Two-Loop Amplitudes for Processes Involving Photons at Hadron Colliders

Dissertation

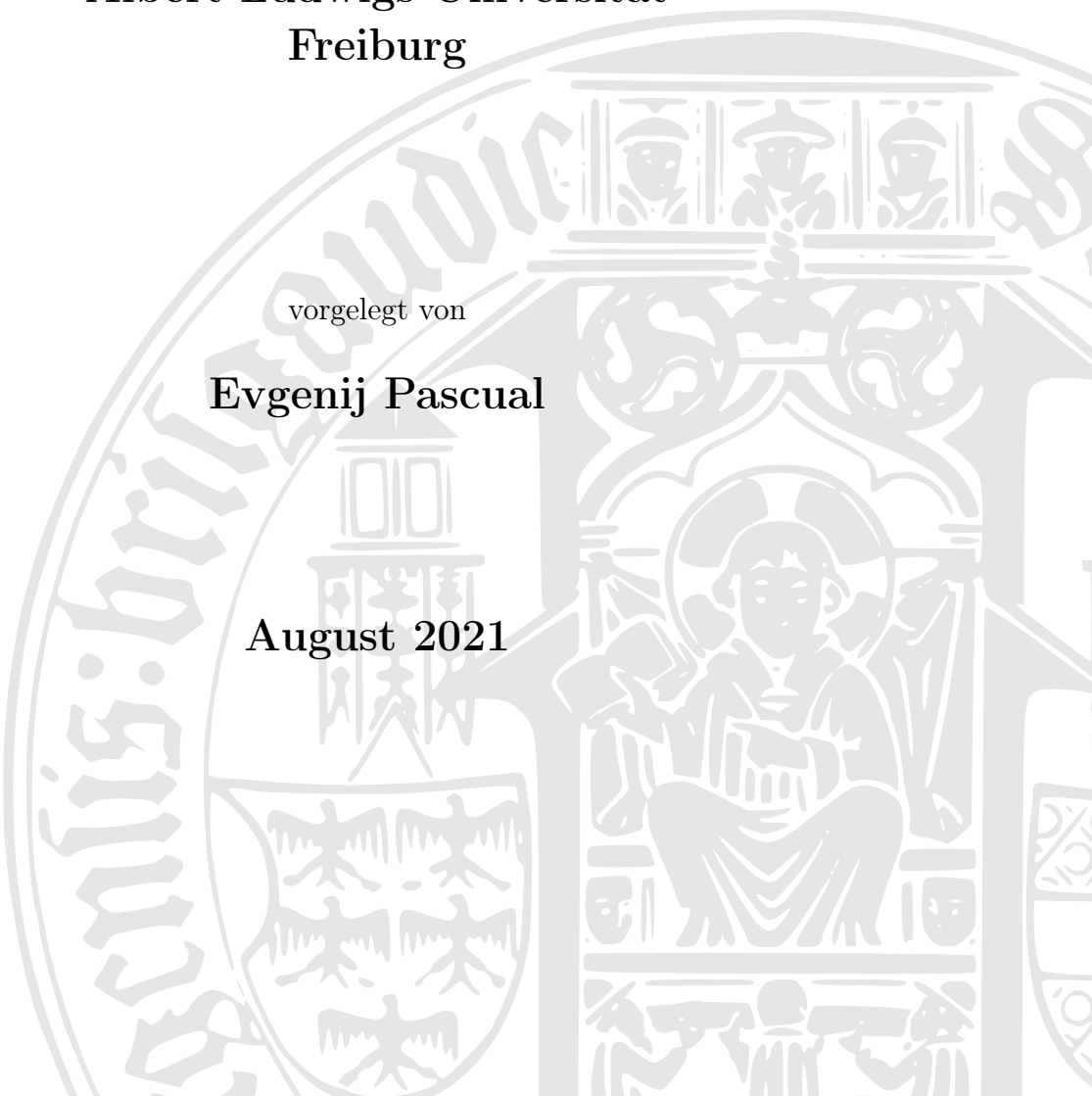
zur Erlangung des Doktorgrades
der Fakultät für Mathematik und Physik der

Albert-Ludwigs-Universität Freiburg

vorgelegt von

Evgenij Pascual

August 2021



Dekan: Prof. Dr. Wolfgang Soergel
Referent: Prof. Dr. Fernando Febres Cordero
Koreferent: Prof. Dr. Stefan Dittmaier
Tag der mündlichen Prüfung: 29.10.2021

Abstract

Studies associated with processes with photons play an essential role at the Large Hadron Collider (LHC). With their clean signal, they contributed significantly to the discovery of the Higgs particle. In this thesis, we provide important building blocks for precision studies of processes including photons. It is expected that in the decade to come, with the runs of the High Luminosity LHC, experimentalists will measure some of these processes at percent level precision. Comparisons to the experiment will require that theoretical predictions include at least next-to-next-to-leading-order (NNLO) QCD and next-to-leading-order (NLO) electroweak corrections. Some of the critical bottlenecks of NNLO calculations are double-virtual contributions, which require two-loop scattering amplitudes. Obtaining those amplitudes can be challenging given the complex Feynman integrals that they include and the large intermediate expressions encountered when computing the coefficients of the integrals. We present the computation of double-virtual corrections to two-photon and three-photon production at the LHC. We obtained the needed multi-loop amplitudes using the numerical unitarity method to determine the master integral coefficients and employing functional reconstruction algorithms. We achieved our result using Caravel, a numerical framework for the calculation of one- and two-loop helicity amplitudes, which we extended to include photons and the capability to operate in the physical region of phase space. Our three-photon production amplitudes are the first publicly available expressions of any two-loop five-particle amplitude ready for phenomenological studies. Our results have been used in a phenomenological calculation that has been compared with measurements by the ATLAS collaboration.

List of Publications

The thesis is based on the following articles:

1. S. Abreu, J. Dormans, F. Febres Cordero, H. Ita, M. Kraus, B. Page, E. Pascual, M. S. Ruf, and V. Sotnikov. *"Caravel: A C++ Framework for the Computation of Multi-Loop Amplitudes with Numerical Unitarity."* Computer Physics Communications, 108069 (2021). `arXiv:2009.11957 [hep-ph]`

2. S. Abreu, B. Page, E. Pascual, V. Sotnikov. *"Leading-Color Two-Loop QCD Corrections for Three-Photon Production at Hadron Colliders"* JHEP, 01:078, 2021. `arXiv:2010.15834 [hep-ph]`

Contents

1 Introduction **1**

2 Theory **5**
2.1 Cross Sections . . . 5
2.2 Gauge Theories . . . 8
2.3 Quantization . . . 19
2.4 Renormalization . . . 30
2.5 Cross Sections Revised . . . 37

3 Methodology **41**
3.1 Helicity Amplitudes . . . 41
3.2 Color Ordering . . . 44
3.3 Recursion Relations . . . 46
3.4 Integration . . . 48
3.5 Integral Functions . . . 51
3.6 Unitarity . . . 53
3.7 Momentum Twistors . . . 56
3.8 Pole Structure . . . 57
3.9 Functional Reconstruction . . . 58

4 Caravel Framework **61**
4.1 Modules . . . 61
4.2 Installation and Configuration . . . 65
4.3 Tests . . . 68
4.4 Dependencies . . . 72
4.5 Other External Tools . . . 77
4.6 Examples . . . 79
4.6.1 Numerical Amplitudes . . . 81
4.6.2 Coefficient Reconstruction . . . 85
4.6.3 Photon Amplitudes . . . 89

5 Two-Loop Two-Parton Three-Photon Amplitudes **91**
5.1 Helicity Amplitudes . . . 91

5.2 Finite-Remainder Definition 95
5.3 Pentagon-Function Decomposition 97
5.4 Amplitude Evaluation 98
5.5 Analytic Master-Integral Coefficients 100
5.6 Analytic Finite Remainders 102
5.7 Validation 103
5.8 Collinear Limits 104
5.9 Ancillary Files 108
5.10 Hard Functions 110

6 Conclusion **114**

Appendix **116**
A Mathematical Formulas 116
B On-Shell Momenta 121
C CaravelGraph Structure 123
D Master Integrals for Three-Photon Production 124

Bibliography **128**

1. Introduction

We can develop a physical understanding of nature by focusing on either more and more complex systems or on nature's fundamental building blocks. Following the latter approach, it is necessary to explore the elementary constituents of matter and study how they interact. For this purpose, physicists employed many tools. These reach from microscopes relying on visible light, to electron microscopy with smaller wavelengths and, in the end, high-energy particle accelerators.

The particle accelerator with the highest beam energy today is the Large Hadron Collider (LHC) at CERN. It is a circular collider with a circumference of 26.7 km [1], designed to achieve maximum center-of-mass energy of 14 TeV. Up to now, experiments at the LHC, most notably the general-purpose experiments ATLAS and CMS, have accumulated an integrated luminosity of about 160 fb^{-1} [2]. Before the LHC, the most powerful accelerator was the Tevatron with a center-of-mass energy of 1.96 TeV [3]. That means that with the invention of the LHC, experimentalists could study a new energy regime. Today and after two runs, the LHC could already advance our understanding of particle physics in multiple ways. The Tevatron had a high enough center-of-mass energy to allow the top quark to be observed [4, 5] and get a reasonable estimate for its mass. However, the production rate was not high enough to study other parameters like its decay width [6]. At the LHC, the top pair production rate is orders of magnitude higher so that the ATLAS Collaboration could determine its decay width with a precision that made a quantitative comparison to theory possible [7]. The most prominent achievement of the LHC was the discovery of the Higgs particle [8, 9] and the measurement of its production and decay mechanisms. With Run 2 of the LHC, the experimental uncertainty for many of the measured cross sections has been reduced significantly [10]. With Run 3 of the LHC coming next year, the collision center-of-mass energy will be increased once again [11]. ATLAS and CMS upgraded the readout of their inner layers during the current shutdown to support higher rates. The upgrade will make it possible to keep the peak luminosity at Run 3 up for a longer time [12], leading to a doubling of the integrated luminosity during Run 3 compared to Run 2. Thus, statistical uncertainties will become significantly smaller, leaving us to expect even more precise results from Run 3.

The Run 3 forecast means that soon experimentalists can test the Standard Model of particle physics with more precise measurements and that phenomenologists will be able to spot tiny deviations. For this to happen, theorists have

to produce precise predictions. To accomplish this task, they expand the cross sections of the studied events perturbatively in the coupling constant of the underlying theory. Currently, next-to-leading-order (NLO) electroweak corrections and next-to-next-to-leading-order (NNLO) QCD corrections are phenomenologically relevant. However, phenomenologists often can use theoretical predictions which are still at NLO in the strong coupling constant only; see for example references [13, 14]. Thus, currently, the challenge for the theorists is to compute NNLO QCD corrections for processes with two or three particles in the final state.

The main building blocks needed to compute cross sections are scattering amplitudes. Higher orders in the coupling constant can come through additional real particles in the final state or additional virtual intermediate particles. The presence of these particles increases the number of interactions, therefore raising the power of coupling constants.

Corrections stemming from real particles in the final state lead to additional scales, so to more phase space integration variables. Phenomenologists perform these high-dimensional integrations numerically using Monte Carlo techniques. They take special precautions when particles become unresolved, either by being soft or collinear to other particles, which leads to singular integrals. They use different methods to deal with this type of problem, like subtraction, like references [15–18], and slicing, e.g., reference [19].

If the corrections come from virtual particles, which is our focus of study here, there are different obstacles to overcome. To achieve sufficiently precise results, we must efficiently integrate over the internal momenta. To this end, we reduce the involved integrals to a suitable basis. This can be done using Passarino–Veltman reduction [20] for NLO calculations or integration-by-parts techniques [21] for NLO or NNLO calculations. After that, we must evaluate the basis integrals, which are called master integrals. In simple cases, this can be done using Feynman parameters or Mellin–Barnes techniques [22], whereas for higher-order corrections with multiple scales the method of differential equations [23] often proves to be more successful. See also reference [24] for a review.

Using integration-by-parts techniques to find a set of master integrals and the method of differential equations to compute these made it possible to find master integrals suitable for all two-to-three processes with massless particles [25]. To further simplify the evaluation, the team of reference [25] presented the master integrals in terms of a smaller set of transcendental functions, which are called pentagon functions.

Once master integrals for the virtual corrections to the process are known, the coefficients of these integrals still have to be determined to arrive at the final result. We could achieve this using a Feynman diagram approach, considering all diagrams contributing to the correction and matching the integral coefficients.

For higher-order QCD calculations, new methods have been developed in the last years, particularly numerical unitarity [26–29]. It made it possible to de-

termine the analytic form of the planar two-loop five-parton amplitudes [30]. Numerical unitarity allows the calculation of master integral coefficients from on-shell gauge-invariant tree-level diagrams, thus reducing computational effort compared to pure Feynman diagrammatic methods.

This technique, applied to NNLO QCD, became publicly available in the Caravel framework [31]. Initially, the team of reference [31] set up Caravel to handle color-ordered diagrams in QCD. Because of the achieved results, this approach looks promising and is currently being applied to various processes. One of the key technical results of this work was to extend the Caravel library to include photons as external states involving upgrades to handle their unordered diagrams effectively.

Processes involving photons are interesting in multiple ways. Photons have a relatively clean signature in particle detectors. Moreover, a key decay of the currently actively studied Higgs particle is into two photons. The calculation of NNLO QCD corrections to three-photon production at the LHC [32] is of particular interest. It plays an important role in QCD phenomenology yielding the first precision phenomenology study for a three-particle final state process. Three-photon production is further relevant for determining anomalous gauge couplings as present in theories beyond the Standard Model [33, 34]. The ATLAS collaboration analyzed the process with Run 1 data [35]. Already here, the theoretical prediction superseded the experimental uncertainty. The effect will be even more significant for a future analysis using the Run 2 data.

In this work, we present the two-loop scattering amplitudes for the process $q\bar{q} \to \gamma\gamma\gamma$ and assemble the so-called hard functions required for NNLO QCD phenomenology. Our results can directly be evaluated using a new pentagon functions library [36], which is significantly faster than the previous version [25]. Using Caravel, we obtained numerical finite field values for the coefficients of the master integrals and the coefficients of the pentagon functions. From these, we got analytic expressions for the coefficients with functional reconstruction methods. Because there are no colored particles in the final state, the amplitudes can be used in existing subtraction algorithms. This has been done in the phenomenological study of reference [37]. The latter results were compared to the calculation of reference [32], finding agreement in differential and total cross sections at the 1 percent level. Furthermore, recently our scattering amplitudes have been reproduced by the independent study of reference [38].

This thesis is organized as follows. In chapter 2, we give a summary of standard procedures for computing physical observables in the context of QFT. In particular, we explain scattering cross sections from an experimental and theoretical point of view. Following this, we state how to quantize expressions in the context of particle physics. In the end, we discuss how to renormalize a theory to deal with occurring singularities. In chapter 3, we define the techniques we use for the calculation of QCD helicity amplitudes. That is, we explain how to treat

regularized amplitudes with quarks and introduce their color decomposition. We give an overview of how to determine master integrals using integration-by-parts techniques and the method of differential equations. Furthermore, we explain the numerical unitarity method used to determine numerical coefficients of the master integrals and functional reconstruction techniques to get their analytic form. In chapter 4, we describe the main building blocks of the Caravel framework. We outline the structure and interplay between its modules, delineate consistency tests and, provide examples. We furthermore depict the external input Caravel relies on to operate. In chapter 5, we present the main result of this work. We compute the analytic form of the two-loop helicity amplitudes for three-photon production at hadron colliders. We explain the structure of the results, how we validated them and how to use them for phenomenological studies. In chapter 6, we present our conclusions and give technical material in a series of appendices.

2. Theory

During a particle collider experiment, various scattering processes take place. The quantities, which the experiment can measure to probe particle physics, are scattering cross sections. Quantum field theories (QFTs) have been proven successful in describing scattering processes and their cross sections. QFTs are developed by postulating a classical Lagrangian density and quantizing it. To arrive at phenomenological predictions, one usually considers perturbation theory in a coupling constant. Within perturbative QFT, ultraviolet infinities occur. These infinities must be regularized and then renormalized to yield results comparable to experiments. Infrared singularities cancel in correctly set up physical observables such as cross sections.

2.1. Cross Sections

The properties of elementary particles can be determined using particle accelerators, which bring particles to collision within particle detector experiments. If the center-of-mass energy of the collision is higher than the mass threshold to produce a certain particle, this particle can be studied using a detector. The accelerator with currently the highest collision energy is the LHC at CERN, which is designed to have a maximum of $\sqrt{s} = 14$ TeV [39]. It also has one of the highest luminosities L [40] which measures the number of collisions that can occur at a given time. In its detectors, most notably the multi-purpose experiments ATLAS and CMS, the proton beams are crossed to produce collisions. If the two beams consist of Gaussian bunches with root mean squares σ_1 and σ_2 and N_1 and N_2 particles each which collide with frequency ν, the luminosity is given by[1]

$$L = \frac{\nu N_1 N_2}{4\pi\sigma_1\sigma_2}\,. \tag{2.1}$$

During a collision of two hadrons, these scatter, and new particles are produced, which can be identified by the detectors. Performing the experiment one notices that the number of these events N of specific particles being produced per time scales with the luminosity

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \sigma_{\mathrm{h}} L\,. \tag{2.2}$$

[1] For simplicity we assume a single bunch. The number of bunches appears as a multiplicative factor N_b, such that $L = \nu N_1 N_2 N_b / 4\pi\sigma_1\sigma_2$.

The proportionality factor σ_{h}, can be written as

$$\sigma_{\mathrm{h}} = \frac{\dot{N}}{L} = \frac{\dot{P}}{\Phi}\,, \tag{2.3}$$

where P is the probability of the process event taking place and Φ is the relative particle flux. σ_h is called the hadronic cross section of the studied process and can be computed in field theory.

The cross section gives us an insight into the probability that an event happens. A related value, which considers the directions and energies of the produced particles is the differential cross section. The differential cross section is a differential $\mathrm{d}\sigma_{\mathrm{h}}$ such that

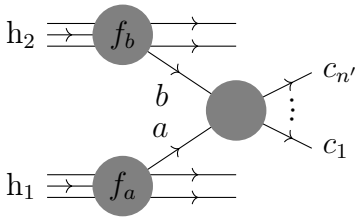
$$\sigma_{\mathrm{h}} = \int \mathrm{d}\sigma_{\mathrm{h}}\,, \tag{2.4}$$

which can be given with respect to various parameters a detector can measure, for example the solid angle Ω

$$\sigma_{\mathrm{h}} = \int \frac{\mathrm{d}\sigma_{\mathrm{h}}}{\mathrm{d}\Omega}\mathrm{d}\Omega\,. \tag{2.5}$$

A central role in the computation of the hadronic cross section is played by the parton model [41]. According to the parton model, the collided protons have an inner structure. The proton's momentum is distributed among the elementary particles inside it, which are the partons. The parton i then has the probability $f(x_i)$ to have the fraction x_i of the total momentum of the proton. f is called a parton distribution function and relies on experimental input to be determined.

During the scattering of two hadrons h_1 and h_2, two partons a and b, one of each hadron, are thought to interact with each other, as shown in the following picture.



During this scattering, the particles $c_1, \ldots, c_{n'}$ are created. With $\mathrm{d}\sigma$ being the differential cross section of the process $ab \to c_1, \ldots, c_{n'}$, the differential cross section on hadron level is given by

$$\mathrm{d}\sigma_{\mathrm{h}} = \int \mathrm{d}x_1 \mathrm{d}x_2 \sum\nolimits_{a,b} f_a(x_1) f_b(x_2) \mathrm{d}\sigma\,. \tag{2.6}$$

Here, f_a and f_b are the parton distribution functions for the partons a and b.

Particles $c_1, \ldots, c_{n'}$ can be identified by analyzing the detector signals. Similarly, experimentalists can determine many of their properties. Within the inner layer of a detector, a magnetic field bends the paths of charged particles, which allows fixing upon their charges and momenta. Calorimeters measure the energies of the particles. There are also specific devices, which help to distinguish between different particles with the same charge. An electron, for example, would lay a track inside the inner layer of a detector and be stopped in the electromagnetic calorimeter producing characteristic bremsstrahlung. Because of an effect called confinement, partons produced during collision wouldn't fly through the detector on their own. They would instead produce a parton shower and eventually hadronize. A hadron would pass the electromagnetic calorimeter. It would end up in a hadronic shower inside the hadronic calorimeter.

The partons' total momentum in the forward direction is only described statistically by the parton distribution functions. Because of that, only the part of the momenta transversal to the beam axis can be constrained by momentum conservation. There is a useful parametrization of the momenta, which considers this. If a beam axis points into z direction, we set

$$p^\mu = (m_T \cosh\eta, p_T \sin\phi, p_T \cos\phi, m_T \sinh\eta)\,, \tag{2.7}$$

with the transverse momentum p_T, the transverse mass m_T and the pseudorapidity η given by

$$p_T = \sqrt{p_x^2 + p_y^2}\,, \quad m_T = \sqrt{p_T^2 + m^2}\,, \quad \text{and} \quad \eta = \frac{1}{2}\log\left(\frac{|\mathbf{p}| + p_z}{|\mathbf{p}| - p_z}\right). \tag{2.8}$$

p_T, m_T, ϕ and differences of η defined in such a way are invariant under boosts in beam direction. The experiments measure the angle ϕ within the plane transverse to the beam, the angle θ to the beam axis and the transverse energy $E_T = E\sin\theta$, which are directly related to the former quantities.

If a parton is created, its energy does not correspond to a single track because of the showers and hadronization occurring. Instead, it is assigned the transverse energy inside a region in the $\phi\eta$-plane measured by

$$R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}\,. \tag{2.9}$$

A precise definition requires an infrared-safe algorithm, the so-called jet algorithm. The individual tracks inside the detector, which belong to a jet, need to be combined, usually using a clustering algorithm such as the anti-k_t algorithm [42]. Similarly, prompt photons have to be distinguished from those produced during a shower, which can be done using Frixione isolation [43].

After performing an analysis of the experiment's results, the differential cross

sections to these parameters, that is, for example, $\frac{\mathrm{d}\sigma}{\mathrm{d}p_T}$, or the difference in angle of two produced particles $\frac{\mathrm{d}\sigma}{\mathrm{d}\Delta\phi}$, can be obtained. To compare to or test the existing particle physics theories, one thus has to determine these cross sections, which is typically done in the framework of perturbative QFT.

2.2. Gauge Theories

Gauge theories describe the dynamics of fields, which are smooth functions on the Minkowski space. As in classical mechanics, with gauge theories, a physical system can be described by its action functional. Some formulas for that are summarized in the appendix A. The most straightforward field theory is a scalar field theory, where the fields are real- or complex-valued. We will use scalar field theories to introduce field dynamics. The process we are interested in is described by QCD, which is a non-abelian gauge theory. To define QCD, we need a description of gauge fields, as well as matter fields. These fields have values in different representations of the Lorentz algebra, which we will describe. We then introduce QCD fields, as well as extensions relevant to our process of interest.

While the properties of the QCD gauge fields differ fundamentally, the description of free QCD gauge fields is similar to a theory whose implications are easily observed, which is electrodynamics. As an introduction to free gauge fields, we describe electrodynamics without matter fields in the following.

Electrodynamics We know from observation that moving charges create an electric field $\mathbf{E}$ and a magnetic field $\mathbf{B}$, for which Maxwell's equations hold. We want to write them in a Lorentz covariant way, which will allow us to find a Lorentz invariant Lagrangian for them. Maxwell's equations are given by

$$\epsilon_{ijk}\partial_j E_k + \partial_0 B_i = 0\,, \tag{2.10}$$
$$\partial_i E_i = \rho\,, \tag{2.11}$$
$$\epsilon_{ijk}\partial_j B_k - \partial_0 E_i = J_i\,, \tag{2.12}$$
$$\partial_i B_i = 0\,, \tag{2.13}$$

where ρ is the charge density of the source and $\mathbf{J}$ its current density.

We now will see that ρ and $\mathbf{J}$ together transform contravariantly under Lorentz transformations, which will help us to understand why the gauge field we are going to introduce transforms in the same way. Because the total charge is conserved, ρ and $\mathbf{J}$ obey a continuity equation

$$\partial_0\rho + \partial_i J_i = \partial_\mu J^\mu = 0\,, \tag{2.14}$$

where we define the four-current $J = (\rho, \mathbf{J})$. We want to examine, how J^μ and the derivative transforms under Lorentz transformations. In the following we write $x^{\mu'} = x'^\mu$. The coordinates x transform under $\Lambda \in \mathrm{O}(1,3)$ according to

$$x^{\mu'} = \Lambda^{\mu'}{}_{\nu} x^\nu \,. \tag{2.15}$$

That is, using the chain rule, derivatives transform according to

$$\partial_{\mu'} = \frac{\partial x^\nu}{\partial x^{\mu'}} \partial_\nu = (\Lambda^{-1})^\nu{}_{\mu'} \partial_\nu \,. \tag{2.16}$$

Applying the continuity equation (2.14) to J and J' yields

$$\partial_\mu J^\mu = 0 = \partial_{\mu'} J^{\mu'} = (\Lambda^{-1})^\nu{}_{\mu'} \partial_\nu J^{\mu'} \,, \tag{2.17}$$

so that it must hold

$$J^{\mu'} = \Lambda^{\mu'}{}_{\nu} J^\nu \,. \tag{2.18}$$

That is, as the coordinates x, J transforms contravariantly under Lorentz transformations.

We next define the gauge field of electrodynamics, which will help us to understand the gauge field of QCD. Because of the fourth of Maxwell's equations (2.13), we can write $\mathbf{B}$ using a vector potential $\mathbf{A}$, as

$$B_i = \epsilon_{ijk} \partial_j A_k \,. \tag{2.19}$$

Then, because of the first of Maxwell's equations (2.10), we have

$$\epsilon_{ijk} \partial_j (E_k + \partial_0 A_k) = 0 \,, \tag{2.20}$$

so that we can define a scalar potential ϕ, satisfying

$$-\partial_i \phi = E_i + \partial_0 A_i \,. \tag{2.21}$$

The potentials ϕ and $\mathbf{A}$ determine the fields $\mathbf{E}$ and $\mathbf{B}$. However, since they define $\mathbf{E}$ and $\mathbf{B}$ through their derivatives, there is an ambiguity of choice of ϕ and $\mathbf{A}$ for fixed $\mathbf{E}$ and $\mathbf{B}$. This degree of freedom is known as gauge symmetry, and the fields ϕ and $\mathbf{A}$ will form the gauge field of electrodynamics. We will discuss gauge fields in more detail later in this section. Because of equations (2.10, 2.13, 2.21), given the potentials ϕ and $\mathbf{A}$, we can choose an arbitrary scalar field α so that

$$A'_i = A_i + \partial_i \alpha \,, \quad \text{and} \quad \phi' = \phi - \partial_0 \alpha \tag{2.22}$$

will define the same fields $\mathbf{E}$ and $\mathbf{B}$. Since the choice of the gauge is unphysical, we can choose one, which best suits our needs. A convenient choice is the Lorenz

gauge, where α is the solution of

$$0 = \partial_i^2 \alpha - \partial_0^2 \alpha \,. \tag{2.23}$$

If we now define the four-potential $A = (\phi, \mathbf{A})$, combining equations (2.12,2.19,2.21) we arrive at

$$\partial_\nu \partial^\nu A^\mu = J^\mu \,. \tag{2.24}$$

We now can apply the known transformations from equations (2.16, 2.18) to see that

$$A^{\mu'} = \Lambda^{\mu'}{}_{\mu} A^\mu \,. \tag{2.25}$$

That means the four-potential also transforms contravariantly under Lorentz transformations. Accordingly, A_μ transforms covariantly under Lorentz transformations.

We can abbreviate the notation if we notice that $\mathbf{E}$ and $\mathbf{B}$ are part of the Field-strength tensor

$$F^{\mu\nu} = \partial^\mu A^\nu - \partial^\nu A^\mu \,. \tag{2.26}$$

Using equations (2.16, 2.25), we see that F also transforms contravariantly under Lorentz transformations. Using F, Maxwell's equations can be written as

$$\begin{aligned} \partial_\mu F^{\mu\nu} &= J^\nu \,, \\ \epsilon^{\mu\nu\rho\sigma} \partial_\nu F_{\rho\sigma} &= 0 \,, \end{aligned} \tag{2.27}$$

where the first is the equation of motion for A. The Lagrangian, which gives these Euler-Lagrange equations should give the same results in every inertial frame, that is it should be Lorentz invariant. It also should not be dependent on an unphysical gauge choice, that is it should be gauge invariant. This means in particular,

$$\mathcal{L}(A^{\mu'}, \partial_{\nu'} A^{\mu'}) = \mathcal{L}(A^\mu, \partial_\nu A^\mu) \,, \tag{2.28}$$

for the transformations (2.25, 2.22). If we consider first the vacuum case $\mathcal{L}_0$, for which $J = 0$, then it can be given by

$$\mathcal{L}_0 = c F_{\mu\nu} F^{\mu\nu} \,, \tag{2.29}$$

Where we now need to determine the coefficient c. As in classical mechanics, the normalization can be constructed by looking at the kinetic energy. It holds that

$$\int \mathrm{d}^3 x \mathcal{L}_0 = T - V \,, \tag{2.30}$$

where the kinetic energy is quadratic in the time derivatives of the field A. Con-

sider for a moment a field with $\partial_i A_0 = 0$, then

$$T = \int \mathrm{d}^3x \frac{\dot{A}_i \dot{A}^i}{2} = -\frac{1}{4}\int \mathrm{d}^3x (F_{0i}F^{0i} + F_{i0}F^{i0}) \,, \tag{2.31}$$

if we write it with the part of the sum $F_{\mu\nu}F^{\mu\nu}$ which contributes here, so it must be

$$\mathcal{L}_0 = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} \,. \tag{2.32}$$

Now we can include back the source to find

$$\mathcal{L} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + A_\mu J^\mu \,. \tag{2.33}$$

Thus, we have constructed a Lagrangian for electromagnetism. To better understand field theory and how it emerges from mechanics we now have a look at scalar field theory.

Scalar Field Theory We now want to find the Lagrangian of a scalar field invariant under Lorentz transformations [44]. A scalar field is a function on the Minkowski space, which transforms as

$$\varphi'(x^{\mu'}) = \varphi'(\Lambda^{\mu'}{}_\nu x^\nu) = \varphi(x^\mu) \,, \tag{2.34}$$

so that one always gets the same result for the field's value, irrespective of the Minkowski space coordinates. That means it transforms under the trivial representation of the Lorentz group. The Higgs field, for example, is scalar. To find the corresponding Lagrangian, we look at the field's values on every spatial point as a generalized coordinate, just as in classical mechanics. For that we discretize the Minkowski space and label every point by an index a. Then we define

$$q_a(t) \equiv \varphi(\vec{x}_a, t) \quad \text{and} \quad \dot{q}_a(t) \equiv \frac{\partial\varphi}{\partial t}(\vec{x}_a, t) \,, \tag{2.35}$$

so that

$$L(t) = \sum\nolimits_a \mathcal{L}_a(q_a(t), \dot{q}_a(t)) = \sum\nolimits_a \frac{\dot{q}_a^2}{2} - V(q_a) \,. \tag{2.36}$$

Now in order to be invariant under Lorentz transformations, a time derivative cannot appear alone in the Lagrangian but rather a four-gradient. The discretized version of the spacial gradient in direction of $\vec{x}_{a+1}$ can be calculated by

$$\partial_{\vec{x}_{a+1}}\varphi(\vec{x}_a, t) = \frac{q_{a+1} - q_a}{\|\vec{x}_{a+1} - \vec{x}_a\|} \,. \tag{2.37}$$

Now if λ is the cell width of our discrete space, we can define k and K so that

$$K(\lambda) = \frac{k(\lambda)}{\|\vec{x}_{a+1} - \vec{x}_a\|} \quad \text{with} \quad \lim_{\lambda \to 0} K(\lambda) = \text{const}\,, \tag{2.38}$$

and set

$$V(q_a) = \frac{K}{2}(q_{a+1} - q_a)^2\,. \tag{2.39}$$

The system then becomes a system of coupled harmonic oscillators

$$L(t) = \sum\nolimits_a \Big(\frac{\dot{q}_a^2}{2} - \frac{K}{2}(q_{a+1} - q_a)^2 \Big)\,. \tag{2.40}$$

Taking the limit $\lambda \to 0$, the sum over the oscillators becomes an integral over space and with equations (2.35) and (2.37) we get

$$L(t) = \int \mathrm{d}^3x \mathcal{L}_0(\varphi, \partial_\mu \varphi)\,, \tag{2.41}$$

with

$$\mathcal{L}_0(\varphi, \partial_\mu \varphi) = -\frac{1}{2}\partial_\mu \varphi \partial^\mu \varphi\,. \tag{2.42}$$

In general, we can add terms with higher power of φ to the potential, where in the minimum of the potential there are no linear terms.

$$\mathcal{L}(\varphi, \partial_\mu \varphi) = -\frac{1}{2}\partial_\mu \varphi \partial^\mu \varphi - \frac{1}{2}m^2\varphi^2 - \frac{g_3}{3!}\varphi^3 - \frac{g_4}{4!}\varphi^4 - \ldots \tag{2.43}$$

The quadratic term is by dimensional analysis related to the mass term and higher terms are interaction terms. Consider a Lagrangian with only the first two terms. Using integration-by-parts we can bring it in the form

$$\mathcal{L}(\varphi, \partial_\mu \varphi) = \varphi(\Box - m^2)\varphi\,, \tag{2.44}$$

with the equation of motion

$$(\Box - m^2)\varphi = 0 \tag{2.45}$$

called the Klein–Gordon equation.

Accordingly, we can consider complex scalar fields ϕ, parametrized additionally by their complex conjugate $\bar{\phi}$. If both, ϕ and $\bar{\phi}$ are to solve the Klein–Gordon equation (2.45), the Lagrangian has to have the form

$$\mathcal{L}_0(\phi, \partial_\mu \phi) = \bar{\phi}(\Box - m^2)\phi\,. \tag{2.46}$$

We notice, that the complex scalar Lagrangian has a U(1) symmetry

$$\phi' = e^{i\alpha}\phi\,. \tag{2.47}$$

Finally, as in the Lagrangian for electromagnetism (2.33), we can introduce a source term to the Lagrangian. In order not to break the U(1) symmetry, we define a complex source χ and extend the Lagrangian to be

$$\mathcal{L} = \bar{\phi}(\Box - m^2)\phi + \bar{\chi}\phi + \bar{\phi}\chi\,. \tag{2.48}$$

Indeed, calculating the Euler–Lagrange equations now yields

$$(\Box - m^2)\phi = -\chi\,, \tag{2.49}$$

and accordingly for the complex conjugate fields, so that χ is indeed a source term.

Bosons and Fermions When a scalar field transforms under the Lorentz group like in equation (2.34), a generic field transforms according to

$$\Phi'(x'^{\mu}) = \rho(\Lambda)\Phi(x^{\mu})\,, \tag{2.50}$$

with ρ some representation of the Lorentz group. For these fields, Lagrangians and thus equations of motion can also be obtained. Inside the Lorentz algebra $\mathfrak{so}(3,1)$, for the generators of rotations J_i and boosts K_i defining the combinations

$$M_i = \frac{J_i + \mathrm{i}K_i}{2} \quad \text{and} \quad N_i = \frac{J_i - \mathrm{i}K_i}{2} \tag{2.51}$$

gives two independent $\mathfrak{su}(2)$ algebras, that is

$$[M_i, M_j] = \mathrm{i}\epsilon_{ijk}M_k\,, \quad [N_i, N_j] = \mathrm{i}\epsilon_{ijk}N_k \quad \text{and} \quad [M_i, N_j] = 0\,. \tag{2.52}$$

That is, the algebra of the Lorentz group is isomorphic to $\mathfrak{su}(2) \oplus \mathfrak{su}(2)$, where every of the $\mathfrak{su}(2)$ can be in any of its representations. The trivial representation transforms scalar fields, so we next look at the fundamental representation. It has a basis given by the Pauli matrices

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\,, \quad \sigma_2 = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}\,, \quad \text{and} \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\,. \tag{2.53}$$

Conventionally we normalize the basis by $\frac{1}{2}$ and call the respective representations with one of the $\mathfrak{su}(2)$ in the fundamental and the other in the trivial representation, the $(\frac{1}{2}, 0)$ and the $(0, \frac{1}{2})$ representations. We get a representation

for the corresponding Lorentz groups by exponentiation, so for a generic Lorentz transformation

$$\Lambda^{\mu}{}_{\nu} = \left(\mathrm{e}^{\mathrm{i}\omega^i K_i}\mathrm{e}^{\mathrm{i}\theta^i J_i}\right)^{\mu}{}_{\nu} \tag{2.54}$$

in the $(\frac{1}{2},0)$ representation we get $J_i = \mathrm{i}K_i = \frac{\sigma_i}{2}$, which then acts on two-dimensional fields as

$$\psi'_a = \left(\mathrm{e}^{\frac{\omega^i\sigma_i}{2}}\mathrm{e}^{\mathrm{i}\frac{\theta^i\sigma_i}{2}}\right)_a{}^b\psi_b\,, \tag{2.55}$$

where the metric is given by

$$\varepsilon^{ab} = -\mathrm{i}\sigma_2\,. \tag{2.56}$$

so that

$$\psi\eta = -\eta\psi\,. \tag{2.57}$$

We call fields transforming under the $(\frac{1}{2},0)$ representation left-handed Weyl spinors. Similarly, fields transforming under the right-handed representation $(0,\frac{1}{2})$, where $M_i = 0$ are called right-handed Weyl spinors and satisfy

$$\eta^{\dot{a}\prime} = \left(\mathrm{e}^{-\frac{\omega^i\sigma_i}{2}}\mathrm{e}^{\mathrm{i}\frac{\theta^i\sigma_i}{2}}\right)^{\dot{a}}{}_{\dot{b}}\eta^{\dot{b}}\,. \tag{2.58}$$

Lagrangians invariant under $(\frac{1}{2},0)$ and $(0,\frac{1}{2})$ transformations with $\sigma^\mu = (1,\vec{\sigma})$ and $\bar{\sigma}^\mu = (1,-\vec{\sigma})$ can be given by

$$\begin{aligned}\mathcal{L}(\psi_a,\partial_\mu\psi_a) &= \mathrm{i}\psi^\dagger_{\dot{a}}\sigma^{\mu\dot{a}b}\partial_\mu\psi_b\,,\\ \mathcal{L}(\eta^{\dot{a}},\partial_\mu\eta^{\dot{a}}) &= \mathrm{i}\eta^{\dagger a}\bar{\sigma}^{\mu}_{a\dot{b}}\partial_\mu\eta^{\dot{b}}\,,\end{aligned} \tag{2.59}$$

which are the equivalent of equation (2.44) for spinors, however first degree on derivatives and without a mass term. The corresponding equations are the Weyl equations

$$\begin{aligned}\mathrm{i}\sigma^{\mu\dot{a}b}\partial_\mu\psi_b &= 0\,,\\ \mathrm{i}\bar{\sigma}^{\mu}_{a\dot{b}}\partial_\mu\eta^{\dot{b}} &= 0\,.\end{aligned} \tag{2.60}$$

Applying both operators to ψ_b or to $\eta^{\dot{b}}$ one after the other yields the Klein–Gordon equation (2.45) for massless spinors, which has plane wave solutions [45]

$$\begin{aligned}\psi_b(x) &= \int\frac{\mathrm{d}^4p}{(2\pi)^4}u_b(p)\mathrm{e}^{\mathrm{i}p_\mu x^\mu}\,,\\ \eta^{\dot{b}}(x) &= \int\frac{\mathrm{d}^4p}{(2\pi)^4}v^{\dot{b}}(p)\mathrm{e}^{\mathrm{i}p_\mu x^\mu}\,.\end{aligned} \tag{2.61}$$

Plugging these solutions into the Weyl equations (2.60) yields

$$\begin{aligned} \sigma^{\mu\dot{a}b} p_\mu u_b &= 0\,, \\ \bar{\sigma}^{\mu}_{a\dot{b}} p_\mu v^{\dot{b}} &= 0\,. \end{aligned} \tag{2.62}$$

Now we have

$$\sigma^{\mu\dot{a}b} p_\mu = \begin{pmatrix} p_0 - p_3 & -p^1 + \mathrm{i}p_2 \\ -p_1 - \mathrm{i}p_2 & p_0 + p_3 \end{pmatrix}\,, \tag{2.63}$$

that is, with $p_\mu p^\mu = 0$, solutions for the Weyl equations in momentum space (2.62) can be given by [31]

$$\begin{aligned} |p\rangle \equiv u_+(p) &= \frac{\sqrt{|p_+|}}{p_+} \begin{pmatrix} p_+ \\ p_+^\perp \end{pmatrix}\,, \\ |p] \equiv v_+(p) &= \frac{1}{\sqrt{|p_+|}} \begin{pmatrix} p_+ \\ p_-^\perp \end{pmatrix}\,, \end{aligned} \tag{2.64}$$

with

$$p_+ = p^0 + p^3\,, \quad p_+^\perp = p^1 + \mathrm{i}p^2\,, \quad \text{and} \quad p_-^\perp = p^1 - \mathrm{i}p^2\,, \tag{2.65}$$

where we normalize so that $\sigma^\mu p_\mu = |p\rangle[p|$.

A way to introduce a mass term is to take the direct sum of the $(\frac{1}{2}, 0)$ and $(0, \frac{1}{2})$ representations of the Lorentz algebra to get another representation. If we define

$$\gamma^\mu = \begin{pmatrix} 0 & \sigma^\mu \\ \bar{\sigma}^\mu & 0 \end{pmatrix}\,, \tag{2.66}$$

then $\Sigma^{\mu\nu} = \gamma^\mu\gamma^\nu - \gamma^\nu\gamma^\mu$ gives a basis for the Lorentz algebra. We then can write a first degree Lagrangian for the bispinors $\psi = (\psi_a, \eta^{\dot{a}})$, invariant under action of the $(\frac{1}{2}, 0) \oplus (0, \frac{1}{2})$ representation as

$$\mathcal{L}_0(\psi, \partial_\mu\psi) = \bar{\psi}(\mathrm{i}\gamma^\mu\partial_\mu - m)\psi\,, \tag{2.67}$$

where $\bar{\psi} = \psi^\dagger\gamma^0$, which is the Dirac Lagrangian. For our purposes it is enough to consider Weyl spinor fields. Left-handed Weyl spinor fields transforming under the $(\frac{1}{2}, 0)$ representation of the Loretnz algebra are called fermions, while right-handed Weyl spinor fields transforming under the $(0, \frac{1}{2})$ representation of the Lorentz algebra are called antifermions. Just as in equation (2.48) for the scalar field, we can introduce fermionic source terms ξ so that the Dirac Lagrangian becomes

$$\mathcal{L} = \bar{\psi}(\mathrm{i}\gamma^\mu\partial_\mu - m)\psi + \bar{\xi}\psi + \bar{\psi}\xi\,, \tag{2.68}$$

With the corresponding Euler–Lagrange equation

$$(\mathrm{i}\gamma^\mu\partial_\mu - m)\psi = -\xi\,, \tag{2.69}$$

and accordingly for the conjugated fields.

Finally, we can consider the $(\frac{1}{2}, \frac{1}{2})$ representation of the Lorentz algebra, that is, we deal with fields acted on by $\mathrm{SU}(2) \otimes \mathrm{SU}(2)$ and thus having two indices. A basis for these is given by σ_μ so we can write

$$\psi_{\alpha\dot{\alpha}} = A_\mu \sigma^\mu_{\alpha\dot{\alpha}} \,. \tag{2.70}$$

These are the covariant vector fields that transform according to

$$A_{\mu'}(x') = \frac{\partial x^\nu}{\partial x^{\mu'}} A_\nu(x) \,, \tag{2.71}$$

and whose Lagrangian is the Lagrangian of electromagnetism as given in equation (2.32). These fields are called vector bosons, like the four-potential in electrodynamics. Just as Weyl spinors, they satisfy a massless Klein–Gordon equation and have plane wave solutions

$$A_\mu = \int \frac{\mathrm{d}^4 p}{(2\pi)^4} \epsilon^\pm_\mu \mathrm{e}^{\mathrm{i} p_\mu x^\mu} \,, \tag{2.72}$$

where the polarization states can now be chosen to

$$\begin{aligned} \epsilon^\mu_+(p,n) &= \frac{\langle n|\bar{\sigma}^\mu|p]}{\sqrt{2}\langle n|p\rangle} \,, \\ \epsilon^\mu_-(p,n) &= -\frac{[n|\sigma^\mu|p\rangle}{\sqrt{2}[n|p]} \,, \end{aligned} \tag{2.73}$$

where n^μ is an auxiliary massless reference vector.

Since the Lagrangian only depends on the derivatives of A_μ, and because of the definition of the field strength tensor (2.26), the Lagrangian is invariant under a symmetry transformation

$$A'_\mu(x) = A_\mu(x) + \partial_\mu \alpha(x) \,, \tag{2.74}$$

for some scalar field $\alpha(x)$, which is called gauge invariance. That is, to describe the physical problem, we have the freedom of fixing the gauge by choosing $\alpha(x)$ appropriate to our needs. Similarly, we see that the fermion Lagrangians (2.59) and (2.67) are invariant under a global U(1) symmetry transformation

$$\psi' = \mathrm{e}^{\mathrm{i}e\alpha} \psi \,. \tag{2.75}$$

To describe the combined system including vector bosons and fermions, we can promote this symmetry to the same local symmetry of the bosonic Lagrangian (2.32) by viewing A_μ as a connection on the principal U(1)-bundle over the

Minkowski space and ψ as a section of the associated spinor bundle. A_μ then induces a connection on the spinor bundle, with the covariant derivative

$$D_\mu = \partial_\mu + \mathrm{i}eA_\mu \,. \tag{2.76}$$

In that picture, the tensor $F_{\mu\nu}$ can be seen as the curvature of the connection given by

$$F_{\mu\nu} = \frac{\mathrm{i}}{e}[D_\mu, D_\nu] \,, \tag{2.77}$$

which in the U(1) case coincides with the previous definition in equation (2.26). For the definition of the connection with the corresponding Dirac operator see appendix A. With the corresponding Dirac operator $\not{D} = \gamma^\mu D_\mu$, the combined Lagrangian

$$\mathcal{L}(\psi, \partial_\mu\psi) = \bar{\psi}(\mathrm{i}\not{D} - m)\psi - \frac{1}{4}F_{\mu\nu}F^{\mu\nu} \,, \tag{2.78}$$

is invariant under a local U(1) symmetry, transforming

$$\psi'(x) = U(x)\psi(x) \tag{2.79}$$

and

$$A'_\mu(x) = U(x)A_\mu U^\dagger(x) + \frac{\mathrm{i}}{e}U(x)\partial_\mu U^\dagger(x) \,, \tag{2.80}$$

which in case of $U(x) \in \mathrm{U}(1)$ amounts to equation (2.74) .

