Feynman loop integrals and their automatic computer-aided evaluation

Ph.D thesis
at the Physics Department,
Johannes Gutenberg-Universität Mainz

Đỗ Hoàng Sơn
born in Lào Cai, Việt Nam

Mainz

May 27, 2003

"Chữ Tâm kia mới bằng ba chữ Tài"

–Nguyễn Du–

Acknowledgements

There is a Vietnamese proverb: “Không thầy đố mày làm nên”. It means, without the education from my professors I would never become a useful man. To accomplish this thesis, I owe so much credit to my professors, my colleagues, my friends and my family.

First of all, I would like to express my deep gratitude to Prof. Dr. Jürgen Körner for his assistance as mentor of my thesis and to Prof. Dr. Dirk Kreimer and Priv. Doz. Dr. Hubert Spiesberger as co-mentors. Without their kind encouragement, support, constructive guidance and also proofreading the manuscript I would not have been able to finish this thesis.

I would like to thank Christian Bauer, Dr. Lars Brücher, Dr. Andrei Davydychev, Dr. Ansgar Denner, Martin Fischer, Dr. Alexander Frink, Dr. Stefan Groote, Markus Knodel, Dr. Richard Kreckel, Dr. Alfonso Leyva, Dr. Chun Liu, and Dr. Marcus Mauser for interesting discussions, suggestions and collaborative work.

I will never forget the hospitality of the ThEP working group in Mainz, for that I would like to thank Mrs. Monique Engler, Mrs. Edda Fischer, Prof. Dr. Nikolaos Papadopoulos, Prof. Dr. Martin Reuter, Prof. Dr. Florian Scheck, Prof. Dr. Karl Schilcher, and my dear colleagues Astrid Bauer, Isabella Bierenbaum, Alexander Holfter, Dagmar Kubistin, Dr. Oliver Lauscher, Mikhail Rogal, Jens Vollinga and other members of the group.

In addition, I would like to thank the DFG and the Graduiertenkolleg ”Eichtheorien-experimentelle Tests und theoretische Grundlagen” for the financial support in the last four years.

The support from the open sources and free software community is of much benefit to my work, for that I want to thank Prof. Dr. Elise de Doncker and Dr. Laurentui Cucos for ParInt program; Dr. Stefan Weinzierl for Nestedsum library.

Deep in my heart, I would like to thank my beloved Prof. Dr. Hoang Ngoc Long and my respectable professors of the "Rencontré du Vietnam" Prof. Dr. Patrick Aurenche, Dr. Geneviève Bélanger, Dr. Nguyen Anh Ky, Dr. Dang Van Soa, Prof. Dr. Tran Minh Tam, Prof. Dr. Tran Thanh Van, Prof. Dr. Pham Xuan Yem for their interest in my progress. Their continuous encouragement is meaningful not only to my work but also to my life.

My thank is also directed to Prof. Dr. Nguyen Huu Chi, Prof. Dr. Hoang Dzung, Prof. Dr. Nguyen Nhat Khanh, Prof. Dr. Nguyen Quoc Khanh at the National university of Hochiminh City for their encouragement.

On the challenging road of my scientific life, I am happy and proud to have a big brother as Prof. Dr. Christian Schubert and great friends as Christophe Derycke, Dr. Nguyen Ky Hoang, Nguyen Van Minh, Dr. Haitham Zaraket. They are always with me to share happiness as well as disappointment.

I also want to thank my dear friend Trinh Anh Minh and Vietnamese community in Mainz for their help. Without their friendship, four year studying in Mainz would be terribly long for me.

I want to thank my parents, my sister and my whole family for their love and support in all respects.

Finally, I want to thank my wife for her support and especially for her present of love, our lovely daughter.

Contents

Chapter 1

Introduction

Calculating Feynman graph and Feynman integrals is a difficult and important task in theoretical high energy physics.

The techniques for doing specific one- or two-loop integrations by hand are well-known and up to two-loop order most of the relevant integrals have been calculated. However, to generalize the techniques to solve a class of arbitrary one- or two-loop integrals (including the general mass case) in a generic and systematic way is far more complicated and is thus an interesting topic to work on.

In the last two decades, there has been much progress in developing techniques to calculate one- and two-loop integrals with the help of computer programs. At the tree-level there already exist many program packages such as GRACE [1] and CompHEP [2]. At the one-loop level there exist packages such as XLOOPS [3], FeynArts, FeynCalc and FF [4, 5, 6, 7]. These packages can perform one-loop one-, two- and three-point tensor integrations in $4 - 2\epsilon$ dimensions. At the two-loop level, however, up to now, there only exist packages for two-point tensor integrals such as XLOOPS, TwoCalc [8] and FeynHiggs [9]. A comprehensive review of the field is presented in [10]. It is known that most of the techniques which use the Feynman parametrisation are limited to certain subclasses of integrals, for instance the rank of the tensor integrand is restricted or only specific mass configurations are allowed.

In this thesis, we will mainly discuss the implementation of a technique for calculating Feynman loop integrals in parallel and orthogonal spaces, which has been developed by D. Kreimer and then successfully implemented into the XLOOPS program package by his collaborators in Mainz [3, 11, 12, 13, 14, 15, 16]. The method opens a new way to solve the Feynman loop integrals systematically and automatically (by computer) without any restriction on parameters like the tensor degree or masses.

We emphasize that in this thesis, we are presenting a technique for calculating general Feynman one- and two-loop tensor integrals, i.e. integrals with arbitrary masses in the denominator (the masses can be zero, large, different or equal) and arbitrary tensor rank in the numerator.

The XLOOPS package was originally written in Maple VR3, a computer algebraic environment [17]. As discussed in [16], there are numerous technical issues of coding that have lead to a termination of developing XLOOPS in the Maple environment. A few of these are:

- Maple does not handle global and local variables well. This deficiency n of Maple may lead to unpredictable errors when a local variable is modified globally.
- Releases 3 to 5 of Maple do not handle expressions larger than 2^{16} terms in a sum. This limiting capacity of Maple is not sufficient for two-loop problems where larger expressions are needed.
- The non-compatibility of subsequent Maple releases means XLOOPS has to provide different versions for every update of Maple. Such a maintenance effort is unreasonable.

In brief, Maple is not a programming language designed for complicated problems. In addition, the above drawbacks and the observation of bugs related to the internal structure of XLOOPS-Maple led to the conclusion that XLOOPS-Maple had reached a state in which further development was almost impossible [18]. Thus the decision was made to look for an alternative symbolic programming language to rewrite XLOOPS, putting it

on more solid grounds. The alternative symbolic programming environment was realized in GiNaC, a C++ library developed and written in Mainz [19].

Inheriting the progress of the method for Feynman loop integration from the XLOOPS project, the aims of this thesis are

- to rewrite XLOOPS in a solid and homogeneous programming environment, the C++ programming language [20] and the GiNaC library [19]. The new version of XLOOPS based on the GiNaC symbolic library written in the C++ programming language is called XLOOPS-GiNaC.
- to re-implement and improve (fix bugs) the procedures for the analytic evaluation of one-loop one-, two- and three-point tensor integrals.
- to implement a new procedure for evaluating two-loop two-point tensor integrals. Based on the one-loop procedures, the divergent part of two-loop two-point integrals is calculated analytically while their finite part is integrated analytically up to a remaining two-fold numerical integration.
- to apply the advantages of the XLOOPS-GiNaC package to solve phenomenological problems of particle physics.

In the second chapter of this thesis, after a short introduction to the standard t'Hooft-Veltman-Passarino method, we present the method for calculating Feynman loop integrals in parallel and orthogonal spaces [11, 21].

In chapter three, we present a tensor reduction algorithm for one-loop one-, two- and three-point integrals of arbitrary tensor degree and arbitrary masses. It is shown that all the tensor one-loop two- and three-point integrals can be reduced to a set of scalar (zero tensor degree) one-loop integrals A_0, B_0 and C_0.

In chapter four, we present a complete calculation of two-loop two-point tensor integrals for arbitrary masses. After doing tensor reduction in parallel and orthogonal spaces

we introduce a suitable set of subtraction terms to split the integral into ultra-violet (UV) divergent and UV-finite parts. It turns out that the UV-finite part of all the non-trivial two-loop two-point tensor integrals can be reduced to a subset of twofold numerical integrals while its UV-divergent part can be expressed in terms of one-loop functions [12].

Although the source code of the XLOOPS-Maple package is available, reading the source code without adequate documentation is a mission impossible. Furthermore, the XLOOPS-Maple package contains some known bugs in both one-loop and two-loop procedures. Because of these difficulties, besides rewriting XLOOPS from scratch, the algorithm for one-loop and two-loop tensor integral reduction presented here is a completely new implementation. In particular the overlapping divergence of the sunrise integral is now calculated correctly in XLOOPS-GiNaC, which was not the case in XLOOPS-Maple. The sunrise integral is a basic integral to which other (non-trivial) two-loop two-point tensor integrals are reduced.

In chapter five, we present a short introduction to the XLOOPS-GiNaC library. We also provide some examples of using the XLOOPS-GiNaC one- and two-loop libraries to calculate some real physical processes. We present the corresponding source code and give a detailed line-by-line description. This may help beginners to get a quick start with their own applications.

In chapter six, we calculate the anomalous magnetic moment g_2 of the nucleon in the framework of the so-called "spontaneous color symmetry breaking" model in QCD [22]. This calculation involves the calculation of one-loop vertex corrections with massive quarks and massive gluons.

The use of XLOOPS in these calculations is important, first because XLOOPS speeds up the calculations and on the other hand, because the applications are important as a test of XLOOPS. Such a real test has only been done by [23] so far.

In the appendix, we provide some mathematical material and explicit expressions for scalar one-loop functions.

Chapter 2

Feynman Loop-Integrals in Orthogonal and Parallel Spaces

In this chapter, we present the mathematical framework for the calculations in the next sections. We will use the dimensional regularization (DR) technique to regularize the UV and infrared (IR) divergences which appear when performing the Feynman loop integration.[1]

The necessary steps in the calculation of cross-sections or decay widths of an interaction process in high energy physics can be summarized as follows:

- For each theory of interaction which may be defined by an associated set of Feynman rules, one constructs all possible Feynman diagrams for a given physical process at a given order of perturbation theory.
- Next, one constructs the amplitude $\mathcal{M}$ of each diagram based on the Feynman rules. The amplitude, in general, is a function of external momenta q_i^μ of the particles in initial and final states and of the masses of particles involved.

[1] In this thesis, only IR-divergences which appear in one-loop integrals are regularized by using DR. For two-loop integrals, we regularize IR-divergences by introducing small masses.

- If the amplitude involves loops, one has to calculate Feynman loop-integrals.
- The cross-sections or decay widths can be obtained by summing the amplitudes of all diagrams, squaring the sum and then integrating it over the phase space of the external momenta where needed.

Calculating the amplitude of a given set of Feynman diagrams at one- and two-loop orders is the main purpose of the XLOOPS project. The next two chapters are dedicated for the step of calculating Feynman loop integrals.

The general form of a Feynman tensor loop integral at the one-loop level can be written as

$$T^{(1)\,\mu_1\ldots\mu_p}_{t_1\ldots t_n} = \int d^D l \, \frac{l^{\mu_1}\ldots l^{\mu_p}}{\prod\limits_{i=1}^{n}\left[(l+q_i)^2 - m_i^2 + i\rho\right]^{t_i}}. \tag{2.1}$$

The integration is performed over the $D = 4 - 2\epsilon$ dimensional space of the internal loop momentum l. The index p is the tensor degree of the integral and can take any non-negative integer value. The index n is the number of propagators in the loop and q_i are external momenta. The masses m_i, in general, can take any real value. We also keep the causality prescription $i\rho$ explicitly. In general, each inverse propagator in the denominator of Eq. (2.1) can take any integer power $t_i \geq 1$.

At the two-loop level, the most general integral structure can be constructed by combinations of internal momenta l and k, the external momenta q_s and masses m_s of particles involved

$$\begin{aligned} T^{(2)\,\mu_1\ldots\mu_p,\nu_1\ldots\nu_r}_{t_1\ldots t_{n_l};\ldots t_{n_l+n_m};\ldots t_{n_l+n_m+n_k}} = \int d^D l \int d^D k \, & \frac{l^{\mu_1}\ldots l^{\mu_p}}{\prod\limits_{s=1}^{n_l}\left[(l+q_s)^2 - m_s^2 + i\rho\right]^{t_s}} \\ & \times \frac{1}{\prod\limits_{s=n_l+1}^{n_l+n_m}\left[(l+k+q_s)^2 - m_s^2 + i\rho\right]^{t_s}} \\ & \times \frac{k^{\nu_1}\ldots k^{\nu_r}}{\prod\limits_{s=n_n+n_m+1}^{n_l+n_m+n_k}\left[(k+q_s)^2 - m_s^2 + i\rho\right]^{t_s}} \end{aligned} \tag{2.2}$$

where n_l, n_k, n_m are the numbers of propagators which carry momentum l, k or $l+k$ respectively.

Evaluating the one-loop tensor integrals, in principle, was completely solved long ago by t'Hooft, Veltman and Passarino [24, 25]. Even though we are not using this conventional technique of calculation in this thesis, we briefly summarize the main steps of this method:

- perform Feynman-Schwinger parametrisation
  (Feynman parametrisation)

$$\frac{1}{P_1^{\alpha_1} P_2^{\alpha_2} \dots P_n^{\alpha_n}} = \frac{\Gamma(\alpha_1+\alpha_2+\dots\alpha_n)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\dots\Gamma(\alpha_n)} \int_0^1 dx_1\, dx_2 \dots dx_n \delta(1-x_1-x_2-\dots x_n) \times \frac{x_1^{\alpha_1-1} x_2^{\alpha_2-1} \dots x_n^{\alpha_n-1}}{[P_1 x_1 + P_2 x_2 + \dots P_n x_n]^{\alpha_1+\alpha_2\dots+\alpha_n}}, \tag{2.3}$$

  (Schwinger parametrisation)

$$\frac{1}{P^\alpha} = \frac{1}{\Gamma(\alpha)} \int_0^\infty dx\, x^{\alpha-1}\, \exp[P] \tag{2.4}$$

  where $P_1, \dots, P_n$ are inverse propagators. The original integrand is transformed into the integration over Feynman-Schwinger parameters.

- the tensor structure of the loop integral after the parametrisation and a shift on the loop momenta is reduced to the form of a tadpole integral

$$A_t^{\mu_1\dots\mu_p}(C) := \int \frac{d^D l}{(2\pi)^D} \frac{l^{\mu_1}\dots l^{\mu_p}}{[l^2+C]^t} \tag{2.5}$$

$$= \frac{(g^{\mu_1\mu_2}\dots g^{\mu_{p-1}\mu_p} + \dots \text{all permutation of } \{\mu_1\dots\mu_p\})}{\prod_{n=0}^{(D-2)/2} (D+2n)} A_t^{p/2}(C)$$

  where C is a function of squares and products of external momenta, masses m_i, and Feynman-Schwinger parameters. The coefficient $A_t^{p/2}(C)$ can be reduced by Eq. (2.5) to the form

$$A_t^{p/2}(C) = \int \frac{d^D l}{(2\pi)^D} \frac{(l^2)^{p/2}}{[l^2+C]^t}. \tag{2.6}$$

  The above tadpole integral is symmetric in l so it vanishes for odd p.

- perform a Wick rotation by the transformation $l^0 \to i\omega$. Then

$$l^2 \;=\; -\omega^2 - \vec{l}^{\,2} \;=\; -l_E^2, \quad l_E = (\,\omega, \vec{l}\,),$$
$$\int_{-\infty}^{\infty} d\,l^0 \to i \int_{-\infty}^{\infty} d\,\omega.$$

- integrate over the loop momenta using DR and obtain

$$A_t^n(C) = i\frac{(-1)^{n-t}}{(4\pi)^2}\left(\frac{4\pi}{C}\right)^{\epsilon} C^{2+n-t}\frac{\Gamma(2+n-\epsilon)}{\Gamma(2-\epsilon)}\frac{\Gamma(t-n-2+\epsilon)}{\Gamma(t)}. \qquad (2.7)$$

- in the last step, one has to integrate over Feynman parameters.

At the two-loop level, one can apply the above procedure one after the other for the internal l- and k-integrals. The method still works well with two-point integrals [26, 27] or, in some cases, for three-point integrals [28]. Nevertheless, this method will run into difficulties when being applied to three- or four-point integrals since the first integration results in a complicated function of the second internal loop momentum. Further, from the point of view of computation, the procedure is difficult to implement.

There are other alternative methods which are often used in the two-loop problem such as differential equations in external Mandelstam variables [28], asymptotic expansions in the momentum space [29] or in the configuration space [30].

In this thesis, the calculation is based on an alternative method which was introduced by Collins [21] and further developed by Kreimer [11, 12]. A similar approach was used in a subsequent paper by Ghinculov and Yao [31]. The advantages of this approach, compared to the conventional and other methods are that it opens an easier way to implement a generic algorithm for evaluating arbitrary tensor loop integrals. A detailed description of the method and its application to one- and two-loop problems will be discussed below and in the next sections of the thesis.

The main idea of the method is to split the space of integration into parallel and its orthogonal complement subspaces spanned by external momenta. Working in this special

representation of space-time, one does not need to perform the Feynman parametrisation but instead performs all integrations in a given reference frame. To regularize UV-divergences, one works in $D = 4-2\epsilon$ dimensional space-time. We define the parallel space as the linear span of the n external momenta q_i^μ $(i = 1, ..., n)$ involved in the integrand. This parallel space has a finite dimension $J \leq D$. The remaining $D - J$ dimensions span an orthogonal complement of the parallel space called the orthogonal space [32].

Once an explicit configuration of external momenta is chosen, the dimension of the parallel space J is known, and the scalar products are written explicitly in terms of the components of the external momenta

$$\begin{aligned} l^2 &= l_0^2 - l_1^2 - \cdots - l_{J-1}^2 - |\vec{l}_\perp|^2, \\ l \cdot q_i &= l_0 q_{i,0} - l_1 q_{i,1} - \cdots - l_{J-1} q_{i,J-1}, \\ k \cdot q_i &= k_0 q_{i,0} - k_1 q_{i,1} - \cdots - k_{J-1} q_{i,J-1}, \\ l \cdot k &= l_0 k_0 - \cdots - l_{J-1} k_{J-1} - |\vec{l}_\perp||\vec{k}_\perp| z \end{aligned} \tag{2.8}$$

where $z = \cos\theta$, and θ is the angle between $\vec{l}_\perp$ and $\vec{k}_\perp$.

In terms of the parallel and orthogonal space components, the general one-loop tensor integral in Eq. (2.1) can be reduced to

$$T^{(1)\,\mu_1\ldots\mu_p}_{t_1\ldots t_n} = \frac{2\pi^{\frac{D-J}{2}}}{\Gamma(\frac{D-J}{2})} \int_{-\infty}^{\infty} dl_0 \cdots \int_{-\infty}^{\infty} dl_{J-1} \int_0^{\infty} dl_\perp \, l_\perp^{D-J-1} \, \mathcal{F}^{\mu_1\ldots\mu_p}(l_\perp, l_0, \ldots, l_{J-1}) \tag{2.9}$$

where $l_\perp = |\vec{l_\perp}|$. The coefficient in front of the integral is the area of the surface of a hyper-sphere in $D - J$ dimensions given by

$$\int d^{D-J}\vec{l}_\perp = \int l_\perp^{D-J-1} dl_\perp \int d\Omega_{D-J-1} \tag{2.10}$$

and

$$\int d\Omega_{D-J-1} = \frac{2\pi^{\frac{D-J}{2}}}{\Gamma(\frac{D-J}{2})}. \tag{2.11}$$

Similarly, the general two-loop tensor integral in Eq. (2.2) can be reduced to

$$
\begin{aligned}
T^{(2)\,\mu_1\ldots\mu_p,\nu_1\ldots\nu_r}_{t_1\ldots t_{n_l};\ldots t_{n_l+n_m};\ldots t_{n_l+n_m+n_k}} &= \frac{2\pi^{\frac{D-J}{2}}}{\Gamma(\frac{D-J}{2})}\frac{2\pi^{\frac{D-J-1}{2}}}{\Gamma(\frac{D-J-1}{2})}\int\limits_{-\infty}^{\infty} dl_0\cdots\int\limits_{-\infty}^{\infty} dl_{J-1}\int\limits_{0}^{\infty} dl_\perp l_\perp^{D-J-1} \\
&\times \int\limits_{-\infty}^{\infty} dk_0\cdots\int\limits_{-\infty}^{\infty} dk_{J-1}\int\limits_{0}^{\infty} dk_\perp k_\perp^{D-J-1}\int\limits_{0}^{\pi} d\theta(\sin\theta)^{D-J-1} \qquad (2.12)\\
&\times \mathcal{G}^{\mu_1\ldots\mu_p,\nu_1\ldots\nu_r}(l_\perp,l_0,\ldots,l_{J-1};k_\perp,k_0,\ldots,k_{J-1};\cos\theta)\,.
\end{aligned}
$$

As a result, the tensor integrals on the left-hand side of Eqs. (2.9, 2.12) are reduced to a set of integrands $\mathcal{F}$ and $\mathcal{G}$ over the $J+1$ and $2(J+1)+1$ real variables $l_0,\ldots l_{J-1}$, $k_0,\ldots k_{J-1}, l_\perp, k_\perp, \theta$ which will be specified later on. The advantages of using the new representation for integrals are that the integrals of the new type can be evaluated without using a Wick rotation and Feynman parametrisation. The method of working in parallel and orthogonal spaces opens a new way for developing a generic and efficient algorithm to do tensor reduction and to evaluate integrals with an arbitrary tensor order.

To have a closer look at the tensor and integration structure of $\mathcal{F}$ and $\mathcal{G}$, we consider the two-point ($J=1$) and three-point functions ($J=2$) in the parallel and orthogonal spaces. The simplest configuration of the external momenta for those two cases is the one in which the incoming particle is at rest. For the two-point diagrams, assuming $q_1^2>0$, there is only one independent momentum and for the three-point diagrams there are two independent momenta. For both two- and three-point integrals, we call q_1 the momentum of the incoming particle. For the three-point integral, the second independent momentum q_2 is one of the outgoing particles. In the rest frame of the incoming particle, one can write

$$
q_1^\mu = (q_{10},0,\vec{0}); \quad q_2^\mu = (q_{20},q_{21},\vec{0}), \qquad (2.13)
$$

where $\vec{0}$ is the $D-2$ dimensional orthogonal zero-vector.

By projecting onto the parallel and orthogonal spaces spanned by q_1 and q_2, the parallel

components of the loop momentum l (and similarly for k) can be constructed as

$$\begin{aligned} l_0 &= \frac{l\cdot q_1}{\sqrt{q_1^2}} && \text{for } J=1 \\ \text{or} & && \\ l_0 &= \frac{l\cdot q_1}{\sqrt{q_1^2}}; \quad l_1 = \frac{l\cdot \hat{q}_2}{\sqrt{\hat{q}_2^2}} && \text{for } J=2 \end{aligned} \tag{2.14}$$

where

$$\hat{q}_2^\mu = q_2^\mu - \frac{q_1\cdot q_2}{q_1^2} q_1^\mu = (0; q_{21}, \vec{0}). \tag{2.15}$$

The orthogonal component is constructed by

$$l_\perp^\mu = \begin{cases} l^\mu - \dfrac{l\cdot q_1}{q_1^2}\, q_1^\mu & \text{for } J=1 \\ l^\mu - \dfrac{l\cdot q_1}{q_1^2}\, q_1^\mu - \dfrac{l\cdot \hat{q}_2}{\hat{q}_2^2}\, \hat{q}_2^\mu & \text{for } J=2. \end{cases} \tag{2.16}$$

The tensor metric of the orthogonal subspace is constructed by

$$g_\perp^{\mu\nu} = \begin{cases} g^{\mu\nu} - \dfrac{q_1^\mu\, q_1^\nu}{q_1^2} & \text{for } J=1 \\ g^{\mu\nu} - \dfrac{q_1^\mu\, q_1^\nu}{q_1^2} - \dfrac{\hat{q}_2^\mu\, \hat{q}_2^\nu}{\hat{q}^2} & \text{for } J=2 \end{cases} \tag{2.17}$$

where

$$g_\perp^{\mu\nu} g_{\perp\,\mu\nu} = D_\perp := D - J. \tag{2.18}$$

Inserting l^μ from Eq. (2.16) into Eq. (2.1), the one-loop integral can be written as a sum of tensor integrals

$$\mathbb{T}^{(1)\,\mu_1\ldots\mu_{p'}}_{t_1\ldots t_n} = \int d^D l\, \frac{l_\perp^{\mu_1}\ldots l_\perp^{\mu_{p'}} \mathcal{P}(l_0,\ldots,l_{J-1})}{\prod_{i=1}^{n} \left[(l+q_i)^2 - m_i^2 + i\rho\right]^{t_i}} \tag{2.19}$$

with a tensor degree p' smaller or equal to the degree p of the original integral. $\mathcal{P}$ is a polynomial of the parallel components $l_0,\ldots,l_{J-1}$. For $p=p'$ one has $\mathcal{P}=1$, and integrals with $p'>p$ originate from terms containing $q_1, \hat{q}_2$. To avoid using different

indices as $p, p', p'', ...$ in intermediate steps of the calculation, the primes will not be written explicitly in the following.

Similarly, the two-loop integrals are reduced to the form

$$\begin{aligned}\mathbb{T}^{(2)\,\mu_1\ldots\mu_p,\nu_1\ldots\nu_r}_{t_1\ldots t_{n_l};\ldots t_{n_l+n_m};\ldots t_{n_l+n_m+n_k}} &= \int d^Dl \int d^Dk \,\frac{1}{\prod\limits_{s=1}^{n_l}\left[(l+q_s)^2-m_s^2+i\rho\right]^{t_s}} \\ &\times \frac{l_\perp^{\mu_1}\ldots l_\perp^{\mu_p} k_\perp^{\nu_1}\ldots k_\perp^{\nu_r}\,\mathcal{P}(l_0,\ldots,l_{J-1},k_0,\ldots,k_{J-1})}{\prod\limits_{s=n_l+1}^{n_l+n_m}\left[(l+k+q_s)^2-m_s^2+i\rho\right]^{t_s}} \\ &\times \frac{1}{\prod\limits_{s=n_l+n_m+1}^{n_l+n_m+n_k}\left[(k+q_s)^2-m_s^2+i\rho\right]^{t_s}}\end{aligned} \tag{2.20}$$

where $\mathcal{P}$ is a polynomial of the parallel components of $l_0,\ldots,l_{J-1},\ k_0,\ldots,k_{J-1}$. The indices p and r are equal to or smaller than their original values in Eq. (2.2).

As in the standard procedure, one makes an ansatz by constructing all possible tensor structures from combinations of q_i^μ and $g_\perp^{\mu\nu}$ with scalar coefficients. Because the tensor structure of the integrands in Eqs. (2.19, 2.20) is transverse [2], a contraction with any combination of q_i^μ vanishes, so the possible ansatz for the $\mathbb{T}^{(1)\,\mu_1\ldots\mu_p}_{t_1\ldots t_n}$ can be constructed by combinations of $g_\perp^{\mu\nu}$ only:

$$\mathbb{T}^{(1)\,\mu_1\ldots\mu_p}_{t_1\ldots t_n} = \frac{(g_\perp^{\mu_1\mu_2}\ldots g_\perp^{\mu_{p-1}\mu_p}+\ldots\text{all permutation of }\{\mu_1\ldots\mu_p\})}{\prod\limits_{j=0}^{(D-2)/2}(D-J+2j)}\, T^{(1)\,(p_0\ldots p_{J-1}\,p_\perp)}_{t_1\ldots t_n} \tag{2.21}$$

where

$$T^{(1)\,(p_0\ldots p_{J-1}\,p_\perp)}_{t_1\ldots t_n} = \int d^Dl\,\frac{l_0^{p_0}\ldots l_{J-1}^{p_{J-1}}\,|l_\perp|^{p_\perp}}{\prod\limits_{i=1}^{n}\left[(l+q_i)^2-m_i^2+i\rho\right]^{t_i}} \tag{2.22}$$

with $p_\perp = p$. When p is odd, the integral vanishes because the integrand is symmetric in $l_\perp$. The exponents $p_0,\ldots p_\perp$ pertaining to the real variables $l_0,\ldots l_\perp$ are non-negative

[2] The terminology *transverse* means that the tensor structure of the integrals contains only orthogonal components of the internal momenta.

integers. They are determined from the polynomials $\mathcal{P}$ from Eq. (2.20). They should not be confused with the Lorentz indices in Eq. (2.1)
Similarly, for the case where p and r are both even, the possible ansatz for the two-loop integrals is

$$\begin{aligned}
\mathbb{T}^{(2)\,\mu_1\ldots\mu_p,\nu_1\ldots\nu_r}_{t_1\ldots t_{n_l};\ldots t_{n_l+n_m+n_k}} = \;& C_0\left(g_\perp^{\mu_1\mu_2}\ldots g_\perp^{\mu_{p-1}\mu_p} g_\perp^{\nu_1\nu_2}\ldots g_\perp^{\nu_{r-1}\nu_r} + \mathrm{sym}(\mu_1\ldots\mu_p;\nu_1\ldots\nu_r)\right)\\
&+ C_2\left(g_\perp^{\mu_1\nu_1} g_\perp^{\mu_2\nu_2} g_\perp^{\mu_3\mu_4}\ldots g_\perp^{\mu_{p-1}\mu_p} g_\perp^{\nu_3\nu_4}\ldots g_\perp^{\nu_{r-1}\nu_r} + \mathrm{sym}(\mu_1\ldots\mu_p;\nu_1\ldots\nu_r)\right)\\
&+ \ldots\ldots\\
&+ C_p\left(g_\perp^{\mu_1\nu_1}\ldots g_\perp^{\mu_p\nu_p}\ldots g_\perp^{\nu_{p+1}\nu_{p+2}}\ldots g_\perp^{\nu_{r-1}\nu_r} + \mathrm{sym}(\mu_1\ldots\mu_p;\nu_1\ldots\nu_r)\right).
\end{aligned} \tag{2.23}$$

Here we suppose that $p \le r$, otherwise we have to exchange $\mu \leftrightarrow \nu$. The ansatz for the case where both p and r are odd and $p \le r$ is

$$\begin{aligned}
\mathbb{T}^{(2)\,\mu_1\ldots\mu_p,\nu_1\ldots\nu_r}_{t_1\ldots t_{n_l};\ldots t_{n_l+n_m+n_k}} = \;& \\
& C_1\left(g_\perp^{\mu_1\nu_1} g_\perp^{\mu_2\mu_3}\ldots g_\perp^{\mu_{p-1}\mu_p} g_\perp^{\nu_2\nu_3}\ldots g_\perp^{\nu_{r-1}\nu_r} + \mathrm{sym}(\mu_1\ldots\mu_p;\nu_1\ldots\nu_r)\right)\\
&+ C_3\left(g_\perp^{\mu_1\nu_1} g_\perp^{\mu_2\nu_2} g_\perp^{\mu_3\nu_3}\ldots g_\perp^{\mu_{p-1}\mu_p} g_\perp^{\nu_4\nu_5}\ldots g_\perp^{\nu_{r-1}\nu_r} + \mathrm{sym}(\mu_1\ldots\mu_p;\nu_1\ldots\nu_r)\right)\\
&+ \ldots\ldots\\
&+ C_p\left(g_\perp^{\mu_1\nu_1}\ldots g_\perp^{\mu_p\nu_p}\ldots g_\perp^{\nu_{p+1}\nu_{p+2}}\ldots g_\perp^{\nu_{r-1}\nu_r} + \mathrm{sym}(\mu_1\ldots\mu_p;\nu_1\ldots\nu_r)\right)
\end{aligned} \tag{2.24}$$

where the abbreviation $\mathrm{sym}(\mu_1\ldots\mu_p;\nu_1\ldots\nu_r)$ stands for all possible symmetric combinations of the indices μ_i and ν_i separately. By contracting the two sides of Eqs. (2.23, 2.24) with appropiate symmetric combinations of $g_\perp^{\mu\nu}$ one obtains a system of equations for the coefficients C_i. The coefficients C_i, in general, are functions of $D_\perp$ and integrals

of the form

$$T^{(2)\ p_0\ldots p_{J-1},p_\perp;\ r_0\ldots r_{J-1},r_\perp;\ p_z}_{t_1\ldots t_{n_l};\ldots t_{n_l+n_m};\ldots t_{n_l+n_m+n_k}} = \int d^Dl \int d^Dk \frac{1}{\prod\limits_{s=1}^{n_l}\left[(l+q_s)^2-m_s^2+i\rho\right]^{t_s}} \times \frac{l_0^{p_0}\ldots l_{J-1}^{p_{J-1}}\ l_\perp^{p_\perp}\ k_0^{r_0}\ldots k_{J-1}^{r_{J-1}}\ k_\perp^{r_\perp}\ z^{p_z}}{\prod\limits_{s=n_l+1}^{n_l+n_m}\left[(l+k+q_s)^2-m_s^2+i\rho\right]^{t_s}} \times \frac{1}{\prod\limits_{s=n_l+n_m+1}^{n_l+n_m+n_k}\left[(k+q_s)^2-m_s^2+i\rho\right]^{t_s}} \tag{2.25}$$

where $z=\cos\theta$ and θ is the angle between $\vec{l}_\perp$ and $\vec{k}_\perp$. The exponents $p_0,\ldots p_{J-1}$, $p_\perp$, $r_0,\ldots r_{J-1}$, $r_\perp$, p_z are non-negative integer numbers (and should not be confused with Lorentz indices).

