

# Numerical multi-loop calculations: tools and applications

S. Borowka<sup>1</sup>, G. Heinrich<sup>2</sup>, S. Jahn<sup>2</sup>, S. P. Jones<sup>2</sup>, M. Kerner<sup>2</sup>,  
J. Schlenk<sup>2</sup>, T. Zirke<sup>2</sup>

<sup>1</sup> Institute for Physics, University of Zürich, Winterthurerstr.190, 8057 Zürich, Switzerland

<sup>2</sup> Max Planck Institute for Physics, Föhringer Ring 6, 80805 München, Germany

E-mail: sborowka@physik.uzh.ch, gudrun@mpp.mpg.de, sjahn@mpp.mpg.de,  
sjones@mpp.mpg.de, kerner@mpp.mpg.de, jschlenk@mpp.mpg.de, zirke@mpp.mpg.de

**Abstract.** We report on the development of tools to calculate loop integrals and amplitudes beyond one loop. In particular, we review new features of the program SECDEC which can be used for the numerical evaluation of parametric integrals like multi-loop integrals.

## 1. Introduction

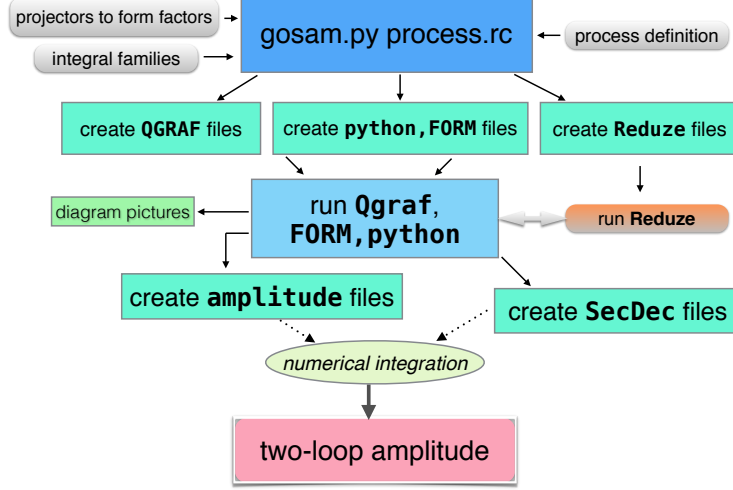
Precision calculations for the LHC and future colliders are entering more and more the domain of predictions beyond the next-to-leading order (NLO). The rapid progress was mainly triggered by the development of better calculational methods and tools, but also by the development of more powerful computing resources. Both aspects lay the groundwork allowing to move towards the automation of calculations beyond NLO.

In these proceedings we will focus on the automation of two-loop calculations, in particular two-loop integrals with several mass scales. While the latter are very difficult to calculate analytically, numerical methods may offer a solution. However, the numerical integration can only succeed if the various types of singularities which may occur in a two-loop amplitude can be isolated or evaded beforehand, preferably in an automated way. The program SECDEC [1–4] is able to perform this task, and to subsequently integrate the resulting finite parameter integrals numerically. In the following we will describe the program SECDEC and some of its new features, as well as its embedding into a framework to generate two-loop amplitudes.

## 2. Amplitude generation at one- and two loops

The steps to be performed in order to generate a virtual amplitude (at in principle any loop order) in an approach based on Feynman diagrams are roughly the following:

- (i) generation of the contributing diagrams in terms of algebraic expressions, based on the Feynman rules derived from a Lagrangian for a certain model,
- (ii) casting the expression into a form “coefficients  $\otimes$  (tensor) loop integrals”,
- (iii) reduction of the loop integrals to a set of master integrals (ideally, this set is a minimal basis in loop integral space) times coefficients, where the latter coefficients will depend on the dimension  $D$  of loop momentum space,



**Figure 1.** Building blocks of GOSAM-2loop. Boxes with rounded corners denote user input. The reduction to master integrals is run separately.

