

Institut für Theoretische Physik  
Fakultät Mathematik und Naturwissenschaften  
Technische Universität Dresden

# **Automating methods to improve precision in Monte-Carlo event generation for particle colliders**

Dissertation  
zur Erlangung des akademischen Grades  
Doctor rerum naturalium

vorgelegt von  
Tanju Gleisberg  
geboren am 27. März 1978 in Dresden

Dresden 2007



Eingereicht am 09.10.2007

1. Gutachter: \_\_\_\_\_

2. Gutachter: \_\_\_\_\_

3. Gutachter: \_\_\_\_\_

Verteidigt am \_\_\_\_\_

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	The event generator SHERPA	10
1.2	Outline of this thesis	11
<b>I</b>	<b>Automatic calculation of tree level cross-sections</b>	<b>13</b>
<b>2</b>	<b>Matrix element generation at tree-level</b>	<b>15</b>
2.1	Automatic matrix element generation with AMEGIC++	16
2.1.1	General procedure	17
2.1.2	New models	19
2.1.3	Treatment for decay chains	22
2.1.4	Accessible processes	23
2.2	Recursion relations based on MHV-amplitudes	24
2.2.1	Partial amplitudes and colour decomposition	24
2.2.2	MHV amplitudes and the CSW technique	25
2.2.3	Accessible processes	28
2.3	Colour dressed Berends-Giele recursion relations	29
2.3.1	General form of the recursion	30
2.3.2	Decomposition of four-point vertices	31
2.3.3	Colour dressed amplitudes	32
2.3.4	Prefactors of amplitudes with external fermions	34
2.3.5	Accessible processes	34

<b>3 Monte-Carlo phase space integration .....</b>	<b>35</b>
3.1 Basic concepts for optimization .....	36
3.2 Integrator techniques .....	37
3.2.1 The self-adaptive integrator VEGAS .....	37
3.2.2 The Multi-Channel method .....	38
3.2.3 Improved Multi-Channeling .....	39
3.3 Automatic generation of phase space maps .....	40
3.4 A general-purpose integrator for QCD-processes .....	42
3.4.1 Antenna generation .....	44
3.4.2 Integrator setup .....	46
3.5 Integration with colour sampling .....	47
3.5.1 Colour sampling .....	48
3.5.2 Integrator setup .....	49
3.5.3 The optimization procedure .....	50
<b>4 Efficiency comparison and results .....</b>	<b>53</b>
4.1 Matrix elements .....	53
4.2 Integration performance and results .....	55
4.3 Conclusions .....	61
<b>II Automating NLO calculations .....</b>	<b>63</b>
<b>5 Automating the Dipole-Subtraction method .....</b>	<b>65</b>
5.1 Brief review of the Catani-Seymour formalism .....	67
5.1.1 NLO cross sections and the subtraction procedure .....	67
5.1.2 Generalization to hadronic initial states .....	70
5.1.3 Observable-independent formulation of the subtraction method .....	71
5.1.4 The dipole subtraction functions .....	72
5.1.5 Integrated dipole terms .....	78
5.1.6 Freedom in the definition of dipole terms .....	80

5.2	Implementation in AMEGIC++	81
5.2.1	Generation of CS dipole terms	82
5.2.2	Generation of the finite part of integrated dipole terms	86
5.2.3	Phase space integration	88
5.2.4	Cuts and analysis framework for NLO calculations	89
5.3	Checks of the implementation	90
5.3.1	Explicit comparisons	90
5.3.2	Test of convergence for the real ME	91
5.3.3	Consistency checks with free parameters	94
5.4	First physical applications	96
5.4.1	Three-jet observables at LEP	96
5.4.2	DIS: $e^- p \rightarrow e^- + jet$	98
5.4.3	$W^-$ production at Tevatron	98
5.5	Conclusions and outlook	101
<b>6</b>	<b>A novel method to evaluate scalar 1-loop integrals</b>	<b>103</b>
6.1	Integral duality: mapping one-loop integrals onto phase-space integrals	104
6.2	Analytic approach to dual integrals	107
6.2.1	Soft integrals	107
6.2.2	Collinear integrals	109
6.2.3	Finite integrals	110
6.3	Numeric approaches	111
6.3.1	Construction of subtraction terms	111
6.3.2	Recursion relations to single out divergent parts	114
6.3.3	Numeric evaluation of finite integrals	116
6.3.4	Example: finite box integral	119
6.4	Conclusions and outlook	120
<b>7</b>	<b>Summary</b>	<b>121</b>
<b>Appendix A</b>	<b>COMIX implementation details</b>	<b>123</b>

A.1	Decomposition of electroweak four-particle vertices	123
A.2	Matrix element generation with COMIX	124
A.3	Lorentz functions	126
A.4	Vertices and Propagators	127
<b>Appendix B</b>	<b>Insertion operators for the dipole subtraction method</b>	<b>131</b>
<b>Appendix C</b>	<b>Relations for the evaluation of dual integrals</b>	<b>135</b>
C.1	Feynman parameterization	135
C.2	Integrals	135
C.3	Hypergeometric functions	136
<b>Bibliography</b>		<b>137</b>

# 1 Introduction

In high-energy physics, particle accelerators are the central experimental facilities to study fundamental principles of nature. There, high-energetic particles are brought to collision in order to explore their reactions under well defined laboratory conditions. Recent examples for such machines are the Large Electron Positron collider (LEP) at CERN, which, until November 2000 performed collisions at a centre-of-mass energy of up to 207 GeV, or the Tevatron at Fermilab, which currently operates proton–anti-proton collisions at an energy of 1.96 TeV.

Both experiments very successfully confirmed the predictions given by the Standard Model (SM) of particle physics to an astonishing precision. Despite its success, the SM is seen to be incomplete for a number of reasons. To name a few, it provides no explanation for the 19 free parameters defining masses and couplings, it does not explain the deep origin of electroweak symmetry breaking nor it gives answers to questions such as for the number of particle families or for incorporated gauge structures. Furthermore, the extrapolation of the SM to energy scales, much higher than the scale where electroweak symmetry breaking occurs ( $\approx 100$  GeV), is problematic from a theoretical point of view, which is referred to as the hierarchy problem. And lastly to be mentioned is the existence of some compelling observations, inexplicable by the SM, namely the neutrino mixing, dark matter and the baryon asymmetry seen in the Universe.

The upcoming start-up of the Large Hadron Collider (LHC) at CERN is expected to open a new era in high-energy physics. It is designed to provide proton-proton collisions at a centre-of-mass energy of 14 TeV, the highest ever been available in a ground-based laboratory. The exploration of the new energy scale will allow for ultimate precision tests of the SM, likely to reveal new physics and to give insights into the questions/problems left by the SM.

To interpret the new data, even after the machine and the detectors are fully understood (being a formidable task on its own), will be an enormous challenge requiring a close collaboration of experimentalists and theorists. Not only that most new physics scenarios will reveal itself in complicated multi-particle final states, the huge phase space available at LHC, together with a very high luminosity, leads to tremendous production rates of SM particles

which have to be understood to a yet unknown precision in order to extract possible new effects. Especially, due to the hadronic initial state at the LHC basically any high-energetic reaction will be accompanied by a number of rather hard jets.

Perturbative calculations form one of the best understood methods to provide predictions for the behaviour of a Quantum Field Theory and to compare them with experimental results. Many of the methods applied in such calculations have found their way into textbooks already decades ago, e.g. [1, 2, 3, 4, 5]. Typically, the perturbation parameter is related to the coupling constant of the theory in question, which in most cases indeed is a small quantity. This also implies that the corresponding fields may asymptotically appear as free fields and thus are the relevant objects of perturbation theory. This is obviously not true for the strong interactions, i.e. Quantum Chromo Dynamics (QCD), where the fields, quarks and gluons, asymptotically are confined in bound states only. This is due to the scaling behavior of the coupling constant of QCD,  $\alpha_S$ , which becomes small only for large momentum transfers. It is the confinement property that to some extent restricts the validity of perturbative calculations in QCD to the realm of processes characterized by large momentum transfers or by other large scales dominating the process.

Thus, a complete, quantum-mechanically correct treatment of a collision is currently far out of reach. Besides the already mentioned non-perturbative confinement phenomenon (and likewise the deconfinement, i.e. the partonic substructure of colliding hadrons), even for perturbatively accessible energy scales a full calculation to a fixed order of the perturbation parameter is restricted to a relatively small number of particles involved. This has mainly a technical reason: even at the lowest, the tree level the number of Feynman amplitudes to be calculated grows factorially with the number of particles. A realistic description of QCD bremsstrahlung is beyond this limit, and can be performed only by imposing further approximations. These normally take into account only leading kinematic logarithms, which dominate the soft and collinear region, and that can be exponentiated to all perturbative orders.

The assumptions, that the properties of a scattering process, related to different energy scales, factorize and that the non-perturbative phenomena can be described by a universal (experimentally measured) parameterization, are the basis for the development of dedicated computer codes, called Monte Carlo event generators. The past and current success of event generators, like PYTHIA [6, 7] or HERWIG [8, 9, 10], in describing a full wealth of various data justifies the underlying hypothesis. Typically, a scattering process is composed out of  $2 \rightarrow 2$  tree level matrix element, which is supplemented with a parton shower algorithm to describe the QCD bremsstrahlung of initial and final state partons. The non-perturbative confinement is treated via phenomenological hadronization models. In view of the new collider era, however, this treatment is insufficient to precisely describe additional hard QCD radiation, not included in the  $2 \rightarrow 2$  core process.

---

A way to systematically improve the precision has been proposed in [11, 12], known as the CKKW merging scheme. This approach allows to include higher order tree level matrix elements in the consideration, such that additional (possibly multiple) hard QCD radiation will be treated using the corresponding matrix element and, the approximative parton shower description only accounts for radiation below a certain scale, defined by a separation parameter. The catch of this method is to avoid a double counting of the QCD radiation, given by the matrix elements and the parton shower. Similar approaches are the LCKKW scheme [13] and the MLM matching [14, 15].

An alternative ansatz is to combine (QCD-) next-to-leading order matrix elements consistently with a parton shower, such that the overall cross section corresponds to the NLO result and the hardest additional QCD emission is accounted for by the real correction part of the matrix element. A first implementation of this idea has been realized for a number of specific processes in MC@NLO [16]. The main difficulty is again to avoid a double counting. Further advances in this direction have been presented by the POWHEG method [17]. A future milestone is clearly given by the development of a merging method, that, similar as the CKKW procedure for leading order matrix elements, combines next-to-leading order matrix elements of different parton multiplicity.

Certainly, a key for the improvement of the precision of event generators is the incorporation of higher fixed-order matrix elements. For a large number of physical questions it is also possible to define observables such, that the impact of soft QCD radiation and confinement effects are small. These observables, usually exclusive in a given final state configuration, can than be related directly to parton level matrix elements, without the need for a full event generator. Examples are, e.g., exclusive jet cross sections, imposing a suitable (infrared safe) jet algorithm such as  $k_T$  [18] or production cross sections for other than strongly interacting particles.

The extremely large phase space at LHC and the anticipated precision sets new demands on the complexity of required matrix elements. For tree level, this task has been fully automated in the past years. Computer codes, usually referred to as parton level generators, have been developed to manage this for the Standard Model and a number of popular extensions without significant user interface. Examples for such programs are the multipurpose codes **ALPGEN** [19], **AMEGIC++** [20], **HELAC/PHEGAS** [21, 22], **MadGraph** [23], and **0’Mega/Whizard** [24]. Typically processes with eight to ten external particles are within the reach of such implementations. Here, the main bottlenecks are the already mentioned factorial growth in the number of amplitudes and the increasing complexity of the multi-particle phase space.

For next-to-leading order matrix elements there is a number of codes available, that have coded manually calculated NLO matrix elements, e.g. **MCFM** [25] and **NLOJET** [26]. So far, no automated tool for the generation of the matrix element itself is available. This is because a true NLO calculation is certainly much more complex than a leading order

one. First of all, some of the essential ingredients, namely the loop or virtual contributions are not under full control yet. On the other hand, virtual and real corrections individually exhibit infrared divergencies which only cancel in the sum and thus require an additional regularization treatment to be evaluated. Despite of the remarkable progress that is made in this field [27, 28, 29, 30], the number of processes known at NLO is still rather concise. Fully differential calculations exists basically for all  $2 \rightarrow 2$ , most  $2 \rightarrow 3$  and a very few  $2 \rightarrow 4$  scattering processes. The complexity of the last-mentioned processes is at the very edge of what is manageable in a manual calculation. Thus, in prospect of the LHC and the large number of processes in quest (also involving possible scenarios beyond the SM), an automatic tool is highly demanded.

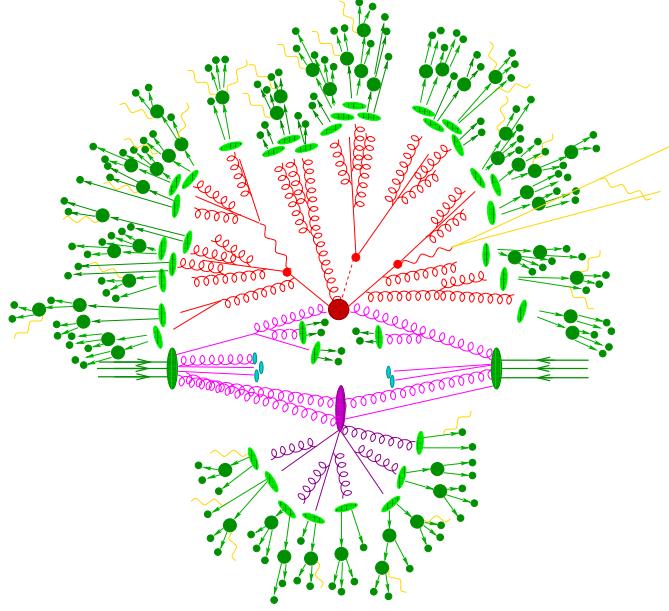
## 1.1 The event generator **SHERPA**

SHERPA, acronym for Simulation of High Energy Reactions of PArticles [31] is a new full multi-purpose event generator, intended to simulate all stages of a high-energy scattering event at lepton and hadron colliders, starting from the hard interaction down to hadrons, observable in a detector. It has been written entirely in the modern object-oriented programming language **C++**. The paradigm of the object-oriented style is reflected in the modularity of SHERPA, naturally imposed by the factorization of a scattering event, which has been discussed above. The emphasis for the whole framework has been placed on an improved simulation of jet-physics. This is realized by SHERPA’s key feature, the implementation of the already mentioned CKKW merging scheme.

The basic idea of CKKW is to divide the phase space for parton emission into a regime of jet production, reflected by appropriate (multi-jet) matrix elements, and a regime of jet evolution, addressed by a parton shower. The borderline between the two regimes is defined by a jet resolution cut, using a  $k_T$  algorithm [32, 33, 18]. To avoid a double counting, each configuration of matrix elements and the parton shower must be made exclusive before added together, done by a reweighting the matrix element through Sudakov form factors. Parton emissions from the shower are, if outside the allowed regime, prevented by a jet veto. As a result one obtains again inclusive event samples, correct up to (next-to) leading logarithmic accuracy, with only a residual dependence on the artificial jet-resolution cut [34, 35, 36, 37].

The main stages in the event generation with SHERPA and corresponding modules are (cf. Fig 1.1):

- Signal process / hard matrix element (central red blob in Fig 1.1), provided by AMEGIC++ [20].
- Initial- and final state parton showers, realized in APACIC++ [38].



**Figure 1.1:** Pictorial representation of an event in a hadron–hadron collisions, according to the factorization approach as realized in SHERPA.

- Underlying event / multiple parton interactions (violet blob in Fig 1.1), provided by AMISIC++.
- Hadronization (light green blobs in Fig 1.1), provided by AHADIC++ [39] or PYTHIA’s Lund string fragmentation [6].
- Decays of unstable primary hadrons and QED bremsstrahlung, provided by the modules HADRONSP++ and PHOTONSP++, respectively.

The overall coordination is performed by the SHERPA framework.

## 1.2 Outline of this thesis

This thesis concerns with the automation of fixed order perturbative calculations.

In Part I methods and implementations dealing with leading order calculations are discussed. Therein, in chapter 2, a number of extensions for the matrix element generator AMEGIC++ are presented. This includes the implementation of several effective interaction models, as well as some technical extensions up an alternative method to compute matrix elements, based on the Cachazo-Svrćek-Witten recursion relation [40]. Further, the implementation of the new matrix element generator COMIX is presented, which, based on Berends-Giele recursion relations [41], has been optimized to deal with exact tree level matrix elements up

to a very high particle multiplicity. chapter 3 is dedicated to phase space integration methods available in the SHERPA package, which have been revised and significantly improved. Strategies are presented to cope with the newly accessible final states. In chapter 4 the efficiency of the previously introduced matrix element and phase space generation methods is analysed and some results will be presented.

Part II is dedicated to strategies and implementations towards a full automation of next-to-leading order matrix element calculations. In chapter 5 the construction of an automatic algorithm to subtract infrared divergences in real QCD corrections through the Catani-Seymour dipole subtraction method [42] is reported. The resulting computer code has been implemented in the matrix element generator AMEGIC++. This will allow for the automatic generation of dipole subtraction terms and their integrals over the one-parton emission phase space for any given process. If the virtual matrix element is provided as well, this then directly leads to an NLO QCD parton level event generator. The abilities of the implementation are demonstrated on a few examples. Chapter 6 reports on a new approach to provide a general automatable method for the calculation of multi-leg one-loop integrals, the main bottleneck for NLO calculations. For scalar integrals systematic ways for the analytic extraction of infrared divergencies are discussed and a numeric strategy to evaluate remaining finite integrals is presented.

A summary of the thesis will be given in chapter 7.

## **Part I**

# **Automatic calculation of tree level cross-sections**



## 2 Matrix element generation at tree-level

The leading term of many observables in high energy particle phenomenology can be related to the calculation of tree level matrix elements. Such matrix elements can be used to directly compute parton level cross sections, or they might be used in the more complex environment of a full Monte Carlo event simulation, where the matrix elements, providing predictions for the highest energy scales are combined with other perturbative (e.g parton showers) and nonperturbativ (e.g. hadronization) approaches to compute hadronic cross sections or any observables related to them.

Although it is in principle straightforward to calculate tree level matrix elements, the computational complexity increases rapidly with the number of involved particles. This calculational task cannot be handled without dedicated computer codes. To name the major difficulties:

1. The number of contributing Feynman diagrams grows roughly factorially with the number of external particles. To give a number, the process  $e^+e^- \rightarrow e^+e^-e^+e^-e^+e^-$  in Standard Model, where only electroweak interactions are involved, requires the consideration of 13896 Feynman diagrams. The situation is even severe for processes involving strong interactions. There, the structure of non-abelian interactions leads to an almost uncontrollable inflation of terms. For instance the process  $gg \rightarrow 6g$  already incorporates 34300 diagrams, which in addition are correlated with a rather complicated SU(3) colour structure. Obviously, the common textbook method of squaring the diagrams by employing completeness relations for the external particles and evaluating the traces would fail to calculate the matrix element, even if implemented in a computer code.
2. In order to calculate hadronic cross sections a number of different parton level processes have to be considered. Especially for processes with hadronic initial state this number gets quickly very large, since it grows roughly exponentially with the number of external partons. For instance the process  $pp \rightarrow 4jets$  is assembled out of 486

parton level processes.

In this chapter recent developments of matrix element generation methods employed in the full event generator SHERPA [31] are discussed. Several, partly complementary, approaches are combined to get the best performance for any process and number of external particles, as well as a good flexibility concerning the underlying quantum field theories, i.e. standard and beyond standard model physics. This includes the following:

1. The fully automatic and general matrix element generator AMEGIC++ [20], which is based on the automatic generation and evaluation of Feynman amplitudes. It provides a high flexibility and has been extensively tested for a large number of processes. In section 2.1 a number of recent extensions is presented. AMEGIC++ can also be used to automatically generate phase space maps corresponding to the amplitude structure, this subject however is saved for chapter 3.
2. One extension of AMEGIC++, separately discussed in section 2.2, covers an alternative approach to construct matrix elements recursively from maximal helicity violating (MHV) amplitudes. Directly employing features of the SU(3) gauge symmetry it leads to much more compact description for QCD dominated processes.
3. A newly developed separate module, called COMIX, constructs matrix elements based on the Berends-Giele [41] recursion relation. This approach, presented in section 2.3, has been shown to have the slowest growth in complexity with the number of external particles [43]. Together with a decomposition into colour flows, which is best suited for sampling over colour assignments instead of computing the full sum, it is the superior method to compute matrix element with a very large number of external legs (typically  $\gtrsim 8$ ).

## 2.1 Automatic matrix element generation with AMEGIC++

AMEGIC++, acronym for (A Matrix Element Generator in C++), is a multi-purpose parton-level generator written in C++. In the current version 2.0[44]<sup>1</sup> it provides a convenient tool for the calculation of cross sections for scattering processes at the tree level in the framework of the SM and a number of popular extensions, such as the MSSM and the ADD model of large extra dimensions [45]. The implementation and cross section results have been

<sup>1</sup> A brief description of AMEGIC++ within the SHERPA framework can be found in [31], whereas a full documentation of the (partly obsolete) version 1.0 is given in [20] with some extensions and results discussed in [45, 46]. An update on the helicity formalism as it is used in the current version is documented in [45].

extensively tested against alternative codes [47, 46]. Apart from being used to compute parton level cross section at tree level, the program can also be employed by the SHERPA event simulation framework [31] to generate single events.

AMEGIC++’s tree level matrix elements further form the basis for the implementation of the next-to-leading order subtraction procedure, subject of chapter 5.

### 2.1.1 General procedure

#### Amplitude generation

The matrix element evaluation in the C++-code AMEGIC++ is performed computing Feynman amplitudes using a helicity method based on the developments in [48, 49, 50]. The fundamental idea of this method is to introduce a basis of massless spinors  $w(k_0, \lambda)$ , which satisfy

$$w(k_0, \lambda) = \lambda \not{k}_1 w(k_0, -\lambda) \quad \text{and} \quad (2.1)$$

$$w(k_0, \lambda) \bar{w}(k_0, \lambda) = \frac{1 + \lambda \gamma^5}{2} \not{k}_0; . \quad (2.2)$$

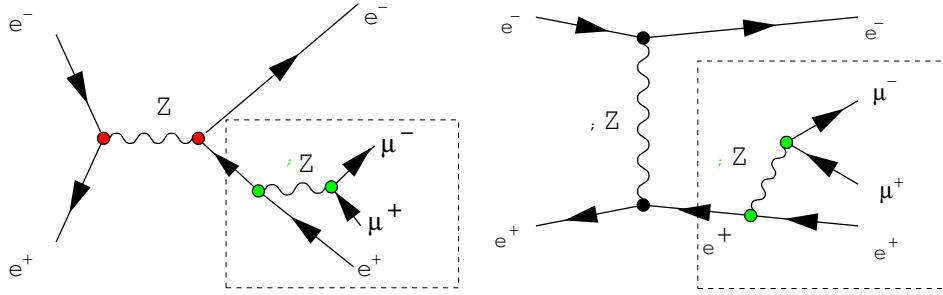
The four-vectors  $k_0$  and  $k_1$  are arbitrary choices satisfying  $k_0^2 = 0$ ,  $k_1^2 = -1$  and  $k_0 \cdot k_1 = 0$ . Any physical spinor state can be projected onto this basis. Using above relations this allows to calculate arbitrary spinor products  $\bar{u}(p_1, \lambda_1)u(p_2, \lambda_2)$  directly, i.e. without the necessity to evaluate traces of spinor products and  $\gamma$ -matrices of squared amplitudes. Details of the implementation can be found in [20, 51, 52].

The procedure to generate matrix elements can be summarized as follows:

1. Generation of all Feynman graphs for a given process and a given set of interactions, imposed by the specified interaction model (SM, MSSM, ADD, ...).
2. Translation of each graph into a complex function of momenta and helicity/spin states of the external particles.
3. Analytical simplification of generated functions, mainly by identifying and factoring out common pieces and the reusage of already calculated pieces. An example for such a manipulation is given in Fig. 2.1. Finally, the resulting formulas are stored in C++-library files, to be compiled and linked to the main code.

Possible colour structures within any amplitude are treated separately, i.e. in the first step all colour factors (the SU(3) structure constants  $f^{abc}$  and  $t_{ij}^a$ ) corresponding to the  $k$ -th amplitude  $A_k$  are collected in  $\mathbf{C}_k$ . Squared matrix elements can thus be written as

$$|M|^2 = \sum_{i,j} A_i \left[ \mathbf{C}_i \cdot \mathbf{C}_j^\dagger \right] A_j^*, \quad (2.3)$$



**Figure 2.1:** Factoring out common pieces of amplitudes with identical colour structure. In the example above, the parts within the boxes are identical, hence the two amplitudes can be added and the terms inside the box can be factored out.

and hence a colour matrix of complex numbers

$$c_{i,j} = \mathbf{C}_i \cdot \mathbf{C}_j^\dagger \quad (2.4)$$

is generated using a set of replacement rules for the colour algebra and stored.

All this is done during a initialization run before the actual calculation.

### Process management

Since typically many parton level processes contribute to jet cross section calculations, usually a long list of processes needs to be computed. The corresponding structure in AMEGIC++ is as follows:

- Any parton level process is represented by the class `Single_Process`,
- `Process_Group` contains a (possibly recursive) list of such processes or groups of processes.

All parton level processes sharing a specific common set of properties are grouped together in two or three levels of groups. In many cases there are subprocesses contributing to the same jet cross section which are very similar. Therefore AMEGIC++ applies a procedure to identify such processes in order to save computer resources and accelerate the calculation. The following checks are performed:

- Direct comparison of amplitudes: check for processes that have identical graphs, where all involved particles have the same masses, widths and underly the same interactions (with coupling constants that differ at most in a constant factor).  
Example: QCD processes that differ in quark flavours only.

- Numerical comparisons: check if the numerical result for a squared matrix element at a given phase space point is the same.  
Example: a quark is replaced by an anti-quark w.r.t. to the other process.

For a set of processes that can be identified by this it is enough to compute one to know them all. In such a case, the corresponding matrix element squared is calculated only once and then recycled by the other processes.

### 2.1.2 New models

The flexible structure of AMEGIC++ allows a relatively simple and completely general implementation of new physics models on the level of vertex Feynman rules. This means that interaction topologies and coupling constants have to be defined, and in case the model implies new Lorentz- or Dirac-structures in the interaction of non-scalar particles this structure has to be coded in a function of quantities provided by the helicity method. While in the original version of this formalism these quantities were solely spinor products (and anything else had to be decomposed into them), the implementation has been extended such, that it allows also to operate directly on vector-, tensor- and Rarita-Schwinger- (spin 3/2) structures. A detailed description how the coding of spin structures works for vertices involving vector- and tensor-particles is given in [52].

Below a number of Lagrangians are presented which have been implemented in the scope of this work.

#### Effective Gluon-Higgs couplings

At tree level the Higgs boson does not couple directly to massless fields such as the gluon. However, going to one-loop level the Higgs boson and the gluon do interact through a loop of massive quarks. The Higgs production cross section from gluon fusion is known for a long time [53],

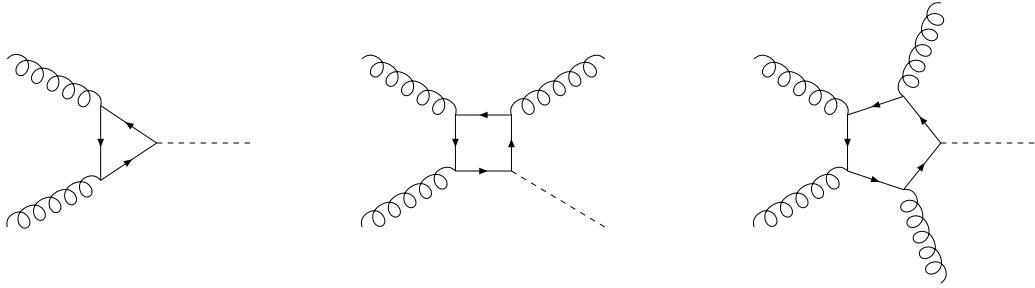
$$\sigma(gg \rightarrow H) = \frac{\alpha_S^2}{\pi} \frac{m_H^2}{256v^2} |A|^2 \delta(s - m_H^2), \quad (2.5)$$

where  $m_H$  is the Higgs boson mass,  $v$  the vacuum expectation value and

$$|A|^2 = \left| \sum_q \tau_q (1 + (1 - \tau_q) f(\tau_q)) \right|^2. \quad (2.6)$$

The sum in Eq. (2.6) runs over all quarks with mass  $m_q$ ,  $\tau_q = \frac{4m_q^2}{m_H^2}$  and

$$f(\tau_q) = \begin{cases} \left[ \sin^{-1}(\sqrt{\tau_q}) \right]^2 & \text{if } \tau_q \geq 1, \\ -\frac{1}{4} \left[ \log \left( \frac{1+\sqrt{1-\tau_q}}{1-\sqrt{1-\tau_q}} \right) - i\pi \right]^2 & \text{if } \tau_q < 1. \end{cases} \quad (2.7)$$



**Figure 2.2:** Effective vertices for gluon-higgs coupling

In the Standard Model the cross section is clearly dominated by the heaviest quark, the top. The relatively large value of the cross section, together with the high rate of gluons in a suitable energy range makes gluon fusion to be the dominant Higgs production channel at LHC.

For a Higgs boson with the mass  $m_H \ll 2m_t$  the top-quarks in the loop are virtual, allowing to interpret the loop as an *effective coupling* between the Higgs boson and gluons. The following effective Lagrangian can be defined [54],

$$\mathcal{L}_{ggH}^{\text{eff}} = g_{ggH} \frac{\alpha_S}{2\pi v} G_{\mu\nu}^a G_a^{\mu\nu} H , \quad (2.8)$$

where  $G_{\mu\nu}^a = \partial_\mu A_\nu^a - \partial_\nu A_\mu^a - gf^{abc} A_\mu^b A_\nu^c$  is the gluon field strength tensor. The effective coupling factor is defined to reproduce the cross section in Eq. (2.5), taking into account only the contribution from the top quark, i.e.

$$g_{ggH} = \tau_t (1 + (1 - \tau_t) f(\tau_t)) . \quad (2.9)$$

In many cases just the limit  $m_t \rightarrow \infty$  is considered,

$$g_{ggH}(m_t \rightarrow \infty) = \frac{2}{3} , \quad (2.10)$$

which underestimates the cross section slightly, so the results obtained are conservative estimations.

It is worth to note that the Lagrangian in Eq. (2.8) not only leads to a *gg – higgs* vertex, but also to *ggg – higgs* and *gggg – higgs* couplings, all displayed in Fig. 2.2. If higher order corrections to the higgs production via gluon fusion are calculated all have to be taken into account to obtain a *SU(3)* gauge invariant result.

All three vertices have been implemented for the standard model Higgs boson, as well as for the two neutral Higgs bosons  $h$  and  $H$ , predicted by supersymmetric models with a minimal

higgs sector. The Lorentz structure for the four- and five-point vertices are identical to the normal three- and four-gluon vertices, except for the fact that momentum conservation must now include the Higgs boson. The  $gg - higgs$  vertex comes with a new structure given by

$$k_1^\mu k_2^\nu - k_1 \cdot k_2 g^{\mu\nu} , \quad (2.11)$$

where  $k_1$  and  $k_2$  are the two gluon momenta.

Similar as for the gluon, an effective coupling induced at one-loop level also exists for the photon [55]. In this case the effective Lagrangian is given by

$$\mathcal{L}_{\gamma\gamma H}^{\text{eff}} = \frac{g_{\gamma\gamma H}}{v} F_{\mu\nu} F^{\mu\nu} H , \quad (2.12)$$

where  $F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu$  is the photon field strength tensor. Due to the absence of a non-abelian coupling for the photon only the  $\gamma\gamma - higgs$  vertex is induced. The effective coupling constant  $g_{\gamma\gamma H}$  can be determined from the  $higgs \rightarrow \gamma\gamma$  branching width. It has a similar expression as in Eq. 2.9, however now with major (and opposite sign) contributions from top-quark and  $W$ -boson loops.

## Anomalous Gauge Couplings

### Triple gauge couplings

In Ref. [56] the most general coupling between two charged vector bosons and a neutral vector boson has been presented,

$$\begin{aligned} \mathcal{L}_{WWV}/g_{WWV} &= ig_1^V (W_{\mu\nu}^\dagger W^\mu V^\nu - W_\mu^\dagger V_\nu W^{\mu\nu}) + i\kappa_V W_\mu^\dagger V_\nu W^{\mu\nu} \\ &+ \frac{i\lambda_V}{m_W^2} W_{\lambda\mu}^\dagger W_\nu^\mu V^{\nu\lambda} - g_4^V W_\mu^\dagger W_\nu (\partial^\mu V^\nu + \partial^\nu V^\mu) \\ &+ g_5^V \epsilon^{\mu\nu\rho\sigma} (W_\mu^\dagger \overleftrightarrow{\partial}_\rho W_\nu) V_\sigma + \frac{i\tilde{\kappa}_V}{2} \epsilon^{\mu\nu\rho\sigma} W_\mu^\dagger W_\nu V_{\rho\sigma} \\ &+ \frac{i\tilde{\lambda}_V}{2m_W^2} \epsilon^{\mu\nu\rho\sigma} W_{\mu\lambda}^\dagger W_\nu^\lambda V_{\rho\sigma} , \end{aligned} \quad (2.13)$$

where  $V^\mu$  stands for either the photon or the  $Z$  field,  $W^\mu$  is the  $W^-$  field,  $W_{\mu\nu} = \partial_\mu W_\nu - \partial_\nu W_\mu$ ,  $V_{\mu\nu} = \partial_\mu V_\nu - \partial_\nu V_\mu$ ,  $\tilde{V}_{\mu\nu} = \frac{1}{2} \epsilon_{\mu\nu\rho\sigma} V^{\rho\sigma}$  and  $(A \overleftrightarrow{\partial}_\mu B) = A(\partial_\mu B) - (\partial_\mu A)B$ . The overall coupling constants are

$$\begin{aligned} g_{WW\gamma} &= -e \quad \text{and} \\ g_{WWZ} &= -e \cot \theta_W . \end{aligned} \quad (2.14)$$

The Lagrangian is not specific for an extension of the Standard Model, instead it parameterizes seven possible independent coupling structures. The gauge symmetry imposed by

the Standard Model give strong constraints on the coefficients introduced in Eq. (2.13)<sup>2</sup>:

$$\begin{aligned} g_1 &= \kappa = 1, \\ \lambda &= g_4 = g_5 = \tilde{\kappa} = \tilde{\lambda} = 0. \end{aligned} \quad (2.15)$$

The implementation can be used to study signals and background in order to identify possible deviations from the Standard Model in a model independent way, or to put constraints on specific models.

### Quartic gauge couplings

In order to preserve unitarity in  $WW$  scattering the Standard Model requires a Higgs boson with a mass below 1 TeV. If such a particle is not found the situation can be described by means of the electroweak effective Lagrangian, as discussed in Refs. [57, 58, 59, 60]. Different models of electroweak symmetry breaking can be parameterized by this Lagrangian. This Lagrangian gives rise to a number of generic four-point interactions between weak bosons, not present in the Standard Model. The subset of the  $SU(2)$ -custodial symmetry conserving operators, usually referred to as  $\mathcal{L}_4$  and  $\mathcal{L}_5$  have been implemented,

$$\mathcal{L}_4 = \alpha_4 e^4 \left( \frac{1}{2} W_\mu^\dagger W^{\dagger\mu} W_\nu W^\nu + \frac{1}{2} (W_\mu^\dagger W^\mu)^2 + \frac{1}{c_W^2} W_\mu^\dagger Z^\mu W_\nu Z^\nu + \frac{1}{4c_W^4} (Z^\mu Z^\mu)^2 \right) \quad (2.16)$$

$$\mathcal{L}_5 = \alpha_5 \left( (W_\mu^\dagger W^\mu)^2 + \frac{1}{c_W^2} W_\mu^\dagger Z^\mu W_\nu Z^\nu + \frac{1}{4c_W^4} (Z^\mu Z^\mu)^2 \right). \quad (2.17)$$

Here,  $W^\mu$  and  $Z^\mu$  are the  $W^-$  and  $Z$  field, respectively,  $c_W = \cos \theta_W$  and  $\alpha_4$  and  $\alpha_5$  are parameters of some electroweak symmetry breaking theory.

### 2.1.3 Treatment for decay chains

Normally a process is defined by specifying all external particles. From this all Feynman graphs are evaluated, which allow to produce the given final state from the initial state. Unfortunately the number of graphs grows roughly factorially with the number of final state particles.

If one studies processes involving the production and decay of heavy particles the sum of all decay products would quickly exceed the number of accessible final state particles. This is mainly due the abundance of continuum graphs, i.e. graphs that do not contain the resonances one is actually interested in. These graphs in many cases only have a tiny contribution to the cross section, especially, as often done in experimental analyses, if phase

---

<sup>2</sup>These are the elementary coefficient at tree level; higher order corrections may induce effective nonzero contributions proportional to  $\lambda$ ,  $g_4$ ,  $g_5$ ,  $\tilde{\kappa}$  and  $\tilde{\lambda}$ .

space cuts are applied that enhance the resonance w.r.t. background. Examples for processes where this might be a concern are the production of top-quarks, or the production of supersymmetric particles, which in most scenarios involve long and complicated decay chains.

The simplest way to calculate cross sections for such cases would be to completely factorize the decay from the production of the heavy intermediate particles, i.e. they are produced on their mass shells and decayed independently. This naive approach, usually called *narrow width approximation*, however neglects some potentially important effects: Firstly, if the intermediate particle has a spin structure the correlation between the production and the decay is lost. Secondly, since the intermediate particles have been produced at a fixed mass, the invariant mass of all decay products is sharply peaked at this value and thus neglecting possibly important off-shell properties.

To directly cope with these effects in AMEGIC++ a production process with decay chains can be specified, which are handled as follows:

1. Graphs are generated separately for the subprocesses, but are combined again at this level,
2. Amplitudes are generated implying the full propagator for intermediate particles.

Clearly, this approach keeps the spin correlation between the production and the decay process. Also the invariant mass of the decay product has the correct distribution according to the mass and the width of the intermediate particle. Of course the approximation is only as good as the assumption that the continuum graphs are negligible. In some cases the matrix element is not fully gauge invariant any more with an invariance violation up to the order of the neglected graphs.

### 2.1.4 Accessible processes

All methods to generate matrix elements using AMEGIC++ are implemented in a general and process- and model-independent way. From this perspective there is no general limit on the complexity of the calculation. Practically one is of course limited by the available computer resources. On a modern PC, matrix elements with roughly up to  $10^4$  amplitudes are feasible, which is in most cases enough for scattering processes with 6-7 final state particles. For pure QCD, especially  $n$ -gluon scattering, the limit is already reached with 4-5 final state particles, which is due to the complicated non-abelian interaction structure.

Applying the decay chain treatment, much more complicated final states can be calculated.

The limit is again given by the number of contributing amplitudes. Processes such as

$$\begin{aligned} p\bar{p} &\rightarrow t[\rightarrow b\mu^+\nu_\mu] \bar{t}[\rightarrow \bar{b}\mu^-\bar{\nu}_\mu] + 2\text{jets} \quad \text{or} \\ p p &\rightarrow W^-[\rightarrow \dots] W^+[\rightarrow \dots] W^-[\rightarrow \dots] W^+[\rightarrow \dots] + 2\text{jets} \end{aligned}$$

are feasible.

The computation of all parton level processes, contributing to a hadron level cross section, is typically practicable for a total number of 6 partons (initial and final state).

## 2.2 Recursion relations based on MHV-amplitudes

Although the methods and algorithms, used in AMEGIC++ to generate matrix elements, provide a fast and very general implementation, it is not the optimal strategy to face processes which involve a large number of strong interactions. The reason is, as already stated, the large number and complications related to the non-abelian gauge theory.

In this chapter a technique will be discussed that allows to compute QCD-dominated matrix elements much more efficiently as it would be possible by evaluating Feynman diagrams.

### 2.2.1 Partial amplitudes and colour decomposition

Any perturbative QCD scattering amplitude  $\mathcal{A}$  can be written as a sum of terms, which factorize into two components, one only depending on the gauge structure and one only depending on the kinematics. Such a factorization is called colour decomposition. For tree-level  $n$ -gluon amplitudes several colour decompositions exist. A very intuitive one is given by [61]

$$\mathcal{M}(ng) = \sum_{P(2,\dots,n)} \text{Tr}(t^{a_1} t^{a_2} \dots t^{a_n}) A(1, 2, \dots, n), \quad (2.18)$$

where  $a_1, a_2, \dots, a_n$  ( $a_i = 1, \dots, 8$ ) label the colours of the gluons,  $t^a$  are the fundamental-representation matrices of  $SU(3)$ . The sum runs over all  $(n-1)!$  permutations of  $(2, \dots, n)$ . Each trace (of the given permutation of  $t^a$ ) corresponds to a particular colour structure. The factor  $A(1, 2, \dots, n)$ , associated with each colour structure is called a partial (or colour-ordered) amplitude. It depends only on the four-momenta  $p_i$  and polarizations of the gluon (all simply labeled by  $i$  in the argument of the amplitude). The point for doing such a decomposition is, as it will be shown below, that partial amplitudes are usually far easier to compute than full amplitudes. Furthermore, partial amplitudes are gauge invariant quantities. The decomposition in Eq. (2.18) is called the fundamental-representation decomposition.

For multi-gluon matrix elements also the following, called the adjoint-representation decomposition, has been introduced in Refs. [62, 63]:

$$\mathcal{M}(ng) = \sum_{P(2, \dots, n-1)} (F^{a_2} F^{a_3} \cdots F^{a_{n-1}})_{a_n}^{a_1} A(1, 2, \dots, n) , \quad (2.19)$$

where  $(F^a)_c^b = -if^{abc}$  are the adjoint-representation matrices of  $SU(3)$ . The partial amplitudes are exactly the same as in Eq. (2.18), however, the sum now only runs over  $(n-2)!$  permutations of  $(2, \dots, n-1)$ , thus reducing the number of partial amplitudes to be calculated. This decomposition directly incorporates the fact that the partial amplitudes, used in Eq. (2.18), are not all linear independent. The number of amplitudes to be calculated there can be reduced imposing the dual Ward identity,

$$0 = A(1, 2, \dots, n) + A(1, 2, \dots, n, n-1) + \dots + A(1, n, 2, \dots, n-1) . \quad (2.20)$$

Thus, for the implementation of multi-gluon processes discussed in this chapter the adjoint-representation decomposition is the method of choice.

Recently a third decomposition for multi-gluon matrix elements has been introduced, the colour-flow decomposition [64]:

$$\mathcal{M}(ng) = \sum_{P(2, \dots, n)} \delta_{j_2}^{i_1} \delta_{j_3}^{i_2} \cdots \delta_{j_1}^{i_n} A(1, 2, \dots, n) \quad (2.21)$$

In this prescription the  $SU(3)$  gluon field is treated as a  $3 \times 3$  matrix  $(A_\mu)^{i\bar{j}}$  rather than a one index field  $A_\mu^a$ .

Except for the adjoint-representation decomposition the relations can be generalized for processes including quarks. The fundamental-representation decomposition for processes including one quark-line reads[61]

$$\mathcal{M}(\bar{q}q + ng) = \sum_{P(1, \dots, n)} (t^{a_1} t^{a_2} \cdots t^{a_n})_{j_q}^{i_q} A(q, 1, 2, \dots, n, \bar{q}) , \quad (2.22)$$

where the sum runs over all gluon permutations. The indices  $i_q$  ( $j_q$ ) label the colour of the quark (anti-quark).

Details of the colour-flow decomposition for processes with one and two quark lines can be found in Ref. [64].

## 2.2.2 MHV amplitudes and the CSW technique

In Ref. [65] it was found that a certain class of colour-ordered tree-level QCD amplitudes, the so-called maximally helicity violating (MHV) amplitudes, is described by impressively simple formulae. An explanation for this has been given in Ref. [66], proposing that the

usual momentum space scattering amplitudes can be Fourier transformed to Twistor space, where they are supported on certain holomorphic curves. Results such as the holomorphy of the tree-level MHV amplitudes, as well as certain differential equations obeyed by higher order amplitudes are direct consequences. At tree level one gains very simple formulae for all amplitudes with at most two negative (positive) helicity partons. Amplitudes with less than two partons of each helicity vanish.

$$\begin{aligned} A_n(1^+, 2^+, \dots, n^+) &= 0, \\ A_n(1^-, 2^+, \dots, n^+) &= 0, \end{aligned} \quad (2.23)$$

where in this notation all momenta and corresponding helicities, labeled explicitly in the superscript, are considered as incoming. The first non-vanishing amplitudes, the MHV amplitudes, contain exactly two partons with opposite helicity and can be written as simple holomorphic (anti-holomorphic) functions. In the notation of Ref. [65] the  $n$ -gluon MHV-amplitudes read<sup>3</sup>

$$A_n(1^+, \dots, i^-, \dots, j^-, \dots, n^+) = \frac{i \langle i j \rangle^4}{\langle 1 2 \rangle \langle 2 3 \rangle \dots \langle n-1 n \rangle \langle n 1 \rangle}, \quad (2.24)$$

and similar for reversed helicities, dubbed  $\overline{\text{MHV}}$ -Amplitudes,

$$A_n(1^-, \dots, i^+, \dots, j^+, \dots, n^-) = \frac{i [i j]^4}{[1 2] [2 3] \dots [n-1 n] [n 1]}. \quad (2.25)$$

The notations  $\langle i j \rangle$  and  $[i j]$  represent spinor products. They can be translated to the notation used earlier in this chapter:

$$\begin{aligned} \langle i j \rangle &\equiv \bar{u}(p_i, +)u(p_j, -) \text{ and} \\ [i j] &\equiv \bar{u}(p_i, -)u(p_j, +), \end{aligned} \quad (2.26)$$

respectively, and thus be calculated employing exactly the same spinor basis that is already used in AMEGIC++.

