UNIVERSITY OF
CAMBRIDGE

# Higher-Order Calculations in Quantum Chromodynamics

**Herschel Aditya Chawdhry**

Department of Physics
University of Cambridge

This dissertation is submitted for the degree of
*Doctor of Philosophy*

Trinity College

November 2020

# Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Chapter 2 is based on the paper [1], which was written in collaboration with A. Mitov. Section 2.2 of that chapter is based on the Part III Project Report that I submitted as part of my MSci degree at the University of Cambridge in 2016.

Sections 3.1 and 3.3 of Chapter 3 are based on the paper [2], which was written in collaboration with M. A. Lim and A. Mitov.

Chapter 4 is based on the paper [3], which was written in collaboration with M. Czakon, A. Mitov, and R. Poncelet.

This thesis is typeset using LaTeX with a template designed by Krishna Kumar [4].

Herschel Aditya Chawdhry
November 2020

# Abstract

## Higher-Order Calculations in Quantum Chromodynamics

Herschel Aditya Chawdhry

In this thesis, several techniques and advances in higher-order Quantum Chromodynamics (QCD) calculations are presented. There is a particular focus on 2-loop 5-point massless QCD amplitudes, which are currently at the frontier of higher-order QCD calculations.

Firstly, we study the Brodsky-Lepage-Mackenzie/Principle of Maximum Conformality (BLM/PMC) method for setting the renormalisation scale, $\mu_R$, in higher-order QCD calculations. We identify three ambiguities in the BLM/PMC procedure and study their numerical impact using the example of the total cross-section for $t\bar{t}$ production at Next-to-Next-to-Leading Order (NNLO) in QCD. The numerical impact of these ambiguities on the BLM/PMC prediction for the cross-section is found to be comparable to the impact of the choice of $\mu_R$ in the conventional scale-setting approach.

Secondly, we introduce a novel strategy for solving integration-by-parts (IBP) identities, which are widely used in the computation of multi-loop QCD amplitudes. We implement the strategy in an efficient `C++` program and hence solve the IBP identities needed for the computation of any planar 2-loop 5-point massless amplitude in QCD. We also derive representative results for the most complicated non-planar family of integrals.

Thirdly, we present an automated computational framework to reduce 2-loop 5-point massless amplitudes to a basis of pentagon functions. It uses finite-field evaluation and interpolation techniques, as well as the aforementioned analytical IBP results. We use this to calculate the leading-colour 2-loop QCD amplitude for $q\bar{q} \to \gamma\gamma\gamma$ and then compute the NNLO QCD corrections to 3-photon production at the LHC. This is the first NNLO QCD calculation for a $2 \to 3$ process. We compare our predictions with the available 8 TeV measurements from the ATLAS collaboration and we find that the inclusion of the NNLO corrections eliminates the existing significant discrepancy with respect to NLO QCD predictions, paving the way for precision phenomenology in this process.

# Acknowledgements

I would like to begin by thanking my supervisor, Prof. Alexander Mitov, for his constant guidance, support, and advice. One could not ask for a better supervisor.

I am very grateful to everyone who has taught me during my eight years at Trinity College and the University of Cambridge. I am likewise grateful to all of my schoolteachers.

It has been a pleasure to work with my excellent colleagues — and by now friends — in the High-Energy Physics group at the Cavendish Laboratory in Cambridge. I have greatly enjoyed our many enlightening discussions during the last few years. I would like to especially thank Dr. Mathieu Pellen for providing helpful comments on an early draft of this thesis.

I would also like to extend a word of thanks to all of my friends in Cambridge and beyond. I hope all the adventures of our student years will lead to many more in the years to come.

I am grateful to the European Union for funding this PhD, as well as for funding several of the schools and workshops that I attended. This research was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 683211).

Finally, I am eternally grateful to Mama, Abi, and Rischabh for their encouragement, advice, support, and love.

*To my parents*

# Table of contents

# List of figures

# List of tables

# Chapter 1

# Introduction

## 1.1 Quantum Chromodynamics

Quantum Chromodynamics (QCD) describes the behaviour of quarks and gluons, which are the fundamental constituents of protons, neutrons, and a variety of more exotic particles, collectively known as hadrons. A core pillar of the Standard Model of Particle Physics, QCD has been very successful in explaining the spectrum of hadrons discovered during the last 100 years[1] as well as in predicting numerous observations and measurements at high-energy particle colliders such as the Large Hadron Collider (LHC).

High-energy QCD calculations exploit the property of *asymptotic freedom*: at energies above $\Lambda_{\text{QCD}} \approx 200$ MeV, the QCD strong coupling constant $\alpha_s$ governing the strength of the Strong Force becomes small[2], as will be explained in Section 1.3. This allows high-energy observables to be calculated as a perturbative series in $\alpha_s$. Calculations at leading order in $\alpha_s$ are comparatively simple, while higher-order calculations quickly increase in complexity but provide predictions of higher precision and accuracy. These higher-order calculations are the subject of this thesis.

The field of higher-order QCD calculations has seen steady progress over time. For processes involving 3 or more particles observed in the final state, current fixed-order QCD predictions are at Next-to-Leading Order (NLO) in $\alpha_s$ and can have uncertainties in excess of 20% [6]. During the last 20 years, pioneering work by a large number of

---

[1] Last year marked the 100th anniversary of the discovery [5] of the proton by Ernest Rutherford.

[2] The term 'strong coupling *constant*' is thus somewhat misleading. The name arose in analogy to the well-known electromagnetic fine-structure constant $\alpha \approx \frac{1}{137}$, whose value shows little variation at the energy scales of interest outside the field of High-Energy Physics.

research groups has gradually enabled most processes with 1 or 2 observed final-state particles to be computed at Next-to-Next-to-Leading Order (NNLO), yielding state-of-the-art predictions with uncertainties typically in the region of just a few percent. The next frontier in NNLO QCD calculations will involve processes with 3 particles observed in the final state and work in this direction currently constitutes a very active area of research. These so-called $2 \to 3$ processes are a major focus of this thesis, and in Chapter 4 we will present the first-ever NNLO QCD calculation for a $2 \to 3$ process: 3-photon production at the LHC. These high-precision calculations are of direct interest to the experimental community: the LHC is due to spend the next two decades producing high-precision measurements which can be compared against high-precision QCD predictions in order to either further validate the Standard Model or provide the first hints of the physics that lies beyond.

The rest of this introductory chapter will summarise the fundamentals of QCD as well as introducing several topics that underpin the material appearing in later chapters.

## 1.2 The QCD Lagrangian

QCD is based on the gauge group $\mathrm{SU}(N_c)$, where $N_c = 3$ is the number of quark colours. As we will see in Section 1.7, it is often helpful to work with generic $N_c$.

The Lagrangian of QCD can be written as:

$$\mathcal{L}_{\text{QCD}} = \mathcal{L}_0 + \mathcal{L}_{\text{gauge-fixing+ghost}} + \mathcal{L}_{\text{counterterms}} \,. \tag{1.1}$$

Let us start with the classical part, $\mathcal{L}_0$.

$$\mathcal{L}_0 = -\frac{1}{4} F^A_{\mu\nu} F^{\mu\nu}_A + \sum_q \bar{q}_a \left( i\gamma^\mu D_\mu - m_q \right) q_a \tag{1.2}$$

Here, $F^A_{\mu\nu}$ is the QCD field strength tensor (defined below), $\{q\}$ are the quark fields, $m_q$ is the mass of quark flavour $q$, the index $a = 1, ..., N_c$ represents the colours of the quarks, and the index $A = 1, ..., N_c^2 - 1$ represents the colours of the gluons. The covariant derivative $D_\mu$ is here given by:

$$D_\mu = \partial_\mu + i g_s \mathcal{A}^A_\mu t^A, \tag{1.3}$$

with $t^A$ defined below. The QCD field strength tensor $F^A_{\mu\nu}$ is defined in terms of the $N_c^2 - 1$ gluon fields $\mathcal{A}^A_\mu$ as:

$$F^A_{\mu\nu} = \partial_\mu \mathcal{A}^A_\nu - \partial_\nu \mathcal{A}^A_\mu - g_s f^{ABC} \mathcal{A}^B_\mu \mathcal{A}^C_\nu , \tag{1.4}$$

where $f^{ABC}$ are the structure constants of the $\mathrm{SU}(N_c)$ gauge group, $g_s$ is the strong coupling constant (although we will typically work with $\alpha_s = \frac{g_s^2}{4\pi}$), and $\mu$ and $\nu$ are Lorentz indices.

The $t^A$ are the generators of $\mathrm{SU}(N_c)$ in the defining representation[3]. There exist many possible explicit matrix representations for the $t^A$. By definition, these representations must satisfy the $\mathrm{SU}(N_c)$ Lie algebra

$$\left[t^A, t^B\right] = i f^{ABC} t^C , \tag{1.5}$$

and they also satisfy the following additional relations:

$$\mathrm{Tr}\left(t^A t^B\right) = \frac{1}{2}\delta^{AB} \tag{1.6}$$

$$t^A_{ab} t^A_{bc} = C_F \delta_{ac} \tag{1.7}$$

$$\left\{t^A, t^B\right\} = \frac{1}{N_c}\delta^{AB} + d^{ABC} t^C . \tag{1.8}$$

The factor of $\frac{1}{2}$ in Equation 1.6 is a matter of convention and sets the overall normalisation of the $t^A$ matrices. The defining representation Casimir is $C_F = \frac{N_c^2-1}{2N_c}$ and the symmetric tensor $d^{ABC}$ is defined as

$$d^{ABC} = 2\,\mathrm{Tr}\left[t^A \left\{t^B, t^C\right\}\right] . \tag{1.9}$$

Besides the defining representation, QCD calculations also employ the adjoint representation, whose matrices $T^A$ are explicitly defined as follows:

$$(T^A)_{BC} = -i f^{ABC} . \tag{1.10}$$

---

[3] The defining representation of $\mathrm{SU}(N_c)$ is a fundamental representation and is often referred to as *the* fundamental representation.

They can be used to form a Casimir operator

$$\mathrm{Tr}\left(T^A T^B\right) = C_A \delta^{AB} \tag{1.11}$$

$$C_A = N_c \tag{1.12}$$

and they also satisfy the SU($N_c$) Lie algebra (Equation 1.5).

## Gauge fixing

The gluon fields $\mathcal{A}_\mu^A$ contain extra, unphysical degrees of freedom. In order for the Green function of the gluons (and hence the gluon propagator) to be well defined, it is necessary to fix the gauge by including additional terms in the Lagrangian. To ensure the physics is not modified by the gauge-fixing terms, one must at the same time introduce unphysical ghost fields which are scalar but obey Fermion statistics.

A common choice is the covariant set of gauges:

$$\mathcal{L}_{\text{gauge-fixing+ghost}} = -\frac{1}{2\xi}\left(\partial^\mu \mathcal{A}_\mu^A\right)^2 + (\partial_\mu \bar{\eta}^A) D^\mu_{AB} \eta^B \,, \tag{1.13}$$

where $\eta$ and $\bar{\eta}$ are ghost and anti-ghost fields, respectively. Here, one can view $\frac{1}{2\xi}$ as a Lagrange multiplier enforcing the gauge-fixing condition $\left(\partial^\mu \mathcal{A}_\mu^A\right)^2 = 0$. One can further fix the gauge by choosing a specific value of $\xi$, such as $\xi = 1$ (Feynman gauge) or $\xi \to 0$ (Landau gauge). Physical predictions are gauge-invariant and therefore do not depend on the value of $\xi$.

Another common choice of gauge is the axial gauge where we instead define

$$\mathcal{L}_{\text{gauge-fixing+ghost}} = -\frac{1}{2\lambda}\left(n^\mu \mathcal{A}_\mu^A\right)^2 + (n_\mu \bar{\eta}^A) D^\mu_{AB} \eta^B \,, \tag{1.14}$$

where $\lambda$ is analogous to the $\xi$ appearing in Equation 1.13, while $n^\mu$ is an arbitrary fixed vector. The axial gauge has the advantage that, if one chooses $\lambda \to 0$, the ghost fields decouple and can therefore be neglected. Furthermore, calculations can often be simplified by making a judicious choice for $n^\mu$. The special case where $n^\mu n_\mu = 0$ and $\lambda = 0$ is known as the lightcone gauge.

## Renormalisation and counterterms

In order to cancel divergences from loop integrals contributing to physical predictions, it is necessary to add counterterms $\delta_i$ to the Lagrangian. As will be further explained in Section 1.6, we use dimensional regularisation to regulate UV and IR divergences by working in $d = 4 - 2\epsilon$ space-time dimensions, and this requires introducing a renormalisation scale, $\mu_R$. The complete QCD Lagrangian, including the combined effects of gauge-fixing and renormalisation, can be found in textbooks such as Ref. [7]. The counterterms have the net effect of changing the normalisation of the fields (including ghost fields), masses, and coupling constant:

$$q_a^{(\text{bare})} = Z_2^{1/2} q_a^{(\text{Renormalised})} \tag{1.15}$$

$$A_\mu^{(\text{bare})} = Z_3^{1/2} A_\mu^{(\text{Renormalised})} \tag{1.16}$$

$$\eta^{(\text{bare})} = Z_{3\eta}^{1/2} \eta^{(\text{Renormalised})} \tag{1.17}$$

$$m_q^{(\text{bare})} = Z_m m_q^{(\text{Renormalised})} \tag{1.18}$$

$$g_s^{(\text{bare})} = Z_g^{1/2} \mu_R^{(4-d)/2} g_s^{(\text{Renormalised})}, \tag{1.19}$$

where $Z_i = 1 + \delta_i$. It is helpful to define $Z_1 = Z_g Z_2 Z_3^{1/2}$.

When working in dimensional regularisation, the poles in $\epsilon$ of the counterterms $\delta_i$ are determined by requiring the Green functions involving loop diagrams to be finite. Here we will state the 1-loop values for the counterterms $\delta_1$, $\delta_2$ and $\delta_3$, which we will use in Section 1.3 to derive the first term of the QCD $\beta$ function. We use the covariant gauges defined in Equation 1.13.

$$\delta_1 = \frac{-1}{\epsilon}\left(\frac{\alpha_s}{4\pi}\right)\left(\xi C_F + \frac{3+\xi}{4} C_A\right) \tag{1.20}$$

$$\delta_2 = \frac{-1}{\epsilon}\left(\frac{\alpha_s}{4\pi}\right)\xi C_F \tag{1.21}$$

$$\delta_3 = \frac{-1}{\epsilon}\left(\frac{\alpha_s}{4\pi}\right)\left(\frac{(3\xi - 13)}{6} C_A + \frac{2}{3} n_f\right). \tag{1.22}$$

Here $\alpha_s \equiv \frac{1}{4\pi}\left[g_s^{(\text{Renormalised})}\right]^2$ and $n_f$ is the number of active quark flavours. The factor of $n_f$ appearing in Equation 1.22 gives rise to the $n_f$-dependence of the $\beta_0$ coefficient of the QCD $\beta$ function, which we will describe in the next section. This is used in the BLM/PMC renormalisation scale-setting method that we will study in Chapter 2.

## 1.3 The running of $\alpha_S$

Quantitative predictions for high-energy QCD observables are typically calculated as a perturbative series in the strong coupling constant $\alpha_s \equiv \frac{g_s^2}{4\pi}$, which quantifies the strength of the Strong Force, analogously to the well-known electromagnetic fine-structure constant $\alpha \approx \frac{1}{137}$. Although the bare strong coupling $\alpha_s^{(\text{bare})}$ is an (infinite) constant, the renormalised strong coupling $\alpha_s$ is dependent on the renormalisation scale $\mu_R$ (when working in dimensional regularisation). The $\mu_R$-dependence of $\alpha_s$ can be obtained using the renormalisation group equation:

$$0 = \frac{d\alpha_s^{(\text{bare})}}{d\log(\mu_R^2)} = \frac{d}{d\log(\mu_R^2)}\left(\mu_R^{4-d}\frac{Z_1^2}{Z_2^2 Z_3}\alpha_s\right). \tag{1.23}$$

Setting $d = 4 - 2\epsilon$ and expanding in $\alpha_s$, we obtain the following:

$$\beta(\alpha_s) = -\epsilon\alpha_s - \frac{d}{d\log(\mu_R^2)}\left(2\delta_1 - 2\delta_2 - \delta_3\right) + \mathcal{O}\left(\alpha_s^2\right), \tag{1.24}$$

where we have defined

$$\beta(\alpha_s) \equiv \frac{d\alpha_s}{d\log(\mu_R^2)}. \tag{1.25}$$

This function, $\beta(\alpha_s)$, is the QCD $\beta$-function and can in general be expanded as a series in $\alpha_s$:

$$\beta(\alpha_s) = -\epsilon\alpha_s - \sum_{n=0}\left(\frac{\alpha_s}{4\pi}\right)^{n+2}\beta_n. \tag{1.26}$$

By substituting Equations 1.20–1.22 into Equation 1.24 and comparing with Equation 1.26, the one-loop beta-function coefficient $\beta_0$ can be seen to take the value shown in Equation 1.27. For reference, the value of the two-loop coefficient, $\beta_1$, is also stated as this will be used in Chapter 2. In recent years, the $\beta$ function has been computed to five loops [8–10].

$$\beta_0 = \frac{11}{3}C_A - \frac{2}{3}n_f \tag{1.27}$$

$$\beta_1 = \frac{34}{3}C_A^2 - \frac{10}{3}n_f - 2C_F n_f \tag{1.28}$$

The renormalisation scale is an arbitrary parameter and could, in principle, be set to any value of one's choosing. However, in practice one usually sets $\mu_R = Q$, where $Q$ is a typical momentum scale in the process. This choice makes $\alpha_s(Q)$ an effective coupling constant which absorbs some of the scale-dependent effects of loop corrections to propagators and vertices.

If $\alpha_s$ is measured at one energy, the differential equation 1.26 can be used to calculate $\alpha_s$ at any other energy. Remarkably, the $\beta$-function is negative[4] and as a result, QCD is strongly interacting at low energies but becomes *asymptotically free* at high energies. At energy scales $Q \gg \Lambda_{\mathrm{QCD}} \sim 200$ MeV, we have $\alpha_s < 1$ and so we can calculate observables as a perturbative series in $\alpha_s(Q)$. It is helpful to explicitly state the first few terms of the solution to the differential equation 1.26:

$$\alpha_s(\mu_2) = \alpha_s(\mu_1) - \beta_0 L \frac{\alpha_s^2(\mu_1)}{4\pi} + \left(\beta_0^2 L^2 - \beta_1 L\right) \frac{\alpha_s^3(\mu_1)}{16\pi^2} + \mathcal{O}\left(\alpha_s^4\right), \tag{1.29}$$

where $\mu_1$ and $\mu_2$ are any two energy scales and $L = \log\left(\frac{\mu_2^2}{\mu_1^2}\right)$. We will use this solution in Chapter 2.

## 1.4 Partons and factorisation

The particles that are collided by the LHC, and the particles that are measured in its detectors, are not the free quarks and gluons shown in the QCD Lagrangian (Equation 1.2). As a result of the running of $\alpha_s$, the interactions of quarks and gluons become strong at energies below $\Lambda_{\mathrm{QCD}}$, causing them to form bound states: protons, neutrons, and other hadrons. The LHC is a proton-proton collider[5] and any quarks or gluons that are produced in a collision will form hadrons before reaching the detector.

The QCD factorisation formula shown in Equation 1.30 is essential for the successful modelling of proton collisions, allowing them to be described in terms of the collisions of two *partons*.

$$\sigma = \sum_i \sum_j \int_0^1 d\xi_1 \int_0^1 d\xi_2 f_i(\xi_1) f_j(\xi_2) \hat{\sigma}_{ij} + \mathcal{O}\left(\frac{\Lambda_{\mathrm{QCD}}^N}{Q^N}\right) \tag{1.30}$$

Here, $\sigma$ is the hadronic cross-section while $\hat{\sigma}_{ij}$ are partonic cross-sections. The two partons, labelled $i$ and $j$, can each be a quark, anti-quark, or gluon. The *higher-twist* corrections $\mathcal{O}\left(\frac{\Lambda_{\mathrm{QCD}}^N}{Q^N}\right)$ can be neglected as long as $\Lambda_{\mathrm{QCD}}^N \ll Q^N$, where $Q$ is a typical

---

[4] Since $N_c = 3$ in QCD, the $\beta$-function is negative for $n_f < 17$. To date, 6 flavours of quarks have been discovered.

[5] Approximately 10% of the running time at the LHC is dedicated to collisions involving heavy ions, but they are outside the scope of this thesis.

momentum scale in the process and $N \in \mathbb{N}$ is a process-dependent constant. The $f_i(\xi)$ are called *Parton Distribution Functions* (PDFs). If an incoming proton has momentum $P^\mu$, one can loosely view $f_i(\xi)$ as the probability density of finding a parton $i$ with momentum $\xi P^\mu$ inside the proton. This probabilistic interpretation of the meaning of $f_i$ is less applicable when working beyond leading order in $\alpha_s$, as the formalism requires initial-state collinear divergences to be absorbed into the PDFs. Note that at the Tevatron, where one of the incoming particles is an anti-proton, it is necessary to replace one of the proton PDFs in Equation 1.30 by the corresponding anti-proton PDF.

The usefulness of the Equation 1.30 is due to the fact that the PDFs are universal: they are independent of the specific process being described by $\sigma$ and $\hat{\sigma}_{ij}$. Since the PDFs describe low-energy, non-perturbative physics, they cannot at present be computed directly. However, the universality of the PDFs ensures that they can be fitted to data from one process and then used to make predictions for a different process (even at a different collider). The fitting of high-quality PDFs is a major topic of research [11–14] and plays an important role in reducing the size of theoretical uncertainties in physical predictions. There is also a promising field of *lattice QCD* research, which aims to directly compute non-perturbative QCD properties, such as PDFs, by using numerical simulations of QCD in a discretised space-time.

The factorisation of $\sigma$ into a partonic cross-section $\hat{\sigma}_{ij}$ and PDFs comes at a cost of introducing an arbitrary and unphysical factorisation scale, $\mu_F$. This scale separates the high-energy, perturbative physics described by $\hat{\sigma}_{ij}$ and the low-energy, non-perturbative physics described by the PDFs. The PDFs and partonic cross-sections carry a dependence on $\mu_F$. In principle, the $\mu_F$-dependence should cancel at the level of the hadronic cross-section, but in practice any perturbative calculation at finite order in $\alpha_s$ will carry a residual dependence on $\mu_F$. We note that the $\mu_F$-dependence of the PDFs is calculable [15–17] and so, unlike the $\xi$-dependence, does not need to be fitted from data. Although in many applications it is common to set $\mu_F = \mu_R$, it is important to remember that the two scales have different origins and are distinct. The BLM/PMC scale-setting method studied in Chapter 2 applies only to $\mu_R$ and not $\mu_F$.

The techniques developed in this thesis relate primarily to the high-energy partonic cross-sections $\hat{\sigma}_{ij}$, since these can be calculated perturbatively in $\alpha_s$. If $\hat{\sigma}_{ij}$ describes the production of quarks or gluons, these will typically produce *jets* containing many more quarks and gluons, which will form colourless bound states (hadrons) before reaching the detector. This is described by *parton showers* and hadronisation models, but they are outside the scope of this thesis.

## 1.5 Higher-order corrections

To make quantitative predictions for a scattering process, one computes the corresponding squared scattering amplitude, $|\mathcal{M}|^2$, which is then integrated over phase-space as necessary to obtain the desired total or differential cross-section. To go beyond leading order in $\alpha_s$, two sources of corrections must be considered: *real* and *virtual*.

Virtual corrections arise from Feynman diagrams containing closed loops. Each loop adds an additional power of $\alpha_s$ at the level of the amplitude $\mathcal{M}$. Each loop also introduces a free momentum parameter (*loop momentum*) over which one must integrate. These *loop integrals* are a major bottleneck in higher-order QCD calculations and are the subject of Chapter 3. In the limit of large loop momentum, loop integrals have ultraviolet (UV) divergences which are removed using the counterterms that were added to the Lagrangian above. The loop integrals also have infrared (IR) divergences, which arise due to singularities in the integrands.

Real corrections arise from additional unresolved particles appearing in the final state. These additional particles are not individually resolved, either because their energy is too low to be detected (*soft particles*), or because they are *collinear* to one of the resolved particles. It is, of course, possible for a particle to be both soft and collinear. Since the additional particles are unresolved, these higher-multiplicity processes contribute to the cross-section describing the resolved particles. Each unresolved particle introduces a factor of $g_s$ at the level of the matrix element $\mathcal{M}$ and hence a factor of $\alpha_s$ at the level of the cross-section. Each unresolved particle also introduces an extra phase-space integral which must be performed over the momentum of the unresolved particle. These phase-space integrals give rise to IR divergences whenever an unresolved particle is either soft or collinear to a resolved particle.[6]

Amplitudes of different multiplicities are calculated independently and are only summed at the level of the cross-section. In an NNLO calculation, one must consider double-real corrections (two unresolved additional particles), real-virtual corrections (one unresolved particle and one loop), and double-virtual corrections (two loops). The double-virtual corrections include the square of the one-loop amplitude, and also the

---

[6] It is also possible to have *initial-state* collinear divergences but, as mentioned in Section 1.4, one can define higher-order PDFs which absorb them.

contraction of the two-loop amplitude with the tree-level (zero-loop) amplitude. These corrections are separately divergent but it can be proven [18, 19] that after summing all the corrections contributing to the cross-section at a given order in $\alpha_s$, the IR divergences cancel and the result is finite. To control and cancel these divergences is highly non-trivial in practice, however: it requires a regulator to be introduced so as to separate, and eventually cancel, the divergences arising from the various contributions. The regulator is treated as a symbolic (rather than numerical) variable throughout the calculation and it allows divergences at intermediate stages of the calculation to be made manifest, typically as poles or logarithms of the regulator variable.

## 1.6 Dimensional regularisation

In this field, the most commonly used regularisation scheme is dimensional regularisation [20], where the intermediate parts of the calculation are analytically continued to $d = 4 - 2\epsilon$ space-time dimensions.[7] More precisely, we work in $d - 1$ spatial dimensions and 1 time dimension. Divergent quantities such as loop integrals are Laurent-expanded in $\epsilon$ and the divergences appear as poles in $\epsilon$.

In dimensional regularisation, the integration measure for the Lagrangian is modified

$$\int d^4x \to \int d^dx, \tag{1.31}$$

which causes the mass dimensions of the fields to change. In order to sensibly perform calculations perturbatively in $\alpha_s$, we would like to keep the couplings dimensionless. This requires us to make the replacement

$$g_s \to \mu_R^\epsilon g_s \tag{1.32}$$

in the Lagrangian, where $\mu_R$ is the *renormalisation scale*, an arbitrary parameter with dimensions of mass. Individual parts of a calculation will become dependent on $\mu_R$ but this dependence should, in principle, cancel at the level of a physical observable. In practice, as we discuss in Chapter 2, any fixed-order calculation will have a residual dependence on $\mu_R$, which can have a significant impact on the numerical values of physical predictions.

---

[7] Another convention sometimes adopted is $d = 4 - \epsilon$ but in this thesis we will always use $d = 4 - 2\epsilon$.

The loop and phase-space integration measures also get modified in dimensional regularisation

$$\alpha_s^L \int d^4k_1 \dots d^4k_L \to \alpha_s^L \mu_R^{2L\epsilon} \int d^dk_1 \dots d^dk_L \tag{1.33}$$

$$\int \prod_i \left[ \frac{d^3p_i}{(2\pi)^3} \frac{1}{2E_i} \right] \to \int \prod_i \left[ \frac{d^{d-1}p_i}{(2\pi)^{d-1}} \frac{1}{2E_i} \right], \tag{1.34}$$

where the factors of $\mu_R$ are due to Equation 1.32 but are more conveniently treated as being part of the loop integral. Divergences – whether UV or IR – appear in loop integrals in the form of poles in $\epsilon$. The ability to use the same regulator to treat UV and IR poles is one of the benefits of dimensional regularisation.

The UV divergences in loop integrals are handled by adding appropriate counterterms, as described in Section 1.2. It is observed that these UV poles always appear in the combination

$$\frac{1}{\epsilon} - \gamma_E + \ln(4\pi), \tag{1.35}$$

where the Euler–Mascheroni constant $\gamma_E$ is related to the derivative of the Gamma function:

$$\gamma_E = - \left. \frac{d\Gamma(x)}{dx} \right|_{x=1} \approx 0.57721566. \tag{1.36}$$

In the Minimal Subtraction (MS) scheme [21, 22], the counterterms are chosen so as to only remove the poles in $\epsilon$. In this thesis we will use the Modified Minimal Subtraction ($\overline{\text{MS}}$) scheme, in which the constants in Equation 1.35 are subtracted alongside the poles.

Besides modifying the coupling constants, loop integrals, and phase-space integrals, dimensional regularisation also gives rise to factors of $d$ when summing over Lorentz indices

$$g^{\mu\nu} g_{\mu\nu} = d \tag{1.37}$$

$$\frac{dk^\mu}{dk^\mu} = \delta^\mu_\mu = d, \tag{1.38}$$

where $k^\mu$ is a $d$-dimensional momentum variable. Furthermore, a careful handling is required for momenta, polarisation vectors, and gamma matrices, and there exist a number of schemes that vary in their treatment of these quantities. Examples include the 't Hooft-Veltman scheme [20], Conventional Dimensional Regularisation [23], and Dimensional Reduction [24]. Note that although Dirac spinors have 4 components, this number is unrelated to the space-time dimension and so is unaffected by dimensional regularisation.

## 1.7 Planar graphs and the large-$N_c$ approximation

When working with SU($N_c$) theories such as QCD, it is helpful to consider the limit $N_c \to \infty$. More precisely, we define an auxiliary coupling $\tilde{\alpha}_s = N_c \alpha_s$ and we take the limit $N_c \to \infty$ while holding $\tilde{\alpha}_s$ constant.[8] Calculating a multi-loop amplitude while treating $1/N_c$ as a small parameter, one usually finds that the leading term in $1/N_c$ is simpler to compute than the full result. This *leading-colour* or *large-$N_c$* limit often provides a good approximation for the full amplitude, the sub-leading terms typically being *colour-suppressed* by a relative factor of $1/N_c^2$, where $N_c = 3$ for QCD.

The reason for the simplification is that there is a direct relation between the power of $N_c$ accompanying a Feynman diagram (as determined by the Feynman rules) and the genus of the surface required to draw the diagram without making lines cross over each other. Feynman diagrams embedded in higher-genus surfaces are associated with more complicated loop integrals, but these diagrams are colour-suppressed by powers of $1/N_c$. In the large-$N_c$ limit, only *planar* Feynman diagrams are required: those that can be drawn on a flat sheet without making any lines cross, while extending all external particle lines to infinity.

An intuitive understanding for the colour-suppression of non-planar graphs can be gained by drawing some planar and non-planar graphs on a sheet of paper and counting the number of closed colour cycles. Each colour cycle enhances the contribution of the graph by a factor of $C_A = N_c$ or $C_F = \frac{N_c^2-1}{2N_c} \approx \frac{1}{2}N_c$. It is quickly seen that whenever two lines cross (the distinguishing feature of non-planar graphs), the number of colour cycles is reduced. A more formal discussion can be found in Ref. [25], which first introduced the large-$N_c$ limit.

The greater complexity of non-planar integrals compared to planar integrals is borne out by experience in a wide variety of multi-loop calculations [26–31]. The contrast between the running times of the calculations for planar and non-planar integrals in Chapter 3 is illustrative of this.

---

[8] This subtlety is important for ensuring that the limit is well-defined, particularly in non-perturbative calculations, although it can be ignored when comparing terms of a given perturbative order in $\alpha_s$.

These considerations will be used in Chapter 4, where we will apply the large-$N_c$ approximation when calculating the scale-independent part of the two-loop finite remainder for $q\bar{q} \to \gamma\gamma\gamma$. (The scale-*dependent* part can be deduced in full colour from the tree-level part and the one-loop finite remainder.)

## 1.8 Polylogarithms and their generalisations

The irrational parts of loop amplitudes in QCD calculations are often expressible in terms of the polylogarithms $\mathrm{Li}_n(z)$, the Riemann zeta function $\zeta(n)$, and their generalisations. In this section we will briefly introduce these functions and discuss some of their basic properties.

The polylogarithm $\mathrm{Li}_n(z)$ can be defined as a series in $z \in \mathbb{C}$

$$\mathrm{Li}_n(z) = \sum_{k=1}^{\infty} \frac{z^k}{k^n}, \tag{1.39}$$

which is convergent for $|z| < 1$ and can be extended to the rest of the complex plane by analytic continuation. This defines the polylogarithm for $n \in \mathbb{C}$, although in QFT applications we restrict ourselves to $n \in \mathbb{N}$. For $n = 1$ this reduces to the Taylor expansion of the ordinary natural logarithm:

$$\mathrm{Li}_1(z) = -\ln(1-z). \tag{1.40}$$

For $n \in \mathbb{N}$, the polylogarithm can also be defined as nested integrals via

$$\mathrm{Li}_n(z) = \int_0^z \frac{1}{y} \mathrm{Li}_{n-1}(y) dy, \tag{1.41}$$

where now we explicitly define $\mathrm{Li}_1(z)$ with Equation 1.40.

The polylogarithms satisfy a number of relations, such as

$$\mathrm{Li}_2(1-z) + \mathrm{Li}_2\left(1 - \frac{1}{z}\right) + \frac{\log(z)^2}{2} = 0. \tag{1.42}$$

It is useful to introduce the concept of *transcendental weight*: a rational function has weight 0, while $\mathrm{Li}_n$ has weight $n$. The weight of the product of two functions is the sum of the weights of the individual functions. All terms in a polylogarithmic identity have the same weight. For example, each term in Equation 1.42 has weight 2.

As can be seen by setting $z = 1$ in Equation 1.39, the polylogarithms are related to the Riemann zeta function $\zeta$ via the relation

$$\mathrm{Li}_n(1) = \zeta(n). \tag{1.43}$$

These numbers $\zeta(n)$ often appear in polylogarithmic identities. We assign $\zeta(n)$ the transcendental weight $n$.

An important generalisation of the polylogarithms, with applications to amplitude calculations, is the class of Multiple Polylogarithms (MPLs) [32], defined by analogy to Equation 1.39 as

$$\mathrm{Li}_{n_1,\ldots,n_N}(z_1,\ldots,z_N) = \sum_{0<k_1<\cdots<k_N} \prod_i \frac{z_i^{k_i}}{k_i^{n_i}}\,. \tag{1.44}$$

One can then straight-forwardly define the Multiple Zeta Value $\zeta(n_1,\ldots,n_N)$ as

$$\zeta(n_1,\ldots,n_N) = \mathrm{Li}_{n_1,\ldots,n_N}(1,\ldots,1) = \sum_{0<k_1<\cdots<k_N} \frac{1}{k_1^{n_1} k_2^{n_2} \ldots k_N^{n_N}}\,. \tag{1.45}$$

As with the ordinary polylogarithms, the MPLs have equivalent integral representations and there are furthermore many known identities describing these functions. The Feynman integrals of interest in Chapters 3 and 4 can ultimately be expressed in terms of so-called *pentagon-functions* [33], which are linear combinations of MPLs. In Section 4.3.1 of Chapter 4, we will derive new identities between these pentagon functions in order to obtain simpler amplitudes for 2 → 3 processes.

## 1.9 Fields and vector spaces

In this section we will introduce two useful algebraic structures which will be encountered in this thesis: *fields* and *vector spaces*.

A field is a set $\mathbb{F}$ equipped with two operations called *addition* ($+ : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$) and *multiplication* ($* : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$), with the following properties:

1. Under addition, the elements of $\mathbb{F}$ form an abelian group. The identity element of this additive group is labelled 0.

2. Under multiplication, the elements of $\mathbb{F} \setminus \{0\}$ form an abelian group, and
$$a * 0 = 0 * a = 0 \quad \forall a \in \mathbb{F}. \tag{1.46}$$

3. Multiplication is distributive over addition:
$$a * (b + c) = a * b + a * c \quad \forall a, b, c \in \mathbb{F}. \tag{1.47}$$

Common examples of fields include the rational numbers $\mathbb{Q}$, the real numbers $\mathbb{R}$, and the complex numbers $\mathbb{C}$. In each of these fields, the standard addition and multiplication operations can easily be verified to satisfy the field properties stated above. On the other hand, the integers $\mathbb{Z}$ do not form a field since, for instance, the integer 2 does not have a multiplicative inverse within $\mathbb{Z}$.

In this thesis, two special types of fields will arise. Firstly, there are the *finite fields*, $\mathbb{F}_q$, which we will discuss in the next section. Secondly, there is $\mathbb{Q}(x_1, \ldots, x_n)$, which is the *field of fractions of the ring of $n$-variable polynomials with rational coefficients.* To define $\mathbb{Q}(x_1, \ldots, x_n)$, let us first define $\mathbb{Q}[x_1, \ldots, x_n]$ to be the set (formally a *ring*) of polynomials in some variables $x_1, \ldots, x_n$ with coefficients in $\mathbb{Q}$.[9] The field $\mathbb{Q}(x_1, \ldots, x_n)$ then consists of the set of rational functions
$$r(x_1, \ldots, x_n) \equiv \frac{p(x_1, \ldots, x_n)}{q(x_1, \ldots, x_n)}, \tag{1.48}$$
where $p, q \in \mathbb{Q}[x_1, \ldots, x_n]$. The reason $\mathbb{Q}(x_1, \ldots, x_n)$ constitutes a field is as follows. $\mathbb{Q}[x_1, \ldots, x_n]$ constitutes an *integral domain*: multiplication in $\mathbb{Q}[x_1, \ldots, x_n]$ is commutative, and the product of any two non-zero polynomials $p_1, p_2 \in \mathbb{Q}[x_1, \ldots, x_n] \setminus \{0\}$ is non-zero. It is known from mathematics that if $R$ is an integral domain, then the set of fractions $\{a/b : a, b \in R\}$ constitutes[10] a field, known as the *field of fractions*, $\mathrm{Frac}(R)$. Therefore $\mathbb{Q}(x_1, \ldots, x_n) \equiv \mathrm{Frac}(\mathbb{Q}[x_1, \ldots, x_n])$ is a field.

Given any field $\mathbb{F}$, a vector space over $\mathbb{F}$ is defined to be a set $V$ equipped with two operations: *vector addition* $(+ : V \times V \to V)$ and *scalar multiplication* $(\cdot : \mathbb{F} \times V \to V)$. These operations must satisfy the following properties:

---

[9] We emphasise that while the coefficients are in $\mathbb{Q}$, no assumptions are made about the variables $x_1, \ldots, x_n$ themselves – they are formally *indeterminates*. We also advise the reader to carefully observe the subtle distinction in notation between a ring of polynomials, which is denoted with *square* brackets, and the associated field of fractions, which is denoted with *round* brackets.

[10] Formally, the elements of the field of fractions are equivalence classes, where given $a, b, c, d \in R$, $a/b$ is equivalent to $c/d$ iff $ad = bc$.

1. $V$ forms an abelian group under vector addition.
2. Scalar multiplication is distributive over vector addition:

$$a \cdot (\mathbf{u} + \mathbf{v}) = a \cdot \mathbf{u} + a \cdot \mathbf{v} \quad \forall a \in \mathbb{F} \text{ and } \mathbf{u}, \mathbf{v} \in V. \tag{1.49}$$

3. Scalar multiplication is distributive over field addition:

$$(a + b) \cdot \mathbf{u} = a \cdot \mathbf{u} + b \cdot \mathbf{u} \quad \forall a, b \in \mathbb{F} \text{ and } \mathbf{u} \in V. \tag{1.50}$$

4. Scalar multiplication and field multiplication are compatible:

$$a \cdot (b \cdot \mathbf{u}) = (a * b) \cdot \mathbf{u} \quad \forall a, b \in \mathbb{F} \text{ and } \mathbf{u} \in V. \tag{1.51}$$

5. The identity element $1 \in \mathbb{F}$ of the field multiplication operation $*$ satisfies:

$$1 \cdot \mathbf{u} = \mathbf{u} \quad \forall \mathbf{u} \in V. \tag{1.52}$$

We will see in Chapter 3 that dimensionally-regulated loop integrals, multiplied by coefficients $c \in \mathbb{Q}(x_1, \ldots, x_K, d)$, where $x_1, \ldots, x_K$ are kinematic invariants and $d$ is the space-time dimension, constitute a vector space over $\mathbb{Q}(x_1, \ldots, x_K, d)$. This will provide a convenient language for discussing loop integrals and the linear relations between them, which are the focus of Chapter 3.

## 1.10 Finite-field techniques

In many particle physics calculations, the final symbolic answers (e.g. the multi-loop integrals or multi-loop amplitudes considered in this thesis) are found to be significantly simpler than the intermediate expressions produced during the calculation. Finite-field techniques present a fruitful means to harness this observation. Finite fields have long been studied by mathematicians, are well-known in the field of computer algebra [34], and have been recently popularised in particle physics by a number of works [35–39]. In this section we will introduce finite fields and explain how they can be employed in our calculations.

The prime field[11] $\mathbb{F}_p$ consists of the set $\{0, 1, 2, \dots, p-1\}$ equipped with the four elementary arithmetic operations: addition, subtraction, multiplication, and division, all of which are performed modulo $p$, a prime number defining the field. Addition, subtraction, and multiplication modulo $p$ are easily understood. Division is the inverse of multiplication: for example in $\mathbb{F}_7$ we have $2 \div 5 \equiv 6 \pmod 7$ because $5 * 6 \equiv 2 \pmod 7$. The result of division modulo $p$ is uniquely defined (as long as $p$ is a prime number) and can easily be computed using the Extended Euclidean Algorithm. Just as with the fields of real and complex numbers, division by zero is undefined in a finite field.

Given a prime number $p$, any rational number $x$ whose denominator is not a multiple of $p$ can be uniquely mapped onto an element $\tilde{x} \in \mathbb{F}_p$. To do so, we write $x = a/b$ (with $a, b \in \mathbb{Z}$ s.t. $p \nmid b$) and then define $\tilde{x} = (a \bmod p) \div (b \bmod p)$, where the symbol '$\div$' indicates division in $\mathbb{F}_p$. Conversely, given $\tilde{x}$, it is possible to uniquely reconstruct $x$ (again using the Extended Euclidean Algorithm), provided that $|a|$ and $|b|$ are known to be bounded from above by $a_{max}$ and $b_{max}$ respectively, with $2a_{max}b_{max} < p$. If the value of $\tilde{x}$ is known in several prime fields $\mathbb{F}_{p_1}, \mathbb{F}_{p_2}, \dots, \mathbb{F}_{p_n}$ then a looser bound suffices: $2a_{max}b_{max} < p_1 p_2 \dots p_n$.

When working with univariate polynomials over the real numbers, it is well known that a polynomial of degree $n$ can be interpolated by sampling its value at $n+1$ points. The same is true for polynomials over a prime field. This can be generalised to multi-variate polynomials and there exist a number of algorithms for performing the reconstruction of multi-variate polynomials (as well as rational functions) in finite fields. In Section 4.3.2 of Chapter 4, we will present an efficient method that allows a polynomial of degree $N$ to be interpolated in $\mathcal{O}(N \log N)$ time, which is significantly better than the standard $\mathcal{O}(N^2)$ or $\mathcal{O}(N^3)$ scaling.

Calculations of amplitudes, as we will see in Chapter 4, produce either rational functions of kinematic variables, or irrational functions whose coefficients are rational functions. Rather than performing the full calculation symbolically, one can use finite fields to perform the calculation at a large number of numerical points and then use rational reconstruction and interpolation to obtain the final answer symbolically. The numerical calculations are fast, even at intermediate stages of the calculation, since all finite-field variables are restricted to lie in the discrete set $\{0, 1, 2, \dots, p-1\}$ and

---

[11] In general, a finite field $\mathbb{F}_{p^k}$ can be defined for any prime number $p$ and positive integer $k$. Finite fields with $k > 1$ find use in many areas including number theory, Galois theory, and cryptography, but they have yet to be applied to the problems of interest in particle physics. In this thesis we will restrict ourselves to *prime fields*, i.e. finite fields with $k = 1$.

so they remain bounded in size throughout the calculation. Unlike floating-point calculations, finite-field calculations have no rounding errors and therefore produce exact answers. The number of numerical evaluations required depends only on the complexity of the final result; it is independent of the complexity of the intermediate results. In Section 4.3 of Chapter 4, we will develop an automated framework that uses finite fields to calculate 2-loop 5-point massless amplitudes and we will apply it to calculate the amplitude $q\bar{q} \to \gamma\gamma\gamma$ at 2 loops in the large-$N_c$ limit of QCD.

