pySecDec as a tool for expansion by regions and loop amplitude evaluation

Emilio Villa*

Institute for Theoretical Physics, Karlsruhe Institute of Technology (KIT),
76131 Karlsruhe, Germany

$\star$ emilio.villa@kit.edu

*15th International Symposium on Radiative Corrections:
Applications of Quantum Field Theory to Phenomenology,
FSU, Tallahasse, FL, USA, 17-21 May 2021*

Abstract

We present the new release of pySecDec, a toolbox for the evaluation of dimensionally regulated parameter integrals. The main new features consist of an automated way to perform expansions, based on the geometric approach to the method of expansion by regions, and a new algorithm to efficiently evaluate linear combinations of integrals as needed for the calculation of scattering amplitudes. The other new features are also summarised briefly.

1 Introduction

The importance of precision calculations for the current and future High Energy Physics program is undeniable, however their complexity increases rapidly with the numer of loops and mass scales. This often prohibits analytic calculations of the underlying scattering amplitudes, such that (semi-)numerical approaches need to be pursued. Furthermore, in certain limits leading to a hierarchy of scales, approximate results can still be accurate enough for phenomenological purposes. This has been exploited for example in Higgs phenomenology, exploiting large-m_t or small-m_b expansions in their respective ranges of validity, see e.g. Ref. [1] for more details.

A systematic approach to the expansion of Feynman integrals is given by the so-called *expansion by regions*, pioneered in Refs. [2–4]. The method, originally developed in the momentum representation of Feynman integrals, was later reformulated in Feynman parameter space [5–8], where it allows for a geometric interpretation. An implementation of the method is available in the code ASY2.M [9], which is also part of the program FIESTA [10–14].

In these proceedings, we present a new version [15] of pySECDEC [16,17], which contains an implementation of expansion by regions based on its geometric formulation. Moreover, we present a new mechanism to efficiently evaluate amplitudes as sums of integrals, in a way which takes into account the relative importance of the individual integrals. Furthermore we briefly introduce the other features of the new release, for more details we refer to [15].

2 Geometric formulation of expansion by regions

2.1 Method

In this section, we discuss the expansion of parametric integrals over polynomials around small values of a so-called *smallness parameter* (e.g. m^2/p^2 in a large-momentum expansion). Feynman integrals can be brought into this form using the Lee-Pomeransky representation [18]:

$$I = \int_0^\infty \frac{d\mathbf{x}}{\mathbf{x}} \mathbf{x}^\nu t^{\nu_{N+1}} \left[\sum_{i=1}^m c_i \mathbf{x}^{\mathbf{p}_i} t^{p_{i,N+1}} \right]^{-\frac{D}{2}} , \tag{1}$$

where t is the smallness parameter, c_i the coefficients of the polynomial, D the space-time dimension and $\mathbf{x}^{\mathbf{a}} = \prod_{j=1}^N x_j^{a_j}$. The exponents can be organised into $(N+1)$-dimensional vectors $\mathbf{p}_i' \equiv (\mathbf{p}_i, p_{i,N+1})$, $\nu' \equiv (\nu, \nu_{N+1})$.

The goal of the method of expansion by regions is to identify all the possible scalings of the integration variables w.r.t. the smallness parameter in the different parts (called *regions*) of the integration domain. Once the integration domain has been divided into regions, the integrand is expanded in each region according to the given scaling and the corresponding expanded integrands are integrated over the *whole* integration domain. The sum of the resulting integrals adds up to the original integral since overlapping terms in general give rise to scaleless integrals that are zero in dimensional regularisation [19]. Moreover, the dimensional regulator ε, or additional analytical regulators, are used to make sure the integrals are well-defined even outside the region of convergence. Let us now sketch the geometric approach and illustrate it with a simple example.
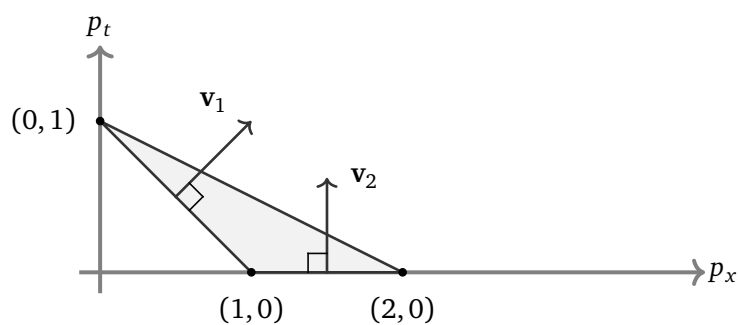
Figure 1: Newton polytope for $P(x,t) = t + x + x^2$, together with the directions of the region vectors (drawn as normal vectors to the facets in positive t-direction).

