THE EUROPEAN
PHYSICAL JOURNAL C

CrossMark

Numerical implementation of the loop–tree duality method

Sebastian Buchta[1,a], **Grigorios Chachamis**[2,b] , **Petros Draggiotis**[3,c], **Germán Rodrigo**[1,d]

[1] Instituto de Física Corpuscular, Universitat de València-Consejo Superior de Investigaciones Científicas, Parc Científic, 46980 Paterna, Valencia, Spain
[2] Instituto de Física Teórica UAM/CSIC, Universidad Autónoma de Madrid, C/ Nicolás Cabrera 15, 28049 Madrid, Spain
[3] Institute of Nuclear and Particle Physics, NCSR "Demokritos", 15310 Agia Paraskevi, Greece

Abstract We present a first numerical implementation of the loop–tree duality (LTD) method for the direct numerical computation of multi-leg one-loop Feynman integrals. We discuss in detail the singular structure of the dual integrands and define a suitable contour deformation in the loop three-momentum space to carry out the numerical integration. Then we apply the LTD method to the computation of ultraviolet and infrared finite integrals, and we present explicit results for scalar and tensor integrals with up to eight external legs (octagons). The LTD method features an excellent performance independently of the number of external legs.

1 Introduction

The recent discovery of the Higgs boson at the LHC represents a great success of the standard model (SM) of elementary particles. With the new run started in 2015 the primary goal is to study its properties in detail and to detect possible extensions to the SM. Precise theory predictions are needed to achieve this goal, which calls for calculations at the next-to-leading order (NLO) and beyond for multi-leg processes.

Computing higher-order corrections in quantum field theory (QFT), in particular in QCD and in the EW sector of the SM is highly challenging. The complexity increases as the number of external particles gets bigger and the order of the perturbative expansion. The task is far from trivial and each step presents its own difficulties: one needs first to generate the virtual and real scattering amplitudes, then carry out the integration over the loop momenta for the virtual contribution and finally perform the phase-space integration for both real and virtual corrections after taking proper care so

that the infrared divergencies cancel. In particular, infrared singularities of the virtual contribution can be subtracted by using appropriate semi-analytical terms and combine them with the ones stemming from the real corrections to produce finite results [1]. Purely numerical approaches to the integration of loop momenta have been discussed extensively in the literature [2–14]. The generation of amplitudes and calculation of cross sections at one loop has seen great progress in recent years and algorithmic calculations at NLO have been automated are now considered standardized, based on purely numerical [15–17] and a mix of analytical and numerical approaches [18–20]. Substantial progress has also been made at higher orders [21–23].

The loop–tree duality (LTD) method [24–37] establishes that generic loop quantities (loop integrals and scattering amplitudes) in any relativistic, local and unitary field theory can be written as a sum of tree-level-like objects obtained after making all possible cuts to the internal lines of the corresponding Feynman diagrams, with one single cut per loop and integrated over a measure that closely resembles the phase space of the corresponding real corrections [24,25]. This duality relation is realized by a modification of the customary $+i0$ prescription of the Feynman propagators and encodes the causal structure of the scattering amplitudes in the expected way. The analysis of the singular behaviour of one-loop integrals and scattering amplitudes in this framework at the integrand level in the loop-momentum space shows that there is a partial cancellation of singularities among different dual contributions such that physical infrared and threshold singularities remain restricted to a compact region of the loop three-momentum [28–30]. This feature opens up the possibility that virtual and real radiative corrections can be brought together under a common integral and be treated simultaneously with Monte Carlo techniques though a convenient mapping of the momenta entering the virtual and real scattering amplitudes [33–36].

[a] e-mail: sbuchta@ific.uv.es

[b] e-mail: chachamis@gmail.com

[c] e-mail: petros.draggiotis@gmail.com

[d] e-mail: german.rodrigo@csic.es

In this work, we present a first numerical implementation of the LTD method and we apply it to the computation of multi-leg one-loop scalar and tensor integrals. The outline of the paper is as follows. In Sect. 2 we review the LTD method at one loop and discuss the singular behaviour of the dual integrand in the loop-momentum space. In Sect. 3 we introduce the contour deformation in the loop three-momentum space which is used for the numerical loop integration. We present explicit numerical results for various external momenta configurations, in Sect. 4 for scalar integrals up to pentagons and in Sect. 5 for up to rank-three tensor integrals with five and six external legs along with some non-trivial heptagon and octagon examples. Finally, our conclusions and outlook are presented in Sect. 6.

2 Loop–tree duality at one loop

We consider a general one-loop N-leg scalar integral (see Fig. 1) in dimensional regularization, with d the number of space-time dimensions,

$$L^{(1)}(p_1, p_2, \ldots, p_N) = \int_\ell \prod_{i \in \alpha_1} G_{\mathrm{F}}(q_i),$$

$$\int_\ell \bullet = -i \, \mu^{4-d} \int \frac{d^d \ell}{(2\pi)^d} \, \bullet, \tag{1}$$

where

$$G_{\mathrm{F}}(q_i) = \frac{1}{q_i^2 - m_i^2 + i0} \tag{2}$$

are Feynman propagators that depend on the loop momentum ℓ, which flows anti-clockwise, and the four-momenta of the external legs p_i, $i \in \alpha_1 = \{1, 2, \ldots N\}$, which are taken as outgoing and are clockwise ordered. The momenta of the internal lines are denoted as $q_{i,\mu} = (q_{i,0}, \mathbf{q}_i)$, where $q_{i,0}$ is the energy (time component) and $\mathbf{q}_i$ are the spatial components, which are defined as $q_i = \ell + k_i$ with $k_i = p_1 + \cdots + p_i$,
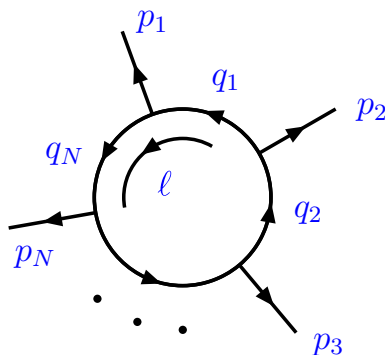

Fig. 1 Momentum configuration of the one-loop N-point integral

and $k_N = 0$ by momentum conservation. We also define $k_{ji} = q_j - q_i$, which, in fact, is independent of the loop momentum ℓ.

The corresponding dual representation of the scalar integral in Eq. (1) is obtained from the loop–tree duality (LTD) theorem [24]:

$$L^{(1)}(p_1, p_2, \ldots, p_N) = -\sum_{i \in \alpha_1} \int_\ell \tilde{\delta}(q_i) \prod_{\substack{j \in \alpha_1 \\ j \neq i}} G_{\mathrm{D}}(q_i; q_j), \tag{3}$$

where

$$G_{\mathrm{D}}(q_i; q_j) = \frac{1}{q_j^2 - m_j^2 - i0 \, \eta k_{ji}}, \tag{4}$$

are dual propagators, η is an arbitrary future-like vector, i.e., a d–dimensional vector that can be either light-like ($\eta^2 = 0$) or time-like ($\eta^2 > 0$) with positive definite energy $\eta_0 \geq 0$, and

$$\tilde{\delta}(q_i) \equiv 2\pi \, i \, \theta(q_{i,0}) \, \delta(q_i^2 - m_i^2), \tag{5}$$

selects the internal loop on-shell modes, $G_{\mathrm{F}}^{-1}(q_i) = 0$, with positive definite energy, $q_{i,0} \geq 0$. Hence, the LTD theorem expresses the usual loop Feynman integral, Eq. (1), as a sum of single-cut phase-space integrals, Eq. (3), with

$$\int_\ell \tilde{\delta}(q_i) \tag{6}$$

as the single-particle phase-space integration measure. The LTD theorem is valid not only for scalar one-loop integrals, but it can straightforwardly be extended to deal with scattering amplitudes [24] and higher orders of the perturbative expansion [26,27].

The integrand of the dual representation of one-loop integrals or scattering amplitudes feature certain types of singularities leading to ultraviolet (UV), infrared (IR) or threshold singularities. This singular behaviour has already been thoroughly discussed in [29,33]. We briefly recapitulate here the main points that are relevant in the present context.

For generic masses, the loop integrand in Eq. (1) becomes singular at the on-shell hyperboloids defined by $q_{i,0}^{(+)} = \sqrt{\mathbf{q}_i^2 + m_i^2 - i0}$ (forward-hyperboloids, positive energy mode) and $q_{i,0}^{(-)} = -\sqrt{\mathbf{q}_i^2 + m_i^2 - i0}$ (backward-hyperboloids, negative energy mode). This is illustrated in Fig. 2 for a given kinematical configuration with three internal loop propagators. Solid lines in Fig. 2 represent the forward on-shell hyperboloids, and dashed lines the backward on-shell hyperboloids. The LTD method is equivalent to evaluating the sum of the integrals along the forward on-shell hyperboloids
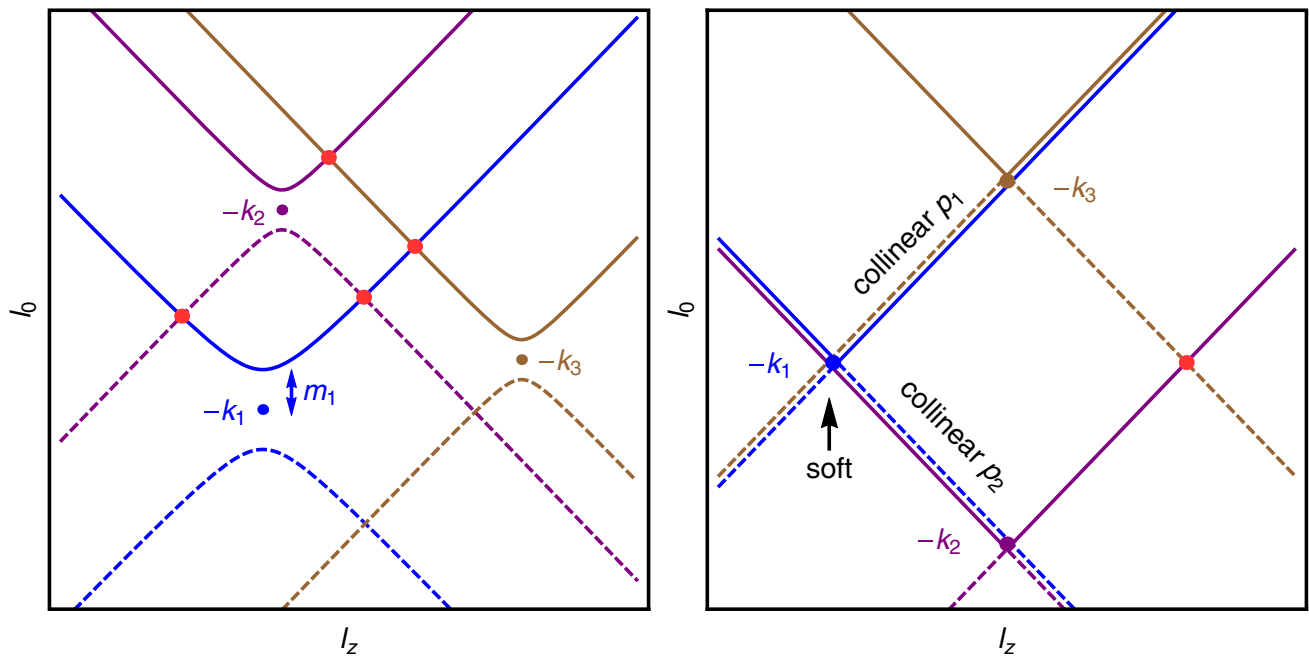
Fig. 2 On-shell hyperboloids for three arbitrary propagators in Cartesian coordinates in the (ℓ_0, ℓ_z) space (*left*). Kinematical configuration with massless propagators leading to infrared singularities (*right*). In the latter case, the on-shell hyperboloids degenerate to light-cones