## 1.11 Structure of this thesis

The rest of this thesis is organised in the following way. In Chapter 2, we will discuss the theoretical uncertainties arising from the arbitrary choice of value for the renormalisation scale, $\mu_R$. In particular, we will examine the Brodsky-Lepage-Mackenzie/Principle of Maximum Conformality (BLM/PMC) approach for removing the ambiguity over the choice of $\mu_R$ and we will discuss the extent to which the method reduces theoretical uncertainties in perturbative QCD predictions. In Chapter 3, we will study Integration-By-Parts (IBP) identities, which are widely used when calculating the myriad multi-loop integrals appearing in higher-order QCD calculations. We will develop new techniques for solving IBP identities and apply them to the integrals required to calculate the $2 \to 3$ processes that are at the frontier of higher-order QCD calculations. The results are then used in Chapter 4 to perform the first-ever NNLO QCD calculation for a $2 \to 3$ process: 3-photon production at the LHC.

# Chapter 2

# Ambiguities of the BLM/PMC renormalisation scale-setting procedure

In any calculation in perturbative Quantum Chromodynamics (QCD) a choice needs to be made for the unphysical renormalisation scale, $\mu_R$. The Brodsky-Lepage-Mackenzie/Principle of Maximum Conformality (BLM/PMC) scale-setting procedure is one proposed method for selecting this ambiguous scale. In this chapter, we identify three ambiguities in the BLM/PMC procedure itself. Their numerical impact is studied using the example of the total cross-section for $t\bar{t}$ production through Next-to-Next-to-Leading Order in QCD. One ambiguity is the arbitrary choice of the value of the highest-order PMC scale. The numerical impact of this choice on the BLM/PMC prediction for the cross-section is found to be comparable to the impact of the choice of $\mu_R$ in the conventional scale-setting approach. Another ambiguity relates to the definitions of the other PMC scales and is similarly found to have a large impact on the cross-section.

## Declaration

This chapter is based on the paper [1], which I wrote jointly with A. Mitov. Section 2.2 of this chapter is based on the Part III Project Report that I submitted as part of my MSci degree at the University of Cambridge in 2016.

## 2.1 Introduction

When performing calculations in Quantum Chromodynamics (QCD), any partonic observable $\rho$ is usually calculated as a perturbative series in the strong coupling constant $\alpha_S$:

$$\rho = \sum_n c_n(\mu_R) \left( \frac{\alpha_S(\mu_R)}{4\pi} \right)^n . \tag{2.1}$$

The renormalisation scale $\mu_R$ is an arbitrary parameter which enters this equation following renormalisation, as discussed in Section 1.3 of Chapter 1. Formally, when working to all orders in $\alpha_S$, the $\mu_R$-dependence of the coefficients $c_n$ exactly compensates that of $\alpha_S$ so that $\rho$ is independent of $\mu_R$. In practice, however, the perturbative series is truncated beyond some finite order, $N$, and this causes $\rho$ to become $\mu_R$-dependent.

Conventionally, in processes with a single hard scale $Q$, one chooses $\mu_R = Q$ on dimensional grounds.[1] The value of $\mu_R$ is then varied in a range $[Q/2, 2Q]$ and the resulting variation in the value of $\rho$ is taken to be representative of the error which arises from omitting the $\mathcal{O}\left(\alpha_S^{N+1}\right)$ terms from Equation 2.1. While the choice of this variation range is a matter of convention, its adequacy is justified *a posteriori* by higher-order calculations.

The Brodsky-Lepage-Mackenzie/Principle of Maximum Conformality (BLM/PMC) method [58–60] has been proposed as a way of removing the renormalisation scale ambiguity. The method is based on an appealing physical motivation and, as explained in Section 2.2, it algorithmically prescribes a “correct” value for the scale $\mu_R$. The method has been applied to a number of processes including Higgs production [61], meson production [62–65], pion form factors [66, 67], $b$-physics [68, 69], and $t\bar{t}$ production [70–75]. Some possible generalisations of the BLM/PMC method have been discussed in Refs. [76–80].

In this chapter we address the following question: *are there any ambiguities associated with the BLM/PMC method and what is their numerical impact?*

Since scale variations are usually interpreted as representing theory uncertainties, the BLM/PMC method might appear to eliminate uncertainties in theoretical predictions. In Sections 2.2 and 2.4, we will discuss the extent to which this is true.

---

[1] Although not essential for the goals of the present work, we would like to mention that more refined arguments for choosing this scale have been given in the literature [40–57]. Such arguments tend to modify the choice $\mu_R = Q$ by a factor of $\mathcal{O}(1)$ and are especially relevant for observables with several kinematic scales.

In order to keep our discussion less abstract we will consider the process of top-pair production at hadron colliders, which is well-suited for this study given that it is fully known through NNLO, has generic kinematics and colour structure, and is very precisely measured. The application of the BLM/PMC method to this process has been extensively studied [70–75]. We expect that many of our findings transcend this particular process.

## 2.2 The BLM/PMC procedure at NNLO

One applies the BLM/PMC method to a partonic observable $\rho$ like the one in Equation 2.1. If this is a hadron collider observable, as we will be considering here, then two qualifications are required.

Firstly, in order to construct the proper hadron-level observable, Equation 2.1 needs to be convolved with parton distribution functions (PDFs) and summed over all possible initial partonic states. Such partonic observables are not uniquely defined since they depend on the scheme used to subtract collinear singularities; we will not be concerned with this here and will assume a given factorisation scheme (the $\overline{\text{MS}}$ scheme is standard).

Secondly, the perturbative coefficients $c_n$ also depend on the unphysical factorisation scale $\mu_F$, which separates the long-distance physics absorbed into the PDFs from the short-distance physics in the perturbative coefficients $c_n$. The BLM/PMC method does not prescribe a value for $\mu_F$. In this chapter we will focus exclusively on the scale $\mu_R$ and will fix the factorisation scale at some standard value, as was also done in the previous BLM/PMC work on the subject [70–75]. For the total top-pair cross-section this is $\mu_F = m_t$ (although a smaller value $\mu_F = m_t/2$ may be more appropriate [55]). In the following we will suppress the explicit dependence of the coefficients $c_n$ and the observable $\rho$ on the partonic channel and factorisation scale.

The idea behind the BLM/PMC method is to first identify the terms proportional to the QCD $\beta$-function coefficients $\beta_i$ inside the partonic coefficients $c_n$ and absorb them into the running coupling by making a suitable choice for the renormalisation scale $\mu_R$.

At Next-to-Leading Order (NLO) in QCD, one can use the BLM method [58] and, for any given process, uniquely fix the value of $\mu_R$ by requiring the LO and NLO perturbative coefficients $c_n$ to be independent of $\beta_i$. As it turns out, however, beyond NLO one cannot absorb all $\beta_i$ coefficients into the running coupling with a single choice of scale. The PMC method [81] extends the BLM idea to higher-order QCD calculations by using a different value for the renormalisation scale at each order in $\alpha_S$. After such a choice the partonic observable $\rho$ in Equation 2.1 takes the form:

$$\rho = \sum_n \tilde{c}_n \left( \frac{\alpha_S(q_n)}{4\pi} \right)^n , \tag{2.2}$$

for some new coefficients $\tilde{c}_n$ and scales $q_n$. These new scales and coefficients are chosen by requiring that

1. the coefficients $\tilde{c}_n$ are independent of $\beta_i$ , and
2. Equations 2.1 and 2.2 agree through order $\alpha_S^N$ .

It is convenient to express the coefficients $c_n$ in Equation 2.1 through a new set of $\beta_i$-independent coefficients $s_{n,k}(\mu_R)$. As mentioned above, in the rest of this paper we will specialize our discussion to the inclusive cross-section for top-quark pair production. This means that the sum in Equations 2.1 and 2.2 goes from $n = 2$ (the LO term) through $n = N = 4$ (the NNLO term). In this context the coefficients $s_{n,k}(\mu_R)$ are defined by means of the following implicit equations:

$$\begin{aligned} c_2 &= s_{2,0} , \\ c_3 &= s_{3,0} + 2 s_{3,1} \beta_0 , \\ c_4 &= s_{4,0} + 2 s_{3,1} \beta_1 + 3 s_{4,1} \beta_0 + 3 s_{4,2} \beta_0^2 . \end{aligned} \tag{2.3}$$

The $\mu_R$-dependence of the coefficients $s_{n,k}$ follows from the requirement that observables are independent of $\mu_R$. In particular, one finds that $s_{n,0}$ have no dependence on $\mu_R$.

We remark on a practical aspect of the procedure outlined above. The $\beta_i$ dependence is inferred from the known $n_f$ dependence of the cross-section by inverting the dependence of $\beta_i$ on $n_f$:

$$\beta_0 = 11 - \frac{2}{3} n_f , \quad \beta_1 = 102 - \frac{38}{3} n_f . \tag{2.4}$$

The above procedure requires the exclusion of $n_f$ contributions from light-by-light type of diagrams that are not associated with coupling renormalisation. In the process at hand, no light-by-light contribution is present in the $q\bar{q}$-initiated contribution. The $gg$-initiated contribution does contain such diagrams at NNLO but these contributions have not been separated in the existing literature. We thus neglect to separate them in this work. To the best of our knowledge they have likewise not been separately accounted for in the previous applications of the BLM/PMC method to top-quark pair production.

The authors of the BLM/PMC method [81] define the PMC coefficients $\tilde{c}_n$ and scales $q_n$ in the following way:

$$\tilde{c}_n = s_{n,0}\,, \tag{2.5}$$

$$\log\left(\frac{q_2^2}{\mu_0^2}\right) = -\frac{s_{3,1}}{s_{2,0}} + \frac{3}{2}\left[\left(\frac{s_{3,1}}{s_{2,0}}\right)^2 - \frac{s_{4,2}}{s_{2,0}}\right]\beta_0\frac{\alpha_S(\mu_*)}{4\pi}\,, \tag{2.6}$$

$$\log\left(\frac{q_3^2}{\mu_0^2}\right) = -\frac{s_{4,1}}{s_{3,0}}\,, \tag{2.7}$$

where $s_{n,k} = s_{n,k}(\mu_0)$.

Two new scales, $\mu_0$ and $\mu_*$, appear in Equations 2.6 and 2.7. The $\mu_0$-dependence of Equations 2.6 and 2.7 is purely formal: it can be shown that the scales $q_2$ and $q_3$ are completely independent of $\mu_0$. In other words, the $\mu_0$ dependence of the functions $s_{n,k}$ is such that all $\mu_0$ dependence in Equations 2.6 and 2.7 cancels between the two sides of those equations. Equation 2.6 also depends on the scale $\mu_*$, whose value is arbitrary. This is so since a change in $\mu_*$ only affects the relation $c_n \leftrightarrow \tilde{c}_n$ with terms beyond NNLO.

One can use Equation 1.29 to verify that the scales $q_2$ and $q_3$ as defined in Equations 2.5 and 2.6 satisfy the above-stated requirement that Equations 2.1 and 2.2 agree through order $\alpha_S^N$.

The term in the square bracket in Equation 2.6 vanishes for observables that respect the so-called large-$\beta_0$ approximation. As follows from Refs. [82–85] this is not the case for top-quark pair production. In the $q\bar{q}$ partonic reaction (introduced in Section 2.3 below) the square bracket term is a pure number (see the related discussion in Ref. [86]) while the corresponding result for the $gg$ reaction is only known as a precise numeric fit [85], rather than analytically.

Following Ref. [71], one also needs to subtract the so-called “Coulomb” terms from all functions $s_{n,k}$ that enter Equations 2.6 and 2.7 for all partonic reactions that contain such terms. The subtraction procedure of the Coulomb terms in top-quark pair production is explained in detail in Section 2.3 below.

Clearly the choice of the scale $\mu_*$ does have an impact on the values of the scales $q_n$ and this represents one ambiguity in the PMC procedure. We find its numerical impact to be small and we suspect it is responsible for the very small "initial renormalisation scale dependence" reported in Ref. [72]. When presenting numerical results in Section 2.4, we will therefore focus on two other ambiguities, which we will now describe and whose numerical impact is larger.

We note that the PMC scales $q_n$ defined above are not the only way to absorb the $\beta_i$-dependence of the coefficients $c_n$ into the running coupling. For example, one could modify Equations 2.6 and 2.7 by defining an alternative set of scales $q'_n$:

$$\log\left(\frac{q_2'^2}{\mu_R^2}\right) = -\frac{s_{3,1}}{s_{2,0}}\,, \tag{2.8}$$

$$\log\left(\frac{q_3'^2}{\mu_R^2}\right) = -\frac{s_{4,1}}{s_{3,0}} + \frac{s_{2,0}}{s_{3,0}}\left[\left(\frac{s_{3,1}}{s_{2,0}}\right)^2 - \frac{s_{4,2}}{s_{2,0}}\right]\beta_0\,, \tag{2.9}$$

which have the advantage of not containing the arbitrary scale $\mu_*$. As before, one can use Equation 1.29 to verify that the scales $q'_n$ satisfy the requirement that Equations 2.1 and 2.2 agree through order $\alpha_S^N$. Clearly, the choice of whether to work with the scales $q_n$ or $q'_n$ represents a second ambiguity in the application of the PMC procedure.[2] In what follows, we carry out our calculations using the original scales $q_n$ as well as the alternative scales $q'_n$ and explore the numerical difference between the two.

The last PMC scale, $q_4$, which appears at NNLO remains arbitrary at this order. Its fixing requires the knowledge of the $\beta_i$-dependent terms in the N$^3$LO coefficient functions for $t\bar{t}$ production. These are not available at present. The arbitrariness of the scale $q_4$ represents the third, and most significant, ambiguity which we have identified in the PMC procedure. In Ref. [81], the choice is made to set it equal to the previous known scale, $q_3$. While this is a plausible choice, we are not aware of a motivation in its favour. In what follows, in order to illustrate the significance of this ambiguity, we explore two choices: $q_4 = q_3$ and $q_4 = m_t$, where $m_t$ is the (pole) mass of the top quark.

We wish to make one remark on the subject of theoretical uncertainties. The BLM/PMC framework asserts that there is a unique "correct" way of choosing the renormalisation scale, and that one should not try to estimate theoretical uncertainties by varying this scale in the manner described in Section 2.1. Nevertheless, we emphasise that the "renormalisation-scale uncertainty" conventionally quoted in perturbative QCD predictions is only a proxy for the error arising from the truncation of the sum in Equation 2.1. Prescribing a procedure to choose $\mu_R$ may remove the way to estimate

---

[2] Arguments in favour of using $q_n$ have been given in Ref. [81].

this error but it cannot remove the error itself, even in the absence of any ambiguities in the scale-setting procedure.[3]

## 2.3 Details about the implementation

All partonic contributions to the total inclusive NNLO cross-section for $t\bar{t}$ production have been calculated in Refs. [82–85], keeping their $n_f$-dependence explicit. As explained above Equation 2.4, we convert this $n_f$-dependence into a dependence on the coefficients $\beta_i$. The factorisation scale is set to $m_t$ in all partonic reactions. The value of the renormalisation scale for each partonic reaction is different, according to what is prescribed for it by the PMC approach. In fact we apply the PMC procedure only to the two dominant partonic channels $gg \to t\bar{t} + X$ and $q\bar{q} \to t\bar{t} + X$. All other contributing partonic reactions are included, as appropriate, only in the predictions for the complete hadron-level cross-section. For these sub-dominant channels, the standard choice $\mu_R = \mu_F = m_t$ is made.

In addition to depending on $\mu_R$ and $\mu_F$, the partonic cross-section coefficients $c_n$ also depend on $m_t$ and the partonic centre-of-mass energy $\hat{s}$ through the following variable:

$$v = \sqrt{1 - 4m_t^2/\hat{s}}\,. \tag{2.10}$$

As mentioned in the previous section, in order to derive the PMC scales $q_2, q_3$ (or $q_2', q_3'$), in each of the $q\bar{q}$ and $gg$ partonic reactions we first subtract the "Coulomb" terms from the functions $s_{3,0}$ and $s_{4,1}$. The explicit expressions for these Coulomb terms can be found in Ref. [87]. The Coulomb terms in the function $s_{4,0}$ are not subtracted since they do not enter the scales $q_2$ and $q_3$ (nor $q_2'$ and $q_3'$) through NNLO. The Coulomb terms are identified as the terms proportional to $1/v$ or $\log(v)/v$ in the series expansion of the functions $s_{3,0}/v$ and $s_{4,1}/v$ around $v = 0$. The subtracted Coulomb contributions include terms $\sim \log(\mu_R)$, as appropriate.

We only apply PMC scale-setting to the part of the partonic cross-section remaining after the subtraction of the Coulomb terms. The Coulomb terms are then added back. Since they constitute only a small part of the partonic cross-section, we do not apply the PMC procedure to the Coulomb terms themselves.

---

[3] In principle, a scale-setting procedure could improve the convergence of a perturbative series. As we will see in Section 2.4, there is no such improvement for this particular procedure and process.

The reason for the separate treatment of the Coulomb terms is that at sufficiently high orders, the integrability of the cross-section requires their factorisation into a toponium-like wave-function. A detailed analysis can be found in Ref. [88].

We find that the subtraction of the Coulomb terms has a large impact on the PMC scales $q_2$ and $q_3$: in fact, failure to subtract the Coulomb terms leads to $q_3 \ll \Lambda_{\rm QCD}$ and hence a divergent cross-section.

Finally, in our numerical predictions for the hadronic $t\bar{t}$ cross-section we use the PDF set `NNPDF3.1` [13] and we set $m_t = 173.3$ GeV. We have verified that the PDF set `CT14` [89] produces similar results to those shown here. We base our numerical calculations on a modified version of the program `Top++` [90].

## 2.4 Results and Discussion

### 2.4.1 PMC scales for $gg$ and $q\bar{q}$ channels

Applying the formulae from Section 2.2, we now derive the PMC scales for the $gg$ and $q\bar{q}$ channels. We remind the reader that these scales depend on the parton level kinematics (through the variable $v$ for the case of the inclusive $t\bar{t}$ cross-section) but are independent of PDFs and, by extension, of the type of collider ($pp$ versus $p\bar{p}$) or collider energy. The results are shown in Figs. 2.1 and 2.2 for, respectively, the $gg$ and $q\bar{q}$ channels.

For the $q\bar{q}$ channel (Fig. 2.2), it is interesting to observe that in the kinematic region $v \in [0.7, 0.8]$, the scale $q_3$ reaches values as low as 4.6 GeV. In fact, without the Coulomb subtraction procedure outlined in Section 2.3, $q_3$ takes values below $10^{-10}$ GeV in this kinematic region. Similar singularities have previously been found when applying BLM scale-setting in vector meson production [64]. Since the separation of the Coulomb terms is motivated by the physics of $t\bar{t}$ production in the kinematic limit $v \to 0$, it is surprising that their treatment should so strongly affect the behaviour of the PMC scales in the region $v \in [0.7, 0.8]$.

### 2.4.2 Cross-sections for $t\bar{t}$ production at LHC13 and Tevatron

Having derived the PMC scales for the $gg$ and $q\bar{q}$ channels, we will now calculate hadron-level cross sections at the 13 TeV LHC and also at the Tevatron. For each collider, we will compare the results from PMC scale-setting to those from the conventional approach ($\mu_R = m_t$). For the latter, uncertainties are computed by varying $\mu_R$ in the

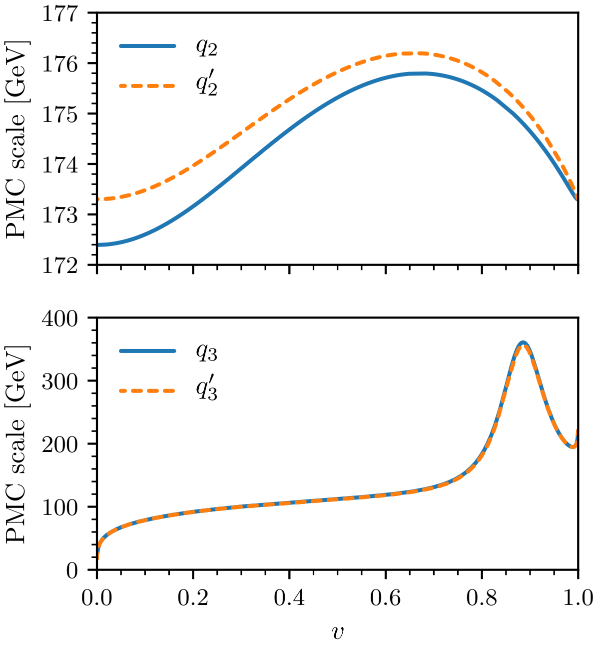


**Fig. 2.1** The PMC scales $q_2$, $q_2'$, $q_3$, $q_3'$ for the $gg$ channel as functions of the relative velocity of the final-state top quarks.

range $(m_t/2, 2m_t)$.[4] We remind the reader that throughout this chapter we have fixed $\mu_F = m_t$.

In this section, we use the "standard" choice of PMC scales, i.e. the scales $q_2$ and $q_3$ as defined in Equations 2.6 and 2.7, and setting $q_4 = q_3$. The effect of alternative choices will be explored in Section 2.4.3.

At the 13 TeV LHC, using the BLM/PMC method, we obtain the following prediction for the total hadron-level cross-section for $pp \to t\bar{t} + X$:

$$\sigma_{\text{BLM/PMC}} = 813 \text{ pb}\,. \tag{2.11}$$

For comparison, the predicted cross-section using conventional scale-setting is:

$$\sigma_{\text{Conventional}} = 794^{+28}_{-39} \text{ pb}\,, \tag{2.12}$$

---

[4] No numerical estimate is made for the theoretical uncertainty in the BLM/PMC predictions, as explained at the end of Section 2.2.

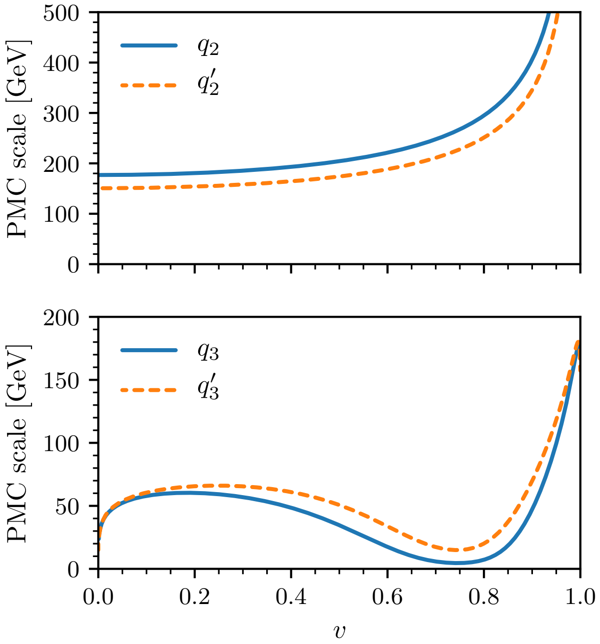


**Fig. 2.2** As in Fig. 2.1 but for the $q\bar{q}$ channel.

**Table 2.1** Contribution of the $q\bar{q}$ and $gg$ channels to $\sigma_{pp\to t\bar{t}+X}$ at the 13 TeV LHC at each order in $\alpha_s$.

| | $q\bar{q}$ channel | | $gg$ channel | |
|---|---|---|---|---|
| | PMC | Conv. | PMC | Conv. |
| $\alpha_s^2$ [pb] | 62.4 | 68.5 | 405.7 | 406.9 |
| $\alpha_s^3$ [pb] | 41.7 | 8.5 | 256.4 | 220.8 |
| $\alpha_s^4$ [pb] | −32.3 | 4.7 | 76.4 | 81.5 |
| NNLO [pb] | 71.8 | $81.8^{+1.9}_{-2.2}$ | 738.4 | $709.2^{+28.1}_{-37.2}$ |

and the most recent precise experimentally-measured values from ATLAS [91] and CMS [92] are:

$$\sigma_{\text{ATLAS}} = 818 \pm 8 \pm 27 \pm 19 \pm 12 \text{ pb}\,, \tag{2.13}$$

$$\sigma_{\text{CMS}} = 803 \pm 2 \pm 25 \pm 20 \text{ pb}\,. \tag{2.14}$$

At the Tevatron, we find the following BLM/PMC prediction for the total $p\bar{p} \to t\bar{t} + X$ cross-section:

$$\sigma_{\text{BLM/PMC}} = 6.48 \text{ pb}\,. \tag{2.15}$$

**Table 2.2** As in Table 2.1 but for $\sigma_{p\bar{p}\to t\bar{t}+X}$ at the Tevatron.

| | $q\bar{q}$ channel | | $gg$ channel | |
|---|---|---|---|---|
| | PMC | Conv. | PMC | Conv. |
| $\alpha_s^2$ [pb] | 4.55 | 4.89 | 0.39 | 0.39 |
| $\alpha_s^3$ [pb] | 3.31 | 0.96 | 0.41 | 0.33 |
| $\alpha_s^4$ [pb] | $-2.24$ | 0.42 | 0.19 | 0.18 |
| NNLO [pb] | 5.62 | $6.27^{+0.16}_{-0.20}$ | 0.98 | $0.91^{+0.07}_{-0.07}$ |

For comparison, the cross-section using conventional scale-setting is:

$$\sigma_{\text{Conventional}} = 7.06^{+0.21}_{-0.25}\text{ pb}\,, \tag{2.16}$$

and the experimentally-measured value [93] is:

$$\sigma_{\text{Experimental}} = 7.60 \pm 0.41\text{ pb}\,. \tag{2.17}$$

To examine the origin of these values, the contributions of the two dominant partonic channels ($gg$ and $q\bar{q}$) to these cross-sections are shown in Table 2.1 for the LHC and Table 2.2 for the Tevatron. In each case, a breakdown is provided, showing the contributions from each power of $\alpha_S$.

In both tables it can be seen that the BLM/PMC procedure leads to a slower convergence than in conventional scale-setting. Similar behaviour has previously been discussed in Refs. [76, 94, 95].

### 2.4.3 Effect of ambiguities

We next explore the effects of the ambiguities in the BLM/PMC procedure which were outlined in Section 2.2. In order to do so, we recompute the above cross-sections using a variety of choices for the PMC scales $(q_2, q_3, q_4)$:

1. $(q_2, q_3, q_3)$ Firstly, we restate the results using the "standard" choice of PMC scales which were used in the previous section.

2. $(q_2, q_3, m_t)$ Secondly, we study the numerical impact of the arbitrary choice of the scale $q_4$ by setting $q_4 = m_t$ rather than $q_4 = q_3$.

3. $(q_2', q_3', q_3')$ Thirdly, we explore the other main ambiguity discussed in Section 2.2, by using the scales $q_n'$ rather than $q_n$.

**Table 2.3** The $gg$ channel's contribution to the LHC13 cross–section for various PMC scale choices.

| | PMC | | | Conv. |
|---|---|---|---|---|
| | $(q_2, q_3, q_3)$ | $(q_2, q_3, m_t)$ | $(q_2', q_3', q_3')$ | $m_t$ |
| $\alpha_s^2$ [pb] | 405.7 | 405.7 | 405.4 | 406.9 |
| $\alpha_s^3$ [pb] | 256.4 | 256.4 | 256.7 | 220.8 |
| $\alpha_s^4$ [pb] | 76.4 | 53.8 | 76.4 | 81.5 |
| NNLO [pb] | 738.4 | 715.9 | 738.5 | $709.2^{+28.1}_{-37.2}$ |

**Table 2.4** As in Table 2.3 but for the $q\bar{q}$ channel.

| | PMC | | | Conv. |
|---|---|---|---|---|
| | $(q_2, q_3, q_3)$ | $(q_2, q_3, m_t)$ | $(q_2', q_3', q_3')$ | $m_t$ |
| $\alpha_s^2$ [pb] | 62.4 | 62.4 | 65.1 | 68.5 |
| $\alpha_s^3$ [pb] | 41.7 | 41.7 | 28.4 | 8.5 |
| $\alpha_s^4$ [pb] | −32.3 | −5.2 | −14.8 | 4.7 |
| NNLO [pb] | 71.8 | 98.9 | 78.7 | $81.8^{+1.9}_{-2.2}$ |

4. ($m_t$) For the purposes of comparison, we also present the results using the conventional choice $\mu_R = m_t$.

The contribution of the $gg$ channel to the 13 TeV LHC cross-section, as predicted by each of these choices of scales, is shown in Table 2.3. Similarly, the contribution of the $q\bar{q}$ channel is shown in Table 2.4. The total cross-section, incorporating the contributions from all partonic channels, is shown in Table 2.5, where alongside the LHC results, we also provide results for the Tevatron.

The ambiguity over whether to choose the scales $q_n$ or the scales $q_n'$ has effects that vary in size between partonic channels. In the $gg$ channel, where $q_n' \approx q_n$ (see Fig. 2.1), the scales $q_n'$ produce similar results to the scales $q_n$, as can be seen in Table 2.3. In

**Table 2.5** Total hadronic cross-section (including all partonic channels) through NNLO.

| | LHC13 | Tevatron |
|---|---|---|
| $\sigma_{\text{PMC}}[q_2, q_3, q_3]$ | 813 | 6.48 |
| $\sigma_{\text{PMC}}[q_2, q_3, m_t]$ | 818 | 8.30 |
| $\sigma_{\text{PMC}}[q_2', q_3', q_3']$ | 820 | 6.97 |
| $\sigma_{\text{Conventional}}[m_t]$ | $794^{+28}_{-39}$ | $7.06^{+0.21}_{-0.25}$ |
| $\sigma_{\text{Experimental}}$ | $818 \pm 36$ [ATLAS]<br>$803 \pm 32$ [CMS] | $7.60 \pm 0.41$ |

the $q\bar{q}$ channel, however, the scales $q'_n$ differ more substantially from $q_n$ (see Fig. 2.2) and the impact on the cross-section is therefore larger, as shown in Table 2.4. The effect of this ambiguity on the overall cross-section, shown in Table 2.5, is therefore more significant at the Tevatron (where the $q\bar{q}$ channel dominates) than at the LHC (where the $gg$ channel dominates).

The ambiguity over the choice of $q_4$ has a large impact on the value of the cross-section in both of the dominant partonic channels (see Tables 2.3 and 2.4). We note that the numerical impact of the choice of $q_4$ on the BLM/PMC predictions is comparable to that of the choice of $\mu_R$ on the conventional predictions. When the contributions from all partonic channels are combined into a hadron-level cross-section (Table 2.5), the effect of the $q_4$ ambiguity somewhat cancels between channels in the LHC cross-section, but is significantly larger in the Tevatron cross-section. In principle, the BLM/PMC method does prescribe a value for $q_4$, but it requires information from the currently unknown N$^3$LO cross-section. Note, however, that a new arbitrary scale, $q_5$, would appear at N$^3$LO — any calculation using the BLM/PMC method will always involve one arbitrary scale.

### 2.4.4 Comparison of strategies to handle the $q_4$ ambiguity

It was found in the previous section that the ambiguity over the highest-order scale, $q_4$, has a significant impact on the prediction for the cross-section. In the literature describing the BLM/PMC method, it is suggested [59, 81] that the ambiguity could have been resolved if we had information from the next perturbative order in $\alpha_S$. In this section, we will explore 4 ways of handling the ambiguity over the highest-order scale, including the suggested approach of "peeking" at the next perturbative order. We choose to work with the NLO cross-section, allowing us the possibility to "peek" at the NNLO cross-section when setting the PMC scales.

Only 2 scales appear in the NLO cross-section: $q_2$ and $q_3$. At this order in perturbation theory, $q_3$ is arbitrary since it relies on information appearing in the NNLO cross-section (see Equations 2.7 and 2.9). We will calculate the NLO cross-section while exploring the following possible choices for the PMC scales ($q_2$, $q_3$):

1. ($q'_2$, $q'_2$) Of the PMC scales defined in Section 2.2, the only one that does not require information from the NNLO calculation is the scale $q'_2$ (see Equation 2.8). (In fact, $q'_2$ was the scale prescribed in the original BLM paper [58].) Hence, one option is to set both PMC scales to be $q'_2$.

**Table 2.6** The $gg$ channel's contribution to the LHC13 cross–section at NLO using various scale choices

| | PMC | | | | Conv. |
|---|---|---|---|---|---|
| | $(q_2', q_2')$ | $(q_2', m_t)$ | $(q_2', q_3')$ | $(q_2, q_3)$ | $m_t$ |
| $\alpha_s^2$ [pb] | 405.4 | 405.4 | 405.4 | 405.7 | 406.9 |
| $\alpha_s^3$ [pb] | 221.3 | 222.3 | 256.7 | 256.4 | 220.8 |
| NLO [pb] | 626.7 | 627.7 | 662.1 | 662.1 | $627.7^{+67.6}_{-63.6}$ |

**Table 2.7** As in Table 2.6 but for the $q\bar{q}$ channel.

| | PMC | | | | Conv. |
|---|---|---|---|---|---|
| | $(q_2', q_2')$ | $(q_2', m_t)$ | $(q_2', q_3')$ | $(q_2, q_3)$ | $m_t$ |
| $\alpha_s^2$ [pb] | 65.1 | 65.1 | 65.1 | 62.4 | 68.5 |
| $\alpha_s^3$ [pb] | 11.2 | 12.2 | 28.4 | 41.7 | 8.5 |
| NLO [pb] | 76.3 | 77.2 | 93.5 | 104.1 | $77.0^{+1.3}_{-3.9}$ |

2. $(q_2',\, m_t)$ To explore the impact of the arbitrary scale $q_3$ without relying on any NNLO information, we can set $q_3 = m_t$ and compare against the results of the previous scale choice.

3. $(q_2',\, q_3')$ If we allow ourselves to peek at the NNLO cross-section, we can use the full NNLO PMC scales $q_2'$ and $q_3'$ defined in Equations 2.8 and 2.9.

4. $(q_2,\, q_3)$ Alternatively, again peeking at the NNLO cross-section, we could choose to use the scales $q_n$ (defined in Equations 2.6 and 2.7) rather than $q_n'$.

The resulting contributions of the $gg$ and $q\bar{q}$ channels to the LHC cross-section are shown in Tables 2.6 and 2.7 respectively. The total NLO cross-section, incorporating the contributions from all partonic channels, is shown in Table 2.8, where alongside the LHC results, we also provide results for the Tevatron.

Comparing the choices $(q_2', q_2')$, $(q_2', m_t)$, and $(q_2', q_3')$, one sees that the numerical impact of the choice of $q_3$ on the PMC prediction can be similar to the impact of the choice of $\mu_R$ on the conventional prediction. This is analogous to the findings of the previous section in relation to the $q_4$ ambiguity at NNLO.

We note that the scale choices $(q_2', q_2')$ and $(q_2', m_t)$ — obtained using only information available at NLO — lead to very different cross-sections compared to the scale choices $(q_2', q_3')$ and $(q_2, q_3)$, which were obtained by peeking at the next perturbative order. In other words, when handling the ambiguous highest-order PMC scale, the two approaches appearing in the literature (to either use an existing PMC scale or instead

**Table 2.8** Total hadronic cross-section (including all partonic channels) through NLO. For NNLO, see Table 2.5.

| | LHC13 | Tevatron |
|---|---|---|
| $\sigma_{\text{PMC}}[q_2', q_2']$ | 709 | 6.52 |
| $\sigma_{\text{PMC}}[q_2', m_t]$ | 711 | 6.51 |
| $\sigma_{\text{PMC}}[q_2', q_3']$ | 762 | 7.86 |
| $\sigma_{\text{PMC}}[q_2, q_3]$ | 773 | 8.59 |
| $\sigma_{\text{Conventional}}[m_t]$ | $711^{+71}_{-69}$ | $6.51^{+0.30}_{-0.44}$ |
| $\sigma_{\text{Experimental}}$ | $818 \pm 36$ [ATLAS]<br>$803 \pm 32$ [CMS] | $7.60 \pm 0.41$ |

peek at the next perturbative order) yield very different results to one another, as well as to other plausible choices for this scale. The arbitrary choice of a value for the highest-order PMC scale thus remains an open problem.

## 2.5 Conclusions

The BLM/PMC procedure is a proposed method for eliminating the renormalisation scale ambiguity in perturbative QCD. In this chapter, we have presented three ambiguities in the BLM/PMC procedure itself. We have studied these ambiguities using the example of $t\bar{t}$ production at NNLO in QCD and have found two of the ambiguities to have a significant numerical impact on the computed cross-sections.

One of these ambiguities lies in the definition of the PMC scales $q_n$: we give an example of an alternative set of scales, $q_n'$, which satisfy the PMC requirement that terms proportional to the QCD $\beta$-function coefficients are to be absorbed into the running coupling. The other ambiguity arises because in any calculation employing the BLM/PMC scale-setting procedure, the highest-order scale (in this case, $q_4$) remains arbitrary. We find the numerical impact of each of these ambiguities to be comparable to the impact of the choice of $\mu_R$ in the conventional scale-setting approach.

In the existing literature on the BLM/PMC method, it is asserted that the $q_4$ ambiguity could in principle be resolved using information from even higher perturbative orders, and that it should otherwise by handled by using an existing PMC scale. We find that the cross-sections arising within these two approaches can differ markedly from one another, as well as from the cross-sections arising from other plausible choices for this scale.

In summary, while the BLM/PMC procedure is well-motivated, it contains important ambiguities with significant numerical impact on the predicted values for physical observables. We also emphasise that even an unambiguous scale-setting prescription would not remove the theoretical uncertainties in physical predictions, since these uncertainties ultimately arise from missing higher orders in $\alpha_S$. We hope our work will lead to an improved understanding of the problem of scale settings which, in turn, should result in improved higher-order predictions for QCD processes.

# Chapter 3

# IBPs for 2-loop 5-point massless QCD amplitudes

In this chapter, we introduce a novel strategy for solving integration-by-parts (IBP) identities, which are widely used in the computation of multi-loop QCD amplitudes. We implement the strategy in an efficient `C++` program and hence solve the IBP identities needed for the computation of any planar 2-loop 5-point massless amplitude in QCD. We also derive representative results for the most complicated non-planar integral family.

## Declaration

Sections 3.1 and 3.3 of this chapter are based on the paper [2], which I wrote in collaboration with M. A. Lim and A. Mitov. The strategy described in Section 3.3 was conceived by A. Mitov. The design and implementation of the IBP-solving `C++` program, described in Section 3.5, are my own.

## 3.1 Introduction

Multi-loop integrals are core ingredients in higher-order QCD calculations. These integrals are used to evaluate multi-loop amplitudes, which provide virtual corrections to processes in perturbative QCD. The evaluation of multi-loop integrals is often the biggest bottleneck in the calculation of virtual QCD corrections. Since these integrals can contain numerous divergences, a direct numerical evaluation is typically not possible. Instead, calculations must to some extent be performed analytically, with divergences usually being regulated with the help of dimensional regularisation. In this chapter, we present our work on solving the integrals contributing to 2-loop 5-point massless QCD amplitudes. We make use of integration-by-parts (IBP) identities, which will be described shortly. We solve the IBP identities with the help of a novel strategy, explained in Section 3.3, and an efficient `C++` program which we describe in detail in Section 3.5. The strategy and the `C++` program are generic and we hope that they can aid the solving of IBPs for other amplitudes beyond the 2-loop 5-point case considered here. The results obtained in this chapter enabled the calculation of NNLO QCD corrections to 3-photon production, which is the first NNLO QCD calculation for a $2 \to 3$ process and will be presented in Chapter 4.

The IBP approach [96, 97] has long been the method of choice for computing multi-loop QCD amplitudes. The method has produced countless results; some recent reviews can be found in Refs. [98, 99]. The IBP approach can be described in the following way. A generic squared or suitably decomposed multi-loop UV-unrenormalised amplitude can be written as

$$M = \sum_{i=1}^{N} f_i I_i \,. \tag{3.1}$$

The above expression follows from a straightforward application of Feynman rules applied to the process at hand and, if appropriate, after summation over spin and/or colour. Throughout this work we assume that all divergences are regulated by working in $d = 4-2\epsilon$ dimensions, as discussed in Section 1.6 of Chapter 1. The coefficients $f_i$ are rational functions of kinematic invariants $\{x_1, \ldots, x_K\}$ and the space-time dimension, while $I_i$ are scalar Feynman integrals. The number $N$ of such integrals tends to be very large and grows quickly with the number of loops, legs, and parameters in the problem.

As will be explained in Section 3.2, the IBP approach makes it possible to express the Feynman integrals $I_i$ appearing in Equation 3.1 as linear combinations of a small number of Feynman integrals $\hat{I}_m$

$$I_i = \sum_{m=1}^{\hat{N}} c_{i,m} \hat{I}_m \,. \tag{3.2}$$

The integrals $\hat{I}_m$ are known as master integrals (or simply masters) and the coefficients $c_{i,m}$ are rational functions of the kinematic invariants and the space-time dimension, i.e. $c_{i,m} \in \mathbb{Q}(x_1, \ldots, x_K, d)$. Thus, in the language of Section 1.9, the integrals $I_i$ and the amplitude $M$ lie in a vector space over $\mathbb{Q}(x_1, \ldots, x_K, d)$, for which the master integrals $\hat{I}_m$ form a basis. We will denote this vector space $V$. The utility of the IBP approach stems from the fact that the dimension of $V$ is small: $\hat{N} \ll N$. For example, for the processes that we consider in this chapter, $\hat{N} \sim \mathcal{O}(10^2)$ while $N \sim \mathcal{O}(10^4)$.

Finally, substituting Equation 3.2 into Equation 3.1, one gets the desired minimal form for the amplitude

$$M = \sum_{m=1}^{\hat{N}} \hat{c}_m \hat{I}_m \,, \text{ with } \quad \hat{c}_m = \sum_{i=1}^{N} c_{i,m} f_i \,. \tag{3.3}$$

The evaluation of the bare amplitude $M$ consists of two steps: firstly, solving the IBP equations in order to derive the required set of coefficients $c_{i,m}$ appearing in Equations 3.2 and 3.3, and secondly evaluating the master integrals $\hat{I}_m$.

The subject of this chapter is the calculation of the coefficients $c_{i,m}$. We note that they are process-independent in the sense that they are the same for every massless two-loop five-point amplitude. Their universality is one of the advantages of the IBP method. All process-specific information is encoded into the coefficients $f_i$, which are comparatively easy to compute.

The master integrals (or basis vectors) $\hat{I}_m$ are also process-independent. We note that in a vector space, the choice of a basis is not unique; moreover, it can happen that two or more master integrals are linearly related to each other via discrete symmetries, such as a permutation of the internal momenta. In the context of the IBP approach, however, such masters have to be treated as independent basis vectors. In this chapter we will not be concerned with the evaluation of the masters themselves, since that is a separate problem, albeit a related one [100–107]. All planar master integrals relevant for the present work were already known in analytic form [108].

The solving of the IBP identities in the past 20 years or so has been based on the Laporta algorithm [109], which will be described in Section 3.2. Many computer implementations of this algorithm exist [110–116]. Although the results are analytic in the kinematic variables, they remain numeric in the powers of the propagators and cannot, therefore, be expected to solve problems of arbitrary complexity. Experience suggests that the evaluation of the massless two-loop five-point QCD amplitudes is at the boundary of what is possible with the existing implementations of the Laporta approach.

Many novel ideas for the solving of the IBP equations have been proposed in the recent past [35, 117–122]. These new ideas and methods have made possible the evaluation of specific/planar all-gluon five-point amplitudes [123–129] as well as some non-planar ones [130]. Ideas towards solving the IBP identities in abstract form have also been put forward [131].

In this chapter we explore a different strategy for solving the IBP identities. We implement this strategy in an efficient, custom-written `C++` program and use it to solve the IBP identities needed to compute the complete set of planar two-loop five-point amplitudes in massless QCD (with quarks and/or gluons) in analytic form. We also present new non-trivial non-planar results. Based on our findings, we expect that our framework could be used to compute the remaining non-planar IBPs in analytic form in a time frame of a few months on a cluster of a few hundred CPUs.

The rest of this chapter is organised in the following way. Section 3.2 explains the origin of the IBP equations and discusses the standard (Laporta) algorithm used to solve them. Section 3.3 presents our strategy for solving IBP systems. Section 3.4 lays out the specifics of the 2-loop 5-point massless integrals to which we will be applying the IBP-solving methods developed in this chapter. Section 3.5 describes the `C++` program written by the author of this thesis for solving IBP systems. Section 3.6 presents the results obtained. Concluding remarks for this chapter are given in Section 3.7.

