Implementation of the $\mathcal{O}(\alpha_t^2)$ MSSM Higgs-mass corrections in `FeynHiggs`

Thomas Hahn[1][*] and Sebastian Paßehr[2][†]

[1]*Max-Planck-Institut für Physik
(Werner-Heisenberg-Institut),
Föhringer Ring 6, D–80805 München, Germany*

[2]*Deutsches Elektronen-Synchrotron DESY,
Notkestraße 85, D–22607 Hamburg, Germany*

We describe the implementation of the two-loop Higgs-mass corrections of $\mathcal{O}(\alpha_t^2)$ in the complex MSSM in `FeynHiggs`. The program for the calculation is comprised of several scripts which flexibly use `FeynArts` and `FormCalc` together with other packages. It is included in `FeynHiggs` and documented here in some detail so that it can be re-used as a template for similar calculations.

[*]email: hahn@mpp.mpg.de
[†]email: sebastian.passehr@desy.de

1 Introduction

The Higgs-like boson discovered at LHC [1, 2] initiated great efforts to pin it down as a particle responsible for electroweak symmetry breaking. Although the measured properties of this new boson are consistent with the expectations for the Standard Model (SM) Higgs boson [3], a variety of beyond-SM interpretations is still possible. One such possibility is the interpretation as a light state within a richer spectrum of scalar particles in the theoretically well motivated Minimal Supersymmetric Standard Model (MSSM). Its Higgs sector consists of two complex scalar doublets leading to five physical Higgs bosons and three (pseudo-) Goldstone bosons. The physical mass eigenstates at the tree level are the neutral CP-even h, H, the CP-odd A, and the charged $H^\pm$ bosons. Their masses can be parameterized by the charged (or CP-odd) Higgs-boson mass and the ratio of the two vacuum expectation values, $\tan\beta = v_2/v_1$. CP-violation in the Higgs sector is induced by complex parameters in other sectors of the MSSM via loop corrections, leading to mixing between h, H, and A in the mass eigenstates [4–7].

Since the masses of the neutral Higgs bosons are strongly affected by loop contributions, a lot of work has been invested into higher-order calculations of the mass spectrum from the SUSY parameters, in the case of the real MSSM [8–23] as well as for the MSSM with complex parameters (cMSSM) [5–7, 24–29]. At one-loop order, the dominant contributions arise from the Yukawa sector with the large top-Yukawa coupling h_t, or $\alpha_t = h_t^2/(4\pi)$. The class of leading two-loop Yukawa-type corrections of $\mathcal{O}(\alpha_t^2)$ has recently been calculated for the case of complex parameters [28, 29]. Together with the full one-loop result [26] and the leading $\mathcal{O}(\alpha_t\alpha_s)$ terms [27] it is now available in the public program `FeynHiggs` [10, 20, 26, 30–32].

The $\mathcal{O}(\alpha_t^2)$ contributions in the cMSSM have been evaluated in the Feynman-diagrammatic approach, extending the on-shell renormalization scheme of Ref. [26] to the two-loop level. This ensures that the obtained analytical results for the renormalized two-loop self-energies can consistently be incorporated in `FeynHiggs`. For implementation in `FeynHiggs` the whole calculation of the $\mathcal{O}(\alpha_t^2)$ contributions was reworked and broken up into several working steps. Together they provide a semi-automated framework for a two-loop calculation in a model with non-trivial renormalization. Although the code is specific to the evaluation of the mentioned corrections, it can be utilized as a template for similar calculations. The aim of this paper is to explain in detail how the code works and what is done at each step, and where process-specific adjustments are necessary.

A note on the software-engineering aspect is in order here, too: as many software packages (both in physics and elsewhere) also `FeynArts` and `FormCalc` have been designed to do a 'complete' job. That is, all the steps from the generation of Feynman diagrams to the numerical computation of a cross-section are executed from a single control program (e.g. a single Mathematica session) – that at least is how the demo programs insinuate usage. There is nothing inherently wrong with this 'monolithic' approach, it is maybe just not obvious how to extend it beyond the limitations of the package(s) used.

The suite of shell scripts and Mathematica packages we present here constitute a showcase of how to use `FeynArts` and `FormCalc` more flexibly, together with other packages. The scripts provide the 'outer hull' of a computation, in particular its compartmentalization, with coordination through a makefile. The packages provide recurring tasks 'library-like' in easily identifiable components.

The present code can be understood in a wider sense as a prototypical implementation of a new way of making flexible use of existing programs, even for very specific tasks, while retaining core functionality of the established packages. It is in many aspects still a proof-of-concept realization since, for example, not all `FormCalc` routines can generically handle diagrams with more than one loop yet. Likewise, a complete setup would include (standardized) converters to and from the more important available software packages, e.g. for multiloop tensor reduction. In that sense our code is not so much a demo program (to be blindly used) than a template (to be intelligently adapted).

The paper is organized as follows: Sect. 2 provides the theoretical foundation of the computation of the Higgs-mass corrections in the cMSSM. The approximations used for the evaluation of the $\mathcal{O}(\alpha_t^2)$ contributions to the Higgs masses are explained in Sect. 3. A detailed description of the implementation of the $\mathcal{O}(\alpha_t^2)$ corrections in `FeynHiggs` is given in Sect. 4.

2 Masses of the Higgs bosons in the Complex MSSM

Since the calculation of the $\mathcal{O}(\alpha_t^2)$ corrections has already been presented in great detail elsewhere [28, 29] we shall recap only the parts necessary for the implementation in `FeynHiggs` here.