with singularities appearing at the intersection of each forward on-shell hyperboloid with the forward of backward on-shell hyperboloid of the other propagators. A crucial point of this discussion is the observation that dual propagators can be rewritten as

$$\tilde{\delta}(q_i)\, G_{\rm D}(q_i; q_j) = i\, 2\pi \frac{\delta(q_{i,0} - q_{i,0}^{(+)})}{2q_{i,0}^{(+)}}$$
$$\times \frac{1}{(q_{i,0}^{(+)} + k_{ji,0})^2 - (q_{j,0}^{(+)})^2}, \tag{7}$$

where $q_{i,0}^{(+)}$ can be interpreted as the loop energy measured along the forward on-shell hyperboloid with origin at $-k_i$. From Eq. (7) it is obvious that dual propagators become singular, $G_{\rm D}^{-1}(q_i; q_j) = 0$, if one of the following conditions is fulfilled:

$$q_{i,0}^{(+)} + q_{j,0}^{(+)} + k_{ji,0} = 0, \tag{8}$$
$$q_{i,0}^{(+)} - q_{j,0}^{(+)} + k_{ji,0} = 0. \tag{9}$$

The first condition, Eq. (8), is satisfied if the forward on-shell hyperboloid of $G_{\rm F}(q_i)$ intersects with the backward on-shell hyperboloid of $G_{\rm F}(q_j)$. The second condition, Eq. (9), is true when the two forward on-shell hyperboloids intersect each other.

The solution to Eq. (8) is an ellipsoid in the loop three-momentum space and requires $k_{ji,0} < 0$. Moreover, since it is the result of the intersection of a forward with a backward on-shell hyperboloid the distance between the two propaga-

tors has to be future-like, $k_{ji}^2 \geq 0$. Actually, internal masses restrict this condition to

$$k_{ji}^2 - (m_j + m_i)^2 \geq 0,$$
$$k_{ji,0} < 0, \quad \text{forward with backward hyperboloids.} \tag{10}$$
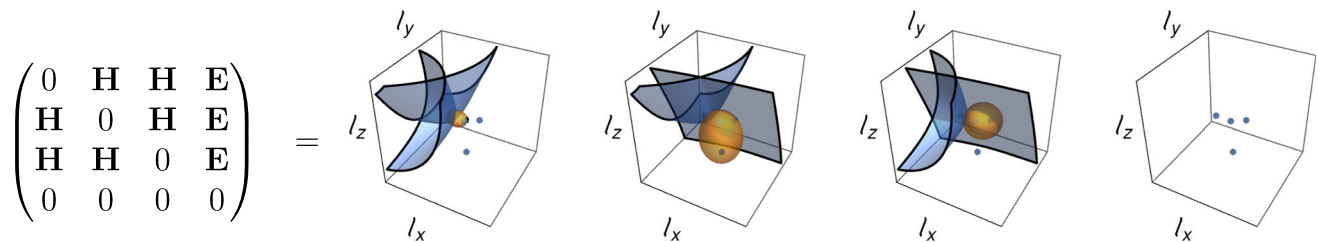
The second equation, Eq. (9), leads to a hyperboloid in the loop three-momentum space, and there are solutions for $k_{ji,0}$ either positive or negative, namely when either of the two momenta is set on-shell. Here, the distance between the momenta of the propagators has to be space-like, although also time-like configurations can fulfil Eq. (9) as far as the time-like distance is small or close to light-like:

$$k_{ji}^2 - (m_j - m_i)^2 \leq 0, \quad \text{two forward hyperboloids.} \tag{11}$$

As was demonstrated in [29], the integrand singularities appearing from the intersection of forward with forward on-shell hyperboloids cancel among dual contributions. One needs to keep in mind that propagators are positive inside the on-shell hyperboloids and negative outside. When integrating along the forward on-shell hyperboloids, every singularity is crossed twice. Firstly when going from the inside to the outside (or from the outside to the inside) and secondly from the outside to the inside (or from the inside to the outside). The crucial point is that the contributions coming from the two integrands have opposite sign and thus cancel out. Note that the imaginary dual prescription $\eta \cdot k_{ji}$ changes sign from the one dual contribution to the other to ensure the cancellation of the singularities. On the contrary, the singu-

$$\begin{pmatrix} 0 & \mathbf{H} & \mathbf{E} \\ \mathbf{H} & 0 & \mathbf{E} \\ 0 & 0 & 0 \end{pmatrix} =$$
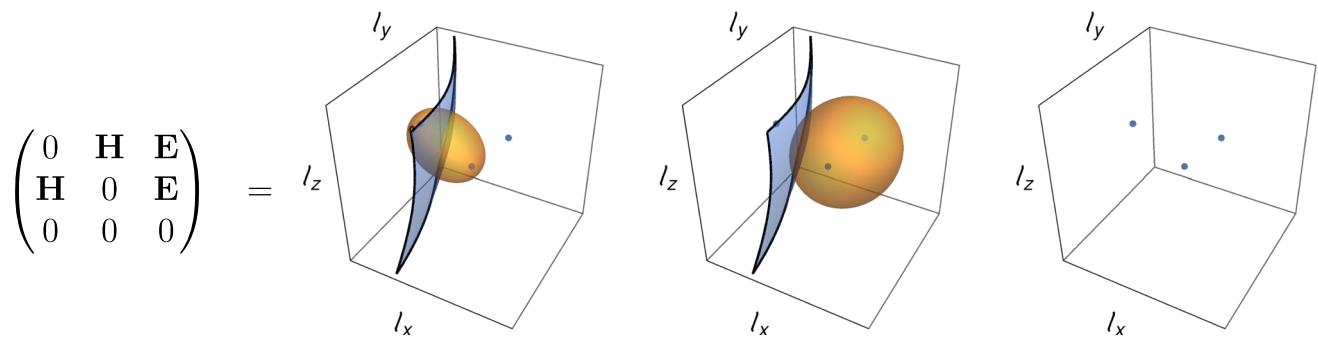
Fig. 3 Singularity matrix of a sample three-point function and the corresponding singularity surfaces of the dual integrands in the loop three-momentum space

$$\begin{pmatrix} 0 & \mathbf{H} & \mathbf{H} & \mathbf{E} \\ \mathbf{H} & 0 & \mathbf{H} & \mathbf{E} \\ \mathbf{H} & \mathbf{H} & 0 & \mathbf{E} \\ 0 & 0 & 0 & 0 \end{pmatrix} =$$

Fig. 4 Singularity matrix of a sample four-point function and the corresponding singularity surfaces of the dual integrands in the loop three-momentum space

larities from the intersection of a forward with a backward on-shell hyperboloid survive because only a single dual contribution leads to that singularity and there is no possibility of cancellation. In the case of integrable singularities, a contour deformation can be employed to integrate numerically these singularities as explained in the next section.

The action of the LTD can be encoded symbolically by the following matrix scheme:

$$G_{\mathrm{F}} \cdot G_{\mathrm{F}} \cdots G_{\mathrm{F}} \xrightarrow{\mathrm{LTD}} \begin{pmatrix} \delta & G_{\mathrm{D}} & G_{\mathrm{D}} & \cdots & G_{\mathrm{D}} \\ G_{\mathrm{D}} & \delta & G_{\mathrm{D}} & \cdots & G_{\mathrm{D}} \\ G_{\mathrm{D}} & G_{\mathrm{D}} & \delta & \cdots & G_{\mathrm{D}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ G_{\mathrm{D}} & G_{\mathrm{D}} & G_{\mathrm{D}} & \cdots & \delta \end{pmatrix}. \quad (12)$$
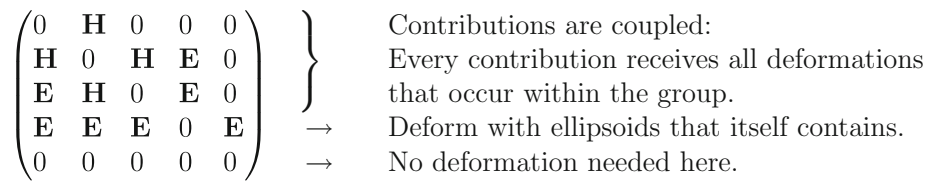
Each line in the matrix to the right of the arrow in Eq. (12) represents a dual contribution with one single propagator on-shell, $\delta = \tilde{\delta}(q_i)$. The column index points to the corresponding dual propagators, $G_{\mathrm{D}} = G_{\mathrm{D}}(q_i; q_j)$. This scheme can now be used to graphically indicate the position of different singularities in a given dual integral. In Figs. 3 and 4 we apply it to a triangle and a box respectively. To be more specific, in Fig. 3 each of the 3D plots in the r.h.s. represents the singularities of any one dual contribution. We plot the ellipsoid (orange surfaces) and hyperboloid (blue surfaces) singularities in the loop three-momentum space. The blue dots are the foci of the on-shell hyperboloids, i.e. $-\mathbf{k}_i$, $i \in \{1, 2, 3\}$. In the l.h.s, we see the singularity

scheme, where the first line of the matrix corresponds to the first plot in the r.h.s., the second line corresponds to the second 3D plot and so on. In the matrix, an H indicates that the corresponding dual propagator from Eq. (12) generates an hyperboloid singularity, E stands for ellipsoid singularities, and zero means no singularity. Similarly, for a four-point function in Fig. 4.