(iv) evaluation of the master integrals, combination with the coefficients forming the amplitude. We are constructing a program package which is designed to perform the steps listed above in an automated way. It partly builds on the program package GOSAM [5, 6] for automated one-loop calculations; for example it also uses QGRAF [7] for the diagram generation. However, the reduction step and the evaluation of the master integrals are much more challenging than at one loop, as the integral basis is usually not known beforehand, and the master integrals may not be available in analytic form. Within the GOSAM-2loop framework, we constructed an interface to the reduction program REDUZE [8, 9]. The reduction to master integrals is then run separately, and the resulting database is imported back into GOSAM-2loop. The latter also contains an interface to the program SECDEC which is used to evaluate the integrals numerically. The flowchart for GOSAM-2loop, shown in Fig. 1, illustrates the main steps.

### 3. The program SecDec

The program SECDEC [1–4] is based on the sector decomposition algorithm described in [10, 11]. It can be applied to multi-loop integrals as well as more general parametric integrals which contain dimensionally regulated end-point singularities. Other public implementations of sector decomposition are described in [12–16].

#### 3.1. Feynman Integrals

We consider scalar multi-loop integrals in order to simplify the notation. The case of tensor integrals is analogous, for details we refer to [11]. A scalar Feynman integral  $I$  in  $D$  dimensions at  $L$  loops with  $N$  propagators, where the propagators can have arbitrary, not necessarily integer powers  $\nu_j$ , has the following representation in momentum space:

$$I = \frac{\mu^{4-D}}{i\pi^{\frac{D}{2}}} \int d^D k_l \frac{1}{\prod_{j=1}^N P_j^{\nu_j}(\{k\}, \{p\}, m_j^2)},$$

$$P_j(\{k\}, \{p\}, m_j^2) = q_j^2 - m_j^2 + i\delta, \quad (1)$$

where the  $q_j$  are linear combinations of external momenta  $p_i$  and loop momenta  $k_l$ . Introducing Feynman parameters and carrying out the momentum integrations leads to

$$G = \frac{(-1)^{N_\nu} \Gamma(N_\nu - LD/2)}{\prod_{j=1}^N \Gamma(\nu_j)} \int_0^\infty \prod_{j=1}^N dx_j x_j^{\nu_j-1} \delta(1 - \sum_{l=1}^N x_l) \frac{\mathcal{U}(\vec{x})^{N_\nu - (L+1)D/2}}{\mathcal{F}(\vec{x}, \{s_{ij}, m_i^2\})^{N_\nu - LD/2}}, \quad (2)$$

with  $N_\nu = \sum_{j=1}^N \nu_j$ . The graph polynomial  $\mathcal{U}$  is a positive-semidefinite function of the Feynman parameters only, while  $\mathcal{F}$  is a function of Feynman parameters and the kinematic invariants characterising the integral. In the region where all Lorentz invariants formed from external momenta are negative, which we call the *Euclidean region*,  $\mathcal{F}$  is also a positive semi-definite function of the Feynman parameters  $x_j$  and the invariants. If some of the invariants (masses, squared external momenta) are zero, the vanishing of  $\mathcal{F}$  may lead to an infrared (soft or collinear) divergence. Thus it depends on the *kinematics* and not only on the topology (as in the ultraviolet case) whether a zero of  $\mathcal{F}$  leads to a divergence. If some of the kinematic invariants are positive,  $\mathcal{F}$  can also vanish at points in Feynman parameter space which are not the endpoints of the parametric integration. To be able to deal with such thresholds (or pseudo-thresholds), a deformation of the integration contour into the complex plane has been implemented [2].

The program SECDEC can construct the graph polynomials  $\mathcal{U}$  and  $\mathcal{F}$  automatically. Acting on polynomials raised to some power which usually contains the dimensional regulator  $\epsilon$ , it will extract the poles in  $1/\epsilon$  and evaluate their coefficients numerically, as explained in the next subsection.

### 3.2. Basic structure of the program

The program consists of two main parts, one being designed for loop integrals, to be found in a directory called `loop`, the other one for more general parametric integrals, located in a directory called `general`. The procedure to isolate the poles in the regulator  $\epsilon$  and to do the subtractions and integrations is very similar in the two branches. However, only the `loop` part contains the possibility of contour deformation, because only for loop integrals the analytic continuation can be performed in an automated way, following Feynman's "*i* $\delta$ " prescription. The basic flowchart of the program is shown in Fig. 2.

