A Flavor Kit for BSM models

Werner Porod 1, Florian Staub 2, Avelino Vicente 3

1 Institut für Theoretische Physik und Astronomie, Universität Würzburg, 97074 Würzburg, Germany
2 BCTP & Physikalisches Institut der Universität Bonn, Nukallee 12, 53115 Bonn, Germany
3 IFPA, Dep. AGO, Université de Liège, Bat B5, Sart-Tilman B-4000 Liège 1, Belgium

BONN-TH-2014-07

Abstract We present a new kit for the study of flavor observables in models beyond the standard model. The setup is based on the public codes SARAH and SPheno and allows for an easy implementation of new observables. The Wilson coefficients of the corresponding operators in the effective lagrangian are computed by SPheno modules written by SARAH. New operators can also be added by the user in a modular way. For this purpose a handy Mathematica package called PreSARAH has been developed. This uses FeynArts and FormCalc to derive the generic form factors at tree- and 1-loop levels and to generate the necessary input files for SARAH. This framework has been used to implement BR(ℓα → ℓβγ), BR(ℓα → 3ℓβ), CR(µ − e, A), BR(τ → P ℓ), BR(h → ℓαℓβ), BR(Z → ℓαℓβ), BR(Bs,d → ℓℓ), BR(B → Xsγ), BR(B → Xsℓℓ), BR(B → Xd,νν), BR(K+ → π+νν), BR(KL → π0νν), ∆MBS, ∆MK, BR(B → Kµµ), BR(B → ℓν), BR(Ds → ℓν) and BR(K → ℓν) in SARAH. Predictions for these observables can now be obtained in a wide range of SUSY and non-SUSY models. Finally, the user can use the same approach to easily compute additional observables.

Contents

1 Introduction ................................................................. 2
2 General strategy: calculation of flavor observables in a nutshell .................. 3
3 Setup ................................................................. 4
  3.1 FlavorKit: usage and goals ............................................. 4
  3.2 Download and installation ............................................. 6
  3.3 Basic usage .................................................. 6
  3.4 Limitations .................................................. 7
4 Advanced usage I: Implementation of new observables using existing operators .......... 8
5 Advanced usage II: Implementation of new operators .................................. 11
  5.1 Introduction .................................................. 11
  5.2 Input for PreSARAH ............................................. 11
  5.3 Operators with massless gauge bosons ................................ 15
  5.4 Combination and normalization of operators ......................... 16
6 Validation ................................................................. 17
7 Conclusion ................................................................. 20
A: Lagrangian ................................................................. 21
B: Operators available by default in the SPheno output of SARAH .................... 22
  B.1: 2-Fermion-1-Boson operators ...................................... 22
  B.2: 4-Fermion operators ............................................. 27
C: Application: Flavor observables implemented in SARAH .......................... 31
  C.1: Lepton flavor observables ......................................... 31
  C.2: Quark flavor observables ......................................... 44
D: Models ................................................................. 64
  D.1: Supersymmetric Models ............................................ 64
  D.2: Non-Supersymmetric Models ...................................... 65
1 Introduction

With the exploration of the terascale, particle physics has entered a new era. On the one hand, the discovery of a Higgs boson at the LHC [1,2] seemingly completed the Standard Model (SM) of particle physics, even though there is still quite some room for deviations from the SM predictions. The observed mass of about 125 GeV in combination with a top quark mass of 173.34 GeV [3] implies within the SM that we potentially live in a meta-stable vacuum [4]. This, together with other observations, like the dark matter relic density or the unification of gauge forces, indicates that there is physics beyond the SM (BSM). Although no sign of new physics has been found so far at the LHC, colliders are not the only places where one can search for new physics. Low energy experiments focused on flavor observables can also play a major role in this regard, since new particles leave their traces via quantum effects in flavor violating processes such as $b \to s \gamma$, $B_s \to \mu^+ \mu^-$ or $\mu \to e\gamma$. In the last few years there has been a tremendous progress in this field, both on the experimental as well as on the theoretical side. In particular, observables from the Kaon- and B-meson sectors, rare lepton decays and electric dipole moments have put stringent bounds on new flavor mixing parameters and/or additional phases in models beyond the SM.

There are several public tools on the market which predict the rates of several flavor observables: superiso [5–7], SUSY_Flavor [8,9], NMSSM-Tools [10], MicrOmegas [11–15], SuperBSG [16], SupeLFV [17], SuseFlav [18], IsaTools [19–24] or SPheno [25,26]. However, all of these codes have in common that they are only valid in the Two-Higgs-doublet model or in the MSSM or simple extensions of it (NMSSM, bilinear R-parity violation). In addition, none of these tools can be easily extended by the user to calculate additional observables. This has made flavor studies beyond the SM a cumbersome task. The situation has changed with the development of SARAH [27–31]. This Mathematica package can be used to generate modules for SPheno, which then can calculate flavor observables at the 1-loop level in a wide range of supersymmetric and non-supersymmetric models [32–34]. However, so far all the information about the underlying Wilson coefficients\(^1\) for the operators triggering the flavor violation as well as the calculation of the observables had been hardcoded in SARAH. Therefore, it was also very difficult for the user to extend the list of calculated observables. The implementation of new operators was even more difficult.

We present a new kit for the study of flavor observables beyond the standard model. In contrast to previous flavor codes, FlavorKit is not restricted to a single model, but can be used to obtain predictions for flavor observables in a wide range of models (SUSY and non-SUSY). FlavorKit can be used in two different ways. The basic usage of FlavorKit allows for the computation of a large number of lepton and quark flavor observables, using generic analytical expressions for the Wilson coefficients of the relevant operators. The setup is based on the public codes SARAH and SPheno, and thus allows for the analytical and numerical computation of the observables in the model defined by the user. If necessary, the user can also go beyond the basic usage and define his own operators and/or observables. For this purpose, a Mathematica package called PreSARAH has been developed. This tool uses FeynArts/FormCalc [35–40] to compute generic expressions for the required Wilson coefficients at the tree- and 1-loop levels. Similarly, the user can easily implement new observables. With all these tools properly combined, the user can obtain analytical and numerical results for the observables of his interest in the model of his choice. To calculate new flavor observables with SPheno for a given model the user only needs the definition of the operators and the corresponding expressions for the observables as well as the model file for SARAH. All necessary calculations are done automatically. We have used this setup to implement $BR(\ell_\alpha \to \ell_\beta \gamma)$, $BR(\ell_\alpha \to 3 \ell_\beta)$, $CR(\mu - e, A)$, $BR(\tau \to P \ell)$, $BR(h \to \ell_\alpha \ell_\beta)$, $BR(B^0_{s,d} \to \ell \ell)$, $BR(B \to X_\gamma \gamma)$, $BR(B \to X_\tau \ell)$, $BR(B \to X_{d,s} \nu \bar{\nu})$, $BR(K^+ \to \pi^+ \nu \bar{\nu})$, $BR(K_L \to \pi^0 \nu \bar{\nu})$, $\Delta M_{B_s}, \Delta M_{K}, \epsilon_K$, $BR(B \to K_{\mu\bar{\nu}})$, $BR(B \to \ell \nu)$, $BR(D_s \to \ell \nu)$ and $BR(K \to \ell \nu)$ in SARAH.

This manual is structured as follows: in the next section we give a brief introduction into the calculation of flavor observables focusing on the main steps that one has to follow. Then we present FlavorKit, our setup to combine FeynArts/FormCalc, SPheno and SARAH in Section 3. In Section 4 we explain how new observables can be added and in Section 5 how the list of operators can be extended by the user. A comparison between FlavorKit and the other public codes is presented in Section 6 taking the MSSM as an example before we conclude in Section 7. The appendix contains information about the existing operators and how they have been combined to compute the different flavor observables.

\(^1\)Sometimes the Wilson coefficients are also referred to as form factors. We will nevertheless stick to the name Wilson coefficients in the following, also for lepton flavor violating processes.
2 General strategy: calculation of flavor observables in a nutshell

Once we have chosen a BSM model \(^2\), our general strategy for the computation of a flavor observable follows these steps:

- **Step 1:** We first consider an effective Lagrangian that includes the operators relevant for the flavor observable of our interest,

\[
\mathcal{L}_{\text{eff}} = \sum_i C_i \mathcal{O}_i.
\] (1)

This Lagrangian consists of a list of (usually) higher-dimensional operators \(\mathcal{O}_i\). The Wilson coefficients \(C_i\) can be induced either at tree or at higher loop levels and include both the SM and the BSM contributions \((C_i = C_i^{\text{SM}} + C_i^{\text{BSM}})\). They encode the physics of our model.

- **Step 2:** The Wilson coefficients are computed diagrammatically, taking into account all possible tree-level and 1-loop topologies leading to the \(\mathcal{O}_i\) operators \(^3\).

- **Step 3:** The results for the Wilson coefficients are plugged in a general expression for the observable and a final result is obtained.

The user has to make a choice in step 1. The list of operators in the effective Lagrangian can be restricted to the most relevant ones or include additional operators beyond the leading contribution, depending on the required level of precision. Usually, the complete set of renormalizable operators contributing to the observable of interest is considered, although in some well motivated cases one may decide to concentrate on a smaller subset of operators. This freedom is not present in step 2. Once the list of operators has been arranged, the computation of the corresponding \(C_i\) coefficients follows from the consideration of all topologies (penguin diagrams, box diagrams, . . . ) leading to the \(\mathcal{O}_i\) operators. This is the most complicated and model dependent step, since it demands a full knowledge of all masses and vertices in the model under study. Furthermore, it may be necessary to compute the coefficients at an energy scale and then obtain, by means of their renormalization group running, their values at a different scale. Finally, step 3 is usually quite straightforward since, like step 1, is model independent. In fact, the literature contains general expressions for most flavor observables, thus facilitating the final step. However, one should be aware that the formulas given in the literature assume that certain operators contribute only sub-dominantly and, thus, omit the corresponding contributions. This is in general justified for the SM but not in a general BSM model. In particular, this is the case for processes involving external neutrinos, which are often assumed to be purely left-handed, making the operators associated to their right-handed components to be neglected.

We will exemplify our strategy using a simple example: \(\text{BR}(\mu \rightarrow e\gamma)\) in the Standard Model extended by right-handed neutrinos and Dirac neutrino masses. The starting point is, as explained above, to choose the relevant operators. In this case, it is well known that only dipole interactions can contribute to to the radiative decay \(\ell_\alpha \rightarrow \ell_\beta \gamma\) at leading order \(^4\). Therefore, the relevant operators are contained in the \(\ell - \ell - \gamma\) dipole interaction Lagrangian. This is in general given by

\[
\mathcal{L}_{\ell\ell\gamma}^{\text{dipole}} = ie m_\ell \bar{\ell}_\alpha \gamma_{\mu} q_{\nu} \left( K_{L}^{\ell} P_{L}^{\ell} + K_{R}^{\ell} P_{R}^{\ell} \right) \ell_\alpha A_{\mu} + \text{h.c.}
\] (2)

Here \(e\) is the electric charge, \(q\) the photon momentum, \(P_{L,R}^{\ell} = \frac{1}{2}(1 \mp \gamma_5)\) are the usual chirality projectors and \(\ell_{\alpha,\beta}\) denote the lepton flavors. This concludes step 1.

The information about the underlying model is encoded in the coefficients \(K_{L,R}^{\ell}\). In the next step, these coefficients have to be calculated by summing up all Feynman diagrams contributing at a given loop level. Expressions for these coefficients for many different models are available in the literature. In the SM only neutrino loops contribute and one finds [41]

\[
K_{L}^{\ell} = \frac{G_{F}}{2\sqrt{2}\pi^{2}} m_{\mu} \sum_{i} \lambda_{i\mu} \lambda_{\ell\ell}^{*} (F_{1} + F_{2})
\] (3)

\[
K_{R}^{\ell} = \frac{G_{F}}{2\sqrt{2}\pi^{2}} m_{e} \sum_{i} \lambda_{i\mu} \lambda_{\ell\ell}^{*} (F_{1} - F_{2})
\] (4)

\(^2\)The current version of FlavorKit can only handle renormalizable operators at this stage of the computation.

\(^3\)In principle, one can go beyond the 1-loop level, although in our case we will restrict our computation to the addition of a few NLO corrections.

\(^4\)At next to leading order, one would also have to consider operators like \(\bar{\mu} \gamma_{\nu} e \bar{\nu} \gamma^{\nu} q\), to be combined with a \(q - q - \gamma\) dipole interaction.
Here, $\lambda_{ij}$ denote the entries of the Pontecorvo-Maki-Nakagawa-Sakata matrix and $F_1$ and $F_2$ are loop functions. One finds approximately $F_1 \simeq -\frac{1}{4} \left( \frac{m_{H^+}}{m_{W^+}} \right)^2$ and $F_2 \simeq 0$. Finally, we just need to proceed to the last step, the computation of the observable. After computing the Wilson coefficients $K_{2}^{L,R}$ it is easy to relate them to $\text{BR}(\mu \to e\gamma)$ by using [42]

$$\Gamma (\ell_\alpha \to \ell_\beta \gamma) = \frac{\epsilon m^5}{4} \left( |K_{2}^L|^2 + |K_{2}^R|^2 \right),$$

(5)

This expression holds for all models. With this final step, the computation concludes.

As we have seen, the main task to get a prediction for $\text{BR}(\mu \to e\gamma)$ in a new model is to calculate $K_{2}^{L,R}$. However, this demands the knowledge of all masses and vertices involved. Moreover, in most cases a numerical evaluation of the resulting loop integrals is also welcome. Therefore, even for a simple process like $\mu \to e\gamma$, a computation from scratch in a new model can be a hard work. In order to solve this practical problem, we are going to present here a fully automatized way to calculate a wide range of flavor observables for several classes of models.

3 Setup

3.1 FlavoredKit: usage and goals

As we have seen, the calculation of flavor observables in a specific model is a very demanding task. A detailed knowledge about the model is required, including

1. expressions for all involved masses and vertices
2. optionally, renormalization group equations to get the running parameters at the considered scale
3. expressions to calculate the operators
4. formulae to obtain the observables from the operators

Nearly all codes devoted to flavor physics have those pieces hardcoded, and they are only valid for a few specific models. The only exception is SPheno, thanks to its extendability with new modules for additional models. These modules are generated by the Mathematica package SARAH and provide all necessary information about the calculation of the (loop corrected) mass spectrum, the vertices and the 2-loop RGEs. These expressions, derived from fundamental principles for any (renormalizable) model, contain all the information required for the computation of flavor observables. In fact, SARAH also provides Fortran code for a set of flavor observables. For this output, generic expressions of the necessary Wilson coefficients have been included. These are matched to the model chosen by the user and related to the observables by the standard formulae available in the literature. However, it was hardly possible for the user to extend the list of observables or operators included in SARAH without a profound knowledge of either the corresponding Mathematica or Fortran code.

We present a new setup to fill this gap in SARAH: FlavoredKit. As discussed in Sec. 2, the critical step in the computation of a flavor observable is the derivation of analytical expressions for the Wilson coefficients of the relevant operators. This step, being model dependent, requires information about the model spectrum and interactions. However, generic expressions can be derived, later to be matched to the specific spectrum and interaction Lagrangian of a given model. For this purpose, we have created a new Mathematica package called PreSARAH. This package uses the power of FeynArts and FormCalc to calculate generic 1-loop amplitudes, to extract the coefficients of the demanded operators, to translate them into the syntax needed for SARAH and to write the necessary wrapper code. PreSARAH works for any 4-fermion or 2-fermion-1-boson operators and will be extended in the future to include other kinds of operators. The current version already contains a long list of fully implemented operators (see Appendix B). The results for the Wilson coefficients obtained with PreSARAH are then interpreted by SARAH, which adapts the generic expressions to the specific details of the model chosen by the user and uses snippets of Fortran code to calculate flavor observables from the resulting Wilson coefficients. As for the operators, there is a long list of observables already implemented (see Appendices C.1 and C.2). Finally, SARAH can be used to obtain analytical output in \LaTeX\ format or to create Fortran modules for SPheno, thus making possible numerical studies.

FlavoredKit can be used in two ways:

\footnote{Recently, PengBSM\textsc{elo} [43] was made public. This code derives analytical expressions for vector penguins for a model defined in the corresponding FeynArts model file.}
Table 1 List of flavor violating processes and observables which have been already implemented in FlavorKit. To the left, observables related to lepton flavor, whereas to the right observables associated to quark flavor. See appendices C.1 and C.2 for the definition of the observables and the relevant references for their calculation.

- **Basic usage:** This is the approach to be followed by the user who does not need any operator nor observable beyond what is already implemented in FlavorKit. In this case, FlavorKit reduces to the standard SARAH package. The user can use SARAH to obtain analytical results for the flavor observables and, if he wants to make numerical studies, to produce Fortran modules for SPheno. For the list of implemented operators we refer to Appendix B, whereas the list of implemented observables is given in Table 1.

- **Advanced usage:** This is the approach to be followed by the user who needs an operator or an observable not included in FlavorKit. In case the user is interested in an operator that is not implemented in FlavorKit, he can define his own operators and get analytical results for their coefficients using PreSARAH. Then the output can be passed to SARAH in order to continue with the basic usage. In case the user is interested in an observable that is not implemented in FlavorKit, this can be easily implemented by the addition of a Fortran file, with a few lines of code relating the observable to the operators in FlavorKit (implemented by default or added by the user). The Fortran files just have to be put together with a short steering file into a specific directory located in the main SARAH directory. Then one can continue with the basic usage.

The combination of PreSARAH together with SARAH and SPheno allows for a modular and precise calculation of flavor observables in a wide range of particles physics models. We have summarized the setup in Fig. 1:
the user provides as input SARAH model files for his favorite models or takes one of the models which are already implemented in SARAH (see Appendix D for a list of models available in SARAH). New observables are implemented by providing the necessary Fortran code to SARAH while new operators can be either implemented by hand or by using PreSARAH which then calls FeynArts and FormCalc for the calculation of the necessary diagrams. However, most users will not require to implement new operators or observables. In this case, the user can simply use SARAH in the standard way and (1) derive analytical results for the Wilson coefficients and observables, and (2) generate Fortran modules for SPheno in order to run numerical analysis.

3.2 Download and installation

FlavorKit involves several public codes. We proceed to describe how to download and install them.

1. **FeynArts/FormCalc**
   FeynArts and FormCalc can be downloaded from
   www.feynarts.de/

   It is also possible to use the script FeynInstall, to be found on the same site, for an automatic installation.

2. **SARAH and PreSARAH**
   SARAH can be downloaded from
   sarah.hepforge.org/

   No installation or compilation is necessary. Both packages just need to be extracted by using tar.
   > tar -xf SARAH-4.2.0
   > tar -xf PreSARAH-1.0.0

   PreSARAH needs the paths to load FeynArts and FormCalc. These have to be provided by the user in the file PreSARAH.ini

   ```
   FeynArtsPackage = "FeynArts/FeynArts.m"
   FormCalcPackage = "FormCalc/FormCalc.m"
   ```

   This would work if FeynArts and FormCalc have been installed in the Application directory of the local Mathematica installation. Otherwise, absolute paths should be used, e.g.

   ```
   FeynArtsPackage = "/home/$user/path/FeynArts\-3.7/FeynArts.m"
   FormCalcPackage = "/home/$user/path/FormCalc\-8.1/FormCalc.m"
   ```

3. **SPheno**

   SPheno can be downloaded from
   spheno.hepforge.org/

   After extracting the package, make is used for the compilation.
   > tar -xf SPheno-3.3.0.tar.gz
   > cd SPheno-3.3.0
   > make

3.3 Basic usage

As explained above, FlavorKit can be used in several ways, depending on the user’s needs and interests. The advanced usage, which involves the introduction of new observables and/or the computation of new operators, is explained in detail in Secs. 4 and 5. Here we focus on the basic usage, which just requires the codes SARAH and SPheno.

SARAH can handle the analytical derivation of all the relevant Wilson coefficients in the model defined by the user. The resulting expressions can be then extracted in \texttt{B\LaTeX} form or used to generate a SPheno module for numerical evaluation. These are the steps to follow in order to use SARAH:
1. **Loading SARAH**: after starting Mathematica, SARAH is loaded via 
   
   `SARAH-4.2.0/SARAH.m`
   
   or via 
   
   `($path$)/SARAH-4.2.0/SARAH.m`
   
   The first choice works if SARAH has been installed in the `Application` directory of Mathematica. Otherwise, the absolute path `([$path])` to the local SARAH installation must be used.

2. **Initialize a model**: as example for the initialization of a model in SARAH we consider the NMSSM:
   
   `Start["NMSSM"]`;

3. **Obtaining the LaTeX output**: the user can get LaTeX output with all the information about the model (including the coefficients for the flavor operators) via 
   
   `ModelOutput[EWSB];`
   
   `MakeTeX[]`;

4. **Obtaining the SPheno code**: to create the SPheno output the user should run 
   
   `MakeSPheno[]`;

   Thanks to FlavorKit, SARAH can also write LaTeX files with the analytical expressions for the Wilson coefficients. These are given individually for each Feynman diagram contributing to the coefficients, and saved in the folder 
   
   `[$SARAH]/Output/[$MODEL]/EWSB/TeX/FlavorKit/`

   For the 4-fermion operators the results are divided into separated files for tree-level contributions, penguins contributions and box contributions. The corresponding Feynman diagrams are drawn by using FeynMF [44]. To compile all Feynman diagrams at once and to generate the pdf file, a shell script called `MakePDF_[$OPERATOR].sh` is written as well by SARAH.

   In case the user is interested in the numerical evaluation of the flavor observables, a SPheno module must be created as explained above. Once this is done, the resulting Fortran code can be used for the numerical analysis of the model. This can be achieved in the following way:

   1. **building SPheno**: as soon as the SPheno output is finished, open a terminal and enter the root directory of the SPheno installation, and create a new subdirectory, copy the SARAH output to that directory and compile it
      
      `cd [$SPheno]`
      
      `mkdir NMSSM`
      
      `cp [$SARAH]/Output/NMSSM/EWSB/SPheno/* NMSSM/`
      
      `make Model=NMSSM`

   2. **Running SPheno**: After the compilation, a new binary `SPhenoNMSSM` is created. This file can be executed providing a standard Les Houches input file (SARAH provides an example file, see the SARAH output folder). Finally, Spheno is executed via
      
      `./bin/SPhenoNMSSM NMSSM/LesHouches.in.NMSSM`
      
      This generates the output file `Spheno.spc.NMSSM`, which contains the blocks `QFVobservables` and `LFVobservables`. In those two blocks, the results for quark and lepton flavor violating observables are given.

      Finally, an even easier way to implement new models in SARAH is the butler script provided with the SUSY Toolbox [45]

      `sarah.hepforge.org/Toolbox/`

3.4 Limitations

FlavorKit is a tool intended to be as general as possible. For this reason, there are some limitations compared to codes which perform specific calculations in a specific model. Here we list the main limitations of FlavorKit:

   – Chiral resummation is not included because of its large model dependence, see e.g. [46] and references therein.
Even though we have included some of the higher order corrections for the SM part of some observables in a parametric way, 2- or higher loop corrections, calculated in the context of the SM or the MSSM for specific observables, are not considered, see for instance [47–54].

4 Advanced usage I: Implementation of new observables using existing operators

In order to introduce new observables to the SPheno output of SARAH, the user can add new definitions to the directories

\[
\text{[$SARAH$]/FlavorKit/[$Type$]/Processes/}
\]

[$Type$] is either LFV for lepton flavor violating or QFV for quark flavor violating observables. The definition of the new observables consists of two files

1. A steering file with the extension .m
2. A Fortran body with the extension .f90

The steering file contains the following information:

- **NameProcess**: a string as name for the set of observables.
- **NameObservables**: names for the individual observables and numbers which are used to identify them later in the SPheno output. The value is a three dimensional list. The first part of each entry has to be a symbol, the second one an integer and the third one a comment to be printed in the SPheno output file (\{(name1,number1,comment1),…\}).
- **NeededOperators**: The operators which are needed to calculate the observables. A list with all operators already implemented in FlavorKit is given in Appendix B. In case the user needs additional operators, this is explained in Sec. 5.
- **Body**: The name (as string) of the file which contains the Fortran code to calculate the observables from the operators.

