Computer Physics Communications 185 (2014) 1828–1834

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Tools for NLO automation: Extension of the golem95C integral library[☆]

J.Ph. Guillet[a], G. Heinrich[b,*], J.F. von Soden-Fraunhofen[b]

[a] LAPTH, Université de Savoie and CNRS, Annecy-le-Vieux, France

[b] Max-Planck-Institut für Physik, Föhringer Ring 6, 80805 München, Germany

ARTICLE INFO

Article history:
Received 9 January 2014
Received in revised form
4 March 2014
Accepted 9 March 2014
Available online 19 March 2014

Keywords:
NLO computations
One-loop diagrams
Tensor integrals
Higher spin representations
Effective theories

ABSTRACT

We present an extension of the program golem95C for the numerical evaluation of scalar integrals and tensor form factors entering the calculation of one-loop amplitudes, which supports tensor ranks exceeding the number of propagators. This extension allows various applications in Beyond the Standard Model physics and effective theories, for example higher ranks due to propagators of spin two particles, or due to effective vertices. Complex masses are also supported. The program is not restricted to the Feynman diagrammatic approach, as it also contains routines to interface to unitarity-inspired numerical reconstruction of the integrand at the tensorial level. Therefore, it can serve as a general integral library in automated programs to calculate one-loop amplitudes.

New version program summary

Program title: golem95-1.3.0

Catalogue identifier: AEEO_v3_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEEO_v3_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: Standard CPC licence, http://cpc.cs.qub.ac.uk/licence/licence.html

No. of lines in distributed program, including test data, etc.: 242036

No. of bytes in distributed program, including test data, etc.: 1092837

Distribution format: tar.gz

Programming language: Fortran95.

Computer: any computer with a Fortran95 compiler.

Operating system: Linux, Unix.

RAM: RAM used per integral/form factor is insignificant

Classification: 4.4, 11.1.

External routines: some finite scalar integrals are called from OneLOop [1,2], the option to call them from LoopTools [3,4] is also implemented.

Catalogue identifier of previous version: AEEO_v2_0

Journal reference of previous version: Comput. Phys. Comm. 182(2011)2276

Does the new version supercede the previous version?: yes

Nature of problem:
evaluation of one-loop multi-leg integrals occurring in the calculation of next-to-leading order corrections to scattering amplitudes in particle physics. In the presence of particles with spin two in the loop, or effective vertices, or certain gauges, tensor integrals where the rank exceeds the number of propagators N are required.

☆ This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author. Tel.: +49 89 32354284.
E-mail address: gudrun@mpp.mpg.de (G. Heinrich).

http://dx.doi.org/10.1016/j.cpc.2014.03.009

Solution method:
extension of the reduction algorithm to rank $r \leq 10$ for $N \leq 4$ and $r \leq N+1$ for $N \geq 5$, which is sufficient for most applications in Beyond the Standard Model Physics.

Reasons for new version:
the previous version was restricted to tensor ranks less than or equal to the number of propagators.

Summary of revisions:
tensor ranks $>N$ are supported, an alternative reduction method for the case of infrared divergent triangles is implemented, numerical stability for the case of small mass differences has been improved.

Running time:
depends on the nature of the problem. A single call to a rank 6 five-point form factor at a randomly chosen kinematic point, using real masses, takes 10^{-3} s on an Intel Core 4 i7-3770 CPU with a 3.4 GHz processor.

1. Introduction

Over the last years, enormous progress has been made to push the calculation of NLO corrections towards a higher number of particles in the final states, i.e. to “multi-leg” amplitudes, in QCD as well as in the electroweak sector.

Nowadays, the efforts are focused on the goals of *automating* multi-leg one-loop calculations and making them *publicly available*. Programs aiming at the complete automation of one-loop amplitude calculations, including the generation of the amplitude requested by the user “on the fly”, are e.g. FeynArts/FormCalc [1,2], `HELAC-NLO` [3], GoSam [4], `aMC@NLO` [5], `BlackHat` [6], `NJET` [7], `OpenLoops` [8], `Recola` [9]. Public programs which contain a collection of pre-generated processes are e.g. `MCFM` [10], `VBFNLO` [11].

An important ingredient for such programs is an integral library containing the one-loop integrals which are the basic building blocks of any one-loop amplitude unless it is calculated purely numerically. Several libraries are publicly available to date: `FF` [12], `Looptools` [13], `QCDLoop` [14], `OneLOop` [15], `golem95C` [16,17], `PJFRY` [18]. A code for the calculation of one-loop four-point functions with complex masses (`DOC`) can be found in [19]. The latter has been integrated into the `LoopTools` library [13] where the complex versions of infrared finite integrals with less than four legs are already implemented. A complete set of scalar four-point integrals, both in dimensional and in mass regularization and valid also for complex masses can be found in [20] in analytic form.

Public programs dedicated to the reduction of multi-leg one-loop amplitudes at integrand level are e.g. `CutTools` [21] and `Samurai` [22,23]. A novel reduction algorithm based on integrand reduction through Laurent series expansion also has been developed [24].

The calculation of scalar one-loop integrals has a long tradition of pioneering work, see e.g. [25–31]. For processes involving unstable particles, these integrals are also required for complex internal masses, in order to be able to work within the so-called “complex-mass scheme” developed in Refs. [32,33]. An extension of the `golem95` library to complex masses has been presented in Ref. [17], called `golem95C`. The strategy in `golem95C` to avoid numerical instabilities due to small Gram determinants is to avoid the reduction to scalar basis integrals[1] in such cases, in favor of turning to the numerical evaluation of a convenient one-dimensional integral representation of the respective tensor integral. Recent new developments in this direction can be found in Ref. [34].

In this article, we present an extension of the `golem95C` library to integrals with tensor ranks r exceeding the number of propagators N. Such integrals occur for example in effective theories (a prominent example is the effective coupling of gluons to the Higgs boson), or in calculations within theories containing spin two particles beyond the leading order. Even though most of the modern methods to reduce one-loop amplitudes do not entirely rely on Feynman diagrams and tensor reduction anymore, the latter is still important in a number of cases. In particular, methods based on the reconstruction of tensor coefficients at integrand level [35,36] have proven very efficient recently [8,9]. Therefore, a tensor and scalar integral library represents an important tool for calculations within and beyond the Standard Model.