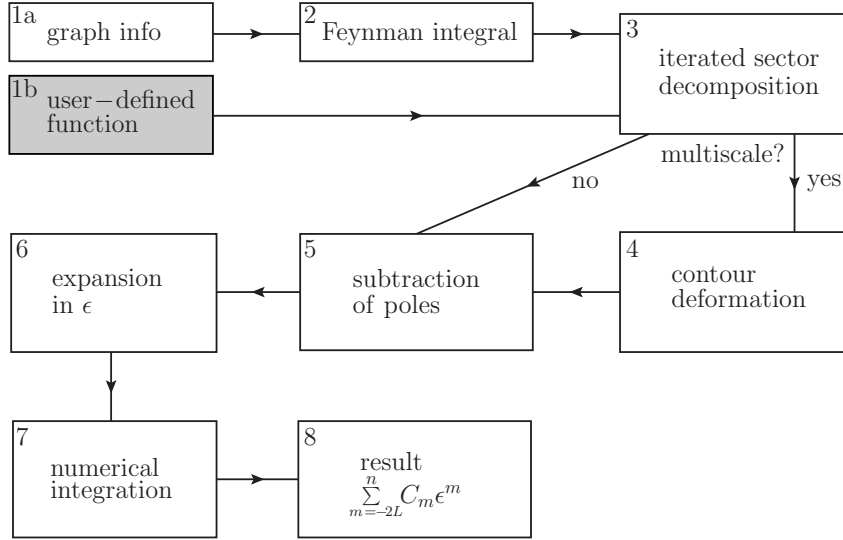
### 3.3. New features

SECDEC-3.0 [4] has a number of new features, aiming more towards applications in which the program is used as part of the calculation of a full virtual amplitude, rather than serving only as a numerical check for master integrals which are calculated analytically. Among these features are:

- The usage on a cluster has been facilitated. This has been achieved by decoupling the parts where Mathematica is needed completely from the numerical part, and by creating a directory structure such that the files for the numerical integration created by SECDEC can be transferred to a cluster, where they are compiled and run. Run scripts are provided for the PBS, LSF and `Condor` batch systems. (The scripts for the LSF system are only available in the most recent version 3.0.9.)

The kinematics can be defined in a simple text file `kinem.input` containing the numerical values for the invariants occurring in the integral, where the program will loop over all kinematic points defined in `kinem.input`. This new way to define the kinematic points to be calculated is also available in single machine mode.

- Integrals with numerators can be calculated not only as contracted tensor integrals, but also by defining the numerators as inverse propagators. Further, the list of propagator powers



**Figure 2.** Flowchart showing the main steps the program performs to produce the numerical result as a Laurent series in  $\epsilon$ .

can also contain zero entries. These features allow an easy interface to integral reduction programs as e.g. **Reduze** [9], where the propagator list defines the integral family, while the list of propagator powers defines the considered integral within this family.

- Two new decomposition strategies, based on algebraic geometry [17, 18], have been implemented. These strategies by construction cannot run into an infinite recursion at the iterated decomposition. Remarkably, the strategy **G2** often leads to a lower number of subsectors than the “default” strategy **X**. This new feature is described in detail in [19].
- The masses contained in the propagators can also be complex. For details we also refer to [19].
- Integrals with propagators which are only linear in the loop momentum can be calculated as well. If these integrals are non-Euclidean, care has to be taken to ensure the correct sign for the contour deformation. The program will assume the same  $i\delta$  prescription as for Feynman propagators which are quadratic in the loop momenta.
- A new treatment of spurious poles which have a worse than logarithmic divergence behaviour (related e.g. to factors of  $x^{-2-\epsilon}$  in the decomposed integrand) is available, which uses integration by parts (IBP) to remap these poles to logarithmic poles. This option is invoked if the parameter **IBPflag** is nonzero.
- The user can define thresholds below which the imaginary part of the integral is known to be zero; in this case **SECDEC** will not attempt to calculate the imaginary part. This can speed up the calculation considerably.
- For the numerical integration, the user can also choose to use an integrator from Mathematica. For one-dimensional integrals, the program automatically switches to **CQUAD** [20], being faster and more accurate than Monte Carlo integration.
- The user interface has been facilitated, decoupling completely the following three types of input: (a) definition of the graph name, desired expansion order in  $\epsilon$  and parameters for the numerical integration (**param.input**), (b) graph definition (**math.m**), (c) definition of the kinematics (**kinem.input**).

