ETH zürich

Extraction of UV part of two-loop tensor integrals

Semester project

GUGLIELMO COLORETTI
Eidgenössische Technische Hochschule Zürich
M. Sc. in Physics, Spring Semester 2020
gcoloretti@student.ethz.ch

Supervised by
Prof. Dr. KIRCH KLAUS STEFAN

Co-supervised by
Prof. Dr. SIGNER ADRIAN

Abstract

We present an algorithm able to extract the UV (ultraviolet) divergent part of a certain type of two-loop tensor integral by reducing them in terms of only one-loop squared and two-loop tadpole integrals. We implement it using Mathematica and tested it on several diagrams. The code can be useful for the computation of anomalous dimensions.

Contents

1. Preliminaries

1.0. Identities

Our object of interest is the following tensor two-loop integral[1]:

$$\frac{k_1^{\mu_1}\dots k_1^{\mu_{n1}} k_2^{\nu_1}\dots k_2^{\nu_{n2}}}{[k_1^2-M_1^2]^{a_1}[k_2^2-M_2^2]^{a_2}[(k_1+k_2)^2-M_{12}^2]^{a_3}[(k_1+p_i)^2-M_i^2]^{a_4}[(k_2+p_j)^2-M_j^2]^{a_5}} \tag{1}$$

where the p's stand for two external momenta and numerators will be freely contracted depending on the specific Feynmann diagram. Our starting point is the belief that any integral of this type could be reduced in a sum of one-loop squared integrals, general two-loop tadpoles and UV finite terms. Since we are interested in extracting the UV divergent part, we can drop these last contributions. In order to massage the integral and lead it to the form we aim at, we use some basic identities, that we already name in the following way for later convenience:[2].

$$\begin{aligned}
&\texttt{tad-k:} && \frac{1}{k^2-M^2} = \frac{1}{k^2-\mu^2} + \frac{M^2-\mu^2}{(k^2-\mu^2)(k^2-M^2)}\\
&\texttt{tad-kp:} && \frac{1}{(k+p)^2-M^2} = \frac{1}{k^2-\mu^2} - \frac{\mu^2+2k.p}{(k^2-\mu^2)[(k+p)^2-M^2]}\\
&\texttt{tad-kk:} && \frac{1}{(k_1+k_2)^2-M^2} = \frac{1}{(k_1+k_2)^2-\mu^2} + \frac{M^2-\mu^2}{[(k_1+k_2)^2-\mu^2][(k_1+k_2)^2-M^2]}\\
&\texttt{oneloop-k1:} && \frac{1}{(k_1+k_2)^2-M^2} = \frac{1}{(k_1^2-\mu^2)} + \frac{(M^2-\mu^2)-2k_1.k_2-{k_2}^2}{({k_1}^2-\mu^2)[(k_1+k_2)^2-M^2]}\\
&\texttt{oneloop-k2:} && \frac{1}{(k_1+k_2)^2-M^2} = \frac{1}{(k_2^2-\mu^2)} + \frac{(M^2-\mu^2)-2k_1.k_2-{k_1}^2}{({k_2}^2-\mu^2)[(k_1+k_2)^2-M^2]}
\end{aligned} \tag{2}$$

where μ is an artificial parameter introduced ad hoc that we should check drops

[1] The integration measure is the canonical one and we work in dimensional regularization scheme; thus the measure is $\tilde{\mu}\int\frac{dk_1^d}{(2\pi)^d}\frac{dk_2^d}{(2\pi)^d}$, where $\tilde{\mu}$ is a mass dimension parameter chosen so that the mass dimension of the integral is the correct one also in d dimensions.

[2] Here and in the following we adopt the convention of Package X' from Mathematica to write scalar product and deal with tensor quantities; e.g. the scalar product of two tensor in Minkowski space is denoted by a dot as $p.q$. More information at `https://packagex.hepforge.org/primer-2.1.1.pdf`

out at the end[3,4] and M is a general mass term. Further, k is a general variable of integration (in our case k_1 or k_2). The goal is to find a way to use this identities in a systematic way to extract the UV divergent part of any integral of the type (1).