In the case that $p+r$ is odd, the integral vanishes. To prove that, we contract $p+r-1$ indices with symmetric combination of $g_\perp^{\mu\nu}$ and end up with the following structure

$$\int d^Dl \int d^Dk \frac{l_\perp^\mu\,\mathcal{P}(l_0\ldots l_{J-1},k_0\ldots k_{J-1},l_\perp,k_\perp,\vec{l}_\perp\cdot\vec{k}_\perp)}{\ldots\ldots\ldots} = q_i^\mu\mathcal{C} \tag{2.26}$$

where the right hand side of Eq. (2.26), a vector in the parallel space with a scalar coefficient $\mathcal{C}$ is the only possible ansatz. On the other hand, the left hand side of Eq. (2.26) is a vector in the orthogonal space and therefore the equality is valid only for $\mathcal{C}=0$, thus the integrals with an odd sum of p and r vanish.

There are two important remarks here:

- in Eq. (2.22), the integrals with odd $p_\perp$ vanish because the denominator of the integrand is symmetric in $l_\perp$.
- in Eq. (2.25), the denominator of the integrand is not symmetric either in $l_\perp$ or in $k_\perp$. In general, p and r can be odd or even. However, as a consequence of the transverse property of $\mathbb{T}^{(2)\ \mu_1\ldots\mu_p,\nu_1\ldots\nu_r}_{t_1\ldots t_{n_l};\ldots t_{n_l+n_m};\ldots t_{n_l+n_m+n_k}}$, one has other constraints:
  - The integrals with odd $p_\perp+r_\perp$ vanish.

- p_z must be smaller or equal to $\min\{p_\perp, r_\perp\}$.
- Both $p_\perp - p_z$ and $r_\perp - p_z$ are even.

In this section, we have presented the mathematical framework to performing the tensor reduction in parallel and orthogonal spaces.

From now on, we call $T^{(1)\ (p_0 \ldots p_{J-1} p_\perp)}_{t_1 \ldots t_n}$ and $T^{(2)\ p_0,\ldots p_{J-1},p_\perp;\ r_0,\ldots r_{J-1},r_\perp;\ p_z}_{t_1 \ldots t_{n_l};\ldots t_{n_l+n_m};\ldots t_{n_l+n_m+n_k}}$ **tensor integrals**.

In the next chapter, we present an algorithm for the automatic calculation of one-loop one-, two- and three-point tensor integrals. An algorithm for two-loop two-point tensor integrals will be discussed in chapter 4.

Chapter 3

One-Loop One-, Two- and Three-Point Integrals

3.1 One-loop two-point tensor integrals

In this section, the algorithm for an automatic calculation of one-loop two-point tensor integrals is presented.

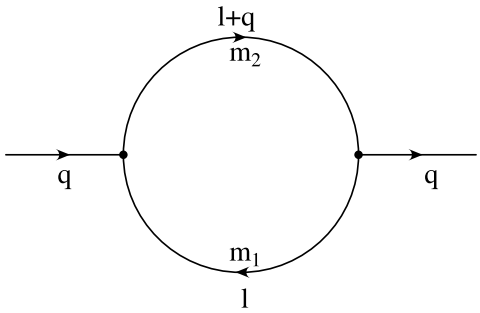


Figure 3.1: Notation for two-point functions

We first consider the case where q^2 is timelike, $q^2 > 0$. Then one can choose a reference frame where $q^\mu = (q_0, 0, 0, 0)$. The general tensor integral for a Feynman diagram shown

in Fig. 3.1 has the form of

$$B^{ij}_{t_1t_2}(q^2;m_1,m_2) := \int d^Dl\,\frac{l_0^i\,l_\perp^j}{P_1^{t_1}P_2^{t_2}} = T^{(1)\,(ij)}_{t_1t_2} \tag{3.1}$$

where $T^{(1)\,(ij)}_{t_1t_2}$ is the notation for tensor integrals introduced in the previous chapter, and

$$\begin{aligned} P_1 &= (l_0+q_0)^2 - l_\perp^2 - m_1^2 + i\rho, \\ P_2 &= l_0^2 - l_\perp^2 - m_2^2 + i\rho. \end{aligned} \tag{3.2}$$

The components of the internal momentum, l_0 and $l_\perp$, span the parallel and orthogonal subspaces, respectively. For brevity, the arguments of $B^{ij}_{t_1t_2}(q^2;m_1,m_2)$ will not be written explicitly if there is no confusion. The integral vanishes unless j is even. In the spacelike and lightlike cases where $q^2<0$ or $q^2=0$ one can choose a reference frame where $q^\mu=(0,q_1,0,0)$ or $q^\mu=(q_0,q_0,0,0)$, respectively, and the integral space can be split into a two-dimensional parallel and a $(D-2)$-dimensional orthogonal subspaces. We will consider these cases later in section 3.2.3. A genuine one-loop integral has $t_1=t_2=1$, but the more general case is needed in the case of the reduction of integrals with more than one loop.

The strategy now is to expand $B^{ij}_{t_1t_2}$ to a sum of simpler integrals. It turns out that the usual scalar one- and two-point integrals

$$A_0 = A_1^0 \qquad \text{(see Eq. (2.7))}$$

and

$$B_0 = B_{11}^{00} \tag{3.3}$$

are sufficient for the expansion. This expansion is possible except for some special cases that we will consider later.

Firstly, consider the general case where $q_0\neq 0$. We then express the numerator of the integral in Eq. (3.1) as a function of P_1 and P_2. From Eq. (3.2) we get:

$$\begin{aligned} l_0 &\to l_0(P_1,P_2,q_0,m_1,m_2) = \frac{1}{2q_0}(P_1-P_2-b_1), \\ l_\perp^2 &\to l_\perp^2(P_1,P_2,q_0,m_1,m_2) = l_0^2 - P_2 + b_2 \end{aligned} \tag{3.4}$$

where

$$\begin{aligned} b_1 &= q_0^2 - m_1^2 + m_2^2, \\ b_2 &= -m_2^2 + i\rho. \end{aligned} \tag{3.5}$$

Inserting Eq. (3.4) into Eq. (3.1) and expanding the numerator of the integrand, one obtains

$$B^{ij}_{t_1 t_2} = \sum_{n,m=0}^{i+j} \mathcal{K}_{nm} \int d^D l \, P_1^{n-t_1} \, P_2^{m-t_2} \tag{3.6}$$

with $\mathcal{K}_{nm}$ being simple functions of q_0, m_1, m_2, and explicitly

$$\int d^D l \, P_1^{n-t_1} \, P_2^{m-t_2} = \begin{cases} 0 & \text{if } n - t_1 \geq 0 \text{ and } m - t_2 \geq 0, \\ \displaystyle\int d^D l \, \frac{1}{P_1^{t_1-n} \, P_2^{t_2-m}} & \text{if } n - t_1 < 0 \text{ and } m - t_2 < 0, \\ \displaystyle\int d^D l \, \frac{P_1^{n-t_1}}{P_2^{t_2-m}} & \text{if } n - t_1 \geq 0 \text{ and } m - t_2 < 0, \\ \displaystyle\int d^D l \, \frac{P_2^{m-t_2}}{P_1^{t_1-n}} & \text{if } n - t_1 < 0 \text{ and } m - t_2 \geq 0. \end{cases} \tag{3.7}$$

The second case actually is a derivative with respect to the square masses of the scalar two-point function B_0 by taking into account that

$$\int d^D l \, \frac{1}{P_1^{t_1} \, P_2^{t_2}} = \frac{1}{(t_1-1)!(t_2-1)!} \left(\frac{\partial}{\partial m_1^2} \right)^{t_1-1} \left(\frac{\partial}{\partial m_2^2} \right)^{t_2-1} B_0(q^2; m_1, m_2) \tag{3.8}$$

where $B_0(q^2; m_1, m_2)$ is defined in Eq. (3.3).

For the last two cases, from Eq. (3.2) one can insert

$$P_1 = P_1(P_2, l_0, q_0) = 2 l_0 q_0 + P_2 + b_1 \tag{3.9}$$

or

$$P_2 = P_2(P_1, l_0, q_0) = P_1 - 2 l_0 q_0 - b_1 \tag{3.10}$$

and expand the numerator of the integrands in Eq. (3.7). The result is a sum of tadpole functions A_t^n, which is again reduced to A_0. The explicit analytical results for the integrals A_0 and B_0 are well-known and will be given in the appendix A.

3.1.1 The case $q_0 = 0$

If $q_0 = 0$, Eq. (3.7) must be rewritten using

$$\begin{aligned} P_1 &= l_0^2 - l_\perp^2 - m_1^2 + i\rho, \\ P_2 &= l_0^2 - l_\perp^2 - m_2^2 + i\rho. \end{aligned} \tag{3.11}$$

If $m_1 = m_2$ then $P_1 = P_2$ and $B^{ij}_{t_1 t_2}$ has a simple form

$$B^{ij}_{t_1 t_2}(q^2, m_1, m_2)\big|_{q_0=0,\, m_1=m_2} = \int d^D l \, \frac{l_0^i \, l_\perp^j}{P_1^{t_1+t_2}(m_1^2)} \tag{3.12}$$

which is actually again the one-loop one-point function A^n_t with $n = i + j$ and $t = t_1 + t_2$.

If $m_1 \neq m_2$, one performs a partial fraction decomposition and obtains

$$\begin{aligned} B^{ij}_{t_1 t_2}(q^2, m_1, m_2)\big|_{q_0=0} &= \int d^D l \, \frac{l_0^i \, l_\perp^j}{P_1 (m_1^2)^{t_1} \, P_2 (m_2^2)^{t_2}} \\ &= \frac{1}{(t_1-1)!} \frac{\partial^{t_1-1}}{\partial (m_1^2)^{t_1-1}} \left(\frac{1}{(m_1^2 - m_2^2)^{t_2}} \int d^D l \, \frac{l_0^i \, l_\perp^j}{P_1(m_1^2)} \right) + \\ &\quad \frac{1}{(t_2-1)!} \frac{\partial^{t_2-1}}{\partial (m_2^2)^{t_2-1}} \left(\frac{1}{(m_2^2 - m_1^2)^{t_1}} \int d^D l \, \frac{l_0^i \, l_\perp^j}{P_2(m_2^2)} \right), \end{aligned} \tag{3.13}$$

which is also a combination of one-loop one-point functions.

3.2 One-loop three-point tensor functions

The notation we use for one-loop three-point functions is shown in Fig. 3.2. We are working in the frame of reference where the external momentum configuration is $q_1^\mu = (q_{10}, 0, 0, 0)$, $q_2^\mu = (q_{20}, q_{21}, 0, 0)$. The parallel space is now two-dimensional and the general form of the one-loop three-point tensor function is

$$\begin{aligned} C^{ijk}_{t_1 t_2 t_3}(q_1, q_2; m_1, m_2, m_3) &:= \int d^D l \, \frac{l_0^i \, l_1^j \, l_\perp^k}{P_1^{t_1} \, P_2^{t_2} \, P_3^{t_3}} \\ &= T^{(1)\,(ijk)}_{t_1 t_2 t_3} \qquad \text{(c.f. Eq. (2.22))} \end{aligned} \tag{3.14}$$

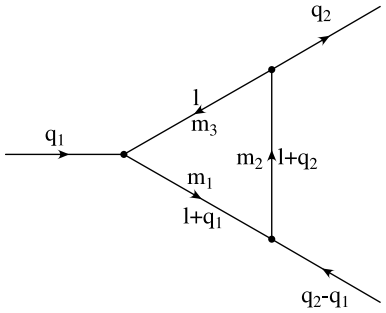


Figure 3.2: Notation for three-point functions

with

$$\begin{aligned} P_1 &= l_0^2 - l_1^2 - l_\perp^2 + 2\, l_0\, q_{10} + q_{10}^2 - m_1^2 + i\rho, \\ P_2 &= l_0^2 - l_1^2 - l_\perp^2 + 2\, l_0\, q_{20} - 2\, l_1\, q_{21} + q_{20}^2 - q_{21}^2 - m_2^2 + i\rho, \\ P_3 &= l_0^2 - l_1^2 - l_\perp^2 - m_3^2 + i\rho \end{aligned} \tag{3.15}$$

where $\{l_0, l_1\}$ and $l_\perp$ span the parallel and orthogonal subspaces, respectively. For brevity, the arguments of the function $C^{ijk}_{t_1 t_2 t_3}(q_1, q_2; m_1, m_2, m_3)$ will not be written explicitly if there is no confusion.

3.2.1 The general case

First, consider the case where $q_{10} \neq 0$ and $q_{21} \neq 0$. We can always express l_0, l_1, $l_\perp$ in terms of P_1, P_2, P_3:

$$\begin{aligned} l_0 &= \frac{1}{2\, q_{10}} (P_1 - P_3 - c_{00}), \\ l_1 &= \frac{1}{2\, q_{10}\, q_{21}} \left[q_{20}\, (P_1 - P_3 - c_{00}) + q_{10}\, (P_3 - P_2) + q_{10}\, c_{11} \right] \\ &= \frac{1}{2\, q_{21}} \left[c_{10}\, l_0 + (P_3 - P_2) + c_{11} \right], \\ l_\perp^2 &= l_0^2 - l_1^2 - P_3 - c_{20} \end{aligned} \tag{3.16}$$

with

$$\begin{aligned} c_{00} &= q_{10}^2 - m_1^2 + m_3^2, \\ c_{10} &= 2\, q_{20}, \\ c_{11} &= m_3^2 - m_2^2 + q_{20}^2 - q_{21}^2, \\ c_{20} &= m_3^2 - i\rho. \end{aligned} \tag{3.17}$$

Again, as in the case of two-point functions, one substitutes Eq. (3.16) into Eq. (3.14) and obtains

$$C^{ijk}_{t_1 t_2 t_3} = \sum_{m,n,r=0}^{i+j+k} \mathcal{K}_{mnr} \int d^D l\, P_1^{(m-t_1)} P_2^{(n-t_2)} P_3^{(r-t_3)} \tag{3.18}$$

with the $\mathcal{K}_{mnr}$ being simple functions of masses and components of external momenta. More explicitly, the integrand on the right-hand side of Eq. (3.18) contains terms like

$$\frac{1}{P_1^{n_1}\, P_2^{n_2}\, P_3^{n_3}}, \quad \frac{P_i^{n_i}\, P_j^{n_j}}{P_k^{n_k}}, \quad \text{and} \quad \frac{P_i^{n_i}}{P_j^{n_j}\, P_k^{n_k}} \tag{3.19}$$

with $i \neq j \neq k$ and positive n_i. Other possible combinations lead to a vanishing integral.

The first group of terms can be obtained from derivatives of scalar three-point functions $C_0 = C^{000}_{111}$ with respect to the masses m_i^2. The analytical result of

$$C_0(q_1, q_2; m_1, m_2, m_3) \;=\; \int d^D l\, \frac{1}{P_1\, P_2\, P_3} \tag{3.20}$$

is given in the appendix A.3.

For the last two cases, using Eq. (3.15) one can expand the numerator in terms of propagators P_{n_j} in the denominator. This expansion is always possible and reduces the second group of terms to one-point functions. Similarly, the third group of terms can be reduced to one-loop two-point functions. It should be noted that in this step we encounter two kinds of one-loop two-point functions. The first kind is the one-loop two-point function with one parallel dimension that was already appeared in the previous section. The second kind is the one-loop two-point function with two parallel dimensions of the internal momentum l that is not trivial and will be given in section 3.2.3 as a separate case.

3.2.2 The case $q_{10}\, q_{21} = 0$

In this case the expansions in Eq. (3.16) cannot be used. However, Eq. (3.14) can be reduced by a partial fraction decomposition. First, we consider the case $q_{10} = 0$ with an arbitrary value of q_{21}.

The case $q_{10} = 0$

Eq. (3.15) is simplified to

$$\begin{aligned} P_1 &= l_0^2 - l_1^2 - l_\perp^2 - m_1^2 + i\rho, \\ P_2 &= l_0^2 - l_1^2 - l_\perp^2 + 2l_0 q_{20} - 2l_1 q_{21} + q_{20}^2 - q_{21}^2 - m_2^2 + i\rho, \\ P_3 &= l_0^2 - l_1^2 - l_\perp^2 - m_3^2 + i\rho. \end{aligned} \tag{3.21}$$

If $m_1 \neq m_3$, partial fraction decomposition leads to a separation into terms which contain only two propagators

$$\begin{aligned} C^{ijk}_{t_1 t_2 t_3}\Big|_{q_{10}=0} &= \prod_{f=1}^{3} \left(\frac{1}{(t_f-1)!} \frac{d^{t_f-1}}{d(m_f^2)^{t_f-1}} \right) \frac{1}{m_1^2 - m_3^2} \int d^D l \, \frac{l_0^i \, l_1^j \, l_\perp^k}{P_1(m_1^2)\, P_2(m_2^2)} \\ &- \prod_{f=1}^{3} \left(\frac{1}{(t_f-1)!} \frac{d^{t_f-1}}{d(m_f^2)^{t_f-1}} \right) \frac{1}{m_1^2 - m_3^2} \int d^D l \, \frac{l_0^i \, l_1^j \, l_\perp^k}{P_2(m_2^2)\, P_3(m_3^2)}. \end{aligned} \tag{3.22}$$

If $m_1 = m_3$, two propagators are equal and it is sufficient to calculate

$$C^{ijk}_{t_1 t_2 t_3}\Big|_{q_{10}=0,\, m_1=m_3} = \int d^D l \, \frac{l_0^i \, l_1^j \, l_\perp^k}{P_1^{t_1+t_3}(m_1^2)\, P_2^{t_2}(m_2^2)}. \tag{3.23}$$

The three-point integrals are then reduced to combinations of two-point integrals with two parallel dimensions. We will treat this class of two-point functions in section 3.2.3.

The case $q_{10} \neq 0$ and $q_{21} = 0$

In this special case, the integrals collapse from two to one parallel dimension and Eq. (3.15) reads

$$\begin{aligned} P_1 &= (l_0 + q_{10})^2 - l_\perp'^2 - m_1^2 + i\rho, \\ P_2 &= (l_0 + q_{20})^2 - l_\perp'^2 - m_2^2 + i\rho, \\ P_3 &= l_0^2 - l_\perp'^2 - m_3^2 + i\rho \end{aligned} \tag{3.24}$$

where the components l_1 and $l_\perp$ can be combined into $l'_\perp$ to form a new $(D-1)$-dimensional orthogonal subspace with

$$l_\perp'^2 = l_1^2 + l_\perp^2. \tag{3.25}$$

Then the integral can be reduced to a simpler integral with only one parallel dimension

$$C^{ijk}_{t_1 t_2 t_3}\Big|_{q_{21}=0} = \int d^D l \, \frac{l_0^i \, l_\perp^{j+k}}{P_1^{t_1} P_2^{t_2} P_3^{t_3}}. \tag{3.26}$$

Again, one can use the same procedure as in the general case

$$\begin{aligned} l_0 \to l_0(P_1, P_3, m_i, q_{10}) &= \frac{1}{2q_{10}}(P_1 - P_3 + m_1^2 - m_3^2 - q_{10}^2), \\ l_\perp \to l_\perp(P_1, P_3, m_i, q_{10}) &= l_0^2 - P_3 - m_3^2 + i\rho \end{aligned} \tag{3.27}$$

that reduces Eq. (3.26) to a form similar to Eq. (3.18).

3.2.3 The two-point integral with two parallel dimensions $\mathbb{B}^{ijk}_{t_1 t_2}$

In the preceding section we have shown that general one-loop three-point tensor functions can be reduced to the usual scalar integrals and one new two-point function corresponding to a tensor component in a two-dimensional parallel space. Explicitly, this integral reads

$$\begin{aligned} \mathbb{B}^{ijk}_{t_1 t_2} &= \int d^D l \, \frac{l_0^i \, l_1^j \, l_\perp^k}{[(l_0 + q_{10})^2 - l_1^2 - l_\perp^2 - m_1^2 + i\rho]^{t_1} \, [(l_0 + q_{20})^2 - (l_1 + q_{21})^2 - l_\perp^2 - m_2^2 + i\rho]^{t_2}} \\ &= \int d^D l \, \frac{(l_0 - q_{10})^i \, l_1^j \, l_\perp^k}{[l_0^2 - l_1^2 - l_\perp^2 - m_1^2 + i\rho]^{t_1} \, [(l_0 + Q_0)^2 - (l_1 + Q_1)^2 - l_\perp^2 - m_2^2 + i\rho]^{t_2}} \end{aligned} \tag{3.28}$$

with $Q_0 = q_{20} - q_{10}$, $Q_1 = q_{21}$; $Q^\mu = q_2^\mu - q_1^\mu$. If $Q_1 = 0$, this integral reduces to the one-loop two-point function in one-dimensional parallel space as found in the previous section. If, on the other hand, $Q_1 \neq 0$, one can always find a Lorentz boost which transforms the integral into a reference frame where the transformed 4-momentum Q'^μ has: (a) only one non-zero component Q'_0 (or Q'_1) if Q is timelike (or spacelike), or (b) $Q'_0 = Q'_1$ if Q is lightlike. The loop momentum has to be boosted accordingly which, however, modifies only the numerator of the integrand in Eq. (3.28). Explicitly, considering the boost

$$\begin{pmatrix} l_0 \\ l_1 \end{pmatrix} = \begin{pmatrix} \gamma & \gamma\beta \\ \gamma\beta & \gamma \end{pmatrix} \begin{pmatrix} l'_0 \\ l'_1 \end{pmatrix}, \tag{3.29}$$

then the three sub-cases are treated as follows.

The timelike case $Q_0^2 - Q_1^2 > 0$

In this case, under the transformation in Eq. (3.29) the integral will be reduced to one-loop two-point functions with a one-dimensional parallel space as found in the previous section:

$$\mathbb{B}^{ijk}_{t_1 t_2} = \int d^D l' \, \frac{\left[\gamma\,(l'_0 + \beta\, l'_1) - q_{10}\right]^i \; \left[\gamma\,(\beta\, l'_0 + l'_1)\right]^j \; l'^k_\perp}{\left[l'^2_0 - l'^2_1 - l'^2_\perp - m_1^2 + i\rho\right]^{t_1} \; \left[(l'_0 + P)^2 - l'^2_1 - l'^2_\perp - m_2^2 + i\rho\right]^{t_2}} \tag{3.30}$$

with $P = \sqrt{Q_0^2 - Q_1^2}$, $\gamma = Q_0/P$ and $\beta = Q_1/Q_0$.

The spacelike case, $Q_0^2 - Q_1^2 < 0$

In this case, the boost with $P = \sqrt{Q_1^2 - Q_0^2}$, $\gamma = Q_1/P$ and $\beta = Q_0/Q_1$ transforms the integral into a reference frame in which

$$\mathbb{B}^{ijk}_{t_1 t_2} = \int d^D l' \, \frac{\left[\gamma\,(l'_0 + \beta\, l'_1) - q_{10}\right]^i \; \left[\gamma\,(\beta\, l'_0 + l'_1)\right]^j \; l'^k_\perp}{\left[l'^2_0 - l'^2_1 - l'^2_\perp - m_1^2 + i\rho\right]^{t_1} \; \left[l'^2_0 - (l'_1 + P)^2 - l'^2_\perp - m_2^2 + i\rho\right]^{t_2}}. \tag{3.31}$$

The components l'_0 and $l'_\perp$ can be combined to form a new $(D-1)$-dimensional orthogonal subspace while l_1 spans the parallel subspace. Using the same procedure as in the previous sections one can reduce the integral completely to scalar one- and two-point functions.

The lightlike case, $Q_0^2 - Q_1^2 = 0$

In this case the transformation in Eq. (3.29) is singular and the integral $\mathbb{B}^{ijk}_{t_1 t_2}$ becomes

$$\mathbb{B}^{ijk}_{t_1 t_2} = \int d^D l \, \frac{(l_0 - q_{10})^i \; l_1^j \; l_\perp^k}{[l_0^2 - l_1^2 - l_\perp^2 - m_1^2 + i\rho]^{t_1} \, [(l_0 + Q_0)^2 - (l_1 + Q_0)^2 - l_\perp^2 - m_2^2 + i\rho]^{t_2}}. \quad (3.32)$$

By solving the system of equations

$$\begin{aligned} P_1 &= l_0^2 - l_1^2 - l_\perp^2 - m_1^2 + i\rho, \\ P_2 &= (l_0 + Q_0)^2 - (l_1 + Q_0)^2 - l_\perp^2 - m_2^2 + i\rho, \end{aligned} \quad (3.33)$$

one obtains

$$\begin{aligned} l_\perp^2 &= l_0^2 - l_1^2 - P_1 - m_1^2 + i\rho, \\ l_1 &= \frac{P_1 - P_2 + m_2^2 - m_1^2}{2Q_0} + l_0. \end{aligned} \quad (3.34)$$

Inserting Eq. (3.34) into Eq. (3.32) and expanding the numerator of the integrand, the integral will be reduced to scalar one-point functions and an integral of the form

$$\mathbb{B}^{i}_{t_1 t_2} = \int d^D l \, \frac{(l_0)^i}{[l_0^2 - l_1^2 - l_\perp^2 - m_1^2 + i\rho]^{t_1} \, [(l_0 + Q_0)^2 - (l_1 + Q_0)^2 - l_\perp^2 - m_2^2 + i\rho]^{t_2}}. \quad (3.35)$$

This integrals cannot be reduced further to a scalar integral by the same method as before. In order to evaluate the integral, one has to introduce a parameter ρ' to shift the external momentum out of the light-cone. For instance, by shifting the first occurrence of Q_0 in Eq. (3.35) to $Q_0 + \rho'$ one can perform the reduction as for the time-like case. The result of the integral will be obtained by performing the limit $\rho' \to 0$ afterward.

This completes the description of our algorithm for the tensor reduction of one-loop, one-, two- and three-point functions.

To summarize, all one-loop one-, two- and three-point tensor integrals can be reduced to a set of only three basic scalar integrals A_0, B_0 and C_0. The explicit analytical results of the basic scalar integrals as well as their UV- and IR-properties can be found in the appendix A or in the literature [11, 24].

3.3 Tests and comparisons of one-loop integrals

The algorithm to calculate one-loop one-, two- and three-point integrals is implemented in the program package XLOOPS-GiNaC (see chapter 5). As a very important part of our work, we performed various tests and comparisons. In order to do that we have used XLOOPS-GiNaC to calculate various physical processes. In chapter 6, we present an application of XLOOPS-GiNaC in evaluating the g_2 form factor from a theory with spontaneously broken color. We then compare the result of XLOOPS-GiNaC and get agreement with the known results in various limiting cases.

The program package is also applied to calculate the electroweak radiative corrections to the physical process $t \to b\ W^-$ [33]. In this calculation, we used XLOOPS-GiNaC to evaluate 18 electroweak one-loop three-point and numerous one-loop two-point diagrams involving a big number of different mass scales. The results of the calculation agree with the results of [34] analytically as well as numerically.

Chapter 4

Two-Loop Two-Point Tensor Integrals

In this chapter, an algorithm for two-loop two-point tensor integrals will be discussed. In general, there are ten different one-particle irreducible (1PI) topologies which can be classified into two groups. The integrals of the first group shown in Fig. 4.1 can be factorized into a product of one-loop functions and thus they are trivial. The main interest of this chapter is in the second group shown in Fig. 4.2.

Since in this chapter we are going to discuss two-point integrals only, it is sufficient to separate components of all momenta into one parallel and $D-1$ orthogonal dimensions. Thus a complete set of tensor indices of the integral representation Eq. (2.25) comprises $p_0, p_\perp, r_0, r_\perp$ and p_z.

4.1 The factorizing topologies

Fig. 4.1 shows five trivial topologies. Their associated integrals can be factorized into products of two one-loop two-point or one-point integrals which were already discussed in the last chapter. The Standard Model possesses only diagrams (a), (b) and (e). The diagrams (c) and (d) which have five- and six-line vertices appear only in effective theories.