In the basis of the tree-level Higgs states the loop-corrected mass matrices are given by [26]

$$\mathbf{M}_0^2 = \begin{pmatrix} m_h^2 - \hat{\Sigma}_{hh} & -\hat{\Sigma}_{hH} & -\hat{\Sigma}_{hA} & -\hat{\Sigma}_{hG} \\ -\hat{\Sigma}_{hH} & m_H^2 - \hat{\Sigma}_{HH} & -\hat{\Sigma}_{HA} & -\hat{\Sigma}_{HG} \\ -\hat{\Sigma}_{hA} & -\hat{\Sigma}_{HA} & m_A^2 - \hat{\Sigma}_{AA} & -\hat{\Sigma}_{AG} \\ -\hat{\Sigma}_{hG} & -\hat{\Sigma}_{HG} & -\hat{\Sigma}_{AG} & m_G^2 - \hat{\Sigma}_{GG} \end{pmatrix}, \tag{2.1}$$

$$\mathbf{M}_\pm^2 = \begin{pmatrix} m_{H^\pm}^2 - \hat{\Sigma}_{H^-H^+} & -\hat{\Sigma}_{H^-G^+} \\ -\hat{\Sigma}_{G^-H^+} & m_{G^\pm}^2 - \hat{\Sigma}_{G^-G^+} \end{pmatrix}. \tag{2.2}$$

where the $\hat{\Sigma}$ denote the renormalized self-energies. In the following we shall neglect the Goldstone matrix elements (greyed out above) since their numerical effect is tiny. Mixing with Goldstone bosons is taken into account inside the loop diagrams and for a consistent renormalization, of course. The physical masses are obtained as the real parts of the poles of the propagator matrix

$$\Delta(p^2) = -\mathrm{i}\left[p^2\mathbb{1} - \mathbf{M}^2\right]^{-1}. \tag{2.3}$$

This is tantamount to finding the zeroes of the determinant of $p^2\mathbb{1} - \mathbf{M}^2$, which is core functionality of `FeynHiggs`.

For the cMSSM, the following contributions to Eq. (2.1) are available in `FeynHiggs`: the one-loop self-energies with full p^2 dependence [26], the leading momentum-independent two-loop $\mathcal{O}(\alpha_s\alpha_t)$ self-energies [27], and now also the $\mathcal{O}(\alpha_t^2)$ [29] self-energies. The subleading two-loop $\mathcal{O}(\alpha_s\alpha_b)$, $\mathcal{O}(\alpha_t\alpha_b)$ parts are implemented only in the rMSSM yet, and are interpolated in the phases when complex parameters are chosen [26].

For the latter corrections the following components have to be computed:

- The unrenormalized genuine two-loop self-energies $\Sigma_{hh}^{(2)}, \Sigma_{hH}^{(2)}, \Sigma_{hA}^{(2)}, \Sigma_{HH}^{(2)}, \Sigma_{HA}^{(2)}, \Sigma_{AA}^{(2)}, \Sigma_{H^+H^-}^{(2)}$ at $p^2 = 0$ in $\mathcal{O}(\alpha_t^2)$ approximation.

- The one-loop diagrams with insertions of one-loop counterterms. For the $\mathcal{O}(\alpha_t^2)$ corrections the couplings and masses of the colored sector and chargino–neutralino sector are affected by this subrenormalization.

- The two-loop counterterms for these self-energies. The counterterms, the imposed renormalization conditions, and all renormalization constants are given in Ref. [29].

- The two-loop tadpoles $T_h^{(2)}, T_H^{(2)}, T_A^{(2)}$ in $\mathcal{O}(\alpha_t^2)$ approximation, which enter the two-loop counterterms.

3 Approximations in the $\mathcal{O}(\alpha_t^2)$ contributions

The dominant terms of the $\mathcal{O}(\alpha_t^2)$ contributions are enhanced by an additional factor m_t^2. The following approximations are applied to the renormalized two-loop self-energies in Eq. (2.1) to yield only this leading part.

3.1 Gaugeless limit

The gauge couplings g_1 and g_2 are set to zero, and also the strong coupling g_s is discarded. Consequently, also the gauge-boson masses M_W and M_Z vanish, while the weak mixing angle θ_w retains its original value. These choices cannot naively be substituted in the original MSSM model file of `FeynArts` [33] where the couplings are not yet expressed in terms of the top-Yukawa coupling $h_t = e\,m_t/(\sqrt{2}\,s_\beta\,s_w\,M_W)$. Furthermore, the non-zero ratios $\delta M_W/M_W$ and $\delta M_Z/M_Z$ appear in the renormalization of h_t, while the gauge-boson-mass renormalization constants δM_W and δM_Z themselves vanish. The technical implementation of these issues is described in Sect. 4.

3.2 Vanishing external momentum

The external momentum is set to zero, i.e. the integrals that appear in the two-loop self-energies and the renormalization constants are calculated at $p^2 = 0$. As an immediate consequence all appearing two-loop integrals are independent of p^2, leading to vacuum diagrams that are known analytically.

3.3 Bottom mass equal to zero

To isolate the leading $\mathcal{O}(\alpha_t^2)$ contributions only, the bottom-quark mass m_b has to be set to zero.

3.4 Consequences of the approximations

The approximations mentioned above modify the MSSM parameters that enter the calculation as follows:

- The Higgs- and Goldstone-boson masses at the tree level become $m_h^2 = m_G^2 = m_{G^\pm}^2 = 0$ and $m_H^2 = m_A^2 = m_{H^\pm}^2$.

- The Higgs-sector mixing angles α and β are related through $\alpha = \beta - \pi/2$.