1.1. Analysis of the reduction rules

Identities (2) are set for our reduction purpose. We can group the first three ones and find a common pattern. Indeed all of these are made by two parts: the first is a tadpole propagator with the artificial μ parameter as mass term and does not alter the respective degree of divergence[5]; the second one instead produces a mixed mass propagator in the external mass M and the artificial parameter μ and, above all, improves the degree of divergence (i.e. makes the term more UV convergent) so that we will eventually drop it. The last rule has a different effect: indeed it is able to drag the divergence between k_1 and k_2; thus it allows us to reduce the divergence in one variable, although it could be potentially costly since it could make divergent the integral in the other variable via the terms in the numerator. We can summarize the behaviour of these rules in the following table[6]:

`tad-k`	$\text{Tadpole}_\mu + \ k^{-\,-}$
`tad-kp`	$\text{Tadpole}_\mu + \ k^{-}$
`tad-kk`	$\text{Tadpole}_\mu + \ (k_1 + k_2)^{-\,-}$
`oneloop-k1`	$\text{Tadpole}_\mu + \ k_1^{-\,-} \ + k_1^- k_2^+ \ + \ k_2^{+\,+}$
`oneloop-k2`	$\text{Tadpole}_\mu + \ k_2^{-\,-} \ + k_2^- k_1^+ \ + \ k_1^{+\,+}$

Table 1: Behaviour of the reduction rules.

In particular, we tried to highlight the behaviour of the last two rules adding colors to identities (2): red stuff gets more UV finite, blue stuff gets more UV divergent.

Therefore, overall, our rules push the divergence towards tadpole-like terms or drag it towards one variable.

[3]This sometimes could be tricky if this parameter ends up in the argument of complicated functions; however we can check the correctness of our result numerically. More about this issue later.

[4]Let us further stress again that this parameter is not to be confused with the parameter $\tilde{\mu}$ coming from the integration measure in d dimensions.

[5]Here we mean that the first piece of identities `tad-k1` and `tad-k2` leave unchanged $dod1$ and $dod2$ respectively, as well as `tad-kk` does not change $dodOA$. See next section for notation.

[6]To clarify notation: $Tadpole_\mu$ stands for a tadpole term with μ mass term, whereas a power in k_1 means the degree of divergence decrease ($-$) or increase ($+$).

1.2. Evaluating the degree of divergence

First we shall define some conditions that allow us to drop the finite part of the integral. The divergences could arise both from k_1 and k_2 terms as well as from their combination. Further we can perform shifts that affect the combined degree of divergence; we shall refer to the latter as overall degree of divergence. Indeed, consider for instance the following integral and shift:

$$\frac{1}{[k_1]^2[k_2]^2[k_1+k_2]^6} \xrightarrow{k_1 \to k_1 - k_2} \frac{1}{[k_1-k_2]^2[k_2]^2[k_1]^6} \tag{3}$$

Before the shift the integral appears to be UV finite in both variables, but afterwards the UV divergence in k_2 is manifest.
Sticking to the notation of (1) we can organize all UV divergences of a general two-loop integral in a list:

$$dod(n_1, n_2, a_1+a_4, a_2+a_5, a_3) \to dod(n_1, n_2, c_1, c_2, c_3), \tag{4}$$

where the n_i stand for numerator powers, a_i stand for propagator powers and we use the notation with c_i for later convenience. The overall degree of divergence is seen to be

$$dodOA = \underbrace{[2\times 4]}_{\substack{\text{Integration}\\ \text{measure}}} + n_1 + n_2 - 2\underbrace{[a_1+a_2+a_3+a_4+a_5]}_{c_1+c_2+c_3}. \tag{5}$$

Taking account of shifts too, three cases can occur:

1. $c_3 \le \min[c_1, c_2] \quad \to \quad \begin{cases} dod1 = n_1 - 2(c_1+c_3) \\ dod2 = n_2 - 2(c_2+c_3) \\ dodOA = n_1+n_2-2(c_1+c_2+c_3) \end{cases}$

2. $c_2 \le \min[c_1, c_3] \quad \to \quad \begin{cases} dod1 = n_1 + n_2 - 2(c_1+c_2) & \boxed{\text{shift: } k_2 \to -k_2 - k_1} \\ dod2 = n_2 - 2(c_2+c_3) \\ dodOA = n_1+n_2-2(c_1+c_2+c_3) \end{cases}$

3. $c_1 \le \min[c_2, c_3] \quad \to \quad \begin{cases} dod1 = n_1 - 2(c_1+c_3) \\ dod2 = n_1 + n_2 - 2(c_1+c_2) & \boxed{\text{shift: } k_1 \to -k_1 - k_2} \\ dodOA = n_1+n_2-2(c_1+c_2+c_3) \end{cases}$

Finally, we recall that for UV convergence we need both degree of divergence in k_1 and k_2 to be less than -4, as well as the overall degree to be less than -8 (namely: $dod1 < -4$, $dod2 < -4$, $dodOA < -8$).

2. Algorithm

2.0. Detailed analysis

We recall the strategy is to reduce the divergent part of the integral (1) to only tadpoles in an artificial scale μ and one-loop squared terms. In the next section we add a flowchart in order to make it as clean as possible, so that one could only look at it if not interested in the reasoning that leads to the reduction algorithm.

TADPOLES

We note that each rule gives rise to terms which are pure tadpoles in μ mass term so that we hope that after applying them a sufficient number of times we are left with pieces that involve only this type of tadpoles. We can obtain this behaviour applying the first three rules repeatedly. In particular we note that we can get rid of the extra scale given by external momenta applying `tad-kp`, which thus leads us towards tadpoles.

ONE-LOOP SQUARED

We need to find a way to drop the mixed propagator in both variables, which is guilty of spoiling the separation in two different one-loop terms. The key idea is to use the last rule. Indeed this rule transforms a propagator in both variables into one depending only in a single variable (the first part in μ tadpole-like term) and a term which increase the divergence in one variable against the other.

Thus, if we have a propagator in one variable lifted to a power high enough, so that we can drag it to the other variable in order to make the integral in the latter UV finite, while still keeping the integral in the first variable UV finite, we can drop the term in both variables and we are left with only one-loop squared integrals.

This is achieved via application of `oneloop-k1` and `oneloop-k2` depending on the condition defined above. Therefore the goal is now to find a way that leads us to meet this condition.

The easiest and most evident thing to do is making more finite (namely, increase the propagator power) the integral in one variable so that we meet the condition above and we can use its finiteness to reduce also the other variable. We can use the first three rules to do so: indeed we sort of move the divergence to the first tadpole-like term. However, it could happen that after applying the first three rules, we are left with a term which is a tadpole in μ mass term multiplied by propagator in both variable with a different mass term[7]. At this point, using the first three rules does not give us any gain since we have propagators with μ mass term and rules in each single variable do not change them. We could apply `tad-kk` to the propagator with both variables and the mass term different from

[7] This is manifestly due to `tad-kk`.

μ, but this does not improve the result towards our one-loop squared reducible condition because, roughly speaking, the power of this propagator compensates with the additional pieces of the rule that should make it more UV finite (see next section for a full explanation). Therefore if the condition above is not met at this stage, we can do nothing but exploit the mixed mass dependence in the propagator with both variables. Indeed we can shift one variable so that we obtain again a propagator in both of them with a mass term different from μ[8]:

$$
\begin{gathered}
\frac{M^2\ k_1.k_1}{\underbrace{[k_1^2-\mu^2]^2[k_2^2-\mu^2]}_{\substack{\text{Cannot reduce via}\\ \texttt{tad-k}}}[(k_1+k_2)^2-M^2][(k_1+k_2)^2-\mu^2]^3} \\
\Big\downarrow\ k_2 \to -k_2-k_1 \\
\frac{M^2\ k_1.k_1}{[k_1^2-\mu^2]^2\underbrace{[k_2^2-M^2]}_{\substack{\text{Can reduce}\\ \text{via } \texttt{tad-k}}}[k_2^2-\mu^2]^3[(k_1+k_2)^2-\mu^2]}
\end{gathered}
\tag{6}
$$

