Towards a better understanding of transverse momentum resummation

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Martin Lambertsen
aus Wyk auf Föhr

Tübingen
2018

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	21.09.2018
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Werner Vogelsang
2. Berichterstatter:	Prof. Dr. Thomas Gutsche

Föör min wüf Franzi, man dring Ocke an min foomen Wehna

Zusammenfassung

In den letzten Jahren bekam die Resummation von kleinen transversalen Impulsen viel Aufmerksamkeit im Feld der störungstheoretischen Quantenchromodynamic. Sie führte zu einem neuen Faktorisierungsschema, bekannt als „TMD“-Faktorisierung, bei der die Partonverteilungs- und Fragmentationsfunktionen abhängig von transversalen Impulsen werden (*t*ransverse *m*omentum *d*ependent) im Gegensatz zur kollinearen Faktorisierung, die transversale Impulse vernachlässigt.

In der vorliegenden Arbeit beschäftigen wir uns mit mehreren Aspekten der Resummation kleiner transversaler Impulse, welche in direktem Zusammenhang mit dem TMD Formalismus steht. Außerdem führen wir eine globale Analyse der Lepton-Winkelverteilungen im Drell-Yan-Prozess durch.

Im Speziellen führen wir sämtliche Arbeiten durch die notwendig sind für die Bestimmung von nichtperturbativen Parametern, die im Resummationsformalismus unverzichtbar sind, da dieser per Konstruktion auch nicht störungstheoretisch zugängliche Impulsbereiche benutzt. Um den Landaupol, ein Relikt aus dem störungstheoretischen Ansatz zur Quantenchromodynamik, zu umgehen folgen wir der Idee ihn in der komplexen Ebene auszuweichen. Zu diesem Zweck untersuchen wir die Problematik der Entwicklung von Partonverteilungsfunctionen zu komplexen Impulsskalen, wobei wir ein besonderes Augenmerk auf Reproduzierbarkeit und Präzision legen. Zuerst betrachten wir die Möglichkeiten Partonverteilungsfunktionen im Mellinraum zu parametrisieren, in dem die DGLAP Evolutionsgleichungen analytisch gelöst werden können um eine Entwicklung zu komplexen Skalen zu ermöglichen. Wir nutzen kubische Splines, die gute Eigenschaften in Bezug auf Reproduzierbarkeit und Präzision aufzeigen, jedoch potentiell hohe numerische Laufzeiten bedingen können. Daher haben wir im zweiten Schritt eine allgemeine Integrationsroutine entwickelt, die sowohl auf einer CPU als auch auf einer GPU genutzt werden kann. Die Nutzung einer GPU kann dabei numerische Berechnung signifikant beschleunigen. Die daraus entstandene Bibliothek ist fähig *jede* numerische Berechnung mit einer allgemeinen Integrationsroutine zu beschleunigen — auch außerhalb der Teilchenphysik. In einem weiteren Projekt untersuchen wir die Abhängigkeit des Sudakov-Formfaktors von der analytischen Form der laufenden Kopplung der starken Wechselwirkung. Dabei finden wir messbare Unterschiede, die durchaus die Konvergenz zukünftiger Bestimmungen von nichtperturbativen Parametern beeinflussen könnten.

In der Arbeit berechnen wir auch den im transversalen Impuls differentiellen Wirkungsquerschnitt für die semiinklusive tiefinelastische Streuung eines Leptons an einem Hadron genauer als jemals zuvor. Dabei reproduzieren wir alle nicht regulären Teile, die üblicherweise resummiert werden. Zusätzlich bestimmen wir alle zwar regulären

jedoch immer noch divergenten Terme, die nach der Resummation der nicht regulären Terme potentiell dominant werden und daher ebenfalls resummiert werden sollten, um eine physikalisch sinnvolle Verteilung zu erhalten.

Die globale Analyse der Lepton-Winkelverteilungen im Drell-Yan-Prozess ist durchgeführt in der nächst zu führenden Ordnung und berücksichtigt alle zur Zeit verfügbaren experimentellen Daten. Die Studie bestätigt die Vorhersagen der störungstheoretischen Quantenchromodynamik zur festen Ordnung.

Summary

Over the last decades the resummation of small transverse momenta has gained a lot of attention in the field of perturbative Quantum Chromodynamics, leading to a whole new factorization scheme, known as the 'TMD' factorization, where the parton distribution and fragmentation functions become *t*ransverse *m*omentum *d*ependent in contrast to the collinear factorization scheme, where transverse momenta are neglected.

In this thesis we investigate several aspects of the transverse momentum resummation, which is intimately related to the TMD formalism. Furthermore we perform a global analysis of the Drell-Yan lepton angular distributions.

In particular we pave the road for a future determination of the nonperturbative parts, which are a vital ingredient in the resummation formalism, as it uses the nonperturbative momentum region by construction. We follow the idea of a prescription which avoids the Landau pole—a remnant of the perturbative ansatz to Quantum Chromodynamics—in the complex plane. For this purpose we investigate the problem to evolve parton distribution functions to complex scales with special attention to reproducibility and precision. First we examine the possibilities to describe parton distribution functions in Mellin space, where the DGLAP evolution equations can be solved analytically allowing to evolve to complex scales. We use cubic splines, which show great properties in terms of reproducibility and precision, while potentially causing large numerical costs. In a second step we developed a general purpose integration routine that can be used on a CPU as well as on a GPU, where the use of the latter accelerates numerical calculations substantially. The resulting library is capable to accelerate *every* numerical calculation including an integration also outside of the field of particle physics. In another project we examine the dependence of the Sudakov form factor on the analytical form of the running coupling constant and demonstrate measurable differences which may influence the convergence of a future fit of the nonperturbative parameters.

Additionally we calculate the transverse momentum differential cross section of the semi inclusive deep inelastic process more precisely than ever before. We reproduce the non-regular parts, which are typically resummed, and determine additionally all terms which are divergent for small transverse momentum but still regular and hence become potentially dominant after resummation and should be resummed too to achieve a physical distribution.

The global analysis of the Drell-Yan lepton angular distributions is performed in next-to-leading order and takes into account all currently available experimental data. The study confirms the fixed-order prediction of perturbative Quantum Chromodynamics.

Contents

Introduction

It's pronounced 'nucular'.

— *Homer J. Simpson (1998)*

The goal of physics is to describe the extremely complex structures found in nature by the formulation of a preferably simple underlying theory, which then in turn illuminates all possible observations (ideally made by reproducible experiments). For example the large amount of chemical substances can be decomposed into aggregates of only a few dozen elements, called atoms. The complex substances and molecules are built out of atoms like lego bricks, hence the original complexity is greatly reduced by describing the atoms and their interactions. Going one step further even the still large number of different atoms can be explained by the fact that they are compound objects of more fundamental particles: protons, neutrons and electrons. The first two of them are again found to be composed out of up- and down-quarks.[1]

Particle physics is concerned with these smallest building blocks that have been found yet. In this case 'small' is synonym to the statement that these particles can be resolved at very high energy. It turns out the higher the energy scale the more nature is described by symmetries. It is therefore not surprising that nowadays one tries to explain low-energy phenomena as a limit of an underlying high-energy theory, partly by breaking some of the symmetries.

The currently widely accepted model of particle physics is commonly referred to as the 'Standard Model', which consists of a minimal number of fundamental particles and interactions, which then build up the rich phenomenology that can be observed in nature. It describes all of the known fundamental forces except gravity, which can still not be unified with the other forces. Guided by the principles we described before it is believed that gravity can be unified with the other forces at an energy scale which is still not attainable by experiments or astronomical observations. The Standard Model describes the electromagnetic, the weak and the strong force with unmatched precision and predictive power. It predicted for example the existence of the top quark, which then could be detected in 1995. Only recently the last building block of the Standard Model, the Higgs boson, has been discovered in 2012 and the Higgs mechanism has been awarded by the Nobel prize only one year later.

However, the Nobel prize of 2015 reminds us that the Standard Model still cannot describe all phenomena we observe, as neutrinos are massless in the Standard Model

[1] Of course other quark flavours are also part of the proton and neutron, however they only contribute as sea particles, which means they only exist for short time scales by the uncertainty principle.

but are known to have mass, which has been demonstrated by the measurement of neutrino oscillations. Another example is the presence of Dark Matter, which can be inferred because of its gravitational effects but is not part of the Standard Model. This observation leads to the surprising conclusion that only roughly 15 percent of the matter in the universe is described by the Standard Model, while all the rest is Dark Matter. This kind of matter does not interact with the particles we know in any way we are able to detect. By other cosmological observations we also know about Dark Energy, which degrades the visible matter to be only about five percent of the total energy in the universe. As a consequence of these deficiencies of the Standard Model one of the most important tasks is to find a theory that can explain phenomena that the Standard Model is not being aware of, id est which are beyond the Standard Model. A vital part of this research is to improve the predictions that are made by the Standard Model to be able to state whether a measurement is possible within the model or not.

As mentioned, one of the forces described in the Standard Model is the strong force, which describes the interaction of colored particles. The force is unique in a sense that particles that interact via the strong force cannot be detected directly but only in colorless, compound states. Still it is arguably the most important force for measurements at hadron colliders, which nowadays produce the highest energy collisions in experiment. While the perturbative approach to Quantum Chromodynamics can describe cross sections very precisely in most regions of the phase space at high energy as it uses the strong coupling constant as perturbative parameter which becomes small at high energies, it fails terribly at some regions with fixed order predictions. The reason is that in these regions large contributions appear which spoil the original perturbative series systematically.

Historically and also still nowadays one of the most important examples is the description of the transverse momentum differential cross section of the Drell-Yan process and semi inclusive deep inelastic scattering. In the former two hadrons interact via an intermediate boson, which then decays into an lepton anti-lepton pair in the final state, while the latter describes the interaction of a lepton with a hadron, which is breaking apart and the final state consists of the scattered lepton as well as one additional hadron originating from the parton struck by the lepton. If the transverse momentum of the intermediate vector boson becomes small the perturbative series in the strong coupling constant collapses as large logarithms of the transverse momentum over the boson mass systematically appear. By a reorganization of the perturbative series such that it takes the logarithms into account, it is possible to restore the predictive power of the theory in this particular part of the phase space. This reorganization is called resummation. However, the small transverse momentum still leads to some ambiguities, as can be expected because Quantum Chromodynamics is not a perturbative theory at low energies. This becomes noticeable by the Landau pole of the strong coupling constant, which bedevils the resummation. As this particular kind of resummation is carried out in impact parameter space instead of momentum space, the inverse Fourier transform induces scales that are smaller than accessible perturbatively. This signals

clearly that also nonperturbative physics has to be taken into account, while regulating the perturbative part for these configurations.

The easiest way to accomplish the regularization of the perturbative part and to consider nonperturbative effects is to make a nonperturbative model in the impact parameter space and fit the parameters of this model to data while simply freezing the perturbative input at some scale; in fact this is the preferred method currently. Unfortunately it bequeaths not only the nonperturbative parameters but also the need to define the scale which has to separate the perturbative from the nonperturbative regime. It turns out that the parameters of the nonperturbative model depend strongly on this choice, but there is no simple physical argument that could fix this ambiguity. An alternative would be to avoid the Landau pole in the complex plane. By the complexification the scale that separates perturbative and nonperturbative parts becomes superfluous and the parameters of the nonperturbative model lose the induced ambiguity. The drawback is a numerically much more complicated task for the inverse Fourier transform.

In this thesis we will perform all kinds of preparations that have to be done to succeed with a global fit using the complexified approach. Further we illuminate topics that are all too often overseen related to the resummation of transverse momenta. Altogether we are confident that it can lead to an improved understanding of the transverse momentum resummation as a whole.

In chapter 1 we introduce the basics of the perturbative approach to Quantum Chromodynamics, starting from its Lagrangian and the renormalization of the theory, introducing its unique features of asymptotic freedom and confinement, proceeding to the treatment of physical observables using factorization and evolution and finally giving a compact introduction into the topic of resummation.

Chapter 2 presents the main work of this thesis: the development of a general-purpose integration routine that can be executed on an ordinary CPU as well as on a GPU, improving our ability to integrate complex functions in a reasonable amount of time. It is the key ingredient for the determination of the parameters of the nonperturbative model used in small transverse momentum resummation for Drell-Yan as well as the SIDIS process. Of course it can accelerate all physical applications that need an integration routine and hence should improve numerical calculations also in many other fields.

In contrast to most other chapters, chapter 3 is not related to transverse momentum resummation. In contrast it treats the angular distribution of the final state leptons in the Drell-Yan process, performing a global analysis for all experimental data sets available, demonstrating the overall very good agreement of theory with experiment.

Chapter 4 returns to the small transverse momentum resummation. It examines the corresponding observable of the SIDIS process taking into account subleading terms for the very first time. These terms do not have to be resummed, as they are still integrable for small transverse momentum, but cannot be neglected as after resummation they are likely the most important terms in the particular part of the phase space.

In chapter 5 we introduce a new method to parametrize parton distribution functions

in Mellin space, which is in contrast to the commonly used method easier to reproduce and to automatize. It has direct impact to the resummation of small transverse momenta, as the resummation causes the parton distribution functions to be evaluated at a complex scale if the impact parameter is complexified. As it is only possible to evolve the parton distribution functions to a complex scale in Mellin space, the pros and cons of the different ways to describe them in Mellin space have to be carefully discussed.

The last chapter 6 checks the numerical dependence of the resummation formula for different perturbative expressions for the running coupling constant. A priori it is not possible to favor one expression over the other, but we demonstrate that the choice influences the numerical result of the resummation formula in a non negligible way and hence becomes an important ingredient for future determinations of the nonperturbative parameters.

We like to highlight two appendices, which contain more than just additional information. First in appendix C we present the software library fi*n*ael, which is the actual implementation of all the theory presented in chapter 2 to perform general-purpose integrations on the GPU and way more, as it contains also many tools, math or Quantum Chromodynamics related classes that simplify the calculations on the GPU, as for example routines to perform the DGLAP evolution. Second in appendix D we present the whole calculation necessary in chapter 4 in great detail starting from well known techniques such that it might be useful for young students and proceeding to more advanced calculations that have not been presented in literature as far as we know, for example the next-to-next-to-leading order expansion of the kinematic delta distribution.

All chapters are written in a way that they can be read without knowledge of the preceding topics. Although this introduces some repetitions especially in the introductions, this way we ensure that it is possible to follow the topic without unnecessary browsing through the thesis.

Chapter 1

Perturbative quantum chromodynamics

> In general we look for a new law by the following process. First we guess it. Then we compute the consequences of the guess to see what would be implied if this law that we guessed is right. Then we compare the result of the computation to nature, with experiment or experience, compare it directly with observation, to see if it works. If it disagrees with experiment it is wrong. In that simple statement is the key to science.
>
> — *Richard P. Feynman*

The Standard Model of particle physics (SM) is one of the most successful theories in the history of science. It describes nature with very high accuracy resting upon few basic principles and assumptions, combining three of the four known fundamental forces. These are the electromagnetic and the weak force, which are unified in the electroweak sector, and the strong force.[1] All of them are described by relativistic quantum field theories. The underlying gauge groups are

$$\mathrm{SU}(3)_c \times \mathrm{SU}(2)_w \times \mathrm{U}(1)_Y \tag{1.1}$$

where the $\mathrm{SU}(2)_w$ describes the weak isospin I_w and $\mathrm{U}(1)_Y$ the weak hypercharge Y. In this thesis we will set them aside, because we will use only elementary parts like the exchange of the neutral photon and Z boson, which are a result of the electroweak sector $\mathrm{SU}(2)_w \times \mathrm{U}(1)_Y$ after symmetry breaking by the Higgs mechanism [1–3]. Instead we focus on the strong force described by $\mathrm{SU}(3)_c$, where the label 'c' stands for *color*, which labels the corresponding theory as *quantum chromodynamics* (QCD). The massive color charged particles are called *quarks*, which are spin one-half fermions interacting electroweakly as well as strongly. The mediator particles of the strong force are spin one bosons called *gluons*. In this chapter we will explore the rich phenomenology of QCD

[1] The force not described in the SM is the gravitational force, which will also play no role in this thesis because it is much weaker than any force described by the SM at the energy and mass scales which are relevant for this work.

in the perturbative approach and will highlight some key differences to the electroweak sector.

1.1 QCD Lagrangian

The history of QCD starts, as for all gauge theories, with the early work of Yang and Mills [4] who found the general Lagrangian being invariant under local gauge transformations of the SU(N) group. Driven by the necessity to introduce colored particles to describe the particle zoo of hadrons and mesons [5, 6] (especially the discovery of the Δ^{++}-baryon, where the additional degree of freedom was crucial to guarantee an antisymmetric wave function), the birth of QCD may be set to the formulation of the gauge theory by Fritzsch, Gell-Mann and Leutwyler [7, 8].

In this section we will construct the Lagrangian of QCD, starting with the part describing free quarks and their anti-particles

$$\mathcal{L}_{\text{free}} \equiv \sum_{q\in\mathcal{Q}} \bar{\psi}_q(\mathrm{i}\not{\partial} - m_q)\psi_q \tag{1.2}$$

We suppress the Dirac indices of the fermionic quark fields ψ_q and the operator $(\mathrm{i}\not{\partial} - m_q)$ as well as the color indices of the fields (the operator is color diagonal anyway). For a more detailed presentation see [9]. The sum runs over all knowns quark species, which are given by

$$\mathcal{Q} \equiv \big\{u, d, s, c, b, t\big\} \tag{1.3}$$

where the labels u, d, s, c, b and t correspond to the up, down, strange, charm, bottom and top quark, respectively. Equation (1.2) is invariant under the global SU(3)$_c$ gauge transformation

$$\psi_q(x) \to \exp\big[-\mathrm{i}g_s\alpha^a t^a\big]\psi_q(x) \tag{1.4}$$

The t^a are the eight generators of the fundamental representation of SU(3) group given by the Gell-Mann matrices up to a factor of one-half [10], the α^a are the gauge parameters and g_s will turn out to be the strong coupling constant. Now by the gauge principle which enforces the symmetry to hold also locally ($\alpha^a \to \alpha^a(x)$) it is necessary to introduce the eight gluon fields G^a_μ which transform under the gauge transformation as

$$G^a_\mu \to G^a_\mu - (\partial_\mu\alpha^a) - g_s f^{abc} G^b_\mu \alpha^c \tag{1.5}$$

The totally antisymmetric structure constants f^{abc} reflect that the gluons belong to the adjoint representation of the underlying group. We can now modify the free Lagrangian (1.2) by replacing the derivative with the covariant derivative

$$D_\mu \equiv \partial_\mu - \mathrm{i}g_s G^a_\mu t^a \tag{1.6}$$

to render it also invariant under local gauge transformations.[2] Adding the kinetic term of the gluons via the corresponding field strength tensor

$$G^a_{\mu\nu} \equiv \partial_\mu G^a_\nu - \partial_\nu G^a_\mu + g_s f^{abc} G^b_\mu G^c_\nu \tag{1.7}$$

we can now write down the classical QCD Lagrangian

$$\mathcal{L}_{\text{classic}} \equiv \sum_{q\in\mathcal{Q}} \bar{\psi}_q (\mathrm{i}\not{D} - m_q)\psi_q - \frac{1}{4} G^a_{\mu\nu} G^{\mu\nu,a} \tag{1.8}$$

At this point a crucial difference to the electromagnetic sector is apparent. Because SU(3) is non-abelian, which manifests in the structure constants f^{abc} being non-zero, the kinetic term of the gluons includes also self interaction terms. In particular, we find terms cubic and quartic in the gluon fields, which belong to a three- and a four-gluon interaction respectively, or to describe it more vividly: gluons themselves are color-charged particles in contrast to photons, which are electromagnetically neutral.

Quantizing the theory is performed by use of the path integral formalism [11]. During the procedure it is necessary to fix the gauge. Two major ways of doing this are widespread in the community. The first one is using physical gauges like the Coulomb or axial gauges. The latter ones are especially useful to determine parton distribution functions, see also section 1.4 or appendix D.2. They are accomplished by choosing a fixed vector n^μ, where we distinguish between pure axial, light-cone and temporal gauges ($n^2 < 0$, $n^2 = 0$ and $n^2 > 0$ respectively), and implementing $n \cdot G^a = 0$ by adding a gauge fixing term to the Lagrangian, which reads

$$\mathcal{L}_{\text{gauge, axial}} \equiv -\frac{\xi}{2}\left(n^\mu G^a_\mu\right)^2 \tag{1.9}$$

with the gauge parameter ξ, which is arbitrary and cancels in all physical observables. It is common to choose it to be $\xi = 1$ (Feynman gauge), $\xi = 0$ (unitary gauge) or $\xi \to \infty$ (Landau gauge).

The second way to fix the gauge is by choosing a covariant gauge which implements $\partial \cdot G^a = 0$, which can be done similarly by adding the gauge fixing term

$$\mathcal{L}_{\text{gauge, cov}} \equiv -\frac{\xi}{2}\left(\partial^\mu G^a_\mu\right)^2 \tag{1.10}$$

By choosing a covariant gauge the resulting gluon propagator simplifies greatly compared to axial gauges, but as a drawback unphysical polarizations appear. These have to be canceled out by introducing an additional field, which is a complex anti-commuting scalar field ϕ^a. The field is associated with the so called Faddeev-Popov ghosts [12]. It is important to mention that these ghosts do not map to any physical particles, but

[2] More precisely the transformation of the gluon fields (1.5) is a consequence of the definition of the covariant derivative (1.6) and the invariance of the free Lagrangian (1.2) under local gauge transformations.

are a pure technical trick to eliminate all non-physical degrees of freedom in gluon polarization sums. It is not possible to have ghosts in initial or final states. They contribute typically only at higher-order calculations. The Lagrangian implementing the ghosts is defined by

$$\mathcal{L}_{\text{ghost}} \equiv \left(\partial^\mu \phi^{\dagger a}\right)\left(D_\mu^{ab}\phi^b\right) \tag{1.11}$$

where D_μ^{ab} is the covariant derivative in the adjoint presentation

$$D_\mu^{ab} \equiv \partial_\mu \delta^{ab} + g_s f^{abc} G_\mu^c \tag{1.12}$$

Finally we may now write down the full QCD Lagrangian by summing up the single parts introduced in the last few paragraphs (equations (1.8), (1.10) and (1.11))

$$\begin{aligned}
\mathcal{L}_{\text{QCD}} &\equiv \mathcal{L}_{\text{classic}} + \mathcal{L}_{\text{gauge, cov}} + \mathcal{L}_{\text{ghost}} \\
&= \sum_{q\in\mathcal{Q}} \bar{\psi}_q (\mathrm{i}\partial\!\!\!/ - m_q)\psi_q - \frac{1}{4}\left(\partial_\mu G_\nu^a - \partial_\nu G_\mu^a\right)^2 - \frac{\xi}{2}(\partial^\mu G_\mu^a)^2 + (\partial_\mu \phi^{\dagger a})(\partial^\mu \phi^a) \\
&\quad + g_s \sum_{q\in\mathcal{Q}} \bar{\psi}_q G\!\!\!/^a t^a \psi_q + g_s f^{abc}(\partial^\mu \phi^{\dagger a}) G_\mu^c \phi^b \\
&\quad - \frac{g_s}{2} f^{abc}\left(\partial_\mu G_\nu^a - \partial_\nu G_\mu^a\right) G^{\mu,b} G^{\nu,c} - \frac{g_s^2}{4} f^{abc} f^{ade} G_\mu^b G_\nu^c G^{\mu,d} G^{\nu,e}
\end{aligned} \tag{1.13}$$

We choose a covariant gauge, as we will typically do in the rest of the thesis unless mentioned otherwise. In the last step we wrote down the terms explicitly, such that we can identify the kinematic terms in the first line, the quark-gluon and the ghost-gluon vertices in the second line and the two self interaction terms in the third line. In appendix A we collect the corresponding Feynman rules.

1.2 Renormalization

As the title of this chapter already points out, we want to study QCD by a perturbative approach, id est we want to expand the elements of the scattering matrix in a series of the strong coupling g_s. This approach is only meaningful if the coupling is small, as we will show in the subsequent section 1.3. Calculations beyond the leading order of the series are vital to obtain predictions of high accuracy. These higher-order calculations include loops, where the momenta flowing through these loops are unconstrained and we have to assume all possible momentum configurations. In particular the momenta may become infinitely large. For QCD—as for most relativistic field theories—these cause divergences in the integrals over the loop momenta. We call such theories ultraviolet (UV) divergent. The divergences are a feature of QCD as a whole and are not specific to the perturbative approach [13].

In order to handle UV divergent theories the first step is to make the divergences manifest. This procedure is called *regularization*, which is a purely technical trick. Several different regularization methods exist. In the early days regularization by a

hard cut-off or the Pauli-Villars method [14] were proposed. Decades later *dimensional regularization* [15, 16] revolutionized the field and is still the dominant method in analytical calculations. In the field of lattice QCD the regularization is mostly performed by using the non-zero lattice spacing as regulator, commonly referred to as lattice regularization [17]. Recently a new method has been introduced, the four dimensional regularization [18], where also the renormalization procedure is altered.[3] However, compared to dimensional regularization all of them have drawbacks as being not invariant under Lorentz or gauge transformations or being very pedestrian for actual calculations. In the following we will therefore discuss and use dimensional regularization only. In dimensional regularization the space-time is expanded to d dimensions, where $d = 4-2\varepsilon$ and, at least for UV divergences, $\varepsilon > 0$. By this shift it is possible to calculate the divergent integrals. The result will naturally depend on ε and the divergences manifest as $1/\varepsilon$ poles.

In dimensional regularization one has to take care of the dimension of the Lagrangian to avoid spurious logarithms of mass dependent quantities. The reason is that the action S has to stay dimensionless also in d dimensions ($\dim[S] = 0$). Therefore the Lagrangian must have mass dimension $\dim[\mathcal{L}] = d$. This causes that in particular the coupling constant has to be redefined to stay dimensionless

$$g_s \to \mu_r^{\varepsilon} g_s \tag{1.14}$$

with μ_r being an arbitrary mass scale, which we will call in the following the *renormalization scale*.

Having the UV poles manifest the *renormalization* can take place. It consists of the statement that the Lagrangian in equation (1.13) does not contain the physical but bare parameters, which will turn out to be divergent themselves. We label the bare parameters by an additional subscript, for example $g_{s,0}$. The physical parameters are then defined by the multiplication of a divergent renormalization constant Z with the divergent bare parameters

$$\psi_q = Z_{\psi_q}^{-1/2} \psi_{q,0} \tag{1.15a}$$
$$m_q = Z_{m_q}^{-1} m_{q,0} \tag{1.15b}$$
$$G_\mu^a = Z_{G^a}^{-1/2} G_{\mu,0}^a \tag{1.15c}$$
$$\xi = Z_\xi^{-1} \xi_0 \tag{1.15d}$$
$$\phi^a = Z_\phi^{-1/2} \phi_0^a \tag{1.15e}$$
$$g_s = Z_g^{-1} g_{s,0} \tag{1.15f}$$

Then the renormalization constants will be defined such that their expansion in the coupling constant cancels exactly the divergences of the bare parameters. As a result

[3] We will see soon that the divergences will be absorbed into a redefinition of the parameters of the Lagrangian. In four dimensional regularization some of them can be absorbed into the vacuum instead.

we find finite physical parameters. The Lagrangian itself is split up into a Lagrangian that is identical to the one in equation (1.13) but now consists of the renormalized fields and parameters and a Lagrangian that absorbs all the rest

$$\mathcal{L}_{\mathrm{QCD},0} = \mathcal{L}_{\mathrm{QCD}} + \mathcal{L}_{\mathrm{ct}} \tag{1.16}$$

We call the terms in $\mathcal{L}_{\mathrm{ct}}$ *counter terms*. From these new Feynman rules can be derived. They have to be taken into account at every order and cancel the UV divergences of the loop diagrams that we still find using $\mathcal{L}_{\mathrm{QCD}}$ (it is still the same as before). For an example see the calculation in appendix D.3.

In order to be renormalizable a theory has to fulfill certain criteria. In general we distinguish three scenarios. A theory with only a finite set of divergent diagrams, such that renormalizing these makes also all diagrams at higher orders finite, is called *super-renormalizable*. An example for a super-renormalizable theory is the scalar ϕ^3-theory in four dimensions. A theory with an infinite number of divergent diagrams is called *renormalizable* if at every order only a finite number is divergent. Furthermore it is necessary that these divergences are accompanied by the parameters of the theory such that the divergences can be absorbed into the renormalization constants Z. Id est, still a finite number of counter terms is sufficient. Finally it is possible that every perturbative order reveals new divergences that need additional parameters to render the observables finite. Because a theory can only have a finite number of parameters, such a theory is *non-renormalizable*. QCD is a renormalizable theory, as has been shown in [19]. In fact, there is proven a more general argument about the renormalizability of non-abelian gauge theories, which includes QCD.[4]

The renormalization of a theory is not unique as it is possible to absorb also finite terms along with the divergences into the renormalization constants. In order to decide which finite terms are subtracted by the renormalization constants, one has to fix the renormalization scheme. In the SM two different schemes are usually applied. For the electroweak part the so called *on-shell scheme* is applied [21, 22], which sets the renormalized parameters equal to the experimentally observed values. In QCD this is not possible, since the quarks are not directly observable (and therefore also not their masses) but are always found in compound states. We will discuss this feature of QCD in detail in the following two sections. Instead in QCD the *modified minimal subtraction scheme* ($\overline{\mathrm{MS}}$—speak 'MS-bar') is used [23]. It is based on the *minimal subtraction scheme* (MS) [24], in which only the $1/\varepsilon$-poles are subtracted by the counter terms. The $\overline{\mathrm{MS}}$ scheme subtracts additionally the constant term $\ln(4\pi) - \gamma_E$ which typically accompanies the poles. γ_E is the Euler-Mascheroni constant. For a more detailed discussion on renormalization of the whole SM see for example [25].

[4] In the same year also the renormalizability of spontaneously broken gauge theories has been shown [20]. As a consequence the QCD as well as the electroweak sector of the SM are renormalizable.

1.3 Asymptotic freedom and confinement

As pointed out in the last section one consequence of renormalization is the introduction of the renormalization scale μ_r. Because the renormalization constants Z depend on μ_r also the coupling g_s and all quark masses m_q depend on this arbitrary scale

$$g_s \to g_s(\mu_r) \tag{1.17a}$$
$$m_q \to m_q(\mu_r) \tag{1.17b}$$

Because the coupling appears only quadratic we define analogously to the fine structure constant in QED α the strong fine structure constant α_s by

$$\alpha_s \equiv \frac{g_s^2}{4\pi} \tag{1.18}$$

However, observables should not depend on an arbitrary scale. To demonstrate this let us consider a dimensionless observable O of a physical process with a dominant momentum Q. Because O is dimensionless it can only depend on other dimensionless arguments. Then the statement that it should be independent of μ_r is given by [26]

$$\begin{aligned} 0 &= \mu_r^2 \frac{\mathrm{d}}{\mathrm{d}\mu_r^2} O\left(\frac{m(\mu_r^2)}{Q}, \frac{Q^2}{\mu_r^2}, \alpha_s(\mu_r^2)\right) \\ &= \left[\mu_r^2 \frac{\partial}{\partial \mu_r^2} + \beta(\alpha_s)\frac{\partial}{\partial \alpha_s} - m\gamma_m(\alpha_s)\frac{\partial}{\partial m}\right] O\left(\frac{m(\mu_r^2)}{Q}, \frac{Q^2}{\mu_r^2}, \alpha_s(\mu_r^2)\right) \end{aligned} \tag{1.19}$$

which is the *renormalization group equation* (RGE). To write down the RGE we introduced the QCD β-function

$$\beta(\alpha_s) \equiv \mu_r^2 \frac{\partial}{\partial \mu_r^2}\alpha_s \tag{1.20}$$

and the anomalous dimension

$$\gamma_m \equiv -\mu_r^2 \frac{1}{m}\frac{\partial}{\partial \mu_r^2} m \tag{1.21}$$

describing the dependence of the coupling and the quark masses on μ_r. Note that the scale dependence of the mass is ultimately a result of our ignorance of the physical masses of the quarks. If they were known, it would be possible to use an on-shell renormalization with no resulting scale dependence.

The β-function as well as the anomalous dimension can be expanded in the coupling. The expansion of the β-function is given in equation (1.26). In case of the coupling we speak of the *running coupling constant* to emphasize the fact that it is not a constant in terms of its scale dependence. Note that we simplified the situation by considering only one mass, which turns out to be justifiable, because we will neglect the quark masses in the following anyway. It can be shown that the effective masses are small for

large momenta in asymptotic free theories [26]. In this section we will demonstrate that QCD is an asymptotic free theory and can therefore safely neglect all quark masses from now on. This simplifies the RGE to

$$\left[\mu_r^2 \frac{\partial}{\partial \mu_r^2} + \beta(\alpha_s) \frac{\partial}{\partial \alpha_s}\right] O\left(\frac{Q^2}{\mu_r^2}, \alpha_s(\mu_r^2)\right) = 0 \tag{1.22}$$

where we omitted the first argument of O because it is trivial after setting all masses to zero

$$O\left(0, \frac{Q^2}{\mu_r^2}, \alpha_s(\mu_r^2)\right) \to O\left(\frac{Q^2}{\mu_r^2}, \alpha_s(\mu_r^2)\right) \tag{1.23}$$

Solving the differential equation (1.20) for α_s we immediately find

$$\ln\left(\frac{Q^2}{\mu_r^2}\right) = \int\limits_{\alpha_s(\mu_r^2)}^{\alpha_s(Q^2)} \frac{\mathrm{d}x}{\beta(x)} \tag{1.24}$$

Differentiating this equation with respect to $\alpha_s(\mu_r^2)$ and Q^2 leads to

$$\frac{\partial \alpha_s(Q^2)}{\partial \alpha_s(\mu_r^2)} = \frac{\beta\big(\alpha_s(Q^2)\big)}{\beta\big(\alpha_s(\mu_r^2)\big)} \tag{1.25a}$$

$$\frac{\partial \alpha_s(Q^2)}{\partial Q^2} = \frac{\beta\big(\alpha_s(Q^2)\big)}{Q^2} \tag{1.25b}$$

With these it is easy to show that $O(1, \alpha_s(Q^2))$ is a solution of the RGE (1.22). This means that the whole scale dependence of the observable is completely determined by the scale dependence of the coupling.

To analyze the scale dependence of the running coupling constant we parametrize the expansion of the β-function in terms of the coupling as

$$\beta(\alpha_s) = -\alpha_s \sum_{n=0}^{\infty} \beta_n \left(\frac{\alpha_s}{4\pi}\right)^{n+1} \tag{1.26}$$

The coefficients β_n are known up to fifth order for the gauge group $SU(N)$ [27–31]. We present the first two coefficients which are given by

$$\beta_0 = \frac{11}{3} C_A - \frac{4}{3} T_R n_f \tag{1.27a}$$

$$\beta_1 = \frac{34}{3} C_A^2 - 4 C_F T_R n_f - \frac{20}{3} C_A T_R n_f \tag{1.27b}$$

where n_f is the number of active quark flavours. C_A and C_F are the quadratic Casimir operators of the adjoint and the fundamental representation of the gauge group. T_R is the Dynkin index of the fundamental representation. For QCD with $N = 3$ the Casimir

operators are given by $C_A = 3$, $C_F = 4/3$ and the Dynkin index is $T_R = 1/2$ and thus the coefficients simplify to

$$\beta_0 = 11 - \frac{2}{3} n_f \tag{1.28a}$$

$$\beta_1 = 102 - \frac{38}{3} n_f \tag{1.28b}$$

Note that the coefficients depend in general on the renormalization scheme, however this is first true for β_2.

We notice that another expansion of the β-function is also commonly used. It is related to the expansion in equation (1.26) by rearranging factors of 4π

$$\beta(\alpha_s) = -\alpha_s^2 \sum_{n=0}^{\infty} b_n \alpha_s^n \tag{1.29}$$

This means the coefficients b_n are related to the β_n by

$$\beta_n = (4\pi)^{n+1} b_n \tag{1.30}$$

Before we solve the differential equation for the running coupling at a fixed order, we want to discuss the qualitative behavior of the coupling depending on the β-function briefly. Based on equation (1.24) we can identify major aspects of the theory. We can distinguish between two cases.

$\boldsymbol{\beta(\alpha_s) > 0}$: In this case the theory has to be perturbative for $Q^2 \to 0$. Because the left-hand side of (1.24) is minus infinity, the coupling $\alpha_s(Q^2)$ has to become smaller than $\alpha_s(\mu_r^2)$. It is even approaching zero, because the β-function starts with a quadratic term, causing a divergence as it should to fulfill (1.24). This behavior is independent of the reference point μ_r. This is called an infrared fix point or *infrared freedom*. The most important example for an infrared free theory is QED. The behavior of the theory for large scales is determined by the β-function at large argument, which is determined by higher orders of the perturbative series. A detailed analysis about this topic can be found in [32].

$\boldsymbol{\beta(\alpha_s) < 0}$: A negative β-function causes the theory to be perturbative for large scales $Q^2 \to \infty$. The left-hand side of equation (1.24) becomes large and in the limit of infinite momentum it becomes plus infinity. As a consequence on the right-hand side the coupling $\alpha_s(Q^2)$ has to become smaller than $\alpha_s(\mu_r^2)$ and with the same argument as in the other case it has to approach zero. The theory has an ultraviolet fix point and is called *asymptotic free*. The most important example is QCD, which is asymptotically free for $n_f < 17$[5] (as the first coefficient

[5] More precisely the limit depends on the number of colors $N_c \equiv N$ and is given by

$$n_f < \frac{11 C_A}{4 T_R} = \frac{11}{2} N_c$$

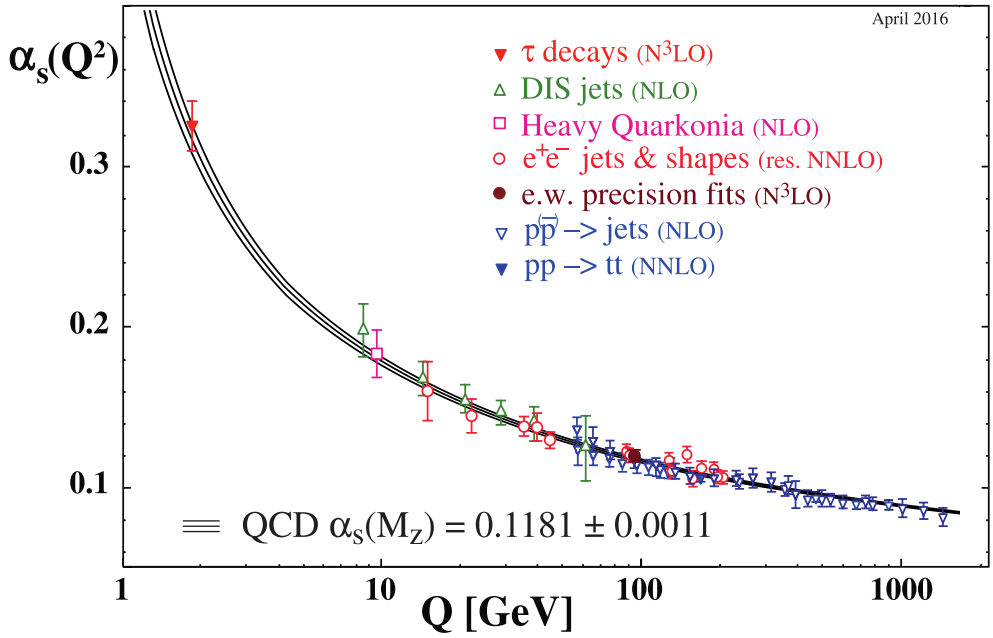


Figure 1.1: Measurements of the strong coupling constant as a function of the energy scale $\alpha_s = \alpha_s(Q^2)$. Figure taken from [36].

β_0 is positive as long as not too many flavours are present, see equation (1.28a)). The asymptotic freedom of QCD has been discovered by Gross and Wilczek [33] and independently by Politzer [34, 35]. In fact the two articles were received only a few days apart and all three of them were awarded the Nobel prize in 2004. In our argument we already used that asymptotic freedom can be justified by the first term of the β-function, as it determines the behavior for small α_s. Higher-order terms cannot change this. Instead they define the coupling at small scales. Again a more detailed analysis is presented in [32].

It is quite easy to solve equation (1.24) at leading order. The result reads

$$\alpha_s(Q^2) = \frac{\alpha_s(\mu_r^2)}{1 + \frac{\beta_0}{4\pi}\alpha_s(\mu_r^2)\ln\left(\frac{Q^2}{\mu_r^2}\right)} \tag{1.31}$$

In chapter 6 we need the coupling at a higher perturbative order. The respective formulas are given there. Equation (1.31) gives us the coupling at a certain scale Q given the coupling at a fixed scale μ_r. Commonly the Z-boson mass M_Z is taken, where $\alpha_s(M_Z^2) = 0.118$. Note that pQCD predicts the variation of the coupling with the energy scale, but cannot predict the overall size. This information has to be extracted from experimental data and is updated frequently [37, 38]. In figure 1.1 we show the current measurements of the strong coupling. We see indeed the asymptotic freedom

of QCD for large scales, which means that quarks and gluon interact weakly in this regime. This confirms the perturbative approach retrospectively. We may estimate the scale at which perturbative QCD becomes valid with $Q \gtrsim 1\,\mathrm{GeV}$. For smaller scales the coupling becomes of order one and perturbative QCD does not apply, as the corrections of higher orders become systematically larger than lower orders. The perturbative formula (1.31) even diverges for small scale Q. The divergence is called the *Landau-pole* Λ_{qcd}. At leading order we find

$$\Lambda_{\mathrm{qcd}} = \mu_r \exp\left(-\frac{2\pi}{\beta_0 \alpha_s(\mu_r^2)}\right) \tag{1.32}$$

Using the Z-boson mass as fix point we can derive the numerical value of the Landau-pole to be $\Lambda_{\mathrm{qcd}} \approx 250\,\mathrm{MeV}$. Inserting the Landau-pole in equation (1.31) we can express the coupling by using the pole as the only parameter

$$\alpha_s(Q^2) = \frac{4\pi}{\beta_0 \ln\left(\frac{Q^2}{\Lambda_{\mathrm{qcd}}^2}\right)} \tag{1.33}$$

At higher orders this formula becomes a series with $\ln^{-1}(Q/\Lambda_{\mathrm{qcd}})$ as the expansion parameter. The next order is reported in chapter 6.

In nature the Landau-pole is nonexistent. The presence of the pole is a relict of the perturbative approach. It indicates the incapacity to describe the strong interaction at small scales. Lattice QCD or Dyson-Schwinger equations can be used to investigate the coupling in this regime [39, 40]. In a nutshell it is found that the coupling becomes nonperturbative but of order one. Some investigations suggest a saturation for small momentum, others find that the coupling decreases again and has a maximum at a few hundred MeV.

The large coupling for small scales has another important phenomenological consequence that we have not mentioned so far, namely *confinement*. It states that the interactions between quarks and gluons become strong at small energy scales. As a consequence they form compound objects which are colorless, called hadrons. To phrase it differently: quarks and gluons are confined in the hadrons. If one separates two colored particles of a hadron the binding energy increases until it is larger than the mass of a new quark anti-quark pair. Because the state including the new pair is favorable the system ends up with two hadrons, which are again colorless. This makes QCD especially demanding because its elementary particles cannot be detected separately. It is also the reason why it is not possible to use an on-shell scheme for the renormalization of the theory.

Confinement is ultimately a result of the non-abelian nature of QCD. Because a gluon does carry color-charge they interact among each other, which gives rise to additional quantum corrections not present for example in QED. These gluon contributions dominate, which can be seen in the coefficient of the β-function in equation (1.27a), where the first summand is due to gluon contributions, while the second summand

originates from quark contributions. This means that the color-charged gluons cause the β-function to be negative and finally make QCD an asymptotic free theory with confinement.[6]

To conclude this section we will finally discuss the importance of QCD in terms of quantum corrections, which are essential for high-accuracy predictions. As can be seen in figure 1.1 for large energies the coupling is of order

$$\alpha_s \sim \mathcal{O}\left(\frac{1}{10}\right) \tag{1.34}$$

while the electromagnetic coupling is about one order of magnitude smaller

$$\alpha \sim \mathcal{O}\left(\frac{1}{100}\right) \tag{1.35}$$

This is the reason why QCD is said to describe the *strong* force. As a consequence QCD corrections are way more important than QED corrections for observables that contain hadronic states. The magnitudes above give a naive approximation about the impact of the corrections. On this basis we may expect a next-to-next-to-leading-order (NNLO) correction in QCD to be of the same size as a next-to-leading-order (NLO) electroweak correction. However, electroweak corrections more often affect the shape of differential observables, as their kinematic dependencies are typically stronger. This means for precision calculations one has to balance the corrections carefully to obtain reasonable results.

1.4 Infrared safety and factorization

In the last section we have shown that the strong coupling α_s is sufficiently small at larges scales to justify perturbative calculations. However, due to confinement the perturbative part is inevitably mingled with the complicated inner structure of hadrons, which is not perturbative. This concerns us in two ways: first in hadron colliders, where obviously it is necessary to distinguish between the inner structure of the hadrons and the actual calculable scattering; second also in experiments with clean initial states, for example e^+e^- colliders, where hadronic states are measured in the final state. Since the detectors are far apart from the collision we will measure hadronized *partons* (quarks or gluons) and consequently have again to distinguish between the calculable scattering and the small-scale (long-distance) effects of hadronization. As a consequence we can use perturbative QCD only for certain observables. They have to be either insensitive to long-distance physics or the long-distance part has to be separable from the perturbatively accessible part.

[6] Due to the central role of the gluon, the experimental evidence was of special interest. It has been found that the gluon with its postulated properties described the data of three-jet events by the PLUTO collaboration at DESY [41, 42] best. This established QCD as the theory of strong interaction.

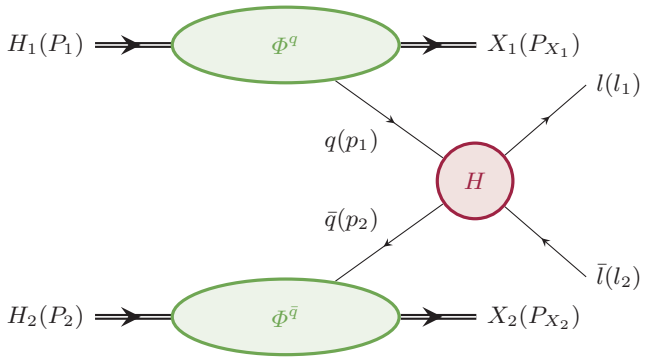


Figure 1.2: Invariant amplitude of the Drell-Yan process at leading order. The incoming hadrons H_i participate in the hard scattering process H via one parton. This is described by the green blobs, which are defined in equations (1.39). The final states consists of the hadronic remnants X_i and the typically measured lepton pair $l\bar{l}$.

The first type of observables, which are insensitive to long-distance physics, are called *infrared safe*. In the perturbative calculation this means that possibly occurring infrared singularities have to cancel among real and virtual corrections. The singularities of the real emissions are typically due to a propagator that is going on-shell.[7] One can track this down to two situations: a collinear or a soft radiation. In the first a parton splits into two collinear partons that share the momentum of their parent parton. In the latter a parton radiates a parton with negligible momentum. In both cases an experiment will not be able to distinguish between the reaction where the radiation has taken place and the reaction where no radiation occurred. Hence also the observable has to be insensitive to these splittings. The classical textbook example for an infrared observable is the inclusive annihilation of a e^+e^- pair into hadrons [43]. Another important infrared-safe observable is the so called thrust distribution [44], which measures how 'jetlike' an event is. This leads to the probably most important observable at current experiments: jets. They describe a bunch of collimated particles. However, no unique definition of a jet exists and the many available definitions differ greatly of what they pronounce to be a jet. Historically the first infrared-safe jet definition has been proposed by Sterman and Weinberg [45]. Currently the mostly used algorithms can be divided into cone [46] and successive recombination algorithms [47–49]. The latter ones are often abbreviated as k_T-algorithms. In this thesis we will not use jets or other observables that do not contain long-distance physics. Therefore we will not go deeper into detail. A more extensive introduction into infrared safety is given for example in [50, 51].

Instead we will use in this thesis the second type of observables exclusively, for which short- and long-distance physics can be separated systematically. The short-

[7] We notice that infrared singularities are present whenever a massless particle becomes soft or collinear to another particle. In the standard model the massless particles are the photon, the gluons and the neutrinos. Because we neglect the mass of the light leptons and quark flavours, we introduce additional infrared singularities. These are a remnant of our treatment but not part of the actual theory. Nevertheless they have to cancel to construct an infrared observable.

distance interactions will be calculable with perturbative QCD, while the long-distance physics has to be measured. The separation is called *factorization*. We will introduce this topic by the example of the Drell-Yan process [52–54], which is for instance the central process of chapter 3. It describes the scattering process of two hadrons which create a lepton pair via an intermediate vector boson. The Drell-Yan process has been extremely popular because of its clean final state and also because it turned out that the perturbative series converges well. In particular we will focus on the photon as the intermediate boson. The process can schematically be described by

$$H_1(P_1) + H_2(P_2) \to \gamma^*(q) + X(P_X) \to l(l_1) + \bar{l}(l_2) + X(P_X) \tag{1.36}$$

where the particle momenta are denoted in parenthesis and X includes all remnants of the incoming hadrons H_i that do not participate in the hard scattering. The following calculations will partly follow [55]. The process is described at leading order by quark-antiquark annihilation and the corresponding invariant amplitude $\mathcal{M}$ is given by

$$\begin{aligned}
&\mathcal{M}\delta^4(P_1 + P_2 - P_X - q) \\
&\quad = \int \frac{\mathrm{d}^4 p_1}{(2\pi)^4} \int \frac{\mathrm{d}^4 p_2}{(2\pi)^4} \int \mathrm{d}^4\eta_1 \, \mathrm{e}^{\mathrm{i}p_1\cdot\eta_1} \int \mathrm{d}^4\eta_2 \, \mathrm{e}^{\mathrm{i}p_2\cdot\eta_2} \\
&\qquad \times H_{ij}(p_1, p_2, l_1, l_2) \, \langle P_X | \bar{\psi}_i(\eta_2)\psi_j(\eta_1) | P_1, P_2 \rangle \, \delta^4(p_1 + p_2 - q)
\end{aligned} \tag{1.37}$$

The momentum of the intermediate boson is given by $q \equiv l_1 + l_2$ and we absorb the leptonic part completely into the hard function H.[8] We denote Dirac indices explicitly, while suppressing color and flavour indices for brevity. The amplitude is shown in figure 1.2, which already implies the factorization. We address this later. First we assume that the hadronic matrix element approximately factorizes

$$\langle P_X | \bar{\psi}_i(\eta_2)\psi_j(\eta_1) | P_1, P_2 \rangle \to \langle P_{X_1} | \psi_j(\eta_1) | P_1 \rangle \, \langle P_{X_2} | \bar{\psi}_i(\eta_2) | P_2 \rangle \tag{1.38}$$

This is consistent with the depiction in figure 1.2. We can now identify the green blobs as Fourier transforms of a hadronic matrix elements where a quark/antiquark is created by acting with the $\psi_i(\eta)/\bar{\psi}_i(\eta)$ on the initial hadron state

$$\Phi_i^q(p, P, P_X) \equiv \int \mathrm{d}^4\eta \, \mathrm{e}^{\mathrm{i}p\cdot\eta} \, \langle P_X | \psi_i(\eta) | P \rangle \tag{1.39a}$$

$$\Phi_i^{\bar{q}}(p, P, P_X) \equiv \int \mathrm{d}^4\eta \, \mathrm{e}^{\mathrm{i}p\cdot\eta} \, \langle P_X | \bar{\psi}_i(\eta) | P \rangle \tag{1.39b}$$

and thus rewrite the invariant amplitude

$$\mathcal{M}\delta^4(P_1 + P_2 - P_X - q) = \int \frac{\mathrm{d}^4 p_1}{(2\pi)^4} \int \frac{\mathrm{d}^4 p_2}{(2\pi)^4} H_{ij}(p_1, p_2, l_1, l_2)\delta^4(p_1 + p_2 - q)$$

[8] In fact it would not be necessary to discuss the Drell-Yan process at all. We could have used vector boson production instead, where we could have omitted the whole leptonic part.

$$\times \int \mathrm{d}^4\eta_1 \; \mathrm{e}^{\mathrm{i}p_1\cdot\eta_1} \langle P_{X_1}|\psi_j(\eta_1)|P_1\rangle$$
$$\times \int \mathrm{d}^4\eta_2 \; \mathrm{e}^{\mathrm{i}p_2\cdot\eta_2} \langle P_{X_2}|\bar{\psi}_i(\eta_2)|P_2\rangle \tag{1.40}$$

where we keep the hadronic matrix elements explicit, because they will be manipulated in the next steps: using the translation operator to shift the argument of the Dirac fields we can make them independent of the space parameters η_i, because the states are all eigenstates of the momentum operator. Therefore we can simplify the expression of the invariant amplitude to

$$\mathcal{M} = H_{ij}(P_1 - P_{X_1}, P_2 - P_{X_2}, l_1, l_2) \langle P_{X_1}|\psi_j(0)|P_1\rangle \langle P_{X_2}|\bar{\psi}_i(0)|P_2\rangle \tag{1.41}$$

where $P_X = P_{X_1} + P_{X_2}$. However, to write down the cross section of the process depicted in figure 1.3 it is more convenient to use the original form

$$\begin{aligned}
\frac{\mathrm{d}\sigma}{\mathrm{d}\mathcal{R}} &= \sum_{q\in\mathcal{Q}} \sum\!\!\!\!\!\!\int_{X_1} \sum\!\!\!\!\!\!\int_{X_2} |\mathcal{M}|^2 (2\pi)^4 \delta^4(P_1 + P_2 - P_X - q) \\
&= \sum_{q\in\mathcal{Q}} \sum\!\!\!\!\!\!\int_{X_1} \int \frac{\mathrm{d}^4p_1}{(2\pi)^4} \int \frac{\mathrm{d}^4p_1'}{(2\pi)^4} \\
&\quad \times \int \mathrm{d}^4\eta_1 \int \mathrm{d}^4\eta_1' \; \mathrm{e}^{\mathrm{i}(p_1\cdot\eta_1 - p_1'\cdot\eta_1')} \langle P_1|\bar{\psi}_l(\eta_1')|P_{X_1}\rangle \langle P_{X_1}|\psi_j(\eta_1)|P_1\rangle \\
&\quad \times \sum\!\!\!\!\!\!\int_{X_2} \int \frac{\mathrm{d}^4p_2}{(2\pi)^4} \int \frac{\mathrm{d}^4p_2'}{(2\pi)^4} \\
&\quad \times \int \mathrm{d}^4\eta_2 \int \mathrm{d}^4\eta_2' \; \mathrm{e}^{\mathrm{i}(p_2\cdot\eta_2 - p_2'\cdot\eta_2')} \langle P_2|\psi_k(\eta_2')|P_{X_2}\rangle \langle P_{X_2}|\bar{\psi}_i(\eta_2)|P_2\rangle \\
&\quad \times H_{ij}(p_1, p_2, l_1, l_2) H^*_{lk}(p_1', p_2', l_1, l_2) (2\pi)^4 \delta^4(p_1 + p_2 - q)
\end{aligned} \tag{1.42}$$

where $\mathrm{d}\mathcal{R}$ denotes the final state phase space of the leptons and also includes the Møller flux factor. Both will play no role in the following discussion. The sum over the quark flavours $q \in \mathcal{Q}$ (see equation (1.3)) is stated explicitly, although the index is suppressed for the Dirac fields and the hard scattering function at the moment. It is important to distinguish the flavour index from the momentum q of the intermediate boson. Further we defined the shorthand notation for the phase space of the hadronic remnant by

$$\sum\!\!\!\!\!\!\int_X \equiv \sum_X \int \frac{\mathrm{d}^3P_X}{2E_{P_X}(2\pi)^3} \tag{1.43}$$

To simplify the expression we notice that for both intermediate states X_i we can use the completeness relation after including the phase space integral

$$\sum\!\!\!\!\!\!\int_X |P_X\rangle \langle P_X| = 1 \tag{1.44}$$

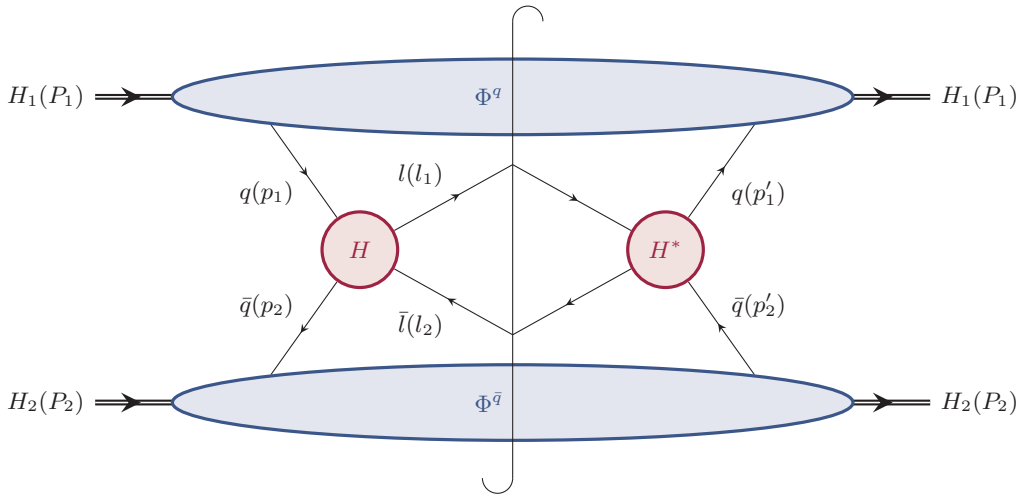


Figure 1.3: Cross section of the Drell-Yan process at leading order. At the final state cut a summation/integration over all states takes place. The quark and antiquark PDF correlators are shown as blue blobs, which are essentially the green blobs of figure 1.2 plus the final state hadron remnants multiplied by their complex conjugates.

As a result we now face matrix elements with two quark fields between two well-defined hadron states instead of a single quark field and an unknown remnant state

$$\begin{aligned}\frac{\mathrm{d}\sigma}{\mathrm{d}\mathcal{R}} = \sum_{q\in\mathcal{Q}} \int \frac{\mathrm{d}^4p_1}{(2\pi)^4} \int \frac{\mathrm{d}^4p_1'}{(2\pi)^4} \int \mathrm{d}^4\eta_1 \int \mathrm{d}^4\eta_1' \; \mathrm{e}^{\mathrm{i}(p_1\cdot\eta_1 - p_1'\cdot\eta_1')} \langle P_1|\bar{\psi}_l(\eta_1')\psi_j(\eta_1)|P_1\rangle \\ \times \int \frac{\mathrm{d}^4p_2}{(2\pi)^4} \int \frac{\mathrm{d}^4p_2'}{(2\pi)^4} \int \mathrm{d}^4\eta_2 \int \mathrm{d}^4\eta_2' \; \mathrm{e}^{\mathrm{i}(p_2\cdot\eta_2 - p_2'\cdot\eta_2')} \langle P_2|\psi_k(\eta_2')\bar{\psi}_i(\eta_2)|P_2\rangle \\ \times H_{ij}(p_1,p_2,l_1,l_2) H_{lk}^*(p_1',p_2',l_1,l_2) (2\pi)^4\delta^4(p_1+p_2-q) \end{aligned} \tag{1.45}$$

This formula is shown schematically in figure 1.3. The blue blobs Φ^q and $\Phi^{\bar{q}}$ are the quark and the antiquark correlators respectively, which we define soon. Further simplification can be achieved by using the invariance under translations of the matrix elements. Similar to the simplification step of the invariant amplitude in equation (1.41) we can use the translation operator to shift the arguments of the fields and make one of the Fourier integrals redundant

$$\begin{aligned}&\int \mathrm{d}^4\eta \int \mathrm{d}^4\eta' \; \mathrm{e}^{\mathrm{i}(p\cdot\eta - p'\cdot\eta')} \langle P|\bar{\psi}_l(\eta')\psi_j(\eta)|P\rangle \\ &= \frac{1}{16} \int \mathrm{d}^4\lambda \; \mathrm{e}^{\frac{\mathrm{i}}{2}(p-p')\cdot\lambda} \int \mathrm{d}^4\kappa \; \mathrm{e}^{\frac{\mathrm{i}}{2}(p+p')\cdot\kappa} \\ &\quad \times \langle P| \, \mathrm{e}^{\frac{\mathrm{i}}{2}\hat{p}\cdot(\lambda-\kappa)} \, \bar{\psi}_l(0) \, \mathrm{e}^{-\frac{\mathrm{i}}{2}\hat{p}\cdot(\lambda-\kappa)} \, \mathrm{e}^{\frac{\mathrm{i}}{2}\hat{p}\cdot(\lambda+\kappa)} \, \psi_j(0) \, \mathrm{e}^{-\frac{\mathrm{i}}{2}\hat{p}\cdot(\lambda+\kappa)} \, |P\rangle \\ &= (2\pi)^4\delta^4(p-p') \int \mathrm{d}^4\kappa \; \mathrm{e}^{\mathrm{i}p\cdot\kappa} \langle P|\bar{\psi}_l(0)\psi_j(\kappa)|P\rangle \end{aligned} \tag{1.46}$$

where we used the sum and the difference of the two integration variables $\lambda_\mu \equiv \eta_\mu + \eta'_\mu$ and $\kappa_\mu \equiv \eta_\mu - \eta'_\mu$. Note that it is essential that initial and final state of the matrix element are equal to achieve the translation invariance. Now eliminating also the momentum integrals via the delta distributions, we can write down the final expression for the cross section

$$\begin{aligned}\frac{\mathrm{d}\sigma}{\mathrm{d}\mathcal{R}} = \sum_{q\in\mathcal{Q}} \int \mathrm{d}^4 p_1 \int \mathrm{d}^4 p_2 \, (2\pi)^4 \delta^4(p_1 + p_2 - q) \\ \times \operatorname{tr}\big(H(p_1, p_2, l_1, l_2)\Phi^q(p_1, P_1) H^*(p_1, p_2, l_1, l_2)\Phi^{\bar{q}}(p_2, P_2)\big)\end{aligned} \tag{1.47}$$

where we defined the *fully unintegrated quark* and *antiquark correlators*

$$\Phi^q_{ij}(p, P) \equiv \int \frac{\mathrm{d}^4\eta}{(2\pi)^4} \, \mathrm{e}^{\mathrm{i}p\cdot\eta} \, \langle P|\bar{\psi}_j(0)\psi_i(\eta)|P\rangle \tag{1.48a}$$

$$\Phi^{\bar{q}}_{ij}(p, P) \equiv \int \frac{\mathrm{d}^4\eta}{(2\pi)^4} \, \mathrm{e}^{\mathrm{i}p\cdot\eta} \, \langle P|\psi_i(0)\bar{\psi}_j(\eta)|P\rangle \tag{1.48b}$$

The typical setup in high energy colliders consists of interacting partons that are almost collinear to their parent hadron. Every hadron contributes exactly one parton. Multiple contributions are power suppressed by the mass of the intermediate boson Q^{-2}. Historically this idea is called the *parton model*, where every parton carries a momentum fraction of the hadron. It was proposed by Feynman in 1969 [56]. Motivated by Feynman's framework Bjorken and Paschos applied the parton model to deep inelastic scattering (DIS) [57]. Subsequently it was adapted also to other processes. More generally we can identify various directions that give contributions of different importance to describe the physical process. This observation is driven by the ordering

$$p_1 \cdot P_1 \ll p_2 \cdot P_1 \tag{1.49a}$$

$$p_2 \cdot P_2 \ll p_1 \cdot P_2 \tag{1.49b}$$

which allows us to approximate the parton momenta in the hard scattering and in the overall delta distribution such that the small terms are neglected. Breaking down also the phase space integrations via

$$\int \mathrm{d}^4 p_1 = \frac{2}{S} \int \mathrm{d}(p_1 \cdot P_1) \int \mathrm{d}(p_1 \cdot P_2) \int \mathrm{d}^2 p_{1,T} \tag{1.50}$$

where $S = (P_1 + P_2)^2$ is the hadronic center of mass energy and we neglect the hadron masses, we can then observe that only the correlators depend on the otherwise ignored components. For the following discussion let us assume that the hadron has a momentum that can be parametrized by the normalized light-cone vector n^μ_+ by $P^\mu \sim n^\mu_+$, which will simplify the notation greatly. This is possible because we treat not only the quarks but also the hadrons as massless $P^2 = 0$. Further we introduce

the conjugated light-cone vector n_-^μ with $n_+ \cdot n_- = 1$. Then the components of a four vector a^μ can be decomposed as

$$a^\mu = (a \cdot n_-)n_+^\mu + (a \cdot n_+)n_-^\mu + a_T^\mu = a_+ n_+^\mu + a_- n_-^\mu + a_T^\mu \tag{1.51}$$

where the transverse component a_T^μ is defined to be orthogonal to the light-cone components. The smallest components that are neglected first are the parton directions, which are opposite to their parent hadrons momentum, id est the $p \cdot P \sim p \cdot n_+$ direction. This leads to the *transverse momentum dependent (TMD) correlator functions*. For example the transverse momentum dependent quark PDF correlator is given by

$$\begin{aligned}
\Phi_{ij}^q(p_+, \vec{p}_T, P) &\equiv \int \mathrm{d}(p \cdot n_+) \Phi_{ij}^q(p, P) \\
&= \int \mathrm{d}p_- \int \frac{\mathrm{d}^4\eta}{(2\pi)^4} \, \mathrm{e}^{\mathrm{i}(p_+\eta_- + p_-\eta_+ - \vec{p}_T \cdot \vec{\eta}_T)} \, \langle P | \bar{\psi}_j(0) \psi_i(\eta_+, \eta_-, \vec{\eta}_T) | P \rangle \\
&= \int \frac{\mathrm{d}\eta_- \mathrm{d}^2\eta_T}{(2\pi)^3} \, \mathrm{e}^{\mathrm{i}(p_+\eta_- - \vec{p}_T \cdot \vec{\eta}_T)} \, \langle P | \bar{\psi}_j(0) \psi_i(0, \eta_-, \vec{\eta}_T) | P \rangle
\end{aligned} \tag{1.52}$$

TMDs describe the partons inside hadrons via their longitudinal momentum fraction as well as transverse momenta. They make it possible to create a three-dimensional image of the hadrons, which causes the large popularity and activity of the field over the last decades [58–70]. Especially for spin dependent observables they introduce a rich phenomenological playground. The Sivers asymmetry alone has attracted an enormous amount of efforts [71–80]. On the other side it is very difficult to conduct TMD studies, as a lot of free parameters have to be determined, which entails the risk of simply fitting data. In this thesis we will not use TMDs explicitly, instead we will use the more common framework in which only the longitudinal momentum fraction is taken into account. By neglecting also the transverse momenta in the hard scattering and the overall delta distribution, only the correlators depend on them. As before we can integrate them out to obtain the *collinear correlator functions*. We write down again only the collinear quark PDF correlator

$$\begin{aligned}
\Phi_{ij}^q(\xi, P) &\equiv \int \mathrm{d}^2 p_T \Phi_{ij}^q(p_+, \vec{p}_T, P) \\
&= \int \frac{\mathrm{d}\eta_-}{2\pi} \, \mathrm{e}^{\mathrm{i}p_+\eta_-} \, \langle P | \bar{\psi}_j(0) \psi_i(0, \eta_-, \vec{0}_T) | P \rangle \\
&= \int \frac{\mathrm{d}\eta}{2\pi} \, \mathrm{e}^{\mathrm{i}\xi P_+ \eta} \, \langle P | \bar{\psi}_j(0) \psi_i(0, \eta, \vec{0}_T) | P \rangle
\end{aligned} \tag{1.53}$$

Note that we describe the longitudinal momentum no longer by the component p_+ but by its longitudinal momentum fraction ξ defined by $p_+ = \xi P_+$. This links us again to the original parton model (see above).

For a complete description of the correlators we should include also additional gluon exchanges between the hadronic matrix elements and the incoming particle legs. Because

these do not necessarily carry a large momentum, it is not possible to neglect them being of higher order in the coupling α_s. It has been shown that the additional terms exponentiate [58, 60, 61, 75]. The result are path ordered exponents called Wilson lines which connect two points in space time. For example the collinear quark PDF correlator reads

$$\Phi^q_{ij}(\xi, P) = \int \frac{\mathrm{d}\eta}{2\pi}\, \mathrm{e}^{\mathrm{i}\xi P_+\eta}\, \langle P|\bar{\psi}_j(0)\mathcal{W}\psi_i(0,\eta,\vec{0}_T)|P\rangle \tag{1.54}$$

with the Wilson line

$$\mathcal{W} \equiv \mathcal{P}\exp\left[-\mathrm{i}g_s t^a \int_0^\eta \mathrm{d}\alpha\, G^a_+(0,\alpha,\vec{0}_T)\right] \tag{1.55}$$

and $\mathcal{P}$ being the path ordering operator. Due to the Wilson lines the definitions of the correlators become gauge independent. However, in light-cone gauge ($G^a_+ = 0$) the line reduces to a unit matrix. This means that equation (1.53) is still the correct definition in light-cone gauge. For more informations on the definitions of Wilson lines also in the TMD formalism, see [55] and references therein. To see a distinct example including a partly different set of correlators see chapter 4 and the corresponding appendix D, where we carefully calculate the semi inclusive DIS (SIDIS) cross section at small transverse momentum, including similar steps that have been presented here so far, but being way more explicit in order to get all factors correct.

For unpolarized cross sections without considering hadronic masses the collinear correlator can be expanded very simply by incorporating the properties of the correlator under different transformations, for example parity or time reversal,

$$\Phi^q(\xi, P) = f_q(\xi)\not{p} \tag{1.56}$$

For polarized states or considering hadron masses, additional terms would be present on the right-hand side.[9] The function $f_q(\xi)$ is called a *parton distribution function* (PDF). In the original parton model it is a probability density for finding the parton q with longitudinal momentum fraction ξ inside the considered hadron. A formal definition can be achieved by taking the trace of equation (1.56) multiplied by $\not{n}_-$. The multiplication projects out the PDF even if we had written down the full expansion of the correlator respecting polarized spin states or the hadron mass. With $\mathrm{tr}(\not{p}\not{n}_-) = 4p\cdot n_- = 4p_+$ we find for the definition of the PDF [81]

$$\begin{aligned} f_q(\xi) &= \frac{1}{4\xi P_+}\,\mathrm{tr}(\Phi^q(\xi,P)\not{n}_-) \\ &= \frac{1}{4\xi P_+}\int \frac{\mathrm{d}\eta}{2\pi}\, \mathrm{e}^{\mathrm{i}\xi P_+\eta}\, \langle P|\bar{\psi}(0)\gamma_+\psi(0,\eta,\vec{0}_T)|P\rangle \end{aligned} \tag{1.57}$$

[9] For TMDs the expansion consists of a larger number of independent functions, describing individual properties of the hadron. Some of these vanish if integrated over transverse momentum, but have important physical meanings. To examine the nature of hadrons they are therefore indispensable.

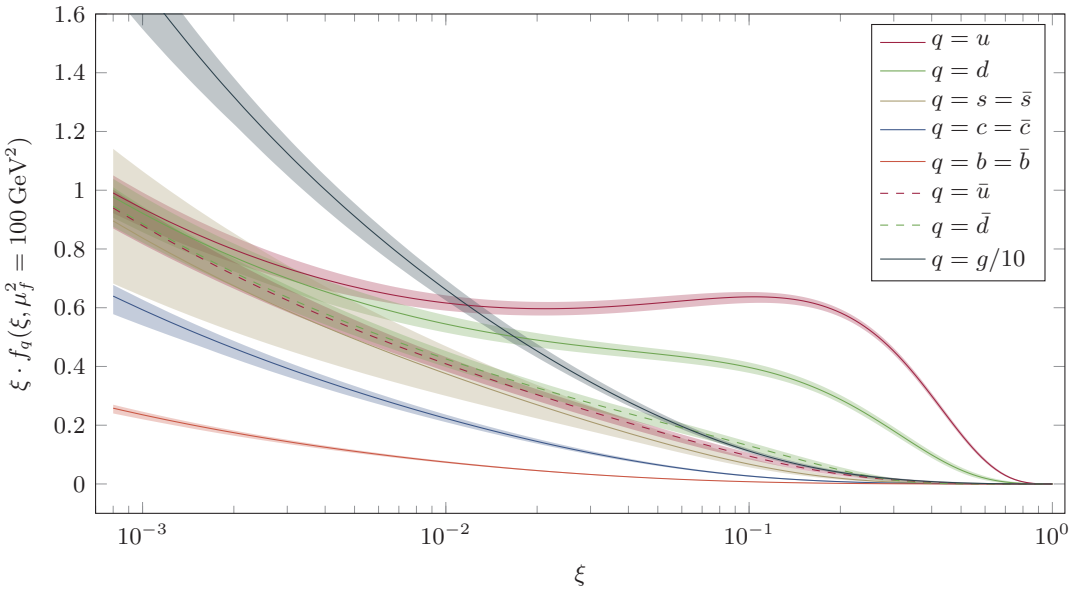


Figure 1.4: Next-to-leading-order proton parton distribution functions fitted by the CTEQ collaboration [82] (CT10) at squared scale $100\,\mathrm{GeV}^2$. The error bands describe the overall fitting error of the 26 parameters used for every quark flavour. The gluon distribution is divided by ten to accomplish a better visualization.

Naturally also for the definition of the PDF a Wilson line has to be taken into account, similarly to the correlator itself. And alike we can deduce a definition of a TMD PDF or the fully unintegrated PDF by applying the same steps to the expression we had before integrating out the transverse and minus components of the quark momentum.

As described before, factorization relies on the separation of short- and long-distance physics. We have seen this already in equation (1.47) where the calculable short-distance physics is described by the hard scattering function H while the long-distance physics has been entailed in the correlators Φ^q and $\Phi^{\bar{q}}$. Until now we have only considered leading order in the coupling α_s. Factorization states that for arbitrary order we can separate the perturbative from the nonperturbative part. For the Drell-Yan process the cross section can be written as [83]

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\eta \mathrm{d}Q^2} = \sum_{a,b\in\mathcal{P}} \int_{x_1}^{1} \mathrm{d}\xi_1 \int_{x_2}^{1} \mathrm{d}\xi_2\, f_a(\xi_1, \mu_f^2) H_{ab}\left(\frac{x_1}{\xi_1}, \frac{x_2}{\xi_2}, Q^2, \frac{\mu_f^2}{Q^2}, \alpha_s\right) f_b(\xi_2, \mu_f^2) \tag{1.58}$$

To obtain this formula from the ones we derived before, we have to specify the phase space of the two leptons $\mathrm{d}\mathcal{R}$, integrating out all degrees of freedom except for the invariant mass of the lepton pair $Q^2 = q^2$ and its rapidity η defined by

$$\eta \equiv \frac{1}{2} \ln\left(\frac{P_1 \cdot q}{P_2 \cdot q}\right) \tag{1.59}$$

Furthermore we have introduced the variables

$$x_{1,2} \equiv \sqrt{\frac{Q^2}{S}}\, \mathrm{e}^{\pm\eta} \tag{1.60}$$

and have to sum over all parton species

$$\mathcal{P} \equiv \mathcal{Q} \cup \overline{\mathcal{Q}} \cup \{g\} \tag{1.61}$$

where $\mathcal{Q}$ contains all quark flavours and is defined in equation (1.3). The bar indicates anti-quarks and the label g stands for the gluon. These sums generalize the leading-order sum in equation (1.47), as in leading order only a $q\bar{q}$ pair can form the partonic initial state, while at higher orders arbitrary combinations are possible. Moreover we have to redefine the hard scattering function H_{ab} to entail the whole trace including all but the PDFs (after replacing the correlators by them) and to gather all additional occurring factors. Finally we introduce a scale μ_f, which is used to renormalize the calculations necessary for H_{ab}. It is called the *factorization scale* and should not be confused with the renormalization scale μ_r we introduced in the previous section, although they are often chosen to be equal. Equation (1.58) is often called the *factorization formula* because it now explicitly contains the ingredients we have talked about before. The partonic scattering is buried in H_{ab} and can be calculated perturbatively. The PDFs have to be measured by experiments, as they are not accessible by perturbative QCD.[10] This may seem as a huge loss of predictive power to pQCD, but because the PDFs are universal, id est they do not depend on the underlying process, it is possible to fit them to a clearly defined set of data and retain predictive power for all other data. Nowadays several collaborations offer PDF sets, the most popular are the PDFs of the CTEQ group [82, 86–88], MSTW or MMHT [89, 90] and of the NNPDF collaboration [91, 92]. All of them consider the proton as the parent hadron. As an example we show the CTEQ set [82] (CT10) in figure 1.4. In the large ξ region around one third the u and d quark distributions dominate, nicely fitting the naive parton model, in which the proton consists of two u quarks and one d quark, which should carry approximately one third of the protons momentum each and compose its quantum numbers. These distributions are called *valence distributions*. Due to gluon interactions they smear out and the generation and annihilation of temporary *sea quarks* has to be added, which contribute mostly for small momentum fractions. This introduces also quark flavours being part of the proton that are not present in the naive parton model.

The distributions have to fulfill certain constraints, which are called *sum rules*. For example they have to reproduce the proton quantum numbers, as mentioned before,

$$\int_0^1 \mathrm{d}\xi \left[f_u(\xi, \mu_f^2) - f_{\bar{u}}(\xi, \mu_f^2) \right] = 2 \tag{1.62a}$$

[10] In the field of lattice QCD much effort has been expended to calculate PDFs directly from the QCD Lagrangian [84, 85]. Nevertheless, current high-precision predictions still rely on the measured PDFs.

$$\int_0^1 \mathrm{d}\xi \left[f_d(\xi, \mu_f^2) - f_{\bar{d}}(\xi, \mu_f^2) \right] = 1 \tag{1.62b}$$

$$\int_0^1 \mathrm{d}\xi \left[f_q(\xi, \mu_f^2) - f_{\bar{q}}(\xi, \mu_f^2) \right] = 0 \qquad \forall q \in \mathcal{Q} \setminus \{u, d\} \tag{1.62c}$$

Further all parton momenta have to add up to the proton momentum

$$\sum_{q \in \mathcal{P}} \int_0^1 \mathrm{d}\xi \, \xi f_q(\xi, \mu_f^2) = 1 \tag{1.63}$$

These already imply for example the isospin, strangeness and the charge of the proton, which are linear combinations of the equations (1.62). As these quantities can be measured directly, they offer a direct access to the PDFs. For example the isospin is measured as a difference of the structure function F_2 in neutrino scattering off protons and neutrons. The corresponding formula is known as the Adler sum rule [93]. Another example relates the baryon number and the strangeness of a nucleon (again measured via neutrino scattering) to the PDFs [94]

$$\sum_{q \in \mathcal{P}} \int_0^1 \mathrm{d}\xi \left[f_q(\xi, \mu_f^2) - f_{\bar{q}}(\xi, \mu_f^2) \right] = 3 \tag{1.64}$$

which is called the Gross-Llewellyn Smith sum rule. An introduction to sum rules can be found in [95]. The PDFs shown in figure 1.4 assume that quark flavours that only appear in the sea have identical quark and anti-quark distributions. Note that this is not necessarily true, as long as the sum rules are respected. Finally we want to point out the very large error band of the s quark, which is a peculiarity of the CT10 fit [82]. The authors point out that they released several formal constraints on the s distribution. The large error band emphasizes that the experimental data taken into account were not sufficient to determine all parameters of the s distribution equally well as for the other ones. More details about common PDF parameterizations is provided in chapter 5 of this thesis, which treats PDFs transformed in Mellin space.

To prove factorization it is essential to show that the factorization formula (1.58) holds to all orders. The proofs are process dependent and we can distinguish between formal proofs to all orders, as for the Drell-Yan process, DIS or semi inclusive DIS [83, 96], while it is verified for other processes by direct calculation at fixed orders, see for example [97, 98]. As a rule of thumb one can assume factorization to be proven the more inclusive the underlying process is. For TMDs a formal proof—again for Drell-Yan and (semi inclusive) DIS—has been given in this decade [62, 63].

We will now demonstrate how to apply the factorization theorem to calculate the hard scattering part H_{ab} in equation (1.58). Because of factorization the partonic

scattering is insensitive to the parent hadrons of the participating partons. This means that we can substitute the hadron, which prevents a completely perturbative calculation in the first place, by another parton. We call the resulting cross section G_{ab}, where a and b are the parton flavours replacing the hadrons

$$G_{ab} = \sum_{c,d\in\mathcal{P}} \int_{x_1}^{1} \mathrm{d}\xi_1 \int_{x_2}^{1} \mathrm{d}\xi_2 \, f_{c/a}(\xi_1, \mu_f^2) H_{cd}\left(\frac{x_1}{\xi_1}, \frac{x_2}{\xi_2}, Q^2, \frac{\mu_f^2}{Q^2}, \alpha_s\right) f_{d/b}(\xi_2, \mu_f^2) \tag{1.65}$$

The PDFs have been replaced by their equivalents containing simple quark states instead of a hadron state, see equation (1.57). These quark states can be written as creation and annihilation operators acting on the vacuum state. Using the definition of a free fermionic field it is possible to calculate the $f_{a/b}$ order by order. For a detailed calculation see [99]. A good summary is provided in appendix B of [50]. The calculation reveals UV as well as IR divergences, which are made manifest by dimensional regularization similar to the renormalization procedure we described in section 1.2. The UV pole is subtracted by a counterterm in the $\overline{\mathrm{MS}}$ scheme, while the IR pole remains. Up to NLO the quark-in-quark PDFs are given by [81]

$$f_{a/b}(\xi) = \delta_{ab}\delta(1-\xi) - \frac{1}{2\varepsilon}\frac{\alpha_s}{\pi} P_{ab}^{(0)}(\xi) + \mathcal{O}(\alpha_s^2) \tag{1.66}$$

where the $P_{ab}^{(0)}(\xi)$ is the leading-order Altarelli-Parisi splitting function defined in equation (1.74). We will discuss its meaning in the next section 1.5. Now we can expand also the cross section G_{ab} and the hard scattering functions in the strong coupling constant

$$G_{ab} = \sum_{n=0}^{\infty} \left(\frac{\alpha_s}{\pi}\right)^n G_{ab}^{(n)} \tag{1.67a}$$

$$H_{ab} = \sum_{n=0}^{\infty} \left(\frac{\alpha_s}{\pi}\right)^n H_{ab}^{(n)} \tag{1.67b}$$

We can calculate the coefficients $G_{ab}^{(n)}$ in perturbation theory. Except for the leading order $n = 0$ we find again UV and IR divergences. The UV divergences are removed as usual in the $\overline{\mathrm{MS}}$ scheme. The IR divergences can be separated into two classes. The first are *soft divergences* which originate from configurations where a parton is emitted with vanishing momentum. They cancel among virtual and real diagrams. The second class are *collinear divergences* which originate from configurations where an emitted parton has a momentum collinear to its parent parton. These have to be eliminated to achieve a finite result. Effectively this is done by shifting them into the PDFs. If and only if these cancel the pole in the PDFs, see equation (1.66), the cross section factorizes with finite expressions as well for the PDFs as for the hard scattering function. We want to note once more that the IR divergences are—at least in this case—a relict

of the massless treatment of the quarks. To be more specific we expand both sides of equation (1.65) up to NLO using (1.66) and (1.67)

$$G_{ab}^{(0)} + \frac{\alpha_s}{\pi} G_{ab}^{(1)} = H_{ab}^{(0)} + \frac{\alpha_s}{\pi} H_{ab}^{(1)} - \frac{1}{2\varepsilon} \frac{\alpha_s}{\pi} \left[\sum_{c \in \mathcal{P}} \int_{x_1}^{1} \mathrm{d}\xi_1 \, P_{ca}^{(0)}(\xi_1) H_{cb}^{(0)} + \sum_{d \in \mathcal{P}} \int_{x_2}^{1} \mathrm{d}\xi_2 \, H_{ad}^{(0)} P_{db}^{(0)}(\xi_2) \right] \tag{1.68}$$

The arguments of G_{ab} and H_{ab} are suppressed for brevity. Now the expression for the hard scattering functions can be identified order by order

$$H_{ab}^{(0)} = G_{ab}^{(0)} \tag{1.69}$$

$$H_{ab}^{(1)} = G_{ab}^{(1)} + \frac{1}{2\varepsilon} \left[\sum_{c \in \mathcal{P}} \int_{x_1}^{1} \mathrm{d}\xi_1 \, P_{ca}^{(0)}(\xi_1) G_{cb}^{(0)} + \sum_{d \in \mathcal{P}} \int_{x_2}^{1} \mathrm{d}\xi_2 \, G_{ad}^{(0)} P_{db}^{(0)}(\xi_2) \right] \tag{1.70}$$

They are finite for $\varepsilon \to 0$ as the poles originating from the PDFs cancel exactly the IR poles in G_{ab}. We can now use the H_{ab} to calculate physical cross sections including hadronic states by inserting the expressions into the factorization formula (1.58) and using measured PDFs describing the nonperturbative physics.

1.5 DGLAP evolution

In the previous section we saw that the factorization introduces an arbitrary scale μ_f, the factorization scale, similar to the renormalization procedure introducing the renormalization scale μ_r. Due to the factorization the PDFs as well as the hard scattering function depend on μ_f, which we also denoted for example in figure 1.4 without mentioning it explicitly in the discussion. In this section we will discuss the corresponding RGE and its consequences. Because we implemented the results in the C++ library fi*n*ael, which is presented in appendix C, we will give a rather technical recapitulation of the topic, mainly following the more general discussion in [100]. To clarify the notation remind the definitions given in equations (1.3) and (1.61). Furthermore we define a function that maps integer numbers to $\mathcal{P}$ (see equation (1.61))

$$f_{\mathrm{idx}} : \{-6, \ldots, 6\} \mapsto \mathcal{P} \tag{1.71}$$

in the natural way ($f_{\mathrm{idx}}(-6) = \bar{t}, \ldots, f_{\mathrm{idx}}(-1) = \bar{u}, f_{\mathrm{idx}}(0) = g, f_{\mathrm{idx}}(1) = u, \ldots$). Finally we introduce the abbreviation $p \equiv f_p$ for a PDF of the flavour p.

It is obvious that physical observables O should not depend on the factorization scale as they should not depend on the renormalization scale

$$\mu_f^2 \frac{\mathrm{d}}{\mathrm{d}\mu_f^2} O = 0 \tag{1.72}$$

One can construct various observables, depending on different parton flavours. By choosing smart linear combinations it is possible to extract the evolution equations for all partons of the form ($p \in \mathcal{P}$)[11]

$$\begin{aligned}\mu_f^2 \frac{\partial}{\partial \mu_f^2} p(x, \mu_f^2) &= \sum_{\tilde{p} \in \mathcal{P}} \int_x^1 \frac{\mathrm{d}\xi}{\xi} \mathcal{P}_{p\tilde{p}}\left(\frac{x}{\xi}, \alpha_s(\mu_f^2)\right) \tilde{p}(\xi, \mu_f^2) \\ &= \sum_{\tilde{p} \in \mathcal{P}} \Big[\mathcal{P}_{p\tilde{p}}(\alpha_s(\mu_f^2)) \otimes \tilde{p}(\mu_f^2)\Big](x) \end{aligned} \tag{1.73}$$

These $2n_f + 1$ coupled differential equations are called the DGLAP-equations, because they have been promoted by Dokshitzer, Gribov and Lipatov, and Altarelli and Parisi [101–103]. The $\mathcal{P}_{p\tilde{p}}$ are the Altarelli-Parisi evolution kernels, which have a perturbative expansion

$$\mathcal{P}_{p\tilde{p}}\big(z, \alpha_s(\mu_f^2)\big) \equiv \sum_{n=0}^{\infty} \left(\frac{\alpha_s(\mu_f^2)}{4\pi}\right)^{n+1} P_{p\tilde{p}}^{(n)}(z) \tag{1.74}$$

At leading order these have the very descriptive meaning of the probability for a parton $\tilde{p}$ to become a parton p with longitudinal momentum fraction z by emitting another parton. Hence they are also known as *splitting functions*. The coefficients are well known up to NNLO [102, 104–121]. They are presented in appendix B up to NLO. We already met the leading-order coefficient in the previous section as part of the PDF corrections, see equation (1.66). This is consistent, as these introduce the scale dependence in the first place and should be driven by the same functions.

In this section we choose the factorization scale to be equal to the renormalization scale $\mu_f = \mu_r$, as we already did in the equations above. It is a very common choice. In a fixed-order calculation one can use the Taylor expansion of $\alpha_s(\mu_f^2)$ in terms of $\alpha_s(\mu_r^2)$, if the two scales differ to retain a description using both independently. For example at NLO this is given by

$$\begin{aligned}\mathcal{P}_{p\tilde{p}}\big(z, \mu_f^2, \alpha_s(\mu_r^2)\big) = &\frac{\alpha_s(\mu_r^2)}{4\pi} P_{p\tilde{p}}^{(0)}(z) \\ &+ \left(\frac{\alpha_s(\mu_r^2)}{4\pi}\right)^2 \left[P_{p\tilde{p}}^{(1)}(z) - \beta_0 \ln\left(\frac{\mu_f^2}{\mu_r^2}\right) P_{p\tilde{p}}^{(0)}(z)\right] + \mathcal{O}\big(\alpha_s^3(\mu_r^2)\big) \end{aligned} \tag{1.75}$$

We aim now at solving the evolution equations (1.73) up to NLO. The analytical solution is expressed most easily in Mellin space (see appendix E), where the convolutions

[11] It is important to notice that it is impossible to extract all evolution equations from one observable. For instance the DIS process can only distinguish between the gluon and the sum of all quarks flavours (which is called the *singlet distribution* and will be defined soon). The process cannot identify different quark flavours.

turn into simple products. Defining the moments of the parton distributions and splitting functions ($p, \tilde{p} \in \mathcal{P}$)

$$p(N, \mu_f^2) \equiv \int_0^1 \mathrm{d}x\, x^{N-1} p(x, \mu_f^2) \tag{1.76a}$$

$$\gamma_{p\tilde{p}}(N, \mu_f^2) \equiv \int_0^1 \mathrm{d}z\, z^{N-1} \mathcal{P}_{p\tilde{p}}\big(z, \alpha_s(\mu_f^2)\big) \tag{1.76b}$$

the DGLAP-equations (1.73) turn into

$$\mu_f^2 \frac{\partial}{\partial \mu_f^2} p(N, \mu_f^2) = \sum_{\tilde{p} \in \mathcal{P}} \gamma_{p\tilde{p}}(N, \mu_f^2) \tilde{p}(N, \mu_f^2) \tag{1.77}$$

From now on we will suppress the arguments for brevity if not necessary for comprehension. To solve the $2n_f + 1$ coupled differential equations, it is advantageous to decouple them to the highest possible extent. For this purpose, we recognize that the coupling to the gluon is flavour independent. Consequently the quark-gluon and the gluon-quark splitting functions are also flavour independent. Hence we define the splitting functions

$$\gamma_{gq} \equiv \gamma_{ga} \tag{1.78a}$$

$$\gamma_{qg} \equiv 2n_f \gamma_{ag} \tag{1.78b}$$

with $a \in \mathcal{Q} \cup \overline{\mathcal{Q}}$. Furthermore it is convenient to use the general structure of the (anti-)quark splitting functions. These can be divided into a flavour diagonal and a non-diagonal part ($a, e \in \mathcal{Q}$ and $\bar{a}, \bar{e} \in \overline{\mathcal{Q}}$)

$$\gamma_{ae} = \delta_{ae} \gamma_{qq}^{\mathrm{v}} + \gamma_{qq}^{\mathrm{s}} \tag{1.79a}$$

$$\gamma_{\bar{a}\bar{e}} = \delta_{\bar{a}\bar{e}} \gamma_{qq}^{\mathrm{v}} + \gamma_{qq}^{\mathrm{s}} \tag{1.79b}$$

The flavour diagonal is also called the 'valence' part and the non-diagonal the 'sea' part, which is indicated by the superscripts v and s. The same can be done for the quark-antiquark splitting functions

$$\gamma_{a\bar{e}} = \delta_{ae} \gamma_{q\bar{q}}^{\mathrm{v}} + \gamma_{q\bar{q}}^{\mathrm{s}} \tag{1.80a}$$

$$\gamma_{\bar{a}e} = \delta_{ae} \gamma_{q\bar{q}}^{\mathrm{v}} + \gamma_{q\bar{q}}^{\mathrm{s}} \tag{1.80b}$$

Including the gluon-gluon splitting function γ_{gg} one has at leading order four independent splitting functions, because at leading order it is not possible to change the quark flavour $\gamma_{qq}^{\mathrm{s}} = \gamma_{q\bar{q}}^{\mathrm{s}} = \gamma_{q\bar{q}}^{\mathrm{v}} = 0$. At NLO one has six independent splitting functions, because at that order $\gamma_{qq}^{\mathrm{s}} = \gamma_{q\bar{q}}^{\mathrm{s}}$. First at NNLO all seven splitting functions are required.