## 3.2 Background: IBPs and the Laporta algorithm

In this section, we will explain how the IBP equations are derived, after which we will briefly discuss the standard method for solving these equations.

Let us start by introducing some notation. The index $i$ labelling the integral $I_i$ (in Equations 3.1 and 3.2) is a composite index. It is natural to express it through the powers of the propagators appearing in the corresponding integral. For example, for a generic $L$-loop integral we have

$$I_i \equiv I(n_1, \ldots, n_P) = \int d^d k_1 \ldots d^d k_L \frac{1}{\Pi_1^{n_1} \ldots \Pi_P^{n_P}} \,. \tag{3.4}$$

The functions $\Pi_j$ are called *propagators* and they are bilinear functions of the loop momenta and external momenta. The explicit form of the propagators appearing in 2-loop 5-point massless integrals will be given in Section 3.4. We note that the indices $n_j$ are not restricted to taking positive values: they are also allowed to be negative, causing the corresponding bilinears $\Pi_j$ to appear in the numerator of the integrand.[1] When working with loop integrals where the numerator is a function of the loop momenta (e.g. $k_1^2(k_2 \cdot p)$, where $p$ is an external momentum), it is often helpful to express these numerators as a linear combination of the bilinears $\{\Pi_j\}$. Sometimes it is necessary to enlarge the set $\{\Pi_j\}$ by introducing a few fictitious propagators in order to obtain a full basis of bilinears through which any numerator can then by expressed. These extra propagators only appear in the numerators of the integrals, and never in the denominators. In the case of the $B$ and $C$ integral families discussed below in Section 3.6, we introduce 3 extra propagators in each family to supplement the 8 propagators originally present.

IBP identities arise from the observation that given a set of propagators $\{\Pi_j\}$ and a set of indices $\{n_j\}$, we can write down the the following identity [96, 97]:

$$0 = \int d^dk_1 \ldots d^dk_L \, \frac{\partial}{\partial k_l^\mu} \left( \frac{v^\mu}{\Pi_1^{n_1} \ldots \Pi_P^{n_P}} \right) , \tag{3.5}$$

where $l \in \{1, \ldots, L\}$ and $v^\mu$ is any linear combination of internal and external momenta. This identity follows from the fundamental theorem of calculus together with the fact that the surface term (which would normally appear on the left-hand side of the identity) can always be neglected when working in dimensional regularisation.[2] The vanishing of the surface term is evident if $\sum_j n_j$ is a sufficiently large positive number, and can in fact be shown to hold for all $n_j \in \mathbb{C}$ by treating the loop integrals as functions of complex $n_j$ and analytically continuing in $n_j$. Alternatively, Equation 3.5 can be seen as a consequence of the translational invariance of dimensionally-regularised integrals:

$$\int d^dk_l \, f(k_l^\mu) = \int d^dk_l \, f(k_l^\mu + q^\mu) , \tag{3.6}$$

where $q^\mu$ can be an arbitrary constant momentum vector. Equation 3.5 contains the derivative of a product, which can be expanded by repeatedly applying the standard identity $\frac{\partial(fg)}{\partial x} = \frac{\partial f}{\partial x} g + f \frac{\partial g}{\partial x}$. The right-hand side of Equation 3.5 thus becomes a linear combination of integrals, whose coefficients are polynomial functions of external kinematic invariants[3] $\{x_1, \ldots, x_K\}$ and the space-time dimension[4]. Thus, given any set

---

[1] Thus, negative indices correspond to numerator powers, whilst positive indices correspond to denominator powers. Although at first sight somewhat confusing, this is the standard convention in this field of research.

[2] Many of the manipulations performed here would be incorrect if performed on unregularised integrals but are valid in dimensional regularisation where all divergent expressions are well-defined.

[3] Any *internal* momenta should, of course, remain inside the integral and can be re-expressed as a linear combination of inverse propagators, as described earlier.

[4] Note that in dimensional regularisation, $\frac{\partial k^\mu}{\partial k^\mu} = \delta^\mu_\mu = d$.

of indices $\{n_j\}$ (this set is known as a *seed*), any choice of $v^\mu$, and any choice of $l$, we can write down a linear combination of integrals (with coefficients in $\mathbb{Q}[x_1, \ldots, x_K, d]$) that sums to zero. This is known as an IBP equation.

It is helpful to notice that differentiating an integrand (and then, if necessary, re-expressing any resulting numerator as a linear combination of inverse propagators) can never introduce new propagators into the denominator (although it may remove, or change the power of, an existing propagator). This observation suggests a natural ordering criterion for the integrals: those with fewer propagators in the denominator are to be considered as being simpler than those with more propagators. In assessing the complexity of an integral, one can furthermore check whether any denominators appear with $n_j > 1$ (we label such denominators as having $n_j - 1$ dots), and also compute the total numerator power. The number of propagators, number of dots, and total numerator power thus provide three criteria for assessing the complexity of an integral: in all three cases, a smaller number indicates a *simpler* integral. A partial order can be defined on the integrals by considering these three criteria in the order given (i.e. the number of propagators takes precedence over the number of dots, which in turn takes precedence over the total numerator power).

For a given family of integrals (defined by the set $\{\Pi_j\}$), an infinite number of IBP equations can be written down by combining different choices of $v^\mu$, $l$, and seed. The Laporta algorithm [109], which we will now describe, allows one to solve the IBP equations for any specific, finite set of desired integrals. The first step in the algorithm is to enumerate a set of seeds covering the desired integrals. Typically, this is done by examining these integrals and identifying largest number of dots ($d_{max}$) and the largest total numerator power ($n_{max}$). One then generates all possible seeds with up to $d_{max}$ dots and total numerator powers of up to $n_{max}$. These seeds are then used together with Equation 3.5 to obtain a large system of IBP equations which are linear in the integrals (although polynomial in the kinematic invariants and in $d$). Next, one performs Gaussian elimination on this linear system. Gaussian elimination requires one to select elements about which to *pivot*. The Laporta algorithm opts at each stage to pivot about the most complicated integral (as defined by the above partial ordering) present in an equation.[5] This choice ensures that more "complicated" integrals in an equation are always eliminated in favour of "simpler" ones. After the Gaussian

---

[5] Since we have a partial order rather than a total order, it is possible for two integrals to be found to have the same complexity. We break ties, arbitrarily but deterministically, by directly treating each of the indices $\{n_j\}$ as an additional measure of the complexity of an integral. (We consider these indices one at a time, from left to right, until the tie is broken.)

elimination has been performed, all the integrals of interest should[6] be found to have been expressed in terms of a small basis of so-called *master* integrals, which are "simple" according to the partial ordering criteria described earlier.

The Laporta algorithm thus achieves the goal, stated in Equation 3.2, of writing each of the desired integrals as a linear combination of master integrals, with coefficients that are rational functions of the external kinematic invariants and the space-time dimension. In the notation of Equation 3.4, we can now re-write Equation 3.2 in the form

$$I(n_1,\ldots,n_P) = \sum_{m=1}^{\hat{N}} c_m(n_1,\ldots,n_P)\hat{I}_m\,. \tag{3.7}$$

We note in passing that just like the index $i$, the index $m$ is also a composite one and in what follows we will sometimes use its explicit form.

It is important to emphasise that in practice, IBP systems are easy to generate but very difficult to solve. The difficulty arises because although the IBP equations are linear in the integrals, the coefficients of these integrals are polynomials in several variables. Thus, whereas Gaussian elimination is ordinarily applied to matrices containing real or complex numbers, in the case of IBP systems we apply Gaussian elimination to a "matrix" whose entries are multi-variate rational functions. During the process of solving the IBPs, these rational functions rapidly grow in size and complexity. This complexity is the core reason for the computational bottleneck that IBP systems produce in the solving of multi-loop amplitudes.

Finally, let us mention that one would ideally let the indices $n_j$ remain symbolic, rather than enumerating an exhaustive list of seeds and solving a specific set of integrals. Solving the IBP equations in this form would give so called *closed-form* solutions, which are explicit functions not only of $d$ and the kinematic invariants, but also of the indices $\{n_j\}$. Currently, such closed-form solutions are only known for the very simplest families of integrals, and are far beyond reach for problems with many legs, loops, and scales.

## 3.3 Strategy for solving the IBP identities

We will now present our strategy for solving IBP identities. Our starting point is the assumption that the IBP system has a solution, i.e. *every* loop integral $I_i$ can be

---

[6] If too restrictive a range of seeds is used (e.g. 0 dots and 1 numerator power), the system might not get fully solved. If this occurs, one typically has to repeat the calculation with a larger range of seeds.

expressed through a set of basis master integrals as in Equation 3.2 and that such a basis set of masters is known. The existence and construction of a finite basis of master integrals is a well-studied problem [132–134]. It was proven in Ref. [135] that the number of master integrals is always finite. There are several ways to construct such a basis. For example, one could solve the IBP system over a restricted set of integrals and/or use numerical values for the kinematic invariants. In any case, finding a basis is not a bottleneck and we consider this step to be trivial. This is certainly true for the two-loop five-point massless amplitudes considered here, where we have easily identified the sets of masters for all topologies.

To solve the IBP identities means that for all required integrals $\{I_i\}$, one must derive all the coefficients $c_{i,m}$ appearing in Equation 3.2 and Equation 3.7. In existing approaches for solving IBP identities, all of the coefficients $c_{i,m}$ are calculated simultaneously. In this work we pursue a different strategy[7] for their solving: we choose a particular master, $\tilde{m}$, and calculate the coefficients $\mathcal{C}_{\tilde{m}} \equiv \{c_{i,\tilde{m}}\}$ of that master *independently* of the coefficients of all other masters. We then repeat for each of the other masters, in turn. Put differently, we split the problem of solving the system of IBP equations into $\hat{N}$ independent problems, one for each of the $\hat{N}$ masters.

This strategy is implemented by constructing the usual set of IBP identities and then *taking coefficients with respect to the basis of master integrals.* For example, in order to derive $\{c_{i,1}\}$ – the coefficients of master $\hat{I}_1$ – we would in principle impose $c_{i,m} = 0$ for all $m \neq 1$ and for all $i$. In practice, we simply remove from the IBP equations all master integrals except $\hat{I}_1$. We then solve the resulting IBP equations, obtaining a solution that is of the form

$$I(n_1, \ldots, n_P) = c_1(n_1, \ldots, n_P)\hat{I}_1\,, \tag{3.8}$$

i.e. we obtain the coefficients $c_1(n_1, \ldots, n_P)$, which are the $\hat{I}_1$-components of the full solution. Repeating the same approach but imposing $c_{i,m} = 0\ \forall m \neq 2$, one derives the coefficients $c_2(n_1, \ldots, n_P)$ and so on. To obtain the complete solution of the IBP system, one simply needs to put together all $\hat{N}$ independently-derived sets of coefficients.

It is easy to see why this strategy leads to the correct solution of the IBP equations. Its correctness follows from the fact that in any vector space, a vector can be written uniquely as a vector superposition of a basis. This means each integral $I_i$ has an

---

[7] Although the strategy presented here was new at the time this work was carried out, we note that the program Kira implemented a similar feature around the same time, offering the option of computing the coefficients of a subset of masters.

expansion in the set of masters $\hat{I}_m$, i.e. at each step the IBP equations could in principle be rewritten as a homogeneous linear combination of all master integrals. Since the IBP equations are themselves linear and homogeneous in terms of the integrals $I_i$, one can see that the IBP equations never mix the coefficients of different master integrals. In essence, our strategy exploits the fact that each of the sets $\mathcal{C}_m$ can be computed in isolation from the others.

The IBP solving strategy described here is independent of the approach used for solving the system of IBP equations. In practice, we will use the standard Laporta algorithm but one does not have to. In fact, we arrived at this idea while trying to find a way for solving the IBP system in closed form. We hope to return to this in a future work.

We have checked the correctness of our strategy in a number of non-trivial examples, such as the complete two-loop four-point amplitude (cross-checked with the program Reduze [112, 113]) and a number of two-loop five-point planar and non-planar cases as explained in detail in Section 3.6.

At this point it will be beneficial to contrast our strategy to the usual way of solving IBP identities and to discuss the origin of increased efficiency. To this end we need to introduce the notion of a *sector* which is well-known in the IBP literature.

A sector is effectively a sub-topology indexed by 0s and 1s and defined by the position of a subset of propagators. (The terms *propagator* and *topology* will be defined in Section 3.4.) For example, $[1, 1, 1, 0, \ldots, 0]$ represents a sector. In the notation of Equation 3.4 this sector contains all integrals $I(n_1, \ldots, n_P)$ for which $n_{1,2,3} > 0$ while $n_{4,\ldots,P} \leq 0$. The number of different propagators that define a sector is called its *weight*. For example, the sector $[1, 1, 1, 0, \ldots, 0]$ is of weight 3. A sector is called a zero-sector if all integrals that belong to it vanish. For the massless two-loop five-point amplitudes, all sectors with weight $< 3$ are zero-sectors and some sectors with weight $\geq 3$ are also zero-sectors.

Our strategy can lead to a more efficient solving of the IBP system for several reasons. Firstly, when working with the coefficients of a single master, many sectors become zero-sectors and thus do not need to be computed.

Secondly, restricting ourselves to the coefficients an individual master at the outset of the calculation simplifies the intermediate steps. The reason is that, taking the example of the Laporta algorithm, the IBP equations that will be solved first are generated from seeds that are in some sense close to the master integrals.[8] In this way the information about the absence of other masters is incorporated into the resulting IBP equations early on in the solving process. In large systems with many masters, our strategy could lead to a significant reduction in the size of the intermediate expressions. This, in turn, would reduce the computer memory requirement that is the limiting factor in solving large problems.

Thirdly, by solving with one master in a given computing job, one can parallelise the problem by running several independent jobs (corresponding to different masters) at the same time. The amount of parallelisation achieved depends primarily on the number of master integrals in the integral family (or families) being solved. However, one should keep in mind that, as we explain in Section 3.6, the running times for different masters can be vastly different. In Section 3.5.3, an additional parallelisation method will be described, which allows the IBPs associated with a single master to be solved using large numbers of CPUs in parallel.

## 3.4 Application to 2-loop 5-point integrals

Although our IBP-solving strategy and `C++` program are generic, we have applied them in particular to the case of 2-loop 5-point massless integrals that contribute to NNLO QCD corrections for several 2 → 3 processes. In this section, we will classify and briefly describe these integrals.

We will use here the example of the squared two-loop amplitude $M = \langle A^{(2)} | A^{(0)} \rangle$ for the process $q\bar{q} \to q'\bar{q}'g$, which is one of the processes contributing to 3-jet production.[9] From the viewpoint of the IBPs it is representative of the other two-loop five-point massless amplitudes.

The integrals appearing in $M$ belong to several *families.* We label the non-planar family $B$ and the planar family $C$. Each family is defined by a set of 11 propagators, shown in Table 3.1.

The momenta $p_1$ and $p_2$ are incoming while $p_3$, $p_4$ and $p_5$ are outgoing. They satisfy $p_1 + p_2 = p_3 + p_4 + p_5$. We define Lorentz-invariant kinematic variables $s_{ij} = (p_i + p_j)^2$. For any integral in a family, the numerator and denominator will each be a product of

---

[8] Assuming that, as is usually the case, the masters are chosen with the help of the same ordering criterion that is used to generate the seeds for solving the IBP equations.

[9] To identify the integrals appearing in this amplitude, we used the program Reduze [112, 113] for the generation of the Feynman diagrams, for their squaring and for the summation over colour and spin traces. Some of those calculations were sped up with the help of the program Form [136].

**Table 3.1** Propagators $\{\Pi_j\}$ defining the integral families $B$ and $C$

| $j$ | $\Pi_j^{(B)}$ | $\Pi_j^{(C)}$ |
|---|---|---|
| 1 | $k_1^2$ | $k_1^2$ |
| 2 | $k_2^2$ | $k_2^2$ |
| 3 | $(k_1+p_1)^2$ | $(k_1+p_1+p_2)^2$ |
| 4 | $(k_1+p_1+p_2)^2$ | $(k_1-k_2)^2$ |
| 5 | $(k_2-p_3)^2$ | $(k_2+p_1)^2$ |
| 6 | $(k_2-k_1-p_3)^2$ | $(k_2+p_1+p_2)^2$ |
| 7 | $(k_2-k_1-p_1-p_2+p_4)^2$ | $(k_2-p_3)^2$ |
| 8 | $(k_2+p_4)^2$ | $(k_1+p_1+p_2-p_3)^2$ |
| 9 | $(k_2+p_1+p_2)^2$ | $(k_1+p_1+p_2-p_3-p_4)^2$ |
| 10 | $(k_2+p_1)^2$ | $(k_2-p_3-p_4)^2$ |
| 11 | $(k_1+p_3)^2$ | $(k_1+p_1)^2$ |

the propagators shown. In each family, the 11 propagators form a complete basis of bilinears that can be used to construct any other expression appearing in the numerator of an integral. For example, $k_2 \cdot p_3 = \frac{1}{2}k_2^2 - \frac{1}{2}(k_2-p_3)^2$ since all particles are massless here.

An integral family can be sub-divided into *topologies*. The denominator of any integral will contain a subset of the propagators defining the integral family. A *topology*[10] groups together integrals whose denominators share propagators. As was mentioned in Section 3.2, the identity (Equation 3.5) that generates the IBP equations cannot generate new denominator propagators that are not already present in the denominators of the seeds. One therefore generates a separate system of IBP equations for each topology. We label the non-planar topologies as $B_n$ and the planar topologies as $C_n$, where the subscript $n$ is a positive integer to distinguish the topologies within in a family, while $B$ and $C$ refer to the families defined above.

There are two non-planar topologies ($B_1$ and $B_2$) and two planar topologies ($C_1$ and $C_2$), each having 8 propagators. The topologies are defined by their highest-weight sectors:

$$\begin{aligned} B_1 &= B\,[1,1,1,1,1,1,1,1,0,0,0] \\ B_2 &= B\,[1,1,1,1,0,1,1,1,0,0,1] \\ C_1 &= C\,[1,1,1,1,1,1,0,1,1,0,0] \\ C_2 &= C\,[1,1,1,1,1,0,0,1,1,0,1] \end{aligned} \tag{3.9}$$

---

[10] The term *topology* here is unrelated to the branch of mathematics sharing the same name.

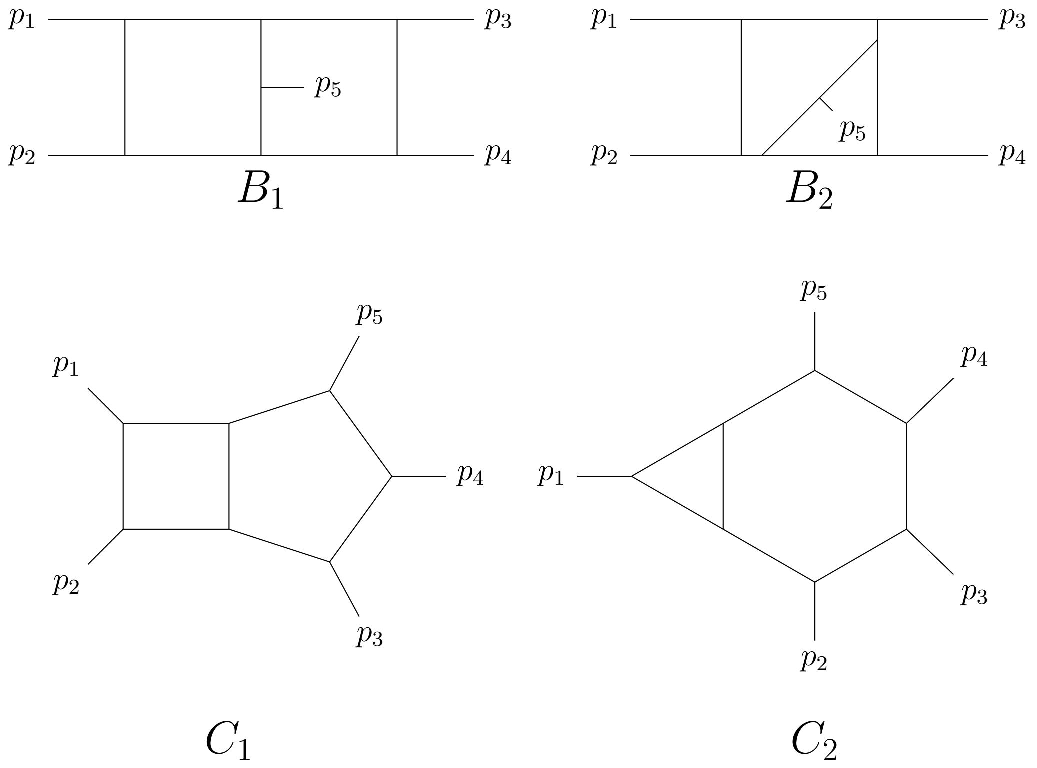


**Fig. 3.1** The integral topologies $B_1$, $B_2$, $C_1$, and $C_2$.

A diagrammatic representation is given in Fig. 3.1. All master integrals needed in the computation of the two planar $C$ topologies have been computed in analytic form [108] within the approach of Ref. [137]. There also exists a numerical library [33] for the evaluation of the $C_1$ master integrals (albeit using a different basis of masters). Work towards the remaining non-planar ones is ongoing [138, 139].

For each of the topologies in Equation 3.9, we have identified the master integrals. We find 113 masters in $B_1$, 75 in $B_2$, 62 in $C_1$, and 28 in $C_2$. Their explicit definitions, in the notation of Table 3.1 and Equation 3.9, can be found in an electronic file attached to the paper [2]. Two of the masters in the $C_1$ topology are equal, being related to one another by the discrete permutation $k_1 \leftrightarrow k_2$, but for the purposes of the IBP reduction using our strategy they must be treated as being distinct.

In order to compute the process $q\bar{q} \to q'\bar{q}'g$, one must solve the $B_1$, $B_2$, $C_1$, and $C_2$ topologies up to the values of $n_{max}$ and $d_{max}$ shown in the Table 3.2. The same range of integrals is required for the calculation of NNLO QCD corrections to 3-photon production, which we present in Chapter 4. We believe this range is in fact sufficient to allow the computation of any 2-loop 5-point massless QCD amplitude.

**Table 3.2** Ranges of integrals required for computing the squared 2-loop unpolarised amplitude $q\bar{q} \to q'\bar{q}'g$

| Topology | $n_{max}$ | $d_{max}$ |
|---|---|---|
| $B_1$ | 4 | 1 |
| $B_2$ | 5 | 1 |
| $C_1$ | 5 | 1 |
| $C_2$ | 4 | 1 |

Prior to our work, solutions to the IBP equations were not available for the required range of integrals shown in Table 3.2. A central result in this thesis is the analytical solution to the IBPs for all $C_n$ integrals within these required ranges. This allows 2-loop 5-point massless QCD amplitudes to be computed in the leading-colour (or large $N_c$) approximation, where non-planar integrals (i.e. those in the $B_n$ topologies) do not contribute. The NNLO QCD corrections to 3-photon production at the LHC, which we present in Chapter 4, are a direct example of the application of these results. Of the $C_n$ topologies, the most difficult to compute is $C_1$, and in the next section we will be using the $C_1$ topology as a benchmark to measure the capability of the computational strategies described there. In addition, we have obtained partial results for the non-planar ($B_n$) topologies, whose difficulty surpasses even $C_1$, and will report on this work as part of Section 3.6.

## 3.5 Computational implementation

To solve the 2-loop 5-point massless IBPs that are the focus of this chapter required going beyond the capabilities of publicly-available IBP programs [110–116]. To this end, the author of this thesis designed and implemented a new IBP-solving program incorporating a number of advanced features to maximise speed and minimise the use of computational resources. In this section, these design features will be described in detail. We note that although our program includes an implementation of the strategy described in Section 3.3, the features that we present in this section are independent of the strategy and could, if desired, be used independently of it.

As was mentioned above, throughout this section we will be using the $C_1$ topology of integrals, which contribute to the 2-loop amplitude for the process $q\bar{q} \to q'\bar{q}'g$, as a benchmark problem in order to illustrate the performance impact of the computational features that we describe.

### 3.5.1 Computer algebra

Solving the IBP equations requires performing billions of arithmetic operations (additions, subtractions, multiplications, and divisions) on multi-variate polynomials and rational functions. Such operations, which are performed symbolically rather than numerically, are highly non-trivial to implement computationally and they constitute a research area in their own right.

Initially we carried out some experimentation with writing algorithms to perform these arithmetical operations. Additions on polynomials are simple to perform: two polynomials, each of length $N$, can trivially be added in $\mathcal{O}(N)$ operations. Polynomial multiplication was also easy to implement and can be carried out on two polynomials, each of length $N$, in $\mathcal{O}(N^2)$ operations by using the standard long-multiplication method learned by schoolchildren.[11] However, multi-variate polynomial division is highly non-trivial to implement, since it requires a robust method for cancelling common factors between the dividend and the divisor. This cancellation of common factors is particularly important when working with rational functions rather than polynomials, as we discovered with the first prototype of our IBP program: the prototype did not implement the cancellation of common factors and when we tested it by solving a simple one-loop example, we found the solved IBPs to contain kinematic variables with exponents as high as 260. After cancelling common factors using Mathematica, the corresponding exponents were found to be no larger than 3. This example demonstrates that a powerful computer algebra system is indispensable when solving large systems of IBPs symbolically.

In view of the the difficulties associated with implementing arithmetical operations on multivariate polynomials and rational functions, we decided to use an external computer algebra package, Fermat [140], which is a leader in this particular area and is used by some of the public IBP programs [113, 115, 116]. Although Fermat is powerful, its primary interface is designed for a human using it interactively via a terminal. The existing IBP programs communicate with Fermat by simulating a human: each polynomial operation is sent as a string, e.g. `"(x^2 - y^2) * (x^2 + y^2)"`, and all outputs are likewise received as strings, e.g. `"x^4 - y^4"`. An alternative `C`-based interface to Fermat exists that allows the direct manipulation of objects (polynomials and rational functions) inside Fermat's internal memory via pointers. We wrote a custom set of `C++` classes to interact with this alternative interface, in order to

---

[11] In fact, polynomial multiplication can elegantly be performed in $\mathcal{O}(N\log(N))$ operations by using finite fields and Fast Fourier Transforms. We did not, however, implement this here.

maximise speed while automating memory management (in order to avoid memory leaks). These `C++` classes represent multi-variable polynomials and rational functions which can be added, subtracted, multiplied, and divided just like native `C++` types such as `int` and `float`, with the underlying operations being performed by FERMAT. As a result of careful optimisations, over 99% of the program's running time can be spent on computationally-expensive symbolic arithmetic operations (i.e. within FERMAT, which runs as a separate process), with our own code accounting for less than 1% of the running time.[12]

The features described so far, combined with the strategy described in Section 3.3, allowed us to solve all $C_1$ integrals with up to 4 numerator powers (i.e. $n_{max} = 4$) and 1 squared denominator (i.e. $d_{max} = 1$), with the slowest job (associated with $C_1$ master 3) requiring 35 days on a single CPU and 113 GB of RAM.

### 3.5.2 Lazy evaluation

Although the features described in Section 3.5.1 enabled us to solve all $C_1$ integrals with up to 4 numerator powers and up to 1 squared denominator, this would not be sufficient to solve our amplitudes of interest, where the most complicated integrals have 5 numerator powers and 1 squared denominator. In this section we will describe a *lazy evaluation* strategy that improved the running speed by an order of magnitude and thus allowed us to solve the additional required integrals. The lazy evaluation setup also laid the foundations for the parallelisation strategy which will be described in Section 3.5.3.

As explained earlier, almost all of the running time of our IBP program is spent on performing symbolic arithmetic operations on multi-variable rational functions, using the FERMAT library. Therefore, in order to achieve a further improvement in speed without modifying the FERMAT library itself, we sought to reduce the number of arithmetic operations performed. Two observations provide clues as to how this could be achieved:

1. Firstly, IBP systems typically contain many redundant equations. When performing Gaussian elimination, these equations acquire the form $0 = 0$, but only after

---

[12] In order to solve the most difficult non-planar IBP systems, we run the parallelised version of our program (described in Section 3.5.3) on High-Performance Computing facilities, where jobs are subject to a time limit of 36 hours. Our jobs can require weeks or months of running and must therefore be paused and resumed every 36 hours. On these machines, $\mathcal{O}\left(10\%\right)$ of a job's running time is spent on this pausing/resuming procedure, but we accept this in view of the $\sim$ 100-fold speed increase offered by the parallelisation.

many substitutions that eventually lead to the cancellation of all terms in the equation. Although these redundant equations can provide a useful verification of the consistency of the IBP system (since an inconsistency will typically lead to a contradictory result like $1 = 0$), they represent unnecessary operations that could in principle be eliminated from the outset, if one knows which equations are redundant.[13]

2. Secondly, it is well-known that in a generic IBP system, many of the equations will contain integrals that do not themselves appear in any amplitude. Although these integrals are of no direct interest, the equations in which they appear are often necessary in order to ensure that the IBP system has minimal rank, i.e. to ensure that all of the integrals which *are* of interest get fully solved. Ideally, one would like to avoid unnecessary auxiliary computations, while still performing all of the computations that are needed to solve the integrals of interest.

In light of this, we adopted a so-called *lazy evaluation* approach, which is well known in the field of computer science. In this approach, no computationally-expensive symbolic arithmetic operations are performed until (and unless) they are required for producing the final output of the program. Until then, the program maintains a network of placeholder objects representing the unknown expressions and their interdependencies. An example of such a network is shown in Fig. 3.2 and will be described shortly.

The IBP calculation has two stages, which we will refer to as the *forwards stage* and the *backwards stage*. In the forwards stage, Gaussian elimination is used to solve the IBP system, but all algebraic operations on rational functions are replaced by computationally-cheap operations that merely manipulate the network of placeholder objects. The IBP solutions produced during this stage are implicit (encoded in the structure of the network), rather than explicit rational functions. Once the forwards stage is complete, we identify the specific placeholder objects that are of interest: those representing the solutions to the integrals we seek to solve. We then proceed to the backwards stage, where we traverse the network and perform the minimal set of algebraic operations needed to obtain explicit rational functions for expressions that have been identified as being of interest. The majority of the running time is occupied

---

[13] We mention the presence of redundant equations because they were one of our motivations for considering lazy evaluation. However, before we implemented lazy evaluation, we tried to use a numerical run (solving the $C_1$ IBPs in the range $(n_{max}, d_{max}) = (4, 1)$, with the variables $s_{ij}$ and $d$ replaced by numbers) in order to identify and remove all redundant equations. Surprisingly, although 40–50% of the IBP equations were found to be redundant, we obtained almost no performance gain by removing them: the speed of solving the IBP equations was unchanged to within 1%.

by the backwards stage, which can take many weeks, whereas the forwards stage takes just a few hours.

An example of a network of lazily-evaluated expressions is shown in Fig. 3.2. The network contains three types of expressions:

1. Symbolic variables, depicted in the figure as squares. There are 5 symbolic variables: 4 kinematic variables[14] $(s_{23}, s_{34}, s_{45}, s_{51})$ and the space-time dimension $d$. All other expressions in the network are ultimately expressible in terms of these 5 variables.

2. Immutable, unevaluated expressions, depicted in the figure as circles. These constitute the bulk of the network. In the case of non-planar 2-loop 5-point IBPs, the graph can contain up to $\mathcal{O}\left(10^9\right)$ such expressions. Each unevaluated expression points to two other expressions, and also has a label to denote whether it is the sum, difference, product, or quotient of the two expressions to which it points.

3. Mutable expressions, depicted in the figure as stars. These are the expressions that appear directly in the system of IBP equations.

The distinction between mutable and immutable expressions warrants further explanation. During the forwards stage, where we perform Gaussian elimination on the IBP system, the equations in the system will repeatedly be added on to each other (in various linear combinations) until the system has been solved. The expressions in the equations will, of course, therefore undergo many changes. If the expressions are lazily evaluated, we must not only keep track of the current value of each expression but also of all previous values of each expression. The following example operations illustrate why this is the case.

1. $q_1 := s_{23} + s_{34}$

2. $q_2 := s_{34} - s_{45}$

3. $q_3 := q_1 + q_2$

4. $q_1 := q_2 + q_3$

---

[14] We have set $s_{12} = 1$ in order to improve the speed and performance of the calculation. The missing factors of $s_{12}$ in the final solutions can easily be restored using dimensional considerations.

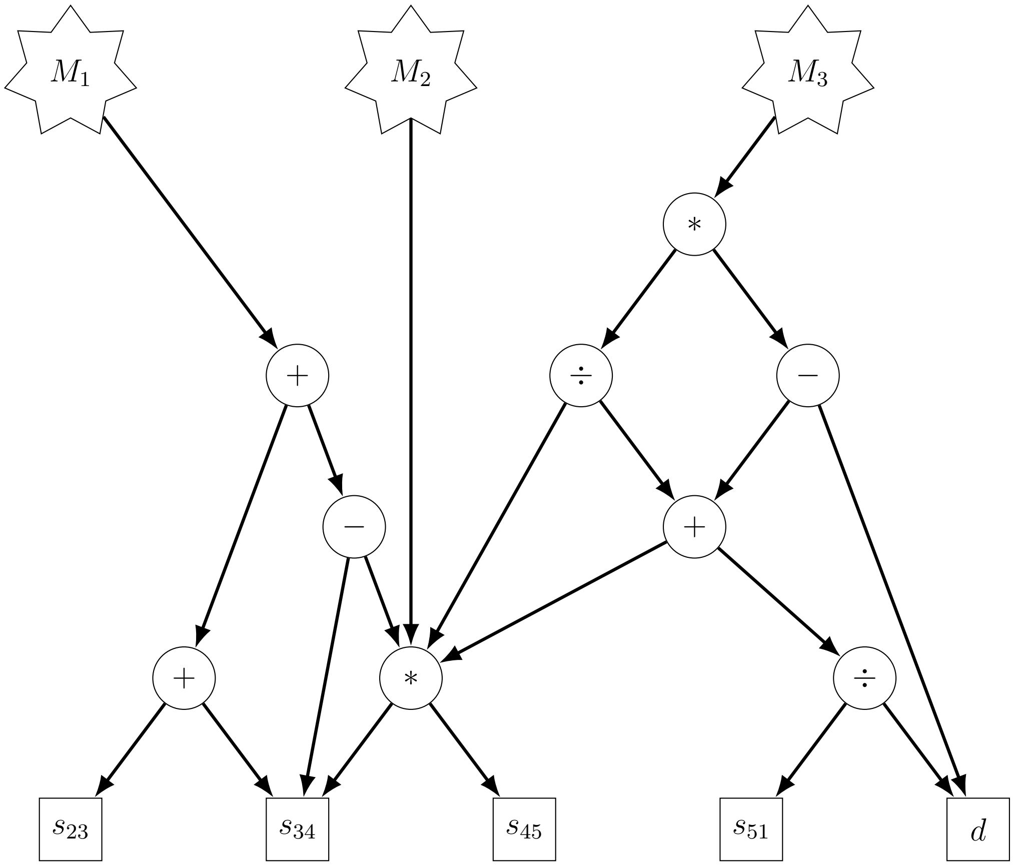
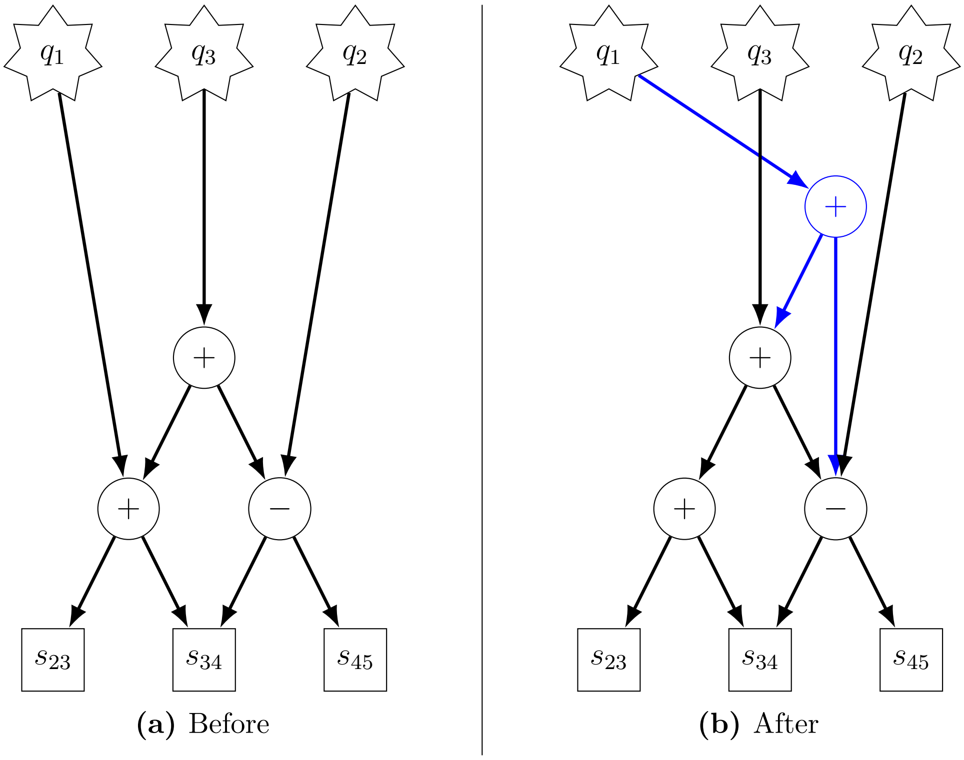


**Fig. 3.2** Network of lazily-evaluated expressions. The stars in the top row represent coefficients of the integrals in the IBP equations. The circles represent immutable, unevaluated expressions. The squares on the bottom row represent the kinematic variables $s_{ij}$ and the space-time dimension $d$. The arrows emanating from an expression identify the other expressions on which it directly depends.



**(a)** Before **(b)** After

**Fig. 3.3** Modifying an expression (here, $q_1$) in a lazy-evaluation network.

Here, $q_n$ are polynomials (in $s_{ij}$) which would constitute the coefficients of the integrals in some system of IBP equations. In this example, $q_1$ initially takes the value $s_{23} + s_{34}$ and this is used to define $q_3$ in step 3. In step 4, the value of $q_1$ changes. If we were explicitly performing the arithmetic operations during the course of the job, we would only need to keep track of the *current* values of $q_1$, $q_2$, and $q_3$. However, when using lazy evaluation, our network of expressions must contain the current value of each expression *as well as all of its historical values*, since one expression might depend on a historical value of another expression. We therefore build most of the lazy-evaluation network using *immutable* expressions, whose value is fixed upon creation (although not explicitly known). In addition, we have class of *mutable* expressions, which represent changeable expressions appearing in the IBP equations. Each mutable expression points at a single immutable expression containing its value. Whenever we wish to modify a mutable expression, we follow the procedure illustrated in Fig. 3.3: we create a new *immutable* expression to represent the modified value, and then we point the mutable expression towards the newly-created immutable expression. The old immutable expression remains unchanged, which ensures that all other expressions in the network remain unchanged.

At first sight, it may appear puzzling that we are able to perform the forwards stage (Gaussian elimination) without explicit knowledge of the expressions being manipulated. In particular, Gaussian elimination requires frequent checks that an expression is not identically zero, in order to avoid dividing by zero. To determine this, we equip each mutable expression with a numerical cache which stores the numerical value of that expression at a few pre-defined numerical points, shown in Table 3.3. The integer values in Points 1-6 are randomly and uniformly drawn from the range $[1, 10^6]$, while the numerators and denominators in Points 7-9 are integers randomly and uniformly drawn from the range $[1, 10^5]$. Although we have occasionally used all 9 points, we usually use only Points 1-6, which (as we will shortly explain), we find to be adequate for our purposes. Whenever we create or update a mutable expression (using the process depicted in Fig. 3.3), we also compute its numerical value at each of the numerical points. This allows us to check at any time whether a mutable expression is likely to be identically zero, simply by checking whether it is zero at all of the numerical points. An expression that is identically zero will, in this procedure, always be correctly identified as such. However, the procedure has a very small chance of producing a *fake zero*, if a non-zero expression happens to vanish at all of the pre-defined numerical points. We estimate the likelihood of a fake zero by keeping a log of *near misses*, where an expression is found to be zero at one or more (but not all) numerical points. In nearly all the jobs that we have run, no near misses occur. In a small number of jobs, 1 or 2 expressions (out of, typically, between $10^6$ and $10^9$ expressions) are near misses. All of the near misses that we have recorded occur because a non-zero expression vanishes at 1 (out of 6) numerical points; we have never seen a case where an expression vanishes at 2, 3, 4, or 5 points. Based on this, we conservatively assume that any non-zero expression will vanish at a single given numerical point with probability $10^{-6}$. We surmise that the probability for a non-zero expression to vanish at all 6 numerical points, and thus produce a fake zero, is extremely small $\left((10^{-6})^6 = 10^{-36}\right)$ and can safely be ignored.

So far, our discussion has focussed on the forwards stage, where we solved the IBP system numerically at a few numerical points (which were shown in Table 3.3), while building up a large network of placeholder objects representing the history all of the arithmetical steps taken to arrive at the numerical solutions. Let us now consider the backwards stage, where we use the network of placeholder objects as a guide to determine the operations to perform (now symbolically rather than numerically) in order to obtain the desired symbolic solutions to the IBP system.

**Table 3.3** Numerical points used in lazy evaluation network to check whether an expression is likely to be identically zero. We normally use Points 1-6, although we have experimented with also including Points 7-9.

| Point | $s_{23}$ | $s_{34}$ | $s_{45}$ | $s_{51}$ | $d$ |
|---|---|---|---|---|---|
| 1 | 400221 | 25015 | 978834 | 635833 | 968990 |
| 2 | 108036 | 941985 | 414720 | 249186 | 175239 |
| 3 | 337524 | 325850 | 577194 | 494707 | 957317 |
| 4 | 10775 | 306600 | 191884 | 374714 | 375070 |
| 5 | 300113 | 294501 | 45932 | 529533 | 752691 |
| 6 | 403090 | 788821 | 251635 | 929621 | 474782 |
| 7 | 65383/52453 | 94515/90421 | 50366/64456 | 53125/82490 | 11966/31470 |
| 8 | 65142/2845 | 99665/28690 | 1043/20268 | 64893/81290 | 8846/78172 |
| 9 | 64281/75827 | 27131/66428 | 31754/18921 | 3501/12273 | 80740/25083 |

As a first step in the backwards stage, we prune the network of placeholder objects: we identify objects representing solutions to the IBP equations and we keep these objects as well as all their dependencies, while discarding all other objects. An example of this is shown in Fig. 3.4. In practice, this pruning process is achieved by making each object in the network keep a count of the number of expressions pointing at it. (This is implemented by means of `C++` smart pointers.) If the count ever reaches zero, the object automatically immediately deletes itself (including all arrows emanating from the object), which may in turn cause other objects to also delete themselves. In the example shown in Fig. 3.4, the explicit deletion of $M_1$ and $M_2$ triggers the automatic deletion of all of the parts of the network shown in red. In general, smart pointers carry the risk of circular dependencies, where two (or more) objects point at each other and thus prevent each other's deletion. In our program, however, the use of immutable expressions (which can only point at other immutable expressions) ensures that circular dependencies cannot form: the expressions are added to the graph one at a time, and each expression can only point at earlier expressions and not at future ones.