Now we can keep on applying the first three rules and reach the condition for one-loop squared reduction. It is worth to highlight a possible issue that could prevent performing the shifts: if in the term in question there are propagators in both external momenta, we cannot perform the shift in any variable. Thus we need to make sure not to meet this condition when applying shifts. This is not hard since we can get rid of external momenta via `tad-kp`, which leaves a tadpole in μ mass term with the same degree of divergence and a term more UV finite depending on the external momenta.

At this moment we realize that the shifts are a powerful weapon and that we can actually try to make the reduction faster exploiting them. Indeed it is worth performing shifts after applying each rule towards the variable which is more UV finite, so that we are pushed to the one-loop squared reducible condition significantly faster. So we apply shifts for two different purposes: sometimes we must necessarily perform them in order to have one-loop squared terms, some other times we perform them to make the reduction faster.

2.1. Implementation

We show here a flow chart that should clarify the previous discussion. We add also some comments regarding the problematic steps.

[8]If we aim to generalize this procedure to general two-loop integrals, we need to pay attention to this shift, since in general it would not be allowed, namely if we have a propagator with $(k1-k2)$.

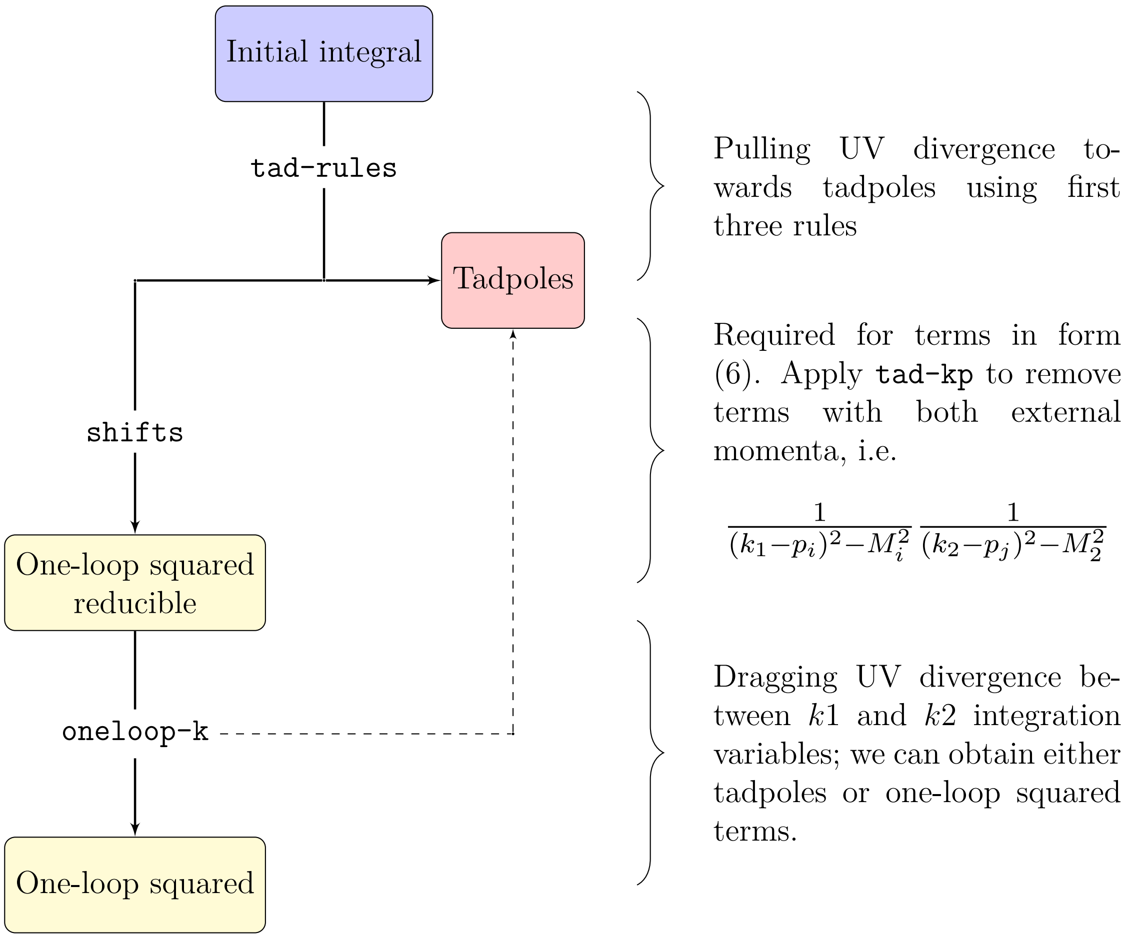

Figure 1: Flowchart for the reduction algorithm.

We thus report the already mentioned condition that ensures us that a certain term is reducible to one-loop squared and tadpoles. Namely we check whether there is a variable elevated to a power high enough, so that we can drag its finiteness to the other UV divergent one, while keeping the first still UV finite. This check is performed in the following way:

ONE-LOOP SQUARED REDUCIBLE CONDITION

$$\boxed{0 \leq 2c_3(5 - |\max[dod1, dod2]|) \leq |\min[dod1, dod2]| - 5},$$

where we stick to the notation of section (1.2.)[9]. The factor of 2 is due to the worst UV divergent term appearing in the expression of the rules (see `oneloop-k` rules in Table 1 (1) or in the list of identities (2)). The factor of c_3 is there since

