MasterTwo: A Package for the Calculation of Two Loop Diagrams

S. Schilling [a,b,1],

[a] *Institute for Theoretical Physics, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland*

[b] *Institute of Molecular Systems Biology, ETH Zurich Wolfgang-Pauli-Str. 16 8093 Zurich, Switzerland*

Abstract

The package MasterTwo allows the calculation of one and two-loop B-decays like $b \to s\,\gamma$ and $b \to sl^+l^-$ in the Standard Model (SM) and extensions of the SM. It can be easily adapted to other types of integrals which can be reduced to scalar integrals independent of external momenta and depending on up to two different masses. It consists of two subpackages, Fermions and Integrals. Fermions covers the standard Dirac Algebra, Integrals the Taylor expansion, partial fraction, tensor reduction and the integration of the thus achieved scalar integrals.

Key words: Scalar two loop integration, heavy mass expansion, recurrence relations, tensor reduction, Dirac algebra

---

Package summary

Manuscript Title: MasterTwo*: A Package for the Calculation of Two Loop Diagrams*
Authors: S.Schilling, P.Liniger
Package Title: MasterTwo
Version: 1.0
Journal Reference:
Catalogue identifier:
Licensing provisions: None
Programming language: MATHEMATICA
Computer: Computers running MATHEMATICA
Operating system: Linux, MacOs, Windows
RAM: Depending on the complexity of the problems
Keywords: Scalar two loop integration, heavy mass expansion, recurrence relations, tensor reduction, Dirac algebra

[1] Corresponding author: schilling.sabine@gmail.com

PACS: 13.25.Hw, 14.80.Cp, 02.70.Wz
Classification: Computer Algebra
Nature of the physical problem: One- and two-loop integrals reducable to scalar integrals independent of external momenta and dependent on up to two different masses.
Solution method: Heavy Mass Expansion and recurrence relations to transform tensor integrals to a larger number of scalar master integrals. Loop integration of the thus obtained scalar master integrals.
Running time: Strongly depending on the problem and nature of diagram being calculated

Contents

1. Introduction

The calculation of loop decays is a tedious work, which can rarely be performed in a reasonable time by calculating all arising diagrams by hand. The package MasterTwo was thus originally designed to automate the calculation of one and two-loop B-decays like $b \to s\ \gamma$ and $b \to sl^+l^-$ in the SM and Two-Higgs-Dublet models. It can be easily adapted to other types of integrals which are reducible to scalar integrals that depend on up to two different masses and independent of external momenta. It enables the user to perform the basic steps of the one and two-loop calculations fully automatically. In contrast to other programmes like 'Reduce' and 'Form' it works completely inside MATHEMATICA. Compared to other programmes like 'HIP', 'Tracer' and 'FeynArts' MasterTwo is much smaller. The package is therefore easy to use and to understand and may be extended and customised quickly.

2. The Package Structure of MasterTwo

MasterTwo consists of two subpackages, Fermions and Integrals:

– Fermions: This package originally written by Patrick Liniger [1] and extended by Kay Bieri contains all routines regarding the Dirac Algebra. A detailed description of its features is given in section 3, a detailed list of all available commands is given in section 5 . Typing the command `FermionsInfo[]` inside a MATHEMATICA session will furthermore list all available commands.

– Integrals: This package summarises all routines concerning the tensor reduction and partial fraction of one and two-loop integrals. Furthermore it allows the integrations of scalar integrals with up to two different masses not depending on external momenta. The theoretical background of this package is given in section 4, the documentation of its routines in section 6.

3. Fermions

Fermions can simplify Dirac expressions in D dimensions with an anticommuting γ_5. It provides the tools for standard operations like contracting indices, sorting expressions and the use of the Dirac equation. To calculate physical quantities as cross sections and decay-rates it allows furthermore to conjugate and square Dirac expressions and to compute traces over products of γ matrices (for details see section 5.3).

3.1. *Declarations and Constants*

Before usage of the package, all arising masses, momenta, indices and polarisation vectors must first be declared. The documentation of the corresponding commands is given in 5.1.

There are a few constants predefined in Fermions:

– `d` denotes the space-time dimension. [2]

– `eps` stands for ϵ.

– `L` and `R` are the left- and the right-projectors, respectively: `L` $= 1/2\ (1 - \gamma_5)$ and `R` $= 1/2\ (1 + \gamma_5)$.

– `Gamma5` stands for γ_5.

– `Unit` denotes the unit matrix.

– `Sigma[mu,nu]` is the tensor $\sigma_{\mu\nu} = i/2[\gamma_\mu, \gamma_\nu]$.

The symbols `L`, `R` and `Gamma5` are treated as projectors and, provided the expression is simple enough, are shifted to the left automatically in order to reduce the number of different terms. An expression like $\gamma_\mu L + R\gamma_\mu$ will therefore automatically be transformed into $2R\gamma_\mu$.

[2] The **capital** letter `D`, which usually denotes space-time dimensions, is already used inside MATHEMATICA to indicate partial derivates. However, for reasons of better readability, D will be used in all formulae of this manual to indicate the space time dimensions.

3.2. *Notation and Syntax*

After all the necessary declarations have been established, the corresponding symbols can be used inside Dirac expressions and alike.

Gamma matrices, tensors and projectors like γ_μ, $\sigma_{\mu\nu}$, `L`, `R` and `L` and R matrices are given as expressions with the head `Dirac`, whereas scalar products are input as the function `Scal`. A few examples of simple structures:

$g_{\mu\nu}$	`Scal[mu,nu]`,
p_μ	`Scal[p,mu]`,
$p \cdot q$	`Scal[p,q]`,
$\mathbf{1}$	`Dirac[]` (unit matrix in Dirac space),
γ_μ	`Dirac[mu]`,
$\not{p}$	`Dirac[p]`,
γ_5	`Dirac[Gamma5]` and similar for L and R,
$\sigma_{\mu\nu}$	`Dirac[Sigma[mu,nu]]`.

Some more complicated structures involving products of γ matrices and four-vectors might read:

$L\gamma_\mu\gamma_\nu\not{p}$	`Dirac[L, mu, nu, p]`,
$p_\mu\gamma^\mu$	`Scal[p, mu] Dirac[mu]`,
$\gamma_\mu(\not{p}+m_b)\not{q}$	`Dirac[mu, p + mb, q]`,
$R(m_b\gamma_\mu+p_\mu)\gamma_\nu$	`Dirac[R, mb mu + Unit Scal[p, mu], nu]`.

Note that masses inside Dirac structures need not to be provided with an extra `Unit` matrix, whereas this is indispensable for other structures like scalar products.

Fermions makes no difference between covariant (up) and contravariant (down) indices. It simply assumes that - if the same index appears twice - one is upper and the other lower and, if requested, takes the sum over them.

3.3. *Dirac Algebra and Naive Dimensional Regularisation*

The D-dimensional metric tensor g is introduced satisfying

$$g_{\mu\nu}g^{\nu\mu} = g^\mu_\mu = D, \tag{1}$$

where $D = 4-2\epsilon$ in all kind of expressions containing Lorentz indices. The Dirac gamma matrices $\gamma^\mu = (\gamma^0, \gamma^i)$, where the Latin index i is employed to denote spatial indices 1,2,3, satisfy the anticommutation relations

$$\{\gamma^\mu, \gamma^\nu\} = 2g^{\mu\nu} = 2g_{\mu\nu}. \tag{2}$$

The γ_5 is defined by

$$\gamma_5 = \gamma^5 = i\gamma^0\gamma^1\gamma^2\gamma^3 \tag{3}$$

and anti-commutes with all γ^μ:

$$\{\gamma^5, \gamma^\mu\} = 0. \tag{4}$$

It has been emphasised in the literature that this rule leads to algebraic inconsistencies [2,3]. Indeed, the naive dimensional regularisation (NDR) is inconsistent with

$$Tr(\gamma^\mu\gamma^\nu\gamma^\rho\gamma^\sigma\gamma_5) \neq 0 \tag{5}$$

for dimensions of space-time $D = 4 - 2\epsilon$, $\epsilon \neq 0$. However the latter condition is often considered to be necessary for an acceptable regularisation, since at $D = 4$ we must find

$$Tr(\gamma^\mu\gamma^\nu\gamma^\rho\gamma^\sigma\gamma_5) = 4\mathrm{i}\varepsilon^{\mu\nu\rho\sigma} \ . \tag{6}$$

Provided one can avoid the calculation of traces like eq. (6) containing γ_5 matrices, it has been demonstrated in many explicit calculations [4] that the NDR gives correct results consistent with schemes without the γ_5 problem. From eq. (3) we get for the projectors $R = (1 + \gamma^5)/2$ and $L = (1 - \gamma^5)/2$

$$\gamma^0 R = L\gamma^0,\ \gamma^0 L = R\gamma^0. \tag{7}$$

The package does not need an explicit representation of the algebra, it can thus handle objects of the form $\gamma_\mu\gamma_\nu$ rather than e.g. $\gamma_0\gamma_2$. The function `DiracAlgebra` performs the standard Dirac algebra according to eqs. (1), (2), (4) and (7). Further functions of FERMIONS (conjugations, traces) are documented in section 5.

4. Integrals

INTEGRALS performs all the steps necessary to transform the integrals into scalar master integrals of up to two different masses and independent of external momenta and their subsequent loop integration. A full list of all the available commands is given by the command `IntegralsInfo[]`. A detailed documentation of all the functions introduced below is given in section 6.

4.0.1. *Additional Declarations*

Some functions of INTEGRALS require the distinction between small and heavy masses or loop momenta and external momenta. Thus for the proper function of these functions additional declarations have to be made. Details can be found in section 6.1.

