DESY-20-162

# munuSSM: A python package for the $\mu$-from-$\nu$ Supersymmetric Standard Model

Thomas Biekötter*
*DESY, Notkestraße 85, 22607 Hamburg, Germany*



**Abstract**

We present the public python package munuSSM that can be used for phenomenological studies in the context of the $\mu$-from-$\nu$ Supersymmetric Standard Model ($\mu\nu$SSM). The code incorporates the radiative corrections to the neutral scalar potential at the full one-loop level. Sizable higher-order corrections, required for an accurate prediction of the SM-like Higgs-boson mass, can be consistently included via an automated link to the public code FeynHiggs. In addition, a calculation of effective couplings and branching ratios of the neutral and charged Higgs bosons is implemented. This provides the required ingredients to check a benchmark point against collider constraints from searches for additional Higgs bosons via an interface to the public code HiggsBounds. At the same time, the signal rates of the SM-like Higgs boson can be tested applying the experimental results implemented in the public code HiggsSignals. The python package is constructed in a flexible and modular way, such that it provides a simple framework that can be extended by the user with further calculations of observables and constraints on the model parameters.

The source code of munuSSM and instructions for the installation are available at:

https://gitlab.com/thomas.biekoetter/munussm

*thomas.biekoetter@desy.de

# 1 Introduction

Supersymmetry (Susy) is one of the prime candidates for physics Beyond the Standard Model (BSM). A particularly well motivated Susy model is the so-called $\mu$-from-$\nu$ Supersymmetric Standard Model ($\mu\nu$SSM) [1, 2]. Beyond the usual benefits of low-scale Susy, i.e., providing a solution to the hierarchy problem and allowing for the unification of the three Standard Model (SM) gauge couplings, the $\mu\nu$SSM incorporates an electroweak seesaw mechanism. Without introducing any scales beyond the Susy-breaking scale $M_S$, the tiny neutrino masses and their mixing pattern can be accommodated assuming neutrino Yukawa couplings $Y^\nu$ of the order of the electron Yukawa coupling. Furthermore, the superpotential is scale invariant and the $\mu$-term of the superpotential of the Minimal Supersymmetric Standard Model (MSSM) is generated dynamically during Electroweak Symmetry Breaking (EWSB). Apart from the Higgs doublets, also the scalar partners of the neutrinos (called sneutrinos) obtain a Vacuum Expectation Value (vev) during EWSB. The tree-level mass of the SM-like Higgs boson receives additional contributions stemming from portal couplings between the Higgs doublet fields and the right-handed sneutrinos. Thus, compared to the MSSM, a value of $\sim 125$ GeV can be achieved with fewer radiative corrections.

The $\mu\nu$SSM is especially interesting in view of EWSB. In this model the stability of the proton is assured by forbidding operators breaking baryon number [2]. However, it does not assume $R$-parity conservation and the breaking of lepton number is induced via terms proportional to $Y^\nu$ by construction. Thus, the left- and right-handed sneutrinos mix with the neutral scalar components of the Higgs doublet superfields. Apart from that, the scalar partners of the leptons (called sleptons) mix with the charged scalar components of the Higgs doublet superfields. Neglecting CP violation, as we will do throughout this paper, the Higgs sector of the $\mu\nu$SSM consists of a total of 8 CP-even, 7 CP-odd, and $2 \times 7$ charged scalars. In addition, there are the usual pseudoscalar and charged Goldstone bosons $G^0$ and $G^\pm$. During EWSB all of the 8 neutral scalar fields obtain a vev. While the mixing of the doublet Higgs bosons and the gauge-singlet right-handed sneutrinos can in principle be arbitrarily large, the mixing between the left-handed sneutrinos and the Higgs doublets is suppressed by the small values of $Y^\nu$. This decoupling is also reflected in a large hierarchy between the vevs. The vevs of the Higgs doublets $v_u$ and $v_d$ and the vevs of the right-handed sneutrinos $v_{iR}$ ($i = 1, 2, 3$) are related to the breaking scale of the EW symmetry and Susy. The vevs of the left-handed sneutrinos $v_{iL}$, on the other hand, are related to the breaking of lepton number, and therefore suppressed by a factor of $Y^\nu$ compared to $v_d$, $v_u$ and $v_{iR}$.

These unique features motivated the precise analysis of the Higgs sector of the model, including the radiative corrections at full one-loop level. At first, we studied a simplified version of the $\mu\nu$SSM with a single right-handed neutrino superfield [3]. Later on, we extended the calculation to the complete model with three right-handed neutrino superfields [4]. In the latter, three non-zero left-handed neutrino masses can already be accommodated at tree level. It was found that for the SM-like Higgs-boson mass the mixing effects between doublet fields and right-handed sneutrinos are important at loop level and have to be taken into account, while the tiny mixing with the left-handed sneutrinos does not play a role. However, the left-handed sneutrinos themselves are subject to potentially large corrections proportional to $Y^\nu A_t Y_t / v_{iL}$, in which the suppression of the factors $Y^\nu$ is compensated by the left-handed vevs in the denominator and a factor of the soft scalar top (called stop) mixing parameter $A_t$ times the top Yukawa coupling $Y_t$.

It is known from the MSSM that corrections to the Higgs-boson mass beyond one-loop level are sizable and have to be taken into account [5–7]. These higher-order contributions should be included in an approximate form also in the $\mu\nu$SSM in order to obtain an accurate prediction. Combining the higher-order effects with the full one-loop result, it was shown that the $\mu\nu$SSM can easily accommodate a Higgs boson at $\sim$ 125 GeV that reproduces the measured signal rates within the current experimental uncertainties [4]. Apart from that, interesting new Higgs physics can be realized at relatively low masses, since the right- and the left-handed sneutrinos could have escaped discovery so far even for masses below 125 GeV [4]. Note that the right-handed sneutrinos are gauge singlets, such that they naturally have reduced couplings to the SM particles. In fact, they only couple to the SM via the mixing with the doublet fields, for instance the SM-like Higgs boson. Such a scenario is particularly interesting, as it can be probed not only by directly searching for additional Higgs bosons, but also indirectly by measuring possible deviations from the SM predictions of the couplings of the Higgs boson at 125 GeV [4, 8]. A possible detection of light left-handed sneutrinos requires dedicated searches when they are the lightest SUSY particle, since their decay must proceed via $R$-parity violating couplings [9–11].

In this paper we present a tool for the phenomenological study of the $\mu\nu$SSM. First and foremost, it makes the one-loop corrections to the Higgs potential publicly available in terms of the momentum-dependent renormalized scalar self energies. These are used in combination with leading higher-order corrections from the public code `FeynHiggs` [7, 12–18] to accurately predict the particle masses of the neutral scalars, in particular the SM-like Higgs-boson mass. In addition, the radiative corrections to the mixing matrix elements are used to obtain effective couplings of the scalars to the SM particles. Furthermore, the calculation of the branching ratios of the neutral and charged scalars is implemented. For decays into SM particles, the branching ratios are obtained by rescaling the SM predictions [19, 20] by the effective couplings. For decays to BSM final states, the branching ratios are calculated from scratch at leading order, however including radiative corrections by rotating the tree-level couplings into the loop-corrected mass eigenstate basis. The effective couplings and branching ratios can be directly interfaced to the public codes `HiggsBounds` [21–26] and `HiggsSignals` [27–30] to test a benchmark point against collider constraints. The interface to `HiggsBounds` also provides the LHC cross sections normalized to the SM prediction, which are extracted from the effective couplings.

The paper is organized as follows. We start by briefly introducing the model in Sect. 2. The overall structure of the code `munuSSM` and its subpackages are described in Sect. 3, paying special attention to the links to other public codes in Sect. 3.1. Afterwards, we explain the installation process and the basic user instructions in Sect. 3.2 and Sect. 3.3. A simple example script is given in Sect. 3.3.1. We conclude in Sect. 4.

## 2 The $\mu$-from-$\nu$ Supersymmetric Standard Model

In this section we provide the basic definitions and conventions under which the model predictions were implemented. A more detailed motivation and a review of the $\mu\nu$SSM can be found in Ref. [31]. The calculation of the radiative corrections to the Higgs potential are described in detail in Refs. [3, 4].

The superpotential of the $\mu\nu$SSM is written as

$$W = \epsilon_{ab} \left( Y^e_{ij} \, \hat{H}^a_d \, \hat{L}^b_i \, \hat{e}^c_j + Y^d_{ij} \, \hat{H}^a_d \, \hat{Q}^b_i \, \hat{d}^c_j + Y^u_{ij} \, \hat{H}^b_u \, \hat{Q}^a_i \, \hat{u}^c_j \right) + \epsilon_{ab} \left( Y^\nu_{ij} \, \hat{H}^b_u \, \hat{L}^a_i \, \hat{\nu}^c_j - \lambda_i \, \hat{\nu}^c_i \, \hat{H}^b_u \hat{H}^a_d \right) + \frac{1}{3} \kappa_{ijk} \hat{\nu}^c_i \hat{\nu}^c_j \hat{\nu}^c_k \, , \tag{1}$$

where $\hat{H}^T_d = (\hat{H}^0_d, \hat{H}^-_d)$ and $\hat{H}^T_u = (\hat{H}^+_u, \hat{H}^0_u)$ are the Higgs doublet superfields, $\hat{Q}^T_i = (\hat{u}_i, \hat{d}_i)$ and $\hat{L}^T_i = (\hat{\nu}_i, \hat{e}_i)$ are the left-chiral quark and lepton superfields, and $\hat{u}^c_j$, $\hat{d}^c_j$, $\hat{e}^c_j$ and $\hat{\nu}^c$ are the right-chiral quark and lepton superfields. $i, j = 1, 2, 3$ are the family indices, and $a, b = 1, 2$ are indices of the fundamental representation of SU(2) with $\epsilon_{ab} = 1$. The colour indices are not written out.

In the framework of low-energy SUSY breaking, the soft Lagrangian of the $\mu\nu$SSM is given by [32]

$$\begin{aligned} -\mathcal{L}_{\text{soft}} = \; & \epsilon_{ab} \left( T^e_{ij} \, H^a_d \, \widetilde{L}^b_{iL} \, \widetilde{e}^*_{jR} + T^d_{ij} \, H^a_d \, \widetilde{Q}^b_{iL} \, \widetilde{d}^*_{jR} + T^u_{ij} \, H^b_u \widetilde{Q}^a_{iL} \widetilde{u}^*_{jR} + \text{h.c.} \right) \\ & + \epsilon_{ab} \left( T^\nu_{ij} \, H^b_u \, \widetilde{L}^a_{iL} \widetilde{\nu}^*_{jR} - T^\lambda_i \, \widetilde{\nu}^*_{iR} \, H^a_d H^b_u + \frac{1}{3} T^\kappa_{ijk} \, \widetilde{\nu}^*_{iR} \widetilde{\nu}^*_{jR} \widetilde{\nu}^*_{kR} \; + \text{h.c.} \right) \\ & + \left( m^2_{\widetilde{Q}} \right)_{ij} \widetilde{Q}^{a*}_{iL} \widetilde{Q}^a_{jL} + \left( m^2_{\widetilde{u}} \right)_{ij} \widetilde{u}^*_{iR} \widetilde{u}_{jR} + \left( m^2_{\widetilde{d}} \right)_{ij} \widetilde{d}^*_{iR} \widetilde{d}_{jR} + \left( m^2_{\widetilde{L}} \right)_{ij} \widetilde{L}^{a*}_{iL} \widetilde{L}^a_{jL} \\ & + \left( m^2_{\widetilde{\nu}} \right)_{ij} \widetilde{\nu}^*_{iR} \widetilde{\nu}_{jR} + \left( m^2_{\widetilde{e}} \right)_{ij} \widetilde{e}^*_{iR} \widetilde{e}_{jR} + m^2_{H_d} H^{a*}_d H^a_d + m^2_{H_u} H^{a*}_u H^a_u \end{aligned} \tag{2}$$

$$\begin{aligned} & + \left( \left( m^2_{H_d \widetilde{L}} \right)_i H^{a*}_d \widetilde{L}^a_{iL} + \text{h.c.} \right) \\ & + \frac{1}{2} \left( M_3 \, \widetilde{g} \, \widetilde{g} + M_2 \, \widetilde{W} \, \widetilde{W} + M_1 \, \widetilde{B}^0 \, \widetilde{B}^0 + \text{h.c.} \right) \, . \end{aligned} \tag{3}$$

The parameters $m^2_{H_d \widetilde{L}}$ are absent at tree level, as they are non-diagonal in field space. However, in the code the terms are taken into account, because they are required for the renormalization of the Higgs potential (see Refs. [3, 4] for details). In addition, flavour mixing is neglected in the quark and the squark sector, such that the corresponding soft mass parameters only have diagonal non-zero entries $m^2_{\widetilde{Q}_i}$, $m^2_{\widetilde{u}_i}$ and $m^2_{\widetilde{d}_i}$. The soft trilinear couplings are written as $T^u_i = A^u_i Y^u_i$, $T^d_i = A^d_i Y^d_i$, where $Y^u_i$ and $Y^d_i$ are the diagonal entries of the Yukawa couplings of the up- and down-type quarks and no summation over repeated indices is implied. In the lepton sector, the flavour symmetries are broken automatically after EWSB. Thus, we decompose the soft trilinear couplings as $T^e_{ij} = A^e_{ij} Y^e_{ij}$ and $T^\nu_{ij} = A^\nu_{ij} Y^\nu_{ij}$, again without summation over repeated indices. The lepton-flavour mixing is suppressed by factors of $Y^\nu_{ij}$ and therefore only sizable for the light left-handed neutrinos. Finally, we write the portal coupling and the self coupling of the right-handed sneutrinos as $T^\lambda_i = A^\lambda_i \lambda_i$ and $T^\kappa_{ijk} = A^\kappa_{ijk} \kappa_{ijk}$, noting that both $\kappa_{ijk}$ and $A^\kappa_{ijk}$ are symmetric under the exchange of indices.

The soft terms together with the $D$-term and $F$-term contributions from the superpotential define the tree-level neutral scalar potential

$$V^{(0)} = V_{\text{soft}} + V_F + V_D \, , \tag{4}$$

with

$$
\begin{aligned}
V_{\text{soft}} = & \left(T^{\nu}_{ij}\, H^0_u\, \widetilde{\nu}_{iL}\, \widetilde{\nu}^*_{jR} - T^{\lambda}_i\, \widetilde{\nu}^*_{iR}\, H^0_d H^0_u + \frac{1}{3} T^{\kappa}_{ijk}\, \widetilde{\nu}^*_{iR} \widetilde{\nu}^*_{jR} \widetilde{\nu}^*_{kR} \ + \text{h.c.}\right) \\
& + \ \left(m^2_{\widetilde{L}}\right)_{ij} \widetilde{\nu}^*_{iL} \widetilde{\nu}_{jL} + \left(m^2_{\widetilde{\nu}}\right)_{ij} \widetilde{\nu}^*_{iR} \widetilde{\nu}_{jR} + m^2_{H_d} H^{0*}_d H^0_d + m^2_{H_u} H^{0*}_u H^0_u \ ,
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
V_F = & \lambda_j \lambda_j H^0_d H^{0^*}_d H^0_u H^{0^*}_u + \lambda_i \lambda_j \tilde{\nu}^*_{iR} \tilde{\nu}_{jR} H^0_d H^{0*}_d + \lambda_i \lambda_j \tilde{\nu}^*_{iR} \tilde{\nu}_{jR} H^0_u H^{0*}_u \\
& + \kappa_{ijk} \kappa_{ljm} \tilde{\nu}^*_{iR} \tilde{\nu}_{lR} \tilde{\nu}^*_{kR} \tilde{\nu}_{mR} - \big(\kappa_{ijk} \lambda_j \tilde{\nu}^*_{iR} \tilde{\nu}^*_{kR} H^{0*}_d H^{0*}_u - Y^{\nu}_{ij} \kappa_{ljk} \tilde{\nu}_{iL} \tilde{\nu}_{lR} \tilde{\nu}_{kR} H^0_u \\
& + \ Y^{\nu}_{ij} \lambda_j \tilde{\nu}_{iL} H^{0*}_d H^{0*}_u H^0_u + Y^{\nu}_{ij} \lambda_k \tilde{\nu}^*_{iL} \tilde{\nu}_{jR} \tilde{\nu}^*_{kR} H^0_d + \text{h.c.}\big) \\
& + Y^{\nu}_{ij} Y^{\nu}_{ik} \tilde{\nu}^*_{jR} \tilde{\nu}_{kR} H^0_u H^{0*}_u + Y^{\nu}_{ij} Y^{\nu}_{lk} \tilde{\nu}_{iL} \tilde{\nu}^*_{lL} \tilde{\nu}^*_{jR} \tilde{\nu}_{kR} + Y^{\nu}_{ji} Y^{\nu}_{ki} \tilde{\nu}_{jL} \tilde{\nu}^*_{kL} H^0_u H^{0*}_u \ ,
\end{aligned} \tag{6}
$$

$$
V_D = \frac{1}{8}\left(g^2_1 + g^2_2\right)\left(\widetilde{\nu}_{iL} \widetilde{\nu}^*_{iL} + H^0_d H^{0^*}_d - H^0_u H^{0^*}_u\right)^2 \ . \tag{7}
$$