MHV-amplitudes with one external quark pair of indices 1 and  $n$  for the quark and anti-quark, respectively, are

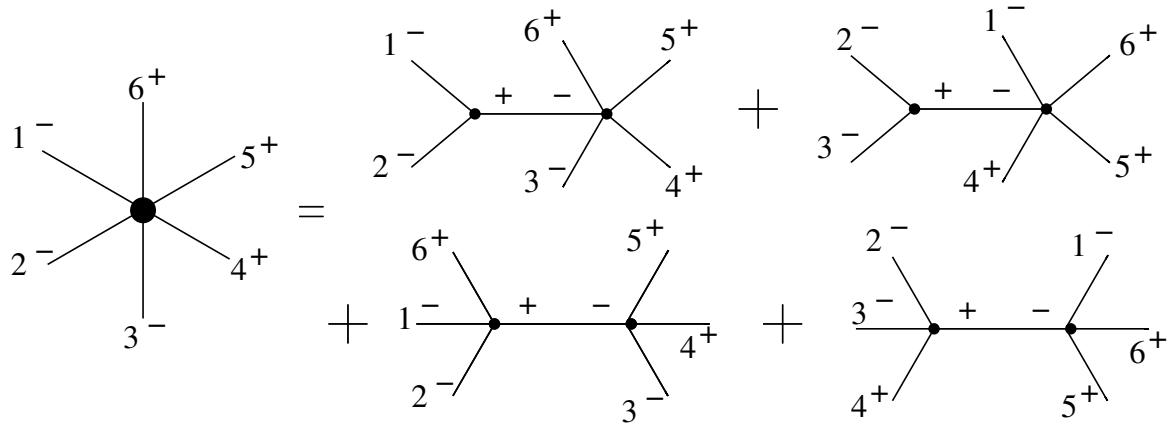
$$\begin{aligned} A_n(q^+, 2^+, \dots, i^-, \dots, n-1^+, \bar{q}^-) &= \frac{i \langle i q \rangle \langle i \bar{q} \rangle^3}{\langle 1 2 \rangle \dots \langle n-1 n \rangle \langle n 1 \rangle}, \\ A_n(q^-, 2^+, \dots, i^-, \dots, n-1^+, \bar{q}^+) &= \frac{-i \langle i q \rangle^3 \langle i \bar{q} \rangle}{\langle 1 2 \rangle \dots \langle n-1 n \rangle \langle n 1 \rangle}. \end{aligned} \quad (2.27)$$

For two quark lines the corresponding MHV amplitudes read

$$\begin{aligned} A_n\left(q^h, \dots, \bar{q}^{-h'}, q'^{h'}, \dots, \bar{q}^{-h}\right) &= \frac{i A_0(h, h') \langle q \bar{q}' \rangle \langle q', \bar{q} \rangle}{\langle 1 2 \rangle \dots \langle n-1 n \rangle \langle n 1 \rangle}, \\ A_n\left(q^h, \dots, \bar{q}^{-h}, q'^{h'}, \dots, \bar{q}'^{-h'}\right) &= \frac{i A_0(h, h') \langle q \bar{q}' \rangle \langle q', \bar{q} \rangle}{\langle 1 2 \rangle \dots \langle n-1 n \rangle \langle n 1 \rangle}. \end{aligned} \quad (2.28)$$

---

<sup>3</sup>Coupling constants have been dropped.



**Figure 2.3:** CSW recursion relation.

where  $A_0(h, h')$  depends on the helicity configuration of the quarks

$$\begin{aligned} A_0(+, +) &= \langle \bar{q}, \bar{q}' \rangle^2, A_0(+, -) = -\langle \bar{q}, q' \rangle^2, \\ A_0(-, +) &= -\langle q, \bar{q}' \rangle^2, A_0(-, -) = -\langle q, q' \rangle^2. \end{aligned} \quad (2.29)$$

The corresponding  $\overline{\text{MHV}}$ -amplitudes with reversed helicities are obtained by the replacement  $\langle \rangle \rightarrow [ ]$ .

In Ref. [40] Cachazo, Svrček and Witten (CSW) proposed an algorithm, how partial amplitudes with more than two helicities of each kind can be derived from a recurrence relation, where the full amplitude is split into colour-ordered  $n$ -point vertices which are connected by scalar propagators. The vertices are off-shell continuations of MHV-amplitudes. The procedure is as follows:

1. A non-MHV amplitude can be decomposed into amplitudes, containing either MHV-vertices or  $\overline{\text{MHV}}$ -vertices.
2. Connecting propagators have opposite helicities entering into the vertices; they come with a factor of  $1/q^2$ , where  $q$  is the propagator momentum.
3. All possible decompositions contribute.
4. The ordering of the full amplitude is preserved in the decomposition.

The decomposition into MHV-vertices is illustrated in Fig. 2.3 for an example six-gluon amplitude. The corresponding partial amplitude reads

Process	Limitations
n gluons	$n \leq 9$
2 quarks + (n-2) gluons	$n \leq 9$
4 quarks + (n-4) gluons	$n \leq 9$
2 quarks + 2 leptons + (n-4) gluons	$n \leq 9$

**Table 2.1:** List of processes, available in AMEGIC++ using MHV-amplitude based recursion relations.

$$\begin{aligned}
A_6(1^-, 2^-, 3^-, 4^+, 5^+, 6^+) = & A_3(1^-, 2^-, p_{12}^+) \frac{1}{p_{12}^2} A_5(p_{12}^-, 3^-, 4^+, 5^+, 6^+) \\
& + A_3(2^-, 3^-, p_{23}^+) \frac{1}{p_{23}^2} A_5(p_{23}^-, 4^+, 5^+, 6^+, 1^-) \\
& + A_4(6^+, 1^-, 2^-, p_{126}^+) \frac{1}{p_{126}^2} A_4(p_{126}^-, 3^-, 4^+, 5^+) \\
& + A_4(2^-, 3^-, 4^+, p_{234}^+) \frac{1}{p_{234}^2} A_4(p_{234}^-, 5^+, 6^+, 1^-) , \quad (2.30)
\end{aligned}$$

where  $p_{ij} = p_i + p_j$  and  $p_{ijk} = p_i + p_j + p_k$ .

It is worth to notice the following:

- The expression  $A_3(1^-, 2^-, p^+)$  exists only as an MHV-vertex for a off-shell momentum  $p$ , as an on-shell amplitude it is identical to zero, cf. Eq. 2.23.
- To decompose a partial amplitude with  $m$  times the helicity “-” (“+”), it is split into  $(m - 1)$  MHV- ( $\overline{\text{MHV}}$ -)vertices, where  $(m - 2)$  propagators must be introduced. The most efficient decomposition is thus given by decomposing for the least number of identical helicities.

Using this technique, amplitudes with an arbitrary number of negative (positive) helicity gluons can be computed. The above formalism was initially proposed for the computation of pure gluonic amplitudes, but it has been extended to many other processes. From the general expression for an MHV supervertex in the  $\mathcal{N} = 4$  supersymmetric Yang-Mills theory, one can read off, for example, appropriate terms with massless fermions [67, 68].

### 2.2.3 Accessible processes

The CSW formalism has been implemented into the matrix element generator AMEGIC++ for the QCD-processes stated above and for a generalization [69], which includes processes with a weak boson. The processes that are currently available and technically feasible are quoted in Tab. 2.1. In the second column the current limit on the number of external

particles is shown. For all processes numerical results have been checked to be identical to the standard techniques employed in AMEGIC++. The CSW implementation can be used as a supplement, wherever the resulting matrix elements are more efficient. A detailed comparison of the efficiencies can be found in chapter 4.

## 2.3 Colour dressed Berends-Giele recursion relations

It has been pointed out, for example in Refs. [43, 70, 71], that the calculation of multi-particle amplitudes is substantially simplified when employing Berends-Giele type recursive relations. One main reason for the simplification is that these relations allow the reuse of basic building blocks of an amplitude, which are the  $m$ -particle internal off-shell currents. Another reason is that they can be easily rewritten to include three-particle vertices only. In the following it will be briefly illuminated why this is a major advantage.

A theory with only one particle type and a recursion relation for internal  $n$ -particle currents is assumed, which is of the functional form

$$J_n(\pi) = P_n(\pi) \sum_{N=1}^n \sum_{\mathcal{P}_N(\pi)} V_N(\pi_1, \dots, \pi_N) J_{i_1}(\pi_1) \dots J_{i_N}(\pi_N) . \quad (2.31)$$

Here  $J_m$  denote unordered  $m$ -particle currents, while  $V_N$  are  $N + 1$ -point vertices and  $P_n$  is a propagator term. The two sums run over all possible vertex types  $V_N$  and all (unordered) partitions  $\mathcal{P}_N(\pi)$  of the set of particles  $\pi$  into  $N$  (unordered) subsets, respectively. The full  $n + 1$ -particle scattering amplitude can be constructed by putting an  $n$ -particle internal off-shell current on-shell and contracting the remaining quantity with the corresponding external one-particle current,

$$A_{n+1}(\pi) = \varepsilon(i) \frac{1}{P_n(\pi \setminus i)} J_n(\pi \setminus i) . \quad (2.32)$$

Now the contribution of vertices of  $N + 1$  external legs to the computation of an  $n$ -particle off-shell current is considered. The number of vertices to evaluate per  $m$ -particle sub-current is the Stirling number of the second kind  $S(m, N)$ , corresponding to the number of partitions of a set  $\pi$  of  $m$  integers into  $N$  subsets. The total number of  $N + 1$ -particle vertices to be calculated thus becomes

$$\sum_{m=N}^n \binom{n}{m} S(m, N) . \quad (2.33)$$

Since the Stirling numbers  $S(m, N)$  are zero for  $m < N$ , the sum can be extend down to

zero, leading to

$$\begin{aligned}
\sum_{m=0}^n \binom{n}{m} S(m, N) &= \sum_{m=0}^n \binom{n}{m} \frac{1}{N!} \sum_{i=0}^N (-1)^i \binom{N}{i} (N-i)^m \\
&= \frac{1}{N!} \sum_{i=0}^N (-1)^i \binom{N}{i} (N+1-i)^n \\
&= \frac{1}{(N+1)!} \sum_{i=0}^N (-1)^i \binom{N+1}{i} (N+1-i)^{n+1} = S(n+1, N+1) .
\end{aligned} \tag{2.34}$$

The question is now, whether a milder growth in computational complexity can be obtained if all  $N+1$ -particle vertices occurring in Eq. (2.31) are decomposed in terms of two or more vertices with fewer number of external legs. When doing so, additional pseudo-particles reflecting the structure of the decomposed vertex must be introduced. Hence the contribution arising from the presence of these pseudo-particles have to be considered as well. The problem can be simplified by assuming that there is only one additional pseudo-particle, which obeys a completely independent recursion relation, such that the full contribution of an  $N+1$ -particle vertex, which is decomposed into an  $M+1$ - and an  $N-M+1$ -particle vertex becomes

$$S(n+1, N+1) \rightarrow S(n+1, M+1) + S(n+1, N-M+1) , \tag{2.35}$$

which can be either bigger or smaller than  $S(n+1, N+1)$ , depending on  $n$ ,  $N$  and  $M$ . With increasing  $n$ , however the right hand side is always smaller such that the vertex decomposition becomes clearly advantageous. Similar arguments hold when introducing more than one pseudo-particle.

From this simple consideration it can be deduced that the aim of any recursive formulation of interaction models must be to reduce the number of external lines at interaction vertices to the lowest possible. Below it will be demonstrated, that within the Standard Model it is possible to reduce  $N_{\max}$  to two, which is the lowest possible number at all. For QCD interactions the results of Ref. [43] are employed, where this task has already been performed and the original Berends-Giele recursive relations have been reformulated to incorporate colour.

### 2.3.1 General form of the recursion

In the following  $\mathcal{J}_\alpha(\pi)$  denotes an unordered SM current of type  $\alpha$ , which receives contributions from all Feynman graphs having as external particles the on-shell SM particles in the set  $\pi$  and one internal particle, described by this current. The index  $\alpha$  in this respect is a multi-index, carrying information on all quantum numbers and eventually on the

pseudo-particle character of the particle. Special currents are given by the external particles' currents. They correspond to external scalars, spinors and polarization vectors, see Appendix A. For them there is also only one multi-index  $\alpha = \alpha_i$  associated with the external particle, whereas in the general case multiple multi-indices may lead to non-vanishing internal currents. Assuming only three-point vertices, any internal SM particle and pseudo-particle off-shell current can be written as

$$\mathcal{J}_\alpha(\pi) = P_\alpha(\pi) \sum_{\mathcal{V}_\alpha^{\alpha_1, \alpha_2}} \sum_{\mathcal{P}_2(\pi)} \mathcal{S}(\pi_1, \pi_2) \mathcal{V}_\alpha^{\alpha_1, \alpha_2}(\pi_1, \pi_2) \mathcal{J}_{\alpha_1}(\pi_1) \mathcal{J}_{\alpha_2}(\pi_2) , \quad (2.36)$$

where  $P_\alpha(\pi)$  is a propagator term depending on the particle type  $\alpha$  and the set  $\pi$ ,  $\mathcal{V}_\alpha^{\alpha_1, \alpha_2}(\pi_1, \pi_2)$  is a vertex depending on  $\alpha$ , the decomposition of the set  $\pi$  into disjoint subsets  $\pi_1$  and  $\pi_2$  and the indices  $\alpha_1$  and  $\alpha_2$  and  $\mathcal{S}(\pi_1, \pi_2)$  is the symmetry factor discussed in Sec. 2.3.4. Superscripts in this context refer to incoming particles, the subscript refers to the particle emerging from the vertex. Correspondingly upper multi-indices denote incoming, lower multi-indices denote outgoing quantities. The sums run over all vertices in the reformulated Standard Model and all unordered partitions  $\mathcal{P}_2$  of the set  $\pi$  into two subsets, respectively. A full unordered  $n$ -particle scattering amplitude is then given by

$$\mathcal{A}(\pi) = \mathcal{J}_{\alpha_n}(n) \frac{1}{P_{\bar{\alpha}_n}(\pi \setminus n)} \mathcal{J}_{\bar{\alpha}_n}(\pi \setminus n) , \quad (2.37)$$

where  $\bar{\alpha}$  denotes a set of reversed particle properties. It has been proved in Ref. [43] that the above form is correct for pure gluonic scattering amplitudes once the four gluon vertex is suitably decomposed into two vertices involving an internal antisymmetric tensor pseudo-particle. This procedure is briefly recalled below.

### 2.3.2 Decomposition of four-point vertices

In Ref. [41] Berends and Giele proposed a method to compute partial amplitudes in a recursive fashion. The basic idea is that, according to the Feynman rules of QCD, an internal  $n$ -gluon current is defined by all contributing Feynman graphs with  $n$  external on-shell gluons and one off-shell gluon.

$$\begin{aligned} J_\mu(1, 2, \dots, n) &= \frac{-ig_{\mu\nu}}{P_{1,n}^2} \left\{ \sum_{k=1}^{n-1} V_3^{\nu\rho\sigma}(P_{1,k}, P_{k+1,n}) J_\rho(1, \dots, k) J_\sigma(k+1, \dots, n) \right. \\ &\quad \left. + \sum_{j=1}^{n-2} \sum_{k=j+1}^{n-1} V_4^{\nu\rho\sigma\lambda} J_\rho(1, \dots, j) J_\sigma(j+1, \dots, k) J_\lambda(k+1, \dots, n) \right\} . \end{aligned} \quad (2.38)$$

Here  $p_i$  denote the momenta of the gluons,  $P_{i,j} = p_i + \dots + p_j$  and  $V_3^{\nu\rho\sigma}(P_{1,k}, P_{k+1,n})$  and  $V_4^{\nu\rho\sigma\lambda}$  are the colour ordered three and four-gluon vertices defined according to Ref. [72],

$$\begin{aligned} V_3^{\nu\rho\sigma}(p, q) &= i \frac{g_s}{\sqrt{2}} (g^{\rho\sigma}(p - q)^\mu + g^{\sigma\nu}(2p + q)^\rho - g^{\nu\rho}(2q + p)^\sigma) , \\ V_4^{\nu\rho\sigma\lambda} &= i \frac{g_s^2}{2} (2g^{\nu\sigma}g^{\rho\lambda} - g^{\nu\rho}g^{\sigma\lambda} - g^{\nu\lambda}g^{\rho\sigma}) . \end{aligned} \quad (2.39)$$

The full colour-ordered  $n$ -gluon amplitude  $A(1, \dots, n)$  is then obtained by putting the  $n-1$ -particle off-shell current  $J_{n-1}(1, \dots, n-1)$  on-shell and contracting it with the external polarization  $\varepsilon_n$ . Employing the tensor-gluon vertex

$$V_T^{\mu\nu\rho\sigma} = \frac{i}{2} \frac{g_s}{\sqrt{2}} (g^{\mu\rho}g^{\nu\sigma} - g^{\mu\sigma}g^{\nu\rho}) , \quad (2.40)$$

and the tensor “propagator”

$$-iD_{\mu\nu}^{\rho\sigma} = -i(g_\mu^\rho g_\nu^\sigma - g_\mu^\sigma g_\nu^\rho) , \quad (2.41)$$

the recursion can be reformulated to give

$$\begin{aligned} J^\mu(1, 2, \dots, n) &= \frac{-ig_{\mu\nu}}{P_{1,n}^2} \sum_{k=1}^{n-1} \left\{ V_3^{\nu\rho\sigma}(P_{1,k}, P_{k+1,n}) J_\rho(1, \dots, k) J_\sigma(k+1, \dots, n) \right. \\ &\quad + V_T^{\nu\rho\alpha\beta} J_\rho(1, \dots, k) J_{\alpha\beta}(k+1, \dots, n) \\ &\quad \left. + V_T^{\sigma\nu\alpha\beta} J_{\alpha\beta}(1, \dots, k) J_\sigma(k+1, \dots, n) \right\} \end{aligned} \quad (2.42)$$

and

$$J^{\alpha\beta}(1, 2, \dots, n) = 6 - iD_{\gamma\delta}^{\alpha\beta} \sum_{k=1}^{n-1} V_T^{\gamma\delta\rho\sigma} J_\rho(1, \dots, k) J_\sigma(k+1, \dots, n) , \quad (2.43)$$

for the gluon and tensor pseudo-particle currents, respectively. Since no external tensor currents exist, all tensor currents with one particle index only are defined as zero.

A decomposition of all remaining four-point vertices in the Standard Model, coming from electroweak interactions is described in Appendix A.1.

### 2.3.3 Colour dressed amplitudes

The next step towards full amplitudes is to include colour. Therefore the colour-flow decomposition, for  $n$ -gluon amplitudes already stated in Eq. (2.21) is used. Although this decomposition is not minimal, it is suited especially for Monte Carlo event generation, sampling over the degrees of freedom in colour instead of summing.

The reason is the following: To obtain a squared matrix element out of any decomposition into partial amplitudes, such as Eqs. (2.18)-(2.22), naively all partial amplitudes must be calculated before squaring in order to catch all quantum interferences. This, however, quickly gets extremely tedious for complex final states, since their number grows factorially with the number of external partons. The situation can be improved by fixing the quantum numbers of the external partons instead of implicitly sum over them when calculating colour factors. For such a colour assignment only a fraction off all partial amplitudes would contribute. In Ref. [64] it has been demonstrated that for the colour-flow decomposition the average number of non-vanishing partial amplitudes per colour assignment has the slowest growth with the number of partons.

Colour dressing means to include the colour factors in Eq. (2.21) into the recursion relation, such that the full (unordered) amplitude is calculated at once. Therefore the recursion relation must produce all partial amplitudes, contributing to a colour assignment. Following Ref. [43] colour dressed gluon and tensor pseudo-particle currents  $\mathcal{J}_{\mu I\bar{J}}$  and  $\mathcal{J}_{\alpha\beta I\bar{J}}$  are defined by

$$\begin{aligned}\mathcal{J}_{\mu I\bar{J}}(1, \dots, n) &= \sum_{\sigma \in S_n} \delta_{I\bar{\sigma}_2} \delta_{i_{\sigma_2} \bar{\sigma}_3} \dots \delta_{i_{\sigma_n} \bar{J}} J_\mu(\sigma_1, \dots, \sigma_n) , \\ \mathcal{J}_{\alpha\beta I\bar{J}}(1, \dots, n) &= \sum_{\sigma \in S_n} \delta_{I\bar{\sigma}_2} \delta_{i_{\sigma_2} \bar{\sigma}_3} \dots \delta_{i_{\sigma_n} \bar{J}} J_{\alpha\beta}(\sigma_1, \dots, \sigma_n) .\end{aligned}\quad (2.44)$$

Denoting the set  $(1, \dots, n)$  of  $n$  particles by  $\pi$ , the following recursive relations for these currents are obtained:

$$\begin{aligned}\mathcal{J}_{\mu I\bar{J}}(\pi) &= D_{\mu I\bar{J}}^{\nu H\bar{G}}(\pi) \left\{ \sum_{\mathcal{P}_2(\pi)} \mathcal{V}_{\nu H\bar{G}}^{\rho K\bar{L}, \sigma M\bar{N}}(\pi_1, \pi_2) \mathcal{J}_{\rho K\bar{L}}(\pi_1) \mathcal{J}_{\sigma M\bar{N}}(\pi_2) \right. \\ &\quad \left. + \sum_{\pi \in \mathcal{OP}(n)} \mathcal{V}_{\nu H\bar{G}}^{\rho K\bar{L}, \alpha\beta M\bar{N}} \mathcal{J}_{\rho K\bar{L}}(\pi_1) \mathcal{J}_{\alpha\beta M\bar{N}}(\pi_2) \right\} , \\ \mathcal{J}_{\alpha\beta I\bar{J}}(\pi) &= D_{\alpha\beta I\bar{J}}^{\gamma\delta H\bar{G}} \sum_{\mathcal{P}_2(\pi)} \mathcal{V}_{\gamma\delta H\bar{G}}^{\rho K\bar{L}, \sigma M\bar{N}} \mathcal{J}_{\rho K\bar{L}}(\pi_1) \mathcal{J}_{\sigma M\bar{N}}(\pi_2) .\end{aligned}\quad (2.45)$$

Here, the colour dressed gluon and tensor pseudo-particle vertices have been defined as

$$\mathcal{V}_{\nu H\bar{G}}^{\rho K\bar{L}, \sigma M\bar{N}}(\pi_1, \pi_2) = \delta_{\bar{G}}^{\bar{L}} \delta^{K\bar{N}} \delta_H^M V_{3\nu}^{\rho\sigma}(\pi_1, \pi_2) + \delta_H^K \delta^{\bar{L}M} \delta_{\bar{G}}^{\bar{N}} V_{3\nu}^{\sigma\rho}(\pi_2, \pi_1) , \quad (2.46)$$

and

$$\mathcal{V}_{\gamma\delta H\bar{G}}^{\rho K\bar{L}, \sigma M\bar{N}} = \delta_{\bar{G}}^{\bar{L}} \delta^{K\bar{N}} \delta_H^M V_{T\gamma\delta}^{\rho\sigma} + \delta_H^K \delta^{\bar{L}M} \delta_{\bar{G}}^{\bar{N}} V_{T\gamma\delta}^{\sigma\rho} . \quad (2.47)$$

A complete proof of these relations can be found in Ref. [43]. The above procedure of colour dressing can easily be generalized to QCD processes including quarks. Since no further elementary QCD four-point interactions exists, no further vertex decomposition has to be performed and therefore no new current types are introduced.

### 2.3.4 Prefactors of amplitudes with external fermions

When calculating currents with an arbitrary number of possibly indistinguishable external fermions, it has to be taken into account, that each Feynman diagram contains a prefactor

$$\mathcal{S} = (-1)^{P_f(\sigma_1, \dots, \sigma_n)}, \quad (2.48)$$

according to the number of fermion permutations  $P_f$  in the external particle assignment  $\vec{\sigma} = (\sigma_1, \dots, \sigma_n)$ . This prefactor must now be defined on a local basis in order to avoid the proliferation of information on different  $\vec{\sigma}$ . To do so, it is sufficient to note that Eq. (2.48) holds on the level of interaction vertices as well. More precisely the local prefactor  $\mathcal{S}(\pi_1, \pi_2)$  of Eq. (2.36) is defined as

$$\mathcal{S}(\pi_1, \pi_2) = (-1)^{P_f(\pi_1, \pi_2)}. \quad (2.49)$$

Here  $P_f(\pi_1, \pi_2)$  counts the number of fermion permutations that is needed to restore a predefined, for example ascending index ordering, when combining the sets  $\pi_1$  and  $\pi_2$  into the set  $\pi = \pi_1 \oplus \pi_2$ . It is easy to see that upon iterating this procedure, correct relative prefactors  $\mathcal{S}$  are obtained for each amplitude.

### 2.3.5 Accessible processes

The Standard Model has been fully implemented so far. All methods for the matrix element generation have been implemented in a general way, such that the limit is only given by the available computer resources. Since the recursion relations have been optimized for processes with a large multiplicity of particles it can be expected manage processes beyond the limits of AMEGIC++. The most significant gain is given for purely strong interacting scattering processes, which is a result of both, the optimized scaling behaviour of the complexity and the used colour treatment. For instance, gluon scattering processes up to  $2 \rightarrow 11$  can be calculated.

# 3 Monte-Carlo phase space integration

Apart from being able to calculate matrix elements at a given phase space point, it is also highly important to be able to integrate efficiently over the phase space. To derive any physical observables typically integrals over final state particles have to be calculated,

$$\int d\Phi_m(p_1, \dots, p_m) |M_m(p_1, \dots, p_m)|^2 F(p_1, \dots, p_m) , \quad (3.1)$$

where  $|M_m(p_1, \dots, p_m)|^2$  is a matrix element with  $m$  final state particles as discussed in the previous chapter, and,

$$d\Phi_m(p_1, \dots, p_m) = \left[ \prod_{i=1}^m \frac{d^4 p_i}{(2\pi)^4} \delta(p_i^2 - m_i^2) \Theta(p_i^0) \right] \delta^4 \left( p^{in} - \sum_{i=1}^n p_i \right) \quad (3.2)$$

being the corresponding  $m$ -particle phase space element<sup>1</sup>. The function  $F(p_1, \dots, p_m)$  defines the observable. For a total cross section it would simply contain  $\Theta$ -functions, defining phase space cuts; more generally it could be any function of final state momenta to define a physical observable.

In general phase space integrals are much too complicated to be solved analytically. For the given example in Eq. (3.1) the phase space has a dimension of  $3m - 4$ . The large dimension clearly favours the Monte-Carlo integration method to be the method of choice. There, an estimate for an integral is generated,

$$\int f dV \rightarrow V \langle f \rangle , \quad (3.3)$$

where  $V$  is the phase space volume  $\int dV$  and  $\langle f \rangle = (\sum f_i) / N$  is the average of the integrand, calculated at  $N$  different (random) points. The statistical error for the estimate is given by

$$E = V \sqrt{\frac{S}{N}} , \quad (3.4)$$

---

<sup>1</sup> The integral in Eq. (3.1) is given for fixed initial state with momenta  $p_a + p_b = p_{in}$ . For particle reactions with non-elementary beam particles, such as hadrons, an additional integral over incoming momentum fractions, convoluting the integral (3.1) with parton density functions, is necessary.

with the variance

$$S = \langle f^2 \rangle - \langle f \rangle^2 . \quad (3.5)$$

A general and simple implementation of the Monte-Carlo method for phase space integrals is RAMBO [73], which maps  $4m$  random variables within the interval  $[0, 1]$  on  $m$  final state momenta, which are uniformly distributed and fulfill all constraints necessary for the four-momentum conservation, cf. Eq. (3.2). Unfortunately matrix elements typically have a very significant and non-uniform structure in phase space - values may vary over many orders of magnitude - causing a uniform momentum distribution to be very inefficient for the integration.

In the following optimization methods will be discussed and implementations presented to provide an improved integration performance.

### 3.1 Basic concepts for optimization

The goal of any optimization procedure is the reduction of the variance, which directly enters in the error estimate of the Monte-Carlo integration method. General methods that can be used to achieve this are *Stratified Sampling* and *Importance Sampling*:

- The method called Stratified Sampling simply subdivides the integration volume into smaller pieces, which are sampled independently. The sizes of the subdivisions and/or the number of points calculated in each subdivision can be chosen such that new error estimate is reduced compared to the undivided estimate. A minimum is obtained if the variances in all subdivisions are identical.
- The general idea behind Importance Sampling is to improve the numerical behaviour of an integrand by a change of integration variables,

$$\int f(x)dx = \int \frac{f(x(y))}{g(x(y))}dy , \quad \text{where} \quad \frac{1}{g} = \frac{dx(y)}{dy} . \quad (3.6)$$

The new variable  $y$  should be chosen in a way such that  $\frac{f}{g}$  is a sufficiently smooth function, leading to a variance for the integration,

$$S = \left\langle \frac{f^2}{g^2} \right\rangle - \left\langle \frac{f}{g} \right\rangle^2 . \quad (3.7)$$

Typically, the weight  $g$  is chosen as a simplification/approximation of  $f$ , such that the integral  $y = \int g dx$  can be analytically solved. This step is necessary to determine the map from a desired weight distribution.

For phase space integrals, a map  $X$ , relating vectors of uniformly distributed random numbers  $\{a_i\}$  inside the interval  $[0, 1]$  to the four-momenta of the external particles of a physical process  $\{p_j\}$ ,

$$\{p_j\} = X(\{a_i\}), \quad (3.8)$$

is in the center of the sampling process. The weight function  $g$ , equal to the inverse of the Jacobian, is then determined by

$$\frac{1}{g} = \frac{d\Phi_n(X(\{a_i\}))}{d\{a_i\}}. \quad (3.9)$$

Both strategies discussed above require a good knowledge of the integrand. To construct a general integrator for squared matrix elements, which should - similar to the matrix element generation itself - require no additional user input, there are basically two ways to incorporate this knowledge:

1. Use information about the integrand that can be obtained automatically during the matrix element generation.
2. Determine properties of the integrand numerically: use methods that adapt to the integrand during the integration procedure.

## 3.2 Integrator techniques

### 3.2.1 The self-adaptive integrator VEGAS

A simple method that automatically adapts to the integrand, not requiring any *a priori* knowledge of the integrand is the VEGAS integrator [74]. It relies mainly on stratified sampling and provides a map  $\xi$  relating uniformly distributed random variables  $\{a_i\}$  to variables distributed according to weights  $v_i$ .

In one dimension the algorithm is given as follows (mapping a random variable  $a$  inside  $[0, 1]$  to a new variable  $\xi(a)$  in the same interval):

1. The phase space is divided into  $N$  intervals,  $x_0 = 0, x_1, x_2, \dots, x_N = 1$ ;  $\Delta x_j = x_{j+1} - x_j$ .
2. The interval  $j$  is selected by the integer value of  $x * N$ , the new variable is given by

$$\xi(a) = x_j + (a - j/N)\Delta x_j. \quad (3.10)$$

3. The corresponding weight is  $\frac{1}{N\Delta x_j}$ .

The integration starts with equally spaced intervals. The adaptation follows the strategy for stratified sampling. It is performed changing sizes of intervals, but keeping their total number fixed. The procedure is as follows:

1. The integrand is evaluated  $M \gg N$  times, variances are stored for each interval separately.
2. Interval borders are changed to concentrate to regions of large variances.
3. Step 1. and 2. are repeated until variances in each interval are of the same size.

The  $n$ -dimensional version of the algorithm simply applies the 1-dimensional version to each dimension independently. On the one hand this guarantees a fast adaptation with almost any number of dimensions. On the other hand only factorizable weight functions can be produced, i.e.  $v = v_1(a_1) \cdot v_2(a_2) \cdots v_n(a_n)$ . For this reason VEGAS cannot be applied directly to phase space integrals.

There are developments of self-adaptive multi-purpose integrators that tried to overcome the restriction to factorizable functions such as FOAM [75] and Parni [76], however in practice it turns out that they adapt reasonable well and fast only for up to 3-4 dimensional integrands.

### 3.2.2 The Multi-Channel method

The structure of squared matrix elements is, apart from some trivial examples, too complicated to directly find a single phase space map and weight in the sense of Eqs. (3.8) and (3.9). It may contain a large number of peaks, which are in general non-factorizable in any single set of variables spanning the phase space.

A generic feature of our integrand is, however, that it can be composed out of amplitudes  $A_i$ :

$$|M|^2 = \left| \sum_{i=1}^n A_i \right|^2 = \sum_{i=1}^n |A_i|^2 + \sum_{i \neq j} A_i A_j^* . \quad (3.11)$$

The amplitudes usually represent a much simple structure. For the construction of phase space integrators typically only the direct squares of amplitudes are considered, for each of which a phase space map is constructed.

The multi-channel method [77] is a very efficient way to combine several such maps as follows:

$$\mathbf{X}(\{a_i\}, \tilde{\alpha}) = X_k(\{a_i\}), \text{ for } \sum_{l=1}^{k-1} \alpha_l < \tilde{\alpha} < \sum_{l=1}^k \alpha_l . \quad (3.12)$$

An additional random number  $\tilde{\alpha}$  is required. The coefficients  $\alpha_k$  are arbitrary within  $\alpha_k > 0$  and  $\sum_k \alpha_k = 1$ . The corresponding phase space weight for the full multi-channel is given by

$$G = \sum_k \alpha_k g_k , \quad (3.13)$$

i.e. the new map represents a distribution given by the sum of single channel weights.

The coefficients  $\alpha_k$  can be adjusted to minimize the variance of the phase space integral, now given by

$$S = W(\alpha) - \left\langle \frac{f}{G} \right\rangle , \quad (3.14)$$

where

$$W(\alpha) = \left\langle \frac{f(\mathbf{X}(\{a_i\}, \tilde{\alpha}))^2}{G^2} \right\rangle = \sum_k \alpha_k \left\langle \frac{f(X_k(\{a_i\}))^2}{G^2} \right\rangle . \quad (3.15)$$

The best coefficients  $\alpha_k$  are adapted during the integration by increasing the relative weight for channels with a large sub-variance estimate  $W_k(\alpha_k) = \frac{\partial}{\partial \alpha_k} W(\alpha)$ , until all  $W_k(\alpha_k)$  are of the same size.

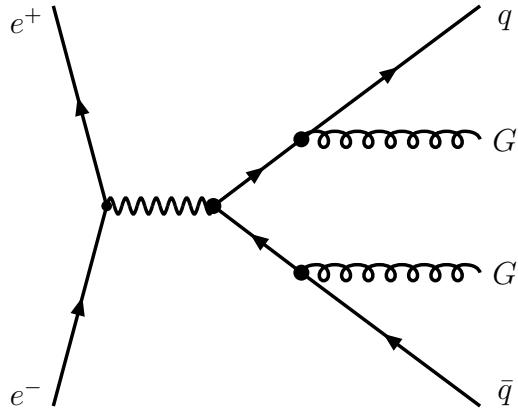
### 3.2.3 Improved Multi-Channeling

Single channels that are constructed to produce weight distributions similar to squared amplitudes will, in practice, only provide a rough approximation to it. The reasons are not only that the weight function must be kept simple enough to be able to analytically integrate it, but also kinematic restrictions due to momentum conservation and possible phase space cuts cannot always be fully considered *a priori*.

These defects can be reduced largely if the VEGAS integrator is applied on top of each channel. Although the full matrix element is not factorizable in its phase space variables, this is typically the case for the structures represented by single channels. In other words, VEGAS is used to adapt selected channels to structures that go beyond their approximations. The idea for this method has been presented in [78].

For each channel VEGAS is used to generate a map  $\xi$  from uniformly distributed random numbers to a non-uniform distribution, still inside the interval  $[0, 1]$ , and a corresponding weight  $v_k$ . To combine this with the multi-channel method, the map  $X(\{a_i\})$  for single channels must meet the requirement to be invertible. The full map reads

$$\mathbf{X}(\{a_i\}, \tilde{\alpha}) = X_k(\xi_k(\{a_i\})) , \quad \text{for } \sum_{l=1}^{k-1} \alpha_l < \tilde{\alpha} < \sum_{l=1}^k \alpha_l . \quad (3.16)$$



**Figure 3.1:** One amplitude for the process  $e^-e^+ \rightarrow q\bar{q}GG$ .

For a momentum configuration  $\{p_j\}$  the weight is therefore given by

$$G(\{p_j\}) = \sum_k \alpha_k g_k(\{p_j\}) v_k(X_k^{-1}(\{p_j\})) . \quad (3.17)$$

The remaining task is to provide the channels which enter into the procedure described above. This will be the subject of the following two sections.

### 3.3 Automatic generation of phase space maps

During the matrix element generation with AMEGIC++ also complete information of Feynman amplitudes is generated. This information can be used for the phase space integration: for each amplitude an appropriate map is generated, which decomposes the phase space according to the propagator structure.

To illustrate the principle, the example amplitude in Fig. (3.1) for the process  $e^-(p_1)e^+(p_2) \rightarrow q(p_3)G(p_4)G(p_5)\bar{q}(p_6)$  is considered. For this configuration the phase space can be factorized as follows:

$$\begin{aligned} d\Phi \propto & d\phi_{34,56} d\cos\theta_{34,56} \\ & \times ds_{34} d\phi_{3,4} d\cos\theta_{3,4} \\ & \times ds_{56} d\phi_{5,6} d\cos\theta_{5,6} , \end{aligned} \quad (3.18)$$

where  $s_{ij} = (p_i + p_j)^2$ . For each decay  $p_{ij} \rightarrow p_i + p_j$  the variable  $\theta_{ij}$  is the angle of the decay products  $i$  and  $j$  in the CM frame of  $p_{ij}$  w.r.t. to some reference axis;  $\phi$  is the corresponding azimuthal angle. Each of the new variables  $s_{34}$ ,  $s_{56}$ ,  $\phi_{34,56}$ ,  $\cos\theta_{34,56}$ ,  $\phi_{3,4}$ ,  $\cos\theta_{3,4}$ ,  $\phi_{5,6}$  and  $\cos\theta_{5,6}$  can now be generated from random numbers  $a$ , according to a given weight distribution.

If the propagators in the example are massless, the Lorentz invariant  $s$  would be generated by

$$s = [as_{\max}^{1-\nu} + (1-a)s_{\min}^{1-\nu}]^{\frac{1}{1-\nu}}, \quad (3.19)$$

with the corresponding weight

$$g = \frac{1-\nu}{s_{\max}^{1-\nu} - s_{\min}^{1-\nu}} \frac{1}{s^\nu}. \quad (3.20)$$

The constants  $s_{\max}$  and  $s_{\min}$  are upper and lower boundaries of the invariant mass, given by the available energy, particle masses and potential phase space cuts. Thus, the weight function is given by a constant factor to account for the phase space volume and a factor  $\frac{1}{s^\nu}$ , to compensate for the peak structure of the propagator;  $\nu$  is an effective exponent for the propagator, subject to choice.

The angular variables for the decay  $p_{ij} \rightarrow p_i + p_j$  are generated by

$$\begin{aligned} \phi &= 2\pi a_1, \\ \cos \theta &= 2a_2 - 1, \end{aligned} \quad (3.21)$$

with the weight

$$w = \frac{2}{\pi} \frac{1}{\lambda(s_{ij}, s_i, s_j)}, \quad (3.22)$$

where

$$\lambda(x, y, z) = \frac{\sqrt{(x-y-z)^2 - 4yz}}{x}. \quad (3.23)$$

The final state four momenta can be easily determined from the phase space variables. The full weight for the phase space map is given by product of the weights for all variables.

Two more building blocks are used for the generation of phase space maps in general, (but did not appeared in the example): For a massive propagator with mass  $M$  and a width  $\Gamma$ , the invariant mass is generated according to

$$s = M^2 + M\Gamma \times \tan(a(y(s_{\max}) - y(s_{\min})) + y(s_{\min})), \quad (3.24)$$

which leads to a weight

$$w = \frac{M\Gamma}{y(s_{\max}) - y(s_{\min})} \frac{1}{(s - M^2)^2 + M\Gamma}, \quad (3.25)$$

where

$$y(x) = \tan^{-1} \left[ \frac{x - M^2}{M\Gamma} \right]. \quad (3.26)$$

The fourth building block accounts for a t-channel configuration. In particular, for the process  $p_1 + p_2 \rightarrow p_3 + p_4$  momenta  $p_3$  and  $p_4$  are constructed with a weight compensating the t-channel propagator with the invariant mass  $t = (p_1 - p_3)^2$ . The kinematic variables used in this case are the angles  $\theta$  and  $\phi$ , similar as for decays, where  $\theta$  is the angle of  $p_3$  w.r.t.  $p_1$  in the CM frame of  $p_1 + p_2$  and  $\phi$  is again the azimuthal angle,

$$\begin{aligned}\phi &= 2\pi a_1, \\ \cos \theta &= x - [a_2(x-1)^{1-\nu} - (1-a_2(x+1))^{1-\nu}]^{\frac{1}{1-\nu}},\end{aligned}\quad (3.27)$$

where

$$x = \left[ \frac{2}{s} (M^2 - s_1 - s_3) + \left(1 + \frac{s_3}{s} - \frac{s_4}{s}\right) \left(1 + \frac{s_1}{s} - \frac{s_2}{s}\right) \right] \frac{1}{\lambda(s, s_1, s_2) \lambda(s, s_3, s_4)}, \quad (3.28)$$

with the corresponding weight

$$w = \frac{4}{\pi} \frac{1}{\lambda(s, s_3, s_4)} \frac{1}{(x+1)^{(1-\nu)} - (x-1)^{(1-\nu)}} \frac{1}{(a - \cos \theta)^\nu}. \quad (3.29)$$

The phase space maps created using this building blocks lead to weight functions roughly approximating the corresponding amplitudes. Many properties are neglected, such as spin correlations in decays and numerators of propagators (the exponents  $\nu$  will be fixed and not adjusted process wise). However, since the channels are put into the multichannel environment described in section 3.2.3, it is only important to provide suitable phase space factorizations. The VEGAS refinement can then easily compensate for defects in the exact weight distribution.

### 3.4 A general-purpose integrator for QCD-processes

As it has been shown in the previous chapter, for purely strong interacting processes naive evaluation of Feynman diagrams is not the most efficient method to calculate matrix elements. Similarly also the phase space integration would suffer from the extreme proliferation of Feynman amplitudes, if used for the construction of phase space maps.

It is well known [79] that the singular behaviour of such processes is roughly described by the so-called *antenna pole structure*. For a  $n$ -particle process this is the sum over all permutations of the antenna function

$$AP_n(p_1, p_2, \dots, p_n) = [(p_1 \cdot p_2)(p_2 \cdot p_3) \dots (p_{n-1} \cdot p_n)(p_n \cdot p_1)]^{-1}. \quad (3.30)$$

Moreover, a single antenna function  $AP_n$  coincides with divergency structure given by the partial amplitude of the same permutation<sup>2</sup>.

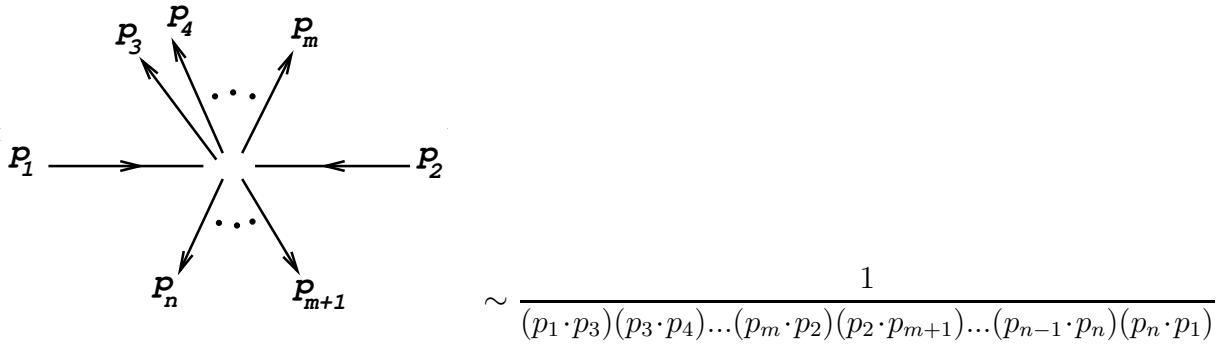
Two algorithms have been presented in the literature to construct phase space maps, leading to momentum distributions following a given antenna function. They are called SARGE [80] and HAAG [81]. The latter turns out to be more efficient, hence this one is used as the basis for a new implementation.

The general idea for the channels proposed in [81] is based on the decomposition of a  $n$ -body phase space

$$\begin{aligned}
d\Phi_n(P; p_1, \dots, p_n) &:= \left( \prod_{i=1}^n d^4 p_i \delta(p_i^2 - m_i^2) \right) \delta^4 \left( \sum_{i=1}^n p_i - P \right) \\
&= ds_{n-1} d\Phi_2(Q_n = P; m_n^2, s_{n-1}; p_n, Q_{n-1}) \\
&\times ds_{n-2} d\Phi_2(Q_{n-1}; m_{n-1}^2, s_{n-2}; p_{n-1}, Q_{n-2}) \\
&\quad \vdots \\
&\times ds_2 d\Phi_2(Q_3; m_3^2, s_2; p_3, Q_2) \\
&\times d\Phi_2(Q_2; m_2^2, m_1^2; p_2, p_1) , \tag{3.32}
\end{aligned}$$

where  $d\Phi_2(Q; s_2, s_1; p_2, p_1)$  is the phase space for the splitting of the virtual momentum  $Q$  into momenta  $p_1$  and  $p_2$  with  $p_1^2 = s_1$  and  $p_2^2 = s_2$ .

In Ref. [81] two algorithms are distinguished, which are referred to as *closed* and *open* antenna and that differ in the further decomposition of  $d\Phi_2$ . Only the closed antenna contains all factors in Eqs. (3.30), while in the open antenna one factor  $(p_i \cdot p_{i+1})$  is missing. Although the closed antenna seems to be more symmetric, in practice it turns out that the open antenna version is more efficient. This is mainly due to the simpler structure and that less additional weight factors appear within the algorithm (which are nonsingular in any of the products  $(p_i \cdot p_j)$ ). In the following only open antennas are discussed. The algorithm will be presented for the case of massless external particles, however, it can be easily generalized to the massive case.



**Figure 3.2:** Antenna configuration.

### 3.4.1 Antenna generation

In the following, a classification of antenna types by the position of the incoming momenta ( $p_0$  and  $p_1$ ) within the antenna is used, see fig 3.2. The type is given by  $\text{Min}(m-2, n-m-2)$ .

The basic building block for the antenna generation is the split of a massive momentum according to the phase space element  $ds d\Phi_2(Q; m^2 = 0, s; p, P; q)$ , where the last argument  $q$  defines an axis for the momentum generation. It is further decomposed by

$$d\Phi_2(Q; m^2 = 0, s; p, P; q) = da d\phi, \quad (3.33)$$

where

$$a = \frac{q \cdot p}{q \cdot P} \quad (3.34)$$

and  $\phi$  is an azimuthal angle around  $q$ .

The phase space for a single split, now defined through the variables  $s$ ,  $a$ ,  $\phi$ , is constructed as follows<sup>3</sup>:

1. Dice  $s$  according to the distribution  $1/s$  in  $[s_{\min}, s_{\max}]$ .
2. Dice  $a$  according to the distribution  $1/a$  in  $[a_{\min}, a_{\max}]$ .
3. Dice  $\phi$  according to a flat distribution in  $[0, 2\pi]$ .