The term $e\bar{\psi}\gamma^\mu A_\mu\psi$ in equation (2.78) describes the interactions between vector bosons and fermions. There are further transformations leaving the Lagrangian (2.78) invariant. The simplest one is the discrete transformation

$$A'_\mu = -A_\mu \,, \tag{2.81}$$

called charge conjugation. Since the bosonic kinetic part of the Lagrangian $-\frac{1}{4}F_{\mu\nu}F^{\mu\nu}$ is quadratic in A_μ, it is invariant under charge conjugation. In order for the rest of the Lagrangian to be invariant under charge conjugation, the fermionic fields ψ have to transform under these so that

$$\bar{\psi}\big(\mathrm{i}\not{\partial} - e\not{A} - m\big)\psi = \bar{\psi}'\big(\mathrm{i}\not{\partial} + e\not{A} - m\big)\psi' \,. \tag{2.82}$$

This can be achieved by setting

$$\psi' = -\mathrm{i}\gamma_2\psi^\star \,, \tag{2.83}$$

with $\psi \mapsto \psi'$ called the charge conjugation operator. Indeed, employing the

Clifford algebra rules, as given in appendix A, and using the basis

$$\gamma'_\mu = \gamma_2 \gamma^\star_\mu \gamma_2 \tag{2.84}$$

for the gamma matrices on the right-hand side yields equation (2.82). Because of the anti-diagonal structure of gamma matrices as seen in equation (2.66) and since the charge conjugation operator acts like the metric ε^{ab} from equation (2.56) on spinors, it sends fermion states to corresponding antifermion states and vice versa. Quantization, as described in section 2.3, makes the eigenvalues of the charge conjugation operator multiplicative quantum numbers. Such an eigenvalue is called charge parity, and it is -1 for photons, as seen from equation (2.81). Thus, charge conjugation provides an essential limitation on allowed field interactions.

Quantum Chromodynamics We now can think of gauge groups other than U(1), for example, SU(2), which occur in the theory of weak interactions, or SU(3) for strong interactions. Then, as in the U(1) case, to make the Lagrangian invariant under a local transformation, say $U(x) \in \mathrm{SU}(3)$, we introduce a connection G^a_μ on the principal SU(3)-bundle, which transforms under the adjoint representation of SU(3). ψ_i is then a section of the associated spinor bundle, transforming under the fundamental representation. The covariant derivative is given by

$$D_{\mu ij} = \partial_\mu \delta_{ij} + \mathrm{i} g_\mathrm{s} G^a_\mu T^a_{ij} \,. \tag{2.85}$$

For SU(3) the structure constants f^{abc} do not vanish, so that the curvature $\mathcal{G}^a_{\mu\nu}$ has the form

$$\mathcal{G}^a_{\mu\nu} = \partial_\mu G^a_\nu - \partial_\nu G^a_\mu + g_\mathrm{s} f^{abc} G^b_\mu G^c_\nu \,. \tag{2.86}$$

QCD is an SU(3) gauge theory making it a non-Abelian gauge theory. The fermions in the fundamental representation of SU(3) are called quarks and the bosons in the adjoint representation of SU(3) are called gluons. In QCD, there are six quark flavors with the same properties but distinct masses. The Lagrangian taking into account all flavors is given by

$$\mathcal{L} = \sum\nolimits_f \bar{\psi}^f_i (\mathrm{i} \not{D}_{ij} - m_f \delta_{ij}) \psi^f_j - \frac{1}{4} \mathcal{G}^a_{\mu\nu} \mathcal{G}^{\mu\nu}_a \,. \tag{2.87}$$

While the masses of the first five quarks are negligible with respect to the energy scale, we do not consider top quark effects, thus we deal with massless quarks in our calculations. In addition to quarks and gluons we want to study effects stemming from quark-photon couplings at NNLO in QCD. For that, we take the gauge group to be $\mathrm{SU}(3) \times \mathrm{U}(1)$ and define connections (G, A) on the principal $\mathrm{SU}(3) \times \mathrm{U}(1)$-bundle. The respective covariant derivative is then given by

$$D_{\mu ij} = \partial_\mu \delta_{ij} + \mathrm{i} g_\mathrm{s} G^a_\mu T^a_{ij} + \mathrm{i} e A_\mu \delta_{ij} \,, \tag{2.88}$$

and the Lagrangian by

$$\mathcal{L} = \mathrm{i} \sum\nolimits_f \bar{\psi}_i^f \not{D}_{ij} \psi_j^f - \frac{1}{4} \mathcal{G}_{\mu\nu}^a \mathcal{G}_a^{\mu\nu} - \frac{1}{4} F_{\mu\nu} F^{\mu\nu} \,, \tag{2.89}$$

where as explained we neglected the quark masses. Now in phenomenological studies photons will also be produced from interactions with electrons and positrons. Because we are only interested in contributions at NNLO in α_s, these are still suppressed so that we can neglect them.

2.3. Quantization

The so far described classical fields cannot explain quantum effects such as discrete energy levels or particle numbers. We want to solve this, by quantizing the field theory. In quantum mechanics, if we have an initial state of an n-particle system, it evolves to a final state of n' particles using the scattering matrix, or S-matrix, which we will introduce. In order to be able to calculate it, we can use the LSZ-formula, which relates the S-matrix elements with vacuum expectation values of the involved field operators. We can calculate vacuum expectation values using the path integral formalism. In order to do that, we will define path integrals for point particles. We will then extend this idea to free scalar fields and after that to interacting scalar fields. In order to calculate vacuum expectation values for theories with interacting fields, we will introduce Feynman diagrams. From Feynman diagrams, we can assemble scattering amplitudes, which determine the S-matrix elements and are the main topic of this work. We finally extend this approach to vector and spinor fields and state the Feynman rules needed to determine QCD scattering amplitudes.

S-Matrix In a quantum mechanical description of particle scattering, an initial state of n particles evolves to a final state of n' particles using an evolution operator called the scattering matrix or S-matrix, which we want to define in the following. In the Schrödinger picture states $|\psi(t)\rangle$ are time dependent and evolve using the Hamiltonian [46]

$$\mathrm{i}\partial_t |\psi(t)\rangle = (\hat{H}_0 + \hat{H}_I)|\psi(t)\rangle \,. \tag{2.90}$$

Equation (2.90) is the Schrödinger equation. We can solve the free equation to

$$|\psi(t)\rangle = U_0(t)|\psi(t_0)\rangle \,, \tag{2.91}$$

where U_0 is the unitary operator

$$U_0(t, t_0) = \exp\big(-\mathrm{i}\hat{H}_0(t - t_0)\big) \tag{2.92}$$

in case of free fields. As for the interaction part, states evolve using the interaction Hamiltonian

$$\mathrm{i}|\psi(t)\rangle_I = \hat{H}_I|\psi(t)\rangle_I\,, \tag{2.93}$$

so that

$$|\psi(t)\rangle_I = U_I(t,t_0)|\psi(t_0)\rangle_I\,. \tag{2.94}$$

Because we cannot solve this equation directly, perturbation theory becomes important. We expand U_I in the coupling constant g, that is for example e for QED or g_s for QCD, where $\hat{H}_I$ is proportional to g.

$$U_I(t,t_0) = \sum_{k=0}^{\infty} g^k U_k(t,t_0)\,. \tag{2.95}$$

Comparing coefficients in g we get

$$\partial_t U_k(t,t_0) = -\mathrm{i}\hat{H}_I(t) g^{k-1} U_{k-1}(t,t_0)\,, \tag{2.96}$$

which can be iteratively solved to

$$U_k(t,t_0) = (-\mathrm{i})^k \int_{t_0}^{t} \mathrm{d}t_1 \cdots \int_{t_0}^{t_{k-1}} \mathrm{d}t_k \hat{H}_I(t_1)\cdots\hat{H}_I(t_k)\,. \tag{2.97}$$

We want to integrate every variable t_i from 0 to t, which we can achieve by using the time ordering operator

$$T[\hat{H}_I(t_{i_1})\cdots\hat{H}_I(t_{i_k})] := \hat{H}_I(t_1)\cdots\hat{H}_I(t_k)\,, \tag{2.98}$$

with $t_1 > \cdots > t_k$. In the integral

$$\begin{aligned}\int_{t_0}^{t} \mathrm{d}t_1 \cdots \int_{t_0}^{t} \mathrm{d}t_k T[\hat{H}_I(t_1)\cdots\hat{H}_I(t_k)] \\ = \sum_{\sigma\in S(k)} \int_{t_0}^{t} \mathrm{d}t_{\sigma(1)} \cdots \int_{t_0}^{t} \mathrm{d}t_{\sigma(k)} \hat{H}_I(t_{\sigma(1)})\cdots\hat{H}_I(t_{\sigma(k)})\end{aligned} \tag{2.99}$$

we can then rename the variables t_i so that

$$= k! \int_{t_0}^{t} \mathrm{d}t_1 \cdots \int_{t_0}^{t} \mathrm{d}t_k \hat{H}_I(t_1)\cdots\hat{H}_I(t_k)\,, \tag{2.100}$$

and finally

$$U_k(t,t_0) = \frac{(-\mathrm{i})^k}{k!} \int_{t_0}^{t} \mathrm{d}t_1 \cdots \int_{t_0}^{t} \mathrm{d}t_k T[\hat{H}_I(t_1)\cdots\hat{H}_I(t_k)]\,, \tag{2.101}$$

which can be employed in the equation (2.95) to get a solution for $|\psi(t)\rangle_I$ using (2.94). We now define the scattering matrix S as the evolution operator from $-\infty$ to ∞

$$\begin{aligned} S = U_I(-\infty, \infty) &= \sum_{k=0}^{\infty} \frac{(-\mathrm{i})^k}{k!} \int_{-\infty}^{\infty} \mathrm{d}t_1 \cdots \int_{-\infty}^{\infty} \mathrm{d}t_k T[\hat{H}_I(t_1) \cdots \hat{H}_I(t_k)] \\ &= \sum_{k=0}^{\infty} \frac{(-\mathrm{i})^k}{k!} \int_{-\infty}^{\infty} \mathrm{d}x_1 \cdots \int_{-\infty}^{\infty} \mathrm{d}x_k T[\hat{\mathcal{H}}_I(t_1) \cdots \hat{\mathcal{H}}_I(t_k)] \,. \end{aligned} \tag{2.102}$$

Note that S is also unitary. The transition amplitude of an initial state of n particles $|i\rangle$ at $t = -\infty$ to a final state of n' particles $|f\rangle$ at $t = \infty$ is then given by

$$\mathcal{T}_{f \leftarrow i} = \langle f|S|i\rangle \,, \tag{2.103}$$

where the right-hand side in turn can be calculated using the LSZ reduction formula, which for a scalar field looks like

$$\begin{aligned} \langle f|S|i\rangle = \mathrm{i}^n \int \mathrm{d}^4 x_1 \mathrm{e}^{\mathrm{i}p_1 x_1} (-\partial_{x_1}^2 + m^2) \cdots \mathrm{i}^{n'} \int \mathrm{d}^4 x_1' \mathrm{e}^{-\mathrm{i}p_1' x_1'} (-\partial_{x_1'}^2 + m^2) \cdots \\ \times \langle 0|T\varphi(x_1) \ldots \varphi(x_1') \ldots |0\rangle \,. \end{aligned} \tag{2.104}$$

Vacuum Expectation Values According to equation (2.104), in order to calculate S-matrix elements $\langle f|S|i\rangle$, we need to be able to determine vacuum expectation values of the form $\langle 0|T\varphi(x_1) \ldots \varphi(x_1') \ldots |0\rangle$. To calculate these vacuum expectation values, we utilize the path integral [47–49]. We will define it for point particles first and extend the definition to fields in the following. To define a path integral, consider a particle in the potential V with the Hamiltonian

$$\hat{H}(\hat{x}, \hat{p}) = \frac{\hat{p}^2}{2m} + V(\hat{x}) \,. \tag{2.105}$$

We want the probability of the particle to evolve from (x', t') to (x'', t'') . This we can get using the time evolution operator [50]. Discretizing the time we have

$$\langle x'', t''|x', t'\rangle = \int \prod\nolimits_i \mathrm{d}x_i \langle x''|\mathrm{e}^{-\mathrm{i}H\delta t}|x_N\rangle \cdots \langle x_1|\mathrm{e}^{-\mathrm{i}H\delta t}|x'\rangle \,, \tag{2.106}$$

where for small δt, using the completeness relation for a momentum basis $\{p_i\}$ we can write each of the factors as

$$\begin{aligned}\langle x_2|\mathrm{e}^{-\mathrm{i}H\delta t}|x_1\rangle &= \int \mathrm{d}p_1 \mathrm{e}^{i(\delta t/2m)p_1^2}\mathrm{e}^{-\mathrm{i}\delta t V(x_1)}\langle x_2|p_1\rangle\langle p_1|x_1\rangle \\ &= \int \frac{\mathrm{d}p_1}{2\pi}\mathrm{e}^{-\mathrm{i}H(p_1,x_1)\delta t}\mathrm{e}^{\mathrm{i}p_1(x_2-x_1)}\,,\end{aligned} \tag{2.107}$$

so that in total

$$\langle x'', t''|x', t'\rangle = \int \prod\nolimits_i \mathrm{d}x_i \prod\nolimits_j \frac{\mathrm{d}p_j}{2\pi}\mathrm{e}^{\mathrm{i}p_j(x_{j+1}-x_j)}\mathrm{e}^{-\mathrm{i}H(p_j,x_j)\delta t}\,.$$

Now take the limit $\delta t \to 0$ so that $\frac{dx_j}{dt} = \frac{x_{j+1}-x_j}{\delta t}$. Then the second product can be written as an integral in the exponential. The integrals over the discrete spaces and momenta x_j and p_j become integrals over possible paths between x' and x'' and the respective initial and final momenta

$$\begin{aligned}\langle x'', t''|x', t'\rangle &= \int \mathcal{D}x\mathcal{D}p \exp\left(\mathrm{i}\int_{t'}^{t''} \mathrm{d}t(p(t)\dot{x}(t) - H(p(t), x(t))\right) \\ &= \int \mathcal{D}x \exp\left(\mathrm{i}\int_{t'}^{t''} \mathrm{d}t L(\dot{x}(t), x(t))\right)\,,\end{aligned} \tag{2.108}$$

where we have assumed H to be of the order p^2 and moved the constant from the p integration into the path integration measure $\mathcal{D}x$. The integral over the paths $x(t)$ is called a path integral.

For time dependent operators, time ordering has to be used, for example

$$\int \mathcal{D}x\mathcal{D}p\, x(t_1)x(t_2)\mathrm{e}^{\mathrm{i}S} = \langle x'', t''|\mathrm{T}[\hat{x}(t_1)\hat{x}(t_2)]|x', t'\rangle\,. \tag{2.109}$$

We can also calculate the evolution of the ground state $|0\rangle$ under a force f, in which case we consider the limit $t' \to -\infty$ and $t'' \to \infty$, which using equation (2.108) results in

$$\langle 0|0\rangle_f = \int \mathcal{D}x\mathcal{D}p \exp\left(\mathrm{i}\int_{-\infty}^{\infty} \mathrm{d}t L(\dot{x}(t), x(t)) + fx\right)\,. \tag{2.110}$$

We can then get operator expectation values using the chain rule for the functional

derivative, see also appendix A, so that for example

$$\begin{aligned}\langle 0|T\hat{x}(t_1)\hat{x}(t_2)|0\rangle &= \int \mathcal{D}x\mathcal{D}p\, x(t_1)x(t_2)\exp\left(\mathrm{i}\int_{-\infty}^{\infty}\mathrm{d}t L(\dot{x}(t),x(t))+fx\right)\\ &= \frac{1}{\mathrm{i}}\frac{\delta}{\delta f(t_1)}\frac{1}{\mathrm{i}}\frac{\delta}{\delta f(t_2)}\langle 0|0\rangle_f\Big|_{f=0}.\end{aligned} \tag{2.111}$$

Similarly with appropriate dependence on $\hat{p}$, $\langle 0|0\rangle_f$ can be simplified to

$$\langle 0|0\rangle_f = \exp\left(\mathrm{i}\int_{-\infty}^{\infty}\mathrm{d}t L_1\left(\frac{1}{\mathrm{i}}\frac{\delta}{\delta f(t)}\right)\right)\int \mathcal{D}x\exp\left(\mathrm{i}\int_{-\infty}^{\infty}\mathrm{d}t\left(L_0(\dot{x},x)+fx\right)\right). \tag{2.112}$$

Free Fields We now want to use the definition of a path integral for point particles we established in equation (2.108) to define a path integral over free fields in a similar manner. The path integral over field configurations will allow us to calculate vacuum expectation values of field operators, as given in equation (2.104). In the case of fields, we have to consider the Lagrangian density $\mathcal{L}$. The position $x(t)$ and the position operator $\hat{x}(t)$ are replaced by the scalar field or field operator, which we both denote by $\varphi(x)$ and the force by a source $J(x)$. Consider a free scalar field with Lagrangian

$$\mathcal{L}_0 = -\frac{1}{2}\partial^\mu\varphi\partial_\mu\varphi - \frac{1}{2}m^2\varphi^2, \tag{2.113}$$

then the evolution of the ground state in presence of the source J is given by

$$Z_0(J) \equiv \langle 0|0\rangle_J = \int \mathcal{D}\varphi\, \mathrm{e}^{\mathrm{i}S_0}, \tag{2.114}$$

with the action

$$S_0 = \int \mathrm{d}^4x(\mathcal{L}_0 + J\varphi), \tag{2.115}$$

where we now integrate over all possible field configurations $\varphi(x)$. Fourier transforming the fields

$$\tilde{\varphi}(p) = \int \mathrm{d}^4x \mathrm{e}^{-\mathrm{i}px}\varphi(x), \tag{2.116}$$

and introducing a shift

$$\tilde{\chi}(p) = \tilde{\varphi}(p) - \frac{\tilde{J}(p)}{p^2+m^2}, \tag{2.117}$$

the action becomes

$$S_0 = \frac{1}{2}\int \frac{\mathrm{d}^4p}{(2\pi)^4}\left(\frac{\tilde{J}(p)\tilde{J}(-p)}{p^2+m^2} - \tilde{\chi}(p)(p^2+m^2)\tilde{\chi}(-p)\right), \tag{2.118}$$

where the χ dependent part corresponds to the Lagrangian $\mathcal{L}_0$ without a source. We impose the normalization condition

$$\int \mathcal{D}\varphi \exp\Big(-\frac{\mathrm{i}}{2}\int \frac{\mathrm{d}^4p}{(2\pi)^4}\tilde{\chi}(p)(p^2+m^2)\tilde{\chi}(-p)\Big) = \langle 0|0\rangle_0 = 1\,, \tag{2.119}$$

so that

$$\begin{aligned} Z_0(J) &= \exp\left(\frac{\mathrm{i}}{2}\int \frac{d^4p}{(2\pi)^4}\frac{\tilde{J}(p)\tilde{J}(-p)}{p^2+m^2}\right) \\ &= \exp\left(\frac{\mathrm{i}}{2}\int \mathrm{d}^4x\mathrm{d}^4x' J(x)\Delta(x-x')J(x')\right), \end{aligned} \tag{2.120}$$

where

$$\Delta(x-x') = \int \frac{\mathrm{d}^4p}{(2\pi)^4}\frac{e^{\mathrm{i}p(x-x')}}{p^2+m^2} \tag{2.121}$$

is the Green's function for the Klein–Gordon equation

$$(\Box - m^2)\Delta(x-x') = \delta^4(x-x')\,. \tag{2.122}$$

$\Delta(x-x')$ is called the Feynman propagator for a free scalar theory.

We calculate vacuum expectation values for fields in analogy to equation (2.111). We get for example

$$\begin{aligned} \langle 0|\mathrm{T}\varphi(x_1)\varphi(x_2)|0\rangle &= \frac{1}{\mathrm{i}}\frac{\delta}{\delta J(x_1)}\frac{1}{\mathrm{i}}\frac{\delta}{\delta J(x_2)}Z_0(J)\Big|_{J=0} \\ &= \frac{1}{\mathrm{i}}\Delta(x_2-x_1) \end{aligned} \tag{2.123}$$

or

$$\begin{aligned} \langle 0|\mathrm{T}\varphi(x_1)\varphi(x_2)\varphi(x_3)\varphi(x_4)|0\rangle = \frac{1}{\mathrm{i}^2}(&\Delta(x_1-x_2)\Delta(x_3-x_4) \\ &+ \Delta(x_1-x_3)\Delta(x_2-x_4) \\ &+ \Delta(x_1-x_4)\Delta(x_2-x_3))\,. \end{aligned} \tag{2.124}$$

Interacting Fields Having established a way to calculate vacuum expectation values for field operators in a free field theory using the path integral formalism, we now want to look at vacuum expectation values for an interacting field theory.

We will notice that these give rise to scattering amplitudes—quantities, which determine S-matrix elements, as we present them for the process $q\bar{q} \to \gamma\gamma\gamma$ in section 5.1 and following. We further define Feynman diagrams, which are connected graphs that allow us to consistently parametrize and evaluate scattering amplitudes. If we have an interacting theory like a φ^3 theory

$$\mathcal{L} = -\frac{1}{2}\partial^\mu\varphi\partial_\mu\varphi - \frac{1}{2}m^2\varphi^2 + \frac{1}{6}g\varphi^3\,, \tag{2.125}$$

to get the ground state evolution, we need again to evaluate the path integral

$$Z(J) = \int \mathcal{D}\varphi\, \exp\left(\mathrm{i}\int \mathrm{d}^4x[\mathcal{L}_0 + \mathcal{L}_1 + J\varphi]\right)\,, \tag{2.126}$$

where

$$\mathcal{L}_0 = -\frac{1}{2}\partial^\mu\varphi\partial_\mu\varphi - \frac{1}{2}m^2\varphi^2 \tag{2.127}$$

is the free field Lagrangian. In analogy to equation (2.118) we can write $Z(J)$ in terms of the path integral for the free theory $Z_0(J)$ from equation (2.120) and a prefactor to

$$Z(J) = \exp\left(\mathrm{i}\int \mathrm{d}^4x\mathcal{L}_1\left(\frac{1}{\mathrm{i}}\frac{\delta}{\delta J(x)}\right)\right) Z_0(J)\,. \tag{2.128}$$

We can then expand the exponentials

$$\begin{aligned} Z(J) &= \exp\left(\frac{\mathrm{i}}{6}g\int \mathrm{d}^4x\left(\frac{1}{\mathrm{i}}\frac{\delta}{\delta J(x)}\right)^3\right) Z_0(J) \\ &= \left(\sum_{V=0}^{\infty}\frac{1}{V!}\left(\frac{\mathrm{i}g}{6}\int \mathrm{d}^4x\left(\frac{1}{\mathrm{i}}\frac{\delta}{\delta J(x)}\right)^3\right)^V\right) \\ &\quad\times\left(\sum_{P=0}^{\infty}\frac{1}{P!}\left(\frac{\mathrm{i}}{2}\int \mathrm{d}^4x'\mathrm{d}^4x'' J(x')\Delta(x'-x'')J(x'')\right)^P\right)\,. \end{aligned} \tag{2.129}$$

Now consider a particular term in the sum over V and P. Every of them has $2P$ sources from the second factor, from which $3V$ vanish after applying the functional derivatives from the first factor. In total, in every term we have

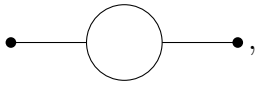
$$E = 2P - 3V \tag{2.130}$$

sources left. We see that not every combination of the terms of the two sums yields a non-vanishing result. Diagrams can represent these non-vanishing terms. For a fixed number of external sources and fixed power of g, every P-term is an edge connecting to a V-term. When all functional derivatives have acted on

one source each, every V-term connects to three P-terms, or twice to the same P-term and once to another P-term. If there is no V-term drawn at the end of a P-term, this represents an external source. These diagrams C_I are called Feynman diagrams, the V-terms vertices and the P-terms propagators. With their help, we can write the path integral as

$$Z(J) = \exp\Big(\sum\nolimits_I C_I\Big)\,, \tag{2.131}$$

where the sum is over all connected diagrams with at least two sources and omitting so-called tadpole diagrams, so subdiagrams with a single source attached to them. For example, for $E = 2$ and $V = 2$ by equation (2.130) we get $P = 4$ and the only diagram contributing is

 ,

the so-called bubble diagram. From $Z(J)$ we then can compute vacuum expectation values, that is for example for four fields

$$\langle 0|T\varphi(x_1)\varphi(x_2)\varphi(x_1')\varphi(x_2')|0\rangle = \frac{1}{\mathrm{i}}\frac{\delta}{\delta J(x_1)}\frac{1}{\mathrm{i}}\frac{\delta}{\delta J(x_2)}\frac{1}{\mathrm{i}}\frac{\delta}{\delta J(x_1')}\frac{1}{\mathrm{i}}\frac{\delta}{\delta J(x_2')}Z(J)\bigg|_{J=0}\,, \tag{2.132}$$

and finally, using the LSZ reduction formula (2.104)

$$\begin{aligned}\langle f|S|i\rangle = \mathrm{i}^4 \int \mathrm{d}^4x_1\mathrm{d}^4x_2\mathrm{d}^4x_1'\mathrm{d}^4x_2'\mathrm{e}^{\mathrm{i}(p_1x_1+p_2x_2-p_1'x_1'-p_2'x_2')} \\ \times(-\partial_{x_1}^2+m^2)(-\partial_{x_2}^2+m^2)(-\partial_{x_1'}^2+m^2)(-\partial_{x_2'}^2+m^2) \\ \times\langle 0|T\varphi(x_1)\varphi(x_2)\varphi(x_1')\varphi(x_2')|0\rangle\,,\end{aligned} \tag{2.133}$$

which using

$$(-\partial_{x_i}^2+m^2)\Delta(x_i-y) = \delta^4(x_i-y)\,, \tag{2.134}$$

and

$$\Delta(y-z) = \int\frac{\mathrm{d}^4p}{(2\pi)^4}\frac{\mathrm{e}^{\mathrm{i}p(y-z)}}{p^2+m^2} \tag{2.135}$$

then becomes

$$\begin{aligned}\langle f|S|i\rangle &= \mathrm{i}g^2(2\pi)^4\delta^4(p_1+p_2-p_1'-p_2') \\ &\quad\times\left(\frac{1}{(p_1+p_2)^2+m^2}+\frac{1}{(p_1-p_1')^2+m^2}+\frac{1}{(p_1-p_2')^2+m^2}\right)+\mathcal{O}(g^4) \\ &= (2\pi)^4\delta^4(p_{\mathrm{in}}-p_{\mathrm{out}})\mathcal{M}_{f\leftarrow i}\,.\end{aligned} \tag{2.136}$$

Every of the terms from $\mathcal{M}_{f\leftarrow i}$ comes from a specific Feynman diagram, for the s-, the t- and for the u-channel. Starting from the other side, we can also draw

all diagrams corresponding to model-specific Feynman rules and then calculate $\langle f|S|i\rangle$ according to an equation similar to (2.136). The process specific quantities $\mathcal{M}_{f\leftarrow i}$ are called scattering amplitudes and are the core of a cross section computation. According to equation (2.129), contributions to the scattering matrix element of higher order in the coupling constants have to come from Feynman diagrams with more vertices. For a fixed number of external legs this is achieved by gradually adding more internal lines, or loops, to the Feynman diagrams. Thus, the fixed-order contributions to $\mathcal{M}_{f\leftarrow i}$ are the tree-level (or no-loop) amplitudes $\mathcal{M}^{(0)}$, the one-loop amplitudes $\mathcal{M}^{(1)}$, the two-loop amplitudes $\mathcal{M}^{(2)}$ and so on. A key goal of this thesis is the computation of the planar two-loop amplitudes relevant for the process $q\bar{q}\to\gamma\gamma\gamma$.

Abelian Bosonic Action The partition function for a free field A_μ is given by

$$Z_0(J)=\int\mathcal{D}A\mathrm{e}^{\mathrm{i}\int\mathrm{d}^4x(-\frac{1}{4}F^{\mu\nu}F_{\mu\nu}+J^\mu A_\mu)}\,, \tag{2.137}$$

In order to find an expression for the propagator $\Delta^{\mu\nu}(x-x')$, as in equation (2.118) we Fourier transform the field A_μ to arrive at

$$S_0=\frac{1}{2}\int\frac{\mathrm{d}^4p}{(2\pi)^4}\left(-\tilde{A}_\mu(p)(p^2g^{\mu\nu}-p^\mu p^\nu)\tilde{A}_\nu(-p)+\tilde{J}^\mu(p)\tilde{A}_\mu(-p)+\tilde{J}^\mu(-p)\tilde{A}_\mu(p)\right). \tag{2.138}$$

In analogy to the free scalar partition function (2.120), introducing a shift to $\tilde{A}_\mu$ and normalizing, we then can write

$$Z_0(J)=\exp\left(\frac{\mathrm{i}}{2}\int\mathrm{d}^4x\mathrm{d}^4x'J_\mu(x)\Delta^{\mu\nu}(x-x')J_\nu(x')\right). \tag{2.139}$$

with the bosonic propagator given by

$$\Delta_{\mu\nu}(x-x')=\int\frac{\mathrm{d}^4p}{(2\pi)^4}\mathrm{e}^{\mathrm{i}p(x-x')}\frac{g_{\mu\nu}}{p^2}\,. \tag{2.140}$$

Now for $J^\mu(x)$ the continuity equation $\partial_\mu J^\mu(x)=0$ holds, which means we have $p_\mu\tilde{J}^\mu(p)=0$ for its Fourier transform. That is, we have the freedom to add products of p^μ to the propagator, without changing the result, so that we can define

$$\Delta^{\mu\nu}(x-x')=\int\frac{\mathrm{d}^4p}{(2\pi)^4}\mathrm{e}^{\mathrm{i}p(x-x')}\frac{1}{p^2}\left(g_{\mu\nu}-(1-\xi)\frac{p_\mu p_\nu}{p^2}\right). \tag{2.141}$$

We see that setting $\xi=0$ makes the integrand of the propagator (2.141) a projector onto the space orthogonal to p^μ. Since it also holds that $p_\mu\tilde{J}^\mu(p)=0$,

only $\tilde{A}_\mu(p)$ with $p^\mu \tilde{A}_\mu(p) = 0$ or $\partial^\mu A_\mu(p) = 0$ contribute the integration over $\mathcal{D}A$ within $Z_0(J)$. Thus, $\xi = 0$ corresponds to the Lorenz gauge. The choice $\xi = 1$ with the simpler propagator (2.140) is called the Feynman gauge, which we use in our calculations.

Fermionic Fields The QED Lagrangian (2.78) also involves fermionic fields, which need to be quantized. For this, we need to compute the purely fermionic partition function

$$Z_0(\xi, \bar{\xi}) = \int \mathcal{D}\psi \mathcal{D}\bar{\psi} \exp\left(\mathrm{i} \int \mathrm{d}^4x \mathcal{L}(\psi, \partial_\mu \psi)\right) , \tag{2.142}$$

where $\mathcal{L}$ is from equation (2.68). Using the anticommutation of the spinor fields (2.57) and with $\Delta_\psi(x - x')$ the Feynman propagator solving

$$\big(\mathrm{i}(\gamma^\mu \partial_\mu)_{ij} - m\delta_{ij}\big)\Delta_{\psi jp}(x - x') = \delta_{ip}\delta^4(x - x') , \tag{2.143}$$

we get

$$Z_0(\xi, \bar{\xi}) = \det(\mathrm{i}\gamma^\mu \partial_\mu - m) \exp\left(\mathrm{i} \int \mathrm{d}^4x \mathrm{d}^4x' \bar{\xi}(x) \Delta_\psi(x - x') \xi(x')\right) , \tag{2.144}$$

where

$$\int \mathcal{D}\psi \mathcal{D}\bar{\psi} \exp\left(\mathrm{i} \int \mathrm{d}^4x \bar{\psi}(\mathrm{i}\gamma^\mu \partial_\mu - m)\psi\right) = \det(\mathrm{i}\gamma^\mu \partial_\mu - m) \tag{2.145}$$

does not depend on the participating fields ξ and $\bar{\xi}$, so that we can move it into the normalization of $Z_0(\xi, \bar{\xi})$ by identifying

$$Z_0(\xi, \bar{\xi}) \equiv \det(\mathrm{i}\gamma^\mu \partial_\mu - m) Z_0(\xi, \bar{\xi}) . \tag{2.146}$$

Quantum Chromodynamics In non-abelian gauge theories, we encounter ambiguity in the definition of our fields because of the gauge symmetry of our Lagrangian. When performing the path integral, configurations equivalent under gauge symmetry are taken into account, thus overcounting. When the Lagrangian of a U(1) theory was symmetrical under the transformation (2.74), as in equation (2.80) the gauge field $G_\mu(x) = G^a_\mu(x) T^a$ of a SU(3) theory transforms according to

$$G'_\mu(x) = U(x) G_\mu(x) U^\dagger(x) + \frac{\mathrm{i}}{g_\mathrm{s}} U(x) \partial_\mu U^\dagger(x) , \tag{2.147}$$

where $U(x) = \exp(-\mathrm{i}g_\mathrm{s}u^a(x)T^a)$ has values in SU(3). If we expand $U(x)$ according to

$$U(x) = 1 - \mathrm{i}g_\mathrm{s}u^a(x)T^a + \mathcal{O}(u^2) \tag{2.148}$$

we can write the transformation (2.147) as

$$G'^a_\mu(x) = G^a_\mu(x) - \big(\delta^{ac}\partial_\mu + g_\mathrm{s}f^{abc}G^b_\mu(x)\big)u^c(x) \equiv G^a_\mu(x) - D^{ac}_\mu u^c(x) \tag{2.149}$$

in the infinitesimal limit. Now we want to fix the Gauge as we did for abelian gauge fields. For this, we need to define a gauge fixing condition $F^a(G) = 0$. Since we want to use the simple propagator (2.140), we extend the Lorenz gauge and set

$$F^a(G(x), x) = \partial^\mu G^a_\mu(x) - g^a(x)\,, \tag{2.150}$$

where $g^a(x)$ will allow us to introduce a gauge fixing parameter ξ as in equation (2.141). Applying (2.149) to $F^a(G)$ we find that $F^a(G)$ transforms under an infinitesimal gauge transformation $U(x)$ according to

$$F'^a(G) = F^a(G) - \partial^\mu D^{ab}_\mu u^b\,. \tag{2.151}$$

Let $u_{F'}(G)$ a solution to $F'^a(G) = 0$. To fix our gauge choice we set

$$\begin{aligned} Z &= \int \mathcal{D}G\mathcal{D}u\delta\big(u - u_{F'}(G)\big)\exp\left(\mathrm{i}\int \mathrm{d}^4x\mathcal{L}\right) \\ &\equiv \int \mathcal{D}G \det\left(\frac{\delta F'}{\delta u}\right)\delta(F')\exp\left(\mathrm{i}\int \mathrm{d}^4x\mathcal{L}\right)\,. \end{aligned} \tag{2.152}$$

Here the functional derivative yields

$$\frac{\delta F'^a(G(x), x)}{\delta u^b(x')} = -\partial^\mu D^{ab}_\mu \delta(x - x')\,, \tag{2.153}$$

and does not depend on u, which is why we could absorb the u integral in the normalization of Z. We can now evaluate the determinant $\det\left(\frac{\delta F'}{\delta u}\right)$ using fermionic fields c, $\bar{c}$ as in equation (2.145), to get

$$Z = \int \mathcal{D}G\mathcal{D}c\mathcal{D}\bar{c}\exp\left(\mathrm{i}\int \mathrm{d}^4x\big(\mathcal{L} + \bar{c}^a\partial^\mu D^{ab}_\mu c^b\big)\right)\,, \tag{2.154}$$

where the unphysical fields c and $\bar{c}$ are called Faddeev–Popov ghost fields. If we now add a term $\frac{-\mathrm{i}}{2\xi}\int \mathrm{d}^4x g^a g^a$ to the Lagrangian, so that, taking into account the gauge fixing condition $F(G) = 0$,

$$Z = \int \mathcal{D}G\mathcal{D}c\mathcal{D}\bar{c}\exp\left(\mathrm{i}\int \mathrm{d}^4x\left(\mathcal{L} + \bar{c}\partial^\mu D^{ab}_\mu c - \frac{1}{2\xi}\partial^\mu G^a_\mu \partial^\nu G^a_\nu\right)\right)\,, \tag{2.155}$$

$$\nu,\ b \quad p \quad \mu,\ a \qquad = \frac{\mathrm{i}}{p^2}\left(-g^{\mu\nu} + (1-\xi)\frac{p^\mu p^\nu}{p^2}\right)\delta^{ab}\,, \qquad\qquad j \xrightarrow[p]{} i \qquad = \frac{\mathrm{i}\delta^{ij}(\not{p}+m)}{p^2-m^2}\,,$$

$$\begin{array}{l} \rho \\ \nu \quad q \quad c \\ b \quad p \quad k \\ \mu \\ a \end{array} \qquad = \frac{g_\mathrm{s} f^{abc}}{\sqrt{2}}\left[g^{\mu\nu}(k-p)^\rho + g^{\nu\rho}(p-q)^\mu + g^{\rho\mu}(q-k)^\nu\right],$$

$$\begin{array}{llll} & a & \nu & \\ \mu & & & b \\ & & & \\ c & & & \sigma \\ & \rho & d & \end{array} \qquad = \begin{array}{l} -\mathrm{i}g_\mathrm{s}^2\left[f^{abe}f^{cde}(g^{\mu\rho}g^{\nu\sigma} - g^{\mu\sigma}g^{\nu\rho}) + f^{ace}f^{bde}(g^{\mu\nu}g^{\rho\sigma} - g^{\mu\sigma}g^{\nu\rho})\right. \\ \qquad \left. + f^{ade}f^{bce}(g^{\mu\nu}g^{\rho\sigma} - g^{\mu\rho}g^{\nu\sigma})\right], \end{array}$$

$$\begin{array}{lll} & \mu & \\ & & a \\ j & & \\ & & i \end{array} \qquad = \frac{\mathrm{i}g_\mathrm{s}}{\sqrt{2}}\gamma^\mu T^a_{ij}\,, \qquad\qquad \begin{array}{l} \mu \\ \\ \\ \end{array} \qquad = \mathrm{i}e\gamma^\mu\,,$$

Table 2.1.: Feynman rules for strongly interacting particles [45]. All momenta are taken outgoing. Wavy lines represent photons, coily lines gluons, and the incoming and outgoing arrows antiquarks and quarks, respectively.

Z yields a propagator as in equation (2.140), which is given by

$$\Delta^{ab}_{\mu\nu}(x-x') = \int \frac{\mathrm{d}^4 p}{(2\pi)^4}\mathrm{e}^{\mathrm{i}p(x-x')}\frac{g_{\mu\nu}\delta^{ab}}{p^2}\,, \tag{2.156}$$

when choosing $\xi = 1$.

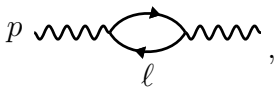
As will become apparent in section 3.6, we will work with gauge-invariant tree-level building blocks in our multi-loop calculations. Because of that, the introduction of Fadeev–Popov ghost fields is not required for our purposes. The Feynman rules for the QCD Lagrangian without the ghost fields are listed in table 2.1. In addition to the rules from table 2.1 we have to integrate over every internal momentum flowing through a closed loop in the diagram.

2.4. Renormalization

In the previous section, we have seen that we can calculate S-matrix elements using scattering amplitudes, which can be parametrized through Feynman diagrams. Because of the form of Feynman propagators, loop Feynman diagrams,

which naturally appear in higher-order terms of the expansion of an amplitude in the coupling constant, will be singular in certain limits of the momentum flowing through the loop. There are two types of divergences we can encounter: infrared (IR) and ultraviolet (UV) divergences. IR divergences stem from low energy, or soft, loop momenta, or loop momenta collinear with the momentum of an external particle. They will cancel with divergences coming from additional real particles at the same order of the coupling constant. UV divergences appear due to large loop momenta. They need to be removed from a scattering amplitude through a redefinition of the corresponding Lagrangian, a procedure known as renormalization. This redefinition and the introduction of new parameters of the theory gives rise to running couplings, that is, couplings, dependent on the energy scale of the experiment. The evolution of the couplings with the energy scale has significant consequences for the underlying theory, which we will address at the end of this section. Before renormalizing, the amplitude's divergences must be made explicit in terms of a regularizing parameter. For example, we could use only values for the momenta within the range we can observe; that is, we cut off values of $|\ell| < \Lambda$ where Λ is then the regulator. However, this cutoff regularization will break the Lorentz invariance of our result. Another common regularization technique is Pauli–Villars regularization [51]. Pauli and Villars introduce a mass as the regulator. Pauli–Villars regularization requires additional mass terms in the Lagrangian to account for the artificial mass. QCD results are usually given using dimensional regularization. It lacks both prementioned disadvantages and can regularize both UV and IR divergences. We will work with dimensionally regularized amplitudes. The dimension of the loop momentum space is then viewed as a complex parameter D of the amplitude. Accordingly, internal states are in D_s dimensions. In that case, there is an ambiguity in defining D_s and how to embed external states in the D_s-dimensional space, which gives rise to different regularization schemes.

Dimensional Regularization If we just calculate a loop diagram using Feynman integrals, it can become infinite. Consider for example a QED bubble diagram



,

which using Feynman rules is given by

$$\mathcal{I}_{\text{bub}} = \int \frac{\mathrm{d}^4\ell}{(2\pi)^4} \frac{\mathcal{N}(p,\ell)}{\ell^2(\ell-p)^2}\,, \tag{2.157}$$

where the numerator $\mathcal{N}(p,\ell)$ is at most quadratic in ℓ. Using polar coordinates we arrive at $\mathcal{I}_{\text{bub}} \propto \int \ell \mathrm{d}\ell$ which diverges. When the integral diverges in the limit

of large momenta we call it UV divergent. Similarly, a scalar triangle integral

$$\mathcal{I}_{\text{tri}} = \int \frac{\mathrm{d}^4\ell}{(2\pi)^4} \frac{1}{\ell^2(\ell - p_1)^2(\ell - p_{12})^2} \tag{2.158}$$

is divergent in the limit of small internal momenta. Either in this case, or if an integral becomes divergent when the internal momentum becomes collinear with an external momentum, we say it is IR divergent. Since physical observables must be finite, the divergences of the bare amplitudes are unphysical and belong to quantities, which are not accessible by experiment. There are multiple ways the divergences can be moved away from the physically observable parts. In order to do so, one needs first to regularize the amplitude, that is to represent the poles in terms of a regularizing variable. The most convenient way to achieve that is by dimensional regularization. Here, the regulator is the dimension of the loop integration. Actually we can only integrate over integer dimensions. However if we parametrize the integrand by the solid angle Ω and the radius, ℓ, and use the transformation

$$\int \mathrm{d}^D\ell = \int \mathrm{d}\Omega \int \ell^{D-1}\mathrm{d}\ell\,, \tag{2.159}$$

the integral becomes a function of D, which can then be analytically continued to arbitrary D. In order to keep the coupling dimensionless, when going from four to D dimensions, we introduce a mass dimension one parameter μ and replace the coupling

$$g \to g\mu^{\frac{4-D}{2}}\,. \tag{2.160}$$

Imagine a tadpole diagram of a scalar theory,

$$I_{\text{tad}} = \int \frac{\mathrm{d}^D\ell}{(2\pi)^D} \frac{1}{\ell^2 + m^2}\,, \tag{2.161}$$

which is obviously UV divergent for $D \geq 2$. Now we want to write down the integral first in terms of an arbitrary dimension D. Using the transformation formula (2.159), we can replace $\mathrm{d}^D\ell$ by spherical coordinates [45]

$$\begin{aligned} \mathrm{d}^D\ell &= \ell^{D-1}\mathrm{d}\ell\mathrm{d}\Omega \\ &= \ell^{D-1}\mathrm{d}\ell\mathrm{d}\varphi \sin\vartheta_1 \mathrm{d}\vartheta_1 \sin^2\vartheta_2 \mathrm{d}\vartheta_2 \cdots \sin^{D-2}\vartheta_{D-2}\mathrm{d}\vartheta_{D-2}\,, \end{aligned} \tag{2.162}$$

with $0 < \ell < \infty$, $0 < \varphi < 2\pi$ and $0 < \vartheta_i < \pi$. Now in this simple case the integrand only depends on the magnitude of the loop momentum ℓ and we can

integrate the ϑ_i using the identity

$$\int_0^\pi \mathrm{d}\vartheta_i \sin^i \vartheta_i = \frac{\Gamma\left(\frac{i+1}{2}\right)\Gamma\left(\frac{1}{2}\right)}{\Gamma\left(\frac{i+2}{2}\right)} . \tag{2.163}$$

With this using $\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$ and canceling terms in the numerator against the denominator we get

$$\Omega = 2\pi \prod_{i=1}^{D-2} \int_0^\pi \sin^i \vartheta_i \mathrm{d}\vartheta_i = \frac{2\pi^{D/2}}{\Gamma\left(\frac{D}{2}\right)} , \tag{2.164}$$

that is

$$I_{\mathrm{tad}}(D) = \frac{\Omega}{(2\pi)^D} \int_0^\infty \mathrm{d}\ell \ell^{D-1} \frac{1}{\ell^2 + m^2} . \tag{2.165}$$

now this integral can be expressed using the beta function

$$\mathrm{B}(\alpha, \gamma) = \int_0^\infty \mathrm{d}y y^{\alpha-1}(1+y)^{-\alpha-\gamma} = \frac{\Gamma(\alpha)\Gamma(\gamma)}{\Gamma(\alpha+\gamma)} \tag{2.166}$$

so that

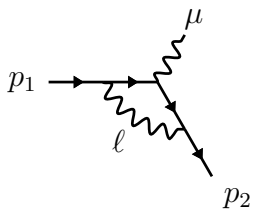
$$I_{\mathrm{tad}}(D) = \frac{(m^2)^{D/2-1}}{(4\pi)^{D/2}} \Gamma(1 - D/2) , \tag{2.167}$$

which we can analytically continue to a function on the complex plane with poles at $D = 4$. We look at dimensionally regularized integrals as functions of ϵ so that $D = 4 - 2\epsilon$ rather than as functions of the dimension D, so that they have poles at $\epsilon = 0$. The later representation we can expand in ϵ, that is

$$I_{\mathrm{tad}}(\epsilon) = \frac{(m^2)^{1-\epsilon}}{(4\pi)^{\epsilon-2}} \Gamma(\epsilon - 1) = \frac{m^2}{16\pi^2} \left(-\frac{1}{\epsilon} - 1 + \gamma_E + \log \frac{m^2}{4\pi} \right) + \mathcal{O}(\epsilon) , \tag{2.168}$$

With $\gamma_E = -\Gamma'(1)$ the Euler–Mascheroni constant.