Following the pruning of the network, we proceed to perform the symbolic arithmetical operations required to obtain explicit solutions to the IBP expressions of interest. This is simply carried out by identifying the expressions that are of direct interest, and attempting to evaluate each one in terms of its two direct dependencies. If the dependencies themselves have yet to be evaluated (which, initially, is most likely to be the case), then we first evaluate these dependencies (proceeding recursively, as necessary). All symbolic evaluations are carried out by Fermat. As soon as any expression becomes explicitly known, we delete its pointers to its dependencies. This

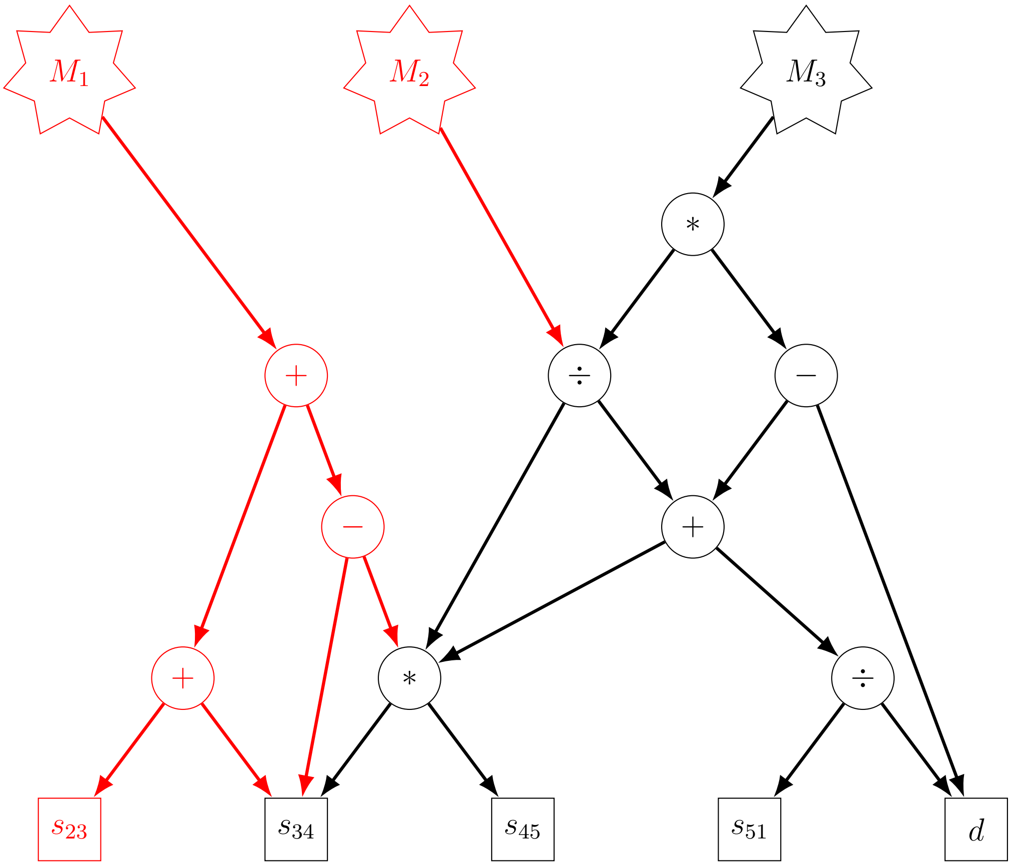


**Fig. 3.4** Pruning the network of lazily-evaluated expressions. In this example, $M_3$ is of interest, while $M_1$ and $M_2$ are not. All of the objects shown in red can therefore be deleted.

automatically causes the dependencies to be deleted (via the pruning mechanism described above) unless the dependencies are shared with other expressions. All of the expressions of interest are thus evaluated, while optimising the use of RAM by continuously pruning expressions that are no longer required.

The lazy-evaluation strategy worked surprisingly well: it increased the program's speed by an order of magnitude and thus allowed us to solve all required[15] $C_1$ integrals with up to 5 numerator powers and up to 1 squared denominator in only twice as much time as was previously spent reaching 4 numerator powers and up to 1 squared denominator. (Ordinarily, the running time increases by an order of magnitude when one extends an IBP system by 1 additional numerator power.) Lazy evaluation also laid the foundation for the parallelisation strategy described below in Section 3.5.3.

### 3.5.3 Parallelisation of individual jobs

In Section 3.3, we described how a system of IBPs can be replaced with $\hat{N}$ smaller systems of equations by taking coefficients with respect to the basis of master integrals. Although this allows for some amount of parallelisation (since the equations associated with different masters are solved independently), most of the computation time is spent on a small fraction of the jobs — those associated with the masters with the fewest propagators. In this section, we will describe a mechanism for further parallelisation, which enables the use of modern computing clusters with hundreds of CPUs running in parallel.

Although the lazy evaluation framework was originally conceived with the aim of minimising the number of computationally-expensive symbolic arithmetical operations, the framework also gives rise to the possibility of large-scale parallelisation. As we discussed in Section 3.5.2, calculations in this framework can be divided into a forwards stage, where the lazy evaluation graph is constructed, and a backwards stage, where the operations represented by the graph are performed symbolically. Since the forwards stage takes a few hours whereas the backwards stage takes several weeks or months, we would like to parallelise the backwards stage. The key observation is that many of

---

[15] The most difficult IBPs to compute are the projections of the top-sector integrals (those with the maximum number of propagators – here 8) onto the lowest-sector master integrals (those with the fewest propagators – here 3). When working with the lowest-sector masters, the lazy-evaluation setup allowed us to restrict ourselves to just solving these difficult top-sector integrals. This is sufficient to fully solve the QCD amplitude $q\bar{q} \to q'\bar{q}'g$. We removed this restriction after implementing the parallelisation procedure described in Section 3.5.3.

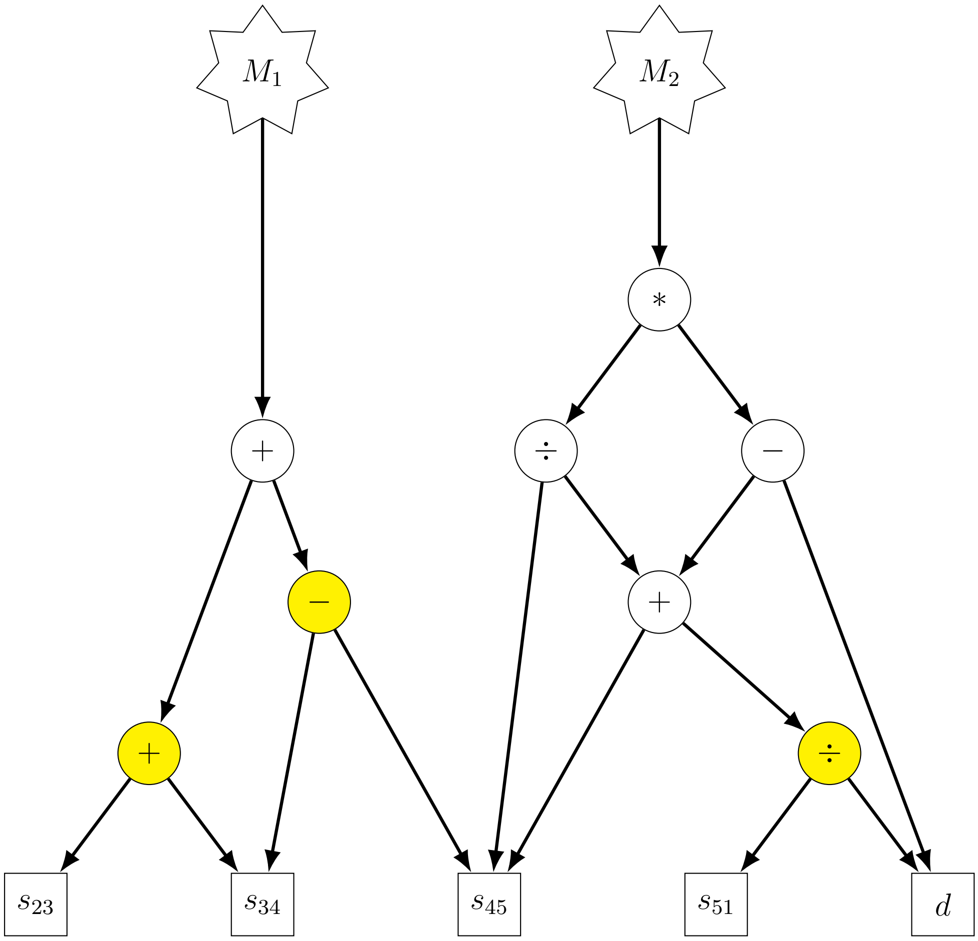


**Fig. 3.5** Parallel evaluation of expressions in a lazy evaluation graph. The three expressions highlighted in yellow are independent of one another. They can be therefore be evaluated in parallel.

the symbolic operations represented in a lazy evaluation network are independent of one another and can therefore be performed in parallel. This is illustrated in Fig. 3.5.

We implement the parallel evaluation of the lazy evaluation graph across many computing threads by designating one thread as the *master* thread and all the other threads as *worker* threads. Each worker thread runs a copy of FERMAT and uses this to evaluate questions (i.e. symbolic arithmetic operations) received from the master thread. All inter-process communication is carried out using the OPENMPI library [141, 142].

The master thread stores the lazy evaluation graph and manages the progression of the calculation. At the beginning of the backwards stage, after the pruning procedure discussed in Section 3.5.2, the master thread arranges all of the unevaluated expressions from the lazy evaluation graph into a one-dimensional list, taking care to avoid double-counting.[16] An *evaluation queue* is then constructed, containing all expressions whose dependencies are known in explicit form. (Initially, the only expressions known in explicit form are the 4 variables $s_{ij}$ and the dimension $d$.) The master thread distributes the expressions in the queue to the worker threads and waits for the results. As soon as a worker sends back an evaluated expression, the master thread saves the result, checks whether the result enables more questions to be added to the evaluation queue, and then finally dispatches more questions to the workers (as long as there are questions available in the queue and workers available to receive them). Throughout the job, the master thread operates the pruning mechanism described in Section 3.5.2 in order to remove any unnecessary expressions from the computer memory.

To maximise the speed, we wish at all times to maximise the number of workers employed in parallel. The length of the evaluation queue determines how much parallelisation is possible: as long as some expressions are still present in the evaluation queue, any additional workers that we add will bring a linear increase in speed. For the integral families that are the focus of this chapter, a typical lazy evaluation network can initially contain $\mathcal{O}\left(10^8\right)$ unknown expressions, although at a given time only a small fraction of these will have explicitly known dependencies and so appear in the evaluation queue. We observe that at the start of a job, the queue may contain $\mathcal{O}\left(10^4\right)$ tasks, and that this gradually decreases as the job progresses.

This parallelised version of our program is equipped with a check-pointing feature which allows jobs to be paused, saved to disk, and later resumed. The feature is implemented with the help of the Cereal library [143] and is useful for several reasons. Firstly, it is essential on computing clusters such as the Cambridge High-Performance Computing (HPC) cluster, which allows up to 1,200 CPUs to be used in parallel but imposes a time-limit of 36 hours on all jobs. Here, check-pointing allows jobs to run for many days or weeks, simply by pausing and resuming the job every 36 hours.[17] Secondly, we can use check-pointing to change the number of CPUs used by a job. Since, as mentioned above, the parallelisability gradually drops as the job progresses, we can avoid idle worker threads by using check-pointing to pause a job and restart it with a smaller number of CPUs. The job can be moved to a different machine before resuming, if desired. Thirdly, check-pointing allows backups to be kept during long-running jobs.

---

[16] I am grateful to Artjoms Iškovs for fruitful discussions on this topic.

[17] This strategy is not an underhand workaround of the Cambridge HPC policies and is in fact actively encouraged.

**Table 3.4** Running times and range of solutions in benchmark $C_1$ topology following successive improvements of IBP program. In all cases, $d_{max} = 1$.

| Program version | Maximum numerator power | Sectors solved | Time (in days) for slowest job |
|---|---|---|---|
| Original (single-core) | 4 | all | 34.5 |
| Lazily evaluated (single-core) | 5 | top sector | 71.5 |
| Lazily evaluated (parallelised) | 5 | all | 12.9 |

Parallelisation produced a good improvement in the speed and capability of our IBP program. In the planar topology $C_1$, where previously we exploited lazy evaluation to restrict ourselves to solving only the integrals appearing in the amplitude $q\bar{q} \to q'\bar{q}'g$, we were now able to remove this restriction and solve *all* $C_1$ integrals with up to 5 numerator powers and 1 squared denominator (i.e. $n_{max} = 5$ and $d_{max} = 1$). We believe this range of integrals is sufficient to solve any 2-loop 5-point massless planar QCD amplitude. Furthermore, despite the enlarged set of integrals being solved, the time required to solve the $C_1$ topology using the parallel version of the code (running on a cluster with up to 75 parallel CPUs) was under 2 weeks, compared to 2-3 months for the single-cored version. This can be seen in Table 3.4.

The performance gain from parallelisation was particularly evident in the non-planar topologies, $B_n$. Here, parallelisation enabled us to obtain solutions for a few of the most complicated non-planar IBP systems. Obtaining these results required thousands of CPU-days and would therefore have needed nearly a decade if running on a single CPU, whereas parallelisation allowed us to solve these systems in just a few weeks. A more detailed account will be given in Section 3.6.2.

## 3.6 Results

Having described our framework for solving the IBP equations, we will now discuss our results, firstly for the planar topologies, $C_n$, and secondly for the non-planar topologies, $B_n$. We will then describe the checks that we have performed on our results.

### 3.6.1 Planar results

In the planar topologies, we have solved all integrals in the ranges specified in Table 3.2, i.e. $(n_{max}, d_{max}) = (5, 1)$ for the $C_1$ topology and $(n_{max}, d_{max}) = (4, 1)$ for the $C_2$

**Table 3.5** Running times (in days) for the IBPs in $C_1$ topology. Only the 3-propagator masters are shown. Solutions range: $(n_{max}, d_{max}) = (5, 1)$. Using lazy evaluation and parallelisation across up to 75 CPUs.

| Master | Forwards stage | Backwards stage | Total |
|---|---|---|---|
| 1 | 0.6 | 2.0 | 2.5 |
| 2 | 0.7 | 12.2 | 12.9 |
| 3 | 0.7 | 7.7 | 8.4 |
| 4 | 0.4 | 0.7 | 1.1 |
| 5 | 0.2 | 0.4 | 0.7 |
| 6 | 0.3 | 1.0 | 1.3 |

topology. Each integral has been expressed as a linear combination of master integrals, with coefficients $c_{i,m}$ that are rational functions of the space-time dimension $d = 4 - 2\epsilon$ and the kinematic invariants $s_{ij} = (p_i + p_j)^2$. Our IBP solutions allow, in the leading-colour limit of QCD, the evaluation of any 2-loop 5-point massless amplitude such as $q\bar{q} \to q'\bar{q}'g$ and $q\bar{q} \to \gamma\gamma\gamma$. As a direct application of this, in Chapter 4 we will present the NNLO QCD corrections to 3-photon production at the LHC.

Although the strategy described in Section 3.3 enables the IBP system to be divided into many separate jobs (one per master integral), we observe (for planar and non-planar topologies alike) that the running times for different master integrals are vastly different. The calculation of the coefficients of the master integrals in the highest-weight sector (i.e. those with the maximum number of propagators in their denominator) is simplest and takes only a few minutes on a single CPU. The calculations corresponding to masters with fewer propagators are far more complex, however, and are many orders of magnitude slower. The projected IBP systems which are most difficult to solve are those corresponding to the masters of lowest weight. In the case of the topologies considered here, this means the masters with 3 propagators. Timings for the lowest-weight masters in the $C_1$ topology are shown in Table 3.5. Even though they have the same weight, the running times for these 6 masters span an order of magnitude.

The sizes of the solutions $c_{i,m}$ associated with different master integrals vary in a similar way to the variation of running times. Master 3 in the $C_1$ topology produced the largest solutions: they are 9.5 GB in size. The combined solutions for all $C_n$ integrals have a total size in excess of 80 GB.

### 3.6.2 Non-planar results

In view of the variation in running times for the IBP systems associated with different masters of a given topology, we chose to approach the non-planar ($B_1$ and $B_2$) IBPs from two opposing angles.

Firstly, we have solved IBP systems associated with the masters belonging to the highest-weight sector (with weight 8) for topologies $B_1$ (9 masters) and $B_2$ (3 masters). The IBP systems associated with these masters are computationally the simplest, and each required a few minutes of computation time while running on a single CPU. Here we have solved all integrals with numerator powers as high as 6 and/or a squared denominator (i.e. $n_{max} = 6$ and $d_{max} = 1$), which exceeds the range of integrals required for QCD amplitudes (see Table 3.2). The speed with which these results were obtained demonstrates an advantage of the strategy presented in Section 3.3.

Secondly, we have applied the lazily-evaluated, highly parallelised version of our program (described in Section 3.5.3) to the IBPs associated with the lowest-weight masters (with weight 3 or 4) in the $B_n$ topologies. The $B_1$ topology is the most complicated topology applicable to 2-loop 5-point massless amplitudes, and the lowest-weight masters in this topology give rise to the most computationally-demanding IBP systems. We have obtained solutions for a few of these most-complicated IBP systems. The solutions give the coefficients of masters 1, 5, and 14 for all $B_1$ integrals in the required range (i.e. $n_{max} = 4, d_{max} = 1$).[18] As shown in Table 3.6, these calculations are very computationally demanding, taking several weeks or months while running on tens or hundreds of CPUs. (We vary the number of CPUs over time in response to variations in the needs of a job. The *Max CPUs* column shows the peak number of CPUs used for a particular job.) The table clearly illustrates the benefits of the parallelisation of individual jobs as described in Section 3.5.3: if using a single CPU, we would have needed 8 years (i.e. 2,900 days) to solve the IBP system associated with master 1 in the $B_1$ topology. For some $B_1$ masters, we find that more than $10^4$ CPUs could usefully be employed in parallel if available, but the Cambridge HPC service limits us to a maximum of 1,200 parallel CPUs.

The results that we have obtained for the $B_1$ topology, particularly in relation to the weight-3 master integrals, suggest that the IBPs associated with the other masters

---

[18] In these IBP systems in the $B_1$ topology, to provide future flexibility, we ran the forwards stage using $(n_{max}, d_{max}) = (5, 1)$ and then used lazy evaluation during the backwards stage to only solve the integrals in the required range $(n_{max}, d_{max}) = (4, 1)$. We have retained the remaining parts of the lazy evaluation graph relating to $n_{max} = 5$. Should we wish to in the future, we can evaluate the remaining parts of the graph in order to extend our solutions to $(n_{max}, d_{max}) = (5, 1)$.

**Table 3.6** Non-planar solutions obtained for $B_1$ topology. Solutions range: $(n_{max}, d_{max}) = (4, 1)$.

| Master | Weight | Wall time *days* | Computing time *CPU days* | Max CPUs *CPUs* | Results size *GB* |
|---|---|---|---|---|---|
| 1 | 3 | 77.8 | 2,900 | 160 | 3.0 |
| 5 | 3 | 41.0 | 1,800 | 640 | 3.4 |
| 14 | 4 | 3.4 | 100 | 60 | 0.4 |

could similarly be solved in a reasonable time frame (i.e. a few months). We observe that although the non-planar results require far more CPU time than the planar results, there is no corresponding increase in the size of the final results. As we will see in Chapter 4, the planar IBP solutions are – although large – still of a size that allows them to be easily used in the numerical evaluation of amplitudes. We therefore expect it to be possible to make similar use of a full set of non-planar IBP solutions. However, we note that the $B_1$ and $B_2$ topologies each have 8 masters of weight 3. Based on our results, we estimate that solving the IBP systems associated with each of these masters would require $\mathcal{O}(10^5)$ CPU hours per master, and a full solution for the non-planar IBPs would therefore require a few million CPU hours.

### 3.6.3 Cross-checks and verification

Our results have been cross-checked in a number of ways. Firstly, the masters for all four topologies in Equation 3.9 have been independently identified using Reduze [112, 113].

Secondly, using the results in Refs. [118, 131] we have related all five $C_1$ integrals with irreducible numerators of power 5 appearing in the amplitude $q\bar{q} \to q'\bar{q}'g$ to integrals with lower numerator powers. Using our calculation for those integrals with lower numerator powers we find complete agreement with our direct calculation of the integrals with numerators of power 5. We have checked that this agreement holds for the projections on to the full set of masters in topology $C_1$. This is a highly non-trivial check for both our calculation and the results in Refs. [118, 131].

Thirdly, we have also checked that our calculation for topology $B_2$ agrees with the results in Ref. [130] by comparing all integrals with numerator powers of 4 (which is the highest numerator power computed in that paper). We find full agreement.

We note that Ref. [122] has claimed to compute the planar integrals with numerator power 5 with the help of the program Fire [111, 115]. However, since that reference does not provide explicit results or details about their calculation, we were unable to compare.

## 3.7 Conclusions

In this chapter, we have developed a framework for evaluating IBP equations, which are often a key bottleneck in the evaluation of multi-loop QCD amplitudes.

We introduced a strategy to break systems of IBP equations into multiple, smaller systems, each of which corresponds to one master integral. These smaller systems are simpler to solve and are mutually independent, which allows them to be solved in parallel to one another. However, most of the running time is spent on a small fraction of these, which are associated with the lowest-weight master integrals.

We designed and implemented an efficient `C++` program for solving IBP equations. The program incorporates the aforementioned strategy as well as a number of features that greatly improve the range of solvable integrals and the time required to compute them. In particular, arranging the symbolic algebraic operations into a network of lazily evaluated expressions produces an order-of-magnitude increase in speed. Parallelising the evaluation of the expressions in the lazy-evaluation network (which is unrelated to the parallelisation obtained by separating masters) produces a further speed increase of two orders of magnitude.

Our strategy and computer program enabled us to obtain new solutions for the planar and non-planar topologies of 2-loop 5-point massless integrals. We have cross-checked these results for consistency with previous results from other groups, where such results are available, and we find full agreement.

In the planar topologies, we have solved all $C_1$ integrals in the range $(n_{max}, d_{max}) = (5, 1)$, and all $C_2$ integrals in the range $(n_{max}, d_{max}) = (4, 1)$. Since the planar master integrals are already known, our solutions allow the evaluation of the leading-colour contribution to any 2-loop 5-point massless QCD amplitude such as $q\bar{q} \to q'\bar{q}'g$ and $q\bar{q} \to \gamma\gamma\gamma$. The NNLO QCD corrections to 3-photon production, which we will present in Chapter 4, are a direct application of these results.

In the non-planar topologies, our strategy of separating the IBPs by master integrals has allowed us to approach the problem from two opposing angles. In the IBP systems associated with the highest-weight $B_1$ and $B_2$ masters, we have solved all integrals in the range $(n_{max}, d_{max}) = (6, 1)$, which exceeds the range required for QCD amplitudes. In addition, we have obtained solutions in the range $(n_{max}, d_{max}) = (4, 1)$ for the IBPs associated with a few lowest-weight $B_1$ master integrals. The former required a few minutes of computation on a single CPU, whereas the latter required several weeks of evaluation on up to 600 CPUs.

As future work, one would like to be able to solve the subleading-colour contributions to 2-loop 5-point massless QCD amplitudes, which would require extending our non-planar solutions to cover all remaining $B_1$ and $B_2$ masters. The low-weight $B_1$ masters for which we have solved the IBPs are representative of the most difficult of these. Our results show that a full solution to non-planar IBPs (for the full ranges of $(n_{max}, d_{max})$ required for QCD amplitudes) is within the technical capabilities of our framework and is limited only by the availability of computing resources. Although a computation of this complexity will require a large quantity of CPU resources (a few million CPU hours, by our estimate), the high parallelisation offered by our program would allow a full solution to be obtained over the course of a few months by running on a computing cluster with several hundred CPUs. We observe that the most complicated $B_1$ solutions are similar in size to the most complicated planar solutions. The latter can be usefully employed in amplitude computations (see Chapter 4) and so we expect a full set of non-planar solutions to be of a size that would allow them to be used in a similar way.

# Chapter 4

# NNLO QCD corrections to three-photon production at the LHC

In this chapter, we compute the NNLO QCD corrections to three-photon production at the LHC. This is the first NNLO QCD calculation for a $2 \to 3$ process. Our calculation is exact, except for the scale-independent part of the two-loop finite remainder which is included in the leading-colour approximation. The required two-loop amplitude is calculated using an automated computational framework. The framework is generic and allows the calculation of any two-loop five-point massless planar amplitudes. We compare our predictions with the available 8 TeV measurements from the ATLAS collaboration and find that the inclusion of the NNLO corrections eliminates the significant existing discrepancy with respect to NLO QCD predictions, paving the way for precision phenomenology in this process.

## Declaration

This chapter is based on the paper [3] which was written in collaboration with M. Czakon, A. Mitov, and R. Poncelet. My primary contribution to this calculation was the creation of an automated computational framework for solving 2-loop 5-point massless amplitudes in terms of transcendental "pentagon functions". This framework is presented in Section 4.3 and one of its key components is the set of 2-loop 5-point IBP solutions obtained in Chapter 3.

## 4.1 Introduction

Over the last decade, Next-to-Next-to-Leading Order (NNLO) QCD calculations for hadron collider processes have sustained tremendous progress. Owing to the development of many independent approaches [56, 144–173] almost all non-loop-induced $2 \to 1$ and $2 \to 2$ processes have now been computed, typically in more than one computational approach. Such massive theoretical progress has led to the creation of public codes and has started to produce valuable and solid LHC phenomenology on a massive scale.

The computation of $2 \to 3$ hadron collider processes represents a natural step beyond the current state of the art in NNLO calculations. Since many of the available computational approaches are generic, they should in principle be able to handle the calculation of double real radiation in $2 \to 3$ processes. The calculation of the real-virtual correction to such processes should in principle also be possible, since numerically stable libraries are available for one-loop amplitudes. The only ingredient for such calculations which is not readily available are the two-loop five-point amplitudes. Thanks to the development of various new methods [2, 33, 108, 122, 130, 138, 139, 174–180], first results for selected helicities, colour structures or kinematics (typically Euclidean) have started to appear. This includes 5-point amplitudes computed in QCD [126–129, 181–187], in pure Yang-Mills [123, 125, 188], in supersymmetric theories [189–192] and in gravity [193–195]. As part of the work presented in this chapter, we calculate the two-loop planar amplitude $q\bar{q} \to \gamma\gamma\gamma$. This is the first time a 2-loop 5-point QCD amplitude has been derived explicitly, in analytic form, in the physical region. We discuss this result at length in Section 4.2.2 below. To calculate this amplitude we have built an automated computational framework, described in Section 4.3, which can be re-used to calculate other 2-loop 5-point massless amplitudes in an automated manner.

The goal of this work is to demonstrate the feasibility of existing calculational technologies to deal with $2 \to 3$ hadron collider processes. We have decided to apply this first-ever NNLO $2 \to 3$ calculation to the process $pp \to \gamma\gamma\gamma + X$. Our motivation for choosing this process is twofold. Firstly, the production of colourless final states has always occupied a special place among hadron collider processes and has been the pioneering work for both $2 \to 1$ and $2 \to 2$ processes. The calculation of the required two-loop amplitudes is made more feasible by the fact that the number of scales involved is relatively small.

Secondly, the process $pp \to \gamma\gamma\gamma + X$ is of direct phenomenological interest. The cross-section for three isolated photons at the LHC 8 TeV was measured in detail by the ATLAS collaboration [196] (and previously in Ref. [197]) and was found to be very significantly above the NLO QCD prediction in a wide kinematic region. Since at NLO the theory error is completely dominated by missing higher-order terms, this process represents a prime case for an NNLO QCD calculation. Indeed, we find that with the help of our calculation this discrepancy can be addressed, as can be seen in the results presented in Section 4.4.

The chapter is organised as follows. In Section 4.2 we explain our calculation, with an emphasis on the derivation of the two-loop amplitude. The automated computational framework developed by the author of this thesis for calculating 2-loop 5-point massless amplitudes is presented in Section 4.3. We then present in Section 4.4 our predictions for the fiducial and differential cross-section. The important question of perturbative convergence in this process is discussed in Section 4.4.4. Our conclusions are given in Section 4.5.

## 4.2 The calculation

In this work we follow the Stripper approach [152–154] previously applied at NNLO in QCD to top-pair [55, 86, 198–200] and inclusive jet [201] production at the LHC. The approach is implemented in a fully-differential partonic Monte Carlo program which can calculate any infrared-safe partonic observable. The technical details about its implementation can be found in Ref. [201].

The complete calculation converges very well in terms of phase-space integration. Not counting the CPU time needed to evaluate the two-loop finite remainder (see Section 4.2.2 for details), it took only about 2k CPU hours to complete. The slowest contribution (about 1k CPU hours) is the real-virtual finite contribution due to the slow evaluation of the 6-point one-loop amplitude with OpenLoops 1. That contribution, however, converges fast in terms of required phase-space points.

The ingredients needed for the present calculation are tree-level amplitudes as well as the finite remainders of one-loop and two-loop amplitudes. Their calculation is described in detail in Section 4.2.1 and Section 4.2.2, respectively. Here we only point out that all required one-loop amplitudes are included exactly, with full colour dependence. The finite remainder of the two-loop amplitude $q\bar{q} \to \gamma\gamma\gamma$ is included in the leading-colour approximation, additionally excluding diagrams with closed fermion loops. The justification for this approximation is given in Section 4.2.2 below.

The infrared subtraction operator (sometimes called $Z$-operator) is given in Ref. [154] and its leading-colour approximation can be found in Ref. [182]. We work in a theory with 5 massless active quark flavours and renormalise the amplitudes accordingly. No loops with massive fermions are included in our calculation. Their effect in the context of diphoton production has been discussed in Ref. [202].

### 4.2.1 Tree-level and one-loop amplitudes

All tree-level diagrams are computed with the help of the library AVHLIB [203, 204]. For the derivation of the two-loop finite remainder, the one-loop amplitude $q\bar{q} \to \gamma\gamma\gamma$ is needed to order $\varepsilon^2$ (where $d = 4 - 2\varepsilon$ is the space-time dimension). We have computed it following the standard Feynman diagram plus Integration-by-Parts (IBP) identities [96, 97] approach. All required master integrals expanded to that order in $\varepsilon$ are available in electronic form in Ref. [184]. The finite remainders for all one-loop amplitudes are obtained from the library OpenLoops [205, 206], while the one-loop squared $q\bar{q} \to \gamma\gamma\gamma$ contribution is taken from the library Recola [207].

Unlike the case of diphoton production, the gluon-initiated one-loop amplitude $gg \to \gamma\gamma\gamma$ vanishes and thus does not contribute to the process studied in this paper. Since the $gg$-flux is sizeable, the vanishing of this contribution is of phenomenological significance and we will discuss it in more detail in Section 4.4.

### 4.2.2 The two-loop amplitude for $q\bar{q} \to \gamma\gamma\gamma$

An important novelty in this work is the calculation of the two-loop amplitude for the process $q\bar{q} \to \gamma\gamma\gamma$. Although our calculation is restricted to the leading-colour approximation, this is the first time a two-loop five-point amplitude is put in a form that can be used in a phenomenological application. For this reason we describe it in detail in this section.

#### Structure of the two-loop amplitude

We need the two-loop amplitude $|M^{(2)}(q\bar{q} \to \gamma\gamma\gamma)\rangle$, multiplied by the Born amplitude $|M^{(0)}(q\bar{q} \to \gamma\gamma\gamma)\rangle$, and summed/averaged over helicities and colour. Its colour decomposition reads:

$$\begin{aligned}\overline{\sum} 2\mathrm{Re}\langle M^{(2)}|M^{(0)}\rangle &= M^{(\mathrm{lc},1)}\left(N_c^3 - 2N_c + 1/N_c\right) + M^{(\mathrm{lc},2)}\left(N_c^3 - N_c\right) \\ &\quad + M^{(\mathrm{f})}\left(N_c^2 - 1\right) + M^{(\mathrm{np})}\left(N_c - 1/N_c\right) , \qquad (4.1)\end{aligned}$$

where $N_c = 3$ is the number of colours.

In this work we simplify the calculation by employing the following approximation:

$$\overline{\sum} 2\mathrm{Re}\langle M^{(2)}|M^{(0)}\rangle \approx N_c^3\left(M^{(\mathrm{lc},1)} + M^{(\mathrm{lc},2)}\right) , \qquad (4.2)$$

i.e. we neglect the non-planar contribution $M^{(\mathrm{np})}$ as well as all contributions $M^{(\mathrm{f})}$ with a fermion loop (both planar and non-planar).

The non-planar contribution $M^{(\mathrm{np})}$ is suppressed by a factor of $1/N_c^2$ relative to the leading-colour contributions. It is thus expected to be numerically subdominant. The non-planar contribution cannot be computed at present since the required IBP solutions (topologies $B_1$ and $B_2$ in the notation of Chapter 3) are not yet fully known.

The contribution $M^{(\mathrm{f})}$ contains all diagrams with one closed fermion loop. Both planar and non-planar diagrams contribute to it. It cannot be currently derived since the required non-planar IBPs (specifically, the topology $B_2$) are not yet known. The term $M^{(\mathrm{f})}$ is suppressed with respect to the leading-colour terms by a single power of $1/N_c$. At the same time some diagrams[1] are enhanced by the number of massless fermion flavours $n_f = 5$. Therefore, although in the strict $N_c \to \infty$ limit $M^{(\mathrm{f})}$ is suppressed relative to the leading-colour contribution, its numerical value may not necessarily be sub-dominant with respect to Equation 4.2. For this reason, to be conservative, one should assume that it is comparable numerically to the leading-colour contribution. As we show in Section 4.4 below (see in particular Fig. 4.15), the numerical impact of the leading-colour approximation Equation 4.2 to the differential cross-section is itself negligible, at the percent level, which *a posteriori* justifies the approximation $M^{(\mathrm{f})} \approx 0$. In the future, once the contributions $M^{(\mathrm{f})}$ and $M^{(\mathrm{np})}$ become known, we can easily update our cross-section predictions.

### Calculation of the two-loop amplitude

We will now give a detailed step-by-step account of the calculation of the two-loop amplitude. In Section 4.3, we will describe an automated framework that we have

---

[1] These are the diagrams that involve no photon coupling to the closed fermion loop.

written to automatically perform most of these steps. We have verified that the results of the automated framework exactly match those obtained by sequentially following the steps in the manner described here.

To compute the two-loop amplitude we use a standard Feynman diagram-based approach. The diagrams are generated with the help of a private software. After multiplying with the Born amplitude and then computing the traces of spin tensors and colour factors, the resulting scalar integrals are mapped to master integrals using the IBP results of Chapter 3. Next, one must insert of the results for the required master integrals. To this end we utilise the results of Ref. [33] where a set of integrals has been explicitly computed in terms of the so-called pentagon functions $f_{ij}$. This set of integrals can be algebraically related to our set of master integrals with the help of the IBP solutions derived in Chapter 3.

At this point the bare amplitude can be computed numerically using the routines for the numerical evaluation of the set of integrals provided with Ref. [33]. We do not follow this approach here for two reasons. Firstly, we would like to provide an explicit analytic result in terms of basic functions, like the set of pentagon functions. Secondly, the complete result involves not just the 61 master integrals but also many integrals that are obtained from them by crossings of the external legs. In practice we have more than 90 sets of crossings that need to be applied to the set of master integrals. While not every master integral will need to be crossed for all crossings, the complete set of integrals, accounting for all crossings, far exceeds the dimension of functions needed to describe the amplitude. For this reason such an approach would not be minimal and could lead to more severe loss of precision during the numerical evaluation.

For the above reasons, we use the explicit representation of master integrals in terms of pentagon functions $f_{ij}$ from Ref. [33] and have applied the momentum crossings directly to those functions. To minimise the set of functions, we have derived various functional identities between those functions with different arguments. The derivation of these identities is described in Section 4.3.1 and we list the identities in Appendix A. As a result, we have derived an explicit expression for the squared amplitude (defined in Equation 4.2) as a polynomial in transcendental constants and $f_{ij}$ functions with various arguments. The coefficients of this polynomial are rational functions of the kinematic invariants. We have factorised and simplified them, in some cases using the finite-field reconstruction package FiniteFlow [38].

Besides the usual $\zeta(2)$ and $\zeta(3)$, a new set of constants collectively called `bc4` appears at weight 4 [33]. Their treatment requires special attention. These constants are associated with the master integrals at weight 4 and, despite being called *constants*, they in general vary with kinematic region. We have accounted for this possibility in the process of applying momentum crossings to the master integrals. Many of these constants take the same value in the various physical regions. We have used the numerical values which are included in the numerical code accompanying Ref. [33]. Similarly, the analytic continuation of the pentagon functions $f_{ij}$ across the various physical regions is performed automatically by the numerical library of Ref. [33]. To check the correctness of our manipulations, we have compared in each physical region the numerical predictions for each master integral, constructed by us as described above, with the numerical value for the master returned directly by the library of Ref. [33] and have found complete agreement. We have also checked many integrals against the numerical program pySecDec [208], finding agreement in all cases.

In summary, we have expressed the complete analytic result for the bare 2-loop amplitude in a basis of around 1800 transcendental terms involving $\zeta(2), \zeta(3)$ and $f_{ij}$ functions plus about 100 terms involving `bc4` “constants” of weight 4.

### Evaluation of the two-loop amplitude

Most of the rational coefficients in the two-loop amplitude are small (i.e. kB size) but some exceed 1MB. When numerically evaluating the amplitude to calculate cross-sections, the loss of numerical precision due to cancellation between the various terms is thus of particular concern. To minimise such cancellation we evaluate all rational coefficients with exact rational arithmetic. Specifically, we rationalise each phase-space point to preserve the accuracy of the original floating point numbers, and then use this rational form to compute the rational coefficients as exact rational numbers. This is implemented with the help of the CLN library [209]. The evaluation is much slower than an evaluation in double precision would be, yet the overall timing is negligible compared to the evaluation of the slowest pentagon functions of weight 4. We have performed various tests for the depth of numerical cancellations and have found them to be under control in all test cases.

The numerical evaluation of the functions $f_{ij}$ is performed with the help of the `C++` library provided with Ref. [33]. The time it takes to evaluate these functions depends strongly on their weight. All functions through weight 3 are standard polylogarithms and can easily be computed with full double precision in negligible time. The functions of weight 4 are the slowest and can take several minutes per phase-space point. Their precision is less than full double precision due to conflicting requirements of precision and speed as well as the numerical stability of the integration routines used for their calculation. With the help of extensive experimentation we have found that computing them with at least 7 significant digits is sufficient for our purposes. To test the depth of numerical cancellations we have also computed the weight 4 functions requiring 5 significant digits. This results in a finite remainder with, typically, 2 significant digits.

It takes about 10-50 min, depending on the phase space point, to compute the finite remainder at a single phase-space point on a single CPU (i.e. without any parallelisation). The average time is about 17 min when a relative precision of $10^{-7}$ for the weight 4 functions is requested. In general, several hundred thousand points are required in order to integrate the three-photon phase space over the required bins. Such a calculation requires significant, cluster-size computer resources. While a one-off evaluation is possible it poses non-trivial problems, especially if re-evaluation of the amplitude is needed (for example for a different setup or collider energy). To minimise this computational effort we have utilised a two-fold strategy.

Firstly, we have produced an optimised set of phase-space points which have been generated according to the Born cross-section. Such an approach has already been used in Refs. [210–214] and it allows us to obtain a good quality double-virtual contribution with a reduced number of events. In this case, we have computed 30k events.

Secondly, for the implementation of two-loop finite remainder we have employed an approach whereby the above-mentioned 30k phase space points have been used to construct a (four-dimensional) interpolating function for the real part of the finite remainder. Constructing multi-dimensional interpolating functions is a hard problem. In our case we have used the purposely-developed library GPTree [215] which uses advances in machine learning to optimise the interpolation tables and to produce an estimate of the interpolation error. The output of the GPTree library is a `C++` library which is portable and very easy to link to a `C++` code and to use. It has the advantage that if more phase-space points are computed in the future, the interpolation tables can be refitted and thus be further improved. We have also found it useful as an additional means to monitor for the appearance of numerical instabilities.

In view of the phenomenological application of the two-loop amplitude Equation 4.1 — see Section 4.4 — we make explicit the scale dependence of its finite reminder $\mathcal{H}^{(2)}$:

$$\mathcal{H}^{(2)}(\mu_R^2) = \mathcal{H}^{(2)}(s_{12}) + \sum_{n=1}^{4} c_n \ln^n\left(\frac{\mu_R^2}{s_{12}}\right) , \tag{4.3}$$

where $s_{12}$ is the partonic centre-of-mass energy squared. Since the coefficients $c_n$ can be determined exactly from the tree-level and one-loop amplitudes, throughout this work the scale dependence of the two-loop finite reminder is included with full colour dependence. Therefore, the approximation Equation 4.2 is applied only to the first term of the RHS of Equation 4.3. As can be seen in Fig. 4.15 below, the numerical impact on the NNLO cross-section of the scale-independent part of the two-loop finite remainder $\mathcal{H}^{(2)}(s_{12})$ is rather small, at the percent level.[2]

## 4.3 Amplitude reduction to pentagon functions

Having given a detailed overview of our calculation, in this section we will describe our framework for the automated reduction of amplitudes to pentagon functions. Our framework is designed to work with any two-loop five-point massless planar amplitudes. If an amplitude has not been summed/averaged over helicities, it needs to have been decomposed into helicity structures, each of which we will then treat as a separate amplitude.

As a preparatory step, it was important to construct a minimal basis of pentagon functions. If one were to instead use a non-minimal set of functions, experience suggests that the resulting expressions for the amplitude would be large in size and therefore more computationally expensive to derive and to use. To construct a minimal set of functions, we derived identities between the pentagon functions so as to eliminate any basis functions that were redundant. Our method for deriving these identities is presented in Section 4.3.1 and the resulting identities are tabulated in Appendix A.

Our main amplitude-reduction framework is based around the finite-field approach, which was introduced in Section 1.10. Calculations within this approach require two ingredients: firstly, an algorithm for the reconstruction of exact symbolic amplitudes from a large number of numerical (i.e. finite-field) evaluations, and secondly an automated method for performing the finite-field evaluations themselves. In Section 4.3.2 we will describe our work on the first problem – interpolation and reconstruction algorithms – after which the remainder of this section will be devoted to our automated framework for performing numerical (i.e. finite-field) evaluations of 2-loop 5-point massless planar amplitudes.

---

[2] The other NNLO contributions – namely the double-real, real-virtual, and scale-*dependent* double-virtual corrections – turn out to be much larger, causing the NNLO prediction for the fiducial cross-section to exceed the NLO prediction by 60%, as we will see in Section 4.4.

### 4.3.1 Pentagon function identities

The pentagon functions, labelled $f_{j,k}(v_1, v_2, v_3, v_4, v_5)$, appear in the amplitude with various cyclic and non-cyclic permutations of their arguments $v_i$. Some of the $f_{j,k}$ can be directly mapped onto the polylogarithms $\text{Li}_n$ which were described in Section 1.8 of Chapter 1 and for which there are many well-known identities, such as Equation 1.42. The higher-weight functions, particularly $f_{4,k}$, are more complicated but are still expected to obey identities of a similar nature. Although analytical methods could in principle be employed to derive and prove some of these identities, here we adopted a numerical (floating-point) approach. The resulting identities are, however, exact. For the avoidance of confusion, we emphasise that these identities were derived with the help of floating-point numerics, since the pentagon functions are irrational, whereas the rest of our amplitude-reduction framework (described in later sections) uses finite-field numerics.

Our starting point is the observation that polylogarithms and their generalisations can be assigned a transcendental weight, $w \in \mathbb{N}$, and that the known identities are linear combinations (with coefficients in $\mathbb{Q}$) of terms of uniform transcendental weight. In the case of the pentagon functions, the weight of $f_{j,k}$ is simply $j$. The Riemann zeta function values $\zeta_n \equiv \zeta(n)$, which sometimes appear in these identities, also have transcendental weight: $\zeta_n$ has weight $n$. We treat these constants on an equal footing with the $f_{j,k}$ functions. If two functions of weight $w_1$ and $w_2$ are multiplied, the product has weight $w_1 + w_2$.

To find identities, we begin by picking a weight, $n$, and constructing a set $S_n$ of all weight-$n$ functions. This includes the pure weight-$n$ functions, as well as products of lower-weight functions whose weights add to give $n$. For example, if $n = 2$ we construct a set $S_2$ containing:

1. the pure weight-2 functions of the form $f_{2,1}(a, b)$
2. the pure weight-2 constant $\zeta_2 = \frac{\pi^2}{6}$
3. products of two weight-1 functions: $f_{1,1}(a)f_{1,1}(b)$.

Here, $a$ and $b$ can each be any of the 10 kinematic invariants $s_{ij}$. We then randomly generate a set $K$ of kinematic points, where a kinematic point corresponds to assigning a floating-point value to each of the 10 kinematic invariants $s_{ij}$. We need the set $K$ to be larger than the set $S_n$. At each kinematic point $k \in K$, we evaluate all functions in $S_n$ numerically (i.e. with floating-point arithmetic). These evaluations can be entered in a matrix, where each row corresponds to a kinematic point $k$ and each column corresponds to an element of our set $S_n$. The identities that we seek can be identified from this matrix by finding columns that are linear combinations of other columns. In other words, the vectors in the kernel of this matrix directly represent pentagon function identities.