Further, tensor reduction, or tensorial reconstruction [36], can be an important “rescue system” for phase space points where cut-based techniques do not provide the desired accuracy. As such, the `golem95C` library is an important ingredient for the automated one-loop program GoSam [4]. The extension of cut based reduction methods at integrand level to higher ranks also has been tackled already in `Samurai`, called `XSamurai` [23]. The higher rank extensions of both `Samurai` and `golem95C` has been applied successfully in various phenomenological applications: the calculation of NLO corrections to Higgs plus two jet [37] and three jet [38] production in gluon fusion and NLO QCD corrections to the production of a jet plus a graviton decaying into two photons [39].

The higher rank integrals (closely related to integrals in more than $D = 4 - 2\epsilon$ space–time dimensions) can be written in terms of some “basis integrals” which can be called from the `golem95C` library presented here. However, the library contains the higher rank form factors in full generality. Therefore it can be used in various applications where the number of loop momenta in the integrand is large, in combination with integrand reduction methods ranging from the traditional tensor reduction to cut-based methods.

This article is organized as follows. In Section 2, we review the theoretical background, while in Section 3 we focus on the new extension of the library to integrals of higher ranks. Section 4 contains installation instructions, while in Section 5 the user can find simple examples of how to run the program. Section 6 contains our conclusions.

2. Theoretical background

The program is an update of the tensor and scalar integral library described in more detail in Ref. [16], based on the formalism developed in Refs. [40,41] to reduce tensor integrals to a convenient set of basis integrals. Similar reduction schemes can be found e.g. in Refs. [30,31,42–48]. Here, we will describe the theoretical

[1] The term “basis integrals” does not refer to a basis in the mathematical sense, as some elements can be linearly dependent, but rather denotes the integrals chosen to be the endpoints of the reduction.

framework only briefly and focus on the new features of the program.

2.1. Form factors

Tensor integrals can be divided into a part containing the Lorentz structure and a part consisting of scalar quantities, which we call *form factors*, denoted by $A^{N,r}_{j_1\ldots j_r}$, $B^{N,r}_{j_1\ldots j_{r-2}}$, $C^{N,r}_{j_1\ldots j_{r-4}}$, $D^{N,r}_{j_1\ldots j_{r-6}}$.

We define an N-point tensor integral of rank r in $D = 4 - 2\epsilon$ dimensions as

$$I_N^{D,\mu_1\ldots\mu_r}(a_1,\ldots,a_r) = \int \frac{d^Dq}{i\pi^{D/2}} \frac{q_{a_1}^{\mu_1}\ldots q_{a_r}^{\mu_r}}{(q_1^2-m_1^2+i\delta)\ldots(q_N^2-m_N^2+i\delta)} \tag{1}$$

where $q_a = q + r_a$, q is the loop momentum, and r_a is a combination of external momenta. Using the shift invariant vectors

$$\Delta_{ij}^{\mu} = r_i^{\mu} - r_j^{\mu}, \tag{2}$$

we can write down a general form factor decomposition of an arbitrary tensor integral

$$\begin{aligned} I_N^{D,\mu_1\ldots\mu_r}(a_1,\ldots,a_r;S) &= \sum_{j_1,\ldots,j_r\in S} \left[\Delta_{j_1\cdot}^{\cdot}\cdots\Delta_{j_r\cdot}^{\cdot}\right]^{\{\mu_1\ldots\mu_r\}}_{\{a_1\ldots a_r\}} A^{N,r}_{j_1\ldots j_r}(S) \\ &+ \sum_{j_1,\ldots,j_{r-2}\in S} \left[g^{\cdot\cdot}\Delta_{j_1\cdot}^{\cdot}\cdots\Delta_{j_{r-2}\cdot}^{\cdot}\right]^{\{\mu_1\ldots\mu_r\}}_{\{a_1\ldots a_r\}} B^{N,r}_{j_1\ldots j_{r-2}}(S) \\ &+ \sum_{j_1,\ldots,j_{r-4}\in S} \left[g^{\cdot\cdot}g^{\cdot\cdot}\Delta_{j_1\cdot}^{\cdot}\cdots\Delta_{j_{r-4}\cdot}^{\cdot}\right]^{\{\mu_1\ldots\mu_r\}}_{\{a_1\ldots a_r\}} C^{N,r}_{j_1\ldots j_{r-4}}(S) \\ &+ \sum_{j_1,\ldots,j_{r-4}\in S} \left[g^{\cdot\cdot}g^{\cdot\cdot}g^{\cdot\cdot}\Delta_{j_1\cdot}^{\cdot}\cdots\Delta_{j_{r-6}\cdot}^{\cdot}\right]^{\{\mu_1\ldots\mu_r\}}_{\{a_1\ldots a_r\}} D^{N,r}_{j_1\ldots j_{r-6}}(S) \\ &+ \cdots. \end{aligned} \tag{3}$$

The notation $[\cdots]^{\{\mu_1\cdots\mu_r\}}_{\{a_1\cdots a_r\}}$ stands for the distribution of the r Lorentz indices μ_i, and the momentum labels a_i, to the vectors $\Delta^{\mu_i}_{j\,a_i}$ and metric tensors in all distinguishable ways. Note that the choice $r_N = 0, a_i = N\ \forall i$ leads to the well known representation in terms of external momenta where the labels a_i are not necessary, but we prefer a completely shift invariant notation here.