For instance, the corresponding file to calculate $\ell_\alpha \rightarrow \ell_\beta \gamma$ reads

```
NameProcess = "LLpGamma";
NameObservables = {
  {muEgamma, 701, "BR(mu->e gamma)"},
  {tauEgamma, 702, "BR(tau->e gamma)"},
  {tauMuGamma, 703, "BR(tau->mu gamma)"};
NeededOperators = {K2L, K2R};
Body = "LLpGamma. f90 ";
```

The observables will be saved in the variables muEgamma, tauEgamma, tauMuGamma and will show up in the spectrum file written by SPheno in the block FlavorKitLFV as numbers 701 to 703.

The file which contains the body to calculate the observables should be standard Fortran90 code. For our example it reads

```
Real(dp) :: width
Integer :: i1 , gt1 , gt2
Do i1=1,3
  If (i1.eq.1) Then ! mu -> e gamma
    gt1 = 2
    gt2 = 1
  ElseIf (i1.eq.2) Then ! tau -> e gamma
    gt1 = 3
    gt2 = 1
  Else ! tau -> mu gamma
    gt1 = 3
    gt2 = 2
  End if
```

Real(dp) is the SPheno internal definition of double precision variables. Similarly one would have to use Complex(dp) for complex double precision variables when necessary.

Besides the operators, the SM parameters given in Table 2 and the hadronic parameters given in Tables 3 and 4 can be used in the calculations. For instance, we used Alpha for $\alpha(0)$ and mf_l which contains the poles masses of the leptons as well as GammaMu and GammaTau for the total widths of $\mu$ and $\tau$ leptons.

| Real Variables          | Real Variables          | Real Variables          | Real Variables          |
|-------------------------|-------------------------|-------------------------|-------------------------|
| AlphaS_MZ $\alpha_S(M_Z)$ | AlphaS_160 $\alpha_S(Q)$ | AlphaS_160 $\alpha_S(Q)$ | AlphaS_MZ $\alpha_S(M_Z)$ |
| sinW2_MZ $\sin(\Theta_W)^2$ at $M_Z$ | sinW2_160 $\sin(\Theta_W)^2$ at $Q$ | sinW2_160 $\sin(\Theta_W)^2$ at $Q$ | sinW2_MZ $\sin(\Theta_W)^2$ at $M_Z$ |
| Alpha_MZ $\alpha(M_Z)\times 10^2$ | Alpha_160 $\alpha(Q)\times 10^2$ | Alpha_160 $\alpha(Q)\times 10^2$ | Alpha_MZ $\alpha(M_Z)\times 10^2$ |
| GammaMu Width $I_\mu$ of $\mu$ | GammaTau Width $I_\tau$ of $\tau$ | GammaTau Width $I_\tau$ of $\tau$ | GammaMu Width $I_\mu$ of $\mu$ |

| Real Vectors of length 3 | Real Vectors of length 3 | Real Vectors of length 3 | Real Vectors of length 3 |
|--------------------------|--------------------------|--------------------------|--------------------------|
| mf_d_160 $m_d(Q)$ | mf_d_160 $m_d(Q)$ | mf_d_160 $m_d(Q)$ | mf_d_160 $m_d(Q)$ |
| mf_u_160 $m_u(Q)$ | mf_u_160 $m_u(Q)$ | mf_u_160 $m_u(Q)$ | mf_u_160 $m_u(Q)$ |
| mf_l_160 $m_l(Q)$ | mf_l_160 $m_l(Q)$ | mf_l_160 $m_l(Q)$ | mf_l_160 $m_l(Q)$ |

| Complex Arrays of dimension 3 x 3 | Complex Arrays of dimension 3 x 3 | Complex Arrays of dimension 3 x 3 | Complex Arrays of dimension 3 x 3 |
|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| CKM_MZ CKM at $M_Z$ | CKM_160 CKM at $Q$ | CKM_160 CKM at $Q$ | CKM_160 CKM at $Q$ |

Table 2 List of SM parameters available in FlavorKit. All hadronic observables are calculated at $Q = 160$ GeV.

By extending or changing the file hadronic_parameters.m in the FlavorKit directory, it is possible to add new variables for the mass or lifetime of mesons. These variables are available globally in the resulting SPheno code. The numerical values for the hadronic parameters can be changed in the Les Houches input file by using the blocks FCN师和FMASS defined in the Flavor Les Houches Accord (FLHA) [55].

It may happen that the calculation of a specific observable has to be adjusted for each model. This is for instance the case when (1) the calculation requires the knowledge of the number of generations of fields, (2) the mass or decay width of a particle, calculated by SPheno, is needed as input, or (3) a rotation matrix of a specific field enters the analytical expressions for the observable. For these situations, a special syntax has been created. It is possible to start a line with @ in the Fortran file. This line will then be parsed by SARAH, and Mathematica commands, as well as SARAH specific commands, can be used. We made use of this functionality in the implementation of $h \rightarrow \ell_\alpha \ell_\beta$. The lines in hLP.f90 read

```fortran
1 ! Check for SM like Higgs
2 @ If [getGen[HiggsBoson] > 1, "hLoc = MaxLoc(Abs(" <>
3  "← ToString[HiggsMixingMatrix]"<>"(2,:)",1)"), "hLoc = 1"]
4 ! Get Higgs mass
5 @ "mh ="<ToString[SPhenoMass[HiggsBoson]]" <>
6  ← If [getGen[HiggsBoson] > 1,"hLoc ", "]
```
The user can define in the SPheno input file. The values can be changed according to the FLHA conventions using the block FCONST in the Les Houches input file.

Table 3 Hadronic parameters used in FlavorKit. These can be changed via FMASS and and FLIFE in the Les Houches input file.

| Decay constant | Variable | default [MeV] | FLHA |
|----------------|----------|---------------|------|
| \( f_K \)     | \( f_{K_{-1}} \) | 176           | FCONST[321,1] |
| \( f_{K^+} \)  | \( f_{K^+_1} \) | 156           | FCONST[323,1] |
| \( f_\phi \)   | \( f_{\phi_{-1}} \) | 118           | FCONST[311,1] |
| \( f_\rho \)   | \( f_{\rho_{-1}} \) | 194           | FCONST[511,1] |
| \( f_{\phi_{1}} \) | \( f_{\phi_{1+}} \) | 234           | FCONST[531,1] |
| \( f_{\phi_{2}} \) | \( f_{\phi_{1+}} \) | 234           | FCONST[521,1] |
| \( f_{\phi_{3}} \) | \( f_{\phi_{1+}} \) | 172           | FCONST[231,1] |
| \( f_{\rho_{0}} \) | \( f_{\rho_{0}} \) | 220           | FCONST[213,1] |
| \( f_{\omega_{0}} \) | \( f_{\omega_{0}} \) | 256           | FCONST[411,1] |
| \( f_{\omega_{1}} \) | \( f_{\omega_{1}} \) | 248           | FCONST[431,1] |

Table 4 Decay constants available in the SPheno output of SARAH. The values can be changed according to the FLHA conventions using the block FCONST in the Les Houches input file.

In this implementation we define an integer \( h\text{Loc} \) that gives the generation index of the SM-like Higgs, to be found among all CP even scalars. In the first line it is checked if more than one scalar Higgs is present. If this is the case, the \( h\text{Loc} \) is set to the component which has the largest amount of the up-type Higgs, if not, it is just put to 1. Of course, this assumes that the electroweak basis in the Higgs sector is always defined as \((d_d, d_u, \ldots)\) as is the case for all models delivered with SARAH. In the second and third lines, the variables \( m_h \) and \( g_mh \) are set to the mass and total width of the SM-like Higgs, respectively. For this purpose, the SARAH commands SPhenoMass[x] and SPhenoWidth[x] are used. They return the name of the variable for the mass and width in SPheno and it is checked if these variables are arrays or not. For the MSSM, the above lines lead to the following code in the SPheno output:

```plaintext
7 ! Get Higgs width
8 @ "g_mh ="ToString[SPhenoWidth[HiggsBoson]] ⇔ ⇔
   ⇔ If [getGen[HiggsBoson] > 1, "(hLoc) ", ""]
```

The user can define in the parameters.m and particles.m file for a given model in SARAH the particles which should be taken to be the CP-even or CP-odd Higgs and the parameter that corresponds to their rotation matrices. This
| getGen[x]                      | returns the number of generations of a particle x |
|------------------------------|--------------------------------------------------|
| getDim[x]                    | returns the dimension of a variable x             |
| SPhenoMass[x]                | returns the name used for the mass of a particle x in the SPheno output |
| SPhenoMassSq[x]              | returns the name used for the mass squared of a particle x in the SPheno output |
| SPhenoWidth[x]               | returns the name used for the width of a particle x in the SPheno output |

| HiggsMixingMatrix             | name of the mixing matrix for the CP even Higgs states in a given model |
| PseudoScalarMixingMatrix      | name of the mixing matrix for the CP odd Higgs states in a given model |

Table 5 SARAH commands which can be used in the input file for the calculation of an observable.

```
! Check for SM like Higgs
hLoc = MaxLoc(Abs(ZH(2,:)),1)

! Get Higgs mass
mh = Mhh(hLoc)

! Get Higgs width
gamh = gThh(hLoc)
```

We give in Table 5 the most important SARAH commands which might be useful in this context. Many more examples are given in Appendix C.1, where we have added all input files for the calculations of flavor observables delivered with SARAH.

5 Advanced usage II: Implementation of new operators

The user can also implement new operators and obtain analytical expressions for their Wilson coefficients. In this case, he will need to use PreSARAH which, with the help of FeynArts and FormCalc, provides generic expressions for the coefficients, later to be adapted to specific models with SARAH.

5.1 Introduction

New operators can be implemented by extending the content of the folder

```
[SARAH]/FlavorKit/[Type]/Operators/
```

In the current version of FlavorKit, 3- and 4-point operators are supported. Each operator is defined by a .m-file. These files contain information about the external particles, the kind of considered diagrams (tree-level, self-energies, penguins, boxes) as well as generic expressions for the coefficients. These expressions, derived from the generic Feynman diagrams contributing to the coefficients, are written in the form of a Mathematica code, which can be used to generate Fortran code.

For the automatization of the underlying calculations we have created an additional Mathematica package called PreSARAH, which can be used to create the files for all 4-fermion as well as 2-fermion-1-boson operators. This package creates not only the infrastructure to include the operators in the SPheno output of SARAH but makes also use of FeynArts and FormCalc to calculate the amplitudes and to extract the coefficient of the demanded operators. It takes into account all topologies depicted in Figs. 2 to 6.

5.2 Input for PreSARAH

In order to derive the results for the Wilson coefficients, PreSARAH needs an input file with the following information:

is done by using the Description statements Higgs or Pseudo-Scalar Higgs as well as Scalar-Mixing-Matrix or Pseudo-Scalar-Mixing-Matrix. If the particle or parameter needed to calculate an observable is not present or has not been defined, the observable is skipped in the SPheno output.
Fig. 2 All topologies considered by PreSARAH to calculate the Wilson coefficients of 2-fermion-1-boson operators. All possible generic combinations of the internal fields are taken into account.

Fig. 3 All tree topologies considered by PreSARAH to calculate the Wilson coefficients of 4-fermion operators. All possible generic combinations of the internal fields are taken into account.

- **ConsideredProcess**: A string which defines the generic type for the process
  - "4Fermion"
  - "2Fermion1Scalar"
  - "2Fermion1Vector"
- **NameProcess**: A string to uniquely define the process
- **ExternalFields**: The external fields. Possible names are ChargedLepton, Neutrino, DownQuark, UpQuark, ScalarHiggs, PseudoScalar, Zboson, Wboson
- **FermionOrderExternal**: the fermion order to apply the Fierz transformation (see the FormCalc manual for more details)
- **NeglectMasses**: which external masses can be neglected (a list of integers counting the external fields)
- **ColorFlow**: defines the color flow in the case of four quark operators. To contract the colors of external fields, ColorDelta is used, i.e ColorFlow = ColorDelta[1,2]*ColorDelta[3,4] assigns \((q^\alpha q_{\alpha})(q'^\beta q_{\beta})\).
- **AllOperators**: a list with the definition of the operators. This is a two dimensional list, where the first entry defines the name of the operator and the second one the Lorentz structure. The operators are expressed in the chiral basis and the syntax for Dirac chains in FormCalc is used:
  - 6 for \(P_L = \frac{1}{2}(1 - \gamma_5)\), 7 for \(P_R = \frac{1}{2}(1 - \gamma_5)\)
  - Lor[1], Lor[2] for \(\gamma_\mu, \gamma_\nu\)
  - ec[3] for the helicity of an external gauge boson.
  - \(k[N]\) for the momentum of the external particle \(N\) (\(N\) is an integer).

7 The particles.m file is used to define for each model which particle corresponds to SM states using the Description statement together with Leptons, Neutrinos, Down-Quarks, Up-Quarks, Higgs, Pseudo-Scalar Higgs, Z-Boson, W-Boson. If there is a mixture between the SM particles and other states (like in R-parity violating SUSY or in models with additional vector quarks/leptons) the combined state has to be labeled according to the description for the SM state. Notice that in the SM Pseudo-Scalar Higgs is just the neutral Goldstone boson. If an external state is not present in a given model or has not been defined as such in the particles.m file the corresponding Wilson coefficients are not calculated by SPheno.
Fig. 4 All self-energy topologies considered by PreSARAH to calculate the Wilson coefficients of 4-fermion operators. All possible generic combinations of the internal fields are taken into account.

- Pair[A,B] is used to contract Lorentz indices. For instance, Pair[k[1],ec[3]] stands for $k_\mu^1 \epsilon^{\mu*}$.
- A Dirac chain starting with a negative first entry is taken to be anti-symmetrized. See the FormCalc manual for more details.

To make the definitions more readable, not the full DiracChain object of FeynArts/FormCalc has to be defined: PreSARAH puts everything with the head Op into a Dirac chain using the defined fermion order. For 4-fermion operators the combination of both operators is written as dot product. For instance Op[6].Op[6] is internally translated into

```
DiracChain[Spinor[k[1],MassEx1,-1],6,Spinor[k[2],MassEx2,1]]*
DiracChain[Spinor[k[3],MassEx3,-1],6,Spinor[k[4],MassEx4,1]]
```

while Op[6] Pair[ec[3],k[1]] becomes

```
DiracChain[Spinor[k[1],MassEx1,-1],6,Spinor[k[2],MassEx2,1]] Pair[ec[3],k[1]]
```

- CombinationGenerations: the combination of external generations for which the operators are calculated by SPheno
- Filters: a list of filters to drop specific diagrams. Possible entries are NoBoxes, NoPenguins, NoTree, NoCrossedDiagrams.
  - Filters = {NoBoxes, NoPenguins} can be used for processes which are already triggered at tree-level
  - Filters = {NoPenguins} might be useful for processes which at the 1-loop level are only induced by box diagrams
Fig. 5 All penguin topologies considered by PreSARAH to calculate the Wilson coefficients of 4-fermion operators. All possible generic combinations of the internal fields are taken into account.

Fig. 6 All box topologies considered by PreSARAH to calculate the Wilson coefficients of 4-fermion operators. All possible generic combinations of the internal fields are taken into account.

– Filters = {NoCrossedDiagrams} is used to drop diagrams which only differ by a permutation of the external fields.

For instance, the PreSARAH input to calculate the coefficient of the $(\bar{\ell}\ell')(\bar{d}d')$ operator reads

```plaintext
1 NameProcess="2L2d*";
2 ConsideredProcess = "4Fermion";
3 ExternalFields={{ChargedLepton , bar[ChargedLepton] ,
4 DownQuark , bar[DownQuark]}};
5 FermionOrderExternal={2,1,4,3};
6 NeglectMasses={1,2,3,4};
7 AllOperators={
8 (* scalar operators *)
9 {OllddSLL , Op[7] , Op[7]} ,
10 {OllddSRR , Op[6] , Op[6]} ,
11 {OllddSRL , Op[6] , Op[7]} ,
12 {OllddSLR , Op[7] , Op[6]} ,
13 (* vector operators *)
14 {OllddVRR , Op[7 , Lor[1] , Op[7 , Lor[1]]]} ,
15 {OllddVLL , Op[6 , Lor[1] , Op[6 , Lor[1]]]} ,
16 {OllddVRL , Op[7 , Lor[1] , Op[6 , Lor[1]]]} ,
17 {OllddVLR , Op[6 , Lor[1] , Op[7 , Lor[1]]]} ,
```
Here, we neglect all external masses in the operators (\text{NeglectMasses}={1,2,3,4}), and the different coefficients of the scalar operators \((\bar{\ell} P_X \ell)(\bar{d} P_Y d)\) are called \(\text{OllddSXY}\), the ones for the vector operators \((\bar{\ell} P_X \gamma_\mu \ell)(\bar{d} P_Y \gamma_\mu d)\) are called \(\text{OllddVYX}\) and the ones for the tensor operators \((\bar{\ell} P_X \sigma_{\mu\nu} \ell)(\bar{\sigma}_{\mu\nu} P_Y d)\) \(\text{OllddTYX}\), with \(X,Y=L,R\).

Notice that \text{FormCalc} returns the results in form of \(P_X \gamma_\mu\) while in the literature the order \(\gamma_\mu P_X\) is often used. Finally, \text{SPheno} will not calculate all possible combinations of external states, but only some specific cases: \(\mu\bar{e}d\), \(\tau\bar{e}d\), \(\tau\bar{\mu}dd\), \(\mu\bar{e}ss\), \(\tau\bar{e}ss\), \(\tau\bar{\mu}ss\).

The input file to calculate the coefficients of the \(\ell-\ell-Z\) operators \((\bar{\ell} P_{L,R} \ell)Z^\mu\) and \((\bar{\ell} P_{\mu L,R} \gamma_\mu \ell)Z^\mu\) is

\begin{verbatim}
NameProcess="Z2l ";
ConsideredProcess = "2Fermion1Vector ";
FermionOrderExternal={1,2};
NeglectMasses={1,2};

ExternalFields= {ChargedLepton , bar[ChargedLepton] , Zboson};
CombinationGenerations = {{1,2} ,{1,3},{2,3}};

AllOperators={
  {OZ2lSL ,Op[7]} , {OZ2lSR ,Op[6]} ,
  {OZ2lVL ,Op[7 , ec [3]]} , {OZ2lVR ,Op[6 , ec [3]]}
};

OutputFile = "Z2l .m";
Filters = { }; 
\end{verbatim}

Note that \text{ExternalFields} must contain first the involved fermions and the boson at the end. Furthermore, in the case of processes involving scalars one can define

\begin{verbatim}
ExternalFields= {ChargedLepton , bar[ChargedLepton] , ScalarHiggs};
CombinationGenerations = {{1,2,ALL} , {1,3,ALL} , {2,3,ALL}};
\end{verbatim}

In this case the operators for all Higgs states present in the considered model will be computed.

5.3 Operators with massless gauge bosons

We have to add a few more remarks concerning 2-fermion-1-boson operators with massless gauge bosons since those are treated in a special way. It is common for these operators to include terms in the amplitude which

\[8\] Here we used \(d\) for the first generation of down-type quarks while in the rest of this manual it is used to summarize all three families.
are proportional to the external masses. Therefore, if one proceeds in the usual way and neglects the external
momenta, some inconsistencies would be obtained. For this reason, a special treatment is in order. In PreSARAH,
when one uses

```
ConsideredProcess = "2Fermion1Vector";
FermionOrderExternal = {1,2};
NeglectMasses = {3};
```

the dependence on the two fermion masses is neglected in the resulting Passarino-Veltman integrals but terms
proportional to $m_f$ and $m_{f_i}$ are kept. This solves the aforementioned potential inconsistencies.

Furthermore, for the dipole operators, defined by

```
{DipoleL, Op[6], Pair[ec[3],k[1]]},
{DipoleR, Op[7], Pair[ec[3],k[1]]},
```

we are using the results obtained by FeynArts and FormCalc and have implemented all special cases for the
involved loop integrals ($C_0, C_{00}, C_1, C_2, C_{11}, C_{12}, C_{22}$) with identical or vanishing internal masses in SPheno.
This guarantees the numerical stability of the results.

The monopole operators of the form $q^2(f_{\gamma\mu}\bar{f}V^\mu)$ are only non-zero for off-shell external gauge bosons,
while PreSARAH always treats all fields as on-shell. Because of this, and to stabilize the numerical evaluation
later on, these operators are treated differently to all other operators: the coefficients are not calculated by
FeynArts and FormCalc but instead we have included the generic expressions in PreSARAH using a special
set of loop functions in SPheno. In these loop functions the resulting Passarino-Veltman integrals are already
combined, leading to well-known expressions in the literature, see [42,56]. They have been cross-checked with
the package Peng4BSM@LO [43]. To get the coefficients for the monopole operators, these have to be given always
in the form

```
{MonopoleL, Op[6, ec[3]] Pair[k[3], k[3]]},
{MonopoleR, Op[7, ec[3]] Pair[k[3], k[3]]},
```

in the input of PreSARAH.

5.4 Combination and normalization of operators

The user can define new operators as combination of existing operators. For this purpose wrapper files con-
taining the definition of the operators can be included in the FlavorKit directories. These files have to begin
with `ProcessWrapper = True;`. This function is also used by PreSARAH in the case of 4-fermion operators: for
these operators the contributions stemming from tree-level, box- and penguin- diagrams are saved separately
and summed up at the end. Thus, the wrapper code for the 4-lepton operators written by PreSARAH reads

```
ProcessWrapper = True;
NameProcess = "4L";
ExternalFields = {ChargedLepton, bar[ChargedLepton], ChargedLepton, ->
                bar[ChargedLepton]};
SumContributionsOperators["4L"] = {
  {O4ISSL, BO4ISSL + PSO4ISSL + PV0ISSL + TVO4ISSL},
  {O4ISR, BO4ISR + PSO4ISR + PV0ISR + TVO4ISR},
  ...}
```

It is also possible to use these wrapper files to change the normalization of the operators. We have made use of
this functionality for the operators with external photons and gluons to match the standard definition used in
literature: it is common to write these operators as $e m_f (f_{\gamma\mu} \bar{f}V^\mu)$, i.e. with the electric coupling (or strong
coupling for gluon operators) and fermion mass factored out. This is done with the files Photon_wrapper.m
and Gluon_wrapper.m, included in the FlavorKit directory of SARAH:

9We note that the coefficients for the operators defined above ($f_{\gamma\mu} \bar{f}V^\mu$) are by a factor of 2 (4) larger than the coefficients
of the standard definition for the dipole operators $f_{\sigma\mu\nu} P_L f q^\nu V^\mu (f_{\sigma\mu\nu} P_L f F^\mu\nu)$. 
First, the coefficients $O_{A2qSL}$ and $O_{A2qSR}$ derived with PreSARAH are matched to the new coefficients $CC7$ and $CC7p$. The same matching is automatically applied also to the SM coefficients $O_{A2qSLSM}$ and $O_{A2qSRSM}$. In a second step, these operators are re-normalized to the standard definition of the Wilson coefficients $C_7$ and $C'_7$.

The initial coefficients $O_{A2qSR}$, $O_{A2qSL}$ are not discarded, but co-exist besides $CC7$, $CC7p$. Thus, the user can choose in the implementation of the observables which operators are more suitable for his purposes.

6 Validation

The validation of the FlavorKit results happened in three steps:

1. Agreement with SM results: we checked that the SM prediction for the observables agree with the results given in literature
2. Independence of scale in loop function: the loop integrals for two and three point functions ($B_i, C_i$) depend on the renormalization scale $Q$. However, this dependence has to drop out for a given process at leading order. We checked numerically that the sum of all diagrams is indeed independent of the choice of $Q$.
3. Comparison with other tools: as we have pointed out in the introduction, there are several public tools which calculate flavor observables mostly in the context of the MSSM. We did a detailed comparison with these tools using the SPheno code produced by SARAH for the MSSM. Some results are presented in the following.

We have compared the FlavorKit results using SARAH4.2.0 and SPheno3.3.0 with

- superiso 3.3
- SUSY_Flavor 1 and 2.1
- MicrOmegas 3.6.7
- SPheno 3.3.0
- a SPheno version produced by SARAH 4.1.0 without the FlavorKit functionality

Since these codes often use different values for the hadronic parameters and calculate the flavor observables at different loop levels, we are not going to compare the absolute numbers obtained by these tools. Instead, we compare the results normalized to the SM prediction of each code and thus define, for an observable $X$, the ratio

$$R(X) = \frac{X_{MSSM}}{X_{SM}}.$$  \hspace{1cm} (6)

$X_{SM}$ is obtained by taking the value of $X$ calculated by each code in the limit of a very heavy SUSY spectrum.