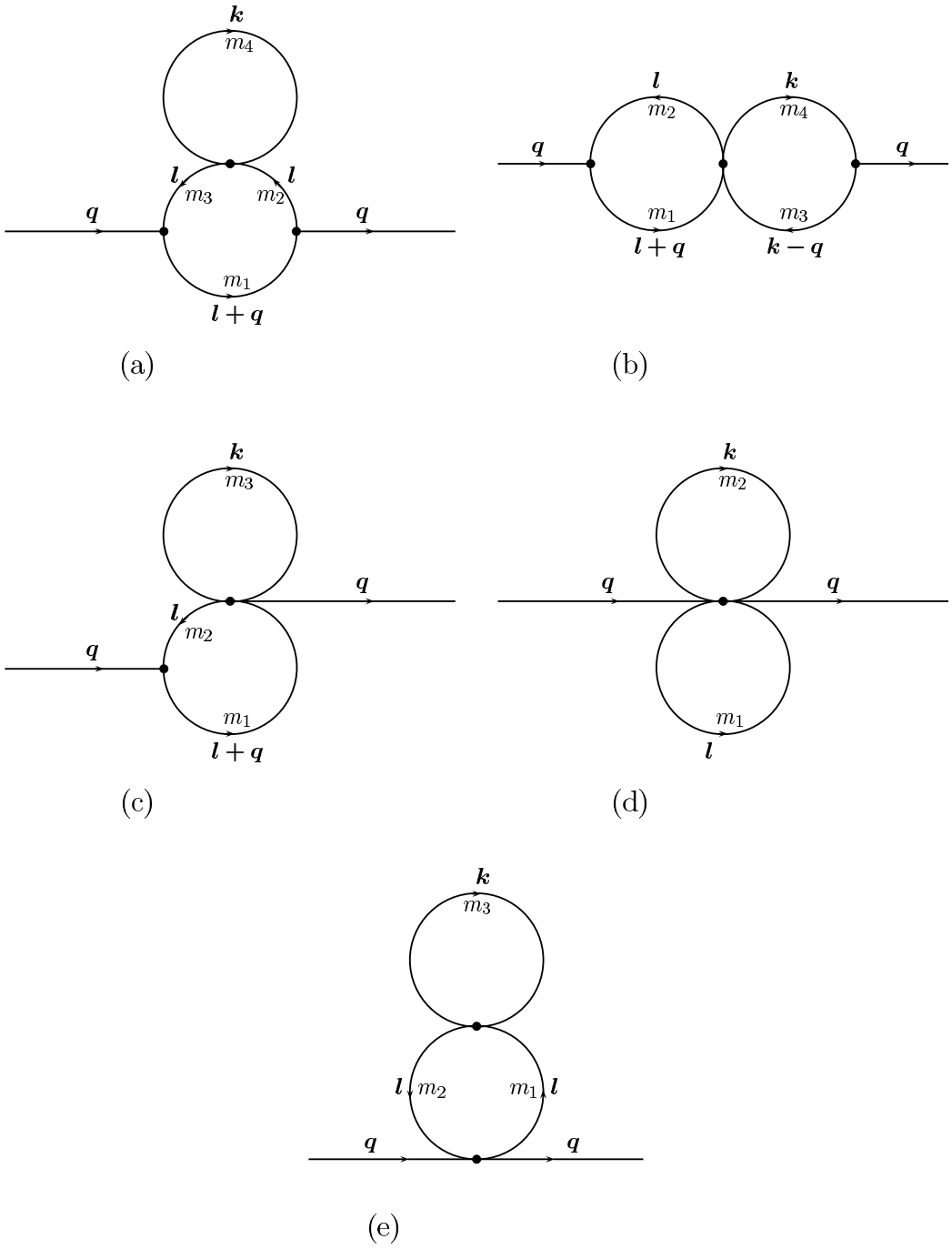


Figure 4.1: Factorizing two-loop two-point topologies.

In our calculations, the diagrams (c) and (d) will appear after tensor reduction of the non-trivial diagrams. The diagram (e) is the factorizing two-loop tadpole topology. It can be evaluated as a special case of the diagram (a) with $m_2 = m_3$ and $t_1 = 0$. The tensor integrals associated with the trivial diagrams have no propagator which contains both loop momenta ($n_m = 0$) and can be described by

$$T^{(2)\,(p_0,p_\perp;r_0,r_\perp;p_z)}_{t_1\ldots t_{n_l};\ ;t_{n_l+1}\ldots t_{n_l+n_k}} \quad = \tag{4.1}$$

$$\int d^Dl \int d^Dk \frac{l_0^{p_0}\ l_\perp^{p_\perp}\ k_0^{r_0}\ k_\perp^{r_\perp}\ z^{p_z}}{\prod\limits_{s=1}^{n_l}\left[(l+q_s)^2 - m_s^2 + i\rho\right]^{t_s} \prod\limits_{s=n_l+1}^{n_l+n_k}\left[(k+q_s)^2 - m_s^2 + i\rho\right]^{t_s}}.$$

Here, we use the same notation as shown in Eq. (2.25). The momenta q_s can take the values q, 0 or $-q$ as presented in Fig. 4.1.

Because the integrand on the right-hand side of Eq. (4.1) does not contain propagators which mix the internal momenta l and k, one can use the relations in Eq. (2.9) and Eq. (2.12) to decompose the integral into a product of one-loop functions

$$\begin{aligned} T^{(2)\,(p_0,p_\perp;r_0,r_\perp;p_z)}_{t_1\ldots t_{n_l};\ ;t_{n_l+1}\ldots t_{n_l+n_k}} \quad &= \quad \frac{\pi^{-\frac{1}{2}}\Gamma(\frac{D-J}{2})}{\Gamma(\frac{D-J-1}{2})} \int\limits_{-1}^{1} dz\ (1-z^2)^{\frac{D-4}{2}}\ z^{p_z} \\ &\times \quad T^{(1)\,(p_0,p_\perp)}_{t_1\ldots t_{n_l}}(l_0,l_\perp)\ T^{(1)\,(r_0,r_\perp)}_{t_{n_l+1}\ldots t_{n_l+n_k}}(k_0,k_\perp) \end{aligned} \tag{4.2}$$

where $z = \cos\theta$ and θ is, as before, the angle between the momenta $\vec{l}_\perp$ and $\vec{k}_\perp$. The functions $T^{(1)}$ are the one-loop one-, two- or three-point integrals discussed in the last chapter. The z-integral is given by

$$\int\limits_{-1}^{1} dz\ (1-z^2)^{\frac{D-4}{2}}\ z^{p_z} = \frac{(1+(-1)^{p_z})}{2}\ \mathcal{B}\left(\frac{D-2}{2}, \frac{1+p_z}{2}\right) \tag{4.3}$$

where $\mathcal{B}$ is Euler's beta function defined by

$$\mathcal{B}(x,y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}. \tag{4.4}$$

In brief, factorizing integrals can be easily and analytically calculated by using the one-loop procedure. In the remaining part of this chapter, we will discuss the calculation of non-trivial topologies shown in Fig. 4.2.

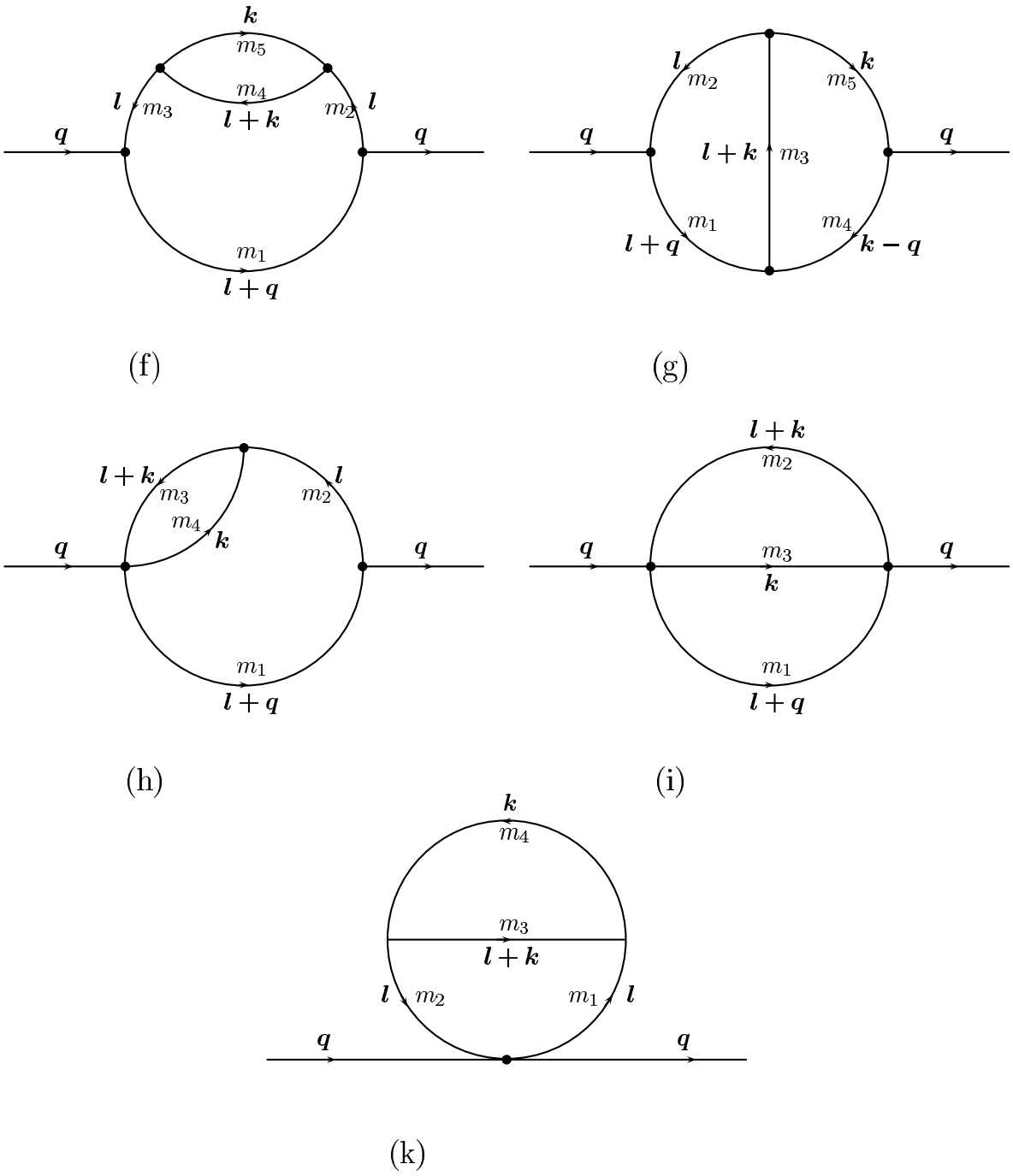


Figure 4.2: The non-trivial two-loop two-point topologies.

4.2 The non-trivial topologies

The explicit formulae of integrals associated with the four non-trivial topologies in Fig. 4.2 are

$$T^{(f)} := T^{(2)\ (p_0,p_\perp,r_0,r_\perp;\ p_z)}_{t_1\,t_2\,t_3;\ t_4;\ t_5} = \tag{4.5}$$

$$\int d^Dl \int d^Dk \frac{l_0^{p_0}\ l_\perp^{p_\perp}\ k_0^{r_0}\ k_\perp^{r_\perp}\ z^{p_z}}{[(l+q)^2-m_1^2]^{t_1}\ [l^2-m_2^2]^{t_2}\ [l^2-m_3^2]^{t_3}\ [(l+k)^2-m_4^2]^{t_4}\ [k^2-m_5^2]^{t_5}},$$

$$T^{(g)} := T^{(2)\ (p_0,p_\perp,r_0,r_\perp;\ p_z)}_{t_1\,t_2;\ t_3;\ t_4\,t_5} = \tag{4.6}$$

$$\int d^Dl \int d^Dk \frac{l_0^{p_0}\ l_\perp^{p_\perp}\ k_0^{r_0}\ k_\perp^{r_\perp}\ z^{p_z}}{[(l+q)^2-m_1^2]^{t_1}\ [l^2-m_2^2]^{t_2}\ [(l+k)^2-m_3^2]^{t_3}\ [(k-q)^2-m_4^2]^{t_4}\ [k^2-m_5^2]^{t_5}},$$

$$T^{(h)} := T^{(2)\ (p_0,p_\perp,r_0,r_\perp;\ p_z)}_{t_1\,t_2;\ t_3;\ t_4} = \tag{4.7}$$

$$\int d^Dl \int d^Dk \frac{l_0^{p_0}\ l_\perp^{p_\perp}\ k_0^{r_0}\ k_\perp^{r_\perp}\ z^{p_z}}{[(l+q)^2-m_1^2]^{t_1}\ [l^2-m_2^2]^{t_2}\ [(l+k)^2-m_3^2]^{t_3}\ [k^2-m_4^2]^{t_4}},$$

$$T^{(i)} := T^{(2)\ (p_0,p_\perp,r_0,r_\perp;\ p_z)}_{t_1;\ t_2;\ t_3} = \tag{4.8}$$

$$\int d^Dl \int d^Dk \frac{l_0^{p_0}\ l_\perp^{p_\perp}\ k_0^{r_0}\ k_\perp^{r_\perp}\ z^{p_z}}{[(l+q)^2-m_1^2]^{t_1}\ [(l+k)^2-m_2^2]^{t_2}\ [k^2-m_3^2]^{t_3}},$$

$$T^{(k)} := T^{(2)\ (p_0,p_\perp,r_0,r_\perp;\ p_z)}_{t_1\,t_2;\ t_3;\ t_4} = \tag{4.9}$$

$$\int d^Dl \int d^Dk \frac{l_0^{p_0}\ l_\perp^{p_\perp}\ k_0^{r_0}\ k_\perp^{r_\perp}\ z^{p_z}}{[l^2-m_1^2]^{t_1}\ [l^2-m_2^2]^{t_2}\ [(l+k)^2-m_3^2]^{t_3}\ [k^2-m_4^2]^{t_4}}$$

where $p_0, p_\perp, r_0, r_\perp, p_z$ are non-negative integers and t_i are positive integers.
The above five integrals are not independent. In fact the integral $T^{(k)}$ can be calculated as a special case of $T^{(f)}$ with $t_1 = 0$. The integral $T^{(f)}$ in turn, can be reduced to a sum

of functions $T^{(h)}$. By partial fractioning one obtains

$$
\begin{aligned}
T^{(f)} &= \frac{1}{(t_2-1)!(t_3-1)!}\left(\frac{\partial}{\partial m_2^2}\right)^{t_2-1}\left(\frac{\partial}{\partial m_3^2}\right)^{t_3-1} \qquad (4.10)\\
&\int d^Dl \int d^Dk \; \frac{l_0^{p_0}\, l_\perp^{p_\perp}\, k_0^{r_0}\, k_\perp^{r_\perp}\, z^{p_z}}{[(l+q)^2-m_1^2]^{t_1}\,[(l+k)^2-m_4^2]^{t_4}\,[k^2-m_5^2]^{t_5}} \;\times\\
&\left(\frac{1}{[l^2-m_2^2]}-\frac{1}{[l^2-m_3^2]}\right)\frac{1}{(m_2^2-m_3^2)}\\
&= \sum_{k=0}^{t_2-1}\left(\begin{array}{c} t_2-1 \\ k \end{array}\right)\frac{(t_2-1-k)!}{(t_2-1)!(t_3-1)!}\left\{\left(\frac{\partial}{\partial m_3^2}\right)^{t_3-1}\left(\frac{\partial}{\partial m_2^2}\right)^{k}\frac{1}{(m_2^2-m_3^2)}\right\} \;\times\\
&\int d^Dl \int d^Dk \frac{l_0^{p_0}\, l_\perp^{p_\perp}\, k_0^{r_0}\, k_\perp^{r_\perp}\, z^{p_z}}{[(l+q)^2-m_1^2]^{t_1}\,[l^2-m_2^2]^{t_2-k}\,[(l+k)^2-m_4^2]^{t_4}\,[k^2-m_5^2]^{t_5}}\\
&- \sum_{k=0}^{t_3-1}\left(\begin{array}{c} t_3-1 \\ k \end{array}\right)\frac{(t_3-1-k)!}{(t_2-1)!(t_3-1)!}\left\{\left(\frac{\partial}{\partial m_3^2}\right)^{k}\left(\frac{\partial}{\partial m_2^2}\right)^{t_2-1}\frac{1}{(m_2^2-m_3^2)}\right\} \;\times\\
&\int d^Dl \int d^Dk \frac{l_0^{p_0}\, l_\perp^{p_\perp}\, k_0^{r_0}\, k_\perp^{r_\perp}\, z^{p_z}}{[(l+q)^2-m_1^2]^{t_1}\,[l^2-m_3^2]^{t_3-k}\,[(l+k)^2-m_4^2]^{t_4}\,[k^2-m_5^2]^{t_5}}
\end{aligned}
$$

where the integral under each summation sign is actually $T^{(h)}$.

The three remaining integrals $T^{(g)}$, $T^{(h)}$ and $T^{(i)}$ are in general UV-divergent. So far, these integrals can be analytically solved only in some special cases. In this thesis, we will use the method proposed by D. Kreimer which enables us to calculate analytically the UV-divergent part of the integrals [12]. The remaining UV-finite part will be analytically integrated as far as possible by using the residue theorem. One finally ends up with a two-dimensional integral representation. This representation is suitable for numerical integration.[1]

[1]The procedure to regulate the IR-divergence by using DR is not discussed in this thesis. It should be a topic for future work.

4.2.1 Tensor reduction

In this section, the integrals $T^{(g)}$, $T^{(h)}$ and $T^{(i)}$ will be reduced to a subset of three integrals which are easier to integrate by the residue theorem. The method of tensor reduction is the same for all three integrals and similar to the one-loop case. Let us consider the case of the so-called master topology

$$T^{(g)}(q^2) = \int d^D l \int d^D k \frac{l_0^{p_0}\, l_\perp^{p_\perp}\, k_0^{r_0}\, k_\perp^{r_\perp}\, z^{p_z}}{P_1^{t_1}(l+q)\, P_2^{t_2}(l)\, P_3^{t_3}(l+k)\, P_4^{t_4}(k-q)\, P_5^{t_5}(k)} \tag{4.11}$$

with

$$\begin{aligned} P_1(l+q) &= (l+q)^2 - m_1^2 + i\,\rho, \\ P_2(l) &= l^2 - m_2^2 + i\,\rho, \\ P_3(l+k) &= (l+k)^2 - m_3^2 + i\,\rho, \\ P_4(k-q) &= (k-q)^2 - m_4^2 + i\,\rho, \\ P_5(k) &= k^2 - m_5^2 + i\,\rho. \end{aligned} \tag{4.12}$$

In order to calculate the integral in Eq. (4.11), one first inserts

$$l_\perp\, k_\perp\, z \;=\; \frac{1}{2}\left[-P_3 + (l_0+k_0)^2 - l_\perp^2 - k_\perp^2 - m_3^2 - i\,\rho\right] \tag{4.13}$$

into the numerator of the integral in Eq. (4.11). One obtains

$$\begin{aligned} T^{(g)}(q^2) &= \int d^D l \int d^D k \sum_{n=0}^{p_z} \left(\begin{array}{c} p_z \\ n \end{array} \right) \frac{(-1)^n}{2^{p_z}}\, P_3^n\, \left[(l_0+k_0)^2 - l_\perp^2 - k_\perp^2 - m_3^2\right]^{p_z-n} \\ &\qquad \times\; \frac{l_0^{p_0}\, k_0^{r_0}\, l_\perp^{p_\perp - p_z}\, k_\perp^{r_\perp - p_z}}{P_1^{t_1}(l+q)\, P_2^{t_2}(l)\, P_3^{t_3}(l+k)\, P_4^{t_4}(k-q)\, P_5^{t_5}(k)} \qquad (4.14) \\ &= \int d^D l \int d^D k \sum_{n=0, n<t_3}^{p_z} \frac{f_n(m_3^2)\;\; l_0^{p_0'}\, k_0^{r_0'}\, l_\perp^{p_\perp'}\, k_\perp^{r_\perp'}}{P_1^{t_1}(l+q)\, P_2^{t_2}(l)\, P_3^{t_3-n}(l+k)\, P_4^{t_4}(k-q)\, P_5^{t_5}(k)} \\ &\quad + \;\theta(p_z - t_3) \int d^D l \int d^D k \sum_{n \geq t_3}^{p_z} \frac{g_n(m_3^2)\;\; l_0^{p_0'}\, k_0^{r_0'}\, l_\perp^{p_\perp'}\, k_\perp^{r_\perp'}\, P_3^{n-t_3}}{P_1^{t_1}(l+q)\, P_2^{t_2}(l)\, P_4^{t_4}(k-q)\, P_5^{t_5}(k)} \end{aligned}$$

where θ is the step function, $f_n(m_3^2)$ and $g_n(m_3^2)$ are functions of m_3^2. Note also that the primed exponents $p'_0, p'_\perp, r'_0, r'_\perp$ depend on $p_0, p_\perp, r_0, r_\perp$, p_z and on n. The second term of the above equation is in the form of the factorizing integrals discussed in section 4.1.

The first term is a sum of integrals of the form

$$\int d^D l \int d^D k \frac{l_0^{p'_0}\, k_0^{r'_0}\, l_\perp^{p'_\perp}\, k_\perp^{r'_\perp}}{P_1^{t_1}(l+q)\, P_2^{t_2}(l)\, P_3^{t_3}(l+k)\, P_4^{t_4}(k-q)\, P_5^{t_5}(k)}. \tag{4.15}$$

Note here that $p'_\perp = p_\perp - p_z$ and $r'_\perp = r_\perp - p_z$ are even (see the remarks at the end of chapter 2) so that one can insert

$$\begin{aligned} l_0 = l_0(P_1, P_2) \;\; , \quad l_\perp^2 = l_\perp^2(P_1, P_2) \; , \\ k_0 = k_0(P_4, P_5) \;\; , \quad k_\perp^2 = k_\perp^2(P_4, P_5) \end{aligned} \tag{4.16}$$

into the numerator. After cancellation of P_i's one obtains various simpler integrals with either reduced numerator or a reduced number of propagators. We discuss the possible terms in turn:

- The scalar term

$$J_0 \;=\; \int d^D l \int d^D k \frac{1}{P_1^{t_1}(l+q)\, P_2^{t_2}(l)\, P_3^{t_3}(l+k)\, P_4^{t_4}(k-q)\, P_5^{t_5}(k)}. \tag{4.17}$$

  This integral is finite so that one can perform a numerical integration directly. A two-fold representation for the numerical integration is discussed in [11] and described in detail in section 4.2.7.

Remark: We will use the same symbols $t_1 \ldots t_5$, p_0, r_0 and $P_1, \ldots, P_5$ when writing down the definitions of J_0 and later of J_1 and J_2. It should be understood that the symbols only have the same form as those in the original integral Eq. (4.15) and do not necessarily have the same values as had occurred in the original tensor integral from which the reduction algorithm started. For instance, when writing

$$P_i^{t_i}(x) = [x^2 - m_i^2 + i\rho]^{t_i}$$

we imply that m_i is some mass, t_i is some positive integer number and x stands for a combination of momenta l, k and q.

- The terms with the integrands of the form

$$\frac{P_1^{s_1}(l+q)}{P_2^{t_2}(l)\,P_3^{t_3}(l+k)\,P_4^{t_4}(k-q)\,P_5^{t_5}(k)}\,,\frac{P_2^{s_2}(l)}{P_1^{t_1}(l+q)\,P_3^{t_3}(l+k)\,P_4^{t_4}(k-q)\,P_5^{t_5}(k)},$$
$$\frac{P_4^{s_4}(k-q)}{P_1^{t_1}(l+q)\,P_2^{t_2}(l)\,P_3^{t_3}(l+k)\,P_5^{t_5}(k)}\,,\frac{P_5^{s_5}(k)}{P_1^{t_1}(l+q)\,P_2^{t_2}(l)\,P_3^{t_3}(l+k)\,P_4^{t_4}(k-q)} \tag{4.18}$$

will be reduced to integrals of the form

$$J_1 \;=\; \int d^Dl \int d^Dk \frac{k_0^{r_0}}{P_1^{t_1}(l+q)\,P_2^{t_2}(l)\,P_3^{t_3}(l+k)\,P_4^{t_4}(k)} \tag{4.19}$$

after inserting P_i's from Eq. (4.12) into numerators, shifting and/or relabeling momenta and rearranging indices. Here, again, the indices occurring in J_1 have to be understood as remarked above.

- Similarly the terms with the integrands

$$\begin{gathered}\frac{P_1^{s_1}(l+q)\,P_4^{s_4}(k-q)}{P_2^{t_2}(l)\,P_3^{t_3}(l+k)\,P_5^{t_5}(k)}\,,\quad \frac{P_1^{s_1}(l+q)\,P_5^{s_5}(k)}{P_2^{t_2}(l)\,P_3^{t_3}(l+k)\,P_4^{t_4}(k-q)},\\ \frac{P_2^{s_2}(l)\,P_4^{s_4}(k-q)}{P_1^{t_1}(l+q)\,P_3^{t_3}(l+k)\,P_5^{t_5}(k)}\,,\quad \frac{P_2^{s_2}(l)\,P_5^{s_5}(k)}{P_1^{t_1}(l+q)\,P_3^{t_3}(l+k)\,P_4^{t_4}(k-q)}\end{gathered} \tag{4.20}$$

will be reduced to integrals of the form [2]

$$J_2 \;=\; \int d^Dl \int d^Dk \frac{l_0^{p_0}\,k_0^{r_0}}{P_1^{t_1}(l+q)\,P_2^{t_2}(l+k)\,P_3^{t_3}(k)}. \tag{4.21}$$

- The terms with the integrands

$$\begin{gathered}\frac{P_1^{s_1}(l+q)\,P_2^{s_2}(l)}{P_3^{t_3}(l+k)\,P_4^{t_4}(k-q)P_5^{t_5}(k)}\,,\quad \frac{P_4^{s_4}(k-q)\,P_5^{s_5}(k)}{P_1^{t_1}(l+q)\,P_2^{t_2}(l)\,P_3^{t_3}(l+k)},\\ \frac{P_1^{s_1}(l+q)\,P_2^{s_2}(l)\,P_4^{s_4}(k-q)}{P_3^{t_3}(l+k)\,P_5^{t_5}(k)}\,,\quad \frac{P_1^{s_1}(l+q)\,P_2^{s_2}(l)\,P_5^{s_5}(k)}{P_3^{t_3}(l+k)\,P_4^{t_4}(k-q)},\\ \frac{P_1^{s_1}(l+q)\,P_4^{s_4}(k-q)\,P_5^{s_5}(k)}{P_2^{t_2}(l)\,P_3^{t_3}(l+k)}\,,\quad \frac{P_2^{s_2}(l)\,P_4^{s_4}(k-q)\,P_5^{s_5}(k)}{P_1^{t_1}(l+q)\,P_3^{t_3}(l+k)}\end{gathered} \tag{4.22}$$

[2] The topology of two-point diagrams with three propagators is called *sunrise topology*. Its associated integrals, such as $T^{(i)}$ or J_2, are called sunrise integrals.

will be reduced to sums of products of one-loop two-point and one-loop tadpole integrals after inserting P_i's from Eq. (4.12) into numerators and then shifting momenta.

- The remaining terms vanish due to the property of the dimensional regularization method

$$\int d^D l \int d^D k \frac{P_1^{s_1}(l+q)\, P_2^{s_2}(l)\, P_4^{s_4}(k-q)\, P_5^{s_5}(k)}{P_3^{t_3}(l+k)} \;=\; 0. \tag{4.23}$$

By the same procedure, the two other tensor integrals which are associated with topologies (h) and (i) can also be reduced to expressions containing J_1, J_2 and products of one-loop functions.

At this step, the three integrals $T^{(g)}$, $T^{(h)}$ and $T^{(i)}$ are reduced to a set of the three integrals J_0, J_1 and J_2. The integrals J_1 and J_2 are, in general, UV-divergent. In the next sections, the procedure of separating the UV-divergent and UV-finite parts of J_1 and J_2 is discussed.

4.2.2 UV-divergences of two-loop integrals

This section is dedicated to a brief discussion of different kinds of UV-divergences of two-loop two-point integrals. The understanding of these divergences is important when solving the integrals J_1 and J_2 in the next steps.

In general, a two-loop integral $T^{(2)\,\mu_1\ldots\mu_p,\nu_1\ldots\nu_r}_{t_1\ldots t_{n_l};\ldots t_{n_l+n_m};\ldots t_{n_l+n_m+n_k}}$ (c.f. Eq. (2.2)) has the following UV-divergences:

- **Overall divergence:** This divergence appears when both $|l|$ and $|k|$ get large. By power counting, the **degree of overall divergence** is defined by

$$w := 2 \sum_{i=1}^{n_l+n_m+n_k} t_i - 2 \times 4 - p - r. \tag{4.24}$$

For example, the degree of overall divergence of the integral J_2 in Eq. (4.21) is

$$w = 2\sum_{i=1}^{3} t_i - 8 - p_0 - r_0. \tag{4.25}$$

If $w \leq 0$, the integral is divergent. However, unlike in the one-loop case, the condition $w > 0$ is not enough for the integral to be convergent. In the two-loop case, one needs also to consider sub-divergences.

- **Sub-divergences:** There are two kinds of sub-divergences:

  a. The divergence appears when $|l| \to \infty$ while $|k|$ is fixed,

  b. The divergence appears when $|k| \to \infty$ while $|l|$ is fixed.

  One can also define **degrees of sub-divergences** w_l and w_k

$$\begin{aligned} w_l &:= 2\sum_{i=1}^{n_l+n_m} t_i - 4 - p, \\ w_k &:= 2\sum_{i=n_l+1}^{n_l+n_m+n_k} t_i - 4 - r. \end{aligned} \tag{4.26}$$

  For instance, the integral J_2 in Eq. (4.21) has

$$\begin{aligned} w_l &= 2(t_1 + t_2) - 4 - p_0, \\ w_k &= 2(t_2 + t_3) - 4 - r_0. \end{aligned} \tag{4.27}$$

The conditions for a graph G (i.e for its associated integral) which contains divergent subgraphs to be convergent are proved in Weinberg's theorem [35]. The theorem can be summarized as follows

- Suppose that a *one-particle irreducible* (1PI) graph G and all its 1PI subgraphs have positive degrees of divergence. Then the graph is finite.

- Suppose that a 1PI graph G has positive degree of overall divergence, but it might have sub-divergences. Then the graph is finite after sub-divergences have been subtracted.

- If a 1PI graph G has non-positive degree of overall divergence w, then its overall counter term is polynomial in the external momenta of G of degree $-w$.

From Eq. (4.24) and Eq. (4.26) one obtains

$$w = w_l + w_k - 2\sum_{i=n_l+1}^{n_m} t_i. \tag{4.28}$$

From this relation, one observes that if an integral has

$$w_l \geq 2\sum_{i=n_l+1}^{n_m} t_i$$
$$\text{or} \tag{4.29}$$
$$w_k \geq 2\sum_{i=n_l+1}^{n_m} t_i\ ,$$

then it will be made finite by using subtraction terms to increase w_k (or w_l) to a positive value. It means that the integral is already finite after subtraction of its sub-divergences. The integral is called *non-degenerate*. Otherwise, if

$$w_l + w_k \leq 2\sum_{i=n_l+1}^{n_m} t_i, \tag{4.30}$$

then one needs not only subtraction terms to remove the sub-divergences but also subtraction terms for the overall divergence. The integral is called *degenerate*.