[9] in particular c_3 is the power of the propagator in $(k_1 + k_2)$.

we need to check this condition for every power of the propagator in $(k_1 + k_2)$. We can now explain a misleading point of the previous section. We stated that applying `tad-kk` to a term which is a full tadpole times a propagator in both variables and with a different mass term, does not improve the result towards the one-loop squared reducible condition above. Indeed, the reason in this case lies in the presence of the power c_3 of the propagator in both variables. Applying `tad-kk` leads to the following[10]:

- $\max[dod1, dod2]$ stays the same;
- $\min[dod1, dod2]$ increases by $+2$;
- c_3 increases by $+1$.

This pushes the term further and further away from meeting the condition for one-loop squared reducibility above. Here is why we are forced to perform shifts.

2.2. Code

Here we clarify certain aspects of our code. Although it might seem that our algorithm is difficult to practically write in a code, we use an escamotage that Mathematica allows. Indeed, being symbolic, we append to each different term flags that mark their nature so that we can easily find them and treat separately. We thus append a flag to each term that basically tells us which rule to apply on them or if they are already in our desired form (tadpoles or one-loop squared). Flags are attached after each check. In particular the flags we use are:

1. Tadpole: we keep them since they are already in our desired form
2. One-loop squared: same as for tadpoles
3. One-loop squared reducible: we apply to all these terms `oneloop-k1` or `oneloop-k2`
4. Shift: this flag is appended if we are able to perform a shift (namely if we do not have both external momenta in k_1 and k_2 propagators). Shifts are then applied to a specific variable depending on the previous condition. If we are able to perform shifts in both variables, we arbitrarily decide to apply it in the first variable; indeed, at the end we will drag the divergences between them and this makes no difference for our purpose.
5. Non-tadpole propagator: we apply `tad-k` to all these terms
6. External momenta propagator: we apply `tad-kp` to all these terms

[10]For definiteness, please refer to equation (6).