As test case we have used the CMSSM. The dependence of a set of flavor observables as function of $m_0$ is shown in Fig. 7 and as function of $M_{1/2}$ in Fig. 8.

We see that all codes show in general the same dependence. However, it is also obvious that the lines are not on
Fig. 7  Comparison of the results for $\text{BR}(B^0_{s,d} \to \mu\mu)$, $\text{BR}(B \to X_s \gamma)$, $\text{BR}(B \to \tau \nu)$, $\Delta M_B$, $\varepsilon_K$, $\text{BR}(K_L \to \pi^0 \nu \bar{\nu})$, $\text{BR}(K^+ \to \pi^+ \nu \bar{\nu})$ as a function of $m_0$ using the FlavorKit (red), superiso (purple), SUSY_Flavor 1 (brown), SUSY_Flavor 2 (green), SPheno (blue), MicrOmegas (orange) and the old implementation in SARAH (red dashed). The three lines for SUSY_Flavor 2 correspond to different options of the chiral resummation. We used $M_{1/2} = 200$ GeV, $A_0 = 0$, $\tan \beta = 10$, $\mu > 0$.

top of each other but differences are present. These differences are based on the treatment of the resummation of the bottom Yukawa couplings, the different order at which SM and SUSY contributions are implemented,
the different handling of the Weinberg angle, and the different level at which the RGE running is taken into account by the tools. Even if a detailed discussion of the differences of all codes might be very interesting it is, of course, far beyond the scope of this paper and would require a combined effort. The important point is that the results of {\textit{FlavorKit}} agree with the codes specialized for the MSSM to the same level as those codes agree among each other. Since the {\textit{FlavorKit}} results for all observables are based on the same generic routines it
might be even more trustworthy than human implementations of the lengthy expressions needed to calculate these observables because it is less error prone. Of course, known 2-loop corrections for the MSSM which are implemented in other tools are missing.

Finally, it is well known that the process $B_{s,d}^0 \rightarrow \ell\bar{\ell}$ has a strong dependence on the value of $\tan \beta$. We show in Fig. 9 that this is reproduced by all codes.

![Fig. 9](image.png)

**Fig. 9** Comparison of $BR(B_{s,d}^0 \rightarrow \mu\bar{\mu})$ (first row) and $BR(B_{s,d}^0 \rightarrow e^+e^-)$ (second row) as function of $\tan \beta$. The color code is the same as in Fig. 7. We used $m_0 = M_{1/2} = 500$ GeV, $A_0 = 0$, $\mu > 0$.

7 Conclusion

We have presented **FlavorKit**, a new setup for the calculation of flavor observables for a wide range of BSM models. Generic expressions for the Wilson coefficients are derived with PreSARAH, a Mathematica package that makes use of **FeynArts** and **FormCalc**. The output of PreSARAH is then passed to SARAH, which generates the Fortran code that allows to calculate numerically the values of these Wilson coefficients with SPheno. The observables are derived by providing the corresponding pieces of Fortran code to SARAH, which incorporates them into the SPheno output. We made use of this code chain to fully implement a large set of important flavor observables in SARAH and SPheno. In fact, due the simplicity of this kit, the user can easily extend the list with his own observables and operators. In conclusion, **FlavorKit** allows the user to easily obtain analytical and numerical results for flavor observables in the BSM model of his choice.

Acknowledgments

We thank Asmaa Abada, Martin Hirsch, Farvah Mahmoudi, Manuel E. Krauss, Kilian Nickel, Ben O’Leary and Cédric Weiland for helpful discussions. FS is supported by the BMBF PT DESY Verbundprojekt 05H2013-THEORIE 'Vergleich von LHC-Daten mit supersymmetrischen Modellen'. WP is supported by the DFG,
A: Lagrangian

In this section we present our notation and conventions for the operators (and their corresponding Wilson coefficients) implemented in PreSARAH. Although a more complete list of flavor violating operators can be built, we will concentrate on those implemented in PreSARAH. If necessary, the user can extend it by adding his/her own operators.

The interaction Lagrangian relevant for flavor violating processes can be written as

$$\mathcal{L}_{\text{FI}} = \mathcal{L}_{\text{LFV}} + \mathcal{L}_{\text{QFV}}.$$  \hspace{1cm} (A.1)

The first piece contains the operators that can trigger lepton flavor violation whereas the second piece contains the operators responsible for quark flavor violation.

The general Lagrangian relevant for lepton flavor violation can be written as

$$\mathcal{L}_{\text{LFV}} = \mathcal{L}_{\ell \ell \gamma} + \mathcal{L}_{\ell \ell Z} + \mathcal{L}_{\ell \ell h} + \mathcal{L}_{4\ell} + \mathcal{L}_{2\ell 2q}.$$  \hspace{1cm} (A.2)

The first term contains the $\ell - \ell - \gamma$ interaction, given by

$$\mathcal{L}_{\ell \ell \gamma} = \bar{\ell}_\beta \gamma \left( K^1_L P_L + K^1_R P_R \right) \ell_\alpha A_\mu + \text{h.c.}.$$  \hspace{1cm} (A.3)

Here $e$ is the electric charge, $q$ the photon momentum, $P_{L,R} = \frac{1}{2} (1 \mp \gamma_5)$ are the usual chirality projectors and $\ell_{\alpha,\beta}$ denote the lepton flavors. For practical reasons, we will always consider the photonic contributions independently, and we will not include them in other vector operators. On the contrary, the $Z$- and Higgs boson contributions will be included whenever possible. Therefore, the $\ell - \ell - Z$ and $\ell - \ell - h$ interaction Lagrangians will only be used for observables involving real $Z$- and Higgs bosons. These two Lagrangians can be written as

$$\mathcal{L}_{\ell \ell Z} = \bar{\ell}_\beta \left[ \gamma^\mu \left( R^1_L P_L + R^1_R P_R \right) + p^\mu \left( R^2_L P_L + R^2_R P_R \right) \right] \ell_\alpha Z_\mu,$$  \hspace{1cm} (A.4)

where $p$ is the $\ell_\beta$ 4-momentum, and

$$\mathcal{L}_{\ell \ell h} = \bar{\ell}_\beta (S_L P_L + S_R P_R) \ell_\alpha h.$$  \hspace{1cm} (A.5)

The general $4\ell$ 4-fermion interaction Lagrangian can be written as

$$\mathcal{L}_{4\ell} = \sum_{I,X,Y=V,T, S,L,R} A^I_{XY} \bar{\ell}_\beta \Gamma_I P_X \ell_\alpha \bar{\ell}_\gamma \Gamma_I P_Y \ell_\gamma + \text{h.c.},$$  \hspace{1cm} (A.6)

where $\ell_{\alpha,\beta,\gamma,\delta}$ denote the lepton flavors and $\Gamma_S = 1$, $\Gamma_V = \gamma_\mu$ and $\Gamma_T = \sigma_{\mu\nu}$. We omit flavor indices in the Wilson coefficients for the sake of clarity. This Lagrangian contains the most general form compatible with Lorentz invariance. The Wilson coefficients $A^V_{LR}$ and $A^S_{RL}$ were included in [57], but absent in [42,58]. As previously stated, the coefficients in Eq.(A.6) do not include photonic contributions, but they include $Z$-boson and scalar ones. Finally, the general $2\ell 2q$ four fermion interaction Lagrangian at the quark level is given by

$$\mathcal{L}_{2\ell 2q} = \mathcal{L}_{2\ell 2d} + \mathcal{L}_{2\ell 2u}$$  \hspace{1cm} (A.7)

where

$$\mathcal{L}_{2\ell 2d} = \sum_{I,X,Y=V,T, S,L,R} B^I_{XY} \bar{d}_\beta \Gamma_I P_X \ell_\alpha \bar{d}_\gamma \Gamma_I P_Y, \text{h.c.}$$  \hspace{1cm} (A.8)

$$\mathcal{L}_{2\ell 2u} = \mathcal{L}_{2\ell 2d|d\rightarrow u, B\rightarrow C}.$$  \hspace{1cm} (A.9)

Here $d_y$ denotes the d-quark flavor.

Let us now consider the Lagrangian relevant for quark flavor violation. This can be written as

$$\mathcal{L}_{\text{QFV}} = \mathcal{L}_{qq\gamma} + \mathcal{L}_{qqg} + \mathcal{L}_{4d} + \mathcal{L}_{2d 2q} + \mathcal{L}_{2d 2u} + \mathcal{L}_{dd H}.$$  \hspace{1cm} (A.10)
The first two terms correspond to operators that couple quark bilinears to massless gauge bosons. These are

\[ \mathcal{L}_{qq\gamma} = e \left[ \bar{d}_\beta \sigma_{\mu\nu} \left( m_{d_\alpha} Q^L_{12} P_L + m_{d_\alpha} Q^R_{12} P_R \right) d_\alpha \right] F_{\mu\nu} \]
\[ \mathcal{L}_{qqg} = g_s \left[ \bar{d}_\beta \sigma_{\mu\nu} \left( m_{d_\alpha} Q^L_{12} P_L + m_{d_\alpha} Q^R_{12} P_R \right) T^a d_\alpha \right] G_{\mu\nu}^a. \]

Here \( T^a \) are \( SU(3) \) matrices. The Wilson coefficients \( Q^{L,R}_{1,2} \) can be easily related to the usual \( C^{\mu\nu}_{\gamma,8} \) coefficients, sometimes normalized with an additional \( \frac{1}{16\pi^2} \) factor. The 4d four fermion interaction Lagrangian can be written as

\[ \mathcal{L}_{4d} = \sum_{I=S,T} \sum_{X,Y=L,R} D_{XY}^I \bar{d}_\beta \Gamma_I P_X d_\alpha \bar{d}_\delta \Gamma_I P_Y d_\gamma + \text{h.c.}, \]

where \( d_\alpha, \beta, \gamma, \delta \) denote the lepton flavors. Again, we omit flavor indices in the Wilson coefficients for the sake of clarity. The 2d2\( \ell \) four fermion interaction Lagrangian is given by

\[ \mathcal{L}_{2d2\ell} = \sum_{I=S,T} \sum_{X,Y=L,R} E_{XY}^I \bar{d}_\beta \Gamma_I P_X d_\alpha \bar{d}_\gamma \Gamma_I P_Y d_\gamma + \text{h.c.}. \]

Here \( \ell_\gamma \) denotes the lepton flavor. \( \mathcal{L}_{2d2\ell} \) should not be confused with \( \mathcal{L}_{2d3\ell} \). In the former case one has QFV operators, whereas in the latter one has LFV operators. This distinction has been made for practical reasons. The 2d2\( \nu \) and \( du\nu \) terms of the QFV Lagrangian are

\[ \mathcal{L}_{2d2\nu} = \sum_{X,Y=L,R} F_{XY}^V \bar{d}_\beta \gamma_\mu P_X d_\alpha \bar{d}_\gamma P_Y \gamma^\mu \gamma + \text{h.c.}, \]
\[ \mathcal{L}_{du\nu} = \sum_{X,Y=L,R} G_{XY}^I \bar{d}_\beta \Gamma_I P_X d_\alpha \bar{d}_\gamma \Gamma_I P_Y + \text{h.c.}. \]

Note that we have not introduced scalar or tensor 2d2\( \nu \) operators, nor tensor \( du\nu \) ones, and that lepton flavor (denoted by the index \( \gamma \)) is conserved in these operators. Finally, we have also included a term in the Lagrangian accounting for operators of the type \((\bar{d}\Gamma d)S\) and \((\bar{d}\Gamma d)P\), where \( S \) (\( P \)) is a virtual \( ^{10} \) scalar (pseudoscalar) state. This piece can be written as

\[ \mathcal{L}_{ddPH} = \bar{d}_\beta \left( H^L_P P_L + H^S_P P_R \right) d_\alpha S + \bar{d}_\beta \left( H^F_P P_L + H^P_P P_R \right) d_\alpha P. \]

### B: Operators available by default in the SPheno output of SARAH

The operators presented in Appendix A have been implemented by using the results of PreSARAH in SARAH. Those are exported to SPheno. We give in the following the list of all internal names for these operators, which can be used in the calculation of new flavor observables.

#### B.1: 2-Fermion-1-Boson operators

These operators are arrays with either two or three elements. While operators involving vector bosons have always dimension \( 3 \times 3 \), those with scalars have dimension \( 3 \times 3 \times n_g \), \( n_g \) is the number of generations of the considered scalar and for \( n_g = 1 \) the last index is dropped.

\( \left( \bar{d}_\beta \sigma_{\mu\nu} \Gamma d_\alpha \right) F_{\mu\nu} \) and \( \left( \bar{d}_\beta \sigma_{\mu\nu} \Gamma d_\alpha \right) G_{\mu\nu} \)

\[ ^{10} \text{We would like to emphasize that our implementation of these operators is only valid for virtual scalars and pseudoscalars. They have been introduced in order to provide the 1-loop vertices necessary for the computation of the double penguin contributions to } \Delta M_{B_d}. \text{ Therefore, they are not valid for observables in which the scalar or pseudoscalar states are real particles.} \]
These operators are derived by PreSARAH with the following input files

**Listing 1 PhotonQQp.m**

```plaintext
NameProcess="Gamma2Q";
ConsideredProcess = "2Fermion1Vector";
FermionOrderExternal={1,2};
NeglectMasses={3};
ExternalFields= {Bar[BottomQuark], BottomQuark, Photon};
CombinationGenerations = {{3,2}};
AllOperators=

{{OA2qSL,Op[7] Pair [ec[3],k[1]]},
 {OA2qSR,Op[6] Pair [ec[3],k[1]]},
 {OA2qVL,Op[7],ec[3}],
 {OA2qVR,Op[6],ec[3]}};
OutputFile = "Gamma2Q.m";
Filters = {};
```

**Listing 2 GluonQQp.m**

```plaintext
NameProcess="Gluon2Q";
ConsideredProcess = "2Fermion1Vector";
FermionOrderExternal={1,2};
NeglectMasses={3};
ExternalFields= {Bar[BottomQuark], BottomQuark, Gluon};
CombinationGenerations = {{3,2}};
AllOperators=

{{OG2qSL,Op[7] Pair [ec[3],k[1]]},
 {OG2qSR,Op[6] Pair [ec[3],k[1]]}};
OutputFile = "Gluon2Q.m";
Filters = {};
```

The normalization is changed to match the standard definitions by

**Listing 3 Photon_wrapper_QFV.m**

```plaintext
ProcessWrapper = True;
NameProcess = "Gamma2Q"
ExternalFields = {Bar[BottomQuark], BottomQuark, Photon};
SumContributionsOperators ["Gamma2Q"] = {
 {CC7, OA2qSL},
 {CC7p, OA2qSR}
};
NormalizationOperators ["Gamma2Q"] ={
 "CC7(2,:) = 0.25_dp∗CC7(2,:)/sqrt (Alpha_160∗4∗Pi)/MFd(2) ",
```

| Variable | Operator | Name | Variable | Operator | Name |
|----------|----------|------|----------|----------|------|
| CC7     | eμdβ(μνPRLdα)Fμν | Q^L | CC8     | gμμdβ(μνPRLdα)Gμν | Q^L |
| CC7p    | eμdβ(μνPRLdα)Fμν | Q^R | CC8p    | gμμdβ(μνPRLdα)Gμν | Q^R |
\[ \ell_\beta \left( q^2 \gamma^\mu + i m_{\ell} \sigma^\mu \nu q_\nu \right) \ell_\alpha A^\mu \]
The normalization is changed to match the standard definitions by

**Listing 6** Photon_wrapper_LFV.m

```
ProcessWrapper = True;
NameProcess = "Gamma2l"
ExternalFields = {bar | ChargedLepton , ChargedLepton , Photon};
SumContributionsOperators ["Gamma2l"] = {
{K1L, OA1L},
{K1R, OA1R},
{K2L, OA2SL},
{K2R, OA2SR}};
NormalizationOperators ["Gamma2l"] = {
"K1L = K1L/sqrt (Alpha_MZ * 4 * Pi)"
"K1R = K1R/sqrt (Alpha_MZ * 4 * Pi)"
"K2L(2,:) = -0.5_dp*K2L(2,:) / sqrt (Alpha_MZ * 4 * Pi)/MFe(2)"
"K2L(3,:) = -0.5_dp*K2L(3,:) / sqrt (Alpha_MZ * 4 * Pi)/MFe(3)"
"K2R(2,:) = -0.5_dp*K2R(2,:) / sqrt (Alpha_MZ * 4 * Pi)/MFe(2)"
"K2R(3,:) = -0.5_dp*K2R(3,:) / sqrt (Alpha_MZ * 4 * Pi)/MFe(3)"
```

In the following we omit flavor indices for the sake of simplicity. These operators are derived by PreSARAH with the following input files

**Listing 7** Z2l.m

```
NameProcess="Z2l";
ConsideredProcess = "2Fermion1Vector";
FermionOrderExternal={1,2};
NeglectMasses={1,2};
ExternalFields= {ChargedLepton , bar | ChargedLepton , Zboson};
CombinationGenerations = {{1,2},{1,3},{2,3}};
AllOperators=
{OZ2lSL,Op[7] Pair | ec [3] , k [1] } , {OZ2lSR,Op[6] Pair | ec [3] , k [1] } ,
{OZ2lVL ,Op[7 , ec [3] ]} , {OZ2lVR ,Op[6 , ec [3] ]} 
OutputFile = "Z2l .m";
```

In the following we omit flavor indices for the sake of simplicity. These operators are derived by PreSARAH with the following input files

**Listing 8** H2l.m

```
NameProcess="H2l";
```

These operators are derived by PreSARAH with the following input files
ConsideredProcess = "2Fermion1Scalar";
FermionOrderExternal = {1, 2};
NeglectMasses = {1, 2};

ExternalFields = {ChargedLepton, bar[ChargedLepton], HiggsBoson};
CombinationGenerations = {{1, 2, ALL}, {1, 3, ALL}, {2, 3, ALL}};

AllOperators = {{OH2qSL, Op[7]},
                {OH2qSR, Op[6]}w}

OutputFile = "H2q.m";

Filters = {};

\[ (\bar{d}\Gamma d) S \text{ and } (\bar{d}\Gamma d) P \]

| Variable Operator Name | Variable Operator Name |
|------------------------|------------------------|
| OH2qSL \( d_P d S H_P^S \) | OH2qSR \( d_P d S H_P^S \) |
| OAh2qSL \( d_P d P H_P^L \) | OAh2qSR \( d_P d P H_P^R \) |

These auxiliary operators are derived by PreSARAH with the following input files

**Listing 9 H2q.m**

NameProcess = "H2q";

\( * \) operators needed for double penguins with internal scalars *
\( * \) we neglect therefore the mass of the scalar in the loop functions *
\( * \) and treat it as massless *

ConsideredProcess = "2Fermion1Scalar";
FermionOrderExternal = {2, 1};
NeglectMasses = {3};

ExternalFields = {DownQuark, bar[DownQuark], HiggsBoson};
CombinationGenerations = {{2, 1, ALL}, {3, 1, ALL}, {3, 2, ALL}};

AllOperators = {{OH2qSL, Op[7]},
                {OH2qSR, Op[6]}w};

OutputFile = "H2q.m";

Filters = {};

**Listing 10 A2q.m**

NameProcess = "A2q";

\( * \) operators needed for double penguins with internal scalars *
\( * \) we neglect therefore the mass of the scalar in the loop functions *
\( * \) and treat it as massless *

ConsideredProcess = "2Fermion1Scalar";
FermionOrderExternal = {2, 1};

\( ^{11} \) The \( (\bar{d}\Gamma d) S \) and \( (\bar{d}\Gamma d) P \) operators have been introduced to compute double penguin corrections to \( \Delta M_{B_s} \), where \( S \) and \( P \) appear as intermediate (virtual) particles. They should not be used in processes where the scalar or pseudoscalar states are real particles because the loop functions are calculated with vanishing external momenta.
NeglectMasses = {3};

ExternalFields = {DownQuark, bar [DownQuark], PseudoScalar};
CombinationGenerations = { {2, 1, ALL}, {3, 1, ALL}, {3, 2, ALL} };

AllOperators = { { OAh2qSL, Op[7] }, { OAh2qSR, Op[6] } };
OutputFile = "A2q.m";
Filters = { };
CombinationGenerations = \{ \{3,1,1,1\}, \{3,1,2,2\}, \{3,1,3,3\}, \\
{3,2,1,1\}, \{3,2,2,2\}, \{3,2,3,3\}\};

AllOperators = \{OddllSLL, Op[7].Op[7]\, ,
{OddllSRR, Op[6].Op[6]\, ,
{OddllSRL, Op[7].Op[6]\, ,
{OddllSLR, Op[7].Op[7]\, ,
\\
{OddllVRR, Op[7].Lor[1].Op[7].Lor[1]]\, ,
{OddllVLL, Op[6].Lor[1].Op[6].Lor[1]]\, ,
{OddllVRL, Op[7].Lor[1].Op[6].Lor[1]]\, ,
{OddllVLR, Op[6].Lor[1].Op[7].Lor[1]]\, ,
\\
{OddllTLL, Op[7].Lor[2].Op[7].Lor[1].Lor[2]]\, ,
{OddllTLR, Op[7].Lor[1].Lor[2].Op[6].Lor[1].Lor[2]]\, ,
{OddllTRL, Op[6].Lor[1].Lor[2].Op[7].Lor[1].Lor[2]]\, ,
{OddllTRR, Op[6].Lor[2].Op[7].Lor[1].Lor[2]]\}
};

\begin{equation}
(\bar{\ell}\Gamma\ell)(d\bar{\Gamma}d') \text{ and } (\bar{\ell}\Gamma\ell)(u\bar{\Gamma}u')
\end{equation}

\begin{tabular}{|c|c|c|c|}
\hline
Variable & Operator & Name & Variable & Operator & Name \\
\hline
OllldSLL & $\bar{\ell}\Gamma\ell$(dP_Ld) & $B^7_{LL}$ & OllhuSLL & $(\bar{\ell}\Gamma\ell)(\bar{u}P_Lu)$ & $C^7_{LL}$ \\
OllldSRR & $\bar{\ell}\Gamma\ell$(dP_Rd) & $B^7_{RR}$ & OllhuSRR & $(\bar{\ell}\Gamma\ell)(\bar{u}P_Ru)$ & $C^7_{RR}$ \\
OllldSRL & $\bar{\ell}\Gamma\ell$(dP_Ld) & $B^7_{LR}$ & OllhuSRL & $(\bar{\ell}\Gamma\ell)(\bar{u}P_Lu)$ & $C^7_{LR}$ \\
OllldSLR & $\bar{\ell}\Gamma\ell$(dP_Rd) & $B^7_{RL}$ & OllhuSLR & $(\bar{\ell}\Gamma\ell)(\bar{u}P_Ru)$ & $C^7_{RL}$ \\
OllldVLL & $(\bar{\ell}\mu\nu P_L\ell)(d\gamma^\mu P_Ld)$ & $B^7_{LL}$ & OllhuVLL & $(\bar{\ell}\gamma P_L\ell)(\bar{u}\gamma^\mu P_Lu)$ & $C^7_{LL}$ \\
OllldVRR & $(\bar{\ell}\mu P_R\ell)(d\gamma^\mu P_Rd)$ & $B^7_{RR}$ & OllhuVRR & $(\bar{\ell}\gamma P_R\ell)(\bar{u}\gamma^\mu P_Ru)$ & $C^7_{RR}$ \\
OllldVLR & $(\bar{\ell}\mu P_L\ell)(d\gamma^\mu P_Ld)$ & $B^7_{LR}$ & OllhuVLR & $(\bar{\ell}\gamma P_L\ell)(\bar{u}\gamma^\mu P_Lu)$ & $C^7_{LR}$ \\
OllldVRL & $(\bar{\ell}\mu P_R\ell)(d\gamma^\mu P_Rd)$ & $B^7_{RL}$ & OllhuVRL & $(\bar{\ell}\gamma P_R\ell)(\bar{u}\gamma^\mu P_Ru)$ & $C^7_{RL}$ \\
OllldTLL & $(\bar{\ell}\gamma P_L\ell)(d\bar{\gamma} P_Ld)$ & $B^7_{LL}$ & OllhuTLL & $(\bar{\ell}\sigma P_L\ell)(\bar{u}\sigma^{\mu\nu} P_Lu)$ & $C^7_{LL}$ \\
OllldTRR & $(\bar{\ell}\gamma P_R\ell)(d\bar{\gamma} P_Rd)$ & $B^7_{RR}$ & OllhuTRR & $(\bar{\ell}\sigma P_R\ell)(\bar{u}\sigma^{\mu\nu} P_Ru)$ & $C^7_{RR}$ \\
OllldTRR & $(\bar{\ell}\gamma P_R\ell)(d\bar{\gamma} P_Rd)$ & $B^7_{RL}$ & OllhuTRR & $(\bar{\ell}\sigma P_R\ell)(\bar{u}\sigma^{\mu\nu} P_Ru)$ & $C^7_{RL}$ \\
OllldTRL & $(\bar{\ell}\gamma P_L\ell)(d\bar{\gamma} P_Ld)$ & $B^7_{RL}$ & OllhuTRL & $(\bar{\ell}\sigma P_L\ell)(\bar{u}\sigma^{\mu\nu} P_Lu)$ & $C^7_{RL}$ \\
\hline
\end{tabular}