To calculate the kernel vectors, we find the MATHEMATICA function `NullSpace` to be useful, although we observe that it performs poorly unless the kernel is very small. This is especially true when working with the functions $f_{4,k}$, whose evaluation precision is lower than that of the lower-weight functions. We therefore take an iterative approach, initially restricting ourselves to a subset $\tilde{S}_n$ of $S_n$ and using `NullSpace` to find any identities in this restricted set of functions. Each identity found allows us to remove one redundant function from $\tilde{S}_n$. We then insert a few more functions from $S_n$ into $\tilde{S}_n$ and search for further identities. This process is repeated until all the functions from $S_n$ have been inserted into $\tilde{S}_n$. The MATHEMATICA function `MatrixRank` is helpful for verifying that the final matrix has full rank, i.e. that the functions in the final set $\tilde{S}_n$ are not related by any further identities. As an aside, we note that since the pentagon function identities are linear (and real), one can separately apply the identities to the real and imaginary parts of the elements of $S_n$. In practice, we derive the identities using the real parts alone, and then confirm that the same identities correctly describe the imaginary parts.

In deriving the identities, it was sometimes useful to work in the *Euclidean* kinematic region, where $s_{ij}$ is positive if $i$ is an incoming particle and $j$ is outgoing (or vice-versa) and negative otherwise. In this region, the functions $f_{j,k}$ can be calculated more quickly and to a higher precision; they also have no imaginary part. As a result, many of the identities can more easily be obtained in the Euclidean region and later extended to the physical kinematic region. However, not all 10 invariants $s_{ij}$ can simultaneously lie in the Euclidean region, and so working in this region requires restricting ourselves to a subset of $S_n$. Some of the identities that we have derived only appear when working in the physical kinematic region, where all $s_{ij}$ are positive.

All of the identities obtained are presented in Appendix A and it can be seen that all appearing numerical coefficients are simple rational numbers such as $\frac{1}{2}$, $\frac{33}{8}$, and $\frac{51}{4}$. We note that a few of the identities require introducing new “constants” $h_{j,k}$ which vary with the kinematic region, just like the `bc4` constants. We expect that it should be possible to either express the $h_{j,k}$ in terms of `bc4`, or instead absorb the $h_{j,k}$ into the corresponding functions $f_{j,k}$, although we have not done so here. All of the identities listed have been verified numerically using new random kinematic points. We find the identities to be correct at the expected level of precision: at least 6 digits for the weight-4 identities, and at least 12 digits for the weight-2 and weight-3 identities.[3]

### 4.3.2 Finite-field interpolation and reconstruction

Let us now turn to the problem of finite-field interpolation and reconstruction, a key ingredient in the finite-field approach to calculating amplitudes. Recent months have seen the emergence of several public [37–39] and private libraries that can interpolate and reconstruct multi-variate polynomials and rational functions from their numerical values in a finite field. We ultimately opted to use the FireFly library [37, 39] for this purpose, which allowed us to focus on the other half of the problem: performing numerical (i.e. finite-field) evaluations of the amplitudes, as we will discuss in Sections 4.3.3 onwards. In this section, however, we would like to describe our work on devising an efficient method for *polynomial interpolation in finite fields*, which is an important ingredient in the above libraries. Our method combines several ideas from the field of computer algebra: Fast Fourier Transforms (FFTs) [34, 216], the Number-Theoretic Transform [217, 218], and the use of Fourier transforms for polynomial interpolation [34, 219, 220]. A method similar to ours was already known in the computer algebra literature (see, for instance, Sections IX.1.2 and IX.1.3 of Ref. [221]), although our work was carried out independently of it and we believe that our proposal to apply such a method to particle physics calculations is new. Indeed, FFT-based techniques are not used in any of the above interpolation libraries, as far as we are aware, and we hope that this work can aid their future development.

Conceptually, polynomial interpolation is an easy problem to solve. Suppose there is a polynomial $q(x)$ of degree $N$ whose explicit form we do not know, but whose values $y_j = q(x_j)$ we know at some points $\{x_0, \ldots, x_N\}$. We can write down an ansatz

$$q(x) = \sum_{i=0}^{N} a_i x^i, \tag{4.4}$$

where $a_i$ are unknown coefficients. This gives us a linear system of equations

$$y_j = \sum_{i=0}^{N} a_i x_j^i \tag{4.5}$$

---

[3] As mentioned earlier, the weight-2 and weight-3 functions are evaluated at double floating-point precision, whereas the weight-4 functions are evaluated at single floating-point precision.

which we can solve using Gaussian elimination to obtain the unknown coefficients $a_i$. Although this method is easy to implement, its performance is poor because Gaussian elimination requires $\mathcal{O}(N^3)$ operations. We wrote a simple implementation for this, which we found to take ∼1 second to interpolate a polynomial where $N = 10^3$, but several minutes when $N = 10^4$. This is consistent with the $\mathcal{O}(N^3)$ scaling and we would furthermore expect it to take several days if $N = 10^5$, and several decades if $N = 10^6$.

Several of the finite-field libraries used in High-Energy Physics implement the so-called Newton interpolation method, whose running time scales as $\mathcal{O}(N^2)$. A more detailed discussion of the Newton method can be found in [37, 38]. While $\mathcal{O}(N^2)$ is clearly an improvement, we will present a method that allows the reconstruction to be performed in *quasi-linear* time: $\mathcal{O}(N \log N)$. Our method is inspired by the fast algorithm [219, 220] for interpolating polynomials over $\mathbb{C}$, which we adapt[4] to work with polynomials over $\mathbb{F}_p$. In a test implementation that we wrote, this method was able to interpolate a polynomial with $N = 10^6$ in less than 1 second.

Our method is based on the well-known Fast Fourier Transform (FFT) class of algorithms, which can compute the Fourier transform

$$\tilde{y}_m = \sum_{n=1}^{N} e^{2\pi i m n/N} y_n \tag{4.6}$$

of a sequence of numbers $\{y_1, \ldots, y_N\}$ in $\mathcal{O}(N \log N)$ time. The standard algorithm [216] recursively divides the sequence into two interleaved halves and computes the Fourier transform of each half separately before merging them to obtain the Fourier transform of the original sequence. This requires $N$ to be divisible by a large power of 2, although generalisations exist that merely require $N$ to have many divisors.

Although the most familiar applications of Fourier transforms use complex numbers (i.e. $y_j \in \mathbb{C}$), it is known from number theory that one can also apply a Fourier transform to sequences of finite-field numbers (i.e. $y_j \in \mathbb{F}_p$). To be able to do so, the finite field needs to contain a primitive root of unity, which is analogous to the complex root of unity $e^{2\pi i/N}$. We define a number $u$ to be a primitive $N$-th root of unity if the elements in the set $\{u, u^2, \ldots, u^N\}$ are mutually distinct and $u^N = 1$. For such a root to exist in a field $\mathbb{F}_p$, a necessary[5] and sufficient condition is that $N$ divides

---

[4] As mentioned above, similar methods can be found in the computer algebra literature, e.g. Ref. [221], although our work was carried out independently.

[5] Lagrange's theorem proves this. The elements (excluding 0) of a finite field $\mathbb{F}_p$ form a multiplicative group of order $p - 1$, while the elements $\{u, u^2, \ldots, u^N\}$ constitute a subgroup of order $N$.

**Table 4.1** Examples of prime fields $\mathbb{F}_p$ for fast polynomial interpolation

| $p$ | Largest $N$ permitted | Primitive $N$-th root in $\mathbb{F}_p$ |
|---|---|---|
| $15 \cdot 2^{27} + 1$ | $2^{27}$ | 11 |
| $17 \cdot 2^{27} + 1$ | $2^{27}$ | 3 |
| $13 \cdot 2^{28} + 1$ | $2^{28}$ | 3 |
| $3 \cdot 2^{30} + 1$ | $2^{30}$ | 5 |
| $29 \cdot 2^{57} + 1$ | $2^{57}$ | 3 |
| $71 \cdot 2^{57} + 1$ | $2^{57}$ | 3 |
| $75 \cdot 2^{57} + 1$ | $2^{57}$ | 7 |
| $27 \cdot 2^{59} + 1$ | $2^{59}$ | 5 |

$p-1$. Recalling that the standard FFT algorithm requires $N$ to be divisible by a large power of 2, our fast interpolation method requires the finite field $\mathbb{F}_p$ to be specially chosen such that $p-1$ is divisible by a large power of 2 (and, of course, $p$ itself must be prime).[6] Some examples of suitable choices of $\mathbb{F}_p$ are given in Table 4.1.

Having set out a way to compute FFTs in finite fields, we can apply this to the problem of polynomial interpolation. The key observation[7] is that if we choose to sample our polynomial $q(x)$ at the specially-chosen values $x_j = u^j$, where $u$ is a primitive $N$-th root of unity as defined above, then the linear system to be solved (Equation 4.5) acquires the form of a Fourier transform (Equation 4.6). In short, by choosing our prime field $\mathbb{F}_p$ such that $p-1$ is divisible by a large power of 2 and by choosing to evaluate our polynomial at the roots of unity $x_j = u^j$, we can interpolate the polynomial in $\mathcal{O}\left(N \log N\right)$ operations by performing an FFT.

In a practical calculation, there may be finite-field points at which it is not possible to evaluate the amplitude. In particular, there are usually poles present in the IBP solutions, master integral solutions, and the amplitude as a whole. Even though a polynomial, by definition, has no poles, polynomial interpolation is often required as part of a larger problem, such as the interpolation of rational functions. Since our method requires the polynomial to be evaluated at the roots of unity, rather than at arbitrary points, we need a procedure to handle cases where some of the required points happen to coincide with poles.

---

[6] Ideally, one would like primes $p$ such that $p-1=2^N$. These are known as *Fermat primes* and surprisingly, only five are known: $(2^1+1)$, $(2^2+1)$, $(2^4+1)$, $(2^8+1)$, $(2^{16}+1)$. It has been conjectured that no other Fermat primes exist.

[7] This obervation was originally made in the context of interpolating polynomials over $\mathbb{C}$. [219, 220]

Let us start by considering the simple case where only one root of unity, say $x_k$, coincides with a pole. The unknown polynomial $q(x)$ can be broken into a sum of two linearly-independent polynomials:

$$q(x) = q_0(x) + \lambda_1 q_1(x), \tag{4.7}$$

where $\lambda_1$ is an arbitrary coefficient to be found, and the polynomials $q_0(x)$ and $q_1(x)$ are required to have the following values when $x$ is a root of unity:

$$q_0(x_j) = \begin{cases} q(x_j) & \text{if } j \neq k \\ 0 & \text{if } j = k \end{cases} \tag{4.8}$$

$$q_1(x_j) = \begin{cases} 0 & \text{if } j \neq k \\ 1 & \text{if } j = k. \end{cases} \tag{4.9}$$

Since $q_0(x)$ and $q_1(x)$ can be evaluated at all roots of unity, they can both be interpolated using our FFT-based method. If we now choose a random new point $\tilde{x}$ that is *not* a root of unity (and furthermore is not a pole) and evaluate $q_0\left(\tilde{x}\right)$, $q_1\left(\tilde{x}\right)$, and $q\left(\tilde{x}\right)$, we can substitute these values into Equation 4.7 to determine $\lambda_1$ and thus obtain an analytical expression for $q(x)$, as required.

This procedure can be generalised to the case where several roots of unity, say $\{x_{k_1}, \ldots, x_{k_n}\}$, coincide with poles. We can generalise Equation 4.7 in the following way:

$$q(x) = q_0(x) + \sum_{i=1}^{n} \lambda_i q_i(x), \tag{4.10}$$

requiring $q_0(x)$ and $q_i(x)$ to have the following values when $x$ is a root of unity:

$$q_0(x_j) = \begin{cases} q(x_j) & \text{if } j \notin \{k_1, \ldots, k_n\} \\ 0 & \text{if } j \in \{k_1, \ldots, k_n\} \end{cases} \tag{4.11}$$

$$q_i(x_j) = \begin{cases} 0 & \text{if } j \neq k_i \\ 1 & \text{if } j = k_i, \end{cases} \tag{4.12}$$

where $i > 0$. As before, we interpolate $q_0(x)$ and $q_i(x)$ using our FFT-based method. By evaluating Equation 4.10 at any $n$ new points that are not roots of unity, we obtain a system of $n$ linear equations constraining the values of $\{\lambda_i\}$. Since we expect $n \ll N$, this small system can straight-forwardly be solved by Gaussian elimination to determine the values of $\{\lambda_i\}$.

This completes our method for the efficient interpolation of polynomials in finite fields. As mentioned, our method can interpolate polynomials with $N = 10^6$ in less than 1 second, whereas an approach based on Gaussian elimination would take many years to solve the same problem. There now exist a number of libraries for finite-field interpolation and we ultimately opted to use the FireFly library [37, 39]. The rest of Section 4.3 will therefore focus on the production of numerical (finite-field) results for 2-loop 5-point massless amplitudes, which are then supplied to FireFly in order to obtain analytical results for the amplitude. Nevertheless, we hope that the efficient method for polynomial interpolation that we have presented here can aid the future development of these interpolation libraries.

### 4.3.3 Epsilon expansion

We would like our final amplitude to be produced as a Laurent series in $\epsilon$. One approach would be to treat $\epsilon$ on an equal footing with the kinematic invariants $s_{ij}$. The rational functions appearing in the reconstructed amplitude would then be functions of $\epsilon$ and $s_{ij}$. After reconstructing these rational functions, they could be expanded in $\epsilon$ in order to obtain the desired Laurent series. However, we find that this approach would cause the degree of the numerators and denominators of the rational functions to increase by as much as 20, resulting in a need for many more numerical evaluations in order to perform the reconstruction.

It is interesting to consider whether Laurent expansions could be obtained numerically when working in finite fields. Specifically, one might hope to treat $\epsilon$ as a numerical (i.e. finite-field valued) variable in the numerical implementation of the amplitude and then obtain the Laurent expansion of the amplitude at a (finite-field valued) kinematic point by combining a few numerical probes of the amplitude at different values of $\epsilon$. Unfortunately, such an approach is precluded by the presence of poles in $\epsilon$ at intermediate stages of the computation (i.e. in the IBP solutions, in the solved master integrals, or in the coefficients of the unreduced amplitude), which prevents us from performing numerical evaluations at $\epsilon = 0$. If $\epsilon$ were a real or complex number, we might try to compute limits of the form

$$\lim_{\epsilon \to 0} f(\epsilon) \tag{4.13}$$

but in our case $\epsilon$ takes discrete values in a finite field, and so the meaning of such a limit is not defined. In fact, let us consider the simple case where $f(\epsilon)$ is a polynomial but we can perform evaluations only at non-zero values of $\epsilon$. For such a case, it can be shown that the only way to obtain any information about $f(0)$ is to first perform sufficiently many evaluations to fully interpolate the polynomial $f(\epsilon)$.[8] This, of course, would remove all of the simplification and efficiency that we hoped to gain in producing the amplitude in the form of a series in $\epsilon$.

Therefore, we instead used MATHEMATICA to *analytically* $\epsilon$-expand (as will be described below) the required IBPs, masters, and amplitude coefficients; their numerical evaluation in a finite field thus produces series in $\epsilon$ rather than pure finite field numbers. Thus, while our `C++` numerical implementation of the amplitude uses finite-field values for the kinematic invariants $s_{ij}$, it keeps $\epsilon$ as a symbolic variable and must be able to dynamically add and multiply Laurent series in $\epsilon$, each of which takes the form

$$\sum_{i=n}^{N} a_i \epsilon^i + \mathcal{O}\left(\epsilon^{N+1}\right), \tag{4.14}$$

where each $a_i$ is a finite-field number. We implement a small `C++` class to represent such a Laurent series. For each series, we store the leading power $n$ together with the $N - n + 1$ coefficients $\{a_i\}$. We perform addition and multiplication on these series by implementing the same procedures that one uses when adding or multiplying Laurent series by hand. We take care to include an extra check after each addition or multiplication, verifying that the highest power $N$ of the output series is consistent with the highest powers of the input series. The underlying finite-field arithmetic on the numbers $a_i$ is performed using the tools supplied with the FIREFLY library.

### 4.3.4 Evaluation of the IBPs

The numerical (i.e. finite-field) evaluation of an amplitude requires us to numerically evaluate the IBP solutions in $\epsilon$-expanded form, as mentioned in Section 4.3.3. The analytical IBP solutions produced in Chapter 3 were rational functions of the space-time dimension $d$ (and of the kinematic invariants $s_{ij}$) so here we have used MATHEMATICA to set $d = 4 - 2\epsilon$ and to expand to ten orders in $\epsilon$. In fact, for the 3-photon amplitude we find that it is sufficient to expand to five orders in $\epsilon$, and during the numerical evaluation of the amplitude we obtain a 2-fold increase in speed and RAM usage by

---

[8] This fact is the basis of a clever cryptographic scheme [222] which allows a secret message to be divided into $N$ pieces in such a way that any $k$ pieces (for a pre-determined value $k < N$) can be used to reconstruct the secret message, while nothing can be deduced about the message by an attacker who has fewer than $k$ pieces.

discarding the remaining terms. Although we obtained solutions to ∼750,000 integrals in Chapter 3, we find that only $\mathcal{O}\left(1\%\right)$ of these contribute to the 3-photon amplitude. We therefore opt to $\epsilon$-expand only these specific integrals. This procedure is fully automated, allowing the $\epsilon$-expansion to easily be performed for any additional integrals required for future amplitude calculations.

As mentioned in Section 4.2.2, many of the integrals in the amplitude appear with crossings applied to the external legs. One approach to computing these integrals would be to apply these momentum crossings analytically to all IBP solutions (e.g. using Mathematica), as a preparatory step before starting any numerical calculations. Here, however, we instead choose to simply apply the appropriate crossing to the kinematic variables $s_{ij}$ during the numerical evaluation of the IBPs associated with these crossed integrals. If an integral appears multiple times in an amplitude, with different crossings, we evaluate the IBPs independently at each crossing. (The crossings must also be applied to the master integrals, as will be described below.) Since in Chapter 3 we had set $s_{12} = 1$, we restore the missing factors of $s_{12}$ when using the IBP solutions. Any momentum crossing is, of course, applied only after the restoration of these factors of $s_{12}$. By applying the momentum crossings numerically rather than analytically, we avoid needing to parse and store (in RAM) numerous crossed versions of the analytical IBP solutions, for use during the numerical evaluation of the amplitude. When evaluating the unpolarised 3-photon amplitude, we observe that even without crossings, these solutions account for the majority of the RAM usage by our program.

### 4.3.5 Evaluation of the masters

Let us now turn to the numerical (i.e. finite-field) evaluation of the master integrals in the planar $C_1$ and $C_2$ topologies (as defined in Equation 3.9 of Chapter 3). We use the results from Ref. [33], which express a special basis set of 61 integrals in the $C_1$ topology in terms of irrational *pentagon functions* and boundary constants (denoted `bc4`). It is in terms of these irrational functions and constants that we eventually wish to express the entire amplitude. We start by using our IBP solutions to analytically map the special basis set of integrals from Ref. [33] onto our own set of 62 $C_1$ master integrals, which are listed in the supplementary material attached to Ref. [2]. Two of our 62 masters integrals are equal to each other, their integrands being interchanged under the symmetry $k_1 \leftrightarrow k_2$, and although these masters had to be kept distinct in order to use the strategy presented in Section 3.3 of Chapter 3, we now use the symmetry in order to set the two masters equal. This allows us to invert the relations and map our $C_1$ masters onto the basis from Ref. [33], and thence onto the pentagon functions and `bc4` constants. This mapping only needs to be produced once, and we have done so analytically using MATHEMATICA.

The mapping from our $C_1$ masters to pentagon functions needs to be expanded as a series in $\epsilon$. The original results in Ref. [33] are already expanded through 5 orders in $\epsilon$, and to these results we add book-keeping terms to account for the missing higher-order terms in $\epsilon$. These book-keeping terms therefore appear (in various combinations) in the mapping that we produce from our own masters to pentagon functions and `bc4` constants. When reconstructing an amplitude, we treat these book-keeping terms as extra irrational functions (on an equal footing with the pentagon functions and `bc4` constants) and reconstruct their rational coefficients at order $\epsilon^{-4}$ through $\epsilon^{0}$; we then verify that these reconstructed coefficients are identically zero, so as to confirm that the missing higher powers of $\epsilon$ in the solutions from Ref. [33] are not required in the amplitude.

It was mentioned above that momentum crossings are applied numerically to the IBPs at the time of numerical evaluation. For the master integrals, however, we apply the momentum crossings analytically in advance: rather than producing a single mapping from the $C_1$ master integrals to pentagon functions and `bc4` constants, we produce 120 copies, one for each of the possible 5! permutations of the 5 external momenta. In each case, the relevant momentum crossing is applied to the kinematic arguments of the irrationals as well as to their rational coefficients. We then use the pentagon-function identities described in Section 4.3.1 to re-express the pentagon functions in terms of our chosen minimal functional basis. Being of minimal size, our set of basis functions is not closed under the 120 crossing operations, which is why the crossings of the masters had to be performed analytically rather than numerically. The analytical, momentum-crossed, $\epsilon$-expanded solutions thus obtained for our $C_1$ master integrals are then used in the numerical evaluation of the amplitude.

The $C_2$ masters can be mapped onto the $C_1$ masters, in some cases after applying a momentum crossing to the $C_1$ masters. We use the mapping shown in Table 4.2, although other choices are also possible. These mappings are straight-forward and are implemented directly in the `C++` code performing the numerical evaluation of the masters. Since the mapping from $C_2$ to $C_1$ requires the application of a momentum crossing, and yet the $C_2$ masters may themselves be required with another momentum crossing, our `C++` framework includes a utility that combines two crossings into a single effective crossing. This is used by the numerical code to retrieve the pre-computed $C_1$

solutions with the correct momentum crossing, ensuring that we can automatically calculate an amplitude where arbitrary crossings may have been applied to the $C_2$ integrals. Having mapped the $C_2$ masters onto the $C_1$ masters, their reduction to pentagon functions and `bc4` constants can be calculated numerically using the existing $C_1$ solutions described above.

### 4.3.6 Simultaneous reconstruction of multiple amplitudes

When evaluating the unpolarised 3-photon amplitude, we observed that the evaluation of the IBPs occupied the majority of the running time for each numerical (i.e. finite-field) probe. Since many amplitudes share integrals in common, one can save significant resources by computing multiple amplitudes simultaneously while only evaluating the IBPs once at each numerical point.

A natural way to implement this would be to build a numerical cache that stores recent numerical IBP results. A cache ensures that at each numerical point, having spent a few minutes evaluating the first amplitude, one can evaluate the other amplitudes much more quickly, since the IBP results for any shared integrals can be immediately retrieved from the cache rather than needing to be evaluated again. One could furthermore envisage making the numerical cache large and saving its contents to disk, so that other amplitudes could in the future be reconstructed by using the same finite-field evaluation points and re-using the saved numerical IBP solutions. We implemented a cache but encountered two practical problems. Firstly, since our computational framework uses multiple CPUs in parallel (with each CPU evaluating the amplitude at a different numerical point), the cache has to be thread-safe to avoid corruptions when multiple CPUs try to write to the cache simultaneously. We resolved this issue using thread-locking to prohibit simultaneous writing from different CPUs. The second problem was the large size of the cache, even if only storing results for the 2 most recent kinematic points computed by each thread. Although a single finite-field number only requires a few bytes of space, the total size of the cache reached hundreds of GB due to the number of integrals (∼6,000), crossings (∼100), threads (∼30), master integrals (61), and powers of epsilon (5). This makes the numerical cache strategy unfeasible to use.

We therefore decided to re-organise the way the amplitudes are numerically evaluated. Instead of using a numerical cache to allow the sharing of IBP results between different amplitudes, we opted to re-organise our calculations so that each IBP result is immediately used and then discarded. Specifically, in the notation of Equation 3.3 from

**Table 4.2** Mapping of $C_2$ masters onto $C_1$ topology. The notation $C_1$x15432 indicates that the permutation $1 \to 5 \to 4 \to 3 \to 2 \to 1$ is applied to the external legs of the given $C_1$ integral.

| $C_2$ master | Mapping onto $C_1$ topology |
|---|---|
| $C_2[0,1,1,1,0,0,0,0,0,0,0]$ | $C_1[0,1,1,1,0,0,0,0,0,0,0]$ |
| $C_2[0,1,0,1,0,0,0,1,0,0,0]$ | $C_1[0,1,0,1,0,0,0,1,0,0,0]$ |
| $C_2[0,0,0,1,1,0,0,1,0,0,0]$ | $C_1[0,0,0,1,1,0,0,1,0,0,0]$ |
| $C_2[0,0,0,1,1,0,0,0,1,0,0]$ | $C_1[0,0,0,1,1,0,0,0,1,0,0]$ |
| $C_2[1,0,1,1,1,0,0,0,0,0,0]$ | $C_1[1,0,1,1,1,0,0,0,0,0,0]$ |
| $C_2[1,0,0,1,1,0,0,1,0,0,0]$ | $C_1[1,0,0,1,1,0,0,1,0,0,0]$ |
| $C_2[0,1,1,1,0,0,0,0,1,0,0]$ | $C_1[0,1,1,1,0,0,0,0,1,0,0]$ |
| $C_2[0,0,1,1,1,0,0,0,1,0,0]$ | $C_1[0,0,1,1,1,0,0,0,1,0,0]$ |
| $C_2[0,1,0,1,0,0,0,1,0,0,1]$ | $C_1\text{x15432}[0,0,1,1,1,0,0,0,1,0,0]$ |
| $C_2[0,1,0,1,0,0,0,0,1,0,1]$ | $C_1\text{x15432}[1,0,1,1,1,0,0,0,0,0,0]$ |
| $C_2[1,0,1,1,1,0,0,1,0,0,0]$ | $C_1[1,0,1,1,1,0,0,1,0,0,0]$ |
| $C_2[0,1,1,1,1,0,0,1,0,0,0]$ | $C_1[0,1,1,1,1,0,0,1,0,0,0]$ |
| $C_2[1,0,1,1,1,0,0,0,1,0,0]$ | $C_1[1,0,1,1,1,0,0,0,1,0,0]$ |
| $C_2[0,1,1,1,1,0,0,0,1,0,0]$ | $C_1[0,1,1,1,1,0,0,0,1,0,0]$ |
| $C_2[-1,1,1,1,1,0,0,0,1,0,0]$ | $C_1[-1,1,1,1,1,0,0,0,1,0,0]$ |
| $C_2[0,1,1,1,0,0,0,1,1,0,0]$ | $C_1[0,1,1,1,0,0,0,1,1,0,0]$ |
| $C_2[1,0,0,1,1,0,0,1,1,0,0]$ | $C_1[1,0,0,1,1,0,0,1,1,0,0]$ |
| $C_2[0,1,0,1,1,0,0,1,1,0,0]$ | $C_1[0,1,0,1,1,0,0,1,1,0,0]$ |
| $C_2[0,0,1,1,1,0,0,1,1,0,0]$ | $C_1[0,0,1,1,1,0,0,1,1,0,0]$ |
| $C_2[0,1,1,1,0,0,0,1,0,0,1]$ | $C_1\text{x15432}[0,0,1,1,1,0,0,1,1,0,0]$ |
| $C_2[0,1,1,1,0,0,0,0,1,0,1]$ | $C_1\text{x15432}[1,0,1,1,1,0,0,1,0,0,0]$ |
| $C_2[0,1,0,1,0,0,0,1,1,0,1]$ | $C_1\text{x15432}[1,0,1,1,1,0,0,0,1,0,0]$ |
| $C_2[1,0,1,1,1,0,0,1,1,0,0]$ | $C_1[1,0,1,1,1,0,0,1,1,0,0]$ |
| $C_2[1,-1,1,1,1,0,0,1,1,0,0]$ | $C_1[1,-1,1,1,1,0,0,1,1,0,0]$ |
| $C_2[0,1,1,1,1,0,0,1,1,0,0]$ | $C_1[0,1,1,1,1,0,0,1,1,0,0]$ |
| $C_2[-1,1,1,1,1,0,0,1,1,0,0]$ | $C_1[-1,1,1,1,1,0,0,1,1,0,0]$ |
| $C_2[0,1,1,1,0,0,0,1,1,0,1]$ | $C_1\text{x15432}[1,0,1,1,1,0,0,1,1,0,0]$ |
| $C_2[-1,1,1,1,0,0,0,1,1,0,1]$ | $\left(\begin{array}{c} C_1\text{x15432}[0,0,1,1,1,0,0,1,1,0,0]- \\ 2C_1\text{x15432}[1,-1,1,1,1,0,0,1,1,0,0] \end{array}\right)$ |

Chapter 3, we maintain running totals of the values of $\hat{c}_m$ for each amplitude. After the evaluation of an IBP reduction coefficient $c_{i,m}$, we multiply by the (amplitude- and integral-dependent) coefficients $f_i$ and immediately substitute into Equation 3.3 in order to update the running totals $\hat{c}_m$ for all the amplitudes. The coefficient $c_{i,m}$ can then be discarded and a new coefficient can be computed. After performing this procedure for all appearing integrals, we then evaluate the master integrals integrals themselves and thus obtain an evaluation for all the amplitudes in terms of pentagon functions and `bc4` constants. This implementation allowed us to simultaneously evaluate and then reconstruct a complete set of *polarised* 2-loop QCD amplitudes for $q\bar{q} \to \gamma\gamma\gamma$, which we intend to publish in a forthcoming paper. We have confirmed that the results agree with our computation of the unpolarised 3-photon amplitude, the latter being sufficient for the phenomenological study presented in the next section. We plan to re-use our framework to produce similar amplitudes involving the production of hard jets.

## 4.4 Phenomenology

### 4.4.1 LHC setup

Our calculational setup follows the 8 TeV ATLAS measurement [196]. The definition of histograms and experimental data is taken from the corresponding HEPData entry [223]. Our event selection is based on the following phase-space cuts:

- $E_T$ cut for the three photons: $E_{T,\gamma_1} > 27$ GeV, $E_{T,\gamma_2} > 22$ GeV and $E_{T,\gamma_3} > 15$ GeV, where $\gamma_1$ represents the hardest photon while $\gamma_3$ is the softest one.
- All photons are required to have $|\eta_\gamma| < 2.37$, excluding the range $1.37 < |\eta_\gamma| < 1.56$.
- Photon separation: the angular distance $\Delta R$ between any two photons is required to be $\Delta R_{ij} > 0.45$, where $\Delta R_{ij} = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2}$.
- A minimum three-photon invariant mass is required: $m_{\gamma\gamma\gamma} > 50$ GeV.
- Following Frixione [224], we impose smooth photon isolation. Specifically, for any angular distance $\Delta R$ away from each photon, subject to $\Delta R \leq \Delta R_0$, we require

$$E_T^{\mathrm{iso}}(\Delta R) < E_T^{\mathrm{max}} \frac{1 - \cos(\Delta R)}{1 - \cos(\Delta R_0)}\,, \tag{4.15}$$

where $\Delta R_0 = 0.4$ and $E_T^{\max} = 10$ GeV. The energy $E_T^{\text{iso}}(\Delta R)$ is defined as

$$E_T^{\text{iso}}(\Delta R) = \sum_i E_{T,i} \Theta(\Delta R - \Delta R_{i,\gamma}) \, . \tag{4.16}$$

The sum in the above equation is over all final-state partons $i$, and $E_{T,i}$ and $\Delta R_{i,\gamma}$ are parton $i$'s transverse energy and angular distance with respect to the photon.

Our calculation uses the `NNPDF31_nnlo_as_0118` PDF set [13]. We have not computed the PDF error; it was estimated in Ref. [196] and found to be below the (NLO) scale variation.

In this work we have utilised two different forms for the dynamic factorisation and renormalisation scales:

$$H_T \equiv \sum_{i=1}^{3} E_{T,\gamma_i} \, , \tag{4.17}$$

$$M_T \equiv \sqrt{p^2_{\gamma\gamma\gamma,T} + m^2_{\gamma\gamma\gamma}} \quad \text{with} \quad p_{\gamma\gamma\gamma} = \sum_{i=1}^{3} p_{\gamma_i} \quad \text{and} \quad m^2_{\gamma\gamma\gamma} = p^2_{\gamma\gamma\gamma} \, . \tag{4.18}$$

Our default central scale choice is $\mu_0 = H_T/4$, which follows from the findings of Ref. [55]. In fact, in the following we have studied the choices $\mu_0 = H_T/n$, with $n = 1, 2, 4$ as well as the alternative choices $\mu_0 = M_T/n$, with $n = 1, 2, 4$, that are based on the transverse mass of the three-photon system. The $M_T$-based scale was used in the latest diphoton production study [225]. We find that the differences between calculations with central scales $M_T/n$ and $H_T/n$, with $n = 1, 2, 4$, are relatively small. Scale variation of the factorisation and renormalisation scales is derived with a standard 7-point variation around the central scale $\mu_0$: we evaluate the cross-sections using all combinations of $\mu_F, \mu_R \in \{\frac{1}{2}\mu_0, \mu_0, 2\mu_0\}$ except for the 2 combinations $(\mu_F, \mu_R) = (\frac{1}{2}\mu_0, 2\mu_0)$ and $(\mu_F, \mu_R) = (2\mu_0, \frac{1}{2}\mu_0)$; the resulting range of the cross-sections is reported as the scale uncertainty.

### 4.4.2 Fiducial cross-section

In Fig. 4.1 we compare ATLAS data [196] with the predictions for the fiducial cross-section as defined in Section 4.4.1. We compare predictions based on 6 different renormalisation and factorisation scales, in LO, NLO and NNLO QCD. In all cases we observe large shifts from LO to NLO and from NLO to NNLO which are much larger

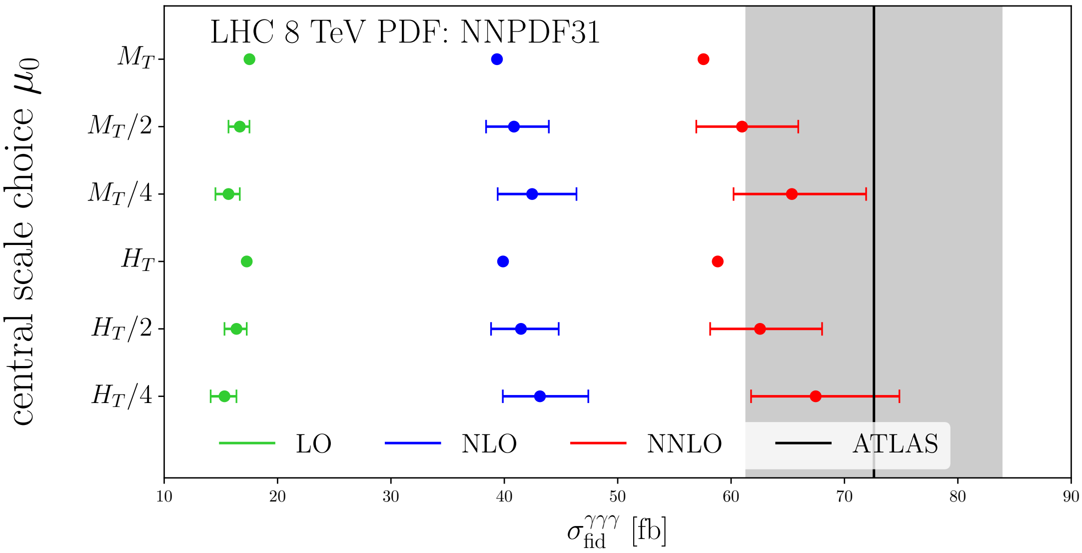


**Fig. 4.1** Predictions for the fiducial cross-section in LO (green), NLO (blue) and NNLO (red) QCD versus ATLAS data (black). Shown are predictions for six scale choices. The error bars on the theory predictions reflect scale variation only. For two of the scales only the central predictions are shown.

than the scale variations at, respectively, LO and NLO. Specifically, for our default scale $\mu_0 = H_T/4$ we have an NLO/LO correction of about 2.8 while the NNLO/NLO correction is about 1.6. We discuss this important feature in Section 4.4.4 below.

Predictions based on the two different scale functional forms shown in Equations 4.17 and 4.18 are rather similar relative to the sizes of scale variations and experimental uncertainties. Therefore, in the following, we will mainly focus our discussion on the $H_T$-based scales.

The scales $\mu_0 = H_T/4$ and $\mu_0 = H_T/2$ both agree with data, especially the $H_T/4$ one. The scale $\mu_0 = H_T$ is only just outside the measurement's uncertainty band. For simplicity in this work we did not compute the full scale variation around the scale $H_T$ (same for $M_T$) which is why only the central value is shown for these two scales. We do not expect the scale variation around the two scales will be much different that the pattern already emerging from Fig. 4.1.

In general we observe that the scale variation increases when going from LO to NNLO and that all scales are consistent at a given order within their scale uncertainties. For a proper interpretation of the reliability of the theoretical predictions it is therefore imperative to understand the issue of perturbative convergence. We devote Section 4.4.4 to this issue but here we only say in advance that we believe the NNLO predictions are probably the first order for which the theory prediction, with its associated scale variation, is reliable.

To summarize, based on the above discussion we conclude that our default scale choice is in perfect agreement with the experimental measurement

$$\begin{aligned} \sigma_{\rm fid}(\text{ATLAS}) &= 72.6 \pm 6.5(\text{stat.}) \pm 9.2(\text{syst.})\,\text{fb}\,, \\ \sigma_{\rm fid}(\text{NNLO QCD};\ H_T/4) &= 67.5^{+7.4\,(11\%)}_{-5.7\,(8\%)}\,(\text{scales})\ \text{fb}\,. \end{aligned} \tag{4.19}$$

Clearly, the inclusion of the NNLO QCD correction plays a crucial role in this agreement.

The Monte Carlo error on the fiducial NNLO prediction is below 1%. The fiducial predictions based on the various scale choices are provided in the supplementary material attached to our paper [3].

### 4.4.3 Differential distributions

A very large number of differential distributions have been measured by the ATLAS collaboration in Ref. [196]. In this work we have computed the theory predictions in NNLO QCD for all of them.

We start by showing the predictions for the $p_T$ distributions of the three individual photons: the hardest one $\gamma_1$ (Fig. 4.2), $\gamma_2$ (Fig. 4.3) and the softest one $\gamma_3$ (Fig. 4.4). We show the absolutely normalised distributions at LO (green), NLO (blue) and NNLO (red) in QCD. The top and middle panels show the central scale predictions and their corresponding 7-point scale variations for our default scale choice $\mu_0 = H_T/4$: the top panel shows the absolutely normalised distributions while the middle one shows the same results but normalised to the NLO central predictions. Shown in black is the ATLAS data. The bottom panels show the central scale predictions for the other 5 scale choices normalised to the central scale value for the default scale $H_T/4$. For a more quantitative comparison we also show the scale variation band of the default scale as well as the ATLAS data.

The plots for all other differential distributions shown next follow the same pattern: in Figures 4.5–4.7 we show the three $\Delta\Phi$ angles between the three pairs of photons; in Figures 4.8–4.10 we show the three rapidity differences $|\Delta\eta|$ between the three pairs of photons; in Figures 4.11–4.13 we show the invariant mass distributions between the three pairs of photons; and finally in Fig. 4.14 we show the invariant mass distribution of the three-photon system.

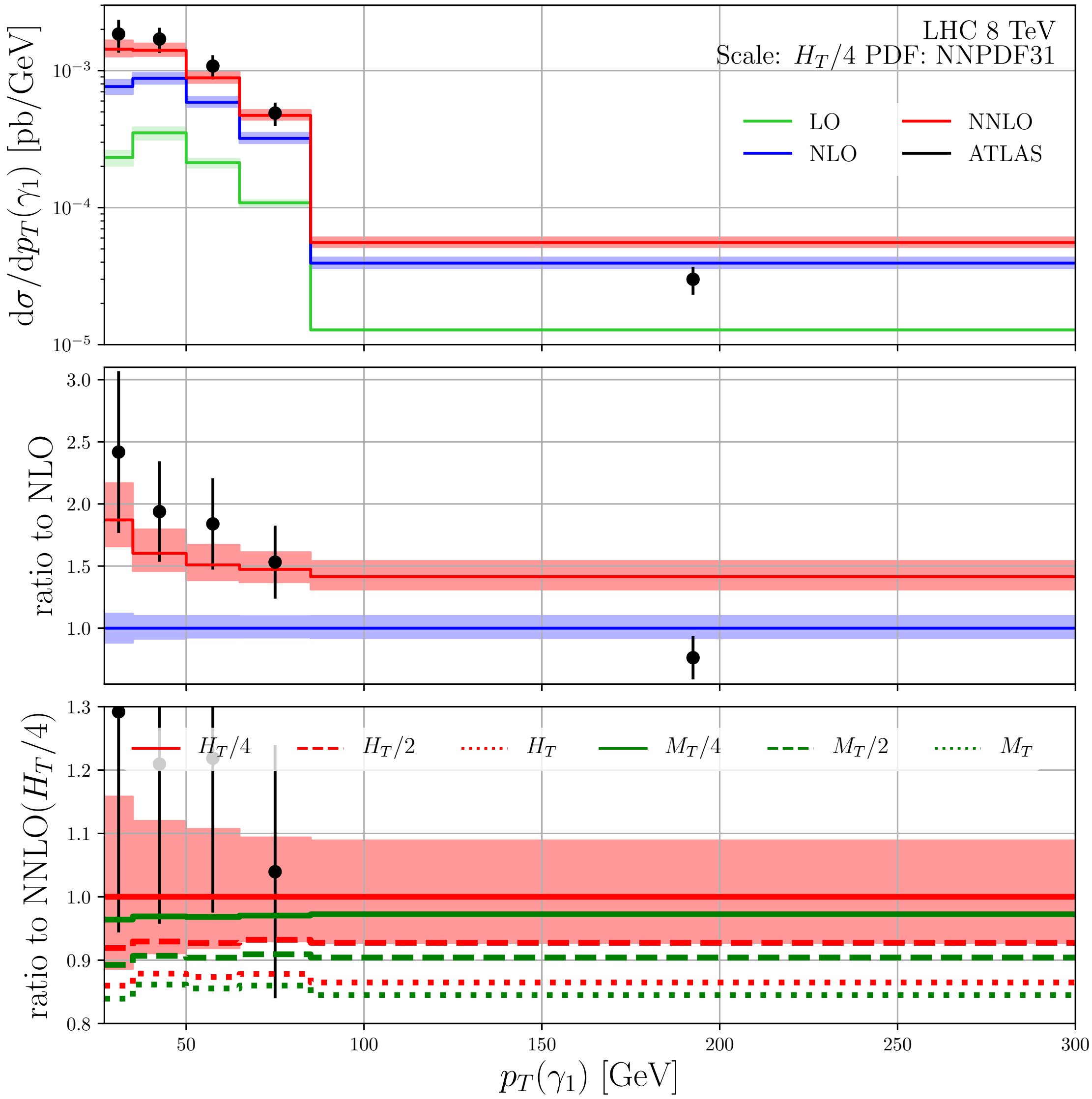


**Fig. 4.2** The $p_T$ distribution of the hardest photon, $\gamma_1$. Top panel shows absolute distributions. Middle panel shows distributions normalised to the NLO. Bottom panel shows NNLO predictions for 6 different scale choices (without further scale variation), normalised to the predictions of the default central choice $\mu_0 = H_T/4$. The bands represent 7-point scale variation about the central scale choice.