In order to find all the relevant regions, we can use the notion of the *Newton polytope* of a polynomial. The Newton polytope can be determined as the convex hull of the exponent vectors or, alternatively, as the intersection of half spaces

$$\Delta' = \bigcap_{f \in F} \left\{ \mathbf{m} \in \mathbb{R}^{N+1} \mid \langle \mathbf{m}, \mathbf{n}_f \rangle + a_f \geq 0 \right\} , \tag{2}$$

where F is the set of polytope facets with inward-pointing normal vectors $\mathbf{n}_f$, $\langle \mathbf{m}, \mathbf{n}_f \rangle$ is the scalar product of $\mathbf{m}$ and $\mathbf{n}_f$, and $a_f \in \mathbb{Z}$. An example of a Newton polytope for the simple polynomial $P(x,t) = t + x + x^2$ is shown in Fig. 1.

The subset of facets with normal vectors pointing in the positive t-exponent direction is $F^+ = \{ f \in F \mid (\mathbf{n}_f)_{N+1} > 0 \}$. It can be shown that the facets belonging to F^+ correspond to all the regions we need to consider [6]. These can be used as input for the change of variables:

$$t \to \prod_{f \in F^+} z_f^{(\mathbf{n}_f)_{N+1}} t \,, \qquad x_i \to \prod_{f \in F^+} z_f^{(\mathbf{n}_f)_i} x_i \,, \tag{3}$$

where for each facet f, $\mathbf{n}_f$ is the vector describing the rescaling applied to $(\mathbf{x}, t)$ and is called the *region vector*.

The transformations (3) lead to the following form of the integral (1):

$$I = \left(\prod_{f \in F^+} z_f^{\langle \mathbf{n}_f, \boldsymbol{\nu}' \rangle + \frac{D}{2} a_f} \right) \int_0^\infty \frac{d\mathbf{x}}{\mathbf{x}} \mathbf{x}^{\boldsymbol{\nu}} t^{\nu_{N+1}} \left[\sum_i c_i \mathbf{x}^{\mathbf{p}_i} t^{p_{i,N+1}} \prod_{f \in F^+} z_f^{\langle \mathbf{n}_f, \mathbf{p}_i' \rangle + a_f} \right]^{-\frac{D}{2}} . \tag{4}$$

The original integral can then be approximated by expanding in z_f in each region while setting the other $z_{f'}$ to one, and then setting z_f to one after the expansion in region f. The final result reads:

$$I = \sum_{f \in F^+} I_f \,, \tag{5}$$

where I_f is the expansion of (4) in z_f with all z set to one after the expansions. The I_f are integrated over the whole integration domain. The procedure outlined above is equivalent to expanding directly in the parameter t after appropriate rescaling of the Feynman parameters, a proof can be found in Ref. [15].

2.2 Usage

In this section we present a comparison between expansion by regions and standard pySECDEC. To this end, we evaluate as a sample integral the two-loop triangle given in Fig. 2 at order ε^0 in the ε-expansion. We consider the limit $s \ll m^2$ and we expand to leading order in the smallness parameter.
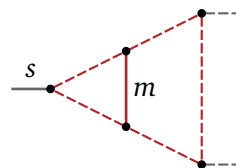
Figure 2: Two-loop triangle with one internal massive line

Fig. 3 shows the integration times for both, expansion by regions (solid line) and pySECDEC (dashed line) plotted against the ratio $r = m^2/s$. One can see how the integration times for pySECDEC blow up as the ratio of scales increases, while expansion by regions is numerically stable over the different orders of magnitude.