During EWSB the neutral scalar fields acquire a vev. We use the decomposition

$$
H^0_d = \frac{1}{\sqrt{2}}\left(H^{\mathcal{R}}_d + v_d + \mathrm{i}\, H^{\mathcal{I}}_d\right) \ , \tag{8}
$$

$$
H^0_u = \frac{1}{\sqrt{2}}\left(H^{\mathcal{R}}_u + v_u + \mathrm{i}\, H^{\mathcal{I}}_u\right) \ , \tag{9}
$$

$$
\widetilde{\nu}_{iR} = \frac{1}{\sqrt{2}}\left(\widetilde{\nu}^{\mathcal{R}}_{iR} + v_{iR} + \mathrm{i}\, \widetilde{\nu}^{\mathcal{I}}_{iR}\right) \ , \tag{10}
$$

$$
\widetilde{\nu}_{iL} = \frac{1}{\sqrt{2}}\left(\widetilde{\nu}^{\mathcal{R}}_{iL} + v_{iL} + \mathrm{i}\, \widetilde{\nu}^{\mathcal{I}}_{iL}\right) \ , \tag{11}
$$

such that the vevs are given by[1]

$$
\langle H^0_d \rangle = \frac{v_d}{\sqrt{2}} \ , \quad \langle H^0_u \rangle = \frac{v_u}{\sqrt{2}} \ , \quad \langle \widetilde{\nu}_{iR} \rangle = \frac{v_{iR}}{\sqrt{2}} \ , \quad \langle \widetilde{\nu}_{iL} \rangle = \frac{v_{iL}}{\sqrt{2}} \ . \tag{12}
$$

The subscripts $^{\mathcal{R}}$ and $^{\mathcal{I}}$ denote CP-even and -odd components of each scalar field, respectively. To make a connection to the SM and the MSSM, we define the parameters

$$
v^2 = v^2_u + v^2_d + v_{iL} v_{iL} \sim 246 \text{ GeV} \quad \text{and} \quad \tan\beta = \frac{v_u}{v_d} \ . \tag{13}
$$

As already mentioned in Sect. 1, the size of the left-handed vevs $v_{iL}$ is suppressed by factors of $Y^{\nu}_{ij}$ compared to the other vevs. Hence, they are of the order of $\sim 10^{-5}$ to $10^{-4}$ GeV. The minimization or tadpole equations relate the soft mass parameters to the vevs. For numerical reasons it is most convenient to use the vevs as input parameters and solve the tadpole equations for the soft masses squared $m^2_{H_d}$, $m^2_{H_u}$, $(m^2_{\widetilde{L}})_{ii}$ and $(m^2_{\widetilde{\nu}})_{ii}$. The precise form of the tadpole equations can be found in Ref. [4].

The expressions for the tree-level masses of all particles of the model in terms of the parameters defined above can be found in Ref. [33]. Since they are rather lengthy we do not

[1] We will refer to the parameters $v_u$, $v_d$, $v_{iL}$ and $v_{iR}$ as vevs interchangeably.

repeat them here. The expressions for the tree-level couplings of the particles are even larger due to the complicated mixing in the scalar sector, such that we do not state them either. Instead, we provide a `FeynArts` [34] modelfile upon request that contains the couplings in the 't Hooft-Feynman gauge in `Mathematica` syntax.[2] The modelfile was initially created with the public tool `SARAH` [37], but further modified by hand to allow the usage of the tool `FormCalc` [38], which by default cannot process the huge expressions for the couplings produced by `SARAH`.

In the code `munuSSM`, the calculation of the tree-level spectrum and the corresponding mixing matrices, as well as the tree-level couplings, are evaluated in Fortran subroutines. This allows for a larger floating-point precision, which is necessary due to the tiny $R$-parity violating mixing effects and the large hierarchy between the masses in the neutral fermion sector. Apart from that, the usage of Fortran vastly improves the running time compared to an implementation in python. We note that the running time is currently dominated by the calculation of the complete set of tree-level couplings.

## 2.1 Radiative corrections in the Higgs sector

The scalar sector of the $\mu\nu$SSM is subject to sizable radiative corrections that have to be taken into account in each phenomenologically viable analysis. Making these corrections available to the public is (so far) the core idea of this project. The objects that contain the corrections are the renormalized scalar self energies $\hat{\Sigma}_{\phi_i\phi_j}(p^2)$, which enter the renormalized inverse propagator matrix of the fields $\phi_i$,

$$\hat{\Gamma}_{ij} = \mathrm{i}\left[p^2\delta_{ij} - (m_i^2 - \hat{\Sigma}_{\phi_i\phi_j}(p^2))\right] \ . \tag{14}$$

In this expression the indices $i$ and $j$ run over the number of fields that mix with each other, $p$ is the momentum and $m_i^2$ are the eigenvalues of the corresponding tree-level mass matrix. Implemented in the code are the corrections to the CP-even and CP-odd neutral scalars $h_i$ and $A_i$. These are given by

$$\hat{\Sigma}_{h_ih_j} = \hat{\Sigma}^{(1)}_{h_ih_j}(p^2) + \hat{\Sigma}^{(2')}_{h_ih_j} + \hat{\Sigma}^{\text{resum.}}_{h_ih_j} \tag{15}$$

$$\hat{\Sigma}_{A_iA_j} = \hat{\Sigma}^{(1)}_{A_iA_j}(p^2) \ . \tag{16}$$

Here, $\hat{\Sigma}^{(1)}_{h_ih_j}$ and $\hat{\Sigma}^{(1)}_{A_iA_j}$ contain the full one-loop corrections, including the momentum dependence. In addition, leading two-loop corrections for the CP-even fields $h_i$ are included in terms of $\hat{\Sigma}^{(2')}_{h_ih_j}$. Finally, higher-order corrections arising from the resummation of logarithmic contributions are taken into account in $\hat{\Sigma}^{\text{resum.}}_{h_ih_j}$.

The corrections beyond one-loop level are taken from the public code `FeynHiggs`. They are crucial to obtain a precise prediction for the SM-like Higgs-boson mass. The one-loop pieces were calculated in Ref. [4] in a mixed $\overline{\text{DR}}$-On Shell (OS) scheme that is consistent

[2] The couplings of the gravitino and the axino, both potential dark matter candidates in the $\mu\nu$SSM [35, 36], are not included. However, they only play a role for the DM phenomenology and are irrelevant for the Higgs and collider physics.

with the one of `FeynHiggs`. For generic scalar fields $\phi_i$, they can be written as

$$\hat{\Sigma}^{(1)}_{\phi_i\phi_j}(p^2) = \Sigma^{(1)}_{\phi_i\phi_j}(p^2) + \frac{1}{2}p^2\left(\delta Z_{\phi_j\phi_i} + \delta Z_{\phi_i\phi_j}\right) \\ - \frac{1}{2}\left(m^2_{\phi_k\phi_j}\delta Z_{\phi_k\phi_i} + m^2_{\phi_i\phi_k}\delta Z_{\phi_k\phi_j}\right) - \delta m^2_{\phi_i\phi_j} \ . \tag{17}$$

$\Sigma^{(1)}_{\phi_i\phi_j}$ denotes the unrenormalized self energies, extracted from the one-particle irreducible scalar two-point functions. The field-renormalization counterterms $\delta Z_{\phi_j\phi_i}$ and the mass counterterms $\delta m^2_{\phi_i\phi_j}$ are defined in a way to cancel all ultraviolet divergences appearing in $\Sigma^{(1)}_{\phi_i\phi_j}$. The finite pieces of the counterterms are defined by the chosen renormalization scheme. The field-renormalization constants are defined as $\overline{\text{DR}}$ parameters, such that they do not contain finite terms. The mass counterterms, on the other hand, are defined in a mixed OS-$\overline{\text{DR}}$ scheme. The gauge-boson masses $M_W$ and $M_Z$ and the tadpole coefficients are renormalized applying OS conditions, such that $\delta m^2_{\phi_i\phi_j}$ contains finite contributions from the corresponding counterterms [4].

Without going into too much detail, we summarize the numerical impact of the radiative corrections on the Higgs-boson masses of the $\mu\nu$SSM in the following. Schematically, a rough approximation of the SM-like Higgs-boson mass is given by

$$m^2_{h^{\text{SM}}} \sim M_Z^2\cos^2(2\beta) + \frac{1}{2}\lambda_i\lambda_i v^2\sin^2(2\beta) + \Delta^{\text{MSSM}}_{\text{(s)top}} + \Delta^{\widetilde{\nu}^{\mathcal{R}}_{iR}}_{\lambda_i^2} \ , \tag{18}$$

where the second term provides the enhancement of the tree-level contribution compared to the MSSM mentioned in Sect. 1. The third term consists of the usual MSSM-like corrections from the stop and the top sector (see Ref. [39] for a review). In the gauge basis, these terms are practically unchanged in the $\mu\nu$SSM. However, the mixing with the right-handed sneutrinos modifies how much of $\Delta^{\text{MSSM}}_{\text{(s)top}}$ is finally attributed to the mass eigenstate of the SM-like Higgs boson. The last term mainly arises from the mixing of the doublet fields with the right-handed sneutrinos. It was already observed in the next-to MSSM (NMSSM) that, in contrast to the tree-level term dependent on $\lambda_i$, the loop-corrections contained in $\Delta^{\widetilde{\nu}^{\mathcal{R}}_{iR}}_{\lambda_i^2}$ are usually negative and can, depending on the size of the mixing, the value of $\tan\beta$ and the self couplings $\kappa_{ijk}$, substantially decrease the prediction for the SM Higgs-boson mass [40]. Due to the presence of three gauge singlet scalars in the $\mu\nu$SSM instead of only one in the NMSSM, the analytic form of $\Delta^{\widetilde{\nu}^{\mathcal{R}}_{iR}}_{\lambda_i^2}$ is much more complicated. However, the numerical analysis of such corrections has shown that it is crucial to take into account independently the contributions from all three right-handed sneutrino for a precise prediction of the SM-like Higgs-boson mass [33].

The radiative corrections to the right-handed sneutrinos themselves are sizable only for small masses in the vicinity of 125 GeV or below [3, 4]. Otherwise, the tree-level mass is already a good estimate. This is due to the fact that the right-handed sneutrinos are gauge singlets and only couple to the SM particle content via a mixing with the Higgs doublet fields. If such mixing exists, the corresponding right-handed sneutrino acquires additional contributions to its mass from $\Delta^{\text{MSSM}}_{\text{(s)top}}$.

Finally, the most interesting radiative corrections are the ones obtained by the left-handed sneutrinos. They are caused by genuine effects of the $\mu\nu$SSM without a correspondence in the

(N)MSSM. It was shown that the dominant contributions arise from the counterterms of the tadpoles, which enter the mass counterterm in Eq. (17) with an inverse factor of the vev of the scalar field under consideration [3]. For the left-handed sneutrinos that means that they are enhanced by the small values of $v_{iL}$. This enhancement can compensate the suppression of factors of $Y^{\nu}_{ij}$ present in lepton-number violating couplings. Here, the corrections are mainly given by the tadpole diagrams with the stops in the loop. The stops are coupled to the left-handed sneutrinos via an $F$-term tree-level coupling between $\widetilde{t}_L$, $\widetilde{t}_R$, $\widetilde{\nu}_{iL}$ and $\widetilde{\nu}_{iR}$, after replacing $\widetilde{\nu}_{iR}$ with the corresponding vev $v_{iR}$. Expanding the complete renormalized self energy in powers of $A^u_3 = A_t$ and $1/v_{iL}$, one finds the very good approximation

$$\hat{\Sigma}^{(1)}_{\widetilde{\nu}^{\mathcal{R}}_{iL}\widetilde{\nu}^{\mathcal{R}}_{iL}} \approx \hat{\Sigma}^{(1)}_{\widetilde{\nu}^{\mathcal{R}}_{iL}\widetilde{\nu}^{\mathcal{R}}_{iL}}\Big|_{\frac{A_t}{v_{iL}}} = \frac{3}{16\pi^2}\frac{v_u v_{iR}}{\sqrt{2}v_{iL}} A_t Y_t^2 Y^{\nu}_{ii}\left(\frac{\log\big(\frac{m^2_{\widetilde{t}_1}}{\mu_R^2}\big)m^2_{\widetilde{t}_1} - \log\big(\frac{m^2_{\widetilde{t}_2}}{\mu_R^2}\big)m^2_{\widetilde{t}_2}}{m^2_{\widetilde{t}_1} - m^2_{\widetilde{t}_2}} - 1\right) , \tag{19}$$

where $m^2_{\widetilde{t}_1}$ and $m^2_{\widetilde{t}_2}$ are the squared stop masses and $\mu_R$ is the renormalization scale. These terms have to be added to the tree-level mass, which is approximately given by

$$\left(m^{(0)}_{\widetilde{\nu}^{\mathcal{R}}_{iL}\widetilde{\nu}^{\mathcal{R}}_{iL}}\right)^2 \approx \frac{Y^{\nu}_{ii} v_u v_{iR}}{\sqrt{2}v_{iL}}\left(-\frac{1}{\sqrt{2}}\kappa_{iii} v_{iR} - A^{\nu}_{ii}\right) . \tag{20}$$

This expression is subject to a renormalization-scale dependence induced by the scale dependence of the $\overline{\mathrm{DR}}$ parameters. The numerically most sizable contribution can be formulated approximately by the scale dependence of $A^{\nu}_{ii}$, whose dominant piece is given by

$$A^{\nu}_{ii}(\mu_R, \mu_0) \approx A^{\nu}_{ii}(\mu_0) + \frac{3}{16\pi^2} Y_t^2 A_t \log\frac{\mu_R^2}{\mu_0^2} , \tag{21}$$

that can be extracted from the $\overline{\mathrm{DR}}$ counterterm of $A^{\nu}_{ii}$ as given in Ref. [4], and where $\mu_0$ is the scale at which the value of $A^{\nu}_{ii}$ is given initially. Combining all this, we find that the one-loop mass is given by

$$\begin{aligned}\left(m^{(1)}_{\widetilde{\nu}^{\mathcal{R}}_{iL}\widetilde{\nu}^{\mathcal{R}}_{iL}}\right)^2 \approx \frac{Y^{\nu}_{ii} v_u v_{iR}}{\sqrt{2}v_{iL}}\Bigg( &-\frac{1}{\sqrt{2}}\kappa_{iii} v_{iR} - A^{\nu}_{ii}(\mu_0) \\ &- \frac{3}{16\pi^2} A_t Y_t^2\left(\frac{\log\big(\frac{m^2_{\widetilde{t}_1}}{\mu_0^2}\big)m^2_{\widetilde{t}_1} - \log\big(\frac{m^2_{\widetilde{t}_2}}{\mu_0^2}\big)m^2_{\widetilde{t}_2}}{m^2_{\widetilde{t}_1} - m^2_{\widetilde{t}_2}} - 1\right)\Bigg) .\end{aligned} \tag{22}$$

The logarithmic terms can be further simplified under the assumption that

$$m^2_{\widetilde{t}_1} - m^2_{\widetilde{t}_2} \ll M_S^2 \approx m^2_{\widetilde{t}_1} \approx m^2_{\widetilde{t}_2} , \tag{23}$$

with $M_S$ being the Susy-breaking scale, such that

$$\left(m^{(1)}_{\widetilde{\nu}^{\mathcal{R}}_{iL}\widetilde{\nu}^{\mathcal{R}}_{iL}}\right)^2 \approx \frac{Y^{\nu}_{ii} v_u v_{iR}}{\sqrt{2}v_{iL}}\left(-\frac{1}{\sqrt{2}}\kappa_{iii} v_{iR} - A^{\nu}_{ii}(\mu_0) - \frac{3}{16\pi^2} A_t Y_t^2 \log\big(\frac{M_S^2}{\mu_0^2}\big)\right) . \tag{24}$$

Note that the renormalization-scale dependence of the radiative corrections given in Eq. (19) drops out once the scale dependence of $A^{\nu}_{ii}$ is considered. Instead, the size of the corrections depends on the input scale of the $\overline{\mathrm{DR}}$ parameters $\mu_0$. The corrections vanish if $\mu_0$ is chosen to be close to the stop masses. Furthermore, it is convenient to choose the renormalization scale $\mu_R$ to be equal to the input scale $\mu_0$, so that the logarithmic term in Eq. (21) vanishes, and the tree-level expectation for the left-handed sneutrino mass given in Eq. (20) is unchanged. This is why in the code presented here the scales are fixed by default to be

$$\mu_0 = \mu_R = M_S \ , \tag{25}$$

such that

$$\left(m^{(1)}_{\widetilde{\nu}^{\mathcal{R}}_{iL}\widetilde{\nu}^{\mathcal{R}}_{iL}}\right)^2 \approx \left(m^{(0)}_{\widetilde{\nu}^{\mathcal{R}}_{iL}\widetilde{\nu}^{\mathcal{R}}_{iL}}\right)^2 \approx \frac{Y^{\nu}_{ii} v_u v_{iR}}{\sqrt{2} v_{iL}} \left( -\frac{1}{\sqrt{2}} \kappa_{iii} v_{iR} - A^{\nu}_{ii}(M_S) \right) \ . \tag{26}$$

Even though in principle any choice for the scales would be equally valid (within a physically reasonable range), the choice given above is highly recommended as long as the calculation of radiative corrections to the slepton masses has not been carried out. The reason is that large loop corrections to the left-handed sneutrinos could artificially change the mass ordering of the left-handed sneutrinos and sleptons, just because they are treated at different orders of perturbation theory, and therefore modify the phenomenology of a benchmark point completely. However, due to the different $D$-term contributions it is known that a left-handed sneutrino of a certain flavour cannot be heavier than the corresponding left-handed slepton, such that these artificial effects are unphysical and must be avoided.