Since the coupling to the gluon does not depend on the flavour or whether it is a quark or an antiquark, *every* difference of two quark distributions does not couple to the gluon. We can therefore define the so-called *nonsinglet* combinations, denoted with the subscript ns

$$q^{\pm}_{\mathrm{ns},ae} = (a \pm \bar{a}) - (e \pm \bar{e}) \tag{1.81a}$$

$$q^{\mathrm{v}}_{\mathrm{ns}} = \sum_{p\in\mathcal{Q}} (p - \bar{p}) \tag{1.81b}$$

where again $a, e \in \mathcal{Q}$. It is easy to prove that

$$\begin{aligned}\mu_f^2 \frac{\partial}{\partial \mu_f^2}(a \pm \bar{a}) = (\gamma^{\mathrm{v}}_{qq} \pm \gamma^{\mathrm{v}}_{q\bar{q}})a + (\gamma^{\mathrm{v}}_{q\bar{q}} \pm \gamma^{\mathrm{v}}_{qq})\bar{a} + \frac{1}{2n_f}(\gamma_{qg} \pm \gamma_{qg})g \\ + \sum_{p\in\mathcal{Q}} (\gamma^{\mathrm{s}}_{qq} \pm \gamma^{\mathrm{s}}_{q\bar{q}})p + \sum_{p\in\overline{\mathcal{Q}}} (\gamma^{\mathrm{s}}_{q\bar{q}} \pm \gamma^{\mathrm{s}}_{qq})p\end{aligned} \tag{1.82}$$

and therefore the nonsinglet combinations evolve as

$$\mu_f^2 \frac{\partial}{\partial \mu_f^2} q^{\pm}_{\mathrm{ns},ae} = (\gamma^{\mathrm{v}}_{qq} \pm \gamma^{\mathrm{v}}_{q\bar{q}}) q^{\pm}_{\mathrm{ns},ae} \equiv \gamma^{\pm}_{\mathrm{ns}} q^{\pm}_{\mathrm{ns},ae} \tag{1.83a}$$

$$\mu_f^2 \frac{\partial}{\partial \mu_f^2} q^{\mathrm{v}}_{\mathrm{ns}} = \left[\gamma^{\mathrm{v}}_{qq} - \gamma^{\mathrm{v}}_{q\bar{q}} + n_f(\gamma^{\mathrm{s}}_{qq} - \gamma^{\mathrm{s}}_{q\bar{q}})\right] q^{\mathrm{v}}_{\mathrm{ns}} \equiv \left[\gamma^{-}_{\mathrm{ns}} + \gamma^{\mathrm{s}}_{\mathrm{ns}}\right] q^{\mathrm{v}}_{\mathrm{ns}} \tag{1.83b}$$

To perform the evolution at higher orders than NLO it is advantageous to introduce $2n_f - 2$ linearly independent combinations of the $q^{\pm}_{\mathrm{ns},ae}$

$$q^{\pm}_{\mathrm{ns},n} \equiv -n(f_{\mathrm{idx}}(n) \pm f_{\mathrm{idx}}(-n)) + \sum_{k=1}^{n} (f_{\mathrm{idx}}(k) \pm f_{\mathrm{idx}}(-k)) \tag{1.84}$$

with $(n \in \{2, \ldots, n_f\})$. We do the same already at NLO to be prepared for generalizations to higher orders. Obviously they evolve similarly

$$\mu_f^2 \frac{\partial}{\partial \mu_f^2} q^{\pm}_{\mathrm{ns},n} = \gamma^{\pm}_{\mathrm{ns}} q^{\pm}_{\mathrm{ns},n} \tag{1.85}$$

and in sum we have now $2n_f - 1$ evolution equations in the nonsinglet case.

Since the nonsinglet case is decoupled from the gluon, one chooses the linear combination which couples maximally with the gluon for the so called *singlet distribution*

$$q_{\mathrm{s}} \equiv \sum_{p\in\mathcal{Q}\cup\overline{\mathcal{Q}}} p \tag{1.86}$$

We want to point out that the DIS process is sensitive only to the singlet distribution and the gluon but not to the various nonsinglet distributions. With the given definitions it is easy to show that the singlet and the gluon evolve with

$$\mu_f^2 \frac{\partial}{\partial \mu_f^2} q_{\mathrm{s}} = \left[\gamma^{+}_{\mathrm{ns}} + n_f(\gamma^{\mathrm{s}}_{qq} + \gamma^{\mathrm{s}}_{q\bar{q}})\right] q_{\mathrm{s}} + \gamma_{qg} g$$

$$\equiv \left[\gamma_{\rm ns}^{+} + \gamma_{\rm ps}\right] q_{\rm s} + \gamma_{qg} g \tag{1.87a}$$

$$\equiv \gamma_{qq} q_{\rm s} + \gamma_{qg} g \tag{1.87b}$$

$$\mu_f^2 \frac{\partial}{\partial \mu_f^2} g = \gamma_{gq} q_{\rm s} + \gamma_{gg} g \tag{1.87c}$$

which is often denoted in a vector notation

$$\mu_f^2 \frac{\partial}{\partial \mu_f^2} \begin{pmatrix} q_{\rm s} \\ g \end{pmatrix} = \begin{pmatrix} \gamma_{qq} & \gamma_{qg} \\ \gamma_{gq} & \gamma_{gg} \end{pmatrix} \begin{pmatrix} q_{\rm s} \\ g \end{pmatrix} \tag{1.88}$$

To restore the original quark distributions one uses the relations ($n \in \{1, \ldots, n_f\}$)

$$\frac{1}{n_f} q_{\rm s} - \frac{1}{n} q_{{\rm ns},n}^{+} + \sum_{k=n+1}^{n_f} \frac{1}{k(k-1)} q_{{\rm ns},k}^{+} = f_{\rm idx}(n) + f_{\rm idx}(-n) \tag{1.89a}$$

$$\frac{1}{n_f} q_{\rm ns}^{\rm v} - \frac{1}{n} q_{{\rm ns},n}^{-} + \sum_{k=n+1}^{n_f} \frac{1}{k(k-1)} q_{{\rm ns},k}^{-} = f_{\rm idx}(n) - f_{\rm idx}(-n) \tag{1.89b}$$

with $q_{{\rm ns},1}^{\pm} \equiv 0$. The left-hand sides can be calculated by using the definitions of the singlet and nonsinglet distributions and indexing the double sum as $n \in \{1, \ldots, n_f\}$

$$\begin{aligned} \sum_{k=n+1}^{n_f} \frac{1}{k(k-1)} \sum_{j=1}^{k} f(j) &= \sum_{j=1}^{n} f(j) \sum_{k=n+1}^{n_f} \frac{1}{k(k-1)} + \sum_{j=n+1}^{n_f} f(j) \sum_{k=j}^{n_f} \frac{1}{k(k-1)} \\ &= \frac{n_f - n}{n_f n} \sum_{j=1}^{n} f(j) + \sum_{j=n+1}^{n_f} f(j) \frac{1 + n_f - j}{n_f (j-1)} \end{aligned} \tag{1.90}$$

which is true for arbitrary functions f that are not singular for integer argument. This is clearly true for all functions we are interested in. The last step of the calculation uses the finite sum

$$\sum_{i=2}^{m} \frac{1}{i(i-1)} = \frac{m-1}{m} \tag{1.91}$$

which can be shown easily by mathematical induction.

We have now finally managed to write down all evolution equations in the form

$$\mu_f^2 \frac{\partial}{\partial \mu_f^2} Q = \Gamma Q \tag{1.92}$$

where Γ and Q are one of the linear combinations of splitting functions and parton distributions in Mellin space respectively. They are scalar quantities for the nonsinglet combinations and a matrix and a vector in the singlet case. As independent splitting functions it is common to choose $\gamma_{\rm ns}^{+}$, $\gamma_{\rm ns}^{-}$ and $\gamma_{\rm ns}^{\rm s}$ in the nonsinglet sector and $\gamma_{\rm ps}$, γ_{qg}, γ_{gq} and γ_{gg} in the singlet sector to solve the evolution equations and perform the

evolution. As already mentioned at leading order the functions γ_{qq}^{s}, $\gamma_{q\bar{q}}^{\mathrm{s}}$ and $\gamma_{q\bar{q}}^{\mathrm{v}}$ all vanish. For our set this translates into $\gamma_{\mathrm{ns}}^{+} = \gamma_{\mathrm{ns}}^{-}$ and $\gamma_{\mathrm{ns}}^{\mathrm{s}} = \gamma_{\mathrm{ps}} = 0$. In NLO it holds $\gamma_{qq}^{\mathrm{s}} = \gamma_{q\bar{q}}^{\mathrm{s}}$ which means that still $\gamma_{\mathrm{ns}}^{\mathrm{s}} = 0$.

For the evolution it is convenient to evolve directly in α_s instead of μ_f^2, which is possible because the coupling is bijective and as long as the factorization and the renormalization scale are linearly dependent. The corresponding change of variables is given by the β-function (see equation (1.20))

$$\mu_f^2 \frac{\partial}{\partial \mu_f^2} Q = \mu_f^2 \frac{\partial \alpha_s}{\partial \mu_f^2} \frac{\partial Q}{\partial \alpha_s} = \beta(\alpha_s) \frac{\partial Q}{\partial \alpha_s} \tag{1.93}$$

Expanding the β-function (1.26) as well as the splitting functions we find [122, 123]

$$\begin{aligned}
\frac{\partial Q}{\partial \alpha_s} &= \frac{1}{\beta(\alpha_s)} \Gamma Q \\
&= -\frac{1}{\beta_0 \alpha_s} \left[1 + \sum_{k=1}^{\infty} \frac{\beta_k}{\beta_0} \left(\frac{\alpha_s}{4\pi} \right)^k \right]^{-1} \left[\sum_{n=0}^{\infty} \Gamma^{(n)} \left(\frac{\alpha_s}{4\pi} \right)^n \right] Q \\
&= -\frac{1}{\beta_0 \alpha_s} \left[\Gamma^{(0)} + \frac{\alpha_s}{4\pi} \left(\Gamma^{(1)} - \frac{\beta_1}{\beta_0} \Gamma^{(0)} \right) + \mathcal{O}(\alpha_s^2) \right] Q \\
&= -\frac{1}{\alpha_s} \left[R_0 + \sum_{k=1}^{\infty} \left(\frac{\alpha_s}{4\pi} \right)^k R_k \right] Q
\end{aligned} \tag{1.94}$$

with the recursively defined coefficients

$$R_k \equiv \begin{cases} \dfrac{1}{\beta_0} \Gamma^{(0)} & \text{if } k = 0 \\ \dfrac{1}{\beta_0} \Gamma^{(k)} - \displaystyle\sum_{n=1}^{k} \frac{\beta_n}{\beta_0} R_{k-n} & \text{if } k > 0 \end{cases} \tag{1.95}$$

Because in the singlet case the matrices R_k do not commute, it is not possible to diagonalize them simultaneously and as a consequence to write a solution in closed exponential form beyond leading order. To stay consistent we will use the same technique for the singlet and nonsinglet case, although the latter has a closed exponential solution we will obtain from the singlet result.

The leading-order solution is given in both cases by

$$Q(\alpha_s(\mu^2)) = \left[\sum_{k=0}^{\infty} \frac{1}{k!} \left(-R_0 \ln \left(\frac{\alpha_s(\mu^2)}{\alpha_s(\mu_0^2)} \right) \right)^k \right] Q(\alpha_s(\mu_0^2)) \equiv \left(\frac{\alpha_s(\mu^2)}{\alpha_s(\mu_0^2)} \right)^{-R_0} Q(\alpha_s(\mu_0^2)) \tag{1.96}$$

evolving from the scale μ_0 to μ. For higher orders we use the following ansatz, which is an expansion around the leading-order solution

$$Q(\alpha_s(\mu^2)) = U(\alpha_s(\mu^2)) \left(\frac{\alpha_s(\mu^2)}{\alpha_s(\mu_0^2)} \right)^{-R_0} U^{-1}(\alpha_s(\mu_0^2)) Q(\alpha_s(\mu_0^2)) \tag{1.97}$$

where $U(\alpha_s)$ is defined by the perturbative expansion

$$U(\alpha_s) = \mathbb{1} + \sum_{n=1}^{\infty} \left(\frac{\alpha_s}{4\pi}\right)^n U_n \tag{1.98}$$

The inverse factor in the ansatz (1.97) ensures that the evolution reduces to unity for $\mu = \mu_0$. Inserting (1.97) on both sides of (1.94) we find

$$\sum_{n=1}^{\infty} \left(\frac{\alpha_s}{4\pi}\right)^n [U_n, R_0] = \sum_{n=1}^{\infty} \left(\frac{\alpha_s}{4\pi}\right)^n (nU_n + R_n) + \sum_{n,m=1}^{\infty} \left(\frac{\alpha_s}{4\pi}\right)^{n+m} R_n U_m \tag{1.99}$$

which leads order by order to the commutation relations

$$[U_n, R_0] = nU_n + R_n + \sum_{m=1}^{n-1} R_n U_{n-m} \equiv nU_n + \tilde{R}_n \tag{1.100}$$

The equations can be solved iteratively. At next-to-leading order we have to solve

$$[U_1, R_0] = U_1 + R_1 \tag{1.101}$$

For this purpose we write the matrix R_0 in terms of its eigenvalues

$$r_0^{\pm} = \frac{1}{2\beta_0}\left(\gamma_{qq}^{(0)} + \gamma_{gg}^{(0)} \pm \sqrt{\left(\gamma_{qq}^{(0)} - \gamma_{gg}^{(0)}\right)^2 + 4\gamma_{gq}^{(0)}\gamma_{qg}^{(0)}}\right) \tag{1.102}$$

and define the normalized (in a sense of complete $P_0^+ + P_0^- = \mathbb{1}$) projectors ($P_0^{\pm} P_0^{\pm} = P_0^{\pm}$) on the corresponding orthogonal ($P_0^{\pm} P_0^{\mp} = 0$) eigenspaces

$$P_0^{\pm} \equiv \frac{1}{r_0^{\pm} - r_0^{\mp}}\left(R_0 - r_0^{\mp}\mathbb{1}\right) \tag{1.103}$$

to decompose R_0

$$R_0 = r_0^+ P_0^+ + r_0^- P_0^- \tag{1.104}$$

and U_1 and R_1

$$U_1 = (P_0^+ + P_0^-)U_1(P_0^+ + P_0^-) = P_0^+ U_1 P_0^+ + P_0^+ U_1 P_0^- + P_0^- U_1 P_0^+ + P_0^- U_1 P_0^- \tag{1.105a}$$

$$R_1 = (P_0^+ + P_0^-)R_1(P_0^+ + P_0^-) = P_0^+ R_1 P_0^+ + P_0^+ R_1 P_0^- + P_0^- R_1 P_0^+ + P_0^- R_1 P_0^- \tag{1.105b}$$

Inserting equations (1.104) and (1.105) into the commutation relation (1.101) we find

$$(r_0^+ - r_0^-)(P_0^- U_1 P_0^+ - P_0^+ U_1 P_0^-) = U_1 + R_1 \tag{1.106}$$

By multiplying the different combinations of the projectors we get

$$P_0^+ U_1 P_0^+ = -P_0^+ R_1 P_0^+ \tag{1.107a}$$

$$P_0^- U_1 P_0^- = -P_0^- R_1 P_0^- \tag{1.107b}$$

$$P_0^+ U_1 P_0^- = \frac{1}{r_0^- - r_0^+ - 1} P_0^+ R_1 P_0^- \tag{1.107c}$$

$$P_0^- U_1 P_0^+ = \frac{1}{r_0^+ - r_0^- - 1} P_0^- R_1 P_0^+ \tag{1.107d}$$

and finally we can express the matrix U_1 by

$$U_1 = -(P_0^+ R_1 P_0^+ + P_0^- R_1 P_0^-) + \frac{1}{r_0^- - r_0^+ - 1} P_0^+ R_1 P_0^- + \frac{1}{r_0^+ - r_0^- - 1} P_0^- R_1 P_0^+ \tag{1.108}$$

It is possible to generalize the result by using the definition of $\tilde{R}_n$ in equation (1.100) to

$$U_n = -\frac{1}{n}(P_0^+ \tilde{R}_n P_0^+ + P_0^- \tilde{R}_n P_0^-) + \frac{1}{r_0^- - r_0^+ - n} P_0^+ \tilde{R}_n P_0^- + \frac{1}{r_0^+ - r_0^- - n} P_0^- \tilde{R}_n P_0^+ \tag{1.109}$$

Note that the poles ($r_0^\pm(N)$) are canceled by the inverse factor U^{-1} in (1.97). For a fixed order this can be shown explicitly by truncating all present series, but the statement is also true on a general basis [100]. The statement is highly nontrivial because not only the sum in the denominator can be zero but also the splitting functions have poles. Also these have to cancel by the inverse factor U^{-1}.

To write down an analytical solution one has to determine the accuracy or the level of truncation. At a fixed order it is possible to use different expressions, which only differ beyond the considered order. These choices make a difference by the handling of the series in equation (1.94) and (1.97). The most obvious choices are

Full solution: Using all orders (in practice to a sufficiently high order) of the β-function and the splitting functions. Note that the $\tilde{R}_n$ (and therefore the U_n) can be calculated to an arbitrary high order because they are defined recursively (although the $\tilde{R}_n$ will miss the $\Gamma^{(n)}$ for n being larger than the considered order). This solution is equivalent to a direct iterative solution in longitudinal momentum space (x-space) of equation (1.73). For this purpose also the α_s solution has to be exact, which means it has to be calculated iteratively instead of using an approximated analytical solution. In the past a major contribution of the detected difference of Mellin and x-space codes [124] had its origin in the different calculation of α_s [100].

α_s-truncated solution: Removing all terms of order α_s^m in equation (1.94) with $m > n$, where n is the considered order. This removes also the higher-order terms that depend on the β-function and the splitting functions of the considered order. At NLO the solution is given in the next-to-last line of equation (1.94).

Truncated solution: Additionally to the truncation of equation (1.94) also the series in equation (1.98) is truncated to the considered order. This means that the

inverse factor U^{-1} is not calculated as an inverse from the previously determined U but is expanded as a series in α_s. As does already the α_s-truncated solution, the truncated solution solves the evolution equation only in the sense of a power expansion. But in contrast to the α_s-truncated solution it does not introduce scheme-dependent higher-order terms.

To conclude the singlet case we will calculate the *truncated solution* of the evolution at NLO. For this purpose it is useful to rewrite the leading-order solution (1.96) in terms of the projectors by using equation (1.104) and $[P_0^+, P_0^-] = 0$

$$\left(\frac{\alpha_s(\mu^2)}{\alpha_s(\mu_0^2)}\right)^{-R_0} = \left(\frac{\alpha_s(\mu^2)}{\alpha_s(\mu_0^2)}\right)^{-r_0^+} P_0^+ + \left(\frac{\alpha_s(\mu^2)}{\alpha_s(\mu_0^2)}\right)^{-r_0^-} P_0^- \tag{1.110}$$

For the truncated solution we can write

$$\left(\mathbb{1} + \frac{\alpha_s}{4\pi} U_1\right)^{-1} = \mathbb{1} - \frac{\alpha_s}{4\pi} U_1 + \mathcal{O}(\alpha_s^2) \tag{1.111}$$

Inserting everything in the ansatz (1.97) we obtain

$$Q(\alpha_s(\mu^2)) = \left[\left(\frac{\alpha_s(\mu^2)}{\alpha_s(\mu_0^2)}\right)^{-r_0^+} U_{+-} + \left(\frac{\alpha_s(\mu^2)}{\alpha_s(\mu_0^2)}\right)^{-r_0^-} U_{-+}\right] Q(\alpha_s(\mu_0^2)) \tag{1.112}$$

where we defined the matrices

$$\begin{aligned} U_{s_1 s_2} = {} & P_0^{s_1} + \frac{\alpha_s(\mu_0^2) - \alpha_s(\mu^2)}{4\pi} P_0^{s_1} R_1 P_0^{s_1} \\ & + \left[\frac{\alpha_s(\mu^2)}{4\pi}\left(\frac{\alpha_s(\mu^2)}{\alpha_s(\mu_0^2)}\right)^{r_0^{s_1} - r_0^{s_2}} - \frac{\alpha_s(\mu_0^2)}{4\pi}\right] \frac{1}{r_0^{s_2} - r_0^{s_1} - 1} P_0^{s_1} R_1 P_0^{s_2} \end{aligned} \tag{1.113}$$

with $s_i \in \{+, -\}$. Here we can see that the poles for $r_0^{\pm} - r_0^{\mp} = 1$ cancel with the prefactor in brackets. The cancellation of the poles of the $r_0^{\pm}(N)$ should be considered as well for an entire analysis. The analytic expression of the truncated solution up to $\mathrm{N^3LO}$ can be found in [122, 123].

As mentioned before the nonsinglet case can easily be extracted from the singlet case. We can define the solution in equation (1.97) analogously but with the simplification that all U_n commute with R_0. Equation (1.100) becomes

$$U_n = -\frac{1}{n} \tilde{R}_n \tag{1.114}$$

and the nonsinglet solutions are given by (note that all quantities are scalar)

$$\frac{Q(\alpha_s(\mu^2))}{Q(\alpha_s(\mu_0^2))} = \left[1 - \sum_{n=1}^{\infty}\left(\frac{\alpha_s(\mu^2)}{4\pi}\right)^n \frac{\tilde{R}_n}{n}\right] \left[1 - \sum_{n=1}^{\infty}\left(\frac{\alpha_s(\mu_0^2)}{4\pi}\right)^n \frac{\tilde{R}_n}{n}\right]^{-1} \left(\frac{\alpha_s(\mu^2)}{\alpha_s(\mu_0^2)}\right)^{-R_0} \tag{1.115}$$

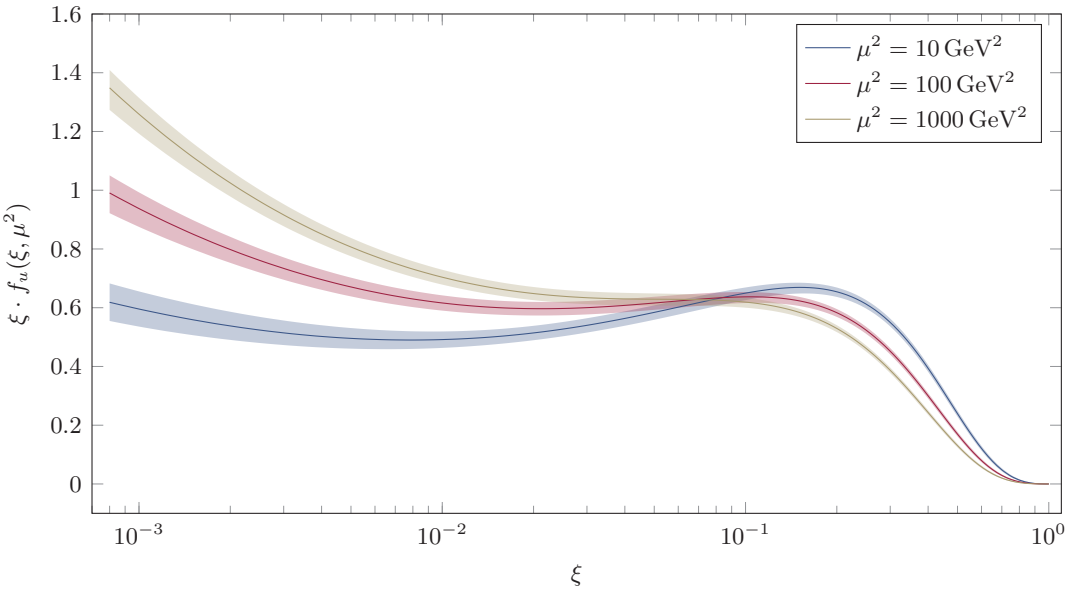


Figure 1.5: Next-to-leading-order up-quark PDFs fitted by the CTEQ collaboration [82] (CT10) at three different scales. The error bands specify the overall fitting error of the 26 parameters used to describe the PDF.

The truncated solutions at NLO are easily derived as

$$\frac{Q(\alpha_s(\mu^2))}{Q(\alpha_s(\mu_0^2))} = \left(\frac{\alpha_s(\mu^2)}{\alpha_s(\mu_0^2)}\right)^{-R_0} \left[1 - \tilde{R}_1 \frac{\alpha_s(\mu^2) - \alpha_s(\mu_0^2)}{4\pi}\right] \tag{1.116}$$

At NLO it is also possible to solve equation (1.94) directly, which gives for the α_s truncated solution the closed exponential form

$$\frac{Q(\alpha_s(\mu^2))}{Q(\alpha_s(\mu_0^2))} = \left(\frac{\alpha_s(\mu^2)}{\alpha_s(\mu_0^2)}\right)^{-R_0} \exp\left[\frac{\alpha_s(\mu_0^2) - \alpha_s(\mu^2)}{4\pi} R_1\right] \tag{1.117}$$

consistent with the truncated result, because $\tilde{R}_1 = R_1$.

In figure 1.5 we show the PDF of the up-quark of the CT10 fit [82] for three different values of the scale. We notice that the evolution causes an enhancement of the small momentum fraction region, while it reduces the probability density for large momentum fraction. This is intuitive as for larger energies the sea particles become more important and easier to generate.

We want to emphasize the importance of the DGLAP evolution for perturbative QCD as a whole. Because low energy physics cannot be described by perturbative QCD we have to fit the PDFs to experimental data, which means a huge loss of predictability for the whole theory. Thanks to the evolution it is not necessary to fit the PDFs for every scale but for only one. Then they can be calculated for all other scales by the

DGLAP evolution. This increases the predictability of the theory tremendously. A typical fit procedure of the PDFs can be separated into the following parts: first for every flavour a sensible parameterization is guessed. Then they are evolved to the scale that dominates the cross section in pQCD. A subset of data is chosen to compare the theoretical description to the experiment. Finally the parameterization is adjusted and the whole procedure starts again until the data are described as precisely as possible. The chosen subset of data is responsible for the restriction of the parameterization. Hence it is a very delicate task to choose as many data as necessary without constricting the predictive efficiency.

1.6 Resummation

So far we considered fixed-order calculations. However, under certain circumstances this approach fails although the small coupling α_s seems to justify the usage of the perturbative series. This is the case if the series is systematically spoiled. The technique widely used to overcome this issue is called *resummation*. It stands for an all-order summation of the terms that spoil the original series. However, resummation is not a definite technique that can be used in every situation. Instead it is a general term that collects many different techniques.

Usually the series is spoiled by large logarithms. A typical perturbative expansion contain terms of the form

$$\alpha_s^n \ln^k(\ldots) \tag{1.118}$$

The power k that can appear at most is dependent on the particular case. Often one encounters $k \leq 2n$ (double logarithmic) or $k \leq n$ (single logarithmic) settings. If the argument of the logarithms becomes very small or very large it counteracts the smallness of the coupling.

In fact we already saw two situations in which a resummation has been taken place, but we did not mention it explicitly. The first was the solution of the renormalization group equation of the coupling itself, see equation (1.31). Expanding the expression

$$\alpha_s(Q^2) = \frac{\alpha_s(\mu^2)}{1 + \frac{\beta_0}{4\pi}\alpha_s(\mu^2)\ln\left(\frac{Q^2}{\mu^2}\right)} = \alpha_s(\mu^2)\sum_{n=0}^{\infty}\left[-\frac{\beta_0}{4\pi}\alpha_s(\mu^2)\ln\left(\frac{Q^2}{\mu^2}\right)\right]^n \tag{1.119}$$

we can identify the logarithms that have been resummed: $\ln(Q/\mu)$. If the ratio of the two scales becomes too large or small we find exactly the situation described above. We can identify it as a single logarithmic resummation. Additionally it demonstrates one possible way to achieve resummation: by solving an appropriate RGE. The second example we already saw is the DGLAP evolution of PDFs. Again a RGE is used to resum the single logarithmic series.

A type of resummation we encounter more explicitly is often in particular regions of phase space, where again large logarithmic contributions arise. For example for the

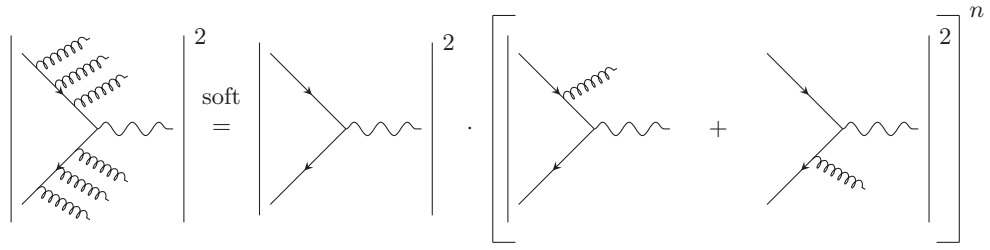


Figure 1.6: Factorization of the invariant amplitude including n soft gluon emissions into a product of the born diagram and the n-th power of a one gluon emission, see equation (1.123b).

Drell-Yan process one will find terms in the partonic part of the form

$$\alpha_s^n \left(\frac{\ln^k(1-z)}{1-z} \right)_+ \tag{1.120}$$

with $k \leq 2n$, $z \equiv Q^2/s$ and $s = \xi_1 \xi_2 S$. The plus distribution is defined by

$$\int_0^1 \mathrm{d}z \left[f(z) \right]_+ g(z) \equiv \int_0^1 \mathrm{d}z \, f(z) \big(g(z) - g(1) \big) \tag{1.121}$$

It regularizes the limit $z \to 1$, where all available energy is transferred into the lepton pair, which means that real higher-order corrections can consist of the emission of soft gluons only. The logarithms are a remnant of the cancellation of virtual and real infrared divergences at the partonic threshold. To achieve a meaningful theoretical prediction we have to resum these terms. This is called *threshold resummation* [125, 126]. It is even more enhanced because of the PDFs that are convoluted with the partonic cross section. They become very big for small momentum fraction, which makes the contribution at $z \to 1$ dominant to a larger domain of the cross section.

We will briefly sketch how the resummation works before presenting a more formal approach. During this and in the subsequent discussions we will follow [50, 127, 128] and references therein. We write down the total Drell-Yan cross section as

$$\sigma(P+n) = \frac{1}{2S} \int \mathrm{d}\Phi_n(P, k_1, \ldots, k_n) |\mathcal{M}_n(P, k_1, \ldots, k_n)|^2 \tag{1.122}$$

where P denotes the produced vector particle, n is the number of gluon emissions, the k_i are the momenta of the gluons, Φ_n is the $n+1$ particle phase space and $\mathcal{M}_n$ the corresponding invariant amplitude. In the infrared limit, where all gluon emissions are soft, the ingredients simplify as

$$\mathrm{d}\Phi_n(P, k_1, \ldots, k_n) \to \mathrm{d}\Phi(P) \prod_{i=1}^{n} \mathrm{d}\Phi_1(k_i) \frac{1}{n!} \tag{1.123a}$$

$$|\mathcal{M}_n(P, k_1, \ldots, k_n)|^2 \to |\mathcal{M}(P)|^2 \big(|\mathcal{M}_1(k_1, \ldots, k_n)|^2 \big)^n \tag{1.123b}$$

Id est the production of the particle P and the emission of the soft gluons factorize into two separate contributions. The factor of $1/n!$ takes into account the bosonic nature of the gluons. A schematic depiction of the factorization of the invariant amplitude is shown in figure 1.6. Summing over all possible numbers of emitted gluons we find an exponentiation of the one gluon emission

$$\sum_{n=0}^{\infty} \sigma(P+n) = \sigma(P) \exp\left[\int \mathrm{d}\Phi_1(k) |\mathcal{M}_1(k)|^2\right] \tag{1.124}$$

If we like to measure not only the total but also a differential cross section, say in the invariant mass Q of the particle P, we have to insert the appropriate condition into the phase space measure

$$\delta(s - Q^2 - 2P \cdot K - K^2) \tag{1.125}$$

where we used the definition of the momentum sum of all emitted gluons

$$K^\mu \equiv \sum_i k_i^\mu \tag{1.126}$$

The term proportional to K^2 can be neglected in the soft limit as being small. After factoring out S from the delta distribution we may use the soft limit, where $P = (\sqrt{S}, 0, 0, 0)$, to introduce the weights

$$w_i \equiv \frac{2k_i^0}{\sqrt{s}} \tag{1.127a}$$

$$w \equiv 1 - z \tag{1.127b}$$

and finally find

$$\sum_{n=0}^{\infty} \sigma(P+n) = \sigma(P) \sum_{n=0}^{\infty} \frac{1}{n!} \left[\int \mathrm{d}\Phi_1\right]^n \left(|\mathcal{M}_1|^2\right)^n \delta\left(w - \sum_{i=0}^{n} w_i\right) \tag{1.128}$$

Apparently the distribution prevents the direct exponentiation. To overcome this problem we transform the cross section into Laplace space with the following integral transformation

$$f(N) = \int_0^\infty \mathrm{d}w \; \mathrm{e}^{-wN} f(w) \tag{1.129}$$

The delta distribution can be used to evaluate the integral and the exponential term factorizes the sum of all weights w_i. Then it is again possible to exponentiate the expression and we can write the resummed cross section in Laplace space as

$$\sigma^{\mathrm{res}}(N) = \sigma_0 \exp\left[\int \mathrm{d}w \; \mathrm{e}^{-wN} \sigma_1(w)\right] \tag{1.130}$$

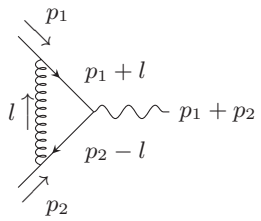


Figure 1.7: One loop QCD correction of the electromagnetic vertex.

The price to pay is the inverse Laplace transform, which is typically not possible analytically and thus a numerical task. This scheme is quite common in many flavours of resummation. However, the exact transform depends on the process and the observable. For example for small transverse momentum resummation it is common to use a Fourier transform, while for Drell-Yan threshold resummation often the Mellin transform is favored. However, all these transforms are related to each other and might be expressed by another after a proper substitution.

With this descriptive picture in mind we can now turn to the more formal approach. As already said, the large logarithms in threshold resummation are a remainder of the cancellation of infrared singularities. In the previous section we got to know the cancellation statement as factorization. Indeed, the resummation is intimately related to the infrared structure of QCD. More precisely, it is based on a further factorization of the cross section in the threshold regime, as we just argued in the previous illustration. Again we will use a simple example which will lead us to more general statements. Let us consider the one loop correction of the electromagnetic quark photon vertex as shown in figure 1.7. The graph G that is depicted in the figure is given in d dimensions by

$$G = \int_0^1 \mathrm{d}\alpha_1 \int_0^1 \mathrm{d}\alpha_2 \int_0^1 \mathrm{d}\alpha_3 \, \delta(1 - \alpha_1 - \alpha_2 - \alpha_3) \int \frac{\mathrm{d}^d l}{(2\pi)^d} \frac{N(l, p_1, p_2)}{D^3} \tag{1.131}$$

where the Feynman parameters α_i disentangle the denominator D which then reads

$$D = \alpha_1 l^2 + \alpha_2 (p_1 + l)^2 + \alpha_3 (p_2 - l)^2 - \mathrm{i}\eta \tag{1.132}$$

and we collect the rest of the structure in the numerator N. A necessary condition to form a singularity is obviously that the denominator vanishes

$$D = 0 \tag{1.133}$$

However this is not sufficient for a singularity as shown in figure 1.8. Due to the regulator $\mathrm{i}\eta$ the integration contour does not cross any singularity. Hence it is possible to deform the contour by Cauchy's theorem to recede from the singularities and the integral is still finite. To actually find a singularity the contour has to be trapped

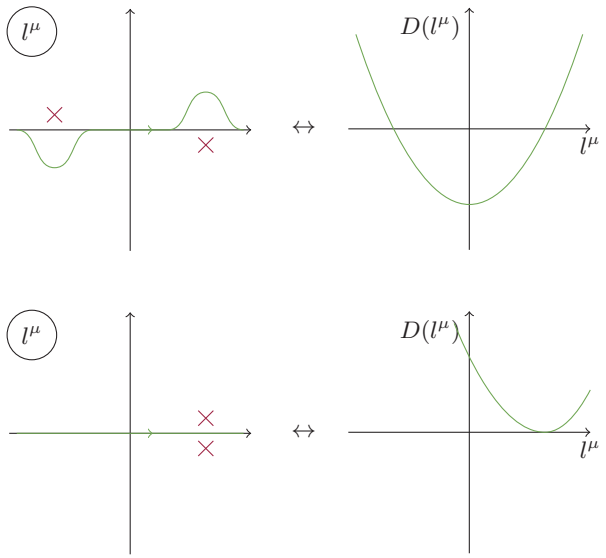


Figure 1.8: Illustration of a pinch singularity. On the left-hand side the position of the poles (red crosses) in the l^μ plain and the integration contour (green) are shown while on the right-hand side the corresponding functional form of the denominator D is shown. In the top row the poles do not coalesce, which makes it possible to deform the contour to recede it from the poles. In the bottom row the poles do coalesce, producing a pinch singularity.

between two of them, which is called a *pinch singularity*. The necessary conditions are given by

$$\frac{\partial D}{\partial l^\mu} = 0 \tag{1.134}$$

which is again illustrated in figure 1.8.

In a more general setup we can consider a graph containing I internal lines and L loops. Then equation (1.131) turns into

$$G = (I-1)! \left[\prod_{i=1}^{I} \int_0^1 \mathrm{d}\alpha_i\right] \delta\Big(1 - \sum_{i=1}^{I} \alpha_i\Big) \prod_{j=1}^{L} \frac{\mathrm{d}^d l_j}{(2\pi)^d} \frac{N(k_i, p_{\text{ext}})}{D^I} \tag{1.135}$$

where the argument p_{ext} is a collective for all external momenta. The internal momenta are labeled by k_i. Then the denominator is given by

$$D = \sum_{i=1}^{I} \alpha_i \big(k_i^2(l_j, p_{\text{ext}}) - m_i^2\big) + \mathrm{i}\eta \tag{1.136}$$

Hence we find a pinch condition for every internal momentum, which is now besides the obvious $D = 0$ given by

$$\frac{\partial D}{\partial l_j^\mu} = 0 \tag{1.137}$$

for all j and μ. Since D is linear in the Feynman parameters α_i, the contour cannot be pinched in an analogous manner to the momenta. Still, for the endpoints of the integrals where $\alpha_i = 0$, the denominator has a singularity, as it has for $k_i^2 = m_i^2$ in which case D becomes independent of α_i. Altogether the conditions are called *Landau equations* and read

$$\begin{aligned} &\alpha_i = 0 \qquad \vee \qquad k_i^2 = m_i^2 \\ &\sum_{i \in \text{loop } l} \alpha_i k_i \varepsilon_{il} = 0 \end{aligned} \tag{1.138}$$

where the incidence matrix ε_{il} is defined by

$$\varepsilon_{il} = \begin{cases} +1 & k_i \text{ in the same direction as } l_l \\ -1 & k_i \text{ in opposite direction to } l_l \\ 0 & \text{otherwise} \end{cases} \tag{1.139}$$

and ensures that all momenta are added vectorially while summing up all loop lines.

Returning to the simple example of the vertex correction shown in figure 1.7 the Landau equations are given by

$$\alpha_1 l^\mu + \alpha_2 (p_1 + l)^\mu - \alpha_3 (p_2 - l)^\mu = 0 \tag{1.140}$$

Three solutions exist to this equation. The first is given if the gluon is collinear to the quark with momentum p_1

$$l^\mu = -z p_1^\mu \tag{1.141a}$$
$$\alpha_1 z = \alpha_2 (1 - z) \tag{1.141b}$$
$$\alpha_3 = 0 \tag{1.141c}$$

the second if the gluon is collinear to the quark with momentum p_2

$$l^\mu = z' p_2^\mu \tag{1.142a}$$
$$\alpha_1 z' = \alpha_3 (1 - z') \tag{1.142b}$$
$$\alpha_2 = 0 \tag{1.142c}$$

and the third if the gluon is soft

$$l^\mu = 0 \tag{1.143a}$$
$$\frac{\alpha_2}{\alpha_1} = \frac{\alpha_3}{\alpha_1} = 0 \tag{1.143b}$$

There also exist configurations where the singularity is collinear as well as soft, which is the case if $z \to 0$ or $z' \to 0$.

A tool exists to represent the solutions of the Landau equations in *reduced diagrams*, in which all lines with vanishing α on the pinch surface are contracted to one point while

every line with $k^2 = m^2$ is left as before, but commonly the decoration is dropped as it is irrelevant for this analysis. To find all solutions to the Landau equations Coleman and Norton invented a graphical method, which simplifies the task enormously [129].

However, the Landau equations are not sufficient for a singularity because the numerator N might cancel it out. It is necessary to perform an IR power counting [83, 99]. It consists of an parameterization of every pinch surface and identification of coordinates being 'intrinsic', id est remaining in the surface, or 'normal', id est moving outside of the surface. We still stick to our example of the vertex correction. Therefore we choose a frame in which

$$p_1^\mu = \frac{Q}{\sqrt{2}}\delta^{\mu+} \tag{1.144a}$$

$$p_2^\mu = \frac{Q}{\sqrt{2}}\delta^{\mu-} \tag{1.144b}$$

and define the parts of the denominator

$$D_1 \equiv l^2 = 2l^+l^- - l_T^2 \tag{1.145a}$$

$$D_2 \equiv (p_1 + l)^2 = 2p_1^+l^- + 2l^+l^- - l_T^2 \tag{1.145b}$$

$$D_3 \equiv (p_2 - l)^2 = -2p_2^-l^+ + 2l^+l^- - l_T^2 \tag{1.145c}$$

where the second equal uses the frame definition and the decomposition of the loop integral is

$$\int \mathrm{d}^4 l = \int \mathrm{d}l^+ \int \mathrm{d}l^- \int \mathrm{d}l_T^2 \int \mathrm{d}\varphi \tag{1.146}$$

Now let us consider the soft pinch surface of equations (1.143), which implies $D_1 = D_2 = D_3 = 0$. For this surface all l^μ are normal variables and there are no intrinsic ones. This means that to leading power in the normal variables the D_i are given by

$$D_1 \to 2l^+l^- - l_T^2 \tag{1.147a}$$

$$D_2 \to 2p_1^+l^- \tag{1.147b}$$

$$D_3 \to -2p_2^-l^+ \tag{1.147c}$$

This gives us a power of four in the denominator as well as in the numerator by the integral measure, which leads to a logarithmic divergence. Let us now turn to the situation in which l is collinear to the particle with momentum p_1, see equations (1.141). Here l^- and l_T^2 are normal variables while l^+ and φ are intrinsic. Hence the D_i are given in leading power in the normal variables by

$$D_1 \to 2l^+l^- - l_T^2 \tag{1.148a}$$

$$D_2 \to 2(p_1^+ + l^+)l^- - l_T^2 \tag{1.148b}$$

$$D_3 \to -2p_2^-l^+ \tag{1.148c}$$

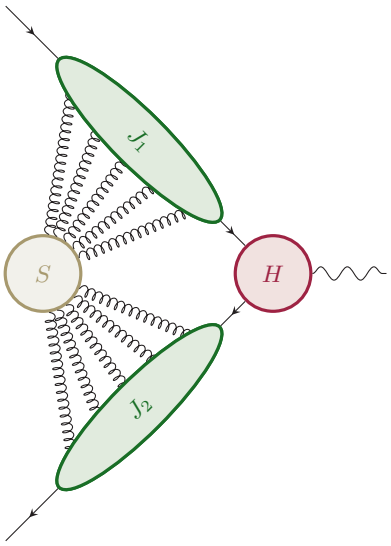


Figure 1.9: Structure of the one loop QCD correction of the electromagnetic vertex close to partonic threshold.

which means the denominator has a power of two as has the integral measure and we find again a logarithmic divergence. The analysis for the situation in which the gluon is collinear to the particle with momentum p_2 is identical, yielding a logarithmic divergence, too.

The analysis of the infrared structure reveals that the considered diagram factorizes at the pinch surfaces as shown in figure 1.9. The hard part H contains only particles that are off-shell, the soft part S on the other hand consists of soft gluons and fermion loops only. The Jet functions J_i collect all terms that are in jet-like parts of the reduced diagrams, including vertices and propagators. In every jet all particles have a large component in the same direction. In the depiction in figure 1.9 we already use that only one line can connect the jets with the hard part. It is possible to show that additional collinear partons do not cause a singular behavior in physical gauges. In covariant gauges the same statement can be made because then all additional lines are equivalent to a rotation of the quark field [83]. Further no lines exist between the hard and the soft part. This is due to the fact that all particles in H are off shell. If an additional soft gluon attaches to the hard part, an additional propagator $1/Q^2$ suppresses the contribution—it belongs to a higher twist.

It is possible to further disentangle the different parts, yielding a factorization of purely scalar functions. Until now the connection between jets and the soft part is given by the following Lorentz structure

$$J_1^{\mu_1 \dots \mu_n} J_2^{\mu_{n+1} \dots \mu_{n+m}} S_{\mu_1 \dots \mu_{n+m}} \tag{1.149}$$

However, all these connections are soft, which leads to further factorization. We will

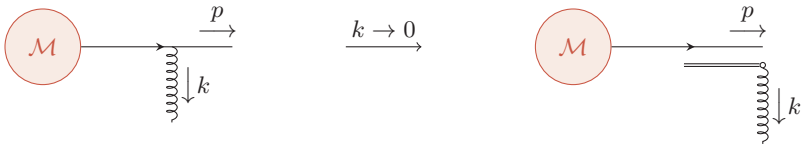


Figure 1.10: Decoupling of a soft gluon attaching to a quark line into the same gluon attaching to an eikonal line that has no information about the original quark line.

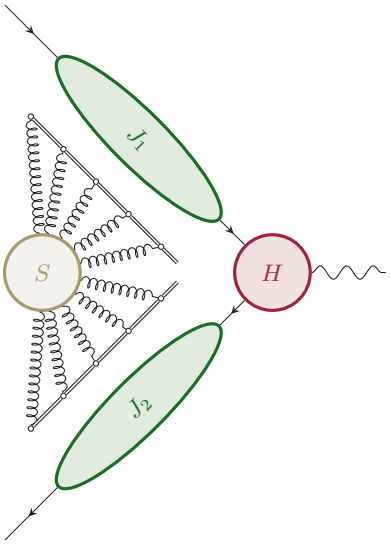


Figure 1.11: Structure of the one loop QCD correction of the electromagnetic vertex close to partonic threshold. In comparison to figure 1.9 the Lorentz structure connecting jets J_i and soft part S is decoupled.

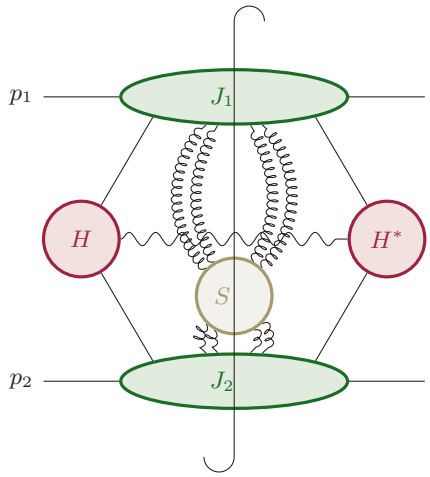


Figure 1.12: Factorization of the Drell-Yan process close to partonic threshold. The final state lepton pair is omitted for clarity.

demonstrate this using figure 1.10. The left diagram is given by

$$\bar{u}(p)\big(-\mathrm{i}g_s t^a \gamma^\mu\big)\frac{\mathrm{i}(\not{p}+\not{k})}{(p+k)^2}\mathcal{M} \tag{1.150}$$

In the soft limit $k^\mu \to 0$ we can neglect the term proportional to k^μ in the numerator and commute the propagator with the vertex. Using the Dirac equation $\bar{u}(p)\not{p} = 0$ we are left with the expression

$$g_s t^a \frac{2p^\mu}{2(p\cdot k)}\big(\bar{u}(p)\mathcal{M}\big) \tag{1.151}$$

The first factor is called an *eikonal* factor, the second factor corresponds to the case where no emission is present. If the momentum p^μ is part of a jet, for example with a large plus component $p^\mu = (p_+, 0, \vec{0}_T)$ the eikonal factor becomes actually independent of it. This means the soft radiation decouples from the quark line, which is usually depicted by an additional double line, called an eikonal line as shown in the right diagram of figure 1.10. In fact it is possible to show this in a much more general way. In a nutshell the soft approximation is used to extract the Lorentz structure of the jets into light-like vectors. Then by use of Ward identities the sum over all possible connections of the gluons to the jets can be replaced by connections to the respective eikonal lines [130–133]. For our example the resulting situation is shown in figure 1.11. Now it is possible to redefine the soft part to include also the eikonal lines and we have succeeded to find a fully factorized form of the vertex at partonic threshold

$$G \sim H \cdot J_1 \cdot J_2 \cdot S \tag{1.152}$$

The additional factorization now opens the window for resummation, in the same way as the factorization of PDFs and hard scattering in the previous section set the

stage for the DGLAP evolution. We will turn again to the Drell-Yan cross section. Using the results we obtained so far, it is possible to write it as [134]

$$\begin{aligned}\hat{\sigma}(w) = H\left(\frac{p_1}{\mu}, \frac{p_2}{\mu}, \xi_1, \xi_2\right) \int \frac{\mathrm{d}w_1}{w_1} \int \frac{\mathrm{d}w_2}{w_2} \int \frac{\mathrm{d}w_3}{w_3} \delta(w - w_1 - w_2 - w_3) \\ \times J_1\left(\frac{p_1 \cdot \xi_1}{\mu}, \frac{w_1 Q}{\mu}\right) J_2\left(\frac{p_2 \cdot \xi_2}{\mu}, \frac{w_2 Q}{\mu}\right) S\left(\frac{w_3 Q}{\mu}, v_i, \xi_i\right)\end{aligned} \tag{1.153}$$

where ξ_i are the gauge vectors defining the jet functions, which are calculated in the $\xi_i \cdot G = 0$ gauge. The v_i are the velocities of the momenta p_i and the weights w_i have been defined in equation (1.127). Here and in the following we suppress flavour indices for simplicity, as the factorization holds regardless of them. The whole situation is depicted in figure 1.12. Note that we connected the jets again with the soft function albeit they factorize as we discussed before, as this is standard convention. Further note that we are just considering the partonic cross section. Hence the hard function H is not to be confused with the function H we encountered in the previous section, where it contained the complete partonic cross section which is now further dissected. Taking now the Laplace moments of the cross section entangles the convolutions and yields

$$\hat{\sigma}(N) = H\left(\frac{p_1 \cdot \xi_1}{\mu}, \frac{p_2 \cdot \xi_2}{\mu}\right) J_1\left(\frac{p_1 \cdot \xi_1}{\mu}, \frac{Q}{\mu N}\right) J_2\left(\frac{p_2 \cdot \xi_2}{\mu}, \frac{Q}{\mu N}\right) S\left(\frac{Q}{\mu N}, v_i, \xi_i\right) \tag{1.154}$$

The cross section is also in Laplace space independent of the arbitrary scale μ

$$\mu \frac{\mathrm{d}}{\mathrm{d}\mu} \hat{\sigma}(N) = 0 \tag{1.155}$$

Introducing the anomalous dimensions of the hard, soft and jet functions

$$\gamma_H\big(\alpha_s(\mu^2)\big) \equiv -\mu \frac{\mathrm{d}}{\mathrm{d}\mu} \ln(H) \tag{1.156a}$$

$$\gamma_S\big(\alpha_s(\mu^2)\big) \equiv -\mu \frac{\mathrm{d}}{\mathrm{d}\mu} \ln(S) \tag{1.156b}$$

$$\gamma_{J_i}\big(\alpha_s(\mu^2)\big) \equiv -\mu \frac{\mathrm{d}}{\mathrm{d}\mu} \ln(J_i) \tag{1.156c}$$

the independence of the cross section with respect to μ becomes

$$\gamma_H\big(\alpha_s(\mu^2)\big) + \gamma_S\big(\alpha_s(\mu^2)\big) + \sum_{i=1}^{2} \gamma_{J_i}\big(\alpha_s(\mu^2)\big) = 0 \tag{1.157}$$

Furthermore gauge invariance, id est

$$(p_1 \cdot \xi_1) \frac{\mathrm{d}}{\mathrm{d}(p_1 \cdot \xi_1)} \hat{\sigma}(N) = 0 \tag{1.158}$$

demands that

$$(p_1 \cdot \xi_1) \frac{\mathrm{d}}{\mathrm{d}(p_1 \cdot \xi_1)} \ln(J_1) = G\left(\frac{p_1 \cdot \xi_1}{\mu}, \alpha_s(\mu^2)\right) + K\left(\frac{Q}{\mu N}, \alpha_s(\mu^2)\right) \tag{1.159}$$

where we defined the functions

$$G\left(\frac{p \cdot \xi}{\mu}, \alpha_s(\mu^2)\right) \equiv -(p \cdot \xi) \frac{\mathrm{d}}{\mathrm{d}(p \cdot \xi)} \ln(H) \tag{1.160a}$$

$$K\left(\frac{Q}{\mu N}, \alpha_s(\mu^2)\right) \equiv -(p \cdot \xi) \frac{\mathrm{d}}{\mathrm{d}(p \cdot \xi)} \ln(S) \tag{1.160b}$$

The same equation holds for the second jet and thus we omit the index from now on. Because the anomalous dimension γ_J does only depend on μ but not on the gauge vectors we can deduce

$$\mu \frac{\mathrm{d}}{\mathrm{d}\mu} K + \mu \frac{\mathrm{d}}{\mathrm{d}\mu} G = 0 \tag{1.161}$$

by taking the derivative of $\ln(J)$ with respect to μ as well as $p \cdot \xi$. Now, by defining the Sudakov anomalous dimension

$$\gamma_K\big(\alpha_s(\mu^2)\big) \equiv -\mu \frac{\mathrm{d}}{\mathrm{d}\mu} K \tag{1.162}$$

which immediately implies

$$\mu \frac{\mathrm{d}}{\mathrm{d}\mu} G = \gamma_K\big(\alpha_s(\mu^2)\big) \tag{1.163}$$

it is possible to integrate back both differential equations for the jet functions, which yields the resummed structure

$$J\left(\frac{p \cdot \xi}{\mu}, \frac{Q}{\mu N}, \alpha_s(\mu^2)\right) = C(\alpha_s(\mu^2)) \exp\left[-\int_{Q/N}^{p \cdot \xi} \frac{\mathrm{d}\lambda}{\lambda} \int_{Q/N}^{\lambda} \frac{\mathrm{d}\mu}{\mu} \gamma_K(\alpha_s(\mu^2))\right] \tag{1.164}$$

where we focus on the most important logarithmic terms. The exact integration requires a very careful calculation, which can be found for example in [135]. To conclude this part of the discussion we have to notice that for QCD processes the hard and the soft function are dependent on the color states. Hence their factorization is more complex but still possible. Instead of scalar functions H and S become matrices in color space. Also the exponentiation for the non-abelian theory is highly non-trivial. One possible proof relies on defining subsets of eikonal diagrams with modified color factors that occur in the exponent, called *webs* [136, 137]. However, the proof is also possible using the RGE approach similar to the jet functions, but becomes more complicated due to the non-abelian nature.

To present some actual results and discuss some phenomenology we write down the cross section following the common notation in [51, 138]. Further we restrict ourselves

to a quark anti-quark pair in the initial state. Then the cross section is given in Mellin moment space by

$$\sigma(N) \sim f_q(N,\mu^2) f_{\bar{q}}(N,\mu^2)\, \omega_{q\bar{q}}\left(N, \frac{Q^2}{\mu^2}, \alpha_s(\mu^2)\right) \tag{1.165}$$

where the $f_q(N)$ denote the PDFs in Mellin space and $\omega_{q\bar{q}}(N)$ the partonic scattering. In the resummed cross section we replace

$$\omega_{q\bar{q}} \to \omega_{q\bar{q}}^{\text{res}} \tag{1.166}$$

and the resummed function is of the form

$$\omega_{q\bar{q}}^{\text{res}}\left(N, \frac{Q^2}{\mu^2}, \alpha_s(\mu^2)\right) = C_{qq}\left(N, \alpha_s(\mu^2), \frac{Q^2}{\mu^2}\right) \Delta_{q\bar{q}}^{\text{DY}}\left(N, \alpha_s(\mu^2), Q^2, \frac{Q^2}{\mu^2}\right) \tag{1.167}$$

where $\Delta_{q\bar{q}}^{\text{DY}}$ is exactly the exponent that consists of gluon radiation at partonic threshold. As expected from the previous calculations in this section it splits up into

$$\ln \Delta_{q\bar{q}}^{\text{DY}}\left(N, \alpha_s(\mu^2), Q^2, \frac{Q^2}{\mu^2}\right) = 2 \ln \Delta_q(N, Q^2, \mu^2) + \ln \Delta^{\text{DY}}(N, Q^2) \tag{1.168}$$

with

$$\ln \Delta_q(N, Q^2, \mu^2) = \int_0^1 \mathrm{d}x \frac{x^{N-1} - 1}{1-x} \int_{\mu^2}^{(1-x)^2 Q^2} \frac{\mathrm{d}\nu}{\nu} A_q(\alpha_s(\nu^2)) \tag{1.169a}$$

$$\ln \Delta^{\text{DY}}(N, Q^2) = \int_0^1 \mathrm{d}x \frac{x^{N-1} - 1}{1-x} D^{\text{DY}}\big(\alpha_s((1-x)^2 Q^2)\big) \tag{1.169b}$$

It is easy to map the functions to the expressions in our previous considerations. The coefficient function C corresponds to the hard function H including all off-shell particles. The jet functions J consisting of collinear radiations map to the process independent Δ_q. Because we have two initial partons, the terms appears in the exponent twice. The soft emissions that are not collinear we denoted as S also exponentiate and give a process dependent exponent Δ^{DY}. The integrands A_q and D^{DY} have perturbative expansions. Using the coupling to a specific order[12] we can solve the integrals in the exponents and sort them respective to their resummation order

$$\omega_{q\bar{q}}^{\text{res}} = \big(1 + \alpha_s C_{qq}^{(1)} + \ldots\big) \exp\big[\ln(N) h_0(\lambda) + h_1(\lambda) + \alpha_s h_2(\lambda) + \ldots\big] \tag{1.170}$$

with $\lambda \equiv \alpha_s b_0 \ln(N)$. This leads us back to the discussion at the very beginning of this section. We pointed out that large logarithms spoil the original perturbative series and

[12] For subtle effects that depend on the chosen form of the coupling, see chapter 6.

resummation reorders it in such a way that we obtain a new perturbative series. This is exactly given by equation (1.170). Every term in the exponent sum corresponds to a resummation order. Each of them consists of all orders in the original series. The leading order, called *leading log* (LL), uses the one of the coefficient function and the term h_0. In next-to-leading log (NLL) the term h_1 is taken into account additionally. For NNLL h_2 but also $C^{(1)}$ have to be considered. In case of the Drell-Yan process the leading term in the exponent is given by

$$h_0^{\mathrm{DY}}(\lambda) = \frac{C_F}{2\pi b_0 \lambda}\big[2\lambda + (1-2\lambda)\ln(1-2\lambda)\big] \tag{1.171}$$

Another common way to characterize the series is called *tower resummation* which first expands the exponential and characterizes the order afterwards

$$\omega_{q\bar{q}}^{\mathrm{res}} = \sum_{k=0}^{\infty} \alpha_s^k C_{k,2k} \ln^{2k}(N) + \sum_{k=0}^{\infty} \alpha_s^k C_{k,2k-1} \ln^{2k-1}(N) + \dots \tag{1.172}$$

where the first sum is called the leading log, first and second the next-to-leading log and so on. It is important to notice that the two definitions are not equal (for more details see appendix D in [9]).

To obtain the desired cross section it is necessary to perform the inverse Mellin transform. However, because of the Landau pole of the coupling, also the resummed expression contains a Landau singularity, which is already present at leading log, see equation (1.171). The logarithm induces a branch cut into the N plane, starting at the Landau pole

$$N_L = \exp\left(\frac{1}{2b_0\alpha_s}\right) \tag{1.173}$$

As is explained in detail in appendix E, the inverse Mellin transform consists of a contour integral that has to be to the right of all poles of the integrand. This is obviously not possible, but it has been shown that integrating to the left of the Landau pole but still to the right of all other poles yields the correct result up to exponentially suppressed corrections [139]. This is called the *minimal prescription*. We notice that it is also common to use Cauchy's theorem to bend the contour of the integral to improve numerical convergence.

However, the resummed cross section is not sufficient to describe the whole cross section. This is evident as terms that are subleading at threshold, id est not singular for $z \to 1$, are not included but may be the main contributors for other kinematic regions. To achieve results that are accurate in both regimes it is necessary to match both terms. This is done by adding the resummed expression at order $\mathrm{N}^k\mathrm{LL}$ with the full result at order $\mathrm{N}^k\mathrm{LO}$. To avoid double counting of terms that occur in both of them, one subtracts the resummed expression expanded to the desired order

$$\mathrm{d}\sigma^{\mathrm{match}} = \mathrm{d}\sigma^{\mathrm{full}}_{\mathrm{N}^k\mathrm{LO}} + \mathrm{d}\sigma^{\mathrm{res}}_{\mathrm{N}^k\mathrm{LL}} - \mathrm{d}\sigma^{\mathrm{res}}_{\mathrm{N}^k\mathrm{LL}}\Big|_{\mathcal{O}(\alpha_s^k)} \tag{1.174}$$

We want to finish this section with a discussion of the generic behavior that can be expected from threshold resummation. Hadronic cross sections at the threshold limit have the generic form

$$\lim_{N\to\infty} \sigma(N) \sim \exp\left[-\ln^2(N)\right] \tag{1.175}$$

which implies a *Sudakov suppression*. But since we resum the correction to the partonic cross section instead of the hadronic one, we also have to consider the contributions of the PDFs. Because they are also Sudakov suppressed for large N

$$\lim_{N\to\infty} f_p(N) \sim \exp\left[-\ln^2(N)\right] \tag{1.176}$$

we find for the partonic cross section indeed a *Sudakov enhancement*

$$\lim_{N\to\infty} \hat{\sigma}(N) \sim \lim_{N\to\infty} \frac{\sigma(N)}{\left(f_p(N)\right)^2} \sim \exp\left[+\ln^2(N)\right] \tag{1.177}$$

This means that the PDFs radiate too few gluons at threshold, so that all in all the effect is an enhancement. That is the reason why threshold resummation is such an important tool to describe experimental data at all.

Chapter 2

Perturbative QCD using a graphics processing unit

> The city's central computer told you? R2D2, you know better than to trust a strange computer!
>
> — *C3PO*

We present the development of the most important ingredient to perform perturbative QCD calculations on a graphics processing unit: a suitable integration routine. For this purpose the famous Vegas routine of Peter Lepage is ported to the GPU and subsequently improvements that become applicable to a GPU are applied. The resulting software component will improve all applications that make use of it, as long as the computational load of the integrand function is large enough, and will not degrade performance compared to other available implementations of the Vegas algorithm if no GPU is used at all. The remaining tasks for a widespread use of GPUs by the community are described and partly also performed, paving the way to faster, more precise and more economic numeric calculations.

2.1 Introduction

All calculations in perturbative QCD incorporate numerical integrations. They cannot be avoided as already the parameterizations of large distance physics are always present and not known analytically. But even without this complication many observables include phase space integrations that cannot be carried out analytically. As a consequence the numerical integration and hence the implementation of the analytical expressions is vital to compute theoretical predictions. As the integrands are often very complicated and even badly converging the algorithm that is mainly used is the Vegas algorithm [140, 141]. Its strength is due to its iterative and adaptive behavior, which works also well in for multidimensional integrands. This is achieved by applying importance instead of stratified sampling for high dimensional integrals. Other very important features are the reliable error estimate, a dimension independent convergence rate and the freedom of the integrand to be not contiguous.

To improve the numerical performance of every algorithm there are two ways without changing the algorithm as a whole, id est on the hardware side. The first is to increase the processor speed, the second to increase the parallelism. In fact both ways have been taken. For example the processor speed of an Intel i386 DX core in 1985 with 12 MHz has grown by magnitudes to the Sandy Bridge in 2011 with 3.7 GHz. The drawback of this improvement is the exploding power consumption as the power is proportional to the frequency cubed. Especially for high performance computing this results in high energy costs also for cooling of the processors. The alternative approach, the increase of parallelism, is much cheaper as the power is only proportional to the number of cores. This drives the massive expansion of computer clusters. Even in personal computers it is nowadays common that the CPU consists of several cores. The idea of parallelism is exhausted by graphics processing units (GPUs). Instead of a few cores they consist of thousands of cores, each of them with a relatively small frequency. This makes them exceptional cheap in terms of frequency per consumed power. Although their original purpose was to compute computer graphics, which also explains their name, they are used also in high performance computing. The field of general purpose computation on GPUs (GPGPU) is still quite young but already established and it is developing fast offering nowadays even GPUs that are primary designed for GPGPU. In the list of supercomputers a growing number is equipped with GPU co-processors [142].

Unfortunately it is not possible to use code that has been written for a common CPU directly on a GPU. In November 2006 the introduction of the NVIDIA compute unified device architecture (CUDA) and the corresponding programming model made GPGPU available for a large community. It offers a C interface but allows also several C++ techniques up to the 2011 standard. Nevertheless existing codes have to be mostly rewritten to be executable on the GPU. The alternative OpenCL can be used also on non NVIDIA GPUs, which offers a even cleaner C++ interface. Presently the competition between CUDA and OpenCL is open and it is impossible to rank one over the other definitely. Another alternative is the OpenACC framework, which follows the idea of OpenMP. By additional pragmas existing code can be transformed by a minimum of effort to produce parallelized executables. The idea relies on a smart compiler optimizing the code respective to the underlying architecture such that the programmer only has to declare certain loops to be parallelizable and the compiler takes care of all the rest. However, in practice still a huge amount of knowledge about the GPU structure is necessary, such that one could have taken CUDA or OpenCL in the first place for all but the simplest applications.

Vegas is a Monte Carlo algorithm, which means that it samples the integrand randomly. From the function values evaluated at the sample points the integral and its error estimate are computed and—in the case of Vegas—an adaption suitable for the integrand is performed. It is trivial that all function evaluations are independent and thus it is possible to compute them in parallel. The Cuba library [143] which besides of Vegas contains several other Monte Carlo integration routines implemented the parallelization for multiple CPUs only recently [144]. Instead of using multiple CPUs the problem seems almost as being designed to be ported for a GPU. A few years

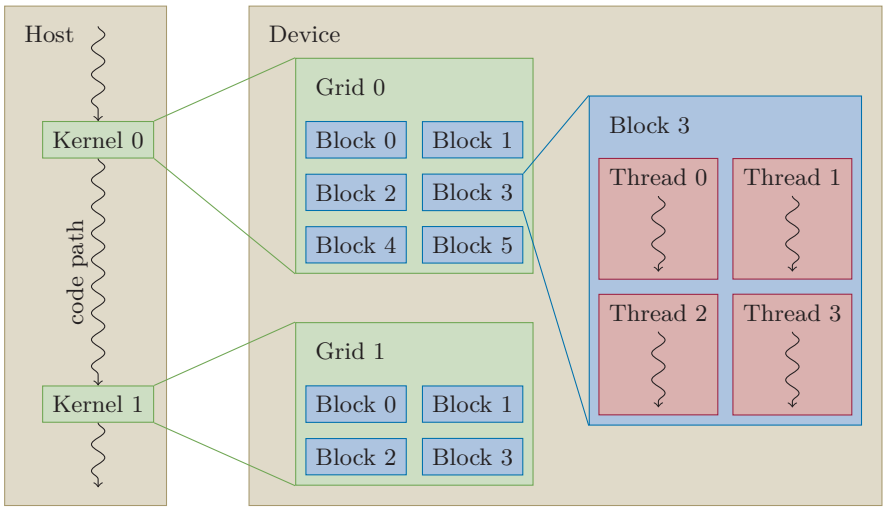


Figure 2.1: Thread hierarchy of the CUDA programming model.

ago the pioneering work [145] made the proof of principle by parallelizing the original program using the CUDA framework.

In this chapter we will present a completely new implementation of the Vegas algorithm, that runs using a CPU as well as on a GPU. It is written entirely in modern C++ and extends the original interface to satisfy the requirements of the special environment of GPUs. For this purpose we will briefly explain the CUDA programming model in section 2.2. Then we will give a detailed presentation of the Vegas algorithm in section 2.3. In the consecutive section 2.4 we highlight the key steps in terms of the parallelization and perform several performance measurements. Finally we discuss a possible way for applications in perturbative QCD using GPUs in section 2.5. The conclusions are given in section 2.6. This chapter will explicitly not present the programming interfaces of Vegas or other developed tools to simplify the use of a GPU. Instead they are presented together with fi*n*ael in appendix C, which currently embeds our Vegas implementation.

2.2 CUDA programming model

The CUDA programming model has been invented to avoid the somewhat complicated detour by graphics APIs or shaders. A simple C interface makes it possible to focus on parallel algorithms instead of the management. It has been taken special care to guarantee a simple execution model combined with communication between concurrent operation branches. We will give in this section only a very brief description to introduce the most important terms. For a more details introduction see the official CUDA programming guide [146] or an introductory text book like [147].

The core of the programming model are so called *kernels*. They are functions that

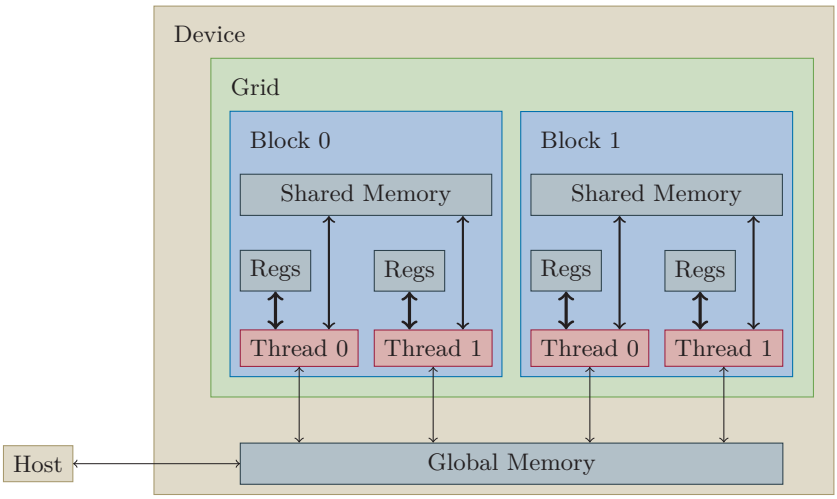


Figure 2.2: Simplified memory hierarchy of the CUDA programming model. The thickness of the lines orders the memory bandwidth.

connect the *host* and the *device*. The term host is used for the (most likely) sequential processing with the CPU. The device is synonym to the GPU. This means every kernel is called from the host and will execute parallel instructions on the device. Each kernel is called for a fixed number of parallel *threads*. They are grouped into *blocks*. All blocks together form the *grid*. The structure is illustrated in figure 2.1. The index of the block and the thread are accessible in the kernel and all subsequently called device functions such that each thread can be identified uniquely. A kernel call has to specify the number of blocks (grid size) and the number of threads per block (block size). The product of both numbers gives the total number of (potentially) parallel executing threads. In general the whole structure may be up to three dimensional due to the origin in 3d computer graphics. For our purposes one dimension will be sufficient. The subdivision of the grid into blocks ensures scalability over several devices with different numbers of multiprocessors. This means that the number of blocks is partitioned to the available multiprocessors. As a consequence threads of different blocks cannot communicate. Because the execution sequence of the blocks in the grid is not guaranteed (as is the thread sequence in a block) it is necessary to translate every problem in such a way that the blocks are completely independent. Inside of a block the threads can communicate via the *shared memory*, see figure 2.2. However in practice the threads are not executed independently but in groups called *warps*, which execute completely synchronous.[1] This implies that possible conditional branches will cause partial serial execution, if threads follow distinct code paths. This effect is called warp divergence and should be avoided to achieve maximal efficiency.

As figure 2.2 indicates several memory types are available on the device. The perhaps

[1] This is not true any more for the brandnew Volta architecture, which introduces independent thread scheduling.

Model	GeForce GTX 680	GeForce GTX TITAN
Architecture	Kepler	Kepler
Compute capability	3.0	3.5
Number of SMs	8	14
Number cores per SM	192	192
Clock rate	1.0585 GHz	875.5 MHz
Global memory	4 GB	6 GB
Constant memory	64 kB	64 kB
L2 cache	512 kB	1.5 MB
Shared memory per SM	48 kB	48 kB
32 bit registers per SM	64 k	64 k
Max. registers per thread	63	255
Warp size	32	32

Table 2.1: Selection of technical specifications of used devices.

most important type is the *global memory*, which can be accessed from every thread on the device as well as from the host and is stable also between different kernel calls. But the bandwidth from the host to the device and vice versa is limited by the PCIe bus. Thus every memory transfer between host and device should be minimized. The bandwith for transfers on the device is distinctly higher, but can be further improved by using the shared memory of every block (streaming multiprocessor SM). In figure 2.2 also the registers of every thread are shown, which have naturally the highest bandwidth. We omitted several caches as the L1 and L2 cache for simplicity.

Every GPU differs in architectural details. This makes it possible to optimize every problem for every architecture individually. Since we aim for a broad application field we cannot assume that a specific architecture dominates. Hence we did not perform any such optimizations despite signaling the compiler the underlying compute capability. As we will see also the very general implementation will give remarkable good results. For our measurement we have two devices at hand: the GeForce GTX 680 and the GeForce GTX TITAN which both belong to the Kepler architecture and have compute capability 3.0 and 3.5, respectively. Some of their specifications are collected in table 2.1. We can detect the tendency of newer devices to have a larger number of cores with lower clock rate, further improving the FLOPS per Watt. However it is not that easy to rank two devices only by comparing their specifications. The actual performance is much more dependent on the considered algorithm, to what extend it utilizes the device and how the ratio of arithmetic operations and memory transfers suits the specific device. However, still the specifications give a rough orientation.

2.3 Vegas algorithm

As already mentioned Vegas is a Monte Carlo integration routine that prefers importance over stratified sampling in high dimensions. Before we describe the algorithm in detail, we will introduce the very basics of the topic. We follow mainly the explanations in [140, 141, 148].

Let us assume a function f we like to integrate in a d-dimensional volume V

$$I = \int_V \mathrm{d}^d x \, f(x) \tag{2.1}$$

Then the central limit theorem used for Monte Carlo integrations states that the value can be approximated by sampling N random points $x_i \in V$ to approximate the integral

$$I_N \equiv \frac{V}{N} \sum_{i=0}^{N-1} f(x_i) \tag{2.2}$$

In the limit of large N the estimate will converge to the exact value

$$\lim_{N\to\infty} I_N = I \tag{2.3}$$

Naturally for finite N the estimate will depend on the randomly chosen samples. This dependency is described by the variance

$$\sigma_N^2 \equiv \frac{1}{N} \left[V \int_V \mathrm{d}^d x \, f^2(x) - I^2 \right] \tag{2.4}$$

Due to the central limit theorem the estimate tends towards a Gaussian distribution. Therefore we can approximate for large N

$$\sigma_N^2 \approx \frac{1}{N-1} \left[\frac{1}{N} \sum_{i=0}^{N-1} f^2(x_i) - I_N^2 \right] \tag{2.5}$$

Note the factor $(N-1)^{-1}$ instead of N^{-1} to correct the sample bias, giving us the unbiased sample variance.

One extremely important advantage of Monte Carlo integrations is their flexibility for non regular integration volumes. If the volume V is difficult to parametrize it is sufficient to use another volume that includes V and using a Heaviside function that excludes the surplus volume in the integrand. This comes with the cost of a potentially larger variance, as all samples outside of V will not contribute to the estimate, effectively reducing the overall number of samples.

Instead of increasing the number of samples we can also combine several estimates to improve the final estimate. This will turn out useful as the final algorithm will adapt

to the integrand in several iterations as we will see soon. Consider m independent estimates $I^i_{N_i}$ with variances $\sigma^i_{N_i}$ and $i \in \{0, \dots m-1\}$. Each estimate may be calculated with a distinct number of samples N_i. Then the total estimate and variance are given by

$$I^m \equiv \left(\sigma^m\right)^2 \sum_{i=0}^{m-1} \frac{I^i_{N_i}}{\left(\sigma^i_{N_i}\right)^2} \tag{2.6a}$$

$$\left(\sigma^m\right)^2 \equiv \left[\sum_{i=0}^{m-1} \frac{1}{\left(\sigma^i_{N_i}\right)^2}\right]^{-1} \tag{2.6b}$$

where $I^i_{N_i}$ and $\sigma^i_{N_i}$ are the estimate and the variance of the iteration i using N_i samples. To kind of measure the consistency of the individual estimates the χ^2 per iteration can be used

$$\frac{\chi^2}{m} \equiv \frac{1}{m-1} \sum_{i=0}^{m-1} \frac{\left(I^i_{N_i} - I^m\right)^2}{\left(\sigma^i_{N_i}\right)^2} \tag{2.7}$$

As it will turn out it is more useful to use the equivalent form

$$\frac{\chi^2}{m} = \frac{1}{m-1} \sum_{i=0}^{m-1} \frac{\left(I^i_{N_i}\right)^2 - I^m \cdot I^i_{N_i}}{\left(\sigma^i_{N_i}\right)^2} \tag{2.8}$$

The equivalence is easy to prove by inserting equation (2.6a), such that the term $\sim (I^m)^2$ cancels against half of the mixed term of the binomial expansion. An increase of the χ^2 per iteration such that it becomes much greater than one indicates that the single estimates are not consistent to each other and should be treated with care.

To keep track of the results over several iterations one simply has to store the value of the three sums

$$\sum_{i=0}^{m-1} \frac{\left(I^i_{N_i}\right)^k}{\left(\sigma^i_{N_i}\right)^2} \qquad , k \in \{0, 1, 2\} \tag{2.9}$$

2.3.1 Importance sampling

The very idea of importance sampling is to sample the random points not uniformly but determined by a normalized probability density p such that the variance is minimized. The normalization is given by

$$\int_V \mathrm{d}^d x \, p(x) = 1 \tag{2.10}$$

Sampling by this density is equivalent to changing the integral measure from $\mathrm{d}^d x$ to $\mathrm{d}^d x \, p(x)$ or

$$I = \int_V \left(\mathrm{d}^d x \, p(x)\right) \frac{f(x)}{p(x)} \tag{2.11}$$

Note that the previous discussed integral estimates use the special case of a constant density $p = V^{-1}$. The estimate of the integral and its variance become

$$I_N = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(x_i)}{p(x_i)} \tag{2.12a}$$

$$\sigma_N^2 = \frac{1}{N} \left[\int_V \mathrm{d}^d x \, \frac{f^2(x)}{p(x)} - I^2 \right] \approx \frac{1}{N-1} \left[\frac{1}{N} \sum_{i=0}^{N-1} \frac{f^2(x_i)}{p^2(x_i)} - I_N^2 \right] \tag{2.12b}$$

To minimize the variance, id est to find a better estimate, we minimize the bracket respective to the probability density constrained by the side condition of its normalization

$$\frac{\delta}{\delta p} \left[\int_V \mathrm{d}^d x \, \frac{f^2(x)}{p(x)} - I^2 + \lambda \int_V \mathrm{d}^d x \, p(x) \right] = 0 \tag{2.13}$$

which results in

$$p(x) = |f(x)| \left[\int_V \mathrm{d}^d x \, |f(x)| \right]^{-1} \tag{2.14}$$

Using this density causes the variance to vanish and the integral estimate is perfect. This is even true for a function that has positive as well as negative values in its image, because we can shift the integrand by a constant such, that the whole image has only one sign. However, at the same time this result is evidently meaningless, because we already have to know the desired integral to determine the optimal probability density.

In the actual algorithm the probability density is assumed to be separable

$$p(x) \equiv \prod_{j=0}^{d-1} p_j(x_j) \tag{2.15}$$

which is important to overcome storage limitations, we will see in subsection 2.3.3. Also the normalization is supposed to hold in every single dimension

$$\int \mathrm{d}x_j \, p_j(x_j) = 1 \qquad \forall j \in \{0, \ldots, d-1\} \tag{2.16}$$

In this case the optimal density in direction i can be computed similar to the general case to be

$$p_i(x_i) = \frac{\tilde{p}(x_i)}{\int \mathrm{d}x_i \, \tilde{p}(x_i)} \tag{2.17}$$

with

$$\tilde{p}(x_i) \equiv \sqrt{\left(\prod_{j \neq i}^{d-1} \int \mathrm{d}x_j \right) \frac{f^2(x)}{\prod_{j \neq i}^{d-1} p_j(x_j)}} \tag{2.18}$$

As we will see soon the Vegas algorithm will try to adapt the optimal density in several iterations, using the single iterations for improvement of the density. Finally we like to comment that the density concentrates the samples where the integrand is largest, explaining the name 'importance sampling'.

2.3.2 Stratified sampling

The idea of stratified sampling is to subdivide the integration volume into n boxes, such that in every box a Monte Carlo integration with in average N/n samples is carried out using equations (2.2) and (2.5). The total estimate and variance are then given by summing up the corresponding equivalents of all boxes

$$I_N = \frac{1}{n} \sum_{i=0}^{n-1} I_{i,N_i} \tag{2.19a}$$

$$\sigma_N^2 = \frac{1}{n^2} \sum_{i=0}^{n-1} \sigma_{i,N_i}^2 \tag{2.19b}$$

with

$$\sum_{i=0}^{n-1} N_i = N \tag{2.20}$$

We like to express the variance of each box by the variance of the function defined by

$$\sigma^2 \equiv \frac{1}{V} \int_V d^d x \, f^2(x) - \left[\frac{1}{V} \int_V d^d x \, f(x) \right]^2 \tag{2.21}$$

They are asymptotically related by

$$\sigma_{i,N_i}^2 \approx \frac{\sigma_i^2}{N_i} \tag{2.22}$$

where σ_i^2 is identical to σ^2 with the integration volume constricted to the volume of box i. Then we may write

$$\sigma_N^2 = \frac{1}{n^2} \sum_{i=0}^{n-1} \frac{\sigma_i^2}{N_i} \tag{2.23}$$

The minimum respective to the number of samples in box j can be calculated to be proportional to the variance in the box. We demonstrate it for the case with only two boxes ($n = 2$). Then the optimum for the first box is given by

$$\frac{\sigma_0}{N_0} = \frac{\sigma_1}{N - N_0} \qquad \Leftrightarrow \qquad \frac{N_0}{N} = \frac{\sigma_0}{\sigma_0 + \sigma_1} \tag{2.24}$$

This means for stratified sampling one samples the region with largest variance. As for the importance sampling precise knowledge about the integrand would be necessary, which makes the integration redundant. We will see in the next subsection how both strategies can still be applied successfully.

2.3.3 Adaption

The problem of importance as well as stratified sampling is that they require the solution of the integral we want to solve to minimize the variance. To come over this problem one makes the algorithm iterative. We start with importance sampling. Therefore we subdivide the integration volume into M^d *increments*[2]. We choose this particular naming because along each axis we define the probability density to be a step function. Normalizing integration limits to $[0, 1]$ in every dimension we define the step function for dimension j by

$$p_j(x_j) \equiv \sum_{i=0}^{M-1} \frac{\Theta(x_j^{i+1} - x_j)\Theta(x_j - x_j^i)}{M\Delta x_j^i} \tag{2.25}$$

where $\Theta(x)$ is the Heaviside function and the x_i^j define the increments along axis j with

$$0 = x_j^0 < x_j^1 < \ldots < x_j^{M-1} < x_j^M = 1 \tag{2.26}$$

such that

$$\sum_{i=0}^{M-1} \Delta x_j^i = 1 \tag{2.27}$$

and $\Delta x_j^i \equiv x_j^{i+1} - x_j^i$. As mentioned before the total probability density is given by the product of the densities in the individual directions, see equation (2.15). We can now see the benefit of the factorization of the probability density, as it reduces the number of data to be stored from M^d to $M \cdot d$. Hence only due to the factorization the adaption becomes usable for reasonable M and large d.

The iterative algorithm consists in principle of three steps

(1) Use a constant density to sample the integration volume with N random points.

(2) Store the result, updating the sums in equation (2.9), and analyze the contributions of the increments to the whole estimate to refine the probability density.

(3) If another iteration is requested sample the integration volume with N random points using the refined probability density. Then goto step (2). Else calculate the cumulative result of all iterations using equations (2.6) and (2.8).

In step (2) the actual adaption takes place. It is designed such that regions with large contributions to the estimate are assigned a large weight. As we have seen previously the optimal probability density (2.17) is similar for every dimension. Thus also the adaption can be performed for every dimension j individually. It is depicted in figure 2.3 in two dimension showing the change from the initial increments due to the adaption

[2]Some Vegas implementations call them bins [149], not to be confused with the bins we will introduce in the actual implementation of stratified sampling.

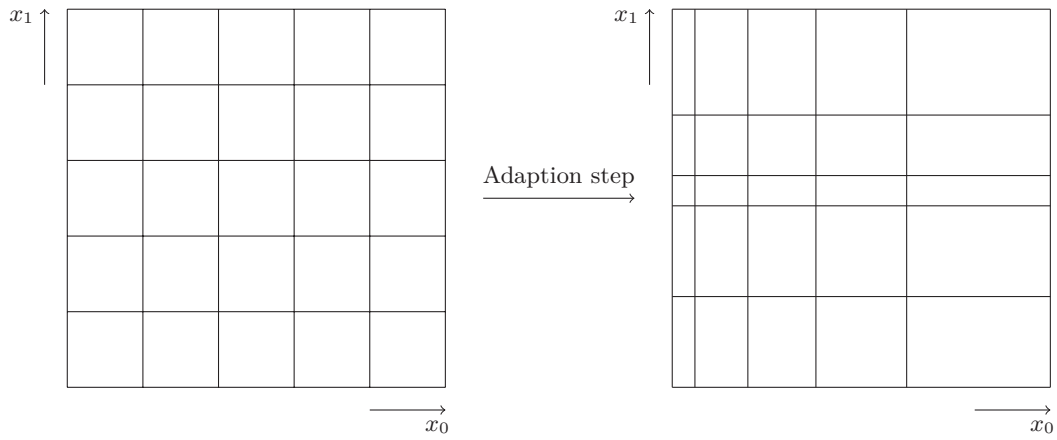


Figure 2.3: Adaption step of Vegas increments in two dimensions.

assuming that large contributions are at low x_0 and medium x_1. To accomplish the adaption one defines

$$w_j^i \equiv \frac{\bar{f}_j^i \Delta x_j^i}{\sum_{k=0}^{M-1} \bar{f}_j^k \Delta x_j^k} \tag{2.28}$$

with the accumulated values that have been sampled in the increment respective to this particular direction

$$\bar{f}_j^i \equiv \sum_{x_j \in [x_j^i, x_j^{i+1}]} \sum_{x_k \neq x_j} \frac{f^2(x)}{\prod_{l \neq j} p_l^2(x_l)} \tag{2.29}$$

Note that in the continuous limit this becomes

$$\bar{f}_j^i \approx \frac{1}{\Delta x_j^i} \int_{x_j^i}^{x_j^{i+1}} \mathrm{d}x_j \left(\prod_{k \neq j} \int_0^1 \mathrm{d}x_k \right) \frac{f^2(x)}{\prod_{l \neq j} p_l(x_l)} \tag{2.30}$$

which nicely suits equation (2.17). A nice description of the algorithm is given in the original paper [140]. There the weights are multiplied by a fictional factor K such that the w_j^i can be interpreted as subincrements, such that every increment is subdivided in at maximum $K + 1$ subincrements per dimension. Then the subincrements are put together in such a way that again M increments are retained. These are defined by containing all the same number of subincrements, which causes a change of their sizes. The algorithm does not alter if the bare w_j^i are used. The adaption is saturated in dimension j, if $w_j^i = w_j^k$ for all i and k. In practice the algorithm has shown to be not stable enough. Therefore the common notion is to damp it to avoid possibly

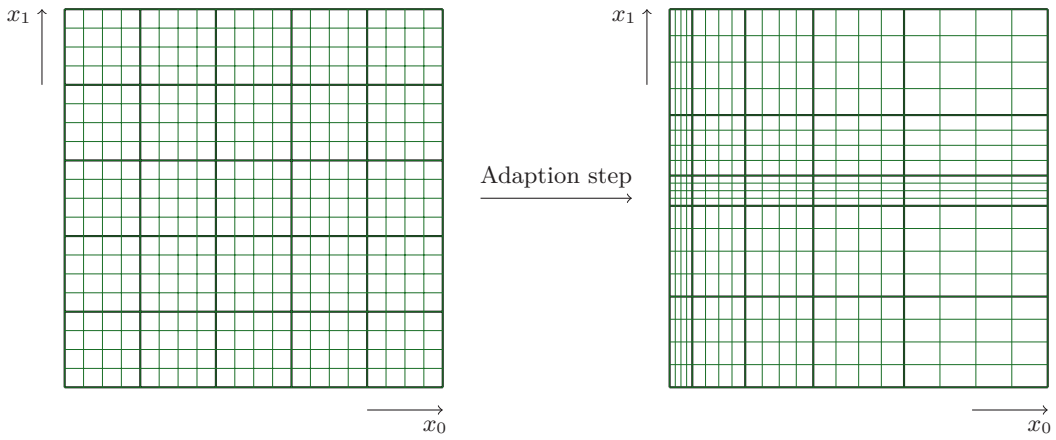


Figure 2.4: Adaption step of Vegas increments (thick black lines) in two dimensions incorporating bins (thin green lines) applying stratified sampling.

destabilizing changes by redefining the weights

$$\tilde{w}_j^i \equiv \left[\frac{w_j^i - 1}{\ln\left(w_j^i\right)}\right]^\alpha \tag{2.31}$$

The parameter α determines the adaption rate. Typical values are between one and two. Note that the adaption can be disabled by setting $\alpha = 0$.

Now let us turn to stratified sampling. It is possible to reuse the increments structure to apply also this distinct sampling strategy. Therefore we subdivide the integration volume into bins[3] but in such a way that the same integer number of bins is in every increment, see figure 2.4. In every of the bins a Monte Carlo integration with at least two samples is performed. To achieve an adaption of the increments to the region with highest variance the weights are not fed by the squared function values but rather with the variance obtained from the minimal Monte Carlo integrations. By this modification the original adaption algorithm can be left unchanged. Because the increments will tighten where the variance of the estimate is largest while holding the number of samples per bin and the number of bins per increment constant, the effective number of samples increases where the variance is largest. In this way the combination of increments and bins constructs the boxes that we introduced to describe stratified sampling. But note that this construction is not perfect, as the boxes do not vary in size and the Monte Carlo integration is performed in each box, while we perform it in each bin and achieve the increase/decrease of samples in a box by varying the size of the increments. However, the net effect is identical, hence the implementation is conform with a stratified sampling.

[3]Some Vegas implementations call these boxes [149]. However this term is occupied by the boxes we defined in subsection 2.3.2.

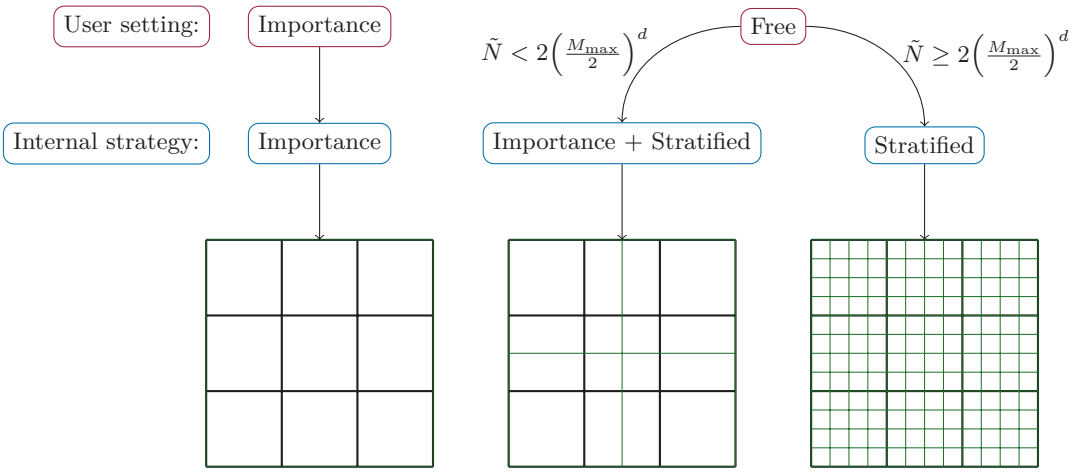


Figure 2.5: Strategy determination of Vegas. If importance sampling is requested explicitly only one bin (thin green lines) is used, containing the whole integration volume. If the strategy is left free, it is selected depending on the number of requested samples $\tilde{N}$, which in this case may differ from the number of actual samples N with $\tilde{N} \geq N$. Increments are drawn with thick black lines.

2.3.4 Strategy determination

This final ingredient to finish the Vegas algorithm is to decide which strategy should be applied. The short answer is: it depends. Naively stratified sampling seems to be more effective as we aim for a reduction of the variance and this does not necessarily originate from the region where the magnitude of the integrand is large. But since every bin has to contain at least two sample points, the total number of samples grows exponentially with the dimension d

$$N = 2N_b^d \tag{2.32}$$

where N_b is the number of bins per dimension. Id est to apply stratified sampling also in high dimensions it would be inevitable to reduce the number of bins and consequently the number of increments tremendously, which would limit the adaption algorithm greatly. And since the ability to adapt the integrand is the major strength of the whole algorithm stratified sampling becomes inferior compared to importance sampling in high dimensions, as importance sampling does not suffer from this problem. To incorporate this fact Vegas selects the adaption strategy considering the actual requested number of samples and the dimension of the integrand. Although this is true in principle the common implementation is slightly more complex as shown in figure 2.5. Vegas uses as maximum of possible increments per dimension $M_{\max}$. Further the number of samples N is not necessarily identical to the number of requested samples $\tilde{N}$. If the user selects importance sampling explicitly, Vegas will use only one bin that includes the whole integration volume, which means it disables the stratification. In this case $M = M_{\max}$ and $N = \tilde{N}$. If the user delegates the decision to Vegas two outcomes

are possible. If the number of requested samples is too small to perform a reasonable stratified sampling (Vegas demands at least two bins per increment per dimension), still importance sampling is taking place. However, in this case the integration volume is divided into several bins, which will in general not coincide with the increments. This is because the total number of bins is limited by half of the samples, while the total number of increments may exceed the number of samples. Because the number of samples per bin is fixed they are distributed somewhat more uniformly in the integration volume. As the bins size will change respective to the increments they do not counteract the importance sampling, although the estimates of the individual bins are not taken into account for the adaption algorithm. For this strategy the number of increments is $M = M_{\max}$ but to accommodate the bins the total number of samples maybe smaller than requested $N \leq \tilde{N}$. If finally the number of requested samples is sufficient to apply stratified sampling, Vegas will align the bins inside each increment, adjust the number of increments and samples such that all have integer values and, of course, use stratified sampling for the adaption of the increments. Due to the adjustments there are in general fewer increments per dimension $M \leq M_{\max}$ and also less samples than requested $N \leq \tilde{N}$. Finally we want to emphasize again that the exponential growth of increments can only be handled as every dimension is treated independently, reducing M^d to $M \cdot d$.

