

PAPER • OPEN ACCESS

Amplitude evaluation with pySecDec: A Higgs + three gluons example

To cite this article: C P Paranjape *et al* 2023 *J. Phys.: Conf. Ser.* **2438** 012136

View the [article online](#) for updates and enhancements.

You may also like

- [Commensurability condition and hierarchy of fillings for FQHE in higher Landau levels in conventional 2DEG systems and in graphene—monolayer and bilayer](#)
Janusz Jacak and Lucjan Jacak
- [Les Houches 2021—physics at TeV colliders: report on the standard model precision wishlist](#)
Alexander Huss, Joey Huston, Stephen Jones et al.
- [Calculation and modular properties of multi-loop superstring amplitudes](#)
G S Danilov



245th ECS Meeting
San Francisco, CA
May 26–30, 2024

PRiME 2024
Honolulu, Hawaii
October 6–11, 2024

Bringing together industry, researchers, and government across 50 symposia in electrochemistry and solid state science and technology

Learn more about ECS Meetings at
<http://www.electrochem.org/upcoming-meetings>

 Save the Dates for future ECS Meetings!

Amplitude evaluation with pySecDec: A Higgs + three gluons example

C P Paranjape¹, G Heinrich² and S P Jones³

¹ Department of Physics, Indian Institute of Technology (Indian School of Mines) Dhanbad, Jharkhand 826004, India

² Institute for Theoretical Physics, Karlsruhe Institute of Technology (KIT), 76128 Karlsruhe, Germany

³ Institute for Particle Physics Phenomenology, Durham University, Durham DH1 3LE, UK

E-mail: ¹ chaitanyaparanjape.18je0240@ap.iitism.ac.in, ² gudrun.heinrich@kit.edu, ³ stephen.jones@durham.ac.uk

Abstract. We present an application of the major new features of pySECDEC, a program designed to numerically calculate multi-loop integrals. One important new feature is the ability to integrate weighted sums of integrals in a way which is optimised to reach a given accuracy goal on the sums rather than on the individual integrals. Another feature is the option to perform asymptotic expansions. These new assets of the program represent an important step towards the efficient evaluation of multi-loop amplitudes in a largely automated way. We will illustrate them in a pedagogical example, through the calculation of the one-loop amplitude for Higgs plus jet production in the gluon channel, mediated by a top quark loop. Numerical results in the heavy top limit, compared to their analytic counterparts, are shown as well.

1. Introduction

Precision calculations are essential for testing and advancing our knowledge of high energy physics. Theoretical predictions need to be precise enough to match the increasing experimental precision and to look for discrepancies hinting towards new physics directions. In light of the data expected at the Large Hadron Collider and potential future colliders, higher order corrections for relevant processes must be evaluated. However, with higher numbers of loops and mass scales, the complexity of such problems increases substantially. Thus, opting for numerical evaluations of the integrals, as well as approximations in certain limits, can often be an efficient way to provide sufficiently accurate phenomenological results.

The program pySECDEC[1, 2] has been further developed towards this goal [3], a detailed description can be found in the contribution of V. Magerya [4]. The purpose of these proceedings is to present two new features resulting from this development with a concrete example. We will describe an explicit demonstration of how to evaluate weighted sums of integrals and how to perform expansion by regions [5, 6]. The implementation of these features represents a step towards the automation of multi-loop calculations. Here we illustrate the methodology by considering the example $gg \rightarrow gH$ at 1-loop, which is part of the amplitude for Higgs plus jet production at the LHC. Full code for this example is available in the secdec repository, see `nodist_examples/gggH1L`.



2. New features of pySecDec

2.1. *sum_package*

The python package `sum_package` can be used to perform the numerical evaluation of weighted sums of Feynman integrals as they occur, for example, in calculations of scattering amplitudes. The evaluation procedure is optimised to reach a given accuracy goal on the sum rather than on the individual integrals. It takes as input a list of regularised integrals through the `package_generators`, which can be loop integrals (defined via `loop_package`) or more general parametric integrals (defined via `make_package`). The `coefficients` (rational functions) can be defined in a separate file, requiring the specification of numerators and denominators for all coefficients. These coefficients depend on the defined `parameters` and `regulators`. An example is shown below.

```
sum_package(
    'F212',
    package_generators,
    regulators = ['eps'],
    requested_orders = [0],
    real_parameters = ['s12', 's13', 's23', 'mtsq', 'mhsq'],
    coefficients = coefficientsF212.coeff,
    complex_parameters = []
)
```

