Integral symmetries with pySecDec

Stephan Jahn[*]
Max-Planck-Institut für Physik, München, GERMANY
E-mail: sjahn@mpp.mpg.de

We describe a new feature of pySecDec, a toolbox for parametric and loop integrals in the context of dimensional regularization. Due to its modular structure, most algorithms implemented in pySecDec's algebraic preprocessing can be called in user-defined codes. Therefore, pySecDec can be applied in contexts beyond numerical evaluation of parameter integrals. We show how the symmetry finder in pySecDec can be used to identify matroid symmetries of anisomorphic graphs. We further discuss its use within Loopedia, a database for loop integrals.

Corfu Summer Institute 2017 "School and Workshops on Elementary Particle Physics and Gravity"
2-28 September 2017
Corfu, Greece

[*]Speaker.

1. Introduction

Multi-loop calculations tend to be computationally intensive. It is therefore essential to simplify all occurring algebraic expressions as much as possible. Often symmetries within the problem can be largely exploited to reduce the complexity of intermediate steps.

Within sector decomposition [1, 2], a single parameter integral is transformed into many integrals with similar integrand structure. However, these resulting integrals can often be identified to be equal at integrand level by rearranging terms and renaming variables. These sector symmetries are identified by a symmetry finder in pySecDec [3], the successor of SecDec [4, 5, 6].

Another complication in multi-loop calculations is the evaluation of the master integrals. Loopedia [7] has been introduced as a database of known integrals to avoid unnecessary re-computations. However, it is currently graph based, which makes it miss so called matroid [8] symmetries; i.e. symmetries between anisomorphic graphs.

It turns out that finding sector and matroid symmetries is essentially the same problem. In this article, we describe an extension of the polynomial canonicalization algorithm proposed in [9]. We then demonstrate how the canonical representation of a Feynman parameterized loop integral is used to identify matroid symmetries.

This article is organized as follows: We summarize the extended polynomial canonicalization algorithm in section 2. Its application to identify matroid symmetries is described in section 3. Prospects for an extension of Loopedia are discussed in section 4. We close with concluding remarks in section 5.

2. Canonical representation of polynomials

A key concept to finding matroid and sector symmetries is to identify two polynomials with one another when permutations of subsets of variables are allowed. For example,

$$\int_0^1 \mathrm{d}x\,\mathrm{d}y\ (x+2y) = \int_0^1 \mathrm{d}x\,\mathrm{d}y\ (y+2x);$$

i.e. the integration variables can be relabelled. Algorithms to find symmetries when all variables are equivalent have been introduced in [9, 10]. Their implementation in pySecDec is discussed in [11]. To find matroid symmetries, arbitrary relabelling of additional parameters (e.g. masses) must be taken into account as well, for example,

$$\int_0^1 \mathrm{d}x\,\mathrm{d}y\ (sx+m_1y+2m_2) \quad \overset{m_1\leftrightarrow m_2}{=\!=\!=} \quad \int_0^1 \mathrm{d}x\,\mathrm{d}y\ (sy+m_2x+2m_1),$$

if m_1 and m_2 can be interchanged. In the following, we describe a generalization of the algorithm in [9] that allows for multiple sets of interchangeable (“equivalent") variables.

As a first step, polynomials are expressed as a matrix of exponents and coefficients. Each row corresponds to one term of the polynomial. The first column consists of the coefficients of the terms while the remaining columns store the powers of the occurring variables. An example is shown in Figure 1.

$$\begin{aligned} &1\cdot m_1^0\cdot m_2^0\cdot s^1\cdot x^1\cdot y^0 \\ +\,&1\cdot m_1^1\cdot m_2^0\cdot s^0\cdot x^0\cdot y^1 \\ +\,&2\cdot m_1^0\cdot m_2^1\cdot s^0\cdot x^0\cdot y^0 \end{aligned} \quad \equiv \quad \begin{array}{cccccc} \text{coefficient} & m_1 & m_2 & s & x & y \\ \hline 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 1 & 0 & 0 & 0 \end{array}$$

Figure 1: Representation of the polynomial $sx+m_1y+2m_2$ as an integer matrix.

If the coefficient is not an integer, it is sufficient to replace each individual coefficient with a unique index during the canonicalization. Multiple polynomials can be canonicalized simultaneously by multiplication of a label to the coefficient of each polynomial or by adding a column with an index identifying the polynomial. Writing the polynomial as an integer matrix facilitates to tackle the problem without a computer algebra system. Finding equality of two polynomials under relabelling corresponds to finding equality of their matrices while allowing for permutations of the rows and columns. Rather than generating and trying all permutations, it is much more efficient to bring the polynomials to compare into a canonical form.