S denotes an ordered set of propagator labels, corresponding to the momenta forming the kinematic matrix $\mathcal{S}$, defined by

$$\mathcal{S}_{ij} = (r_i - r_j)^2 - m_i^2 - m_j^2, \quad i,j \in \{1,\ldots,N\}. \tag{4}$$

We should point out that the form factors of type $D^{N,r}_{j_1,\ldots,j_{r-6}}$ and beyond, i.e. form factors associated with three or more metric tensors, are not needed for integrals where the rank r does not exceed the number N of propagators, no matter what the value of N is. This is because integrals with $N \geq 6$ can be reduced algebraically to pentagons, without generating higher dimensional remainder terms, using the following formula recursively [40]

$$I_N^{D,\mu_1\ldots\mu_r}(a_1,\ldots,a_r;S) = -\sum_{j\in S} \mathcal{C}^{\mu_r}_{ja_r} I_{N-1}^{D,\mu_1\ldots\mu_{r-1}}(a_1,\ldots,a_{r-1};S\setminus\{j\}) \quad (N\geq 6),$$

where $\mathcal{C}^{\mu}_{ja}$ is the solution of the equation

$$\sum_{j\in S} \mathcal{S}_{ij}\,\mathcal{C}^{\mu}_{j\,a} = \Delta^{\mu}_{i\,a}, \quad a\in S. \tag{5}$$

If $N \geq 7$, or in the case of exceptional kinematics, $\mathcal{S}$ is not invertible, so Eq. (5) does not have a unique solution. However, an explicit solution can be constructed as shown in [40,49]. In this sense the tensor reduction of N-point integrals with $N \geq 6$ is trivial: integrals with $N \geq 6$ can be reduced iteratively to 5-point integrals. Therefore, form factors for $N \geq 6$ are never needed. If the rank r of an N-point integral does not exceed N, Lorentz structures carrying at most two factors of $g^{\mu\nu}$ are sufficient, as the maximal form factor needed is $C_j^{5,5}$, corresponding to a rank five 5-point integral. However, for $r > N$, we can also have e.g. rank six 5-point integrals, which contain Lorentz structures involving three factors of $g^{\mu\nu}$, and therefore form factors which go beyond the types $A^{N,r}$, $B^{N,r}$, $C^{N,r}$ are needed.

The form factors are linear combinations of algebraic reduction coefficients, derived from the matrix $\mathcal{S}$, and N-point integrals with $N \leq 4$. Explicit expressions for $r \leq N$ are given in [40].

2.2. Integrals

The `golem95C` program uses the fact that tensor integrals are related to Feynman parameter integrals with Feynman parameters in the numerator.

The general relation between tensor integrals and parameter integrals with Feynman parameters in the numerator is well known [50,30,41]:

$$\begin{aligned} I_N^{D,\mu_1\ldots\mu_r}(a_1,\ldots,a_r;S) &= (-1)^r \sum_{m=0}^{[r/2]} \left(-\frac{1}{2}\right)^m \\ &\times \sum_{j_1\cdots j_{r-2m}=1}^{N} \left[(g^{\cdot\cdot})^{\otimes m}\,\Delta_{j_1\cdot}^{\cdot}\cdots\Delta_{j_r\cdot}^{\cdot}\right]^{\{\mu_1\cdots\mu_r\}}_{\{a_1\cdots a_r\}} \\ &\times I_N^{D+2m}(j_1,\ldots,j_{r-2m};S), \end{aligned} \tag{6}$$

where $I_N^{D+2m}(j_1,\ldots,j_{r-2m};S)$ is an integral with Feynman parameters in the numerator. $[r/2]$ stands for the nearest integer less or equal to $r/2$ and the symbol $\otimes m$ indicates that m powers of the metric tensor are present. Feynman parameter integrals corresponding to diagrams where propagators $l_1,\ldots,l_m$ are omitted with respect to the "maximal" topology can be defined as

$$\begin{aligned} I_N^D(j_1,\ldots,j_r;S\setminus\{l_1,\ldots,l_m\}) &= (-1)^N\Gamma\left(N-\frac{D}{2}\right) \\ &\times \int \prod_{i=1}^{N} dz_i\,\delta\left(1-\sum_{k=1}^{N} z_k\right)\delta(z_{l_1})\ldots\delta(z_{l_m})z_{j_1}\ldots z_{j_r}\left(R^2\right)^{D/2-N} \\ R^2 &= -\frac{1}{2}\sum_{i,j=1}^{N} z_i\mathcal{S}_{ij}z_j - i\delta. \end{aligned} \tag{7}$$

The program `golem95C` reduces the integrals internally to a set of basis integrals, i.e. the endpoints of the reduction. The choice of the basis integrals can have important effects on the numerical stability in certain kinematic regions. Our reduction endpoints are 4-point functions in 6 dimensions I_4^6, which are IR and UV finite, 4-point functions in $D+4$ dimensions, and various 2-point and 3-point functions, some of the latter with Feynman parameters in the numerator. This provides us with a convenient separation of IR and UV divergences, as the IR poles are exclusively contained in the triangle functions.

Note that I_3^{D+2} and I_4^{D+4} are UV divergent, while I_3^D can be IR divergent. In the code, the integrals are represented as arrays containing the coefficients of their Laurent expansion in $\epsilon = (4-D)/2$.

If the endpoints of the reduction contain finite scalar four-point integrals, the latter have not all been coded explicitly, but some are automatically called from the library `OneLOop` [15].

3. Extension of the program to higher rank

We focus here on the new features, for more details on the software components which are the same as in version 1.2, we refer to [16,17] and to the documentation contained in the program.

The main new feature consists in the extension of the program to calculate tensor integrals where the tensor rank is higher than the number of propagators. Such integrals are needed for example in processes containing the effective vertex coupling the Higgs boson to gluons in the large top mass limit, or in theories involving spin-2 particles. Examples of diagrams where integrals with rank $r = N + 1$ are needed are shown in Fig. 1, taken from an NLO calculation of diphoton + jet production via graviton exchange [39].

Another new feature of version 1.3 is the implementation of an improved formula for massive two-point functions in cases where the difference of the masses in the two propagators becomes small. This leads to much more stable results in the limit $m_1^2 \to m_2^2$. Further, we implemented an alternative reduction method for infrared divergent triangles, where $\det \mathcal{S} = 0$. While for $r \leq N$ the tensor integrals are calculated explicitly, as described in [16], for $r > N$ they are reduced by Passarino–Veltman reduction.