4.0.2. *Representatin of Propagators*

The propagator structure of one-loop integrals like

$$\frac{1}{(q_1^2 - m_1^2)^{n_1}} \tag{8}$$

is written in the programme as

$$\mathtt{AD}[\underbrace{\mathtt{den}[\mathtt{q}_1, \mathtt{m}_1], \ldots, \mathtt{den}[\mathtt{q}_1, \mathtt{m}_1]}_{\mathrm{n}_1 \text{ times}}]. \tag{9}$$

In analogy, the propagator structure of two-loop integrals

$$\frac{1}{(q_1^2 - m_1^2)^{n_1}(q_2^2 - m_2^2)^{n_2}((q_1 + q_2)^2 - m_3^2)^{n_3}} \tag{10}$$

is written as

$$\texttt{AD}[\underbrace{\texttt{den}[\texttt{q}_1,\texttt{m}_1],\ldots,\texttt{den}[\texttt{q}_1,\texttt{m}_1]}_{\texttt{n}_1 \text{ times}},\underbrace{\texttt{den}[\texttt{q}_2,\texttt{m}_2],\ldots,\texttt{den}[\texttt{q}_2,\texttt{m}_2]}_{\texttt{n}_2 \text{ times}},\underbrace{\texttt{den}[\texttt{q}_1+\texttt{q}_2,\texttt{m}_3],\ldots,\texttt{den}[\texttt{q}_1+\texttt{q}_2,\texttt{m}_3]}_{\texttt{n}_3 \text{ times}}]. \tag{11}$$

4.1. *Colour Algebra*

Integrals with outgoing gluons or quarks can lead to a quite complicated colour structures. The following relations can be derived in the fundamental representation of $SU(N)$ [5]:

$$f^{bac}\mathbf{T}^c\mathbf{T}^b = \frac{1}{2}\, i\, N\mathbf{T}^a, \tag{12}$$

$$\begin{aligned}\mathbf{T}^c\mathbf{T}^d f^{dba} f^{acb} &= \mathbf{T}^c\mathbf{T}^d N\delta^{dc} \\ &= \frac{N^2-1}{2N}N = \frac{N^2-1}{2},\end{aligned} \tag{13}$$

$$\mathbf{T}^a\mathbf{T}^e f^{adc} f^{dek} f^{ckb} = \frac{N}{2}\mathbf{T}^a\mathbf{T}^e f^{abe} = -i\frac{N^2}{4}\mathbf{T}^b. \tag{14}$$

The function `Color` applies eqs. (12-14) for the special case $N = 3$. Note that structure constants f^{abc} are represented in the programme as `SUNF[a,b,c]`, whereas products of generators $\mathbf{T}^a\mathbf{T}^b$ are represented as `SUNT[a,b]`.

4.2. *From Tensor Integrals to Scalar Integrals*

Integrals was originally designed to facilitate the calculation of Wilson coefficients of mass dimension six operators of effective Hamiltonians of the the rare decays $b \to s\,\gamma$ and $b \to sl^+l^-$ in the SM and Two-Higgs-Dublet models. In the corresponding integrals two heavy mass scales arise: the top-mass and the W-mass (SM) or the charged Higgs mass (THDM). A typical propagator structure is given by

$$I = \frac{1}{(q_1^2 - m_1^2)^{n_1}(q_1^2 - m_2^2)^{n_2}((q_2+k_1)^2 - m_2^2)^{n_3}((q_1+q_2+k_2) - m_2^2)^{n_4}}, \tag{15}$$

where q_1 and q_2 are the loop momenta, k_1 and k_1 the external momenta, $n_j \geq 0$ and $\sum_j n_j = 6$. The exact calculation of two-loop graphs with two mass scales is technically very demanding. Thus at the moment exact results for diagrams with more than one mass scale do not exist beyond one-loop. Therefore the Heavy Mass Expansion (HME) [6], an asymptotic expansion in small momenta and masses, is used.

4.2.1. *Heavy Mass Expansion*

The basic idea of the Heavy Mass Expansion (HME) is to use the hierarchy of mass scales and momenta to reduce complicated two-loop calculations to simpler ones. The following assumptions are made:

(i) All the masses of a given Feynman diagram Γ can be divided into a set of large $\underline{M} = \{M_1, M_2, \ldots\}$ and small $\underline{m} = \{m_1, m_2, \ldots\}$ masses.

(ii) All external momenta $\underline{k} = \{k_1, k_2, \ldots\}$ are small compared to the scale of the large masses $\underline{M}$.

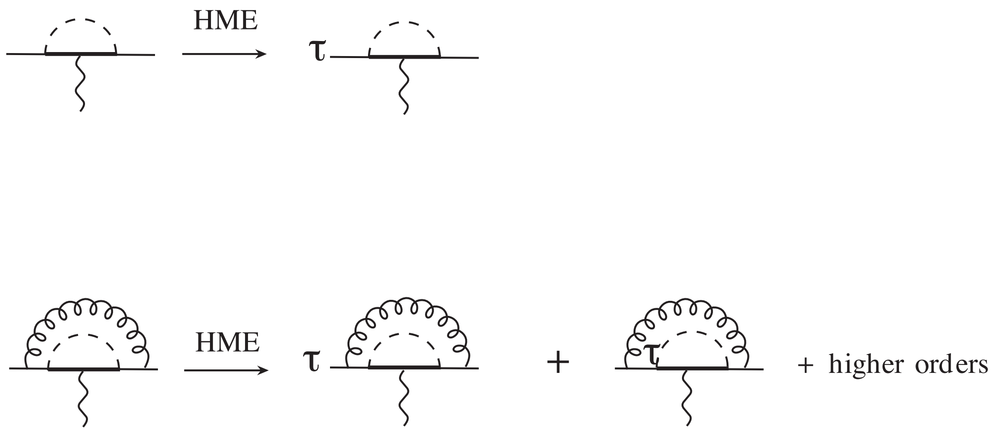

Fig. 1. Expansion of the full theory in the HME for the example process $b \to s\,\gamma$. τ symbolises the Taylor expansion in small masses and momenta as described in eq. (16). Thick lines stand for heavy quarks (in this example the top mass), dashed lines heavy bosons like the $W^{\pm}, \pi^{\pm}$ in the SM or the charged Higgs in the THDM. In line two we show the two subdiagrams needed to be evaluated in the HME.

The ansatz is that the dimensionally regularised (unrenormalised) Feynman integral F_Γ associated with the Feynman diagram Γ can be written as

$$F_\Gamma \overset{M\to\infty}{\sim} \sum_\gamma F_{\Gamma/\gamma} \circ \mathcal{T}_{\underline{k}^\gamma, \underline{m}^\gamma} F_\gamma(\underline{k}^\gamma, \underline{m}^\gamma, \underline{M}), \tag{16}$$

where the sum is performed over all subgraphs γ of Γ which fulfil the following two conditions simultaneously:

– γ contains all lines with heavy masses ($\underline{M}$),
– γ consists of connected [3] components that are one-particle-irreducible with respect to the lines with small masses ($\underline{m}$).

The operator $\mathcal{T}$ performs a Taylor expansion in the variables k_i^2/M_j^2 and m_l^2/M_j^2, where k_i belongs to $\underline{k}^\gamma$, the set of external momenta with respect to the subgraph γ. m_l belongs to the set of light masses $\underline{m}^\gamma$ of γ. M_j is the heavy mass of the propagator to which the light mass or the external momenta belong to.

After Taylor expansion of the two-loop integrals we need to deal with the calculation of a large number of rather simple integrals. Matching these results with effective low energy theories we find out to which mass power we must expand the Taylor series in the HME. In order to calculate Wilson coefficients of the rare b-decays $b \to s\,\gamma$ and $b \to sl^+l^-$ up to $\mathcal{O}(\alpha_s)$-precision we have to match to an effective theory with operators of mass dimension six. Therefore it is sufficient to expand the integrands up to second order in external momenta and small masses. Expansion up to higher order in the external momenta would correspond to Wilson coefficients of operators of higher mass dimensions and can therefore be safely neglected. The Taylor expansion of the Feynman integrands in external momenta, as well as setting all the light masses to zero, creates spurious infrared divergences which can be regularised dimensionally. All these divergences cancel out in the matching conditions relating the full and the effective theory Green functions.

[3] A graph is called connected when it can not be separated into two or more distinct pieces without cutting any line.

Taylor Expansion

The expansion in external momenta is performed by the function `TaylorExpansion`. It performs the expansion of each propagator in external momenta up to $\mathcal{O}[(\text{external momenta})^2/M^2]$

$$\frac{1}{(q_i+k)^2-M^2} = \frac{1}{q_i^2-M^2}\left[1-\frac{k^2+2kq_i}{q_i^2-M^2}+\frac{4(kq_i)^2}{(q_i^2-M^2)^2}\right]+\mathcal{O}[k^4/M^4], \tag{17}$$

$$\begin{aligned}\frac{1}{(q_1+q_2+k)^2-M^2} =& \frac{1}{(q_1+q_2)^2-M^2}\\ &\left[1-\frac{k^2+2kq_1+2kq_2}{(q_1+q_2)^2-M^2}+\frac{4(kq_1)^2+4(kq_2)^2+8q_1kq_2}{((q_1+q_2)^2-M^2)^2}\right]\\ &+\mathcal{O}[k^4/M^4],\end{aligned} \tag{18}$$

where q_i $(i=1,2)$ are the loop momenta, M is a heavy mass and k an arbitrary external momentum.
The expansion in small masses up to second order

$$\frac{1}{q_i^2-m^2} = \frac{1}{q_i^2}\left[1+\frac{m^2}{q^2}\right]+\mathcal{O}[m^4/q^4], \tag{19}$$

where m is a small mass, is performed by the function `TaylorMass`. The function expands automatically in all masses not declared as heavy masses with `DeclareHeavyMass`.

Scaling

The routine `Scaling` multiplies all light masses and external momenta with a factor x and sets all terms x^n with $n > 2$ to zero. This is justified in the calculation of Wilson coefficients corresponding to operators of mass dimension six. Keeping terms with $n > 2$ would correspond to the calculation of contributions to Wilson coefficients of higher mass dimensions.