Fig. 4 shows the behaviour of $|R_n/R_a|$, the modulus of the ratio of the finite parts obtained numerically to the analytic result (from Ref. [20]), with increasing ratio $r = m^2/s$ of the kinematic scales, for both the numerical result obtained with expansion by regions and pySECDEC. The shaded areas are given by adding and subtracting the numerical uncertainty to the result.
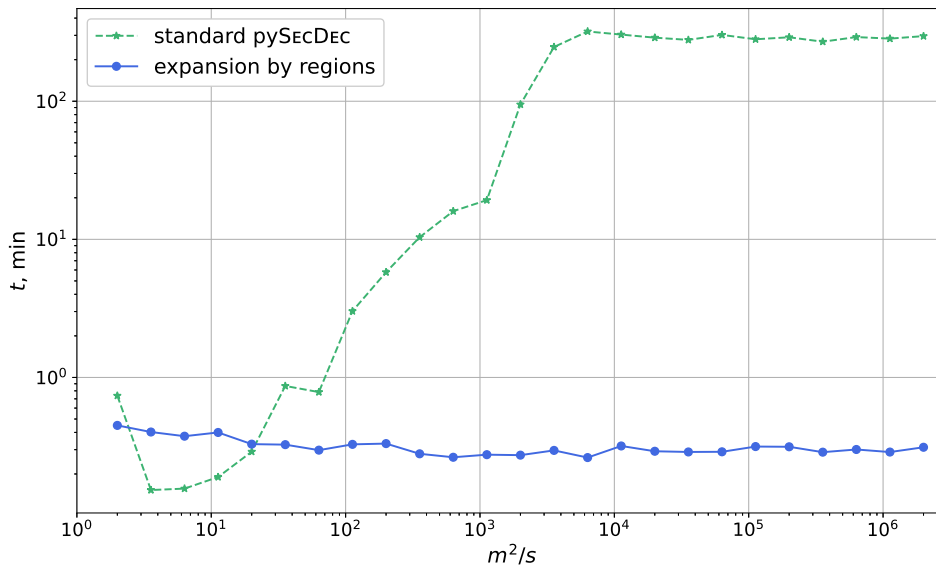
Figure 3: Scan over different orders of magnitude of $r = m^2/s$ for the two-loop triangle given in Fig 2. Integration times (in minutes) are plotted against r. The relative accuracy goal is 10^{-3}; the wall clock limit has been set to 5 hours.

The accuracy goal is fixed to 10^{-3}. Notice that the integration stops whenever the accuracy goal or the maximum number of evaluations are reached, meaning that the actual final uncertainty can be smaller than the required accuracy. Fig. 4 shows that the uncertainty due to the expansion is no longer dominant compared to the integration error if $m^2/s \gtrsim 40$, and that the results from pySECDEC and expansion by regions are compatible with each other in this case.

Combining the information of the two plots, it is clear that whenever the approximation due to the expansion is negligible compared to the numerical uncertainty of the result, using the expansion by regions option instead of standard pySECDEC is the better choice to evaluate the integral in the given kinematic limit.

As a final remark, it is worth to point out that for some integrals, the above considerations might hold only much deeper in the kinematic limit.

3 Amplitude evaluation and other new features

3.1 Amplitude evaluation

With the new release of pySECDEC, a new function called `sum_package` has been introduced. The function implements the algorithm described in [21] and generates a library that allows for the fast and efficient evaluation of amplitudes, i.e. linear combinations of integrals with coefficients depending on kinematic invariants and the space-time dimension. The efficient evaluation is achieved by an algorithm that takes into account how much the individual integrals contribute to the total, such that the accuracy goal for less dominant integrals can be lower than the accuracy demanded for the total amplitude. In this section we present a simple usage example, for more details we refer to Ref. [15].