<sup>2</sup>This is obvious when, e.g., inspecting MHV-amplitudes for the  $n$ -gluon process, Eq. (2.24):

$$|A_n (1^+, \dots, i^-, \dots, j^-, \dots, n^+)|^2 = \frac{(p_i \cdot p_j)^4}{(p_1 \cdot p_2)(p_2 \cdot p_3)\dots(p_{n-1} \cdot p_n)(p_n \cdot p_1)}, \quad (3.31)$$

where the property of the spinor products  $\langle ij \rangle \langle ij \rangle^* = p_i \cdot p_j$  was used. The recursion relation for non-MHV configurations leads to slightly different structures in which, however, still most of the factors in the denominator agree with the antenna function.

<sup>3</sup>frame dependent quantities are defined in the CM frame of  $Q$  with the  $z$ -Axis along  $q$

4. Momenta are given by

$$\begin{aligned} p = (e, p^1, p^2, p^3) &= \left( \frac{Q^2 - s}{2\sqrt{Q^2}}, \vec{p} \right), \\ P = (E, P^1, P^2, P^3) &= \left( \frac{Q^2 + s}{2\sqrt{Q^2}}, -\vec{p} \right), \\ \vec{p} &= \left( h \cos \phi, h \sin \phi, \frac{Q^2(1 - 2a) - s}{2\sqrt{Q^2}} \right), \quad h = \sqrt{Q^2 a(1 - a) - as}. \end{aligned} \quad (3.35)$$

5. The weight is given by

$$\frac{g(s_{\min}, s_{\max})}{s} \frac{g(a_{\min}, a_{\max})}{a} \frac{1}{2\pi}, \quad g(x_{\min}, x_{\max}) = \log \frac{x_{\max}}{x_{\min}}. \quad (3.36)$$

## Type 0

The phase space for this configuration can be obtained by a direct multiple application of the basic building block:

$$\begin{aligned} d\Phi_n(p_1, p_2; p_3, \dots, p_n) &= ds_3 d\Phi_2(Q_2 = p_1 + p_2; 0, s_3; p_3, Q_3; p_2) \\ &\times ds_4 d\Phi_2(Q_3; 0, s_4; p_4, Q_4; p_3) \\ &\vdots \\ &\times ds_{n-2} d\Phi_2(Q_{n-3}; 0, s_{n-2}; p_{n-2}, Q_{n-2}; p_{n-3}) \\ &\times d\Phi_2(Q_{n-2}; 0, 0; p_n, p_{n-1}; p_{n-2}) . \end{aligned} \quad (3.37)$$

The corresponding total weight is given by

$$w \sim \frac{\prod_{j=2}^{n-2} p_j \left( \sum_{i=j+1}^n p_i \right)}{\prod_{j=4}^{n-2} \left( \sum_{i=j}^n p_i \right)^2} \frac{1}{(p_2 \cdot p_3)(p_3 \cdot p_4) \cdots (p_{n-1} \cdot p_n)}, \quad (3.38)$$

where the contributions from boundary dependent functions  $g$  have been omitted.

## Type 1

For this configuration the following phase space decomposition is considered:

$$\begin{aligned} d\Phi_n(p_1, p_2; p_3, \dots, p_n) &= ds_3 d\Phi_2(Q_2 = p_1 + p_2; 0, s_3; p_3, Q_3; p_1) \\ &\times ds_4 d\Phi_2(Q_3; 0, s_4; p_4, Q_4; p_2) \\ &\times ds_5 d\Phi_2(Q_4; 0, s_5; p_5, Q_5; p_4) \\ &\vdots \\ &\times ds_{n-2} d\Phi_2(Q_{n-3}; 0, s_{n-2}; p_{n-2}, Q_{n-2}; p_{n-3}) \\ &\times d\Phi_2(Q_{n-2}; 0, 0; p_n, p_{n-1}; p_{n-2}) . \end{aligned} \quad (3.39)$$

In the first momentum split ( $d\Phi_2(Q_2 = p_1 + p_2; 0, s_3; p_3, Q_3; p_1)$ ) the variable  $a$  is diced now according the distribution  $\frac{1}{a(1-a)}$ . All following splits are generated according to the basic building block. The corresponding total weight is given by

$$w \sim p_1 \cdot (p_1 + p_2 - p_3) p_2 \cdot (p_1 + p_2 - p_3) \frac{\prod_{j=4}^{n-2} p_j \left( \sum_{i=j+1}^n p_i \right)}{\prod_{j=4}^{n-2} \left( \sum_{i=j}^n p_i \right)^2} \frac{1}{(p_1 \cdot p_3)(p_2 \cdot p_4) \cdots (p_{n-1} \cdot p_n)}. \quad (3.40)$$

### Type $k$ ( $\geq 2$ )

In this case the following decomposition is considered:

$$\begin{aligned} d\Phi_n(p_1, p_2; p_3, \dots, p_n) &= ds_3 ds_k d\Phi_2(Q_2 = p_1 + p_2; s_3, s_{k+1}; Q_3, Q_{k+1}; p_1) \\ &\times ds_4 d\Phi_2(Q_3; 0, s_4; p_4, Q_4; p_1) \\ &\times ds_5 d\Phi_2(Q_4; 0, s_5; p_5, Q_5; p_4) \\ &\vdots \\ &\times d\Phi_2(Q_{k-1}; 0, 0; p_k, p_{k+1}; p_{k-1}) \\ &\times ds_{k+2} d\Phi_2(Q_{k+1}; 0, s_{k+2}; p_{k+2}, Q_{k+2}; p_2) \\ &\times ds_{k+3} d\Phi_2(Q_{k+2}; 0, s_{k+3}; p_{k+3}, Q_{k+3}; p_{k+2}) \\ &\vdots \\ &\times d\Phi_2(Q_{n-2}; 0, 0; p_n, p_{n-1}; p_{n-2}). \end{aligned} \quad (3.41)$$

All splittings are generated according to the basic building block. The corresponding total weight is given by

$$\begin{aligned} w \sim p_2 \cdot (p_{k+2} + \dots + p_n) &\frac{\prod_{j=3}^{k-1} p_j \left( \sum_{i=j+1}^{k+1} p_i \right)}{\prod_{j=3}^{k-1} \left( \sum_{i=j}^{k+1} p_i \right)^2} \frac{\prod_{j=k+2}^{n-2} p_j \left( \sum_{i=j+1}^n p_i \right)}{\prod_{j=k+2}^{n-2} \left( \sum_{i=j}^n p_i \right)^2} \\ &\times \frac{1}{(p_1 \cdot p_3)(p_3 \cdot p_4) \cdots (p_k \cdot p_{k+1})(p_2 \cdot p_{k+2}) \cdots (p_{n-1} \cdot p_n)}. \end{aligned} \quad (3.42)$$

### 3.4.2 Integrator setup

For a  $n$ -particle process.  $(n-1)!$  different channels can be constructed. They are suitable to be directly combined in the improved multichannel method, presented in section 3.2.3. However the optimization procedure can be simplified significantly using a symmetry between the channels:

Two channels of the same type are identical up to a permutation of the final state momenta<sup>4</sup>. Thus the optimization parameters, which include individual VEGAS-maps and the relative weights  $\alpha$ , can be reused by all channels of a given type and optimized collectively. Of course, for the case of VEGAS-maps, the random numbers have to be permuted the same way as the final state momenta.

## 3.5 Integration with colour sampling

In this section, a special integrator is presented to be used for the integration of multi-gluon scattering processes together with a sampling over physical colour states for the external gluons.

As it has already been pointed out in section 2.3, the full summation over all colour configurations becomes unfeasible beyond a certain number of external partons. Instead, the better strategy is to use Monte-Carlo techniques for the summation over external colour assignments in a given SU(3) decomposition. Such non-interfering subsets of a QCD process typically have a strongly reduced number of contributing partial amplitudes compared to the full process. This issue has been studied in [64] for a number of different decompositions, i.e. for fundamental-representation decomposition, for adjoint-representation decomposition and for the colour-flow decomposition, which has been presented therein. The conclusion is, that the colour-flow decomposition, although not minimal concerning the number of required partial amplitudes, is the best suited method for a sampling over colour assignments if the number of external partons is large for the following reasons:

1. It provides the slowest growth in the average number of partial amplitudes per non-vanishing colour assignment.
2. It is relatively simple to determine all contributing partial amplitudes for a given colour assignment.
3. All colour factors are trivial to calculate, since they are given by a product of Kronecker-deltas, cf. Eq. (2.21).

Generally, the peaking behaviour of the colour-assigned cross section is rather complex within the phase space and strongly different for different colour assignments. The idea is now to construct integrators specific for a given colour assignment, based on the knowledge

---

<sup>4</sup>Of course, implying such a symmetry on the integrator is only meaningful if the integrand itself yields this symmetry. This is clearly given for  $n$ -gluon processes, but also when summing over all parton level processes contributing to hadronic cross sections such as  $pp \rightarrow n$  jets. The performance might suffer if just integrating single unsymmetric subprocesses alone.

about contributing partial amplitudes. The algorithm described below is constructed for gluon scattering processes. A generalization to any parton scattering processes is possible and will be discussed elsewhere [82].

### 3.5.1 Colour sampling

In the colour flow decomposition each external gluon is labeled by a colour index  $i$  and a anti-colour index  $j$ . The colour state for a  $n$ -gluon scattering is thus given by specifying all indices  $i_1, \dots, i_n$  and  $j_1, \dots, j_n$  out of three values  $(R, G, B)$  for each index.

A colour flow (and thus an ordering) is specified by a permutation  $\sigma \in S_{n-1}$  of external particles,

$$(1, \sigma_2, \sigma_3, \dots, \sigma_n) . \quad (3.43)$$

The colour flow is contributing to an assignment, if

$$\delta_{j_{\sigma_2}}^{i_1} \delta_{j_{\sigma_3}}^{i_{\sigma_2}} \dots \delta_{j_1}^{i_{\sigma_n}} = 1 . \quad (3.44)$$

It is relatively easy to construct an algorithm which determines all valid flows from a given assignment:

1. Starting with gluon 1, for the second gluon in the order the index  $j_{\sigma_2}$  must be equal to  $i_1$ .
2. One of the possible second gluons is selected, a third gluon is determined the same way.
3. If it is possible to build up a chain which includes all gluons and meets  $i_{\sigma_n} = j_1$ , a valid flow is found.
4. Selecting systematically the first, second, third, ... possible gluon in each branch all valid flows are determined.

The simplest way of choosing a colour assignment is accomplished by randomly selecting the  $2n$  colours for the  $i$ - and  $j$ -indices. Each colour is chosen with an equal probability, leading to a weight of  $1/3^{2n}$ . However, only a small fraction of those assignments will have at least one colour flow. A trivial (but not sufficient) condition for non-vanishing assignments is, that the number of  $i$ -indices carrying the colour  $R$  ( $G, B$ ) must be equal to the number of  $j$ -indices carrying this colour.

A more efficient way to determine assignments is as follows:

1. Only the  $n$   $i$ -indices are selected randomly.

2. One permutation  $\sigma$  of  $n$  particles is selected randomly (and uniformly distributed). The colours for the  $j$ -indices are given by

$$j_k = i_{\sigma_k}, \quad \text{for } k = 1, \dots, n \quad (3.45)$$

3. The weight for selecting a assignment is given by

$$w = \frac{r!g!b!}{3^n n!}, \quad (3.46)$$

where  $r, g, b$  are the number  $i$ -indices carrying the colours  $R, G, B$ , respectively, with  $r + g + b = n$ .

Clearly, assignments generated by this algorithm will always fulfil the trivial condition, mentioned above. Moreover, the weight is roughly proportional to the number of colour flows to the diced assignment and thus already corresponds to some extend to the expected cross section for the colour configuration.

### 3.5.2 Integrator setup

As basic building blocks the HAAG-channels that have been discussed in section 3.4.1 will be reused. They provide efficient integrators for squared partial amplitudes. Unlike partial amplitudes, however, the channels are not invariant under inversion of the order of arguments (which is due to the fact that the *open antenna* algorithm has been used). To ease the further discussion, the channels are symmetrized by combining each channel with its reversed argument version, i.e one of the two configurations is chosen with equal probability and the weight is given by the average of the two.

The integrator for a given colour assignment is constructed as follows: for each colour flow the HAAG-channel to the corresponding partial amplitude is added to a multi-channel integrator, cf. section 3.2.3. However, with a growing number of external particles one has to face the following problem:

Although the average number of contributing colour flows per colour assignment is relatively low in this decomposition, the maximal number grows factorially. Thus it is quickly getting impossible to store all data associated with the multi-channel, i.e. the contributing HAAG-channels and the internal  $\alpha$ -parameters. The situation is even worse if it is intended to sample over all colour assignments, which number is growing exponentially with the number of particles. The way out is not to store anything, but generate the integrator *on the flight*.

The fast algorithm to provide all colour flows from a colour assignment is essential for this step: for a single phase space point one has to loop three times though the list of all colour flows (which can't be stored as well):

1. To determine the normalization of the coefficients  $\alpha_k$  for each phase space channel within the multi-channel integrator, cf. Eqs. (3.12) and (3.13).
2. To select one channel with a probability given by the relative weight  $\alpha_k$  to generate a phase space point, and
3. to compute the multi-channel weight corresponding to the phase space point.

Strategies how to define suitable  $\alpha_k$  (beyond equal weights for all channels) are discussed below. For the HAAG-channels itself, only one per type (as defined in section 3.4.1) has to be stored. Together with a corresponding permutation of final state particles they can be reused anywhere in the *on the flight* algorithm.

### 3.5.3 The optimization procedure

The proposed integrator contains a number of parameters which can be adjusted or adapted to improve the variance during the calculation:

- VEGAS maps within the HAAG channels,
- the relative weights  $\alpha_k$  in the multi-channel generator,
- probabilities to select colour assignments beyond the algorithms given above.

The usage of self adaptive methods is somewhat limited due to the fact that the number of those parameters extremely increase with the number of particles involved in the process. Not only that it becomes impossible to calculate the matrix element for enough phase space points to adapt each parameter individually, at some point all those parameters cannot even be stored.

Thus the following strategy is applied:

1. Optimization of the VEGAS maps refining the HAAG-channels:

The number of structurally different HAAG-channels is very limited, only one channel per type. Their optimization is done before the actual integration starts. For this purpose only single squared partial amplitudes are calculated to optimize the corresponding channel<sup>5</sup>. This not only speeds up the calculation, it also provides a much cleaner environment for the adaptation of the VEGAS-maps. In this step it is

---

<sup>5</sup>During this step the full result can not be determined since potential interferences between partial amplitudes are ignored. However, it is sufficient for computing the leading  $1/N_C$  limit for  $n$  gluon processes, using the fact that in the colour flow decomposition (as well as in fundamental-representation decomposition) interferences are always subleading.

summed over all helicities. Cross sections  $\sigma_t$ , given by the integration of a squared partial amplitude of type  $t$  over the allowed phase space, are stored.

2. The actual integration run:

No further optimization is done. The channels are used as they came out of the optimization step, including the VEGAS-map and a  $\alpha_k$  parameter proportional to the cross section of the corresponding squared partial amplitude,  $\sigma_t$ .

Best performance is achieved if the colour assignment is also selected with a probability proportional to the sum of cross sections of contributing squared partial amplitudes (determined in step 1), instead of the weight given in Eq. (3.46). To do so, the total normalization for the new weight must be determined summing over all colour assignments. For  $n$ -gluon processes this is given by a following simple formula:

$$N = (n-2)! * 3^n * \sum_{i=0}^{n-2} \sigma_{\min(i, n-i-2)}, \quad (3.47)$$

where the  $\sigma_{\min(i, n-i-2)}$  is the cross section of a squared partial amplitude of the type “ $\min(i, n-i-2)$ ”. The reweighting can be done by a simple *hit-or-miss* method.

For the integration run it is a subject of choice whether to sum or to sample over helicities. All practical tests for up to the 11-gluon process favoured the summation. Beyond that, however, it seems to become to costly to compute summed matrix elements, thus a sampling should be considered.



# 4 Efficiency comparisson and results

In the previous chapters several methods and implementations for matrix element and phase space generation have been presented. In the following they will be compared to justify the optimal method. The focus hereby is on the calculation of cross sections for pure QCD processes.

## 4.1 Matrix elements

Firstly, the time necessary to calculate a matrix element at a given phase space point is compared. For the matrix elements generated with AMEGIC++ using the two available algorithms, i.e. direct evaluation of Feynman diagrams and the MHV-amplitude based recursion relations (CSW-rules), this can be done directly, since in both cases exactly the same thing is calculated: matrix elements summed over all colour- and helicity-configurations. Of course, numerical results obtained with either method are identical (up to numeric precision). Tab. 4.1 lists evaluation times for a single ME in both methods for processes, available using the MHV technique. Clearly, the more symmetric a process is (w.r.t. QCD interactions), the greater the gain employing the new method. The largest difference is found for pure gluon scattering, where with the new technique already for the  $gg \rightarrow 4g$  process a factor of 166 is gained. For a scattering of more than six gluons the evaluation of Feynman diagrams becomes unjustifiable complicated and is thus not listed. The reason for this pattern is that the non-abelian interactions lead to an extreme inflation of diagrams to be evaluated, while MHV amplitudes inherently incorporate SU(3)-symmetries allowing leading to a significant and nontrivial simplification. On the other hand, for processes involving electroweakly interacting particles there is no gain with the new method. Thus, the application area for the CSW technique is pure QCD.

Tab. 4.2 compares the evaluation time of partial amplitudes for gluon scattering, generated employing the Berends-Giele (BG) recursion and the CSW recursion relation, both summed over helicity configurations<sup>1</sup>. In particular for two and three gluon production processes, all

---

<sup>1</sup> Since the implementation of the BG recursion relation in COMIX is based on a colour-flow decomposition

Process	Time per ME Conventional	Time per ME CSW rules	Conventional / CSW rules
$gg \rightarrow 2g$	231 $\mu$ s	2.62 $\mu$ s	8.8
$gg \rightarrow 3g$	1.16 ms	0.0368 ms	31
$gg \rightarrow 4g$	314 ms	1.89 ms	166
$gg \rightarrow 5g$	-	0.058 s	
$gg \rightarrow 6g$	-	7.80 s	
$gg \rightarrow 7g$	-	532 s	
$q\bar{q} \rightarrow 2g$	8.29 $\mu$ s	2.01 $\mu$ s	4.1
$q\bar{q} \rightarrow 3g$	155 $\mu$ s	24.7 $\mu$ s	6.3
$q\bar{q} \rightarrow 4g$	18.4 ms	1.25 ms	15
$q\bar{q} \rightarrow 5g$	4.64 s	0.0364 s	127
$q\bar{q} \rightarrow 6g$	-	3.79 s	
$q\bar{q} \rightarrow q\bar{q}$	4.52 $\mu$ s	4.00 $\mu$ s	1.1
$q\bar{q} \rightarrow q\bar{q} g$	51.3 $\mu$ s	19.8 $\mu$ s	2.6
$q\bar{q} \rightarrow q\bar{q} 2g$	3.17 ms	0.81 ms	3.9
$q\bar{q} \rightarrow q\bar{q} 3g$	654 ms	17.2 ms	38
$q\bar{q} \rightarrow q\bar{q} 4g$	-	1.51 s	
$q\bar{q} \rightarrow q'\bar{q}'$	2.55 $\mu$ s	1.87 $\mu$ s	1.4
$q\bar{q} \rightarrow q'\bar{q}' g$	18.8 $\mu$ s	9.38 $\mu$ s	2.0
$q\bar{q} \rightarrow q'\bar{q}' 2g$	615 $\mu$ s	407 $\mu$ s	1.5
$q\bar{q} \rightarrow q'\bar{q}' 3g$	118 ms	8.83 ms	13
$q\bar{q} \rightarrow q'\bar{q}' 4g$	-	757 ms	
$\bar{q}q' \rightarrow W^- (\rightarrow e^- \bar{\nu}_e)$	2.53 $\mu$ s	3.78 $\mu$ s	0.67
$\bar{q}q' \rightarrow W^- (\rightarrow e^- \bar{\nu}_e) g$	6.32 $\mu$ s	7.75 $\mu$ s	0.81
$\bar{q}q' \rightarrow W^- (\rightarrow e^- \bar{\nu}_e) 2g$	18.8 $\mu$ s	41.7 $\mu$ s	0.45
$\bar{q}q' \rightarrow W^- (\rightarrow e^- \bar{\nu}_e) 3g$	243 $\mu$ s	471 $\mu$ s	0.51
$\bar{q}q' \rightarrow W^- (\rightarrow e^- \bar{\nu}_e) 4g$	22.1 ms	34.1 ms	0.65

**Table 4.1:** Computation time for full matrix elements summed over colour and helicity.

Displayed times are averages for a single evaluation, employing the conventional formalism in AMEGIC++ and the Cachazo-Svrcek-Witten (CSW) recursion relation. The numbers were generated on a 3200+ AMD Athlon<sup>TM</sup> 64 CPU.

Process	Time per ME BG	Time per ME CSW	BG/CSW
$gg \rightarrow 2g$	$60.0 \mu s$	$4.91 \mu s$	12
$gg \rightarrow 3g$	$185 \mu s$	$9.85 \mu s$	18
$gg \rightarrow 4g$	$547 \mu s$	$132 \mu s$	4.1
$gg \rightarrow 5g$	$1.62 \text{ ms}$	$0.655 \text{ ms}$	2.5
$gg \rightarrow 6g$	$4.85 \text{ ms}$	$11.3 \text{ ms}$	0.43
$gg \rightarrow 7g$	$14.9 \text{ ms}$	$39.4 \text{ ms}$	0.38
$gg \rightarrow 8g$	$48.4 \text{ ms}$	-	
$gg \rightarrow 9g$	$166 \text{ ms}$	-	
$gg \rightarrow 10g$	$619 \text{ ms}$	-	

**Table 4.2:** Average computation time of partial amplitudes in multi-gluon scattering, summed over all helicity configurations. Displayed times are averages for a single evaluation, employing the Berends-Giele (BG) recursion and the CSW recursion. The numbers were generated on a 2.80 GHz Pentium® 4 CPU.

non-vanishing helicity configurations are given solely by MHV- ( $\overline{\text{MHV}}$ -) amplitudes. Thus the method based on them will be superior; for the BG recursion relation Feynman amplitudes have to be evaluated, leading to more complicated expressions. However, the CSW recursion relation is rather complicated and leads to relative high proliferation of contributing subamplitudes, such that the growth in complexity is much faster. The computation time given in Tab. 4.2 increases approximately as  $7^n$ , while the BG-method leads to a scaling approximately as  $3.3^n$ . Starting from  $gg \rightarrow 6g$  scattering the latter will be more efficient.

In Tab. 4.3 computation times for the BG recursion are listed for full colour-dressed amplitudes, when summing and when sampling over helicity configurations. Of course it is much faster to compute a matrix element for a single helicity state than for all states. What is more efficient for the calculation of cross sections, however, can only be decided together with the phase space integration method. For matrix elements summed over helicity a scaling as  $6^n$  is observed, while for the case of sampling the evaluation time scales as  $3.2^n$ .

## 4.2 Integration performance and results

The overall integration performance depends on a number of factors:

- the quality and complexity of phase space maps,

it would be unfair to compare complete colour summed matrix elements. This decomposition is well suited for a sampling over colour assignments, but would be rather inefficient to sum over them.

Process	Time per ME summed helicities	Time per ME sampled helicities	ratio
$gg \rightarrow 2g$	73.1 $\mu$ s	25.2 $\mu$ s	2.9
$gg \rightarrow 3g$	339 $\mu$ s	59.5 $\mu$ s	5.7
$gg \rightarrow 4g$	1.67 ms	0.149 ms	11
$gg \rightarrow 5g$	8.98 ms	0.427 ms	21
$gg \rightarrow 6g$	49.6 ms	1.39 ms	36
$gg \rightarrow 7g$	298 ms	4.32 ms	69
$gg \rightarrow 8g$	1.99 s	13.6 ms	146
$gg \rightarrow 9g$	13.1 s	43.7 ms	300
$gg \rightarrow 10g$	96 s	138 ms	695

**Table 4.3:** Computation time for multi-gluon scattering matrix elements sampled over colour configurations. Displayed times are averages for a single evaluation of the colour-dressed BG recursion relation, when summing and when sampling over helicity configurations, respectively. The numbers were generated on a 2.80 GHz Pentium® 4 CPU.

- the adaptation procedure: required number matrix element evaluations for adaptation and the quality of the final map, and
- the question whether it is summed or sampled over some degrees of freedom (colour and helicity).

Below the strategies for phase space integration, discussed in chapter 3, are compared.

Unless stated otherwise, all partonic cross sections are obtained for LHC with a proton centre-of-mass of 14 TeV. The parton distribution function CTEQ6M [83] has been employed, factorization and renormalization scales have been fixed to  $\hat{s}$ . As a phase space cut, simply each parton is required to satisfy the  $k_T$ -jet criterion [18] with  $Q_{\text{cut}} = 20$  GeV and the jet resolution parameter  $D = 1$ .

To start with, in Tab. 4.2 the performance for integrators, using the phase space channels, presented in sections 3.3 and 3.4.2, are compared for multi-gluon scattering and for hadronic cross sections. The matrix elements were generated using the CSW recursion and are summed over helicity and colour, which is the standard AMEGIC++ setup. The statistical errors are given after the evaluation of 310000 phase space points, which includes the full adaptation procedure. The efficiency is the ratio of the mean and the maximum weight for single event, i.e. it corresponds to the unweighting efficiency using a hit-or-miss method. Except for the rather simple phase space of  $2 \rightarrow 2$  scattering the HAAG-channels are clearly superior w.r.t. both criteria, the statistical error and the unweighting efficiency.

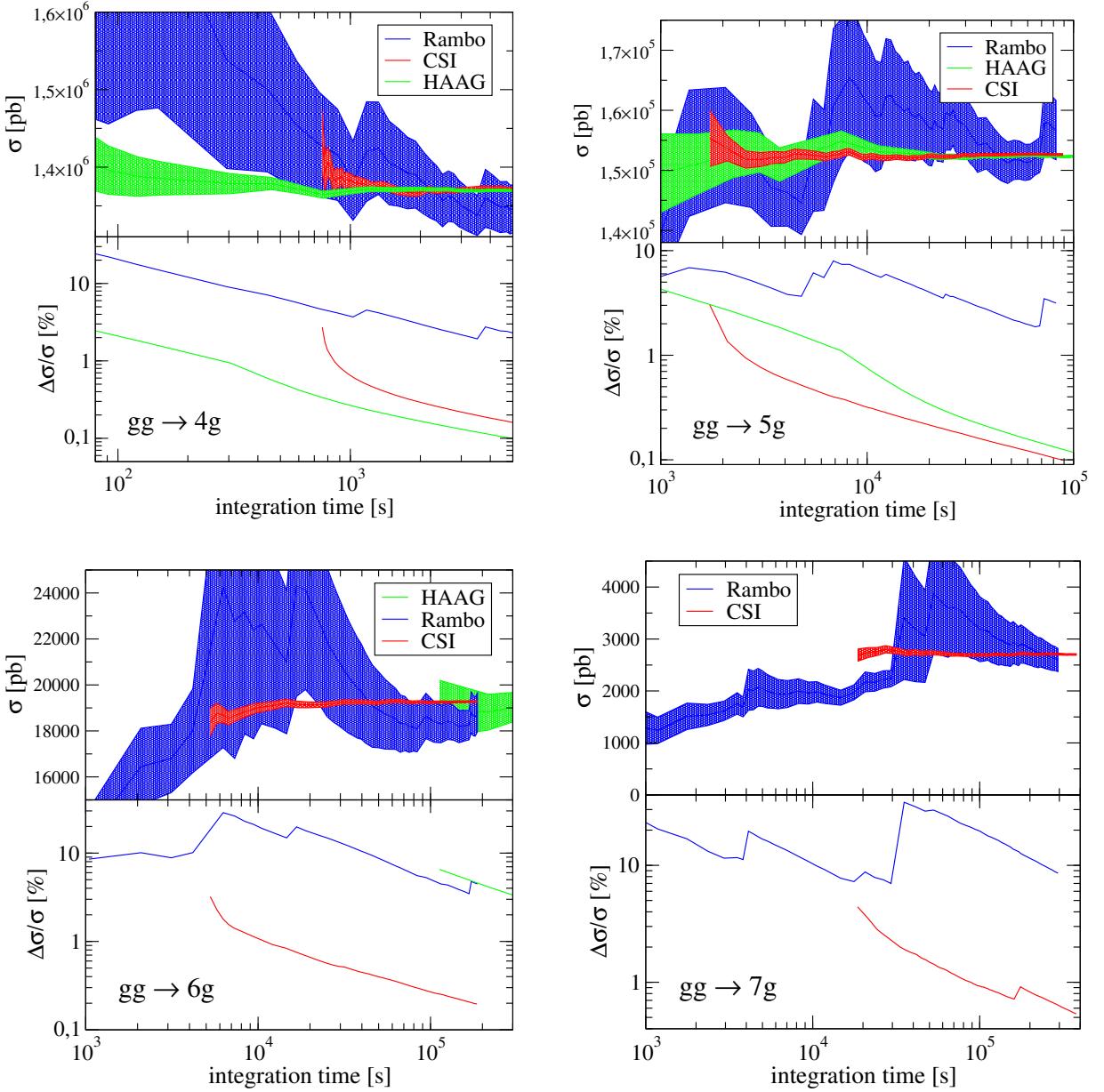
Process	HAAG		Amegic MC	
	stat. error	efficiency	stat. error	efficiency
$gg \rightarrow gg$	0.11 %	18 %	0.09 %	14 %
$jj \rightarrow jj$	0.11 %	9.4 %	0.10 %	9.3 %
$gg \rightarrow 3g$	0.18 %	7.0 %	0.28 %	0.50 %
$pp \rightarrow 3j$	0.20 %	2.2 %	0.29 %	0.43 %
$gg \rightarrow 4g$	0.23 %	1.5 %	0.56 %	0.12 %
$pp \rightarrow 4j$	0.25 %	0.49 %	0.56 %	0.06 %
$gg \rightarrow 5g$	0.29 %	0.78 %	-	-
$pp \rightarrow 5j$	0.28 %	0.29 %	-	-
$gg \rightarrow 6g$	0.38 %	0.28 %	-	-

**Table 4.4:** Integration performance for matrix elements summed over colour and helicity configurations. The integrator HAAG has been described in section 3.3 and Amegic MC denotes the implementation described in section 3.4.2. Statistical errors are given after the evaluation of 310000 phase space points. The efficiency states the fraction of events, passing an unweighting procedure. Results have been obtained for LHC energies and requiring all gluons/jets to satisfy the  $k_T$ -jet criterion with  $Q_{\text{cut}} = 20$  GeV and  $D = 1$ .

As a reference in Tab. 4.2 the results are listed for HAAG with the VEGAS improved multi-channel method (cf. section 3.2.3) switched off, and for the flat phase space generator RAMBO [73]. Clearly, VEGAS improves the efficiency significantly, the statistical error is reduced by roughly a factor of 2 and the unweighting efficiency even by an order of magnitude. RAMBO, on the other hand, cannot compete with the optimized integrators.  $2 \rightarrow 6$  scattering in pure QCD is about the upper limit, feasible for an approach that sums over colour. Beyond that not only the matrix element becomes too costly to calculate, also the number of phase space channels for the optimized sampling methods gets extremely large, such that the calculation of the multi-channel weight becomes very time consuming and the optimization procedure suffers.

Figs. 4.1 and 4.2 compare the overall integration performance between different integration methods for multi-gluon scattering. Three integrators are compared, the flat phase space generator RAMBO, the general-purpose QCD-integrator HAAG (cf. section 3.4.2) and the special colour sampling integrator presented in section 3.5 (CSI). Each integrator employs matrix element samples, for which it performs best:

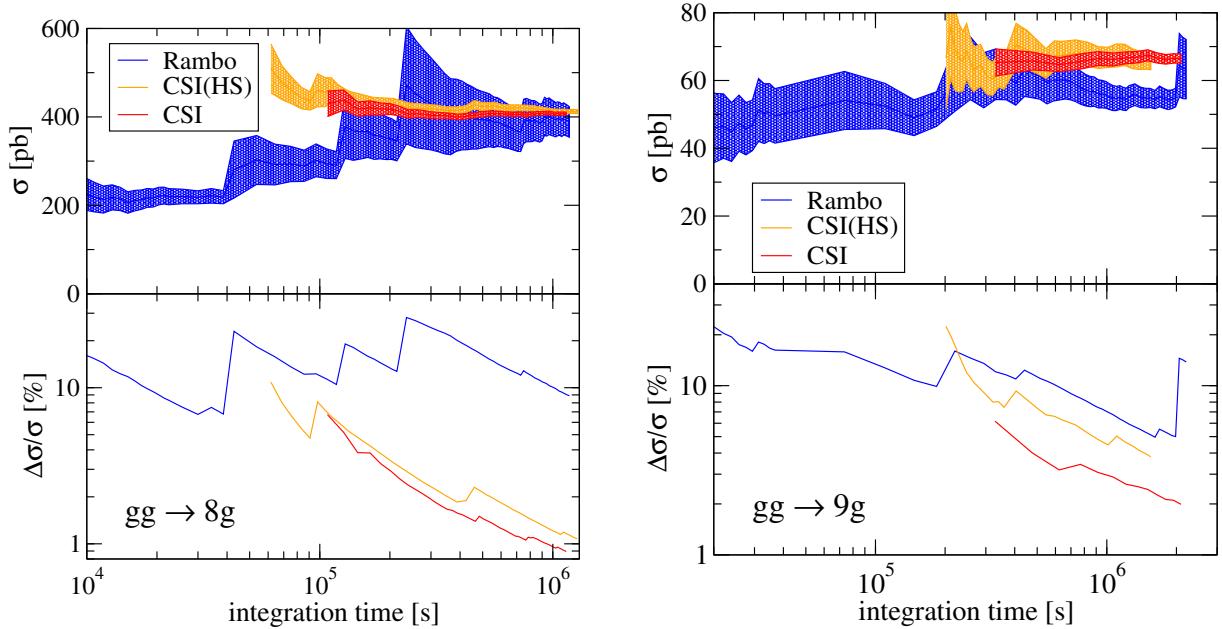
- RAMBO: ME sampled over colour and helicity,
- HAAG: ME summed over colour and helicity,



**Figure 4.1:** The overall integration performance for multi-gluon scattering. The upper panels display the Monte Carlo estimate of the cross section with the corresponding  $1\sigma$  statistical error band as a function of the total integration time. The lower panels show the relative statistical error. HAAG denotes the phase space integrator described in section 3.4.2, applied on colour- and helicity-summed ME, generated using the CSW recursion. CSI denotes the integrator discussed in section 3.5, applied on colour-sampled and helicity-summed ME, generated using the BG recursion. Results for RAMBO were generated using colour- and helicity-sampled ME form the BG recursion. Calculations have been performed on a 2.66 GHz Xeon<sup>TM</sup> CPU

Process	HAAG w/o VEGAS		RAMBO	
	stat. error	efficiency	stat. error	efficiency
$gg \rightarrow gg$	0.25 %	9.0 %	0.33 %	0.43 %
$jj \rightarrow jj$	0.25 %	6.6 %	0.38 %	0.41 %
$gg \rightarrow 3g$	0.36 %	1.1 %	0.92 %	0.09 %
$pp \rightarrow 3j$	0.37 %	0.55 %	1.8 %	2.8e-2 %
$gg \rightarrow 4g$	0.50 %	0.14 %	2.0 %	1.0e-2 %
$pp \rightarrow 4j$	0.52 %	0.09 %	2.3 %	7.1e-3 %
$gg \rightarrow 5g$	0.68 %	0.02 %	3.6 %	2.4e-3 %
$pp \rightarrow 5j$	0.72 %	0.01 %	4.0 %	1.6e-3 %

**Table 4.5:** Integration performance for matrix elements summed over colour and helicity configurations for the same setup as in Tab. 4.2. The left columns list the results obtained employing the HAAG-integrator without an additional VEGAS refinement. The results in the right column are generated using a flat momentum distribution, generated by RAMBO.



**Figure 4.2:** The overall integration performance for multi-gluon scattering, continued from Fig. 4.1. Additionally, for the CSI a sampling over helicity is considered, denoted by CSI(HS).

Process	cross section [pb]
$gg \rightarrow 2g$	$3.3521(31) \times 10^8$
$jj \rightarrow 2j$	$5.3122(54) \times 10^8$
$gg \rightarrow 3g$	$1.3118(24) \times 10^7$
$pp \rightarrow 3j$	$2.5041(50) \times 10^7$
$gg \rightarrow 4g$	$1.3699(14) \times 10^6$
$pp \rightarrow 4j$	$3.0885(78) \times 10^6$
$gg \rightarrow 5g$	$1.5250(15) \times 10^5$
$pp \rightarrow 5j$	$4.0287(67) \times 10^5$
$gg \rightarrow 6g$	$1.9282(38) \times 10^4$
$gg \rightarrow 7g$	$2703(14)$
$gg \rightarrow 8g$	$407.0(36)$
$gg \rightarrow 9g$	$66.5(13)$
$gg \rightarrow 10g$	$15.2(26)$

**Table 4.6:** Multi-gluon and hadronic cross sections at LHC. In parenthesis the statistical error is stated in units of the last digit of the cross section.

- CSI: ME sampled over colour and summed over helicity.

For the processes  $gg \rightarrow 8g$  and  $gg \rightarrow 9g$  additionally a sampling over helicity has been considered (labeled by CSI(HS)).

Where applicable, the phase space optimization procedure is included in the integration time, i.e. at the start all multichannel weights and VEGAS maps are uniform. For all displayed gluon multiplicities the optimized integrators lead to a statistical error smaller by roughly a factor of 10 compared to RAMBO. For small multiplicities it seems to be most efficient to sum over colour configurations. From  $gg \rightarrow 6g$  on, however, this approach cannot compete anymore.

With increasing multiplicity the CSI requires significantly more time until a first estimate can be given. This is due to the fact that the optimization takes place before the actual integration starts. Although with RAMBO very quickly estimates can be obtained, they might to be unreliable for short integration times: cross sections and errors tend to be underestimated, since small but strongly enhanced phase space regions can be easily missed by a flat momentum distribution. Beyond  $gg \rightarrow 9g$  processes the CSI method suffers from the increasing complexity of helicity summed ME, cf. Tab. 4.3. In this case the usage of helicity sampled ME can be considered. Although the overall performance is inferior compared to summed ME, it is still more efficient (and reliable w.r.t. error estimates) than RAMBO.

Process	$\sqrt{s}$ [GeV]	cross section [pb]		
		COMIX	Ref. [64]	Ref. [84]
$gg \rightarrow 8g$	1500	0.755(3)	0.70(4)	0.719(19)
$gg \rightarrow 9g$	2000	0.305(2)	0.30(2)	
$gg \rightarrow 10g$	2500	0.101(7)	0.097(6)	
$gg \rightarrow 11g$	3000	0.0188(20)		

**Table 4.7:** Cross sections multi-gluon scattering at a CM energy of  $\sqrt{s}$ , using the phase space cuts specified in Eq. (4.1), compared to literature results. In parenthesis the statistical error is stated in units of the last digit of the cross section.

Employing the best matrix element generation and integration method for each process, tree level cross section with a very high statistical precision can be obtained up to extremely high multiplicities. Results for the setup specified above are listed in Tab. 4.6.

To compare the new matrix element and phase space generation methods to know results, the setup used in Refs. [64, 84] is considered. There, gluons at a fixed CM energy are scattered employing the phase space cuts

$$p_{Ti} > 60 \text{ GeV} , \quad |\eta_i| < 2 , \quad \Delta R_{ij} > 0.7 , \quad (4.1)$$

on each final state gluon  $i$  (pair of gluons  $i, j$ ). The cuts have been chosen to widely avoid the divergent regions of the matrix element, such that even from a flat phase space generator reliable results can be obtained. The cross sections and statistical errors are listed in Tab. 4.7. The reference results are reproduced within statistical uncertainties. The new phase space generator allows to strongly reduce statistical errors for  $gg \rightarrow 8g$  and  $gg \rightarrow 9g$ . A full tree level cross section for  $gg \rightarrow 11g$  has not yet appeared in the literature.

## 4.3 Conclusions

In the past three chapters a number of extensions and new concepts for the calculation of tree level cross sections in SHERPA have been presented. This included the automated generation of matrix elements and the integration over the multi-particle phase space. In particular, the treatment of QCD-amplitudes has been improved to overcome the technical limitations given by conventional methods. Within this work the Monte-Carlo integration methods have been completely revised and significantly improved. For the case multi-gluon scattering this has been carried to extremes: the new methods allow the calculation of cross sections, with a precision and parton multiplicity that have not been accessible before.

The overall performance for the calculation of QCD cross sections using various techniques

has been compared, the conclusion is the following: for final states with up to 4-5 partons the usage the CSW-technique together with the integrator presented in section 3.4.2 perform best, beyond that the Berends-Giele recursion and the integration techniques which involve colour sampling (section 3.5) are superior. Other processes have not been analysed in full detail yet, but first test indicate that, again, for smaller final states (typically up to 5-6 particles for full tree level ME) the matrix element generation methods implemented in AMEGIC++ are favoured, whereas more complex final states are the domain of the Berends-Giele recursion, implemented in COMIX.

## Part II

# Automating NLO calculations



# 5 Automating the Dipole-Subtraction method

For most of the relevant observables in particle phenomenology, the leading term of the perturbative expansion can be related to tree-level diagrams. However, for many practical purposes, tree-level calculations are not sufficient. This is due to a number of reasons: first of all, many measurements aim at the extraction of fundamental parameters. However, in Quantum Field Theories, parameters are subject to corrections, which usually exhibit ultra-violet divergences. These divergences are dealt with through the renormalization procedure, which can be done in a scheme- and scale-dependent way only, see e.g. [4, 85]. Therefore, in order to extract parameters from the comparison of a (perturbative) calculation with experimental data, the calculation itself must contain the same kind of quantum corrections necessitating their renormalization. Second, it should be stressed that in tree-level calculations, there are some choices to be made, concerning the scale at which inputs such as the coupling constant, quark masses or parton distribution functions are taken. In principle, different scale choices are equivalent, and renormalization group theory guarantees that, when taking into account all orders, the effect of scale choices vanishes. At leading order (LO), however, their impact may still be significant, such that tree-level calculations merely give the order of magnitude for corresponding cross sections etc.; a prime example for this is the production of a Higgs boson in gluon fusion processes, where only the next-to-next-to leading order correction significantly reduces the scale dependence and produces a stable result [86, 87]. Thus, aiming at any more precise prediction, higher-order calculations are a crucial ingredient of phenomenological analyses.

But although indispensable, so far there is no fully automated tool available for QCD calculations at next-to leading order (NLO), i.e. at the one-loop level. This is because a true NLO calculation is certainly much more complex than a leading order (LO) one. First of all, some of the essential ingredients, namely the loop or virtual contributions are not under full control yet. In general, up to now calculations of these corrections to physical processes are limited to contributions containing five- and in some cases six-point functions, see for example [25, 26, 28, 29, 30]. But even to reach the level of known scalar master integrals is far

from being trivial; the tensor reduction necessary for this step [88] results in a proliferation of terms with non-trivial cancellations among them, which render the implementation in a computer code a major effort. A new approach to face some of those problems is reported in chapter 6. On the other hand, some of the loop corrections exhibit not only ultraviolet divergences to be renormalized, but also infrared divergences. They also need to be regularized, but then they must be canceled against similar infrared divergences stemming from the real contributions. This basically translates into canceling divergences in phase space volumes of different dimensionality. The cancellation in fact is one of the most important consequences of the Kinoshita-Lee-Nauenberg or mass factorization theorems [89, 90]. However, in order to practically achieve the cancellation, the real infrared divergences also need to be regularized. Essentially, there are two ways of doing this.

One method, also known as phase-space slicing [91, 92, 93, 94, 95, 96], bases on dividing the phase space of the additional real emission into an infrared-safe (hard) and a infrared-divergent (soft) region. The division is usually performed by subjecting pairs of particles to an invariant mass criterion. Then, the soft region is integrated analytically in  $d$  dimensions. Typically, in this step, the helicity-summed matrix element squared is approximated by its double-pole (or eikonal) limit. The result of the analytical integration will contain single or double poles of the form  $1/(d-4)$  or  $1/(d-4)^2$ , respectively. They typically are accompanied with logarithms of the invariant mass criterion. Such logarithms, but with opposite sign, also appear in the numerical evaluation of the full matrix element squared for real emission in the hard region of phase space, performed in 4 dimensions. In principle, these two potentially large contributions (logarithms of a potentially small quantity) originate from the unphysical division of the phase space and should thus cancel. Therefore, the key issue thus in phase space slicing is to adjust the parameters of the procedure such that the dependence on the slicing parameter is minimized. So far, this adjustment has been done manually only and this is one of the reasons why other methods have become more popular with practitioners of NLO calculations.

Such alternative methods of dealing with the real infrared divergences base on directly subtracting them [42, 97, 98, 99, 100, 101, 102, 103]. At NLO level the subtraction in all methods is performed such that the additional particle is added to the leading order matrix element in a well-defined way through terms which, on one hand, exhibit the correct divergent behavior in the soft and collinear limit, and, on the other hand, can easily be integrated over the full  $d$ -dimensional phase space of the extra particle. The idea is then that the so subtracted matrix element squared is finite and thus can safely be integrated numerically in 4 dimensions. On the other hand, the subtraction term is added to the virtual bit and integrated analytically in  $d$  dimensions. Again, it exhibits single or double poles of the form  $1/(d-4)$  or  $1/(d-4)^2$ , respectively. These poles again cancel the infrared poles of the virtual contributions. The fact that there are universal subtraction terms, i.e. terms

which will cancel the infrared divergences in a process-independent manner, is one of the main reasons why subtraction methods have become increasingly popular in past years and why they have been used for many of the state-of-the-art calculation of NLO corrections to physical processes, like for instance [25, 26, 27, 28, 29].

The universality of the subtraction terms also allow for an automated treatment of real infrared divergences. It is the subject of this chapter to report on a fully automated, process-independent implementation of one of the popular subtraction procedures, ready for use in realistic NLO calculations. Therefore, in section 5.1, the anatomy of QCD NLO calculations will be formalized in a more mathematical language and the chosen subtraction method, the Catani-Seymour dipole subtraction [42] will briefly be reviewed in its original form for massless particles. Although its extension to massive particles [101] is straightforward from an algorithmic point of view, this work concentrates on the massless case only. In section 5.2, the fully automated implementation of the corresponding massless dipole subtraction of arbitrary matrix elements into the matrix element generator AMEGIC++ [20] will be presented in some detail. Some simple tests of the implementation will be discussed in section 5.3, before some physical applications and the comparison with results from the literature will round off the presentation in section 5.4.