4.2.2. *Partial Fraction Decomposition and Simplification of Numerators*

The routine `PartialFractionOne` (one-loop case) and `PartialFractionTwo` (two-loop case) allows a reduction of all the integrals to those in which a single mass parameter occurs in the propagator denominators together with a given loop momentum by applying the following partial fraction decomposition

$$\frac{1}{(q^2-m_1^2)(q^2-m_2^2)} = \frac{1}{m_1^2-m_2^2}\left[\frac{1}{q^2-m_1^2}-\frac{1}{q^2-m^2}\right], \tag{20}$$

$$\frac{q^2}{(q^2-m_1^2)(q^2-m_2^2)} = \frac{1}{m_1^2-m_2^2}\left[\frac{m_1^2}{q^2-m_1^2}-\frac{m_2^2}{q^2-m_2^2}\right]. \tag{21}$$

Furthermore the routines perform the following relations to get successively rid of loop momenta in the numerator:

$$
\begin{aligned}
\frac{(q_i^2)^n}{q_i^2 - m^2} =& (q_i^2)^{n-1} + \frac{(q_i^2)^{n-1} m^2}{q_i^2 - m^2} \ (i = 1, 2), \qquad (22)\\
\frac{(q_1 q_2)^n}{(q_1^2 - m_1^2)(q_2^2 - m_2^2)((q_1 + q_2)^2 - m_3^2)} =& \frac{1}{2}(q_1 q_2)^{n-1} \left[\frac{1}{(q_1^2 - m_1^2)(q_2^2 - m_2^2)} \right.\\
&- \frac{1}{(q_2^2 - m_2^2)((q_1 + q_2)^2 - m_3^2)}\\
&- \frac{1}{(q_1^2 - m_2^2)((q_1 + q_2)^2 - m_3^2)}\\
&\left. + \frac{m_3^2 - m_1^2 - m_2^2}{(q_1^2 - m_1^2)(q_2^2 - m_2^2)((q_1 + q_2)^2 - m_3^2)} \right] \qquad (23)
\end{aligned}
$$

with $n \geq 1$. In a last step all vanishing massless integrals are set to zero [7]:

$$
\int d^D q \frac{1}{q^{2\alpha}} = 0. \qquad (24)
$$

4.2.3. *Tensor Reduction*

The idea of the tensor reduction is to express tensor integrals in terms of scalar integrals. As integrals over an antisymmetric integrand with symmetric integration boundaries are zero, all integrands with an odd number of loop momenta $q_i^\alpha, (i = 1, 2)$ in the nominator can be set to zero before performing the proper tensor reduction. The basic relations for the tensor reduction of one-loop integrals are given by

$$
\int d^D q \, q^{\alpha_1} q^{\alpha_2} A(q^2) = \frac{1}{D} \int d^D q^2 g^{\alpha_1 \alpha_2} A(q^2), \qquad (25)
$$

$$
\begin{aligned}
\int d^D q \, q^{\alpha_1} q^{\alpha_2} q^{\alpha_3} q^{\alpha_4} A(q^2) =& \frac{1}{D^2 + 2D}\\
&\int d^D q^4 (g^{\alpha_1 \alpha_2} g^{\alpha_3 \alpha_4} + g^{\alpha_1 \alpha_3} g^{\alpha_2 \alpha_4} + g^{\alpha_1 \alpha_4} g^{\alpha_2 \alpha_3}) A(q^2), \qquad (26)
\end{aligned}
$$

$$
\int d^D q \, q^{\alpha_1} q^{\alpha_2} \ldots q^{\alpha_{2k}} A(q^2) = \frac{\Gamma(2 - \epsilon)}{2^k \Gamma(2 - \epsilon + k)} \int d^D q^{2k} \mathbf{X}^{(\mathbf{k})} \, A(q^2), \qquad (27)
$$

where $A(q^2)$ is an arbitrary scalar function depending on Lorentz invariants of the loop momentum q and masses. Usually it is a product of powers of propagators

$$
\frac{1}{(q^2 - m^2)^{n_1}} \qquad (28)
$$

times a polynomial of q^2. $\mathbf{X}^{(\mathbf{k})}$ stands for permutations of metric tensor components $g^{\alpha_j \alpha_k}$. The routine `TensorOne` performs the tensor reduction of one-loop integrals for up to nine Lorentz indices. Results are Taylor expanded in ϵ up to second order. The one-loop relations eqs. (25-27) can be generalised to the case of two-loop integrals [8,9]

$$\int d^Dq_1d^Dq_2\ \ q_1^{\alpha_1}q_2^{\alpha_2}A(q_1,q_2) = \frac{1}{D}\int d^Dq_1d^Dq_2\ A(q_1,q_2)\ (q_1\cdot q_2)g^{\alpha_1\alpha_2}, \tag{29}$$

$$\begin{aligned}\int d^Dq_1d^Dq_2\ \ A(q_1,q_2)q_1^{\alpha_1}q_1^{\alpha_2}q_1^{\alpha_3}q_2^{\alpha_4} =& \frac{1}{D^2+2D}\int d^Dq_1d^Dq_2\ \ A(q_1,q_2)q_1^2\ (q_1\cdot q_2)\\ &(g^{\alpha_1\alpha_2}g^{\alpha_3\alpha_4}+g^{\alpha_1\alpha_3}g^{\alpha_2\alpha_4}+g^{\alpha_1\alpha_4}g^{\alpha_2\alpha_3})\end{aligned} \tag{30}$$

$$\begin{aligned}\int d^Dq_1d^Dq_2\ A(q_1,q_2)q_1^{\alpha}q_1^{\beta}q_2^{\gamma}q_2^{\delta} =& \frac{1}{D^3+D^2-2D}\int d^Dq_1d^Dq_1\ A(q_1,q_2)\\ &\big[\big((1+D)q_1^2q_2^2-2(q_1\cdot q_2)^2\big)\,g^{\alpha_1\alpha_2}g^{\alpha_3\alpha_4}\\ &+\big(-q_1^2q_2^2+D(q_1\cdot q_2)^2\big)\,g^{\alpha_1\alpha_3}g^{\alpha_2\alpha_4}\\ &+g^{\alpha_1\alpha_4}g^{\alpha_2\alpha_3})\big],\end{aligned} \tag{31}$$

where $A(q_1,q_2)$ is an arbitrary scalar function of q_1 and q_2 and arbitrary masses. It is usually a product of powers of propagators

$$\frac{1}{(q_1^2-m_1^2)^{n_1}(q_2^2-m_2^2)^{n_2}((q_1+q_2)^2-m_3^2)^{n_3}} \tag{32}$$

times a polynomial in q_1^2, q_1^2, q_1q_2, but the concrete form of this function has no importance for the tensor reduction. The function `TensorTwo` performs the two-dimensional tensor reduction for up to four Lorentz indices. In the case of factorising integrals corresponding to $c=0$ in the integrand of eq. (32), `TensorTwo` performs a one-dimensional tensor reduction calling `TensorOne`. From the tensor reduction we obtain additional terms of q_1^2, q_2^2 or q_1q_2 in the numerator. This makes a subsequent usage of the identities `PartialFractionOne` and `PartialFractionTwo`, described in section 4.2.2, necessary.

4.2.4. *Substitutions*

The function `Substitutions` makes substitution in the integrands of factorising two-loop-integrals such that the propagator structure contains no overlapping loop momenta by applying the following relations

$$\int d^Dq_1d^Dq_2\frac{S(q_1,q_2)}{(q_1^2-m_1^2)^{n_1}((q_1+q_2)^2-m_2^2)^{n_3}} = \int d^Dq_1d^Dq_2\frac{S(q_1,q_2-q_1)}{(q_1^2-m_1^2)^{n_1}(q_2^2-m_2^2)^{n_3}}, \tag{33}$$

$$\int d^Dq_1d^Dq_2\frac{S(q_1,q_2)}{(q_2^2-m_1^2)^{n_2}((q_1+q_2)^2-m_2^2)^{n_3}} = \int d^Dq_1d^Dq_2\frac{S(q_1-q_2,q_2)}{(q_2^2-m_1^2)^{n_1}(q_1^2-m_2^2)^{n_3}}, \tag{34}$$

where $S(q_1,q_2)$ is a polynomial in q_1^2, q_2^2 and $q_1\cdot q_2$. A subsequent final partial fraction of the so obtained integrands leads then to the desired scalar integrals.

4.2.5. *Transforming the Propagator Structure*

The routine `SimplifyPropagator` transforms the propagator structure of a scalar loop integrals to the forms needed for the loop integration functions. Thus the propagator structure of non-factorising scalar two-loop integrals eq. (11) is transformed to the short form

$$\mathtt{G}[\mathtt{i}[\mathtt{m}_1,\mathtt{n}_1],\mathtt{i}[\mathtt{m}_2,\mathtt{n}_2],\mathtt{i}[\mathtt{m}_3,\mathtt{n}_3]]. \tag{35}$$

The routine replaces the propagator structure of factorising two-lop integrals

$$\mathtt{AD}[\underbrace{\mathtt{den}[\mathtt{q_1},\mathtt{m_1}],....,\mathtt{den}[\mathtt{q_1},\mathtt{m_1}]}_{\mathrm{n_1\ times}},\underbrace{\mathtt{den}[\mathtt{q_2},\mathtt{m_2}],....,\mathtt{den}[\mathtt{q_2},\mathtt{m_2}]}_{\mathrm{n_2\ times}}] \tag{36}$$

with

$$\mathtt{AD}[\mathtt{i}[\mathtt{m_1},\mathtt{n_1}],\mathtt{i}[\mathtt{m_2},\mathtt{n_2}]] \tag{37}$$

and the propagator structure of one-loop integrals

$$\frac{1}{(q_1^2-m_1^2)^{n_1}} \tag{38}$$

by

$$\mathtt{AD}[\mathtt{i}[\mathtt{m_1},\mathtt{n_1}]]. \tag{39}$$

`SimplifyPropagator` orders furthermore scalar two-loop integrals with one vanishing mass in the denominator in such a way that the propagator denominator with overlapping loop momenta has always no additional mass term ($m_3 = 0$):

$$\begin{aligned} &\int d^Dq_1d^Dq_2\frac{1}{(q_1^2)^{n_3}(q_2^2-m_2^2)^{n_2}((q_1+q_2)^2-m_1^2)^{n_1}} \\ =&\int d^Dq_1d^Dq_2\frac{1}{(q_1^2-m_2^2)^{n_2}(q_2^2)^{n_3}((q_1+q_2)^2-m_1^2)^{n_1}} \\ =&\int d^Dq_1d^Dq_2\frac{1}{(q_1^2-m_1^2)^{n_1}(q_2^2-m_2^2)^{n_2}((q_1+q_2)^2)^{n_3}}. \end{aligned} \tag{40}$$

The last line of eq. (40) is the ordering of propagator denominators needed for the following loop integration routines. This ordering is necessary, as the two-loop integrals are represented in the programme as a non-commuting list.