To canonicalize a polynomial, we suggest a generalization of the algorithm in [9]. The key idea is to build a canonical representation column by column. In the following detailed description, we assume that the first column of the matrix corresponds to the coefficients, the next n_1 columns to the powers of the first set of equivalent parameters, the next n_2 columns to the powers of the second set of equivalent parameters and so on.

(i) Write the polynomial as an integer matrix, see above.
(ii) Make n_1 copies of the matrix. In the i$^{\text{th}}$ copy, swap the column of the i$^{\text{th}}$ parameter parameter with the column of the first parameter.
(iii) In all copies, sort rows lexicographically by the columns corresponding to the coefficient and the first parameter.
(iv) Discard all but those copies with the lexicographically smallest columns corresponding to the first parameter.
(v) For all remaining copies of the matrix, make n_1-1 new copies and swap the column corresponding to the i$^{\text{th}}$ parameter with the column corresponding to the second parameter in each i$^{\text{th}}$ new copy.
(vi) In all copies, sort rows lexicographically by the columns corresponding to the coefficient and the first two parameters.
(vii) Discard all but those copies with the lexicographically smallest columns corresponding to the second parameter. Note that some of the remaining matrices are likely to appear multiple times. It is strongly advisable to check for and delete repetitions in this step as well.
(viii) Repeat the procedure with the remaining parameters that are equivalent to the first parameter.
(ix) Repeat all previous steps for all remaining matrices for every group of equivalent variables. When considering the j$^{\text{th}}$ group of equivalent parameters, always include the coefficient and all $n_{i<j}$ variables in the sorting steps; i.e. a after a column has been included in the sorting, it should be included in all later sortings.

(x) Pick the lexicographically smallest matrix as the canonical representation.

Our generalization to the algorithm in [9] is the second to last step, where we repeat the original algorithm for all sets of equivalent parameters.

3. Matroid symmetries

Matroid symmetries are manifest in Feynman parameterization by comparing the Symanzik polynomials $\mathscr{U}$ and $\mathscr{F}$ under relabellings of the external momenta, the masses, and the Feynman parameters. Consequently, to systematically identify matroid symmetries within a given database (e.g. `Loopedia`), we first Feynman parameterize all the loop integrals as

$$G = (-1)^{N_\nu} \frac{\Gamma(N_\nu - LD/2)}{\prod_{j=1}^{N} \Gamma(\nu_j)} \int_0^\infty \prod_{j=1}^{N} dx_j\, x_j^{\nu_j - 1}\, \delta(1 - \sum_{l=1}^{N} x_l)\, \frac{\mathscr{U}^{N_\nu - (L+1)D/2}}{\mathscr{F}^{N_\nu - LD/2}} \,, \tag{3.1}$$

where N is the number of propagators, ν_j the power of the j^{th} propagator, $N_\nu = \sum_{j=1}^{N} \nu_j$, L the number of loops and D the dimension to compute the integral in. We restrict the following discussion to integrals without inverse or dotted propagators; i.e. to integrals with all $\nu_j = 1$, which also implies $N_\nu = N$.

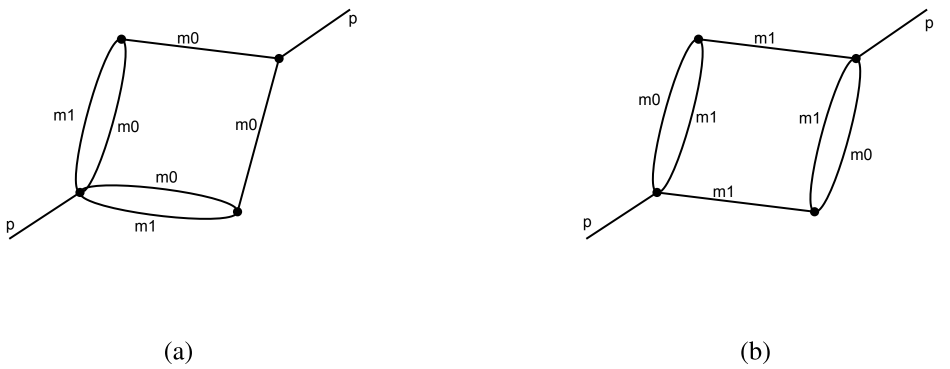

Figure 2: Example of anisomorphic graphs corresponding to the same integral.