2.4 Vegas using a GPU

The algorithm described in the last section is greatly parallelizable, as the evaluations of the integrand function of all samples are independent. Therefore the naive approach is simple. In every iteration step:

(1) Generate $N \cdot d$ random numbers to determine the random points on the host.[4]

(2) Transfer the random numbers together with other settings to the device.

(3) Launch a kernel to calculate a random point and subsequently the integrand function in every thread, storing the result on the device.

(4) Copy the results to the host.

(5) Reduce the results.

(6) Perform post analysis, including the adaption of the increments.

2.4.1 Performance measure

The actual Vegas version we developed is presented in appendix C.6. We will refer to it as 'fi*n*ael Vegas'. A special feature of fi*n*ael Vegas is that it is able to perform the

[4]Parallel generation of random numbers is not trivial. We will treat this problem in subsection 2.4.2.

System	Host only & GTX 680	GTX TITAN
CPU	Intel Core i5-4460 (Haswell)	Intel Xeon E5-2609
Clock rate	3.20 GHz	2.50 GHz
Memory	8 GB	32 GB
Cache	6 MB	10 MB
OS	openSUSE Leap 42.2	openSUSE 13.2

Table 2.2: Selection of technical specifications of used hosts.

exact same integration on the host as well as on the device. This not only simplifies debugging of the integrand, which is especially difficult in parallel applications. It guarantees that fi*n*ael Vegas can be used even if no GPU or a specialized compiler is available. Every C++ compiler should be sufficient. As an independent opponent we have chosen the freely available and widely used Vegas implementation of the GNU Scientific Library (GSL) [149].

To measure the performance of the distinct Vegas versions, we use as measure the time per integration sample. The actual tool is part of fi*n*ael and presented in appendix C. Note that, in computing communities two other measures are often favored. The first of them is floating point operations per second (FLOPS). It relies on the idea that a processing unit has a theoretical limit and for a given algorithm one can deduce how good it utilizes the unit. This number is also quiet stable for different processing units. The second measure is the memory bandwidth, which is especially useful for problems that are not compute but memory bound, which is often the case for algorithms using a GPU. However, both suffer from the problem that one can increase the measure by superfluous instructions. And, more important to us, they require knowledge of the number of FLOPs[5] or memory transfers. As Vegas is a general purpose integrator, it is impossible to know how many FLOPs will be executed in the kernel or how many data are copied along with the functor that is integrated. This means although time is in general a more unstable measure, as it depends heavily on the underlying hardware, it will give us a clear picture of the potential of Vegas on a GPU without usurping to be generally true for all given integrands.

The underlying hardware specifications of the host systems are given in table 2.2. The technical data of the GPUs has already been reported in table 2.1. The compiler and their settings are given in table 2.3.

As first test case we choose the inverse Mellin transform of PDF splines, which are presented in chapter 5. In fact the results in figure 5.5 are obtained using the GPU to be able to use a sufficient number of samples to reduce the variance such that the spline fluctuations are not multiplexed by the noise of Vegas. The integrand seems to be a good representative for a large class of integrands, where a parallelization should

[5]Note that the 's' is not a capital letter, meaning FLOPs being the plural of FLOP (floating point operation), in contrast to FLOPS (floating point operations per second).

System	Host only	GTX 680	GTX TITAN
Host compiler	g++ 6.2.1	g++ 5.3.1	g++ 4.8.3
CUDA compiler	—	nvcc release 8.0	nvcc release 7.0
Optimization Level	3	3	3
std	C++14	C++11	C++11
arch	—	sm_30	sm_35

Table 2.3: Compiler versions and flags.

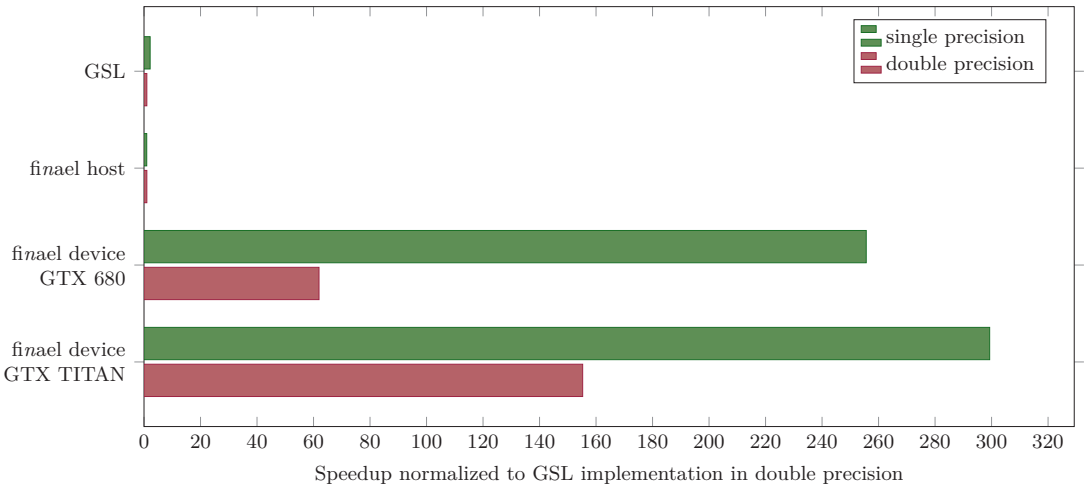


Figure 2.6: Speedup against the GSL vegas function of the naive implementation of finael Vegas. See text for details of the measurement and peculiarities of the different setups.

be considered, as it contains a lot of FLOPs, concentrating the total execution time in this function for a serial application. In the actual measurement we use 10^5 sample points per integration, which are enough to utilize the device but are not chosen to favor it, which would be the case the the number of samples were a power of the warp size, which is 32. To measure the times we used a simple tool that is presented in appendix C.2 using the `std::chrono::steady_clock`. The measurement consists of a warm up integration that is ignored and then 20 integrations consisting of eight iterations. The time measured is divided by the total number of evaluated samples for every integration. The averaged results are compared to the times achieved by the GSL routine in double precision, giving the speedup shown in figure 2.6 for single as well as double precision. For scientific applications in most cases only double precision is relevant. However, since the devices of our tests are mainly optimized for single precision, it is worthwhile to consider them too to demonstrate the full potential of GPUs.

The first observation in figure 2.6 is that the host version of fi*n*ael shows the same performance as the GSL implementation for double precision. For single precision, however, GSL seems to be faster by roughly a factor of two. This is surprising for several reasons. First is is not possible to use the GSL routine in single precision at all. To perform the measurement we had to cast between the GSL Vegas and the integrand function to achieve single precision execution at least in the integrand function. This means that the routine itself is unchanged. Since the integrand function is identical for all measurements, we would not expect any improvement due to the Vegas routine but only the integrand, which in turn should show up also in the fi*n*ael host version. However, for some reason the compiler seems to be able to perform much more efficient optimizations in the precision mixing version of GSL than for the single precision only executions with fi*n*ael. Now let us turn to the speedup achieved by the GPUs. For the GTX 680 the computation is more than 60 times faster for double and more than 250 times faster for single precision. The large difference is due to the preference of the device for single precision, as discussed before. The GTX TITAN contains a lot more processors designed for double precision, which results in a large gain in this respect. For double precision it is more than 150 times faster than the host version, for single precision even almost 300 times faster. Albeit we had to change the setup slightly for the GTX TITAN in double precision. It turned out that the block dimension of 512 we used for all other time measurements with fi*n*ael is not optimal for the given integrand. Therefore we used a block dimension of 32 instead.

2.4.2 Random number generation

We have demonstrated the overwhelming superiority of GPUs in terms of execution time. However, this comes with a cost: large memory consumption. This is troublesome in two ways. First it limits the number of samples by the available memory and second we know that the memory bandwidth between host and device is quite slow, giving also some impact on the execution time.

For a serial execution it is easiest to hold only that much memory as necessary, which is mostly even on the stack. But parallelized we have to store for example the results of all samples for subsequent processing. These are inevitable, but in our naive implementation we also have to store the random numbers that are used to determine the sample point. We now aim for a direct production of the random numbers on the device to come over this problem.

Generating random numbers on a computer is obviously a very delicate task, as all computations are deterministic by construction. To obtain random numbers it would be necessary to observe a physical process which underlies true randomness. Indeed the computer has access to suitable observables, for example the fluctuations of the supply voltage. However, in practice this is cumbersome. Additionally it is not necessary to have real randomness. Instead it is sufficient to produce numbers that are deterministic but seem to be random. Functions that provide such numbers are called *pseudo random number generators* (PRNGs). The randomness of them can be tested against expectations that would hold for truly randomness. The to our knowledge currently most extensive tests are provided by the TestU01 library [150], also known by the names of its three levels 'Small Crush', 'Crush' and 'Big Crush' consisting of 10, 96 and 160 tests respectively. A random number generator that passes all of these tests is called Crush-resistant. But this is not necessary for most practical issues, although it is surely comforting. We also like to mention that pseudo randoms numbers even have a benefit over truly random numbers: they are reproducible, which is extremely useful for comparisons or debugging purposes.

In our naive implementation of Vegas we used a very common class of PRNGs, which are linear congruential generators (LCGs). They consist of a function f that generates a sequence of integers

$$f(n) = (a \cdot n + c) \mod m \tag{2.33}$$

using the result as next input, so f is called a state transition function. Then a subsequently applied function g (the output function) converts the integers uniformly into a real value, mostly in the interval $(0, 1)$, for example

$$g(n) = \frac{n}{m} \tag{2.34}$$

Here the multiplier a and the modulus m are positive integers. The constant c has to be smaller than m and is mostly chosen to be zero, in which case the generator is a *multiplicative linear congruential generator* (MLCG).[6] The quality of a MLCG is determined by the choice of a and m. Extensive studies have been performed to find good values, see for example [153]. To further improve the generators several generators can be combined as proposed in [154]. In our naive implementation we followed their best values for a combination of two distinct MLCGs. A very impressive fact about the

[6] Another way to generate numbers that seems to be random are so called quasi random sequences, which sample a given volume more uniformly as a MLCG does. A particular useful algorithm are Sobol sequences [151, 152]. We will however not consider them in this chapter.

generator is that is passes all but one test of the Small Crush test battery, while the famous Mersenne twister algorithm [155] which is also used in the GSL Vegas version fails two of the tests. To further improve the generator we adopted the idea proposed in [148] that an additional shuffle board will break up still present serial correlations. Note that by using a shuffle board one looses the ability of a MLCG to jump forward in the sequence that still holds after the combination of several MLCGs [154]. Further we did not prove whether this really improves the randomness in terms of Crush-resistance.

In principle MLCGs are capable to be used in parallelized applications, since different seeds will produce distinct sequences. However, as just described, they are known to be flawed and typically do not pass randomness tests. Although this will to some certainty not affect the quality of Vegas we decided us for another ansatz proposed in [156], which is Crush-resistant and explicitly manufactured for the parallel use on GPUs. In contrast to 'conventional' MLCGs that focus on the state transition function f by finding good values for a and m while the output function g mapping the integers to the interval $(0, 1)$ is trivial, they use a very simple state transition function, which degrades it to a kind of simple counter. Hence these generator type is called 'counter based'. To achieve the randomness they focus instead on the output function g, using advanced cryptographic techniques. As a result the PRNG Philox is presented, which has all described properties. Additionally it is very attractive due to its very small internal state and huge speed on the device. On the host it is still reasonable fast achieving half the speed of the Mersenne Twister.

To measure the effect of the PRNG we isolate the process of generating the random numbers (if this happens on the host) and the kernel call that calculates the integrand function, which is reduced to a sum over all coordinates in the hypercube, to need only a small amount of time in the execution. Additionally we perform a synchronization after the kernel to avoid a flawed measurement by the asynchronous execution of the kernel with the host code. The results are shown in figure 2.7 for 10^6 sample points in 20 dimensions. The large number of dimensions focuses the measurement on the production of random numbers. A larger number of dimensions was not possible for single precision without changing the code base. The reason is that the total number of increments M^d exceeds the limit of single precision for larger dimensions. For consistency we did not increase the number of dimensions for double precision. To average the process time per sample, the measurement consists of a warm up run of one iteration followed by the measured integration, consisting of 20 iterations. The time is taken for every single iteration and averaged for the presented results in figure 2.7.

For the host system we detect that the Philox PRNG is only slightly faster than the two combined MLCGs with additional shuffle board, which we called 'L'Ecuyer', after its inventor. For the devices we measure also a large improvement of the process time using the Philox PRNG, which can be ascribed to two effects. First, as mentioned before, the memory transfer of the random numbers from the host to the device is extremely slow. And second the random numbers themselves are generated in parallel instead of serially. Note that the GTX TITAN system is slower than the GTX 680 system using the L'Ecuyer PRNG, as the host system is slower, which overcompensates

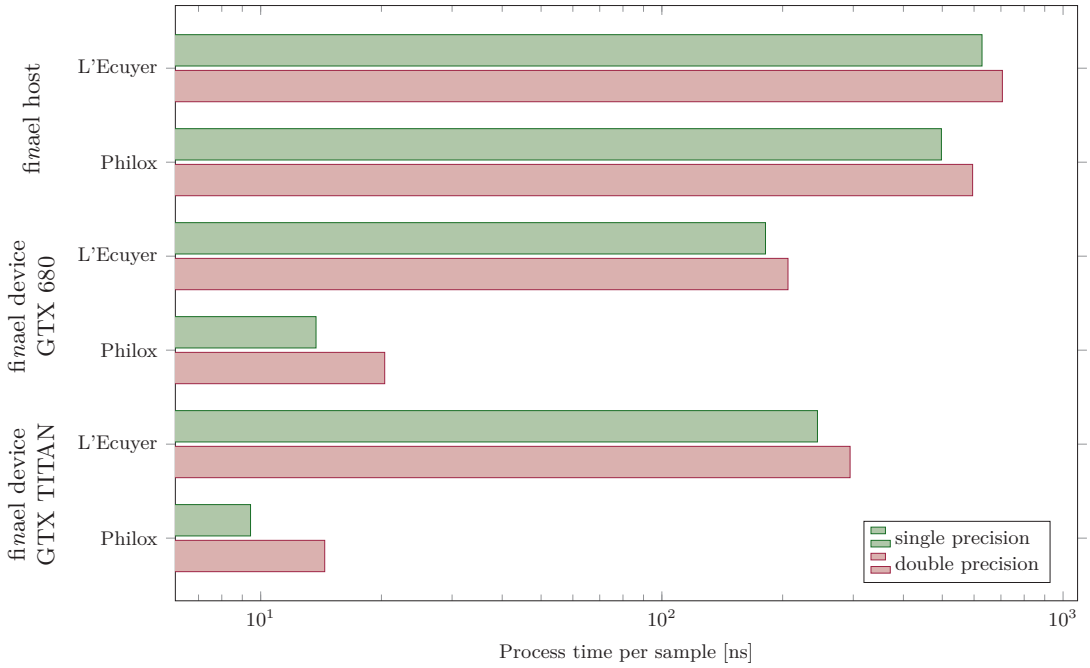


Figure 2.7: Execution time of the generation of random numbers and a minimal integration kernel of Vegas for the L'Ecuyer and the Philox PRNG. Details of the measurement are provided in the text.

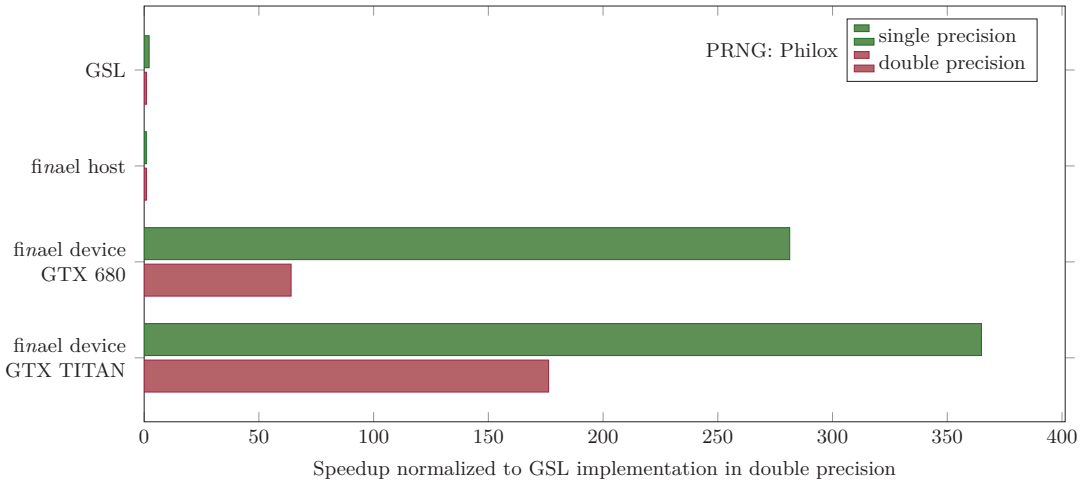


Figure 2.8: Same as figure 2.6, but using the Philox PRNG for the fi*n*ael implementations.

the superiority of the GTX TITAN over the GTX 680, and for this measurement the random number generation is dominant and executed on the host.

Summarizing we can improve Vegas for the host as well as for the device by using Philox as PRNG, not only in respect of memory as initially intended, but also in terms of speed. Further the quality of the produced random numbers improves and becomes Crush-resistant. To quantify the speed improvement for a typical use case we repeat the measurement of the last subsection (see figure 2.6) but with a Philox PRNG instead of the naive implementation using the L'Ecuyer PRNG. The results are shown in figure 2.8. As we expect, the times do not change or only insignificantly for the host version. On the devices however we detect a measurable improvement. This means that the usage of a device made the originally compute bound calculation less compute but more memory bound. However, compared to other applications of GPGPU as lattice QCD, which are almost completely memory bound, we are still mostly compute bound, although the memory transfers contribute to a measurable degree. The improvement is dependent on the GPU, as we find for the GTX 680 a modest improvement of three percent for double precision, but a already sizable improvement of ten percent for single precision. In case of the GTX TITAN setup the improvement is even bigger, being about 13 percent for double and 22 percent for single precision. As a consequence also the relative factors to the host version improve significantly in these cases. The single precision calculation on the GTX 680 is now more than 280 times faster than the double precision calculation on the host. For the GTX TITAN the calculations are more than 170 faster for double and more than 360 times faster for single precision.

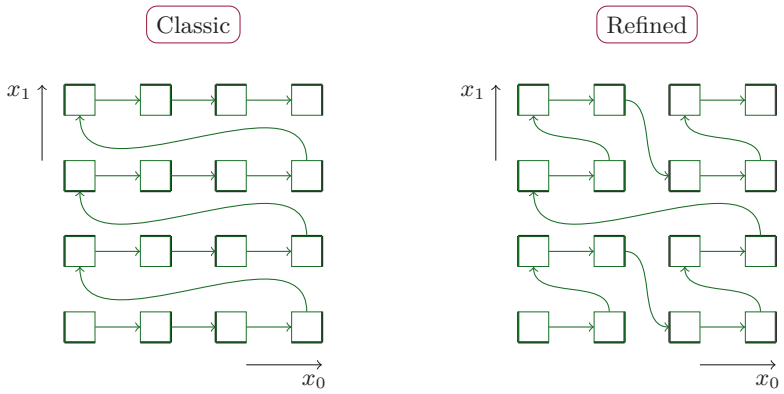


Figure 2.9: Processing of bins in two dimensions using the classic manner (left) and refined for stratified sampling (right). The bins (thin green lines) are separated for clarity. Increment borders are drawn with thick black lines. The green arrows indicate the processing sequence of the bins.

2.4.3 Sample reduction

Optimizing the PRNG is certainly the most important step to reduce the memory usage of Vegas. Having this done, the program is only left with two large blocks of memory: the results of the samples (N times the size of the floating point type) and their positions in respect to the segments for the adaption algorithm ($N \cdot d$ times the size of an integer). Clearly the first one cannot be avoided, while for the second one it is possible, at least for stratified sampling. As shown in figure 2.5 all bins are aligned in the segments. Because we know of every sample its parent bin, we can also deduce its parent segment. Note that this is not possible for the two other internal strategies, as a bin may be part of several segments of even contain them completely. However, for stratified sampling the knowledge of the concrete increment by the sample index makes storing the segment indices superfluous. Additionally reduction is besides matrix matrix multiplication perhaps *the* classic example for using a GPU at all, which implies possible speed improvements, too. For us this is even more interesting, as the samples are stored on the device anyway. So we do not only benefit by the faster reduction, but also of the reduced amount of memory that has to be copied to the host after the reduction (one floating point type per segment).

Unfortunately it is not that easy. First the classical processing of the bins it not compatible with typical reduction algorithms, which require the data in sequential memory for maximal efficiency. Hence we changed the processing as shown in figure 2.9 for two dimensions. Instead of processing all bins in one dimension, while holding all other bin indices constant, the processing is separated into a processing of segments in the manner as before the bins and a processing of bins in the segments. Hence all data that belong to a segment will reside next to each other in the memory. The penalty of the refined processing sequence is the higher computational cost for calculating the bin index, which should however be negligible for integrands with high computational

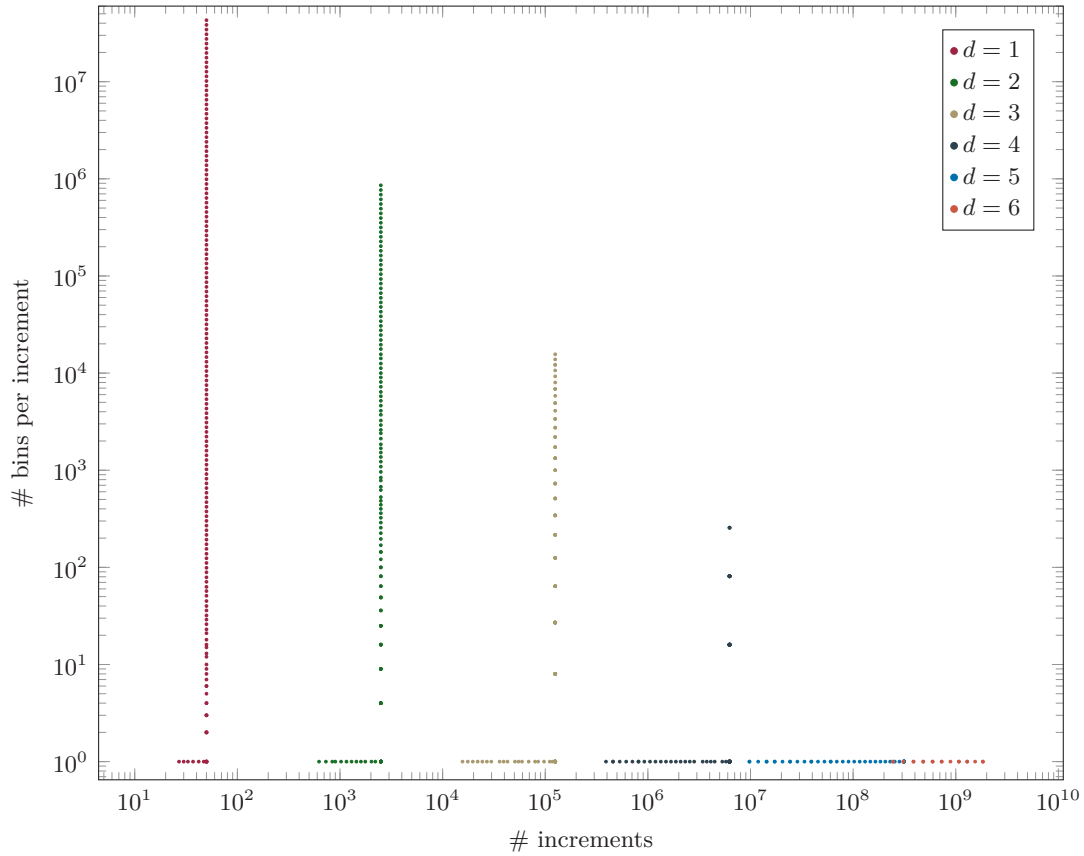


Figure 2.10: Selection of Vegas configurations of bins and increments for $d \in \{1, \ldots, 6\}$.

costs.

The second point to be considered for reducing the data on the device is its utilization and the actual benefit in terms of saved memory transfer in respect to the results of the samples. For this purpose we investigate the actual number of bins per increment depending on the dimension as shown in figure 2.10. The single points represent a possible configuration where stratified sampling can be applied. We selected them by dividing the interval of requested sample points $\tilde{N}$ between two and the limit of unsigned integers on our machine logarithmically. For most configurations every bin contains two samples. For some it might become larger, especially as long as there are very few bins per increment. This causes in figure 2.10 that several points are indistinguishable. Large gaps between points are a quite reliable indication. However, for the following analysis the exact number of samples N does not add any value. Still the product of both axes in figure 2.10 is a good approximation for N. First we can observe that the number of increments in our current implementation is limited to $M_{\max} = 50$ per dimension. For few samples the number is reduced to align with the bins, approaching the maximal value as the number of samples increases. For all these

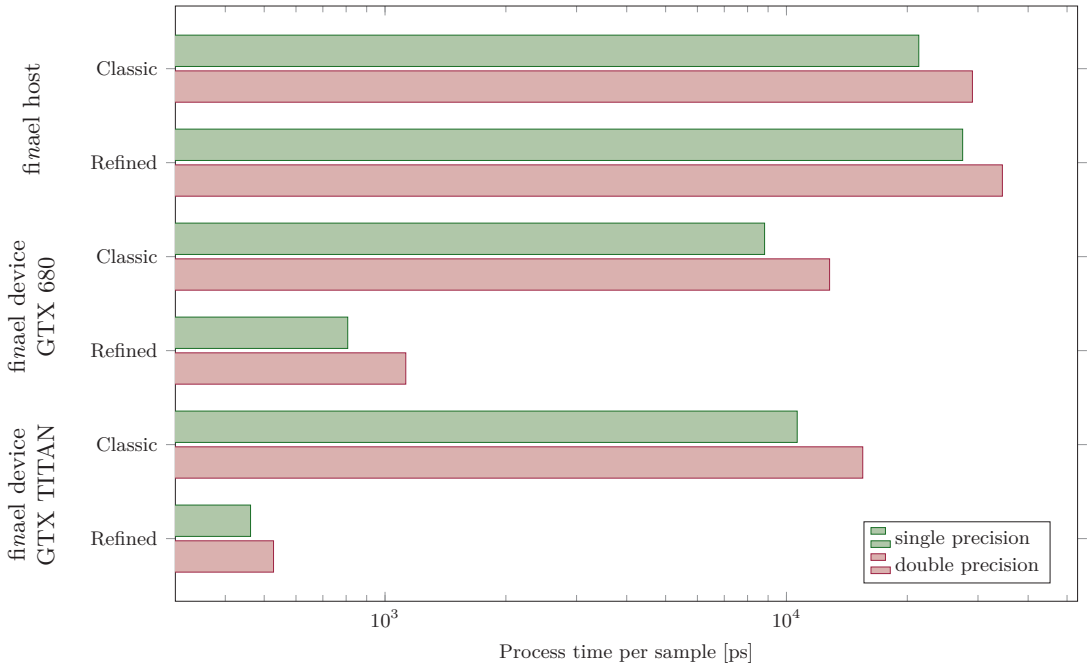


Figure 2.11: Execution time of the classic reduction and the refined version. Bin processing is included in the measurement. Further details are provided in the text.

configurations every increment contains only one bin. Assuming two samples per bin, the amount of data that have to be copied to the host does not reduce at all, as the two function values are replaced by an Monte Carlo estimate and its variance. Furthermore, to be efficient, the device has to be fully utilized, which means that all configurations with less than a few thousand bins per increment are likely to be doomed in terms of speed. Hence only for $d \leq 3$ there seem to be configurations that may provide additional speed improvement beside the memory benefit.

In our first attempt we implemented the reducer very naive, such that one iterates over all increments, reducing one after another. This is obviously a bad idea for all but the points in the top half in figure 2.10, as it causes at least M^d (most likely inefficient) kernel calls followed by small copy instructions. Hence we improved the approach by using the smallest possible multiple of the warp size for a single increment, pooling several increments in one kernel call. This reduces the area, where the device is inefficient due to too small utilization to the bottom left corner in figure 2.10.

To demonstrate the potential use of the refined reduction also in terms of speed under certain circumstances we choose the following setup: the time measurement includes the evaluation of the samples points as well as the reduction. Former is included because we have also to take account for the more complex bin processing. Apart from that the integrand is $f(x) = x$ in one dimension (equivalent to the measurement in the last subsection), such that also the PRNGs computation time is reduced. To focus on

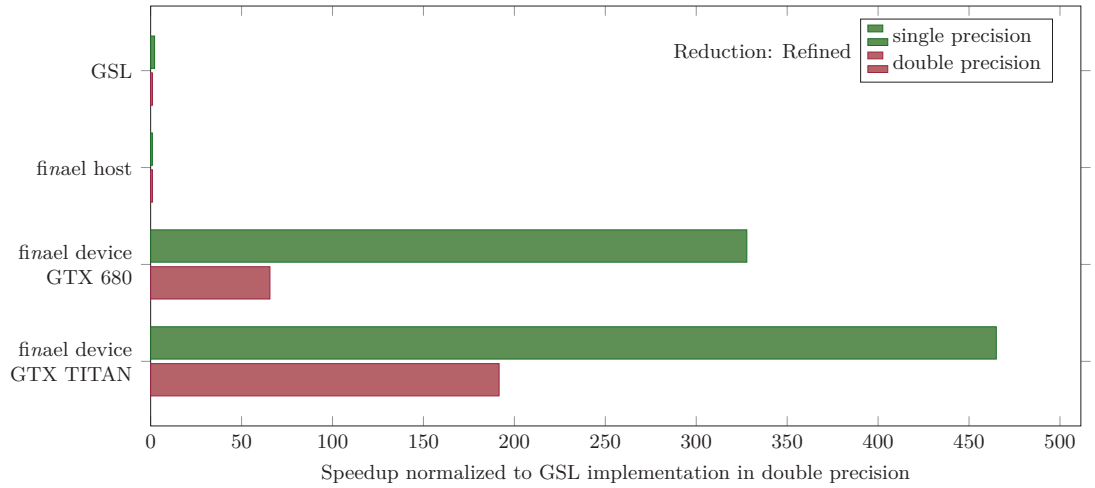


Figure 2.12: Same as figure 2.8, but using the refined reduction for the fi*n*ael implementations.

the reduction, we increase the number of samples points to 10^8. The results of the measurement are shown in figure 2.11.

The host system becomes measurable slower, which can, as expected, be ascribed to the more complicated bin processing using several more divisions, which have a especially large computational cost. On the GPUs however the refined reduction pays off, being more than one order of magnitude faster, despite the bin processing penalty. As we had already seen for the random number generation the otherwise superior GTX TITAN system is slower for the classic reduction as the GTX 680 system due to the slower host processor.

Summarizing we have to record that despite the clear benefit of saved memory the situation is not that easy for the reduction. Its effect on the speed does not only depend on the system but most notably on the specific parameters N and d. If the memory is not the limiting factor the perhaps best policy is to explicitly test all configurations for every use case to determine the fastest possible setup. However, in all cases one will benefit by a more precise result, as the reduction does add values which should potentially be of equal size. In the classic approach it is likely that in a configuration with many samples the values that are added last do not contribute due to the finite precision of floating point numbers.

Concluding we like to examine again the speedup of the setup of the measurements in figure 2.6 and 2.8 using the refined reduction. The results are shown in figure 2.12. Not visible due to the enormous speedup for the GPUs and expected from the previous discussion is the only slight slowdown for the host versions compared to the ones using the classic reduction, which is approximately a half percent. This basically means: Nothing changes, the speed of the host version is identical (up to sub permil level) to the one using the GSL implementation. For the GTX 680 we obtain a improvement of about two percent for double and 16 percent for single precision compared to the classic

reduction. This means for that the speedup compared to the GSL implementation increases to more than 65 for double and more than 325 for single precision. For the GTX TITAN the effects are more pronounced, giving a speedup of about eight percent for double and 27 percent for single precision compared to the versions using the classic reduction. This results in the total speedup against the GSL routine of more than 190 for double and more than 460 for single precision. As we now know this large improvement is due to two effects. First the reduction is much faster on the device. Second less memory has to be copied from the device to the host (100 floating point numbers compared to 10^5 for the classical reduction). Note that the number of samples is still quite small and we expect an even larger effect if it is increased. Note also that this shows again the shift of the completely compute bound problem on the CPU to partly being memory bound on a GPU.

2.5 Towards perturbative QCD applications

The main task to compute perturbative QCD observables using a GPU has definitely been done with the development of a GPU capable version of Vegas presented in the last section. In contrast to earlier studies our implementation does not rely on specializations for a particular device and is designed for direct use without any initial training by offering an easy to use interface. The only restriction is that every user has still to keep in mind that a GPU is used, id est that the integrated function is executable and that memory has to be allocated on the device. For the latter point we developed a tool to manage the memory transfer in a secure and straightforward way, see appendix C.2. The former point is typically not that difficult, as long as the code does not rely on algorithms and classes of the standard library. Note that even this restriction is quite small, as it is only because of the fact that the implementations are not guaranteed to use memory that is not allocated on the device. For most functions however it would in principle be sufficient to declare them to be executable on the device.

So the remaining task is definitely the transformation of legacy codes. These are to a large degree still written in Fortran. We notice that due to the PGI compiler it is nowadays also possible to use Fortran directly on the device. However, this does not give any direct benefit, because also for the PGI compiler the legacy codes had to be rewritten to satisfy the modern requirements and would make them C like anyway. So a clean rebuild in C++ seems to be advantageous, given the strong impression that the future development will mainly be based on C/C++. Fortunately many applications do not rely on large code bases, allowing a fast transformation. This is true up to two obvious exceptions. The first is the use of soft functions such as PDFs or FFs. The groups that offer modern sets are mostly focused on the determination of their sets and do only offer minimal interfaces. This and the fact that every group offers a distinct interface makes this approach depressing. But we a lucky. A much more promising way has shown up by the currently developed C++ library LHAPDF version 6 [157], which

offers access to a large class of sets using one unique interface. If it turns out to be possible to redesign this one library to allow for GPU usage, the whole task would be done. The second exception are evolution codes that evolve the soft functions from one scale to another. In turn of the project described in chapter 5 that is concerned of soft functions in Mellin space, we developed a GPU capable DGLAP evolution version up to next-to-leading order, presented in appendix C.7, which can act as the starting point for an evolution library with a more extensive functionality.

2.6 Conclusions

We have presented a brand new implementation of the Vegas integration algorithm that can be executed on the host as well as on the device. We measured the performance by an toy example integrand with very large computational cost, that simulates a typical integrand that may be used in perturbative QCD calculations. The performance on the host can compete with the current GSL implementation. Using the device outperforms the host version by orders of magnitude. The precise performance is dependent on the actual device and is likely to increase further for newer devices. We presented several optimizations we applied to the naive implementation, which further increased the performance. Finally we achieved a speedup compared to the GSL routine in double precision of about 190 in double and 460 in single precision using a GTX TITAN on a even slower host system as the system that had been used to measure the execution times for the GSL routine. Actually only calculations in double precision matter in high precision calculations, however the single precision performance can be taken to indicate the potential of the GPU, which are nowadays still mainly optimized for single precision. This is likely to change in the future, as GPUs become more and more important for GPGPU, such that specialized cards are manufactured that do not focus on graphics processing, where single precision is sufficient. Taking the development of host CPUs in the last decades as an example, where initially double precision calculations had also been much slower but are now comparably fast respective to single precision calculations, the future of GPGPU seems bright. Having that said, we like to note that the simple statement ‘newer is better’ does not hold on GPUs, as their performance depends mainly how well the given integrand uses the particular hardware specifications. For example we had access to a GeForce GTX 580, which belongs to the compute capability 2.0, for one day during the development. At that stage this otherwise inferior card outperformed the GTX 680 by 35 percent for single and more than 50 percent for double precision, taking the total process time as a measure. Obviously our tool integrand did match the architecture better.

We argued that the main limitations of the GPGPU approach are due to finite memory. Our improvements had been performed mainly to reduce this limitation, improving the performance as a byproduct. Especially the reduction of samples is difficult because of the adaptive nature of Vegas. Our optimizations in this regard can only be applied for stratified sampling, id est if enough samples are used. We discussed

that even then the performance depends heavily on the dimension of the integral. A possible future extension is to optimize the reduction also for increments that contain less bins than the warp size. Another improvement that might be made in the future is to split up the kernels again into smaller parts. As long as all parts do still utilize the device, the performance overhead should not be significant. But this extends the use also to systems with very restricted memory, as the memory use can be controlled by the size of the parts. Besides this opens also the way to use multiple GPUs in parallel, as the parts are independent and can again be parallelized, which further improves the performance.

We briefly discussed how to transform legacy codes to use them also with a GPU, hinting that the only remaining important task seems to be a reasonable portation of the LHAPDF library to be CUDA compatible. Then the typical computational tasks that are encountered in perturbative QCD are a perfect match for using a GPU as they are entirely compute bound. Such the use of GPUs can decrease execution times or increase precision or both, while at the same time reducing the energy cost of the computations.

Chapter 3

Drell-Yan lepton angular distributions in perturbative QCD

We present a comprehensive comparison of the available experimental data for the Drell-Yan lepton angular coefficients λ and ν to calculations at leading and next-to-leading order (NLO) of perturbative QCD. To obtain the NLO corrections, we make use of publicly available numerical codes that allow us to compute the Drell-Yan cross section at second order in perturbation theory and from which the contributions we need can be extracted. Our comparisons show that overall perturbative QCD is able to describe the experimental data rather well, even in the fixed-target regime. There appears to be little evidence for effects that go beyond fixed-order collinear factorized perturbation theory, although the presence of such effects is not ruled out. We also address the recent ATLAS data which show tension with NLO theory for the coefficient ν. All results of this chapter have been published previously [158, 159]. The authors acknowledge support by the state of Baden-Württemberg through bwHPC [160].

3.1 Introduction

It has been known for a long time [161, 162] that leptons produced in the Drell-Yan process $H_1H_2 \to \ell\bar{\ell}X$ may show nontrivial angular distributions. We denote the momentum of the intermediate virtual boson $V \in \{\gamma^*, Z\}$ that decays into the lepton pair by q. In a specific rest frame of the virtual boson (for our purposes, the Collins-Soper frame [161], see figure 3.1) we can define polar and azimuthal lepton decay angles θ and ϕ, respectively. Considering, for simplicity, a situation where contributions by Z-bosons are negligible and only the exchange of an intermediate virtual photon is relevant, one can show that the cross section differential in d^4q and $\mathrm{d}\Omega \equiv \mathrm{d}\cos(\theta)d\phi$ may be written as [163]

$$\frac{\mathrm{d}\sigma}{\mathrm{d}^4q\mathrm{d}\Omega} = \frac{\alpha^2}{2\pi N_c Q^2 s^2}\Big[W_T\big(1+\cos^2(\theta)\big) + W_L\big(1-\cos^2(\theta)\big) + W_\Delta \sin(2\theta)\cos(\phi) + W_{\Delta\Delta}\sin^2(\theta)\cos(2\phi)\Big] \quad (3.1)$$

where α is the fine structure constant, $N_c = 3$ the number of colors in QCD, $Q^2 = q^2$ and s the c.m.s. energy squared of the incoming hadrons H_1 and H_2. The structure

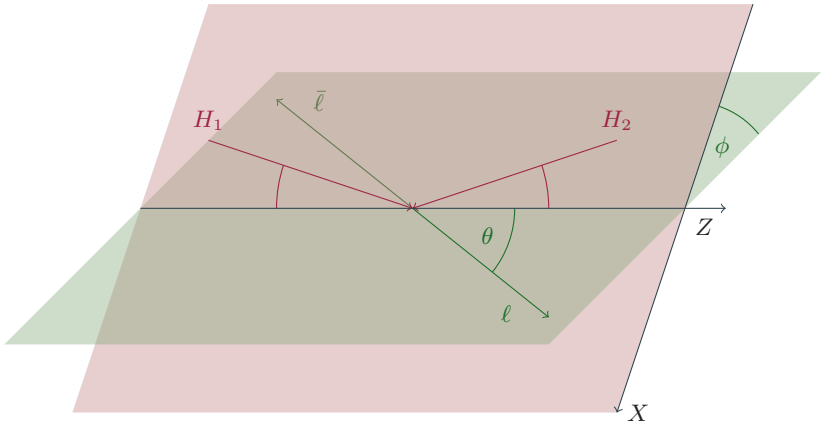


Figure 3.1: Illustration of the Collins-Soper frame, which is a rest frame of the virtual boson. As such, one is free to define two angles to specify the frame. In the Collins-Soper frame this freedom is used to define the Z-axis such, that it bisects the angle between the three-vectors of the two incoming hadrons. The X-axis defines in conjunction with the Z-axis the hadron plane. For non-trivial angular dependence the incoming hadrons are not back-to-back due to initial state radiations.

functions W_T, W_L, W_Δ, $W_{\Delta\Delta}$ are functions of q. They parametrize the hadronic tensor as

$$\begin{aligned} W^{\mu\nu} = &-(g^{\mu\nu} - T^\mu T^\nu)(W_T + W_{\Delta\Delta}) - 2X^\mu X^\nu W_{\Delta\Delta} \\ &+ Z^\mu Z^\nu (W_L - W_T - W_{\Delta\Delta}) - (X^\mu Z^\nu + Z^\mu X^\nu) W_\Delta \end{aligned} \tag{3.2}$$

where X, Y, Z and T are the set of orthonormal axes that one introduces in the Collins-Soper frame. If also Z-bosons contribute, there are additional angular terms and structure functions in the cross section formula. For details of the derivation of the cross section (also for discussion of other related reference frames), see references [161, 162, 164–167].

From the differential cross section one easily derives an expression for the normalized decay angle distribution

$$\frac{\mathrm{d}N}{\mathrm{d}\Omega} \equiv \left(\frac{\mathrm{d}\sigma}{\mathrm{d}^4 q}\right)^{-1} \frac{\mathrm{d}\sigma}{\mathrm{d}\Omega \mathrm{d}^4 q} \tag{3.3}$$

in terms of the structure functions. Using equation (3.1) we obtain

$$\begin{aligned} \frac{\mathrm{d}N}{\mathrm{d}\Omega} = \frac{3}{8\pi} \frac{1}{2W_T + W_L} \Big[&W_T\big(1 + \cos^2(\theta)\big) + W_L\big(1 - \cos^2(\theta)\big) \\ &+ W_\Delta \sin(2\theta)\cos(\phi) + W_{\Delta\Delta} \sin^2(\theta)\cos(2\phi) \Big] \end{aligned} \tag{3.4}$$

One usually writes this as

$$\frac{\mathrm{d}N}{\mathrm{d}\Omega} = \frac{3}{4\pi} \frac{1}{\lambda + 3} \Big[1 + \lambda \cos^2(\theta) + \mu \sin(2\theta)\cos(\phi) + \frac{\nu}{2} \sin^2(\theta)\cos(2\phi) \Big] \tag{3.5}$$

where

$$\lambda = \frac{W_T - W_L}{W_T + W_L} \qquad \mu = \frac{W_\Delta}{W_T + W_L} \qquad \nu = \frac{2W_{\Delta\Delta}}{W_T + W_L} \tag{3.6}$$

Much effort has gone into studies of these angular coefficients λ,μ and ν, both experimentally and theoretically. On the experimental side, measurements of the coefficients are by now available over a wide range of kinematics, from fixed-target energies [168–171] all the way to the Tevatron [172] $p\bar{p}$ and the LHC pp colliders [173, 174]. In the fixed-target regime various combinations of beams and targets are available; data have been taken with pion beams off nuclear (tungsten) targets [168, 169] and also for pp and pd collisions [170, 171]. The experimental results are typically given as functions of the transverse momentum q_T of the virtual boson, in a certain range of the lepton pair mass, $Q \equiv \sqrt{Q^2}$. For the fixed-target data, q_T is limited to a few GeV and Q is usually around 5 to 10 GeV. This is very different for the high-energy collider measurements which are carried out around $Q = m_Z$, where m_Z is the Z-boson mass. The range in q_T explored here is much larger and reaches to almost 100 GeV at the Tevatron and even much beyond that at the LHC.

The lowest-order (LO) partonic channel $q\bar{q} \to V(\to \ell\bar{\ell})$ with collinear incoming partons leads to the prediction $\lambda = 1$, $\mu = \nu = 0$. However, for this process the virtual photon has vanishing transverse momentum, $q_T = 0$, so it cannot contribute to the cross section at finite q_T. The situation changes when 'intrinsic' parton transverse momenta are taken into account. The coefficient ν, especially, which corresponds to a $\cos(2\phi)$ dependence in azimuthal angle, has received a lot of attention in this context since it was discovered [175] that it may probe interesting novel parton distribution functions of the nucleon, known as Boer-Mulders functions [176]. These functions represent a transverse-polarization asymmetry of quarks inside an unpolarized hadron and are 'T-odd' and hence related to nontrivial (re)scattering effects in QCD (see [74, 177, 178]). Detailed phenomenological [179, 180] or model-based [59, 181, 182] studies have been presented that confront the fixed-target experimental data with theoretical expectations based on the Boer-Mulders functions.

Already the early theoretical studies [183–188] revealed that also plain perturbative-QCD radiative effects lead to departures from the simple prediction $\lambda = 1$, $\mu = \nu = 0$, starting from $\mathcal{O}(\alpha_s)$ with the processes $q\bar{q} \to Vg$ and $qg \to Vq$. At $q_T \neq 0$ in fact the latter processes become the LO ones. A venerable result of [162, 189] obtained on the basis of these LO reactions is the *Lam-Tung relation*,

$$1 - \lambda - 2\nu = 0 \tag{3.7}$$

which holds separately for both partonic channels in the Collins-Soper frame [161]. Next-to-leading-order (NLO) corrections to the cross sections relevant for the angular coefficients have first been derived in references [190, 191]. These suggest overall modest $\mathcal{O}(\alpha_s^2)$ effects on λ, μ, ν, so that also the Lam-Tung relation, although found to be violated at NLO, still holds to fairly good approximation. The data from the fixed-target experiment E615 [168] indicate a violation of the Lam-Tung relation, while

the other fixed-target sets are overall consistent with it, as are the Tevatron data [172]. A clear violation of the Lam-Tung relation, on the other hand, was observed recently at the highest energies, in pp collisions at the LHC [173, 174].

In the following sections, we take a fresh look at the Drell-Yan angular dependences in the framework of perturbative QCD. Specifically, we present an exhaustive comparison of the LO and NLO QCD predictions for the parameters λ and ν with the experimental data, over the whole energy range available. Rather than attempting to retrieve the results of [190, 191], we determine new NLO predictions. For this purpose, we use the publicly available codes FEWZ (version 3.1) [192–194] and DYNNLO [195, 196]. These allow us to compute the full Drell-Yan cross section at next-to-next-to-leading (NNLO) order of QCD, when $q\bar{q} \to V$ is the LO process. As discussed above, the contributions to the angular coefficients that we are interested in are at nonvanishing q_T, so that the order α_s^2 in this case is only NLO. Since all $\mathcal{O}(\alpha_s^2)$ contributions are included in the FEWZ and DYNNLO codes, we can therefore use these codes to extract the angular coefficients λ, μ, ν at NLO, providing a new and entirely independent calculation.

To our knowledge, such a comprehensive analysis has never been performed in the past. Our study was very much inspired by the recent work [197], in which the LHC results of the CMS collaboration [173] for the angular coefficients were analyzed on general theoretical grounds, attributing the observed violation of the Lam-Tung relation to a 'noncoplanarity' of the axis of the incoming partons with respect to the hadron plane, which may be constrained by the combined Tevatron and LHC data. As the authors of [197] pointed out, the most likely physical explanation for the CMS result on the violation of the Lam-Tung relation is QCD radiative effects at NLO (or beyond). We indeed confirm this in our study. New results from ATLAS [174], on the other hand, show tension with the theoretical results even at NLO, as will be discussed in section 3.4.

We push the purely perturbative framework also to the fixed-target regime, where there have been hardly any phenomenological analyses of the Drell-Yan angular coefficients in the context of hard-scattering QCD. Reference [198] presents results at the energy of the NA10 experiment; however the kinematics relevant at NA10 was not properly implemented. Of course, in the fixed-target regime q_T can become quite small, smaller than, say, 1 GeV or so. For such low values one does not expect fixed-order perturbation theory to provide reliable results for cross sections, even if Q is relatively large. Intrinsic transverse momenta of the initial partons may become relevant, among them precisely the Boer-Mulders functions mentioned earlier. The possible role of higher-twist contributions has been discussed as well [199–202]. Furthermore, as is well known, large logarithmic perturbative corrections of the form $\alpha_s^k \log^m(Q^2/q_T^2)/q_T^2$, $m \in \{1, \ldots, 2k-1\}$, appear in calculations at fixed perturbative order k, as a result of soft-gluon emission. In order to describe the cross sections, one needs to resum these corrections to all orders in the strong coupling and also implement nonperturbative contributions (see especially [203–208] and references therein or the discussion in chapter 4). As was discussed in references [164–166], such corrections will likely cancel to a significant degree in the angular coefficients λ and ν, since the same type of leading

logarithms occur in the numerator and denominator for both quantities. Also, it is expected [165, 166] that the Lam-Tung relation will remain essentially untouched by the soft-gluon effects.

Thus, although clearly collinear perturbation theory at fixed-order (NLO) that we will use here cannot provide a completely adequate framework for describing cross sections in all kinematic regimes of interest for the angular coefficients, our results to be presented below yield important benchmarks, in our view. In the light of the observations concerning the soft-gluon effects mentioned above, it appears likely that fixed-order perturbation theory will work much better for ratios of cross sections than for the cross sections themselves. In fact, we will find that we can describe most data sets well, and that we do not find any clear-cut evidence for nontrivial additional contributions to be attributed to parton intrinsic momenta. We stress that QCD radiative effects are typically not considered at all when for example Boer-Mulders functions are extracted from data for ν (although the conceptual framework for such a combined analysis is available [209]). At the very least, our results establish the relevance of the radiative effects for phenomenological studies of the Drell-Yan angular dependences.

In section 3.2 we will present the extraction method of the angular coefficients from the available Drell-Yan NNLO codes. Section 3.3 shows the phenomenological result for all fixed-target experiments and for the collider experiments at Tevatron and of the CMS collaboration. Section 3.4 does the same for the ATLAS data and adds some investigations to the coefficient ν and its trend from leading to next-to-leading order. Finally we will conclude our work in section 3.5.

3.2 Extraction of angular coefficients

It is actually relatively straightforward to use the Monte Carlo codes FEWZ [192–194] and DYNNLO [195, 196] to determine the angular coefficients λ, μ and ν. The programs allow us to compute cross sections over suitable ranges of any kinematic variable, providing full control over the four-momenta of the produced particles. As already pointed out in [162], the structure functions W_T, W_L, W_Δ, $W_{\Delta\Delta}$ may be projected out by computing the following combinations of cross sections:

$$2W_T + W_L = \mathcal{N}\frac{\mathrm{d}\sigma}{\mathrm{d}^4q} \tag{3.8a}$$

$$W_T - W_L = \frac{8}{3}\mathcal{N}\left[\frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\left(|\cos(\theta)| > \frac{1}{2}\right) - \frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\left(|\cos(\theta)| < \frac{1}{2}\right)\right] \tag{3.8b}$$

$$W_\Delta = \frac{\pi}{2}\mathcal{N}\left[\frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\left(\sin(2\theta)\cos(\phi) > 0\right) - \frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\left(\sin(2\theta)\cos(\phi) < 0\right)\right] \tag{3.8c}$$

$$W_{\Delta\Delta} = \frac{\pi}{2}\mathcal{N}\left[\frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\left(\cos(2\phi) > 0\right) - \frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\left(\cos(2\phi) < 0\right)\right] \tag{3.8d}$$

where $\mathcal{N} = 12\pi^3(Qs/\alpha)^2$. Using equations (3.6), the angular coefficients follow immediately:

$$\lambda = 12\frac{\frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\Big(|\cos(\theta)| > \frac{1}{2}\Big) - \frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\Big(|\cos(\theta)| < \frac{1}{2}\Big)}{3\frac{\mathrm{d}\sigma}{\mathrm{d}^4q} - 4\Big[\frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\Big(|\cos(\theta)| > \frac{1}{2}\Big) - \frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\Big(|\cos(\theta)| < \frac{1}{2}\Big)\Big]} \tag{3.9a}$$

$$\mu = \frac{9\pi}{4}\frac{\frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\big(\sin(2\theta)\cos(\phi) > 0\big) - \frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\big(\sin(2\theta)\cos(\phi) < 0\big)}{3\frac{\mathrm{d}\sigma}{\mathrm{d}^4q} - 4\Big[\frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\Big(|\cos(\theta)| > \frac{1}{2}\Big) - \frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\Big(|\cos(\theta)| < \frac{1}{2}\Big)\Big]} \tag{3.9b}$$

$$\nu = \frac{9\pi}{2}\frac{\frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\big(\cos(2\phi) > 0\big) - \frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\big(\cos(2\phi) < 0\big)}{3\frac{\mathrm{d}\sigma}{\mathrm{d}^4q} - 4\Big[\frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\Big(|\cos(\theta)| > \frac{1}{2}\Big) - \frac{\mathrm{d}\sigma}{\mathrm{d}^4q}\Big(|\cos(\theta)| < \frac{1}{2}\Big)\Big]} \tag{3.9c}$$

We note that equations (3.8) and therefore also equations (3.9) are valid both for exchanged photons and Z bosons. As mentioned earlier, in cases where Z bosons contribute the cross section has additional angular pieces (see also [173, 197])

$$\begin{aligned}\frac{\mathrm{d}N}{\mathrm{d}\Omega} \sim \Big[&W_T\Big(1+\cos^2(\theta)\Big) + W_L\Big(1-\cos^2(\theta)\Big) + W_\Delta \sin(2\theta)\cos(\phi) \\ &+ W_{\Delta\Delta}\sin^2(\theta)\cos(2\phi) + W_5\sin(\theta)\cos(\phi) + W_6\cos(\theta) \\ &+ W_7\sin^2(\theta)\sin(2\phi) + W_8\sin(2\theta)\sin(\phi) + W_9\sin(\theta)\sin(\phi)\Big]\end{aligned} \tag{3.10}$$

However, it is easy to show, that the angular pieces associated to W_i, $i \in \{5,\ldots,9\}$ do not survive the integrations in equations (3.8) and 3.9, respectively. Thus the presence of Z-bosons introduces no further complication to extract the angular coefficients.

The remaining task is to determine the kinematical variables that appear in equations (3.9) from the momenta of the outgoing leptons given in the Monte Carlo integration codes of [192–196]. To this end, we use that the momentum of one lepton, written in the Collins-Soper frame as (see figure 3.1)

$$\ell^\mu_{\mathrm{cs}} = \frac{Q}{2}\begin{pmatrix}1\\ \sin(\theta)\cos(\phi)\\ \sin(\theta)\sin(\phi)\\ \cos(\theta)\end{pmatrix} \tag{3.11}$$

becomes in the hadronic c.m.s.

$$\ell^\mu_{\mathrm{cm}} = \frac{1}{2}\begin{pmatrix}q_0(1+\sin(\alpha)\sin(\theta)\cos(\phi)) + q_L\cos(\alpha)\cos(\theta)\\ q_T\cos(\varphi) + Q\frac{\sin(\theta)}{\cos(\alpha)}(\cos(\phi)\cos(\varphi) - \cos(\alpha)\sin(\phi)\sin(\varphi))\\ q_T\sin(\varphi) + Q\frac{\sin(\theta)}{\cos(\alpha)}(\cos(\phi)\sin(\varphi) + \cos(\alpha)\sin(\phi)\cos(\varphi))\\ q_L(1+\sin(\alpha)\sin(\theta)\cos(\phi)) + q_0\cos(\alpha)\cos(\theta)\end{pmatrix} \tag{3.12}$$

via the Lorentz boost (see [210] for the special case $\varphi = 0$ or [211] for arbitrary φ but

different parity)

$$\left(B_{\mathrm{cs}\to\mathrm{cm}}\right)^{\mu}{}_{\nu} = \begin{pmatrix} \frac{q_0}{Q} & \frac{q_0}{Q}\sin(\alpha) & 0 & \frac{q_L}{Q}\cos(\alpha) \\ \frac{q_T}{Q}\cos(\varphi) & \frac{\cos(\varphi)}{\cos(\alpha)} & -\sin(\varphi) & 0 \\ \frac{q_T}{Q}\sin(\varphi) & \frac{\cos(\varphi)}{\cos(\alpha)} & \cos(\varphi) & 0 \\ \frac{q_L}{Q} & \frac{q_L}{Q}\sin(\alpha) & 0 & \frac{q_0}{Q}\cos(\alpha) \end{pmatrix} \tag{3.13}$$

where

$$\sin(\alpha) \equiv \frac{q_T}{\sqrt{Q^2+q_T^2}} \qquad \cos(\alpha) \equiv \frac{Q}{\sqrt{Q^2+q_T^2}} \tag{3.14}$$

and where q_0 and q_L are the energy and the longitudinal component (with respect to the collision axis) of the virtual boson in the hadronic c.m.s., so that $q^{\mu}_{\mathrm{cm}} = (q_0, q_T\cos(\varphi), q_T\sin(\varphi), q_L)$. To project out the combinations of trigonometric functions in equations (3.8), we introduce

$$\mathcal{P}_1^{\mu} \equiv \begin{pmatrix} q_L \\ 0 \\ 0 \\ q_0 \end{pmatrix} \qquad \mathcal{P}_2^{\mu} \equiv q_T \begin{pmatrix} 0 \\ \cos(\varphi) \\ \sin(\varphi) \\ 0 \end{pmatrix} \qquad \mathcal{P}_3^{\mu} \equiv q_T \begin{pmatrix} 0 \\ \sin(\varphi) \\ -\cos(\varphi) \\ 0 \end{pmatrix} \tag{3.15}$$

With these we can calculate the trigonometric functions by

$$\cos(\theta) = -\frac{2\ell_{\mathrm{cm}}\cdot\mathcal{P}_1}{(Q^2+q_T^2)\cos(\alpha)} \tag{3.16a}$$

$$\sin(2\theta)\cos(\phi) = \frac{4\ell_{\mathrm{cm}}\cdot\mathcal{P}_1}{Q^2+q_T^2}\left[\frac{q_T}{Q} + \frac{2\ell_{\mathrm{cm}}\cdot\mathcal{P}_2}{q_T Q}\right] \tag{3.16b}$$

$$\cos(2\phi) = 1 - \frac{2}{q_T^2}(\ell_{\mathrm{cm}}\cdot\mathcal{P}_3)^2\left[\frac{Q^2}{4} - \frac{(\ell_{\mathrm{cm}}\cdot\mathcal{P}_1)^2}{Q^2+q_T^2}\right]^{-1} \tag{3.16c}$$

The four-momentum of the lepton in the hadronic c.m.s. is provided in the Monte Carlo integration codes, while that of the virtual boson is fixed by the external kinematics. Writing $\ell^{\mu}_{\mathrm{cm}} = (\ell^0_{\mathrm{cm}}, \ell^1_{\mathrm{cm}}, \ell^2_{\mathrm{cm}}, \ell^3_{\mathrm{cm}})$, we have

$$\ell_{\mathrm{cm}}\cdot\mathcal{P}_1 = q_L\ell^0_{\mathrm{cm}} - q_0\ell^3_{\mathrm{cm}} \tag{3.17a}$$

$$\ell_{\mathrm{cm}}\cdot\mathcal{P}_2 = -q_T(\ell^1_{\mathrm{cm}}\cos(\varphi) + \ell^2_{\mathrm{cm}}\sin(\varphi)) \tag{3.17b}$$

$$\ell_{\mathrm{cm}}\cdot\mathcal{P}_3 = q_T(\ell^2_{\mathrm{cm}}\cos(\varphi) - \ell^1_{\mathrm{cm}}\sin(\varphi)) \tag{3.17c}$$

Inserting these expressions into equations (3.16), one can now easily implement the appropriate cuts in the codes so that the structure functions W_T, W_L, W_Δ, $W_{\Delta\Delta}$ and the angular coefficients λ, μ, ν can be extracted via equations (3.8) and (3.9), respectively. We will present only the angular coefficients, because they do not depend on the lepton pair mass Q. Since the experiments always integrate over a whole range of lepton pair masses, one had to take different Q into account to calculate the structure functions.

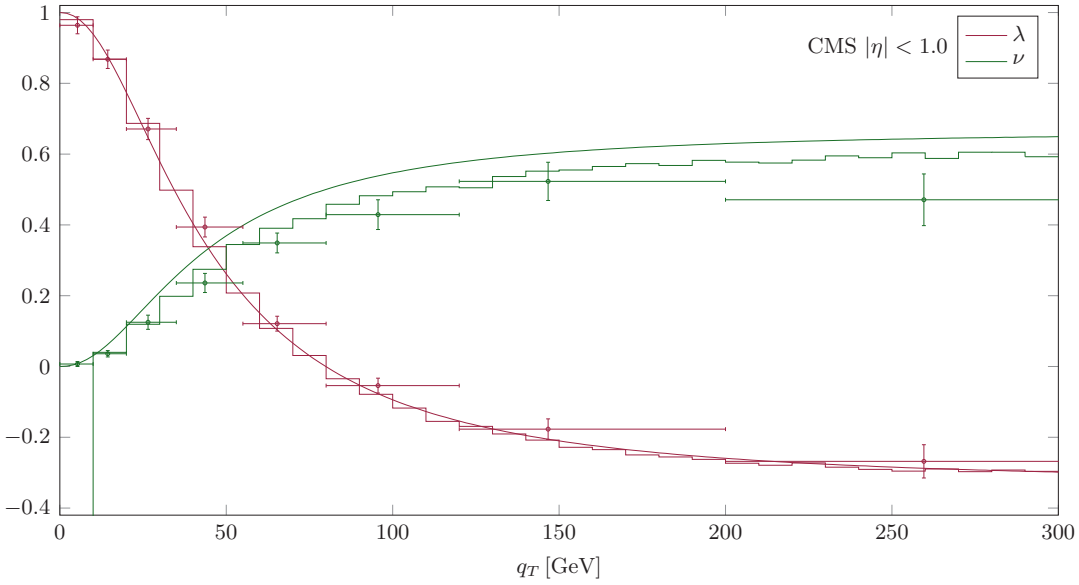


Figure 3.2: Comparison of LO (lines) and NLO (FEWZ [192–194], histograms) theoretical results to the CMS data [173] for the angular coefficients λ and ν taken at $\sqrt{s} = 8\,\text{TeV}$. We have integrated over $81 \leq Q \leq 101\,\text{GeV}$ and over a central rapidity interval $|\eta| < 1$ of the virtual boson.

3.3 Comparison to data

We now present comparisons of the theoretical predictions at LO and NLO to the available experimental data for the angular coefficients λ and ν. We do not show any results for the coefficient μ which comes out always extremely small and in fact usually consistent with zero both in the theoretical calculation and in experiment, within the respective uncertainties. We first note that we have validated our technique for extracting the Drell-Yan angular coefficients from the FEWZ (version 3.1) [192–194] and DYNNLO [195, 196] codes by writing a completely independent LO code. We have found perfect agreement between this code and the LO results we extracted from FEWZ and DYNNLO. In the figures below, the LO curves will always refer to those from our own code. We also note that the NLO results we show in the following have all been obtained with the FEWZ code. We have compared to the results of DYNNLO and found excellent consistency of the two codes both at LO and NLO.

Although the implementation of equations (3.8) and the relevant kinematics into the FEWZ or DYNNLO codes is relatively straightforward, the computational load for performing a comprehensive comparison of the data with NLO theory is very large. To obtain the NLO results presented in this a the following section, we have run an equivalent of one 3.20 GHz Intel Quad-Core i5-3470 CPU using all of its cores for more than two and a half years.[1]. In order to collect sufficiently high statistics at

[1]The reported time is a very rough extrapolation. In fact we used multiple desktop machines and the

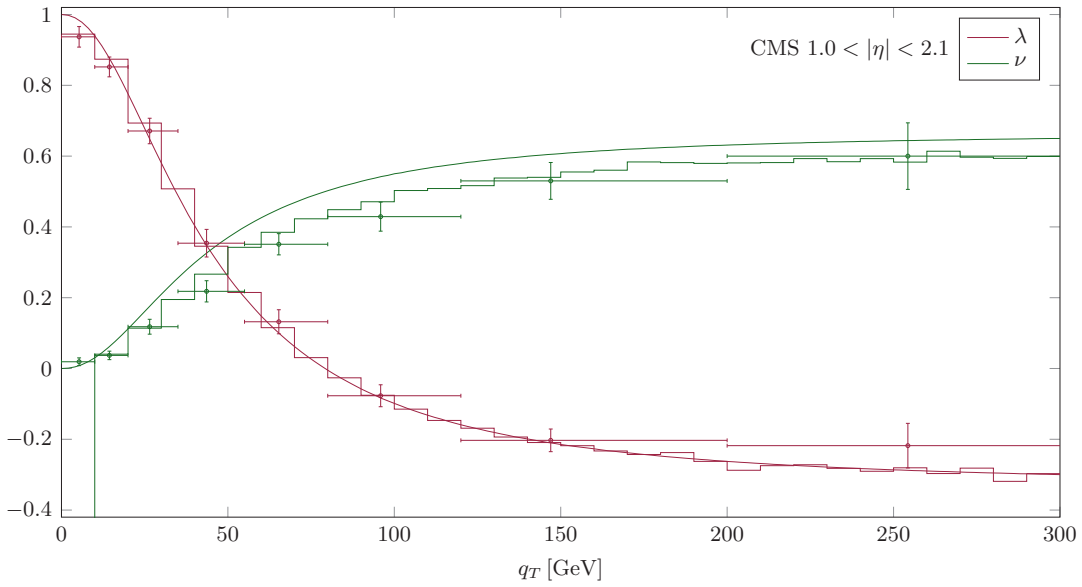


Figure 3.3: Same as figure 3.2, but for a more forward/backward rapidity interval $1 < |\eta| < 2.1$.

very high values of q_T, where the cross section drops very rapidly, we have performed dedicated runs for which we have implemented cuts on the low-q_T region, forcing the Monte Carlo integration to sample high q_T. We also note that typically the result for the lowest-q_T bin is unreliable, since this bin contains the (NNLO) contributions at $q_T = 0$. Nonetheless, our results are sufficiently accurate in all regions of interest and thus allow us to derive solid conclusions. We mention that we also had to modify the codes to accommodate pion beams and nuclear (deuteron/tungsten) targets. This implementation was always checked against our own LO code.

Throughout this paper, we use the parton distribution functions of the proton of reference [89], adopting their NLO (LO) set for the NLO (LO) calculation. The choice of parton distributions has a very small effect on the Drell-Yan angular coefficients. When dealing with nuclear targets (tungsten was used for all of the pion scattering experiments and deuterons for one set of E866 measurements) we compute the parton distributions of the nucleus just by considering the relevant isospin relations for protons and neutrons, averaging over the appropriate proton and neutron number. We do not add any other nuclear effects. For the parton distributions of the pion, we use the set in [212]; the set in [213] would give very similar results. Finally, our choice for the factorization and renormalization scales will always be $\mu = Q$. We have checked that other possible scale choices such as $\mu = \sqrt{Q^2 + q_T^2}$ do not change the results for the angular coefficients significantly even at LO, making an impact of at most a

bwHPC cluster [160] The time estimate is based on the time average of the calculations performed on desktop machines equipped with one 3.20 GHz Intel Quad-Core i5-3470 CPU. Then we assume that all calculations have been performed on this machines.

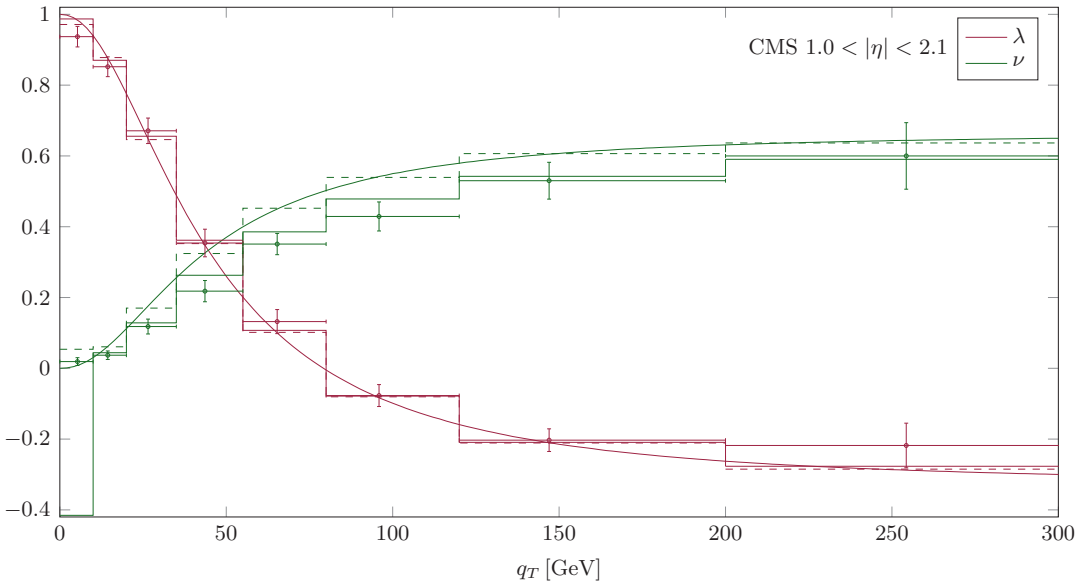


Figure 3.4: Same as figure 3.3, but with the NLO theoretical results integrated over the eight q_T bins used by CMS. In this figure, the dashed histograms show the LO results and the solid ones the NLO results. To guide the eye, we also show the LO results from figure 3.3 as smooth lines.

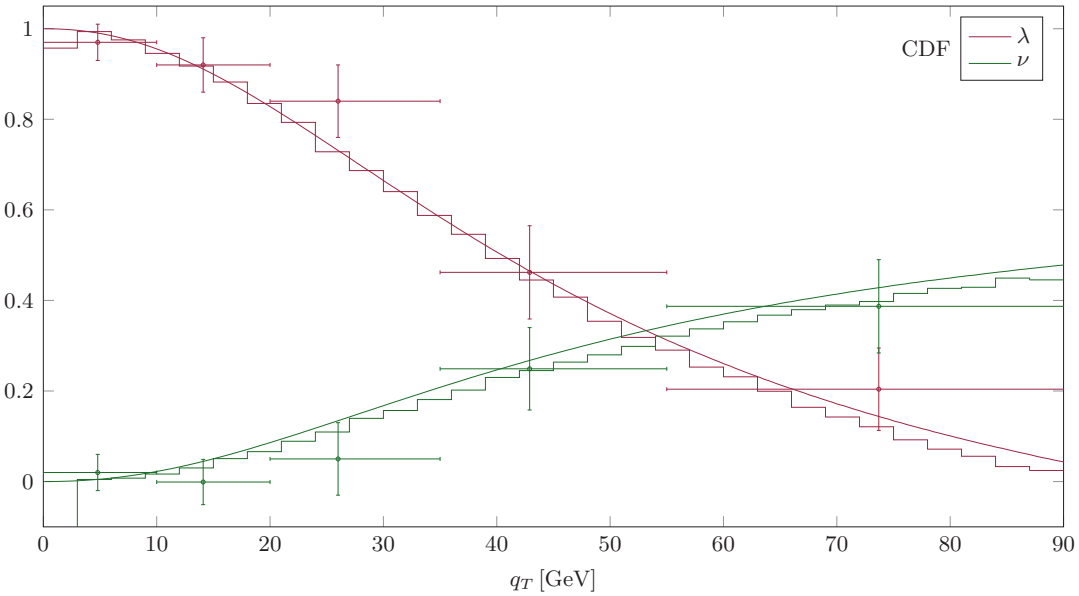


Figure 3.5: Comparison of LO (lines) and NLO (FEWZ [192–194], histograms) theoretical results to the CDF data [172] for the angular coefficients λ and ν taken in $p\bar{p}$ scattering at $\sqrt{s} = 1960\,\text{GeV}$. We have integrated over $66 \leq Q \leq 116\,\text{GeV}$ and over $|\eta| < 3.6$ of the virtual boson.

few percent, and only at high values of q_T. Here we have simultaneously varied the scales in the cross sections appearing in the numerators *and* in the denominators of the angular coefficients; relaxing this condition one would likely be able to generate a larger dependence on the choice of scale. On the other hand, as is known from previous calculations [192–196], the scale dependence of the Drell-Yan cross section is overall much reduced at higher orders anyway.

We present our results essentially in the order of decreasing energy, starting with a comparison to the high-energy collider data from the CMS experiment [173] and Tevatron [172]. The data of the ATLAS experiment [174] will be accompanied by a more detailed analysis, which is therefore postponed to section 3.4. The reason for this ordering is that for these data sets Q is very large, $Q \approx m_Z$, so that perturbative methods should be well justified. The transverse momentum q_T varies over a broad range, taking low values as well as values of order Q. At the lower end, where $q_T \ll Q$, it may well be necessary to perform an all-order resummation of perturbative double logarithms in q_T/Q in order to describe the Drell-Yan cross section properly. However, as mentioned in the introduction (section 3.1), such logarithms are expected to cancel to a large extent in the angular coefficients [164–166]. Thus, if ever fixed-order perturbative QCD predictions are able to provide an adequate description of the angular coefficients, it should be in the kinematic regimes explored at the LHC and Tevatron.

Figures 3.2 and 3.3 show our results for λ and ν compared to the CMS data [173], for two separate bins in the rapidity of the virtual boson,

$$\eta \equiv \frac{1}{2} \log\Big(\frac{q_0 + q_L}{q_0 - q_L}\Big) \tag{3.18}$$

Additionally for all plots we applied a kinematical cut to the lepton pair mass, which is close to the Z-boson mass: $81 \leq Q \leq 101\,\text{GeV}$ [214]. We assume that the same cuts have been applied in [173]. Further we note that CMS presents their data in terms of a different set of angular coefficients termed A_0, A_1, A_2, A_3, which are directly related to the coefficients we use here. In particular, we have

$$\lambda = \frac{2 - 3A_0}{2 + A_0} \qquad \text{and} \qquad \nu = \frac{2A_2}{2 + A_0} \tag{3.19}$$

As in reference [197], in order to present a full comparison in terms of λ and ν, we transform the experimental data correspondingly. Here we have propagated the experimental uncertainties, albeit without taking into account any correlations. This might lead to an overestimation of the experimental errors. The lines in the figures show our LO results for the coefficients. As one can see, they qualitatively follow the trend of the data, but for the coefficient ν a clear deviation between data and LO theory is observed. This is precisely the finding also emphasized in reference [197] where it was argued (without explicit NLO calculation) that the discrepancy ought to be related to higher-order QCD effects. Indeed, this is what we find. The NLO results (histograms) show a markedly better agreement with the data, which in fact is nearly perfect. The coefficient λ, on the other hand, changes only marginally from LO to NLO. As is visible

in the figures, the results at very high values of q_T are numerically less accurate, as shown by the somewhat erratic behavior of the histograms. In order to collect higher statistics, we have also performed runs for which we integrated over only eight q_T bins, choosing exactly the ones used in the experimental analysis. The corresponding results are shown in figure 3.4 for the rapidity range $1 < |\eta| < 2.1$. Our goal was to make sure that the numerical uncertainty for these bins is much smaller than the experimental one *even* in the bin at highest q_T. The figure once more impressively shows how NLO theory leads to an excellent description of the CMS data.

It is interesting to note that NLO FEWZ results were also shown in the CMS paper [173]. However, the agreement with the data for the coefficient A_2 (which multiplies the $\cos(2\phi)$ dependence of the cross section) reported there appears to be not quite as good as the one we find for our coefficient ν. It is conceivable that our computation of the coefficients via equations. (3.8) is numerically more stable.[2]

We next turn to the comparison to the CDF data [172] taken in $p\bar{p}$ collisions at $\sqrt{s} = 1960\,\text{GeV}$ at the Tevatron with lepton pair masses of $66 \leq Q \leq 116\,\text{GeV}$ and a allowed rapidity interval of $|\eta| < 3.6$. The results are shown in figure 3.5. We observe that both the LO and the NLO results are in good agreement with the data, NLO doing a bit better overall. Both coefficients λ and ν decrease slightly when going to NLO. For ν, this effect is less pronounced than for the LHC case, which may be attributed to a much stronger contribution by the $q\bar{q}$ channel in the present $p\bar{p}$ case, which receives smaller radiative corrections. Again, this feature was predicted phenomenologically in reference [197].

We now consider the fixed-target regime, where we start with a comparison to the Fermilab E866/NuSea data taken with an 800 GeV proton beam in pp [171] and pd [170] scattering. The data are taken with lepton pair masses of $4.5 \leq Q \leq 15\,\text{GeV}$ excluding the bottonium region $9 \leq Q \leq 10.7\,\text{GeV}$ and with the kinematical cut $0 \leq x_F \leq 0.8$, with the Feynman variable x_F counted as positive in the forward direction of the proton beam:

$$x_F = \frac{2q_L}{\sqrt{s}} \tag{3.20}$$

The comparisons to the two data sets are shown in figures 3.6 and 3.7. We first note that the pp data are overall in much better agreement with the theoretical curves than the pd ones. For pp scattering, the coefficient λ is well described, given the relatively large experimental uncertainties. There is a slight trend in the data for the coefficient ν to be lower than the theoretical prediction. The NLO corrections in fact provide a slight improvement here. For pd scattering, the two data points for ν at the highest q_T are clearly below theory even at NLO. The coefficient λ is not well described, neither at LO nor at NLO. An important point to note in this context is the positivity constraint [162]

$$W_L \geq 0 \tag{3.21}$$

[2] FEWZ itself provides the angular coefficients A_i (see equations (3.19)). Unfortunately this coefficients do not fulfill the Lam-Tung relation at leading order, where the relation holds exactly. Again it is likely that our computation of the coefficients is more stable. However, it is not specified in [173] how exactly the coefficients are deduced and which version of FEWZ is used.

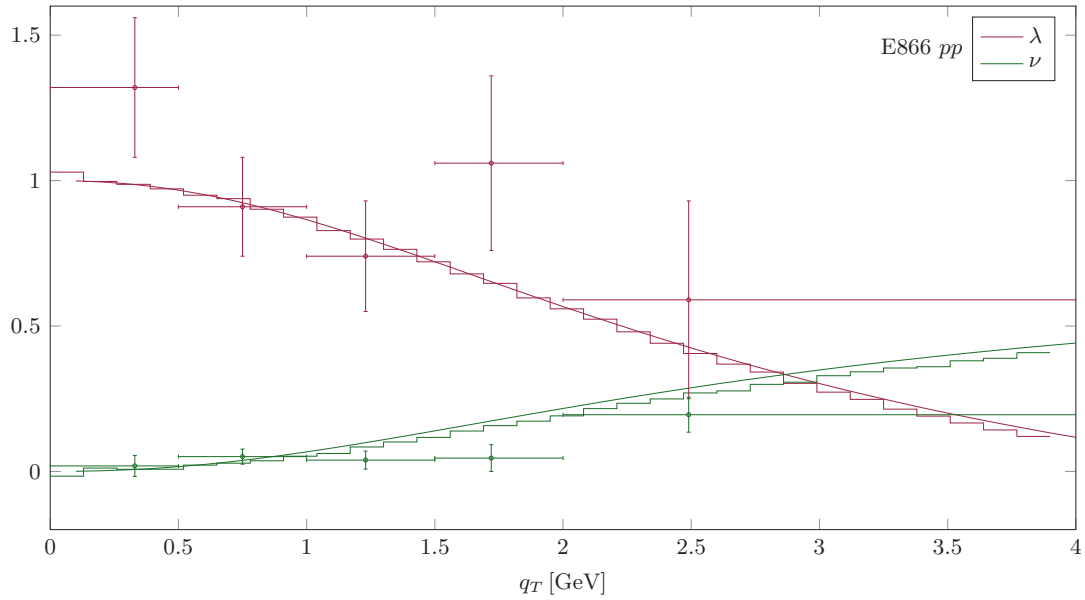


Figure 3.6: Comparison of LO (lines) and NLO (FEWZ [192–194], histograms) theoretical results to the pp scattering data from E866 [171] for the angular coefficients λ and ν taken with an 800 GeV beam. Error bars are statistical only. We have integrated over the mass range $4.5 \leq Q \leq 15\,\text{GeV}$, excluding the bottomonium region $9 \leq Q \leq 10.7\,\text{GeV}$. We have also integrated over $0 \leq x_F \leq 0.8$, see equation (3.20).

which immediately implies

$$\lambda \leq 1 \tag{3.22}$$

This condition is completely general and relies only on the hermiticity of the neutral current. It is interesting to observe that the pd data shown in figure 3.6 are only in borderline agreement with this positivity constraint.

Going further down in energy, we finally discuss the data from the π + tungsten scattering experiments NA10 [168] and E615 [169]. NA10 used three different energies for the incident pions, $E_\pi = 286$, 194 and 140 GeV, while E615 operated a pion beam with energy 252 GeV. The NA10 experiments use a lower cut on the lepton pair mass of $Q \geq 4\,\text{GeV}$ (except for the pion beam with 194 GeV, where the mass is $Q \geq 4.05\,\text{GeV}$) excluding the bottonium region $8.5 \leq Q \leq 11\,\text{GeV}$. Furthermore a cut $0 \leq x_\pi \leq 0.7$ is applied, where

$$x_\pi = \frac{1}{2}\left(x_F + \sqrt{x_F^2 + \frac{4Q^2}{s}}\right) \tag{3.23}$$

and x_F is again the Feynman variable in equation (3.20), which is counted positive in the forward direction of the pion beam. E615 uses the lepton pair masses $4.05 \leq Q \leq 8.55\,\text{GeV}$ and applies the additional cuts $0 \leq x_F \leq 1$ and $0.2 \leq x_\pi \leq 1$, where the definitions are the same as for NA10.

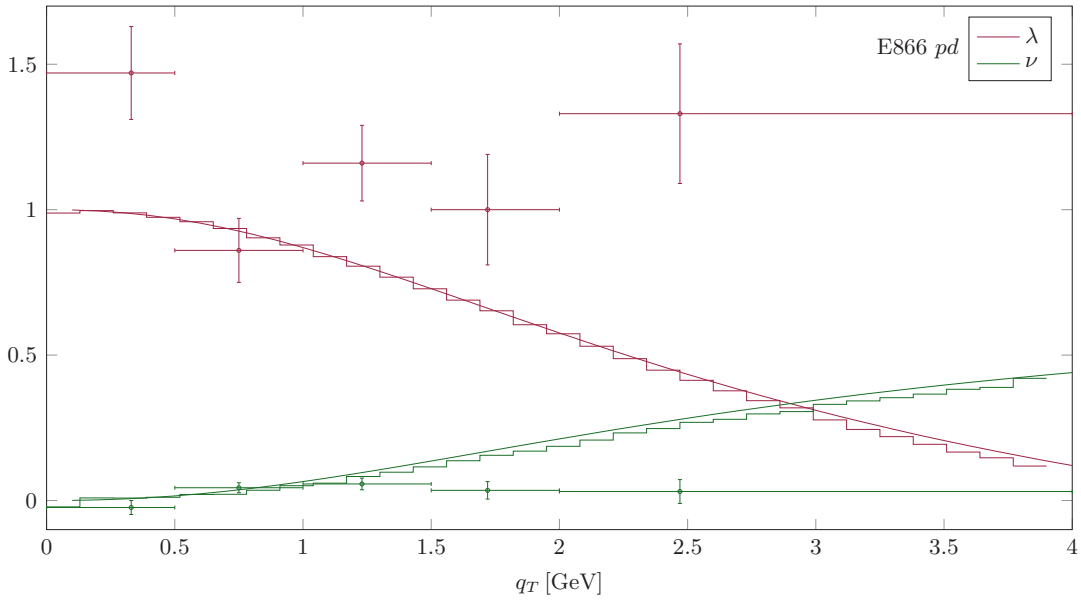


Figure 3.7: Same as figure 3.6, but for *pd* scattering. Data are from reference [170]

Figures 3.8, 3.9 and 3.10 show the comparisons of our LO and NLO results for λ and ν to the NA10 data. The NLO corrections are overall small for ν, but for λ they become more pronounced toward larger q_T. We note that NLO results for one of the NA10 energies were also reported in reference [198], where however not the appropriate kinematical regime in Q was chosen, leading to an underestimate of ν which has unfortunately given rise to the general notion in the literature that perturbative QCD cannot describe the Drell-Yan angular coefficients. We also note that for the kinematics used in [198] the NLO corrections appear to be somewhat smaller than the ones we find here. The three cases shown in figures 3.8, 3.9 and 3.10 have in common that the data for ν are well described, perhaps slightly less so for the pion energy 194 GeV. The experimental uncertainties for the coefficient λ are very large, and it is not possible to draw solid conclusions from the comparison. We note that wherever there are tensions between data and theory concerning λ, the data tend to lie uncomfortably close to (or even above) the positivity constraint $\lambda \leq 1$.

In case of E615, we find the results shown in figure 3.11. We observe that neither the description of λ nor that of ν is good. The NLO corrections are overall small and thus do not change this picture. It is clear that on the basis of the data one would derive a significant violation of the Lam-Tung relation (3.7), since λ and ν both enter the relation with the same sign, and the data for both λ and ν are higher than theory (the latter satisfying the relation at LO). It is worth pointing out, however, that the experimental uncertainties are large and, more importantly, again the data show a certain tension with respect to the positivity limit (3.22).

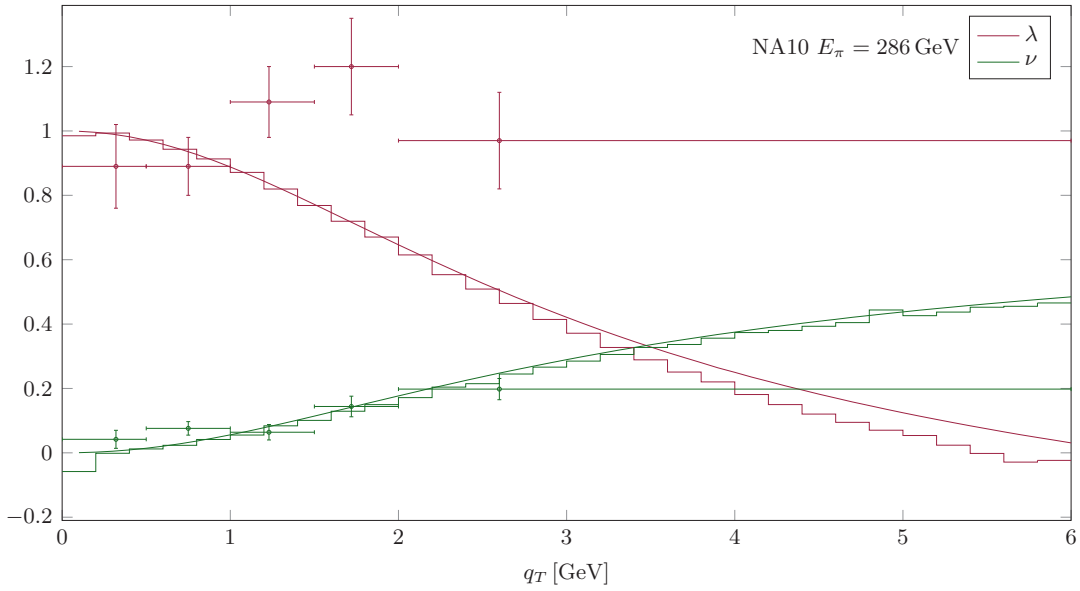


Figure 3.8: Comparison of LO (lines) and NLO (FEWZ [192–194], histograms) theoretical results of the angular coefficients λ and ν to the π+tungsten scattering data from NA10 [168] taken with pion beam energy $E_\pi = 286\,\text{GeV}$. Error bars are statistical only. We have integrated over the mass range $Q \geq 4\,\text{GeV}$, excluding the bottomonium region $8.5 \leq Q \leq 11\,\text{GeV}$. We have also implemented the cut $0 \leq x_\pi \leq 0.7$, see equation (3.23).

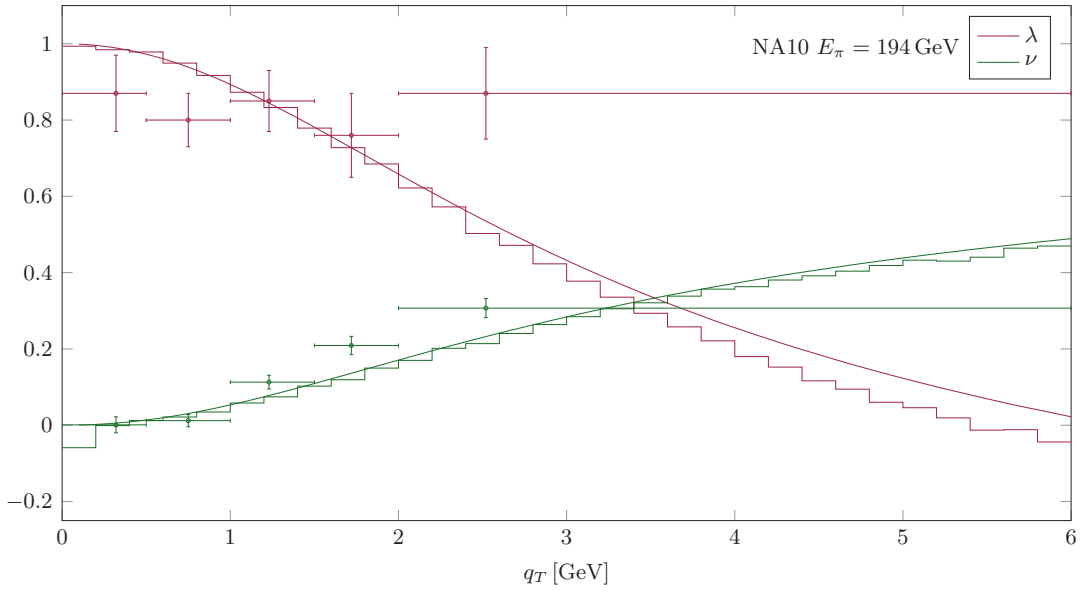


Figure 3.9: Same as figure 3.8, but at pion energy $E_\pi = 194\,\text{GeV}$ and integrated over $Q \geq 4.05\,\text{GeV}$.

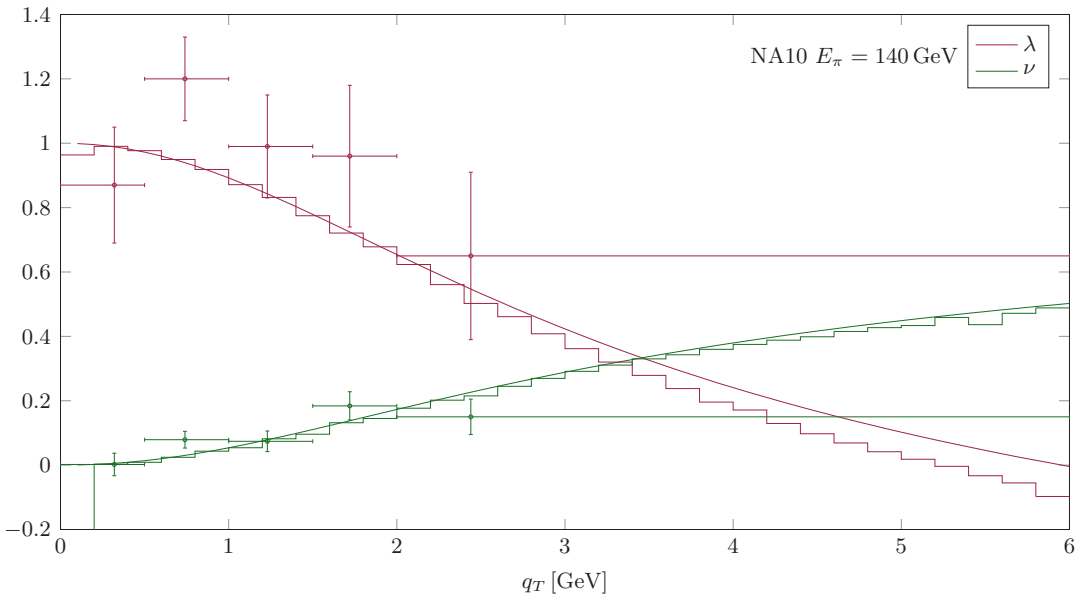


Figure 3.10: Same as figure 3.8, but at pion energy $E_\pi = 140\,\mathrm{GeV}$.