The example discussed here is contained in `examples/yyyy1L` of the public code and
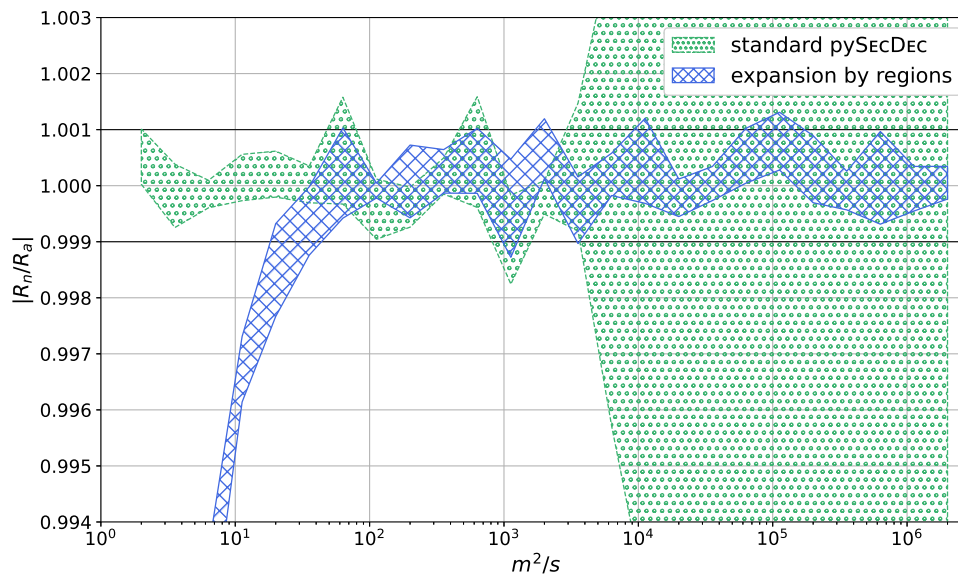
Figure 4: Scan over different orders of magnitude of $r = m^2/s$ for the two-loop triangle given in Fig 2. The modulus of the ratio between the numerical result and the analytic result, $|R_n/R_a|$, is plotted against r. The bands indicate the numerical uncertainties on the result.

calculates the one-loop 4-photon amplitude $\mathcal{M}^{++--}$. The amplitude can be expressed in terms of one-loop 4-point and 2-point integrals as

$$\mathcal{M}^{++--} = -8\left\{3(4-D)I_4^{D+4}(t,u) + \frac{t^2+u^2}{s}I_4^{D+2}(t,u) + \frac{t-u}{s}\left(I_2^D(u) - I_2^D(t)\right)\right\}, \quad (6)$$

where $D = 4 - 2\varepsilon$ and s, t, u are the usual Mandelstam invariants. Note that I_4^{D+4} is UV divergent, $I_4^{D+4} = 1/(6\varepsilon)$+finite, and therefore provides the rational part.

The example folder contains four files:

- `coefficients.py`: here the coefficient functions of the integrals are defined as a list. Each coefficient function has the arguments `numerators,denominators,parameters`, where the field `parameters` contains the names of the kinematic invariants. The polynomials in the numerator and denominator can also depend on the regulator ε.

- `integrals.py`: here the integrals are given as a list containing the "master" integrals $I_2^D(u)$, $I_2^D(t)$, $I_4^{D+2}(t,u)$, $I_4^{D+4}(t,u)$. Note that the ordering of the integrals and the corresponding coefficients should be the same.

- `generate_yyyy1L.py`: imports the integral definitions and the coefficients and runs `sum_package`.

- `integrate_yyyy1L.py`: performs the numerical integration, here the user can choose the integrator settings.

With the values of the kinematics invariants set to $t = -1.3, u = -0.8, s = -t - u$, the result reads

$$
\begin{aligned}
\mathcal{M}^{++--} = {} & (+0. \pm 2.1 \cdot 10^{-16})\,\varepsilon^{-1} + \\
& + (-28.431595834 \pm 5.4 \cdot 10^{-10} + (-1.3 \cdot 10^{-10} \pm 6.4 \cdot 10^{-10})\,i) + \\
& + \mathcal{O}(\varepsilon)\,,
\end{aligned}
\tag{7}
$$

in agreement with the analytic result. For more complicated amplitudes, the coefficients and the integral list will be much more complicated, however the structure is the same. Therefore pySECDEC can be used like an integral library for master integrals whose analytic expressions are unknown.