Consider the NLO approximation of the massless quark-photon-vertex as in section 2.3.



Using Feynman rules, it is given by [45]

$$V^\mu = -(e\mu^{\frac{4-D}{2}})^3 \int \frac{\mathrm{d}^D \ell}{(2\pi)^D} \frac{\bar{u}(p_2)\gamma^\nu(\not{\ell} + \not{p}_2)\gamma^\mu(\not{\ell} - \not{p}_1)\gamma^\nu v(p_1)}{(\ell + p_2)^2(\ell - p_1)^2\ell^2} . \tag{2.169}$$

In order to bring the integral into a form we can evaluate we use Feynman parameters, given by

$$\frac{1}{\rho_1 \cdots \rho_n} = (n-1)! \int_0^1 \mathrm{d}x_1 \cdots \int_0^{x_{n-2}} \mathrm{d}x_{n-1} \frac{1}{\Big(\rho_1 + x_1(\rho_2 - \rho_1) + \cdots + x_{n-1}(\rho_n - \rho_{n-1})\Big)^n} \,. \tag{2.170}$$

Using the fact that the quarks are massless and performing the Clifford algebra in the numerator, we arrive at

$$V^\mu = -2\mathrm{i}\gamma^\mu e^2 \mu^{4-D} \int \mathrm{d}x_1 \int \mathrm{d}x_2 \frac{\mathrm{d}^D \ell}{(2\pi)^D} \frac{\frac{(D-2)^2}{D}\ell^2 + s((2-D)xy + 2x + 2y - 2)}{(\ell^2 + sxy)^3} \,, \tag{2.171}$$

where the first, ℓ^2 proportional, term V_1^μ of the integrand is UV divergent, whereas the second s proportional V_2^μ is IR divergent. In particular, we can perform the IR divergent integral over the loop momentum using

$$\int \frac{\mathrm{d}^D \ell}{(2\pi)^D} \frac{1}{(\ell^2 - \Delta)^3} = \frac{-\mathrm{i}}{2(4\pi)^{\frac{D}{2}}} \frac{1}{\Delta^{3-\frac{D}{2}}} \Gamma\left(3 - \frac{D}{2}\right) \,, \tag{2.172}$$

where $\Delta = -sxy$ yielding

$$V_2^\mu = \frac{\gamma^\mu e^2}{8\pi^2} (4\pi)^\epsilon \left(-\frac{\mu^2}{s}\right)^\epsilon \left(-\frac{4}{\epsilon^2} + \frac{-4 + 2\gamma_E}{\epsilon} + \frac{-54 + 24\gamma_E - 6\gamma_E^2 + \pi^2}{12} + \mathcal{O}(\epsilon)\right) \,. \tag{2.173}$$

So using regularization, we can make the divergence of the integral explicit.

There is the question of how many dimensions D_s the state space has and what happens with the external particles. In conventional dimensional regularization (CDR) both, the external and internal particle states are in $D_s = D = 4 - 2\epsilon$ dimensions. CDR comes in handy when implementing the amplitude calculation because one does not have to distinguish between internal and external particles and can apply the same Feynman rules. In the case of CDR, however, we would also need more helicity states to consider when dealing with helicity amplitudes. There the t' Hooft–Veltman (HV) scheme [52] is better suited, which leaves the external states in four dimensions and only promotes the internal states to $D_s = 4 - 2\epsilon$ dimensions. In the context of numerical unitarity, there is also the four-dimensional helicity scheme [53]. It uses the fact that tree amplitudes play an essential role for numerical unitarity and sets $D_s = 4$. While one must state the utilized scheme, users can transform from one consistent scheme to another. Here we use the HV scheme.

Counterterm Lagrangian After having regularized the results, they are still diverging for $\epsilon \to 0$. To get measurable quantities, we need to apply renormalization in the next step. In this process, we introduce additional unphysical renormalization parameters of the Lagrangian. These parameters allow us to rewrite the bare Lagrangian of our theory in terms of a renormalized Lagrangian and a counterterm Lagrangian, which we will define in the following. The counterterm Lagrangian is defined so that contributions from it cancel divergences of the original bare Lagrangian, making its definition ambiguous. We discuss different renormalization schemes, addressing this ambiguity.

In (2.168) the mass m is contributing to the pole. It thus needs to be renormalized for the result to match the experiment. In (2.169) the electric charge e needs to be renormalized. In a general context, this means renormalizing a coupling g of the theory. Also, the fields ψ and A_μ themselves have to be renormalized. In that process, we replace the bare fields, masses, and couplings of the original Lagrangian with renormalized quantities. We index the bare quantities with a 0 and renormalized quantities with an R. The bare and renormalized quantities are related by normalization factors Z_i [45], that is

$$\psi_0 = \sqrt{Z_\psi}\psi^{\mathrm{R}}, \qquad A_{\mu,0} = \sqrt{Z_A}A_\mu^{\mathrm{R}}, \qquad m_0 = Z_m m_{\mathrm{R}}, \qquad g_0 = \mu^{\frac{4-D}{2}} Z_g g_{\mathrm{R}}\,. \tag{2.174}$$

If we now write $Z_i = 1 + \delta_i$, the Lagrangian for the bare fields $\mathcal{L}_0$ consists of two terms

$$\mathcal{L}_0 = \mathcal{L}_{\mathrm{R}} + \mathcal{L}_C\,, \tag{2.175}$$

with $\mathcal{L}_{\mathrm{R}}$ the renormalized Lagrangian and $\mathcal{L}_C$ the counterterm Lagrangian responsible for the cancellation of infinities. To this end, the counterterms δ_i are functions that depend on the regulators like ϵ so that no poles appear in the finite result. There is an ambiguity about how one can choose the counterterms. For example, one could add arbitrary functions of the regulators with no poles to them, and they still would cancel the poles stemming from $\mathcal{L}_0$. A choice of the renormalization scheme fixes these ambiguities. The renormalization scheme distinguishes different kinds of renormalization. It has to be stated when comparing different results. The scheme, which just cancels the poles with no additional functions added to the counterterms, is called the minimal subtraction scheme. When we renormalize, we use the $\overline{\mathrm{MS}}$ scheme, where constants are added to the counterterms to make the result simpler. Because of the solid angle integration as in equation (2.164), for every pole, there will always be the terms $\gamma_E - \log 4\pi$ as seen for example in equation (2.168). These terms are added to the counterterms of the minimal subtraction scheme to get the $\overline{\mathrm{MS}}$ scheme. The transition from the minimal subtraction scheme to the $\overline{\mathrm{MS}}$ scheme can also be achieved by

adjusting the renormalization scale μ as in equation (2.160) by

$$\mu^2 \to \mu^2 \frac{\mathrm{e}^{\gamma_E}}{4\pi} \,. \tag{2.176}$$

The renormalization scale is unphysical, and thus physical predictions cannot depend on it. Because of that, phenomenologists use it to measure the quality of a theoretical perturbative approximation. Cross sections are then obtained at various values for the scale, and they look at how their values change with the scale to estimate the error.

Running Couplings Having defined the parameters of the renormalized theory, we can look at the evolution of its couplings with the energy scale of the experiment. We will see that couplings evolve according to a renormalization group equation (RGE). The RGE tells us that the electric charge will increase with energy, while the strong coupling constant decreases with the energy scale. The latter phenomenon is known as asymptotic freedom. The RGE is characterized by employing a beta function, which defines the evolution of the couplings. We will give the beta function coefficients relevant for NNLO QCD.

When two particles interact, at leading-order, we describe the interaction with a single propagator. Now at higher-order in the coupling constant, we can extend this interaction with virtual particles, for example, adding a tadpole as in equation (2.168) or a bubble to the propagator line. As we discussed, these integrals are divergent. A good first approximation for all possible diagrams which can extend a propagator is to consider all possible bubbles added

where every of them, similarly to the tadpole diagram (2.168) will contribute a characteristic factor $x = \frac{g_0}{3\pi} \log \frac{Q^2}{m^2}$ where Q^2 is the momentum transfer of the propagator and $g_0 = g(\mu^2)$ is the bare coupling constant. That is, including all terms, we get a geometric series $1 + x + x^2 + \cdots = \frac{1}{1-x}$. This is the leading log approximation. Because the coupling g_0 accompanying the propagator cannot be measured directly, we can move the divergences inside there. We end up with

$$g(Q^2) = \frac{g(\mu^2)}{1 - \frac{g(\mu^2)}{3\pi} \log \frac{Q^2}{\mu^2}} \,. \tag{2.177}$$

This is an example for a running coupling, as it appears in QED. Equation (2.177) can equivalently be written as

$$\frac{1}{g(Q^2)} = \frac{1}{g(\mu^2)} - \frac{1}{3\pi} \log \frac{Q^2}{\mu^2} \,. \tag{2.178}$$

Differentiating this with respect to $\log Q^2$ we get

$$\frac{\mathrm{d}g}{\mathrm{d}Q^2} = \frac{g^2}{3\pi} \equiv \beta(g)\,, \tag{2.179}$$

which is the renormalization group equation, where

$$\beta(\alpha) = -\alpha\left(\beta_0\frac{\alpha}{4\pi} - \beta_1\left(\frac{\alpha}{4\pi}\right)^2 + \cdots\right) \tag{2.180}$$

is called the beta function and describes the evolution of the coupling with the energy transfer. From it we can see whether a coupling will increase or decrease with the energy. For QED $\beta(e) > 0$, or $\beta_0 < 0$, so the coupling will increase with energy.

For QCD, in addition to fermion loops, gluon loops need to be taken into account, which results in

$$\beta_0 = \frac{N_c}{3}\left(11 - 2\frac{N_f}{N_c}\right)\,. \tag{2.181}$$

Here $\beta_0 > 0$ so that the coupling will decrease with energy for QCD which is called asymptotic freedom. When one considers NNLO contributions as we do, the second coefficient of the beta function becomes important, which is

$$\beta_1 = \frac{N_c^2}{3}\left(17 - \frac{13}{2}\frac{N_f}{N_c}\right)\,. \tag{2.182}$$

With these coefficients, the QCD bare and renormalized couplings $\alpha_0 = g_{s,0}^2/4\pi$ and $\alpha_s = g_\mathrm{s}^2/4\pi$ are related by [54]

$$\alpha_0 S_\epsilon = \alpha_s\left(1 - \frac{\beta_0}{\epsilon}\frac{\alpha_s}{4\pi} + \left(\frac{\beta_0^2}{\epsilon^2} - \frac{\beta_1}{\epsilon}\right)\left(\frac{\alpha_s}{4\pi}\right)^2 + \mathcal{O}(\alpha_s^3)\right)\,, \tag{2.183}$$

with $S_\epsilon = (4\pi)^\epsilon \mathrm{e}^{-\epsilon\gamma_E}$ and $\gamma_E = -\Gamma'(1)$.

Renormalized virtual amplitudes are UV-finite; however, they still can have IR poles. According to the KLM theorem, these IR poles have to cancel against the IR poles of the real contributions to a cross section at the same order of expansion in the coupling constant. The IR pole structure of QCD amplitudes at NNLO is known for generic processes, which we will further discuss in section 3.8.

2.5. Cross Sections Revised

Now that we have defined the basic tools used in particle physics, we can get back to the measurable cross sections. As mentioned in section 2.1, we get a

cross section by

$$\sigma = \frac{\dot{P}}{\Phi}\,. \tag{2.184}$$

The differential cross section is then given by

$$\mathrm{d}\sigma = \frac{\mathrm{d}\dot{P}}{\Phi}\,, \tag{2.185}$$

with $\dot{P}$ the probability of a process taking place per unit time T and Φ the particle flux. Since the probability has to be normalized to one, it is given by

$$P = \frac{|\langle f|S|i\rangle|^2}{\langle f|f\rangle\langle i|i\rangle}\,. \tag{2.186}$$

Using equation (2.136) it holds that

$$|\langle f|S|i\rangle|^2 = [(2\pi)^4\delta^4(p_{\mathrm{in}} - p_{\mathrm{out}})]^2|\mathcal{M}_{f\leftarrow i}|^2\,.$$

Since one of the delta functions already sets the momenta on-shell, we can rewrite this as

$$= (2\pi)^4\delta^4(p_{\mathrm{in}} - p_{\mathrm{out}})(2\pi)^4\delta^4(0)|\mathcal{M}_{f\leftarrow i}|^2\,,$$

where writing

$$\delta(p) = \frac{1}{(2\pi)^4}\int \mathrm{d}^4x\mathrm{e}^{\mathrm{i}px} \tag{2.187}$$

with the space-time volume of integration VT for the second term we get

$$(2\pi)^4\delta^4(0) = VT\,. \tag{2.188}$$

A Lorentz invariant definition of the states $|i\rangle$ and $|f\rangle$ yields the normalization

$$\langle p|p\rangle = (2\pi)^3 2p^0\delta^3(0)\,, \tag{2.189}$$

which again using (2.187) yields

$$= 2p^0V\,. \tag{2.190}$$

With E_1 and E_2 the energies of the incoming particles we arrive at

$$\dot{P} = \frac{(2\pi)^4\delta^4(p_{\mathrm{in}} - p_{\mathrm{out}})|\mathcal{M}_{f\leftarrow i}|^2}{4E_1E_2V\prod_{j=1}^{n'}2p'^0_jV}\,, \tag{2.191}$$

The corresponding differential $\mathrm{d}\dot{P}$ within a phase space volume is then

$$\mathrm{d}\dot{P} = \frac{(2\pi)^4\delta^4(p_{\mathrm{in}} - p_{\mathrm{out}})V|\mathcal{M}_{f\leftarrow i}|^2}{4E_1E_2V^2\prod_{j=1}^{n'} 2p'^0_j V}\prod_{j=1}^{n'}\mathrm{d}^3p_j = \frac{|\mathcal{M}_{f\leftarrow i}|^2}{4E_1E_2V}\mathrm{dLIPS}_{n'}(p_1+p_2) \quad (2.192)$$

where we define the Lorentz invariant phase space differential to be

$$\mathrm{dLIPS}_{n'}(p) = (2\pi)^4\delta^4(p - p_{\mathrm{out}})\prod_{j=1}^{n'}\frac{\mathrm{d}^3p_j}{2p_j^0}\,. \quad (2.193)$$

Taking the second incoming particle to be at rest, the flux is given by

$$\Phi = \frac{|\mathbf{p}_1|}{E_1V}\,. \quad (2.194)$$

Thus, the differential cross section from equation (2.185) becomes

$$\mathrm{d}\sigma = \frac{1}{4|\mathbf{p}_1|\sqrt{s}}|\mathcal{M}_{f\leftarrow i}|^2\mathrm{dLIPS}_{n'}(p_1+p_2)\,. \quad (2.195)$$

From this, we can get the total cross section by integration over a solid phase space angle

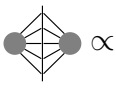
$$\sigma = \int \mathrm{d}\Omega\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega}\,. \quad (2.196)$$

So, to calculate the cross section for a process, it is most important to calculate the interaction part of the scattering matrix squared $|\mathcal{M}_{f\leftarrow i}|^2$. This can be done perturbatively in the coupling constants involved in the process

$$\mathrm{d}\sigma = \mathrm{d}\sigma^{\mathrm{LO}} + \frac{\alpha_s}{4\pi}\mathrm{d}\sigma^{\mathrm{NLO}} + \left(\frac{\alpha_s}{4\pi}\right)^2\mathrm{d}\sigma^{\mathrm{NNLO}} + \mathcal{O}(\alpha_s^3)\,. \quad (2.197)$$

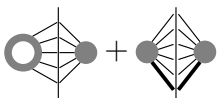
Looking at the Feynman rules from section 2.3 we realize that the lowest order contribution to $|\mathcal{M}_{f\leftarrow i}|^2$ and thus to $\mathrm{d}\sigma$ comes from tree-level amplitudes.

$$d\sigma^{\mathrm{LO}} = \propto \mathcal{M}_{n'}^{(0)\dagger}\mathcal{M}_{n'}^{(0)}\mathrm{dLIPS}_{n'}(p_1+p_2)$$



These are the leading-order contributions. At NLO, there are contributions with one more loop—the virtual corrections—and one more leg—the real corrections.

$$d\sigma^{\mathrm{NLO}} = d\sigma^{\mathrm{V}} + d\sigma^{\mathrm{R}} =$$



,

that is

$$\begin{aligned} \mathrm{d}\sigma^{\mathrm{V}} &\propto \left(\mathcal{M}_{n'}^{(0)\dagger}\mathcal{M}_{n'}^{(1)} + \mathcal{M}_{n'}^{(1)\dagger}\mathcal{M}_{n'}^{(0)}\right) \mathrm{dLIPS}_{n'}(p_1 + p_2)\,, \\ \mathrm{d}\sigma^{\mathrm{R}} &\propto \left|\mathcal{M}_{n+1}^{(0)}\right|^2 \mathrm{dLIPS}_{n'+1}(p_1 + p_2)\,. \end{aligned} \tag{2.198}$$

Finally, at NNLO, there are double-virtual, real-virtual, and double-real corrections to the differential cross section

$$d\sigma^{\mathrm{NNLO}} = d\sigma^{\mathrm{VV}} + d\sigma^{\mathrm{RV}} + d\sigma^{\mathrm{RR}}$$

with the single contributions combined in analogy to the NLO case given in equation (2.198). These contributions are necessary to obtain the total IR-finite result for the differential cross section to a particular order in the coupling constant. Due to the occurring integrals over the internal phase space, the double-virtual contributions are the bottleneck of the calculations. This work concentrates on the calculation of these virtual and double-virtual contributions to processes involving partons and photons.

3. Methodology

To make the theory comparable to the particle physics collision experiment, we need to calculate some fiducial and differential cross sections for various observables, starting from the Lagrangian of the theory we want to study. The best way to do so is by using Feynman diagrams; that is, one can draw all diagrams needed for the process one considers, calculate their values using Feynman rules, and sum over them. These calculations turn out to be complex in case of high multiplicity or multiple loops, so some alternative methods are required to simplify them.

3.1. Helicity Amplitudes

We use helicity amplitudes as a basis for the amplitude of the considered process. To get the scattering matrix element square, one must perform a sum over the squared helicity amplitudes with all possible helicities. As described in section 2.4, virtual amplitudes can exhibit soft and collinear divergences, which need to be regularized. We want to regularize our amplitudes using the 't Hooft–Veltman scheme [55], which defines the internal momenta in D dimensions and the external still in 4 dimensions. Because of this, we embed our external states into a D_s dimensional state space $\mathbf{S}_{[D_s]}$, where D_s is taken as the dimensionality of the spin space for internal particles ($D_s \geq D$). We then find the analytic dependence of the amplitude on D_s and consider the limit $D_s \to D$. External gluon vector states $\epsilon_\pm$ are embedded trivially [56, 57]

$$\epsilon_j^\mu = \begin{cases} \epsilon_\pm^\mu, & j \leq 2, \mu < 4 \\ 0, & j \leq 2, \mu \geq 4 \\ \delta_{(j+1)}^\mu, & j > 2\,. \end{cases} \tag{3.1}$$

For the external fermion spinor states $u_\lambda^{\bar{\alpha}}$ we factorize $\mathbf{S}_{[D_s]} = \mathbf{S}_{[4]} \otimes \mathbf{S}_{[D_s-4]}$ and set

$$u_{\lambda,i}^{\bar{\alpha}\hat{\alpha}} = u_\lambda^{\bar{\alpha}} \delta_i^{\hat{\alpha}}, \quad i = 1, \ldots, 2^{D_s/2-2} \tag{3.2}$$

with $\lambda = \pm\frac{1}{2}$ so that we in total have $2^{D_s/2-1}$ spinor states.

Then the k-th perturbative expansion coefficient $M^{(k)}$ of the amplitude can be written as a vector in the space of the $(D_s - 4)$-dimensional tensors v_i with

coefficients $M_i^{(k)}$, which contain no spinor indices

$$M^{(k)} = \sum_n v_n M_n^{(k)} .$$

For the case of one spinor chain, because of the properties of gamma matrices [58] we can absorb the $\mathbf{S}_{[D_s-4]}$ part of our amplitude into $v \in \mathbf{S}_{[D_s-4]}$ leaving an $\mathbf{S}_{[D_s-4]}$ scalar $M_0^{(k)}$ [57], that is

$$M^{(k)} = v M_0^{(k)}, \quad v_{\hat{\alpha}}^{\hat{\beta}} = \delta_{\hat{\alpha}}^{\hat{\beta}}. \tag{3.3}$$

The dependence of $M_0^{(k)}$ on D_s has now to be established. In analogy to the discussion in reference [59], we find that the second-order of the amplitude's expansion can be written as

$$M^{(2)}(D_s) = \sum\nolimits_{i=0}^{2} \tilde{\mathcal{K}}_i (D_s - 6)^i . \tag{3.4}$$

Just as in equations (3.1) and (3.2) we factorize our D_s metric

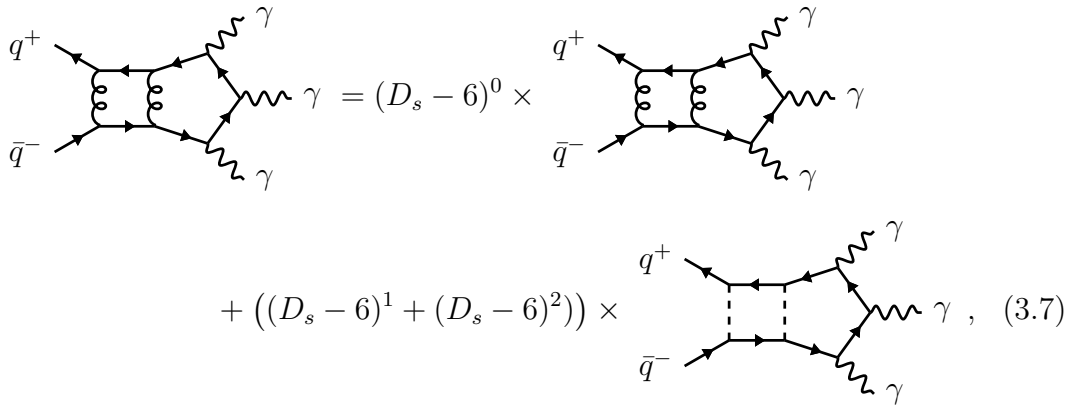
$$g_{[D_s]}^{\mu\nu} = g_{[6]}^{\mu\nu} + g_{[D_s-6]}^{\mu\nu} , \tag{3.5}$$

and γ matrices [57]

$$(\gamma_{[D_s]}^{\mu})_{a\kappa}^{b\lambda} = \begin{cases} (\gamma_{[6]}^{\mu})_a^b \delta_\kappa^\lambda, & \mu < 6 \\ (\gamma_{[6]}^{\star})_a^b (\gamma_{[D_s-6]}^{\mu-6})_\kappa^\lambda, & \mu \geq 6 \end{cases} , \tag{3.6}$$

with $\gamma^\star$ fulfilling $\{\gamma_{[6]}^\star, \gamma_{[6]}^\mu\} = 0$ and $\gamma_{[6]}^\star \gamma_{[6]}^\star = 1$. Because there are no momenta with components in $D_s - 6$ to contract with, the only objects in the $(D_s - 6)$-dimensional chains will be metric tensors $g_{[D_s-6]}^{\mu\nu}$. Diagrammatically they can be accounted for by adding a scalar particle, which represents the $(D_s - 6)$-

dimensional gluon after taking the trace over $\mu_i \geq D_s - 6$, as for example



(3.7)

where the loop momenta on the left-hand side are six-dimensional, and the spin states are D_s-dimensional. The dashed lines represent scalars. The diagrams on the right-hand side are all evaluated using six-dimensional states only. We end up with a quadratic function in D_s. Taking the limit $D_s \to D$ gives us the desired result. In the case of a single pair of quarks, for every order of perturbation k one single coefficient $\mathcal{M}^{(k)}$ has to be determined by

$$\mathcal{M}^{(k)} = M_0^{(k)} = w^0 M^{(k)} \tag{3.8}$$

with $(w^0)^\lambda_\kappa = \delta^\lambda_\kappa$.

The helicity amplitudes need to be normalized in order to be Lorentz invariant. With $\mathcal{C} = (h_1, \ldots, h_n)$ the helicities of the amplitude, and $\mathbf{v}$, $\mathbf{f}^\pm$ the indices of the vector-bosons and fermions in the order of $\mathcal{C}$ we define [31]

$$\Phi_{\mathcal{C}} = \prod_{i=1}^{n_v} \omega_{\mathsf{v}_i}^{\mathrm{sign}(h_{\mathsf{v}_i})} \prod_{i=1}^{n_f} \eta_{\mathsf{f}_i^- \mathsf{f}_i^+} \,, \tag{3.9}$$

in the case that the corresponding tree-level amplitude $\mathcal{M}^{(0)}$ vanishes and

$$\Phi_{\mathcal{C}} = \frac{1}{\mathcal{M}^{(0)}(1_{f_1}^{h_1}, \ldots, n_{f_n}^{h_n})} \tag{3.10}$$

in case of a non-vanishing tree-level amplitude. Here, n_v is the number of vector bosons of the considered process and $2n_f$ the number of participating fermions,

so that $n = n_v + 2n_f$. The corresponding values for $\omega_i^{\pm}$ and η_{ij} are

$$\begin{aligned} \omega_1^+ &= \frac{[12]\langle 32\rangle}{\langle 13\rangle}, \qquad \omega_i^+ = \frac{\langle 13\rangle}{\langle i1\rangle^2 [12]\langle 32\rangle} \quad \text{for} \quad i \geq 2, \\ \omega_i^- &= \frac{1}{\omega_i^+}, \qquad \eta_{ij} = \langle i k_{ij}\rangle [k_{ij} j], \end{aligned} \tag{3.11}$$

with $k_{ij} = \min\{k \mid k \neq i, k \neq j\}$. With this spinor weight we can normalize the amplitude $\mathcal{M}^{(k)}(1_{f_1}^{h_1}, \ldots, n_{f_n}^{h_n})$ by

$$\mathcal{M}^{(k)}(1_{f_1}^{h_1}, \ldots, n_{f_n}^{h_n}) = e_q^3 \Phi_{\mathcal{C}} \, \overline{\mathcal{M}}^{(k)}(1_{f_1}^{h_1}, \ldots, n_{f_n}^{h_n}) \tag{3.12}$$

This definition makes $\overline{\mathcal{M}}^{(k)}(1_{f_1}^{h_1}, \ldots, n_{f_n}^{h_n})$ a Lorentz invariant quantity.

3.2. Color Ordering

The QCD Feynman rules have SU(3) indices in them, which propagate to the amplitude and can be simplified [60]. As can be seen from the Feynman rules, the gluon vertices contribute SU(3) structure constants f^{abc} to the amplitude, whereas quark-gluon vertices contribute SU(3) generators T^a_{ij}. For now, we keep the degree 3 arbitrary; that is, we consider SU(N_c) in order not to lose it within the calculations. The structure constants are defined by

$$[T^a, T^b] = \mathrm{i}\sqrt{2} f^{abc} T^c = \tilde{f}^{abc} T^c, \tag{3.13}$$

where we always sum over repeated indices here. From equation (3.13) we get

$$\tilde{f}^{abc} = \big(\mathrm{Tr}(T^a T^b T^c) - \mathrm{Tr}(T^a T^c T^b)\big). \tag{3.14}$$

With the normalization $\mathrm{Tr}(T^a T^b) = \delta^{ab}$, the generators obey the Fierz identity

$$T^a_{ij} T^a_{kl} = \delta_{il}\delta_{kj} - \frac{1}{N_c}\delta_{ij}\delta_{kl}. \tag{3.15}$$

The Fierz identity is helpful because it replaces all the generators within the propagators of the considered diagram by contractions. For example if we consider the gluon diagram in figure 3.1, the color factor is according to the Feynman rules

$$\begin{aligned} \tilde{f}^{aed}\tilde{f}^{ebc} &= \big(\mathrm{Tr}(T^a T^e T^d) - \mathrm{Tr}(T^a T^d T^e)\big)\big(\mathrm{Tr}(T^e T^b T^c) - \mathrm{Tr}(T^e T^c T^b)\big) \\ &= T^a_{ij} T^e_{jk} T^d_{ki} T^e_{lm} T^b_{mn} T^c_{nl} - T^a_{ij} T^d_{jk} T^e_{ki} T^e_{lm} T^b_{mn} T^c_{nl} \\ &\quad - T^a_{ij} T^e_{jk} T^d_{ki} T^e_{lm} T^c_{mn} T^b_{nl} + T^a_{ij} T^d_{jk} T^e_{ki} T^e_{lm} T^c_{mn} T^b_{nl}. \end{aligned}$$

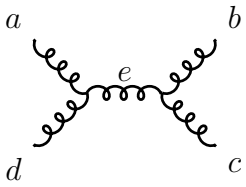


Figure 3.1.: s-channel leading-order gluon-gluon scattering diagram, the letters are the color indices of the gluons.

Now we can use the Fierz identity on the propagator with the index e. The second term in equation (3.15) is suppressed by $\frac{1}{N_c}$, which means contributions from this factor will be suppressed by $\frac{1}{N_c^2}$ in the cross section. We leave them away here, which is the leading-color approximation:

$$\approx T^a_{ij}T^b_{jn}T^c_{nk}T^d_{ki} - T^a_{ij}T^d_{jk}T^b_{kn}T^c_{ni} - T^a_{ij}T^c_{jn}T^b_{nk}T^d_{ki} + T^a_{ij}T^d_{jk}T^c_{kn}T^b_{ni}\,.$$

Performing the same calculation in the t- and the u-channel we can find a closed expression for the whole amplitude, by multiplying every trace with a respective amplitude calculated from Feynman rules without any color factors, which exactly cancels the signs of the traces.

$$\overline{\mathcal{M}}^{(0)}_{\text{l.c.}}(g_1g_2 \to g_3g_4) = \sum_{S_n/Z_n} \text{Tr}(T^{\sigma(1)}T^{\sigma(2)}T^{\sigma(3)}T^{\sigma(4)})\mathcal{A}^{(0)}(\sigma(1),\sigma(2),\sigma(3),\sigma(4))\,. \tag{3.16}$$

These $\mathcal{A}$ are called color-ordered amplitudes. They are simpler to calculate because they have a fixed order of the colored particles. For example, the u-channel belongs to another color-ordered amplitude than the s- and the t-channel. When calculating color-ordered amplitudes, we don't have to deal with the color factors. The color decomposition of an n-gluon amplitude has the same appearance. With the rules (3.14) and (3.15), we can find color decompositions for amplitudes with quarks, uncolored particles like photons and loops. Uncolored particles' Feynman rules do not contribute any $\text{SU}(N_c)$ generators to the amplitude and can be left out when performing the color decomposition. Therefore, they also don't appear ordered in the partial amplitudes $\mathcal{A}$. The decomposition for the amplitude $\overline{\mathcal{M}}^{(0)}_{l.c.}(q\bar{q} \to g\gamma)$ has a single term

$$\mathcal{A}^{(0)}_{\text{l.c.}}(q_1\bar{q}_2 \to g_3\gamma_4) = T^a_{ij}\mathcal{A}^{(0)}(1,2,3,4)\,, \tag{3.17}$$

where now $\mathcal{A}^{(0)}(1,2,3,4)$ consists of all partonic channels.

We can find a consistent color decomposition for pieces with the same number of closed fermion loops in the case of loop amplitudes. At leading-color, every gluon or open fermion loop contributes a factor of N_c and a closed fermion loop

a factor N_f, which is the number of fermions in the loop, that is [54]

$$\overline{\mathcal{M}}^{(k)}(g_1,\ldots,g_n) = (N_c)^k \sum_{S_n/Z_4} \mathrm{Tr}(T^{a_{\sigma(1)}}\cdots T^{a_{\sigma(n)}})\mathcal{A}^{(k)}(\sigma(1),\ldots,\sigma(n))$$

$$\overline{\mathcal{M}}^{(k)}(q,\bar{q},g_1,\ldots,g_n,\gamma,\ldots,\gamma) = (N_c)^k \sum_{S_n}(T^{a_{\sigma(1)}}\cdots T^{a_{\sigma(n)}})_{ij} \times \mathcal{A}^{(k)}(q,\bar{q},\sigma(1),\ldots,\sigma(n),\gamma,\ldots,\gamma)\,,$$

where the generator string $(T^{a_{\sigma(1)}}\cdots T^{a_{\sigma(n)}})_{ij} = \delta_{ij}$ if no external gluons are present and an N_c factor has to be replaced by an N_f factor for every closed fermion loop.

3.3. Recursion Relations

We have seen that one can determine the differential cross section for a particle scattering process up to the desired order in the coupling constant by calculating the relevant amplitudes and integrating over phase space. The amplitudes, in turn, can be determined by deriving the Feynman rules from the Lagrangian of the considered theory, drawing all possible diagrams, and calculating the amplitudes according to the rules. For tree amplitudes, where no integration over the internal momenta takes place, one can build numerical and analytical recursion relations to get the amplitude. If we want to have the analytic expression for an amplitude, we can use a BCFW-type recursion [61, 62]. For numerical evaluations, a convenient approach is to use Berends–Giele recursion [63], which we describe here.

As the amplitude is given by the sum of all Feynman diagrams, we can equivalently write it as

$$\mathcal{A}(\psi_1,\ldots,\psi_n) = \psi_n \cdot J(\psi_1,\ldots,\psi_{n-1}) \tag{3.18}$$

where ψ_i are states of the external particles and the product $\cdot$ depends on the state of the particles, that is it is given by $\eta_{\mu\nu}$ for gluons or photons or by $\epsilon^{\alpha\beta}$ for quarks. $J(\psi_1,\ldots,\psi_{n-1})$ is called the $(n-1)$-current [60] of the amplitude $\mathcal{A}$ and consists of all the Feynman diagrams belonging to the amplitude, without the leg n. This current can be represented diagrammatically as shown in figure 3.2. The legs can be gluons, photons or quarks and the blob represents ways allowed by the Feynman rules to connect the legs. The $(n-1)$-current $J(\psi_1,\ldots,\psi_{n-1})$ can be obtained from bottom to top, by first defining the 1-currents, then building 2-currents from these using Feynman rules, until we connected all particles to

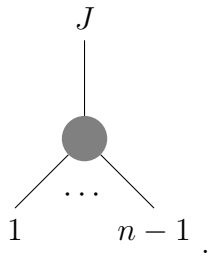


Figure 3.2.: The $(n-1)$-current corresponding to the amplitude $A(\psi_1, \ldots, \psi_n)$

obtain the $(n-1)$-current. The 1-currents are given by the external states

$$J(\psi_i) = \psi_i \,. \tag{3.19}$$

From these, we can build the 2-currents using the Feynman rules, that is we can build the 2-current $J(\psi_i, \psi_{i+1})$ from the Feynman rule V for the states ψ_i and ψ_{i+1}

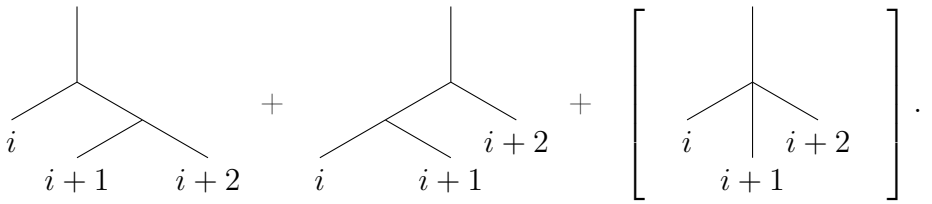
$$J(\psi_i, \psi_{i+1}) = V(\psi_i, \psi_{i+1}) \,. \tag{3.20}$$

While gluons and quarks are color-ordered, photons can attach everywhere to a quark line in color-ordered amplitudes, so that if the second particle in the current is a photon it does not have to be adjacent. While the 1- and $(n-1)$-currents were on-shell, the 2-currents are off-shell and come with a quark or gluon propagator $\tilde{\Delta}(p)$, p being the momentum flowing through the propagator.

As for 3-currents, multiple corresponding Feynman diagrams connect the three particles—the s- and the t-channel in the planar case. Also, we need not to forget to act with propagators on the off-shell 2-currents. That is, the three-current for states ψ_i, ψ_{i+1} and ψ_{i+2} is given by

$$\begin{aligned} J(\psi_i, \psi_{i+1}, \psi_{i+2}) = {} & V\big(J(\psi_i), \tilde{\Delta}(p_{i+1,i+2}) J(\psi_{i+1}, \psi_{i+2})\big) \\ & + V\big(\tilde{\Delta}(p_{i,i+1}) J(\psi_i, \psi_{i+1}), J(\psi_{i+2})\big) \\ & [+ V\big(J(\psi_i), J(\psi_{i+1}), J(\psi_{i+2})\big)] \,. \end{aligned} \tag{3.21}$$

Depending on the model, there can also be a four-vertex, as it is the case for gluons, which is the last line. This graphically can be represented by the diagrams

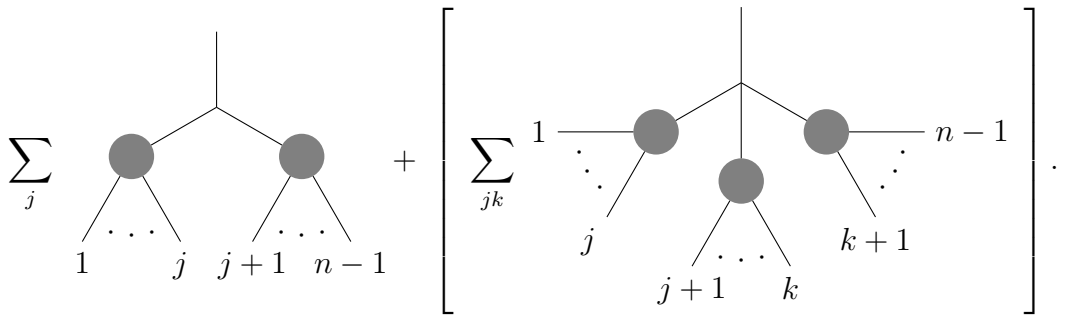


From this and other 3-currents, for example $J(\psi_{i-1}, \psi_i, \psi_{i+1})$, we can recursively construct the $(n-1)$-current $J(\psi_1, \ldots, \psi_{n-1})$. Following the same steps like we

did to come from 2-currents to 3-currents we get

$$\begin{aligned} & J(\psi_1, \ \ldots, \psi_{n-1}) \\ = & \sum_{j=1}^{n-2} V\big([\tilde{\Delta}(p_{1j})]J(\psi_1,\ldots,\psi_j), [\tilde{\Delta}(p_{j+1,n-1})]J(\psi_{j+1},,\ldots,\psi_{n-1})\big) \\ & + \Bigg[\sum_{j=1}^{n-3}\sum_{k=j+1}^{n-2} V\big([\tilde{\Delta}(p_{1j})]J(\psi_1,\ldots,\psi_j), [\tilde{\Delta}(p_{j+1,k})]J(\psi_{j+1},\ldots,\psi_k), \\ & \qquad [\tilde{\Delta}(p_{k+1,n-1})]J(\psi_{k+1},\ldots,\psi_{n-1})\big)\Bigg], \end{aligned} \tag{3.22}$$

where the propagators only appear in front of off-shell currents. $J(\psi_1, \ldots, \psi_{n-1})$ can be graphically represented by



3.4. Integration

So with the above recursion relations, we can calculate tree-level amplitudes. At higher orders of the expansion in the coupling constant, integrals over the internal momenta appear, and it is a central problem in amplitude calculations to solve those integrals. If one writes down all the Feynman diagrams appearing in a one- or two-loop amplitude, there will be many different integrals with some Lorentz tensors in the numerator. These tensor integrals will then be contracted with the external states after integration to give the amplitude. It is, in most relevant cases, impossible or at least time-consuming to calculate those integrals. Thus, it is beneficial to find a basis for the integrals involved in a process. We then can write the amplitude as a linear combination in that basis. The basis should be simple so that the integration can be performed either analytically or numerically in a reasonable time. For this purpose, integration-by-parts identities [21] and differential equations [64] turned out to be most useful. We can use the convenient fact that the integration of total derivatives vanishes in dimensional regularization, which can be seen by transforming into position space [21] or by observing that the surface terms disappear if we choose the dimension of

the internal momenta to be low enough. Because the dimensionally-regularized amplitude is analytic in D, they also have to vanish in an arbitrary dimension [55]. One can use an appropriate IBP vector u^μ as an insertion of an integral and calculate the derivative to find relations between integrals of the form

$$0 = \int [\mathrm{d}\ell] \frac{\partial}{\partial \ell_i^\mu} \frac{u^\mu}{\rho_1 \cdots \rho_n} \,. \tag{3.23}$$

Here is a simple example of how to do that [65]. If we take a regularized massive bubble integral $\mathcal{I}(\alpha_1, \alpha_2)$ with two propagators, the first appearing α_1 times and the second α_2 times and choose $u^\mu = \ell^\mu$, we get

$$0 = \int \frac{\mathrm{d}^D \ell}{\mathrm{i}\pi^{D/2}} \frac{\partial}{\partial \ell^\mu} \frac{\ell^\mu}{(-\ell^2 + m^2)^{\alpha_1} (-(\ell + p)^2 + m^2)^{\alpha_2}} \,, \tag{3.24}$$

to get a relation between those bubble integrals with different numbers α_i of inverse propagators $\rho_1 = -\ell^2 + m^2$ and $\rho_2 = -(\ell + p)^2 + m^2$ we calculate the derivatives of the propagators

$$\ell^\mu \partial_\mu \rho_1 = -2\ell^2 = 2\rho_1 - 2m^2 \,, \tag{3.25}$$

$$\ell^\mu \partial_\mu \rho_2 = -2\ell^2 - 2\ell \cdot p = \rho_2 + \rho_1 + p^2 - 2m^2 \,. \tag{3.26}$$

That is

$$\begin{aligned} 0 &= \int [\mathrm{d}\ell] \partial_\mu \frac{\ell^\mu}{\rho_1^{\alpha_1} \rho_2^{\alpha_2}} \\ &= \int [\mathrm{d}\ell] \left(\frac{\delta_\mu^\mu}{\rho_1^{\alpha_1} \rho_2^{\alpha_2}} - \frac{\alpha_1}{\rho_1^{\alpha_1+1} \rho_2^{\alpha_2}} \ell^\mu \partial_\mu \rho_1 - \frac{\alpha_2}{\rho_1^{\alpha_1} \rho_2^{\alpha_2+1}} \ell^\mu \partial_\mu \rho_2 \right) \\ &= \int [\mathrm{d}\ell] \left(\frac{D}{\rho_1^{\alpha_1} \rho_2^{\alpha_2}} - \frac{2\alpha_1}{\rho_1^{\alpha_1} \rho_2^{\alpha_2}} + \frac{2m^2}{\rho_1^{\alpha_1+1} \rho_2^{\alpha_2}} - \frac{\alpha_2}{\rho_1^{\alpha_1} \rho_2^{\alpha_2}} - \frac{\alpha_2}{\rho_1^{\alpha_1-1} \rho_2^{\alpha_2+1}} + \alpha_2 \frac{2m^2 - p^2}{\rho_1^{\alpha_1} \rho_2^{\alpha_2+1}} \right) , \end{aligned}$$

so we get

$$\begin{aligned} 0 = {} & (D - 2\alpha_1 - \alpha_2) \mathcal{I}(\alpha_1, \alpha_2) - \alpha_2 \mathcal{I}(\alpha_1 - 1, \alpha_2 + 1) \\ & + 2m^2 \mathcal{I}(\alpha_1 + 1, \alpha_2) + (2m^2 - p^2) \alpha_2 \mathcal{I}(\alpha_1, \alpha_2 + 1) \,. \end{aligned} \tag{3.27}$$

Now we can choose the momentum routing to be in the opposite direction inside the loop, that is $\mathcal{I}$ is symmetrical under the change of α_1 and α_2, and get a second equation of this kind with α_1 and α_2 interchanged. These equations we can solve for $\mathcal{I}(\alpha_1, \alpha_2+1)$ and $\mathcal{I}(\alpha_1+1, \alpha_2)$ in terms of $\mathcal{I}(\alpha_1, \alpha_2)$ and $\mathcal{I}(\alpha_1-1, \alpha_2+1)$—that is in terms of integrals with a smaller total power of propagators. Furthermore, by looking at equation (3.27) we see that an integral of the form $\mathcal{I}(\alpha_1, 0)$ can

be expressed in terms of $\mathcal{I}(\alpha_1 - 1, 0)$ and thus in terms of $\mathcal{I}(1, 0)$. That way every integral $\mathcal{I}(\alpha_1, \alpha_2)$ in the family can be expressed by only two integrals, for example $\mathcal{I}(1, 0)$ and $\mathcal{I}(1, 1)$, which are then the "master integrals" for the vector space of $\mathcal{I}(\alpha_1, \alpha_2)$. This method can be also used for multi-loop integrals [66, 67] and can as well be implemented recursively [68], which has been done resulting in various programs for that purpose [69–72].