Listing 13 2L2d.m

```
NameProcess = "2L2d";
```

Considered Process = "4Fermion";
Fermion Order External = \{2,1,4,3\};
Neglect Masses = \{1,2,3,4\};

External Fields = \{Charged Lepton, \bar{Charged Lepton}, Down Quark, \bar{Down Quark}\};
Combination Generations = \{\{2,1,1,1\}, \{3,1,1,1\}, \{3,2,1,1\},
\{2,1,2,2\}, \{3,1,2,2\}, \{3,2,2,2\}\};

All Operators = \{
\{OlluddSLL, Op[7]. Op[7]\},
\{OlluddSRR, Op[6]. Op[6]\},
\{OlluddSLR, Op[7]. Op[6]\},
\{OlluddVRR, Op[7, Lor[1]]. Op[7, Lor[1]]\},
\{OlluddVLL, Op[6, Lor[1]]. Op[6, Lor[1]]\},
\{OlluddVLR, Op[6]. Op[7]\},
\{OlluddVRL, Op[7]. Op[6]\},
\{OlluddTLL, Op[−7, Lor[1]]. Op[−7, Lor[1]]\},
\{OlluddTLR, Op[−7, Lor[1]]. Op[−6, Lor[1]]\},
\{OlluddTRL, Op[−6, Lor[1]]. Op[−7, Lor[1]]\},
\{OlluddTRR, Op[−6, Lor[1]]. Op[−6, Lor[1]]\}
};

Listing 14 2L2u.m

Name Process = "2L2u";

Considered Process = "4Fermion";
Fermion Order External = \{2,1,4,3\};
Neglect Masses = \{1,2,3,4\};

External Fields = \{Charged Lepton, \bar{Charged Lepton}, Up Quark, \bar{Up Quark}\};
Combination Generations = \{\{2,1,1,1\}, \{3,1,1,1\}, \{3,2,1,1\}\};

All Operators = \{
\{OlluuSLL, Op[7]. Op[7]\},
\{OlluuSRR, Op[6]. Op[6]\},
\{OlluuSLR, Op[7]. Op[6]\},
\{OlluuVRR, Op[7, Lor[1]]. Op[7, Lor[1]]\},
\{OlluuVLL, Op[6, Lor[1]]. Op[6, Lor[1]]\},
\{OlluuVLR, Op[6]. Op[7]\},
\{OlluuVRL, Op[7]. Op[6]\},
\{OlluuTLL, Op[−7, Lor[1]]. Op[−7, Lor[1]]\},
\{OlluuTLR, Op[−7, Lor[1]]. Op[−6, Lor[1]]\},
\{OlluuTRL, Op[−6, Lor[1]]. Op[−7, Lor[1]]\},
\{OlluuTRR, Op[−6, Lor[1]]. Op[−6, Lor[1]]\}
};

\((\bar{d} \Gamma d)(\bar{d} \Gamma' d)\) and \((\bar{\ell} \Gamma \ell)(\bar{\ell} \Gamma' \ell)\)
\[
\text{ConsideredProcess} = \text{"4Fermion"}; \\
\text{FermionOrderExternal} = \{2,1,4,3\}; \\
\text{NeglectMasses} = \{1,2,3,4\}; \\
\text{ExternaFields} = \{\text{DownQuark}, \bar{\text{DownQuark}}, \text{DownQuark}, \bar{\text{DownQuark}}\}; \\
\text{FermionOrderExternal} = \{2,1,4,3\}; \\
\text{CombinationGenerations} = \{\{3,1,3,1\}, \{3,2,3,2\}, \{2,1,2,1\}\}; \\
\text{ColorFlow} = \text{ColorDelta}[1,2], \text{ColorDelta}[3,4]; \\
\text{CombinationGenerations} = \{\{2,1,1,1\}, \{3,1,1,1\}, \{3,2,2,2\}\}; \\
\text{AllOperators} = \{\{4d\text{SLL}, \text{Op}[7], \text{Op}[7]\}, \\
\{4d\text{SRR}, \text{Op}[6], \text{Op}[6]\}, \\
\{4d\text{SLR}, \text{Op}[6], \text{Op}[7]\}, \\
\{4d\text{SRL}, \text{Op}[7], \text{Op}[6]\}, \\
\{4d\text{VLL}, \text{Op}[7], \text{Lor}[1], \text{Op}[6], \text{Lor}[1]\}, \\
\{4d\text{VLR}, \text{Op}[7], \text{Lor}[1], \text{Op}[6], \text{Lor}[1]\}, \\
\{4d\text{VRR}, \text{Op}[7], \text{Lor}[1], \text{Op}[6], \text{Lor}[1]\}, \\
\{4d\text{VRL}, \text{Op}[7], \text{Lor}[1], \text{Op}[6], \text{Lor}[1]\}, \\
\{4d\text{TLL}, \text{Op}[7], \text{Lor}[1], \text{Op}[7], \text{Lor}[1]\}, \\
\{4d\text{TLR}, \text{Op}[7], \text{Lor}[1], \text{Op}[7], \text{Lor}[1]\}, \\
\{4d\text{TRL}, \text{Op}[7], \text{Lor}[1], \text{Op}[7], \text{Lor}[1]\}, \\
\{4d\text{TRR}, \text{Op}[7], \text{Lor}[1], \text{Op}[7], \text{Lor}[1]\}\}; \\
\text{Filters} = \{\text{NoPenguins}\}; \\
\]
\[ (\bar{d}^\mu P_L u)(\bar{\ell}^\mu P_L \nu) \]

| Variable | Operator | Name | Variable | Operator | Name |
|----------|----------|------|----------|----------|------|
| OdulvVLL | \((\bar{\gamma} \mu P_L u)(\bar{\ell}^\mu P_L \nu)\) | \(G_{LL}^V\) | OdulvSLL | \((\bar{d}^\mu P_L u)(\bar{\ell}^\mu P_L \nu)\) | \(G_{LL}^S\) |
| OdulvVRR | \((\bar{\gamma} \mu P_R u)(\bar{\ell}^\mu P_R \nu)\) | \(G_{RR}^V\) | OdulvSRR | \((\bar{d}^\mu P_R u)(\bar{\ell}^\mu P_R \nu)\) | \(G_{RR}^S\) |
| OdulvVLR | \((\bar{\gamma} \mu P_L u)(\bar{\ell}^\mu P_R \nu)\) | \(G_{LR}^V\) | OdulvSLR | \((\bar{d}^\mu P_L u)(\bar{\ell}^\mu P_R \nu)\) | \(G_{LR}^S\) |
| OdulvVRL | \((\bar{\gamma} \mu P_R u)(\bar{\ell}^\mu P_L \nu)\) | \(G_{RL}^V\) | OdulvSRL | \((\bar{d}^\mu P_R u)(\bar{\ell}^\mu P_L \nu)\) | \(G_{RL}^S\) |

Listing 17 du\_lv.m

NameProcess = "dulv";
ConsideredProcess = "4Fermion";
FermionOrderExternal = \(\{2,1,3,4\}\);
NeglectMasses = \(\{1,2,3,4\}\);

ExternalFields = \{DownQuark, bar[UpQuark], Neutrino, bar[ChargedLepton]\};

CombinationGenerations = Flatten[Table[\{\{3,1,i,j\},\{3,2,i,j\},\{2,2,i,j\},\{2,1,i,j\}\},\{i,1,3\},\{j,1,3\}\},2];
Clear[i,j];

AllOperators = \{\{OdulvSLL, Op[7] . Op[7]\},
\{OdulvSRR, Op[6] . Op[6]\},
\{OdulvVLR, Op[7] . Op[1]\},
\{OdulvVRL, Op[6] . Op[1]\},
\{OdulvVRR, Op[7] . Lor[1] . Op[7] . Lor[1]\},
\{OdulvVLL, Op[6] . Lor[1] . Op[6] . Lor[1]\},
\{OdulvVLR, Op[6] . Lor[1] . Op[7] . Lor[1]\},
\{OdulvVRL, Op[7] . Lor[1] . Op[6] . Lor[1]\}
\};

Filters = \{NoCrossedDiagrams\};

C: Application: Flavor observables implemented in S\textsc{arah}

C.1: Lepton flavor observables

Lepton flavor violation in the SM or MSSM without neutrino masses vanishes exactly. Even adding Dirac neutrino masses to the SM predicts LFV rates which are far beyond the experimental reach. However, many extensions of the SM can introduce new sources for LFV of a size which is testable nowadays. The best-known
examples are SUSY and non-SUSY models with high- or low-scale seesaw mechanism, models with vector-like leptons and SUSY models with $R$-parity violation, see for instance Refs. [32, 42, 58–89].

We discuss in the following the implementation of the most important LFV observables in SARAH and SPheno using the previously defined operators which are calculated by SPheno.

C.1.1: $\ell_\alpha \rightarrow \ell_\beta \gamma$

The decay width is given by [42]

$$\Gamma (\ell_\alpha \rightarrow \ell_\beta \gamma) = \frac{\alpha m_\ell^5}{4} \left( |K_{L}\rangle^2 + |K_{R}\rangle^2 \right), \quad (C.18)$$

where $\alpha$ is the fine structure constant and the dipole Wilson coefficients $K_{L,R}$ are defined in Eq.(A.3).

Listing 18  LLgGamma.m

```plaintext
NameProcess = "LLpGamma";
NameObservables = {{muEgamma, 701, "BR(mu->e gamma)",
{tauEgamma, 702, "BR(tau->e gamma)",
{tauMuGamma, 703, "BR(tau->mu gamma)"}};
NeededOperators = {K2L, K2R};
Body = "LLpGamma.f90";
```

Listing 19  LLgGamma.f90

```plaintext
Real(dp) :: width
Integer :: i1 , gt1 , gt2

Do i1=1,3
If ( i1 .eq.1) Then ! mu -> e gamma
 gt1 = 2
 gt2 = 1
Elseif ( i1 .eq.2) Then ! tau -> e gamma
 gt1 = 3
 gt2 = 1
Else ! tau -> mu gamma
 gt1 = 3
 gt2 = 2
End if
End do
```

C.1.2: $\ell_\alpha \rightarrow 3 \ell_\beta$

The decay width is given by

$$\Gamma (\ell_\alpha \rightarrow 3 \ell_\beta) = \frac{m_\ell^5}{512 \pi^3} \left[ e^4 \left( |K_L^2|^2 + |K_R^2|^2 \right) \left( \frac{16}{3} \log \frac{m_\ell}{m_\beta} - \frac{22}{3} \right) \right]$$

(C.19)

$$+ \frac{1}{24} \left( |A_{LL}^S|^2 + |A_{RR}^S|^2 \right) + \frac{1}{12} \left( |A_{LR}^S|^2 + |A_{RL}^S|^2 \right)$$

$$+ \frac{2}{3} \left( |A_{LL}^V|^2 + |A_{RR}^V|^2 \right) + \frac{1}{3} \left( |A_{LR}^V|^2 + |A_{RL}^V|^2 \right) + 6 \left( |A_{LL}^T|^2 + |A_{RR}^T|^2 \right)$$

$$+ \frac{e^2}{3} \left( K_L A_{RR}^T + K_R A_{LL}^T + c.c. \right) - \frac{2e^2}{3} \left( K_L A_{RR}^V + K_R A_{LL}^V + c.c. \right)$$

$$- \frac{4e^2}{3} \left( K_L A_{RL}^V + K_R A_{LR}^V + c.c. \right)$$

$$- \frac{1}{2} \left( A_{LL}^T A_{RR}^T + A_{RR}^T A_{LL}^T + c.c. \right) - \frac{1}{6} \left( A_{LL}^V A_{RL}^V + A_{RR}^V A_{LR}^V + c.c. \right) .$$

Here we have defined

$$\hat{A}_{XY} = A_{XY} + e^2 K_{1}^{X} \quad (X, Y = L, R).$$

(C.20)

The mass of the leptons in the final state has been neglected in this formula, with the exception of the dipole terms $K_{L,R}^2$, where an infrared divergence would otherwise occur due to the massless photon propagator.

Eq.(C.19) is in agreement with [58], but also includes the coefficients $A_{LL}^S$ and $A_{RR}^S$.

---

**Listing 20 Lto3Lp.m**

```plaintext
NameProcess = "Lto3Lp";
NameObservables = {{BRmuTo3e, 901, "BR(mu→3e)",
{BRtauTo3e, 902, "BR(tau→3e)",
{BRtauTo3mu, 903, "BR(tau→3mu)"}
};
ExternalStates = {Electron};
NeededOperators = {K1L, K1R, K2L, K2R,
O4lSLL, O4lSRR, O4lSRL, O4lSLR,
O4lVRR, O4lVLL, O4lVRL, O4lVLR,
O4lTLL, O4lTRR};
Body = "Lto3Lp. f90 ";
```

---

**Listing 21 Lto3Lp.f90**

```plaintext
Complex(dp) :: cK1L, cK1R, cK2L, cK2R
Complex(dp) :: CSLL, CSRR, CSLR, CSL, CVLL, &
& CVRR, CVLR, CVRL, CTL, CTRR
Real(dp) :: BRdipole, BRscalar, BRvector, BRtensor
Real(dp) :: BRmix1, BRmix2, BRmix3, BRmix4, GammaLFV
Real(dp) :: e2, e4
Integer :: i1, gt1, gt2, gt3, gt4
```

---
Elseif (i1 .eq. 2) Then
  gt1 = 3
  gt2 = 1
Else
  gt1 = 3
  gt2 = 2
End if

cK1L = K1L(gt1 , gt2)
cK1R = K1R(gt1 , gt2)
cK2L = K2L(gt1 , gt2)
cK2R = K2R(gt1 , gt2)

CSLL = O4lSLL(gt1 , gt2 , gt3 , gt4)
CSRR = O4lSRR(gt1 , gt2 , gt3 , gt4)
CSLR = O4lSLR(gt1 , gt2 , gt3 , gt4)
CSRL = O4lSLR(gt1 , gt2 , gt3 , gt4)

CSLL = O4lSLL(gt1 , gt2 , gt3 , gt4)
CSRR = O4lSRR(gt1 , gt2 , gt3 , gt4)

CVLL = CVLL + e2∗cK1L
CVRR = CVRR + e2∗cK1R
CVLR = CVLR + e2∗cK1L
CVRL = CVRL + e2∗cK1R

CTLL = O4lTLL(gt1 , gt2 , gt3 , gt4)
CTRR = O4lTRR(gt1 , gt2 , gt3 , gt4)

! Photonic dipole contributions
BRdipole = (Abs(cK2L)**2 + Abs(cK2R)**2) & & *(16._dp∗Log(mf_l(gt1)/mf_l(gt2))−22._dp)/3._dp

! Scalar contributions
BRscalar = (Abs(CSLL)**2 + Abs(CSRR)**2) / 24._dp & & *(Abs(CSLR)**2 + Abs(CSRL)**2) / 12._dp

! Vector contributions
BRvector = 2._dp*(Abs(CVLL)**2 + Abs(CVRR)**2) / 3._dp & & *(Abs(CVLR)**2 + Abs(CVRL)**2) / 3._dp

! Tensor contributions
BRtensor = 6._dp*(Abs(CTLL)**2 + Abs(CTRR)**2)

! Mix: dipole x scalar
BRmix1 = 2._dp/3._dp*Real(cK2L∗Conjg(CSRL) + cK2R∗Conjg(CSRL), dp)

! Mix: dipole x vector
BRmix2 = −4._dp/3._dp*Real(cK2L∗Conjg(CVRL) + cK2R∗Conjg(CVRL), dp) & & −8._dp/3._dp*Real(cK2L∗Conjg(CVRR) + cK2R∗Conjg(CVLL), dp)

! Mix: scalar x vector
BRmix3 = −1._dp/3._dp*Real(CSLR∗Conjg(CVLR) + CSRL∗Conjg(CVLR), dp)

! Mix: scalar x tensor
BRmix4 = −1._dp*Real(CSLL∗Conjg(CTLL) + CSRR∗Conjg(CTRR), dp)

GammaLFV = oo512pi3∗mf_l(gt1)**5 & & *(e4∗BRdipole + BRscalar + BRvector + BRtensor & & + e2∗BRmix1 + e2∗BRmix2 + BRmix3 + BRmix4)

! taking alpha(Q=0) instead of alpha(m_Z) as this contains most of the ! running of the Wilson coefficients
C.1.3: Coherent $\mu - e$ conversion in nuclei

The conversion rate, relative to the the muon capture rate, can be expressed as [90,91]

$$\text{CR}(\mu - e, \text{Nucleus}) = \frac{p_e E_e m_{\mu}^3 G_F^2}{8 \pi^2 Z} \alpha^3 Z_{\text{eff}}^2 F_p^2 \times \left\{ \begin{array}{l} (Z + N) \left(g_{LV}^{(0)} + g_{LS}^{(0)}\right) + (Z - N) \left(g_{LV}^{(1)} + g_{LS}^{(1)}\right)^2 + \\ (Z + N) \left(g_{RV}^{(0)} + g_{RS}^{(0)}\right) + (Z - N) \left(g_{RV}^{(1)} + g_{RS}^{(1)}\right)^2 \end{array} \right\} \frac{1}{\Gamma_{\text{capt}}}.$$  \hspace{1cm} (C.21)

$Z$ and $N$ are the number of protons and neutrons in the nucleus and $Z_{\text{eff}}$ is the effective atomic charge [92]. Similarly, $G_F$ is the Fermi constant, $F_p$ is the nuclear matrix element and $\Gamma_{\text{capt}}$ represents the total muon capture rate. $\alpha$ is the fine structure constant, $p_e$ and $E_e$ ($\approx m_{\mu}$ in the numerical evaluation) are the momentum and energy of the electron and $m_{\mu}$ is the muon mass. In the above, $g_{XK}^{(0)}$ and $g_{XK}^{(1)}$ (with $X = L, R$ and $K = S, V$) can be written in terms of effective couplings at the quark level as

$$g_{XK}^{(0)} = \frac{1}{2} \sum_{q=u,d,s} \left( g_{XK(q)}(q) G_K^{(q,p)} + g_{XK(q)}(q) G_K^{(q,n)} \right) ,$$

$$g_{XK}^{(1)} = \frac{1}{2} \sum_{q=u,d,s} \left( g_{XK(q)}(q) G_K^{(p)} - g_{XK(q)}(q) G_K^{(n)} \right).$$  \hspace{1cm} (C.22)

For coherent $\mu - e$ conversion in nuclei, only scalar ($S$) and vector ($V$) couplings contribute. Furthermore, sizable contributions are expected only from the $u, d, s$ quark flavors. The numerical values of the relevant $G_K$ factors are [90,93]

$$G_V^{(u,p)} = G_V^{(d,n)} = 2 ; \quad G_V^{(d,p)} = G_V^{(u,n)} = 1 ;$$

$$G_S^{(u,p)} = G_S^{(d,n)} = 5.1 ; \quad G_S^{(d,p)} = G_S^{(u,n)} = 4.3 ;$$

$$G_S^{(s,p)} = G_S^{(s,n)} = 2.5. \hspace{1cm} \text{(C.23)}$$

Finally, the $g_{XK(q)}$ coefficients can be written in terms of the Wilson coefficients in Eqs.(A.3), (A.8) and (A.9) as

$$g_{LV(q)} = \frac{\sqrt{2}}{G_F} \left[ \frac{\alpha^2}{2} Q_q \left( K_1^L - K_2^R \right) - \frac{1}{2} \left( C_{LL}^{V} + C_{LR}^{V} \right) \right] \hspace{1cm} \text{(C.24)}$$

$$g_{RV(q)} = g_{LV(q)} |_{L \rightarrow R} \hspace{1cm} \text{(C.25)}$$

$$g_{LS(q)} = -\frac{\sqrt{2}}{G_F} \frac{1}{2} \left( C_{LL}^{S} + C_{LR}^{S} \right) \hspace{1cm} \text{(C.26)}$$

$$g_{RS(q)} = g_{LS(q)} |_{L \rightarrow R} \hspace{1cm} \text{(C.27)}$$

Here $Q_q$ is the quark electric charge ($Q_d = -1/3$, $Q_u = 2/3$) and $C_{LL}^{IXK} = B_K^{XY} (C_{XY}^{K})$ for d-quarks (u-quarks), with $X = L, R$ and $K = S, V$. 