3.2 Other new features

In this section, we briefly introduce the other new features of pySECDEC. The following changes have been made compared to pySECDEC version 1.4.5:

- The contour deformation parameters λ_i are now reduced automatically if the original values lead to an invalid contour. This removes the "*sign check error*" that was one of the most frequent issues in previous versions of the code.

- The `WorkSpace` parameter of FORM [22,23] is now automatically increased if FORM fails due to insufficient `WorkSpace`. Users are no longer required to adjust the `form_work_space` parameter.

- pySECDEC can now be easily installed from the Python Package Index[1] using

  ```
  python3 -m pip install --user pySecDec
  ```

- The functions `series_to_ginac`, `series_to_sympy`, `series_to_maple` and `series_to_mathematica` have been added. They convert the output of pySECDEC to a syntax more suitable for use in combination with various computer algebra systems.

- The support for Python version 2.7 was dropped in favour of version 3.6 or newer. Python is now always invoked as `python3`.

The following new functions have been introduced:

- `make_regions` provides a package generator to perform expansion by regions of general parameter integrals.

- `loop_regions` is a wrapper around `make_regions` which simplifies the application of expansion by regions to loop integrals.

- `sum_package` has been discussed in the previous section.

These functions are documented in detail in the online documentation[2] distributed with the code.

---

[1] https://pypi.org/project/pySecDec/
[2] https://secdec.readthedocs.io

4 Conclusions

In these proceedings, we have presented a new implementation of the method of regions in the program pySecDec. We have shown that for integrals with a sufficiently large hierarchy among the kinematic invariants, the timings and accuracy of expansion by regions remain approximately constant as the ratio of scales increases, while the standard numerical evaluation by pySecDec faces convergence issues. This gives the user a powerful tool to get faster and more accurate results in certain kinematic limits.

In addition, we have introduced a mechanism in pySecDec that allows for the fast and efficient evaluation of linear combinations of integrals with coefficients that depend on kinematics and regulators, typically occurring in multi-loop scattering amplitudes. Each term in the sum is evaluated with a number of sampling points determined such that the global accuracy goal for the sum is reached most efficiently. Together with the other features of the new release, this makes pySecDec a tool suited for the evaluation of multi-loop amplitudes in a largely automated way.

Acknowledgements

I would like to thank the pySecDec members for the fruitful collaboration.

Funding information   This research was supported in part by the COST Action CA16201 ("Particleface") of the European Union and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant 396021762 - TRR 257.

References

[1] G. Heinrich, *Collider physics at the precision frontier*, Phys. Rep. **922**, 1 (2021), doi:10.1016/j.physrep.2021.03.006.

[2] V. Smirnov, *Renormalization and asymptotic expansions*, vol. 14, Birkhäuser (1991).

[3] M. Beneke and V. A. Smirnov, *Asymptotic expansion of Feynman integrals near threshold*, Nucl. Phys. B **522**, 321 (1998), doi:10.1016/S0550-3213(98)00138-2.

[4] V. A. Smirnov and E. R. Rakhmetov, *The regional strategy in the asymptotic expansion of two-loop vertex feynman diagrams*, Theor Math Phys **120**, 870 (1999), doi:10.1007/BF02557396.

[5] V. A. Smirnov, *Problems of the strategy of regions*, Phys. Lett. B **465**, 226 (1999), doi:10.1016/S0370-2693(99)01061-8.

[6] A. Pak and A. Smirnov, *Geometric approach to asymptotic expansion of Feynman integrals*, Eur. Phys. J. C **71**, 1626 (2011), doi:10.1140/epjc/s10052-011-1626-1.

