integrand of F (so before integration). Here we remark that p_i and p_j can be any momentum. However, the equation will yield different relations only if p_i and p_j are part of the same loop momentum basis, or p_j is the external momentum. Thus, in total there are $L(L+K)$ independent partial integration equations.

Let us consider the following (generalised) diagram:



$$= \int d^D p_1 d^D p_2 \frac{1}{(p_1^2)^{n_1}(p_2^2)^{n_2}(p_1-p_2)^{2n_3}(Q+p_1)^{2n_4}}\,, \tag{20}$$

to which we apply the IBP identity $Q^\mu \frac{\partial}{\partial p_2^\mu}$:

$$\begin{aligned}
Q^\mu \frac{\partial}{\partial p_2^\mu} \circ F &= \int d^D p_1 d^D p_2 Q^\mu \frac{\partial}{\partial p_2^\mu}\left[\frac{1}{(p_1^2)^{n_1}(p_2^2)^{n_2}(p_1-p_2)^{2n_3}(Q+p_1)^{2n_4}}\right] \\
&= \int d^D p_1 d^D p_2 \left[-2n_2 \frac{Q\cdot p_2}{(p_1^2)^{n_1}(p_2^2)^{n_2+1}(p_3^2)^{n_3}(p_4^2)^{n_4}} - 2n_3 \frac{Q\cdot p_1 - Q\cdot p_2}{(p_1^2)^{n_1}(p_2^2)^{n_2}(p_3^2)^{n_3+1}(p_4^2)^{n_4}}\right] \\
&= \left[-2n_2 \frac{Q\cdot p_2}{p_2^2} + 2n_3 \frac{Q\cdot p_2}{p_3^2} - n_3 \frac{p_4^2}{p_3^2} + n_3 \frac{p_1^2}{p_3^2} + n_3 \frac{Q^2}{p_3^2}\right] \circ F = 0\,,
\end{aligned} \tag{21}$$

where we have suppressed the integration and written the diagram in the basis $\{p_1^2, p_2^2, p_3^2, p_4^2, 2Q\cdot p_2\}$. For each index, described by n_i, we introduce a raising and lowering operator $\boldsymbol{N}_i^+$ and $\boldsymbol{N}_i^-$ (capitalising the index letter and making it bold), which raise or lower the power of the index. By definition, we consider the powers of the numerator to be negative. Thus, the raising operator on a numerator will decrease its power. We get:

$$\left[-n_2 \boldsymbol{N}_5^+ \boldsymbol{N}_2^+ + n_3\left(\boldsymbol{N}_5^+ \boldsymbol{N}_3^+ - \boldsymbol{N}_4^- \boldsymbol{N}_3^+ + \boldsymbol{N}_1^- \boldsymbol{N}_3^+ + Q^2 \boldsymbol{N}_3^+\right)\right] F = 0\,. \tag{22}$$

Alternatively, we can write:

$$\begin{aligned}
& n_2 F(n_1, n_2+1, n_3, n_4, n_5+1) + n_3 F(n_1, n_2, n_3+1, n_4, n_5+1) \\
& - n_3 F(n_1, n_2, n_3+1, n_4-1, n_5) - n_3 F(n_1-1, n_2, n_3+1, n_4, n_5)
\end{aligned}$$

$$+ n_3 Q^2 F(n_1, n_2, n_3 + 1, n_4, n_5) = 0\,. \tag{23}$$

Systems of IBP identities can be solved such that repeated application of a rule always decreases one of the lines to zero. Below we show two IBP identities that can always be applied if the integral has a certain substructure. In section 3.4.1 we discuss the rule of the triangle, and in section 3.4.2 we construct the diamond rule. Finally, we sketch the construction of custom rules in section 3.4.3.

3.4.1 *The rule of the triangle*

Since the 1980s, the so-called *triangle rule* [43, 103] has been used for removing a propagator line from diagrams corresponding to a certain class of integrals. Any topology that has the following substructure can be simplified using the triangle rule:

$$F(a_1, a_2, b, c_1, c_2) = \int d^D k \frac{k^{\mu_1} \dots k^{\mu_N}}{\left[(k+p_1)^2 + m_1^2\right]^{a_1} \left[(k+p_2)^2 + m_2^2\right]^{a_2} (k^2)^b (p_1^2 + m_1^2)^{c_1} (p_2^2 + m_2^2)^{c_2}}, \tag{24}$$

where b, c_1, c_2 are positive integers. The diagram corresponding to this integral is shown in fig. 18.

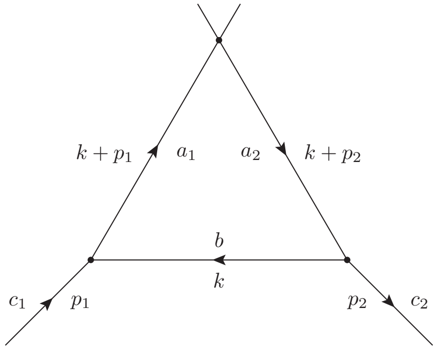


Figure 18: A triangle subtopology where the loop momentum k is assigned to the central line. p_1 and p_2 are external momenta. a_1, a_2, b, c_1, and c_2 represent the powers of their associated propagators.

We write out the IBP relation $\frac{\partial}{\partial k_\mu} k_\mu \circ F = 0$ to obtain

$$1 = \frac{1}{D + N - a_1 - a_2 - 2b} \Big[a_1 \boldsymbol{A}_1^+ (\boldsymbol{B}^- - \boldsymbol{C}_1^-) + a_2 \boldsymbol{A}_2^+ (\boldsymbol{B}^- - \boldsymbol{C}_2^-) \Big], \tag{25}$$

where again $\boldsymbol{A}_i^+$, $\boldsymbol{B}^-$, and $\boldsymbol{C}_i^-$ are operators acting on an integral that increase the power a_i by one, decrease the power b by one, and decrease the power c_i by one, respectively. Numerators that are expressed in dot products of k and an external line,

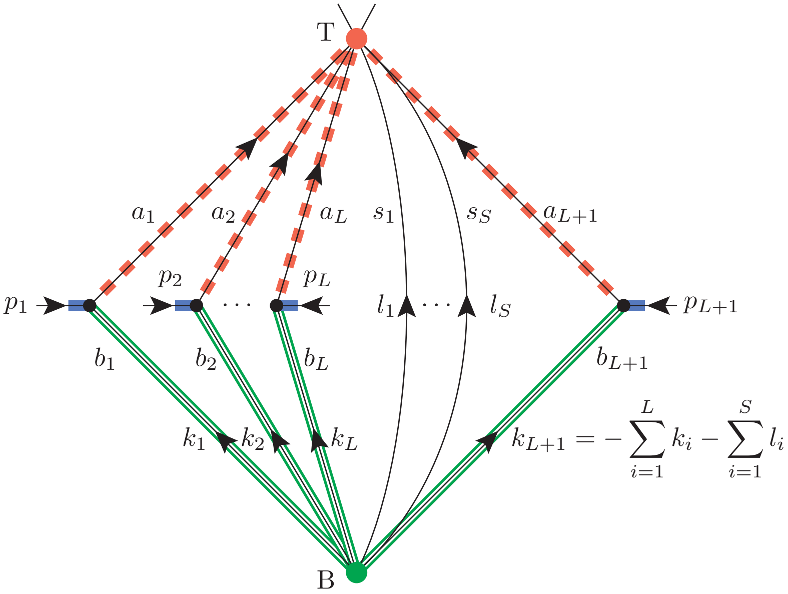


Figure 19: $(L+S)$-loop diamond-shaped diagram. $(L+1)$-lines have external connections and S-lines do not. Red with dashed lines, green with double lines, and blue with thick lines represent upper, lower, and external lines of the diamond, respectively. Label T represents the top vertex, and B the bottom vertex. k_i, p_i, and l_i are momenta, and a_i, b_i, and s_i are the powers of their associated propagators.

contribute as a constant N to the rule. The rule of the triangle can be recursively applied to remove one of the propagators associated with k, p_1, or p_2 from the system.

The recursion in the triangle rule can be explicitly 'solved' [97], such that the solution is expressed as a linear combination of integrals for which either b, c_1, or c_2 is 0. The advantage of the summed system over the recursion is that it generates fewer intermediate terms and it cannot have *spurious poles*: terms in which the factor $D + N - a_1 - a_2 - 2b$ becomes proportional to ϵ more than once during the full recursion.

3.4.2 *The rule of the diamond*

Next, we derive a rule for a new substructure we call a *diamond*. Consider the following family of Feynman integrals in D-dimensions arising from the $(L+S)$-loop diagram in Fig. 19:

$$F(\{a_i\},\{b_i\}) = \left[\prod_{i=1}^{L} \int d^D k_i\right] \left[\prod_{i=1}^{S} \int d^D l_i\right]$$

$$\times \left[\prod_{i=1}^{L+1} \frac{k_i^{\mu_1^{(i)}} \dots k_i^{\mu_{N_i}^{(i)}}}{\left[(k_i+p_i)^2+m_i^2\right]^{a_i}(k_i^2)^{b_i}}\right] \left[\prod_{i=1}^{S} \frac{l_i^{\nu_1^{(i)}} \dots l_i^{\nu_{R_i}^{(i)}}}{(l_i^2)^{s_i}}\right]. \tag{26}$$

The diagram consists of $(L+1)$ paths from the top vertex T to the bottom vertex B with an external connection in between, and S lines without external connections. The upper, lower, and external lines of the diamond are represented by red with dashed lines, green with double lines, and blue with thick lines, respectively. The lines without external connections, we call *spectator lines*. In principle any pair of spectator lines can be seen as a two point function which can be reduced to a single line by integration. This line would then have a power that is not an integer. Depending on the complete framework of the reductions this may or may not be desirable. Hence we leave the number of spectators arbitrary. In any case, the contribution of the spectators is a constant (see below), which allows us to characterise integrals in the family only by $2(L+1)$ indices a_i and b_i, and not by s_i. Without loss of generality, we assign loop momenta k_i to the lower lines of the diamond as well as l_i to the spectator lines, except the last diamond line which is fixed by momentum conservation:

$$k_{L+1} = -\sum_{i=1}^{L} k_i - \sum_{i=1}^{S} l_i. \tag{27}$$

In contrast, we do not require any constraints on the momentum conservation at the top vertex in the arguments below, hence any number of external lines can be attached to this point. In the middle of the diamond, external lines with momentum p_i are attached by three-point vertices. The upper lines in the diamond may have masses m_i, whereas the lower lines in the diamond and the spectator lines have to be massless. In addition, we allow arbitrary tensor structures of k_i and l_i with homogeneous degrees N_i and R_i, respectively, in the numerator.

Constructing the IBP identity corresponding to the operator

$$\sum_{i=1}^{L} \frac{\partial}{\partial k_i} \cdot k_i + \sum_{i=1}^{S} \frac{\partial}{\partial l_i} \cdot l_i, \tag{28}$$

straightforwardly gives the following operator identity:

$$(L+S)D + \sum_{i=1}^{L+1}(N_i - a_i - 2b_i) + \sum_{i=1}^{S}(R_i - 2s_i) = \sum_{i=1}^{L+1} a_i \boldsymbol{A}_i^{+}\left[\boldsymbol{B}_i^{-} - (p_i^2 + m_i^2)\right]. \tag{29}$$

Here $\boldsymbol{A}_i^{+}$ and $\boldsymbol{B}_i^{-}$ are understood as operators increasing a_i and decreasing b_i by one, respectively, when acting on $F(\{a_i\},\{b_i\})$. Note that operators changing the spectator indices s_i are absent in the identity.

For a typical usage of eq. (29), one may identify a diamond structure as a subgraph in a larger graph. If the line with the momentum p_i has the same mass m_i as the corresponding upper line, the term $(p_i^2 + m_i^2)$ in the identity reads as an operator $\boldsymbol{C}_i^{-}$

decreasing the corresponding index c_i of the power of the propagator $(p_i^2 + m_i^2)^{-c_i}$ in the larger graph by one. Applying the rule

$$1 = \frac{1}{E} \sum_{i=1}^{L+1} a_i \boldsymbol{A}_i^+ (\boldsymbol{B}_i^- - \boldsymbol{C}_i^-), \tag{30}$$

where

$$E = (L+S)D + \sum_{i=1}^{L+1} (N_i - a_i - 2b_i) + \sum_{i=1}^{S} (R_i - 2s_i), \tag{31}$$

decreases $\sum_{i=1}^{L+1}(b_i + c_i)$ of integrals appearing in the right-hand side, at the cost of increasing $\sum_{i=1}^{L+1} a_i$. Starting from positive integer indices b_i and c_i, one can repeatedly use the rule until one of either b_i or c_i is reduced to zero.[1]

The above diamond rule contains the conventional triangle rule as a special case. For the one-loop case $L = 1$ and $S = 0$, the two lower lines may be identified as a single line and the triangle integral in eq. (24) can be reproduced. Correspondingly, the IBP identity (30) becomes eq. (25).

Below we will (A) derive an explicit summation formula for the recursion in the diamond rule. Then (B) we will provide four examples of the diamond structure.

(A) Summation rule

We start by considering the possible connectivities in the diamond structure. If we allow for some external lines to be directly connected to each other, we get at least one triangle that can be used for the triangle rule: suppose the external momenta of k_i and of k_j are connected and identified with p_{ij}, then this triangle is k_i, k_j, p_{ij}. In this case, the triangle rule generates fewer terms and is preferred to the diamond rule. Thus, we only consider the case where the diamond does not have direct connections of external lines.

We follow the same procedure as outlined in [97]. First, we rewrite eq. (30) as:

$$F = \left[\sum_{i=1}^{L+1} a_i \boldsymbol{A}_i^+ (\boldsymbol{B}_i^- - \boldsymbol{C}_i^-) \right] \boldsymbol{E}^{-1} F, \tag{32}$$

where $\boldsymbol{E}$ is the operator $(L+S)D + \sum_{i=1}^{L+1}(N_i - a_i - 2b_i) + \sum_{i=1}^{S}(R_i - 2s_i)$. We split our solution in two classes $\boldsymbol{A}_i^+ \boldsymbol{B}_i^-$ and $\boldsymbol{A}_i^+ \boldsymbol{C}_i^-$ satisfying

$$\boldsymbol{E}^{-1}(\boldsymbol{A}_i^+ \boldsymbol{B}_i^-) = (\boldsymbol{A}_i^+ \boldsymbol{B}_i^-)(\boldsymbol{E}+1)^{-1}, \qquad \boldsymbol{E}^{-1}(\boldsymbol{A}_i^+ \boldsymbol{C}_i^-) = (\boldsymbol{A}_i^+ \boldsymbol{C}_i^-)(\boldsymbol{E}-1)^{-1}. \tag{33}$$

We identify the first class with the label $+$, since it increases E by 1, and the latter with the label $-$, since it decreases E by 1. The remaining part of the derivation is analogous to the one in [97].

[1] Note that a_i are allowed to be non-integers provided the denominator in the right-hand side of eq. (30) never vanishes.

Finally, we obtain the explicit summation formula:

$$
\begin{aligned}
&F(\{a_i\},\{b_i\},\{c_i\}) = \\
&\quad \sum_{r=1}^{L+1}\Bigg[\Bigg(\prod_{\substack{i=1\\ i\neq r}}^{L+1}\sum_{k_i^+=0}^{b_i-1}\Bigg)\Bigg(\prod_{i=1}^{L+1}\sum_{k_i^-=0}^{c_i-1}\Bigg)(-1)^{k^-}\frac{k_r^+(k^+ + k^- - 1)!}{\prod_{i=1}^{L+1}k_i^+!k_i^-!}(E+k^+)_{-k^+-k^-} \\
&\quad \times\Bigg(\prod_{i=1}^{L+1}(a_i)_{k_i^+ + k_i^-}\Bigg)F\left(\{a_i+k_i^+ + k_i^-\},\{b_i-k_i^+\},\{c_i-k_i^-\}\right)\Bigg]_{k_r^+=b_r} \\
&\quad + \sum_{r=1}^{L+1}\Bigg[\Bigg(\prod_{i=1}^{L+1}\sum_{k_i^+=0}^{b_i-1}\Bigg)\Bigg(\prod_{\substack{i=1\\ i\neq r}}^{L+1}\sum_{k_i^-=0}^{c_i-1}\Bigg)(-1)^{k^-}\frac{k_r^-(k^+ + k^- - 1)!}{\prod_{i=1}^{L+1}k_i^+!k_i^-!}(E+k^+ +1)_{-k^+-k^-} \\
&\quad \times\Bigg(\prod_{i=1}^{L+1}(a_i)_{k_i^+ + k_i^-}\Bigg)F\left(\{a_i+k_i^+ + k_i^-\},\{b_i-k_i^+\},\{c_i-k_i^-\}\right)\Bigg]_{k_r^-=c_r},
\end{aligned}
\tag{34}
$$

where $k^+ = \sum_{i=1}^{L} k_i^+$, $k^- = \sum_{i=1}^{L} k_i^-$, and $(a)_b$ is the rising Pochhammer symbol $\Gamma(a+b)/\Gamma(a)$. The first term decreases the power b_r to 0, and the second term decreases c_r to 0. The only significant difference between the two terms is the $+1$ in the Pochhammer symbol.

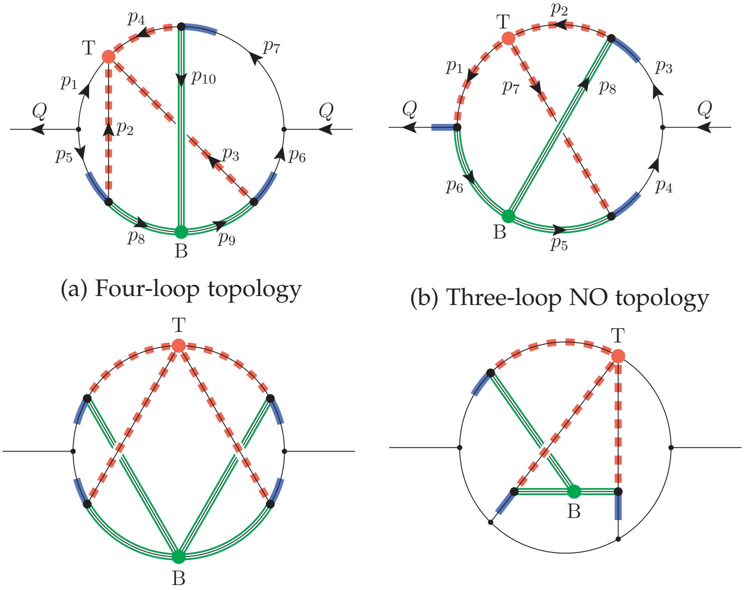
Because the Pochhammer symbol that depends on E only appears once in each term, powers of $1/\epsilon^2$ or higher cannot occur. Thus, the explicit summation formula for the diamond rule does not have spurious poles.

(B) Examples

Four examples of diamond structures are displayed in Fig. 20. The role of each line in the diamond rule is highlighted by different colours and shapes. Red dashed lines, green double lines, and blue thick lines represent upper, lower, and external lines of the diamond, respectively. Label T represents the top vertex, and B the bottom vertex. In Fig. 20a a four-loop diagram is displayed. For this diagram, the line of either p_5, p_6, p_7, p_8, p_9, or p_{10} can be removed by recursive use of the diamond rule or by the explicit formula given in the previous section. The irreducible numerators of this diagram are selected as $Q{\cdot}p_8$, $Q{\cdot}p_{10}$, $p_5{\cdot}p_{10}$, and $p_5{\cdot}p_7$, such that they adhere to the tensorial structure in the diamond rule. The last numerator, $p_5{\cdot}p_7$, lies outside of the diamond and does not interfere with the rule.

If, in this figure, we draw an additional line from the top (T) to the bottom (B) vertex, we obtain the simplest non-trivial propagator topology with a spectator line. As a five-loop diagram it is unique.

In Fig. 20b the three-loop master topology NO is displayed. $Q{\cdot}p_5$ is chosen as irreducible numerator. One of the lines attached to the diamond is actually an off-shell external line. In general, if the line with momentum p_{L+1} is one of the external momenta of the larger graph, the factor $(p_{L+1}^2 + m_{L+1}^2)$ is just a constant with respect to the loop integration and has no role for reducing the complexity of



(a) Four-loop topology

(b) Three-loop NO topology

(c) Five-loop topologies

Figure 20: Four topologies with highlighted diamond structures. Red with dashed lines, green with double lines, and blue with thick lines represent upper, lower, and external lines of the diamond, respectively. Label T represents the top vertex, and B the bottom vertex. (a) shows a four-loop topology which can be completely reduced. (b) shows the three-loop NO master topology, for which a modified form of the diamond rule can be applied to lower the power of line p_1 to 1. (c) shows two five-loop topologies, which the diamond rule can be applied to.

the integral. As a result, the rule (30) is not applicable to remove one of the internal lines. Even for such cases, one can still find a useful rule by shifting $a_{L+1} \to a_{L+1} - 1$:

$$1 = \frac{1}{p_{L+1}^2 + m_{L+1}^2} \left[\sum_{i=1}^{L} \frac{a_i}{a_{L+1} - 1} \boldsymbol{A}_i^+ \boldsymbol{A}_{L+1}^- (\boldsymbol{B}_i^- - \boldsymbol{C}_i^-) - \frac{E+1}{a_{L+1} - 1} \boldsymbol{A}_{L+1}^- + \boldsymbol{B}_{L+1}^- \right], \quad (35)$$

which decreases at least a_{L+1} or b_{L+1} by one. Repeated use of this rule from positive integer a_{L+1} and b_{L+1} reduces a_{L+1} or b_{L+1} to 1. For the NO topology, this variant yields the rule to reduce the line p_1 to 1 in Mincer [32, 33].[2]

In Fig. 20c we show two five-loop topologies for which the diamond rule can eliminate one line. The first diagram is unique in the sense that it is the simplest diagram for which $L = 3$, $S = 0$. The second diagram is a typical representative

[2] The triangle rule counterpart of this variant was used to reduce the peripheral lines of the massless two-loop propagator-type diagrams with non-integer powers of the central line to unity.

of the 29 five-loop topologies with $L = 2$, $S = 0$ and all three p-momenta of the diamond internal.

3.4.3 *Custom solutions*

Using the combination of above rules, many topologies can be simplified. However, if the Feynman diagram does not contain a triangle, a diamond, an insertion, or a carpet, the IBPs have to be solved manually. One well-known example is the reduction of the three-loop non-planar diagram [43], which paved the way for fast three-loop reductions.

Let us consider a straightforward example with a system of two equations:

$$\begin{aligned} &1)\ Z(n_1, n_2, n_3) = n_1 Z(n_1 - 1, n_2, n_3) + \mathbf{Z(n_1, n_2 + 1, n_3)} \\ &2)\ Z(n_1, n_2, n_3) = n_3 Z(n_1, n_2, n_3 - 1) + n_2 \mathbf{Z(n_1, n_2 + 1, n_3)} \end{aligned} \tag{36}$$

Here we see that we have two terms that only increase indices without lowering any others, highlighted in bold. These terms have to be removed in order to have a reduction scheme that, when repeatedly applied, always lowers one of the indices. Solving it analytically, we get:

$$Z(n_1, n_2, n_3) = \frac{1}{1 - n_2} \Big(n_3 Z(n_1, n_2, n_3 - 1) - n_1 n_2 Z(n_1 - 1, n_2, n_3) \Big) \tag{37}$$

This rule can be applied until $n_2 = 1$ or until $n_1 = 0$ or $n_3 = 0$, since the latter cases mean we have removed a propagator. The next step would be to solve $Z(n_1, \mathbf{1}, n_3)$, since any configuration with $n_2 > 1$ will have $n_1 = 0$ or $n_3 = 0$.

In general, it is quite difficult to solve these systems parametrically. Therefore, many computations are performed using the Laporta method [42, 94, 95, 100]. In its basic form, each parameter is substituted by a number, and every configuration is solved separately. The solutions to the system are brute force Gaussian eliminations, which is a time consuming process.

The Laporta system above for $Z(1, 2, 1)$ looks like:

$$\begin{aligned} &1)\ Z(1, 2, 1) = Z(0, 2, 1) + \mathbf{Z(1, 3, 1)} \\ &2)\ Z(1, 2, 1) = Z(1, 2, 0) + 2\mathbf{Z(1, 3, 1)} \\ &\Rightarrow Z(1, 2, 1) = 2Z(0, 2, 1) - Z(1, 2, 0)\ . \end{aligned} \tag{38}$$

If we had to solve $Z(2, 2, 1)$, we would have to solve the system of eq. (36) for $Z(2, 2, 1)$ and for $Z(1, 2, 1)$.

Some topologies are called *master topologies*: these cannot be guaranteed to get a line removed through IBPs, and they have to be solved using different methods. An example is the simple system above, that could be used to reduce n_2 to 1, but not to 0. In all cases we have encountered, the IBP reduction rules can be used to simplify the edge powers to 1, and the numerator powers to 0.

In this section we will give heuristics on how to solve more complicated IBP systems parametrically. In an N-loop propagator graph we have $N+1$ independent vectors: the external vector Q and N loop momenta p_i, where $i = 1, \ldots, N$. Together there are $(N+2)(N+1)/2$ independent variables. One of them, Q^2, can be used to set the scale. Hence there are $(N+2)(N+1)/2 - 1$ variables in the loops. Because there are at most $3N-1$ propagators, the remaining variables will be in the numerator and there is often quite some freedom as to which variables to choose. In topologies in which there are fewer propagators there will correspondingly be more variables in the numerator. The efficiency of the reduction depends critically on the selected numerators. In the Mincer program the numerators were chosen to be dot products, such as $2p_7 \cdot p_8$ for the ladder topology or $2Q \cdot p_2$ for the Benz topology. Alternatively, one could use extra squared momenta such p_9^2 with $p_9 = p_7 - p_8$. The advantage of the invariant method is that when rewriting the numerators to a new basis after a line removal, more invariants of the old basis can be a part of the new basis. The advantage of using dot products is that integration of one-loop subintegrals and the use of the rule of the triangle/diamond generates fewer terms compared to using invariants. Especially the simpler structure for integrating one-loop two-point subgraphs is important, since we apply this rule as early as possible to reduce the number of loops (and thus the number of parameters). Hence we choose to use dot products for the variables in the numerator in Forcer.

In the reduction routines we represent the integrals by a function Z with 14 variables (for fewer than four loops there will naturally be fewer variables) in which powers of variables in the denominators are given by positive numbers and powers in the numerator by negative numbers, as is commonly used. For example:

$$Z(1,1,1,1,1,1,1,1,1,1,-1,-2,-2,-2) \tag{39}$$

is a four-loop integral with four dot products. One dot product has power one, and the other three have two powers. Each of the 10 denominators has power 1. Note that all information about the topology or the choice of dot products is erased in this notation. Once the IBP relations are constructed, such information should be kept by different means. We note that some indices may be associated with propagators that have non-integer powers if insertions are involved (see section 3.5.3).

We define the integral in which all denominators have power one (possibly with an extra multiple of ϵ) and all numerators have power zero to have *complexity* zero. For each extra power of a denominator or of a numerator the complexity is increased by one. When we construct the IBPs parametrically the variables are represented by parameters $n_1 \ldots n_{14}$ in which at least three represent numerators. Now we define the integral with just $n_1 \ldots n_{14}$ as arguments to have complexity zero and again raising the value of a denominator by one, or subtracting one from a numerator raises the complexity by one. To improve readability, we represent denominators by parameters n and numerators by parameters k in the examples.

We redefine Z by adding *minus* the complexity as the first argument.[3] For example:

$$Z(-3;2,1,1,1,1,1,1,1,1,1,0,-1,-1), \tag{40}$$

$$Z(-1;n_1,n_2,n_3+1,n_4,n_5,n_6,n_7,n_8,n_9,n_{10},n_{11}-1,k_{12}+1,k_{13}-1,k_{14}-1). \tag{41}$$

In general, the goal is to construct a rule under which the basic complexity 0 integral $Z(0;n_1,\ldots,n_{14})$ is expressed in terms of other complexity 0 integrals or in terms of integrals with negative complexity.

The remainder of this section is structured as follows. We provide heuristics to find reduction rules in section 3.5.1. Next, we generate new IBP rules in section 3.5.2. We describe special rules for diagrams with non-integer powers in section 3.5.3. Finally, we summarise our solving strategy in section 3.5.4.

3.5.1 *Heuristics and equation generation*

For four-loop diagrams there are at first instance 20 unique IBP relations, formed from the operation $\frac{\partial}{\partial p^\mu} q^\mu$, where p is one of the four loop momenta and q is one of the four loop momenta or the external momentum. This set of equations can often be simplified by a Gaussian elimination of the more complex integrals. We call the simplified set of equations S_0. The most complex terms in S_0 have complexity 2 and have one raised denominator and one raised irreducible numerator. This is a direct consequence of the IBP structure.

In the set S_0 one can distinguish several types of reduction identities. The nicest identities are the ones that lower the complexity, sometimes even by more than one unit. An example is

$$\begin{aligned} 0 = {} & Z(-2;\ldots,n+1,\ldots,k-1,\ldots)\cdot n \\ & +Z(0;\ldots,n,\ldots,k,\ldots)+\ldots\ , \end{aligned} \tag{42}$$

where both a propagator and numerator are raised in the complexity 2 term. By shifting $n \to n-1$ and $k \to k+1$, we find the reduction rule:

$$Z(0;\ldots,n,\ldots,k,\ldots) = \frac{-1}{n-1}\left[Z(2;\ldots,n-1,\ldots,k+1,\ldots)+\ldots\right]. \tag{43}$$

Such identities are used for the simultaneous reduction of two variables. Since the equation will vanish once $n=1$ or 'overshoot' when $k=0$, it can only be used to speed up a reduction. Consequently, rules for the individual reduction of n and k are still required.

In what follows we will omit the last step of shifting the equation such that the highest complexity term becomes complexity 0. We also omit the coefficients of the Z functions when they are deemed irrelevant and we do not consider integrals with lines missing to be Z-integrals.

[3] We use minus the complexity, so that Form prints the integrals with the highest complexity first.

We now study relations that raise only one coefficient:

$$\begin{aligned} 0 = {} & Z(-1;\ldots,n+1,\ldots)\cdot n \\ & +Z(0;\ldots)+\ldots\,. \end{aligned} \tag{44}$$

Repeated application of this relation will either take the variable n down to one, or eventually create integrals in which one or more of the other lines are missing. For non-master topologies, at some point we find an equation that looks like

$$\begin{aligned} 0 = {} & Z(-1;\ldots,n+1,\ldots)\cdot(\epsilon+\ldots) \\ & +Z(0;\ldots)+\ldots\,. \end{aligned} \tag{45}$$

This equation has an ϵ in the coefficient, which means it does not vanish if $n = 1$. As a result, it can be used to reduce n to 0.

If after there is an equation in which the highest complexity is zero and the integral for which none of the parameters has been raised or lowered is present, there is a good chance that one can eliminate at least one line in that topology by repeated application of this identity, provided that there are no lines with a non-integer power. One example of such an equation is the rule of the triangle. The finding of more such equations while investigating the IBP systems of five-loop propagator diagrams led to the discovery of the diamond rule [10].

3.5.2 *Reduction rules beyond S_0*

Even though the triangle and diamond rule can be derived from equations in the set S_0, the set generally does not contain enough equations to reduce a topology straight away. Therefore, we expand our system by taking the set S_0 and constructing all equations in which either one denominator has been raised by one or one numerator lowered by one (which means that there is one more power of that variable because the numerators 'count negative'). This set is called S_1, since the IBPs are constructed from a complexity one integral. In total, we now have $20 + 280$ equations. Similarly we could construct the set S_2 by raising the complexity of one of the variables in the set S_1 in all possible ways, generating an additional 2100 equations. Usually S_2 is not needed. In some cases we may need a set like S_{-1} in which the complexity of one of the variables has been lowered, or even $S_{1,-1}$ in which one has been raised and one has been lowered.

The essence of our method is to construct the combined sets S_0 and S_1 and use Gaussian elimination to remove all objects of complexities 3 and 2 from the equations. The remaining equations only have objects of complexity one or lower. Out of these equations we construct an elimination scheme by defining an order of the variables, and we select for each variable an equation to eliminate it. For a denominator variable this is ideally an equation with a single term in which a variable n has been raised and all other parameters are at their default values: $Z(-1, n_1, n_2, \ldots, n+1, \ldots)$ Once we have such an equation we can lower $n + m$, with m being a positive integer, in all other equations to n. Since we know that after this either one of the other n_i will be 0 (meaning the reduction is done) or $n = 1$, we can assume from this point on that $n = 1$. Thus, in all other equations we now set n to 1, lowering the number of parameters by one. Similarly the numerator variables are worked up from $n - m$ to n after which this variable is given the value zero. The order of elimination and the selection of the equations is critical: one of our early carefully selected schemes resulted in a benchmark run of 53000 seconds, whereas a scheme with a different variable order and a more sophisticated combination of the equations, performed the same test in 555 seconds.

Above we gave an example of a simple, useful equation. However, sometimes these equations are not there. Below we discuss several other types of equations one may encounter. One example is if there are more integrals of complexity one:

$$\begin{aligned}0 = {} & Z(-1;\ldots)\\ & +Z(-1;\ldots)\\ & +Z(-1;\ldots)\\ & +Z(-1;\ldots,n+1,\ldots)\cdot n\\ & +Z(0;\ldots)+\ldots\ ,\end{aligned} \tag{46}$$

where the last complexity one term can be used to eliminate the variable n (provided that all other parameters have not been raised/lowered in this term), but this goes at the cost of increasing the number of terms with the same complexity. When the scheme is not carefully selected, the number of terms in the intermediate stages may become very large and the rational polynomials could become complicated.

A convenient subclass of the type shown in eq. (46) is one that increases an index in only a single term in the equation, independent of the complexity:

$$\begin{aligned}0 = {} & Z(-1;n_1+1,\ldots)\cdot n_1\\ & +Z(-1;n_1,\ldots)\\ & +Z(0;n_1,\ldots)\\ & +Z(1;n_1,\ldots)\\ & +\ldots\ .\end{aligned} \tag{47}$$

As a result, the equation can be used to lower the value of this variable at any level of complexity c:

$$\begin{aligned}0 = {} & Z(-c;n_1+1,\ldots)\cdot n_1\\ & +Z(-c;n_1,\ldots)\\ & +Z(-c+1;n_1,\ldots)\\ & +Z(-c+2;n_1,\ldots)\\ & +\ldots\ .\end{aligned} \tag{48}$$

We emphasise that we apply these equations to any value of n_1, so also to terms that look like $Z(-2,n_1+2,\ldots)$. In Form this can be done with a pattern match:

```
id Z(-c?,n1?,...,n14?) =  Z(-c,n1-1,...)/(n1-1)
                        + Z(-c+1,n1-1,...)/(n1-1)
                        + Z(-c+2,n1-1,...)/(n1-1)
                        + ...;
```

These equations are convenient because after applying them and after setting the variable to 1, there will not be a single term in the remaining equations in which there is a number greater than 1 in its position. We will later see why this is desirable.

The next type of equations also has more than one term at complexity one, but there is no clean reduction of a given variable:

$$\begin{aligned} 0 = {} & Z(-1;n_1+1,n_2-1,n_3+1,\ldots)\cdot n_1 \\ & +Z(-1;n_1+1,\ldots)\cdot n_1 \\ & +Z(-1;n_1,\ldots) \\ & +Z(-1;n_1,\ldots) \\ & +Z(0;\ldots)+\ldots\,. \end{aligned} \tag{49}$$

In the numerical case, one just moves the second term to the left and either n_1 will be reduced to one or n_2 will eventually become zero. However, in the derivation of the scheme one needs to apply this equation inside other parametric equations and more care is called for. One should apply the equation as many times as needed until terms either have n_1 (or n_1-1, etc.) or the n_2 position has n_2-1. This means that for the integral

$$Z(-1;n_1+1,n_2+2,\ldots) \tag{50}$$

equation (49) will have to be used up to three times. Once n_1 has been set equal to one, one may end up with terms such as

$$Z(-1;2,n_2-1,\ldots)\,, \tag{51}$$

which are undesirable. The solution to this problem is to try to deal with n_2 immediately after n_1. Once we can put n_2 equal to one, n_2-1 becomes zero and hence it is an integral with a missing line. If one waits with the n_2 reduction and does another variable first, one risks that n_2 is raised because the equation for the other variable could have a term with n_2+1 and then one would end up with an integral of the type

$$Z(-1;2,n_2,\ldots)\,. \tag{52}$$

This introduces either unresolved integrals or loops in the reduction scheme. It is also possible that one has reductions with two such conditions as we saw above. This requires great care in the selection of the next equation. We have not run into impossible situations at this stage.

Another case is one where the coefficient limits its application. For example

$$\begin{aligned} 0 = {} & Z(-1;n_1+1,\ldots)\cdot(n_1-n_2) \\ & +Z(0;\ldots)+\ldots \end{aligned} \tag{53}$$

cannot be applied when $n_1 = n_2$. Such rules could be very compact and are therefore used as a special case while the case in which n_1 is equal to n_2 is handled by a more general rule with less favourable properties but effectively one parameter less.

By far the most difficult equations are of the type

$$\begin{aligned} 0 = {} & Z(-1; n_1+1, n_2-1, n_3+1, \ldots) \cdot a(n_1, n_2, n_3) \\ & + Z(-1; n_1+1, n_2+1, n_3-1, \ldots) \cdot b(n_1, n_2, n_3) \\ & + Z(-1; n_1+1, \ldots) \\ & + Z(-1; n_1, \ldots) \\ & + Z(0; \ldots) + \ldots \;, \end{aligned} \tag{54}$$

where a and b are coefficient functions. We call this type a *yoyo*. As a recursion it will never end, because the values of n_2 and n_3 will keep going up and down. There are various ways to resolve this. The first is to construct a new type of recursion. This is done by applying the equation twice:

$$\begin{aligned} & Z(n_1+1, n_2, n_3, \ldots) \rightarrow \\ & \qquad + a(n_1, n_2, n_3) Z(n_1+1, n_2-1, n_3+1, \ldots) \\ & \qquad + b(n_1, n_2, n_3) Z(n_1+1, n_2+1, n_3-1, \ldots) + \ldots \\ & \quad \rightarrow + a(n_1, n_2, n_3) a(n_1, n_2-1, n_3+1) Z(n_1+1, n_2-2, n_3+2, \ldots) \\ & \qquad + b(n_1, n_2, n_3) b(n_1, n_2+1, n_3-1) Z(n_1+1, n_2+2, n_3-2, \ldots) \\ & \qquad + (a(n_1, n_2, n_3) b(n_1, n_2+1, n_3-1) + \\ & \qquad\quad b(n_1, n_2, n_3) a(n_1, n_2-1, n_3+1)) Z(n_1+1, n_2, n_3, ...) + \ldots \end{aligned} \tag{55}$$

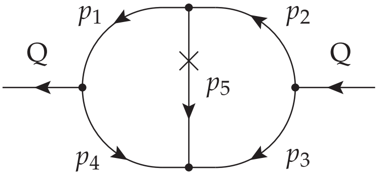
By moving the third term to the left one has a new recursion with a shift of two units. This procedure can be repeated i times until both $n_2 - 2^i$ and $n_3 - 2^i$ are less than one. The price to pay for this solution is high: fractions become enormously complicated and the number of terms could become very large.

An improved solution is to find another equation with a similar yoyo and combine the equations in such a way that one of the yoyo terms is eliminated. After this, one has a regular condition. We call this 'breaking the yoyo'. There is another way to break the yoyo that will be introduced below. We had to apply both methods of breaking the yoyo several times in the creation of the reduction schemes for the master topologies.

A final consideration is the structure of the coefficients of the integrals. In principle it is not very difficult to construct a reduction scheme from the available equations. The problem is that most schemes will end up with rational coefficients that take many megabytes to store because there are still quite a few variables in them. During the derivation this may cause problems with the limitations of the computer algebra system that is used (in our case Form). More importantly, the evaluation of such rational polynomials in the application of the reduction scheme to millions of integrals will render the reductions impossibly slow and hence useless for all practical purposes. Thus, if the coefficients are too large, an alternative reduction has to be found.

3.5.3 *Identities for topologies with insertions*

When a topology contains a line that does not have an integer power, the method of the previous section has to be slightly extended. Such cases occur either when the input diagram(s) can be written with a higher-order propagator in it, or when during the reduction a two-point function can be integrated out. If the resulting topology needs a custom reduction, we not only have to lower powers of denominators and numerators, but we also have to bring the powers of the non-integer lines to a canonical value, which we take to be $1 + m\epsilon$ for some positive integer m. As an example, we consider the two-loop `t1star05` topology (see also refs. [32, 33])



,

which has an ϵ in index 5, indicated by a single cross. We call such a cross an *insertion*. We have the relation:

$$Z_{\text{t1star05}}(n_1, n_2, n_3, n_4, n_5) = Z_{\text{t1}}(n_1, n_2, n_3, n_4, n_5 + \epsilon), \tag{56}$$

where the topology `t1` is the same two-loop topology but without any implicit non-integer powers. Since the ϵ can never be removed from the index during the reduction, we suppress it in our notation for `t1star05`. The IBPs for `t1star5` are generated from those of `t1` by a substitution $n_5 \to n_5 + \epsilon$. Typically, one tries to first reduce the integer indices $n_1, \ldots, n_4$ to 1. During these reductions, the contribution to the integral complexity from n_5 could be taken as the absolute value of the difference to penalise any change of n_5, or just be ignored to allow any change:

$$\textbf{Complexity}(n_5 + m_5) = |m_5|, \qquad \text{or} \quad \textbf{Complexity}(n_5 + m_5) = 0. \tag{57}$$

After all $n_1, \ldots, n_4$ are 1, we reduce the remaining index n_5 to 1, which may be positive or negative at this point. To derive a rule for the positive n_5 case, the complexity of n_5 can be defined as usual for a propagator:

$$\textbf{Complexity}(n_5 + m_5) = m_5, \qquad \text{for a rule with } n_5 > 1. \tag{58}$$

On the other hand, for the negative n_5 case, the complexity of n_5 can be defined as usual for a numerator:

$$\textbf{Complexity}(n_5 + m_5) = -m_5, \qquad \text{for a rule with } n_5 < 1. \tag{59}$$

In this way, all integrals belonging to `t1star05` can be reduced to the master integral $Z_{\text{t1star05}}(1, 1, 1, 1, 1)$ and integrals with simpler topologies.

3.5.4 *Solving strategy*

The heuristics for 'solving' a topology can now be outlined in a list of nine steps.

1. Select a numerator basis. The quality of the IBPs will depend on this choice.

2. Construct the IBP identities.

3. (Important) Use a type of Gaussian elimination to simplify the IBP identities, minimising the number of terms with the highest complexity. We call this set S_0. Most of the time this simplification can be done in an automated way. Only for the most difficult cases we have applied manual interference to obtain better results.

4. Construct the set S_1 by generating all possible options of raising an index in S_0. This gives terms of complexity 2 and 3. Use Gaussian elimination to eliminate all those terms. The remaining set of equations has terms of at most complexity 1.

5. (Important) Use the equations of the set S_0 (applying it to any complexity and configuration as in eq. 48) to eliminate as many complexity one terms as possible. This can simplify the following task and results in simpler formulas in the final reduction program. It also breaks up some difficult yoyos.

6. Determine an order of elimination of the variables. Often the first variables are rather obvious from the presence of simple reduction equations. Some variables may not be so obvious and one may have to experiment. The resulting programs may differ by orders of magnitude in their efficiency. Here is where either human intelligence, or a cleverly written AI program can help.

7. In many cases, one cannot find a decent equation for the last variable. This can be because either the results have become extremely lengthy, or one has discarded some long equations that seemed of no further relevance. In that case, the almost complete reduction scheme is applied to the set S_0. This will give a number of varieties of the final reduction(s). One can select the shortest one.

8. (Checking) Now apply the custom reduction scheme to the set S_0 with numbers for the variables and make sure that master integrals are indeed irreducible, and that the program does not get caught in loops. There may be equations remaining which only consist of integrals with missing lines. We did not take relations between those into account.

9. Combine all reductions and useful double reduction equations (equations that need at least two variables that are above their minimal complexity) based on S_0 or substitutions made during the Gaussian elimination. Together this forms the reduction procedure for the given topology.

In some cases the resulting schemes were still deemed too slow and more exhaustive methods were asked for. In such cases the sets S_2 and S_{-1} were also constructed and many different ways of combining the equations were tried automatically. Such programs could take much time because of the very complicated rational polynomials in the parameters of the integrals, but they eventually did result in a number of shorter reductions.

A number of Form procedures has been constructed to execute the above steps. The most laborious step is to determine a proper order for the elimination sequence, and which equations to use for each. Furthermore, we had a case (the bebe topology of section 3.6) in which there were no good reductions for two of the variables, unless we used two of the equations in the set S_0 to eliminate them with a complexity raising operation. It also reduced the number of remaining equations to 18 and hence left fewer options during the remaining parts of the derivation.

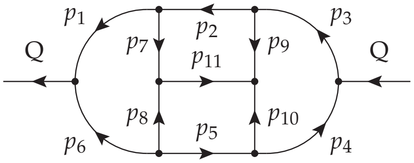
There are two major reasons why some reduction rules perform faster than others. The first reason is that even though a rule may have only one $Z(-1;...)$ term, it could be that the sub-leading terms increase the value of a variable that was set to 1 in one of the early steps of the scheme (see eq. 51). This forces the program back to an earlier reduction rule of the scheme, even though now at a lower complexity. The second reason is the coefficient growth: if a rule has a particularly complicated overall coefficient, it multiplies every term in the RHS and all subsequent terms will have rather lengthy rational polynomials in ϵ. Expanding in ϵ (see appendix 3.9) can alleviate some of these problems, provided one expands deep enough to avoid issues with spurious poles.

Determining the order of elimination seems suited for AI techniques, such as Monte Carlo Tree Search (see, e.g., [67]). One could use the number of top complexity terms, the number of lower complexity terms, the number of spectators and the size of the most complicated rational polynomial as parameters for an evaluation function for a given scheme and then use this in a MCTS to find an optimal scheme. This is currently under investigation. It should be noted that such type of use of AI for precisely this purpose was already hinted at in ref. [104].

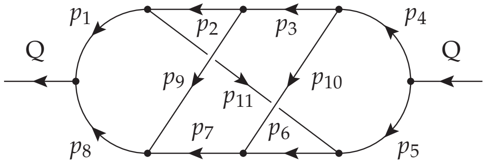
3.6 THE 21 TOPOLOGIES THAT NEED CUSTOM REDUCTIONS

Because we integrate over one-loop two-point functions, our classification of the master integrals differs from refs. [105, 106]. In general, any diagram that factorises we do not consider a master topology. The master diagrams that contain one-loop two-point functions that cannot be factorised, will have slightly different values, since we integrate out the bubble. The full list with the values of the master integrals in our convention are given in [1].

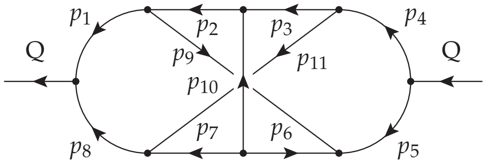
There are eight four-loop master integrals, excluding the diagrams in which a 2-point function can be integrated out.



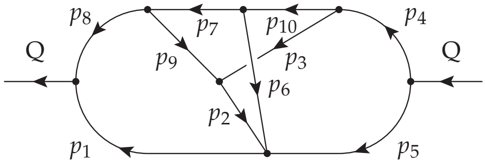
Topology name: **haha**, master.
Momenta: p_1, p_2, p_4, p_5.
Numerators: $2Q \cdot p_2, 2Q \cdot p_5, 2p_1 \cdot p_4$



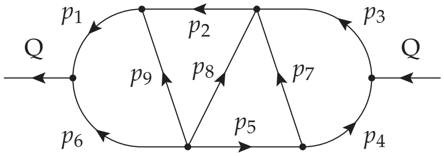
Topology name: **no1**, master.
Momenta: p_1, p_2, p_3, p_4.
Numerators: $2p_2 \cdot p_4, 2Q \cdot p_2, 2Q \cdot p_3$.



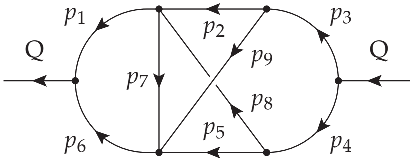
Topology name: **no2**, master.
Momenta: p_1, p_2, p_3, p_4.
Numerators: $2Q \cdot p_2, 2p_1 \cdot p_4, 2Q \cdot p_3$.



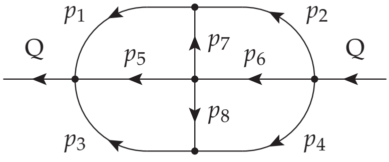
Topology name: **no6**, master.
Momenta: p_1, p_2, p_3, p_4.
Numerators: $2\ p_1 \cdot p_2, 2\ p_2 \cdot p_4, 2Q \cdot p_2, 2Q \cdot p_3$.



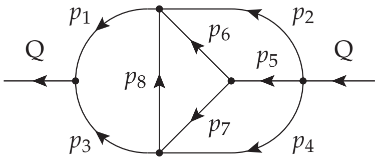
Topology name: **lala**, master.
Momenta: p_1, p_2, p_4, p_5.
Numerators: $2Q \cdot p_5, 2Q \cdot p_2, 2p_1 \cdot p_4,$
$2p_1 \cdot p_5, 2p_2 \cdot p_4$.



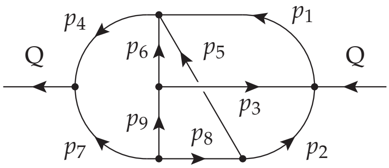
Topology name: **nono**, master.
Momenta: $p_1, p_2, p_3, p_{10} = p_2 + p_8$.
Numerators: $2\ p_2 \cdot p_8, 2\ p_6 \cdot p_7, 2Q \cdot p_2,$
$2p_1 \cdot p_2, 2p_7 \cdot p_9$.



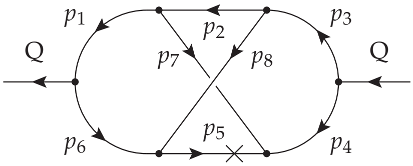
Topology name: **cross**, master.
Momenta: p_1, p_2, p_3, p_4.
Numerators: $2Q \cdot p_1, 2Q \cdot p_2, 2Q \cdot p_3,$
$2Q \cdot p_4, 2p_1 \cdot p_4, 2p_2 \cdot p_3$.



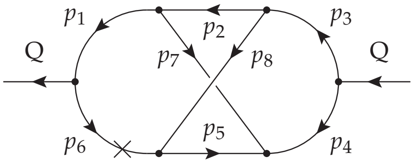
Topology name: **bebe**, master.
Momenta: p_1, p_2, p_4, p_6.
Numerators: $2Q \cdot p_2, 2Q \cdot p_4, 2Q \cdot p_6,$
$2p_1 \cdot p_2, 2p_2 \cdot p_6, 2p_1 \cdot p_4$.



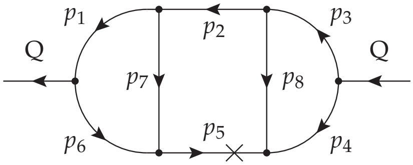
Topology name: **bubu**, not a master.
Momenta: p_2, p_3, p_8, p_9.
Numerators: $2Q \cdot p_2, 2Q \cdot p_8, 2p_2 \cdot p_3,$
$2p_2 \cdot p_9, 2p_3 \cdot p_8$.