In the next sections, we will construct subtraction terms which can be used to remove both the overall and sub-divergences of the integrals J_1 and J_2. We will see that the construction of subtraction terms for the degenerate integrals is slightly different from the non-degenerate case. The difference is due to the fact that one has to deal with spurious IR-divergences which might be introduced by subtraction terms.

Note on overlapping divergences

It is possible that the considered graph contains two sub-graphs which may share an internal line. In this case, we cannot regard the sub-integrals associated with these sub-graphs

as independent integrals. It is said that the graph contains an *overlapping divergence*. In our calculation, this case appears in all topologies. However, as a result of the tensor reduction, the overlapping divergence remains only in the integral J_2. Historically, to treat overlapping divergent graphs was a difficult task. However, the method which we will present in this thesis works well also for the overlapping divergent integrals.

4.2.3 The integral J_1 and the subtraction procedure

As discussed in the previous sections, the tensor reduction results in three integrals J_0, J_1 and J_2. J_0 is UV-finite so that one can integrate it numerically [11]. To solve a non-degenerate integral such as J_1, a very efficient and practical method was introduced by D. Kreimer [12]. The method is known not to work in degenerate cases, such as J_2. However, with a modification that we propose in section 4.2.5, the method can be safely extended to solve degenerate integrals. In this section, we will use D. Kreimer's standard procedure to solve the integral of the type J_1 before going to discuss an extension for degenerate integrals. We start with the integral

$$\begin{aligned} J_1 &= \int d^D l \int d^D k \; \mathcal{J}^{r_0}_{1\,\{t_i\}} \,, \qquad (i = 1, 2, 3, 4) \\ \mathcal{J}^{r_0}_{1\,\{t_i\}} &= \frac{k_0^{r_0}}{P_1^{t_1}(l+q)\, P_2^{t_2}(l)\, P_3^{t_3}(l+k)\, P_4^{t_4}(k)} \end{aligned} \tag{4.31}$$

with $t_i \geq 1$, $r_0 \geq 0$ and

$$\begin{aligned} P_1 &= (l+q)^2 - m_1^2 + i\rho, \\ P_2 &= l^2 - m_2^2 + i\rho, \\ P_3 &= \widetilde{P}_3 - m_3^2 + i\rho, \\ P_4 &= \widetilde{P}_4 - m_4^2 + i\rho \end{aligned} \tag{4.32}$$

where we have introduced massless inverse propagators

$$\begin{aligned} \widetilde{P}_3 &= (l+k)^2, \\ \widetilde{P}_4 &= k^2. \end{aligned} \tag{4.33}$$

By power counting one obtains

$$\begin{aligned} w &= 2(t_1+t_2+t_3+t_4)-8-r_0, \\ w_l &= 2(t_1+t_2+t_3)-4 \quad \geq \quad 2t_3 \quad > \quad 0, \\ w_k &= 2(t_3+t_4)-4-r_0. \end{aligned} \tag{4.34}$$

If $w_k > 0$, the integral is finite since $t_1, t_2 \geq 1$ and $w_l > 0$. For this case a two-fold integral representation will be presented in Eq. (4.73) with $x_k = 0$. If $w_k \leq 0$ one can separate the integral into two parts, an UV-finite part which can be integrated numerically in $D = 4$ dimensions and an UV-divergent part which can be integrated analytically in D dimensions [12]. Let us introduce a subtraction term for the k-integral

$$\begin{aligned} K^{x_k} &:= \left[1 - \frac{P_3^{t_3}\, P_4^{t_4}}{\widetilde{P}_3^{t_3}\, \widetilde{P}_4^{t_4}}\right]^{x_k} \\ &= 1 + K_0^{x_k}, \\ K_0^{x_k} &= \sum_{n=1}^{x_k} \left(\begin{array}{c} x_k \\ n \end{array} \right) (-1)^n \left(\frac{P_3^{t_3} P_4^{t_4}}{\widetilde{P}_3^{t_3}\, \widetilde{P}_4^{t_4}}\right)^n . \end{aligned} \tag{4.35}$$

For $|k| \to \infty$, K behaves likes k^{-1} so that the factor K^{x_k} improves the degree of divergence for the k-integral by x_k. It is easy to prove that there always exists a value of x_k such that $w_k + x_k = 1$ and at the same time $w + x_k > 0$. This means, the integral

$$\int d^D l \int d^D k \; \mathcal{J}^{r_0}_{1\,\{t_i\}} \; K^{x_k} \tag{4.36}$$

is UV-finite. So, by introducing a subtraction term K^{x_k}, both the k sub-divergence and the overall divergence are removed at the same time.

This procedure can be safely applied also to the more general integral [3]

$$J^{p_0,r_0}_{1\,\{t_i\}} = \int d^D l \int d^D k \frac{l_0^{p_0}\; k_0^{r_0}}{P_1^{t_1}(l+q)\, P_2^{t_2}(l)\, P_3^{t_3}(l+k)\, P_4^{t_4}(k)} \tag{4.37}$$

with p_0, $r_0 \geq 0$ and if w_l satisfies the condition

$$w_l = 2(t_1+t_2+t_3)-4-p_0 \quad \geq \quad 2t_3. \tag{4.38}$$

[3] In fact, this integral can be always reduced back to J_1 and J_2. However to improve the performance of XLOOPS-GiNaC we will not reduce all integrals to J_1 and J_2 if they are non-degenerate ones.

This condition assures that the integral is non-degenerate and thus with a choice of x_k such that $x_k + w_k = 1$ one can find subtraction terms for Eq. (4.37) which render it convergent.

In order to obtain analytically integrable subtraction terms, we use the massless propagators $1/\widetilde{P}_3^{t_3}$ and $1/\widetilde{P}_4^{t_4}$ to construct K^{x_k}. However, this leads to spurious IR-divergences when applying the procedure to the degenerate case as we will see later. For now, we are safe to use the subtraction term K^{x_k} with $x_k + w_k = 1$ to separate UV-divergences from J_1.

Using Eq. (4.35) we can write

$$J_1 = \int d^4l \int d^4k \; \mathcal{J}^{r_0}_{1\,\{t_i\}} \; K^{x_k} - \int d^Dl \int d^Dk \; \mathcal{J}^{r_0}_{1\,\{t_i\}} \; K_0^{x_k}. \tag{4.39}$$

On the right-hand side, the first term is finite so that one can perform the numerical integration in $D = 4$ dimensions as discussed in section 4.2.7. The integration in the last term can be done analytically. To have a closer look, let us consider the integrand of this term:

$$\begin{aligned}
\mathcal{J}^{r_0}_{1\,\{t_i\}} \; K_0^{x_k} &= \sum_{n=1}^{x_k} \binom{x_k}{n} (-1)^n \frac{k_0^{r_0}}{P_1^{t_1}\, P_2^{t_2}} \frac{\left(P_3^{t_3}\, P_4^{t_4}\right)^{n-1}}{\left(\widetilde{P}_3^{t_3}\, \widetilde{P}_4^{t_4}\right)^{n}} \\
&= \sum_{n=1}^{x_k} \binom{x_k}{n} (-1)^n \frac{k_0^{r_0}}{P_1^{t_1}\, P_2^{t_2}} \frac{\left[\left(\widetilde{P}_3 - m_3^2\right)^{t_3} \left(\widetilde{P}_4 - m_4^2\right)^{t_4}\right]^{n-1}}{\left[\widetilde{P}_3^{t_3}\, \widetilde{P}_4^{t_4}\right]^{n}} \\
&= \sum_{n=1}^{x_k} \binom{x_k}{n} (-1)^n \sum_{n_3=0}^{t_3(n-1)} \sum_{n_4=0}^{t_4(n-1)} \binom{t_3(n-1)}{n_3} \binom{t_4(n-1)}{n_4} \\
&\quad (-m_3^2)^{n_3} (-m_4^2)^{n_4} \frac{k_0^{r_0}}{P_1^{t_1} P_2^{t_2} \widetilde{P}_3^{t_3+n_3} \widetilde{P}_4^{t_4+n_4}}.
\end{aligned} \tag{4.40}$$

Thus one obtains the second term (the subtraction term) as a sum of integrals of the form

$$\int d^Dl d^Dk \frac{k_0^{r_0}}{P_1^{t_1}(l+q) P_2^{t_2}(l) \widetilde{P}_3^{t_3}(l+k) \widetilde{P}_4^{t_4}(k)}. \tag{4.41}$$

This integral is a special case of the more general one

$$\widetilde{J}_1 \;=\; \int d^D l d^D k \frac{l_0^{p_0}\, k_0^{r_0}}{P_1^{t_1}(l+q) P_2^{t_2}(l) \widetilde{P}_3^{t_3}(l+k) \widetilde{P}_4^{t_4}(k)} \tag{4.42}$$

where $t_i \in \mathbb{N}$ and p_0 and r_0 are non-negative integers.

The degree of divergence of each term of the sum of the integrand Eq. (4.40) is

$$\begin{aligned} w' &= 2\sum_{i=1}^{4} t_i + 2(n_3+n_4) - 8 - r_0 \\ &= w + 2(n_3+n_4). \end{aligned} \tag{4.43}$$

To absorb the UV-divergences of J_1, only the divergent terms in Eq. (4.40) are needed. A minimal set of subtraction terms is determined by $n_3 + n_4 \leq -\frac{w}{2}$.

In the next section, we present the analytical solution of the integral $\widetilde{J}_1$.

4.2.4 The integral $\widetilde{J}_1$

This section is dedicated to the analytical solution of the integrals $\widetilde{J}_1$ which are subtraction terms of J_1 in Eq. (4.40). We first write the integral more explicitly

$$\widetilde{J}_1 \;=\; \int d^D l d^D k \frac{l_0^{p_0}\, k_0^{r_0}}{P_1^{t_1}(l+q)\; P_2^{t_2}(l)\; [(l+k)^2]^{t_3}\; [k^2]^{t_4}} \tag{4.44}$$

and rewrite the k-integral in a Lorentz-covariant form

$$= \int d^D l \frac{l_0^{p_0}}{P_1^{t_1}(l+q)\; P_2^{t_2}(l)} \frac{q_{\mu_1} \cdots q_{\mu_{r_0}}}{q_0^{r_0}} \underbrace{\int d^D k \frac{k^{\mu_1} \cdots k^{\mu_{r_0}}}{\left[(l+k)^2\right]^{t_3} \left[(k)^2\right]^{t_4}}}_{B_{t_3\, t_4}^{\mu_1 \cdots \mu_{r_0}}(l^2;0,0) \quad \text{(see Eq. (3.1))}} .$$

The analytical result of the k-integral on the right-hand side can be obtained directly by taking the derivative r_0-times with respect to l^{μ_i} of $B_{t_3\, t_4}^{(0)}(l^2;0,0)$ [36], where

$$\begin{aligned} B_{t_3\, t_4}^{(0)}(l^2;0,0) &= \int d^D k \frac{1}{\left[(l+k)^2\right]^{t_3} \left[k^2\right]^{t_4}} \\ &= \frac{i\,(-1)^{t_3+t_4} \pi^{D/2} \Gamma(t_3+t_4-D/2)\, \mathcal{B}(D/2-t_4, D/2-t_3)}{\Gamma(t_3)\Gamma(t_4)} (-l^2)^{D/2-t_3-t_4} \end{aligned} \tag{4.45}$$

and $\mathcal{B}$ is the beta function (c.f. Eq. (4.4)).

Performing the derivatives with respect to l^{μ_i} of $B^{(0)}_{t_3\,t_4}(l^2;0,0)$ to obtain $B^{\mu_1\cdots\mu_{r_0}}_{t_3\,t_4}(l^2;0,0)$ then contracting it with $\frac{q_{\mu_1}\cdots q_{\mu_{r_0}}}{q_0^{r_0}}$, one obtains

$$\int d^Dk\frac{k_0^{r_0}}{\left[(l+k)^2\right]^{t_3}\left[k^2\right]^{t_4}} = \frac{i\,(-1)^{D/2}(\pi)^{D/2}}{\Gamma(t_3)\Gamma(t_4)}\sum_{j=0}^{[r_0]/2}\frac{r_0!\,(-l_0)^{r_0-2j}\,(l^2)^{j+D/2-t_3-t_4}}{(r_0-2j)!\,j!\,2^{2j}} \times\Gamma(t_4+t_4-D/2-j)\,\mathcal{B}(D/2-t_4+r_0-j,D/2-t_3+j) \tag{4.46}$$

with $[r_0]=r_0$ if r_0 is even, and $[r_0]=r_0-1$ if r_0 is odd. Substituting the right-hand side of Eq. (4.46) into Eq. (4.44), one ends up with three-point one-loop integrals of the form

$$C^{p_0'\,0\,0}_{\alpha\,t_1\,t_2}=\mathfrak{C}\int d^Dl\frac{l_0^{p_0'}}{(l^2)^\alpha\,P_1^{t_1}(l+q)\,P_2^{t_2}(l)}\quad\text{(see Eq. (3.14))} \tag{4.47}$$

where α is an integer when $D=4$, and $\mathfrak{C}$ is a coefficient which depends on D,t_3,t_4 and r_0.

Because the l-integral is still UV-divergent, one can not set $D=4$, so α is in general a non-integer number. To calculate this kind of integral, an algebraic approach is proposed in [12] by using massive one-loop tadpole subtraction terms. However, again, the method can not be applied to solve the integral Eq.(4.47) if $t_2=0$ which is the case for the sunrise topology. In this thesis, we will use the hypergeometrical function to obtain an explicit expression for this integral when $t_2=0$. Using the same reduction procedure as in the one-loop case, the integral on the right-hand side of Eq.(4.47) (in the case $t_2=0$) can be reduced to the form

$$\mathfrak{L}=\int d^Dl\frac{(l^2)^n}{[(l+q)^2-m_1^2+i\rho]^t[l^2]^\alpha} \tag{4.48}$$

where α is a non-integer number. Now, using Feynman parametrisation as in Eq. (2.3)

and after shifting the momentum, one obtains

$$\begin{aligned}
\mathfrak{L} &= \frac{\Gamma(t+\alpha)}{\Gamma(t)\Gamma(\alpha)} \int_0^1 dx \int d^D l \, \frac{x^{t-1}(1-x)^{\alpha-1}\left[l^2 - 2x\, l\cdot q + q^2 x^2\right]^n}{\left[l^2 - q^2 x(z-(1-x)) + i\rho\right]^{t+\alpha}} \qquad (4.49)\\
&= \frac{\Gamma(t+\alpha)}{\Gamma(t)\Gamma(\alpha)} \int_0^1 dx \sum_{j_1=0}^{n} \sum_{j_2=0}^{j_1} \binom{n}{j_1} \binom{j_1}{j_2} x^{t-1}(1-x)^{\alpha-1}\\
&\times \int d^D l \, \frac{(l^2)^{n-j_1}(-2x\, l\cdot q)^{j_2}(q^2x^2)^{j_1-j_2}}{\left[l^2 - q^2 x(z-(1-x)) + i\rho\right]^{t+\alpha}}\\
&= \frac{\Gamma(t+\alpha)}{\Gamma(t)\Gamma(\alpha)} \int_0^1 dx \sum_{j_1=0}^{n} \sum_{j_2=0}^{j_1} \binom{n}{j_1} \binom{j_1}{j_2} (q^2)^{j_1-j_2/2}(-2)^{j_2} \frac{\Gamma(D/2)\Gamma(j_2/2+1/2)}{\Gamma(1/2)\Gamma(D/2+j_2/2)}\\
&\times \int d^D l \, \frac{x^{t+2j_1-j_2-1}(1-x)^{\alpha-1}(l^2)^{n-j_1+j_2/2}}{\left[l^2 - q^2 x(z-(1-x)) + i\rho\right]^{t+\alpha}}
\end{aligned}$$

where j_2 is even, $z = \frac{m^2}{q^2}$ and the relations Eqs. (2.5, 2.7) are used in the last line. Using the ansatz of the one-loop tadpole integral Eq. (2.7) one obtains

$$\mathfrak{L} = \sum_{j_1=0}^{n} \sum_{j_2=0}^{j_1} \mathfrak{C} \int_0^1 dx \; x^{a-1}(1-x)^{a'-1}\left[(z-i\rho)-x\right]^{-b_1} \qquad (4.50)$$

with

$$\begin{aligned}
a &= \alpha,\\
a' &= \left(j_1 + n - \alpha + \frac{D-j_2}{2}\right), \qquad (4.51)\\
b_1 &= -\left(n - j_1 - t - \alpha + \frac{D+j_2}{2}\right)
\end{aligned}$$

and

$$\begin{aligned}
\mathfrak{C} &= i \binom{n}{j_1}\binom{j_1}{j_2} (2)^{j_2}\, \pi^{D/2}\, (-1)^{n-j_1+\frac{3j_2}{2}-t-\alpha}\, (q^2)^{n-t-\alpha+D/2} \qquad (4.52)\\
&\times \frac{\Gamma(D/2+n-j_1+j_2/2)\Gamma(t-n+j_1+\alpha-\frac{j_2+D}{2})\Gamma(D/2)\Gamma(j_2/2+1/2)}{\Gamma(t)\Gamma(\alpha)\Gamma(D/2+j_2/2)}.
\end{aligned}$$

With the help of the formulae Eq. (B.4) and Eq. (B.5), the integral over x can be integrated in terms of hypergeometrical functions

$$\mathfrak{L} = \sum_{j_1=0}^{n} \sum_{j_2=0}^{j_1} \mathfrak{C} \ (z - i\rho)^{-b_1} \ \mathcal{B}(a, a') \ {}_2F_1 \left(a, b_1, a + a', \frac{1}{z - i\rho} \right). \tag{4.53}$$

For the scalar sunrise integral, this result is in agreement with [37]. For the general tensor case, an expansion method (at the pole $D = 4$) for the hypergeometrical functions ${}_2F_1$ is presented in [38]. From the computational point of view, the expression Eq. (4.53) is ready to be evaluated with the help of nested sums as introduced in [38].

4.2.5 The degenerate integral J_2 and the sunrise topology

In this section, we consider the integral J_2 introduced in Eq. (4.21) which appeared as a result of the tensor reduction in section 4.2.1

$$J_2 \ = \ \int d^D l \ d^D k \ \mathcal{J}_{2\,\{t_i\}}^{p_0,r_0} \qquad\qquad (i = 1,\, 2,\, 3)$$

with

$$\mathcal{J}_{2\,\{t_i\}}^{p_0,r_0} = \frac{l_0^{p_0} \, k_0^{r_0}}{P_1^{t_1}(l+q) \, P_2^{t_2}(l+k) \, P_3^{t_3}(k)} \tag{4.54}$$

and

$$\begin{aligned} P_1(l+q) &= (l+q)^2 - m_1^2 + i\rho, \\ P_2(l+k) &= (l+k)^2 - m_2^2 + i\rho, \\ P_3(k) &= k^2 - m_3^2 + i\rho. \end{aligned} \tag{4.55}$$

By power counting, one obtains

$$\begin{aligned} w_l &= 2(t_1 + t_2) - 4 - p_0, \\ w_k &= 2(t_2 + t_3) - 4 - r_0, \\ w &= w_l + w_k - 2t_2. \end{aligned} \tag{4.56}$$

If $w_l \geq 2t_2$ or $w_k \geq 2t_2$ one can use the same procedure as described in section 4.2.3. However, the procedure fails in the degenerate case where $w_l + w_k < 2t_2$. To see the problem, let us consider a simple case, where $0 < w_l < 2t_2$ and $w_k \leq 0$. The integral contains overall and k-sub-divergences. Now, if one uses the subtraction term K^{x_k} as presented in Eq. (4.35) with $w'_k = w_k + x_k = 1$ then the overall divergence of the integral after subtraction is

$$\begin{aligned} w' &= w_l + w'_k - 2t_2 \\ &= w_l + 1 - 2t_2 \\ &\leq 0. \end{aligned} \tag{4.57}$$

The integral is still overall divergent.

A naive approach to this problem is to increase x_k such that $x_k + w_k > 1$ and $w' = w_l + w_k + x_k - 2t_2 = 1$. However, this is dangerous because K^{x_k} is constructed by massless propagators and an IR-divergence will be introduced into the k-sub-integral if $x_k + w_k > 1$.

Another approach is to use the *forest formula* based on the BPHZ prescription [39] to find suitable subtraction terms . The use of the forest formula may open a more general way to implement a renormalization procedure into our method. Even though this is an interesting topic to work on in the future, in this thesis, an alternative approach using massive propagators to construct the subtraction terms is used to solve the degenerate integral J_2.

With a massive subtraction term, one can chose x_k as large as needed to remove the overall divergence without introducing spurious IR-divergences. On the other hand, one has to solve more complicated integrals.

We now suppose that $w_l \leq 0$, $w_k \leq 0$ and $w \leq 0$ [4]. In order to remove all divergences, one needs subtraction terms to remove the overall-, the l- and the k-sub-divergences. The trick is to use a massive subtraction term to increase the degree of l-sub-divergence to $2t_2$ and isolate the overall divergence in a scalar two-loop vacuum integral which can

[4] If $w_k > 0$ (or $w_l > 0$) we need only massive subtraction terms for the l- (or the k)-integral.

be integrated analytically. The k-sub-divergence is still subtracted off by the massless subtraction term as before. We define subtraction terms

$$\begin{aligned} L^{x_l} &:= \left(1-\frac{P_1^{t_1}}{\overline{P}_1^{\,t_1}}\right)^{x_l} &&= 1+L_0^{x_l}, \\ K^{x_k} &:= \left(1-\frac{P_2^{t_2}P_3 t_3}{\widetilde{P}_2^{t_2}\widetilde{P}_3^{t_3}}\right)^{x_k} &&= 1+K_0^{x_k} \end{aligned} \tag{4.58}$$

where

$$\begin{aligned} \overline{P}_1 &:= l^2-m_1^2+i\rho, \\ \widetilde{P}_2 &:= (l+k)^2, \\ \widetilde{P}_3 &:= k^2, \\ L_0^{x_l} &:= \sum_{n=1}^{x_l}\left(\begin{array}{c} x_l \\ n \end{array}\right)(-1)^n\left(\frac{P_1^{t_1}}{\overline{P}_1^{\,t_1}}\right)^n, \\ K_0^{x_k} &:= \sum_{n=1}^{x_k}\left(\begin{array}{c} x_k \\ n \end{array}\right)(-1)^n\left(\frac{P_2^{t_2}P_3^{t_3}}{\widetilde{P}_2^{t_2}\widetilde{P}_3^{t_3}}\right)^n. \end{aligned}$$

x_l and x_k are chosen such that

$$\begin{aligned} x_l+w_l &= 2t_2, \\ x_k+w_k &= 1. \end{aligned} \tag{4.59}$$

In contrast to K, the subtraction term L is constructed by a massive propagator $1/\overline{P}_1$ that allows one to choose x_l big enough to make the whole integral finite without introducing an IR-divergence. Using subtraction terms L and K, the integral can be rewritten as follows

$$\begin{aligned} J_2 &= \int d^4l\, d^4k\, \mathcal{J}_{2\,\{t_1\}}^{p_0,r_0}\, L^{x_l}\, K^{x_k} - \int d^Dl\, d^Dk\, \mathcal{J}_{2\,\{t_i\}}^{p_0,r_0}\, L_0^{x_l} \\ &\quad - \int d^Dl\, d^Dk\, \mathcal{J}_{2\,\{t_i\}}^{p_0,r_0}\, L_0^{x_l}\, K_0^{x_k} - \int d^Dl\, d^Dk\, \mathcal{J}_{2\,\{t_i\}}^{p_0,r_0}\, K_0^{x_k}. \end{aligned} \tag{4.60}$$

The first term on the right-hand side of Eq. (4.60) is finite; so it can be integrated numerically in $D=4$ dimensions. The last two terms are massless k-integrals with

integrands

$$\mathcal{J}^{p_0,r_0}_{2\,\{t_i\}}\; K_0^{x_k} \;=\; \sum_{n_1=1}^{x_k} \left(\begin{array}{c} x_k \\ n_1 \end{array} \right) (-1)^{n_1} \frac{\left(P_2^{t_2} P_3^{t_3}\right)^{n_1-1} l_0^{p_0}\; k_0^{r_0}}{P_1^{t_1} \left(\widetilde{P}_2^{t_2} \widetilde{P}_3^{t_3}\right)^{n_1}}, \tag{4.61}$$

$$\mathcal{J}^{p_0,r_0}_{2\,\{t_i\}}\; L_0^{x_l}\; K_0^{x_k} \;=\; \sum_{n_1=1}^{x_k} \sum_{n_2=1}^{x_l} \left(\begin{array}{c} x_k \\ n_1 \end{array} \right) \left(\begin{array}{c} x_l \\ n_2 \end{array} \right) (-1)^{n_1+n_2} \frac{\left(P_1^{t_1}\right)^{n_2-1} \left(P_2^{t_2} P_3^{t_3}\right)^{n_1-1} l_0^{p_0}\; k_0^{r_0}}{(\overline{P}_1^{\,t_1})^{n_2} \left(\widetilde{P}_2^{t_2} \widetilde{P}_3^{t_3}\right)^{n_1}}$$

which can be evaluated using $\widetilde{J}_1$ (see section 4.2.4). The second term of Eq. (4.60) is the tensor vacuum two-loop integral and has the form

$$\int d^D l\; d^D k \frac{l_0^{p_0}\; k_0^{r_0}}{\left[l^2 - m_1^2 + i\rho\right]^{t_1} \left[(l+k)^2 - m_2^2 + i\rho\right]^{t_2} \left[k^2 - m_3^2 + i\rho\right]^{t_3}} \tag{4.62}$$

$$= \frac{q_{\mu_1} \ldots q_{\mu_{p_0}} q_{\nu_1} \ldots q_{\nu_{r_0}}}{q^{r_0+p_0}} \int d^D l\; d^D k \frac{l^{\mu_1} \ldots l^{\mu_{p_0}} k^{\nu_1} \ldots k^{\nu_{r_0}}}{\left[l^2 - m_1^2 + i\rho\right]^{t_1} \left[(l+k)^2 - m_2^2 + i\rho\right]^{t_2} \left[k^2 - m_3^2 + i\rho\right]^{t_3}}.$$

By using the decomposition in Lorentz covariant terms like in Eqs. (2.23, 2.24) and then contracting with the external momenta q_{μ_i}, the above integral is decomposed into massive vacuum integrals of the form

$$\int d^D l\; d^D k \frac{(l^2)^n (l\cdot k)^m (k^2)^p}{\left[l^2 - m_1^2 + i\rho\right]^{t_1} \left[(l+k)^2 - m_2^2 + i\rho\right]^{t_2} \left[k^2 - m_3^2 + i\rho\right]^{t_3}} \tag{4.63}$$

where n, m, p are non-negative integer numbers. By inserting

$$\begin{aligned} l^2 &= \left[l^2 - m_1^2 + i\rho\right] + m_1^2 - i\rho, \\ 2\; l\cdot k &= \left[(l+k)^2 - m_2^2 + i\rho\right] - l^2 - k^2 + m_2^2 - i\rho, \\ k^2 &= \left[k^2 - m_3^2 + i\rho\right] + m_3^2 - i\rho \end{aligned} \tag{4.64}$$

into the numerator and then canceling the terms in the denominator of Eq. (4.63), the integral is reduced to products of one-loop integrals and scalar massive vacuum integrals of the form

$$S_{\{t_1,t_2,t_3\}} \;=\; \int d^D l\; d^D k \frac{1}{\left[l^2 - m_1^2 + i\rho\right]^{t_1} \left[(l+k)^2 - m_2^2 + i\rho\right]^{t_2} \left[k^2 - m_3^2 + i\rho\right]^{t_3}}. \tag{4.65}$$

An efficient recursion relation to reduce Eq. (4.62) to Eq. (4.65) and the analytical solution of the massive vacuum integrals $S_{\{t_1,t_2,t_3\}}$ is presented in [40, 41].

In the next section, we present another approach to solve the massive vacuum integral, which is more flexible from the computational point of view.

4.2.6 The massive vacuum integral $S_{\{t_1,t_2,t_3\}}$

To calculate the massive vacuum integral $S_{\{t_1,t_2,t_3\}}$, it is sufficient to consider the special case $t_1 = t_2 = t_3 = 1$. If $t_1 > 1$ (or t_2 or $t_3 > 1$) then the integral is non-degenerate and can be evaluated as described in section 4.2.3. Let us consider the integral

$$S_{\{1,1,1\}} = \int d^D l \; d^D k \frac{1}{P_1 \, P_2 \, P_3}$$

with

$$\begin{aligned} P_1 &= l^2 - m_1^2 + i\rho, \\ P_2 &= (l+k)^2 - m_2^2 + i\rho, \\ P_3 &= k^2 - m_3^2 + i\rho. \end{aligned} \tag{4.66}$$

We apply the following transformation

$$\begin{aligned} S_{\{1,1,1\}} &= \int d^D l \; d^D k \left\{ \frac{1}{P_1 \, P_2 \, P_3} - \frac{1}{\widetilde{P}_1 \, P_2 \, P_3} \right\} + \int d^D l \; d^D k \; \left\{ \frac{1}{\widetilde{P}_1 \, P_2 \, P_3} - \frac{1}{\widetilde{P}_1 \, \widetilde{P}_2 \, P_3} \right\} \\ &\quad + \int d^D l \; d^D k \; \frac{1}{\widetilde{P}_1 \, \widetilde{P}_2 \, P_3} \qquad (4.67) \\ &= \int d^D l \; d^D k \; \frac{m_1^2}{P_1 \, \widetilde{P}_1 \, P_2 \, P_3} + \int d^D l \; d^D k \; \frac{m_2^2}{\widetilde{P}_1 \, P_2 \, \widetilde{P}_2 \, P_3} \\ &\quad + \int d^D l \; d^D k \; \frac{1}{\widetilde{P}_1 \, \widetilde{P}_2 \, P_3} \end{aligned}$$

with

$$\widetilde{P}_1 = l^2, \tag{4.68}$$

$$\widetilde{P}_2 = (l+k)^2. \tag{4.69}$$

The first two terms of Eq. (4.67) are actually integrals of the type J_1 which are discussed in section 4.2.3. The last term which contains the overall divergence is now isolated in a simple integral $\widetilde{J}_1$ with $t_1 = 0$ (see section 4.2.4) which can be integrated analytically.