The function generates a library `F212`, the coefficients are contained in the file `coefficientsF212`. They are rational polynomials of the kinematic invariants and the regulators, and they need to be given in the same order as the integrals in `package_generators`. An example of a such file is shown below where the `M1num` and `M1den` are the numerator and denominator of the `P1` coefficient which can be defined appropriately (e.g. `M1num = ['-(16*eps*(s13*s23))']`). The `parameters` should contain all the real and complex parameters defined above.

```
coeff = [
    # P1   g1(mtsq)  coefficient
    [Coefficient(M1num,M1den,parameters),
    ...
    # P14  g9(s23,s13,mhsq,mtsq) coefficient
    Coefficient(M14num,M14den,parameters)]
]
```

The loop integrals can be defined as an ordered list in a standard manner with pySECDEC methods from `loop_package` like `LoopIntegralFromGraph` or `LoopIntegralFromPropagators`. We use the latter and define all the integrals based on the same list of propagators, adjusting their powers correspondingly in `powerlist`.

The algorithm in `sum_package` attempts to obtain the overall precision requested by the user while evaluating the superposition of given integrals with their coefficients. This avoids spending too much time on integrals which do not give a large contribution to the full amplitude.

2.2. *Expansion by regions*

Expansion by regions is a systematic method for expressing Feynman integrals as a series expansion in terms of a ratio of scales, summed over different regions defined by a scale hierarchy. The method has been pioneered in Refs. [7, 5, 8] and further developed in Refs. [9, 6, 10, 11]. A standard example [6] is the integral :

$$I_2 = \mu^{2\epsilon} \int d^d k \frac{1}{(k+p)^2 (k^2 - m^2)^2} \quad (1)$$

Considering the case $|p^2| \gg m^2$, we can expand the integrand in two regions, hard ($k \sim p$) and soft ($k \sim m$).

$$\begin{aligned}
 \text{(hard)} : \quad & \frac{1}{(k+p)^2(k^2-m^2)^2} \rightarrow \frac{1}{(k+p)^2(k^2)^2} \left(1 + \frac{2m^2}{k^2} + \dots\right) \quad \text{for } |m^2| < |k^2| \\
 \text{(soft)} : \quad & \frac{1}{(k+p)^2(k^2-m^2)^2} \rightarrow \frac{1}{p^2(k^2-m^2)^2} \left(1 - \frac{k^2 + 2p \cdot k}{p^2} + \dots\right) \quad \text{for } k^2 + 2k \cdot p < p^2
 \end{aligned} \tag{2}$$

Integrating all the terms over the complete integration domain, there will be no double counting because the ‘overlap’ terms are scaleless integrals which are zero in dimensional regularisation (Working in $d = 4 - 2\varepsilon$ dimensions).

Expansion by regions can be implemented for loop integrals `li_z` through the `loop_regions` method. The scale hierarchy can be defined by multiplying the invariants of the standard loop integral `li` appropriately with a `smallness_parameter` $\sim z$. Expansion about $z=0$ up to a user-defined order is performed, then `sum_package` sums over all the regions (with $z=1$).

```

li_z = psd.loop_integral.LoopIntegralFromPropagators(
propagators=li.propagators, loop_momenta=li.loop_momenta,
regulators=li.regulators, powerlist=li.powerlist,
replacement_rules = [ ('p1*p1', '0'), ('p2*p2', '0'), ('p3*p3', '0'),
('p1*p2', '(s12*z)/2'), ('p1*p3', '(s13*z)/2'), ('p2*p3', '(s23*z)/2'),
('mt**2', 'mts'), ('(p1*p2)+(p2*p3)+(p1*p3)', '((mhsq)*z)/2') ]

generators_args = loop_regions(
name = "g9(s23,s13,mhsq,mts)",
loop_integral=li_z,
smallness_parameter = "z",
expansion_by_regions_order=1)

sum_package(name, generators_args, li_z.regulators,
requested_orders = [1],
real_parameters = ['s12', 's13', 's23', 'mts', 'mhsq', 'z'],
complex_parameters = [])

```

For integrals with large scale differences between its invariants, standard pySECDEC can face convergence problems due to large cancellations. In the case of large scale hierarchies, the uncertainty introduced through the truncation of the series in the expansion by regions method can be considerably lower and therefore expansion by regions is the method of choice in such cases. For more details we refer to [3] or section 3.2.7 of [12].

3. Application: Higgs plus three-gluon amplitude at one loop

Now we illustrate the application of the above features by calculating an example from Higgs + jet production in the gluon channel at 1-loop. Although there are 7 different partonic channels that contribute, the calculational methodology for the fixed order matrix elements is similar for all sub-processes. Thus, we consider specifically the $gg \rightarrow gH$ process here. A more complete discussion of this amplitude can be found in [13].