The next release of SECDEC will also contain the possibility to create a library containing the sector functions produced by the program, such that the integrals can be linked to the calculation of a full amplitude. This will allow to adjust the precision to which an individual integral is calculated according to its importance within the full amplitude.

Finally, for SECDEC-4.0, the algebraic part will be completely restructured, and will be implemented in `python` rather than `Mathematica`. This will be described in more detail in the following subsection.

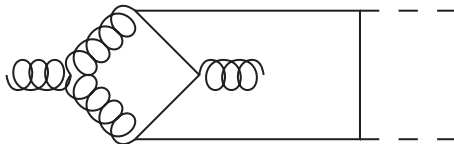
### 3.4. Implementation in python

The forthcoming implementation of the algebraic part of SECDEC in `python` has several advantages. One of them is certainly given by the fact that `python` is an open source software. In addition, the new implementation offers much more modularity. The various steps the program performs, like construction of the graph polynomials, iterated decomposition, subtraction,  $\epsilon$ -expansion (see also Fig. 2), are available in the form of `python` modules and can be called separately. The decomposition can act on products of polynomials raised to some power, so it is not limited to a loop integral structure of the form  $\mathcal{F}^{\text{expoF}}\mathcal{U}^{\text{expoU}}$  times a numerator (see Eq. (2)). Further, the new implementation allows to perform an expansion in several regulators, not only the dimensional regulator  $\epsilon$ . The additional regulator could stem for example from analytic regularisation [21] or from parameter integrals containing the string tension [22].

An alpha-version of the `python` implementation can be found at <http://secdec.hepforge.org>, where a manual is also linked.

## 4. Applications: Two-loop four-point integrals with two mass scales

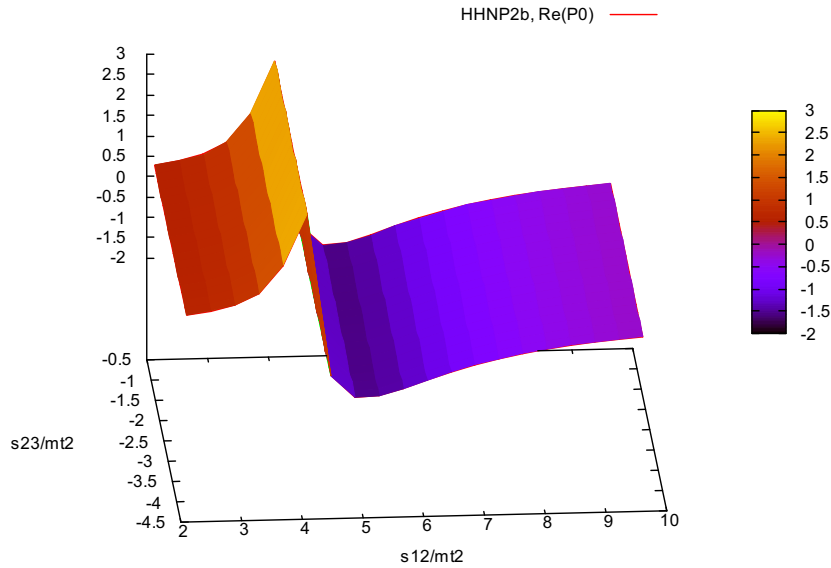
As an example for an integral where the analytic result is unknown, we give results for a non-planar two-loop four-point function (see Fig. 3) containing both internal masses ( $m_t$ ) and massive external legs with a different mass ( $m_H$ ). The result for this diagram is of the form  $I = P_1/\epsilon + P_0$ . Numerical results in the form of an interpolated scan over 20 values for  $s_{12}/m_t^2$  times 20 values for  $s_{23}/m_t^2$  for the real part of the finite term  $P_0$  are shown in Fig. 4.



**Figure 3.** Example of a non-planar two-loop box diagram with two mass scales. Solid lines denote massive propagators ( $m_t$ ). The dashed lines denote massive legs with a mass ( $m_H$ ) different from the internal mass.