The case of complex masses already was supported in version 1.2 [17], however we would like to stress that complex masses are also fully supported for the higher rank integrals.

The version 1.2 of golem95C contained already an interface to integrand reduction methods, where the tensor coefficients in the numerator are reconstructed numerically [36]. This interface also has been extended to be able to deal with higher rank integrals. For this purpose, it is important to split the momenta in the numerator into a 4-dimensional part and a $(D-4)$-dimensional part, $k^\mu_{(D)} = \hat{k}^\mu_{(4)} + \tilde{k}^\mu_{(-2\epsilon)}$, $k^2_{(D)} = \hat{k}^2 + \tilde{k}^2$. This splitting leads to integrals with powers of $\tilde{k}^2$ in the numerator, where in the case of higher rank integrals, additional $\tilde{k}^2$-integrals are needed. These integrals are given in Appendix A in two forms, a general analytic form, as well as explicit forms as implemented in the program. The most general form, to the best of our knowledge, has not been published before, so it may also be useful as a reference for related calculations in the future.

The call syntax for the higher rank form factors is analogous to the one for $r \leq N$, where the form factors defined in Eq. (3) have been extended to include higher ranks. The code is built such that the reduction procedure is valid for arbitrary ranks. Form factors up to rank ten for $N \leq 4$ and up to rank six for $N = 5$ have been hardcoded, additional ones can be easily generated using the python script `gen_form_factors.py` in the subdirectory `tool/highrank`. Explicit examples how to call the form factor for a rank four triangle, a rank five box and a rank six pentagon are given in Section 5.

4. Installation

The program can be downloaded as golem95-1.3.0.tar.gz from the following URL: `http://golem.hepforge.org/95/`. The installation instructions given below also can be found in the `README` file coming with the code.

The installation setup is based on autotools [51]. To install the `golem95C` library, type the following commands:

```
./configure [--prefix=mypath]
[--precision=quadruple] [FC=compiler]
make
make install
```

The `--prefix` option denotes the installation prefix, under which the directories `lib/` and `include/` are generated. If no option is given, on a Linux system the configure script would choose `prefix=/usr/local`. The argument `--precision` selects double or quadruple precision to be used in the library; it should be noted that quadruple precision is not supported by all Fortran compilers and that `precision=double` is the default value. If the variable FC is not set the first Fortran compiler which is automatically detected will be used. Another variable commonly used is `FCFLAGS` which allows one to pass compiler flags to the Fortran compiler.

As an alternative to the call of finite box integrals from `OneLOop` [15], it is possible to call finite scalar box and triangle integrals with internal masses from `LoopTools` [1]. To use this option, the user should (a) install `LoopTools`, and (b) use the option `--with-looptools=path_to_libooptools.a` for the configure script, i.e. type the following commands:

```
./configure [--prefix=mypath]
[--with-looptools=path_to/libooptools.a]
[--precision=quadruple] [FC=compiler]
[F77= fortran77compiler]
make
make install.
```

5. Usage and examples

Examples for the usage of the program can be found in the subdirectory `demos`. The examples for the higher rank form factors are also described below.

The basic structure of a program using the calculation of N-point scalar integrals or form factors by `golem95C` is:

`call initgolem95(N)`

…fill kinematic matrix $\mathcal{S}$ …

`call preparesmatrix()`

…evaluate integrals/form factors …

`call exitgolem95()`

A simple example for the calculation of a scalar three-point integral, which is evaluated by calling the form factor `A30`, is given in Listing 1.

Listing 1: example of a program calling a scalar three-point integral.

```
program main

  use precision_golem ! to get the type ki (for real and complex)
  use matrice_s       ! needed for initgolem95, s_mat, etc.
  use constante       ! contains useful constants
  use form_factor_type
  use parametre       ! contains default parameter settings
  use form_factor_3p ! module containing the three-point form factors
  implicit none

  type(form_factor) :: res
  real(ki) :: s1, s2, s3, m1sq, m2sq, m3sq

  s1 = 1._ki
  s2 = 0._ki
  s3 = 0._ki
  m1sq = 2._ki
  m2sq = 0._ki
  m3sq = 0._ki

  call initgolem95(3) ! initialization of caching system and 3x3 matrix S

  ! definition of the kinematic matrix S

  s_mat(1,:) = (/ -m1sq*2._ki , s2-m1sq-m2sq, s1-m1sq-m3sq /)
  s_mat(2,:) = (/ s2-m1sq-m2sq, -m2sq*2._ki , s3-m2sq-m3sq /)
  s_mat(3,:) = (/ s1-m1sq-m3sq, s3-m2sq-m3sq, -m3sq*2._ki /)

  call preparesmatrix()

  ! call the scalar triangle

  res = a30(s_null)

  write (6,*) 'result='
  write (6,'("   1/epsilon^2 * (",e16.10,1x,"+ I*",1x,e16.10,")")') real(res%a,ki),aimag(res%a)
  write (6,'("+ 1/epsilon   * (",e16.10,1x,"+ I*",1x,e16.10,")")') real(res%b,ki),aimag(res%b)
  write (6,'("+ 1           * (",e16.10,1x,"+ I*",1x,e16.10,")")') real(res%c,ki),aimag(res%c)

  call exitgolem95()

end program main
```

5.1. Three-point rank 4 example

The executable for this example is created by `make demo_3point`. The program `demos/ demo_three_point.f90` contains as an option the call to a rank four three-point form factor,

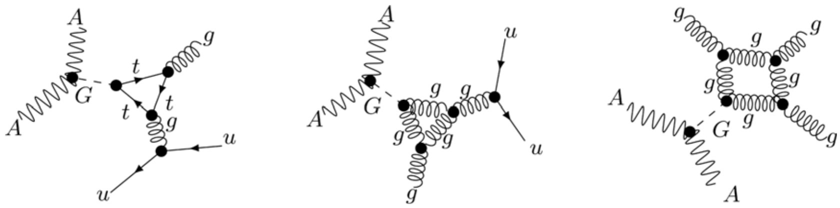

Fig. 1. Examples of diagrams with graviton exchange, where rank four 3-point and rank five 4-point integrals are needed. The graviton is denoted by G.