# 3 The python package `munuSSM`

In this section we present the general structure of the code, which is also depicted in Fig. 1. The main package is called `munuSSM` and it contains the subpackages `crossSections`, `decays`, `effectiveCouplings`, `higgsBounds` and `standardModel`. Note that some of the modules are written in Fortran and compiled to python libraries using the compiler `f2py` from `NumPy`.

The usage of Fortran has several advantages. Firstly, numerical calculations are much faster in a statically typed language like Fortran. In addition, the numerical precision of floating-point numbers can be enhanced to quadruple precision in Fortran. In the context of the $\mu\nu$SSM, this turned out to be necessary due to the hierarchical structure of particle masses and mixing patterns. In particular, the seesaw mechanism leads to a mass matrix for the neutral fermions whose eigenvalues range from sub-eV to the TeV values, which is numerically challenging to diagonalize. Finally, the codes that are interfaced are all written in Fortran, such that it is much easier to use their libraries within a Fortran routine that is subsequently compiled to python. For the user of the package `munuSSM` the usage of Fortran is largely irrelevant. The only thing that is important is that the parameters of the model are not saved as usual python float objects and `NumPy` arrays, but as `numberQP` and `arrayQP` objects, that are defined in the module `dataObjects`. To obtain the values as floats or float arrays, the user just has to type `a.float` in case of `a` being an instance of `numberQP` or `arrayQP`.

- munuSSM
  - benchmarkPoint
  - benchmarkPointFromFile
  - constants
  - dataObjects
  - CalcDepParas [F]
  - CalcLoopMasses [F]
  - FHgetMTMB [F]
  - FHselfenergies [F]
  - OneLoopcntrs [F]
  - RGEs [F]
  - SelfEnergiesAA [F]
  - SelfEnergies [F]
  - TLcpls [F]
  - TLspec [F]
  - TLTPsolver [F]
  - crossSections
    - sleptons
  - decays
    - pseudoscalars
    - scalars
    - sleptons
    - twobody
    - util
  - effectiveCouplings
    - particles
    - pseudoscalars
    - scalars
    - sleptons
    - util
  - higgsBounds
    - HBHSmixed [F]
    - HBmixed [F]
    - HSSMhadr [F]
    - util
    - wrappers
  - standardModel
    - higgs
    - util
      - alphaS

Figure 1: General structure of the code. Packages are indicated with the squared boxes, and the vertical lines indicate to which package each module belongs. Modules written in Fortran are marked with an [F].

The user interface is defined in the class `BenchmarkPointFromFile`. This class inherits the methods of the class `BenchmarkPoint` to construct and analyse a benchmark point. In addition, it reads the input parameters from a file during the initialization. Within this class, the subpackes are utilized to calculate the branching ratios and cross sections of the scalar particles, while the modules of the main package `munuSSM` perform the calculation of the radiatively corrected particle spectrum. Before explaining the methods of the `BenchmarkPoint` class and how the user can call them, we briefly explain the role of each subpackage and give some details on the implementation.

- `crossSections` contains the calculation of cross sections of particles at the LHC or any other future collider. So far the only cross section implemented is the charged Higgs-boson production in the $pp \to H^{\pm}tb$ channel, which is however only relevant for the charged scalar of the $\mu\nu$SSM corresponding to the charged Higgs boson of the MSSM. For the remaining sleptons, the couplings to quarks are suppressed by the smallness of lepton-number violation, as is their mixing with the MSSM-like charged Higgs boson. The above mentioned cross section is implemented in the form of a spline interpolation as a function of $\tan\beta$ and the charged Higgs-boson mass in the 2HDM limit [20], therefore lacking subdominant SUSY-QCD corrections. For the charged scalars corresponding to the sleptons, the main production channel is the production in pairs, which is currently not yet implemented.[3] The cross sections of the neutral scalars are obtained via the interface to `HiggsBounds`, based on the effective couplings calculated in the supackage `effectiveCouplings` (see below).

- `decays` calculates the decay widths and branching ratios of all neutral and charged Higgs bosons. The decay widths of decays into SM particles are implemented via a rescaling of the SM prediction for a Higgs boson of the same mass, again utilizing the effective couplings calculated in the `effectiveCouplings` package. The SM predictions are implemented in the form of cubic spline interpolations of data tables published in Refs. [19, 20], based on the results obtained with the codes `HDECAY` [41–43] and `PROPHECY4F` [44, 45]. The decays into BSM particle final states are considered at leading order, however using the Higgs-boson couplings rotated into the radiatively corrected mass eigenstate basis, therefore taking into account the propagator corrections calculated in the main package `munuSSM`. The implementation of these decays follows the general approach of Ref. [46]. Using the couplings in the loop-corrected basis corresponds to taking into account the finite wave-function renormalization factors (or $Z$-factors) in the limit of vanishing momentum [14, 46]. For the accurate prediction of the Higgs-boson masses, it is recommended to include the momentum dependence of the radiative corrections. Strictly speaking, this leads to the fact that the mixing matrices will not be unitary anymore. Fortunately, these effects are numerically negligible except for extreme cases. So far, the only three-body decays considered are the decays into off-shell vector bosons, whose corresponding decay widths are included in the SM prediction for the decays into a pair of massive gauge bosons.

[3] This is partially due to the fact that the pair-production cross sections of charged Higgs bosons is currently unused within `HiggsBounds`, even though it can be given as input [26].

**BenchmarkPoint**

| | |
|---|---|
| `calc_tree_level_spectrum(self)` | Calculates the particle spectrum at tree level. |
| `calc_tree_level_couplings(self)` | Calculates the complete set of couplings at tree level. |
| `calc_one_loop_counterterms(self)` | Calculates the counterterms used in the renormalized one-loop self energies. |
| `calc_one_loop_self_energies(self, even, odd, p2_Re, p2_Im)` | Calculates the values of the renormalized one-loop self energies for the CP-even scalars if `even=1` and for the CP-odd scalars if `odd=1` for a given momentum $p$, where `p2_Re` is the real part and `p2_Im` is the imaginary part of $p^2$. |
| `calc_two_loop_self_energies(self, p2_Re, p2_Im)` | Calculates the values of the renormalized self energies with the full one-loop and partial higher-order corrections for the CP-even scalars for a given momentum $p$, where `p2_Re` is the real part and `p2_Im` is the imaginary part of $p^2$. |
| `calc_loop_masses(self, even=2, odd=1, accu=1.e-5, momentum_mode=1)` | Calculates the loop corrected scalar masses. `even` is the loop level considered for the CP-even scalars and possible values are 0, 1 and 2 for tree level, one-loop level or one-loop level supplemented by higher-order corrections. `odd` is the loop level considered for the CP-odd scalars and possible values are 0 and 1 for tree level or one-loop level. `accu` is the numerical precision of the iterative procedure of the matrix diagonalization if `momentum_mode=1`, in which case the momentum dependence of the self energies is taking into account. For `momentum_mode=0` the momentum is set to zero. |
| `calc_effective_couplings(self)` | Calculates the effective couplings of the CP-even and the CP-odd scalars |
| `calc_branching_ratios(self)` | Calculates the decay widths and branching ratios of the CP-even, the CP-odd and the charged scalars. |

Table 1: Class methods of the class `BenchmarkPoint` as defined in the module `benchmarkPoint`.

- effectiveCouplings calculates the effective couplings of the neutral scalars, defined as the coupling strength normalized to the one of a hypothetical SM Higgs boson having the same mass. The precise definition of these coefficients can be found in Ref. [26]. Loop-induced couplings, as the ones to photons or gluons, are calculated using the general expressions for the form factors as can be found in Ref. [47]. Resummed higher-order corrections proportional to $\tan\beta$ are implemented for couplings to the third generation of down-typ fermions in terms of the quantities $\Delta_b$ and $\Delta_\tau$ following Ref. [47]. As already mentioned before, the effective couplings are used to calculate decays into SM particles. Apart from that, they are given as input to the code HiggsBounds, which uses them to calculate the production cross sections at LEP, Tevatron and the LHC.[4]

- higgsBounds constructs the input arrays for the interface to HiggsBounds and HiggsSignals. In addition, it provides a wrapper class to directly call both codes from within python.[5] Since we interact with both external codes via their Fortran libraries, we can save additional results beyond the usual output, such as cross sections. Via the higgsBounds subpackage, a given set of benchmark points of the $\mu\nu$SSM can easily be tested against constraints from collider searches and the signal rates of the SM-like Higgs boson.

- standardModel contains the data tables of the SM predictions for decay widths of the Higgs boson as given in Refs. [19, 20]. The data is given for different mass intervals. The subpackage constructs spline interpolations of the data and provides functions taking the Higgs-boson mass as input to extract the data. The maximum value for the particle mass of the data tables is at around 1 TeV. If within the decay subpackage larger masses appear, the values are extrapolated based on the known leading mass dependence [47].

Having explained the role of the subpackages, we now turn to the main package munuSSM. Therein, basically all calculations are performed within Fortran modules. During the initialization of a benchmark point, the Fortran modules CalcDepParas and TLTPsolver set up the complete set of model parameters. The latter solves the tadpole equations for the diagnoal soft mass parameters given the vevs as input. The module FHgetMTMB calls FeynHiggs to extract the top-quark and bottom-quark masses used in the scalar self energies (see Sect. 3.1.1 for details). The tree-level spectrum and the couplings are calculated in the modules TLspec and TLcpls. These are then used for the calculation of the renormalized self energies. They are implemented in the form as shown in Eq. (17). The counterterms are independent of the momentum, so that they are calculated only once in the module OneLoopcntrs. Once they are available, the one-loop part of the self energies of the CP-even and the CP-odd scalars are calculated in the modules SelfEnergies and SelfEnergiesAA. The contributions beyond

[4] In the traditional effective-coupling input of HiggsBounds, the effective couplings are also used internally to calculate branching ratios. In our interface, we use a mixed input in which the branching ratios are given as additional input as calculated in the subpackage decays. Effectively, this corresponds to the LHA input format of HiggsBounds.

[5] A stand-alone python wrapper for HiggsBounds can be found under https://gitlab.com/thomas.biekoetter/higgsbounds_python_wrapper.

one-loop level are extracted from `FeynHiggs` in the module `FHselfenergies`. Finally, the loop-corrected scalar spectrum is calculated in the module `CalcLoopMasses` by finding the zeros of the determinant of the inverse propagator matrix shown in Eq. (14).

The user interface is defined via the methods of the python class `BenchmarkPoint`, such that the Fortran modules described before do not have to be called directly by the user. The complete list of public routines is listed in Tab. 1. Note that an instance of this class should be created via its subclass `BenchmarkPointFromFile`, which contains additional routines to read the parameter values from an input file. Because of potentially large corrections to the masses of the left-handed sneutrinos (see Sect. 2.1), the renormalization scale $\mu_R$ at which the radiative corrections are evaluated is set to be equal to the SUSY-breaking scale $M_S$ at which the $\overline{\mathrm{DR}}$ SUSY parameters are defined. While these are given as input by the user, the SM parameters are set to default values in the module `constants`. In this module, also the value for $M_S$ is fixed to 1 TeV by default. This value should only be changed by the user if the stop masses are much heavier than 1 TeV. Note, however, that in such a situation the Feynman-diagrammatic fixed-order calculation applied in this code is not the most accurate one and a hybrid approach (as is implemented in `FeynHiggs`) incorporating effective field theory calculations is required.

For a phenomenological study of a benchmark point, the most interesting routines for the user are `calc_loop_masses` to obtain a precise prediction for the particle spectrum and `calc_branching_ratios` to obtain the branching ratios of the scalars. The remaining functions can be called directly by the user, but will usually be called only internally, as they provide the required quantities for the above mentioned functions. For instance, if the user calls

```
pt.calc_loop_masses(2, 1, momentum_mode=1)
```

with `pt` being an instance of `BenchmarkPointFromFile`, it is internally checked if the tree-level masses and couplings are already available. If they are not, the functions `calc_tree_level_spectrum` and `calc_tree_level_couplings` are automatically called before calculating the radiative corrections. With `momentum_mode=1` we choose to take into account the momentum dependence of the radiative corrections. `momentum_mode=0` selects the limit of vanishing external momentum. This options is less precise but faster, because the inverse propagator matrix has to be diagonalized only once, while an iterative procedure is applied for `momentum_mode=1`. In the same manner, if

```
pt.calc_branching_ratios()
```

is called, the effective couplings are required for the rescaling of the SM predictions, such that internally `calc_effective_couplings` is called if it has not already been called before. In App. B we state the exact form of the return values and class attributes set by each function shown in Tab. 1. Basic user instructions are given in Sect. 3.3. Before that we provide some details on the interfaces to the other public codes.

## 3.1 Interfaces

The package `munuSSM` makes use of other public codes for some of the model predictions. This codes are downloaded and installed automatically during the installation of the main package (see Sect. 3.2). The interfaces utilize the Fortran libraries of the codes. In the following we briefly describe the information provided by the codes and how they are called internally.

### 3.1.1 FeynHiggs

For the accurate prediction of the SM-like Higgs-boson mass, a pure one-loop calculation is not sufficient. Fortunately, the dominant higher-order corrections can be taken over from the MSSM. However, one has to take care of a consistent combination of the one-loop corrections calculated in the full $\mu\nu$SSM and the higher-order corrections known from the MSSM. This is why in Refs. [3, 4] the renormalization prescription of the one-loop calculation in the $\mu\nu$SSM was closely based on the one implemented in the public MSSM code `FeynHiggs`, such that the higher-order corrections could be supplement from there.

The radiative corrections to the scalar masses and mixings are given by the renormalized self energies $\hat{\Sigma}(p^2)$ that enter the inverse propagator matrix as shown in Eq. (14). Schematically, the self energies of the CP-even Higgs bosons are implemented as

$$\hat{\Sigma} = \hat{\Sigma}^{(1)}_{\mu\nu\mathrm{SSM}} - \hat{\Sigma}^{(1)}_{\mathrm{FeynHiggs}} + \hat{\Sigma}^{(1)+(2')+\mathrm{resum.}}_{\mathrm{FeynHiggs}} \ . \tag{27}$$

The piece $\hat{\Sigma}^{(1)}_{\mu\nu\mathrm{SSM}}$ is the full one-loop result including all couplings of the $\mu\nu$SSM, and renormalized according to Eq. (17). The numerical evaluation of the loop functions appearing in $\hat{\Sigma}^{(1)}_{\mu\nu\mathrm{SSM}}$ is achieved via a link to the public code `LoopTools` [38]. Imaginary parts of the loop momentum $p^2$ are considered via a Taylor expansion with respect to $\mathrm{Im}(p^2)$ up to first order. To this piece we add the full `FeynHiggs v.2.16.1` result including the approximate two-loop contributions and the contributions obtained from the resummation of logarithmic contributions denoted by the term $\hat{\Sigma}^{(1)+(2')+\mathrm{resum.}}_{\mathrm{FeynHiggs}}$. Since this piece also contains the MSSM one-loop result, these terms have to be subtracted again to avoid a double counting. This is done by calling `FeynHiggs` a second time with the flag `looplevel` set to 1, yielding $\hat{\Sigma}^{(1)}_{\mathrm{FeynHiggs}}$, which is then subtracted from the sum.