With the integration-by-parts method above, we can write an integral as a linear combination of master integrals. Now we can find other relations to finally obtain expressions for the master integrals using differential equations [73]. Consider the bubble integral $\mathcal{I}(\alpha_1, \alpha_2)$ from above. The relation (3.27) can be iterated to be able to write the integral in terms of two master integrals, for example $\mathcal{I}(1, 0)$ and $\mathcal{I}(1, 1)$. However, these diverge at least logarithmically, so a smarter choice might be $\mathcal{I}(2, 1)$ and $\mathcal{I}(3, 0)$, which are UV-finite. All massive tadpole integrals can be written down in terms of gamma functions, so to be able to evaluate $\mathcal{I}(\alpha_1, \alpha_2)$ it remains to calculate $\mathcal{I}(2, 1)$. The idea for that is now to differentiate with respect to the invariants of the integral, say m^2. Because m^2 is a linear term in both of the propagators, we get

$$\partial_{m^2}\mathcal{I}(2, 1) = -2\mathcal{I}(3, 1) - \mathcal{I}(2, 2)\,. \tag{3.28}$$

Now we use equation (3.27) with one time setting $\alpha_1 = 1, \alpha_2 = 2$ and one time $\alpha_1 = 2, \alpha_2 = 1$. Adding both results yields

$$0 = 2(D - 5)\mathcal{I}(2, 1) - 2\mathcal{I}(3, 0) + (4m^2 - p^2)\big(\mathcal{I}(2, 2) + 2\mathcal{I}(3, 1)\big)\,, \tag{3.29}$$

which in turn can be solved for the right-hand side of equation (3.28) so that it only depends on the chosen master integrals $\mathcal{I}(2, 1)$ and $\mathcal{I}(3, 0)$. This differential equation is a first-order inhomogeneous differential equation and can be solved using standard techniques, in particular by series expansion in the dimensional regulator ϵ, as

$$\mathcal{I}(2, 1) = \sum\nolimits_k \epsilon^k \left.\frac{\partial_\epsilon^k}{k!}\mathcal{I}(2, 1)\right|_{\epsilon=0}\,. \tag{3.30}$$

If we have more master integrals, using IBP identities, in general the differential equations for the basis integrals $\mathbf{f} = (f_i)_i$ depending on the invariants $\mathbf{x} = (x_i)_i$ can be brought into the form [67]

$$\partial_i \mathbf{f}(\epsilon, \mathbf{x}) = A_i(\epsilon, \mathbf{x})\mathbf{f}(\epsilon, \mathbf{x})\,. \tag{3.31}$$

Equation (3.30) reads in this case

$$\mathbf{f}(\epsilon, \mathbf{x}) = \sum\nolimits_{k=-p}^{\infty} \epsilon^k \mathbf{f}^{(k)}(\mathbf{x})\,, \tag{3.32}$$

where the maximum pole order is $p = 2L$ with L the number of loops. If we also expand $A_i(\epsilon, \mathbf{x})$, that is

$$A_i(\epsilon, \mathbf{x}) = \sum\nolimits_{k=0}^{\infty} \epsilon^k A_i^{(k)}(\mathbf{x}) \,, \tag{3.33}$$

and plug the both expansions into equation (3.31), we get

$$\begin{aligned} \partial_i \mathbf{f}(\epsilon, \mathbf{x}) &= \partial_i \left(\sum\nolimits_k \epsilon^k \mathbf{f}^{(k)}(\mathbf{x}) \right) \\ &= \sum\nolimits_k \epsilon^k \partial_i \mathbf{f}^{(k)}(\mathbf{x}) \\ &= A_i(\epsilon, \mathbf{x}) \mathbf{f}(\epsilon, \mathbf{x}) \\ &= \left(\sum\nolimits_j \epsilon^j A_i^{(j)}(\mathbf{x}) \right) \left(\sum\nolimits_k \epsilon^k \mathbf{f}^{(k)}(\mathbf{x}) \right) \\ &= \sum\nolimits_{j,k} \epsilon^j A_i^{(j)}(\mathbf{x}) \epsilon^k \mathbf{f}^{(k)}(\mathbf{x}) \,, \end{aligned} \tag{3.34}$$

that is for example for two-loop, where $\mathbf{f}^{(-5)}$ is known to vanish, comparing ϵ coefficients we have

$$\begin{aligned} \partial_i \mathbf{f}^{(-4)}(\mathbf{x}) &= A_i^{(0)} \mathbf{f}^{(-4)}(\mathbf{x}) \,, \\ \partial_i \mathbf{f}^{(-3)}(\mathbf{x}) &= A_i^{(0)} \mathbf{f}^{(-3)}(\mathbf{x}) + A_i^{(1)} \mathbf{f}^{-4}(\mathbf{x}) \,, \end{aligned} \tag{3.35}$$

and so on, so for the finite term we have five coupled differential equations to solve. It becomes particularly easy to solve those differential equations if we can bring them in the so-called canonical form

$$\partial_i \mathbf{f}(\epsilon, \mathbf{x}) = \epsilon A_i(\mathbf{x}) \mathbf{f}(\epsilon, \mathbf{x}) \,. \tag{3.36}$$

Then equations (3.35) read

$$\begin{aligned} \partial_i \mathbf{f}^{(-4)}(\mathbf{x}) &= 0 \,, \\ \partial_i \mathbf{f}^{(-3)}(\mathbf{x}) &= A_i(\mathbf{x}) \mathbf{f}^{(-4)}(\mathbf{x}) \,, \\ \partial_i \mathbf{f}^{(-2)}(\mathbf{x}) &= A_i(\mathbf{x}) \mathbf{f}^{(-3)}(\mathbf{x}) \,, \end{aligned} \tag{3.37}$$

and so on, which means the differential equations decouple and can be solved order by order. With this, the task of solving differential equations for an integral basis is shifted towards finding a canonical form for the equations [67].

3.5. Integral Functions

From equations (3.37) we see that the $(k-p)$-th term of the expansion of $\mathbf{f}(\mathbf{x})$ is given by a k-fold iterated integral, which are called Chen iterated integrals [74].

We say k is the transcendental weight of the iterated integral. In general, the coefficients of those iterated integrals are simpler than the coefficients of master integrals. Also, they are often simpler to evaluate than master integrals. Thus, in phenomenology rather coefficients of iterated integrals are used than the ones of master integrals. A Chen iterated integral of transcendental weight k has the form [75]

$$h_k = \int_a^b \mathrm{d}\log R_1 \circ \cdots \circ \mathrm{d}\log R_k \,, \tag{3.38}$$

with $\circ$ given by

$$\int_a^b \mathrm{d}\log R_1 \circ \cdots \circ \mathrm{d}\log R_k = \int_a^b \left(\int_a^t \mathrm{d}\log R_1 \circ \cdots \circ \mathrm{d}\log R_{k-1} \right) \mathrm{d}\log R_k(t) \,. \tag{3.39}$$

Iterated integrals are characterized by the terms R_i. The arguments of the iterated integrals R_i needed to represent an integral basis are called the alphabet $\{W_i\}$ of the basis, with W_i being the letters. To find an iterated integral basis, the alphabet and the R_i have to be found first [76]. In the special case where the alphabet of the iterated integral basis can be rationally parametrized, the basis can be given in terms of so-called Goncharov or multiple polylogarithms, recursively defined by [23, 77, 78]

$$G(a_1, \ldots, a_k, z) = \int_0^z \frac{\mathrm{d}t}{t - a_1} G(a_2, \ldots, a_k, t) \,, \quad \text{and} \quad G(z) = 1 \,, \tag{3.40}$$

if at least one of the a_i does not vanish, or by

$$G(a_1, \ldots, a_k, z) = \frac{1}{k!} \log^k z \,, \tag{3.41}$$

if all $a_i = 0$. Because, in the case of MPLs, the integrations come with only linear factors, MPL bases relevant for calculations at low loop order can be efficiently evaluated, and there are multiple tools to do so [79–82]. Because a helicity amplitude can be written as a linear combination of master integrals, and the master integrals, in turn, can be written in terms of iterated integrals h_i, we can write the whole amplitude in terms of iterated integrals. For a two-loop amplitude this means [30, 83, 84]

$$\mathcal{A}_h^{(2,j)} = \sum_{i \in B} \sum_{k=-4}^{0} \epsilon^k d_{k,i} h_i + \mathcal{O}(\epsilon) \,, \tag{3.42}$$

with h_i the MPLs, h the particle's helicities and j the number of closed fermion loops. Because also the one-loop amplitudes can be written in the same iterated

integral basis, the finite remainders $R_h^{(2,j)}$, which can be obtained from the two-loop amplitudes using one-loop amplitudes and which are further discussed in section 3.8 can also be represented using the same iterated integrals

$$R_h^{(2,j)} = \sum_{i \in B} r_i h_i \,. \tag{3.43}$$

MPLs are useful to parametrize four-point amplitudes. Five-point amplitudes depend on more scales and thus need another parametrization. A basis of iterated integrals called pentagon functions [25, 36] is available to parametrize all massless five-point amplitudes. In their role as master integral parametrizations, iterated integrals like MPLs and pentagon functions are called integral functions.

3.6. Unitarity

Having reduced the occurring integrals as described in section 3.4, a color-ordered L-loop amplitude can be written in terms of a master integral decomposition

$$\mathcal{A}^{(L)} = \sum\nolimits_{\Gamma,i} c_{\Gamma,i} \mathcal{I}_{\Gamma,i} \,, \tag{3.44}$$

where the master integrals

$$\mathcal{I}_{\Gamma,i} = \int d^D \ell_l \frac{m_{\Gamma,i}(\ell_l)}{\prod_{j \in P_\Gamma} \rho_j}, \tag{3.45}$$

with insertions $m_{\Gamma,i}(\ell_l)$ and the coefficients $c_{\Gamma,i}$ depend on the propagator structures Γ, that is, which propagator factors appear in the integrals.

Having talked about integrals $\mathcal{I}_{\Gamma,i}$, there is the question about how to obtain their coefficients $c_{\Gamma,i}$. For this task we used numerical unitarity [26–29] as it is described in [30]. In numerical unitarity we consider the integrand

$$\mathcal{A}^{(2)}(\ell_l) = \sum_{\Gamma \in \Delta} \sum_{i \in M_\Gamma \cup S_\Gamma} c_{\Gamma,i} \frac{m_{\Gamma,i}(\ell_l)}{\prod_{j \in P_\Gamma} \rho_j} \,, \tag{3.46}$$

with Δ the set of all possible propagator structures for the process, P_Γ the set of all inverse propagators ρ_j in Γ, M_Γ the set of master integrands and S_Γ the set of surface terms. To get the master integral coefficients $c_{\Gamma,i}$ with $i \in M_\Gamma$ we sample the expression (3.46) on multiple internal on-shell points, meaning points where $\ell_l \to \ell_l^\Gamma$ so that $\rho_j(\ell_l^\Gamma) = 0$. Here, the left-hand side of (3.46) factorizes into a product of tree amplitudes corresponding to the vertices $i \in T_\Gamma$ appearing in the

propagator structure Γ, so that

$$\sum_{\text{states}} \prod_{i \in T_\Gamma} \mathcal{A}_i^{(0)}(\ell_l^\Gamma) = \sum_{\Gamma' \geq \Gamma, i \in M_{\Gamma'} \cup S_{\Gamma'}} \frac{c_{\Gamma',i} m_{\Gamma',i}(\ell_l^\Gamma)}{\prod_{j \in (P_{\Gamma'} \setminus P_\Gamma)} \rho_j(\ell_l^\Gamma)}. \tag{3.47}$$

Depending on Γ, on the right-hand side we now have to take into account all propagator structures Γ' with $P_{\Gamma'} \supseteq P_\Gamma$, which defines a partial ordering $\Gamma' \geq \Gamma$ iff $P_{\Gamma'} \supseteq P_\Gamma$. After sampling over different values of ℓ_l^Γ we can solve the system of equations (3.47) numerically for the $c_{\Gamma,i}$.

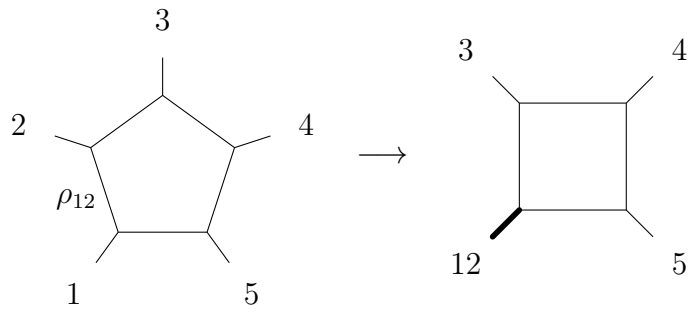
At one-loop this means for example that the pentagon coefficient can be obtained from a single evaluation of the equation

$$\sum_{\text{states}} \left(\mathcal{A}_1^{(0)} \mathcal{A}_2^{(0)} \mathcal{A}_3^{(0)} \mathcal{A}_4^{(0)} \mathcal{A}_5^{(0)} \right) (\ell^{\text{pent}}) = c_{\text{pent}} m_{\text{pent}}(\ell^{\text{pent}}) \,. \tag{3.48}$$

Having that coefficient, in order to get the box coefficient for say the box with legs one and two pinched, one has to evaluate the equation

$$\sum_{\text{states}} \left(\mathcal{A}_{12}^{(0)} \mathcal{A}_3^{(0)} \mathcal{A}_4^{(0)} \mathcal{A}_5^{(0)} \right) (\ell^{\text{box},12}) = \frac{c_{\text{pent}} m_{\text{pent}}(\ell^{\text{box},12})}{\rho_{12}(\ell^{\text{box},12})} + c_{\text{box},12} m_{\text{box},12}(\ell^{\text{box},12}) \,, \tag{3.49}$$

where ρ_{12} is the propagator between leg 1 and leg 2



and to proceed in that manner until one gets the bubble coefficients [27, 59].

The left-hand side of equation (3.47) can be evaluated using off-shell recursion techniques as in section 3.3. As for the right-hand side, to span the integrand space, we need the master integrands from M_Γ and the surface terms from S_Γ [29]. To find them, we think about what numerator insertions we can have in general. Numerator insertions have to be Lorentz invariant. Thus they can be generally parametrized by scalar products of the internal momenta ℓ_l^μ. Valid factors can be internal momenta, external p_i^μ or vectors n_j^μ in the space transverse to the space of the external momenta. Some of these scalar products like ℓ_l^2 will be canceled by the respective propagator or linear combinations thereof. Thus integrals with this insertion belong to a propagator structure $\Gamma' < \Gamma$, that is

$P_{\Gamma'} \subset P_\Gamma$ as described before. That means they are not needed to parametrize the integrand. We use only the scalar products irreducible by propagators (ISPs) for the integrand parametrization. These ISPs are then our variables.

We want the integrals to be UV-finite. Thus, for every propagator structure, we cannot go higher than the number of propagators in the polynomial degree of the ISPs. Their polynomial degree gives us an upper bound on the dimension of the space of numerator insertions, which we then subdivide into master integrals M_Γ and surface terms S_Γ. We determine M_Γ and S_Γ by generating as many S_Γ as possible. The remaining space is spanned by master integrands, resulting in a small basis.

To generate the surface terms, we first notice that a loop integral with an ℓ^μ insertion has to be inside the physical space due to Lorentz symmetry. The contraction of the integral with a vector from outside of the physical space, such as $\mu_{ij} = \ell_i^{(D-4)} \cdot \ell_j^{(D-4)}$ then vanishes. Thus, odd multiples of transverse ISPs vanish and therefore belong to S_Γ. Even multiples of these ISPs do not vanish. However, algorithmically we can bring them into a form, where they vanish by a method called "traceless completion". For the remaining surface terms integration-by-parts as described in section 3.4 is used, that is, we need appropriate IBP vectors u_i^μ for equation (3.23). However, in order to stay in the same propagator structure, we use unitarity-compatible IBP relations [85, 86] which are characterized by the equation

$$\sum\nolimits_i u_i^\mu \frac{\partial}{\partial \ell_i^\mu} \rho_k = f_k \rho_k \,, \tag{3.50}$$

where f_k are polynomial functions of the ℓ_i^μ. This condition ensures that we stay in the same propagator structure. More importantly, we do not generate doubled propagators, which are not compatible with unitarity. Equation (3.50) is a syzygy equation, that is, an equation with polynomial coefficients, which can be solved using algebraic geometry methods [87]. When a set of polynomials f_k^s and solutions $u_{i,s}^\mu$ parametrized by s to (3.50) is found, we can generate surface terms from them. By iteratively applying integration-by-parts r times we see that if $u_{i,s}^\mu$ solves equation (3.23), then also $t_r(\ell) u_{i,s}^\mu$ with a degree r polynomial $t_r(\ell)$. For a set of polynomials $t_r(\ell)$ we thus can generate a set of surface terms by plugging $t_r(\ell) u_{i,s}^\mu$ into equation (3.23)

$$\begin{aligned}
0 &= \int [\mathrm{d}\ell] \frac{\partial}{\partial \ell_i^\mu} \frac{t_r(\ell) u_{i,s}^\mu}{\rho_1 \cdots \rho_n} \\
&= \int [\mathrm{d}\ell] \left(\frac{u_{i,s}^\mu}{\rho_1 \cdots \rho_n} \frac{\partial}{\partial \ell_i^\mu} t_r(\ell) + \frac{t_r(\ell)}{\rho_1 \cdots \rho_n} \frac{\partial}{\partial \ell_i^\mu} u_{i,s}^\mu - \sum\nolimits_k \frac{t_r(\ell) u_{i,s}^\mu}{\rho_1 \cdots \rho_k^2 \cdots \rho_n} \frac{\partial}{\partial \ell_i^\mu} \rho_k \right) \\
&= \int [\mathrm{d}\ell] \left(\frac{u_{i,s}^\mu}{\rho_1 \cdots \rho_n} \frac{\partial}{\partial \ell_i^\mu} t_r(\ell) + \frac{t_r(\ell)}{\rho_1 \cdots \rho_n} \frac{\partial}{\partial \ell_i^\mu} u_{i,s}^\mu - \sum\nolimits_k \frac{t_r(\ell)}{\rho_1 \cdots \rho_n} \sum\nolimits_k f_k^s \right) ,
\end{aligned}$$

where we used equation (3.50) in the last step. That is, with a set of IBP generating vectors $u^{\mu}_{i,s}$ and a set of polynomials $t_r(\ell)$ for a propagator structure Γ we can generate a set of surface terms $m_{\Gamma,(r,s)}$ by [88]

$$m_{\Gamma,(r,s)} = u^{\nu}_{i,s} \frac{\partial t_r(\ell_l)}{\partial \ell^{\nu}_i} + t_r(\ell_l) \left(\frac{\partial u^{\nu}_{i,s}}{\partial \ell^{\nu}_i} - \sum_{k \in P_\Gamma} f^s_k \right). \tag{3.51}$$

3.7. Momentum Twistors

The four-momenta of an n-particle process normally would have $4n$ free parameters. However, due to Lorentz invariance of the system, there are six parameters less. Furthermore, every of the n particles obeys the on-shell condition $p_i^2 = 0$. Finally, there is the momentum conservation fixing another four parameters. In the end we are left with $3n-10$ free parameters [89, 90]. These parameters can be a set of independent Mandelstam invariants $s_{ij} = (p_i + p_j)^2$. Although it is easy to get Mandelstam invariants from four-momenta, the parametrization of four-momenta by Mandelstam invariants results in irrational expressions, which are harder to handle. The external momenta are conveniently parametrized by so-called momentum twistor variables [91]. Setting $s_{12} = 1$ with $\vec{s} = (s_{23}, x, s_{45}, s_{15})$ as parameters we generate the external momenta according to

$$p^{\mu}_i = \frac{1}{2} \tilde{\lambda}^T_i \sigma^{\mu} \lambda_i \,, \tag{3.52}$$

where

$$\begin{pmatrix} \lambda_1 & \cdots & \lambda_5 \\ \mu_1 & \cdots & \mu_5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1+\frac{1}{x} & 1+\frac{1}{x}+\frac{x-s_{23}+s_{45}}{x s_{15}} \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & \frac{s_{23}}{x} & 1 \\ 0 & 0 & 1 & 1 & 1-\frac{s_{45}}{s_{23}} \end{pmatrix} \tag{3.53}$$

four a five-point or with $\vec{s} = (s, t)$

$$\begin{pmatrix} \lambda_1 & \cdots & \lambda_4 \\ \mu_1 & \cdots & \mu_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\frac{1}{s} & -\frac{1}{s}-\frac{1}{t} \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.54}$$

for a four-point process with

$$\tilde{\lambda}_i = \frac{\langle i, i+1 \rangle \mu_{i-1} + \langle i+1, i_1 \rangle \mu_i + \langle i-1, i \rangle \mu_{i+1}}{\langle i, i+1 \rangle \langle i-1, i \rangle} \tag{3.55}$$

where $\langle i,j\rangle = \det(\{\lambda_i,\lambda_j\})$ and $[i,j] = \det(\{\tilde{\lambda}_j,\tilde{\lambda}_i\})$. After the calculation has been performed, the dependence of the result on s_{12} can be obtained from dimensional analysis. The twistor parametrization naturally rationalizes the square root tr_5 of the Gram determinant Δ_5 of the independent momenta participating in the process

$$(\mathrm{tr}_5)^2 = (p_i \cdot p_j)_{i,j=1,\ldots,4} = \Delta_5\,, \tag{3.56}$$

where tr_5 is given by

$$\mathrm{tr}_5 = 4\mathrm{i}\varepsilon^{\mu\nu\rho\sigma} p_1^\mu p_2^\nu p_3^\rho p_4^\sigma\,. \tag{3.57}$$

Here $\varepsilon(p_1,p_2,p_3,p_4) = \varepsilon_{\mu\nu\rho\sigma}p_1^\mu p_2^\nu p_3^\rho p_4^\sigma$, with $\varepsilon_{\mu\nu\rho\sigma}$ the Levi–Civita symbol.

3.8. Pole Structure

Virtual QCD amplitudes have UV, soft and collinear poles. While the UV poles are canceled by renormalization, the IR, so the soft and collinear poles, still appear in the renormalized amplitudes. The IR pole structure of virtual QCD amplitudes is known explicitly up to the second order in the strong coupling constant expansion [92]. One can view the soft divergence of a one-loop diagram by inserting a soft gluon between two legs i and j [93]. As explained in section 3.2 about color ordering, this brings a factor of $T^a_{c_i c_{i'}} T^a_{c_j c'_j}$ to the tree amplitude. Also, after integration, dimensional regularization as in section 2.4 gives a factor of the form $\frac{1}{\epsilon^2}\left(-\frac{\mu^2}{s_{ij}}\right)^\epsilon + \mathcal{O}(1/\epsilon)$. That is, at most double poles appear at one-loop and that means quartic poles will appear at two-loop order. Performing the color algebra results in [54]

$$\mathcal{A}_R^{(1)} = \mathbf{I}^{(1)}(\epsilon)\mathcal{A}_R^{(0)} + \mathcal{O}(\epsilon^0)\,, \tag{3.58}$$

with

$$\mathbf{I}^{(1)}(\epsilon) = -\frac{e^{\gamma_E\epsilon}}{\Gamma(1-\epsilon)}\sum_{i=1}^{n}\gamma_{a_i,a_{i+1}}(-s_{i,i+1}-\mathrm{i}\varepsilon)^{-\epsilon} \tag{3.59}$$

at $\mu = 1$. The color factors $\gamma_{a_i,a_{i+1}}$ depend on the partons a_i and a_{i+1}.

Similarly, the renormalized two-loop amplitude can be written in the form

$$\mathcal{A}_R^{(2)} = \mathbf{I}^{(2)}(\epsilon)\mathcal{A}_R^{(0)} + \mathbf{I}^{(1)}(\epsilon)\mathcal{A}_R^{(1)} + \mathcal{O}(\epsilon^0)\,, \tag{3.60}$$

with

$$\mathbf{I}^{(2)} = -\frac{1}{2}\mathbf{I}^{(1)}(\epsilon)\mathbf{I}^{(1)}(\epsilon) - \frac{\beta_0}{N_c\epsilon}\mathbf{I}^{(1)}(\epsilon) + \frac{e^{-\gamma_E\epsilon}\Gamma(1-2\epsilon)}{\Gamma(1-\epsilon)}\left(\frac{\beta_0}{N_c\epsilon}+K\right)\mathbf{I}^{(1)}(2\epsilon) + \mathbf{H}(\epsilon), \tag{3.61}$$

where

$$K = \frac{67}{9} - \frac{\pi^2}{3} - \frac{10N_f}{9N_c}\,. \tag{3.62}$$

Like the color factors $\gamma_{a_i,a_{i+1}}$, the operator $\mathbf{H}(\epsilon)$ depends on the particles involved in the process. In section 5.2 we will give the color factors relevant for three-photon production.

Looking at equations (3.58) and (3.60) we see that the genuine one- and two-loop information of the respective amplitudes is within the finite remainders $\mathcal{R}^{(l)}$ given by

$$\begin{aligned}
\mathcal{R}^{(0)} &= \mathcal{A}_R^{(0)}, \\
\mathcal{R}^{(1)} &= \mathcal{A}_R^{(1)} - \mathbf{I}^{(1)}\mathcal{A}_R^{(0)} \ + \mathcal{O}(\epsilon), \\
\mathcal{R}^{(2)} &= \mathcal{A}_R^{(2)} - \mathbf{I}^{(1)}\mathcal{A}_R^{(1)} - \mathbf{I}^{(2)}\mathcal{A}_R^{(0)} \ + \mathcal{O}(\epsilon)\,.
\end{aligned} \tag{3.63}$$

3.9. Functional Reconstruction

The analytic form of the master integral coefficients $c_{\Gamma,i}$ as in equation (3.44) or the remainder coefficients r_i of the integral functions h_i as in equation (3.43) can be functionally reconstructed, as described in references [30, 83]. Where four-point results can be reconstructed in one variable, for five-point processes we use the twistor parametrization (3.52), which naturally rationalizes tr_5. As $\mathrm{tr}_5^2 = \Delta_5$, the coefficients are at most linear in tr_5. Because of that, for every coefficient $c_{\Gamma,i}$, we need two evaluations to probe the tr_5 dependence—one with $\operatorname{sign}\mathrm{tr}_5 = \mathrm{str}_5 = 1$ and one with $\mathrm{str}_5 = -1$. The flip $\mathrm{tr}_5 \to -\mathrm{tr}_5$ is accomplished by

$$x \to \bar{x} = \frac{s_{23}(s_{23} - s_{45} - s_{15})(s_{45} - s_{23} + x)}{(s_{45} - 1)s_{15}x - s_{23}(s_{23} - s_{45} - s_{15} - x)}\,. \tag{3.64}$$

The tr_5 dependent coefficients $c_{\Gamma,i}(\vec{s}, \mathrm{tr}_5)$ are then given by

$$c_{\Gamma,i}(\vec{s}, \mathrm{tr}_5) = c^+_{\Gamma,i}(\vec{s}) + \mathrm{tr}_5 c^-_{\Gamma,i}(\vec{s})\,, \tag{3.65}$$

where

$$\begin{aligned}
c^+_{\Gamma,i}(\vec{s}) &= \frac{1}{2}\big[c_{\Gamma,i}\big(\vec{s}(x)\big) + c_{\Gamma,i}\big(\vec{s}(\bar{x})\big)\big]\,, \\
c^-_{\Gamma,i}(\vec{s}) &= \frac{1}{2\mathrm{tr}_5}\big[c_{\Gamma,i}\big(\vec{s}(x)\big) - c_{\Gamma,i}\big(\vec{s}(\bar{x})\big)\big]\,.
\end{aligned} \tag{3.66}$$

First, we want to find the denominators of $c^\pm_{\Gamma,i}(\vec{s})$. To this end, we make an ansatz

$$c^\pm_{\Gamma,i}(\vec{s}) = \frac{n^\pm_{\Gamma,i}(\vec{s})}{\prod_j W^{\mathbf{q}_i^j}(\vec{s})}\,, \tag{3.67}$$

with W_i being the letters of the alphabet of our integral basis. To find the denominators, we consider curves through the parameter space $\{\vec{s}(t)\}$ so that the invariants are linear in t and every letter W_i of the alphabet is a distinct function $W_i(\vec{s}(t))$ of the parameter on that curve. We then reconstruct $c^{\pm}_{\Gamma,i}(\vec{s}(t))$ using Thiele's method. To this end, we have to numerically evaluate the amplitude multiple times, where the evaluations are performed in a finite field. Thiele's method works as follows [94]. Given $y_0, \ldots, y_N$ we write a rational function $f(x)$ as a continued fraction

$$\begin{aligned} f(x) &= a_0 + \cfrac{x - y_0}{a_1 + \cfrac{x-y_1}{\cdots + \cfrac{x-y_{N-1}}{a_N}}} \\ &= a_0 + (x - y_0)\left(a_1 + (x - y_1)\left(\cdots + \frac{x - y_{N-1}}{a_N}\right)^{-1}\right)^{-1} . \end{aligned} \tag{3.68}$$

As seen from equation (3.68), when one evaluates $f(y_0)$, the whole second term vanishes and one obtains a_0. Then with a_0 being known, evaluating at y_1 gives us a_1. Thus, one gets the unknown coefficients a_i by iteratively evaluating

$$a_0 = f(y_0), \quad a_1 = \frac{y_1 - y_0}{f(y_1) - a_0}, \quad \ldots, \quad a_{N+1} = 0. \tag{3.69}$$

Having obtained the analytic form of $c^{\pm}_{\Gamma,i}(\vec{s}(t))$, we match the denominator against the letters $W_i(\vec{s}(t))$. This way, we never have to use multivariate reconstruction of rational functions, although it would be possible with Peraro's method as described in reference [94]. Next, we need to reconstruct the polynomials $n_{\Gamma,i}$ in four variables for a five-point process, for which we use the Newton method [94]. Because we chose the invariants to be linear in t during the application of Thiele's method, we know the degree R of the $n_{\Gamma,i}$, which tells us how long the reconstruction will need. The reconstruction of a polynomial works using Newton's method. Given $y_0, \ldots y_N$ we write a polynomial $p(x)$ using Newton's formula

$$p(x) = a_0 + (x - y_0)\Big(a_1 + (x - y_1)\big(\cdots + (x - y_{N-1})a_N\big)\Big) . \tag{3.70}$$

One sees, if we evaluate at y_0 then we get the coefficient a_0. Knowing a_0, we can get a_1 by evaluating at y_1 and so on, that is

$$a_0 = p(y_0), \quad a_1 = \frac{p(y_1) - a_0}{y_1 - y_0}, \quad \ldots, \quad a_{N+1} = 0 . \tag{3.71}$$

For two variables we write

$$p(x_1, x_2) = a_0(x_1) + (x_2 - y_0)\Big(\cdots + (x_2 - y_{N-1})a_N(x_1)\Big) \tag{3.72}$$

and reconstruct

$$a_0(x_1) = p(x_1, y_0), \quad a_1(x_1) = \frac{p(x_1, y_0) - a_0(x_1)}{y_1 - y_0}, \quad \ldots \tag{3.73}$$

That is, we can reconstruct a polynomial $p(x_1, \ldots, x_k)$ recursively, by looking at it as a polynomial in one variable with coefficients $a(x_1, \ldots, x_{k-1})$. For a reconstruction of a dense degree R polynomial in n variables $\binom{n+R}{n}$ evaluations are needed.

With the above procedure we arrive at an analytic result for the coefficients $c_{\Gamma,i}$ from equation (3.44) or r_i from equation (3.43) in a finite field, which we want to lift to the rational numbers. The lifting is possible with a single finite field if the rational numbers in the lift's image are not too large. To assure that, we decompose the finite field result into partial fractions using a variant [30] of Leinartas' algorithm [95]. This approach is still insufficient for some rational numbers to lift to them using only one finite field evaluation, so we numerically evaluate the amplitudes in a few different finite fields. Knowing the function shape of the coefficients from the functional reconstruction, we can deduce the missing numbers in these finite fields, which eventually allows us to fix the coefficients in all finite fields without functionally reconstructing them a second time.

Since the coefficients of the finite remainder decomposition r_i as in equation (3.43) in general have a smaller polynomial degree and fewer terms as functions of the Mandelstam invariants than the master integral coefficients $c_{\Gamma,i}$, it is often desirable to reconstruct these.

Recently, the authors of reference [31] could assemble the methods mentioned in this chapter to a publicly available code called Caravel, which is described in the next chapter.

4. Caravel Framework

Caravel is a C++-17 framework for the numerical evaluation of multi-loop amplitudes based on the numerical unitarity method. It also provides interfaces for the evaluation of master integrals as well as functional reconstruction tools. It has been used to determine the analytic coefficients of the integral-function decomposition of the remainders of five-parton amplitudes in the Euclidean region in the computations [30, 83] and coefficients for the four-graviton amplitudes [88]. A vital part of this work focuses on constructing this library which we have published in reference [31]. Furthermore, we implemented the required components for evaluating amplitudes involving photons numerically and for the reconstruction of the coefficients of these amplitudes and the corresponding finite remainders.

In Caravel, an integrand of an amplitude is parametrized based on reference [29], as explained in section 3.6. Through this parametrization, its decomposition coefficients directly correspond to the coefficients of a basis of master integrals. If one can evaluate the chosen basis integrals, with the coefficients delivered by Caravel one can get the full amplitude on a given phase space point.

Through numerical unitarity, Caravel determines the coefficients of the master-integrand decomposition by solving a linear system of equations combining needed on-shell limits. Depending on the task one wants Caravel to perform, it relies on external software. Also, some parts of the calculation are process-dependent and are not automatically generated, such as the surface terms or the embraced cut diagrams and their hierarchy. The library needs them as external input.

In this chapter, we present the internal structure of Caravel, technical details for its installation, dependencies, validation of the code, and a series of examples to showcase the usage of the library. We emphasize aspects, which are relevant for the computation of amplitudes involving photons.

4.1. Modules

As numerical unitarity is model-independent, we developed Caravel to have the functionality to allow the usage of multiple theory models. Also, for different physical applications, there are often multiple approaches, where each of them has advantages for one application and disadvantages for another. Caravel was designed to be highly modular. With this modularity, it is easier to add new physical models on the one hand and implement different features in multiple ways

on the other hand and then use the best one depending on the task. Moreover, one can decide what modules to include before compilation, such that the built version of CARAVEL does not become unnecessarily large. In figure 4.1 we show the main modules of CARAVEL and on what they depend. A red arrow means that a component produces certain results, and a blue arrow that a component expects some given input. The modules with red boxes around them need external input produced by other means. **GraphLibrary** with the red background is a module designed to process and classify graphs. It is needed to define the propagator structures Γ highlighted in red in figure 4.1. The black arrows mean dependencies. **OnshellStrategies** is an example of a module implemented in multiple ways with a possibility to choose one, depending on whether one wants to calculate in floating-point or finite field numbers.

In the following, we describe the functionality of the modules and how they are related to each other.

Core: This is the module offering basic objects that most other modules need. Every other module depends on the **Core** module. Here, the types one can select for the states and momenta in CARAVEL are defined, such as floating-point, finite field, or arbitrary-precision rational. Additionally, interfaces to internal or external linear algebra tools, which work in floating-point precision or finite fields, are implemented here. This module also includes classes to represent kinematical objects in D-dimensional Minkowski space like spinors and vectors, which are, for example, needed for the calculation of tree amplitudes and 'cuts'—that is, products of trees—like on the left-hand side of equation (3.47).

GraphLibrary: This is also an essential module, which allows the classification of graphs. For example, the cut diagrams Γ are internally represented as graphs, and a partial ordering is defined on them dependent on how many propagators are present. The tree diagrams in the vertices of the cut diagrams, which in the on-shell limits are calculated via recursion relations, are also represented by graphs dressed with the particle properties. These objects are all inherited from a basic graph class.

FunctionalReconstruction: This module is used in the outermost shell of a calculation for the reconstruction of the analytic form of the coefficients $c_{\Gamma,i}$ from equation (3.44) or r_i from equation (3.43). Here, the univariate reconstruction of rational functions as well as the multivariate reconstruction of polynomials is implemented, as needed to get the analytic form of the coefficients $c_{\Gamma,i}$ or r_i. Tools for parallelization through the Message Passing Interface (MPI) and multithreading are included.

OnShellStrategies: Here, tools are implemented to calculate on-shell loop momenta ℓ_l^Γ for one- and two-loop propagator structures Γ. These are needed

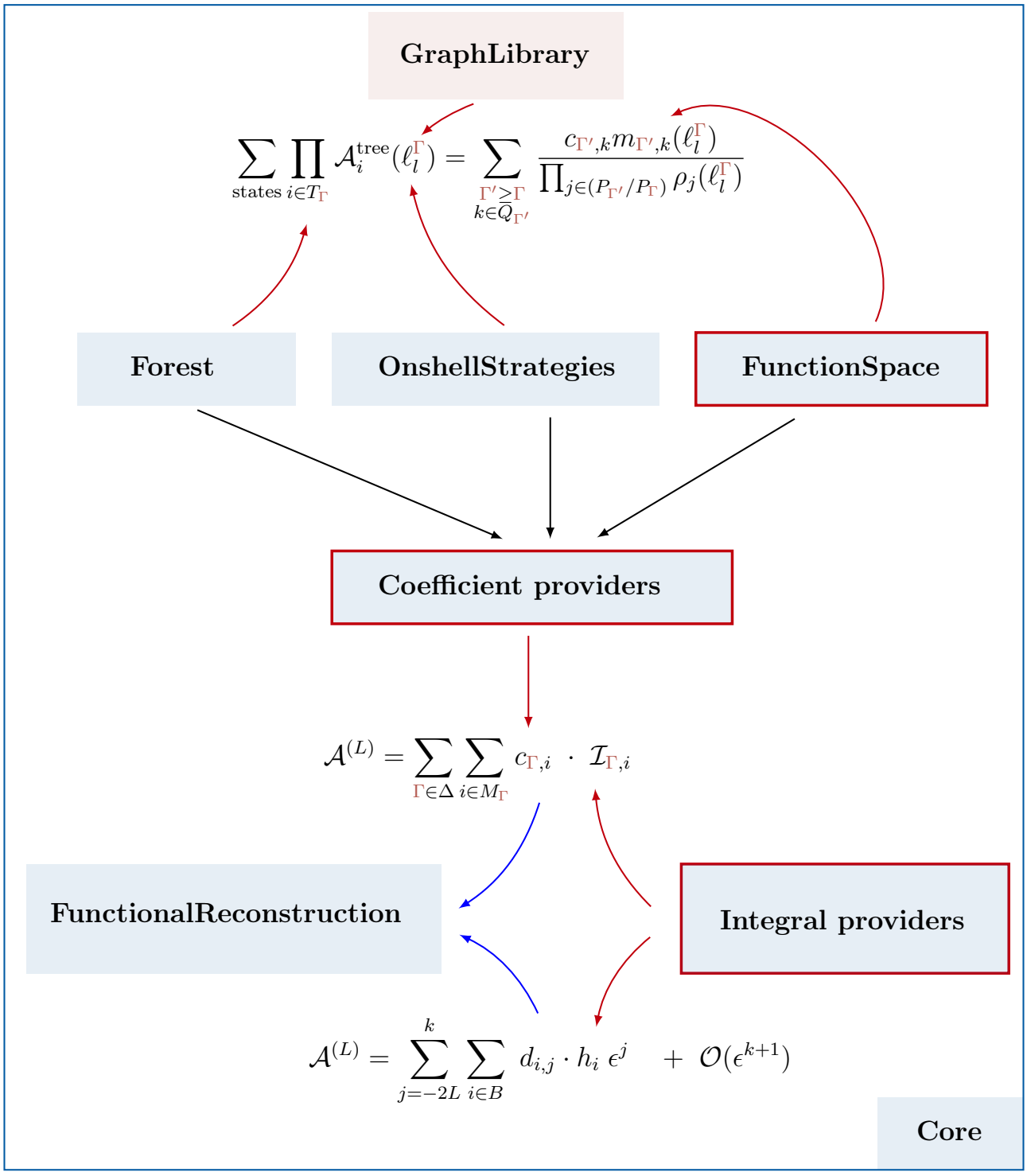


Figure 4.1.: Here, the internal structure of CARAVEL is shown. Dependencies of the modules are indicated by black arrows, input by blue arrows, and output by red arrows. The module **GraphLibrary** classifies the graphs for the propagator structures Γ written in red. Every module depends on the **Core** module.

on both sides of the cut equation (3.47). On the one hand, to evaluate the tree vertices on the left-hand side of equation (3.47) by the **Forest** module, where the internal legs of each amplitude are determined using **OnShell-Strategies**. On the other hand, to evaluate the integrands $m_{\Gamma,i}$, which are provided by the **FunctionSpace** module. The module has a strategy design pattern allowing to add strategies that might suit future purposes, such as massive propagators. So far, there is a finite field strategy called `surd` parametrization, which is described in reference [96] and the `osm` strategy for floating-point kinematics which is described in appendix B.

Forest: In the on-shell limits as in equation (3.47), the integrand of the amplitude can be represented by a product of trees $\mathcal{A}_i^{(0)}$. In the **Forest** module, these tree-level amplitudes are implemented in terms of a Berends–Giele [97] recursion, as described in section 3.3. Because some legs of the vertices represented as trees are loop legs, the recursion is written in D_s dimensions. According to the 't Hooft–Veltman scheme [55] the external particles are kept four-dimensional. The module is divided into the generation of the needed recursions by the `Forest` class, which does not use phase space information, and an evaluation of the trees by the `Builder` class template. Depending on the type of the provided external momenta, it is initialized for finite field types, arbitrary precision using `GMP`, or multiple fixed precision types using the `QD` library. The `Forest` class decides how the trees $\mathcal{A}_i^{(0)}$ are constructed depending on a set of Feynman rules defined in a `Model` class.

FunctionSpace: Here the master integrands as well as surface terms $m_{\Gamma,i}(\ell_l)$ as in equation (3.46) are implemented alongside with the relevant numerator insertions, such as Mandelstam invariants, inverse propagators, or μ insertions, see section 3.6. The code for these insertions is generated alongside with the **Integral providers** using the external tools described later in section 4.5.

Integral providers: Multiple sets of integrals are included in CARAVEL. At first, there are the integrals from reference [98] in the class `IntegralsBH` which were being used for NLO $Wb\bar{b}$ + jet predictions and thus have massive propagators and are also available in the physical region. Then there are two-loop four-point integrals in terms of MPLs, which can be evaluated using `GiNaC` [79]. Finally, there are the integrals from reference [30, 83] for two-loop five-point processes in the Euclidean region.

Coefficient providers: This module uses the **Forest** module to calculate the left-hand side of the linear system of equations (3.47) and solves it for the coefficients $c_{\Gamma,i}$. In the one-loop case, there is the generic `AmpEng` class, which can do the task without additional input. For the case of two-loop

amplitudes, there is the `AmplitudeCoefficients` class, which depends on input from the external **process-library** module, which holds information on the propagator structure hierarchy Δ and the color structure of the amplitude.

PhaseSpace: Here the momentum twistor parametrizations as in section 3.7 for four-, five-, or six-point kinematics are implemented. A hard-coded example phase space point from reference [99] is implemented for the five-point case. Functions to get the Mandelstam invariants and tr_5 from a momentum configuration are implemented here for four-, five-, and six-point kinematics.

PoleStructure: Here, the known pole structure [92] of the one- and two-loop amplitudes is implemented, as described in section 3.8. A sound consistency check for amplitudes for new processes is whether they exhibit the known pole structure, which can be tested using this module. The checks are performed on the level of integral functions, so the module relies on the **Integral providers** module. The module can subtract the pole structure from the respective amplitudes to produce finite remainders as in equation (3.63) as input for the **FunctionalReconstruction** module.

4.2. Installation and Configuration

Before going public, some results, as the analytic five-gluon [83], five-parton [30] and four-graviton [88] amplitudes, were obtained using Caravel. The public version available now [31] showcases these calculations. To obtain the Caravel repository, to build it in a `build` subdirectory of the repository directory, and to install it at the path `<install-dir>`, one has to run the following commands.

```
1 > git clone https://gitlab.com/caravel-public/caravel.git
2 > cd caravel
3 > mkdir build
4 > cd build
5 > meson .. -Dprefix=<install-dir>
6 > ninja
7 > ninja install
```

Listing 4.1: Commands to clone and install Caravel.

Further information on the installation of Caravel can be found in the file `INSTALL.md` in the root directory of the repository, see also section 4.4.

If one installs Caravel as suggested in listing 4.1, only the basic modules like **Core** and **Forest** are going to be available. The functionality to perform calculations in finite fields or floating-point precision higher than double precision

and the integral libraries are optional. Thus, with the default setting only tree amplitudes can be calculated. One can then add and remove modules with the following command.

```
> meson configure -D <opt1>=<val1> -D <opt2>=<val2> ...
```

The list of configuration options can be shown running

```
> meson configure
```

without specifying options. The output of `meson configure` shows the available compiler settings and options specific to CARAVEL. Using the compiler settings one can for example increase the optimization by using `-Doptimization=3`.