$A34(1, 1, 2, 3, S)$. The integers in the argument list denote the labels of the corresponding momenta in the form factor representation, i.e. $A^{3,4}_{1123}$ in the notation of Eq. (3), denoting the coefficient of the term $\Delta^{\mu1}_{1\,a_1}\Delta^{\mu2}_{1\,a_2}\Delta^{\mu3}_{2\,a_3}\Delta^{\mu4}_{3\,a_4}$ in the form factor decomposition of the Lorentz structure. Note that using $a_j = 3$ and $r_3 = 0$, $\Delta_{i\,a_j}$ is simply replaced by r_i. The call syntax is the same as for the lower rank form factors. The results are written to the file `test3point.txt`. As a check for the user, the results to be obtained when calling $A34(1, 1, 2, 3, S)$ are listed in `table_of_results_3point_option_n.txt`.

The option n in `table_of_results_3point_option_n.txt` denotes different choices for the kinematics of the three-point function. Running the executable `demo_3point`, the user is prompted to give (a) the option for the kinematics and (b) the option for the rank (or other features). The higher rank three-point example corresponds to option (b) number 7.

5.2. Four-point rank 5 example

The executable for this example is created by `make demo_4point`. The program `demos/ demo_four_point.f90` contains as an option the call to a rank five four-point form factor, $A45(1, 2, 2, 3, 3, S)$. The integers in the argument list denote the labels of the corresponding momenta in the form factor representation, i.e. $A^{4,5}_{12233}$ in the notation of Eq. (3), denoting the coefficient of the term $\Delta^{\mu1}_{1\,a_1}\Delta^{\mu2}_{2\,a_2}\Delta^{\mu3}_{2\,a_3}\Delta^{\mu4}_{3\,a_4}\Delta^{\mu5}_{3\,a_5}$ in the form factor decomposition of the Lorentz structure. The call syntax is the same as for the lower rank form factors. The results are written to the file `test4point.txt`. As a check for the user, the results to be obtained when calling $A45(1, 2, 2, 3, 3, S)$ are listed in `table_of_results_4point_option_n.txt`, where again n denotes different choices for the kinematics. Once the kinematics is chosen, the rank five 4-point example corresponds to option (b) number 9.

5.3. Five-point rank 6 example

The executable for this example is created by `make demo_5point`. The program `demos/demo_five_point.f90` contains as an option the call to a rank six five-point form factor, $A56(1, 1, 2, 3, 4, 5, S)$. The integers in the argument list denote the labels of the corresponding momenta in the form factor representation, i.e. $A^{5,6}_{112345}$ in the notation of Eq. (3), denoting the coefficient of the term $\Delta^{\mu1}_{1\,a_1}\Delta^{\mu2}_{1\,a_2}\Delta^{\mu3}_{2\,a_3}\Delta^{\mu4}_{3\,a_4}\Delta^{\mu5}_{4\,a_5}\Delta^{\mu6}_{5\,a_6}$ in the form factor decomposition of the Lorentz structure. The call syntax is the same as for the lower rank form factors. The results are written to the file `test5point.txt`. As a check for the user, the results to be obtained when calling $A56(1, 1, 2, 3, 4, 5, S)$ (option 5 in `demo_five_point.f90`) are listed in `table_of_results_5point.txt`.

6. Conclusions

We have presented an extension of the program `golem95C` which provides a library of scalar and tensor integrals where the tensor rank r, denoting the number of loop momenta in the numerator, can exceed the number of propagators N. We have implemented ranks up to $r = 10$ for $N \leq 4$, and 5-point and 6-point functions up to rank $r = N + 1$. The use of both real or complex masses is possible within the same setup. The program, which is an extension of an earlier version of the `golem95C` library, now also can be used in the presence of effective vertices and in models where the presence of spin two particles can require integrals with higher ranks than usually needed in renormalizable theories. Due to an appropriate interface, the program can be used both within a traditional tensor reduction approach as well as within a unitarity-inspired numerical reconstruction of the integrand at the tensorial level. The program is publicly available at http://golem.hepforge.org/95/.

Acknowledgments

We would like to thank the GoSam collaboration for feedback on `golem95C`, and in particular Gavin Cullen for numerous tests and helpful discussions, Edoardo Mirabella for useful comments, and Hans van Deurzen, Gionata Luisoni and Sadok Zidi for comparisons.

Appendix. $\tilde{k}$-integrals

Here we give a list of integrals which are needed if the momenta in the numerator are split into a 4-dimensional and a $(D-4)$-dimensional part, according to $k^\mu_{(D)} = \hat{k}^\mu_{(4)} + \tilde{k}^\mu_{(-2\epsilon)}$, $k^2_{(D)} = \hat{k}^2 + \tilde{k}^2$. Using $\hat{q}_a = \hat{k} + r_a$, where $\hat{k}$ is the loop momentum in 4 dimensions, and r_a is a combination of external momenta, we define

$$I_N^{D,\alpha;\mu_1\ldots\mu_r}(a_1,\ldots,a_r;S) \equiv \int \frac{\mathrm{d}^D k}{i\pi^{D/2}} \frac{\left(\tilde{k}^2\right)^\alpha \hat{q}^{\mu_1}_{a_1}\cdots\hat{q}^{\mu_r}_{a_r}}{\prod_{j=1}^{N}(q_j^2 - m_j^2 + i\delta)}. \tag{A.1}$$