4.3. *Loop Integration of Scalar One Loop Integrals*

After Taylor expansion, tensor reduction and subsequent partial fractions the one-loop tensor integrals are transformed to a bigger number of scalar integrals proportional to [10]:

$$\begin{aligned} \mu^{2\epsilon}\int\frac{d^D\,q}{(2\pi)^{-2\epsilon}}\frac{1}{(q^2-m^2)^n} &= \frac{\mu^{2\epsilon}}{(2\pi)^{-2\epsilon}}\frac{\pi^{D/2}\,\Gamma(1+\epsilon)}{(m^2)^{n-D/2}}\,C_n^{(1)} \\ &= \frac{\pi^2}{(m^2)^{n-2}}\,\left(\left(\frac{\mu^2}{m^2}\right)^\epsilon\,2^{2\epsilon}\pi^\epsilon\,\Gamma(1+\epsilon)\right)\,C_n^{(1)} \\ &= \frac{\pi^2}{(m^2)^{n-2}}N_\epsilon^{(1)}(m)\,C_n^{(1)}, \end{aligned} \tag{41}$$

where for arbitrary n and m [11]:

$$N_\epsilon^{(1)}(m) = \left(\frac{\mu^2}{m^2}\right)^\epsilon 2^{2\epsilon}\pi^\epsilon\,\Gamma(1+\epsilon)$$

$$= 1 - \epsilon\kappa + \epsilon^2\left(\frac{1}{12}\pi^2 + \frac{1}{2}\kappa(m)^2\right) + \mathcal{O}(\epsilon)^3, \tag{42}$$

$$\kappa(m) = \gamma_E - \ln(4\pi) + \ln\frac{m^2}{\mu^2}, \tag{43}$$

$$C_n^{(1)} = i\frac{(-1)^n}{(n-1)!}(1+\epsilon)_{n-3}, \tag{44}$$

which vanishes for $n \leq 0$. In eq. (44) we introduced the Pochhammer symbol

$$(a)_k = \frac{\Gamma(a+k)}{\Gamma(a)} = \begin{cases} a(a+1)(a+2)...(a+k-1), & k \geq 1, \\ 1, & k = 0, \\ 1/[(a-1)(a-2)...(a-|k|)], & k \leq -1 \end{cases} \tag{45}$$

for integer k and complex a. The prefactors of $C_n^{(1)}$ are chosen such that $C_n^{(1)}$ is free of common factors of the one-loop integration. The factor $N_\epsilon^{(1)}(m_1)$ summarises the ϵ-dependent part of the common prefactors. The function `ScalIntOne` performs the scalar one-loop integration by replacing the propagator structure `AD[i[m,n]]` by the right hand side of eq. (41):

$$\mathtt{AD[i[m,n]]} \rightarrow \frac{\pi^2}{(m^2)^{n-2}}\mathtt{Ne[m]}\ \mathtt{C}_\mathtt{n}^{(1)}, \tag{46}$$

where `Ne[m]` corresponds to eq. (42) and $\mathtt{C}_\mathtt{n}^{(1)}$ to eq. (44) up to second order in `eps`. The final result is expanded in `eps` up to first order.

4.4. *Loop Integration of Scalar Two Loop Integrals*

4.4.1. *Recurrence Relations*

In this section we will show how to reduce scalar two loop integrals independent of external momenta to master integrals, where the highest power of all appearing propagators is one. These master integrals can then be automatically integrated.

We will first give a general derivation of the recurrence relations for arbitrary masses m_1, m_2 and m_3 for the integral

$$G^{m_1,m_2,m_3}_{n_1,n_2,n_3} \equiv \frac{\mu^{4\epsilon}}{(2\pi)^{-4\epsilon}}\int d^Dq_1 d^Dq_2 \frac{1}{(q_1^2-m_1^2)^{n_1}(q_2^2-m_2^2)^{n_2}((q_1+q_2)^2-m_3^2)^{n_3}}. \tag{47}$$

Its propagator structure is written in the package as $\mathtt{G[i[m_1,n_1],i[m_2,n_2],i[m_3,n_3]]}$. The derivation of the recurrence relations starts with the following identities [12]:

$$\int d^Dq_1 d^Dq_2 \frac{\partial}{\partial q_1^\mu}\left(\frac{q_1^\mu}{(q_1^2-m_1^2)^{n_1}(q_2^2-m_2^2)^{n_2}((q_1+q_2)^2-m_3^2)^{n_3}}\right) = 0, \tag{48}$$

$$\int d^Dq_1 d^Dq_2 \frac{\partial}{\partial q_2^\mu}\left(\frac{q_2^\mu}{(q_1^2-m_1^2)^{n_1}(q_2^2-m_2^2)^{n_2}((q_1+q_2)^2)-m_3^2)^{n_3}}\right) = 0, \tag{49}$$

$$\int d^Dq_1 d^Dq_2 \frac{\partial}{\partial q_1^\mu}\left(\frac{q_1^\mu}{(q_1^2-m_1^2)^{n_1}((q_1+q_2)^2-m_2^2)^{n_2}(q_2^2-m_3^2)^{n_3}}\right) = 0. \tag{50}$$

Substitutions of the integration variables in eq. (48) lead to eqs. ((49)-(50)). With the help of the the Gaussian integral theorem we can transform the integral to a vanishing surface integral with symmetric boundaries and an asymmetric integrand. In oder to simplify the notation we will use

$$G^{m_1,m_2,m_3}_{n_1,n_2,n_3} \equiv G_{n_1,n_2,n_3} \tag{51}$$

in this section. From eq. (48) we get

$$\begin{aligned}(D-2n_1-n_3)G_{n_1,n_2,n_3} =& 2n_1 m_1^2 G_{n_1+1,n_2,n_3} + n_3(G_{n_1-1,n_2,n_3+1} - G_{n_1,n_2-1,n_3+1}) \\ &+ n_3(m_1^2 - m_2^2 + m_3^2)G_{n_1,n_2,n_3+1}.\end{aligned} \tag{52}$$

From eq. (49) or directly by replacing $n_1 \leftrightarrow n_2$ and $m_1 \leftrightarrow m_2$ in eq. (52) we get

$$\begin{aligned}(D-2n_2-n_3)G_{n_1,n_2,n_3} =& 2n_2 m_2^2 G_{n_1,n_2+1,n_3} + n_3(G_{n_1,n_2-1,n_3+1} - G_{n_1-1,n_2,n_3+1}) \\ &+ n_3(m_2^2 - m_1^2 + m_3^2)G_{n_1,n_2,n_3+1}.\end{aligned} \tag{53}$$

From eq. (50) we obtain

$$\begin{aligned}(D-2n_1-n_2)G_{n_1,n_2,n_3} =& 2n_1 m_1^2 G_{n_1+1,n_2,n_3} + n_2(G_{n_1-1,n_2+1,n_3} - G_{n_1,n_2+1,n_3-1}) \\ &+ n_2(m_1^2 + m_2^2 - m_3^2)G_{n_1,n_2+1,n_3}.\end{aligned} \tag{54}$$

The last three equations connect integrals with the sum of powers $n_1 + n_2 + n_3$ with integrals where the sum of the powers is lowered by 1. They form an equation system, which can be used to extract the integrals
G_{n_1+1,n_2,n_3}, G_{n_1,n_2+1,n_3} G_{n_1,n_2,n_3+1}. Solving it we obtain the following recurrence relations [12]:

$$\begin{aligned}G_{n_1+1,n_2,n_3} =& \frac{1}{n_1\ m_1^2\ \Delta(m_1,m_2,m_3)} \\ & \big\{\big[n_2\ (m_1^2 - m_3^2)(m_1^2 - m_2^2 + m_3^2) + n_3\ (m_1^2 - m_2^2)(m_1^2 + m_2^2 - m_3^2) \\ & +D\ m_1^2\ (-m_1^2 + m_2^2 + m_3^2) - n_1\ \Delta\ (m_1,m_2,m_3)\big]\ G_{n_1,n_2,n_3} \\ & + n_2\ m_2^3(m_1^2 - m_2^2 + m_3^2)\ [G_{n_1,n_2+1,n_3-1} - G_{n_1-1,n_2+1,n_3}] \\ & + n_3\ m_3^2(m_1^2 + m_2^2 - m_3^2)[G_{n_1,n_2-1,n_3+1} - G_{n_1-1,n_2,n_3+1}]\big\}\end{aligned} \tag{55}$$

with the determinant of the corresponding equation system

$$\Delta(m_1,m_2,m_3) = 2(m_1^2m_2^2 + m_1^2m_3^2 + m_2^2m_3^2) - (m_1^4 + m_2^4 + m_3^4). \tag{56}$$