In both cases, the hyperboloid singularities always appear pairwise across the dual contributions. This is not by accident. Due to the symmetry of Eq. (9) under the exchange of i (i counts dual contributions) and j (j counts leg positions) the hyperboloid singularities always appear in pairs and are distributed symmetrically around the main diagonal. Inspecting Eq. (8), which is the defining equation for ellipsoid singularities we see that this equation is not symmetric under the exchange of indices. Thus for every ellipsoid singularity we have a zero as its counterpart.

At this point, we have established that the hyperboloid singularities do cancel among dual contributions and therefore we do not need to treat them in any special manner. Still, though, they do have an impact on the way we need to deform our contour. This is due to the fact that in order to preserve the cancellation of hyperboloid singularities, dual contributions featuring the same hyperboloid singularity must receive the same deformation. To further illustrate this point, let us look at the pentagon example shown in Fig. 5. In Fig. 5, contributions one, two and three are coupled via their common hyperboloid singularities. Thus, they need to receive the very

Fig. 5 Five-point function with dual contributions coupled by hyperboloid singularities

$$
\begin{pmatrix}
0 & \mathbf{H} & 0 & 0 & 0 \\
\mathbf{H} & 0 & \mathbf{H} & \mathbf{E} & 0 \\
\mathbf{E} & \mathbf{H} & 0 & \mathbf{E} & 0 \\
\mathbf{E} & \mathbf{E} & \mathbf{E} & 0 & \mathbf{E} \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\left.\begin{array}{l} \\ \\ \\ \\ \\ \end{array}\right\}
$$

Contributions are coupled:
Every contribution receives all deformations
that occur within the group.
$\rightarrow$ Deform with ellipsoids that itself contains.
$\rightarrow$ No deformation needed here.

same deformation that accounts for all the ellipsoid singularities occurring within those contributions. These are found at position four of the second contribution and positions one and four of the third contribution. The fourth dual contribution is not coupled to any other contribution and a standalone deformation can be applied. The fifth contribution does not require any treatment.

As a general strategy, one organizes the dual contributions into groups. A group is a set of pairwise coupled contributions. To each of the groups a contour deformation is applied independently from the others. Within a group every contribution receives the same deformation, which accounts for all the ellipsoids of the group. Turning back to the example in Fig. 5, we have three groups: the first group involves contributions one to three, the second group is contribution four and the third group is contribution five. Finally, all the dual contributions with the corresponding deformations are summed up together and computed numerically simultaneously. This is necessary because dual propagators are linear in the loop momentum such that individual dual integrals are more singular in the UV than the original integral. The spurious UV behaviour of the dual integrals, however, cancels in the sum and does not interfere with the deformations, which are suppressed at large loop momentum. Actual UV poles can be subtracted with suitable counter-terms [33] that neither interfere with the other singularities. This makes it unnecessary to provide UV information in the singularity matrix, and motivates the fact of considering only UV-finite integrals in this paper.

3 The deformation of the contour

As we saw in Sect. 2, the ellipsoid singularities (forward–backward type) lead to integrable threshold singularities that lie on the real axis. To deal with them, we need to deform the integration path into the imaginary space. Every valid deformation must satisfy a certain set of requirements [3]:

1. The deformation has to respect the $i0$-prescription of the propagator. In general, a contour deformation in the loop three-momentum space has the form

$$
\boldsymbol{\ell} \rightarrow \boldsymbol{\ell}' = \boldsymbol{\ell} + i\boldsymbol{\kappa}, \tag{13}
$$

where $\boldsymbol{\kappa}$ is a function of the loop momentum $\boldsymbol{\ell}$ and the external momenta. In our case, we want to perform the integration over a product of dual propagators. Inserting Eq. (13) into the on-shell energy relation, we obtain

$$
q_{i,0}^{(+)} = \sqrt{-\boldsymbol{\kappa}^2 + 2i\boldsymbol{\kappa}\cdot\mathbf{q}_i + \mathbf{q}_i^2 + m_i^2 - i0}. \tag{14}
$$

The $i0$-prescription tells us in which direction to deform when coming close to a singularity. Hence, any valid deformation must match this prescription. Consequently, we need to have

$$
\boldsymbol{\kappa}\cdot\mathbf{q}_i < 0. \tag{15}
$$

2. The deformation should vanish at infinity: We are looking for a deformation that does not change the actual value of the integral. We do not want $|\boldsymbol{\kappa}|$ to grow for $|\boldsymbol{\ell}| \rightarrow \infty$. An easy way to satisfy this condition is to choose $\boldsymbol{\kappa}$ such that $|\boldsymbol{\kappa}| \rightarrow 0$ as $|\boldsymbol{\ell}| \rightarrow \infty$.[1]

With these conditions in mind, we construct the deformation in the following way: As explained in Sect. 2, we first organize the dual contributions into groups. For every ellipsoid singularity of the group we include a factor:

$$
\lambda_{ij}\left(\frac{\mathbf{q}_i}{\sqrt{\mathbf{q}_i^2}} + \frac{\mathbf{q}_j}{\sqrt{\mathbf{q}_j^2}}\right)\exp\left(-\frac{G_{\mathrm{D}}^{-2}(q_i;q_j)}{A_{ij}}\right), \tag{16}
$$

with $\mathbf{q}_i = \boldsymbol{\ell} + \mathbf{k}_i$ and $\boldsymbol{\ell}$ the loop three-momentum. The deformation factor in Eq. (16) consists of two main components. The first component defines the direction of the deformation and is given by the sum of the two unit vectors $\mathbf{q}_i/\sqrt{\mathbf{q}_i^2}$ and $\mathbf{q}_j/\sqrt{\mathbf{q}_j^2}$. As shown in Fig. 6 the vectors $\mathbf{q}_i$ and $\mathbf{q}_j$ have their origin in $-\mathbf{k}_i$ and $-\mathbf{k}_j$, respectively, and the deformation is designed to point to the outside of the singularity ellipsoid. For an efficient numerical implementation, however, we

---

[1] Strictly speaking, there is a third condition. The deformation must vanish at the position of soft or collinear singularities. This point is of importance for the matching of soft and collinear singularities between real and virtual corrections [33]. If the deformation shifts those singularities together with everything else, the cancellation will be spoiled. However, in the scope of this article, we are only dealing with infrared finite diagrams.
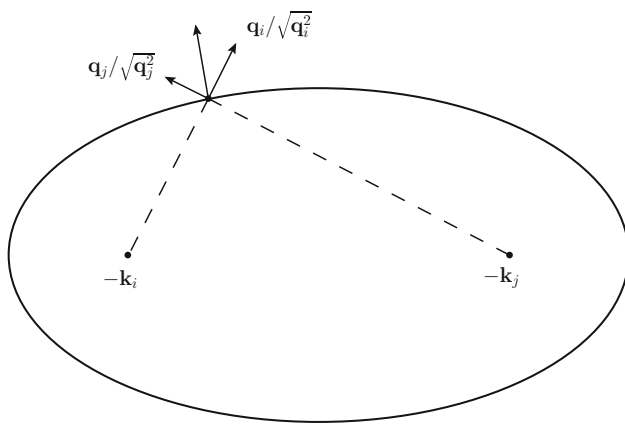
Fig. 6 Two-dimensional slice of the ellipsoid singularity of dual contribution i at position j. The resulting vector gives the direction of the deformation

should also take into account in the selection of the deformation parameters that for massive propagators the vectors $-\mathbf{k}_i$ and $-\mathbf{k}_j$ might be slightly displaced from the true focal points of the ellipsoid. Inside the ellipsoid, the sum of the two unit vectors $\mathbf{q}_i/\sqrt{\mathbf{q}_i^2}$ and $\mathbf{q}_j/\sqrt{\mathbf{q}_j^2}$ helps to flatten the deformation and indeed they cancel each other along the major axis of the ellipsoid. By choosing all the scaling parameters $\lambda_{ij} < 0$ for all possible combinations $\{ij\}$ we satisfy the first condition in Eq. (15).

The second component in Eq. (16) is the exponential factor $\exp(-G_D^{-2}(q_i; q_j)/A_{ij})$, which suppresses the deformation at infinity. At singular points, $G_D^{-2}(q_j; q_i)$ vanishes and thus the deformation reaches its maximum. Far away from the singularity, $-G_D^{-2}(q_j; q_i)$ reaches a large negative value and thus the exponential tends rapidly to zero. Finally, the factor λ_{ij} is a scaling factor, and A_{ij} determines the width of the deformation. The indices ij in λ_{ij} and A_{ij} indicate that those parameters can be selected individually for each deformation contribution for optimization purposes. Although they can all be chosen differently, selecting $\lambda_{ij} = \lambda = -0.5$ and $A_{ij} = A = 10^6$ produces very good results for most of the examples that we are presenting within this paper. As a rule of thumb, the optimal choice is $A_{ij} \approx 10^2 Q^4$ where Q^2 is the physical scale of the process under consideration. Then we sum over the entire group of coupled singularities and arrive at

$$\kappa = \sum_{i,j \in \text{group}} \lambda_{ij} \left(\frac{\mathbf{q}_i}{\sqrt{\mathbf{q}_i^2}} + \frac{\mathbf{q}_j}{\sqrt{\mathbf{q}_j^2}} \right) \exp\left(-\frac{G_D^{-2}(q_j; q_i)}{A_{ij}} \right). \tag{17}$$

There is no formal proof that Eq. (17) always satisfies the condition in Eq. (15) at the intersection of several ellipsoid singularities. The deformation parameters can, however, be adjusted in that case to force the overall deformation to occur

in the right direction. This was necessary in a very few cases of the tests studied for the preparation of this paper and will be subject of improvement and optimization in the next version of the program. The corresponding Jacobian can be calculated analytically or numerically; in our current implementation we have chosen the analytic way.

4 Results for multi-leg scalar one-loop integrals

We have implemented the LTD method in a C++ code and all the results in this paper were obtained on a desktop machine with an Intel i7 (3.4 GHz) processor with 8 cores and 16 GB of RAM. The program uses the Cuba library [39] as a numerical integrator. The user needs only to input the number of external legs, the external momenta, the internal masses and has the freedom to change the parameters λ_{ij} and A_{ij} of the contour deformation, although standard values that work in most of the cases are provided by default. The momenta and masses can be read in from a text file. The user can choose between Cuhre [40,41] and VEGAS [42], and give the desired number of evaluations. At run time, the code performs the following steps:

1. Reads in masses and external momenta.
2. Checks where ellipsoid singularities occur.
3. Checks where hyperboloid singularities occur, groups the dual contributions accordingly and applies the contour deformation.
4. Calls the integrator with the sum of all the deformed dual contributions.