Our goal is to identify anisomorphic graphs that have an equivalent Feynman parameter representation. Consider for example the two graphs shown in Figure 2. Computing their corresponding

Symanzik polynomials with `pySecDec`, we get

$$
\begin{aligned}
\mathscr{U}_a =& + x_0x_1x_3 + x_0x_1x_4 + x_0x_2x_3 + x_0x_2x_4 + x_0x_3x_4 + x_0x_3x_5 + x_0x_4x_5 + x_1x_3x_5 \\
& + x_1x_4x_5 + x_2x_3x_5 + x_2x_4x_5 + x_3x_4x_5 \\
\mathscr{F}_a =& + m_1^2x_0x_1x_3x_4 + m_1^2x_0x_1x_3x_5 + m_1^2x_0x_1x_4^2 + m_1^2x_0x_1x_4x_5 + m_1^2x_0x_2x_3x_4 + m_1^2x_0x_2x_3x_5 \\
& + m_1^2x_0x_2x_4^2 + m_1^2x_0x_2x_4x_5 + m_1^2x_0x_3x_4^2 + 2m_1^2x_0x_3x_4x_5 + m_1^2x_0x_3x_5^2 + m_1^2x_0x_4^2x_5 \\
& + m_1^2x_0x_4x_5^2 + m_1^2x_1x_3x_4x_5 + m_1^2x_1x_3x_5^2 + m_1^2x_1x_4^2x_5 + m_1^2x_1x_4x_5^2 + m_1^2x_2x_3x_4x_5 \\
& + m_1^2x_2x_3x_5^2 + m_1^2x_2x_4^2x_5 + m_1^2x_2x_4x_5^2 + m_1^2x_3x_4^2x_5 + m_1^2x_3x_4x_5^2 + m_0^2x_0^2x_1x_3 \\
& + m_0^2x_0^2x_1x_4 + m_0^2x_0^2x_2x_3 + m_0^2x_0^2x_2x_4 + m_0^2x_0^2x_3x_4 + m_0^2x_0^2x_3x_5 + m_0^2x_0^2x_4x_5 \\
& + m_0^2x_0x_1^2x_3 + m_0^2x_0x_1^2x_4 + 2m_0^2x_0x_1x_2x_3 + 2m_0^2x_0x_1x_2x_4 + m_0^2x_0x_1x_3^2 + 2m_0^2x_0x_1x_3x_4 \\
& + 2m_0^2x_0x_1x_3x_5 + 2m_0^2x_0x_1x_4x_5 + m_0^2x_0x_2^2x_3 + m_0^2x_0x_2^2x_4 + m_0^2x_0x_2x_3^2 + 2m_0^2x_0x_2x_3x_4 \\
& + 2m_0^2x_0x_2x_3x_5 + 2m_0^2x_0x_2x_4x_5 + m_0^2x_0x_3^2x_4 + m_0^2x_0x_3^2x_5 + 2m_0^2x_0x_3x_4x_5 + m_0^2x_1^2x_3x_5 \\
& + m_0^2x_1^2x_4x_5 + 2m_0^2x_1x_2x_3x_5 + 2m_0^2x_1x_2x_4x_5 + m_0^2x_1x_3^2x_5 + 2m_0^2x_1x_3x_4x_5 + m_0^2x_2^2x_3x_5 \\
& + m_0^2x_2^2x_4x_5 + m_0^2x_2x_3^2x_5 + 2m_0^2x_2x_3x_4x_5 + m_0^2x_3^2x_4x_5 - p^2x_0x_1x_2x_3 - p^2x_0x_1x_2x_4 \\
& - p^2x_0x_1x_3x_4 - p^2x_0x_2x_3x_5 - p^2x_0x_2x_4x_5 - p^2x_0x_3x_4x_5 - p^2x_1x_2x_3x_5 - p^2x_1x_2x_4x_5 \\
& - p^2x_1x_3x_4x_5 \\
\mathscr{U}_b =& \mathscr{U}_a(x_0 \leftrightarrow x_1) \\
\mathscr{F}_b =& \mathscr{F}_a(x_0 \leftrightarrow x_1, m_0 \leftrightarrow m_1);
\end{aligned}
\tag{3.2}
$$

i.e. their Symanzik polynomials $\mathscr{U}$ and $\mathscr{F}$ are equal with the permutations $x_0 \leftrightarrow x_1$ and $m_1 \leftrightarrow m_2$. The labelling of the Feynman parameters x_0 to x_5 depends on the ordering of the internal and external lines in the input to `pySecDec` and is therefore arbitrary. The masses m_0 and m_1 are arbitrary and therefore interchangeable. As a consequence, their Feynman representations are equivalent since the two graphs have the same numbers of loops and propagators.

To identify permutations of the external momentum labels, we keep all scalar products as they are except for squared momenta. Squares of external momenta may be set equal to an internal mass and are therefore considered equivalent to the masses.

In addition to relabellings, any one of the external momenta can be eliminated by momentum conservation. Eliminating a different external momentum leads to in general inequivalent representations of the integral that are not necessarily related by relabelling. To identify all symmetries, in principle all canonical representations with either external momentum eliminated must be considered. However, inserting momentum conservation is the only operation in the canonicalization process that can change the total number of appearing variables and the number of terms. We therefore only consider the representations with the lowest number of variables and, out of those, the representations with the lowest number of terms. Considering these shortest representations only should still identify all matroid symmetries, as long as every vertex connects to at most one external leg.