[7] B. Ananthanarayan, A. Pal, S. Ramanan and R. Sarkar, *Unveiling regions in multi-scale Feynman integrals using singularities and power geometry*, Eur. Phys. J. C **79**, 57 (2019), doi:10.1140/epjc/s10052-019-6533-x.

[8] B. Ananthanarayan, A. B. Das and R. Sarkar, *Asymptotic analysis of Feynman diagrams and their maximal cuts*, Eur. Phys. J. C **80**, 1131 (2020), doi:10.1140/epjc/s10052-020-08609-0.

[9] B. Jantzen, A. V. Smirnov and V. A. Smirnov, *Expansion by regions: revealing potential and Glauber regions automatically*, Eur. Phys. J. C **72**, 2139 (2012), doi:10.1140/epjc/s10052-012-2139-2.

[10] A. V. Smirnov and M. N. Tentyukov, *Feynman Integral Evaluation by a Sector decomposiTion Approach (FIESTA)*, Comput. Phys. Commun. **180**, 735 (2009), doi:10.1016/j.cpc.2008.11.006.

[11] A. V. Smirnov, V. A. Smirnov and M. Tentyukov, *FIESTA 2: Parallelizeable multiloop numerical calculations*, Comput. Phys. Commun. **182**, 790 (2011), doi:10.1016/j.cpc.2010.11.025.

[12] A. V. Smirnov, *FIESTA 3: Cluster-parallelizable multiloop numerical calculations in physical regions*, Comput. Phys. Commun. **185**, 2090 (2014), doi:10.1016/j.cpc.2014.03.015.

[13] A. V. Smirnov, *FIESTA4: Optimized Feynman integral calculations with GPU support*, Comput. Phys. Commun. **204**, 189 (2016), doi:10.1016/j.cpc.2016.03.013.

[14] A. V. Smirnov, N. D. Shapurov and L. I. Vysotsky, *FIESTA5: numerical high-performance Feynman integral evaluation*, arXiv:2110.11660.

[15] G. Heinrich, S. Jahn, S. P. Jones, M. Kerner, F. Langer, V. Magerya, A. Põldaru, J. Schlenk and E. Villa, *Expansion by regions with pySecDec*, Comput. Phys. Commun. **273**, 108267 (2022), doi:10.1016/j.cpc.2021.108267.

[16] S. Borowka, G. Heinrich, S. Jahn, S. P. Jones, M. Kerner, J. Schlenk and T. Zirke, *pySecDec: A toolbox for the numerical evaluation of multi-scale integrals*, Comput. Phys. Commun. **222**, 313 (2018), doi:10.1016/j.cpc.2017.09.015.

[17] S. Borowka, G. Heinrich, S. Jahn, S. P. Jones, M. Kerner and J. Schlenk, *A GPU compatible quasi-Monte Carlo integrator interfaced to pySecDec*, Comput. Phys. Commun. **240**, 120 (2019), doi:10.1016/j.cpc.2019.02.015.

[18] R. N. Lee and A. A. Pomeransky, *Critical points and number of master integrals*, J. High Energ. Phys. **11**, 165 (2013), doi:10.1007/JHEP11(2013)165.

[19] B. Jantzen, *Foundation and generalization of the expansion by regions*, J. High Energ. Phys. **12**, 076 (2011), doi:10.1007/JHEP12(2011)076.

[20] J. Fleischer, A. V. Kotikov and O. L. Veretin, *Analytic two-loop results for self-energy- and vertex-type diagrams with one non-zero mass*, Nucl. Phys. B **547**, 343 (1999), doi:10.1016/S0550-3213(99)00078-4.

[21] M. Kerner, *NLO corrections to Higgs boson pair production in gluon fusion*, Proc. Sci. **260**, 023 (2016), doi:10.22323/1.260.0023.

[22] J. Kuipers, T. Ueda and J. A. M. Vermaseren, *Code optimization in FORM*, Comput. Phys. Commun. **189**, 1 (2015), doi:10.1016/j.cpc.2014.08.008.

[23] B. Ruijl, T. Ueda and J. Vermaseren, *FORM version 4.2*, arXiv:1707.06453.