---

100 End If
Listing 22 MuEconversion.m

```
NameProcess = "MuEconversion";
NameObservables = {{CRmuEAl, 800, "CR(mu−e, Al)"},
                   {CRmuETi, 801, "CR(mu−e, Ti)"},
                   {CRmuESr, 802, "CR(mu−e, Sr)"},
                   {CRmuESb, 803, "CR(mu−e, Sb)"},
                   {CRmuEAu, 804, "CR(mu−e, Au)"},
                   {CRmuEPb, 805, "CR(mu−e, Pb)"}};

NeededOperators = {K1L, K1R, K2L, K2R,
                   OllddSLL, OllddSRR, OllddSRL, OllddSLR, OllddVLL, OllddVRR,
                   OllddVLR, OllddVLR, OllddTLL, OllddTLR, OllddTRL, OllddTRR,
                   OlluuSLL, OlluuSRR, OlluuSRL, OlluuSLR, OlluuVLL, OlluuVRR,
                   OlluuVLR, OlluuVLR, OlluuTLL, OlluuTLL, OlluuTRL, OlluuTLL, OlluuTTRR};

Body = "MuEconversion.f90 ";
```

Listing 23 MuEconversion.f90

```
Complex(dp) :: gPLV(3), gPRV(3)
Complex(dp), Parameter :: mat0(3,3)=0._dp
Real(dp) :: Znuc ,Nnuc, Zeff , Fp, GammaCapt, GSp(3), GSn(3), &
           & Gn(3), Gv(3), e2
Complex(dp) :: Lcont ,Rcont ,gLs(3), gRV(3), gLV(3), g0LV, g0RS, &
             & g0LV, g0RV, g1LS, g1RS, g1LV, g1RV
Integer :: i1, i2

! Coherent mu−e conversion in nuclei
! Observable implemented by W. Porod, F. Staub and A. Vicente
! Based on Y. Kuno, Y. Okada, Rev. Mod. Phys. 73 (2001) 151 [hep−ph/9909265]
! and E. Arganda et al, JHEP 0710 (2007) 104 [arXiv:0707.2955]

e2 = 4._dp*Pi*Alpha_MZ
!
! 1: uu
! 2: dd
! 3: ss

! vector couplings

gLV(1) = 0.5_dp*(OlluuVLL(2,1,1,1) + OlluuVLR(2,1,1,1))
gLV(2) = 0.5_dp*(OlluuVLL(2,1,1,1) + OlluuVLR(2,1,1,1))
gLV(3) = 0.5_dp*(OlluddVLL(2,1,1,1) + OlluddVLR(2,1,1,1))
gRV(1) = 0.5_dp*(OlluuVLL(2,1,1,1) + OlluuVLR(2,1,1,1))
gRV(2) = 0.5_dp*(OlluddVLL(2,1,1,1) + OlluddVLR(2,1,1,1))
gRV(3) = 0.5_dp*(OlluddVLL(2,1,1,1) + OlluddVLR(2,1,1,1))

gLV = gLV + gLV

gRV = gRV + gRV
```

scalar couplings

\[ g_{LS}(1) = 0.5 \times (OlluuSLL(2,1,1,1) + OlluuSLR(2,1,1,1)) \]
\[ g_{RS}(1) = 0.5 \times (OlluuSRL(2,1,1,1) + OlluuSRR(2,1,1,1)) \]
\[ g_{LS}(2) = 0.5 \times (OlllddSLL(2,1,1,1) + OlllddSLR(2,1,1,1)) \]
\[ g_{RS}(2) = 0.5 \times (OlllddSRL(2,1,1,1) + OlllddSRR(2,1,1,1)) \]
\[ g_{LS}(3) = 0.5 \times (OlllddSLL(2,1,2,2) + OlllddSLR(2,1,2,2)) \]
\[ g_{RS}(3) = 0.5 \times (OlllddSRL(2,1,2,2) + OlllddSRR(2,1,2,2)) \]

\[ g_{LS} = -g_{LS} \times \sqrt{2} \times G_F / G_F \]
\[ g_{RS} = -g_{RS} \times \sqrt{2} \times G_F / G_F \]

Do i1=1,6
   If (i1.eq.1) Then
      Znuc=13._dp
      Nnuc=14._dp
      Zeff=11.5_dp
      Fp=0.64_dp
      Gamma_Capt=4.64079e-19_dp
   Else If (i1.eq.2) Then
      Znuc=22._dp
      Nnuc=26._dp
      Zeff=17.6_dp
      Fp=0.54_dp
      Gamma_Capt=1.70422e-18_dp
   Else If (i1.eq.3) Then
      Znuc=38._dp
      Nnuc=42._dp
      Zeff=25.0_dp
      Fp=0.39_dp
      Gamma_Capt=4.61842e-18_dp
   Else If (i1.eq.4) Then
      Znuc=51._dp
      Nnuc=70._dp
      Zeff=29.0_dp
      Fp=0.32_dp
      Gamma_Capt=6.71711e-18_dp
   Else If (i1.eq.5) Then
      Znuc=79._dp
      Nnuc=118._dp
      Zeff=33.5_dp
      Fp=0.16_dp
      Gamma_Capt=8.59868e-18_dp
   Else If (i1.eq.6) Then
      Znuc=82._dp
      Nnuc=125._dp
      Zeff=34.0_dp
      Fp=0.15_dp
      Gamma_Capt=8.84868e-18_dp
   End If
End Do

! numerical values

! based on Y. Kuno, Y. Okada, Rev. Mod. Phys. 73 (2001) 151 [hep-ph/9909265]
! and T. S. Kosmas et al, PLB 511 (2001) 203 [hep-ph/0102101]

GSp=(/5.1,4.3,2.5/)
GSn=(/4.3,5.1,2.5/)
GVp=(/2.0,1.0,0.0/)
GVn=(/1.0,2.0,0.0/)

Do i2=1,3
C.1.4: \( \tau \rightarrow P\ell \)

Our analytical expressions for \( \tau \rightarrow P\ell \), where \( \ell = e, \mu \) and \( P \) is a pseudoscalar meson, generalize the results in [94]. The decay width is given by

\[
\Gamma (\tau \rightarrow \ell P) = \frac{1}{4\pi} \frac{\lambda^{1/2}(m_{\tau}, m_{\ell}, m_P)}{m_{\tau}} \sum_{i, f} |M_{\tau\ell P}|^2, \tag{C.28}
\]

where the averaged squared amplitude can be written as

\[
\frac{1}{2} \sum_{i,f} |M_{\tau\ell P}|^2 = \frac{1}{4m_{\tau}} \sum_{i,j,S,V} \left[ 2m_{\ell} m_{\tau} \left( a_P a_P^{*} + b_P b_P^{*} \right) + (m_{\ell}^2 + m_{\tau}^2 - m_P^2) \left( a_P a_P^{*} + b_P b_P^{*} \right) \right]. \tag{C.29}
\]

The coefficients \( a_P^{S,V} \) and \( b_P^{S,V} \) can be expressed in terms of the Wilson coefficients in Eqs. (A.8) and (A.9) as

\[
a_P^S = \frac{1}{2} f_\pi \sum_{X=L,R} \left[ \frac{D_X^S(P)}{m_d} \left( B_{\ell X}^S + B_{R X}^S \right) + \frac{D_X^S(P)}{m_u} \left( C_{\ell X}^S + C_{R X}^S \right) \right], \tag{C.30}
\]

\[
b_P^S = \frac{1}{2} f_\pi \sum_{X=L,R} \left[ \frac{D_X^S(P)}{m_d} \left( B_{R X}^S - B_{\ell X}^S \right) + \frac{D_X^S(P)}{m_u} \left( C_{R X}^S - C_{\ell X}^S \right) \right], \tag{C.31}
\]

\[
a_P^V = \frac{1}{4} f_\pi C(P)(m_{\tau} - m_{\ell}) \left[ -B_{LL}^V + B_{LR}^V + B_{RL}^V + B_{RR}^V + C_{LL}^V - C_{LR}^V + C_{RL}^V - C_{RR}^V \right], \tag{C.32}
\]

\[
b_P^V = \frac{1}{4} f_\pi C(P)(m_{\tau} + m_{\ell}) \left[ -B_{LL}^V + B_{LR}^V + B_{RL}^V - B_{RR}^V + C_{LL}^V - C_{LR}^V - C_{RL}^V + C_{RR}^V \right]. \tag{C.33}
\]
In these expressions \( m_d \) and \( m_u \) are the down- and up-quark masses, respectively, \( f_\pi \) is the pion decay constant and the coefficients \( C(P), D^{d,u}_{L,R}(P) \) take different forms for each pseudoscalar meson \( P \) \cite{94}. For \( P = \pi \) one has

\[
C(\pi) = 1 \\
D^d_L(\pi) = -\frac{m^2_\pi}{4} \\
D^u_L(\pi) = \frac{m^2_\pi}{4},
\]

for \( P = \eta \)

\[
C(\eta) = \frac{1}{\sqrt{6}} \left( \sin \theta_\eta + \sqrt{2} \cos \theta_\eta \right) \\
D^d_L(\eta) = \frac{1}{4\sqrt{3}} \left[ (3m^2_\pi - 4m^2_K) \cos \theta_\eta - 2\sqrt{2}m^2_K \sin \theta_\eta \right] \\
D^u_L(\eta) = \frac{1}{4\sqrt{3}} m^2_\pi \left( \cos \theta_\eta - \sqrt{2} \sin \theta_\eta \right),
\]

and for \( P = \eta' \)

\[
C(\eta') = \frac{1}{\sqrt{6}} \left( \sqrt{2} \sin \theta_\eta - \cos \theta_\eta \right) \\
D^d_L(\eta') = \frac{1}{4\sqrt{3}} \left[ (3m^2_\pi - 4m^2_K) \sin \theta_\eta + 2\sqrt{2}m^2_K \cos \theta_\eta \right] \\
D^u_L(\eta') = \frac{1}{4\sqrt{3}} m^2_\pi \left( \sin \theta_\eta + \sqrt{2} \cos \theta_\eta \right).
\]

Here \( m_\pi \) and \( m_K \) are the masses of the neutral pion and Kaon, respectively, and \( \theta_\eta \) is the \( \eta - \eta' \) mixing angle.

In addition, \( D^{d,u}_{R}(P) = -\left( D^{d,u}_{L}(P) \right)^* \).

Notice that the Wilson coefficients in Eq.(C.33) include all pseudoscalar and axial contributions to \( \tau \rightarrow \ell P \). Therefore, this goes beyond some well-known results in the literature, see for example \cite{94,95}, where box contributions were neglected.

#### Listing 24 TauLMeson.m

```plaintext
NameProcess = "TauLMeson";
NameObservables = {{{BrTautoEPi , 2001, "BR(tau->e pi)",
    (BrTautoEEta , 2002, "BR(tau->e eta)")},
    (BrTautoEEtap, 2003, "BR(tau->e eta')"),
    (BrTautoMuPi, 2004, "BR(tau->mu pi)")},
    (BrTautoMuEta, 2005, "BR(tau->mu eta)")},
    (BrTautoMuEtap , 2006, "BR(tau->mu eta')");
NeededOperators = {OllddSLL, OllddSRR, OllddSRL, OllddSLR,
    OlluuSLL, OlluuSRR, OlluuSRL, OlluuSLR,
    OlluuVRR, OlluuVLL, OlluuVRL, OlluuVLR,
    OlluuSLL, OlluuSRR, OlluuSRL, OlluuSLR,
    OlluuVRR, OlluuVLL, OlluuVRL, OlluuVLR};
Body = "\text{TauLMeson.f90}";
```

#### Listing 25 TauLMeson.f90

```plaintext
Real(dp) :: Fpi , thetaEta , mPi, mK, mEta, mEtap, meson_abs_T2, cont , &
    mP, CP, factor , BR
Complex(dp) :: BSLL, BSLR, BSRL, BSRR, BLL, BLR, BVR, BVRR, &
    CSLL, CSLR, CSRL, CSRR, CVLL, CVLR, CVRL, CVRR, aP(2), bP(2), &
    DLdP, DRdP, DLuP, DRuP
Integer :: i1 , i2 , out , k1, k2
```

"
! tau \rightarrow 1 \text{ meson}

! Observable implemented by W. Porod, F. Staub and A. Vicente

! Generalizes the analytical expressions in

! E. Arganda et al., JHEP 0806 (2008) 079 [arXiv:0803.2039]

F_{\pi}=0.0924 \text{ dp} \quad \text{Pion decay constant in GeV}

\theta_\eta=\pi/10. \text{ dp} \quad \eta-\eta' \text{ mixing angle}

m_{\pi}=0.13497 \text{ dp} \quad \text{Pion mass in GeV}

m_{K}=0.49761 \text{ dp} \quad \text{Kaon mass in GeV}

m_{\eta}=0.548 \text{ dp} \quad \text{Eta mass in GeV}

m_{\eta'}=0.958 \text{ dp} \quad \text{Eta' mass in GeV}

\text{Mesons:}

1: \pi^0

2: \eta

3: \eta'

\text{Do } i_1=1,3

If (i_1 .eq.1) Then !1: \pi^0

m^P = m_{\pi^0}

CP = 1._dp

DLdP = - m_{\pi^0}^2/4._dp

DrdP = - Conjg(DLdP)

DLuP = m_{\pi^0}^2/4._dp

DrUP = - Conjg(DLuP)

Else If (i_1 .eq.2) Then !2: \eta

m^P = m_{\eta}

CP = (\sin(\theta_\eta) + \sqrt{2}_dp \cos(\theta_\eta))/\sqrt{6}_dp

DLdP = 1._dp/(4._dp \sqrt{3}_dp) * ((3._dp m_{\pi^0} m_{K}^2 - 4._dp m_{K}^2 \sin(\theta_\eta)) &

& \cos(\theta_\eta) - 2.dp \sqrt{2}_dp m_{K}^2 \sin(\theta_\eta))

DrdP = - Conjg(DLdP)

DLuP = 1._dp/(4._dp \sqrt{3}_dp) * m_{\pi^0} m_{K}^2 \sin(\theta_\eta) &

& - \sqrt{2}_dp \sin(\theta_\eta)

DrUP = - Conjg(DLuP)

Else If (i_1 .eq.3) Then !3: \eta'

m^P = m_{\eta'}

CP = (\sqrt{2}_dp \sin(\theta_\eta) - \cos(\theta_\eta))/\sqrt{6}_dp

DLdP = 1._dp/(4._dp \sqrt{3}_dp) * ((3._dp m_{\pi^0} m_{K}^2 - 4._dp m_{K}^2 \cos(\theta_\eta)) &

& \sin(\theta_\eta) + 2.dp \sqrt{2}_dp m_{K}^2 \cos(\theta_\eta))

DrdP = - Conjg(DLdP)

DLuP = 1._dp/(4._dp \sqrt{3}_dp) * m_{\pi^0} m_{K}^2 \cos(\theta_\eta) +

& \sqrt{2}_dp \cos(\theta_\eta)

DrUP = - Conjg(DLuP)

End If

Leptons:

!1:e

!2:mu

Do i_2=1,2

If (i_2 .eq.1) Then ! tau \rightarrow e P

out = 1

Elseif (i_2 .eq.2) Then ! tau \rightarrow \mu P

out = 2

End if

! d–quark coefficients

BSLL = OllddSLL(3, out, 1, 1)

BSLR = OllddSLR(3, out, 1, 1)

BSRL = OllddSRL(3, out, 1, 1)

BSRR = OllddSRR(3, out, 1, 1)

BVLK = OllddVLK(3, out, 1, 1)

BVLR = OllddVLR(3, out, 1, 1)

BVRL = OllddVRL(3, out, 1, 1)

BVRR = OllddVRR(3, out, 1, 1)

! u–quark coefficients

CSLL = OlluuSLL(3, out, 1, 1)
CSLR = OlluuSLR(3, out, 1, 1)
CSRL = OlluuSLR(3, out, 1, 1)
CSRR = OlluuSRR(3, out, 1, 1)
CVLL = OlluuVLL(3, out, 1, 1)
CVLR = OlluuVLR(3, out, 1, 1)
CVRL = OlluuVRL(3, out, 1, 1)
CVRR = OlluuVRR(3, out, 1, 1)

! aP, bP scalar
aP(1) = Fpi/2._dp ∗ (DLdP/mf_d(1) ∗ (BSLL+BSRL) + DRdP/mf_d(1) ∗ (BSLR+BSRR)) &
& + DLuP/mf_u(1) ∗ (CSLL+CSRL) + DRuP/mf_u(1) ∗ (CSLR+CSRR))
bP(1) = Fpi/2._dp ∗ (DLdP/mf_d(1) ∗ (BSRL−BSLL) + DRdP/mf_d(1) ∗ (BSRR−BSLR)) &
& + DLuP/mf_u(1) ∗ (CSRL−CSLL) + DRuP/mf_u(1) ∗ (CSRR−CSLL))

! aP, bP vector
aP(2) = Fpi/4._dp ∗ Cp ∗ (mf_l(3)−mf_l(out)) ∗ (−BVLL−BVLR−BVRL−BVRR) &
& + CVLL+CVR−CVRL−CVRR)
bP(2) = Fpi/4._dp ∗ Cp ∗ (mf_l(3)+mf_l(out)) ∗ (−BVLL+BVLR−BVRL−BVRR) &
& + CVLL−CVR+CVRL−CVRR)

! averaged squared amplitude
meson_abs_T2=0._dp
Do k1=1,2
Do k2=1,2
cont=2._dp ∗ mf_l(out) ∗ mf_l(3) ∗ (aP(k1) ∗ conjg(aP(k2))) &
& − bP(k1) ∗ conjg(bP(k2))−
& (mf_l(3)^2+mf_l(out)^2−mp^2) ∗ (aP(k1) ∗ conjg(aP(k2))+ &
& & bP(k1) ∗ conjg(bP(k2)))
meson_abs_T2=meson_abs_T2+cont
End Do
End Do
meson_abs_T2=meson_abs_T2/(2._dp ∗ mf_l(3))

! branching ratio
factor=oo4pi ∗ sqrt(lamb(mf_l(3)^2,mf_l(out)^2,mp^2)) &
& /(mf_l(3)^2 ∗ GammaTau) ∗ 0.5_dp
BR=factor * meson_abs_T2
If (i1 .eq.1) Then ! pi
If (i2 .eq.1) Then
BrTautoEPi = BR
Else
BrTautoMuPi = BR
End If
Elseif (i1 .eq.2) Then ! eta
If (i2 .eq.1) Then
BrTautoEEta = BR
Else
BrTautoMuEta = BR
End If
Else ! eta
If (i2 .eq.1) Then
BrTautoEEtap = BR
Else
BrTautoMuEtap = BR
End If
End if
End Do
End Do
Contains
Real(dp) Function lamb(x,y,z)
Real(dp), Intent(in)::x,y,z
lamb=(x−y−z)^2−4._dp*x*y
End Function lamb
C.1.5: $h \rightarrow \ell_\alpha \ell_\beta$

The decay width is given by [96]

$$
\Gamma(h \rightarrow \ell_\alpha \ell_\beta) \equiv \Gamma(h \rightarrow \ell_\alpha \bar{\ell}_\beta) + \Gamma(h \rightarrow \bar{\ell}_\alpha \ell_\beta) = \frac{1}{16\pi m_h} \left[ 1 - \left( \frac{m_\ell_\alpha + m_\ell_\beta}{m_h} \right)^2 \right]^2 \times \left[ \left( \frac{m_h^2 - m_\ell_\alpha^2 - m_\ell_\beta^2}{m_h^2} \right) \left( |S_L|^2 + |S_R|^2 \right)_{\alpha\beta} - 4m_\ell_\alpha m_\ell_\beta \text{Re}(S_LS_R^{\alpha\beta}) \right]^{1/2} + (\alpha \leftrightarrow \beta)
$$

(C.43)

Listing 26 hLlp.m

```plaintext
NameProcess = "hLlp";
NameObservables = {{BrhtoMuE, 1101, "BR(h \rightarrow e \ mu)"},
{BrhtoTauE, 1102, "BR(h \rightarrow e \ tau)"},
{BrhtoTauMu, 1103, "BR(h \rightarrow \mu \ tau)"}};
NeededOperators = {OH2lSL, OH2lSR};
Body = "hLlp. f90 ";
```

Listing 27 hLlp.f90

```plaintext
Real(dp) :: width1 , width2 , width , mh, gamh, kinfactor
Complex(dp) :: SL1, SR1, SL2, SR2
Integer :: i1 , gt1 , gt2 , hLoc

!! h \rightarrow \ell \ell'
Observable implemented by W. Porod, F. Staub and A. Vicente
Based on E. Arganda et al, PRD 71 (2005) 035011 [hep-ph/0407302]

!! NEXT LINE HAVE TO BE PARSED BY SARAH
Checking if there are several generations of Scalars and what is the SM-like doublet
@ If [getGen[ HiggsBoson]>1, "hLoc = \rightarrow MaxLoc(Abs("<\rightarrow ToString [HiggsMixingMatrix]"<(2.:)) ,1)", "hLoc = 1"]
@ "mh = "<\rightarrow ToString [SPhenoMass[ HiggsBoson]]<\rightarrow If [getGen[ HiggsBoson]>1, "(hLoc)", ""]
@ "gamh = "<\rightarrow ToString [SPhenoWidth[ HiggsBoson]]<\rightarrow If [getGen[ HiggsBoson]>1, "(hLoc)", ""]
If (. not .L_BR) gamh = 4.5E-3_dp ! Decays not calculated; using SM value
Do i1=1,3
If (i1.eq.1) Then ! h \rightarrow e \mu
  gt1 = 1
  gt2 = 2
Elseif (i1.eq.2) Then ! h \rightarrow e \tau
  gt1 = 1
  gt2 = 3
Else ! h \rightarrow \mu \tau
  gt1 = 2
  gt2 = 3
End if
! width = Gamma(h \rightarrow \bar{\ell}_1 \ell_2) + Gamma(h \rightarrow \bar{\ell}_1 \bar{\ell}_2)
SL1 = OH2iSL(gt1 , gt2 , hLoc)
SR1 = OH2iSR(gt1 , gt2 , hLoc)
SL2 = OH2iSL(gt2 , gt1 , hLoc)
```

\[ \text{SR2} = \text{OH2} \text{SR} (gt2, gt1, hLoc) \]

\[
\text{kinfactor} = (1 - \frac{\text{mf}_l (gt1) + \text{mf}_l (gt2)}{\text{mh}})^2 & (1 - \frac{\text{mf}_l (gt1) - \text{mf}_l (gt2)}{\text{mh}})^2 \]

\[
\text{width1} = \left( \text{mh}^2 - \text{mf}_l (gt1)^2 - \text{mf}_l (gt2)^2 \right) \times \left( |R_{L1}|^2 + |R_{R1}|^2 \right) \nonumber \]

\[
\text{width2} = \left( \text{mh}^2 - \text{mf}_l (gt1)^2 - \text{mf}_l (gt2)^2 \right) \times \left( |R_{L2}|^2 + |R_{R2}|^2 \right) \nonumber \]

\[
\text{width} = \frac{\text{oo16pi} / \text{mh}}{\text{sqrt} \left( \text{kinfactor} \right)} \times (\text{width1} + \text{width2})
\]

\[
\text{If} \ (i1 . \text{eq.1}) \text{ Then} \]
\[
\text{BrhtoMuE} = \text{width} / (\text{width} + \text{gamh})
\]

\[
\text{Elseif} \ (i1 . \text{eq.2}) \text{ Then}
\]
\[
\text{BrhtoTauE} = \text{width} / (\text{width} + \text{gamh})
\]

\[
\text{Else}
\]
\[
\text{BrhtoTauMu} = \text{width} / (\text{width} + \text{gamh})
\]

\[
\text{End if}
\]

\[
\text{End do}
\]

\[ C.1.6: Z \to \ell_\alpha \ell_\beta \]

The decay width is given by \[ \text{(C.44)} \]

\[
\Gamma (Z \to \ell_\alpha \ell_\beta) \equiv \Gamma (Z \to \ell_\alpha \bar{\ell}_\beta) + \Gamma (Z \to \bar{\ell}_\alpha \ell_\beta) = \frac{m_Z}{48 \pi} \left[ 2 \left( |R_{L1}|^2 + |R_{R1}|^2 \right) + \frac{m_\beta^2}{4} \left( |R_{L2}|^2 + |R_{R2}|^2 \right) \right],
\]

where the charged lepton masses have been neglected.
C.2: Quark flavor observables

QFV has been observed and its description in the SM due to the CKM matrix is well established. However, the large majority of BSM models causes additional contributions which have to be studied carefully, see for instance Refs. [98–122].

We give also here a description of the implementation of the different observables using the operators present in the SPheno output of SARAH.

C.2.1: \(B_{s,d}^0 \to \ell^+\ell^−\)

Our analytical results for \(B_{s,d}^0 \to \ell^+\ell^−\) follow [103]. The \(B^0 \equiv B_{s,d}^0\) decay width to a pair of charged leptons can be written as

\[
\Gamma \left( B^0 \to \ell^+_\alpha \ell^-_\beta \right) = \frac{|\mathcal{M}_{\ell\ell}|^2}{16\pi M_B} \left[ \left( 1 - \left( \frac{m_{\ell_\alpha} + m_{\ell_\beta}}{m_B} \right)^2 \right) \left( 1 - \left( \frac{m_{\ell_\alpha} - m_{\ell_\beta}}{m_B} \right)^2 \right) \right]^{1/2}.
\]

(C.45)

Here

\[
|\mathcal{M}_{\ell\ell}|^2 = 2|F_S|^2 \left[ m_B^2 - \left( m_{\ell_\alpha} + m_{\ell_\beta} \right)^2 \right] + 2|F_P|^2 \left[ m_B^2 - \left( m_{\ell_\alpha} - m_{\ell_\beta} \right)^2 \right] + 2|F_V|^2 \left[ m_B^2 \left( m_{\ell_\alpha} - m_{\ell_\beta} \right)^2 - \left( m_{\ell_\alpha}^2 - m_{\ell_\beta}^2 \right)^2 \right] + 2|F_A|^2 \left[ m_B^2 \left( m_{\ell_\alpha} + m_{\ell_\beta} \right)^2 - \left( m_{\ell_\alpha}^2 - m_{\ell_\beta}^2 \right)^2 \right] + 4 \text{Re}(F_S F_V^∗) \left( m_{\ell_\alpha} - m_{\ell_\beta} \right) \left[ m_B^2 + \left( m_{\ell_\alpha} + m_{\ell_\beta} \right)^2 \right] + 4 \text{Re}(F_P F_A^∗) \left( m_{\ell_\alpha} + m_{\ell_\beta} \right) \left[ m_B^2 - \left( m_{\ell_\alpha} - m_{\ell_\beta} \right)^2 \right].
\]

(C.46)

and the \(F_X\) coefficients are defined in terms of our Wilson coefficients as\(^\text{12}\)

\[
F_S = \frac{i}{4} m_B^2 \frac{f_B}{m_d + m_d} \left( E_{LL}^S + E_{LR}^S - E_{RR}^S - E_{RL}^S \right)
\]

(C.47)

\[
F_P = \frac{i}{4} m_B^2 \frac{f_B}{m_d + m_d} \left( -E_{LL}^S + E_{LR}^S - E_{RR}^S + E_{RL}^S \right)
\]

(C.48)

\[
F_V = \frac{i}{4} f_B \left( E_{LL}^V + E_{LR}^V - E_{RR}^V - E_{RL}^V \right)
\]

(C.49)

\[
F_A = \frac{i}{4} f_B \left( -E_{LL}^V + E_{LR}^V - E_{RR}^V + E_{RL}^V \right).
\]

(C.50)

\(^\text{12}\)Notice that our effective Lagrangian differs from the one in [103] by a \(1/(4\pi)^2\) factor. This relative factor has been absorbed in the expression for \(\mathcal{M}_{\ell\ell}\), see Eq.(C.46).
where $f_B \equiv f_B^{00}$ is the $B^0_{d,s}$ decay constant and $m_{d,s}$ are the masses of the quarks contained in the $B$ meson, $B^0_d \equiv \bar{b}d$ and $B^0_s \equiv \bar{b}s$. In the lepton flavor conserving case, $\alpha = \beta$, the $F_V$ contribution vanishes. In this case, the results in [103] are in agreement with previous computations [123,124].