The grouping described in Sect. 2 and the implementation of the deformations are fully automated. We use MATHEMATICA 10.0 [43] to generate random momenta and masses to scan as much of the phase space as possible, to ensure that the program works properly in all regions. For our numerical results, the routine Cuhre was used unless otherwise stated. The momenta and masses of all the sample phase-space points used in the following sections are collected in Appendix A. We mainly used LoopTools 2.10 [38] and also SecDec 3.0 [44] to produce reference results to compare with.

4.1 Scalar triangles

We consider first infrared finite scalar triangle integrals. The sample point P1 in Table 1 has all internal masses equal while P2 has three different internal masses. Momenta and masses were chosen randomly between -100 and $+100$ GeV. Similarly, P3 in Table 1 has all internal masses equal whereas in P4 all three of them have different values.

Table 1 Sample scalar triangles

	Scalar triangle	Real part	Imaginary part
P1	LoopTools	-5.85694×10^{-5}	
	LTD	$-5.85685(24) \times 10^{-5}$	
P2	LoopTools	-3.39656×10^{-7}	
	LTD	$-3.39688(53) \times 10^{-7}$	
P3	LoopTools	5.37305×10^{-4}	-6.68103×10^{-4}
	LTD	$5.37307(9) \times 10^{-4}$	$-6.68103(9) \times 10^{-4}$
P4	LoopTools	-5.61370×10^{-7}	-1.01665×10^{-6}
	LTD	$-5.61363(83) \times 10^{-7}$	$-1.01666(8) \times 10^{-6}$

Table 2 Sample scalar boxes

	Scalar box	Real part	Imaginary part
P5	LoopTools	2.15339×10^{-13}	
	LTD	$2.15319(52) \times 10^{-13}$	
P6	LoopTools	1.39199×10^{-11}	
	LTD	$1.39199(6) \times 10^{-11}$	
P7	LoopTools	-2.38766×10^{-10}	-3.03080×10^{-10}
	LTD	$-2.38775(76) \times 10^{-10}$	$-3.03063(76) \times 10^{-10}$
P8	LoopTools	-4.27118×10^{-11}	4.49304×10^{-11}
	LTD	$-4.27120(95) \times 10^{-11}$	$4.49307(95) \times 10^{-11}$
P9	LoopTools	-7.37897×10^{-11}	-1.19657×10^{-10}
	LTD	$-7.37916(782) \times 10^{-11}$	$-1.19649(78) \times 10^{-10}$
P10	LoopTools	-1.85544×10^{-10}	2.13553×10^{-10}
	LTD	$-1.85548(8) \times 10^{-10}$	$2.13554(8) \times 10^{-10}$

Points with momentum configurations that do not need deformation (i.e. whose loop integral is purely real) are computed in well below one second with a precision of at least 4 digits. For points with momentum configurations that require deformation, the calculation time increases to typically 3–15 s.

An important check of our implementation is the mass scan around threshold. In Fig. 7 all internal masses are equal, i.e. $m_i = m$, $i \in \{1, 2, 3\}$, and the centre-of-mass energy s was kept constant while varying the mass m. The calculation time remains constant for all mass values.

4.2 Scalar boxes

Next we consider infrared finite box scalar integrals. To get good precision (4 digits), for boxes that need deformation we use 4–5×10^6 Cuhre calls, whereas for phase-space points with no deformation only 5×10^4 calls, the same as in the triangle case. This is reflected in the running times. Points with deformation require about 15 s whereas points with no deformation well below one second. While it is practically guaranteed to get the no-deformation points with good precision,

for points with deformation the quality of results depends on the proper choice of the deformation parameters. Therefore, we mainly focus our attention to such points in the following. The sample points P5 and P7 of Table 2 correspond to a momentum configuration in which all four internal masses are equal. In P6 and P8 all masses are different. In P9 two adjacent internal lines have equal masses as well as the two opposing ones. P10 represents a situation in which opposite lines have equal masses.

We perform again a mass scan (see Fig. 8) with all internal masses equal, i.e. $m_i = m$, $i \in 1, 2, 3, 4$. The centre-of-mass energy s was kept constant while the mass m was varied. The program deals well with all kinds of boxes, even when many different kinematical scales are involved. In Fig. 8, two thresholds are crossed at $2m/\sqrt{s} = 0.65$ and 1. From right to left, the number of ellipsoid singularities grows by one after each threshold is crossed, starting from one to end up to three.
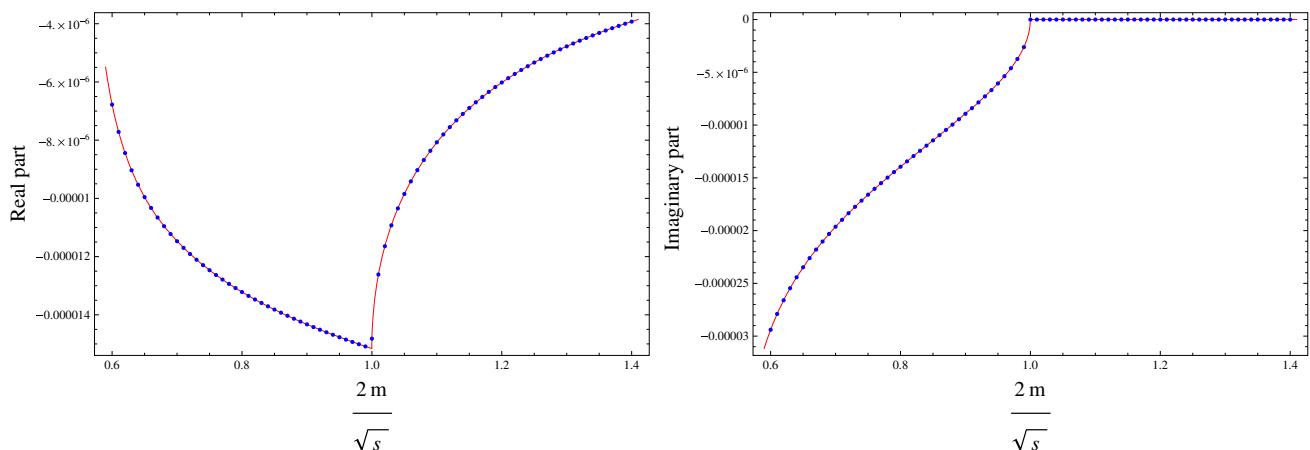
Fig. 7 Mass scan of the region around threshold. The *red curve* is done with LoopTools and the *blue points* are obtained with the LTD method
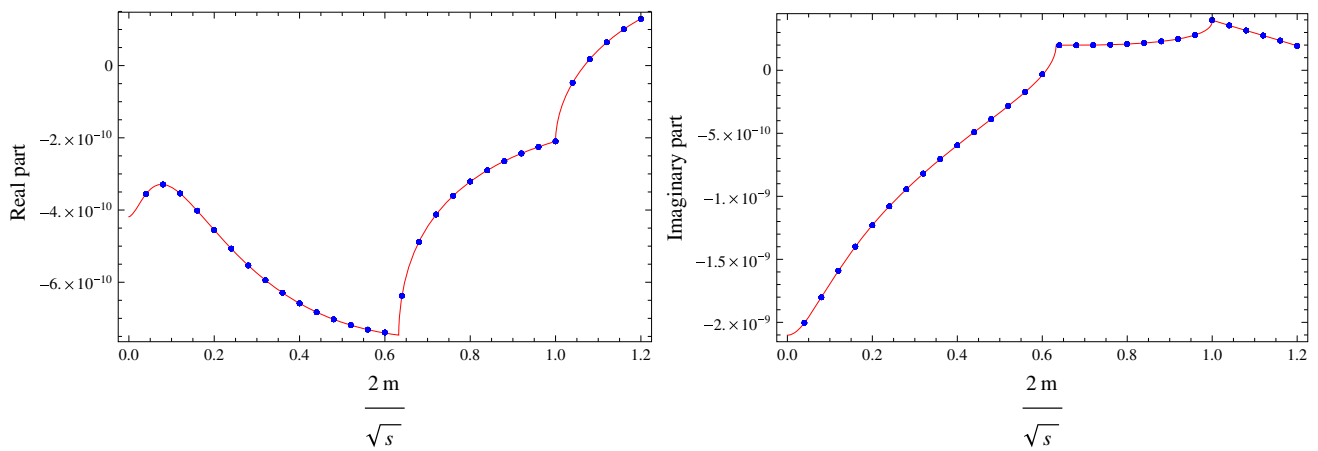
Fig. 8 Mass scan of a box integral. The *red curve* is done with LoopTools and the *blue points* are obtained with the LTD method

Table 3 Sample scalar pentagons

	Scalar pentagon	Real part	Imaginary part
P11	LoopTools	-1.24025×10^{-13}	
	LTD	$-1.24027(16) \times 10^{-13}$	
P12	LoopTools	-1.48356×10^{-14}	
	LTD	$-1.48345(116) \times 10^{-14}$	
P13	LoopTools	1.02350×10^{-11}	1.40382×10^{-11}
	LTD	$1.02353(1) \times 10^{-11}$	$1.40385(1) \times 10^{-11}$
P14	LoopTools	-1.52129×10^{-15}	-1.17401×10^{-14}
	LTD	$-1.52657(602) \times 10^{-15}$	$-1.17483(60) \times 10^{-14}$
P15	LoopTools	-4.29464×10^{-15}	-6.55440×10^{-14}
	LTD	$-4.29520(845) \times 10^{-15}$	$-6.55433(85) \times 10^{-14}$

4.3 Scalar pentagons

Let us now turn to pentagon diagrams. No-deformation points are computed with 10^5 evaluations which takes about 0.5 s. Points with deformation demand 5×10^6 evaluations to maintain the level of precision of the triangles and boxes. This results in an average calculation time of about 30 s.