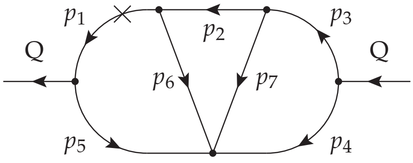
Topology name: **nostar5**, master.
Momenta: p_1, p_2, p_3.
Numerators: $2Q \cdot p_2$,



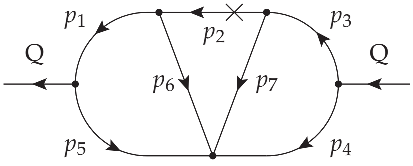
Topology name: **nostar6**, master.
Momenta: p_1, p_2, p_3.
Numerators: $2Q \cdot p_2$,



Topology name: **lastar5**, not a master.
Momenta: p_1, p_2, p_3.
Numerators: $2p_1 \cdot p_3$,



Topology name: **fastar1**, not a master.
Momenta: p_1, p_2, p_3.
Numerators: $2p_1 \cdot p_3, 2Q \cdot p_2$,



Topology name: **fastar2**, master.
Momenta: p_1, p_2, p_3.
Numerators: $2p_1 \cdot p_3, 2Q \cdot p_2$,

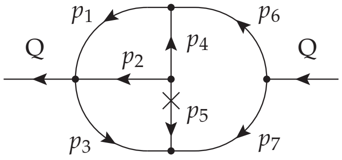
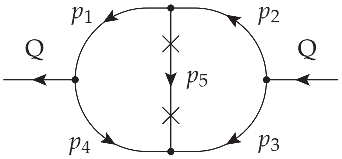
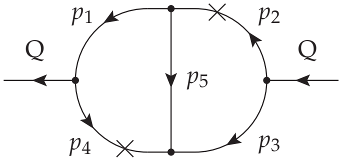
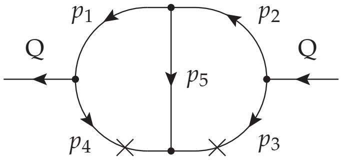
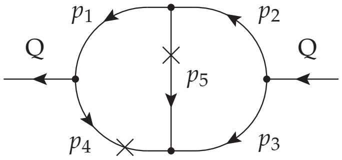
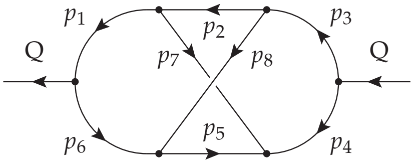
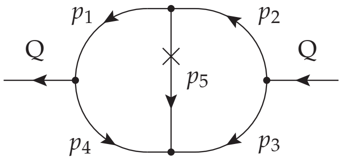
 p_1 p_6 Q p_4 Q p_2 p_5 p_3 p_7	Topology name: **bustar5**, not a master. Momenta: p_4, p_5, p_6. Numerators: $2Q \cdot p_4, 2Q \cdot p_5$,
	Topology name: **t1star55**, master. Momenta: p_1, p_2.
	Topology name: **t1star24**, master. Momenta: p_1, p_2.
	Topology name: **t1star34**, not a master. Momenta: p_1, p_2.
	Topology name: **t1star45**, master. Momenta: p_1, p_2.
 p_1 p_3 p_2 Q p_7 p_8 Q p_5 p_6 p_4	Topology name: **no**, master. Momenta: p_1, p_2, p_3. Numerators: $2Q \cdot p_2$, Remarks: Already in Mincer.
	Topology name: **t1star05**, master. Momenta: p_1, p_2. Remarks: Already present in Mincer.

Table 3: Table of all the topologies that require a custom reduction.

For these master integrals we have to design a custom scheme in which the parameters are reduced, one by one, to the value they have in the master integral. In addition (and perhaps surprisingly) there are four non-master topologies that need such a custom reduction.

Only when all but a few parameters are set to 1, do we find a relation to reduce an edge to 0. In this category there is one at the four-loop level, two at the three-loop level (with one non-integer edge) and one at the two-loop level (with two non-integer edges). In total we need 21 custom reduction schemes. All other topologies can be dealt with using generic formulas that can either eliminate a line or integrate out a loop. We list all topologies that need a custom reduction scheme in table 3.

In order to choose the best reduction schemes for the topologies in table 3, we measure the performance of a complete calculation of the integrals with all indices raised by 1 (a complexity 14 integral at four loops). By performing a complete calculation, we confirm that the number of terms with a simpler topology created by the reduction rules does not cause bottlenecks. Additionally, we confirm that for the case where all indices are raised by 2 (a complexity 28 integral at four loops), the reduction is still performing well.

We note that the ordering of variables in the reductions scheme is not the only relevant parameter. The choice of numerators can influence the presence of non-leading terms, which after the Gaussian elimination become leading terms. Such terms can spoil the efficiency of certain reduction rules. In particular the three complicated topologies nono, bebe, and no2 are sensitive to the choice of dot products.

Most schemes could be derived using the heuristics introduced in section 3.5, by selecting the reduction variable that corresponds to the shortest reduction rule. However, there are a few derivations that need more care. For nono, one needs to avoid a circular path in a special way. The formulas for the last two variables, n_4 and n_8, can only be obtained by reusing the original set S_0. At this point one uses either combinations of nearly all equations to obtain very lengthy formulas (> 1000 lines) or one uses a relatively short formula with a term that sends the reduction back to a previous rule, because it contains a term with $n_{11} = -1$. This would normally introduce a loop, but by sending only this term through the unfinished scheme and combining the result with the remaining part of the formula, we obtain a compact reduction formula for n_4 (39 terms).

The bebe reduction is more complicated as it does not yield a regular reduction for n_1 and n_3. However, in the set S_0 there are equations that can be used for their reduction, provided we are willing to raise the complexity. This does not agree with the automatic nature of our derivation tools, and hence some work needs to be guided by hand. Furthermore, we can no longer use a number of equations from the S_0 set for generating reduction rules for other variables. As a consequence, we are left with far fewer equations after the Gaussian eliminations, although their number is still sufficient for the next 11 variables. Eventually the n_2 variable has to be obtained again from the S_0 set.

For the construction of a reduction scheme the bubu topology is by far the most complicated, even though it is not a master topology. There are five different numerators and the elimination of the last numerator needs to be split into several cases, each with a rather lengthy formula involving complicated rational polynomials. In order to prevent a blow-up of terms, the order of elimination of the variables is critical, as well as using the equations obtained during the Gaussian elimination that give a direct reduction of the complexity. It took more than two months to find a first suitable reduction scheme.

We use the S_0 set and equations that come from the Gaussian elimination before we start with the 14 reduction identities of the complete schemes. This speeds up the reduction by a factor two or more, because these equations are usually much more compact and will often reduce the complexity immediately. It turns out that the final result is very sensitive to how we use these equations, because sometimes there are options when there is more than one term with the highest complexity, and also the order in which they are applied is relevant. Additionally one has to be careful with this ordering to avoid loops in the reduction. Unfortunately, it is not always possible to indicate which ordering is optimal, because some orderings may yield a faster scheme at the cost of more spectator terms and/or higher powers of ϵ in the rational polynomials.

Considering the amount of work involved in deriving the schemes, it is quite conceivable that better schemes will be found. It seems to be a good candidate for the application of automated AI techniques.

3.7 THE FORCER FRAMEWORK

In essence, the Forcer program provides a method to reduce each topology to simpler ones. There is quite some freedom: sometimes multiple reduction rules can be applied, sometimes it is best to use a different set of independent momenta, etc. In order to obtain the best performance, all decisions in the Forcer program are precomputed by a script: for each topology the action is hard-coded and the momentum rewrites are known. The advantage of this method is that costly optimisations, such as choosing an optimal basis for each topology, have no runtime cost.

The remainder of this section is structured as follows. In section 3.7.1 we describe the construction of the reduction graph. Next, the execution of the reduction graph is shown in section 3.7.2. Finally, we give an example of the treatment of a topology in section 3.7.3.

3.7.1 *Reduction graph generation*

Before going into details, we first give an overview of the program. The program structure can be viewed as a directed acyclic graph (DAG), the *reduction graph*, where the nodes are topologies and each edge indicates a transition from one topology to another when a propagator is removed. As a result, each node may have more than one parent. The root nodes of the reduction graph are the top-level topologies, which are topologies that only have three-point vertices. All tadpole topologies will be zero, so they are not included in the graph. To reduce the number of topologies, propagators with the same momentum are always merged.

Each node represents a topology, which consists of a graph with a certain fixed labelling of all the propagators, including momentum directions, and a fixed set of irreducible numerators. Each topology also has an action that determines how it can be reduced. They are, in order of preference: integration of a two-point function, carpet rule, triangle/diamond rule, or a custom reduction. Each topology contains transitions to other topologies for all removable edges (edges with integer power). Even though the specific rule may not be able to nullify any propagator in the graph, the dot product rewrites may, so therefore we generate all possible transitions. If there are lines missing, in most cases the topology action is not executed and the topology is automatically rewritten to another. The exception is for integrating insertions: insertions are guaranteed to reduce the number of loops, which simplifies the dot product basis. Thus, first rewriting the dot products to a new topology would be wasteful.

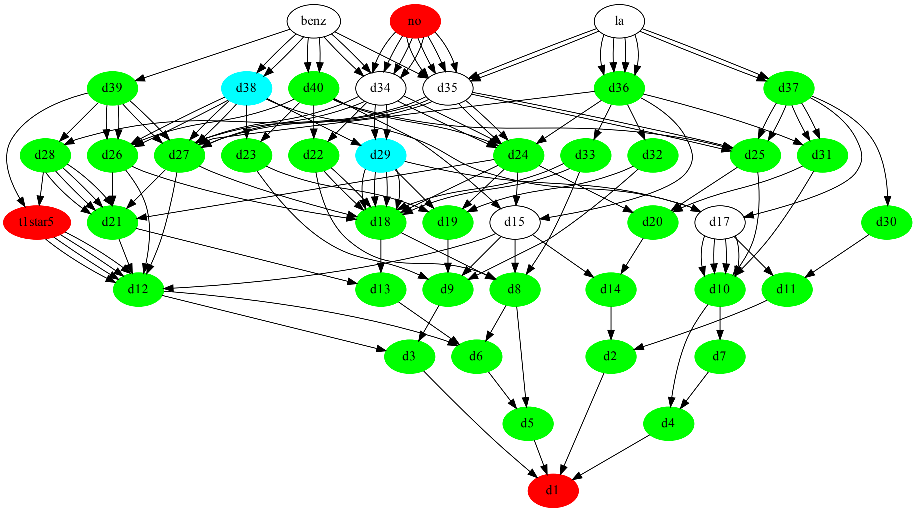


Figure 21: The three-loop reduction graph. Each node represents a topology, and each arrow a transition if a certain line is removed. The colour defines the topology action: white means the triangle or diamond rule, cyan the carpet rule, green the insertion rule, and red a custom reduction.

In figure 21 the reduction graph is displayed for three-loop massless propagator graphs. The names of the topologies are automatically generated. Every arrow denotes a transition that occurs when a propagator is removed. Multiply arrows could point to the same node if the resulting diagram is isomorphic. An example of this is `t1star5` (same as `t1star05` in section 3.6), where removing any of the four outside lines results in the same topology. The central line cannot be removed, since it has a non-integer power. The four-loop reduction graph, with over 400 nodes, is far too large to display.

The reduction graph is generated from the top-level topologies down. For every topology, a new one is generated where a particular line is missing, For this new topology, we determine its action. Next, we generate a dot product basis that is compatible with the action, e.g., for the insertion rule all dot products should only involve at most one of the two momenta. We also determine its automorphisms (graph symmetries), so that we can map every topology instance to the same unique form (we will go into more detail about this in the next section).

The dot product basis is chosen according to the following three rules: (1) it is compatible with the action, (2) it minimises the number of terms created when rewriting from the parent topology. As a criterion we choose the sum of the square of the number of terms that are created in rewriting each dot product. (3) The dot products are chosen in line with the symmetries of the topology.

We summarise the generation of the reduction graph in algorithm 2.

```
Input  : top-level topologies $T$
Output: reduction tree $T_{\text{all}}$
$T_{\text{all}} \leftarrow T$ ;
foreach $t \in T$ do determine action and
  automorphisms;
while $T \neq \emptyset$ do
    take $t \in T$;
    foreach propagator $p \in t$ do
        $h \leftarrow$ new topology without $p$ ;
        if $h' \in T_{all}$ isomorphic to $h$ then
            construct mapping from $h \to h'$;
        else
            determine action for $h$;
            generate dot product basis for $h$;
            generate automorphisms for $h$;
            generate mapping of dot products from
              $t$ to $h$;
            add node $h$ to tree $T_{\text{all}}$;
            $T \leftarrow T \cup \{h\}$;
        end
    end
end
```

Algorithm 2: Reduction graph generation.

The reduction graph is generated with a Python script, using `igraph` [107] for a basic graph representation of the topologies and for the isomorphism algorithm. Since by default only simple graphs (without self-edges and duplicate edges) are supported by the isomorphism algorithm, we merge all double edges and use a custom function to determine if the topologies are truly isomorphic (one could view duplicate edges with possible insertions as a special edge colouring). This function enforces that the number of duplicate edges is the same, and that the distribution of insertions over duplicate edges is the same. Additionally, we generate all possible permutations over similar duplicate edges, to generate the edge isomorphisms. The reduction graph contains 438 topologies and requires 40 000 lines of Form code.

3.7.2 *Reduction graph execution*

So far, we have discussed the generation of the reduction graph. Now we consider how the graph is processed in runtime.

As input, we have integrals that are labelled by the name of their topology in a symbol. In contrast to Mincer, the input expressions can contain multiple topologies. In Forcer, every topology is put in a separate expression and is hidden. The topologies are processed one by one, in the order of the number of edges. When a topology is treated, the expression is unhidden, the integrals are symmetrised using automorphisms, the topology action is executed, and finally, the resulting integrals are rewritten to their new topology. The topologies in the output are either master integrals, which require no further reductions, or topologies with fewer lines. These topologies will be merged into the designated expression for that topology. All the masters integrals are stored in their own expression.

After rewriting dot products, multiple edges could have vanished. Some of the integrals that remain could have become massless tadpoles, which are zero in dimensional regularisation. A table is used that maps the topology and a list of missing edges to zero if the resulting topology is a tadpole.

The execution of the reduction graph is summarised in algorithm 3.

3.7.3 *Example*

Below we give an example of the treatment of a topology. The topology is depicted in figure 22, and is internally called `d366`.

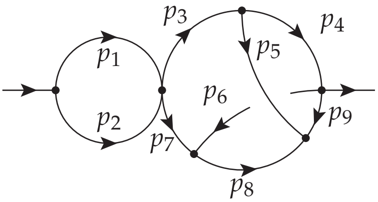
In the input, the integral is represented as follows by a compact notation in terms of symbols only:

```
Md366/i1/i2/i3/i4^2/i5/i6/i7/i8/i9*i10*i11*i13;
```

where `Md366` is the marker of the topology and the powers of $\texttt{i}_n$ represent the propagator and numerator powers. In this example we have three additional powers: $1/p_4^2$, $Q \cdot p_4$, $p_1 \cdot p_6$, and $p_1 \cdot p_4$. Since all rules are precomputed, the information of the topology such as the vertex structure, momentum flow, non-integer powers of lines and which dot products are in the basis, is never stored in the terms that are processed. Instead, the topology marker `Md366` will be used to call the correct routines.

```
Input   : input integrals $I$
Input   : reduction graph $T_{\text{all}}$
convert $I$ to FORCER topologies ;
foreach $t \in I$ do  put in its own expression $E_t$ and
  deactivate;
for $l = 11$ to 1 do
    foreach $t \in T_{all}$ with $l$ edges (any order) do
        activate expression with topologies $t$ ;
        symmetrise terms (apply automorphisms) ;
        perform reduction operation (triangle,
          carpet, etc.) ;
        rewrite result with missing lines to FORCER
          topologies $h_i \in T_{\text{all}}$ ;
        move the terms with topology $h_i$ to $E_{h_i}$;
    end
end
```

Algorithm 3: Reduction graph execution.



$$n_{10} = Q \cdot p_4 \qquad n_{13} = p_1 \cdot p_4$$
$$n_{11} = Q \cdot p_6 \qquad n_{14} = p_1 \cdot p_6$$
$$n_{12} = p_1 \cdot p_3$$

Figure 22: FORCER topology d366.

When treating topology d366, we first apply symmetries to make sure that similar configurations of d366 are merged. We use the automorphisms of the graph, of which there are four: $(p_1 \leftrightarrow p_2) \times (p_4 \leftrightarrow p_6, p_3 \leftrightarrow p_7, p_7 \leftrightarrow p_8)$. However, since there may be dot products in these momenta, the symmetry may be broken unless the set of dot products maps into itself. For the symmetry $(p_1 \leftrightarrow p_2)$, the dot products $p_1 \cdot p_3$, $p_1 \cdot p_4$, $p_1 \cdot p_6$ should be absent. The other symmetry can only be applied when $p_1 \cdot p_3$ is absent.

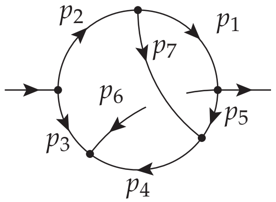
To find the smallest isomorphism, we hash the powers of the i, and take the smallest. The code is displayed in listing 3.2.

LISTING 3.2: FORM code for finding the smallest isomorphism

```
if (match(Md366*<1/i1^n1?$n1>*...*<1/i14^n14?$n14>));
  if (($n12==0)&&($n13==0)&&($n14==0));
    #call hash(0,$n14,$n13,$n12,$n11,$n10,$n9,$n8,$n7,$n6,$n5,$n4,$n3,
         $n1,$n2)
    #call hash(1,$n14,$n13,$n12,$n10,$n11,$n9,$n5,$n3,$n4,$n8,$n6,$n7,
        $n1,$n2)
  endif;
  if (($n12==0));
    #call hash(2,$n14,$n13,$n12,$n11,$n10,$n9,$n8,$n7,$n6,$n5,$n4,$n3,
        $n2,$n1)
    #call hash(3,$n13,$n14,$n12,$n10,$n11,$n9,$n5,$n3,$n4,$n8,$n6,$n7,
        $n2,$n1)
  endif;
* stores best hash in $bestiso
  #call smallesthash(0,1,2,3)
  if ($bestiso == 0); Multiply replace_(i1,i2,i2,i1);
  elseif ($bestiso == 1); Multiply sign_($n10+$n11+$n13+$n14)
    *replace_(i1,i2,i2,i1,i3,i7,i4,i6,i5,i8,i6,i4,i7,i3,i8,i5,i10,i11,
        i11,i10);
  elseif ($bestiso == 3); Multiply sign_($n10+$n11+$n13+$n14)
    *replace_(i3,i7,i4,i6,i5,i8,i6,i4,i7,i3,i8,i5,i10,i11,i11,i10,i13,
        i14,i14,i13);
  endif;
endif;
```

The action that will be performed in d366 is the integration of the left bubble, p_1 and p_2. As can be seen in figure 22, all relevant dot products are written only in terms of p_1 and none in terms of p_2, in alignment with the insertion rule. The dot products that involve p_1 can all be re-expressed in terms of inverse propagators after integrating the insertion. The two dot products that remain, $Q \cdot p_4$, and $Q \cdot p_6$ (represented by i10 and i11 respectively) have to be rewritten to the new topology.

The new topology is called d118:



$$n_8 = Q \cdot p_4$$
$$n_9 = Q \cdot p_7$$

where we have suppressed the ϵ power of the external line.

In listing 3.3, we display the mapping from d366 to d118, which includes rewriting the old dot products.

Listing 3.3: Form code for rewriting dot products

```
Multiply replace_(i3,j2,i4,j1,i5,j7,i6,j6,i7,j3,i8,j4,i9,j5);
id i10 = Q^2/2+j2/2-j3/2-j9;
id i11 = -Q^2/2+j2/2-j3/2+j8;
Multiply replace_(Md366,Md118,<j1,i1>,...,<j7,i7>,j8,-i8,j9,i9);
```

3.8 usage

The Forcer program can be downloaded from `https://github.com/benruijl/forcer`. Currently, the latest development version of Form is required, which can be obtained from `https://github.com/vermaseren/form`. The generation scripts require Python 2.7, Python 3 or higher as well igraph [107], numpy [108] and sympy [109].

An example of Forcer input is displayed in listing 3.4.

Listing 3.4: Example input for Forcer

```
#-
#include forcer.h

L F =
 +1/<p1.p1>/.../<p6.p6>*Q.p3*Q.p4*vx(Q,p1,p5,p6)*vx(-p1,p2,p3)*vx(-p5,-
     p6,p4)*vx(-Q,-p2,-p3,-p4)
 +1/<p1.p1>/.../<p5.p5>*vx(-Q,p2,p3)*vx(p1,-p2,p5)*vx(-p1,p4,Q)*vx(-p3,-
     p4,-p5)*ex(p1,p4)
;

#call Forcer(msbarexpand=4)
B ep;
P +s;
.end
```

After `forcer.h` is included, the input integral can be defined. This is done by specifying the vertex structure using `vx`. The external momentum should be called `Q`. The propagators and momenta can simply be multiplied in, as shown in the example above. Insertions on lines can be specified using the `ex` function. In the second integral above `ex(p1,p4)` means that there is a single ϵ on the propagator associated with momentum p_1, and one on p_4. The topologies will automatically be matched to Forcer's internal topologies. The dot products will also automatically be rewritten (see subsection 3.10.4).

By calling the `Forcer` procedure, the integrals are computed. The optional argument `msbarexpand` can give the (unrenormalised) answer expanded in $\overline{\text{MS}}$. Otherwise, the result will be given exactly in terms of the master integrals and rational coefficients. Other options include `polyratfunexpand=div` and `polyratfunexpand=`

maxpow, which enable the expansions of rational coefficients in ϵ at intermediate steps using the Form statement PolyRatFun (see 3.9).

3.9 expansions

In principle the coefficients of the integrals can be kept as rational polynomials in D or ϵ. However, the nature of the reductions is such that these polynomials can contain very high powers in their numerators and denominators. Adding such rational polynomials is easily the most costly operation during the reductions. During the development of the Forcer program, we have encountered polynomials with powers of ϵ that went over 700, and that was not even for a complete reduction. In practice one needs such 'precision' only in very rare cases, such as when one needs to change dimensions during or after the reduction. In our program this is not necessary, and hence a better strategy is to expand these polynomials to a finite power of ϵ. The main problem is that we do not know in advance how many powers are needed. The reductions will at times generate extra powers of $1/\epsilon$ (spurious poles) that will only cancel near the end of the reduction when all terms that contribute to a given master integral are added. An exact solution for the spurious problem is a denominator notation [110], but to make this workable Form still needs supporting facilities.

We have opted for a method in which the reduction formulas still use rational polynomials, but after each step they are expanded to sufficient depth. It is possible to make a special trial run to determine how many powers are needed. In this trial run only the minimum power of ϵ is kept with the coefficient one, to avoid that such terms can cancel. Avoiding all calculations, such a run can be relatively fast, provided that the main computational effort is in the Forcer part of the program (it usually is). After the run, one can see how deep the expansions have to be. We usually take the worst value that we encounter for all diagrams and add one 'guard power'. With this value the program generates the proper tables for the various Pochhammer symbols and other objects that may need expansions. Then during the actual reductions the rational polynomials will be expanded to the proper depth.

A simpler and safer method is to run the whole calculation twice with different settings for the expansion depth and observe at which power of ϵ the coefficients change. This is similar to running numerical programs with different floating point precisions to study the numerical instabilities.

Form has options to use expansions in its coefficients. The command PolyRatFun, rat(divergence,variable) keeps only the lowest power of ϵ. Generally, the program is quite fast in this mode. To expand, the statement PolyRatFun,rat(expand, variable,maxpow) can be used. These commands are implemented in the latest development version of Form.

The interface provided in the previous section expects scalar integrals as input. In order to compute Feynman diagrams, process-specific preprocessing has to be performed. Since the actual implementation is highly dependent on conventions, we will only sketch certain parts.

The program Qgraf [111] provides a convenient way to generate the Feynman graphs that are needed for the actual calculations, because it can generate Form compatible output. However, the challenge remains of converting the diagrams as presented by Qgraf to something that the Forcer program can deal with. This involves mapping the topology and momenta of the diagrams to Forcer's internal notation. For this purpose, the Python program that generates the reduction graph also generates a file called `notation.h` which contains a specification of all topologies in such a way that a conversion program can use it for

1. topology recognition,
2. labelling the momenta and their directions for each line,
3. using symmetry transformations.

Each topology is represented by a term in Form notation. Two typical terms are displayed in listing 3.5.

Listing 3.5: Two entries in the notation file

```
+vx(-Q,p4,p5)
 *vx(p3,-p4,p11)
 *vx(p6,p7,p10)
 *vx(p2,-p3,-p10)
 *vx(p1,-p2,p9)
 *vx(-p5,-p6,-p9)
 *vx(-p7,p8,-p11)
 *vx(-p1,-p8,Q)
 *SYM()
 *SYM(Q,-Q,p1,-p5,p2,p6,p3,-p7,p4,-p8,p5,-p1,p6,p2,p7,-p3,p8,-p4,p9,-
     p9,p10,-p10,p11,-p11)
 *SYM(Q,-Q,p1,-p4,p2,-p3,p3,-p2,p4,-p1,p5,-p8,p6,p7,p7,p6,p8,-p5,p9,
     p11,p11,p9)
 *SYM(p1,p8,p2,p7,p3,-p6,p4,p5,p5,p4,p6,-p3,p7,p2,p8,p1,p9,-p11,p10,-
     p10,p11,-p9)
 *TOPO(Mno2)

+vx(-Q,p3,p4)
 *vx(p2,-p3,p7)
 *vx(p1,-p2,p6)
 *vx(-p1,p5,Q)
```

```
    *vx(-p4,-p5,-p6,-p7)
    *ex(p2)
    *SYM()
    *SYM(Q,-Q,p1,-p3,p2,-p2,p3,-p1,p4,p5,p5,p4,p6,p7,p7,p6)
    *TOPO(Mfastar2)
```

The first term indicates the no2 topology. The function vx indicates the vertices and the momenta belonging to that vertex. Negative momenta are incoming. The function TOPO has a symbol as an argument that indicates the topology. In the Forcer program terms that are in the notation of a given topology are labelled with one power of the corresponding symbol. The function SYM describes a symmetry operation of the topology. The Form statement

```
id,once,SYM(?a) = replace_(?a);
```

will execute such an operation. In practice one could use it as described in listing 3.6.

Listing 3.6: Form code for applying symmetries

```
    id vx(?a) = f1(vx(?a));
    repeat id f1(x1?)*f1(x2?) = f1(x1*x2);
    repeat id SYM(?a)*f1(x?) = f1(x)*f2(x*replace_(?a));
    id f1(x1?)*f2(x2?) = x2;
    id f2(x?) = 1;
```

This process makes for each occurrence of the function SYM a copy of the contents of the function f1 in which the corresponding symmetry operation has been applied. Because the normal ordering algorithm of Form puts the smallest of the functions f2 first, we end up with the smallest representation of the term. If this is applied at a later stage in the program more statements may be needed, because there may be more objects than vx.

The notation file includes more topologies than actually exist in the Forcer reduction graph, since physical diagrams can have duplicate momenta. If this is the case, the term in the notation file also contains a function ID, for example ID(p4,-p5), indicating that p_4 and $-p_5$ are actually the same momentum. After the topology is matched and the labelling is done, the following ID function can be applied.

```
id ID(p1?,p2?) = replace_(p1,p2);
```

The first step in determining the topology of a diagram is to read the notation.h file, number its topologies, and store each of them in a dollar variable with a name that is labelled by this number. We also store the names of the topologies in such an array of dollar variables. The topology of a diagram can now be determined by trying whether one of the topologies can be substituted in the term. If this pattern matching involves wildcards, and the match of the wildcards is stored inside dollar variables we can use this to relabel the diagram itself and bring it to the notation of the topology. The main problem is creating the match structure, since we need wildcards for all the momenta followed by the name of a dollar variable. This issue is resolved with the dictionary feature of Form. The essential part of the code is shown in listing 3.7.

Listing 3.7: Form code for topology matching

```
#OpenDictionary wildmom
    #do i = 1,‘$MAXPROPS’
        #add p‘i’: "p‘i’?$p‘i’"
    #enddo
#CloseDictionary

#do i = 1,‘$MAXPROPS’
    $p‘i’ = p‘i’;
#enddo

#UseDictionary wildmom($)
#do i = 1,‘$numtopo’
    if ( match(‘$topo‘i’’) );
        $toponum = ‘i’;
        goto caught;
    endif;
#enddo
#CloseDictionary
label caught;

Multiply replace_(Q,Q,<$p1,p1>,...,<$p‘$MAXPROPS’,p‘$MAXPROPS’>)*topo(
     $toponum);
```

When we try to match, the printing of the ‘$topo‘i’’ variable will result in objects like vx(p1?$p1,p2?$p2,p3?$p3)*... rather than the vx(p1,p2,p3)*... that it actually contains. This way the $-variables get the value of the momenta in the diagram for which we want to determine the topology and the notation. The final replace substitutes these momenta by the value they have in the topology file.

It is possible to speed up the process considerably by hashing the topologies by the number of vertices and by first stripping the signs of the momenta. These signs can be recovered in a later step.

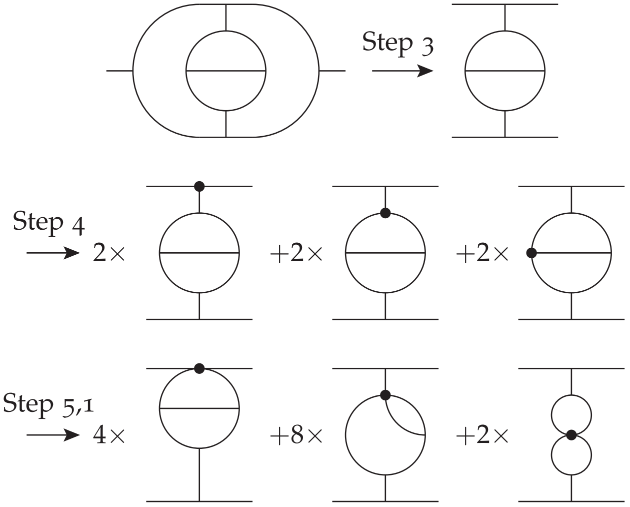
The remainder of this section is structured as follows. In section 3.10.1 we describe how to filter self-energies. The colour split-off is shown in section 3.10.2. In section 3.10.3 we describe the diagram database. Finally, we construct the momentum substitution routines in section 3.10.4.

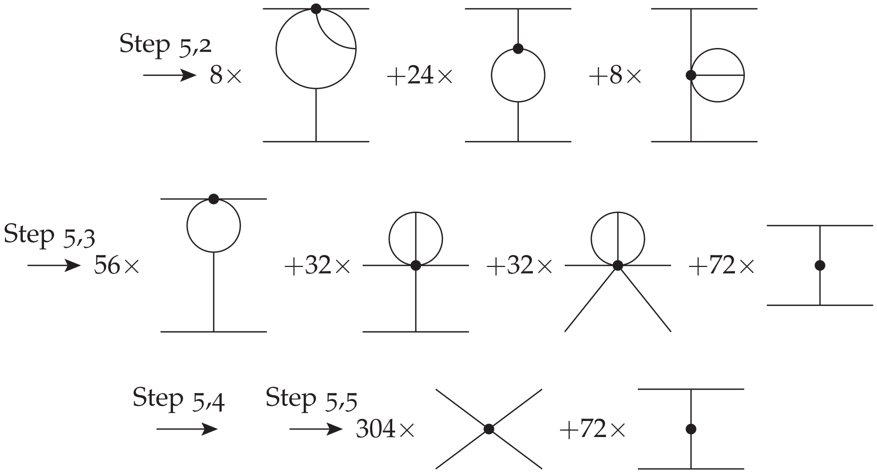
3.10.1 *Self-energy filtering*

An optimisation is to filter self-energy insertions from the QGRAF output. Here we present an algorithm that can detect one particle *reducible* propagator insertions.

1. Select a representative for a one-loop propagator. A representative is a single diagram that occurs in this propagator. For the ghost and the quark propagators this is trivial, since there is only a single diagram. For the gluon we select the diagram with the ghost loop (not forgetting the minus sign).
2. In the propagators we indicate the number of loops with an extra parameter. Adjacent loop representatives are combined and their number of loops is the sum of those parameters. This means that the representative of a three-loop gluon propagator is a chain of three one-loop diagrams, each with a ghost loop.
3. Next we make a copy of all remaining vertices into a function `acc`. In this function we remove all vertices that have an external line.
4. In the function `acc` we start selecting one vertex in all possible ways.
5. If this special vertex has more than two lines, it 'consumes' in all possible ways one of its neighbouring vertices, removing the connecting momentum. If the same momentum connects twice to the new vertex, it is removed as well.
6. We keep doing this until either the super-vertex in one of the terms has two lines remaining in which case we can eliminate the whole diagram as it is part of a propagator, or we cannot remove any more lines. If all possibilities end in the last way we keep the diagram.

Let us show this diagrammatically for a non-trivial diagram:





In the example, the diagram can be eliminated at the moment the super-vertex with just two lines appears. This is at step 5,3. We did not stop at that point because we wanted to show how the other possibilities develop for diagrams that would survive.

The above algorithm can be programmed rather easily in Form with the new id,all option of the id statement. For instance step 4 is just the statement

```
id,all,v(?a) = w(?a);
```

in which `v` represents the vertices and `w` is the super-vertex. This is followed by a symmetrisation to reduce the number of different diagrams. A complete procedure that works for all types of diagrams, independent of the number of external lines or loops contains 30 Form statements. The elimination of insertions simplifies the calculation considerably, because multi-loop gluon propagator insertions have many diagrams. This is particularly important when calculating moments of splitting and coefficient functions in DIS.

3.10.2 *Colour split-off*

We split each diagram in its colour part and its 'Lorentz' part before applying the Feynman rules. The 4-gluon vertex is split up into three terms with their own overall colour factor. Technically it is not required to do the split-off at this stage, but the remaining program will be considerably faster when the colour is a global factor.

To compute the colour factor we use a modified version of the `color` package of ref. [112]. It has been observed that even when one may have 100 000 diagrams or more, there are usually at most a few hundred different colour factors to be worked out. Hence, the way to process these factors is by pulling all colour objects into a function `color` and then, after using colour projectors on the external lines, only working out the colour bracket. The process is shown in listing 3.8.

Listing 3.8: Form code for backeting in a function

```
Normalize color;
B color;
.sort: Prepare color;
Keep brackets;
Argument color;
  #call color
  #call simpli
EndArgument;
```

Replacing every `.sort` by the procedure described in listing 3.9, guarantees that each different colour object is worked out only once.

Listing 3.9: Form code for a bracketed sort

```
#procedure SORT(text)
  EndArgument;
  B color;
  .sort:‘text’;
  Keep Brackets;
  Argument color;
#endprocedure
```

3.10.3 *Diagram database*

Diagrams with the same topology and colour factor are grouped together in superdiagrams. The superdiagrams provide a convenient way to distribute the work over multiple computers. This grouping can speed up the calculation by a factor three.

We use the minos database program provided (with its source code) in the Form pages to store the superdiagrams. After each superdiagram is computed, it is multiplied with its colour factors. Finally, the values of all superdiagrams are added. Only at this stage do we substitute the formulas for the insertion propagators and the master integrals. Up until the substitution of the master integrals the results are exact to all orders in ϵ if one uses rational polynomials in ϵ for the coefficients of the terms.

3.10.4 *Momentum substitutions*

After the Feynman rules have been applied, the integrals are in a form in which they can be converted to Forcer's basis for the topologies. The reduction to this basis needs to be done with great care as it is quite easy to generate an extremely large number of terms. This process is split up into two components: rewriting the momenta to a momentum basis and rewriting the dot products to Forcer's basis.

The momentum basis should contain all the momenta of the irreducible dot products belonging to this Forcer topology. The other basis elements are obtained by an exhaustive search that tries to minimise the number of terms that will be created when rewriting to the basis. The optimisation criterion is the sum of the square of the number of terms that get created for all the momentum and dot product rewrites.

In order to prevent a blow-up in the number of terms, we create a layered rewrite of momenta. This layering is constructed automatically and makes the momentum rewrites order dependent:

```
p9.p?!{p9,}=+p2.p+p7.p+p11.p-Q.p;
p5.p?!{p5,}=-p11.p-p3.p+Q.p;
p6.p?!{p6,}=-p2.p+p3.p-p7.p;
p1.p?!{p1,p4}=+Q.p-p8.p;
p10.p?!{p10,}=+p2.p-p3.p;
p4.p?!{p4,p1}=+p11.p+p3.p;
p7.p?!{p7,}=+p8.p-p11.p;
```

$\longrightarrow$

```
p9.p?!{p9,}=-p6.p-p5.p;
p5.p?!{p5,}=-p4.p+Q.p;
p6.p?!{p6,}=-p10.p-p7.p;
p1.p?!{p1,p4}=+Q.p-p8.p;
p10.p?!{p10,}=+p2.p-p3.p;
p4.p?!{p4,p1}=+p11.p+p3.p;
p7.p?!{p7,}=+p8.p-p11.p;
```

Because some terms will merge during the momentum rewrites, the layered approach is much faster. Note that dot products will not be rewritten if they are elements of the dot product basis.

Finally, the dot products are rewritten, straight to the internal Forcer notation. This is displayed in listing 3.10.

Listing 3.10: Form code for dot product rewriting

```
id Q.p1 = Q^2/2+i1/2-i8/2;
id p1.p2 = i1/2+i2/2-i9/2;
id p2.p3 = -i10/2+i2/2+i3/2;
id Q.p4 = Q^2/2+i4/2-i5/2;
id p3.p4 = -i11/2+i3/2+i4/2;
id p1.p3 = -Q^2/2+i11/2+i13+i14-i4/2+i5/2-i7/2+i8/2;
id p2.p4 = -Q^2/2-i1/2+i12+i13+i5/2-i6/2+i8/2+i9/2;
```

We note that in the actual code there will be `.sort` statements between the id statements and that there are extra optimisations in place to prevent excessive term generation.

3.11 examples and performance

The Forcer program has recently been used in many large calculations. As a first demonstration of its capabilities, the four-loop QCD beta function has been recomputed [7, 9], and it agrees with refs. [113, 114]. Other major computations (refs. [2, 4, 5, 115, 116]) will be discussed in chapter 4 and chapter 6.

Below we demonstrate some benchmarks of the Forcer program. We start with some specific configurations, displayed in table 4. We have chosen top-level topologies for the benchmark, since these are the most time-consuming ones. In their reduction, many other master topologies (and thus custom reductions) are encountered. The topology la4 is the four-loop ladder topology.

ID	Configuration	Time (s)
no1	$Z(-14;2,2,2,2,2,2,2,2,2,2,2,-1,-1,-1)$	10476
no2	$Z(-14;2,2,2,2,2,2,2,2,2,2,2,-1,-1,-1)$	147
haha	$Z(-14;2,2,2,2,2,2,2,2,2,2,2,-1,-1,-1)$	338
la4	$Z(-14;2,2,2,2,2,2,2,2,2,2,2,-1,-1,-1)$	68
no2	$Z(-17;2,2,2,2,2,2,2,2,2,2,2,-2,-2,-2)$	370
la4	$Z(-20;2,2,3,2,2,2,2,3,2,2,2,-2,-2,-3)$	2848
haha	$Z(-20;2,1,2,2,1,2,1,2,2,2,2,-4,-4,-4)$	12943
la4	$Z(-20;2,1,2,2,1,2,1,2,2,2,2,-4,-4,-4)$	117906

Table 4: Benchmark for several specific configurations, using 4 cores.

Next, we compute samples of configurations with a specific complexity of the top-level non-planar master integral `no1` and `no2`. In figure 23, we show the total wall-clock time of computing 10 and 100 samples for a given complexity at the same time, using 4 cores. We observe that even though the difference in number of samples is a factor 10, the computation time increases only by about 20%. This demonstrates that the Forcer program makes use of symmetries and grouping, which cause shared configurations deeper in the reduction process to merge. Additionally, the graph shows that the computation time scales exponentially in the complexity with a base of about 2.5.

Finally, figure 24 shows the timings for computing four-loop QCD self-energies for a certain maximum power of the (unrenormalised) gauge parameter ξ. Here $\xi = 0$ corresponds to the Feynman gauge. In our setup, all techniques discussed in section 3.10 are applied.

The background field propagator in figure 24 can be used to obtain the beta function without computing an additional propagator and a vertex [117, 118]. Interestingly, the curve for the background-gluon is quite similar to that for the gluon, even though one may expect that the background-gluon to be more time consuming than the gluon propagator because of extra vertices. The high performance can be understood by the fact that we are using superdiagrams; as we have seen in figure 23, the increase of the number of terms does not matter much, provided complexities of integrals are similar, and there are many chances for merges and cancellations of coefficients of the integrals at intermediate stages in the reduction.

Using the background field method, we are able to compute the four-loop beta function for Yang-Mills theory with fermions in less than three minutes in the

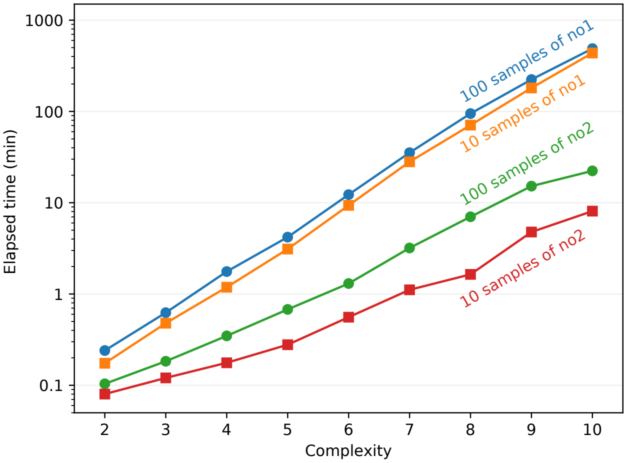


Figure 23: A benchmark (wall time) for the complete reduction of `no1` and `no2` configurations, using 4 cores (2.6 GHz). The line with the dots indicates the joint computation time of 100 sampled configurations, the line with the squares the computation time of 10 samples. Even though 10 times more integrals are computed, the computation time is only 20% longer. The scaling in complexity is exponential: each increase in complexity increases the computation time by 2.5.

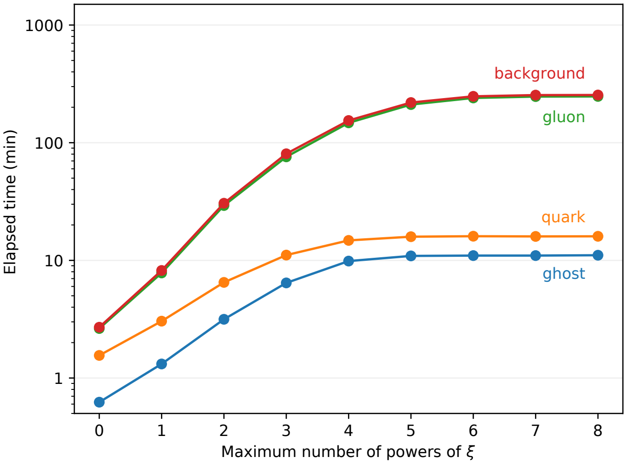


Figure 24: A benchmark (wall time) for computing four-loop QCD self-energies (ghost, quark, gluon and background-gluon) on a 32-core machine. The two curves for the gluon and background-gluon almost coincide.

Feynman gauge, and in about four hours for all powers of the gauge parameter on a single machine with 32 cores.

3.12 CHAPTER CONCLUSION

We are now able to answer

> **RQ2:** *How can we construct a program that can compute four-loop massless propagator integrals more efficiently?*

We have shown how the Forcer program has been constructed, what algorithms it uses and demonstrated its performance [1]. We have derived the diamond rule, which is able to efficiently reduce integrals with a certain substructure [10]. Next, we have shown how to derive parametric reduction rules. In addition, we have outlined how Forcer may be used for computing physical diagrams.

3.12.1 *Findings and main conclusion*

We have shown that the Forcer program is able to compute the four-loop beta function in only three minutes on a 32-core machine. We have given benchmarks of configurations that would take months to compute with Laporta-like methods. Additionally, the Forcer program has already been used for some large calculations at four- and five-loop accuracy [2, 4, 5, 7, 9, 115, 116]. These computations will be discussed in chapter 4 and chapter 6. From our benchmarks and the use in large calculations, we may conclude that Forcer is an efficient program for computing four-loop massless propagator integrals.

3.12.2 *Future research*

An interesting area for future research is to extend Forcer to five loops. Most parts of the program are easily extended, since the predominantly automatic construction of the program is not limited to four loops. We have even computed 30% of the diagrams of the five-loop gluon propagator using the five-loop Forcer. The challenge is that over 200 topologies have an unknown reduction scheme. If the heuristics for deriving reduction schemes can be extended and fully automated, a full five-loop Forcer can be constructed shortly after. The idea is quite challenging: the number of parameters that have to be reduced grows from 14 at four loops to 20 at five loops.

4 RESULTS FOR FOUR-LOOP COMPUTATIONS

In this chapter we use the Forcer program, constructed in chapter 3 as part of the answer to **RQ2**, to compute two classes of four-loop objects. Class 1 concerns propagators and vertices of QCD, and Class 2 concerns Mellin moments of splitting functions and coefficient functions.

CLASS 1 The first class of objects that we are going to compute are the finite pieces of propagators and vertices to any power in $\epsilon = (D-4)/2$, where D is the space-time dimension. Up until this point, only the poles and the ε^0 coefficients are known, since they have been used in computations of the basic renormalisation group functions of QCD [23, 24, 113, 114, 119–126] which have recently reached five-loop accuracy [4, 40, 41, 127–129]. In the modified minimal subtraction ($\overline{\text{MS}}$) scheme [130, 131], defined by subtracting the poles in ε together with a fixed term that occurs in dimensional regularisation , these functions are obtained by computing single poles in corresponding Green's functions. Since only the poles are required, the above four- and five-loop results were first obtained using the method of infrared rearrangement [4, 124, 132–138] which simplifies computations without changing the ultraviolet singular structure, but modifies the finite parts. Therefore, it is not possible to compute the complete finite piece using these standard methods.

Our aim is to provide the self-energies and a set of vertices with one vanishing external momentum for massless QCD at four-loop accuracy. The unrenormalised results are exact in terms of ε, and four-loop master integrals [105, 106]. The computation has been performed for a general gauge group and in an arbitrary covariant linear gauge, by using the Forcer program [1, 7, 9] for massless four-loop propagator-type integrals. For the vertices, setting one of the momenta to zero effectively reduces vertex integrals to propagator-type integrals. In QCD this does not create Infrared (IR) divergences, which means the poles do not change. At the three-loop level, similar computations were performed in ref. [139], but with an expansion in ε. In addition, studies of QCD vertices in perturbation theory for various configurations include refs. [140–152].

We compute all QCD vertices in a general linear covariant gauge, with the exception of the four-gluon vertex for which there are at least three difficulties: first, two momenta have to be nullified before the diagrams become propagator-like. Second, the number of diagrams is large at four loops. Third, the colour structure for a generic group is no longer an overall factor, but will be term dependent.

A direct application of our results is to compute conversion factors for renormalisation group functions from the $\overline{\text{MS}}$ scheme to momentum subtraction schemes, see, e.g., refs. [140, 143]. In a later chapter (6.5), we will use the results presented here to convert the five loop beta function to the MiniMOM scheme [153]. The MiniMOM scheme is a momentum subtraction scheme that is more convenient than $\overline{\text{MS}}$ for comparing QCD in the perturbative and non-perturbative regime.

CLASS 2 The second class of objects we are going to compute are Mellin moments of four-loop splitting functions and coefficient functions. Even though for most cases three-loop accuracy is adequate, there are at least two cases where the next order is of interest due to (1) very high requirements on the theoretical accuracy, such as in the determination of the strong coupling constant α_s from deep-inelastic scattering (DIS), see, e.g., [154], or (2) a slow convergence of the perturbation series, such as for Higgs production in proton-proton collisions, see, e.g., [82, 155].

At present, a direct computation of the four-loop splitting functions appears to be too difficult. Work on low-integer Mellin moments of these functions started ten years ago [156]; until recently only the $N = 2$ and $N = 4$ moments had been obtained of the quark+antiquark non-singlet splitting function $P_{\text{ns}}^{(3)+}$ together with the $N = 3$ result for its quark–antiquark counterpart $P_{\text{ns}}^{(3)-}$ [157–159].

The goal in this chapter is to employ the FORCER program to extend the MINCER-based fixed Mellin-N calculations of refs. [84–86] to four-loop accuracy. We will use the optical theorem method [83–86] to compute low-N splitting functions and coefficient functions. Next, we will use the operator product expansion method [158, 160] to compute higher moments of splitting functions.

4

We now provide the layout of this chapter. In section 4.1, we define the group notations. In section 4.2, we compute Yang-Mills propagators and vertices with a vanishing momentum (objects of Class 1). Next, we compute Mellin moments of splitting functions and coefficient functions (objects of Class 2) in section 4.3. Finally, we provide the chapter conclusion in section 4.4.

4.1 GROUP NOTATIONS

In this section we will introduce our notations for the group invariants appearing in the remainder of this thesis. T^a are the generators of the representation of the fermions, and f^{abc} are the structure constants of the Lie algebra of a compact simple Lie group,

$$T^a T^b - T^b T^a \;=\; i f^{abc}\, T^c \,. \tag{60}$$

The quadratic Casimir operators C_F and C_A of the N-dimensional fermion and the N_A-dimensional adjoint representation are given by $[T^a T^a]_{ik} = C_F \delta_{ik}$ and $f^{\,acd} f^{\,bcd} = C_A \delta^{\,ab}$, respectively. The trace normalisation of the fermion representation is $\text{Tr}(T^a T^b) = T_F \delta^{\,ab}$. At $L \geq 3$ loops also quartic group invariants enter the results. These can be expressed in terms of contractions of the totally symmetric tensors

$$\begin{aligned} d_F^{abcd} &= \frac{1}{6}\,\text{T}r(T^a\, T^b\, T^c\, T^d \,+\, \text{ five } bcd \text{ permutations })\;, \\ d_A^{abcd} &= \frac{1}{6}\,\text{T}r(C^a C^b C^c C^d \,+\, \text{ five } bcd \text{ permutations })\;. \end{aligned} \tag{61}$$

Here the matrices $[\,C^a\,]_{bc} = -if^{abc}$ are the generators of the adjoint representation. It should be noted that in QCD-like theories without particles that are colour neutral, Furry's theorem [161] prevents the occurrence of symmetric tensors with an odd number of indices.

For fermions transforming according to the fundamental representation and the standard normalisation of the SU(N) generators, these 'colour factors' have the values

$$\begin{aligned} T_F &= \frac{1}{2}, \; C_A = N, \; C_F = \frac{N_A}{2N} = \frac{N^2-1}{2N}, \; \frac{d_A^{abcd}d_A^{abcd}}{N_A} = \frac{N^2(N^2+36)}{24}, \\ \frac{d_F^{abcd}d_A^{abcd}}{N_A} &= \frac{N(N^2+6)}{48}, \quad \frac{d_F^{abcd}d_F^{abcd}}{N_A} = \frac{N^4-6N^2+18}{96\,N^2}. \end{aligned} \tag{62}$$

The results for QED (i.e., the group U(1)) are obtained for $C_A = 0$, $d_A^{abcd} = 0$, $C_F = 1$, $T_F = 1$, $d_F^{abcd} = 1$, and $N_A = 1$. For a discussion of other gauge groups the reader is referred to ref. [113].

4.2 PROPAGATORS AND VERTICES

In this section we will present the computation of four-loop QCD propagators and vertices with a vanishing momentum.

First, we summarise the notations for self-energies and vertex functions with one vanishing momentum presented in section 4.2.1 to section 4.2.4. In most cases we follow the conventions in ref. [139].[1] Next, we describe our renormalisation method in section 4.2.5 and how to compute anomalous dimensions in section 4.2.6. Finally, we present the results of our computation in section 4.2.7.