- The stop masses are derived as the eigenvalues of

$$\begin{pmatrix} m_{\tilde{t}_L}^2 + m_t^2 & m_t X_t^* \\ m_t X_t & m_{\tilde{t}_R}^2 + m_t^2 \end{pmatrix}, \tag{3.1}$$

  with $X_t = A_t - \mu^*/t_\beta$, where the soft-SUSY-breaking parameters $m_{\tilde{t}_{L,R}}^2$ and A_t enter.

- Only one sbottom remains in the Feynman diagrams with $m_{\tilde{b}}^2 = m_{\tilde{t}_L}^2$. The sbottom mixing matrix is set to the unity matrix.

- Of the neutralinos and charginos only the original Higgsinos $\tilde{\chi}_{3,4}^0$, $\tilde{\chi}_2^\pm$ occur in the Feynman diagrams and their degenerate masses are equal to $|\mu|$. The mixing matrices $\mathbf{N}$ for the neutralinos and $\mathbf{U}$, $\mathbf{V}$ for the charginos simplify to

$$\mathbf{N} = \begin{pmatrix} \begin{pmatrix} \mathrm{e}^{\frac{\mathrm{i}}{2}\phi_{M_1}} & 0 \\ 0 & \mathrm{e}^{\frac{\mathrm{i}}{2}\phi_{M_2}} \end{pmatrix} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{2}}\mathrm{e}^{\frac{\mathrm{i}}{2}\phi_\mu} \begin{pmatrix} 1 & -1 \\ \mathrm{i} & \mathrm{i} \end{pmatrix} \end{pmatrix}, \qquad \mathbf{U} = \begin{pmatrix} \mathrm{e}^{\mathrm{i}\phi_{M_2}} & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\phi_\mu} \end{pmatrix}, \qquad \mathbf{V} = \mathbb{1}. \tag{3.2}$$

- Many couplings are simplified. Technically this is taken care of by a modified version of the `FeynArts` model file for the complex MSSM, generated as described in Sect. 4. Explicit results can also be found in Ref. [29].

4 Implementation in `FeynHiggs`

Our code for the calculation of the renormalized two-loop self-energies at $\mathcal{O}(\alpha_t^2)$ is based on several Mathematica Notebooks used in the former calculation [29]. From the aspect of software development these Notebooks had various shortcomings: duplicate code (e.g. gaugeless limit implemented multiply), parallel instructions poured all over, even the requirement to run with a particular Mathematica version – signs that the use of different packages together with various special requirements was nontrivial and that the reorganization was helpful. The new code is included in the `FeynHiggs` tarball and, while still specific to the corrections mentioned, is very adaptable and may serve as a template for similar calculations, which is the reason we describe it here in some detail.

The calculation has been divided into seven working steps implemented by the scripts described in the following sections. Additionally, several packages that are required for specific purposes are accessed during each step. A makefile coordinates the entire calculation. The total running time is about 15–20 minutes and the final output is a highly optimized and very compact Fortran code with a file size of about 350 kBytes. The result can directly be moved into the `FeynHiggs` source tree. The following diagram gives a flow chart of scripts and the external software packages used:
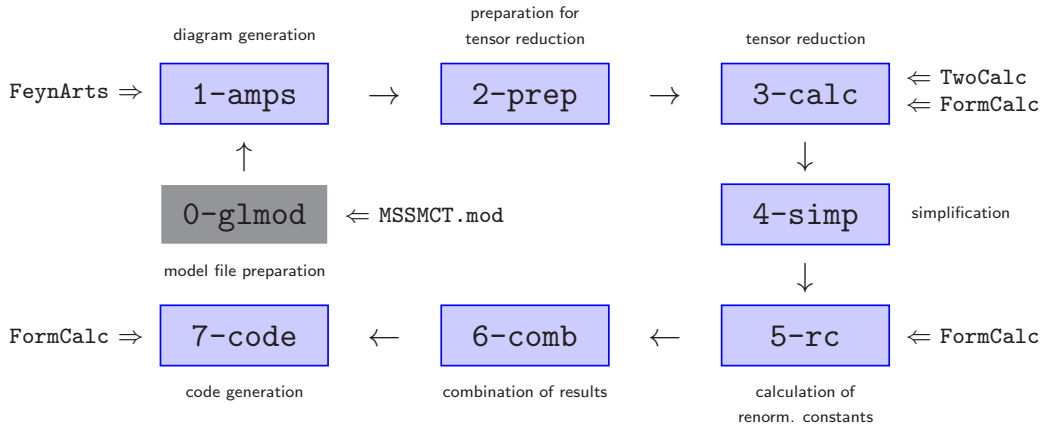
The advantages of this setup are twofold: First, the compartmentalization takes care of 'concurrent' software packages (which cannot be loaded into the same Mathematica session, typically because of symbol conflicts). Likewise, if an extension of this code required a different package for a particular step (say, tensor reduction), only this one step would need to be modified, and the additional package could be an arbitrary one, not just a Mathematica package, since it is invoked from a shell script. Second, the makefile sees to it that only the necessary parts of the calculation are redone in case of a change, which was a particular time-saver during the debugging phase.

Some details of the software setup:

- The code described below is included in `FeynHiggs` 2.11.0 or later, in the `gen/tlsp` tree.

- The shell scripts (`/bin/sh`) in the 'scripts' subdirectory perform the actual calculation. They are executed directly from the command-line (i.e. not loaded within Mathematica), usually by the makefile. Internally they invoke the Mathematica Kernel through the shell's 'here documents' as described in Ref. [34].

- The Mathematica programs in the 'packages' subdirectory provide functions for specific purposes and are read by the scripts.