For this procedure to be consistent, it is crucial that the one-loop piece of the $\mu\nu$SSM is calculated using the same set of parameters as is used in `FeynHiggs`. In particular, this concerns the values of the top-quark mass and the bottom-quark mass, from which the corresponding Yukawa couplings $Y_t$ and $Y_b$ are derived.[6] This is achieved by a slightly modified version of the `FeynHiggs` routine `FHGetPara`, which is called during the initialization of an instance of `BenchmarkPointFromFile`. For the top quark, the pole mass $M_t$ is given as input and `FHGetPara` returns the $\overline{\mathrm{MS}}$ value of the top-quark mass at the scale $M_t$ in the SM at NNLO $\overline{m}_t^{\overline{\mathrm{MS}},\mathrm{SM}}(M_t)$, which is used in `FeynHiggs` for the calculation of $\hat{\Sigma}^{(1)+(2')+\mathrm{resum.}}_{\mathrm{FeynHiggs}}$. In principle, the value is different when the log resummation is switched off with `loglevel=0`, such that the value of $\overline{m}_t^{\overline{\mathrm{MS}},\mathrm{SM}}(M_t)$ would be different in $\hat{\Sigma}^{(1)}_{\mathrm{FeynHiggs}}$, yielding a mismatch compared to $\hat{\Sigma}^{(1)}_{\mu\nu\mathrm{SSM}}$. To avoid that, we set by hand the flag `loglevelmt=3` in the `FeynHiggs` routine `FHSetFlags`, so that the same value of $\overline{m}_t^{\overline{\mathrm{MS}},\mathrm{SM}}(M_t)$ is used independently of the flag `loglevel`.

[6]Note that the strong QCD coupling constant $\alpha_s$ does not enter at one-loop level.

In a similar way, we obtain $\overline{m}_b^{\overline{\rm DR},\rm MSSM}(M_S)$, i.e., the MSSM $\overline{\rm DR}$-renormalized value of the bottom-quark mass at the scale $M_S$, in the modified routine `FHGetPara`. We extract the value used by `FeynHiggs` when called with `looplevel=1`. In contrast to $\overline{m}_t^{\overline{\rm MS},\rm SM}(M_t)$, which is given by SM RGEs, the precise value of $\overline{m}_b^{\overline{\rm DR},\rm MSSM}(M_S)$ depends also on the Susy parameters, mainly via the so-called $\Delta_b$-corrections. Apart from that, it is different when called with `looplevel=2`. However, for the prescription in Eq. (27) to be consistent, this is not a problem as long as we assure that the value of the quark masses in $\hat{\Sigma}^{(1)}_{\mu\nu\rm SSM}$ and $\hat{\Sigma}^{(1)}_{\rm FeynHiggs}$ are identical.

For the remaining MSSM one-loop contributions, arising from loop diagrams with particles inserted in the loop that are not (s)tops or (s)bottoms, the double-counting is automatically avoided due to the cancellation between $\hat{\Sigma}^{(1)}_{\rm FeynHiggs}$ and $\hat{\Sigma}^{(1)+(2')+\rm resum.}_{\rm FeynHiggs}$, because they do not depend on the flags `looplevel` or `loglevel`. Thus, only the one-loop result in the full model contained in $\hat{\Sigma}^{(1)}_{\mu\nu\rm SSM}$ contributes for these sectors. This is important because they might be substantially modified compared to the MSSM. For instance, due to the presence of the portal couplings $\lambda_i$, the tree-level masses of the doublet-like Higgs bosons receive additional contributions, so that loop diagrams with Higgs bosons in the loop have to be accounted for in $\hat{\Sigma}^{(1)}_{\mu\nu\rm SSM}$, while the corresponding diagrams from the MSSM should drop out.

Our approach using `FeynHiggs` does not capture the modifications of the tree-level Higgs sector of the $\mu\nu$SSM compared to the MSSM proportional to $\lambda_i$ within the contributions beyond one-loop level. They would enter in the approximate two-loop result via the fixed-order terms of $\mathcal{O}(\alpha_t^2, \alpha_b^2, \alpha_b\alpha_t)$, in which the Higgs bosons appear as internal particles in the corresponding loop diagrams. However, this is a subleading effect as long as the corrections to the doublet fields are dominant. Also, it is the best possible approximation while the calculation of the two-loop contributions in the full model is not carried out. Nevertheless, for small values of $\tan\beta$ and large values of $\lambda_i$ this leads to a potential source of theory uncertainty for the prediction of the SM-like Higgs-boson mass. In comparison to neglecting the contributions beyond one-loop level entirely, our numerical results of Refs. [3, 4] showed that even in these cases the prediction for the Higgs-boson mass improves when taking the approximate MSSM contributions into account. The same conclusion was drawn in other analyses using `FeynHiggs` for similar extensions of the MSSM [48–50].

Once the renormalized self energies are constructed, the inverse propagator matrix is diagonalized using the public Fortran library `Diag` [51]. If the momentum dependence is taken into account, the loop-corrected pole masses are given by the zeros of the determinant of the inverse propagator matrix, which are calculated by an iterative procedure.

#### 3.1.2 HiggsBounds

To test a set of benchmark points against collider constraints from searches for BSM scalars, an interface to the public code `HiggsBounds` is implemented. With `pts` being a single instance or a list of instances of the class `BenchmarkPoint`, the user can call the function

```
check_higgsbounds(pts)
```

defined in the module `util` of the subpackage `higgsBounds`. This function first calls the method `_setup_higgsbounds` for each instance of `BenchmarkPoint` in `pts`, which will subsequently call `calc_effective_couplings` and `calc_branching_ratios` (see Tab. 1) in case

**self.HiggsBounds**

| | | |
|---|---|---|
| `result` | `(23, )` | The first element is the global `HiggsBounds` result with `1=allowed` and `0=forbidden`. The following 22 elements are the results for each neutral CP-even and CP-odd scalar and each charged scalar, in this order with ascending masses. |
| `chan` | `(23, )` | As before, but each element gives the channel number of the most sensitive search. The experimental search corresponding to the channel number can be found in the file `Key.dat` that is produced automatically by `HiggsBounds`. |
| `obsratio` | `(23, )` | As before, but each element gives the ratio of predicted and observed channel rate. |
| `ncombined` | `(23, )` | As before, but each element gives the number of scalars whose signal rates were combined and assumed to be contributing to the search channel. |
| `XSsingleH` | `(15, 4)` | Hadronic inclusive single-Higgs production cross section at the Tevatron with 2 TeV and the LHC with 7, 8 and 13 TeV center-of-mass energy for each CP-even and CP-odd scalar, normalized to the SM prediction. |
| `XSggH` | `(15, 4)` | As before, but for the gluon fusion process. |
| `XSbbH` | `(15, 4)` | As before, but for the $b\bar{b}$ associated process. |
| `XSVBF` | `(15, 4)` | As before, but for the vector boson fusion process. |
| `XSWH` | `(15, 4)` | As before, but for the production in association with a $W$ boson. |
| `XSZH` | `(15, 4)` | As before, but for the production in association with a $Z$ boson. |
| `XSttH` | `(15, 4)` | As before, but for the $t\bar{t}$ associated process. |
| `XStH_tchan` | `(15, 4)` | As before, but for single $t$ associated production through $t$-channel exchange. |
| `XStH_schan` | `(15, 4)` | As before, but for single $t$ associated production through $s$-channel exchange. |
| `XSqqZH` | `(15, 4)` | As before, but for quark-initiated production in association with a $Z$ boson. |
| `XSggZH` | `(15, 4)` | As before, but for gluon-initiated production in association with a $Z$ boson. |

Table 2: Form of the dictionary `HiggsBounds` containing the results of the `HiggsBounds` routine as set by the function `check_higgsbounds`. The first column lists the keys of the dictionary. The items of each key are `NumPy` arrays with the shape given in the second column. The third column explains the meaning of each entry.

they have not been called before. Based on the effective couplings and the branching ratios, `check_higgsbounds` will then construct the input for `HiggsBounds` for the whole set of points. Finally, the `HiggsBounds` library is accessed via the Fortran module `HBmixed`. This module is called within the wrapper class `Mixed` defined in the subpackage `higgsBounds`.

The results are saved as dictionaries which are set as class attributes to each benchmark point contained in `pts`. If `pt` is an instance of `BenchmarkPoint`, the results are saved in:

```
pt.HiggsBounds
```

This dictionary has the elements listed in Tab. 2. The meaning of each entry of the dictionary corresponds to the original definitions within `HiggsBounds` [26]. The user can check if the benchmark point is excluded depending on the value:

```
pt.HiggsBounds['result'][0]
```

It is 1 if the point is allowed and 0 if any of the scalars is excluded. With the remaining elements of this array, the user can verify which of the scalars are excluded. The experimental search responsible for the exclusion can be obtained by comparing the channel number saved under the key `chan` with the list of applied experimental searches saved in the file `Key.dat` that `HiggsBounds` creates automatically. The cross sections for the neutral scalars that are calculated by `HiggsBounds` rely on the effective couplings calculated before.

### 3.1.3 HiggsSignals

In addition to the test against cross-section limits using `HiggsBounds`, it is possible to verify whether a benchmark point contains a Higgs boson at $\sim$ 125 GeV that correctly accommodates the measured signal rates of the SM-like Higgs boson. This is done via an interface to the public code `HiggsSignals`. Since `HiggsSignals` relies on the `HiggsBounds` subroutines to read the theoretical input, it is reasonable to combine both tests into a single function call. We provide the function

```
check_higgsbounds_higgssignals(pts)
```

defined in the module `util` of the subpackage `higgsBounds`. As before, `pts` can be a single instance or a list of instances of the class `BenchmarkPoint`. Executing the above command will call both `HiggsBounds` and `HiggsSignals` via the Fortran module `HBHSmixed`. For a better interpretation of the $\chi^2$ test performed by `HiggsSignals`, `HiggsSignals` is called a second time via the Fortran module `HSSMhadr`, providing a reference $\chi^2_{\rm SM}$ value based on the SM predictions using the same set of experimental measurements.

The complete result of the function `check_higgsbounds_higgssignals` is saved as dictionaries in the class attributes:

```
pt.HiggsBounds
pt.HiggsSignals
```

As already mentioned before, `pt` is an instance of the class `BenchmarkPoint` contained in `pts`. The dictionary `pt.HiggsBounds` was already introduced in Sect. 3.1.2 (see also Tab. 2). The dictionary `pt.HiggsSignals` contains the `HiggsSignals` results. The whole list of entries is given in Tab. 3. For the interpretation of the fit, the most valuable information is provided by the global $\chi^2$ value contained in

**self.HiggsSignals**

| | |
|---|---|
| `Chisq_mu` | The $\chi^2_\mu$ result regarding the signal-rate measurements. |
| `Chisq_mh` | The $\chi^2_{m_h}$ result regarding the mass measurements. |
| `Chisq` | The total $\chi^2$ result, i.e., $\chi^2 = \chi^2_\mu + \chi^2_{m_h}$. |
| `nobs` | The total number of observables considered in the $\chi^2$ test. |
| `Pvalue` | The $p$ value derived from the $\chi^2$ result assuming one free parameter. |
| `Delta_Chisq_mu` | The difference $\chi^2_\mu - \chi^2_{\mu,\mathrm{SM}}$, where $\chi^2_{\mu,\mathrm{SM}}$ is the Standard Model reference $\chi^2$ evaluated using the same set of signal-rate measurements. |
| `Delta_Chisq_mh` | As before, but for the difference $\chi^2_{m_h} - \chi^2_{m_h,\mathrm{SM}}$ using the same set of mass measurements. |
| `Delta_Chisq` | As before, but for the difference $\chi^2 - \chi^2_{\mathrm{SM}}$, where $\chi^2_{\mathrm{SM}} = \chi^2_{\mu,\mathrm{SM}} + \chi^2_{m_h,\mathrm{SM}}$. |

Table 3: Form of the dictionary `HiggsSignals` containing the results of the `HiggsSignals` routine as set by the function `check_higgsbounds_higgssignals`. The first column lists the keys of the dictionary. The second column explains the meaning of each entry.

```
pt.HiggsSignals['Chisq']
```

and the difference of this value to the SM reference value contained in:

```
pt.HiggsSignals['Delta_Chisq']
```

We leave it to the user to decide which values are considered to represent an accurate fit to the experimental data. For more information about the interpretation of the `HiggsSignals` results we refer to Ref. [29]. We recommend to define a criteria based on the difference between the $\chi^2$ value and the SM reference value $\chi^2_{\mathrm{SM}}$, instead of only taking into account the $\chi^2$ value of the benchmark point alone.

## 3.2 Installation

To install the package `munuSSM` you need the version control system `git`, working compilers for Fortran, c and c++ (recommended `gfortran` and `gcc`), and `cmake` for the installation of `HiggsBounds` and `HiggsSignals`. All of this is already installed on a regular unix machine. You can clone the repository with `SSH` by typing:

```
git clone git@gitlab.com:thomas.biekoetter/munussm.git
```

Alternatively, you can clone the repository with `HTTPS` by typing:

```
git clone https://gitlab.com/thomas.biekoetter/munussm.git
```

Then the package can be installed by entering the directory and executing the makefile:

```
cd munussm
make all
```

You can specify the python version used for the installation by typing, for instance:

```
make all PC=python3.6
```

We stress that python version 2 is not supported. Furthermore, if you wish to specify the `gnu` compiler versions, you can type, for instance:

```
make all FC=gfrotran-10 CC=gcc-10 CXX=g++-10
```

During the installation process, the external libraries `Diag`, `LoopTools`, `FeynHiggs`, `HiggsBounds` and `HiggsSignals` are installed in the directory `external`. Once the installation process terminated, the package is installed in your python environment and can be imported with:

```
import munuSSM
```

## 3.3 Usage

Only basic knowledge of the python programming language is required to use the package `munuSSM`. So far, the only possibility to create an instance of the class `BenchmarkPoint` is via the subclass `BenchmarkPointFromFile`. A benchmark point is initialized by doing:

```
from munuSSM.benchmarkPointFromFile import BenchmarkPointFromFile
pt = BenchmarkPointFromFile(file=FILENAME)
```

Here, `FILENAME` is the path to the input file containing the values of the free parameters. The format of the input file is depicted in Listing (1) in App. A. Example input files can also be found in the folder `example`. In the input files it is important that the order of the lines remains unchanged and that the parameter values start with the first character of each line. Every character beyond the # sign is treated as a comment. As already explained in Sect. 3, the SUSY parameters are $\overline{\mathrm{DR}}$ parameters assumed to be given at the `Susy`-breaking scale $M_S$, which is by default set to 1 TeV in the module `constants`.

Once the benchmark point `pt` is initialized, the methods defined in Tab. 1 can be called. For example, the tree-level spectrum and the complete set of couplings can be obtained with:

```
pt.calc_tree_level_spectrum()
pt.calc_tree_level_couplings()
```

Strictly speaking, only the second line would have been sufficient, since the couplings need the mixing matrices as input, which are calculated when calling `calc_tree_level_spectrum`. Therefore, this function is called automatically when `calc_tree_level_couplings` is called in case the mixing matrices are not yet available. We can obtain the loop-corrected scalar masses by typing:

```
pt.calc_loop_masses(
    even=2,
    odd=1,
    momentum_mode=1)
```

Here, we explicitly set the loop order for the neutral CP-even scalars to 2 and for the CP-odd scalars to 1. The loop order `even=2` includes also the contributions from the resummation of logarithmic terms (see Sect. 2.1). In addition, we choose to take into account the momentum dependence of the radiative corrections by setting `momentum_mode=1`, which is the recommended value. The values of the arguments shown above correspond to the default values of the arguments, such that in this case it would have been sufficient to call:

```
pt.calc_loop_masses()
```

The loop corrected scalar masses are saved in the class attributes `pt.Masshh_2L` and `pt.MassAh_L`. The latter also contains the mass of the unphysical Goldstone boson with a mass of $\sim M_Z$.