Having discussed the analytical solution of the divergent terms, in order to complete the calculation, one needs to find numerical solutions of the integrals which appear in the finite terms of Eq. (4.39) and Eq. (4.60)

4.2.7 The numerical integration of finite integrals

As discussed in the previous sections, the finite parts of $T^{(f)}, T^{(g)}, T^{(h)}$ can be obtained via the three finite integrals

$$J_0|_{D=4} \quad , \qquad \int d^4l \int d^4k \; \mathcal{J}^{r_0}_{1\,\{t_i\}} \; K^{x_k} \quad \text{and} \qquad \int d^4l \int d^4k \; \mathcal{J}^{p_0\,r_0}_{2\,\{t_i\}} \; L^{x_l} \; K^{x_k}. \tag{4.70}$$

These are integrals of five real variables $l_0, k_0, l_\perp, k_\perp$ and z (see Eq. (2.15)). In contrast to the one-loop problem no analytical solution is known yet for two-loop integrals with arbitrary masses so that we will integrate these three integrals numerically. In fact, the integrals over the last three variables $l_\perp, k_\perp$ and z can be integrated analytically so that the finite parts of the two-point two-loop integrals can be obtained in terms of two-dimensional integrals. Let us rewrite

$$J_0|_{D=4} \;\; = \;\; \int\limits_{-\infty}^{\infty} dl_0 \; dk_0 \; \widehat{J}_0 \tag{4.71}$$

with

$$\widehat{J}_0 = \int\limits_0^\infty dl_\perp \; dk_\perp \int\limits_{-1}^{1} dz \; \frac{l_\perp^2 \; k_\perp^2}{P_1^{t_1}(l+q) \; P_2^{t_2}(l) \; P_3^{t_3}(l+k) \; P_4^{t_4}(k-q) \; P_5^{t_5}(k)}. \tag{4.72}$$

The second finite integral can also be rewritten as a two-dimensional integral over l_0 and k_0

$$\int d^4l \int d^4k \; \mathcal{J}^{r_0}_{1\,\{t_i\}} \; K^{x_k} \;\; = \;\; \sum_{n=0}^{x_k} \binom{x_k}{n} (-1)^n \int\limits_{-\infty}^{\infty} dl_0 \; dk_0 \; k_0^{r_0} \; \mathfrak{J}_1^n \tag{4.73}$$

with

$$\mathfrak{J}_1^n = 8\pi^2 \int\limits_0^\infty dl_\perp \; dk_\perp \int\limits_{-1}^{1} dz \; l_\perp^2 \; k_\perp^2 \;\; \frac{1}{P_1^{t_1} \; P_2^{t_2}} \frac{\left(P_3^{t_3} \; P_4^{t_4}\right)^{n-1}}{\left(\widetilde{P}_3^{t_3}(l+k) \; \widetilde{P}_4^{t_4}\right)^n}. \tag{4.74}$$

The third finite integral can be rewritten as follows

$$\int d^4l \int d^4k \; \mathcal{J}^{p_0\,r_0}_{2\,\{t_i\}} \; L^{x_l} \; K^{x_k} \;\; = \;\; \sum_{n_1=0}^{x_k} \sum_{n_2=0}^{x_l} \binom{x_k}{n_1} \binom{x_l}{n_2} \; (-1)^{n_1+n_2} \int\limits_{-\infty}^{\infty} dl_0 \; dk_0 \; l_0^{p_0} \; k_0^{r_0} \; \mathfrak{J}_2^{n_1,n_2}$$

with

$$\mathfrak{J}_2^{n_1,n_2} = 8\pi^2 \int_0^\infty dl_\perp\, dk_\perp \int_{-1}^{1} dz\; l_\perp^2\; k_\perp^2\; \frac{\left(P_1^{t_1}\right)^{n_2-1}\left(P_2^{t_2}P_3^{t_3}\right)^{n_1-1}}{P_1^{t_1}\; P_2^{t_2}\; P_3^{t_3}\; (\overline{P}_1^{t_1})^{n_2}\left(\widetilde{P}_2^{t_2}\widetilde{P}_3^{t_3}\right)^{n_1}}. \tag{4.75}$$

The integrals $\widehat{J}_0$, $\widehat{\mathfrak{J}}_1$ and $\widehat{\mathfrak{J}}_2$ are well defined. Further, $\widehat{\mathfrak{J}}_1$ and $\widehat{\mathfrak{J}}_2$ can be decomposed into integrals of the form

$$\widehat{J}_1 = 8\pi^2 \int_0^\infty dl_\perp dk_\perp \int_{-1}^{1} dz\; \frac{l_\perp^2\; k_\perp^2}{P_1^{t_1}(l+q)\; P_2^{t_2}(l)\; P_3^{t_3}(l+k)\; P_4^{t_4}(k)} \tag{4.76}$$

and

$$\widehat{J}_2 \;=\; 8\pi^2 \int_0^\infty dl_\perp dk_\perp \int_{-1}^{1} dz\; \frac{l_\perp^2\; k_\perp^2}{P_1^{t_1}(l+q)\; P_2^{t_2}(l+k)\; P_3^{t_3}(k)} \tag{4.77}$$

which are also well defined. In the case of J_0, the method of integration over $l_\perp$, $k_\perp$ and z using the residuum theorem is discussed in detail in [11]. In contrast to J_0, the integrands of $\widehat{J}_1$ and $\widehat{J}_2$ do not decrease rapidly enough for $l_\perp \to \infty$ and for $k_\perp \to \infty$ so that they can only be evaluated as improper integrals. In the following, we present the solutions of these three integrals, while their detailed calculation is given in appendix C.

- The solution of the integral $\widehat{J}_0$

$$\begin{aligned}\widehat{J}_0 \;&=\; 8\pi^2 \int_0^\infty dl_\perp dk_\perp \int_{-1}^{1} dz\; \frac{l_\perp^2\; k_\perp^2}{P_1^{t_1}(l+q)\; P_2^{t_2}(l)\; P_3^{t_3}(l+k)\; P_4^{t_4}(k-q)\; P_5^{t_5}(k)} \qquad (4.78)\\ &=\; \prod_{i=1}^{5} \frac{1}{(t_i-1)!}\frac{d^{t_i-1}}{d(m_i^2)^{t_i-1}}\; \widehat{J}_{0\,\{1,1,1,1,1\}}\end{aligned}$$

with

$$\begin{aligned}\widehat{J}_{0\,\{1,1,1,1,1\}} \;&=\; 8\pi^2 \int_0^\infty dl_\perp dk_\perp \int_{-1}^{1} dz\; \frac{l_\perp^2\; k_\perp^2}{P_1(l+q)\; P_2(l)\; P_3(l+k)\; P_4(k-q)\; P_5(k)}\\ &=\; \frac{4\pi^4}{(w_1^2-w_2^2)(w_4^2-w_5^2)}\left\{\mathcal{L}_{134}+\mathcal{L}_{235}-\mathcal{L}_{234}-\mathcal{L}_{135}\right\} \qquad (4.79)\end{aligned}$$

where

$$\begin{aligned} w_1 &= \sqrt{(l_0+q_0)^2-m_1^2+i\rho} \;\; ; \;\; w_2 = \sqrt{l_0^2-m_2^2+i\rho} \;\; ; \\ w_3 &= \sqrt{(l_0+k_0)^2-m_3^2+i\rho} \;\; ; \;\; w_4 = \sqrt{(k_0-q_0)^2-m_4^2+i\rho} \;\; ; \\ w_5 &= \sqrt{k_0^2-m_5^2+i\rho} \end{aligned} \tag{4.80}$$

and

$$\begin{aligned} \mathcal{L}_{i\,j\,k} &= \ln(w_i+w_j+w_k), \\ \vec{q} &= (q_0,\vec{0}). \end{aligned} \tag{4.81}$$

- The solution of the integral $\widehat{J}_1$

$$\begin{aligned} \widehat{J}_1 &= 8\pi^2 \int\limits_0^\infty dl_\perp dk_\perp \int\limits_{-1}^{1} dz \; \frac{l_\perp^2 \; k_\perp^2}{P_1^{t_1}(l+q) \; P_2^{t_2}(l) \; P_3^{t_3}(l+k) \; P_4^{t_4}(k)} \\ &= \prod_{i=1}^{4} \frac{1}{(t_i-1)!} \frac{d^{t_i-1}}{d(m_i^2)^{t_i-1}} \; \widehat{J}_{1\;\{1,1,1,1\}} \end{aligned} \tag{4.82}$$

with

$$\begin{aligned} \widehat{J}_{1\;\{1,1,1,1\}} &= 8\pi^2 \int\limits_0^\infty dl_\perp dk_\perp \int\limits_{-1}^{1} dz \; \frac{l_\perp^2 \; k_\perp^2}{P_1(l+q) \; P_2(l) \; P_3(l+k) \; P_4(k)} \\ &= \frac{4\pi^4}{(w_1^2-w_2^2)} \left\{ \mathcal{L}_{234} - \mathcal{L}_{134} \right\} \end{aligned} \tag{4.83}$$

and

$$\begin{aligned} w_1 &= \sqrt{(l_0+q_0)^2-m_1^2+i\rho} \;\; ; \;\; w_2 = \sqrt{l_0^2-m_2^2+i\rho} \;\; ; \\ w_3 &= \sqrt{(l_0+k_0)^2-m_3^2+i\rho} \;\; ; \;\; w_4 = \sqrt{k_0^2-m_4^2+i\rho} \;\; . \end{aligned} \tag{4.84}$$

- The solution of the integral $\widehat{J}_2$

$$\begin{aligned}\widehat{J}_2 &= 8\pi^2 \int_0^\infty dl_\perp dk_\perp \int_{-1}^{1} dz \, \frac{l_\perp^2 \, k_\perp^2}{P_1^{t_1}(l+q) \, P_2^{t_2}(l+k) \, P_3^{t_3}(k)} \\ &= \prod_{i=1}^{3} \frac{1}{(t_i-1)!} \frac{d^{t_i-1}}{d(m_i^2)^{t_i-1}} \, \widehat{J}_{2\,\{1,1,1\}}\end{aligned} \tag{4.85}$$

with

$$\begin{aligned}\widehat{J}_{2\,\{1,1,1\}} &= 8\pi^2 \int_0^\infty dl_\perp dk_\perp \int_{-1}^{1} dz \, \frac{l_\perp^2 \, k_\perp^2}{P_1(l+q) \, P_2(l+k) \, P_3(k)} \\ &= -4\pi^4 \, \ln(w_1+w_2+w_3)\end{aligned} \tag{4.86}$$

and

$$\begin{aligned} w_1 &= \sqrt{(l_0+q_0)^2 - m_1^2 + i\rho} \;\; ; \;\; w_2 = \sqrt{(l_0+k_0)^2 - m_2^2 + i\rho} \;\; ; \\ w_3 &= \sqrt{k_0^2 - m_3^2 + i\rho}. \end{aligned} \tag{4.87}$$

4.3 Tests and comparisons of two-loop two-point integrals

The algorithm to calculate two-loop self-energy integrals is implemented in the program package XLOOPS-GiNaC (see chapter 5). In this section, we present results of the calculation of two-loop self-energy integrals using XLOOPS-GiNaC and compare them with the existing results [30, 37, 42, 43].

The result of XLOOPS-GiNaC consists of two parts. One part is evaluated analytically in terms of the one-loop and/or hypergeometrical functions. The other one is evaluated numerically by using Monte Carlo or Quasi-Monte Carlo methods with the help of the routines Vegas [44] or ParInt [45].

In the following, the sum of the analytical value and of the average of the result of the numerical integration will be called the *main value*. The error of Vegas (or ParInt) will be called the *error*.

p^2	A	B	C	D	E	F
2.66667	−8.454752	−8.45044	−8.445577	−8.45038	−8.450650	± 5.1525E-4
1.77778	−8.286016	−8.28753	−8.286116	−8.28747	−8.287799	± 5.3794E-4
1.18519	−8.186514	−8.18486	−8.184417	−8.18481	−8.184650	± 6.2748E-4
0.790124	−8.116017	−8.11883	−8.118665	−8.11878	−8.118499	± 6.7367E-4
0.526749	−8.077571	−8.07583	−8.075740	−8.07577	−8.075849	± 6.6106E-4
0.351166	−8.053491	−8.04759	−8.047528	−8.04754	−8.047599	± 6.8391E-4
0.234111	−8.033367	−8.02896	−8.028900	−8.02890	−8.030249	± 6.9644E-4
					−8.029050	± 1.7597E-4
0.156074	−8.013727	−8.01662	−8.016561	−8.01656	−8.017200	± 7.4516E-4
0.104049	−8.018667	−8.00843	−8.008371	−8.00837	−8.008700	± 7.4386E-4
0.069366	−7.999787	−8.00298	−8.002926	−8.00292	−8.002699	± 7.5543E-4
0.046244	−7.997535	−7.99936	−7.999303	−7.99930	−7.999599	± 7.6981E-4
0.030829	−7.985865	−7.99695	−7.996891	−7.99689	−7.996850	± 7.8173E-4
0.020553	−8.006871	−7.99534	−7.995285	−7.99528	−7.994649	± 8.2185E-4
0.013702	−7.900682	−7.99427	−7.994214	−7.99421	−7.993350	± 8.5223E-4
0.009135	−7.957410	−7.99356	−7.993501	−7.99350	−7.991799	± 8.7328E-4
0.006090	−7.793835	−7.99308	−7.993026	−7.99302	−7.991899	± 9.0757E-4

Table 4.1: Real part of $T_{1234N}(p^2; m_1^2, m_2^2, m_3^2, m_4^2)$ for small values of p^2. The masses are $m_1 = 1$, $m_2 = 3$, $m_3 = 5$ and $m_4 = 7$ (GeV). Results in columns A-D are reproduced from [37].

We first consider the finite integral

$$T_{1234N}(p^2; m_1^2, m_2^2, m_3^2, m_4^2) := T_{1234}(p^2; m_1^2, m_2^2, m_3^2, m_4^2) - T_{1234}(p^2; m_1^2, m_2^2, 0, 0) \quad (4.88)$$

with

$$T_{1234}(p^2; m_1^2, m_2^2, m_3^2, m_4^2) = \int d^D l \, d^D k \, \frac{1}{[(l+p)^2 - m_1^2 + i\rho][l^2 - m_2^2 + i\rho][(l+k)^2 - m_3^2 + i\rho][k^2 - m_4^2 + i\rho]}. \quad (4.89)$$

In table 4.1, the four columns A, B, C, D are from [37]. In column A, the values are

from a two-dimensional numerical integral representation [46]. Column B shows values obtained by using a series expansion of hypergeometrical functions [37]. The results in column C are from an approximation using three terms of a Taylor expansion [47]. Finally, the results in column D are obtained from a one-dimensional integral [37].

We add results obtained by XLOOPS-GiNaC for the real part of T_{1234N} in column E together with errors in column F.[5] Here, we use $10^5 \times 10$ iterations of Vegas. The calculation takes less than 1 minute for each integration on a computer with 1.6 GHz Athlon CPU and 512 MB of RAM.

In Fig. (4.3), T_{1234N} is plotted as a function of the external momentum p^2. Because our results agree very well with the others, we present another plot in Fig. (4.3) to compare the differences. In order to do that, we will compare our results with the ones in column D which were evaluated by a one-dimensional integral. In this figure, we plot the relative errors (error normalized by main value) of XLOOPS-GiNaC and the differences between the main values of XLOOPS-GiNaC and the results in the column D of [37] normalized by the main value of XLOOPS-GiNaC. As shown in Fig. (4.3), the relative differences are at order 10^{-4}. We also observed that there is instability at the point $p^2 = 0.234111$. The agreement can be improved by increasing the number of sample points or iterations of Vegas as presented in the second entry of the table at $p^2 = 0.234111$. We do not see any special problem at this value of p^2, so we believe that the occurrence of the large deviation here is purely accidental

As a representative of the sunrise integrals, we calculate the finite integral T_{123N} and compare our results with the results of [42]. The finite integral T_{123N} is defined by

$$\begin{aligned} T_{123N}(p^2; m_1^2, m_2^2, m_3^2) &:= T_{123}(p^2; m_1^2, m_2^2, m_3^2) - T_{123}(p^2; m_1^2, 0, m_3^2) \\ &\quad - T_{123}(p^2; 0, m_2^2, m_3^2) + T_{123}(p^2; 0, 0, m_3^2) \end{aligned} \tag{4.90}$$

[5] The C/C++ program of this calculation can be found in the file `T1234N.cpp` of the XLOOPS-GiNaC package.

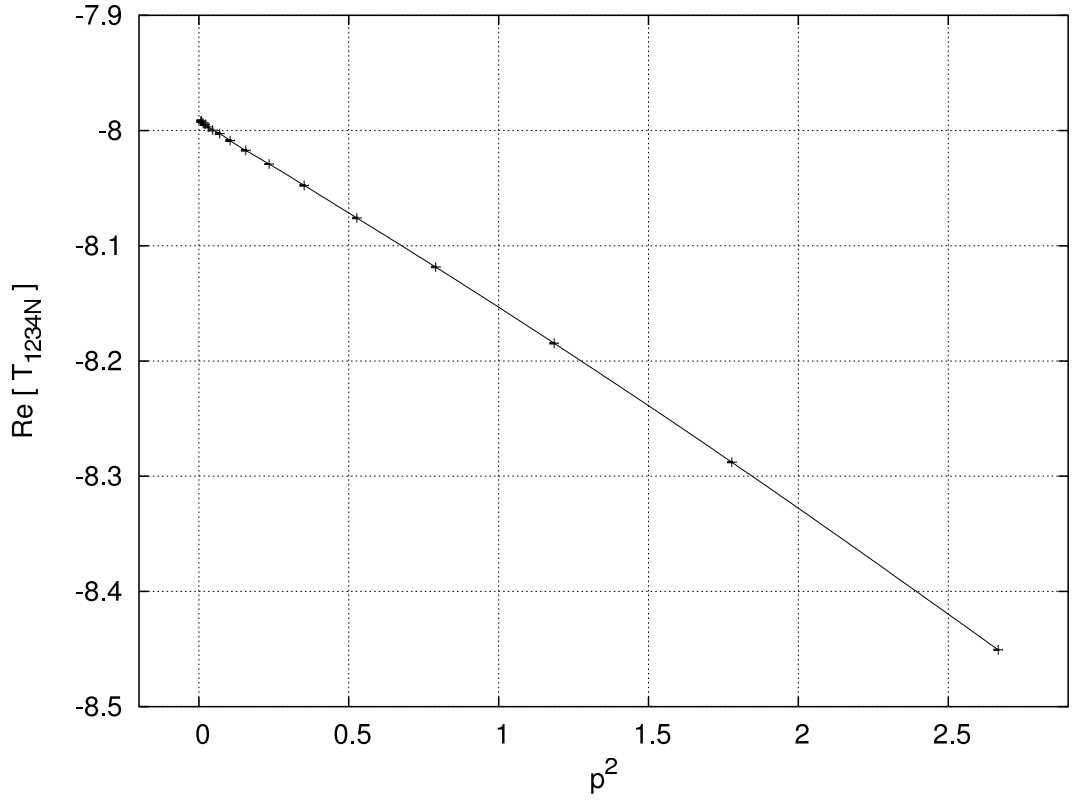


Figure 4.3: The real part of $T_{1234N}(p^2; m_1^2, m_2^2, m_3^2, m_4^2)$ as a function of p^2. The masses are $m_1 = 1, m_2 = 3, m_3 = 5$ and $m_4 = 7$ (GeV).

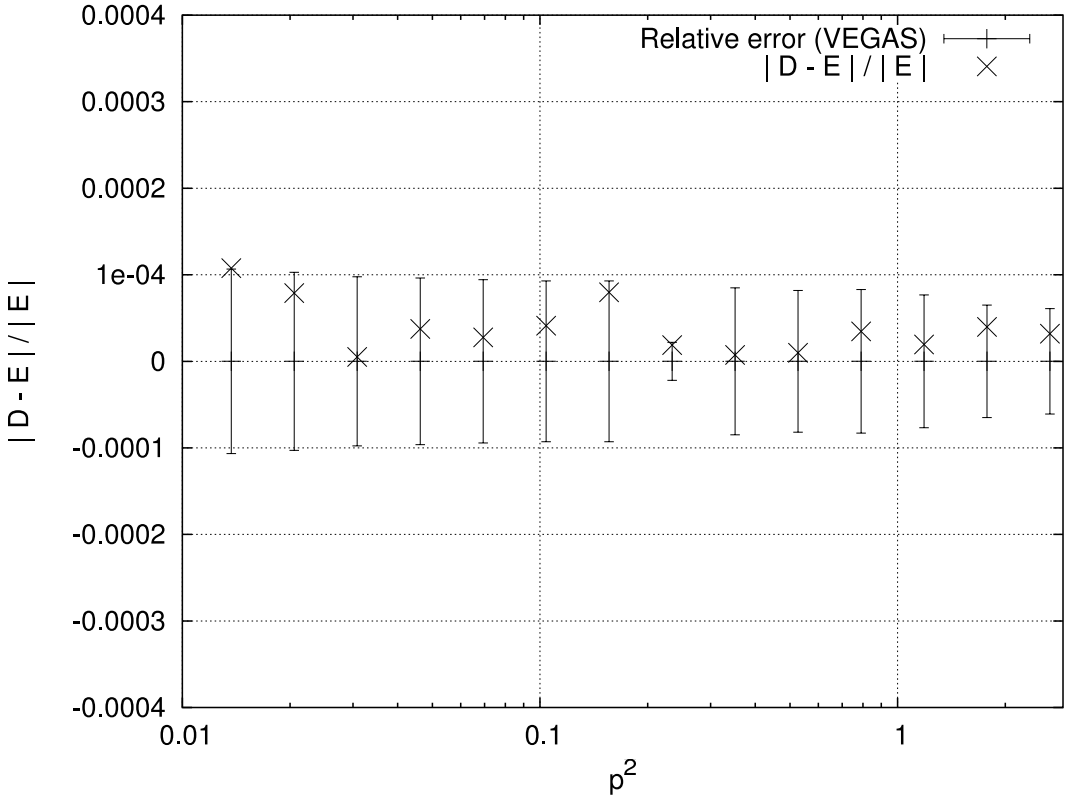


Figure 4.4: The error of XLOOPS-GiNaC and the relative difference between our results and of one-dimensional integral representation of [37], $\left|\frac{D-E}{E}\right|$, of T_{1234N} from Table. 4.1.

with

$$T_{123}(p^2; m_1^2, m_2^2, m_3^2) = \int d^D l \, d^D k \, \frac{1}{[(l+p)^2 - m_1^2 + i\rho][(l+k)^2 - m_2^2 + i\rho][k^2 - m_3^2 + i\rho]}. \quad (4.91)$$

In table 4.2 and table 4.3 we show the results of XLOOPS-GiNaC and the results of [42] for small and large p^2 with different values for the masses.[6] For the p^2-values above the threshold (large p^2), we also have a non-zero imaginary part, which is shown in the last two columns of table 4.3. Each result in the table is calculated in less than one minute by XLOOPS-GiNaC on a computer with Athlon 1.6GHz CPU and 512 MB RAM. As seen in the tables, the differences are of order 10^{-3}. The accuracy can be improved by increasing the number of iterations of Vegas and so the time of calculation also increases.

For a complete calculation of a sunrise integral, we compare the result of XLOOPS-GiNaC with the result of [30, 43].[7] We will follow the conventions of [43] where the sunrise integral is defined by

$$F_0(D, m_1^2, m_2^2, m_3^2, p^2) = \frac{\mu^{8-2D}}{((2\pi)^{D-2})^2} \int d^D k_1 \int d^D k_2 \, \frac{1}{(k_1^2 + m_1^2)(k_2^2 + m_2^2)((p - k_1 - k_2)^2 + m_3^2)} \quad (4.92)$$

where

$$\mu = m_1 + m_2 + m_3. \quad (4.93)$$

The integral $F_j(D, m_1^2, m_2^2, m_3^2, p^2)$ can be parametrized by

$$F_0(D, m_1^2, m_2^2, m_3^2, p^2) = C^2(D) \left\{ \frac{1}{(D-4)^2} \, F_0^{(-2)} + \frac{1}{(D-4)} F_0^{(-1)} + F_0^{(0)} \right\} \quad (4.94)$$

with

$$C(D) = (2\sqrt{\pi})^{4-D} \Gamma(3 - D/2). \quad (4.95)$$

[6] The C/C++ program of this calculation can be found in the files `T123N_below.cpp` and `T123N_above.cpp` of the XLOOPS-GiNaC package.

[7] The C/C++ program of this calculation can be found in the file `groote_remidi.cpp` of the XLOOPS-GiNaC package.

p^2	m_1	m_2	m_3	Re T_{123N}	Error
9	3	3	10	−7.31298	
				−7.31423	± 0.00113
20	2	3	10	−4.14938	
				−4.10781	± 0.00044
30	1	2	10	−0.81176	
				−0.81148	± 0.00034
49	1	1	10	−0.31675	
				−0.32018	± 0.00005
50	3	4	15	−7.94710	
				−7.95379	± 0.00080
25	2	2	10	−2.33538	
				−2.33527	± 0.00025
100	3	4	20	−6.01715	
				−6.02022	± 0.00167
150	3	4	20	−6.39036	
				−6.38877	± 0.00017
150	5	5	25	−14.5339	
				−14.5780	± 0.00326
200	5	5	25	−15.0523	
				−15.0542	± 0.00192

Table 4.2: Comparison for small p^2 of the real part of the subtracted sunrise integral T_{123N}. In each box, in the first entry we show the value of Table 1 of [42] (small p^2 series expansion). The second entry is our result with its numerical error in the last column.

p^2	m_1	m_2	m_3	Re T_{123N}	Vegas Error	Im T_{123N}	Vegas Error
80	2	3	2	0.58743		−11.2628	
				0.58585	± 0.00048	−11.2632	± 0.0002
100	3	3	3	−1.28285		−20.8996	
				−1.28227	± 0.00056	−20.9000	± 0.0005
100	2	3	4	−0.32864		−11.8459	
				−0.32028	± 0.00118	−11.8460	± 0.0002
150	3	4	4	−1.26795		−26.4912	
				−1.21248	± 0.00140	−26.4912	± 0.0005
150	2	4	3	1.56625		−11.9689	
				1.55999	± 0.00079	−11.9690	± 0.0003
150	3	3	4	0.98658		−16.1021	
				0.99128	± 0.00614	−16.1026	± 0.0004
150	3	4	5	−4.76259		−29.6012	
				−4.76273	± 0.00075	−29.6012	± 0.0006
200	2	3	4	1.69608		−6.02417	
				1.69549	± 0.00022	−6.02412	± 0.0001
200	3	4	4	1.86356		−20.3985	
				1.85942	± 0.00175	−20.3987	± 0.0004
250	4	4	4	2.64395		−27.6090	
				2.65248	± 0.00893	−27.6098	± 0.0006

Table 4.3: Comparison for large p^2 of the real and imaginary part of the subtracted sunrise integral T_{123N}. In each box, in the first entry we show the value of Table 2 of [42] (large p^2 series expansion). The second entry is our result with its numerical error in the next column.

For the divergent terms we obtain

$$F_0^{(-2)} = -\frac{1}{8}\sum_{i=1}^{3} m_i^2, \tag{4.96}$$

$$F_0^{(-1)} = \frac{1}{8}\left\{-\frac{p^2}{4} + \frac{3}{2}\sum_{i=1}^{3} m_i^2 - \sum_{i=1}^{3} m_i^2 \ln\left(\frac{m_i^2}{\mu^2}\right)\right\} \tag{4.97}$$

which is in agreement with the results of [30, 43]. Note that our result differs with the one of [43] by a minus sign in the term $\frac{p^2}{4}$ of $F_0^{(-1)}$. This is due to the fact that in [43] the integral is evaluated in Euclidian space.

In table 4.4 we present the numerical values of $\frac{F_0^{(0)}}{\mu^2}$ as a function of $\frac{p^2}{\mu^2}$. The abbreviations which are used in the table are

$$\begin{aligned} p_{ps1,r}^2 &= \frac{(m_1 - m_2 - m_3)^2}{\mu^2}, \\ p_{ps2,r}^2 &= \frac{(m_1 - m_2 + m_3)^2}{\mu^2}, \\ p_{ps3,r}^2 &= \frac{(m_1 + m_2 - m_3)^2}{\mu^2}. \end{aligned} \tag{4.98}$$

In column A we show the results of [43] using differential equation techniques. The results of XLOOPS-GiNaC and their errors are shown in columns B and C. The difference of the results is of order 10^{-3} where we use $10^5 \times 10$ iterations per integration for Vegas.

4.4 Summary for the two-loop self-energy integrals

We have shown that by using an appropriate set of subtraction terms, all the two-loop two-point integrals with arbitrary tensor degree and masses can be expanded in terms of one-loop functions and a finite set of two-fold integrals which are suited for numerical evaluation.

By applying the above method for calculating two-loop self-energy integrals, we get good agreement with results of other authors. The $\mathcal{O}(\epsilon^{-2})$ and $\mathcal{O}(\epsilon^{-1})$ terms of the integrals J_1 and J_2 agree analytically with the results of [29, 30, 43, 48, 49] for the arbitrary mass case

p_r^2	A	B	C
-1	-0.279454902855371	-0.279819	$\pm$6.6E-4
-0.99	$-0.2798928396415(5)$	-0.280351	$\pm$0.9E-4
$p_{ps3,r}^2$	-0.280281633048667	-0.280442	$\pm$4.4E-4
-0.9	$-0.2836780811878(5)$	-0.283169	$\pm$5.6E-4
$p_{ps2,r}^2$	-0.286233415451605	-0.285265	$\pm$16.3E-4
-0.825	$-0.2866587055221(5)$	-0.281268	$\pm$4.2E-4
$p_{ps1,r}^2$	-0.286906928933491	-0.286141	$\pm$5.7E-4
-0.8	$-0.2876221116285(5)$	-0.288281	$\pm$5.4E-4
-0.1	$-0.3101507246241(3)$	-0.310516	$\pm$1.9E-4
0	-0.312816604092084	-0.312800	$\pm$0.1E-4

Table 4.4: Comparison for the sunrise integral. The masses (in GeV) are $m_1 = 1$, $m_2 = 9$, $m_3 = 200$ and $\mu = m_1 + m_2 + m_3$. Results in columns A are reproduced from [43]. The results of XLOOPS-GiNaC and their errors are shown in column B and C, respectively.

as well as at the threshold and peseudo-thresholds. The finite terms of J_1 and J_2 ($\mathcal{O}(\epsilon^0)$) are compared numerically and agree very well with [30, 37, 42, 43] using a reasonable amount of computer time.

In the next section, the implementation of the procedure in the computer program XLOOPS-GiNaC is discussed in detail.

Chapter 5

The Program XLOOPS-GiNaC

To have a computer program that can calculate thousands of Feynman loop diagrams for a given physical process is a wish of high energy physicists on the way to understand higher order radiative corrections.