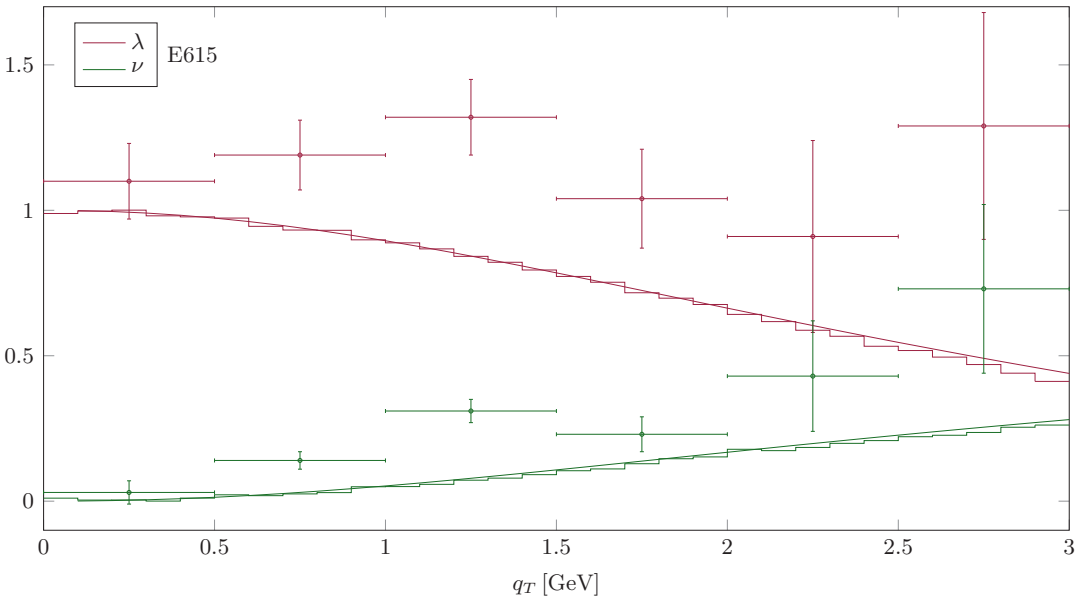


Figure 3.11: Comparison of LO (lines) and NLO (FEWZ [192–194], histograms) theoretical results of the angular coefficients λ and ν to the π+tungsten scattering data from E615 [169] taken with pion beam energy $E_\pi = 252\,\mathrm{GeV}$. We have integrated over the mass range $4.05 \leq Q \leq 8.55\,\mathrm{GeV}$. We have also implemented the cuts $0 \leq x_F \leq 1$ and $0.2 \leq x_\pi \leq 1$, see equations(3.20) and (3.23).

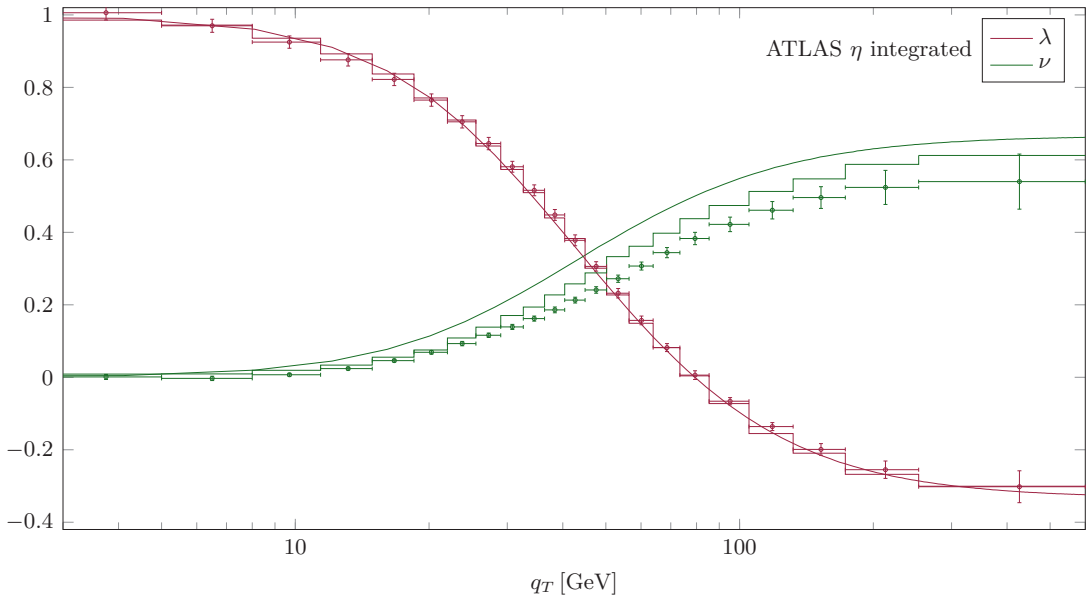


Figure 3.12: Comparison of LO (lines) and NLO (FEWZ [192–194], histograms) theoretical results for the angular coefficients λ and ν to the ATLAS data [174]. The pp scattering data are taken at $\sqrt{s} = 8\,\text{TeV}$, integrated over the mass range $80 \leq Q \leq 100\,\text{GeV}$ of the produces lepton pair. Further we integrated for the theoretical predictions over the rapidity $|\eta| < 8$ also of the virtual boson.

3.4 ATLAS data analysis

In summer 2016 the ATLAS Collaboration published new data on the angular coefficients at $\sqrt{s} = 8\,\text{TeV}$ for Z-Boson events [174]. The uncertainties quoted by ATLAS are overall much smaller than the CMS ones [173]. The results are shown in figures 3.12, 3.13, 3.14 and 3.15. ATLAS provides the angular coefficients just as CMS in terms of A_1, A_2, A_3 and A_4. We transformed them accordingly to equations (3.19). As for the CMS data we propagated the experimental errors without knowledge of present correlations. As a consequence the errors in our plots might overestimate the real error. For the coefficient λ the data confirm theory both in leading and in next-to-leading order which show little difference. This is different for the coefficient ν. Unlike for the CMS data, even NLO fails to describe the data well, as is evident from figures 3.12, 3.13 and 3.14, although next-to-leading order makes a essential improvement towards the data. The agreement is reasonable for the extremely forward/backward rapidity bin shown in figure 3.15, where however the experimental uncertainties are large.

The tension between theory and data observed in the figures requires further attention. It appears unlikely that effects related to the logarithms in q_T/Q mentioned above or to transverse-momentum dependent parton distributions would have any impact on the predicted ν at transverse momenta of the order of a few tens of GeV or even above. If the data persist, NNLO effects would arguably offer the only viable explanation for

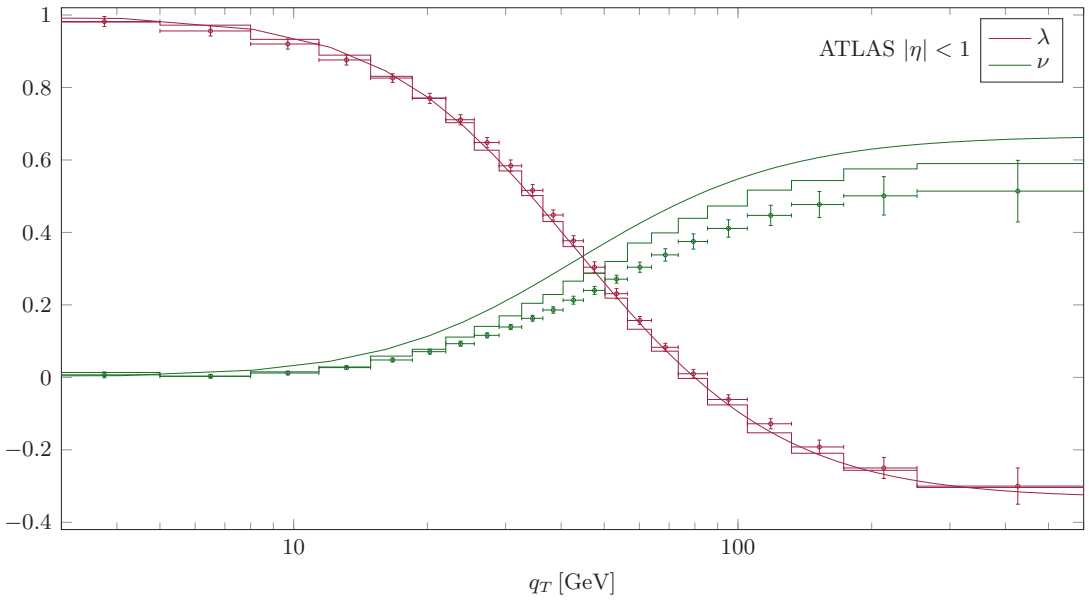


Figure 3.13: Same as figure 3.12, but integrated over a central rapidity interval $|\eta| < 1$ of the virtual boson only.

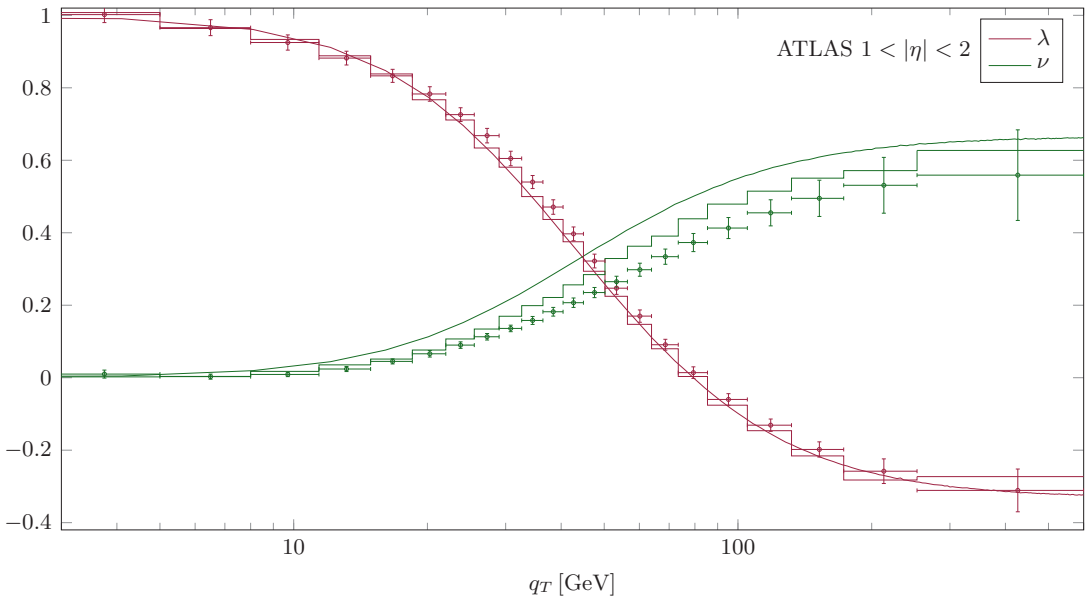


Figure 3.14: Same as figure 3.12, but integrated over a forward/backward rapidity interval $1 < |\eta| < 2$ of the virtual boson only.

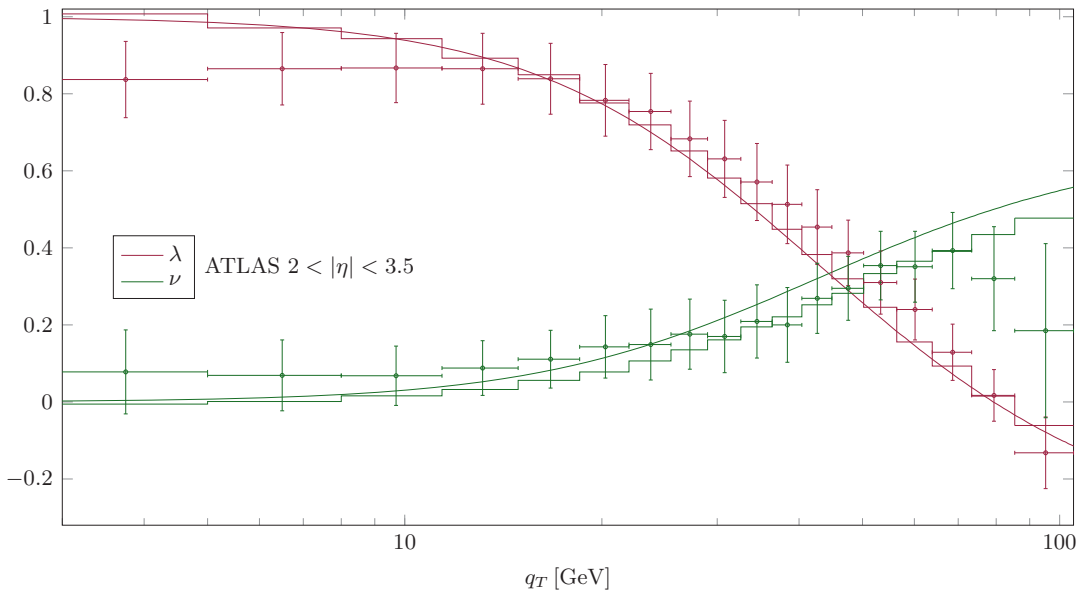


Figure 3.15: Same as figure 3.12, but integrated over an extreme forward/backward rapidity interval $2 < |\eta| < 3.5$ of the virtual boson only.

the observed trend. Fortunately, NNLO predictions are expected to become available very soon [215, 216]. In the meantime, we investigate some features of the NLO results. Figure 3.16 shows the 'K-factors' relevant for the coefficient ν, that is,

$$K_{TL} \equiv \frac{(W_T + W_L)_{\mathrm{NLO}}}{(W_T + W_L)_{\mathrm{LO}}} \tag{3.24a}$$

$$K_{\Delta\Delta} \equiv \frac{(W_{\Delta\Delta})_{\mathrm{NLO}}}{(W_{\Delta\Delta})_{\mathrm{LO}}} \tag{3.24b}$$

As one can see, the K-factor is much bigger for the denominator of ν than for the numerator. If this trend continues at the next order, a smaller ν will result, which will be in better agreement with the data. We note, however, that the additional NNLO contributions would have to be quite large in order to lead to a good description of the data. Figure 3.17 shows the different channels contributing to the numerator of ν. We distinguish among the gg, qg and $q\bar{q} + qq$ subprocesses, where in the latter case all possible flavour contributions are included. It is straightforward to extract the subprocess contributions gg and $q\bar{q} + qq$ by suitably setting the quark or gluon PDFs to zero. The channel qg is taken as the difference of the full result and all other channels. We always keep all channels in the denominator of ν. In figure 3.17 it can be clearly seen, that from LO to NLO all channels receive corrections that drive the result closer towards the data. The qg channel is the major contributor and decreases considerably at NLO. The $q\bar{q} + qq$ contribution becomes smaller in the q_T regime of interest, despite the fact that a lot of more initial states are possible at NLO. Finally the gg channel,

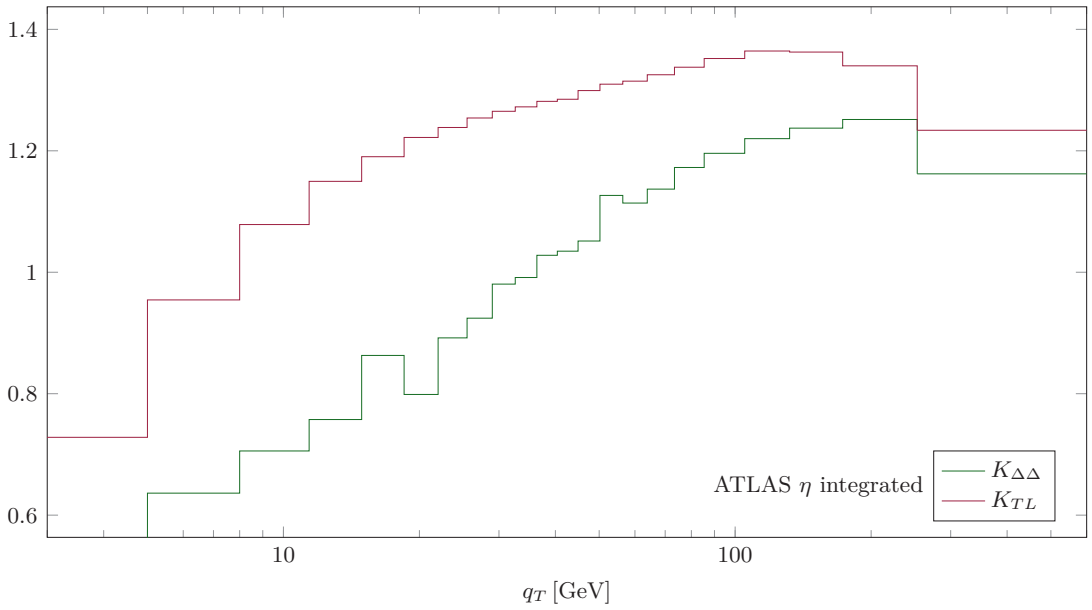


Figure 3.16: K-factors defined in equations (3.24). LO and NLO are calculated by FEWZ [192–194]. The kinematic regime is identical to figure 3.12.

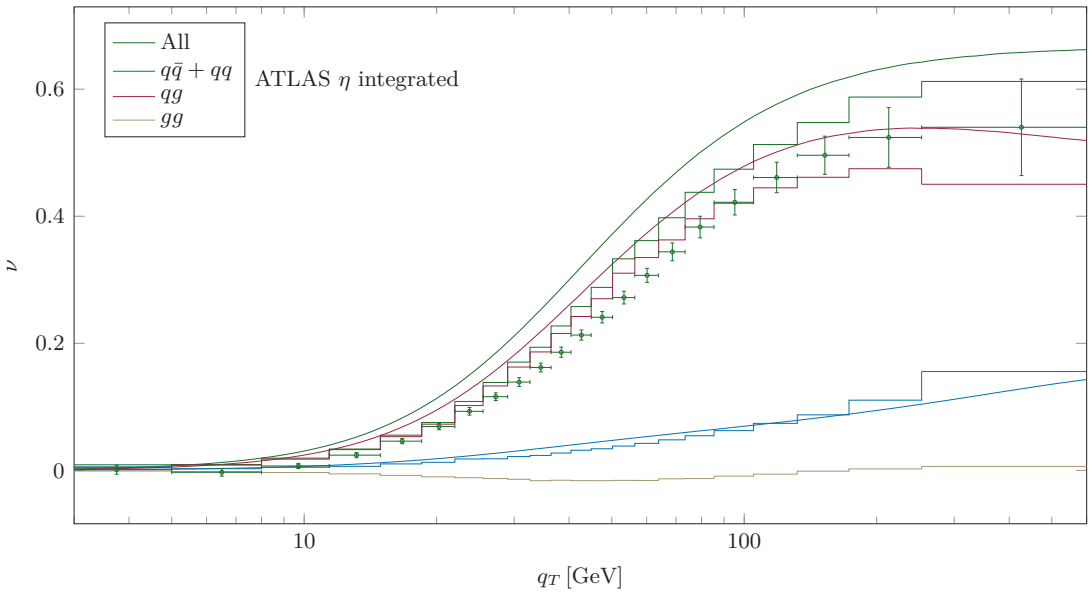


Figure 3.17: Illustration of the contributions by the different partonic channels contributing to the angular coefficient ν. Lines are LO, histograms NLO. Latter are computed by FEWZ [192–194]. To calculate the gg and the $q\bar{q} + qq$ channels all quark PDFs and the gluon PDF are set to zero respectively. The qg channel is calculated as difference of the full result and the gluon-only and quarks-only channels. The denominator of ν is always calculated with all channels enabled. The kinematic regime is identical to figure 3.12.

which is new at NLO, contributes negatively, lowering the overall ν.

3.5 Conclusions

We have presented detailed and exhaustive comparisons of data for the Drell-Yan lepton angular coefficients λ and ν to LO and NLO perturbative-QCD calculations. To obtain NLO results, we have employed public codes that allow us to compute the full Drell-Yan cross section at NNLO, and in which the angular pieces we are interested in are contained.

Our numerical results show that overall perturbative QCD is able to describe the experimental data quite well, except for the new ATLAS data. For the CMS data the agreement is very good, when the NLO corrections are taken into account. This finding is in line with arguments made in the recent literature [197]. Also the Tevatron data are very well described at NLO. Toward the fixed-target regime, we again find an overall good agreement, with possible exceptions for the E866 *pd* data set for ν at high q_T and for the E615 data. We remark that the latter data sets carry large uncertainties and also hint at tensions with the positivity constraint $\lambda \leq 1$.

To be sure, the description of the cross sections that enter the angular coefficients requires input beyond fixed-order QCD perturbation theory, notably in terms of resummations of logarithms in q_T/Q and of transverse-momentum dependent parton distributions. On the other hand, based on the angular coefficients alone, in our view there is no convincing evidence for any effects other than the ones we have considered here. In particular, we argue that one should dispel the myth that perturbative QCD is not able to describe the Drell-Yan angular coefficients, which in fact has been iterated over and over in the literature. While we most certainly do not wish to exclude the presence of contributions by the Boer-Mulders effect in the $\cos(2\phi)$ part of the angular distribution, it is also clear from our study that future phenomenological studies of the effect should incorporate the QCD radiative effects.

It will be interesting to see whether the gap between theory and the ATLAS data will be bridged by the NNLO contributions. Concerning the fixed-target regime, our results clearly make the case for additional new precision data for the Drell-Yan angular coefficients that would allow to convincingly establish whether there are departures from the 'plain' QCD radiative effects we have considered here. We hope that such data will be forthcoming from measurements at the COMPASS [217] or E906 [218] experiments, or possibly at RHIC.

Chapter 4

Divergent terms for SIDIS at small transverse momentum

We present an improved calculation of the SIDIS process at small transverse momentum q_T. For the first time all divergent terms for $q_T \to 0$ are included, which marks a milestone on the way to a complete resummation formalism that incorporates not only non-regular terms. We write down the explicit analytical form of the cross section including the Y-term that has to be used to adjust the low to the high q_T region. Our detailed calculation illustrates the crucial steps which are also applicable to other processes as the Drell-Yan process or electron-positron annihilation, where similar results can now be computed in the same way. The computations in this chapter have been performed in collaboration with Marc Schlegel.

4.1 Introduction

The transverse momentum distribution has been of special interest in the history of perturbative QCD, because the corresponding variables include QCD interactions already at the leading order. This is especially true for the perhaps most important examples respective to transverse momentum which are the Drell-Yan process $H_1+H_2 \to l+\bar{l}+X$ and the SIDIS process $l+H \to l+h+X$. While the leading-order diagrams without any QCD corrections consist of purely electromagnetic interactions (besides the parton distributions and in case of SIDIS the fragmentation function), these are also constraint to have no transverse momentum at all. First at $\mathcal{O}(\alpha_s)$ it is possible to find a non-zero transverse momentum. This next-to-leading order in terms of the strong coupling constant is therefore the leading order for transverse momentum distributions.

While the cross sections obtained by plain perturbative QCD in collinear factorization describe the experimental data quite well as long as all involved scales are large, they fail to describe the kinematic region where the transverse momentum q_T is much smaller as the hard scale of the process $q_T \ll Q$. The effect is well understood: as the transverse momentum becomes small, higher perturbative orders can only emit soft and collinear gluons. This leads to an imperfect cancellation of real and virtual diagrams, leaving large logarithms as remainder. At fixed order n these are of the form

$$\alpha_s^n \frac{1}{q_T^2} \ln^m\left(\frac{Q^2}{q_T^2}\right) \tag{4.1}$$

with $m \in \{1, \ldots, 2n-1\}$. If the transverse momentum becomes smaller than the hard scale of the process $q_T \ll Q$ the logarithms spoil the perturbarive series in the strong coupling constant α_s. But it is possible to reorder the series such that the perturbative expansion is described in terms of the logarithms itself. This is called the resummation of the cross section, as it is obtained by summing up (resumming) all logarithms of a specific power compared to the power of α_s up to all orders of the original expansion. The techniques for resummation have been developed first for electron-positron-annihilation and the Drell-Yan process [219–225]. This progress peaked at the development of the so called CSS formalism [225], which is still one of the standard methods for the resummation of small transverse momenta. A decade later the results had also been applied to the SIDIS process [226–230]. Subsequently also spin dependent processes, again first the Drell-Yan and later on the SIDIS process, have been calculated with incorporated resummation of small transverse momenta [98, 231–237], where for SIDIS an extensive collection for various combinations of polarized incoming and/or outgoing particles have been calculated in [237]. Beside small transverse momentum resummation has been generalized with the introduction of the TMD (transverse momentum dependent) formalism [62], which is equivalent for a specific choice of two auxiliary scales $\zeta_F = \zeta_D = Q^2$, differing only in higher orders of α_s [238].

Technically the resummation is performed in the impact parameter space, which is the Fourier space corresponding to the transverse momentum. In this space momentum conservation is guaranteed. Schematically the cross section is written as

$$\frac{\mathrm{d}\sigma}{\mathrm{d}q_T} \sim \int \mathrm{d}^2\vec{b}_T \; \mathrm{e}^{\mathrm{i}\vec{b}_T \cdot \vec{q}_T} \, W(b_T) + Y(q_T) \tag{4.2}$$

where in W the soft gluon contributions, more precisely all terms that are not regular for $q_T \to 0$, are resummed. It is dominant for the region with $q_T \ll Q$. But because it contains only terms that are at least proportional to q_T^{-2} it cannot describe the cross section for large q_T. The Y term is defined to compensate this deficit such that the cross section is described well for all q_T. It roughly contains all terms that are finite or at most logarithmically divergent for $q_T \to 0$. Further it is designed to guarantee a smooth transition of the two kinematical regimes [239].

As has already been pointed out in the original paper proposing of the CSS formalism, it is necessary to treat the singularities that are regular [225]. They promote a cutoff by replacing

$$Y(q_T) \to Y\left(\sqrt{q_T^2 + q_{T,\min}^2}\right) \tag{4.3}$$

where $q_{T,\min}$ defines the cutoff, and suggest a value of around $q_{T,\min} \approx 300\,\mathrm{MeV}$. At this point we want to perform further investigations and identify precisely all terms that are divergent as q_T approaches zero. A strong motivation is that by the common resummation procedure the terms that are divergent but are still regular for $q_T \to 0$ maybe overly suppressed, since they are inhibited in the region where their major contribution lies. For our calculation we will choose the unpolarized SIDIS process as

an example. But it can be applied similarly for other processes, unpolarized as well as polarized.

This chapter is organized as follows: in section 4.2 we introduce the kinematic used throughout our calculation, in the sections 4.3 to 4.5 we compute the unpolarized SIDIS cross section with special care to describe all divergent terms precisely even if they are regular. At NLO we encounter three different channels: the qq channel with a quark in the initial state and a quark that hadronizes into the measured hadron, the gq channel, where the measured hadron originates from a gluon, and the gq channel, where the initial parton is a gluon. Each of these channels is computed in a designated section, 4.3, 4.4 and 4.5 respectively. As side product we reproduce the unpolarized results in [237]. In section 4.6 we finally conclude the results and discuss their impact and effects for a more general resummation in the future. While in this chapter we present the most important steps, we provide also an exhausting detailed calculation in appendix D, which we will refer to during the calculation if appropriate.

4.2 Kinematics

As explained in the introduction we will perform the calculation through the example of the SIDIS process

$$l(l_1) + H(P) \to l(l_2) + h(P_h) + X(P_X) + Y(P_Y) \tag{4.4}$$

where the momenta of the particles are denoted in brackets. X and Y include all particles that are not measured, with X including the remnants of the initial state hadron that do not participate in the hard scattering and Y including particles in addition to h originating from a final state parton of the hard scattering process. Similar to [237] we will perform our calculation in the hadron-frame [240] (which is the infinite-momentum frame of the initial hadron and the virtual photon) . We will adapt also the notation in [237] to simplify the comparison of similar terms, except for using light-cone vectors instead of naive four vectors. Light-cone vectors will be indicated by square brackets. To describe the process we will use five Lorentz invariant variables

$$S = (P + l_1)^2 \tag{4.5a}$$

$$Q^2 = -q^2 = -(l_1 - l_2)^2 \tag{4.5b}$$

$$x = \frac{Q^2}{2P \cdot q} \tag{4.5c}$$

$$z = \frac{P \cdot P_h}{P \cdot q} \tag{4.5d}$$

$$\cosh(\psi) = \frac{2xS}{Q^2} - 1 \tag{4.5e}$$

where S is the center of mass energy squared, q^μ is the momentum of the virtual photon, x is the Bjorken scaling variable known from the DIS process, z is the equivalent scaling

variable for the final state hadron and ψ parametrizes the lepton momenta (see below). The transverse component of q^μ is defined by

$$q_t^\mu = q^\mu - \frac{1}{P \cdot P_h}\big((P_h \cdot q)P^\mu + (P \cdot q)P_h^\mu\big) \tag{4.6}$$

which is spacelike as is q^μ. Its magnitude is defined equally via

$$q_T = \sqrt{-q_t^2} \tag{4.7}$$

In terms of the variables the momenta are given in our particular frame by

$$P^\mu = \frac{Q}{\sqrt{2}x} n_+^\mu \tag{4.8a}$$

$$P_h^\mu = \frac{zQ}{\sqrt{2}} m^\mu \tag{4.8b}$$

$$q^\mu = \frac{Q}{\sqrt{2}}\big[-1, 1, \vec{0}_T\big] \tag{4.8c}$$

$$l_1^\mu = \frac{Q}{2}\left[\frac{1}{\sqrt{2}}(\cosh(\psi) - 1), \frac{1}{\sqrt{2}}(\cosh(\psi) + 1), \sinh(\psi)\cos(\phi), \sinh(\psi)\sin(\phi)\right] \tag{4.8d}$$

$$l_2^\mu = \frac{Q}{2}\left[\frac{1}{\sqrt{2}}(\cosh(\psi) + 1), \frac{1}{\sqrt{2}}(\cosh(\psi) - 1), \sinh(\psi)\cos(\phi), \sinh(\psi)\sin(\phi)\right] \tag{4.8e}$$

where ϕ is the angle between the lepton and the hadron plane and we introduced the normalized light-cone vectors

$$n_+^\mu = \big[1, 0, \vec{0}_T\big] \tag{4.9a}$$

$$n_-^\mu = \big[0, 1, \vec{0}_T\big] \tag{4.9b}$$

$$m^\mu = \left[\frac{q_T^2}{Q^2}, 1, \frac{\sqrt{2}q_T}{Q}, 0\right] \tag{4.9c}$$

$$\overline{m}^\mu = \left[1, 4\frac{Q^2}{q_T^2}, 2\sqrt{2}\frac{Q}{q_T}, 0\right] \tag{4.9d}$$

with $n_\pm^2 = m^2 = \overline{m}^2 = 0$ and $n_+ \cdot n_- = m \cdot \overline{m} = 1$. Obviously in our frame the transverse component is shifted completely to the measured hadron.

We will decompose the parton momenta of the incoming parton p and the fragmenting parton p_h via a Sudakov decomposition [241]

$$p^\mu = (p \cdot n_-)n_+^\mu + (p \cdot n_+)n_-^\mu + p_\perp^\mu \tag{4.10a}$$

$$p_h^\mu = (p_h \cdot \overline{m})m^\mu + (p_h \cdot m)\overline{m}^\mu + p_{h\perp}^\mu \tag{4.10b}$$

where the orthogonal components to the light-cone vectors are defined by

$$p_\perp^\mu = p_\nu(g^{\mu\nu} - n_+^\mu n_-^\nu - n_-^\mu n_+^\nu) \tag{4.11a}$$

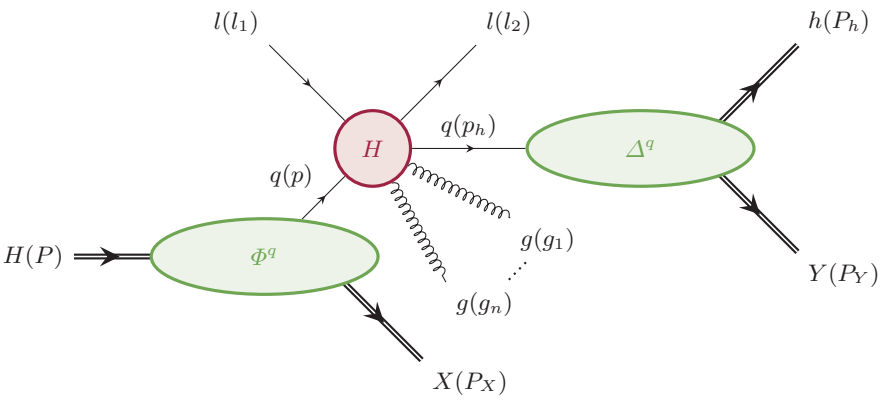


Figure 4.1: Invariant amplitude $\mathcal{M}_{qq}$ of the SIDIS process with additional n gluons in the final state. The hard scattering amplitude is depicted with the red blob H, while soft functions are depicted with green blobs. For this channel these are in particular the Fourier transform of a hadronic matrix element where a quark is created Φ^q and equally the Fourier transform of a hadronic matrix element where a quark creates a hadronic state containing one measured hadron Δ^q. The definitions can be read of easily in equation (4.14) Particle momenta are denoted in brackets.

$$p_{h\perp}^{\mu} = p_{h\nu}(g^{\mu\nu} - m^{\mu}\overline{m}^{\nu} - \overline{m}^{\mu}m^{\nu}) \tag{4.11b}$$

Finally we define the phase space of the measured particles including the Møller flux factor in d dimensions

$$\mathrm{d}\mathcal{R}_d \equiv \frac{1}{4P\cdot l_1}\frac{\mathrm{d}^{d-1}l_2}{2E_2(2\pi)^{d-1}}\frac{\mathrm{d}^{d-1}P_h}{2E_h(2\pi)^{d-1}} \tag{4.12}$$

with E_2 and E_h being the energy of the measured lepton and hadron respectively. To compare with [237] we use

$$\frac{\mathrm{d}\sigma}{\mathrm{d}x\mathrm{d}Q^2\mathrm{d}z\mathrm{d}q_T^2\mathrm{d}\phi} \leftrightarrow \frac{1}{(4\pi)^5}\frac{z}{Q^2}\frac{\mathrm{d}\sigma}{\mathrm{d}\mathcal{R}_4} \tag{4.13}$$

where we integrated out the azimuthal angle of the measured hadron ϕ_h to get from the right to the left-hand side, giving us a trivial factor of 2π. As we will see, all results will be independent of ϕ_h, which justifies the integration.

4.3 Quark PDF and quark FF—the qq channel

In this section we will present the calculations that incorporate a quark PDF and a quark FF. The corresponding invariant amplitude $\mathcal{M}_{qq}$ is depicted in figure 4.1 for additional n gluons in the final state. Using the common Feynman rules it is equal to

$$\mathcal{M}_{qq}\delta^d\left(P + l_1 - P_X - l_2 - P_h - P_Y - \sum_{m=0}^{n} g_m\right)$$

$$
\begin{aligned}
&= \int \frac{\mathrm{d}^d p}{(2\pi)^d} \int \frac{\mathrm{d}^d p_h}{(2\pi)^d}\, \delta^d\Big(l_1 + p - l_2 - p_h - \sum_{m=1}^{n} g_m\Big) \\
&\quad \times \left[\int \mathrm{d}^d \eta\, \mathrm{e}^{\mathrm{i}p\cdot\eta}\, \langle P_X|\psi_j^B(\eta)|P\rangle\right] H_{ij}^{AB} \left[\int \mathrm{d}^d \eta_h\, \mathrm{e}^{-\mathrm{i}p_h\cdot\eta_h}\, \langle P_Y, P_h|\bar{\psi}_i^A(\eta_h)|0\rangle\right] \\
&= \langle P_X|\psi_j^B(0)|P\rangle\, H_{ij}^{AB}\, \langle P_Y, P_h|\bar{\psi}_i^A(0)|0\rangle \\
&\quad \times \delta^d\Big(P + l_1 - P_X - l_2 - P_h - P_Y - \sum_{m=0}^{n} g_m\Big)
\end{aligned} \tag{4.14}
$$

where ψ_i^A is a quark field with Dirac index i and color index A. We absorbed the hard scattering as well as the leptonic parts and the additional gluon emissions into H. Then the differential cross section is given by the sum over all possible numbers of emitted gluons

$$
\begin{aligned}
\frac{\mathrm{d}\sigma_{qq}}{\mathrm{d}\mathcal{R}_d} &= \left[\sum_{n=0}^{\infty} \prod_{m=1}^{n} \sum_{\lambda_m, A_m} \int \frac{\mathrm{d}^d g_m}{(2\pi)^{d-1}} \delta((g_m)^2)\Theta(g_m^0)\right] \int \mathrm{d}^d p \int \mathrm{d}^d p_h \\
&\quad \times (2\pi)^d \delta^d\Big(l_1 - l_2 + p - p_h - \sum_{m=0}^{n} g_m\Big) \frac{1}{N_c}\, \mathrm{tr}\big(H\Phi^q(p,P)\gamma^0 H^\dagger \gamma^0 \Delta^q(p_h, P_h)\big)
\end{aligned} \tag{4.15}
$$

where the trace is over Dirac as well as color indices. λ_m and A_m are the polarization and the color state of the m-th emitted gluon. Further we introduced the fully unintegrated quark PDF correlator

$$
\Phi_{ij}^q(p,P) \equiv \int \frac{\mathrm{d}^d \eta}{(2\pi)^d}\, \mathrm{e}^{\mathrm{i}p\cdot\eta}\, \langle P|\bar{\psi}_j^A(0)\psi_i^A(\eta)|P\rangle \tag{4.16}
$$

and the fully unintegrated quark FF correlator

$$
\Delta_{ij}^q(p,P) \equiv \frac{1}{N_c} \sum\!\!\!\!\!\!\int_Y \int \frac{\mathrm{d}^d \eta}{(2\pi)^d}\, \mathrm{e}^{\mathrm{i}p\cdot\eta}\, \langle 0|\psi_i^A(\eta)|P, P_Y\rangle\, \langle P, P_Y|\bar{\psi}_j^A(0)|0\rangle \tag{4.17}
$$

where we use the shorthand notation

$$
\sum\!\!\!\!\!\!\int_X \equiv \sum_X \int \frac{\mathrm{d}^{d-1} P_X}{2E_{P_X}(2\pi)^{d-1}} \tag{4.18}
$$

for the phase space integrations of the unobserved remnants. We will perform the calculation in collinear factorization. In order to do this we use the collinear expansion, which assumes that the initial parton is collinear to its parent hadron and carries the momentum fraction ξ. Similarly the measured hadron is assumed to be collinear with its parent parton, carrying a momentum fraction ζ. Mathematically we drop the terms proportional to the transverse directions $\sim p_\perp^\mu$ and $\sim p_{h\perp}^\mu$ as well as the terms that

are $\sim n_-^\mu$ for the initial and $\sim \overline{m}^\mu$ for the final parton (see the Sudakov decomposition in equations (4.10)) in the hard scattering as well as in the overall delta distribution. This leaves us with

$$p^\mu \approx (p \cdot n_-) n_+^\mu \equiv \xi P^\mu \tag{4.19a}$$

$$p_h^\mu \approx (p_h \cdot \overline{m}) m^\mu \equiv \frac{1}{\zeta} P_h^\mu \tag{4.19b}$$

After the collinear expansions we are able to integrate out the corresponding degrees of freedom in the correlators, resulting in the collinear correlation functions

$$\Phi_{ij}^{q,\mathrm{coll}}(\xi, P) \equiv \int \frac{\mathrm{d}\eta}{2\pi} \, \mathrm{e}^{\mathrm{i}(P\cdot n_-)\xi\eta} \, \langle P | \bar{\psi}_j^A(0) [0, \eta n_-] \psi_i^A(\eta n_-) | P \rangle \tag{4.20a}$$

$$\begin{aligned} \Delta_{ij}^{q,\mathrm{coll}}(\zeta, P_h) \equiv \frac{1}{N_c} \sum\!\!\!\!\!\!\int_Y \int \frac{\mathrm{d}\eta}{2\pi} \, \mathrm{e}^{\mathrm{i}\frac{(P_h\cdot\overline{m})}{\zeta}\eta} \\ \times \langle 0 | \psi_i^A(\eta\overline{m}) [\eta\overline{m}, \infty] | P_h, P_Y \rangle \, \langle P_h, P_Y | [\infty, 0] \bar{\psi}_j^A(0) | 0 \rangle \end{aligned} \tag{4.20b}$$

In the definition of the correlators we included Wilson lines (gauge links), denoted in square brackets. These take into account additional gluons that may connect the soft parts of the process with the particles involved in the hard process. These cannot be ignored by arguing that they are of higher order, because the transfered momentum is not necessarily large. All these additional terms would enter with additional gluon fields in the correlators. It can be shown, that the sum over all possible numbers of additional gluons exponentiates, yielding the Wilson lines [58, 60, 61, 75] (for an explicit calculation that computes the first order of the gauge links see appendix B in [242]). For our unpolarized calculation the correlators can be parameterized by [55, 242–244]

$$\Phi^{q,\mathrm{coll}}(\xi, P) = \frac{1}{2} f_q(\xi) \not{n}_+ \tag{4.21a}$$

$$\Delta^{q,\mathrm{coll}}(\zeta, P_h) = \frac{1}{\zeta} D_q(\zeta) \not{n} \tag{4.21b}$$

where $f_q(\xi)$ is the ordinary unpolarized parton distribution function for quark q with momentum fraction ξ and $D_q(\zeta)$ is the unpolarized fragmentation function where the measured hadron carries momentum fraction ζ of its parent quark q. The differential cross section can thus be written as

$$\begin{aligned} \frac{\mathrm{d}\sigma_{qq}}{\mathrm{d}\mathcal{R}_d} = \left[\sum_{n=0}^{\infty} \prod_{m=1}^{n} \sum_{\lambda_m, A_m} \int \frac{\mathrm{d}^d g_m}{(2\pi)^{d-1}} \delta((g_m)^2) \Theta(g_m^0) \right] \int_0^1 \mathrm{d}\xi \int_0^1 \frac{\mathrm{d}\zeta}{\zeta} f_q(\xi) D_q(\zeta) \frac{Q^2}{4} \frac{z}{x\zeta^2} \\ \times (2\pi)^d \delta^d \left(l_1 - l_2 + \xi P - \frac{1}{\zeta} P_h - \sum_{m=0}^{n} g_m \right) \frac{1}{N_c} \operatorname{tr}\left(H \not{n}_+ \gamma^0 H^\dagger \gamma^0 \not{n} \right) \end{aligned} \tag{4.22}$$

Fur our purpose—a calculation up to NLO—we need the hard diagrams containing no gluon ($n = 0$), which are the leading-order diagram and all virtual diagrams up to

order α_s, and the diagrams containing one real gluon in the final state ($n = 1$). We start with $n = 0$. For this an exhaustive calculation is presented in appendix D.3.1.1. The result is given by

$$\frac{\mathrm{d}\sigma_{qq}^{n=0}}{\mathrm{d}\mathcal{R}_d} = \Gamma(1-\varepsilon)\left(\frac{q_T^2}{4\pi}\right)^{\varepsilon} \frac{2\pi(4\pi)^4\alpha^2}{zQ^2}\delta(q_T^2)(\mathcal{A}_1 + \varepsilon\mathcal{A}_2)e_q^2 f_q(x)D_q(z) \times \left[1 - \frac{\alpha_s C_F}{2\pi}\left(\frac{4\pi\mu^2}{Q^2}\right)^{\varepsilon} \frac{\Gamma(1+\varepsilon)\Gamma^2(1-\varepsilon)}{\Gamma(1-2\varepsilon)}\left[\frac{2}{\varepsilon^2} + \frac{3}{\varepsilon} + 8\right]\right] \tag{4.23}$$

where we introduce analogous to [226, 237, 245, 246] the structure functions

$$\mathcal{A}_1 = 1 + \cosh^2(\psi) \tag{4.24a}$$
$$\mathcal{A}_2 = -2 \tag{4.24b}$$
$$\mathcal{A}_3 = -\cos(\phi)\sinh(2\psi) \tag{4.24c}$$
$$\mathcal{A}_4 = \cos(2\phi)\sinh^2(\psi) \tag{4.24d}$$

and used the global $\delta(q_T^2)$ to eliminate contributions to the structure functions $\mathcal{A}_k$ with $k \in \{3, 4\}$. Further $\varepsilon > 0$ is the parameter that defines the dimension d via $d = 4 - 2\varepsilon$. Note that in the literature another set of structure functions is also widely used (see for example [247] and references therein), which relies on the kinematic variable y, which is the transferred energy fraction of the lepton to the hadron and is defined by

$$y \equiv \frac{P \cdot q}{P \cdot l_1} \tag{4.25}$$

The set of structure functions is a simple linear combination of the $\mathcal{A}_k$ and defined by

$$\mathcal{B}_1 \equiv 1 - y + \frac{y^2}{2} = \frac{y^2}{4}\mathcal{A}_1 \tag{4.26a}$$
$$\mathcal{B}_2 \equiv 1 - y = \frac{y^2}{4}(\mathcal{A}_1 + \mathcal{A}_2) \tag{4.26b}$$
$$\mathcal{B}_3 \equiv (2-y)\sqrt{1-y}\cos(\phi) = -\frac{y^2}{4}\mathcal{A}_3 \tag{4.26c}$$
$$\mathcal{B}_1 \equiv (1-y)\cos(2\phi) = \frac{y^2}{4}\mathcal{A}_4 \tag{4.26d}$$

Now we turn to $n = 1$. Details of the calculation are again provided in appendix D.3.1.2. The result for the cross section is given by[1]

$$\frac{\mathrm{d}\sigma_{qq}^{n=1}}{\mathrm{d}\mathcal{R}_d} = (4\pi)^4 \frac{\alpha^2\alpha_s\mu^{2\varepsilon}}{2zQ^4}\sum_k \mathcal{A}_k$$

[1] Note that we vary from the results in [237, 245, 246] by a factor of $Q^4/(x^2S^2)$. This will hold for the whole calculation, id est also for all other channels. The discrepancy is given exactly by a factor of y^{-2}, because $xyS = Q^2$. Interestingly this is part of the prefactor of the conversion from the $\mathcal{B}_k$ to the $\mathcal{A}_k$.

$$\times \int_x^1 \frac{\mathrm{d}\xi}{\xi} \int_z^1 \frac{\mathrm{d}\zeta}{\zeta} e_q^2 f_q(\xi) D_q(\zeta) \hat{\sigma}^{n=1}_{qq,k} \delta\left(\left(1-\frac{\xi}{x}\right)\left(1-\frac{\zeta}{z}\right) - \frac{q_T^2}{Q^2}\right) \tag{4.27}$$

where we characterize the partonic cross sections $\hat{\sigma}^{n=1}_{qq,k}$ by a series over the dimensional regulator ε

$$\hat{\sigma}^{n=1}_{qq,k} \equiv \sum_{i=0}^{\infty} \varepsilon^i \hat{\sigma}^{n=1}_{qq,k,i} \tag{4.28}$$

The non-zero coefficients are given by

$$\hat{\sigma}^{n=1}_{qq,1,0} = 2C_F \frac{xz}{\xi\zeta}\left[\frac{1}{Q^2 q_T^2}\left(Q^4 \frac{\xi^2\zeta^2}{x^2 z^2} + \left(Q^2 - q_T^2\right)^2\right) + 6\right] \tag{4.29a}$$

$$\hat{\sigma}^{n=1}_{qq,1,1} = -2C_F \frac{Q^2}{q_T^2}\left(1 + \frac{\zeta}{z} + \frac{\xi}{x} - 4\left[\frac{x}{\xi} + \frac{z}{\zeta}\right] + 5\frac{xz}{\xi\zeta}\right) + C_F - 2C_F \frac{q_T^2}{Q^2}\frac{xz}{\xi\zeta} \tag{4.29b}$$

$$\hat{\sigma}^{n=1}_{qq,2,0} = 8C_F \frac{xz}{\xi\zeta} \tag{4.29c}$$

$$\hat{\sigma}^{n=1}_{qq,2,1} = 2C_F \frac{Q^2}{q_T^2}\left(\frac{\xi\zeta}{xz} - 3\frac{xz}{\xi\zeta} + 4\left[\frac{x}{\xi} + \frac{z}{\zeta}\right] - 4\right) - 4C_F \frac{xz}{\xi\zeta} + 2C_F \frac{xz}{\xi\zeta}\frac{q_T^2}{Q^2} \tag{4.29d}$$

$$\hat{\sigma}^{n=1}_{qq,2,2} = 2C_F \frac{Q^2}{q_T^2}\left(\frac{\xi\zeta}{xz} - \frac{xz}{\xi\zeta} - 2\left[\frac{\zeta}{z} + \frac{\xi}{x}\right] + 4\right) + 4C_F \frac{xz}{\xi\zeta} - 2C_F \frac{xz}{\xi\zeta}\frac{q_T^2}{Q^2} \tag{4.29e}$$

$$\hat{\sigma}^{n=1}_{qq,3,0} = 4C_F \frac{xz}{\xi\zeta}\frac{Q^2 + q_T^2}{Q q_T} \tag{4.29f}$$

$$\hat{\sigma}^{n=1}_{qq,3,1} = 4C_F \frac{x(\zeta - z) + z(\xi - x)}{\xi\zeta}\frac{Q}{q_T} \tag{4.29g}$$

$$\hat{\sigma}^{n=1}_{qq,4,0} = 4C_F \frac{xz}{\xi\zeta} \tag{4.29h}$$

$$\hat{\sigma}^{n=1}_{qq,4,1} = -4C_F \frac{xz}{\xi\zeta} \tag{4.29i}$$

Up to higher twist corrections the derived formulas precisely describe the transverse momentum distribution. To regulate it for $q_T \to 0$ we have to expand the delta distribution in equation (4.27). As the most divergent terms are $\sim q_T^{-2}$ the expansion has to include all terms up to $\mathcal{O}(q_T^2)$ to guarantee that all divergent terms are taken into account. The calculation regarding the expansion of the delta distribution is presented in appendix D.4, the result is given in equation (D.84). We will use the alternative form of the generalized plus distribution (for its definition see equation (D.83)) in terms of ordinary plus and delta distributions, given in equation (D.86). Besides we adjust the prefactor to be consistent with the leading order in equation (4.23). Note

that the factor including the transverse momentum, which is required to perform the regularization, is only introduced by this adjustment. Moreover the global prefactor will not participate when the dimension will be set to four ($\varepsilon = 0$), because no finite terms are generated by it after regularization. Nevertheless we will not drop it for clarity and the result reads

$$\begin{aligned}\frac{\mathrm{d}\sigma_{qq}^{n=1}}{\mathrm{d}\mathcal{R}_d} \approx (4\pi)^4 \frac{\alpha^2\alpha_s}{2zQ^4}\Gamma(1-\varepsilon)\left(\frac{q_T^2}{4\pi}\right)^{\varepsilon} \sum_k \mathcal{A}_k \int_x^1 \mathrm{d}\xi \int_z^1 \mathrm{d}\zeta \\ \times \Bigg[\delta(\xi-x)\delta(\zeta-z)\ln\left(\frac{Q^2}{xzq_T^2}\right) + \frac{\delta(\xi-x)}{(\zeta-z)_+} + \frac{\delta(\zeta-z)}{(\xi-x)_+} \\ + \frac{xzq_T^2}{Q^2}\Bigg\{\delta(\xi-x)\delta(\zeta-z)\partial_\xi\partial_\zeta\left[2+\ln\left(\frac{Q^2}{xzq_T^2}\right)\right] \\ + \left[\frac{\delta(\xi-x)}{(\zeta-z)_+} + \frac{\delta(\zeta-z)}{(\xi-x)_+}\right]\partial_\xi\partial_\zeta \\ - \frac{\delta(\xi-x)\delta(1-\zeta)}{1-z} - \frac{\delta(\zeta-z)\delta(1-\xi)}{1-x}\Bigg\}\Bigg] \\ \times \frac{1}{\Gamma(1-\varepsilon)}\left(\frac{4\pi\mu^2}{q_T^2}\right)^{\varepsilon}\frac{xz}{\xi\zeta}e_q^2 f_q(\xi)D_q(\zeta)\hat{\sigma}_{qq,k}^{n=1}\end{aligned} \tag{4.30}$$

For the regularization of the expressions as $q_T \to 0$ only the leading-order terms of the delta distribution expansion are necessary, because the next-to-leading-order terms produce no non-regular divergences. This means we can follow the standard way described in [224]. A detailed analysis of the regularization is provided in the appendix D.5, including the renormalization of the PDF and FF. In order to present the result of this section we have to define the convolutions

$$f_q(x) \otimes g \equiv \int_x^1 \frac{\mathrm{d}\xi}{\xi} f_q(\xi) g\left(\frac{x}{\xi}\right) \tag{4.31a}$$

$$g \otimes D_q(z) \equiv \int_z^1 \frac{\mathrm{d}\zeta}{\zeta} D_q\left(\frac{z}{\zeta}\right) g(\zeta) \tag{4.31b}$$

the plus distribution respective to the transverse momentum

$$\int_0^{Q_T^2} \mathrm{d}q_T^2\, f(q_T^2)\big[g(q_T^2)\big]_+ \equiv \int_0^{Q_T^2} \mathrm{d}q_T^2\, \big[f(q_T^2) - f(0)\big] g(q_T^2) \tag{4.32}$$

where Q_T is the kinematic limit of the transverse momentum

$$Q_T^2 = Q^2 \frac{(1-x)(1-z)}{xz} \tag{4.33}$$

and the plus distribution regarding momentum fractions

$$\int_a^1 \frac{\mathrm{d}y}{y} \frac{f(y)}{(1-y)_+} \equiv \int_a^1 \frac{\mathrm{d}y}{y} \frac{f(y)-f(1)}{1-y} + f(1)\ln\left(\frac{1-a}{a}\right) \tag{4.34}$$

Furthermore we will use the ordinary leading order qq splitting function P_{qq} defined by equations (1.76b), (1.87b) and (B.1) and define in the spirit of [224] the abbreviations

$$c_q(y) \equiv (1-y) - 4\delta(1-y) \tag{4.35a}$$

$$\tilde{P}^i(y) \equiv \frac{y^i}{(1-y)_+} \tag{4.35b}$$

We present the result by dividing it into three parts

$$\frac{\mathrm{d}\sigma_{qq}}{\mathrm{d}\mathcal{R}_4} = e_q^2 \frac{2\pi(4\pi)^4\alpha^2}{zQ^2}\left[X_{qq}^{(1)}\delta(q_T^2) + \frac{\alpha_s}{2\pi}\big(X_{qq}^{(2)} + Y_{qq}\big)\right] \tag{4.36}$$

The first part obviously describes all parts that contribute exclusively for zero transverse momentum

$$\begin{aligned} X_{qq}^{(1)} = \mathcal{A}_1\Bigg[f_q(x)D_q(z) + \frac{\alpha_s}{2\pi}\Bigg\{ &\ln\left(\frac{Q_T^2}{\mu^2}\right)\big(f_q(x)\big[P_{qq}\otimes D_q(z)\big] + D_q(z)\big[f_q(x)\otimes P_{qq}\big]\big) \\ &+ f_q(x)D_q(z)C_F\left(3\ln\left(\frac{Q^2}{Q_T^2}\right) - \ln^2\left(\frac{Q^2}{Q_T^2}\right)\right) \\ &+ C_F D_q(z)\big[f_q(x)\otimes c_q\big] + C_F f_q(x)\big[D_q(z)\otimes c_q\big]\Bigg\}\Bigg] \end{aligned} \tag{4.37}$$

which contains the leading-order and also next-to-leading-order contributions. Further the structure function $\mathcal{A}_1$ is sufficient. The second part collects all terms that are divergent as $q_T \to 0$. This implies all terms that are usually resummed, id est the non integrable terms but also all integrable terms that are less but still divergent. Because the leading order is already described by $X_{qq}^{(1)}$ we factored out the coupling constant α_s in the definition. The term then reads

$$\begin{aligned} X_{qq}^{(2)} = \mathcal{A}_1 C_F f_q(x)D_q(z)&\left[2\left(\frac{\ln\left(\frac{Q^2}{q_T^2}\right)}{q_T^2}\right)_+ - \frac{3}{(q_T^2)_+}\right] \\ &+ \frac{\mathcal{A}_1}{(q_T^2)_+}\big(f_q(x)\big[P_{qq}\otimes D_q(z)\big] + D_q(z)\big[f_q(x)\otimes P_{qq}\big]\big) \\ &+ \frac{C_F}{Q^2}\big[4\mathcal{A}_1 + 4\mathcal{A}_2 + 2\mathcal{A}_4\big] f_q(x)D_q(z)\ln\left(\frac{Q^2}{q_T^2}\right) \end{aligned}$$

$$
\begin{aligned}
&+\frac{2C_F}{Q^2}\mathcal{A}_3\frac{Q}{q_T}\left[f_q(x)D_q(z)\ln\left(\frac{Q^2}{q_T^2}\right)+D_q(z)\big[f_q(x)\otimes\tilde{P}^2\big]+f_q(x)\big[D_q(z)\otimes\tilde{P}^2\big]\right]\\
&+\frac{2C_F}{Q^2}\mathcal{A}_1\ln\left(\frac{Q^2}{q_T^2}\right)\\
&\quad\times\big[2f_q(x)D_q(z)-xf_q'(x)D_q(z)-zf_q(x)D_q'(z)+xzf_q'(x)D_q'(z)\big] \qquad (4.38)
\end{aligned}
$$

where a prime denotes the first derivative of a function. In Y_{qq} all terms that are finite for $q_T\to 0$ are collected

$$
\begin{aligned}
Y_{qq}=&\frac{2C_F}{Q^2}\big[2\mathcal{A}_1+2\mathcal{A}_2+\mathcal{A}_4\big]\big(D_q(z)\big[f_q(x)\otimes\tilde{P}^2\big]+f_q(x)\big[D_q(z)\otimes\tilde{P}^2\big]\big)\\
&+\frac{C_F}{Q^2}\mathcal{A}_1\frac{q_T^2}{Q^2}\left[f_q(x)D_q(z)\ln\left(\frac{Q^2}{q_T^2}\right)+D_q(z)\big[f_q(x)\otimes\tilde{P}^2\big]+f_q(x)\big[D_q(z)\otimes\tilde{P}^2\big]\right]\\
&+\frac{2C_F}{Q^2}\mathcal{A}_3\frac{q_T}{Q}\left[f_q(x)D_q(z)\ln\left(\frac{Q^2}{q_T^2}\right)+D_q(z)\big[f_q(x)\otimes\tilde{P}^2\big]+f_q(x)\big[D_q(z)\otimes\tilde{P}^2\big]\right]\\
&+\frac{2C_F}{Q^2}\left(2\mathcal{A}_2+\mathcal{A}_4+\mathcal{A}_3\left[\frac{Q}{q_T}+\frac{q_T}{Q}\right]\right)\frac{q_T^2}{Q^2}\\
&\quad\times\Bigg[\big(xzf_q'(x)D_q'(z)-2xf_q'(x)D_q(z)-2zf_q(x)D_q'(z)+4f_q(x)D_q(z)\big)\\
&\qquad\times\left[2+\ln\left(\frac{Q^2}{q_T^2}\right)\right]\\
&\qquad+xzD_q'(z)\big[f_q'(x)\otimes\tilde{P}^2\big]+xzf_q'(x)\big[D_q'(z)\otimes\tilde{P}^2\big]-2xD_q(z)\big[f_q'(x)\otimes\tilde{P}^2\big]\\
&\qquad-2xf_q'(x)\big[D_q(z)\otimes\tilde{P}^3\big]-2zD_q'(z)\big[f_q(x)\otimes\tilde{P}^3\big]-2zf_q(x)\big[D_q'(z)\otimes\tilde{P}^2\big]\\
&\qquad+4D_q(z)\big[f_q(x)\otimes\tilde{P}^3\big]+4f_q(x)\big[D_q(z)\otimes\tilde{P}^3\big]\Bigg]\\
&+\frac{C_F}{Q^2}\mathcal{A}_1\left(4\frac{q_T^2}{Q^2}+\frac{q_T^4}{Q^4}\right)\\
&\quad\times\Bigg[\big(4f_q(x)D_q(z)-2xf_q'(x)D_q(z)-2zf_q(x)D_q'(z)\big)\left[2+\ln\left(\frac{Q^2}{q_T^2}\right)\right]\\
&\qquad-2xD_q(z)\big[f_q'(x)\otimes\tilde{P}^2\big]-2xf_q'(x)\big[D_q(z)\otimes\tilde{P}^3\big]-2zD_q'(z)\big[f_q(x)\otimes\tilde{P}^3\big]\\
&\qquad-2zf_q(x)\big[D_q'(z)\otimes\tilde{P}^2\big]+4D_q(z)\big[f_q(x)\otimes\tilde{P}^3\big]+4f_q(x)\big[D_q(z)\otimes\tilde{P}^3\big]\Bigg]\\
&+\frac{C_F}{Q^2}\mathcal{A}_1\Bigg[-2x\big(D_q(z)\big[f_q'(x)\otimes\tilde{P}^2\big]+f_q'(x)\big[D_q(z)\otimes\tilde{P}^3\big]+2f_q'(x)D_q(z)\big)\\
&\qquad\qquad-2z\big(D_q'(z)\big[f_q(x)\otimes\tilde{P}^3\big]+f_q(x)\big[D_q'(z)\otimes\tilde{P}^2\big]+2f_q(x)D_q'(z)\big)
\end{aligned}
$$

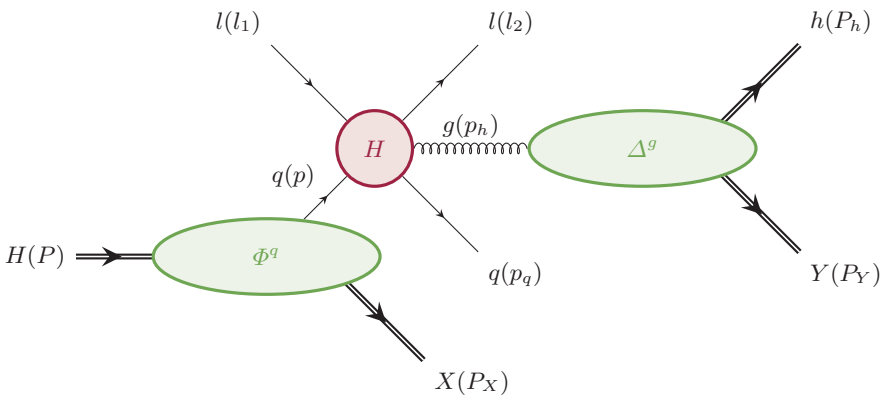


Figure 4.2: Invariant amplitude $\mathcal{M}_{gq}$ of the SIDIS process. The hard scattering amplitude is depicted with the red blob H, while soft functions are depicted with green blobs. For this channel these are in particular the Fourier transform of a hadronic matrix element where a quark is created Φ^q and the Fourier transform of a hadronic matrix element where a gluon creates a hadronic state containing one measured hadron Δ^g. The definitions can be read of easily in equation (4.40). Particle momenta are denoted in brackets.

$$
\begin{aligned}
&\quad + 4D_q(z)\big[f_q(x)\otimes\tilde{P}^3\big] + 4f_q(x)\big[D_q(z)\otimes\tilde{P}^3\big] + 8f_q(x)D_q(z)\Bigg] \\
&+ \frac{1}{Q^2}\mathcal{A}_1 xz\Bigg[\left(4\frac{q_T^2}{Q^2}+\frac{q_T^4}{Q^4}\right)C_F\big(D_q'(z)\big[f_q'(x)\otimes\tilde{P}^2\big] + f_q'(x)\big[D_q'(z)\otimes\tilde{P}^2\big]\big) \\
&\quad + D_q'(z)\big[f_q'(x)\otimes P_{qq}\big] + f_q'(x)\big[P_{qq}\otimes D_q'(z)\big] - 3C_F f_q'(x)D_q'(z) \\
&\quad + \left(4\frac{q_T^2}{Q^2}+\frac{q_T^4}{Q^4}\right)C_F f_q'(x)D_q'(z)\left[2+\ln\left(\frac{Q^2}{q_T^2}\right)\right] + 4C_F f_q'(x)D_q'(z)\Bigg]
\end{aligned}
\tag{4.39}
$$

4.4 Quark PDF and gluon FF—the gq channel

This section is concerned with the terms that incorporate a quark PDF but a gluon FF. The corresponding invariant amplitude $\mathcal{M}_{gq}$ is shown in figure 4.2. We will follow the same steps as in the previous section. By using the Feynman rules the invariant amplitude can be expressed by

$$
\begin{aligned}
&\mathcal{M}_{gq}\delta^d\big(P+l_1-P_X-l_2-P_h-P_Y-p_q\big) \\
&\quad = \int\frac{\mathrm{d}^d p}{(2\pi)^d}\int\frac{\mathrm{d}^d p_h}{(2\pi)^d}\,\delta^d\big(l_1+p-l_2-p_h-p_q\big)
\end{aligned}
$$

$$
\begin{aligned}
&\times \left[\int \mathrm{d}^d\eta\ \mathrm{e}^{\mathrm{i}p\cdot\eta}\,\langle X|\psi_i^A(\eta)|P\rangle\right] H_{i,A}^{a,\mu}\left[\int \mathrm{d}^d\eta_h\ \mathrm{e}^{-\mathrm{i}p_h\cdot\eta_h}\,\langle P_Y,P_h|G_\mu^a(\eta_h)|0\rangle\right]\\
&= \frac{-\mathrm{i}}{(P_h+P_Y)\cdot\overline{m}}\,\langle X|\psi_i^A(0)|P\rangle\, H_{i,A}^{a,\mu}\,\langle P_Y,P_h|\overline{m}^\nu G_{\nu\mu}^a(0)|0\rangle\\
&\quad\times\delta^d\big(P+l_1-P_X-l_2-P_h-P_Y-p_q\big)
\end{aligned}
\tag{4.40}
$$

As before ψ_i^A denotes a quark field with Dirac index i and color index A, where the fundamental representation is used. Now also a gluon field is present, denoted by G_μ^a with μ being the Lorentz index and a the color index of the adjoint representation. $G_{\mu\nu}^a$ is the corresponding field strength tensor. The hard scattering H encloses the leptonic parts of the amplitude again. Its indices are adapted to its external interface compared to the hard scattering function we defined in the previous section. The cross section deduced from the amplitude is given by

$$
\begin{aligned}
\frac{\mathrm{d}\sigma_{gq}}{\mathrm{d}\mathcal{R}_d} &= \int \frac{\mathrm{d}^dp_q}{(2\pi)^{d-1}}\delta((p_q)^2)\Theta(p_q^0)\int \mathrm{d}^dp\int \mathrm{d}^dp_h\,(2\pi)^d\delta^d(l_1-l_2+p-p_h-p_q)\\
&\quad\times\frac{1}{(p_h\cdot\overline{m})^2}\frac{1}{N_c}\,\mathrm{tr}\big(H^\mu\Phi^q(p,P)\gamma^0H^{\dagger,\nu}\Delta_{\nu\mu}^g(p_h,P_h)\big)
\end{aligned}
\tag{4.41}
$$

where the trace is again over all indices. In this case these are in particular the Dirac indices of H and Φ^q and both types of color indices (fundamental and adjoint representation) that are embodied in H. We keep the Lorentz indices of H and Δ^g explicit. Latter is the fully unintegrated gluon FF correlator defined by

$$
\begin{aligned}
\Delta_{\nu\mu}^g(p,P) \equiv \frac{1}{N_c^2-1}\sum\!\!\!\!\!\!\int_Y \int \frac{\mathrm{d}^d\eta}{(2\pi)^d}\,\mathrm{e}^{\mathrm{i}p\cdot\eta}\\
\times\langle 0|\overline{m}^\rho G_{\rho\nu}^a(0)|P,P_Y\rangle\,\langle P,P_Y|\overline{m}^\sigma G_{\sigma\mu}^a(\eta)|0\rangle
\end{aligned}
\tag{4.42}
$$

while Φ^q is again the quark PDF correlator defined in equation (4.20a). Note that the complex conjugated hard part is accompanied by only one γ^0, because it contains an external quark. We use again the collinear expansion to accomplish collinear factorization for the following computation. That means we approximate the parton momenta in the hard scattering part and in the delta distribution by means of equations (4.19) which allows us to integrate out the other degrees of freedom in the unintegrated correlator and to find the collinear gluon FF correlation function

$$
\begin{aligned}
\Delta_{\nu\mu}^{g,\mathrm{coll}}(\zeta,P_h) \equiv \frac{1}{N_c^2-1}\sum\!\!\!\!\!\!\int_Y \int \frac{\mathrm{d}\eta}{2\pi}\,\mathrm{e}^{\mathrm{i}\frac{(P_h\cdot\overline{m})}{\zeta}\eta}\\
\times\langle 0|\overline{m}^\rho G_{\rho\nu}^a(\eta\overline{m})[\eta\overline{m},\infty]|P_h,P_Y\rangle\,\langle P_h,P_Y|[\infty,0]\overline{m}^\sigma G_{\sigma\mu}^a(0)|0\rangle
\end{aligned}
\tag{4.43}
$$

where we have included the appropriate Wilson lines as before. The quark PDF correlator is again parametrized according to equation (4.21a). For the gluon FF

correlator the parametrization is given by

$$\Delta^{g,\text{coll}}_{\nu\mu}(\zeta, P_h) = \frac{p_h \cdot \overline{m}}{\zeta} D_g(\zeta) \sum_\lambda \epsilon^*_\mu(m,\lambda)\epsilon_\nu(m,\lambda) \tag{4.44}$$

Here ϵ_μ is the polarization vector of the hadronizing gluon and D_g the ordinary unpolarized collinear gluon fragmentation function. We use Feynman gauge, which means the polarization sum gives

$$\Delta^{g,\text{coll}}_{\nu\mu}(\zeta, P_h) = -g_{\mu\nu} \frac{p_h \cdot \overline{m}}{\zeta} D_g(\zeta) \tag{4.45}$$

and we may write the differential cross section in collinear factorization

$$\begin{aligned}\frac{\mathrm{d}\sigma_{gq}}{\mathrm{d}\mathcal{R}_d} = \frac{\pi}{zQ^2} \int_x^1 \frac{\mathrm{d}\xi}{\xi} \int_z^1 \frac{\mathrm{d}\zeta}{\zeta} f_q(\xi) D_g(\zeta) \delta\left(\left(1-\frac{\zeta}{z}\right)\left(1-\frac{\xi}{x}\right) - \frac{q_T^2}{Q^2}\right) \\ \times \frac{-g_{\mu\nu}}{N_c} \operatorname{tr}\left(H^\mu \not{p} \gamma^0 H^{\dagger,\nu}\right)\end{aligned} \tag{4.46}$$

where we already used the d dimensional delta distribution to evaluate the p_q integral (in contrast to the qq channel, here we only have to take care for this kinematics). The partonic scattering amplitude is computed in appendix D.3.2. Inserting the result into the hadronic cross section yields

$$\begin{aligned}\frac{\mathrm{d}\sigma_{gq}}{\mathrm{d}\mathcal{R}_d} = (4\pi)^4 \frac{\alpha^2 \alpha_s \mu^{2\varepsilon}}{2zQ^4} \sum_k \mathcal{A}_k \\ \times \int_x^1 \frac{\mathrm{d}\xi}{\xi} \int_z^1 \frac{\mathrm{d}\zeta}{\zeta} e_q^2 f_q(\xi) D_g(\zeta) \hat{\sigma}_{gq,k} \delta\left(\left(1-\frac{\xi}{x}\right)\left(1-\frac{\zeta}{z}\right) - \frac{q_T^2}{Q^2}\right)\end{aligned} \tag{4.47}$$

with

$$\hat{\sigma}_{gq,k} \equiv \sum_{i=0}^{\infty} \varepsilon^i \hat{\sigma}_{gq,k,i} \tag{4.48}$$

and the non-zero coefficients are given by

$$\hat{\sigma}_{gq,1,0} = 2C_F \left[\frac{Q^2}{q_T^2}\frac{x}{\xi}\left(1-\frac{z}{\zeta}\right)\left(\frac{\xi^2\zeta^2}{x^2z^2} + \frac{\zeta^2}{z^2}\left(1-\frac{z}{\zeta}\right)^2\right) + 4\frac{x}{\xi} + \frac{z}{\zeta} - 5\frac{xz}{\xi\zeta}\right] \tag{4.49a}$$

$$\begin{aligned}\hat{\sigma}_{gq,1,1} = 2C_F \Bigg[\frac{q_T^2}{Q^2}\left(1-\frac{x}{\xi}\right)\left(1-\frac{z}{\zeta}\right) - \frac{Q^2}{q_T^2}\left(1-\frac{z}{\zeta}\right) \\ \times \left\{\frac{\xi}{x}\left(1-\frac{\zeta}{z}\right)^2\left(\frac{\xi}{x}-2\right) - \left(1-\frac{\zeta}{z}\right)\left(3+\frac{\zeta}{z}\right) + \frac{x}{\xi}\left(5-4\frac{\zeta}{z}\right)\right\}\Bigg]\end{aligned} \tag{4.49b}$$

$$\hat{\sigma}_{gq,2,0} = 8C_F \frac{x}{\xi}\left(1-\frac{z}{\zeta}\right) \tag{4.49c}$$

$$\hat{\sigma}_{gq,2,1} = 2C_F \frac{Q^2}{q_T^2}\left(1-\frac{z}{\zeta}\right)\left[4-6\frac{\zeta}{z}+\frac{x}{\xi}\left(\frac{\zeta^2}{z^2}+4\frac{\zeta}{z}-4\right)+\frac{\xi}{x}\left(1+\frac{\zeta^2}{z^2}\right)\right] \tag{4.49d}$$

$$\hat{\sigma}_{gq,2,2} = -2C_F \frac{Q^2}{q_T^2}\left(1-\frac{z}{\zeta}\right)\left[\frac{x\zeta^2}{\xi z^2}+2\frac{\zeta}{z}\left(1-\frac{\zeta}{z}\right)+\frac{\xi}{x}\left(1-\frac{\zeta}{z}\right)^2\right] \tag{4.49e}$$

$$\hat{\sigma}_{gq,3,0} = -4C_F\left[\frac{Q}{q_T}\frac{x\zeta}{\xi z}\left(1-\frac{z}{\zeta}\right)^2+\frac{q_T}{Q}\frac{xz}{\xi\zeta}\right] \tag{4.49f}$$

$$\hat{\sigma}_{gq,3,1} = -4C_F \frac{Q}{q_T}\left(1-\frac{z}{\zeta}\right)\left(\frac{\zeta}{z}+2\frac{x}{\xi}-\frac{x\zeta}{\xi z}-1\right) \tag{4.49g}$$

$$\hat{\sigma}_{gq,4,0} = 4C_F \frac{x}{\xi}\left(1-\frac{z}{\zeta}\right) \tag{4.49h}$$

$$\hat{\sigma}_{gq,4,1} = -4C_F \frac{x}{\xi}\left(1-\frac{z}{\zeta}\right) \tag{4.49i}$$

As we have done for the $n = 1$ case of the qq channel we now apply the approximation of the delta distribution and adjust the prefactor to match the leading-order expression, see the similar transition form equation (4.27) to (4.30). Then the regularization for $q_T \to 0$ is performed, including the renormalization of the PDF and the FF, where we encounter the splitting function P_{gq} as defined in (B.1d) and additionally define

$$c_g^i(y) \equiv y^i \tag{4.50}$$

Then the final result of this channel can be expressed by

$$\frac{\mathrm{d}\sigma^{gq}}{\mathrm{d}\mathcal{R}_4} = e_q^2 \frac{2\pi(4\pi)^4\alpha^2}{zQ^2}\frac{\alpha_s}{2\pi}\left[X_{gq}^{(1)}\delta(q_T^2)+X_{gq}^{(2)}+Y_{gq}\right] \tag{4.51}$$

where again $X_{gq}^{(1)}$ contains all terms that contribute only for $q_T = 0$, $X_{gq}^{(2)}$ is accounting all divergent terms as $q_T \to 0$ and Y_{gq} collects all finite terms. In contrast to the qq channel all terms are of order α_s, which motivates a slightly different segmentation of the result. In particular we have factored out the overall factor $\alpha_s/(2\pi)$ from $X_{gq}^{(1)}$. The coefficients are then given by

$$X_{gq}^{(1)} = \mathcal{A}_1\left[\ln\left(\frac{Q_T^2}{\mu^2}\right) f_q(x)\left[P_{gq}\otimes D_g(z)\right]+C_F f_q(x)\left[D_g(z)\otimes c_g^1\right]\right] \tag{4.52}$$

$$X_{gq}^{(2)} = \mathcal{A}_1\left[\frac{1}{(q_T^2)_+} f_q(x)\left[P_{gq}\otimes D_g(z)\right]+\frac{C_F}{Q^2}x f_q'(x) D_g(z)\ln\left(\frac{Q^2}{q_T^2}\right)\right]$$

$$- \mathcal{A}_3 \frac{2C_F}{Qq_T} f_q(x) \big[D_g(z) \otimes (c_g^0 - c_g^1) \big] \tag{4.53}$$

$$\begin{aligned}
Y_{gq} = \mathcal{A}_1 \frac{C_F}{Q^2} \Bigg[& D_g(z) \big[f_q(x) \otimes c_g^1 \big] + 4 f_q(x) \big[D_g(z) \otimes c_g^1 \big] \\
& + xz f_q'(x) \big[D_g'(z) \otimes c_g^{-1} \big] + x D_g(z) \big[f_q'(x) \otimes \tilde{P}^0 \big] + x f_q'(x) \big[D_g(z) \otimes \tilde{P}^0 \big] \\
& + 2x f_q'(x) D_g(z) + \big[x f_q'(x) - 2 f_q(x) \big] \\
& \quad \times \Bigg(\big[D_g(z) \otimes \big(c_g^0 + c_g^1 - 2c_g^2 \big) \big] + z \Bigg[D_g'(z) \otimes \left(\frac{1}{C_F} P_{gq} - c_g^{-1} \right) \Bigg] \Bigg) \\
& + \frac{q_T^2}{Q^2} \Bigg(2 D_g(z) \big[f_q(x) \otimes \big(\tilde{P}^2 - 6\tilde{P}^3 \big) \big] + 2 f_q(x) \big[D_g(z) \otimes \big(4\tilde{P}^2 - 9\tilde{P}^3 \big) \big] \\
& \qquad + 2x D_g(z) \big[f_q'(x) \otimes \big(3\tilde{P}^2 - \tilde{P}^1 \big) \big] \\
& \qquad + 2x f_q'(x) \big[D_g(z) \otimes \big(5\tilde{P}^3 - 3\tilde{P}^2 \big) \big] \\
& \qquad + z D_g'(z) \big[f_q'(x) \otimes \big(2\tilde{P}^3 - \tilde{P}^2 \big) \big] \\
& \qquad + z f_q(x) \big[D_g'(z) \otimes \big(9\tilde{P}^2 - 8\tilde{P}^1 \big) \big] \\
& \qquad + xz D_g'(z) \big[f_q'(x) \otimes c_g^1 \big] + 4xz f_q'(x) \big[D_g'(z) \otimes c_g^1 \big] \\
& \qquad + \left[2 + \ln \left(\frac{Q^2}{q_T^2} \right) \right] \\
& \qquad \quad \times \big[z f_q(x) D_g'(z) + 4x f_q'(x) D_g(z) - 10 f_q(x) D_q(z) \big] \Bigg) \Bigg] \\
& + \big(2\mathcal{A}_2 + \mathcal{A}_4 \big) \frac{2C_F}{Q^2} \\
& \quad \times \Bigg[f_q(x) \big[D_g(z) \otimes c_g^1 \big] \\
& \qquad + \frac{q_T^2}{Q^2} \Bigg(-2 D_g(z) \big[f_q(x) \otimes \tilde{P}^3 \big] + 2 f_q(x) \big[D_g(z) \otimes \big(\tilde{P}^2 - 2\tilde{P}^3 \big) \big] \\
& \qquad\quad \big(x f_q'(x) D_g(z) - 2 f_q(x) D_g(z) \big) \left[2 + \ln \left(\frac{Q^2}{q_T^2} \right) \right] \\
& \qquad\quad + x D_g(z) \big[f_q'(x) \otimes \tilde{P}^2 \big] - x f_q'(x) \big[D_g(z) \otimes \big(\tilde{P}^2 - 2\tilde{P}^3 \big) \big]
\end{aligned}$$

$$
\begin{aligned}
&\qquad - 2z f_q(x)\big[D'_g(z)\otimes c_g^1\big] + xz f'_q(x)\big[D'_g(z)\otimes c_g^1\big]\Big)\Bigg] \\
&- \mathcal{A}_3 \frac{2C_F}{Q^2}\frac{q_T}{Q} \\
&\times \Bigg[f_q(x) D_g(z) \ln\!\left(\frac{Q^2}{q_T^2}\right) + D_g(z)\big[f_q(x)\otimes \tilde{P}^2\big] + f_q(x)\big[D_g(z)\otimes \tilde{P}^2\big] \\
&\quad + \big(x f'_q(x) - 2 f_q(x)\big)\Big(2\big[D_g(z)\otimes c_g^2\big] + z\big[D'_g(z)\otimes (c_g^0 - c_g^1)\big]\Big) \\
&\quad + \frac{q_T^2}{Q^2}\Bigg(\Big(4 f_q(x) D_g(z) - 2x f'_q(x) D_g(z) - 2z f_q(x) D'_g(z) + xz f'_q(x) D'_g(z)\Big) \\
&\qquad\qquad \times \left[2 + \ln\!\left(\frac{q_T^2}{Q^2}\right)\right] \\
&\qquad\qquad + 4\Big(D_g(z)\big[f_q(x)\otimes \tilde{P}^3\big] + f_q(x)\big[D_g(z)\otimes \tilde{P}^3\big]\Big) \\
&\qquad\qquad - 2x\Big(D_g(z)\big[f'_q(x)\otimes \tilde{P}^2\big] + f'_q(x)\big[D_g(z)\otimes \tilde{P}^3\big]\Big) \\
&\qquad\qquad - 2z\Big(D'_g(z)\big[f_q(x)\otimes \tilde{P}^3\big] + f_q(x)\big[D'_g(z)\otimes \tilde{P}^2\big]\Big) \\
&\qquad\qquad + xz\Big(D'_g(z)\big[f'_q(x)\otimes \tilde{P}^2\big] + f'_q(x)\big[D'_g(z)\otimes \tilde{P}^2\big]\Big)\Bigg)\Bigg]
\end{aligned}
\tag{4.54}
$$

4.5 Gluon PDF and quark FF—the qg channel

The final channel contributing at $\mathcal{O}(\alpha_s)$ to the cross section includes a gluon PDF and a quark FF. The related invariant amplitude $\mathcal{M}_{qg}$ is shown in figure 4.3. As in the sections before we can read off the analytical expression

$$
\begin{aligned}
&\mathcal{M}_{qg}\delta^d\big(P + l_1 - P_X - l_2 - P_h - P_Y - p_{\bar{q}}\big) \\
&\quad = \int \frac{\mathrm{d}^d p}{(2\pi)^d} \int \frac{\mathrm{d}^d p_h}{(2\pi)^d}\, \delta^d\big(l_1 + p - l_2 - p_h - p_q\big) \\
&\qquad \times \left[\int \mathrm{d}^d\eta\ \mathrm{e}^{\mathrm{i}p\cdot\eta}\,\langle X|G^a_\mu(\eta)|P\rangle\right] H^{a,\mu}_{i,A}\left[\int \mathrm{d}^d\eta_h\ \mathrm{e}^{-\mathrm{i}p_h\cdot\eta_h}\,\langle P_Y, P_h|\bar{\psi}^A_i(\eta_h)|0\rangle\right] \\
&\quad = \frac{\mathrm{i}}{(P - P_X)^+}\,\langle X|G^a_{+\mu}(0)|P\rangle\, H^{a,\mu}_{i,A}\,\langle P_Y, P_h|\bar{\psi}^A_i(0)|0\rangle \\
&\qquad \times \delta^d\big(P + l_1 - P_X - l_2 - P_h - P_Y - p_q\big)
\end{aligned}
\tag{4.55}
$$

We denote the plus component of a four vector by a superscript plus $p^+ \equiv p\cdot n_-$. All other notations are similar to the notations we introduced during the computation of

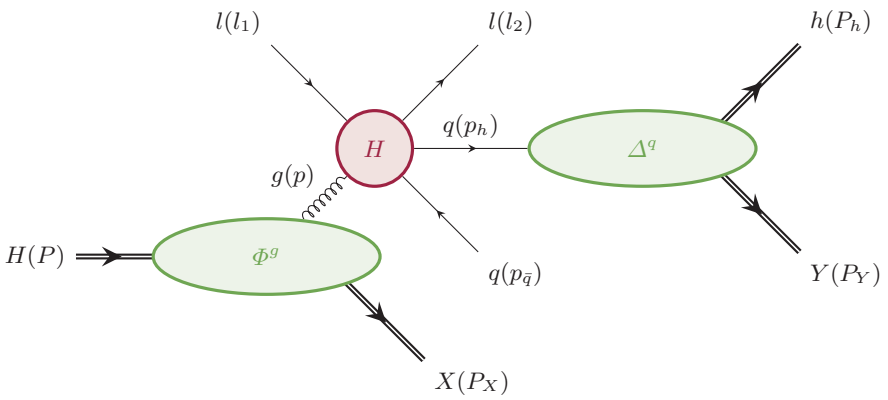


Figure 4.3: Invariant amplitude $\mathcal{M}_{qg}$ of the SIDIS process. The hard scattering amplitude is depicted with the red blob H, while soft functions are depicted with green blobs. For this channel these are in particular the Fourier transform of a hadronic matrix element where a gluon is created Φ^g and the Fourier transform of a hadronic matrix element where a quark creates a hadronic state containing one measured hadron Δ^q. The definitions can be read of easily in equation (4.55). Particle momenta are denoted in brackets.

the other channels. The corresponding differential cross section is given by

$$\begin{aligned}\frac{\mathrm{d}\sigma_{qg}}{\mathrm{d}\mathcal{R}_d} = \int \frac{\mathrm{d}^d p_{\bar{q}}}{(2\pi)^{d-1}} \delta((p_{\bar{q}})^2)\Theta(p^0_{\bar{q}}) \int \mathrm{d}^d p \int \mathrm{d}^d p_h \, (2\pi)^d \delta^d(l_1 - l_2 + p - p_h - p_{\bar{q}}) \\ \times \frac{1}{(k_1^+)^2} \frac{1}{N_c^2 - 1} \operatorname{tr}\big(H^\mu \Phi^g_{\nu\mu} (H^{\dagger,\nu}\gamma^0)\Delta^q\big)\end{aligned} \tag{4.56}$$

where we defined the unintegrated gluon PDF correlator

$$\Phi^g_{\nu\mu} \equiv \int \frac{\mathrm{d}^d \eta}{(2\pi)^d} \, \mathrm{e}^{\mathrm{i}p\cdot\eta} \, \langle P | G^a_{+\nu}(0) G^a_{+\mu}(\eta) | P \rangle \tag{4.57}$$

The quark FF correlator Δ^q_{ij} has been defined in equation (4.17). Applying the collinear expansion (see equations (4.19)) in the partonic parts and the overall delta distribution lets us integrate out the non-collinear degrees of freedom of the correlators. The collinear quark FF correlator is given in equation (4.20b), for the gluon PDF correlator we find

$$\Phi^{g,\mathrm{coll}}_{\nu\mu} = \int \frac{\mathrm{d}\eta}{2\pi} \, \mathrm{e}^{\mathrm{i}P^+\xi\eta} \, \langle P | G^a_{+\nu}(0) [0, \eta n_-] G^a_{+\mu}(\eta n_-) | P \rangle \tag{4.58}$$

where as always we included the appropriate gauge link. The parameterization of the collinear quark FF correlator is given in equation (4.21b) and the collinear gluon PDF correlator can be parametrized in Feynman gauge in terms of the ordinary gluon PDF f_g by

$$\Phi^{g,\mathrm{coll}}_{\nu\mu} = \frac{p^+}{2} f_g(\xi) \sum_\lambda \epsilon^*_\mu(n_+, \lambda)\epsilon_\nu(n_+, \lambda) = -g_{\mu\nu} \frac{p^+}{2} f_g(\xi) \tag{4.59}$$

Putting everything together the cross section can be expressed by

$$\frac{\mathrm{d}\sigma_{qg}}{\mathrm{d}\mathcal{R}_d} = \frac{\pi}{zQ^2}\int_x^1 \frac{\mathrm{d}\xi}{\xi}\int_z^1 \frac{\mathrm{d}\zeta}{\zeta} f_g(\xi) D_q(\zeta)\delta\left(\left(1-\frac{\zeta}{z}\right)\left(1-\frac{\xi}{x}\right)-\frac{q_T^2}{Q^2}\right) \times \frac{-g_{\mu\nu}}{N_c^2-1}\,\mathrm{tr}\big(H^\mu H^{\dagger,\nu}\gamma^0 \not{p}_h\big) \tag{4.60}$$

Similar to the gq but in contrast to the qq channel we have used the d dimensional delta distribution to eliminate the $p_{\bar{q}}$ integral. Details to the calculation of the partonic invariant amplitude are presented in appendix D.3.3. The result is given by

$$\frac{\mathrm{d}\sigma_{gq}}{\mathrm{d}\mathcal{R}_d} = (4\pi)^4\frac{\alpha^2\alpha_s\mu^{2\varepsilon}}{2zQ^4}\sum_k \mathcal{A}_k \times \int_x^1 \frac{\mathrm{d}\xi}{\xi}\int_z^1 \frac{\mathrm{d}\zeta}{\zeta} e_q^2 f_g(\xi) D_q(\zeta)\hat{\sigma}_{qg,k}\delta\left(\left(1-\frac{\xi}{x}\right)\left(1-\frac{\zeta}{z}\right)-\frac{q_T^2}{Q^2}\right) \tag{4.61}$$

with

$$\hat{\sigma}_{qg,k} \equiv \sum_{i=0}^{\infty}\varepsilon^i \hat{\sigma}_{qg,k,i} \tag{4.62}$$

and the non-zero coefficients

$$\hat{\sigma}_{qg,1,0} = 2T_R\frac{x}{\xi}\left(1-\frac{x}{\xi}\right)\left[\frac{Q^2}{q_T^2}\left(\frac{\xi^2\zeta^2}{x^2z^2}-2\frac{\xi\zeta}{xz}+2\right)+10-2\frac{\xi}{x}-2\frac{\zeta}{z}\right] \tag{4.63a}$$

$$\hat{\sigma}_{qg,1,1} = 2T_R\frac{Q^2}{q_T^2}\frac{\zeta^2}{z^2}\left(1-\frac{\xi}{x}\right) \tag{4.63b}$$

$$\hat{\sigma}_{qg,2,0} = 16T_R\frac{x}{\xi}\left(1-\frac{x}{\xi}\right) \tag{4.63c}$$

$$\hat{\sigma}_{qg,2,1} = 2T_R\frac{Q^2}{q_T^2}\left(1-\frac{x}{\xi}\right)\left[2\frac{\zeta^2}{z^2}\left(\frac{x}{\xi}-1\right)+\frac{\xi}{x}\left(\frac{\zeta^2}{z^2}-2\frac{\zeta}{z}+2\right)\right] \tag{4.63d}$$

$$\hat{\sigma}_{qg,2,2} = 2T_R\frac{Q^2}{q_T^2}\frac{\zeta^2}{z^2}\left(1-\frac{\xi}{x}\right) \tag{4.63e}$$

$$\hat{\sigma}_{qg,3,0} = 2T_R\frac{x}{\xi}\left(1-\frac{x}{\xi}\right)\frac{2}{q_TQ}\left[2(Q^2+q_T^2)-Q^2\frac{\xi\zeta}{xz}\right] \tag{4.63f}$$

$$\hat{\sigma}_{qg,3,1} = -2T_R\frac{q_T}{Q}\left(1-\frac{x}{\xi}\right)^2\left(1-2\frac{z}{\zeta}\right) \tag{4.63g}$$

$$\hat{\sigma}_{qg,4,0} = 8T_R\frac{x}{\xi}\left(1-\frac{x}{\xi}\right) \tag{4.63h}$$

In order to regularize the expression for $q_T \to 0$ it is again necessary to adjust the prefactor to match the global prefactor of the leading order. After using the expansion formula of the delta distribution the regularization can be performed including the renormalization of the PDF and the FF, for this channel with the appropriate splitting function P_{qg}, see equation (B.1e). After setting ε to zero the result can be splitted up similarly to the gq channel

$$\frac{\mathrm{d}\sigma_{qg}}{\mathrm{d}\mathcal{R}_4} = e_q^2 \frac{2\pi(4\pi)^4\alpha^2}{zQ^2}\frac{\alpha_s}{2\pi}\big[X_{qg}^{(1)}\delta(q_T^2) + X_{qg}^{(2)} + Y_{qg}\big] \tag{4.64}$$

where $X_{qg}^{(1)}$ collects all terms that contribute at $q_T = 0$ only, $X_{qg}^{(2)}$ includes all terms that are divergent as $q_T \to 0$ and Y_{qg} contains all the rest. Their analytical expressions are

$$X_{qg}^{(1)} = \mathcal{A}_1\left[\ln\left(\frac{Q_T^2}{\mu^2}\right) D_q(z)\big[f_g(x)\otimes P_{qg}\big] + T_R D_q(z)\big[f_g(x)\otimes c_g^0\big]\right] \tag{4.65}$$

$$\begin{aligned} X_{qg}^{(2)} = {} & \frac{\mathcal{A}_1}{(q_T^2)_+} D_q(z)\big[f_g(x)\otimes P_{qg}\big] + \frac{T_R}{Q^2}\mathcal{A}_1 \ln\left(\frac{Q^2}{q_T^2}\right)\big(z f_g(x) D_q'(z) - f_g(x) D_q(z)\big) \\ & + \mathcal{A}_3 \frac{2T_R}{Q q_T} D_q(z)\big[f_g(x)\otimes(2c_g^2 - c_g^1)\big] \end{aligned} \tag{4.66}$$

$$\begin{aligned} Y_{qg} = \mathcal{A}_1 \frac{T_R}{Q^2}\Bigg[& 2D_q(z)\big(4\big[f_g(x)\otimes c_g^2\big] - \big[f_g(x)\otimes c_g^1\big]\big) \\ & + xzD_q'(z)\frac{1}{T_R}\big[f_g'(x)\otimes P_{qg}\big] + xD_q(z)\big[f_g'(x)\otimes\big(c_g^0 - 2c_g^2\big)\big] \\ & + 2\big(z f_g(x) D_q'(z) - f_g(x) D_q(z)\big) \\ & + zD_q'(z)\big[f_g(x)\otimes\big(3\tilde{P}^2 - 8\tilde{P}^3 + 6\tilde{P}^4\big)\big] + z f_g(x)\big[D_q'(z)\otimes\big(2\tilde{P}^1 - \tilde{P}^0\big)\big] \\ & + D_q(z)\big[f_g(x)\otimes\big(\tilde{P}^2 + 4\tilde{P}^3 - 6\tilde{P}^4\big)\big] + f_g(x)\big[D_q(z)\otimes\big(\tilde{P}^0 + -2\tilde{P}^2\big)\big] \\ & + \frac{q_T^2}{Q^2}\Bigg(xzD_q'(z)\big[f_g'(x)\otimes\big(8c_g^2 - 2c_g^1\big)\big] + xD_q(z)\big[f_g'(x)\otimes\big(2c_g^1 - 10c_g^2\big)\big] \\ & \quad + \left[2 + \ln\left(\frac{Q^2}{q_T^2}\right)\right]\big(6z f_g(x) D_q'(z) - 8 f_g(x) D_q(z)\big) \\ & \quad + zD_q'(z)\big[f_g(x)\otimes\big(2\tilde{P}^2 - 20\tilde{P}^3 + 24\tilde{P}^4\big)\big] \\ & \quad + z f_g(x)\big[D_q'(z)\otimes\big(-2\tilde{P}^0 + 8\tilde{P}^1\big)\big] - 8 f_g(x)\big[D_q(z)\otimes\tilde{P}^2\big] \\ & \quad + D_q(z)\big[f_g(x)\otimes\big(-2\tilde{P}^2 + 24\tilde{P}^3 - 30\tilde{P}^4\big)\big]\Bigg)\Bigg] \end{aligned}$$

$$
\begin{aligned}
&+ (2\mathcal{A}_2 + \mathcal{A}_4)\frac{4T_R}{Q^2} \\
&\quad\times \Bigg[D_q(z)\big[f_g(x)\otimes c_g^2\big] \\
&\qquad + \frac{q_T^2}{Q^2}\Bigg(\big(xzD_q'(z) - xD_q(z)\big)\big[f_g'(x)\otimes c_g^2\big] \\
&\qquad\qquad + \left[2 + \ln\left(\frac{Q^2}{q_T^2}\right)\right]\big(zf_g(x)D_q'(z) - f_g(x)D_q(z)\big) \\
&\qquad\qquad + zf_g(x)\big[D_q'(z)\otimes \tilde{P}^1\big] - zD_q'(z)\big[f_g(x)\otimes\big(2\tilde{P}^3 - 3\tilde{P}^4\big)\big] \\
&\qquad\qquad + D_q(z)\big[f_g(x)\otimes\big(2\tilde{P}^3 - 3\tilde{P}^4\big)\big] - f_g(x)\big[D_q(z)\otimes\tilde{P}^2\big]\Bigg)\Bigg] \\
&+ 2\mathcal{A}_3 T_R\frac{q_T}{Q^3} \\
&\quad\times \Bigg[2D_q(z)\big[f_g(x)\otimes c_g^2\big] + \left[2 + \ln\left(\frac{Q^2}{q_T^2}\right)\right]\big(zf_g(x)D_q'(z) - 2f_g(x)D_q(z)\big) \\
&\qquad + xzD_q'(z)\big[f_g'(x)\otimes\big(2c_g^2 - c_g^1\big)\big] - 2xD_q(z)\big[f_g'(x)\otimes c_g^2\big] \\
&\qquad - zD_q'(z)\big[f_g(x)\otimes\big(6\tilde{P}^3 - 6\tilde{P}^4 - \tilde{P}^2\big)\big] - zf_g(x)\big[D_q'(z)\otimes\big(\tilde{P}^0 - 2\tilde{P}^1\big)\big] \\
&\qquad + 2D_q(z)\big[f_g(x)\otimes\big(2\tilde{P}^3 - 3\tilde{P}^4\big)\big] - 2f_g(x)\big[D_q(z)\otimes\tilde{P}^2\big] \\
&\qquad + \frac{q_T^2}{Q^2}\Bigg(\big(xzD_q'(z) - xD_q(z)\big)\big[f_g'(x)\otimes c_g^2\big] \\
&\qquad\qquad + \left[2 + \ln\left(\frac{Q^2}{q_T^2}\right)\right]\big(zf_g(x)D_q'(z) - f_g(x)D_q(z)\big) \\
&\qquad\qquad + \big(zD_q'(z) - D_q(z)\big)\big[f_g(x)\otimes\big(3\tilde{P}^4 - 2\tilde{P}^3\big)\big] \\
&\qquad\qquad + zf_g(x)\big[D_q'(z)\otimes\tilde{P}^1\big] - f_g(x)\big[D_q(z)\otimes\tilde{P}^2\big]\Bigg)\Bigg]
\end{aligned}
\tag{4.67}
$$

4.6 Result and Conclusions

In this chapter we have presented a full NLO calculation of the SIDIS process that is capable to describe the cross section for small or even zero transverse momentum q_T. While this has been done before decades ago, concentrating on the non-regular terms as $q_T \to 0$, which are then subsequently resummed up to all orders to get a finite expression even for vanishing q_T, our calculation includes for the first time not only the non-regular but all divergent terms as $q_T \to 0$. This is mainly achieved by a

higher-order expansion of the overall delta distribution, which limited the access to the sub-dominant but still divergent terms. Now it can be guaranteed that all divergent terms are taken into account. The overall analytical result of this chapter is given by the sum over the three channels that contribute at NLO and all possible quark flavours

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\mathcal{R}_4} \equiv \sum_{q\in\mathcal{Q}\cup\overline{\mathcal{Q}}} \left[\frac{\mathrm{d}\sigma_{qq}}{\mathrm{d}\mathcal{R}_4} + \frac{\mathrm{d}\sigma_{gq}}{\mathrm{d}\mathcal{R}_4} + \frac{\mathrm{d}\sigma_{qg}}{\mathrm{d}\mathcal{R}_4} \right] \tag{4.68}$$

where the result of the single channels have been reported in equations (4.36), (4.51) and (4.64) for the qq, gq and qg channel respectively. The sum runs over all quark and anti-quark flavours. For this purpose we use the set $\mathcal{Q}$ of all quark flavours defined in equation (1.3).