The branching ratios of the neutral and charged Higgs bosons can be obtained by calling:

```
pt.calc_branching_ratios()
```

This will save the various branching ratios of the neutral CP-even and CP-odd scalars and the charged scalars in the objects:

```
pt.BranchingRatiosh
pt.BranchingRatiosA
pt.BranchingRatiosX
```

The corresponding decay widths and also the total decay widths are stored in the objects:

```
pt.Gammash
pt.GammasA
pt.GammasX
```

These objects are lists of dictionaries for each scalar particle. In the dictionaries, the different final states are labeled by the keys, and the value corresponding to each key is a `NumPy` array in which each index corresponds to a family index of the final state particles (see Tab. 12 in App. B for the definition of each entry). For instance, the branching ratio for the decay $h_8 \to h_1\ h_2$ is saved in `pt.BranchingRatiosh[7]['hhh'][0,1]`.[7] It is important to note that the same decay with the family indices in the final state switched, i.e., $h_8 \to h_2\ h_1$, is saved separately in `pt.BranchingRatiosh[7]['hhh'][1,0]`, so that the full branching ratio for the decay into this final state is given by the sum. The reason for this definition is that this allows to calculate the total decay widths by simply summing over all elements of each array contained in the dictionary corresponding to each particle. The program calculates the branching ratios using the neutral scalar masses and mixing matrices at the highest loop level available. It will warn the user during the calculation if only the tree-level spectrum is used. To avoid these warnings, the user should call `calc_loop_masses` before calling `calc_branching_ratios`.

Finally, the collider constraints can be checked by calling `HiggsBounds` and `HiggsSignals`:

```
from munuSSM.higgsBounds.util import \
check_higgsbounds_higgssignals

check_higgsbounds_higgssignals(pt)
```

[7]Note that indices in python start with 0, so that the index 7 selects the particle $h_8$ etc.

One restriction is that the HiggsBounds libraries can only be called once within a python session. If one wants to check several benchmark points, one has to initialize them first, save them in a list, and call the function with this list as argument:

```
pt_1 = BenchmarkPointFromFile(file=FILENAME_1)
pt_2 = BenchmarkPointFromFile(file=FILENAME_2)
...
pt_N = BenchmarkPointFromFile(file=FILENAME_N)

pts = [pt1, pt2, ..., ptN]

check_higgsbounds_higgssignals(pts)
```

To only obtain the HiggsBounds result, one can call:

```
from munuSSM.higgsBounds.util import check_higgsbounds

check_higgsbounds(pts)
```

### 3.3.1 Example

An example script can be found in the file example.py in the folder example. The source code of the script is:

```
from munuSSM.benchmarkPointFromFile import BenchmarkPointFromFile
from munuSSM.higgsBounds.util import check_higgsbounds_higgssignals


FILENAME = "paras.in"

print("Reading parameters from file " + FILENAME)
pt = BenchmarkPointFromFile(file=FILENAME)

print("Calculating loop corrections...")
pt.calc_loop_masses(momentum_mode=1)

print(
    "Checking against collider constraints using " +
    "HiggsBounds and HiggsSignals...")
check_higgsbounds_higgssignals(pt)

print("The masses of the scalars are:")
i = 1
for m in pt.Masshh_2L:
    print("  Mh" + str(i) + " = " + str(m.float))
    i += 1

print("The masses of the pseudoscalars are:")
i = 1
for m in pt.MassAh_L:
    print("  MA" + str(i) + " = " + str(m.float))
    i += 1
```

```
print("The masses of the charged scalars are:")
i = 1
for m in pt.MassHpm:
    print("  MHpm" + str(i) + " = " + str(m.float))
    i += 1

if pt.HiggsBounds['result'][0] == 0:
    print("  The point is excluded.")
    i = 1
    for r in pt.HiggsBounds['result'][1:]:
        if r == 0:
            if i <= 8:
                print("    Excluded particle: H" + str(i))
            elif i <= 15:
                print("    Excluded particle: A" + str(i - 8))
            else:
                print("    Excluded particle: Hpm" + str(i - 15))
            print(
                "    r_obs = " +
                str(pt.HiggsBounds['obsratio'][i]))
            print(
                "    Number of search channel: " +
                str(pt.HiggsBounds['chan'][i]) +
                " (see Key.dat for more info)")
        i += 1
else:
    print("  The point is allowed.")

ChisqHS = pt.HiggsSignals['Delta_Chisq']
if ChisqHS > 6.18:
    print("  No scalar with SM signal rates at ~ 125 GeV.")
else:
    print("  A SM-like Higgs boson is present.")
```

In this script the benchmark point is initialized with the parameter values defined in the input file `paras.in` situated in the `example` folder. Afterwards, the loop-corrected Higgs-boson masses are calculated. Finally, `HiggsBounds` and `HiggsSignals` are called and the results are printed. For each excluded Higgs boson, the script prints the channel number of the responsible experimental search and the value `obsratio` as calculated by `HiggsBounds`. In the last step, the script decides whether a SM-like Higgs boson at $\sim 125$ GeV is present based on the value `Delta_Chisq` as calculated by `HiggsSignals`.

# 4 Conclusion and outlook

In this paper we present the public code `munuSSM`: A flexible python package for the phenomenological analysis of the $\mu$-from-$\nu$ Supersymmetric Standard Model. The code incorporates a calculation of the radiatively corrected Higgs-boson masses. The precision of the prediction for the SM-like Higgs-boson mass is at a comparable level to the ones of spectrum generators for the MSSM. This is achieved by a full one-loop renormalization of the Higgs potential and consistently supplementing higher-order corrections known from the MSSM via an interface to the public code FeynHiggs. For obvious reasons, this approach does not capture effects beyond one-loop level genuine to the $\mu\nu$SSM. For the SM-like Higgs-boson mass, these contributions are expected to be substantially smaller than the MSSM-like contributions considered here for phenomenologically viable points. Nevertheless, for an estimate of the theory uncertainty this fact should be kept in mind.

In addition, the package munuSSM provides a calculation of effective couplings and branching ratios of the abundant scalars, pseudoscalars and sleptons of the model. Based on these results, a set of benchmark points can easily be checked against collider constraints from the Tevatron, LEP and the LHC via a user-friendly interface to the public code HiggsBounds. Furthermore, the presence in the spectrum of a Higgs boson reproducing the measured signal rates of the SM Higgs boson at $\sim 125$ GeV can be verified via an interface to the public code HiggsSignals. Since both codes are accessed via their Fortran libraries, they can be utilized to extract other useful quantities which would not be directly accessible via the simpler command-line or SLHA-file input methods. For instance, we obtain the LHC cross sections for the neutral scalars as they are derived within HiggsBounds from the effective couplings. For a better interpretation of the HiggsSignals results, we provide a SM reference $\chi^2$ that can be taken into account when deciding whether a benchmark point is excluded or not.

The package munuSSM is a suitable framework for the implementation of further calculations and predictions related to the $\mu\nu$SSM. The modular structure of the code permits its extension without having to know the details of the already available features. In many cases, basic ingredients for the implementation of new features, such as the couplings and the mixing matrices, are already available, providing a starting point for the exploration of other sectors of the model.

# Acknowledgements

I thank S. Heinemeyer and C. Muñoz for the collaboration in calculating the radiative corrections and for carefully reading the manuscript. I thank S. Brass and I. Sobolev for testing the pre-release and providing important feedback. In addition, I thank H. Bahl, F. Domingo, S. Heinemeyer, S. Paßehr, I. Sobolov and G. Weiglein for helpful correspondence regarding FeynHiggs. I thank T. Stefaniak and J. Wittbrodt for helpful correspondence regarding HiggsBounds and HiggsSignals. This work is supported by the Deutsche Forschungsgemeinschaft under Germany's Excellence Strategy EXC2121 "Quantum Universe" - 390833306.

# References

[1] D. Lopez-Fogliani and C. Muñoz, "Proposal for a Supersymmetric Standard Model", *Phys. Rev. Lett.* **97** (2006) 041801, hep-ph/0508297.

[2] C. Muñoz, "Phenomenology of a New Supersymmetric Standard Model: The mu nu SSM", *AIP Conf. Proc.* **1200** (2010), no. 1, 413–416, arXiv:0909.5140.

[3] T. Biekötter, S. Heinemeyer, and C. Muñoz, "Precise prediction for the Higgs-boson masses in the $\mu\nu$SSM", *Eur. Phys. J. C* **78** (2018), no. 6, 504, arXiv:1712.07475.

[4] T. Biekötter, S. Heinemeyer, and C. Muñoz, "Precise prediction for the Higgs-Boson masses in the $\mu\nu$SSM with three right-handed neutrino superfields", *Eur. Phys. J. C* **79** (2019), no. 8, 667, arXiv:1906.06173.

[5] R.-J. Zhang, "Two loop effective potential calculation of the lightest CP even Higgs boson mass in the MSSM", *Phys. Lett. B* **447** (1999) 89–97, hep-ph/9808299.

[6] J. R. Espinosa and R.-J. Zhang, "MSSM lightest CP even Higgs boson mass to O(alpha(s) alpha(t)): The Effective potential approach", *JHEP* **03** (2000) 026, hep-ph/9912236.

[7] S. Heinemeyer, W. Hollik, and G. Weiglein, "The Masses of the neutral CP - even Higgs bosons in the MSSM: Accurate analysis at the two loop level", *Eur. Phys. J. C* **9** (1999) 343–366, hep-ph/9812472.

[8] E. Kpatcha, R. Ruiz de Austri, D. E. López-Fogliani, and C. Muñoz, "Impact of Higgs physics on the parameter space of the $\mu\nu$SSM", *Eur. Phys. J. C* **80** (2020), no. 4, 336, arXiv:1910.08062.

[9] P. Ghosh, I. Lara, D. E. Lopez-Fogliani, C. Muñoz, and R. Ruiz de Austri, "Searching for left sneutrino LSP at the LHC", *Int. J. Mod. Phys. A* **33** (2018), no. 18n19, 1850110, arXiv:1707.02471.

[10] I. Lara, D. E. López-Fogliani, C. Muñoz, N. Nagata, H. Otono, and R. Ruiz De Austri, "Looking for the left sneutrino LSP with displaced-vertex searches", *Phys. Rev. D* **98** (2018), no. 7, 075004, arXiv:1804.00067.

[11] E. Kpatcha, I. Lara, D. E. López-Fogliani, C. Muñoz, N. Nagata, H. Otono, and R. Ruiz De Austri, "Sampling the $\mu\nu$SSM for displaced decays of the tau left sneutrino LSP at the LHC", *Eur. Phys. J. C* **79** (2019), no. 11, 934, arXiv:1907.02092.

[12] S. Heinemeyer, W. Hollik, and G. Weiglein, "FeynHiggs: A Program for the calculation of the masses of the neutral CP even Higgs bosons in the MSSM", *Comput. Phys. Commun.* **124** (2000) 76–89, hep-ph/9812320.

[13] G. Degrassi, S. Heinemeyer, W. Hollik, P. Slavich, and G. Weiglein, "Towards high precision predictions for the MSSM Higgs sector", *Eur. Phys. J. C* **28** (2003) 133–143, hep-ph/0212020.

[14] M. Frank, T. Hahn, S. Heinemeyer, W. Hollik, H. Rzehak, and G. Weiglein, "The Higgs Boson Masses and Mixings of the Complex MSSM in the Feynman-Diagrammatic Approach", *JHEP* **02** (2007) 047, hep-ph/0611326.

[15] T. Hahn, S. Heinemeyer, W. Hollik, H. Rzehak, and G. Weiglein, "High-Precision Predictions for the Light CP -Even Higgs Boson Mass of the Minimal Supersymmetric Standard Model", *Phys. Rev. Lett.* **112** (2014), no. 14, 141801, arXiv:1312.4937.

[16] H. Bahl and W. Hollik, "Precise prediction for the light MSSM Higgs boson mass combining effective field theory and fixed-order calculations", *Eur. Phys. J. C* **76** (2016), no. 9, 499, arXiv:1608.01880.

[17] H. Bahl, S. Heinemeyer, W. Hollik, and G. Weiglein, "Reconciling EFT and hybrid calculations of the light MSSM Higgs-boson mass", *Eur. Phys. J. C* **78** (2018), no. 1, 57, arXiv:1706.00346.

[18] H. Bahl, T. Hahn, S. Heinemeyer, W. Hollik, S. Paßehr, H. Rzehak, and G. Weiglein, "Precision calculations in the MSSM Higgs-boson sector with FeynHiggs 2.14", *Comput. Phys. Commun.* **249** (2020) 107099, arXiv:1811.09073.

[19] **LHC Higgs Cross Section Working Group** Collaboration, S. Heinemeyer, C. Mariotti, G. Passarino, R. Tanaka, *et al.*, "Handbook of LHC Higgs Cross Sections: 3. Higgs Properties", arXiv:1307.1347.

[20] **LHC Higgs Cross Section Working Group** Collaboration, D. de Florian *et al.*, "Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector", `arXiv:1610.07922`.

[21] P. Bechtle, O. Brein, S. Heinemeyer, G. Weiglein, and K. E. Williams, "HiggsBounds: Confronting Arbitrary Higgs Sectors with Exclusion Bounds from LEP and the Tevatron", *Comput. Phys. Commun.* **181** (2010) 138–167, `arXiv:0811.4169`.

[22] P. Bechtle, O. Brein, S. Heinemeyer, G. Weiglein, and K. E. Williams, "HiggsBounds 2.0.0: Confronting Neutral and Charged Higgs Sector Predictions with Exclusion Bounds from LEP and the Tevatron", *Comput. Phys. Commun.* **182** (2011) 2605–2631, `arXiv:1102.1898`.

[23] P. Bechtle, O. Brein, S. Heinemeyer, O. Stal, T. Stefaniak, G. Weiglein, and K. Williams, "Recent Developments in HiggsBounds and a Preview of HiggsSignals", *PoS* **CHARGED2012** (2012) 024, `arXiv:1301.2345`.

[24] P. Bechtle, O. Brein, S. Heinemeyer, O. Stål, T. Stefaniak, G. Weiglein, and K. E. Williams, "HiggsBounds – 4: Improved Tests of Extended Higgs Sectors against Exclusion Bounds from LEP, the Tevatron and the LHC", *Eur. Phys. J. C* **74** (2014), no. 3, 2693, `arXiv:1311.0055`.

[25] P. Bechtle, S. Heinemeyer, O. Stal, T. Stefaniak, and G. Weiglein, "Applying Exclusion Likelihoods from LHC Searches to Extended Higgs Sectors", *Eur. Phys. J. C* **75** (2015), no. 9, 421, `arXiv:1507.06706`.

[26] P. Bechtle, D. Dercks, S. Heinemeyer, T. Klingl, T. Stefaniak, G. Weiglein, and J. Wittbrodt, "HiggsBounds-5: Testing Higgs Sectors in the LHC 13 TeV Era", `arXiv:2006.06007`.

[27] P. Bechtle, S. Heinemeyer, O. Stål, T. Stefaniak, and G. Weiglein, "*HiggsSignals*: Confronting arbitrary Higgs sectors with measurements at the Tevatron and the LHC", *Eur. Phys. J. C* **74** (2014), no. 2, 2711, `arXiv:1305.1933`.

[28] O. Stål and T. Stefaniak, "Constraining extended Higgs sectors with HiggsSignals", *PoS* **EPS-HEP2013** (2013) 314, `arXiv:1310.4039`.

[29] P. Bechtle, S. Heinemeyer, O. Stål, T. Stefaniak, and G. Weiglein, "Probing the Standard Model with Higgs signal rates from the Tevatron, the LHC and a future ILC", *JHEP* **11** (2014) 039, `arXiv:1403.1582`.

[30] P. Bechtle, S. Heinemeyer, T. Klingl, T. Stefaniak, G. Weiglein, and J. Wittbrodt, "HiggsSignals-2: Probing new physics with precision Higgs measurements in the LHC 13 TeV era", IFT-UAM/CSIC-20-081, to be published.

[31] D. E. Lopez-Fogliani and C. Muñoz, "Searching for Supersymmetry: The $\mu\nu$SSM", `arXiv:2009.01380`.

[32] A. Brignole, L. E. Ibanez, and C. Muñoz, "Soft supersymmetry breaking terms from supergravity and superstring models", *Adv. Ser. Direct. High Energy Phys.* **18** (1998) 125–148, `arXiv:hep-ph/9707209`.

[33] T. Biekötter, "Phenomenology of the Higgs sectors of the $\mu\nu$SSM and the N2HDM", PhD thesis, U. Autonoma, Madrid (main), 2019.