Next we describe the available options.

caravel-debug: With this option one can make CARAVEL print additional information when it runs. Users can create a file `debug.dat` in the execution directory where they are to specify the file and function from which they want to see more output. For example, adding the line

```
IntegrandHierarchy.cpp finish_construction
```

to `debug.dat` will make CARAVEL print out the cuts used for the two-loop amplitude one computes together with information about their relations in the hierarchy Δ. Lines in `debug.dat` can be commented out using `#` in front.

double-inverter: To solve the linear system of equations (3.47) in floating-point precision, CARAVEL relies on the external linear algebra library `Eigen` [100]. Alternatively, for double precision calculations, one can use LAPACK [101]. In the case of LAPACK, users can set the path to the library with the **lapack-path** option in case it cannot be found by `meson`.

finite-fields: Users must activate this option if they want to perform calculations in finite fields. It will initialize the classes depending on kinematics with the finite field type and compile functions implemented differently for finite fields, such as the solutions of linear systems of equations. For this option to work the `GMP` [102] library has to be available and seen by `meson`.

field-ext-fermions: This option has to be set if one wants to calculate amplitudes involving fermions using finite field arithmetic because the option applies fermion-specific changes to the code. When fermions are involved, during numerical evaluations, the **Forest** currents need to be algebraically extended to be able to deal with square roots stemming from the generation of on-shell momenta.

gravity-model: With this option, one can select which gravity models to compile. To reproduce the results from reference [88] one has to set this to `Cubic`, which activates the compilation of the cubic Einstein–Hilbert gravity model [103].

precision-QD: By default, floating-point calculations are performed in double precision. With this option, CARAVEL can increase the precision using the external `QD` [104] library to double-double or quadruple-double precision. For even higher precision, the option **precision-arbitrary** can be used. In this case, instead of the `QD` library, the `GMP` [102] and `MPFR` [105] libraries are required.

integrals: When one wants to not only calculate master integral coefficients, but coefficients $d_{k,i}$ of integral functions as explained in section 3.5, the **IntegralLibrary** module has to be compiled, which is done with this option. One can further select whether a representation with MPLs requiring `GiNaC` [79] for four-point amplitudes or a representation with pentagon functions requiring the `pentagon-library` [25] for five-point amplitudes shall be used, with the respective values `goncharovs` and `pentagons`. For the three-photon production in this work, CARAVEL rather produces output compatible with the `PentagonFunctions++` [36] library. The corresponding functionality is selectable by setting **integrals** to `pentagons-new`.

doxygen: With this setting an HTML documentation of the CARAVEL routines will be compiled. It can then be found under

```
<install-dir>/share/doc/Caravel/html/index.html
```

The documentation requires the external `Doxygen` [106] package to be installed on the system.

ds-strategy: Here, one can set how the D_s-dependence of the amplitudes shall be obtained. It is either by particle content as described in section 3.1 or by determining the coefficients of the amplitude as a D_s polynomial through multiple evaluations in different D_s values. While the first approach is faster, it is currently not implemented for gravity calculations.

instantiate-rational: With this option, various functions can be called directly using rational numbers rather than finite fields in exact calculations. This option is only of use when obtaining numerical results.

timing: With this option at the end of execution CARAVEL will show how much time was needed for the steps of the calculation. That is, it will show in particular how long the off-shell recursion, the inversion of the linear system of equations (3.47), and the evaluation of the surface terms have taken.

CARAVEL can be used for different purposes which need different modules, like floating-point calculations versus finite field calculations, numerical evaluations versus functional reconstruction, or QCD amplitudes versus gravity amplitudes. Thus, often only a fraction of the listed options need to be activated for a specific task.

4.3. Tests

CARAVEL comes equipped with several test suites. That way, developers can rigorously check whether they broke some functionality while introducing another. To run the tests on the CARAVEL installation, after running `ninja install` the command `meson test` or `ninja test` has to be executed. The tests performed are separated into four test suites. The first is the suite `unit_test`, which contains fast tests that make simple consistency checks. There is a separate `integrals` test suite, which compares the implementation of the integrals against reference values. The third is the suite `big_test`, with tests that check complete modules for consistency. The fourth is the suite `full_amplitudes`, with tests investigating whether some four- and five-gluon amplitudes reproduce the results from reference [54]. The suites can be selected individually by typing

```
> meson test --suite Caravel:<suite name>
```

Depending on the chosen options at installation as described in section 4.2, CARAVEL will be installed with different modules. Thus, CARAVEL performs more or fever tests depending on the options. All available tests can be run by executing

```
> ninja test
```

Next, we describe the tests included in each suite.

unit_test: This suite checks the individual building blocks of CARAVEL for consistency. If the user did not select any options, no integral representations, gravity models, or extensions specific to fermions in D_s dimensions would be installed. It would still be possible to calculate tree amplitudes with floating-point precision. In this case, CARAVEL can still perform various unit tests because they are made to check a specific functionality with few dependencies on other modules. In particular, the **GraphLibrary** will be checked for consistency, the input parser dealing with the command-line options passed to CARAVEL, as well as the vector boson implementation, which is valid for every value of D_s. Although no integral representations are available with that choice of options, the integral classes and methods are installed, and multiple checks for their consistency as well as their decomposition into integral functions are performed inside `MasterIntegral_test`. Furthermore, suppose a set of integrals has been made available by enabling

the according **integrals** option. In that case, it becomes possible to check the implementation of the pole structure of amplitudes and its subtraction to determine finite remainders according to equations (3.63) using the `PoleStructure` and `Remainder` unit tests. The pole structure and remainder checks are done on the level of integral functions, for which at least one set of integrals has to be available. To this end, some pole operators are checked for consistency, and the poles of partonic amplitudes are calculated without amplitude evaluation. In particular, the option **field-ext-fermions** and time-consuming coefficient determination are not needed to check the pole subtraction.

Gravity models like the Einstein–Hilbert model are heavy and thus not activated by default. As soon as at least one gravity model is installed, the tensor algebra needed to deal with graviton states, including corresponding completeness relations, is going to be checked for consistency inside the `tensorbasis` test.

With the installation option **finite-fields** CARAVEL gains multiple applications. With this setting, the exact fermionic Clifford algebra as described in appendix A becomes available and is checked for consistency. Moreover, it becomes possible to functionally reconstruct amplitude and remainder coefficients as described in section 3.9, so with the option **finite-fields** set to `true`, simple reconstructions using Peraro's as well as Thiele's algorithm [94] are performed. As described in section 3.9, the reconstruction of amplitude coefficients is simplified by the fact that their denominators are polynomials in the alphabet of the integral basis. CARAVEL can determine the denominator by reconstructing it in one variable and then fitting the alphabet's letters to it. This functionality is also checked by fitting the denominators of a known test function inside of the `guess_denominators` test, which is enabled when running the unit test suite with the option **finite-fields** selected.

integrals: Running this suite checks the implementation of the integrals selected during installation. Integrals available to select are the massive one-loop integrals defined in all physical regions from the `BlackHat` library, Goncharov polylogarithms evaluated with the help of `GiNaC` for $2 \to 2$ processes, or pentagon functions for $2 \to 3$ processes. If Goncharov polylogarithms are available, that is, the option **integrals** set at least to `goncharovs`, CARAVEL will perform consistency checks on them. Also, it will check some simpler constants and functions, like the logarithm, against reference values. As the evaluation of Goncharov polylogarithms depends on `GiNaC`, the CARAVEL–`GiNaC` interface will also be checked when running this test suite. Similarly, if **integrals** is set to `all` or to `pentagons`, the available massless

five-point integrals in the Euclidean region are checked against reference values. As described in section 4.4, these tests rely on the `pentagon-library` and thus check Caravel's `pentagon-library` interface.

big_test: This suite consists of fewer tests, where each checks the interplay between multiple modules. It should be run when the correct performance of the individual classes has been verified with appropriate unit tests. As mentioned, it is possible to calculate tree amplitudes with Caravel with no additional options selected during installation. With these settings, the `big_test` suite checks some partonic tree amplitudes against reference values. It also checks gluonic MHV amplitudes against simple analytic expressions from reference [107].

While there have been unit tests for simple graviton amplitudes, selecting a gravity model makes respective tree amplitudes with up to six gravitons available as a big test to check against reference values.

For the determination of loop amplitude or remainder coefficients, some linear algebra functionality has to be installed to invert the linear system of equations (3.47). If this is the case, in particular if LAPACK or `Eigen` is available to Caravel, one-loop partonic amplitudes can be checked against results from `BlackHat`. Also, the automatic one-loop or two-loop four-point **Coefficient provider** `AmpEng` is checked. In particular, the coefficient determination is tested by extracting the sunrise coefficient of a four-gluon process, thus checking the proper implementation of the whole propagator hierarchy. A similar check for non-planar propagator structures is performed with a four-graviton process if the cubic Einstein–Hilbert model is available.

If the option **integrals** was set at least to `pentagons`, the interface to the `pentagon-library` is tested, comparing the evaluation of selected pentagon functions against reference values.

Again, if the option **finite-fields** is set to `true`, Caravel can perform more tests. Additional checks on gravity tree amplitudes become possible. Also, partonic one-loop amplitudes are checked against the results of reference [54] using either the `IntegralsBH` implementation of the integrals with `1L_with_IntegralsBH`, or using `GiNaC` if the option **integrals** was set at least to `goncharovs` with `1L_QCD_Amplitudes`. For sufficient precision of intermediate expressions, this check needs the option **precision-QD** to be set to `HP`. This test involves all of the off-shell recursion mechanisms for partonic amplitudes to calculate the left-hand side of (3.47), thus providing a check for this part of Caravel.

If the option **integrals** was set to `pentagons` and finite fields are available, a full remainder calculation for the single-minus two-loop five-gluon

amplitude is performed. CARAVEL then checks that the poles can be reproduced by the strategy described in section 3.8. This test is called `mpppp_remainder_check` and is the most extended check performed in this suite at the same time testing the whole numerical part of CARAVEL for QCD amplitudes.

If **integrals** are set to `goncharovs` and finite fields are available, one-loop checks against reference values for the three available gravity models are performed. Also, now a reconstruction of a gluonic four-point MHV amplitude is performed, testing the functional reconstruction capabilities of CARAVEL in one variable.

full_amplitudes: In this suite, selected complete two-loop amplitudes are calculated numerically and checked against values from reference [54]. The individual tests first reproduce the values from reference [54]. After that, every master integral coefficient up to finite order in the regularization parameter is checked against individually. The comparison of the coefficients allows finding errors specific to individual propagator structures more efficiently. Every of the checked amplitudes needs the option **finite-fields** to be activated. If the option **integrals** is set to `goncharovs`, four-point amplitudes are going to be checked and if it is set to `pentagons`, five-point amplitudes are going to be checked. If the option **field-ext-fermions** is activated, in addition to gluonic amplitudes, amplitudes involving quarks are checked. In the cases where the coefficients can be compared exactly, the amplitudes are compared to benchmark values with a tolerance ranging from three digits for an $\overline{\text{MHV}}$ gluonic five-point amplitude to nine digits for partonic four-point amplitudes. The amplitude tests are all executed by the same example code, which is presented in section 4.6 and was extended with the checks of amplitudes and individual master integral coefficients.

The unit tests and the integral tests are located in the `unit_tests` directory and the big tests as well as the full amplitude tests in the `tests` directory. In addition to be part of a suite, every test can be run individually when executing

```
> meson test <test name>
```

For example, to check CARAVEL against available one-loop values one has to run

```
> meson test 1L_QCD_Amplitudes
```

For this test to be available the option **finite-fields** has to be activated, the option **precision-QD** to be set at least to `HP` and the option **integrals** to be set to `all` or `goncharovs` during installation.

4.4. Dependencies

Notes on CARAVEL's dependencies are available with the files `README.md` and `INSTALL.md` from the CARAVEL repository. Here we give further hints on their installation and where the dependencies are needed.

meson and ninja: As we described in section 4.1, CARAVEL consists of multiple modules. Each of these modules, in turn, consists of multiple individual files. This distinction makes the compilation of CARAVEL nontrivial so that we rely on building automation for that task. To build CARAVEL, we use the `meson` build generator [108]. In comparison to the widely used `autotools` [109], `meson`'s build files are easier to read and understand. Furthermore, we compared the compilation time of CARAVEL with `meson` with respect to `autotools` at an early stage of development. In this comparison, CARAVEL was build considerably faster using `meson` than `autotools`. `meson` is available publicly [108] and depends itself on Python 3 [110]. If the Python package installer `pip` [111] is used, `meson` can be integrated in the available Python system by running

```
> sudo pip install meson
```

for the whole operating system or by running

```
> pip install --user meson
```

for the user who is executing the command.

In contrast to `autotools`, `meson` does not create build strings to be executed by a shell itself but relies on other low-level tools based on Python. As a back end to meson, we use the build system `ninja` [112] for this task. `ninja` is available publicly [112]. Like meson, it can conveniently be installed using `pip` by running

```
> sudo pip install ninja
```

for the whole operating system or by running

```
> pip install --user ninja
```

for the user who is executing the command.

With `meson` and `ninja` installed on the system, CARAVEL can be installed with the minimal set of functionality. For some tasks, however, additional software libraries are needed. In particular, CARAVEL needs additional libraries for exact and high-precision floating-point calculations, for the inversion of the linear system of equations (3.47), and the evaluation of master integrals.

QD: Without any extension, CARAVEL can produce results in double floating-point precision. These can exhibit limited accuracy, so that one might be interested in performing calculations with more precise floating-point arithmetic. For this task, CARAVEL uses the `QD` [104] library, which efficiently implements the double-double and quadruple-double types. It is available publicly [104]. To do so and to install `QD`, the commands in listing 4.2 can be used.

```
1 > wget https://www.davidhbailey.com/dhbsoftware/\
2 > qd-2.3.22.tar.gz
3 > tar -xzf qd-2.3.22.tar.gz
4 > cd qd-2.3.22
5 > autoreconf -i
6 > ./configure --prefix=<install-dir> --enable-shared=yes
7 > make
8 > make install
```

Listing 4.2: Commands to download, build and install the `QD` library

The appropriate initializations and methods inside CARAVEL can be activated using the **precision-QD** option as described in section 4.2.

GMP and MPFR: If more than quadruple-double precision is needed, CARAVEL can produce arbitrary precision results using the `MPFR` library [105]. `MPFR` is a C library and CARAVEL uses a C++ wrapper [113] to access it. The wrapper comes with CARAVEL and does not need to be installed additionally. `MPFR` in turn depends on the `GMP` library [102] for the treatment of large integer numbers. `GMP` is available publicly [102]. `MPFR` should come with version higher than 4.0.0 and `GMP` with version higher than 6.2.0 in order for CARAVEL to work properly. As described in section 4.2, arbitrary precision calculations can be enabled using the installation option **precision-arbitrary**. To install `GMP` 6.2.1, the commands in listing 4.3 can be used.

```
1 > wget https://gmplib.org/download/gmp/gmp-6.2.1.tar.xz
2 > tar -xf gmp-6.2.1.tar.xz
3 > cd gmp-6.2.1/
4 > mkdir build
5 > cd build
6 > ../configure --prefix=<install-dir> --enable-cxx
7 > make
8 > make install
```

Listing 4.3: Commands to download, build and install the `GMP` library

When installing `GMP`, the newest version number can be found in reference [102] and then used instead of 6.2.1 in the commands of listing 4.3. When `GMP` is available, to install `MPFR`, 4.1.0 the commands of listing 4.4 can be used.

```
1 > wget https://www.mpfr.org/mpfr-current/mpfr-4.1.0.tar.gz
2 > tar -xzf mpfr-4.1.0.tar.gz
3 > cd mpfr-4.1.0/
4 > mkdir build
5 > cd build
6 > ../configure --prefix=<install-dir> \
7 > --with-gmp=<gmp-install-dir>
8 > make
9 > make install
```

Listing 4.4: Commands to download, build and install the `MPFR` library

Same as for `GMP`, the newest version number of `MPFR` can be found in reference [105] and then used instead of 4.1.0 in the commands of listing 4.4. `GMP` is also used in the context of rational numbers arithmetic. Here, the numerators and denominators of some master integral coefficients exceed the 64 bit available to machine integer variables and are represented by particular arbitrary precision types. For this task, CARAVEL uses arbitrary size numbers represented by the classes `BigRat` and `BigInt`, which in turn rely on the `GMP` library.

Eigen: Starting from one-loop amplitudes, to solve the linear system of equations (3.47), we need to invert the matrix $m_{\Gamma',i}(\ell_i^\Gamma)$ containing the propagator structure specific integrands, as has been described in section 3.6. The inversion is done in various ways, dependent on the field used in the calculation. For floating-point precision the inversion is performed through a PLU decomposition of the matrix using `Eigen` [100] by default. `Eigen` is a templated linear algebra library and, as such, can invert matrices with entries in all floating-point types available to CARAVEL. To install the current stable version of `Eigen`, the commands of listing 4.5 can be used.

```
1 > git clone https://gitlab.com/libeigen/eigen.git
2 > cd eigen
3 > mkdir build
4 > cd build
5 > cmake .. -DCMAKE_INSTALL_PREFIX=<install-dir> \
6 > -DPKGCONFIG_INSTALL_DIR=lib/pkgconfig
7 > make install
```

Listing 4.5: Commands to download, build and install the `Eigen` library

LAPACK: Specifically for the case of double precision, additional routines for the determination of the coefficients $c_{\Gamma',i}$ in equation (3.47) are implemented. As described in section 4.2, as an alternative to `Eigen`, the LAPACK library [101] can be selected as the linear algebra package for this task by setting the installation option **double-inverter** to `lapack`. To install LAPACK 3.9.0, the commands of listing 4.6 can be used.

```
1 > wget https://github.com/Reference-LAPACK/lapack/archive/\
2 > v3.9.0.tar.gz
3 > tar -xzf v3.9.0.tar.gz
4 > cd lapack-3.9.0
5 > mkdir build
6 > cd build
7 > meson .. -Dprefix=<install-dir>
8 > ninja install
```

Listing 4.6: Commands to download, build and install the LAPACK library

To take advantage of the most recent bug fixes, the newest version number of LAPACK as found in reference [101] can be used instead of 3.9.0 when executing the commands of listing 4.6. Note that it might be necessary to pass the `<install-dir>` of LAPACK to CARAVEL by setting the installation option **lapack-path**. With LAPACK selected, the integrand matrix from equation (3.47) is inverted by PLU decomposition in case of square matrices or using QR decomposition otherwise. In some specific cases, this combination can be more precise than using `Eigen` and its decomposition routines.

CLN and GiNaC: The last set of dependencies of CARAVEL is used for integral evaluation needed to obtain numerical results for one- or two-loop amplitudes and their remainders. For four-point processes CARAVEL relies on the MPLs implemented in `GiNaC` [79]. `GiNaC` itself needs `CLN` [114] as additional dependency if the MPLs have to be evaluated at high-precision. To install `CLN` 1.3.6, the commands of listing 4.7 can be used.

```
1 > wget https://www.ginac.de/CLN/cln-1.3.6.tar.bz2
2 > tar xvjf cln-1.3.6.tar.bz2
3 > cd cln-1.3.6
4 > ./configure --prefix=<install-dir>
5 > make
6 > make install
```

Listing 4.7: Commands to download, build and install the `CLN` library

`CLN`'s newest version number can be found on in reference [114] and then used instead of 1.3.6 in the commands of listing 4.7. To install `GiNaC` 1.7.10, the commands of listing 4.8 can be used.

```
1 > wget https://ginac.de/ginac-1.7.10.tar.bz2
2 > tar xvjf ginac-1.7.10.tar.bz2
3 > cd ginac-1.7.10
4 > ./configure --prefix=<install-dir>
5 > make
6 > make install
```

Listing 4.8: Commands to download, build and install the `GiNaC` library

Just as in the case of `CLN`, `GiNaC`'s most current version number can be found in reference [79] and then used instead of 1.7.10 in the commands of listing 4.8.

pentagon-library: The integral functions needed for five-point processes are not available within `GiNaC`. There is a convenient basis of transcendental functions spanning both the massless one-loop and the massless two-loop amplitudes, which are called pentagon functions [25]. With Caravel we provide an implementation of a full set of pentagon functions derived from a version described by reference [25] which Caravel can link against. This version is called `pentagon-library` and is available publicly [31]. To install `pentagon-library`, the commands of listing 4.9 can be used.

```
1 > git clone git@gitlab.com:caravel-public/pentagon-library.git
2 > cd pentagon-library/
3 > mkdir build
4 > cd build
5 > meson .. -Dprefix=<install-dir> -Dlibdir=lib
6 > ninja
7 > ninja install
```

Listing 4.9: Commands to download, build and install the `pentagon-library`.

The values of the amplitudes in the reference [54] were produced using `pentagon-library`, however from Carvavel 0.2.0 on it is also possible to produce output compatible with the faster `PentagonFunctions++` [36].

Open MPI: Caravel is written in a parallelized manner—multiple instances of the `Builder` class from the module **Forest** can be distributed among the available processor cores of a machine to speed up coefficient determination. Also, it is natural to parallelize the functional reconstruction of coefficients

by distributing their evaluation among the computers participating in a computer cluster. For these tasks CARAVEL relies on `MPI` [115], where the results from references [30, 83, 84, 88] were obtained using the `open-mpi` implementation [116]. To install `open-mpi` 4.0.5, the commands of listing 4.10 can be used.

```
1 > wget https://download.open-mpi.org/release/open-mpi/v4.0/\
2 > openmpi-4.0.5.tar.gz
3 > tar xzf openmpi-4.0.5.tar.gz
4 > cd openmpi-4.0.5
5 > ./configure --prefix=<install-dir>
6 > make
7 > make install
```

Listing 4.10: Commands to download, build and install `open-mpi`.

`open-mpi`'s newest version number can be found in reference [116] and then used instead of v4.0 and 4.0.5 in the commands of listing 4.10. Parallelized programs using CARAVEL can then be run by typing

```
mpirun -np <ncores> <program-name> "<par1>" "<par2>" ...
```

where `<ncores>` is the number of processor cores which shall be dedicated to the CARAVEL program `<program-name>` and `<par`*i*`>` are the parameters of the program `<program-name>`, which have to be enclosed in quotation marks `""`. The names of the examples inside the `examples` directory as well as the names of the programs inside the `my_programs` directory have to start with the string `mpi_` or `MPI_` or to end with the string `_mpi` or `_MPI` respectively if they shall be run using `MPI`; see also the file `CONTRIBUTING.md` from CARAVEL's repository.

4.5. Other External Tools

In section 4.1 we described CARAVEL's modules, which are linked against by the programs using CARAVEL for amplitude evaluation and coefficient reconstruction. Some of those modules are not fully automated and depend on external input in some calculation steps. Furthermore, some parts of the source code of the modules are machine-generated. To generate this input and the mentioned source code, we use external tools, which will have to be adjusted to future processes, for which CARAVEL will calculate coefficients and amplitudes.

In particular, the generation of the propagator structure hierarchy Δ as in equation (3.46) is automated in the case of one-loop amplitudes and two-loop four-point amplitudes. However, Δ has to be generated and saved to a headed list style file individually for every considered two-loop five-point amplitude. When

calculating two-loop corrections to processes with three particles in the final state, CARAVEL obtains information about the relevant propagator structures and the color decomposition of the considered amplitude, as described in section 3.2, at runtime from the corresponding headed list file. We call the file containing propagator structure hierarchy and color information as well as the tool used for its generation the "**process-library**". The **process-library** is produced using `qgraf` [117]. To obtain the color decomposition and to generate the hierarchy Δ for an amplitude, the **process-library** uses the color-dressed equivalents of the possible tree amplitudes $\mathcal{A}_i^{(0)}(\ell_l^\Gamma)$ on the left-hand side of equation (3.47) as vertices in the `qgraf` model file. Using this model file makes `qgraf` generate all color-dressed cuts relevant for the considered amplitude. In a second step, a color decomposition [118, 119] as described in section 3.2 is performed on the cut diagrams. They are then assembled into a hierarchy according to their propagator structures. Their position in the hierarchy defines the partial ordering $\geq$ on them as described in section 3.6. Finally, diagrams with D_s-dimensional internal quarks are simplified using scalar fields as described in section 3.1, which can be used if the option **ds-strategy** is set to `decomposition`. The result of **process-library** is then an input file for CARAVEL specific to the defining amplitude. It contains the color indices of the amplitude, the cut diagrams defining the left-hand side of equation (3.47) and their relations among them due to propagator cutting. During the execution of CARAVEL the **Forest** module then obtains the cut diagrams from the **process-library** to calculate the trees $\mathcal{A}_i^{(0)}(\ell_l^\Gamma)$ on the left-hand side of equation (3.47), while the **Coefficient provider** uses the relations among their propagator structures to perform the summation on the right-hand side of the equation.

There is another large block of CARAVEL's source code generated by external tools. It is responsible for the evaluation of the master integrals $\mathcal{I}_{\Gamma,i}$ as defined in equation (3.45) as well as the parametrization of the master integrands and surface terms on the right-hand side of equation (3.47). This part of CARAVEL is generated with the help of the scripts bundled to form the external tool **CaravelDatabase**. **CaravelDatabase** expects a definition of a master integral basis and appropriate surface terms as input. Every integral of such a basis contains the exact propagator structure information encoded in a `CaravelGraph` structure which is described in appendix C. Furthermore, it features its defining numerator insertion as described in section 3.6 and lastly information about its integral function decomposition as introduced in section 3.5. For each of the given basis integrals and surface terms, the scripts of **CaravelDatabase** transform this information to C++ code. They add the integrand or surface term to the **FunctionSpace** module as well as the corresponding integral function decomposition to the **IntegralLibrary** in the case of master integrands. Lastly, the `meson` build files for the **FunctionSpace** and the **IntegralLibrary** modules are updated to include routines to build the added integral basis. The new basis becomes available to

programs linking against Caravel after recompilation.

Caravel can determine coefficients of a master integral decomposition of an amplitude as in equation (3.44) as well as of an integral function decomposition of an amplitude or a remainder as in equations (3.42) and (3.43). The integral function decompositions are obtained from the master integral decompositions by writing every master integral as a linear combination of integral functions. For massless five-point master integrals such a representation is available as a headed list file within the `Mathematica` package `PentagonMI` [36]. The external tool **IntegralReplacements** converts an integral function decomposition inside `PentagonMI` into an input file readable by Caravel. For the determination of the integral function decomposition of an amplitude, after the **Coefficient provider** has calculated the integral coefficients, the **Integral provider** decomposes the master integrals it has got from **CaravelDatabase** into corresponding integral functions according to the information from **IntegralReplacements**. It then combines the integral function coefficients with the ones from **Coefficient providers** resulting in an integral function decomposition of the amplitude or the remainder.

Finally, there is a publicly available tool, which automatizes the installation process of Caravel's dependencies as described in section 4.4. It is called **caravel-deps-setup** and is available publicly [31]. To use it, `Rust` [120] has to be installed. This can be done by executing

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

from the command line and following the on-screen instructions. To install Caravel's dependencies, the commands of listing 4.11 can be used, after which Caravel can be built and installed as described in section 4.2.

```
1 > git clone git@gitlab.com:caravel-public/caravel-deps-setup.git
2 > cd caravel-deps-setup
3 > ./target/release/caravel-deps-setup <install-dir> -s
4 > source source.sh
```

Listing 4.11: Commands to download, build and install all Caravel dependencies.

4.6. Examples

We now describe how to link against Caravel and how to select what scattering amplitudes one wants to compute with it. More information on the presented examples can be found in the file `examples/README.md` from Caravel's repository. The examples we provide can calculate numerical values for tree-level, one- and two-loop amplitudes with four or five external particles and functionally reconstruct the respective master integral coefficients. Caravel directly computes

the color-stripped helicity amplitudes $\mathcal{A}^{(L)}$ from equation (3.44) numerically, or analytically reconstructs the master integral coefficients $c_{\Gamma,i}$ from equation (3.44). The amplitudes $\mathcal{A}^{(L)} = \mathcal{A}^{(L)}(1_{f_1}^{h_1}, \ldots, n_{f_n}^{h_n})$ depend on $n = 4$ or $n = 5$ particles. Every particle k has a defined type f_k and a helicity h_k. As the external particles are in four dimensions, they have two possible helicity states given in equation (2.64) if the particle is a fermion or in equation (2.73) if the particle is a vector boson. In addition, for QCD amplitudes, we consider the leading-color approximation as discussed in section 3.2. In particular, we look at the limit where the number of colors N_c is large, keeping the fraction N_f/N_c constant, where N_f is the number of fermions in closed fermion loops. With this assumption, the amplitudes $\mathcal{A}^{(L)}$ can be further decomposed to

$$\mathcal{A}^{(L)}(1_{f_1}^{h_1}, \ldots, n_{f_n}^{h_n}) = \mathcal{A}^{(L)[0]} + \frac{N_f}{N_c}\mathcal{A}^{(L)[1]} + \ldots + \left(\frac{N_f}{N_c}\right)^L \mathcal{A}^{(L)[L]}\,. \tag{4.1}$$

CARAVEL then computes the amplitudes $\mathcal{A}^{(L)[i]}$. Note that for gravity amplitudes the color-stripped leading-color amplitudes $\mathcal{A}^{(L)[0]}$ are the same as the amplitudes $\mathcal{M}^{(L)}$ as in equation (3.12). We factor out a normalization of the integrals so that they are given by

$$\mathcal{I}_{\Gamma,i} = \int \left(\prod_{j=1}^{L} \frac{d^D \ell_j}{(2\pi)^D}\right) \frac{m_{\Gamma,i}(\ell_l)}{\prod_{k\in P_\Gamma} \rho_k(\ell_l)}\,. \tag{4.2}$$

Particle types Arguments can be passed to CARAVEL using headed list expressions. One can pass the information about the process one wants to compute the amplitude for with the `PartialAmplitudeInput` parameter, as for example

```
PartialAmplitudeInput[Particles[Particle[u,1,qbp],
                                Particle[ub,2,qm],
                                Particle[photon,3,m],
                                Particle[photon,4,p],
                                Particle[photon,5,p]],
                      ColourStructure[NfPowers[1]]]
```

The fermions `u`, `ub`, `d` and `db` can have the helicity states `qbp` and `qm`, the vector bosons `gluon` and `photon` have states `p` and `m` and gravitons `G` have states `hpp` and `hmm`. The selectable particles with their respective helicity states are shown in table 4.1. With the additional parameter `ColourStructure[NfPowers[`i`]]` one can set the number of closed fermion loops in the desired amplitude, so the number i in $\mathcal{A}^{(L)[i]}$ in equation (4.1). Note that even though $\mathcal{A}^{(2)[1]}(1_q^{h_1}, 2_{\bar{q}}^{h_2}, 3_\gamma^{h_3}, 4_\gamma^{h_4}, 5_\gamma^{h_5})$ includes non-planar contributions in the leading-color approximation, CARAVEL will always calculate the partial amplitude spanned by the planar subset of the corresponding master integrals, as further described in section 5.1.

Type	`field`	`state`
Gluon	`gluon`	`p`, `m`
Photon	`photon`	`p`, `m`
Quark	`q`, `u`, `d`, `c`, `s`, `b`	`qbp`, `qm`
Anti-quark	`qb`, `ub`, `db`, `cb`, `sb`, `bb`	`qbp`, `qm`
Graviton	`G`	`hpp`, `hmm`

Table 4.1.: The fields, which can be a parameter of `Particle` with their respective helicity states.

4.6.1. Numerical Amplitudes

CARAVEL can either evaluate amplitudes on a given phase space point, including the evaluation of integral functions delivered by external libraries it links to, or functionally reconstruct the amplitude's coefficients either of a master integral or integral function decomposition. The release version of CARAVEL [31] comes with examples for both tasks. Here we describe the numerical calculation of tree, one- and two-loop amplitudes, and two-loop finite remainders.

Kinematics If one does not specify anything, the numerical example programs will take values for the external momenta, which yield the results of reference [54]. However, one can specify the kinematics by defining twistor parameters as described in section 3.7, which then determine the kinematics. In order to do that, one has to provide the option `TwistorParameters` with two parameters `s12` and `s23` for four-point scattering and five parameters `x0`, ..., `x4` in the case of five-point scattering.

Tree-Level Amplitudes The example program to calculate tree-level amplitudes is called `treeamp`. To specify the process for the tree-level amplitude it is enough to pass the `Particles` list to it

```
> ./treeamp "Particles[Particle[..],...,Particle[..]]"
```

The result is normalized as described in section 3.1. To calculate gravity tree-level amplitudes the option **gravity-model** has to be set to `Cubic`. There are two possibilities to select on what phase space point the amplitude is to evaluate. If one does not provide any additional information, CARAVEL will generate a random phase space point and print out the point in addition to the result of `treeamp`. Or one creates a file called `treeampPSP.dat` in the execution directory and saves the phase space point one wants the tree-level amplitude to be evaluated on to the file. For example one can create the file with the entries

```
  1/3       1/3    -2        2
 -5/16      1/4    -9/64    15/64
329/144  -355/144  17/16   -25/48
-83/36    271/144  69/64  -329/192
```

which makes CARAVEL evaluate the tree-level amplitude defined by the particles in the `Particle` list on the point

$$p_1 = \left(\frac{1}{3}, \frac{1}{3}, -2, 2\right) , \qquad p_2 = \left(-\frac{5}{16}, \frac{1}{4}, -\frac{9}{64}, \frac{15}{64}\right) ,$$
$$p_3 = \left(\frac{329}{144}, -\frac{355}{144}, \frac{17}{16}, -\frac{25}{48}\right) , \qquad p_4 = \left(-\frac{83}{36}, \frac{271}{144}, \frac{69}{64}, -\frac{329}{192}\right) .$$

For finite field kinematics the installation option **finite-fields** has to be selected and an additional parameter `Cardinality[p]` has to be passed to `treeamp`

```
> ./treeamp "Particles[...]" "Cardinality[p]"
```

`Cardinality[p]` selects the characteristic of the field, which has to be a prime number larger than 2^{30}, but smaller than 2^{31} because CARAVEL implements a 32 bit Barrett reduction algorithm [121] for the modular arithmetic. The file `src/misc/Moduli.hpp` provides many possible cardinalities, where CARAVEL also chooses from during rational reconstruction. The metric $(+,-,+,-)$ is used with finite field kinematics. The specification of a phase space point inside `treeampPSP.dat` is mandatory for finite field calculations.

To calculate results with double-double or quadruple-double precision the desired precision has to be passed to `treeamp` using the `HighPrecision[prec]` setting, where `prec=HP` for double-double and `prec=VHP` for quadruple-double precision. For this feature to work, the option `precision-QD` has to be set to the respective value of `prec` during installation.

One-Loop Amplitudes The example program to evaluate one-loop amplitudes up to the order ϵ^2 in the regularization parameter is `amplitude_evaluator_1l`, as needed, for example, for the determination of the pole structure of the respective two-loop amplitudes. The example can evaluate partonic four- and five-point amplitudes and thus can reproduce the results from reference [54]. Here, the whole `PartialAmplitudeInput` list is needed to be passed to `amplitude_evaluator_1l`. As has been described at the beginning of the section, it is possible to select to calculate $\mathcal{A}^{(1),[1]}$, that is an amplitude with a closed fermion loop instead of $\mathcal{A}^{(1)[0]}$ by passing `ColourStructure[NfPowers[1]]` as a second argument to the `PartialAmplitudeInput` list.

By default, the evaluation is performed on the phase space point given by

$$\begin{aligned} p_1 &= \left(\frac{1}{2}, \frac{45}{272}, \frac{45\mathrm{i}}{272}, \frac{1}{2}\right), & p_4 &= \left(-\frac{1169}{2652}, \frac{2165}{10608}, -\frac{13459\mathrm{i}}{38896}, -\frac{5075}{9724}\right), \\ p_2 &= \left(-\frac{1}{2}, 0, 0, \frac{1}{2}\right), & & \\ p_3 &= \left(\frac{21}{26}, -\frac{21}{26}, -\frac{5\mathrm{i}}{26}, -\frac{5}{26}\right), & p_5 &= \left(-\frac{973}{2652}, \frac{581}{1326}, \frac{1813\mathrm{i}}{4862}, -\frac{2779}{9724}\right), \end{aligned} \tag{4.3}$$

which is the same as the one used in reference [54]. It corresponds to the Mandelstam invariants

$$\begin{aligned} s_{12} = -1, \quad s_{23} = -8/13, \quad s_{34} = -1094/2431, \\ s_{45} = -7/17, \quad s_{51} = -749/7293\ . \end{aligned} \tag{4.4}$$

Suppose one wants to evaluate the amplitude on a different phase space point. In that case, one has to define it with the additional parameter `TwistorParameters` as described in the kinematics paragraph above. The command to compute a one-loop four-point amplitude with $s_{12} = -\frac{1}{3}$ and $s_{23} = -\frac{1}{5}$ for example has the form

```
> ./amplitude_evaluator_1l "PartialAmplitudeInput[...]" \
                           "TwistorParameters[-1/3, -1/5]"
```

Note that without additional settings, the example has access to integrals in the Euclidean region only, so that the parameters have to be chosen in such a way that all Mandelstam invariants become negative.

In contrast to `treeamp`, `amplitude_evaluator_1l` works only in finite fields. That is, the option **finite-fields** has to be enabled for it to compile. Furthermore, loop-level amplitudes need at least one set of integrals to be available, and they need the option **field-ext-fermions** to be enabled if some of the involved particles are fermions. The example becomes available if the options **finite-fields** as well as **field-ext-fermions** were enabled during installation and if the option **integrals** has been set to `all`.

The amplitudes are normalized by the corresponding tree-level amplitudes as in equation (3.10) if these do not vanish. If the tree-level amplitudes do vanish, they are normalized by the spinor weight factor given in equation (3.9).

Two-loop Amlitudes To evaluate two-loop amplitudes, the example program called `amplitude_evaluator_2l` can be used. It requires input in the same format as `amplitude_elaluator_1l`, which has been described before. Just as `amplitude_evaluator_1l`, it is capable of calculating four- or five-point QCD amplitudes, at two-loop order. By default, the amplitude is evaluated on the

point defined in equation (4.3), thus reproducing the results of reference [54]. Without any additional settings, it can evaluate an amplitude on a Euclidean point, which can be provided using the `TwistorParameters` argument in the same way as for `amplitude_evaluator_1l`. Having read the particle types and helicities defining the two-loop amplitude from the command line, the **Coefficient provider** determines the values of the corresponding master integral coefficients in multiple finite fields, from which it then reconstructs the values for the coefficients in the rational numbers. The coefficient determination is performed distributed among all processor cores available on the computer. At the next step, the **Integral provider** combines the integral coefficient of every master integral with the coefficients of its integral function decomposition and performs cancellations between the same integral functions appearing in different master integrals contributing to the amplitude. In this way, the amplitude is represented as an integral function decomposition, with the coefficients being series expansions in the regularization parameter. At the last step, the amplitude is evaluated by evaluating the integral functions either with `GiNaC` for four-point processes or with the `pentagon-library` for five-point processes. In analogy to the one-loop example, the two-loop amplitude is normalized by the corresponding tree-level amplitude as in equation (3.10) if it does not vanish. It is normalized by the respective spinor weight factor given in equation (3.9) if the corresponding tree-level amplitude does vanish. Just as `amplitude_evaluator_1l`, this example works in finite fields only. When in `amplitude_evaluator_1l` the amplitude was determined as a series expansion in the regularization parameter up to second order, in the two-loop analogue `amplitude_evaluator_2l` the amplitude is determined up to finite order. `amplitude_evaluator_2l` needs the same installation options to be enabled as `amplitude_evaluator_1l` in order to compile. That is for the example to work, the installation option **finite-fields** has to be activated, **field-ext-fermions** set to `true` and **integrals** set to `all`. In addition to the command line settings available for `amplitude_evaluator_1l`, one can add `Verbosity[All]` to see additional output concerning the currently evaluated generation of the coefficient hierarchy. While all available one-loop amplitudes are evaluated in a couple of seconds, N_f^0 two-loop amplitudes need minutes for the rational reconstruction of the master integral coefficients and again $\mathcal{O}(10\text{min})$ for the evaluation of the integral functions in double precision. The time for the first evaluation of an amplitude will be considerably longer than the times for the succeeding. This difference occurs because some hierarchy information and the universal denominators of the amplitude are cached in a warm-up file at the first run so that they don't need to be calculated in future runs.

Finite Remainder The example `finite_remainder_2l` is a combination of the two examples described before and needs the same command line input. There

is an additional verbosity setting `Verbosity[Remainder]` available. This setting will print explicit information about the coefficients of the integral functions that parametrize the remainder and their poles. In contrast to the one- and two-loop examples `finite_remainder_2l` only accepts five-point input. Because of that, among the activated options **finite-fields** and **field-ext-fermions** it is sufficient that **integrals** is set to **pentagons**. When evaluating `finite_remainder_2l`, after the two-loop amplitude has been computed as in `amplitude_evaluator_2l`, the correct pole structure as described in section 3.8 is subtracted from it. During this procedure, the relevant one-loop amplitudes are evaluated. The so obtained integral function coefficients are evaluated in multiple finite fields to reconstruct them in rational numbers. After the subtraction of the poles, the program checks whether all divergences have vanished indeed, which provides an additional consistency check of the amplitude, which is not present in `amplitude_evaluator_2l`. As the timings for one-loop amplitudes are negligible regarding the two-loop timings, for the same processes, the evaluation of the remainders takes roughly the same time as the evaluation of the two-loop amplitudes.

4.6.2. Coefficient Reconstruction

While in the previous examples, the respective amplitudes were evaluated on one phase space point numerically, CARAVEL can reconstruct coefficients either of a master integral or integral function decomposition of an amplitude or a remainder. The coefficients have to be evaluated on multiple phase space points during their functional reconstruction. Thus, it is convenient to parallelize this procedure, which CARAVEL does by distributing the **Coefficient providers** either among the available threads on a local machine, among multiple cores, or multiple computers on a cluster using `MPI` [115]. The examples of functional coefficient reconstruction coming with CARAVEL and presented here use `MPI` for parallelization.

CARAVEL has methods for univariate rational and multivariate polynomial reconstruction. Univariate reconstruction is used to reconstruct four-point amplitudes, where, by setting $s = 1$ and later regaining the s dependence from pdimensional analysis, the reconstruction can be performed in one variable $x = t/s$. It furthermore is used to determine the denominators of five-point amplitudes as functions of the respective master integral basis alphabet. Multivariate reconstruction is needed for five-point processes where, by again setting $s_{12} = 1$, four independent parameters span the phase space and thus remain for reconstruction. There are examples for both univariate and multivariate reconstruction coming with CARAVEL. They have the same output format, which is described in the following paragraph.

`NumeratorLabel`	Master-integral numerator
`scalar`	1
`mu2`	μ^2
`mu4`	$(\mu^2)^2$
`dbTensor`	$(p_1 + \ell_2)^2$

Table 4.2.: Possible numerator insertions as given in the output files of the functional reconstruction programs and corresponding expressions. The momentum routing defining `dbTensor` can be read off from figure 4.2.

Output The output of the reconstruction programs is saved to the subdirectory `analytics` of the execution directory. The `analytics` directory, in turn, has three subdirectories. In the first subdirectory called `amplitudes_<char>`, the finite field coefficients together with strings identifying the corresponding integrals are saved in headed list format. Here, `<char>` is the characteristic of the finite field set by the program, in which the calculation has been performed. The integral strings are headed lists with the head naming the integral according to its propagator structure, like `Box` or `BubbleBubble`. They contain the integral's numerator insertion as given in table 4.2 as the first entry and a list of inverse propagators `D`i as a second entry. The μ insertions appearing at one-loop level are products of the $(D-4)$-dimensional part of the loop momenta, which does not lie in the physical space spanned by the external momenta, that is

$$\mu^2 = \ell^{(D-4)} \cdot \ell^{(D-4)}\,, \quad \text{where} \quad \ell^{(D-4)} \cdot k_i = 0\,. \tag{4.5}$$

The integral strings have thus the form

```
PropagatorStructure[NumeratorLabel, {D1, D2, ..., Dm}]
```

The inverse propagators are given in terms of the internal momenta `l1` and `l2` and the external momenta labeled `k`i, where all momenta are defined outgoing. After having determined the analytic form of the coefficients in a finite field, the example programs attempt to reconstruct the values of the coefficients in the rational numbers. When the program obtains a candidate set of coefficients in the rational numbers, it evaluates the amplitude numerically in a different finite field and compares the result against the rational coefficients mapped to that finite field. If both evaluations agree, the rational reconstruction has been successful. In that case a second directory called `amplitudes_rational` is created, which contains the analytic form of the coefficients in the rational numbers. The files in `amplitudes_rational` have the same structure as the ones in the `amplitudes_<char>` directory. The functionally reconstructed amplitudes are normalized the same way as the numerically evaluated, that is by their respective

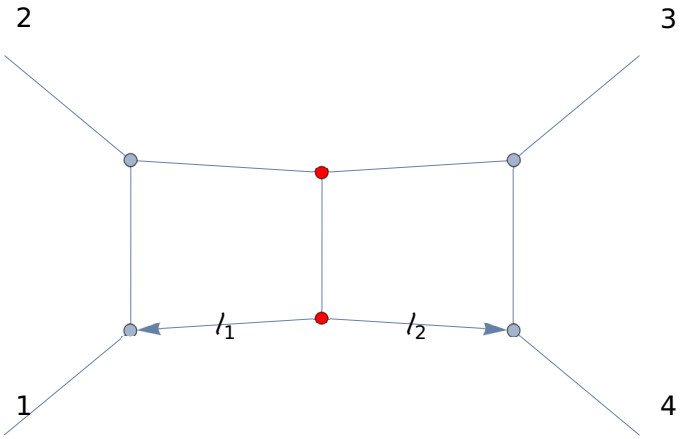


Figure 4.2.: Here the momentum routing for the integral string `BoxBox` produced by the analytic reconstruction examples is shown. All momenta are defined as outgoing, the external momenta k_i are labeled by their indices i.

tree amplitudes as in equation (3.10) if these do not vanish or by the corresponding spinor weight factor given in equation (3.9) if the respective tree amplitudes do vanish. Inside CARAVEL, propagator structures are represented using `CaravelGraph` lists as they are described in appendix C, rather than by sets of inverse propagators. In order to relate the both propagator structure descriptions, inside the `analytics` directory there is a third directory called `integral_info`, where a mapping between the integral strings used to parametrize the amplitude and corresponding `CaravelGraph` structures is defined. A graph structure facilitates the representation of an integral and can for example be used by the code `math/CaravelGraph.m` provided with CARAVEL to display the propagator structure graphs. To do that, using `Mathematica` one can evaluate

```
getMathGraph[CaravelGraph[...]]
```

from `math/CaravelGraph.m` on the graph one wishes to display, which results in a representation of the graph's momentum routing as shown for example in figure 4.2. In the following we describe the both coefficient reconstruction examples in more detail.