- As in the FeynArts model file for the MSSM [33] we choose the particle labels

$$h = \texttt{h0}, \quad H = \texttt{HH}, \quad A = \texttt{A0}, \quad H^+ = \texttt{Hp}, \quad H^- = \texttt{Hm}.$$

- Each script is run from the command line with up to two arguments, e.g.

    scripts/1-amps ARG$_1$ ARG$_2$

    where ARG$_1$ = h0h0, h0HH, h0A0, HHHH, HHA0, A0A0, HmHp (self-energies),
                    h0, HH, A0 (tadpoles),
        ARG$_2$ = 0 for virtual two-loop diagrams,
              1 for one-loop diagrams with one-loop counterterms.

    That is, each self-energy/tadpole is computed and stored individually. Arguments are checked upon entry and meaningless arguments are omitted (e.g. no ARG$_2$ after combination of virtual and counterterm diagrams).

- Symbolic expressions are generally output to the 'm' directory, Fortran code to the 'f' directory.

- Input and output files are defined in statements 'in=...' and 'out=...' in the first few lines of each script. In lieu of *in vivo* debugging, e.g. setting breakpoints or inspecting variables, which is not easily possible with a shell script we have set up detailed log files. A script's session dialog is written to the output file with .log.gz appended. For example, m/h0h0/6-comb.log.gz is the log of the run that made m/h0h0/6-comb.

4.1 Model-file preparation, Gaugeless Limit (0-glmod)

Our calculation uses the FeynArts model file for the MSSM with counterterms [35]. Since the calculation is carried out in the gaugeless limit (cf. Sect. 3), it speeds up calculations to prepare a variant of the model file where this limit is taken already at the level of the Feynman rules. At this stage we also introduce h_t (where $h_t^2 = 4\pi\alpha_t$) and substitute A_t by $X_t + \mu^*/t_\beta$. This is carried out by the 0-glmod script:

- Load the MSSMCT.mod model file.

- Modify/simplify couplings and remove those that become zero.

- Write out an MSSMCTgl.mod model file.

The original MSSMCT.mod model file contains both the counterterm vertices and the definitions of the (one-loop) renormalization constants. We moved the latter to model/MSSMCT.rc1 to make modification of the Feynman rules easier.

For each step we shall summarize usage as follows:

- Command line: model/0-glmod

- Main packages used: packages/Gaugeless.m, packages/XtSimplify.m

- Input: model/MSSMCT.mod.in

- Output: model/MSSMCT.mod, model/MSSMCTgl.mod

4.2 Step 1: Diagram generation (`1-amps`)

Diagrams are generated with `FeynArts` [36] using the `MSSMCTgl.mod` model file prepared above. Customized wrapper functions for the `FeynArts` functions are provided by `packages/FASettings.m`.

They ensure that all parts of the calculation consistently use the same `FeynArts` settings. Also diagram selection is simplified by filters defined in `FASettings.m`. Defining a diagram filter which works for an arbitrary topology is not entirely straightforward at the two-loop level but since only a few topologies contribute we have chosen a per-topology approach where particle choices can be made for each propagator.

For example, the two-loop self-energy diagrams for the neutral Higgs bosons are picked with the following statement:

```
sel[0][S[_] -> S[_]] = {
  t[3] && htb[6],
  t[3] && tb[6],
  t[3] && tb[6],
  t[3] && t[4] && htb[5],
  t[3] && htb[5|6],
  t[3] && htb[5],
  t[3] && t[5],
  t[5] && ht[3|4],
  t[3|4|5] && ht[3|4|5] }
```

The nine elements of the list correspond to the nine two-loop topologies of Fig. 1. For example, the entry `t[3]` mandates that propagator #3 must carry a top or stop. Likewise, `htb[5|6]` enforces a Higgs/Higgsino, top/stop, or bottom/sbottom on propagators #5 and/or #6.

- Command line: `scripts/1-amps SE CTO`

  `SE` is one of the self-energy/tadpole names given above (`h0h0`, `HHHH`, etc.) and `CTO` is the counterterm order, i.e. 0 for the virtual two-loop diagrams and 1 for the one-loop plus counterterm diagrams.

- Main packages used: `packages/FASettings.m`, `packages/Gaugeless.m`

- Input: (none)

- Output: `m/SE/1-amps.CTO` (amplitude), `m/SE/1-amps.CTO.ps.gz` (diagrams)

4.3 Step 2: Preparation for Tensor Reduction (`2-prep`)

The tensor reduction is traditionally the step which increases the number of terms in the amplitude most, therefore we inserted Step 2 before and Step 4 after the tensor reduction to keep the number of terms as small as possible at all times.

Several simplifications are carried out in Step 2: Firstly, the limit $p^2 \to 0$ is applied. This limit is part of the approximation employed here and moreover admits closed-form expressions (logs and dilogs) for the two-loop integrals.

Secondly, simplifications are performed in particular on the ubiquitous sfermion mixing matrix elements[1] $U_{ij} \equiv \mathtt{USf}[i,j]$ $(i,j = 1,2)$. For 2×2 matrices the unitarity relations $UU^\dagger = U^\dagger U = \mathbb{1}$ are of course easily written out, e.g. $U_{11}U_{11}^* + U_{12}U_{12}^* = 1$; the problem is that Mathematica by itself rarely arranges expressions in just the way that any of these sums can be substituted directly. This can be improved by adding definitions for single summands, e.g. $|U_{22}|^2 = |U_{11}|^2$. Because these apply