[34] T. Hahn, "Generating Feynman diagrams and amplitudes with FeynArts 3", *Comput. Phys. Commun.* **140** (2001) 418–431, `arXiv:hep-ph/0012260`.

[35] G. A. Gómez-Vargas, D. E. López-Fogliani, C. Muñoz, and A. D. Perez, "MeV-GeV $\gamma$-ray telescopes probing axino LSP/gravitino NLSP as dark matter in the $\mu\nu$SSM", *JCAP* **01** (2020) 058, `arXiv:1911.03191`.

[36] G. A. Gómez-Vargas, D. E. López-Fogliani, C. Muñoz, and A. D. Perez, "MeV-GeV $\gamma$-ray telescopes probing gravitino LSP with coexisting axino NLSP as dark matter in the $\mu\nu$SSM", `arXiv:1911.08550`.

[37] F. Staub, "SARAH 4 : A tool for (not only SUSY) model builders", *Comput. Phys. Commun.* **185** (2014) 1773–1790, `arXiv:1309.7223`.

[38] T. Hahn and M. Perez-Victoria, "Automatized one loop calculations in four-dimensions and D-dimensions", *Comput. Phys. Commun.* **118** (1999) 153–165, `hep-ph/9807565`.

[39] P. Draper and H. Rzehak, "A Review of Higgs Mass Calculations in Supersymmetric Models", *Phys. Rept.* **619** (2016) 1–24, `arXiv:1601.01890`.

[40] U. Ellwanger, C. Hugonie, and A. M. Teixeira, "The Next-to-Minimal Supersymmetric Standard Model", *Phys. Rept.* **496** (2010) 1–77, `arXiv:0910.1785`.

[41] A. Djouadi, J. Kalinowski, and M. Spira, "HDECAY: A Program for Higgs boson decays in the standard model and its supersymmetric extension", *Comput. Phys. Commun.* **108** (1998) 56–74, `hep-ph/9704448`.

[42] M. Spira, "QCD effects in Higgs physics", *Fortsch. Phys.* **46** (1998) 203–284, `hep-ph/9705337`.

[43] J. Butterworth *et al.*, "THE TOOLS AND MONTE CARLO WORKING GROUP Summary Report from the Les Houches 2009 Workshop on TeV Colliders", in "6th Les Houches Workshop on Physics at TeV Colliders". 3 2010. `arXiv:1003.1643`.

[44] A. Bredenstein, A. Denner, S. Dittmaier, and M. Weber, "Precise predictions for the Higgs-boson decay H $\to$ WW/ZZ $\to$ 4 leptons", *Phys. Rev. D* **74** (2006) 013004, `hep-ph/0604011`.

[45] A. Bredenstein, A. Denner, S. Dittmaier, and M. Weber, "Radiative corrections to the semileptonic and hadronic Higgs-boson decays H $\to$ W W / Z Z $\to$ 4 fermions", *JHEP* **02** (2007) 080, `hep-ph/0611234`.

[46] M. D. Goodsell, S. Liebler, and F. Staub, "Generic calculation of two-body partial decay widths at the full one-loop level", *Eur. Phys. J. C* **77** (2017), no. 11, 758, `arXiv:1703.09237`.

[47] M. Spira, "Higgs Boson Production and Decay at Hadron Colliders", *Prog. Part. Nucl. Phys.* **95** (2017) 98–159, `arXiv:1612.07651`.

[48] P. Drechsel, R. Gröber, S. Heinemeyer, M. M. Muhlleitner, H. Rzehak, and G. Weiglein, "Higgs-Boson Masses and Mixing Matrices in the NMSSM: Analysis of On-Shell Calculations", *Eur. Phys. J. C* **77** (2017), no. 6, 366, `arXiv:1612.07681`.

[49] W. G. Hollik, S. Liebler, G. Moortgat-Pick, S. Paßehr, and G. Weiglein, "Phenomenology of the inflation-inspired NMSSM at the electroweak scale", *Eur. Phys. J. C* **79** (2019), no. 1, 75, `arXiv:1809.07371`.

[50] W. G. Hollik, S. Liebler, G. Moortgat-Pick, S. Paßehr, and G. Weiglein, "Phenomenological consequences of Higgs inflation in the NMSSM at the electroweak scale", *PoS* **ICHEP2018** (2019) 455, `arXiv:1811.12838`.

[51] T. Hahn, "Routines for the diagonalization of complex matrices", `physics/0607103`.

# A Example input file

Listing 1: Example input file with random parameter values

```
# munuSSM SUSY PARAMETERS ###############################
2.0             # TanBe
0.00015         # vL_1
0.00061         # vL_2
0.00049         # vL_3
500.0           # vR_1
500.0           # vR_2
500.0           # vR_3
0.2             # lam_1
0.2             # lam_2
0.2             # lam_3
0.3             # kap_111
0.0             # kap_112
0.0             # kap_113
0.0             # kap_122
0.0             # kap_123
0.0             # kap_133
0.4             # kap_222
0.0             # kap_223
0.0             # kap_233
0.42            # kap_333
0.5e-07         # Yv_11
0.0             # Yv_12
0.0             # Yv_13
0.0             # Yv_21
1.7e-07         # Yv_22
0.0             # Yv_23
0.0             # Yv_31
0.0             # Yv_32
2.252e-08       # Yv_33
# munuSSM SOFT PARAMETERS ###############################
0.0             # ml2_12
0.0             # ml2_13
0.0             # ml2_23
0.0             # mlHd2_1
0.0             # mlHd2_2
0.0             # mlHd2_3
0.0             # mv2_12
0.0             # mv2_13
0.0             # mv2_23
1000000.0       # mq2_11 (Squark flavour mixing not yet included)
1000000.0       # mq2_22
1000000.0       # mq2_33
1000000.0       # mu2_11
1000000.0       # mu2_22
1000000.0       # mu2_33
1000000.0       # md2_11
1000000.0       # md2_22
1000000.0       # md2_33
1000000.0       # me2_11
```

```
0.0                # me2_12
0.0                # me2_13
1000000.0          # me2_22
0.0                # me2_23
1000000.0          # me2_33
-1000.0            # Au_11
-1000.0            # Au_22
-1000.0            # Au_33
-1000.0            # Ad_11
-1000.0            # Ad_22
-1000.0            # Ad_33
-1000.0            # Ae_11
0.0                # Ae_12
0.0                # Ae_13
0.0                # Ae_21
-1000.0            # Ae_22
0.0                # Ae_23
0.0                # Ae_31
0.0                # Ae_32
-1000.0            # Ae_33
-1000.0            # Av_11
0.0                # Av_12
0.0                # Av_13
0.0                # Av_21
-1000.0            # Av_22
0.0                # Av_23
0.0                # Av_31
0.0                # Av_32
-1000.0            # Av_33
200.0              # Alam_1
200.0              # Alam_2
200.0              # Alam_3
-300.0             # Akap_111
0.0                # Akap_112
0.0                # Akap_113
0.0                # Akap_122
0.0                # Akap_123
0.0                # Akap_133
-300.0             # Akap_222
0.0                # Akap_223
0.0                # Akap_233
-300.0             # Akap_333
400.0              # M1
600.0              # M2
2000.0             # M3
```

# B Return values and class attributes

In the following tables we list the attributes that are set for an instance of the class `BenchmarkPoint` and the return values for each method defined in the class. In most cases the objects listed in the tables are of type `NumberQP` or `ArrayQP`, as defined in the module `dataObjects` (see Sect. 3). We remind the reader that the values can be obtained in terms of regular floats or `NumPy` float arrays by typing `a.float` if `a` is of type `NumberQP` or `ArrayQP`.

| `self.__init__(file)` | | |
|---|---|---|
| `Delta` | | Value of the UV divergent piece of loop integrals $\Delta = 1/\varepsilon^{\mathrm{UV}}$ |
| `MT_POLE` | | Top quark pole mass $M_t$ |
| `MB_MB` | | Bottom quark mass at the scale $\overline{m}_b(m_b)$ |
| `ScaleFac` | | Renormalization scale in powers of $M_t$: $\mu_R/M_t$ |
| `DRbarScale` | | Input scale of $\overline{\mathrm{DR}}$ parameters $\mu_0$ |
| `MW` | | Mass of the $W$ boson $M_W$ |
| `MZ` | | Mass of the $Z$ boson $M_Z$ |
| `GF` | | Fermi constant $G_F$ |
| `AlfaS` | | Strong QCD coupling constant $\alpha_S(M_Z)$ |
| `MC` | | Charme quark mass $m_c$ |
| `MS` | | Strange quark mass $m_s$ |
| `MU` | | Up quark mass $m_u$ |
| `MD` | | Down quark mass $m_d$ |
| `ML` | | Tauon mass $m_\tau$ |
| `MM` | | Muon mass $m_\mu$ |
| `ME` | | Electron mass $m_e$ |
| `MT` | | SM-$\overline{\mathrm{MS}}$ top quark mass $\overline{m}_t^{\overline{\mathrm{MS}},\mathrm{SM}}(M_t)$ |
| `MB` | | MSSM-$\overline{\mathrm{DR}}$ bottom quark mass $\overline{m}_b^{\overline{\mathrm{DR}},\mathrm{MSSM}}(M_S)$ |
| `TB` | | Ratio of doublet vevs $\tan\beta$ |
| `vL` | `(3, )` | Left-handed sneutrino vevs $v_{iL}$ |
| `vR` | `(3, )` | Right-handed sneutrino vevs $v_{iR}$ |
| `lam` | `(3, )` | Superpotential couplings $\lambda_i$ |
| `kap` | `(3, 3, 3)` | Superpotential couplings $\kappa_{ijk}$ |
| `Yv` | `(3, 3)` | Neutrino Yukawa couplings $Y^\nu_{ij}$ |
| `mlHd2` | `(3, )` | Soft mass parameters $(m^2_{H_d\widetilde{L}})_i$ |
| `mq2` | `(3, 3)` | Soft mass parameters $(m^2_{\widetilde{Q}})_{ij}$ |
| `mu2` | `(3, 3)` | Soft mass parameters $(m^2_{\widetilde{u}})_{ij}$ |
| `md2` | `(3, 3)` | Soft mass parameters $(m^2_{\widetilde{d}})_{ij}$ |
| `me2` | `(3, 3)` | Soft mass parameters $(m^2_{\widetilde{e}})_{ij}$ |
| `Au` | `(3, 3)` | Soft trilinear parameters $A^u_{ij}$ |
| `Ad` | `(3, 3)` | Soft trilinear parameters $A^d_{ij}$ |
| `Ae` | `(3, 3)` | Soft trilinear parameters $A^e_{ij}$ |
| `Av` | `(3, 3)` | Soft trilinear parameters $A^\nu_{ij}$ |
| `Alam` | `(3, )` | Soft trilinear parameters $A^\lambda_i$ |
| `Ak` | `(3, 3, 3)` | Soft trilinear parameters $A^\kappa_{ijk}$ |
| `M1` | | Gaugino mass parameter $M_1$ |

| | | |
|---|---|---|
| M2 | | Gaugino mass parameter $M_2$ |
| M3 | | Gaugino mass parameter $M_3$ |
| CTW | | Cosine of weak mixing angle $c_w$ |
| STW | | Sine of weak mixing angle $s_w$ |
| g1 | | $\mathrm{U}(1)_Y$ gauge coupling $g_1$ |
| g2 | | $\mathrm{SU}(2)_L$ gauge coupling $g_2$ |
| g3 | | $\mathrm{SU}(3)_c$ gauge coupling $g_3$ |
| v | | SM vev $v$ |
| vd | | Down-type vev $v_d$ |
| vu | | Up-type vev $v_u$ |
| Yu | (3, 3) | Up-type quark Yukawa couplings $Y^u_{ij}$ |
| Yd | (3, 3) | Down-type quark Yukawa couplings $Y^d_{ij}$ |
| Ye | (3, 3) | Charged lepton Yukawa couplings $Y^e_{ij}$ |
| Tlam | (3, ) | Soft trilinear couplings $T^\lambda_i$ |
| Tk | (3, 3, 3) | Soft trilinear couplings $T^\kappa_{ijk}$ |
| Tu | (3, 3) | Soft trilinear couplings $T^u_{ij}$ |
| Td | (3, 3) | Soft trilinear couplings $T^d_{ij}$ |
| Te | (3, 3) | Soft trilinear couplings $T^e_{ij}$ |
| Tv | (3, 3) | Soft trilinear couplings $T^\nu_{ij}$ |
| MuDimSq | | Squared renormalization scale $\mu_R^2$ |
| MUE | | Effective $\mu$ parameter: $\mu = \lambda_i v_{iR}/\sqrt{2}$ |
| mHd2 | | Down-type Higgs mass parameter $m^2_{H_d}$ |
| mHu2 | | Up-type Higgs mass parameter $m^2_{H_u}$ |
| mv2 | (3, 3) | Right-handed sneutrino mass parameters $(m^2_{\widetilde{\nu}})_{ij}$ |
| ml2 | (3, 3) | Left-handed sneutrino and slepton mass parameters $(m^2_{\widetilde{L}})_{ij}$ |

Table 4: Class attributes set for an instance of the class `BenchmarkPointFromFile` during initialization. The second column shows the shape of the objects of the type `arrayQP`. If no shape is shown the object is of type `numberQP`.

| `self.calc_tree_level_spectrum()` | | |
|---|---|---|
| MassSt | (2, ) | Stop masses $m_{\widetilde{t}_i}$ |
| ZT | (2, 2) | Stop mixing matrix $Z^{\widetilde{t}}_{ij}$ |
| MassSc | (2, ) | Scalar charme quark masses $m_{\widetilde{c}_i}$ |
| ZC | (2, 2) | Scalar charme quark mixing matrix $Z^{\widetilde{c}}_{ij}$ |
| MassSu | (2, ) | Scalar up quark masses $m_{\widetilde{u}_i}$ |
| ZU | (2, 2) | Scalar up quark mixing matrix $Z^{\widetilde{u}}_{ij}$ |
| MassSb | (2, ) | Sbottom masses $m_{\widetilde{b}_i}$ |
| ZB | (2, 2) | Sbottom mixing matrix $Z^{\widetilde{b}}_{ij}$ |
| MassSs | (2, ) | Scalar strange quark masses $m_{\widetilde{s}_i}$ |
| ZS | (2, 2) | Scalar strange quark mixing matrix $Z^{\widetilde{s}}_{ij}$ |
| MassSd | (2, ) | Scalar down quark masses $m_{\widetilde{d}_i}$ |
| ZD | (2, 2) | Scalar down quark mixing matrix $Z^{\widetilde{d}}_{ij}$ |
| Masshh | (8, ) | Neutral CP-even scalar masses $m_{h_i}$ |
| ZH | (8, 8) | Neutral CP-even scalar mixing matrix $Z^h_{ij}$ |

| | | |
|---|---|---|
| `MassAh` | `(8, )` | Neutral CP-odd scalar masses $m_{A_i}$ (including the Goldstone boson) |
| `ZA` | `(8, 8)` | Neutral CP-odd scalar mixing matrix $Z^{A}_{ij}$ |
| `MassHpm` | `(8, )` | Charged scalar masses $m_{H^{\pm}_i}$ (including the Goldstone boson) |
| `ZP` | `(8, 8)` | Charged scalar mixing matrix $Z^{H^{\pm}}_{ij}$ |
| `MassCha` | `(5, )` | Charged fermion masses $m_{\chi^{\pm}_i}$ |
| `ZEL` | `(5, 5)` | Left-handed charged fermion mixing matrix $Z^{\chi^{\pm}_L}_{ij}$ |
| `ZER` | `(5, 5)` | Right-handed charged fermion mixing matrix $Z^{\chi^{\pm}_R}_{ij}$ |
| `MassChi` | `(10, )` | Neutral fermion masses $m_{\chi^0_i}$ |
| `UV_Re` | `(10, 10)` | Real part of the neutral fermion mixing matrix $Z^{\chi^0}_{ij}$ |
| `UV_Im` | `(10, 10)` | Imaginary part of the neutral fermion mixing matrix $Z^{\chi^0}_{ij}$ |

Table 5: Class attributes set for an instance of the class `BenchmarkPointFromFile` by the method `calc_tree_level_spectrum`. The second column shows the shape of the objects of the type `arrayQP`.