## 5.1 Brief review of the Catani-Seymour formalism

### 5.1.1 NLO cross sections and the subtraction procedure

Cross sections at NLO precision are given by

$$\sigma = \sigma^{\text{LO}} + \sigma^{\text{NLO}} , \quad (5.1)$$

where the LO part  $\sigma^{\text{LO}}$  is obtained by integrating the exclusive cross section in Born approximation over the available phase space of the  $m$  final state particles and, eventually, over the Bjorken- $x$  of incident partons. Ignoring this additional complication for the sake of a compact notation, The LO cross section is thus given by

$$\sigma^{\text{LO}} = \int_m d^{(4)}\sigma^B , \quad (5.2)$$

where

$$d^{(4)}\sigma^B = d^{(4)}\Phi^{(m)} |\mathcal{M}_m|^2 F_J^{(m)} . \quad (5.3)$$

Here,  $d^{(4)}\Phi^{(m)}$  denotes the phase space element of  $m$  particles, taken in four dimensions,  $\mathcal{M}_m$  is the matrix element for the process under consideration, and  $F_J^{(m)}$  is a function of cuts defining the jets etc.. As already indicated, here and in the following, the superscripts in the

integral denote the dimensionality of the integration. In order to obtain a meaningful result to be compared with experimental data, typically isolation cuts are applied on the outgoing particles, which may also serve the purpose of keeping the integral finite. A typical criterion for example is to identify outgoing partons with jets and thus apply jet definition cuts on the partons such that they are all well separated in phase space. Anyway, the cuts will not be stated explicitly in the integral, but they are understood implicitly with the integration, including suitable generalizations in  $d$  dimensions, where necessary. Thus, the integration of the Born level cross section can directly be carried out in four space-time dimensions, as indicated in the equation.

In view of the dipole subtraction formulae, it is useful to introduce at this point bras and kets  ${}_m\langle 1, \dots, m' |$  and  $| 1, \dots, m' \rangle_m$ . They denote states of  $m$  final state partons partons labeled by 1 to  $m'$  and are vectors in colour and helicity space. Introducing, in a similar fashion, vectors for the spins and colours, matrix elements thus can be written as

$$\mathcal{M}_m^{c_i, s_i} = ({}_m\langle c_1, \dots, c_m | \otimes {}_m\langle s_1, \dots, s_m |) | 1, \dots, m \rangle_m. \quad (5.4)$$

Therefore, in this notation, the matrix element squared, summed over final state colours and spins reads

$$|\mathcal{M}_m|^2 = {}_m\langle 1, \dots, m | 1, \dots, m \rangle_m. \quad (5.5)$$

The NLO part of the cross section consists of two contributions, each of which increases the order of  $\alpha_S$ . First, there are emissions of an additional parton, i.e. real corrections, denoted by  $d^{(d)}\sigma^R$ . Second, there are virtual (one-loop) corrections to the born matrix element, here denoted by  $d^{(d)}\sigma^V$ . Thus,

$$\sigma^{\text{NLO}} = \int d^{(d)}\sigma^{\text{NLO}} = \int_{m+1} d^{(d)}\sigma^R + \int_m d^{(d)}\sigma^V. \quad (5.6)$$

The two integrals on the right-hand side of Eq. (5.6) are separately infrared divergent in four dimensions, and are therefore taken in  $d$  dimensions. For the real correction, the divergences arise when the additional parton becomes soft or collinear w.r.t. some other parton, leading to on-shell propagators in the matrix element. For the virtual correction, the divergence comes with the integration over the unrestricted loop momentum, such that again a propagator goes on-shell. As already stated in the introduction, now the celebrated theorem of Kinoshita, Lee and Nauenberg [89, 90] comes to help and guarantees an exact cancellation of two divergent contributions, thus keeping their sum finite<sup>1</sup>. Setting  $d = 4 + 2\epsilon$  in the following, the divergences will manifest themselves in double and single poles, i.e. as

---

<sup>1</sup> In fact, this is only guaranteed for infrared-safe quantities. More specifically, if  $F_J^{(n)}$  defines jets in terms of the momenta of an  $m$ -parton final state (taken at Born level), infrared safety demands that  $F_J^{(m+1)} \rightarrow F_J^{(m)}$  in cases where the  $m+1$ - and  $m$ -parton configurations become kinematically degenerate.

$1/\epsilon^2$  and  $1/\epsilon$ , respectively. In principle, cancellation of the poles then solves the problem; in practice, however, the direct applicability of the equations above to real physical processes is limited since analytical integration over a multi-particle phase space in  $d$  dimensions with cuts in many cases is beyond current abilities.

Therefore, a detour has to be taken. The idea is to construct a subtraction term for the real emission contribution, which encodes all of its infrared divergences, but can analytically be integrated over in  $d$  dimensions. In this way the infrared pole structure of the real part with its  $1/\epsilon$  and  $1/\epsilon^2$  poles is exhibited and cancels the corresponding virtual contributions. Subtracting this term from the real emission contribution and adding it to the virtual corrections then eliminates the infrared divergences in both parts. The subtracted real matrix element squared then is finite and thus its full  $(m+1)$ -particle phase space can safely be integrated over in four dimensions. In this way, the subtraction term aims at an infrared regularization of the two contributions at integrand level.

$$\begin{aligned} \sigma^{\text{NLO}} &= \int_{m+1} d^{(d)}\sigma^R - \int_{m+1} d^{(d)}\sigma^A + \int_{m+1} d^{(d)}\sigma^A + \int_m d^{(d)}\sigma^V \\ &\rightarrow \int_{m+1} \left[ d^{(4)}\sigma^R - d^{(4)}\sigma^A \right] + \int_{m+1} d^{(d)}\sigma^A + \int_m d^{(d)}\sigma^V. \end{aligned} \quad (5.7)$$

The catch of the subtraction method now is that the subtraction terms can be obtained from the Born terms in a straightforward way and that only the phase space integral of the extra particle has to be taken in  $d$  dimensions, while the phase space for the remaining  $m$  particles can be taken in four dimensions. This is similar to the way, the loop terms are evaluated. There, only the loop integration is performed in  $d$  dimensions, whereas the phase space of the outgoing particles is done in four dimensions. Therefore, the final structure reads

$$\sigma^{\text{NLO}} = \int_{m+1} \left[ d^{(4)}\sigma^R - d^{(4)}\sigma^A \right] + \int_m \left[ \int_{\text{loop}} d^{(d)}\sigma^V + \int_1 d^{(d)}\sigma^A \right]_{\epsilon=0}. \quad (5.8)$$

Both integrands now are finite, allowing all integrations to be performed numerically. In contrast to some other regularization methods (like, e.g., phase space slicing) the subtraction method does not rely on any approximation and does not introduce any ambiguous and/or unphysical cut-off scales etc., as long as the integration of  $d^{(d)}\sigma^A$  can exactly and analytically be performed.

In [42] a general expression for  $d^{(d)}\sigma^A$  has been presented, called the dipole factorization formula, allowing to write

$$d^{(d)}\sigma^A = \sum_{\text{dipoles}} d^{(4)}\sigma^B \otimes d^{(d)}V_{\text{dipole}} \quad (5.9)$$

such that, symbolically,

$$\int_{m+1} d^{(d)}\sigma^A = \sum_{\text{dipoles}} \int_m d^{(4)}\sigma^B \otimes \int_1 d^{(d)}V_{\text{dipole}} = \int_m [d^{(4)}\sigma^B \otimes I], \quad (5.10)$$

where

$$I = \sum_{\text{dipoles}} \int_1^d V_{\text{dipole}}. \quad (5.11)$$

Here the sum of the dipole terms  $V_{\text{dipole}}$  contains all soft and collinear divergences of the real emission pattern. This factorization formula is suited for any process with massless partons, and fulfills all the requirements mentioned above. An extension to massive partons has been presented in [101].

However, as already mentioned in the introduction, in this publication only the massless case will be considered. In order to provide a self-contained description, all necessary analytic expressions will be listed in this publication.

### 5.1.2 Generalization to hadronic initial states

The cross sections discussed so far were given for point-like initial states. For cross sections in hadron collisions, however, the differential cross sections above must be convoluted with parton distribution functions (PDFs):

$$\sigma(p, p') = \sum_{a,b} \int_0^1 d\eta f_a(\eta, \mu_F^2) \int_0^1 d\eta' f_b(\eta', \mu_F^2) [\sigma_{ab}^{\text{LO}}(\eta p, \eta' p') + \sigma_{ab}^{\text{NLO}}(\eta p, \eta' p', \mu_F^2)]. \quad (5.12)$$

Here the subscripts on the cross section denote the flavours of the incoming partons; for the total cross section a sum over them has to be performed. For the NLO part, now the higher-order corrections residing in the PDFs must be taken care of. This is done by supplementing the NLO part with a collinear subtraction term  $d\sigma_{ab}^C$ , such that

$$\sigma_{ab}^{\text{NLO}}(p_a, p_b, \mu_F^2) = \int_{m+1}^1 d^{(d)}\sigma_{ab}^R(p_a, p_b) + \int_m^1 d^{(d)}\sigma_{ab}^V(p_a, p_b) + \int_m^1 d^{(d)}\sigma_{ab}^C(p_a, p_b, \mu_F^2). \quad (5.13)$$

This new term contains collinear singularities, incorporated in  $1/\epsilon$ -terms and reads

$$\begin{aligned} d^{(d)}\sigma_{ab}^C(p_a, p_b, \mu_F^2) = & -\frac{\alpha_S}{2\pi} \frac{1}{\Gamma(1-\epsilon)} \sum_{c,d} \int_0^1 dz \int_0^1 d\bar{z} \left\{ d^{(4)}\sigma_{cd}^B(z p_a, \bar{z} p_b) \right. \\ & \cdot \left[ \delta_{bd} \delta(1-\bar{z}) \left( -\frac{1}{\epsilon} \left( \frac{4\pi\mu^2}{\mu_F^2} \right)^\epsilon P_{ac}(z) + K_{ac}^{\text{F.S.}}(z) \right) \right. \\ & \left. \left. + \delta_{ac} \delta(1-z) \left( -\frac{1}{\epsilon} \left( \frac{4\pi\mu^2}{\mu_F^2} \right)^\epsilon P_{bd}(\bar{z}) + K_{bd}^{\text{F.S.}}(\bar{z}) \right) \right] \right\}. \end{aligned} \quad (5.14)$$

The collinear subtraction term is factorization-scale and scheme dependent. This scheme dependence resides in the terms  $K^{\text{F.S.}}$ , which, for the common  $\overline{MS}$ -scheme vanish, i.e. in this scheme all terms  $K^{\text{F.S.}} = 0$ . However, this scheme dependence cancels similar terms in the PDFs such that, taken together, the full hadronic cross section again is scheme-independent.

In the case of incoming hadrons, the subtraction method is applied to  $\sigma^{\text{NLO}}(p_a, p_b, \mu_F^2)$  as described before, with the only difference that in this case the singularities of  $d\sigma_{ab}^V$  only cancel in the sum

$$\int_m \left[ \int_{\text{loop}} d^{(d)}\sigma_{ab}^V + \int_1 d^{(d)}\sigma_{ab}^A + d^{(d)}\sigma_{ab}^C \right]_{\epsilon=0} . \quad (5.15)$$

### 5.1.3 Observable-independent formulation of the subtraction method

Up to now, the  $d\sigma$  denoted cross sections in a broad sense. To be a bit more specific consider the following expression for a cross section at Born-level and the corresponding next-to leading order expression:

$$\begin{aligned} \sigma^{\text{LO}} &= \int d\Phi^{(m)}(p_1, \dots, p_m) \left| M^{(m)}(p_1, \dots, p_m) \right|^2 F^{(m)}(p_1, \dots, p_m) \\ \sigma^{\text{NLO}} &= \int d\Phi^{(m+1)}(p_1, \dots, p_{m+1}) \left| M^{(m+1)}(p_1, \dots, p_{m+1}) \right|^2 F^{(m+1)}(p_1, \dots, p_{m+1}) \\ &\quad + \int d\Phi^{(m)}(p_1, \dots, p_m) \left| V^{(m)}(p_1, \dots, p_m) \right|^2 F^{(m)}(p_1, \dots, p_m) , \end{aligned} \quad (5.16)$$

where  $d\Phi^{(n)}$  represents an  $n$ -particle phase space element, and  $M^{(m)}$ ,  $M^{(m+1)}$  and  $V^{(m)}$  are the LO matrix element, the NLO real matrix element and the NLO virtual correction matrix element, respectively.  $F^{(n)}$  is a function that defines a cross section or an observable in terms of the  $n$ -parton momentum configuration. In general, the function  $F$  may contain  $\theta$ -functions (to define cuts and corresponding total cross sections),  $\delta$ -functions (defining differential cross sections), kinematic factors or any combination of these.

However arbitrary this sounds, there is a formal requirement on this function  $F$ , namely that in the soft and collinear limits, i.e. for cases where one parton becomes collinear w.r.t. another one or where one parton becomes soft, the function  $F^{(m+1)}$  reduces to  $F^{(m)}$ :

$$\begin{aligned} F^{(m+1)}(p_1, \dots, p_i = \lambda q, \dots, p_{m+1}) &\rightarrow F^{(m)}(p_1, \dots, p_{m+1}) \text{ for } \lambda \rightarrow 0 \\ F^{(m+1)}(p_1, \dots, p_i, \dots, p_j, \dots, p_{m+1}) &\rightarrow F^{(m)}(p_1, \dots, p, \dots, p_{m+1}) \text{ for } p_i \rightarrow zp, p_j \rightarrow (z-1)p \\ F^{(m)}(p_1, \dots, p_m) &\rightarrow 0 \text{ for } p_i \cdot p_j \rightarrow 0 . \end{aligned} \quad (5.17)$$

The first two conditions define infrared-safe observables - to phrase it intuitively this means that such infrared-safe quantities must not be altered by additional soft or collinear activity. The last condition above is required to properly define the Born cross section.

Applying the subtraction method to the NLO-part of Eq.(5.16) results in

$$\begin{aligned} \sigma^{\text{NLO}} = & \int d\Phi^{(m+1)} \left[ \left| M^{(m+1)}(p_1, \dots, p_{m+1}) \right|^2 F^{(m+1)}(p_1, \dots, p_{m+1}) \right. \\ & \left. - \sum_{k \neq i \neq j} \mathcal{D}_{ij,k}(p_1, \dots, p_{m+1}) F^{(m)}(p_1, \dots, \tilde{p}_{ij}, \tilde{p}_k, \dots, p_{m+1}) \right] \\ & + \int d\Phi^{(m)} \left[ \left| V^{(m)}(p_1, \dots, p_m) \right|^2 + \left( \int d[\tilde{p}] \mathcal{D}_{ij,k}(p_1, \dots, p_{m+1}) \right) \right] F^{(m)}(p_1, \dots, p_m), \end{aligned} \quad (5.18)$$

where  $d[\tilde{p}]$  is the phase space element for the 1-parton phase space.

In order to have an identity between the subtracted terms and the added term, both the  $(m+1)$ -parton contribution and the  $m$ -parton contribution have to be subjected to the same function  $F$ . To be able to perform the integration over the one-parton phase space independent of the observable this function therefore must be  $F^{(m)}$ . In the case of the  $(m+1)$ -parton contribution  $F^{(m)}$  is applied to the  $m$ -parton configuration, generated by corresponding mapping given in the prescription of the dipole function.

#### 5.1.4 The dipole subtraction functions

The universality of the soft and collinear limits of QCD matrix elements are the basis for the construction of the dipole subtraction terms. In both limits any matrix element squared for  $m+1$ -partons factorizes into an  $m$ -parton matrix element times a (singular) factor.

To be specific, consider first the soft limit of the matrix element, given by the momentum  $p_j$  of parton  $j$  becoming soft, i.e.  $p_j^\mu = \lambda q^\mu$  with  $\lambda \rightarrow 0$ . Then, employing

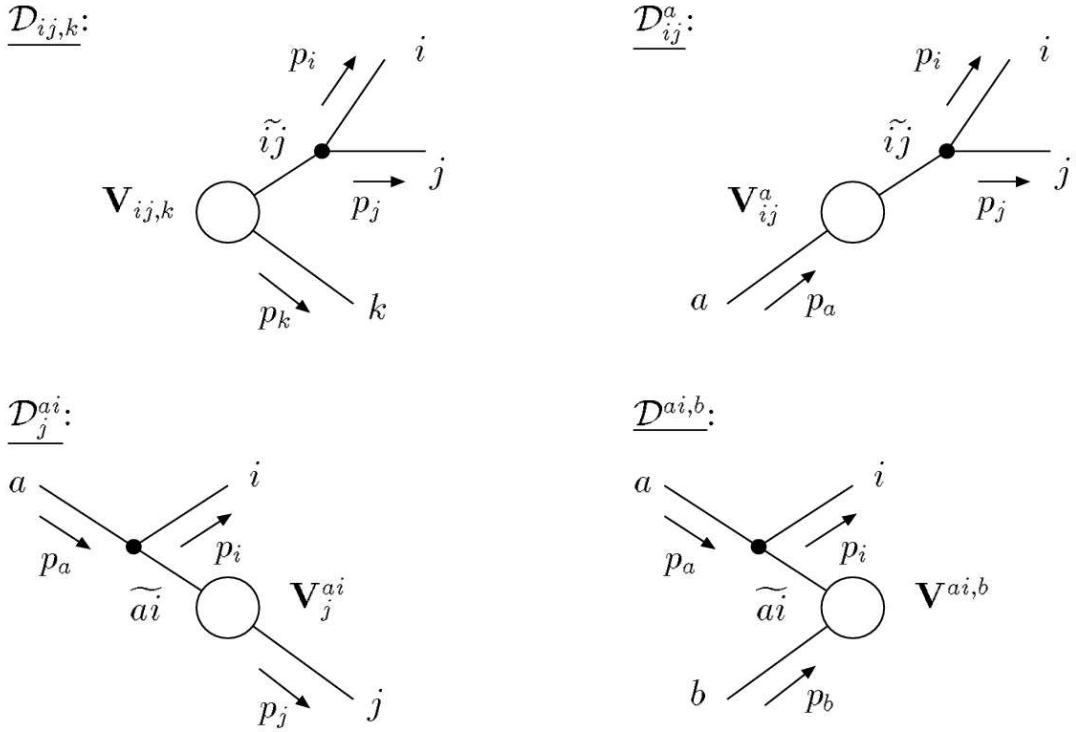
$$\frac{p_i p_k}{(p_i q)(p_k q)} = \frac{p_i p_k}{(p_i q)[(p_i + p_k)q]} + \frac{p_i p_k}{[(p_i + p_k)q](p_k q)}, \quad (5.19)$$

the soft limit reads

$$\begin{aligned} & {}_{m+1}\langle 1, \dots, j, \dots, m+1 | 1, \dots, j, \dots, m+1 \rangle_{m+1} \\ & \longrightarrow -\frac{1}{\lambda^2} 8\pi\mu^{2\epsilon} \alpha_S \sum_{i, k \neq i} \langle 1, \dots, i, \dots, m+1 \left| \frac{p_k p_i \mathbf{T}_k \cdot \mathbf{T}_i}{(p_i q)[(p_i + p_k)q]} \right| 1, \dots, k, \dots, m+1 \rangle_m. \end{aligned} \quad (5.20)$$

In a similar way the limit where two partons  $i$  and  $j$  become collinear is defined through  $p_j \rightarrow (1-z)/z p_i$ . In this limit the  $(m+1)$  parton matrix element can be rewritten as

$$\begin{aligned} & {}_{m+1}\langle 1, \dots, m+1 | 1, \dots, m+1 \rangle_{m+1} \\ & \longrightarrow \frac{1}{p_i p_j} 4\pi\mu^{2\epsilon} \alpha_S \langle 1, \dots, m+1 \left| \hat{P}_{(ij),i}(z, k_\perp) \right| 1, \dots, m+1 \rangle_m, \end{aligned} \quad (5.21)$$



**Figure 5.1:** Classification of dipole functions.

where, again, the  $\hat{P}_{(ij),i}(z, k_\perp)$  are the well-known Altarelli-Parisi splitting functions.

Then, the actual dipole function generating the limit, where one of the partons  $i, j$  of a  $m + 1$ -parton configuration becomes soft or both partons become collinear to each other, symbolically has the following structure:

$$\mathcal{D}_{ij,k} = {}_m\langle 1, \dots, i\tilde{j}, \dots, \tilde{k}, \dots, m || 1, \dots, i\tilde{j}, \dots, \tilde{k}, \dots, m \rangle_m \otimes \mathbf{V}_{ij,k}, \quad (5.22)$$

with the non-singular  $m$ -parton matrix element  ${}_m\langle \dots || \dots \rangle_m$  and the operator  $\mathbf{V}_{ij,k}$ , describing the splitting of the parton  $(ij)$ . Here, and in the following, the splitting kernels  $\mathbf{V}_{ij,k}$  are matrices in the helicity space of the emitter. The dipole function also involves a third parton as 'spectator'. This parton in fact is identical with the colour partner  $k$  in the soft limit, Eq. (5.20). The form of the subtraction means that kinematically,  $3 \rightarrow 2$  mappings are considered

$$p_i, p_j, p_k \rightarrow \tilde{p}_{ij}, \tilde{p}_k, \quad (5.23)$$

such that all involved partons are allowed to remain on their mass shells.

In general the splitting parton (called 'emitter') and the spectator can be both, initial and final state particles. This discriminates four different types of dipole functions, displayed in Fig. 5.1.

The full subtraction term for any matrix element with  $(m+1)$  partons in the final state is given by the sum of all possible dipole functions. For the most general case with two partons in the initial state, therefore

$$d\sigma^A = \left[ \sum_{k \neq i \neq j} \mathcal{D}_{ij,k} + \left\{ \sum_{i \neq j} \mathcal{D}_{ij}{}^a + \sum_{k \neq i} \mathcal{D}_k{}^{ai} + \sum_i \mathcal{D}^{ai,b} + (a \leftrightarrow b) \right\} \right] d\Phi^{(m+1)} . \quad (5.24)$$

In the following the explicit expressions for the dipole functions will be listed. The corresponding one-parton phase space integrated subtraction terms are discussed in Sec. 5.1.5.

### Final state emitters with final state spectators

The dipole contribution  $\mathcal{D}_{ij,k}$  for the singular limit  $p_i \cdot p_j \rightarrow 0$ , where all three involved partons are in the final state, is given by

$$\begin{aligned} & \mathcal{D}_{ij,k}(p_1, \dots, p_{m+1}) \\ &= -\frac{1}{2p_i \cdot p_j} \cdot \left\langle 1, \dots, \tilde{ij}, \dots, \tilde{k}, \dots, m+1 \left| \frac{\mathbf{T}_k \cdot \mathbf{T}_{ij}}{\mathbf{T}_{ij}^2} \mathbf{V}_{ij,k} \right| 1, \dots, \tilde{ij}, \dots, \tilde{k}, \dots, m+1 \right\rangle_m . \end{aligned} \quad (5.25)$$

It is obtained from an  $(m+1)$ -parton matrix element by replacing the partons  $i$  and  $j$  with a single parton  $\tilde{ij}$ , the emitter, and the parton  $k$  is replaced by  $\tilde{k}$ , the spectator. The flavours of emitter and spectator are assigned as follows: The spectator  $\tilde{k}$  remains unchanged, and the emitter  $\tilde{ij}$  is defined by the splitting process  $\tilde{ij} \rightarrow i+j$ . The product of colour charges in the numerator of Eq. (5.25) introduces an extra colour correlation in the  $m$ -parton matrix element.

The kinematics of the splitting are described by the following variables

$$y_{ij,k} = \frac{p_i p_j}{p_i p_j + p_j p_k + p_k p_i} , \quad \tilde{z}_i = \frac{p_i p_k}{p_j p_k + p_i p_k} = 1 - \tilde{z}_j . \quad (5.26)$$

and to obtain the momenta  $\tilde{ij}$  and  $\tilde{k}$  in the  $m$ -parton configuration the following map is being used:

$$\tilde{p}_k^\mu = \frac{1}{1 - y_{ij,k}} p_k^\mu , \quad \tilde{p}_{ij}^\mu = p_i^\mu + p_j^\mu - \frac{y_{ij,k}}{1 - y_{ij,k}} p_k^\mu . \quad (5.27)$$

Obviously, four-momentum conservation is exactly fulfilled, i.e.

$$p_i^\mu + p_j^\mu + p_k^\mu = \tilde{p}_{ij}^\mu + \tilde{p}_k^\mu \quad (5.28)$$

and all partons remain on their mass shell,

$$p_i^2 = p_j^2 = p_k^2 = \tilde{p}_{ij}^2 = \tilde{p}_k^2 = 0 . \quad (5.29)$$

The splitting matrices, which are related to the  $d$ -dimensional Altarelli-Parisi splitting functions, depend on the spin indices of the emitter parton. For the case of a quark splitting (using helicity indices  $s$  and  $s'$ ) the kernel is a matrix in helicity space, whereas for gluon splittings (to a quark-anti-quark pair or to gluons), the splitting matrices are given by Lorentz tensors. This yields

$$\begin{aligned} \langle s | V_{q_i g_j, k}(\tilde{z}_i; y_{ij,k}) | s' \rangle &= 8\pi\mu^{2\epsilon}\alpha_S C_F \left[ \frac{2}{1 - \tilde{z}_i(1 - y_{ij,k})} - (1 + \tilde{z}_i) - \epsilon(1 - \tilde{z}_i) \right] \delta_{ss'} , \\ \langle \mu | V_{q_i \bar{q}_j, k}(\tilde{z}_i; y_{ij,k}) | \nu \rangle &= 8\pi\mu^{2\epsilon}\alpha_S T_R \left[ -g^{\mu\nu} - \frac{2}{p_i p_j} (\tilde{z}_i p_i - \tilde{z}_j p_j)^\mu (\tilde{z}_i p_i - \tilde{z}_j p_j)^\nu \right] , \\ \langle \mu | V_{g_i g_j, k}(\tilde{z}_i; y_{ij,k}) | \nu \rangle &= 16\pi\mu^{2\epsilon}\alpha_S C_A \left[ -g^{\mu\nu} \left( \frac{1}{1 - \tilde{z}_i(1 - y_{ij,k})} + \frac{1}{1 - \tilde{z}_j(1 - y_{ij,k})} - 2 \right) \right. \\ &\quad \left. + (1 - \epsilon) \frac{2}{p_i p_j} (\tilde{z}_i p_i - \tilde{z}_j p_j)^\mu (\tilde{z}_i p_i - \tilde{z}_j p_j)^\nu \right] , \end{aligned} \quad (5.30)$$

respectively. The dipole terms given in this section are sufficient for the subtraction procedure in the case of non-hadronic initial states such as  $e^-e^+$ -annihilation.

### Final state emitters with initial state spectators

For the case of an emitting final state parton, the presence of an initial state spectator results in additional contributions to the singular limit  $p_i \cdot p_j \rightarrow 0$  of the full  $m + 1$ -parton matrix element. The corresponding dipole terms in this case are given by

$$\begin{aligned} \mathcal{D}_{ij}^a(p_1, \dots, p_{m+1}; p_a, \dots) &= -\frac{1}{2p_i \cdot p_j} \frac{1}{x_{ij,a}} \cdot \left\langle 1, \dots, i\tilde{j}, \dots, m+1; \tilde{a}, \dots \left| \frac{\mathbf{T}_a \cdot \mathbf{T}_{ij}}{\mathbf{T}_{ij}^2} \mathbf{V}_{ij}^a \right| 1, \dots, i\tilde{j}, \dots, m+1; \tilde{a}, \dots \right\rangle_{m,a} . \end{aligned} \quad (5.31)$$

The kinematic variables now read

$$x_{ij,a} = 1 - \frac{p_i p_j}{(p_i + p_j)p_a} , \quad \tilde{z}_i = \frac{p_i p_a}{p_j p_a + p_i p_a} = 1 - \tilde{z}_j \quad (5.32)$$

and the momenta of the  $m$ -parton configuration are obtained by the map

$$\tilde{p}_a^\mu = x_{ij,a} p_a^\mu , \quad \tilde{p}_{ij}^\mu = p_i^\mu + p_j^\mu - (1 - x_{ij,a})p_a^\mu . \quad (5.33)$$

Again, four-momentum conservation is trivially fulfilled and the partons remain massless.

The corresponding splitting functions used in Eq. (5.31) read

$$\begin{aligned}
\langle s | V_{q_i g_j}^a(\tilde{z}_i; x_{ij,a}) | s' \rangle &= 8\pi\mu^{2\epsilon}\alpha_S C_F \left[ \frac{2}{1 - \tilde{z}_i + (1 - x_{ij,a})} - (1 + \tilde{z}_i) - \epsilon(1 - \tilde{z}_i) \right] \delta_{ss'} , \\
\langle \mu | V_{q_i \bar{q}_j}^a(\tilde{z}_i; x_{ij,a}) | \nu \rangle &= 8\pi\mu^{2\epsilon}\alpha_S T_R \left[ -g^{\mu\nu} - \frac{2}{p_i p_j} (\tilde{z}_i p_i - \tilde{z}_j p_j)^\mu (\tilde{z}_i p_i - \tilde{z}_j p_j)^\nu \right] , \\
\langle \mu | V_{g_i g_j}^a(\tilde{z}_i; x_{ij,a}) | \nu \rangle &= 16\pi\mu^{2\epsilon}\alpha_S C_A \left[ -g^{\mu\nu} \left( \frac{1}{1 - \tilde{z}_i + (1 - x_{ij,a})} + \frac{1}{1 - \tilde{z}_j + (1 - x_{ij,a})} - 2 \right) \right. \\
&\quad \left. + (1 - \epsilon) \frac{2}{p_i p_j} (\tilde{z}_i p_i - \tilde{z}_j p_j)^\mu (\tilde{z}_i p_i - \tilde{z}_j p_j)^\nu \right] . \tag{5.34}
\end{aligned}$$

### Initial state emitters with final state spectators

The next type of dipole function now covers initial state singularities  $p_a \cdot p_i \rightarrow 0$  with final state spectators, given by

$$\begin{aligned}
\mathcal{D}_k^{ai}(p_1, \dots, p_{m+1}; p_a, \dots) &= -\frac{1}{2p_a \cdot p_i} \frac{1}{x_{ik,a}} \left\langle 1, \dots, \tilde{k}, \dots, m+1; \tilde{ai}, \dots \left| \frac{\mathbf{T}_k \cdot \mathbf{T}_{ai}}{\mathbf{T}_{ai}^2} \mathbf{V}_k^{ai} \right| 1, \dots, \tilde{k}, \dots, m+1; \tilde{ai}, \dots \right\rangle_{m,a} . \tag{5.35}
\end{aligned}$$

The parton  $\tilde{ai}$ , which enters into the  $m$ -parton matrix element on the r.h.s. of Eq. (5.35) is given by the splitting of the initial state parton  $a \rightarrow \tilde{ai} + i$ . The relevant kinematic variables in this case are

$$x_{ik,a} = 1 - \frac{p_i p_k}{(p_k + p_i)p_a} , \quad u_i = \frac{p_i p_a}{p_i p_a + p_k p_a} = 1 - u_k , \tag{5.36}$$

and the momenta for the  $m$ -parton configuration are obtained by

$$\tilde{p}_{ai}^\mu = x_{ik,a} p_a^\mu , \quad \tilde{p}_k^\mu = p_k^\mu + p_i^\mu - (1 - x_{ik,a}) p_a^\mu . \tag{5.37}$$

The splitting matrices  $\mathbf{V}_k^{ai}$  in Eq. (5.35) are

$$\begin{aligned}
\langle s | V_k^{q_a g_i}(u_i; x_{ik,a}) | s' \rangle &= 8\pi\mu^{2\epsilon}\alpha_S C_F \left[ \frac{2}{1 - x_{ik,a} + u_i} - (1 + x_{ik,a}) - \epsilon(1 - x_{ik,a}) \right] \delta_{ss'} , \\
\langle s | V_k^{g_a \bar{q}_i}(u_i; x_{ik,a}) | s' \rangle &= 8\pi\mu^{2\epsilon}\alpha_S C_F [1 - \epsilon - 2x_{ik,a}(1 - x_{ik,a})] \delta_{ss'} , \\
\langle \mu | V_k^{q_a q_i}(u_i; x_{ik,a}) | \nu \rangle &= 8\pi\mu^{2\epsilon}\alpha_S T_R \left[ -g^{\mu\nu} x_{ik,a} + \frac{2u_i u_k}{p_i p_k} \frac{1 - x_{ik,a}}{x_{ik,a}} \left( \frac{p_i}{u_i} - \frac{p_k}{u_k} \right)^\mu \left( \frac{p_i}{u_i} - \frac{p_k}{u_k} \right)^\nu \right] , \\
\langle \mu | V_k^{g_a g_i}(u_i; x_{ik,a}) | \nu \rangle &= 16\pi\mu^{2\epsilon}\alpha_S C_A \left[ -g^{\mu\nu} \left( \frac{1}{1 - x_{ik,a} + u_i} - 1 + x_{ik,a}(1 - x_{ik,a}) \right) \right. \\
&\quad \left. + (1 - \epsilon) + \frac{u_i u_k}{p_i p_k} \frac{1 - x_{ik,a}}{x_{ik,a}} \left( \frac{p_i}{u_i} - \frac{p_k}{u_k} \right)^\mu \left( \frac{p_i}{u_i} - \frac{p_k}{u_k} \right)^\nu \right] . \tag{5.38}
\end{aligned}$$

The three dipole types discussed up to now (FF, IF, FI) are sufficient to construct the subtraction term  $d\sigma^A$  for processes with exactly one initial state parton, i.e. DIS configurations.

### Initial state emitters with initial state spectators

The remaining dipole function, only required by processes with two initial state partons, covers the case where both, the emitter and the spectator, are initial state particles,

$$\begin{aligned} & \mathcal{D}^{ai,b}(p_1, \dots, p_{m+1}; p_a, p_b) \\ &= -\frac{1}{2p_a \cdot p_i} \frac{1}{x_{i,ab}} \cdot \left\langle \tilde{1}, \dots, m \tilde{+} 1; \tilde{ai}, b \left| \frac{\mathbf{T}_b \cdot \mathbf{T}_{ai}}{\mathbf{T}_{ai}^2} \mathbf{V}^{ai,b} \right| 1, \dots, m \tilde{+} 1; \tilde{ai}, b \right\rangle_{m,ab}. \end{aligned} \quad (5.39)$$

To describe the splitting, the following kinematic variables are used

$$x_{i,ab} = 1 - \frac{p_i p_a + p_i p_b}{p_a p_b}, \quad \tilde{v}_i = \frac{p_a p_i}{p_a p_b}. \quad (5.40)$$

The construction of the  $m$ -parton kinematics for this dipoles differs from the other three cases. The reason is that in this case the emitter and the spectator are fixed to remain along the beam axis. Therefore **all** final state momenta (not only momenta of QCD partons) are transformed according to the map

$$\tilde{p}_{ai}^\mu = x_{i,ab} p_a^\mu, \quad \tilde{p}_j^\mu = p_j^\mu - \frac{2p_j \cdot (K + \tilde{K})}{(K + \tilde{K})^2} (K + \tilde{K})^\mu + \frac{2p_j \cdot K}{K^2} \tilde{K}^\mu, \quad (5.41)$$

where

$$K^\mu = p_a^\mu + p_b^\mu - p_i^\mu \quad \text{and} \quad \tilde{K}^\mu = \tilde{p}_{ai}^\mu + p_b^\mu. \quad (5.42)$$

The momentum of the spectator  $p_b$  remains unchanged. The transformation above can also be interpreted as applying a rotation and a boost turning initial state momenta back to the beam axis after a mapping similar to the first three cases of dipole functions. Indeed it can be shown that the transformation of final state momenta in Eq. (5.41) is just a Lorentz transformation.

However, in this case, the splitting matrices read

$$\begin{aligned} \langle s | V^{q_a q_i, b}(x_{i,ab}) | s' \rangle &= 8\pi\mu^{2\epsilon}\alpha_S C_F \left[ \frac{2}{1 - x_{i,ab}} - (1 + x_{i,ab}) - \epsilon(1 - x_{i,ab}) \right] \delta_{ss'}, \\ \langle s | V^{g_a \bar{q}_i, b}(x_{i,ab}) | s' \rangle &= 8\pi\mu^{2\epsilon}\alpha_S T_R [1 - \epsilon - 2x_{i,ab}(1 - x_{i,ab})] \delta_{ss'}, \\ \langle \mu | V^{q_a q_i, b}(\tilde{v}_i; x_{ik,a}) | \nu \rangle &= 8\pi\mu^{2\epsilon}\alpha_S C_F \left[ -g^{\mu\nu} x_{i,ab} + \frac{2}{\tilde{v}_i p_i \cdot p_b} \frac{1 - x_{i,ab}}{x_{i,ab}} (p_i - \tilde{v}_i p_k)^\mu (p_i - \tilde{v}_i p_k)^\nu \right], \\ \langle \mu | V^{g_i g_a, b}(\tilde{v}_i; x_{ik,a}) | \nu \rangle &= 16\pi\mu^{2\epsilon}\alpha_S C_A \left[ -g^{\mu\nu} \left( \frac{x_{i,ab}}{1 - x_{i,ab}} + x_{i,ab}(1 - x_{i,ab}) \right) \right. \\ &\quad \left. + (1 - \epsilon) \frac{1}{\tilde{v}_i p_i \cdot p_b} \frac{1 - x_{i,ab}}{x_{i,ab}} (p_i - \tilde{v}_i p_k)^\mu (p_i - \tilde{v}_i p_k)^\nu \right]. \end{aligned} \quad (5.43)$$

### 5.1.5 Integrated dipole terms

#### Phase space factorization

In order to combine the poles of the subtraction function and the virtual matrix element the subtraction function has to be integrated analytically over the one-parton phase space of the respective splitting. The rules for the momentum mapping from 3 to 2 parton phase spaces have been constructed in Secs. 5.1.4-5.1.4 such that the corresponding phase space exactly factorizes.

As an example, and in order to fix the notation, the case of a final-final dipole,  $\mathcal{D}_{ij,k}$ , will be discussed in the following. There, the three-particle phase space for the partons  $i$ ,  $j$  and  $k$  (all other partons are not affected by the splitting and will be omitted) in  $d$  dimensions is given by

$$d\phi(p_i, p_j, p_k; Q) = \frac{d^d p_i}{(2\pi)^{d-1}} \delta_+(p_i^2) \frac{d^d p_j}{(2\pi)^{d-1}} \delta_+(p_j^2) \frac{d^d p_k}{(2\pi)^{d-1}} \delta_+(p_k^2) (2\pi)^d \delta^{(d)}(Q - p_i - p_j - p_k). \quad (5.44)$$

This can be factorized in terms of the mapped momenta, such that

$$d\phi(p_i, p_j, p_k; Q) = d\phi(\tilde{p}_{ij}, \tilde{p}_k; Q) [dp_i(\tilde{p}_{ij}, \tilde{p}_k)], \quad (5.45)$$

where  $[dp_i(\tilde{p}_{ij}, \tilde{p}_k)]$ , written in terms of the kinematic variables defined in section 5.1.4, reads

$$\begin{aligned} [dp_i(\tilde{p}_{ij}, \tilde{p}_k)] &= \frac{(2\tilde{p}_{ij}\tilde{p}_k)^{1-\epsilon}}{16\pi^2} \frac{d\Omega^{(d-3)}}{(2\pi)^{1-2\epsilon}} d\tilde{z}_i dy_{ij,k} \theta(\tilde{z}_i(1-\tilde{z}_i))\theta(y_{ij,k}(1-y_{ij,k})) \\ &\quad \cdot (\tilde{z}_i(1-\tilde{z}_i))^{-\epsilon} (1-y_{ij,k})^{1-2\epsilon} y_{ij,k}^{-\epsilon}. \end{aligned} \quad (5.46)$$

Within the dipole function only the splitting function itself depends on the variables  $\tilde{z}_i$  and  $y_{ij,k}$ . Thus, the integration in  $d$  dimensions can be performed once and for all, independent of the specific scattering process under consideration. The result of the integration for each splitting type can be expanded as a Laurent series including double poles ( $\sim 1/\epsilon^2$ ), single poles ( $\sim 1/\epsilon$ ), and finite terms ( $\sim \epsilon^0$ ). Further terms of  $\mathcal{O}(\epsilon)$  are unimportant here and will be left out.

All results for the final-final and for all other dipole types can be found in [42].

#### Full result

Having at hand the integrals for each dipole function, all individual dipoles present in a specific process can be collected to yield the overall infrared divergence of the subtraction term. Then, the starting point for the calculation of jet cross sections in the dipole subtraction

formalism reads

$$\sigma^{\text{NLO}} = \sum_{\{m+1\}} \int_{m+1} [d\sigma_{\{m+1\}}^R|_{\epsilon=0} - d\sigma_{\{m+1\}}^A|_{\epsilon=0}] + \int_m \left[ \sum_{\{m\}} d\sigma_{\{m\}}^V + \sum_{\{m+1\}} \int_1 d\sigma_{\{m+1\}}^A \right]_{\epsilon=0}, \quad (5.47)$$

where  $\sum_{\{m+1\}}$  denotes the sum over all parton-level processes. However, the important point here is to exactly cancel the poles of the corresponding individual one-loop parton-level processes, which is done exclusively for each momentum and flavour constellation. Therefore, for each specific  $m$ -parton process at NLO only a selection of dipole functions related to  $(m+1)$ -parton processes contributes to the cancellation of the virtual divergences. In [42] it has been shown that this amounts to an effective reordering of phase space integrals and sums over parton configurations, such that

$$\sigma^{\text{NLO}} = \sum_{\{m+1\}} \int_{m+1} [d\sigma_{\{m+1\}}^R|_{\epsilon=0} - d\sigma_{\{m+1\}}^A|_{\epsilon=0}] + \sum_{\{m\}} \int_m [d\sigma_{\{m\}}^V + d\sigma_{\{m\}}^{\tilde{A}}]_{\epsilon=0}, \quad (5.48)$$

where  $d\sigma_{\{m\}}^{\tilde{A}}$  is the integrated dipole term that collects the integrals of all dipole functions and thus cancels the singularities of  $d\sigma_{\{m\}}^V$ . It is explicitly given by

$$d\sigma_{\{m\}}^{\tilde{A}} = [d\sigma_{\{m\}}^B \times \mathbf{I}(\epsilon)], \quad (5.49)$$

where  $d\sigma_{\{m\}}^B \times \mathbf{I}(\epsilon)$  is a shorthand for the following procedure: Write down the expression for  $d\sigma_{\{m\}}^B$ , and replace the corresponding squared Born-level matrix element

$$|\mathcal{M}_{\{m\}}|^2 = {}_m\langle 1, \dots, m | 1, \dots, m \rangle_m \quad (5.50)$$

with

$${}_m\langle 1, \dots, m | \mathbf{I}(\epsilon) | 1, \dots, m \rangle_m, \quad (5.51)$$

using the insertion operator  $\mathbf{I}(\epsilon)$  as defined below.

Finally, the full result for the integrated dipole term and the collinear counterterm as defined in Eq. (5.14) for the most general case with hadronic initial states reads

$$\begin{aligned} d\sigma_{ab}^{\tilde{A}}(p_a, p_b) + d\sigma_{ab}^C(p_a, p_b, \mu_F^2) &= [d\sigma_{ab}^B(p_a, p_b) \times \mathbf{I}(\epsilon)] \\ &+ \sum_{a'} \int_0^1 dx \left[ \left( \mathbf{K}^{a,a'}(x) + \mathbf{P}^{a,a'}(xp_a, x; \mu_F^2) \right) \times d\sigma_{a'b}^B(xp_a, p_b) \right] \\ &+ \sum_{b'} \int_0^1 dx \left[ \left( \mathbf{K}^{b,b'}(x) + \mathbf{P}^{b,b'}(xp_b, x; \mu_F^2) \right) \times d\sigma_{ab'}^B(p_a, xp_b) \right], \end{aligned} \quad (5.52)$$

where  $a$  and  $b$  again specify the initial state partons. The summation over  $a'$  and  $b'$  runs over all parton flavours, i.e. it includes gluons, quarks and anti-quarks occurring in the PDF.

The insertion operator  $\mathbf{I}$  reads

$$\mathbf{I}(\{p\}; \epsilon) = -\frac{\alpha_S}{2\pi} \frac{1}{\Gamma(1-\epsilon)} \sum_I \frac{1}{\mathbf{T}_I^2} \mathcal{V}_I(\epsilon) \sum_{I \neq J} \mathbf{T}_I \cdot \mathbf{T}_J \left( \frac{4\pi\mu^2}{2p_I p_J} \right)^\epsilon, \quad (5.53)$$

where the indices  $I$  and  $J$  run over initial and final state partons. The universal singular functions  $\mathcal{V}_I(\epsilon)$  depend merely on the flavour of  $I$  and are given by

$$\begin{aligned} \mathcal{V}_q(\epsilon) &= C_F \left[ \frac{1}{\epsilon^2} + \frac{3}{2\epsilon} + 5 - \frac{\pi^2}{2} + \mathcal{O}(\epsilon) \right] \\ \mathcal{V}_g(\epsilon) &= \frac{C_A}{\epsilon^2} + \left( \frac{11}{6}C_A - \frac{2}{3}T_R N_f \right) \frac{1}{\epsilon} + C_A \left( \frac{50}{9} - \frac{\pi^2}{2} \right) - T_R N_f \frac{16}{9} + \mathcal{O}(\epsilon), \end{aligned} \quad (5.54)$$

with  $N_f$  being the number of contributing quark flavours.

The complete singular structure in Eq. (5.52) is contained in  $[\mathrm{d}\sigma_{ab}^B(p_a, p_b) \times \mathbf{I}(\epsilon)]$  and the sum  $[\mathrm{d}\sigma_{ab}^B(p_a, p_b) \times \mathbf{I}(\epsilon)] + \mathrm{d}\sigma_{ab}^V(p_a, p_b)$  must be finite for  $\epsilon \rightarrow 0$ .