Therefore we use nested while cycles both to append flags to terms and to perform the desired operation. Indeed, a cycle marks each term with the respective flag and another one performs on each term the required operation, depending on the flag that marks it. Cycles stop when all terms have a tadpole or a one-loop squared flag attached.

We implemented the algorithm in a working code with Mathematica.[11]

3. Example

We tested our reduction algorithm on several examples, extracting also the UV poles up to second order. Since in this project we are interested in the reduction algorithm and we want to be concise, we will not spend many words regarding the techniques used (for references see [1], [2]). One-loop squared terms are solved using Package X‘. Tadpoles are trickier: we perform the tensor decomposition by hand to reduce them to scalar tadpoles. These in turn are reduced to a minimal set of master tadpoles using integration-by-parts identities. The remaining irreducible master integrals are then calculated using Mellin-Barnes techniques.

We report here the result for a highly non-trivial example. We consider the most complicated tensor structure for the following diagram:

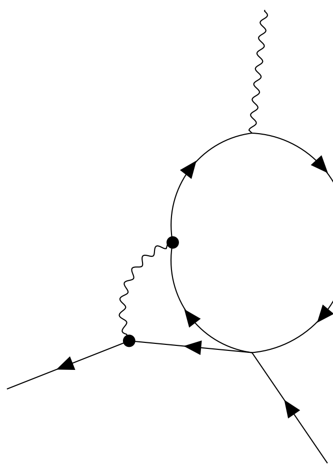

Figure 2: Feynman diagram of the example.

The structure thus reads

$$\frac{k_2^\mu k_2^\nu k_2^\rho k_1^\sigma}{[k_1^2][(k_1+p_1)^2-M_1^2][(k_1+k_2)^2-M_x^2][k_2^2-M_x^2][(k_2+q)^2-M_x^2]} \tag{7}$$

[11]To obtain the package, please write to *gcoloretti@student.ethz.ch*.

and the kinematics

$$q = p_2 - p_1, \quad p_1^2 = M_1^2, \quad p_2^2 = M_2^2, \quad q^2 = 0 \tag{8}$$

The reduction outputs 166 terms: 56 tadpoles and 110 one-loop squared elements. The reduction leads to the following poles (we retain only second order in the expansion):