---

[1]The `USf` actually carry two more indices for sfermion type and generation: `USf[i,j,t,g]`, which we suppress here.
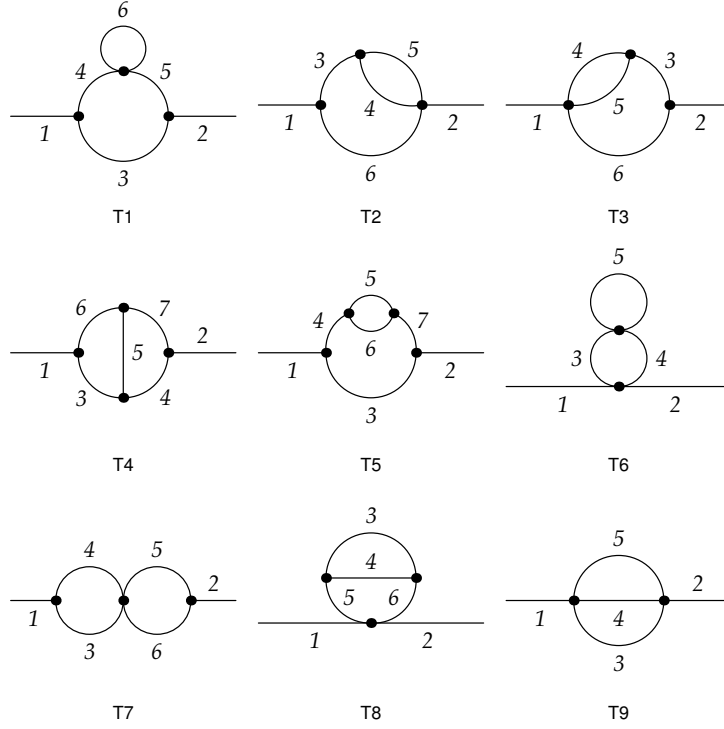
Figure 1. The 1PI two-point two-loop topologies with propagator numbers.

while Mathematica ponders the simplification strategy they increase the incentive for `Simplify` to choose the unitarity-simplified version. Mathematica does not allow to assign directly to the product of two matrix elements, however (technically the assignment is too 'deep' for Mathematica's upvalues[2]), hence we must introduce intermediate symbols:

$$\texttt{UCSf}[i,j] = \texttt{USf}[i,j]\,\texttt{USf}[i,j]^*, \tag{4.1}$$

$$\texttt{UCSf}[i,3] = \texttt{USf}[i,1]\,\texttt{USf}[i,2]^*, \qquad \texttt{UCSf}[3,3] = \texttt{USf}[1,1]\,\texttt{USf}[2,2]^*, \tag{4.2}$$

$$\texttt{UCSf}[3,j] = \texttt{USf}[1,j]\,\texttt{USf}[2,j]^*, \qquad \texttt{UCSf}[3,4] = \texttt{USf}[1,2]\,\texttt{USf}[2,1]^*. \tag{4.3}$$

The unitarity relations can now easily be formulated through the `UCSf`, e.g. `UCSf[2,2] = UCSf[1,1]`. The package `USfSimplify.m` contains the complete set of such rules and the function `USfSimplify` which encodes the above procedure.

After applying unitarity it is observed that almost all `UCSf` in the amplitude appear in the form of only four linear combinations:

$$\texttt{U2s1}[x] = (\texttt{UCSf}[1,3]\,x^* + \texttt{UCSf}[1,3]^*\,x)/2\,, \tag{4.4}$$

$$\texttt{U2s2}[x] = (\texttt{UCSf}[1,3]\,x^* - \texttt{UCSf}[1,3]^*\,x)/(2\mathrm{i})\,, \tag{4.5}$$

$$\texttt{U2c1}[x] = (\texttt{UCSf}[3,3]\,x^* + \texttt{UCSf}[3,4]^*\,x)/2\,, \tag{4.6}$$

$$\texttt{U2c2}[x] = (\texttt{UCSf}[3,3]\,x^* - \texttt{UCSf}[3,4]^*\,x)/(2\mathrm{i})\,. \tag{4.7}$$

---

[2]Mathematica associates a definition like `A*B^2 = 2` (internal notation: `Times[A, B^2] = 2`) with the multiplication function `Times`. For efficiency and also because `Times` is a protected function, Mathematica alternately allows the association with `A`, called an upvalue of `A`. The association cannot be made with `B`, however, because `B` appears squared and hence the assignment would be more than one level 'above' `B`.

Substituting these combinations indeed shortens the amplitude by about 1/3. Again, this choice is empirical and may well be specific to the corrections computed here. All relations for substitution and simplification are contained in `packages/U2Simplify.m`. Note in particular that, despite the suggestive notation, Eqs. (4.6) and (4.7) are complex quantities since the two summands are not conjugates of each other.

- Command line: `scripts/2-prep SE CTO`

  `SE` and `CTO` as in Step 1.

- Main packages used: `packages/USfSimplify.m`, `packages/U2Simplify.m`, `packages/SimplificationDefinitions.m`

- Input: `m/SE/1-amps.CTO`

- Output: `m/SE/2-prep.CTO`

4.4 Step 3: Tensor Reduction (`3-calc`)

The tensor reduction of the loop integrals is done through relatively straightforward application of the packages `TwoCalc` [37, 38] (for the virtual two-loop diagrams, `CTO` = 0) and `FormCalc` [39] (for the one-loop plus counterterm diagrams, `CTO` = 1).

Noteworthy here is perhaps that `TwoCalc` and `FormCalc` cannot be loaded into one Mathematica session due to symbol conflicts, but this is easily accomodated in the shell-script setup, i.e. the shell script invokes not one but two different Mathematica sessions depending on the value of `CTO`.

After the reduction, scaleless integrals (which vanish in dimensional regularization) are removed using `packages/UseSimplePackage.m`.

- Command line: `scripts/3-calc SE CTO`

  `SE` and `CTO` as in Step 1.