This chapter can be seen as a starting point for a more complete description of the physics of small transverse momenta. Future work might incorporate it to generate a more general resummation formalism. In order to do this, we want to make some remarks. Our careful calculation revealed divergent terms that depend not on the PDF and FF of the process but their derivatives. In order to handle these it might be necessary to perform a Mellin transformation (see appendix E), which allows to perform a integration by parts to shift the derivatives on analytically manageable expressions (namely powers of the transformed variable). The transformation is desirable anyway, as it turns all the convolutions into simple products. Additionally it is known from the original resummation formalism that it has to be handled in Fourier space of the transverse momentum, the impact parameter space. A very simple attempt would be to exponentiate all divergent terms without having regard for higher-order implications, as it is performed in [224]. Unfortunately we encounter an integral in the exponent that we cannot solve analytically in such a way that it would be numerically possible to perform the Fourier and double Mellin inverse transforms.

A byproduct of the calculation is the gain of knowledge of the Y term that contains all finite (and in the literature all regular) terms. This is particularly helpful for a subsequent matching procedure once a resummation has been performed, independent of whether the resummation accounts for non-regular or divergent terms.

Finally the key step—the expansion of the delta distribution—does not depend on the SIDIS process, which makes it straightforward to apply the calculation also for other processes like the Drell-Yan process or electron-positron annihilation. This is also not limited to unpolarized cross sections but can be extended to polarized cross sections as well.

Chapter 5

Parton distribution functions in Mellin space

> In science if you know what you are doing you should not be doing it. In engineering if you do not know what you are doing you should not be doing it. Of course, you seldom, if ever, see either pure state.
>
> — *Richard Hamming*

In this chapter we will present various methods of describing the parton distribution functions and fragmentation functions in Mellin space. We will give an overview over the most common technique and then introduce a method, that has not been used for this purpose in the past. We will discuss its benefits and also its drawbacks in detail. Finally we will present numerical implementations in Fortran and C++, showing their public interfaces and discussing their ability for future improvements and extendability.

5.1 Introduction

Since all particles interacting via the strong force can only be found in bound states, one of the most demanding and important tasks to get reliable results from perturbative QCD is the treatment of nonperturbative parts. As has been pointed out in the introductory chapter 1 one key point, one may say the starting point, of perturbative QCD is to disentangle short-distance and long-distance interactions. While the former can be calculated inside the framework, the latter have to be gathered by phenomenological studies. For collinear factorization the nonperturbative parts, that cannot be retrieved directly by the theory, are the collinear parton distribution functions, which are usually referred to simply as the parton distribution functions (PDFs).

In the last decades several vendors for parton distribution functions produced a large variety of sets that are distributed and can be implemented for numerical studies incorporating perturbative QCD. Most of them are concerned with the unpolarized proton, as are the perhaps widest known sets of the CTEQ group [82, 86–88], MSTW and MMHT [89, 90] and the NNPDF collaboration [91, 92]. These are updated regularly

and offer a wide range of different error sets, number of flavours and incorporated electroweak effects for the available perturbative orders. Beside a lot of other sets have been developed, of especial interest are perhaps the sets constructed using only the data of one specific collider, as there are the sets of HERA [248] or the Jefferson Laboratory [249, 250]. Additionally there has been much work performed to describe the polarized proton, starting with the early publications [251–253] and also up-to-date sets like [254]. Although the proton is the particle, which is investigated most, descriptions of other particles as pions [212, 213, 255] and large nuclei, for example [256], are available. Naturally the second long-distance process, fragmentation, describing the fragmentation process of final state partons into observable hadrons, has also been modeled by various groups [257–260]. For the purpose of this topic, fragmentations functions and parton distribution functions are similar. We will therefore reference to both as parton distribution functions or PDFs. Finally there is also ongoing work beside the collinear factorization on creating a reasonable description for generalized parton distribution functions (GPDs) [261] and transverse momentum dependent parton distribution functions (TMDs). For the latter a remarkable advance has been obtained in the last years to perform a global fit [79, 80, 207, 262]. We will not address GPDs or TMDs in this chapter. But in principle the technique that will be introduced can also be applied to them.

The large amount of different vendors and sets required a reference book to follow recent developments. This need has been satisfied by the Durham high energy particles data project on their website [263][1], directing to the several original sources of the PDF provider. A second need arising by the large amount of vendors is an unified interface to enable easy comparisons between different sets for all computations relying on them. This is offered by the LHAPDF library [157]. It provides an universal interface and makes many different sets easily interchangeable and comparable in every customer application. Until LHAPDF version 5 the program has been written in Fortran, presumably because most customers also used Fortran in that time, but offering also a C++ interface. Since 2013 the program is distributed as version 6 written in C++, superseding version 5 in every manner. Still Fortran is taken account of by an interface that is identical to the one of version 5. By its nature LHAPDF is also a tremendous reference for existing parton distributions. However, it should be mentioned, that not all existing sets are available. First LHAPDF only supports collinear PDFs. Second also some collinear sets are not available at LHAPDF, as is the DSS set [258, 259] for collinear fragmentation functions.

The common procedure to produce PDFs is to find an appropriate functional form for the PDFs in terms of the longitudinal momentum fraction of the parton inside the hadron x (or of the hadron 'inside' the parton in case of the fragmentation functions) at one specific scale μ_0^2. This functional form is typically driven by the boundary conditions, that $xf_i(x,\mu^2) \to 0$ for $x \to 1$ and for $x \to 0$, where f_i is the parton

[1] The website has been superseded lately, but the current version does (at least at the moment) not offer the list, that is available at the reference.

distribution function for parton i (sometimes also a superscript indicates the parent hadron, which we suppress for clarity). Therefore the generic PDF parameterization is

$$xf_i(x,\mu_0^2) = c_{i,0}x^{c_{i,1}}(1-x)^{c_{i,2}}P_i(x) \tag{5.1}$$

where $P_i(x)$ is a polynomial. However, the details are much more sophisticated, as is for example the MMHT parameterization [90]

$$xf_i(x,\mu_0^2) = c_{i,0}x^{c_{i,2}}(1-x)^{c_{i,3}}\left[1+\sum_{k=1}^{4}c_{k,i,4}T_k^{\mathrm{Ch}}(1-2x^k)\right] \tag{5.2}$$

with $T_k^{\mathrm{Ch}}(y)$ being Chebyshev polynomials. Most parametrizations of the CTEQ group do even alter the generic parameterization [86]

$$xf_i(x,\mu_0^2) = c_{i,0}x^{c_{i,1}}(1-x)^{c_{i,2}}\,\mathrm{e}^{c_{i,3}x}\Big(1+\mathrm{e}^{c_{i,4}}\,x\Big)^{c_{i,5}} \tag{5.3}$$

To obtain the PDFs at an arbitrary scale μ^2 the DGLAP evolution presented in section 1.5 is applied. With this machinery a χ^2 fit is performed to carefully selected data to determine the parameters of the initial parameterization.

An alternative way is applied by the NNPDF collaboration using neural networks. The technique contains of generating replicas of the experimental data (typically ≈ 1000 to achieve a reasonable error estimate) with artificial data generated by Monte Carlo inputs. The functional form is extremely over-parameterized. To avoid overlearning, each replica divides its data into a training and a validation set. The fit is performed in the usual manner only on the training set, the validation set checks every fit-iteration that no overlearning occurs, typically by stopping the fit, if the new parameters are worse than the previous ones.

As different are the techniques and implementation details of the various set vendors, all of them provide their results in the same way. As the functions are dependent on x and μ^2, the vendors offer grid files, spanning the supported (x,μ^2) plane, and a program containing an appropriate user interface and interpolation routines to access (x,μ^2) points in between the grid points. Some vendors also deliver an extrapolation routine to reach areas not included in their grid.

For many calculations in pertubative QCD it is beneficial to transform the analytical expressions into Mellin space (for details about the Mellin transform see appendix E). Typical reasons are the need to perform an additional DGLAP evolution, resummation of soft gluon logarithms at edges of the phase space, where the soft gluon kernels are analytically only available in Mellin space, or simple the demand to turn multiple convolutions into products. In any case one needs the parton distribution functions also in Mellin space. Interestingly many fits itself are performed in Mellin space, as for example GRV [255] or DSS [258, 259]. Unfortunately it is difficult use their fit parameters directly, because by construction these are given at one initial scale only. To obtain reliable results it would be necessary to implement the DGLAP evolution in the exact same way, as was done for the fitting procedure.

The common way to extract Mellin PDFs from a given set at one fixed scale, is a second fit. One chooses a specific generic parameterization like in equation (5.1) (see for example [264])

$$f_i(x,\mu_0^2) = c_{i,0}x^{c_{i,1}}(1-x)^{c_{i,2}}(1+c_{i,3}\sqrt{x}+c_{i,4}x+\ldots) \tag{5.4}$$

with an appropriate number of terms inside the polynomial (note that for the following the polynomial can contain every sort of power). This form translates into beta functions $B(N,M)$ in Mellin space

$$\begin{aligned} f_i(N,\mu_0^2) &= \int_0^1 \mathrm{d}x\, x^{N-1} f_i(x,\mu_0^2) \\ &= c_{i,0}\Big[B(N+c_{i,1},c_{i,2}+1) + c_{i,3}B(N+c_{i,1}+0.5,c_{i,2}+1) \\ &\qquad + c_{i,4}B(N+c_{i,1}+1,c_{i,2}+1)+\ldots\Big] \end{aligned} \tag{5.5}$$

which can be calculated numerically without need of an integration and thus makes the PDFs available in Mellin space. However, this procedure has to be followed for every parton flavour and for every PDF set separately. The quality of the fit depends strongly on the chosen parameterization, which might also be dependent on the parton flavour. Furthermore results depend on the fit which exacerbates reproduction, unless the fit is specified exactly. Most likely the fit will perform better, the closer the chosen parameterization is to the parameterization of the PDF set. It is worth noticing that for example the CTEQ parameterization in equation (5.3) can never be matched, because the Mellin transform of the exponential leads to a result, that cannot be expressed without the need of an additional integration, which in most cases could have been the convolution with the original set in the first place. Still, with sufficient effort very good results can be obtained.

To achieve also a scale dependence in Mellin space on can add a logarithmic dependence on the parameters $c_{i,j}$, like

$$c_{i,j} \to c_{i,j} + d_{i,j}\ln\left(\frac{\mu^2}{\mu_0^2}\right) \tag{5.6}$$

and new parameters $d_{i,j}$ for every parton flavour i. Most parameterizations are more sophisticated, adding fit parameters also inside the logarithm. However, the main goal is to simulate the logarithmic DGLAP evolution and find a reasonable description of the PDFs set for the kinematics under consideration. The resulting parameterization in Mellin space can be used for every scale that is covered by the fit, but naturally the precision of the fit will suffer slightly.

In the following sections we will introduce a new technique to describe parton distribution functions in Mellin space, by using simple cubic splines to interpolate the PDFs. Then we will demonstrate their superior precision over existing Mellin

PDF descriptions, especially their uniform quality for all different parton flavours. Subsequently we will demonstrate the use of the method using the calculation of a resummed Drell-Yan cross section differential in the mass of the intermediate vector boson and discuss possible benefits in analytical calculations. Finally we will give a brief introduction into two libraries (Fortran and C++) that provide an easy to use access to the Mellin PDFs.

5.2 Cubic spline approximation

A very common problem in several disciplines, especially in engineering, is the search for a smooth interpolation between fixed points or an approximation of a known but difficult functional form. While a pure polynomial interpolation leads to sometimes uncontrollable oscillations, the only slightly more difficult approach of splines yields typically to a more stable interpolation. In particular if the goal is the approximation of a known function splines typically give a much better result because of their flexibility. Splines are classified by the degree of the polynomials in between the sampling points. A higher degree guarantees smoother transitions between to concatenated polynomials (in the sense of differentiability at the sampling points), but needs also higher computational cost to calculate the splines and, as we will see, also higher computational cost to use them in Mellin space. A good balance between these concerns is a cubic spline, which is therefore our choice.

In case of parton distribution functions, every distributed set is given as grid in x, μ^2 space. In the following we will consider a fixed scale μ^2 only. We therefore will drop the scale argument. Further the procedure does not depend on the parton flavour and we can drop the subscript to indicate the flavour, too. For a grid with n sampling points in x direction, which means we have the points $f_j \equiv f(x_j)$ with $j \in \{1, \dots, n\}$, we define the cubic spline by its $n-1$ polynomials defined in between the f_j ($k \in \{2, \dots, n\}$)

$$s_k : [x_{k-1}, x_k] \to \mathbb{R} \tag{5.7a}$$

$$x \mapsto \frac{p_k(x - x_{k-1})^3 + p_{k-1}(x_k - x)^3}{6h_k} + \left[\frac{f_k}{h_k} - \frac{h_k}{6}p_k\right](x - x_{k-1}) + \left[\frac{f_{k-1}}{h_k} - \frac{h_k}{6}p_{k-1}\right](x_k - x) \tag{5.7b}$$

with $h_k \equiv x_k - x_{k-1}$ and the parameters p_j. Latter are constrained by the demands of continuity and differentiability at the sampling points

$$s_k(x_{k-1}) = f_{k-1} \tag{5.8a}$$

$$s_k(x_k) = f_k \tag{5.8b}$$

$$\partial_x s_k(x_k) = \partial_x s_{k+1}(x_k) \tag{5.8c}$$

$$\partial_x^2 s_k(x_k) = \partial_x^2 s_{k+1}(x_k) \tag{5.8d}$$

Further one has to decide the behavior at the boundaries. One common choice is what we will call a 'natural' spline which implies

$$\partial_x^2 s_2(x_1) = \partial_x^2 s_n(x_n) = 0 \tag{5.9}$$

Another common choice is to fix the first derivatives to be equal to the analytical derivative of the function

$$\partial_x s_2(x_1) = \partial_x f_1 \tag{5.10a}$$
$$\partial_x s_n(x_n) = \partial_x f_n \tag{5.10b}$$

We will call this choice 'fixed'. Our ansatz in equation (5.7) fulfills three of the constraints by construction ((5.8a), (5.8b) and (5.8d)). Equation (5.8c) results in a set of $n-2$ linear independent equations ($i \in \{2, \dots, n-1\}$)

$$\frac{1}{6}p_{i-1}(x_i - x_{i-1}) + \frac{1}{3}p_i(x_{i+1} - x_{i-1}) + \frac{1}{6}p_{i+1}(x_{i+1} - x_i) = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} - \frac{f_i - f_{i-1}}{x_i - x_{i-1}} \tag{5.11}$$

The final two equations are given by the boundary conditions. In case of a natural spline, equation (5.9) translates to

$$p_1 = p_n = 0 \tag{5.12}$$

For a fixed spline one receives

$$p_1 = -\frac{p_2}{2} + \frac{3}{x_2 - x_1}\left[\frac{f_2 - f_1}{x_2 - x_1} - \partial_x f_1\right] \tag{5.13a}$$
$$p_n = -\frac{p_{n-1}}{2} + \frac{3}{x_n - x_{n-1}}\left[\partial_x f_n - \frac{f_n - f_{n-1}}{x_n - x_{n-1}}\right] \tag{5.13b}$$

If the function is not known analytically (the spline interpolates a pure grid), the derivative can be approximated by the difference quotient of the two sample points at the boundary, which simplifies equations (5.13) to

$$p_1 = -\frac{p_2}{2} \tag{5.14a}$$
$$p_n = -\frac{p_{n-1}}{2} \tag{5.14b}$$

To solve the full set of coupled linear equations ((5.11) and depending on the spline type (5.12), (5.13) or (5.14)), we will use the Doolittle decomposition, sometimes also referred to as LU decomposition. The procedure and the simplifications to the general algorithm because of the especially easy form of the system is described in appendix F.

In figure 5.1 we show a comparison of the original CT10n central set [82] to calculated natural splines over the full range of longitudinal momentum fractions x provided by the set for the u quark, the $\bar{d}$ quark and the gluon. The splines use the native grid of

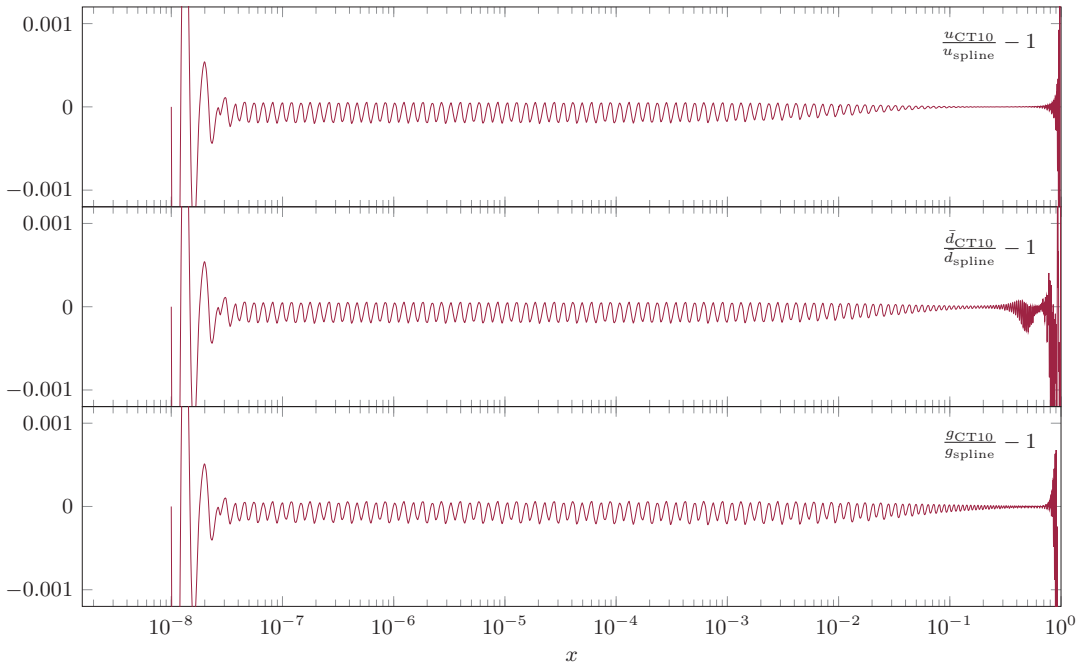


Figure 5.1: Comparision of the interpolation spline with the original PDF routine (CT10n central set [82]) over the whole provided range for the longitudinal momentum fraction x. The top, middle and bottom panel show the comparision for the u quark, the $\bar{d}$ quark and the gluon respectively. The scale for all panels is $\mu = 4.75\,\mathrm{GeV}$.

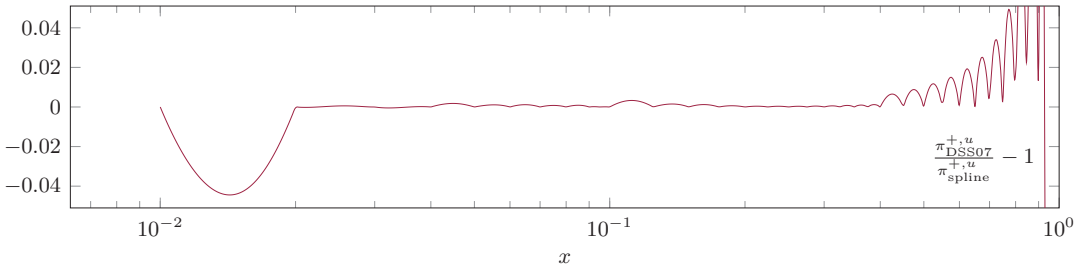


Figure 5.2: Comparision of the interpolation spline with the original fragmentation function routine (DSS07 set using the DSS14 interface [258, 259]) by taking the π^+ distribution inside an u quark. The plot covers the whole provided range for the longitudinal momentum fraction x. The scale is $\mu = 10\,\text{GeV}$.

the CT10n set at sample points. The scale of the PDFs is chosen to be $\mu = 4.75\,\text{GeV}$, which equals also one μ in the CT10n grid. Both choices ensure, that the spline uses only data that are directly coming from the PDF fit and the interpolation routine of the set does not influence the spline sample points. We will always use natural splines, because the approximations at the boundaries (5.14) seems ambitious, as the derivatives of the PDFs change rapidly in this regions. As the splines are equal to the input at the sample points, the comparison is in fact a comparison between the interpolation routine provided by the PDF group and the spline interpolation. It is worth noting that none of both will be equal to the functional form including full DGLAP evolution, as is applied in the PDF fitting procedure.

In the top panel of figure 5.1 the distributions of the u quark are compared. We see a nearly perfect agreement between the native interpolation and the spline. In the main range of the distribution the differences are in sub per mill level. The splines oscillate around the PDF set, which is a typical behavior. At the boundaries of the grid, the quality of the spline suffers, indicating that the PDF itself becomes awkward, as these regions are difficult to restrict in PDF fits. Summarized the spline quality is best in the region where also the PDFs are known best and which is most likely the dominant region for perturbative calculations.

The middle and the bottom panel of figure 5.1 show the corresponding graphs for the $\bar{d}$ quark and the gluon respectively. In the majority of the x range the ratio behaves exactly as discussed for the u quark. For the gluon a small modulation is apparent for $x \in \left[10^{-7}, 10^{-2}\right]$. The $\bar{d}$ quark, which is especially difficult to describe with a fit performed with a parameterization similar to (5.4), is also slightly worse described for large x, again indicating a less smooth behavior of the underlying sample points. But overall all flavours are described equally well and uniformly.

Figure 5.2 shows the similar plot for the DSS routine [258, 259], where we selected the π^+ distribution inside the u quark at scale $\mu = 10\,\text{GeV}$. As before the latter choice ensures no interference of the interpolation routine shipped with the DSS14 set with the spline interpolation, as the sample points equal grid points that directly originate by the DSS fit. As demonstrated in figure 5.1 all other choices of the final state partons

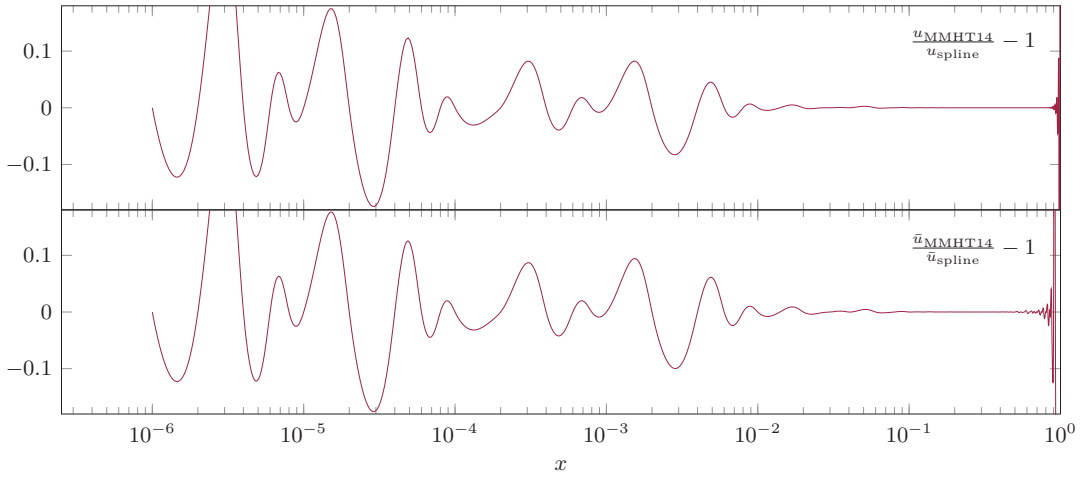


Figure 5.3: Comparision of the interpolation spline with the original MMHT routine [89, 90]. The top panel shows the comparison for the u quark distribution, the bottom panel for the $\bar{u}$ quark distribution. The plot covers the whole provided range for the longitudinal momentum fraction x. The scale of both PDFs is $\mu = 10\,\text{GeV}$.

or hadrons will result in similar plots and it is therefore sufficient to analyze only one representative. The first point we recognize is, that the difference between the result of the DSS interpolation routine and the spline is still small, although not in sub per mill level. This is expected, as the sample points are much more scattered as they are for the CTEQ sets. Again the behavior at the boundaries of the supported longitudinal momentum fraction range is more difficult, as the original fit is less constrained. An entirely new feature is the bumping behavior, which is also apparent in the region where DSS and spline interpolation agree well, although less pronounced. As it turns out, this behavior is due to the interpolation routine of DSS, which simply fails to interpolate a smooth curve through the grid points. In this setup it is reasonable that the spline interpolation is closer to the analytical expression used in the DSS fit at their own interpolation routine. For a direct use of the DSS routine the mistake is negligible because it occurs in the highly suppressed region $x \to 1$. However, for all attempts to describe the set with Mellin transformable expressions like (5.4) that means a residual impact, as long as it is not taken care of checking against the grid points only. This will ultimately lead to an overall worse description of the set.

Now we consider the MMHT set [90] in figure 5.3, showing the comparison of the original set for the u and the $\bar{u}$ quark in the top and bottom panel respectively. The scale $\mu = 10\,\text{GeV}$ is again chosen to match the underlying MMHT grid, as are the x sample points for the spline. We can see immediately that the interpolation is much worse than for the previous examined sets. The main reason will be the again very sparse grid and therefore sample points. However, in the region that is well constrained by experimental data, both interpolations are again equal up to sub per mill level. And in the light of the discussion to the DSS interpolation, one can state that it is at least

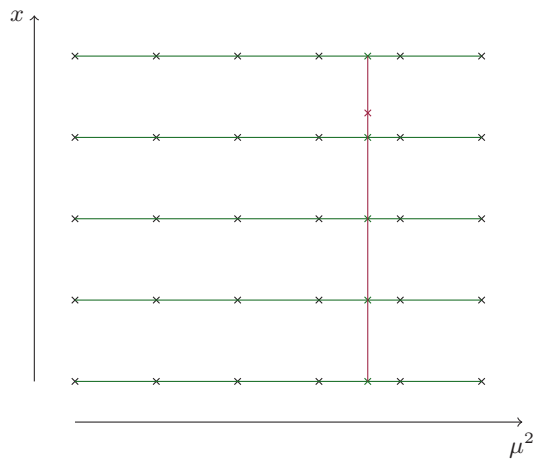


Figure 5.4: Illustration of an interpolation by splines in two dimensions. The black crosses represent the initial grid. The red cross is the point where the interpolation is needed. In a first step several splines are calculated (green lines) to establish a new grid (green crosses). Then a spline is calculated to finally interpolate the function at the desired point (red line).

not obvious which interpolation routine is doing the better job. From a technical point of view the splines are likely to be superior because of their flexibility between two sample points. On the other hand this may smoothen the distribution, while also the original analytical form used in the MMHT fit is more turbulent. However, it is clear that the difference will diminish if more sample points are taken into account, although this combines two consecutive interpolations. A second observation is that for the $\bar{u}$ quark distribution and large x one can see a premature decrease of similarity of the interpolations compared to the u quark case. This is not as we have seen for $\bar{d}$ of the CTEQ set because of the more difficult form to interpolate. Instead the origin is a sign change of the MMHT distribution. While in the naive parton model this seems unphysical, higher orders allow negative PDFs, as long as the sum rules are satisfied (the integral stays positive). However, the sign change causes in figure 5.3 the indication of a very large difference between the interpolations, which is not true, as only the nulls are slightly shifted.

Finally we want to discuss the option of interpolating the PDFs also in both dimensions, the longitudinal momentum fraction x as well as the scale μ^2. In principle this is done simply by splitting the computation into two parts, as illustrated in figure 5.4. First one calculates a spline for every point in one dimension. The resulting splines are taken as input for the final spline in the other dimension. The choice which dimension should be taken for the final spline depends on the specific need. For PDFs it is most likely that, if a one dimensional spline is not sufficient, one needs only one specific point. For this scenario the decisive factor is the computation time. As the Doolittle algorithm is of $\mathcal{O}(n)$ the dimension in which multiple splines are defined (in figure 5.4

the μ^2 dimension) is not strictly determined by the number of points in it. But it is reasonable that it is superior to define as less splines as possible. However, this should be tested. In this thesis we did not use or implement multidimensional splines, although it would be straightforward. But as we want to use the spline to integrate cross sections in Mellin space, the computation of the two dimensional splines should occur inside the integrand, which could increase the computational load dramatically (depending on the percentage it would take of the total integrand computation). Therefore we will restrict ourselves to one dimensional splines and situations, where only one scale is needed. Nevertheless it is worth to think about the two dimensional splines to replace also the interpolation routine in scale of the PDF vendors. The very same procedure can be applied to higher dimensions, opening the spline interpolation also to GPDs and TMDs, which depend on three parameters.

5.3 Splines in Mellin Space

To perform the Mellin transformation we have to notice, that the complete spline is defined by the piecewise definitions given in equation (5.7) accompanied by Heaviside functions

$$s(x) = \sum_{k=2}^{n} s_k(x)\Theta(x_k - x)\Theta(x - x_{k-1}) = \sum_{k=2}^{n}\sum_{m=0}^{3} e_{m,k}x^m\Theta(x_k - x)\Theta(x - x_{k-1}) \tag{5.15}$$

where we disentangled the monomials with the coefficients

$$e_{m,k} = \begin{cases} d_k & \text{for } m = 0 \\ c_k & \text{for } m = 1 \\ b_k & \text{for } m = 2 \\ a_k & \text{for } m = 3 \end{cases} \tag{5.16}$$

and

$$a_k = \frac{p_k - p_{k-1}}{6(x_k - x_{k-1})} \tag{5.17a}$$

$$b_k = \frac{p_{k-1}x_k - p_k x_{k-1}}{2(x_k - x_{k-1})} \tag{5.17b}$$

$$c_k = \frac{f_k - f_{k-1}}{x_k - x_{k-1}} + \frac{1}{6}\big(x_k - x_{k-1}\big)\big(p_{k-1} - p_k\big) + \frac{p_k x_{k-1}^2 - p_{k-1}x_k^2}{2\big(x_k - x_{k-1}\big)} \tag{5.17c}$$

$$d_k = \frac{f_{k-1}x_k - f_k x_{k-1}}{x_k - x_{k-1}} + \frac{1}{6}\left[\big(x_k - x_{k-1}\big)\big(p_k x_{k-1} - p_{k-1}x_k\big) + \frac{p_{k-1}x_k^3 - p_k x_{k-1}^3}{x_k - x_{k-1}}\right] \tag{5.17d}$$

Then the transformation is straight forward

$$\begin{aligned} S(N) &= \int_0^1 \mathrm{d}x\, x^{N-1} \sum_{k=2}^{n} s_k(x)\Theta(x_k - x)\Theta(x - x_{k-1}) \\ &= \sum_{k=2}^{n} \int_{x_{k-1}}^{x_k} \mathrm{d}x \sum_{m=0}^{3} x^{N-1+m} e_{m,k} \end{aligned} \tag{5.18}$$

We can simplify the analytic expression in equation (5.18) sorting the coefficients to the same sample point

$$\begin{aligned} S(N) &= \sum_{k=2}^{n} \sum_{m=0}^{3} \frac{e_{m,k}}{N+m} (x_k^{N+m} - x_{k-1}^{N+m}) \\ &= \sum_{m=0}^{3} \frac{1}{N+m} \left[e_{m,n} x_n^{N+m} - e_{m,2} x_1^{N+m} + \sum_{k=2}^{n-1} x_k^{N+m} (e_{m,k} - e_{m,k+1}) \right] \\ &= \sum_{k=1}^{n} \sum_{m=0}^{3} g_{m,k} \frac{x_k^N}{N+m} \end{aligned} \tag{5.19}$$

with the final set of coefficients

$$g_{m,k} = x_k^m \cdot \begin{cases} -e_{m,2} & \text{for } k = 1 \\ e_{m,n} & \text{for } k = n \\ e_{m,k} - e_{m,k+1} & \text{otherwise} \end{cases} \tag{5.20}$$

Using equations (5.11) reveals that the $g_{m,k}$ are not independent for $k \notin \{1, n\}$

$$g_{2,k} = -g_{1,k} \tag{5.21a}$$

$$g_{3,k} = -g_{0,k} \tag{5.21b}$$

One can therefore disentangle the sum again and write the Mellin moment of the spline as

$$S(N) = \sum_{m=0}^{3} \frac{g_{m,1} x_1^N + g_{m,n} x_n^N}{N+m} + \sum_{k=2}^{n-1} 2 x_k^N \left(\frac{g_{0,k}}{N(N+2)} + \frac{g_{1,k}}{(N+1)(N+3)} \right) \tag{5.22}$$

However, although this is in principle a simplification, we will use the form in equation (5.19) in the following analytical expressions to hold these clear. Nevertheless equations (5.21) may be useful for numerical implementations. The current realizations described in section 5.5 do not use (5.21).

The main result of the transform is that splines are described by simple sums in Mellin space. The coefficients $g_{m,k}$ depend only on the spline parameters p_k and the grid

itself. They are therefore constants for a single spline. All dependence on the Mellin variable N is in powers of the sample points and the denominator in equation (5.19). They have therefore a very simple pole structure, with only four poles, at the integers 0, -1, -2 and -3, which makes them especially handy for the transformation back into x space. This transformation involves a subtlety regarding the integration contour $\mathcal{C}$

$$s(x) = \int_{\mathcal{C}} \frac{\mathrm{d}N}{2\pi\mathrm{i}} x^{-N} S(N) = \int_{\mathcal{C}} \frac{\mathrm{d}N}{2\pi\mathrm{i}} x^{-N} \sum_{k=1}^{n} \sum_{m=0}^{3} \frac{x_k^N g_{m,k}}{N+m} \tag{5.23}$$

Analytically the exact contour plays no role, as long as we do not cross any poles, because we integrate a holomorphic function. But numerically the integral will only converge if we choose a path that is falling off rapidly. This goal cannot be accomplished for all terms in the k sum, because the asymptotic behavior of the integral is dependent on the relative position of x to the sample points x_k

$$x^{-N} x_k^N = \exp\left[N \ln\left(\frac{x_k}{x}\right)\right] \tag{5.24}$$

This means the contour should approach $\mathrm{Re}(N) \to -\infty$ as long as $x < x_k$ and $\mathrm{Re}(N) \to +\infty$ as soon as $x > x_k$. We will call the contours $\mathcal{C}_-$ and $\mathcal{C}_+$ respectively. As a consequence we have to split up the sum over the sample points depending on the value of x

$$\begin{aligned} s(x) &= \sum_{k;x_k>x} \sum_{m=0}^{3} g_{m,k} \int_{\mathcal{C}_-} \frac{\mathrm{d}N}{2\pi\mathrm{i}} \frac{\mathrm{e}^{N \ln\left(\frac{x_k}{x}\right)}}{N+m} + \sum_{k;x_k\le x} \sum_{m=0}^{3} g_{m,k} \int_{\mathcal{C}_+} \frac{\mathrm{d}N}{2\pi\mathrm{i}} \frac{\mathrm{e}^{-N \ln\left(\frac{x}{x_k}\right)}}{N+m} \\ &\equiv s_-(x) + s_+(x) \end{aligned} \tag{5.25}$$

The contour $\mathcal{C}_+$ can be closed without enclosing any pole. So the integral vanishes

$$s_+(x) = \sum_{k;x_k\le x} \sum_{m=0}^{3} g_{m,k} \int_{\mathcal{C}_+} \frac{\mathrm{d}N}{2\pi\mathrm{i}} \frac{\mathrm{e}^{-N \ln\left(\frac{x}{x_k}\right)}}{N+m} = 0 \tag{5.26}$$

The integral following the contour $\mathcal{C}_-$ can be calculated easily by use of the residuum theorem. Every term in the m sum results in one term (because there is only one pole). The sum over k then collapses as it represents a telescope sum

$$s(x) = s_-(x) = \sum_{m=0}^{3} \sum_{k;x_k>x} \frac{x^m}{x_k^m} g_{m,k} = \sum_{m=0}^{3} x^m e_{m,\tilde{k}} \tag{5.27}$$

where $\tilde{k} \equiv \min\{k | x_k > x\}$. This is exactly the same as the spline in its definition (5.15) because the Heaviside functions restrict the splines to their domain $x \in [x_{k-1}, x_k]$, as is implicit in their original definition in equation (5.7).

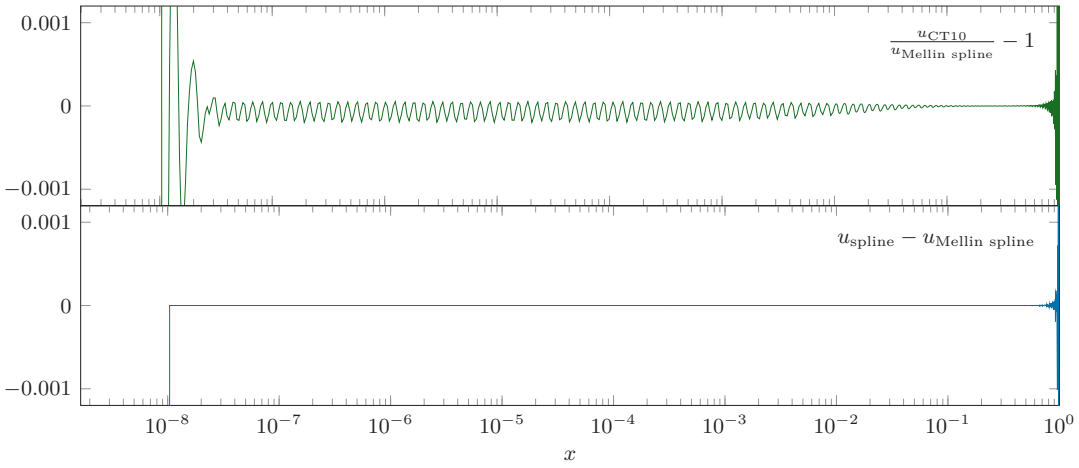


Figure 5.5: Comparision of the spline in Mellin space with the original PDF set (top panel) and with the parent spline in x space (bottom panel) for the same PDF set as shown in figure 5.1. The spline in Mellin space has been transformed into x space by equation (5.25).

	$\{k\|x_k = x\} \in$	s_-/s_x	s_+/s_x
$x = x_1$	s_-	$\frac{1}{5}$	0
	s_+	1	$-\frac{1}{5}$
$x \neq x_1$	s_-	1	0
	s_+	1	0

Table 5.1: Contributions of the sums s_- and s_+ to the numerical value of the spline in x space s_x. We distinct the scenarios of testing the first grid point $x = x_1$ or any other valid point $x \neq x_1$. In both cases we further distinct, whether a the grid point with $x_k = x$ is counted to s_- or s_+.

All the calculations in the section apply similar to non-cubic splines. The degree of the spline will determine the number of summands in the m sum. Changing the degree will therefore linearly increase the computational cost to calculate the spline in Mellin space numerically. Naturally the concrete form of the coefficients $e_{m,k}$ will also change. However, they will always be easily computed from the sample points (x_k, f_k) and the spline parameters p_k.

We will now compare the Mellin space splines to the original PDFs. This is done numerically by applying the transformation in equation (5.25). To achieve a good precision we use 10^5 points in a monte carlo integration tool. As the computational load is quite large, we use the integrator described in chapter 2. The result is shown in figure 5.5. As expected analytically the spline in Mellin space shows the exact same behavior as the spline in x space (compare to upper panel of figure 5.1). In the bottom panel of figure 5.5 we show that it is in fact almost exactly the same as the spline

in x space. The difference that can be seen for $x \to 1$ is only due to the integration error. Because in this region the function is almost zero, the integration error rises up to a few per cent. The only difference that is not of this nature is located at one point: the lowest sample point. This is true for every set and for every flavour. As this is no continuous effect, it is not possible to visualize it in the plot properly. In table 5.1 we document the numerical values of the compared quantities at $x = x_1$ and $x \neq x_1$, where x_1 is the lowest sample point of an arbitrary set. Although a spline is by construction identical to the original set at any sample point, the Mellin spline exhibits a large difference. Even more astonishing is the fact, that the difference is described by a constant factor of the correct result for every set[2] and also for the different types of the spline (natural and fixed). We did not examine the reason of this behavior, as it is not important for every possible use of the splines we can think of. It is extremely unlikely to use exactly the lowest sample point. Even if, it indicates that one should use another PDF set anyway, as the current does obviously not cover the needed range.

Table 5.1 demonstrates that the decision to put x into the s_+ sum gives at least the correct result for s_- also at $x = x_1$. Anyway this is already justified by equation (5.27). If the sum would include $x_k = x$ we would end up in the interval defined by $\tilde{k} = \min\{k|x_k \geq x\}$, which would be the wrong interval for $x_k = x$.

Furthermore we want to emphasize that table 5.1 and equation 5.27 demonstrate that (also numerically) only the s_- sum contributes, reducing the computational load and more importantly using only the parts of the splines that describe the PDFs best. This is of especial interest for sets, where small x are not described too well by the default grid, see for example figure 5.3. If needed, it is always possible to improve the description of the PDF by the spline by using more grid points. In this manner one trades accuracy against computational load (in the Mellin space—the x space splines will be mainly unaffected in this point). We emphasize again that by introducing another than the native grid of the PDF will mingle the interpolation of the spline with the interpolation of the PDF set. However, as the sets should provide a reliable interpolation compared to their analytical form, this seems acceptable. Especially because in use cases at least the scale μ^2 will most likely not meet the grid of the set and the interpolation of it will be applied in that direction anyway. However, the interpolation in scale will be different to the interpolation in longitudinal momentum fraction. The former is well described by a logarithmic dependence, why the latter might be more complicated.

5.4 Calculating a cross section with Mellin splines

To demonstrate the use of splines in Mellin space we will show the accompanying calculation for a specific cross section and compare the numerical result with a result

[2] In some configurations one does not see this effect. The reason is a numerical inaccuracy that causes $x \neq x_1$. As pointed out, the slightest shift from the lowest sample point suffices to obtain the expected result.

that uses an additional fit similar to equation (5.4). For this purpose we choose the resummed unpolarized Drell-Yan cross section, which has been calculated in [265]. We have been able to receive the numerical code, which used the GRV parton distribution functions [255] for the numerical calculations in [265]. Because the current state of our numerical spline implementations considers only PDF sets of this decade (see section 5.5), we exchanged the set by an already existing fit to the CT10 set [87] which is valid for scales μ^2 in 10 to 640 GeV^2 and longitudinal momentum fractions x in 0.01 to 0.9 for valence and 0.01 to 0.6 for sea quarks. This assures a sufficient proximity to the set accessible in our Fortran code implementing the spline interpolation, which is the CT12 set [82].[3]

For the analytical discussion the exact formular of the cross section is not necessary. It will therefore not be repeated here, as it is described in detail in [265]. The only important point for our discussion is, that the cross section (and also differential cross sections) take the form

$$\sigma(\tau) = \int_{\mathcal{C}} \frac{\mathrm{d}N}{2\pi\mathrm{i}} \tau^{-N} h(N) f(N) \tilde{f}(N) \tag{5.28}$$

where $\tau = Q^2/s \in \big[0,1\big]$, $f(N)$ and $\tilde{f}(N)$ are the parton distribution functions of the incoming protons and $h(N)$ contains all the details we do not need for the discussion. We also omitted the flavour indices of the quarks and the corresponding sum, that composes the possible initial states with appropriate terms in $h(N)$ (which would also be decorated by the indices), as they also play no role in the following argument. The only point that matters, is that the PDFs are not necessarily the same, which is indicated by the tilde. Now using splines we can set $f(N) \to S(N)$ and $\tilde{f}(N) \to \tilde{S}(N)$, getting the expression

$$\sigma(\tau) = \int_{\mathcal{C}} \frac{\mathrm{d}N}{2\pi\mathrm{i}} \tau^{-N} h(N) \sum_{i,j=1}^{n} \sum_{k,m=0}^{3} g_{k,i} \tilde{g}_{j,m} \frac{x_i^N x_j^N}{(N+k)(N+m)} \tag{5.29}$$

where n is the number of sample points, which we assume to be equal for $S(N)$ and $\tilde{S}(N)$ for simplicity. The general case with different underlying grids does not add any difficulties, only larger expressions. Now the same argument as in section 5.3 applies: the contour $\mathcal{C}$ has to be chosen according to the asymptotic behavior of the integrand. Assuming that $h(N)$ does not change this behavior, the criterion for one single summand is given by

$$\tau^{-N} x_i^N x_j^N = \exp\left[N \ln\left(\frac{x_i x_j}{\tau} \right) \right] \tag{5.30}$$

[3] Confusingly the title of [82] states to correspond to CT10. But also the standalone codes of CT12 states to correspond to [82].

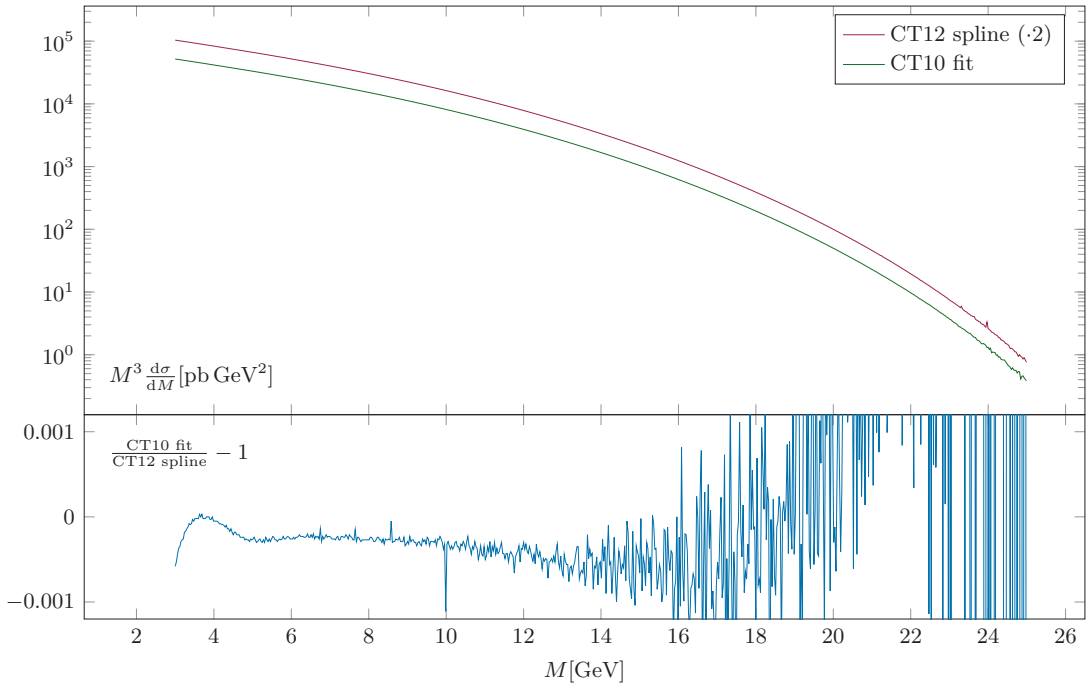


Figure 5.6: Unpolarized Drell-Yan cross section for a $p\bar{p}$ collider with $\sqrt{s} = 30\,\text{GeV}$. The numerical code has been taken from [265]. The upper panel in this plot represents their right panel of figure 4, except for the used PDF set. While in [265] the GRV set [255] is used, we use the CT10 set [87] for the conventional fit and CT12 [82] for the spline interpolation. To distinguish the two plots, we multiplied the spline based cross section with two. The lower panel depicts the quotient of the two approaches.

which implies, that for $\tau < x_i x_j$ the contour should approach $\text{Re}(N) \to -\infty$ while for $\tau > x_i x_j$ it should approach $\text{Re}(N) \to \infty$. We call these again $\mathcal{C}_-$ and $\mathcal{C}_+$. The sums split up accordingly

$$\begin{aligned}\sigma(\tau) = &\sum_{\substack{i,j \\ x_i x_j > \tau}} \sum_{k,m=0}^{3} g_{k,i} \tilde{g}_{m,j} \int_{\mathcal{C}_-} \frac{\mathrm{d}N}{2\pi \mathrm{i}} h(N) \frac{\exp\left[N \ln\left(\frac{x_i x_j}{\tau}\right)\right]}{(N+k)(N+m)} \\ &+ \sum_{\substack{i,j \\ x_i x_j \leq \tau}} \sum_{k,m=0}^{3} g_{k,i} \tilde{g}_{m,j} \int_{\mathcal{C}_+} \frac{\mathrm{d}N}{2\pi \mathrm{i}} h(N) \frac{\exp\left[-N \ln\left(\frac{\tau}{x_i x_j}\right)\right]}{(N+k)(N+m)} \\ \equiv &\, \sigma_-(\tau) + \sigma_+(\tau) \end{aligned} \tag{5.31}$$

In principle the term σ_+ should vanish, as the contour of the inverse Mellin transform has to be on the right-hand side of all poles of the integrand. This is not possible for the resummed cross section we want to investigate, because the Landau pole induces a branch cut along the real axis. However, it has been shown that it is valid to integrate

on the left-hand side of the Landau pole. The procedure is known as the minimal prescription [126, 139, 266]. Nevertheless we will ignore σ_+, as it is by orders of magnitude smaller than σ_- for small boson masses M and consistent with zero for intermediate and large M. In figure 5.6 we show the resummed Drell-Yan cross section for a $p\bar{p}$ collider running at the center of mass energy $\sqrt{s} = 30\,\text{GeV}$ [265, FIG.4, right panel]. As mentioned before we changed the originally used PDF set [255] to a fit to the CT10 PDFs [87]. For the spline we used CT12 [82], which was the latest set available at the time the Fortran spline code was written. We can confirm that the calculation with the spline agrees with the conventional approach. For $M < 16\,\text{GeV}$ they agree on sub per mill level. The fluctuations in the lower panel are due to the error of the integration routine. Interestingly, the stated error estimate is lower for the splines as for the fit, which is most likely a numerical artifact. For $M > 16\,\text{GeV}$ the fluctuations increase, as the integration becomes more unstable (also visible in the upper panel). Nevertheless we can detect a slight trend of the fitted PDFs to produce a larger cross section than the splines do. Because it is not possible to decide which one gives the 'correct' result, we want to emphasize that the fit has a limited validity range for $x \to 1$, which becomes more stressed at the kinematic threshold $\tau \to 1$ (which is $M \to \sqrt{s}$).

An additional feature of the spline interpolation is, that it might be possible to extract more informations about the cross section at threshold. As remnant of the minimal prescription it is known that the differential cross section is not zero at the kinematic threshold $\tau = 1$. It is possible that the spline will not show this remnant, as long as one only considers the part integrated along $\mathcal{C}_-$. This would also be in agreement with the observation that the spline produces smaller values for large M (however, this is speculative because we use a slightly different set and we already know that the PDF fit might not describe the PDFs accurate enough in this region). Even more interesting it might be possible, that the $\mathcal{C}_+$ part reveals the error made by the estimate of the minimal prescription. However, for this purpose it is necessary to perform a very sophisticated integration, as the standard Monte Carlo integration does not sufficiently converge to get any reliable numbers. One main problem is the computational load that is accompanied by using the spline interpolation. We started to address this problem in chapter 2.

Finally we will briefly study the case that the PDFs are the main source of poles of the integrand in Mellin space. The common fit, resulting in beta functions, see equation (5.5), includes an infinite number of poles close to the real axis with $\text{Re}(N) < 0$. As mentioned before cubic splines exhibit only four poles. If $h(N)$ itself has no poles at all, we are able to calculate the cross section analytically. We split the result for the Drell-Yan example into the diagonal and non-diagonal parts

$$\sigma_- = \sum_{\substack{i,j \\ x_i x_j > \tau}} \sum_{m=0}^{3} g_{m,i} \tilde{g}_{m,j} \left[\partial_N h(N) \exp\left[N \ln\left(\frac{x_i x_j}{\tau} \right) \right] \right]_{N=-m}$$

$$
\begin{aligned}
&+ \sum_{\substack{i,j \\ x_i x_j > \tau}} \sum_{\substack{k,m=0 \\ k \neq m}}^{3} g_{k,i} \tilde{g}_{m,j} \left[\frac{h(-k)\left(\frac{x_i x_j}{\tau}\right)^{-k}}{m-k} + \frac{h(-m)\left(\frac{x_i x_j}{\tau}\right)^{-m}}{k-m} \right] \\
&= \sum_{\substack{i,j \\ x_i x_j > \tau}} \sum_{m=0}^{3} \left(\frac{\tau}{x_i x_j}\right)^{m} g_{m,i} \tilde{g}_{m,j} \left[h(-j) \ln\left(\frac{x_i x_j}{\tau}\right) + \left[\partial_N h(N)\right]_{N=-j} \right] \\
&+ \sum_{\substack{i,j \\ x_i x_j > \tau}} \sum_{\substack{k,m=0 \\ k \neq m}}^{3} \frac{g_{k,i} \tilde{g}_{m,j}}{m-k} \left[h(-k) \left(\frac{\tau}{x_i x_j}\right)^{k} - h(-m) \left(\frac{\tau}{x_i x_j}\right)^{m} \right]
\end{aligned}
\tag{5.32}
$$

If $h(N)$ contains a finite number of poles, the computation can still be performed analytically with additional terms from the Residuum theorem.

5.5 Numerical implementation of PDF Mellin splines

Because the calculation of a spline is a task that is equal for every possible PDF set, it is worth writing a program that performs this task and offers an easy to use interface for subsequent use. In this section we will kindly introduce a Fortran as well as a C++ implementation.

5.5.1 Fortran implementation

The basic functionality of the Fortran implementation is hidden inside the `module PdfSplineModule`, which should by design not be of interest for the user. However, it also contains some important 'static' variables

pdfSpline.f (extract)

```
      module PdfSplineModule
! Module that contains the spline function not to use by the user
! directly and global variables
!----------------------------------------------------------------------
      implicit none
      save
! maximum number of grid points
      integer, parameter :: nmax = 1000

! default values if not specified by user
      integer :: st_int = 0 ! natural spline
      integer :: iset_int = 1 ! CT12
      integer :: mmhtih_int = 0 ! MMHT central pdfs
      integer :: dssih_int = 1 ! DSS hadron type: pion
      integer :: dssic_int = 0 ! DSS hadron charge: average
      integer :: dssio_int = 1 ! DSS order: NLO
      double precision :: q2_int = 1000.d0
      character(len=100) :: file_int = 'ct10n.00' ! CT12 central NLO
```

```
! Further variables
      integer :: ngrid = -1 ! number of grid points
```

The perhaps most important point in using the Fortran routines is the limitation to 1000 sample points, which we have chosen to be fixed, as it is tedious to allocate memory dynamically. The chosen maximal number should be sufficient for most grids and not too large to stress too much memory. However, if one desires another maximum, it is sufficient to change the parameter `nmax`. All other values shown are the defaults, that will be explained in the following.

All functionality that a typical user needs is provided in

pdfSplineInterface.f

```
! include to use the functionality of pdfSpline.f
      interface
         subroutine PdfSplineSet(st,iset,mmhtih,dssih,dssic,dssio,q2,
     &                           file)
            integer,intent(in),optional :: st,iset,mmhtih,dssih,dssic,
     &                                     dssio
            double precision,intent(in),optional :: q2
            character(len=*),intent(in),optional :: file
         end subroutine PdfSplineSet
      end interface
      interface PdfSplineGrid
         subroutine PdfSplineGrid_twoarray(xarray,narray)
            integer,intent(in) :: narray(:)
            double precision, intent(in) :: xarray(:)
         end subroutine PdfSplineGrid_twoarray
         subroutine PdfSplineGrid_simple(xmin,n)
            integer,intent(in) :: n
            double precision, intent(in) :: xmin
         end subroutine PdfSplineGrid_simple
         subroutine PdfSplineGrid_onearray(xarray)
            double precision, intent(in) :: xarray(:)
         end subroutine PdfSplineGrid_onearray
         subroutine PdfSplineGrid_onearrayreal(xarray)
            real, intent(in) :: xarray(:)
         end subroutine PdfSplineGrid_onearrayreal
      end interface
      interface
         subroutine PdfSplineGet(iparton,nout,xout,paraout)
            integer,intent(in) :: iparton
            integer,intent(out) :: nout
            double precision,intent(out) :: xout(:), paraout(:,:)
         end subroutine PdfSplineGet
      end interface
      interface
         double complex function PdfSplineM(iparton,N)
            integer,intent(in) :: iparton
            double complex,intent(in) :: N
         end function PdfSplineM
      end interface
      interface
```

```
      double precision function PdfSplineX(iparton,x)
         integer,intent(in) :: iparton
         double precision, intent(in) :: x
      end function PdfSplineX
   end interface
   interface
      subroutine PdfSplinePara(io)
         integer, intent(in),optional :: io
      end subroutine PdfSplinePara
   end interface
   interface
      double precision function PdfSplineAlphas(q2,iord,fr2,mur,
&                                               asmur,mc,mb,mt)
         double precision, intent(in) :: q2
         integer, intent(in), optional :: iord
         double precision, intent(in), optional :: fr2,mur,asmur,
&                                                  mc,mb,mt
      end function PdfSplineAlphas
   end interface
```

which has to be included into a customer program to get access to the spline functionality. To initialize the calculation of the parameters of a spline is is mandatory to use the `subroutine PdfSplineSet(st, iset, mmhtih, dssih, dssix, dssio, q2, datafile)`. The first parameter `st` determines the spline type. `st = 0` is the natural spline, `st = 1` a fixed spline which uses the underlying PDF interpolation routine to determine the first derivatives (see equations (5.13)) and `st = 2` is a fixed spline which uses the grid itself to calculate the derivatives (see equations (5.14)). The second parameter `iset` determines the underlying PDF set. Currently three sets are implemented: CT12 [82] (`iset = 1`), MMHT14 [90] (`iset = 2`) and DSS07 [258] (`iset = 3`), which where the up to date sets of that collaborations at the production time of the spline code. All other parameters are passed to the actual PDF routines, the prefixes indicate to which set they apply. If a parameter does not apply to a set, its value is indifferent. The default values where already given in the first listing of this section. The `PdfSplineGrid` contains several ways to call a subroutine to define the sampling points for the spline. The simplest one is to pass an array, that contains all x_i, where the last point has to be equal to one (`subroutine PdfSplineGrid(xarray)`). As an alternative one can pass the number of grid points n and the smallest point of the grid $x_{\min}$ (`subroutine PdfSplineGrid(xmin, n)`). The sampling points will then be distributed logarithmically in the interval $\left[x_{\min}, 1\right]$. Finally one can use a mixture, passing two arrays. One defines points in x space, the other defines the number of points sampling the PDF in the intervals defined by the first array (`subroutine PdfSplineGrid(xarray, narray)`). These will again be distributed logarithmically in the corresponding intervals. If no grid is specified before the first call of `PdfSplineSet`, the program will use the grid points defined by the PDF collaborations. The remaining functions can be called after a spline has been calculated. The `subroutine PdfSplineGet` returns the number of sampling points `nout`, the sampling points `xout` and the spline coefficients `paraout`, which are defined

by $g_{m,k}/x_k^m$ (see equation (5.20)). The latter is two dimensional, the first index mapping to m and the second to k. The `function PdfSplineX(iparton, x)` returns the value of the spline of a parton at the given longitudinal momentum fraction. The variable `iparton` maps to the flavours according to the function defined in equation (1.71), but omitting the top quark. The `function PdfSplineM(iparton, N)` is the equivalent in Mellin space. The `subroutine PdfSplinePara(io)` is a diagnostic function to print the parameters that determine the spline and also the spline coefficients into the i/o unit `io`. The argument defaults to `io=-5`, which is the terminal. Finally the interface offers the `function PdfSplineAlphas(q2, iord, fr2, mur, asmur, mc, mb, mt)` which selects the appropriate α_s routine of the selected PDF set in `PdfSplineSet`. All arguments are passed to these functions and are indifferent if the underlying set does not use them. They are defaulted to `iord=2` (which corresponds to NNLO), `fr2=1.d0` (the ratio of the factorization and the renormalization scales squared), `mur=1.d0` (the renormalization scale), `asmur=0.5d0` (the value of α_s at the renormalization scale), `mc=1.4d0` (charm mass), `mb=4.75d0` (bottom mass) and `mt=1.d10` (top mass).

To extend the program to support additional PDF sets one has to add a new case into the `subroutine PdfSplineCalc` (which is a member of the `module PdfSplineModule`) to calculate the PDF values at the sampling points. Further one has to extend the `subroutine PdfSplineSet` with the standard sampling points and possibly new parameters. The same is true for the `subroutine PdfSplinePara` and `function PdfSplineAlphas`.

5.5.2 C++ implementation

Because C++ allows the more sophisticated concept of object orientated programming it is easy to implement the splines in a much more generic way as it has been done for Fortran. We wrote two standalone classes, `template<typename T> Spline` and `template<typename T> MellinSpline` that cover the basic functionalities of a cubic spline and of the spline in Mellin space. The template parameter determines the floating point type (and therefore the numerical precision). As we do throughout this thesis, we omit the namespaces that save the global namespace from pollution in all listings. The details of this classes do not matter in connection with the PDF splines, as we fused their functionality into a base class, that can be used to implement various PDF sets.

PdfSplineBase.h (extract)

```
template<class Function, class ... Types> class PdfSplineBase
{
public:
   using SplineType = interpolation::SplineType;
   using value_type = typename Function::value_type;

   static constexpr unsigned int coefficientsPerPoint =
   ↪  interpolation::MellinSpline<value_type>::coefficientsPerPoint;
```

```
    virtual ~PdfSplineBase() = default;

    void set( const SplineType type );

    template<class PointIt> void set( PointIt firstPoint, PointIt lastPoint );

    value_type getTruePdf( const value_type& xValue ) const;

    std::size_t numberPoints() const;

    value_type getPointAt( const std::size_t index ) const;

    value_type getMinimalPoint() const;

    value_type getMaximalPoint() const;

    bool isInRange( const value_type& point ) const;

    SplineType getType() const;

    interpolation::SplinePiece<value_type> getPiece( const std::size_t index ) const;

    std::vector<value_type> getCoefficientsAt( const std::size_t index ) const;

    value_type operator()( const value_type& point ) const;

    template<template<typename> class Complex> Complex<value_type> operator()( const
    ↪  Complex<value_type>& mellinVariable ) const;

protected:
    Function function_;

    template<class PointIt> PdfSplineBase( PointIt firstPoint, PointIt lastPoint, const
    ↪  SplineType type, Types ... args )
       : function_( args... )
       // further initialization

    void update();
};
```

The template requires a functor and all types that have to be passed to the constructor of this functor. Furthermore the functor has to provide a typedef `value_type` that will determine the floating point type of the underlying `Spline` and `MellinSpline`. After construction it is possible to alter the spline with the two `set` functions. The iterator `PointIt` has to be at least an input iterator. The function `getTruePdf` provides access to the underlying functor. The member functions containing the word `Point` are self-explanatory as is `isInRange`. The function `getType` return the type of the spline. In the current state the C++ implementation contains only the two modes `SplineType::NATURAL` and `SplineType::FIXED`. The latter one corresponds to equations (5.14). The alternative of using equations (5.13) could be implemented easily, if needed. The function `getPiece` returns a struct called `SplinePiece`, which is a simple functor containing the single cubic polynomial that corresponds to the interval specified by the index. The perhaps most important function to use the

splines in an integration code is `getCoefficientsAt` which returns the coefficients $g_{m,k}$ (see equation (5.20)), where the `index` maps to k and the returned `std::vector` contains the elements specified by m. Finally the parenthesis operators calculate the spline in longitudinal momentum space and in Mellin space. To child classes the underlying functor is exposed, forwarding the responsibility to implement setter functions specialized on the specific spline (as is for example the current scale, or the considered parton). All these setter have to call `update` to update the spline.

Currently the following sets are implemented

- The sets CT10 [87], CT10N [87], CT10NN [82] and CT14 [88] of the CTEQ group
- The MMHT set [90]
- The DSS fragmentation sets DSS07 [258] and DSS14 [259]

We use the standalone codes of the different groups to calculate the PDFs. Because all of them provide Fortran interfaces exclusively[4], we have to access these, which makes it a bit more complicated as strictly necessary. Hence we will shortly discuss the CTEQ implementation as an example.

To discriminate different sets of the same group we use a simple enum

ct_settings.h

```
enum class CtSet {CT10, CT10N, CT10NN, CT14};
```

which is used as template parameter for the class that inherits from the generic base class `PdfSplineBase`. The functor itself just calls the Fortran routine to retrieve the original values. In case of the CTEQ routine we also have to initialize the routine (which is named 'set' by the CTEQ collaborators, which we adapt) and finally have to provide the `value_type`.

CtFunctor.h

```
template<CtSet ctSet> class CtFunctor
{
public:
   using value_type = double;

   CtFunctor( const int inputFlavour, const double inputScale, const std::string&
   ↪  inputSetName )
      : scale_( inputScale )
      , flavour_( inputFlavour )
```

[4] MMHT also provides a C++ interface. However, the provided code does not fulfill C++ standards. For example, it pollutes the global namespace in the header file with the whole `std` namespace and ignores the one definition rule for their class invariants. Although this particular issues could be fixed easily, we dismiss this interface fearing more subtle bugs.

```
   {
      CtFortranCodeSelector<ctSet>::set( inputSetName );
   }

   double operator()( const double x ) const
   {
      return CtFortranCodeSelector<ctSet>::call( flavour_, x, scale_ );
   }

   double scale_;
   int flavour_;
};
```

The functor contains members for all variables that are needed by the CTEQ routine to calculate the PDF function. The class `CtFortranCodeSelector` ensures that the appropriate Fortran routine is called, which are hidden in an additional namespace

ct_fortran.h

```
namespace fortran {
extern "C"
{
   void setct12_( const char tablefile[40] );
   double ct12pdf_( const int* parton, const double* x, const double* q );

   void setct14_( const char tablefile[40] );
   double ct14pdf_( const int* parton, const double* x, const double* q );
}
} // namespace fortran
```

It is evident that one has to link against the precompiled Fortran routines that will be needed.

Connecting the loose ends the child class of `PdfSplineBase` treats all specific setter functions for the CTEQ routines.

CtSpline.h (extract)

```
template<CtSet ctSet, typename T = double> class CtSpline : public
↪  PdfSplineBase<CtFunctor<ctSet>, int, double, std::string>
{
private:
   using PdfSpline = PdfSplineBase<CtFunctor<ctSet>, int, double, std::string>;

public:
   using SplineType = typename PdfSpline::SplineType;

   template<class PointIt> CtSpline( PointIt firstPoint, PointIt lastPoint, const
   ↪  SplineType type, const std::string& setName, const PartonFlavour flavour, const T
   ↪  scale );
```

```
    template<class PointIt> CtSpline( PointIt firstPoint, PointIt lastPoint, const
    ↪  std::string& setName, const PartonFlavour flavour, const T scale );

    CtSpline( const SplineType type, const std::string& setName, const PartonFlavour
    ↪  flavour, const T scale )
       : CtSpline( CtDefaultGrid<T, ctSet>::POINTS.begin(), CtDefaultGrid<T,
       ↪  ctSet>::POINTS.end(), type, setName, flavour, scale )
    {}

    CtSpline( const std::string& setName, const PartonFlavour flavour, const T scale );

    void setDatafile( const std::string& inputSetName );

    void setDefaultPoints();

    void setScale( const T scale );

    void setFlavour( const PartonFlavour inputFlavour );
};
```

These functions are obviously `setDatafile` (CTEQ uses a char array to determine various subsets, including error sets), `setDefaultPoints` (settings the sample points equal to the grid describing the CTEQ fit), `setScale` and `setFlavour`. All their implementations have to call a `set` function of `PdfSplineBase` or change the function object `PdfSpline::function_` in the base class and subsequently invoke the `PdfSpline::update` function to recalculate the spline. Finally we want to point out three details. First the default grid is stored in a static class `CtDefaultGrid<T, CtSet>`, which is used in the constructors that do not specify the sampling points via the iterators `PointIt`. Second the spline defaults to be a natural spline, as all spline classes we wrote do.[5] Third we use also here a template to choose the floating point type. This would not be necessary, because the underlying routines only support double precision, which is also the default. Nevertheless it is useful to use the template, as it is easier to integrate into costumer code that can discriminate between different floating point types.

As demonstrated in the example the extendability of the C++ implementation of PDF splines is straightforward and should in principle be able to adapt to every imaginable situation. Furthermore the object orientated design allows punctual improvements to be applied easily and secure.

5.6 Conclusions

In this chapter we presented an entirely new method to describe parton distributions functions in Mellin space, by interpolating them with cubic splines and transforming

[5] The alternative is the fixed spline using the underlying grid (see equations (5.14)). The version corresponding to equations (5.13) has not been implemented. To catch it up a simple change in the `Spline` class is sufficient.

the splines, which yields a very easy functional form. We have demonstrated that the interpolation is extremely precise and pointed out the procedure is completely automatable in contrast to the usual technique. In fact we found for at least one PDF set that the spline, inheriting the native grid, is even superior to the interpolation routine distributed with the PDF set itself.

We mentioned that the computational load is likely to increase when using splines for numerical integrations in Mellin space. The load increases with the number of sample points used to determine the functional form of the spline. As a consequence the only non automatic task in using splines is to adjust the number of sample points to balance between the needs of precision and fast computation. We shortly discussed the possibility of extending the interpolation to multiple dimensions, but without performing subsequent calculations.

To test the spline interpolation in a realistic scenario, we chose a resummed Drell-Yan cross section differential in the mass of the intermediate vector boson. We found a perfect description of the cross section. Albeit paid with additional computational load, the approach with the splines can in principle again be automated for all PDF sets, which can such be compared in a much more satisfying way, as the PDF representatives in Mellin space do not depend on the partly difficult fit procedure that has to be applied with the usual method. We discussed the possibility to be able to further analyze resummed cross sections, as the splines induce a separation of the analytical expression to ensure the convergence of the inverse Mellin transform. In our example this might lead, with sufficient computational power, to an approximation of the remnant introduced by the minimal prescription, that sets the Landau pole to the right of the contour integral of the inverse transform. Subsequently we discussed the case, where the PDFs are the major sources of poles. Expressed by splines only a finite number of poles occur, enabling an analytical calculation of the inverse Mellin transform.

Finally we presented numerical implementations for Fortran and C++, that implement the calculation of the splines for several PDF sets. In both programming languages we provide an easy to use interface. We discussed the ability of extensions to additional PDF sets and explained the necessary steps to perform this task.

As we pointed out in the introduction, a crucial ingredient to every calculation in perturbative QCD are the PDFs. As the presence of the LHAPDF library [157] indicates, it is important for many researchers to exchange the used PDF set to proof results on their reliability and/or to estimate theoretical uncertainties by the PDFs not provided in their error sets, which is ultimately an estimate on our ignorance of the nonperturbative part of QCD. The results and programs introduced here extend this feature to all computations performed in Mellin space. They open the window to a fully automated use and exchange of PDFs in computations of physical cross sections. Actually it would be a sensible matter to interface the LHAPDF library itself and, on the long run, to integrate it into the library.

The higher computational load associated with the use of splines will most likely become more unimportant in future thanks to the ongoing development of compu-

tational resources. Particularly the use of graphics cards (see chapter 2) will lower the overall computation time. This will not change the situation in principle, as the conventional approach will most likely need always less computational time, but the absolute difference will become tremendously smaller, making the splines to a contender, especially because it is always possible to trade speed for precision and vise versa in a precise and reproducible manner.

Chapter 6

Sudakov form factor

Quot capita tot sensus.

— Indian doctor
(Asterix and the Magic Carpet)

We investigate the numerical dependence of the Sudakov form factor on the analytical choice of the strong coupling constant by the example of small transverse momentum resummation. The choice is necessary to solve the integrals inside the form factors. We discuss also the closely related TMD evolution in respect to the same choice. We are able to demonstrate a sizable dependence on this choice in both formalisms. The analysis suggests that also all consecutive findings depend on it. To obtain reproducibility we recommend that this subtle detail should always be mentioned, although the expressions are equivalent in terms of the perturbative order under consideration.

6.1 Introduction

At the edges of phase space it is a common feature of perturbative QCD calculations to encounter large logarithms. Well known examples are the systems of high mass or high momentum parts (typically the transverse momentum) close to the kinematic threshold or diametrically systems with a very small momentum part. In all these regions only additional soft gluons can be produced in higher orders, which spoil the cancellation of virtual and real singularities, resulting in large logarithmic contributions. This particular configurations are called Sudakov regimes. As a consequence of the large logarithms the perturbative series collapses, as contributions in higher orders are systematically larger than those in lower orders. A lot of work has been performed to access control over the perturbative regimes [126, 219–222, 225, 267–270]. The general solution is to resum the perturbative series to all orders, which results in principle in a reordering of the series such, that again the largest contributions are taken into account to a well defined order. As one speaks of leading order (LO) and next-to-leading order (NLO) and so on in perturbative QCD, the resummation accuracy is usually called to be leading log (LL), next-to-leading log (NLL) and so forth. In general the result of the resummation is an additional multiplicative factor, which is the exponentiated Sudakov form factor.

The details of the resummation vary tremendously between the different possible thresholds and processes. We will pick the resummation of small transverse momenta, as in this field an entire new formalism, the TMD formalism [62, 69, 221, 222, 225, 271], has been invented to describe associated experimental data. The abbreviation TMD stands for transverse momentum dependent, because the parton distribution functions (PDFs) and/or fragmentation functions (FFs) become dependent on the transverse momentum. It is not surprising that this formalism gives analytically the same results as the plain resummation formalism, but interestingly it leads to a different kind of thinking about which terms should be taken into account for a consistent calculation in a specified order.

In the resummation community the Sudakov form factor is commonly written as

$$S(b_T, Q^2) = -\int\limits_{b_0^2/b_T^2}^{Q^2} \frac{\mathrm{d}k_T^2}{k_T^2}\left[A(\alpha_s(k_T^2))\ln\left(\frac{Q^2}{k_T^2}\right) + B(\alpha_s(k_T^2))\right] \tag{6.1}$$

where $b_0 = 2\,\mathrm{e}^{-\gamma_E}$ and the functions $A(\alpha_s)$ and $B(\alpha_s)$ are series in the strong coupling defined by

$$A(\alpha_s) = \sum_{n=1}^{\infty} A_n\left(\frac{\alpha_s}{\pi}\right)^n \tag{6.2a}$$

$$B(\alpha_s) = \sum_{n=1}^{\infty} B_n\left(\frac{\alpha_s}{\pi}\right)^n \tag{6.2b}$$

The coefficients A_n and B_n can be calculated by solving the corresponding renormalization group equation or by comparing the expanded resumed result to fixed-order calculations. In case of small transverse momenta the first coefficients are given by [50, 272–275]

$$A_1 = C_F \tag{6.3a}$$

$$A_2 = \frac{C_F}{2}\left[C_A\left(\frac{67}{18} - \frac{\pi^2}{6}\right) - \frac{10}{9}T_R n_f\right] \tag{6.3b}$$

$$B_1 = -\frac{3}{2}C_F \tag{6.3c}$$

$$B_2 = C_F^2\left(-\frac{3}{16} + \frac{\pi^2}{4} - 3\zeta(3)\right) + C_F C_A\left(-\frac{3155}{432} + \frac{11\pi^2}{36} + 5\zeta(3)\right) + C_F n_f\left(\frac{247}{216} - \frac{\pi^2}{18}\right) \tag{6.3d}$$

In terms of the TMD formalism $A(\alpha_s)$ and $B(\alpha_s)$ are expressed as anomalous dimensions of the TMD PDFs $\gamma_F(\alpha_s)$ and the Collins-Soper evolution kernel $\gamma_K(\alpha_s)$. The

resummation exponent is translated into the evolution of the TMDs [239]

$$\int_{\mu_b}^{Q} \frac{\mathrm{d}\mu}{\mu}\left[\gamma_F(\mu,1) - \gamma_K(\mu)\ln\left(\frac{Q}{\mu}\right)\right] = \frac{1}{2}S(b_T,Q^2) \tag{6.4}$$

with the anomalous dimensions in leading order

$$\gamma_K(\mu) = \alpha_s(\mu^2)\frac{2C_F}{\pi} \tag{6.5}$$

$$\gamma_F\left(\mu,\frac{Q^2}{\mu^2}\right) = \alpha_s(\mu^2)\frac{C_F}{\pi}\left[\frac{3}{2} - \ln\left(\frac{Q^2}{\mu^2}\right)\right] \tag{6.6}$$

and the scale

$$\mu_b = \frac{b_0}{b_T} \tag{6.7}$$

The factor one-half originates from the fact that in the considered cases the Sudakov factor accompanies two PDFs (or FFs) while these have separated evolution factors in the TMD scheme, each responsible for half of the Sudakov form factor.

The evolution between two scales Q_0^2 and Q^2 can be achieved by dividing two TMDs written with an input at scale μ_b and evolved by (6.4) as defined in [276]

$$R(b_T,Q^2,Q_0^2) = \exp\left[\ln\left(\frac{Q}{Q_0}\right)\int_{Q_0}^{\mu_b}\frac{\mathrm{d}\mu}{\mu}\gamma_K(\mu) + \int_{Q_0}^{Q}\frac{\mathrm{d}\mu}{\mu}\gamma_F\left(\mu,\frac{Q^2}{\mu^2}\right)\right] \tag{6.8}$$

This form is consequently related to the Sudakov factor of the resummation formalism by

$$\ln(R(b_T,Q^2,Q_0^2)) = \frac{1}{2}[S(b_T,Q^2) - S(b_T,Q_0^2)] \tag{6.9}$$

Regardless of the underlying formalism it is necessary to introduce a description to avoid the Landau pole of the running coupling α_s. The most popular is the b_* prescription [222, 223, 225], which is basically a smoothened cutoff. This can be achieved in various ways, again the perhaps most popular among these is

$$b_*(b_T) = \frac{b_T}{\sqrt{1+\frac{b_T^2}{b_{\max}^2}}} \tag{6.10}$$

where the parameter $b_{\max}$ is responsible for the separation of the perturbative and the nonperturbative regime. It applies to the scale μ_b and the lower limit of the Sudakov integral (6.1), which is exactly μ_b^2. The numerical results depend strongly on $b_{\max}$, common choices are $0.5\,\mathrm{GeV}^{-1}$ [204] or $1.5\,\mathrm{GeV}^{-1}$ [205], which has a large impact on the deduced nonperturbative parameters [205].

In the following section we will investigate the numerical dependencies of the Sudakov form factor on the choice of the analytical form of α_s. We will discuss in detail the different phenomenological behavior induced by the different routines for two sensible choices of the coefficients in equation (6.2) taken into account. These reflect the previously mentioned different preferences of the resummation and the TMD community. Further we will examine the effect of using the difference of two Sudakov form factors as implied by equation (6.8).

6.2 Dependence on the running coupling

In calculations that incorporate small transverse momentum resummation (or evolution of TMDs) a numerical integration is carried out for the inverse Fourier transform as the resummation has to be carried out in impact parameter space. Because the integral of the Sudakov form factor is exponentiated, it is not feasible to perform also this integration numerically, in which case the running coupling constant could in principle be implemented by a Runge-Kutta method. But the computational load for nested integrations is way too large. It is therefore necessary to perform the integration in the Sudakov factor analytically. For this calculations multiple ways to parameterize the running coupling are possible that are equivalent in terms of the perturbative order they consider.

We will investigate two forms for the running coupling. The first is the analytical solution of the renormalization group equation (see (1.24)) and relates the coupling at the desired scale μ^2 to the coupling at an arbitrary scale μ_0^2, as can be found for example in the appendix of [277] at NLO

$$\alpha_{s,\mathrm{NLO}}(\mu^2) = \alpha_{s,\mathrm{LO}}(\mu^2)\left[1 - \frac{b_1}{b_0}\alpha_{s,\mathrm{LO}}(\mu^2)\ln\left(1 + b_0\alpha_s(\mu_0^2)\ln\left(\frac{\mu^2}{\mu_0^2}\right)\right)\right] \tag{6.11}$$

with the LO expression

$$\alpha_{s,\mathrm{LO}}(\mu^2) = \frac{\alpha_s(\mu_0^2)}{1 + b_0\alpha_s(\mu_0^2)\ln\left(\frac{\mu^2}{\mu_0^2}\right)} \tag{6.12}$$

where the b_i are related to the coefficients of the β function and are defined in equation (1.30). An alternative representation is computed by determining the Landau pole Λ_{qcd} in (6.12) by the means of a very precise measured coupling at another scale, which is typically the Z boson mass m_Z. Generalizing the approach by replacing the Z boson mass with arbitrary scales, the running coupling can be expressed as series in $\ln^{-1}(\mu/\Lambda_{\mathrm{qcd}})$

$$\alpha_{s,\mathrm{NLO}}^{\Lambda_{\mathrm{qcd}}}(\mu^2) = \alpha_{s,\mathrm{LO}}^{\Lambda_{\mathrm{qcd}}}(\mu^2)\left[1 - \frac{b_1}{b_0}\alpha_{s,\mathrm{LO}}^{\Lambda_{\mathrm{qcd}}}(\mu^2)\ln\left(\ln\left(\frac{\mu^2}{\Lambda_{\mathrm{qcd}}^2}\right)\right)\right] \tag{6.13}$$

with

$$\alpha_{s,\mathrm{LO}}^{\Lambda_{\mathrm{qcd}}}(\mu^2) = \frac{1}{b_0 \ln\left(\frac{\mu^2}{\Lambda_{\mathrm{qcd}}^2}\right)} \tag{6.14}$$

This form has been used for example by the CTEQ collaboration to determine their parton distribution function sets up to version 6 [86]. In our numerical analysis we will do the same for the final computation of α_s, which has to be performed also when using equations (6.11) and (6.12). The values of Λ_{qcd} depend on the number of active quark flavours. In practice they are also used to smoothen the transition between two different numbers of active flavours. We will use the numerical values from the CTEQ6 routine, which base on $\alpha_s(m_Z) = 0.118$ [86]

$$\Lambda_{\mathrm{qcd}}^{n_f=4} = \begin{cases} 215\,\mathrm{MeV} & \text{for LO} \\ 326\,\mathrm{MeV} & \text{for NLO} \end{cases} \tag{6.15a}$$

$$\Lambda_{\mathrm{qcd}}^{n_f=5} = \begin{cases} 165\,\mathrm{MeV} & \text{for LO} \\ 226\,\mathrm{MeV} & \text{for NLO} \end{cases} \tag{6.15b}$$

where $\Lambda_{\mathrm{qcd}}^{n_f=4}$ is used for all scales smaller than the bottom quark mass $m_b = 4.5\,\mathrm{GeV}$, $\Lambda_{\mathrm{qcd}}^{n_f=5}$ for all scales larger than the bottom quark mass.[1]

Naturally also the analytical expressions used for the Sudakov form factor depend on the chosen form of the coupling. We discriminate the parts of the Sudakov exponent by the corresponding coefficient A_i or B_i and the perturbation order used for the coupling constant, id est LO uses (6.12) or (6.14) while NLO uses (6.11) or (6.13) to solve the integral (6.1). The analytical expressions derived by the α_s expressed by the series in $\ln^{-1}(\mu/\Lambda_{\mathrm{qcd}})$ have an additional superscript Λ_{qcd}. The analytical results are presented in appendix G.

In the sense of resummation one speaks of leading log accuracy (LL) with the dominant terms in the reordered perturbative series. These are given by

$$S_{\mathrm{LL}} = S_{A_1}^{\mathrm{LO}} \tag{6.16a}$$

$$S_{\mathrm{LL}}^{\Lambda_{\mathrm{qcd}}} = S_{A_1}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}} \tag{6.16b}$$

It is quite natural to use the LO α_s expressions for LL, because the next order would already include terms that are sub-leading. The next order (NLL) is constructed by collecting two additional towers of logarithms. The Sudakov form factors in NLL are given by

$$S_{\mathrm{NLL}} = S_{A_1}^{\mathrm{NLO}} + S_{B_1}^{\mathrm{LO}} + S_{A_2}^{\mathrm{LO}} \tag{6.17a}$$

$$S_{\mathrm{NLL}}^{\Lambda_{\mathrm{qcd}}} = S_{A_1}^{\mathrm{NLO},\Lambda_{\mathrm{qcd}}} + S_{B_1}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}} + S_{A_2}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}} \tag{6.17b}$$

[1] The best value for the bottom mass is currently given in $\overline{\mathrm{MS}}$ scheme by $m_b \approx 4.18\,\mathrm{GeV}$ [36]. We stay with the mass used by the CTEQ collaboration for consistency.

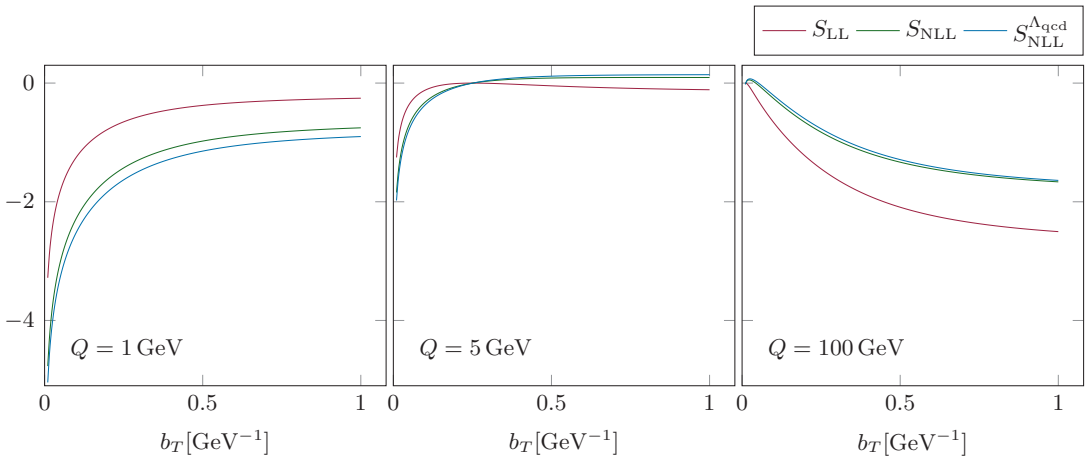


Figure 6.1: Comparison of the Sudakov form factors defined in equations (6.16) and (6.17) for three different energy scales, $Q = 1\,\mathrm{GeV}$, $Q = 5\,\mathrm{GeV}$ and $Q = 100\,\mathrm{GeV}$. $S_{\mathrm{LL}}^{\Lambda_{\mathrm{qcd}}}$ is equal to S_{LL} and therefore not shown additionally.

In this order the NLO contributions of A_1 are taken into account, while for B_1 and A_2 only the LO terms produce the logarithms to the desired accuracy. This has already been pointed out in [275].