The integrals with additional powers of $(\tilde{k}^2)^\alpha$ in the numerator are related to integrals in higher dimensions by

$$\begin{aligned} I_N^{D,\alpha;\mu_1\ldots\mu_r}&(a_1,\ldots,a_r;S) \\ &= (-1)^{r+\alpha}\frac{\Gamma(\alpha+D/2-2)}{\Gamma(D/2-2)}\sum_{m=0}^{\lfloor r/2\rfloor}\left(-\frac{1}{2}\right)^m \\ &\quad\times\sum_{j_1,\ldots,j_{r-2m}=1}^{N}[\underbrace{\hat{g}^{\cdot\cdot}\cdots\hat{g}^{\cdot\cdot}}_{m}\Delta^{\cdot}_{j_1\cdot}\cdots\Delta^{\cdot}_{j_{r-2m}\cdot}]^{\mu_1\ldots\mu_r}_{a_1\ldots a_r} \\ &\quad\times I_N^{D+2\alpha+2m}(j_1,\ldots,j_{r-2m};S), \end{aligned} \tag{A.2}$$

where $\Delta^\mu_{ij} = r^\mu_i - r^\mu_j$. Note that for $D = 4 - 2\epsilon$, the prefactor in Eq. (A.2) reads

$$\frac{\Gamma(\alpha+D/2-2)}{\Gamma(D/2-2)} = \frac{\Gamma(\alpha-\epsilon)}{\Gamma(-\epsilon)} = -\epsilon(\alpha-1)! + \mathcal{O}(\epsilon^2). \tag{A.3}$$

Therefore, integrals with $\alpha > 0$ will only contribute if the integrals $I_N^{D+2\alpha+2m}$ are UV divergent, since we can drop terms of $\mathcal{O}(\epsilon)$

for one-loop applications. The coefficient of the UV pole of these integrals, which is projected out in this way, will contribute to the so-called "rational part" of an amplitude. For $D = 4 - 2\epsilon$, the integral $I_N^{D+2\alpha+2m}$ will be proportional to $\Gamma(\epsilon-\eta)$, with $\eta = 2-N+\alpha+m$. Therefore, it will contain a UV divergence if $\eta \geq 0$.

The results for those integrals which are relevant for the rational part can be given in a general form [52].

$$\epsilon I_N^{D+2\alpha+2m}(l_1,\ldots,l_r;S) = \frac{(-1)^N}{2^\eta \eta!}\sum_{j_1,\ldots,j_{2\eta}=1}^{N} \mathcal{S}_{j_1j_2}\cdots\mathcal{S}_{j_{2\eta-1}j_{2\eta}} P_N(l_1,\ldots,l_r,j_1,\ldots,j_{2\eta})$$
$$\eta = 2-N+\alpha+m, \tag{A.4}$$

where we define

$$P_{t_1,t_2,\ldots,t_N} = \frac{\prod_{j=1}^{N}(t_j!)}{\left(N-1+\sum_{i=1}^{N} t_i\right)!} \tag{A.5}$$

to arrive at $P_N(j_1,\ldots,j_s)$, which counts the indices in an expression:

$$P_N(j_1,\ldots,j_s) = P_{\left(\sum_{i=1}^{s}\delta_{1,j_i}\right),\ldots,\left(\sum_{i=1}^{s}\delta_{N,j_i}\right)}. \tag{A.6}$$

Further, $\mathcal{S}_{j_1j_2}$ denotes an element of the kinematic matrix $\mathcal{S}$.

The integrals $I_N^{D+2\alpha+2m}$ will be UV divergent for $2\alpha + 2m \geq 2N - 4$. Below give the relevant explicit expressions, derived from Eq. (A.4). Terms which will be of order $\mathcal{O}(\epsilon)$ are dropped.

$$\epsilon I_N^{D-4+2N}(S) = \frac{(-1)^N}{(N-1)!} \tag{A.7a}$$

$$\epsilon I_N^{D-4+2N}(l_1;S) = \frac{(-1)^N}{N!} \tag{A.7b}$$

$$\epsilon I_N^{D-4+2N}(l_1,l_2;S) = \frac{(-1)^N}{(N+1)!}\left(1+\delta_{l_1l_2}\right) \tag{A.7c}$$

$$\epsilon I_N^{D-4+2N}(l_1,l_2,l_3;S) = \frac{(-1)^N}{(N+2)!} \times \left(1+\delta_{l_1l_2}+\delta_{l_1l_3}+\delta_{l_2l_3}+2\delta_{l_1l_2}\delta_{l_2l_3}\right) \tag{A.7d}$$

$$\begin{aligned}\epsilon I_N^{D-4+2N}(l_1,l_2,l_3,l_4;S) = \frac{(-1)^N}{(N+3)!} &\times (\delta_{l_1l_2}(6\delta_{l_1l_3}\delta_{l_2l_4}+2\delta_{l_1l_3}+2\delta_{l_2l_4}+\delta_{l_3l_4})\\ &+2\delta_{l_3l_4}(\delta_{l_1l_3}+\delta_{l_2l_4})+\delta_{l_1l_3}\delta_{l_2l_4}+\delta_{l_1l_4}\delta_{l_2l_3}\\ &+\delta_{l_1l_2}+\delta_{l_1l_3}+\delta_{l_1l_4}+\delta_{l_2l_3}+\delta_{l_2l_4}+\delta_{l_3l_4}+1)\end{aligned} \tag{A.7e}$$

$$\epsilon I_N^{D-4+2(N+1)}(S) = \frac{(-1)^N}{2(N+1)!}\left(\sum_{j_1,j_2=1}^{N}\mathcal{S}_{j_1j_2} + tr\,\{\mathcal{S}\}\right) \tag{A.7f}$$

$$\epsilon I_N^{D-4+2(N+1)}(l_1;S) = \frac{(-1)^N}{2(N+2)!}\sum_{j_1,j_2=1}^{N}\mathcal{S}_{j_1j_2}\left(1+\delta_{j_1j_2}\right) \times \left(1+\delta_{l_1j_1}+\delta_{l_1j_2}\right) \tag{A.7g}$$

$$\begin{aligned}\epsilon I_N^{D-4+2(N+1)}(l_1,l_2;S) = \frac{(-1)^N}{2(N+3)!}\sum_{j_1,j_2=1}^{N}\mathcal{S}_{j_1j_2}\\ \times (\delta_{j_1j_2}(6\delta_{j_1l_1}\delta_{j_2l_2}+2\delta_{j_1l_1}+2\delta_{j_2l_2}+\delta_{l_1l_2})\\ +2\delta_{l_1l_2}(\delta_{j_1l_1}+\delta_{j_2l_2})+\delta_{j_1l_1}\delta_{j_2l_2}+\delta_{j_1l_2}\delta_{j_2l_1}\\ +\delta_{j_1j_2}+\delta_{j_1l_1}+\delta_{j_1l_2}+\delta_{j_2l_1}+\delta_{j_2l_2}+\delta_{l_1l_2}+1).\end{aligned} \tag{A.7h}$$