Exploiting the tensor structure of the amplitude and imposing all the necessary constraints like parity and gauge invariance, the absolute square of the matrix element can be expressed in terms of four form factors, F_{212} , F_{312} , F_{332} and F_{311} . Additionally, the form factors F_{212} , F_{332} and F_{311} are related due to the permutation invariance of the amplitude under the exchange of

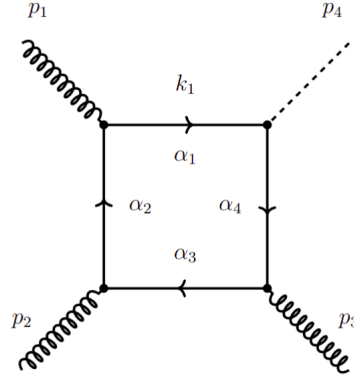


Figure 1. $gg \rightarrow gH$ mediated by a top quark loop

gluons. The complete amplitude thus can be expressed in terms of the two independent form factors F_{212} and F_{312} . These form factors depend on 9 master integrals related to our topology and 5 crossed integrals. Therefore the amplitude can be expressed as a linear combination of these 14 integrals : $F = \sum_{i=1}^{14} c_i J_i$ where the J_i can be written as:

$$\begin{aligned}
 1. \quad g_1(m_t^2) &= I_{2,0,0,0} & 6. \quad g_5(s_{12}, m_t^2) &= I_{1,1,1,0} & 11. \quad g_8(s_{23}, m_H^2, m_t^2) &= I_{1,1,0,1} \\
 2. \quad g_2(s_{12}, m_t^2) &= I_{2,0,1,0} & 7. \quad g_5(s_{13}, m_t^2) &= I_{1,1,1,0} & 12. \quad g_9(s_{12}, s_{23}, m_H^2, m_t^2) &= I_{1,1,1,1} \\
 3. \quad g_2(s_{13}, m_t^2) &= I_{2,0,1,0} & 8. \quad g_6(s_{23}, m_t^2) &= I_{0,1,1,1} & 13. \quad g_9(s_{12}, s_{13}, m_H^2, m_t^2) &= I_{1,1,1,1} \\
 4. \quad g_3(s_{23}, m_t^2) &= I_{0,2,0,1} & 9. \quad g_7(s_{12}, m_H^2, m_t^2) &= I_{1,0,1,1} & 14. \quad g_9(s_{23}, s_{13}, m_H^2, m_t^2) &= I_{1,1,1,1} \\
 5. \quad g_4(m_H^2, m_t^2) &= I_{2,0,0,1} & 10. \quad g_7(s_{13}, m_H^2, m_t^2) &= I_{1,0,1,1} , & &
 \end{aligned} \tag{3}$$

where

$$I_{\alpha_1, \alpha_2, \alpha_3, \alpha_4} = \int \frac{d^d k_1}{i\pi^{d/2}} \frac{1}{D_1^{\alpha_1} D_2^{\alpha_2} D_3^{\alpha_3} D_4^{\alpha_4}} \tag{4}$$

with the propagators defined appropriately. The kinematic invariants are defined as $s_{ij} = (p_i + p_j)^2$. The form factors are related by $F_{212}(s_{12}, s_{13}, s_{23}) = F_{311}(s_{13}, s_{23}, s_{12}) = F_{332}(s_{23}, s_{12}, s_{13})$, such that only $F_{212}(s_{12}, s_{13}, s_{23})$ and $F_{312}(s_{12}, s_{13}, s_{23})$ are independent.

The amplitude is most conveniently expressed in terms of helicity amplitudes. As each gluon can have two helicities, there are in total 2^3 helicity amplitudes. Due to the parity and permutation invariance, we again have only two independent helicity amplitudes, corresponding to the two independent form factors, related by:

$$\begin{aligned}
 \mathcal{M}_{gg \rightarrow gH}^{h_1 h_2 h_3} &= -\mathcal{M}_{gg \rightarrow gH}^{-h_1 -h_2 -h_3} \\
 \mathcal{M}_{gg \rightarrow gH}^{++-}(s_{12}, s_{13}, s_{23}) &= \mathcal{M}_{gg \rightarrow gH}^{+++}(s_{13}, s_{23}, s_{12}) = \mathcal{M}_{gg \rightarrow gH}^{--+}(s_{23}, s_{12}, s_{13}) \\
 \mathcal{M}_{gg \rightarrow gH}^{+++} &= \frac{\sqrt{s_{12} s_{13} s_{23}}}{\sqrt{2}} \cdot \left(\frac{s_{12}}{2s_{13}} F_{212} + \frac{s_{23}}{2s_{12}} F_{332} + \frac{s_{13}}{2s_{23}} F_{311} + F_{312} \right) \\
 \mathcal{M}_{gg \rightarrow gH}^{++-} &= \sqrt{\frac{s_{12} s_{23}}{s_{13}}} \cdot \frac{s_{12}}{2\sqrt{2}} \cdot F_{212} .
 \end{aligned} \tag{5}$$

The last two expressions are valid only up to an overall phase factor. Thus, the absolute square of the matrix element is given by:

$$|\mathcal{M}_{gg \rightarrow gH}|^2 = 2 \cdot \left(|\mathcal{M}_{gg \rightarrow gH}^{+++}|^2 + |\mathcal{M}_{gg \rightarrow gH}^{++-}|^2 + |\mathcal{M}_{gg \rightarrow gH}^{--+}|^2 + |\mathcal{M}_{gg \rightarrow gH}^{---}|^2 \right) . \tag{6}$$

We also give the analytic expression for the amplitude squared in heavy top limit [13], which we will use to compare with our numerical results including the top-quark mass dependence, in the limit of large top quark mass,

$$|\mathcal{M}_{gg \rightarrow gH}^{m_t \rightarrow \infty}|^2 = \frac{4}{9} \cdot \frac{N_c(N_c^2 - 1)\alpha_s^3}{\pi v^2} \cdot \frac{(m_H^8 + s_{12}^4 + s_{13}^4 + s_{23}^4)}{s_{12}s_{13}s_{23}} + \mathcal{O}(\epsilon) \quad (7)$$

For convenience, we normalise the amplitude by extracting a factor of K , defined by

$$|\mathcal{M}_{gg \rightarrow gH}^{m_t \rightarrow \infty}|^2 = K |\mathcal{A}_{gg \rightarrow gH}^{m_t \rightarrow \infty}|^2, \quad K = \frac{N_c(N_c^2 - 1)m_t^4\alpha_s^3}{\pi v^2}, \quad (8)$$

such that the quantity we use to check our numerical results is

$$|\mathcal{A}_{gg \rightarrow gH}^{m_t \rightarrow \infty}|^2 = \frac{4}{9} \frac{1}{m_t^4} \cdot \frac{(m_H^8 + s_{12}^4 + s_{13}^4 + s_{23}^4)}{s_{12}s_{13}s_{23}} + \mathcal{O}(\epsilon). \quad (9)$$

4. Numerical evaluation with pySecDec

4.1. Using `sum_package`

Because the form factors are expressed as weighted sums of integrals, $F = \sum_{i=1}^{14} c_i J_i$, we can use `sum_package` to numerically evaluate the sum directly, for both the form factors as well as the squared helicity amplitudes represented as sums of form factors. Using the `sum_package` syntax shown above, we can generate the library files for each of the two independent form factors. To integrate, we load the library file for F_{212} and F_{312} (`F212_pylink.so`, `F312_pylink.so`), choose an integrator (e.g. `Qmc` or `Vegas`) and provide the kinematic point to evaluate the integral at. The amplitude is evaluated with a relative numerical accuracy of $\sim 10^{-7}$ in 5-10 seconds with a regular CPU@1.6 GHz.

4.2. Expansion by regions

Let us now demonstrate the evaluation of the master integrals by expanding them in inverse powers of m_t^2 , which will act as the smallness parameter in the heavy top limit. Analytic expression for the first order expansion of the integral $g_2(s_{12}, m_t^2)$ can be written as [13]:

$$g_2(s_{12}, m_t^2) = (-1)^3 \Gamma(3 - d/2) \frac{1}{m_t^2} \left(\frac{\mu^2}{m_t^2}\right)^{2-d/2} \cdot \left(\frac{1}{2} + \frac{s_{12}}{m_t^2} \cdot \frac{3-d/2}{12} + \mathcal{O}\left(\frac{1}{m_t^4}\right)\right) \quad (10)$$

Following the procedure outlined earlier, we can numerically evaluate the integral with expansion by regions, at a kinematic point where all invariants are small compared to the squared top mass:

$$[s_{12}, s_{13}, s_{23}, m_t^2, m_H^2] = [0.0009, -0.0003, -0.000442873775, 1.00, 0.000157126225]$$

The explicit conditions we use for the Integration are:

```
expansion_by_regions_order = 1, requested_orders = [1], z = 1,
Integrator = Qmc, transform="korobov3", fitfunction="polysingular" .
```

Refer to the Table 1 for integration results. We observe a very small numerical uncertainty and perfect agreement with the analytic results, illustrating also the ease of using expansion by regions method within pySECDEC.