In Table 3 we display a collection of pentagon example results for different kinematical configurations. In P11 and P13 all internal masses are equal; in P14 they are all distinct from each other and in P15 we have $m_1 = m_2 = m_3 \neq m_4 = m_5$. Our implementation of the LTD method shows its robustness by producing accurate results regardless of the kinematical situation. This statement is further supported by an energy scan which we performed and which is shown in Fig. 9. The centre-of-mass energy s is varied. This is realized by varying p_3 while keeping p_3^2 constant. Of course, due to momentum conservation, this involves $p_4^2 = (p_1 + p_2 + p_3)^2$ not being constant. In this scan, we cross three thresholds at $s \approx -8.5 \times 10^3, -13.5 \times t 10^3$ and -21×10^3 GeV^2, which divide the scan into four zones. From right to left, we start with zero ellipsoid singularities in the first zone, then we have

one in the second zone, two in the third zone and finally one in the leftmost zone.
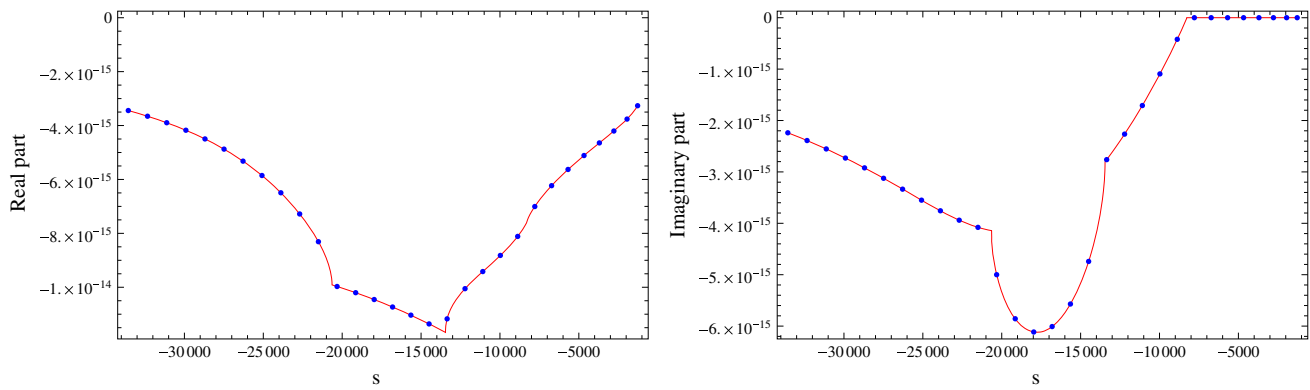
5 Tensor loop integrals

The LTD relation for scalar loop integrals can easily be extended to deal tensor integrals. As long as the quantum field theory is local and unitary, these tensor factors do not lead to additional singularities [24] and the LTD method can then be applied in a straightforward manner. If the one-loop integral features a non-trivial numerator $\mathcal{N}(\ell, \{p_i\})$,

$$L^{(1)}(p_1, \ldots, p_N; \mathcal{N}(\ell, \{p_i\}))$$
$$= \int_\ell \mathcal{N}(\ell, \{p_i\}) \prod_{i \in \alpha_1} G_F(q_i), \tag{18}$$

then the LTD theorem takes the form

$$L^{(1)}(p_1, \ldots, p_N; \mathcal{N}(\ell, \{p_i\}))$$
$$= -\sum_{i \in \alpha_1} \int_\ell \tilde{\delta}(q_i) \mathcal{N}(\ell, \{p_i\}) \prod_{\substack{j \in \alpha_1 \\ j \neq i}} G_D(q_i; q_j). \tag{19}$$

Fig. 9 Energy scan of a scalar pentagon. The *red curve* is done with LoopTools and the *blue points* are obtained with the LTD method

While the numerator is formally left unchanged, there is actually a potential implication. The presence of the dual delta function demands $q_{i,0}^{(+)} = \sqrt{\mathbf{q}_i^2 + m_i^2 - i0}$, which is equivalent to

$$\ell_0 = -k_{i,0} + \sqrt{\mathbf{q}_i^2 + m_i^2 - i0}. \tag{20}$$

In other words, whenever we perform a single cut of a Feynman graph, the numerator has to be evaluated at the position of the cut which is fixed by the dual delta function. As a direct consequence, the numerator takes a different form in each dual contribution.

Another important aspect to take into consideration is the cancellation of singularities among dual contributions. Here, we would like to make explicit that the numerators do not spoil the cancellation of the hyperboloid singularities. A typical numerator is a polynomial of scalar products of the loop momentum with external momenta: $\ell \cdot p_k$. Let us see what happens to a single factor when it hits the singularity. Note, first, that the hyperboloid singularity is given by Eq. (9) which we rewrite in the more suitable form

$$q_{i,0}^{(+)} - k_{i,0} = q_{j,0}^{(+)} - k_{j,0}. \tag{21}$$

Using Eq. (20), the loop momentum ℓ contracted with some external momentum p_k is

$$\begin{aligned}
\ell \cdot p_k \mid_{\text{ith cut}} &= (q_{i,0}^{(+)} - k_i) p_{k,0} - \boldsymbol{\ell} \cdot \mathbf{p}_k \\
&= (q_{j,0}^{(+)} - k_j) p_{k,0} - \boldsymbol{\ell} \cdot \mathbf{p}_k = \ell \cdot p_k \mid_{\text{jth cut}},
\end{aligned} \tag{22}$$

where we have used Eq. (21). This means that the numerators of two dual contributions i and j take the same value at their common pole, thus leaving the cancellation of hyperboloid singularities intact. This is an important property to take advantage of, because it allows us to straightforwardly apply the LTD method to such diagrams without any additional effort.

5.1 Tensor pentagons

Next, we investigate tensor pentagon integrals at the one-loop level with numerators up to rank three. The number of evaluations is chosen to be the same as in the scalar case, i.e. 10^5 times for no-deformation points and 5×10^6 times for phase-space points that require deformation. This results in calculation times of about 1 and 30 s, respectively.

In Table 4 we show a selection of sample points. The reference points P24 and P26 feature the rank-two numerator $(\ell \cdot p_3) \times (\ell \cdot p_4)$ while P25 and P27 have the numerator $(\ell \cdot p_3) \times (\ell \cdot p_4) \times (\ell \cdot p_5)$. All the points have all internal masses equal. P27 actually contains six ellipsoid singularities whereas the other sample points have two to three. We include this point to demonstrate that the program does well even under such challenging circumstances.

For tensor pentagons and hexagons, we have used the program SecDec [44] to cross-check our results. We have run SecDec taking no care to optimize its runtime. This means that in the following, whenever we present the running times of SecDec we do it for completeness reasons and not because we imply that our code compares better or worse with SecDec. A proper comparison of our implementation with available codes is beyond the scope of this paper.

We have performed several different scans; a sample is presented in Fig. 10. In this energy scan, we varied p_1 and thus the centre-of-mass energy $s = (p_1 + p_2)^2$, similar to what we have done with scalar pentagons. The corresponding numerator function is $(\ell \cdot p_1) \times (\ell \cdot p_2) \times (\ell \cdot p_3)$, which means that both numerator and denominator change in the scan. In Fig. 10, one can see that the LTD method is able to successfully pass this test.

5.2 Tensor hexagons

In this subsection, we compute hexagon tensor integrals. The number of evaluations for no-deformation points is 10^6 and for deformation points 8×10^6. The typical corresponding

Table 4 Tensor pentagons involving numerators of rank two and rank three

Tensor pentagon	Rank		Real part	Imaginary part	Time (s)
P16	2	LoopTools	-1.86472×10^{-8}		
		SecDec	$-1.86471(2) \times 10^{-8}$		45
		LTD	$-1.86462(26) \times 10^{-8}$		1
P17	3	LoopTools	1.74828×10^{-3}		
		SecDec	$1.74828(17) \times 10^{-3}$		550
		LTD	$1.74808(283) \times 10^{-3}$		1
P18	2	LoopTools	-1.68298×10^{-6}	1.98303×10^{-6}	
		SecDec	$-1.68307(56) \times 10^{-6}$	$1.98279(90) \times 10^{-6}$	66
		LTD	$-1.68298(74) \times 10^{-6}$	$1.98299(74) \times 10^{-6}$	36
P19	3	LoopTools	-8.34718×10^{-2}	1.10217×10^{-2}	
		SecDec	$-8.33284(829) \times 10^{-2}$	$1.10232(107) \times 10^{-2}$	1501
		LTD	$-8.34829(757) \times 10^{-2}$	$1.10119(757) \times 10^{-2}$	38
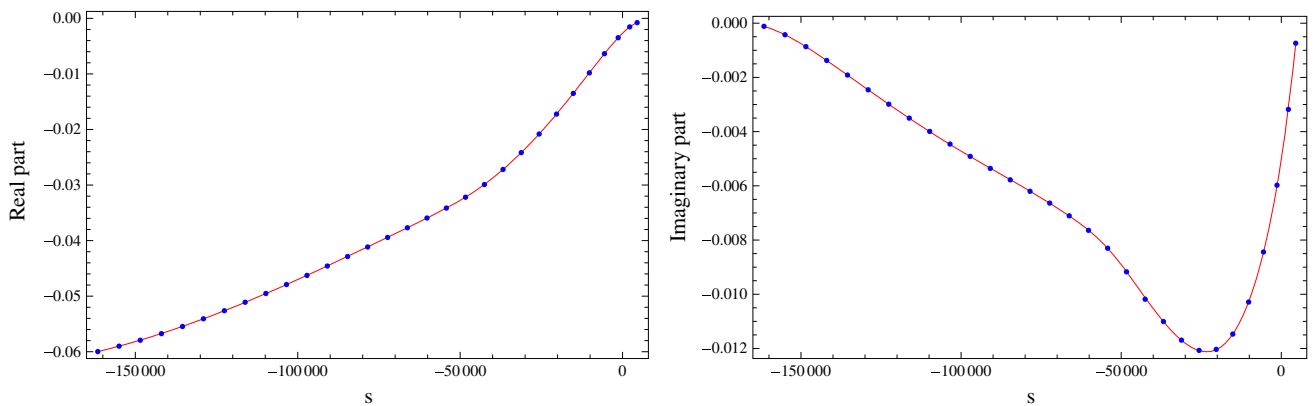
Fig. 10 Energy scan of a rank-three tensor pentagon around the threshold region. The *red curve* is done with LoopTools and the *blue points* are obtained with the LTD method

computation times are about 10 and 75 s, respectively. Since LoopTools can provide results only up to pentagons, we used exclusively here the program SecDec to cross-check.

We present a selection of sample points in Table 5. P28 and P30 feature the rank-one numerator $\ell \cdot p_1$, the former has all internal masses different, in the latter they are all equal. P29 has six distinct internal masses and the numerator function $(\ell \cdot p_2) \times (\ell \cdot p_4) \times (\ell \cdot p_6)$, P31 possesses the numerator $(\ell \cdot p_2) \times (\ell \cdot p_5)$ and six different masses, as well. Finally, P32 has all momenta distinct form each other and exhibits the numerator $(\ell \cdot p_4) \times (\ell \cdot p_5) \times (\ell \cdot p_6)$.