We also give here the explicit formulas for rank 6 pentagon integrals involving $\tilde{k}^2$ terms, because they are special to higher rank extensions. Again, terms of $\mathcal{O}(\epsilon)$ are dropped.

$$I_5^{D,3}(S) = \int \frac{d^Dk}{i\pi^{D/2}}\frac{\left(\tilde{k}^2\right)^3}{\prod_{j=1}^{5}(q_j^2-m_j^2+i\delta)} = -\frac{1}{12} \tag{A.8}$$

$$I_5^{D,2;\mu_1\mu_2}(a_1,a_2;S) = \int \frac{d^Dk}{i\pi^{D/2}}\frac{\left(\tilde{k}^2\right)^2\hat{q}_{a_1}^{\mu_1}\hat{q}_{a_2}^{\mu_2}}{\prod_{j=1}^{5}(q_j^2-m_j^2+i\delta)} = -\frac{1}{48}g^{\mu_1\mu_2} \tag{A.9}$$

$$I_5^{D,1;\mu_1\cdots\mu_4}(a_1,\ldots,a_4;S) = \int \frac{d^Dk}{i\pi^{D/2}}\frac{\tilde{k}^2\hat{q}_{a_1}^{\mu_1}\ldots\hat{q}_{a_4}^{\mu_4}}{\prod_{j=1}^{5}(q_j^2-m_j^2+i\delta)} = -\frac{1}{96}\left[g^{\mu_1\mu_2}g^{\mu_3\mu_4}+g^{\mu_1\mu_3}g^{\mu_2\mu_4}+g^{\mu_1\mu_4}g^{\mu_2\mu_3}\right]. \tag{A.10}$$

As another example, we give the expressions for the rational part of box integrals in $D+6$ dimensions, needed for rank six four-point functions, which can be obtained from (A.7f):

$$\epsilon I_4^{D+6}(S) = \frac{1}{240}\left(\sum_{i,j=1}^{4}(\Delta_{ij}^2-m_i^2-m_j^2)-2\sum_{i=1}^{4}m_i^2\right) + \mathcal{O}(\epsilon). \tag{A.11}$$

References

[1] T. Hahn, M. Perez-Victoria, Automatized one-loop calculations in four and D dimensions, Comput. Phys. Commun. 118 (1999) 153–165.
[2] S. Agrawal, T. Hahn, E. Mirabella, FormCalc 7.5. *PoS*, LL2012 (2012) 046.
[3] G. Bevilacqua, M. Czakon, M.V. Garzelli, A. van Hameren, A. Kardos, et al., HELAC-NLO, Comput. Phys. Commun. 184 (2013) 986–997.
[4] G. Cullen, N. Greiner, G. Heinrich, G. Luisoni, P. Mastrolia, et al., Automated One-Loop Calculations with GoSam, Eur. Phys. J. C72 (2012) 1889.
[5] V. Hirschi, R. Frederix, S. Frixione, M.V. Garzelli, F. Maltoni, et al., Automation of one-loop QCD corrections, J. High Energy Phys. 1105 (2011) 044.
[6] C.F. Berger, Z. Bern, L.J. Dixon, F. Febres Cordero, D. Forde, et al., An automated implementation of on-shell methods for one-loop amplitudes, Phys. Rev. D78 (2008) 036003.
[7] S. Badger, B. Biedermann, P. Uwer, V. Yundin, NLO QCD corrections to multi-jet production at the LHC with a centre-of-mass energy of $\sqrt{s} = 8$ TeV, Phys. Lett. B718 (2013) 965–978.
[8] F. Cascioli, P. Maierhöfer, S. Pozzorini, Scattering amplitudes with open loops, Phys. Rev. Lett. 108 (2012) 111601.
[9] S. Actis, A. Denner, L. Hofer, A. Scharf, S. Uccirati, Recursive generation of one-loop amplitudes in the Standard Model, J. High Energy Phys. 1304 (2013) 037.
[10] J. Campbell, R.K. Ellis, Next-to-leading order corrections to W + 2jet and Z + 2jet production at hadron colliders, Phys. Rev. D65 (2002) 113007.
[11] K. Arnold, et al., VBFNLO: a parton level Monte Carlo for processes with electroweak bosons, Comput. Phys. Commun. 180 (2009) 1661–1670.
[12] G.J. van Oldenborgh, FF: a Package to evaluate one loop Feynman diagrams, Comput. Phys. Commun. 66 (1991) 1–15.
[13] T. Hahn, Feynman Diagram Calculations with FeynArts, FormCalc, and LoopTools. *PoS*, ACAT2010 (2010) 078.
[14] R.K. Ellis, G. Zanderighi, Scalar one-loop integrals for QCD, J. High Energy Phys. 02 (2008) 002.
[15] A. van Hameren, OneLOop: for the evaluation of one-loop scalar functions, Comput. Phys. Commun. 182 (2011) 2427–2438.
[16] T. Binoth, J.Ph. Guillet, G. Heinrich, E. Pilon, T. Reiter, Golem95: a numerical program to calculate one-loop tensor integrals with up to six external legs, Comput. Phys. Commun. 180 (2009) 2317–2330.
[17] G. Cullen, J.Ph. Guillet, G. Heinrich, T. Kleinschmidt, E. Pilon, et al., Golem95C: A library for one-loop integrals with complex masses, Comput. Phys. Commun. 182 (2011) 2276–2284.
[18] J. Fleischer, T. Riemann, V. Yundin, PJFry: A C++ package for tensor reduction of one-loop Feynman integrals, 2011.
[19] Dao Thi Nhung, Le Duc Ninh, D0C: a code to calculate scalar one-loop four-point integrals with complex masses, Comput. Phys. Commun. 180 (2009) 2258–2267.
[20] A. Denner, S. Dittmaier, Scalar one-loop 4-point integrals, Nuclear. Phys. B844 (2011) 199–242.