In figure 6.1 we show the Sudakov form factors in LL and NLL for three different energy scales. The left panel shows the quite low scale $Q = 1\,\mathrm{GeV}$, the middle panel a scale which is a little bit larger than the bottom quark mass $Q = 5\,\mathrm{GeV}$ and the right panel a scale close to the mass of the Z boson $Q = 100\,\mathrm{GeV}$. In LL the results of the two underlying formulas are identical $S_{\mathrm{LL}} = S_{\mathrm{LL}}^{\Lambda_{\mathrm{qcd}}}$, as should be expected, and are plotted only once. For NLL the situation is different. Although the expressions are conformable they are not identical. This means the choice of the analytical form of α_s introduces an ambiguity for the Sudakov form factor not only for the analytical expressions but also for the numerical results. At $Q = 1\,\mathrm{GeV}$ the expression defined by S_{NLL} is larger than the corresponding $S_{\mathrm{NLL}}^{\Lambda_{\mathrm{qcd}}}$, while for the large scale $Q = 100\,\mathrm{GeV}$ the situation is vice versa. For $Q = 5\,\mathrm{GeV}$ the two expressions cross each other: for small impact parameter b_T S_{NLL} is larger, for large impact parameter smaller than $S_{\mathrm{NLL}}^{\Lambda_{\mathrm{qcd}}}$. A more precise comparison between the two expressions is shown in figure 6.2. It illustrates the difference of the two analytical expressions in the form it is present in a resummed cross section. The most impressive effect can be detected for $Q = 1\,\mathrm{GeV}$, where the relative difference is over 30 % for small impact parameter and still more than 15 % in the limit of large impact parameters. For larger scales the effects are somewhat smaller, but still not negligible. For $Q = 5\,\mathrm{GeV}$ we even detect the largest slope with an overall difference of almost 20 %. Finally, keeping in mind that conventionally a Gaussian in b_T space is used to parameterize the nonperturbative input, which emphasizes the small b_T region, just the largest difference will matter most. This is interestingly counteracting the Sudakov form factor, which suppresses small impact parameters (except for large

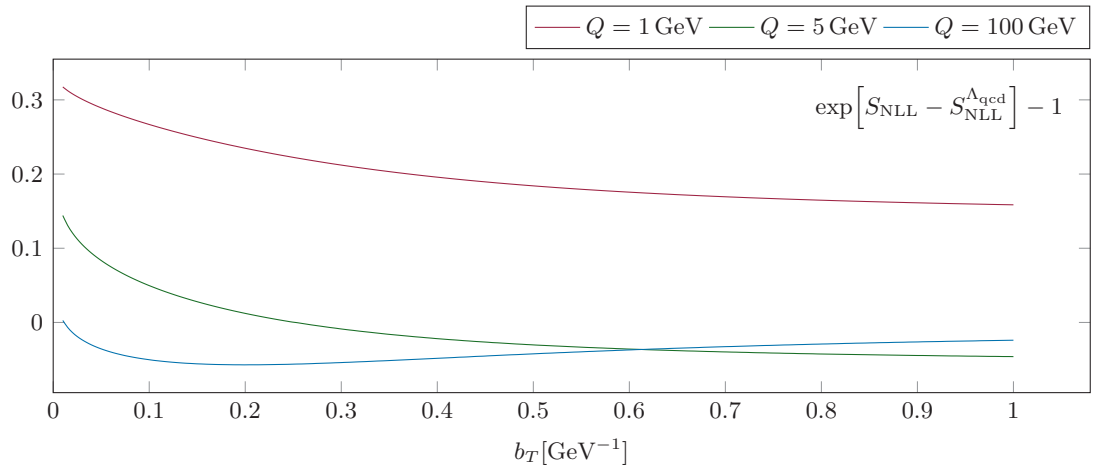


Figure 6.2: Ambiguity of the Sudakov exponentials at NLL corresponding to the Sudakov form factors shown in figure 6.1.

scales).

As already mentioned earlier we have to compute $\alpha_s(Q^2)$ to obtain a numerical result for S_{NLL}. In this computation we used the LO expression for α_s for LL and the NLO expression for *all* terms contributing to NLL. Using the LO order expression would be in clear conflict of using higher-order terms for A_1. This would have a clear impact on the results that has been shown in figures 6.1 and 6.2. For $Q = 1\,\mathrm{GeV}$ now S_{NLL} would be smaller. The ambiguity would be even more dependent on the impact parameter (roughly $-30\,\%$ for small and $-5\,\%$ for large b_T). For the slightly higher scale $Q = 5\,\mathrm{GeV}$ the situation would be similar than before, but with S_{NLL} giving the smaller result. For large scales $Q = 100\,\mathrm{GeV}$ the situation would become much worse with a ambiguity of roughly $-40\,\%$ for large impact parameters. A mixed strategy using the (N)LO expression to compute $\alpha_s(Q^2)$ for all terms that contain the (N)LO terms to calculate the Sudakov form factor would give the least dependent expressions. For the modest scale of $Q = 5\,\mathrm{GeV}$ there would be hardly any difference on the chosen form of the coupling constant. Also for the high scale $Q = 100\,\mathrm{GeV}$ the dependence would be rather flat, negligible for small and roughly $1\,\%$ for large b_T. Only for the low scale $Q = 1\,\mathrm{GeV}$ the ambiguity would be larger than $-5\,\%$ for small impact parameter but becomes smaller in magnitude for large impact parameter. However, although this result is the best in terms of least ambiguity it is also rather constructed, because in practice it is completely unusual to use different orders of the running coupling depending on the origin of a particular term. It is standard to use the highest order that is incorporated in the calculation and as it is presented in figures 6.1 and 6.2.

We note that the asymptotic behavior is determined by the b_* prescription, which causes a convergence of the impact parameter to the value chosen for b_{max}. Throughout this chapter we choose $b_{\mathrm{max}} = 0.5\,\mathrm{GeV}^{-1}$. The other common value $b_{\mathrm{max}} = 1.5\,\mathrm{GeV}^{-1}$ would not change the behavior for small impact parameters, but affect the limit the

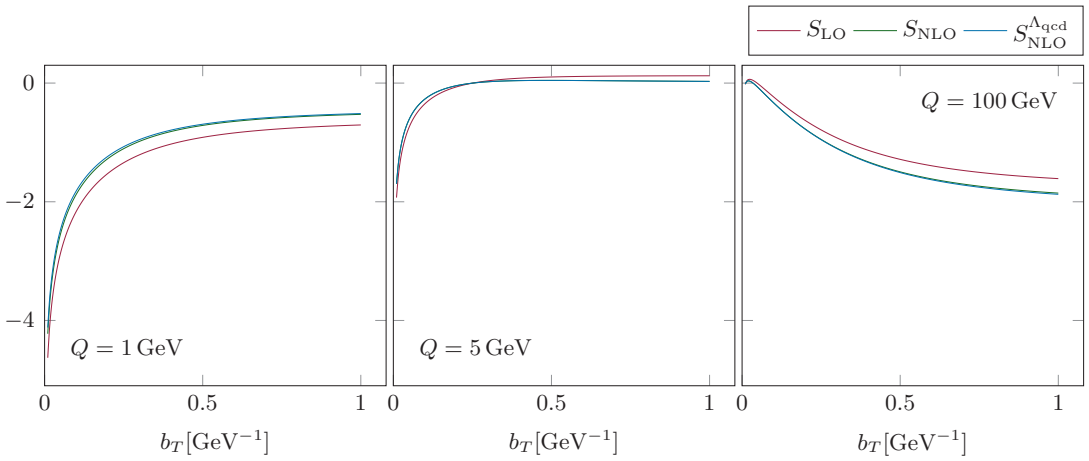


Figure 6.3: Same as figure 6.1 but using the Sudakov form factors defined in equations (6.18) and (6.19). Again the leading-order terms are equal $S_{\mathrm{LO}} = S_{\mathrm{LO}}^{\Lambda_{\mathrm{qcd}}}$ and only S_{LO} is shown.

expressions would converge to (while naturally converging later). The Sudakov form factor would be larger for small scales, while smaller for large scales. The ambiguity would be quite similar for small and modest scales but widely increased for large scales.

We now turn to a more sensible description in terms of TMDs. As mentioned in the introduction the Sudakov form factor is expressed in the TMD formalism by anomalous dimensions for the TMD PDFs and the Collins-Soper evolution kernel. As these combine the $A(\alpha_s)$ and $B(\alpha_s)$ of the Sudakov form factor it is not natural to distinguish orders by the logarithmic terms described by them. Instead it is common practice to use a consistent order for both of the anomalous dimensions. We express these again in terms known from the resummantion formalism to simplify comparisons. The leading-order terms of the TMD formalism are therefore given by

$$S_{\mathrm{LO}} = S_{A_1}^{\mathrm{LO}} + S_{B_1}^{\mathrm{LO}} \tag{6.18a}$$

$$S_{\mathrm{LO}}^{\Lambda_{\mathrm{qcd}}} = S_{A_1}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}} + S_{B_1}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}} \tag{6.18b}$$

Again we use the leading-order expression of α_s. Comparing to the leading log definitions in equations (6.16) we add the term corresponding to B_1 to include all terms that are formally given by $\gamma_F(\alpha_s)$ and $\gamma_K(\alpha_s)$ in leading order. The NLO is constructed accordingly

$$S_{\mathrm{NLO}} = S_{A_1}^{\mathrm{NLO}} + S_{B_1}^{\mathrm{NLO}} + S_{A_2}^{\mathrm{NLO}} + S_{B_2}^{\mathrm{NLO}} \tag{6.19a}$$

$$S_{\mathrm{NLO}}^{\Lambda_{\mathrm{qcd}}} = S_{A_1}^{\mathrm{NLO},\Lambda_{\mathrm{qcd}}} + S_{B_1}^{\mathrm{NLO},\Lambda_{\mathrm{qcd}}} + S_{A_2}^{\mathrm{NLO},\Lambda_{\mathrm{qcd}}} + S_{B_2}^{\mathrm{NLO},\Lambda_{\mathrm{qcd}}} \tag{6.19b}$$

Compared to the definitions of the NLL form factor in (6.17) we added again the term associated with $B(\alpha_s)$, which is B_2. Further we use the NLO expressions for all terms

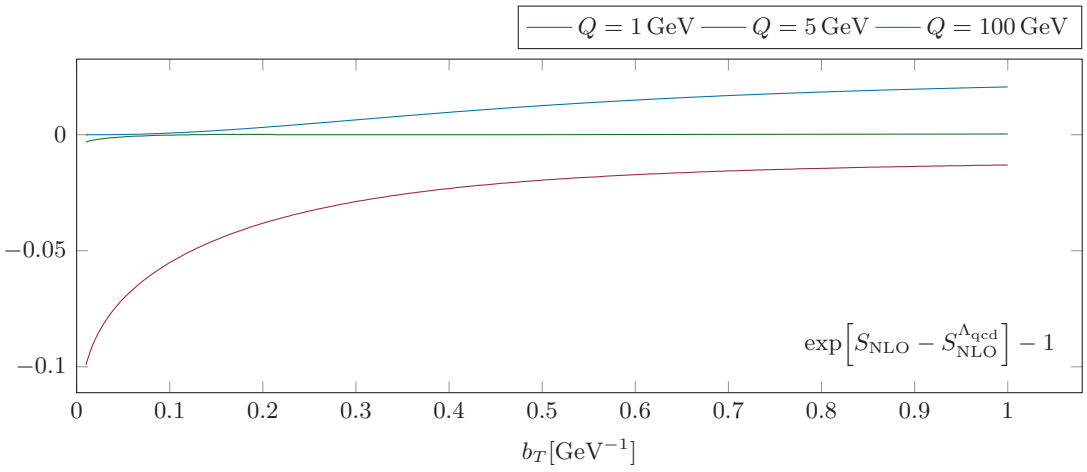


Figure 6.4: Ambiguity of the Sudakov exponentials at NLO corresponding to the Sudakov form factors shown in figure 6.3.

and not only for the terms coming with A_1. Again it does not make sense in the TMD framework to distinguish the used order for the coefficients that come with the same order of α_s. If also the terms proportional to A_2 and B_2 should be evaluated with the NLO expression is questionable. However, we decided to treat all terms equally in this respect.

In figure 6.3 we present the numerical values obtained by using equations (6.18) and (6.19) for the scales $Q = 1\,\mathrm{GeV}$, $Q = 5\,\mathrm{GeV}$ and $Q = 100\,\mathrm{GeV}$ as before. Again the leading-order terms are equal for both analytical treatments $S_{\mathrm{LO}} = S_{\mathrm{LO}}^{\Lambda_{\mathrm{qcd}}}$ and are plotted only once. In NLO the results seem to be also almost equal as is confirmed in figure 6.4. For the scale $Q = 5\,\mathrm{GeV}$ there is hardly any ambiguity between the two analytical forms, also for $Q = 100\,\mathrm{GeV}$ the difference is only about one percent for large impact parameter. Only for the low scale $Q = 1\,\mathrm{GeV}$ we can detect a measurable difference especially for small impact parameter of roughly 10 %. The analysis according to the choice of the final formula for the running coupling does not make sense in the realm of TMD evolution, because the order of the encountered coefficients is always identical to the order of the sudakov form factors. Changing the parameter b_{max} to $1.5\,\mathrm{GeV}^{-1}$ induces the same changes as discussed previously: low scales have a larger form factor, large scales a smaller form factor and the ambiguity for large scales is increased.

Interestingly the TMD like treatment has opposite effects on the form factor compared to the resummation like treatment in terms of the changes from LO to NLO. It is a good approximation that in every region where the NLO form factor is larger than the LO one for resummation it is smaller for TMDs and vice versa. The same is qualitatively true for the two analytical forms we analyze (although quantitatively very different as discussed before). Especially for the large scale under consideration large impact parameters are much more suppressed in the TMD like than in the resummation like

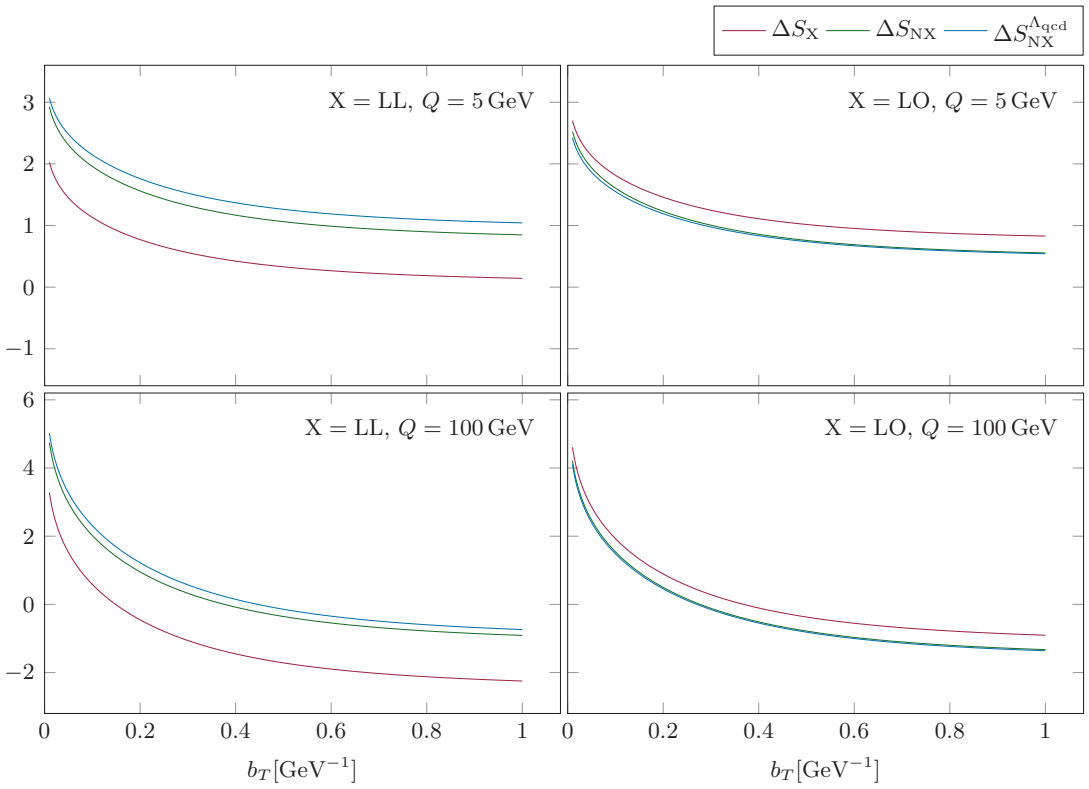


Figure 6.5: Differences of two Sudakov form factors as defined in equation (6.20). The label X distinguishes between the definition of the Sudakov form factor collecting certain logarithmic contributions and the Sudakov factor equivalent to the analytical form coming from the TMD evolution kernels.

treatment.

Finally we want to discuss the numerical consequences of the method of using the TMD evolution kernels to write the TMDs at arbitrary scale by evolving them from a fixed input scale, for example chosen to be $Q_0 = 1\,\mathrm{GeV}$ [276], instead of the scale given by μ_b. This is achieved by dividing the two distribution functions incorporating the evolution from μ_b to their scale (Q or Q_0 respectively), canceling the intrinsic distribution function evaluated at scale μ_b and giving an expression that evolves the TMD between two arbitrary scales, see equation (6.9). Therefore we will analyze

$$\Delta S(Q) \equiv S(Q) - S(Q_0 = 1\,\mathrm{GeV}) \tag{6.20}$$

This particular way of using the Sudakov form factor has been used in resummation as well as TMD schemes. The corresponding definiton for the TMDs is given in [276], as it has been introduces in the previous section, see equations (6.8) and (6.9).

The results are shown in figures 6.5 and 6.6, where as always the superscript Λ_{qcd} indicates the use of the expansion in $\ln^{-1}(\mu/\Lambda_{\mathrm{qcd}})$, whereas no superscript indicates

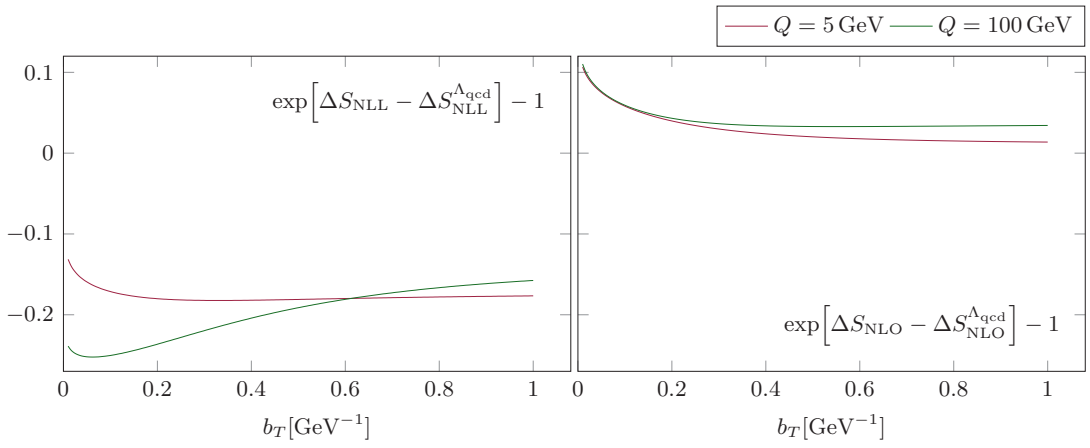


Figure 6.6: Ambiguity of the difference of two Sudakov form factors ΔS as shown in figure 6.5.

the use of the solution of the renormalization group equation to solve the integrals in the Sudakov form factor. In figure 6.5 the subscript distinguishes the different terms that are naturally taken into account by the definitions made in (6.16), (6.17), (6.18) and (6.19). First we can detect that the ambiguity is dominated by the terms at Q_0. Because these disagree in sign for the resummation and the TMD forms, also the ambiguity of ΔS differs in sign. In the construction of ΔS the ambiguity is enhanced, as it has different signs for Q_0 and the other two considered scales. The only exception is for small impact parameter and $Q = 5\,\mathrm{GeV}$, especially for the resummation like Sudakov. In general the ambiguity of the resummation like form factor is much larger than for the TMD like. In the former the ambiguity is between 10 % and 30 % with a even large value in the limit of large impact parameter. In the latter the ambiguity is smaller than 10 % and only at percent level for large impact parameters. For both the ambiguity is mainly enhanced for large scales Q.

6.3 Conclusions

In this chapter we presented a detailed analysis of the Sudakov form factor in respect of its numerical treatment. Because it is defined by an integral inside an exponential it is numerically extremely time consuming to calculate the integral itself, where it could be treated using an iterative determination of the strong coupling constant α_s . The only alternative is to solve the integral analytically. To perform this integration it is necessary to choose a representation of α_s. Because to a fixed order more than one such representation exists an ambiguity is introduced. The effects of this ambiguity should be expected to be small as all these representations are correct to their specific order. However, this is not the case. Our analysis reveals a measurable impact of the chosen analytical form of α_s on the Sudakov form factor as soon as non-leading terms are taken into account. Because the form factor is exponentiated this effect becomes

even more important.

We investigated two sensible forms of the Sudakov form factor. First we took care of taking only terms that contribute to the same 'logarithm towers' as is often performed by the resummation community. Second we chose the terms according to the TMD formalism where the Sudakov is decomposed into two integrals originating from the TMD PDF anomalous dimension and the Collins-Soper evolution kernel, where the order of α_s stays to be the decisive parameter. Both forms are equivalent in the leading order but lead to numerical differences depending on the analytical form of α_s for higher orders. The ambiguity introduced by this difference is larger for the resummed expression (at NLL) than for the evolution of TMDs (at NLO), although the former sorts the perturbative series more carefully. Finally we determined the possibility to use a difference of two Sudakov form factors instead of one. This leads in most regions to an enhancement of the ambiguity, especially if one of the scales is small.

Our results imply that all findings that are obtained by resummation, perhaps not only for the small transverse momentum, are dependent on the chosen form of α_s. This is especially true for all fits that are performed to determine nonperturbative parts that occur in this calculations, because these parts are usually parametrized by Gaussian factors that emphasize the region of small impact parameters, which is especially sensitive to the found ambiguity. It is even possible that the seemingly unimportant choice of the analytical form of α_s determines whether a fit converges or not. We suggest that for the sake of reproducibility this detail should be mentioned wherever it is used. Besides this work demonstrates that the method of using the difference of two Sudakov form factors is even more ambiguous as it emphasizes the disagreements of different scales. All consequences for this method follow analogously.

Chapter 7

Conclusions and outlook

I love it when a plan comes together.
— *John 'Hannibal' Smith*

The Standard Model is the currently the most successful theory in particle physics. Part of it is Quantum Chromodynamics, which is concerned with the strong force responsible for the interaction of colored particles. To describe high-energy collisions the preferred tool is perturbative Quantum Chromodynamics. For many observables it is sufficient to expand the perturbative series in the strong coupling constant to the first or second order to achieve a reasonably good prediction. However, in certain regions of the phase space the series fails to describe observations. This is due to the incomplete cancellation of real and virtual diagram contributions at higher orders. Logarithmic remainders are left behind which spoil the series expansion in these phase space regions. To get a reliable theoretical prediction it is necessary to reorganize the perturbative series in terms of the strong coupling constant *and* the logarithmic terms. To perform the 'resummation' it is unavoidable to transform the formula of the cross section to achieve the required factorization. The kind of transformation depends on the specific cross section. In this thesis we treated the resummation of small transverse momenta, which is typically achieved in impact parameter space—the Fourier transform of the transverse momentum. Since the perturbative expansion of Quantum Chromodynamics is only valid at high momenta but the inverse transformation technically uses all possible momenta, it is necessary to regularize the analytical formula, expressing the need to consider also nonperturbative effects.

In this thesis we treated several different aspects of the transverse momentum resummation. The main goal was to further develop the 'complex prescription', which bypasses the problem of finding a meaningful extension of the perturbative resummation expression without suppressing physical content by avoiding the Landau pole in the complex plane. With this in mind we extended a common general purpose integration routine by completely rewriting it to be usable on a CPU as well as a GPU. The latter allows to perform the more complicated numerical calculations induced by the complex prescription in a reasonable amount of time. Further we investigated in a new method to parametrize the parton distribution functions in Mellin moment space as by the complex prescription it becomes necessary to evolve the parton distribution

functions to a complex scale, since the analytical solution of the DGLAP evolution equations is given in Mellin moment space. Finally we examined the numerical behavior of the Sudakov form factor for different expressions of the running coupling, which might play an important role for the future determination of the nonperturbative ingredients. In fact all these investigations are made in advance of a global fit to determine a reasonable nonperturbative input. To make the fit reachable the numerical speed offered by the GPU is crucial, while the other topics are mainly important for precision and reproducibility. Especially the specific form of the Sudakov form factor can potentially determine whether a fit converges at all.

Furthermore we investigated the transverse momentum resummation of the semi inclusive deep inelastic scattering process. For the first time we calculated also subleading terms, developing techniques that can also be used for different processes. This can serve as the first step to a more complete understanding of the transverse momentum resummation as a whole, if it is possible to resum the found terms similar to the leading contributions.

Finally, out of the line, we performed a global analysis for all currently available experimental data of Drell-Yan angular distributions to a higher order than has been done before. The results imply a very good agreement of theory with experiment, confirming Quantum Chromodynamics and hence the Standard Model.

Appendices

Appendix A

QCD Feynman rules

> The palest of ink is better than the best memory.
>
> — *Chinese proverb*

In this appendix we present the QCD Feynman rules as they follow from the Lagrangian in equation (1.13). In the first line of the final step in (1.13) are the kinematic terms that correspond to the propagators[1] shown in figure A.1. Note that in the literature another way to define the gauge fixing term is also common, with

$$\xi \to \frac{1}{\lambda} \tag{A.1}$$

which changes the gauge dependent term in the gluon propagator as

$$\frac{\xi - 1}{\xi} \to 1 - \lambda \tag{A.2}$$

The second line of equation (1.13) consists of the quark-gluon and the ghost-gluon interaction. The corresponding diagrams are depicted in figure A.2. Note that the ghost-gluon interaction depends on the momentum of the ghost. Finally in the third line of equation (1.13) the QCD feature that causes the most important difference to quantum electrodynamics—the self interaction of the gluons due to the non-abelian structure of QCD—is shown in figure A.3.

[1] As mentioned in section 1.1 we present the covariant gauges, which simplifies the gluon propagator but needs the ghost field to compensate unphysical polarization states of the gluon.

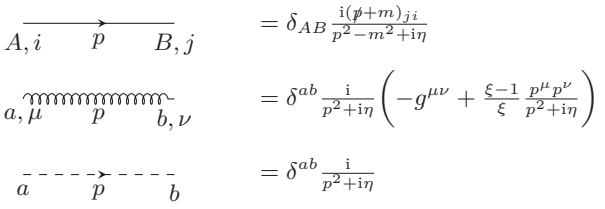


Figure A.1: QCD Feynman rules for quark, gluon and ghost propagator. The quark line is solid, the gluon line curly and the ghost is indicated by a dashed line. The momentum flow p is always from left to right.

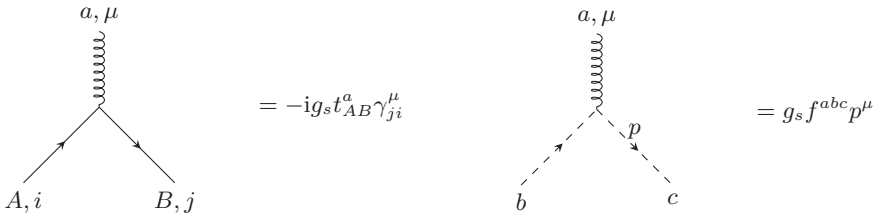


Figure A.2: QCD Feynman rules for the quark-gluon and the ghost-gluon vertex. The lines decorations are similar to figure A.1. The momentum flow of the ghost is from the left to the right.

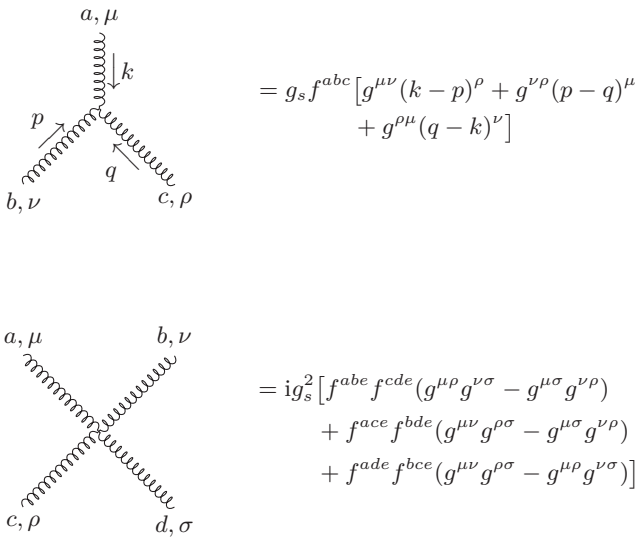


Figure A.3: QCD Feynman rules for the gluon self interactions, i.e. the three and the four gluon vertex.

Appendix B

QCD splitting functions

In this appendix we present the analytical expressions of the six independent Altarelli-Parisi splitting functions up to next-to-leading order as defined in equation (1.74). The leading-order expressions have been calculated in [102]

$$P_{\mathrm{ns}}^{\pm(0)}(z) = C_F\left[3\delta(1-z) + 2\frac{1+z^2}{(1-z)_+}\right] \tag{B.1a}$$

$$P_{\mathrm{ns}}^{\mathrm{s}(0)}(z) = 0 \tag{B.1b}$$

$$P_{\mathrm{ps}}^{(0)}(z) = 0 \tag{B.1c}$$

$$P_{qq}^{(0)}(z) = 2C_F\frac{1+(1-z)^2}{z} \tag{B.1d}$$

$$P_{qg}^{(0)}(z) = 4n_f T_R\big(z^2+(1-z)^2\big) \tag{B.1e}$$

$$P_{gg}^{(0)}(z) = \frac{11C_A - 4n_f T_R}{3}\delta(1-z) + 4C_A\left(\frac{z}{(1-z)_+} + \frac{1-z}{z} + z(1-z)\right) \tag{B.1f}$$

and the next-to-leading expressions in [104, 105, 107–113]

$$\begin{aligned}
P_{\mathrm{ns}}^{+(1)}(z) = 4C_F^2\Bigg[&\delta(1-z)\left(\frac{3}{8} - \frac{\pi^2}{2} + 6\zeta(3)\right) - \frac{1}{(1-z)_+}\big(4\ln(z)\ln(1-z) + 3\ln(z)\big) \\
&+ 2S_2(z)\left(\frac{2}{1+z} - 1 + z\right) - \frac{1}{2}(1+z)\ln^2(z) \\
&+ 2(1+z)\ln(z)\ln(1-z) + (2-z)\ln(z) - (1-z)\Bigg] \\
+ 4C_F C_A\Bigg[&\delta(1-z)\left(\frac{17}{24} + \frac{11}{18}\pi^2 - 4\zeta(3)\right) \\
&+ \frac{1}{(1-z)_+}\left(4\ln^2(z) + \frac{11}{3}\ln(z) + \frac{67}{9} - \frac{\pi^2}{3}\right) \\
&- S_2(z)\left(\frac{2}{1+z} - 1 + z\right) - \frac{1}{2}(1+z)\ln^2(z) - \frac{11}{6}(1+z)\ln(z) \\
&+ \left(\frac{\pi^2}{6} - \frac{151}{18}\right)(1+z) + \frac{28}{3}\Bigg] \\
- 4C_F T_R n_f\Bigg[&\delta(1-z)\left(\frac{1}{6} + \frac{2}{9}\pi^2\right) + \frac{4}{3}(1-z)
\end{aligned}$$

$$+\left(\frac{2}{3}\ln(z)+\frac{10}{9}\right)\left(\frac{2}{(1-z)_+}-1-z\right)\Bigg] \tag{B.2a}$$

$$P_{\rm ns}^{-(1)}(z) = P_{\rm ns}^{+(1)}(z) - 16C_F\left(C_F-\frac{C_A}{2}\right)\left[(1+z)\ln(z)+2(1-z)+S_2(z)\left(\frac{2}{1+z}-1+z\right)\right] \tag{B.2b}$$

$$P_{\rm ns}^{{\rm s}(1)}(z) = 0 \tag{B.2c}$$

$$P_{\rm ps}^{(1)}(z) = 4C_F T_R n_f\left[-2(1+z)\ln^2(z)+\left(10z+\frac{16}{3}z^2+2\right)\ln(z) + \frac{40}{9}\frac{1}{z}-\frac{112}{9}z^2+4(3z-1)\right] \tag{B.2d}$$

$$\begin{aligned}P_{gq}^{(1)}(z) = {}& 4C_F^2\left[-\frac{5}{2}-\frac{7}{2}z+\left(2+\frac{7}{2}z\right)\ln(z)-\left(1-\frac{1}{2}z\right)\ln^2 -2z\ln(1-z)\right.\\ &\left.-\left(2\ln(1-z)+\ln^2(1-z)\frac{1+(1-z)^2}{z}\right)\right]\\ &+4C_F C_A\left[\frac{28}{9}+\frac{65}{18}z+\frac{44}{9}z^2-\left(12+5z+\frac{8}{3}z^2\right)\ln(z)+(4+z)\ln^2(z)\right.\\ &+2z\ln(1-z)-S_2(z)\frac{1+(1+z)^2}{z}\\ &\left.+\left(\frac{1}{2}-2\ln(z)\ln(1-z)+\frac{1}{2}\ln^2(z)+\frac{11}{3}\ln(1-z)\right.\right.\\ &\left.\left.+\ln^2(1-z)-\frac{\pi^2}{6}\right)\frac{1+(1-z)^2}{z}\right]\\ &-4C_F T_R n_f\left[\frac{4}{3}z+\left(\frac{20}{9}+\frac{4}{3}\ln(1-z)\right)\frac{1+(1-z)^2}{z}\right]\end{aligned} \tag{B.2e}$$

$$\begin{aligned}P_{qg}^{(1)}(z) = {}& 4C_F T_R n_f\left[4-9z-(1-4z)\ln(z)-(1-2z)\ln^2(z)+4\ln(1-z)\right.\\ &\left.+\left(2\ln^2\left(\frac{1-z}{z}\right)-4\ln\left(\frac{1-z}{z}\right)-\frac{2\pi^2}{3}+10\right)\left(z^2+(1-z)^2\right)\right]\\ &+4C_A T_R n_f\left[\frac{40}{9z}+\frac{182}{9}+\frac{14}{9}z+\left(\frac{136}{3}z-\frac{38}{3}\right)\ln(z)-4\ln(1-z)\right.\\ &-(2+8z)\ln^2(z)+2\left(z^2+(1+z)^2\right)S_2(z)\\ &\left.-\left(\ln^2(z)-\frac{44}{3}\ln(z)+2\ln^2(1-z)-4\ln(1-z)\right.\right.\\ &\left.\left.-\frac{\pi^2}{3}+\frac{218}{9}\right)\left(z^2+(1-z)^2\right)\right]\end{aligned} \tag{B.2f}$$

$$P_{gg}^{(1)}(z) = 4C_F T_R n_f\left[-\delta(1-z)-16+8z+\frac{20}{3}z^2+\frac{4}{3z}-(6+10z)\ln(z)\right.$$

$$
\begin{aligned}
&\quad - 2(1+z)\ln^2(z)\Big] \\
&+ 4C_A T_R n_f \Big[-\frac{4}{3}\delta(1-z) + 2(1-z) + \frac{26}{9}\left(z^2 - \frac{1}{z}\right) - \frac{4}{3}(1+z)\ln(z) \\
&\quad - \frac{20}{9}\left(\frac{1}{(1-z)_+} + \frac{1}{z} - 2 + z(1-z)\right)\Big] \\
&+ 4C_A^2 \Big[\left(\frac{8}{3} + 3\zeta(3)\right)\delta(1-z) + \frac{27}{2}(1-z) + \frac{67}{9}\left(z^2 - \frac{1}{z}\right) \\
&\quad - \left(\frac{25}{3} - \frac{11}{3}z + \frac{44}{3}z^2\right)\ln(z) + 4(1+z)\ln^2(z) \\
&\quad + 2S_2(z)\left(\frac{1}{1+z} - \frac{1}{z} - 2 - z(1+z)\right) \\
&\quad + \left(\frac{67}{9} - 4\ln(z)\ln(1-z) + \ln^2(z) - \frac{\pi^2}{3}\right) \\
&\quad \times \left(\frac{1}{(1-z)_+} + \frac{1}{z} - 2 + z(1-z)\right)\Big]
\end{aligned}
\tag{B.2g}
$$

A full summary is provided also in [278]. C_F, C_A and T_R are the usual abbreviations of the SU(3) Casimir operators. The function $S_2(z)$ is defined by

$$
S_2(z) \equiv \int_{\frac{z}{1+z}}^{\frac{1}{1+z}} \frac{\mathrm{d}x}{x} \ln\left(\frac{1-x}{x}\right) \tag{B.3}
$$

and $\zeta(x)$ is the Riemann zeta function.

In fact most of the calculations have been performed in Mellin space [104, 105, 107–109, 113]. Subsequently it has been made much effort to the Mellin transformation and its continuation for complex Mellin moments [279–283] (even more important at NNLO). The implementation of the DGLAP evolution presented in section C.7 is also performed in Mellin space. The relevant moments are defined as Mellin transform of the splitting functions in longitudinal momentum space, see equation (1.76b). At leading order they are given by

$$
\gamma_{\mathrm{ns}}^{\pm(0)}(N) = C_F\left[3 + \frac{2}{N(N+1)} - 4S_1(N)\right] \tag{B.4a}
$$

$$
\gamma_{\mathrm{ns}}^{\mathrm{s}(0)}(N) = 0 \tag{B.4b}
$$

$$
\gamma_{\mathrm{ps}}^{(0)}(N) = 0 \tag{B.4c}
$$

$$
\begin{aligned}
\gamma_{qq}^{(0)}(N) &= 2C_F\left[\frac{2}{N-1} - \frac{2}{N} + \frac{1}{N+1}\right] \\
&= 2C_F\frac{N(N+1)+2}{N(N-1)(N+1)}
\end{aligned}
\tag{B.4d}
$$

$$
\gamma_{qg}^{(0)}(N) = 4n_f T_R\left[\frac{1}{N} - \frac{2}{N+1} + \frac{2}{N+2}\right]
$$

$$= 4n_f T_R \frac{N(N+1)+2}{N(N+1)(N+2)} \tag{B.4e}$$

$$\gamma_{gg}^{(0)}(N) = \frac{11C_A - 4n_f T_R}{3} + 4C_A\left[\frac{2N(N+1)+2}{N(N-1)(N+1)(N+2)} - S_1(N)\right] \tag{B.4f}$$

while the next-to-leading-order coefficients read

$$\begin{aligned}
\gamma_{\rm ns}^{\pm(1)}(N) = 4C_F^2\Bigg[&\frac{3}{8} - 2S_1(N)\frac{2N+1}{N^2(N+1)^2} - 3S_2(N) - 8\tilde{S}(N,\pm1) + S_3'\left(\frac{N}{2},\pm1\right)\\
&- 2\left(2S_1(N) - \frac{1}{N(N+1)}\right)\left(S_2(N) - S_2'\left(\frac{N}{2},\pm1\right)\right)\\
&+ \frac{3N^3+N^2-1}{N^3(N+1)^3} \pm 2\frac{2N^2+2N+1}{N^3(N+1)^3}\Bigg]\\
+ 2C_F C_A\Bigg[&\frac{17}{12} - \frac{134}{9}S_1(N) + \frac{22}{3}S_2(N) + 8\tilde{S}(N,\pm1) - S_3'\left(\frac{N}{2},\pm1\right)\\
&+ 2\left(2S_1(N) - \frac{1}{N(N+1)}\right)\left(2S_2(N) - S_2'\left(\frac{N}{2},\pm1\right)\right)\\
&+ \frac{151N^4+236N^3+88N^2+3N+18}{9N^3(N+1)^3} \mp 2\frac{2N^2+2N+1}{N^3(N+1)^3}\Bigg]\\
+ \frac{2C_F T_R n_f}{3}\Bigg[&\frac{40}{3}S_1(N) - 8S_2(N) - 1 - \frac{4}{3}\frac{11N^2+5N-3}{N^2(N+1)^2}\Bigg]
\end{aligned} \tag{B.5a}$$

$$\gamma_{\rm ns}^{\rm s(1)}(N) = 0 \tag{B.5b}$$

$$\gamma_{\rm ps}^{(1)}(N) = 8C_F T_R n_f \frac{5N^5+32N^4+49N^3+38N^2+28N+8}{(N-1)N^3(N+1)^3(N+2)^2} \tag{B.5c}$$

$$\begin{aligned}
\gamma_{gq}^{(1)}(N) = 2C_F^2\Bigg[&\left(10S_1(N) - 2S_2(N) - 2(S_1(N))^2\right)\frac{N^2+N+2}{(N-1)N(N+1)}\\
&- \frac{4}{(N+1)^2}S_1(N)\\
&- \frac{12N^6+30N^5+43N^4+28N^3-N^2-12N-4}{(N-1)N^3(N+1)^3}\Bigg]\\
+ 4C_F C_A\Bigg[&\left((S_1(N))^2 + S_2(N) - S_2'\left(\frac{N}{2},1\right)\right)\frac{N^2+N+2}{(N-1)N(N+1)}\\
&- \frac{17N^4+41N^2-22N-12}{3(N-1)^2N^2(N+1)}S_1(N)\\
&+ \frac{1}{9(N-1)^2N^3(N+1)^3(N+2)^2}\\
&\times\left(109N^9+621N^8+1400N^7+1678N^6+695N^5-1031N^4\right.\\
&\left.- 1304N^3 - 152N^2 + 432N + 144\right)\Bigg]
\end{aligned}$$

$$+\frac{16}{3}C_F T_R n_f\left[\left(S_1(N)-\frac{8}{3}\right)\frac{N^2+N+2}{(N-1)N(N+1)}+\frac{1}{(N+1)^2}\right] \tag{B.5d}$$

$$\begin{aligned}\gamma_{qg}^{(1)}(N) = 4C_F T_R n_f\Bigg[&\left(2(S_1(N))^2-2S_2(N)+5\right)\frac{N^2+N+2}{N(N+1)(N+2)}-\frac{4}{N^2}S_1(N)\\&+\frac{11N^4+26N^3+15N^2+8N+4}{N^3(N+1)^3(N+2)}\Bigg]\\&+4C_A T_R n_f\Bigg[2\left(S_2(N)-(S_1(N))^2-S_2'\left(\frac{N}{2},1\right)\right)\frac{N^2+N+2}{N(N+1)(N+2)}\\&\quad+8S_1(N)\frac{2N+3}{(N+1)^2(N+2)^2}\\&\quad+\frac{2}{(N-1)N^3(N+1)^3(N+2)^3}\\&\qquad\times\left(N^9+6N^8+15N^7+25N^6+36N^5+85N^4+128N^3\right.\\&\qquad\left.+104N^2+64N+16\right)\Bigg]\end{aligned} \tag{B.5e}$$

$$\begin{aligned}\gamma_{gg}^{(1)}(N) = &\frac{16C_A T_R n_f}{3}\left[\frac{5}{3}S_1(N)-1-\frac{1}{3}\frac{19N^4+38N^3+47N^2+28N+6}{(N-1)N^2(N+1)^2(N+2)}\right]\\&-4C_F T_R n_f\left[1+2\frac{2N^6+4N^5+N^4-10N^3-5N^2-4N-4}{(N-1)N^3(N+1)^3(N+2)}\right]\\&-2C_A^2\Bigg[\frac{134}{9}S_1(N)+16S_1(N)\frac{2N^5+5N^4+8N^3+7N^2-2N-2}{(N-1)^2N^2(N+1)^2(N+2)^2}-\frac{16}{3}\\&\quad+8\frac{N^2+N+1}{(N-1)N(N+1)(N+2)}S_2'\left(\frac{N}{2},1\right)-4S_1(N)S_2'\left(\frac{N}{2},1\right)\\&\quad+8\tilde{S}(N,1)-S_3'\left(\frac{N}{2},1\right)\\&\quad-\frac{1}{9(N-1)^2N^3(N+1)^3(N+2)^3}\\&\qquad\times\left(457N^9+2742N^8+6040N^7+6098N^6+1567N^5-2344N^4\right.\\&\qquad\left.-1632N^3+560N^2+1488N+576\right)\Bigg]\end{aligned} \tag{B.5f}$$

The functions $S_i(N)$, $S_i'(N,\eta)$ and $\tilde{S}(N,\eta)$ are defined by

$$S_1(N)\equiv\sum_{m=1}^{N}\frac{1}{m}=\psi_0(N+1)+\gamma_E \tag{B.6a}$$

$$S_2(N)\equiv\sum_{m=1}^{N}\frac{1}{m^2}=-\psi_1(N+1)+\zeta(2) \tag{B.6b}$$

$$S_3(N) \equiv \sum_{m=1}^{N} \frac{1}{m^3} = \frac{1}{2}\psi_2(N+1) + \zeta(3) \tag{B.6c}$$

$$S_i'(N,\eta) \equiv 2^{i-1} \sum_{m=1}^{2N} \frac{1+(-1)^m}{m^i} = \frac{1+\eta}{2} S_i(N) + \frac{1-\eta}{2} S_i\left(N-\frac{1}{2}\right) \tag{B.6d}$$

$$\begin{aligned}\tilde{S}(N,\eta) &\equiv \sum_{m=1}^{N} \frac{(-1)^m}{m^2} S_1(m) \\ &= -\frac{5}{8}\zeta(3) \\ &\quad + \eta\left[\frac{S_1(N)}{N^2} - \frac{\zeta(2)}{2}\left(\psi_0\left(N+\frac{1}{2}\right) - \psi_0\left(\frac{N}{2}\right)\right) + \int_0^1 \mathrm{d}z\, z^{N-1} \frac{\mathrm{Li}_2(z)}{1+z}\right]\end{aligned} \tag{B.6e}$$

where γ_E and $\psi_m(N)$ are the Euler–Mascheroni constant and the polygamma function of order m respectively.

Appendix C

fi*n*ael

> Talk is cheap. Show me the code.
> — *Linus Torvalds*

In this appendix we present fi*n*ael. Its name is a recursive acronym in the spirit of GNU or Ti*k*Z and stands for 'fi*n*ael is not an entire library'. This already indicates that fi*n*ael is not a classic library, as it serves multiple purposes. Its perhaps most important part, which initially initiated the whole project, is the general purpose Monte Carlo integration routine Vegas (for a theoretical description and performance measurements see chapter 2), that can be executed on an ordinary CPU but also using a GPU. But several other services are offered by fi*n*ael, which we will present here, starting from very general topics and dealing more specific parts in the subsequent sections. We will not present every single class, but the classes that are intended for direct use or are absolutely necessary to understand other code listings.

During the development of fi*n*ael we took special care to produce modern, object oriented and well designed C++ code. Although the initial goal was an increase of performance, we decided us for better design, if both interests were in conflict and the performance penalty was not overwhelming.

To prevent naming conflicts with other libraries all classes are encapsulated in the `namespace finael`. Also the distinct parts are separated into further namespaces. These indicate by the prefix 'cu', whether they are designed with the GPU in mind (but work on the CPU as well). Thus namespaces lacking the prefix are for CPU use only. In the listings as well as in the text we will omit the namespaces, if not necessary for insight. The naming conventions are defined by: namespaces, variables and functions are `writtenLikeThis`, while classes `LookLikeThis`. The only exception are the structs that contain constants only, which are written exclusively in capital letters.

fi*n*ael is written header-only similar to the boost library [284]. This has mainly two reasons. First it is not heavy weight and should not have too large impact on compilation times. Second almost all classes are templates. The code is available online [285]. It is published under the MIT license [286].

C.1 Common

To discriminate host (CPU) from device (GPU) code we use in all classes that are designed for use with the GPU the definitions

cudaHostDevice.h

```
#ifdef __CUDACC__
   #define CUDA_HOST_DEVICE __host__ __device__
   #define CUDA_HOST __host__
   #define CUDA_DEVICE __device__
   enum class CudaHostDeviceType {HOST, DEVICE};
#else // __CUDACC__
   #define CUDA_HOST_DEVICE
   #define CUDA_HOST
   #define CUDA_DEVICE
   enum class CudaHostDeviceType {HOST};
#endif // __CUDACC__
```

The macros are applied to functions whether they can be executed on the host, the device or both. The **`enum class CudaHostDeviceType`** is designated to be used as template parameter for several classes to be assigned to the respective processing unit type. If no GPU (and hence typically also no nvcc compiler) is available everything is defined such, that it works smoothly with standard C++ compilers.

C.2 Tools

When working with a GPU the very first step is always to identify the GPU and set the device. Except for the situation of a single GPU on a system with only one user, this ensures that indeed a device is ready to execute kernels or perform memory transfers. To simplify the procedure we offer the very simple tool `DeviceSelector` with the interface

DeviceSelector.h

```
class DeviceSelector
{
public:
   DeviceSelector() = delete;

   static int setFirstAvailable();
};
```

Compiled with the nvcc compiler it will loop through all GPUs connected to the system and try to occupy one of them. The return value is the index of the GPU that has been set successfully. If no device is available it exits the program. For pure host compilers

the function always returns zero.

To receive more basic informations about the device one may use the helper class `DeviceInfo`, which collects some data of the current device and can print them into `std::cout`. Its synopsis is

DeviceInfo.h

```
class DeviceInfo
{
public:
   static constexpr unsigned int hostBlockDimDefault = 1024;
   static constexpr unsigned int hostWarpSize = 32;

   CUDA_HOST DeviceInfo();

   CUDA_HOST void update();

   CUDA_HOST unsigned int getMaxBlockDim() const;

   CUDA_HOST unsigned int getWarpSize() const;

   CUDA_HOST void print() const;
};
```

The perhaps most difficult part when handling with GPU code is proper memory handling. For this purpose we wrote the **`class SmartArray`**, which is our work horse for the memory management of CUDA applications. Its public interface is

SmartArray.h

```
template<typename T, CudaHostDeviceType hostDeviceType = CudaHostDeviceType::HOST>
class SmartArray
{
public:
   using type = T;
   static constexpr CudaHostDeviceType hostOrDevice = hostDeviceType;

   friend class SmartArray<T, CudaHostDeviceType::HOST>;
   #ifdef __CUDACC__
   friend class SmartArray<T, CudaHostDeviceType::DEVICE>;
   #endif // __CUDACC__

   CUDA_HOST SmartArray();

   CUDA_HOST explicit SmartArray( const unsigned int sizeIn );

   template<class InputIter>
   CUDA_HOST SmartArray( InputIter first, InputIter last );

   CUDA_HOST SmartArray( const SmartArray<T, hostDeviceType>& orig );

   template<CudaHostDeviceType SourceType>
   CUDA_HOST SmartArray( const SmartArray<T, SourceType>& orig );
```

```
    CUDA_HOST ~SmartArray();

    CUDA_HOST SmartArray<T, hostDeviceType>& operator=( SmartArray<T, hostDeviceType> orig
    ↪  );

    template<CudaHostDeviceType SourceType>
    CUDA_HOST SmartArray<T, hostDeviceType>& operator=( const SmartArray<T, SourceType>&
    ↪  orig );

    template<class Array>
    CUDA_HOST SmartArray<T, hostDeviceType>& operator=( const Array& orig );

    CUDA_HOST_DEVICE T& operator[]( const unsigned int index )

    CUDA_HOST_DEVICE T& at( const unsigned int index )

    CUDA_HOST_DEVICE const T& operator[]( const unsigned int index ) const;

    CUDA_HOST_DEVICE const T& at( const unsigned int index ) const;

    CUDA_HOST T getToHost( const unsigned int index ) const;

    CUDA_HOST T getToHostWithIndexCheck( const unsigned int index ) const;

    CUDA_HOST void newArray( const unsigned int newSize );

    template<CudaHostDeviceType SourceType, class Array>
    CUDA_HOST void newAssign( const Array& orig );

    CUDA_HOST const T* data() const;

    CUDA_HOST_DEVICE unsigned int size() const;

    CUDA_HOST unsigned int useCount() const;

    void swap( SmartArray<T, hostDeviceType>& orig );
};
```

It is obviously a mixture of a container class and a smart pointer. In principle it shares the basic idea of the `std::shared_ptr`, being a reference counting object. All initiated objects live on the host, but manage memory on the host or device depending on their second template argument `hostDeviceType`. Copy construction and assignment are extremely cheap if both objects have the exact same type due to the reference counting ansatz which does not perform a deep copy. A copy construction or an assignment of two `SmartArrays` with different `CudaHostDeviceTypes` creates a completely new object, copying the managed data from the host to the device or vice versa. To do the same if both objects are of the same type, one can use the function `newAssign`. But it is important that the template argument `Array` is a type that controls contiguous memory, otherwise the behavior is undefined. The same is true for the corresponding assignment operator and (of course) the constructor using iterators. To check the number of objects referring to the same data, use the function `useCount`. The part

that is more related to a container class manifests in typical member functions like the subscript operators, the `at` or the `size` function. They offer access to the managed memory on the host or device, depending on the `CudaHostDeviceType`. For device arrays the member functions `getToHost` and `getToHostWithIndexCheck` copy also single data back to the host. However, this is quite ineffective and should be used only if absolutely necessary. The index checking functions will print a warning via `printf`, because throwing an exception is not feasible on the device.

To make a full C++ feeling possible and avoid raw pointers everywhere we also needed to write a stack object for arrays, similar to `std::array`, as `SmartArray` is a heap object like `std::vector`. Albeit we decided to alter the interface a bit, being now

Array.h

```
template<typename T, unsigned int size>
class Array
{
public:
   using value_type = T;
   static constexpr unsigned int SIZE = size;

   CUDA_HOST_DEVICE Array();

   explicit CUDA_HOST_DEVICE Array( const T& value );

   CUDA_HOST_DEVICE Array( const T (&data_in)[size] );

   CUDA_HOST_DEVICE void set( const unsigned int index, const T& value );

   CUDA_HOST_DEVICE void setAt( const unsigned int index, const T& value );

   CUDA_HOST_DEVICE void set( const T& value );

   CUDA_HOST_DEVICE void set( const T (&data_in)[size] );

   CUDA_HOST_DEVICE T& operator[]( const unsigned int index );

   CUDA_HOST_DEVICE T& at( const unsigned int index );

   CUDA_HOST_DEVICE const T& operator[]( const unsigned int index ) const;

   CUDA_HOST_DEVICE const T& at( const unsigned int index ) const;
};
```

The obvious difference to `std::array` are the functions `set` and `setAt` and the additional constructors. They all aim for a more convenient construction and change of all data stored in the object. The function taking only a `value` will assign that value to all indices at once. The functions with pointer arguments are mainly to allow kind of `std::initializer_list` like behavior. The functions taking an `index` as well as a `value` are superfluous due to the standard access functions but are provided as they give a more complete interface having the other `set` functions anyway. Similar

to the subscript operator and the `at` function, which give access to the managed data, also the `set` and the `setAt` function access the data without and with an index check respectively.

Finally we missed also `std::pair` on the device, and thus defined the simple struct

Pair.h

```
template<typename T1, typename T2>
struct Pair
{
   using first_type = T1;
   using second_type = T2;

   T1 first;
   T2 second;

   CUDA_HOST_DEVICE Pair();

   CUDA_HOST_DEVICE Pair( T1 firstInput, T2 secondInput );
};
```

We like to mention that `std::pair` is sufficient to access the members (and clearly the types) as long as no member functions are used on the device. However, this excludes constructors and assignment operators (except the copy constructor during the kernel call). To perform these quite desirable actions also on the device one has to use `Pair`.

Only for the host we provide a simple tool to measure execution times, which has also been used for all measurements in chapter 2. The synopsis is

StopWatch.h

```
template<class stdChronoDuration>
class StopWatch
{
public:
   using clock = std::chrono::steady_clock;

   StopWatch();

   ~StopWatch();
};
```

and it works pretty simple: in the constructor the time measurement start, the destructor stops the measurement and prints the past time into `std::cout`. Hence to measure the time of a scope one simply has to initiate one instance of `StopWatch` with the desired precision as template argument.

C.3 Math

Since the CUDA API is pure C, several classes are missing that are indispensable for physical computations. Hence fi*n*ael includes some classes to meet the demands.[1]

The very first are complex numbers, which are not a native type in C or C++. Its interface is

Complex.h

```
template<typename T>
class Complex
{
public:
   using value_type = T;

   CUDA_HOST_DEVICE Complex( T realIn, T imagIn );

   explicit CUDA_HOST_DEVICE Complex( T realIn );

   CUDA_HOST_DEVICE Complex();

   CUDA_HOST_DEVICE constexpr T real() const;

   CUDA_HOST_DEVICE constexpr T imag() const;

   CUDA_HOST_DEVICE Complex& operator+=( const Complex& rhs );

   CUDA_HOST_DEVICE Complex& operator+=( const T rhs );

   CUDA_HOST_DEVICE Complex& operator-=( const Complex& rhs );

   CUDA_HOST_DEVICE Complex& operator-=( const T rhs );

   CUDA_HOST_DEVICE Complex& operator*=( const Complex& rhs );

   CUDA_HOST_DEVICE Complex& operator*=( const T rhs );

   CUDA_HOST_DEVICE Complex& operator/=( const Complex &rhs );

   CUDA_HOST_DEVICE Complex& operator/=( const T rhs );

   CUDA_HOST_DEVICE Complex operator-() const;

   CUDA_HOST_DEVICE friend Complex operator+( Complex lhs, const T rhs);

   CUDA_HOST_DEVICE friend Complex operator+( const T lhs, Complex rhs);

   CUDA_HOST_DEVICE friend Complex operator-( Complex lhs, const T rhs);

   CUDA_HOST_DEVICE friend Complex operator-( const T lhs, const Complex& rhs);
```

[1] We recognize that since we wrote most of the mathematical classes the thrust library [287], which is shipped together with the CUDA runtime environment, implemented many of them very similar. However, as we implemented them they were still missing or lacked features that we decided to be vital. This may have changed in the meantime and in the long term it is likely that a change to thrust will be beneficial.

```
    CUDA_HOST_DEVICE friend Complex operator*( Complex lhs, const T rhs);

    CUDA_HOST_DEVICE friend Complex operator*( const T lhs, Complex rhs);

    CUDA_HOST_DEVICE friend Complex operator/( Complex lhs, const T rhs);

    CUDA_HOST_DEVICE friend Complex operator/( const T lhs, const Complex& rhs);

    CUDA_HOST_DEVICE bool operator==( const Complex& rhs ) const;

    CUDA_HOST_DEVICE bool operator!=( const Complex& rhs ) const;

    CUDA_HOST_DEVICE T norm() const;

    CUDA_HOST_DEVICE T abs() const;

    CUDA_HOST_DEVICE bool isZero() const;

    CUDA_HOST_DEVICE T arg() const;

    CUDA_HOST_DEVICE Complex conj() const;

    CUDA_HOST_DEVICE Complex exp() const;

    CUDA_HOST_DEVICE Complex log() const;

    CUDA_HOST_DEVICE Complex<T> pow( T exponent ) const;

    CUDA_HOST_DEVICE Complex<T> pow( const Complex<T>& exponent ) const;

    CUDA_HOST_DEVICE Complex<T> sin() const;

    CUDA_HOST_DEVICE Complex<T> cos() const;
};
```

Additionally all functions have free counterparts to offer a more natural use of `Complex` instances, for example `log( complex )` instead of `complex.log()`. Note that we define the arithmetic operators as friend functions (which are declared outside the class) to allow implicit type conversions (see Item 46 in [288]), such that for example a `Complex<float>` can be multiplied by an `int` without explicit conversion, which makes long physical formulas much better manageable. To our knowledge this is currently not possible with `thrust::complex` or even `std::complex`.

Second we offer the classes `Vector` and `Matrix`, which represent mathematical vectors and matrices. Because the interfaces are of similar length as the one of `Complex` we will omit them here and refer to the actual code [285] for brevity. They again use the trick to define the arithmetic operators as friend functions inside the classes to allow type conversions. Further the `Vector` class offers a natural scalar product by overloading the `operator*`. Similar the `Matrix` class overloads the `operator*` to allow natural matrix-matrix multiplication as well as matrix-vector and vector-matrix multiplications with `Vector`. Note that both classes are stack objects in contrast to other libraries

that are specialized for linear algebra, which use heap objects to handle very large vectors and matrices. Hence `Vector` and `Matrix` are mainly designed to simplify the calculations for example for the DGLAP evolution (see sections 1.5 and C.7).

To allow the computation of a DGLAP evolution several functions have to be implemented, defined in equations (B.6), which we named

floratosFunctions.h

```
template<unsigned int index, template<typename> class Complex, typename T>
CUDA_HOST_DEVICE Complex<T> sFloratos( const Complex<T>& argument );

template<unsigned int index, Sign sign, template<typename> class Complex, typename T>
CUDA_HOST_DEVICE inline Complex<T> sPrimeFloratos( const Complex<T>& argument );

template<Sign sign, template<typename> class Complex, typename T>
CUDA_HOST_DEVICE Complex<T> sTildeFloratos( const Complex<T>& argument );
```

as at least to our knowledge the first two are defined in [113] for the first time. Still we look out for more descriptive names and would be glad about reasonable suggestions. The template parameter `Sign` is a simple enum defined by

definitions.h

```
enum class Sign {PLUS, MINUS};
```

Further we implemented several loose functions

generalFunctions.h

```
template<Sign sign, typename T>
CUDA_HOST_DEVICE inline T addOrSubstract( const T& value1, const T& value2 );

template<typename T, unsigned int N>
struct Pow
{
   CUDA_HOST_DEVICE static constexpr T pow( const T& base );
};

template<typename T>
struct Pow<T, 1>
{
   CUDA_HOST_DEVICE static constexpr T pow( const T& base );
};

template<typename T>
struct Pow<T, 0>
{
   CUDA_HOST_DEVICE static constexpr T pow( const T& /* base */ );
};
```

```
template<typename T>
CUDA_HOST_DEVICE T pow( const T& base, const unsigned int exponent);

template<typename T>
CUDA_HOST_DEVICE T pow( const T& base, const int exponent );

template<template<typename> class Complex, typename T>
CUDA_HOST_DEVICE inline Complex<T> sinpi( const Complex<T>& complex );

template<template<typename> class Complex, typename T>
CUDA_HOST_DEVICE inline Complex<T> cospi( const Complex<T>& complex );
```

From these the struct `Pow` is interesting for integer exponents that are known at compile time, as it can make extensive use of optimizations, especially when the compiler flag 'fast-math' is used. The functions `sinpi` and `cospi` calculate the sine and cosine of pi times their argument and use specialized functions of the CUDA API if compiled with the nvcc compiler.

Finally we offer several functions related to the gamma function. While omitting the gamma function itself we implemented its logarithm because it is less prone to overflows as a functor

LogGamma.h

```
class LogGamma
{
public:
   static constexpr int approximationLimit = 15;

   template<template<typename> class Complex, typename T>
   CUDA_HOST_DEVICE Complex<T> operator()( const Complex<T>& arg ) const;

   template<typename T>
   CUDA_HOST_DEVICE T operator()( const T& arg ) const;

   template<template<typename> class Complex, typename T>
   CUDA_HOST_DEVICE Complex<T> calcIgnoringLogBranch( const Complex<T>& arg ) const;
};
```

The `approximationLimit` defines the real part of the argument that has to be reached to use a series expansion. For smaller real parts either a reflection formula or a calculation by recurrence takes place, depending on the actual argument and the type `T`, which limits the use of the reflection formula for large imaginary parts. Additional to the parenthesis operators `LogGamma` offers also a possibly faster function which calculates not necessarily the principal value of the logarithm. However, the speed improvement should be negligible in most circumstances. In the very same manner we implemented the polygamma function in a very generic way

PolyGamma.h

```
template<unsigned int derivative>
class PolyGamma
{
public:
   static constexpr int approximationLimit = 10;

   template<template<typename> class Complex, typename T>
   CUDA_HOST_DEVICE Complex<T> operator()( const Complex<T>& argument ) const;

   template<typename T>
   CUDA_HOST_DEVICE T operator()( const T& argument ) const;
};
```

So far it is implemented for `derivative` $\in \{0, 1, 2\}$. If higher derivatives are needed it is easily extendable as the algorithm is independent of it, but only certain details have to be provided in the helper class `PolyGammaDetailsProvider`. Eventually we implemented the Beta function again in the very same manner

Beta.h

```
class Beta
{
public:
   template<template<typename> class Complex, typename T>
   CUDA_HOST_DEVICE Complex<T> operator()( const Complex<T>& arg1, const Complex<T>& arg2
   ↪  ) const;

   template<typename T>
   CUDA_HOST_DEVICE T operator()( T arg1, T arg2 ) const;
};
```

C.4 Interpolation

The basis of the interpolation of parton distribution functions presented in chapter 5 is a `Spline` class which in turn is used to define a `MellinSpline` class. Both can be calculated using a natural spline (see equation (5.9)) or being fixed at the limits of its domain (see equations (5.10)). This is controlled by

spline_settings.h

```
enum class SplineType {NATURAL, FIXED};
```

The public interface of the spline class is given by

Spline.h

```
template<typename T>
class Spline
{
public:
   template<class PointIt, class Functor>
   Spline( PointIt firstPoint, PointIt lastPoint, const Functor& function );

   template<class PointIt, class Functor>
   Spline( PointIt firstPoint, PointIt lastPoint, const Functor& function, const
   ↪  SplineType type );

   explicit Spline( const MellinSpline<T>& mellinSpline );

   template<class PointIt, class Functor>
   void set( PointIt firstPoint, PointIt lastPoint, const Functor& function, const
   ↪  SplineType type );

   template<class PointIt, class Functor>
   void set( PointIt firstPoint, PointIt lastPoint, const Functor& function );

   template<class Functor>
   void set( const Functor& function, const SplineType type );

   template<class Functor>
   void set( const Functor& function );

   void set ( const MellinSpline<T>& mellinSpline );

   std::size_t numberPoints() const;

   T getMinimalPoint() const;

   T getMaximalPoint() const;

   T getPointAt( const std::size_t index ) const;

   bool isInRange( const T& point ) const;

   SplineType getType() const;

   SplinePiece<T> getPiece( const std::size_t index ) const;

   T operator()( const T& point ) const;
};
```

Although we were mostly interested in the Mellin transform of the splines, we made `Spline` a fully functional functor. The member functions are self-explanatory, except for `getPiece` which returns the cubic interpolation `SplinePiece`, which is again a functor itself but with public data members called `constant`, `linear`, `square` and `cubic`. Using the `Spline` class we define the `MellinSpline` with the following public interface

MellinSpline.h

```
template<typename T>
class MellinSpline
{
public:
   static constexpr std::size_t coefficientsPerPoint = 4;

   template<class PointIt, class Functor>
   MellinSpline( PointIt firstPoint, PointIt lastPoint, const Functor& function );

   template<class PointIt, class Functor>
   MellinSpline( PointIt firstPoint, PointIt lastPoint, const Functor& function, const
   ↪  SplineType type );

   explicit MellinSpline( const Spline<T>& spline );

   template<class PointIt, class Functor>
   void set( PointIt firstPoint, PointIt lastPoint, const Functor& function );

   template<class PointIt, class Functor>
   void set( PointIt firstPoint, PointIt lastPoint, const Functor& function, const
   ↪  SplineType type );

   template<class Functor>
   void set( const Functor& function, const SplineType type );

   template<class Functor>
   void set( const Functor& function );

   void set( const Spline<T>& spline );

   SplineType getType() const;

   std::size_t numberPoints() const;

   T getMinimalPoint() const;

   T getMaximalPoint() const;

   T getPointAt( const std::size_t index ) const;

   std::vector<T> getCoefficientsAt( const std::size_t index ) const;

   template<template<typename> class Complex>
   Complex<T> operator()( const Complex<T>& mellinVariable ) const;
};
```

which is obviously very similar to the interface of `Spline`. However, the functor now takes a complex argument. Further the `getCoefficientsAt` member function returns the coefficients that define the `MellinSpline` at a fixed index, see equation (5.16). Both classes are at the current state only capable of cubic splines, which may be generalized in the future. The constant `coefficientsPerPoint` does reflect just this fact. However, with these classes at hand it is trivial to transfer the needed coefficients to the device and use the splines at explained in chapter 5.

C.5 Pseudo random number generators

As explained in more detail in section 2.4.2 the generation of random numbers on a computer is a delicate task. As we argued there we limit ourselves to pseudo random number generators. In particular we implemented a generic multiplicative linear congruential generator (MLCG) by

Mlcg.h

```
template<std::size_t mod, std::size_t mult>
class Mlcg
{
public:
   static constexpr std::size_t modulus = mod;
   static constexpr std::size_t multiplier = mult;

   Mlcg( std::size_t seed, std::size_t warmUp );

   explicit Mlcg( std::size_t seed );

   std::size_t getRandomNumber();
};
```

where the constructor with the `warmUp` argument proceeds accordingly many steps in the series. The member function `getRandomNumber` proceeds another step and returns the respective 'random' integer. We did not add any function that maps the integers into the real interval $(0, 1)$, as it is not intended for direct use, but as component of the PRNG intended by L'Ecuyer [154]

Lecuyer.h

```
template<typename T, std::size_t shuffleTableSize>
class Lecuyer
{
public:
   Lecuyer();

   T getRandomNumber();

private:
   using Mlcg1 = Mlcg<2147483563, 40014>;
   using Mlcg2 = Mlcg<2147483399, 40692>;
   static constexpr std::size_t mlcgSeed = 1;
   static constexpr std::size_t mlcgWarmup = 23;
   static constexpr T inverseModulusMlcg1 = 1 / static_cast<T>( Mlcg1::modulus );
   static constexpr std::size_t maxMlcg1 = Mlcg1::modulus - 1u;
   static const T maxOutput;
};
```

which combines two MLCGs. Our implementation uses a fixed seed and warm up length for both MLCGs. Especially for single precision we guard the output by `maxOutput`, which is the largest number smaller than one that is representable for the respective precision. Additionally we added a shuffle board as intended in [148], which is fed by the instance of `Mlcg1`. Note that `Lecuyer` is the only PRNG offered by fi*n*ael that does not need any third party software.

Further we implemented an interface for the PRNG of the cuRAND library [289]

CurandWrapper.h

```
template<typename T>
class CurandWrapper
{
public:
#ifdef __CUDACC__
   CUDA_HOST CurandWrapper( void );

   CUDA_HOST ~CurandWrapper( void );

   CUDA_HOST cuTool::SmartArray<T, CudaHostDeviceType::DEVICE> getRandomNumbers( const
   ↪  unsigned int numberRandomNumbers );
#endif // __CUDACC__
};
```

which generates the random numbers directly on the device and is also only useful for this particular situation. Currently the `CurandWrapper` uses the cuRAND Merseinne Twister, which can however be changed in one line of the constructor, if intended. In any case we have to warn about the implementation, because the intrinsic mapping of cuRAND is on the interval $(0, 1]$, which may be fatal in many applications that rely on the assumption that the limits are excluded.

The most important PRNG in fi*n*ael is relying on Philox [156]. Also here we had only to write a wrapper class that uses Philox in the way we intend

PhiloxWrapper.h

```
template<typename T>
class PhiloxWrapper
{
public:
   CUDA_HOST_DEVICE PhiloxWrapper( const unsigned int counter1 = 0xdecafbad, const
   ↪  unsigned int counter2 = 0xf00dcafe, const unsigned int counter3 = 0xdeadbeef,
   ↪  const unsigned int counter4 = 0xbeeff00d );

   CUDA_HOST_DEVICE void setCounter( const unsigned int value, const unsigned int index
   ↪  );

   CUDA_HOST_DEVICE void prepareForKernel();
```

```
    CUDA_HOST_DEVICE T getRandomNumber( const unsigned int value, const unsigned int index
    ↪  = 0u );

    CUDA_HOST_DEVICE T getRandomNumber();
};
```

Several counters of Philox control distinct sequences of 'random' numbers. They can be set by `setCounter` or also before calculating the next number in the `getRandomNumber` function with the appropriate arguments and `index` $\in \{0, \ldots, 3\}$. Note that this is possible on the device as well as the host. However, Philox has a very specific problem when we like to produce independent number for distinct kernel calls. As its whole state is on the stack, it becomes copied to the device in a kernel call, which will not alter the internal state of the host object that will then be copied to the next kernel, producing the very same numbers. To handle this problem we offer the function `prepareForKernel`, which increases a counter that cannot be accessed by `setCounter`. In this way the kernels can be counted and each of them will have access to a unique number sequence. Hence the most simple use is to call `prepareForKernel` before every kernel call that uses a `PhiloxWrapper` instance and then use the `getRandomNumber` function with two arguments to produce a number sequence for every thread by setting the first argument to the thread index in the grid and using the default value for the second argument.

We have to mention one noteworthiness of the way we include the current Philox version 1.09. To compile with as many warning flags as possible we temporarily ignore the 'old-style-cast' and the 'sign-conversion' warnings for the GNU compilers and suppress the 'code_is_unreachable' diagnose of the nvcc compiler. Both issues have been reported and we were told that they can cause no actual harm.

C.6 Vegas

Vegas is the main part of fi*n*ael. It performs an iterative and adaptive Monte-Carlo integration. The corresponding theory is presented in chapter 2. Because of its particular importance we will split the presentation of the interface into two parts. One for the ordinary user and one dedicated for programmers who like to exchange components or add functionality. In both parts we will try to explain the reasons of particular decisions in terms of the code design.

C.6.1 User guide

Before we get to the interface of Vegas, we like to present some enums that are used to specify possible options in Vegas

optionFlags.h

```
enum class PrngFlag {DEFAULT, LECUYER, CURAND, PHILOX};
enum class StrategyFlag {IMPORTANCE, FREE};
enum class ReductionFlag {CLASSIC, REFINED};
enum class VerbosityFlag {QUIET, GENTLE, VERBOSE};
```

The `PrngFlag` determines the PRNG. However, as we will see it is not possible to choose every PRNG in every case. Therefore the value `DEFAULT` will select a reasonable choice depending on other settings. The `StrategyFlag` will determine the adaption strategy that is used between two iteration steps, see figure 2.5. In case of `IMPORTANCE` no stratification will be applied and the adaption is performed by importance sampling. For `FREE` there is always a stratification, however the actual strategy depends on the number of samples that will be requested and may be importance sampling (for few sample points) or stratified sampling (for many sample points). The `ReductionFlag` determines the way the function values calculated at the sample points are reduced to obtain the Monte-Carlo result and the weights for the adaption. If the flag is set to `REFINED` this will only apply, if the strategy will be determined to be stratified. In all other cases the `CLASSIC` reduction is used anyway. Finally the `VerbosityFlag` decides how many informations will be printed to a defined output stream, which is by default the output stream `std::cout`. If set `QUIET` Vegas will print nothing as all, if it is set to `GENTLE` it will print its settings in the beginning of the integration and intermediate results of every iteration. If Vegas is `VERBOSE` it will additionally print out informations on the current increments in every iteration. Note that this will slow down Vegas eventually, as in the current version it performs a second reduction, and is therefore not useful for production runs, where the performance matters. This should not be a too hard restriction, as one is typically interested in the increments only during debugging or while exploring the integrands behavior.

Now it is time for the public interface of Vegas, which is given by

Vegas.h

```
template<unsigned int dim, typename T = double, CudaHostDeviceType hostOrDevice =
↪  CudaHostDeviceType::HOST, PrngFlag prngInputFlag = PrngFlag::DEFAULT>
class Vegas
{
public:
   using type = T;
   using Volume = VolumeType<T, dim>;
   using IntegrandPoint = IntegrandPointType<T, dim>;
   static constexpr PrngFlag prngFlag = PrngSelector<hostOrDevice,
   ↪  prngInputFlag>::prngFlag;
   static constexpr unsigned int dimension = dim;
   static constexpr CudaHostDeviceType hostDeviceType = hostOrDevice;

   CUDA_HOST Vegas();
```

```
    template<class Functor>
    CUDA_HOST Results<T> integrate( const Functor& functor );

    template<class Functor>
    CUDA_HOST Results<T> integrateKeepingOldGrid( const Functor& functor );

    template<class Functor>
    CUDA_HOST Results<T> integrateKeepingOldGridAndResults( const Functor& functor );

    template<class Functor>
    CUDA_HOST Results<T> integrateKeepingAllSettings( const Functor& functor );

    CUDA_HOST void setNumberSamples( unsigned int numberSamples );

    CUDA_HOST void setNumberIterations( unsigned int numberIterations );

    CUDA_HOST void setStrategy( StrategyFlag strategy );

    CUDA_HOST void setAdaptionRate( T adaptionRate );

    CUDA_HOST void setReduction( ReductionFlag reduction );

    CUDA_HOST void setBlockDim( unsigned int blockDim );

    CUDA_HOST void setVerbosity( VerbosityFlag verbosity );

    CUDA_HOST void setOutputStream( std::ostream& stream );

    CUDA_HOST void setPrintPrecision( T precision );
};
```

To initiate an instance of Vegas one has to decide several arguments that cannot be changed at runtime: its template parameters. The first parameter `dim` is the dimension of the integrand, the second parameter is the floating point type. For both it seems to be nonsense to change them while one wants to keep the rest of the internal state of Vegas. Further the dimension determines the size of several arrays, such that in this way exactly the needed amount of memory is used. The floating point type has to be a template parameter, as we perform several specializations based on this type. The third parameter determines whether the integration will take place on the host or the device. We chose `HOST` as default value, such that a version that does not specify the arguments explicitly will at least compile on every system. The final forth parameter determines the PRNG. It is given to the `PrngSelector`, which decides depending on the `CudaHostDeviceType` and the flag which PRNG will be taken. This disentangles Vegas from the PRNG such, that only one PRNG instance will be an actual member of Vegas and it is also not necessary to provide a functioning PRNG for every choice, which is for example impossible in a machine without CUDA in case of the cuRAND PRNG. The current defaults are `LECUYER` for the host and `PHILOX` for the device. The former being chosen because it will work whether the external Philox is present or not. For the device the benefits of Philox are so large, that we decided for it nevertheless.

Further we want to discourage the use of `CURAND`, as it might hit exactly the limit of the integration volume, which is often catastrophic. It is still provided only for testing reasons.

All other settings of Vegas can be changed by the user at runtime by the respective member functions

`setNumberSamples`: Sets the requested number of samples. Note that for a stratified strategy (`StrategyFlag::FREE`) the actual number of samples will typically be a bit smaller to accomplish the alignment of increments and bins. Then the given number will be the upper limit of the number of samples. The default value is 10^4.

`setNumberIterations`: Sets the number of iterations that are performed for a single integration. The default is 5.

`setStrategy`: Sets the requested strategy. However, the final strategy will be determined by the algorithms depicted in figure 2.5. The default value is `FREE`.

`setAdaptionRate`: Sets the variable α defined in equation (2.31). The most reasonable region is $\alpha \in [1, 2]$. For $\alpha = 0$ the adaption is disabled. The default value is $\alpha = 1.5$.

`setReduction`: Sets the reduction mode. The default value is `CLASSIC`. The flag `REFINED` will only apply if also the strategy is determined to be stratified sampling.

`setBlockDim`: Sets the number of threads in one block. The default value is 32, which is the warp size of all currently available GPUs. The block dimension can influence the performance of a GPU significantly. It is therefore recommended to test several values to find the optimum for a given integrand. These should however always be a multiple of the warp size. The maximum for GPUs of compute capability of 3.0 or higher is 1024, for older GPUs it is 512.

`setVerbosity`: Determines the amount of diagnostic output that will be printed by Vegas during the integration. More details have already been provided after the listing containing the different option flags. The default value is `GENTLE`.

`setOutputStream`: Determines the output stream used by Vegas to print the diagnostic output. The default is `std::cout`.

`setPrintPrecision`: Sets the number of digits used for floating point numbers in the diagnostic output. The default is 5.

The remaining public member functions are designated for the actual integration. In the original version of Lepage they were controlled by an additional argument. We tried to make the differences explicit by the function names:

`integrate` will reset the internal state and perform a new integration.

`integrateKeepingOldGrid` will reset the results of previous integrations, but keep the increments. This is especially useful if one uses an initial integration with few samples to determine increments that suit the integrand well and then perform an integration with many samples to get the result.

`integrateKeepingOldGridAndResults` will additionally keep the results, which will be used in the cumulative integral estimate to determine the final result, see equations (2.6) as well as (2.8).

`integrateKeepingAllSettings` will ignore also changes to the settings made before the call of the function except for the number of iterations.

The class `Results` which is returned by all these functions is a simple struct containing the estimate, the variance and the χ^2 that have been calculated

Results.h

```
template<typename T>
struct Results
{
public:
   T integral;
   T standardDeviation;
   T chiSquaredPerIteration;

   CUDA_HOST Results( T integralIn, T standardDeviationIn, T chiSquaredPerIterationIn );
};
```

The final ingredient for a successful use of the fi*n*ael implementation of Vegas are three requirements that the functor class of the integrand has to meet:

(1) It has to be functor, id est it has to overload the parenthesis operator. To use it on the device the parenthesis operator has to be declared as being `CUDA_DEVICE`, or better `CUDA_HOST_DEVICE`. The same is true for all functions that are subsequently called by the parenthesis operator. The parenthesis operator has to be declared constant.

(2) The argument of the parenthesis operator has to be `IntegrandPointType`, which is in the current implementation equivalent to `Array<T, dim>`.

(3) It has to publicly provide the limits of the rectangular integration volume via a variable `volume`, which have to be of type `VolumeType`.

The types necessary for the latter two points are provided in the file

integrandTypes.h

```
template<typename T, unsigned int dim>
using VolumeType = cuTool::Array<cuTool::Pair<T,T>, dim>;

template<typename T, unsigned int dim>
using IntegrandPointType = cuTool::Array<T, dim>;
```

which is guaranteed to be included by Vegas. The types can also be deduced from Vegas directly (see listing 'Vegas.h').

C.6.1.1 Example: integration of a d dimensional Gaussian

To demonstrate how to use Vegas we integrate a simple Gaussian. The respective program could look like this:

Integrate d dimensional Gaussian

```
#include <iostream>

#include "pathToFinael/finael/tools/DeviceSelector.h"
#include "pathToFinael/finael/math/constants.h"
#include "pathToFinael/finael/vegas/Vegas.h"

using namespace finael;
using namespace finael::cuVegas;


template<unsigned int dim, typename T> class Gauss
{
public:
   VolumeType<T, dim> volume;
   using Point = IntegrandPointType<T, dim>;

   CUDA_HOST Gauss()
      : expectedResult( pow( cuMath::CONST<T>::PI, static_cast<T>( dim ) / 2 ) )
   {
      for (auto dimIdx = 0u; dimIdx < dim; ++dimIdx)
      {
         volume[dimIdx].first = 0;
         volume[dimIdx].second = 1;
      }
   }

   CUDA_HOST_DEVICE T operator()( const Point& x ) const
   {
      T result( 1 );
      for (auto dimIdx = 0u; dimIdx < dim; ++dimIdx)
      {
         T den = 1/( 1 - x[dimIdx] );
         T t = x[dimIdx]*den;
         result *= 2 * exp( - (t * t) ) * den * den;
      }
```

```
        return result / expectedResult;
    }

private:
    const T expectedResult;
};


int main()
{
    finael::cuTool::DeviceSelector::setFirstAvailable();
    using floatType = double;
    constexpr auto dim = 2u;
    constexpr CudaHostDeviceType deviceType = CudaHostDeviceType::HOST;

    Gauss<dim, floatType> gauss;
    Vegas<dim, floatType, deviceType, PrngFlag::PHILOX> vegas;

    vegas.setNumberSamples( 10000000u );
    vegas.setReduction( ReductionFlag::REFINED );
    vegas.setBlockDim( 512u );

    auto results = vegas.integrate( gauss );
    std::cout << "Estimate = " << results.integral << " +/- " << results.standardDeviation
    ↪  << std::endl;
}
```

We include the `DeviceSelector` just in case we liked to use a GPU. Further we include some mathematical constants including π, because we want to normalize the integration to its known result, which is $\pi^{d/2}$ as we will use the symmetry of the Gaussian to reduce the integration volume. Finally we include Vegas itself. Then we define the functor we like to integrate: `Gauss`. It declares the `volume` of the integration publicly and fills its data in the constructor on the host. The parenthesis operator can be executed on the host as well as on the device and is declared constant. Its argument type is `IntegrandPointType` of the respective type and dimensionality. The argument is taken by reference to increase speed for dimensions larger than one. In the integrand the volume is stretched from $(0,1)^d$ to $(0,\infty)^d$, applying the respective Jacobian to the result and the Gaussian function value is calculated. The first line in `main` sets the device if a GPU is available. The following three lines set the floating type, dimensionality and whether the integration has to be executed on the host or the device, respectively. Then instances `gauss` and `vegas` are created and some of the settings of the latter are changed. Finally the integration is performed by `vegas.integrate( gauss )` and the results are printed to the screen via `std::cout`.

Note that all members of `Gauss` are also transferred to the device, as long as they are on the stack, here `expectedResult`. If a large amount of memory is necessary we recommend to use the `SmartArray`, see section C.2, to allocate heap memory or global memory for host or device, respectively. A plain `std::vector` will surely work on the host, but fail on the device.

C.6.2 Programmers guide

We tried to write all code in such a style that it should be readable and easily extendable for every C++ programmer. However, we like to explain one particular component, which is explicitly designed to allow extensions in a native way: the PRNG. It consists of three classes. The first is the `PrngSelector`, which simply works by template specialization, mapping the input flag to the flag that will be used by Vegas internally. This flag can be used to feed the `PrngTypeSetter`, which then in turn offers the actual types needed

PrngTypeSetter.h

```
template<PrngFlag prng>
class PrngTypeSetter {};

template<>
class PrngTypeSetter<PrngFlag::LECUYER>
{
private:
   static constexpr std::size_t shuffleBoardSize = 32;
public:
   template<typename T>
   using PrngType = prng::Lecuyer<T,shuffleBoardSize>;

   template<unsigned int dim, typename T, CudaHostDeviceType hostDeviceType>
   using KernelPrngType = KernelLecuyer<dim, T, shuffleBoardSize, hostDeviceType>;
};

template<>
class PrngTypeSetter<PrngFlag::CURAND>
{
public:
   template<typename T>
   using PrngType = prng::CurandWrapper<T>;

   template<unsigned int dim, typename T, CudaHostDeviceType hostDeviceType>
   using KernelPrngType = KernelCurand<dim, T, hostDeviceType>;
};

template<>
class PrngTypeSetter<PrngFlag::PHILOX>
{
public:
   template<typename T>
   using PrngType = bool; // not relevant, philox has complete functionality already in
   ↪  VegasKernelPhilox

   template<unsigned int dim, typename T, CudaHostDeviceType hostDeviceType>
   using KernelPrngType = KernelPhilox<dim, T, hostDeviceType>;
};
```

The client is the third class

Prng.h

```
template<unsigned int dim, typename T, CudaHostDeviceType hostDeviceType, PrngFlag
↪  prngFlag>
class Prng
{
public:
   typename PrngTypeSetter<prngFlag>::template KernelPrngType<dim, T, hostDeviceType>
   ↪  kernelPrng;

   CUDA_HOST void prepareForIntegration( const unsigned int numberSamples )
   {
      kernelPrng.prepareForIntegration( prng, numberSamples );
   }

private:
   typename PrngTypeSetter<prngFlag>::template PrngType<T> prng;
};
```

where we can see the purpose of the two types defined in `PrngTypeSetter`. `kernelPrng` is the actual provider of the 'random' number sequence. It will be passed to the kernel calculating the samples. The second class of type `PrngType` can support the `kernelPrng` in a function that is performed before every iteration step. As we see for `Philox` the respective type can be set to a pure dummy type, if it is not needed. For `Lecuyer` it is the actual PRNG, while the `kernelPrng` is an array that stores the numbers and returns them if requested. As an example we present the `KernelPhilox`

KernelPhilox.h

```
template<unsigned int dim, typename T, CudaHostDeviceType hostDeviceType>
class KernelPhilox
{
public:
   CUDA_HOST void prepareForIntegration( const bool, const unsigned int /* numberSamples
   ↪  */ )
   {
      philox.prepareForKernel();
   }

   CUDA_HOST_DEVICE T getRandomNumber( const unsigned int index )
   {
      return philox.getRandomNumber( index );
   }

private:
   prng::PhiloxWrapper<T> philox;
};
```

where we can identify the necessary handling as explained in section C.5.

This means to add a new RNG is quite simple:

(1) Add a flag in the options.

(2) Modify the `PrngSelector` if and only if the default RNG has to be changed.

(3) Add a specialization of `PrngTypeSetter` that defines the types that are used for the new RNG.

(4) Write at least a class for the `kernelPrng` that respects its implicit interface used in `Prng`. If necessary also a supporter class, using the `PrngType`.

C.7 QCD

fi*n*ael has two distinct parts that are concerned about QCD. The first one is the interpolation of PDFs in terms of cubic splines. This part is for the host only and extensively presented and discussed in chapter 5. The second part is dedicated to be used on the host as well as on the device, allowing perturbative QCD calculations on the GPU. It is still in an early stage of development, however we will highlight some key features that are already implemented while indicating also the points that should be improved in the future.

Currently one implementation of the strong coupling constant is present, which is the simple series using the Landau pole as parameter.

AlphaS.h

```
template<unsigned short numberFlavours, PQcdOrder order, typename T>
class AlphaS
{
public:
   using ScaleType = T;
   static constexpr unsigned short NUMBERFLAVOURS = numberFlavours;
   static constexpr PQcdOrder ORDER = order;

   CUDA_HOST_DEVICE T operator()( T scale ) const;
};
```

We are also still set to use a fixed flavour scheme only. As we will argue in a moment this is sufficient for the first project we are planning. It is apparent that we use again enum classes to specify possible options

definitions.h

```
enum class PartonFlavour {G, U, UBAR, D, DBAR, S, SBAR, C, CBAR, B, BBAR, T, TBAR};
enum class FlavourMode {ONE, TWO, THREE, FOUR, FIVE, SIX};
enum class Hadron {PION, KAON, PROTON, ALL};
enum class PQcdOrder {LO, NLO};
```

The very idea of the project is to find a reasonable alternative to the so called b_* prescription, which is used to control the Sudakov form factor for large impact

parameters. The b_* prescription is a smooth cutoff, which leaves the scene with the problematic choice to decide which values of the impact parameter should still be perturbative and which are considered to be nonperturbative. A proposed alternative is the so called contour prescription [290] and has been applied in phenomenological studies for Drell-Yan [291] and SIDIS [237]. It suggests to avoid the Landau pole by making the impact parameter complex. The charm of the contour prescription is due to the fact that its results do not depend on any parameters, because the integrand are typically holomorphic. Still nonperturbative physics has to be taken into account, which typically is done by multiplying an additional Gaussian distribution, whose width has to be determined by comparison to data. While the width is strongly dependent on the cutoff parameter of the b_* prescription [64, 204, 205], the contour prescription does not contain such an ambiguity. Recently it has been discovered that the original contour prescription does produce unphysical results for the Sivers function, but could be fixed by the price of a restriction to the complex contour [9]. However, to perform a fit to deduce the nonperturbative gaussian parameter using the contour prescription it is necessary to evolve PDFs to a complex scale. The evolution has to take place in Mellin space, as only in Mellin space we have a closed analytical form for the evolution, see section 1.5. Hence we implemented the DGLAP evolution such, that the type of the scale can be freely chosen, including to be complex. The two key classes are the container of the PDFs

PartonDistribution.h

```
template<unsigned short numFlavours, typename DistType, typename ScaleType>
class PartonDistribution
{
public:
   static constexpr unsigned short NUMBERFLAVOURS = numFlavours;
   using value_type = DistType;
   using scale_type = ScaleType;
   using Singlet = cuMath::Vector<2, value_type>;
   using NonSingletNonValence = cuMath::Vector<numFlavours - 1, value_type>;

   CUDA_HOST_DEVICE PartonDistribution( const cuTool::Array<value_type, 2*numFlavours +
   ↪  1>& inputData, const scale_type& inputScale );

   template<PartonFlavour flavour>
   CUDA_HOST_DEVICE value_type get() const;

   CUDA_HOST_DEVICE scale_type getScale() const;

   CUDA_HOST_DEVICE void evolve( const EvolutionKernelContainer<value_type>&
   ↪  evolutionKernel, const scale_type& newScale );
};
```

which manages the linear combinations of the distinct parton flavours. It can be evolved by handling the struct `EvolutionKernelContainer` to it, which is exactly what is done by the member function `evolve` of

DglapEvolution.h

```
template<class AlphaS, typename ScalarType>
class DglapEvolution
{
public:
   using ScaleType = typename AlphaS::ScaleType;
   static constexpr unsigned short NUMBERFLAVOURS = AlphaS::NUMBERFLAVOURS;
   static constexpr PQcdOrder ORDER = AlphaS::ORDER;

   template<class Complex>
   CUDA_HOST_DEVICE void evolve( PartonDistribution<NUMBERFLAVOURS, Complex, ScaleType>&
   ↪  dist, const ScaleType& newScale, const Complex& mellinMoment ) const;
};
```

The evolution is driven by the template parameter `AlphaS` and hence is sufficient generic to be used for also for other versions of the running coupling constant. It is currently also limited to a fixed number of flavours, because the evolution to a complex scale does not allow a sensible change of the number of flavours anyway. It does provide evolution in leading and next-to-leading order using the truncated solution, see again section 1.5. It is also using a rich inner structure which should allow for an easy extension to a scheme with a variable number of flavours.

C.8 Tests and Debugging

We have to admit that the tests of fi*n*ael is its weakest point in terms of proper software development and a lot has to be done to cover fi*n*ael as a whole. We use the Boost testing library [284]. The following parts are covered with tests, that can help for future improvements and developments

(1) Most of the math classes are tested extensively. Reference values have been computed with Mathematica.

(2) The interpolation classes have been tested against a few simple examples that have been checked by hand.

(3) The stack `Array` and `Pair` are tested throughout.

(4) All QCD classes have been tested. As reference we took an legacy Fortran code performing the DGLAP evolution that has been modified to be capable to calculate in double precision. This code has been widely used for many years and is said to be trustworthy.

All other classes have been written with extreme care. Especially Vegas and its components has been checked at several stages against the original implementation of Lepage, however we did not deduce expressive tests, as it turns out to be quite difficult

due to the exposed meaning of the used PRNGs. Nevertheless this should be done in the future.

To simplify debugging on the device a macro is used, which is only active if the nvcc compiler defines `__CUDACC_DEBUG__`. It is placed after every interaction with the device (memory allocations, copies, freeing memory and kernel calls). It stops asynchronous execution of the device and the host code to correctly assign CUDA errors and will print errors of the CUDA environment via `printf` before exiting the whole program. Due to the macro nature it is then possible to track down the exact action that did fail, which hopefully simplifies debugging to a large degree.

Appendix D

Details to calculations in chapter 4

In this chapter we provide details and calculations that have been omitted in chapter 4. In section D.1 we clarify some issues concerned with the particular phase space we choose. In section D.2 we illuminate the calculations with the involved correlators, how to generate them and how to integrate out all but the collinear degree of freedom. In section D.3 we present the computation of the partonic cross sections. In section D.4 the most important part of the calculations is presented in detail: the expansion of the delta distribution. The final section D.5 comments on the regularization of the transverse momentum distribution.