Univariate Two-Loop Coefficients The both reconstruction examples are parallelized using `MPI`. The first example program performing a functional reconstruction is called `4parton_2loop_analytics_MPI`. It reconstructs the master integral coefficients of two-loop four-point QCD amplitudes. To define the amplitude which is to be reconstructed it requires the same input as the numerical programs described in section 4.6.1 and is run using `mpirun` as described in section 4.4. For example, the command

```
> mpirun -np <ncores> ./4parton_2loop_analytics_MPI \
  "PartialAmplitudeInput[Particles[Particle[gluon,1,p],
  Particle[gluon,2,p],Particle[gluon,3,p],
  Particle[gluon,4,p]]]"
```

reconstructs the master integral coefficients for the amplitude $\mathcal{A}^{(2)[0]}(1_g^+, 2_g^+, 3_g^+, 4_g^+)$, which will then be normalized by the corresponding spinor weight factor as defined in equation (3.9). The parameter `<ncores>` specifies the number of processor cores which should be dedicated to the reconstruction. The rational and finite field coefficients for the amplitude will be given as a series in the dimensional regularization parameter `ep` with the series coefficients being rational functions in terms of the Mandelstam variables $s = (p_1 + p_2)^2$ and $t = (p_2 + p_3)^2$. In order for `4parton_2loop_analytics_MPI` to compile, the installation options **finite-fields** and **field-ext-fermions** have to be activated. As in the case of functional reconstruction, the integrals are not evaluated, so the option **integrals** is not required for the example to work.

Multivariate One-Loop Coefficients Finally, there is an example, which performs multivariate reconstruction. Functional reconstruction in multiple variables is more involved, since as described in section 3.9 prior to reconstruction the denominator of the coefficients has to be fitted by the letters of the integral basis alphabet first. As the reconstruction of most of the two-loop five-point amplitudes is computationally intense, the example `5parton_1loop_analytics_MPI` is provided with CARAVEL to demonstrate multivariate functional reconstruction. . The example reconstructs master integral coefficients for one-loop five-point amplitudes and can be executed the same way like `4parton_2loop_analytics_MPI` now specifying five instead of four particles. That is for example, the command

```
> mpirun -np <ncores> ./5parton\_1loop\_analytics\_MPI \
  "PartialAmplitudeInput[Particles[Particle[gluon,1,p],
  Particle[gluon,2,p],Particle[gluon,3,p],
  Particle[gluon,4,p],Particle[gluon,5,p]]]"
```

reconstructs the coefficients for the amplitude $\mathcal{A}^{(1)[0]}(1_g^+, 2_g^+, 3_g^+, 4_g^+, 5_g^+)$, which will be normalized by the spinor weight factor defined in equation (3.9). In order to be available `5parton_1loop_analytics_MPI` needs the same installation options enabled as `4parton_2loop_analytics_MPI`, that is **finite-fields** and **field-ext-fermions**. The coefficients are presented as functions in the momentum twistors $x_0, \ldots, x_4$ as defined in section 3.7. Since in the one-loop case, the representation of the integrals is straightforward, `5parton_1loop_analytics_MPI` does not define any mappings of the used integral strings to the respective `CaravelGraph` structures. Instead, the `integral_info` directory contains mappings to `Strand` structures as they are defined in appendix C, which are sufficient

to characterize one-loop amplitudes. Every entry of a `Strand` list corresponds to a vertex of the propagator structure. It contains the information on whether the vertex is massless and a list of the indices of the momenta attached.

4.6.3. Photon Amplitudes

As part of this thesis we extended CARAVEL to be able to calculate amplitudes including photons.

In particular, we extended the QCD `Model` as part of the **Forest** module, which the authors of reference [30] have already used to determine the coefficients for five-parton amplitudes. In particular, we added photon fields and corresponding Feynman rules. Unlike gluons, photons have no color and appear unordered in the color-ordered diagrams. The functionality to treat unordered particles correctly was already implemented in CARAVEL, as it has been used for the four-graviton calculation [88]. We adapted it to implement photons as colorless particles, where special attention had to be paid since now colored as well as colorless particles could appear within one diagram, which did not happen for any other `Model` before. We took care of this feature by introducing an unordered container for uncolored particles for every off-shell current. As described in section 2.3, the quark-photon vertex differs from the quark-gluon vertex by a factor of $\sqrt{2}$. We implemented the quark-photon vertex in analogy to the already available quark-gluon vertex, where the information about the additional factor has been stored inside the object representing the coupling of the vertex. The coupling factors are not considered during the numerical evaluation of the tree-level amplitudes defining a propagator structure. Instead, they are combined symbolically, thus allowing calculations in a finite field without field extension.

To determine the relevant propagator structures for two-loop amplitudes involving photons, the **process-library** as described in section 4.5 had to be extended by photons. In particular, all relevant vertices of cut diagrams involving photons had to be added to the **process-library** to generate the corresponding **process-library** files. As with the off-shell currents extended by photons, special attention had to be paid to account for both colored and colorless particles appearing in one vertex.

We updated the external library **IntegralReplacements** and extending the module **Integral providers**. In particular, we implemented the integral function decomposition of the master integral basis we used for the photon amplitudes, as given in appendix D, into the integral functions of reference [36]. This decomposition allows an efficient evaluation of two-loop five-point processes in the physical region. Since the integral functions parametrize both the two- and the one-loop amplitudes for a process, the decomposition made it possible to reconstruct the coefficients of the finite remainder decomposition of the respective process in the physical region.

As the three-photon production amplitudes were the first which were calculated in the physical region by CARAVEL, the pole operators defined in the module **PoleStructure** had to be analytically continued appropriately.

Photons are available starting from version 0.2.0 of CARAVEL. From that version, the functional reconstruction examples described in section 4.6.2 can be used to reconstruct integral function coefficients for one-loop three-photon as well as two-loop two-photon production. As described in the previous paragraphs, in order to run the functional reconstruction examples, the installation options **finite-fields** and **field-ext-fermions** have to be activated. To reconstruct the one-loop five-point amplitude $\mathcal{A}^{(1)[0]}(1_q^+, 2_{\bar{q}}^-, 3_\gamma^+, 4_\gamma^+, 5_\gamma^+)$, which is the term $A^{(2,0)}$ in equation (5.9), one then has to execute

```
> mpirun -np <ncores> ./examples/5parton_1loop_analytics_MPI\
"PartialAmplitudeInput[Particles[Particle[u,1,qbp],\
                                 Particle[ub,2,qm],\
                                 Particle[photon,3,p],\
                                 Particle[photon,4,p],\
                                 Particle[photon,5,p]]]"\
  "CaravelSettingsInput[Setting[integrals::integral_family_1L,\
                                pentagon_functions_new],\
                        Setting[general::master_basis,\
                                pentagon_functions_new]]"
```

This example relies on the new pentagon functions as in reference [36] to be available, which is accounted for by setting the **integrals** option to `pentagons-new`. Similarly, to reconstruct the two-loop four-point amplitude $\mathcal{A}^{(2)[0]}(1_q^+, 2_{\bar{q}}^-, 3_\gamma^+, 4_\gamma^+)$, which also is the term $A^{(2,0)}$ in equation (5.9), one has to execute

```
> mpirun -np <ncores> ./examples/4parton_2loop_analytics_MPI\
"PartialAmplitudeInput[Particles[Particle[u,1,qbp],\
                                 Particle[ub,2,qm],\
                                 Particle[photon,3,p],\
                                 Particle[photon,4,p]]]"
```

In order for this example to work, the installation option **integrals** has to be set at least to `goncharovs`.

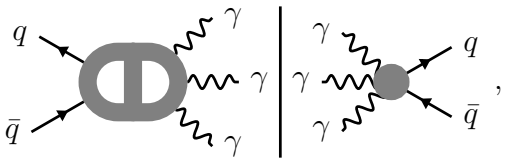
5. Two-Loop Two-Parton Three-Photon Amplitudes

This chapter presents our computation of the leading-color two-loop scattering amplitudes for the process $q\bar{q} \to \gamma\gamma\gamma$. These are the amplitudes required to construct the double-virtual contributions to the $\mathcal{O}(\alpha_s^2)$ corrections to three-photon production at hadron colliders. The calculation is performed with the numerical unitarity method, employing exact evaluations in a finite field of characteristic $\mathcal{O}(2^{32})$. Those evaluations are combined with a functional reconstruction algorithm to extract the analytic form of the amplitudes' master integral coefficients [94]. The contents of this chapter have been published in reference [84].

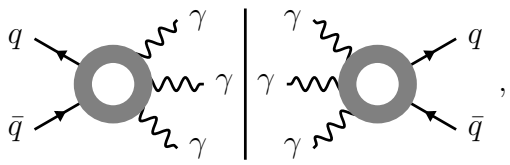
Our results have been employed in an NNLO QCD study for three-photon production at the LHC [37], which also compares to ATLAS data [35]. Previous to our publication, a similar calculation was performed [32]. However, the used expressions for corresponding leading-color two-loop amplitudes were too complex to be published in an analytic form. We considerably improved on this, producing compact, stable, and fast-to-execute expressions. More recently, the same group has reproduced our results [38].

5.1. Helicity Amplitudes

The two-loop interference diagrams contributing to NNLO QCD corrections to three-photon production at hadron colliders are schematically written as



and



where the first figure corresponds to two-loop amplitudes interfering with the tree-level amplitude and the second to the one-loop amplitude squared. Note that there is no tree amplitude for the process $gg \to \gamma\gamma\gamma$. Thus, the two-loop amplitude for gluon fusion does not contribute to the three-photon production at NNLO. Also, the $gg \to \gamma\gamma\gamma$ one-loop amplitude square vanishes due to charge conservation, as described in section 2.2.

The process involves five massless particles. As said in section 3.7, the amplitude is therefore a function of five Mandelstam invariants, which we choose as

$$\begin{gathered} s_{12} = (p_1+p_2)^2\,, \quad s_{23} = (p_2+p_3)^2\,, \quad s_{34} = (p_3+p_4)^2\,, \\ s_{45} = (p_4+p_5)^2\,, \quad s_{15} = (p_1+p_5)^2\,, \end{gathered} \tag{5.1}$$

as well as

$$\mathrm{tr}_5 = 4\mathrm{i}\varepsilon(p_1,p_2,p_3,p_4). \tag{5.2}$$

Here $\varepsilon(p_1,p_2,p_3,p_4) = \varepsilon_{\mu\nu\rho\sigma}p_1^\mu p_2^\nu p_3^\rho p_4^\sigma$, with $\varepsilon_{\mu\nu\rho\sigma}$ the Levi–Civita symbol. We use an all-outgoing convention for the momenta of the particles. The five Mandelstam invariants parametrize the amplitude we chose in equation (5.1). Nevertheless, since the photons have no color, all possible invariants s_{ij} with $i,j = 1,\ldots,5$ will appear in the analytic expressions we obtain for the amplitude. In the case of $\mathcal{A}_{+++}$ for example, if an expression depends on s_{23}, there will be analogous expressions depending on s_{24} and s_{25}, because photons 4_γ^+ and 5_γ^+ take the same places as 3_γ^+ in the Feynman diagrams representing $\mathcal{A}_{+++}$. This symmetry, in turn, means that there is no Euclidean region for the process $q\bar{q} \to \gamma\gamma\gamma$. Indeed, photons can attach at every point of the quark line so that the Euclidean region would correspond to every $s_{ij} < 0$. This is not possible, as follows e.g. from $0 = p_5^2 = (p_1+p_2+p_3+p_4)\cdot p_5 = s_{15}+s_{25}+s_{35}+s_{45}$. However, the physical region for the process is simple, because there are no photons in the initial state. According to equation (5.2), for physical, that is real, momenta then $(\mathrm{tr}_5)^2$ is negative. Since p_1 and p_2 are incoming, they have negative energy components in our convention. Using the Cauchy–Schwarz inequality, from this follows

$$s_{12}, s_{34}, s_{45} > 0\,, \qquad s_{23}, s_{15} < 0\,, \qquad \mathrm{tr}_5^2 < 0\,. \tag{5.3}$$

The process we have to consider is

$$q(p_1,h_1) + \bar{q}(p_2,h_2) \;\to\; \gamma(p_3,h_3) + \gamma(p_4,h_4) + \gamma(p_5,h_5)\,, \tag{5.4}$$

where p_i are the momenta and h_i the helicities of the particles. As we describe in section 3.2, we decompose the amplitude $\mathcal{M}(1_q^{h_1}, 2_{\bar{q}}^{h_2}, 3_\gamma^{h_3}, 4_\gamma^{h_4}, 5_\gamma^{h_5})$ as a linear combination of a color-dependent part and a kinematics-dependent part. To normalize the amplitudes as in section 3.1, we additionally strip off a spinor

Helicity	Expression	Helicity	Expression
$+-\ +++$	$\mathcal{A}_{+++}(1,2,3,4,5)$	$-+\ +++$	$\mathcal{A}_{+++}(2,1,3,4,5)$
$+-\ -++$	$\mathcal{A}_{-++}(1,2,3,4,5)$	$-+\ -++$	$\mathcal{A}_{-++}(2,1,3,4,5)$
$+-\ +-+$	$\mathcal{A}_{-++}(1,2,4,3,5)$	$-+\ +-+$	$\mathcal{A}_{-++}(2,1,4,3,5)$
$+-\ ++-$	$\mathcal{A}_{-++}(1,2,5,3,4)$	$-+\ ++-$	$\mathcal{A}_{-++}(2,1,5,3,4)$
$+-\ --+$	$\mathbf{P}\mathcal{A}_{-++}(2,1,5,3,4)$	$-+\ --+$	$\mathbf{P}\mathcal{A}_{-++}(1,2,5,3,4)$
$+-\ +--$	$\mathbf{P}\mathcal{A}_{-++}(2,1,3,4,5)$	$-+\ +--$	$\mathbf{P}\mathcal{A}_{-++}(1,2,3,4,5)$
$+-\ -+-$	$\mathbf{P}\mathcal{A}_{-++}(2,1,4,3,5)$	$-+\ -+-$	$\mathbf{P}\mathcal{A}_{-++}(1,2,4,3,5)$
$+-\ ---$	$\mathbf{P}\mathcal{A}_{+++}(2,1,3,4,5)$	$-+\ ---$	$\mathbf{P}\mathcal{A}_{+++}(1,2,3,4,5)$

Table 5.1.: Relations to get every needed amplitude from the amplitudes we compute as defined in equation (5.6). $\mathbf{P}$ is parity conjugation here. The indices i_j in $\mathcal{A}_{\pm++}(i_1,\ldots,i_5)$ signify at what position the momentum p_{i_j} has to be inserted into $\mathcal{A}_{\pm++}$ to get the expression with the desired helicity.

weight factor, such that we can write the amplitude as

$$\mathcal{M}(1_q^{h_1},2_{\bar{q}}^{h_2},3_\gamma^{h_3},4_\gamma^{h_4},5_\gamma^{h_5}) := e_q^3\delta_{i_1 i_2}\Phi(1_q^{h_1},2_{\bar{q}}^{h_2},3_\gamma^{h_3},4_\gamma^{h_4},5_\gamma^{h_5})\mathcal{A}(1_q^{h_1},2_{\bar{q}}^{h_2},3_\gamma^{h_3},4_\gamma^{h_4},5_\gamma^{h_5})\,, \tag{5.5}$$

where e_q is the electric charge of the quark q, Φ the spinor weight given in equations (3.10) and (3.9) and $\mathcal{A}(1_q^{h_1},2_{\bar{q}}^{h_2},3_\gamma^{h_3},4_\gamma^{h_4},5_\gamma^{h_5})$ the helicity amplitudes which we compute. It turns out that there is only one independent two-loop helicity amplitude relevant for NNLO corrections to the process. Thus, for this process, it is convenient to use helicity amplitudes.

With

$$\begin{aligned}\mathcal{A}_{+++}(1,2,3,4,5) &:= \mathcal{A}(1_q^+,2_{\bar{q}}^-,3_\gamma^+,4_\gamma^+,5_\gamma^+)\,,\\ \mathcal{A}_{-++}(1,2,3,4,5) &:= \mathcal{A}(1_q^+,2_{\bar{q}}^-,3_\gamma^-,4_\gamma^+,5_\gamma^+)\,,\end{aligned} \tag{5.6}$$

and $\mathbf{P}\mathcal{A}_{+++}$, $\mathbf{P}\mathcal{A}_{-++}$ their parity conjugated versions we can get all 16 helicity amplitudes by permutation of the external legs and parity conjugation, see table 5.1. Here, the amplitudes have to be evaluated in the physical region (5.3). The tree-level amplitudes corresponding to $\mathcal{A}_{+++}$ and $\mathcal{A}_{---}$ always vanish, hence for NNLO QCD corrections only $\mathcal{A}_{-++}$ is required. We have still computed all amplitudes to all orders in the regularization parameter for completeness (for example, all are needed as part of the N^{3}LO correction to the process, where they are combined with the corresponding non-vanishing one-loop amplitudes).

The spinor weights are given by

$$\Phi_{+++} = \frac{[31]\langle 12\rangle^3\langle 13\rangle}{\langle 14\rangle^2\langle 15\rangle^2\langle 23\rangle^2}\,, \qquad \Phi_{-++} = \frac{1}{\mathcal{A}^{(0)}_{-++}}, \tag{5.7}$$

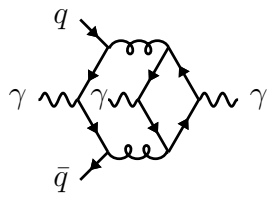


Figure 5.1.: Non-planar N_f contribution to the NNLO correction to $q\bar{q} \to \gamma\gamma\gamma$.

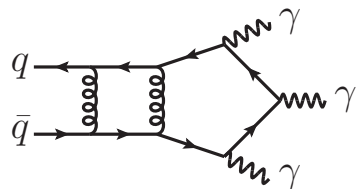
where the spinor products $\langle ij \rangle$ and $[ij]$ were defined in section 3.7, and $\mathcal{A}^{(0)}_{-++}$ is the tree-level amplitude. If we were to include all possible light quark loops, non-planar diagrams as, for example, shown in figure 5.1 would have to be considered. Although the calculation of non-planar massless five-particle Feynman integrals is complete [36, 122], in this work, we focus on calculating the planar leading-color contributions. The one- and two-loop amplitudes can be written as a sum of gauge-invariant parts

$$\begin{aligned} \mathcal{A}^{(1)} &= C_F A^{(1)}, \\ \mathcal{A}^{(2)} &= C_F^2 B^{(2,0)} + C_F C_A B^{(2,1)} \\ &\quad + C_F T_F N_f A^{(2,N_f)} + C_F T_F \sum\nolimits_{f=1}^{N_f} Q_f^2 \, \tilde{A}^{(2,N_f)}, \end{aligned} \tag{5.8}$$

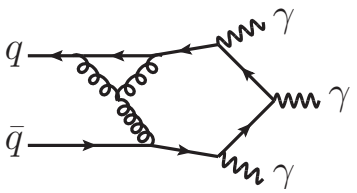
with $C_F = (N_c^2 - 1)/2N_c$ being the quadratic Casimir operator of the fundamental representation of the quarks, $T_F = 1/2$ and Q_f the ratios of the electric charges of a quark which has flavor f to the quark q from the initial state. For the contributions $B^{(2,0)}$ and $B^{(2,1)}$, there are no quark loops in the diagrams. Contributions $A^{(2,N_f)}$ have one N_f quark loop, where no photon attaches to. Contributions $\tilde{A}^{(2,N_f)}$ have also one N_f quark loop, where because of charge parity conservation exactly two photons attach to. Here, also non-planar diagrams contribute, as for example the one in figure 5.1. Example diagrams for every of the terms are given in figure 5.2. Taking the leading-color approximation where N_c is large and N_f/N_c is constant, terms scaling like $\mathcal{O}(N_c^{-2})$ are suppressed. One would expect that contributions from $A^{(2,N_f)}$ and $\tilde{A}^{(2,N_f)}$ are of the same order, but when looking at the analogous process $q\bar{q} \to \gamma\gamma$ the contributions from $\tilde{A}^{(2,N_f)}$ are actually subleading with respect to $A^{(2,N_f)}$ [37], so we expect them also to be small in our case. In the large N_c approximation and additionally expecting $\tilde{A}^{(2,N_f)}$ being small we thus can write

$$\begin{aligned} \mathcal{A}^{(2)} &= \frac{N_c^2}{4} \left(A^{(2,0)} + \mathcal{O}(N_c^{-2}) \right) + C_F T_F N_f A^{(2,N_f)}, \\ A^{(2,0)} &= B^{(2,0)} + 2B^{(2,1)}. \end{aligned} \tag{5.9}$$

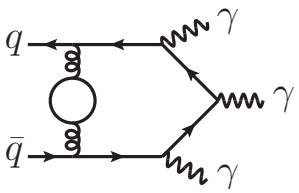
In the following, we focus on the calculation of the yet unknown ingredients for



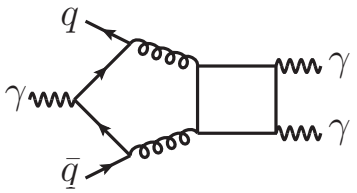
(a) Diagram proportional to C_F^2



(b) Diagram proportional to $C_F C_A$



(c) Diagram proportional to N_f



(d) Diagram proportional to $\sum_f Q_f^2$

Figure 5.2.: Example diagrams for each of the gauge-invariant contributions in equation (5.8).

the matrix element square of the process $q\bar{q} \to \gamma\gamma\gamma$, which are the "partial" helicity amplitudes $A^{(2,0)}_{+++}$, $A^{(2,N_f)}_{+++}$, $A^{(2,0)}_{-++}$ and $A^{(2,N_f)}_{-++}$. Every of the four pieces possesses a master integral decomposition as in equation (3.44). For example, the $A^{(2,0)}$ pieces can be constructed from the 8-propagator diagrams—the "maximal cuts"—shown in figure 5.3.

Since the photons are not color-ordered, they appear in every possible permutation. The cut hierarchy for this amplitude is then composed of 1945 propagator structures.

5.2. Finite-Remainder Definition

To simplify the analytic expressions $\mathcal{A}_{\pm++}$, it is convenient to remove all information that is intrinsically associated with lower-loop contributions [92]. Specifically, we subtract all IR divergent structures, including the $\mathcal{O}(\varepsilon^0)$ finite pieces, as has been described in section 3.8.

We define the so-called finite remainder [30, 92]

$$\mathcal{R} = \mathcal{A}_R^{(2)} - \mathbf{I}^{(1)}\mathcal{A}_R^{(1)} - \mathbf{I}^{(2)}\mathcal{A}_R^{(0)} + \mathcal{O}(\epsilon), \tag{5.10}$$

as described in section 3.8. The $\mathcal{A}_R^{(L)} \equiv \mathcal{A}_{R,\pm++}^{(L)}$ are the renormalized leading-color

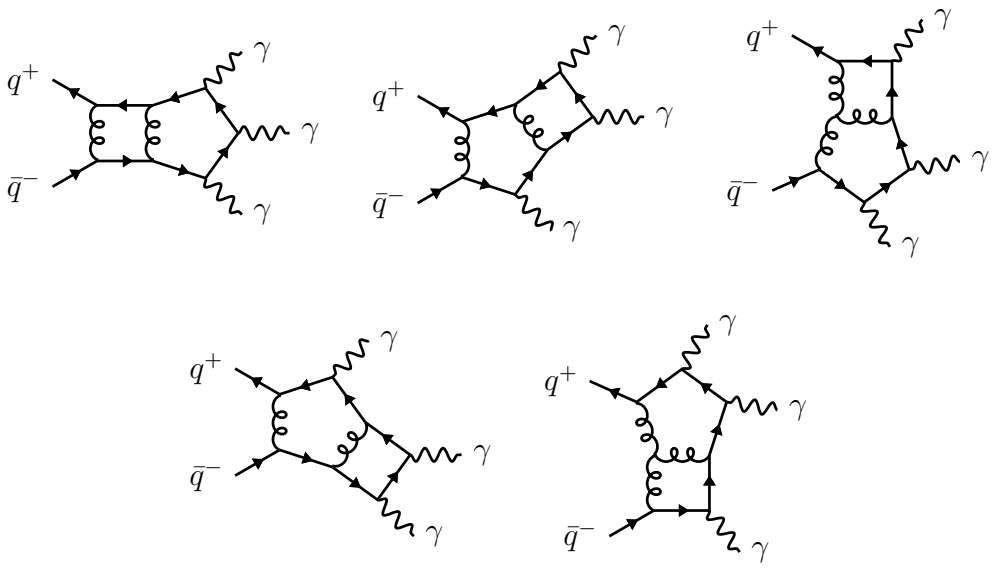


Figure 5.3.: Maximal cut diagrams contributing to the $\mathcal{A}^{(2,0)}$ partial amplitude (see equation (5.9)).

L-loop amplitudes, which are related to the bare amplitudes by [54]

$$\begin{aligned}
\mathcal{A}_R^{(0)} &= \mathcal{A}^{(0)}, \\
\mathcal{A}_R^{(1)} &= S_\epsilon^{-1}\mathcal{A}^{(1)}, \\
\mathcal{A}_R^{(2)} &= S_\epsilon^{-2}\mathcal{A}^{(2)} - \frac{\beta_0}{N_c\epsilon}S_\epsilon^{-1}\mathcal{A}^{(1)}\,,
\end{aligned} \tag{5.11}$$

where $S_\epsilon = (4\pi)^\epsilon e^{-\epsilon\gamma_E}$, γ_E is the Euler-Mascheroni constant and

$$\beta_0 = \frac{N_c}{3}\left(11 - 2\frac{N_f}{N_c}\right). \tag{5.12}$$

We use the 't Hooft–Veltman regularization scheme [55] as it was discussed in section 2.4. We have established the dependence of the amplitudes on D_s as it was explained in section 3.1 and consider now the amplitude at $D_s = 4-2\epsilon$. The functions $\mathbf{I}^{(k)}$ in equation (5.10) for the physical region are given by

$$\mathbf{I}^{(1)} \equiv \mathbf{I}^{(1)}(\epsilon) = -\frac{e^{\gamma_E\epsilon}}{\Gamma(1-\epsilon)}\left(\frac{1}{\epsilon^2} + \frac{3}{2\epsilon}\right)(-s_{12} - \mathrm{i}\varepsilon)^{-\epsilon}\,, \tag{5.13}$$

and $\mathbf{I}^{(2)}$ has a universal structure as in equation (3.61). In this case,

$$\mathbf{H}(\epsilon) = \frac{2e^{\gamma_E \epsilon} H_q}{\epsilon\Gamma(1-\epsilon)}\,, \tag{5.14}$$

with

$$H_q = \left(\frac{7\zeta_3}{4} + \frac{409}{864} - \frac{11\pi^2}{96}\right) + \left(\frac{\pi^2}{48} - \frac{25}{216}\right)\frac{N_f}{N_c}\,. \tag{5.15}$$

Using the equations from this section allows to regain the bare two-loop amplitudes up to finite order from the remainder $\mathcal{R}$ or vice versa. Similar to equation (5.9), in the leading-color approximation the remainders can be split into the part coming from diagrams with no closed fermion loops and a part coming from diagrams with one closed fermion loop, that is

$$\begin{aligned} \mathcal{R}^{(1)} &= C_F R^{(1)}, \\ \mathcal{R}^{(2)} &= \frac{N_c^2}{4}\left(R^{(2,0)} + \mathcal{O}(N_c^{-2})\right) + C_F T_F N_f R^{(2,N_f)}\,. \end{aligned} \tag{5.16}$$

5.3. Pentagon-Function Decomposition

The master integrals for one-loop amplitudes cannot be expressed through the two-loop master integrals at integrand level. That means, we also cannot parametrize the remainders by two-loop master integrals only anymore, as seen from equation (5.10). As we discussed in section 3.5, both, the one- and two-loop amplitudes, however are spanned by a basis of the same transcendental functions called pentagon functions [25, 76] $\{h_i\}$, which can be evaluated efficiently [36]. Since, as discussed in section 3.8, two-loop amplitudes have at most quartic poles in the regularization parameter, with the help of pentagon functions we can write the amplitudes from equations (5.8) and (5.9) as

$$A_h^{(2,j)} = \sum\nolimits_i \sum\nolimits_{k=-4}^{0} \epsilon^k d_{k,i}^j h_i + \mathcal{O}(\epsilon)\,, \tag{5.17}$$

and

$$A_h^{(1)} = \sum\nolimits_i \sum\nolimits_{k=-2}^{2} \epsilon^k d_{k,i}^{(1)} h_i + \mathcal{O}(\epsilon^3)\,, \tag{5.18}$$

where we have to expand $A_h^{(1)}$ to the second order in the regulator because of the double pole of $\mathbf{I}^{(1)}$. Thus, the remainders from equation (5.16) can also be written in terms of pentagon functions

$$R_h^{(2,j)} = \sum_{i\in B} r_i h_i\,. \tag{5.19}$$

Not only can these functions be efficiently evaluated, but their space has a smaller dimension than the one of master integrals which will become clear in the comparison of the amplitude and remainders reconstruction.

5.4. Amplitude Evaluation

To numerically compute the two-loop amplitude for three-photon production, we have extended the setup of reference [30]. As we described in sections 4.5 and 4.6.3, we have generated all color-dressed cut diagrams relevant for the amplitudes $A_h^{(2,j)}$ extending the **process-library** by photons, which uses `qgraf` [117] for the diagram generation. We then have decomposed the cut diagrams into a color- and a kinematics-dependent part following the procedure from references [118, 119]. We have defined a partial ordering on the set of the cut diagrams, thus defining the propagator structure hierarchy Δ for the process. We have used the master integrands and unitarity-compatible surface terms as described in [96] and section 3.6. In order to get the master integral coefficients $c_{\Gamma,i}$ from equation (3.45), we sampled values for the loop momenta ℓ_l^Γ which set the propagators of Γ on-shell, thus filling the linear system of equations (3.47). We evaluated the left-hand side of equation (3.47) using the off-shell recursion [63] of the Caravel framework [31], which we extended in such a way that it could perform calculations involving photons as we described in section 4.6.3. In particular, we have introduced color-unordered photon states and the quark-photon vertex to the recursion. The numerical evaluation was performed in a finite field of characteristic $\mathcal{O}(2^{32})$ employing Barrett reduction [121, 123]. After sampling a sufficient number of momenta ℓ_l^Γ, we have solved the linear system of equations (3.47) by PLU factorization and back substitution. With this procedure, given an external phase space point, we can extract all the master integral coefficients of equation (3.44) numerically.

We provide reference values for the amplitudes and remainders from equations (5.9) and (5.16) as well as the corresponding one-loop amplitudes evaluated on the phase space point

$$\begin{aligned} &s_{12} = 1.322500000, \quad s_{23} = -0.994109498, \quad s_{34} = 0.264471591, \\ &s_{45} = 0.267126049, \quad s_{15} = -0.883795230, \quad \mathrm{tr}_5 = -0.11382836\mathrm{i}, \end{aligned} \tag{5.20}$$

in tables 5.2, 5.3 and 5.4. The invariants are assumed to be exactly these values, without any (floating-point) rounding performed.

	ϵ^{-2}	ϵ^{-1}	ϵ^{0}	ϵ^{1}	ϵ^{2}
$A^{(1)}_{-++}$	-1.000000000	-3.174284697 $-3.141592654\mathrm{i}$	-3.437681197 $-16.69077768\mathrm{i}$	-4.542364174 $-48.29215997\mathrm{i}$	-28.34154945 $-104.73071151\mathrm{i}$
$A^{(1)}_{+++}$	0	0	-122.4876141 $-218.2099911\mathrm{i}$	-613.1620024 $-1772.249665\mathrm{i}$	-1264.781477 $-6727.583766\mathrm{i}$

Table 5.2.: Reference evaluations of all independent bare one-loop amplitudes on the phase space point of (5.20).

	ϵ^{-4}	ϵ^{-3}	ϵ^{-2}	ϵ^{-1}	ϵ^{0}
$A^{(2,0)}_{-++}$	0.500000000	2.257618031 $+3.141592654\mathrm{i}$	-3.317245357 $+20.90350063\mathrm{i}$	-55.54942686 $+44.34772277\mathrm{i}$	-248.7699347 $-87.79211669\mathrm{i}$
$A^{(2,N_f)}_{-++}$	0	0.1666666667	1.335872677 $+1.047197551\mathrm{i}$	4.646264515 $+12.872514370\mathrm{i}$	10.33373680 $+83.15472523\mathrm{i}$
$A^{(2,0)}_{+++}$	0	0	122.4876141 $+218.2099911\mathrm{i}$	-132.6755953 $+2049.613188\mathrm{i}$	-9927.845724 $+3575.607623\mathrm{i}$
$A^{(2,N_f)}_{+++}$	0	0	0	81.65840942 $+145.4733274\mathrm{i}$	895.9475013 $+2327.538099\mathrm{i}$

Table 5.3.: Reference evaluations of all independent bare two-loop amplitudes on the phase space point of (5.20).

$R^{(1)}_{-++}$	$-5.281908761 - 6.718468192\mathrm{i}$
$R^{(1)}_{+++}$	$-122.4876141 - 218.2099911\mathrm{i}$
$R^{(2,0)}_{-++}$	$-17.93042514 - 84.48074943\mathrm{i}$
$R^{(2,N_f)}_{-++}$	$8.536235118 + 25.51694192\mathrm{i}$
$R^{(2,0)}_{+++}$	$-2043.581205 - 3461.464426\mathrm{i}$
$R^{(2,N_f)}_{+++}$	$327.6279319 + 861.8112864\mathrm{i}$

Table 5.4.: Reference evaluations of all independent one-loop and two-loop remainders on the phase space point of (5.20).

5.5. Analytic Master-Integral Coefficients

The phenomenologically relevant NNLO information for a process can be obtained from the finite remainders as discussed in section 5.2, which are simpler than complete two-loop amplitudes. For example, dimensionally-regularized two-loop amplitudes generally have quartic poles in the regularization parameter, and every pole coefficient is a rational function of the external kinematics. In contrast, the remainder is finite in the regularization parameter. That is why in previous two-loop calculations [30, 83, 88] performed with the help of Caravel only the remainders were reconstructed. If desired, one can then obtain the amplitude using the pole operators as described in section 5.2.

However, there are so far no publicly available analytic two-loop five-point amplitudes to all orders. Thus, we can only estimate the difference in complexity between remainder and amplitude. We want to directly compare the sizes of the expressions and thus their evaluation time. Thus, besides the finite remainders, we have also determined the two-loop amplitudes to all orders in the regularization parameter ϵ. In addition to this quantitative difference in performance between amplitudes and remainders, the two-loop amplitudes up to order ϵ^2 can be used to determine the pole structure of three-loop amplitudes.

Excluding the propagator structure Γ from the following discussion, according to equation (3.44), the two loop partial helicity amplitudes $A_h^{(2,0)}$ and $A_h^{(2,N_f)}$ from equation (5.9), all have the form

$$A_h^{(2,j)} = \sum\nolimits_i c_i(\epsilon, \vec{s}, \mathrm{tr}_5)\mathcal{I}_i(\epsilon, \vec{s}, \mathrm{tr}_5)\,, \tag{5.21}$$

with $\vec{s} = \{s_{12}, s_{23}, s_{34}, s_{45}, s_{15}\}$ and tr_5 as in equations (5.1) and (5.2). We use a so-called pure basis of integrals [124], that is a basis consisting of integrals where all residues of an integral are the same up to a sign. Therefore, the structure of the coefficients c_i as functions in ϵ does not depend on the phase space point. Thus we can determine the highest power κ_i of the numerator of c_i as a polynomial in ϵ, as well as its denominator which turns out not to depend on kinematics, during a warm-up run. At this stage, these quantities are reconstructed using Thiele's formula, from some evaluations of the amplitude in finite fields. Thus the coefficients can be written in the form

$$c_i(\epsilon, \vec{s}, \mathrm{tr}_5) = \frac{1}{P_i(\epsilon)} \sum_{k=0}^{\kappa_i} \epsilon^k\, c_{i,k}(\vec{s}, \mathrm{tr}_5)\,, \tag{5.22}$$

and κ_i and $P_i(\epsilon)$ are known after the warm-up evaluation. As stated in equation (3.56), $(\mathrm{tr}_5)^2$ is the Gram determinant and thus polynomial in the Mandelstam invariants. That is, the coefficients $c_{i,k}(\vec{s}, \mathrm{tr}_5)$ are at most linear in tr_5, so that

we can write

$$c_{i,k}(\vec{s}, \mathrm{tr}_5) = c^{+}_{i,k}(\vec{s}) + \mathrm{tr}_5\, c^{-}_{i,k}(\vec{s})\,, \tag{5.23}$$

as already was stated in section 3.9. Now since tr_5 is not rational, $c_{i,k}(\vec{s}, \mathrm{tr}_5)$ is also not rational and we cannot reconstruct it employing finite fields. But, we can use the parity-conjugated phase space point with $-\mathrm{tr}_5$, such that for $c^{\pm}_{i,k}(\vec{s})$ one arrives at

$$\begin{aligned} c^{+}_{i,k}(\vec{s}) &= \frac{1}{2}\Big(c_{i,k}\left(\vec{s}, \mathrm{tr}_5\right) + c_{i,k}\left(\vec{s}, -\mathrm{tr}_5\right)\Big), \\ c^{-}_{i,k}(\vec{s}) &= \frac{1}{2\,\mathrm{tr}_5}\Big(c_{i,k}\big(\vec{s}, \mathrm{tr}_5\big) - c_{i,k}\big(\vec{s}, -\mathrm{tr}_5\big)\Big), \end{aligned} \tag{5.24}$$

where we notice that the left-hand side is a function of $\vec{s}$ only. The task to determine the analytic form of the amplitude to all orders in the regularization parameter becomes then equivalent to the reconstruction of the coefficients $c^{\pm}_{i,k}(\vec{s})$ in the five independent Mandelstam invariants s_{ij}, where we set $s_{12} = 1$ for the reconstruction, since we can impose the dependence of the analytic results on s_{12} through dimensional analysis. Note that to determine one value for one coefficient $c^{\pm}_{i,k}(\vec{s})$ one needs to evaluate the amplitude two times, one with positive tr_5 and one with negative tr_5.

In order to obtain the analytic form of the coefficients $c^{\pm}_{i,k}$, we use multivariate functional reconstruction, as it was described in section 3.9. The complexity of the amplitude as a master integral decomposition is then characterized by the number of independent coefficients $c^{\pm}_{i,k}(\vec{s})$ and their maximal degree as a polynomial in the invariants. Recall from section 3.9 that the denominators of the coefficients can be obtained from a reconstruction in one variable. They resemble the physical poles of the amplitude and the degenerate points of master integrals, i.e., points where linear combinations of master integrals with rational coefficients in the invariants become zero. Thus, they are simple in comparison to the numerators. We show the data characterizing the complexity of the amplitudes $A^{(2,j)}_h$ in table 5.5. On the one hand, the maximal degrees are similar to those of the remainders of the five-parton amplitudes [30]. So, approximately the same number of points and thus evaluations are needed for the reconstruction. The time for the reconstruction is thus of the same order as for five parton remainders. On the other hand, since photons are not color-ordered in contrast to gluons, their permutations appear in every amplitude so that more integrals are needed to parametrize them. Also, as just explained, amplitudes are expected to be more complex than remainders. This is seen in the fact that the number of independent coefficients is much higher for the photon amplitudes as seen in the last column of table 5.5 when compared to the respective number of coefficients in reference [30], which was at most 95.

Helicity	Max degree	# independent $c_{i,k}^{\pm}$
$A_{-++}^{(2,0)}$	32	1320
$A_{-++}^{(2,N_f)}$	20	203
$A_{+++}^{(2,0)}$	27	1244
$A_{+++}^{(2,N_f)}$	18	130

Table 5.5.: Information about the complexity of the coefficients $c_{i,k}^{\pm}$ for the relevant amplitudes in equation (5.9). Max degree means the polynomial degree of the most complex coefficient $c_{i,k}^{\pm}$ and # independent $c_{i,k}^{\pm}$ means how many coefficients are needed to parametrize the amplitude.

Helicity	Max degree	# independent $r_i^{\pm}$	Max weight
$R_{-++}^{(2,0)}$	30	171	4
$R_{-++}^{(2,N_f)}$	13	57	3
$R_{+++}^{(2,0)}$	16	62	2
$R_{+++}^{(2,N_f)}$	12	12	1

Table 5.6.: Information about the complexity of the finite remainder coefficients in equation (5.19). Max degree means the polynomial degree of the most complex coefficient $r_i^{\pm}$ and # independent $r_i^{\pm}$ means how many coefficients are needed to parametrize the remainder. Max weight means the highest transcendental weight appearing in the integral function decomposition in equation (5.19) as described in section 3.5.

5.6. Analytic Finite Remainders

After we have determined the partial helicity amplitudes $A_h^{(2,j)}$ from equation (5.9) by reconstruction and simplified them by multivariate partial fractioning as described in section 3.9, we have obtained the remainders using the equations from section 5.2. In analogy to equation (5.23) for the master integral coefficients we wrote the pentagon function coefficients r_i from equation (5.19) as

$$r_i(\vec{s}, \mathrm{tr}_5) = r_i^+(\vec{s}) + \mathrm{tr}_5\, r_i^-(\vec{s})\,, \tag{5.25}$$

such that we could compare to the data from table 5.5. We present the data characterizing the complexity of the remainders in table 5.6. Here, we show the same characterizing properties of the remainders as we did for the amplitudes in table 5.5, and add information about the highest transcendental weight of the

appearing pentagon functions. As explained in section 3.5, the transcendental weight describes how many times one needs to integrate an algebraic function to get the respective function. It thus gives a measure for the evaluation time of the function. That is, we see that not only the coefficients r_i of the remainder $R^{(2,0)}_{-++}$ are more complex than the other two-loop remainders, but also the respective integral functions h_i. There are 3518 basis functions in total, where 277 have weight four [36]. We also see that fewer coefficients are needed to parametrize a remainder than to parametrize an amplitude. Thus, phenomenological studies can benefit from the simplicity of remainders. For the reconstruction, however, the difference in complexity is negligible because the overall maximal degrees of the polynomials $r_i^{\pm}$ and $c_{i,k}^{\pm}$ are similar. There is another fact that we observed, which is not shown in the table. Although the denominators of the remainders are like the ones of the coefficients determined by reconstruction in one variable and matching against the non-planar alphabet, there is no pole at tr_5^2 appearing in the remainders, whereas it appears in coefficient functions. That means there is more chance to get numerical instabilities during phase space integration when one decides to use amplitudes compared to remainders in our case.

5.7. Validation

We validated our results in multiple ways. A consistency check of both the finite remainders and the two-loop amplitudes, in general, is to look at their poles, which were described in sections 3.8 and 5.2. We verified that the remainders (5.10) when obtained from the amplitudes using equation (5.11) and following indeed were finite, which is done explicitly for instance in the example program `finite_remainder_2l`, as described in section 4.6. Because, as seen in equation (5.10), the remainder is obtained from the two-loop amplitudes, this also provides a consistency check for the two-loop amplitudes.

After reconstructing the coefficients of the amplitudes and remainders, we compared the results to purely numerical calculations. This checks that the reconstruction worked as expected, which means, for example, that the denominators of the coefficients can indeed be parametrized by the alphabet of the integral basis as seen in equation (3.67).

In contrast to the five-parton calculation [30], we have used the integral functions of reference [36] for the remainder calculation as described in section 4.6.3, and extended Caravel with them. We reproduced the numeric five-parton results from reference [99] in the physical region to check that we added them correctly.

In table 5.1 we describe relations, which allow obtaining all relevant helicity amplitudes for two-loop three-photon production from only one amplitude. As a consistency check, we calculated every helicity configuration numerically. We

then compared the results to the values obtained from permuting external momenta and parity conjugation as stated in table 5.1 finding complete agreement.

To calculate the two-loop remainders, we used one-loop amplitudes calculated by CARAVEL, which have been implemented together with the two-loop amplitudes and thus not checked before. Also, as described in section 4.1, the coefficients of two-loop five-point amplitudes are determined with a **Coefficient provider** of a different type than the one used for one-loop amplitudes. We have checked the one-loop coefficients for correctness by reproducing results by `OpenLoops` [125] up to finite order in the regularization parameter. Furthermore, we have used the fact that at one-loop, when replacing the weak coupling constant by the strong coupling constant and summing amplitudes with three permuted gluons instead of three photons, contributions stemming from gluon self-interactions exactly cancel. We checked explicitly that this procedure reproduces the one-loop amplitudes for three-photon production.

While the amplitude coefficients we have determined are new to the public, the two-photon amplitudes are available for a few years [126]. In addition to the three-photon amplitude coefficients, we have reconstructed the one- and two-loop two-photon coefficients with CARAVEL and compared them to the results of reference [126] finding agreement on the provided remainders. We have also used the two-photon amplitudes to check the behavior of our three-photon amplitudes in the limit of two particles going collinear, as described in the following section.

Finally, when publishing, we had the advantage that the amplitude, although not made publicly available, was already obtained in reference [32]. There are no results in [32] we could check against directly. The phenomenological study in reference [37] however, which has used our results was consistent with the outcome of reference [32]. This agreement indirectly checks our implementation. Furthermore, the authors of [32] checked that their implementation of the amplitude agrees with ours later on in reference [38].

5.8. Collinear Limits

We obtained a vital check of our three-photon amplitudes by taking collinear limits since these can be calculated from known lower multiplicity results [126] using universal functions called splitting functions [127, 128]. If two of the external momenta become collinear, say p_i and p_j, that means their spacial components are proportional, so $\mathbf{p}_i = x\mathbf{p}_j$. We denote this limit by $i||j$. Because the particles are massless, we then have $(p_i^0)^2 = \mathbf{p}_i^2 = x^2\mathbf{p}_j^2 = x^2(p_j^0)^2$ so that p_i and p_j are proportional as four-momenta, that is $p_i = xp_j$. The total momentum of the collinear legs i and j is given by $p_P := p_i + p_j$, where for the later calculation we define z as the energy proportion of the leg i with respect to p_P so that in the

collinear limit

$$p_i \to z p_P \qquad \text{and} \qquad p_j \to (1-z) p_P \,. \tag{5.26}$$

If $z = 0$, then the whole energy of the collinear particles is carried by particle j, so that particle i is soft and vice versa if $z = 1$.