**Listing 30 B0ll.m**

```plaintext
NameProcess = "B0toLL";
NameObservables = {{BrB0dEE, 4000, "BR(B^0_d -> e e)",
{ratioB0dEE, 4001, "BR(B^0_d -> e e)/BR(B^0_d -> e e)_SM"},
{BrB0sEE, 4002, "BR(B^0_s -> e e)"},
{ratioB0sEE, 4003, "BR(B^0_s -> e e)/BR(B^0_s -> e e)_SM"},
{BrB0dMuMu, 4004, "BR(B^0_d -> mu mu)"},
{ratioB0dMuMu, 4005, "BR(B^0_d -> mu mu)/BR(B^0_d -> mu mu)_SM"},
{BrB0sMuMu, 4006, "BR(B^0_s -> mu mu)"},
{ratioB0sMuMu, 4007, "BR(B^0_s -> mu mu)/BR(B^0_s -> mu mu)_SM"},
{BrB0dTauTau, 4008, "BR(B^0_d -> tau tau)"},
{ratioB0dTauTau, 4009, "BR(B^0_d -> tau tau)/BR(B^0_d -> tau tau)_SM"},
{BrB0sTauTau, 4010, "BR(B^0_s -> tau tau)"},
{ratioB0sTauTau, 4011, "BR(B^0_s -> tau tau)/BR(B^0_s -> tau tau)_SM"}};

NeededOperators = {OddllSLL, OddllSRR, OddllSRL, OddllSLR, OddllVRR, OddllVLL, OddllVRL, OddllVLR, OddllSLLSM, OddllSRRSM, OddllSRLSM, OddllSLRSM, OddllVRRSM, OddllVLLSM, OddllVRLSM, OddllVLRSM};

Body = "B0ll.f90";
```

**Listing 31 B0ll.f90**

```plaintext
Real(dp) :: AmpSquared, AmpSquared2, AmpSquared_SM, AmpSquared2_SM, &
& width_SM, width
Real(dp) :: MassB0s, MassB0d, fBs, fBd, TauB0s, TauB0d
Real(dp) :: hbar=6.58211899E−25_dp
Real(dp) :: MassB0, MassB02, fB0, GammaB0
Complex(dp) :: CS(4), CV(4), CT(4)
Complex(dp) :: FS=0._dp, FP=0._dp, FV=0._dp, FA=0._dp
Integer :: i1, gt1, gt2, gt3, gt4

! Using global hadronic data
fBd = f_B0d_CONST
fBs = f_B0s_CONST
TauB0d = tau_B0d
TauB0s = tau_B0s
MassB0d = mass_B0d
MassB0s = mass_B0s

Do i1=1,6
  gt1 = 3
  If (i1.eq.1) Then ! B0 -> e+ e−
    MassB0 = MassB0d
  Else if (i1.eq.2) Then ! B0s -> e+ e−
    MassB0 = MassB0s
  MassB02 = MassB0d**2
  fB0 = fBd
  GammaB0 = (hbar)/(TauB0d)
  gt2 = 1
  gt3 = 1
  gt4 = 1
End Do
```

! Based on A. Dedes et al, PRD 79 (2009) 055006 [arXiv:0812.4320]
fB0 = fBs
GammaB0 = (hbar)/(TauB0s)
gt2 = 2
gt3 = 1
gt4 = 1
Else if (i1.eq.3) Then ! B0d -> mu+ mu-
MassB0 = MassB0d
MassB02 = MassB0d**2
fB0 = fBd
GammaB0 = (hbar)/(TauB0d)
gt2 = 1
gt3 = 2
gt4 = 2
Else if (i1.eq.4) Then ! B0s -> mu+ mu-
MassB0 = MassB0s
MassB02 = MassB0s**2
fB0 = fBs
GammaB0 = (hbar)/(TauB0s)
gt2 = 1
gt3 = 2
gt4 = 2
Else if (i1.eq.5) Then ! B0d -> tau+ tau-
MassB0 = MassB0d
MassB02 = MassB0d**2
fB0 = fBd
GammaB0 = (hbar)/(TauB0d)
gt2 = 1
gt3 = 3
gt4 = 3
Else if (i1.eq.6) Then ! B0s -> tau+ tau-
MassB0 = MassB0s
MassB02 = MassB0s**2
fB0 = fBs
GammaB0 = (hbar)/(TauB0s)
gt2 = 2
gt3 = 3
gt4 = 3
End if

! BSM contributions
CS(1) = OddllSRR(gt1, gt2, gt3, gt4)
CS(2) = OddllSRL(gt1, gt2, gt3, gt4)
CS(3) = OddllSLL(gt1, gt2, gt3, gt4)
CS(4) = OddllSLR(gt1, gt2, gt3, gt4)
CV(1) = OddllVLL(gt1, gt2, gt3, gt4)
CV(2) = OddllVLR(gt1, gt2, gt3, gt4)
CV(3) = OddllVRR(gt1, gt2, gt3, gt4)
CV(4) = OddllVRL(gt1, gt2, gt3, gt4)
FS= 0.25_dp * MassB02 * fB0 / (MFd(gt1)+MFd(gt2)) * (CS(1)+CS(2) - CS(3) - CS(4))
FP= 0.25_dp * MassB02 * fB0 / (MFd(gt1)+MFd(gt2)) * (-CS(1) - CS(2) - CS(3) + CS(4))
FV= -0.25_dp * fB0 * (CV(1)+CV(2) - CV(3) - CV(4))
FA= -0.25_dp * fB0 * (-CV(1) - CV(2) + CV(3) + CV(4))

AmpSquared = 2 * abs(FS)**2 * (MassB02 - (mf_l(gt3)+mf_l(gt4))**2) &
& + 2 * abs(FP)**2 * (MassB02 - (mf_l(gt3)+mf_l(gt3))**2) &
& + 2 * abs(FV)**2 * (MassB02 - (mf_l(gt3)-mf_l(gt4))**2) &
& + 2 * abs(FA)**2 * (MassB02 - (mf_l(gt3)-mf_l(gt3))**2) &
& + 4 * REAL(FS*conjugate(FV)) * (mf_l(gt3) - mf_l(gt4)) * (MassB02 &
& + (mf_l(gt3)-mf_l(gt4))**2) &
& + 4 * REAL(FP*conjugate(FA)) * (mf_l(gt3) - mf_l(gt4)) * (MassB02 &
& - (mf_l(gt3)-mf_l(gt4))**2) &
& + 4 * REAL((mf_l(gt4)+mf_l(gt3))**2) &
width = oo16pi * AmpSquared / MassB0 &
& sqrt(1 - ((mf_l(gt4)+mf_l(gt3))/MassB0)**2) &
\[ \text{If (i1.Eq.1) Then} \]
\[ \text{BrB0dEE= width / GammaB0} \]
\[ \text{ratioB0dEE= width / width_SM} \]
\[ \text{Else If (i1.Eq.2) Then} \]
\[ \text{BrB0sEE= width / GammaB0} \]
\[ \text{ratioB0sEE= width / width_SM} \]
\[ \text{Else If (i1.Eq.3) Then} \]
\[ \text{BrB0dMuMu= width / GammaB0} \]
\[ \text{ratioB0dMuMu= width / width_SM} \]
\[ \text{Else If (i1.Eq.4) Then} \]
\[ \text{BrB0sMuMu= width / GammaB0} \]
\[ \text{ratioB0sMuMu= width / width_SM} \]
\[ \text{Else If (i1.Eq.5) Then} \]
\[ \text{BrB0dTauTau= width / GammaB0} \]
\[ \text{ratioB0dTauTau= width / width_SM} \]
\[ \text{Else If (i1.Eq.6) Then} \]
\[ \text{BrB0sTauTau= width / GammaB0} \]
\[ \text{ratioB0sTauTau= width / width_SM} \]
\[ \text{End If} \]
\[ \text{End do} \]

C.2.2: \( \bar{B} \to X_s\gamma \)

The branching ratio for \( \bar{B} \to X_s\gamma \), with a cut \( E_\gamma > 1.6 \text{ GeV} \) in the \( \bar{B} \) rest frame, can be obtained as [104,125]

\[
\begin{align*}
\text{BR}(\bar{B} \to X_s\gamma)_{E_\gamma > 1.6\text{ GeV}} = 10^{-4} & \left[ a_{SM} + a_{77} \left( |\delta C_7^{(0)}|^2 + |\delta C_7^{(0)}|^2 \right) + a_{88} \left( |\delta C_8^{(0)}|^2 + |\delta C_8^{(0)}|^2 \right) + a_{78} \left( |\delta C_7^{(0)} \delta C_8^{(0)}|^2 + |\delta C_7^{(0)} \delta C_8^{(0)*}|^2 \right) \right. \\
& \left. + \text{Re} \left( a_{77} \delta C_7^{(0)} + a_{88} \delta C_8^{(0)} + a_{78} \left( \delta C_7^{(0)} \delta C_8^{(0)*} + \delta C_7^{(0)*} \delta C_8^{(0)} \right) \right) \right], \\
\end{align*}
\]
where $a_{SM} = 3.15$ is the NNLO SM prediction [51,126], the other $a$ coefficients in Eq.(C.51) are found to be

\[
a_{77} = 4.743
\]
\[
a_{88} = 0.789
\]
\[
a_{7} = -7.184 + 0.612i
\]
\[
a_{8} = -2.225 - 0.557i
\]
\[
a_{78} = 2.454 - 0.884i
\]

and we have defined $\delta C_i^{(0)} = C_i^{(0)} - C_i^{(0)\text{SM}}$. Finally, the $C_i^{(0)}$ coefficients can be written in terms of $Q_{1,2}^{L,R}$ in Eqs.(A.11) and (A.12) as

\[
C_7^{(0)} = n_{CQ} Q_1^R
\]
\[
C_7^{(0)} = n_{CQ} Q_1^L
\]
\[
C_8^{(0)} = n_{CQ} Q_2^R
\]
\[
C_8^{(0)} = n_{CQ} Q_2^L
\]

where $n_{CQ}^{-1} = -\frac{G_F}{4\sqrt{2}\pi} V_{td} V_{ts}^*$ and $V$ is the Cabibbo-Kobayashi-Maskawa (CKM) matrix.

**Listing 32 bsGamma.m**

```plaintext
NameProcess = "bsGamma";
NameObservables = {{BrBsGamma, 200, "BR(B^->X_s gamma)"},
{ratioBsGamma, 201, "BR(B^->X_s gamma)/BR(B^->X_s gamma)_SM"}};
NeededOperators = {CC7, CC7p, CC8, CC8p,
CC7SM, CC7pSM, CC8SM, CC8pSM};
Body = "bsGamma. f90 ";
```

**Listing 33 bsGamma.f90**

```plaintext
Integer :: gt1 , gt2
Complex(dp) :: norm, delta_C7_0 , delta_C7p_0 , delta_C8_0 , delta_C8p_0
Real(dp) :: NNLO_SM
!
! \bar{B}^-> X_s gamma (Egamma > 1.6 GeV)
! Observable implemented by W. Porod, F. Staub and A. Vicente
! Based on E. Lunghi, J. Matias, JHEP 0704 (2007) 058 [hep-ph/0612166]
!
!
! normalization of our Wilson coefficients
! relative to the ones used in hep-ph/0612166
norm = -CKM_160(3,3)*Conjg(CKM_160(gt1, gt2))*Alpha_160/ &
& (8._dp*Pi*sinW2_160*mW2)
!
! Wilson coefficients
delta_C7_0 = (CC7(gt1, gt2)-CC7SM(gt1, gt2))/norm
delta_C7p_0 = (CC7p(gt1, gt2)-CC7pSM(gt1, gt2))/norm
delta_C8_0 = (CC8(gt1, gt2)-CC8SM(gt1, gt2))/norm
delta_C8p_0 = (CC8p(gt1, gt2)-CC8pSM(gt1, gt2))/norm
!
! NNLO SM prediction
! as obtained in M. Misiak et al, PRL 98 (2007) 022002
! and M. Misiak and M. Steinhauser, NPB 764 (2007) 62
NNLO_SM=3.15_dp
!
BrBsGamma=NNLO_SM+4.743_dp*(Abs(delta_C7_0)**2+Abs(delta_C7p_0)**2)&
&+0.789_dp*(Abs(delta_C8_0)**2+Abs(delta_C8p_0)**2)&
```
C.2.3: $\bar{B} \rightarrow X_s \ell^+\ell^-$

Our results for $\bar{B} \rightarrow X_s \ell^+\ell^-$ are based on [106], expanded with the addition of prime operators contributions [127]. The branching ratios for the $\ell = e$ case can be written as

$$10^7 \text{BR}(\bar{B} \rightarrow X_s e^+e^-) = 2.3148 - 0.001658 \text{Im}(R_{10}) + 0.0005 \text{Im}(R_{10}R_s^* + R_{10}'R_s'^*)$$

+ 0.0523 Im($R_7$) + 0.02266 Im($R_7R_s^* + R_7'R_s'^*$) + 0.00496 Im($R_7R_s^* + R_7'R_s'^*$)

+ 0.00518 Im($R_8$) + 0.0261 Im($R_9R_s^* + R_9'R_s'^*$) − 0.00621 Im($R_9$) − 0.5420 Re($R_{10}$)

− 0.03340 Re($R_7$) + 0.0153 Re($R_7R_s^* + R_7'R_s'^*$) + 0.0673 Re($R_7R_s^* + R_7'R_s'^*$)

− 0.86916 Re($R_7R_s^* + R_7'R_s'^*$) − 0.0135 Re($R_8$) + 0.00185 Re($R_8R_10^* + R_8'R_10'^*$)

− 0.09921 Re($R_8R_s + R_8'R_s'^*$) + 2.833 Re($R_9$) − 0.10698 Re($R_9R_10^* + R_9'R_10'^*$)

+ 11.0348 ($|R_{10}|^2 + |R_{10}'|^2$) + 0.2804 ($|R_7|^2 + |R_7'|^2$)

+ 0.003763 ($|R_8|^2 + |R_8'|^2$) + 1.527 ($|R_9|^2 + |R_9'|^2$),

(C.57)

whereas for the $\ell = \mu$ case one gets

$$10^7 \text{BR}(\bar{B} \rightarrow X_s \mu^+\mu^-) = 2.1774 - 0.001658 \text{Im}(R_{10}) + 0.0005 \text{Im}(R_{10}R_s^* + R_{10}'R_s'^*)$$

+ 0.0534 Im($R_7$) + 0.02266 Im($R_7R_s^* + R_7'R_s'^*$) + 0.00496 Im($R_7R_s^* + R_7'R_s'^*$)

+ 0.00527 Im($R_8$) + 0.0261 Im($R_9R_s^* + R_9'R_s'^*$) − 0.0115 Im($R_9$) − 0.5420 Re($R_{10}$)

+ 0.0208 Re($R_7$) + 0.0153 Re($R_7R_s^* + R_7'R_s'^*$) + 0.0648 Re($R_7R_s^* + R_7'R_s'^*$)

− 0.8545 Re($R_7R_s^* + R_7'R_s'^*$) − 0.00938 Re($R_8$) + 0.00185 Re($R_8R_10^* + R_8'R_10'^*$)

− 0.0981 Re($R_8R_s + R_8'R_s'^*$) + 2.6917 Re($R_9$) − 0.10698 Re($R_9R_10^* + R_9'R_10'^*$)

+ 10.7652 ($|R_{10}|^2 + |R_{10}'|^2$) + 0.2880 ($|R_7|^2 + |R_7'|^2$)

+ 0.003763 ($|R_8|^2 + |R_8'|^2$) + 1.527 ($|R_9|^2 + |R_9'|^2$).

(C.58)

Here we have defined the ratios of Wilson coefficients

$$R_{7,8} = \frac{Q_{1,2}^R}{Q_{1,2}^L}, \quad R_{7,8}' = \frac{Q_{1,2}^L}{Q_{1,2}^R}$$

(C.59)

as well as

$$R_{9,10} = \frac{E_{LL}^V \pm E_{LR}^V}{E_{LL}^V \pm E_{LR}^V}, \quad R_{9,10}' = \frac{E_{RR}^V \pm E_{RL}^V}{E_{RR}^V \pm E_{RL}^V}.$$  

(C.60)

Listing 34 BtoSLL.m

| NameProcess = "BtoSLL"; |
| NameObservables = [{BrBtoSEE, 5000, "BR(B-> s e e)"}, |
| \{{ratioBtoSEE, 5001, "BR(B-> s e e)/BR(B-> s e e)_SM"}, |
| \{{BrBtoSMuMu, 5002, "BR(B-> s mu mu)"}, |
| \{{ratioBtoSMuMu, 5003, "BR(B-> s mu mu)/BR(B-> s mu mu)_SM"}}; |
NeededOperators = {OddllVRR, OddllVLL, OddllVRL, OddllVLR, 
CC7, CC7p, CC8, CC8p, 
OddllVRRSM, OddllVLLSM, OddllVRLSM, OddllVLRSM, 
CC7SM, CC7pSM, CC8SM, CC8pSM 
};

Body = "BtoSLL.f90";

### Listing 35 BtoSLL.f90

```fortran
Complex(dp) :: c7(2), c7p(2), c8(2), c8p(2), r7, r7p, r8, r8p, norm, &
& r9(2), r9p(2), r10(2), r10p(2), &
& c9ee(2), c9pee(2), c10ee(2), c10pee(2), &
& c9_cee(2), c9_ppee(2), c10_cee(2), c10_ppee(2), &
& c9mm(2), c9mm(2), c10mm(2), c10mm(2), c9_cmm(2), &
& c9p_cmm(2), c10_cmm(2), c10p_cmm(2)

! Observable implemented by W. Porod, F. Staub and A. Vicente
! Based on T. Huber et al, NPB 740 (2006) 105, [hep-ph/0512066]
! Prime operators added after private communication with E. Lunghi

! Wilson coefficients

c7(1) = CC7(3,2)
c7(2) = CC7SM(3,2)
c8(1) = CC8(3,2)
c8(2) = CC8SM(3,2)
c8p(1) = CC8p(3,2)
c8p(2) = CC8pSM(3,2)
c9ee(1) = OddllVLL(3,2,1,1)+OddllVLR(3,2,1,1)
c9ee(2) = OddllVLLSM(3,2,1,1)+OddllVLRSM(3,2,1,1))
c9mm(1) = OddllVLL(3,2,2,2)+OddllVLR(3,2,2,2)
c9mm(2) = OddllVLLSM(3,2,2,2)+OddllVLRSM(3,2,2,2)
c9pee(1) = OddllVRR(3,2,1,1)+OddllVRL(3,2,1,1)
c9pee(2) = OddllVRRSM(3,2,1,1)+OddllVRLSM(3,2,1,1)
c9mm(1) = OddllVRR(3,2,2,2)+OddllVRL(3,2,2,2)
c9mm(2) = OddllVRRSM(3,2,2,2)+OddllVRLSM(3,2,2,2)
c10ee(1) = OddllVLL(3,2,1,1)+OddllVLR(3,2,1,1)
c10ee(2) = OddllVLLSM(3,2,1,1)+OddllVLRSM(3,2,1,1)
c10mm(1) = OddllVLL(3,2,2,2)+OddllVLR(3,2,2,2)
c10mm(2) = OddllVLLSM(3,2,2,2)+OddllVLRSM(3,2,2,2)
c10pee(1) = OddllVRR(3,2,1,1)+OddllVRL(3,2,1,1)
c10pee(2) = OddllVRRSM(3,2,1,1)+OddllVRLSM(3,2,1,1)

d9(1) = c9ee(1)/c9ee(2)
d9(2) = c9mm(1)/c9mm(2)
d9p(1) = c9pee(1)/c9ee(2)
d9p(2) = c9pmm(1)/c9mm(2)
d10(1) = c10ee(1)/c10ee(2)
d10(2) = c10mm(1)/c10mm(2)
```

! ratios
C.2.4: $B^+ \to K^+\ell^+\ell^-$

Our results for $B^+ \to K^+\ell^+\ell^-$ are based on the expressions given in [102]. The branching ratio for $B^+ \to K^+\mu^+\mu^-$ in the high-$q^2$ region, $q^2$ being the dilepton invariant mass squared, can be written as

$$ \text{BR}\left( B^+ \to K^+\mu^+\mu^- \right) |_{q^2 \in [14,18,22] \text{GeV}^2} \approx 1.11 + 0.22 \left( c_{\mu}^{\text{NP}} + C_7 \right) + 0.27 \left( c_{\mu}^{\text{NP}} + C_6 \right) - 0.27 \left( c_{10}^{\text{NP}} + C_{10} \right). $$

(C.61)
The coefficients in Eq. (C.61) can be related to the ones in our generic Lagrangian as

\[ C_{7}^{NP} = n_{CQ} \left( Q_{1}^{R} - Q_{1}^{R, SM} \right) \]  
(C.62)

\[ C_{7}' = n_{CQ} Q_{1}^{T} \]  
(C.63)

\[ C_{9}^{NP} = n_{CQ} \left( E_{L}^{V} + E_{L}^{R} \right) \]  
(C.64)

\[ C_{9}' = n_{CQ} \left( E_{R}^{V} + E_{R}^{L} \right) \]  
(C.65)

\[ C_{10}^{NP} = n_{CQ} \left( E_{L}^{V} - E_{L}^{R} \right) \]  
(C.66)

\[ C_{10}' = n_{CQ} \left( E_{R}^{V} - E_{R}^{L} \right) \]  
(C.67)

where the normalization factor \( n_{CQ} \) was already defined after Eq. (C.56).