## 5. Conclusions

We have presented new features of the program SECDEC, which is a program to factorise poles from dimensionally regulated parameter integrals, and to evaluate the pole coefficients numerically. The embedding of SECDEC into GOSAM-2loop, a general framework for the calculation of 2-loop amplitudes, has also been briefly described. The new features of SECDEC and further developments of these programs aim at the usage of SECDEC within the calculation of full two-loop virtual amplitudes in cases where the analytic result for most of the master integrals is unknown, rather than the calculation of just a few kinematic points to check an analytic calculation of individual integrals. This development is in line with the rapid developments in high performance computing.



**Figure 4.** Real part of the finite part of the graph shown in Fig. 3, for the values  $m_t = 173$  GeV,  $m_H = 125$  GeV,  $u = -s - t + 2m_H^2$ .

## Acknowledgments

GH would like to thank the organizers of ACAT2016 for the nice conference. This research was supported in part by the Research Executive Agency (REA) of the European Union under the Grant Agreement PITN-GA2012316704 (HiggsTools). S. Borowka gratefully acknowledges financial support by the ERC Advanced Grant MC@NNLO (340983).

## References

- [1] Carter J and Heinrich G 2011 *Comput.Phys.Commun.* **182** 1566–1581 (*Preprint* 1011.5493)
- [2] Borowka S, Carter J and Heinrich G 2013 *Comput.Phys.Commun.* **184** 396–408 (*Preprint* 1204.4152)
- [3] Borowka S and Heinrich G 2013 *Comput.Phys.Commun.* **184** 2552–2561 (*Preprint* 1303.1157)
- [4] Borowka S, Heinrich G, Jones S P, Kerner M, Schlenk J and Zirke T 2015 *Comput. Phys. Commun.* **196** 470–491 (*Preprint* 1502.06595)
- [5] Cullen G, Greiner N, Heinrich G, Luisoni G, Mastrolia P, Ossola G, Reiter T and Tramontano F 2012 *Eur. Phys. J.* **C72** 1889 (*Preprint* 1111.2034)
- [6] Cullen G *et al.* 2014 *Eur. Phys. J.* **C74** 3001 (*Preprint* 1404.7096)
- [7] Nogueira P 1993 *J. Comput. Phys.* **105** 279–289
- [8] Studerus C 2010 *Comput. Phys. Commun.* **181** 1293–1300 (*Preprint* 0912.2546)
- [9] von Manteuffel A and Studerus C 2012 (*Preprint* 1201.4330)
- [10] Binoth T and Heinrich G 2000 *Nucl. Phys.* **B585** 741–759 (*Preprint* hep-ph/0004013)
- [11] Heinrich G 2008 *Int. J. Mod. Phys.* **A23** 1457–1486 (*Preprint* 0803.4177)
- [12] Bogner C and Weinzierl S 2008 *Comput. Phys. Commun.* **178** 596–610 (*Preprint* 0709.4092)
- [13] Gluza J, Kajda K, Riemann T and Yundin V 2011 *Eur.Phys.J.* **C71** 1516 (*Preprint* 1010.1667)
- [14] Smirnov A and Tentyukov M 2009 *Comput.Phys.Commun.* **180** 735–746 (*Preprint* 0807.4129)
- [15] Smirnov A V 2014 *Comput.Phys.Commun.* **185** 2090–2100 (*Preprint* 1312.3186)
- [16] Smirnov A V 2015 (*Preprint* 1511.03614)
- [17] Kaneko T and Ueda T 2010 *Comput. Phys. Commun.* **181** 1352–1361 (*Preprint* 0908.2897)
- [18] Kaneko T and Ueda T 2010 (*Preprint* 1004.5490)

- [19] Schlenk J and Zirke T 2016 *Proceedings, 12th International Symposium on Radiative Corrections (Radcor 2015) and LoopFest XIV (Radiative Corrections for the LHC and Future Colliders)* (Preprint 1601.03982)
- [20] Gonnet P 2010 *CoRR* **abs/1006.3962** URL <http://arxiv.org/abs/1006.3962>
- [21] Becher T and Bell G 2012 *Phys. Lett.* **B713** 41–46 (Preprint 1112.3907)
- [22] Puhlfürst G and Stieberger S 2016 *Nucl. Phys.* **B902** 186–245 (Preprint 1507.01582)