5.3 Some non-trivial examples beyond six external legs

Finally, we will present examples of one-loop heptagons and octagons. In particular, we will present a scalar and a rank-two tensor heptagon with all propagator masses equal and a scalar heptagon with all propagator masses different and exactly the same configuration for an octagon, in

total six points (P25–P30, to be found in Appendix A.1). In Tables 6 and 7, rank = 0 stands for scalar. The numerators of both the tensor heptagon and octagon diagrams are equal to $(\ell \cdot p_2) \times (\ell \cdot p_4)$. The external momenta for this subsection were generated by using the flat phase-space generator RAMBO [45]. The results shown here were obtained by using LDT only and they all require a contour deformation. The number of evaluations and the running times are similar to the ones in the previous subsection.

6 Conclusions and outlook

The loop–tree duality has many appealing theoretical properties for the calculation of processes with many external legs. In this paper, we have investigated the practicability of a first numerical implementation of the LTD method.

In our analysis of the singular behaviour of the loop integrand, we found two distinct types of singularities: Ellip-

Table 5 Tensor hexagons involving numerators of rank-one to three

Tensor hexagon	Rank		Real part	Imaginary part	Time (s)
P20	1	SecDec	$-1.21585(12) \times 10^{-15}$		36
		LTD	$-1.21552(354) \times 10^{-15}$		6
P21	3	SecDec	$4.46117(37) \times 10^{-9}$		5498
		LTD	$4.461369(3) \times 10^{-9}$		11
P22	1	SecDec	$1.01359(23) \times 10^{-15}$	$2.68657(26) \times 10^{-15}$	33
		LTD	$1.01345(130) \times 10^{-15}$	$2.68633(130) \times 10^{-15}$	72
P23	2	SecDec	$2.45315(24) \times 10^{-12}$	$-2.06087(20) \times 10^{-12}$	337
		LTD	$2.45273(727) \times 10^{-12}$	$-2.06202(727) \times 10^{-12}$	75
P24	3	SecDec	$-2.07531(19) \times 10^{-6}$	$6.97158(56) \times 10^{-7}$	14280
		LTD	$-2.07526(8) \times 10^{-6}$	$6.97192(8) \times 10^{-7}$	85

Table 6 Scalar and rank-two tensor heptagons

Heptagon	Rank	Real part	Imaginary part
P25	0	$-3.45252(78) \times 10^{-10}$	$-1.50195(78) \times 10^{-9}$
P26	0	$-3.65854(79) \times 10^{-9}$	$-5.84357(79) \times 10^{-9}$
P27	2	$-2.59635(78) \times 10^{-8}$	$-1.14108(78) \times 10^{-7}$

Table 7 Scalar and rank-two tensor octagons

Octagon	Rank	Real part	Imaginary part
P28	0	$-2.07946(63) \times 10^{-11}$	$9.43953(63) \times 10^{-11}$
P29	0	$6.826303(63) \times 10^{-10}$	$9.17379(63) \times 10^{-10}$
P30	2	$-3.77449(65) \times 10^{-10}$	$2.82760(65) \times 10^{-9}$

soid singularities which require the application of contour deformation and hyperboloid singularities that occur pairwise and cancel among different dual contributions. In order to preserve their cancellation, dual contributions featuring the same hyperboloid singularity pair must receive the same contour deformation. This leads to the following algorithm: Sets of pairwise coupled dual contributions are organized into groups. Each group is deformed independently from the others and each dual contribution of such a group receives the exact same contour deformation, which accounts for all the ellipsoid singularities of the entire group.

We applied a contour deformation that efficiently deals with the ellipsoid singularities by meeting all the important criteria [3]. This setup has been successful in the calculation of finite multi-leg scalar and tensor integrals. We found the results to be in very good agreement with the reference values produced by LoopTools and SecDec. An important further check of our implementation presented here was various scans which show that the code handles equally well broad slices of the phase space. The code excels in cases that involve many external legs as it shows a modest increase in running times in comparison to cases with fewer legs. From this first study, we can be optimistic that our implementation of the LTD method offers a competitive alternative for computing multi-scale, multi-leg scalar and tensor one-loop integrals.

In this paper, we have considered IR- and UV-finite integrals, specifically loop integrals that only feature integrable threshold singularities. The final aim is to implement virtual and real contributions simultaneously in such a way that IR and UV singularities are cancelled locally at integrand level. Therefore, there is no need to evaluate the poles of loop scattering amplitudes and only threshold singularities need a numerical treatment in order to obtain predictions for physical observables. The UV singularities can be subtracted with suitable unintegrated counter-terms, and the IR singularities can be cancelled by establishing momentum mappings between the virtual and real kinematics, as explained in Refs. [33,35–37]. With the results presented in this paper, and the four-dimensional unsubtraction (FDU) method presented in Refs. [33,35–37], it is possible to afford multi-leg calculations at NLO, with either massless or massive virtual and external particles. The extension to NNLO has also been anticipated in [35], and the LTD approach is also more direct than other methods in asymptotic expansions [37]. The implementation of the method in a multipurpose Monte Carlo event generator and further applications are currently under investigation.

Appendix A: External momenta and internal masses of the sample points

Here we give the external momenta and internal masses of the different phase-space points and scans shown in Sects. 4 and 5. Due to momentum conservation $p_N = -\sum_{i=1}^{N-1} p_i$, it is sufficient to give only the momenta p_1 to p_{N-1}. Momenta and masses are implicitly given in GeV.

To produce the energy scans (Figs. 9, 10), we varied the external momentum p_1 by multiplying with the square root of some scaling parameter λ (not to be confused with the scaling parameter λ_{ij} of the contour deformation). This is indicated where we give the respective momenta.

A.1: Individual sample points

Figure 3 $p_1 = (44.38942, 17.84418, 12.70440, -23.67441)$
         $p_2 = (11.62982, -35.11756, -9.52573, 1.27635)$
         $m_1 = m_2 = m_3 = 7.89824$

Figure 4 $p_1 = (95.95213, 65.25140, -40.62468, 30.93648)$
         $p_2 = (68.47023, -60.09584, 18.23998, 84.29507)$
         $p_3 = (12.99839, 12.08603, -99.08246, -34.58997)$
         $m_1 = m_2 = m_3 = m_4 = 11.50163$

$P1$ $p_1 = (5.23923, -4.18858, 0.74966, -3.05669)$
     $p_2 = (6.99881, -2.93659, 5.03338, 3.87619)$
     $m_1 = m_2 = m_3 = 7.73358$

$P2$ $p_1 = (13.42254, 58.79478, -73.11858, -91.95015)$
     $p_2 = (81.65928, -68.52173, 8.75578, -95.05353)$
     $m_1 = 49.97454, m_2 = 86.92490, m_3 = 80.22567$

$P3$ $p_1 = (10.51284, 6.89159, -7.40660, -2.85795)$
     $p_2 = (6.45709, 2.46635, 5.84093, 1.22257)$
     $m_1 = m_2 = m_3 = 0.52559$

$P4$ $p_1 = (95.77004, 31.32025, -34.08106, -9.38565)$
     $p_2 = (94.54738, -53.84229, 67.11107, 45.56763)$
     $m_1 = 83.02643, m_2 = 76.12873, m_3 = 55.00359$

$P5$ $p_1 = (31.54872, -322.40325, 300.53015, -385.58013)$
     $p_2 = (103.90430, 202.00974, -451.27794, -435.12848)$
     $p_3 = (294.76653, 252.88958, 447.09194, 311.71630)$
     $m_1 = m_2 = m_3 = m_4 = 4.68481$

$P6$ $p_1 = (50.85428, -55.74613, 11.69987, 94.92591)$
     $p_2 = (0.69914, 67.19262, -5.78627, 91.52776)$
     $p_3 = (52.35768, 76.32258, 43.82222, 13.05874)$
     $m_1 = 54.29650, m_2 = 53.54058, m_3 = 55.96814,$
     $m_4 = 51.74438$

$P7$ $p_1 = (62.80274, -49.71968, -5.53340, -79.44048)$
     $p_2 = (48.59375, -1.65847, 34.91140, 71.89564)$
     $p_3 = (76.75934, -19.14334, -17.10279, 30.22959)$
     $m_1 = m_2 = m_3 = m_4 = 9.82998$

$P8$ $p_1 = (98.04093, 77.37405, 30.53434, -81.88155)$
     $p_2 = (73.67657, -53.78754, 13.69987, 14.20439)$
     $p_3 = (68.14197, -36.48119, 59.89499, -81.79030)$
     $m_1 = 81.44869, m_2 = 94.39003, m_3 = 57.53145,$
     $m_4 = 0.40190$

$P9$ $p_1 = (76.50219, -72.36197, 10.95225, -99.79612)$
     $p_2 = (99.02723, 27.27133, -25.11907, 86.10825)$
     $p_3 = (64.19420, 13.10011, 18.37737, -29.16095)$
     $m_1 = m_2 = 37.77809, m_3 = m_4 = 36.84323$

$P10$ $p_1 = (13.62303, -64.20757, -17.59085, -8.81785)$
      $p_2 = (96.67650, 89.65623, -18.47276, 40.73203)$
      $p_3 = (66.21913, -39.49917, 3.640139, -82.31669)$
      $m_1 = m_3 = 64.67282, m_2 = m_4 = 51.13181$

$P11$ $p_1 = (33.74515, 45.72730, 31.15254, -7.47943)$
      $p_2 = (31.36435, -41.50734, 46.47897, 2.04203)$
      $p_3 = (4.59005, 17.07010, 32.65403, 41.93628)$
      $p_4 = (29.51054, -28.25963, 46.17333, -35.08918)$
      $m_1 = m_2 = m_3 = m_4 = m_5 = 5.01213$

$P12$ $p_1 = (33.76482, 45.44063, -10.68084, 16.41925)$
      $p_2 = (72.93498, 67.49170, -11.81485, -36.28455)$
      $p_3 = (8.01673, -49.40112, -66.09200, -0.11414)$
      $p_4 = (-86.54188, -97.01228, 68.12494, 32.94875)$
      $m_1 = 98.42704, m_2 = 28.89059, m_3 = 40.51436,$
      $m_4 = 75.45643, m_5 = 11.08327$