4.2.1 *Self energies*

The gluon, ghost and quark self-energies (figure 25) are of the form

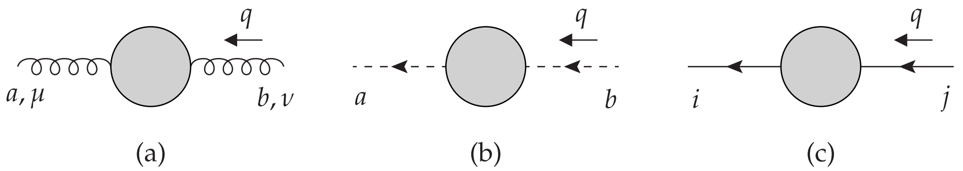
$$\Pi^{ab}_{\mu\nu}(q) = -\delta^{ab}(q^2 g_{\mu\nu} - q_\mu q_\nu)\Pi(q^2), \tag{63}$$

$$\tilde{\Pi}^{ab}(q) = \delta^{ab} q^2 \tilde{\Pi}(q^2), \tag{64}$$

$$\Sigma^{ij}(q) = \delta^{ij} \not{q}\, \Sigma_V(q^2). \tag{65}$$

The colour indices are understood such that a and b are for the adjoint representation of the gauge group, i and j for the representation to which the quarks transform. In eq. (63) we have used the fact that the Ward identities render the gluon propagator transversal. The 'form factors' $\Pi(q^2)$, $\tilde{\Pi}(q^2)$ and $\Sigma_V(q^2)$ can easily be extracted from contributions of the corresponding one-particle irreducible diagrams by applying

[1] We note that these conventions may be different from the ones commonly used in the literature. In fact, the Feynman rules in FORCER are different as well, and hence we occasionally had to convert intermediate results from one convention to the other and back.



(a) (b) (c)

Figure 25: The gluon, ghost and quark self-energies $\Pi^{ab}_{\mu\nu}(q)$ (a), $\tilde{\Pi}^{ab}(q)$ (b) and $\Sigma^{ij}(q)$ (c).

projection operators [139] (the same holds for the vertex functions discussed below). They are related to the full gluon, ghost and quark propagators as follows:

$$D^{ab}_{\mu\nu}(q) = \frac{\delta^{ab}}{-q^2}\left[\left(-g_{\mu\nu} + \frac{q_\mu q_\nu}{q^2}\right)\frac{1}{1+\Pi(q^2)} - \xi\,\frac{q_\mu q_\nu}{q^2}\right], \tag{66}$$

$$\Delta^{ab}(q) = \frac{\delta^{ab}}{-q^2}\,\frac{1}{1+\tilde{\Pi}(q^2)}, \tag{67}$$

$$S^{ij}(q) = \frac{\delta^{ij}}{-q^2}\,\frac{\not{q}}{1+\Sigma_V(q^2)}. \tag{68}$$

Here the Landau gauge corresponds to $\xi = 0$, and the Feynman gauge to $\xi = 1$. We note that this convention differs from that in the widely used Form version [33] of the Mincer program [32] for three-loop self-energies, where the symbol `xi` represents $1 - \xi$.

4.2.2 *Triple-gluon vertex*

Without loss of generality, one can set the momentum of the third gluon to zero, as depicted in figure 26. Then the triple-gluon vertex can be written in the following form:

$$\Gamma^{abc}_{\mu\nu\rho}(q,-q,0) = -igf^{abc}\left[(2g_{\mu\nu}q_\rho - g_{\mu\rho}q_\nu - g_{\rho\nu}q_\mu)\,T_1(q^2) - \left(g_{\mu\nu} - \frac{q_\mu q_\nu}{q^2}\right)q_\rho T_2(q^2)\right], \tag{69}$$

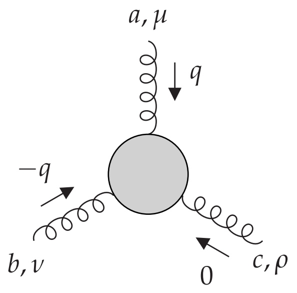


Figure 26: The triple-gluon vertex with one vanishing momentum, $\Gamma^{abc}_{\mu\nu\rho}(q,-q,0)$.

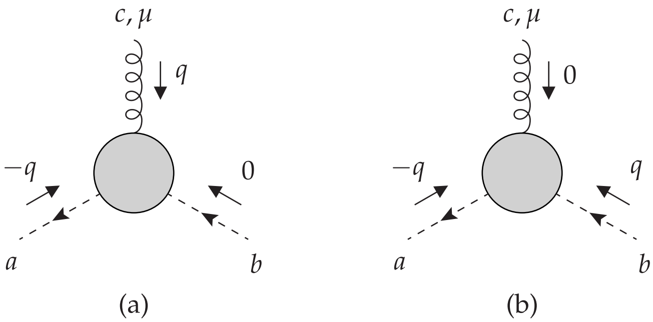


Figure 27: The ghost-gluon vertex: (a) $\tilde{\Gamma}^{abc}_{\mu}(-q, 0; q)$ has a vanishing incoming ghost momentum and (b) $\tilde{\Gamma}^{abc}_{\mu}(-q, q; 0)$ a vanishing gluon momentum.

where g is the coupling constant and f^{abc} are the structure constants of the gauge group in eq. 60. The first term in the square bracket corresponds to the tree-level vertex while the second term arises from radiative corrections, i.e., at the tree-level the form factors $T_{1,2}(q^2)$ read

$$T_1(q^2)\big|_{\text{tree}} = 1, \qquad T_2(q^2)\big|_{\text{tree}} = 0. \tag{70}$$

Because of Furry's theorem [161] and the fact that we have no colour-neutral particles, symmetric invariants with an odd number of indices cannot occur for internal fermion lines. Neither can such invariants occur for the adjoint representation. Hence, if we project out a d^{abc} structure, we would get a scalar invariant with an odd number of f tensors, and such a combination must be zero. This has been checked explicitly to the equivalent of six-loop vertices in ref [112]. Due to the bosonic property of gluons, the totally antisymmetric colour factor f^{abc} leads to antisymmetric Lorentz structure as in eq. (69). One could consider another Lorentz structure,

$$-igf^{abc} q_\mu q_\nu q_\rho T_3(q^2). \tag{71}$$

However, a Slavnov-Taylor identity requires $T_3(q^2)$ to vanish [139].

4.2.3 *Ghost-gluon vertex*

Since the tree-level vertex is proportional to the outgoing ghost momentum in our convention, nullifying this momentum gives identically zero in perturbation theory. Therefore, we only have two possibilities to set one of the external momenta to zero. One is the incoming ghost momentum and the other is the gluon momentum (figure 27):

$$\tilde{\Gamma}^{abc}_{\mu}(-q, 0; q) = -igf^{abc} q_\mu \tilde{\Gamma}_h(q^2), \tag{72}$$

$$\tilde{\Gamma}^{abc}_{\mu}(-q, q; 0) = -igf^{abc} q_\mu \tilde{\Gamma}_g(q^2). \tag{73}$$

The subscript h of $\tilde{\Gamma}_h(q^2)$ indicates the function with vanishing incoming ghost momentum, whereas g of $\tilde{\Gamma}_g(q^2)$ denotes the vanishing gluon momentum. These functions are equal to one at the tree-level,

$$\tilde{\Gamma}_h(q^2)\big|_{\text{tree}} = \tilde{\Gamma}_g(q^2)\big|_{\text{tree}} = 1. \tag{74}$$

4.2.4 *Quark-gluon vertex*

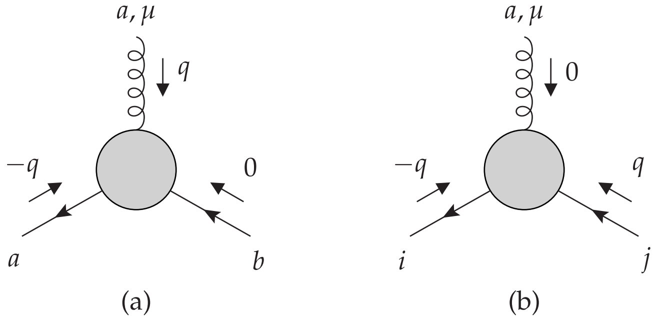


Figure 28: The quark-gluon vertex: (a) $\Lambda^a_{\mu,ij}(-q,0;q)$ has a vanishing incoming quark momentum and (b) $\Lambda^a_{\mu,ij}(-q,q;0)$ a vanishing gluon momentum.

We consider the case of a vanishing incoming quark momentum and the case of a vanishing gluon momentum (figure 28). Nullifying the outgoing quark momentum gives the same result as nullifying the incoming quark momentum. Then the vertex can be written as

$$\Lambda^a_{\mu,ij}(-q,0;q) = gT^a_{ij}\Big[\gamma_\mu \Lambda_q(q^2) + \gamma^\nu\Big(g_{\mu\nu} - \frac{q_\mu q_\nu}{q^2}\Big)\Lambda^T_q(q^2)\Big], \tag{75}$$

$$\Lambda^a_{\mu,ij}(-q,q;0) = gT^a_{ij}\Big[\gamma_\mu \Lambda_g(q^2) + \gamma^\nu\Big(g_{\mu\nu} - \frac{q_\mu q_\nu}{q^2}\Big)\Lambda^T_g(q^2)\Big]. \tag{76}$$

T^a_{ij} are the generators of the representation for the quarks. The subscript q indicates the functions with vanishing incoming quark momentum and g indicates those with vanishing gluon momentum. At the tree-level we have

$$\Lambda_q(q^2)\big|_{\text{tree}} = \Lambda_g(q^2)\big|_{\text{tree}} = 1, \tag{77}$$

$$\Lambda^T_q(q^2)\big|_{\text{tree}} = \Lambda^T_g(q^2)\big|_{\text{tree}} = 0. \tag{78}$$

4.2.5 *Renormalisation*

All the quantities we compute contain divergences. The theory of renormalisation states that for QCD these can be absorbed into redefinitions of the interaction strength, mass parameter, and field definitions. This implies that the observables such as the mass of a particle, and the strength of the strong coupling constant depend on the energy scale at which one measures. In this section we will define renormalisation constants which make the observable quantities finite.

In a generic renormalisation scheme R, the respective renormalisations of the gluon, ghost and quark fields can be written as

$$(A^B)^a_\mu = \sqrt{Z_3^R}\,(A^R)^a_\mu, \tag{79}$$

$$(\eta^B)^a = \sqrt{\tilde{Z}_3^R}\,(\eta^R)^a, \tag{80}$$

$$\psi^B_{if} = \sqrt{Z_2^R}\,\psi^R_{if}. \tag{81}$$

The superscript "B" indicates a bare (divergent) quantity and "R" a renormalised (finite) one. For the coupling constant, we define $a = \alpha_s/(4\pi) = g^2/(16\pi^2)$. Then a and the gauge parameter ξ are renormalised in dimensional regularisation ($D = 4 - 2\varepsilon$) as follows:

$$a^B = \mu^{2\varepsilon} Z_a^R a^R, \tag{82}$$

$$\xi^B = Z_3^R \xi^R. \tag{83}$$

Here μ is the 't Hooft mass scale, which is added to make the coupling constant dimensionless. We have used the fact that the gauge parameter is also renormalised by the gluon field renormalisation constant, $Z_\xi^R = Z_3^R$. The renormalisation of the self-energies and vertex functions is performed as

$$1 + \Pi^R = Z_3^R(1 + \Pi^B), \tag{84}$$

$$1 + \tilde{\Pi}^R = \tilde{Z}_3^R(1 + \tilde{\Pi}^B), \tag{85}$$

$$1 + \Sigma_V^R = Z_2^R(1 + \Sigma_V^B), \tag{86}$$

and

$$T_i^R = Z_1^R T_i^B, \qquad i = 1, 2, \tag{87}$$

$$\tilde{\Gamma}_i^R = \tilde{Z}_1^R \tilde{\Gamma}_i^B, \qquad i = h, g, \tag{88}$$

$$\Lambda_i^R = \bar{Z}_1^R \Lambda_i^B, \qquad \Lambda_i^{T,R} = \bar{Z}_1^R \Lambda_i^{T,B}, \qquad i = q, g, \tag{89}$$

where the vertex renormalisation constants are related to the field and coupling renormalisation constants via the Slavnov-Taylor identities by

$$\sqrt{Z_a^R Z_3^R} = \frac{Z_1^R}{Z_3^R} = \frac{\tilde{Z}_1^R}{\tilde{Z}_3^R} = \frac{\bar{Z}_1^R}{Z_2^R}. \tag{90}$$

In MS-like schemes, the renormalisation constants contain only pole terms with respect to ε and thus take the form

$$Z_i^{\mathrm{MS}} = 1 + \sum_{l=1}^{\infty} a^l Z_i^{\mathrm{MS},(l)} = 1 + \sum_{l=1}^{\infty} a^l \sum_{n=1}^{l} \frac{Z_i^{\mathrm{MS},(l,n)}}{\varepsilon^n}. \tag{91}$$

The coefficients $Z_i^{\text{MS},(l,n)}$ are determined order by order in such a way that any renormalised Green's function becomes finite. Below we give an example for the three-loop background field propagator G. In the background field the field strength renormalisation is simply $Z_a^{-1/2}$ (see section 6.2). We assume we are in the Landau gauge ($\xi = 0$), so that we need not worry about renormalising ξ:

$$
\begin{aligned}
G_B &= 1 + a_B G_1 + a_B^2 G_2 + a_B^3 G_3 + \dots\,, \\
G_R &= Z_a^{-1}(1 + a Z_a G_1 + a^2 Z_a^2 G_2 + a^3 Z_a^3 G_3 + \dots) \\
&= 1 + a(G_1 - Z^{\text{MS},(1)}) + a^2(G_2 + {Z^{\text{MS},(1)}}^2 - Z^{\text{MS},(2)}) \\
&\quad + a^3(G_3 + G_2 Z^{\text{MS},(1)} - {Z^{\text{MS},(1)}}^3 + 2 Z^{\text{MS},(1)} Z^{\text{MS},(2)} - Z^{\text{MS},(3)}) + \dots\,.
\end{aligned}
\tag{92}
$$

If we introduce the pole operator $K(\sum_{i=-\infty}^{\infty} \frac{x_i}{\varepsilon^i}) = \sum_{i=-\infty}^{-1} \frac{x_i}{\varepsilon^i})$, which takes the pole part of a Laurent series in ε, we can write:

$$
\begin{aligned}
Z^{\text{MS},(1)} &= K(G_1)\,, \\
Z^{\text{MS},(2)} &= K(G_2) + K(G_1)^2\,, \\
Z^{\text{MS},(3)} &= K(G_3) + K\left(G_2 K(G_1)\right) - K(G_1)^3 + 2K(G_1)\left(K(G_2) + K(G_1)^2\right)\,.
\end{aligned}
\tag{93}
$$

In the case of computing other self-energies, the field strength renormalisation will not be $Z_a^{-1/2}$, but will have its own Z. This means a system of equations has to be solved. For perturbatively renormalisable theories, such a system can always be solved order by order.

4.2.6 *Anomalous dimensions*

Renormalisation introduces an arbitrary scale μ on which the bare quantities do not depend. Let us take for example the gluon propagator with zero quark masses $R(Q, a, \xi, \mu) = Z_3 R_B(Q, Z_a a_B, Z_3 \xi_B)$, where we suppress the MS label and we have used the fact that gauge invariance ensures that the gauge parameter is renormalised with Z_3. If we enforce that $\frac{dR_B}{d\mu} = 0$ we obtain the following Callan-Symanzik renormalisation group equation [162, 163]:

$$
\begin{aligned}
\mu^2 \frac{d}{d\mu^2} R(Q,a) &= \left[\mu^2 \frac{\partial}{\partial \mu^2} + \mu^2 \frac{da}{d\mu^2}\frac{\partial}{\partial a} + \mu^2 \xi \frac{d \ln \xi}{d\mu^2}\frac{\partial}{\partial \xi} - \mu^2 \frac{d \ln Z_3}{d\mu^2}\right] R(Q,a) \\
&\equiv \left[\mu^2 \frac{\partial}{\partial \mu^2} + \tilde{\beta}\frac{\partial}{\partial a} + \xi \gamma_3 \frac{\partial}{d\xi} - \gamma_3\right] R(Q,a) = 0\,,
\end{aligned}
\tag{94}
$$

where the quantity γ_3 is the anomalous dimension of the external gluon field and $\tilde{\beta}$ is the D-dimensional beta function. The anomalous dimension γ_3 describes how the field evolves with the energy scale, and $\tilde{\beta}$ describes how the strong coupling constant evolves with the energy scale.

We rewrite:

$$\begin{aligned}\tilde{\beta} &\equiv \mu^2 \frac{da}{d\mu^2} = \mu^2 \frac{dZ_a^{-1} a_B}{d\mu^2} \\ &= \mu^2 Z_a^{-1} \frac{da_B}{d\mu^2} + \mu^2 a_B \frac{dZ_a^{-1}}{d\mu^2} \\ &= -\varepsilon a + \mu^2 a Z_a^{-1} \frac{dZ_a}{da} \frac{da}{d\mu^2} \\ &= -\varepsilon a + a \frac{d \ln Z}{da} \tilde{\beta} \\ &\equiv -\varepsilon a + \beta\,,\end{aligned} \tag{95}$$

where β is the beta function in four dimensions and we used that the beta function is gauge independent in MS-like schemes.

For both the beta function and anomalous dimension, the following reasoning holds (where γ is the anomalous dimension associated with Z):

$$\begin{aligned}Z\gamma = Z\mu^2 \frac{d \ln Z}{d\mu^2} &= \mu^2 \frac{\partial Z}{\partial a} \frac{da}{d\mu^2} + \mu^2 \frac{\partial Z}{\partial \xi} \frac{d\xi}{d\mu^2} \\ &= \frac{\partial Z}{\partial a}(-\varepsilon a + \beta) + \frac{\partial Z}{\partial \xi} \gamma_3,\end{aligned} \tag{96}$$

This equation holds for any power of ε, so we compare the ε^0 part (for which β and γ_3 drop out):

$$\gamma = -\sum_{n=1}^{\infty} n a^n Z^{\mathrm{MS},(n,1)}, \qquad \beta = -\sum_{n=1}^{\infty} n a^n Z_a^{\mathrm{MS},(n,1)}\,. \tag{97}$$

Even though the anomalous dimensions are thus only comprised of the simple poles of Z in MS-like renormalisation schemes, there is no loss of information: the entire Z can be reconstructed from (96). This also means that the higher-order poles of Z are completely determined by lower-order renormalisation group contributions. Consequently, the higher-order poles serve as a check for higher-order results.

4.2.7 *Computations and checks*

The results in this section are obtained by direct computation using the Forcer package, as described in chapter 3. The topologies are mapped to a built-in Forcer topology, after nullifying a leg for the vertices. To extract the form factors defined above, a generalisation of the projection operators in ref. [139] to a generic gauge group is used. Then the Feynman rules are applied. The remaining Lorentz-scalar integrals (which include loop-momenta numerators) are computed by the Forcer program.

The computation time varied between an hour and a week, on a single computer. The easy cases, such as the ghost propagator and quark propagator took an hour.

The gluon propagators and ghost-gluon-gluon vertex and quark-gluon-gluon vertex took about eight hours per configuration. The triple gluon vertex was the hardest case and took a week per configuration on a single machine with 24 cores. Had we chosen to compute with an expansion in ε, the computations would have been much faster.

We have checked our setup and results in various ways.

- The longitudinal component of the gluon self-energy $\delta^{ab} q_\mu q_\nu \Pi_L(q^2)$, see eq. (63), was shown to be zero by an explicit calculation at the four-loop level.
- The form factor $T_3(q^2)$ of the triple-gluon vertex in eq. (71) was computed and indeed vanished at the four-loop level.
- All the self-energies and vertex functions computed in this work were compared up to three loops with those in ref. [139]. Note that the finite parts of the vertex-function results in ref. [139] are only correct for SU(N) gauge groups, since the presence of quartic Casimir operators was not taken into account in the reconstruction of the general case. This fact was also noted in ref. [129].
- The four-loop renormalisation constants and anomalous dimensions for the case of SU(N) and a general linear covariant gauge were provided in ancillary files of ref. [126]. Directly after Forcer was completed, we established agreement with those results. For a generic group our results are in agreement with ancillary files of ref. [129].
- We remark that the ghost-gluon vertex is unrenormalised $\tilde{Z}_1^{\overline{\mathrm{MS}}} = \tilde{Z}_3^{\overline{\mathrm{MS}}} \sqrt{Z_a^{\overline{\mathrm{MS}}} Z_3^{\overline{\mathrm{MS}}}}$ $= 1$ in the Landau gauge. Moreover, our results confirm that the vertex has no radiative corrections when the incoming ghost momentum is nullified (i.e., $\tilde{\Gamma}_h^{\overline{\mathrm{MS}}} = 1$) in the Landau gauge up to four loops.

Since the results are rather lengthy, we will not include them in this thesis. All result can be obtained in a digital form as ancillary files to the article [2]. The files contain the bare results for the self-energies and vertices in terms of master integrals with coefficients that are exact in for any dimension D, as well as the results in the $\overline{\mathrm{MS}}$ scheme for $D = 4$.

4.3 splitting functions and coefficient functions

In order to describe collisions involving protons in colliders, one effectively describes the interactions between a particle from a proton, called a parton, and a probe. The probe could be a parton or any of the force carriers. The interaction depends on the relative momenta of the particles, their energy, and their type (up quark, down quark, gluon, etc). Consequently, an accurate model of the proton structure is required. A critical ingredient is the *parton density function* (pdf), which captures the probability that a certain particle with a certain (collinear) momentum fraction is inside the proton. A pdf has to be experimentally determined, since it involves low-energy

QCD which is outside the regime of perturbation theory. It also depends on the renormalisation scale, and this dependence can be determined by making precise computations in perturbative QCD.

Since QFT predicts that particles can briefly split up into others (for example a quark splitting into a quark with smaller momentum and a gluon), quantum corrections will influence the pdfs. How sensitive the system is to these virtual particles depends on the energy at which we are measuring. Below we show the DGLAP equation [164–166], which describes the dependence of a pdf f_i for a parton i (could be all light quarks, light antiquarks and the gluon), depending on an energy scale μ^2 (we set the factorisation scale to the renormalisation scale without loss of generality):

$$\frac{\partial f_i(x,\mu^2)}{\partial \mu^2} = \frac{\alpha_s(\mu^2)}{2\pi} \int_x^1 \frac{dz}{z} \left[\sum_j f_j(z,\mu^2) P_{ij}\left(\frac{x}{z}, \alpha_s(\mu^2)\right) \right] \tag{98}$$

where x is the fraction of the proton momentum, j sums over all possible partons, the integral is over all possible momentum fractions, and P_{ij} are the *splitting functions*: a quantity related to the probability that parton j will split up into i and other particles. What this equation in essence describes is that the probability of finding a parton i with momentum fraction x is changed by the event where a parton with a higher energy splits up into a parton of type i with exactly momentum fraction x.

Equation (98) is a complicated integro-differential equation. To solve it, we first turn the convolution of the splitting function and the pdf in eq. (98) into an ordinary product using a Mellin transform:

$$f_i(N,\mu^2) = \int_0^1 dx\, x^{N-1} f_i(x,\mu^2)\ . \tag{99}$$

The Mellin moments of splitting functions can be decoupled into a $2n_f - 1$ scalar equations and a 2x2 flavour-singlet system, see, e.g., [5, 8]. The non-singlet splitting functions $P_{\rm ns}^{\pm}$ are the two combinations of quark–quark and quark–anti-quark splitting functions relevant to the $2n_f - 2$ flavour differences

$$q_{{\rm ns},ik}^{\pm} = q_i \pm \bar{q}_i - (q_k \pm \bar{q}_k) \tag{100}$$

of quark distributions that evolve as scalars [167].

Since even or odd Mellin moments of splitting functions are anomalous dimensions, they are often expressed as such. We are going to calculate them up to fourth order in the reduced coupling constant $a = a_s/4\pi$:

$$\gamma_{\rm ns}^{\pm}(N) = -P_{\rm ns}^{\pm}(N) = -aP_{\rm ns}^{\pm(0)}(N) - a^2 P_{\rm ns}^{\pm(1)}(N) - a^3 P_{\rm ns}^{\pm(2)}(N) - a^4 P_{\rm ns}^{\pm(3)}(N)\ . \tag{101}$$

We will discuss two different methods to compute Mellin moments of splitting functions. The first is the optical theorem method described in section 4.3.1, and the second is the operator method described in section 4.3.2. Finally, we compute the axial vector current in 4.3.3.

4.3.1 *Optical theorem method*

The higher Mellin moments N can be obtained for the splitting functions $P_{ij}(N)$, the better an approximation can be made for the splitting functions in x-space. We use exactly the same method as described in [83–86], only we use Forcer instead of Mincer. In summary, using the optical theorem and a dispersion relation, the bare structure functions can be expressed as forward amplitudes. After applying a harmonic projection to isolate the desired coefficient, the Mellin moment of the structure function is expressed in terms of a propagator integral. These integrals are then computed with Forcer. The $1/\varepsilon$ poles provide the splitting functions. The finite piece after renormalisation yields the coefficient function.

Using this procedure, we have computed up to $N = 6$ for the non-singlet splitting functions and coefficient functions [8]. In figure 29 we show the results for the first Mellin moments. The dashed and dotted lines are the Padé approximations from the three-loop results of ref. [34]. For the singlet case, we have computed up to $N = 4$.

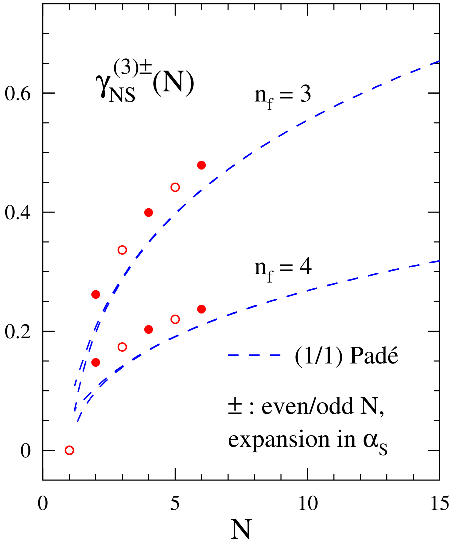


Figure 29: The lowest three even-N and odd-N values, respectively, of the anomalous dimensions $\gamma_{NS}^{(3)+}$ and $\gamma_{NS}^{(3)-}$, compared to Padé estimates derived from the NNLO results of ref. [34].

We have computed significantly more moments for specific colour factors. For diagrams with a high number of (light) fermion loops, indicated by powers of n_f, the complexity of the diagrams is simplified: for P_{qq} there are no four-loop diagrams without insertions that have a fermion loop. In figure 30 we show *all* three-loop graphs with two fermion loops, where a single fermion insertion on one of the gluon lines is understood. These graphs have been evaluated to the 40th Mellin moment, leading to more than complexity 80 integrals (as defined in section 3.5).

This provided enough information to reconstruct the analytic form in N using the LLL-algorithm [88, 168]. With this method, we have computed the n_f^2 contribution for the four-loop non-singlet splitting function and the n_f^3 contribution to the four-loop singlet splitting function [5].

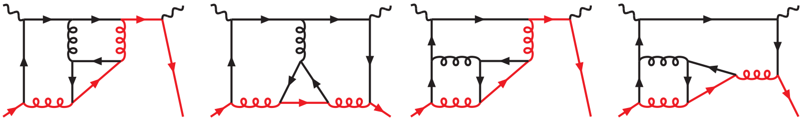


Figure 30: All three-loop diagrams with n_f^2 for P_{qq}, where a single fermion insertion on one of the gluon propagators is understood. There are no four-loop diagrams of this kind.

4.3.2 *Operator method*

Another way to compute moments of splitting functions is the light-cone operator product expansion (OPE) method [158, 160]. Although there are some challenges in constructing the operators, the advantage is that the complexity of the integrals scales like N, whereas for the optical theorem method it scales like $2N$. The coefficient functions are not addressed in this approach.

We have computed the non-singlet splitting function up to $N = 16$ moments [115]. In the large-n_c limit, up to $N = 19$ has been computed. Eighteen moment were used for a full reconstruction of the N dependence, and the nineteenth was used to confirm the result. Since in the large-n_c limit, $\gamma_{\text{NS}}^{-} = \gamma_{\text{NS}}^{+}$, both even and odd moments could be used in the reconstruction.

In figure 31 we show how the large-n_c limit matches up with the first 16 moments of the non-singlet splitting function, when studying the n_f^0 and n_f^1 coefficient. We expect an error of about 10% percent ($\frac{n_c}{n_c^3}$) for QCD that will decrease at large N. We see that even for low N, the error is less than 10% and that the (relative) error decreases fast with high N.

In figure 32 we show how moments of the non-singlet splitting function compare to the large-n_c result, for physical values of n_f. For $n_f = 3, n_f = 4$, and $n_f = 6$, we see that the error is small, even for low values of N. For the physically important $n_f = 5$, sizeable cancellations between non-n_f and n_f pieces result in a loss of accuracy for the large-n_c limit. To improve the large-n_c approximation, some (approximations of) non-leading large-n_c contributions have to be computed as well.

Since we have reconstructed the non-singlet splitting function for all N, we can study the large-N behaviour. It has the following form in $\overline{\text{MS}}$ [169–171]:

$$\gamma = A \ln N + B + C\frac{1}{N}\ln N + D\frac{1}{N} + \dots \tag{102}$$

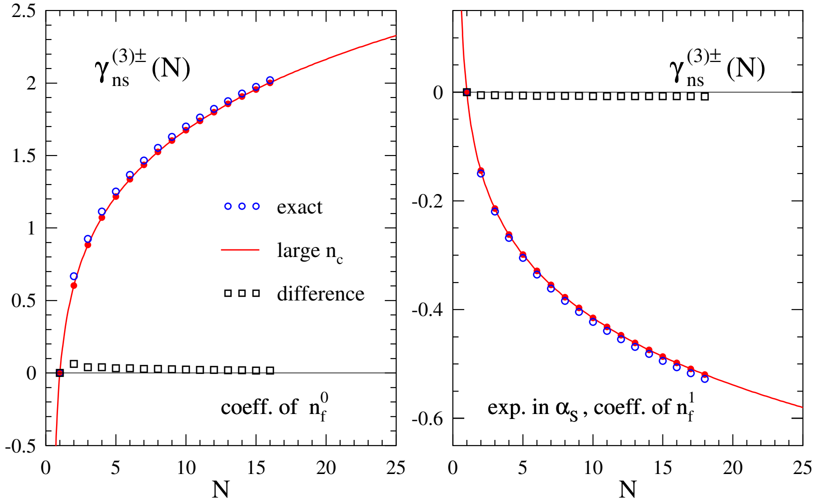


Figure 31: The large n_c limit versus the first Mellin moments of the non-singlet splitting function. The error is small, even for low N.

where A is a quantity known as the cusp anomalous dimension. This quantity is relevant beyond pdf evolution and is for example used in soft-gluon exponentiation and soft-collinear effective theories (SCET).

Taking the limit of $N \to \infty$ on our result yields a new $n_c^3 C_F$ term for the four-loop cusp anomalous dimension:

$$\gamma_{\rm cusp}^{(3)} = + n_c^3 C_F \biggl(+ \frac{84278}{81} - \frac{88832}{81}\zeta_2 + \frac{20992}{27}\zeta_3 + 1804\zeta_4 - \frac{352}{3}\zeta_3\zeta_2 \\ - 352\zeta_5 - 32\zeta_3^2 - 876\zeta_6 \biggr) + \dots .$$

The $n_c^3 C_F$ term was soon afterwards confirmed via a calculation of the γqq form factor in the large-n_c limit [172].

4.3.3 *Axial vector current*

In order to compute the vector-axial interference structure function F_3, it is necessary to know the anomalous dimension of the axial vector current $\gamma^\mu\gamma^5$ to correct the treatment of γ_5. This can be retrieved from the ratio of the vector current (quark-quark-photon vertex) and axial vector current (quark-quark-Z vertex). The quark-quark-Z current contains a γ^5, which in its standard description as $i\gamma_0\gamma_1\gamma_2\gamma_3$ is strictly 4-dimensional instead of D-dimensional. To work around this issue we use

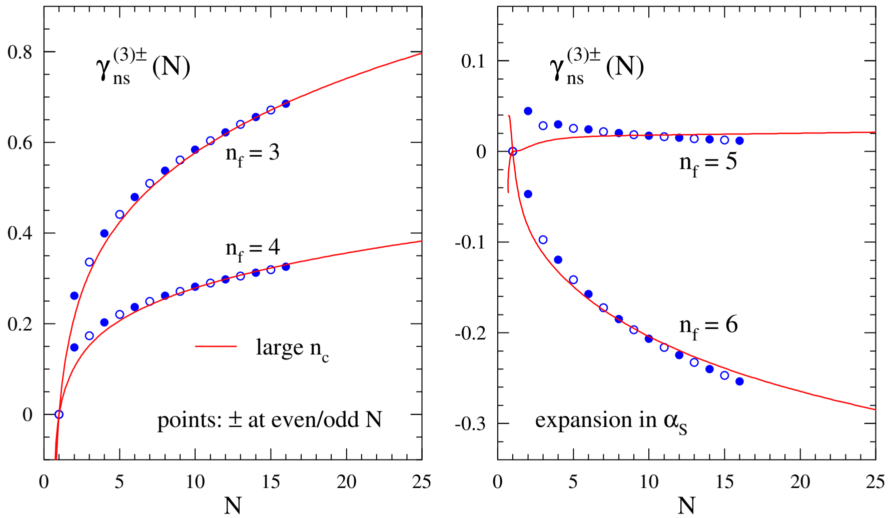


Figure 32: Comparison of the large n_c limit to Mellin moments of the non-singlet splitting functions for physical values of n_f.

the so-called Larin prescription of [173, 174]: $\gamma^\mu\gamma^5 = i\epsilon_{\mu\nu\rho\sigma}\gamma^\nu\gamma^\rho\gamma^\sigma$. As a projector, we use:

$$P = \frac{\epsilon_{\mu\nu\rho\sigma}\gamma^\nu\gamma^\rho\gamma^\sigma}{4(D-1)(D-2)(D-3)} \,, \tag{103}$$

which makes the tree-level contribution 1.

However, the Larin prescription breaks the axial Ward identity [173, 174]. As a result, an additional renormalisation for the axial vector current relative to the vector current is required. We define two quantities: Z_A and Z_5. Z_A is the additional renormalisation required for the axial current, and Z_5 is an extra finite renormalisation. Z_5 is used to map the renormalised axial current onto the renormalised vector current to 'fix' the anomaly. Given the renormalisation of the vector current V and the axial current A:

$$V_R = Z_{VC} V_B \,, \qquad A_R = Z_{AC} A_B \,, \tag{104}$$

where Z_{VC} and Z_{AC} contain the quark field renormalisation Z_2, the renormalisation of the strong coupling Z_a, and the renormalisation of the gauge parameter Z_3.

From these quantities, we can compute Z_A and Z_5:

$$Z_A = \frac{Z_{AC}}{Z_{VC}} \,, \qquad Z_5 = \lim_{\epsilon\to 0} \frac{V_R}{A_R} \,. \tag{105}$$

Both Z_A and Z_5 are gauge invariant. We have checked this by running with all powers of the gauge parameter.

Calculating the traces is the most expensive part of the computation. Using the optimisation from [175], we could have made the calculation much faster. The total computation takes 107 hours to compute on a single 2.4 GHz core. Below we present the results.

$$
\begin{aligned}
Z_A^{(2)} &= \frac{1}{\epsilon}\left(\frac{22}{3}C_A C_F - \frac{8}{3}C_F T_F n_f\right), \\
Z_A^{(3)} &= \frac{1}{\epsilon^2}\left(\frac{3578}{81}{C_A}^2 C_F - \frac{308}{9}C_A {C_F}^2 - \frac{1664}{81}C_A C_F T_F n_f + \frac{64}{9}{C_F}^2 T_F n_f + \frac{32}{81}C_F {T_F}^2 {n_f}^2\right) \\
&\quad + \frac{1}{\epsilon}\left(-\frac{484}{27}{C_A}^2 C_F + \frac{352}{27}C_A C_F T_F n_f - \frac{64}{27}C_F {T_F}^2 {n_f}^2\right), \\
Z_A^{(4)} &= \frac{1}{\epsilon^3}\Bigg[{C_A}^3 C_F\left(\frac{36607}{108} - 154\zeta_3\right) + {C_A}^2 {C_F}^2\left(-\frac{29309}{54} + 440\zeta_3\right) + C_A {C_F}^3\left(\frac{935}{6}\right. \\
&\quad \left. - 264\zeta_3\right) + {C_A}^2 C_F T_F n_f\left(-\frac{16058}{81} - \frac{8}{3}\zeta_3\right) + C_A {C_F}^2 T_F n_f\left(\frac{4762}{27} - \frac{304}{3}\zeta_3\right) \\
&\quad + {C_F}^3 T_F n_f\left(-\frac{80}{3} + 96\zeta_3\right) + C_A C_F {T_F}^2 {n_f}^2\left(\frac{734}{81} + \frac{64}{3}\zeta_3\right) + {C_F}^2 {T_F}^2 {n_f}^2\left(\frac{80}{27}\right. \\
&\quad \left. - \frac{64}{3}\zeta_3\right) + \frac{208}{81}C_F {T_F}^3 {n_f}^3\Bigg] + \frac{1}{\epsilon^2}\left(-\frac{26411}{162}{C_A}^3 C_F + \frac{605}{9}{C_A}^2 {C_F}^2\right. \\
&\quad + \frac{1262}{9}{C_A}^2 C_F T_F n_f - \frac{176}{9}C_A {C_F}^2 T_F n_f - \frac{824}{27}C_A C_F {T_F}^2 {n_f}^2 - \frac{16}{9}{C_F}^2 {T_F}^2 {n_f}^2 \\
&\quad \left. + \frac{32}{81}C_F {T_F}^3 {n_f}^3\right) + \frac{1}{\epsilon}\left(\frac{1331}{27}{C_A}^3 C_F - \frac{484}{9}{C_A}^2 C_F T_F n_f + \frac{176}{9}C_A C_F {T_F}^2 {n_f}^2\right. \\
&\quad \left. - \frac{64}{27}C_F {T_F}^3 {n_f}^3\right).
\end{aligned}
\tag{106}
$$

$$
\begin{aligned}
Z_5^{(1)} &= 4C_F, \\
Z_5^{(2)} &= \frac{107}{9}C_A C_F - 22{C_F}^2 - \frac{4}{9}C_F T_F n_f, \\
Z_5^{(3)} &= {C_A}^2 C_F\left(\frac{2147}{27} - 56\zeta_3\right) + C_A {C_F}^2\left(-\frac{5834}{27} + 160\zeta_3\right) + {C_F}^3\left(\frac{370}{3} - 96\zeta_3\right) \\
&\quad + C_A C_F T_F n_f\left(-\frac{712}{81} - \frac{64}{3}\zeta_3\right) + {C_F}^2 T_F n_f\left(\frac{124}{27} + \frac{64}{3}\zeta_3\right) - \frac{208}{81}C_F {T_F}^2 {n_f}^2, \\
Z_5^{(4)} &= {C_A}^3 C_F\left(\frac{324575}{648} - \frac{10498}{27}\zeta_3 + 231\zeta_4 + \frac{4120}{3}\zeta_5\right) + {C_A}^2 {C_F}^2\left(-\frac{10619}{6} + 1570\zeta_3\right. \\
&\quad \left. - 660\zeta_4 - \frac{17020}{3}\zeta_5\right) + C_A {C_F}^3\left(\frac{232949}{108} - \frac{3332}{3}\zeta_3 + 396\zeta_4 + 6760\zeta_5\right) \\
&\quad + {C_F}^4\left(-\frac{1553}{2} - 564\zeta_3 - 1840\zeta_5\right) + \frac{d_F^{abcd} d_A^{abcd}}{N_R}\left(\frac{16}{3} - 760\zeta_3 - 960\zeta_5\right)
\end{aligned}
$$

$$
\begin{aligned}
&+ {C_A}^2 C_F T_F n_f \left(-\frac{18841}{162} - \frac{2236}{9}\zeta_3 + 4\zeta_4 + \frac{400}{3}\zeta_5 \right) + C_A {C_F}^2 T_F n_f \left(\frac{3410}{81} \right. \\
&\left. + \frac{2624}{9}\zeta_3 + 152\zeta_4 + \frac{80}{3}\zeta_5 \right) + {C_F}^3 T_F n_f \left(\frac{1700}{27} - \frac{232}{3}\zeta_3 - 144\zeta_4 - 160\zeta_5 \right) \\
&+ \frac{d_F^{abcd} d_F^{abcd}}{N_R} n_f \left(\frac{304}{3} - 64\zeta_3 \right) + C_A C_F {T_F}^2 {n_f}^2 \left(-\frac{146}{27} + \frac{352}{9}\zeta_3 - 32\zeta_4 \right) \\
&+ {C_F}^2 {T_F}^2 {n_f}^2 \left(\frac{2876}{81} - \frac{352}{9}\zeta_3 + 32\zeta_4 \right) + C_F {T_F}^3 {n_f}^3 \left(-\frac{40}{9} + \frac{128}{27}\zeta_3 \right). \qquad (107)
\end{aligned}
$$

We have verified the correctness of these two quantities, by verifying fermion number conservation ($\gamma^{-}_{\mathrm{NS}}(N=1) = 0$), and by confirming the a_s^4 contribution to the Gross-Llewellyn-Smith (GLS) sum rule with [176].

4.4 CHAPTER CONCLUSION

In this chapter we have computed two classes of four-loop objects.

First, we have computed the finite pieces of all the QCD propagators and vertices with one vanishing momentum with generic colour group exactly in terms of master integrals. The results of these calculations can be used to convert quantities such as the beta function from $\overline{\mathrm{MS}}$ to any momentum subtraction scheme with a nullified momentum (see section 6.5).

4

Second, we have computed Mellin moments of four-loop splitting functions and coefficient functions. These are used as basic ingredients for collision processes, such as Higgs production [81]. We have computed Mellin moments $N = 2, 4, 6$ for the non-singlet case and $N = 2, 4$ for the singlet case. Additionally, we have calculated $N = 1, 3, 5$ of vector-axial interference F_3 [8]. By computing to $N = 40$ and beyond, we have reconstructed the all-N n_f^2 contribution to the four-loop non-singlet splitting function and the n_f^3 contribution to the four-loop singlet splitting function [5].

Using the OPE method, we have computed up to $N = 16$ for the non-singlet splitting function. For the large-n_c limit, we have computed up to $N = 19$ [115]. This allowed for an all-N reconstruction and yielded a new term to the four-loop planar cusp anomalous dimension.

4.4.1 *Findings and main conclusion*

We have computed the complete finite piece of the four-loop propagators and vertices and four-loop higher Mellin moments, using the FORCER program. So far, other currently existing programs were unable to calculate these objects. Thus, these results serve as valuable contributions to the field of high precision calculations. Moreover, we may also conclude that the effectiveness of FORCER has given an adequate and sufficient answer to **RQ2**.

4.4.2 *Future research*

A first goal for future research is to compute the finite pieces of propagators and vertices at five loops. However, this is a formidable challenge. So far, most methods used for computations at five loops use Infrared Rearrangement (IRR), which modifies the finite terms. A direct computation would require a five-loop Forcer equivalent.

A second topic for future research is to compute higher Mellin moments. Calculating Mellin moments using the optical theorem method becomes quite hard for P_{gg} after $N = 4$. Due to its scaling behaviour, the OPE method is promising to compute more moments of the singlet case. An approximation of the x-space four-loop splitting functions derived from these future results will improve the predictions for three-loop Higgs production [81, 82, 155].

5 INFRARED REARRANGEMENT

In this chapter we focus on answering

> **RQ3**: *Can we compute the poles of five-loop diagrams using only four-loop diagrams more efficiently?*

The appearance of divergences (poles) has perhaps been the most serious problem of Quantum Field Theory (QFT). In particular, the treatment of Ultraviolet (UV) divergences in Feynman diagrams at higher loops took many years to develop. Besides UV divergences, another class of divergences are encountered in the limit of vanishing internal and external masses. These divergences are known as collinear and soft divergences and are often collectively called Infrared (IR) divergences. Whereas the UV and initial state-collinear divergences can be renormalised into physical parameters and parton densities respectively, the soft IR divergences are known to cancel in the sum over all Feynman diagrams [177, 178] contributing to a particular observable.

Dimensional regularisation [179, 180] is the prevailing way to deal with the divergences of a QFT as it conserves both Lorentz and Gauge invariance. Furthermore it regulates UV and IR divergences at the same time and in a similar fashion, by expressing all divergences as poles in $\epsilon = (4 - D)/2$, where D is the dimension of space-time. This makes it particularly convenient for analytic calculations, which are often the preferred way to perform multi-loop calculations. The problem to obtain a Laurent series in ϵ of a general higher-loop Feynman integral remains a challenge and is an active field of research. Driven by the need for precision, enormous progress has been made in the development of general methods, based on differential equations [181–183], Mellin-Barnes representations [184–187], sector decomposition [188–190], analytic regularisation [191] and finite Master integrals [192] using integration-by-parts (IBP) identities [42, 92, 93, 95, 96]. While these methods are general, some either require computationally expensive IBP identities or lead to intractable large expressions, which make analytic evaluations very challenging.

A method to renormalise a Feynman diagram or amplitude is the Bogolioubov, Parasiuk, Hepp and Zimmermann (BPHZ) renormalisation scheme [190, 193, 194]. The BPHZ method is implemented by acting onto a given Feynman integral with the recursive BPHZ R-operation and is based solely on the graph theoretic properties of the underlying Feynman graph. The R-operation subtracts from a Feynman graph a number of counterterms which precisely capture the complicated combinatoric structure of the superficial, sub-, and overlapping UV divergences present in Feynman diagrams at arbitrary loop order. The BPHZ renormalisation prescription is not unique, in the sense that the definition of a counterterm operation in the R-operation can be adjusted to change to another renormalisation scheme. As such the

BPHZ renormalisation can also be defined in the minimal subtraction (MS) scheme [130] of dimensional regularisation [195]. Since the counterterms generated by the R-operation belong to a simpler class of Feynman graphs (lower loops or factorisable) than the original Feynman graph, the R-operation provides a prescription to compute the Laurent series in ϵ of any IR convergent Feynman graph. Interestingly, it has been shown the combinatorial structure of the R-operation gives rise to a Hopf algebra [196, 197].

However, in QFTs which contain massless particles the R-operation is not sufficient to render all Feynman graphs finite, due to the presence of IR divergences. A generalisation of the R-operation called the R^*-operation, was suggested more than thirty years ago by Chetyrkin, Tkachov and Smirnov in [45, 133, 134]. The R^*-operation is capable of subtracting both the ultraviolet and the infrared divergences of Euclidean *non-exceptional Feynman graphs* (Feynman graphs with non-exceptional external momenta). As a result, a powerful technique known as Infrared Rearrangement (IRR) [132] can be applied to reroute the external momenta of a Feynman diagram and to set masses to zero, in such a way as to maximally simplify its calculation. Critically, the IRR procedure does not alter the behaviour of the superficial UV divergence of a Feynman graph, but may lead to the creation of new IR divergences. The R^*-operation can be used to track and subtract these extra IR divergences. In analogy to Zimmermann's forest formula as a solution to the recursive R-operation, a solution to the recursive R^*-operation can be written as a generalised infrared forest formula [45, 135]. Several theorems concerning the validity and correctness of the R^*-operation have been proven in [135].

The R^*-operation has been used in a large number of milestone multi-loop Quantum Field theoretic calculations, such as the recent computation of the five-loop beta function in QCD [4, 40] or the calculation of the five- and six–loop anomalous dimensions in ϕ^4-theory [198–202]. Other applications include the calculation of the hadronic R-ratio [176, 203], the quark mass and field anomalous dimensions at five-loop in QCD [127, 204], as well as the inclusive Higgs decay rate into a light quarks [205]. The applications of the R^*-operation may be classified into two different types:

a) the local R^*-method,

b) the global R^*-method.

The local R^*-method is based on directly applying the R^*-operator to individual Feynman diagrams. The global R^*-method seeks to globally IR rearrange an entire amplitude (a sum of appropriately weighted Feynman graphs) instead. By applying the R^*-operator to this decomposition, it has been possible to work out global counterterms in a number of different calculations. In fact, all applications of the R^*-operation in QCD beyond the three-loop level (except for [4], which uses the method described in this work) have been based on this global approach, whereas the local R^*-method has been used beyond three loops only in ϕ^4-theory. There are at least two complications that arise in a direct application of the R^*-operation to

QCD. First, there is a performance challenge of constructing counterterms for billions of integrals at the five-loop level in the input. Second, QCD diagrams introduce irreducible numerators, which require careful treatment. The R^* methods which have been advocated and used in the calculations in ϕ^4-theory [45, 206–208] are not sufficient to deal with these extra complications. However, the advantage of the local approach over the global approach is that the same procedure can be applied to any process on a term by term basis. Instead, the global R^*-method has to be worked out independently for different correlators, which was highly challenging for the five loop QCD beta function [40] and has at present only been achieved for the $SU(3)$ gauge group.

In this work we shall develop a local R^*-framework which allows us to compute the pole parts of non-exceptional Feynman graphs with arbitrary numerators, including tensors. To achieve this goal we will identify the basic building blocks of all UV counterterms as the set of scaleless vacuum tensor Feynman graphs with logarithmic superficial degree of divergence. Such a Logarithmic Tensor Vacuum Graph (LTVG) has more symmetry than the graph it was derived from and allows for more dot products to be rewritten, which vastly reduces their number and their complexity. We will further show that all IR counterterms can be neatly extracted from the UV counterterms of LTVGs. This framework has already been used to compute the five loop beta function for a general gauge group [4]. In this work we present the results for the poles of all non-factorisable (those which do not factorise into products of Feynman graphs of lower loops) Feynman graphs appearing in ϕ^3-theory in four dimensions. A subset of these, so far unknown, Feynman graphs is likely to provide good candidates for master integrals. We anticipate that these results will provide a useful cross-check for future evaluations by alternative methods, which may also include finite parts.

The chapter is organised as follows. In section 5.1 we will review the notions of power counting for both UV and IR–divergences which can occur in non-exceptional Feynman graphs. In section 5.2 we will review the R-operation. In section 5.3 we discuss contraction anomalies and how to extend the R-operation to Feynman graphs with generic numerators. In section 5.4 we review the R^*-operation, introduce LTVGs, derive a new representation of the IR counterterm, and discuss how to extend the R^*-operation to tensor Feynman graphs. In section 5.5 we show applications at five loops. We discuss some differences between our method and the literature in section 5.6. Finally, we provide conclusions and an outlook in section 5.7.

5.1 DIVERGENCES IN EUCLIDEAN NON-EXCEPTIONAL FEYNMAN GRAPHS

We impose that all Feynman graphs to be considered in the following are Euclidean and will always have *non-exceptional external momenta*. To be precise, this means that no linear combination of a subset of external momenta $p_1, \dots, p_n$ vanishes:

$$\sum_{i\in I} p_i \neq 0, \qquad \text{for } I \text{ any subset of } \{1, \dots, n\}. \tag{108}$$

The divergences which exist in non-exceptional Feynman graphs can be classed into two types:

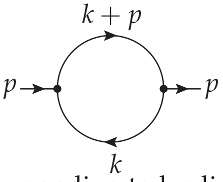
(i) UV divergences, related to infinite loop momentum configurations,

(ii) IR divergences, related to vanishing loop momentum configurations.

In the following we shall review the basic notions of power counting for UV and IR divergences (section 5.1.1 and section 5.1.2 respectively) and thereby introduce the necessary language which will be needed later to define the R- and R^*-operations.