Listing 36 BtoKLL.m

```
NameProcess = "BtoKLL";
NameObservables = {{BrBtoKmumu, 6000, "BR(B -> K mu mu)",
{ratioBtoKmumu, 6001, "BR(B -> K mu mu)/BR(B -> K mu mu)_SM"}};
NeededOperators = {OddllVRR, OddllVLL, OddllVRL, OddllVLR, CC7, CC7p,
OddllVRRSM, OddllVLLSM, OddllVRLSM, OddllVLRSM, CC7SM, CC7pSM} ;
Body = "BtoKLL.f90";
```

Listing 37 BtoKLL.f90

```
Complex(dp) :: c7NP, c7p , c9NP, c9p , c10NP, c10p , norm
Real(dp) :: GF

! B^+ -> K^+ l+ l^- (14.18 GeV^2 < q^2 < 22 GeV^2)
! Observable implemented by W. Porod, F. Staub and A. Vicente
! Based on W. Altmannshofer, D. M. Straub, EPJ C 73 (2013) 2646
! [arXiv:1308.1501]

c7NP = (CC7(3,2) - CC7SM(3,2))
c7p = CC7p(3,2)
c9NP = (OddllVLL(3,2,1,1)+OddllVLR(3,2,1,1) -
& (OddllVLLSM(3,2,1,1)+OddllVLRSM(3,2,1,1)))
c9p = (OddllVRR(3,2,1,1)+OddllVRL(3,2,1,1))
c10NP = (OddllVLL(3,2,1,1)-OddllVLR(3,2,1,1) -
& (OddllVLLSM(3,2,1,1)-OddllVLRSM(3,2,1,1)))
c10p = (OddllVRR(3,2,1,1)-OddllVRL(3,2,1,1))

! running GF
GF = (Alpha_160*4._dp*Pi/sinW2_160)/mw^2*sqrt2/8._dp
! normalization of our Wilson coefficients
! relative to the ones used in arXiv:1308.1501
norm = - oo16pi2*4._dp*GF/sqrt2*CKM_160(3,3)*Conjg(CKM_160(3,2))
! Branching ratio in the high q^-2 region
! q^2 in [14.18,22] GeV^2
BrBtoKmumu = (1.11_dp + 0.22_dp*(c7NP+c7p)/norm +
& 0.27_dp*(c9NP+c9p)/norm - 0.27_dp*(c10NP+c10p)/norm)
! ratio relative to SM
ratioBtoKmumu = BrBtoKmumu/1.11_dp
! total BR
BrBtoKmumu = BrBtoKmumu*1.0E-7_dp
```
C.2.5: $\bar{B} \to X_{d,s} \nu \bar{\nu}$

The branching ratio for $\bar{B} \to X_{q} \nu \bar{\nu}$, with $q = d, s$, is given by [105]

$$
\text{BR} ( \bar{B} \to X_{q} \nu \bar{\nu}) = \frac{\alpha^2}{4\pi^2 \sin^4 \theta_W} \left| V_{tb} V_{tq}^* \right|^2 \frac{BR (\bar{B} \to X_{c} \bar{\nu} \bar{\nu}) \kappa(0)}{f(m_c) \kappa(m_c)} \sum f \left[ |c_L|^2 + |c_R|^2 \right] f(m_q) - 4 \text{Re} \left( c_L c_R^* \right) \epsilon_q f(m_q) .
$$

(C.68)

The sum runs over the three neutrinos and $\hat{m}_i \equiv m_i/m_b$. The functions $f(x)$ and $\kappa(m_c)$ represent the phase-space and the 1-loop QCD corrections, respectively. In case of $\kappa(m_c)$, one needs the numerical values $\kappa(0) = 0.83$ and $\kappa(\hat{m}_c) = 0.88$. The functions $f(x)$ and $\tilde{f}(x)$ take the form

$$
f(x) = 1 - 8x^2 + 8x^6 - x^8 - 24x^4 \log x
$$

(C.69)

and

$$
\tilde{f}(x) = 1 + 9x^2 - 9x^4 - x^6 + 12x^2 (1 + x^2) \log x.
$$

(C.70)

Finally, $\text{BR} (\bar{B} \to X_{c} \nu \bar{\nu})_{\text{exp}} = 0.101$ [128] and the coefficients $c_L$ and $c_R$ are given by

$$
c_L = n_{BX \nu \nu}^q F_{LL} \quad \text{(C.71)}
$$

$$
c_R = n_{BX \nu \nu}^q F_{RL} \quad \text{(C.72)}
$$

where $(n_{BX \nu \nu}^q)^{-1} = \frac{4GF}{\sqrt{2} \sin \theta_W} V_{tb} V_{tq}^*$ is the relative factor between our Wilson coefficients and the ones in [105].

Listing 38 BtoQnnu.m

```plaintext
NameProcess = "BtoQnu";

NameObservables = {{BrBtoSnunu, 7000, "BR(B -> s nu nu)",
{ratioBtoSnunu, 7001, "BR(B -> s nu nu)/BR(B -> s nu nu)_SM"},
{BrBtoDnunu, 7002, "BR(B -> D nu nu)",
{ratioBtoDnunu, 7003, "BR(B -> D nu nu)/BR(B -> D nu nu)_SM")};

NeededOperators = {OddvvVRR, OddvvVLL, OddvvVRL, OddvvVLR,
OddvvVRRSM, OddvvVLLSM, OddvvVRLSM, OddvvVLRSM};

Body = "BtoQnu.m.90 *;"
```

Listing 39 BtoQnnu.f90

```plaintext
Complex(dp) :: cL, cR, br, br_SM, cL_SM, cR_SM, norm
Real(dp) :: f_mq, tf_mq, kappa_0, kappa_c, f_mc, BrBXeNu, sw2, mq
Real(dp) :: prefactor, factor1, factor2, GF
Integer :: out, i1, i2

\bar{B} \to X_{d\nu}

kappa_0 = 0.830_dp
kappa_c = 0.88_dp
sw2 = sinw2_160
GF = (Alpha_160 * 4._dp * Pi/sinW2_160)/mw * 2*sqrt2 /8._dp

Do out = 1,2
If (out.eq.1) Then ! B \to X_d nu nu
mq = mf_d(1)/mf_d(3)
BrBXeNu = 0.101_dp ! PDG central value
sw2 = sinw2_160
GF = (Alpha_160 * 4._dp * Pi/sinW2_160)/mw * 2*sqrt2 /8._dp

Do out = 1,2
If (out.eq.1) Then ! B \to X_d nu nu
mq = mf_d(1)/mf_d(3)
norm = Alpha_160 * 4._dp * GF / sqrt2 / (2._dp * pi * sinw2_160) * &
        & Conjg(CKM_160(3,3)*Conjg( CKM_160(3,1) ))
```

\end{document}
Elif ! B -> X_s nu nu
  \[ m_q = \frac{m_f}{m_f(3)} \]
  \[ \text{norm} = A_{\text{\Lambda}} + \frac{A}{\sqrt{2}} \]
  \[ \text{Conjg}(\text{CKM}_{\text{\Lambda}}(3,3)) \]
  End if

! f and tilde f functions

\[ f_{\text{mq}} = 1. - 8. \cdot m_q \cdot m_q^2 + 8. \cdot m_q \cdot m_q^6 \]
\[ \text{tf}_{\text{mq}} = 1. + 9. \cdot m_q \cdot m_q^2 - 9. \cdot m_q \cdot m_q^4 + m_q^6 \]

\[ \text{prefactor} = A_{\text{\Lambda}} \cdot \frac{A}{\sqrt{2}} \cdot \Big( \frac{1}{(\text{f}\_\text{mc} \cdot \text{kappa}_c) \cdot \text{kappa}_0} \Big) \]

\[ \text{br} = 0. \]
\[ \text{br}_\text{SM} = 0. \]

Do i1 = 1, 3
  Do i2 = 1, 3
    ! BSM
    cL = OddVVLL(3, out, i1, i2) / norm
    cR = OddVVRL(3, out, i1, i2) / norm
    br = br + factor1 * (Abs(cL) \cdot Abs(cR) + & & factor2 * Real(cL, Conjg(cR), dp)

    ! SM
    cL = OddVVLLSM(3, out, i1, i2) / norm
    cR = OddVVRLSM(3, out, i1, i2) / norm
    br_\text{SM} = br_\text{SM} + factor1 * (Abs(cL) \cdot Abs(cR) + & & factor2 * Real(cL, Conjg(cR), dp)
  End Do
End do
If (out = 1) Then ! B -> X_d nu nu
  BrToDnu = \text{prefactor} \cdot br \cdot Abs(\text{CKM}_{\text{\Lambda}}(3,1)) \cdot \text{br}\_\text{SM} = \text{br}/\text{br}\_\text{SM}
Else ! B -> X_s nu nu
  BrToSnu = \text{prefactor} \cdot br \cdot Abs(\text{CKM}_{\text{\Lambda}}(3,2)) \cdot \text{br}\_\text{SM} = \text{br}/\text{br}\_\text{SM}
End if
End Do

C.2.6: \( K \rightarrow \pi \nu \bar{\nu} \)

Following [105], the branching ratios for rare Kaon decays involving neutrinos in the final state can be written as

\[
\text{BR} \left( K^+ \rightarrow \pi^+ \nu \bar{\nu} \right) = 2r_1 r_2 r_{K^+} \sum_f \left[ (\text{Im}\lambda_t X_f)^2 + (\text{Re}\lambda_t X_{NL} + \text{Re}\lambda_t X_f)^2 \right]
\]

\[
\text{BR} \left( K_L \rightarrow \pi^0 \nu \bar{\nu} \right) = 2r_1 r_{K_L} \sum_f (\text{Im}\lambda_c X_f)^2,
\]

where the sums are over the three neutrino species, \( X_{NL} = 9.78 \cdot 10^{-4} \) is the SM NLO charm correction [48,129], \( \lambda_t = V_{ts}^* V_{td} \) and \( \lambda_c = V_{cs}^* V_{cd} \), the coefficients \( r_1, r_2, r_{K^+} \) and \( r_{K_L} \) take the numerical values

\[
\begin{align*}
r_1 &= 1.17 \cdot 10^{-4} \\
r_2 &= 0.24 \\
r_{K^+} &= 0.901 \\
r_{K_L} &= 0.944
\end{align*}
\]
and $X_f$ contains the Wilson coefficients contributing to the processes, $F_{LL}^V$ and $F_{RL}^V$, as

$$X_f = n_{K\pi\nu\nu} \left( F_{LL}^V + F_{RL}^V \right).$$

(C.76)

Here $n_{K\pi\nu\nu}^{-1} = \frac{4G_F}{\sqrt{2}} \frac{\alpha_s}{2\pi} \sin^2 \theta_W V_{ts}^* V_{td}$.

Listing 40 KtoPInunu.m

```plaintext
1 NameProcess = "KtoPInunu";
2 NameObservables = {{BrKptoPipnunu, 8000, "BR(K^+ -> pi^+ nu nu)":
3 {ratioKptoPipnunu, 8001, "BR(K^+ -> pi^+ nu nu)/BR(K^+ -> pi^+ nu \rightarrow\nu)_SM"},
4 {BrKtoPInunu, 8002, "BR(K_L -> pi^0 nu nu)"},
5 {ratioKtoPInunu, 8003, "BR(K_L -> pi^0 nu nu)/BR(K_L -> pi^0 nu \rightarrow\nu)_SM"}};
6 NeededOperators = {OddvvVRR, OddvvVLL, OddvvVRL, OddvvVLR};
7 Body = "KtoPInunu.f90 ";
```

Listing 41 KtoPInunu.f90

```plaintext
1 Complex(dp) :: br , r1 , r2 , rKp , rKl , Xx, XNL, Lt , Lc
2 Complex(dp) :: Xx_SM, br_SM, norm
3 Real(dp) :: GF
4 Integer :: out , i1 , i2
5
6 ! K->pi nu nu
7 ! Observable implemented by W. Porod, F. Staub and A. Vicente
8 ! Based on C. Bobeth et al, NPB 630 (2002) 87 [hep-ph/0112305]
9
10 GF = (Alpha_160 * 4._dp * Pi / sinW2_160) / mw**2 * sqrt2 / 8._dp
11 norm = Alpha_160 * 4._dp * GF / sqrt2 / (2._dp * pi * sinw2_160) & & * Conjug(CKM_160(3,2)) * CKM_160(3,1)
12
13 r1 = 1.17E-4_dp
14 r2 = 0.24_dp
15 rKp = 0.901
16 rKl = 0.944
17
18 ! SM NLO charm correction
19 ! See G. Buchalla and A. Buras, NPB 412 (1994) 106 and NPB 548 (1999) 309
20 XNL = 9.78E-4_dp
21
22 ! out = 1 : K^+ -> pi^+ nu nu
23 ! out = 2 : K_L -> pi^0 nu nu
24 ! out = 1,2
25 ! Do out = 1,2
26 ! br = 0._dp
27 ! br_SM = 0._dp
28 ! Do i1 = 1,3
29 ! Do i2 = 1,3
30
31 ! Xx = ((OddvvVLL(2,1,i1,i2) + OddvvVRL(2,1,i1,i2)) / norm)
32 ! Xx_SM = ((OddvvVLLSM(2,1,i1,i2) + OddvvVRLSM(2,1,i1,i2)) / norm)
33 ! Lt = Conjug(CKM_160(3,2)) * CKM_160(3,1)
34 ! Lc = Conjug(CKM_160(2,2)) * CKM_160(2,1)
35 ! If (out.==1) Then
36 ! br = br + Aimag(Xx*Lt)**2 + (Real(Lc*XNL,dp) + Real(Xx*Lt,dp))**2
37 ! br_SM = br_SM + Aimag(Xx_SM*Lt)**2 + & & (Real(Lc*XNL,dp) + Real(Xx_SM*Lt,dp))**2
38 ! Else
39 ! br = br + Abs(Aimag(Xx*Lt))**2
40 ! br_SM = br_SM + Abs(Aimag(Xx_SM*Lt))**2
41 ! End if
```

The $\Delta M_{B_{q}}$ mass difference can be written as [108,130]

$$\Delta M_{B_{b}} = \frac{G_{F}}{2\pi} m_{B_{q}} \eta_{v} f_{B_{q}}^{2} V_{1q}^{\text{eff}} |F_{tt}^{q}|,$$  \hfill (C.77)

where $q = s, d$, $m_{B_{q}}$ and $f_{B_{q}}$ are the $B_{q}$ mass and decay constant, respectively, $\eta_{v} = 0.55$ is a QCD factor [47,131], $\hat{B}_{B_{q}}$ is a non-perturbative parameter (with values $\hat{B}_{B_{s}} = 1.26$ and $\hat{B}_{B_{d}} = 1.33$, obtained from recent lattice computations [132]) and $|V_{1q}^{\text{eff}}|^{2} = (V_{2q}^{\text{eff}})^{2}$. $F_{tt}^{q}$ is given by

$$F_{tt}^{q} = S_{0}(x_{t}) + \frac{1}{4r} C_{\text{new}}^{VLL}$$

$$+ \frac{1}{4r} C_{\text{new}}^{VRR} + C_{1}^{LR} C_{1}^{LR} + C_{2}^{LR} C_{2}^{LR}$$

$$+ \tilde{P}_{1}^{SLL} (c_{1}^{SLL} + c_{1}^{SRR}) + \tilde{P}_{2}^{SLL} (C_{2}^{SLL} + C_{2}^{SRR})$$  \hfill (C.78)

where $r = 0.985$ [47], $x_{t} = \frac{m_{t}^{2}}{m_{W}^{2}}$, with $m_{t}$ the top quark mass, the $P$ coefficients take the numerical values

$$\tilde{P}_{1}^{LR} = -0.71$$
$$\tilde{P}_{2}^{LR} = 0.90$$
$$\tilde{P}_{1}^{SLL} = -0.37$$
$$\tilde{P}_{2}^{SLL} = -0.72$$  \hfill (C.79)

and the function

$$S_{0}(x_{t}) = \frac{4x_{t} - 11x_{t}^{2} + x_{t}^{3}}{4(1 - x_{t})^{2}} - \frac{3x_{t}^{2} \log x_{t}}{2(1 - x_{t})^{3}}$$  \hfill (C.80)

was introduced by Inami and Lim in [133] and given, for example, in [134]. Finally, the coefficients in Eq. (C.78) are related to the $D_{X}^{Y}$ coefficients in Eq.(A.13) as

$$C_{\text{new}}^{VLL} = n_{\Delta}^{q} (D_{LL}^{V} - D_{LL}^{V,SM})$$
$$C_{\text{new}}^{VRR} = n_{\Delta}^{q} D_{RR}^{V}$$
$$C_{1}^{LR} = n_{\Delta}^{q} (D_{LR}^{V} + D_{RL}^{V})$$
$$C_{2}^{LR} = n_{\Delta}^{q} (D_{LR}^{S} + D_{RL}^{S} + \delta_{2}^{LR})$$
$$C_{1}^{SLL} = n_{\Delta}^{q} (D_{LL}^{S} + \delta_{1}^{SLL})$$
$$C_{1}^{SRR} = n_{\Delta}^{q} (D_{RR}^{S} + \delta_{1}^{SRR})$$
$$C_{2}^{SLL} = n_{\Delta}^{q} D_{LL}^{S}$$
$$C_{2}^{SRR} = n_{\Delta}^{q} D_{RR}^{S}$$
$$C_{1}^{LR} = n_{\Delta}^{q} (D_{LR}^{V} + D_{RL}^{V})$$
$$C_{2}^{LR} = n_{\Delta}^{q} (D_{LR}^{S} + D_{RL}^{S} + \delta_{2}^{LR})$$
$$C_{1}^{SLL} = n_{\Delta}^{q} (D_{LL}^{S} + \delta_{1}^{SLL})$$
$$C_{1}^{SRR} = n_{\Delta}^{q} (D_{RR}^{S} + \delta_{1}^{SRR})$$
$$C_{2}^{SLL} = n_{\Delta}^{q} D_{LL}^{S}$$
$$C_{2}^{SRR} = n_{\Delta}^{q} D_{RR}^{S}$$

C.2.7: $\Delta M_{B_{s,d}}$

The $B_{q}^{0} - \bar{B}_{q}^{0}$ mass difference can be written as [108,130]
where the factor \((n_q^2)^{-1}\) normalizes our Wilson coefficients to the ones in \([108, 130]\). The corrections \(\delta_{LR}^f\), \(\delta_{SLL}^f\), and \(\delta_{SRR}^f\) are induced by double penguin diagrams mediated by scalar and pseudoscalar states \([108, 130]\). These 2-loop contributions may have a sizable impact in some models, and their inclusion is necessary in order to achieve a precise result for \(\Delta M_{B_q}\). They can be written as

\[
\begin{align*}
\delta_{LR}^f &= -\frac{H_{L}^{S,P} \left(H_{R}^{S,P}\right)^*}{m_{S,P}^2} \\
\delta_{SLL}^f &= -\frac{(H_{L}^{S,P})^2}{2 m_{S,P}^2} \\
\delta_{SRR}^f &= -\frac{(H_{L}^{S,P})^2}{2 m_{S,P}^2}
\end{align*}
\] (C.89)

where \(H_{L}^{S,P}\) and \(H_{R}^{S,P}\) are defined in Eq.(A.17). The double penguin corrections in Eqs.(C.89)-(C.91) are obtained by summing up over all scalar and pseudoscalar states in the model.

Listing 42 DeltaMBq.m

```plaintext
NameProcess = "DeltaMBq";
NameObservables = {{DeltaMBs, 1900, "Delta(M_{Bs})"},
                   {ratioDeltaMBs, 1901, "Delta(M_{Bs})/Delta(M_{Bs})_SM"},
                   {DeltaMBq, 1902, "Delta(M_{Bd})"},
                   {ratioDeltaMBq, 1903, "Delta(M_{Bd})/Delta(M_{Bd})_SM"}};
ExternalStates = {Fd};
NeededOperators = {O4dSLL, O4dSRR, O4dSRL, O4dSLR, O4dVRR, O4dVLL,
                   O4dVLLSM, O4dVRL, O4dVLR, O4dTLR, O4dTRL, O4dTRR};
IncludeSMprediction["DeltaMBq"] = False;
```

Listing 43 DeltaMBq.f90

```plaintext
Complex(dp) :: MBq, etaB, FBq2, BBq, Ftt, Veff2, r, &
               P1bLR, P2bLR, P1bSLL, P2bSLL, norm, &
               CVLLnew, CVVRR, C1LR, C2LR, C1SLL, C1SRR, C2SLL, C2SRR
Real(dp) :: hbar, xt, GF
Real(dp) :: mS
Complex(dp) :: HL, HR, AL, AR
Integer :: i1, iS
```

Delta \(M_{Bd, Bs}\) 

Observable implemented by W. Porod, F. Staub and A. Vicente 
Based on A. J. Buras et al., NPB 619 (2001) 434 [hep-ph/0107048] 
and NPB 659 (2003) 3 [hep-ph/0210145]

\(\text{QCD factor, see A. J. Buras et al., NPB 47 (1990) 491}
\)
and J. Urban et al., NPB 523 (1998) 40 
\(\text{etaB = 0.55}\) 
\(\text{GF = (Alpha_160*4._dp*Pi/sinW2_160)/mw**2*sinW2_160/sqrt2/8._dp}\)
Do i1 = 1, 2
If (i1 .eq. 1) Then ! Delta M_Bd
MBq = mass_B0d
FBq2 = f_B0d_CONST**2
BBq = 1.26_dp ! see arXiv:0910.2928
Veff2 = Conjg(Conjg(CKM_160(3,3))*CKM_160(3,1))**2
Else ! Delta M_Bs
MBq = mass_B0s
FBq2 = f_B0s_CONST**2
BBq = 1.33_dp ! see arXiv:0910.2928
Veff2 = Conjg(Conjg(CKM_160(3,3))*CKM_160(3,2))**2
End if

! normalization factor
norm = GF**2*mw2/(16._dp*pi**2)*Veff2

! Wilson coefficients
CVLLnew = (O4dVLL(3,3,3,3)−O4dVLLSM(3,3,3,3))/norm ! we remove the SM contribution
C1VRR = O4dVRR(3,3,3,3)/norm
C1LR = (O4dVLR(3,3,3,3)+O4dVRL(3,3,3,3))/norm
C2LR = (O4dSLR(3,3,3,3)+O4dSRL(3,3,3,3))/norm
C1SLL = O4dSLL(3,3,3,3)/norm
C1SRR = O4dSRR(3,3,3,3)/norm
C2SLL = O4dTLL(3,3,3,3)/norm
C2SRR = O4dTRR(3,3,3,3)/norm

! Double Higgs penguins
If [getGen[HiggsBoson] > 1, "Do iS = 1, "<>toString [getGen[HiggsBoson]], ""]
If [getGen[HiggsBoson] > 1, "HL = OH2qSL(3,3,iS)", "HL = OH2qSL(3,3,iS)"
If [getGen[HiggsBoson] > 1, "HR = OH2qSR(3,3,iS)", "HR = OH2qSR(3,3,iS)"
If [getGen[HiggsBoson] > 1, "mS = <>SPhenoMassSq[HiggsBoson,iS]", "mS = <>SPhenoMassSq[HiggsBoson,iS]"
C2LR = C2LR−HL*Conjg(HR)/(mS*norm)
C1SLL = C1SLL−0.5_dp*HL*conjg(HR)/(mS*norm)
C1SRR = C1SRR−0.5_dp*HR*conjg(HR)/(mS*norm)
If [getGen[HiggsBoson] > 1, "End Do", ""]

If [getGen[PseudoScalar] > 1, "Do iS = <>toString [getGenSPhenoStart[PseudoScalar]<>, ""]
If [getGen[PseudoScalar] > 1, "AL = OAh2qSL(3,3,iS)", "AL = OAh2qSL(3,3,iS)"
If [getGen[PseudoScalar] > 1, "AR = OAh2qSR(3,3,iS)", "AR = OAh2qSR(3,3,iS)"
If [getGen[PseudoScalar] > 1, "mS = <>SPhenoMassSq[PseudoScalar,iS]", "mS = <>SPhenoMassSq[PseudoScalar,iS]"
C2LR = C2LR−AL*Conjg(AR)/(mS*norm)
C1SLL = C1SLL−0.5_dp*AL*conjg(AR)/(mS*norm)
C1SRR = C1SRR−0.5_dp*AR*conjg(AR)/(mS*norm)
If [getGen[PseudoScalar] > 1, "End Do", ""]

Ftt = S0xt(xt) + CVLLnew/(4._dp*r) + & 
& CVRR/(4._dp*r) + P1bLR+C1LR + P2bLR+C2LR + & 
& P1bSSL*(C1SLL + C1SRR) + P2bSSL*(C2SLL + C2SRR)

If (i1 .eq. 1) Then ! Delta M_Bd
ratioDeltaMBq = Abs(Ftt/S0xt(xt))
DeltaMBq = G_F*2*mw2/(6._dp*Pi**2) * & 
& MBq*etaB*BBq*FBq2*Veff2*abs(Ftt)*1.e−12_dp/hbar
Else ! Delta M_Bs
ratioDeltaMBs = Abs(Ftt/S0xt(xt))
DeltaMBs = G_F*2*mw2/(6._dp*Pi**2) * & 
& MBq*etaB*BBq*FBq2*Veff2*abs(Ftt)*1.e−12_dp/hbar
End if
End Do
C.2.8: $\Delta M_K$ and $\varepsilon_K$

$\Delta M_K$ and $\varepsilon_K$, the observables associated to $K^0 - \bar{K}^0$ mixing, can be written as [9, 134]

$$\Delta M_K = 2 \text{Re} \langle K^0 | H_{\text{eff}}^{S=2} | K^0 \rangle$$

$$\varepsilon_K = -\frac{e^{\pi/4}}{\sqrt{2} \Delta M_K} \text{Im} \langle K^0 | H_{\text{eff}}^{S=2} | K^0 \rangle.$$  \hspace{1cm} (C.92)

The matrix element in Eqs. (C.92) and (C.93) is given by

$$\langle K^0 | H_{\text{eff}}^{S=2} | K^0 \rangle = f_V \left( D_{LL}^V + D_{RR}^V \right) + f_S \left( D_{LL}^S + D_{RR}^S \right) + f_T \left( D_{LL}^T + D_{RR}^T \right),$$

$$+ f_{LR} \left( D_{LR}^V + D_{RL}^V \right) + f_{LR}^2 \left( D_{LR}^V + D_{RL}^V \right).$$ \hspace{1cm} (C.94)

The $f$ coefficients are

$$f_V = \frac{1}{3} m_K f_K B_1^{V,LL}(\mu)$$

$$f_S = -\frac{5}{24} \left( \frac{m_K}{m_s + m_d(\mu)} \right)^2 m_K f_K B_1^{S,LL}(\mu)$$

$$f_T = -\frac{1}{2} \left( \frac{m_K}{m_s(\mu) + m_d(\mu)} \right)^2 m_K f_K B_1^{S,LR}(\mu)$$

$$f_{LR} = -\frac{1}{6} \left( \frac{m_K}{m_s(\mu) + m_d(\mu)} \right)^2 m_K f_K B_2^{LR}(\mu)$$

$$f_{LR}^2 = \frac{1}{4} \left( \frac{m_K}{m_s(\mu) + m_d(\mu)} \right)^2 m_K f_K B_2^{LR}(\mu)$$ \hspace{1cm} (C.95)

where $\mu = 2$ GeV is the energy scale at which the matrix element is computed and $f_K$ the Kaon decay constant. The values of the quark masses at $\mu = 2$ GeV are given by $m_d(\mu) = 7$ MeV and $m_s(\mu) = 125$ MeV (see table 1 in [98]), whereas the $B_1^{V}$ coefficients have the following values at $\mu = 2$ GeV [136]: $B_1^{V,LL}(\mu) = 0.61$, $B_1^{S,LL}(\mu) = 0.76$, $B_1^{S,LR}(\mu) = 0.51$, $B_1^{LR}(\mu) = 0.96$ and $B_2^{LR}(\mu) = 1.3$.