This chapter is an introduction to the computer program XLOOPS-GiNaC. The program is a computational implementation of the algorithms for the one- and two-loop integrations which are presented in the previous chapters. This chapter can also be used as a user manual of the XLOOPS-GiNaC program package. Readers are expected to have a basic knowledge of the C/C^{++} programming language. In addition, readers are strongly recommended to read the GiNaC manual [51], in particular to have an understanding of the concept of GiNaC's functions which we use extensively in XLOOPS-GiNaC.

The aim of the project XLOOPS is to provide a computer program for calculating one-particle irreducible (1PI) Feynman diagrams with one or two closed loops for arbitrary processes in the Standard Model and related theories of particle physics. All necessary integrals are evaluated with arbitrary tensor degree, masses and momenta. XLOOPS was originally written in the language of the Maple computer algebra system [3].

The XLOOPS-GiNaC package is a new implementation of the one-loop and two-loop integrals based on GiNaC, a C/C^{++} library for symbolic computation. Currently it consists

of a library of GiNaC functions for one-loop one-, two- and three-point integrals and a library of C/C++ functions for two-loop two-point integrals of any tensor rank.

At the time being, the structure of the XLOOPS-GiNaC package is shown in Fig. 5.1. Written in the standard C/C++ programming language, based on the GiNaC library,

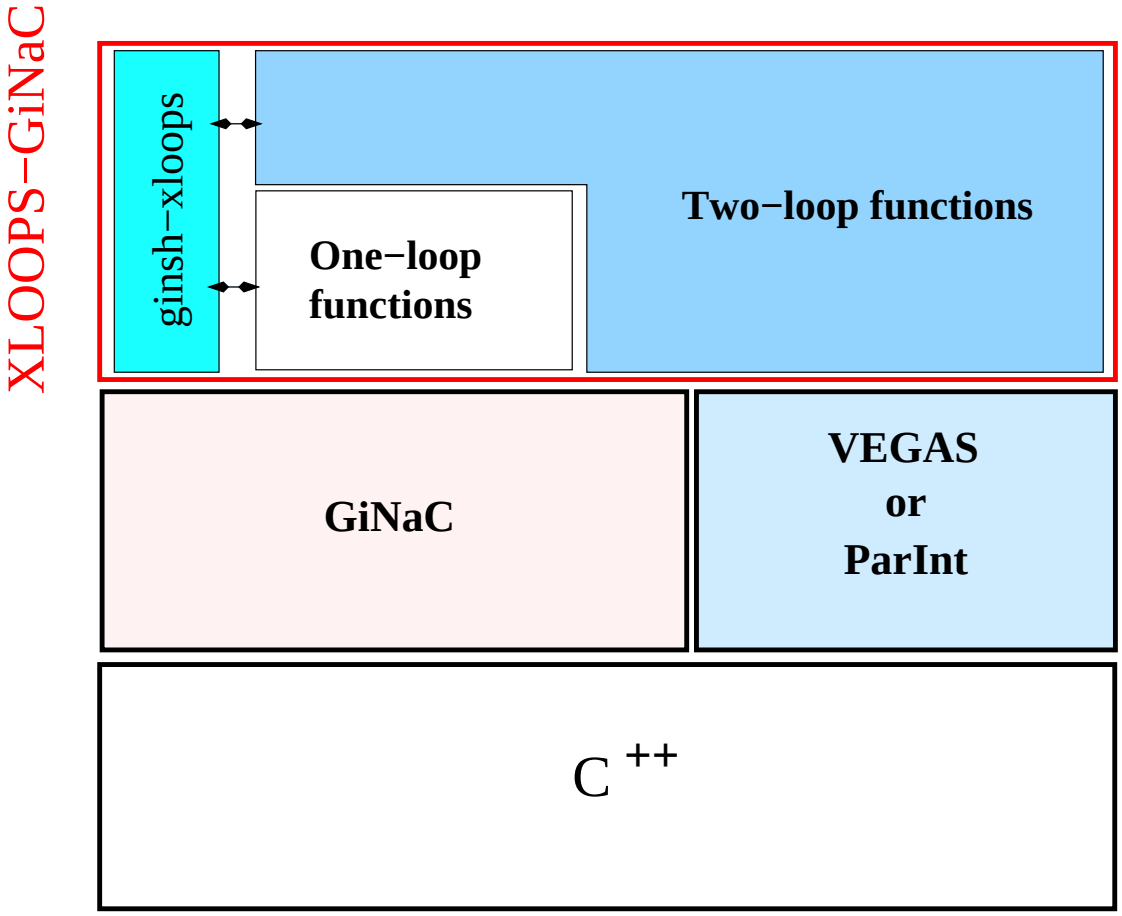


Figure 5.1: Structure of XLOOPS GiNaC

XLOOPS-GiNaC's one-loop functions are able to evaluate one-loop one-, two- and three-point integrals analytically with arbitrary masses and to any tensor rank. On top of one-loop functions, the two-loop functions separate a two-loop integral into divergent and finite parts. The divergent part is then evaluated analytically in terms of one-loop and hypergeometrical functions [38] while the finite part is reduced to a subset of two-fold

integrals which can be integrated numerically by VEGAS [44] or ParInt [45].[1]

XLOOPS-GiNaC calculates one- and two-loop Feynman integrals within the framework of dimensional regularization. The dimensional regularization is used to regulate both UV- and IR-divergences of one-loop functions and UV-divergences of two-loop functions. The IR-divergences of two-loop functions have to be regularized by introducing a small mass (mass-regularization).

As a general remark, in this chapter all source code of examples, their outputs and notations related to the XLOOPS-GiNaC program are written in `verbatim` text mode.

In the following sections, a detailed description of implemented functions for one- and two-loop integrals in XLOOPS-GiNaC is given.

5.1 XLOOPS-GiNaC one-loop functions

The following one-loop functions are implemented in XLOOPS-GiNaC.

5.1.1 Definition and prototype

All one-loop functions are implemented as GiNaC functions.[2] They can be called from *ginsh-xloops*, a version of the interactive GiNaC *shell*, or from a C/C++ program like usual C/C++ functions. The C/C++ prototype of a one-loop `N`-point function is

```
static ex OneLoopNPt( const ex & p0, ... , const ex & pperp,
                      const ex & m1, ... , const ex & mN,
                      const ex & t1, ... , const ex & tN,
                      const ex & rho );
```

[1] At the moment, only two-loop two-point integrals exist as described in this thesis.

[2] As any symbolic language, GiNaC allows a function to be declared as a symbolic object. Such a function is called a *GiNac function*. However, GiNaC is a C/C++ library, so that a GiNaC function is also a C/C++ function.

Here, for the time being, `N` can have values $1, 2, 3$. This is the implementation of the integral $T^{(1)\,(p_0 \ldots p_{J-1} p_\perp)}_{t_1 \ldots t_n}$ defined in chapter 2 (c.f. Eq. (2.22)).

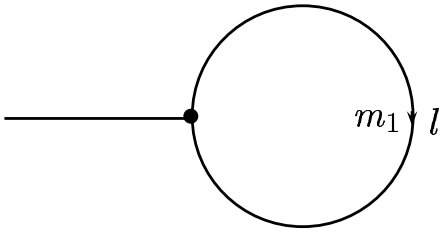
In addition, a set of functions `OneLoopTensNPt` for *conventional* Lorentz tensor integrals is provided. In contrast to the `OneLoopNPt` functions, the `OneLoopTensNPt` functions are implemented as C/C++ functions (so they can not be called from *ginsh-xloops*):

```
extern ex OneLoopTensNPt( unsigned n, const ex &m1,..., const ex &mN,
                          const ex &t1,..., const ex &tN, const ex &rho,
                          const ex &eps, GiNaC::lst &Cl,
                          const GiNaC::lst & indices = GiNaC::lst() );
```

The return value of the function is an analytical expression of type `ex`, a basis data type of GiNaC which handles a symbolic expression. The last argument of the `OneLoopTensNPt` functions is a `list` of Lorentz indices of the numerator. If this argument is not specified, XLOOPS-GiNaC will generate the indices automatically by using the `get_index()` method (see GiNaC document [51]).

The detailed description of the one-loop functions as well as their arguments is as follows:

- One-point functions



$$\texttt{OneLoop1Pt(p, m1, t1, rho)} := \int d^D l \, \frac{(l^2)^{\frac{p}{2}}}{(l^2 - m_1^2 + i\rho)^{t_1}}. \tag{5.1}$$

$$\texttt{OneLoopTens1Pt(n, m1, t1, rho, eps, C)} := \int d^D l \, \frac{l_{\mu_1} \ldots l_{\mu_n}}{(l^2 - m_1^2 + i\rho)^{t_1}}. \tag{5.2}$$

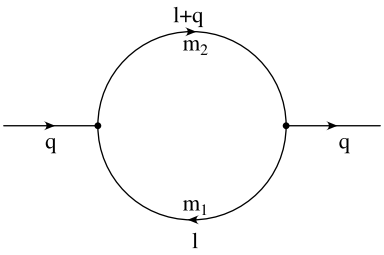
$$\texttt{Scalar1Pt(m1, rho)} := \int d^D l \, \frac{1}{(l^2 - m_1^2 + i\rho)}. \tag{5.3}$$

On the left-hand side we have written the XLOOPS-GiNaC functions which are associated to the Feynman loop integrals on the right-hand side. The arguments of these functions are self-defined from their associated integrals on the right-hand side. The causality parameter $\rho \geq 0$ is always specified explicitly in all XLOOPS-GiNaC functions. In fact, in a physical calculation, we are interested in the integrals at the limit $\rho \to 0$. The limit of ρ should be taken with care at the end of the calculation. The reason for its presence is simply a technical issue. For each single step of the evaluation, this parameter is needed to help GiNaC to determine the correct branch cut of its complex functions such as `log()` or `sqrt()`. We suggest that after substituting `rho` by zero at the end of the calculation, users should always check if the $\rho = 0$ result agrees with the result when `rho` tends to zero. This can be done by substituting `rho` with a small positive value. For instance, `rho` $= 10^{-15} m^2$ (where m is the smallest involved mass) is a good check.

All XLOOPS-GiNaC functions evaluate Feynman loop integrals in $D = 4 - 2\epsilon$ dimensions so that a dimensional parameter `eps` must be explicitly provided.

The return value of `OneLoopTensNPt()` is stored in the variable `C` (GiNaC data type `list`).

- Two-point functions



$$\texttt{OneLoop2Pt(p0, p1, q0, m1, m2, t1, t2, rho)} := \int d^D l \, \frac{l_0^{p_0} \, l_\perp^{p_1}}{[(l+q)^2 - m_1^2 + i\rho]^{t_1} \, [l^2 - m_2^2 + i\rho]^{t_2}} \tag{5.4}$$

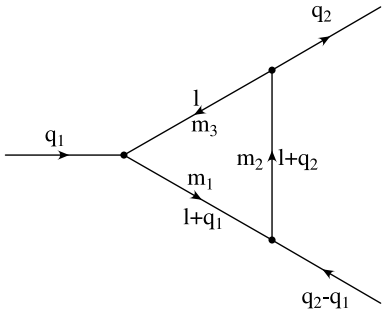
$$\texttt{OneLoopTens2Pt(n, q, m1, m2, t1, t2, rho, eps, C)} \tag{5.5}$$
$$:= \int d^D l \, \frac{l_{\mu_1} \ldots l_{\mu_n}}{\left[(l+q)^2 - m_1^2 + i\rho\right]^{t_1} \left[l^2 - m_2^2 + i\rho\right]^{t_2}}$$

$$\texttt{Scalar2Pt(q0, m1, m2, t1, t2, rho)} := \tag{5.6}$$
$$\int d^D l \, \frac{1}{\left[(l+q)^2 - m_1^2 + i\rho\right]^{t_1} \left[l^2 - m_2^2 + i\rho\right]^{t_2}}$$

The meaning of the arguments should again be obvious. The components of momenta in parallel and orthogonal spaces are

$$q_0 = \sqrt{q^2}; \qquad l_0 = \frac{l \cdot q}{\sqrt{q^2}}; \qquad l_\perp = \sqrt{l_0^2 - l^2}. \tag{5.7}$$

- Three-point functions



`OneLoop3Pt(p0, p1, p2, q10, q20, q21, m1, m2, m3, t1, t2, t3, rho)`

$$:= \int d^D l \, \frac{l_0^{p_0} \, l_1^{p_1} \, l_\perp^{p_1}}{P_1^{t_1} \, P_2^{t_2} \, P_3^{t_3}} \tag{5.8}$$

`OneLoopTens3Pt(n, q10, q20, q21, m1, m2, m3, t1, t2, t3, rho, eps, C)`

$$:= \int d^D l \, \frac{l_{\mu_1} \ldots l_{\mu_n}}{P_1^{t_1} \, P_2^{t_2} \, P_3^{t_3}} \tag{5.9}$$

`Scalar3Pt(q10, q20, q21, m1, m2, m3, t1, t2, t3, rho)`

$$:= \int d^D l \, \frac{1}{P_1^{t_1} \, P_2^{t_2} \, P_3^{t_3}} \tag{5.10}$$

where

$$\begin{aligned}
P_1 &= (l+q_1)^2 - m_1^2 + i\rho, \\
P_2 &= (l+q_2)^2 - m_2^2 + i\rho, \\
P_3 &= l^2 - m_3^2 + i\rho, \\
q_1 &= (q_{10}, 0, \vec{0}), \\
q_2 &= (q_{20}, q_{21}, \vec{0}), \\
l &= (l_0, l_1, \vec{l_\perp}), \\
q_{20} &= \frac{q_1 \cdot q_2}{\sqrt{q_1^2}}, \quad q_{21} = \sqrt{q_{20}^2 - q_2^2}.
\end{aligned}$$

5.1.2 Associated functions

- Series expansion

  The `series` expansion method is used to expand one- and two-loop functions and to obtain its divergent parts as a pole in the dimensional regularization parameter ϵ.

  The actual Laurent series expansion of an expression containing one-loop functions is done as usual using the GiNaC method

  ```
  ex ex::series(const ex &r, int order) const;
  ```

  The first argument of `series` is a relation which defines the parameter to be expanded and the expansion point. For an expression which contains one-loop functions, the first argument must be $\epsilon = 0$. The second argument of `series` is the number of terms with the lowest orders of the expansion. The following is an example of using *ginsh-xloops*

```
> a = OneLoop1Pt(0,m,1,rho);
Scalar1Pt(m,rho)
> s = a.series(eps==0,2);
(I*Pi^2*m^2)*eps^(-1)
+ (-I*Pi^2*log(m^2)*m^2+I*m^2*(Pi^2*(1-Euler)-Pi^2*log(Pi)))
+ Order(eps)
```

- Tensor reduction

  To simplify the output of a one-loop calculation, sometimes it is necessary to rewrite the output in terms of `ScalarNPt()` functions. For that purpose, a function `tensor_reduction()` is implemented. It reduces a tensor one-loop function to a linear combination of `ScalarNPt()` functions. The C/C++ prototype of the tensor reduction function is

  ```
  ex tensor_reduction(const ex &a, const ex &eps);
  ```

  where the first argument, `a`, is an expression containing one-loop functions and the second argument, `eps`, supplies the dimensional regularization parameter. A simple example to call `tensor_reduction()` in *ginsh-xloops* is:

  ```
  > a = OneLoop2Pt(1,0,q,m1,m2,1,1,rho);
  OneLoop2Pt(1,0,q,m1,m2,1,1,rho)
  > tensor_reduction(a,eps);
  1/2*q^(-1)*Scalar1Pt(m2,rho) - 1/2*q^(-1)*Scalar1Pt(m1,rho)
  + (-1/2*m2^2*q^(-1)+1/2*m1^2*q^(-1)-1/2*q)*Scalar2Pt(q,m1,m2,1,1,rho)
  ```

  Often it is useful to call `tensor_reduction()` first to obtain a readable output. After that, one can use the method `series()` to get an explicit result in terms of (poly)logarithmic functions.

- Abbreviations and the `subs_Rex()` function

  For the general case, the result of a series expansion contains expansion coefficients of $\mathcal{R}$-functions (see appendix B). These are implemented as C/C++ functions `R`x`ex`y`()`. They can be substituted by their corresponding (poly)logarithms and associated functions using the C/C++ function

  ```
  ex subs_Rex(const ex &e);
  ```

5.1.3 Special cases and limitations

The special cases for one-loop integrals handled by XLOOPS-GiNaC are listed in tables 5.1, 5.2, and 5.3.

If the one-loop functions are to be evaluated at critical points (such as vanishing masses or momenta), the values of the arguments must be supplied from the beginning. Evaluating the general function first and then substituting the variables at a critical point may yield singularities such as divided by zero.

All special cases of one-loop two- and three-point functions are listed in tables 5.1, 5.2 and 5.3. The notations used in these tables can be found in appendix A.3.

`OneLoop2Pt()`		
$q_0 \neq 0$		Reduced to one- or two-point scalar integrals and their derivatives (with respect to masses)
$q_0 = 0$	$m_1 = m_2$	Reduced to one-point tensor integrals
	$m_1 \neq m_2$	

Table 5.1: Special cases for tensor reduction of `OneLoop2Pt()`.

5.2 XLOOPS-GiNaC two-loop two-point functions

The following two-loop functions are implemented in XLOOPS-GiNaC

5.2.1 Definition and prototype

The two-loop two-point integrals are divided into two groups, factorizing and non-factorizing topologies.

The integrals associated with the non-factorizing topologies are implemented as C/C++ functions and can not be called from *ginsh-xloops*. The prototype of this kind of two-loop two-point functions is

`OneLoop3Pt()`				
$q_{10}=0$	$q_{21}=0$	$m_1=m_3$ $m_1\neq m_3$		Reduced to two-point integrals
	$q_{21}\neq 0$	$m_1=m_3$ $m_1\neq m_3$	$(q_{20}-q_{10})^2-q_{21}^2\neq 0$ $\wedge(q_{20}-q_{10})\neq 0$	Lorentz boost $\rightarrow$ two-point integrals
			$(q_{20}-q_{10})^2-q_{21}^2=0$ $\vee(q_{20}-q_{10})=0$	Reduced to two-point integrals
$q_{10}\neq 0$	$q_{21}=0$	$m_1=m_3$ $m_1\neq m_3$		Reduced to scalar three-point and two-point integrals
	$q_{21}\neq 0$	$m_1=m_3$ $m_1\neq m_3$	$(q_{20}-q_{10})^2-q_{21}^2\neq 0$ $\wedge(q_{20}-q_{10})\neq 0$	Reduced to scalar three-point and two-point integrals (by Lorentz boost)
			$(q_{20}-q_{10})^2-q_{21}^2=0$ $\vee(q_{20}-q_{10})=0$	Reduced to scalar three-point, Reduced to two-point integrals

Table 5.2: Special cases for tensor reduction of `OneLoop3Pt()` .

```
extern ex TwoLoop2PtN( const ex &p0, const ex &p1,const ex &r0,
                       const ex &r1, const ex &pz, const ex &q,
                       const ex &m , const ex &t, const ex &rho,
                       const ex &eps, const ex &NumParm );
```

where q is an external momentum. $N = 1, 2, 3, 4$ is the topology index (see below). The arguments `p0, p1, r0, r1, pz` are the exponents of l_0, $l_\perp$, k_0, $k_\perp$ and z, respectively.

Before calling a two-loop two-point function, one has to construct the following lists:

- a `list` of involved masses `m`

  ```
  lst m(m1, ..., mM);
  ```

  where M is the number of particles involved. Each component of a list can be of GiNaC or C/C++ data type.

- a `list` of exponents of the propagators `t`

  ```
  lst t(t1, ..., tM);
  ```

Scalar3Pt()				
$q_{10} = 0$	$m_1 = m_3$		$(q_{20} - q_{10}) \neq 0$	$\surd$
	$m_1 \neq m_3$		$(q_{20} - q_{10}) = 0$	not implemented
$q_{10} \neq 0$	IR	$q_{21} \neq 0$	$(q_{20} - q_{21}) = 0$	$\surd$
			$(q_{20} - q_{21} - q_{10}) = 0$	$\surd$
			$(q_{20} + q_{21}) = 0$	$\surd$
			$(q_{20} + q_{21} - q_{10}) = 0$	$\surd$
			$y_{kl}^{(3)} = 0 \wedge c_{kl} = 0$	$\surd$
			$y_{kl}^{(3)} = 0 \wedge c_{kl} \neq 0$	not implemented
			other cases	$\surd$
		$q_{21} = 0$	$q_{20} = 0$	$\surd$
			$(q_{20} - q_{10}) = 0$	$\surd$
			$(q_{20} - q_{10}) \neq 0$	not implemented

Table 5.3: All special cases of `Scalar3Pt()`. The abbreviation IR stands for possible IR-divergences cases (c.f. appendix A.3).

- a `list` of parameters for numerical integration `NumParm`

```
lst NumParm(x, y);
```

Other arguments are topology dependent and are defined below.

The return value of a non-factorizing two-loop two-point function is a `list` of two components. The first component is an expression representing the finite part, which is a function of two integration variables `x, y` (see section 4.2.7) which are declared in the argument `NumParm`. To integrate (numerically) the finite part, two numerical integration functions `VNumInt()` and `PNumInt()` are provided in XLOOPS-GiNaC. A detailed description of these two functions is given in section 5.2.2

The second component of the return value is an expression representing the analytical part which is reduced to one-loop functions. Thus all associated functions of one-loop functions can be applied to this term.

In contrast to the non-factorizing integrals, the integrals associated with the factorizing

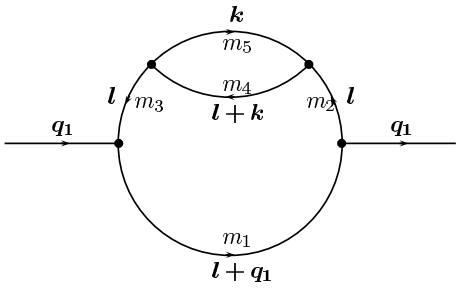
topologies are implemented as GiNaC functions. The prototype of this kind of two-loop two-point functions is the following:

```
static ex TwoLoop2Pt$N$( const ex &p0, const ex &p1,const ex &r0,
                         const ex &r1, const ex &pz, const ex &q,
                         const ex &m , const ex &t, const ex &rho )
```

where $N = 5, 6, 7, 8$ is the topology index explained in the following. The return value of a factorizing function is an expression of one-loop functions.

The detailed description of two-loop two-point functions are given below.

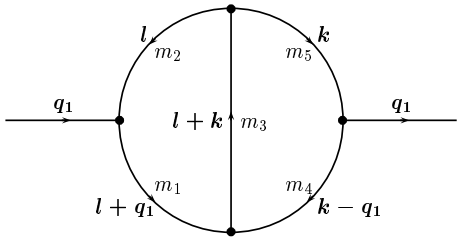
- Topology 1:



$$\texttt{TwoLoop2Pt1(p0, p1, r0, r1, s, q0, m, t, rho, eps, NumParm)}$$
$$:= \int d^D l \int d^D k \, \frac{(l_0)^{p_0}\,(l_1)^{p_1}\,(k_0)^{r_0}\,(k_1)^{r_1}\,(z)^s}{[(l+q_1)^2 - m_1^2 + i\rho]^{t_1}\,[l^2 - m_2^2 + i\rho]^{t_2}\,[l^2 - m_3^2 + i\rho]^{t_3}} \times \frac{1}{[(l+k)^2 - m_4^2 + i\rho]^{t_4}\,[k^2 - m_5^2 + i\rho]^{t_5}}$$

where $q_1 = (q_0, \vec{0})$ and $\vec{0}$ is null vector in the $D-1$ dimensional orthogonal space.

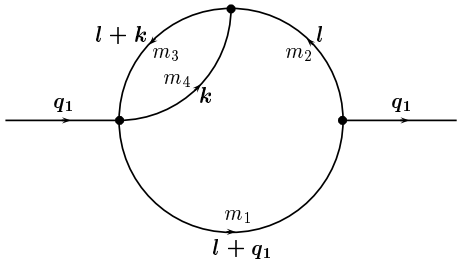
- Topology 2:



`TwoLoop2Pt2(p0, p1, r0, r1, s, q0, m, t, rho, eps, NumParm)`

$$:= \int d^D l \int d^D k \, \frac{(l_0)^{p_0} \, (l_1)^{p_1} \, (k_0)^{r_0} \, (k_1)^{r_1} \, (z)^s}{[(l+q_1)^2 - m_1^2 + i\rho]^{t_1} \, [l^2 - m_2^2 + i\rho]^{t_2} \, [(l+k)^2 - m_3^2 + i\rho]^{t_3}} \times \frac{1}{[(k-q_1)^2 - m_4^2 + i\rho]^{t_4} \, [k^2 - m_5^2 + i\rho]^{t_5}}$$

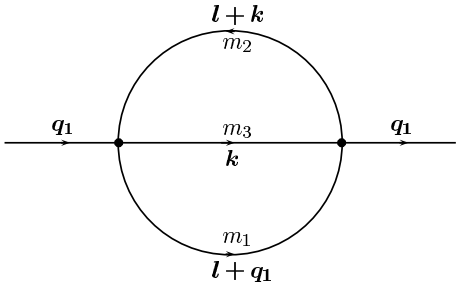
- Topology 3:



`TwoLoop2Pt3(p0, p1, r0, r1, s, q0, m, t, rho, eps, NumParm)`

$$:= \int d^D l \int d^D k \, \frac{(l_0)^{p_0} \, (l_1)^{p_1} \, (k_0)^{r_0} \, (k_1)^{r_1} \, (z)^s}{[(l+q_1)^2 - m_1^2 + i\rho]^{t_1} \, [l^2 - m_2^2 + i\rho]^{t_2}} \times \frac{1}{[(l+k)^2 - m_3^2 + i\rho]^{t_3} \, [k^2 - m_4^2 + i\rho]^{t_4}}$$

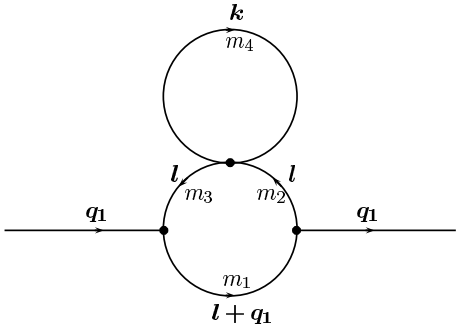
- Topology 4:



`TwoLoop2Pt4(p0, p1, r0, r1, s, q0, m, t, rho, eps, NumParm)`

$$:= \int d^D l \int d^D k \, \frac{(l_0)^{p_0} \, (l_1)^{p_1} \, (k_0)^{r_0} \, (k_1)^{r_1} \, (z)^s}{[(l+q_1)^2 - m_1^2 + i\rho]^{t_1} \, [(l+k)^2 - m_2^2 + i\rho]^{t_2} \, [k^2 - m_3^2 + i\rho]^{t_3}}$$

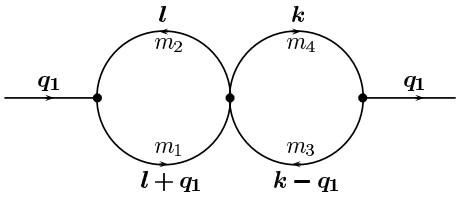
- Topology 5:



`TwoLoop2Pt5(p0, p1, r0, r1, s, q0, m, t, rho)`

$$:= \int d^D l \int d^D k \, \frac{(l_0)^{p_0} \, (l_1)^{p_1} \, (k_0)^{r_0} \, (k_1)^{r_1} \, (z)^s}{[(l+q_1)^2 - m_1^2 + i\rho]^{t_1} \, [l^2 - m_2^2 + i\rho]^{t_2}} \times \frac{1}{[l^2 - m_3^2 + i\rho]^{t_3} \, [k^2 - m_4^2 + i\rho]^{t_4}}$$

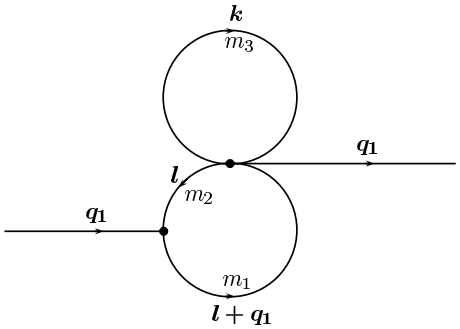
- Topology 6:



`TwoLoop2Pt6(p0, p1, r0, r1, s, q0, m, t, rho)`

$$= \int d^D l \int d^D k \, \frac{(l_0)^{p_0} \, (l_1)^{p_1} \, (k_0)^{r_0} \, (k_1)^{r_1} \, (z)^s}{[(l+q_1)^2 - m_1^2 + i\rho]^{t_1} \, [l^2 - m_2^2 + i\rho]^{t_2}} \times \frac{1}{[(k-q_1)^2 - m_3^2 + i\rho]^{t_3} \, [k^2 - m_4^2 + i\rho]^{t_4}}$$

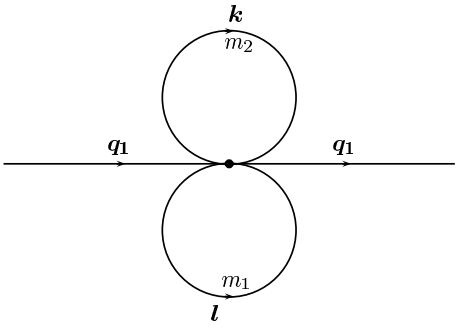
- Topology 7:



`TwoLoop2Pt7(p0, p1, r0, r1, s, q0, m, t, rho)`

$$= \int d^D l \int d^D k \, \frac{(l_0)^{p_0} \, (l_1)^{p_1} \, (k_0)^{r_0} \, (k_1)^{r_1} \, (z)^s}{[(l+q_1)^2 - m_1^2 + i\rho]^{t_1} \, [l^2 - m_2^2 + i\rho]^{t_2} \, [k^2 - m_3^2 + i\rho]^{t_3}}$$

- Topology 8:




```
TwoLoop2Pt8(p0, p1, r0, r1, s, q0, m, t, rho)
```

$$= \int d^D l \int d^D k \, \frac{(l_0)^{p_0} \, (l_1)^{p_1} \, (k_0)^{r_0} \, (k_1)^{r_1} \, (z)^s}{[(l+q_1)^2 - m_1^2 + i\rho]^{t_1} \, [k^2 - m_2^2 + i\rho]^{t_2}}$$

5.2.2 Associated functions

- All the one-loop associated functions can be applied to the divergent part of two-loop functions.

There are two supported functions for numerical integration, `VNumInt()` and `PNumInt()` which are based on the routines Vegas [44] and ParInt [45], respectively.

- The function `VNumInt()`

  The finite part of a two-loop two-point function can be integrated numerically using the Vegas routine [44] by calling the function `VNumInt()`

  ```
  extern ex VNumInt(const ex &integrand, int samples1, int iterations1,
                    int samples2, int iterations2, const ex &NumParm );
  ```

  The first argument, `integrand`, is integrated adaptively by Vegas. First, Vegas establishes a grid with `iterations1` iterations and `samples1` sample points per iteration. The established grid is then used for a second run. Only results of the steps in the second run are taken into account to estimate the final result.