5.1.1 *UV divergences in Feynman graphs*

An important notion in UV power counting is the Superficial Degree of Divergence (SDD). To compute the superficial degree of divergence $\omega(G)$ of a Feynman graph G, one rescales all of its independent loop momenta k_i with a parameter λ, i.e., $k_i \to \lambda k_i$. The leading power of λ in the limit $\lambda \to \infty$ then defines the superficial degree of divergence of the Feynman graph. If $\omega \geq 0$, the integral is called superficially UV divergent. Let us consider a simple example:

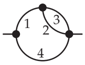


$$G = \quad = \int \frac{d^D k}{i\pi^{D/2}} \frac{1}{k^2(k+p)^2} \,. \tag{109}$$

Rescaling $k \to \lambda k$ and expanding to leading order around $\lambda \to \infty$ we get

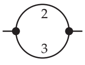
$$\lim_{\lambda\to\infty} G \sim \lambda^{D-4} \int \frac{d^D k}{i\pi^{D/2}} \frac{1}{k^4} \,. \tag{110}$$

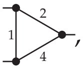
Hence in the limit $D = 4$ we find $\omega(G) = 0$, which is referred to as a logarithmic divergence. The notion of superficial degree of divergence can also be used to identify subdivergences, where only some loop momenta k_i diverge. Let us remind the reader at this point that in any parametrisation each loop momentum always flows around in a "loop". Thus, the contributions to a subdivergence are due to the propagators in the loop and the loop momentum in the numerator. This implies that UV divergences are always associated to one-particle-irreducible (1PI)[1] subgraphs of loop one or higher. We will call any 1PI subgraph, which has a non-negative superficial degree of divergence, a *UV subgraph*. As an example consider

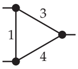


$$G = \quad , \qquad \omega(G) = 0 \,, \tag{111}$$

where we have introduced the line labels $1, 2, 3, 4$. Let us call the subgraph consisting of lines $a, b, c, \ldots$ and those vertices connecting them as $\gamma_{abc\ldots}$, such that







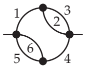
$$\gamma_{23} = \quad , \qquad \gamma_{124} = \quad , \qquad \gamma_{134} = \tag{112}$$

[1] A graph is 1PI if it can not be separated into two by cutting any one propagator.

The superficial degrees of divergence of these subgraphs are given by

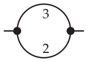
$$\omega(\gamma_{23}) = 0, \qquad \omega(\gamma_{124}) = -2, \qquad \omega(\gamma_{134}) = -2, \tag{113}$$

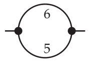
It is easy to see that γ_{23} is nothing but an insertion of our previous example and as such it can diverge by itself. This is an instance of a *UV subdivergence*. We note that the other one-loop subgraphs are finite. Another example is given by

$$G = $$


$$. \tag{114}$$

This Feynman graph has the following UV subgraphs:

$$\gamma_{23} = $$


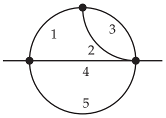
$$, \qquad \gamma_{56} = $$


$$, \tag{115}$$

with superficial degrees of divergence

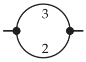
$$\omega(\gamma_{23}) = 0, \qquad \omega(\gamma_{56}) = 0\,. \tag{116}$$

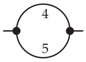
The subgraphs γ_{23} and γ_{56} are *strongly disjoint*: they have no common lines or vertices. As such, the loop momenta in both graphs can diverge independently. For dimensionless vertices, the same holds for *weakly disjoint* subgraphs, which may share vertices. We will call any set of pairwise strongly disjoint UV subgraphs *UV disjoint*. As a last example let us consider

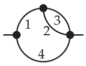
$$G = $$


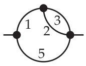
$$, \qquad \omega(G) = 2\,. \tag{117}$$

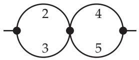
The five UV subdivergences of this Feynman graph are given by

$$\gamma_{23} = $$


$$, \gamma_{45} = $$


$$, \gamma_{1234} = $$


$$, \gamma_{1235} = $$


$$, \gamma_{2345} = $$


$$, \tag{118}$$

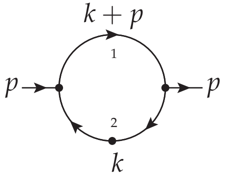
with

$$\omega(\gamma_{23}) = \omega(\gamma_{45}) = \omega(\gamma_{1234}) = \omega(\gamma_{1235}) = \omega(\gamma_{2345}) = 0\,. \tag{119}$$

All of these subgraphs pairwise overlap: they share at least one common vertex or line. Thus, no combination of these subgraphs can diverge independently.

5.1.2 *IR divergences in Feynman graphs*

In analogy to the UV one can quantify the degree of IR divergence of a Feynman graph or subgraph G by introducing the notion of an *IR superficial degree of divergence* $\tilde{\omega}(G)$. Let us consider the simple example

$$G = $$


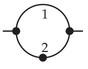
$$= \int \frac{d^D k}{i\pi^{D/2}} \frac{1}{(k^2)^2 (k+p)^2}\,. \tag{120}$$

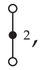
We will use a small dot to indicate a vertex. A vertex which has only two edges in effect creates a squared propagator. Such squared propagators are the simplest instance of an IR divergence in non-exceptional Feynman graphs. We note that a graph can only have a superficial IR divergence if no external momenta flow through it. To study the IR properties of the line with a dot, we route the external momentum through the top line and compute its superficial degree of divergence by rescaling, $k \to \lambda k$. This time we wish to extract the leading power $-\tilde{\omega}$ of λ in the limit $\lambda \to 0$. Performing this rescaling and taking the limit we find

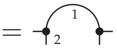
$$\lim_{\lambda \to 0} G \sim \lambda^{D-4} \frac{1}{p^2} \int \frac{d^D k}{i\pi^{D/2}} \frac{1}{k^4} . \tag{121}$$

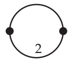
And thus we get that $\tilde{\omega}(G) = 0$ for $D = 4$, indicating that the integral diverges logarithmically in the IR. One interesting difference between IR and UV divergences is that IR divergences are not usually associated to 1PI subgraphs, but they are (with minor exceptions) associated to connected subgraphs, which, as it turns out, is why many of the features of UV divergences explained above extend to the case of IR divergences. The exception is always related to self-energy insertions, of which we will give an example further below. This case was in fact missed in the original R^*-paper [133] and was later corrected in [134].

We will use the notation $\gamma'_{abc...}$ to identify a certain *IR subgraph* containing lines $a, b, c, \ldots$ and those vertices which connect only to $a, b, c, \ldots$, but no other lines. That is, the graph γ' does not contain any vertices through which it is connected to the remaining graph; such vertices will be called external to the graph γ'. We define the graph $\bar{\gamma} = G \backslash \gamma'$ to be the *remaining graph* where the IR subgraph is deleted from G. Further we define the *contracted vacuum graph* $\tilde{\gamma} = G/\bar{\gamma}$ by contracting $\bar{\gamma}$ to a point in G. An example is given by

$G =$ 

$, \gamma'_2 =$ 

$, \quad \bar{\gamma}_2 =$ 

$, \quad \tilde{\gamma}_2 =$ 

$,$ (122)

where we have use filled dots to denote internal vertices and hollow dots to denote external vertices. The momenta of the IR subgraph are considered as external to the remaining graph and are indicated with amputated lines in $\bar{\gamma}$. It is worth remarking here that the associated contracted vacuum graph $\tilde{\gamma}$ contains the same IR divergent behaviour as the original graph, even though it is scaleless and thus vanishing in dimensional regularisation. Indeed, the integrand of eq.(121) is nothing but the integrand of $\tilde{\gamma}_2$.

Now that we have defined the appropriate notations, we describe the conditions for an IR subgraph to be *IR irreducible* (IRI) [134]:

(i) No external momentum flows into an internal vertex of γ',

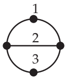
(ii) γ' cannot contain massive lines,

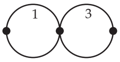
(iii) the associated contracted (vacuum) graph $\tilde{\gamma}$ cannot contain cut-vertices [2],

[2] A cut-vertex is a vertex which when cut separates a graph into two or more disconnected subgraphs.

(iv) [for insertions] each connected component in the remaining graph $\bar{\gamma}$ should be 1PI after shrinking massive lines and welding the vertices together that have external momenta attached to it.

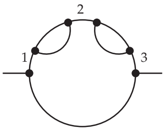
Rule (i) and (ii) follow straightforwardly from the fact that such propagators are IR regulated. Rule (iii) prevents cases such as:

$G =$ 

$, \tilde{\gamma}_{13} =$ 

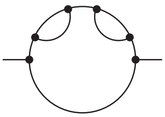
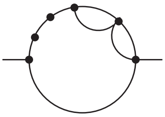
$, \qquad (123)$

where in fact γ_1' and γ_3' are two separate IR subgraphs. Finally, rule (iv) treats IRI subgraphs that appear disjoint in the diagram:

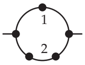
$G =$ 

$, \qquad (124)$

where γ_{12}', γ_{13}', and γ_{23}' are not IRI if lines 1,2,3 are all massless propagators and no additional external momentum flows through them. In that case, only γ_{123}' is IRI, which can easily be seen by this equivalence:

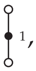

$=$

$. \qquad (125)$

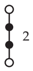
Let us now consider the following case,

$G =$ 

$. \qquad (126)$

This graph contains two distinct IR subgraphs

$\gamma_1' =$ 

$, \qquad \gamma_2' =$ 

$, \qquad (127)$

with the following degree of divergences:

$$\tilde{\omega}(\gamma_1') = 0, \qquad \tilde{\omega}(\gamma_2') = 2\,. \qquad (128)$$

One may wonder whether γ_1' and γ_2' could diverge simultaneously. However, since momentum conservation at each vertex demands the incoming momentum to flow through at least one of the two propagators, only one of the momenta can vanish 'at a time'. It is useful to define, in analogy to the UV, the notion of IR disjointness. Loosely speaking we will call any set of non-overlapping (no common lines or internal vertices) IR subgraphs which can diverge simultaneously *IR disjoint*. This can be formulated more precisely as follows. A set S' of IR subgraphs of G is said to be IR disjoint if the following criteria are met:

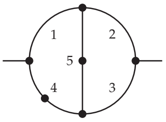
(i) the IR subgraphs in S' are pairwise non-overlapping,

(ii) no other IR subgraphs can be composed from any subset of IR subgraphs in S',

(iii) the remaining graph $G \setminus S'$, defined by deleting in G all lines and internal vertices of all IR subgraphs in S', is connected.

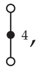
For the Feynman graph above, each of γ'_1 or γ'_2 then form themselves an IR disjoint set (of one element) but their union does not. To see this in practice let us consider the Feynman graph

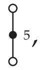
$$G =$$

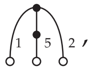


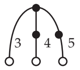
$$, \qquad (129)$$

which has the following IR subgraphs,

$$\gamma'_4 =$$


$$, \quad \gamma'_5 =$$


$$, \quad \gamma'_{125} =$$


$$, \quad \gamma'_{345} =$$


$$(130)$$

with superficial degree of divergences,

$$\tilde{\omega}(\gamma'_4) = \tilde{\omega}(\gamma'_5) = \tilde{\omega}(\gamma'_{125}) = 0, \qquad \tilde{\omega}(\gamma'_{345}) = 2\,. \qquad (131)$$

Let us also consider the IR disjoint sets of IR subgraphs, which can occur in this example. From momentum flow considerations one can see that the only possible choice of having several disconnected IR divergences occurring simultaneously would be $\{\gamma'_4, \gamma'_5\}$. However momentum conservation (at the vertex connecting lines 3, 4 and 5) would in this case also force propagator 3 to vanish, leading to the divergence γ'_{345}, which we have already covered. In such a case we say that union of the IR subgraphs γ'_3 and γ'_4 *compose* γ'_{345}. Thus, this combination is prohibited by rule (ii).

5.2 THE R-OPERATION IN THE MS-SCHEME

In the following we will review the R-operation in the MS-scheme [195]. In the MS-scheme divergences are isolated as poles in the dimensional regulator ϵ. It is convenient to introduce a pole operator K, which acting on an arbitrary meromorphic function $F(\epsilon)$, with Laurent series

$$F(\epsilon) = \sum_{n=-\infty}^{\infty} c_n \epsilon^n\,, \qquad (132)$$

will return only its poles, i.e.,

$$KF(\epsilon) = \sum_{n=1}^{\infty} \frac{c_{-n}}{\epsilon^n}\,. \qquad (133)$$

The K-operation acting on a product of meromorphic functions fulfils what is known as a *Rota-Baxter algebra*:

$$K(AB) = K(AK(B)) + K(BK(A)) - K(A)K(B)\,. \tag{134}$$

The R-operation will make any purely UV divergent Feynman graph G finite. The following equation must therefore hold in any renormalisation scheme:

$$K\,R\,G = 0\,. \tag{135}$$

Writing $R = 1 + \delta R$, where δR denotes the counterterms generated by R, we obtain:

$$KG = -K\delta RG\,. \tag{136}$$

In section 5.2.1 we will define the R-operation in the MS-scheme. Subsequently, we will show some examples of the R-operation in section 5.2.2.

5.2.1 *Definition of the R-operation in the MS-scheme*

The R-operation renders a Feynman graph finite by subtracting from it all of its UV divergences. In section 5.1.1 we introduced the concept of a UV disjoint set of subgraphs. This concept lies at the very heart of the R-operation and makes up one of the two fundamental ingredients with which the R-operation is equipped. These are:

(i) a notion of a set of disjoint connected subgraphs,

(ii) a (non-unique) counterterm operation.

Ingredient (i) informs the R-operation about the set of distinct UV divergences which may appear in the domain of loop-integration. The R-operation then associates a counterterm to each UV disjoint set of UV subgraphs. Mathematically the R-operation is then expressed as a sum over the different required counterterms:

$$RG = \sum_{S\in W(G)} \Delta(S) * G/S\,. \tag{137}$$

Here $\Delta(S)$ denotes the counterterm operation acting on the "singular" part of G specified by S, while G/S represents the "non-singular" part of G, constructed by contracting in G all subgraphs γ to points. G/S is also called the remaining diagram. S, sometimes referred to as a *spinney*, is a set of UV disjoint UV subgraphs of G and $W(G)$, which is sometimes referred to as a *wood*, is the set of all such sets S, which can be constructed from the UV subgraphs of G. $W(G)$ also includes the spinney containing the full graph, i.e., $\{G\}$. The $*$-operation takes the role of an insertion operator in the presence of non-logarithmic divergences, but reduces to a simple product for logarithmic divergences. In the case of $S = \{G\}$, we obtain the trivial identity

$$\Delta(\{G\}) * G/\{G\} = \Delta(G) * 1 = \Delta(G)\,, \tag{138}$$

to include a counterterm for the *superficial divergence* of G. Further $W(G)$ includes the empty set $\emptyset$, whose counterterm is included simply to include the full graph G in the sum, i.e.,

$$\Delta(\emptyset) * G/\emptyset = 1 * G = G\,. \qquad (139)$$

Up until this point, the R-operation is general, but we haven't defined yet what the counterterm operation Δ is. Although it has to satisfy certain criteria, the counterterm operation Δ is not unique. This non-uniqueness is directly related to the renormalisation scheme and regulator dependence which is always present in any (non-finite) Quantum Field Theory. As such, the non-uniqueness of the counterterm operation should come as no surprise. Nevertheless a minimal prescription, the MS-scheme [130], can be defined in dimensional regularisation, and it is this scheme which we shall employ in the following. The counterterm $\Delta(S)$ must isolate the divergence associated to the disjoint singular subgraphs γ in S. Given their disjointness it is clear that $\Delta(S)$ must factorise,

$$\Delta(S) = \prod_{\gamma\in S} \Delta(\gamma)\,. \qquad (140)$$

Let us now discuss the minimal counterterm operation $\Delta(\gamma)$. One may be tempted to replace $\Delta(\gamma)$ with $-K\gamma$, i.e., its divergent parts in MS. However, $-K\gamma$ is not a local operation if γ contains subdivergences. These subdivergences will already have been accounted for in the sum over all S. The solution to this problem is to isolate the superficial divergence of γ by subtracting from it all its subdivergences. This can be achieved by using a variant of the R-operation, where we just omit the counterterm of the full graph G in the sum. We will call this operation the $\bar{R}$-operation:

$$\bar{R}G = \sum_{S\in\bar{W}(G)} \Delta(S) * G/S, \quad \text{with} \quad \bar{W}(G) = W(G)\setminus\{G\}\,. \qquad (141)$$

5

Finally we must add the K-operator in order to comply with the MS-scheme (where no finite pieces are kept). This leads us to the definition of the counterterm operation:

$$\Delta(\gamma) = -K\bar{R}\gamma\,. \qquad (142)$$

The counterterm operation of a graph $\Delta(\gamma)$ in the MS-scheme can be shown to be a polynomial in the external momenta and masses of the graph γ of homogeneous degree $\omega(\gamma)$, see e.g. [195]. This implies that the counterterm operation can be replaced with its Taylor expanded version:

$$\Delta(\gamma) = \sum_{n=0}^{\omega(\gamma)} \mathcal{T}^{(n)}_{\{p_i\}}\Delta(\gamma), \qquad (143)$$

where $\{p_i\} = \{p_1,\ldots,p_n\}$ shall denote the set of external momenta of γ and $\mathcal{T}^{(n)}_{\{p_i\}}$ is a Taylor expansion operator, defined in the usual sense:

$$\mathcal{T}^{(w)}_{\{p_i\}} f(\{p_i\}) = \sum_{\alpha_1+\ldots+\alpha_n=w} \prod_{i=1}^{n} \frac{(p_i\cdot\partial_{p'_i})^{\alpha_i}}{\alpha_i!} f(\{p'_i\})\Big|_{p'_i=0}, \qquad \partial^\mu_p = \frac{\partial}{\partial p_\mu}\,. \qquad (144)$$

We can now use the fact that derivatives with respect to external momenta commute with the $\bar{R}$ operation,

$$[\partial^{\mu}_{p_i}, \bar{R}] = 0\,, \tag{145}$$

which is true as long as no IR divergences are created, see [195] for a proof, to derive

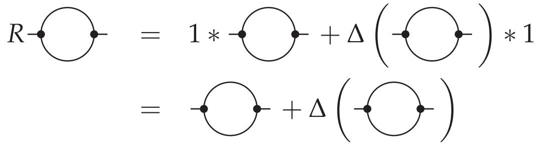
$$\Delta(\gamma) = -\sum_{n=0}^{\omega(\gamma)} K\bar{R}\ \mathcal{T}^{(n)}_{\{p_i\}}\gamma\,. \tag{146}$$

Thus, we can further simplify the expression for the UV counterterm by replacing γ with its Taylor expanded version. Finally this leads us to the definition of the $*$-product in the presence of higher degree divergences, whose task is to insert the polynomial dependence on the external momenta $\{p_i\}$ back into the contracted graph G/S.

We shall see some examples of the R-operation for massive diagrams below.

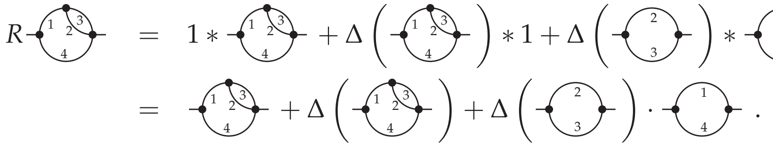
5.2.2 *Examples of R-operations*

We shall start with our trivial example from above



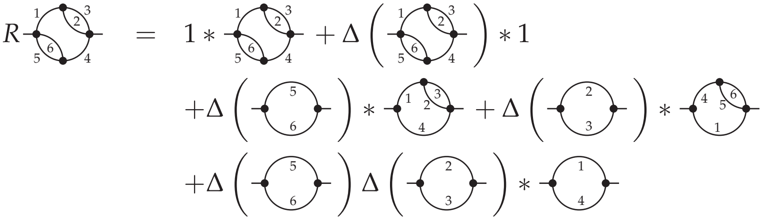
(147)

Note that we dropped the $*$ for the standard multiplication $\cdot$ in the second line. A less trivial example is given by



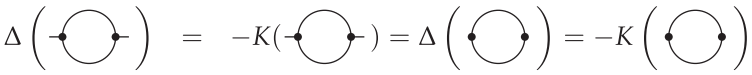
(148)

A three-loop example is given by

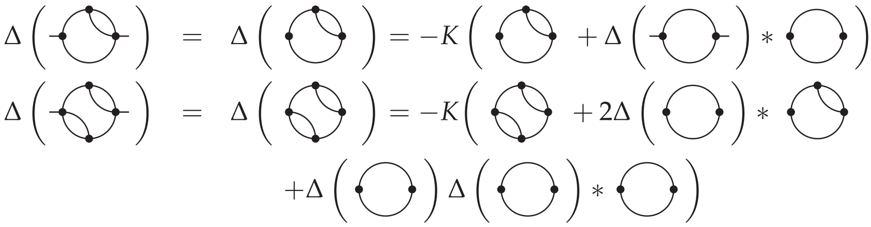


(149)

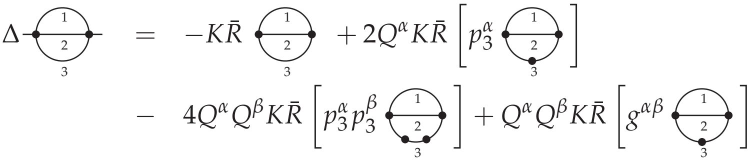
Assuming that all propagators are massive, we can recursively obtain the values of Δ from those of massive tadpoles:



(150)



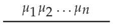
Next, we consider a massive quadratic graph, which we Taylor expand:



We now see the emergence of tensor diagrams. In the next chapter we provide a consistent R formalism to compute those graphs.

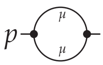
5.3 THE R-OPERATION FOR GENERIC FEYNMAN GRAPHS IN MS

Let us call a *generic Feynman graph* any Feynman graph which contains products of vectors or scalar products of loop momenta in the numerator. In order to diagrammatically denote generic Feynman graphs we introduce the following Feynman rule:



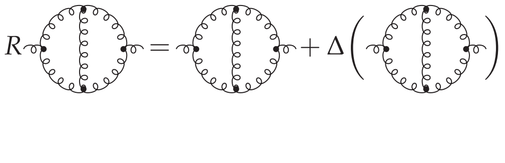
$$= \frac{k^{\mu_1}k^{\mu_2}\dots k^{\mu_n}}{k^2}\,. \qquad (155)$$

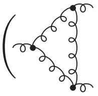
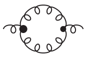
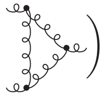
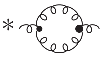
An example of a simple generic Feynman graph is given by



$$p = \int \frac{d^Dk}{i\pi^{D/2}}\frac{k^\mu}{k^2}\frac{(k-p)_\mu}{(k-p)^2}\,. \qquad (156)$$

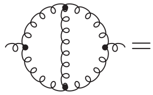
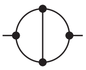
Before diving into the subtleties related to the application of the R-operation to generic Feynman graphs, let us briefly consider the possible uses of such a formalism and consider the potential difficulties which one may encounter. First of all, it is important to understand that the R-operation in the MS-scheme that we have presented above can readily be applied to Feynman graphs containing particles of arbitrary spin: besides scalars, also fermions, photons, gluons, gravitinos or gravitons, etc., connected by appropriate vertices, can straightforwardly be accommodated. Furthermore, the R-operation can be used to construct the renormalisation counterterms diagram by diagram, in a way completely equivalent to that of renormalising the operators of the corresponding Lagrangian. It is instructive to see how this would work for a simple example in QCD:



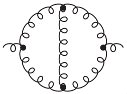
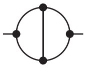
$+\Delta\Big($  $\Big) *$  $+\Delta\Big($  $\Big) *$  $.$ (157)

Here the slightly fatter dot is used to indicate the insertion point of the triangle counterterm into the remaining graph. We remark that the triangle counterterm is *a* 3-gluon vertex: that is, it is a local operator linear in the three external momenta and depends on the three external colour and Lorentz indices.

The fact that in the MS-scheme the counterterm operation Δ is based on the K-operation leads to what one may call *contraction anomalies*. To illustrate this, let us consider the numerators created by the evaluating the Feynman rules[3], i.e.,

 $= \sum_{k=0}^{2} \epsilon^k N_k(\{q_i, p_i\})$  $.$ (158)

Here $N_k(\{k_i, p_i\})$ are polynomials in the external and internal momenta, p_i and q_i respectively, and we have made the dependence on ϵ explicit. The contraction anomaly now emerges in the following inequality:

$R\Bigg($  $\Bigg) \neq \sum_{k=0}^{2} \epsilon^k R\Big(N_k(\{q_i, p_i\})$  $\Big),$ (159)

and follows from the fact that the K-operation does not generally commute with factors of ϵ. The right hand side will thus lead to a different result for the renormalised Feynman graph. While both sides of (159) would still be finite, the right hand side will no longer correspond to the finite value which one should have obtained in the *original* MS-scheme. Given the argument above, it is clear that the R-operation for a generic Feynman graph brings with it a certain arbitrariness when it comes to the definition of its renormalisation scheme. Consequently, the operation $\bar{R}$ cannot be used to correctly extract the UV counterterm of a QCD diagram after the contraction of the Feynman rules has been carried out. Nevertheless, the R-operation can still be used after contractions to build valid counterterms which render any Feynman graph finite. In turn, these counterterms can be used to extract the poles in ϵ of a given Feynman graph via eq. (136): $K(G) = -K\delta RG$.

In section 5.3.1 we discuss contraction anomalies concerning tensor reductions. Subsequently, we consider a second kind of contraction anomalies due to counterterm factorisation in section 5.3.2.

[3] Performing the Lorentz contractions among vertices and propagators and evaluating traces of gamma-matrices.

5.3.1 *Contraction anomalies and tensor reduction*

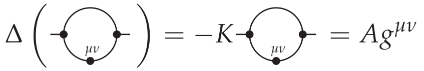
The counterterm operation Δ and the K-operation do not naively commute with the contraction operation, as was discussed in [e.g., 195, 209]. It is an intricate feature of dimensional regularisation. In general we have

$$P^{\mu_1\ldots\mu_n}\Delta\left(\gamma_{\mu_1\ldots\mu_n}\right) \neq \Delta\left(P^{\mu_1\ldots\mu_n}\gamma_{\mu_1\ldots\mu_n}\right) \,, \tag{160}$$

where the *contraction operator* P may be any even rank tensor built purely from products of metric tensors $g^{\mu\nu}$:

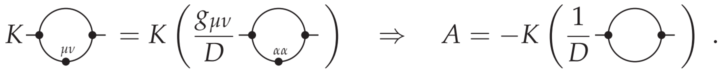
$$P^{\mu_1\ldots\mu_n} = \sum_\sigma c_\sigma g^{\mu_{\sigma(1)}\mu_{\sigma(2)}}\cdots g^{\mu_{\sigma(n-1)}\mu_{\sigma(n)}} \,. \tag{161}$$

This means that special care has to be taken to reduce counterterms of tensor graphs to counterterms of scalar diagrams. Let us consider some examples to clarify this issue further. A simple one-loop example is given by



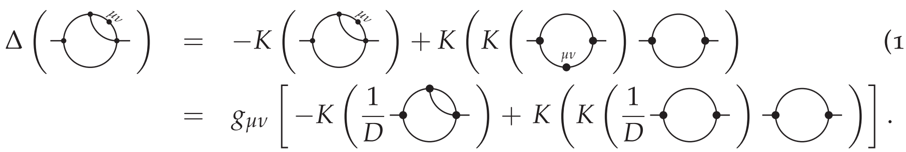
(162)

where A is some constant to be determined. In general one may expect that there could be a second tensor structure of kind $Q^\mu Q^\nu$, but this is excluded since a counterterm of logarithmic SDD cannot depend on its external momenta. One may be tempted to extract the value of A by contracting both sides of (162) with a projector $g_{\mu\nu}/D$. However, this will lead to the wrong result, precisely because contraction does not commute with the K-operator. The correct result for the coefficient A is obtained by performing the tensor reduction inside the K-operation:



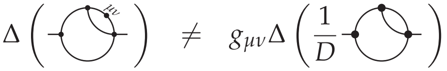
(163)

Here we used that the metric tensor can be commuted with a K-operation as long as this does not lead to any contractions inside that K. Applying a tensor reduction for a Δ-operation at higher loops is more involved: a safe procedure is to first apply the counterterm operation recursively to obtain an expression in terms of nested K-operations, and then apply the tensor reduction iteratively, starting inside the most inner K and then commuting tensors outwards. Let us consider how this works for the following simple two loop example:

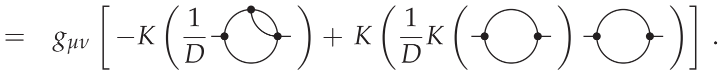


(164)

We see that one cannot naively extract the value of Δ of a tensor Feynman graph from that of its tensor reduced version:



(165)


= $g_{\mu\nu}$ $\Big[-K\Big(\frac{1}{D}$ $\Big) + K\Big(\frac{1}{D}K\Big($ $\Big)$ $\Big)\Big]$.

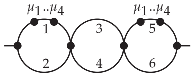
5.3.2 *Contraction anomalies and counterterm factorisation*

A second kind of contraction anomalies is encountered when one considers the counterterm operation acting on a product of graphs. For factorising scalar graphs the following factorisation formula holds (see appendix A for an inductive proof):

$$\Delta(G_1G_2) = \Delta(G_1)\Delta(G_2)\,. \tag{166}$$

The formula no longer holds when G_1 and G_2 are replaced with tensor graphs. To clarify this issue, let us consider the following example:

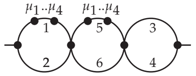
$$G_a =$$



$$. \tag{167}$$

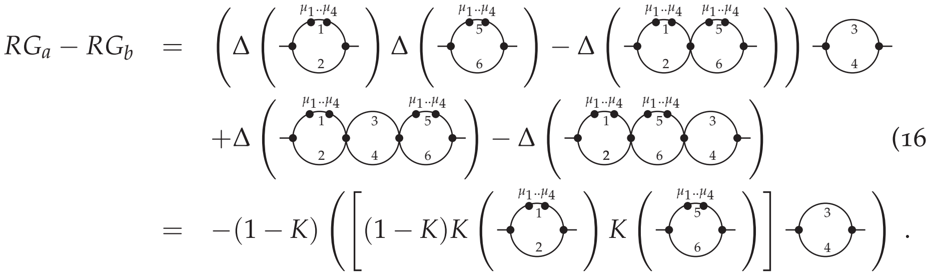
It is perhaps worth reiterating that this generic graph is rather unphysical. In *physical* graphs a Lorentz index of a particular propagator would always be contracted with its neighbouring vertices. Any uncontracted indices that enter a counterterm in a physical Feynman graph would thus always be "external" and should be treated as commuting with respect to the K-operation. This is rather different for the graph under consideration. Its indices are in some sense internal and yet they are uncontracted. A problem when acting the R-operation onto this graph becomes apparent when representing the graph in a different order:

$$G_b =$$



$$. \tag{168}$$

The difference of acting the R-operation on G_a and G_b then becomes


$RG_a - RG_b$ = $\Big(\Delta\Big($ $\Big)\Delta\Big($ $\Big) - \Delta\Big($ $\Big)\Big)$
$+\Delta\Big($ $\Big) - \Delta\Big($ $\Big)$
= $-(1-K)\Big(\Big[(1-K)K\Big($ $\Big)K\Big($ $\Big)\Big]$ $\Big)$.
$\mu_1..\mu_4$
1
2
3
4
5
6

$$(169)$$

$$= \text{finite but non-zero,}$$

where we used

$$R = (1-K)\bar{R}, \tag{170}$$

which in some sense guarantees finiteness. It shows, however, that reordering of the subgraphs effectively results in a transition to a different renormalisation scheme, as we anticipated in our discussion in the beginning of this chapter. The difference in schemes may be traced back to the breaking of (166), which is caused by a violation of the identity

$$K(K(A)K(B)) = K(A)K(B) \tag{171}$$

if A and B are contracted tensors. Another way to phrase this observation is that the Rota-Baxter algebra of eq.(134) is not valid for contracted tensor graphs.

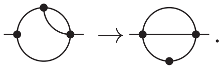
Even though both RG_a and RG_b are finite, this calculation clearly shows that the counterterm operation Δ is not invariant under reorderings. A possible choice to enforce the re-ordering property is to use the replacement

$$\Delta(G_1 G_2) \to \Delta(G_1)\Delta(G_2) \tag{172}$$

A convenient way to incorporate (172) into the R-operation is to introduce a modified notion of UV disjointness. If one defines a spinney to consist of a set of weakly disjoint subgraphs, i.e., subgraphs which contain no common lines but may have common vertices, the right hand side of (172) is automatically produced. Apart from leading to perfectly finite results, this *weakly disjoint renormalisation scheme* has the advantages that it restores invariance under re-ordering of cut-vertex connected subgraphs. It further simplifies calculations since the right hand side of (172) generally leads to fewer terms than the left hand side. A proof for the finiteness of the factorised renormalisation scheme is sketched for factorised graphs in appendix B.

5.4 THE R*-OPERATION

The R^*-operation [45, 133, 134] extends the R-operation to the subtraction of infrared divergences of Euclidean Feynman graphs with non-exceptional external momenta. The main power of the R^*-formalism derives from a trick known as infrared rearrangement (IRR). The trick uses the feature that the counterterm operation associated to a logarithmic superficially UV divergent Feynman graph is independent of its external momenta and masses. Infrared rearrangement allows one to compute the counterterm operation Δ of a graph G from a simpler rearranged version of it, called G'. We have essentially performed IR rearrangements in the context of the standard R-operation, which in the presence of internal masses can be exploited to set all external momenta to zero. In the absence of masses in propagators, IR rearrangement may lead to infrared divergences. This is illustrated in the following example:



(173)

In order to subtract the newly created IR divergences, the R-operation must be equipped with an infrared counterterm operation. We shall review the defintion of the R^*-operation in the following and extend it to generic Feynman graphs. We will furthermore introduce the concept of vacuum Feynman graphs of logarithmic SDD which will take centre stage in our formalism. We will also show that the concept offers an alternative description of the IR counterterm operation.

The remainder of this section is structured as follows. In section 5.4.1 we define the R^*-operation. We construct the infrared counterterm operation in section 5.4.2. Next, we provide some examples of the R^*-operation in section 5.4.3. Finally, we discuss properties of logarithmic vacuum graphs in section 5.4.4.

5.4.1 *Definition of the R^*-operation*

In section 5.1.2 we have already introduced the notions of IR subgraphs and IR disjoint sets of IR divergences. These are all the necessary notions to readily generalise the R-operation. First, we note that a given subgraph can never at the same time become IR and UV divergent. This trivially implies that overlapping UV IR counterterms do not need to be considered. The R^*-operation acting on a given graph G then takes the following form:

$$R^* G = \sum_{\substack{S \in W(G), S' \in W'(G) \\ S \cap S' = \emptyset}} \widetilde{\Delta}(S') * \Delta(S) * G/S \setminus S' \,. \tag{174}$$

We will explain the details of equation (174) in the following. The sum goes over all non-intersecting sets S and S', where S is a *UV spinney*, as before a set of UV disjoint UV subgraphs, and S', is an *IR spinney*, a set of IR disjoint IR subgraphs. As before $W(G)$ is the set of all UV spinneys S of G, whereas $W'(G)$ is the set of all IR spinneys S' of G. An efficient algorithm to construct the IR spinney is given in appendix C.

The IR counterterm operation $\widetilde{\Delta}$ factorises, similarly to the UV counterterm operation Δ, over disjoint IR subgraphs:

$$\widetilde{\Delta}(S') = \prod_{\gamma' \in S'} \widetilde{\Delta}(\gamma') \,, \qquad \Delta(S) = \prod_{\gamma \in S} \Delta(\gamma) \,. \tag{175}$$

The reduced graph $G/S \setminus S'$ is constructed by first shrinking all UV subgraphs in S into points in G, identically as in the case of the R-operation, and then deleting in G/S all the lines and vertices of all IR subgraphs in S'. The UV counterterm operation Δ is defined, identically to the case of the $\bar{R}$-operation, to isolate the superficial UV divergence. This is achieved by subtracting from the given Feynman graph or subgraph all UV subdivergences as well as all IR divergences. The UV counterterm operation is then defined as

$$\Delta(G) = -K\bar{R}^* G \,, \tag{176}$$

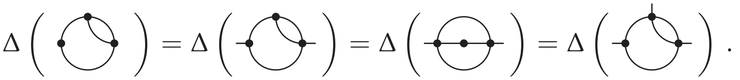
with the $\bar{R}^*$-operation being defined recursively via

$$\bar{R}^* G = \sum_{\substack{S \in \bar{W}(G), S' \in W'(G) \\ S \cap S' = \emptyset}} \widetilde{\Delta}(S') * \Delta(S) * G/S \setminus S' \,. \tag{177}$$

The proper UV wood $\bar{W}(G)$ (as in the case of the $\bar{R}$-operation) includes all UV spinneys apart from the one which consists of the graph G itself. Equivalently, we can extract the same UV counterterm from a set of *massless* vacuum integrals by Taylor expanding the graph G up to order $\omega(G)$ around all of its external momenta $\{p_i\}$ and masses $\{m_i\}$. We will call the scaleless vacuum tensor graphs of logarithmic SDD *logarithmic tensor vacuum graphs* (LTVGs). It leads us to the alternative definition in terms of LTVGs:

$$\Delta(G) = -K\bar{R}^* \mathcal{T}^{(\omega(G))}_{\{p_i,m_i\}} G\,. \tag{178}$$

In contrast to eq.(146) we have dropped in the above all terms in the Taylor series but the massless logarithmic ones. This relies entirely on the property that in dimensional regularisation all scaleless integrals are zero. Since non-logarithmic integrals require a scale, which is absent, they must be zero. For LTVGs this statement must be understood as a cancellation of poles between IR and UV divergences. For a massless vacuum graph G the statement is thus that unless the vacuum graph is an LTVG, its UV counterterm must be set to zero. This has some far-reaching consequences: the UV counterterm of *any* generic Feynman graph can *always* be expressed as a counterterm of an LTVG. Consequently, the counterterm operation of many seemingly different graphs can easily be related to one another. In order to compute the counterterm of an LTVG, one can then always choose to compute it from a convenient single scale graph. For example, the UV counterterm of the following two-loop vacuum, propagator or vertex Feynman graphs are all identical, whether they include massive lines or not:



(179)

Such relations, when combined with rewriting dot products and a powerful graph canonicalisation algorithm that maps each isomorphic graph to the same graph, can be used to reduce the number of unique counterterms to a comparably small number, even at five loops.

The IR counterterm operation $\widetilde{\Delta}$ can be defined analogously by isolating the superficial IR divergence of the IR subgraph by subtracting from it all UV divergences as well as all IR subdivergences. For this concept to make sense it is useful to recall the definition of the contracted vacuum IR subgraph which we introduced in section 5.1.2. Given some IR subgraph G', and its associated contracted vacuum graph $\tilde{G}$, we define

$$\widetilde{\Delta}(G') = -K\underline{R}^* \tilde{G}\,, \tag{180}$$

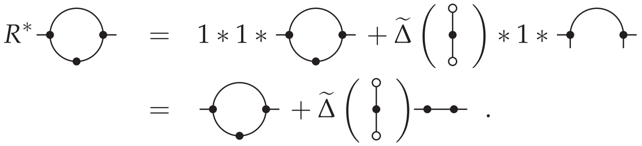
with the $\underline{R}^*$-operation defined recursively via

$$\underline{R}^* \tilde{G} = \sum_{\substack{S\in W(\tilde{G}), S'\in \underline{W}'(\tilde{G}) \\ S\cap S'=\emptyset}} \widetilde{\Delta}(S') * \Delta(S) * \tilde{G}/S \setminus S'\,. \tag{181}$$

The proper IR wood $\underline{W}(G)$ then includes all IR spinneys apart from the one which consists of the (vacuum) graph G itself. We remark that for Euclidean non-exceptional

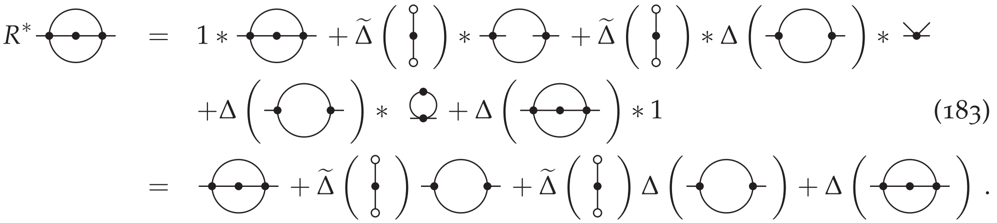
Feynman graphs only vacuum graphs can actually carry superficial IR divergences. For IR divergences with higher than logarithmic SDD, the internal momenta of the IR subgraphs γ' should be viewed as external to $G/S \setminus S'$ and $\widetilde{\Delta}(\gamma')$ is promoted to become a Taylor expansion operator acting on the remaining graph $G/S \setminus S'$. To understand the origin of this prescription, it is useful to review how the factorisation of the IR counterterm really comes to be. This will be discussed in more detail in the next section.

To familiarise the reader with the underlying simplicity of the procedure, which is easily lost in the formalism, let us for now give a few examples of the R^*-operation. The simplest example is given by a one-loop bubble with a dotted line:



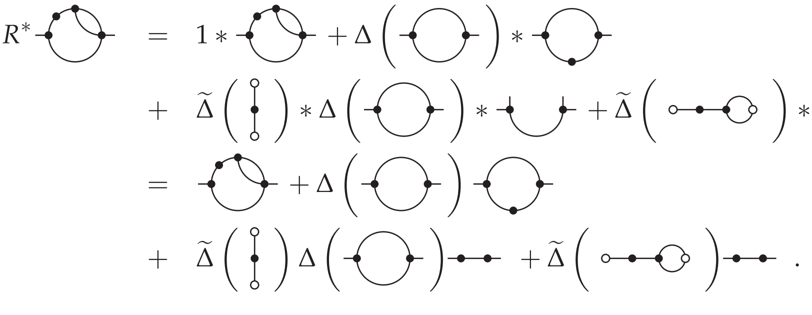
(182)

Here we use truncated lines to indicate that the *infrared propagators* enter as external lines into the vertices of the remaining graph. In the second line we have evaluated the $*$-product in the IR counterterm, which for logarithmic SDD simply results in setting the external IR legs to zero. We further used the trivial identity $1 * 1 * G = 1 * G = G$. We now present a slightly more complicated example at two loops:



(183)

This example illustrates the interplay of the subtraction of IR and UV divergences. Notice that one of the counterterms for the UV subdivergence vanishes as it contains a massless tadpole. Below is an example containing a two-loop IR subgraph:



(184)

5.4.2 *The infrared counterterm operation*

While our presentation of the IR counterterm will differ in parts from the literature [45, 133–135, 206, 207], it is nevertheless consistent with all accounts given there, at least when it comes to the evaluation of logarithmic IR divergences in scalar theories. Let us now study the singular behaviour of the integrand of the graph G in the limit where the momenta $k_1, \ldots, k_n$ which are contained in a given IR divergent subgraph γ' approach zero. In order to make the singular behaviour of G explicit, let us write

$$G(\{k_i\}) = \tilde{\gamma}(\{k_i\}) \cdot (G \setminus \gamma')(\{k_i\}) \,, \tag{185}$$

where we remind the reader that $\tilde{\gamma}$ is the contracted IR vacuum graph (see eq. 122). Even though the IR divergence of degree $\tilde{\omega}$ is entirely captured by the factor $\tilde{\gamma}$, the remaining graph $G \setminus \gamma'$ still depends on the momenta $\{k_i\}$ as external momenta, which flow into $G \setminus \gamma'$ through those vertices which connect it to γ'. Having made this dependence explicit, we now Taylor expand the remaining graph around $k_i = 0$ up to and including order $k_i^{\tilde{\omega}}$. The singular behaviour of G in the limit $k_i \to 0$ is then entirely captured by

$$G(\{k_i\}) = \sum_{r=0}^{\tilde{\omega}} \tilde{\gamma}(\{k_i\}) \cdot \mathcal{T}^{(r)}_{\{k_i\}}(G \setminus \gamma')(\{k_i\}) + \mathcal{O}(k_i^0) \,. \tag{186}$$

To build a counterterm for γ' we have to isolate its superficial IR divergence. We can accomplish this by introducing the operator $K\underline{R}^*$ which will now act not only on $\tilde{\gamma}_i$, but also on the polynomial k_i-dependent terms which are created by the Taylor expansion. Explicitly, this means that we can identify:

$$\begin{aligned} \widetilde{\Delta}(\gamma') * G \setminus \gamma' = - \sum_{\alpha_1 + .. + \alpha_n = \tilde{\omega}} \quad & K\underline{R}^* \left(\tilde{\gamma}(\{k_i\}) \prod_{i=1}^{n} k_i^{\mu_{i1}} \cdots k_i^{\mu_{i\alpha_i}} \right) \\ & \cdot \left[\left(\prod_{i=1}^{n} \frac{1}{\alpha_i!} \partial^{\mu_{i1}}_{k'_i} \cdots \partial^{\mu_{i\alpha_i}}_{k'_i} \right) (G \setminus \gamma')(\{k'_i\}) \right]_{k'_i = 0} \,, \end{aligned} \tag{187}$$

where we have dropped all orders in the Taylor expansion other than $\tilde{\omega}$, since these would give rise to scaleless vacuum graphs of non-logarithmic SDD which vanish under the operation $K\underline{R}^*$. Furthermore, we have introduced a set of dummy Lorentz indices μ_{ij} with $i \in \{1, .., n\}$ and $j \in \{1, .., \alpha_i\}$. In conclusion we can write:

$$\widetilde{\Delta}(\gamma') = -K\underline{R}^* \tilde{\gamma} \, \mathcal{T}^{(\tilde{\omega}(\gamma'))}_{\{k_i\}} \,. \tag{188}$$

We see that $\widetilde{\Delta}(\gamma')$ is promoted to a Taylor expansion operator which acts onto the remaining graph. This should be contrasted to the UV counterterm where the Taylor expansion operator acts on the UV subgraph and simply inserts the polynomial dependence of the counterterm into the remaining graph as a vertex. Thus whereas the UV counterterm is local in configuration space, the IR counterterm is local in momentum space. We remark that the IR counterterm is usually presented after integration as a sum over derivatives of the Dirac-delta function times Z-factors. Our representation is completely analogous to this representation, as one can easily show. However it further illuminates that the same Taylor expansion technique, which can be used to derive the UV counterterm in momentum space a la Zimmermann [194], can be used in complete analogy for the case of IR divergences.

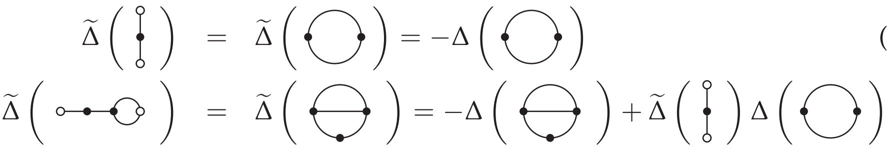
From eq.(187) we see that the Taylor expansion makes the IR subgraph logarithmic by multiplying $\tilde{\gamma}$ with monomials of IR momenta. In effect this procedure thus relates the IR counterterm operation of IR subgraphs of arbitrary SDD to IR counterterm operations of LTVGs. Let us denote such an LTVG by $\tilde{\gamma}_{\log}$. Although $\widetilde{\Delta}\tilde{\gamma}_{\log}$ is in principle a well defined operation, one never actually has to explicitly compute it. Instead, the value of $\widetilde{\Delta}(\tilde{\gamma}_{\log})$ can always be extracted recursively from the action of Δ on $\tilde{\gamma}_{\log}$ and its subdivergences [45, 133, 135]. The conversion between these two operations can be established from the equation

$$R^{*}\tilde{\gamma}_{\log} = 0\,, \tag{189}$$

which follows immediately from the fact that scaleless vacuum graphs vanish in dimensional regularisation. Expanding the left hand-side and rearranging we then obtain

$$\widetilde{\Delta}(\tilde{\gamma}_{\log}) = -\Delta(\tilde{\gamma}_{\log}) - \sum_{\substack{S\in \bar{W}_{\emptyset}(\tilde{\gamma}_{\log}), S'\in \underline{W}'_{\emptyset}(\tilde{\gamma}_{\log})\\ S\cap S'=\emptyset}} \widetilde{\Delta}(S') * \Delta(S) * \tilde{\gamma}_{\log}/S\setminus S' \tag{190}$$

where $\bar{W}_{\emptyset}$ and $\underline{W}_{\emptyset}$ are proper UV and IR -woods which exclude the empty graph. The sum over IR and UV spinneys in the above equation may be simplified further by imposing the requirement $S\cup S' = \{\tilde{\gamma}_{\log}\}$. Terms in the sum not satisfying this requirement would be proportional to scaleless vacuum graphs and hence vanish. Below are some simple examples for rewriting IR in terms of UV counterterms.



(191)

More examples, including IR counterterms with higher SDDs, will be given in the next section.

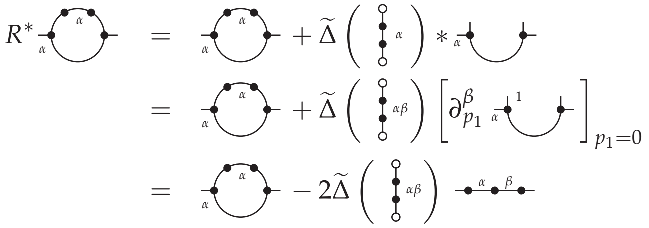
5.4.3 *Examples of R^* for generic Feynman graphs*

The methods which we presented in section 5.3 for dealing with arbitrary numerators and tensors within the R-operation can be applied in the same manner when dealing with IR divergences in the context of the R^*-operation. This follows mostly from the fact that the values of IR counterterms can be extracted from UV counterterms. However, subtleties arise when IR divergences of higher SDDs are encountered; in

5

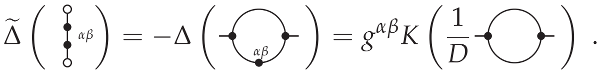
particular when they neighbour higher order UV divergences. In the following we will illustrate how the methods presented above are fully sufficient to tackle all of these cases.