`self.calc_tree_level_couplings()`

| | | | | | |
|---|---|---|---|---|---|
| `hhhh` | `(8, 8, 8, 8)` | $\Gamma_{h_i h_j h_k h_l}$ | `hhh` | `(8, 8, 8)` | $\Gamma_{h_i h_j h_k}$ |
| `AAh` | `(8, 8, 8)` | $\Gamma_{A_i A_j h_k}$ | `AAAA` | `(8, 8, 8, 8)` | $\Gamma_{A_i A_j A_k A_l}$ |
| `AAhh` | `(8, 8, 8, 8)` | $\Gamma_{A_i A_j h_k h_l}$ | `AXX` | `(8, 8, 8)` | $\Gamma_{A_i H^{\pm}_j H^{\mp}_k}$ |
| `hXX` | `(8, 8, 8)` | $\Gamma_{h_i H^{\pm}_j H^{\mp}_k}$ | `AAXX` | `(8, 8, 8, 8)` | $\Gamma_{A_i A_j H^{\pm}_k H^{\mp}_l}$ |
| `AhXX` | `(8, 8, 8, 8)` | $\Gamma_{A_i h_j H^{\pm}_k H^{\mp}_l}$ | `XXXX` | `(8, 8, 8, 8)` | $\Gamma_{H^{\pm}_i H^{\mp}_j H^{\pm}_k H^{\mp}_l}$ |
| `hhXX` | `(8, 8, 8, 8)` | $\Gamma_{h_i h_j H^{\pm}_k H^{\mp}_l}$ | | | |
| `ChaChaA1` | `(5, 5, 8)` | $\Gamma^{-}_{\chi^{\pm}_i \chi^{\mp}_j A_k}$ | `ChaChaA2` | `(5, 5, 8)` | $\Gamma^{+}_{\chi^{\pm}_i \chi^{\mp}_j A_k}$ |
| `ChaChah1` | `(5, 5, 8)` | $\Gamma^{-}_{\chi^{\pm}_i \chi^{\mp}_j h_k}$ | `ChaChah2` | `(5, 5, 8)` | $\Gamma^{+}_{\chi^{\pm}_i \chi^{\mp}_j h_k}$ |
| `ChaChiX1` | `(5, 10, 8)` | $\Gamma^{-}_{\chi^{\pm}_i \chi^{0}_j H^{\mp}_k}$ | `ChaChiX2` | `(5, 10, 8)` | $\Gamma^{+}_{\chi^{\pm}_i \chi^{0}_j H^{\mp}_k}$ |
| `ChiChaX1` | `(10, 5, 8)` | $\Gamma^{-}_{\chi^{0}_i \chi^{\mp}_j H^{\pm}_k}$ | `ChiChaX2` | `(10, 5, 8)` | $\Gamma^{+}_{\chi^{0}_i \chi^{\mp}_j H^{\pm}_k}$ |
| `ChiChiA1` | `(10, 10, 8)` | $\Gamma^{-}_{\chi^{0}_i \chi^{0}_j A_k}$ | `ChiChiA2` | `(10, 10, 8)` | $\Gamma^{+}_{\chi^{0}_i \chi^{0}_j A_k}$ |
| `ChiChih1` | `(10, 10, 8)` | $\Gamma^{-}_{\chi^{0}_i \chi^{0}_j h_k}$ | `ChiChih2` | `(10, 10, 8)` | $\Gamma^{+}_{\chi^{0}_i \chi^{0}_j h_k}$ |
| `ChaChay1` | `(5, 5)` | $\Gamma^{-}_{\chi^{\pm}_i \chi^{\mp}_j \gamma}$ | `ChaChay2` | `(5, 5)` | $\Gamma^{+}_{\chi^{\pm}_i \chi^{\mp}_j \gamma}$ |
| `ChaChaZ1` | `(5, 5)` | $\Gamma^{-}_{\chi^{\pm}_i \chi^{\mp}_j Z}$ | `ChaChaZ2` | `(5, 5)` | $\Gamma^{+}_{\chi^{\pm}_i \chi^{\mp}_j Z}$ |
| `ChaChiW1` | `(5, 10)` | $\Gamma^{-}_{\chi^{\pm}_i \chi^{0}_j W^{\mp}}$ | `ChaChiW2` | `(5, 10)` | $\Gamma^{+}_{\chi^{\pm}_i \chi^{0}_j W^{\mp}}$ |
| `ChiChaW1` | `(10, 5)` | $\Gamma^{-}_{\chi^{0}_i \chi^{\mp}_j W^{\pm}}$ | `ChiChaW2` | `(10, 5)` | $\Gamma^{+}_{\chi^{0}_i \chi^{\pm}_j W^{\pm}}$ |
| `ChiChiZ1` | `(10, 10)` | $\Gamma^{-}_{\chi^{0}_i \chi^{0}_j Z}$ | `ChiChiZ2` | `(10, 10)` | $\Gamma^{+}_{\chi^{0}_i \chi^{0}_j Z}$ |
| `ASbSb` | `(8, 2, 2)` | $2\Gamma_{A_i \widetilde{b}_j \widetilde{\bar{b}}_k}$ | `ASsSs` | `(8, 2, 2)` | $2\Gamma_{A_i \widetilde{s}_j \widetilde{\bar{s}}_k}$ |
| `ASdSd` | `(8, 2, 2)` | $2\Gamma_{A_i \widetilde{d}_j \widetilde{\bar{d}}_k}$ | `AStSt` | `(8, 2, 2)` | $2\Gamma_{A_i \widetilde{t}_j \widetilde{\bar{t}}_k}$ |
| `AScSc` | `(8, 2, 2)` | $2\Gamma_{A_i \widetilde{c}_j \widetilde{\bar{c}}_k}$ | `ASuSu` | `(8, 2, 2)` | $2\Gamma_{A_i \widetilde{u}_j \widetilde{\bar{u}}_k}$ |
| `hSbSb` | `(8, 2, 2)` | $12\Gamma_{h_i \widetilde{b}_j \widetilde{\bar{b}}_k}$ | `hSsSs` | `(8, 2, 2)` | $12\Gamma_{h_i \widetilde{s}_j \widetilde{\bar{s}}_k}$ |
| `hSdSd` | `(8, 2, 2)` | $12\Gamma_{h_i \widetilde{d}_j \widetilde{\bar{d}}_k}$ | `hStSt` | `(8, 2, 2)` | $12\Gamma_{h_i \widetilde{t}_j \widetilde{\bar{t}}_k}$ |
| `hScSc` | `(8, 2, 2)` | $12\Gamma_{h_i \widetilde{c}_j \widetilde{\bar{c}}_k}$ | `hSuSu` | `(8, 2, 2)` | $12\Gamma_{h_i \widetilde{u}_j \widetilde{\bar{u}}_k}$ |
| `AASbSb` | `(8, 8, 2, 2)` | $12\Gamma_{A_i A_j \widetilde{b}_k \widetilde{\bar{b}}_l}$ | `AASsSs` | `(8, 8, 2, 2)` | $12\Gamma_{A_i A_j \widetilde{s}_k \widetilde{\bar{s}}_l}$ |
| `AASdSd` | `(8, 8, 2, 2)` | $12\Gamma_{A_i A_j \widetilde{d}_k \widetilde{\bar{d}}_l}$ | `AAStSt` | `(8, 8, 2, 2)` | $12\Gamma_{A_i A_j \widetilde{t}_k \widetilde{\bar{t}}_l}$ |

| | | | | | |
|---|---|---|---|---|---|
| AAScSc | (8, 8, 2, 2) | $12\Gamma_{A_iA_j\widetilde{c}_k\widetilde{c}_l}$ | AASuSu | (8, 8, 2, 2) | $12\Gamma_{A_iA_j\widetilde{u}_k\widetilde{u}_l}$ |
| hhSbSb | (8, 8, 2, 2) | $12\Gamma_{h_ih_j\widetilde{b}_k\widetilde{b}_l}$ | hhSsSs | (8, 8, 2, 2) | $12\Gamma_{h_ih_j\widetilde{s}_k\widetilde{s}_l}$ |
| hhSdSd | (8, 8, 2, 2) | $12\Gamma_{h_ih_j\widetilde{d}_k\widetilde{d}_l}$ | hhStSt | (8, 8, 2, 2) | $12\Gamma_{h_ih_j\widetilde{t}_k\widetilde{t}_l}$ |
| hhScSc | (8, 8, 2, 2) | $12\Gamma_{h_ih_j\widetilde{c}_k\widetilde{c}_l}$ | hhSuSu | (8, 8, 2, 2) | $12\Gamma_{h_ih_j\widetilde{u}_k\widetilde{u}_l}$ |
| XXZ | (8, 8) | $\Gamma_{H_i^\pm H_j^\mp Z}$ | XXy | (8, 8) | $\Gamma_{H_i^\pm H_j^\mp \gamma}$ |
| XXZZ | (8, 8) | $\Gamma_{H_i^\pm H_j^\mp ZZ}$ | XXyZ | (8, 8) | $\Gamma_{H_i^\pm H_j^\mp \gamma Z}$ |
| XStSb | (8, 2, 2) | $4\Gamma_{H_i^\pm\widetilde{t}_j\widetilde{b}_k}$ | XScSs | (8, 2, 2) | $4\Gamma_{H_i^\pm\widetilde{c}_j\widetilde{s}_k}$ |
| XSuSd | (8, 2, 2) | $4\Gamma_{H_i^\pm\widetilde{u}_j\widetilde{d}_j}$ | | | |

Table 6: Class attributes set for an instance of the class `BenchmarkPointFromFile` by the method `calc_tree_level_couplings`. The name of each attribute is given by `cpl_[1]_[2]` with `[1]` being the string given in the first column and `[2]` being `Re` for the real part and `Im` for the imaginary part of the couplings. The second column shows the shape of the objects of the type `arrayQP`. The couplings of fermions to scalars are decomposed as $\Gamma = \Gamma^-\omega^- + \Gamma^+\omega^+$ with $\omega^\pm = (1 \pm \gamma_5)/2$. The couplings of fermions to vector bosons are decomposed as $\Gamma_\mu = \Gamma^-\gamma_\mu\omega^- + \Gamma^+\gamma_\mu\omega^+$.

| `self.calc_one_loop_counterterms()` | | |
|---|---|---|
| dZhh | (8, 8) | Field renormalization counterterms $\delta Z_{h_ih_j}$ |
| dZAA | (8, 8) | Field renormalization counterterms $\delta Z_{A_iA_j}$ |
| dTphi_Re | (8, ) | Real part of the tadpole counterterms in the gauge basis $\mathrm{Re}(\delta T_{\phi_i})$ |
| dTphi_Im | (8, ) | Imaginary part of the tadpole counterterms in the gauge basis $\mathrm{Im}(\delta T_{\phi_i})$ |
| dMW2 | | $W$-boson mass counterterm $\delta M_W^2$ |
| dMZ2 | | $Z$-boson mass counterterm $\delta M_Z^2$ |
| dmlHd2 | (3, ) | Soft mass parameter counterterms $(\delta m^2_{H_d\widetilde{L}})_i$ |
| dml2Sum12 | | Soft mass parameter counterterm $\delta(m^2_{\widetilde{L}})_{12}$ |
| dml2Sum13 | | Soft mass parameter counterterm $\delta(m^2_{\widetilde{L}})_{13}$ |
| dml2Sum23 | | Soft mass parameter counterterm $\delta(m^2_{\widetilde{L}})_{23}$ |
| dmv2Sum12 | | Soft mass parameter counterterm $\delta(m^2_{\widetilde{\nu}})_{12}$ |
| dmv2Sum13 | | Soft mass parameter counterterm $\delta(m^2_{\widetilde{\nu}})_{13}$ |
| dmv2Sum23 | | Soft mass parameter counterterm $\delta(m^2_{\widetilde{\nu}})_{23}$ |
| dvL2 | (3, ) | Vev counterterms $\delta v_{iL}^2$ |
| dvR2 | (3, ) | Vev counterterms $\delta v_{iR}^2$ |
| dv2 | | Vev counterterm $\delta v^2$ |
| dTanBe | | Parameter counterterm $\delta \tan\beta$ |
| dlam | (3, ) | Superpotential parameter counterterms $\delta\lambda_i$ |
| dkap | (3, 3, 3) | Superpotential parameter counterterms $\delta\kappa_{ijk}$ |
| dYv | (3, 3) | Superpotential parameter counterterms $\delta Y_{ij}^\nu$ |
| dTlam | (3, ) | Soft parameter counterterms $\delta T_i^\lambda$ |
| dTk | (3, 3, 3) | Soft parameter counterterms $\delta T_{ijk}^\kappa$ |
| dM1 | | Gaugino mass parameter counterterm $\delta M_1$ |
| dM2 | | Gaugino mass parameter counterterm $\delta M_2$ |
| dM2phiphi_Re | (8, 8) | Real part of neutral CP-even scalar mass matrix counterterms in gauge basis $\mathrm{Re}(\delta M^2_{\phi_i\phi_j})$ |

| | | |
|---|---|---|
| dM2phiphi_Im | (8, 8) | Imaginary part of neutral CP-even scalar mass matrix counterterms in gauge basis $\mathrm{Im}(\delta M^2_{\phi_i\phi_j})$ |
| dM2sigsig_Re | (8, 8) | Real part of neutral CP-odd scalar mass matrix counterterms in gauge basis $\mathrm{Re}(\delta M^2_{\sigma_i\sigma_j})$ |
| dM2sigsig_Im | (8, 8) | Imaginary part of neutral CP-odd scalar mass matrix counterterms in gauge basis $\mathrm{Im}(\delta M^2_{\sigma_i\sigma_j})$ |
| dM2hh_Re | (8, 8) | Real part of neutral CP-even scalar mass matrix counterterms in tree-level mass eigenstate basis $\mathrm{Re}(\delta M^2_{h_ih_j})$ |
| dM2hh_Im | (8, 8) | Imaginary part of neutral CP-even scalar mass matrix counterterms in tree-level mass eigenstate basis $\mathrm{Im}(\delta M^2_{h_ih_j})$ |
| dM2AA_Re | (8, 8) | Real part of neutral CP-odd scalar mass matrix counterterms in tree-level mass eigenstate basis $\mathrm{Re}(\delta M^2_{A_iA_j})$ |
| dM2AA_Im | (8, 8) | Imaginary part of neutral CP-odd scalar mass matrix counterterms in tree-level mass eigenstate basis $\mathrm{Im}(\delta M^2_{A_iA_j})$ |

Table 7: Class attributes set for an instance of the class `BenchmarkPointFromFile` by the method `calc_one_loop_counterterms`. The second column shows the shape of the objects of the type `arrayQP`. If no shape is shown the object is of the type `numberQP`. The counterterms are calculated including the UV divergent piece proportional to $\Delta = 1/\varepsilon^{\mathrm{UV}}$ (see Tab. 4) The exact definitions of the counterterms can be found in Ref. [4].

| `self.calc_one_loop_self_energies(even,odd,p2_Re,p2_Im)` | | |
|---|---|---|
| hhSERen_Re | (8, 8) | Real part of renormalized one-loop neutral CP-even self energies $\mathrm{Re}(\hat{\Sigma}_{h_ih_j}(p^2))$ with $\mathrm{Re}(p^2)$ given as `p2_Re` and $\mathrm{Im}(p^2)$ given as `p2_Im` |
| hhSERen_Im | (8, 8) | See above, but the imaginary part $\mathrm{Im}(\hat{\Sigma}_{h_ih_j}(p^2))$ |
| AASERen_Re | (8, 8) | Real part of renormalized one-loop neutral CP-odd self energies $\mathrm{Re}(\hat{\Sigma}_{A_iA_j}(p^2))$ with $\mathrm{Re}(p^2)$ given as `p2_Re` and $\mathrm{Im}(p^2)$ given as `p2_Im` |
| AASERen_Im | (8, 8) | See above, but the imaginary part $\mathrm{Im}(\hat{\Sigma}_{h_ih_j}(p^2))$ |

Table 8: The method `calc_one_loop_self_energies` of the class `BenchmarkPointFromFile` returns the values of the renormalized neutral scalar one-loop self energies at the given momentum. The returned object is a dictionary with the keys given in the first column. The first two keys are present if `even=1` is chosen and the latter two keys are present if `odd=1` is chosen. The values belonging to each key are objects of type `arrayQP` with the shape given in the second column.

| `self.calc_two_loop_self_energies(p2_Re,p2_Im)` | | |
|---|---|---|
| hhSERen_Re | (8, 8) | Real part of renormalized neutral CP-even self energies including corrections beyon one-loop level $\mathrm{Re}(\hat{\Sigma}_{h_ih_j}(p^2))$ with $\mathrm{Re}(p^2)$ given as `p2_Re` and $\mathrm{Im}(p^2)$ given as `p2_Im` |
| hhSERen_Im | (8, 8) | See above, but the imaginary part $\mathrm{Im}(\hat{\Sigma}_{h_ih_j}(p^2))$ |