[21] G. Ossola, C.G. Papadopoulos, R. Pittau, CutTools: a program implementing the OPP reduction method to compute one-loop amplitudes, J. High Energy Phys. 03 (2008) 042.
[22] P. Mastrolia, G. Ossola, T. Reiter, F. Tramontano, Scattering AMplitudes from unitarity-based reduction algorithm at the integrand-level, J. High Energy Phys. 08 (2010) 080.
[23] H. van Deurzen, Associated Higgs production at NLO with GoSam, Acta Phys. Polon. B44 (11) (2013) 2223.
[24] P. Mastrolia, E. Mirabella, T. Peraro, Integrand reduction of one-loop scattering amplitudes through Laurent series expansion, J. High Energy Phys. 1206 (2012) 095.
[25] G.J. van Oldenborgh, J.A.M. Vermaseren, New algorithms for one loop integrals, Z. Phys. C46 (1990) 425–438.
[26] G. 't Hooft, M.J.G. Veltman, Scalar one loop integrals, Nuclear Phys. B153 (1979) 365–401.
[27] K. Fabricius, I. Schmitt, Calculation of dimensionally regularized box graphs in the zero mass case, Z. Phys. C3 (1979) 51–53.
[28] W. Beenakker, A. Denner, Infrared divergent scalar box integrals with applications in the electroweak standard model, Nuclear Phys. B338 (1990) 349–370.
[29] A. Denner, U. Nierste, R. Scharf, A Compact expression for the scalar one loop four point function, Nuclear Phys. B367 (1991) 637–656.
[30] Z. Bern, L.J. Dixon, D.A. Kosower, Dimensionally regulated one loop integrals, Phys. Lett. B302 (1993) 299–308.
[31] Z. Bern, L.J. Dixon, D.A. Kosower, Dimensionally regulated pentagon integrals, Nuclear Phys. B412 (1994) 751–816.
[32] A. Denner, S. Dittmaier, M. Roth, D. Wackeroth, Predictions for all processes $e^+e^- \to 4$ fermions + gamma, Nuclear Phys. B560 (1999) 33–65.
[33] A. Denner, S. Dittmaier, M. Roth, L.H. Wieders, Electroweak corrections to charged-current $e^+e^- \to 4$ fermion processes: technical details and further results, Nuclear Phys. B724 (2005) 247–294.
[34] J.Ph. Guillet, E. Pilon, M. Rodgers, M.S. Zidi, Stable one-dimensional integral representations of one-loop N-point functions in the general massive case: i - three point functions, J. High Energy Phys. 1311 (2013) 154.
[35] A. van Hameren, Multi-gluon one-loop amplitudes using tensor integrals, J. High Energy Phys. 0907 (2009) 088.
[36] G. Heinrich, G. Ossola, T. Reiter, F. Tramontano, Tensorial reconstruction at the integrand level, J. High Energy Phys. 1010 (2010) 105.
[37] H. van Deurzen, N. Greiner, G. Luisoni, P. Mastrolia, E. Mirabella, et al., NLO QCD corrections to the production of Higgs plus two jets at the LHC, Phys. Lett. B721 (2013) 74–81.
[38] G. Cullen, H. van Deurzen, N. Greiner, G. Luisoni, P. Mastrolia, et al., NLO QCD corrections to Higgs boson production plus three jets in gluon fusion, Phys. Rev. Lett. 111 (2013) 131801.
[39] N. Greiner, G. Heinrich, J. Reichel, J.F. von Soden-Fraunhofen, NLO QCD corrections to diphoton plus jet production through graviton exchange, J. High Energy Phys. 1311 (2013) 028.
[40] T. Binoth, J.Ph. Guillet, G. Heinrich, E. Pilon, C. Schubert, An algebraic/numerical formalism for one-loop multi-leg amplitudes, J. High Energy Phys. 10 (2005) :015. hep-ph/0504267 version 3.
[41] T. Binoth, J.P. Guillet, G. Heinrich, Reduction formalism for dimensionally regulated one-loop N-point integrals, Nuclear Phys. B572 (2000) 361–386.
[42] G. Duplancic, B. Nizic, Reduction method for dimensionally regulated one-loop N-point Feynman integrals, Eur. Phys. J. C35 (2004) 105–118.
[43] W.T. Giele, E.W.N. Glover, A calculational formalism for one-loop integrals, J. High Energy Phys. 04 (2004) 029.
[44] F. del Aguila, R. Pittau, Recursive numerical calculus of one-loop tensor integrals, J. High Energy Phys. 07 (2004) 017.
[45] A. van Hameren, J. Vollinga, S. Weinzierl, Automated computation of one-loop integrals in massless theories, Eur. Phys. J. C41 (2005) 361–375.
[46] A. Denner, S. Dittmaier, Reduction schemes for one-loop tensor integrals, Nuclear Phys. B734 (2006) 62–115.
[47] J. Fleischer, T. Riemann, A complete algebraic reduction of one-loop tensor Feynman integrals, Phys. Rev. D83 (2011) 073004.
[48] J. Fleischer, T. Riemann, A solution for tensor reduction of one-loop N-point functions with $N \geq 6$, Phys. Lett. B707 (2012) 375–380.
[49] J. Fleischer, T. Riemann, V. Yundin, New results for algebraic tensor reduction of Feynman integrals, PoS RADCOR2011 (2011) 011.
[50] A.I. Davydychev, A simple formula for reducing Feynman diagrams to scalar integrals, Phys. Lett. B263 (1991) 107–111.
[51] G.V. Vaughan, B. Elliston, T. Tromey, I.L. Taylor, GNU Autoconf, Automake, and Libtool: expert insight into porting software and building large projects using GNU Autotools, New Riders, Indianapolis, 2000.
[52] Thomas Reiter, Automated evaluation of one-loop six-point processes for the LHC., Ph.D. Thesis, 2009, arXiv:0903.0947.