Below we show an example of a diagram with a linear IR divergence:



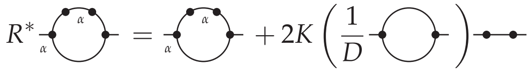
(192)

Here p_1 is used to denote the momentum of the IR leg flowing into the remaining diagram. In the second line we evaluated the linear order term of the $p_1 = 0$ Taylor expansion of the remaining graph. We see that this Taylor expansion leaves us with an IR counterterm of a rank 2 tensor. This term can be evaluated using eq.(190) and the tensor reduction method introduced in section 5.3.1:



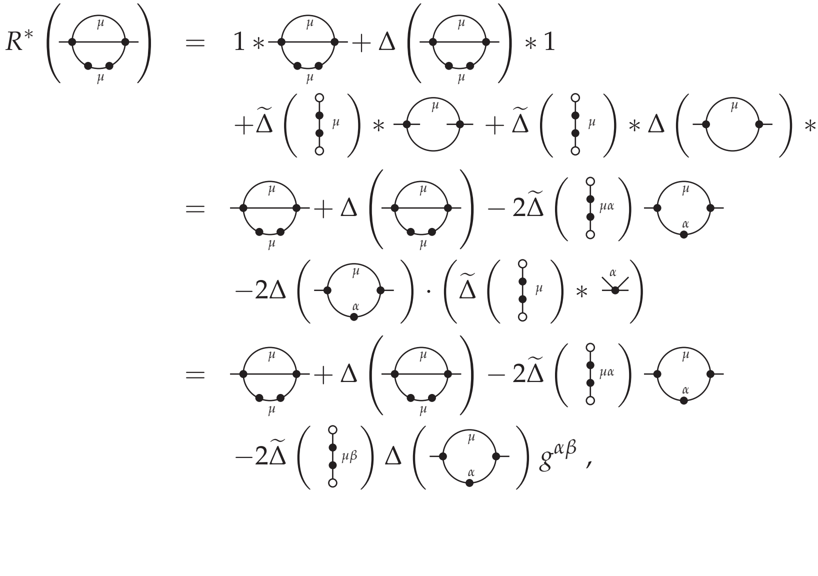
(193)

Inserting this expression back into eq.(192) we then obtain:



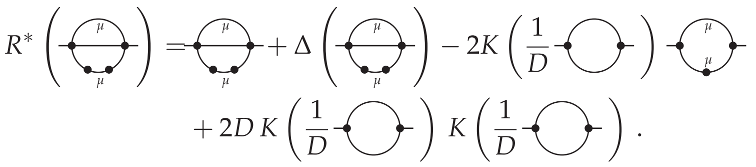
(194)

The insertion of higher order UV counterterms and higher order IR counterterms does not commute in general: it is crucial to first insert the UV counterterm and only then to apply the Taylor expansion corresponding to the IR counterterm. This is illustrated in the following example:



(195)

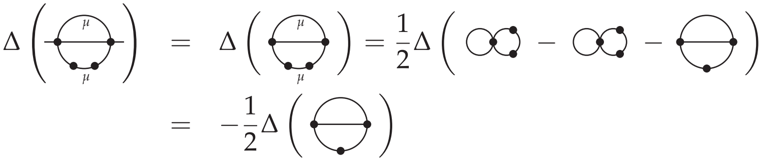
where the p^α created by the UV Taylor expansion was acted upon by the IR Taylor expansion. Substituting the results for the tensor reduced counterterms we then obtain



(196)

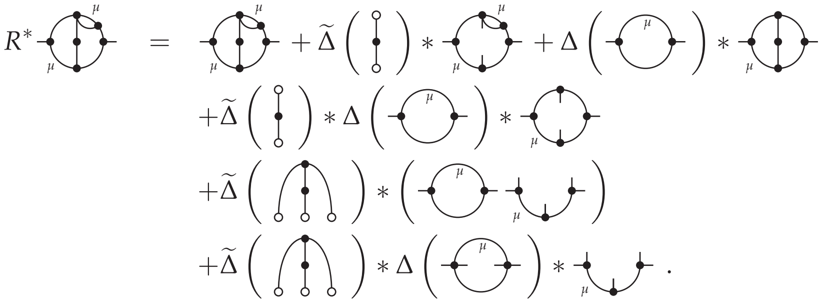
The factor of D is produced outside any K-operation and signifies the interplay between neighbouring higher order IR and UV divergences. The fact that the subtraction of IR and UV divergences does not commute in general was also discussed in [206].

We can drastically simplify the computation of the UV counterterm of the above diagram by dropping the external momenta, rewriting the dot product, and using symmetry:



(197)

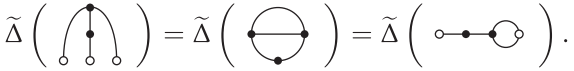
This shows how LTVGs can be conveniently used to simplify the calculation of counterterms. Let us now study



(198)

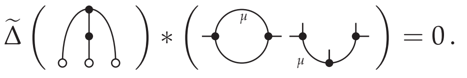
(199)

This example shows several interesting features. First, we consider the three-line IR subgraph which appears in the second and third line. By constructing its associated contracted vacuum graph we can relate its IR counterterm to one which we already computed in eq.(192):



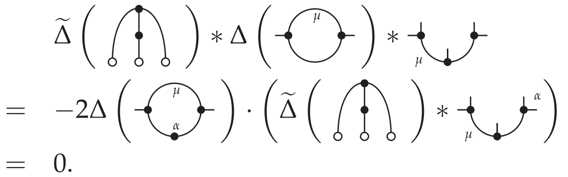
(200)

Second, we notice that this IR subgraph is disjoint in the original graph, a feature which was already discussed for a simpler example in section 5.1.2. As a result, the remaining graph splits into two disjoint components. Since one of the two components has no dependence on external momenta, it becomes scaleless when it is acted upon by the Taylor expansion operator, and as a result vanishes:



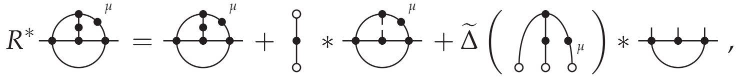
(201)

Third, we see that the counterterm in eq.(200) has a linear UV subgraph that will generate a momentum that belongs to the IR in the remaining diagram. In this case, the IR subgraph has logarithmic SDD, which means that after Taylor expanding the IR — setting all IR momenta to 0 — this counterterm will vanish:



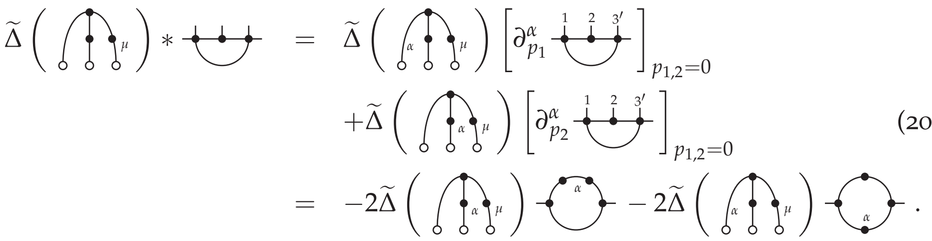
(202)

Thus we see that both counterterms containing this particular IR subgraph vanish, although for completely different reasons. We continue with another example:



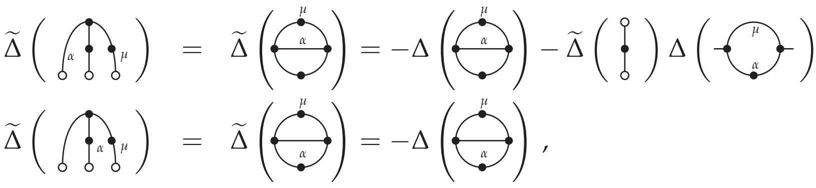
(203)

which shows an IR divergence of linear SDD with three IR legs entering the remaining graph. Here one can choose to Taylor expand around any independent set of IR subgraph momenta. Taking $p_3 = p_{3'} \equiv -p_1 - p_2$ we may Taylor expand around $p_1 = p_2 = 0$:



(204)

The IR counterterms are recursively converted to UV counterterms, i.e.,



(205)

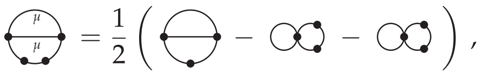
and then tensor reduced as discussed before.

5.4.4 *Properties of logarithmic vacuum graphs*

We will now summarise a few properties of counterterms of LTVGs. Even though the UV counterterm operation does not generally commute with contraction, it is additive under integrand relations:

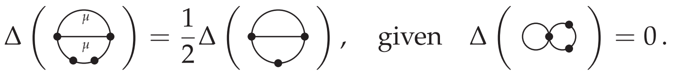
$$\Delta(G_1 + G_2) = \Delta(G_1) + \Delta(G_2)\,. \qquad (206)$$

In stark contrast, it appears that $\widetilde{\Delta}$ does not satisfy an analog of the additivity property in eq.(206). Consider for instance the graph:



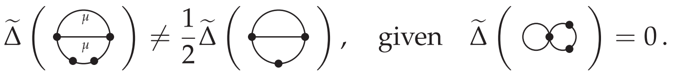
(207)

for which one can easily confirm confirm additivity:



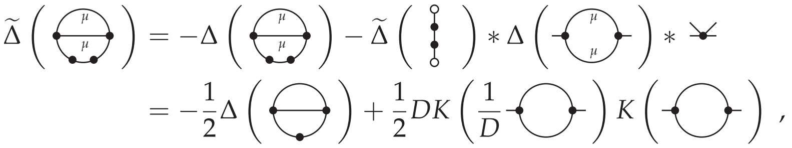
(208)

However the same is not true for the IR counterterm:



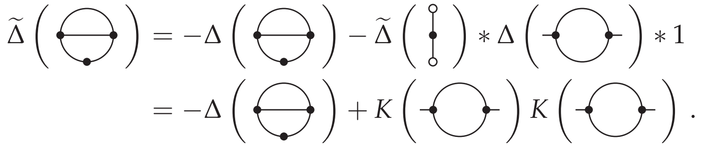
(209)

This can be verified by direct computation:


Δ̃ ( ) = −Δ ( ) − Δ̃ ( ) ∗ Δ ( ) ∗
μ
μ
= −1/2 Δ ( ) + 1/2 DK (1/D) K ( ) ,

(210)

while instead



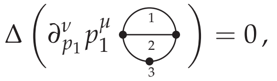
(211)

We note that the reason for this apparent disagreement stems from a non-cancellation of Ds in $K(\frac{1}{D}A)D \neq K(A)$. While a consistent scheme may exist where the IR counterterms could be chosen to be additive, this scheme would likely destroy some of the nice properties of the UV counterterm operation. One could imagine that such a scheme arises naturally if the R^*-operation was to be formulated in configuration space, where the roles of IR and UV are effectively interchanged. The absence of additivity of the IR counterterm may appear to be a hindrance in calculations, but it does not present a practical limitation since it is always possible to rewrite IR counterterms in terms of UV counterterms of LTVGs via eq.(190). In turn, the LTVGs can be simplified using the additivity property of eq.(206).

A second useful property of the UV counterterm operation is that it commutes with uncontracted differentiation operators. This also allows one to derive IBP-like relations:

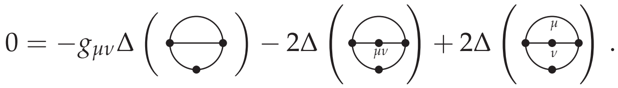
$$\Delta(\partial^{\mu}_{p_i} p^{\nu}_j \tilde{\gamma}(\{p_k\})) = 0\,, \tag{212}$$

where $\{p_i\}$ is a set of independent momenta spanning the LTVG $\tilde{\gamma}$. When applying eq.(212) to LTVGs of a certain tensorial rank, it returns relations among LTVGs with the tensorial rank raised by up to two. An example is given by the IBP


$\Delta \left(\partial^{\nu}_{p_1} p^{\mu}_1 \right) = 0\,,$
1
2
3

(213)

which yields the relation:

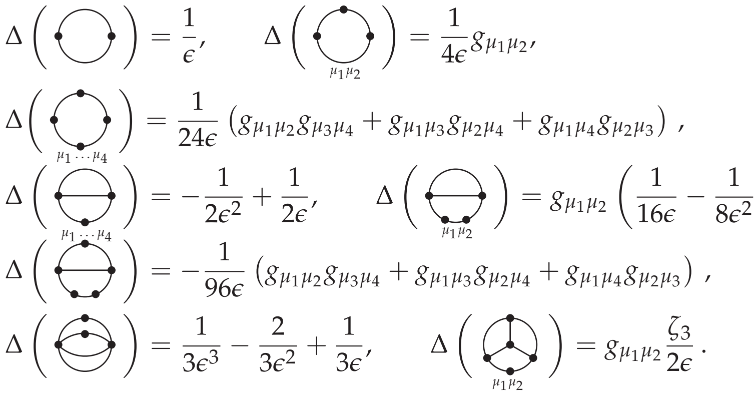

$0 = -g_{\mu\nu}\Delta\left(\right) - 2\Delta\left(\right) + 2\Delta\left(\right)\,.$
$\mu\nu$
μ
ν

(214)

IBP relations thus allow one to find relations between counterterms of LTVGs of different tensorial rank, which can be used to simplify calculations. Let us finish this section by giving examples for some common UV counterterms of LTVGs. We shall use the following normalisation for the integration measure of each independent loop momentum:

$$\mu^{2\epsilon} e^{\epsilon(\gamma_E + \zeta_2/2)} \int \frac{d^D k}{\pi^{D/2}}\,, \tag{215}$$

with γ_E the Euler-Mascheroni constant and ζ_n the Riemann zeta function.


$\Delta\left(\right) = \frac{1}{\epsilon},$
$\Delta\left(\right) = \frac{1}{4\epsilon} g_{\mu_1\mu_2},$
$\mu_1\mu_2$
$\Delta\left(\right) = \frac{1}{24\epsilon}\left(g_{\mu_1\mu_2}g_{\mu_3\mu_4} + g_{\mu_1\mu_3}g_{\mu_2\mu_4} + g_{\mu_1\mu_4}g_{\mu_2\mu_3}\right),$
$\mu_1 \cdots \mu_4$
$\Delta\left(\right) = -\frac{1}{2\epsilon^2} + \frac{1}{2\epsilon},$
$\Delta\left(\right) = g_{\mu_1\mu_2}\left(\frac{1}{16\epsilon} - \frac{1}{8\epsilon^2}\right),$
$\mu_1\mu_2$
$\Delta\left(\right) = -\frac{1}{96\epsilon}\left(g_{\mu_1\mu_2}g_{\mu_3\mu_4} + g_{\mu_1\mu_3}g_{\mu_2\mu_4} + g_{\mu_1\mu_4}g_{\mu_2\mu_3}\right),$
$\mu_1 \cdots \mu_4$
$\Delta\left(\right) = \frac{1}{3\epsilon^3} - \frac{2}{3\epsilon^2} + \frac{1}{3\epsilon},$
$\Delta\left(\right) = g_{\mu_1\mu_2}\frac{\zeta_3}{2\epsilon}\,.$
$\mu_1\mu_2$

(216)

5.5 applications of r*

In this section we shall demonstrate the R^* method introduced in the previous sections to compute the pole part of a number of complicated, and so far unknown, five loop propagator integrals. This is achieved through the relations introduced in section 5.2:

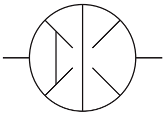
$$KG = -K\delta R^* G, \qquad \delta R^* = R^* - 1\,, \tag{217}$$

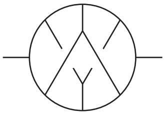
which allows one to compute the poles of any L-loop propagator integral from propagator integrals of maximally $L-1$ loops, a fact which follows straight forwardly from the structure of the R^*-operation.

We have implemented the algorithms described in this work in two independent computer codes. One is written entirely in Form [102] and the other is mostly written in Maple. We interfaced both of these implementations with the Forcer program [1, 7, 9], a highly efficient Form program that uses parametric integration-by-parts reduction rules to reduce any propagator up to four loops to a set of known master integrals. This makes it possible to compute all five-loop propagator integrals from the knowledge of up to four-loop integrals. The combination of the R^* algorithm presented in this work and the Forcer program was used in the recent computation of the five-loop beta function for Yang-Mills theory with fermions (as will be discussed in chapter 6) [4], which took three days on a cluster. The latest version of the code can perform the same calculation in six days on one 32-core machine. It is worth emphasising that our application of the R^*-operation differs in spirit not only from the global approach of [40], but also from the local approach which was taken for instance in [199–201] to compute the six loop anomalous dimensions in scalar ϕ^4 theory. Indeed, the local R^* approach taken there was based on the application of $K\bar{R}^*$, which allows one to directly isolate the renormalisation group constant of the relevant order. Our approach focuses on simply computing the poles of a given amplitude from which the corresponding renormalisation group constants can of course be extracted. In the case of non-scalar QFTs, such as QCD, there are advantages in our approach. While the UV counterterm operation $K\bar{R}^*$ is sensitive to the contraction anomalies, discussed extensively in section 5.3, the operation $K\delta R^*$ is insensitive to them. This observation allows us to make full use of integrand (and even integral) relations of generic Feynman graphs. In contrast the traditional local approach with $K\bar{R}^*$ would not easily allow for the use of such relations.

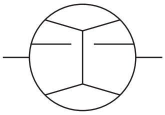
We shall start by presenting results for the poles of a number of five-loop integrals in ϕ^3-theory in four dimensions. While these integrals are in fact not superficially UV divergent, they do contain highly intricate IR divergences with high SDDs. Furthermore, some of these integrals are part of a yet to be found basis of five-loop master integrals, and we thus anticipate that the explicit results for their poles will constitute useful benchmarks for future evaluations of their finite parts. All results in the following will be normalised according to eq.(215), and we set $Q^2 = \mu^2 = 1$, where Q is the (Euclidean) external momentum.

Below we present the pole part of five ϕ^3 topologies:

K  $= -\dfrac{56}{\epsilon}\zeta_7$ (218)

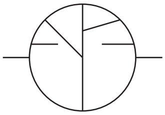
K 

$$= \frac{4}{\epsilon^2}\zeta_5 + \frac{1}{\epsilon}\left(10\zeta_6 + 4\zeta_3^2\right) \tag{219}$$

K 

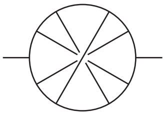
$$= -\frac{9}{10\epsilon^3}\zeta_3 + \frac{1}{\epsilon^2}\left(-\zeta_5 - \frac{27}{20}\zeta_4 - \frac{81}{20}\zeta_3\right) \tag{220}$$

$$+\frac{1}{\epsilon}\left(-\frac{5}{2}\zeta_6 + \frac{1}{5}\zeta_5 - \frac{243}{40}\zeta_4 + \frac{159}{8}\zeta_3 - 5\zeta_3^2\right)$$

K 

$$= \frac{1}{5\epsilon^3}\zeta_3 + \frac{1}{\epsilon^2}\left(\frac{3}{10}\zeta_4 + \frac{1}{2}\zeta_3\right) \tag{221}$$

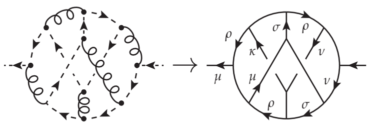
$$+\frac{1}{\epsilon}\left(-\frac{147}{40}\zeta_7 + \frac{11}{15}\zeta_5 + \frac{3}{4}\zeta_4 - \frac{641}{60}\zeta_3 + \frac{1}{5}\zeta_3^2\right)$$

K 

$$= \frac{6}{\epsilon^2}\zeta_5 + \frac{1}{\epsilon}\left(42\zeta_7 + 15\zeta_6 + 21\zeta_5 + 6\zeta_3^2\right) \tag{222}$$

The computation of the pole parts only takes a few seconds on a single core. A list of all top-level five-loop ϕ^3 topologies will be provided on `arXiv.org` as an attachment to this article.

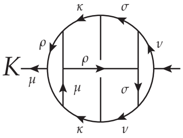
Next, we present the pole parts of some five-loop ghost propagator diagrams, where we enforce that the ghost line goes through all vertices. If we use the Feynman gauge, the QCD diagram has a one-to-one correspondence to a generic Feynman graph. This is illustrated below:



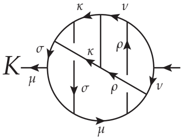
$$= \frac{1}{\epsilon^2}\left(\frac{11}{2560} - \frac{1}{64}\zeta_5 + \frac{3}{256}\zeta_3\right) + \frac{1}{\epsilon}\left(\frac{551}{5120} - \frac{5}{128}\zeta_6 - \frac{109}{256}\zeta_5 + \frac{9}{512}\zeta_4 + \frac{729}{2560}\zeta_3 + \frac{1}{32}\zeta_3^2\right) + \mathcal{O}(\epsilon^0) \tag{223}$$

Every Feynman diagram of this type has five dot products. As a result, there will be many tensor UV subgraphs. We have not rewritten the dot products to a basis, since this will create higher-order UV and IR divergences. Any speed gains from the simplified topologies are negated by expensive Taylor expansions and tensor reductions.

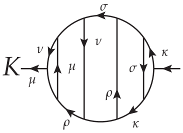
Below we present three more examples:



$$= \frac{1}{\epsilon^2}\left(\frac{1}{512} - \frac{1}{320}\zeta_3\right) + \frac{1}{\epsilon}\left(\frac{337}{5120} - \frac{161}{1280}\zeta_7 - \frac{25}{128}\zeta_5 - \frac{3}{640}\zeta_4 + \frac{341}{1280}\zeta_3 - \frac{9}{160}\zeta_3^2\right) \tag{224}$$

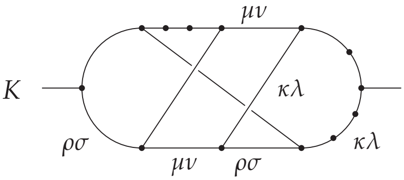


$$= \frac{1}{\epsilon^2}\left(\frac{1}{2560} + \frac{1}{64}\zeta_5 - \frac{3}{128}\zeta_3\right) + \frac{1}{\epsilon}\left(-\frac{17}{5120} + \frac{161}{640}\zeta_7 + \frac{5}{128}\zeta_6 + \frac{19}{128}\zeta_5 - \frac{9}{256}\zeta_4 - \frac{1291}{2560}\zeta_3 + \frac{1}{40}\zeta_3^2\right) \tag{225}$$



$$= \frac{1}{\epsilon^2}\left(-\frac{7}{5120} - \frac{1}{128}\zeta_5 + \frac{1}{128}\zeta_3\right) + \frac{1}{\epsilon}\left(\frac{73}{15360} + \frac{441}{2560}\zeta_7 - \frac{5}{256}\zeta_6 - \frac{29}{64}\zeta_5 + \frac{3}{256}\zeta_4 + \frac{951}{2560}\zeta_3 - \frac{71}{640}\zeta_3^2\right) \tag{226}$$

Finally, we present an example of a hard four-loop diagram:



$$= -\frac{13}{2304\epsilon^4} + \frac{1789}{55296\epsilon^3} + \frac{91757}{331776\epsilon^2} - \frac{17}{256\epsilon^2}\zeta_3 + \frac{1}{\epsilon}\left(\frac{199997}{248832} + \frac{5}{4}\zeta_5 - \frac{51}{512}\zeta_4 - \frac{17797}{13824}\zeta_3\right) \tag{227}$$

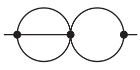
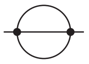
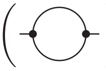
The diagram of eq. (227) has six dot products, one quadratic IR line, two quadratic UVs, and several logarithmic UV and IR subgraphs. The diagram requires 187 unique, scalarised counterterms to compute the diagram. It is interesting to compare the time required for computing the poles of this Feynman graph with the R^*-method with the time which a direct computation takes using the Forcer program. On a single core it takes 150 seconds to obtain the result with the R^*-operation. A direct computation with the Forcer program takes 675 seconds, 4.5 times as long. The reason why a direct computation is slower is because this particular integral requires reductions of seven master topologies at four loops, which are generally slow. In contrast, the R^*-operation only requires counterterms of up to three loops and thus avoids these complex reductions. This shows that sometimes it is beneficial to use R^*, even if a direct reduction is available.

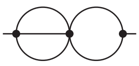
In this section we are going to discuss three differences of the R^*-method proposed in this work with those proposed in the literature, viz. application of the d'Alambertian versus Taylor expansion, treatment of the IR counterterm operation, and factorisation of the R^*-operation.

D'ALEMBERTIAN VERSUS TAYLOR EXPANSION In [206] it is proposed that the UV counterterm of a single scale quadratic integral can be computed from logarithmic ones by taking the d'Alembertian:

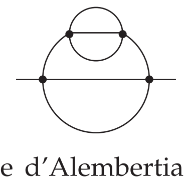
$$\Delta(G) = Q^2 \Delta\left(\frac{\Box}{2D} G\right), \qquad \Box = g^{\mu\nu} \frac{\partial}{\partial Q^\mu} \frac{\partial}{\partial Q^\nu} \,. \tag{228}$$

To evaluate the d'Alembertian acting on a Feynman graph, one must choose a path which routes the flow of the external momentum through the graph. Whereas the value of the integral $\Box G$ should be independent of the path taken, this is not necessarily true for the counterterm operation $\Delta(\frac{\Box}{2D}G)$ which as we discussed in section 5.3 is not generally invariant under contractions with $g_{\mu\nu}$. Below we give an example where this problem becomes apparent.

$\Delta\Bigg($  $\Bigg) = -\Delta\Bigg($  $\Bigg)\,\Delta\Bigg($  $\Bigg) = \frac{1}{4\epsilon^2}\,,$ (229)

$\Delta\Bigg(\frac{\Box}{2D}$  $\Bigg) = \frac{1}{4\epsilon^2} - \frac{1}{8\epsilon}\,.$ (230)

We can easily see from the cutvertex rule that the UV counterterm cannot have a $\frac{1}{\epsilon}$ pole. In most ϕ^4 topologies one may get the correct result by using the d'Alembertian, but this is in no way guaranteed. Consider for example the following diagram.



(231)

If one chooses to evaluate the d'Alembertian along a path through the sunrise subgraph on the top, a wrong answer is obtained. Instead, the only safe procedure for computing the UV counterterm of higher order UV or IR subgraphs, is to perform a Taylor expansion. That is, instead of the d'Alembertian the following differential operator should be used:

$$\frac{1}{2} Q_\mu Q_\nu \partial^\mu \partial^\nu \,, \tag{232}$$

which is guaranteed to commute with the counterterm operation.

ir counterterm operation The IR counterterm operation proposed in [45, 133, 134, 206–208] is usually formulated as

$$\widetilde{\Delta}(\gamma') = P_{\gamma'}(\{\partial_{p_i}\}) \prod_{k=1}^{l(\gamma')} (2\pi)^D \mu^{-2\epsilon} \delta^{(D)}(p_i)\,. \tag{233}$$

Here $l(\gamma')$ is the number of loops of the IR subgraph γ', $P_{\gamma'}(\{\partial_{p_i}\})$ is a homogeneous polynomial of degree $\tilde{\omega}(\gamma')$ in the differential operators $\{\partial_{p_1},\ldots,\partial_{p_l}\}$ and $\{p_1,\ldots,p_{l(\gamma')}\}$ is a set of independent momenta spanning the IR subgraph. The strategy to compute the IR counterterm in [45, 133, 134, 206–208] relies on setting up a system of equations, by inserting the IR subgraph γ' into several suitable graphs, and demanding the coefficients of $P'_\gamma(\{\partial_{p_i}\})$ to render this system finite after applying the R^*-operation. It is straightforward to show that our definition of the IR counterterm given in eq.(188) leads to a similar form to that of eq.(233). The main difference in our approach is that we directly compute the coefficients of $P'_\gamma(\{\partial_{p_i}\})$ from the values of tensor IR subgraphs of logarithmic SDD. The advantage of [45, 133, 134, 206–208] is that no extra tensor reduction due to the Taylor expansion has to be considered. Even though no such examples exist in the literature, this approach will require tensor reduction as well when applied to generic Feynman graphs. Since our entire setting relies on reducing both UV and IR counterterms to a common basis of LTVGs this is a small price to pay and allows a unified setting for the computation of the UV and IR counterterms, whose linear dependence is neatly expressed through eq.(190).

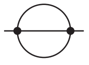
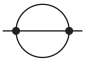
factorisation of the r*-operation A third point we wish to raise, concerns the factorisation of the R^*-operation into a pure UV subtraction R-operation and a pure IR subtraction operation $\tilde{R}$, as noted in, e.g., [135, 207]:

$$R^* = \tilde{R}\,R\,. \tag{234}$$

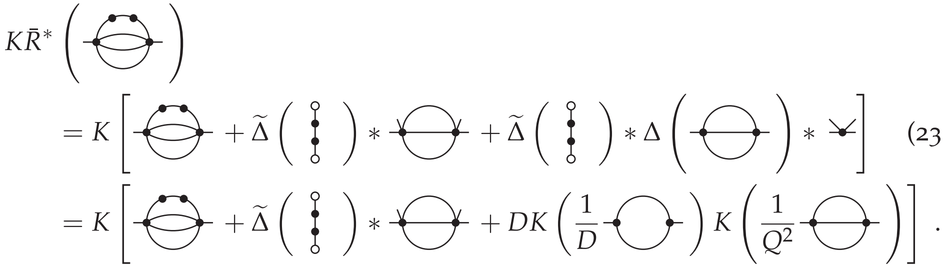
It has already been shown in [206, 207] that for local R^*, higher degree IR divergences the $\tilde{R}$ and R do not generally commute:

$$R^* \neq R\,\tilde{R}. \tag{235}$$

We wish to point out that even $R^* = \tilde{R}\,R$ cannot be naively applied to generic Feynman graphs and is explicitly broken in graphs where higher degree IR subgraphs neighbour higher degree UV subgraphs. To illustrate this, we use the same example as in [206] where it was used to show the non-commutativity of the UV and IR counterterm operators. Using the relation

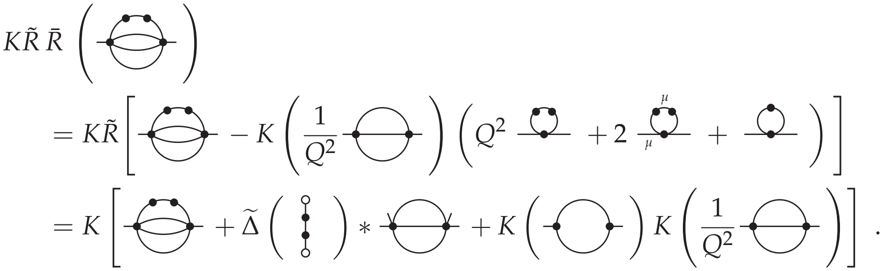
$\Delta\Bigg($  $\Bigg) = -K\Bigg(\frac{1}{Q^2}$  $\Bigg)\,Q^2,$ (236)

which is valid only because the subdivergences of this Feynman graph are vanishing, we get:



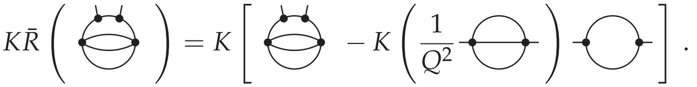
(237)

In contrast, the factorised approach yields



(238)

Thus we see that different results are obtained using the factorised or non-factorised approach. By IR rearrangement, we can rewrite the counterterm to be IR finite:



(239)

The result of eq.(239) agrees with eq.(237), as was shown in [206] as well, and thereby clearly falsifies the approach taken in eq.(238). This example further illustrates the importance of factors of D which are created in the interplay of higher degree IR and UV divergences.

5.7 CHAPTER CONCLUSION

We will now answer

> **RQ3**: *Can we compute the poles of five-loop diagrams using only four-loop diagrams more efficiently?*

The R^*-operation is a powerful tool to compute the poles of arbitrary Euclidean Feynman graphs with non-exceptional external momenta from simpler Feynman graphs. In this chapter we have extended the R^*-operation to Feynman graphs with arbitrary numerators and of arbitrary tensorial rank. Since the local R^*-operation had previously only been applied to scalar theories, we have vastly generalised its range of applicability.

The methods proposed in this chapter make full use of rewriting the counterterm operations for arbitrary divergent UV and IR subgraphs in terms of scaleless tensor vacuum graphs of logarithmic superficial degree of divergence, which we called LTVGs. This concept, which to the best of our knowledge has not previously been employed to this extent, allows one to take advantage of the enhanced symmetry properties of vacuum graphs. We analysed contraction anomalies, which are easily traceable within dimensional regularisation, and provided a consistent scheme that uses LTVGs as basic building blocks for UV and IR counterterms. Additionally, we have refined the definition and evaluation of the IR counterterm operation, so that it resembles its UV counterpart.

5.7.1 *Findings and main conclusion*

Our two main contributions are (1) the generalisation of the R^*-method and (2) its implementation in an efficient computer code. Additionally, we have verified the correctness of our method and code by computing the five loop beta function of QCD with an arbitrary simple gauge group [4] (see chapter 6). In this chapter we provided results for the poles of all five-loop top-level propagator graphs in ϕ^3 theory in four dimensions, as well as several explicit results for five-loop ghost propagator graphs with highly non-trivial numerator structures. From the results we derive our main conclusion: the method presented in this chapter provides an efficient way to compute the poles of complicated Feynman diagrams, even at five loops.

5.7.2 *Future research*

We see three lines of future research. First, we envisage that these results will provide useful cross-checks once analytic computations, which also contain the finite parts of these Feynman graphs, become available.

Second, our R^* method may shed some further light on an old puzzle related to the absence of certain higher zeta values in the anomalous dimensions, such as the beta function in QCD. While some explanations for this phenomenon have been given in [105], we believe that the many relations among LTVGs should allow to further illuminate the origin of the absence of these zeta-values, as they can clearly be traced to the UV counterterms of only a handful of LTVGs.

Third, among many other applications, these techniques could be used to compute Mellin moments of splitting functions at five loops, a problem which is currently well out of reach with any other method known to us.

6 THE FIVE-LOOP BETA FUNCTION

In this chapter we report on the first computation of the five loop beta function for a general Yang-Mills theory with one set of fermions. This is only possible by combining the results of **RQ2**, the Forcer program for four-loop calculations described in chapter 3, and **RQ3**, the R^*-operation described in chapter 5.

The beta function governs how the strength of the strong interaction scales with the energy. The determination of the (sign of the) leading one-loop coefficient β_0 [23, 24], soon followed by the calculation of the two-loop correction β_1 [119, 210] shows that the strong interaction vanishes at large energies (or very short distances). This *asymptotic freedom* means that QCD is a viable theory for the strong interaction. The discovery of asymptotic freedom was awarded the Nobel Prize for Physics in 2004.

The scale dependence ('running') of the renormalised coupling constant α_i can be written in perturbation theory as

$$\frac{da}{d\ln\mu^2} \;=\; \beta(a) \;=\; -\sum_{n=0}^{\infty} \beta_n\, a^{n+2}\;, \quad a \;=\; \frac{\alpha_\mathrm{i}(\mu)}{4\pi} \tag{240}$$

where μ is the renormalisation scale.

The renormalisation-scheme dependent three-loop (next-to-next-to-leading order, N^{2}LO) and four-loop (next-to-next-to-next-to-leading order, N^{3}LO) coefficients β_2 and β_3 were computed in refs. [122, 211] and [113, 114] in minimal subtraction schemes [130, 131] of dimensional regularisation [179, 180].

Precise determination of the beta function is important for all renormalisation group improved perturbation theory calculations. Below we mention two important use-cases. First, in the past years, N^{2}LO accuracy has been reached for many processes at high-energy colliders. N^{3}LO corrections have been determined for structure functions in inclusive deep-inelastic scattering (DIS) [98, 212] and for the total cross section for Higgs-boson production at hadron colliders [82, 155]. Second, we have computed moments of coefficient functions for DIS at N^{4}LO [8]. Obtaining full results at this order would virtually remove the uncertainty due to the truncation of the series of massless perturbative QCD in determinations of the strong coupling constant α_s from the scaling violations of structure functions in DIS.

The corresponding five-loop contributions to the beta functions of QCD, with all colour factors 'hard-wired', and QED have already been computed in refs. [40, 213]. Their leading large-n_f contributions have long been known [214], and the sub-leading large-n_f terms have been checked and generalized to a general simple gauge group in ref. [128]. The real tour de force of ref. [40] though, are the parts proportional to n_f^0, n_f^1 and n_f^2 which together required more than a year of computations on 20 multi-core workstations in a highly non-trivial theoretical framework. These critical parts have neither been extended to a general gauge group nor validated by a second independent calculation.

In the following chapter we address this issue and present the five-loop beta function for a general simple compact gauge group. Unlike the calculations in refs. [23, 24, 113, 114, 119, 120, 122, 210, 211], we have employed the background field method (see section 6.2), and the R^* method (see chapter 5).

Finally, we transform the five-loop Yang-Mills beta function from $\overline{\text{MS}}$ to the MiniMOM scheme [153]. The MiniMOM scheme is more convenient than $\overline{\text{MS}}$ for analysis in the non-perturbative regime of QCD.

The remainder of this chapter is structured as follows. We first explore five optimisations in section 6.1, after which we define the background field in section 6.2. Next, we present the computation of the five-loop beta function for Yang-Mills theory with fermions in section 6.3. We discuss the results in section 6.4. In section 6.5, we transform the five-loop beta function from $\overline{\text{MS}}$ to the MiniMOM scheme. Finally, we present the chapter conclusions in section 6.6.

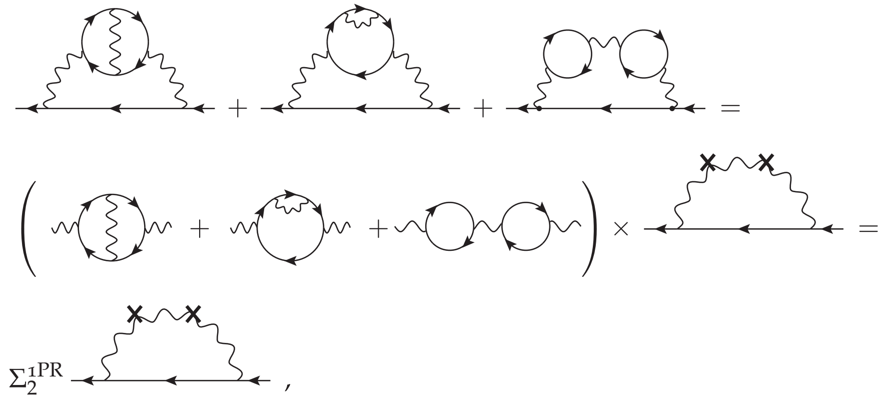
6.1 OPTIMISATIONS

Performing computations at five loops introduces at least five new bottlenecks compared to four loops. (1) The number of diagrams and their complexity grow exponentially. (2) The substitution of the Feynman rules is slow and creates millions of term. (3) The Taylor expansion to make the diagrams logarithmic creates many terms. (4) The number of counterterms grows exponentially. (5) Tensors of rank 10 have to be reduced, which involves solving large systems.

In this section we address these issues by presenting five optimisations, namely improved treatment of propagator insertions in section 6.1.1, delayed Feynman rule substitution in section 6.1.2, improved rules to make diagrams logarithmic in section 6.1.3, a canonical form algorithm for Feynman diagrams in section 6.1.4, and an efficient tensor reduction algorithm in section 6.1.5.

6.1.1 *Treatment of propagator insertions*

Many of the higher-loop corrections are self-energies of propagators in the diagram. Due to the local nature of the Feynman rules, these self-energies only depend on their external momentum (there are no contractions with other parts of the larger diagram), so they can be 'factorised' out (see also 3.10):

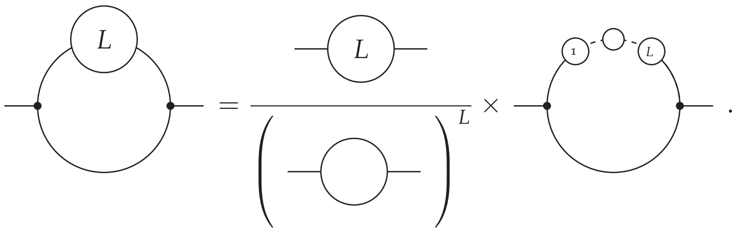


(241)

where the L-loop self-energy is replaced by $(p^2)^{-\varepsilon L}$ in the larger diagram (marked by L crosses). In a sense, the subdiagram is integrated out. The resulting simpler topology is multiplied by the one-particle-reducible L-loop self-energy. Since the L-loop subdiagram is of lower order, these quantities have already been computed and can easily be tabulated to prevent recomputations. For example, a five-loop diagram may contain the expensive 4-loop gluon propagator as a subdiagram.

For the R^*-operation, this representation has an issue: the non-integer power hides UV-divergent subdiagrams, which should be subtracted. However, since the exact contents of the $(p^2)^{-\varepsilon L}$ is factorised out, we may replace it with *any* L-loop subdiagram. Therefore we choose the simplest configuration: L scalar one-loop bubbles side by side.

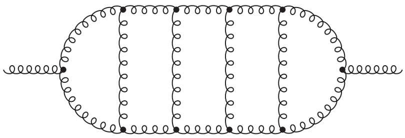
Thus, for the R^*-operation we can remove propagator insertions by using the following relation:



(242)

6.1.2 *Delayed Feynman rule substitution*

Substituting the Feynman rules creates many terms. For example, the following fully gluonic five-loop graph creates 12 029 521 scalar integrals in the Feynman gauge:



. (243)

The source of the blow-up is the Feynman rule for the triple gluon vertex, which can be written in the following way:

$$v_{3g}(p_1^{\mu,a}, p_2^{\nu,b}, p_3^{\rho,c}) = -if^{abc}\left[(p_1-p_2)^\rho g_{\mu\nu} + (2p_2+p_1)^\mu g_{\nu\rho} + (-2p_1-p_2)^\nu g_{\mu\rho}\right] . \tag{244}$$

Thus, for every vertex, six terms are created, of which some will evaluate to the same expression due to symmetries. For all these terms, expensive operations such as Taylor expansions and divergent subgraph recognitions have to be performed. However, these operations only depend on the momentum powers and are invariant under the way the momenta contract. So, we rewrite the triple gluon vertex in a way that exposes the momenta, but keeps all the contractions unsubstituted:

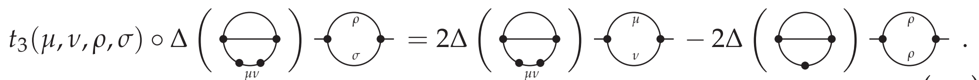
$$v_{3g}(p_1^{\mu,a}, p_2^{\nu,b}, p_3^{\rho,c}) = -if^{abc}p_1^\sigma t_3(\sigma,\nu,\rho,\mu) + if^{abc}p_2^\sigma t_3(\sigma,\mu,\rho,\nu) , \tag{245}$$

where

$$t_3(\mu,\nu,\rho,\sigma) = g_{\mu\rho}g_{\nu\sigma} + g_{\mu\sigma}g_{\nu\rho} - 2g_{\mu\nu}g_{\rho\sigma} . \tag{246}$$

After rewriting v_{3g} in terms of t_3, there are only $2^{10} = 1024$ terms for the Feynman diagram in eq. (243). We can keep our input in this compactified notation for as long as the actual contractions are not important, which is right until the tensor reduction.

We define the operation $\circ$ that applies the remaining Feynman rules to all components of the R^*-operation. For example:



(247)

We stress that for this particular case contraction is *necessary*.

Similar rules can be devised for the other vertices and for the trace of gamma matrices. At five loops, the substitution of t_3 and similar structures is an expensive part of the calculation, since the number of generated terms is high.

6.1.3 *Rules to make Feynman diagrams logarithmic*

Since we compute

$$K(G) = -K\delta R^* G , \tag{248}$$

we can make G logarithmic, by d'Alembertian-like derivatives inside $K(G)$. The advantage over just Taylor expanding $\Delta(G)$, is that we are allowed to contract inside $K(G)$. We first consider the following linear integral F:

$$F = Q^\mu F^\mu, \qquad F^\mu = Q^\mu (Q^2)^{-\epsilon L} f(\epsilon) \ . \tag{249}$$

We take the following derivative and solve the new equation for F:

$$\begin{aligned}
\partial_\mu F^\mu &= (1-2\epsilon L)(Q^2)^{-\epsilon L} f(\epsilon) \\
f(\epsilon) &= \frac{1}{1-2\epsilon L}(Q^2)^{\epsilon L}\partial_\mu F^\mu \\
F &= \frac{Q^2}{1-2\epsilon L}\partial_\mu F^\mu \ .
\end{aligned} \tag{250}$$

Thus, we have re-expressed F into a logarithmic integral with a new dot product that is internal to the graph. For the R^*-operation, this is generally better than a dot product with Q, since a dot product with the external momentum always requires a projection whereas internal momenta may be in the same divergent subdiagram.

For quadratic integrals F, we can apply the d'Alembertian to achieve the same effect:

$$\begin{aligned}
F &= (Q^2)^{1-\epsilon L} f(\epsilon) \\
\Box F &= 4(\epsilon L - 1)(\epsilon(L+1)-2)(Q^2)^{1-\epsilon L} f(\epsilon) \\
F &= \frac{Q^2}{4(\epsilon L - 1)(\epsilon(L+1)-2)}\Box F \ .
\end{aligned} \tag{251}$$

For quadratic integrals of the form $Q^\mu Q^\nu F^{\mu\nu}$, we can derive a special rule as well:

$$F = \frac{\partial_\mu \partial_\nu F^{\mu\nu}(2-2\epsilon L) + \partial_\alpha \partial^\alpha F^{\mu\mu}}{-8(-1+\epsilon L)(-3+\epsilon+\epsilon L)(-2+\epsilon+\epsilon L)} \tag{252}$$

For integrals with more Qs, or higher than quadratic ones, there are not sufficient ways available to contract existing vectors to solve the system. To make these integrals logarithmic, one could use Euler's homogeneous function theorem:

$$Q^\alpha \partial_\alpha F = nF \ , \tag{253}$$

where n is the order in Q. For a Feynman integral of degree S we have $n = S - 2\epsilon L$. Thus we formulate:

$$F = \left(\frac{1}{S-2\epsilon L} Q^\alpha \partial_\alpha\right)^S F \tag{254}$$

Every time a derivative is taken, another Q-path can be chosen through the diagram, to limit the growth of the number of terms. However, since the shape of eq. (254) is similar to a Taylor expansion, it is faster to only perform the Taylor expansion on $\Delta(G)$.

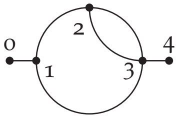
It is important to construct the counterterms of the rearranged G' at the same time as the subdivergences of G. Since many counterterms of G are also in G', the number of relevant counterterms is reduced. This saves a factor three at four loops.

6.1.4 *Canonical forms for Feynman diagrams*

The R^*-operation applied to five-loop diagrams will create many counterterms. In order to reduce computation time, it is important to compute the counterterms of a specific graph only once. In turn, this requires an efficient way to detect if two graphs are equal. One straightforward option is to keep a list of all the graphs that have already been processed and test for isomorphisms on every element of the list until one is found. If no match is found, the current graph can be added to the list. The two downsides of this method are that (1) an isomorphism test can be rather slow at five loops and (2) that the list of topologies grows rapidly.

A better alternative is to construct a *canonical form* of a graph. A canonical form is an isomorphism of the graph that is designated as the smallest by some yet to be defined measure. To test for equality, one can simply compare the canonical forms. Since isomorphy is first and foremost a property of the vertices, we give each vertex a label from 1 to n. For simplicity, let us consider a graph that has no dot products and only has edges with power 1.

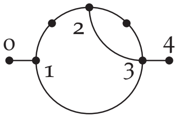
We convert our graph to an edge representation:



$$= e(0,1)e(1,2)e(2,3)e(2,3)e(1,3)e(3,4) \; . \tag{255}$$

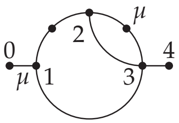
Here, $e(n_1, n_2)$ is the edge function, in which we place the smallest vertex index as the first argument. The *edge list* is a lexicographically sorted list of edge functions, as is shown in eq. (255). Now we define the smallest isomorphism of a graph as the vertex labelling for which the edge list is lexicographically smallest.[1]

We can easily extend the graph notation to a graph where propagators can have different powers, by introducing a third argument to the edge function e:



$$= e(0,1,1)e(1,2,2)e(2,3,1)e(2,3,2)e(1,3,1)e(3,4,1) \; , \tag{256}$$

where we again make sure that the first two arguments of $e(n_1, n_2, \ldots)$ are sorted. To add support for dot products and tensors, we extend the edge function even further:



$$= e(0,1,1,\mu)e(1,2,2)e(2,3,1)e(2,3,2,\mu)e(1,3,1)e(3,4,1) \; . \tag{257}$$

[1] In our program, we use the internal (deterministic) sorting order of Form to determine the smallest isomorphism instead.

We define the canonical signs of the momenta such that they always flow from the smallest vertex label to the highest. If a transformation changes the order, we flip the sign if the number of vectors in the momentum is odd:

$$e(2,1,n,\mu_1,\ldots,\mu_k) = (-1)^k e(1,2,n,\mu_1,\ldots,\mu_k)\,. \tag{258}$$

Finally, the momentum label p_i of each edge is uniquely defined by the position i of the edge in the edge list.

Now that most properties of the Feynman integral are captured in the extended edge list and we have defined which edge list is smallest, we use McKay's canonicalisation algorithm [215] to efficiently rewrite the complete Feynman integral to canonical form. A simplified version of this algorithm is implemented in Form code.

6.1.5 *Efficient tensor reduction*

It can be shown that the tensor reduction of ultraviolet and infrared subtraction terms, required for the R^*-operation, is equivalent to the tensor reduction of tensor vacuum bubble integrals. In general tensor vacuum integrals can be reduced to linear combinations of products of metric tensors $g^{\mu\nu}$ whose coefficients are scalar vacuum integrals. Specifically a rank r tensor, $T^{\mu_1\cdots\mu_r}$, is written as a linear combination of $n = r!/2^{(r/2)}/(r/2)!$ combinations of $(r/2)$ metric tensors with coefficients c_σ, i.e.,

$$T^{\mu_1\cdots\mu_r} = \sum_{\sigma\in{}_2S_r} c_\sigma\, T_\sigma^{\mu_1\ldots\mu_r}\,, \qquad T_\sigma^{\mu_1\cdots\mu_r} = g^{\mu_{\sigma(1)}\mu_{\sigma(2)}}\cdots g^{\mu_{\sigma(r-1)}\mu_{\sigma(r)}}\,. \tag{259}$$

Here we define ${}_2S_r$ as the group of permutations which do *not* leave the tensor $T_\sigma^{\mu_1\cdots\mu_r}$ invariant. The coefficients c_σ can be obtained by acting onto the tensor $T^{\mu_1\cdots\mu_r}$ with certain projectors $P_\sigma^{\mu_1\ldots\mu_r}$, such that

$$c_\sigma = P_\sigma^{\mu_1\cdots\mu_r} T_{\mu_1\ldots\mu_r}\,. \tag{260}$$

From this it follows that the orthogonality relation,

$$P_\sigma^{\mu_1\cdots\mu_r} T_{\tau,\mu_1\ldots\mu_r} = \delta_{\sigma\tau}\,, \tag{261}$$

must hold, where δ is the Kronecker-delta. Since the projector $P_\sigma^{\mu_1\cdots\mu_r}$ of each tensor can also be written in terms of a linear combination of products of metric tensors, inverting an $n\times n$ matrix determines all the projectors. However, there are two issues. The first is that the size of the matrix grows rather rapidly as r increases. Instead of solving an $n\times n$ linear system, the symmetry group of the metric tensors can be utilised to reduce the size of the system. From eq. (261) it follows that the projector P_σ is in the same symmetry group (the group of permutations which leave it invariant) as T_σ. For example, given a permutation $\sigma_1 = (123...(r-1)r)$,

$$T_{\sigma_1}^{\mu_1\cdots\mu_r} = g^{\mu_1\mu_2}g^{\mu_3\mu_4}\cdots g^{\mu_{r-1}\mu_r}\,. \tag{262}$$

The corresponding projector $P_{\sigma_1}^{\mu_1\cdots\mu_r}$ must be symmetric under interchanges of indices such as $\mu_1 \leftrightarrow \mu_2$, $(\mu_1,\mu_2)\leftrightarrow(\mu_3,\mu_4)$ and so on. Grouping the metric tensors by the symmetry leads to the fact that P_σ is actually written in a linear combination of a small number of m tensors instead of n ($m \leq n$),

$$P_\sigma^{\mu_1 \cdots \mu_r} = \sum_{k=1}^{m} b_k \sum_{\tau \in A_m^\sigma} T_\tau^{\mu_1 \cdots \mu_r} . \tag{263}$$

The set of groups $\{A_k^\sigma | k = 1..m\}$ must therefore each be closed under the permutations which leaves T_σ invariant and at the same time their union must cover once the group ${}_2S_n$. Contracting P_σ with a representative in each group gives an $m \times m$ matrix which can be inverted to yield the coefficients b_k. The number of unknowns m is $m = 5$ for $r = 8$ and $m = 22$ for $r = 16$, whereas we have $n = 105$ for $r = 8$ and $n = 2027025$ for $r = 16$. The comparison of these numbers illustrates that the exploitation of the symmetry of the projectors makes it possible to find the tensor reduction even for very large values of r, which could never have been obtained by solving the $n \times n$ matrix.