To relate our five-point amplitudes to lower multiplicity results we use the kinematic part of the amplitude $\mathcal{M}$ as in equation (5.5), which we call the color-stripped amplitude $\bar{\mathcal{A}}$ consisting of the partial amplitude $\mathcal{A}$ including the spinor weight normalization

$$\bar{\mathcal{A}}(1_q^{h_1}, 2_{\bar{q}}^{h_2}, 3_\gamma^{h_3}, 4_\gamma^{h_4}, 5_\gamma^{h_5}) \equiv \Phi(1_q^{h_1}, 2_{\bar{q}}^{h_2}, 3_\gamma^{h_3}, 4_\gamma^{h_4}, 5_\gamma^{h_5}) \mathcal{A}(1_q^{h_1}, 2_{\bar{q}}^{h_2}, 3_\gamma^{h_3}, 4_\gamma^{h_4}, 5_\gamma^{h_5}) \,. \tag{5.27}$$

In the collinear limit, the legs i^{h_i} and j^{h_j} behave like a single leg P^h, where in addition, collinear divergences can occur. These can be represented by so-called splitting functions $\mathrm{Split}_{P^h}(i^{h_i}, j^{h_j})$ [127, 128], which are three-point helicity amplitudes with pairwise proportional external legs. In the limit $i||j$ the color-stripped amplitude $\bar{\mathcal{A}}$ factorizes and can be written as

$$\bar{\mathcal{A}}(i^{h_i}, j^{h_j}, \dots) \xrightarrow{i||j} \sum\nolimits_h \mathrm{Split}_{P^h}(i^{h_i}, j^{h_j}) \bar{\mathcal{A}}(\bar{P}^{-h}, \dots) \,, \tag{5.28}$$

where if $\bar{\mathcal{A}}(i^{h_i}, j^{h_j}, \dots)$ is an n-point amplitude, $\bar{\mathcal{A}}(\bar{P}^{-h}, \dots)$ is an $(n-1)$-point amplitude. Since $\mathrm{Split}_{P^h}(i^{h_i}, j^{h_j})$ are also amplitudes we can expand them in the bare strong coupling constant α_s^0

$$\begin{aligned} \mathrm{Split}_{P^h}(i^{h_i}, j^{h_j}) = \mathrm{Split}^{(0)}_{P^h}(i^{h_i}, j^{h_j}) + \frac{\alpha_s^0}{2\pi} \mathrm{Split}^{(1)}_{P^h}(i^{h_i}, j^{h_j}) + \\ + \left(\frac{\alpha_s^0}{2\pi}\right)^2 \mathrm{Split}^{(2)}_{P^h}(i^{h_i}, j^{h_j}) + \mathcal{O}\left((\alpha_s^0)^3\right) . \end{aligned} \tag{5.29}$$

Expanding the splitting functions and the color-stripped amplitudes in equation (5.28) and comparing coefficients we can obtain factorization formulas for the collinear one-loop amplitude $\bar{\mathcal{A}}^{(1)}$

$$\begin{aligned} \bar{\mathcal{A}}^{(1)}(i^{h_i}, j^{h_j}, \dots) \xrightarrow{i||j} \sum\nolimits_h \Big(& \mathrm{Split}^{(1)}_{P^h}(i^{h_i}, j^{h_j}) \bar{\mathcal{A}}^{(0)}(\overline{P}^{-h}, \dots) \\ & + \mathrm{Split}^{(0)}_{P^h}(i^{h_i}, j^{h_j}) \bar{\mathcal{A}}^{(1)}(\overline{P}^{-h}, \dots) \Big) \,, \end{aligned} \tag{5.30}$$

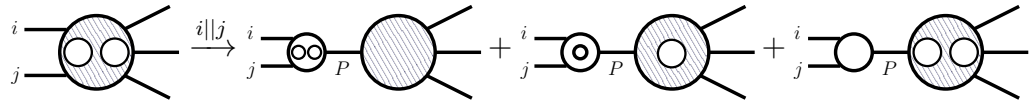


Figure 5.4.: Graphic representation for equation (5.31). The white circles on the left-hand side are represent the different orders of the splitting functions $\mathrm{Split}^{(l)}_{P^h}(i^{h_i}, j^{h_j})$ in the bare strong coupling constant. The gray circles are the different orders of the color-stripped amplitudes $\bar{\mathcal{A}}$. The circles with zero, one or two circles inside represent tree, one- and two-loop contributions respectively.

and for the collinear two-loop amplitude $\bar{\mathcal{A}}^{(2)}$

$$\begin{aligned}\bar{\mathcal{A}}^{(2)}(i^{h_i}, j^{h_j}, \ldots) \xrightarrow{i||j} \sum\nolimits_h \Big(&\mathrm{Split}^{(2)}_{P^h}(i^{h_i}, j^{h_j}) \bar{\mathcal{A}}^{(0)}(\overline{P}^{-h}, \ldots) \\ &+ \mathrm{Split}^{(1)}_{P^h}(i^{h_i}, j^{h_j}) \bar{\mathcal{A}}^{(1)}(\overline{P}^{-h}, \ldots) \\ &+ \mathrm{Split}^{(0)}_{P^h}(i^{h_i}, j^{h_j}) \bar{\mathcal{A}}^{(2)}(\overline{P}^{-h}, \ldots) \Big) \,. \end{aligned} \tag{5.31}$$

Before checking the two-loop behavior, we have performed the same checks on $\bar{\mathcal{A}}^{(1)}$ in the collinear limit, verifying the consistency of our one-loop amplitudes and splitting functions $\mathrm{Split}^{(1)}_{P^h}(i^{h_i}, j^{h_j})$. Similar to cut diagrams, the factorization in equation (5.31) can be represented graphically, as seen in figure 5.4. We note that the color-stripped amplitude $\bar{\mathcal{A}}$ does not have any collinear divergences, if the right-hand side of equation (5.31) vanishes. The l-loop splitting functions $\mathrm{Split}^{(l)}_{P^h}(i^{h_i}, j^{h_j})$ can be written by means of the process dependent tree vertex $\mathrm{Split}^{(0)}_{P^h}(i^{h_i}, j^{h_j})$ and transcendental functions $\rho^{(l)}_{P^h}(i^{h_i}, j^{h_j})$ in s_{ij} and z [128, 129]

$$\mathrm{Split}^{(l)}_{P^h}(i^{h_i}, j^{h_j}) = \rho^{(l)}_{P^h}(i^{h_i}, j^{h_j}) \mathrm{Split}^{(0)}_{P^h}(i^{h_i}, j^{h_j}) \,, \tag{5.32}$$

which also depend on the regularization parameter ϵ and the state dimension of the internal particles D_s. Looking at equations (5.31) and (5.32) we can often find out that an amplitude is regular in the collinear limit without knowing the transcendental part $\rho^{(l)}_{P^h}(i^{h_i}, j^{h_j})$.

Photons do not carry color and thus appear unordered in the color-stripped amplitudes. Because of that, the limits where different photons with the same helicity go collinear are equivalent. In addition, if amplitudes involving a quark vanish, the same holds for respective amplitudes with an antiquark. Thus, in order to check the collinear behavior according to equation (5.31) we have to consider the three limits (γ, γ), $(q, \bar{q})$ and (q, γ) (or $(\bar{q}, \gamma)$), for every helicity pairing (h_i, h_j). We discuss these three limits one by one in the following.

$(q, \bar{q})$ limit: In this limit $s_{12} = (p_1 + p_2)^2 = p_P^2 = 0$, which does not lie in the physical region (5.3) of three-photon production, so it cannot be checked.

(γ, γ) limit: Since there is no tree vertex $\text{Split}^{(0)}_{P^h}(i_\gamma^{h_i}, j_\gamma^{h_j})$, all splitting functions $\text{Split}^{(l)}_{P_h}(i_\gamma^{h_i}, j_\gamma^{h_j})$ in equation (5.32) and thus the right-hand side of equation (5.31) vanish. That is, our computed amplitudes should be regular in the (γ, γ) limit, which we have checked numerically.

(q, γ) and $(\bar{q}, \gamma)$ limit: In that case it is simpler to check $\bar{\mathcal{A}}_{+++}(i_q^{h_i}, j_\gamma^+)$ and $\bar{\mathcal{A}}_{-++}(i_q^{h_i}, j_\gamma^-)$ first, because the corresponding color-stripped tree-level amplitude on the right-hand side of equation (5.31) is $\bar{\mathcal{A}}^{(0)}_{++} = 0$. That is, for these cases we don't have to calculate $\text{Split}^{(2)}_{\bar{q}^h}(i_q^{h_i}, j_\gamma^{h_j})$ to obtain the right-hand side of equation (5.31). To obtain $\text{Split}^{(1)}_{\bar{q}^h}(i_q^{h_i}, j_\gamma^{h_j})$, we employ the results for $\text{Split}^{(1)}_{\bar{q}^h}(i_q^{h_i}, j_g^{h_j})$ from reference [130], where appropriate color factors have to be used to account for the switch from gluons to photons. The relevant transcendental functions $\rho^{(1)}_{\bar{q}^\mp}(i_q^\pm, j_\gamma^{h_j})$ from equation (5.32) are then given by

$$\rho^{(1)}_{\bar{q}^+}(i_q^-, j_\gamma^-) = \rho^{(1)}_{\bar{q}^-}(i_q^+, j_\gamma^+) = F(z, s_{ij}, \epsilon) - z\, r_\Gamma(\epsilon) \left(\frac{-s_{ij}}{\mu^2}\right)^{-\epsilon} \frac{D_s - 2}{4(1-\epsilon)(1-2\epsilon)}\,, \tag{5.33}$$

for splittings into particles with the same helicity and by

$$\rho^{(1)}_{\bar{q}^+}(i_q^-, j_\gamma^+) = \rho^{(1)}_{\bar{q}^-}(i_q^+, j_\gamma^-) = F(z, s_{ij}, \epsilon)\,, \tag{5.34}$$

for splittings into particles with different helicities. Here, $F(z, s_{ij}, \epsilon)$ is defined as

$$F(z, s_{ij}, \epsilon) = r_\Gamma(\epsilon) \left(\frac{-s_{ij}}{\mu^2}\right)^{-\epsilon} \frac{1}{\epsilon^2} \sum_{m=1}^{\infty} \epsilon^m \text{Li}_m\left(\frac{-z}{1-z}\right)\,, \tag{5.35}$$

and the polylogarithms Li_m as [131]

$$\begin{aligned} \text{Li}_1(x) &= -\log(1-x)\,, \\ \text{Li}_m(x) &= \int_0^x \frac{\text{dt}}{t} \text{Li}_{m-1}(t)\,. \end{aligned}$$

Finally, the normalization factor is

$$r_\Gamma(\epsilon) = \frac{1}{(4\pi)^\epsilon} \frac{\Gamma(1-\epsilon)^2 \Gamma(1+\epsilon)}{\Gamma(1-2\epsilon)}\,. \tag{5.36}$$

We have obtained the tree-level vertices $\text{Split}^{(0)}_{\bar{q}^\mp}(i_q^\pm, j_\gamma^{h_j})$, which are needed to calculate $\text{Split}^{(1)}_{\bar{q}^\mp}(i_q^\pm, j_\gamma^{h_j})$ by means of equation (5.32), from the corresponding Feynman rule as given in section 2.3. With the above constraints we could verify numerically that equation (5.31) holds in the (q, γ) and $(\bar{q}, \gamma)$ limit.

The last type of check we had to perform was the collinear limit of $\bar{\mathcal{A}}_{-++}(i_q^{h_i}, j_\gamma^+)$, where the color-stripped tree-level amplitude on the right-hand side of equation (5.31) is $\bar{\mathcal{A}}^{(0)}_{-+}$ and does not vanish so that the two-loop splitting function $\text{Split}^{(2)}_{\bar{q}^\mp}(i_q^\pm, j_\gamma^+)$ has to be taken into account. We did not evaluate $\text{Split}^{(2)}_{\bar{q}^\mp}(i_q^\pm, j_\gamma^+)$ explicitly. Instead, we used the fact that if we normalize it to the corresponding tree-level splitting function according to equation (5.32), it should depend only on the Mandelstam invariant s_{ij} and the energy fraction z. It does not depend on other Mandelstam invariants in this case. We then solved (5.31) first for the one- and then for the two-loop splitting functions normalized by the tree-level value. Evaluating them on multiple phase space points, keeping z and s_{ij} constant and changing the remaining Mandelstam invariants, we could verify that the splitting functions remained unchanged as expected.

5.9. Ancillary Files

We provide our results for the one- and two-loop amplitudes $A_h^{(1)}$, $A_h^{(2,0)}$, and $A_h^{(2,N_f)}$ as defined in equations (5.8) and (5.9) as well as the finite remainders $R_h^{(1)}$, $R_h^{(2,0)}$ and $R_h^{(2,N_f)}$ as in equation (5.16) in headed list ancillary files coming with reference [84]. The one- and two-loop amplitudes as well as the remainders are in the subdirectories `oneLoopAmplitudes`, `twoLoopAmplitudes` and `remainders` respectively. The strings `pmppp` and `pmmpp` in the filenames denote the helicities of the external particles of the respective amplitude, that is whether $h = +++$ or $h = -++$. The strings `nf0` and `nf1` for the two-loop amplitudes or the optional string `nf` for the remainders are there to distinguish between the amplitudes $A_h^{(2,0)}$ and remainders $R_h^{(2,0)}$ without closed fermion loops and the amplitudes $A_h^{(2,N_f)}$ and remainders $R_h^{(2,N_f)}$ with closed fermion loops. The one-loop amplitudes are given as a two-dimensional list $(\mathcal{I}_i, c_i(\vec{s}, D_s))_i$, with the master integrals needed to represent the amplitude in the first entry and their coefficients in the second entry, so that the one-loop amplitudes can be obtained by summing

$$A_h^{(1)} = \sum\nolimits_i c_i(\vec{s}, D_s)\mathcal{I}_i \,. \tag{5.37}$$

The coefficients are functions of Mandelstam invariants and $D_s = D = 4-2\epsilon$. The remaining amplitudes and remainders are given using tags f_j for often occurring polynomials to save space.

As for the remainders, every file consists of a three-dimensional list with the

form $((\text{PPair}[a_i, b_i], F_i)_i, (f_j)_j)$ where a_i and b_i are dimensionless functions of the polynomials f_j. The F_i are integral functions in the notation of the package `PentagonMI` [36]. They can be evaluated using `PentagonMI`, or using reference values for the evaluation on the point from equation (5.20) in the file `numerics/Fvalues.m` of reference [84]. The remainders can then be obtained from the lists in the `remainders` directory by calculating

$$R = \sum\nolimits_i \Big(a_i\big(\vec{f}(\vec{s})\big) + \frac{\text{tr}_5}{s_{12}^2} b_i\big(\vec{f}(\vec{s})\big) \Big) F_i \,. \tag{5.38}$$

The two-loop amplitudes follow a similar pattern. Their coefficients are saved as a two-dimensional list $((\text{PPair}[a_i, b_i])_i, (f_j)_j)$. The master integrals $\mathcal{I}_i$ are identical for all helicity configurations of one specific amplitude and therefore saved in the files labeled `IntegralsMI_nf0.m` and `IntegralsMI_nf1.m` once per helicity configuration. In analogy to equation (5.38), the two-loop amplitudes can be assembled from those lists in the respective coefficient files by calculating

$$A^{(2,j)} = \sum\nolimits_i \Big(a_i\big(\vec{f}(\vec{s}), \epsilon\big) + \frac{\text{tr}_5}{s_{12}^2} b_i\big(\vec{f}(\vec{s}), \epsilon\big) \Big) \mathcal{I}_i \,. \tag{5.39}$$

This procedure and a test, whether the formulas from section 5.2 indeed deliver the remainders R from the amplitudes A is saved in the file `example_assembly.m`.

The integral strings which can be found in the one-loop files `IntegralsMI_nf0.m` and `IntegralsMI_nf1.m` follow a similar pattern to the integral strings Caravel uses as output as described in the reconstruction examples in section 4.6.2. Now they have the form

```
MI[PropagatorStructure, i, mandelstam1, mandelstam2, ...]
```

where as in section 4.6, `PropagatorStructure` is a name describing the graph representation of the integral, such as `TwoMassTriangle` or `PentagonBox`. Instead of the `NumeratorLabel`, as described in section 4.6.2 for Caravel output, now the master integrals are indexed with `i`. The numerators for each master integral can be found in the files `MasterIntegralDefinitions.m`. In analogy to table 4.2, the numerator insertions for three-photon production are given in table 5.7, where

$$\mu_{ij} = \ell_i^{D-4} \cdot \ell_j^{D-4} \,. \tag{5.40}$$

In contrast to the Caravel examples in section 4.6, the `MI[]` strings employed in the photon files don't provide information about the propagators of the integral. We list these in appendix D.

`l[i]`	ℓ_i
`p[k]`	p_k
`sq[v]`	v^2
`sp[v[i], v[j]]`	$v_i \cdot v_j$
`tr5[1, 2, 3, 4]`	tr_5
`mu[i, j]`	μ_{ij}
`D`	D

Table 5.7.: Numerator insertion naming of the master integrals in the ancillary files to three-photon production amplitudes.

5.10. Hard Functions

We have shown the analytic structure of one- and two-loop amplitudes as well as two-loop remainders relevant for three-photon production at hadron colliders. However, as has been shown [132], the double-virtual NNLO contributions to the differential cross section can be determined from tree amplitudes and finite remainders only, which lack the redundant information stemming from lower-loop contributions. By summation over possible helicity states and color these remainders and tree amplitudes can be assembled into so-called hard functions H [133]. Like the amplitude, the hard function can be expanded in the strong coupling constant α_s

$$H = H^{(0)} + \frac{\alpha_s}{2\pi} H^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2 H^{(2)} + \mathcal{O}(\alpha_s^3)\,, \tag{5.41}$$

where we factor out the Born-level contribution so that $H^{(0)} = 1$. The NLO correction $H^{(1)}$ is then given by

$$H^{(1)} = \frac{1}{\sum_h \left|\mathcal{M}_h^{(0)}\right|^2} C_F \left(\sum_h \left|\mathcal{M}_h^{(0)}\right|^2 2\mathrm{Re}\left[R_h^{(1)}\right] \right)\,, \tag{5.42}$$

and the NNLO correction $H^{(2)}$ by

$$\begin{aligned} H^{(2)} = \frac{1}{\sum_h \left|\mathcal{M}_h^{(0)}\right|^2} \Bigg(& \sum_h \frac{N_c^2}{4} N_c \left|\Phi_h R_h^{(1)}\right|^2 + \\ & \sum_h \left|\mathcal{M}_h^{(0)}\right|^2 2\mathrm{Re}\left[\frac{N_c^2}{4} R_h^{(2,0)} + C_F T_F N_f R_h^{(2,N_f)}\right] \Bigg)\,. \end{aligned} \tag{5.43}$$

For $H^{(2)}$, the contribution proportional to $\left|R_h^{(1)}\right|^2$ is the convolution of the two one-loop amplitudes as described in section 2.5 and the one proportional to the remainder $R_h^{(2,0)}$ and $R_h^{(2,N_f)}$ defined in equation (5.16) is the convolution of a tree with a two-loop amplitude. Taking the normalizations of the remainders into account, the tree matrix element $\mathcal{M}_h^{(0)}$ is set to

$$\left|\mathcal{M}_h^{(0)}\right|^2 = e_q^6 N_c |\Phi_h|^2 \left|A_h^{(0)}\right|^2 . \tag{5.44}$$

The calculation of the hard functions $H^{(1)}$ and $H^{(2)}$ is facilitated by the permutation relations in table 5.1. Also, the respective two-loop remainders don't have to be evaluated, because the tree amplitudes $\mathcal{M}_{+++}^{(0)}$ vanish. This means, $H^{(1)}$ and $H^{(2)}$ can be obtained from permutation and conjugation of the analytic form of the two one-loop remainders $R_{+++}^{(1)}$ and $R_{-++}^{(1)}$ and the two two-loop remainders $R_{-++}^{(2,0)}$ and $R_{-++}^{(2,N_f)}$. The hard functions have been implemented in the C++ library `FivePointAmplitudes-cpp`, which can be obtained from the website of reference [84]. To install it, the commands of listing 5.1 can be used.

```
1 > git clone git@gitlab.com:five-point-amplitudes/\
2 > FivePointAmplitudes-cpp.git
3 > cd FivePointAmplitudes-cpp
4 > mkdir build
5 > cd build
6 > meson .. -Dprefix=<install-dir>
7 > ninja install
```

Listing 5.1: Commands to download, build and install the `FivePointAmplitudes` library

To obtain the values for the hard functions $H^{(1)}$ and $H^{(2)}$ one can use the example inside `examples/example_H2`, which calculates their values on the phase space point given in equation (5.20), where the phase space point and the renormalization scale can be set inside the example. The evaluation time is dominated by the integral function evaluation and takes about five seconds in total using double precision. As for numerical stability, $H^{(2)}$ has been evaluated in double precision to $H^{(2)}_{\text{double}}$ and in quadruple-double precision to $H^{(2)}_{\text{quad}}$ on 90000 phase space points [37]. Assuming $H^{(2)}_{\text{quad}}$ to be correct within double precision, the value

$$d = -\log_{10}\left|\frac{H^{(2)}_{\text{double}}}{H^{(2)}_{\text{quad}}} - 1\right| \tag{5.45}$$

is a measure of the correct digits of $H^{(2)}_{\text{double}}$ and shown in figure 5.5.

In the figure, one sees that the results are correct to more than four digits on the bulk of the phase space points, whereas less than one per mille has less than

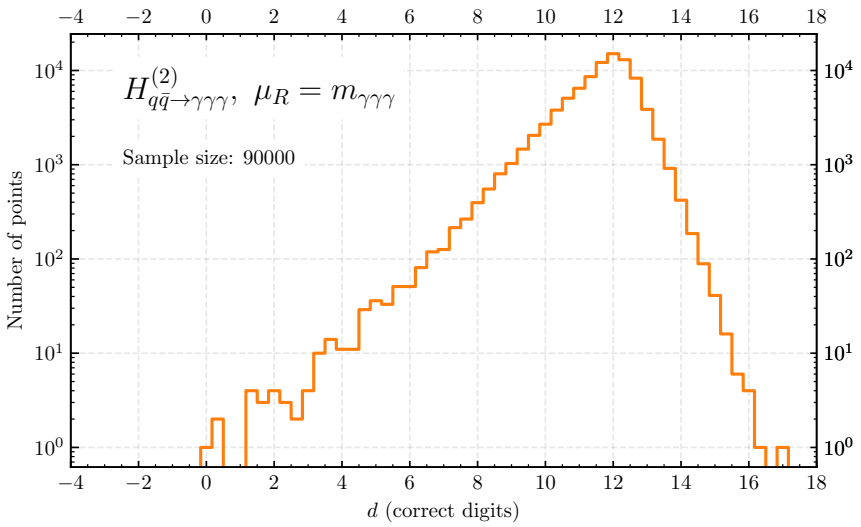


Figure 5.5.: The number of correct digits d as in equation (5.45) of the evaluation of the two-loop hard function $H^{(2)}$ in double precision [37]. The authors set the renormalization scale here to the invariant mass of the three photons.

four correct digits. The low accuracy in that regions could be traced back to the vanishing of letters of the alphabet of the chosen integral function basis, which means that one should take care that no letters vanish when choosing the phase space integration contour.

The hard functions implemented in `FivePointAmplitudes-cpp` have been employed in the Matrix framework [134] and used in the phenomenological study of reference [37]. They agree fully with the previous study of reference [32] and show good agreement with the experimental study [35] on ATLAS 8 TeV data, as for example seen in the invariant mass distribution of the three-photon system plot shown in figure 5.6. One can see that the Matrix result agrees with the data within one standard deviation, whereas the data is underestimated using NLO predictions in the lower bins. The azimuthal angles $\Delta\phi$, the pseudorapidities $\Delta\eta$, and the transverse momentum distributions behave similarly [37]. One also sees in the lower diagram that the NNLO contributions are sizable: they are up to twice as high as the NLO contributions to the invariant mass distribution and reached a factor of three compared to other observables. That means that including NNLO contributions is crucial for accuracy, for example, in a 13 TeV study of this process. Although the double-virtual contributions are relatively small, as was the case for two-photon production [135], when using q_T-subtraction [136] they can reach up to 6 % of the NNLO corrections [37] thus making them noticeable.

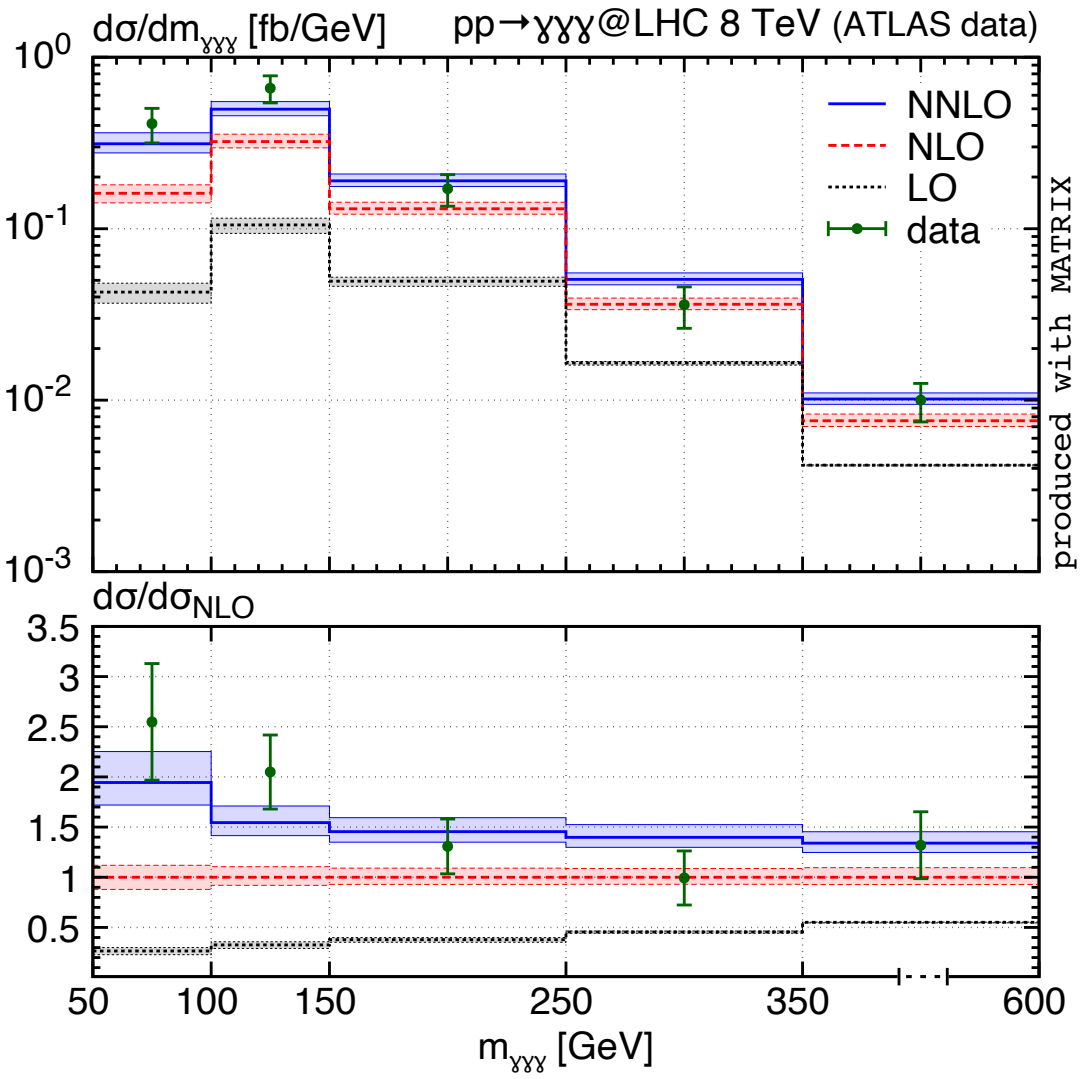


Figure 5.6.: Invariant mass distribution of the three-photon system produced by MATRIX compared to ATLAS 8 TeV data as well as the enhancement of NNLO to NLO [37]. This studies used our hard functions defined in equations (5.41), (5.42) and (5.43).

6. Conclusion

In this thesis, we have obtained the analytic form of the two-loop leading-color scattering amplitudes for three-photon production at hadron colliders for the first time. We have used the numerical unitarity method to obtain the coefficients of a decomposition of the amplitude into master integrands and surface terms. The decomposition was performed in a finite field and to all orders in the regularization parameter. We have employed the available setup to determine the finite remainder coefficients of the planar two-loop five-parton amplitudes in the Euclidean region as described in reference [30]. We have extended the setup by colorless photon fields and made it work in the physical region of phase space. After establishing a framework for the numerical evaluation of the master integral coefficients for amplitudes involving photons, we have obtained the analytic form of the amplitudes. We have observed that the denominators of the coefficients are given by products of letters of the master integral basis alphabet and have matched them accordingly. We have then reconstructed their numerators using multivariate polynomial reconstruction algorithms. The master integral expressions we provide with the analytic coefficients can be evaluated in the physical region using the pentagon functions implementation [36]. We have assembled the hard functions required to calculate NNLO QCD corrections to three-photon production at the LHC with our amplitudes.

The authors of reference [37] have used our results for their phenomenological study. They have performed systematic comparisons against three-photon production data from the ATLAS Collaboration [35].

We have calculated the master integral and finite remainder coefficients numerically and functionally reconstruct them using the Caravel framework, which we extended to meet this task. We have presented the public version of Caravel in reference [31]. Caravel is a series of C++ libraries that implement the multi-loop numerical unitarity method. It is written in a modular way allowing to extend it by different QFT models. Provided with a decomposition of the relevant integrands into master integrals and surface terms, Caravel can determine the master integral coefficients either in floating-point precision or in a finite field. Applying rational reconstruction from multiple finite field evaluations Caravel can determine exact rational values for the coefficients. It represents four-point results via multiple polylogarithms [77] and five-point results in terms of pentagon functions [25, 36]. Caravel can extract the analytic form of these coefficients by using functional reconstruction algorithms [94]. We

have explained how the modules of Caravel work, and we have described how to use the consistency tests and examples provided with the software. We particularly emphasized the changes coming with version 0.2.0 of Caravel, which allow the determination of master integral coefficients for amplitudes involving photons. In version 0.2.0, we included colorless particles in the framework, as required when studying processes with photons. Furthermore, we provided an interface to the recent pentagon function implementation in the physical region as described in reference [36].

Multiple research projects are natural extensions of the presented work. We have considered contributions from closed fermion loops within our three-photon production calculation, particularly fermion loops without external photons. These contributions are gauge-invariant and do not include any non-planar Feynman diagrams. A non-planar master integral basis is available in principle, and there are already results including non-planar contributions for similar processes [137, 138]. However, there is no public implementation of an integrand decomposition into master integrands and surface terms, considering non-planar contributions. We expect a parametrization of non-planar two-loop five-point integrands to be available soon, which we can employ to extend the present calculation by non-planar terms. A further possible extension of the presented results is related to the Caravel framework itself. While the current version of Caravel is capable of calculating all the amplitudes obtained with its help before, users cannot choose any new amplitudes. Crucial steps of the calculation like the access to master integrals, the generation of the cut hierarchy, and the decomposition of the result in terms of integral functions rely on private codes to be extended to different models. We need to check and automate these codes to make them public so that the user can obtain amplitudes for a broader range of processes. The next step would be to get full-color results for three-photon production and calculate the coefficients of other amplitudes with photons in the final state, such as the production of two jets with a photon. For some of these processes, results already have been published [137–139]. Furthermore, since the master integrals for one massive particle in the final state are already available [140], vector-boson-plus-four-parton amplitudes are within reach. Finally, it will be exciting to work towards cross-section predictions and NNLO phenomenology for multi-particle processes at the LHC.

Appendix

We present next a series of appendices with complementary information. In appendix A we give some definitions and relations within the calculus of variations, representation theory and gauge theory important for the development of a QFT. In appendix B we explain how the internal momenta which are on-shell within a certain propagator structure Γ are generated. In appendix C, we give a definition for a propagator structure with a certain momentum routing, as it is used inside Caravel. In appendix D we define the master integrals used as a basis for our one- and two-loop three-photon production results.

A. Mathematical Formulas

A functional is a function of functions

$$\mathcal{F}\colon C^2(X) \to \mathbb{R}\,. \tag{A.1}$$

The variation of the functional is then given by

$$\delta\mathcal{F}[u,\phi] = \frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{F}[u+\epsilon\phi]\bigg|_{\epsilon=0}\,, \tag{A.2}$$

for an arbitrary test function ϕ. The functional derivative for $\mathcal{F}$ with respect to a function u is a function

$$f(x) = \frac{\delta\mathcal{F}[u]}{\delta u(x)}\,, \tag{A.3}$$

such that

$$\int f(x)\phi(x)\mathrm{d}x = \delta\mathcal{F}[u,\phi] \tag{A.4}$$

for all test functions ϕ. A necessary condition for a minimizer u of $\mathcal{F}$ is the variational principle

$$\delta\mathcal{F}[u,\phi] = 0\,, \tag{A.5}$$

for all test functions ϕ on X. For variational integrals

$$\mathcal{F} = \int_X \mathcal{L}(x,u,Du)\mathrm{d}x \tag{A.6}$$

from the variational principle we can derive the Euler–Lagrange equations

$$D\mathcal{L}_{Du^i}(x,u,Du) - \mathcal{L}_{u^i}(x,u,Du) = 0$$

for all i. We call the integrand $\mathcal{L}$ the Lagrangian here. In a field theory, we set $\mathcal{F} = S$ the action of the system. We write $u = \varphi$ for the fields. The equations of motion which tell us how the fields obeying the theory will behave are given by the corresponding Euler–Lagrange equations.

A Lagrangian is invariant under a group G action iff for all $g \in G$ we have $\mathcal{L}(g\varphi) = \mathcal{L}(\varphi)$. A group has a representation—a vector space V together with a map $\cdot\colon G \times V \to V$, $v \mapsto g \cdot v$, so that $e \cdot v = v$ and for all g in G it holds that $g_1(g_2 \cdot v) = (g_1 g_2) \cdot v$. Matrix Lie groups, so subgroups of the group of invertible $n \times n$ matrices $\mathrm{GL}(n, \mathbb{C})$, have a fundamental representation with $V = \mathbb{C}^n$ and $\cdot$ given by the multiplication of the respective matrices with v in V. A matrix Lie group also has a Lie algebra, $\mathfrak{g}$ which is the vector space of all matrices $T^{(\Phi)}$ so that for all real t the exponential $e^{\mathrm{i}tX}$ is an element of the Lie group in the representation Φ. The elements of its basis $T_a^{(\Phi)}$ are called the generators of the matrix Lie group. The Lie algebra also has a map $[\cdot,\cdot]\colon \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ given by the commutator $(A, B) \mapsto AB - BA$. For every Lie algebra there are defining structure constants ${f_{ab}}^c$ so that $[T_a^{(\Phi)}, T_b^{(\Phi)}] = \mathrm{i}{f_{ab}}^c T_c^{(\Phi)}$. With these, every matrix Lie group has an additional representation—the adjoint representation with $(T_a^{(\mathrm{adj})})_b{}^c = -\mathrm{i}{f_{ab}}^c$.

We can define a fiber bundle P with the base manifold being an oriented manifold X with Minkowski signature and the fibers being matrix Lie groups, that is P is locally homeomorphic to $U \times G$ with $G \in \mathrm{GL}(n, \mathbb{C})$ and $U \subset \mathbb{R}^{1,3}$. These fiber bundles are called principal bundles and we refer to their structure groups as gauge groups. We can also define a spinor bundle over the Minkowski space. For that, we first define a Clifford algebra [141]

$$\begin{aligned} \mathrm{Cl}_{1,3}(\mathbb{R}) &= T(\mathbb{R}^{1,3}) / \langle v^\mu v^\nu - \eta_{\mu\nu} v^\mu v^\nu \rangle \\ &= \mathrm{Cl}^0_{1,3}(\mathbb{R}) \oplus \mathrm{Cl}^1_{1,3}(\mathbb{R})\,, \end{aligned} \tag{A.7}$$

with $T(\mathbb{R}^{1,3})$ being the tensor algebra of $\mathbb{R}^{1,3}$ which induces a finite grading on $\mathrm{Cl}_{1,3}(\mathbb{R})$. Note that in that case

$$\begin{aligned} (v+w)^\mu (v+w)^\nu &= v^\mu v^\nu + v^\mu w^\nu + w^\mu v^\nu + w^\mu w^\nu \\ &= \eta_{\mu\nu} v^\mu v^\nu + 2\eta_{\mu\nu} v^\mu w^\nu + \eta_{\mu\nu} w^\mu w^\nu\,, \end{aligned} \tag{A.8}$$

which by using again the quotient from equation (A.7) means

$$v^\mu w^\nu + w^\mu v^\nu = 2\eta_{\mu\nu} v^\mu w^\nu\,, \tag{A.9}$$

which is the defining property of gamma matrices, that is they can be used as

generators for the Clifford algebra. Consider as another example $\mathrm{Cl}_{0,3}(\mathbb{R})$, which for orthonormal basis vectors e_i is generated by 1, e_1, e_2, e_3 and the combinations e_1e_2, e_1e_3 and e_2e_3 and $e_1e_2e_3$. Since it holds that

$$\left(\frac{1 \pm e_1e_2e_3}{2}\right)^2 = 1\,, \quad \left(\frac{e_1e_2 \mp e_3}{2}\right)^2 = \left(\frac{e_1e_3 \mp e_2}{2}\right)^2 = \left(\frac{e_2e_3 \mp e_1}{2}\right)^2 = -1\,, \tag{A.10}$$

we can identify $\mathrm{Cl}_{0,3}(\mathbb{R}) \cong \mathbb{H} \oplus \mathbb{H}$ with $\mathrm{Cl}^0_{0,3}(\mathbb{R}) \cong \mathbb{H}$. Consider now the map

$$\begin{aligned} \mathrm{Cl}_{0,3}(\mathbb{R}) &\to \mathrm{Cl}^0_{1,3}(\mathbb{R}) \\ v_0 + v_1 &\mapsto v_0 + v_1 e_0\,. \end{aligned} \tag{A.11}$$

Since in the second term all odd basis elements are mapped onto even ones, which are not there in $\mathrm{Cl}^0_{0,3}(\mathbb{R})$, the map is an isomorphism, in particular we can identify $\mathrm{Cl}^0_{1,3}(\mathbb{R}) \cong \mathbb{H} \oplus \mathbb{H}$. $\mathrm{Cl}_{1,3}(\mathbb{R}) \otimes \mathbb{C}$ has a unique complex representation

$$S^{\mathbb{C}}_{1,3}(\mathbb{R}) = S^{\mathbb{C}+}_{1,3}(\mathbb{R}) \oplus S^{\mathbb{C}-}_{1,3}(\mathbb{R}) \tag{A.12}$$

so that $\mathrm{Cl}_{1,3}(\mathbb{R}) \otimes \mathbb{C} \cong \mathrm{End}(S^{\mathbb{C}}_{1,3}(\mathbb{R}))$. $S^{\mathbb{C}\pm}_{1,3}(\mathbb{R})$ are closed under the action of $\mathrm{Cl}^0_{1,3}(\mathbb{R}) \otimes \mathbb{C}$, while

$$\mathrm{Cl}^1_{1,3}(\mathbb{R}) \otimes \mathbb{C} \cong \mathrm{Hom}(S^{\mathbb{C}\pm}_{1,3}, S^{\mathbb{C}\mp}_{1,3})\,. \tag{A.13}$$

We then define the Spin group to consist of normalized elements of an even grading of $\mathrm{Cl}_{1,3}(\mathbb{R})$ and the $\mathrm{Spin}^{\mathbb{C}}$ group to consist of normalized elements of an even grading of $\mathrm{Cl}_{1,3}(\mathbb{R}) \otimes \mathbb{C}$, that is

$$\begin{aligned} \mathrm{Spin}_{1,3}(\mathbb{R}) &= \{v_1v_2\cdots \in \mathrm{Cl}^0_{1,3}(\mathbb{R}) \mid \eta_{\mu\nu}v_i^\mu v_i^\nu = 1\}\,, \\ \mathrm{Spin}^{\mathbb{C}}_{1,3}(\mathbb{R}) &= \{v_1v_2\cdots z \in \mathrm{Cl}^0_{1,3}(\mathbb{R}) \otimes \mathbb{C} \mid \eta_{\mu\nu}v_i^\mu v_i^\nu = 1\}\,. \end{aligned} \tag{A.14}$$

As an example, consider here $\mathrm{Spin}_{0,3}(\mathbb{R}) \subseteq \mathrm{Cl}^0_{0,3}(\mathbb{R}) = \mathbb{H}$. The normalized v_i can be taken as the basis vectors e_i here. As can be seen from equation (A.10), the identification $\mathrm{Cl}_{0,3}(\mathbb{R}) \cong \mathbb{H} \oplus \mathbb{H}$ relates the even components 1, e_1e_2, e_1e_3 and e_2e_3 with a basis of $\mathbb{H}$. Since for the elements $v_1v_2\cdots \in \mathrm{Spin}_{0,3}(\mathbb{R})$ it has to hold that $\eta_{\mu\nu}v^\mu v^\nu = 1$, it follows that the $v_1v_2\cdots$ must themselves be of unit length, that is $v_1v_2\cdots = c^0 1 + c^1e_1e_2 + c^2e_1e_3 + c^3e_2e_3$ with $c^ic_i = 1$. We can thus identify $v_1v_2\cdots = c^\mu\sigma_\mu$ and

$$\mathrm{Spin}_{0,3}(\mathbb{R}) \cong \mathrm{SU}(2). \tag{A.15}$$

As another example consider $\mathrm{Spin}_{1,3}(\mathbb{R}) \subseteq \mathrm{Cl}^0_{1,3}(\mathbb{R})$ with $\mathrm{Spin}_{0,3}(\mathbb{R}) \subset \mathrm{Spin}_{1,3}(\mathbb{R})$, because $\mathrm{Cl}^0_{0,3}(\mathbb{R}) \subset \mathrm{Cl}_{0,3}(\mathbb{R}) \cong \mathrm{Cl}^0_{1,3}(\mathbb{R})$. Since there are $S^3 \cong \mathrm{SU}(2)$ possibilities to embed $\mathbb{R}^{0,3}$ into $\mathbb{R}^{0,4}$, we get

$$\mathrm{Spin}_4(\mathbb{R}) \cong \mathrm{SU}(2) \times \mathrm{SU}(2)\,, \tag{A.16}$$

where we can wick rotate the $\mathbb{R}^{0,4} \to \mathbb{R}^{1,3}$ which amounts to

$$\mathrm{Spin}_{1,3}(\mathbb{R}) \cong \mathrm{SL}(2,\mathbb{C})\,. \tag{A.17}$$

As a final example consider $\mathrm{Spin}^{\mathbb{C}}_{1,3}(\mathbb{R})$. It is $\mathrm{Spin}_{1,3}(\mathbb{R}) \times \mathrm{U}(1) \twoheadrightarrow \mathrm{Spin}^{\mathbb{C}}_{1,3}(\mathbb{R})$, where $(z, z^{-1}) \mapsto 1$ so that

$$\mathrm{Spin}^{\mathbb{C}}_{1,3}(\mathbb{R}) \cong \mathrm{SL}(2,\mathbb{C}) \times \mathrm{U}(1)/\mathbb{Z}_2\,. \tag{A.18}$$

Note that $\mathrm{Spin}_{1,3}(\mathbb{R})$ is a double cover of $\mathrm{SO}^+(1,3)$. To see that, for $v_1 v_2 \cdots v_k \in \mathrm{Cl}_{1,3}(\mathrm{R})$ define the transposition by

$$(v_1 v_2 \cdots v_k)^t = v_k \cdots v_2 v_1\,. \tag{A.19}$$

Because of the quotient in the definition (A.7) we can write

$$\mathrm{Spin}_{1,3}(\mathbb{R}) = \{v \in \mathrm{Cl}^0_{1,3}(\mathbb{R}) \mid v v^t = 1\}\,. \tag{A.20}$$

Furthermore set

$$\begin{gathered} \epsilon\colon \mathrm{Cl}_{1,3}(\mathbb{R}) \to \mathrm{Cl}_{1,3}(\mathbb{R}) \\ v_0 + v_1 \mapsto v_0 - v_1 \end{gathered} \tag{A.21}$$

then $v^\star = \epsilon(v)^t$ is an automorphism of $\mathrm{Cl}_{1,3}(\mathbb{R})$. Choose a $v \in \mathbb{R}^{1,3}$, then the map

$$\begin{gathered} \mathrm{Spin}_{1,3}(\mathbb{R}) \to \mathrm{SO}^+(1,3) \\ w \mapsto \rho_w \end{gathered} \tag{A.22}$$

with

$$\rho_w(v) = w v w^\star \tag{A.23}$$

defines a double covering

$$\mathrm{SO}^+(1,3) \cong \mathrm{SL}(2,\mathbb{C})/\mathbb{Z}_2\,. \tag{A.24}$$

In analogy, since

$$\mathrm{Spin}^{\mathbb{C}}_{1,3}(\mathbb{R})/\mathbb{Z}_2 \cong \mathrm{SO}^+(1,3) \times \mathrm{U}(1)\,, \tag{A.25}$$

$\mathrm{Spin}^{\mathbb{C}}_{1,3}(\mathbb{R})$ is the double cover of $\mathrm{SO}^+(1,3) \times \mathrm{U}(1)$. The complex $\mathrm{Cl}_{1,3}(\mathbb{R})$ representation $S^{\mathbb{C}}_{1,3}(\mathbb{R})$ can be restricted to a complex $\mathrm{Spin}_{1,3}(\mathbb{R})$ representation

$$\Delta^{\mathbb{C}}_{1,3}(\mathbb{R}) = \Delta^{\mathbb{C}+}_{1,3}(\mathbb{R}) \oplus \Delta^{\mathbb{C}-}_{1,3}(\mathbb{R}) : \mathrm{Spin}_{1,3}(\mathbb{R}) \to \mathrm{Aut}(S^{\mathbb{C}}_{1,3}(\mathbb{R}), S^{\mathbb{C}}_{1,3}(\mathbb{R}))\,. \tag{A.26}$$

Elements of $S^{\mathbb{C}}_{1,3}(\mathbb{R})$ under the action of $\Delta^{\mathbb{C}}_{1,3}(\mathbb{R})$ are called spinors. Since we established that $\mathrm{Spin}_{1,3}(\mathbb{R}) \cong \mathrm{SL}(2,\mathbb{C})$, $\Delta^{\mathbb{C}}_{1,3}(\mathbb{R})$ is the standard representation of $\mathrm{SL}(2,\mathbb{C})$ on $\mathbb{C}^2$, which is then called the Weyl representation. If $\Delta^{\mathbb{C}}_{1,3}(\mathbb{R})$ admits a

real structure, it is called the Majorana representation. Since $\Delta^{\mathbb{C}}_{1,3}(\mathbb{R})(-1) = -1$, it can be uniquely extended to a representation

$$\Delta^{\mathbb{C}}_{1,3,\mathbb{C}}(\mathbb{R})\colon \operatorname{Spin}^{\mathbb{C}}_{1,3}(\mathbb{R}) \to \mathrm{GL}(S^{\mathbb{C}}_{1,3}(\mathbb{R}), \mathbb{C})\,. \tag{A.27}$$

Now consider the tangent bundle TX of X regarded as the associated vector bundle to the frame bundle $F(TX)$ with the structure group being $\mathrm{SO}^+(1,3)$. We can promote $F(TX)$ to a $\operatorname{Spin}_{1,3}(\mathbb{R})$ bundle P if the double cover as given in equation (A.22) can be extended equivariantly to a map $P \to F(TX)$, which is true in most cases, especially if X is parallelizable. P is then called a spin structure for X. In analogy, if $P^{\mathbb{C}}$ is a $\operatorname{Spin}^{\mathbb{C}}_{1,3}(\mathbb{R})$ bundle and the double cover can be extended equivariantly to a map $P^{\mathbb{C}} \to F(TX)$, $P^{\mathbb{C}}$ is called a $\operatorname{Spin}^{\mathbb{C}}$ structure for X. It is always possible to define a $\operatorname{Spin}^{\mathbb{C}}$ structure for oriented four-dimensional manifolds. A subbundle of the $\operatorname{Spin}^{\mathbb{C}}$ bundle is the determinant line bundle $\mathcal{L} \to X$, with fiber $\operatorname{Spin}^{\mathbb{C}}_{1,3}(\mathbb{R})/\operatorname{Spin}_{1,3}(\mathbb{R})$. Since X admits Spin and $\operatorname{Spin}^{\mathbb{C}}$ structures P and $P^{\mathbb{C}}$, there also are associated complex spin bundles

$$S^{\mathbb{C}}_{1,3}(P) = P \times_{\operatorname{Spin}_{1,3}(\mathbb{R})} S^{\mathbb{C}}_{1,3}(\mathbb{R}) \quad \text{and} \quad S^{\mathbb{C}}_{1,3}(P^{\mathbb{C}}) = P \times_{\operatorname{Spin}^{\mathbb{C}}_{1,3}(\mathbb{R})} S^{\mathbb{C}}_{1,3}(\mathbb{R}) \tag{A.28}$$

induced by the representations $\Delta^{\mathbb{C}}_{1,3}(\mathbb{R})$ and $\Delta^{\mathbb{C}}_{1,3,\mathbb{C}}(\mathbb{R})$.