D.1 Phase space identities

In this section we mainly show the transformation from the generic phase space in four dimensions $\mathrm{d}\mathcal{R}_4$ given in equation (4.12) to the description used in [237]. The result has been reported in equation (4.13). For this purpose we explicitly give the momenta of the final state particles in terms of the variables chosen in [237]

$$l_{2,z} = \frac{1}{2}\sqrt{Q^2} \tag{D.1a}$$

$$|l_{2,T}| = xS\sqrt{\frac{1}{Q^2} - \frac{1}{xS}} \tag{D.1b}$$

$$P_{h,z} = \frac{z_f}{2}\frac{1}{\sqrt{Q^2}}\left(q_T^2 - Q^2\right) \tag{D.1c}$$

$$|P_{h,T}| = z_f\sqrt{q_T^2} \tag{D.1d}$$

where we used cylinder coordinates. We recognize that the z component of the lepton momentum is positive by its definition. This gives us an additional factor of 2 by rewriting the negative part in terms of the positive. Second we use that nothing will depend on the azimuthal angle of the measured hadron ϕ_h, which we integrate out

$$\int \mathrm{d}^3 l_2 = 2\int_0^\infty \mathrm{d}l_{2,z}\int_0^\infty \mathrm{d}|l_{2,T}|\,|l_{2,T}|\int_0^{2\pi}\mathrm{d}\phi \tag{D.2a}$$

$$\int \mathrm{d}^3 P_h = 2\pi \int_{-\infty}^{\infty} \mathrm{d}P_{h,z} \int_0^{\infty} \mathrm{d}|P_{h,T}|\,|P_{h,T}| \tag{D.2b}$$

Now the Jacobian is given by

$$\frac{\partial\{l_{2,z}, |l_{2,T}|, P_{h,z}, |P_{h,T}|\}}{\partial\{x, Q^2, z, q_T^2\}} = \frac{(Q^2+q_T^2)zS}{16Q^3 q_T} \frac{\cosh(\psi)}{\sinh(\psi)} \tag{D.3}$$

The final ingredients to find equation (4.13) are the Møller flux factor $4P\cdot l_1 = 2S$ and the energies of the final state particles $E_2 E_h = z\cosh(\psi)(Q^2+q_T^2)/4$.

D.2 Heading for correlator definitions

In this section a few identities which lead to the definitions of the various correlators we encounter in the main chapter 4 will be shown that have been used without explicit mentioning there.

We start with the calculation in equation (4.14), where the invariant amplitude $\mathcal{M}_{qq}$ in figure 4.1 is calculated. First it is implicitly used that the matrix element containing the full final and initial state factorizes. Then we recognize that the states are eigenstates of the momentum operator, which is used to shift the argument of the field operators

$$\begin{aligned}
\int \mathrm{d}^d\eta\, \mathrm{e}^{\mathrm{i}p\cdot\eta} \langle P_X|\, \psi_j^B(\eta)\, |P\rangle &= \int \mathrm{d}^d\eta\, \mathrm{e}^{\mathrm{i}p\cdot\eta} \langle P_X|\, \mathrm{e}^{\mathrm{i}\hat{p}\cdot\eta}\, \psi_j^B(0)\, \mathrm{e}^{-\mathrm{i}\hat{p}\cdot\eta}\, |P\rangle \\
&= \int \mathrm{d}^d\eta\, \mathrm{e}^{\mathrm{i}(p+P_X-P)\cdot\eta} \langle P_X|\psi_j^B(0)|P\rangle \\
&= \langle P_X|\psi_j^B(0)|P\rangle\, (2\pi)^d \delta^d(p-(P-P_X))
\end{aligned} \tag{D.4}$$

The delta distribution is then used to eliminate the p integral. Exactly the same is done for the matrix element of the fragmenting quark.

Now consider the differential cross section in equation (4.15). Omitting the phase space of the final state gluons it is given by

$$\begin{aligned}
\frac{\mathrm{d}\sigma_{qq}}{\mathrm{d}\mathcal{R}_d} \sim \sum\!\!\!\!\!\!\int_{X,Y} & (2\pi)^d \delta^d\Big(P + l_1 - P_X - l_2 - P_h - P_Y - \sum_{m=0}^{n} g_m\Big) \\
& \times \langle P_h, P_Y|\bar{\psi}_i^A(0)|0\rangle\, H_{ij}^{AB}\, \langle P_X|\psi_j^B(0)|P\rangle \\
& \times \langle P|\bar{\psi}_k^C(0)|P_X\rangle\, (\gamma^0 H^\dagger \gamma^0)_{kl}^{CD}\, \langle 0|\psi_l^D(0)|P_h, P_Y\rangle
\end{aligned} \tag{D.5}$$

where we use the shorthand notation defined in equation (4.18) for the phase spaces of the unobserved remnants. To simplify the expression and introduce the correlators

as defined in equations (4.16) and (4.17) we rewrite the matrix element $\langle P_X|\psi_j^B(0)|P\rangle$ again by using the momentum operator

$$\begin{aligned}
&\langle P_X|\psi_j^B(0)|P\rangle\,\delta^d\Big(P+l_1-P_X-l_2-P_h-P_Y-\sum_{m=0}^{n}g_m\Big)\\
&=\int\frac{\mathrm{d}^dp}{(2\pi)^d}(2\pi)^d\delta^d(p-(P-P_X))\,\langle P_X|\psi_j^B(0)|P\rangle\\
&\quad\times\delta^d\Big(p+l_1-l_2-P_h-P_Y-\sum_{m=0}^{n}g_m\Big)\\
&=\int\frac{\mathrm{d}^dp}{(2\pi)^d}\int\mathrm{d}^d\eta\;\mathrm{e}^{\mathrm{i}\eta\cdot(p-(P-P_X))}\,\langle P_X|\psi_j^B(0)|P\rangle\,\delta^d\Big(p+l_1-l_2-P_h-P_Y-\sum_{m=0}^{n}g_m\Big)\\
&=\int\mathrm{d}^dp\int\frac{\mathrm{d}^d\eta}{(2\pi)^d}\;\mathrm{e}^{\mathrm{i}\eta\cdot p}\,\langle P_X|\psi_j^B(\eta)|P\rangle\,\delta^d\Big(p+l_1-l_2-P_h-P_Y-\sum_{m=0}^{n}g_m\Big)
\end{aligned}\tag{D.6}$$

The very same calculation is performed for the matrix element $\langle 0|\psi_l^d(0)|P_h,P_Y\rangle$

$$\begin{aligned}
&\langle 0|\psi_l^D(0)|P_h,P_Y\rangle\,\delta^d\Big(p+l_1-l_2-P_h-P_Y-\sum_{m=0}^{n}g_m\Big)\\
&=\int\mathrm{d}^dp_h\int\frac{\mathrm{d}^d\eta_h}{(2\pi)^d}\;\mathrm{e}^{\mathrm{i}\eta_h\cdot p_h}\,\langle 0|\psi_l^D(\eta_h)|P_h,P_Y\rangle\,\delta^d\Big(p+l_1-l_2-p_h-\sum_{m=0}^{n}g_m\Big)
\end{aligned}\tag{D.7}$$

and we can write the differential cross section as

$$\begin{aligned}
\frac{\mathrm{d}\sigma_{qq}}{\mathrm{d}\mathcal{R}_d}\sim\int\mathrm{d}^dp\int\mathrm{d}^dp_h\,(2\pi)^d\delta^d\Big(p-p_h+l_1-l_2-\sum_{m=0}^{n}g_m\Big)H_{ij}^{AB}(\gamma^0H^\dagger\gamma^0)_{kl}^{CD}\\
\times\sum\!\!\!\!\!\!\int_X\int\frac{\mathrm{d}^d\eta}{(2\pi)^d}\,\mathrm{e}^{\mathrm{i}\eta\cdot p}\,\langle P|\bar\psi_k^C(0)|P_X\rangle\,\langle P_X|\psi_j^B(\eta)|P\rangle\\
\times\sum\!\!\!\!\!\!\int_Y\int\frac{\mathrm{d}^d\eta_h}{(2\pi)^d}\,\mathrm{e}^{\mathrm{i}\eta_h\cdot p_h}\,\langle 0|\psi_l^D(\eta_h)|P_h,P_Y\rangle\,\langle P_h,P_Y|\bar\psi_i^A(0)|0\rangle
\end{aligned}\tag{D.8}$$

At this point we identify the unit operator

$$\sum\!\!\!\!\!\!\int_X|P_X\rangle\,\langle P_X|=1\tag{D.9}$$

Note that this is not possible for the Y remnant, because the state is not complete—the measured hadron h is missing. Further we use that in the matrix elements only color-diagonal terms can contribute

$$\langle\ldots|\psi^A\psi^B|\ldots\rangle=\frac{\delta^{AB}}{N_c}\,\langle\ldots|\psi^C\psi^C|\ldots\rangle\tag{D.10}$$

Otherwise it would be impossible to form a colorless hadron state in the complex conjugated part of the diagrams. Now we can identify the correlators defined in equations (4.16) and (4.17) and finally find

$$\begin{aligned}\frac{\mathrm{d}\sigma_{qq}}{\mathrm{d}\mathcal{R}_d} \sim \int \mathrm{d}^d p \int \mathrm{d}^d p_h \, (2\pi)^d \delta^d\Big(p - p_h + l_1 - l_2 - \sum_{m=0}^{n} g_m\Big) \\ \times \frac{1}{N_c} H_{ij}^{AB} (\gamma^0 H^\dagger \gamma^0)_{kl}^{BA} \Phi_{jk}^q(p,P) \Delta_{li}^q(p_h, P_h)\end{aligned} \tag{D.11}$$

which is identical to equation (4.15).

The last step is the integration of the correlators, which is possible once the collinear expansion has been applied for all other parts of the cross section. Writing the momentum integral explicitly in its components

$$\int \mathrm{d}^d p = \int \mathrm{d}(p \cdot n_-) \int \mathrm{d}(p \cdot n_+) \int \mathrm{d}^{d-2} \vec{p}_\perp \tag{D.12}$$

only the correlator depends on $(p \cdot n_+) = p_-$ and $\vec{p}_\perp$

$$\begin{aligned}&\int \mathrm{d}p_- \int \mathrm{d}^{d-2}\vec{p}_\perp \Phi_{ij}^q(p,P) \\ &= \int \mathrm{d}p_- \int \mathrm{d}^{d-2}\vec{p}_\perp \int \frac{\mathrm{d}^d \eta}{(2\pi)^d} \, \mathrm{e}^{\mathrm{i}(p_+\eta_- + p_-\eta_+ - \vec{p}_\perp \cdot \vec{\eta}_\perp)} \, \langle P | \bar{\psi}_j^A(0) \psi_i^A(\eta) | P \rangle \\ &= \int \frac{\mathrm{d}^d \eta}{(2\pi)^d} (2\pi)^{d-1} \delta(\eta_+) \delta^{d-2}(\vec{\eta}_\perp) \, \mathrm{e}^{\mathrm{i}p_+\eta_-} \, \langle P | \bar{\psi}_j^A(0) \psi_i^A(\eta) | P \rangle \\ &= \int \frac{\mathrm{d}\eta_-}{2\pi} \, \mathrm{e}^{\mathrm{i}p_+\eta_-} \, \langle P | \bar{\psi}_j^A(0) \psi_i^A(\eta_- n_-) | P \rangle \\ &= \int \frac{\mathrm{d}\eta}{2\pi} \, \mathrm{e}^{\mathrm{i}\xi(P\cdot n_-)\eta} \, \langle P | \bar{\psi}_j^A(0) \psi_i^A(\eta n_-) | P \rangle\end{aligned} \tag{D.13}$$

Again the very same calculation is performed for the fragmentations correlation function

$$\begin{aligned}&\int \mathrm{d}(p_h \cdot m) \int \mathrm{d}^{d-2} \vec{p}_{h\perp} \, \Delta_{ij}^q(p_h, P_h) \\ &= \int \frac{\mathrm{d}\eta}{2\pi} \, \mathrm{e}^{\mathrm{i}\frac{1}{\zeta}(P_h \cdot \overline{m})\eta} \, \langle 0 | \psi_i^A(\eta \overline{m}) | P_h, P_Y \rangle \, \langle P_h, P_Y | \bar{\psi}_j^A(0) | 0 \rangle\end{aligned} \tag{D.14}$$

After inclusion of the Wilson lines we find the final definitions of the collinear correlation functions in equations (4.20).

Note that the remaining momentum integrals are rewritten in terms of the momentum fractions ξ and ζ defined in equations (4.19)

$$\int_0^{P\cdot n_-} \mathrm{d}(p \cdot n_-) = \frac{Q}{\sqrt{2}x} \int_0^1 \mathrm{d}\xi \tag{D.15a}$$

$$\int\limits_{P_h\cdot\overline{m}}^{\infty} \mathrm{d}(p_h\cdot\overline{m}) = \frac{zQ}{\sqrt{2}}\int\limits_0^1 \frac{\mathrm{d}\zeta}{\zeta^2} \tag{D.15b}$$

For the gluon correlators very similar steps are necessary. We start with the gluon fragmentation correlator, which is used in section 4.4. In order to perform the calculation in equation (4.40) the matrix element creating a quark is treated as before but the matrix element where a gluon creates the final hadronic state needs some special attention. In order to find a gauge invariant expression we insert a multiplicative one and perform an integration by parts, assuming the fields to vanish at the edges of space-time

$$\begin{aligned}
&\int \mathrm{d}^d\eta_h\ \mathrm{e}^{-\mathrm{i}p_h\cdot\eta_h}\,\langle P_Y, P_h|G^a_\mu(\eta_h)|0\rangle \\
&= \frac{1}{p_h\cdot\overline{m}}\int \mathrm{d}^d\eta_h\,\overline{m}_\nu\Big(\mathrm{i}\partial^\nu_{\eta_h}\,\mathrm{e}^{-\mathrm{i}\eta_h\cdot p_h}\Big)\,\langle P_Y, P_h|G^a_\mu(\eta_h)|0\rangle \\
&= \frac{-\mathrm{i}}{p_h\cdot\overline{m}}\int \mathrm{d}^d\eta_h\ \mathrm{e}^{-\mathrm{i}p_h\cdot\eta_h}\,\langle P_Y, P_h|\big[(\overline{m}\cdot\partial)G^a_\mu\big](\eta_h)|0\rangle
\end{aligned} \tag{D.16}$$

The field $\big[(\overline{m}\cdot\partial)G^a_\mu\big]$ is the physical field that has to be present in the correlator, in particular it is the field strength tensor in axial gauge $\overline{m}\cdot G^a = 0$

$$(\overline{m}\cdot\partial)G^a_\mu = \overline{m}^\nu\big[\partial_\nu G^a_\mu - \partial_\mu G^a_\nu + g_s f^{abc}G^b_\nu G^c_\mu\big] = \overline{m}^\nu G^a_{\nu\mu} \tag{D.17}$$

where g_s is the strong coupling parameter in the QCD Lagrangian. Because the field strength tensor is gauge invariant also the correlator is gauge invariant. We are therefore free to choose any gauge in the following calculations. Note that the choice of the direction $\overline{m}^\mu$ is arbitrary, but is the only component that will survive the collinear expansion and is therefore our choice from the beginning. After the manipulation shown above the correlator can easily be adjusted by use of the momentum operator similar to equation (D.4) yielding the result in equation (4.40).

The calculation to find the differential cross section in equation (4.41) is completely analogous as for the qq channel. The only difference is the used formula to state that the matrix element is color diagonal

$$\langle|\big[(\overline{m}\cdot\partial)G^a_\mu\big]\big[(\overline{m}\cdot\partial)G^b_\mu\big]|\rangle = \frac{\delta^{ab}}{N_c^2-1} \tag{D.18}$$

because the adjoint instead of the fundamental representation is used. The integration of the non-collinear degrees of freedom is as before.

Now we turn towards the gluon PDF correlator which is used in section 4.5. The calculation in equation (4.55) is the same as before. The only difference is that the relevant direction is given by $p^+ = p\cdot n_-$ instead of $p_h\cdot\overline{m}$. Accordingly during the calculation we use the $G^+ = 0$ gauge until we find again a gauge invariant expression,

namely the field strength tensor as before

$$\int \mathrm{d}^d\eta\ \mathrm{e}^{\mathrm{i}p\cdot\eta}\,\langle P_X|G^a_\mu(\eta)|P\rangle = \frac{\mathrm{i}}{p^+}\int \mathrm{d}^d\eta\ \mathrm{e}^{\mathrm{i}p\cdot\eta}\,\langle P_X|\big[\partial^+ G^a_\mu\big](\eta)|P\rangle \tag{D.19}$$

Again the momentum operator is used to shift the correlator and find the last line in equation (4.55). The calculation to obtain equation (4.56) can be performed with the steps we already presented for the other channels.

D.3 Partonic cross sections

In this section we will calculate the partonic cross sections of the SIDIS process up to next-to-leading order explicitly. In order to do so, we will use the TRACER package [292] to evaluate Dirac traces. For completeness we collect here the necessary scalar products that has to be given to the package

$$p\cdot p_h = \xi\frac{z}{\zeta}(P\cdot q) = \frac{\xi}{x}\frac{z}{\zeta}\frac{Q^2}{2} \tag{D.20a}$$

$$p\cdot q = \frac{\xi}{x}\frac{Q^2}{2} \tag{D.20b}$$

$$p_h\cdot q = \frac{1}{2}\frac{z}{\zeta}(q_T^2 - Q^2) \tag{D.20c}$$

$$l_1\cdot p = \frac{\xi}{x}\frac{Q^2}{4}(\cosh(\psi)+1) \tag{D.20d}$$

$$l_1\cdot p_h = \frac{z}{\zeta}\frac{1}{4}\big[\cosh(\psi)\big(Q^2+q_T^2\big) + \big(q_T^2 - Q^2\big) - 2q_T Q\sinh(\psi)\cos(\phi)\big] \tag{D.20e}$$

$$l_2\cdot p = x(l_1 - q)\cdot P = \frac{\xi}{x}\frac{Q^2}{4}(\cosh(\psi)-1) \tag{D.20f}$$

$$l_2\cdot p_h = \frac{z}{\zeta}\frac{1}{4}\big[\cosh(\psi)\big(Q^2+q_T^2\big) - \big(q_T^2 - Q^2\big) - 2q_T Q\sinh(\psi)\cos(\phi)\big] \tag{D.20g}$$

$$l_1\cdot l_2 = -l_1\cdot q = l_2\cdot q = \frac{Q^2}{2} \tag{D.20h}$$

D.3.1 Quark PDF and quark FF—the qq channel

For this configuration we have to consider two distinct cases: virtual diagrams including the leading-order diagram with no gluon in the final state $n = 0$ and real diagrams with (at next-to-leading order) one gluon in the final state $n = 1$.

D.3.1.1 Virtual diagrams

We start with the virtual diagrams, containing also the leading order α_s^0. The invariant amplitude of the partonic process is shown in figure D.1. It means we consider the

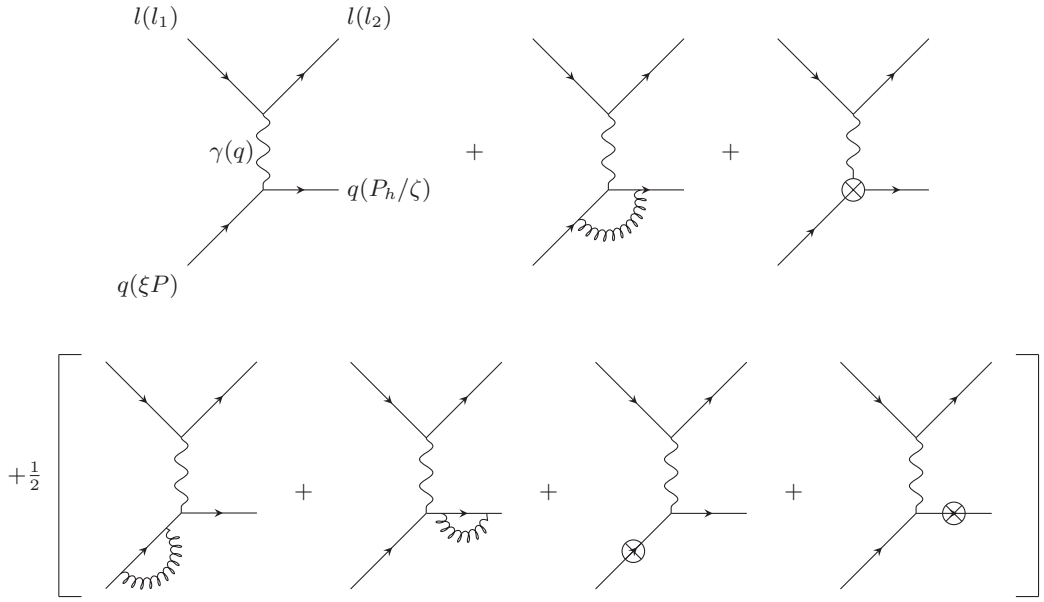


Figure D.1: Partonic invariant amplitude of SIDIS process with a quark in the initial state and a hadronizing quark in the final state. The momenta are denoted in the first diagram only, but are identical in all others. The cross symbols indicate the renormalization counter terms for the vertex and the quark propagator.

summand with $n = 0$ in equation (4.22)

$$\frac{\mathrm{d}\sigma_{qq}^{n=0}}{\mathrm{d}\mathcal{R}_d} = \frac{1}{2}\int_0^1 \frac{\mathrm{d}\xi}{\xi}\int_0^1 \frac{\mathrm{d}\zeta}{\zeta}\frac{1}{\zeta} f_q(\xi) D_q(\zeta) \\ \times (2\pi)^d \delta^d\left(l_1 - l_2 + \xi P - \frac{1}{\zeta}P_h\right)\frac{1}{N_c}\,\mathrm{tr}\big(H \not{p} \gamma^0 H^\dagger \gamma^0 \not{p}_h\big) \tag{D.21}$$

where we used the collinear expansion via

$$n_+^\mu = \frac{\sqrt{2}}{Q}\frac{x}{\xi}p^\mu \tag{D.22a}$$

$$m^\mu = \frac{\sqrt{2}}{Q}\frac{\zeta}{z}p_h^\mu \tag{D.22b}$$

to replace the light-cone variables with the appropriate particle momenta. In the next step we decompose the delta distribution

$$\delta^d\left(l_1 - l_2 + \xi P - \frac{1}{\zeta}P_h\right) = \frac{2}{Q^2}\frac{\zeta^2 x}{z}\delta(\xi - x)\delta(\zeta - z)\delta^{(d-2)}\begin{pmatrix} q_T\cos(\phi_h) \\ q_T\sin(\phi_h) \\ \vdots \end{pmatrix} \tag{D.23}$$

Note that we used the angular ϕ_h that is ignored throughout this calculation otherwise (even in the definition of the kinematics, which implied $\phi_h = 0$). Only in this point it is more descriptive, because it shows us, that the distribution will set $q_T = 0$, which we already used for the computation of the '+' direction of the distribution. The transverse delta distribution shows us that the origin will be picked out, which justifies the arbitrary choice of ϕ_h in the rest of the calculation a posteriori. Finally we can rewrite the transverse distribution by

$$\begin{aligned} 1 &\stackrel{!}{=} \int \mathrm{d}^{d-2}q_T\, \delta^{(d-2)} \begin{pmatrix} q_T\cos(\phi_h) \\ q_T\sin(\phi_h) \\ \vdots \end{pmatrix} \\ &= \int \mathrm{d}q_T\, q_T^{d-3}\delta(q_T) \int \mathrm{d}\Omega_{d-2} \\ &= \int \mathrm{d}q_T\, q_T^{d-3}\delta(q_T) \frac{2\pi^{1-\varepsilon}}{\Gamma(1-\varepsilon)} \end{aligned} \tag{D.24}$$

where ε is the regulating parameter that specifies the dimension $d = 4 - 2\varepsilon$. The above equation means in a distributional sense

$$\delta^{(d-2)} \begin{pmatrix} q_T\cos(\phi_h) \\ q_T\sin(\phi_h) \\ \vdots \end{pmatrix} = \delta(q_T) q_T^{2\varepsilon-1} \frac{\Gamma(1-\varepsilon)}{2\pi^{1-\varepsilon}} \tag{D.25}$$

Now we can use non-standard analysis by writing $2q_T\delta(q_T^2) = \delta(q_T)$. Putting everything together we find

$$\begin{aligned} \frac{\mathrm{d}\sigma_{qq}^{n=0}}{\mathrm{d}\mathcal{R}_d} = \frac{(2\pi)^{4-2\varepsilon}\Gamma(1-\varepsilon)}{\pi^{1-\varepsilon}} \frac{q_T^{2\varepsilon}}{zQ^2}\delta(q_T^2) \int_0^1 \mathrm{d}\xi \int_0^1 \mathrm{d}\zeta\, f_q(\xi) D_q(\zeta)\delta(\xi - x)\delta(\zeta - z) \\ \times \frac{1}{N_c}\,\mathrm{tr}\big(H \not{p} \gamma^0 H^\dagger \gamma^0 \not{p}_h\big) \end{aligned} \tag{D.26}$$

where we already used $\xi = x$ to simplify the expression. Now we have to calculate the trace including the hard part depicted in figure D.1. We will call the loop momenta g which will always run counter clockwise. Note also the factor one-half in the second row of the figure, which originates from the square root in the definition of the renormalization factor of the quark fields Z_{ψ_q}. Using the Feynman rules we find

$$\begin{aligned} &\bar{u}_i^A(p_h) H_{ij}^{AB} u_j^B(p) \\ &= \bar{u}(l_2)(-\mathrm{i}e\gamma^\mu)u(l_1)\left(\frac{-\mathrm{i}g_{\mu\nu}}{q^2 + \mathrm{i}\eta}\right) \\ &\quad \times \bar{u}^A(p_h)\Bigg[(-\mathrm{i}ee_q\gamma^\nu)\delta_{ab} \end{aligned}$$

$$
\begin{aligned}
&+ \int \frac{\mathrm{d}^d g}{(2\pi)^d} (-\mathrm{i} g_s \mu^\varepsilon \gamma^\rho t^a_{AC}) \frac{\mathrm{i}(\not{p}_h - \not{g})}{(p_h - g)^2 + \mathrm{i}\eta} (-\mathrm{i} e e_q \gamma^\nu) \\
&\qquad \times \frac{\mathrm{i}(\not{p} - \not{g})}{(p-g)^2 + \mathrm{i}\eta} (-\mathrm{i} g_s \mu^\varepsilon \gamma^\sigma t^b_{CB}) \frac{-\mathrm{i}\delta^{ab} g_{\rho\sigma}}{g^2 + \mathrm{i}\eta} \\
&+ (-\mathrm{i} e e_q \gamma^\nu) \left(-\frac{C_F \alpha_s}{4\pi} \frac{1}{\varepsilon} \frac{(4\pi)^\varepsilon}{\Gamma(1-\varepsilon)} \right) \delta_{AB} \\
&+ \frac{1}{2} \int \frac{\mathrm{d}^d g}{(2\pi)^d} (-\mathrm{i} e e_q \gamma^\nu) \frac{\mathrm{i}\not{p}}{p^2 + \mathrm{i}\eta} (-\mathrm{i} g_s \mu^\varepsilon \gamma^\rho t^a_{AC}) \\
&\qquad \times \frac{\mathrm{i}(\not{p} - \not{g})}{(p-g)^2 + \mathrm{i}\eta} (-\mathrm{i} g_s \mu^\varepsilon \gamma^\sigma t^b_{CB}) \frac{-\mathrm{i}\delta^{ab} g_{\rho\sigma}}{g^2 + \mathrm{i}\eta} \\
&+ \frac{1}{2} \int \frac{\mathrm{d}^d g}{(2\pi)^d} (-\mathrm{i} g_s \mu^\varepsilon \gamma^\rho t^a_{AC}) \frac{\mathrm{i}(\not{p}_h - \not{g})}{(p_h - g)^2 + \mathrm{i}\eta} (-\mathrm{i} g_s \mu^\varepsilon \gamma^\sigma t^b_{CB}) \\
&\qquad \times \frac{\mathrm{i}\not{p}_h}{p_h^2 + \mathrm{i}\eta} (-\mathrm{i} e e_q \gamma^\nu) \frac{-\mathrm{i}\delta^{ab} g_{\rho\sigma}}{g^2 + \mathrm{i}\eta} \\
&+ \frac{1}{2} \left(-\mathrm{i} \frac{C_F \alpha_s}{4\pi} \frac{1}{\varepsilon} \frac{(4\pi)^\varepsilon}{\Gamma(1-\varepsilon)} \not{p}_h \right) \frac{\mathrm{i}\not{p}_h}{p_h^2 + \mathrm{i}\eta} (-\mathrm{i} e e_q \gamma^\nu) \delta_{AB} \\
&+ \frac{1}{2} (-\mathrm{i} e e_q \gamma^\nu) \frac{\mathrm{i}\not{p}}{p^2 + \mathrm{i}\eta} \left(-\mathrm{i} \frac{C_F \alpha_s}{4\pi} \frac{1}{\varepsilon} \frac{(4\pi)^\varepsilon}{\Gamma(1-\varepsilon)} \not{p} \right) \delta_{AB} \Bigg] u^B(p) \\
= \; & \frac{4\pi\alpha e_q \delta_{AB}}{Q^2} \bar{u}(l_2) \gamma^\mu u(l_1) \\
& \times \bar{u}^A(p_h) \Bigg[-\mathrm{i}\gamma_\mu - 4\pi\alpha_s C_F \\
&\qquad \times \mu^{2\varepsilon} \int \frac{\mathrm{d}^d g}{(2\pi)^d} \Bigg\{ \frac{\gamma^\rho (\not{p}_h - \not{g}) \gamma_\mu (\not{p} - \not{g}) \gamma_\rho}{\left[(p_h - g)^2 + \mathrm{i}\eta\right] \left[(p-g)^2 + \mathrm{i}\eta\right] \left[g^2 + \mathrm{i}\eta\right]} \\
&\qquad\qquad + \frac{1}{2} \frac{\gamma_\mu \not{p} \gamma^\rho (\not{p} - \not{g}) \gamma_\rho}{\left[p^2 + \mathrm{i}\eta\right] \left[(p-g)^2 + \mathrm{i}\eta\right] \left[g^2 + \mathrm{i}\eta\right]} \\
&\qquad\qquad + \frac{1}{2} \frac{\gamma^\rho (\not{p}_h - \not{g}) \gamma_\rho \not{p}_h \gamma_\mu}{\left[(p_h - g)^2 + \mathrm{i}\eta\right] \left[p_h^2 + \mathrm{i}\eta\right] \left[g^2 + \mathrm{i}\eta\right]} \Bigg\} \Bigg] u^B(p)
\end{aligned}
\tag{D.27}
$$

In the first step we wrote down all terms corresponding to the single diagrams in figure D.1, reading from top to bottom and from left to right. On the right-hand side of the equation we used matrix notation for the Dirac space, suppressing the indices, while keeping them explicit in color space. $u(k)$ denotes a Dirac spinor with momentum k, where we suppress the spin argument because we consider unpolarized particles only. We keep this in mind and will sum over outgoing and average over incoming spin states. e_q is the fraction of the elementary charge e of the involved quark flavour and η regularized the propagators. We notice that all counter terms cancel each other, which means that occurring ultraviolet poles have to cancel too. We denoted the parameter

giving the offset of four dimensions for ultraviolet poles as always with $\varepsilon > 0$. For infrared poles we will have to shift the dimension into the different direction. In that case we change the sign of ε and denote this with $\tilde{\varepsilon} \equiv -\varepsilon > 0$. By this manner we can assure that no spurious cancellation between ultraviolet and infrared poles occurs. Before we take the trace we will calculate the loop integrals, which we define by

$$I^{(1)}_{qq,\mu} \equiv \mu^{2\varepsilon} \int \frac{\mathrm{d}^d g}{(2\pi)^d} \frac{\gamma^\rho (\not{p}_h - \not{g}) \gamma_\mu (\not{p} - \not{g}) \gamma_\rho}{\left[(p_h - g)^2 + \mathrm{i}\eta\right] \left[(p-g)^2 + \mathrm{i}\eta\right] \left[g^2 + \mathrm{i}\eta\right]} \tag{D.28a}$$

$$I^{(2)}_{qq,\mu} \equiv \frac{\mu^{2\varepsilon}}{2} \int \frac{\mathrm{d}^d g}{(2\pi)^d} \frac{\gamma_\mu \not{p} \gamma^\rho (\not{p} - \not{g}) \gamma_\rho}{\left[p^2 + \mathrm{i}\eta\right] \left[(p-g)^2 + \mathrm{i}\eta\right] \left[g^2 + \mathrm{i}\eta\right]} \tag{D.28b}$$

$$I^{(3)}_{qq,\mu} \equiv \frac{\mu^{2\varepsilon}}{2} \int \frac{\mathrm{d}^d g}{(2\pi)^d} \frac{\gamma^\rho (\not{p}_h - \not{g}) \gamma_\rho \not{p}_h \gamma_\mu}{\left[(p_h - g)^2 + \mathrm{i}\eta\right] \left[p_h^2 + \mathrm{i}\eta\right] \left[g^2 + \mathrm{i}\eta\right]} \tag{D.28c}$$

We start with solving $I^{(1)}_{qq,\mu}$ by using two Feynman parameters x and y

$$I^{(1)}_{qq,\mu} = 2\mu^{2\varepsilon} \int_0^1 \mathrm{d}x \int_0^{1-x} \mathrm{d}y \int \frac{\mathrm{d}^d g}{(2\pi)^d} \frac{\gamma^\rho (\not{p}_h - \not{g}) \gamma_\mu (\not{p} - \not{g}) \gamma_\rho}{\Big[\big(g - (xp_h + yp)\big)^2 - 2xy(p \cdot p_h) + \mathrm{i}\eta\Big]^3} \tag{D.29}$$

Now we shift the loop momentum by defining $r = g - (yp + xp_h)$ and use the Dirac algebra in d dimensions, in particular $\gamma^\rho \gamma_\nu \gamma_\mu \gamma_\sigma \gamma_\rho = -2\gamma_\sigma \gamma_\mu \gamma_\nu + (4-d)\gamma_\nu \gamma_\mu \gamma_\sigma$ to find

$$\begin{aligned} I^{(1)}_{qq,\mu} = 4\mu^{2\varepsilon} \int_0^1 \mathrm{d}x \int_0^{1-x} \mathrm{d}y \int \frac{\mathrm{d}^d r}{(2\pi)^d} & \frac{1}{\left[r^2 - 2xy(p \cdot p_h) + \mathrm{i}\eta\right]^3} \\ \times \Big\{ & -(1-\varepsilon) \not{r} \gamma_\mu \not{r} + (1-\varepsilon)\big[x(1-x) \not{p}_h \gamma_\mu \not{p}_h + y(1-y) \not{p} \gamma_\mu \not{p}\big] \\ & + \not{p}_h \gamma_\mu \not{p} \big[\varepsilon(1-x)(1-y) - xy\big] + \not{p} \gamma_\mu \not{p}_h \big[\varepsilon xy - (1-x)(1-y)\big] \Big\} \end{aligned} \tag{D.30}$$

Luckily it is not necessary to calculate all of the terms, because we already know that they are accomplished by two Dirac spinors. Using the Dirac equation the expression simplifies to

$$\begin{aligned} \bar{u}(p_h) I^{(1)}_{qq,\mu} u(p) = 4\mu^{2\varepsilon} \int_0^1 \mathrm{d}x \int_0^{1-x} \mathrm{d}y \int \frac{\mathrm{d}^d r}{(2\pi)^d} & \frac{1}{\left[r^2 - 2xy(p \cdot p_h) + \mathrm{i}\eta\right]^3} \\ \times \bar{u}(p_h) \Big\{ & -(1-\varepsilon) \not{r} \gamma_\mu \not{r} + \not{p} \gamma_\mu \not{p}_h \big[\varepsilon xy - (1-x)(1-y)\big] \Big\} u(p) \end{aligned} \tag{D.31}$$

where we ignore the color indices of the quark spinors for the moment, as the color state is irrelevant in this discussion. We can split up the second structure by

$$\not{p} \gamma_\mu \not{p}_h = 2\not{p}_h p^\mu - 2(p \cdot p_h)\gamma^\mu + 2p_h^\mu \not{p} - \not{p}_h \gamma^\mu \not{p} \tag{D.32}$$

and using again the Dirac equation, we are left with only two integrals

$$I^{(1a)}_{qq,\mu} \equiv -4(1-\varepsilon)\mu^{2\varepsilon} \int_0^1 \mathrm{d}x \int_0^{1-x} \mathrm{d}y \int \frac{\mathrm{d}^d r}{(2\pi)^d} \frac{\not{r}\gamma_\mu \not{r}}{\left[r^2 - 2xy(p\cdot p_h) + \mathrm{i}\eta\right]^3} \tag{D.33a}$$

$$I^{(1b)}_{qq,\mu} \equiv -8\mu^{2\varepsilon}(p\cdot p_h)\gamma_\mu \int_0^1 \mathrm{d}x \int_0^{1-x} \mathrm{d}y \int \frac{\mathrm{d}^d r}{(2\pi)^d} \frac{\left[\varepsilon xy - (1-x)(1-y)\right]}{\left[r^2 - 2xy(p\cdot p_h) + \mathrm{i}\eta\right]^3} \tag{D.33b}$$

To solve the first of them, we write

$$I^{(1a)}_{qq,\mu} = -4(1-\varepsilon)\mu^{2\varepsilon}\gamma^\rho\gamma_\mu\gamma^\sigma \int_0^1 \mathrm{d}x \int_0^{1-x} \mathrm{d}y\, I^{(1a)}_{\rho\sigma}(x,y) \tag{D.34}$$

with

$$I^{(1a)}_{\rho\sigma}(x,y) = \int \frac{\mathrm{d}^d r}{(2\pi)^d} \frac{r_\rho r_\sigma}{\left[r^2 - 2xy(p\cdot p_h) + \mathrm{i}\eta\right]^3} = g_{\rho\sigma} I^{(1a)}(x,y) \tag{D.35}$$

where the last step lists all possible Lorentz structures. Contracting the equation with $g^{\rho\sigma}$ and splitting up the integrand yields

$$\begin{aligned} I^{(1a)}(x,y) &= \frac{1}{d}\int \frac{\mathrm{d}^d r}{(2\pi)^d} \frac{r^2}{\left[r^2 - 2xy(p\cdot p_h) + \mathrm{i}\eta\right]^3} \\ &= \frac{1}{d}\int \frac{\mathrm{d}^d r}{(2\pi)^d} \left[\frac{1}{\left[r^2 - 2xy(p\cdot p_h) + \mathrm{i}\eta\right]^2} + \frac{2xy(p\cdot p_h)}{\left[r^2 - 2xy(p\cdot p_h) + \mathrm{i}\eta\right]^3}\right] \end{aligned} \tag{D.36}$$

To solve the integrals we perform a Wick rotation into the Euclidean space, denoted with a subscript E, and use spheric coordinates

$$\begin{aligned} I^{(1a)}(x,y) &= \frac{\mathrm{i}}{d}\int \frac{\mathrm{d}^d r_E}{(2\pi)^d} \left[\frac{1}{\left[r^2 + 2xy(p\cdot p_h) - \mathrm{i}\eta\right]^2} + \frac{2xy(p\cdot p_h)}{\left[r^2 + 2xy(p\cdot p_h) - \mathrm{i}\eta\right]^3}\right] \\ &= \frac{\mathrm{i}}{d}\int \mathrm{d}\Omega_d \\ &\quad \times \int_0^\infty \frac{\mathrm{d}|r_E|}{(2\pi)^d} \left[\frac{|r_E|^{d-1}}{\left[|r_E|^2 + 2xy(p\cdot p_h) - \mathrm{i}\eta\right]^2} - \frac{2xy(p\cdot p_h)|r_E|^{d-1}}{\left[|r_E|^2 + 2xy(p\cdot p_h) - \mathrm{i}\eta\right]^3}\right] \end{aligned} \tag{D.37}$$

We may now drop the regulator η and substitute

$$|r_E| = \sqrt{2xy(p\cdot p_h)}\sqrt{\frac{\tau}{1-\tau}} \tag{D.38}$$

to solve the integral by using the definition of the Beta function

$$I^{(1a)}(x,y) = \frac{\mathrm{i}}{d}\frac{2\pi^{2-\varepsilon}}{\Gamma(2-\varepsilon)}\frac{1}{(2\pi)^{4-2\varepsilon}}(2xy(p\cdot p_h))^{-\varepsilon}\frac{1}{2}\left[\frac{\Gamma(\varepsilon)\Gamma(2-\varepsilon)}{\Gamma(2)} - \frac{\Gamma(1+\varepsilon)\Gamma(2-\varepsilon)}{\Gamma(3)}\right] \tag{D.39}$$

Inserting into $I^{(1a)}_{qq,\mu}$ and using $\gamma^\rho\gamma_\mu\gamma_\rho = -2(1-\varepsilon)\gamma_\mu$ gives

$$\begin{aligned}
I^{(1a)}_{qq,\mu} &= 4\mathrm{i}\gamma_\mu\frac{(1-\varepsilon)^2}{2-\varepsilon}\mu^{2\varepsilon}\frac{1}{(4\pi)^{2-\varepsilon}}\left[\frac{\Gamma(\varepsilon)}{\Gamma(2)} - \frac{\Gamma(1+\varepsilon)}{\Gamma(3)}\right]\int_0^1 \mathrm{d}x \int_0^{1-x}\mathrm{d}y\,(2xy(p\cdot p_h))^{-\varepsilon} \\
&= \frac{2\mathrm{i}\gamma_\mu}{(4\pi)^2}\left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)^{\varepsilon}(1-\varepsilon)^2\Gamma(\varepsilon)\int_0^1 \mathrm{d}x \int_0^{1-x}\mathrm{d}y\,(xy)^{-\varepsilon} \\
&= \frac{2\mathrm{i}\gamma_\mu}{(4\pi)^2}\left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)^{\varepsilon}(1-\varepsilon)\Gamma(\varepsilon)\int_0^1 \mathrm{d}x\,x^{-\varepsilon}(1-x)^{1-\varepsilon} \\
&= \frac{\mathrm{i}\gamma_\mu}{(4\pi)^2}\left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)^{\varepsilon}\frac{1-\varepsilon}{1-2\varepsilon}\frac{\Gamma(\varepsilon)\Gamma^2(1-\varepsilon)}{\Gamma(1-2\varepsilon)}
\end{aligned} \tag{D.40}$$

where we used again the definition of the Beta function to express the final integral in terms of Gamma functions. To solve $I^{(1b)}_{qq,\mu}$ we note that the r integral is equal to the second term in $I^{(1a)}(x,y)$ up to a prefactor. We read off the result

$$\begin{aligned}
I^{(1b)}_{qq,\mu} = &-\frac{2\mathrm{i}\gamma_\mu}{(4\pi)^2}\left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)^{\varepsilon}\Gamma(1+\varepsilon) \\
&\times\int_0^1 \mathrm{d}x \int_0^{1-x}\mathrm{d}y\,\big[(1-x)(1-y)-\varepsilon xy\big](xy)^{-1-\varepsilon}
\end{aligned} \tag{D.41}$$

To solve the y integral we have to choose $\varepsilon < 0$ indicating an infrared pole. As described before we will denote this with $\tilde{\varepsilon} = -\varepsilon$. Finally we use again the definition of the Beta function

$$\begin{aligned}
I^{(1b)}_{qq,\mu} &= -\frac{2\mathrm{i}\gamma_\mu}{(4\pi)^2}\left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)^{-\tilde{\varepsilon}}\Gamma(1-\tilde{\varepsilon}) \\
&\quad\times\int_0^1 \mathrm{d}x\,x^{-1+\tilde{\varepsilon}}(1-x)^{1+\tilde{\varepsilon}}\left(\frac{1}{\tilde{\varepsilon}(1+\tilde{\varepsilon})}+x\right) \\
&= -\frac{\mathrm{i}\gamma_\mu}{(4\pi)^2}\left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)^{-\tilde{\varepsilon}}\frac{\Gamma(1-\tilde{\varepsilon})\Gamma^2(1+\tilde{\varepsilon})}{\Gamma(1+2\tilde{\varepsilon})}\frac{2+\tilde{\varepsilon}^2}{\tilde{\varepsilon}^2(1+2\tilde{\varepsilon})}
\end{aligned} \tag{D.42}$$

Our final result for $I^{(1)}_{qq,\mu}$ is therefore

$$I^{(1)}_{qq,\mu} = -\frac{\mathrm{i}\gamma_\mu}{(4\pi)^2}\left[\left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)^{-\tilde{\varepsilon}}\frac{\Gamma(1-\tilde{\varepsilon})\Gamma^2(1+\tilde{\varepsilon})}{\Gamma(1+2\tilde{\varepsilon})}\frac{2+\tilde{\varepsilon}^2}{\tilde{\varepsilon}^2(1+2\tilde{\varepsilon})} - \left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)^{\varepsilon}\frac{1-\varepsilon}{1-2\varepsilon}\frac{\Gamma(\varepsilon)\Gamma^2(1-\varepsilon)}{\Gamma(1-2\varepsilon)}\right] \tag{D.43}$$

We continue with solving $I^{(2)}_{qq,\mu}$ by using $\gamma_\rho\gamma_\nu\gamma^\rho = (2-d)\gamma_\nu$, introducing a Feynman parameter x to simplify the second and third factor in the denominator and shifting the loop momentum by defining $r = g - p(1-x)$

$$\begin{aligned} I^{(2)}_{qq,\mu} &= \mu^{2\varepsilon}\frac{2-d}{2\big[p^2+\mathrm{i}\eta\big]}\int\frac{\mathrm{d}^d g}{(2\pi)^d}\int\limits_0^1\mathrm{d}x\,\frac{\gamma_\mu(p^2-\not{p}\not{g})}{\Big[\big(g-p(1-x)\big)^2+\mathrm{i}\eta\Big]^2} \\ &= \mu^{2\varepsilon}\frac{2-d}{2\big[p^2+\mathrm{i}\eta\big]}\int\limits_0^1\mathrm{d}x\int\frac{\mathrm{d}^d r}{(2\pi)^d}\,\frac{\gamma_\mu(xp^2-\not{p}\not{r})}{(r^2+\mathrm{i}\eta)^2} \end{aligned} \tag{D.44}$$

The second term in the numerator is antisymmetric under $r \to -r$ and is consequently zero. For the rest we perform again a Wick rotation into the Euclidean space and use spheric coordinates finding

$$\begin{aligned} I^{(2)}_{qq,\mu} &= \mathrm{i}\gamma_\mu\frac{(2-d)\mu^{2\varepsilon}}{2}\int\frac{\mathrm{d}^d r_E}{(2\pi)^d}\,\frac{1}{(r_E^2-\mathrm{i}\eta)^2}\int\limits_0^1\mathrm{d}x\,x \\ &= \mathrm{i}\gamma_\mu\frac{(2-d)\mu^{2\varepsilon}}{4(2\pi)^d}\int\limits_0^\infty\mathrm{d}|r_E|\frac{1}{|r_E|^{1+2\varepsilon}}\int\mathrm{d}\Omega_d \\ &= \mathrm{i}\gamma_\mu\frac{(2-d)\mu^{2\varepsilon}}{4(2\pi)^d}\frac{2\pi^{d/2}}{\Gamma\left(\frac{d}{2}\right)}\int\limits_0^\infty\frac{\mathrm{d}y}{y^{1+2\varepsilon}} \end{aligned} \tag{D.45}$$

The integral $I^{(3)}_{qq,\mu}$ is completely analogous and gives the same result, which means

$$I^{(2)}_{qq,\mu} + I^{(3)}_{qq,\mu} = \frac{-\mathrm{i}\gamma_\mu}{(4\pi)^2}\frac{2\big(4\pi\mu^2\big)^\varepsilon}{\Gamma(1-\varepsilon)}\int\limits_0^\infty\frac{\mathrm{d}y}{y^{1+2\varepsilon}} \tag{D.46}$$

The integral which is left is sometimes set directly to zero, because it is scaleless [62]. But this mingles ultraviolet and infrared poles, because it is UV as well as IR divergent. We prefer to keep track of the types of the singularities. Therefore we introduce the arbitrary number λ which separates the two regimes. It will cancel as soon as we

develop the expressions in terms of ε. This means our final result for the two integrals is

$$\begin{aligned} I^{(2)}_{qq,\mu} + I^{(3)}_{qq,\mu} &= \frac{-\mathrm{i}\gamma_\mu}{(4\pi)^2} \frac{2\big(4\pi\mu^2\big)^\varepsilon}{\Gamma(1-\varepsilon)} \left[\int_0^\lambda \frac{\mathrm{d}y}{y^{1+2\varepsilon}} + \int_\lambda^\infty \frac{\mathrm{d}y}{y^{1+2\varepsilon}} \right] \\ &= \frac{-\mathrm{i}\gamma_\mu}{(4\pi)^2} \left[\frac{1}{\tilde{\varepsilon}} \left(\frac{4\pi\mu^2}{\lambda^2} \right)^{-\tilde{\varepsilon}} \frac{1}{\Gamma(1+\tilde{\varepsilon})} + \frac{1}{\varepsilon} \left(\frac{4\pi\mu^2}{\lambda^2} \right)^{\varepsilon} \frac{1}{\Gamma(1-\varepsilon)} \right] \end{aligned} \tag{D.47}$$

Now we can put all terms together and plug the results into equation (D.27)

$$\begin{aligned} &\bar{u}_i^A(p_h) H_{ij}^{AB} u_j^B(p) \\ &= -\frac{4\pi\mathrm{i}\alpha e_q \delta_{AB}}{Q^2} \bar{u}(l_2)\gamma^\mu u(l_1) \bar{u}^A(p_h)\gamma_\mu u^B(p) \\ &\quad \times \left[1 - \frac{\alpha_s C_F}{4\pi} \right. \\ &\qquad \times \left\{ \frac{1}{\tilde{\varepsilon}^2} \left(\frac{4\pi\mu^2}{2(p\cdot p_h)} \right)^{-\tilde{\varepsilon}} \frac{\Gamma(1-\tilde{\varepsilon})\Gamma^2(1+\tilde{\varepsilon})}{\Gamma(1+2\tilde{\varepsilon})} \frac{2+\tilde{\varepsilon}^2}{1+2\tilde{\varepsilon}} + \frac{1}{\tilde{\varepsilon}} \left(\frac{4\pi\mu^2}{\lambda^2} \right)^{-\tilde{\varepsilon}} \frac{1}{\Gamma(1+\tilde{\varepsilon})} \right. \\ &\qquad\quad \left. \left. - \frac{1}{\varepsilon} \left(\frac{4\pi\mu^2}{2(p\cdot p_h)} \right)^{\varepsilon} \frac{1-\varepsilon}{1-2\varepsilon} \frac{\Gamma(1+\varepsilon)\Gamma^2(1-\varepsilon)}{\Gamma(1-2\varepsilon)} + \frac{1}{\varepsilon} \left(\frac{4\pi\mu^2}{\lambda^2} \right)^{\varepsilon} \frac{1}{\Gamma(1-\varepsilon)} \right\} \right] \end{aligned} \tag{D.48}$$

At this point we take the Taylor series for ε and partially also for $\tilde{\varepsilon}$. For the latter we taylor only the trivial factors and the terms involving the parameter λ, because we aim to achieve the standard way to present the result

$$\begin{aligned} \bar{u}_i^A(p_h) H_{ij}^{AB} u_j^B(p) = &-\frac{4\pi\mathrm{i}\alpha e_q \delta_{AB}}{Q^2} \bar{u}(l_2)\gamma^\mu u(l_1) \bar{u}^A(p_h)\gamma_\mu u^B(p) \\ &\times \left[1 - \frac{\alpha_s C_F}{4\pi} \right. \\ &\quad \times \left\{ \left(\frac{4\pi\mu^2}{2(p\cdot p_h)} \right)^{-\tilde{\varepsilon}} \frac{\Gamma(1-\tilde{\varepsilon})\Gamma^2(1+\tilde{\varepsilon})}{\Gamma(1+2\tilde{\varepsilon})} \left[\frac{2}{\tilde{\varepsilon}^2} - \frac{4}{\tilde{\varepsilon}} + 9 + \mathcal{O}(\tilde{\varepsilon}) \right] \right. \\ &\qquad + \frac{1}{\tilde{\varepsilon}} + \gamma_E + \ln\left(\frac{4\pi\mu^2}{\lambda^2} \right) + \mathcal{O}(\tilde{\varepsilon}) \\ &\qquad - \frac{1}{\varepsilon} - 1 + \gamma_E - \ln\left(\frac{4\pi\mu^2}{2(p\cdot p_h)} \right) + \mathcal{O}(\varepsilon) \\ &\qquad \left. \left. + \frac{1}{\varepsilon} - \gamma_E - \ln\left(\frac{4\pi\mu^2}{\lambda^2} \right) + \mathcal{O}(\varepsilon) \right\} \right] \end{aligned} \tag{D.49}$$

where γ_E is the Euler–Mascheroni constant. Obviously all ultraviolet divergences cancel as expected, as does the parameter λ. To simplify the remaining terms, we use the expansion

$$\left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)^{-\tilde{\varepsilon}}\frac{\Gamma^2(1+\tilde{\varepsilon})\Gamma(1-\tilde{\varepsilon})}{\Gamma(1+2\tilde{\varepsilon})}\left[\frac{1}{\tilde{\varepsilon}}-1\right]=\frac{1}{\tilde{\varepsilon}}-1+\gamma_E-\ln\left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)+\mathcal{O}(\tilde{\varepsilon}) \quad \text{(D.50)}$$

and replace $\tilde{\varepsilon}$ again with $\varepsilon=-\tilde{\varepsilon}$

$$\begin{aligned}\bar{u}_i^A(p_h)H_{ij}^{AB}u_j^B(p)=&-\frac{4\pi\mathrm{i}\alpha e_q\delta_{AB}}{Q^2}\bar{u}(l_2)\gamma^\mu u(l_1)\bar{u}^A(p_h)\gamma_\mu u^B(p)\\&\times\left[1-\frac{\alpha_s C_F}{4\pi}\left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)^{\varepsilon}\frac{\Gamma(1+\varepsilon)\Gamma^2(1-\varepsilon)}{\Gamma(1-2\varepsilon)}\left[\frac{2}{\varepsilon^2}+\frac{3}{\varepsilon}+8\right]\right]\end{aligned} \quad \text{(D.51)}$$

The result means, that the NLO contributions of the virtual diagrams are a trivial factor in terms of the Dirac and color structure, which should be expected, because the virtual diagrams do not change the 'interface' of the diagrams. This means the trace in equation (D.26) is given by (implicitly averaging over incoming and summing over outgoing spin states, which adds a factor one-half)

$$\begin{aligned}\frac{1}{N_c}\operatorname{tr}\left(H\not{p}\gamma^0H^\dagger\gamma^0\not{p}_h\right)=&\frac{(4\pi)^2\alpha^2e_q^2}{2Q^4}\operatorname{tr}\left(\not{l}_2\gamma^\mu\not{l}_1\gamma^\nu\right)\operatorname{tr}\left(\gamma_\mu\not{p}\gamma_\nu\not{p}_h\right)\\&\times\left[1-\frac{\alpha_s C_F}{4\pi}\left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)^{\varepsilon}\frac{\Gamma(1+\varepsilon)\Gamma^2(1-\varepsilon)}{\Gamma(1-2\varepsilon)}\left[\frac{2}{\varepsilon^2}+\frac{3}{\varepsilon}+8\right]\right]\end{aligned} \quad \text{(D.52)}$$

For the traces we use the TRACER package [292] with the scalar products defined in equations (D.20)

$$\begin{aligned}\frac{1}{N_c}\operatorname{tr}\left(H\not{p}\gamma^0H^\dagger\gamma^0\not{p}_h\right)=&2(4\pi)^2\alpha^2e_q^2\frac{\xi z}{x\zeta}\left(1-2\varepsilon-\frac{q_T^2}{Q^2}+\frac{Q^2+q_T^2}{Q^2}\cosh^2(\psi)\right.\\&\left.-2\frac{q_T}{Q}\cos(\phi)\cosh(\psi)\sinh(\psi)\right)\\&\times\left[1-\frac{\alpha_s C_F}{2\pi}\left(\frac{4\pi\mu^2}{2(p\cdot p_h)}\right)^{\varepsilon}\frac{\Gamma(1+\varepsilon)\Gamma^2(1-\varepsilon)}{\Gamma(1-2\varepsilon)}\left[\frac{2}{\varepsilon^2}+\frac{3}{\varepsilon}+8\right]\right]\end{aligned} \quad \text{(D.53)}$$

Sorting in terms of the structure functions A_k defined in equations (4.24) and inserting into equation (D.26) gives the result reported in (4.23).

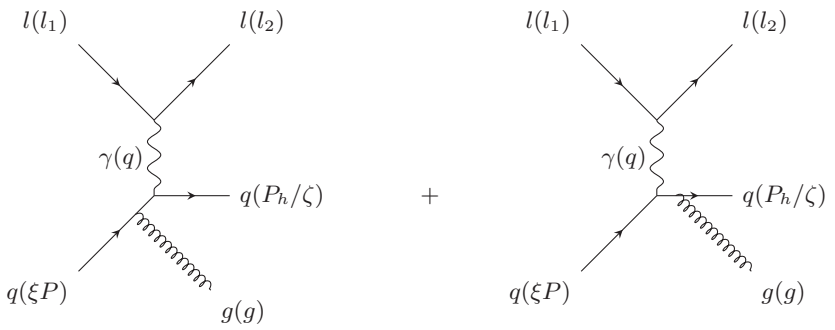


Figure D.2: Partonic invariant amplitude of SIDIS process with a quark in the initial state. The final state consists of a hadronizing quark and an unobserved gluon.

D.3.1.2 Real diagrams

We turn to the partonic diagrams containing one real gluon emission, as shown in figure D.2, which means that we evaluate the summand with $n = 1$ in equation (4.22)

$$\frac{\mathrm{d}\sigma_{qq}^{n=1}}{\mathrm{d}\mathcal{R}_d} = \sum_{\lambda,A} \int \frac{\mathrm{d}^d g}{(2\pi)^{d-1}} \delta((g)^2)\Theta(g^0) \int_0^1 \mathrm{d}\xi \int_0^1 \frac{\mathrm{d}\zeta}{\zeta} f_q(\xi) D_q(\zeta) \frac{Q^2}{4} \frac{z}{x\zeta^2} \\ \times (2\pi)^d \delta^d\left(l_1 - l_2 + \xi P - \frac{1}{\zeta} P_h - g\right) \frac{1}{N_c} \operatorname{tr}\left(H \not{n}_+ \gamma^0 H^\dagger \gamma^0 \not{n}\right) \tag{D.54}$$

We use the d dimensional delta distribution to evaluate the phase space integral of the gluon

$$\frac{\mathrm{d}\sigma_{qq}^{n=1}}{\mathrm{d}\mathcal{R}_d} = \frac{\pi Q^2}{2N_c} \frac{z}{x} \int_0^1 \mathrm{d}\xi \int_0^1 \frac{\mathrm{d}\zeta}{\zeta} \frac{1}{\zeta^2} f_q(\xi) D_q(\zeta) \delta\left(\left(q + \xi P - \frac{1}{\zeta} P_h\right)^2\right) \\ \times \Theta\left(q^0 + \xi P^0 - \frac{1}{\zeta} P_h^0\right) \sum_{\lambda,A} \operatorname{tr}\left(H \not{n}_+ \gamma^0 H^\dagger \gamma^0 \not{n}\right) \tag{D.55}$$

Then we rewrite the remaining delta distribution in terms of our specific reference system (see section 4.2)

$$\delta\left(\left(q + \xi P - \frac{1}{\zeta} P_h\right)^2\right) = \frac{\zeta}{zQ^2} \delta\left(\left(1 - \frac{\zeta}{z}\right)\left(1 - \frac{\xi}{x}\right) - \frac{q_T^2}{Q^2}\right) \tag{D.56}$$

and the Heaviside function

$$\Theta\left(q^0 + \xi P^0 - \frac{1}{\zeta} P_h^0\right) = \Theta\left(\frac{\xi}{x} - \frac{z}{\zeta} - \frac{z}{\zeta} \frac{q_T^2}{Q^2}\right) \tag{D.57}$$

We can now argue on the limits of the ξ and the ζ integrals. By the means of the delta distribution it has to be $\xi > x$ and $\zeta > z$ or $\xi < x$ and $\zeta < z$. The Heaviside function

restricts us to the first combination, namely $\xi > x$ and $\zeta > z$. Using the collinear expansion as in equations (D.22) we find

$$\begin{aligned}\frac{\mathrm{d}\sigma_{qq}^{n=1}}{\mathrm{d}\mathcal{R}_d} = \frac{\pi}{Q^2}\frac{1}{z}\int_x^1 \frac{\mathrm{d}\xi}{\xi}\int_z^1 \frac{\mathrm{d}\zeta}{\zeta} f_q(\xi) D_q(\zeta)\delta\left(\left(1-\frac{\zeta}{z}\right)\left(1-\frac{\xi}{x}\right)-\frac{q_T^2}{Q^2}\right) \\ \times \sum_{\lambda,A}\frac{1}{N_c}\operatorname{tr}\left(H \not{p} \gamma^0 H^\dagger \gamma^0 \not{p}_h\right)\end{aligned} \tag{D.58}$$

Finally we have to calculate H shown in figure D.2. With the standard Feynman rules in $d = 4 - 2\varepsilon$ dimensions and using Feynman gauge we find

$$\begin{aligned}&\bar{u}_i^A(p_h) H_{ij}^{AB} u_j^B(p) \\ &= \bar{u}(l_2)\left(-\mathrm{i}e\gamma^\mu\right)u(l_1)\left(\frac{-\mathrm{i}g_{\mu\nu}}{q^2+\mathrm{i}\eta}\right) \\ &\quad\times \bar{u}^A(p_h)\Bigg[(-\mathrm{i}ee_q\gamma^\nu)\frac{\mathrm{i}(\not{p}_h - \not{q})}{(p_h-q)^2+\mathrm{i}\eta}(-\mathrm{i}g_s\mu^\varepsilon\gamma^\rho t_{AB}^a)\epsilon_\rho^*(p-p_h+q,\lambda) \\ &\qquad\qquad + (-\mathrm{i}g_s\mu^\varepsilon\gamma^\rho t_{AB}^a)\epsilon_\rho^*(p-p_h+q,\lambda)\frac{\mathrm{i}(\not{p}+\not{q})}{(p+q)^2+\mathrm{i}\eta}(-\mathrm{i}ee_q\gamma^\nu)\Bigg]u^B(p) \\ &= -\mathrm{i}\frac{e^2 e_q g_s \mu^\varepsilon}{Q^2} t_{AB}^a \bar{u}(l_2)\gamma^\mu u(l_1) \\ &\quad\times \bar{u}^A(p_h)\left[\frac{\gamma_\mu(\not{p}_h-\not{q})\not{\epsilon}^*(p-p_h+q,\lambda)}{(p_h-q)^2+\mathrm{i}\eta} + \frac{\not{\epsilon}^*(p-p_h+q,\lambda)(\not{p}+\not{q})\gamma_\mu}{(p+q)^2+\mathrm{i}\eta}\right]u^B(p)\end{aligned} \tag{D.59}$$

As before we suppress the spin argument of the Dirac spinors u. ϵ^μ is the polarization vector of the final state gluon. We suppress Dirac indices on the right-hand side of the equation again for brevity. Then the trace is given by (implicitly averaging over incoming and summing over outgoing spin states)

$$\begin{aligned}&\sum_{\lambda,A}\frac{1}{N_c}\operatorname{tr}\left(H\not{p}\gamma^0 H^\dagger\gamma^0\not{p}_h\right) \\ &= -\frac{e^4 e_q^2 g_s^2 \mu^{2\varepsilon} C_F}{2Q^4}\operatorname{tr}\left(\not{l}_2\gamma^\mu\not{l}_1\gamma^\nu\right) \\ &\quad\times\Bigg[\frac{\operatorname{tr}\left(\gamma_\mu(\not{p}_h-\not{q})\gamma^\rho\not{p}\gamma_\rho(\not{p}_h-\not{q})\gamma_\nu\not{p}_h\right)}{(p_h-q)^4} + \frac{\operatorname{tr}\left(\gamma_\mu(\not{p}_h-\not{q})\gamma^\rho\not{p}\gamma_\nu(\not{p}+\not{q})\gamma_\rho\not{p}_h\right)}{\left[(p_h-q)^2+\mathrm{i}\eta\right]\left[(p+q)^2-\mathrm{i}\eta\right]} \\ &\qquad + \frac{\operatorname{tr}\left(\gamma^\rho(\not{p}+\not{q})\gamma_\mu\not{p}\gamma_\rho(\not{p}_h-\not{q})\gamma_\nu\not{p}_h\right)}{\left[(p+q)^2+\mathrm{i}\eta\right]\left[(p_h-q)^2-\mathrm{i}\eta\right]} + \frac{\operatorname{tr}\left(\gamma^\rho(\not{p}+\not{q})\gamma_\mu\not{p}\gamma_\nu(\not{p}+\not{q})\gamma_\rho\not{p}_h\right)}{(p+q)^4}\Bigg]\end{aligned} \tag{D.60}$$

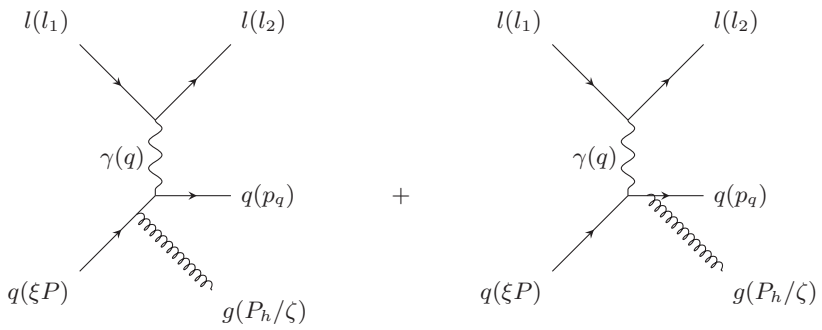


Figure D.3: Partonic invariant amplitude of SIDIS process with a quark in the initial state. The final state consists of a hadronizing gluon and an unobserved quark.

Here the color trace has been performed already and we used Feynman gauge by writing the polarization sum

$$\sum_{\lambda} \epsilon_{\mu}^{*}(p,\lambda)\epsilon_{\nu}(p,\lambda) = -g_{\mu\nu} \tag{D.61}$$

Now using the TRACER package [292] with the scalar product given in equations (D.20), sorting the result in terms of the structure functions defined in equations (4.24) and replacing q_T^2 in the denominator by means of the overall delta distribution if possible yields the result given in equation (4.27).

D.3.2 Quark PDF and gluon FF—the gq channel

We continue with the channel including an incoming quark and an hadronizing gluon. The partonic invariant amplitude is shown in figure D.3. The figure is apparently similar to figure D.2 despite of the outgoing momenta, which imply that this time the gluon will hadronize into a measured hadron, while the quark will be unobserved. But due to the different nature of the gluon, this has a quite deep impact on H which reads now

$$\begin{aligned}
H_{i,A}^{a,\mu} u_i^A(p) &= \bar{u}(l_2)(-\mathrm{i}e\gamma^{\rho})u(l_1)\left(\frac{-\mathrm{i}g_{\rho\sigma}}{q^2+\mathrm{i}\eta}\right) \\
&\quad \times \bar{u}^B(q+p-p_h)\Bigg[(-\mathrm{i}ee_q\gamma^{\sigma})\frac{\mathrm{i}(\not{p}-\not{p}_h)}{(p-p_h)^2+\mathrm{i}\eta}(-\mathrm{i}g_s\mu^{\varepsilon}\gamma^{\mu}t_{BA}^a) \\
&\qquad + (-\mathrm{i}g_s\mu^{\varepsilon}\gamma^{\mu}t_{BA}^a)\frac{\mathrm{i}(\not{q}+\not{p})}{(q+p)^2+\mathrm{i}\eta}(-\mathrm{i}ee_q\gamma^{\sigma})\Bigg]u^A(p) \\
&= -\frac{\mathrm{i}e^2e_qg_s\mu^{\varepsilon}}{Q^2}t_{BA}^a\bar{u}(l_2)\gamma_{\sigma}u(l_1) \\
&\quad \times \bar{u}^B(q+p-p_h)\left[\frac{\gamma^{\sigma}(\not{p}-\not{p}_h)\gamma^{\mu}}{(p-p_h)^2+\mathrm{i}\eta} + \frac{\gamma^{\mu}(\not{q}+\not{p})\gamma^{\sigma}}{(q+p)^2+\mathrm{i}\eta}\right]u^A(p)
\end{aligned} \tag{D.62}$$

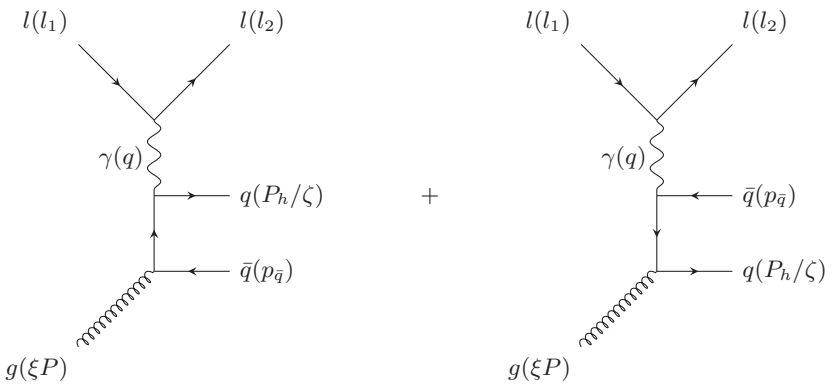


Figure D.4: Partonic invariant amplitude of SIDIS process with a gluon in the initial state. The final state consists of a hadronizing quark and an unobserved antiquark.

where we already replaced the placeholder variable $p_q = p + q - p_h$ by the means of the overall delta distribution and suppressed the Dirac indices on the right-hand side for brevity. This means the trace in equation (4.46) can be written as

$$\begin{aligned}\frac{-g_{\mu\nu}}{N_c}\,\mathrm{tr}\big(H^{\mu}\not{p}\gamma^0 H^{\dagger,\nu}\big) = -\frac{e^4 e_q^2 g_s^2 \mu^{2\varepsilon} C_F}{2Q^4}\,\mathrm{tr}\big(\not{l}_2\gamma_\sigma \not{l}_1\gamma_\rho\big) \\ \times\Bigg[\frac{\mathrm{tr}\big((\not{q}+\not{p}-\not{p}_h)\gamma^\sigma(\not{p}-\not{p}_h)\gamma^\mu\not{p}\gamma_\mu(\not{p}-\not{p}_h)\gamma^\rho\big)}{(p-p_h)^4} \\ + \frac{\mathrm{tr}\big((\not{q}+\not{p}-\not{p}_h)\gamma^\sigma(\not{p}-\not{p}_h)\gamma^\mu\not{p}\gamma^\rho(\not{q}+\not{p})\gamma_\mu\big)}{(p-p_h)^2(q+p)^2} \\ + \frac{\mathrm{tr}\big((\not{q}+\not{p}-\not{p}_h)\gamma^\mu(\not{q}+\not{p})\gamma^\sigma\not{p}\gamma_\mu(\not{p}-\not{p}_h)\gamma^\rho\big)}{(q+p)^2(p-p_h)^2} \\ + \frac{\mathrm{tr}\big((\not{q}+\not{p}-\not{p}_h)\gamma^\mu(\not{q}+\not{p})\gamma^\sigma\not{p}\gamma^\rho(\not{q}+\not{p})\gamma_\mu\big)}{(q+p)^4}\Bigg]\end{aligned} \tag{D.63}$$

where, as always, we implicitly summed over outgoing and averaged over incoming spin states. The expression is again calculated with TRACER [292] and the scalar products given in equations (D.20). The result is then sorted in terms of the structure functions $\mathcal{A}_k$ defined in (4.24) and the overall delta distributions is used to replace factors of q_T^2 in denominators if possible, which results in the hadronic differential cross section in equation (4.47).

D.3.3 Gluon PDF and quark FF—the qg channel

We conclude this section with the channel including a gluon PDF and a quark FF. The partonic invariant amplitude is shown in figure D.4. The analytical expression reads

$$\begin{aligned}
\bar{u}_i^A(p_h)H_{i,A}^{a,\mu} &= \bar{u}(l_2)(-\mathrm{i}e\gamma^\rho)u(l_1)\left(\frac{-\mathrm{i}g_{\rho\sigma}}{q^2+\mathrm{i}\eta}\right) \\
&\quad\times \bar{u}_i^A(p_h)\Bigg[(-\mathrm{i}ee_q\gamma^\sigma)\frac{\mathrm{i}(\not{p}_h-\not{q})}{(p_h-q)^2+\mathrm{i}\eta}(-\mathrm{i}g_s\mu^\varepsilon\gamma^\mu t^a_{AB}) \\
&\qquad + (-\mathrm{i}g_s\mu^\varepsilon\gamma^\mu t^a_{AB})\frac{\mathrm{i}(\not{p}_h-\not{p})}{(p_h-p)^2+\mathrm{i}\eta}(-\mathrm{i}ee_q\gamma^\sigma)\Bigg]v^B(p-p_h+q) \\
&= -\frac{\mathrm{i}e^2e_qg_s\mu^\varepsilon}{Q^2}t^a_{AB}\bar{u}(l_2)\gamma_\sigma u(l_1) \\
&\quad\times \bar{u}_i^A(p_h)\left[\frac{\gamma^\sigma(\not{p}_h-\not{q})\gamma^\mu}{(p_h-q)^2+\mathrm{i}\eta}+\frac{\gamma^\mu(\not{p}_h-\not{p})\gamma^\sigma}{(p_h-p)^2+\mathrm{i}\eta}\right]v^B(p-p_h+q)
\end{aligned} \tag{D.64}$$

Like always we suppressed Dirac indices on the right-hand side of the equation for brevity. Further we replaced the placeholder variable $p_{\bar{q}}$ in figure D.4 by the means of the overall delta distribution $p_{\bar{q}} = p - p_h + q$. Inserting the expression in the trace of equation (4.60) yields

$$\begin{aligned}
\frac{-g_{\mu\nu}}{N_c^2-1}\operatorname{tr}\big(H^\mu H^{\dagger,\nu}\gamma^0\not{p}_h\big) &= -\frac{e^4e_q^2g_s^2\mu^{2\varepsilon}T_R}{2Q^4}\operatorname{tr}\big(\not{l}_2\gamma_\sigma\not{l}_1\gamma_\rho\big) \\
&\quad\times\Bigg[\frac{\operatorname{tr}\big(\not{p}_h\gamma^\sigma(\not{p}_h-\not{q})\gamma^\mu(\not{p}-\not{p}_h+\not{q})\gamma_\mu(\not{p}_h-\not{q})\gamma^\rho\big)}{(p_h-q)^4} \\
&\qquad+\frac{\operatorname{tr}\big(\not{p}_h\gamma^\sigma(\not{p}_h-\not{q})\gamma^\mu(\not{p}-\not{p}_h+\not{q})\gamma^\rho(\not{p}_h-\not{p})\gamma_\mu\big)}{(p_h-q)^2(p_h-p)^2} \\
&\qquad+\frac{\operatorname{tr}\big(\not{p}_h\gamma^\mu(\not{p}_h-\not{p})\gamma^\sigma(\not{p}-\not{p}_h+\not{q})\gamma_\mu(\not{p}_h-\not{q})\gamma^\rho\big)}{(p_h-p)^2(p_h-q)^2} \\
&\qquad+\frac{\operatorname{tr}\big(\not{p}_h\gamma^\mu(\not{p}_h\not{p})\gamma^\sigma(\not{p}-\not{p}_h+\not{q})\gamma^\rho(\not{p}_h-\not{p})\gamma_\mu\big)}{(p_h-p)^4}\Bigg]
\end{aligned} \tag{D.65}$$

where we summed over all outgoing and averaged over all incoming spin states. The color trace combined with the overall color prefactor gives in contrast to the other channels the Casimir operator of the fundamental representation

$$\frac{\operatorname{tr}\big(t^at^a\big)}{N_c^2-1} = T_R \tag{D.66}$$

The Dirac traces are again calculated by using TRACER [292] and the scalar products given in equations (D.20). Sorting the result in terms of the structure functions $\mathcal{A}_k$ defined in equations (4.24) and replacing q_T^2 in the denominator by means of the overall delta distribution yields the expression reported in equation (4.61).

D.4 Delta distribution expansion

In this section we will systematically expand the delta distribution that we encounter in our calculation, see equations (4.27), (4.47) and (4.61). The leading order in terms of small q_T has been used for years [164, 237], but to our knowledge it has never been expanded to higher orders. In our particular case this is necessary to gather all terms that diverge for small transverse momentum.

We perform the calculation in terms of the q_T^2 differential SIDIS process, where in our particular frame q_T characterizes the transverse momentum of the measured hadron. Compared to equations (4.27), (4.47) and (4.61) we change the delta distribution in terms of

$$\delta\left(\left(1-\frac{\xi}{x}\right)\left(1-\frac{\zeta}{z}\right)-\frac{q_T^2}{Q^2}\right)=xz\delta\left((\xi-x)(\zeta-z)-\frac{q_T^2}{\Lambda^2}\right) \tag{D.67}$$

with the abbreviation $\Lambda^2=Q^2/(xz)$, which does not depend on q_T. Further we notice that the integral limits are by means of the delta distribution (compare also to [237, 245, 246])

$$\int_x^1 \mathrm{d}\xi \int_z^1 \mathrm{d}\zeta \to \int_{\xi_\mathrm{min}}^1 \mathrm{d}\xi \int_{\zeta_\mathrm{min}}^1 \mathrm{d}\zeta \tag{D.68}$$

with

$$\xi_\mathrm{min}=\xi^*(1) \tag{D.69a}$$
$$\zeta_\mathrm{min}=\zeta^*(1) \tag{D.69b}$$

and

$$\xi^*(\zeta)=x\left(1+\frac{z}{\zeta-z}\frac{q_T^2}{Q^2}\right)=x+\frac{1}{\zeta-z}\frac{q_T^2}{\Lambda^2} \tag{D.70a}$$
$$\zeta^*(\xi)=z\left(1+\frac{x}{\xi-x}\frac{q_T^2}{Q^2}\right)=z+\frac{1}{\xi-x}\frac{q_T^2}{\Lambda^2} \tag{D.70b}$$

For the following calculation we use an arbitrary test function $f(\xi,\zeta)$ and consider the integral

$$I\equiv\int_{\xi_\mathrm{min}}^1 \mathrm{d}\xi \int_{\zeta_\mathrm{min}}^1 \mathrm{d}\zeta\, f(\xi,\zeta)\delta\left((\xi-x)(\zeta-z)-\frac{q_T^2}{\Lambda^2}\right) \tag{D.71}$$

In principle one could split up the integral, but because it is symmetric under $x\leftrightarrow z$ and $f(\xi,\zeta)\leftrightarrow f(\zeta,\xi)$ we perform the calculation antisymmetric and make the result symmetric later on.

First we use the delta distribution to get rid of one integration, then we take the taylor series to expand the test function at $q_T^2/\Lambda^2 = 0$

$$I = \int\limits_{\xi_{\min}}^{1} \frac{\mathrm{d}\xi}{\xi - x} f(\xi, \zeta^*(\xi)) = \sum_{n=0}^{\infty} \frac{I_n}{n!} \left(\frac{q_T^2}{\Lambda^2}\right)^n$$

where we defined the coefficients

$$I_n \equiv \int\limits_{\xi_{\min}}^{1} \frac{\mathrm{d}\xi}{(\xi - x)^{n+1}} \frac{\partial^n f(\xi, z)}{\partial \zeta^n} \tag{D.72}$$

The partial derivative is concerned to the second argument of the function $f(\xi, \zeta)$ and evaluated at $\zeta = z$. We handle I_n by applying n times integration by parts to reduce the power of the denominator

$$I_n = \frac{I_n^{(1)}}{n!} - \sum_{k=1}^{n} \frac{(k-1)!}{n!} \left[(\xi - x)^{-k} \frac{\partial^{2n-k} f(\xi, z)}{\partial \xi^{n-k} \partial \zeta^n} \right]_{\xi_{\min}}^{1} \tag{D.73}$$

with the definition

$$I_n^{(1)} = \int\limits_{\xi_{\min}}^{1} \frac{\mathrm{d}\xi}{\xi - x} \frac{\partial^{2n} f(\xi, z)}{\partial \xi^n \partial \zeta^n} \tag{D.74}$$

By inserting a zero

$$I_n^{(1)} = \frac{\partial^{2n} f(x, z)}{\partial \xi^n \partial \zeta^n} \int\limits_{\xi_{\min}}^{1} \frac{\mathrm{d}\xi}{\xi - x} + \int\limits_{\xi_{\min}}^{1} \frac{\mathrm{d}\xi}{\xi - x} \left[\frac{\partial^{2n} f(\xi, z)}{\partial \xi^n \partial \zeta^n} - \frac{\partial^{2n} f(x, z)}{\partial \xi^n \partial \zeta^n} \right] \tag{D.75}$$

we can easily integrate the first summand. For the second summand we split up the integral at x. Then we approximate the integral from $\xi_{\min}$ to x by evaluating at the upper limit and multiplying with the volume of the integral

$$\begin{aligned} I_n^{(1)} &= \frac{\partial^{2n} f(x, z)}{\partial \xi^n \partial \zeta^n} \ln\left(\frac{1 - x}{\xi_{\min} - x}\right) + \int\limits_{x}^{1} \frac{\mathrm{d}\xi}{\xi - x} \left[\frac{\partial^{2n} f(\xi, z)}{\partial \xi^n \partial \zeta^n} - \frac{\partial^{2n} f(x, z)}{\partial \xi^n \partial \zeta^n} \right] \\ &\quad + \int\limits_{\xi_{\min}}^{x} \frac{\mathrm{d}\xi}{\xi - x} \left[\frac{\partial^{2n} f(\xi, z)}{\partial \xi^n \partial \zeta^n} - \frac{\partial^{2n} f(x, z)}{\partial \xi^n \partial \zeta^n} \right] \\ &\approx \frac{\partial^{2n} f(x, z)}{\partial \xi^n \partial \zeta^n} \ln\left((1 - x)(1 - z)\frac{\Lambda^2}{q_T^2}\right) + \int\limits_{x}^{1} \frac{\mathrm{d}\xi}{\xi - x} \left[\frac{\partial^{2n} f(\xi, z)}{\partial \xi^n \partial \zeta^n} - \frac{\partial^{2n} f(x, z)}{\partial \xi^n \partial \zeta^n} \right] \end{aligned}$$

$$
\begin{aligned}
&+ (x-\xi_{\min}) \lim_{\xi\to x} \frac{1}{\xi - x}\left[\frac{\partial^{2n} f(\xi,z)}{\partial\xi^n\partial\zeta^n} - \frac{\partial^{2n} f(x,z)}{\partial\xi^n\partial\zeta^n}\right] \\
&= \frac{\partial^{2n} f(x,z)}{\partial\xi^n\partial\zeta^n} \ln\left((1-x)(1-z)\frac{\Lambda^2}{q_T^2}\right) + \int_x^1 \frac{\mathrm{d}\xi}{\xi - x}\left[\frac{\partial^{2n} f(\xi,z)}{\partial\xi^n\partial\zeta^n} - \frac{\partial^{2n} f(x,z)}{\partial\xi^n\partial\zeta^n}\right] \\
&\quad - \frac{1}{1-z}\frac{q_T^2}{\Lambda^2}\frac{\partial^{2n+1} f(x,z)}{\partial\xi^{n+1}\partial\zeta^n}
\end{aligned}
\tag{D.76}
$$

Note that this approximation is the only non-trivial during the whole calculation. While all other steps are easily extendable to produce arbitrary orders in terms of q_T^2 it would be necessary to find a reasonable approximation for the integral that supports higher orders and can be used in the further steps.

We can now collect all terms we found to far and extend the primitive integral in equation (D.73). For the expression at $\xi_{\min}$ we apply a Taylor series for the function $f(\xi_{\min}, z)$ at $q_T^2/\Lambda^2 = 0$

$$
\begin{aligned}
I \approx &\sum_{n=0}^{\infty} \frac{1}{(n!)^2}\left(\frac{q_T^2}{\Lambda^2}\right)^n \\
&\times \left[\frac{\partial^{2n} f(x,z)}{\partial\xi^n\partial\zeta^n} \ln\left((1-x)(1-z)\frac{\Lambda^2}{q_T^2}\right) + \int_x^1 \frac{\mathrm{d}\xi}{\xi - x}\left[\frac{\partial^{2n} f(\xi,z)}{\partial\xi^n\partial\zeta^n} - \frac{\partial^{2n} f(x,z)}{\partial\xi^n\partial\zeta^n}\right]\right. \\
&\quad - \frac{1}{1-z}\frac{q_T^2}{\Lambda^2}\frac{\partial^{2n+1} f(x,z)}{\partial\xi^{n+1}\partial\zeta^n} - \sum_{k=1}^{n}(k-1)!(1-x)^{-k}\frac{\partial^{2n-k} f(1,z)}{\partial\xi^{n-k}\partial\zeta^n} \\
&\quad \left. + \sum_{k=1}^{n}(k-1)!\left((1-z)\frac{\Lambda^2}{q_T^2}\right)^k \sum_{i=0}^{\infty}\frac{1}{i!}\left((1-z)\frac{\Lambda^2}{q_T^2}\right)^{-i}\frac{\partial^{2n-k+i} f(x,z)}{\partial\xi^{n-k+i}\partial\zeta^n}\right]
\end{aligned}
\tag{D.77}
$$

At this point we are able to identify all terms that contribute up to q_T^2/Λ^2. It is even possible to identify most terms that contribute to higher orders. But as already mentioned we might miss some terms due to the approximation in the previous step. We explain the identification that includes the triple sum, which is the most difficult. No term with $n = 0$ is valid, because the k sum gives zero. At $\mathcal{O}(1)$ and for $n \geq 1$ the only term that contributes at that order is with $k = n$ and $i = 0$. At order q_T^2/Λ^2 we have two non-zero contributions. The first one with $k = n$ and $i = 1$ and the second one with $k = n - 1$ (which implies $n \geq 2$) and $i = 0$. The resulting two sums have the similar structure in terms of x and z and can be unified easily

$$
\begin{aligned}
I \approx\ & f(x,z)\ln\left((1-x)(1-z)\frac{\Lambda^2}{q_T^2}\right) + \int_x^1 \frac{\mathrm{d}\xi}{\xi - x}\big[f(\xi,z) - f(x,z)\big] \\
&+ \sum_{n=1}^{\infty}\frac{(1-z)^n}{n\cdot n!}\frac{\partial^n f(x,z)}{\partial\zeta^n}
\end{aligned}
$$

$$+\frac{q_T^2}{\Lambda^2}\left[\frac{\partial^2 f(x,z)}{\partial\xi\partial\zeta}\ln\left((1-x)(1-z)\frac{\Lambda^2}{q_T^2}\right)+\int\limits_x^1\frac{\mathrm{d}\xi}{\xi-x}\left[\frac{\partial^2 f(\xi,z)}{\partial\xi\partial\zeta}-\frac{\partial^2 f(x,z)}{\partial\xi\partial\zeta}\right]\right.$$
$$-\frac{1}{1-z}\frac{\partial f(x,z)}{\partial\xi}-\frac{1}{1-x}\frac{\partial f(1,z)}{\partial\zeta}$$
$$\left.+\frac{\partial^2 f(x,z)}{\partial\xi\partial\zeta}+\sum_{n=2}^{\infty}\frac{(1-z)^{n-1}}{(n-1)\cdot n!}\frac{\partial^{n+1}f(x,z)}{\partial\xi\partial\zeta^n}\right] \tag{D.78}$$

The final task is to symmetresize the result. The $\mathcal{O}(1)$ terms can be modified by rewriting the sum via

$$\sum_{n=1}^{\infty}\frac{(1-z)^n}{n\cdot n!}\frac{\partial^n f(x,z)}{\partial\zeta^n}=\int\limits_z^1\frac{\mathrm{d}\zeta}{\zeta-z}\sum_{n=1}^{\infty}\frac{(\zeta-z)^n}{n!}\frac{\partial^n f(x,z)}{\partial\zeta^n}$$
$$=\int\limits_z^1\frac{\mathrm{d}\zeta}{\zeta-z}\big[f(x,\zeta)-f(x,z)\big] \tag{D.79}$$

where we identified the taylor series of $f(\zeta,x)$ in the second step. The same strategy applies to the sum that is left at $\mathcal{O}(q_T^2/\Lambda^2)$

$$\sum_{n=2}^{\infty}\frac{(1-z)^{n-1}}{(n-1)\cdot n!}\frac{\partial^{n+1}f(x,z)}{\partial\xi\partial\zeta^n}=\frac{\partial}{\partial\xi}\int\limits_z^1\frac{\mathrm{d}\zeta}{(\zeta-z)^2}\left[f(x,\zeta)-f(x,z)-(\zeta-z)\frac{\partial f(x,z)}{\partial\xi\partial\zeta}\right]$$
$$=\int\limits_z^1\frac{\mathrm{d}\zeta}{(\zeta-z)^2}\left[\frac{\partial f(x,\zeta)}{\partial\xi}-\frac{\partial f(x,z)}{\partial\xi}-(\zeta-z)\frac{\partial^2 f(x,z)}{\partial\xi\partial\zeta}\right] \tag{D.80}$$

The ξ integral can be manipulated such, that the symmetry becomes manifest. Note that our shorthand notation for the derivatives evaluated at a given value gets at its limit in the first step, where we introduce a term $\partial_\zeta f(x,z)$ that does not depend on ξ. To mark this explicit we denote this particular term by $g(z)\equiv f(x,z)$ and release the notation once no ambiguity is left

$$\int\limits_x^1\frac{\mathrm{d}\xi}{\xi-x}\left[\frac{\partial^2 f(\xi,z)}{\partial\xi\partial\zeta}-\frac{\partial^2 f(x,z)}{\partial\xi\partial\zeta}\right]$$
$$=\int\limits_x^1\frac{\mathrm{d}\xi}{\xi-x}\left[\frac{\partial}{\partial\xi}\left(\frac{\partial f(\xi,z)}{\partial\zeta}-\frac{\partial g(z)}{\partial\zeta}\right)-\frac{\partial^2 f(x,z)}{\partial\xi\partial\zeta}\right]$$
$$=\left[\frac{1}{\xi-x}\left(\frac{\partial f(\xi,z)}{\partial\zeta}-\frac{\partial g(z)}{\partial\zeta}\right)\right]_x^1$$

$$
\begin{aligned}
&+\int_x^1 \frac{\mathrm{d}\xi}{(\xi-x)^2}\left[\frac{\partial f(\xi,z)}{\partial\zeta}-\frac{\partial g(z)}{\partial\zeta}-(\xi-x)\frac{\partial^2 f(x,z)}{\partial\xi\partial\zeta}\right]\\
&=\frac{1}{1-x}\left(\frac{\partial f(1,z)}{\partial\zeta}-\frac{\partial f(x,z)}{\partial\zeta}\right)-\frac{\partial^2 f(x,z)}{\partial\xi\partial\zeta}\\
&\quad+\int_x^1 \frac{\mathrm{d}\xi}{(\xi-x)^2}\left[\frac{\partial f(\xi,z)}{\partial\zeta}-\frac{\partial f(x,z)}{\partial\zeta}-(\xi-x)\frac{\partial^2 f(x,z)}{\partial\xi\partial\zeta}\right] \qquad \text{(D.81)}
\end{aligned}
$$

This means our symmetric result is given by

$$
\begin{aligned}
I &= f(x,z)\ln\left(\frac{\Lambda^2}{q_T^2}\right)+\int_x^1 \mathrm{d}\xi\,\frac{f(\xi,z)}{(\xi-x)_+}+\int_z^1 \mathrm{d}\zeta\,\frac{f(x,\zeta)}{(\zeta-z)_+}\\
&\quad+\frac{q_T^2}{\Lambda^2}\left[\frac{\partial^2 f(x,z)}{\partial\xi\partial\zeta}\ln\left(\frac{\Lambda^2}{q_T^2}\right)+\int_x^1 \frac{\mathrm{d}\xi}{(\xi-x)_+^2}\frac{\partial f(\xi,z)}{\partial\zeta}+\int_z^1 \frac{\mathrm{d}\zeta}{(\zeta-z)_+^2}\frac{\partial f(x,\zeta)}{\partial\xi}\right]\\
&\quad+\mathcal{O}\left(\frac{q_T^4}{\Lambda^4}\right) \qquad \text{(D.82)}
\end{aligned}
$$

where we defined the generalized plus distribution

$$
\begin{aligned}
\int_a^1 \frac{\mathrm{d}y}{(y-a)_+^m}f(y) &\equiv \int_a^1 \frac{\mathrm{d}y}{(y-a)^m}\left(f(y)-\sum_{k=0}^{m-1}\frac{f^{(k)}(a)}{k!}(y-a)^k\right)\\
&\quad+\frac{f^{(m-1)}(a)}{(m-1)!}\ln(1-a)-\sum_{k=0}^{m-2}\frac{m-k-1}{k!}\frac{f^{(k)}(a)}{(1-a)^{m-k-1}} \qquad \text{(D.83)}
\end{aligned}
$$

In a distributional sense the result reads

$$
\begin{aligned}
&\delta\left((\xi-x)(\zeta-z)-\frac{q_T^2}{\Lambda^2}\right)\\
&=\delta(\xi-x)\delta(\zeta-z)\ln\left(\frac{\Lambda^2}{q_T^2}\right)+\frac{\delta(\xi-x)}{(\zeta-z)_+}+\frac{\delta(\zeta-z)}{(\xi-x)_+}\\
&\quad+\frac{q_T^2}{\Lambda^2}\left[\delta(\xi-x)\delta(\zeta-z)\partial_\xi\partial_\zeta\ln\left(\frac{\Lambda^2}{q_T^2}\right)+\frac{\delta(\xi-x)}{(\zeta-z)_+^2}\partial_\xi+\frac{\delta(\zeta-z)}{(\xi-x)_+^2}\partial_\zeta\right]+\mathcal{O}\left(\frac{q_T^4}{\Lambda^4}\right) \qquad \text{(D.84)}
\end{aligned}
$$

With the analogous calculation as in equation (D.81) we can reduce the plus distribution with $m=2$ into more convenient delta and plus distributions with $m=1$ by

$$
\int_a^1 \frac{\mathrm{d}y}{(y-a)_+^2}f(y)=\int_a^1 \frac{\mathrm{d}y}{(y-a)^2}\big(f(y)-f(a)-f'(a)(y-a)\big)+f'(a)\ln(1-a)-\frac{f(a)}{1-a}
$$

$$= \left[-\frac{f(y)-f(a)}{y-a}\right]_a^1 + \int_a^1 \frac{\mathrm{d}y}{y-a}\big(f'(y)-f'(a)\big) + f'(a)\ln(1-a) - \frac{f(a)}{1-a}$$
$$= f'(a) - \frac{f(1)}{1-a} + \int_a^1 \frac{\mathrm{d}y}{(y-a)_+} f(y) \tag{D.85}$$

or in a distributional sense

$$\frac{1}{(y-a)_+^2} = \delta(y-a)\partial_y - \frac{\delta(1-y)}{1-a} + \frac{1}{(y-a)_+}\partial_y \tag{D.86}$$

D.5 Regularization of transverse momentum distribution

In this section we will provide some details to the regularization procedure of the q_T^2 differential SIDIS cross section for $q_T^2 = 0$.