The second issue with tensors of high rank is the large number of intermediate terms that are created. Even though the system for the projector can be solved efficiently, $\mathcal{O}(n^2)$ terms will be created, of which some will merge due to symmetry. Let us consider rank 6, with 15 terms:

$$c_1 g^{\mu_1\mu_2} g^{\mu_3\mu_4} g^{\mu_5\mu_6} + c_2 g^{\mu_1\mu_3} g^{\mu_2\mu_4} g^{\mu_5\mu_6} + \ldots . \tag{264}$$

In most practical situations there is symmetry, both on the inside of the object that will be projected as on the outside. For example

$$A(p_1^{\mu_1} p_1^{\mu_2} p_1^{\mu_3} p_1^{\mu_4} p_2^{\mu_5} p_2^{\mu_6}) p_3^{\mu_1} p_3^{\mu_2} p_4^{\mu_3} p_4^{\mu_4} p_4^{\mu_5} p_4^{\mu_6} \tag{265}$$

is symmetric in exchanges of $\mu_1, \ldots, \mu_4$ and μ_5, μ_6 inside A, and is symmetric in μ_1, μ_2 and $\mu_3, \ldots, \mu_6$ outside A. The symmetry inside the object A will enforce that coefficient c_1 and c_2 (and others) will actually be the same. The symmetry on the outside will cause terms to merge. In the end, we could have used the symmetrised variant of eq. (264) instead:

$$c_1 \cdot (g^{\mu_1\mu_2} g^{\mu_3\mu_4} g^{\mu_5\mu_6} + 2 g^{\mu_1\mu_3} g^{\mu_2\mu_4} g^{\mu_5\mu_6}) + c_3 (2 g^{\mu_1\mu_2} g^{\mu_3\mu_5} g^{\mu_4\mu_6} + 10 g^{\mu_1\mu_5} g^{\mu_2\mu_6} g^{\mu_3\mu_4}). \tag{266}$$

We see that only two coefficients have to be computed instead of 15 and that there are only 4 terms in the output instead of 15. The challenge is to prevent these terms from being created in the first place by exploiting symmetry, instead of starting from eq. (264). We make use of the optimised FORM command `dd_`, which creates the tensor structure $T^{\mu_1, \ldots, \mu_r}$ without generating duplicates. If we evaluate `dd_(p1,p1,p1,p1,p2,p2)` and strip the coefficient we get `p1.p1^2*p2.p2+p1.p1*p1.p2^2`. These two terms represent the structure outside of c_1 and c_3 in (266). For each of these two terms, we solve for the coefficient. Next, we recreate the metric structures that would give this specific contraction.

A term generated by `dd_` consists of two different factors: $(p \cdot p)^a$ and $(p_1 \cdot p_2)^a$. For $(p \cdot p)^a$, we collect all possible indices involved with p. For eq. (265), this would be $\mu_1, \ldots, \mu_4$. Then we select all possible ways to get $2a$ elements from that list with `distrib_`. Next, we use $dd\_$ on those indices. Thus, for $p_1 \cdot p_1$ in the example we would get $g^{\mu_1\mu_2} + g^{\mu_1\mu_3} + g^{\mu_2\mu_3}$. For cases such as $(p_1 \cdot p_2)^a$, we select a from the list of indices associated to p_1 and a from the list of p_2. Then we permute over the list of p_2. Using this algorithm, one can generate all possible contractions from the result without generating duplicates. To apply the outside symmetry, one can easily fill in the outside momenta associated to the indices instead of the indices themselves. `distrib_` and `dd_` will take the symmetry into account automatically.

6.2 THE BACKGROUND FIELD

A convenient and efficient method to extract the Yang-Mills beta function is to make use of the background field. We will briefly review this formalism. We start with the Lagrangian of Yang-Mills theory coupled to fermions in a non-trivial (often the fundamental) representation of the gauge group, the theory for which we will present the five-loop beta-function in section 6.3.

The Lagrangian of this theory can be decomposed as

$$\mathcal{L}_{\text{YM+FER}} = \mathcal{L}_{\text{CYM}} + \mathcal{L}_{\text{GF}} + \mathcal{L}_{\text{FPG}} + \mathcal{L}_{\text{FER}} \,. \tag{267}$$

Here the classical Yang-Mills Lagrangian (CYM), a gauge-fixing term (GF), the Faddeev-Popov ghost term (FPG) and the fermion term (FER) are given by

$$\begin{aligned}
\mathcal{L}_{\text{CYM}} &= -\frac{1}{4} F^a_{\mu\nu}(A) F^{\mu\nu}_a(A) \,, \\
\mathcal{L}_{\text{GF}} &= -\frac{1}{2\xi} (G^a)^2 \,, \\
\mathcal{L}_{\text{FPG}} &= -\eta^\dagger_a \, \partial^\mu D^{ab}_\mu(A) \, \eta_b \,, \\
\mathcal{L}_{\text{FER}} &= \sum_{i,j,f} \bar{\psi}_{if} (i \not{D}_{ij}(A) - m_f \delta_{ij}) \, \psi_{jf} \,.
\end{aligned} \tag{268}$$

In the fermion term the sum goes over colours i, j, and n_f flavours f, and we use the standard Feynman-slash notation. The field strength is given by

$$F^a_{\mu\nu}(A) = \partial_\mu A^a_\nu - \partial_\nu A^a_\mu + g f^{abc} A^b_\mu A^c_\nu \tag{269}$$

and the covariant derivatives are defined as

$$\begin{aligned}
D^{ab}_\mu(A) &= \delta^{ab}\partial_\mu - g f^{abc} A^c_\mu \,, \\
D^\mu_{ij}(A) &= \delta_{ij}\partial^\mu - ig \, T^a_{ij} A^\mu_a \,.
\end{aligned} \tag{270}$$

The conventions associated to the generators T^a and structure constants f^{abc} of the gauge group will be explained in section 4.1. The gauge-fixing term depends on making a suitable choice for G^a, which is usually taken as $G^a = \partial^\mu A^a_\mu$.

The background-field Lagrangian is derived by decomposing the gauge field as

$$A^a_\mu(x) \;=\; B^a_\mu(x) + \hat{A}^a_\mu(x)\,, \tag{271}$$

where $B^a_\mu(x)$ is the *classical* background field while $\hat{A}^a_\mu(x)$ contains the *quantum* degrees of freedom of the gauge field $A^a_\mu(x)$. The background-field Lagrangian is then written as

$$\mathcal{L}_{\mathrm{BYM+FER}} \;=\; \mathcal{L}_{\mathrm{BCYM}} + \mathcal{L}_{\mathrm{BGF}} + \mathcal{L}_{\mathrm{BFPG}} + \mathcal{L}_{\mathrm{BFER}}\,. \tag{272}$$

$\mathcal{L}_{\mathrm{BCYM}}$ and $\mathcal{L}_{\mathrm{BFER}}$ are derived simply by substituting eq. (271) into the corresponding terms in the Yang-Mills Lagrangian. However a clever choice exists [117, 118] for the ghost and gauge fixing terms, which allows this Lagrangian to maintain explicit gauge invariance for the background field $B^a_\mu(x)$, while fixing only the gauge freedom of the quantum field $\hat{A}^a_\mu(x)$. The gauge fixing then uses instead

$$G^a \;=\; D^{ab}_\mu(B)\hat{A}^\mu_b\,, \tag{273}$$

while the ghost term is given by

$$\mathcal{L}_{\mathrm{BFPG}} \;=\; -\eta^\dagger_a\, D^{ab;\mu}(B)\, D^{bc}_\mu(B+\hat{A})\,\eta_c\,. \tag{274}$$

The Lagrangian $\mathcal{L}_{\mathrm{BYM+FER}}$ then gives rise to additional interactions which are different from the normal QCD interactions of the quantum field $\hat{A}^a_\mu(x)$ also contain interactions of $B^a_\mu(x)$ with all other fields.

A remarkable fact is found when considering the renormalisation of this Lagrangian. Indeed it turns out, see e.g., [117, 118], that the coupling renormalisation, $g \to Z_g\, g$, which determines the beta function, is directly related to the renormalisation of the background field, $B \to B Z_B$, via the identity:

$$Z_g\sqrt{Z_B} \;=\; 1\,. \tag{275}$$

When working in the Landau gauge, the only anomalous dimension needed in the background field gauge formalism is then the beta function. However in the Feynman gauge the gauge parameter ξ requires the renormalisation constant Z_ξ – which equals the gluon field renormalisation constant – but only to one loop lower. In turn this allows one to extract the beta function from the single equation

$$Z_B(1+\Pi_B(Q^2;Z_\xi\xi,Z_g g)) \;=\; \text{finite}, \tag{276}$$

with

$$\Pi^{\mu\nu}_B(Q;Z_\xi\xi,Z_g g) \;=\; (Q^2 g^{\mu\nu} - Q^\mu Q^\nu)\,\Pi_B(Q^2;Z_\xi\xi,Z_g g)\,, \tag{277}$$

where $\Pi^{\mu\nu}_B(Q^2;\xi,g)$ is the bare self energy of the background field. This self-energy is computed by keeping the fields B external while the only propagating fields are $\hat{A},\eta$ and ψ.

Obtaining the beta function through the background field gauge is faster and simpler than the traditional method of computing the gluon propagator, ghost propagator and ghost-ghost-gluon vertex due to a lower total number of diagrams and the above reduction to a scalar renormalisation. The number of diagrams for the background field propagator at four loops is 7435, whereas the smallest combination needed for the beta function (gluon and ghost propagator + ghost-ghost-gluon vertex) requires $5245 + 1507 + 18034 = 24786$ diagrams. This speed-up is also evident from figure 24, which shows that calculating the background field propagator takes about as much time as computing the gluon propagator.

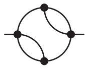
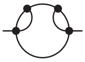
The additional Feynman rules required for the background field propagator are displayed in table 5. The vertices with two background fields attached to it yield 0, since these are tadpoles.

Now that we have discussed several optimisations and defined the background field propagator, we can focus on the computation of the five-loop beta function.

6.3 Diagram computations and analysis

As outlined in 6.2, it is possible to extract the five-loop beta function from the poles (in the dimensional regulator ϵ) of the bare background field self-energy $\Pi_B(Q)$. It is beyond current computational capabilities to calculate the required five-loop propagator integrals directly. The main obstacle preventing such an attempt is the difficulty of performing the required IBP reductions (see section 3.5).

However, the problem can be simplified via the use of the R^*-operation. This allows us to decompose the five-loop integral into a five-loop counterterm and counterterms of lower loops. We infrared-rearrange (IRR) the five-loop counter term to a carpet integral. For example:

 $\rightarrow$  . (278)

These carpet integrals can always be reduced to four loops (see section 3.3). If the counterterm is not logarithmic, we first apply the rules provided in section 6.1.3. After we use the Forcer program to compute the four-loop integrals.

The Feynman diagrams for the background propagator up to five loops have been generated using Qgraf [111]. They have then been heavily manipulated by a Form [101, 102, 216] program that determines the topology and calculates the colour factor using the program of ref. [112]. Additionally, it merges diagrams of the same topology, colour factor, and maximal power of n_f into meta diagrams for computational efficiency. Integrals containing massless tadpoles or symmetric colour tensors with an odd number of indices have been filtered out from the beginning. Lower-order self-energy insertions have been 'factorised' out. In this manner we arrive at 2 one-loop, 9 two-loop, 55 three-loop, 572 four-loop and 9414 five-loop meta diagrams. We refer the reader to section 3.10 for a more detailed description.

The diagrams up to four loops have been computed earlier to all powers of the gauge parameter using the Forcer program [1, 7, 9]. For the time being, our five-loop

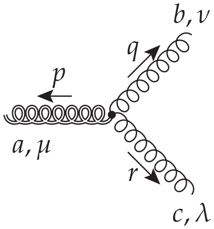
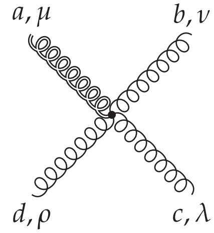
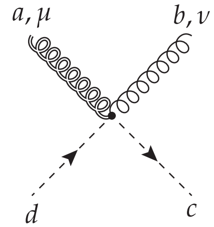
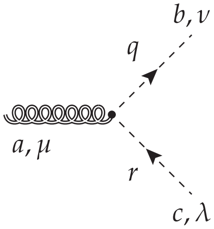
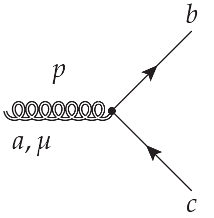
	$g f_{abc}\Big[g_{\mu\lambda}\left(p-r-\frac{q}{\xi}\right)_{\nu} + g_{\nu\lambda}\,(r-q)_{\mu} + g_{\mu\nu}\left(q-p+\frac{r}{\xi}\right)_{\lambda}\Big]$
	$-ig^2\Big[f_{abx}f_{xcd}\,(g_{\mu,\lambda}g_{\nu,\rho}-g_{\mu,\rho}g_{\nu,\lambda}) + f_{adx}f_{xbc}\,(g_{\mu,\nu}g_{\lambda,\rho}-g_{\mu,\lambda}g_{\nu,\rho}) + f_{acx}f_{xbd}\,(g_{\mu,\nu}g_{\lambda,\rho}-g_{\mu,\rho}g_{\nu,\lambda})\Big]$
	$-ig^2 f_{cax}f_{xbd}g_{\mu\nu}$
	$-gf_{bac}(r+q)_{\mu}$
	$-ig\gamma^{\mu}(t^{a})^{b}_{c}$

Table 5: Additional Feynman rules for the background field. Note the dependence on $1/\xi$ in the background-gluon-gluon vertex.

computation has been restricted to the Feynman gauge, $\xi_F = 1 - \xi = 0$. An extension to the first power in ξ_F would be considerably slower; the five-loop computation for a general ξ would be impossible without substantial further optimisations of our code. Instead of varying ξ, we have checked our computations by verifying the relation $Q_\mu Q_\nu \, \Pi_B^{\mu\nu} = 0$ required by eq. (277). This check took considerably more time than the actual determination of β_4 due to the increase in tensor rank.

The five-loop diagrams have been calculated on computers with a combined total of more than 500 cores, 80% of which are older and slower by a factor of almost three than the latest Intel Xeon 2.6 GHz workstations. One core of the latter performs a 'raw-speed' Form benchmark, a four-dimensional trace of 14 Dirac matrices, in about 0.02 seconds which corresponds to 50 'form units' (fu) per hour. The total CPU time for the five-loop diagrams was $3.8 \cdot 10^7$ seconds which corresponds to about $2.6 \cdot 10^5$ fu on the computers used. The TForm parallelisation efficiency for single meta diagrams run with 8 or 16 cores was roughly 0.5; the whole calculation of β_4, distributed 'by hand' over the available machines, finished in three days.

For comparison, the corresponding R^* computation for $\xi_F = 0$ at four loops required about 10^3 fu, which is roughly the same as for the first computation of the four-loop beta function to order ξ_F^1 by a totally different method given in ref. [113]. The computation with the Forcer program at four and fewer loops is much faster, as demonstrated in section 3.11.

The determination of Z_B from the unrenormalised background propagator is performed by imposing, order by order, the finiteness of its renormalised counterpart. The beta function can simply be read off from the $1/\varepsilon$ coefficients of Z_B. If the calculation is performed in the Landau gauge, the gauge parameter does not have to be renormalised. In a k-th order expansion about the Feynman gauge at five loops, the $L < 5$ loop contributions are needed up to ξ_F^{5-L}. The four-loop renormalisation constant for the gauge parameter is not determined in the background field and has to be 'imported'. In the present $k = 0$ case, the terms already specified in ref. [114] would have been sufficient had we not performed the four-loop calculation to all powers of ξ_F anyway.

6.4 results and discussion

Before we present our new results, it may be convenient to recall the beta function (240) up to four loops [23, 24, 113, 114, 119, 120, 122, 210, 211] in terms of the colour factors defined in section 4.1,

$$\beta_0 = \frac{11}{3} C_A - \frac{4}{3} T_F n_f \, , \tag{279}$$

$$\beta_1 = \frac{34}{3} C_A^2 - \frac{20}{3} C_A T_F n_f - 4 C_F T_F n_f \, , \tag{280}$$

$$\beta_2 = \frac{2857}{54} C_A^3 - \frac{1415}{27} C_A^2 T_F n_f - \frac{205}{9} C_F C_A T_F n_f + 2 C_F^2 T_F n_f$$

$$+\,\frac{44}{9}\,C_F\,T_F^2\,n_f^2\;+\;\frac{158}{27}\,C_A\,T_F^2\,n_f^2\;, \tag{281}$$

$$\begin{aligned}
\beta_3 \;=\;& C_A^4\left(\frac{150653}{486}-\frac{44}{9}\,\zeta_3\right)+\frac{d_A^{abcd}d_A^{abcd}}{N_A}\left(-\frac{80}{9}+\frac{704}{3}\,\zeta_3\right)\\
&+C_A^3\,T_F\,n_f\left(-\frac{39143}{81}+\frac{136}{3}\,\zeta_3\right)+C_A^2\,C_F\,T_F\,n_f\left(\frac{7073}{243}-\frac{656}{9}\,\zeta_3\right)\\
&+C_A\,C_F^2\,T_F\,n_f\left(-\frac{4204}{27}+\frac{352}{9}\,\zeta_3\right)+\frac{d_F^{abcd}d_A^{abcd}}{N_A}\,n_f\left(\frac{512}{9}-\frac{1664}{3}\,\zeta_3\right)\\
&+46\,C_F^3\,T_F\,n_f\;+C_A^2T_F^2n_f^2\left(\frac{7930}{81}+\frac{224}{9}\,\zeta_3\right)+C_F^2\,T_F^2\,n_f^2\left(\frac{1352}{27}-\frac{704}{9}\,\zeta_3\right)\\
&+C_A\,C_F\,T_F^2\,n_f^2\left(\frac{17152}{243}+\frac{448}{9}\,\zeta_3\right)+\frac{d_F^{abcd}d_F^{abcd}}{N_A}\,n_f^2\left(-\frac{704}{9}+\frac{512}{3}\,\zeta_3\right)\\
&+\frac{424}{243}\,C_A\,T_F^3\,n_f^3\;+\;\frac{1232}{243}\,C_F\,T_F^3\,n_f^3\;,
\end{aligned} \tag{282}$$

where n_f is the number of fermion (in QCD, quark) flavours. β_n are the same in all MS-like schemes [130, 131], i.e. within the class of renormalisation schemes which differ only by a shift of the scale μ.

Below we will present the results (A) for a generic Yang-Mills theory, (B) for QCD, and (C) for QED.

(A) YANG-MILLS In the same notation and scheme, the five-loop contribution reads

$$\begin{aligned}
\beta_4 \;=\;& C_A^5\left(\frac{8296235}{3888}-\frac{1630}{81}\,\zeta_3+\frac{121}{6}\,\zeta_4-\frac{1045}{9}\,\zeta_5\right)\\
&+\frac{d_A^{abcd}d_A^{abcd}}{N_A}\,C_A\left(-\frac{514}{3}+\frac{18716}{3}\,\zeta_3-968\,\zeta_4-\frac{15400}{3}\,\zeta_5\right)\\
&+C_A^4\,T_F\,n_f\left(-\frac{5048959}{972}+\frac{10505}{81}\,\zeta_3-\frac{583}{3}\,\zeta_4+1230\,\zeta_5\right)\\
&+C_A^3\,C_F\,T_F\,n_f\left(\frac{8141995}{1944}+146\,\zeta_3+\frac{902}{3}\,\zeta_4-\frac{8720}{3}\,\zeta_5\right)\\
&+C_A^2\,C_F^2\,T_F\,n_f\left(-\frac{548732}{81}-\frac{50581}{27}\,\zeta_3-\frac{484}{3}\,\zeta_4+\frac{12820}{3}\,\zeta_5\right)\\
&+C_A\,C_F^3\,T_F\,n_f\left(3717+\frac{5696}{3}\,\zeta_3-\frac{7480}{3}\,\zeta_5\right)-C_F^4\,T_F\,n_f\left(\frac{4157}{6}+128\,\zeta_3\right)\\
&+\frac{d_A^{abcd}d_A^{abcd}}{N_A}\,T_F\,n_f\left(\frac{904}{9}-\frac{20752}{9}\,\zeta_3+352\,\zeta_4+\frac{4000}{9}\,\zeta_5\right)\\
&+\frac{d_F^{abcd}d_A^{abcd}}{N_A}\,C_A\,n_f\left(\frac{11312}{9}-\frac{127736}{9}\,\zeta_3+2288\,\zeta_4+\frac{67520}{9}\,\zeta_5\right)
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{d_F^{abcd} d_A^{abcd}}{N_A} C_F\, n_f \left(-320 + \frac{1280}{3}\,\zeta_3 + \frac{6400}{3}\,\zeta_5 \right) \\
&+ C_A^3\, T_F^2\, n_f^2 \left(\frac{843067}{486} + \frac{18446}{27}\,\zeta_3 - \frac{104}{3}\,\zeta_4 - \frac{2200}{3}\,\zeta_5 \right) \\
&+ C_A^2\, C_F\, T_F^2\, n_f^2 \left(\frac{5701}{162} + \frac{26452}{27}\,\zeta_3 - \frac{944}{3}\,\zeta_4 + \frac{1600}{3}\,\zeta_5 \right) \\
&+ C_F^2\, C_A\, T_F^2\, n_f^2 \left(\frac{31583}{18} - \frac{28628}{27}\,\zeta_3 + \frac{1144}{3}\,\zeta_4 - \frac{4400}{3}\,\zeta_5 \right) \\
&+ C_F^3\, T_F^2\, n_f^2 \left(-\frac{5018}{9} - \frac{2144}{3}\,\zeta_3 + \frac{4640}{3}\,\zeta_5 \right) \\
&+ \frac{d_F^{abcd} d_A^{abcd}}{N_A} T_F\, n_f^2 \left(-\frac{3680}{9} + \frac{40160}{9}\,\zeta_3 - 832\,\zeta_4 - \frac{1280}{9}\,\zeta_5 \right) \\
&+ \frac{d_F^{abcd} d_F^{abcd}}{N_A} C_A\, n_f^2 \left(-\frac{7184}{3} + \frac{40336}{9}\,\zeta_3 - 704\,\zeta_4 + \frac{2240}{9}\,\zeta_5 \right) \\
&+ \frac{d_F^{abcd} d_F^{abcd}}{N_A} C_F\, n_f^2 \left(\frac{4160}{3} + \frac{5120}{3}\,\zeta_3 - \frac{12800}{3}\,\zeta_5 \right) \\
&+ C_A^2\, T_F^3\, n_f^3 \left(-\frac{2077}{27} - \frac{9736}{81}\,\zeta_3 + \frac{112}{3}\,\zeta_4 + \frac{320}{9}\,\zeta_5 \right) \\
&+ C_A\, C_F\, T_F^3\, n_f^3 \left(-\frac{736}{81} - \frac{5680}{27}\,\zeta_3 + \frac{224}{3}\,\zeta_4 \right) \\
&+ C_F^2\, T_F^3\, n_f^3 \left(-\frac{9922}{81} + \frac{7616}{27}\,\zeta_3 - \frac{352}{3}\,\zeta_4 \right) \\
&+ \frac{d_F^{abcd} d_F^{abcd}}{N_A} T_F\, n_f^3 \left(\frac{3520}{9} - \frac{2624}{3}\,\zeta_3 + 256\,\zeta_4 + \frac{1280}{3}\,\zeta_5 \right) \\
&+ C_A\, T_F^4\, n_f^4 \left(\frac{916}{243} - \frac{640}{81}\,\zeta_3 \right) - C_F\, T_F^4\, n_f^4 \left(\frac{856}{243} + \frac{128}{27}\,\zeta_3 \right) \,. \qquad (283)
\end{aligned}
$$

ζ denotes the Riemann zeta function with $\zeta_3 \cong 1.202056903$, $\zeta_4 = \pi^4/90 \cong 1.08232323$ and $\zeta_5 \cong 1.036927755$. As expected from the lower-order and QED results, higher values of the zeta function do not occur despite their occurrence in the results for individual diagrams; for further discussions see ref. [105, 204, 213].

(B) QCD Inserting the group factors of $SU(3)$ as given in eq. (62) leads to the QCD results

$$
\begin{aligned}
\beta_0 &= 11 - \frac{2}{3}\,n_f\,, \qquad \beta_1 = 102 - \frac{38}{3}\,n_f\,, \\
\beta_2 &= \frac{2857}{2} - \frac{5033}{18}\,n_f + \frac{325}{54}\,n_f^2\,,
\end{aligned}
$$

$$
\begin{aligned}
\beta_3 &= \frac{149753}{6} + 3564\,\zeta_3 + n_f\left(-\frac{1078361}{162} - \frac{6508}{27}\zeta_3\right) \\
&\quad + n_f^2\left(\frac{50065}{162} + \frac{6472}{81}\zeta_3\right) + \frac{1093}{729}n_f^3 \qquad (284)
\end{aligned}
$$

and

$$
\begin{aligned}
\beta_4 &= \frac{8157455}{16} + \frac{621885}{2}\zeta_3 - \frac{88209}{2}\zeta_4 - 288090\,\zeta_5 \\
&+ n_f\left(-\frac{336460813}{1944} - \frac{4811164}{81}\zeta_3 + \frac{33935}{6}\zeta_4 + \frac{1358995}{27}\zeta_5\right) \\
&+ n_f^2\left(\frac{25960913}{1944} + \frac{698531}{81}\zeta_3 - \frac{10526}{9}\zeta_4 - \frac{381760}{81}\zeta_5\right) \\
&+ n_f^3\left(-\frac{630559}{5832} - \frac{48722}{243}\zeta_3 + \frac{1618}{27}\zeta_4 + \frac{460}{9}\zeta_5\right) + n_f^4\left(\frac{1205}{2916} - \frac{152}{81}\zeta_3\right) (285)
\end{aligned}
$$

In truncated numerical form β_3 and β_4 are given by

$$
\begin{aligned}
\beta_3 &\cong 29242.964 - 6946.2896\,n_f + 405.08904\,n_f^2 + 1.499314\,n_f^3\,, \qquad (286) \\
\beta_4 &\cong 537147.67 - 186161.95\,n_f + 17567.758\,n_f^2 - 231.2777\,n_f^3 - 1.842474\,n_f^4 \ (287)
\end{aligned}
$$

In contrast to β_0, β_1, and β_2, which change sign at about $n_f = 16.5$, 8.05, and 5.84 respectively, β_3 and β_4 are positive (except at very large n_f for β_4), but have a (local) minimum at $n_f \simeq 8.20$ and $n_f \simeq 6.07$.

(c) QED The corresponding analytical result for QED, in the same renormalisation scheme(s) but defined without the overall minus sign in eq. (240) is given by

$$
\begin{aligned}
\beta_0 &= \frac{4}{3}n_f\,, \quad \beta_1 = 4\,n_f\,, \quad \beta_2 = -2\,n_f - \frac{44}{9}n_f^2\,, \\
\beta_3 &= -46\,n_f + n_f^2\left(\frac{760}{27} - \frac{832}{9}\zeta_3\right) - \frac{1232}{243}n_f^3 \qquad (288)
\end{aligned}
$$

and

$$
\begin{aligned}
\beta_4 &= n_f\left(\frac{4157}{6} + 128\,\zeta_3\right) + n_f^2\left(-\frac{7462}{9} - 992\,\zeta_3 + 2720\,\zeta_5\right) \\
&\quad + n_f^3\left(-\frac{21758}{81} + \frac{16000}{27}\zeta_3 - \frac{416}{3}\zeta_4 - \frac{1280}{3}\zeta_5\right) + n_f^4\left(\frac{856}{243} + \frac{128}{27}\zeta_3\right) \ (289)
\end{aligned}
$$

The (corresponding parts of the) results (283), (285) and (289) are in complete agreement with the findings of refs. [40, 128, 213, 214]. Consequently, eq. (289) also agrees with the result for QED at $n_f = 1$, which was obtained in ref. [44] somewhat earlier than the general result [213].

In the following sections we will analyse three properties of the beta function. First, we study the n_f-dependence in section 6.4.1. Then, we analyse the N-dependence in section 6.4.2. Finally, we study the cumulative effects of the QCD beta function corrections in section 6.4.3

6.4.1 *Analysis of n_f-dependence in QCD*

As already noted in ref. [40], the five-loop *QCD* coefficient of the beta function is rather small [recall that we use a convenient but very small expansion parameter in eq. (240)]. Indeed, for the physically relevant values of n_f the expansion in powers of α_s reads

$$\begin{aligned}
\widetilde{\beta}(\alpha_s, n_f\!=\!3) &= 1 + 0.565884\,\alpha_s + 0.453014\,\alpha_s^2 + 0.676967\,\alpha_s^3 + 0.580928\,\alpha_s^4\ , \\
\widetilde{\beta}(\alpha_s, n_f\!=\!4) &= 1 + 0.490197\,\alpha_s + 0.308790\,\alpha_s^2 + 0.485901\,\alpha_s^3 + 0.280601\,\alpha_s^4\ , \\
\widetilde{\beta}(\alpha_s, n_f\!=\!5) &= 1 + 0.401347\,\alpha_s + 0.149427\,\alpha_s^2 + 0.317223\,\alpha_s^3 + 0.080921\,\alpha_s^4\ , \\
\widetilde{\beta}(\alpha_s, n_f\!=\!6) &= 1 + 0.295573\,\alpha_s - 0.029401\,\alpha_s^2 + 0.177980\,\alpha_s^3 + 0.001555\,\alpha_s^4\ , \qquad (290)
\end{aligned}$$

where $\widetilde{\beta} \equiv -\beta(a_s)/(a_s^2\beta_0)$ has been re-expanded in powers of $\alpha_s = 4\pi\, a_s$. Clearly there is no sign so far of a possible divergence of the perturbation series for this quantity.

In order to further illustrate the n_f-dependent convergence (or the lack thereof) of the beta function of QCD, we introduce the quantity

$$\widehat{\alpha}_s^{(n)}(n_f) \;=\; 4\pi \left| \frac{\beta_{n-1}(n_f)}{4\,\beta_n(n_f)} \right| . \qquad (291)$$

Recalling the normalisation (240) of our expansion parameter, $\widehat{\alpha}_s^{(n)}(n_f)$ represents the value of α_s for which the n-th order correction is 1/4 of that of the previous order. Therefore, $\alpha_s \lesssim \widehat{\alpha}_s^{(n)}(n_f)$ defines (somewhat arbitrarily due to the choice of a factor of 1/4) a region of fast convergence of $\beta(\alpha_s, n_f)$. Obviously, the absolute size of the n-th and $(n\!-\!1)$-th order effects are equal for $\alpha_s = 4\,\widehat{\alpha}^{\,(n)}(n_f)$. Thus the quantity (291) also indicates where the expansion appears not to be reliable anymore, $\alpha_s \gtrsim 4\,\widehat{\alpha}_s^{(n)}(n_f)$, for a given value of n_f that is not too close to zeros or minima of the coefficients β_{n-1} and β_n.

6.4.2 *Analysis of N-dependence in SU(N)*

It is interesting to briefly study the N-dependence of the convergence behaviour for the case of SU(N) gauge theories. For our brief illustration we confine ourselves to pure Yang-Mills theory, $n_f = 0$, and consider

$$\widehat{\alpha}_{\mathrm{YM}}^{(n)}(N) \;=\; 4\pi\, N \left| \frac{\beta_{n-1}(N)}{4\,\beta_n(N)} \right| \;, \tag{292}$$

where the factor N compensates the leading large-N dependence N^{n+1} of β_n, i.e., the parameter that needs to be small in $SU(N)$ Yang-Mills theory is not α_{YM} but $N\alpha_{\mathrm{YM}}$.

The quantities (291) and (292) are displayed in the left and right panel of figure 33, respectively. The behaviour of $\widehat{\alpha}_{\mathrm{s}}^{(n)}$ at the upper end of the n_f range shown in the figure is affected by the zeros and minima of the coefficients $\beta_n > 0$ mentioned below eq. (287). The N-dependence of $\widehat{\alpha}_{\mathrm{YM}}$ for pure Yang-Mills theory, where only terms with N^{n+1} and N^{n-1} enter β_n (the latter only at $n \geq 4$ via $d_A^{abcd} d_A^{abcd} / N_A$, cf. eq. (62) above), is rather weak. With only the curves up to four loops, one might be tempted to draw conclusions from the shrinking of the 'stable' α_{s} region from NLO to N^{2}LO and from N^{2}LO to N^{3}LO that are not supported by the N^{4}LO (five-loop) results of ref. [40] and the present section.

6.4.3 *Cumulative effects of the QCD beta function corrections*

Finally, we briefly illustrate the cumulative effect of the orders up to N^{4}LO on the beta function of QCD and the scale dependence of the strong coupling constant α_{s} in figure 34. For this illustration we set $n_f = 4$ and choose, in order to only show the differences caused by the running of the coupling, an order-independent value of $\alpha_{\mathrm{s}} = 0.2$ at $\mu^2 = 40\ \mathrm{GeV}^2$. A realistic order dependence of α_{s} at this scale, as determined from the scaling violations in DIS, would be 0.208, 0.201, 0.200, and 0.200 at NLO, N^{2}LO, N^{3}LO, and N^{4}LO, respectively [98].

Adding the N^{4}LO contributions changes the beta function by less than 1% at $\alpha_{\mathrm{s}} = 0.47$ for $n_f = 4$ and at $\alpha_{\mathrm{s}} = 0.39$ for $n_f = 3$; the corresponding values at N^{3}LO are 0.29 and 0.26. The N^{4}LO effect on the values of α_{s} as shown in figure 34 are as small as 0.08% (0.4%) at $\mu^2 = 3\ \mathrm{GeV}^2$ (1 GeV^2); the corresponding N^{3}LO corrections are 0.5% (2%). Of course, these results do not preclude sizeable purely non-perturbative corrections, but it appears that the perturbative running of α_{s} is now fully under control for all practical purposes.

6

6.5 QCD BETA FUNCTION IN THE MINIMOM SCHEME

Unlike the $\overline{\mathrm{MS}}$ scheme, momentum subtraction schemes are defined in a regularisation-independent way. In these schemes, the field renormalisations are performed such

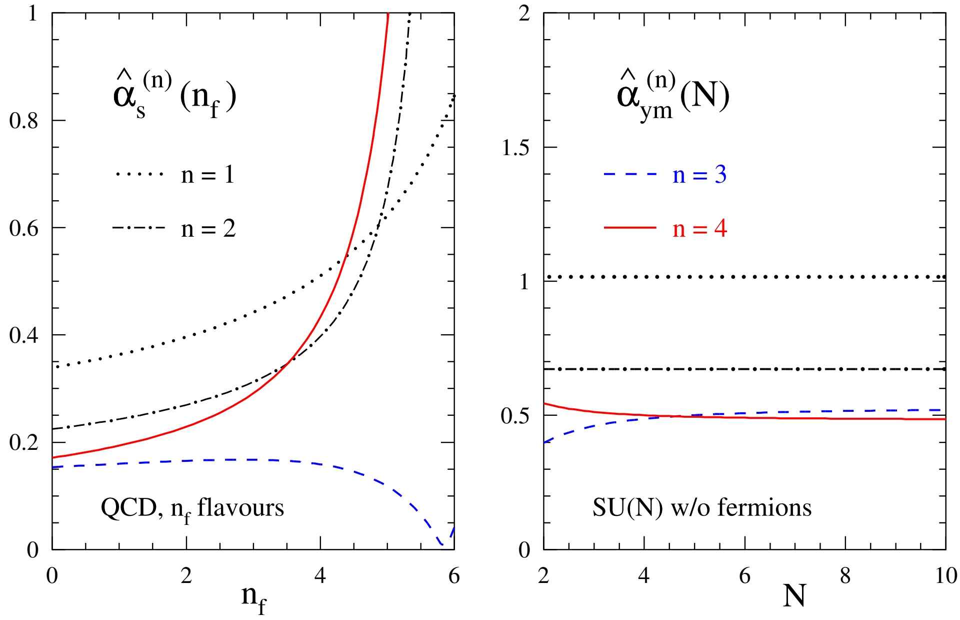


Figure 33: The values (291) and (292) of the coupling constants of QCD (left) and pure SU(N) Yang-Mills theory (right) for which the absolute size of the N^{n}LO contribution to the beta function is a quarter of that of the N^{n-1}LO term for $n = 1, 2, 3$ (dashed curves) and 4 (solid curves).

that finite radiative corrections on propagators are absorbed as well as divergences and hence they coincide with their tree-level values at the renormalisation point. Then one of (or an arbitrary linear combination of) the vertex functions is normalised to its tree-level value and the other vertices are fixed via the Slavnov-Taylor identities. Common choices for the subtraction point of the vertex are a symmetric point (referred as MOM schemes) and an asymmetric point where one of the momenta is nullified, sometimes referred as $\widetilde{\text{MOM}}$ schemes. The latter choice corresponds to our result for the vertex functions, given in section 4.2. Indeed, ref. [139] derived four-loop beta functions in four particular $\widetilde{\text{MOM}}$ schemes from that in the $\overline{\text{MS}}$ scheme by computing conversion factors via finite parts of two- and three-point functions in the $\overline{\text{MS}}$ scheme.

As an example application, we provide the five-loop beta function in the minimal momentum subtraction (MiniMOM) scheme introduced in ref. [153], thus extending previous results [153, 217] by one order in the coupling constant. This scheme, see the preceding references for a detailed discussion, is more convenient than $\overline{\text{MS}}$ for extending analyses of the strong coupling constant and its scale dependence into the non-perturbative regime, e.g., via lattice QCD; for a recent analysis see ref. [218]. In the perturbative regime the MiniMOM scheme provides an alternative to $\overline{\text{MS}}$ for studying the behaviour and truncation uncertainty of the perturbation series for

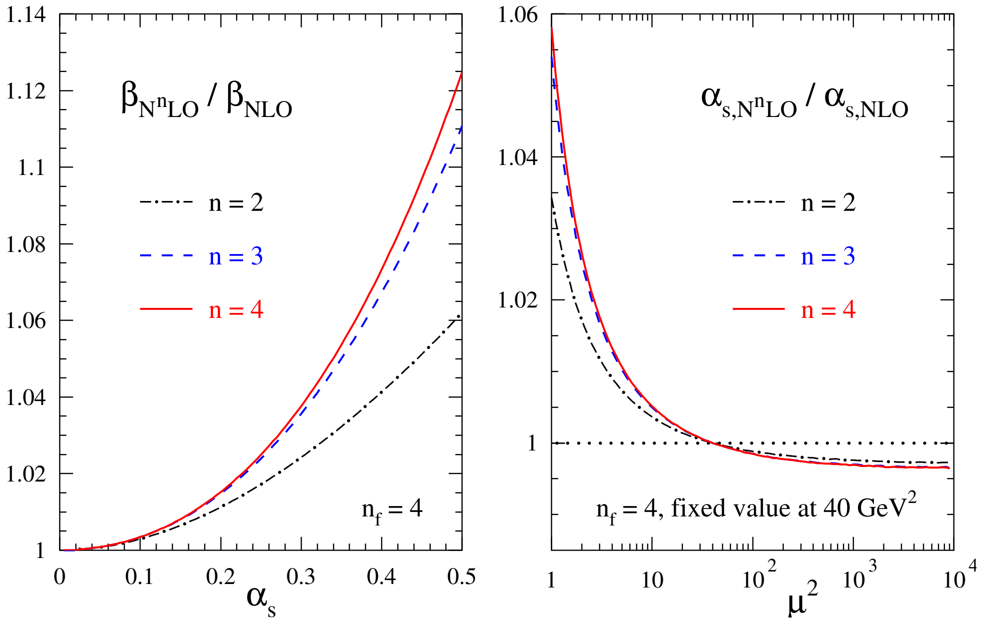


Figure 34: Left panel: The total N^{2}LO, N^{3}LO and N^{4}LO results for the beta function of QCD for four flavours, normalised to the NLO approximation. Right panel: The resulting scale dependence of α_s for a value of 0.2 at 40 GeV^2, also normalised to the NLO result in order to show the small higher-order effects more clearly, for the scale range 1 GeV$^2 \leq \mu^2 \leq 10^4$ GeV^2.

benchmark quantities such as the R-ratio in e^+e^- annihilation and the Higgs-boson decay to gluons, see refs. [219, 220].

In the MiniMOM scheme [153], the self-energies are completely absorbed into the field renormalisation constants at the subtraction point $q^2 = -\mu^2$:

$$1 + \Pi^{\mathrm{MM}}(-\mu^2) \;=\; Z_3^{\mathrm{MM}}\left[1 + \Pi^B(-\mu^2)\right] \;=\; 1, \tag{293}$$

$$1 + \tilde{\Pi}^{\mathrm{MM}}(-\mu^2) \;=\; \tilde{Z}_3^{\mathrm{MM}}\left[1 + \tilde{\Pi}^B(-\mu^2)\right] \;=\; 1, \tag{294}$$

$$1 + \Sigma_V^{\mathrm{MM}}(-\mu^2) \;=\; Z_2^{\mathrm{MM}}\left[1 + \Sigma_V^B(-\mu^2)\right] \;=\; 1. \tag{295}$$

Here the superscript "MM" indicates a quantity in the MiniMOM scheme. In addition, motivated by the non-renormalisation of the ghost-gluon vertex in the Landau gauge [221], the vertex renormalisation constant for this vertex is chosen the same as that in $\overline{\mathrm{MS}}$,

$$\tilde{Z}_1^{\mathrm{MM}} \;=\; \tilde{Z}_1^{\overline{\mathrm{MS}}}, \tag{296}$$

which is equal to one in the Landau gauge.

The above renormalisation conditions lead to the following relations for the coupling constant and gauge parameter in the two schemes:

$$a^{\mathrm{MM}}(\mu^2) \;=\; a^{\overline{\mathrm{MS}}}(\mu^2)\,\frac{1}{\left[1+\Pi^{\overline{\mathrm{MS}}}(-\mu^2)\right]\left[1+\tilde{\Pi}^{\overline{\mathrm{MS}}}(-\mu^2)\right]^2}\,, \tag{297}$$

$$\xi^{\overline{\mathrm{MS}}}(\mu^2) \;=\; \xi^{\mathrm{MM}}(\mu^2)\,\frac{1}{1+\Pi^{\overline{\mathrm{MS}}}(-\mu^2)}\,. \tag{298}$$

eq. (297) allows one to convert a value of $\alpha_s^{\overline{\mathrm{MS}}}$ to α_s^{MM}. For example, $\alpha_s^{\overline{\mathrm{MS}}}(M_Z^2) = 0.118$ leads to $\alpha_s^{\mathrm{MM}}(M_Z^2) = 1.096\,\alpha_s^{\overline{\mathrm{MS}}}(M_Z^2)$ for QCD in the Landau gauge with $n_f = 5$ quark flavours. The general expansion of eq. (297) is given in [2].

The scale dependence of the coupling constant in eq. (297) in this scheme is given by

$$\beta^{\mathrm{MM}} \;=\; \mu^2\frac{da^{\mathrm{MM}}}{d\mu^2} \;=\; \frac{\partial a^{\mathrm{MM}}}{\partial a^{\overline{\mathrm{MS}}}}\,\beta^{\overline{\mathrm{MS}}} \;+\; \frac{\partial a^{\mathrm{MM}}}{\partial \xi^{\overline{\mathrm{MS}}}}\,\gamma_3^{\overline{\mathrm{MS}}}\xi^{\overline{\mathrm{MS}}}, \tag{299}$$

where we have used the beta function and gluon field anomalous dimension in $\overline{\mathrm{MS}}$,

$$\beta^{\overline{\mathrm{MS}}} \;=\; \mu^2\frac{da^{\overline{\mathrm{MS}}}}{d\mu^2}\,, \tag{300}$$

$$\gamma_3^{\overline{\mathrm{MS}}}\xi^{\overline{\mathrm{MS}}} \;=\; \mu^2\frac{d\xi^{\overline{\mathrm{MS}}}}{d\mu^2}\,. \tag{301}$$

Note that the right-hand side of eq. (297), and hence that of eq. (299), is naturally given in terms of $a^{\overline{\mathrm{MS}}}$ and $\xi^{\overline{\mathrm{MS}}}$. One has to convert them into a^{MM} and ξ^{MM} by inverting the series of eq. (297) and by using eq. (298).[2]

Having results for the four-loop self-energies in the $\overline{\mathrm{MS}}$ scheme at hand (see section 4.2), one can obtain the five-loop beta function in the MiniMOM scheme from the five-loop beta function [4, 40] and the four-loop gluon field anomalous dimension in the $\overline{\mathrm{MS}}$ scheme. The result for SU(3) in the Landau gauge ($\xi^{\mathrm{MM}} = 0$) reads

$$\beta^{\mathrm{MM}} \;=\; -\sum_{l=0}^{4}\left(a^{\mathrm{MM}}\right)^{l+2}\beta_l^{\mathrm{MM}} \;+\; \mathcal{O}\Big(\left(a^{\mathrm{MM}}\right)^7\Big), \tag{302}$$

with

$$\beta_0^{\mathrm{MM}} = 11 - \frac{2}{3}n_f,$$

$$\beta_1^{\mathrm{MM}} = 102 - \frac{38}{3}n_f,$$

[2] In ref. [153], the results are presented in $\xi^{\overline{\mathrm{MS}}}$ instead of ξ^{MM}. In contrast, in ref. [217] the conversion from $\xi^{\overline{\mathrm{MS}}}$ to ξ^{MM} was performed. The results become the same in the Landau gauge $\xi^{\overline{\mathrm{MS}}} = \xi^{\mathrm{MM}} = 0$. The same is true for the "$\widetilde{\mathrm{MOM}}$h" scheme of ref. [139].

$$
\begin{aligned}
\beta_2^{\mathrm{MM}} &= \left(\frac{28965}{8} - \frac{3861}{8}\zeta_3\right) + n_f\left(-\frac{7715}{12} + \frac{175}{12}\zeta_3\right) + {n_f}^2\left(\frac{989}{54} + \frac{8}{9}\zeta_3\right),\\
\beta_3^{\mathrm{MM}} &= \left(\frac{1380469}{8} - \frac{625317}{16}\zeta_3 - \frac{772695}{32}\zeta_5\right) + n_f\left(-\frac{970819}{24} + \frac{516881}{72}\zeta_3\right.\\
&\quad \left. + \frac{1027375}{144}\zeta_5\right) + {n_f}^2\left(\frac{736541}{324} - \frac{6547}{27}\zeta_3 - \frac{9280}{27}\zeta_5\right) + {n_f}^3\left(-\frac{800}{27} + \frac{16}{9}\zeta_3\right),\\
\beta_4^{\mathrm{MM}} &= \left(\frac{3248220045}{256} - \frac{1064190195}{512}\zeta_3 - \frac{4922799165}{512}\zeta_5 - \frac{7696161}{64}{\zeta_3}^2\right.\\
&\quad \left. + \frac{21619456551}{4096}\zeta_7\right) + n_f\left(-\frac{115659378547}{31104} + \frac{10327103555}{20736}\zeta_3 + \frac{18219328375}{6912}\zeta_5\right.\\
&\quad \left. + \frac{82869}{32}{\zeta_3}^2 - \frac{24870449471}{18432}\zeta_7\right) + {n_f}^2\left(\frac{833934985}{2592} - \frac{13019053}{1296}\zeta_3 - \frac{65264845}{324}\zeta_5\right.\\
&\quad \left. + \frac{59531}{36}{\zeta_3}^2 + \frac{26952037}{432}\zeta_7\right) + {n_f}^3\left(-\frac{3249767}{324} - \frac{129869}{162}\zeta_3 + \frac{299875}{54}\zeta_5\right.\\
&\quad \left. - \frac{2240}{27}{\zeta_3}^2\right) + {n_f}^4\left(\frac{2617}{27} + \frac{304}{27}\zeta_3 - \frac{1760}{27}\zeta_5\right).
\end{aligned} \tag{303}
$$

Due to its length, we do not show the result with a generic group and arbitrary covariant linear gauge here. Instead, we refer the readers to ref. [2]. Our result agrees with the result given in ref. [217] up to four loops. As is well known, the first coefficient β_0^{MM} is scheme independent. The next coefficient β_1^{MM} has a gauge dependence and the universal value is obtained only in the Landau gauge. The last coefficient β_4^{MM} is the new result. In the $\overline{\mathrm{MS}}$ scheme, some of higher values of the zeta function (e.g., ζ_3^2, ζ_6 and ζ_7 at five loops) do not occur, for a discussion of this issue see refs. [105, 203, 204]. In contrast, one cannot expect their absence in the MiniMOM scheme. Indeed eq. (303) includes terms with ζ_3^2 and ζ_7, and for $\xi^{\mathrm{MM}} \neq 0$ also ζ_6 occurs.

The numerical values of the above beta function for three to five quark flavours are

$$
\begin{aligned}
\tilde{\beta}^{\mathrm{MM}}(n_f = 3) &= 1 + 0.5658842421\alpha_s^{\mathrm{MM}} + 0.9419859046(\alpha_s^{\mathrm{MM}})^2\\
&\quad + 2.304494526(\alpha_s^{\mathrm{MM}})^3 + 6.647485913(\alpha_s^{\mathrm{MM}})^4,\\
\tilde{\beta}^{\mathrm{MM}}(n_f = 4) &= 1 + 0.4901972247\alpha_s^{\mathrm{MM}} + 0.6452147391(\alpha_s^{\mathrm{MM}})^2\\
&\quad + 1.638457168(\alpha_s^{\mathrm{MM}})^3 + 3.466865543(\alpha_s^{\mathrm{MM}})^4,\\
\tilde{\beta}^{\mathrm{MM}}(n_f = 5) &= 1 + 0.4013472477\alpha_s^{\mathrm{MM}} + 0.3288519562(\alpha_s^{\mathrm{MM}})^2\\
&\quad + 1.026892491(\alpha_s^{\mathrm{MM}})^3 + 0.8417657296(\alpha_s^{\mathrm{MM}})^4,
\end{aligned} \tag{304}
$$

where $\tilde{\beta} \equiv \beta(a)/(-\beta_0 a^2)$ has been re-expanded in powers of $\alpha_s = 4\pi a$. These values may be compared with those in the $\overline{\text{MS}}$ scheme [4, 40] reading

$$\begin{aligned}
\tilde{\beta}^{\overline{\text{MS}}}(n_f = 3) &= 1 + 0.5658842421\alpha_s^{\overline{\text{MS}}} + 0.4530135791(\alpha_s^{\overline{\text{MS}}})^2 \\
&\quad + 0.6769674420(\alpha_s^{\overline{\text{MS}}})^3 + 0.5809276379(\alpha_s^{\overline{\text{MS}}})^4, \\
\tilde{\beta}^{\overline{\text{MS}}}(n_f = 4) &= 1 + 0.4901972247\alpha_s^{\overline{\text{MS}}} + 0.3087903795(\alpha_s^{\overline{\text{MS}}})^2 \\
&\quad + 0.4859007965(\alpha_s^{\overline{\text{MS}}})^3 + 0.2806008338(\alpha_s^{\overline{\text{MS}}})^4, \\
\tilde{\beta}^{\overline{\text{MS}}}(n_f = 5) &= 1 + 0.4013472477\alpha_s^{\overline{\text{MS}}} + 0.1494273313(\alpha_s^{\overline{\text{MS}}})^2 \\
&\quad + 0.3172233974(\alpha_s^{\overline{\text{MS}}})^3 + 0.08092104151(\alpha_s^{\overline{\text{MS}}})^4.
\end{aligned} \tag{305}$$

Obviously, the MiniMOM coefficients in eqs. (304) are (much) larger than their $\overline{\text{MS}}$ counterparts in eqs. (305) starting from the second order; moreover, they exhibit a definite growth with the order that is absent in the $\overline{\text{MS}}$ case. One may expect that this behaviour, and the larger value of α_s^{MM}, is more than compensated by smaller expansion coefficients for observables, leading to a better overall convergence in MOM-like schemes. However, this issue has been studied up to four loops in some detail for the R-ratio in electron-positron annihilation, without arriving at such a clear-cut conclusion [219].