Similarly, the other integrals composing the amplitude can be evaluated in the heavy top limit using `expansion by regions`. It is known that during the expansion by regions procedure, some integrals may not be fully regulated by dimensional regularisation at intermediate

Table 1. Integration results of $g_2(s_{12}, m_t^2)$

Coefficient of	Analytic result in HTL	Full Numerical result	Numerical uncertainty
ϵ^0	-0.5000749999999999	-0.5000750000000000	$\sim 10^{-17}$
ϵ^1	0.288576123625634	0.288576123625634	$\sim 10^{-17}$

stages and therefore an additional regulator needs to be introduced. In our case $g_5(s_{12}, m_t^2)$ shows such a behaviour. Additional regulators can be used straightforwardly in pySECDEC. We introduce the additional regulator `n1` to regulate these spurious singularities by additional propagator powers, explicitly we use `powerlist=[1+n1, 1+ n1/2, 1+n1/3, 0+n1/5]`. For more details about these regulators, we refer to [14, 15, 16, 3].

This completes the demonstration of `sum_package` and `expansion by regions` within pySECDEC for the evaluation of the $gg \rightarrow gH$ amplitude at 1-loop. Even though an easy example has been shown here for demonstration purposes, the method can be applied straightforwardly to more complicated amplitudes and therefore is a step towards the automation of the calculation of multi-loop scattering amplitudes.

Acknowledgments

CPP would like to thank the German Academic Exchange Service (DAAD) for offering support with the DAAD WISE Scholarship. This research was supported in parts by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant 396021762 - TRR 257. We would like to thank our pySECDEC collaborators Matthias Kerner, Vitaly Magerya, Andres Poldaru and Johannes Schlenk.

References

- [1] Borowka S, Heinrich G, Jahn S, Jones S P, Kerner M, Schlenk J and Zirke T 2018 *Comput. Phys. Commun.* **222** 313–326 (*Preprint hep-ph/1703.09692*)
- [2] Borowka S, Heinrich G, Jahn S, Jones S P, Kerner M and Schlenk J 2019 *Comput. Phys. Commun.* **240** 120–137 (*Preprint comp-ph/1811.11720*)
- [3] Heinrich G, Jahn S, Jones S P, Kerner M, Langer F, Magerya V, Pöldaru A, Schlenk J and Villa E 2022 *Comput. Phys. Commun.* **273** 108267 (*Preprint hep-ph/2108.10807*)
- [4] Magerya V 2022 (*Preprint hep-ph/2202.13647*)
- [5] Beneke M and Smirnov V A 1998 *Nucl. Phys. B* **522** 321–344 (*Preprint hep-ph/9711391*)
- [6] Jantzen B 2011 *JHEP* **12** 076 (*Preprint hep-ph/1111.2589*)
- [7] Smirnov V A 1991 *Renormalization and asymptotic expansions* vol 14
- [8] Smirnov V A and Rakhmetov E R 1999 *Theor. Math. Phys.* **120** 870–875 (*Preprint hep-ph/9812529*)
- [9] Pak A and Smirnov A 2011 *Eur. Phys. J. C* **71** 1626 (*Preprint 1011.4863*)
- [10] Ananthanarayan B, Pal A, Ramanan S and Sarkar R 2019 *Eur. Phys. J. C* **79** 57 (*Preprint 1810.06270*)
- [11] Semenova T Y, Smirnov A V and Smirnov V A 2019 *Eur. Phys. J. C* **79** 136 (*Preprint hep-th/1809.04325*)
- [12] Heinrich G 2021 *Phys. Rept.* **922** 1–69 (*Preprint hep-ph/2009.00516*)
- [13] Boggia M *et al.* 2018 *J. Phys. G* **45** 065004 (*Preprint hep-ph/1711.09875*)
- [14] Smirnov V A 1997 *Phys. Lett. B* **404** 101–107 (*Preprint hep-ph/9703357*)
- [15] Becher T and Neubert M 2011 *Eur. Phys. J. C* **71** 1665 (*Preprint hep-ph/1007.4005*)
- [16] Becher T, Broggio A and Ferroglia A 2015 vol 896 (Springer) (*Preprint hep-ph/1410.1892*)