Replacing $n_1 \leftrightarrow n_2$ and $m_1 \leftrightarrow m_2$ in eq. (55) we get

$$\begin{aligned}G_{n_1,n_2+1,n_3} =& \frac{1}{n_2\ m_2^2\ \Delta(m_1,m_2,m_3)} \\ & \big\{\big[n_1\ (m_2^2 - m_3^2)(m_2^2 - m_1^2 + m_3^2) + n_3\ (m_2^2 - m_1^2)(m_1^2 + m_2^2 - m_3^2) \\ & +D\ m_2^2\ (-m_2^2 + m_1^2 + m_3^2) - n_2\ \Delta(m_1,m_2,m_3)\big]\ G_{n_1,n_2,n_3} \\ & + n_1\ m_1^2(m_2^2 - m_1^2 + m_3^2)\ [G_{n_1+1,n_2,n_3-1} - G_{n_1+1,n_2-1,n_3}] \\ & + n_3\ m_3^2(m_1^2 + m_2^2 - m_3^2)\ [G_{n_1-1,n_2,n_3+1} - G_{n_1,n_2-1,n_3+1}]\big\}.\end{aligned} \tag{57}$$

In analogy, we get by replacing $n_1 \leftrightarrow n_3$ and $m_1 \leftrightarrow m_3$ in eq. (55)

$$
\begin{aligned}
G_{n_1,n_2,n_3+1} =&\frac{1}{n_3\ m_3^2\ \Delta(m_1,m_2,m_3)} \\
&\big\{\big[n_1(m_3^2-m_2)^2)(m_2^2-m_1^2+m_3^2)+n_2(m_3^2-m_1^2)(m_1^2+m_3^2-m_2^2) \\
&+D\ m_3^2(-m_3^2+m_1^2+m_2^2)-n_3\ \Delta(m_1,m_2,m_3)\big]\ G_{n_1,n_2,n_3} \\
&+n_1\ m_1^2(m_2^2-m_1^2+m_3^2)\,[G_{n_1+1,n_2-1,n_3}-G_{n_1+1,n_2,n_3-1}] \\
&+n_2\ m_2^2(m_1^2+m_3^2-m_2^2)\,[G_{n_1-1,n_2+1,n_3}-G_{n_1,n_2+1,n_3-1}]\big\}. \qquad (58)
\end{aligned}
$$

The general recurrence relations eqs (55), (57) and (58) are implemented in the rule `recurrence` which is part of the integration routine `ScalIntTwoThreeMasses`.

Recurrence Relations for Scalar Integrals with One Massless Propagator

In the following we consider the special case, that one of the masses in eq. (55) vanishes. Without restrictions we can choose this mass to be m_3. Taking the limit $m_3 \to 0$ we get from eqs. (55) and (57)

$$
\begin{aligned}
G^{m_1,m_2,0}_{n_1+1,n_2,n_3} &= \frac{1}{m_1^2\ n_1(1-x)}\ \big\{[D-n_1-n_2-n_3+x(n_1-n_3)]G^{m_1,m_2,0}_{n_1n_2n_3} \\
&\qquad + xn_2\ \Big[G^{m_1,m_2,0}_{n_1-1,n_2+1,n_3}-G^{m_1,m_2,0}_{n_1,n_2+1,n_3-1}\Big]\Big\}, \qquad (59)
\end{aligned}
$$

$$
\begin{aligned}
G^{m_1,m_2,0}_{n_1,n_2+1,n_3} &= -\frac{1}{m_2^2 n_2 x(1-x)}\ \big\{[x(D-n_1-n_2-n_3)+n_2-n_3]G^{m_1,m_2,0}_{n_1n_2n_3} \\
&\qquad + n_1\ \Big[G^{m_1,m_2,0}_{n_1+1,n_2-1,n_3}\Big]\Big\}, \qquad (60)
\end{aligned}
$$

where $x = m_2^2/m_1^2$ [12]. From eq. (58) we see that the limit $m_3 \to 0$ does not exist for $G^0_{n_1,n_2,n_3+1}$. The recurrence relation for $G^0_{n_1,n_2,n_3+1}$ in this limit can be derived from eq. (55) by eliminating $G^0_{n_1+1,n_2,n_3}$ with the help of eq. (57) and $G^0_{n_1,n_2+1,n_3}$ with the help of eq. (58). Thus we obtain

$$
\begin{aligned}
G^{m_1,m_2,0}_{n_1,n_2,n_3+1} &= \frac{1}{m_1^2 n_3(1-x)^2}\ \big\{[(1+x)(-D)+2n_2+(1+3x)n_3]G^{m_1,m_2,0}_{n_1n_2n_3} \\
&\qquad + \quad 2xn_2\ \Big[G^{m_1,m_2,0}_{n_1,n_2+1,n_3-1}-G^{m_1,m_2,0}_{(n_1-1)(n_2+1)n_3}\Big] \\
&\qquad + (1-x)n_3\ \Big[G^{m_1,m_2,0}_{n_1(n_2-1)(n_3+1)}-G^{m_1,m_2,0}_{(n_1-1)n_2(n_3+1)}\Big]\Big\}. \qquad (61)
\end{aligned}
$$

Eqs. (59-61) are implemented in the rule `recurrenceb`. This rule is part of the integration routine `ScalIntTwo`.

4.4.2. *Loop Integration of Master Integrals*

In the last section we have shown how to reduce scalar two-loop integrals independent of external momenta to master integrals. This section will focus on the loop integration of special cases of these master integrals. We will focus on integrals with only two different masses ($m_1 = m_3$) and the case of one vanishing mass ($m_3 = 0$) in eq. (47).

Scalar Two loop Integrals with Two Different Masses

The routine `ScalIntTwoThreeMasses` can automatically perform the integration of scalar two-loop integrals of the type of eq. (47) for the special case $m_1 = m_3$. In a first step the integrands are ordered by making the following substitutions

$$\begin{aligned}
&\int \frac{d^D q_1 d^D q_2}{(2\pi)^{-4\epsilon}} \frac{1}{(q_1^2-m_1^2)^{n_1}(q_2^2-m_1^2)^{n_2}((q_1-q_2)^2-m_2^2)^{n_3}} \\
&= \int \frac{d^D q_1 d^D q_2}{(2\pi)^{-4\epsilon}} \frac{1}{(q_1^2-m_2^2)^{n_3}(q_2^2-m_1^2)^{n_1}((q_1+q_2)^2-m_1^2)^{n_2}} \\
&= \int \frac{d^D q_1 d^D q_2}{(2\pi)^{-4\epsilon}} \frac{1}{(q_1^2-m_1^2)^{n_1}(q_2^2-m_2^2)^{n_3}((q_1+q_2)^2-m_1^2)^{n_2}}\,, \qquad (62)
\end{aligned}$$

where the order of the integrands given in the last line of eq. (62) is the order needed for the following loop integration.

With the help of the recurrence relations eqs. (55), (57) and (58) we can reduce integrals of the form eq. (62) to the following master integral:

$$\begin{aligned}
G_{1\ 1\ 1}^{m_1\ m_2\ m_1} &= \frac{\mu^{4\epsilon}}{(2\pi)^{-4\epsilon}} \int d^D q_1 d^D q_2 \frac{1}{(q_1^2-m_1^2)^{n_1}(q_2^2-m_2^2)^{n_2}((q_1+q_2)^2-m_1^2)^{n_3}} \\
&= \pi^4\, m_1^2\, N_\epsilon^{(2)}(m_1)\, C_{1\ 1\ 1}^{m_1 m_2 m_1\ (2)}, \qquad (63)
\end{aligned}$$

where $N_\epsilon^{(2)}(m_1)$ collects all ϵ-dependent parts of the common prefactors of the two-loop integrals. It is given by

$$\begin{aligned}
N_\epsilon^{(2)}(m_1) = (N_\epsilon^{(1)}(m_1))^2 &= \left(\frac{\mu^2}{m_1^2}\right)^{2\epsilon} 2^{4\epsilon}\pi^{2\epsilon}\,\Gamma(1+\epsilon)^2 \\
&= 1 - 2\epsilon\kappa(m_1) + \epsilon^2\left(\frac{1}{6}\pi^2 + 2\kappa(m_1)^2\right) + \mathcal{O}(\epsilon)^3. \qquad (64)
\end{aligned}$$

and $C_{1\ 1\ 1}^{m_1 m_2 m_1\ (2)}$ by [12]

$$\begin{aligned}
C_{1\ 1\ 1}^{m_1 m_2 m_1\ (2)} &= \frac{1}{(1-\epsilon)(1-2\epsilon)}\left[-\frac{1}{\epsilon^2}\left(1+\frac{x}{2}\right)\right. \\
&\left. + \frac{1}{\epsilon}\left(x\,\log(x)\right) - \frac{1}{2}\left(x\log(x)^2\right) + \left(2-\frac{x}{2}\right)\,\phi(x)\right]. \qquad (65)
\end{aligned}$$

The function $\phi(x)$ depends on the mass relation between m_1 and m_2.