$P13$ $p_1 = (1.58374, 6.86200, -15.06805, -10.63574)$
      $p_2 = (7.54800, -3.36539, 34.57385, 27.52676)$
      $p_3 = (43.36396, -49.27646, -25.35062, -17.68709)$
      $p_4 = (22.58103, 38.31530, -14.67581, -3.08209)$
      $m_1 = m_2 = m_3 = m_4 = m_5 = 2.76340$

$P14$ $p_1 = (-89.85270, 69.44839, -96.30496, 14.47549)$
$p_2 = (-81.61779, 6.89065, 1.76775, 18.39834)$
$p_3 = (-89.80789, 24.32486, 48.73341, 0.74094)$
$p_4 = (-43.20198, -85.34635, 92.38148, 93.84802)$
$m_1 = 22.21430, m_2 = 15.84324, m_3 = 34.80431,$
$m_4 = 27.53390, m_5 = 29.19823$

$P15$ $p_1 = (94.79774, -70.04005, -84.77221, 36.09812)$
$p_2 = (-42.15872, -36.33754, -14.72331, -41.24018)$
$p_3 = (73.77293, 88.37064, 33.47296, -24.17542)$
$p_4 = (81.85638, 77.17370, -62.39774, -6.89737)$
$m_1 = m_2 = m_3 = 1.30619, m_4 = m_5 = 1.26692$

$P16$ $p_1 = (69.70234, 62.68042, 25.44429, -97.78603)$
$p_2 = (-65.98494, -85.19920, 98.05702, -70.89141)$
$p_3 = (-26.75642, -30.42288, -26.84633, 14.81944)$
$p_4 = (-69.44800, 56.74842, -32.23649, 96.45829)$
$m_1 = m_2 = m_3 = m_4 = m_5 = 87.00572$

$P17$ $p_1 = (-45.80756, 95.63842, -55.04954, 44.01174)$
$p_2 = (36.09562, 52.66752, -11.22354, -87.48918)$
$p_3 = (-4.90798, 41.11273, 14.29379, 2.15944)$
$p_4 = (49.48233, 40.26756, -23.16581, -96.89362)$
$m_1 = m_2 = m_3 = m_4 = m_5 = 56.97318$

$P18$ $p_1 = (-18.90057, -97.14671, 44.69176, -16.67528)$
$p_2 = (-70.86315, -81.27489, -3.71628, 18.79403)$
$p_3 = (-89.53092, 50.02356, 33.39784, -51.66031)$
$p_4 = (-96.59097, -34.80215, -83.24353, 44.73888)$
$m_1 = m_2 = m_3 = m_4 = m_5 = 43.87459$

$P19$ $p_1 = (-88.70322, 37.98826, 62.19352, -35.86433)$
$p_2 = (-58.60617, -58.60074, -83.75298, 61.78210)$
$p_3 = (-83.73607, 46.98912, 67.44602, 78.40612)$
$p_4 = (-96.41508, 71.69925, -14.47818, -61.82390)$
$m_1 = m_2 = m_3 = m_4 = m_5 = 16.73899$

$P20$ $p_1 = (-3.43584, 4.73492, 17.31242, 61.53467)$
$p_2 = (12.12233, 32.23256, 87.57836, -58.25073)$
$p_3 = (-38.67209, -54.27020, 21.15570, 79.15640)$
$p_4 = (-90.90573, -79.70266, -88.26463, -66.00973)$
$p_5 = (-34.40043, -88.73043, 84.41781, -4.21221)$
$m_1 = 54.36459, m_2 = 30.96600, m_3 = 51.03652,$
$m_4 = 16.03115, m_5 = 2.25657, m_6 = 59.45020$

$P21$ $p_1 = (-9.85384, 15.70678, 80.94234, -84.96387)$
$p_2 = (90.11707, -74.59469, -70.73997, 54.32748)$
$p_3 = (-55.84212, -34.47531, -87.20597, -27.73882)$

$p_4 = (16.72808, 64.83574, -31.16733, 63.94189)$
$p_5 = (-42.62943, 49.91058, -46.12974, 59.76096)$
$m_1 = 42.61768, m_2 = 22.13590, m_3 = 34.87263,$
$m_4 = 54.00634, m_5 = 79.54844, m_6 = 87.50131$

$P22$ $p_1 = (35.27512, 36.08798, -89.66662, 18.22907)$
$p_2 = (-32.58939, 14.45447, 86.93898, -47.20827)$
$p_3 = (-76.40210, -62.22587, -63.59955, 41.03465)$
$p_4 = (-2.30248, 0.45058, -76.74256, -64.19292)$
$p_5 = (-88.80252, 18.06504, -6.53891, 49.34535)$
$m_1 = m_2 = m_3 = m_4 = m_5 = m_6 = 82.87370$

$P23$ $p_1 = (-99.20747, -68.16217, 95.24772, 68.87644)$
$p_2 = (-95.09224, 78.51258, -82.38270, 20.36899)$
$p_3 = (-56.04092, 22.93681, -72.82681, 96.81954)$
$p_4 = (78.53840, -86.40143, -82.49674, -57.42855)$
$p_5 = (13.70265, 77.87278, 99.79126, 8.31677)$
$m_1 = 63.23680, m_2 = 86.48449, m_3 = 44.51361,$
$m_4 = 79.73599, m_5 = 74.43246, m_6 = 70.11421$

$P24$ $p_1 = (-70.26380, 96.72681, 21.66556, -37.40054)$
$p_2 = (-13.45985, 2.12040, 3.20198, 91.44246)$
$p_3 = (-62.59164, -29.93690, -22.16595, -58.38466)$
$p_4 = (-67.60797, -83.23480, 18.49429, 8.94427)$
$p_5 = (-34.70936, -62.59326, -60.71318, 2.77450)$
$m_1 = 94.53242, m_2 = 64.45092, m_3 = 74.74299,$
$m_4 = 10.63129, m_5 = 31.77881, m_6 = 23.93819$

$P25$ $p_1 = (-2.50000, 0, 0, -2.50000)$
$p_2 = (-2.50000, 0, 0, 2.50000)$
$p_3 = (-0.43177, 0.07607, -0.18817, 0.38109)$
$p_4 = (-1.40059, 0.41563, 1.31602, -0.23875)$
$p_5 = (-0.63761, 0.45225, 0.41589, -0.17047)$
$p_6 = (-2.07065, -1.16073, -1.66377, 0.41493)$
$m_1 = 4.50676, m_2 = 4.50676, m_3 = 4.50676,$
$m_4 = 4.50676, m_5 = 4.50676, m_6 = 4.50676,$
$m_7 = 4.50676$

$P26$ $p_1 = (-2.50000, 0, 0, -2.50000)$
$p_2 = (-2.50000, 0, 0, 2.50000)$
$p_3 = (-0.43177, 0.07607, -0.18817, 0.38109)$
$p_4 = (-1.40059, 0.41563, 1.31602, -0.23875)$
$p_5 = (-0.63761, 0.45225, 0.41589, -0.17047)$
$p_6 = (-2.07065, -1.16073, -1.66377, 0.41493)$
$m_1 = 4.50676, m_2 = 2.81491, m_3 = 1.42763,$
$m_4 = 7.62154, m_5 = 5.26917, m_6 = 3.52104,$
$m_7 = 5.88815$

$P27$ $p_1 = (-2.50000, 0, 0, -2.50000)$
$p_2 = (-2.50000, 0, 0, 2.50000)$
$p_3 = (-0.43177, 0.07607, -0.18817, 0.38109)$
$p_4 = (-1.40059, 0.41563, 1.31602, -0.23875)$
$p_5 = (-0.63761, 0.45225, 0.41589, -0.17047)$
$p_6 = (-2.07065, -1.16073, -1.66377, 0.41493)$
$m_1 = 4.50676, m_2 = 4.50676, m_3 = 4.50676,$
$m_4 = 4.50676, m_5 = 4.50676, m_6 = 4.50676,$
$m_7 = 4.50676$

$P28$ $p_1 = (-2.50000, 0, 0, -2.50000)$
$p_2 = (-2.50000, 0, 0, 2.50000)$
$p_3 = (-0.42766, 0.04111, -0.18082, 0.38536)$
$p_4 = (-0.90714, 0.28930, 0.85932, 2.80593)$
$p_5 = (-0.41425, 0.32955, 0.24948, -0.02757)$
$p_6 = (-1.90735, -0.95093, -1.46021, 0.77557)$
$p_7 = (-0.27116, 0.15567, 0.03964, -0.21846)$
$m_1 = 4.50676, m_2 = 4.50676, m_3 = 4.50676,$
$m_4 = 4.50676,$
$m_5 = 4.50676, m_6 = 4.50676, m_7 = 4.50676,$
$m_8 = 4.50676$

$P29$ $p_1 = (-2.50000, 0, 0, -2.50000)$
$p_2 = (-2.50000, 0, 0, 2.50000)$
$p_3 = (-0.42766, 0.04111, -0.18082, 0.38536)$
$p_4 = (-0.90714, 0.28930, 0.85932, 2.80593)$
$p_5 = (-0.41425, 0.32955, 0.24948, -0.02757)$
$p_6 = (-1.90735, -0.95093, -1.46021, 0.77557)$
$p_7 = (-0.27116, 0.15567, 0.03964, -0.21846)$
$m_1 = 4.50676, m_2 = 2.81491, m_3 = 1.42763,$
$m_4 = 7.62154,$
$m_5 = 5.26917, m_6 = 3.52104, m_7 = 5.88815,$
$m_8 = 4.42252$

$P30$ $p_1 = (-2.50000, 0, 0, -2.50000)$
$p_2 = (-2.50000, 0, 0, 2.50000)$
$p_3 = (-0.42766, 0.04111, -0.18082, 0.38536)$
$p_4 = (-0.90714, 0.28930, 0.85932, 2.80593)$
$p_5 = (-0.41425, 0.32955, 0.24948, -0.02757)$
$p_6 = (-1.90735, -0.95093, -1.46021, 0.77557)$
$p_7 = (-0.27116, 0.15567, 0.03964, -0.21846)$
$m_1 = 4.50676, m_2 = 4.50676, m_3 = 4.50676,$
$m_4 = 4.50676,$
$m_5 = 4.50676, m_6 = 4.50676, m_7 = 4.50676,$
$m_8 = 4.50676$

6.1 Energy and mass scans

Figure 7 $p_1 = (27.95884, 25.55639, -29.88288, -2.17433)$
$p_2 = (27.45521, -7.81292, 3.19651, 6.05088)$
$6.05088 \leq m_1 = m_2 = m_3 \leq 31.53414$