4. Matroid symmetries in Loopedia

`Loopedia` is a database of Feynman integrals that allows to search the database for literature on a given graph rather than conventional search keys like author, title, etc. However, Loopedia currently only shows results corresponding to graphs that are isomorphic to the input graph. Articles considering the corresponding integral are missed if they are listed in the database under a different graph that is related by a matroid. We now comment on how to obtain a canonical form of the integrals present in the Loopedia database. Given a canonical representation for every graph, Loopedia could point the user to other graphs that are related by a matroid symmetry e.g. in the single graph view or in the graph browser.

Loopedia considers three different types of masses: Nonzero masses, special masses and general masses. Special masses denote a particular mass for which the integral has been computed. A general mass denotes a mass that can be set an arbitrary value including zero, while a nonzero mass can take any nonzero value. When searching for symmetries, any relabelling within these three groups of masses must be considered. The squared momenta of the external legs are parameterized as masses out of the same sets as the internal masses; i.e. an internal mass squared and and external momentum squared are interchangeable if they belong to the same class of masses.

After inserting momentum conservation (in all possible ways as discussed in the previous section), the remaining scalar products form the kinematic invariants. These invariants and the Feynman parameters form the remaining sets of equivalent variables where relabellings should be considered.

5. Conclusion

We have extended an algorithm [9] to canonicalize polynomials with each other under relabelling of groups of variables. Our extension accounts for multiple sets of parameters to be considered equivalent, e.g. masses, Feynman parameters, etc. The extended canonicalization algorithm is implemented in pySecDec.

We have stressed the usability of pySecDec beyond numerical evaluations of loop integrals. As an example, we have discussed how the symmetry finder can be applied to identify matroid symmetries between anisomorphic graphs. We further have discussed prospects of extending Loopedia to identify such matroid symmetries.

Acknowledgments

I thank my fellow SecDec and Loopedia developer teams, Christian Bogner, Sophia Borowka, Thomas Hahn, Gudrun Heinrich, Stephen Jones Matthias Kerner, Andreas von Manteuffel, Martin Michel, Erik Panzer, Johannes Schlenk, and Tom Zirke, for helpful discussions and the successful collaboration. I further thank Ben Ruijl for completing our implementation of Pak's original algorithm.

References

[1] T. Binoth and G. Heinrich, *An automatized algorithm to compute infrared divergent multi-loop integrals*, *Nucl. Phys.* **B585** (2000) 741–759, [hep-ph/0004013].

[2] G. Heinrich, *Sector Decomposition*, *Int. J. Mod. Phys.* **A23** (2008) 1457–1486, [0803.4177].

[3] S. Borowka, G. Heinrich, S. Jahn, S. P. Jones, M. Kerner, J. Schlenk et al., *pySecDec: a toolbox for the numerical evaluation of multi-scale integrals*, Comput. Phys. Commun. **222** (2018) 313–326, [1703.09692].

[4] J. Carter and G. Heinrich, *SecDec: A general program for sector decomposition*, Comput.Phys.Commun. **182** (2011) 1566–1581, [1011.5493].

[5] S. Borowka, J. Carter and G. Heinrich, *Numerical Evaluation of Multi-Loop Integrals for Arbitrary Kinematics with SecDec 2.0*, Comput.Phys.Commun. **184** (2013) 396–408, [1204.4152].

[6] S. Borowka, G. Heinrich, S. P. Jones, M. Kerner, J. Schlenk and T. Zirke, *SecDec-3.0: numerical evaluation of multi-scale integrals beyond one loop*, Comput. Phys. Commun. **196** (2015) 470–491, [1502.06595].

[7] C. Bogner, S. Borowka, T. Hahn, G. Heinrich, S. P. Jones, M. Kerner et al., *Loopedia, a Database for Loop Integrals*, Comput. Phys. Commun. **225** (2018) 1–9, [1709.01266].

[8] C. Bogner and S. Weinzierl, *Feynman graph polynomials*, Int. J. Mod. Phys. **A25** (2010) 2585–2618, [1002.3458].

[9] A. Pak, *The Toolbox of modern multi-loop calculations: novel analytic and semi-analytic techniques*, J. Phys. Conf. Ser. **368** (2012) 012049, [1111.0868].

[10] B. D. McKay and A. Piperno, *Practical graph isomorphism, ii*, Journal of Symbolic Computation **60** (2014) 94 – 112.

[11] S. P. Jones and B. Ruijl, *To appear in the proceedings of the 18th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2017), Journal of Physics: Conference Series* (2017) .