– If

$$0 < x = \frac{m_2^2}{m_1^2} < 1, \qquad (66)$$

then $\phi(x)$ is given by

$$\phi(x) = 4\sqrt{\frac{x}{4-x}}\,\mathrm{Cl}_2\,\left(2\arcsin\left(\frac{\sqrt{\mathrm{x}}}{2}\right)\right), \qquad (67)$$

where Cl_2 is Clausen's integral function [13]

$$\mathrm{Cl}_2(\theta) = S_2(\theta) = \Im[\mathrm{Li}_2(\mathrm{e}^{\mathrm{i}\theta})] = -\int_0^\theta \mathrm{dt}\ln\left|2\sin\left(\frac{\mathrm{t}}{2}\right)\right|$$

$$. \qquad (68)$$

– If $x > 1$ then

$$\phi(x) = \frac{1}{\lambda(x)} \left[-4\mathrm{Li}_2\left(\frac{1-\lambda(\mathrm{x})}{2}\right) + 2\ln^2\left(\frac{1-\lambda(\mathrm{x})}{2}\right) - \ln^2(\mathrm{x}) + \frac{\pi^2}{3}\right], \tag{69}$$

where

$$\lambda(x) = \sqrt{1 - \frac{4}{x}}. \tag{70}$$

Scalar Two Loop Integrals with One Mass Scale

If in eq. (63) all masses are equal we obtain [12]

$$G_{1\ 1\ 1}^{m_1\ m_1\ m_1} = \pi^4\ m_1^2\, N_\epsilon^{(2)}(m_1)\, C_{1\ 1\ 1}^{m_1 m_1 m_1\ (2)}, \tag{71}$$

where

$$C_{1\ 1\ 1}^{m_1 m_1 m_1\ (2)} = \frac{1}{(1-\epsilon)(1-2\epsilon)}\left(-\frac{3}{2\epsilon^2} + 2\sqrt{3}\mathrm{Cl}_2\left(\frac{\pi}{3}\right),\right). \tag{72}$$

where $\mathrm{Cl}_2(\pi/3) = 1.0149417...$ is the maximum of Clausen's integral[12] [4] . The function `ScalIntTwoThreeMasses` applies the substitutions of eq. (62) and the recurrence relations eqs. (55), (57) and (58) to propagator structures of the form $\mathtt{G[i[m_1, n_1], i[m_2, n_2], i[m_1, n_3]]}$. This leads to numerous terms proportional to $\mathtt{G[i[m_1, 1], i[m_2, 1], i[m_1, 1]]}$, which can be replaced by the master integral (65):

$$\mathtt{G[i[m_1, 1], i[m_2, 1], i[m_1, 1]]} \to \pi^4\ m_1^2\, \mathtt{N2[m1]})\, C_{111}^{(2)}, \tag{73}$$

where `N2[m1]` correspond to eq. (64) up to second order in `eps` and $C_{111}^{(2)}$ to eq. (65) for $m_1 \neq m_2$ and to eq. (72) for $m_1 = m_2$. The final result is expanded up to zeroth order in `eps`.

Scalar Two Loop Integrals with One Massless Propagator

The function `ScalIntTwo` is able to perform the loop integration for integrals of type eq. (47), if one of the three masses in the propagators is zero. We can choose this to be m_3, as all other cases can be tranformed to this special case with the help of eq. (40) by the routine `SimplifyPropagator`. Then we get for the D-dimensional two-loop integral [10]

$$\begin{aligned} G_{n1,n2,n3}^{m_1,m_2,0} &= \frac{\mu^{4\epsilon}}{(2\pi)^{-4\epsilon}} \int \frac{d^D q_1\ d^D q_2}{(q_1^2 - m_1^2)^{n_1}(q_2^2 - m_2^2)^{n_2}[(q_1 - q_2)^2]^{n_3}} \\ &= \frac{\mu^{4\epsilon}\pi^D}{(2\pi)^{-4\epsilon}} \frac{\Gamma(1+\epsilon)^2}{(m_1^2)^{n_1+n_2+n_3-D}} C_{n_1 n_2 n_3}^{(2)} \\ &= \frac{\pi^4}{(m_1^2)^{n_1+n_2+n_3-4}} \left(\left(\frac{\mu^2}{m_1^2}\right)^{2\epsilon} 2^{4\epsilon}\pi^{2\epsilon}\ \Gamma(1+\epsilon)^2\right)\ C_{n_1 n_2 n_3}^{(2)} \\ &= \frac{\pi^4}{(m_1^2)^{n_1+n_2+n_3-4}} N_\epsilon^{(2)}(m_1)\ C_{n_1 n_2 n_3}^{(2)} \end{aligned} \tag{74}$$

[4] Clausen's integral is part of MATHEMATICA's special function package MathWorld'SpecialFunctions' which can be downloaded from http://library.wolfram.com/infocenter/MathSource/4775/. It is integated in the package Integrals.m, where `ClausenCl[n,x]` gives the Clausen function of order n.

with arbitrary integer powers n_1, n_2 and n_3 and with m_1 and $m_2 \neq 0$. All the two-loop integrals defined in eq. (74) vanish when either n_1 or n_2 is non-positive.

Performing the integration in eq. (74), we have to distinguish the following cases of non-vanishing integrals:

a) two of the masses are equal,
b) the second mass m_2 vanishes,
c) the masses m_1 and m_2 are different,
d) one of the powers n_i ($i = 1, 2, 3$) is zero (factorising two-loop integrals).

As the first three cases have the prefactor $N_\epsilon^{(2)}(m_1)/((m_1^2)^{n_1+n_2+n_3-4})$ in common, we will only display the corresponding values of $C^{(2)}_{n_1 n_2 n_3}$:

a) With the help of Feynman-parameterisation [10] we get for two equal masses $m_1 = m_2$ from eq. (74)

$$C^{(2)}_{n_1 n_2 n_3} = (-1)^{n_1+n_2+n_3+1} \frac{(2-\epsilon)_{-n_3}(1+\epsilon)_{n_1+n_3-3}(1+\epsilon)_{n_2+n_3-3}}{(n_1-1)!(n_2-1)!(n_1+n_2+n_3-4+2\epsilon)_{n_3}}. \tag{75}$$

b) If in eq. (74) the second mass m_2 vanishes, we again derive with the help of Feynman-parameterisation

$$\begin{aligned} C^{(2)}_{n_1 n_2 n_3} &= (-1)^{n_1+n_2+n_3+1} \\ &\quad \frac{(1+2\epsilon)_{n_1+n_2+n_3-5}(1+\epsilon)_{n_2+n_3-3}(1-\epsilon)_{1-n_2}(1-\epsilon)_{1-n_3}}{(n_1-1)!(n_2-1)!(n_3-1)!(1-\epsilon)(1-\frac{1}{3}\pi^2\epsilon^2+\mathcal{O}(\epsilon^3))}. \end{aligned} \tag{76}$$

c) If $m_1 \neq m_2$ and none of the two masses vanishes, the routine `ScalInt` reduces all integrals with three positive indices to a term proportional to the master integral $G^{m_1\ m_2\ 0}_{1\ \ 1\ \ 1}$ with the help of recurrence relations eqs. (59-61). The corresponding $C^{(2)}_{111}$ is given by

$$\begin{aligned} C^{(2)}_{111} =& \frac{1}{2(1-\epsilon)(1-2\epsilon)} \\ & \left[-\frac{1+x}{\epsilon^2} + \frac{2}{\epsilon} x \ln x + (1-2x)\ln^2 x + 2(1-x)\mathrm{Li}_2\left(1-\frac{1}{\mathrm{x}}\right) + \mathcal{O}(\epsilon) \right], \end{aligned} \tag{77}$$

where the dilogarithm Li_2 is given by [5]

$$\begin{aligned} \mathrm{Li}_2(\mathrm{x}) &= -\int_0^x \frac{\ln(1-t)}{t} dt = -\int_0^1 \frac{\ln(1-xt)}{t} dt \\ &= -\int_{1-x}^1 \frac{\ln(t)}{1-t} dt = \int_0^1 \frac{\ln(t)}{t-1/x} dt \end{aligned} \tag{78}$$

and x by eq. (66) [5]

[5] The definitions of eq. (78) correspond to the definitions used in MATHEMATICA [?]. Unfortunately this is not the case for the conventions used in MAPLE [15]. We have the following connections between both conventions

$$\mathrm{Li}_2^{\mathrm{MATHEMATICA}}(1-\mathrm{x}) = \mathrm{Li}_2^{\mathrm{MAPLE}}(\mathrm{x}).$$

d) When two indices are positive, but one of the n_i in eq. (74) equals zero, the two-loop integrals reduce to products of one-loop integrals. Without restriction we can choose $n_3 = 0$ and obtain from eq. (41)

$$G^{m_1,m_2,0}_{n1,n2,0} = \frac{\pi^4}{(m_1^2)^{n_1+n_2-4}} \, N_\epsilon^{(1)}(m_1) N_\epsilon^{(1)}(m_2) \, C^{(1)}_{n_1} C^{(1)}_{n_2}, \tag{79}$$

where

$$C^{(1)}_{n_1} C^{(1)}_{n_2} = -\frac{(-1)^{n_1}(-1)^{n_2}}{(n_1-1)!(n_2-1)!} (1+\epsilon)_{n_1-3}(1+\epsilon)_{n_2-3}. \tag{80}$$

If $m_1 \neq m_2$ we get

$$N_\epsilon^{(1)}(m_1) N_\epsilon^{(1)}(m_2) = N_\epsilon^2(m_1) \, \left(1 - \ln(x)\,\epsilon + \frac{1}{2}\,\ln^2(x)\,\epsilon^2\right) + \mathcal{O}(\epsilon^3) \tag{81}$$

with x defined in eq. (66). If both masses equal m_1, we simply have

$$(N_\epsilon^{(1)}(m_1))^2 = N_\epsilon^{(2)}(m_1) \tag{82}$$

defined in eq. (64); if both masses equal m_2 we derive

$$(N_\epsilon^{(1)}(m_2))^2 = N_\epsilon^{(2)}(m_1)\,(1 - 2\epsilon\log(x) + 2\epsilon^2\log(x)^2) \; + \mathcal{O}(\epsilon^3). \tag{83}$$

The routine `ScalIntTwo` performs the integration in all these cases:

a) $(m_1 = m_2)$ and b) $(m_2 = 0)$:
The propagator structures is replaced with the right hand side of eqs. (75)-(76) respectively:

$$\mathtt{G[i[m_1,n_1],i[m_2,n_2],i[0,n_3]} \to \frac{\pi^4}{(m_1^2)^{n_1+n_2+n_3-4}} \mathtt{N2[m_1]} C^{(2)}_{n_1 n_2 n_3}, \tag{84}$$

where `N2[m1]` corresponds to eq. (64) up to second order in `eps` and $C^{(2)}_{n_1 n_2 n_3}$ is given by eq. (75) in case a) and by eq. (76) in case b).