Figure 8 $p_1 = (67.40483, 49.44993, -20.67085, 48.63654)$
$p_2 = (54.64295, -58.23071, 9.55042, -16.59411)$
$p_3 = (41.37620, 11.75178, -40.77655, -8.25014)$
$2.33822 \leq m_1 = m_2 = m_3 = m_4$
≤ 70.14658

Figure 9 $p_1 = \sqrt{\lambda}(-15.22437, -26.74156, 6.65483,$
$29.13661),$ $\lambda \in [1, 30],$
$p_2 = (-91.22611, -63.97875, 55.07507, -52.90153)$
$p_3 = (0.95105, 75.90791, -10.13814, -88.40860)$
$p_4 = (43.04908, 77.11321, -50.69469, -7.60198)$
$m_1 = 49.12560, m_2 = 57.87487, m_3 = 26.47098$
$m_4 = 0.42094, m_5 = 62.31320$

Figure 10 $p_1 = \sqrt{\lambda}(-51.76504, -81.75539, -46.42422,$
$-40.15540), \lambda \in [1, 30],$
$p_2 = (-63.76533, -2.53015, 16.27485, 69.16770)$
$p_3 = (-78.50262, 46.32052, 13.19246, -54.00166)$
$p_4 = (25.40582, 81.48058, 39.11105, 93.24648)$
$m_1 = 78.45208, m_2 = 42.71315, m_3 = 91.94256,$
$m_4 = 61.59730, m_5 = 16.75672$

References

1. S. Catani, M.H. Seymour, The Dipole formalism for the calculation of QCD jet cross-sections at next-to-leading order. Phys. Lett. B **378**, 287 (1996). arXiv:hep-ph/9602277
2. D.E. Soper, QCD calculations by numerical integration. Phys. Rev. Lett. **81**, 2638 (1998). arXiv:hep-ph/9804454
3. D.E. Soper, Techniques for QCD calculations by numerical integration. Phys. Rev. D **62**, 014009 (2000). arXiv:hep-ph/9910292
4. D.E. Soper, Choosing integration points for QCD calculations by numerical integration. Phys. Rev. D **64**, 034018 (2001). arXiv:hep-ph/0103262
5. M. Kramer, D.E. Soper, Next-to-leading order numerical calculations in Coulomb gauge. Phys. Rev. D **66**, 054017 (2002). arXiv:hep-ph/0204113
6. A. Ferroglia, M. Passera, G. Passarino, S. Uccirati, All purpose numerical evaluation of one loop multileg Feynman diagrams. Nucl. Phys. B **650**, 162 (2003). arXiv:hep-ph/0209219
7. Z. Nagy, D.E. Soper, General subtraction method for numerical calculation of one loop QCD matrix elements. JHEP **0309**, 055 (2003). arXiv:hep-ph/0308127
8. Z. Nagy, D.E. Soper, Numerical integration of one-loop Feynman diagrams for N-photon amplitudes. Phys. Rev. D **74**, 093006 (2006). arXiv:hep-ph/0610028
9. M. Moretti, F. Piccinini, A.D. Polosa, A fully numerical approach to one-loop amplitudes. arXiv:0802.4171 [hep-ph]

10. W. Gong, Z. Nagy, D.E. Soper, Direct numerical integration of one-loop Feynman diagrams for N-photon amplitudes. Phys. Rev. D **79**, 033005 (2009). arXiv:0812.3686 [hep-ph]

11. W. Kilian, T. Kleinschmidt, Numerical evaluation of Feynman loop integrals by reduction to tree graphs. arXiv:0912.3495 [hep-ph]

12. S. Becker, C. Reuschle, S. Weinzierl, Numerical NLO QCD calculations. JHEP **1012**, 013 (2010). arXiv:1010.4187 [hep-ph]

13. S. Becker, C. Reuschle, S. Weinzierl, Efficiency improvements for the numerical computation of NLO corrections. JHEP **1207**, 090 (2012). arXiv:1205.2096 [hep-ph]

14. S. Becker, S. Weinzierl, Direct contour deformation with arbitrary masses in the loop. Phys. Rev. D **86**, 074009 (2012). arXiv:1208.4088 [hep-ph]

15. G. Bevilacqua, M. Czakon, M.V. Garzelli, A. van Hameren, A. Kardos, C.G. Papadopoulos, R. Pittau, M. Worek, Helac-nlo. Comput. Phys. Commun. **184**, 986 (2013). arXiv:1110.1499 [hep-ph]

16. F. Cascioli, P. Maierhofer, S. Pozzorini, Scattering amplitudes with open loops. Phys. Rev. Lett. **108**, 111601 (2012). arXiv:1111.5206 [hep-ph]

17. G. Cullen et al., G*OSAM*-2.0: a tool for automated one-loop calculations within the Standard Model and beyond. Eur. Phys. J. C **74**(8), 3001 (2014). arXiv:1404.7096 [hep-ph]

18. T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert, J. Winter, Event generation with SHERPA 1.1. JHEP **0902**, 007 (2009). arXiv:0811.4622 [hep-ph]

19. S. Frixione, B.R. Webber, The MC and NLO 3.4 event generator. arXiv:0812.0770 [hep-ph]

20. J. Alwall et al., The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. JHEP **1407**, 079 (2014). arXiv:1405.0301 [hep-ph]

21. G. Passarino, An approach toward the numerical evaluation of multiloop Feynman diagrams. Nucl. Phys. B **619**, 257 (2001). arXiv:hep-ph/0108252

22. C. Anastasiou, S. Beerli, A. Daleo, Evaluating multi-loop Feynman diagrams with infrared and threshold singularities numerically. JHEP **0705**, 071 (2007). arXiv:hep-ph/0703282

23. S. Becker, S. Weinzierl, Direct numerical integration for multi-loop integrals. Eur. Phys. J. C **73**, 2321 (2013). arXiv:1211.0509 [hep-ph]

24. S. Catani, T. Gleisberg, F. Krauss, G. Rodrigo, J.C. Winter, From loops to trees by-passing Feynman's theorem. JHEP **0809**, 065 (2008). arXiv:0804.3170 [hep-ph]

25. G. Rodrigo, S. Catani, T. Gleisberg, F. Krauss, J.C. Winter, From multileg loops to trees (by-passing Feynman's Tree Theorem). Nucl. Phys. Proc. Suppl. **183**, 262 (2008). arXiv:0807.0531 [hep-th]

26. I. Bierenbaum, S. Catani, P. Draggiotis, G. Rodrigo, A tree-loop duality relation at two loops and beyond. JHEP **1010**, 073 (2010). arXiv:1007.0194 [hep-ph]

27. I. Bierenbaum, S. Buchta, P. Draggiotis, I. Malamos, G. Rodrigo, Tree-loop duality relation beyond simple poles. JHEP **1303**, 025 (2013). arXiv:1211.5048 [hep-ph]

28. I. Bierenbaum, P. Draggiotis, S. Buchta, G. Chachamis, I. Malamos, G. Rodrigo, News on the loop–tree duality. Acta Phys. Polon. B **44**, 2207 (2013)

29. S. Buchta, G. Chachamis, P. Draggiotis, I. Malamos, G. Rodrigo, On the singular behaviour of scattering amplitudes in quantum field theory. JHEP **1411**, 014 (2014). arXiv:1405.7850 [hep-ph]

30. S. Buchta, G. Chachamis, I. Malamos, I. Bierenbaum, P. Draggiotis, G. Rodrigo, The loop–tree duality at work. PoS LL **2014**, 066 (2014). arXiv:1407.5865 [hep-ph]

31. S. Buchta, Theoretical foundations and applications of the Loop-Tree Duality in Quantum Field Theories., PhD thesis, Universitat de València, 2015. arXiv:1509.07167 [hep-ph]

32. S. Buchta, G. Chachamis, P. Draggiotis, I. Malamos, G. Rodrigo, Towards a numerical implementation of the loop–tree duality method. Nucl. Part. Phys. Proc. **258–259**, 33 (2015). arXiv:1509.07386 [hep-ph]

33. R.J. Hernández-Pinto, G.F.R. Sborlini, G. Rodrigo, Towards gauge theories in four dimensions. JHEP **1602**, 044 (2016). arXiv:1506.04617 [hep-ph]

34. G.F.R. Sborlini, R. Hernández-Pinto, G. Rodrigo, From dimensional regularization to NLO computations in four dimensions. PoS **EPS-HEP2015**, 479 (2015). arXiv:1510.01079 [hep-ph]

35. G.F.R. Sborlini, F. Driencourt-Mangin, R. Hernández-Pinto, G. Rodrigo, Four-dimensional unsubtraction from the loop–tree duality. JHEP **1608**, 160 (2016). arXiv:1604.06699 [hep-ph]

36. G.F.R. Sborlini, F. Driencourt-Mangin, G. Rodrigo, Four-dimensional unsubtraction with massive particles. JHEP **1610**, 162 (2016). arXiv:1608.01584 [hep-ph]

37. F. Driencourt-Mangin, G. Rodrigo, G.F.R. Sborlini, Universal dual amplitudes and asymptotic expansions for $gg \to H$ and $H \to \gamma\gamma$ in four dimensions. arXiv:1702.07581 [hep-ph]

38. T. Hahn, M. Perez-Victoria, Automatized one loop calculations in four-dimensions and D-dimensions. Comput. Phys. Commun. **118**, 153 (1999). arXiv:hep-ph/9807565

39. T. Hahn, CUBA: a library for multidimensional numerical integration. Comput. Phys. Commun. **168**, 78 (2005). arXiv:hep-ph/0404043

40. J. Berntsen, T.O. Espelid, A. Genz, An adaptive algorithm for the approximate calculation of multiple integrals. ACM Trans. Math. Softw. **17**, 437–451 (1991)

41. J. Berntsen, T.O. Espelid, A. Genz, An adaptive multidimensional integration routine for a vector of integrals. ACM Trans. Math. Softw. **17**, 452–456 (1991)

42. G.P. Lepage, Vegas: An Adaptive Multidimensional Integration Program. Report No CLNS-80/447

43. Wolfram Research, Inc., Mathematica, Version 10.0 (2015)

44. S. Borowka, G. Heinrich, S.P. Jones, M. Kerner, J. Schlenk, T. Zirke, SecDec-3.0: numerical evaluation of multi-scale integrals beyond one loop. Comput. Phys. Commun. **196**, 470 (2015). arXiv:1502.06595 [hep-ph]

45. R. Kleiss, W.J. Stirling, S.D. Ellis, Comput. Phys. Commun. **40**, 359 (1986). doi:10.1016/0010-4655(86)90119-0