Table 9: The method `calc_two_loop_self_energies` of the class `BenchmarkPointFromFile` returns the values of the renormalized neutral scalar self energies at the given momentum including higher-order corrections beyond one-loop level. The returned object is a dictionary with the keys given in the first column. The values belonging to each key are objects of type `arrayQP` with the shape given in the second column.

| self.calc_loop_masses(even=2,odd=1,accu=1.e-5,momentum_mode=1) | | |
|---|---|---|
| Masshh_L | (8, ) | Loop-corrected CP-even scalar masses at one-loop level $m_{h_i}^{(1)}$ |
| ZH_L_Re | (8, 8) | Real part of loop-corrected CP-even scalar mixing matrix at one-loop level $\mathrm{Re}(Z_{ij}^{h,(1)})$ |
| ZH_L_Im | (8, 8) | Imaginary part of loop-corrected CP-even scalar mixing matrix at one-loop level $\mathrm{Im}(Z_{ij}^{h,(1)})$ |
| Masshh_2L | (8, ) | Loop-corrected CP-even scalar masses including one-loop and higher-order corrections $m_{h_i}^{(2')}$ |
| ZH_2L_Re | (8, 8) | Real part of loop-corrected CP-even scalar mixing matrix including one-loop and higher-order corrections $\mathrm{Re}(Z_{ij}^{h,(2')})$ |
| ZH_L_Im | (8, 8) | Imaginary part of loop-corrected CP-even scalar mixing matrix including one-loop and higher-order corrections $\mathrm{Im}(Z_{ij}^{h,(2')})$ |
| MassAh_L | (8, ) | Loop-corrected CP-odd scalar masses at one-loop level $m_{A_i}^{(1)}$ (including the goldstone boson) |
| ZA_L_Re | (8, 8) | Real part of loop-corrected CP-odd scalar mixing matrix at one-loop level $\mathrm{Re}(Z_{ij}^{A,(1)})$ |
| ZA_L_Im | (8, 8) | Imaginary part of loop-corrected CP-odd scalar mixing matrix at one-loop level $\mathrm{Im}(Z_{ij}^{A,(1)})$ |

Table 10: The method `calc_loop_masses` calculates the loop-corrected neutral scalar spectrum. For `even=1` the attributes `Masshh_L`, `ZH_L_Re` and `ZH_L_Im` are set. For `even=2` the attributes `Masshh_2L_Re`, `ZH_2L_Re` and `ZH_2L_Im` are set. For `odd=1` the attributes `MassAh_L`, `ZA_L_Re` and `ZA_L_Im` are set. The values of each attribute are objects of type `arrayQP` with the shape given in the second column.

| self.calc_effective_couplings() | | |
|---|---|---|
| ScalarCpls.chuu | (8, ) | $c_{h_i u\bar{u}}$ |
| ScalarCpls.chdd | (8, ) | $c_{h_i d\bar{d}}$ |
| ScalarCpls.chbb | (8, ) | $c_{h_i b\bar{b}}$ |
| ScalarCpls.chll | (8, ) | $c_{h_i l\bar{l}}$ |
| ScalarCpls.chtautau | (8, ) | $c_{h_i \tau\bar{\tau}}$ |
| ScalarCpls.chVV | (8, ) | $c_{h_i VV}$ |
| ScalarCpls.chgg | (8, ) | $c_{h_i gg}$ |
| ScalarCpls.chyy | (8, ) | $c_{h_i \gamma\gamma}$ |
| ScalarCpls.chAZ | (8, 8) | $c_{h_i A_j Z}$ |
| Scalarcpls.chXW | (8, 8) | $c_{h_i H_j^\pm W^\mp}$ |
| PseudoscalarCpls.cAuu | (8, ) | $c_{A_i u\bar{u}}$ |
| PseudoscalarCpls.cAdd | (8, ) | $c_{A_i d\bar{d}}$ |
| PseudoscalarCpls.cAbb | (8, ) | $c_{A_i b\bar{b}}$ |
| PseudoscalarCpls.cAll | (8, ) | $c_{A_i l\bar{l}}$ |
| PseudoscalarCpls.cAtautau | (8, ) | $c_{A_i \tau\bar{\tau}}$ |
| PseudoscalarCpls.cAVV | (8, ) | $c_{A_i VV}$ |
| PseudoscalarCpls.cAgg | (8, ) | $c_{A_i gg}$ |

| | | |
|---|---|---|
| PseudoscalarCpls.cAyy | (8, ) | $c_{A_i\gamma\gamma}$ |
| Pseudocalarcpls.cAXW | (8, 8) | $c_{A_i H_j^\pm W^\mp}$ |

Table 11: The method calc_effective_couplings calculates the effective coupling coefficients, i.e., the couplings normalized to the SM prediction, for the neutral scalars. The couplings between $h_i A_j Z$, $h_i H_j^\pm W^\mp$ and $A_i H^\pm W^\mp$ do not have an analogue in the SM. Instead, $c_{h_i A_j Z}$ is given in factors of $e/(2 s_w c_w)$, and $c_{h_i H_j^\pm W^\mp}$ and $c_{A_i H_j^\pm W^\mp}$ in factors of $e/(2 s_w)$. The objects ScalarCpls and PseudoscalarCpls are set as attributes of the instance of BenchmarkPointFromFile. They are themselves instances of the classes Scalars and Pseudoscalars defined in the modules scalars and pseudoscalars of the subpackage effectiveCouplings. Thus, the first column shows the attributes of the instance of BenchmarkPointFromFile. They are NumPy arrays containing floats, with the shape given in the second column.

| self.calc_branching_ratios() | | | |
|---|---|---|---|
| Gammash[i] | hChiChi | (10, 10) | $\Gamma(h_i \to \chi_j^0 \chi_k^0)$ |
| | hChaCha | (5, 5) | $\Gamma(h_i \to \chi_j^\pm \chi_k^\mp)$ |
| | hbb | | $\Gamma(h_i \to b\bar{b})$ |
| | htt | | $\Gamma(h_i \to t\bar{t})$ |
| | hcc | | $\Gamma(h_i \to c\bar{c})$ |
| | hss | | $\Gamma(h_i \to s\bar{s})$ |
| | hgg | | $\Gamma(h_i \to gg)$ |
| | hyy | | $\Gamma(h_i \to \gamma\gamma)$ |
| | hZZ | | $\Gamma(h_i \to ZZ)$ |
| | hWW | | $\Gamma(h_i \to W^\pm W^\mp)$ |
| | hhh | (8, 8) | $\Gamma(h_i \to h_j h_k)$ |
| | hAA | (7, 7) | $\Gamma(h_i \to A_j A_k)$ |
| | hAZ | (7, ) | $\Gamma(h_i \to A_j Z)$ |
| | hXW | (7, 2) | $\Gamma(h_i \to H_j^{\pm,\mp} W^{\pm,\mp})$ |
| | hXX | (7, 7) | $\Gamma(h_i \to H_j^\pm H_k^\mp)$ |
| | hStSt | (2, 2) | $\Gamma(h_i \to \widetilde{t}_j \bar{\widetilde{t}}_k)$ |
| | hScSc | (2, 2) | $\Gamma(h_i \to \widetilde{c}_j \bar{\widetilde{c}}_k)$ |
| | hSuSu | (2, 2) | $\Gamma(h_i \to \widetilde{u}_j \bar{\widetilde{u}}_k)$ |
| | hSbSb | (2, 2) | $\Gamma(h_i \to \widetilde{b}_j \bar{\widetilde{b}}_k)$ |
| | hSsSs | (2, 2) | $\Gamma(h_i \to \widetilde{s}_j \bar{\widetilde{s}}_k)$ |
| | hSdSd | (2, 2) | $\Gamma(h_i \to \widetilde{d}_j \bar{\widetilde{d}}_k)$ |
| | Tot | | $\Gamma_{h_i}^{\mathrm{Tot}}$ |
| BranchingRatiosh[i] | hChiChi | (10, 10) | $\mathrm{Br}(h_i \to \chi_j^0 \chi_k^0)$ |
| | hChaCha | (5, 5) | $\mathrm{Br}(h_i \to \chi_j^\pm \chi_k^\mp)$ |
| | hbb | | $\mathrm{Br}(h_i \to b\bar{b})$ |
| | htt | | $\mathrm{Br}(h_i \to t\bar{t})$ |
| | hcc | | $\mathrm{Br}(h_i \to c\bar{c})$ |
| | hss | | $\mathrm{Br}(h_i \to s\bar{s})$ |
| | hgg | | $\mathrm{Br}(h_i \to gg)$ |
| | hyy | | $\mathrm{Br}(h_i \to \gamma\gamma)$ |
| | hZZ | | $\mathrm{Br}(h_i \to ZZ)$ |

| | | | |
|---|---|---|---|
| | hWW | | $\mathrm{Br}(h_i \to W^\pm W^\mp)$ |
| | hhh | (8, 8) | $\mathrm{Br}(h_i \to h_j h_k)$ |
| | hAA | (7, 7) | $\mathrm{Br}(h_i \to A_j A_k)$ |
| | hAZ | (7, ) | $\mathrm{Br}(h_i \to A_j Z)$ |
| | hXW | (7, 2) | $\mathrm{Br}(h_i \to H_j^{\pm,\mp} W^{\pm,\mp})$ |
| | hXX | (7, 7) | $\mathrm{Br}(h_i \to H_j^\pm H_k^\mp)$ |
| | hStSt | (2, 2) | $\mathrm{Br}(h_i \to \widetilde{t}_j \widetilde{\bar{t}}_k)$ |
| | hScSc | (2, 2) | $\mathrm{Br}(h_i \to \widetilde{c}_j \widetilde{\bar{c}}_k)$ |
| | hSuSu | (2, 2) | $\mathrm{Br}(h_i \to \widetilde{u}_j \widetilde{\bar{u}}_k)$ |
| | hSbSb | (2, 2) | $\mathrm{Br}(h_i \to \widetilde{b}_j \widetilde{\bar{b}}_k)$ |
| | hSsSs | (2, 2) | $\mathrm{Br}(h_i \to \widetilde{s}_j \widetilde{\bar{s}}_k)$ |
| | hSdSd | (2, 2) | $\mathrm{Br}(h_i \to \widetilde{d}_j \widetilde{\bar{d}}_k)$ |
| GammasA[i] | AChiChi | (10, 10) | $\Gamma(A_i \to \chi_j^0 \chi_k^0)$ |
| | AChaCha | (5, 5) | $\Gamma(A_i \to \chi_j^\pm \chi_k^\mp)$ |
| | Abb | | $\Gamma(A_i \to b\bar{b})$ |
| | Att | | $\Gamma(A_i \to t\bar{t})$ |
| | Acc | | $\Gamma(A_i \to c\bar{c})$ |
| | Ass | | $\Gamma(A_i \to s\bar{s})$ |
| | Agg | | $\Gamma(A_i \to gg)$ |
| | Ayy | | $\Gamma(A_i \to \gamma\gamma)$ |
| | AAh | (7, 8) | $\Gamma(A_i \to A_j h_k)$ |
| | AhZ | (8, ) | $\Gamma(A_i \to h_h Z)$ |
| | AXW | (7, 2) | $\Gamma(A_i \to H_j^{\pm,\mp} W^{\pm,\mp})$ |
| | AXX | (7, 7) | $\Gamma(A_i \to H_j^\pm H_k^\mp)$ |
| | AStSt | (2, 2) | $\Gamma(A_i \to \widetilde{t}_j \widetilde{\bar{t}}_k)$ |
| | AScSc | (2, 2) | $\Gamma(A_i \to \widetilde{c}_j \widetilde{\bar{c}}_k)$ |
| | ASuSu | (2, 2) | $\Gamma(A_i \to \widetilde{u}_j \widetilde{\bar{u}}_k)$ |
| | ASbSb | (2, 2) | $\Gamma(A_i \to \widetilde{b}_j \widetilde{\bar{b}}_k)$ |
| | ASsSs | (2, 2) | $\Gamma(A_i \to \widetilde{s}_j \widetilde{\bar{s}}_k)$ |
| | ASdSd | (2, 2) | $\Gamma(A_i \to \widetilde{d}_j \widetilde{\bar{d}}_k)$ |
| | Tot | | $\Gamma_{A_i}^{\mathrm{Tot}}$ |
| BranchingRatiosA[i] | AChiChi | (10, 10) | $\mathrm{Br}(A_i \to \chi_j^0 \chi_k^0)$ |
| | AChaCha | (5, 5) | $\mathrm{Br}(A_i \to \chi_j^\pm \chi_k^\mp)$ |
| | Abb | | $\mathrm{Br}(A_i \to b\bar{b})$ |
| | Att | | $\mathrm{Br}(A_i \to t\bar{t})$ |
| | Acc | | $\mathrm{Br}(A_i \to c\bar{c})$ |
| | Ass | | $\mathrm{Br}(A_i \to s\bar{s})$ |
| | Agg | | $\mathrm{Br}(A_i \to gg)$ |
| | Ayy | | $\mathrm{Br}(A_i \to \gamma\gamma)$ |
| | AAh | (7, 8) | $\mathrm{Br}(A_i \to A_j h_k)$ |
| | AhZ | (8, ) | $\mathrm{Br}(A_i \to h_h Z)$ |
| | AXW | (7, 2) | $\mathrm{Br}(A_i \to H_j^{\pm,\mp} W^{\pm,\mp})$ |
| | AXX | (7, 7) | $\mathrm{Br}(A_i \to H_j^\pm H_k^\mp)$ |
| | AStSt | (2, 2) | $\mathrm{Br}(A_i \to \widetilde{t}_j \widetilde{\bar{t}}_k)$ |

| | | | |
|---|---|---|---|
| | AScSc | (2, 2) | $\mathrm{Br}(A_i \to \widetilde{c}_j \widetilde{\bar{c}}_k)$ |
| | ASuSu | (2, 2) | $\mathrm{Br}(A_i \to \widetilde{u}_j \widetilde{\bar{u}}_k)$ |
| | ASbSb | (2, 2) | $\mathrm{Br}(A_i \to \widetilde{b}_j \widetilde{\bar{b}}_k)$ |
| | ASsSs | (2, 2) | $\mathrm{Br}(A_i \to \widetilde{s}_j \widetilde{\bar{s}}_k)$ |
| | ASdSd | (2, 2) | $\mathrm{Br}(A_i \to \widetilde{d}_j \widetilde{\bar{d}}_k)$ |
| GammasX[i] | XhX | (8, 7) | $\Gamma(H_i^\pm \to h_j H_k^\pm)$ |
| | XAX | (7, 7) | $\Gamma(H_i^\pm \to A_j H_k^\pm)$ |
| | XStSb | (2, 2) | $\Gamma(H_i^\pm \to \widetilde{t}_j \widetilde{\bar{b}}_k)$ |
| | XScSs | (2, 2) | $\Gamma(H_i^\pm \to \widetilde{c}_j \widetilde{\bar{s}}_k)$ |
| | XSuSd | (2, 2) | $\Gamma(H_i^\pm \to \widetilde{u}_j \widetilde{\bar{d}}_k)$ |
| | XhW | (8, ) | $\Gamma(H_i^\pm \to h_j W^\pm)$ |
| | XAW | (7, ) | $\Gamma(H_i^\pm \to A_j W^\pm)$ |
| | XChaChi | (5, 10) | $\Gamma(H_i^\pm \to \chi_j^\pm \chi_k^0)$ |
| | Xtb | | $\Gamma(H_i^\pm \to t\bar{b})$ |
| | Xcs | | $\Gamma(H_i^\pm \to c\bar{s})$ |
| | Tot | | $\Gamma_{H_i^\pm}^{\mathrm{Tot}}$ |

Table 12: The method `calc_branching_ratios` calculates the decay widths and the branching ratios of the neutral and charged scalars. The results are set as attributes `Gammah`, `BranchingRatioh`, `GammaA`, `BranchingRatioA`, `GammaX` and `BranchingRatiosX`, which are lists of dictionaries, to the instance of the class `BenchmarkPointFromFile`. Each dictionary contains the decay widths or branching ratios of a particle $h_i$, $A_i$ or $H_i^\pm$, where $i = 1, 8$ for the CP-even scalars and $i = 1, 7$ for the pseudoscalars and the charged scalars. The different final states of the decays as given in the fourth column correspond to the key values of the dictionaries given in the second column. The values corresponding to each key are floats if no shape is shown in the third column, and `NumPy` arrays with the given shape otherwise.