- Main packages used: `TwoCalc`, `FormCalc`, `packages/FCSettings.m`, `packages/UseSimplePackage.m`

- Input: `m/SE/2-prep.CTO`

- Output: `m/SE/3-calc.CTO`

4.5 Step 4: Simplification after Tensor Reduction (`4-simp`)

Step 4 is the counterpart of Step 2 after tensor reduction and reduces the size of each individual amplitude before they are combined in Step 6.

The key to decent performance comes from a 'tag' function[3] `DiagMark[`m_i`]` which the `FeynArts` wrapper functions of Step 1 (`packages/FASettings.m`) inserted for each diagram, where m_i are the masses that run in the loop. This `DiagMark` function induces a partitioning, i.e. the entire amplitude can be written as the sum of pieces, each of which is multiplied by a different `DiagMark`. Typically only few simplifications can be made across these pieces, hence one can restrict application of a simplification function `SIMP` to each piece, as in

    Collect[AMP, _DiagMark, SIMP]

Mathematica's `Collect` function reorders `AMP` as a linear combination of `DiagMark` instances and applies `SIMP` to each coefficient, which is of course much faster than applying it to the entire expression.

---

[3]The `DiagMark` function was originally developed for Refs. [40, 41].

- Command line: `scripts/4-simp` `SE` `CTO`

  `SE` and `CTO` as in Step 1.

- Main packages used: `packages/SimplificationDefinitions.m`, `packages/FCSettings.m`

- Input: `m/SE/3-calc.CTO`

- Output: `m/SE/4-simp.CTO`

4.6 Step 5: Calculation of the Renormalization Constants (`5-rc`)

The one-loop Renormalization Constants (RCs) are defined in `MSSMCT.rc1` (see Sect. 4.1), where the δM_V^2 ($V = W, Z$) require special attention, however. As already pointed out in Sect. 3, they are proportional to M_V^2 and would naively vanish in the gaugeless limit, while $\delta M_V^2/M_V^2$ must not vanish. Technically, we accomplish this by explicitly writing out the proportionality to M_V^2: `dMVsq1` $\rightarrow$ `dMVsq1MV2*MV2`, such that a possible `MV2` in the denominator of the coefficient of `dMVsq1` can cancel.

For the two-loop RCs of the cMSSM no complete set of definitions is available as a model file yet.[4] We added the necessary ones in the approximation relevant here in `model/MSSMCT.rc2`. For better extendability we tried to preserve the mathematical notation as much as possible, e.g. the two-loop wavefunction renormalization is defined via (the mass counterterm is defined separately)

```
dMHmat[Msq_, dZ1_, dZ2_, dM1_] :=
  1/2 (Msq.dZ2 + ConjugateTranspose[dZ2].Msq) +
  1/2 (ConjugateTranspose[dZ1].dM1 + ConjugateTranspose[dM1].dZ1) +
  1/4 ConjugateTranspose[dZ1].Msq.dZ1
```

which preserves the matrix notation of Eq. (2.30) of [29].

The RCs are further expanded in the dimension parameter $\varepsilon = (D - 4)/2$. For example, the one-loop top-mass RC, `dMf1[3,3]`, is decomposed as

```
dMf1[3,3] = RC[-1, dMf1[-1,3,3]] + RC[0, dMf1[0,3,3]]
```

where the two terms on the r.h.s. represent the coefficients of ε^{-1} and ε^0, respectively. The loop integrals are expanded with the `ExpandDel` function and powers of ε are collected with `DelSeries`, both defined in `packages/ExpandDel.m`. The notation allows to easily extract the terms actually needed in the final result in Step 6.

The $\mathcal{O}(\varepsilon^1)$ parts of the one-loop integrals often cancel in the final result. This cancellation should (and in the present case has been) checked, yet it needs some effort to see the coefficients of the ε^1-terms actually vanish. Once this is established, however, it is useful to set the ε^1-terms to zero manually, which is faster and gives a more compact result. In `packages/ExpandDel.m` this is done through rules defined in `AssumeVanishingDel`. To re-enable checking, set `AssumeVanishingDel = {}`.

The actual calculation of the RCs is carried out with `FormCalc`, for each self-energy and tadpole. So as not to re-compute RCs many times, they are cached in the `m/rc` subdirectory. Take care that this cache is not controlled by the makefile.

- Command line: `scripts/5-rc` `SE`

  `SE` as in Step 1.

- Main packages used: `packages/RenConst.m`, `packages/ExpandDel.m`

- Input: `m/SE/4-simp.1`

- Output: `m/SE/5-rc`, `m/rc/*`

---

[4]A complete list for the counterterms of the Higgs sector in the complex MSSM can be found in Ref. [29].

4.7 Step 6: Combining Results (6-comb)

The final analytical step for each self-energy/tadpole joins the amplitudes with the renormalization constants and, unless a debug flag is set, retains only the coefficient of ε^0, i.e. the finite part. It also applies a (mostly empirical) blend of simplification functions to get the compact expression for code generation.

Two of these simplifications merit a closer look, for they affect also the numerical precision of the final result. The difference and sum of the sfermion mass squares not only appear frequently in the amplitude, they are also computed as a by-product of diagonalization anyway, with effectively no loss in precision even for degenerate sfermion masses. We thus substitute

```
MSf2[3,t,g] = MSf2[2,t,g] - MSf2[1,t,g],
MSf2[4,t,g] = MSf2[1,t,g] + MSf2[2,t,g].
```

Considering that in the gaugeless limit with vanishing bottom mass the remaining sbottom mass square is equal to the stop-sector breaking parameter, $m_{\tilde{b}}^2 = m_{\tilde{t}_L}^2$, the difference of stop and sbottom mass squares can be computed with better precision, too, which is particularly relevant for the $H^\pm$ self-energy where this difference appears in denominators. Hence we also introduce

```
MSq2Diff[i,j] = MSf2[i,4,3] - MSf2[j,3,3].
```

- Command line: scripts/6-comb SE

  SE as in Step 1.