A non-parallel C version of the routine NVegas [44] is included as part of the XLOOPS-GiNaC package.

The return value of `VNumInt()` is a list of four components. The first two components are the estimated values of the real part and its error. The last two components are the estimated values of the imaginary part and its error.

- The function `PNumInt()`
  A parallelized version of the numerical integration function is provided by the function `PNumInt()`

```
extern ex PNumInt(const ex &indr, int iterations,
                  const ex &NumParm, int node);
```

  The function `PNumInt()` calls an adaptive parallel quasi-Monte Carlo routine of the program package ParInt [45]. The number of iterations is specified in the variable `iterations`. The number of computers in the parallel network is declared in the variable `node`.

  The return value of the function `PNumInt()` has the same structure as that of the function `VNumInt()`.

  To use this function, ParInt and MPICH (a GNU/GPL version of Message-Passing Interface needed for parallel computing) must be installed. A detailed description of ParInt and MPICH can be found in [45] and [50].

5.2.3 Limitations and performance

The two-loop two-point functions have the following limitations:

- The speed of an evaluation depends strongly on the power of the available CPU and the memory. For a complicated calculation, such as a two-loop two-point tensor rank four or higher, a computer with 512 MB of RAM is recommended.

- Users may have to smooth the integrand manually before calling `VNumInt()` or `PNumInt()` to obtain a stable result. For instance, the transformation.

$$\int_0^1 dx\ f(x) \to \int_0^1 dy\ \frac{f(x)}{\beta x^{\beta-1}}\bigg|_{y=x^\beta}, \qquad \beta > 1 \tag{5.11}$$

  may help when $f(x)$ has a large peak at $x = 1$. We refer readers to [44] for more detail.

In order to provide a quick look at the performance of XLOOPS-GiNaC, we calculate the master two-loop two-point integral

$$\int d^D l \int d^D k \frac{l_0^{p_0} l_\perp^{p_\perp} k_0^{r_0} k_\perp^{r_\perp}}{P_1(l+q)P_2(l)P_3(l+k)P_4(k-q)P_5(k)} \tag{5.12}$$

with

$$\begin{aligned} P_1(l+q) &= (l+q)^2 - m_1^2 + i\rho, \\ P_2(l) &= l^2 - m_2^2 + i\rho, \\ P_3(l+k) &= (l+k)^2 - m_3^2 + i\rho, \\ P_4(k-q) &= (k-q)^2 - m_4^2 + i\rho, \\ P_5(k) &= k^2 - m_5^2 + i\rho. \end{aligned} \tag{5.13}$$

In Fig. 5.2, the CPU time of this calculation is plotted as a function of the tensor degree d where [3]

$$d = p_0 + p_\perp + r_0 + r_\perp. \tag{5.14}$$

In the figure, at each value of d, the time of calculation is presented as a bar. In fact, the calculation time depends on the values of p_0, $p_\perp$, r_0, $r_\perp$. A fit with the function

$$f(d) = \exp(A\,d + B) \tag{5.15}$$

[3] We do not take into account the time for the numerical evaluation since it depends on the smoothness of the integrand and the number of sampling points.

is made to see the trend. A and B are fit parameters. From the plot one can observe that the CPU time increases exponentially with the tensor degree d. In fact, when d gets large, the number of terms in the tensor reduction as well as in the subtraction procedure increase exponentially. Handling such large amount of terms requires more computer resources and thus slows down the speed of the calculation.

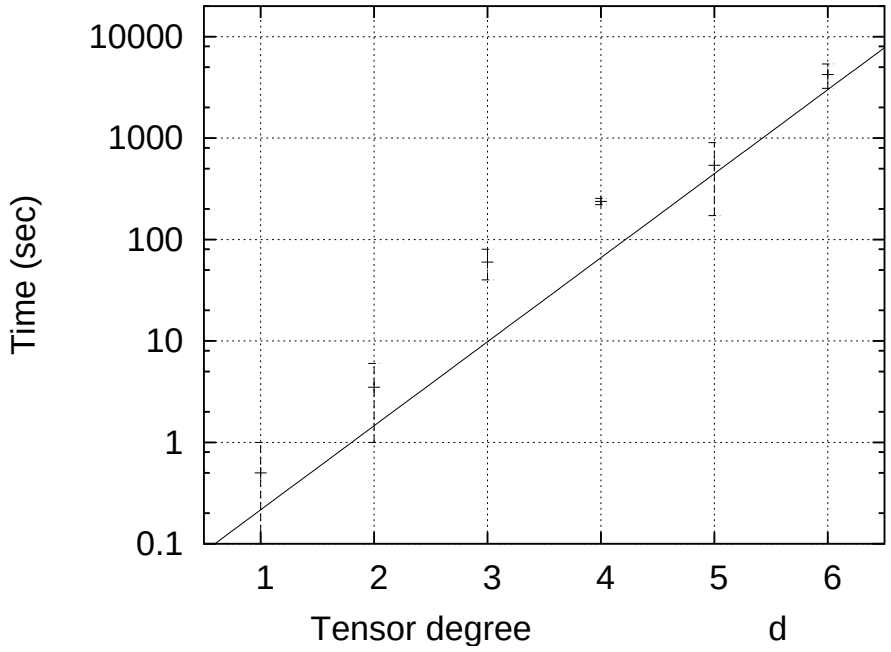


Figure 5.2: Performance of two-loop two-point functions

5.3 Examples for one- and two-loop integrals

5.3.1 One-loop examples

XLOOPS-GiNaC is implemented as a C/C++ library (i.e. an Application Programming Interface, or, API). Therefore its functions can be called from a user's application program to evaluate a given Feynman diagram.

This section specifies how to use XLOOPS-GiNaC in this fashion, including some explanations of some sample codes and details on how to compile and link against the XLOOPS-GiNaC library.

It is easiest to start with a simple example of an application program before going into more details. Let us consider the program presented in Example 1. At first, this example demonstrates how to compile a user application program using the XLOOPS-GiNaC library. Second, it demonstrates how to perform a Lorentz decomposition of a tensor integral

$$\int d^{4-2\epsilon}l \, \frac{l^{\mu_1} \, l^{\mu_2}}{[(l+q)^2 - m_1^2 + i\rho][l^2 - m_2^2 + i\rho]} \tag{5.16}$$

by using the function `OneLoopTens2Pt()` and how its result looks like.

- Example 1: `test.cpp`

```
1  #include <iostream>
2  #include "xloops.h"
3  using namespace std;
4  using namespace xloops;
5
6  int main(void)
7  {
8    symbol  q("q"), m1("m1"), m2("m2"),  eps("eps"), rho("rho");
9    lst C;
10   ex T = OneLoopTens2Pt(2, q, m1, m2, 1, 1, rho, eps, C);
11   cout << "Tensor structure: " << T << endl;
12   cout << "Coefficients: " << C << endl;
13   return 0;
14 }
```

Save this file with the name `test.cpp` and suppose that the GiNaC and XLOOPS-GiNaC libraries are already installed in the directories `/project/GiNaC` and `/project/xloops`, respectively. In a GNU/Linux system, the above program can be compiled as follows

```
$ c++ test.cpp -o test -I/project/GiNaC/ginac -I/project/xloops/lib
   -L/project/GiNaC/ginac/.libs -lginac -L/project/xloops/lib -lxloops
```

The command creates an executable file `test`. Now, one can execute the program `test` from the command line

```
$ ./test
```

The output looks like this

```
Tensor structure: q~mu1*q~mu2*C20 + eta~mu1~mu2*C21
Coefficients: {  C21 == -(3-2*eps)^(-1)*OneLoop2Pt(0,2,q,m1,m2,1,1,rho),
                 C20 == (3-2*eps)^(-1)*q^(-2)*OneLoop2Pt(0,2,q,m1,m2,1,1,rho)
                        + q^(-2)*OneLoop2Pt(2,0,q,m1,m2,1,1,rho)             }
```

where `eta~mu1~mu2` is the metric tensor $g^{\mu_1\mu_2}$. The meaning of the varous lines of the program `test.cpp` is as follows.

The header file `xloops.h` and the command `namespace xloops` are included (line 2 and line 4) to provide the necessary prototypes and definitions.

Inside `main()`, in line 8, the masses m_1, m_2, the external momentum q, the dimensional parameter ϵ and the causality parameter ρ are declared as variables of type `symbol`, a GiNaC data type.

In line 10, the function `OneLoopTens2Pt()` is called. The first argument of this function is the tensor rank, which is equal to two in this example. The return value `T` is a Lorentz decomposition

$$T = q^{\mu_1} q^{\mu_2} C_{20} + g^{\mu_1\mu_2} C_{21}. \tag{5.17}$$

The coefficients C_{20} and C_{21} are stored in a list `C` which is declared in line 9. Each coefficient C_{20} and C_{21} is a function of one-loop functions which again can be evaluated in the same manner from a user's program or from *ginsh-xloops* as presented in section 5.1.2. For instance, the following C/C^{++} code will be used to evaluate the coefficient C_{20}

```
1 ------------------------------------------------------------
2 ..........
3 ex res = ((C.op(1)).op(1)).subs(m1==m2);
4 cout << "Coefficient C20: " << endl;
5 cout << ((res.series(eps==0,2))).normal() << endl;
6 ..........
7 ------------------------------------------------------------
```

In line 3 of the above code, we extract the expression on the right hand side of the first member of the list C, i.e. of C_{20}. To have a simple output to present here, we consider the simple case $m_1 = m_2$. The result is series-expanded at the pole $\epsilon = 0$ up to order $\mathcal{O}(\epsilon^1)$ and then printed to the standard output:

```
Coefficient C20:
(-(1/12*I)*Pi^2*q^2+(1/2*I)*Pi^2*m2^2)*eps^(-1)
+ (-(1/2*I)*Pi^2*m2^2*log(Pi)+1/12*Pi^3*q^2 -(1/6*I)*Pi^2*m2^2*log(m2^2)
  -(1/2*I)*Pi^2*m2^2*Euler+(1/12*I)*Pi^2*Euler*q^2-(2/9*I)*Pi^2*q^2
  +(7/6*I)*Pi^2*m2^2-1/3*Pi^3*m2^2+(1/12*I)*Pi^2*log(Pi)*q^2
  -(1/12*I)*Pi^2*R2ex1(I*rho-m2^2,-1/4*q^2)*q^2
  +(1/3*I)*Pi^2*m2^2*R2ex1(I*rho-m2^2,-1/4*q^2))
+ Order(eps)
```

5.3.2 A two-loop example

Having discussed a one-loop example, we now turn to a two-loop example. Example 2 is used to calculate the sunrise two-loop integral

$$\int d^D l \; d^D k \frac{1}{[(l+q)^2 - m_1^2][(l+k)^2 - m_2^2][k^2 - m_3^2]} \tag{5.18}$$

- Example 2:

```
1 #include <iostream>
2 #include "xloops.h"
3 using namespace std;
4 using namespace xloops;
5
```

```
6  void main(int argc, char **argv)
7  {
8    symbol m1("m1"), m2("m2"), m3("m3"), q("q"), eps("eps"), rho("rho");
9    symbol x("x[0]"), y("x[1]");
10   lst NumParm(x, y);
11   lst m(m1, m2, m3);
12   lst  t(1,1,1);
13
14   ex c = TwoLoop2Pt4(0,0, 0, 0, 0, q, m, t , rho, eps, NumParm);
15
16   ex UV = (c.op(1)).series(eps==0,4);
17   ex res1 = normal(UV.coeff(eps,-2))*pow(eps,-2)
18             + normal((UV.coeff(eps,-1)).subs(rho==0))*pow(eps,-1);
19   ex temp = (UV.coeff(eps,0)).subs(lst(q == 175, m1==175, m2==175,
20                                        m3==175, rho==pow(10,-15)));
21
22   ex finite = (c.op(0)).subs(lst(q == 175, m1==175, m2==175,
23                                  m3==175, rho==pow(10,-15)));
24
25   ex res = VNumInt(finite, 10000, 5, 100000, 5, NumParm);
26
27   cout << "The divergent part is: " << endl;
28   cout << res1 << endl;
29   cout << "The value of the finite term at the pseudo-threshold is" << endl;
30   cout <<  (res.op(0)+res.op(2)*I+temp).evalf()  << endl;
31   cout << "with errors of the real and imaginary part: " << endl;
32   cout <<  "+/-" << abs(res.op(1)) << "and +/-" << abs(res.op(3))*I << endl;
33 }
```

As in Example 1, the header file `xloops.h` is included in line 3 to provide the prototypes and definitions needed for the one- and two-loop functions.

From lines 8 to 12, masses, momentum and other parameters are declared.

In line 14, the function `TwoLoop2Pt4()` is called. The evaluated value is stored in the variable `c`.

The analytical part of the result is extracted, series-expanded and then stored in `UV`, line 16. The coefficients of the $\mathcal{O}(\epsilon^{-2})$ and $\mathcal{O}(\epsilon^{-1})$ terms are collected in `res1` (lines 17-18) and printed out as the divergent part in lines 27 - 28. The coefficient of the $\mathcal{O}(\epsilon^0)$ term of the analytical part is collected in `temp` (line 19). Because the $\mathcal{O}(\epsilon^0)$ term should be summed with the value of the numerical part obtained in later steps (line 30), explicit numerical

values of masses, external momentum and parameters are provided by a substitution in line 19.

The numerical part of `c` is collected in `finite` (lines 22-23) and can be used in the numerical integration via Vegas. Explicit numerical values of masses, external momentum and parameters must be provided. The numerical values of masses, the momentum q and parameters are chosen explicitly at the pseudo-threshold ($m_1 = m_2 = m_3 = \sqrt{q^2} = 175$). For this case the analytical result is known [29].

From the output of the program

```
--------------------------------------------------------------------------
The divergent part is:

(-1/2*Pi^4*m2^2-1/2*Pi^4*m1^2-1/2*Pi^4*m3^2)*eps^(-2)+
(-3/2*Pi^4*m2^2+Pi^4*m3^2*log(Pi)+Pi^4*log(m1^2)*m1^2-3/2*Pi^4*m1^2
 +Pi^4*m3^2*log(m3^2)+1/4*Pi^4*q^2+Pi^4*m2^2*log(m2^2)+Pi^4*log(Pi)*m2^2
 +Pi^4*m3^2*Euler+Pi^4*Euler*m1^2-3/2*Pi^4*m3^2+Pi^4*log(Pi)*m1^2
 +Pi^4*m2^2*Euler)*eps^(-1)

The value of the finite term at the pseudo-threshold is:

-1.0772268158E9+51943.5*I

with errors of the real and imaginary part:

+/-136.3 and +/-165.7*I
--------------------------------------------------------------------------
```

we see that, at the pseudo-threshold ($m_1 = m_2 = m_3 = \sqrt{q^2} = 175$), this result agrees with the analytical results of [29, 48].

The above two examples and a `Makefile` script can be found in the directory `example` of the XLOOPS-GiNaC package.

In the program package XLOOPS-GiNaC one can find more examples which may help the beginners to start their own programs easier.

Chapter 6

The g_2 form factor from a theory with spontaneously broken color

In this chapter, we will present an application of XLOOPS-GiNaC to calculate the g_2 form factor of the nucleon-nucleon-ρ-meson vertex at the one-loop level. In this calculation, the interaction of nucleons and mesons is considered in the frame-work of an effective theory of strong interactions at long distances [22]. In this effective theory, the vacuum of QCD is characterized by the Higgs mechanism. Color is "spontaneously broken" by a quark-antiquark condensate in the octet representation. The vector mesons transform like the massive gluons. The baryons transform like quarks. The identification of massive gluons with the vector mesons and of the quarks with the baryons are called gluon-meson and quark-baryon dualities.

In this theory, the nucleon-nucleon-ρ-meson vertex is identified with a quark-quark-gluon vertex where quarks have the nucleon mass and the gluon takes the mass of the ρ-meson. In the next section, we will calculate the g_2 form factor of the nucleon-nucleon-ρ-meson vertex at the one-loop level. We are interested in the g_2 form factor because it can be measured from nuclear experiments. That means, the g_2 form factor is sensitive to a test of the theory.

6.1 The g_2 form factor at one-loop order

There are two Feynman diagrams that contribute to the quark-quark-gluon vertex. The

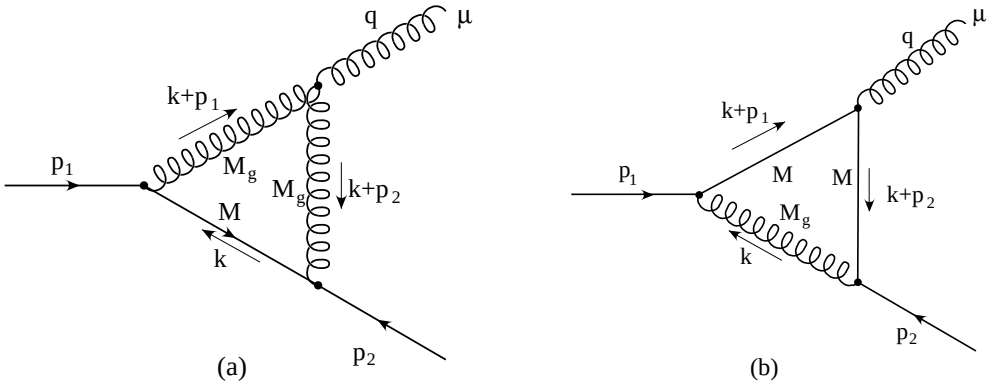


Figure 6.1: Feynman diagrams contributing to the quark-quark-gluon vertex

amplitude is written as

$$\mathcal{A}^{\mu} \;=\; \bar{u}(p_2)\,\Gamma^{\mu}\,u(p_1)\,\epsilon^{\mu}. \tag{6.1}$$

In this calculation, we follow the convention in [52] for the Feynman rules.
In the general covariant gauge, the contributions of the two Feynman graphs in Fig. 6.1(a) and Fig. 6.1(b) are given by

$$\begin{aligned}
\Gamma^{\mu\,(a)} \;=\; & \int \frac{d^D\,k}{(2\pi)^D}\,\left\{(ig_s)\,t^k\,\gamma^{\beta}\,\frac{i\,(-\not{k}+M)}{k^2-M^2+i\rho}\,(ig_s)\,t^j\,\gamma^{\alpha}\right. \\
& \frac{i}{(k+p_1)^2-M_g^2+i\rho}\left(-g_{\alpha\nu}+(1-\xi)\frac{(k+p_1)_{\alpha}(k+p_1)_{\nu}}{(k+p_1)^2-\xi M_g^2+i\rho}\right) \\
& g_s\,f^{ijk}\,((-q-k-p_1)^{\sigma}\,g^{\mu\nu}+(2k+p_1+p_2)^{\mu}\,g^{\nu\sigma}+(q-k-p_2)^{\nu}\,g^{\sigma\mu}) \\
& \left.\frac{i}{(k+p_2)^2-M_g^2+i\rho}\left(-g_{\sigma\beta}+(1-\xi)\frac{(k+p_2)_{\sigma}(k+p_2)_{\beta}}{(k+p_2)^2-\xi M_g^2+i\rho}\right)\right\}.
\end{aligned} \tag{6.2}$$

$$\begin{aligned}
\Gamma^{\mu\,(b)} \;=\; & \int \frac{d^D\,k}{(2\pi)^D}\,\left\{(i\,g_s)\,t^j\,\gamma^{\alpha}\,\frac{i\,(\not{k}+\not{p}_2+M)}{(k+p_2)^2-M^2+i\rho}(i\,g_s)\,t^i\,\gamma^{\mu}\,\frac{i\,(\not{k}+\not{p}_1+M)}{(k+p_1)^2-M^2+i\rho}\right. \\
& \left.(i\,g_s)\,t^j\,\gamma^{\nu}\,\frac{i}{k^2-M_g^2+i\rho}\left(-g_{\nu\alpha}+(1-\xi)\frac{k_{\nu}k_{\alpha}}{k^2-\xi M_g^2+i\rho}\right)\right\}.
\end{aligned} \tag{6.3}$$

where M is the nucleon mass, M_g is the mass of the ρ-meson, $q = p_1 - p_2$ and $D = 4-2\epsilon$. Next, one parametrizes the amplitude according to the decomposition

$$\mathcal{A}^\mu \;=\; (ig_s)\, t^i\, \bar{u}(p_2) \left(\gamma^\mu \; F_1(q^2) + \frac{i\, \sigma^{\mu\nu}\, q_\nu}{2M} \; F_2(q^2) \right) u(p_1)\; \epsilon^\mu. \tag{6.4}$$

The anomalous magnetic moment g_2 is identify with the Pauli form factor $F_2(0)$ at zero momentum transfer $q^2 = 0$.

After doing the color algebra and the loop integration one finds the following analytical results in the Landau gauge ($\xi = 0$) for the Pauli form factor $F_2(q^2 = 0)$

$$\begin{aligned} F_2^{(a)}(q^2=0) \;&=\; \frac{g_s^2}{16\,\pi^2}\frac{C_A}{M^4}\left\{ M^2(3M^2-2M_g^2) + (M^4 - 3\,M^2M_g^2 + M_g^4)\ln\left(\frac{M_g^2}{M^2}\right) \right. \\ &\left. -\frac{iM_g^2(5M^4-5M^2M_g^2+M_g^4)}{\sqrt{4M^2M_g^2-M_g^4}}\ln\left(\frac{M_g^2 - i\sqrt{4M^2M_g^2-M_g^4}}{M_g^2+i\sqrt{4M^2M_g^2-M_g^4}}\right)\right\}, \end{aligned} \tag{6.5}$$

$$\begin{aligned} F_2^{(b)}(q^2=0) \;&=\; -\frac{g_s^2}{16\,\pi^2}\left(C_F - \frac{1}{2}C_A\right)\frac{2}{M^4} \\ &\left\{ M^2(-M^2+2M_g^2) - M_g^2(-M^2+2M_g^2)\ln\left(\frac{M_g^2}{M^2}\right) \right. \\ &\left. +\frac{iM_g^2(2M^4-4M^2M_g^2+M_g^4)}{\sqrt{4M^2M_g^2-M_g^4}}\ln\left(\frac{M_g^2 - i\sqrt{4M^2M_g^2-M_g^4}}{M_g^2+i\sqrt{4M^2M_g^2-M_g^4}}\right)\right\} \end{aligned} \tag{6.6}$$

where the two color factors C_A and C_F are given by

$$C_A \;=\; N_C\;, \qquad\qquad C_F \;=\; \frac{N_C^2-1}{2N_C}\;. \tag{6.7}$$

The contributions are purely real and UV convergent.
We have repeated the calculation in the general covariant gauge and also in the unitary gauge in which the gluon propagator takes the form

$$\tilde{G}_{\mu\nu}(k) \;=\; \frac{-i}{k^2 - M_g^2 + i\rho}\left(g_{\mu\nu} - \frac{k_\mu k_\nu}{M_g^2+i\rho}\right) \tag{6.8}$$

and found the result for $F_2(q^2 = 0)$ to be gauge-independent. However, for $q^2 \neq 0$ one finds a gauge-dependent result for the two contributions.

The numerical results with $M_g = 0.77$ GeV, $M = 0.93827$ GeV are

$$F_2^{(a)}(q^2=0) = -\frac{g_s^2}{16\,\pi^2}\,C_A\;(0.27805650140). \tag{6.9}$$

$$F_2^{(b)}(q^2=0) = \frac{g_s^2}{16\,\pi^2}\left(C_F - \frac{1}{2}C_A\right)\;(0.50897242443). \tag{6.10}$$

In order to compare with the result of [53] we take the limit $M_g \to 0$ and obtain

$$F_2^{(a)}(q^2=0)\Big|_{M_g\to 0} = \frac{g_s^2}{16\pi^2}\,C_A\left\{3 + \ln\left(\frac{M_g^2}{M^2}\right)\right\}\;, \tag{6.11}$$

Using the one-loop substitution rule

$$\ln\left(\frac{M_g^2}{M^2}\right)\Big|_{M_g\to 0} \leftrightarrow \frac{1}{\epsilon_{IR}} - \gamma_E + \ln\left(\frac{4\pi}{M^2}\right) \tag{6.12}$$

we reproduce the QCD result of [53] given in dimensional regularization

$$F_2^{(a)}(q^2=0)\Big|_{M_g\to 0} = \frac{g_s^2}{16\pi^2}\,C_A\left\{\frac{1}{\epsilon_{IR}} + 3 - \gamma_E + \ln\left(\frac{4\pi}{M^2}\right)\right\}\;, \tag{6.13}$$

Note that in the limit $M_g \to 0$, the Pauli form factor $F_2^{(a)}(q^2=0)$ is IR-divergent.

In the limit $M_g \to 0$ and setting $C_A = 0$, $C_F = 1$, $g_s \to e$ the Pauli form factor $F_2^{(b)}(q^2=0)$ from Fig. 6.1(b) reads

$$F_2^{(b)}(q^2=0)\Big|_{M_g\to 0} = \frac{e^2}{8\pi^2} = \frac{\alpha}{2\pi}\;, \tag{6.14}$$

which is in agreement with the known QED result first derived by Schwinger in 1948.

To obtain the factors $F_2^{(a)}(q^2)$ and $F_2^{(b)}(q^2)$, we used XLOOPS-GiNaC to do the tensor reduction and the loop integration. The results of XLOOPS-GiNaC then are exported to Mathematica for further simplification.

Chapter 7

Conclusions

In this thesis, we have presented an algorithm for the calculation of one-loop one-, two-, three-point and two-loop two-point Feynmann integrals in the framework of the parallel and orthogonal space method.

For the one-loop case, all the tensor integrals are first reduced to scalar functions and then analytically evaluated in terms of $\mathcal{R}$-functions. The algorithm for the one-loop calculation is actually well-known since long and was already implemented in the Maple version of XLOOPS.

For the two-loop two-point case, we propose a new subtraction procedure to solve the degenerate two-loop two-point integrals. It turns out that all the degenerate two-loop two-point integrals can be reduced to the non-degenerate case and analytical two-loop integrals (such as vacuum and $\widetilde{J}_1$ integrals) by using an appropriate subtraction procedure.

The new procedure completes the method of [12] and opens a systematic way to solve arbitrary two-loop two-point tensor integrals. The nightmare of the so called "sunset problem" is no longer there in the XLOOPS project.

The algorithm for the calculation of one-loop one-, two-, three-point and two-loop two-point Feynmann integrals is successfully implemented in the program package XLOOPS-GiNaC. The program is then tested by various checks. For the one-loop modules, we

used the program to calculate a concrete problem, the g_2 form factor from a theory with spontaneously broken color (see chapter 6). The result agrees with that of other authors in various limiting cases. For the two-loop modules, we have presented many comparisons with the known numerical as well as analytical results (see section 4.3). High accuracy results of two-loop two-point integrals can be obtained in a reasonable time. All the source codes of those tests are included as part of XLOOPS-GiNaC.

In the first part of the thesis, we have attempted to present the calculations as close as possible to their implementation in the XLOOPS-GiNaC program package so that this part can serve as a comprehensive documentation for further developments. In contrast, the second part is dedicated to the end users. Therefore, the explicitness and the ease of use are our priorities. In this part, the implementation of the one-loop and two-loop procedures is realized in the XLOOPS-GiNaC program package, a C/C^{++} library.

With the present implementation, IR-divergences of two-loop integrals have to be avoided by choosing non-zero masses where needed. Future work should provide an effective and consistent procedure for the extraction of IR-divergent parts. This is needed in order to cancel these contributions with corresponding terms coming from photons or gluon emission.

Appendix A

Scalar one-loop integrals

In this section, we provide the explicit solutions of the tadpole $A_t^n(m^2)$ (c.f. Eq. (2.7)) and the scalar one-loop integrals $B_0(q^2; m_1^2, m_2^2)$ and $C_0(q_1^2, q_2^2; m_1^2, m_2^2, m_3^2)$ (c.f. Eqs. (3.3, 3.20)). Details of the calculation of these integrals in terms of $\mathcal{R}$-function can be found in [11, 13]. Other references can be found in [54].

A.1 One-loop tadpole integral

$$\begin{aligned} A_t^n(m^2) &= \int \frac{d^D l}{(2\pi)^D} \frac{(l^2)^n}{[l^2+m^2]^t} \\ &= i\frac{(-1)^{n-t}}{(4\pi)^2} \left(\frac{4\pi}{m^2}\right)^{\epsilon} (m^2)^{2+n-t} \frac{\Gamma(2+n-\epsilon)}{\Gamma(2-\epsilon)} \frac{\Gamma(t-n-2+\epsilon)}{\Gamma(t)}. \end{aligned} \tag{A.1}$$

A.2 Scalar one-loop two-point integral B_0

$$
\begin{aligned}
B_0(q^2;m_1,m_2) &= \int \frac{d^Dl}{(2\pi)^D}\frac{1}{[(l+q)^2-m_1^2+i\rho][l^2-m_2^2+i\rho]} \qquad (A.2)\\
&= \frac{i}{(2\pi)^D}\,\frac{e^{i\pi\epsilon}}{2q_0}\,\frac{\pi^{\frac{D-1}{2}}}{(1/2-\epsilon)}\,\Gamma\left(\frac{1}{2}-\epsilon\right)\,\mathcal{B}\left(\frac{1}{2},\epsilon\right)\,\times\\
&\quad\left[\;\left(\frac{q_0}{2}+M_d\right)\;\mathcal{R}_{-\epsilon}\left(-\frac{1}{2}+\epsilon,1;-m_1^2+i\rho,-\left(\frac{q_0}{2}+M_d\right)^2\right)\right.\\
&\quad\left.+\;\left(\frac{q_0}{2}-M_d\right)\;\mathcal{R}_{-\epsilon}\left(-\frac{1}{2}+\epsilon,1;-m_2^2+i\rho,-\left(\frac{q_0}{2}-M_d\right)^2\right)\;\right]
\end{aligned}
$$

where

$$
\begin{aligned}
M_d &= \frac{m_1^2-m_2^2}{2\,q_0},\\
q &= (q_0,\vec{0})
\end{aligned}
$$

and the $\mathcal{R}$-function is defined in appendix B.