$$\begin{aligned}
&\frac{1}{144\epsilon^2}\Big[p_1^\sigma(-2p_1^\rho g^{\mu\nu} - 2p_1^\nu g^{\mu\rho} - 2p_1^\mu g^{\nu\rho} + 3q^\rho g^{\mu\nu} + 3q^\nu g^{\mu\rho} + 3q^\mu g^{\nu\rho} \\
&\quad - (M_1^2 + 3M_x^2)(g^{\mu\nu}g^{\sigma\rho} + g^{\mu\rho}g^{\sigma\nu} + g^{\sigma\mu}g^{\nu\rho})\Big] + \\
&\frac{1}{3456\epsilon}\Big[144p_1^\sigma q^\rho \log(\frac{\tilde{\mu}^2}{M_1^2})g^{\mu\nu} + 2p_1^\rho(-48p_1^\sigma \log(\frac{\tilde{\mu}^2}{M_1^2})g^{\mu\nu} - 12p_1^\mu g^{\sigma\nu} - 40p_1^\sigma g^{\mu\nu} \\
&\quad + 9q^\nu g^{\sigma\mu} + 9q^\mu g^{\sigma\nu} + 9q^\sigma g^{\mu\nu}) + 144p_1^\sigma q^\nu \log(\frac{\tilde{\mu}^2}{M_1^2})g^{\mu\rho} - 2p_1^\nu(48p_1^\sigma \log(\frac{\tilde{\mu}^2}{M_1^2})g^{\mu\rho} \\
&\quad + 12p_1^\rho g^{\sigma\mu} + 12p_1^\mu g^{\sigma\rho} + 40p_1^\sigma g^{\mu\rho} - 9q^\rho g^{\sigma\mu} - 9q^\mu g^{\sigma\rho} - 9q^\sigma g^{\mu\rho}) \\
&\quad + 144p_1^\sigma q^\mu \log(\frac{\tilde{\mu}^2}{M_1^2})g^{\nu\rho} - 96p_1^\mu p_1^\sigma \log(\frac{\tilde{\mu}^2}{M_1^2})g^{\nu\rho} - 48M_1^2 \log(\frac{\mu^2}{M_1^2})g^{\mu\nu}g^{\sigma\rho} \\
&\quad - 48M_1^2 \log(\frac{\tilde{\mu}^2}{M_1^2})g^{\mu\rho}g^{\sigma\nu} - 48M_1^2 \log(\frac{\tilde{\mu}^2}{M_1^2})g^{\sigma\mu}g^{\nu\rho} - 79M_1^2 g^{\mu\nu}g^{\sigma\rho} \\
&\quad - 79M_1^2 g^{\mu\rho}g^{\sigma\nu} - 79M_1^2 g^{\sigma\mu}g^{\nu\rho} - 9M_2^2 g^{\mu\nu}g^{\sigma\rho} - 9M_2^2 g^{\mu\rho}g^{\sigma\nu} - 9M_2^2 g^{\sigma\mu}g^{\nu\rho} \\
&\quad - 144M_x^2 \log(\frac{\tilde{\mu}^2}{M_x^2})g^{\mu\nu}g^{\sigma\rho} - 144M_x^2 \log(\frac{\tilde{\mu}^2}{M_x^2})g^{\mu\rho}g^{\sigma\nu} - 144M_x^2 \log(\frac{\tilde{\mu}^2}{M_x^2})g^{\sigma\mu}g^{\nu\rho} \\
&\quad - 348M_x^2 g^{\mu\nu}g^{\sigma\rho} - 348M_x^2 g^{\mu\rho}g^{\sigma\nu} - 348M_x^2 g^{\sigma\mu}g^{\nu\rho} + 18p_1^\mu q^\rho g^{\sigma\nu} + 18p_1^\mu q^\nu g^{\sigma\rho} \\
&\quad + 222p_1^\sigma q^\rho g^{\mu\nu} + 222p_1^\sigma q^\nu g^{\mu\rho} + 18p_1^\mu q^\sigma g^{\nu\rho} + 222p_1^\sigma q^\mu g^{\nu\rho} - 80p_1^\mu \mathrm{p}1^\sigma g^{\nu\rho} \\
&\quad - 72q^\nu q^\rho g^{\sigma\mu} - 72q^\mu q^\rho g^{\sigma\nu} - 72q^\mu q^\nu g^{\sigma\rho} - 72q^\rho q^\sigma g^{\mu\nu} - 72q^\nu q^\sigma g^{\mu\rho} \\
&\quad - 72q^\mu q^\sigma g^{\nu\rho}\Big] + \\
&O(\epsilon^0)
\end{aligned} \tag{9}$$

Acknowledgements

I would like to thank professor Kirch Klaus Stefan for letting me be part of this project. I further thank professor Signer Adrian for the opportunity and honor to work in his group at Paul Scherrer Institute, as well as Engel Tim Benjamin for having been patient with me and for his fundamental help in designing the algorithm and dealing with the UV extraction.

Bibliography

[1] S. Weinzierl, *The Art Of Computing Loop Integrals*, https://arxiv.org/abs/hep-ph/0604068.

[2] V. A. Smirnov, *Analytic tools for Feynman integrals*, http://dx.doi.org/10.1007/978-3-642-34886-0.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

EXTRACTION OF UV PART OF TWO-LOOP TENSOR INTEGRALS

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	First name(s):
COLORETTI	GUGLIELMO

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	Signature(s)
ZÜRICH, 28/07/2020	Guglielmo Coloretti

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.