c) The integrals are first reduced to the masterintegral, which can then be automatically integrated:

$$\begin{aligned} \mathtt{G[i[m_1,n_1],i[m_2,n_2],i[0,n_3]]} &\to \mathrm{prefac}(\mathtt{n_1,n_2})\;\mathtt{G[i[m_1,1],i[m_2,1],i[0,1]]} \\ &\to \mathrm{prefac}(\mathtt{n_1,n_2})\,\pi^4\;m_1^2\,\mathtt{N2[m_1]}\,\mathtt{C}^{;(2)}_{\mathtt{111}}, \end{aligned} \tag{85}$$

where the prefactor $\mathrm{prefac}(\mathtt{n_1,n_2})$ depends on the powers n_1 and n_2 and $\mathtt{C}^{(2)}_{\mathtt{111}}$ is given by eq. (77).

d) Factorising two-loop integrals can be directly integrated by making the replacement

$$\mathtt{AD[i[m_1,n_1],i[m_2,n_2]]} \to \frac{\pi^4\;\mathtt{Ne[m_1]Ne[m_2]}}{(m_1^2)^{n_1+n_2-4}} C^{(1)}_{n_1} C^{(1)}_{n_2}, \tag{86}$$

where $C^{(1)}_{n_1} C^{(1)}_{n_2}$ is given by eq. (80). The replacement rules `nerules` will express the product $\mathtt{Ne[m_1]Ne[m_2]}$ in terms proportional to `N2[m1]` according to eqs. (81-83), if $\mathtt{m_2}$ is replaced by $\sqrt{\mathtt{x1}}\;\mathtt{m_1}$. Note that in the package `x1` (not `x`) denotes the mass relation m_2^2/m_1^2.

All the the results of `ScalIntTwo` are expanded in `eps` up to zeroth order.

5. Documentation of Fermions

5.1. *Declarations*

`DeclareMass[MT,MW,...]` used to declare all appearing masses.
`DeclareMomentum[q1,q2,k, ...]` used to declare all appearing momenta.
`DeclareIndex[mu,nu,rho,sigma,...]` used to declare all appearing Lorentz indices.
`DeclarePolarizationVector[epsilon]` used to declare polarisation vectors, which are treated, except for their properties under conjugation, in the same way as momenta.
`DeclarePolarizationVector[epsilon,k]` additionaly sets $epsilon(k) \cdot k = 0$.

All of these functions can be called with an arbitrary number of arguments. When using one of the newer MATHEMATICA front-ends, it is also possible to use indices in Greek letters like μ instead of `mu`.

5.2. *Dirac Algebra*

`DiracLinearity[expr]` expands all sums within `Dirac[]` and takes prefactors of masses, momenta and indices out of `Dirac[]`. It does the same for `Scal[]`.
`DiracAlgebra[expr]` performs the standard Dirac algebra according to eqs. (1), (2) and (4).
`ContractIndex[expr,{mu,nu,...}]` contracts all Lorentz indices given in curly brackets. For longer exressions `Expand` (or `DiracLinearity`) may have to be used first.
`ContractAllIndices[expr]` contracts all silent indices. For longer expressions `DiracLinearity` may have to be used first.
`DiracSort[expr,reflist]` orders any sufficiently simple expression of γs in the order specified in `reflist` (a list containing all the momenta and indices appearing in `expr`). For longer `expr` `DiracCollect[expr]` has to be used first. Projectors (`L`, `R` or `Gamma5`) as well as all momenta have to appear in the `reflist`.
`UseDiracEquation[{p,mp},expr,{q,mq}]` sorts `expr` and uses the Dirac equation for particles (as in $\bar{u}(p)$ `expr` $u(q)$). For antiparticles the corresponding syntax is `UseDiracEquation[{p,-mp},expr,{q,-mq}]`([as in $\bar{v}(p)$ `expr` $v(q)$).
In analogy `UseDiracEquation[{p,mp},expr,{}]` and `UseDiracEquation[{},exp,{q,mq}]` can be used.
`DiracScalExpand[expr]` expands all arguments in `Dirac[]` and `Scal[]` (this may be needed in order to contract indices).
`DiracCollect[expr]` collects all different Dirac structures.
`DiracFactor[expr]` functions like `DiracCollect[expr]` and additionally factorises the coefficient of each of these different structures.

5.3. *Squaring and Traces*

The functions presented above are useful at the amplitude level of a high-energy calculation. In order to obtain physical quantities as cross-sections and decay-rates, it is necessary to have the tools to conjugate or square the expressions and to compute traces over products of γ matrices. This functionality is provided by the following commands:

`DiracAdjunction` computes the Dirac adjoint ($\bar{M} = \gamma^0 M^\dagger \gamma^0$) of a product of γ matrices.

`DiracSquare[expr,one,two]` returns the trace of `expr` · `one` · `DiracAdjunction[expr]` · `two`, where `one` and `two` have to be `Dirac[]` expressions.

`DiracTrace[Dirac[...]]` represents the trace over the Dirac expression; the trace is not evaluated. Non-Dirac expressions may taken out of `DiracTrace[...]` using `DiracTraceLinearity`. To evaluate the trace, `DiracTraceAlgebra` has to be applied to the expression.

`LearnDiracTraceRule[Dirac[k1,k2,...,kn]]` increases the speed of calculations of traces with projectors or with many momenta by adding rules to `ExtendedDiracTraceList`. Only momenta, indices or projectors are allowed as input to this command. If e.g. `Dirac[k1,...k10]` is entered the routine will also learn the rule for any shorter expression of this form (e.g. `Dirac[k1,...k8]` and `Dirac[k1,...,k6]`). No rules for the scalar products of these momenta, such as `Scal[k1,k2] = 0` are allowed to be implemented.

Traces are evaluated strictly in $D = 4$. For traces over long products of γ matrices it is highly recommended to use `LearnDiracTraceRule` first in order to significantly speed up the calculation.

Traces involving γ_5 (and L or R) will generally produce terms involving the ε-tensor (the Levi-Cività symbol). The functions handling this object are:

`Epsilon[a,b,c,d]` is the completely antisymmetric tensor in four dimensions. The convention $\epsilon^{0123} = -1$ is applied.

`EpsilonSort[expr,reflist]` sorts expressions in `Epsilon[...]` according to `reflist`.

`ContractScalEps[expr]` contracts expressions like `Epsilon[a,...] Scal[a,...]`.

`EpsilonEpsilonContract` contracts products of the form `Epsilon[a,b,c,d]` `Epsilon[a,e,f,g]`. The same indices have to be the first in the list, otherwise `EpsilonSort` has to be used first.

5.4. *Setting Scalar Products On Shell and Replacing of Scalar Products*

In the calculation of physical high energy quantities, momenta are often restricted by the requirement that particles are on their mass shell. Furthermore four-momentum conservation allows to express scalar products by other scalar products thus reducing the number of different terms. The functions tailored for these needs are:

`SetOnShell[{p1, m1}, {p2, m2}]` sets $p1 \cdot p1 = m1^2$ and $p2 \cdot p2 = m2^2$.

`ReplScal[Scal[p1,p2], p1+p2+q1==q2]` generates a replacement list of the form $p1 \cdot p2 \to \frac{1}{2}(-(p1^2) - 2p1 \cdot q1 - p2^2 - 2p2 \cdot q1 - 2q1 \cdot q1 + q2^2)$ obtained by squaring both sides of the identity given as second argument of the function. So `q2==p1+p2+q1` will produce a different result than `-q1==p1+p2-q2`. `-q-p1==p2-p` will return an empty list.

6. Documentation of Integrals

6.1. *Additional Declarations*

`DeclareSmallMass[MU, MD]`: Needed for `Scaling`.
`DeclareHeavyMass[MT,MW]`: Needed for `TaylorMass`.
`DeclareExternalMomentum[k1,k2]`: Needed for `Scaling` and `TaylorExpansion`.
`DeclareLoopMomentum[q1,q2]`: Needed for `TaylorExpansion`.

6.2. *Transformation of the Integrals to Scalar Integrals*

`Color` replaces colour structures depending on generators and structure constants of $SU(3)_c$ on expressions only depending on generators or scalars corresponding to eqs. (12-14).

`TaylorExpansion` expands denominators of the form `AD[..., den[q+k, m],...]`, where `q` is a loop momentum or the sum of loop momenta, in k up to second order. Note that loop momenta have to be declared with `DeclareLoopMomentum` first.

`TaylorMass` expands denominators of the form `AD[..., den[q, m], ...]`, where `q` is a loop momentum, in `m` up to second order, if m is NOT declared as heavy mass with `DeclareHeavyMass`.

`Scaling` multiplies all momenta declared as external with `DeclareExternalMomentum` and all masses declared as small with `DeclareSmallMass` with a factor x and sets then all powers x^n with $n > 2$ a to zero.

`PartialFractionOne` makes partial fraction of the denominators in the one-loop case according to eqs. (20-21) and successively gets rid of loop momenta from the numerators successively according to eq. (22).

`PartialFractionTwo` makes partial fraction of the denominators in the two-loop case according to eqs. (20-21) and successively gets rid of loop momenta from the numerator according to eqs. (22)-(23) and sets all vanishing massless integrals to zero.

`TensorOne[expr,var]` performs the one-dimensional tensor reduction in `var`. It assumes that the denominator of `expr` is an arbitrary scalar function depending on Lorentz invariants of var. It can handle expressions `expr` with up to 9 Lorentz Indices. Results are Taylor expanded in `eps` up to second order.

`TensorTwo[expr,var,var2]` performs a two dimensional tensor reduction of expressions `expr` with up to five Lorentz Indices assuming that the denominator of `expr` is an arbitrary scalar function of the variables `var1` and `var2`. If the numerator of `expr` depends only on `var` (`var2`), it performs a one-dimensional tensor reduction in `var` (`var2`) using `TensorOne[expr,var]` (`TensorOne[expr,var2]`). Expressions like `Scal[var,var]` are treated like `Scal[var,lor1]Scal[var,lor1]`, where `lor1` is a Lorentz index. This artificially increases the number of used Lorentz indices. Therefore it is recommended to set all quadratic scalar products to a dummy variable before performing the tensor reduction. Results are expanded in `eps` up to second order. Factorising two-loop integrals have to be tensor-reduced with `TensorOne` before usage of `TensorTwo`.