A.3 Scalar one-loop three-point integral C_0

$$
C_0(q_1^2,q_2^2;m_1^2,m_2^2,m_3^2)=\int\frac{d^Dl}{(2\pi)^D}\frac{1}{[(l+q_1)^2-m_1^2+i\rho][(l+q_2)^2-m_2^2+i\rho][l^2-m_3^2+i\rho]}
$$

$$
\begin{aligned}
&= -i\,\pi^{2-\epsilon}\,\Gamma(\epsilon)\,\mathcal{B}(2\epsilon,1)\sum_{k=1}^{3}\frac{I_R(k)}{a_{k1}b_{k2}-a_{k2}b_{k1}}\sum_{l=1}^{2}(-1)^l\;S_{kl}^{-\epsilon}\;\times \qquad (A.3)\\
&\left[\;\mathcal{R}_{-2\epsilon}\left(\epsilon,\,\epsilon,\,1;\;y_{kl}^{(1)},\,y_{kl}^{(2)},\,y_{kl}^{(3)}\right)+\mathcal{R}_{-2\epsilon}\left(\epsilon,\epsilon,1;-y_{kl}^{(1)},-y_{kl}^{(2)},-y_{kl}^{(3)}\right)\;\right]
\end{aligned}
$$

with the following abbreviations

$$\begin{aligned} y_{kl}^{(1)} &= \frac{c_{kl} + \sqrt{c_{kl}^2 + (b_{kl}^2 - a_{kl}^2)(m_k^2 - i\rho)}}{a_{kl} - b_{kl}}, \\ y_{kl}^{(2)} &= \frac{c_{kl} - \sqrt{c_{kl}^2 + (b_{kl}^2 - a_{kl}^2)(m_k^2 - i\rho)}}{a_{kl} - b_{kl}}, \\ y_{kl}^{(3)} &= \frac{\frac{c_{k2}}{a_{k2} + b_{k2}} - \frac{c_{k1}}{a_{k1} + b_{k1}}}{\frac{a_{k2}}{a_{k2} + b_{k2}} - \frac{a_{k1}}{a_{k1} + b_{k1}}}, \\ S_{kl} &= \frac{a_{kl} - b_{kl}}{a_{kl} + b_{kl}} - i\rho \end{aligned} \tag{A.4}$$

and

$$\begin{aligned} a_{11} &= 2(q_{20} - q_{10}) & a_{12} &= -2q_{10} \\ b_{11} &= -2q_{21} & b_{12} &= 0 \\ c_{11} &= (q_{20} - q_{10})^2 - q_{21}^2 - m_2^2 + m_1^2 & c_{12} &= q_{10}^2 - m_3^2 + m_1^2 \\ \\ a_{21} &= -2(q_{20} - q_{10}) & a_{22} &= -2q_{20} \\ b_{21} &= 2q_{21} & b_{22} &= 2q_{21} \\ c_{21} &= (q_{20} - q_{10})^2 - q_{21}^2 - m_1^2 + m_2^2 & c_{22} &= q_{20}^2 - q_{21}^2 - m_3^2 + m_2^2 \\ \\ a_{31} &= 2q_{10} & a_{32} &= 2q_{20} \\ b_{31} &= 0 & b_{32} &= -2q_{21} \\ c_{31} &= q_{10}^2 - m_1^2 + m_3^2 & c_{32} &= q_{20}^2 - q_{21}^2 - m_2^2 + m_3^2 \end{aligned} \tag{A.5}$$

where q_{10}, q_{20} and q_{21} are the parallel components of the external momenta q_1 and q_2

$$\begin{aligned} q_1 &= (q_{10}, 0, \vec{0}), \\ q_2 &= (q_{20}, q_{21}, \vec{0}). \end{aligned} \tag{A.6}$$

The coefficient $I_R(k)$ is used to treat the IR-divergences. If the integral C_0 is IR-divergent then one of the three terms under the first summation sign in Eq. (A.3) vanishes. There are three possible configurations of masses and external momenta in which C_0 is IR-divergent. Those are:

- The case $m_1^2 = 0$, $q_1^2 = m_3^2$ and $q_2^2 = m_2^2$. In this case, the first term ($k = 1$) under the first summation sign in Eq. (A.3) vanishes, thus we can exclude this term by choosing

$$I_R(1) = 0, \qquad I_R(2) = 1, \qquad I_R(3) = 1. \tag{A.7}$$

- The case $m_2^2 = 0$, $q_2^2 = m_3^2$ and $(q_2 - q_1)^2 = m_2^2$. Similar to the above case, the values of $I_R(k)$ are

$$I_R(1) = 1, \qquad I_R(2) = 0, \qquad I_R(3) = 1. \tag{A.8}$$

- The last case is $m_3^2 = 0$, $q_2^2 = m_2^2$ and $q_1^2 = m_1^2$ with

$$I_R(1) = 1, \qquad I_R(2) = 1, \qquad I_R(3) = 0. \tag{A.9}$$

Appendix B

The $\mathcal{R}$-function

In this section we present the definition and notation of $\mathcal{R}$-functions as far as needed in this thesis. The notation follows Carlson's book [55], where the reader can find the complete theory of $\mathcal{R}$-functions. For useful and special formulae which are used in one-loop calculations, we refer the reader to the theses of D. Kreimer [11] and L. Brücher [13].

Dirichlet measure:
Let $b \in \mathbb{C}^k_>$, $k \geq 2$, and let $E = E_{k-1}$ be the standard simplex in $\mathbb{R}^{k-1}$. The complex measure μ_b defined on E by

$$d\mu_b(u) = \frac{1}{\mathcal{B}(b)} u_1^{b_1-1} \dots u_{k-1}^{b_{k-1}-1} (1 - u_1 - \dots - u_{k-1})^{b_k-1} du_1 \dots du_{k-1} \tag{B.1}$$

will be called a Dirichlet measure. $\mathbb{C}_> = \{z \in \mathbb{C} : z \neq 0, |\mathrm{ph}\, z| < \pi/2\}$ is the open right complex half-plane.

Definition:
Let μ_b be a Dirichlet measure on the standard simplex $E \subset \mathbb{R}^{k-1}$, $k \geq 2$. Let H be a half-plane in $\mathbb{C} - \{0\}$. Let $\Omega = H$ if $t \in \mathbb{C} - \mathbb{N}$, but if $t \in \mathbb{N}$, let $\Omega = \mathbb{C}$. For every $z \in \Omega^k$ define

$$R_t(b, z) = \int_E (u \cdot z)^t d\mu_b(u). \tag{B.2}$$

If $k = 1$, define $R_t(b, z) = z^t$.

The $\mathcal{R}$-function can be defined by the following fundamental integral representation

$$\int_x^\infty (t-x)^{c+\alpha-1} \prod_{i=1}^{k} (z_i + w_i t)^{-b_i} dt = \mathcal{B}(-t, c+t) \prod_{i=1}^{k} w_i^{-b_i} \mathcal{R}_t(b, x + zw^{-1}) \tag{B.3}$$

where zw^{-1} is the k-tuple $(z_1 w_1^{-1} \dots z_k w_k^{-1})$.

For the calculation in chapter 4, the following relation is useful

$$\begin{aligned} \int_x^y (t-x)^{a-1} (y-t)^{a'-1} \prod_{i=1}^{k} (z_i + w_i t)^{-b_i} dt \quad &= \\ &= \mathcal{B}(a, a') (y-x)^{a+a'-1} \prod_{i=1}^{k} (z_i + w_i x)^{-b_i} \mathcal{R}_a \left(b, \frac{z+wy}{z+wx} \right). \end{aligned} \tag{B.4}$$

The hypergeometrical function ${}_2F_1$ can be expressed with the help of specific $\mathcal{R}$-function

$${}_2F_1(\alpha, \beta; \gamma, x) = \mathcal{R}_{-\alpha} \left(\gamma - \beta, \beta; 1, 1 - x\gamma \right). \tag{B.5}$$

When performing a series expansion of a $\mathcal{R}$-function at the pole $\epsilon = 0$, the following abbreviations are used in XLOOPS-GiNaC:

$$\begin{aligned}
\texttt{R2ex1}(x,y) &= \sqrt{1-\frac{x}{y}}\,\left[\ln\left(1-\sqrt{1-\frac{x}{y}}\right)-\ln\left(1+\sqrt{1-\frac{x}{y}}\right)+i\pi\right] \\
&\quad -\ln(-x)-i\pi\ .
\end{aligned} \tag{B.6}$$

$$\begin{aligned}
\texttt{R2ex2}(x,y) &= \left(1+\sqrt{1-\frac{x}{y}}\right)\,Li_2\left(1-\frac{1-\sqrt{1-\frac{x}{y}}}{1+\sqrt{1-\frac{x}{y}}}\right) \\
&\quad +\left(1+\sqrt{1-\frac{x}{y}}\right)\,\left[\ln\left(1-\sqrt{1-\frac{x}{y}}\right)\right]^2 \\
&\quad +\left(1-\sqrt{1-\frac{x}{y}}\right)\,Li_2\left(1-\frac{1+\sqrt{1-\frac{x}{y}}}{1-\sqrt{1-\frac{x}{y}}}\right) \\
&\quad +\left(1-\sqrt{1-\frac{x}{y}}\right)\,\left[\ln\left(1+\sqrt{1-\frac{x}{y}}\right)\right]^2 \\
&\quad +\frac{1}{2}\,(\ln y)^2+2\,(\ln x)^2-2\,\ln x\,\ln y \\
&\quad +(\ln y-2\,\ln x)\,\left[\left(1+\sqrt{1-\frac{x}{y}}\right)\,\ln\left(1-\sqrt{1-\frac{x}{y}}\right)\right. \\
&\quad \left.+\left(1-\sqrt{1-\frac{x}{y}}\right)\,\ln\left(1+\sqrt{1-\frac{x}{y}}\right)\right] \\
&\quad -i\pi\,\frac{\sqrt{x-y}}{\sqrt{-y}}\,\left[2\,\ln 2+\ln(x-y)\right]\ .
\end{aligned} \tag{B.7}$$

$$\begin{aligned}
\texttt{R3ex2}(x,y,z) &= 2\,\ln\left(1-\frac{x}{z}\right)\,\eta(x,z)+2\,\ln\left(1-\frac{y}{z}\right)\,\eta(y,z) \\
&\quad +2\,Li_2\left(1-\frac{x}{z}\right)+2\,Li_2\left(1-\frac{y}{z}\right)+2\,(\ln z)^2\,.
\end{aligned} \tag{B.8}$$

with

$$\eta(a,b)=2\pi i\,[\theta(-\mathrm{Im}\ a)\,\theta(-\mathrm{Im}\ b)\,\theta(\mathrm{Im}(ab))-\theta(\mathrm{Im}\ a)\,\theta(\mathrm{Im}\ b)\,\theta(-\mathrm{Im}(ab))]\,. \tag{B.9}$$

Appendix C

The residue theorem and numerical integrals of two-loop diagrams

In this chapter, the explicit calculation of the integrals $\widehat{J}_1$ and $\widehat{J}_2$ introduced in section 4.2.7 is presented. For $\widehat{J}_0$, the calculation is given in detail in [11].

C.1 The integral $\widehat{J}_1$

We first rewrite the integral explicitly in terms of parallel and orthogonal momentum variables

$$\begin{aligned}\widehat{J}_1 &= \int_0^\infty dl_\perp dk_\perp \int_{-1}^{1} dz \, \frac{8\pi^2 \; l_\perp^2 \; k_\perp^2}{[(l+q)^2 - m_1^2 + i\rho][l^2 - m_2^2 + i\rho][(l+k)^2 - m_3^2 + i\rho][k^2 - m_4^2 + i\rho]} \\ &= 8\pi^2 \int_0^\infty dl_\perp \; dk_\perp \; \frac{l_\perp^2 \; k_\perp^2}{[(l+q)^2 - m_1^2 + i\rho][l^2 - m_2^2 + i\rho][k^2 - m_4^2 + i\rho]} \int_{-1}^{1} dz \; \frac{1}{[a+bz]}\end{aligned}$$

where

$$(l+k)^2 - m_3^2 + i\rho \;=\; a - bz$$

and

$$\begin{aligned} a &= (l_0+k_0)^2 - l_\perp^2 - k_\perp^2 - m_3^2 + i\rho, \\ b &= 2 l_\perp k_\perp. \end{aligned}$$

Performing the z-integral one obtains

$$\begin{aligned} \widehat{J}_1 &= 8\pi^2 \int_0^\infty dl_\perp\, dk_\perp\, \frac{l_\perp^2\, k_\perp^2}{[(l+q)^2 - m_1^2 + i\rho][l^2 - m_2^2 + i\rho][k^2 - m_4^2 + i\rho]} \\ &\qquad \times \frac{\ln(a+b) - \ln(a-b)}{b} \\ &= 4\pi^2 \int_0^\infty dl_\perp\, dk_\perp\, \frac{l_\perp\, k_\perp\, [\,\ln(a+b) - \ln(a-b)\,]}{[(l+q)^2 - m_1^2 + i\rho][l^2 - m_2^2 + i\rho][k^2 - m_4^2 + i\rho]}. \end{aligned}$$

Now, we rewrite

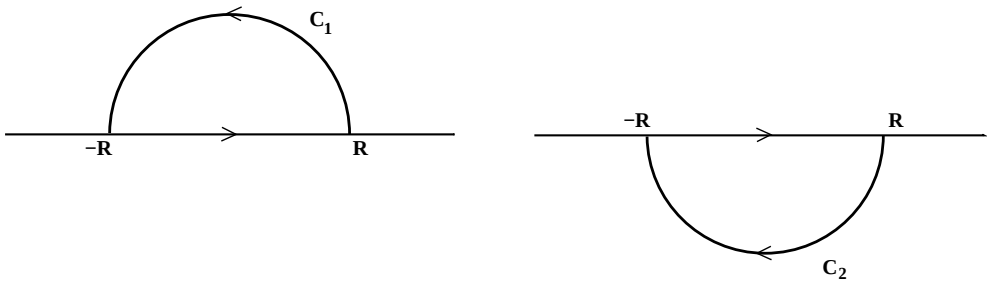
$$\begin{aligned} a+b &= w_3^2 - (l_\perp - k_\perp)^2, \\ a-b &= w_3^2 - (l_\perp + k_\perp)^2, \\ (l+q)^2 - m_1^2 + i\rho &= w_1^2 - l_\perp^2, \\ l^2 - m_2^2 + i\rho &= w_2^2 - l_\perp^2, \\ k^2 - m_4^2 + i\rho &= w_4^2 - k_\perp^2, \end{aligned}$$

with

$$\begin{aligned} w_1^2 &= (l_0+q_0)^2 - m_1^2 + i\rho, \\ w_2^2 &= l_0^2 - m_2^2 + i\rho, \\ w_3^2 &= (l_0+k_0)^2 - m_3^2 + i\rho, \\ w_4^2 &= k_0^2 - m_4^2 + i\rho. \end{aligned}$$

Since the integral is even under the transformation $l_\perp \leftrightarrow -l_\perp, k_\perp \leftrightarrow -k_\perp$, one can change the limit of the $l_\perp$- and $k_\perp$-integrals and obtains

$$\begin{aligned} \widehat{J}_1 &= -\pi^2 \int_{-\infty}^\infty dl_\perp\, dk_\perp\, \frac{l_\perp\, k_\perp}{(l_\perp^2 - w_1^2)\,(l_\perp^2 - w_2^2)\,(k_\perp^2 - w_4^2)} \;\times \\ &\qquad [\,\ln(w_3 - l_\perp + k_\perp) + \ln(w_3 + l_\perp - k_\perp) - \ln(w_3 - l_\perp - k_\perp) - \ln(w_3 + l_\perp + k_\perp)\,]\,. \end{aligned} \tag{C.1}$$



(a) (b)

Figure C.1: The contour for the $k_\perp$-integral

Here, we used the relations

$$\begin{aligned}\ln(w_3^2-(l_\perp+k_\perp)^2) &= \ln(w_3-l_\perp-k_\perp)+\ln(w_3+l_\perp+k_\perp),\\ \ln(w_3^2-(l_\perp-k_\perp)^2) &= \ln(w_3-l_\perp+k_\perp)+\ln(w_3+l_\perp-k_\perp).\end{aligned}$$

This can be done because $\arg(w_3+l_\perp+k_\perp)+\arg(w_3-l_\perp-k_\perp)<\pi$ and $\arg(w_3+l_\perp-k_\perp)+\arg(w_3-l_\perp+k_\perp)<\pi$ for real $l_\perp$ and $k_\perp$.

Now, we apply the residue theorem to calculate the $k_\perp$-integral. For the first and the fourth terms, we close the contour in the upper half-plane (see Fig. C.1-a). For the second and the third terms, we close the contour in the lower half-plane (see Fig. C.1-b). In contrast to the case of $\widehat{J}_0$, the integrals over the half circles C_1 and C_2 do not vanish. However, we will prove that the contour integrals cancel each other and thus we are still able to apply the residue theorem.

Considering the $k_\perp$-integral of the first term

$$\begin{aligned} I_1 &= \int\limits_{-\infty}^{\infty} dk_\perp \, \frac{k_\perp \ln(w_3 - l_\perp + k_\perp)}{(k_\perp - w_4)(k_\perp + w_4)} \qquad \text{(C.2)} \\ &= \lim_{R\to\infty} \left\{ \oint\limits_{C_1} dk_\perp \, \frac{k_\perp \ln(w_3 - l_\perp + k_\perp)}{(k_\perp - w_4)(k_\perp + w_4)} - \int\limits_{C_1} dk_\perp \, \frac{k_\perp \ln(w_3 - l_\perp + k_\perp)}{(k_\perp - w_4)(k_\perp + w_4)} \right\} \\ &= i\pi \ln(w_3 - l_\perp + w_4) - \lim_{R\to\infty} \int\limits_{C_1} dk_\perp \, \frac{k_\perp \ln(w_3 - l_\perp + k_\perp)}{(k_\perp - w_4)(k_\perp + w_4)}. \end{aligned}$$

Similarly one obtains

$$\begin{aligned} I_4 &= -\int\limits_{-\infty}^{\infty} dk_\perp \, \frac{k_\perp \ln(w_3 + l_\perp + k_\perp)}{(k_\perp - w_4)(k_\perp + w_4)} \qquad \text{(C.3)} \\ &= -i\pi \ln(w_3 + l_\perp + w_4) + \lim_{R\to\infty} \int\limits_{C_1} dk_\perp \, \frac{k_\perp \ln(w_3 + l_\perp + k_\perp)}{(k_\perp - w_4)(k_\perp + w_4)}. \end{aligned}$$

Applying the transformation

$$\int\limits_{C_1} f(z)dz = \lim_{R\to\infty} \int\limits_0^\pi f(Re^{i\theta}) \, i\, Re^{i\theta} d\theta, \qquad \text{(C.4)}$$

we obtain for the sum of the contour integrals I_1 and I_4

$$\begin{aligned} I_1 + I_4 &= i\pi \left[\ln(w_3 - l_\perp + w_4) - \ln(w_3 + l_\perp + w_4) \right] \qquad \text{(C.5)} \\ &\quad + \lim_{R\to\infty} \int\limits_0^\pi \frac{R^2 e^{2i\theta} \left[\ln(Re^{i\theta} + w_3 + l_\perp) - \ln(Re^{i\theta} + w_3 - l_\perp) \right]}{(Re^{i\theta} - w_4)(Re^{i\theta} + w_4)} \, d\theta. \end{aligned}$$

The second integral vanishes so that one obtains

$$I_1 + I_4 = i\pi \left[\ln(w_3 - l_\perp + w_4) - \ln(w_3 + l_\perp + w_4) \right]. \qquad \text{(C.6)}$$

Similarly, the integrals in the lower half-plane of the second and the third terms result in

$$I_2 + I_3 = i\pi \left[\ln(w_3 - l_\perp + w_4) - \ln(w_3 + l_\perp + w_4) \right] \qquad \text{(C.7)}$$

so that

$$\widehat{J}_1 = 2i\pi^3 \int_{-\infty}^{\infty} dl_\perp \frac{l_\perp \, [\, \ln(w_3 + l_\perp + w_4) - \ln(w_3 - l_\perp + w_4) \,]}{(l_\perp + w_1)(l_\perp - w_1)(l_\perp + w_2)(l_\perp - w_2)}. \tag{C.8}$$

Because the integrand decreases rapidly for $l_\perp \to \infty$ we can apply the residue theorem to the $l_\perp$-integral and obtain

$$\widehat{J}_1 = \frac{4\pi^4}{w_1^2 - w_2^2} \, [\, \ln(w_2 + w_3 + w_4) - \ln(w_1 + w_3 + w_4) \,]. \tag{C.9}$$

C.2 The integral $\widehat{J}_2$

The integral $\widehat{J}_2$ can be solved in the same manner as for $\widehat{J}_1$.

$$\widehat{J}_2 \, = \, 8\pi^2 \int_0^{\infty} dl_\perp dk_\perp \int_{-1}^{1} dz \, \frac{l_\perp^2 \, k_\perp^2}{[(l+q)^2 - m_1^2 + i\rho][(l+k)^2 - m_2^2 + i\rho][k^2 - m_3^2 + i\rho]}. \tag{C.10}$$

After performing the z- and $k_\perp$-integrals in the same manner as for $\widehat{J}_1$ one obtains

$$\begin{aligned} \widehat{J}_2 \;&=\; 2\,i\,\pi^3 \int_{-\infty}^{\infty} dl_\perp \, \frac{l_\perp \, [\, \ln(w_2 - l_\perp + w_3) - \ln(w_2 + l_\perp + w_3) \,]}{(l_\perp^2 - w_1^2)} \\ &=\; 2\,i\,\pi^3 \left[\int_{-\infty}^{\infty} dl_\perp \, \frac{\ln(w_2 + w_3 - l_\perp)}{(l_\perp - w_1)} - \int_{-\infty}^{\infty} dl_\perp \, \frac{\ln(w_2 + w_3 + l_\perp)}{(l_\perp - w_1)} \right] \end{aligned} \tag{C.11}$$

with

$$\begin{aligned} w_1^2 \;&=\; (l_0 + q_0)^2 - m_1^2 + i\rho, \\ w_2^2 \;&=\; (l_0 + k_0)^2 - m_2^2 + i\rho, \\ w_3^2 \;&=\; k_0^2 - m_3^2 + i\rho. \end{aligned} \tag{C.12}$$

Now we close the contour in the lower half-plane for the first integral, and in the upper half-plane for the second one and obtain

$$\begin{aligned} \widehat{J}_2 &= -4\pi^4 \ln(w_1+w_2+w_3) \\ &- \lim_{R\to\infty} \int_0^\pi d\theta \left[\frac{\ln(w_2+w_3-Re^{-i\theta})}{(Re^{-i\theta}-w_1)} - \frac{\ln(w_2+w_3+Re^{i\theta})}{(Re^{i\theta}-w_1)} \right]. \end{aligned} \tag{C.13}$$

The second term vanishes in the limit of $R \to \infty$ so we obtain

$$\widehat{J}_2 = -4\pi^4 \ln(w_1+w_2+w_3). \tag{C.14}$$

Appendix D

XLOOPS-GiNaC installation

The computer program XLOOPS-GiNaC can be downloaded from
`http://wwwthep.physik.uni-mainz.de\~xloops`
To be installed, XLOOPS-GiNaC requires the GiNaC library for symbolic computation in C/C++, which can be obtained from
`http://www.ginac.de`
The required minimum version one has to install is specified in the file `INSTALL` that comes with the XLOOPS-GiNaC package. For two-loop integrals, the *nestedsums* library is required. The *nestedsums* library can be downloaded from
`http://www.fis.unipr.it/~stefanw/download/nestedsums-1.1.0.tar.gz`

Note: Due to the technical issue of programming, the module for two-loop integrals is not ready for end users at the moment. Only developers of XLOOPS-GiNaC can download the latest version from the CVS.

XLOOPS-GiNaC uses GNU `autoconf`, so the installation is as easy as this:

```
$ ./configure
$ make CXXFLAGS = "-O1"
[ Become root if users want to install the program in the system directory]
# make install
# make check CXXFLAGS = "-O1"
```

The `configure` script accepts a number of options to enable and disable various features. For a complete list, type:

```
$ ./configure --help
```

The most important option is `--prefix=PREFIX` which allows one to install the package in a directory different from the default `/usr/local`. Also, the `CXXFLAGS` flag, which is used for compilation optimization should not be set higher than level 1. This inconvenience comes from the fact that XLOOPS-GiNaC contains some long expressions. If the `CXXFLAGS` flag is set higher than 1, the C/C++ compilers (such as `GNU gcc`) will require a lot of resources to optimize the binary code of the XLOOPS-GiNaC library. This inconvenience should be fixed in future version.

Bibliography

[1] T. Ishikawa, T. Kanedo, K. Kato, S. Kawabata, Y. Shimizu, H. Tanaka, KEK preprint KEK-92-19.

[2] E.E. Boos, M.N. Dubinin, V.A. Ilin, A.E. Pukhov, V.I. Savrin, Moscow State U. preprint SNUTP-94-116 [hep-ph/9503280].

[3] L. Brücher, J. Franzkowski, D. Kreimer, Comput. Phys. Commun. **85** (1995) 153; Comput. Phys. Commun. **107** (1997) 281;
L. Brücher, J. Franzkowski, A. Frink, D. Kreimer, Nucl. Instrum. Meth. **A389** (1997) 323.

[4] J. Küblbeck, M. Böhm, A. Denner, Comp. Phys. Commun. **60** (1990) 165.

[5] R. Mertig, M. Böhm, A. Denner, Comp. Phys. Commun. **64** (1991) 345.

[6] G.J. van Oldenborgh, Rep. No. NIKHEP-H/90-15, Amsterdam, 1990.

[7] T. Hahn, Nucl. Phys. Proc. Suppl. **89** (2000) 231 [hep-ph/0005029].

[8] G. Weiglein, R. Scharf, M. Böhm, Nucl. Phys. **B 416** (1994) 606.

[9] S. Heinemeyer, W. Hollik, G. Weiglein, Comput. Phys. Commun. **124** (2000) 76.

[10] G. Weiglein, hep-ph/0109237.

[11] D. Kreimer, Dissertation, Universität Mainz 1992.

[12] D. Kreimer, Mod. Phys. Lett. **A9** (1994) 1105.

[13] L. Brücher, Dissertation, Universität Mainz 1997.

[14] J. Franzkowski, Dissertation, Universität Mainz 1997.

[15] A. Frink, Dissertation, Universität Mainz 2000.

[16] C. Bauer, H.S. Do, Comput. Phys. Commun, **144** (2002) 154 [hep-ph/0102231].

[17] B.W. Char et al., *Maple V language reference manual*, Springer, New York, 1991.

[18] H.S. Do, J.G. Körner, S. Wiermann, MZ-TH/99-69, internal report.

[19] C. Bauer, A. Frink, R. Kreckel, arXiv: cs.SC/0004015, to appear in J. Symb. Comput.

[20] American National Standards Institute, ISO/IEC 14882-1998(E), *Programming Languages — C/C++*, 1998.

[21] J. Collins, *Renormalization*, Cambridge Univ. Press, 1984.

[22] C. Wetterich, Phys. Rev. **D 64** (2001) [hep-ph/0008150].

[23] A. Frink, J.G. Körner, J.B. Tausk, hep-ph/9709490.

[24] G. 't Hooft, M. Veltman, Nucl. Phys. **B 153** (1979) 365.

[25] G. Passarino, M. Veltman, Nucl. Phys. **B 160** (1979) 151.

[26] G. Weiglein, R. Scharf, M. Bohm, Nucl. Phys. **B 416** (1994) 606.

[27] S. Bauberger, G. Weiglein, hep-ph/9611445.

[28] O.V. Tarasov, hep-ph/9703319.

[29] F.A. Berends, A.I. Davydychev, N.I. Ussyukina, Phys. Lett. **B 426** (1998) 95;
A.I. Davydychev, J.B. Tausk, Nucl. Phys. **B 397** (1993) 123.

[30] S. Groote, A.A. Pivovarov, Nucl. Phys. **B 580** (2000) 459.

[31] A. Ghinculov and York-Peng Yao, Nucl. Phys. **B 516** (1998) 385 [hep-ph/9702266].

[32] D. Kreimer, Z. Phys. **C 54** (1992) 667.

[33] H.S. Do, S. Groote, J.G. Körner and M.C. Mauser, to be publ. in Phys. Rev. **D** [hep-ph/0209185].

[34] A. Denner and T. Sack, Nucl. Phys. **B358** (1991) 46.

[35] S. Weinberg, *The Quantum Theory of Fields* Cambridge University Press, (1996).

[36] D. Bailin, A. Love, *Introduction to Gauge Field Theory*, IOP Publishing, ISBN 0-7503-0281-X, 1993.

[37] S. Bauberger, F.A. Berends, M. Böhm, M. Buza, Nucl. Phys. **B 434** (1995) 383 [hep-ph/9409388].

[38] S. Weinzierl, Comp. Phys. Comm. **145** (2002) 357 [math-ph/0201011].

[39] N.N. Bogoliubov and O. Parasiuk, *Acta Math.* **97** (1957) 227;
K. Hepp, Comm. Math. Phys. **2** (1966) 301;
W. Zimmerman, Comm. Math. Phys. **15** (1969) 208.

[40] A.I. Davydychev, V.A. Smirnov, J.B. Tausk, Nucl. Phys. **B 410** (1993) 325 [hep-ph/9307371].

[41] A.I. Davydychev, J.B. Tausk, Nucl. Phys. **B 397** (1993) 123.

[42] F.A. Berends, M. Böhm, M. Buza, R. Scharf, Z. Phys. **C 63** (1994) 227 .

[43] M. Caffo, H. Czyż and Remiddi, hep-ph/0203256.

[44] G.P. Lepage, CLNS-80/447 (1980);
G.P. Lepage, J. Comp. Phys. 27 (1978) 192;
R. Kreckel, MZ-TH/97-30.

[45] L. Cucos and E. de Doncker *Distributed QMC Algorithms: New Strategies and Performance Evaluation*, Proceedings of the High Performance Computing Symposium 2002 (HPC'02), (2002);

R. Zanny, K. Kaugars, and E. de Doncker, Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'01), (2001), pp.674;
`http://www.cs.wmich.edu/parint`.

[46] F.A. Berends, J.B. Tausk, Nucl. Phys. **B 421** (1994) 456.

[47] J.B. Tausk, Leiden (1993).

[48] P.N. Maher, L. Durand, K. Riesselmann, Phys. Rev. **D 48** (1993) 1061.

[49] D.J. Broadhurst, Z. Phys. **C 47** (1990) 115.

[50] W.D. Gropp and E. Lusk, Mathematics and Computer Science Division, Argonne National Laboratory, 1996. ANL-96/6.

[51] `http://www.ginac.de`

[52] M.E. Peskin, D.V. Schroeder, *An Introduction to Quantum Field Theory*, Addison-Wesley, 1995.

[53] A. Davydychev, P. Osland and L. Saks , Phys. Rev. **D 63** [hep-ph/0008171].

[54] G. t'Hooft, M. Veltman, Nucl. Phys. **B 153** (1979) 365;
G. Passarino, M. Veltman, Nucl. Phys. **B 160** (1979) 151;
K. Aoki, Z. Hioki, R. Kawabe, M. Konuma, T. Muta, Suppl. Prog. Theor. Phys. **73** (1982);
M. Böhm, W. Hollik, H. Spiesberger, Fortschr. Phys. **34** (1986) 11;
A. Denner, Fortschr. Phys. **41** (1993) 4;

[55] B.C. Carlson, *Special functions of applied mathematics*, Academic press, Inc. 1977.