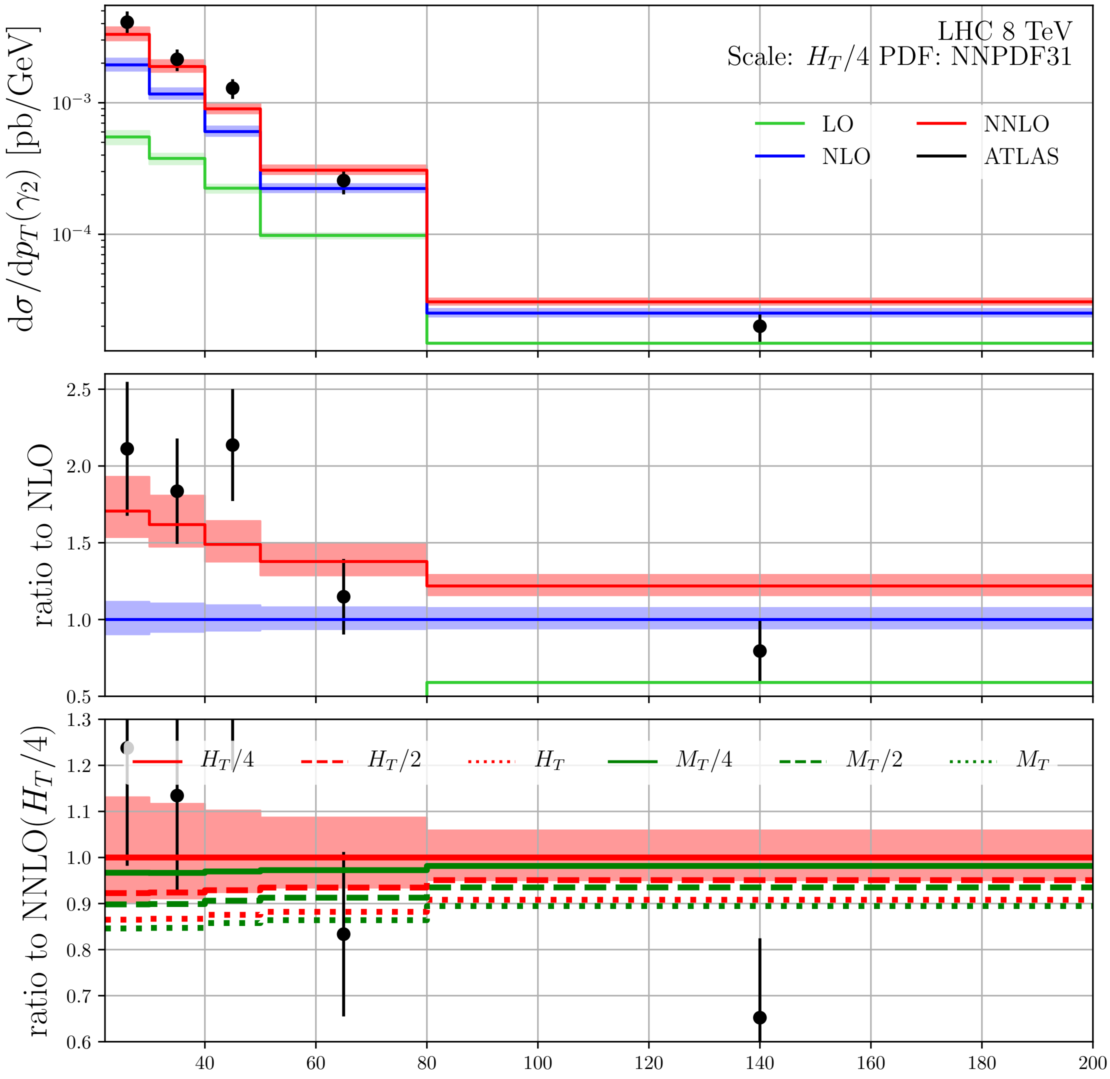


**Fig. 4.3** As in Fig. 4.2 but for the second-hardest photon, $\gamma_2$.

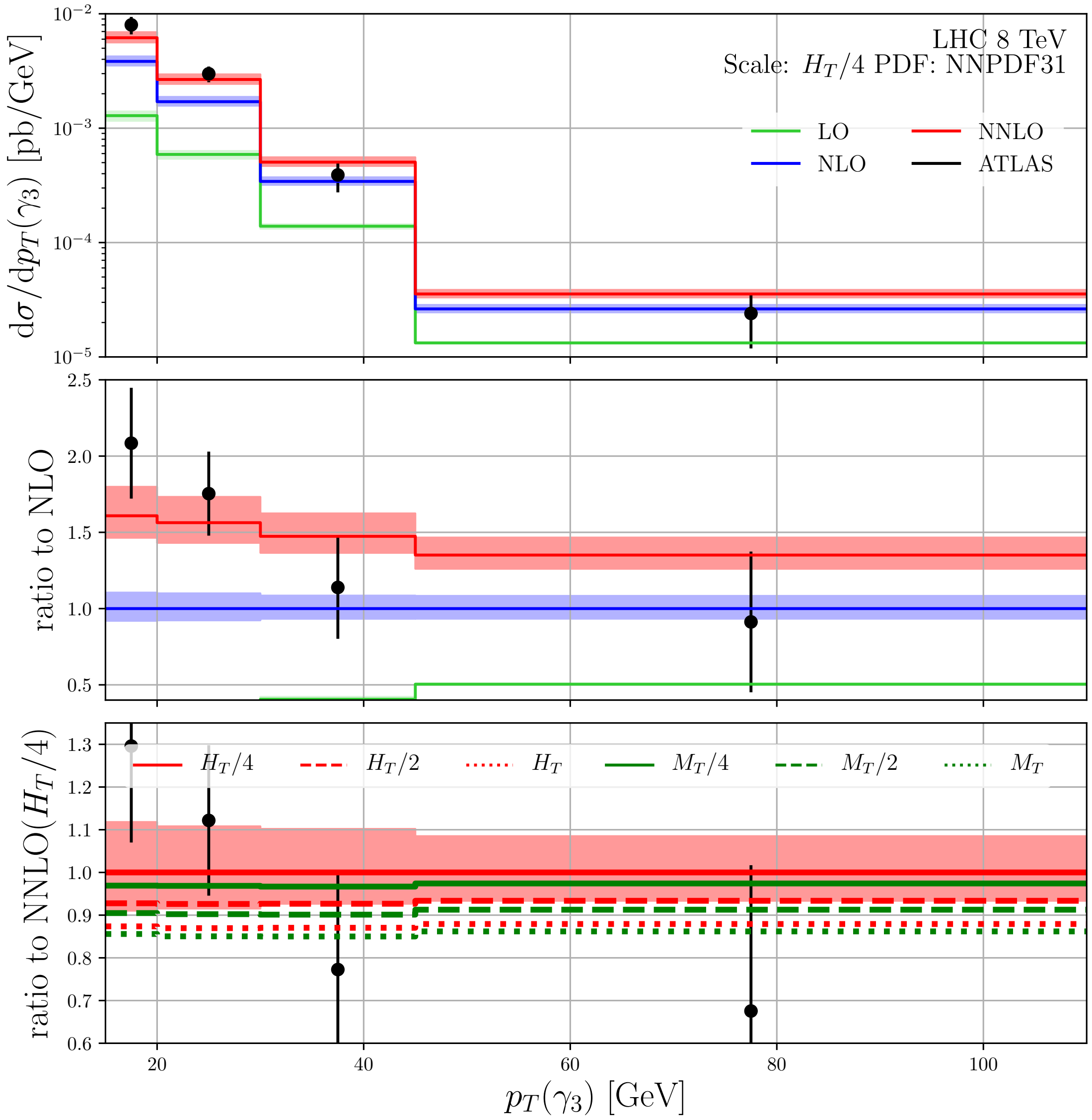


**Fig. 4.4** As in Fig. 4.2 but for the softest photon, $\gamma_3$.

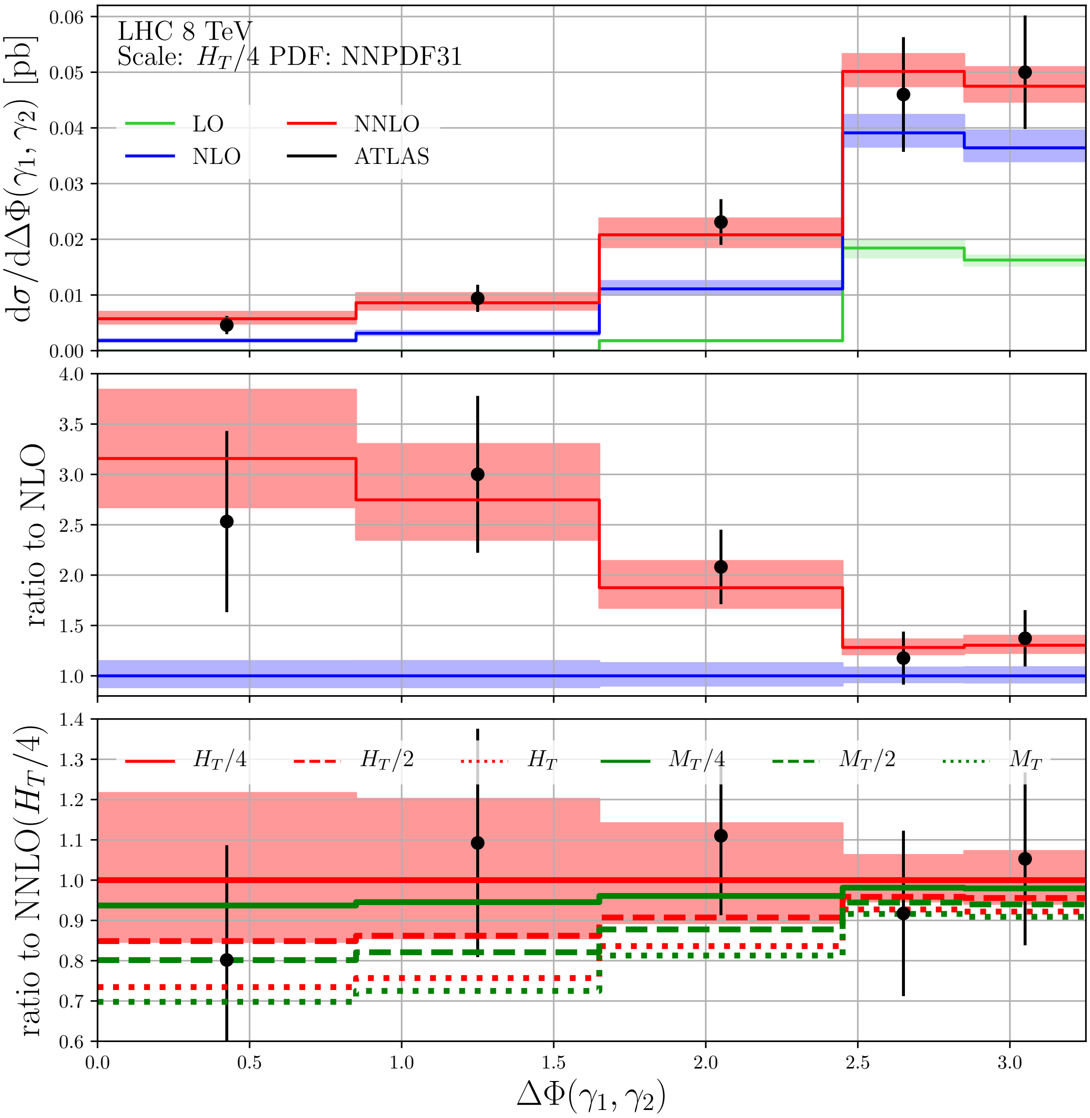


**Fig. 4.5** As in Fig. 4.2 but for the $\Delta\Phi(\gamma_1,\gamma_2)$ distribution.

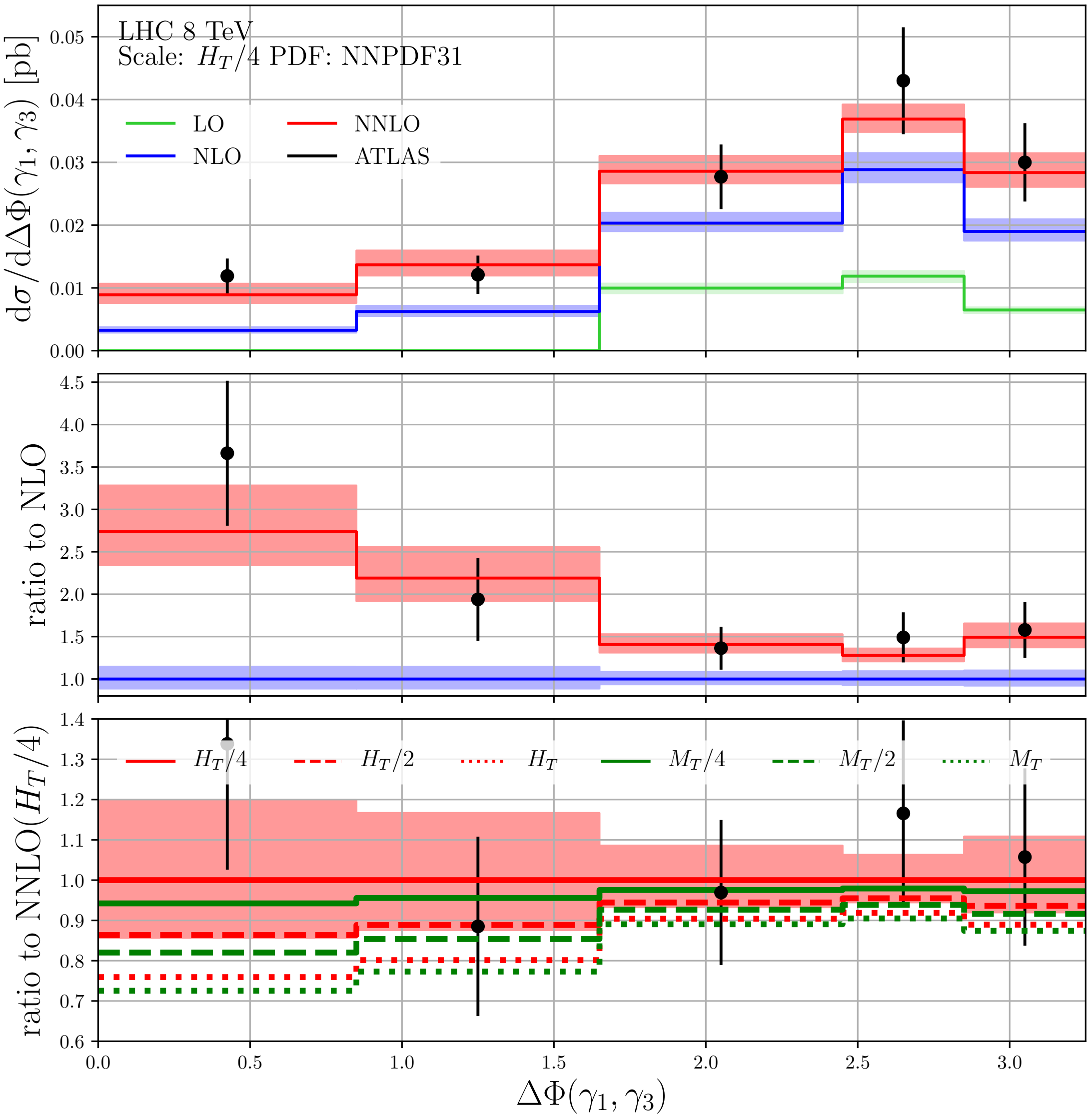


**Fig. 4.6** As in Fig. 4.2 but for the $\Delta\Phi(\gamma_1, \gamma_3)$ distribution.

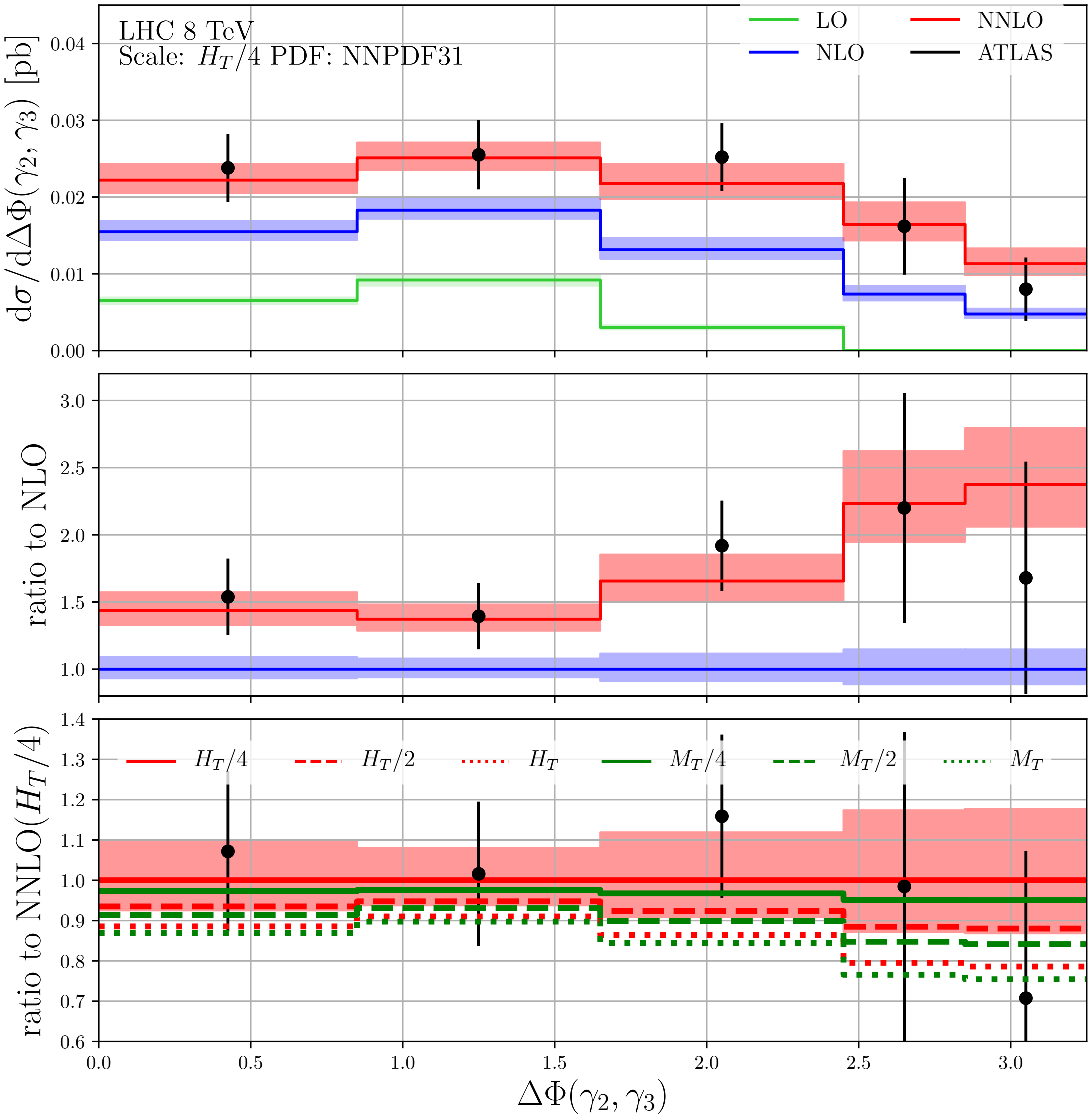


**Fig. 4.7** As in Fig. 4.2 but for the $\Delta\Phi(\gamma_2,\gamma_3)$ distribution.

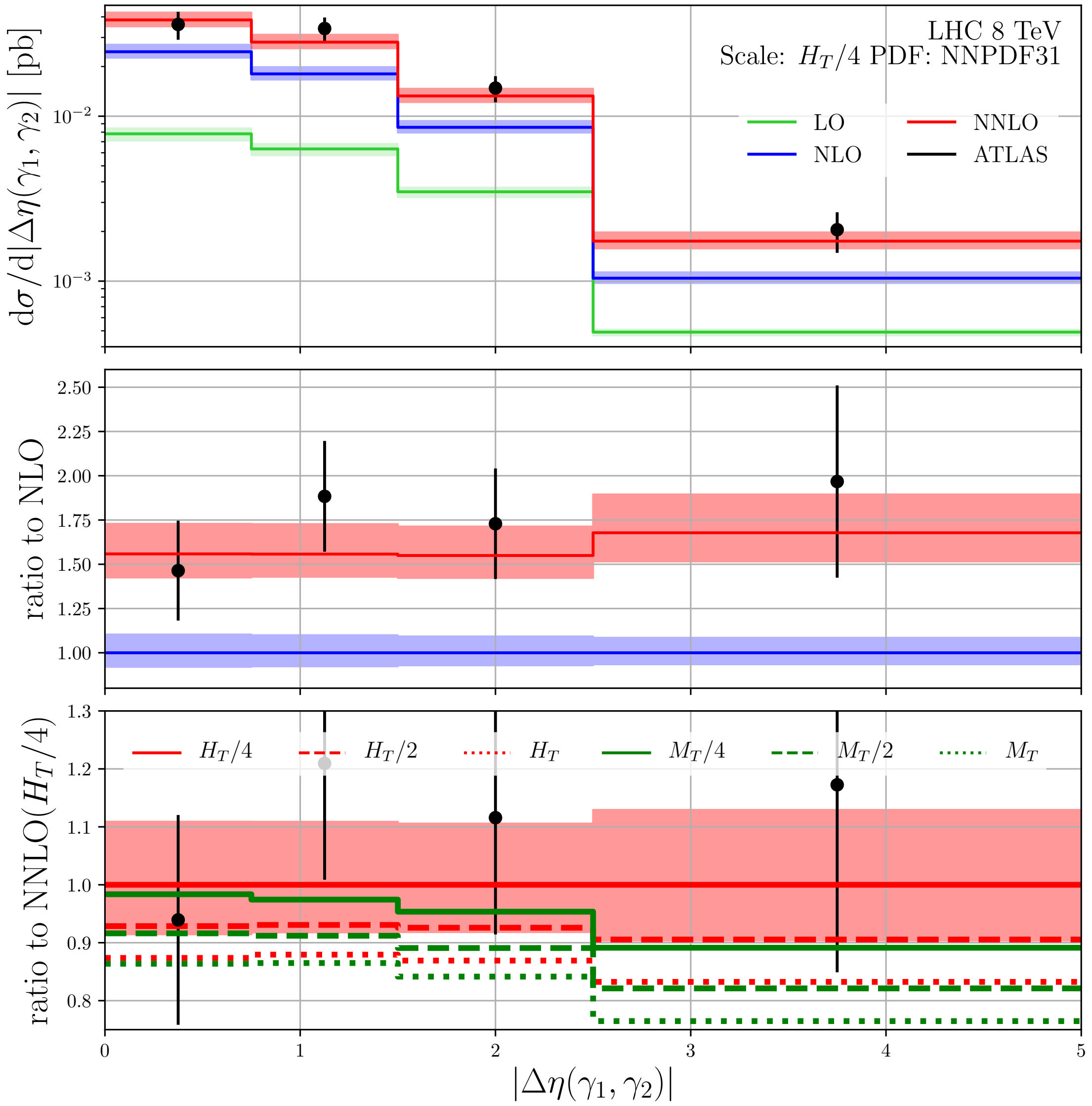


**Fig. 4.8** As in Fig. 4.2 but for the $|\Delta\eta(\gamma_1,\gamma_2)|$.

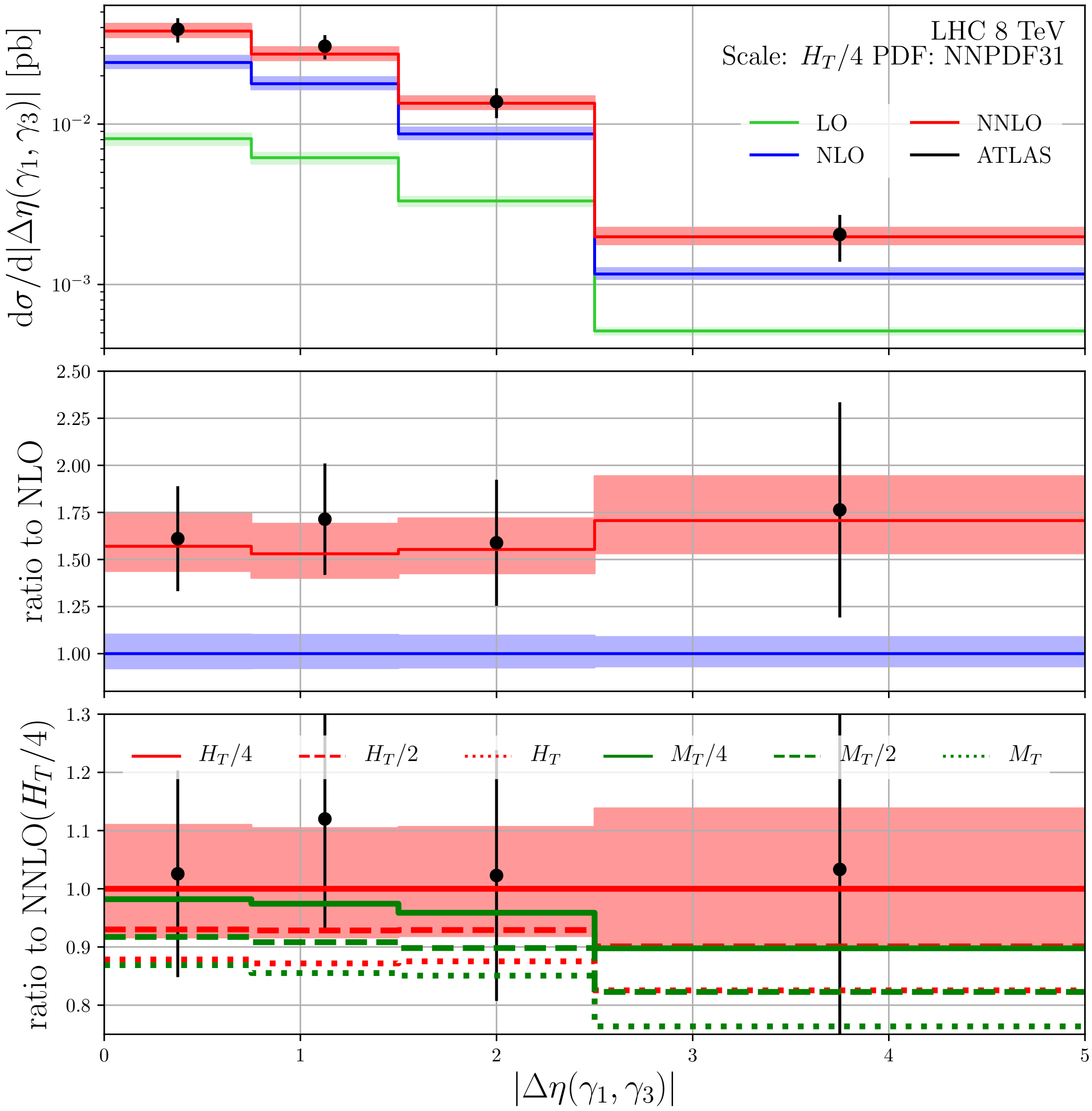


**Fig. 4.9** As in Fig. 4.2 but for the $|\Delta\eta(\gamma_1,\gamma_3)|$ distribution.

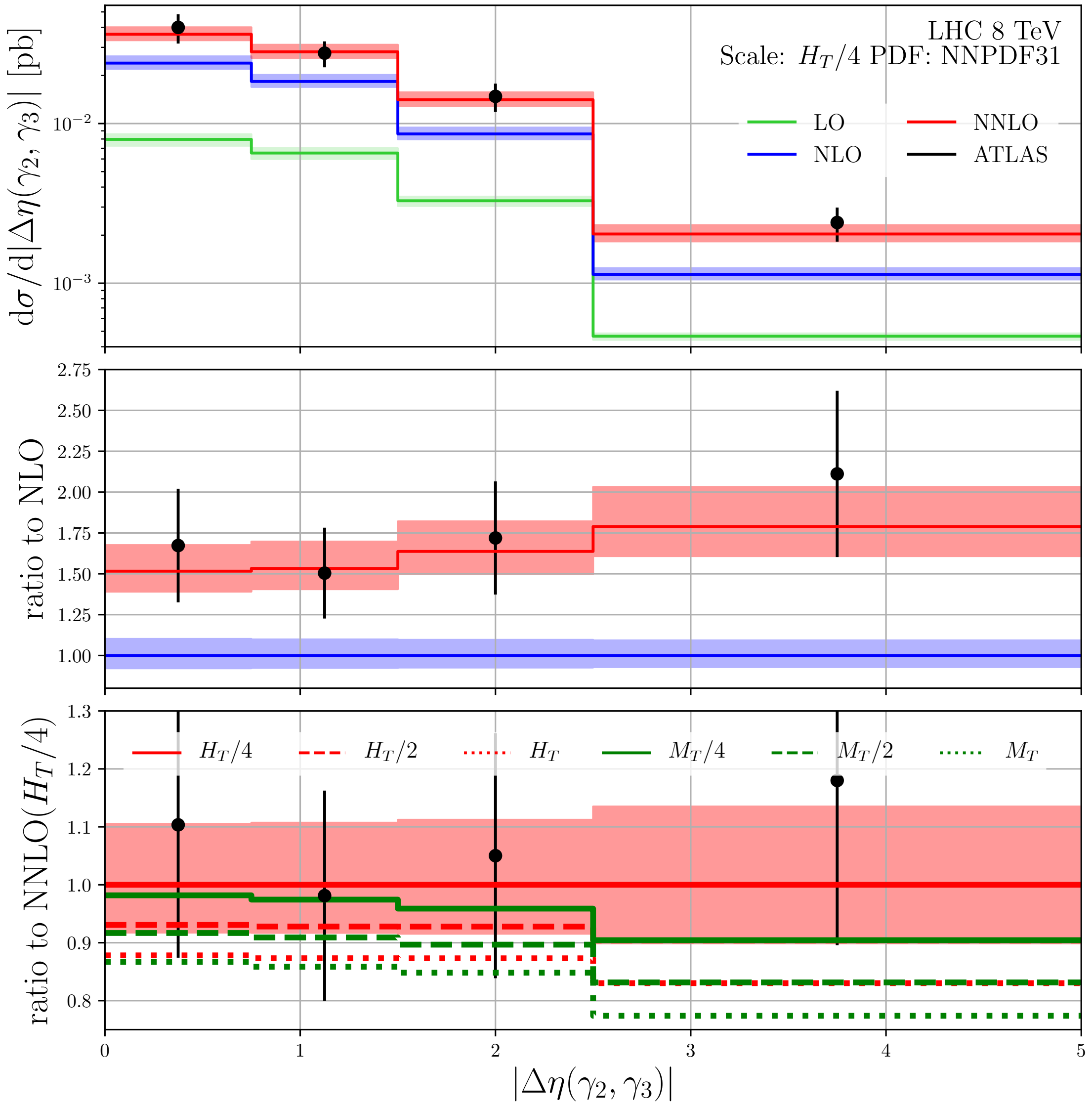


**Fig. 4.10** As in Fig. 4.2 but for the $|\Delta\eta(\gamma_2,\gamma_3)|$ distribution.

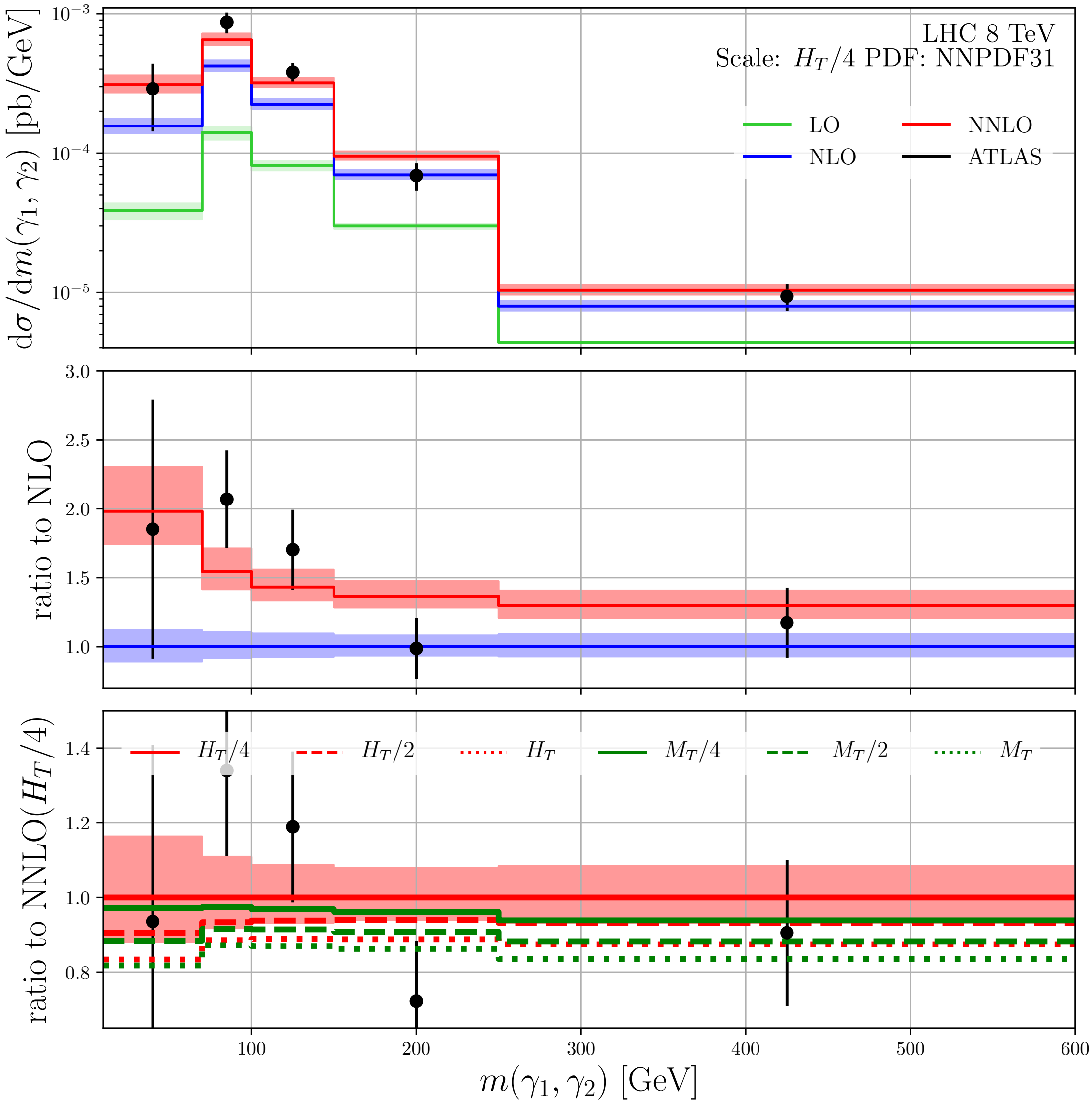


**Fig. 4.11** As in Fig. 4.2 but for the $m(\gamma_1, \gamma_2)$ distribution.

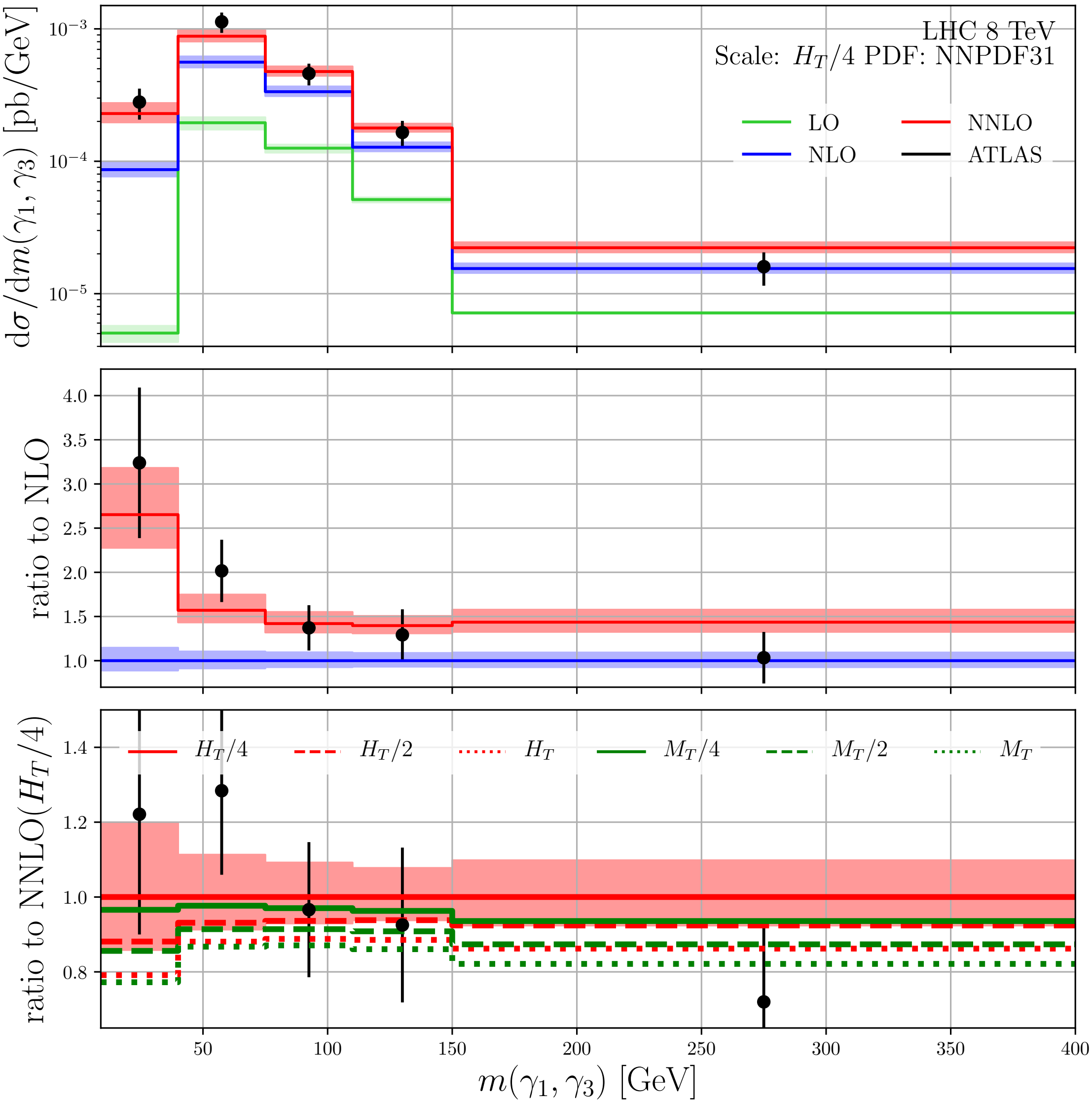


**Fig. 4.12** As in Fig. 4.2 but for the $m(\gamma_1, \gamma_3)$ distribution.

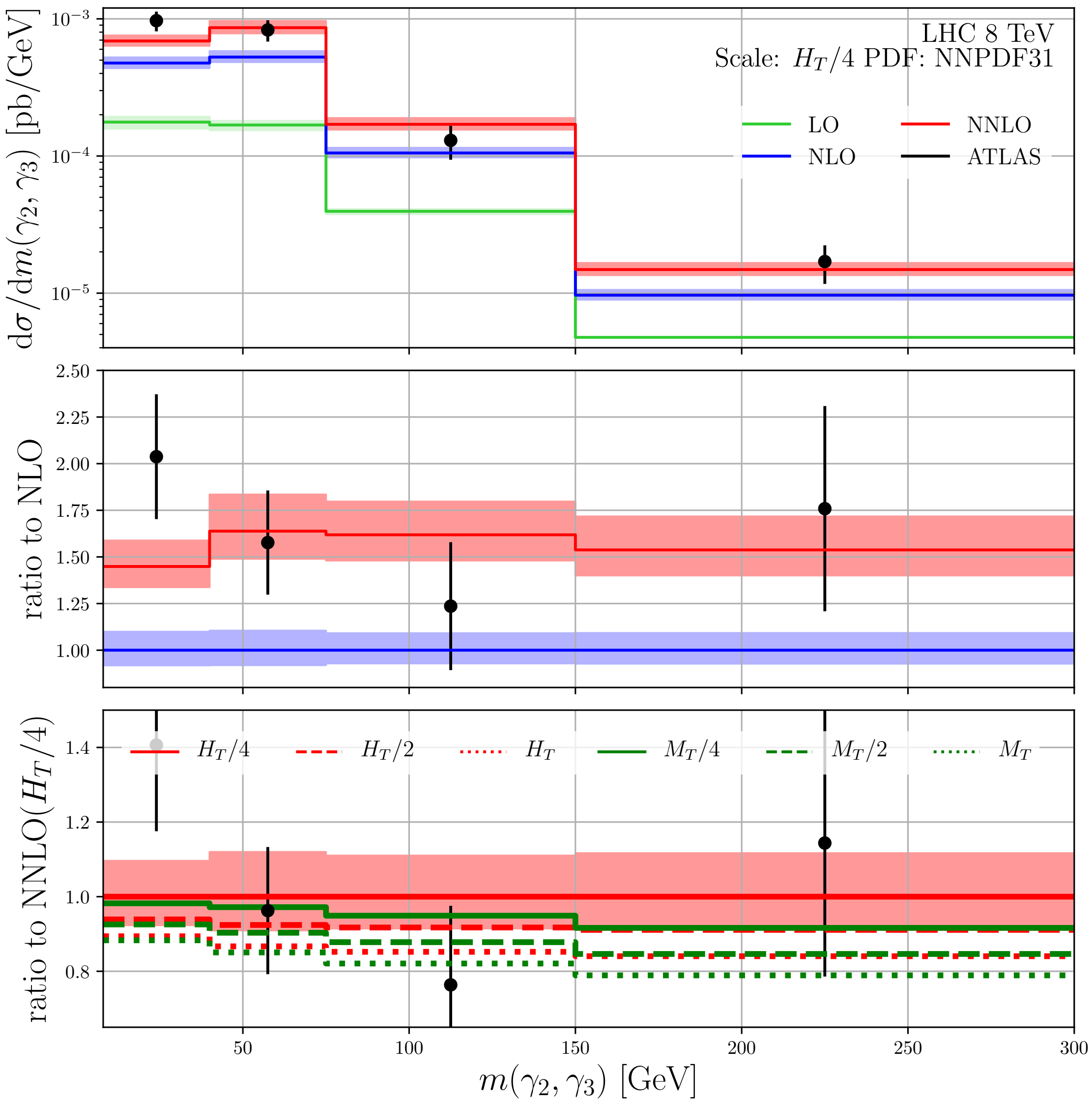


**Fig. 4.13** As in Fig. 4.2 but for the $m(\gamma_2, \gamma_3)$ distribution.

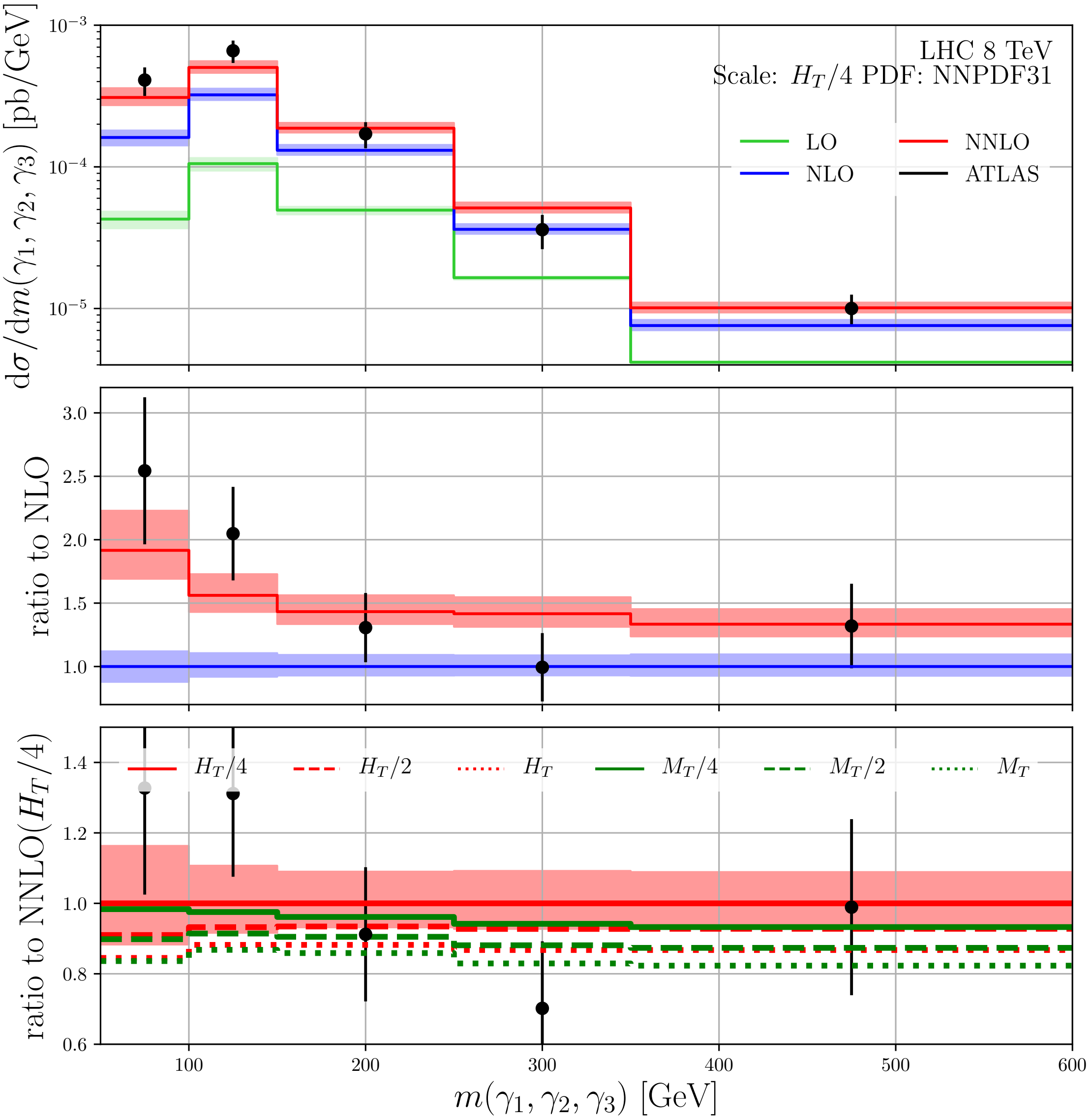


**Fig. 4.14** As in Fig. 4.2 but for the $m(\gamma_1, \gamma_2, \gamma_3)$ distribution.

Overall, a very consistent picture arises from all differential distributions, both in relation to the properties of the theory predictions as well as in relation to their agreement with data.