The spinors so far describe free theories. Lastly, we want to introduce connections on our Spin and $\operatorname{Spin}^{\mathbb{C}}$ bundles. Let $\pi\colon P \to X$ be a principal bundle with structure group G and Lie algebra $\mathfrak{g}$, then $VP = \ker(d\pi)$ is a subbundle of P called the vertical bundle and HP so that $T_pP = H_pP \oplus V_pP$ everywhere the horizontal bundle. A connection is a subbundle $\{H_p\}_p \subseteq HP$. Consider now a one-form

$$\omega_p : T_pP \to V_pP \to \mathfrak{g}\,, \tag{A.29}$$

so that $\ker(T_pP \to V_pP) = H_p$ and $V_pP \to \mathfrak{g}$ is the isomorphism given by the derivative of the action of G. We can promote $\{H_p\}_p$ to a connection on an associated vector bundle $E = P \times_G X$, given by the covariant derivative $\nabla\colon \Omega^0(X, \mathbb{R}^{1,3}) \to \Omega^1(X, \mathbb{R}^{1,3})$ defined for a section $\Omega^0(X, \mathbb{R}^{1,3})$ by for every $x \in X$ and every $\tau \in T_xX$ defining a curve γ so that $\tau = \frac{d}{dt}\gamma\big|_{t=0}$ and promoting γ to a curve $p(t)$ on P tangent to $\{H_p\}_p$. Then $\sigma|_\gamma = (x(t), p(t))$ and

$$\nabla(\sigma)(\tau) = \left(p, \frac{\partial x}{\partial t}\Big|_{t=0}\right). \tag{A.30}$$

For every connection one-form ω, we can define a two-form

$$F = d\omega + \frac{1}{2}\omega \wedge \omega\,, \tag{A.31}$$

which is the curvature or the field strength with respect to ω. There is a unique

connection on the frame bundle $F(TX)$ given by

$$\nabla_{e_\mu}(e_\nu)\Gamma^\rho_{\mu\nu}e_\rho\,, \tag{A.32}$$

where the Christoffel symbols $\Gamma^\rho_{\mu\nu}$ are defined as metric compatible, that is

$$\nabla_{e_\rho}(e_\mu)^\sigma e_{\nu,\sigma} + e^\sigma_\mu \nabla_{e_\rho}(e_\nu)_\sigma = \partial_\rho e^\sigma_\mu e_{\nu,\sigma} \tag{A.33}$$

and torsion free, so

$$\nabla_{e_\mu}(e_\nu) = \nabla_{e_\nu}(e_\mu)\,, \tag{A.34}$$

which is called the Levi-Civita connection. We get a Spin connection, if we promote the frame bundle $F(TX)$ to a spin structure P. The horizontal distribution on P is then given by the inverse of the differential of the double cover. We also get a connection on $S^{\mathbb{C}}_{1,3}(P)$ as the associated vector bundle to P. If ω is its connection one-form and $\sigma(u) = (u, s(u))$ the section, it is locally given by

$$\nabla(\sigma)(u) = \left(u, ds(u) + \frac{1}{2}\sum\nolimits_{\mu<\nu}\omega_{\nu\mu}e_\mu e_\nu s(u)\right)\,. \tag{A.35}$$

On a Spin$^{\mathbb{C}}$ bundle, we additionally have a $U(1)$ connection A on the determinant line bundle. Locally with e the basis vector in the section of $\mathcal{L}$ the connection has the form

$$\nabla_e(\sigma)(u) = \left(u, \frac{ds(u)}{de} + \frac{1}{2}\Big(\mathrm{i}A(e) + \sum\nolimits_{\mu<\nu}\omega_{\nu\mu}(e)e_\mu e_\nu\Big)\cdot s(u)\right)\,. \tag{A.36}$$

The connection A is the photon field. Using (A.36), we can define the Dirac operator on $S^{\mathbb{C}}_{1,3}(P)$, describing the dynamics of the fields. Locally it is given by

$$\not{\partial}_A(\sigma)(u) = e^\tau_\sigma \frac{ds(u)}{de^\tau_\sigma} + \frac{1}{2}\left(A_\tau(e_\sigma)e^\tau_\sigma + \sum\nolimits_{\nu<\rho}\omega_{\rho\nu}(e_\sigma)(e_\nu e_\rho)\right)s(u)\,. \tag{A.37}$$

B. On-Shell Momenta

To solve the linear system of equations (3.47) for the integrand coefficients, we have to evaluate the left-hand side on multiple points ℓ_l^Γ, which set the propagators defined by Γ on-shell. To this end, we extended the approach described in references [27, 59] to two loops. We first determine the loop momentum of one of the loops ℓ_1 by considering its one-loop diagram, where the other loop is pinched to a point. We then determine the remaining loop momentum ℓ_2 by looking at the other loop, now with a middle rung, where ℓ_1 is treated like an external momentum—as pictured in figure B.1. Note that this way, both of the representations have identical propagators cut. In every loop, we then parametrize the

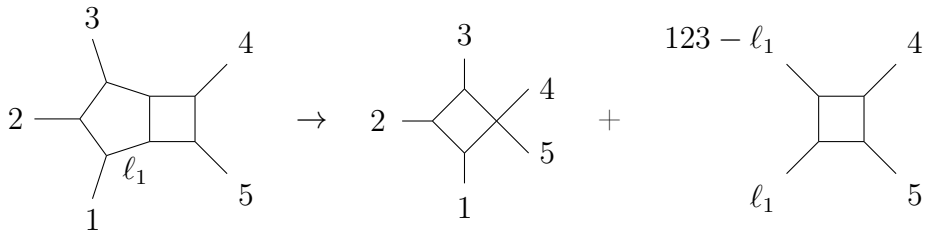


Figure B.1.: splitting of a two-loop cut into two one-loop cuts with the same propagators

loop momenta ℓ_1 and ℓ_2 by the inverse propagators ρ and some free to choose parameters α. We then can sample over α to get enough values for the left-hand side of equation (3.47) to solve the system of equations. To this end, we split the loop momentum ℓ_l into its physical part $r_i(\rho)v^i$ and its transverse part $\alpha^i n_i$

$$\ell_l = V + \alpha^j n_j = r_i(\rho)v^i + \alpha^j n_j \,. \tag{B.1}$$

The physical part $r_i(\rho)v^i$ is spanned by the vectors $v^i = (G^{-1})^{ij}p_j$ with $G_{ij} = p_i{\cdot}p_j$ the gram matrix, which are then dual to the momenta p_i meaning $p_i{\cdot}v^j = \delta_i^j$. The coefficients $r_i(\rho)$ can be determined from the inverse propagators $\rho_i = (\ell_l - p_{1i})^2$ to be

$$r_i(\rho) = \frac{1}{2}(\rho_{i-1} - \rho_i + p_{1i}^2 - p_{1,i-1}^2) \,. \tag{B.2}$$

The momenta p_i are the ones defining a cut; that is, for a bubble cut, there would be one momentum, for a triangle two, for a box three, and for a pentagon four. The transverse momenta n_i then have to span the $(D-4)$-dimensional space transverse to the one spanned by p_i. To this end, we choose arbitrary vectors t_i and project them onto the transverse space creating vectors u_i by

$$u_{i\mu} = g^{\perp}_{\mu\nu}t^\nu = (\eta_{\mu\nu} - v^i_\mu p_{i\nu})t^\nu \,. \tag{B.3}$$

In a last step we orthonormalize the u_i using Gram–Schmidt orthonormalization to arrive at the vectors n_i. Because the n_i are orthonormal, using equation (B.1) we get the condition

$$0 = \ell_l^2 = V^2 + \alpha_i\alpha^i \tag{B.4}$$

for the α_i, so that in case of codimension one we can solve for $\alpha_1 = \mathrm{i}\sqrt{V^2}$. For higher codimension we can parametrize α_1 and α_2 by

$$\alpha_1 = \frac{1}{2}\left(t - \frac{V^2 + \sum_{i>2}\alpha_i^2}{t}\right), \quad \text{and} \quad \alpha_2 = \frac{1}{2\mathrm{i}}\left(t + \frac{V^2 + \sum_{i>2}\alpha_i^2}{t}\right), \tag{B.5}$$

where the α_i with $i > 2$ can be freely chosen.

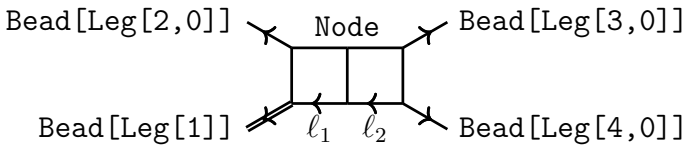


Figure C.1.: An exemplary double box `CaravelGraph` with the `Beads` and the last `Node` displayed. The `Links` with the loop momenta ℓ_1 and ℓ_2 read `Link[LoopMomentum[1]]` and `Link[LoopMomentum[-2]]` respectively. The first node connects ℓ_{-2}, the middle `Strand` and ℓ_1.

By this procedure, for every propagator structure Γ, we parametrize the loop momenta ℓ_1 and ℓ_2 by the propagators defining the cut and free variables $t, \alpha_3, \alpha_4, \ldots$. Lastly, for the propagators defining the cut, we set $\rho_i = 0$. This choice yields a parametrization of the solution space of the equations defining the cut.

C. CaravelGraph Structure

For every propagator structure Γ in equation (3.46), the headed list `CaravelGraph` defines the propagator structure and the loop momentum routing. A diagrammatic representation of an example `CaravelGraph` can be seen in figure C.1. A `CaravelGraph` consists of one or two `Nodes`, dependent on whether the propagator structure factorizes into two one-loop structures or not. The `Nodes` are the vertices connected to the middle rung. Every `Node` can have zero or one `Legs`, which represent the attached momenta. Every momentum is considered outgoing. Every `Leg` has one or two attributes, which are an index associated to it and an optional zero if the `Leg` is massless.

The second entry of `CaravelGraph` describes the `Connection` between the `Nodes`. It can have two or three rungs or `Strands`, dependent on whether the propagator structure is factorized or not. Every `Strand` goes from the first to the second `Node` or from the unique `Node` to itself if the propagator structure factorizes. The `Strands` consist of `Beads`, which have the attached `Legs` as attributes and the `Links` connecting the `Beads`. The first `Links` of the first and last `Strands` have an attribute `LoopMomentum` with an index with the absolute value 1 or 2 for the first and last `Strand` respectively. These define the positions of the loop momenta ℓ_1 and ℓ_2 within the propagator structure. A positive index signifies that the `LoopMomentum` flows along with the `Strand` from the first to the last `Node`, a negative index signifies that the `LoopMomentum` flows in the opposite direction.

The third entry of the `CaravelGraph` is a name describing the propagator structure. It has an optional index or further description in case the propagator structure is ambiguous within the basis. The suffix `Simple` signifies that there

are no Legs attached to the Nodes, SemiSimple means there is exactly one Leg attached to a Node and Generic means there are two Legs attached to the Nodes, one to each Node.

The CaravelGraph entry for the example in figure C.1 is

```
CaravelGraph[Nodes[Node[], Node[]],
             Connection[Strand[Link[LoopMomentum[1]], Bead[Leg[1]],
                              Link[], Bead[Leg[2, 0]], Link[]],
                        Strand[Link[]],
                        Strand[Link[LoopMomentum[-2]], Bead[Leg[4, 0]],
                              Link[], Bead[Leg[3, 0]], Link[]]],
              "BoxBoxSimple"]
```

D. Master Integrals for Three-Photon Production

The master integrals in the ancillary files for one- and two-loop amplitudes for three-photon production do not carry information about their propagator structures. Because of this, they are listed in table D.1 for one-loop amplitudes and in tables D.2, D.3 and D.4 for two-loop amplitudes. To not produce singularities, propagators must appear only once per integral, which is the case for the chosen bases. As in other Caravel studies, the external momenta are considered outgoing. Only the integrals corresponding to the clockwise ordering of the external particles are shown here. The integral basis consists of multiple integrals with the same name but different permutations of the Mandelstam invariants.

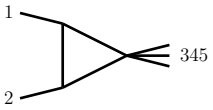
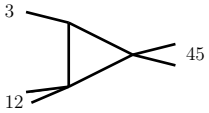
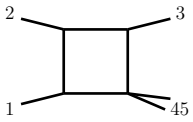
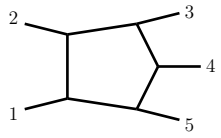
`MI["Bubble", i, s12]`	
`MI["OneMassTriangle", i, s12]`	
`MI["TwoMassTriangle", i, s12, s45]`	
`MI["OneMassBox", i, s45, s12, s23]`	
`MI["Pentagon", i, s12, s23, s45, s45, s15]`	

Table D.1.: One-loop master integrals momentum routing.

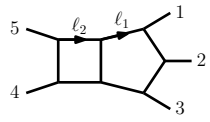
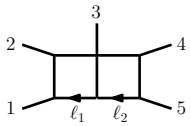
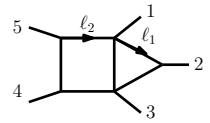
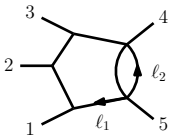
`MI["PentagonBox", i, s12, s23, s34, s15, s45]`	
`MI["FivePointDoubleBox", i, s12, s23, s34, s15, s45]`	
`MI["FivePointBoxTriangle", i, s12, s23, s34, s15, s45]`	
`MI["PentagonBubble", i, s12, s23, s34, s15, s45]`	

Table D.2.: Two-loop five-point master integrals momentum routing.

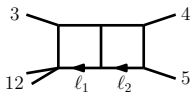
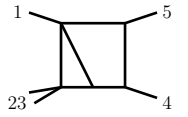
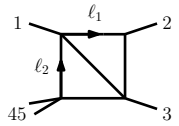
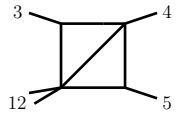
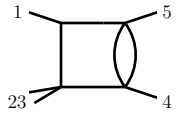
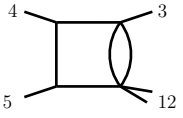
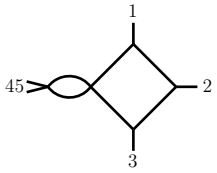
`MI["OneMassDoubleBox", i, s12, s34, s45]`	
`MI["FourPointBoxTriangle", i, s23, s14, s45]`	
`MI["OneMassSlashedBoxHard", i, s45, s12, s23]`	
`MI["OneMassSlashedBoxEasy", i, s12, s34, s45]`	
`MI["OneMassBoxBubbleHard", i, s23, s14, s45]`	
`MI["OneMassBoxBubbleEasy", i, s12, s34, s45]`	
`MI["FactorizedBoxBubble", i, s45, s12, s23]`	

Table D.3.: Two-loop four-point master integrals momentum routing.

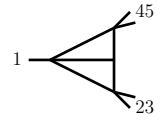
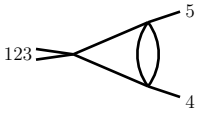
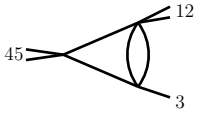
`MI["TwoMassSlashedTriangle", i, s23, s45]`	
`MI["OneMassTriangleBubble", i, s45]`	
`MI["TwoMassTriangleBubble", i, s12, s45]`	
`MI["ThreePointFactorizedBubbleBubble", i, s23, s45]`	
`MI["FactorizedBubbleBubble", i, s45]`	
`MI["Sunrise", i, s34]`	

Table D.4.: Two-loop three- and two-point master integrals momentum routing.

Bibliography

[1] O. S. Brüning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole, and P. Proudlock, *LHC Design Report*, CERN Yellow Reports: Monographs (CERN, Geneva, 2004).

[2] R. Bruce, N. Fuster-Martínez, A. Mereghetti, D. Mirarchi, and S. Redaelli, "Review of LHC Run 2 Machine Configurations," in *9th LHC Operations Evian Workshop* (Geneva, Switzerland, 2019).

[3] D. Bandurin *et al.*, "Review of Physics Results from the Tevatron," Int. J. Mod. Phys. A **30**, 1541001 (2015), arXiv:1409.4861 [hep-ex].

[4] S. Abachi *et al.* (D0), "Observation of the top quark," Phys. Rev. Lett. **74**, 2632–2637 (1995), arXiv:hep-ex/9503003.

[5] F. Abe *et al.* (CDF), "Evidence for top quark production in $\bar{p}p$ collisions at $\sqrt{s} = 1.8$ TeV," Phys. Rev. D **50**, 2966–3026 (1994).

[6] W. Bernreuther, "Top quark physics at the LHC," J. Phys. G **35**, 083001 (2008), arXiv:0805.1333 [hep-ph].

[7] T. Dado, T. Peiffer, A. Quadt, M. Racko, E. Shabalina, and S. Tokar (ATLAS Collaboration), *Measurement of the top-quark decay width in top-quark pair events in the dilepton channel at $\sqrt{s} = 13$ TeV with the ATLAS detector*, Tech. Rep. ATLAS-CONF-2019-038 (CERN, Geneva, 2019).

[8] G. Aad *et al.* (ATLAS), "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC," Phys. Lett. B **716**, 1–29 (2012), arXiv:1207.7214 [hep-ex].

[9] S. Chatrchyan *et al.* (CMS), "Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC," Phys. Lett. B **716**, 30–61 (2012), arXiv:1207.7235 [hep-ex].

[10] S. Manzoni, *Physics with Photons Using the ATLAS Run 2 Data: Calibration and Identification, Measurement of the Higgs Boson Mass and Search for Supersymmetry in Di-Photon Final State*, Springer Theses (Springer International Publishing, 2019).

[11] J. Boyd, "LHC Run-2 and Future Prospects," in *2019 European School of High-Energy Physics* (2020) arXiv:2001.04370 [hep-ex].

[12] N. Karastathis *et al.*, "LHC Run 3 Configuration Working Group Report," in *9th LHC Operations Evian Workshop* (Geneva, Switzerland, 2019).

[13] ATLAS collaboration, *Search for high-mass dilepton resonances using* $139\,\mathrm{fb}^{-1}$ *of pp collision data collected at* $\sqrt{s} = 13\,\mathrm{TeV}$ *with the ATLAS detector*, Tech. Rep. ATLAS-CONF-2019-001 (CERN, Geneva, 2019).

[14] A. M. Sirunyan *et al.* (CMS), "Measurement of the differential cross sections for the associated production of a W boson and jets in proton-proton collisions at $\sqrt{s} = 13$ TeV," Phys. Rev. D **96**, 072005 (2017), arXiv:1707.05979 [hep-ex].

[15] S. Frixione, Z. Kunszt, and A. Signer, "Three jet cross-sections to next-to-leading order," Nucl. Phys. B **467**, 399–442 (1996), arXiv:hep-ph/9512328.

[16] S. Catani and M. H. Seymour, "A General algorithm for calculating jet cross-sections in NLO QCD," Nucl. Phys. B **485**, 291–419 (1997), [Erratum: Nucl.Phys.B 510, 503–504 (1998)], arXiv:hep-ph/9605323.

[17] S. Catani, S. Dittmaier, M. H. Seymour, and Z. Trocsanyi, "The Dipole formalism for next-to-leading order QCD calculations with massive partons," Nucl. Phys. B **627**, 189–265 (2002), arXiv:hep-ph/0201036.

[18] M. Czakon, C. G. Papadopoulos, and M. Worek, "Polarizing the Dipoles," JHEP **08**, 085 (2009), arXiv:0905.0883 [hep-ph].

[19] B. W. Harris and J. F. Owens, "The Two cutoff phase space slicing method," Phys. Rev. D **65**, 094032 (2002), arXiv:hep-ph/0102128.

[20] G. Passarino and M. J. G. Veltman, "One Loop Corrections for e+ e- Annihilation Into mu+ mu- in the Weinberg Model," Nucl. Phys. B **160**, 151–207 (1979).

[21] K. Chetyrkin and F. Tkachov, "Integration by parts: The algorithm to calculate β-functions in 4 loops," Nuclear Physics B **192**, 159 – 204 (1981).

[22] S. Weinzierl, "Introduction to Feynman Integrals," in *6th Summer School on Geometric and Topological Methods for Quantum Field Theory* (2010) arXiv:1005.1855 [hep-ph].

[23] J. M. Henn, "Lectures on differential equations for Feynman integrals," J. Phys. A **48**, 153001 (2015), arXiv:1412.2296 [hep-ph].

[24] G. Heinrich, "Collider Physics at the Precision Frontier," Phys. Rept. **922**, 1–69 (2021), arXiv:2009.00516 [hep-ph].

[25] T. Gehrmann, J. Henn, and N. Lo Presti, "Pentagon functions for massless planar scattering amplitudes," JHEP **10**, 103 (2018), arXiv:1807.09812 [hep-ph].

[26] G. Ossola, C. G. Papadopoulos, and R. Pittau, "Reducing full one-loop amplitudes to scalar integrals at the integrand level," Nucl. Phys. B **763**, 147–169 (2007), arXiv:hep-ph/0609007.

[27] R. Ellis, W. Giele, and Z. Kunszt, "A Numerical Unitarity Formalism for Evaluating One-Loop Amplitudes," JHEP **03**, 003 (2008), arXiv:0708.2398 [hep-ph].

[28] C. Berger, Z. Bern, L. Dixon, F. Febres Cordero, D. Forde, H. Ita, D. Kosower, and D. Maitre, "An Automated Implementation of On-Shell Methods for One-Loop Amplitudes," Phys. Rev. D **78**, 036003 (2008), arXiv:0803.4180 [hep-ph].

[29] H. Ita, "Two-loop Integrand Decomposition into Master Integrals and Surface Terms," Phys. Rev. D **94**, 116015 (2016), arXiv:1510.05626 [hep-th].

[30] S. Abreu, J. Dormans, F. Febres Cordero, H. Ita, B. Page, and V. Sotnikov, "Analytic Form of the Planar Two-Loop Five-Parton Scattering Amplitudes in QCD," JHEP **05**, 084 (2019), arXiv:1904.00945 [hep-ph].

[31] S. Abreu, J. Dormans, F. Febres Cordero, H. Ita, M. Kraus, B. Page, E. Pascual, M. Ruf, and V. Sotnikov, "Caravel: A C++ framework for the computation of multi-loop amplitudes with numerical unitarity," Computer Physics Communications , 108069 (2021), `https://gitlab.com/caravel-public`, arXiv:2009.11957 [hep-ph].

[32] H. A. Chawdhry, M. L. Czakon, A. Mitov, and R. Poncelet, "NNLO QCD corrections to three-photon production at the LHC," JHEP **02**, 057 (2020), arXiv:1911.00479 [hep-ph].

[33] G. Das and P. Mathews, "Neutral Triple Vector Boson Production in Randall-Sundrum Model at the LHC," Phys. Rev. D **92**, 094034 (2015), arXiv:1507.08857 [hep-ph].

[34] N. Toro and I. Yavin, "Multiphotons and photon jets from new heavy vector bosons," Phys. Rev. D **86**, 055005 (2012), arXiv:1202.6377 [hep-ph].

[35] M. Aaboud *et al.* (ATLAS), "Measurement of the production cross section of three isolated photons in pp collisions at $\sqrt{s} = 8$ TeV using the ATLAS detector," Phys. Lett. B **781**, 55–76 (2018), arXiv:1712.07291 [hep-ex].

[36] D. Chicherin and V. Sotnikov, "Pentagon Functions for Scattering of Five Massless Particles," JHEP **12**, 167 (2020), arXiv:2009.07803 [hep-ph].

[37] S. Kallweit, V. Sotnikov, and M. Wiesemann, "Triphoton production at hadron colliders in NNLO QCD," Phys. Lett. B **812**, 136013 (2021), arXiv:2010.04681 [hep-ph].

[38] H. A. Chawdhry, M. Czakon, A. Mitov, and R. Poncelet, "Two-loop leading-color helicity amplitudes for three-photon production at the LHC," (2020), arXiv:2012.13553 [hep-ph].

[39] L. Evans and P. Bryant, "LHC machine," Journal of Instrumentation **3**, S08001–S08001 (2008).

[40] C. Biscari, "Accelerators R&D," PoS **EPS-HEP2009**, 019 (2009).

[41] R. Feynman, "The behavior of hadron collisions at extreme energies," Conf.Proc. **C690905**, 237–258 (1969).

[42] M. Cacciari, G. P. Salam, and G. Soyez, "The anti-k_t jet clustering algorithm," JHEP **04**, 063 (2008), arXiv:0802.1189 [hep-ph].

[43] S. Frixione, "Isolated photons in perturbative QCD," Phys. Lett. B **429**, 369–374 (1998), arXiv:hep-ph/9801442.

[44] H. Năstase, *Classical field theory* (Cambridge University Press, 2019).

[45] M. D. Schwartz, *Quantum field theory and the standard model* (Cambridge University Press, 2014).

[46] R. D'Auria and M. Trigiante, *From Special Relativity to Feynman Diagrams: A Course of Theoretical Particle Physics for Beginners*, UNITEXT (Springer Milan, 2011).

[47] P. A. Dirac, "The lagraingian in quantum mechanics," *Physikalische Zeitschrift der Sowjetunion*, , 64–72 (1933).

[48] R. P. Feynman, *The principle of least action in quantum mechanics*, Ph.D. thesis, Princeton U. (1942).

[49] R. P. Feynman, "Space-time approach to nonrelativistic quantum mechanics," Rev. Mod. Phys. **20**, 367–387 (1948).

[50] M. Srednicki, *Quantum field theory* (Cambridge University Press, 2007).

[51] W. Pauli and F. Villars, "On the invariant regularization in relativistic quantum theory," Rev. Mod. Phys. **21**, 434–444 (1949).

[52] G. 't Hooft and M. Veltman, "Regularization and renormalization of gauge fields," Nuclear Physics B **44**, 189–213 (1972).

[53] Z. Bern and D. A. Kosower, "The computation of loop amplitudes in gauge theories," Nuclear Physics B **379**, 451–561 (1992).

[54] S. Abreu, F. Febres Cordero, H. Ita, B. Page, and V. Sotnikov, "Planar Two-Loop Five-Parton Amplitudes from Numerical Unitarity," JHEP **11**, 116 (2018), arXiv:1809.09067 [hep-ph].

[55] G. 't Hooft and M. Veltman, "Regularization and Renormalization of Gauge Fields," Nucl. Phys. B **44**, 189–213 (1972).

[56] D. A. Kosower, "The Spinor helicity method in dimensional regularization," Phys. Lett. B **254**, 439–443 (1991).

[57] V. Sotnikov, *Scattering amplitudes with the multi-loop numerical unitarity method*, Ph.D. thesis, Freiburg U. (2019).

[58] M. Veltman, "GAMMATRICA," Nucl. Phys. B **319**, 253–270 (1989).

[59] W. T. Giele, Z. Kunszt, and K. Melnikov, "Full one-loop amplitudes from tree amplitudes," JHEP **04**, 049 (2008), arXiv:0801.2237 [hep-ph].

[60] L. J. Dixon, "Calculating scattering amplitudes efficiently," in *Theoretical Advanced Study Institute in Elementary Particle Physics (TASI 95): QCD and Beyond* (1996) pp. 539–584, arXiv:hep-ph/9601359.

[61] R. Britto, F. Cachazo, B. Feng, and E. Witten, "Direct proof of tree-level recursion relation in Yang-Mills theory," Phys. Rev. Lett. **94**, 181602 (2005), arXiv:hep-th/0501052.

[62] S. Badger, E. Glover, V. Khoze, and P. Svrcek, "Recursion relations for gauge theory amplitudes with massive particles," JHEP **07**, 025 (2005), arXiv:hep-th/0504159.

[63] F. A. Berends and W. Giele, "Recursive Calculations for Processes with n Gluons," Nucl. Phys. B **306**, 759–808 (1988).

[64] A. V. Kotikov, "Differential equations method. new technique for massive feynman diagram calculation," Physics Letters B **254**, 158–164 (1991).

[65] J. M. Henn and J. C. Plefka, *Scattering Amplitudes in Gauge Theories*, Vol. 883 (Springer, Berlin, 2014).

[66] V. A. Smirnov, *Evaluating Feynman Integrals*, 2004th ed., Springer Tracts in Modern Physics (Springer, 2004).

[67] J. M. Henn, "Multiloop integrals in dimensional regularization made simple," Phys. Rev. Lett. **110**, 251601 (2013), arXiv:1304.1806 [hep-th].

[68] S. Laporta, "High precision calculation of multiloop Feynman integrals by difference equations," Int. J. Mod. Phys. A **15**, 5087–5159 (2000), arXiv:hep-ph/0102033.

[69] A. V. Smirnov, "Algorithm FIRE – Feynman Integral REduction," JHEP **10**, 107 (2008), arXiv:0807.3243 [hep-ph].

[70] A. von Manteuffel and C. Studerus, "Reduze 2 - Distributed Feynman Integral Reduction," (2012), arXiv:1201.4330 [hep-ph].

[71] P. Maierhöfer, J. Usovitsch, and P. Uwer, "Kira—A Feynman integral reduction program," Comput. Phys. Commun. **230**, 99–112 (2018), arXiv:1705.05610 [hep-ph].

[72] J. Klappert, F. Lange, P. Maierhöfer, and J. Usovitsch, "Integral reduction with Kira 2.0 and finite field methods," Comput. Phys. Commun. **266**, 108024 (2021), arXiv:2008.06494 [hep-ph].

[73] A. V. Kotikov, "Differential equations method: New technique for massive Feynman diagrams calculation," Phys. Lett. B **254**, 158–164 (1991).

[74] K.-T. Chen, "Iterated path integrals," Bulletin of the American Mathematical Society **83**, 831–879 (1977).

[75] A. B. Goncharov, M. Spradlin, C. Vergu, and A. Volovich, "Classical Polylogarithms for Amplitudes and Wilson Loops," Phys. Rev. Lett. **105**, 151605 (2010), arXiv:1006.5703 [hep-th].

[76] D. Chicherin, J. Henn, and V. Mitev, "Bootstrapping pentagon functions," JHEP **05**, 164 (2018), arXiv:1712.09610 [hep-th].

[77] A. B. Goncharov, "Polylogarithms in arithmetic and geometry," in *Proceedings of the International Congress of Mathematicians* (Springer, 1995) pp. 374–387.

[78] C. Duhr, "Mathematical aspects of scattering amplitudes," in *Theoretical Advanced Study Institute in Elementary Particle Physics: Journeys Through the Precision Frontier: Amplitudes for Colliders* (2014) arXiv:1411.7538 [hep-ph].

[79] J. Vollinga, "GiNaC: Symbolic computation with C++," Nucl. Instrum. Meth. A **559**, 282–284 (2006), `https://ginac.de/`, arXiv:hep-ph/0510057.

[80] E. Panzer, "Algorithms for the symbolic integration of hyperlogarithms with applications to Feynman integrals," Comput. Phys. Commun. **188**, 148–166 (2015), arXiv:1403.3385 [hep-th].

[81] C. Duhr and F. Dulat, "PolyLogTools — polylogs for the masses," JHEP **08**, 135 (2019), arXiv:1904.07279 [hep-th].

[82] H. Frellesvig, "Generalized Polylogarithms in Maple," (2018), arXiv:1806.02883 [hep-th].

[83] S. Abreu, J. Dormans, F. Febres Cordero, H. Ita, and B. Page, "Analytic Form of Planar Two-Loop Five-Gluon Scattering Amplitudes in QCD," Phys. Rev. Lett. **122**, 082002 (2019), arXiv:1812.04586 [hep-ph].

[84] S. Abreu, B. Page, E. Pascual, and V. Sotnikov, "Leading-Color Two-Loop QCD Corrections for Three-Photon Production at Hadron Colliders," JHEP **01**, 078 (2021), `https://gitlab.com/five-point-amplitudes`, arXiv:2010.15834 [hep-ph].

[85] J. Gluza, K. Kajda, and D. A. Kosower, "Towards a Basis for Planar Two-Loop Integrals," Phys. Rev. D **83**, 045012 (2011), arXiv:1009.0472 [hep-th].

[86] R. M. Schabinger, "A New Algorithm For The Generation Of Unitarity-Compatible Integration By Parts Relations," JHEP **01**, 077 (2012), arXiv:1111.4220 [hep-ph].

[87] K. J. Larsen and Y. Zhang, "Integration-by-parts reductions from unitarity cuts and algebraic geometry," Phys. Rev. D **93**, 041701 (2016), arXiv:1511.01071 [hep-th].

[88] S. Abreu, F. Febres Cordero, H. Ita, M. Jaquier, B. Page, M. Ruf, and V. Sotnikov, "Two-Loop Four-Graviton Scattering Amplitudes," Phys. Rev. Lett. **124**, 211601 (2020), arXiv:2002.12374 [hep-th].

[89] V. Barone and E. Predazzi, *High-energy particle diffraction* (Springer Science & Business Media, 2002).

[90] S. Badger, H. Frellesvig, and Y. Zhang, "A Two-Loop Five-Gluon Helicity Amplitude in QCD," JHEP **12**, 045 (2013), arXiv:1310.1051 [hep-ph].

[91] A. Hodges, "Eliminating spurious poles from gauge-theoretic amplitudes," JHEP **05**, 135 (2013), arXiv:0905.1473 [hep-th].

[92] S. Catani, "The Singular behavior of QCD amplitudes at two loop order," Phys. Lett. B **427**, 161–171 (1998), arXiv:hep-ph/9802439.

[93] Z. Kunszt, A. Signer, and Z. Trocsanyi, "Singular terms of helicity amplitudes at one loop in QCD and the soft limit of the cross-sections of multiparton processes," Nucl. Phys. B **420**, 550–564 (1994), arXiv:hep-ph/9401294.

[94] T. Peraro, "Scattering amplitudes over finite fields and multivariate functional reconstruction," JHEP **12**, 030 (2016), arXiv:1608.01902 [hep-ph].

[95] E. K. Leinartas, "Factorization of rational functions of several variables into partial fractions," Izv. Vyssh. Uchebn. Zaved. Mat. , 47–51 (1978).

[96] S. Abreu, F. Febres Cordero, H. Ita, B. Page, and M. Zeng, "Planar Two-Loop Five-Gluon Amplitudes from Numerical Unitarity," Phys. Rev. D **97**, 116014 (2018), arXiv:1712.03946 [hep-ph].

[97] F. Berends and W. Giele, "Recursive calculations for processes with n gluons," Nuclear Physics B **306**, 759 – 808 (1988).

[98] F. Anger, F. Febres Cordero, H. Ita, and V. Sotnikov, "NLO QCD predictions for $Wb\bar{b}$ production in association with up to three light jets at the LHC," Phys. Rev. D **97**, 036018 (2018), arXiv:1712.05721 [hep-ph].

[99] S. Badger, C. Brønnum-Hansen, T. Gehrmann, H. B. Hartanto, J. Henn, N. A. Lo Presti, and T. Peraro, "Applications of integrand reduction to two-loop five-point scattering amplitudes in QCD," PoS **LL2018**, 006 (2018), arXiv:1807.09709 [hep-ph].

[100] B. Jacob and G. Guennebaud, "Eigen (3.3.9)," `http://eigen.tuxfamily.org/` (2020).

[101] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed. (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999) `http://www.netlib.org/lapack/`.

[102] T. Granlund and the GMP development team, *GNU MP: The GNU Multiple Precision Arithmetic Library*, 5th ed. (2012), `http://gmplib.org/`.

[103] C. Cheung and G. N. Remmen, "Hidden Simplicity of the Gravity Action," JHEP **09**, 002 (2017), arXiv:1705.00626 [hep-th].

[104] Y. Hida, S. Li, and D. Bailey, "Quad-double arithmetic: Algorithms, implementation, and application," (2001), `https://www.davidhbailey.com/dhbsoftware/`.

[105] GNU Project, "MPFR (4.1.0)," `https://www.mpfr.org/` (2020).

[106] D. van Heesch, "Doxygen (1.9.1)," `http://www.doxygen.nl/` (2021).

[107] S. Weinzierl, "Tales of 1001 Gluons," Phys. Rept. **676**, 1–101 (2017), arXiv:1610.05318 [hep-th].

[108] J. Pakkanen, "Meson (0.58.0)," `https://mesonbuild.com/` (2021).

[109] GNU Project, "Automake (1.16.3)," `https://www.gnu.org/software/automake/` (2020).

[110] Python Software Foundation, "Python (3.9.5)," `https://www.python.org/` (2021).

[111] I. Bicking, "pip (21.1.1)," `https://pypi.org/project/pip/` (2021).

[112] E. Martin, "Ninja (1.10.2)," `https://ninja-build.org/` (2020).

[113] P. Holoborodko, "mpreal (3.6.8)," `https://github.com/advanpix/mpreal/` (2020).

[114] B. Haible and R. B. Kreckel, "CLN (1.3.6)," `https://www.ginac.de/CLN/` (2019).

[115] L. Clarke, I. Glendinning, and R. Hempel, "The mpi message passing interface standard," Int. J. Supercomput. Appl. **8** (1996), 10.1007/978-3-0348-8534-8_21.

[116] The Open MPI Project, "Open MPI (4.0.5)," `https://www.open-mpi.org/` (2020).

[117] P. Nogueira, "Automatic Feynman graph generation," J. Comput. Phys. **105**, 279–289 (1993), `http://cfif.ist.utl.pt/~paulo/qgraf.html`.

[118] A. Ochirov and B. Page, "Full Colour for Loop Amplitudes in Yang-Mills Theory," JHEP **02**, 100 (2017), arXiv:1612.04366 [hep-ph].

[119] A. Ochirov and B. Page, "Multi-Quark Colour Decompositions from Unitarity," JHEP **10**, 058 (2019), arXiv:1908.02695 [hep-ph].

[120] Mozilla Research, "Rust (1.53.0)," `https://www.rust-lang.org/` (2021).

[121] P. Barrett, "Implementing the rivest shamir and adleman public key encryption algorithm on a standard digital signal processor," in *Advances in Cryptology — CRYPTO' 86*, edited by A. M. Odlyzko (Springer Berlin Heidelberg, Berlin, Heidelberg, 1987) pp. 311–323.

[122] D. Chicherin, T. Gehrmann, J. M. Henn, P. Wasser, Y. Zhang, and S. Zoia, "All master integrals for three-jet production at next-to-next-to-leading order," Phys. Rev. Lett. **123**, 041603 (2019).

[123] J. van der Hoeven, G. Lecerf, and G. Quintin, "Modular SIMD arithmetic in mathemagix," CoRR **abs/1407.3383** (2014), arXiv:1407.3383.

[124] N. Arkani-Hamed, J. L. Bourjaily, F. Cachazo, and J. Trnka, "Local Integrals for Planar Scattering Amplitudes," JHEP **06**, 125 (2012), arXiv:1012.6032 [hep-th].

[125] F. Buccioni, J.-N. Lang, J. M. Lindert, P. Maierhöfer, S. Pozzorini, H. Zhang, and M. F. Zoller, "OpenLoops 2," Eur. Phys. J. C **79**, 866 (2019), arXiv:1907.13071 [hep-ph].

[126] E. Glover and M. Tejeda-Yeomans, "Two loop QCD helicity amplitudes for massless quark massless gauge boson scattering," JHEP **06**, 033 (2003), arXiv:hep-ph/0304169.

[127] D. A. Kosower, "All order collinear behavior in gauge theories," Nucl. Phys. B **552**, 319–336 (1999), arXiv:hep-ph/9901201.

[128] Z. Bern, L. J. Dixon, and D. A. Kosower, "Two-loop g —> gg splitting amplitudes in QCD," JHEP **08**, 012 (2004), arXiv:hep-ph/0404293.

[129] S. D. Badger and E. W. N. Glover, "Two loop splitting functions in QCD," JHEP **07**, 040 (2004), arXiv:hep-ph/0405236.

[130] Z. Bern, V. Del Duca, W. B. Kilgore, and C. R. Schmidt, "The infrared behavior of one loop QCD amplitudes at next-to-next-to leading order," Phys. Rev. D **60**, 116001 (1999), arXiv:hep-ph/9903516.

[131] L. Lewin, "Polylogarithms and associated functions," New York (1981).

[132] S. Weinzierl, "Does one need the $O(\epsilon)-$ and $O(\epsilon^2)$-terms of one-loop amplitudes in an NNLO calculation ?" Phys. Rev. D **84**, 074007 (2011), arXiv:1107.5131 [hep-ph].

[133] S. Catani, L. Cieri, D. de Florian, G. Ferrera, and M. Grazzini, "Universality of transverse-momentum resummation and hard factors at the NNLO," Nucl. Phys. B **881**, 414–443 (2014), arXiv:1311.1654 [hep-ph].

[134] M. Grazzini, S. Kallweit, and M. Wiesemann, "Fully differential NNLO computations with MATRIX," Eur. Phys. J. C **78**, 537 (2018), arXiv:1711.06631 [hep-ph].

[135] J. M. Campbell, R. Ellis, Y. Li, and C. Williams, "Predictions for diphoton production at the LHC through NNLO in QCD," JHEP **07**, 148 (2016), arXiv:1603.02663 [hep-ph].

[136] S. Catani and M. Grazzini, "An NNLO subtraction formalism in hadron collisions and its application to Higgs boson production at the LHC," Phys. Rev. Lett. **98**, 222002 (2007), arXiv:hep-ph/0703012.

[137] S. Badger, C. Brønnum-Hansen, D. Chicherin, T. Gehrmann, H. B. Hartanto, J. Henn, M. Marcoli, R. Moodie, T. Peraro, and S. Zoia, "Virtual QCD corrections to gluon-initiated diphoton plus jet production at hadron colliders," (2021), arXiv:2106.08664 [hep-ph].

[138] B. Agarwal, F. Buccioni, A. von Manteuffel, and L. Tancredi, "Two-loop helicity amplitudes for diphoton plus jet production in full color," (2021), arXiv:2105.04585 [hep-ph].

[139] H. A. Chawdhry, M. Czakon, A. Mitov, and R. Poncelet, "Two-loop leading-colour QCD helicity amplitudes for two-photon plus jet production at the LHC," (2021), arXiv:2103.04319 [hep-ph].

[140] S. Abreu, H. Ita, F. Moriello, B. Page, W. Tschernow, and M. Zeng, "Two-Loop Integrals for Planar Five-Point One-Mass Processes," JHEP **11**, 117 (2020), arXiv:2005.04195 [hep-ph].

[141] J. W. Morgan, *The Seiberg-Witten Equations and Applications to the Topology of Smooth Four-Manifolds* (Princeton University Press, 2014).

Acknowledgements

I thank Fernando Febres Cordero for always having an open door allowing fruitful discussions throughout my doctoral studies. I thank Harald Ita for putting my work into a broader context and helpful suggestions on how to proceed. I thank Wladimir Tschernow for always being ready to discuss problems as soon as he is in the office, Christopher Schwan for many technical insights, and Jerry Dormans, Giuseppe De Laurentis, Maximilian Klinkert, and Michael Ruf for lively group meetings. I thank Samuel Abreu and Ben Page for the collaboration in the three-photon calculation. I especially thank Vasily Sotnikov for the fruitful cooperation and constant pressure during the three-photon calculation, making the results possible. I thank my parents Galina and Sergej, my brother Alex Kossinz and my wife Lilian, for the help during the beginning of the corona pandemic.