The finite insertion operators  $\mathbf{K}$  and  $\mathbf{P}$  are given by

$$\begin{aligned} \mathbf{K}^{a,a'}(x) &= \frac{\alpha_S}{2\pi} \left\{ \bar{K}^{aa'}(x) - K_{\mathrm{F.S.}}^{aa'}(x) \right. \\ &\quad \left. + \delta^{aa'} \sum_i \mathbf{T}_i \cdot \mathbf{T}_a \frac{\gamma_i}{\mathbf{T}_i^2} \left[ \left( \frac{1}{1-x} \right)_+ + \delta(1-x) \right] - \frac{\mathbf{T}_b \cdot \mathbf{T}_{a'}}{\mathbf{T}_a^2} \tilde{K}^{a,a'}(x) \right\} \end{aligned} \quad (5.55)$$

and

$$\mathbf{P}^{a,a'}(\{p\}; x; \mu_F^2) = \frac{\alpha_S}{2\pi} P^{a,a'}(x) \frac{1}{\mathbf{T}_b^2} \sum_{I \neq b} \mathbf{T}_I \cdot \mathbf{T}_b \ln \frac{\mu_F^2}{2xp_a p_I}. \quad (5.56)$$

Note that here the index  $i$  runs over final state partons only. The flavour-dependent functions  $\bar{K}^{aa'}(x)$ ,  $\tilde{K}^{a,a'}(x)$ , and  $P^{a,a'}(x)$  are defined in Appendix B. As already mentioned, the factorization-scheme dependent function  $K_{\mathrm{F.S.}}^{aa'}(x)$  vanishes in the commonly used  $\overline{\mathrm{MS}}\text{-}$  scheme.

To obtain the final result for processes with no initial state partons only the  $\mathbf{I}$ -term needs to be considered in Eq. (5.52). For processes with one initial state parton only, the result is obtained by using the  $\mathbf{I}$ -term and one of the two integrals over  $\mathbf{K}$  and  $\mathbf{P}$  only, while omitting the contribution of  $\tilde{K}^{a,a'}(x)$ .

### 5.1.6 Freedom in the definition of dipole terms

As stressed before, the singular limits of the dipole functions are fixed by the requirement to cancel the singularities of the real correction matrix element. However, away from this limit there is some freedom for modifications.

One possible modification has been presented in [26], where a parameter  $\alpha$  has been introduced which cuts off a dipole function for phase space regions far enough away from the corresponding singularity. Its main advantage lies in a significant reduction of the average number of dipoles terms to be calculated for each phase space point of the  $(m + 1)$ -parton phase space of the real correction term. This constitutes an important alleviation of the calculational burden, since the total number of dipole terms grows approximately as  $m^3$ . The  $\alpha$ -modified subtraction terms also allow nontrivial checks of the implementation, since the total result must be independent of  $\alpha$ .

The  $\alpha$ -modified dipole functions have been defined as follows:

$$\begin{aligned}\mathcal{D}'_{ij,k} &= \mathcal{D}_{ij,k} \theta(\alpha - y_{ij,k}) , \\ \mathcal{D}'^a_{ij} &= \mathcal{D}^a_{ij} \theta(\alpha - 1 + x_{ij,a}) , \\ \mathcal{D}'^{ai}_k &= \mathcal{D}^{ai}_k \theta(\alpha - u_i) , \\ \mathcal{D}'^{ai,b} &= \mathcal{D}^{ai,b} \theta(\alpha - \tilde{v}_i) .\end{aligned}\tag{5.57}$$

They will be employed later, in the implementation presented in this paper. Of course, such a redefinition of the splitting kernels also requires a recalculation of their integrals. The new  $\alpha$ -dependent insertion operators **I** and **K** have been presented in [26].

Another simple modification is the addition of finite terms to the splitting functions, such as

$$\begin{aligned}V'_{ij,k} &= V_{ij,k} + y_{ij,k} * C , \\ V'^a_{ij} &= V^a_{ij} + (1 - x_{ij,a}) * C , \\ V'^{ai}_k &= V^{ai}_k + u_i * C , \\ V'^{ai,b} &= V^{ai,b} + \tilde{v}_i * C .\end{aligned}\tag{5.58}$$

The constant  $C$  directly ends up as a finite term in the integral of the splitting function and thus it can be easily included in the insertion operators of **I** and **K**, too. This again allows checks of the implementation, but it can also be employed to improve the numerical behaviour of the phase space integrals and to reduce the number of negative events.

## 5.2 Implementation in AMEGIC++

The Catani-Seymour dipole subtraction terms have been implemented in full generality into the automatic matrix element generator AMEGIC++, based on its version 2.0. In particular this translates into AMEGIC++ being able to automatically generate all relevant parts of the NLO matrix element within the subtraction method except for the virtual matrix element. It can be applied to any process with massless partons for which the real correction ME can be

generated, an extension to allow also for massive particles is foreseen. This includes standard model processes as well as implemented extensions, as long as there are no new strongly interacting particles involved. For standard model processes the boundary is currently at about six-eight partons (initial and final state).

The new implementation aimed at a maximal reuse of already developed automated methods of amplitude generation and process management. A brief overview over the relevant parts of the code on which this implementation is based can be found in section 2.1.1.

### 5.2.1 Generation of CS dipole terms

#### Colour and spin correlations

The starting point of the Catani-Seymour algorithm is detailed in Eqs. (5.48) and (5.49), supplemented with expressions like the one in Eq. (5.22) for the individual dipole subtraction terms. The latter states that for any given process the Catani-Seymour dipole subtraction term for the real  $(m + 1)$ -parton correction term consists of the corresponding  $m$ -parton matrix element at Born level plus an additional operator that acts on colour and spin space. For the latter, only the limit  $\epsilon \rightarrow 0$  needs to be considered.

- Colour operator:

In all four dipoles, Eq. (5.25), (5.31), (5.35), and (5.39) colour-correlated tree-amplitudes of the form

$$|M_m^{i,k}|^2 = \langle 1, \dots, m | \mathbf{T}_i \cdot \mathbf{T}_k | 1, \dots, m \rangle_m \quad (5.59)$$

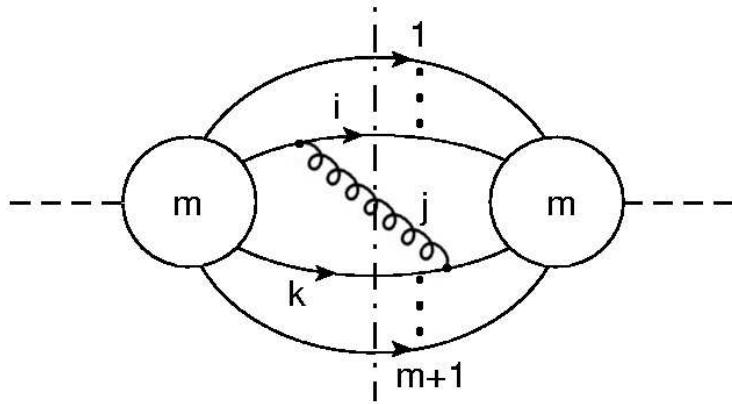
occur, where  $i$  labels the emitter and  $k$  the spectator. Denoting the colour indices of the external legs of the tree process explicitly by  $a_j$  and  $b_j$ , this can be cast into

$$\begin{aligned} & |M_m^{i,k}|^2 \\ &= \langle 1^{a_1} \dots i^{a_i} \dots k^{a_k} \dots m^{a_m} | \delta_{a_1 b_1} \dots T_{a_i b_i}^c \dots T_{a_k b_k}^c \dots \delta_{a_m b_m} | 1^{b_1} \dots i^{b_i} \dots k^{b_k} \dots m^{b_m} \rangle_m, \end{aligned} \quad (5.60)$$

where  $T_{ab}^c = i f^{acb}$ , if the associated particle is a gluon, and  $T_{ij}^c = t_{ij}^c$ , if the associated particle is a quark. In other words, the colour structure for dipole terms can be generated by adding a gluon connecting the emitter with the spectator as illustrated in Fig. 5.2. The colour matrix for a dipole term is recomputed after this insertion using the available evaluation tool in AMEGIC++.

- Spin space:

For a quark splitting all spin-matrices are just proportional to  $\delta_{ss'}$ , translating the quark spin to be exactly the same as for the Born-level  $m$ -parton matrix element.



**Figure 5.2:** Construction of the colour matrix for dipole terms: a gluon connects emitter and the spectator.

For the case of a gluon splitting, however, there are non-trivial correlation matrices. All of them can be cast into the generic form

$$V^{\mu\nu} = \langle \mu | V | \nu \rangle \propto -g^{\mu\nu} + \frac{\tilde{p}^\mu \tilde{p}^\nu}{B \tilde{p}^2}, \quad (5.61)$$

where  $B$  and  $\tilde{p}$  are functions of the kinematic variables and momenta of the corresponding splitting. Their values are listed in Table 5.1.

The structure of the splitting tensor as given in Eq. (5.61) is very similar to the polarization sum for massive vector bosons in unitary gauge, except for the factor  $B$  and the fact that  $\tilde{p}$  can be timelike or spacelike. This analogy can be used to replace the tensor by a polarization sum, i.e.

$$-g^{\mu\nu} + \frac{\tilde{p}^\mu \tilde{p}^\nu}{B \tilde{p}^2} = \sum_\lambda \xi^\lambda \epsilon_\lambda^\mu(\tilde{p}, B) (\epsilon_\lambda^\nu(\tilde{p}, B))^* . \quad (5.62)$$

Here the summation index  $\lambda$  runs over four values,  $+$ ,  $-$ ,  $l$  and  $s$ .  $\xi^\lambda$  is a sign that cannot be absorbed into the polarization vectors  $\epsilon_\lambda$ . For a gauge boson with momentum

$$\tilde{p}^\mu = \left( \tilde{p}_0, |\tilde{\vec{p}}| \sin \theta \cos \phi, |\tilde{\vec{p}}| \sin \theta \sin \phi, |\tilde{\vec{p}}| \cos \theta \right), \quad (5.63)$$

the polarization vectors are defined as

$$\begin{aligned}
\epsilon_{\pm}^{\mu} &= \frac{1}{\sqrt{2}} (0, \cos \theta \cos \phi \mp i \sin \phi, \cos \theta \sin \phi \pm i \cos \phi, -\sin \theta) , \\
\epsilon_l^{\mu} &= \frac{1}{\tilde{p}^2} \left( |\tilde{\vec{p}}|, \tilde{p}_0 \frac{\tilde{\vec{p}}}{|\tilde{\vec{p}}|} \right) , \\
\epsilon_s^{\mu} &= \sqrt{\frac{1-B}{B, \tilde{p}^2}} \tilde{p}^{\mu} ,
\end{aligned} \tag{5.64}$$

dipole type	splitting:	$\tilde{p}^\mu$	$B$
FF	$g \rightarrow q\bar{q}$	$\tilde{z}_i p_i^\mu - \tilde{z}_j p_j^\mu$	$1/(4\tilde{z}_i \tilde{z}_j)$
	$g \rightarrow gg$		$\left(2 - \frac{1}{1-\tilde{z}_i(1-y_{ij,k})} - \frac{1}{1-\tilde{z}_j(1-y_{ij,k})}\right) / (2\tilde{z}_i \tilde{z}_j)$
FI	$g \rightarrow q\bar{q}$	$\tilde{z}_i p_i^\mu - \tilde{z}_j p_j^\mu$	$1/(4\tilde{z}_i \tilde{z}_j)$
	$g \rightarrow gg$		$\left(2 - \frac{1}{1-\tilde{z}_i+(1-x_{ij,a})} - \frac{1}{1-\tilde{z}_j+(1-x_{ij,a})}\right) / (2\tilde{z}_i \tilde{z}_j)$
IF	$g \rightarrow q\bar{q}$	$\frac{p_i^\mu}{u_i} - \frac{p_k^\mu}{u_k}$	$-\frac{1}{4} x_{ik,a}^2 / (1 - x_{ik,a})$
	$g \rightarrow gg$		$\frac{1}{2} \left(1 - \frac{1}{1-x_{ik,a}+u_i} - x_{ik,a}(1 - x_{ik,a})\right) x_{ik,a} / (1 - x_{ik,a})$
II	$g \rightarrow q\bar{q}$	$p_i^\mu - \tilde{v}_i p_k^\mu$	$-\frac{1}{4} x_{i,ab}^2 / (1 - x_{i,ab})$
	$g \rightarrow gg$		$-\frac{1}{2} \left(\frac{1}{1-x_{ik,a}} + (1 - x_{ik,a})\right) x_{ik,a}^2 / (1 - x_{ik,a})$

**Table 5.1:** Values for the functions defined in Eq. (5.61). The variables are defined in the corresponding sections 5.1.4-5.1.4. The dipole type FF refers to the case where emitter and spectator are final state partons, IF refers to the case where the emitter is an initial state parton and the spectator a final state parton, etc..

and the sign factors are given by

$$\xi^\pm = 1, \quad \xi^l = \begin{cases} -1 & \text{if } \tilde{p}^2 < 0 \\ +1 & \text{if } \tilde{p}^2 > 0 \end{cases}, \quad \xi^s = \begin{cases} +1 & \text{if } \tilde{p}^2 < 0 \\ -1 & \text{if } \tilde{p}^2 > 0; B > 1 \\ +1 & \text{if } \tilde{p}^2 > 0; B < 1 \end{cases}. \quad (5.65)$$

In order to calculate the dipole matrix element, the polarization vectors of the splitting gluon are then replaced by the ones defined above.

### Organization and process management

To construct all dipole functions necessary to cancel the infrared divergencies of a given parton level real-correction process firstly all pairs of partons have to be determined that might emerge from the splitting of an emitter parton (initial state partons are charge conjugated for this procedure). This might be any quark (or anti-quark) and a gluon, two gluons or a quark and an anti-quark of the same flavour. Secondly, each of those pairs is combined with any possible third parton (acting as spectator) to define all possible dipole functions.

Any individual dipole function is thus specified by:

1. type (the specific combination of initial and final state for emitter and spectator),
2. the specific flavours involved in the splitting, and
3. the corresponding  $m$ -parton matrix element and its emitter and spectator particles.

In order to construct the individual dipole functions, given by

$$\mathcal{D} = A_i C'_{ij} A_j^* F(\dots) . \quad (5.66)$$

the following ingredients are necessary:

1. A rule to map the  $(m + 1)$ -parton phase space onto an  $m$ -parton phase space.
2. The corresponding splitting function for the dipole. This consists of two parts, a scalar function  $F(\dots)$  of the kinematic variables of the splitting and a spin correlation matrix. As discussed above, for quark splittings the matrix is simply  $\delta_{ss'}$ , for gluon splitting the matrix is represented by an outer product of pseudo-polarization vectors, which are also functions of the kinematic variables of the splitting.
3. The colour matrix  $C'_{ij}$ , respecting the extra colour correlation.
4. Amplitudes  $A_i$  of the corresponding  $m$ -parton matrix elements. For gluon splitting cases these amplitudes have to be calculated replacing polarization vectors of the splitting gluon by the pseudo-polarization vectors introduced above.

The calculation of any dipole function is organized in the class `Single_DipoleTerm`, each instance of this class representing one dipole. This class controls the ingredients for the calculation: Firstly there is a Born-level  $m$ -parton matrix element of the original AMEGIC++ implementation, just extended such that it includes the additional colour correlation. Secondly there is a class `Dipole_Splitting_Base` that completely organizes the splitting function itself. Specified by the type of the dipole (initial and final states for emitter and spectator) and the type of the splitting (determined by the contributing flavours) it takes care of the mapping between the  $m + 1$ -parton and the  $m$ -parton phase spaces and of the calculation of the splitting function (including the polarization vectors to encode the spin correlation).

Above that the class `Single_Real_Correction` handles all contributions to an infrared regularized parton level process. This consists firstly of an  $(m + 1)$ -parton tree level matrix element in the original AMEGIC++ implementation. Secondly it contains a list of single dipole functions, simply determined by looping over all partons and selecting valid dipole configurations. The classes `Single_Real_Correction` and `Single_Process` are derived from a common base class in a way such that the class `Process_Group` can be reused to also organize the infrared regularized parton level process in groups of common features up to all subprocesses contributing to a jet cross section.

Similarly to the case of tree level processes in AMEGIC++, also here a mapping of parton level processes that lead to identical or proportional results can be used to speed up the calculation and save computer resources. To this end, the following automatic identification strategies are implemented:

- If two real correction processes can be mapped (using strategies described in section 2.1.1) then also the whole `Single_Real_Correction` is mapped.
- For single dipole terms a unique identification algorithm proceeds as follows: Two terms can be mapped if the included  $m$ -parton process can be mapped and if the three particle labels (numbering the the external particles of the real correction process) to identify a dipole are identical.
- Many of the born matrix elements within the dipole terms will be identical. However, since different dipoles require different momentum mappings they have to be recalculated. Only the calculation routine can be shared.

### 5.2.2 Generation of the finite part of integrated dipole terms

#### Analytical structure of the full result

The starting point of the discussion of the finite pieces of the integrated dipole terms is Eq. (5.52), where now the phase space integration as well as the summation and integration over the incoming parton flavours and momenta is made explicit. Then, terms inside the  $m$ -parton integral come from subtraction terms integrated over the phase space of the extra parton emission and from the collinear counterterm for the general case of a NLO cross section with initial state partons. The terms inside the  $(m+1)$ -parton phase space integral in contrast corresponds to the dipole subtraction bit. Altogether, and including the convolution with parton distribution, the relevant term to be evaluated can thus be cast into

$$\begin{aligned}
& \sum_{a,b} \int d\eta_1 d\eta_2 f_a(\eta_1, \mu_F^2) f_b(\eta_2, \mu_F^2) \left\{ \int_{m+1} d\sigma_{ab}^A(\eta_1 p, \eta_2 \bar{p}) + \int_m d\sigma_{ab}^C(\eta_1 p, \eta_2 \bar{p}, \mu_F^2) \right\} \\
&= \sum_{a,b} \int d\eta_1 d\eta_2 f_a(\eta_1, \mu_F^2) f_b(\eta_2, \mu_F^2) \left\{ \int_m \left[ d\sigma_{ab}^B(\eta_1 p, \eta_2 \bar{p}) \times \mathbf{I}(\epsilon) \right] \right. \\
&\quad + \sum_{a'} \int_0^1 dx \int_m \left[ \left( \mathbf{K}^{a,a'}(x) + \mathbf{P}^{a,a'}(x\eta_1 p, x; \mu_F^2) \right) \times d\sigma_{a'b}^B(x\eta_1 p, \eta_2 \bar{p}) \right] \\
&\quad \left. + \sum_{b'} \int_0^1 dx \int_m \left[ \left( \mathbf{K}^{b,b'}(x) + \mathbf{P}^{b,b'}(x\eta_2 \bar{p}, x; \mu_F^2) \right) \times d\sigma_{ab'}^B(\eta_1 p, x\eta_2 \bar{p}) \right] \right\}. \tag{5.67}
\end{aligned}$$

The only correlation of the insertion operators  $\mathbf{I}$ ,  $\mathbf{P}$ , and  $\mathbf{K}$  with the Born level matrix element is within colour space. To be more specific, this implies that only the following structures emerge

$$\begin{aligned}
d\sigma_{ab}^B(p_a, p_b) &= {}_m \langle 1, \dots, m; a, b | 1, \dots, m; a, b \rangle_m \text{ and} \\
d\sigma_{ab}^{B(i,j)}(p_a, p_b) &= {}_m \langle 1, \dots, m; a, b | \mathbf{T}_i \cdot \mathbf{T}_j | 1, \dots, m; a, b \rangle_m
\end{aligned} \tag{5.68}$$

for all  $i \neq j$ , where  $i$  and  $j$  may label both final and initial state partons. Since any of the appearing matrix elements with insertion operators can be written as a sum of such structures, the colour factors will be skipped in the following and the operators will be treated simply as scalar functions.

The terms  $\mathbf{P}$  and  $\mathbf{K}$  induce dependences on  $x$ , which combined yield result in the structure

$$(g(x))_+ + \delta(1-x)h(x) + k(x) . \quad (5.69)$$

Here,  $h(x)$  and  $k(x)$  are regular functions in  $x$  and the '+'-distribution is defined by its action on a generic test function  $a(x)$

$$\int_0^1 dx a(x) (g(x))_+ = \int_0^1 dx [a(x) - a(1)] g(x) . \quad (5.70)$$

Then the r.h.s. of Eq. (5.67) can be cast into the form

$$\begin{aligned} & \sum_{a,b} \int d\eta_1 d\eta_2 f_a(\eta_1, \mu_F^2) f_b(\eta_2, \mu_F^2) \\ & \int_m \left\{ \mathbf{I}(\epsilon) d\sigma_{ab}^B(\eta_1 p, \eta_2 \bar{p}) \right. \\ & + \sum_{a'} \left[ \int_0^1 dx \left( g^{a,a'}(x) [d\sigma_{a'b}^B(x\eta_1 p, \eta_2 \bar{p}) - d\sigma_{a'b}^B(\eta_1 p, \eta_2 \bar{p})] + k^{a,a'}(x) d\sigma_{a'b}^B(x\eta_1 p, \eta_2 \bar{p}) \right) \right. \\ & \quad \left. + h^{a,a'}(1) d\sigma_{a'b}^B(\eta_1 p, \eta_2 \bar{p}) \right] \\ & + \sum_{b'} \left[ \int_0^1 dx \left( g^{b,b'}(x) [d\sigma_{ab'}^B(\eta_1 p, x\eta_2 \bar{p}) - d\sigma_{ab'}^B(\eta_1 p, \eta_2 \bar{p})] + k^{b,b'}(x) d\sigma_{ab'}^B(\eta_1 p, x\eta_2 \bar{p}) \right) \right. \\ & \quad \left. \left. + h^{b,b'}(1) d\sigma_{ab'}^B(\eta_1 p, \eta_2 \bar{p}) \right] \right\} . \end{aligned} \quad (5.71)$$

The functions  $g^{a,a'}(x)$ ,  $k^{a,a'}(x)$ , and  $h^{a,a'}(1)$  can be read off the corresponding functions in App. B.

Computationally the most demanding part is the actual Born-level cross section  $d\sigma_{ab}^B$ , due to its potentially expensive multi-particle matrix element, which typically suffers from factorial growth with the number of external particles. Thus, the calculation can be significantly accelerated if the expression is rearranged such that  $d\sigma_{ab}^B$  has to be computed only once for a single configuration at a given phase space point. This can be achieved by changing the integration variables  $\eta$  to  $\eta' = x\eta$ . After renaming  $\eta'$  back to  $\eta$  and reordering the

summation over  $a$  and  $a'$  ( $b$  and  $b'$ ) the expression above reads

$$\begin{aligned} \sum_{a,b} \int d\eta_1 d\eta_2 f_a(\eta_1, \mu_F^2) f_b(\eta_2, \mu_F^2) \int_m d\sigma_{ab}^B(\eta_1 p, \eta_2 \bar{p}) \times \left\{ \mathbf{I}(\epsilon) \right. \\ + \sum_{a'} \int_{\eta_1}^1 dx \left[ \frac{f_{a'}(\eta_1/x, \mu_F^2)}{x f_a(\eta_1, \mu_F^2)} \left( g^{a',a}(x) + k^{a',a}(x) \right) - \frac{f_{a'}(\eta_1, \mu_F^2)}{f_a(\eta_1, \mu_F^2)} g^{a',a}(x) \right] \\ + \sum_{a'} \frac{f_{a'}(\eta_1, \mu_F^2)}{f_a(\eta_1, \mu_F^2)} \left( h^{a',a} - G^{a',a}(\eta_1) \right) \\ + \sum_{b'} \int_{\eta_2}^1 dx \left[ \frac{f_{b'}(\eta_2/x, \mu_F^2)}{x f_b(\eta_2, \mu_F^2)} \left( g^{b',b}(x) + k^{b',b}(x) \right) - \frac{f_{b'}(\eta_2, \mu_F^2)}{f_b(\eta_2, \mu_F^2)} g^{b',b}(x) \right] \\ \left. + \sum_{b'} \frac{f_{b'}(\eta_2, \mu_F^2)}{f_b(\eta_2, \mu_F^2)} \left( h^{b',b} - G^{b',b}(\eta_2) \right) \right\}, \end{aligned} \quad (5.72)$$

where the  $G^{a,b}(\eta) = \int_0^\eta dx g^{a,b}(x)$  are analytically computed.

The insertion operator  $\mathbf{I}(\epsilon)$ , Eq. (5.53) is given as a Laurent series in  $\epsilon$ . For the implementation the interesting part is  $\propto \epsilon^0$ , since the poles must have been analytically extracted before<sup>2</sup>.

## Implementation and Organization

The numerical calculation of the finite contributions from integrated counterterms is organized as Eq. (5.72) suggests, i.e. the basic unit (class `Single_Virtual_Correction`) covers everything that is associated with a specific  $m$ -parton cross section.

For the actual calculation, basically all colour correlated matrix elements in Eq. (5.68) are necessary. The contributing amplitudes are, of course, the same for all of them, only the colour matrix is different. Therefore, a generalized version of `Single_Process` is employed that is able to deal with a multitude of colour matrices to calculate all required matrix elements at once. Anything else needed for the calculation of the finite contribution is a long list of rather simple scalar functions and constants. The integration over  $x$  is done numerically, i.e. for each set of external momenta  $x$  is diced within the corresponding interval.

### 5.2.3 Phase space integration

Together with the automatic generation of matrix elements AMEGIC++ also generates specific, process-dependent phase-space mappings for efficient integration. The underlying principles and the general procedure have been described in chapter 3. This implementation

<sup>2</sup> For testing purposes, however, it is trivial to also determine the coefficients of the  $\epsilon^{-2}$ - and  $\epsilon^{-1}$ -poles and to compare with known results of virtual correction terms.

can be widely reused for the phase space integrals coming from the subtraction procedure, which necessitates the evaluation of two independent integrals, namely integrals over the  $m$ -parton and the  $(m + 1)$ -parton phase space. In both cases mappings generated for the tree level process of the same dimensionality are used.

For the integration of the  $(m + 1)$ -parton phase space soft and collinear regions must be included. In this case the lower limit for the invariant masses of many propagators (e.g. Eq. (3.19)) must be zero. To keep the integral over the weight finite the adjustable exponents  $\nu$  in Eqs. (3.19) and (3.27) must be properly set. The actual shape of those propagators is hard to specify *a priori*. It depends on the jet definition and on the balance between the real correction process and the subtraction term (the integrand can be positive or negative). Taken together, however, it seems not unreasonable to assume a small exponent. Using the improved multi-channel method, described in section 3.2.3, the incorporated VEGAS refinement adapts very good to the actual shape of the concerned propagators and the final integration efficiency after optimization has only a weak dependence on the initial values of  $\nu$ . Since the VEGAS algorithm optimizes on the variance of the integrand it can, to some extend, also deal with the numerical problems related to “missed binning”, which will be discussed in the following section.

The  $m$ -parton phase space is much simpler. Since most parts of the integrand are proportional to the born matrix element it tends to work very well with the original tree process phase space setup.

### 5.2.4 Cuts and analysis framework for NLO calculations

Triggers and observables for NLO calculations have to be chosen with care. The general strict requirement not to spoil the cancelation of infrared divergencies has already been discussed in section 5.1.3.

Before going into any details concerning cuts, it is important to notice that a rule is mandatory of how cuts act on the different contributions to the NLO cross section. This rule must exist in a  $m$ -parton and a  $(m + 1)$ -parton version, where the latter needs to satisfy the conditions of infrared safety in degenerate phase space regions. In practical terms, this implies that the  $(m + 1)$ -parton version of the cut must allow for exactly one parton to become soft or collinear, while the  $m$ -parton version has to omit all singular regions.

Second, Eq. (5.18) requires for the cut of the  $m + 1$  phase space integral to be applied separately to the real correction process (using the  $m + 1$ -parton version) and to each dipole term (using the  $m$ -parton version, applied on the momenta of the mapped  $m$ -parton configuration). In general there might be kinematic configurations, where the real correction process ends up outside the accepted phase space region but some dipole terms do not and

vice versa. This leads to the problem of “missed binning”: if such a configuration occurs close to a singular region, large contributions result, which do not cancel completely. Ultimately, this leads to large numerical fluctuations, which need to be addressed. This is a common issue for all subtraction methods.

So far, the following cuts have been made available in AMEGIC++:

- A simple cut for jets is implemented as follows: a suitable jet algorithm (e.g.  $k_T$ ) [32, 33, 18, 104] is used to construct jets from the final state partons and their momenta. Then the number of jets above a given  $p_T$ -cut is counted. A phase space point is valid if this number is greater or equal  $m$ .
- Of course also cuts that only act on particles not taking part in strong interactions can be applied. If initial-initial dipoles are present this also has to be done separately for the real correction and for the dipole terms, since the momentum mapping in this case modifies all final state particle momenta. Implemented are cuts on invariant masses, on total or transverse energies, on rapidities or on particle angles w.r.t. the beam.

Sherpa’s ANALYSIS-package has been extended to be able to deal with weighted events from the NLO subtraction procedure. For example, and to be more specific, consider the case of a cross section which is differential to some infrared safe quantity  $F$ , i.e. a distribution to be binned in a histogram  $dF$ . For the  $m$ -parton integral no special treatment is mandatory: for a given momentum configuration,  $dF$  can directly be evaluated and filled into the corresponding bin. For the real correction and the dipole subtraction functions in the  $(m + 1)$ -parton integral,  $F$  has to be evaluated for each contribution separately, similar to the phase-space cut. Again, the problem of “missed binnings” appears, if contributions to a single event end up in more than one bin.

## 5.3 Checks of the implementation

In this section a number of tests of the correct implementation of the subtraction algorithm and of the integration routines are described. These tests are mainly technical in nature, results relating to truly physical observables are discussed in section 5.4.

### 5.3.1 Explicit comparisons

Before moving on to technical checks, it is worth stating that a number of direct comparisons of individual terms from the program presented here with those obtained from M. Seymour’s Fortran code DISENT have been performed. The latter is a dedicated program to compute

NLO cross sections for the deep inelastic scattering processes  $e^- p \rightarrow e^- + jet$ ,  $e^- p \rightarrow e^- + 2jets$  and for electron-positron annihilation to two and three jets. This direct comparison is possible, since DISENT uses exactly the same subtraction formalism, allowing to compare individual terms at given phase space points. All terms listed in the following showed full agreement of the two codes, up to the numerical precision.

The comparison included:

- Dipole subtraction terms for the real correction:  
all flavour configurations for dipoles with final state emitters and spectators as well as for dipoles with initial state emitters / final state spectators and final state emitters / initial state spectators have been checked.
- Terms from the finite part of the insertion operator  $\mathbf{I}$ ,  
cf. Eqs. (5.49) and (5.53).
- Terms from the insertion operators  $\mathbf{K}$  and  $\mathbf{P}$  for the case of one initial state parton,  
cf. Eq. (5.52) and the implemented version Eq. (5.72).

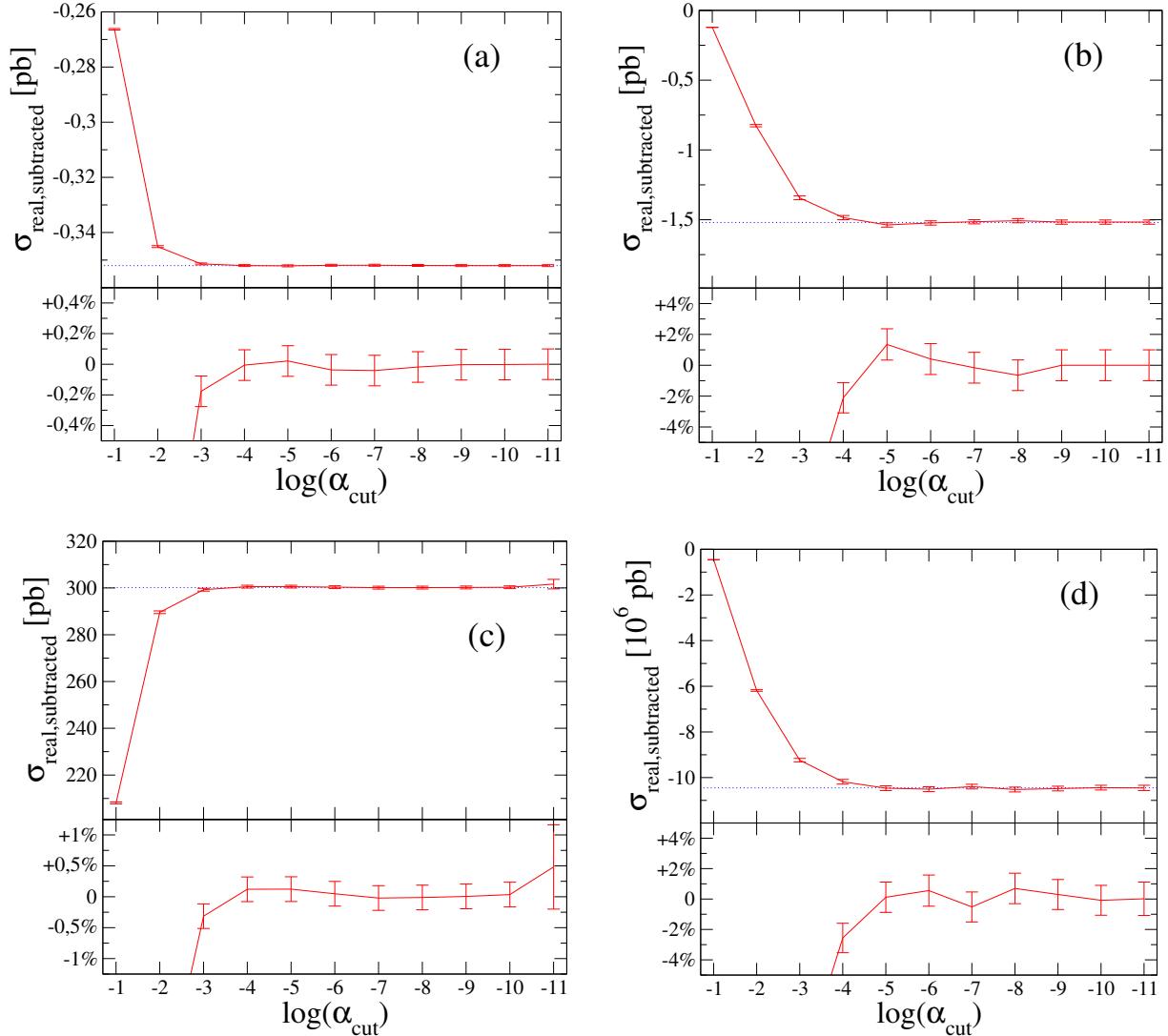
Furthermore integrated results of the virtual and real parts of the NLO corrections in this subtraction scheme were compared and agreed within statistical errors for all accessible processes.

### 5.3.2 Test of convergence for the real ME

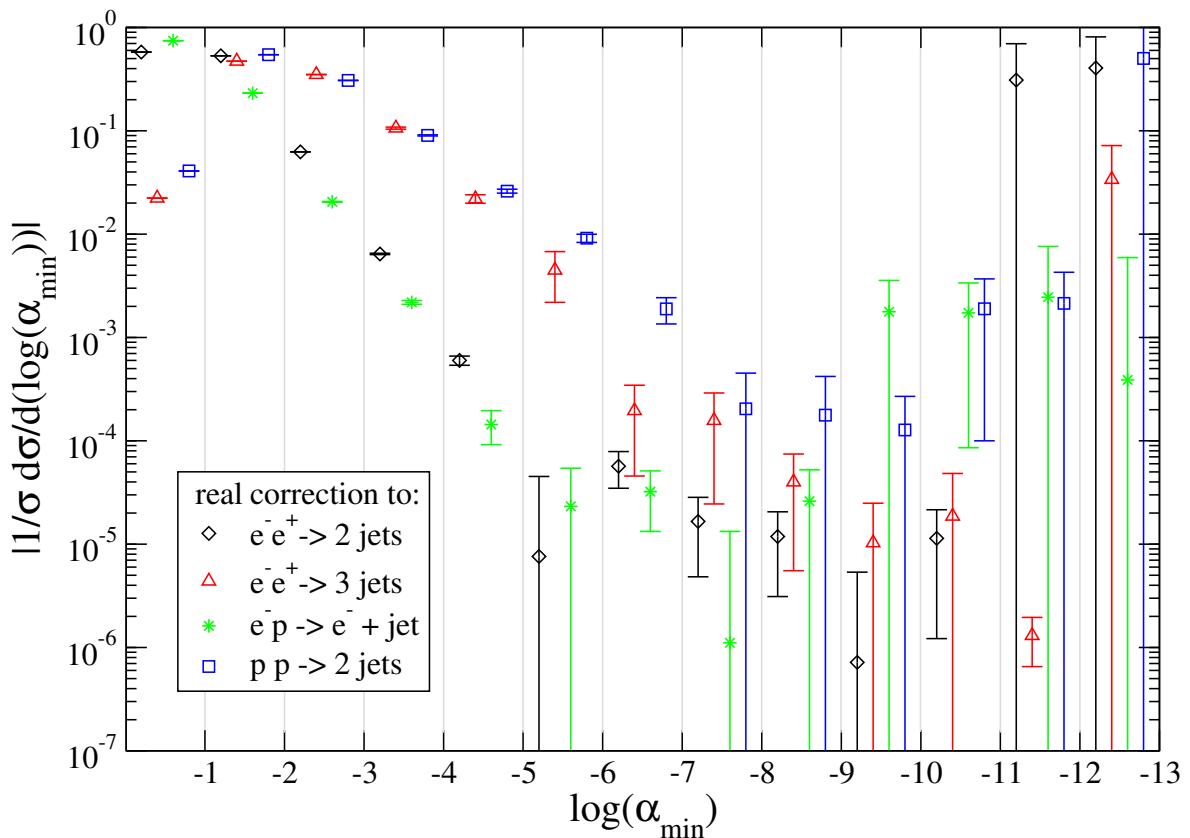
An obvious first technical check of the overall package consists of testing the convergence behaviour of the dipole subtraction terms close to the singular region. To this end, the  $m + 1$ -parton phase space of the regularized real correction part is numerically integrated over. The crucial issue is to ensure that the integrand remains finite over the full phase space, in addition the performance of the integration algorithms deserve some consideration.

Clearly, for the numerical calculation a small phase space region around each singular configuration has to be cut out. Although the dipole terms are expected to become equal to the matrix elements there, technically speaking infinite or very large numbers must be subtracted in this region, leading to large fluctuations and hence to errors due to the limited numerical precision at which the calculation is performed. Therefore a variable  $\alpha_{\min}$  is introduced, which on the basis of kinematic variables of corresponding dipole functions, reads as follows:

$$\alpha_{\min} = \min_{\text{dipoles}} (a_{\text{dipole}}) , \quad (5.73)$$



**Figure 5.3:** Dependence of the subtracted real emission cross section on  $\alpha_{\text{cut}}$  for (a):  $e^-e^+ \rightarrow 2\text{jets}$ ; (b):  $e^-e^+ \rightarrow 3\text{jets}$ , both at a CM energy of 100 GeV; (c):  $e^-p \rightarrow e^- + \text{jet}$  with a 50 GeV electron beam and protons at 500 GeV; (d):  $pp \rightarrow 2\text{jets}$  at a CM energy of 14 TeV. To obtain a well-defined LO cross section for (b) at least two jets with a  $k_{\perp}^{\text{DUR.}} > 10$  GeV, for (c) a transverse energy of the scattered  $e^- > 10$  GeV and for (d) at least two jets with  $p_{\perp} > 40$  GeV are required.



**Figure 5.4:** Normalized absolute values of cross sections in bins of  $\alpha_{\min}$ . Setups and phase space cuts are the same as in Fig. 5.3.

where

$$a_{\text{dipole}} = \begin{cases} y_{ij,k} & \text{for } \mathcal{D}_{ij,k} - \text{dipoles (final state emitter, final state spectator)} \\ 1 - x_{ij,a} & \text{for } \mathcal{D}_{ij}^a - \text{dipoles (final state emitter, initial state spectator)} \\ u_i & \text{for } \mathcal{D}_k^{ai} - \text{dipoles (initial state emitter, final state spectator)} \\ \tilde{v}_i & \text{for } \mathcal{D}^{ai,b} - \text{dipoles (initial state emitter, initial state spectator)} \end{cases} \quad (5.74)$$

This parameter  $\alpha$  serves as a cut-off in such a way that for an externally given parameter  $\alpha_{\text{cut}}$  kinematic configurations with  $\alpha_{\text{min}} < \alpha_{\text{cut}}$  are omitted.

In Fig. 5.3 the dependence of the subtracted cross section on  $\alpha_{\text{cut}}$  for four sets of real correction processes, namely  $e^-e^+ \rightarrow 3\text{jets}$ ,  $e^-e^+ \rightarrow 4\text{jets}$ ,  $e^-p \rightarrow e^- + 2\text{jets}$  and  $pp \rightarrow 3\text{jets}$ . All types of dipoles and splitting functions contribute to the dipole terms which are necessary to regularize those processes. It is apparent that for  $\alpha_{\text{cut}} \sim 10^{-5}$  the cross section stabilizes close to its final value.

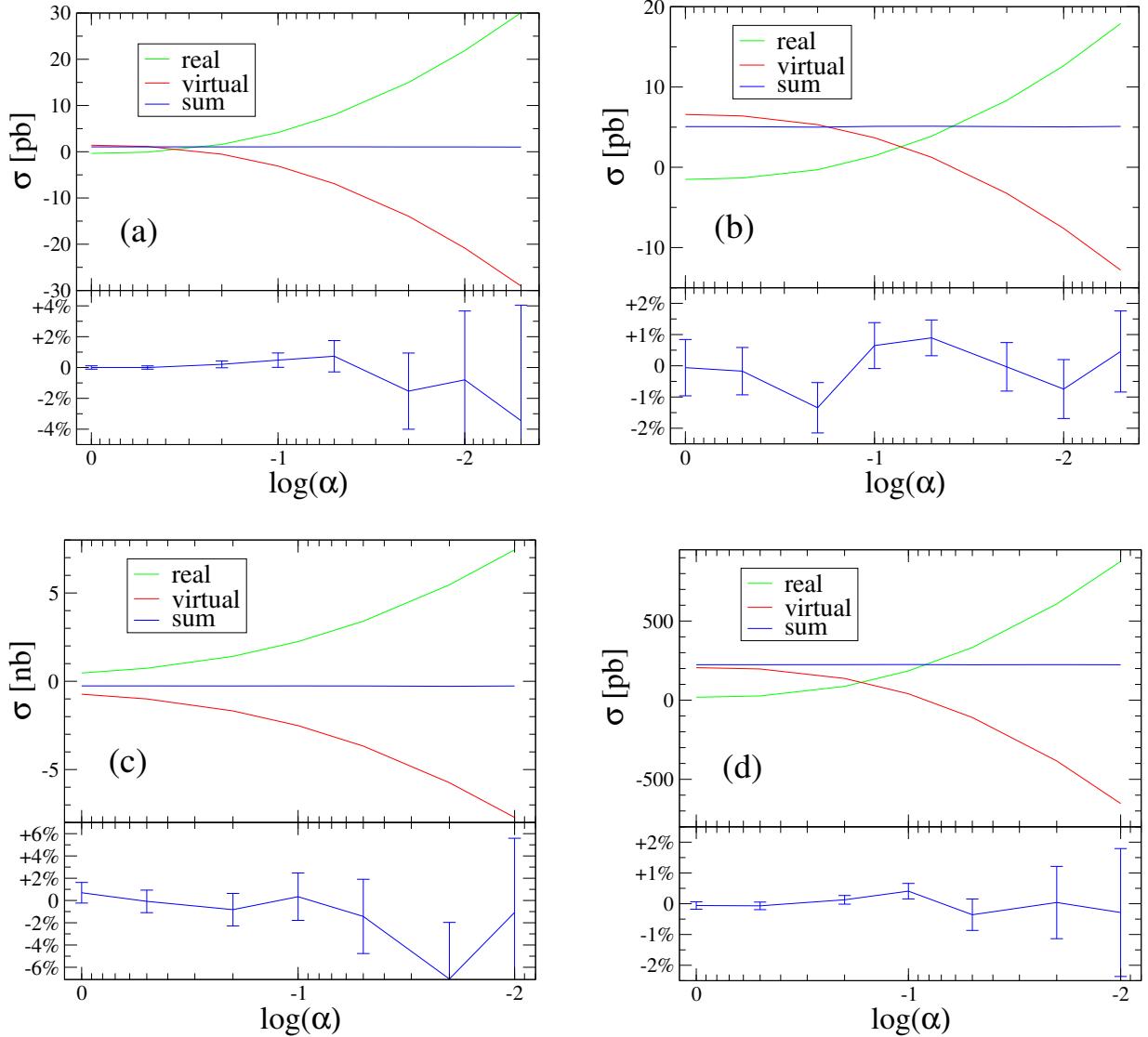
To study the numerical behaviour near the singularity in more detail, in Fig. 5.4 the absolute value of the subtracted cross section, binned in intervals of  $\alpha_{\text{min}}$  is depicted. For all studied processes the contribution to the cross section drops down by at least four orders of magnitude with decreasing  $\alpha_{\text{min}}$  and confirms the observations for full subtracted cross sections made before.

The strong increase accompanied with statistical errors of 100% or larger for  $\alpha_{\text{min}}$  values below  $10^{-9} - 10^{-11}$  signals defects due to the limited numerical precision (*double* precision  $\sim 10^{-12}$ ). One reason is the already mentioned numerical problem when subtracting extreme large and almost equal numbers. Another reason is the precision of the momentum four-vectors itself, because the precision of the external particles residing on their  $m = 0$  mass shell is also limited by the numerical precision. This of course may consequently lead to errors of that order in the matrix element calculation. Thus, Fig. 5.4 allows to determine best choices for  $\alpha_{\text{cut}}$ , somewhere between  $10^{-9}$  and  $10^{-11}$ .

### 5.3.3 Consistency checks with free parameters

In section 5.1.6 ways of modifying the subtraction terms without changing the singular behaviour have been discussed. Such modifications can be employed for non-trivial tests of the implementation, since the modifications will affect both, the real part and the virtual part of the NLO cross sections, with their sum remaining constant.

In Fig. 5.5 the total NLO correction for the cross sections of  $e^-e^+ \rightarrow 2\text{jets}$ ,  $e^-e^+ \rightarrow 3\text{jets}$ ,  $e^-p \rightarrow e^- + \text{jet}$  and  $p\bar{p} \rightarrow W^- \rightarrow e^-\bar{\nu}_e$  and their real and virtual contributions are displayed as functions of the parameter  $\alpha$ , as introduced in section 5.1.6. The fact that the sum remains constant within statistical errors provides a non-trivial confirmation of the correct implementation of the algorithm. It should be noted here that the calculation of the cross



**Figure 5.5:** NLO corrections as a function of the parameter  $\alpha$  in the definition of the subtraction terms (see Eq. (5.57)) for the total cross sections of (a)  $e^-e^+ \rightarrow 2jets$ , (b)  $e^-e^+ \rightarrow 3jets$ , (c)  $e^-p \rightarrow e^- + jet$  and (d)  $p\bar{p} \rightarrow W^- \rightarrow e^-\bar{\nu}_e$ . The results and error bars in the difference plots are determined after calculating 500000 phase space points of the real contribution, which typically dominates the total statistical error.

section of the processes under consideration invokes all types of dipole functions as well as the most general case of the insertion operators from the integrated dipole terms.