The most notable feature evident in all differential distributions are the large jumps from LO to NLO and from NLO to NNLO. The difference between orders is much larger than the corresponding scale variations at LO and NLO which, in principle, raises the question of the validity of perturbative convergence in this process. This behaviour closely resembles the behaviour already discussed for the fiducial cross-section. At this point we will only mention that we believe the NNLO QCD predictions is likely already a reliable prediction which can be confidently compared to data. We provide a detailed discussion of this point to Section 4.4.4.

The second notable feature is the good overall agreement between the NNLO QCD predictions, based on a scale $H_T/4$, and the data. While in most distributions there are bins that do not agree with the NNLO prediction, the overall shape and normalisation of all distributions are clearly correctly described at NNLO. In fact, the deviations observed in some of the bins could be due to larger statistical fluctuations in data.[9] An improved future measurement would clearly be very useful to clarify this. Interestingly, the best-described distributions are $\Delta\Phi$ and $\Delta\eta$ and we in fact observe perfect agreement there between the NNLO predictions and the data, for all pairs of photons.

The relative Monte Carlo error on the differential NNLO predictions shown here is below 3 percent. The theoretical predictions for all distributions, based on our default scale $H_T/4$, are available in the supplementary material attached to our paper [3].

We would like to stress that in this calculation we have only accounted for the QCD corrections through NNLO. Other theoretical contributions should at this point also be revisited. These include electroweak corrections and the effects arising from photon isolation. Effects due to PDFs appear to be subdominant to the scale variation at NNLO but this should also be cross-checked in a more complete study. The issue of the “best” scale choice remains an open question, just as we saw in Chapter 2, and at this level of precision seems to be a dominant source of theoretical uncertainty. For a detailed phenomenological study, the Monte Carlo error of the predictions shown here can be improved further. Finally, for completeness, one would like to have the complete

---

[9] At low $p_T$, the differential NNLO prediction for each photon is slightly lower than the measurements (see Figs. 4.2–4.4), albeit almost always consistent within $1\sigma$. Transverse-momentum resummation may be beneficial here, although the deviation is too small (compared to the theoretical and experimental uncertainties) for us to make a definitive statement.

NNLO prediction by including the subleading-colour scale-independent contributions to the two-loop finite remainder that were neglected in this work, although we expect them to be phenomenologically insignificant.

### 4.4.4 Discussion of perturbative convergence

As in diphoton production [202, 225, 226], the inclusive production of three photons exhibits behaviour that at first glance seems inconsistent with perturbative convergence. Indeed, as emphasised above, we observe large jumps from LO to NLO and from NLO to NNLO. These jumps are much larger than the corresponding scale variation bands at LO and NLO. This behaviour is evident in all differential distributions as well as in the fiducial cross-section. Specifically, we recall that for our default scale choice, the fiducial cross sections at NLO exceed those at LO by a factor of 2.8 while the NNLO/NLO K-factor is about 1.6. This behaviour is very similar to the behaviour encountered in diphoton production.

Various arguments have been given in the past for the appearance of such large K-factors in diphoton production. Two of those arguments are the presence of asymmetric cuts imposed on the two photons as well as the sizeable loop-induced $gg \to \gamma\gamma$ contribution. While these arguments have their merit, it is easy to see that they are not the drivers behind the behaviour we are trying to understand in diphoton (as well as three-photon) production. For example, the asymmetric cuts should not play an appreciable role for three-photon production because the Born state is naturally asymmetric. Similarly, while the loop-induced reaction is very large relative to the LO diphoton cross-section, its relative contribution at NNLO is not that sizeable — only on the order of 10% [225]. While such a contribution is important it is not large enough to be the driver behind the large K-factors observed in both processes. In fact, this issue can be cleanly understood in three-photon production process where the corresponding loop-induced amplitude $gg \to \gamma\gamma\gamma$ vanishes.

The above analysis of the $gg$-driven correction shows a very important point, namely the role the initial-state flux plays in the apparent perturbative convergence of these two processes. To quantify this in Fig. 4.15 we show the composition of the fiducial cross-section at LO, NLO and NNLO organised by initial-state partonic reactions. We show the results for three different $H_T$-based central scales; the results for the corresponding $M_T$-based scales are very similar.

What we observe in Fig. 4.15 is very illuminating. First, we note that the $gg$ flux does contribute (due to double real emissions and collinear subtractions) although

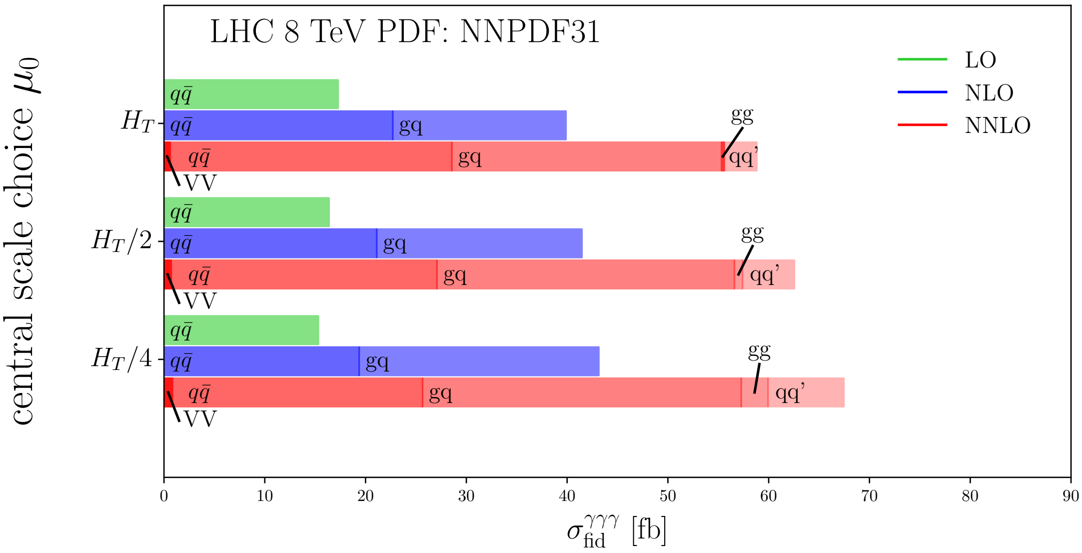


**Fig. 4.15** Anatomy of higher-order QCD corrections to the three-photon fiducial cross-section at LO (green), NLO (blue) and NNLO (red) by partonic channels for three different central scale choices. Also shown is the contribution from the scale-independent part of the two-loop finite remainder (VV) computed in our approximation defined in Section 4.2.2.

its effects are marginal, in the range of a few percent, depending on the choice of scale. Clearly, despite the fact the $gg$ flux is very large, its effect on the cross-section is nevertheless negligible because the corresponding partonic cross-sections are simply very small. The large gluon PDF does have a substantial impact on the three-photon cross-section but this happens through the $gq$ reaction. As also emphasised in Ref. [225] for the case of diphoton production, the $gq$ reaction starts to contribute only at NLO. This leads to a very unique interplay between purely partonic contributions, including their radiative corrections, and partonic fluxes. Specifically, the $q\bar{q}$ contribution receives a sizeable (although not huge) NLO radiative correction. At NLO this contribution is now dwarfed by the newly generated $gq$ correction which at this point is only LO. At NNLO the $q\bar{q}$ result gets another significant yet moderate correction, and the $gq$ reaction also receives sizeable but reasonable perturbative correction. At NNLO two new channels open: $gg$ and $qq'$, the latter being much more significant than the former. As also concluded in Ref. [225] for the case of diphoton production, NNLO is the first order where all large partonic reactions have already been included together with higher-order corrections to the largest ones. Once can therefore reasonably expect that from this point onwards, the yet-higher order N$^3$LO corrections to be derived in the future are likely to start showing a more convergent behaviour.

Before closing this section we would like to emphasise that the pattern of scale dependence observed when going from LO through NNLO should not be viewed as anomalous. The fact that scale dependence increases towards NNLO is simply due to the fact that the scale variation at the lower orders is artificially small and that, as explained in this section, at each new order through NNLO new large partonic reactions enter the process, thus increasing the overall scale dependence. The arguments given here imply that starting at $\mathrm{N^3LO}$, the scale variation should start to decrease. This will be very interesting to check in the future. In summary, in our view, the above arguments imply that the scale dependence of the NNLO prediction is likely not to be artificially small.

## 4.5 Conclusions

In this chapter, we have calculated the NNLO QCD corrections to three-photon production at the LHC. Our calculation is complete except for the scale-independent part of the two-loop finite remainder which is included in the leading-colour approximation. We estimate the effect of the missing two-loop contributions and we expect that they are phenomenologically insignificant.

Our calculation is the first NNLO calculation for a $2 \to 3$ scattering process. Although the production of colourless final states is not as complicated as a generic $2 \to 3$ reaction, we believe that our calculation clearly demonstrates that current computational technology is capable of dealing with the complicated structure of infrared singularities in multi-final-state processes. In particular, based on previous experience with dijet production (which was computed within the same Stripper framework as the present calculation) we think that the NNLO computation of three jets at the LHC is feasible.

An important part of this work is the creation of an automated computational framework to calculate 2-loop 5-point massless amplitudes. The framework uses finite-field methods to numerically evaluate the unreduced amplitude coefficients, IBP solutions, and master integral solutions, applying momentum crossings where necessary. These numerical results are combined and interpolated in order to simultaneously generate multiple amplitudes in exact, fully analytic form, defined directly in the physical region. We have used this framework to calculate the 2-loop amplitude $q\bar{q} \to \gamma\gamma\gamma$ in the leading-colour approximation, which is a key ingredient in this NNLO $2 \to 3$ calculation. To the best of our knowledge, this is the first time that a 2-loop

5-point amplitude has been expressed in a form readily available for phenomenological applications. Our computational framework is automated and we plan to use it in the near future to produce polarised and unpolarised amplitudes involving the production of combinations of photons and jets.

Beyond the production of the 2-loop amplitude in analytical form, we have extensively investigated the question of numerical stability and have been able to evaluate the amplitude numerically in about 30k phase-space points with sufficient numerical precision. This problem is highly non-trivial due to the large size of the amplitude and the large number of independent transcendental functions that appear in it.

The evaluation of the two-loop amplitude is expensive in terms of CPU time. We have investigated two possibilities to mitigate this problem: one involves specially generated phase-space points that accelerate the convergence of the phase-space integration, while the other involves the construction of a four-dimensional interpolating function which internally utilises machine-learning techniques. We find that these two approaches lead to compatible predictions within the corresponding Monte Carlo errors.

We observe that the structure of higher-order corrections in inclusive three-photon production is very interesting and closely resembles the structure known from diphoton production. We find very large higher-order corrections: the NLO prediction for the fiducial cross-section is larger than the LO prediction by a factor of 2.8 while the NNLO prediction exceeds the NLO prediction by a factor of 1.6. We have presented a detailed analysis of the anatomy of the higher-order corrections in this process and have concluded that the NNLO prediction is likely to be reliable.

Finally, we have compared our predictions with the high-quality LHC data available from the ATLAS Collaboration. We find that the sometimes huge discrepancies between QCD predictions and data noted previously at NLO are absent at NNLO and that the NNLO prediction agrees well with data for all distributions. This result clearly demonstrates how indispensable higher-order corrections are to quantitative phenomenological LHC analyses.

# Chapter 5

# Conclusion

In this thesis, higher-order QCD calculations were performed with the ultimate aim of increasing the precision of theoretical predictions for high-energy collider processes. After an initial study discussing the extent to which scale-setting methods can reduce theoretical uncertainties in higher-order QCD predictions, the majority of this thesis focussed on developing calculational techniques to enable the NNLO QCD frontier to be advanced to encompass $2 \to 3$ processes.

The dependence of perturbative QCD calculations on the arbitrary renormalisation scale $\mu_R$ is conventionally used as a means to estimate the theoretical uncertainties arising from the missing higher-order terms. The BLM/PMC scale-setting method is sometimes advocated as a way to eliminate the ambiguities over the value of $\mu_R$, by stipulating that all terms proportional to the QCD $\beta$-function coefficients are to be absorbed into the running coupling. Here, the BLM/PMC method for setting the renormalisation scale $\mu_R$ was studied and three ambiguities were identified in the BLM/PMC procedure itself. The numerical impact of these three ambiguities was examined using the example of the total cross-section at NNLO in QCD for $t\bar{t}$ production at the LHC and Tevatron. One of the ambiguities was found to have a negligible numerical impact on the cross section and we suspect that it is responsible for the very small "initial renormalization scale dependence" reported in Ref. [72]. The effect of the other two ambiguities was found to be much larger, and was in fact comparable to the impact of the choice of $\mu_R$ on the conventional cross-section. To try to resolve the most significant ambiguity, several approaches were studied but the cross-sections arising within these approaches were found to differ markedly from one another. Thus, while the BLM/PMC procedure is interesting and well-motivated, its effect on cross-sections might be better viewed as changing the central values of predictions, rather than removing theoretical uncertainties. It is important to emphasise that even an unambiguous scale-setting prescription would not remove the theoretical uncertainties in fixed-order QCD predictions, since these uncertainties arise from missing higher orders in $\alpha_S$. Calculating processes to higher orders in $\alpha_S$ is therefore indispensable in order to reduce these uncertainties.

A central ingredient in higher-order QCD calculations is the evaluation of multi-loop integrals and multi-loop amplitudes. Here, a novel strategy was introduced for solving IBP identities, which are widely used in computing multi-loop QCD integrals. The strategy was implemented in an efficient `C++` program, which incorporates a number of features that greatly increase the range of integrals that can be solved and reduce the time required to compute them. In particular, arranging symbolic algebraic operations into a network of lazily-evaluated expressions was found to improve the speed by an order of magnitude. By employing large-scale parallelisation of the evaluation of the expressions in the lazy-evaluation graph, the speed was increased by a further two orders of magnitude. A complete set of IBP solutions was hence produced for the calculation of any planar 2-loop 5-point massless QCD amplitude. These solutions enabled the calculation of the NNLO QCD corrections to 3-photon production described below. Furthermore, illustrative results for the associated non-planar integrals were also obtained, demonstrating that the program would be capable of producing a complete solution to the non-planar IBPs on a timescale of several months, subject to the availability of a computing cluster with several hundred CPUs.

Calculating multi-loop amplitudes is highly non-trivial, even after the required multi-loop integrals have been solved. Here, a computational framework was created for the automated calculation of 2-loop 5-point massless planar amplitudes. With the help of the IBP results from Chapter 3, the master integral solutions from Ref. [33], and finite-field interpolation [37], this framework produces amplitudes in fully-analytic form, defined directly in the physical kinematic region. The 2-loop leading-colour amplitude $q\bar{q} \to \gamma\gamma\gamma$ was thus calculated.

The culmination of the above work was the calculation of the NNLO QCD corrections to 3-photon production at the LHC. This is an important milestone, marking the first NNLO QCD calculation for a $2 \to 3$ process. Beyond this symbolic significance, the NNLO QCD 3-photon result is of phenomenological interest, particularly because the cross-section at NNLO exceeds the NLO predictions by about 60%. This increase brings the predictions (fiducial and differential alike) into line with the experimental measurements at the LHC, thus resolving a notable discrepancy between theory and experiment.

The last 20 years have seen remarkable progress in NNLO QCD calculations as increasingly-complicated $2 \to 1$ and $2 \to 2$ processes have been computed. It is anticipated that the coming years will see similar developments for $2 \to 3$ processes. Our automated framework for 2-loop 5-point amplitude calculations can be be used to compute multiple amplitudes simultaneously, which makes it well-suited for several avenues of future work. One avenue is the calculation of helicity amplitudes, since each 5-particle process requires the calculation of up to $2^5 = 32$ helicity amplitudes.[1] We are currently calculating helicity amplitudes for 3-photon production and our preliminary results suggest that the helicity amplitudes will be much more compact than the spin-averaged amplitude calculated in Chapter 4. Another avenue is the calculation of 2-loop amplitudes for processes involving jets: 3-jet production, $jj\gamma$-production, and $j\gamma\gamma$-production. Unlike 3-photon production, which (in the absence of real radiation) involves a single partonic process $q\bar{q} \to \gamma\gamma\gamma$, the production of jets involves a multitude of partonic channels. For example, 3-jet production requires calculating amplitudes for $gg \to ggg$, $q\bar{q} \to ggg$, $q\bar{q} \to q\bar{q}g$, and $q\bar{q} \to q'\bar{q}'g$, as well as all of their possible momentum crossings. As well as enabling the calculation of NNLO QCD corrections to more $2 \to 3$ processes, these 2-loop 5-point amplitudes will also be key ingredients for calculating N$^3$LO corrections for $2 \to 2$ processes. Looking further ahead, many $2 \to 3$ processes of interest involve massive particles such as $W$ and $H$ bosons, introducing fresh challenges for NNLO QCD calculations, including the need to calculate 2-loop 5-point integrals in the presence of additional kinematic (mass) scales. We are hopeful that the rapid ongoing advances in the calculation of real and virtual corrections will soon allow these processes to be computed. These high-precision theoretical predictions — together with increasingly high-precision measurements at the LHC — will enable the most stringent tests to date between theory and experiment for these complicated processes. This will either further confirm the validity of the Standard Model of Particle Physics or perhaps provide the first clues about what might lie beyond it.

---

[1] We say *up to 32* because these helicity combinations are not always independent: depending on the process, some helicity amplitudes might be related to each other by symmetries such as momentum crossings, charge conjugation, and parity.

# References

[1] Herschel A. Chawdhry and Alexander Mitov. Ambiguities of the principle of maximum conformality procedure for hadron collider processes. *Phys. Rev. D*, 100(7):074013, 2019. `http://doi.org/10.1103/PhysRevD.100.074013`.

[2] Herschel A. Chawdhry, Matthew A. Lim, and Alexander Mitov. Two-loop five-point massless QCD amplitudes within the integration-by-parts approach. *Phys. Rev. D*, 99(7):076011, 2019. `http://doi.org/10.1103/PhysRevD.99.076011`.

[3] Herschel A. Chawdhry, Michal Czakon, Alexander Mitov, and Rene Poncelet. NNLO QCD corrections to three-photon production at the LHC. *JHEP*, 02:057, 2020. `http://doi.org/10.1007/JHEP02(2020)057`.

[4] Krishna Kumar. A LaTeX PhD thesis template for the Cambridge University Engineering Department. `https://github.com/kks32/phd-thesis-template`.

[5] E. Rutherford. Collision of $\alpha$ particles with light atoms. IV. An anomalous effect in nitrogen. *Phil. Mag.*, 90(sup1):31–37, 2010.

[6] Ansgar Denner, Jean-Nicolas Lang, and Mathieu Pellen. Full NLO QCD corrections to off-shell ttbb production. arXiv:2008.00918.

[7] Matthew D. Schwartz. *Quantum Field Theory and the Standard Model*. Cambridge University Press, 3 2014.

[8] P.A. Baikov, K.G. Chetyrkin, and J.H. Kühn. Five-Loop Running of the QCD coupling constant. *Phys. Rev. Lett.*, 118(8):082002, 2017.

[9] Thomas Luthe, Andreas Maier, Peter Marquard, and York Schröder. Towards the five-loop Beta function for a general gauge group. *JHEP*, 07:127, 2016.

[10] F. Herzog, B. Ruijl, T. Ueda, J.A.M. Vermaseren, and A. Vogt. The five-loop beta function of Yang-Mills theory with fermions. *JHEP*, 02:090, 2017.

[11] S. Alekhin, J. Blümlein, S. Moch, and R. Placakyte. Parton distribution functions, $\alpha_s$, and heavy-quark masses for LHC Run II. *Phys. Rev. D*, 96(1):014011, 2017.

[12] Tie-Jiun Hou et al. New CTEQ global analysis of quantum chromodynamics with high-precision data from the LHC. 12 2019.

[13] Richard D. Ball et al. Parton distributions from high-precision collider data. *Eur. Phys. J. C*, 77(10):663, 2017.

[14] L.A. Harland-Lang, A.D. Martin, P. Motylinski, and R.S. Thorne. Parton distributions in the LHC era: MMHT 2014 PDFs. *Eur. Phys. J. C*, 75(5):204, 2015.

[15] V.N. Gribov and L.N. Lipatov. Deep inelastic e p scattering in perturbation theory. *Sov. J. Nucl. Phys.*, 15:438–450, 1972.

[16] Yuri L. Dokshitzer. Calculation of the Structure Functions for Deep Inelastic Scattering and e+ e- Annihilation by Perturbation Theory in Quantum Chromodynamics. *Sov. Phys. JETP*, 46:641–653, 1977.

[17] Guido Altarelli and G. Parisi. Asymptotic Freedom in Parton Language. *Nucl. Phys. B*, 126:298–318, 1977.

[18] T. Kinoshita. Mass singularities of Feynman amplitudes. *J. Math. Phys.*, 3:650–677, 1962.

[19] T.D. Lee and M. Nauenberg. Degenerate Systems and Mass Singularities. *Phys. Rev.*, 133:B1549–B1562, 1964.

[20] Gerard 't Hooft and M.J.G. Veltman. Regularization and Renormalization of Gauge Fields. *Nucl. Phys. B*, 44:189–213, 1972.

[21] Gerard 't Hooft. Dimensional regularization and the renormalization group. *Nucl. Phys. B*, 61:455–468, 1973.

[22] Steven Weinberg. New approach to the renormalization group. *Phys. Rev. D*, 8:3497–3509, 1973.

[23] John C. Collins. *Renormalization: An Introduction to Renormalization, The Renormalization Group, and the Operator Product Expansion*, volume 26 of *Cambridge Monographs on Mathematical Physics.* Cambridge University Press, Cambridge, 1986.

[24] Warren Siegel. Supersymmetric Dimensional Regularization via Dimensional Reduction. *Phys. Lett. B*, 84:193–196, 1979.

[25] Gerard 't Hooft. A Planar Diagram Theory for Strong Interactions. *Nucl. Phys. B*, 72:461, 1974.

[26] Vladimir A. Smirnov. Analytical result for dimensionally regularized massless on shell double box. *Phys. Lett. B*, 460:397–404, 1999.

[27] J.B. Tausk. Nonplanar massless two loop Feynman diagrams with four on-shell legs. *Phys. Lett. B*, 469:225–234, 1999.

[28] Johannes M. Henn, Kirill Melnikov, and Vladimir A. Smirnov. Two-loop planar master integrals for the production of off-shell vector bosons in hadron collisions. *JHEP*, 05:090, 2014.

[29] Fabrizio Caola, Johannes M. Henn, Kirill Melnikov, and Vladimir A. Smirnov. Non-planar master integrals for the production of two off-shell vector bosons in collisions of massless partons. *JHEP*, 09:043, 2014.

[30] Andreas von Manteuffel and Lorenzo Tancredi. A non-planar two-loop three-point function beyond multiple polylogarithms. *JHEP*, 06:127, 2017.

[31] Rutger H. Boels, Tobias Huber, and Gang Yang. Four-Loop Nonplanar Cusp Anomalous Dimension in N=4 Supersymmetric Yang-Mills Theory. *Phys. Rev. Lett.*, 119(20):201601, 2017.

[32] Alexander B. Goncharov. Multiple polylogarithms, cyclotomy and modular complexes. *Math. Res. Lett.*, 5:497–516, 1998.

[33] T. Gehrmann, J.M. Henn, and N.A. Lo Presti. Pentagon functions for massless planar scattering amplitudes. *JHEP*, 10:103, 2018.

[34] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, $3^{\text{rd}}$ edition, 2013.

[35] Andreas von Manteuffel and Robert M. Schabinger. A novel approach to integration by parts reduction. *Phys. Lett. B*, 744:101–104, 2015.

[36] Tiziano Peraro. Scattering amplitudes over finite fields and multivariate functional reconstruction. *JHEP*, 12:030, 2016.

[37] Jonas Klappert and Fabian Lange. Reconstructing rational functions with FireFly. *Comput. Phys. Commun.*, 247:106951, 2020.

[38] Tiziano Peraro. FiniteFlow: multivariate functional reconstruction using finite fields and dataflow graphs. *JHEP*, 07:031, 2019.

[39] Jonas Klappert, Sven Yannick Klein, and Fabian Lange. Interpolation of Dense and Sparse Rational Functions and other Improvements in `FireFly`. 4 2020.

[40] P. Nason, S. Dawson, and R.Keith Ellis. The One Particle Inclusive Differential Cross-Section for Heavy Quark Production in Hadronic Collisions. *Nucl. Phys. B*, 327:49–92, 1989. [Erratum: Nucl.Phys.B 335, 260–260 (1990)].

[41] W. Beenakker, W.L. van Neerven, R. Meng, G.A. Schuler, and J. Smith. QCD corrections to heavy quark production in hadron hadron collisions. *Nucl. Phys. B*, 351:507–560, 1991.

[42] Roger P. Roess, Elena S. Prassas, and William R. McShane. *Traffic engineering*. Pearson, London, 4th edition, 2011. Cantab UL 426.b.201.2.

[43] Michelangelo L. Mangano, Paolo Nason, and Giovanni Ridolfi. Heavy quark correlations in hadron collisions at next-to-leading order. *Nucl. Phys. B*, 373:295–345, 1992.

[44] S. Catani, F. Krauss, R. Kuhn, and B.R. Webber. QCD matrix elements + parton showers. *JHEP*, 11:063, 2001.

[45] F. Maltoni, Z. Sullivan, and S. Willenbrock. Higgs-Boson Production via Bottom-Quark Fusion. *Phys. Rev. D*, 67:093005, 2003.

[46] Eduard Boos and Tilman Plehn. Higgs boson production induced by bottom quarks. *Phys. Rev. D*, 69:094005, 2004.

[47] Johan Alwall et al. Comparative study of various algorithms for the merging of parton showers and matrix elements in hadronic collisions. *Eur. Phys. J. C*, 53:473–500, 2008.

[48] Christian W. Bauer and Bjorn O. Lange. Scale setting and resummation of logarithms in pp —> V + jets. 5 2009.

[49] Kirill Melnikov and Giulia Zanderighi. W+3 jet production at the LHC as a signal or background. *Phys. Rev. D*, 81:074025, 2010.

[50] Valentin Ahrens, Andrea Ferroglia, Matthias Neubert, Ben D. Pecjak, and Li Lin Yang. Renormalization-Group Improved Predictions for Top-Quark Pair Production at Hadron Colliders. *JHEP*, 09:097, 2010.

[51] C.F. Berger, Z. Bern, Lance J. Dixon, F. Febres Cordero, D. Forde, T. Gleisberg, H. Ita, D.A. Kosower, and D. Maitre. Precise Predictions for W + 4 Jet Production at the Large Hadron Collider. *Phys. Rev. Lett.*, 106:092001, 2011.

[52] Ansgar Denner, Stefan Dittmaier, Stefan Kallweit, and Stefano Pozzorini. NLO QCD corrections to off-shell top-antitop production with leptonic decays at hadron colliders. *JHEP*, 10:110, 2012.

[53] Charalampos Anastasiou, Claude Duhr, Falko Dulat, Elisabetta Furlan, Thomas Gehrmann, Franz Herzog, Achilleas Lazopoulos, and Bernhard Mistlberger. High precision determination of the gluon fusion Higgs boson cross-section at the LHC. *JHEP*, 05:058, 2016.

[54] Radja Boughezal, Xiaohui Liu, and Frank Petriello. A comparison of NNLO QCD predictions with 7 TeV ATLAS and CMS data for $V$+jet processes. *Phys. Lett. B*, 760:6–13, 2016.

[55] Michal Czakon, David Heymes, and Alexander Mitov. Dynamical scales for multi-TeV top-pair production at the LHC. *JHEP*, 04:071, 2017.

[56] James Currie, Aude Gehrmann-De Ridder, Thomas Gehrmann, E.W.N. Glover, Alexander Huss, and Joao Pires. Precise predictions for dijet production at the LHC. *Phys. Rev. Lett.*, 119(15):152001, 2017.

[57] Michal Czakon, Andrea Ferroglia, David Heymes, Alexander Mitov, Ben D. Pecjak, Darren J. Scott, Xing Wang, and Li Lin Yang. Resummation for (boosted) top-quark pair production at NNLO+NNLL' in QCD. *JHEP*, 05:149, 2018.

[58] Stanley J. Brodsky, G.Peter Lepage, and Paul B. Mackenzie. On the Elimination of Scale Ambiguities in Perturbative Quantum Chromodynamics. *Phys. Rev. D*, 28:228, 1983.

[59] Stanley J. Brodsky and Xing-Gang Wu. Scale Setting Using the Extended Renormalization Group and the Principle of Maximum Conformality: the QCD Coupling Constant at Four Loops. *Phys. Rev. D*, 85:034038, 2012. [Erratum: Phys.Rev.D 86, 079903 (2012)].

[60] Stanley J. Brodsky and Leonardo Di Giustino. Setting the Renormalization Scale in QCD: The Principle of Maximum Conformality. *Phys. Rev. D*, 86:085026, 2012.

[61] M. Spira, A. Djouadi, D. Graudenz, and P.M. Zerwas. Higgs boson production at the LHC. *Nucl. Phys. B*, 453:17–82, 1995.

[62] Andrei V. Belitsky and Dieter Mueller. Hard exclusive meson production at next-to-leading order. *Phys. Lett. B*, 513:349–360, 2001.

[63] D.Yu. Ivanov, L. Szymanowski, and G. Krasnikov. Vector meson electroproduction at next-to-leading order. *JETP Lett.*, 80:226–230, 2004. [Erratum: JETP Lett. 101, 844 (2015)].

[64] I.V. Anikin, B. Pire, L. Szymanowski, O.V. Teryaev, and S. Wallon. On BLM scale fixing in exclusive processes. *Eur. Phys. J. C*, 42:163–168, 2005.

[65] A.V. Belitsky and A.V. Radyushkin. Unraveling hadron structure with generalized parton distributions. *Phys. Rept.*, 418:1–387, 2005.

[66] Stanley J. Brodsky, Chueng-Ryong Ji, Alex Pang, and David G. Robertson. Optimal renormalization scale and scheme for exclusive processes. *Phys. Rev. D*, 57:245–252, 1998.

[67] A.P. Bakulev, A.V. Radyushkin, and N.G. Stefanis. Form-factors and QCD in space - like and time - like region. *Phys. Rev. D*, 62:113001, 2000.

[68] M.B. Voloshin. Precision determination of alpha-s and m(b) from QCD sum rules for b anti-b. *Int. J. Mod. Phys. A*, 10:2865–2880, 1995.

[69] Ikaros I.Y. Bigi, Mikhail A. Shifman, and N. Uraltsev. Aspects of heavy quark theory. *Ann. Rev. Nucl. Part. Sci.*, 47:591–661, 1997.

[70] Stanley J. Brodsky and Xing-Gang Wu. Eliminating the Renormalization Scale Ambiguity for Top-Pair Production Using the Principle of Maximum Conformality. *Phys. Rev. Lett.*, 109:042002, 2012.

[71] Stanley J. Brodsky and Xing-Gang Wu. Application of the Principle of Maximum Conformality to Top-Pair Production. *Phys. Rev. D*, 86:014021, 2012. [Erratum: Phys.Rev.D 87, 099902 (2013)].

[72] Stanley J. Brodsky and Xing-Gang Wu. Application of the Principle of Maximum Conformality to the Top-Quark Forward-Backward Asymmetry at the Tevatron. *Phys. Rev. D*, 85:114040, 2012.

[73] Sheng-Quan Wang, Xing-Gang Wu, Zong-Guo Si, and Stanley J. Brodsky. Application of the Principle of Maximum Conformality to the Top-Quark Charge Asymmetry at the LHC. *Phys. Rev. D*, 90(11):114034, 2014.

[74] Sheng-Quan Wang, Xing-Gang Wu, Zong-Guo Si, and Stanley J. Brodsky. Predictions for the Top-Quark Forward-Backward Asymmetry at High Invariant Pair Mass Using the Principle of Maximum Conformality. *Phys. Rev. D*, 93(1):014004, 2016.

[75] Sheng-Quan Wang, Xing-Gang Wu, Zong-Guo Si, and Stanley J. Brodsky. A precise determination of the top-quark pole mass. *Eur. Phys. J. C*, 78(3):237, 2018.

[76] A.L. Kataev and S.V. Mikhailov. Generalization of the Brodsky-Lepage-Mackenzie optimization within the $\{\beta\}$-expansion and the principle of maximal conformality. *Phys. Rev. D*, 91(1):014007, 2015.

[77] A.L. Kataev. The generalized BLM approach to fix scale-dependence in QCD: the current status of investigations. *J. Phys. Conf. Ser.*, 608(1):012078, 2015.

[78] A.L. Kataev and S.V. Mikhailov. The $\{\beta\}$-expansion formalism in perturbative QCD and its extension. *JHEP*, 11:079, 2016.

[79] S.S. Aleshin, A.L. Kataev, and K.V. Stepanyantz. The three-loop Adler $D$-function for $\mathcal{N} = 1$ SQCD regularized by dimensional reduction. *JHEP*, 03:196, 2019.

[80] A.V. Garkusha, A.L. Kataev, and V.S. Molokoedov. Renormalization scheme and gauge (in)dependence of the generalized Crewther relation: what are the real grounds of the $\beta$-factorization property? *JHEP*, 02:161, 2018.

[81] Stanley J. Brodsky, Matin Mojaza, and Xing-Gang Wu. Systematic Scale-Setting to All Orders: The Principle of Maximum Conformality and Commensurate Scale Relations. *Phys. Rev. D*, 89:014027, 2014.

[82] Peter Bärnreuther, Michal Czakon, and Alexander Mitov. Percent Level Precision Physics at the Tevatron: First Genuine NNLO QCD Corrections to $q\bar{q} \to t\bar{t} + X$. *Phys. Rev. Lett.*, 109:132001, 2012.

[83] Michal Czakon and Alexander Mitov. NNLO corrections to top-pair production at hadron colliders: the all-fermionic scattering channels. *JHEP*, 12:054, 2012.

[84] Michal Czakon and Alexander Mitov. NNLO corrections to top pair production at hadron colliders: the quark-gluon reaction. *JHEP*, 01:080, 2013.

[85] Michał Czakon, Paul Fiedler, and Alexander Mitov. Total Top-Quark Pair-Production Cross Section at Hadron Colliders Through $O(\alpha_S^4)$. *Phys. Rev. Lett.*, 110:252004, 2013.

[86] Michal Czakon, Paul Fiedler, and Alexander Mitov. Resolving the Tevatron Top Quark Forward-Backward Asymmetry Puzzle: Fully Differential Next-to-Next-to-Leading-Order Calculation. *Phys. Rev. Lett.*, 115(5):052001, 2015.

[87] Martin Beneke, Michal Czakon, Pietro Falgari, Alexander Mitov, and Christian Schwinn. Threshold expansion of the gg(qq-bar) —> QQ-bar + X cross section at O(alpha(s)**4). *Phys. Lett. B*, 690:483–490, 2010.

[88] Martin Beneke and Pedro Ruiz-Femenia. Threshold singularities, dispersion relations and fixed-order perturbative calculations. *JHEP*, 08:145, 2016.

[89] Sayipjamal Dulat, Tie-Jiun Hou, Jun Gao, Marco Guzzi, Joey Huston, Pavel Nadolsky, Jon Pumplin, Carl Schmidt, Daniel Stump, and C.P. Yuan. New parton distribution functions from a global analysis of quantum chromodynamics. *Phys. Rev. D*, 93(3):033006, 2016.

[90] Michal Czakon and Alexander Mitov. Top++: A Program for the Calculation of the Top-Pair Cross-Section at Hadron Colliders. *Comput. Phys. Commun.*, 185:2930, 2014.

[91] Morad Aaboud et al. Measurement of the $t\bar{t}$ production cross-section using $e\mu$ events with b-tagged jets in pp collisions at $\sqrt{s}$=13 TeV with the ATLAS detector. *Phys. Lett. B*, 761:136–157, 2016. [Erratum: Phys.Lett.B 772, 879–879 (2017)].

[92] Albert M Sirunyan et al. Measurement of the $\mathrm{t\bar{t}}$ production cross section, the top quark mass, and the strong coupling constant using dilepton events in pp collisions at $\sqrt{s} = 13$ TeV. *Eur. Phys. J. C*, 79(5):368, 2019.

[93] Timo Antero Aaltonen et al. Combination of Measurements of the Top-Quark Pair Production Cross Section from the Tevatron Collider. *Phys. Rev. D*, 89(7):072001, 2014.

[94] G. Grunberg and A.L. Kataev. On Some possible extensions of the Brodsky-Lepage-MacKenzie approach beyond the next-to-leading order. *Phys. Lett. B*, 279:352–358, 1992.

[95] Gorazd Cvetič and A.L. Kataev. Adler function and Bjorken polarized sum rule: Perturbation expansions in powers of the SU($\mathrm{N}_c$) conformal anomaly and studies of the conformal symmetry limit. *Phys. Rev. D*, 94(1):014006, 2016.

[96] F.V. Tkachov. A Theorem on Analytical Calculability of Four Loop Renormalization Group Functions. *Phys. Lett. B*, 100:65–68, 1981.

[97] K.G. Chetyrkin and F.V. Tkachov. Integration by Parts: The Algorithm to Calculate beta Functions in 4 Loops. *Nucl. Phys. B*, 192:159–204, 1981.

[98] A.G. Grozin. Integration by parts: An Introduction. *Int. J. Mod. Phys. A*, 26:2807–2854, 2011.

[99] A.V. Kotikov and S. Teber. Multi-loop techniques for massless Feynman diagram calculations. *Phys. Part. Nucl.*, 50(1):1–41, 2019.

[100] A.V. Kotikov. Differential equations method: New technique for massive Feynman diagrams calculation. *Phys. Lett. B*, 254:158–164, 1991.

[101] O.V. Tarasov. Connection between Feynman integrals having different values of the space-time dimension. *Phys. Rev. D*, 54:6479–6490, 1996.

[102] Ettore Remiddi. Differential equations for Feynman graph amplitudes. *Nuovo Cim. A*, 110:1435–1452, 1997.

[103] T. Gehrmann and E. Remiddi. Differential equations for two loop four point functions. *Nucl. Phys. B*, 580:485–518, 2000.

[104] P.A. Baikov and K.G. Chetyrkin. Four Loop Massless Propagators: An Algebraic Evaluation of All Master Integrals. *Nucl. Phys. B*, 837:186–220, 2010.

[105] Johannes M. Henn. Multiloop integrals in dimensional regularization made simple. *Phys. Rev. Lett.*, 110:251601, 2013.

[106] Mario Argeri and Pierpaolo Mastrolia. Feynman Diagrams and Differential Equations. *Int. J. Mod. Phys. A*, 22:4375–4436, 2007.

[107] Johannes M. Henn. Lectures on differential equations for Feynman integrals. *J. Phys. A*, 48:153001, 2015.

[108] Costas G. Papadopoulos, Damiano Tommasini, and Christopher Wever. The Pentabox Master Integrals with the Simplified Differential Equations approach. *JHEP*, 04:078, 2016.

[109] S. Laporta. High precision calculation of multiloop Feynman integrals by difference equations. *Int. J. Mod. Phys. A*, 15:5087–5159, 2000.

[110] Charalampos Anastasiou and Achilleas Lazopoulos. Automatic integral reduction for higher order perturbative calculations. *JHEP*, 07:046, 2004.

[111] A.V. Smirnov. Algorithm FIRE – Feynman Integral REduction. *JHEP*, 10:107, 2008.

[112] C. Studerus. Reduze-Feynman Integral Reduction in C++. *Comput. Phys. Commun.*, 181:1293–1300, 2010.

[113] A. von Manteuffel and C. Studerus. Reduze 2 - Distributed Feynman Integral Reduction. 1 2012.

[114] R.N. Lee. Presenting LiteRed: a tool for the Loop InTEgrals REDuction. 12 2012.

[115] Alexander V. Smirnov. FIRE5: a C++ implementation of Feynman Integral REduction. *Comput. Phys. Commun.*, 189:182–191, 2015.

[116] Philipp Maierhöfer, Johann Usovitsch, and Peter Uwer. Kira—A Feynman integral reduction program. *Comput. Phys. Commun.*, 230:99–112, 2018.

[117] R.N. Lee. Group structure of the integration-by-part identities and its application to the reduction of multiloop integrals. *JHEP*, 07:031, 2008.

[118] Janusz Gluza, Krzysztof Kajda, and David A. Kosower. Towards a Basis for Planar Two-Loop Integrals. *Phys. Rev. D*, 83:045012, 2011.

[119] Robert M. Schabinger. A New Algorithm For The Generation Of Unitarity-Compatible Integration By Parts Relations. *JHEP*, 01:077, 2012.

[120] Harald Ita. Two-loop Integrand Decomposition into Master Integrals and Surface Terms. *Phys. Rev. D*, 94(11):116015, 2016.

[121] Kasper J. Larsen and Yang Zhang. Integration-by-parts reductions from unitarity cuts and algebraic geometry. *Phys. Rev. D*, 93(4):041701, 2016.

[122] Rutger H. Boels, Qingjun Jin, and Hui Luo. Efficient integrand reduction for particles with spin. 2 2018.

[123] David C. Dunbar and Warren B. Perkins. Two-loop five-point all plus helicity Yang-Mills amplitude. *Phys. Rev. D*, 93(8):085029, 2016.

[124] David C. Dunbar, John H. Godwin, Guy R. Jehu, and Warren B. Perkins. Analytic all-plus-helicity gluon amplitudes in QCD. *Phys. Rev. D*, 96(11):116013, 2017.

[125] Simon Badger, Gustav Mogull, Alexander Ochirov, and Donal O'Connell. A Complete Two-Loop, Five-Gluon Helicity Amplitude in Yang-Mills Theory. *JHEP*, 10:064, 2015.

[126] Simon Badger, Hjalte Frellesvig, and Yang Zhang. A Two-Loop Five-Gluon Helicity Amplitude in QCD. *JHEP*, 12:045, 2013.

[127] T. Gehrmann, J.M. Henn, and N.A. Lo Presti. Analytic form of the two-loop planar five-gluon all-plus-helicity amplitude in QCD. *Phys. Rev. Lett.*, 116(6):062001, 2016. [Erratum: Phys.Rev.Lett. 116, 189903 (2016)].

[128] Simon Badger, Christian Brønnum-Hansen, Heribertus Bayu Hartanto, and Tiziano Peraro. First look at two-loop five-gluon scattering in QCD. *Phys. Rev. Lett.*, 120(9):092001, 2018.

[129] Samuel Abreu, Fernando Febres Cordero, Harald Ita, Ben Page, and Mao Zeng. Planar Two-Loop Five-Gluon Amplitudes from Numerical Unitarity. *Phys. Rev. D*, 97(11):116014, 2018.

[130] Janko Böhm, Alessandro Georgoudis, Kasper J. Larsen, Hans Schönemann, and Yang Zhang. Complete integration-by-parts reductions of the non-planar hexagon-box via module intersections. *JHEP*, 09:024, 2018.

[131] David A. Kosower. Direct Solution of Integration-by-Parts Systems. *Phys. Rev. D*, 98(2):025008, 2018.

[132] P.A. Baikov. A Practical criterion of irreducibility of multi-loop Feynman integrals. *Phys. Lett. B*, 634:325–329, 2006.

[133] Roman N. Lee and Andrei A. Pomeransky. Critical points and number of master integrals. *JHEP*, 11:165, 2013.

[134] Alessandro Georgoudis, Kasper J. Larsen, and Yang Zhang. Azurite: An algebraic geometry based package for finding bases of loop integrals. *Comput. Phys. Commun.*, 221:203–215, 2017.

[135] A.V. Smirnov and A.V. Petukhov. The Number of Master Integrals is Finite. *Lett. Math. Phys.*, 97:37–44, 2011.