- Main packages used: packages/FinalSimp.m, packages/ExpandDel.m

- Input: m/SE/4-simp.0, m/SE/4-simp.1, m/rc/*

- Output: m/SE/6-comb

4.8 Step 7: Generating Code (7-code)

The last step writes out optimized Fortran code using FormCalc's code-generation functions [42]. Self-energies and tadpoles are arranged in three groups, such that FeynHiggs can also perform a partial evaluation depending on its flags (e.g. if 2×2 mixing only is requested):

1. A0A0, HmHp,
2. h0h0, HHHH, h0HH, h0, HH,
3. h0A0, HHA0, A0.

Abbreviations are introduced for both loop integrals and common subexpressions, which shortens the entire expression significantly. Some variables need to be disambiguated, for example the sfermion masses in the gaugeless limit must not use the same identifiers as other sfermion masses in FeynHiggs.

Lastly, the handwritten part of the code from the 'static' subdirectory is copied to the Fortran output directory 'f', which is then complete and replaces the entire src/TwoLoop/TLsp branch of the FeynHiggs source tree.

- Command line: scripts/7-code

- Main packages used: FormCalc, packages/FCSettings.m

- Input: m/*/6-comb, static/*

- Output: f/*

5 Conclusions

We have described the implementation of the $\mathcal{O}(\alpha_t^2)$ Higgs-mass corrections for the complex MSSM in `FeynHiggs`. The original calculation of Ref. [29] was restructured entirely for this purpose and the new code is included in the `FeynHiggs` source code, available at `http://feynhiggs.de`. The calculation is split into seven tasks which were explained in detail.

While each step is specific to the corrections computed here the code can be used as a template due to its modular structure and adapted with little effort to similar calculations. The scripting framework we presented can also be considered a showcase of how to flexibly use `FeynArts` and `FormCalc` together with other packages to generalize applicability beyond one-loop level.

As with all practical recipes, the proof is in the pudding. Our description herein should suffice to get an overview of the code and its structure. What cannot be adequately conveyed in writing is that significant effort has been spent to make the code readable and 'obvious'.

Acknowledgments

References

[1] G. Aad *et al.* [ATLAS Collaboration], Phys. Lett. B **716** (2012) 1 [arXiv:1207.7214].

[2] S. Chatrchyan *et al.* [CMS Collaboration], Phys. Lett. B **716** (2012) 30 [arXiv:1207.7235].

[3] G. Piacquadio [ATLAS Collaboration], J. Bendavid [CMS Collaboration], M. Dührssen [ATLAS+CMS Collaborations], talks given at the 50th Rencontres de Moriond, La Thuile, Aosta valley, Italy, March 2015.

[4] A. Pilaftsis, Phys. Rev. D **58** (1998) 096010 [hep-ph/9803297], Phys. Lett. B **435** (1998)88 [hep-ph/9805373].

[5] D.A. Demir, Phys. Rev. D **60** (1999) 055006 [hep-ph/9901389].

[6] A. Pilaftsis, C.E.M. Wagner, Nucl. Phys. B **553** (1999) 3 [hep-ph/9902371].

[7] S. Heinemeyer, Eur. Phys. J. C **22** (2001) 521 [hep-ph/0108059].

[8] J.A. Casas, J.R. Espinosa, M. Quiros, A. Riotto, Nucl. Phys. B **436** (1995) 3 [Erratum-ibid. B **439** (1995) 466] [hep-ph/9407389]. M.S. Carena, J.R. Espinosa, M. Quiros, C.E.M. Wagner, Phys. Lett. B **355** (1995) 209 [hep-ph/9504316].

[9] S. Heinemeyer, W. Hollik, G. Weiglein, Phys. Rev. D **58** (1998) 091701 [hep-ph/9803277], Phys. Lett. B **440** (1998) 296 [hep-ph/9807423].

[10] S. Heinemeyer, W. Hollik, G. Weiglein, Eur. Phys. J. C **9** (1999) 343 [hep-ph/9812472].

[11] S. Heinemeyer, W. Hollik, G. Weiglein, Phys. Lett. B **455** (1999) 179 [hep-ph/9903404]. M.S. Carena, H.E. Haber, S. Heinemeyer, W. Hollik, C.E.M. Wagner, G. Weiglein, Nucl. Phys. B **580** (2000) 29 [hep-ph/0001002].

[12] S. Heinemeyer, W. Hollik, H. Rzehak, G. Weiglein, Eur. Phys. J. C **39** (2005) 465 [hep-ph/0411114].

[13] S. Borowka, T. Hahn, S. Heinemeyer, G. Heinrich, W. Hollik, Eur. Phys. J. C **74** (2014) 2994 [arXiv:1404.7074].

[14] G. Degrassi, S. Di Vita, P. Slavich, Eur. Phys. J. C **75** (2015) 2, 61 [arXiv:1410.3432 [hep-ph]].

[15] S. Borowka, T. Hahn, S. Heinemeyer, G. Heinrich, W. Hollik, arXiv:1505.03133 [hep-ph].

[16] R. Harlander, P. Kant, L. Mihaila, M. Steinhauser, Phys. Rev. Lett. **100** (2008) 191602; *ibid.* **101** (2008) 039901 [arXiv:0803.0672], JHEP **1008** (2010) 104 [arXiv:1005.5709].