`SimplifyPropagator` brings propagator structures to the form needed for loop integration.

6.3. *Integration of Scalar Integrals*

`ScalIntOne[expr]` allows the calculation of scalar one-loop integrals by replacing the propagator structure `AD[i[m,n]]` by the right hand side of eq. (41):

$$\mathtt{AD[i[m,n]]} \to \frac{\pi^2}{(m^2)^{n-2}} \mathtt{Ne(m)}\ \mathtt{C}_\mathtt{n}^{(1)}, \tag{87}$$

where `Ne(m)` corresponds to eq. (42) and $\mathtt{C}_\mathtt{n}^{(1)}$ to eq. (44) up to second order in `eps`. Results are expanded up to first order in `eps`.

`ScalIntTwo[expr]` allows the calculation of scalar integrals independent of external momenta and with one massless propagator. It replaces in `expr` propagator structures of the form `G[i[m1,n1],i[m2,n2],i[0,n3]]` with the analytical result of the corresponding scalar twoloop integral $G^{m1,m2,0}_{n1,n2,n3}$ as defined in eqs. (84-86). Note that the result is expanded in `eps` up to zeroth order.

`ScalIntTwoThreeMasses[expr]` allows the calculation of scalar loop integrals independent of external momenta and with up two different masses expressions of the form `G[i[m1,n1],i[m2,n2],i[m1,n3]]` with the analytical result for scalar two-loop integrals of the form $G^{m1,m2,m1}_{n1,n2,n3}$ as defined in eq. (63). Note that the result is expanded in `eps` up to zeroth order.

`nerules` are replacement rules allowing to express prefactors `Ne[M1]Ne[M2]` in terms proportional to `Ne[M2]`, if`M2` is given as `M2=Sqrt[x1]*M1`.

7. Installation Instructions

The package MasterTwo can be downloaded from
`https://github.com/shhschilling/MasterTwo/blob/master/ManualMasterTwo.pdf`

7.1. *Installation under Linux*

Copy the zip file MasterTwo-1.0.zip to your disk and unpack it with

```
> gunzip MasterTwo-1.0.zip
```

Change to the MasterTwo-1.0 directory

```
> cd MasterTwo-1.0
```

and change the permission of the installation script:

```
> chmod +x MasterTwoInstall
```

Execute it with

```
> ./MasterTwoInstall
```

and follow the instructions. The installation package will update the init.m file in the .Mathematica/Autoload/ directory, so that you can load the package without having to give the whole path.

Uninstallation under Linux

Run the program

```
> ./MasterTwoUninstall
```

in the installation directory of MasterTwo.

7.2. *Installation under Windows and MacOs*

MasterTwo has to be copied to one of Mathematica?s Autoload path to make Mathematica aware of the package. Typing

```
$Path
```

on Mathematicas command line identifies the Autoload paths on your system.

Put then the file `Fermions.m, Integrals.m` and `MasterTwo.m` in one of the Autoload paths of your Mathematica installation.

Close Mathematica.

In your next Mathematica session, you can call the package package MasterTwo by typing

```
<<MasterTwo`
```

on the command line.

From here on all the MasterTwo commands are available. The package is equipped with an on-line help to each command. `MasterTwoInfo[]` produces a list with all the available commands and `?command` prints a short information on syntax and effect of `command`.

8. Generation of the Integrands: FeynArts and MasterTwo

The natural starting point for the generation of the integrands of the one and two-loop integrals to be integrated is the usage of the programme FeynArts [16]. The existing model files for the SM-model and the SUSY extensions can be easily adapted to the conventions needed for the processes to be calculated. But the Feynman-amplitudes generated by FeynArts are not appropriate for the routines used in MasterTwo. Thus the function `FeynArtsToMasterTwo` translates Standard Model output generated by FeynArts into a form adapted for the further usage of MasterTwo. In the following we list the most important automatic replacements:

Renaming of the headers

$$\begin{aligned} \texttt{FeynAmp}[\texttt{GraphName}[...], \texttt{Integral}[...], \texttt{c}] &\to \texttt{c}, \\ \texttt{FermionChain} &\to \texttt{Dirac}, \\ \texttt{PropagatorDenominator} &\to \texttt{den}, \\ \texttt{MetricTensor} &\to \texttt{Scal}, \\ \texttt{FourVector} &\to \texttt{Scal} \end{aligned}$$

Note that in the first replacement only the integrand of the integral is kept. Thus in the final calculation of scalar integrals we will actually replace the propagator structure of scalar integrands with the value of the corresponding scalar integral.

Replacements in Fermion chains

$$\begin{aligned} \texttt{ChiralityProjector}[-1] &\to \texttt{L}, \\ \texttt{ChiralityProjector}[1] &\to \texttt{R}, \\ \texttt{DiracSlash}[a] &\to \texttt{a} \end{aligned}$$

Renaming of the momenta

FEYNARTS declares internal momenta by `FourMomentum[Internal, i]`, external momenta as `FourMomentum[External, j]`, where $\mathtt{i} = 1, ..., \mathtt{l}$ ($\mathtt{j} = 1, ..., \mathtt{k}$) stands for the `i.` (`j.`) internal (external) momenta appearing in the diagram. `FeynArtsToMasterTwo` makes then the following replacements:

$$\mathtt{FourMomentum}[\mathtt{Internal}, \mathtt{j}] \rightarrow \mathtt{qj},$$
$$\mathtt{FourMomentum}[\mathtt{Outgoing}, \mathtt{j}] \rightarrow \mathtt{kj}$$

Renaming of Lorentz indices

$$\mathtt{DiracMatrix}[\mathtt{Index}[\mathtt{Lorentz}, \mathtt{a}]] \rightarrow \mathtt{lora}$$

Dirac Spinors

Dirac Spinors are by default set to one by making the replacement

$$\mathtt{DiracSpinor}[\mathtt{a\_}] \rightarrow \mathtt{Times}[]$$

If the user wants to use the Dirac equation or is interested in the calculation of squared matrix elements etc. this replacement should be commented.

The function `FeynArtsToMasterTwo` depends very much on the concrete process to be calculated and has to be adapted when using new model files, calculating different processes, using newer versions of `FeynArts` etc.

9. Example of a Two Loop Diagram

In fig. 2 we show an example diagram of the two-loop decay $b \rightarrow s\ \gamma$. Its calculation with the help of MASTERTWO is given in the file `Example.nb` included in this distribution.

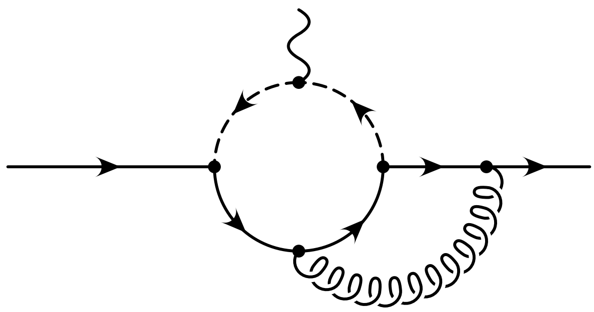

Fig. 2. Example: A one-particle irreducible two-loop diagram for $b \rightarrow s\gamma$. The external quark lines (solid) denote the incoming b-quark and the outgoing s-quark, respectively. The wavy line denotes a virtual photon. The internal dashed-, solid- and curly lines denote the charged W boson $W^{\pm}$, the t-quark and the gluon, respectively.

10. Acknowledgements

S. Schilling would like to thank D. Wyler for helpful discussions. Special thanks to M. Misiak and J. Urban for fruitful discussions and advice regarding the technical details of the two-loop calculations, P. Liniger for the provision of the original code and documentation of FERMIONS as well as K. Bieri for some routines concerning the tensor reduction now implemented in INTEGRALS. This work was partially supported by the Swiss National Foundation.

References

[1] Liniger, P., *Charmless B-Meson Decays in the Standard Model*, PhD thesis, University of Bern, 2001.
[2] Breitenlohner, P. and Maison, D., Commun. Math. Phys. **52** (1977) 11,39,55.
[3] Bonneau, G., Nucl. Phys. **B177** (1981) 523.
[4] Buras, A. J. and Weisz, P. H., Nucl. Phys. **B333** (1990) 66.
[5] Field, R., *Applications of perturbative QCD*, Addison-Wesley, 1989.
[6] Smirnov, V. A., Mod. Phys. Lett. **A10** (1995) 1485.
[7] Muta, T., World Sci. Lect. Notes Phys. **57** (1998) 1.
[8] Chetyrkin, K. G., Misiak, M., and Munz, M., Nucl. Phys. **B518** (1998) 473.
[9] Urban, J., *Test des Standardmodells im B-Mesonen-Sektor durch Mischungsphänomene und seltene Zerfälle*, PhD thesis, TU Dresden, 1999.
[10] Bobeth, C., Misiak, M., and Urban, J., Nucl. Phys. **B574** (2000) 291.
[11] Collins, J. C., *Renormalization. An introduction to renormalization, the renormalization group, and the operator product expansion*, Cambridge, Uk: Univ. Pr., 1984.
[12] Davydychev, A. I. and Tausk, J. B., Nucl. Phys. **B397** (1993) 123.
[13] Abramowitz, M. and Stegun, I. E., *Handbook of Mathematical Functions*, New York: Dover, 1972.
[14] Wolfram, *MATHEMATICA, Version 6.0.2*, Champaign, Illinois: Wolfram Research, Inc., 2008.
[15] Heck, A., *Introduction to Maple*, New York: Springer, 1993.
[16] Hahn, T., Comput. Phys. Commun. **140** (2001) 418.