[136] J.A.M. Vermaseren. New features of FORM. 10 2000.

[137] Costas G. Papadopoulos. Simplified differential equations approach for Master Integrals. *JHEP*, 07:088, 2014.

[138] Dmitry Chicherin, Johannes Henn, and Vladimir Mitev. Bootstrapping pentagon functions. *JHEP*, 05:164, 2018.

[139] Dmitry Chicherin, Johannes M. Henn, and Emery Sokatchev. Scattering Amplitudes from Superconformal Ward Identities. *Phys. Rev. Lett.*, 121(2):021602, 2018.

[140] R. H. Lewis. Fermat: A computer algebra system for polynomial and matrix computation. `http://home.bway.net/lewis/`.

[141] MPI Forum. MPI: A message-passing interface standard. Version 3.1. Technical report, University of Tennessee, 2015.

[142] Edgar Gabriel et al. Open MPI: Goals, concept, and design of a next generation MPI implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, pages 97–104, Budapest, Hungary, September 2004.

[143] W. Shane Grant and Randolph Voorhies. cereal - A C++11 library for serialization. 2017. `http://uscilab.github.io/cereal/`.

[144] R. Hamberg, W.L. van Neerven, and T. Matsuura. A complete calculation of the order $\alpha - s^2$ correction to the Drell-Yan $K$ factor. *Nucl. Phys. B*, 359:343–405, 1991. [Erratum: Nucl.Phys.B 644, 403–404 (2002)].

[145] T. Binoth and G. Heinrich. An automatized algorithm to compute infrared divergent multiloop integrals. *Nucl. Phys. B*, 585:741–759, 2000.

[146] Robert V. Harlander and William B. Kilgore. Next-to-next-to-leading order Higgs production at hadron colliders. *Phys. Rev. Lett.*, 88:201801, 2002.

[147] Charalampos Anastasiou and Kirill Melnikov. Higgs boson production at hadron colliders in NNLO QCD. *Nucl. Phys. B*, 646:220–256, 2002.

[148] Charalampos Anastasiou, Kirill Melnikov, and Frank Petriello. A new method for real radiation at NNLO. *Phys. Rev. D*, 69:076010, 2004.

[149] Charalampos Anastasiou, Lance J. Dixon, Kirill Melnikov, and Frank Petriello. High precision QCD at hadron colliders: Electroweak gauge boson rapidity distributions at NNLO. *Phys. Rev. D*, 69:094008, 2004.

[150] A. Daleo, T. Gehrmann, and D. Maitre. Antenna subtraction with hadronic initial states. *JHEP*, 04:016, 2007.

[151] Stefano Catani and Massimiliano Grazzini. An NNLO subtraction formalism in hadron collisions and its application to Higgs boson production at the LHC. *Phys. Rev. Lett.*, 98:222002, 2007.

[152] M. Czakon. A novel subtraction scheme for double-real radiation at NNLO. *Phys. Lett. B*, 693:259–268, 2010.

[153] M. Czakon. Double-real radiation in hadronic top quark pair production as a proof of a certain concept. *Nucl. Phys. B*, 849:250–295, 2011.

[154] M. Czakon and D. Heymes. Four-dimensional formulation of the sector-improved residue subtraction scheme. *Nucl. Phys. B*, 890:152–227, 2014.

[155] Radja Boughezal, Fabrizio Caola, Kirill Melnikov, Frank Petriello, and Markus Schulze. Higgs boson production in association with a jet at next-to-next-to-leading order in perturbative QCD. *JHEP*, 06:072, 2013.

[156] James Currie, Aude Gehrmann-De Ridder, E.W.N. Glover, and Joao Pires. NNLO QCD corrections to jet production at hadron colliders from gluon scattering. *JHEP*, 01:110, 2014.

[157] Radja Boughezal, Christfried Focke, Xiaohui Liu, and Frank Petriello. $W$-boson production in association with a jet at next-to-next-to-leading order in perturbative QCD. *Phys. Rev. Lett.*, 115(6):062002, 2015.

[158] Matteo Cacciari, Frédéric A. Dreyer, Alexander Karlberg, Gavin P. Salam, and Giulia Zanderighi. Fully Differential Vector-Boson-Fusion Higgs Production at Next-to-Next-to-Leading Order. *Phys. Rev. Lett.*, 115(8):082002, 2015. [Erratum: Phys.Rev.Lett. 120, 139901 (2018)].

[159] Radja Boughezal, John M. Campbell, R. Keith Ellis, Christfried Focke, Walter Giele, Xiaohui Liu, Frank Petriello, and Ciaran Williams. Color singlet production at NNLO in MCFM. *Eur. Phys. J. C*, 77(1):7, 2017.

[160] Fabrizio Caola, Kirill Melnikov, and Raoul Röntsch. Nested soft-collinear subtractions in NNLO QCD computations. *Eur. Phys. J. C*, 77(4):248, 2017.

[161] Massimiliano Grazzini, Stefan Kallweit, and Marius Wiesemann. Fully differential NNLO computations with MATRIX. *Eur. Phys. J. C*, 78(7):537, 2018.

[162] Fabrizio Caola, Kirill Melnikov, and Raoul Röntsch. Analytic results for color-singlet production at NNLO QCD with the nested soft-collinear subtraction scheme. *Eur. Phys. J. C*, 79(5):386, 2019.

[163] Gabor Somogyi, Zoltan Trocsanyi, and Vittorio Del Duca. Matching of singly- and doubly-unresolved limits of tree-level QCD squared matrix elements. *JHEP*, 06:024, 2005.

[164] A. Gehrmann-De Ridder, T. Gehrmann, and E.W.Nigel Glover. Antenna subtraction at NNLO. *JHEP*, 09:056, 2005.

[165] G. Abelof and A. Gehrmann-De Ridder. Antenna subtraction for the production of heavy particles at hadron colliders. *JHEP*, 04:063, 2011.

[166] James Currie, E.W.N. Glover, and Steven Wells. Infrared Structure at NNLO Using Antenna Subtraction. *JHEP*, 04:066, 2013.

[167] Jonathan Gaunt, Maximilian Stahlhofen, Frank J. Tackmann, and Jonathan R. Walsh. N-jettiness Subtractions for NNLO QCD Calculations. *JHEP*, 09:058, 2015.

[168] Roberto Bonciani, Stefano Catani, Massimiliano Grazzini, Hayk Sargsyan, and Alessandro Torre. The $q_T$ subtraction method for top quark production at hadron colliders. *Eur. Phys. J. C*, 75(12):581, 2015.

[169] Franz Herzog. Geometric IR subtraction for final state real radiation. *JHEP*, 08:006, 2018.

[170] L. Magnea, E. Maina, G. Pelliccioli, C. Signorile-Signorile, P. Torrielli, and S. Uccirati. Local analytic sector subtraction at NNLO. *JHEP*, 12:107, 2018. [Erratum: JHEP 06, 013 (2019)].

[171] Lorenzo Magnea, Ezio Maina, Giovanni Pelliccioli, Chiara Signorile-Signorile, Paolo Torrielli, and Sandro Uccirati. Factorisation and Subtraction beyond NLO. *JHEP*, 12:062, 2018.

[172] Stefano Catani, Simone Devoto, Massimiliano Grazzini, Stefan Kallweit, and Javier Mazzitelli. Top-quark pair production at the LHC: Fully differential QCD predictions at NNLO. *JHEP*, 07:100, 2019.

[173] Fabrizio Caola, Kirill Melnikov, and Raoul Röntsch. Analytic results for decays of color singlets to $gg$ and $q\bar{q}$ final states at NNLO QCD with the nested soft-collinear subtraction scheme. *Eur. Phys. J. C*, 79(12):1013, 2019.

[174] Mao Zeng. Differential equations on unitarity cut surfaces. *JHEP*, 06:121, 2017.

[175] Samuel Abreu, Ben Page, and Mao Zeng. Differential equations from unitarity cuts: nonplanar hexa-box integrals. *JHEP*, 01:006, 2019.

[176] D. Chicherin, T. Gehrmann, J.M. Henn, N.A. Lo Presti, V. Mitev, and P. Wasser. Analytic result for the nonplanar hexa-box integrals. *JHEP*, 03:042, 2019.

[177] Adam Kardos. A new reduction strategy for special negative sectors of planar two-loop integrals without Laporta algorithm. 12 2018.

[178] 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(4):041603, 2019.

[179] Dominik Bendle, Janko Böhm, Wolfram Decker, Alessandro Georgoudis, Franz-Josef Pfreundt, Mirko Rahn, Pascal Wasser, and Yang Zhang. Integration-by-parts reductions of Feynman integrals using Singular and GPI-Space. *JHEP*, 02:079, 2020.

[180] Costas G. Papadopoulos and Christopher Wever. Internal Reduction method for computing Feynman Integrals. *JHEP*, 02:112, 2020.

[181] Simon Badger, Christian Brønnum-Hansen, Thomas Gehrmann, Heribertus Bayu Hartanto, Johannes Henn, Nicola Adriano Lo Presti, and Tiziano Peraro. Applications of integrand reduction to two-loop five-point scattering amplitudes in QCD. *PoS*, LL2018:006, 2018.

[182] 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.

[183] Simon Badger, Christian Brønnum-Hansen, Heribertus Bayu Hartanto, and Tiziano Peraro. Analytic helicity amplitudes for two-loop five-gluon scattering: the single-minus case. *JHEP*, 01:186, 2019.

[184] 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(8):082002, 2019.

[185] 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.

[186] S. Badger, D. Chicherin, T. Gehrmann, G. Heinrich, J.M. Henn, T. Peraro, P. Wasser, Y. Zhang, and S. Zoia. Analytic form of the full two-loop five-gluon all-plus helicity amplitude. *Phys. Rev. Lett.*, 123(7):071601, 2019.

[187] Heribertus Bayu Hartanto, Simon Badger, Christian Brønnum-Hansen, and Tiziano Peraro. A numerical evaluation of planar two-loop helicity amplitudes for a W-boson plus four partons. *JHEP*, 09:119, 2019.

[188] Simon Badger, Gustav Mogull, and Tiziano Peraro. Local integrands for two-loop all-plus Yang-Mills amplitudes. *JHEP*, 08:063, 2016.

[189] Z. Bern, M. Czakon, D.A. Kosower, R. Roiban, and V.A. Smirnov. Two-loop iteration of five-point N=4 super-Yang-Mills amplitudes. *Phys. Rev. Lett.*, 97:181601, 2006.

[190] Pierpaolo Mastrolia, Edoardo Mirabella, Giovanni Ossola, and Tiziano Peraro. Integrand-Reduction for Two-Loop Scattering Amplitudes through Multivariate Polynomial Division. *Phys. Rev. D*, 87(8):085026, 2013.

[191] Samuel Abreu, Lance J. Dixon, Enrico Herrmann, Ben Page, and Mao Zeng. The two-loop five-point amplitude in $\mathcal{N} = 4$ super-Yang-Mills theory. *Phys. Rev. Lett.*, 122(12):121603, 2019.

[192] D. Chicherin, T. Gehrmann, J.M. Henn, P. Wasser, Y. Zhang, and S. Zoia. Analytic result for a two-loop five-particle amplitude. *Phys. Rev. Lett.*, 122(12):121602, 2019.

[193] David C. Dunbar, Guy R. Jehu, and Warren B. Perkins. Two-Loop Gravity amplitudes from four dimensional Unitarity. *Phys. Rev. D*, 95(4):046012, 2017.

[194] Dmitry Chicherin, Thomas Gehrmann, Johannes M. Henn, Pascal Wasser, Yang Zhang, and Simone Zoia. The two-loop five-particle amplitude in $\mathcal{N} = 8$ supergravity. *JHEP*, 03:115, 2019.

[195] Samuel Abreu, Lance J. Dixon, Enrico Herrmann, Ben Page, and Mao Zeng. The two-loop five-point amplitude in $\mathcal{N} = 8$ supergravity. *JHEP*, 03:123, 2019.

[196] Morad Aaboud et al. 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.

[197] Georges Aad et al. Search for new phenomena in events with at least three photons collected in $pp$ collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector. *Eur. Phys. J. C*, 76(4):210, 2016.

[198] Michal Czakon, David Heymes, and Alexander Mitov. High-precision differential predictions for top-quark pairs at the LHC. *Phys. Rev. Lett.*, 116(8):082003, 2016.

[199] Michal Czakon, Paul Fiedler, David Heymes, and Alexander Mitov. NNLO QCD predictions for fully-differential top-quark pair production at the Tevatron. *JHEP*, 05:034, 2016.

[200] Arnd Behring, Michal Czakon, Alexander Mitov, Andrew S. Papanastasiou, and Rene Poncelet. Higher order corrections to spin correlations in top quark pair production at the LHC. *Phys. Rev. Lett.*, 123(8):082001, 2019.

[201] Michał Czakon, Andreas van Hameren, Alexander Mitov, and Rene Poncelet. Single-jet inclusive rates with exact color at $\mathcal{O}\ (\alpha_s^4)$. *JHEP*, 10:262, 2019.

[202] John M. Campbell, R.Keith Ellis, Ye Li, and Ciaran Williams. Predictions for diphoton production at the LHC through NNLO in QCD. *JHEP*, 07:148, 2016.

[203] Andreas van Hameren. avhlib library. `https://bitbucket.org/hameren/avhlib`.

[204] M. Bury and A. van Hameren. Numerical evaluation of multi-gluon amplitudes for High Energy Factorization. *Comput. Phys. Commun.*, 196:592–598, 2015.

[205] Fabio Cascioli, Philipp Maierhofer, and Stefano Pozzorini. Scattering Amplitudes with Open Loops. *Phys. Rev. Lett.*, 108:111601, 2012.

[206] Federico Buccioni, Jean-Nicolas Lang, Jonas M. Lindert, Philipp Maierhöfer, Stefano Pozzorini, Hantian Zhang, and Max F. Zoller. OpenLoops 2. *Eur. Phys. J. C*, 79(10):866, 2019.

[207] Stefano Actis, Ansgar Denner, Lars Hofer, Jean-Nicolas Lang, Andreas Scharf, and Sandro Uccirati. RECOLA: REcursive Computation of One-Loop Amplitudes. *Comput. Phys. Commun.*, 214:140–173, 2017.

[208] S. Borowka, G. Heinrich, S. Jahn, S.P. Jones, M. Kerner, J. Schlenk, and T. Zirke. pySecDec: a toolbox for the numerical evaluation of multi-scale integrals. *Comput. Phys. Commun.*, 222:313–326, 2018.

[209] Bruno Haible, Richard B. Kreckel, and Alexei Sheplyakov. CLN library. `https://www.ginac.de/CLN/`.

[210] Giuseppe Bevilacqua, Michal Czakon, Andreas van Hameren, Costas G. Papadopoulos, and Malgorzata Worek. Complete off-shell effects in top quark pair hadroproduction with leptonic decay at next-to-leading order. *JHEP*, 02:083, 2011.

[211] G. Bevilacqua, H.B. Hartanto, M. Kraus, and M. Worek. Top Quark Pair Production in Association with a Jet with Next-to-Leading-Order QCD Off-Shell Effects at the Large Hadron Collider. *Phys. Rev. Lett.*, 116(5):052003, 2016.

[212] G. Bevilacqua, H.B. Hartanto, M. Kraus, and M. Worek. Off-shell Top Quarks with One Jet at the LHC: A comprehensive analysis at NLO QCD. *JHEP*, 11:098, 2016.

[213] S.P. Jones, M. Kerner, and G. Luisoni. Next-to-Leading-Order QCD Corrections to Higgs Boson Plus Jet Production with Full Top-Quark Mass Dependence. *Phys. Rev. Lett.*, 120(16):162001, 2018.

[214] G. Bevilacqua, H.B. Hartanto, M. Kraus, T. Weber, and M. Worek. Hard Photons in Hadroproduction of Top Quarks with Realistic Final States. *JHEP*, 10:158, 2018.

[215] Z. Kassabov. Zaharid/GPTree: Interpolation library using Gaussian Processes and KDTrees. `https://doi.org/10.5281/zenodo.3571309`.

[216] James W. Cooley and John W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Math. Comput.*, 19:297–301, 1965.

[217] J. M. Pollard. The fast fourier transform in a finite field. *Mathematics of Computation*, 25(114):365–374, 1971.

[218] R. Agarwal and C. Burrus. Fast convolution using fermat number transforms with applications to digital filtering. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 22(2):87–97, 1974.

[219] W. M. Gentleman and G. Sande. Fast fourier transforms: For fun and profit. In *Proceedings of the November 7-10, 1966, Fall Joint Computer Conference*, AFIPS '66 (Fall), page 563–578, New York, NY, USA, 1966. Association for Computing Machinery.

[220] K. Singhal and J. Vlach. Interpolation using the fast fourier transform. *Proceedings of the IEEE*, 60(12):1558–1558, 1972.

[221] John D. Lipson. *Elements of algebra and algebraic computing.* Addison-Wesley, Reading, Massachusetts, 1st edition, 1981. ISBN 0-201-04115-4.

[222] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979.

[223] Georges Aad et al. (ATLAS collaboration). HEPData repository. `https://www.hepdata.net/record/ins1644367`.

[224] Stefano Frixione. Isolated photons in perturbative QCD. *Phys. Lett. B*, 429:369–374, 1998.

[225] Stefano Catani, Leandro Cieri, Daniel de Florian, Giancarlo Ferrera, and Massimiliano Grazzini. Diphoton production at the LHC: a QCD study up to NNLO. *JHEP*, 04:142, 2018.

[226] Stefano Catani, Leandro Cieri, Daniel de Florian, Giancarlo Ferrera, and Massimiliano Grazzini. Diphoton production at hadron colliders: a fully-differential QCD calculation at NNLO. *Phys. Rev. Lett.*, 108:072001, 2012. [Erratum: Phys.Rev.Lett. 117, 089901 (2016)].

# Appendix A

# Pentagon function identities

Here we present the pentagon function identities which we derived as described in Section 4.3.1 of Chapter 4. The identities will be grouped by their transcendental weight.

## A.1 Notation

The pentagon functions are defined in Ref. [33] in the generic form $f^{(i)}_{j,k}(v_1, v_2, v_3, v_4, v_5)$, where the index $(i)$ cyclically permutes the arguments $\{v_n\}$. We find it convenient to introduce a modified notation for two reasons: firstly, in many functions, some of the 5 variables $v_n$ are in fact dummy variables. Secondly, the identities that we present here include non-cyclic permutations of the arguments $\{v_n\}$ as well as including linear combinations of $v_n$. We label our functions $f_{j,k}$ and distinguish them from the original functions $f^{(i)}_{j,k}$ by the lack of the superscript $(i)$. Our functions are defined in terms of $f^{(i)}_{j,k}$ in the following way:

$$\begin{aligned}
f_{1,1}(a) &\equiv f^{(1)}_{1,1}(a,\_,\_,\_,\_)\\
f_{2,1}(a,b) &\equiv f^{(1)}_{2,1}(a,\_,b,\_,\_)\\
f_{3,1}(a,b) &\equiv f^{(1)}_{3,1}(a,\_,b,\_,\_)\\
f_{3,2}(a,b) &\equiv f^{(1)}_{3,2}(a,\_,b,\_,\_)\\
f_{3,3}(a,b,c) &\equiv f^{(1)}_{3,3}(a,b,\_,c,\_)\\
f_{3,4}(a,b,c,d,e) &\equiv f^{(1)}_{3,4}(a,b,c,d,e)\\
f_{4,1}(a,b) &\equiv f^{(1)}_{4,1}(a,\_,b,\_,\_)\\
f_{4,2}(a,b) &\equiv f^{(1)}_{4,2}(a,\_,b,\_,\_)
\end{aligned}$$

$$\begin{aligned}
f_{4,3}(a,b) &\equiv f_{4,3}^{(1)}(a,\_,b,\_,\_)\\
f_{4,4}(a,b,c) &\equiv f_{4,4}^{(1)}(a,b,\_,c,\_)\\
f_{4,5}(a,b,c) &\equiv f_{4,5}^{(1)}(a,b,\_,c,\_)\\
f_{4,6}(a,b,c) &\equiv f_{4,6}^{(1)}(a,\_,b,\_,c)\\
f_{4,7}(a,b,c) &\equiv f_{4,7}^{(1)}(a,\_,b,c,\_)\\
f_{4,8}(a,b,c) &\equiv f_{4,8}^{(1)}(\_,a,\_,b,c)\\
f_{4,9}(a,b,c) &\equiv f_{4,9}^{(1)}(a,\_,b,c,\_)\\
f_{4,10}(a,b,c,d,e) &\equiv f_{4,10}^{(1)}(a,b,c,d,e)\\
f_{4,11}(a,b,c,d,e) &\equiv f_{4,11}^{(1)}(a,b,c,d,e)\\
f_{4,12}(a,b,c,d,e) &\equiv f_{4,12}^{(1)}(a,b,c,d,e).
\end{aligned}$$

Here we have used the symbol _ to denote any dummy arguments appearing in the functions that were defined in Ref. [33].

Some of the weight-4 identities require introducing a set of ‘constants’ $h_{j,k}$. The value of each ‘constant’ varies with the kinematic region, just like the $bc4$ ‘constants’. We retain the regional dependence of the constants by giving them kinematic arguments: $h_{j,k}(a,b,\dots)$. These kinematic arguments are only present in order to determine the kinematic region and they do not otherwise affect the value of the $h_{j,k}$ constants. We define these $h_{j,k}$ constants implicitly via the weight-4 identities themselves. We expect that it should in fact be possible to express all of the $h_{j,k}$ constants in terms of the $bc4$ constants.

## A.2 Weight 2

There is one weight-2 identity:

$$f_{2,1}(a,b)+f_{2,1}(b,a)=\frac{1}{2}\left[f_{1,1}(a)-f_{1,1}(b)\right]^2. \qquad \text{(A.1)}$$

## A.3 Weight 3

The weight-3 identities are as follows:

$$f_{3,2}(b,a)=f_{3,1}(a,b) \qquad \text{(A.2)}$$

$$\begin{aligned} f_{3,3}(a,b,c) = 7\zeta_3 &- \zeta_2 f_{1,1}(a) - \zeta_2 f_{1,1}(b) + 3\zeta_2 f_{1,1}(c) - f_{1,1}(a)f_{1,1}(b)f_{1,1}(c) \\ &+ f_{1,1}(a)f_{1,1}(c)^2 + f_{1,1}(b)f_{1,1}(c)^2 - f_{1,1}(c)^3 - \zeta_2 f_{1,1}(c-a-b) \\ &+ f_{1,1}(a)f_{1,1}(b)f_{1,1}(c-a-b) - f_{1,1}(a)f_{1,1}(c)f_{1,1}(c-a-b) \\ &- f_{1,1}(b)f_{1,1}(c)f_{1,1}(c-a-b) + f_{1,1}(c)^2 f_{1,1}(c-a-b) \\ &- f_{3,3}(a,c-a-b,c) - f_{3,3}(c-a-b,b,c) \end{aligned} \tag{A.3}$$

$$f_{3,3}(a,b,c) = f_{3,3}(b,a,c) \tag{A.4}$$

$$f_{3,4}(a,b,c,d,e) = f_{3,4}(e,d,c,b,a) \tag{A.5}$$

$$f_{3,4}(a,b,c,d,e) = f_{3,4}(e,a,b,c,d). \tag{A.6}$$

## A.4 Weight 4

The weight-4 identities are as follows:

$$f_{4,2}(a,b) = f_{4,1}(b,a) \tag{A.7}$$

$$f_{4,3}(a,b) = -f_{4,3}(b,a) \tag{A.8}$$

$$f_{4,4}(a,b,c) = f_{4,4}(b,a,c) \tag{A.9}$$

$$\begin{aligned} f_{4,6}(a,b,c) = h_{4,6}(a,b,c) &+ \frac{1}{2} f_{1,1}(a)^3 f_{1,1}(c) - \frac{3}{2} f_{1,1}(a)^2 f_{1,1}(b) f_{1,1}(c) \\ &+ \frac{3}{2} f_{1,1}(a) f_{1,1}(b)^2 f_{1,1}(c) - 2 f_{1,1}(b)^3 f_{1,1}(c) + \frac{3}{2} f_{1,1}(b)^2 f_{1,1}(c)^2 \\ &- 3 f_{1,1}(a) f_{1,1}(c) f_{2,1}(a,b) + 6 f_{1,1}(b) f_{1,1}(c) f_{2,1}(a,b) - 3 f_{1,1}(c)^2 f_{2,1}(a,b) \\ &+ 6 f_{1,1}(c) f_{3,1}(a,b) + 3 f_{1,1}(c) f_{3,1}(b,a) + f_{4,5}(a,c,b) \end{aligned} \tag{A.10}$$

$$\begin{aligned} f_{4,8}(a,c,b) = h_{4,8}(a,c,b) &+ 3\zeta_3 f_{1,1}(a) + \frac{3}{2}\zeta_2 f_{1,1}(a)^2 + \frac{9}{8} f_{1,1}(a)^4 - \frac{17}{4} f_{1,1}(a)^3 f_{1,1}(b) \\ &+ \frac{21}{8} f_{1,1}(a)^2 f_{1,1}(b)^2 - \frac{3}{4} f_{1,1}(a) f_{1,1}(b)^3 + \frac{3}{16} f_{1,1}(b)^4 - \frac{15}{4} f_{1,1}(a)^3 f_{1,1}(c) \\ &+ 15 f_{1,1}(a)^2 f_{1,1}(b) f_{1,1}(c) - 9 f_{1,1}(a) f_{1,1}(b)^2 f_{1,1}(c) + 2 f_{1,1}(b)^3 f_{1,1}(c) \end{aligned}$$

$$\begin{aligned}
&+\frac{21}{8}f_{1,1}(a)^2f_{1,1}(c)^2-9f_{1,1}(a)f_{1,1}(b)f_{1,1}(c)^2+3f_{1,1}(b)^2f_{1,1}(c)^2\\
&-\frac{3}{4}f_{1,1}(a)f_{1,1}(c)^3+f_{1,1}(b)f_{1,1}(c)^3+\frac{3}{16}f_{1,1}(c)^4-\frac{3}{2}f_{1,1}(a)^2f_{2,1}(a,b)\\
&+3f_{1,1}(a)f_{1,1}(b)f_{2,1}(a,b)-\frac{3}{2}f_{1,1}(b)^2f_{2,1}(a,b)+6f_{1,1}(a)f_{1,1}(c)f_{2,1}(a,b)\\
&-6f_{1,1}(b)f_{1,1}(c)f_{2,1}(a,b)-3f_{1,1}(c)^2f_{2,1}(a,b)-\frac{3}{2}f_{1,1}(a)^2f_{2,1}(a,c)\\
&+9f_{1,1}(a)f_{1,1}(b)f_{2,1}(a,c)-3f_{1,1}(b)^2f_{2,1}(a,c)+3f_{1,1}(a)f_{1,1}(c)f_{2,1}(a,c)\\
&-3f_{1,1}(b)f_{1,1}(c)f_{2,1}(a,c)-\frac{3}{2}f_{1,1}(c)^2f_{2,1}(a,c)+3f_{2,1}(a,b)f_{2,1}(a,c)\\
&+3f_{1,1}(a)f_{3,1}(a,b)-3f_{1,1}(b)f_{3,1}(a,b)-6f_{1,1}(c)f_{3,1}(a,b)\\
&+3f_{1,1}(a)f_{3,1}(a,c)-3f_{1,1}(b)f_{3,1}(a,c)-3f_{1,1}(c)f_{3,1}(a,c)\\
&+3f_{1,1}(a)f_{3,1}(b,a)-3f_{1,1}(b)f_{3,1}(b,a)-6f_{1,1}(c)f_{3,1}(b,a)\\
&+3f_{1,1}(a)f_{3,1}(c,a)-9f_{1,1}(b)f_{3,1}(c,a)-3f_{1,1}(c)f_{3,1}(c,a)\\
&+3f_{1,1}(a)f_{3,3}(b,c,a)-3f_{1,1}(b)f_{3,3}(b,c,a)-3f_{1,1}(c)f_{3,3}(b,c,a)\\
&-\frac{3}{2}f_{4,3}(b,a)-\frac{3}{2}f_{4,3}(c,a)+\frac{1}{2}f_{4,4}(c,b,a)-f_{4,5}(c,b,a)+f_{4,7}(a,b,c)
\end{aligned}\tag{A.11}$$

$$\begin{aligned}
f_{4,9}(a,b,c)={}&h_{4,9}(a,b,c)+\frac{4}{3}f_{1,1}(a)^3f_{1,1}(b)-\frac{4}{3}f_{1,1}(a)^3f_{1,1}(c)-4f_{1,1}(a)f_{1,1}(b)^2f_{1,1}(c)\\
&+\frac{2}{3}f_{1,1}(b)^3f_{1,1}(c)+4f_{1,1}(a)f_{1,1}(b)f_{1,1}(c)^2-\frac{2}{3}f_{1,1}(b)f_{1,1}(c)^3\\
&+8f_{1,1}(a)f_{1,1}(c)f_{2,1}(a,b)-4f_{1,1}(b)f_{1,1}(c)f_{2,1}(a,b)\\
&-8f_{1,1}(a)f_{1,1}(b)f_{2,1}(a,c)+4f_{1,1}(b)f_{1,1}(c)f_{2,1}(a,c)-8f_{1,1}(c)f_{3,1}(a,b)\\
&+8f_{1,1}(b)f_{3,1}(a,c)-8f_{1,1}(c)f_{3,1}(b,a)+8f_{1,1}(b)f_{3,1}(c,a)+f_{4,9}(a,c,b)
\end{aligned}\tag{A.12}$$

$$f_{4,10}(a,b,c,d,e)=f_{4,10}(a,e,d,c,b)\tag{A.13}$$

$$\begin{aligned}
f_{4,11}(c,b,e,a,d)={}&h_{4,11}(c,b,e,a,d)+6\zeta_3f_{1,1}(b)+\frac{9}{2}f_{1,1}(a)^3f_{1,1}(b)+3\zeta_2f_{1,1}(b)^2\\
&-\frac{45}{8}f_{1,1}(a)^2f_{1,1}(b)^2+\frac{9}{2}f_{1,1}(a)f_{1,1}(b)^3-\frac{15}{16}f_{1,1}(b)^4-6\zeta_3f_{1,1}(c)\\
&-\frac{9}{2}f_{1,1}(a)^3f_{1,1}(c)-3\zeta_2f_{1,1}(c)^2+\frac{15}{4}f_{1,1}(a)^2f_{1,1}(c)^2
\end{aligned}$$

$$
\begin{aligned}
&+\frac{3}{2}f_{1,1}(a)f_{1,1}(c)^3-\frac{21}{16}f_{1,1}(c)^4-3\zeta_3f_{1,1}(d)+\frac{9}{2}\zeta_2f_{1,1}(a)f_{1,1}(d)\\
&-\frac{3}{2}f_{1,1}(a)^3f_{1,1}(d)+3f_{1,1}(a)^2f_{1,1}(b)f_{1,1}(d)-6f_{1,1}(a)f_{1,1}(b)^2f_{1,1}(d)\\
&+\frac{21}{4}f_{1,1}(b)^3f_{1,1}(d)-\frac{3}{2}\zeta_2f_{1,1}(c)f_{1,1}(d)-\frac{3}{4}f_{1,1}(a)^2f_{1,1}(c)f_{1,1}(d)\\
&-3f_{1,1}(a)f_{1,1}(b)f_{1,1}(c)f_{1,1}(d)+\frac{3}{4}f_{1,1}(b)^2f_{1,1}(c)f_{1,1}(d)\\
&+\frac{3}{2}f_{1,1}(a)f_{1,1}(c)^2f_{1,1}(d)+\frac{3}{4}f_{1,1}(b)f_{1,1}(c)^2f_{1,1}(d)-\frac{1}{2}f_{1,1}(c)^3f_{1,1}(d)\\
&-\frac{3}{2}\zeta_2f_{1,1}(d)^2+\frac{3}{2}f_{1,1}(a)^2f_{1,1}(d)^2+\frac{9}{4}f_{1,1}(a)f_{1,1}(b)f_{1,1}(d)^2\\
&-\frac{33}{4}f_{1,1}(b)^2f_{1,1}(d)^2+\frac{3}{4}f_{1,1}(c)^2f_{1,1}(d)^2-\frac{3}{2}f_{1,1}(a)f_{1,1}(d)^3\\
&+\frac{31}{4}f_{1,1}(b)f_{1,1}(d)^3-\frac{1}{2}f_{1,1}(c)f_{1,1}(d)^3-\frac{21}{8}f_{1,1}(d)^4+3\zeta_3f_{1,1}(e)\\
&-\frac{9}{2}\zeta_2f_{1,1}(a)f_{1,1}(e)+\frac{3}{2}f_{1,1}(a)^3f_{1,1}(e)+\frac{3}{2}\zeta_2f_{1,1}(b)f_{1,1}(e)\\
&+\frac{3}{4}f_{1,1}(a)^2f_{1,1}(b)f_{1,1}(e)-\frac{9}{4}f_{1,1}(a)f_{1,1}(b)^2f_{1,1}(e)+\frac{1}{2}f_{1,1}(b)^3f_{1,1}(e)\\
&+\frac{3}{4}f_{1,1}(a)^2f_{1,1}(c)f_{1,1}(e)+3f_{1,1}(a)f_{1,1}(b)f_{1,1}(c)f_{1,1}(e)\\
&-\frac{3}{4}f_{1,1}(b)^2f_{1,1}(c)f_{1,1}(e)-\frac{21}{2}f_{1,1}(a)f_{1,1}(c)^2f_{1,1}(e)\\
&-\frac{3}{4}f_{1,1}(b)f_{1,1}(c)^2f_{1,1}(e)+\frac{15}{4}f_{1,1}(c)^3f_{1,1}(e)\\
&+\frac{21}{4}f_{1,1}(b)^2f_{1,1}(d)f_{1,1}(e)+\frac{3}{2}f_{1,1}(a)f_{1,1}(c)f_{1,1}(d)f_{1,1}(e)\\
&-\frac{9}{4}f_{1,1}(c)^2f_{1,1}(d)f_{1,1}(e)+\frac{3}{4}f_{1,1}(a)f_{1,1}(d)^2f_{1,1}(e)\\
&-9f_{1,1}(b)f_{1,1}(d)^2f_{1,1}(e)+\frac{19}{4}f_{1,1}(d)^3f_{1,1}(e)+\frac{3}{2}\zeta_2f_{1,1}(e)^2\\
&-\frac{27}{8}f_{1,1}(a)^2f_{1,1}(e)^2-\frac{21}{8}f_{1,1}(b)^2f_{1,1}(e)^2+\frac{51}{4}f_{1,1}(a)f_{1,1}(c)f_{1,1}(e)^2\\
&-6f_{1,1}(c)^2f_{1,1}(e)^2-\frac{3}{4}f_{1,1}(a)f_{1,1}(d)f_{1,1}(e)^2\\
&+\frac{27}{4}f_{1,1}(b)f_{1,1}(d)f_{1,1}(e)^2+\frac{3}{2}f_{1,1}(c)f_{1,1}(d)f_{1,1}(e)^2\\
&-\frac{33}{8}f_{1,1}(d)^2f_{1,1}(e)^2-\frac{15}{4}f_{1,1}(a)f_{1,1}(e)^3-f_{1,1}(b)f_{1,1}(e)^3\\
&+\frac{11}{4}f_{1,1}(c)f_{1,1}(e)^3+\frac{5}{4}f_{1,1}(d)f_{1,1}(e)^3-\frac{9}{8}f_{1,1}(e)^4\\
&+3f_{1,1}(a)^2f_{2,1}(a,b)-6f_{1,1}(a)f_{1,1}(b)f_{2,1}(a,b)+3f_{1,1}(b)^2f_{2,1}(a,b)\\
&+9f_{1,1}(a)f_{1,1}(d)f_{2,1}(a,b)-12f_{1,1}(b)f_{1,1}(d)f_{2,1}(a,b)
\end{aligned}
$$

$$\begin{aligned}
&-\frac{3}{2}f_{1,1}(c)f_{1,1}(d)f_{2,1}(a,b)+\frac{15}{4}f_{1,1}(d)^2f_{2,1}(a,b)\\
&+\frac{3}{2}f_{1,1}(a)f_{1,1}(e)f_{2,1}(a,b)-3f_{1,1}(b)f_{1,1}(e)f_{2,1}(a,b)\\
&+\frac{3}{2}f_{1,1}(c)f_{1,1}(e)f_{2,1}(a,b)-3f_{1,1}(a)^2f_{2,1}(a,c)\\
&+6f_{1,1}(a)f_{1,1}(c)f_{2,1}(a,c)-3f_{1,1}(c)^2f_{2,1}(a,c)\\
&-\frac{3}{2}f_{1,1}(a)f_{1,1}(d)f_{2,1}(a,c)-\frac{3}{2}f_{1,1}(b)f_{1,1}(d)f_{2,1}(a,c)\\
&+3f_{1,1}(c)f_{1,1}(d)f_{2,1}(a,c)-9f_{1,1}(a)f_{1,1}(e)f_{2,1}(a,c)\\
&+\frac{3}{2}f_{1,1}(b)f_{1,1}(e)f_{2,1}(a,c)+12f_{1,1}(c)f_{1,1}(e)f_{2,1}(a,c)\\
&-\frac{15}{4}f_{1,1}(e)^2f_{2,1}(a,c)+\frac{15}{4}f_{1,1}(a)^2f_{2,1}(c,e)\\
&-12f_{1,1}(a)f_{1,1}(c)f_{2,1}(c,e)+\frac{9}{2}f_{1,1}(c)^2f_{2,1}(c,e)\\
&+\frac{3}{2}f_{1,1}(a)f_{1,1}(d)f_{2,1}(c,e)+\frac{9}{4}f_{1,1}(d)^2f_{2,1}(c,e)\\
&+9f_{1,1}(a)f_{1,1}(e)f_{2,1}(c,e)-9f_{1,1}(c)f_{1,1}(e)f_{2,1}(c,e)\\
&-3f_{1,1}(d)f_{1,1}(e)f_{2,1}(c,e)+\frac{9}{2}f_{1,1}(e)^2f_{2,1}(c,e)+\frac{15}{4}f_{1,1}(a)^2f_{2,1}(d,b)\\
&-12f_{1,1}(a)f_{1,1}(b)f_{2,1}(d,b)+\frac{9}{2}f_{1,1}(b)^2f_{2,1}(d,b)\\
&+9f_{1,1}(a)f_{1,1}(d)f_{2,1}(d,b)-9f_{1,1}(b)f_{1,1}(d)f_{2,1}(d,b)\\
&+\frac{9}{2}f_{1,1}(d)^2f_{2,1}(d,b)+\frac{3}{2}f_{1,1}(a)f_{1,1}(e)f_{2,1}(d,b)\\
&-3f_{1,1}(d)f_{1,1}(e)f_{2,1}(d,b)+\frac{9}{4}f_{1,1}(e)^2f_{2,1}(d,b)+\frac{3}{2}f_{1,1}(b)^2f_{2,1}(d,e)\\
&+\frac{3}{2}f_{1,1}(c)^2f_{2,1}(d,e)+\frac{3}{2}f_{1,1}(a)f_{1,1}(d)f_{2,1}(d,e)\\
&-3f_{1,1}(b)f_{1,1}(d)f_{2,1}(d,e)+3f_{1,1}(c)f_{1,1}(d)f_{2,1}(d,e)\\
&+3f_{1,1}(d)^2f_{2,1}(d,e)+\frac{3}{2}f_{1,1}(a)f_{1,1}(e)f_{2,1}(d,e)\\
&+3f_{1,1}(b)f_{1,1}(e)f_{2,1}(d,e)-3f_{1,1}(c)f_{1,1}(e)f_{2,1}(d,e)\\
&-3f_{1,1}(d)f_{1,1}(e)f_{2,1}(d,e)+3f_{1,1}(e)^2f_{2,1}(d,e)-6f_{1,1}(a)f_{3,1}(a,b)\\
&+6f_{1,1}(b)f_{3,1}(a,b)-12f_{1,1}(d)f_{3,1}(a,b)-3f_{1,1}(e)f_{3,1}(a,b)\\
&+6f_{1,1}(a)f_{3,1}(a,c)-6f_{1,1}(c)f_{3,1}(a,c)+3f_{1,1}(d)f_{3,1}(a,c)\\
&+12f_{1,1}(e)f_{3,1}(a,c)-6f_{1,1}(a)f_{3,1}(b,a)+6f_{1,1}(b)f_{3,1}(b,a)\\
&-9f_{1,1}(d)f_{3,1}(b,a)-3f_{1,1}(e)f_{3,1}(b,a)-9f_{1,1}(a)f_{3,1}(b,d)
\end{aligned}$$

$$
\begin{aligned}
&+ 9f_{1,1}(b)f_{3,1}(b,d) - 9f_{1,1}(d)f_{3,1}(b,d) + 3f_{1,1}(e)f_{3,1}(b,d)\\
&+ 6f_{1,1}(a)f_{3,1}(c,a) - 6f_{1,1}(c)f_{3,1}(c,a) + 3f_{1,1}(d)f_{3,1}(c,a)\\
&+ 9f_{1,1}(e)f_{3,1}(c,a) + 9f_{1,1}(a)f_{3,1}(c,e) - 9f_{1,1}(c)f_{3,1}(c,e)\\
&- 3f_{1,1}(d)f_{3,1}(c,e) + 9f_{1,1}(e)f_{3,1}(c,e) - 12f_{1,1}(a)f_{3,1}(d,b)\\
&+ 9f_{1,1}(b)f_{3,1}(d,b) - 9f_{1,1}(d)f_{3,1}(d,b) + 3f_{1,1}(b)f_{3,1}(d,e)\\
&- 3f_{1,1}(c)f_{3,1}(d,e) - 6f_{1,1}(d)f_{3,1}(d,e) + 6f_{1,1}(e)f_{3,1}(d,e)\\
&+ 12f_{1,1}(a)f_{3,1}(e,c) - 9f_{1,1}(c)f_{3,1}(e,c) + 9f_{1,1}(e)f_{3,1}(e,c)\\
&+ 3f_{1,1}(b)f_{3,1}(e,d) - 3f_{1,1}(c)f_{3,1}(e,d) - 6f_{1,1}(d)f_{3,1}(e,d)\\
&+ 6f_{1,1}(e)f_{3,1}(e,d) - 6f_{1,1}(a)f_{3,3}(a,d,b) + 6f_{1,1}(b)f_{3,3}(a,d,b)\\
&- 6f_{1,1}(d)f_{3,3}(a,d,b) - \frac{3}{2}f_{1,1}(e)f_{3,3}(a,d,b) + 6f_{1,1}(a)f_{3,3}(a,e,c)\\
&- 6f_{1,1}(c)f_{3,3}(a,e,c) + \frac{3}{2}f_{1,1}(d)f_{3,3}(a,e,c) + 6f_{1,1}(e)f_{3,3}(a,e,c)\\
&+ 3f_{1,1}(b)f_{3,3}(b,e,d) - 3f_{1,1}(d)f_{3,3}(b,e,d) + 3f_{1,1}(e)f_{3,3}(b,e,d)\\
&- \frac{3}{2}f_{1,1}(d)f_{3,3}(c,b,a) + \frac{3}{2}f_{1,1}(e)f_{3,3}(c,b,a) - 3f_{1,1}(c)f_{3,3}(d,c,e)\\
&- 3f_{1,1}(d)f_{3,3}(d,c,e) + 3f_{1,1}(e)f_{3,3}(d,c,e) + 3f_{4,3}(b,a) + \frac{9}{2}f_{4,3}(b,d)\\
&- 3f_{4,3}(c,a) + \frac{9}{2}f_{4,3}(e,c) - \frac{3}{2}f_{4,3}(e,d) + \frac{1}{2}f_{4,4}(c,d,e) + f_{4,4}(d,a,b)\\
&- f_{4,4}(e,a,c) - \frac{1}{2}f_{4,4}(e,b,d) - f_{4,5}(a,d,b) + f_{4,5}(a,e,c)\\
&+ f_{4,5}(b,e,d) - f_{4,5}(c,d,e) - f_{4,5}(d,a,b) + f_{4,5}(e,a,c)\\
&+ f_{4,11}(b,c,d,a,e)
\end{aligned}
\tag{A.14}
$$

$$
f_{4,12}(a,b,c,d,e) = f_{4,12}(e,d,c,b,a) \tag{A.15}
$$

$$
f_{4,12}(a,b,c,d,e) = f_{4,12}(e,a,b,c,d). \tag{A.16}
$$