[17] R.-J. Zhang, Phys. Lett. B **447** (1999) 89 [hep-ph/9808299]. J.R. Espinosa, R.-J. Zhang, Nucl. Phys. B **586** (2000) 3 [hep-ph/0003246], JHEP **0003** (2000) 026 [hep-ph/9912236]. J.R. Espinosa, I. Navarro, Nucl. Phys. B **615** (2001) 82 [hep-ph/0104047]. G. Degrassi, P. Slavich, F. Zwirner, Nucl. Phys. B **611** (2001) 403 [hep-ph/0105096]. R. Hempfling, A.H. Hoang, Phys. Lett. B **331** (1994) 99 [hep-ph/9401219]. A. Brignole, G. Degrassi, P. Slavich, F. Zwirner, Nucl. Phys. B **643** (2002) 79 [hep-ph/0206101]. A. Dedes, G. Degrassi, P. Slavich, Nucl. Phys. B **672** (2003) 144 [hep-ph/0305127].

[18] J.R. Espinosa, R.-J. Zhang, Nucl. Phys. B **586** (2000) 3 [hep-ph/0003246].

[19] A. Brignole, G. Degrassi, P. Slavich, F. Zwirner, Nucl. Phys. B **631** (2002) 195 [hep-ph/0112177].

[20] G. Degrassi, S. Heinemeyer, W. Hollik, P. Slavich, G. Weiglein, Eur. Phys. J. C **28** (2003) 133 [hep-ph/0212020].

[21] S. Heinemeyer, W. Hollik, G. Weiglein, Phys. Rept. **425**, 265 (2006) [hep-ph/0412214].

[22] B.C. Allanach, A. Djouadi, J. L. Kneur, W. Porod, P. Slavich, JHEP **0409** (2004) 044 [hep-ph/0406166].

[23] S.P. Martin, Phys. Rev. D **65** (2002) 116003 [hep-ph/0111209], Phys. Rev. D **66** (2002) 096001 [hep-ph/0206136], Phys. Rev. D **67** (2003) 095012 [hep-ph/0211366], Phys. Rev. D **68** (2003) 075002 [hep-ph/0307101], Phys. Rev. D **70** (2004) 016005 [hep-ph/0312092], Phys. Rev. D **71** (2005) 016012 [hep-ph/0405022], Phys. Rev. D **71** (2005) 116004 [hep-ph/0502168]. S.P. Martin, D.G. Robertson, Comput. Phys. Commun. **174** (2006) 133 [hep-ph/0501132].

[24] S.Y. Choi, M. Drees, J.S. Lee, Phys. Lett. B **481** (2000) 57 [hep-ph/0002287]. T. Ibrahim, P. Nath, Phys. Rev. D **63** (2001) 035009 [hep-ph/0008237], Phys. Rev. D **66** (2002) 015005 [hep-ph/0204092].

[25] M.S. Carena, J.R. Ellis, A. Pilaftsis, C.E.M. Wagner, Nucl. Phys. B **586** (2000) 92 [hep-ph/0003180].

[26] M. Frank, T. Hahn, S. Heinemeyer, W. Hollik, H. Rzehak, G. Weiglein, JHEP **0702** (2007) 047 [hep-ph/0611326].

[27] S. Heinemeyer, W. Hollik, H. Rzehak, G. Weiglein, Phys. Lett. B **652** (2007) 300 [arXiv:0705.0746].

[28] W. Hollik, S. Paßehr, Phys. Lett. B **733**, 144 (2014) [arXiv:1401.8275].

[29] W. Hollik, S. Paßehr, JHEP **1410** (2014) 171 [arXiv:1409.1687].

[30] S. Heinemeyer, W. Hollik, G. Weiglein, Comput. Phys. Commun. **124** (2000) 76 [hep-ph/9812320].

[31] T. Hahn, S. Heinemeyer, W. Hollik, H. Rzehak, G. Weiglein, Comput. Phys. Commun. **180** (2009) 1426.

[32] T. Hahn, S. Heinemeyer, W. Hollik, H. Rzehak, G. Weiglein, Nucl. Phys. Proc. Suppl. **205-206** (2010) 152 [arXiv:1007.0956].

[33] T. Hahn, C. Schappacher, Comput. Phys. Commun. **143** (2002) 54 [hep-ph/0105349].

[34] T. Hahn, J.I. Illana, Nucl. Phys. Proc. Suppl. **160** (2006) 101 [hep-ph/0607049].

[35] T. Fritzsche, T. Hahn, S. Heinemeyer, F. von der Pahlen, H. Rzehak, C. Schappacher, Comput. Phys. Commun. **185** (2014) 1529 [arXiv:1309.1692].

[36] T. Hahn, Comput. Phys. Commun. **140** (2001) 418 [hep-ph/0012260].

[37] G. Weiglein, R. Scharf, M. Böhm, Nucl. Phys. B **416** (1994) 606 [hep-ph/9310358].

[38] G. Weiglein, R. Mertig, R. Scharf, M. Böhm, in: New computing techniques in physics research II, 617–623, La Londe-les-Maures 1992.

[39] T. Hahn, M. Perez-Victoria, Comput. Phys. Commun. **118** (1999) 153 [hep-ph/9807565].

[40] S. Heinemeyer, D. Stöckinger, G. Weiglein, Nucl. Phys. B **690** (2004) 62 [hep-ph/0312264].

[41] S. Heinemeyer, D. Stöckinger, G. Weiglein, Nucl. Phys. B **699** (2004) 103 [hep-ph/0405255].

[42] T. Hahn, PoS ACAT **2010** (2010) 078 [arXiv:1006.2231].