By using the same number of phase space points for each integral and comparing statistical errors, it can be seen that this parameter can also be used to optimize the numerical behaviour. Clearly, best results are obtained if the values of the virtual and real contributions are both as small as possible, thus reducing the size of the fluctuations. It should be noted that the error bars in Fig. 5.5 are given not including the leading order part of the cross section. Since relative errors for the latter can be expected to be much smaller if evaluated for the same number of phase space points, the (relative) statistical error for the full NLO cross section will be significantly reduced.

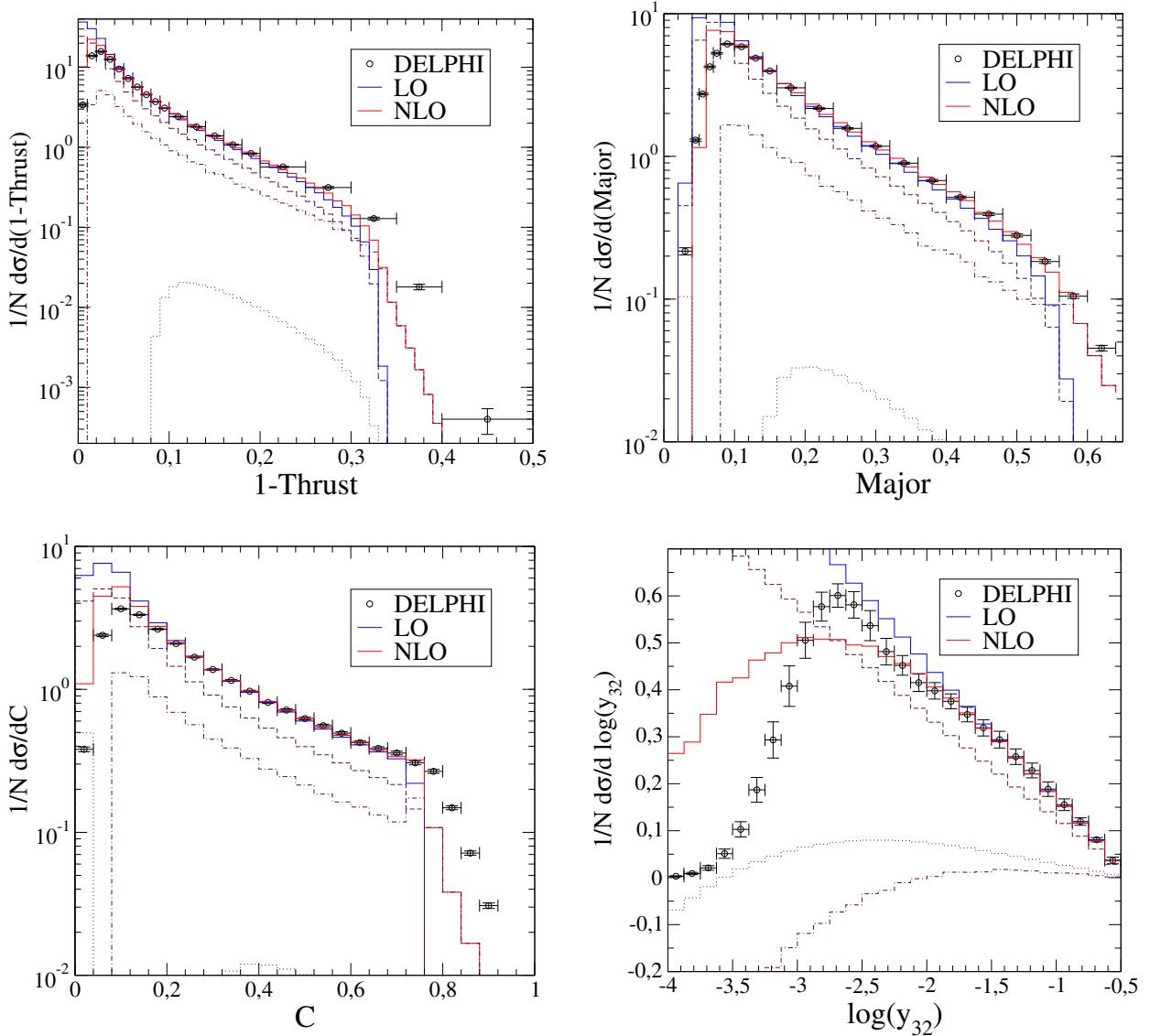
## 5.4 First physical applications

In this section, some simple applications will demonstrate the performance of the dipole subtraction procedure, as implemented, for the calculation of physically relevant observables. The born matrix elements, dipole subtraction terms to regularize the real correction and corresponding finite terms to be added to the virtual correction were generated automatically by AMEGIC++. The one-loop amplitudes have been explicitly implemented for the considered processes.

### 5.4.1 Three-jet observables at LEP

To compute three jet cross section at next-to-leading order the one-loop matrix element given in [97] has been implemented. The expression given there is averaged over the direction of incoming momenta, which is sufficient for observables that are not correlated to the beam direction.

In Fig. 5.6 LO and NLO predictions are displayed for observables sensitive to  $\mathcal{O}(\alpha_S)$ . In particular, the event shape observables 1-Thrust, Major, C-parameter and the Durham  $3 \rightarrow 2$  jet rate are compared with measurements performed at LEP on the  $Z^0$ -peak by DELPHI [105]. All data are normalized to unity. The normalization for the calculated cross section, however, is somewhat complicated. This is because in the calculation three-jet events are required in each case, translating into the necessity to apply a phase space cut. On the other hand, the data are more inclusive and also include comparably soft regions, where fixed-order perturbation theory is known to fail and must be supported by resummation techniques. The normalization for the calculations has thus be chosen such that it agrees with data in the “safe” regions. This exposes the differences between LO and NLO calculations best. As a consequence, the corresponding normalization factor of



**Figure 5.6:** The event shape observables 1-Thrust, Major, C-parameter and the Durham  $3 \rightarrow 2$  jet rate at LEP I compared to measurements by DELPHI [105]. The LO and the NLO predictions have been normalized to the data separately in a region, where agreement can be expected. The dashed, dashed-dotted and the dotted lines are the Born, real and virtual contribution to the NLO cross sections, respectively.

both calculations is not identical. From the results of Fig. 5.6 it can be deduced that for all observables the range described sufficiently well by the calculation is extended for the NLO calculations. For both, the region described by soft physics (left side in all plots) as well as phase space regions populated by additional hard QCD radiation (right side in all event shape plots) the prediction has been improved.

### 5.4.2 DIS: $e^- p \rightarrow e^- + jet$

The one loop matrix element for this process is given by the well known expression

$$|M|_{(1-loop)}^2 = |M|_{(born)}^2 \frac{C_F \alpha_S}{2\pi} \frac{1}{\Gamma(1-\epsilon)} \left( \frac{4\pi\mu^2}{Q^2} \right)^\epsilon \left\{ -\frac{2}{\epsilon^2} - \frac{3}{\epsilon} - 8 + \mathcal{O}(\epsilon) \right\}, \quad (5.75)$$

where  $Q^2 = -q^2 > 0$  with  $q$  the momentum transfer between the electron and the proton.

Fig. 5.7 shows differential cross sections w.r.t. the transverse momentum and rapidity of the scattered electron and the hardest jet at leading and at next-to-leading order. The CM-energy has been taken as  $\sqrt{10^5}$  GeV, corresponding to a 50 GeV electron beam and a 500 GeV proton beam. The parton distribution function CTEQ6M [83] has been employed, factorization and renormalization scales have been fixed to  $Q^2$ . A phase space cut on the electron of  $p_T > 10$  GeV has been imposed. The NLO correction for this setup is comparably small, for the total cross section it is of the order of 5% and negative. The ratios of NLO and LO calculation, however, are not constant for all observables. At NLO the cross section rises for increasing momentum transfer, up to a correction of 40% for transverse momenta of electron and jet of the order of 150 GeV.

### 5.4.3 $W^-$ production at Tevatron

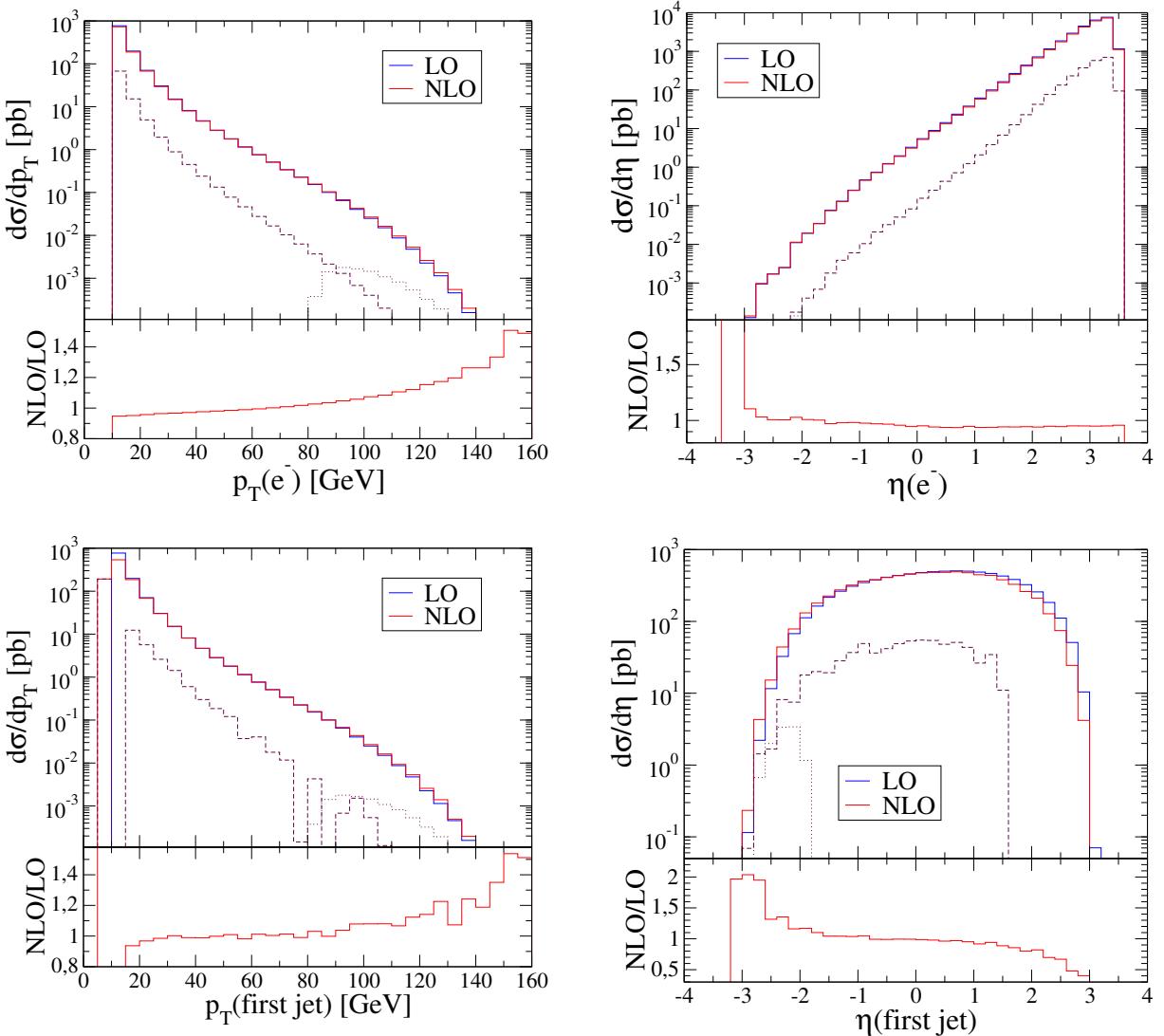
The one-loop virtual contribution to this process can be obtain by crossing relations from Eq. (5.75) and is given by

$$|M|_{(1-loop)}^2 = |M|_{(born)}^2 \frac{C_F \alpha_S}{2\pi} \frac{1}{\Gamma(1-\epsilon)} \left( \frac{4\pi\mu^2}{Q^2} \right)^\epsilon \left\{ -\frac{2}{\epsilon^2} - \frac{3}{\epsilon} - 8 + \pi^2 + \mathcal{O}(\epsilon) \right\}, \quad (5.76)$$

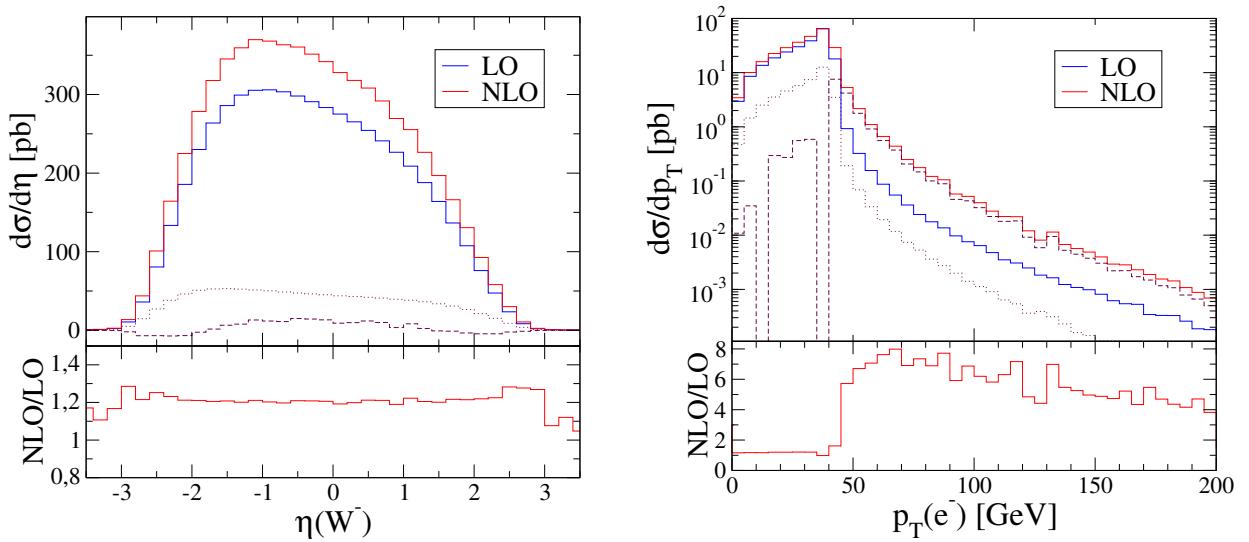
where now  $Q^2 = \hat{s}$ , the CM energy squared of the incoming partons.

Fig. 5.8 shows cross sections for Tevatron Run II, differential in the rapidity of the  $W^-$ -boson and the transverse momentum of the electron, respectively. The parton distribution function CTEQ6M [83] has been employed, factorization and renormalization scales have been fixed to  $m_W^2$ .

The total and differential cross sections are in full agreement with predictions obtained using the next-to-leading order parton level generator MCFM [25].



**Figure 5.7:** Distribution of transverse momentum (left plots) and rapidity, defined in the beam CM frame (right plots), of the scattered electron and of the hardest jet in deep inelastic scattering, calculated at leading order and next-to-leading order. The CM-energy has been taken as  $\sqrt{10^5}$  GeV. A phase space cut on the electron ( $p_T > 10$  GeV) has been applied. For the rapidity distribution of the first jet a  $p_T > 15$  GeV has been required. Dashed and dotted lines denote the real and the virtual corrections to the Born cross section, respectively. The lower panels of each plot show the ratio between the leading order and the next-to-leading order results.



**Figure 5.8:** Rapidity distribution of the  $W^-$ -boson (left plot) and transverse momentum of the electron for the process  $p\bar{p} \rightarrow W^- \rightarrow e^-\bar{\nu}_e$  at Tevatron Run II, calculated at leading order and next-to-leading order. Dashed and dotted lines denote the real and the virtual corrections to the Born cross section, respectively. The lower panels of each plot show the ratio between the leading order and the next-to-leading order results.

## 5.5 Conclusions and outlook

In this chapter a fully automated implementation of the Catani-Seymour dipole formalism in the framework of the matrix element generator AMEGIC++ has been presented. It allows to automatically generate the process-dependent real correction terms for given Born cross sections with massless external particles and the corresponding real subtraction terms. The integration of the subtracted real correction terms is performed automatically with a multi-channel method, giving rise to an appreciable convergence. The implementation has carefully been checked for correctness, invoking consistency checks with free finite terms which may be added to the subtraction terms. Through the explicit inclusion of virtual terms a next-to-leading order parton-level calculator is so available.

In the future, the code will be further updated to include massive external particles and to provide a full parton-level generator at NLO.



# 6 A novel method to evaluate scalar 1-loop integrals

In the previous chapter the automation of the Catani-Seymour dipole subtraction method has been reported, which generates any part of a next-to-leading order calculation except for the virtual correction, which remains a major bottleneck.

For the treatment of virtual corrections, the integral over the loop momentum has to be performed in  $d = 4 - 2\epsilon$  dimensions. Towards analytic approaches to solve such integrals, pioneering work has been performed long time ago by t’Hooft and Veltman [106], providing explicit solutions of basic scalar 1-point, 2-point, 3-point and 4-point one-loop integrals. Together with the Passarino-Veltman tensor reduction scheme [88] this allows to determine analytic solutions for  $2 \rightarrow 2$  scattering processes at non-exceptional phase space configurations.

For  $n$ -point integrals with  $n > 4$  a number of tensor reductions schemes have been elaborated [107, 108, 109, 110, 111, 112], that allow to reduce those integrals to sets of 2- to 4-point master integrals. However, the main drawback of all those methods is the appearance Gram- or other kinematical determinants in the denominator, which become singular for certain configurations of external momenta and thus spoil the numerical stability of the solutions. Another, more technical difficulty when using such schemes is in the inflation in the number of remaining finite integrals to be solved for processes with  $n > 5$ . This motivates why so far most of the state-of-the art calculations are constrained to processes involving at worst pentagon (5-point) diagrams.

Because of the complications to generalize solutions towards increasing  $n$ , being inherent to all known analytic methods, various numerical or semi-numerical methods have been developed recently for the calculation of  $n$ -point one-loop functions. In the approach of [113], which has been elaborated for one-loop integrals with up to six external legs, the Feynman-parameter integrals are rewritten in such a way, that they can be numerically integrated in a stable way. A semi-numerical approach that relies on the subtraction diagram by diagram of UV and infrared divergences has been advocated in [114]. A different semi-

numerical method express one-loop integrals in terms of one- and two-dimensional parameter integrals which are suitable for numerical integration [115]. A fully numerical approach to calculate loop integrals by contour integration has been proposed in [116]. In another numerical method integrals in Feynman-parameter representation have been numerically performed with a small but finite " $i\varepsilon$ " from the propagator denominators and a subsequent extrapolation  $\varepsilon \rightarrow 0$  [117]. A formalism to directly extract the UV and infrared divergences from the one loop graph and express them in the basis set of divergent  $d$  dimensional triangle graphs, and use of recursion relations has been constructed in [111]. A numerical method to evaluate loop integrals from their Mellin-Barnes integral representation has been presented in [118]. Further approaches can be found in [119, 120, 121, 122]. In most cases the  $n$ -point functions are evaluated by using different tricks to deal with the corresponding Feynman-parameter representation of the loop integral, in such a way, that after subtracting its divergences it becomes numerically stable and can be integrated numerically in  $d = 4$  space-time dimensions. The dimension of the Feynman parameter space, however, grows linearly with the multiplicity of the loop integral. Very recently there has also been some development on numerical methods for loop integrals, based on the the relation of scattering amplitudes to the Twistor space, found by Witten [66]. Examples for such approaches are given in [123, 124].

In this chapter a new semi-numerical method for the evaluation of scalar one-loop amplitudes is explored, that relates the loop integral to a number phase-space like integrals. The main advantages of this method are the obvious singularity structure of those integrals and that, after the extraction of divergencies, the remaining finite integrals can be directly evaluated using well-established Monte-Carlo methods. Concerning the number of external legs all steps taken are completely general. In section 6.1 the general idea for the underlying decomposition is presented and the generated integrals are classified. Potential analytic solutions for those integrals are investigated in section 6.2 and section 6.3 is dedicated to (semi-)numerical solution strategies.

## 6.1 Integral duality: mapping one-loop integrals onto phase-space integrals

A general one-loop scalar  $n$ -point function with massless internal lines in dimensional regularization is considered:

$$L^{(n)} = -i \mu^{4-d} \int \frac{d^d q}{(2\pi)^d} \prod_{i=1}^n \frac{1}{(q + k_i)^2 + i0} , \quad (6.1)$$

with  $q^\mu$  being the loop momentum and  $k_i = \sum_{j=1}^{i-1} p_j$  is a sum over four-momenta of the external legs, denoted by  $p_i^\mu$  ( $i = 1, \dots, n$ ). They carry external masses  $p_i^2 = m_i^2$ , however a

subset of those momenta can be massless. By definition, all external momenta are considered to be outgoing, so that  $p_1^\mu + \dots + p_n^\mu = 0$ .

In Eq. (6.1)  $d = 4 - 2\epsilon$  denotes the number of space-time dimensions, regularizing simultaneously ultraviolet and infrared divergences. For the following procedure, however, only ultraviolet finite one-loop integrals are considered. This constrains the method to loop integrals with  $n \geq 4$  legs (boxes), i.e. it is not appropriate for two- and three-point functions. Those however are simple enough to be evaluated in the standard way and will not be considered in this work.

The first step in the evaluation is to close the integration contour on the lower half plane, and to apply the residue theorem. This corresponds to the replacement

$$\prod_{i=1}^n \frac{1}{(q + k_i)^2 + i0} \rightarrow \sum_{i=1}^n \left[ -2\pi i \delta_+((q + k_i)^2) \right] \prod_{j=1, j \neq i}^n \frac{1}{(q + k_j)^2}. \quad (6.2)$$

Furthermore a shift in the four-momentum is performed:  $q^\mu \rightarrow (q - k_i)^\mu$ , for each term of the sum independently,

$$L^{(n)}(p_1, \dots, p_n) = - \sum_{i=1}^n I^{(n-1)}(-p_i, p_i + p_{i+1}, \dots, \sum_{j=i}^{i+n-2} p_j), \quad (6.3)$$

where

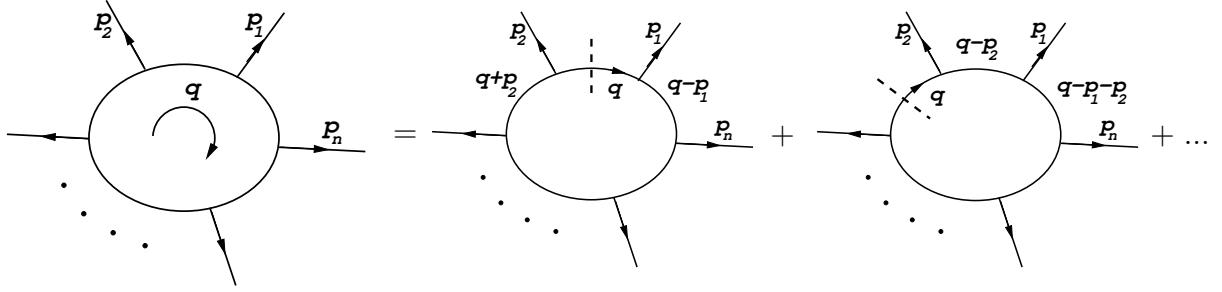
$$I^{(n)}(k_1, \dots, k_n) = \mu^{4-d} \int \frac{d^d q}{(2\pi^{d-1})} \delta_+(q^2) \left[ \prod_{j=1}^n \frac{1}{2q \cdot k_j + k_j^2} \right], \quad (6.4)$$

and momenta  $p_i$  with  $i > n$  are understood as  $p_{n+i} \equiv p_i$ . The resulting integrals  $I^{(n)}$  are dubbed *dual integrals*.

Following this procedure, which is illustrated in Fig. 6.1, amounts to setting subsequently each internal line on-shell, and finally substituting the loop integral by a sum of integrals that resemble bremsstrahlung or phase space integrals, with momentum integration however not being constrained by momentum conservation. The resulting integrals may still exhibit infrared divergences. One encounters two different classes of infrared singular bremsstrahlung integrals. In analogy with the integration over the phase space of a real emission they will be called collinear and soft, respectively. Dual integrals are either soft or collinear divergent or infrared finite, depending on whether the external legs connected by the internal line that is set on-shell are massive or massless.

Integrals are defined as soft when the momenta of the two external legs adjacent to the internal on-shell line are both light-like. Soft integrals have the form

$$I_s^{(n)}(p, \bar{p}; \{k_i, s_i\}) \equiv \mu^{4-d} \int \frac{d^d q}{(2\pi)^{d-1}} \delta_+(q^2) \frac{p \cdot \bar{p}}{p \cdot q \bar{p} \cdot q} \prod_{i=1}^n \left[ \frac{1}{2k_i \cdot q + s_i} \right], \quad (6.5)$$



**Figure 6.1:** Decomposition of a loop integral into a sum of dual integrals. The dashed lines in the right hand side correspond to the  $\delta_+(q^2)$  function in Eq. (6.4) and thus amount to setting the corresponding propagator on shell. The sum is taken over all propagators in the loop.

with  $p$  and  $\bar{p}$  the light-like momenta ( $p^2 = \bar{p}^2 = 0$ ),  $s_i \neq 0$ . Ultraviolet finiteness restricts the integrals  $I_s^{(n)}$  to  $n \geq 1$ . Infrared singularities of soft integrals arise as double and single poles in the  $\epsilon \rightarrow 0$  expansion. Collinear integrals appear when only one of the two adjacent external legs is light-like and the other is either time-like or space-like. Collinear integrals are given through

$$I_c^{(n)}(p; \{k_i, s_i\}) \equiv \mu^{4-d} \int \frac{d^d q}{(2\pi)^{d-1}} \delta_+(q^2) \frac{1}{p \cdot q} \prod_{i=1}^n \left[ \frac{1}{2k_i \cdot q + s_i} \right], \quad (6.6)$$

where  $p^2 = 0$ , with the restriction of  $n \geq 2$  because of UV finiteness. They exhibit single infrared poles at most. Finally, cuts over internal lines surrounded by two massive external legs are infrared finite. Finite integrals are given by

$$I_f^{(n)}(\{k_i, s_i\}) \equiv \mu^{4-d} \int \frac{d^d q}{(2\pi)^{d-1}} \delta_+(q^2) \prod_{i=1}^n \left[ \frac{1}{2k_i \cdot q + s_i} \right]. \quad (6.7)$$

Following this approach any one-loop  $n$ -point function with  $n \geq 4$ , can be decomposed into a linear combination of soft, collinear and finite phase-space-like integrals with  $(n-1)$ -propagators:

$$L^{(n)} = - \sum_{i=1}^n I^{(n-1)}(\{-p_i, m_i^2\}, \{p_{i+1}, m_{i+1}^2\}, \{k_j, s_j = k_j^2\}), \quad (6.8)$$

where  $k_j = p_{i+1} + \dots + p_{i+1+j}$  with  $j = 1, \dots, n-2$  and again  $p_i$  with  $i > n$  are understood

as  $p_{n+i} \equiv p_i$ . In particular,

$$\begin{aligned}
 I^{(n-1)} &= -\frac{1}{2s_{i,i+1}} I_s^{(n-3)}(p_i, p_{i+1}; \{k_j, s_j\}) \quad \text{if} \quad m_i = m_{i+1} = 0, \\
 I^{(n-1)} &= -\frac{1}{2} I_c^{(n-2)}(p_i; \{p_{i+1}, m_{i+1}^2\}, \{k_j, s_j\}), \quad \text{if} \quad m_i = 0, \\
 I^{(n-1)} &= \frac{1}{2} I_c^{(n-2)}(p_{i+1}; \{-p_i, m_i^2\}, \{k_j, s_j\}), \quad \text{if} \quad m_{i+1} = 0, \\
 I_f^{(n-1)} &= I_f^{(n-1)}(\{k_j, s_j\}), \quad \text{otherwise.}
 \end{aligned} \tag{6.9}$$

The advantages of this approach are evident when the number of external legs is large. Methods based on Feynman parameterizations might become soon quite cumbersome. This approach singles out directly the infrared divergences of the loop integral, and therefore is quite appropriate for a systematic calculation of multi-leg scattering amplitudes. Furthermore, since it establishes a correspondence with phase-space integration, Monte Carlo methods can be used for numerical evaluations of the finite contributions.

## 6.2 Analytic approach to dual integrals

The integrals, defined in Eqs. (6.5)-(6.7), are well-defined if the denominators in the product over propagators, not adjacent to the internal on-shell line, are positive definite in the whole phase space of the momentum  $q$ . This is satisfied if all kinematic parameters  $k_i$  yield  $k_i \cdot q > 0$  (i.e. the energy component  $k_{i0} > 0$ ) and  $s_i = k_i^2 > 0$ . The following results will be obtained, restricting to this *safe* region only. It should be noticed, however, that the decomposition of any physical one-loop integral always leads to dual integrals in the safe region as well as to integrals beyond this region. In the latter case additional singularities may appear in the integrand, when one of denominators  $2k_i \cdot q + s_i$  becomes zero for a certain momentum configuration. Unlike the soft and collinear singularities this, however, does not lead to a divergence in the four dimensional limit. Solutions for such cases must be obtained by a proper analytic continuation from the safe region. This is beyond the scope of this work and will be addressed elsewhere [125].

### 6.2.1 Soft integrals

It is convenient to work in the light-cone coordinate system where all four-momenta, e.g.  $k^\mu$ , are written in terms of their plus and minus components,  $k_\pm = (k^0 \pm k^3)/\sqrt{2}$ , and transverse momentum,  $k_\perp^\mu$ , in  $d-2$  space dimensions. The scalar product of two four-momenta  $p^\mu$  and  $k^\mu$  is given in that system by

$$p \cdot k = p_+ k_- + p_- k_+ - p_\perp \cdot k_\perp. \tag{6.10}$$

The soft integral in Eq. (6.5) is considered in the frame where the light-like external momenta  $p$  and  $\bar{p}$  are directed along the plus and minus light-cone directions:  $p = (p_+, 0, \vec{0})$ , and  $\bar{p} = (0, p_-, \vec{0})$ , respectively. To simplify the discussion it is assumed that the energy components of  $p$  and  $\bar{p}$  (and thus  $p_+$  and  $p_-$ ) are positive. This is no loss of generality, since the soft integral is invariant under the replacement  $p \rightarrow -p$  or  $\bar{p} \rightarrow -\bar{p}$ . Integrating Eq. (6.5) over  $q_+ = q_\perp^2/(2q_-)$  with the delta function, the soft integral takes the form

$$I_s^{(n)}(p, \bar{p}; \{k_i, s_i\}) = \mu^{4-d} \int_0^\infty \frac{dz}{z} \int \frac{d^{d-2}q_\perp}{(2\pi)^{d-1}} \frac{1}{q_\perp^2} \prod_{i=1}^n \left[ \frac{z_i}{(q_\perp - z_i k_{\perp i})^2 + z_i^2 k_i^2 + z_i s_i} \right], \quad (6.11)$$

where  $z = p_+ q_- = p \cdot q$ , and  $z_i = z/(p \cdot k_i)$  were introduced. Next<sup>1</sup>, a Feynman parameterization is introduced and a shift in the transversal momentum  $q_\perp \rightarrow q_\perp + \sum_{i=1}^n c_i z_i k_{\perp i}$  is performed. The integration over the angular variables is now trivial and casts the integral to

$$\begin{aligned} I_s^{(n)}(p, \bar{p}; \{k_i, s_i\}) &= \frac{\mu^{4-d} \Gamma(n+1)}{\prod_{i=1}^n p \cdot k_i} \int_0^\infty dz z^{n-1} \int_0^1 dx \int_0^1 dc_1 \dots dc_n \delta(1-x-\sum_{j=1}^n c_j) \\ &\times \frac{\Omega_{(d-2)}}{2(2\pi)^{d-1}} \int_0^\infty dq_\perp^2 (q_\perp^2)^{\frac{d-4}{2}} \\ &\left( q_\perp^2 + \sum_{i,j=1}^n c_i c_j z_i z_j k_i \cdot k_j + x \sum_{i=1}^n c_i z_i^2 (k_i^2 + k_{\perp i}^2) + \sum_{i=1}^n c_i z_i s_i \right)^{-(n+1)}, \end{aligned} \quad (6.12)$$

where

$$\Omega_{(d-2)} = \frac{2\pi^{\frac{d-2}{2}}}{\Gamma(\frac{d-2}{2})} \quad (6.13)$$

is the transverse solid angle. The remaining integrations over the transverse and the longitudinal loop-momentum,  $q_\perp^2$  and  $z$ , respectively, are now straightforward, and the soft integral becomes

$$\begin{aligned} I_s^{(n)}(p, \bar{p}; \{k_i, s_i\}) &= -\frac{2\mu^{2\epsilon} \Gamma(1-\epsilon) \Gamma(n+2\epsilon)}{\epsilon (4\pi)^{2-\epsilon} \prod_{i=1}^n p \cdot k_i} \int_0^1 dx \int_0^1 dc_1 \dots dc_n \delta(1-x-\sum_{j=1}^n c_j) \\ &\times \left( \sum_{i,j} \frac{c_i c_j k_i \cdot k_j}{p \cdot k_i p \cdot k_j} + x \sum_i \frac{c_i (k_i^2 + k_{\perp i}^2)}{(p \cdot k_i)^2} \right)^\epsilon \left( \sum_i \frac{c_i s_i}{p \cdot k_i} \right)^{-n-2\epsilon}, \end{aligned} \quad (6.14)$$

where

$$k_i^2 + k_{\perp i}^2 = \frac{2p \cdot k_i \bar{p} \cdot k_i}{p \cdot \bar{p}}. \quad (6.15)$$

<sup>1</sup>Relations, necessary to perform the following steps explicitly are listed in Appendix C.

Then, a further change of variables  $c_i = (1 - x)\lambda_i$  is performed, leading to

$$\begin{aligned} I_s^{(n)}(p, \bar{p}; \{k_i, s_i\}) &= -\frac{2\mu^{2\epsilon}\Gamma(1-\epsilon)\Gamma(n+2\epsilon)}{\epsilon(4\pi)^{2-\epsilon}\prod_{i=1}^n p \cdot k_i} \int_0^1 d\lambda_1 \dots d\lambda_n \delta(1 - \sum_{j=1}^n \lambda_j) \\ &\times \int_0^1 dx (1-x)^{-1-\epsilon} (a + b x)^\epsilon c^{-n-2\epsilon}, \end{aligned} \quad (6.16)$$

where

$$\begin{aligned} a &= \sum_{i,j=1}^n \lambda_i \lambda_j \frac{k_i \cdot k_j}{p \cdot k_i p \cdot k_j}, \\ b &= \frac{2}{p \cdot \bar{p}} \sum_{i=1}^n \lambda_i \frac{\bar{p} \cdot k_i}{p \cdot k_i} - a, \\ c &= \sum_{i=1}^n \lambda_i \frac{s_i}{p \cdot k_i}. \end{aligned} \quad (6.17)$$

Integrating over  $x$ , one finally gets

$$\begin{aligned} I_s^{(n)}(p, \bar{p}; \{k_i, s_i\}) &= \frac{2\mu^{2\epsilon}\Gamma(1-\epsilon)\Gamma(n+2\epsilon)}{\epsilon^2(4\pi)^{2-\epsilon}\prod_{i=1}^n p \cdot k_i} \int_0^1 d\lambda_1 \dots d\lambda_n \delta(1 - \sum_{j=1}^n \lambda_j) \\ &\times a^\epsilon c^{-n-2\epsilon} {}_2F_1\left(1, -\epsilon; 1 - \epsilon; -\frac{b}{a}\right). \end{aligned} \quad (6.18)$$

For soft dual integrals obtained by the decomposition of a box diagram (i.e. for  $n = 1$ ), the solution can be read directly from Eq. (6.18) without further integration. It is given by

$$I_s^{(n=1)}(p, \bar{p}; k, s) = \frac{2\Gamma(1-\epsilon)\Gamma(1+2\epsilon)}{\epsilon^2(4\pi)^{2-\epsilon}s} \left(\frac{\mu^2 k^2}{s^2}\right)^\epsilon {}_2F_1\left(1, -\epsilon; 1 - \epsilon; -\frac{\mathbf{k}_\perp^2}{k^2}\right), \quad (6.19)$$

where the Hypergeometric function has the following expansion

$${}_2F_1(1, -\epsilon, 1 - \epsilon, z) = 1 + \epsilon \ln(1 - z) - \epsilon^2 \text{Li}_2(z) + \mathcal{O}(\epsilon^3). \quad (6.20)$$

### 6.2.2 Collinear integrals

The external light-like momentum  $p^\mu$  in Eq. (6.6) is set along the plus direction,  $p = (p_+, 0, \vec{0})$ . Again it is assumed that the energy component of  $p$  and thus  $p_+$  are positive. This is no loss of generality because of the identity

$$I_c^{(n)}(p; \{k_i, s_i\}) \equiv -I_c^{(n)}(-p; \{k_i, s_i\}). \quad (6.21)$$

The integration over  $q_+ = q_\perp^2/(2q_-)$  is performed with the delta function. The collinear integral then takes the form

$$I_c^{(n)}(p; \{k_i, s_i\}) = \mu^{4-d} \int_0^\infty \frac{dz}{2z^2} \int \frac{d^{d-2}q_\perp}{(2\pi)^{d-1}} \prod_{i=1}^n \left[ \frac{z_i}{(q_\perp - z_i k_\perp)^2 + z_i^2 k_i^2 + z_i s_i} \right], \quad (6.22)$$

using the definitions  $z = p_+ q_- = p \cdot q$ , and  $z_i = z/(p \cdot k_i)$ . Similar to the case of soft integrals, a Feynman parameterization is introduced in the transverse space to integrate over the transverse components of the loop-momentum. After shifting the transverse loop-momentum  $q_\perp \rightarrow q_\perp + \sum_{i=1}^n c_i z_i k_{\perp i}$ , the integration over the angular variables becomes trivial, and the collinear integral thus reads

$$\begin{aligned} I_c^{(n)}(p; \{k_i, s_i\}) &= \frac{\mu^{4-d} \Gamma(n)}{2 \prod_{i=1}^n p \cdot k_i} \int_0^\infty dz z^{n-2} \int_0^1 dc_1 \dots dc_n \delta(1 - \sum_{j=1}^n c_j) \\ &\times \frac{\Omega_{(d-2)}}{2(2\pi)^{d-1}} \int_0^\infty dq_\perp^2 (q_\perp^2)^{\frac{d-4}{2}} \left( q_\perp^2 + \sum_{i,j=1}^n c_i c_j z_i z_j k_i \cdot k_j + \sum_{i=1}^n c_i z_i s_i \right)^{-n}. \end{aligned} \quad (6.23)$$

After integrating over the transverse and the longitudinal loop-momentum,  $q_\perp^2$  and  $z$  respectively, the collinear integral is given by

$$\begin{aligned} I_c^{(n)}(p; \{k_i, s_i\}) &= -\frac{\mu^{2\epsilon} \Gamma(1-\epsilon) \Gamma(n-1+2\epsilon)}{\epsilon (4\pi)^{2-\epsilon} \prod_{i=1}^n p \cdot k_i} \\ &\times \int_0^1 dc_1 \dots dc_n \delta(1 - \sum_{j=1}^n c_j) \left( \sum_{i,j=1}^n \frac{c_i c_j k_i \cdot k_j}{p \cdot k_i p \cdot k_j} \right)^\epsilon \left( \sum_{i=1}^n \frac{c_i s_i}{p \cdot k_i} \right)^{1-n-2\epsilon} \end{aligned} \quad (6.24)$$

For integrals with  $n = 2$  non-singular propagators that further satisfy  $(k_1 - k_2)^2 = 0$ , the specific solution is given by

$$\begin{aligned} I_c^{(2)}(p; k_1, s_1 = k_1^2, k_2, s_2 = k_2^2) &= \frac{c_\Gamma}{\epsilon^2 p \cdot (k_2 s_1 - k_1 s_2)} \\ &\times \left[ (-s_1)^{-\epsilon} {}_2F_1 \left( 1, -\epsilon; 1 - \epsilon; \frac{s_1 p \cdot (k_2 - k_1)}{p \cdot (k_2 s_1 - k_1 s_2)} \right) \right. \\ &\left. - (-s_2)^{-\epsilon} {}_2F_1 \left( 1, -\epsilon; 1 - \epsilon; \frac{s_2 p \cdot (k_2 - k_1)}{p \cdot (k_2 s_1 - k_1 s_2)} \right) \right] \end{aligned} \quad (6.25)$$

### 6.2.3 Finite integrals

For finite bremsstrahlung integrals, the light cone coordinate system can be defined by an auxiliary light-like vector ( $p^2 = 0$ ,  $p_+ > 0$ ) which is used to fix the transverse plane. Of course, any choice should lead to the same result. Performing the integration over the delta function yields

$$I_f^{(n)}(\{k_i, s_i\}) = \mu^{4-d} \int_0^\infty \frac{dz}{2z} \int \frac{d^{d-2} q_\perp}{(2\pi)^{d-1}} \prod_{i=1}^n \left[ \frac{z_i}{(q_\perp - z_i k_{\perp i})^2 + z_i^2 k_i^2 + z_i s_i} \right]. \quad (6.26)$$

Now running the same steps as for the soft and collinear integrals, introducing a Feynman parameterization and integrating over  $q_\perp^2$  and  $z$ , leads to

$$\begin{aligned} I_f^{(n)}(\{k_i, s_i\}) &= \frac{\mu^{2\epsilon} \Gamma(1 - \epsilon) \Gamma(n - 2 + 2\epsilon)}{(4\pi)^{2-\epsilon} \prod_{i=1}^n p \cdot k_i} \\ &\times \int_0^1 dc_1 \dots dc_n \delta(1 - \sum_{j=1}^n c_j) \left( \sum_{i,j=1}^n \frac{c_i c_j k_i \cdot k_j}{p \cdot k_i p \cdot k_j} \right)^{-1+\epsilon} \left( \sum_{i=1}^n \frac{c_i s_i}{p \cdot k_i} \right)^{2-n-2\epsilon} \end{aligned} \quad (6.27)$$

To obtain an explicit solution, even the simplest case  $n = 3$  would imply at least two nontrivial integrations. This will not be further considered since finite integrals are directly accessible with numeric methods.

## 6.3 Numeric approaches

From the expressions obtained in the previous section it is clear that it will be relatively hard to obtain fully analytic solutions for dual integrals, for all but the simplest cases. This section is dedicated to study numeric approaches instead. Finite dual integrals are directly suitable for a numeric integration in four space-time dimensions. For soft and collinear integrals, however, the corresponding divergences have to be extracted first. Below two ways are proposed that allow to perform this task for an arbitrary number of external legs.

### 6.3.1 Construction of subtraction terms

The idea is to construct counterterms, similar as it is done with dipole terms for the real correction part. They must cancel exactly the infrared divergences of dual integrals on integrand level, such that the difference of the dual integral and the counterterm can be obtained by numerical integration in four dimensions. Furthermore counterterms must be simple enough to obtain an analytic result in  $d$  dimensions. The following notation will be used:

$$I^{(n)} = \int dz d^{d-2} q_\perp f^{(n)}(z, q_\perp) = \Delta I^{(n)} + I^{(n)\text{ct}} , \quad (6.28)$$

where

$$\Delta I^{(n)} = \int dz d^2 q_\perp [f^{(n)}(z, q_\perp) - f^{(n)\text{ct}}(z, q_\perp)]_{\epsilon=0} \quad (6.29)$$

is the finite difference term to be evaluated numerically, and

$$I^{(n)\text{ct}} = \int dz d^{d-2} q_\perp f^{(n)\text{ct}}(z, q_\perp) \quad (6.30)$$

the integral over the counterterm.

## Soft integrals

A suitable counterterm for soft integrals can be constructed by neglecting the angular correlation in the transverse momentum. Introducing the same coordinates as for the full dual integral in Eq. (6.11), the counterterm is given by<sup>2</sup>

$$I_s^{(n)\text{ct}}(p, \bar{p}; \{k_i, s_i\}) = \mu^{4-d} \int_0^\infty \frac{dz}{z} \int \frac{d^{d-2}q_\perp}{(2\pi)^{d-1}} \frac{1}{q_\perp^2} \prod_{i=1}^n \left[ \frac{z_i}{q_\perp^2 + z_i^2 (k_i^2 + k_{\perp i}^2) + z_i s_i} \right]. \quad (6.31)$$

Introducing Feynman parameterization and integrating over the longitudinal and transverse momentum one gets

$$I_s^{(n)\text{ct}}(p, \bar{p}; \{k_i, s_i\}) = \frac{2\mu^{2\epsilon}\Gamma(1-\epsilon)\Gamma(n+2\epsilon)}{\epsilon^2 (4\pi)^{2-\epsilon} \prod_{i=1}^n p \cdot k_i} \int_0^1 d\lambda_1 \dots d\lambda_n \delta(1 - \sum_{j=1}^n \lambda_j) (a+b)^\epsilon c^{-n-2\epsilon}, \quad (6.32)$$

where the abbreviations  $a$ ,  $b$  and  $c$  have been defined in Eq. (6.17). The advantage of this expression with respect to Eq. (6.16) is, that all the dependencies on the Feynman parameters have become linear.

Although the soft integral with one non-singular propagator is fully integrable (see Eq. (6.19)), for completeness the result for its divergent component is given by:

$$I_s^{(1)\text{ct}}(p, \bar{p}; \{k_1, s_1\}) = \frac{2c_\Gamma}{\epsilon^2 s_1} \frac{\Gamma(1-2\epsilon)\Gamma(1+2\epsilon)}{\Gamma(1+\epsilon)\Gamma(1-\epsilon)} \left( \frac{2}{p \cdot \bar{p}} \right)^\epsilon a_1^{-\epsilon} b_1^{-\epsilon}, \quad (6.33)$$

where  $a_i = s_i/(p \cdot k_i)$  and  $b_i = s_i/(\bar{p} \cdot k_i)$ . It is easy to verify that both expressions have identical expansions to the orders  $\epsilon^{-2}$  and  $\epsilon^{-1}$ . Since in Feynman parameterization the number of integration variables increases with the number of non-singular propagators it turns out to be easier to derive results for  $n > 1$  directly from the integral representation given in Eq. (6.31). After a trivial angular integration, the integration over the modulus squared of the transverse loop momentum  $q_\perp^2$  can be evaluated by partial fractioning of the non-singular propagators. The remaining integral over the longitudinal loop momentum, after further partial fractioning, has always the form

$$\begin{aligned} \int_0^\infty dz z^{-1-\epsilon} (1+t_1 z)^{-1-\epsilon} (1+t_2 z)^{-1} &= \frac{\Gamma(1+2\epsilon)\Gamma(1-2\epsilon)}{2\epsilon} \left[ (t_1 - t_2)^{-1-\epsilon} t_2^{1+2\epsilon} \right. \\ &\quad \left. - \frac{2\Gamma(1-\epsilon)t_1^\epsilon}{\Gamma(1-2\epsilon)\Gamma(1+\epsilon)} {}_2F_1 \left( 1, -\epsilon; -2\epsilon; \frac{t_2}{t_1} \right) \right]. \end{aligned} \quad (6.34)$$

<sup>2</sup>Although the construction rule for the counterterm is general, it is reminded that the solutions presented here are only strictly valid in a limited kinematic region, given by  $k_i \cdot q > 0$  and  $s_i = k_i^2 > 0$ . The fully general result has to be obtained by a proper analytic continuation, which will be presented elsewhere [125].