6.6 CHAPTER CONCLUSIONS

We have presented five methods to improve the performance of the R^*-operation when applied to five-loop diagrams: (1) extraction of propagator insertions, (2) delayed Feynman rule substitution, (3) efficient rules to make diagrams logarithmic, (4) a canonical form algorithm suitable for Feynman diagrams, and (5) an efficient tensor reduction routine.

Next, we have defined the background field, which makes it convenient to extract the five-loop beta function. Using the background field propagator, we have computed the five-loop (next-to-next-to-next-to-next-to-leading order, N^{4}LO) coefficient β_4 of the renormalisation-group beta function in MS-like schemes for Yang-Mills theory with a simple compact Lie group and one set of n_f spin-1/2 fermions. This computation confirms and extends the QCD and QED results first obtained, respectively, in ref. [40] – where also some direct phenomenological applications to α_s determinations and Higgs-boson decay have already been discussed – and ref. [213]. It also agrees with the high-n_f partial results of refs. [128, 214]. We have verified our result and method by confirming the transversality of the background field propagator.

We have illustrated the size of the resulting N^{4}LO corrections to the scale dependence of the coupling constant for α_s-values relevant to $\overline{\text{MS}}$, the default scheme for higher-order calculations and analyses in perturbative QCD. For physical values of n_f, the N^{4}LO corrections to the beta function are much smaller than the N^{3}LO contributions and amount to 1% or less, even for α_s-values as large as 0.4. More generally, there is no evidence of any increase of the coefficients indicative of a non-convergent perturbative expansion for the beta functions of QCD and SU(N) gauge theories.

The computations make extensive use of Forcer, as described in chapter 3, and the R^*-operation described in chapter 5.

Using the four-loop propagators and vertices computed in chapter 4, we have determined the five-loop beta function in the MiniMOM scheme of ref. [153], i.e., we have extended the result of refs. [153, 217] by one order in the coupling constant α_s.

A Form file with our result for the coefficient β_4 and its lower-order counterparts can be obtained from the preprint server arXiv in the source of [4]. The beta function in the MiniMOM scheme with full gauge dependence and for generic colour group is provided as an ancillary file on arXiv of [2].

6.6.1 *Findings and main conclusion*

The main contribution of this chapter is the computation of the five-loop beta function for Yang-Mills theory with fermions. For this computation both the Forcer program (the answer to **RQ2**) and the new generalised R^* method with its computer code (the answer to **RQ3**) were critical. Our computation took six days on a 32-core machine. The QCD result we verified from [40] took 1.5 years on 20 workstations with 8 cores.

Our main conclusion therefore reads as follows: we have succeeded to compute the five-loop beta function in six days, verifying the existing QCD result and extending it to a generic Yang-Mills theory with fermions.

6.6.2 *Future work*

As of this moment we envisage two future projects. First, we have seen that the beta function in the MiniMOM scheme appears to have a higher-order structure which is quite different from that in the $\overline{\text{MS}}$ scheme, thus inviting further studies especially for the physical case of QCD in four dimensions.

Second, the beta function could be extended to six loops. However, this will be a tremendous challenge for at least three reasons: (1) a five-loop Forcer equivalent has to be built, (2) substituting the Feynman rules will create billions of terms, and (3) the number of counterterms will become enormous.

CONCLUSIONS 7

In this chapter we start by answering the three research questions in sec. 7.1. Using these answers, we address the problem statement in 7.2. Finally, we provide an outlook in sec. 7.3.

7.1 ANSWERS TO THE RESEARCH QUESTIONS

> **RQ1:** *To what extent can the number of arithmetic operations of large multivariate polynomials be reduced?*

In chapter 2 we consider various algorithms to improve expression simplification. We find that the state space of Horner schemes is flat, which makes it a good candidate for Stochastic Hill Climbing [11]. We show that for relevant polynomials derived from scattering experiments there is a speed gain of about a factor ten compared to MCTS methods to find a near-optimal solution. Since evaluations are slow, this means less time has to be spent in creating an expression suitable for Monte Carlo integration. The quality of the solution is often more than an order of magnitude better than the input.

> **RQ2:** *How can we construct a program that can compute four-loop massless propagator integrals more efficiently?*

In chapter 3 we constructed the Forcer program, which uses parametric integration-by-parts (IBP) reductions to reduce four-loop massless propagator integrals. We have demonstrated that Forcer is much faster than its competitors, and is able to compute the four-loop beta function in only 3 minutes [1, 9]. In chapter 4 we have computed physically relevant processes at four loops using Forcer [5, 8, 115]. The three most important calculations are (1) the finite pieces of the propagators and three-vertices with one vanishing momentum [2], (2) the computation of the non-singlet splitting function to $N = 16$ [115], and (3) the reconstruction of the large-n_c leading to a new term in the cusp anomalous dimension [115, 116].

> **RQ3:** *To what extent can we compute the poles of five-loop diagrams using only four-loop diagrams more efficiently?*

In chapter 5 we have generalised the R^*-operation to be applicable to Feynman diagrams with arbitrary numerator structure [3]. This allows for the computation of the poles of a much broader class of integrals. The R^*-method will generate many counterterms, but we describe how to exploit symmetries between them to reduce their number. After more optimisations described in 6, we have computed the five-loop beta function for Yang-Mills theory with fermions [4]. This confirms the QCD result from [40] and is an important ingredient for future calculations. Our calculation took six days on one 32-core machine. The computation in [40] took 1.5 years on 20 machines with 8 cores.

7.2 ANSWER TO THE PROBLEM STATEMENT

Now that we have addressed the research questions, we are able to answer the problem statement.

> **Problem statement:** *In what way can we improve the performance of QFT calculations with respect to obtaining more precise predictions?*

In answering **RQ1**, we have sped up Monte Carlo integration by improving the input expressions. Since more samples can be made, the precision of the result is increased. Our code is used in pySecDec [222], and by the GRACE collaboration [223].

In answering **RQ2**, and **RQ3** we have developed methods to drastically improve calculations of physical observables at four and five loops. Especially the computation of splitting functions and the five-loop beta function, are valuable basic ingredients in many other calculations.

Based on our findings, we may conclude that we have improved the performance of QFT calculations in three different regions. Since all our methods can be applied in practice to compare theory to experiment in colliders, we may conclude that we have improved the precision of predictions.

7.3 FUTURE RESEARCH

Below we provide four areas for future research, viz. (A) expression simplification, (B) IBP reductions, (C) Mellin moment computations, and (D) Higgs decay calculations.

(A) EXPRESSION SIMPLIFICATION We have shown that applying Horner's rule and removal of common subexpressions leads to much smaller polynomials. Polynomials could be simplified even further if algebraic structures are recognised. An example is identifying squares:

$$2ab + b^2 + 2ac + 2bc + c^2 \rightarrow (a+b+c)^2 - a^2 \ . \tag{306}$$

Recognising which terms to combine into a square in order to maximally reduce the expression is difficult (especially if numerical stability has to be taken into account as well). A first option is to see if Monte Carlo Tree Search [67] can be applied to find the best way to complete the squares. The action in each state could be the selection of a monomial that should be included in the square. A challenge is to find heuristics to guide the random playout, such that fewer samples are required.

A second option is to train a neural network to identify which monomials should be used to complete a square. The input of the network could be the exponent array of the polynomial. The output layer could yield a binary value for each monomial that determines whether it is included in the square or not. One of the challenges of a neural network is to keep the number of weights down, so that the network can be trained faster. For images, convolutional neural networks are successful, since they exploit the idea that parts of images can form a pattern by themselves [224, 225]. Presumably, something similar can be realised for expressions, but it is not obvious which monomials of the expression form a substructure that is analogous to a subrectangle in an image.

(b) ibp reductions We have shown that parametric integration-by-parts reduction rules can provide faster reductions than Laporta methods. At the moment the reduction rules require some manual intervention. If an algorithm could be devised that automatically finds high-quality reduction rules, it would mean a revolution in the field.

Currently, we are working on studying and implementing some ideas from Boolean Satisfiability problems, by defining constraints on terms that should be removed from the system. Our latest effort can reduce some hard systems, but it may require more than 500 gigabytes of disk space before a solution is found.

Additionally, it is worthwhile to study which IBP equations actually contribute to the final reduction rule. Since most equations drop out in our experience, skipping these equations from the start may save a large amount of time.

(c) mellin moments We have computed four-loop Mellin moments of splitting functions. A major challenge is that the complexity of the integrals scales linearly with the Mellin moment N. This makes it very time consuming to compute higher Mellin moments. The OPE method yields better scaling and may allow us to compute more Mellin moments. The hard part is that operators have to be constructed, which is especially difficult for the gluon. We expect new results soon [115].

Using the R^*-operation, splitting functions may be computed at five loops as well. One difficult point for the optical theorem method is that the harmonic projection creates many terms. If this operation could be postponed until after expensive operations in the R^*-routines, similar to the delayed Feynman rules, the computation could be performed much faster.

(d) higgs decay Using the R^*-operation and Forcer, we may be able to compute the Higgs decay to gluons, $H \to gg$, to five loops. The challenge is that the process consists of quartically divergent diagrams. As a result, the diagrams have to be Taylor expanded to the fourth order, which creates many terms and high-tensor subgraphs. One way to speed up the program is to choose a convenient infrared rearrangement (IRR) that places the line with the worst IR-divergence between the external lines. As a result, fewer subdiagrams will be created. Alternatively, we could add a mass to that line, which reduces the number of counterterms even further. Since the mass is only on one line, the massive part can be factorised out as a one-loop bubble.

We hope the computation will be completed within a few months of this writing.

REFERENCES

[1] B. Ruijl, T. Ueda and J.A.M. Vermaseren, *Forcer, a FORM program for the parametric reduction of four-loop massless propagator diagrams*, arXiv:1704.06650.

[2] B. Ruijl, T. Ueda, J.A.M. Vermaseren and A. Vogt, *Four-loop QCD propagators and vertices with one vanishing external momentum*, *JHEP* **2017** (2017) 1, http://dx.doi.org/10.1007/JHEP06(2017)040.

[3] F. Herzog and B. Ruijl, *The R^*-operation for Feynman graphs with generic numerators*, *JHEP* **2017** (2017) 37, http://dx.doi.org/10.1007/JHEP05(2017)037.

[4] F. Herzog, B. Ruijl, T. Ueda, J.A.M. Vermaseren and A. Vogt, *The five-loop beta function of Yang-Mills theory with fermions*, *JHEP* **02** (2017) 090 [arXiv:1701.01404].

[5] J. Davies, A. Vogt, B. Ruijl, T. Ueda and J.A.M. Vermaseren, *Large-n_f contributions to the four-loop splitting functions in QCD*, *Nucl. Phys.* **B915** (2017) 335.

[6] F. Herzog, B. Ruijl, T. Ueda, J.A.M. Vermaseren and A. Vogt, *FORM, Diagrams and Topologies*, *PoS* **LL2016** (2016) 073 [arXiv:1608.01834].

[7] T. Ueda, B. Ruijl and J.A.M. Vermaseren, *Forcer: a FORM program for 4-loop massless propagators*, *PoS* **LL2016** (2016) 070 [arXiv:1607.07318].

[8] B. Ruijl, T. Ueda, J.A.M. Vermaseren, J. Davies and A. Vogt, *First Forcer results on deep-inelastic scattering and related quantities*, *PoS* **LL2016** (2016) 071, https://arxiv.org/abs/1605.08408.

[9] T. Ueda, B. Ruijl and J.A.M. Vermaseren, *Calculating four-loop massless propagators with Forcer*, *J. Phys. Conf. Ser.* **762** (2016) 012060 [arXiv:1604.08767].

[10] B. Ruijl, T. Ueda and J. Vermaseren, *The diamond rule for multi-loop Feynman diagrams*, *Phys. Lett.* **B746** (2015) 347 [arXiv:1504.08258].

[11] B. Ruijl, J. Vermaseren, A. Plaat and H.J. van den Herik, *Why Local Search Excels in Expression Simplification*, http://arxiv.org/abs/1409.5223, 2014, arXiv:1409.5223.

[12] B. Ruijl, J. Vermaseren, A. Plaat and J. van den Herik, *Problems and New Solutions in the Boolean Domain*, ch. Simplification of Extremely Large Expressions, p. 76. Cambridge Scholars Publishing, 2016.

[13] B. Ruijl, J. Vermaseren, A. Plaat and H.J. van den Herik, *HEPGAME and the Simplification of Expressions*, *Proceedings of the 11th International Workshop on Boolean Problems* (2014), http://arxiv.org/abs/1405.6369.

[14] B. Ruijl, J. Vermaseren, A. Plaat and H.J. van den Herik, *Combining Simulated Annealing and Monte Carlo Tree Search for Expression Simplification*, *Proceedings of ICAART Conference 2014* **1** (2014) 724 [`arXiv:1312.0841`], `http://arxiv.org/abs/1312.0841`.

[15] M. Breidenbach, J.I. Friedman, H.W. Kendall, E.D. Bloom, D.H. Coward, H. DeStaebler et al., *Observed behavior of highly inelastic electron-proton scattering*, *Phys. Rev. Lett.* **23** (Oct, 1969) 935, `https://link.aps.org/doi/10.1103/PhysRevLett.23.935`.

[16] E.D. Bloom, D.H. Coward, H. DeStaebler, J. Drees, G. Miller, L.W. Mo et al., *High-energy inelastic $e - p$ scattering at 6° and 10°*, *Phys. Rev. Lett.* **23** (Oct, 1969) 930, `https://link.aps.org/doi/10.1103/PhysRevLett.23.930`.

[17] W. Heisenberg, *Über den anschaulichen inhalt der quantentheoretischen kinematik und mechanik*, *Zeitschrift für Physik* **43** (1927) 172, `http://dx.doi.org/10.1007/BF01397280`.

[18] M. Born, W. Heisenberg and P. Jordan, *Zur quantenmechanik. ii.*, *Zeitschrift für Physik* **35** (1926) 557, `http://dx.doi.org/10.1007/BF01379806`.

[19] S.L. Glashow, *Partial Symmetries of Weak Interactions*, *Nucl. Phys.* **22** (1961) 579.

[20] F. Englert and R. Brout, *Broken Symmetry and the Mass of Gauge Vector Mesons*, *Phys. Rev. Lett.* **13** (1964) 321.

[21] P.W. Higgs, *Broken Symmetries and the Masses of Gauge Bosons*, *Phys. Rev. Lett.* **13** (1964) 508.

[22] G.S. Guralnik, C.R. Hagen and T.W.B. Kibble, *Global Conservation Laws and Massless Particles*, *Phys. Rev. Lett.* **13** (1964) 585.

[23] D.J. Gross and F. Wilczek, *Ultraviolet Behavior of Nonabelian Gauge Theories*, *Phys. Rev. Lett.* **30** (1973) 1343.

[24] H.D. Politzer, *Reliable Perturbative Results for Strong Interactions?*, *Phys. Rev. Lett.* **30** (1973) 1346.

[25] R. Oerter, *The theory of almost everything: The Standard Model, the unsung triumph of modern physics*, Pi Press, New York (2006), `https://cds.cern.ch/record/1544493`.

[26] L. Evans and P. Bryant, *Lhc machine*, *Journal of Instrumentation* **3** (2008) S08001, `http://stacks.iop.org/1748-0221/3/i=08/a=S08001`.

[27] ATLAS collaboration, G. Aad et al., *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, *Phys. Lett.* **B716** (2012) 1 [`arXiv:1207.7214`].

[28] CMS collaboration, S. Chatrchyan et al., *Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC*, *Phys. Lett.* **B716** (2012) 30 [arXiv:1207.7235].

[29] G. Gabrielse, D. Hanneke, T. Kinoshita, M. Nio and B.C. Odom, *New Determination of the Fine Structure Constant from the Electron g Value and QED*, *Phys. Rev. Lett.* **97** (2006) 030802.

[30] S. Laporta, *High-precision calculation of the 4-loop contribution to the electron g-2 in QED*, arXiv:1704.06996.

[31] M.J.G. Veltman and D.N. Williams, *Schoonschip '91*, hep-ph/9306228.

[32] S. Gorishnii, S. Larin, L. Surguladze and F. Tkachov, *Mincer: Program for Multiloop Calculations in Quantum Field Theory for the Schoonschip System*, *Comput.Phys.Commun.* **55** (1989) 381.

[33] S. Larin, F. Tkachov and J. Vermaseren, *The FORM version of MINCER*, preprint NIKHEF-H-91-18, 1991, http://www.nikhef.nl/~t68/FORMapplications/Mincer/mincer.ps.

[34] S. Moch, J.A.M. Vermaseren and A. Vogt, *The Three loop splitting functions in QCD: The Nonsinglet case*, *Nucl. Phys.* **B688** (2004) 101 [hep-ph/0403192].

[35] A. Vogt, S. Moch and J.A.M. Vermaseren, *The Three-loop splitting functions in QCD: The Singlet case*, *Nucl. Phys.* **B691** (2004) 129 [hep-ph/0404111].

[36] G.P. Lepage, *VEGAS: an adaptive multidimensional integration program*, preprint CLNS-80/447, 1980.

[37] M.E. Peskin and D.V. Schroeder, *An Introduction To Quantum Field Theory (Frontiers in Physics)*, Westview Press (1995).

[38] R.P. Feynman, *Space-time approach to quantum electrodynamics*, *Phys. Rev.* **76** (Sep, 1949) 769, https://link.aps.org/doi/10.1103/PhysRev.76.769.

[39] R.P. Feynman, *Space-time approach to non-relativistic quantum mechanics*, *Rev. Mod. Phys.* **20** (Apr, 1948) 367, https://link.aps.org/doi/10.1103/RevModPhys.20.367.

[40] P.A. Baikov, K.G. Chetyrkin and J.H. Kühn, *Five-Loop Running of the QCD coupling constant*, *Phys. Rev. Lett.* **118** (2017) 082002 [arXiv:1606.08659].

[41] T. Luthe, A. Maier, P. Marquard and Y. Schröder, *Five-loop quark mass and field anomalous dimensions for a general gauge group*, *JHEP* **01** (2017) 081 [arXiv:1612.05512].

[42] S. Laporta, *High precision calculation of multiloop Feynman integrals by difference equations*, *Int.J.Mod.Phys.* **A15** (2000) 5087 [hep-ph/0102033].

[43] K. Chetyrkin and F. Tkachov, *Integration by Parts: The Algorithm to Calculate beta Functions in 4 Loops*, *Nucl.Phys.* **B192** (1981) 159.

[44] A.L. Kataev and S.A. Larin, *Analytical five-loop expressions for the renormalization group QED β-function in different renormalization schemes*, *Pisma Zh. Eksp. Teor. Fiz.* **96** (2012) 64 [arXiv:1205.2810].

[45] V.A. Smirnov and K.G. Chetyrkin, *R* Operation in the Minimal Subtraction Scheme*, *Theor. Math. Phys.* **63** (1985) 462.

[46] F.A. Berends, H. Kuijf, B. Tausk and W.T. Giele, *On the production of a W and jets at hadron colliders*, *Nucl. Phys.* **B357** (1991) 32.

[47] T. Stelzer and W.F. Long, *Automatic generation of tree level helicity amplitudes*, *Comput. Phys. Commun.* **81** (1994) 357 [hep-ph/9401258].

[48] J.M. Campbell and R.K. Ellis, *An Update on vector boson pair production at hadron colliders*, *Phys. Rev.* **D60** (1999) 113006 [hep-ph/9905386].

[49] R. Boughezal, J.M. Campbell, R.K. Ellis, C. Focke, W. Giele, X. Liu et al., *Color singlet production at NNLO in MCFM*, *Eur. Phys. J.* **C77** (2017) 7 [arXiv:1605.08011].

[50] J. Fujimoto, T. Ishikawa, K. Kato, T. Kaneko, N. Nakazawa, Y. Shimizu et al., *GRACE with FORM*, *Nucl. Phys. Proc. Suppl.* **160** (2006) 150.

[51] W. Horner, *A New Method of Solving Numerical Equations of All Orders by Continuous Approximation*, W. Bulmer & Co (1819), http://books.google.nl/books?id=2fLpGwAACAAJ.

[52] D.E. Knuth, *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1997).

[53] J. Kuipers, A. Plaat, J. Vermaseren and J. van den Herik, *Improving multivariate Horner schemes with Monte Carlo Tree Search*, *Computer Physics Communications* (2013), http://www.sciencedirect.com/science/article/pii/S0010465513001689.

[54] T. Cazenave, *Nested Monte-Carlo Search*, in *Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA,*, p. 456, 2009.

[55] A.V. Aho, R. Sethi and J.D. Ullman, *Compilers: Principles, Techniques and Tools*, Addison-Wesley (1988).

[56] C.E. Leiserson, L. Li, M.M. Maza and Y. Xie, *Efficient Evaluation of Large Polynomials*, in *In Proc. International Congress of Mathematical Software - ICMS 2010*, Springer, 2010.

[57] M.A. Breuer, *Generation of Optimal Code for Expressions via Factorization*, *Commun. ACM* **12** (June, 1969) 333, http://doi.acm.org/10.1145/363011.363153.

[58] P.S. Wang, *An improved multivariate polynomial factoring algorithm*, *Mathematics of Computation* **32** (1978) 1215, http://www.jstor.org/stable/2006346.

[59] W. Wu, J. Chen and Y. Feng, *Sparse bivariate polynomial factorization*, *Science China Mathematics* **57** (2014) 2123, http://dx.doi.org/10.1007/s11425-014-4850-y.

[60] M. Ceberio and V. Kreinovich, *Greedy Algorithms for Optimizing Multivariate Horner Schemes*, *SIGSAM Bull.* **38** (Mar., 2004) 8, http://doi.acm.org/10.1145/980175.980179.

[61] G.M. Adelson-Velsky, V.L. Arlazarov and M.V. Donskoy, *Algorithms for Games*, Springer-Verlag New York, Inc., New York, NY, USA (1988).

[62] F. Glover, *Future paths for integer programming and links to artificial intelligence*, *Computers & Operations Research* **13** (1986) 533 , http://www.sciencedirect.com/science/article/pii/0305054886900481.

[63] S. Lin and B.W. Kernighan, *An Effective Heuristic Algorithm for the Traveling-Salesman Problem*, *Operations Research* **21** (1973) 498, http://dx.doi.org/10.2307/169020.

[64] K. Helsgaun, *An effective implementation of the lin-kernighan traveling salesman heuristic*, *European Journal of Operational Research* **126** (2000) 106.

[65] C.S. Lee, M.H. Wang, G. Chaslot, J.B. Hoock, A. Rimmel, O. Teytaud et al., *The Computational Intelligence of MoGo Revealed in Taiwan's Computer Go Tournaments*, *IEEE Trans. Comput. Intellig. and AI in Games* **1** (2009) 73.

[66] R. Hayward, *Mohex wins hex tournament*, *International Computer Games Association (ICGA)* **32** (2009) 114.

[67] C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. Cowling, P. Rohlfshagen et al., *A Survey of Monte Carlo Tree Search Methods*, *Computational Intelligence and AI in Games, IEEE Transactions on* **4** (2012) 1.

[68] L. Kocsis and C. Szepesvári, *Bandit based Monte-Carlo Planning*, in *In: ECML-06. LNCS 4212*, p. 282, Springer, 2006.

[69] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, *Optimization by simulated annealing*, *Science* **220** (1983) 671.

[70] L. Kocsis and C. Szepesvári, "Discounted UCB. Video Lecture." 2006.

[71] J. Hashimoto, A. Kishimoto, K. Yoshizoe and K. Ikeda, *Accelerated UCT and Its Application to Two-Player Games*, *Lecture Notes in Computer Science* **7168** (2012) 1 , http://id.nii.ac.jp/0023/00008786.

[72] H. Hoos and T. Stützle, *Stochastic Local Search: Foundations & Applications*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2004).

[73] M. Dorigo and L.M. Gambardella, *Ant colony system: A cooperative learning approach to the traveling salesman problem*, *Trans. Evol. Comp* **1** (Apr., 1997) 53, http://dx.doi.org/10.1109/4235.585892.

[74] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1989).

[75] E. Aarts and J.K. Lenstra, eds., *Local Search in Combinatorial Optimization*, 1st ed., John Wiley & Sons, Inc., New York, NY, USA (1997).

[76] M. Miki, T. Hiroyasu and T. Jitta, *Adaptive simulated annealing for maximum temperature.*, in *SMC*, p. 20, IEEE, 2003. DOI.

[77] W. Ben-Ameur, *Computing the initial temperature of simulated annealing*, *Comput. Optim. Appl.* **29** (Dec., 2004) 369, http://dx.doi.org/10.1023/B:COAP.0000044187.23143.bd.

[78] P. Tian, J. Ma and D.M. Zhang, *Application of the simulated annealing algorithm to the combinatorial optimisation problem with permutation property: An investigation of generation mechanism*, *European Journal of Operational Research* **118** (1999) 81 , http://www.sciencedirect.com/science/article/pii/S0377221798003087.

[79] G. Reinelt, *TSPLIB: Library of TSP Instances*, https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/, 2008.

[80] J. Vermaseren, "Form source code." 1989–present.

[81] C. Anastasiou, C. Duhr, F. Dulat, F. Herzog and B. Mistlberger, *Higgs Boson Gluon-Fusion Production in QCD at Three Loops*, *Phys. Rev. Lett.* **114** (2015) 212001 [arXiv:1503.06056].

[82] C. Anastasiou, C. Duhr, F. Dulat, E. Furlan, T. Gehrmann, F. Herzog et al., *High precision determination of the gluon fusion Higgs boson cross-section at the LHC*, *JHEP* **05** (2016) 058 [arXiv:1602.00695].

[83] S.A. Larin, F.V. Tkachov and J.A.M. Vermaseren, *The O (alpha-s**3) QCD correction to the lowest moment of the longitudinal structure function in deep inelastic electron - nucleon scattering*, *Phys. Lett.* **B272** (1991) 121.

[84] S.A. Larin, T. van Ritbergen and J.A.M. Vermaseren, *The Next next-to-leading QCD approximation for nonsinglet moments of deep inelastic structure functions*, *Nucl. Phys.* **B427** (1994) 41.

[85] S.A. Larin, P. Nogueira, T. van Ritbergen and J.A.M. Vermaseren, *The Three loop QCD calculation of the moments of deep inelastic structure functions*, *Nucl. Phys.* **B492** (1997) 338 [hep-ph/9605317].

[86] A. Retey and J.A.M. Vermaseren, *Some higher moments of deep inelastic structure functions at next-to-next-to-leading order of perturbative QCD*, *Nucl. Phys.* **B604** (2001) 281 [hep-ph/0007294].

[87] J. Blumlein and J.A.M. Vermaseren, *The 16th moment of the non-singlet structure functions F(2)(x,Q**2) and F(L)(x,Q**2) to O(alpha**3(S))*, *Phys. Lett.* **B606** (2005) 130 [hep-ph/0411111].

[88] S. Moch, J.A.M. Vermaseren and A. Vogt, *The Three-Loop Splitting Functions in QCD: The Helicity-Dependent Case*, *Nucl. Phys.* **B889** (2014) 351 [arXiv:1409.5131].

[89] P.A. Baikov, *Explicit solutions of the three loop vacuum integral recurrence relations*, *Phys. Lett.* **B385** (1996) 404 [hep-ph/9603267].

[90] P.A. Baikov, *Explicit solutions of the multiloop integral recurrence relations and its application*, *Nucl. Instrum. Meth.* **A389** (1997) 347 [hep-ph/9611449].

[91] P.A. Baikov, *A Practical criterion of irreducibility of multi-loop Feynman integrals*, *Phys. Lett.* **B634** (2006) 325 [hep-ph/0507053].

[92] C. Anastasiou and A. Lazopoulos, *Automatic integral reduction for higher order perturbative calculations*, *JHEP* **0407** (2004) 046 [hep-ph/0404258].

[93] A. Smirnov, *Algorithm FIRE – Feynman Integral REduction*, *JHEP* **0810** (2008) 107 [arXiv:0807.3243].

[94] A.V. Smirnov, *FIRE5: a C++ implementation of Feynman Integral REduction*, *Comput.Phys.Commun.* **189** (2014) 182 [arXiv:1408.2372].

[95] C. Studerus, *Reduze-Feynman Integral Reduction in C++*, *Comput.Phys.Commun.* **181** (2010) 1293 [arXiv:0912.2546].

[96] A. von Manteuffel and C. Studerus, *Reduze 2 - Distributed Feynman Integral Reduction*, arXiv:1201.4330.

[97] F. Tkachov, *An Algorithm for Calculating Multiloop Integrals*, *Theor.Math.Phys.* **56** (1983) 866.

[98] J.A.M. Vermaseren, A. Vogt and S. Moch, *The Third-order QCD corrections to deep-inelastic scattering by photon exchange*, *Nucl. Phys.* **B724** (2005) 3 [hep-ph/0504242].

[99] R. Lee, *Presenting LiteRed: a tool for the Loop InTEgrals REDuction*, `arXiv:1212.2685`.

[100] R.N. Lee, *LiteRed 1.4: a powerful tool for reduction of multiloop integrals*, *J.Phys.Conf.Ser.* **523** (2014) 012059 [`arXiv:1310.1145`].

[101] M. Tentyukov and J.A.M. Vermaseren, *The Multithreaded version of FORM*, *Comput. Phys. Commun.* **181** (2010) 1419 [`hep-ph/0702279`].

[102] J. Kuipers, T. Ueda, J.A.M. Vermaseren and J. Vollinga, *FORM version 4.0*, *Comput. Phys. Commun.* **184** (2013) 1453 [`arXiv:1203.6543`].

[103] F. Tkachov, *A Theorem on Analytical Calculability of Four Loop Renormalization Group Functions*, *Phys.Lett.* **B100** (1981) 65.

[104] J.A.M. Vermaseren, *The rules of physics*, *Nucl. Instrum. Meth.* **A534** (2004) 232.

[105] P.A. Baikov and K.G. Chetyrkin, *Four Loop Massless Propagators: An Algebraic Evaluation of All Master Integrals*, *Nucl. Phys.* **B837** (2010) 186 [`arXiv:1004.1153`].

[106] R.N. Lee, A.V. Smirnov and V.A. Smirnov, *Master Integrals for Four-Loop Massless Propagators up to Transcendentality Weight Twelve*, *Nucl. Phys.* **B856** (2012) 95 [`arXiv:1108.0732`].

[107] G. Csardi and T. Nepusz, *The igraph software package for complex network research*, *InterJournal* **Complex Systems** (2006) 1695, `http://igraph.org`.

[108] S. van der Walt, S.C. Colbert and G. Varoquaux, *The NumPy array: A structure for efficient numerical computation*, *Computing in Science & Engineering* **13** (2011) 22 [`arXiv:1102.1523`], `http://www.numpy.org`.

[109] A. Meurer et al., *SymPy: symbolic computing in Python*, *PeeJ Computer Science* (2017) 3:e103, `http://www.sympy.org`.

[110] J.A.M. Vermaseren, *Status of FORM*, talk presented in Durbach, 2014.

[111] P. Nogueira, *Automatic Feynman graph generation*, *J. Comput. Phys.* **105** (1993) 279.

[112] T. van Ritbergen, A.N. Schellekens and J.A.M. Vermaseren, *Group theory factors for Feynman diagrams*, *Int. J. Mod. Phys.* **A14** (1999) 41 [`hep-ph/9802376`].

[113] T. van Ritbergen, J.A.M. Vermaseren and S.A. Larin, *The Four loop beta function in quantum chromodynamics*, *Phys. Lett.* **B400** (1997) 379 [`hep-ph/9701390`].

[114] M. Czakon, *The four-loop QCD beta-function and anomalous dimensions*, *Nucl. Phys.* **B710** (2005) 485 [`hep-ph/0411261`].

[115] S. Moch, B. Ruijl, T. Ueda, J. Vermaseren and A. Vogt, “Tbd.” 2017.

[116] B. Ruijl, *Towards five loop calculations in QCD*, seminar in Zurich, December 2016, `http://www.physik.uzh.ch/en/seminars/ttpseminar/HS2016.html`.

[117] L.F. Abbott, *The Background Field Method Beyond One Loop*, *Nucl. Phys.* **B185** (1981) 189.

[118] L.F. Abbott, M.T. Grisaru and R.K. Schaefer, *The Background Field Method and the S Matrix*, *Nucl. Phys.* **B229** (1983) 372.

[119] D.R.T. Jones, *Two Loop Diagrams in Yang-Mills Theory*, *Nucl. Phys.* **B75** (1974) 531.

[120] E. Egorian and O.V. Tarasov, *Two Loop Renormalization of the QCD in an Arbitrary Gauge*, *Teor. Mat. Fiz.* **41** (1979) 26.

[121] R. Tarrach, *The Pole Mass in Perturbative QCD*, *Nucl. Phys.* **B183** (1981) 384.

[122] S.A. Larin and J.A.M. Vermaseren, *The Three loop QCD Beta function and anomalous dimensions*, *Phys. Lett.* **B303** (1993) 334 [`hep-ph/9302208`].

[123] O.V. Tarasov, *Anomalous dimensions of quark masses in three loop approximation*, preprint `JINR-P2-82-900`, 1982.

[124] K.G. Chetyrkin, M. Misiak and M. Munz, *Beta functions and anomalous dimensions up to three loops*, *Nucl. Phys.* **B518** (1998) 473 [`hep-ph/9711266`].

[125] J.A.M. Vermaseren, S.A. Larin and T. van Ritbergen, *The four loop quark mass anomalous dimension and the invariant quark mass*, *Phys. Lett.* **B405** (1997) 327 [`hep-ph/9703284`].

[126] K.G. Chetyrkin, *Four-loop renormalization of QCD: Full set of renormalization constants and anomalous dimensions*, *Nucl. Phys.* **B710** (2005) 499 [`hep-ph/0405193`].

[127] P.A. Baikov, K.G. Chetyrkin and J.H. Kühn, *Quark Mass and Field Anomalous Dimensions to $\mathcal{O}(\alpha_s^5)$*, *JHEP* **10** (2014) 076 [`arXiv:1402.6611`].

[128] T. Luthe, A. Maier, P. Marquard and Y. Schröder, *Towards the five-loop beta function for a general gauge group*, *JHEP* **07** (2016) 127 [`arXiv:1606.08662`].

[129] T. Luthe, A. Maier, P. Marquard and Y. Schröder, *Complete renormalization of QCD at five loops*, *JHEP* **03** (2017) 020 [`arXiv:1701.07068`].

[130] G. 't Hooft, *Dimensional regularization and the renormalization group*, *Nucl. Phys.* **B61** (1973) 455.

[131] W.A. Bardeen, A.J. Buras, D.W. Duke and T. Muta, *Deep Inelastic Scattering Beyond the Leading Order in Asymptotically Free Gauge Theories*, *Phys. Rev.* **D18** (1978) 3998.

[132] A.A. Vladimirov, *Method for Computing Renormalization Group Functions in Dimensional Renormalization Scheme*, *Theor. Math. Phys.* **43** (1980) 417.

[133] K.G. Chetyrkin and F.V. Tkachov, *Infrared R-operation and ultraviolet counterterms in the MS-scheme*, *Phys. Lett.* **B114** (1982) 340.

[134] K.G. Chetyrkin and V.A. Smirnov, *R^*-operation corrected*, *Phys. Lett.* **B144** (1984) 419.

[135] K.G. Chetyrkin, *Combinatorics of $\mathbf{R}$-, $\mathbf{R}^{-1}$-, and $\mathbf{R}^*$-operations and asymptotic expansions of feynman integrals in the limit of large momenta and masses*, `arXiv:1701.08627`.

[136] K.G. Chetyrkin, *Correlator of the quark scalar currents and $\Gamma_{tot}(H \to hadrons)$ at $\mathcal{O}(\alpha_s^3)$ in pQCD*, *Phys. Lett.* **B390** (1997) 309 [`hep-ph/9608318`].

[137] K.G. Chetyrkin, *Corrections of order α_s^3 to R_{had} in pQCD with light gluinos*, *Phys. Lett.* **B391** (1997) 402 [`hep-ph/9608480`].

[138] M. Misiak and M. Munz, *Two loop mixing of dimension five flavor changing operators*, *Phys. Lett.* **B344** (1995) 308 [`hep-ph/9409454`].

[139] K.G. Chetyrkin and A. Retey, *Three loop three linear vertices and four loop similar to MOM beta functions in massless QCD*, `hep-ph/0007088`.

[140] W. Celmaster and R.J. Gonsalves, *The Renormalization Prescription Dependence of the QCD Coupling Constant*, *Phys. Rev.* **D20** (1979) 1420.

[141] J.S. Ball and T.W. Chiu, *Analytic Properties of the Vertex Function in Gauge Theories. 2.*, *Phys. Rev.* **D22** (1980) 2550 [*Erratum ibid.* **D23** (1981) 3085].

[142] P. Pascual and R. Tarrach, *Slavnov-Taylor Identities in Weinberg's Renormalization Scheme*, *Nucl. Phys.* **B174** (1980) 123 [*Erratum ibid.* **B181** (1981) 546].

[143] E. Braaten and J.P. Leveille, *Minimal Subtraction and Momentum Subtraction in QCD at Two Loop Order*, *Phys. Rev.* **D24** (1981) 1369.

[144] A.I. Davydychev, P. Osland and O.V. Tarasov, *Three gluon vertex in arbitrary gauge and dimension*, *Phys. Rev.* **D54** (1996) 4087 [*Erratum ibid.* **D59** (1999) 109901] [`hep-ph/9605348`].

[145] A.I. Davydychev, P. Osland and L. Saks, *Quark mass dependence of the one loop three gluon vertex in arbitrary dimension*, *JHEP* **08** (2001) 050 [`hep-ph/0105072`].

[146] M. Binger and S.J. Brodsky, *The Form-factors of the gauge-invariant three-gluon vertex*, *Phys. Rev.* **D74** (2006) 054016 [`hep-ph/0602199`].

[147] A.I. Davydychev, P. Osland and O.V. Tarasov, *Two loop three gluon vertex in zero momentum limit*, *Phys. Rev.* **D58** (1998) 036007 [`hep-ph/9801380`].

[148] A.I. Davydychev and P. Osland, *On-shell two loop three gluon vertex*, *Phys. Rev.* **D59** (1999) 014006 [hep-ph/9806522].

[149] A.I. Davydychev, P. Osland and L. Saks, *Quark gluon vertex in arbitrary gauge and dimension*, *Phys. Rev.* **D63** (2001) 014022 [hep-ph/0008171].

[150] K.G. Chetyrkin and T. Seidensticker, *Two loop QCD vertices and three loop MOM beta functions*, *Phys. Lett.* **B495** (2000) 74 [hep-ph/0008094].

[151] J.A. Gracey, *Two loop QCD vertices at the symmetric point*, *Phys. Rev.* **D84** (2011) 085011 [arXiv:1108.4806].

[152] J.A. Gracey, *Off-shell two-loop QCD vertices*, *Phys. Rev.* **D90** (2014) 025014 [arXiv:1406.0649].

[153] L. von Smekal, K. Maltman and A. Sternbeck, *The Strong coupling and its running to four loops in a minimal MOM scheme*, *Phys. Lett.* **B681** (2009) 336 [arXiv:0903.1696].

[154] S. Alekhin, J. Blümlein and S.O. Moch, *α_s from global fits of parton distribution functions*, *Mod. Phys. Lett.* **A31** (2016) 1630023.

[155] C. Anzai, A. Hasselhuhn, M. Höschele, J. Hoff, W. Kilgore, M. Steinhauser et al., *Exact N^{3}LO results for qq′ to H + X*, *JHEP* **07** (2015) 140 [arXiv:1506.02674].

[156] P.A. Baikov and K.G. Chetyrkin, *New four loop results in QCD*, *Nucl. Phys. Proc. Suppl.* **160** (2006) 76.

[157] V.N. Velizhanin, *Four loop anomalous dimension of the second moment of the non-singlet twist-2 operator in QCD*, *Nucl. Phys.* **B860** (2012) 288 [arXiv:1112.3954].

[158] V.N. Velizhanin, *Four loop anomalous dimension of the third and fourth moments of the non-singlet twist-2 operator in QCD*, arXiv:1411.1331.

[159] P.A. Baikov, K.G. Chetyrkin and J.H. Kühn, *Massless Propagators, $R(s)$ and Multiloop QCD*, *Nucl. Part. Phys. Proc.* **261-262** (2015) 3 [arXiv:1501.06739].

[160] N.H. Christ, B. Hasslacher and A.H. Mueller, *Light cone behavior of perturbation theory*, *Phys. Rev.* **D6** (1972) 3543.

[161] W.H. Furry, *A Symmetry Theorem in the Positron Theory*, *Phys. Rev.* **51** (1937) 125.

[162] C.G. Callan, *Broken scale invariance in scalar field theory*, *Phys. Rev. D* **2** (Oct, 1970) 1541, https://link.aps.org/doi/10.1103/PhysRevD.2.1541.

[163] K. Symanzik, *Small distance behaviour in field theory and power counting*, *Communications in Mathematical Physics* **18** (1970) 227, http://dx.doi.org/10.1007/BF01649434.

[164] V.N. Gribov and L.N. Lipatov, *Deep inelastic e p scattering in perturbation theory*, *Sov. J. Nucl. Phys.* **15** (1972) 438.

[165] Y.L. Dokshitzer, *Calculation of the Structure Functions for Deep Inelastic Scattering and e+ e- Annihilation by Perturbation Theory in Quantum Chromodynamics.*, *Sov. Phys. JETP* **46** (1977) 641.

[166] G. Altarelli and G. Parisi, *Asymptotic Freedom in Parton Language*, *Nucl. Phys.* **B126** (1977) 298.

[167] W. Furmanski and R. Petronzio, *Lepton - Hadron Processes Beyond Leading Order in Quantum Chromodynamics*, *Z. Phys.* **C11** (1982) 293.

[168] V.N. Velizhanin, *Three loop anomalous dimension of the non-singlet transversity operator in QCD*, *Nucl. Phys.* **B864** (2012) 113 [arXiv:1203.1022].

[169] G.P. Korchemsky, *Asymptotics of the Altarelli-Parisi-Lipatov Evolution Kernels of Parton Distributions*, *Mod. Phys. Lett.* **A4** (1989) 1257.

[170] S. Albino and R.D. Ball, *Soft resummation of quark anomalous dimensions and coefficient functions in MS-bar factorization*, *Phys. Lett.* **B513** (2001) 93 [hep-ph/0011133].

[171] Yu.L. Dokshitzer, G. Marchesini and G.P. Salam, *Revisiting parton evolution and the large-x limit*, *Phys. Lett.* **B634** (2006) 504 [hep-ph/0511302].

[172] J. Henn, A.V. Smirnov, V.A. Smirnov, M. Steinhauser and R.N. Lee, *Four-loop photon quark form factor and cusp anomalous dimension in the large-N_c limit of QCD*, arXiv:1612.04389.

[173] D.A. Akyeampong and R. Delbourgo, *Dimensional regularization, abnormal amplitudes and anomalies*, *Nuovo Cim.* **A17** (1973) 578.

[174] S.A. Larin, *The Renormalization of the axial anomaly in dimensional regularization*, *Phys. Lett.* **B303** (1993) 113 [hep-ph/9302240].

[175] S. Moch, J.A.M. Vermaseren and A. Vogt, *On γ_5 in higher-order QCD calculations and the NNLO evolution of the polarized valence distribution*, *Phys. Lett.* **B748** (2015) 432 [arXiv:1506.04517].

[176] P.A. Baikov, K.G. Chetyrkin, J.H. Kuhn and J. Rittinger, *Complete $\mathcal{O}(\alpha_s^4)$ QCD Corrections to Hadronic Z-Decays*, *Phys. Rev. Lett.* **108** (2012) 222003 [arXiv:1201.5804].

[177] T. Kinoshita, *Mass singularities of Feynman amplitudes*, *J. Math. Phys.* **3** (1962) 650.

[178] T.D. Lee and M. Nauenberg, *Degenerate Systems and Mass Singularities*, *Phys. Rev.* **133** (1964) B1549.

[179] C.G. Bollini and J.J. Giambiagi, *Dimensional Renormalization: The Number of Dimensions as a Regularizing Parameter*, *Nuovo Cim.* **B12** (1972) 20.

[180] G. 't Hooft and M.J.G. Veltman, *Regularization and Renormalization of Gauge Fields*, *Nucl. Phys.* **B44** (1972) 189.

[181] A.V. Kotikov, *Differential equations method: New technique for massive Feynman diagrams calculation*, *Phys. Lett.* **B254** (1991) 158.

[182] T. Gehrmann and E. Remiddi, *Differential equations for two loop four point functions*, *Nucl. Phys.* **B580** (2000) 485 [hep-ph/9912329].

[183] J.M. Henn, *Multiloop integrals in dimensional regularization made simple*, *Phys. Rev. Lett.* **110** (2013) 251601 [arXiv:1304.1806].

[184] V.A. Smirnov, *Analytical result for dimensionally regularized massless on shell double box*, *Phys. Lett.* **B460** (1999) 397 [hep-ph/9905323].

[185] J.B. Tausk, *Nonplanar massless two loop Feynman diagrams with four on-shell legs*, *Phys. Lett.* **B469** (1999) 225 [hep-ph/9909506].

[186] C. Anastasiou and A. Daleo, *Numerical evaluation of loop integrals*, *JHEP* **10** (2006) 031 [hep-ph/0511176].

[187] M. Czakon, *Automatized analytic continuation of Mellin-Barnes integrals*, *Comput. Phys. Commun.* **175** (2006) 559 [hep-ph/0511200].

[188] T. Binoth and G. Heinrich, *An automatized algorithm to compute infrared divergent multiloop integrals*, *Nucl. Phys.* **B585** (2000) 741 [hep-ph/0004013].

[189] M. Roth and A. Denner, *High-energy approximation of one loop Feynman integrals*, *Nucl. Phys.* **B479** (1996) 495 [hep-ph/9605420].

[190] K. Hepp, *Proof of the Bogolyubov-Parasiuk theorem on renormalization*, *Commun. Math. Phys.* **2** (1966) 301.

[191] E. Panzer, *On hyperlogarithms and Feynman integrals with divergences and many scales*, *JHEP* **03** (2014) 071 [arXiv:1401.4361].

[192] A. von Manteuffel, E. Panzer and R.M. Schabinger, *A quasi-finite basis for multi-loop Feynman integrals*, *JHEP* **02** (2015) 120 [arXiv:1411.7392].

[193] N.N. Bogoliubov and O.S. Parasiuk, *On the Multiplication of the causal function in the quantum theory of fields*, *Acta Math.* **97** (1957) 227.

[194] W. Zimmermann, *Convergence of Bogolyubov's method of renormalization in momentum space*, *Commun. Math. Phys.* **15** (1969) 208.

[195] W.E. Caswell and A.D. Kennedy, *A simple approach to renormalization theory*, *Phys. Rev.* **D25** (1982) 392.

[196] D. Kreimer, *On the Hopf algebra structure of perturbative quantum field theories*, *Adv. Theor. Math. Phys.* **2** (1998) 303 [q-alg/9707029].

[197] A. Connes and D. Kreimer, *Renormalization in quantum field theory and the Riemann-Hilbert problem. 1. The Hopf algebra structure of graphs and the main theorem*, *Commun. Math. Phys.* **210** (2000) 249 [hep-th/9912092].

[198] K.G. Chetyrkin, A.L. Kataev and F.V. Tkachov, *Five Loop Calculations in the $g\phi^4$ Model and the Critical Index η*, *Phys. Lett.* **B99** (1981) 147.

[199] S.G. Gorishnii, S.A. Larin, F.V. Tkachov and K.G. Chetyrkin, *Five Loop Renormalization Group Calculations in the $g\phi^4$ in Four-dimensions Theory*, *Phys. Lett.* **B132** (1983) 351.

[200] H. Kleinert, J. Neu, V. Schulte-Frohlinde, K.G. Chetyrkin and S.A. Larin, *Five loop renormalization group functions of O(n) symmetric phi**4 theory and epsilon expansions of critical exponents up to epsilon**5*, *Phys. Lett.* **B272** (1991) 39 [hep-th/9503230].

[201] M. Kompaniets and E. Panzer, *Renormalization group functions of ϕ^4 theory in the MS-scheme to six loops*, *PoS* **LL2016** (2016) 038 [arXiv:1606.09210].

[202] D.V. Batkovich, K.G. Chetyrkin and M.V. Kompaniets, *Six loop analytical calculation of the field anomalous dimension and the critical exponent η in $O(n)$-symmetric φ^4 model*, *Nucl. Phys.* **B906** (2016) 147 [arXiv:1601.01960].

[203] P.A. Baikov, K.G. Chetyrkin and J.H. Kühn, *Vector Correlator in Massless QCD at Order $O(\alpha_s^4)$ and the QED beta-function at Five Loop*, *JHEP* **07** (2012) 017 [arXiv:1206.1284].

[204] P.A. Baikov, K.G. Chetyrkin and J.H. Kühn, *Five-loop fermion anomalous dimension for a general gauge group from four-loop massless propagators*, arXiv:1702.01458.

[205] P.A. Baikov, K.G. Chetyrkin and J.H. Kuhn, *Scalar correlator at O(alpha(s)**4), Higgs decay into b-quarks and bounds on the light quark masses*, *Phys. Rev. Lett.* **96** (2006) 012003 [hep-ph/0511063].

[206] H. Kleinert and V. Schulte-Frohlinde, *Critical Properties of ϕ^4-Theories*, World Scientific (2001).

[207] S. Larin and P. van Nieuwenhuizen, *The Infrared R* operation*, hep-th/0212315.

[208] D.V. Batkovich and M. Kompaniets, *Toolbox for multiloop Feynman diagrams calculations using R^* operation*, *J. Phys. Conf. Ser.* **608** (2015) 012068 [arXiv:1411.2618].

[209] E.R. Speer, *Contraction Anomalies in Dimensional Renormalization*, *Nucl. Phys.* **B134** (1978) 175.

[210] W.E. Caswell, *Asymptotic Behavior of Nonabelian Gauge Theories to Two Loop Order*, *Phys. Rev. Lett.* **33** (1974) 244.

[211] O.V. Tarasov, A.A. Vladimirov and A.Yu. Zharkov, *The Gell-Mann-Low Function of QCD in the Three Loop Approximation*, *Phys. Lett.* **B93** (1980) 429.

[212] S. Moch, J.A.M. Vermaseren and A. Vogt, *Third-order QCD corrections to the charged-current structure function F(3)*, *Nucl. Phys.* **B813** (2009) 220 [arXiv:0812.4168].

[213] P.A. Baikov, K.G. Chetyrkin, J.H. Kuhn and J. Rittinger, *Vector Correlator in Massless QCD at Order $O(\alpha_s^4)$ and the QED beta-function at Five Loop*, *JHEP* **07** (2012) 017 [arXiv:1206.1284].

[214] J.A. Gracey, *The QCD Beta function at O(1/N(f))*, *Phys. Lett.* **B373** (1996) 178 [hep-ph/9602214].

[215] B.D. McKay, *Practical Graph Isomorphism*, *Congressus Numerantium* **30** (1981) 45, http://users.cecs.anu.edu.au/~bdm/nauty/pgi.pdf.

[216] J.A.M. Vermaseren, *New features of FORM*, math-ph/0010025.