The most demanding case is the channel including quark PDFs and FFs, as it is the only one that includes divergent terms from virtual contributions in next-to-leading order. Our starting point will be equation (4.23), which includes the leading order and the virtual contributions to the next-to-leading order, and equation (4.30), which describes the next-to-leading-order contributions coming from real gluon emissions. First we recognize that only terms $\sim q_T^{-2}$ can contribute to the regularization, which excludes immediately all terms that originate from the higher-order expansion of the delta distribution which has been carried out in section D.4. Second we notice that all terms that come with the structure functions $\mathcal{A}_3$ and $\mathcal{A}_4$ do not include any singularities in terms of ε poles. Consequently only terms proportional to the structure functions $\mathcal{A}_1$ and $\mathcal{A}_2$ have to be regularized. Finally we divide the partonic cross sections that are of interest into singular and finite parts by defining

$$\hat{\sigma}_{qq,1,0}^{n=1,\mathrm{sing}} \equiv 2C_F \frac{Q^2}{q_T^2}\left(1 + \frac{\xi\zeta}{xz}\right) \tag{D.87a}$$
$$\hat{\sigma}_{qq,1,1}^{n=1,\mathrm{sing}} \equiv -2C_F \frac{Q^2}{q_T^2}\left(1 + \frac{\zeta}{z} + \frac{\xi}{x} - 4\left[\frac{x}{\xi} + \frac{z}{\zeta}\right] + 5\frac{xz}{\xi\zeta}\right) \tag{D.87b}$$
$$\hat{\sigma}_{qq,2,1}^{n=1,\mathrm{sing}} \equiv 2C_F \frac{Q^2}{q_T^2}\left(\frac{\xi\zeta}{xz} - 3\frac{xz}{\xi\zeta} + 4\left[\frac{x}{\xi} + \frac{z}{\zeta}\right] - 4\right) \tag{D.87c}$$
$$\hat{\sigma}_{qq,2,2}^{n=1,\mathrm{sing}} \equiv 2C_F \frac{Q^2}{q_T^2}\left(\frac{\xi\zeta}{xz} - \frac{xz}{\xi\zeta} - 2\left[\frac{\zeta}{z} + \frac{\xi}{x}\right] + 4\right) \tag{D.87d}$$

and

$$\hat{\sigma}_{qq,i,j}^{n=1,\mathrm{fin}} \equiv \hat{\sigma}_{qq,i,j}^{n=1} - \hat{\sigma}_{qq,i,j}^{n=1,\mathrm{sing}} \tag{D.88}$$

For the regularization procedure we follow closely [224] by using the distributional identities

$$\frac{1}{(q_T^2)^{1+\varepsilon}} = \frac{1}{(q_T^2)_+} + \delta(q_T^2)\left(-\frac{1}{\varepsilon} + \ln(Q_T^2)\right) + \mathcal{O}(\varepsilon) \tag{D.89a}$$

$$\frac{\ln(q_T^2)}{(q_T^2)^{1+\varepsilon}} = \left(\frac{\ln(q_T^2)}{q_T^2}\right)_+ + \delta(q_T^2)\left(-\frac{1}{\varepsilon^2} + \frac{1}{2}\ln^2(Q_T^2)\right) + \mathcal{O}(\varepsilon) \tag{D.89b}$$

where the plus distribution is defined in equation 4.32. Using this identities we can easily rewrite the singular terms

$$\begin{aligned}
\frac{\mathrm{d}\sigma_{qq}^{\mathrm{sing}}}{\mathrm{d}\mathcal{R}_d} &= \frac{\mathrm{d}\sigma_{qq}^{n=0}}{\mathrm{d}\mathcal{R}_d} + (4\pi)^4 \frac{\alpha^2\alpha_s}{2zQ^4}\Gamma(1-\varepsilon)\left(\frac{q_T^2}{4\pi}\right)^{\varepsilon} \int_x^1 \mathrm{d}\xi \int_z^1 \mathrm{d}\zeta \\
&\quad \times \left[\delta(\xi-x)\delta(\zeta-z)\ln\left(\frac{Q^2}{xzq_T^2}\right) + \frac{\delta(\xi-x)}{(\zeta-z)_+} + \frac{\delta(\zeta-z)}{(\xi-x)_+}\right] \\
&\quad \times \frac{1}{\Gamma(1-\varepsilon)}\left(\frac{4\pi\mu^2}{q_T^2}\right)^{\varepsilon} \frac{xz}{\xi\zeta} e_q^2 f_q(\xi) D_q(\zeta)\left[\mathcal{A}_1 \hat{\sigma}_{qq,1}^{n=1,\mathrm{sing}} + \mathcal{A}_2 \hat{\sigma}_{qq,2}^{n=1,\mathrm{sing}}\right] \\
&= \Gamma(1-\varepsilon)\left(\frac{q_T^2}{4\pi}\right)^{\varepsilon} \frac{2\pi(4\pi)^4\alpha^2}{zQ^2} \\
&\quad \times \Bigg[\delta(q_T^2)(\mathcal{A}_1 + \varepsilon\mathcal{A}_2) e_q^2 f_q(x) D_q(z) \\
&\qquad \times \left(1 - \frac{\alpha_s C_F}{2\pi}\left(\frac{4\pi\mu^2}{Q^2}\right)^{\varepsilon} \frac{\Gamma(1+\varepsilon)\Gamma^2(1-\varepsilon)}{\Gamma(1-2\varepsilon)}\left[\frac{2}{\varepsilon^2} + \frac{3}{\varepsilon} + 8\right]\right) \\
&\qquad + \frac{\alpha_s C_F}{2\pi} \int_x^1 \mathrm{d}\xi \int_z^1 \mathrm{d}\zeta \\
&\qquad \times \left(\delta(\xi-x)\delta(\zeta-z)\ln\left(\frac{Q^2}{xzq_T^2}\right) + \frac{\delta(\xi-x)}{(\zeta-z)_+} + \frac{\delta(\zeta-z)}{(\xi-x)_+}\right) \\
&\qquad \times \frac{1}{\Gamma(1-\varepsilon)}\left(\frac{4\pi\mu^2}{q_T^2}\right)^{\varepsilon} \frac{xz}{\xi\zeta} \frac{e_q^2 f_q(\xi) D_q(\zeta)}{2C_F Q^2}\left[\mathcal{A}_1 \hat{\sigma}_{qq,1}^{n=1,\mathrm{sing}} + \mathcal{A}_2 \hat{\sigma}_{qq,2}^{n=1,\mathrm{sing}}\right]\Bigg] \\
&= \Gamma(1-\varepsilon)\left(\frac{q_T^2}{4\pi}\right)^{\varepsilon} \frac{2\pi(4\pi)^4\alpha^2}{zQ^2}(\mathcal{A}_1 + \varepsilon\mathcal{A}_2) e_q^2 \\
&\quad \times \Bigg[\delta(q_T^2) f_q(x) D_q(z) \\
&\qquad \times \left(1 - \frac{\alpha_s C_F}{2\pi}\left(\frac{4\pi\mu^2}{Q^2}\right)^{\varepsilon} \frac{\Gamma(1+\varepsilon)\Gamma^2(1-\varepsilon)}{\Gamma(1-2\varepsilon)}\left[\frac{2}{\varepsilon^2} + \frac{3}{\varepsilon} + 8\right]\right)
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{(4\pi\mu^2)^\varepsilon}{\Gamma(1-\varepsilon)} \frac{\alpha_s C_F}{2\pi} f_q(x) D_q(z) \left(\delta(q_T^2) \left[\frac{2}{\varepsilon^2} - \ln^2(Q_T^2) \right] - 2 \left(\frac{\ln(q_T^2)}{q_T^2} \right)_+ \right) \\
&+ \frac{(4\pi\mu^2)^\varepsilon}{\Gamma(1-\varepsilon)} \frac{\alpha_s}{2\pi} \Big\{ f_q(x) \big[P_{qq} \otimes D_q(z) \big] + D_q(z) \big[f_q(x) \otimes P_{qq} \big] \\
&\qquad + C_F f_q(x) D_q(z) \big[2\ln(Q^2) - 3 \big] \Big\} \\
&\quad \times \left(\delta(q_T^2) \left[-\frac{1}{\varepsilon} + \ln(Q_T^2) \right] \right) \\
&+ C_F \delta(q_T^2) \left(D_q(z) \int_x^1 \frac{\mathrm{d}\xi}{\xi} f_q(\xi) \frac{\xi - x}{\xi} + f_q(x) \int_z^1 \frac{\mathrm{d}\zeta}{\zeta} D_q(\zeta) \frac{\zeta - z}{\zeta} \right) \Bigg]
\end{aligned} \tag{D.90}
$$

where we used the convolutions defined in equations (4.31) and the leading-order splitting function P_{qq}, which is defined by the equations (B.1), (1.87b) and (1.76b). Further the plus distribution has been modified to accommodate the terms $\sim \ln(x)$ and $\sim \ln(z)$ and is defined in equation (4.34). Note that we omitted already some terms of order $\mathcal{O}(\varepsilon)$ to shorten the expression. By similar reasoning we introduced terms of order $\mathcal{O}(\varepsilon)$ to factor out the structure functions. This concerns especially the last line coming with structure function $\mathcal{A}_2$. Note that for the whole expression the term $\hat{\sigma}_{qq,2,2}^{n=1,\mathrm{sing}}$ does not contribute at all, because it is zero for $\xi = x$ and $\zeta = z$.

To complete the regularization procedure we finally have to renormalize the PDFs and FFs via

$$
f_q^{\mathrm{ren}}(x) = f_q(x) - \frac{\alpha_s}{2\pi} \left(\frac{1}{\varepsilon} - \gamma_E + \ln(4\pi) \right) \big(\big[f_q(x) \otimes P_{qq} \big] + \big[f_g(x) \otimes P_{qg} \big] \big) \tag{D.91a}
$$

$$
D_q^{\mathrm{ren}}(z) = D_q(z) - \frac{\alpha_s}{2\pi} \left(\frac{1}{\varepsilon} - \gamma_E + \ln(4\pi) \right) \big(\big[P_{qq} \otimes D_q(z) \big] + \big[P_{gq} \otimes D_g(z) \big] \big) \tag{D.91b}
$$

which is in the $\overline{\mathrm{MS}}$ scheme. We will immediately drop the index 'ren' for brevity and replace the product of both functions by

$$
\begin{aligned}
f_q(x) D_q(z) \to\ & f_q(x) D_q(z) \\
&+ \frac{\alpha_s}{2\pi} \left(\frac{1}{\varepsilon} - \gamma_E + \ln(4\pi) \right) \Big(f_q(x) \big[P_{qq} \otimes D_q(z) \big] + D_q(z) \big[f_q(x) \otimes P_{qq} \big] \\
&\qquad + f_q(x) \big[P_{gq} \otimes D_g(z) \big] + D_q(z) \big[f_g(x) \otimes P_{qg} \big] \Big) \\
&+ \mathcal{O}(\alpha_s^2)
\end{aligned} \tag{D.92}
$$

Now we may expand the final expression in equation (D.90), where we use only the renormalization terms including P_{qq}. The other terms will be needed for the other channels. As expected all poles cancel and we can move from d to 4 dimensions by

setting $\varepsilon = 0$. Some reorganization very similar to the calculation in [224], including the usage of

$$\frac{\ln(Q^2)}{(q_T^2)_+} - \left(\frac{\ln(q_T^2)}{q_T^2}\right)_+ = \left(\frac{\ln\left(\frac{Q^2}{q_T^2}\right)}{q_T^2}\right)_+ \tag{D.93}$$

gives all $\delta(q_T^2)$ and all distribution parts for the divergent terms in the final results for a single flavour q given in equations (4.36), (4.51) and (4.64). The additional divergent terms and all finite terms are easily obtained by the terms that has not been considered yet. These are namely

(1) the finite parts $\hat{\sigma}_{qq,1,j}^{n=1,\text{fin}}$ and $\hat{\sigma}_{qq,2,j}^{n=1,\text{fin}}$ and the terms accompanying the structure functions $\mathcal{A}_3$ and $\mathcal{A}_4$ combined with the leading-order expansion of the delta distribution

(2) All terms originating from the next-to-leading-order terms of the delta distribution expansion. The terms $\sim \delta(1-\zeta)$ and $\sim \delta(1-\xi)$ give no finite contribution, because the PDFs as well as the FFs vanish at their threshold $f_q(1) = D_q(1) = 0$.

The regularization for the other channels is tremendously easier because they do not encounter virtual diagrams. Also the real diagrams to not contribute any $1/\varepsilon^2$ poles, because the possible sources (terms with $\ln(q_T^2)/q_T^{2+2\varepsilon}$) vanish by means of a delta distribution. For the gq channel it is due setting $\zeta = z$ while all terms in the qg channel vanish for $\xi = x$. The only poles that appear are canceled by the renormalization of the PDF and FF in equation (D.92).

Appendix E

Mellin transform

The Mellin transform is an integral transform closely related to the Fourier transform. It is named after the Finnish mathematician Hjalmar Mellin. The transform is defined by

$$f(N) = \mathcal{M}_f(N) = \int_0^1 \mathrm{d}x\, x^{N-1} f(x) \tag{E.1}$$

$f(N)$ is called the Mellin moment of the function $f(x)$. Sometimes it is denoted f^N to distinguish the appearance from $f(x)$. The relation to the Fourier transform $\mathcal{F}$is given by $\mathcal{M}_f(N) = \mathcal{F}_{f\circ\exp}(\mathrm{i}N)$, where the upper limit of the Mellin transform is infinity instead of one. For every use case in this thesis the integrand has the domain $\big[0,1\big]$ and we can therefore use 1 as upper limit. The inverse transformation is defined by an line integral in the complex plane, see figure E.1

$$f(x) = \mathcal{M}_f^{-1}(x) = \int_{\mathcal{C}_N} \frac{\mathrm{d}N}{2\pi\mathrm{i}} x^{-N} f(N) \tag{E.2}$$

where the contour $\mathcal{C}_N$ is usually chosen to be a straight line parallel to the imaginary axis

$$\begin{aligned} \mathcal{C}_N : \big(-\infty, \infty\big) &\to \mathbb{C} \\ t &\mapsto c + \mathrm{i}t \end{aligned} \tag{E.3}$$

where $c \in \mathbb{R}^+$. It has been shown that the inverse transformation exists only, if the following premises are met $(0 < a < c < b)$

(1) The transformation (E.1) converges absolutely and $f(N)$ is analytic in the set $\big\{N \in \mathbb{C} \,|\, a < \mathrm{Re}\big(N\big) < b\big\}$.

(2) $f(N) \to 0$ continuous for $N = c + \mathrm{i}t \to c \pm \mathrm{i}\infty$.

(3) $f(x)$ is a partwise continuous function.

If these are met, it is easy to show that the inverse transformation behaves as expected by closing the contour as indicated in figure E.1

$$f(M) = \int_0^1 \mathrm{d}x\, x^{M-1} \int_{\mathcal{C}_N} \frac{\mathrm{d}N}{2\pi\mathrm{i}} x^{-N} f(N)$$

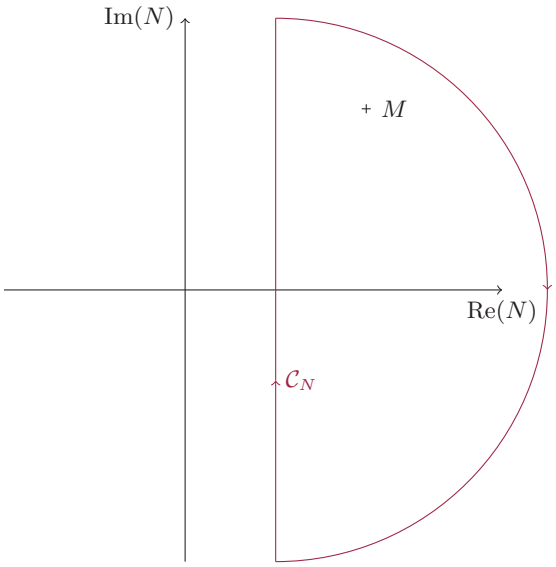


Figure E.1: Illustration of the inverse Mellin transform in equation (E.2) with the contour draws in red. The straight line represents the initial definition. The closing arc encloses the point M in the calculation (E.4). All other possible poles of the integrand are on the left-hand side of the contour.

$$
\begin{aligned}
&= \int_{\mathcal{C}_N} \frac{\mathrm{d}N}{2\pi\mathrm{i}} \frac{f(N)}{M-N} \\
&= (-1)\,\mathrm{Res}\left[-\frac{f(N)}{N-M}\right]_{N=M} \\
&= f(M)
\end{aligned} \tag{E.4}
$$

where we induced by the solution of the x integral, that for $\mathrm{Im}(N) = \mathrm{Im}(N)$ the real parts have to fulfill $\mathrm{Re}(N) < \mathrm{Re}(M)$, which means that the contour has to pass M on the left-hand side. This results in one Term in the use of the Residuum theorem. All possible other poles of the integrand are passed on the right-hand side and do therefore not contribute.

The key point why the Mellin transform is useful, is its property to convert convolutions into products

$$
\begin{aligned}
\mathcal{M}_{f\otimes g}(N) &= \int_0^1 \mathrm{d}x\, x^{N-1} \int_x^1 \frac{\mathrm{d}y}{y} f(y) g\left(\frac{x}{y}\right) \\
&= \int_0^1 \mathrm{d}x\, x^{N-1} \int_0^1 \mathrm{d}y \int_0^1 \mathrm{d}z\, f(y) g(z) \delta(x - yz)
\end{aligned}
$$

$$= \int_0^1 \mathrm{d}y\, y^{N-1} f(y) \int_0^1 \mathrm{d}z\, z^{N-1} g(z)$$
$$= \mathcal{M}_f(N)\mathcal{M}_g(N) \tag{E.5}$$

The most famous uses of these property is the analytical solution of the DGLAP equations (see section 1.5) and for resummation of threshold logarithms (see section 1.6), which also exponentiate in Mellin space.

Appendix F

Doolittle decomposition

The doolittle decomposition is a linear algebra method to decompose a matrix into two triangle matrices. It is also known as LU decomposition, as one of the matrices is a lower triangular matrix (L) and the other an upper triangular matrix (U). It is usually applied to solve linear equation systems, as the two result systems with the triangular matrices are solved easily.

Let A be an invertible matrix with rank n with the elements a_{ij}. The L matrix is a composition of Frobenius matrices L_m defined by the column m, which is applied to the result of the $m-1$ previous Frobenius matrices multiplied with A.

$$l_{im} = -\frac{a_{im}^{(n-1)}}{a_{mm}^{(n-1)}} \tag{F.1}$$

The upper index indicates that the elements in A alter every step. The procedure eliminates all entries in the lower half of A, transforming it into the upper triangle matrix U. In the meanwhile the product of the inverse matrices form the lower triangle matrix L.

$$A = (L_{n-1} \cdot \ldots \cdot L_1)^{-1}(L_{n-1} \cdot \ldots \cdot L_1 \cdot A) \equiv L \cdot U \tag{F.2}$$

The inverse of a Frobenius matrix is again a Frobenius matrix, where the elements l_{im} change their sign. The algorithm requires a non-zero diagonal, which can be achieved in every step by exchanging rows.

A linear equation system $Ax = y$ can therefore decomposed into two systems. The first step is to solve $Lz = y$, the second step consists of solving the system $Ux = z$. Both steps are exceptionally easy because both are triangle systems. The first is solved by

$$z_i = \frac{y_i}{(L)_{ii}} - \begin{cases} 0 & i = 1 \\ \sum_{j=1}^{i-1} \frac{(L)_{ij}}{(L)_{ii}} z_j & i \in \{2, \ldots, n\} \end{cases} \tag{F.3}$$

the second by

$$x_i = \frac{z_i}{(U)_{ii}} - \begin{cases} 0 & i = n \\ \sum_{j=i+1}^{n} \frac{(U)_{ij}}{(U)_{ii}} x_j & i \in \{1, \ldots, n-1\} \end{cases} \tag{F.4}$$

Numerically the first system can be solved along with the decomposition, because the linear equation system requires in every iteration only input that is already generated by the decomposition at the same iteration.

In chapter 5 we use the Doolittle composition for the linear equation system that determines the coefficients of cubic splines. This specific system embodies some simplifications. First the system is diagonal dominant, which implies that for the decomposition a row exchange is never needed. Second the only non-zero entries are on the diagonal and the two minor diagonals. This induces also the L (U) matrices to have only non-zero values in the diagonal and the lower (upper) minor diagonal. This greatly simplifies the numerical implementation of the decomposition as well as solving the resulting equation systems, because the sums over all non-diagonal elements collapse.

Appendix G

Analytical expressions of Sudakov form factor

In chapter 6 we discuss the numerical behavior of the Sudakov form factor depending on the underlying analytical form of the strong coupling constant α_s. The analytical expressions that are needed for this discussion are provided here. We define the quantities computed with equations (6.11) and (6.12), which are the analytical solutions of the renormalization group equation of the running coupling by specifying the corresponding coefficients A_i and B_i defined in equation (6.2). Further we indicate the order of the formula used for α_s. The results are

$$\begin{aligned} S_{A_1}^{\mathrm{LO}}(b_T, Q^2) &= -\frac{A_1}{\pi} \int_{\mu_b^2}^{Q^2} \frac{\mathrm{d}k_T^2}{k_T^2} \alpha_{s,\mathrm{LO}}(k_T^2) \ln\left(\frac{Q^2}{k_T^2}\right) \\ &= \frac{A_1}{\pi} \frac{1}{b_0} \left[\frac{\ln(L)}{\alpha_s(Q^2) b_0} + \ln\left(\frac{Q^2}{\mu_b^2}\right) \right] \end{aligned} \tag{G.1a}$$

$$S_{A_1}^{\mathrm{NLO}}(b_T, Q^2) = S_{A_1}^{\mathrm{LO}}(b_T, Q^2) + \frac{A_1}{\pi} \frac{b_1}{b_0^3} \left[\frac{1}{2} \ln^2(L) + \frac{1 + \ln(L)}{L} - 1 \right] \tag{G.1b}$$

$$S_{B_1}^{\mathrm{LO}}(b_T, Q^2) = \frac{B_1}{\pi} \frac{1}{b_0} \ln(L) \tag{G.1c}$$

$$S_{B_1}^{\mathrm{NLO}}(b_T, Q^2) = S_{B_1}^{\mathrm{LO}}(b_T, Q^2) - \frac{B_1}{\pi} \frac{b_1}{b_0^2} \frac{\alpha_s(Q^2)}{L} \big[L - 1 - \ln(L) \big] \tag{G.1d}$$

$$S_{A_2}^{\mathrm{LO}}(b_T, Q^2) = \frac{A_2}{\pi^2} \frac{1}{b_0^2} \frac{1}{L} \big[L - 1 - L \ln(L) \big] \tag{G.1e}$$

$$\begin{aligned} S_{A_2}^{\mathrm{NLO}}(b_T, Q^2) = S_{A_2}^{\mathrm{LO}}(b_T, Q^2) &+ \frac{A_2}{\pi^2} \frac{b_1 \alpha_s(Q^2)}{108 b_0^4 L^3} \\ &\times \big[(1 - L) \big\{ 54 b_0 (1 - 3L) L + \alpha_s(Q^2) b_1 (19 L (1 + L) - 8) \big\} \\ &\quad + 6 \ln(L) \big\{ 18 b_0 (1 - 2L) L + b_1 \alpha_s(Q^2) [9L - 4 + (9L - 6) \ln(L)] \big\} \big] \end{aligned} \tag{G.1f}$$

$$S_{B_2}^{\mathrm{LO}}(b_T, Q^2) = \frac{B_2}{\pi^2} \frac{\alpha_s^2(Q^2)}{L} \ln\left(\frac{\mu_b^2}{Q^2}\right) \tag{G.1g}$$

$$S_{B_2}^{\mathrm{NLO}}(b_T,Q^2) = S_{B_2}^{\mathrm{LO}}(b_T,Q^2) + \frac{B_2}{\pi^2}\frac{b_1\alpha_s^2(Q^2)}{54b_0^3L^3} \times \big[27b_0L(1-L^2) - 4b_1\alpha_s(Q^2)(1-L^3) - 6\ln(L)\big\{b_1\alpha_s(Q^2)(2+3\ln(L)) - 9b_0L\big\}\big] \tag{G.1h}$$

where we introduced the abbreviation

$$L \equiv 1 + \alpha_s(Q^2)b_0\ln\left(\frac{\mu_b^2}{Q^2}\right) \tag{G.2}$$

The identical calculation can also be performed using the expansion of α_s in terms of $\ln^{-1}(\mu/\Lambda_{\mathrm{qcd}})$, defined in equations (6.13) and (6.14) for NLO and LO respectively. We discriminate these formulas for the Sudakov factor with an additional superscript Λ_{qcd}. They are given by

$$S_{A_1}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}}(b_T,Q^2) = \frac{A_1}{\pi}\frac{1}{b_0}\left[\ln\left(\frac{Q^2}{\mu_b^2}\right) + L_\Lambda(Q^2)\ln\left(\frac{L_\Lambda(\mu_b^2)}{L_\Lambda(Q^2)}\right)\right] \tag{G.3a}$$

$$S_{A_1}^{\mathrm{NLO},\Lambda_{\mathrm{qcd}}}(b_T,Q^2) = S_{A_1}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}}(b_T,Q^2) + \frac{A_1}{\pi}\frac{b_1}{b_0^3} \times \left[\frac{L_\Lambda(Q^2)}{L_\Lambda(\mu_b^2)}\big(1+\ln(L_\Lambda(\mu_b^2))\big) - (1+\ln(L_\Lambda(Q^2))) + \frac{\ln^2(L_\Lambda(\mu_b^2)) - \ln^2(L_\Lambda(Q^2))}{2}\right] \tag{G.3b}$$

$$S_{B_1}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}}(b_T,Q^2) = \frac{B_1}{\pi}\frac{1}{b_0}\ln\left(\frac{L_\Lambda(\mu_b^2)}{L_\Lambda(Q^2)}\right) \tag{G.3c}$$

$$S_{B_1}^{\mathrm{NLO},\Lambda_{\mathrm{qcd}}}(b_T,Q^2) = S_{B_1}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}}(b_T,Q^2) + \frac{B_1}{\pi}\frac{b_1}{b_0^3}\left[\frac{1+\ln(L_\Lambda(\mu_b^2))}{L_\Lambda(\mu_b^2)} - \frac{1+\ln(L_\Lambda(Q^2))}{L_\Lambda(Q^2)}\right]$$

$$S_{A_2}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}}(b_T,Q^2) = \frac{A_2}{\pi^2}\frac{1}{b_0^2}\left[\ln\left(\frac{L_\Lambda(Q^2)}{L_\Lambda(\mu_b^2)}\right) - \frac{1}{L_\Lambda(\mu_b^2)}\ln\left(\frac{Q^2}{\mu_b^2}\right)\right] \tag{G.3d}$$

$$S_{A_2}^{\mathrm{NLO},\Lambda_{\mathrm{qcd}}}(b_T,Q^2) = S_{A_2}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}}(b_T,Q^2) + \frac{A_2}{\pi^2}\frac{b_1}{108b_0^6L_\Lambda^3(\mu_b^2)L_\Lambda^2(Q^2)} \times \Big[6L_\Lambda(\mu_b^2)\Big\{L_\Lambda^2(Q^2)\ln(L_\Lambda(\mu_b^2))\Big(5b_1 - 18b_0^2L_\Lambda(\mu_b^2)\Big) + 3b_1L_\Lambda^2(Q^2)\ln^2(L_\Lambda(\mu_b^2)) + L_\Lambda^2(\mu_b^2)\ln(L_\Lambda(Q^2)) \times \Big(18b_0^2L_\Lambda(Q^2) - 5b_1 - 3b_1\ln(L_\Lambda(Q^2))\Big)\Big\} + \ln\left(\frac{Q^2}{\mu_b^2}\right)\Big\{54b_0^2L_\Lambda(\mu_b^2)L_\Lambda(Q^2)(L_\Lambda(Q^2) - 3L_\Lambda(\mu_b^2))$$

$$
\begin{aligned}
&+ b_1(19L_\Lambda(\mu_b^2)(L_\Lambda(\mu_b^2) + L_\Lambda(Q^2)) - 8L_\Lambda^2(Q^2)) \\
&- 12L_\Lambda^2(Q^2)\ln(L_\Lambda(\mu_b^2)) \\
&\times \Big(2b_1 - 9b_0^2 L_\Lambda(\mu_b^2) + 3b_1\ln(L_\Lambda(\mu_b^2))\Big)\Big\}\Big]
\end{aligned}
$$

$$
S_{B_2}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}}(b_T, Q^2) = \frac{B_2}{\pi^2}\frac{1}{b_0^2}(L_\Lambda^{-1}(Q^2) - L_\Lambda^{-1}(\mu_b^2)) \tag{G.3e}
$$

$$
\begin{aligned}
S_{B_2}^{\mathrm{NLO},\Lambda_{\mathrm{qcd}}}(b_T, Q^2) = S_{B_2}^{\mathrm{LO},\Lambda_{\mathrm{qcd}}}(b_T, Q^2) &+ \frac{B_2}{\pi^2}\frac{b_1}{54b_0^6 L_\Lambda^3(\mu_b^2)L_\Lambda^3(Q^2)} \\
&\times \Big[27b_0^2 L_\Lambda(\mu_b^2)L_\Lambda(Q^2)(L_\Lambda^2(Q^2) - L_\Lambda^2(\mu_b^2)) \\
&+ 4b_1(L_\Lambda^3(\mu_b^2) - L_\Lambda^3(Q^2)) - 18b_1 L_\Lambda^3(Q^2)\ln^2(L_\Lambda(\mu_b^2)) \\
&- 6L_\Lambda^3(Q^2)\ln(L_\Lambda(\mu_b^2))\Big(2b_1 - 9b_0^2 L_\Lambda(\mu_b^2)\Big) \\
&+ 6L_\Lambda^3(\mu_b^2)\ln(L_\Lambda(Q^2))\Big(2b_1 - 9b_0^2 L_\Lambda(Q^2) + 3b_1\ln(L_\Lambda(Q^2))\Big)\Big]
\end{aligned} \tag{G.3f}
$$

where we introduced the abbreviation

$$
L_\Lambda(\mu^2) \equiv \ln\left(\frac{\mu^2}{\Lambda_{\mathrm{qcd}}^2}\right) \tag{G.4}
$$

Acknowledgments

First and foremost I would like to thank my supervisor *Prof. Dr. Werner Vogelsang* for giving me the opportunity to write my PhD thesis in his group. I am very grateful for the possibility to work on many different research topics and for giving me the freedom to follow my own projects. Furthermore I am thankful for facilitating my attendance at several summer schools and workshops to present parts of my research.

I am grateful to *Prof. Dr. Thomas Gutsche* for many advices and for sharing his tremendous wisdom during the lunch breaks. And of course I thank him for providing the second appraisal of this thesis.

I want to thank my collaborators during the time of my doctorate: *Dr. Marc Schlegel* for the very interesting insights in the methods used by the TMD community, which entered in the calculations concerned with the small transverse momentum resummation for semi inclusive deep inelastic scattering. *Julius Steiglechner* for a great time spent on a project presented in his master thesis and for sharing his knowledge about modern project management. *Richard Nies* for mastering an ancient Fortran code to deliver reliable test values for implementing the DGLAP evolution on the GPU.

Many thanks go to *Dr. Patriz Hinderer* for generating a great atmosphere in our shared office for almost four years. It has been a pleasure to talk and discuss with him about physics as well as everything else. I want also thank his successor *Fausto Frisenna*. Although we only spent a few month as office mates, I enjoyed this time as well and wish him good luck for his ongoing master thesis.

I want to express my gratitude to *Dr. Hannes Vogt*, who offered me the access to the GPUs of his group and roused the passion for well-designed software. The intensive discussions about C++ and CUDA have been enlightening. Additionally I thank him for constantly providing my workstation with up-to-date software.

Moreover I like to thank *Dr. Christoph Borschensky* for proofreading the first chapter of this thesis.

For the very same reason I have to thank *Dr. Davide Campagnari*, whom I am also grateful for many LaTeX related discussions, constantly reminding me what real dedication is about.

I thank *Sabine Werner* and *Ingrid Estiry* for their support in all organizational issues.

For an nice atmosphere at work I thank all members and regular guests of the seventh floor, including Dr. Daniele Anderle, Dr. Julien Baglio, Pit Burgbacher, Dr. Giuseppe Burgio, Gabriele Coniglio, Ehsan Ebadati, Dr. Jan Heffner, Felix Hekhorn, Dr. Ilkka Helenius, Prof. Dr. Barbara Jäger, Dr. Tom Kaufmann, Matthias Kesenheimer, Dr. Valery Lyubovitskij, Prof. Asmita Mukherjee, Dr. Junya Nakamura, Dr. Markus Quandt, Prof. Dr. Hugo Reinhardt, Dr. Felix Ringer, Dr. Marco Stratmann, Dr. Juraj Streicher,

Rouven Veigel, Marina Walt, Dr. Peter Watson. I like to emphasize *Dr. Lukas Salfelder*, who became a good friend during our jointly spent time, and *Dr. Peter Vastag*, who has been a good friend since the first semesters of our diploma studies in Tübingen.

Finally I like to thank my girlfriend *Franziska Schmidt*, my betrothed *Franziska Schmidt* and my wife *Franziska Lambertsen*, all of them enduring me many years and supporting me in every possible way.

List of Figures

List of Tables

Bibliography

[1] F. Englert and R. Brout, 'Broken Symmetry and the Mass of Gauge Vector Mesons', Physical Review Letters **13**, 321–323 (1964).

[2] P. W. Higgs, 'Broken Symmetries and the Masses of Gauge Bosons', Physical Review Letters **13**, 508–509 (1964).

[3] G. S. Guralnik, C. R. Hagen, and T. W. B. Kibble, 'Global Conservation Laws and Massless Particles', Physical Review Letters **13**, 585–587 (1964).

[4] C. N. Yang and R. L. Mills, 'Conservation of Isotopic Spin and Isotopic Gauge Invariance', Physical Review **96**, 191–195 (1954).

[5] O. W. Greenberg, 'Spin and unitary-spin independence in a paraquark model of baryons and mesons', Physical Review Letters **13**, 598–602 (1964).

[6] M. Y. Han and Y. Nambu, 'Three-triplet model with double $SU(3)$ symmetry', Physical Review **139**, B1006–B1010 (1965).

[7] H. Fritzsch and M. Gell-Mann, 'Current algebra: Quarks and what else?', eConf **C720906V2**, 135–165 (1972), arXiv:`hep-ph/0208010`.

[8] H. Fritzsch, M. Gell-Mann, and H. Leutwyler, 'Advantages of the color octet gluon picture', Physics Letters B **47**, 365–368 (1973).

[9] M. Lambertsen, 'Studies of the Evolution of Transverse Momentum dependent Parton Distributions' (Eberhard Karls Universität Tübingen, 2013), 171 pp.

[10] M. Gell-Mann, 'Symmetries of baryons and mesons', Physical Review **125**, 1067–1084 (1962).

[11] E. S. Abers and B. W. Lee, 'Gauge theories', Physics Reports **9**, 1–2 (1973).

[12] L. Faddeev and V. Popov, 'Feynman diagrams for the Yang-Mills field', Physics Letters B **25**, 29–30 (1967).

[13] J. Polchinski, 'Renormalization and effective lagrangians', Nuclear Physics B **231**, 269–295 (1984).

[14] W. Pauli and F. Villars, 'On the Invariant Regularization in Relativistic Quantum Theory', Reviews of Modern Physics **21**, 434–444 (1949).

[15] G. 't Hooft and M. Veltman, 'Regularization and renormalization of gauge fields', Nuclear Physics B **44**, 189–213 (1972).

[16] C. G. Bollini and J. J. Giambiagi, 'Dimensional renorinalization : The number of dimensions as a regularizing parameter', Il Nuovo Cimento B (1971-1996) **12**, 20–26 (1972).

[17] G. 't Hooft, 'Lattice regularization of gauge theories without loss of chiral symmetry', Physics Letters B **349**, 491–498 (1995), arXiv:hep-th/9411228.

[18] R. Pittau, 'A four-dimensional approach to quantum field theories', Journal of High Energy Physics **2012**, 151 (2012), arXiv:1208.5457.

[19] G. 't Hooft, 'Renormalization of massless Yang-Mills fields', Nuclear Physics B **33**, 173–199 (1971).

[20] G. 't Hooft, 'Renormalizable Lagrangians for massive Yang-Mills fields', Nuclear Physics B **35**, 167–188 (1971).

[21] D. A. Ross and J. C. Taylor, 'Renormalization of a unified theory of weak and electromagnetic interactions', Nuclear Physics B **51**, 125–144 (1973).

[22] D. A. Ross and J. C. Taylor, 'Erratum: Renormalization of a unified theory of weak and electromagnetic interactions', Nuclear Physics B **58**, 643 (1973).

[23] W. A. Bardeen, A. J. Buras, D. W. Duke, and T. Muta, 'Deep-inelastic scattering beyond the leading order in asymptotically free gauge theories', Physical Review D **18**, 3998–4017 (1978).

[24] G. 't Hooft, 'Dimensional regularization and the renormalization group', Nuclear Physics B **61**, 455–468 (1973).

[25] L. Salfelder, 'Processes with weak gauge boson pairs at hadron colliders : precise predictions and future prospects' (Eberhard Karls Universität Tübingen, 2016), 186 pp.

[26] S. Weinberg, 'New Approach to the Renormalization Group', Physical Review D **8**, 3497–3509 (1973).

[27] O. V. Tarasov, A. A. Vladimirov, and A. Y. Zharkov, 'The gell-mann-low function of QCD in the three-loop approximation', Physics Letters B **93**, 429–432 (1980).

[28] S. A. Larin and J. A. M. Vermaseren, 'The three-loop QCD β-function and anomalous dimensions', Physics Letters B **303**, 334–336 (1993), arXiv:hep-ph/9302208.

[29] T. van Ritbergen, J. A. M. Vermaseren, and S. A. Larin, 'The four-loop β-function in Quantum Chromodynamics', Physics Letters B **400**, 379–384 (1997), arXiv:hep-ph/9701390.

[30] M. Czakon, 'The four-loop QCD β-function and anomalous dimensions', Nuclear Physics B **710**, 485–498 (2005), arXiv:hep-ph/0411261.

[31] T. Luthe, A. Maier, P. Marquard, and Y. Schröder, 'Towards the five-loop Beta function for a general gauge group', Journal of High Energy Physics **2016**, 127 (2016), arXiv:1606.08662.

[32] W. Vogelsang, 'Quantenfeldtheorie und Elementarteilchenphysik', 2013.

[33] D. J. Gross and F. Wilczek, 'Ultraviolet behavior of non-abelian gauge theories', Physical Review Letters **30**, 1343–1346 (1973).

[34] H. D. Politzer, 'Reliable Perturbative Results for Strong Interactions?', Physical Review Letters **30**, 1346–1349 (1973).

[35] H. D. Politzer, 'Asymptotic freedom: An approach to strong interactions', Physics Reports **14**, 129–180 (1974).

[36] C. Patrignani et al. (Particle Data Group), 'Review of Particle Physics', Chinese Physics C **40**, 100001 (2016).

[37] S. Bethke, 'World Summary of α_s (2012)', Nuclear Physics B - Proceedings Supplements **234**, 229–234 (2013), arXiv:1210.0325.

[38] S. Bethke, G. Dissertori, and G. Salam, 'World Summary of α_s (2015)', EPJ Web of Conferences **120**, 07005 (2016).

[39] A. Sternbeck, E.-M. Ilgenfritz, M. Müller-Preussker, and A. Schiller, 'Towards the infrared limit in $SU(3)$ Landau gauge lattice gluodynamics', Physical Review D **72**, 014507 (2005), arXiv:hep-lat/0506007.

[40] C. S. Fischer, 'Infrared properties of QCD from Dyson–Schwinger equations', Journal of Physics G: Nuclear and Particle Physics **32**, R253 (2006), arXiv:hep-ph/0605173.

[41] C. Berger et al. (PLUTO Collaboration), 'Jet analysis of the $\Upsilon(9.46)$ decay into charged hadrons', Physics Letters B **82**, 449–455 (1979).

[42] C. Berger et al. (PLUTO Collaboration), 'Topology of the Υ-decay', Zeitschrift für Physik C Particles and Fields **8**, 101–114 (1981).

[43] D. E. Soper, in *Flavor Physics for the Millennium — Proceedings of the Theoretical Advanced Study Institute in Elementary Particle Physics* (WORLD SCIENTIFIC, 09/01/2001), pp. 267–316, arXiv:hep-ph/0011256.

[44] E. Farhi, 'Quantum Chromodynamics Test for Jets', Physical Review Letters **39**, 1587–1588 (1977).

[45] G. Sterman and S. Weinberg, 'Jets from Quantum Chromodynamics', Physical Review Letters **39**, 1436–1439 (1977).

[46] G. C. Blazey et al., 'Run II Jet Physics: Proceedings of the Run II QCD and Weak Boson Physics Workshop', QCD and weak boson physics in Run II. Proceedings **Batavia, USA, March 4-6, June 3-4, November 4-6, 1999**, 47–77 (2000), arXiv:hep-ex/0005012.

[47] S. D. Ellis and D. E. Soper, 'Successive combination jet algorithm for hadron collisions', Physical Review D **48**, 3160–3166 (1993), arXiv:hep-ph/9305266.

[48] S. Catani, Y. L. Dokshitzer, M. H. Seymour, and B. R. Webber, 'Longitudinally-invariant $k_\perp$-clustering algorithms for hadron-hadron collisions', Nuclear Physics B **406**, 187–224 (1993).

[49] Y. L. Dokshitzer, G. D. Leder, S. Moretti, and B. R. Webber, 'Better jet clustering algorithms', Journal of High Energy Physics **1997**, 001 (1997), arXiv:hep-ph/9707323.

[50] F. M. Ringer, 'Threshold Resummation and Higher Order Effects in QCD' (Eberhard Karls Universität Tübingen, 06/2015), 253 pp.

[51] P. Hinderer, 'Higher Order Studies in Perturbative QCD' (Eberhard Karls Universität Tübingen, 2017), 238 pp.

[52] S. D. Drell and T.-M. Yan, 'Massive Lepton-Pair Production in Hadron-Hadron Collisions at High Energies', Physical Review Letters **25**, 316–320 (1970).

[53] S. D. Drell and T.-M. Yan, 'Erratum: Massive Lepton-Pair Production in Hadron-Hadron Collisions at High Energies', Physical Review Letters **25**, 902–902 (1970).

[54] S. D. Drell and T.-M. Yan, 'Partons and their applications at high energies', Annals of Physics **66**, 578–623 (1971).

[55] W. J. den Dunnen, 'Polarization effects in proton-proton collisions within the Standard Model and beyond' (University of Amsterdam, 02/15/2013), 167 pp.

[56] R. P. Feynman, 'Very high-energy collisions of hadrons', Physical Review Letters **23**, 1415–1417 (1969).

[57] J. D. Bjorken and E. A. Paschos, 'Inelastic electron-proton and γ-proton scattering and the structure of the nucleon', Physical Review **185**, 1975–1982 (1969).

[58] D. Boer and P. J. Mulders, 'Color gauge invariance in the Drell–Yan process', Nuclear Physics B **569**, 505–526 (2000), arXiv:hep-ph/9906223.

[59] D. Boer, S. J. Brodsky, and D. S. Hwang, 'Initial-state interactions in the unpolarized Drell-Yan process', Physical Review D **67**, 054003 (2003), arXiv:hep-ph/0211110.

[60] A. V. Belitsky, X. Ji, and F. Yuan, 'Final state interactions and gauge invariant parton distributions', Nuclear Physics B **656**, 165–198 (2003), arXiv:hep-ph/0208038.

[61] M. G. A. Buffing and P. J. Mulders, 'Gauge links for transverse momentum dependent correlators at tree-level', Journal of High Energy Physics **2011**, 65 (2011), arXiv:1105.4804.

[62] J. C. Collins, *Foundations of perturbative QCD*, 1st ed. (Cambridge University Press, 2011).

[63] J. C. Collins, 'New definition of TMD parton densities', International Journal of Modern Physics: Conference Series **4**, 85–96 (2011), arXiv:1107.4123.

[64] P. Sun and F. Yuan, 'Transverse momentum dependent evolution: Matching semi-inclusive deep inelastic scattering processes to Drell-Yan and W/Z boson production', Physical Review D **88**, 114012 (2013), arXiv:1308.5003.

[65] J. P. Ma and G. P. Zhang, 'QCD corrections of all structure functions in transverse momentum dependent factorization for Drell-Yan processes', Journal of High Energy Physics, 100 (2014), arXiv:1308.2044.

[66] S. Melis, 'Phenomenology of TMDs', EPJ Web of Conferences **85**, edited by U. D'Alesio and F. Murgia, 01001 (2015), arXiv:1412.1719.

[67] U. D'Alesio, M. G. Echevarria, S. Melis, and I. Scimemi, 'TMDs: Evolution, modeling, precision', EPJ Web of Conferences **85**, edited by U. D'Alesio and F. Murgia, 02003 (2015), arXiv:1410.4522.

[68] C. Lorcé, B. Pasquini, and P. Schweitzer, 'Unpolarized transverse momentum dependent parton distribution functions beyond leading twist in quark models', Journal of High Energy Physics, 103 (2015), arXiv:1411.2550.

[69] J. Collins and T. Rogers, 'Understanding the large-distance behavior of transverse-momentum-dependent parton densities and the Collins-Soper evolution kernel', Physical Review D **91**, 074020 (2015), arXiv:1412.3820.

[70] M. G. A. Buffing, A. Mukherjee, and P. J. Mulders, 'Universality of TMD correlators', EPJ Web of Conferences **85**, edited by U. D'Alesio and F. Murgia, 02001 (2015), arXiv:1409.7257.

[71] D. Sivers, 'Single-spin production asymmetries from the hard scattering of pointlike constituents', Physical Review D **41**, 83–90 (1990).

[72] D. Sivers, 'Hard-scattering scaling laws for single-spin production asymmetries', Physical Review D **43**, 261–263 (1991).

[73] J. C. Collins, 'Fragmentation of Transversely Polarized Quarks Probed in Transverse Momentum Distributions', Nuclear Physics B **396**, 161–182 (1992), arXiv:hep-ph/9208213.

[74] J. C. Collins, 'Leading-twist single-transverse-spin asymmetries: Drell-Yan and deep-inelastic scattering', Physics Letters B **536**, 43–48 (2002), arXiv:hep-ph/0204004.

[75] D. Boer, P. J. Mulders, and F. Pijlman, 'Universality of T-odd effects in single spin and azimuthal asymmetries', Nuclear Physics B **667**, 201–241 (2003), arXiv:hep-ph/0303034.

[76] M. Anselmino et al., 'Extracting the Sivers function from polarized semi-inclusive deep inelastic scattering data and making predictions', Physical Review D **72**, 094007 (2005), arXiv:hep-ph/0507181.

[77] S. M. Aybat, J. C. Collins, J. W. Qiu, and T. C. Rogers, 'QCD evolution of the Sivers function', Physical Review D **85**, 034043 (2012), arXiv:1110.6428.

[78] D. Boer, 'TMD evolution of the Sivers asymmetry', Nuclear Physics B **874**, 217–229 (2013), arXiv:1304.5387.

[79] M. Anselmino, M. Boglione, J. O. G. H., S. Melis, and A. Prokudin, 'Unpolarised Transverse Momentum Dependent Distribution and Fragmentation Functions from SIDIS Multiplicities', Journal of High Energy Physics, 005 (2014), arXiv:1312.6261.

[80] M. G. Echevarria, A. Idilbi, Z.-B. Kang, and I. Vitev, 'QCD Evolution of the Sivers Asymmetry', Physical Review D **89**, 074013 (2014), arXiv:1401.5078.

[81] J. C. Collins and D. E. Soper, 'Parton distribution and decay functions', Nuclear Physics B **194**, 445–492 (1982).

[82] J. Gao et al., 'CT10 next-to-next-to-leading order global analysis of QCD', Physical Review D **89**, 033009 (2014), arXiv:1302.6246.

[83] J. C. Collins, D. E. Soper, and G. F. Sterman, 'Factorization of Hard Processes in QCD', Adv.Ser.Direct.High Energy Phys. **5**, 1–91 (1988), arXiv:hep-ph/0409313.

[84] X. Ji, 'Parton Physics on a Euclidean Lattice', Physical Review Letters **110**, 262002 (2013), arXiv:1305.1539.

[85] Y.-Q. Ma and J.-W. Qiu, 'Extracting Parton Distribution Functions from Lattice QCD Calculations', (2014), arXiv:1404.6860.

[86] J. Pumplin et al., 'New Generation of Parton Distributions with Uncertainties from Global QCD Analysis', Journal of High Energy Physics, 012 (2002), arXiv:hep-ph/0201195.

[87] H. L. Lai et al., 'New parton distributions for collider physics', Physical Review D **82**, 074024 (2010), arXiv:1007.2241.

[88] S. Dulat et al., 'New parton distribution functions from a global analysis of quantum chromodynamics', Physical Review D **93**, 033006 (2016), arXiv:1506.07443.

[89] A. D. Martin, W. J. Stirling, R. S. Thorne, and G. Watt, 'Parton distributions for the LHC', The European Physical Journal C **63**, 189–285 (2009), arXiv:0901.0002.

[90] L. A. Harland-Lang, A. D. Martin, P. Motylinski, and R. S. Thorne, 'Parton distributions in the LHC era: MMHT 2014 PDFs', The European Physical Journal C **75**, 1–53 (2015), arXiv:1412.3989.

[91] L. D. Debbio, S. Forte, J. I. Latorre, A. Piccione, and J. Rojo (NNPDF Collaboration), 'Neural network determination of parton distributions: the nonsinglet case', Journal of High Energy Physics **2007**, 039 (2007), arXiv:hep-ph/0701127.

[92] R. D. Ball et al. (NNPDF Collaboration), 'Parton distributions with LHC data', Nuclear Physics B **867**, 244–289 (2013), arXiv:1207.1303.

[93] S. L. Adler, 'Sum Rules Giving Tests of Local Current Commutation Relations in High-Energy Neutrino Reactions', Physical Review **143**, 1144–1155 (1966).

[94] D. J. Gross and C. H. L. Smith, 'High-energy neutrino-nucleon scattering, current algebra and partons', Nuclear Physics B **14**, 337–347 (1969).

[95] I. Hinchliffe and A. Kwiatkowski, 'Parton-model sum rules', Annual Review of Nuclear and Particle Science **46**, 609–645 (1996), arXiv:hep-ph/9604210.

[96] J. C. Collins and D. E. Soper, 'The Theorems of Perturbative QCD', Annual Review of Nuclear and Particle Science **37**, 383–409 (1987).

[97] X. Ji, J. P. Ma, and F. Yuan, 'QCD factorization for semi-inclusive deep-inelastic scattering at low transverse momentum', Physical Review D **71**, 034005 (2005), arXiv:hep-ph/0404183.

[98] X. Ji, J. P. Ma, and F. Yuan, 'QCD factorization for spin-dependent cross sections in DIS and Drell-Yan processes at low transverse momentum', Physics Letters B **597**, 299–308 (2004), arXiv:hep-ph/0405085.

[99] G. F. Sterman, *An Introduction to Quantum Field Theory* (Cambridge University Press, 08/1993), 592 pp.

[100] A. Vogt, 'Efficient evolution of unpolarized and polarized parton distributions with QCD-Pegasus', Computer Physics Communications **170**, 65–92 (2005), arXiv:hep-ph/0408244.

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

[102] G. Altarelli and G. Parisi, 'Asymptotic freedom in parton language', Nuclear Physics B **126**, 298–318 (1977).

[103] Y. L. Dokshitzer, 'Calculation of the Structure Functions for Deep Inelastic Scattering and e^+e^- Annihilation by Perturbation Theory in Quantum Chromodynamics.', Sov.Phys.JETP **46**, 641–653 (1977).

[104] E. G. Floratos, D. A. Ross, and C. T. Sachrajda, 'Higher-order effects in asymptotically free gauge theories: The anomalous dimensions of Wilson operators', Nuclear Physics B **129**, 66–88 (1977).

[105] E. G. Floratos, D. A. Ross, and C. T. Sachrajda, 'Erratum: Higher-order effects in asymptotically free gauge theories: the anomalous dimensions of Wilson operators [Nucl. Phys. B129 (1977) 66]', Nuclear Physics B **139**, 545–546 (1978).

[106] G. Altarelli, R. K. Ellis, and G. Martinelli, 'Large perturbative corrections to the Drell-Yan process in QCD', Nuclear Physics B **157**, 461–497 (1979).

[107] E. G. Floratos, D. A. Ross, and C. T. Sachrajda, 'Higher-order effects in asymptotically free gauge theories', Nuclear Physics B **152**, 493–520 (1979).

[108] A. González-Arroyo, C. López, and F. J. Ynduráin, 'Second-order contributions to the structure functions in deep inelastic scattering (I). Theoretical calculations', Nuclear Physics B **153**, 161–186 (1979).

[109] G. Curci, W. Furmanski, and R. Petronzio, 'Evolution of parton densities beyond leading order', Nuclear Physics B **175**, 27–92 (1980).

[110] W. Furmanski and R. Petronzio, 'Singlet parton densities beyond leading order', Physics Letters B **97**, 437–442 (1980).

[111] E. G. Floratos, R. Lacaze, and C. Kounnas, 'Space and time like cut vertices in QCD beyond the leading order—non-singlet sector', Physics Letters B **98**, 89–95 (1981).

[112] E. G. Floratos, R. Lacaze, and C. Kounnas, 'Space- and time-like cut vertices in QCD beyond the leading order—The singlet sector', Physics Letters B **98**, 285–290 (1981).

[113] E. G. Floratos, C. Kounnas, and R. Lacaze, 'Higher order QCD effects in inclusive annihilation and deep inelastic scattering', Nuclear Physics B **192**, 417–462 (1981).

[114] R. Hamberg and W. L. van Neerven, 'The correct renormalization of the gluon operator in a covariant gauge', Nuclear Physics B **379**, 143–171 (1992).

[115] W. Vogelsang, 'Rederivation of the Spin-dependent Next-to-leading Order Splitting Functions', Physical Review D **54**, 2023–2029 (1995), arXiv:hep-ph/9512218.

[116] R. K. Ellis and W. Vogelsang, 'The evolution of parton distributions beyond leading order: the singlet case', (1996), arXiv:hep-ph/9602356.

[117] W. Vogelsang, 'The spin-dependent two-loop splitting functions', Nuclear Physics B **475**, 47–72 (1996), arXiv:hep-ph/9603366.

[118] W. Vogelsang, 'The Polarized Two-loop Splitting Functions', Nuclear Physics B **475**, 47–72 (1996), arXiv:hep-ph/9607223.

[119] S. Moch, J. A. M. Vermaseren, and A. Vogt, 'The three-loop splitting functions in QCD: the non-singlet case', Nuclear Physics B **688**, 101–134 (2004), arXiv:hep-ph/0403192.

[120] A. Vogt, S. Moch, and J. A. M. Vermaseren, 'The three-loop splitting functions in QCD: the singlet case', Nuclear Physics B **691**, 129–181 (2004), arXiv:hep-ph/0404111.

[121] S. Moch, J. A. M. Vermaseren, and A. Vogt, 'The three-loop splitting functions in QCD: The helicity-dependent case', Nuclear Physics B **889**, 351–400 (2014), arXiv:1409.5131.

[122] R. K. Ellis, Z. Kunszt, and E. M. Levin, 'The evolution of parton distributions at small x', Nuclear Physics B **420**, 517–549 (1994).

[123] R. K. Ellis, Z. Kunszt, and E. M. Levin, 'Erratum: The evolution of parton distributions at small x [Nucl. Phys. B 420, 517 (1994)]', Nuclear Physics B **433**, 498 (1995).

[124] J. Blümlein et al., 'A Detailed comparison of NLO QCD evolution codes', in Future Physics at HERA (09/18/1996), p. 199, arXiv:hep-ph/9609400.

[125] G. F. Sterman, 'Summation of large corrections to short-distance hadronic cross sections', Nuclear Physics B **281**, 310–364 (1987).

[126] S. Catani and L. Trentadue, 'Resummation of the QCD perturbative series for hard processes', Nuclear Physics B **327**, 323–352 (1989).

[127] E. Laenen, 'Resummation for observables at TeV colliders', Pramana **63**, 1225 (2004).

[128] E. Laenen, 'Resummation in QCD in 4 lectures', Saint-Jacut-de-la-Mer, 06/2017.

[129] S. Coleman and R. E. Norton, 'Singularities in the physical region', Il Nuovo Cimento (1955-1965) **38**, 438–442 (1965).

[130] G. T. Bodwin, 'Factorization of the Drell-Yan cross section in perturbation theory', Physical Review D **31**, 2616–2642 (1985).

[131] G. T. Bodwin, 'Erratum: Factorization of the Drell-yan cross section in perturbation theory', Physical Review D **34**, 3932–3932 (1986).

[132] J. C. Collins, D. E. Soper, and G. Sterman, 'Factorization for short distance hadron-hadron scattering', Nuclear Physics B **261**, 104–142 (1985).

[133] J. C. Collins, D. E. Soper, and G. Sterman, 'Soft gluons and factorization', Nuclear Physics B **308**, 833–856 (1988).

[134] H. Contopanagos, E. Laenen, and G. Sterman, 'Sudakov factorization and resummation', Nuclear Physics B **484**, 303–327 (1997), arXiv:`hep-ph/9604313`.

[135] C. J. Borschensky, 'Precision calculations for coloured supersymmetric particle production at the Large Hadron Collider' (Westfälische Wilhelms-Universität Münster, 2016), 249 pp.

[136] J. G. M. Gatheral, 'Exponentiation of eikonal cross sections in nonabelian gauge theories', Physics Letters B **133**, 90–94 (1983).

[137] J. Frenkel and J. C. Taylor, 'Non-abelian eikonal exponentiation', Nuclear Physics B **246**, 231–245 (1984).

[138] A. Vogt, 'Next-to-next-to-leading logarithmic threshold resummation for deep-inelastic scattering and the Drell–Yan process', Physics Letters B **497**, 228–234 (2001), arXiv:`hep-ph/0010146`.

[139] S. Catani, M. L. Mangano, P. Nason, and L. Trentadue, 'The Resummation of Soft Gluon in Hadronic Collisions', Nuclear Physics B **478**, 273–310 (1996), arXiv:`hep-ph/9604351`.

[140] G. P. Lepage, 'A new algorithm for adaptive multidimensional integration', Journal of Computational Physics **27**, 192–203 (1978).

[141] G. P. Lepage, *VEGAS - An Adaptive Multi-dimensional Integration Program*, CLNS-80/447 (03/1980).

[142] TOP500 The List, *TOP500 Supercomputer*, `https://www.top500.org/` (visited on 11/13/2017).

[143] T. Hahn, 'Cuba—a library for multidimensional numerical integration', Computer Physics Communications **168**, 78–95 (2005), arXiv:`hep-ph/0404043`.

[144] T. Hahn, 'Concurrent Cuba', (2014), arXiv:1408.6373.

[145] J. Kanzaki, 'Monte Carlo integration on GPU', The European Physical Journal C **71**, 1–7 (2011), arXiv:1010.2107.

[146] NVIDIA, *CUDA C Programming Guide*, http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html (visited on 11/14/2017).

[147] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming* (Pearson Education, 07/2010), xx, 290.

[148] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Second Edition (William H. Press, 2002), xxvi, 925.

[149] *GNU Scientific Library*, https://www.gnu.org/software/gsl/ (visited on 11/13/2017).

[150] P. L'Ecuyer and R. Simard, 'TestU01: A C Library for Empirical Testing of Random Number Generators', ACM Trans. Math. Softw. **33**, 22:1–22:40 (2007).

[151] I. M. Sobol', 'On the distribution of points in a cube and the approximate evaluation of integrals', USSR Computational Mathematics and Mathematical Physics **7**, 86–112 (1967).

[152] I. A. Antonov and V. M. Saleev, 'An economic method of computing LP_τ-sequences', USSR Computational Mathematics and Mathematical Physics **19**, 252–256 (1979).

[153] P. L'Ecuyer, 'Tables of linear congruential generators of different sizes and good lattice structure', Mathematics of Computation of the American Mathematical Society **68**, 249–260 (1999).

[154] P. L'Ecuyer, 'Efficient and Portable Combined Random Number Generators', Commun. ACM **31**, 742–751 (1988).

[155] M. Matsumoto and T. Nishimura, 'Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator', ACM Trans. Model. Comput. Simul. **8**, 3–30 (1998).

[156] J. K. Salmon, M. A. Moraes, R. O. Dror, and D. E. Shaw, 'Parallel Random Numbers: As Easy As 1, 2, 3', in Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC '11 (2011), 16:1–16:12.

[157] A. Buckley et al., 'LHAPDF6: parton density access in the LHC precision era', The European Physical Journal C **75**, 132 (2015), arXiv:1412.7420.

[158] M. Lambertsen and W. Vogelsang, 'Drell-Yan lepton angular distributions in perturbative QCD', Physical Review D **93**, 114013 (2016), arXiv:1605.02625.

[159] M. Lambertsen and W. Vogelsang, 'Drell-Yan lepton angular distributions in perturbative QCD', Proceedings of Science **QCDEV2016**, 045 (2016).

[160] Karlsruher Institut für Technologie, *bwHPC-C5*, (12/18/2015) http://www.bwhpc-c5.de/en/index.php (visited on 01/01/2017).

[161] J. C. Collins and D. E. Soper, 'Angular distribution of dileptons in high-energy hadron collisions', Physical Review D **16**, 2219–2225 (1977).

[162] C. S. Lam and W.-K. Tung, 'Systematic approach to inclusive lepton pair production in hadronic collisions', Physical Review D **18**, 2447–2461 (1978).

[163] R. J. Oakes, 'Muon pair production in strong interactions', Il Nuovo Cimento A Series 10 **44**, 440–447 (1966).

[164] D. Boer and W. Vogelsang, 'Drell-Yan lepton angular distribution at small transverse momentum', Physical Review D **74**, 014004 (2006), arXiv:hep-ph/0604177.

[165] E. L. Berger, J. W. Qiu, and R. A. Rodriguez-Pedraza, 'Transverse momentum dependence of the angular distribution of the Drell-Yan process', Physical Review D **76**, 074006 (2007), arXiv:0708.0578.

[166] E. L. Berger, J.-W. Qiu, and R. A. Rodriguez-Pedraza, 'Angular distribution of leptons from the decay of massive vector bosons', Physics Letters B **656**, 74–78 (2007), arXiv:0707.3150.

[167] J.-C. Peng and J.-W. Qiu, 'Novel phenomenology of parton distributions from the Drell–Yan process', Progress in Particle and Nuclear Physics **76**, 43–75 (2014), arXiv:1401.0934.

[168] M. Guanziroli et al. (NA 10 Collaboration), 'Angular distributions of muon pairs produced by negative pions on deuterium and tungsten', Zeitschrift für Physik C Particles and Fields **37**, 545–556 (1988).

[169] J. S. Conway et al., 'Experimental study of muon pairs produced by 252-GeV pions on tungsten', Physical Review D **39**, 92–122 (1989).

[170] L. Y. Zhu et al. (FNAL E866/NuSea Collaboration), 'Measurement of Angular Distributions of Drell-Yan Dimuons in $p+d$ Interactions at 800 GeV/c', Physical Review Letters **99**, 082301 (2007), arXiv:hep-ex/0609005.

[171] L. Y. Zhu et al. (FNAL E866/NuSea Collaboration), 'Measurement of Angular Distributions of Drell-Yan Dimuons in $p+p$ Interactions at 800 GeV/c', Physical Review Letters **102**, 182001 (2009), arXiv:0811.4589.

[172] T. Aaltonen et al. (CDF Collaboration), 'First Measurement of the Angular Coefficients of Drell-Yan e^+e^- Pairs in the Z Mass Region from $p\overline{p}$ Collisions at $\sqrt{s} = 1.96$ TeV', Physical Review Letters **106**, 241801 (2011), arXiv:1103.5699.

[173] V. Khachatryan et al. (CMS Collaboration), 'Angular coefficients of Z bosons produced in pp collisions at $\sqrt{s} = 8$ TeV and decaying to $\mu^+\mu^-$ as a function of transverse momentum and rapidity', Physics Letters B **750**, 154–175 (2015), arXiv:1504.03512.

[174] G. Aad et al. (ATLAS Collaboration), 'Measurement of the angular coefficients in Z-boson events using electron and muon pairs from data taken at $\sqrt{s} = 8\,\text{TeV}$ with the ATLAS detector', Journal of High Energy Physics **2016**, 159 (2016), arXiv:1606.00689.

[175] D. Boer, 'Investigating the origins of transverse spin asymmetries at BNL RHIC', Physical Review D **60**, 014012 (1999), arXiv:hep-ph/9902255.

[176] D. Boer and P. J. Mulders, 'Time-reversal odd distribution functions in leptoproduction', Physical Review D **57**, 5780–5786 (1998), arXiv:hep-ph/9711485.

[177] S. J. Brodsky, D. S. Hwang, and I. Schmidt, 'Final-state interactions and single-spin asymmetries in semi-inclusive deep inelastic scattering', Physics Letters B **530**, 99–107 (2002), arXiv:hep-ph/0201296.

[178] S. J. Brodsky, D. S. Hwang, and I. Schmidt, 'Initial-state interactions and single-spin asymmetries in Drell–Yan processes', Nuclear Physics B **642**, 344–356 (2002), arXiv:hep-ph/0206259.

[179] V. Barone, S. Melis, and A. Prokudin, 'Azimuthal asymmetries in unpolarized Drell-Yan processes and the Boer-Mulders distributions of antiquarks', Physical Review D **82**, 114025 (2010), arXiv:1009.3423.

[180] Z. Lu and I. Schmidt, 'The $\cos 2\phi$ azimuthal asymmetry of unpolarized dilepton production at the Z pole', Physical Review D **84**, 094002 (2011), arXiv:1107.4693.

[181] Z. Lu and B.-Q. Ma, 'Azimuthal asymmetry in unpolarized πN Drell–Yan process', Physics Letters B **615**, 200–206 (2005), arXiv:hep-ph/0504184.

[182] B. Pasquini and P. Schweitzer, 'Pion transverse momentum dependent parton distributions in a light-front constituent approach, and the Boer-Mulders effect in the pion-induced Drell-Yan process', Physical Review D **90**, 014050 (2014), arXiv:1406.2056.

[183] C. S. Lam and W.-K. Tung, 'Structure function relations at large transverse momenta in Lepton-pair production processes', Physics Letters B **80**, 228–231 (1979).

[184] J. C. Collins, 'Simple Prediction of Quantum Chromodynamics for Angular Distribution of Dileptons in Hadron Collisions', Physical Review Letters **42**, 291–294 (1979).

[185] K. Kajantie, J. Lindfors, and R. Raitio, 'QCD angular correlations for muon pair production', Physics Letters B **74**, 384–388 (1978).

[186] J. Cleymans and M. Kuroda, 'Angular distribution of dileptons in hadronic collisions', Nuclear Physics B **155**, 480–492 (1979).

[187] J. Cleymans and M. Kuroda, 'Erratum: Angular distribution of dileptons in hadronic collisions [Nucl. Phys. B 155, 488 (1979)]', Nuclear Physics B **160**, 510 (1979).

[188] J. Lindfors, 'Angular Distribution of Large q_T Muon Pairs in Different Reference Frames', Physica Scripta **20**, 19 (1979).

[189] C. S. Lam and W.-K. Tung, 'Parton-model relation without quantum-chromodynamic modifications in lepton pair production', Physical Review D **21**, 2712–2715 (1980).

[190] E. Mirkes, 'Angular decay distribution of leptons from W-bosons at NLO in hadronic collisions', Nuclear Physics B **387**, 3–85 (1992).

[191] E. Mirkes and J. Ohnemus, 'Angular distributions of Drell-Yan lepton pairs at the Fermilab Tevatron: Order α_s^2 corrections and Monte Carlo studies', Physical Review D **51**, 4891–4904 (1995), arXiv:hep-ph/9412289.

[192] K. Melnikov and F. Petriello, 'Electroweak gauge boson production at hadron colliders through $\mathcal{O}(\alpha_s^2)$', Physical Review D **74**, 114017 (2006), arXiv:hep-ph/0609070.

[193] R. Gavin, Y. Li, F. Petriello, and S. Quackenbush, 'FEWZ 2.0: A code for hadronic Z production at next-to-next-to-leading order', Computer Physics Communications **182**, 2388–2403 (2011), arXiv:1011.3540.

[194] Y. Li and F. Petriello, 'Combining QCD and electroweak corrections to dilepton production in the framework of the FEWZ simulation code', Physical Review D **86**, 094034 (2012), arXiv:1208.5967.

[195] S. Catani, L. Cieri, G. Ferrera, D. de Florian, and M. Grazzini, 'Vector Boson Production at Hadron Colliders: A Fully Exclusive QCD Calculation at Next-to-Next-to-Leading Order', Physical Review Letters **103**, 082001 (2009), arXiv:0903.2120.

[196] S. Catani and M. Grazzini, 'Next-to-Next-to-Leading-Order Subtraction Formalism in Hadron Collisions and its Application to Higgs-Boson Production at the Large Hadron Collider', Physical Review Letters **98**, 222002 (2007), arXiv:hep-ph/0703012.

[197] J.-C. Peng, W.-C. Chang, R. E. McClellan, and O. Teryaev, 'Interpretation of angular distributions of Z-boson production at colliders', Physics Letters B **758**, 384–388 (2016), arXiv:1511.08932.

[198] A. Brandenburg, O. Nachtmann, and E. Mirkes, 'Spin effects and factorization in the Drell-Yan process', Zeitschrift für Physik C Particles and Fields **60**, 697–709 (1993).

[199] E. L. Berger and S. J. Brodsky, 'Quark Structure Functions of Mesons and the Drell-Yan Process', Physical Review Letters **42**, 940–944 (1979).

[200] A. Brandenburg, S. J. Brodsky, V. V. Khoze, and D. Müller, 'Angular Distributions in the Drell-Yan Process: A Closer Look at Higher Twist Effects', Physical Review Letters **73**, 939–942 (1994), arXiv:hep-ph/9403361.

[201] K. J. Eskola, P. Hoyer, M. Vänttinen, and R. Vogt, 'Higher-twist effects in the Drell-Yan angular distribution', Physics Letters B **333**, 526–530 (1994), arXiv:hep-ph/9404322.

[202] J. Zhou, F. Yuan, and Z.-T. Liang, 'Drell–Yan lepton pair azimuthal asymmetry in hadronic processes', Physics Letters B **678**, 264–268 (2009), arXiv:0901.3601.

[203] P. M. Nadolsky and C.-P. Yuan, 'Soft parton radiation in polarized vector boson production: theoretical issues', Nuclear Physics B **666**, 3–30 (2003), arXiv:hep-ph/0304001.

[204] F. Landry, R. Brock, P. M. Nadolsky, and C.-P. Yuan, 'Fermilab Tevatron run-1 Z boson data and the Collins-Soper-Sterman resummation formalism', Physical Review D **67**, 073016 (2003), arXiv:hep-ph/0212159.

[205] A. V. Konychev and P. M. Nadolsky, 'Universality of the Collins-Soper-Sterman nonperturbative function in vector boson production', Physics Letters B **633**, 710–714 (2006), arXiv:hep-ph/0506225.

[206] P. Sun, J. Isaacson, C.-P. Yuan, and F. Yuan, 'Universal Non-perturbative Functions for SIDIS and Drell-Yan Processes', (2014), arXiv:1406.3073.

[207] U. D'Alesio, M. G. Echevarria, S. Melis, and I. Scimemi, 'Non-perturbative QCD effects in q_T spectra of Drell-Yan and Z-boson production', Journal of High Energy Physics **2014**, 098 (2014), arXiv:1407.3311.

[208] S. Catani, D. Florian, G. Ferrera, and M. Grazzini, 'Vector boson production at hadron colliders: transverse-momentum resummation and leptonic decay', Journal of High Energy Physics **2015**, 047 (2015), arXiv:1507.06937.

[209] A. Bacchetta, D. Boer, M. Diehl, and P. J. Mulders, 'Matches and mismatches in the descriptions of semi-inclusive processes at low and high transverse momentum', Journal of High Energy Physics **2008**, 023 (2008), arXiv:0803.0227.

[210] S. Arnold, A. Metz, and M. Schlegel, 'Dilepton production from polarized hadron hadron collisions', Physical Review D **79**, 034005 (2009), arXiv:0809.2262.

[211] M. Boglione and S. Melis, 'Polarized and unpolarized Drell-Yan angular distribution in the helicity formalism', Physical Review D **84**, 034038 (2011), arXiv:1103.2084.

[212] P. J. Sutton, A. D. Martin, R. G. Roberts, and W. J. Stirling, 'Parton distributions for the pion extracted from Drell-Yan and prompt photon experiments', Physical Review D **45**, 2349–2359 (1992).

[213] M. Glück, E. Reya, and A. Vogt, 'Pionic parton distributions', Zeitschrift für Physik C Particles and Fields **53**, 651–655 (1992).

[214] V. Khachatryan et al. (CMS Collaboration), 'Measurement of the Z boson differential cross section in transverse momentum and rapidity in proton–proton collisions at 8 TeV', Physics Letters B **749**, 187–209 (2015), arXiv:1504.03511.

[215] A. G.-D. Ridder, T. Gehrmann, E. W. N. Glover, A. Huss, and T. A. Morgan, 'NNLO QCD corrections for Drell-Yan p_T^Z and $\phi_\eta^\star$ observables at the LHC', Journal of High Energy Physics **2016**, 94 (2016), arXiv:1610.01843.

[216] A. G.-D. Ridder, T. Gehrmann, E. W. N. Glover, A. Huss, and T. A. Morgan, 'The NNLO QCD corrections to Z boson production at large transverse momentum', Journal of High Energy Physics **2016**, 133 (2016), arXiv:1605.04295.

[217] M. Chiosso, 'Polarised Drell-Yan measurements at COMPASS', EPJ Web of Conferences **85**, edited by U. D'Alesio and F. Murgia, 02036 (2015).

[218] K. Nakano (E906/SeaQuest Collaboration), 'Measurement of Boer-Mulders Function via Drell-Yan Process by SeaQuest Experiment at Fermilab', International Journal of Modern Physics: Conference Series **40**, 1660041 (2016).

[219] Y. L. Dokshitzer, D. I. D'yakonov, and S. I. Troyan, 'On the transverse momentum distribution of massive lepton pairs', Physics Letters B **79**, 269–272 (1978).

[220] G. Parisi and R. Petronzio, 'Small transverse momentum distributions in hard processes', Nuclear Physics B **154**, 427–440 (1979).

[221] J. C. Collins and D. E. Soper, 'Back-to-back jets in QCD', Nuclear Physics B **193**, 381–443 (1981).

[222] J. C. Collins and D. E. Soper, 'Errata: Back-to-back jets in QCD', Nuclear Physics B **213**, 545 (1983).

[223] J. C. Collins and D. E. Soper, 'Back-to-back jets: Fourier transform from b to k_T', Nuclear Physics B **197**, 446–476 (1982).

[224] G. Altarelli, R. K. Ellis, M. Greco, and G. Martinelli, 'Vector boson production at colliders: A theoretical reappraisal', Nuclear Physics B **246**, 12–44 (1984).

[225] J. C. Collins, D. E. Soper, and G. F. Sterman, 'Transverse momentum distribution in Drell-Yan pair and W and Z boson production', Nuclear Physics B **250**, 199–224 (1985).

[226] R. Meng, F. I. Olness, and D. E. Soper, 'Semi-inclusive deeply inelastic scattering at small q_T', Physical Review D **54**, 1919–1935 (1996), arXiv:hep-ph/9511311.

[227] P. Nadolsky, D. R. Stump, and C.-P. Yuan, 'Semi-inclusive hadron production at DESY HERA: The effect of QCD gluon resummation', Physical Review D **61**, 014003 (1999), arXiv:hep-ph/9906280.

[228] P. Nadolsky, D. R. Stump, and C.-P. Yuan, 'Erratum: Semi-inclusive hadron production at DESY HERA: The effect of QCD gluon resummation [Phys. Rev. D 61, 014003 (2000)]', Physical Review D **64**, 059903 (2001).

[229] P. M. Nadolsky, D. R. Stump, and C.-P. Yuan, 'Phenomenology of multiple parton radiation in semi-inclusive deep-inelastic scattering', Physical Review D **64**, 114011 (2001), arXiv:hep-ph/0012261.

[230] P. M. Nadolsky, D. R. Stump, and C.-P. Yuan, 'Azimuthal asymmetries at HERA: theoretical aspects', Physics Letters B **515**, 175–180 (2001), arXiv:hep-ph/0012262.

[231] A. Weber, 'Soft gluon resummations for polarized Drell-Yan dimuon production', Nuclear Physics B **382**, 63–96 (1992).

[232] A. Weber, 'Soft gluon resummation for the single-spin production of $W^{\pm}$-bosons', Nuclear Physics B **403**, 545–571 (1993).

[233] D. Boer, 'Double transverse spin asymmetries in vector boson production', Physical Review D **62**, 094029 (2000), arXiv:hep-ph/0004217.

[234] D. Boer, 'Sudakov suppression in azimuthal spin asymmetries', Nuclear Physics B **603**, 195–217 (2001), arXiv:hep-ph/0102071.

[235] A. Idilbi, X. Ji, J. P. Ma, and F. Yuan, 'Collins-Soper equation for the energy evolution of transverse-momentum and spin dependent parton distributions', Physical Review D **70**, 074021 (2004), arXiv:hep-ph/0406302.

[236] H. Kawamura, J. Kodaira, H. Shimizu, and K. Tanaka, 'The Dilepton Q_T Spectrum in Transversely Polarized Drell-Yan Process in QCD', Progress of Theoretical Physics **115**, 667–672 (2006), arXiv:hep-ph/0512137.

[237] Y. Koike, J. Nagashima, and W. Vogelsang, 'Resummation for polarized semi-inclusive deep-inelastic scattering at small transverse momentum', Nuclear Physics B **744**, 59–79 (2006), arXiv:hep-ph/0602188.

[238] S. M. Aybat and T. C. Rogers, 'Transverse momentum dependent parton distribution and fragmentation functions with QCD evolution', Physical Review D **83**, 114042 (2011), arXiv:1101.5057.

[239] M. Boglione, J. O. Gonzalez Hernandez, S. Melis, and A. Prokudin, 'A study on the interplay between perturbative QCD and CSS/TMD formalism in SIDIS processes', Journal of High Energy Physics, 095 (2015), arXiv:1412.1383.

[240] R. Meng, F. I. Olness, and D. E. Soper, 'Semi-inclusive deeply inelastic scattering at electron-proton colliders', Nuclear Physics B **371**, 79–110 (1992).

[241] V. V. Sudakov, 'Vertex parts at very high-energies in quantum electrodynamics', Sov.Phys.JETP **3**, 65–71 (1956).

[242] M. Schlegel, 'Time-reversal odd effects in semi-inclusive deep-inelastic scattering' (Ruhr-Universität Bochum, 2006), 157 pp.

[243] P. J. Mulders, 'Transverse momentum dependence in structure functions in hard scattering processes', 2001.

[244] K. Goeke, A. Metz, and M. Schlegel, 'Parameterization of the quark–quark correlator of a spin-$\frac{1}{2}$ hadron', Physics Letters B **618**, 90–96 (2005), arXiv:https://arxiv.org/abs/hep-ph/0504130v1.

[245] Y. Koike and J. Nagashima, 'Double spin asymmetries for large-p_T hadron production in semi-inclusive DIS', Nuclear Physics B **660**, 269–288 (2003), arXiv:hep-ph/0302061.

[246] Y. Koike and J. Nagashima, 'Erratum to: "double spin asymmetries for large-p_T hadron production in semi-inclusive DIS"[Nucl. Phys. B 660 (2003) 269]', Nuclear Physics B **742**, 312–316 (2006), arXiv:hep-ph/0302061.

[247] A. Bacchetta et al., 'Semi-inclusive deep inelastic scattering at small transverse momentum', Journal of High Energy Physics **2007**, 093 (2007), arXiv:hep-ph/0611265.

[248] F. D. Aaron et al. (H1 and ZEUS Collaborations), 'Combined measurement and QCD analysis of the inclusive $e^{\pm}p$ scattering cross sections at HERA', Journal of High Energy Physics **2010**, 109 (2010), arXiv:0911.0884.

[249] A. Accardi et al., 'New parton distributions from large-x and low-Q^2 data', Physical Review D **81**, 034016 (2010), arXiv:0911.2254.

[250] J. F. Owens, A. Accardi, and W. Melnitchouk, 'Global parton distributions with nuclear and finite-Q^2 corrections', Physical Review D **87**, 094012 (2013), arXiv:1212.1702.

[251] M. Glück, E. Reya, M. Stratmann, and W. Vogelsang, 'Next-to-leading-order radiative parton model analysis of polarized deep inelastic lepton-nucleon scattering', Physical Review D **53**, 4775–4786 (1996), arXiv:hep-ph/9508347.

[252] T. Gehrmann and W. J. Stirling, 'Polarized parton distributions in the nucleon', Physical Review D **53**, 6100–6109 (1996), arXiv:hep-ph/9512406.

[253] M. Glück, E. Reya, M. Stratmann, and W. Vogelsang, 'Models for the polarized parton distributions of the nucleon', Physical Review D **63**, 094005 (2001), arXiv:hep-ph/0011215.

[254] E. Leader, A. V. Sidorov, and D. B. Stamenov, 'New analysis concerning the strange quark polarization puzzle', Physical Review D **91**, 054017 (2015), arXiv:1410.1657.

[255] M. Glück, E. Reya, and A. Vogt, 'Dynamical Parton Distributions Revisited', The European Physical Journal C **5**, 461–470 (1998), arXiv:hep-ph/9806404.

[256] D. de Florian and R. Sassot, 'Nuclear parton distributions at next to leading order', Physical Review D **69**, 074028 (2004), arXiv:hep-ph/0311227.

[257] L. Bourhis, M. Fontannaz, J. P. Guillet, and M. Werlen, 'Next-to-leading order determination of fragmentation functions', The European Physical Journal C - Particles and Fields **19**, 89–98 (2001), arXiv:hep-ph/0009101.

[258] D. de Florian, R. Sassot, and M. Stratmann, 'Global analysis of fragmentation functions for pions and kaons and their uncertainties', Physical Review D **75**, 114010 (2007), arXiv:hep-ph/0703242.

[259] D. de Florian, R. Sassot, M. Epele, R. J. Hernández-Pinto, and M. Stratmann, 'Parton-to-pion fragmentation reloaded', Physical Review D **91**, 014035 (2015), arXiv:1410.6027.

[260] E. Leader, A. V. Sidorov, and D. B. Stamenov, 'Determination of the fragmentation functions from an NLO QCD analysis of the HERMES data on pion multiplicities', Physical Review D **93**, 074026 (2016), arXiv:1506.06381.

[261] A. V. Vinnikov, 'Code for prompt numerical computation of the leading order GPD evolution', (2006), arXiv:hep-ph/0604248.

[262] A. Signori, A. Bacchetta, M. Radici, and G. Schnell, 'Investigations into the flavor dependence of partonic transverse momentum', Journal of High Energy Physics **2013**, 194 (2013), arXiv:1309.3507.

[263] The Durham HepData Project, *Durham PDF server*, http://hepdata.cedar.ac.uk/pdfs/ (visited on 01/18/2017).

[264] T. Kaufmann, 'R dependence of jet cross sections' (Eberhard Karls Universität Tübingen, 2014).

[265] H. Shimizu, G. F. Sterman, W. Vogelsang, and H. Yokoya, 'Dilepton production near partonic threshold in transversely polarized $\bar{p}p$ collisions', Physical Review D **71**, 114007 (2005), arXiv:hep-ph/0503270.

[266] S. Catani and L. Trentadue, 'Comment of QCD exponentiation at large x', Nuclear Physics B **353**, 183–IN2 (1991).

[267] Y. L. Dokshitzer, D. I. D'yakonov, and S. I. Troyan, 'Hard semi-inclusive processes in QCD', Physics Letters B **78**, 290–294 (1978).

[268] G. Parisi, 'Summing large perturbative corrections in QCD', Physics Letters B **90**, 295–296 (1980).

[269] G. Curci and M. Greco, 'Large infra-red corrections in QCD processes', Physics Letters B **92**, 175–178 (1980).

[270] G. Curci and M. Greco, 'Soft corrections to the Drell-Yan process in QCD', Physics Letters B **102**, 280–284 (1981).

[271] J. Collins et al., 'Relating transverse-momentum-dependent and collinear factorization theorems in a generalized formalism', Physical Review D **94**, 034014 (2016), arXiv:1605.00671.

[272] J. Kodaira and L. Trentadue, 'Summing soft emission in QCD', Physics Letters B **112**, 66–70 (1982).

[273] C. Davies and W. J. Stirling, 'Non-leading corrections to the Drell-Yan cross section at small transverse momentum', Nuclear Physics B **244**, 337–348 (1984).

[274] C. Davies, B. R. Webber, and W. J. Stirling, 'Drell-Yan cross sections at small transverse momentum', Nuclear Physics B **256**, 413–433 (1985).

[275] S. Catani, E. D'Emilio, and L. Trentadue, 'The gluon form factor to higher orders: Gluon-gluon annihilation at small Q_t', Physics Letters B **211**, 335–342 (1988).

[276] M. Anselmino, M. Boglione, and S. Melis, 'Strategy towards the extraction of the Sivers function with transverse momentum dependent evolution', Physical Review D **86**, 014028 (2012), arXiv:1204.1239.

[277] S. Catani, M. L. Mangano, and P. Nason, 'Sudakov resummation for prompt-photon production in hadron collisions', Journal of High Energy Physics **1998**, 024 (1998), arXiv:hep-ph/9806484.

[278] R. K. Ellis, W. J. Stirling, and B. R. Webber, *QCD and collider physics*, 1. paperback (Cambridge Univ. Press, Cambridge, 2003), XIV, 435.

[279] J. Blümlein and S. Kurth, 'Harmonic sums and Mellin transforms up to two-loop order', Physical Review D **60**, 014018 (1999), arXiv:hep-ph/9810241.

[280] J. Blümlein, 'Analytic continuation of Mellin transforms up to two-loop order', Computer Physics Communications **133**, 76–104 (2000), arXiv:hep-ph/0003100.

[281] J. Blümlein, 'Algebraic relations between harmonic sums and associated quantities', Computer Physics Communications **159**, 19–54 (2004), arXiv:hep-ph/0311046.

[282] J. Blümlein and V. Ravindran, '$\mathcal{O}(\alpha_s^2)$ timelike Wilson coefficients for parton-fragmentation functions in Mellin space', Nuclear Physics B **749**, 1–24 (2006), arXiv:hep-ph/0604019.

[283] J. Blümlein, 'Structural relations of harmonic sums and Mellin transforms up to weight $w = 5$', Computer Physics Communications **180**, 2218–2249 (2009), arXiv:0901.3106.

[284] *Boost C++ Libraries*, http://www.boost.org/ (visited on 11/19/2017).

[285] M. Lambertsen, *Finael*, https://bitbucket.org/mlambertsen/finael (visited on 03/23/2018).

[286] Open Source Initiative, *The MIT License*, https://opensource.org/licenses/MIT (visited on 11/19/2017).

[287] *Thrust*, (01/26/2012) https://developer.nvidia.com/thrust (visited on 11/25/2017).

[288] S. Meyers, *Effective C++: 55 Specific Ways to Improve Your Programs and Designs, 3rd Edition*, 3rd (Addison-Wesley Professional. Part of the Addison-Wesley Professional Computing Series series., 05/12/2005), 320 pp.

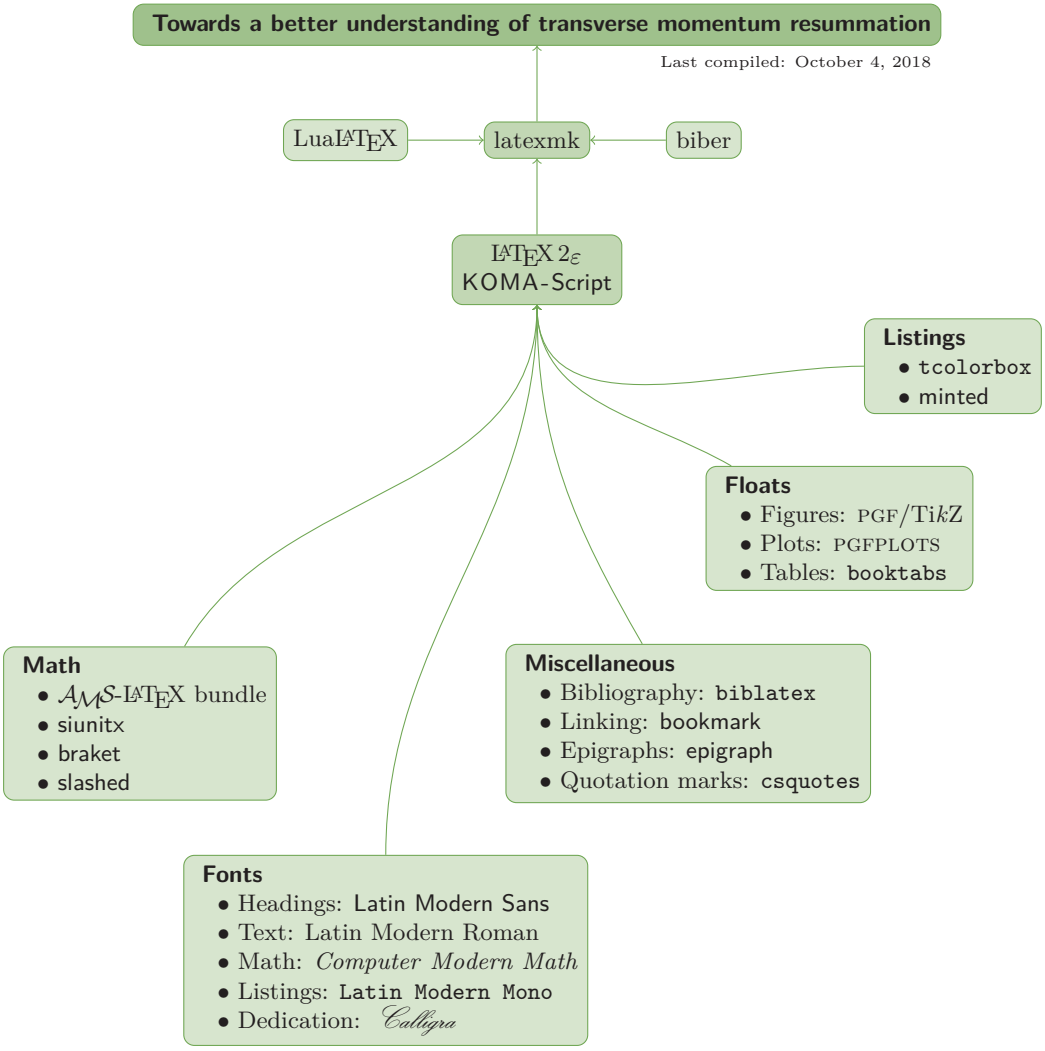
[289] *cuRAND*, (08/20/2011) https://developer.nvidia.com/curand (visited on 12/04/2017).

[290] E. Laenen, G. F. Sterman, and W. Vogelsang, 'Higher-Order QCD Corrections in Prompt Photon Production', Physical Review Letters **84**, 4296–4299 (2000), pmid: 10990670.

[291] A. Kulesza, G. F. Sterman, and W. Vogelsang, 'Joint resummation in electroweak boson production', Physical Review D **66**, 014011 (2002), arXiv:hep-ph/0202251.

[292] M. Jamin and M. E. Lautenbacher, 'TRACER version 1.1: A Mathematica package for γ-algebra in arbitrary dimensions', Computer Physics Communications **74**, 265–288 (1993).

Colophon


Towards a better understanding of transverse momentum resummation
Last compiled: October 4, 2018
LuaLaTeX
latexmk
biber
LaTeX 2ε
KOMA-Script
Listings
• tcolorbox
• minted
Floats
• Figures: PGF/TikZ
• Plots: PGFPLOTS
• Tables: booktabs
Math
• AMS-LaTeX bundle
• siunitx
• braket
• slashed
Miscellaneous
• Bibliography: biblatex
• Linking: bookmark
• Epigraphs: epigraph
• Quotation marks: csquotes
Fonts
• Headings: Latin Modern Sans
• Text: Latin Modern Roman
• Math: Computer Modern Math
• Listings: Latin Modern Mono
• Dedication: Calligra