As in [9], we treat the SM contribution separately. We define $D_{LL}^V = D_{LL}^{V,SM} + D_{LL}^{V,BSM}$. For $D_{LL}^{V,BSM}$ one just subtracts the SM contributions to $D_{LL}^V$, whereas for $D_{LL}^{V,SM}$ one can use the results in [136–138], where the relevant QCD corrections are included,

$$D_{LL}^{V,SM} = \frac{G_F m_W^2}{4\pi^2} \left[ \lambda_1^2 \eta_1 S_0(x_c) + \lambda_2^2 \eta_2 S_0(x_t) + 2\lambda_3^2 \eta_3 S_0(x_e, x_t) \right].$$ \hspace{1cm} (C.100)

Here $x_i = m_i^2/m_w^2$, $\lambda_i = V_{us}^* V_{cd}$ and $S_0(x)$ and $S_0(x, y)$ are the Inami-Lim functions [133]. $S_0(x)$ was already defined in Eq. (C.80), whereas $S_0(x_c, x_t)$ is given by [134]

$$S_0(x_c, x_t) = x_c \left[ \frac{x_t}{x_c} - \frac{3x_t}{4(1-x_t)} - \frac{3x_t^2 \log x_t}{4(1-x_t)^2} \right].$$ \hspace{1cm} (C.101)

In the last expression we have kept only terms linear in $x_c \ll 1$. Finally, the $\eta_i$ coefficients comprise short distance QCD corrections. Their numerical values are $\eta_{1,2,3} = (1.44, 0.57, 0.47)$ [138].

Note that we have chosen a value for $\eta_1$ which results from our numerical values for $\alpha_s(m_Z)$ and $m_c(m_c)$, see table 5 in [138].
Listing 44  KKmix.m

NameProcess = "KKmix";
NameObservables = {{DeltaMK, 9100, "Delta(M_K)"},
{ratioDeltaMK, 9102, "Delta(M_K)/Delta(M_K)_SM"},
{epsK, 9103, "epsilon_K"},
{ratioepsK, 9104, "epsilon_K/epsilon_K^SM"}};

NeededOperators = {O4dSLL, O4dSRR, O4dSLR, O4dSLR, O4dVRR, O4dVRL, O4dVLR, O4dVLL, O4dTLR, O4dTRL, O4dTRR,
O4dSLLSM, O4dSRRSM, O4dSRLSM, O4dSLRSM, O4dVRRSM, O4dVLLSM, ←
O4dVRLSM, O4dVLLSM};

7

Body = "KKmix.f90 ";
Listing 45  KKmix.f90

Real(dp) :: b_VLL, b_SLL1, b_SLL2, b_LR1, b_LR2
Real(dp) :: ms_mu, md_mu
Complex(dp) :: CVLL, CVRR, CSLL, CSR, CTL, CTR, CLR1, CLR2
Complex(dp) :: fV, fS, fT, fLR1, fLR2, cVLLSM
Complex(dp) :: fK, M_K, H2eff, DeltaMK_SM, epsK_SM
Real(dp) :: norm, hbar, xt, xc, GF
Integer :: i1
Real(dp), Parameter :: eta_tt = 0.57_dp, eta_ct = 0.47_dp, &
& eta_cc = 1.44_dp
! Parameters from S. Herrlich and U. Nierste NPB 476 (1996) 27
! −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
! Delta M_K and epsilon_K
! Observables implemented by W. Porod, F. Staub and A. Vicente
! Based on A. Crivellin et al, Comput. Phys. Commun. 184 (2013) 1004 [ arXiv :1203.5023]
! −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
! using globally defined hadronic parameters
M_K = mass_K0
xt = mf_u(3)**2 / mW**2
xc = mf_u(2)**2 / mW**2
GF = (Alpha_160 * 4._dp * Pi/sinW2_160)/mw**2 * sqrt2 / 8._dp
! Coefficients at mu = 2 GeV
! See A. J. Buras et al, NPB 605 (2001) 600 [hep-ph/0102316]
b_VLL = 0.61_dp
b_SLL1 = 0.76_dp
b_SLL2 = 0.51_dp
b_LR1 = 0.96_dp
b_LR2 = 1.3_dp
! Quark mass values at mu = 2 GeV
! See M. Ciuchini et al, JHEP 9810 (1998) 008 [hep-ph/9808328] – Table 1
md_mu = 0.007_dp
ms_mu = 0.125_dp
IV = 1._dp / 3._dp * M_K / f_k**2 * b_VLL
fS = -5._dp / 24._dp * M_K / f_k**2 * (M_K / (ms_mu + md_mu))**2 * b_SLL1
fT = -1._dp / 2._dp * M_K / f_k**2 * (M_K / (ms_mu + md_mu))**2 * b_SLL2
fLR1 = -1._dp / 6._dp * M_K / f_k**2 * (M_K / (ms_mu + md_mu))**2 * b_LR1
fLR2 = 1._dp / 4._dp * M_K / f_k**2 * (M_K / (ms_mu + md_mu))**2 * b_LR2
! SM contribution
! Based on the results by S. Herrlich and U. Nierste
! NPB 419 (1994) 292, PRD 52 (1995) 6505 and NPB 476 (1996) 27
cVLLSM = eta_cc * ( Conjg(CKM_160(2,2)) * CKM_160(2,1))**2 * S0xt(xc) &
& + eta_tt * ( Conjg(CKM_160(3,2)) * CKM_160(3,1))**2 * S0xt(xt) &
Although $P \rightarrow \ell \nu$, where $P = q q'$ is a pseudoscalar meson, does not violate quark flavor, we have included it in the list of observables for practical reasons, as it can be computed with the same ingredients as the QFV observables. The decay width for the process $P \rightarrow \ell \alpha \nu$ is given by [139]

$$
\Gamma(P \rightarrow \ell \alpha \nu) = \left| G_{Pf} f_p (m_P^2 - m_{\ell \alpha}^2) \right|^2 \frac{8 \pi m_P}{2 m_P^2} \sum_{\nu} \left| V_{q \nu} m_{\ell \alpha} + \frac{m_{\ell \alpha}}{2 \sqrt{2}} (G_{L \ell \nu}^V - G_{R \ell \nu}^V) + \frac{m_P^2}{2 \sqrt{2} (m_q + m_{q'}) (G_{LR}^S - G_{LR}^V)} \right|^2.
$$
Here $f_P$ is the meson decay constant, $m_q$ and $m_q'$ are the masses of the quarks in the meson and the Wilson coefficients $G_{XY}$ are defined in Eq.(A.16). The sum in Eq.(C.102) is over the three neutrinos (whose masses are neglected).

Each $P \to \ell_\alpha \nu$ decay width is plagued by hadronic uncertainties. However, by taking the ratios

$$R_P = \frac{\Gamma(P \to e\nu)}{\Gamma(P \to \mu\nu)}$$

the hadronic uncertainties cancel out to a good approximation, allowing for a precise theoretical determination. In case of $R_K$, the SM prediction includes small electromagnetic corrections that account for internal bremsstrahlung and structure-dependent effects [140]. This leads to an impressive theoretical uncertainty of $\delta R_K/R_K \sim 0.1\%$, making $R_P$ the perfect observable to search for lepton flavor universality violation [141].

**Listing 46 Plnu.m**

```plaintext
NameProcess = "Plnu";
NameObservables = {{BrDmunu, 300, "BR(D->mu nu)",
    {ratioDmunu, 301, "BR(D->mu nu) / BR(D->mu nu)_SM"},
    {BrDsmunu, 400, "BR(Ds->mu nu)",
    {ratioDsmunu, 401, "BR(Ds->mu nu) / BR(Ds->mu nu)_SM"},
    {BrBmunu, 500, "BR(B->mu nu)",
    {ratioBmunu, 501, "BR(B->mu nu) / BR(B->mu nu)_SM"},
    {BrBtaunu, 502, "BR(B->tau nu)",
    {ratioBtaunu, 503, "BR(B->tau nu) / BR(B->tau nu)_SM"},
    {BrKmunu, 600, "BR(K->mu nu)",
    {ratioKmunu, 601, "BR(K->mu nu) / BR(K->mu nu)_SM"},
    {RK, 602, "R_K = BR(K->e nu) / (K->mu nu)_SM"},
    {RKSM, 603, "R_K^SM = BR(K->e nu)_SM / (K->mu nu)_SM")};
NeededOperators = {OdulvSLL, OdulvSRL, OdulvSLR, OdulvVRL, OdulvVRR, OdulvVLL, OdulvVLR, OdulvVRS, OdulvVLRs, OdulvVLLs, OdulvVRLs, OdulvVRLsm};
Body = "Plnu.f90";
```

**Listing 47 Plnu.f90**

```plaintext
Integer :: gt1, gt2, i1, i2, iP
Complex(dp) :: br, br_SM
Real(dp) :: m_M, f_M, tau_M, mlep, mq1, mq2, hbar, ratio, &
& BrKenu_SM, BRKenu, QED

! P -> l nu
! Observable implemented by W. Porod, F. Staub and A. Vicente
! Based on J. Barranco et al., arXiv:1303.3896
hbar = 6.58211889e-25_dp
! Electromagnetic correction to R_K
! See V. Cirigliano, I. Rosell, PRL 99 (2007) 231801 [arXiv:0707.3439]
QED = -3.6e-2_dp
! meson parameters
Do iP=1,4
  If (iP .eq.1) Then ! Ds-meson
    gt1 = 2
    gt2 = 2
  m_M = mass_Dsp
  f_M = f_Dsp_CONST
```

tau_M = tau_DSp/hbar
Elseif (iP .eq.2) Then ! B-meson
  gt1 = 3
gt2 = 1
m_M = mass_Bp
f_M = f_Bp_CONST
tau_M = tau_Bp/hbar
Elseif (iP .eq.3) Then ! Kaon
  gt1 = 2
gt2 = 1
m_M = mass_Kp
f_M = f_Kp_CONST
tau_M = tau_Kp/hbar
End if
mq1 = mf_u_160(gt2)
mq2 = mf_d_160(gt1)
Do i1=1,3
  br = 0._dp
  br_SM = 0._dp
  mlep = mf_l(i1)
  Do i2=1,3
    br = br + ((OdulvVLL(gt1 , gt2 , i1 , i2 ) - OdulvVLR(gt1 , gt2 , i1 , i2 ))*mlep/ &
    & (2._dp*sqrt2) &
    & + m_M*2*(OdulvSRL(gt1 , gt2 , i1 , i2 ) - OdulvSLL(gt1 , gt2 , i1 , i2 ))/ &
    & (2._dp*sqrt2*(mq1+mq2)))
br_SM = br_SM+ (OdulvVLLSM(gt1 , gt2 , i1 , i2 ) - OdulvVLRSM(gt1 , gt2 , i1 , i2 )) &
    & *mlep/(2._dp*sqrt2)
  End Do
ratio = Abs(br/br_SM)**2
br = oo8pi*tau_M*(f_M)**2*M_M*Abs(br)**2*(1._dp - mlep**2/M_M**2)**2 ! G_F already \rightleftarrow \leftin coefficients included
If (iP .eq.1) Then !! Ds-meson
  If ( i1 .eq.2) Then ! Ds->munu
    BrDsmunu = br
    ratioDsmunu = ratio
  Elseif ( i1 .eq.3) Then ! Ds->taunu
    BrDstaunu = br
    ratioDstaunu = ratio
  End if
Elseif (iP .eq.2) Then ! B-meson
  If ( i1 .eq.2) Then ! B->munu
    BrBmunu = br
    ratioBmunu = ratio
  Else ! B->taunu
    BrBtaunu = br
    ratioBtaunu = ratio
  End if
Else If (iP .eq.3) Then !! Kaon
  If ( i1 .eq.1) Then ! K->e nu
    BrKenu = br
    BrKenuSM = BrKenu*ratio
  Elseif (i1 .eq.2) Then ! K->munu
    BrKmunu = br
    ratioKmunu = ratio
    RK = BrKenu/BrKmunu*(1+QED)
    RKSM = BrKenuSM/BrKmunu*ratio*(1+QED)
  End if
D: Models

The following models are included in the public version of SARAH and can now be used together with the FlavorKit to get predictions for the different observables.

D.1: Supersymmetric Models

- Minimal supersymmetric standard model (see Ref. [142] and references therein)
  - With general flavor and CP structure (MSSM)
  - Without flavor violation (MSSM/NoFV)
  - With explicit CP violation in the Higgs sector (MSSM/CPV)
  - In SCKM basis (MSSM/CKM)
- Singlet extensions:
  - Next-to-minimal supersymmetric standard model (NMSSM, NMSSM/NoFV, NMSSM/CPV, NMSSM/CKM) (see Refs. [143,144] and references therein)
  - near-to-minimal supersymmetric standard model (near-MSSM) [145]
  - General singlet extended, supersymmetric standard model (SMSSM) [145,146]
  - DiracNMSSM (DiracNMSSM) [147,148]
- Triplet extensions
  - Triplet extended MSSM (TMSSM) [149]
  - Triplet extended NMSSM (TNMSSM) [150]
- Models with R-parity violation [151–158]
  - bilinear RpV (MSSM-RpV/Bi)
  - Lepton number violation (MSSM-RpV/LnV)
  - Only trilinear lepton number violation (MSSM-RpV/TriLnV)
  - Baryon number violation (MSSM-RpV/BnV)
  - µνSSM (munuSSM) [159,160]
- Additional $U(1)'s$
  - $U(1)$-extended MSSM (UMSSM) [145]
  - secluded MSSM (secluded-MSSM) [161]
  - minimal $B−L$ model (B−L-SSM) [162–165]
  - minimal singlet-extended $B−L$ model (N-B−L-SSM)
- SUSY-scale seesaw extensions
  - inverse seesaw (inverse-Seesaw) [166,167]
  - linear seesaw (LinSeesaw) [166,168]
  - singlet extended inverse seesaw (inverse-Seesaw-NMSSM) [169]
  - inverse seesaw with $B−L$ gauge group (B−L-SSM-IS) [170]
  - minimal $U(1)_{R}\times U(1)_{B−L}$ model with inverse seesaw (BLinvSeesaw) [74,171]
- Models with Dirac Gauginos
  - MSSM/NMSSM with Dirac Gauginos (DiracGauginos) [172–174]
  - minimal R-Symmetric SM (MRSSM) [175]
  - Minimal Dirac Gaugino supersymmetric standard model (MDGSSM) [86]
- High-scale extensions
  - Seesaw 1 - 3 (SU(5) version), (Seesaw1,Seesaw2,Seesaw3) [63, 65, 68, 176, 177]
  - Left/right model (LR) (Omega) [178,179]
  - Quiver model (QEW12, QEW1d2L3) [180]
D.2: Non-Supersymmetric Models

- Standard Model (SM) (8M). Standard model in CKM basis (SM/CKM) (see for instance Ref. [181] and references therein)
- inert Higgs doublet model (Inert) [182]
- B-L extended SM (B-L-SM) [183–185]
- B-L extended SM with inverse seesaw (B-L-SM-IS) [186]
- SM extended by a scalar color octet (SM-SC) [187]
- Two Higgs doublet model (THDM) (see for instance Ref. [188] and references therein)
- Singlet extended SM (SSM) [189]
- Singlet Scalar DM (SSDM) [190]

References

1. ATLAS Collaboration, G. Aad et al., Phys.Lett. B716 (2012), 1–29, [1207.7214].
2. CMS Collaboration, S. Chatrchyan et al., Phys.Lett. B716 (2012), 30–61, [1207.7235].
3. ATLAS Collaboration, CMS Collaboration, D0 Collaboration, (2014), 1403.4427.
4. D. Buttazzo, G. Degrassi, P. Giardino, I. G. Giudice, F. Sala, et al., JHEP 1312 (2013), 089, [1307.3536].
5. F. Mahmoudi, Comput.Phys.Commun. 178 (2008), 745–754, [0710.2067].
6. F. Mahmoudi, Comput.Phys.Commun. 180 (2009), 1579–1613, [0808.3144].
7. F. Mahmoudi, Comput.Phys.Commun. 180 (2009), 1718–1719.
8. J. Rosiek, F. Chankowski, A. Dedes, S. Jaeger, and P. Tanedo, Comput.Phys.Commun. 181 (2010), 2180–2205, [1003.4260].
9. A. Crivellin, J. Rosiek, F. Chankowski, A. Dedes, S. Jaeger, et al., Comput.Phys.Commun. 184 (2013), 1004–1032, [1203.5023].
10. U. Ellwanger and C. Hugonie, Comput.Phys.Commun. 177 (2007), 399–407, [hep-ph/0612134].
11. G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, Comput.Phys.Commun. 149 (2002), 103–120, [hep-ph/0112278].
12. G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, Comput.Phys.Commun. 174 (2006), 577–604, [hep-ph/0405253].
13. G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, Comput.Phys.Commun. 176 (2007), 367–382, [hep-ph/0607059].
14. G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, Comput.Phys.Commun. 180 (2009), 747–767, [0803.2360].
15. G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, Comput.Phys.Commun. 185 (2014), 960–985, [1305.0237].
16. G. Degrassi, P. Gambino, and P. Slavich, Comput.Phys.Commun. 179 (2008), 759–771, [0712.3265].
17. B. Murakami, Comput.Phys.Commun. 185 (2014), 622–637, [1302.4469].
18. D. Chowdhury, R. Garani, and S. K. Vempati, Comput.Phys.Commun. 184 (2013), 899–918, [1109.3551].
19. F. E. Paige, S. D. Protopopescu, H. Baer, and X. Tata, (2003), hep-ph/0312045.
20. H. Baer, C. Balazs, A. Belyaev, R. Dermisek, A. Mafi, et al., Nucl.Instrum.Meth. A502 (2003), 560–563.
21. H. Baer, F. E. Paige, S. D. Protopopescu, and X. Tata, (1999), hep-ph/0001086.
22. F. E. Paige, S. D. Protopopescu, H. Baer, and X. Tata, (1998), hep-ph/9804321.
23. H. Baer, F. E. Paige, S. D. Protopopescu, and X. Tata, (1993), hep-ph/9305342.
24. W. Porod, Comput.Phys.Commun. 153 (2003), 275–315, [hep-ph/0301110].
25. W. Porod and F. Staub, Comput.Phys.Commun. 183 (2012), 2458–2469, [1104.1573].
26. F. Staub, (2008), 0806.0538.
27. F. Staub, Comput.Phys.Commun. 181 (2010), 1077–1086, [0909.2863].
28. F. Staub, Comput.Phys.Commun. 182 (2011), 808–833, [1002.0840].
29. F. Staub, Computer Physics Communications 184 (2013), pp. 1792–1809, [1207.0906].
30. F. Staub, Comput.Phys.Commun. 185 (2014), 1773–1790, [1309.7223].
31. F. Staub, Nucl.Phys.Proc.Supp. 209 (2019), 1283–1286, [1809.00977].
32. F. Staub, Nucl.Phys.Proc.Suppl. 240 (2014), 231–236, [hep-ph/0005029].
33. F. Staub, Nucl.Phys.Proc.Suppl. 135 (2004), 333–337, [hep-ph/0406288].
34. S. T. Hahn, eConf C050318 (2005), 0604, [hep-ph/0506201].
35. F. T. Hahn, Nucl.Phys.Proc.Suppl. 90 (1995), 337–354, [hep-ph/9503351].
36. S. T. Hahn, Nucl.Phys.Proc.Suppl. 89 (2000), 251–258, [hep-ph/0010029].
37. S. T. Hahn, Nucl.Phys.Proc.Suppl. 135 (2004), 333–337, [hep-ph/0406288].
38. S. T. Hahn, eConf C050318 (2005), 0604, [hep-ph/0506201].
39. B. C. Nejad, T. Hahn, J. N. Lang, and E. Mirabella, (2013), 1310.0274.
40. E. Ma and A. Pranumultha, Phys.Rev. D24 (1981), 1410.
41. T. Hisano, T. Moroi, K. Tobe, and M. Yamaguchi, Phys.Rev. D53 (1996), 2442–2459, [hep-ph/9510309].
42. A. Bednyakov and S. Tanyildiz, (2013), 1311.5546.
43. T. Ohl, Comput.Phys.Commun. 90 (1995), 340–354, [hep-ph/9503351].
44. F. T. Hahn, T. Ohl, W. Porod, and C. Speckner, Comput.Phys.Commun. 183 (2012), 2165–2206, [1109.5147].
45. A. Crivellin, L. Hofer, and J. Rosiek, JHEP 1107 (2011), 017, [1103.4272].
46. A. J. Buras, M. Jamin, and P. H. Weisz, Nucl.Phys. B347 (1990), 491–536.
183. W. Emam and S. Khalil, Eur.Phys.J. C52 (2007), 625–633, [0704.1395].
184. L. Basso, A. Belyaev, S. Moretti, and C. H. Shepherd-Themistocleous, Phys.Rev. D80 (2009), 055030, [0812.4313].
185. L. Basso, A. Belyaev, S. Moretti, and G. M. Pruna, JHEP 0910 (2009), 006, [0903.4777].
186. S. Khalil, Phys.Rev. D82 (2010), 077702, [1004.0613].
187. H. H. Patel, M. J. Ramsey-Musolf, and M. B. Wise, Phys.Rev. D88 (2013), no. 1, 015003, [1303.1140].
188. G. Branco, P. Ferreira, L. Lavoura, M. Rebelo, M. Sher, et al., Phys.Rept. 516 (2012), 1–102, [1106.0034].
189. D. O’Connell, M. J. Ramsey-Musolf, and M. B. Wise, Phys.Rev. D75 (2007), 037701, [hep-ph/0611014].
190. A. Goudelis, Y. Mambrini, and C. Yaguna, JCAP 0912 (2009), 008, [0909.2799].