[217] J.A. Gracey, *Renormalization group functions of QCD in the minimal MOM scheme*, *J. Phys.* **A46** (2013) 225403 [*Erratum ibid.* **A48** (2015) 119501] [arXiv:1304.5347].

[218] C. Ayala, G. Cvetic, R. Kogerler and I. Kondrashuk, *Nearly perturbative lattice-motivated QCD coupling with zero IR limit*, arXiv:1703.01321.

[219] J. Gracey, *Momentum subtraction and the R ratio*, *Phys. Rev.* **D90** (2014) 094026 [arXiv:1410.6715].

[220] D.M. Zeng, S.Q. Wang, X.G. Wu and J.M. Shen, *The Higgs-boson decay $H \to gg$ up to α_s^5-order under the minimal momentum space subtraction scheme*, *J. Phys.* **G43** (2016) 075001 [arXiv:1507.03222].

[221] J.C. Taylor, *Ward Identities and Charge Renormalization of the Yang-Mills Field*, *Nucl. Phys.* **B33** (1971) 436.

[222] S. Borowka, G. Heinrich, S. Jahn, S.P. Jones, M. Kerner, J. Schlenk et al., *pySecDec: a toolbox for the numerical evaluation of multi-scale integrals*, arXiv:1703.09692.

[223] J. Fujimoto et al., *The GRACE project: QCD, SUSY, multi-loop*, *PoS* **RADCOR2011** (2011) 012.

[224] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, *Gradient-based learning applied to document recognition*, in *Proceedings of the IEEE*, p. 2278, 1998.

[225] A. Krizhevsky, I. Sutskever and G.E. Hinton, *Imagenet classification with deep convolutional neural networks*, in *Advances in Neural Information Processing Systems 25* (F. Pereira, C.J.C. Burges, L. Bottou and K.Q. Weinberger, eds.), p. 1097. Curran Associates, Inc., 2012.

APPENDICES

Below we present three appendices. In appendix A we present a proof for the cutvertex rule for scalar diagrams (see section 5.3.2), and in appendix B for tensor diagrams (see section 5.3.2). Finally, we provide an IR subgraph search algorithm (see section 5.4.1) in appendix C.

A
CUTVERTEX RULE FOR SCALAR DIAGRAMS

The cutvertex rule states that

$$\Delta(\gamma_1\gamma_2) = \Delta(\gamma_1)\Delta(\gamma_2)\,. \tag{307}$$

This statement can be proven by induction. We start by proving that the statement holds true for the trivial case, where both γ_1 and γ_2 contain no subdivergences. This can be proven as follows:

$$\begin{aligned}\Delta(\gamma_1\gamma_2) &= -K\bar{R}(\gamma_1\gamma_2)\\ &= -K\big(\gamma_1\gamma_2 + \Delta(\gamma_1)\gamma_2 + \Delta(\gamma_2)\gamma_1\big)\\ &= -K\big((\gamma_1 + \Delta(\gamma_1))(\gamma_2 + \Delta(\gamma_2)) - \Delta(\gamma_1)\Delta(\gamma_2)\big)\\ &= -K\big(R(\gamma_1)R(\gamma_2) - \Delta(\gamma_1)\Delta(\gamma_2)\big)\\ &= K\big(\Delta(\gamma_1)\Delta(\gamma_2)\big)\\ &= \Delta(\gamma_1)\Delta(\gamma_2)\ .\end{aligned} \tag{308}$$

Now we can prove inductively that the same holds for the general case, where we assume that both γ_1 and γ_2 have subdivergences. That is, we show that

$$\Delta(G_1G_2) = \Delta(G_1)\Delta(G_2) \tag{309}$$

holds, assuming the induction hypothesis $\Delta(\gamma_1\gamma_2) = \Delta(\gamma_1)\Delta(\gamma_2)$ where γ_1 and γ_2 are subgraphs of G_1 and G_2 respectively. Let us start with the definition:

$$\begin{aligned}\Delta(G_1G_2) &= -K\bar{R}(G_1G_2)\\ &= -K\sum_{S\in\bar{W}(G_1G_2)} \Delta(S) * G_1G_2/S\,.\end{aligned} \tag{310}$$

We will now use the fact that we can write

$$\bar{W}(G_1G_2) = W(G_1)\times W(G_2)\setminus\{\{G_1\},\{G_2\}\} \tag{311}$$

with × denoting the Cartesian product of two sets. This in turn implies

$$\Delta(G_1G_2) = -K\Bigg[\sum_{S_1\in W(G_1)}\sum_{S_2\in W(G_2)} \Delta(S_1S_2) * G_1G_2/S_1/S_2\ - \Delta(G_1)\Delta(G_2)\Bigg] \tag{312}$$

Assuming the induction hypothesis $\Delta(S_1S_2) = \Delta(S_1)\Delta(S_2)$ we then get

$$\begin{aligned}\Delta(G_1G_2) &= -K\Big[R(G_1)R(G_2) - \Delta(G_1)\Delta(G_2)\Big]\\ &= \Delta(G_1)\Delta(G_2)\,.\end{aligned} \tag{313}$$

B CUTVERTEX RULE FOR TENSOR DIAGRAMS

If weakly non-overlapping (no common edges) subgraphs γ_1 and γ_2 contain contracted Lorentz indices, one has in general

$$K\big(\Delta(\gamma_1)\Delta(\gamma_2)\big) \neq \Delta(\gamma_1)\Delta(\gamma_2)\,. \tag{314}$$

This means that the proof for the factorisation of the counterterm operation Δ given in appendix A breaks down. As a result, it is rather difficult to derive a corresponding generalised "cut-vertex rule" for the case of contracted tensor subgraphs that does not result in a change of renormalisation scheme. However, when one is interested only in computing the poles of a factorised Feynman graph G_1G_2 via the use of the identity

$$KG = -K\,\delta RG\,, \tag{315}$$

we will show that the following cutvertex rule still holds:

$$\Delta(G_1G_2) \to \Delta(G_1)\Delta(G_2)\,. \tag{316}$$

We can actually prove this statement rather easily by noting that the R-operation computed with eq. (316) results in the following replacement:

$$R(G_1G_2) \to R(G_1)R(G_2)\,. \tag{317}$$

We can now write

$$\delta R(G_1G_2) = R(G_1G_2) - G_1G_2 = R(G_1)R(G_2) - G_1G_2 + \xi\,, \tag{318}$$

where ξ denotes the "error" one makes by computing with eq. (316). From this it follows that

$$\xi = R(G_1G_2) - R(G_1)R(G_2)\,. \tag{319}$$

Given that ξ is manifestly finite, we obtain:

$$K\xi = 0 \Rightarrow K\delta R(G_1G_2) = KR(G_1)R(G_2) - KG_1G_2\,. \tag{320}$$

This completes the proof that the poles of a factorised graph can be computed by consistently applying eq. (316), even though the UV counterterm is in a different renormalisation scheme.

B

C

IR SUBGRAPH SEARCH

One question that remains is how to find all IR subgraphs. Since the IR graphs could be disconnected, it is not as straightforward as for the UV. Below we describe a method to find the complete IR spinney at once.

In section 5.1 the contracted IR subgraph $\tilde{\gamma}$ was defined by contracting the remaining graph (or quotient) graph $\bar{\gamma} = G \setminus \gamma'$ to a point in G, i.e.,

$$\tilde{\gamma} = G/\bar{\gamma}\,. \tag{321}$$

In fact this observation generalizes further to the case of IR spinneys S':

$$\tilde{S} = G/\bar{S}, \qquad \bar{S} = G \setminus S', \qquad \tilde{S} = \prod_i \tilde{\gamma}_i\,. \tag{322}$$

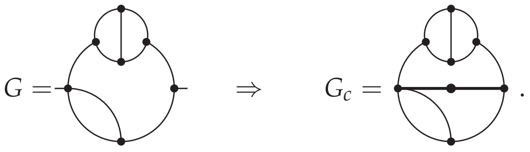
The different $\tilde{\gamma}_i$ are then only connected through cut-vertices in $\tilde{S}$. This dual description of contracted IR spinneys offers the possibility for an alternative IR search procedure by searching instead for valid remaining graphs. An easy identification of valid remaining graphs can be obtained from the contracted massless vacuum graph G_c of the graph G itself, which is defined by contracting in G all the external lines in a single vertex and contracting all massive lines into points.

All valid remaining graphs can then be identified with all spinneys of G_c, which include the formerly external lines. More precisely, we have the relation:

$$W'(G) = \{\tilde{S}\} = \{\, G_c/S | S \in W(G_c)\,,\, l_E(G) \subset S\,, \tilde{\omega}(G_c/S) \geq 0\,\}\,, \tag{323}$$

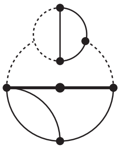
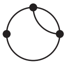
where $l_E(G)$ is the set of external lines of G. This allows one to construct a simple algorithm to find all IR spinneys by finding and combining 1PI subgraphs, similar to the construction of the UV spinney. A further advantage of this method is that disconnected IR subgraphs, such as the example we gave in eq.(124), are automatically included in this alternative search method.

It is instructive to see how this works in an example. Consider the following graph and its associated contracted vacuum graph:



(324)

Here we have indicated the contracted external lines in G_c with a thicker line. An example for a UV spinney in G_c and its associated IR spinney (in this case consisting of a single IR subgraph) is given by

$S =$  $\Rightarrow \qquad \tilde{S} = G_c/S =$  . (325)

Here we used dashed lines to indicate those lines not contained in the spinney S. These dashed lines become the IR spinney after shrinking the disconnected components of S to points in G_c.

SUMMARY

One of the primary goals of physics is to understand what the fundamental properties of Nature are. A highly successful tool to achieve this is Quantum Field Theory (QFT), which describes the interactions of indivisible particles. Built on the QFT framework, the Standard Model is the current best theory of the fundamental structure of Nature. Physics beyond the Standard Model will likely only lead to small deviations of the outcome of collisions in particle accelerators. Thus, very accurate predictions have to be made to detect new physics. This thesis will study the computational problems that arise when trying to improve precision.

In chapter 1 we define the research questions and the problem statement.

> **Problem statement:** *In what way can we improve the performance of QFT calculations with respect to obtaining more precise predictions?*

We focus on three computational and combinatorial problems of QFT calculations that we deemed the most urgent ones: (1) slowness of Monte Carlo integrations, (2) slowness in the computation of massless propagator integrals, and (3) slowness when computing the poles of Feynman diagrams.

PROBLEM 1 Monte Carlo methods are frequently used in QFT calculations. After the integrals are rewritten to a suitable polynomial representation, they are sampled millions of times. Some of these polynomials have more than a hundred thousand terms, making evaluation very slow. Thus, the first research question is as follows.

> **Research question 1:** *To what extent can the number of arithmetic operations of large multivariate polynomials be reduced?*

In chapter 2 we investigate ways to simplify expressions, using Horner schemes and Common Subexpression Elimination (CSEE). Our approach applies Monte Carlo Tree Search (MCTS), a search procedure that has been successful in AI. We use it to find near-optimal Horner schemes. Although MCTS finds good solutions, this approach gives rise to two further challenges. (1) MCTS (with UCT) introduces a constant, C_p that governs the balance between exploration and exploitation. This constant has to be tuned manually. (2) There should be more guided exploration at the bottom of the tree, since the current approach reduces the quality of the solution towards the end of the expression. To address both issues, we investigate NMCS (Nested Monte Carlo Search), but find that NMCS is computationally infeasible for our problem. Then, we modify the UCT formula by introducing a dynamic exploration-exploitation parameter T that decreases linearly with the iteration number. Consequently, we provide a performance analysis. We observe that a variable C_p solves the two problems: it yields more exploration at the bottom and as a result the tuning problem has been simplified. The region in C_p for which good values are found is increased by more than a tenfold.

Next, we consider Stochastic Local Search methods, since these methods do not have the problem of performing little optimisation in the order of the final variables. We investigate the state space properties of Horner schemes and find that the domain is relatively flat and contains only a few local minima. As a result, the Horner space is appropriate to be explored by Stochastic Hill Climbing (SHC), which has only two parameters: the number of iterations (computation time) and the neighbourhood structure. We find a suitable neighbourhood structure, leaving only the allowed computation time as a parameter. We perform a range of experiments. The results obtained by SHC are similar or better than those obtained by MCTS, which means that the number of operations is at least an order of magnitude smaller than the input. Furthermore, we show that SHC obtains the good results at least 10 times faster. Since the evaluation time of Monte Carlo integrators is proportional to the number of operations, their performance is improved.

PROBLEM 2 Most integrals that can be computed analytically, are calculated by using Integration by Parts (IBP) identities to express integrals into simpler ones. This method is generally quite slow and often requires months of computation time on a cluster. Hence, our second research question reads as follows.

> **Research question 2:** *How can we construct a program that can compute four-loop massless propagator integrals more efficiently?*

In chapter 3, we explain the construction of Forcer, a Form program for the reduction of four-loop massless propagator-type integrals to master integrals. The resulting program performs parametric IBP reductions similar to the three-loop Mincer program. We show how one can solve many systems of IBP identities parametrically in a computer-assisted manner. Next, we discuss the structure of the Forcer program, which involves recognizing reduction actions for each topology, applying symmetries, and transitioning between topologies after edges have been removed. This part is entirely precomputed and automatically generated. We give examples of recent applications of Forcer, and study the performance of the program. We show that the four-loop beta function can be computed in three minutes on a 32-core machine. Finally, we demonstrate how to use the Forcer package and sketch how to prepare physical diagrams for evaluation by Forcer.

In chapter 4 we have computed the self-energies and a set of three-particle vertex functions for massless QCD at the four-loop level in the $\overline{\text{MS}}$ renormalisation scheme, using the Forcer program. The vertex functions are evaluated at points where one of the momenta vanishes. Analytical results are obtained for a generic gauge group and with the full gauge dependence, which was made possible by extensive use of the Forcer program for massless four-loop propagator integrals. The bare results in dimensional regularisation are provided in terms of master integrals and rational coefficients; the latter are exact in any space-time dimension. Our results can be used for further precision investigations of the perturbative behaviour of the theory in schemes other than $\overline{\text{MS}}$.

Additionally, we compute Mellin moments of four-loop splitting functions and coefficient functions. These are used as basic ingredients for collision processes, such as Higgs production. We compute Mellin moments $N = 2, 4, 6$ for the non-singlet case and $N = 2, 4$ for the singlet case. Furthermore, we calculate $N = 1, 3, 5$ of vector-axial interference F_3. By computing to $N = 40$ and beyond, we reconstruct the all-N n_f^2 contribution to the four-loop non-singlet splitting function and the n_f^3 contribution to the four-loop singlet splitting function. Using the OPE method, we calculate up to $N = 16$ for the non-singlet splitting function. For the large-n_c limit, we compute up to $N = 19$. This allows for an all-N reconstruction and yields a new term to the four-loop planar cusp anomalous dimension.

PROBLEM 3 For five-loop calculations in QCD, so far only the poles of integrals have been computed, as the finite pieces are too difficult. One way to compute the poles is with the R^*-operation, which is complicated and quite slow. Hence, we formulate our third research question as follows.

> **Research question 3:** *To what extent can we compute the poles of five-loop diagrams using only four-loop diagrams more efficiently?*

In chapter 5 we extend the R^*-operation to Feynman graphs with arbitrary numerators, including tensors. We also provide a novel way of defining infrared counterterms which closely resembles the definition of its ultraviolet counterpart. We further express both infrared and ultraviolet counterterms in terms of scaleless vacuum graphs with a logarithmic degree of divergence. By exploiting symmetries, integrand and integral relations, which the counterterms of scaleless vacuum graphs satisfy, we can vastly reduce their number and complexity. A FORM implementation of this method was used to compute the poles in the dimensional regulator of all top-level propagator graphs at five loops in four dimensional ϕ^3 theory.

In chapter 6 we compute the five-loop corrections to the scale dependence of the renormalised coupling constant (the beta function) for QCD, its generalisation to non-Abelian gauge theories with a simple compact Lie group, and for QED. Our analytical result, obtained using the background field method, infrared rearrangement via the new diagram-by-diagram implementation of the R^*-operation and the FORCER program for massless four-loop propagators, confirms the QCD and QED results obtained by only one group before. The numerical size of the five-loop corrections is briefly discussed in the standard $\overline{\text{MS}}$ scheme for QCD with n_f flavours and for pure SU(N) Yang-Mills theory. Their effect in QCD is much smaller than the four-loop contributions, even at rather low scales. Additionally, we derive the five-loop beta function in a relatively common alternative, the minimal momentum subtraction (MiniMOM) scheme using the propagators and vertices computed in chapter 4. The computation of the five-loop beta function took six days on a 32-core machine.

CONCLUSION Finally, in chapter 7 we again consider the problem statement and research questions. Based on our findings, we may conclude that we have improved the performance of QFT calculations in three different regions. Since all our methods can be applied in practice to compare theory to experiment in colliders, we conclude that we have improved the precision of predictions.

SAMENVATTING

Een van de hoofddoelen van natuurkunde is begrijpen wat de fundamentele eigenschappen van de natuur zijn. Een zeer succesvol raamwerk hiervoor is Quantum Veldentheorie (QFT), dat de interacties tussen ondeelbare deeltjes beschrijft. Het Standaard Model, gebouwd op QFT, is de beste theorie over de fundamentele structuur van de natuur. Natuurkunde buiten het Standaard Model leidt waarschijnlijk tot zeer kleine afwijkingen van de uitkomst van botsingsexperimenten in deeltjesversnellers. Vandaar dat zeer precieze voorspellingen nodig zijn om nieuwe fysica te detecteren. Deze thesis zal de computationele problemen bestuderen die ontstaan bij het verbeteren van precisie.

In hoofdstuk 1 definiëren we de onderzoeksvragen en de probleemstelling.

> **Probleemstelling:** Op welke manier kunnen we de prestaties van QFT-berekeningen met betrekking tot het verkrijgen van preciezere voorspellingen verbeteren?

We richten ons op drie computationele en combinatorische problemen van QFT-berekeningen die wij het urgentst achten: (1) traagheid van Monte Carlo-integraties, (2) traagheid van het berekenen van massaloze propagatorintegralen en (3) traagheid van het berekenen van de polen van Feynman-diagrammen.

PROBLEEM 1 Monte Carlo-methodes worden vaak gebruikt in QFT-berekeningen. Nadat de integralen omgeschreven zijn in een geschikte polynomiale representatie, worden ze miljoenen keren geëvalueerd. Sommige van deze polynomen hebben meer dan honderdduizend termen, waardoor de evaluaties erg traag zijn. Dus, de eerste onderzoeksvraag luidt als volgt.

> **Onderzoeksvraag 1:** Tot hoever kan het aantal operaties van grote multivariabele polynomen gereduceerd worden?

In hoofdstuk 2 onderzoeken we manieren om uitdrukkingen te vereenvoudigen met behulp van Horner-schema’s en verwijdering van vaker voorkomende subuitdrukkingen (CSEE). Onze methode gebruikt Monte Carlo Boom Zoeken (MCTS), een zoekprocedure die heel succesvol is in kunstmatige intelligentie. We gebruiken MCTS om bijna-optimale Horner-schema’s te vinden. Ondanks het feit dat MCTS goede oplossingen vindt, heeft deze benadering twee uitdagingen. (1) MCTS (met UCT) introduceert een constante, C_p, die de balans tussen exploratie en exploitatie regelt. Deze constante moet handmatig afgestemd worden. (2) Er moet meer gerichte exploratie aan het einde van de boom zijn, aangezien de huidige methode de kwaliteit van de oplossingen negatief beïnvloedt. Om beide problemen aan te pakken, bestuderen we Genest Monte Carlo Zoeken (NMCS), maar we zien dat NMC computationeel onhaalbaar is voor ons probleem. Vervolgens passen we de UCT-formule aan door de introductie van een dynamische exploratie-exploitatie parameter T, die lineair afneemt met het iteratienummer. Daarna meten we het effect op de prestaties. We zien dat een variabele C_p de twee problemen oplost: er is meer exploratie aan het eind van de boom en het afstemmingsprobleem versimpelt. Het gebied in C_p waarin goede waarden gevonden worden, is met meer dan een factor tien vergroot.

Vervolgens bestuderen we Stochatische Lokale Zoekmethodes, aangezien deze methodes het probleem van te weinig optimalisatie voor de laatste variabelen in het schema helemaal niet hebben. We onderzoeken de faseruimte-eigenschappen van Horner-schema's en zien dat het domein relatief plat is en dat er slechts enkele lokale minima zijn. Als gevolg kan Stochastisch Heuvel Klimmen (SHC) de Horner-ruimte goed verkennen. SHC heeft slechts twee parameters: het aantal iteraties (rekentijd) en de burenstructuur. We vinden een geschikte burenstructuur, waardoor alleen de totale rekentijd als parameter overblijft. Hierna doen we een reeks experimenten. De resultaten die we met SHC behalen zijn gelijkwaardig of beter dan die van MCTS, wat inhoudt dat het totale aantal operaties met meer dan een orde van grootte kleiner is dan het origineel. Bovendien behaalt SHC de resultaten zeker tien keer sneller. Aangezien de evaluatietijd van Monte Carlo integrators proportioneel met het aantal operaties is, hebben we de prestaties verbeterd.

PROBLEEM 2 De meeste integralen die analytisch berekend worden, worden uitgerekend met de hulp van Partiële Integratie (IBP) identiteiten die integralen in simpelere varianten uitdrukken. Deze methode is erg langzaam in het algemeen en vergt soms maanden rekentijd op een cluster. Dus, onze tweede onderzoeksvraag luidt als volgt.

> **Onderzoeksvraag 2:** Hoe kunnen we een programma construeren dat vier-lus massaloze propagatorintegralen efficiënter berekent?

In hoofdstuk 3 presenteren we de constructie van FORCER, een FORM-programma voor de reductie van vier-lus massaloze propagator-type integralen tot master integralen. Het resulterende programma voert parametrische IBP-reducties uit, net zoals het drie-lus programma MINCER. We demonstreren hoe men veel systemen van IBP-identiteiten parametrisch kan oplossen in een computer-geassisteerde manier. Vervolgens bespreken we de structuur van het FORCER-programma, zoals het herkennen van reductie-acties voor iedere topologie, het toepassen van symmetrieën, en de overgang tussen topologieën als er een lijn verwijderd wordt. Dit gedeelte is volledig vooraf uitgerekend en automatisch gegenereerd. We geven voorbeelden van recente toepassingen van FORCER en bestuderen de prestaties van het programma. We laten zien dat de vier-lus beta-functie in drie minuten berekend kan worden op een 32-kernen machine. Tot slot demonstreren we hoe het FORCER-pakket gebruikt kan worden en schetsen we hoe fysische diagrammen klaargemaakt kunnen worden voor evaluatie door FORCER.

In hoofdstuk 4 berekenen we zelfenergieën en een verzameling van drie-deeltjes-vertexfuncties voor massaloze QCD op vier lussen in het $\overline{\text{MS}}$ renormalisatieschema met het FORCER programma. De vertexfuncties worden geëvalueerd op het punt waar een van de momenta verdwijnt. We presenteren analytische resultaten voor een generieke ijkgroep met volledige ijkafhankelijkheid. De ongerenormaliseerde resultaten in dimensionele regularisatie worden gegeven in termen van masterintegralen en rationale coefficiënten, welke exact zijn in de ruimte-tijd-dimensie. Onze resultaten kunnen gebruikt worden voor verder precisie-onderzoek naar het storingsgedrag van de theorie in andere schema's dan $\overline{\text{MS}}$.

Daarnaast hebben we ook Mellin-momenten van vier-lus splitsingsfuncties en coefficiëntfuncties uitgerekend. Deze worden gebruikt als basisingrediënt voor botsingsprocessen, zoals Higgs-productie. We berekenen Mellin momenten $N = 2, 4, 6$ voor het niet-singlet geval en $N = 2, 4$ voor het singlet-geval. Bovendien berekenenen we $N = 1, 3, 5$ voor de vector-axiaal-interferentieterm F_3. Doordat we verder dan $N = 40$ hebben uitgerekend, kunnen we de alle-N n_f^2-bijdrage aan de vier-lus niet-singlet splitsingsfunctie en de n_f^3-bijdrage aan de vier-lus singlet splitsingsfunctie reconstrueren. Met behulp van de OPE-methode, berekenen we tot en met $N = 16$ voor de niet-singlet splitsingsfunctie uit. In de grote-n_c-limiet berekenen we tot en met $N = 19$. Dit stelt ons in staat om een alle-N-reconstructie te maken, die een nieuwe term oplevert in de vier-loop planaire spits anomale dimensie.

PROBLEEM 3 Voor vijf-lus berekeningen in QCD zijn tot nu toe alleen de polen van objecten berekend, omdat het bemachtigen van de eindige termen te moeilijk is. Een manier om polen te berekenen is met de R^*-operatie, die gecompliceerd en traag is. Dus, onze derde onderzoeksvraag luidt als volgt.

> **Onderzoeksvraag 3:** In hoeverre kunnen we de polen van vijf-lus diagrammen efficiënter uitrekenen met alleen maar vier-lus diagrammen?

In hoofdstuk 5 breiden we de R^*-operatie uit tot Feynman-diagrammen met willekeurige tellers, inclusief tensoren. We geven ook een nieuwe manier om infrarood-contratermen te definiëren die lijkt op de definitie van zijn ultraviolette tegenhanger. Verder drukken we zowel infrarode als ultraviolette contratermen uit in termen van schaalloze vacuümdiagrammen met logaritmische graad van divergentie. Door symmetrieën en integrand- en integraalrelaties uit te buiten, kunnen we het aantal contratermen en hun complexiteit terugbrengen. Een FORM-implementatie van deze methode is gebruikt bij de berekening van de polen in de dimensionele regulator van alle top-niveau propagatordiagrammen op vijf lussen in vierdimensionale ϕ^3-theorie.

In hoofdstuk 6 rekenen we de vijf-lus correctie van de schaalafhankelijkheid van de gerenormaliseerde koppelingsconstante uit (de beta functie) voor QCD, zijn generalisatie tot niet-Abelse ijktheorieën met een simpele compacte Lie-groep, en voor QED. Onze analytische resultaten, bemachtigd met de achtergrondveldmethode, infrarode verwisseling via de nieuwe diagram-per-diagram-implementatie van de R^*-operatie en het FORCER-programma voor massaloze vier-lus propagatoren, bevestigen de QCD en QED resultaten die slechts door één andere groep uitgerekend zin. We bekijken de numerieke grootte van de vijf-lus correcties in het standaard

$\overline{\text{MS}}$-schema voor QCD met n_f quarksmaken en voor pure SU(N) Yang-Mills-theorie. Het effect in QCD is veel kleiner dan de vier-lus bijdrage, zelfs voor lage schalen. Verder leiden we de vijf-lus betafunctie af in een relatief vaak voorkomend alternatief schema, het minimale momentum subtractieschema (MiniMOM) met behulp van de propagatoren en vertices berekend in hoofdstuk 4. De berekening van de vijf-lus betafunctie kostte zes dagen op een 32-kernen machine.

CONCLUSIE Tot slot staan we opnieuw stil bij de probleemstelling en onderzoeksvragen in hoofdstuk 7. Gebaseerd op onze vondsten, concluderen we dat we de prestaties van QFT-berekeningen hebben verbeterd in drie verschillende gebieden. Aangezien al onze methodes in de praktijk toegepast kunnen worden om theorie te vergelijken met het experiment in deeltjesversnellers, concluderen we dat we de precisie van voorspellingen verbeterd hebben.

CURRICULUM VITAE

Ben Ruijl was born in Geleen, the Netherlands on 9 November 1989. He graduated cum laude from the Graaf Huyn College in 2008. Pursuing deeper knowledge of physics, he obtained his Bachelor degree in Physics and Astronomy in 2011 from the Radboud University Nijmegen. He graduated cum laude with a Master's degree in Theoretical Particle Physics in 2013 from Radboud University Nijmegen.

Ben's interests are not limited to physics. He spent a considerable amount of his spare and professional time programming and solving combinatorial problems. A PhD research position in the project HEPGAME, which tries to apply artificial intelligence methods to problems in high energy physics, presented an ideal match. In 2013 Ben joined the project. The PhD research was conducted at Nikhef Amsterdam, and the Leiden Institute for Advanced Computer Science of Leiden University.

ACKNOWLEDGEMENTS

This thesis would not have been the same without the support of the people listed below.

I wish to thank my close collaborators Takahiro Ueda, Franz Herzog, Andreas Vogt, and Ali Mirsoleimani. During the many discussions we have had, you have taught me a lot. Not only about physics, or computer science, but also about being a good researcher. Thank you for such a pleasant collaboration!

I have had the pleasure to be active at Tilburg University, Leiden University, and Nikhef. In all three places I have felt at home due to the warmth of my colleagues. I would like to thank my co-workers at Leiden: Jaap, Aske, Ali, Ramin, Bilal, Hafeez, Dave, Joke, Walter, Maarten, Edwin, Nanda, and Frank for our nice conversations. Next, a big thank you to all the people at Nikhef: Jos, Giulio, Gillian, Tom, Jorinde, Remco, Sabrina, Robert, Robbert, Jan-Willem, Eric, Piet, Marieke, Bert, Lisa, Thomas, Jonathan, Jordy, Eleni, Elena, Lorenzo, Andreas, Satish, Juan, Jort, Ruben, Michael, Jacopo, Wouter, Elisa, Mathias, Gilberto, Domenico, Andrea, Maarten, and Giuseppe. I have had a great time!

Of course, the last four years would not have been the same without friends. I would like to thank Jeff, Dorien, Gijs, Farah for the fantastic film nights and endless banter. Fiona, thanks for the laughter and support! Next, thank you to my games group and study buddies: Gillian, Roel, Daan, Jorn, Erik, Jan-Willem, Maarten, Remco, Sanne, and Elze. Thank you Marijn and Pim for all the fun nights with good food! A big shout-out to my group of high-school friends (the NT-ers): Jeroen, Vince, Lars, Mariska, Roel, Evelien, Lieke, Martijn, Niels, Martijn, Rob, Michel, Ellen, Bart, Anne, Lauraine for all the fun we had in the last 15 years. Finally, thank you to Anna-Sophie for the time we spent together.

Daarnaast wil ik ook mijn familie bedanken voor de steun en het vertrouwen tijdens de afgelopen vier jaar. Pap, Mam, Paul, bedankt! En natuurlijk bedank ik opa Jeff, die zo graag bij de verdediging had willen zijn.

Thank you all for your kindness and support!

SIKS DISSERTATION SERIES

2011 01 Botond Cseke (RUN), Variational Algorithms for Bayesian Inference in Latent Gaussian Models

02 Nick Tinnemeier (UU), Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language

03 Jan Martijn van der Werf (TUE), Compositional Design and Verification of Component-Based Information Systems

04 Hado van Hasselt (UU), Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference

05 Bas van der Raadt (VU), Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.

06 Yiwen Wang (TUE), Semantically-Enhanced Recommendations in Cultural Heritage

07 Yujia Cao (UT), Multimodal Information Presentation for High Load Human Computer Interaction

08 Nieske Vergunst (UU), BDI-based Generation of Robust Task-Oriented Dialogues

09 Tim de Jong (OU), Contextualised Mobile Media for Learning

10 Bart Bogaert (UvT), Cloud Content Contention

11 Dhaval Vyas (UT), Designing for Awareness: An Experience-focused HCI Perspective

12 Carmen Bratosin (TUE), Grid Architecture for Distributed Process Mining

13 Xiaoyu Mao (UvT), Airport under Control. Multiagent Scheduling for Airport Ground Handling

14 Milan Lovric (EUR), Behavioral Finance and Agent-Based Artificial Markets

15 Marijn Koolen (UvA), The Meaning of Structure: the Value of Link Evidence for Information Retrieval

16 Maarten Schadd (UM), Selective Search in Games of Different Complexity

17 Jiyin He (UVA), Exploring Topic Structure: Coherence, Diversity and Relatedness

18 Mark Ponsen (UM), Strategic Decision-Making in complex games

19 Ellen Rusman (OU), The Mind's Eye on Personal Profiles

20 Qing Gu (VU), Guiding service-oriented software engineering - A view-based approach

21 Linda Terlouw (TUD), Modularization and Specification of Service-Oriented Systems

22 Junte Zhang (UVA), System Evaluation of Archival Description and Access

23 Wouter Weerkamp (UVA), Finding People and their Utterances in Social Media

24 Herwin van Welbergen (UT), Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior

25 Syed Waqar ul Qounain Jaffry (VU), Analysis and Validation of Models for Trust Dynamics

26 Matthijs Aart Pontier (VU), Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots

27 Aniel Bhulai (VU), Dynamic website optimisation through autonomous management of design patterns

28 Rianne Kaptein (UVA), Effective Focused Retrieval by Exploiting Query Context and Document Structure

29 Faisal Kamiran (TUE), Discrimination-aware Classification

30 Egon van den Broek (UT), Affective Signal Processing (ASP): Unraveling the mystery of emotions

31 Ludo Waltman (EUR), Computational and Game-Theoretic Approaches for Modeling Bounded Rationality

32 Nees-Jan van Eck (EUR), Methodological Advances in Bibliometric Mapping of Science

33 Tom van der Weide (UU), Arguing to Motivate Decisions

34 Paolo Turrini (UU), Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations

35 Maaike Harbers (UU), Explaining Agent Behavior in Virtual Training

36 Erik van der Spek (UU), Experiments in serious game design: a cognitive approach

37 Adriana Burlutiu (RUN), Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference

38 Nyree Lemmens (UM), Bee-inspired Distributed Optimization

39 Joost Westra (UU), Organizing Adaptation using Agents in Serious Games

40 Viktor Clerc (VU), Architectural Knowledge Management in Global Software Development

41 Luan Ibraimi (UT), Cryptographically Enforced Distributed Data Access Control

42 Michal Sindlar (UU), Explaining Behavior through Mental State Attribution

43 Henk van der Schuur (UU), Process Improvement through Software Operation Knowledge

44 Boris Reuderink (UT), Robust Brain-Computer Interfaces

45 Herman Stehouwer (UvT), Statistical Language Models for Alternative Sequence Selection

46 Beibei Hu (TUD), Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work

47 Azizi Bin Ab Aziz (VU), Exploring Computational Models for Intelligent Support of Persons with Depression

48 Mark Ter Maat (UT), Response Selection and Turn-taking for a Sensitive Artificial Listening Agent

49 Andreea Niculescu (UT), Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

---

2012 01 Terry Kakeeto (UvT), Relationship Marketing for SMEs in Uganda

02 Muhammad Umair (VU), Adaptivity, emotion, and Rationality in Human and Ambient Agent Models

03 Adam Vanya (VU), Supporting Architecture Evolution by Mining Software Repositories

04 Jurriaan Souer (UU), Development of Content Management System-based Web Applications

05 Marijn Plomp (UU), Maturing Interorganisational Information Systems

06 Wolfgang Reinhardt (OU), Awareness Support for Knowledge Workers in Research Networks

07 Rianne van Lambalgen (VU), When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions

08 Gerben de Vries (UVA), Kernel Methods for Vessel Trajectories

09 Ricardo Neisse (UT), Trust and Privacy Management Support for Context-Aware Service Platforms

10 David Smits (TUE), Towards a Generic Distributed Adaptive Hypermedia Environment

11 J.C.B. Rantham Prabhakara (TUE), Process Mining in the Large: Preprocessing, Discovery, and Diagnostics
12 Kees van der Sluijs (TUE), Model Driven Design and Data Integration in Semantic Web Information Systems
13 Suleman Shahid (UvT), Fun and Face: Exploring non-verbal expressions of emotion during playful interactions
14 Evgeny Knutov (TUE), Generic Adaptation Framework for Unifying Adaptive Web-based Systems
15 Natalie van der Wal (VU), Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.
16 Fiemke Both (VU), Helping people by understanding them - Ambient Agents supporting task execution and depression treatment
17 Amal Elgammal (UvT), Towards a Comprehensive Framework for Business Process Compliance
18 Eltjo Poort (VU), Improving Solution Architecting Practices
19 Helen Schonenberg (TUE), What's Next? Operational Support for Business Process Execution
20 Ali Bahramisharif (RUN), Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing
21 Roberto Cornacchia (TUD), Querying Sparse Matrices for Information Retrieval
22 Thijs Vis (UvT), Intelligence, politie en veiligheidsdienst: verenigbare grootheden?
23 Christian Muehl (UT), Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction
24 Laurens van der Werff (UT), Evaluation of Noisy Transcripts for Spoken Document Retrieval
25 Silja Eckartz (UT), Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application
26 Emile de Maat (UVA), Making Sense of Legal Text
27 Hayrettin Gurkok (UT), Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games
28 Nancy Pascall (UvT), Engendering Technology Empowering Women
29 Almer Tigelaar (UT), Peer-to-Peer Information Retrieval
30 Alina Pommeranz (TUD), Designing Human-Centered Systems for Reflective Decision Making
31 Emily Bagarukayo (RUN), A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure
32 Wietske Visser (TUD), Qualitative multi-criteria preference representation and reasoning
33 Rory Sie (OUN), Coalitions in Cooperation Networks (COCOON)
34 Pavol Jancura (RUN), Evolutionary analysis in PPI networks and applications
35 Evert Haasdijk (VU), Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics
36 Denis Ssebugwawo (RUN), Analysis and Evaluation of Collaborative Modeling Processes
37 Agnes Nakakawa (RUN), A Collaboration Process for Enterprise Architecture Creation
38 Selmar Smit (VU), Parameter Tuning and Scientific Testing in Evolutionary Algorithms
39 Hassan Fatemi (UT), Risk-aware design of value and coordination networks
40 Agus Gunawan (UvT), Information Access for SMEs in Indonesia

41 Sebastian Kelle (OU), Game Design Patterns for Learning

42 Dominique Verpoorten (OU), Reflection Amplifiers in self-regulated Learning

43 Withdrawn

44 Anna Tordai (VU), On Combining Alignment Techniques

45 Benedikt Kratz (UvT), A Model and Language for Business-aware Transactions

46 Simon Carter (UVA), Exploration and Exploitation of Multilingual Data for Statistical Machine Translation

47 Manos Tsagkias (UVA), Mining Social Media: Tracking Content and Predicting Behavior

48 Jorn Bakker (TUE), Handling Abrupt Changes in Evolving Time-series Data

49 Michael Kaisers (UM), Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions

50 Steven van Kervel (TUD), Ontologogy driven Enterprise Information Systems Engineering

51 Jeroen de Jong (TUD), Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

---

2013 01 Viorel Milea (EUR), News Analytics for Financial Decision Support

02 Erietta Liarou (CWI), MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing

03 Szymon Klarman (VU), Reasoning with Contexts in Description Logics

04 Chetan Yadati (TUD), Coordinating autonomous planning and scheduling

05 Dulce Pumareja (UT), Groupware Requirements Evolutions Patterns

06 Romulo Goncalves (CWI), The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience

07 Giel van Lankveld (UvT), Quantifying Individual Player Differences

08 Robbert-Jan Merk (VU), Making enemies: cognitive modeling for opponent agents in fighter pilot simulators

09 Fabio Gori (RUN), Metagenomic Data Analysis: Computational Methods and Applications

10 Jeewanie Jayasinghe Arachchige (UvT), A Unified Modeling Framework for Service Design.

11 Evangelos Pournaras (TUD), Multi-level Reconfigurable Self-organization in Overlay Services

12 Marian Razavian (VU), Knowledge-driven Migration to Services

13 Mohammad Safiri (UT), Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly

14 Jafar Tanha (UVA), Ensemble Approaches to Semi-Supervised Learning Learning

15 Daniel Hennes (UM), Multiagent Learning - Dynamic Games and Applications

16 Eric Kok (UU), Exploring the practical benefits of argumentation in multi-agent deliberation

17 Koen Kok (VU), The PowerMatcher: Smart Coordination for the Smart Electricity Grid

18 Jeroen Janssens (UvT), Outlier Selection and One-Class Classification

19 Renze Steenhuizen (TUD), Coordinated Multi-Agent Planning and Scheduling

20 Katja Hofmann (UvA), Fast and Reliable Online Learning to Rank for Information Retrieval

21 Sander Wubben (UvT), Text-to-text generation by monolingual machine translation

22 Tom Claassen (RUN), Causal Discovery and Logic

23 Patricio de Alencar Silva (UvT), Value Activity Monitoring

24 Haitham Bou Ammar (UM), Automated Transfer in Reinforcement Learning

25 Agnieszka Anna Latoszek-Berendsen (UM), Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System

26 Alireza Zarghami (UT), Architectural Support for Dynamic Homecare Service Provisioning

27 Mohammad Huq (UT), Inference-based Framework Managing Data Provenance

28 Frans van der Sluis (UT), When Complexity becomes Interesting: An Inquiry into the Information eXperience

29 Iwan de Kok (UT), Listening Heads

30 Joyce Nakatumba (TUE), Resource-Aware Business Process Management: Analysis and Support

31 Dinh Khoa Nguyen (UvT), Blueprint Model and Language for Engineering Cloud Applications

32 Kamakshi Rajagopal (OUN), Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development

33 Qi Gao (TUD), User Modeling and Personalization in the Microblogging Sphere

34 Kien Tjin-Kam-Jet (UT), Distributed Deep Web Search

35 Abdallah El Ali (UvA), Minimal Mobile Human Computer Interaction

36 Than Lam Hoang (TUe), Pattern Mining in Data Streams

37 Dirk Börner (OUN), Ambient Learning Displays

38 Eelco den Heijer (VU), Autonomous Evolutionary Art

39 Joop de Jong (TUD), A Method for Enterprise Ontology based Design of Enterprise Information Systems

40 Pim Nijssen (UM), Monte-Carlo Tree Search for Multi-Player Games

41 Jochem Liem (UVA), Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning

42 Léon Planken (TUD), Algorithms for Simple Temporal Reasoning

43 Marc Bron (UVA), Exploration and Contextualization through Interaction and Concepts

---

2014 01 Nicola Barile (UU), Studies in Learning Monotone Models from Data

02 Fiona Tuliyano (RUN), Combining System Dynamics with a Domain Modeling Method

03 Sergio Raul Duarte Torres (UT), Information Retrieval for Children: Search Behavior and Solutions

04 Hanna Jochmann-Mannak (UT), Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation

05 Jurriaan van Reijsen (UU), Knowledge Perspectives on Advancing Dynamic Capability

06 Damian Tamburri (VU), Supporting Networked Software Development

07 Arya Adriansyah (TUE), Aligning Observed and Modeled Behavior

08 Samur Araujo (TUD), Data Integration over Distributed and Heterogeneous Data Endpoints

09 Philip Jackson (UvT), Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language

10 Ivan Salvador Razo Zapata (VU), Service Value Networks

11 Janneke van der Zwaan (TUD), An Empathic Virtual Buddy for Social Support

12 Willem van Willigen (VU), Look Ma, No Hands: Aspects of Autonomous Vehicle Control

13 Arlette van Wissen (VU), Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains
14 Yangyang Shi (TUD), Language Models With Meta-information
15 Natalya Mogles (VU), Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare
16 Krystyna Milian (VU), Supporting trial recruitment and design by automatically interpreting eligibility criteria
17 Kathrin Dentler (VU), Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability
18 Mattijs Ghijsen (UVA), Methods and Models for the Design and Study of Dynamic Agent Organizations
19 Vinicius Ramos (TUE), Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support
20 Mena Habib (UT), Named Entity Extraction and Disambiguation for Informal Text: The Missing Link
21 Kassidy Clark (TUD), Negotiation and Monitoring in Open Environments
22 Marieke Peeters (UU), Personalized Educational Games - Developing agent-supported scenario-based training
23 Eleftherios Sidirourgos (UvA/CWI), Space Efficient Indexes for the Big Data Era
24 Davide Ceolin (VU), Trusting Semi-structured Web Data
25 Martijn Lappenschaar (RUN), New network models for the analysis of disease interaction
26 Tim Baarslag (TUD), What to Bid and When to Stop
27 Rui Jorge Almeida (EUR), Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty
28 Anna Chmielowiec (VU), Decentralized k-Clique Matching
29 Jaap Kabbedijk (UU), Variability in Multi-Tenant Enterprise Software
30 Peter de Cock (UvT), Anticipating Criminal Behaviour
31 Leo van Moergestel (UU), Agent Technology in Agile Multiparallel Manufacturing and Product Support
32 Naser Ayat (UvA), On Entity Resolution in Probabilistic Data
33 Tesfa Tegegne (RUN), Service Discovery in eHealth
34 Christina Manteli (VU), The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems.
35 Joost van Ooijen (UU), Cognitive Agents in Virtual Worlds: A Middleware Design Approach
36 Joos Buijs (TUE), Flexible Evolutionary Algorithms for Mining Structured Process Models
37 Maral Dadvar (UT), Experts and Machines United Against Cyberbullying
38 Danny Plass-Oude Bos (UT), Making brain-computer interfaces better: improving usability through post-processing.
39 Jasmina Maric (UvT), Web Communities, Immigration, and Social Capital
40 Walter Omona (RUN), A Framework for Knowledge Management Using ICT in Higher Education
41 Frederic Hogenboom (EUR), Automated Detection of Financial Events in News Text
42 Carsten Eijckhof (CWI/TUD), Contextual Multidimensional Relevance Models
43 Kevin Vlaanderen (UU), Supporting Process Improvement using Method Increments

44 Paulien Meesters (UvT), Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden.
45 Birgit Schmitz (OUN), Mobile Games for Learning: A Pattern-Based Approach
46 Ke Tao (TUD), Social Web Data Analytics: Relevance, Redundancy, Diversity
47 Shangsong Liang (UVA), Fusion and Diversification in Information Retrieval

---

2015 01 Niels Netten (UvA), Machine Learning for Relevance of Information in Crisis Response
02 Faiza Bukhsh (UvT), Smart auditing: Innovative Compliance Checking in Customs Controls
03 Twan van Laarhoven (RUN), Machine learning for network data
04 Howard Spoelstra (OUN), Collaborations in Open Learning Environments
05 Christoph Bösch (UT), Cryptographically Enforced Search Pattern Hiding
06 Farideh Heidari (TUD), Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes
07 Maria-Hendrike Peetz (UvA), Time-Aware Online Reputation Analysis
08 Jie Jiang (TUD), Organizational Compliance: An agent-based model for designing and evaluating organizational interactions
09 Randy Klaassen (UT), HCI Perspectives on Behavior Change Support Systems
10 Henry Hermans (OUN), OpenU: design of an integrated system to support lifelong learning
11 Yongming Luo (TUE), Designing algorithms for big graph datasets: A study of computing bisimulation and joins
12 Julie M. Birkholz (VU), Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks
13 Giuseppe Procaccianti (VU), Energy-Efficient Software
14 Bart van Straalen (UT), A cognitive approach to modeling bad news conversations
15 Klaas Andries de Graaf (VU), Ontology-based Software Architecture Documentation
16 Changyun Wei (UT), Cognitive Coordination for Cooperative Multi-Robot Teamwork
17 André van Cleeff (UT), Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs
18 Holger Pirk (CWI), Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories
19 Bernardo Tabuenca (OUN), Ubiquitous Technology for Lifelong Learners
20 Lois Vanhée (UU), Using Culture and Values to Support Flexible Coordination
21 Sibren Fetter (OUN), Using Peer-Support to Expand and Stabilize Online Learning
22 Zhemin Zhu (UT), Co-occurrence Rate Networks
23 Luit Gazendam (VU), Cataloguer Support in Cultural Heritage
24 Richard Berendsen (UVA), Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation
25 Steven Woudenberg (UU), Bayesian Tools for Early Disease Detection
26 Alexander Hogenboom (EUR), Sentiment Analysis of Text Guided by Semantics and Structure
27 Sándor Héman (CWI), Updating compressed colomn stores
28 Janet Bagorogoza (TiU), Knowledge Management and High Performance; The Uganda Financial Institutions Model for HPO
29 Hendrik Baier (UM), Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains
30 Kiavash Bahreini (OU), Real-time Multimodal Emotion Recognition in E-Learning

31 Yakup Koç (TUD), On the robustness of Power Grids

32 Jerome Gard (UL), Corporate Venture Management in SMEs

33 Frederik Schadd (TUD), Ontology Mapping with Auxiliary Resources

34 Victor de Graaf (UT), Gesocial Recommender Systems

35 Jungxao Xu (TUD), Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction

---

2016 01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines

02 Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow

03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support

04 Laurens Rietveld (VU), Publishing and Consuming Linked Data

05 Evgeny Sherkhonov (UVA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers

06 Michel Wilson (TUD), Robust scheduling in an uncertain environment

07 Jeroen de Man (VU), Measuring and modeling negative emotions for virtual training

08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data

09 Archana Nottamkandath (VU), Trusting Crowdsourced Information on Cultural Artefacts

10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms

11 Anne Schuth (UVA), Search Engines that Learn from Their Users

12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems

13 Nana Baah Gyan (VU), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach

14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization

15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments

16 Guangliang Li (UVA), Socially Intelligent Autonomous Agents that Learn from Human Reward

17 Berend Weel (VU), Towards Embodied Evolution of Robot Organisms

18 Albert Meroño Peñuela (VU), Refining Statistical Data on the Web

19 Julia Efremova (Tu/e), Mining Social Structures from Genealogical Data

20 Daan Odijk (UVA), Context & Semantics in News & Web Search

21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground

22 Grace Lewis (VU), Software Architecture Strategies for Cyber-Foraging Systems

23 Fei Cai (UVA), Query Auto Completion in Information Retrieval

24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach

25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior

26 Dilhan Thilakarathne (VU), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains

27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media

28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control

29 Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning

30 Ruud Mattheij (UvT), The Eyes Have It

31 Mohammad Khelghati (UT), Deep web content monitoring

32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations

33 Peter Bloem (UVA), Single Sample Statistics, exercises in learning from just one example

34 Dennis Schunselaar (TUE), Configurable Process Trees: Elicitation, Analysis, and Enactment

35 Zhaochun Ren (UVA), Monitoring Social Media: Summarization, Classification and Recommendation

36 Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies

37 Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry

38 Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design

39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect

40 Christian Detweiler (TUD), Accounting for Values in Design

41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance

42 Spyros Martzoukos (UVA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora

43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice

44 Thibault Sellam (UVA), Automatic Assistants for Database Exploration

45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control

46 Jorge Gallego Perez (UT), Robots to Make you Happy

47 Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks

48 Tanja Buttler (TUD), Collecting Lessons Learned

49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis

50 Yan Wang (UVT), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

---

2017 01 Jan-Jaap Oerlemans (UL), Investigating Cybercrime

02 Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation

03 Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines

04 Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store

05 Mahdieh Shadi (UVA), Collaboration Behavior

06 Damir Vandic (EUR), Intelligent Information Systems for Web Product Search

07 Roel Bertens (UU), Insight in Information: from Abstract to Anomaly

08 Rob Konijn (VU) , Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery

09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text

10 Robby van Delden (UT), (Steering) Interactive Play Behavior

11 Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment

12 Sander Leemans (TUE), Robust Process Mining with Guarantees

13 Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology

14 Shoshannah Tekofsky (UvT), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior

15 Peter Berck (RUN), Memory-Based Text Correction

16 Aleksandr Chuklin (UVA), Understanding and Modeling Users of Modern Search Engines

17 Daniel Dimov (UL), Crowdsourced Online Dispute Resolution

18 Ridho Reinanda (UVA), Entity Associations for Search

19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval

20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility

21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)

22 Sara Magliacane (VU), Logics for causal inference under uncertainty

23 David Graus (UVA), Entities of Interest — Discovery in Digital Traces

24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning

25 Veruska Zamborlini (VU), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search

26 Merel Jung (UT), Socially intelligent robots that understand and respond to human touch

27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors

28 John Klein (VU), Architecture Practices for Complex Contexts

29 Adel Alhuraibi (UvT), From IT-Business Strategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT

30 Wilma Latuny (UvT), The Power of Facial Expressions