The Hypergeometric functions has the expansion

$${}_2F_1(1, -\epsilon; -2\epsilon; z) = \frac{1}{2(1-z)} \left[ 2 - z - \epsilon z \ln 1 - z + \frac{\epsilon^2}{2} z \left( \ln^2(1-z) + 4 \text{Li}_2(z) \right) \right] + \mathcal{O}(\epsilon^3). \quad (6.35)$$

This allows to analytically integrate the counterterm of the soft integral independently of the number of propagators, although the expressions might become quite cumbersome due to the higher number of partial fractionings needed to evaluate this integral for higher multiplicities. In particular, the following expression for the divergent component can be obtained for the two-propagator soft integral

$$\begin{aligned} I_s^{(2)\text{ct}}(p, \bar{p}; \{k_i, s_i\}) &= \frac{2 c_\Gamma}{\epsilon^2 s_1 s_2} \frac{\Gamma(1-2\epsilon)\Gamma(1+2\epsilon)}{\Gamma(1+\epsilon)\Gamma(1-\epsilon)} \left( \frac{2}{p \cdot \bar{p}} \right)^\epsilon \frac{a_1 a_2}{a_1 - a_2} \\ &\times \left[ a_2^{-1-\epsilon} b_2^{-\epsilon} {}_2F_1 \left( 1, -\epsilon; -2\epsilon; \frac{a_1 b_2 - a_2 b_1}{b_1(a_1 - a_2)} \right) \right. \\ &\quad \left. - a_1^{-1-\epsilon} b_1^{-\epsilon} {}_2F_1 \left( 1, -\epsilon; -2\epsilon; \frac{a_1 b_2 - a_2 b_1}{b_2(a_1 - a_2)} \right) \right]. \end{aligned} \quad (6.36)$$

Although it is not obvious at a first sight, this expression is symmetric under the exchange  $a_i \leftrightarrow b_i$ . The solution to the three-propagators soft approximation is

$$\begin{aligned} I_s^{(3)\text{ct}}(p, \bar{p}; \{k_i, s_i\}) &= \frac{2 c_\Gamma}{\epsilon^2 s_1 s_2 s_3} \frac{\Gamma(1-2\epsilon)\Gamma(1+2\epsilon)}{\Gamma(1+\epsilon)\Gamma(1-\epsilon)} \left( \frac{2}{p \cdot \bar{p}} \right)^\epsilon \frac{a_1 a_2 a_3}{\sum_{i=1}^3 c_i(a_{i+1} - a_{i+2})} \\ &\times \left\{ \sum_{i=1}^3 \frac{c_i - c_{i+1}}{a_i - a_{i+1}} \left[ a_i^{-1-\epsilon} b_i^{-\epsilon} {}_2F_1 \left( 1, -\epsilon; -2\epsilon; \frac{b_i(c_i - c_{i+1})}{a_i - a_{i+1}} \right) \right. \right. \\ &\quad \left. \left. - a_{i+1}^{-1-\epsilon} b_{i+1}^{-\epsilon} {}_2F_1 \left( 1, -\epsilon; -2\epsilon; \frac{b_{i+1}(c_i - c_{i+1})}{a_i - a_{i+1}} \right) \right] \right\}, \end{aligned} \quad (6.37)$$

with  $c_i = a_i/b_i$ . These expressions are sufficient to evaluate loops with up to six legs (hexagons). Iterating this procedure, the result can be extended to an arbitrary number of external legs. For  $n$ -propagators, the result is given by

$$\begin{aligned} I_s^{(n)\text{ct}}(p, \bar{p}; \{k_i, s_i\}) &= \frac{2 c_\Gamma}{\epsilon^2 \prod_{i=1}^n p \cdot k_i} \frac{\Gamma(1-2\epsilon)\Gamma(1+2\epsilon)}{\Gamma(1+\epsilon)\Gamma(1-\epsilon)} \left( \frac{2}{p \cdot \bar{p}} \right)^\epsilon \\ &\times \left\{ \sum_{j < k} \frac{(c_j - c_k)^{n-2}}{a_j - a_k} \prod_{l \neq j, k} \frac{1}{(a_j - a_k)(c_j - c_l) - (a_j - a_l)(c_j - c_k)} \right. \\ &\quad \times \left[ a_k^{-1-\epsilon} b_k^{-\epsilon} {}_2F_1 \left( 1, -\epsilon; -2\epsilon; \frac{b_k(c_j - c_k)}{a_j - a_k} \right) \right. \\ &\quad \left. \left. - a_j^{-1-\epsilon} b_j^{-\epsilon} {}_2F_1 \left( 1, -\epsilon; -2\epsilon; \frac{b_j(c_j - c_k)}{a_j - a_k} \right) \right] \right\}. \end{aligned} \quad (6.38)$$

Expanding the full result up to  $\mathcal{O}(\epsilon^{-1})$  yields

$$\begin{aligned} I_s^{(n)\text{ct}}(p, \bar{p}; \{k_i, s_i\}) &= \frac{2 c_\Gamma}{\epsilon^2 \prod_{i=1}^n s_i} \frac{\Gamma(1-2\epsilon)\Gamma(1+2\epsilon)}{\Gamma(1+\epsilon)\Gamma(1-\epsilon)} \left( \frac{2}{p \cdot \bar{p}} \right)^\epsilon \\ &\times \left[ \sum_{j=1}^n \left( \prod_{k \neq j} \frac{a_k}{a_k - a_j} \right) a_j^{-\epsilon} + (b_i \leftrightarrow a_i) - 1 \right] + \mathcal{O}(\epsilon^0), \end{aligned} \quad (6.39)$$

where the coefficients for the  $\epsilon^{-2}$ - and  $\epsilon^{-1}$ -poles must be identical to the ones derived from the full soft dual integral.

### Collinear integrals

Completely analogously to the soft integral a counterterm for the collinear integral can be constructed,

$$I_c^{(n)\text{ct}}(\{k_i, s_i\}) = \mu^{4-d} \int_0^\infty \frac{dz}{2z^2} \int \frac{d^{d-2}q_\perp}{(2\pi)^{d-1}} \prod_{i=1}^n \left[ \frac{z_i}{q_\perp^2 + z_i^2 (k_i^2 + k_{\perp i}^2) + z_i s_i} \right]. \quad (6.40)$$

The full solution in that case is given by

$$\begin{aligned} I_c^{(n)\text{ct}}(p; \{k_i, s_i\}) &= -\frac{c_\Gamma}{2\epsilon^2 \prod_{i=1}^n p \cdot k_i} \frac{\Gamma(1-2\epsilon)\Gamma(1+2\epsilon)}{\Gamma(1+\epsilon)\Gamma(1-\epsilon)} \left( \frac{2}{p \cdot \bar{p}} \right)^\epsilon \\ &\times \left\{ \sum_{j < k} \frac{(c_j - c_k)^{n-2}}{a_j - a_k} \prod_{l \neq j, k} \frac{1}{(a_j - a_k)(c_j - c_l) - (a_j - a_l)(c_j - c_k)} \right. \\ &\times \left[ a_k^{-\epsilon} b_k^{-\epsilon} {}_2F_1 \left( 1, -\epsilon; 1 - 2\epsilon; \frac{b_k(c_j - c_k)}{a_j - a_k} \right) \right. \\ &\left. \left. - a_j^{-\epsilon} b_j^{-\epsilon} {}_2F_1 \left( 1, -\epsilon; 1 - 2\epsilon; \frac{b_j(c_j - c_k)}{a_j - a_k} \right) \right] \right\}, \end{aligned} \quad (6.41)$$

where an auxiliary light-like vector  $\bar{p}$  has been introduced.

### 6.3.2 Recursion relations to single out divergent parts

A different strategy to extract the divergences from the dual integrals can be formulated recursively. There, a dual integral with  $n$  propagators is related to dual integrals of smaller complexity and a finite integral.

## Soft integrals

For the soft dual integral the recursion relation is given by

$$\begin{aligned} I_s^{(n)}(p, \bar{p}; k_1, \dots, k_n) &= \frac{1}{k_1^2} I_s^{(n-1)}(p, \bar{p}; k_2, \dots, k_n) \\ &\quad - \left[ \frac{2p \cdot k_1}{k_1^2} I_c^{(n)}(p; k_1, \dots, k_n) + (p \leftrightarrow \bar{p}) \right] \\ &\quad + \frac{2}{k_1^2} I_{s,fin}^{(n)}(p, \bar{p}; k_1, \dots, k_n), \end{aligned} \quad (6.42)$$

with the IR finite term, which is suitable for a numeric integration in four space-time dimensions,

$$I_{s,fin}^{(n)}(p, \bar{p}; k_1, \dots, k_n) = \int \frac{2 d^2 q}{(2\pi)^3} \delta_+(q^2) \frac{p \cdot q \bar{p} \cdot k_1 + p \cdot k_1 \bar{p} \cdot q - p \cdot \bar{p} q \cdot k_1}{2 p \cdot q \bar{p} \cdot q - p \cdot \bar{p} q^2} \left[ \prod_{j=1}^n \frac{1}{2q \cdot k_j + k_j^2} \right] \quad (6.43)$$

Introducing again light cone coordinates and integrating over  $q_+ = q_\perp^2/(2q_-)$  with the delta function, one gets

$$I_{s,fin}^{(n)} = \int_0^\infty \frac{dz}{z} \int \frac{d^2 q_\perp}{(2\pi)^3} \frac{2k_{1\perp} \cdot q_\perp}{q_\perp^2} \prod_{i=1}^n \left[ \frac{z_i}{(q_\perp - z_i k_{\perp i})^2 + z_i^2 k_i^2 + z_i s_i} \right]. \quad (6.44)$$

Although the integral  $I_{s,fin}^{(n)} = \int dz d^2 q_\perp f_{s,fin}^{(n)}(z, q_\perp)$  is finite in four dimensions, the integrand diverges as  $z \rightarrow 0$ . This numerical defect can be easily fixed by symmetrizing the integrand w.r.t. to the angle in  $q_\perp$ , replacing

$$f_{s,fin}^{(n)}(z, q_\perp) \rightarrow \frac{1}{2} \left( f_{s,fin}^{(n)}(z, q_\perp) + f_{s,fin}^{(n)}(z, -q_\perp) \right). \quad (6.45)$$

Note that the recursion relation in Eq. (6.42) is only one out of  $n$  possible versions, obtained by replacing  $k_1$  with any  $k_i$ .

## Collinear integrals

For the collinear dual integral the recursion relation is given by

$$\begin{aligned} I_c^{(n)}(p; k_1, \dots, k_n) &= \frac{1}{a_1 - a_2} \left[ \frac{a_1}{s_1} I_c^{(n-1)}(p; k_2, \dots, k_n) - \frac{a_2}{s_2} I_c^{(n-1)}(p; k_1, k_3, \dots, k_n) \right. \\ &\quad \left. - \frac{a_1 a_2}{s_1 s_2} I_{c,fin}^{(n)}(p; k_1, \dots, k_n) \right] \end{aligned} \quad (6.46)$$

with the IR finite term

$$I_{c,fin}^{(n)}(p; k_1, \dots, k_n) = \int \frac{2 d^2 q}{(2\pi)^3} \delta_+(q^2) \frac{p \cdot k_2 q \cdot k_1 + p \cdot k_1 q \cdot k_2}{p \cdot q} \left[ \prod_{j=1}^n \frac{1}{2q \cdot k_j + k_j^2} \right]. \quad (6.47)$$

In light cone coordinates and after integrating out the delta function this term reads

$$I_{c,fin}^{(n)}(p; k_1, \dots, k_n) = \int_0^\infty \frac{dz}{z^2} \int \frac{d^2 q_\perp}{(2\pi)^3} \left( \frac{\tilde{k}_{12}\bar{p}}{p\bar{p}} - \frac{\tilde{k}_{12\perp}q_\perp}{z} \right) \times \prod_{i=1}^n \left[ \frac{z_i}{(q_\perp - z_i k_{\perp i})^2 + z_i^2 k_i^2 + z_i s_i} \right], \quad (6.48)$$

with the auxiliary light-like vector  $\bar{p}$  and the shorthand  $\tilde{k}_{12} = p \cdot k_2 k_1 - p \cdot k_1 k_2$ . Equivalent recursion relations are given if  $k_1$  and  $k_2$  are replaced by any other pair of momenta  $k_i$  and  $k_j$ .

Both relations can be applied repeatedly to soft or collinear integrals with any number of propagators, until only the simplest (UV finite) integrals,  $I_s^{(1)}$  and  $I_c^{(2)}$ , and a number of finite integrals are left. Thus only  $I_s^{(1)}$  and  $I_c^{(2)}$  have to be obtained analytically.

Unfortunately this method leads to a high proliferation in the number of terms, which grows roughly exponentially with the number of nonsingular propagators. Nevertheless it still can be a useful method: since all finite integrals are evaluated in the same phase space, they can be combined on integrand level and/or Monte Carlo techniques can be applied, such as sampling over different contributions instead of summing.

### 6.3.3 Numeric evaluation of finite integrals

From the approaches presented above the following infrared finite integrals have emerged:

- finite dual integrals, Eq. (6.26),
- integrals from the counterterm approach, Eq. (6.29), and
- finite integrals from the recursion relation, Eqs. (6.44) and (6.48).

After performing the trivial integration over the delta function in all cases the three dimensional phase space  $dz d^2 q_\perp$  is left for the numerical integration.

When discussing analytic solutions, so far the conditions

$$\begin{aligned} k_i \cdot q &> 0 \quad \text{and} \\ s_i &> 0 \end{aligned} \quad (6.49)$$

always have been imposed. To study general numeric strategies these constraints will be omitted from now. Clearly, outside the region defined in (6.49) the nonsingular propagators may cause further poles. These poles, however, are in principle integrable and do not lead

to physical divergences. To determine solutions in such a case the integration contour must be made unique. The product of nonsingular propagators in the integrand is thus given by

$$P = \prod_{i=1}^n \left[ \frac{z_i}{(q_\perp - z_i k_{\perp i})^2 + z_i^2 k_i^2 + z_i s_i + \xi_i i0} \right], \quad (6.50)$$

where the sign  $\xi_i = \pm 1$  in front of the “ $i0$ ” is determined by the analytic continuation rule for dual integrals (respecting the “ $i0$ ” from the Feynman propagators in the full loop integral, Eq. (6.1)). To be as general as possible, the actual sign will be kept as a free input parameter for the following considerations.

The structure of all integrands is similar, dominated by the product  $P$  of nonsingular propagators or, in case of  $\Delta I$ , the difference of  $P$  and its simplified counterterm version, obtained by neglecting the angular correlation in the perpendicular momentum. The general strategy for the numerical integration is discussed for the example of a finite dual integral. It is straightforward to apply it to any other of the mentioned finite integrals.

### The phase space map

Firstly, the transverse momentum  $q_\perp$  is replaced by the new variable  $u_\perp = q_\perp/z$ . The finite dual integral then reads

$$I_f^{(n)}(\{k_i, s_i\}) = \int_0^\infty \frac{z}{2} dz \int \frac{d^2 u_\perp}{(2\pi)^3} \prod_{i=1}^n \left[ \frac{1}{z_i ((p k_i u_\perp - k_{\perp i})^2 + k_i^2) + s_i \pm i0} \right]. \quad (6.51)$$

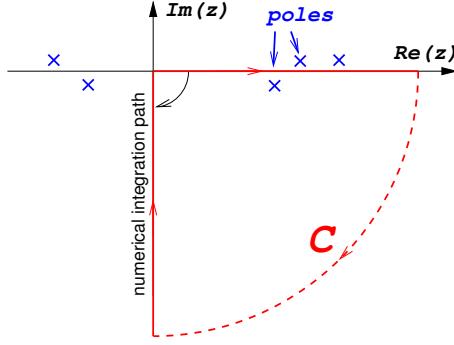
The main advantage of the new parameterization is that each nonsingular propagator now only has a linear dependence on  $z$ .

The integration will be carried out directly in the three dimensions of the loop momentum,  $z$  and  $u_\perp$ , where  $u_\perp = (u \cos \phi, u \sin \phi)$ . In order to apply Monte Carlo methods, the integration phase space must be mapped on a finite interval, such that the integrand remains finite. The following map is suitable for this task:

$$\begin{aligned} \phi &= 2\pi a_1, \\ u &= e^{\tan(\pi(a_2 - 1/2))}, \\ z &= e^{\tan(\pi(a_3 - 1/2))}, \end{aligned} \quad (6.52)$$

where  $a_1, a_2, a_3$  are random variables in the interval  $[0, 1]$ . The corresponding Jacobian reads

$$\mathcal{J} = \frac{uz}{\cos^2(\pi(a_2 - 1/2)) \cos^2(\pi(a_3 - 1/2))}. \quad (6.53)$$



**Figure 6.2:** Rotation of the integration contour for  $z$  to a pole-free path.

### Pole treatment

In the case that poles from nonsingular propagators are present, a naive numerical integration would fail in their neighbourhood. The problem can be overcome by shifting the integration contour for  $z$  from the positive real axis to a pole-free path in the complex  $z$ -plane applying the residue theorem. The location of the poles is easily determined:

$$z_{0i} = -\frac{pk_i(s_i + \xi_i i0)}{(pk_i u_\perp - k_{\perp i})^2 + k_2^2}. \quad (6.54)$$

Thus all poles are located along the real  $z$ -axis. As indicated in Fig. 6.2 the integration contour for  $z$  can be shifted to the pole-free imaginary axis. All finite integrals emerged by decomposing at least box-integrals vanish quickly enough for  $z \rightarrow \infty$ , such that the path closing the contour at infinity does not contribute. Formally the integral is transformed as follows:

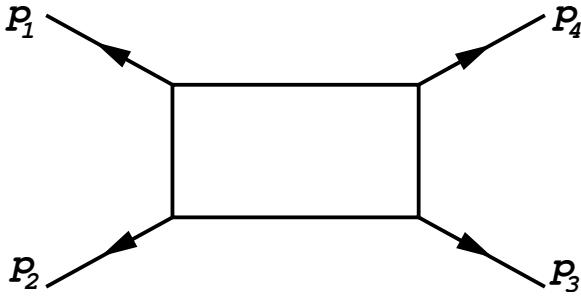
$$\begin{aligned} \int dz du_\perp f(z, u_\perp) &\rightarrow \int dz du_\perp e^{i\pi\lambda} f(e^{i\pi\lambda} z, u_\perp) \\ &\quad + 2i\pi\lambda \int du_\perp \sum_{z_0(u_\perp) \in \mathcal{C}} \text{Res}_{z_0} f(z_0, u_\perp), \end{aligned} \quad (6.55)$$

where  $\lambda = \pm 1$  determines whether the new integration path is along the positive or negative imaginary axis. The second term in Eq. (6.55) sums the residues of all poles inside the contour  $\mathcal{C}$ . The residues are calculated as follows

$$\text{Res}_{z_0} f(z_0, u_\perp) = \frac{1}{\frac{d}{dz} (f(z_0, u_\perp)^{-1})|_{z=z_0}}. \quad (6.56)$$

Thus, if present, residue terms yield an additional integral, now over the two dimensional phase space of  $du_\perp$ , to be performed also numerically.

The sign parameters  $\xi_i$  determine whether a residue is inside or outside the considered contour. Clearly, different choices lead to different results. The choice of the actual integration



**Figure 6.3:** Setup for a box diagram.

path can be optimized, such, that the number of residues to be calculated is minimal. The following cases can be distinguished:

1. No poles along the positive real  $z$ -axis: no contour shift, no residues to be calculated;
2. All poles along the positive real  $z$ -axis have a infinitesimal positive (negative) imaginary part: the contour shift to the negative (positive) imaginary axis, no residues to be calculated;
3. Poles are located on both sides of the positive real  $z$ -axis: the contour shift such that the least number of poles is crossed, corresponding residues must be calculated.

#### 6.3.4 Example: finite box integral

The procedure for a numerical evaluation of finite integrals has been implemented in a C++-code. To improve the convergence a self-adaptive integrator, based on the FOAM[75] algorithm has been used to generate the random numbers  $a_1, a_2, a_3$  in Eq. (6.52) non-uniformly. This allows an appropriate adaptation to the integrands, which are in general not factorizable in their single variables. All types of finite integrals,  $I_f^n$ ,  $\Delta I_{s/c}^{(n)}$  and  $I_{s/c,fin}^{(n)}$  converge well for a number of tested configurations stemming from calculations up to hexagon loops.

As an explicit example to verify the numeric integration procedure and the decomposition of a loop diagram into dual integrals, a box diagram, massive in all external legs is considered, cf. Fig (6.3). This example involves only finite dual integrals, some of which enclose poles at the positive real  $z$ -axis to be treated with the procedure described above.

The computation has been done for the set of four-momenta, listed in Tab. 6.1, all defined as outgoing momenta. Tab. 6.2 summarizes the numeric results, with statistical errors on the last digit given in parenthesis. In [126] an analytic expression for the four-mass box is given. For the momenta from Tab. 6.1 this expression yields the result

$$I = -5.701407 \times 10^{-8} - i * 3.308353 \times 10^{-8},$$

$i$	$p^\mu (E, p_x, p_y, p_z)$	$m = \sqrt{p^2}$
1	(-50.0099990, 0., 0., -50.0000000)	1
2	(-50.1597448, 0., 0., 50.0000000)	4
3	(49.6056853, -32.3801259, -13.4122873, -35.0479958)	2
4	(50.5640585, 32.3801259, 13.4122873, 35.0479958)	10

**Table 6.1:** Example setup for the finite box.

$I_f^{(3)}(-p_1, p_2, p_{23})$	$-2.7827(4) \times 10^{-8}$	$-i * 3.8598(9) \times 10^{-9}$
$I_f^{(3)}(-p_2, p_3, p_{34})$	$-1.1440(1) \times 10^{-8}$	$+i * 0$
$I_f^{(3)}(-p_3, p_4, p_{14})$	$-9.166(3) \times 10^{-9}$	$-i * 2.9224(3) \times 10^{-8}$
$I_f^{(3)}(-p_4, p_1, p_{12})$	$-8.583(1) \times 10^{-9}$	$+i * 0$
sum	$-5.7019(5) \times 10^{-8}$	$-i * 3.3083(4) \times 10^{-8}$

**Table 6.2:** Numeric results for the four-mass box using momenta from Tab. 6.1. The contributions from the four finite dual integrals and their sum are listed.

which is in full agreement with the numerically obtained result.

## 6.4 Conclusions and outlook

In this section a new promising method to evaluate  $n$ -point loop integrals has been presented. The heart of this approach is the decomposition of loop integrals to dual, phase-space like integrals. In particular semi-numerical strategies to solve dual integrals have been considered. The structure of those integrals allow the application of Monte-Carlo methods, developed for phase space integrals. The numerical performance is very promising and the increase of complexity with a rising number of external legs is manageable.

In order to make this method suitable for physical applications two issues are still under consideration: firstly, a procedure for a correct analytic continuation of dual integrals has to be formulated, and secondly, the method must be generalized to tensor integrals. There, a straightforward decomposition leads to a number of dual integrals containing additional numerator terms, for which similar techniques as for the basic dual integrals can be applied.

## 7 Summary

The subject of this thesis was the development of tools for the automated calculation of exact matrix elements, which are a key for the systematic improvement of precision and confidence for theoretical predictions. In particular, the LHC sets new benchmarks on the complexity of multi-particle final states, for which, to cope with, new strategies for the calculation of ME were designed and implemented.

Part I of this thesis concentrates on the calculations of cross sections at tree level. A number of extensions have been implemented in the matrix element generator `AMEGIC++`, namely new interaction models such as effective loop-induced couplings of the Higgs boson with massless gauge bosons, required for a number of channels for the Higgs boson search at LHC and anomalous gauge couplings, parameterizing a number of models beyond the SM. Further a special treatment to deal with complicated decay chains of heavy particles has been constructed. A significant effort went into the implementation of methods to push the limits on particle multiplicities. This was most crucial for pure QCD final states, which are produced with extremely high rates at LHC, but, applying traditional methods, display the most severe growth in the calculational complexity with number of involved particles. Two recursive methods have been implemented, the Cachazo-Svrček-Witten recursion and the colour dressed Berends-Giele recursion. For the latter the new module `COMIX` has been added to the `SHERPA` framework. The Monte-Carlo phase space integration techniques have been completely revised, which led to significantly reduced statistical error estimates when calculating cross sections and a greatly improved unweighting efficiency for the event generation. Special integration methods have been developed to cope with the newly accessible final states. The event generation framework `SHERPA` directly benefits from those new developments, improving the precision and the efficiency.

Part II was addressed to the automation of QCD calculations at next-to-leading order. Although highly demanded, so far no fully automated tool is available for this task. In this thesis (Chapter 5) a code has been developed, that, for the first time fully automates the real correction part of a NLO calculation. To calculate the correction for a  $m$ -parton process obeying the Catani-Seymour dipole subtraction method the following components are provided:

1. the corresponding  $m + 1$ -parton tree level matrix elements,
2. a number dipole subtraction terms to remove the soft and collinear divergencies
3. the finite part of the integrated subtraction terms, added back to make the full real correction term independent of the regularization method.

Furthermore, integrators for all necessary phase space integrals are provided. The new implementation is based on the matrix element generator AMEGIC++. The resulting code can easily be extended to a full parton-level generator at NLO by supplementing it with a library of one-loop matrix elements. The limitations on the number of external particles are the same as for tree level matrix elements and thus beyond the currently available one-loop matrix elements. For the community of physicist performing one-loop calculations this tool can provide a significant facilitation: although this part of the calculation is completely described in Ref. [42], the practical realization is quite tedious in most cases.

The virtual matrix elements are currently far from being automated. The main difficulties arise for the evaluation of  $n$ -point one-loop integrals for  $n > 4$ . Most promising candidates to resolve this obstacle are semi-numerical approaches to integrate over the loop momentum. Such a technique has been explored in Chapter 6. A new decomposition of loop integrals into phase-space integrals is proposed and semi-numerical strategies to evaluate the new integrals were developed. Although some details still have to be finalized, this approach indicates a number of promising features: Firstly, the extraction of infrared divergencies is straightforward after the decomposition, leaving numerically well-behaved finite integrals, accessible to Monte-Carlo phase space integration techniques. Furthermore the procedure is completely general for an arbitrary numbers of legs and has only a decent growth in complexity.

Altogether it can be concluded that the matrix element generators AMEGIC++ and COMIX together with the phase space integration form state-of-the-art tools, making the event generator SHERPA being well prepared for the challenges of the LHC.

Towards a full automation of next-to-leading order calculations a significant progress could be made.

# Appendix A COMIX implementation details

## A.1 Decomposition of electroweak four-particle vertices

A decomposition of four particle vertices with  $W$ -bosons only is given by

$$\mathcal{V}_{W^{-\nu}}^{W^{-\rho}, W^{+\sigma}, W^{-\lambda}} \rightarrow \mathcal{V}_{W^{-\nu}}^{W^{-\rho}, Z_4\gamma\delta} \cdot P_{Z_4\gamma\delta}^{\alpha\beta} \cdot \mathcal{V}_{Z_4\alpha\beta}^{W^{+\sigma}, W^{-\lambda}} + \mathcal{V}_{W^{-\nu}}^{W^{-\lambda}, Z_4\gamma\delta} \cdot P_{Z_4\gamma\delta}^{\alpha\beta} \cdot \mathcal{V}_{Z_4\alpha\beta}^{W^{+\sigma}, W^{-\rho}}. \quad (\text{A.1})$$

Here  $Z_4$  denotes a new antisymmetric tensor pseudo-particle introduced for the vertex decomposition. Its interaction vertex reads

$$\begin{aligned} \mathcal{V}_{W^{-\nu}}^{W^{-\rho}, Z_4\gamma\delta} &= \frac{i}{2} g_w (g_\nu^\gamma g^{\rho\delta} - g_\nu^\delta g^{\rho\gamma}), \\ \mathcal{V}_{Z_4\alpha\beta}^{W^{+\sigma}, W^{-\rho}} &= \frac{i}{2} g_w (g_\alpha^\sigma g_\beta^\rho - g_\alpha^\rho g_\beta^\sigma). \end{aligned} \quad (\text{A.2})$$

To obtain correct signs of four-particle vertices, the tensor pseudo-particle “propagators” are defined as

$$P_{\alpha\mu\nu}^{\rho\sigma} = \kappa_\alpha D_{\mu\nu}^{\rho\sigma} \quad \text{where} \quad \kappa_\alpha = \begin{cases} -i & \text{if } \alpha = Z_4 \\ i & \text{else} \end{cases}, \quad (\text{A.3})$$

and where  $D_{\mu\nu}^{\rho\sigma}$  is given by Eq. (2.41). Note that the  $Z_4$  pseudo-particle is not its own antiparticle. This definition prevents double counting of four-particle vertices involving the  $W$  boson and constructing fake  $WWWW$  vertices with all  $W$ s having the same charge. The four-particle vertices involving  $W$  bosons, photons and  $Z$ -bosons are decomposed as follows

$$\begin{aligned} \mathcal{V}_{W^{-\nu}}^{A\rho, W^{-\sigma}, A\lambda} &\rightarrow \mathcal{V}_{W^{-\nu}}^{A\rho, W_4^{-\gamma\delta}} \cdot P_{W_4^{-\gamma\delta}}^{\alpha\beta} \cdot \mathcal{V}_{W_4^{-\alpha\beta}}^{W^{-\sigma}, A\lambda} + \mathcal{V}_{W^{-\nu}}^{A\lambda, W_4^{-\gamma\delta}} \cdot P_{W_4^{-\gamma\delta}}^{\alpha\beta} \cdot \mathcal{V}_{W_4^{-\alpha\beta}}^{W^{-\sigma}, A\rho}, \\ \mathcal{V}_{W^{-\nu}}^{A\rho, W^{-\sigma}, Z\lambda} &\rightarrow \mathcal{V}_{W^{-\nu}}^{A\rho, W_4^{-\gamma\delta}} \cdot P_{W_4^{-\gamma\delta}}^{\alpha\beta} \cdot \mathcal{V}_{W_4^{-\alpha\beta}}^{W^{-\sigma}, Z\lambda} + \mathcal{V}_{W^{-\nu}}^{Z\lambda, W_4^{-\gamma\delta}} \cdot P_{W_4^{-\gamma\delta}}^{\alpha\beta} \cdot \mathcal{V}_{W_4^{-\alpha\beta}}^{W^{-\sigma}, A\rho}, \\ \mathcal{V}_{W^{-\nu}}^{Z\rho, W^{-\sigma}, Z\lambda} &\rightarrow \mathcal{V}_{W^{-\nu}}^{Z\rho, W_4^{-\gamma\delta}} \cdot P_{W_4^{-\gamma\delta}}^{\alpha\beta} \cdot \mathcal{V}_{W_4^{-\alpha\beta}}^{W^{-\sigma}, Z\lambda} + \mathcal{V}_{W^{-\nu}}^{Z\lambda, W_4^{-\gamma\delta}} \cdot P_{W_4^{-\gamma\delta}}^{\alpha\beta} \cdot \mathcal{V}_{W_4^{-\alpha\beta}}^{W^{-\sigma}, Z\rho}. \end{aligned} \quad (\text{A.4})$$

A new tensor pseudo-particle  $W_4^-$  is introduced, whose interaction vertices are defined as

$$\begin{aligned}\mathcal{V}_{W^-}^{A\rho, W_4^- \gamma\delta} &= \frac{i}{2} g_w \cos \theta_W (g_\nu^\gamma g^{\rho\delta} - g_\nu^\delta g^{\rho\gamma}) , \\ \mathcal{V}_{W_4^- \alpha\beta}^{W^- \sigma, A\rho} &= \frac{i}{2} g_w \sin \theta_W (g_\alpha^\sigma g_\beta^\rho - g_\alpha^\rho g_\beta^\sigma) , \\ \mathcal{V}_{W^-}^{Z\rho, W_4^- \gamma\delta} &= \frac{i}{2} g_w \cos \theta_W (g_\nu^\gamma g^{\rho\delta} - g_\nu^\delta g^{\rho\gamma}) , \\ \mathcal{V}_{W_4^- \alpha\beta}^{W^- \sigma, Z\rho} &= \frac{i}{2} g_w \cos \theta_W (g_\alpha^\sigma g_\beta^\rho - g_\alpha^\rho g_\beta^\sigma) .\end{aligned}\quad (\text{A.5})$$

Corresponding vertices exist for  $W^+$  /  $W^-$  bosons. The decomposition of four particle vertices involving the Higgs boson introduces a new scalar pseudo-particle, which is denoted by  $h_4$ . In order not to generate fake four particle vertices it is by definition not its own antiparticle. The corresponding vertices read

$$\begin{aligned}\mathcal{V}_h^{h, h, h} &\rightarrow \mathcal{V}_h^{h, h_4} \cdot P_{h_4} \cdot \mathcal{V}_{h_4}^{hh} , \\ \mathcal{V}_h^{h, Z\mu, Z\nu} &\rightarrow \mathcal{V}_h^{h, h_4} \cdot P_{h_4} \cdot \mathcal{V}_{h_4}^{Z\mu, Z\nu} , \\ \mathcal{V}_h^{h, W^+\mu, W^-\nu} &\rightarrow \mathcal{V}_h^{h, h_4} \cdot P_{h_4} \cdot \mathcal{V}_{h_4}^{W^+\mu, W^-\nu} .\end{aligned}\quad (\text{A.6})$$

where the interactions of the  $h_4$  pseudo-particle are defined by

$$\begin{aligned}\mathcal{V}_{h_4}^{h, h} &= i \frac{m_h^2}{v^2} , \\ \mathcal{V}_{h_4}^{Z\mu, Z\nu} &= -i \frac{g_w^2}{2 \cos^2 \theta_W} g^{\mu\nu} , \\ \mathcal{V}_h^{h, h_4} &= i , \\ \mathcal{V}_{h_4}^{W^+\mu, W^-\nu} &= -i \frac{g_w^2}{2} g^{\mu\nu} ,\end{aligned}\quad (\text{A.7})$$

and where the scalar “propagator” of the  $h_4$  pseudo-particle

$$P_{h_4} = i \quad (\text{A.8})$$

has been introduced. Since all remaining vertices in the Standard Model are three point vertices, the vertex decomposition is hereby complete. Finally all vertices employed in the recursive relations are listed in A.4.

## A.2 Matrix element generation with COMIX

The spinor basis introduced in Ref. [127] is used. The  $\gamma$ -matrices are taken in the Weyl representation, i.e.

$$\gamma_\mu = \begin{pmatrix} 0 & \sigma_\mu \\ \bar{\sigma}_\mu & 0 \end{pmatrix} , \quad \gamma^5 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad (\text{A.9})$$

where  $\sigma_\mu = (1, -\vec{\sigma})$  and  $\bar{\sigma}_\mu = (1, \vec{\sigma})$ . and  $\sigma^i$  are the Pauli matrices. Defining  $p^\pm = p^0 \pm p^3$  and  $p_\perp = p^1 + ip^{2-1}$  as well as  $\bar{p} = \text{sgn}(p_0) |\vec{p}|$  and  $\hat{p} = (\bar{p}, \vec{p})$ , a possible set of Eigenspinors to the Dirac equations is then given by

$$u_+(p, m) = \frac{1}{\sqrt{2\bar{p}}} \begin{pmatrix} \sqrt{p_0 - \bar{p}} \chi_+(\hat{p}) \\ \sqrt{p_0 + \bar{p}} \chi_+(\hat{p}) \end{pmatrix}, \quad v_-(p, m) = \frac{1}{\sqrt{2\bar{p}}} \begin{pmatrix} -\sqrt{p_0 - \bar{p}} \chi_+(\hat{p}) \\ \sqrt{p_0 + \bar{p}} \chi_+(\hat{p}) \end{pmatrix}, \quad (\text{A.10})$$

$$u_-(p, m) = \frac{1}{\sqrt{2\bar{p}}} \begin{pmatrix} \sqrt{p_0 + \bar{p}} \chi_-(\hat{p}) \\ \sqrt{p_0 - \bar{p}} \chi_-(\hat{p}) \end{pmatrix}, \quad v_+(p, m) = \frac{1}{\sqrt{2\bar{p}}} \begin{pmatrix} \sqrt{p_0 + \bar{p}} \chi_-(\hat{p}) \\ -\sqrt{p_0 - \bar{p}} \chi_-(\hat{p}) \end{pmatrix}. \quad (\text{A.11})$$

Here the Weyl spinors have been defined by

$$\chi_+(\hat{p}) = \frac{1}{\sqrt{\hat{p}^+}} \begin{pmatrix} \hat{p}^+ \\ \hat{p}_\perp \end{pmatrix} = \begin{pmatrix} \sqrt{\hat{p}^+} \\ \sqrt{\hat{p}^-} e^{i\phi_{\hat{p}}} \end{pmatrix}, \quad \chi_-(\hat{p}) = \frac{e^{i\pi}}{\sqrt{\hat{p}^+}} \begin{pmatrix} -\hat{p}_\perp^* \\ \hat{p}^+ \end{pmatrix} = \begin{pmatrix} \sqrt{\hat{p}^-} e^{-i\phi_{\hat{p}}} \\ -\sqrt{\hat{p}^+} \end{pmatrix}, \quad (\text{A.12})$$

and are orthogonal and normalized to  $2 |\hat{p}_0|$ . The Eigenspinors  $u_\pm$  and  $v_\pm$  are thus orthogonal and normalized to  $2m$  and  $-2m$ , respectively. These spinor states are, of course, equivalent to those used in AMEGIC++, cf. section 2.1.1, corresponding to a certain choice of the auxiliary vectors  $k_0$  and  $k_1$ . The main difference is, that in COMIX spinors are computed explicitly and products being evaluated by a matrix multiplication.

Polarization vectors for external vector bosons are constructed according to Refs. [128]. For massless gauge bosons they can be defined via

$$\varepsilon_\pm^\mu(p, k) = \pm \frac{\langle k^\mp | \gamma^\mu | p^\mp \rangle}{\sqrt{2} \langle k^\mp | p^\pm \rangle}, \quad (\text{A.13})$$

where  $k$  is an arbitrary light-like gauge vector, which must not be parallel to the momentum  $p$ . For massive bosons

$$\varepsilon_\pm^\mu(p, k) = \pm \frac{\langle k^\mp | \gamma^\mu | b^\mp \rangle}{\sqrt{2} \langle k^\mp | b^\pm \rangle}, \quad \varepsilon_0^\mu(p, k) = \frac{1}{m} (\langle b^+ | \gamma^\mu | b^- \rangle - \kappa \langle k^+ | \gamma^\mu | k^- \rangle), \quad (\text{A.14})$$

is used, where

$$b = p - \kappa k, \quad \kappa = \frac{p^2}{2pk} \quad (\text{A.15})$$

and again  $k$  is an arbitrary light-like vector. The gauge vectors  $k$  have no physical meaning and thus any scattering amplitude must be independent of the explicit values of  $k$ . This fact is employed in numerical implementations of the above equations to perform a check of gauge invariance.

As pointed out in Sec. 2.3, within the Standard Model tensor particles never occur as real states, such that there is no need to explicitly construct polarization tensors.

<sup>1</sup>Note that the  $x$ -,  $y$ - and  $z$ -directions are not fixed but can be defined through any orthogonal set of vectors. In the following it will be referred to the definition of these directions as the spinor gauge. The arbitrariness of the spinor gauge is employed in the numerical implementation to perform a check for gauge invariance of the amplitude.

Vertex ID	Lorentz structures		
FFS			
FFV <sup>-</sup>			
FFV <sup>+</sup>			
VVS			
VVV(p, q)			
VVT			

Table A.1: Lorentz structures of Standard Model interactions.

### A.3 Lorentz functions

In this appendix explicit expressions for all possible Lorentz vertex structures occurring in the Berends-Giele recursion defined by the Standard Model are listed. They are sorted by ascending spin of the connecting particles and employ the following notation.

- S Scalar,
- F Fermion,
- V Vector Boson,
- T Antisymmetric tensor of rank two.

It is stressed again that all interaction terms occurring in the standard model Lagrangian yield no more than three-particle vertices of the above defined particle types with the possible couplings listed in Appendix A.4. The Quantities listed in Tab. A.1 in explicit form are

given by

$$\bar{u}j^\mu\gamma_\mu\frac{1-\gamma^5}{2} = (0, 0, \bar{u}_0j^- - \bar{u}_1j_\perp, -\bar{u}_0j_\perp^* + \bar{u}_1j^+) , \quad j^\mu\gamma_\mu\frac{1-\gamma^5}{2}v = \begin{pmatrix} 0 \\ 0 \\ j^+v_0 + j_\perp^*v_1 \\ j_\perp v_0 + j^-v_1 \end{pmatrix} , \quad (\text{A.16})$$

$$\bar{u}\gamma^\mu\frac{1-\gamma^5}{2}v = \begin{pmatrix} \bar{u}_0v_2 + \bar{u}_1v_3 \\ \bar{u}_0v_3 + \bar{u}_1v_2 \\ i(\bar{u}_1v_2 - \bar{u}_0v_3) \\ \bar{u}_0v_2 - \bar{u}_1v_3 \end{pmatrix} , \quad (\text{A.17})$$

$$\bar{u}j^\mu\gamma_\mu\frac{1+\gamma^5}{2} = (\bar{u}_2j^+ + \bar{u}_3j_\perp, \bar{u}_2j_\perp^* + \bar{u}_3j^-, 0, 0) , \quad j^\mu\gamma_\mu\frac{1+\gamma^5}{2}v = \begin{pmatrix} j^-v_2 - j_\perp^*v_3 \\ -j_\perp v_2 + j^+v_3 \\ 0 \\ 0 \end{pmatrix} , \quad (\text{A.18})$$

$$\bar{u}\gamma^\mu\frac{1+\gamma^5}{2}v = \begin{pmatrix} \bar{u}_2v_0 + \bar{u}_3v_1 \\ -\bar{u}_2v_1 - \bar{u}_3v_0 \\ i(\bar{u}_2v_1 - \bar{u}_3v_0) \\ -\bar{u}_2v_0 + \bar{u}_3v_1 \end{pmatrix} , \quad (\text{A.19})$$

$$\Gamma^{\nu\sigma\rho}(p, q)\varepsilon_\sigma\varepsilon'_\rho = \varepsilon\varepsilon'(p-q)^\nu + \varepsilon'(2q+p)\varepsilon^\nu - \varepsilon(2p+q)\varepsilon'^\nu , \quad (\text{A.20})$$

$$\tau^{\mu\nu}(\varepsilon, \varepsilon') = \frac{1}{2}(g^{\mu\sigma}g^{\nu\rho} - g^{\mu\rho}g^{\nu\sigma})\varepsilon_\sigma\varepsilon'_\rho . \quad (\text{A.21})$$

Note that due to the antisymmetry of  $\tau^{\mu\nu}$ , the following replacement can be made

$$\frac{1}{2}(g_\alpha^\mu g_\beta^\nu - g_\beta^\mu g_\alpha^\nu)\tau^{\alpha\beta} = \tau^{\mu\nu} , \quad (\text{A.22})$$

which leads to an asymmetric form of the VVT vertex, and a slight decrease in evaluation time.

## A.4 Vertices and Propagators

In this appendix explicitly all vertices occurring in the recursive relations for the Standard Model as formulated in Sec. 2.3 are listed. Their Lorentz structures are defined in Appendix A.3.

### QCD interactions

$$\begin{array}{c} q, K \\ \swarrow \quad \searrow \\ \text{---} \quad \text{---} \\ \text{---} \quad \text{---} \\ g, H\bar{G} \end{array} = -i \frac{g_s}{\sqrt{2}} \left[ \delta_H^K \delta_{\bar{G}}^{\bar{L}} - \frac{1}{N_C} \delta_{H\bar{G}} \delta^{K\bar{L}} \right] (\text{FFV}^- + \text{FFV}^+)$$

$$\begin{array}{c} g(p), K\bar{L} \\ \text{---} \quad \text{---} \\ \text{---} \quad \text{---} \\ g, H\bar{G} \end{array} = i \frac{g_s}{\sqrt{2}} \left[ \delta_H^M \delta^{K\bar{N}} \delta_{\bar{G}}^{\bar{L}} \text{VVV}(p, q) - \delta_H^K \delta^{\bar{L}M} \delta_{\bar{G}}^{\bar{N}} \text{VVV}(q, p) \right]$$

$$\begin{array}{c} g, K\bar{L} \\ \text{---} \quad \text{---} \\ \text{---} \quad \text{---} \\ g_4, H\bar{G} \end{array} = i \frac{g_s}{\sqrt{2}} \left[ \delta_H^M \delta^{K\bar{N}} \delta_{\bar{G}}^{\bar{L}} + \delta_H^K \delta^{\bar{L}M} \delta_{\bar{G}}^{\bar{N}} \right] \text{VVT}$$

$$I \xrightarrow{q} H = i \delta_I^H \frac{\hat{p} + m}{p^2 - m^2}$$

$$\mu, I\bar{J} \xrightarrow{g} \nu, H\bar{G} = i \delta_I^H \delta_{\bar{J}}^{\bar{G}} \frac{-g_{\mu\nu}}{p^2}$$

$$\rho\sigma, I\bar{J} \xrightarrow{g_4} \mu\nu, H\bar{G} = -i \delta_I^H \delta_{\bar{J}}^{\bar{G}} D_{\mu\nu}^{\rho\sigma}$$

### QED interactions

$$\begin{array}{c} f \\ \swarrow \quad \searrow \\ \text{---} \quad \text{---} \\ \gamma \end{array} = -i g_e Q_f (\text{FFV}^- + \text{FFV}^+)$$

$$\begin{array}{c} f \\ \xrightarrow{q} \end{array} = i \frac{\hat{p} + m_f}{p^2 - m_f^2}$$

$$\mu \xrightarrow{\gamma} \nu = i \frac{-g_{\mu\nu} + p^\mu p^\nu / p^2}{p^2}$$

### Electroweak interactions

$$\begin{array}{c} h \\ \text{---} \quad \text{---} \\ \text{---} \quad \text{---} \\ h \end{array} = i \frac{3m_h^2}{v}$$

Diagram 1: Vertex for  $h$  and  $h_4$ .

$$\begin{array}{ccc} \text{Diagram: } h \text{ (dashed)} \text{ and } h_4 \text{ (dashed)} & = & i \frac{m_h^2}{v^2} \end{array}$$

Diagram 2: Vertex for  $f$  and  $\bar{f}$ .

$$\begin{array}{ccc} \text{Diagram: } f \text{ (solid)} \text{ and } \bar{f} \text{ (solid)} & = & -i \frac{m_f}{v} \text{ FFS} \end{array}$$

Diagram 3: Vertex for  $f$  and  $\bar{f}$  with  $Z$ .

$$\begin{array}{ccc} \text{Diagram: } f \text{ (solid)} \text{ and } \bar{f} \text{ (solid)} \text{ with } Z \text{ (wavy)} & = & -i \frac{g_w}{2 \cos \theta_W} \{ (V_f + A_f) \text{ FFV}^- + (V_f - A_f) \text{ FFV}^+ \} \end{array}$$

Diagram 4: Vertex for  $f$  and  $\bar{f}'$  with  $W^+$ .

$$\begin{array}{ccc} \text{Diagram: } f \text{ (solid)} \text{ and } \bar{f}' \text{ (solid)} \text{ with } W^+ \text{ (wavy)} & = & -i \frac{g_w}{\sqrt{2}} T_{ff'}^+ \text{ FFV}^- \end{array}$$

Diagram 5: Vertex for  $W/Z$  and  $W/Z$ .

$$\begin{array}{ccc} \text{Diagram: } W/Z \text{ (wavy)} \text{ and } W/Z \text{ (dashed)} & = & -i \frac{g_w m_{W/Z}}{\lambda_{W/Z}} VVS \quad \text{where} \quad \begin{array}{ll} \lambda_W = & 1 \\ \lambda_Z = & \cos \theta_W \end{array} \end{array}$$

Diagram 6: Vertex for  $W/Z$  and  $W/Z$ .

$$\begin{array}{ccc} \text{Diagram: } W/Z \text{ (wavy)} \text{ and } W/Z \text{ (dashed)} & = & -i \frac{g_w^2}{2 \lambda_{W/Z}^2} VVS \quad \text{where} \quad \begin{array}{ll} \lambda_W = & 1 \\ \lambda_Z = & \cos \theta_W \end{array} \end{array}$$

Diagram 7: Vertex for  $W^-(p)$  and  $W^+(q)$ .

$$\begin{array}{ccc} \text{Diagram: } W^-(p) \text{ (wavy)} \text{ and } W^+(q) \text{ (dashed)} & = & i g_w \kappa_{A/Z} VVV(p, q) \quad \text{where} \quad \begin{array}{ll} \kappa_A = & \sin \theta_W \\ \kappa_Z = & \cos \theta_W \end{array} \end{array}$$

Diagram 8: Vertex for  $W^-$  and  $W^+$ .

$$\begin{array}{ccc} \text{Diagram: } W^- \text{ (wavy)} \text{ and } W^+ \text{ (dashed)} & = & i g_w VVT \end{array}$$

Diagram 9: Vertex for  $W^-$  and  $A/Z$ .

$$\begin{array}{ccc} \text{Diagram: } W^- \text{ (wavy)} \text{ and } A/Z \text{ (dashed)} & = & i g_w \kappa_{A/Z} VVT \quad \text{where} \quad \begin{array}{ll} \kappa_A = & \sin \theta_W \\ \kappa_Z = & \cos \theta_W \end{array} \end{array}$$

Diagram 10: Propagator for  $h$ .

$$\begin{array}{ccc} \text{Diagram: } h \text{ (dashed)} & = & \frac{i}{p^2 - m_h^2} \end{array}$$

Diagram 11: Propagator for  $h_4$ .

$$\begin{array}{ccc} \text{Diagram: } h_4 \text{ (dashed)} & = & i \end{array}$$

Diagram 12: Propagator for  $W/Z$ .

$$\begin{array}{ccc} \text{Diagram: } \mu \text{ (wavy)} \text{ and } \nu \text{ (solid)} & = & i \frac{-g_{\mu\nu} + p^\mu p^\nu / m_{W/Z}^2}{p^2 - m_{W/Z}^2} \end{array}$$

$$\begin{array}{ccc} \mu\nu \cdots \stackrel{Z_4}{\cdots} \rho\sigma & = & -i D_{\mu\nu}^{\rho\sigma} \\ & & \mu\nu \cdots \stackrel{W_4^\pm}{\cdots} \rho\sigma & = & i D_{\mu\nu}^{\rho\sigma} \end{array}$$

Here the definition

$$V_f = T_f^3 - 2Q_f \sin^2 \theta_W, \quad A_f = T_f^3 \quad (\text{A.23})$$

was used.

# Appendix B Insertion operators for the dipole subtraction method

In this appendix, the ingredients of the master equation Eq. (5.52),

$$\begin{aligned}
d\sigma_{ab}^{\tilde{A}}(p_a, p_b) + d\sigma_{ab}^C(p_a, p_b, \mu_F^2) &= [d\sigma_{ab}^B(p_a, p_b) \times \mathbf{I}(\epsilon)] \\
&+ \sum_{a'} \int_0^1 dx \left[ \left( \mathbf{K}^{a,a'}(x) + \mathbf{P}^{a,a'}(xp_a, x; \mu_F^2) \right) \times d\sigma_{a'b}^B(xp_a, p_b) \right] \\
&+ \sum_{b'} \int_0^1 dx \left[ \left( \mathbf{K}^{b,b'}(x) + \mathbf{P}^{b,b'}(xp_b, x; \mu_F^2) \right) \times d\sigma_{ab'}^B(p_a, xp_b) \right] ,
\end{aligned} \tag{B.1}$$

will be repeated.  $a$  and  $b$  specify initial state partons, and the sum runs over all accessible  $a'$  and  $b'$  occurring in the PDF. The insertion operator  $\mathbf{I}$  is given by

$$\mathbf{I}(\{p\}; \epsilon) = -\frac{\alpha_S}{2\pi} \frac{1}{\Gamma(1-\epsilon)} \sum_I \frac{1}{\mathbf{T}_I^2} \mathcal{V}_I(\epsilon) \sum_{I \neq J} \mathbf{T}_I \cdot \mathbf{T}_J \left( \frac{4\pi\mu^2}{2p_I p_J} \right)^\epsilon, \tag{B.2}$$

cf. Eq. (5.53), and again the indices  $I$  and  $J$  run over all initial and final state partons, while the universal functions  $\mathcal{V}_I(\epsilon)$ , encoding the singularity structure, merely depend on the flavour of  $I$  and read

$$\mathcal{V}_I(\epsilon) = \mathbf{T}_I^2 \left( \frac{1}{\epsilon^2} - \frac{\pi^2}{3} \right) + \gamma_I \left( \frac{1}{\epsilon} + 1 \right) + K_I + \mathcal{O}(\epsilon), \tag{B.3}$$

cf. Eq. (5.54). The individual  $\gamma_I$  and  $K_I$  will be listed in Eqs. (B.7) and (B.8).

The factorization scale dependent terms are proportional to insertion operators  $\mathbf{P}^{a,a'}(\{p\}, xp_a, x; \mu_F^2)$ , which read

$$\mathbf{P}^{a,a'}(\{p\}, xp_a, x; \mu_F^2) = \frac{\alpha_S}{2\pi} P^{aa'}(x) \frac{1}{\mathbf{T}_{a'}^2} \sum_{I \neq a'} \mathbf{T}_I \cdot \mathbf{T}_{a'} \ln \frac{\mu_F^2}{2xp_a \cdot p_I}. \tag{B.4}$$

The regularized Altarelli-Parisi kernels  $P^{ab}(x)$  are listed in Eq. (B.9).

The factorization-scheme dependent terms are proportional to the initial-state insertion operators  $\mathbf{K}$ . For one initial-state hadron only, this operator reads

$$\mathbf{K}^{a,a'}(x) = \frac{\alpha_S}{2\pi} \left\{ \bar{K}^{aa'}(x) - K_{F.S.}^{aa'}(x) + \delta^{aa'} \sum_i \frac{\gamma_i \mathbf{T}_i \cdot \mathbf{T}'_a}{\mathbf{T}_i^2} \left[ \left( \frac{1}{1-x} \right)_+ + \delta(1-x) \right] \right\}, \quad (B.5)$$

with the functions  $K_{F.S.}^{aa'}(x)$  and  $\bar{K}^{aa'}(x)$  given below, cf. Eqs. (B.10) and (B.12), and with the  $\gamma_i$  listed in Eq. (B.7). Note that the subscript “F.S.” denotes the factorization scheme. For two initial state partons, the initial-state insertion operator is given by Eq. (5.55),

$$\begin{aligned} \mathbf{K}^{a,a'}(x) = & \frac{\alpha_S}{2\pi} \left\{ \bar{K}^{aa'}(x) - K_{F.S.}^{aa'}(x) \right. \\ & \left. + \delta^{aa'} \sum_i \mathbf{T}_i \cdot \mathbf{T}_a \frac{\gamma_i}{\mathbf{T}_i^2} \left[ \left( \frac{1}{1-x} \right)_+ + \delta(1-x) \right] - \frac{\mathbf{T}_b \cdot \mathbf{T}_{a'}}{\mathbf{T}_a^2} \tilde{K}^{a,a'}(x) \right\} \end{aligned} \quad (B.6)$$

with the functions  $\tilde{K}^{aa'}(x)$  given in Eq. (B.11).

The  $\gamma_I$  and  $K_I$  occurring in Eqs. (B.3) are related to integrals of the Altarelli-Parisi kernels listed below, Eq. (B.9), and read

$$\gamma_q = \gamma_{\bar{q}} = \frac{3}{2} C_F, \quad \gamma_g = \frac{11}{6} C_A - \frac{2}{3} T_R N_f \quad (B.7)$$

and

$$K_q = K_{\bar{q}} = \left( \frac{7}{2} - \frac{\pi^2}{6} \right) C_F, \quad K_g = \left( \frac{67}{18} - \frac{\pi^2}{6} \right) C_A - \frac{10}{9} T_R N_f, \quad (B.8)$$

respectively. The Altarelli-Parisi kernels emerging in the factorization-scale dependent terms of Eq. (B.4) are

$$\begin{aligned} P^{qg}(x) = P^{\bar{q}g}(x) &= C_F \frac{1 + (1-x)^2}{x} \\ P^{gq}(x) = P^{g\bar{q}}(x) &= T_R [x^2 + (1-x)^2] \\ P^{qq}(x) = P^{\bar{q}\bar{q}}(x) &= C_F \left( \frac{1+x^2}{1-x} \right)_+ \\ P^{gg}(x) &= 2C_A \left[ \left( \frac{1}{1-x} \right)_+ + \frac{1-x}{x} - 1 + x(1-x) \right] + \delta(1-x) \left[ \frac{11}{6} C_A - \frac{2}{3} T_R N_f \right]. \end{aligned} \quad (B.9)$$

The functions  $\bar{K}^{ab}(x)$  are explicitly given as

$$\begin{aligned}
\bar{K}^{q\bar{q}}(x) &= \bar{K}^{\bar{q}q}(x) = 0, \\
\bar{K}^{qg}(x) &= \bar{K}^{\bar{q}g}(x) = P^{qg}(x) \ln \frac{1-x}{x} + C_F x, \\
\bar{K}^{gq}(x) &= \bar{K}^{g\bar{q}}(x) = P^{gq}(x) \ln \frac{1-x}{x} + 2T_R x(1-x), \\
\bar{K}^{qq}(x) &= \bar{K}^{\bar{q}\bar{q}}(x) = C_F \left[ \left( \frac{2}{1-x} \ln \frac{1-x}{x} \right)_+ - (1+x) \ln \frac{1-x}{x} + (1-x) \right] \\
&\quad - \delta(1-x)(5-\pi^2)C_F, \\
\bar{K}^{gg}(x) &= 2C_A \left[ \left( \frac{1}{1-x} \ln \frac{1-x}{x} \right)_+ + \ln \frac{1-x}{x} \left( \frac{1-x}{x} - 1 + x(1-x) \right) \right] \\
&\quad - \delta(1-x) \left[ \left( \frac{50}{9} - \pi^2 \right) C_A - \frac{16}{9} T_R N_f \right], \tag{B.10}
\end{aligned}$$

whereas the functions  $\tilde{K}^{ab}(x)$  read

$$\begin{aligned}
\tilde{K}^{q\bar{q}}(x) &= \tilde{K}^{\bar{q}q}(x) = 0, \\
\tilde{K}^{qg}(x) &= \tilde{K}^{\bar{q}g}(x) = P^{qg}(x) \ln(1-x), \\
\tilde{K}^{gq}(x) &= \tilde{K}^{g\bar{q}}(x) = P^{gq}(x) \ln(1-x), \\
\tilde{K}^{qq}(x) &= \tilde{K}^{\bar{q}\bar{q}}(x) = C_F \left[ \left( \frac{2}{1-x} \ln(1-x) \right)_+ - \frac{\pi^2}{3} \delta(1-x) - (1+x) \ln(1-x) \right], \\
\tilde{K}^{gg}(x) &= C_A \left[ \left( \frac{2}{1-x} \ln(1-x) \right)_+ - \frac{\pi^2}{3} \delta(1-x) \right. \\
&\quad \left. + 2 \left( \frac{1-x}{x} - 1 + x(1-x) \right) \ln(1-x) \right]. \tag{B.11}
\end{aligned}$$

Finally, the factorization-scheme dependent terms are given through

$$\begin{aligned}
K_{\bar{M}S}^{ab}(x) &= 0, \\
K_{\text{DIS}}^{q\bar{q}} = K_{\text{DIS}}^{\bar{q}q} &= C_F \left[ \frac{1+x^2}{1-x} \left( \ln \frac{1-x}{x} - \frac{3}{4} \right) + \frac{9+5x}{4} \right]_+, \\
K_{\text{DIS}}^{gq} = K_{\text{DIS}}^{g\bar{q}} &= T_R \left[ (x^2 + (1-x)^2) \ln \frac{1-x}{x} + 8x(1-x) - 1 \right], \\
K_{\text{DIS}}^{qg} = K_{\text{DIS}}^{\bar{q}g} &= -K_{\text{DIS}}^{qg}, \\
K_{\text{DIS}}^{gg} &= -2N_f K_{\text{DIS}}^{qg}, \\
K_{\text{DIS}}^{\bar{q}q} = K_{\text{DIS}}^{q\bar{q}} &= 0. \tag{B.12}
\end{aligned}$$



# Appendix C Relations for the evaluation of dual integrals

## C.1 Feynman parameterization

Feynman parameterization usually refers to the following replacement

$$\frac{1}{A_1 \cdots A_n} = (n-1)! \int_0^1 dx_1 \cdots \int_0^1 dx_n \delta \left( 1 - \sum_{i=1}^n x_i \right) \frac{1}{(\sum_{i=1}^n x_i A_i)^n}. \quad (\text{C.1})$$

This can be used to solve integrals of the structure

$$\begin{aligned} \int_0^\infty dz \prod_{i=1}^n \frac{1}{z + a_i} &= (n-1)! \int_0^\infty dz \int_0^1 dx_1 \cdots \int_0^1 dx_n \delta \left( 1 - \sum_{i=1}^n x_i \right) \frac{1}{(z + \sum_{i=1}^n x_i a_i)^n} \\ &= (n-2)! \int_0^1 dx_1 \cdots \int_0^1 dx_n \delta \left( 1 - \sum_{i=1}^n x_i \right) \frac{1}{(\sum_{i=1}^n x_i a_i)^{n-1}}. \end{aligned} \quad (\text{C.2})$$

## C.2 Integrals

The following definite integrals have been used to evaluate dual integrals:

$$\int_0^1 dx (1-x)^{-\epsilon-1} (a+bx)^\epsilon = -\frac{1}{\epsilon} a^\epsilon {}_2F_1 \left( 1, -\epsilon; 1-\epsilon; -\frac{b}{a} \right), \quad (\text{C.3})$$

$$\int_0^\infty dx \frac{x^{\mu-1}}{(ax+b)^\nu} = a^{-\mu} b^{\mu-\nu} \frac{\Gamma(\mu)\Gamma(\nu-\mu)}{\Gamma(\nu)}. \quad (\text{C.4})$$

### C.3 Hypergeometric functions

The hypergeometric function e.g. in Eq. (6.18) has the following integral representations:

$$\begin{aligned} {}_2F_1(1, -\epsilon; 1 - \epsilon; z) &= -\epsilon \int_0^1 dt \frac{(1-t)^{-1-\epsilon}}{(1-zt)^{-\epsilon}} = -\epsilon \int_0^1 dt \frac{(t)^{-1-\epsilon}}{1-zt} \\ &= 1 - \epsilon z \int_0^1 dt \frac{(t)^{-\epsilon}}{1-zt} \end{aligned} \quad (\text{C.5})$$

$$= 1 + \epsilon \ln(1-z) + \epsilon^2 \int_0^1 dt \quad (\text{C.6})$$

The expansion in powers of  $\epsilon$  is given by:

$${}_2F_1(1, -\epsilon; 1 - \epsilon; z) = 1 + \epsilon \ln(1-z) - \epsilon^2 \left[ \text{Li}_2(z) + \sum_{k=1}^{+\infty} \epsilon^k \text{Li}_{k+2}(z) \right], \quad (\text{C.7})$$

where the dilogarithm function  $\text{Li}_2$  is

$$\text{Li}_2(x) \equiv - \int_0^1 \frac{dt}{t} \ln(1-xt), \quad (\text{C.8})$$

and the polylogarithms  $\text{Li}_{k+1}(x)$  (with  $k = 2, 3, \dots$ ) are defined by

$$\text{Li}_{k+1}(x) \equiv \frac{(-1)^k}{(k-1)!} \int_0^1 \frac{dt}{t} (\ln t)^{k-1} \ln(1-xt). \quad (\text{C.9})$$

Other hypergeometric functions required for dual and associated integrals can be obtained from the following relations:

$${}_2F_1(\alpha, \beta; \gamma; z) = (1-z)^{-\alpha} {}_2F_1\left(\alpha, \gamma - \beta; \gamma; \frac{z}{z-1}\right), \quad (\text{C.10})$$

$$= (1-z)^{-\beta} {}_2F_1\left(\beta, \gamma - \alpha; \gamma; \frac{z}{z-1}\right), \quad (\text{C.11})$$

$$= (1-z)^{\gamma - \alpha - \beta} {}_2F_1(\gamma - \alpha, \gamma - \beta; \gamma; z). \quad (\text{C.12})$$

In particular, the following functions are used:

$${}_2F_1(1, -\epsilon; 1 - 2\epsilon; z) = 1 + \epsilon \ln(1-z) + \frac{\epsilon^2}{2} (\ln^2(1-z) + 4\text{Li}_2(z)) + \mathcal{O}(\epsilon^3), \quad (\text{C.13})$$

$${}_2F_1(1, -\epsilon; -2\epsilon; z) = \frac{1}{1-z} \left( 1 - \frac{z}{2} {}_2F_1(1, -\epsilon; 1 - 2\epsilon; z) \right). \quad (\text{C.14})$$

# Bibliography

- [1] J. D. Bjorken and S. D. Drell, “Relativistic quantum field theory. (german translation),” Bibliograph.Inst./mannheim 1967, 409 P.(B.i.- hochschultaschenbuecher, Band 101).
- [2] C. Itzykson and J. B. Zuber, *Quantum Field Theory*. New York, USA: McGraw-Hill, 1980.
- [3] F. Halzen and A. D. Martin, *Quarks and leptons: An introductory course in modern particle physics*. New York, USA: John Wiley & Sons, 1984.
- [4] D. Y. Bardin and G. Passarino, “The standard model in the making: Precision study of the electroweak interactions,” Oxford, UK: Clarendon (1999) 685 p.
- [5] M. Bohm, A. Denner, and H. Joos, “Gauge theories of the strong and electroweak interaction,” Stuttgart, Germany: Teubner (2001) 784 p.
- [6] T. Sjostrand, “High-energy physics event generation with pythia 5.7 and jetset 7.4,” *Comput. Phys. Commun.*, vol. 82, pp. 74–90, 1994.
- [7] T. Sjostrand, S. Mrenna, and P. Skands, “Pythia 6.4 physics and manual,” *JHEP*, vol. 05, p. 026, 2006.
- [8] G. Corcella *et al.*, “Herwig 6: An event generator for hadron emission reactions with interfering gluons (including supersymmetric processes),” *JHEP*, vol. 01, p. 010, 2001.
- [9] G. Corcella *et al.*, “Herwig 6.5 release note,” 2002.
- [10] S. Gieseke, A. Ribon, M. H. Seymour, P. Stephens, and B. Webber, “Herwig++ 1.0: An event generator for e+ e- annihilation,” *JHEP*, vol. 02, p. 005, 2004.
- [11] S. Catani, F. Krauss, R. Kuhn, and B. R. Webber, “Qcd matrix elements + parton showers,” *JHEP*, vol. 111, p. 063, 2001.
- [12] F. Krauss, “Matrix elements and parton showers in hadronic interactions,” *JHEP*, vol. 0208, p. 015, 2002.

- [13] L. Lonnblad, “Correcting the colour-dipole cascade model with fixed order matrix elements,” *JHEP*, vol. 05, p. 046, 2002.
- [14] M. L. Mangano, M. Moretti, and R. Pittau, “Multijet matrix elements and shower evolution in hadronic collisions:  $W b$  anti- $b$  + (n)jets as a case study,” *Nucl. Phys.*, vol. B632, pp. 343–362, 2002.
- [15] M. L. Mangano, M. Moretti, F. Piccinini, and M. Treccani, “Matching matrix elements and shower evolution for top-quark production in hadronic collisions,” *JHEP*, vol. 01, p. 013, 2007.
- [16] S. Frixione and B. R. Webber, “The mc@nlo 3.3 event generator,” 2006.
- [17] S. Frixione, P. Nason, and C. Oleari, “Matching nlo qcd computations with parton shower simulations: the powheg method,” 2007.
- [18] S. Catani, Y. L. Dokshitzer, M. H. Seymour, and B. R. Webber, “Longitudinally invariant  $k(t)$  clustering algorithms for hadron hadron collisions,” *Nucl. Phys.*, vol. B406, pp. 187–224, 1993.
- [19] M. L. Mangano, M. Moretti, F. Piccinini, R. Pittau, and A. D. Polosa, “Alpgen, a generator for hard multiparton processes in hadronic collisions,” *JHEP*, vol. 07, p. 001, 2003.
- [20] F. Krauss, R. Kuhn, and G. Soff, “Amegic++ 1.0: A matrix element generator in c++,” *JHEP*, vol. 02, p. 044, 2002.
- [21] A. Kanaki and C. G. Papadopoulos, “Helac: a package to compute electroweak helicity amplitudes,” *Comput. Phys. Commun.*, vol. 132, pp. 306–315, 2000.
- [22] C. G. Papadopoulos, “Phegas: A phase space generator for automatic cross-section computation,” *Comput. Phys. Commun.*, vol. 137, pp. 247–254, 2001.
- [23] F. Maltoni and T. Stelzer, “Madevent: Automatic event generation with madgraph,” *JHEP*, vol. 02, p. 027, 2003.
- [24] M. Moretti, T. Ohl, and J. Reuter, “O’mega: An optimizing matrix element generator,” 2001.
- [25] J. Campbell and R. K. Ellis, “Next-to-leading order corrections to  $w + 2$ jet and  $z + 2$ jet production at hadron colliders,” *Phys. Rev.*, vol. D65, p. 113007, 2002.
- [26] Z. Nagy, “Next-to-leading order calculation of three-jet observables in hadron hadron collision,” *Phys. Rev.*, vol. D68, p. 094002, 2003.

[27] Z. Nagy and Z. Trocsanyi, “Three-jet event-shapes in lepton proton scattering at next-to-leading order accuracy,” *Phys. Lett.*, vol. B634, pp. 498–503, 2006.

[28] J. M. Campbell, R. K. Ellis, and G. Zanderighi, “Next-to-leading order higgs + 2 jet production via gluon fusion,” *JHEP*, vol. 10, p. 028, 2006.

[29] S. Dittmaier, P. Uwer, and S. Weinzierl, “Nlo qcd corrections to t anti-t + jet production at hadron colliders,” *Phys. Rev. Lett.*, vol. 98, p. 262002, 2007.

[30] T. Binoth, G. Heinrich, T. Gehrmann, and P. Mastrolia, “Six-photon amplitudes,” *Phys. Lett.*, vol. B649, pp. 422–426, 2007.

[31] T. Gleisberg, S. Höche, F. Krauss, A. Schälicke, S. Schumann, and J. Winter, “Sherpa 1.α, a proof-of-concept version,” *JHEP*, vol. 02, p. 056, 2004.

[32] S. Catani, Y. L. Dokshitzer, M. Olsson, G. Turnock, and B. R. Webber, “New clustering algorithm for multi - jet cross-sections in e+ e- annihilation,” *Phys. Lett.*, vol. B269, pp. 432–438, 1991.

[33] S. Catani, Y. L. Dokshitzer, and B. R. Webber, “The k-perpendicular clustering algorithm for jets in deep inelastic scattering and hadron collisions,” *Phys. Lett.*, vol. B285, pp. 291–299, 1992.

[34] F. Krauss, A. Schälicke, S. Schumann, and G. Soff, “Simulating W / Z + jets production at the Tevatron,” 2004.

[35] F. Krauss, A. Schalicke, S. Schumann, and G. Soff, “Simulating w / z + jets production at the cern lhc,” *Phys. Rev.*, vol. D72, p. 054017, 2005.

[36] T. Gleisberg, F. Krauss, A. Schalicke, S. Schumann, and J.-C. Winter, “Studying w+ w- production at the fermilab tevatron with sherpa,” *Phys. Rev.*, vol. D72, p. 034028, 2005.

[37] S. Hoche *et al.*, “Matching parton showers and matrix elements,” 2006.

[38] R. Kuhn, F. Krauss, B. Ivanyi, and G. Soff, “Apacic++ 1.0: A parton cascade in c++,” *Comput. Phys. Commun.*, vol. 134, pp. 223–266, 2001.

[39] J.-C. Winter, F. Krauss, and G. Soff, “A modified cluster-hadronization model,” *Eur. Phys. J.*, vol. C36, pp. 381–395, 2004.

[40] F. Cachazo, P. Svrček, and E. Witten, “Mhv vertices and tree amplitudes in gauge theory,” *JHEP*, vol. 09, p. 006, 2004.

[41] F. A. Berends and W. T. Giele, “Recursive calculations for processes with n gluons,” *Nucl. Phys.*, vol. B306, p. 759, 1988.

[42] S. Catani and M. H. Seymour, “A general algorithm for calculating jet cross sections in nlo qcd,” *Nucl. Phys.*, vol. B485, pp. 291–419, 1997.

[43] C. Duhr, S. Höche, and F. Maltoni, “Color-dressed recursive relations for multi-parton amplitudes,” *JHEP*, vol. 08, p. 062, 2006.

[44] T. Gleisberg, S. Schumann, and F. Krauss in preparation.

[45] T. Gleisberg, F. Krauss, K. T. Matchev, A. Schälicke, S. Schumann, and G. Soff, “Helicity formalism for spin-2 particles,” *JHEP*, vol. 09, p. 001, 2003.

[46] K. Hagiwara *et al.*, “Supersymmetry simulations with off-shell effects for lhc and ilc,” *Phys. Rev.*, vol. D73, p. 055005, 2006.

[47] T. Gleisberg, F. Krauss, C. G. Papadopoulos, A. Schälicke, and S. Schumann, “Cross sections for multi-particle final states at a linear collider,” *Eur. Phys. J.*, vol. C34, pp. 173–180, 2004.

[48] P. De Causmaecker, R. Gastmans, W. Troost, and T. T. Wu, “Helicity amplitudes for massless qed,” *Phys. Lett.*, vol. B105, p. 215, 1981.

[49] R. Kleiss and W. J. Stirling, “Spinor techniques for calculating  $p$  anti- $p \rightarrow w^- / z^0 + jets$ ,” *Nucl. Phys.*, vol. B262, pp. 235–262, 1985.

[50] A. Ballestrero, E. Maina, and S. Moretti, “Heavy quarks and leptons at e+ e- colliders,” *Nucl. Phys.*, vol. B415, pp. 265–292, 1994.

[51] R. Kuhn, “Event generation at lepton colliders,” PhD Thesis.

[52] T. Gleisberg, “Helicity formalism for exotic physics scenarios at cillider experiments,” Master Thesis.

[53] F. Wilczek, “Decays of heavy vector mesons into higgs particles,” *Phys. Rev. Lett.*, vol. 39, p. 1304, 1977.

[54] S. Dawson, “Radiative corrections to higgs boson production,” *Nucl. Phys.*, vol. B359, pp. 283–300, 1991.

[55] L. Bergstrom and G. Hulth, “Induced higgs couplings to neutral bosons in e+ e- collisions,” *Nucl. Phys.*, vol. B259, p. 137, 1985.

[56] K. Hagiwara, R. D. Peccei, D. Zeppenfeld, and K. Hikasa, “Probing the weak boson sector in  $e+ e^- \rightarrow w^+ w^-$ ,” *Nucl. Phys.*, vol. B282, p. 253, 1987.

[57] T. Appelquist and C. W. Bernard, “Strongly interacting higgs bosons,” *Phys. Rev.*, vol. D22, p. 200, 1980.

[58] A. C. Longhitano, “Low-energy impact of a heavy higgs boson sector,” *Nucl. Phys.*, vol. B188, p. 118, 1981.

[59] T. Appelquist and G.-H. Wu, “The electroweak chiral lagrangian and new precision measurements,” *Phys. Rev.*, vol. D48, pp. 3235–3241, 1993.

[60] F. Gangemi, G. Montagna, M. Moretti, O. Nicrosini, and F. Piccinini, “Electroweak physics in six-fermion final states at future e+ e- colliders,” 1999.

[61] M. L. Mangano, S. J. Parke, and Z. Xu, “Duality and multi-gluon scattering,” *Nucl. Phys.*, vol. B298, p. 653, 1988.

[62] V. Del Duca, L. J. Dixon, and F. Maltoni, “New color decompositions for gauge amplitudes at tree and loop level,” *Nucl. Phys.*, vol. B571, pp. 51–70, 2000.

[63] V. del Duca, A. Frizzo, and F. Maltoni, “Factorization of tree qcd amplitudes in the high-energy limit and in the collinear limit,” *Nucl. Phys.*, vol. B568, pp. 211–262, 2000.

[64] F. Maltoni, K. Paul, T. Stelzer, and S. Willenbrock, “Color-flow decomposition of qcd amplitudes,” *Phys. Rev.*, vol. D67, p. 014026, 2003.

[65] S. J. Parke and T. R. Taylor, “An amplitude for n gluon scattering,” *Phys. Rev. Lett.*, vol. 56, p. 2459, 1986.

[66] E. Witten, “Perturbative gauge theory as a string theory in twistor space,” *Commun. Math. Phys.*, vol. 252, pp. 189–258, 2004.

[67] G. Georgiou, E. W. N. Glover, and V. V. Khoze, “Non-mhv tree amplitudes in gauge theory,” *JHEP*, vol. 07, p. 048, 2004.

[68] V. V. Khoze, “Gauge theory amplitudes, scalar graphs and twistor space,” 2004.

[69] L. J. Dixon, E. W. N. Glover, and V. V. Khoze, “Mhv rules for higgs plus multi-gluon amplitudes,” *JHEP*, vol. 12, p. 015, 2004.

[70] C. G. Papadopoulos and M. Worek, “Multi-parton cross sections at hadron colliders,” *Eur. Phys. J.*, vol. C50, pp. 843–856, 2007.

[71] M. Dinsdale, M. Ternick, and S. Weinzierl, “A comparison of efficient methods for the computation of born gluon amplitudes,” *JHEP*, vol. 03, p. 056, 2006.

[72] L. J. Dixon, “Calculating scattering amplitudes efficiently,” 1996.

[73] R. Kleiss, W. J. Stirling, and S. D. Ellis, “A new monte carlo treatment of multiparticle phase space at high-energies,” *Comput. Phys. Commun.*, vol. 40, p. 359, 1986.

[74] G. P. Lepage, “Vegas - an adaptive multi-dimensional integration program,” CLNS-80/447.

[75] S. Jadach, “Foam: A general purpose monte carlo cellular algorithm,” Prepared for 31st International Conference on High Energy Physics (ICHEP 2002), Amsterdam, The Netherlands, 24-31 Jul 2002.

[76] A. van Hameren, “Adaptive channels for data analysis and importance sampling,” 2003.

[77] R. Kleiss and R. Pittau, “Weight optimization in multichannel monte carlo,” *Comput. Phys. Commun.*, vol. 83, pp. 141–146, 1994.

[78] T. Ohl, “Vegas revisited: Adaptive monte carlo integration beyond factorization,” *Comput. Phys. Commun.*, vol. 120, pp. 13–19, 1999.

[79] J. G. M. Kuijf, “Multiparton production at hadron colliders,” RX-1335 (LEIDEN).

[80] P. D. Draggiotis, A. van Hameren, and R. Kleiss, “Sarge: An algorithm for generating qcd antennas,” *Phys. Lett.*, vol. B483, pp. 124–130, 2000.

[81] A. van Hameren and C. G. Papadopoulos, “A hierarchical phase space generator for qcd antenna structures,” *Eur. Phys. J.*, vol. C25, pp. 563–574, 2002.

[82] T. Gleisberg and S. Höche in preparation.

[83] J. Pumplin *et al.*, “New generation of parton distributions with uncertainties from global qcd analysis,” *JHEP*, vol. 07, p. 012, 2002.

[84] F. Caravaglios, M. L. Mangano, M. Moretti, and R. Pittau, “A new approach to multi-jet calculations in hadron collisions,” *Nucl. Phys.*, vol. B539, pp. 215–232, 1999.

[85] P. Pascual and R. Tarrach, “Qcd: Renormalization for the practitioner,” *Lect. Notes Phys.*, vol. 194, pp. 1–277, 1984.

[86] R. V. Harlander and W. B. Kilgore, “Next-to-next-to-leading order higgs production at hadron colliders,” *Phys. Rev. Lett.*, vol. 88, p. 201801, 2002.

[87] C. Anastasiou and K. Melnikov, “Higgs boson production at hadron colliders in nnlo qcd,” *Nucl. Phys.*, vol. B646, pp. 220–256, 2002.

[88] G. Passarino and M. J. G. Veltman, “One loop corrections for e+ e- annihilation into mu+ mu- in the weinberg model,” *Nucl. Phys.*, vol. B160, p. 151, 1979.

[89] T. Kinoshita, “Mass singularities of feynman amplitudes,” *J. Math. Phys.*, vol. 3, pp. 650–677, 1962.

[90] T. D. Lee and M. Nauenberg, “Degenerate systems and mass singularities,” *Phys. Rev.*, vol. 133, pp. B1549–B1562, 1964.

[91] H. Baer, J. Ohnemus, and J. F. Owens, “A next-to-leading logarithm calculation of jet photoproduction,” *Phys. Rev.*, vol. D40, p. 2844, 1989.

[92] B. W. Harris and J. F. Owens, “The two cutoff phase space slicing method,” *Phys. Rev.*, vol. D65, p. 094032, 2002.

[93] K. Fabricius, I. Schmitt, G. Kramer, and G. Schierholz, “Higher order perturbative qcd calculation of jet cross- sections in e+ e- annihilation,” *Zeit. Phys.*, vol. C11, p. 315, 1981.

[94] G. Kramer and B. Lampe, “Jet cross-sections in e+ e- annihilation,” *Fortschr. Phys.*, vol. 37, p. 161, 1989.

[95] W. T. Giele and E. W. N. Glover, “Higher order corrections to jet cross-sections in e+ e- annihilation,” *Phys. Rev.*, vol. D46, pp. 1980–2010, 1992.

[96] W. T. Giele, E. W. N. Glover, and D. A. Kosower, “Higher order corrections to jet cross-sections in hadron colliders,” *Nucl. Phys.*, vol. B403, pp. 633–670, 1993.

[97] R. K. Ellis, D. A. Ross, and A. E. Terrano, “The perturbative calculation of jet structure in e+ e- annihilation,” *Nucl. Phys.*, vol. B178, p. 421, 1981.

[98] S. Frixione, Z. Kunszt, and A. Signer, “Three-jet cross sections to next-to-leading order,” *Nucl. Phys.*, vol. B467, pp. 399–442, 1996.

[99] D. A. Kosower, “Antenna factorization of gauge-theory amplitudes,” *Phys. Rev.*, vol. D57, pp. 5410–5416, 1998.

[100] J. M. Campbell, M. A. Cullen, and E. W. N. Glover, “Four jet event shapes in electron positron annihilation,” *Eur. Phys. J.*, vol. C9, pp. 245–265, 1999.

[101] S. Catani, S. Dittmaier, M. H. Seymour, and Z. Trocsanyi, “The dipole formalism for next-to-leading order qcd calculations with massive partons,” *Nucl. Phys.*, vol. B627, pp. 189–265, 2002.

[102] D. A. Kosower, “Antenna factorization in strongly-ordered limits,” *Phys. Rev.*, vol. D71, p. 045016, 2005.

[103] G. Somogyi and Z. Trocsanyi, “A new subtraction scheme for computing qcd jet cross sections at next-to-leading order accuracy,” 2006.

[104] G. C. Blazey *et al.*, “Run ii jet physics,” 2000.

- [105] P. Abreu *et al.*, “Tuning and test of fragmentation models based on identified particles and precision event shape data,” *Z. Phys.*, vol. C73, pp. 11–60, 1996.
- [106] G. ’t Hooft and M. J. G. Veltman, “Scalar one loop integrals,” *Nucl. Phys.*, vol. B153, pp. 365–401, 1979.
- [107] J. Fleischer, F. Jegerlehner, and O. V. Tarasov, “Algebraic reduction of one-loop feynman graph amplitudes,” *Nucl. Phys.*, vol. B566, pp. 423–440, 2000.
- [108] Z. Bern, L. J. Dixon, and D. A. Kosower, “Dimensionally regulated pentagon integrals,” *Nucl. Phys.*, vol. B412, pp. 751–816, 1994.
- [109] T. Binoth, J. P. Guillet, and G. Heinrich, “Reduction formalism for dimensionally regulated one-loop n- point integrals,” *Nucl. Phys.*, vol. B572, pp. 361–386, 2000.
- [110] G. Duplancic and B. Nizic, “Reduction method for dimensionally regulated one-loop n- point feynman integrals,” *Eur. Phys. J.*, vol. C35, pp. 105–118, 2004.
- [111] W. T. Giele and E. W. N. Glover, “A calculational formalism for one-loop integrals,” *JHEP*, vol. 04, p. 029, 2004.
- [112] W. Giele, E. W. N. Glover, and G. Zanderighi, “Numerical evaluation of one-loop diagrams near exceptional momentum configurations,” *Nucl. Phys. Proc. Suppl.*, vol. 135, pp. 275–279, 2004.
- [113] A. Ferroglio, M. Passera, G. Passarino, and S. Uccirati, “All-purpose numerical evaluation of one-loop multi-leg feynman diagrams,” *Nucl. Phys.*, vol. B650, pp. 162–228, 2003.
- [114] Z. Nagy and D. E. Soper, “General subtraction method for numerical calculation of one-loop qcd matrix elements,” *JHEP*, vol. 09, p. 055, 2003.
- [115] T. Binoth, G. Heinrich, and N. Kauer, “A numerical evaluation of the scalar hexagon integral in the physical region,” *Nucl. Phys.*, vol. B654, pp. 277–300, 2003.
- [116] Y. Kurihara and T. Kaneko, “Numerical contour integration for loop integrals,” *Comput. Phys. Commun.*, vol. 174, pp. 530–539, 2006.
- [117] E. de Doncker *et al.*, “Loop integration results using numerical extrapolation for a non-scalar integral,” *Nucl. Instrum. Meth.*, vol. A534, pp. 269–273, 2004.
- [118] C. Anastasiou and A. Daleo, “Numerical evaluation of loop integrals,” *JHEP*, vol. 10, p. 031, 2006.
- [119] R. K. Ellis, W. T. Giele, and G. Zanderighi, “Semi-numerical evaluation of one-loop corrections,” *Phys. Rev.*, vol. D73, p. 014027, 2006.

---

- [120] A. Denner and S. Dittmaier, “Reduction schemes for one-loop tensor integrals,” *Nucl. Phys.*, vol. B734, pp. 62–115, 2006.
- [121] F. del Aguila and R. Pittau, “Recursive numerical calculus of one-loop tensor integrals,” *JHEP*, vol. 07, p. 017, 2004.
- [122] A. van Hameren, J. Vollinga, and S. Weinzierl, “Automated computation of one-loop integrals in massless theories,” *Eur. Phys. J.*, vol. C41, pp. 361–375, 2005.
- [123] G. Ossola, C. G. Papadopoulos, and R. Pittau, “Reducing full one-loop amplitudes to scalar integrals at the integrand level,” *Nucl. Phys.*, vol. B763, pp. 147–169, 2007.
- [124] R. K. Ellis, W. T. Giele, and Z. Kunszt, “A numerical unitarity formalism for evaluating one-loop amplitudes,” 2007.
- [125] S. Catani, T. Gleisberg, F. Krauss, G. Rodrigo, and J. Winter in preparation.
- [126] G. Duplancic and B. Nizic, “Ir finite one-loop box scalar integral with massless internal lines,” *Eur. Phys. J.*, vol. C24, pp. 385–391, 2002.
- [127] K. Hagiwara and D. Zeppenfeld, “Helicity amplitudes for heavy lepton production in  $e^+e^-$  annihilation,” *Nucl. Phys.*, vol. B274, p. 1, 1986.
- [128] S. Dittmaier, “Weyl-van-der-waerden formalism for helicity amplitudes of massive particles,” *Phys. Rev.*, vol. D59, p. 016007, 1999.



# List of publications

## Journals & eprints

1. T. Gleisberg and F. Krauss,  
“*Automating dipole subtraction for QCD NLO calculations*”,  
arXiv:0709.2881 [hep-ph].
2. T. Gleisberg, F. Krauss, A. Schälicke, S. Schumann, G. Soff and J. Winter,  
“*Studying W+ W- production at the Fermilab Tevatron with SHERPA*”,  
Phys.Rev.D72:034028,2005.
3. T. Gleisberg, F. Krauss, C.G. Papadopoulos, A. Schälicke and S. Schumann,  
“*Cross sections for multi-particle final states at a linear collider*”,  
Eur.Phys.J.C34:173-180,2004.
4. T. Gleisberg, S. Höche, F. Krauss, A. Schälicke, S. Schumann and J. Winter,  
“*SHERPA 1.alpha, a proof-of-concept version*”,  
JHEP 0402:056,2004.
5. T. Gleisberg, F. Krauss, K. T. Matchev, A. Schälicke, S. Schumann and G. Soff,  
“*Helicity formalism for spin-2 particles*”,  
JHEP 0309:001,2003.

## Conference proceedings

1. T. Gleisberg, S. Hoeche, F. Krauss, R. Matyszkiewicz, M. Schoenherr, S. Schumann,  
F. Siegert and J. Winter,  
“*New trends in modern event generators*”,  
arXiv:0705.4648 [hep-ph].
2. S. Alekhin, S. *et al.*  
“*HERA and the LHC - A workshop on the implications of HERA for LHC physics.*”,  
hep-ph/0601013 and hep-ph/0601013.
3. T. Gleisberg, S. Höche, F. Krauss, A. Schälicke, S. Schumann and J. Winter,  
“*Event generator for the LHC*”,  
Nucl.Instrum.Meth.A559:242-245,2006 (hep-ph/0508315).

4. T. Gleisberg, S. Höche, F. Krauss, A. Schälicke, S. Schumann, J. Winter and G. Soff,  
“*Predictions for multi-particle final states with SHERPA*”,  
Czech.J.Phys.55:B529-B536,2005 (hep-ph/0409122).
5. T. Gleisberg, S. Höche, F. Krauss, A. Schälicke, S. Schumann, J. Winter and G. Soff,  
“*Towards a fragmentation model for Sherpa*”,  
hep-ph/0408245.
6. T. Gleisberg, S. Höche, F. Krauss, A. Schälicke, S. Schumann, J. Winter and G. Soff,  
“*New tools for automatic cross section calculation*”,  
hep-ph/0407366
7. T. Gleisberg, S. Höche, F. Krauss, A. Schälicke, S. Schumann, J. Winter and G. Soff,  
“*Monte Carlo models at the LHC*”,  
hep-ph/0407365.
8. A. Schälicke, T. Gleisberg, S. Höche, J. Winter, F. Krauss, S. Schumann and G. Soff,  
“*Event generator for particle production in high-energy collisions*”,  
Prog.Part.Nucl.Phys.53:329-338,2004 (hep-ph/0311270).

# Acknowledgements

Firstly, I would like to thank Frank Krauss for the continuous support and inspiration during the whole period of this work. I am very grateful for all the opportunities and contacts to the right people that he made possible.

I am much obliged to Michael Kobel for his support and supervision in the final phase of my PhD period.

My gratitude goes to all the current and former members of the Sherpa group, Jan Winter, Steffen Schumann, Stefan Höche, Andreas Schälicke, Frank Siegert and Marek Schönherr for the enjoyable and successful collaboration, and all we shared for work and beyond.

I would like to thank Mike Seymour, Stefano Catani, German Rodrigo, Tancredi Carli and Michelangelo Mangano for the fruitful collaboration and the help and support they gave my during my PhD period.

Thanks are due to the CERN theory group for the kind hospitality during the 11 months of my EST period. I thank the Marie Curie Fellowship program for funding this.

I express my gratitude to the members of the Institute for Theoretical Physics, especially the secretary Gundula Schädlich.

Finally, I thank my friends, my parents and especially my girlfriend Katharina for their encouragement, understanding and love.



## **Versicherung**

Hiermit versichere ich, daß ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Die vorliegende Arbeit wurde am Institut für Theoretische Physik der Technischen Universität Dresden unter wissenschaftlicher Betreuung von Herrn Prof. Dr. Michael Kobel und Herrn Dr. Frank Krauss angefertigt.

Es haben keine früheren erfolglosen Promotionsverfahren stattgefunden.

Hiermit erkenne ich die Promotionsordnung der Fakultät Mathematik und Naturwissenschaften der Technischen Universität Dresden vom 20. März 2000 an.

Dresden, den 08.10.2007

Tanju Gleisberg