THE EUROPEAN
PHYSICAL JOURNAL C

CrossMark

ROSETTA: an operator basis translator for standard model effective field theory

Adam Falkowski[1], Benjamin Fuks[2], Kentarou Mawatari[3], Ken Mimasu[4,a], Francesco Riva[5], Verónica Sanz[4]

[1] Laboratoire de Physique Théorique, Bat. 210, Université Paris-Sud, 91405 Orsay, France
[2] Département Recherches Subatomiques, Institut Pluridisciplinaire Hubert Curien, Université de Strasbourg/CNRS-IN2P3, 23 rue du Loess, 67037 Strasbourg, France
[3] Theoretische Natuurkunde and IIHE/ELEM, Vrije Universiteit Brussel, and International Solvay Institutes, Pleinlaan 2, 1050 Brussels, Belgium
[4] Department of Physics and Astronomy, University of Sussex, Brighton BN1 9QH, UK
[5] CERN, Theory Division, 1211 Geneva, Switzerland

Abstract We introduce ROSETTA, a program allowing for the translation between different bases of effective field theory operators. We present the main functions of the program and provide an example of usage. One of the Lagrangians which ROSETTA can translate into has been implemented into FEYNRULES, which allows ROSETTA to be interfaced into various high-energy physics programs such as Monte Carlo event generators. In addition to popular bases choices, such as the Warsaw and Strongly Interacting Light Higgs bases already implemented in the program, we also detail how to add new operator bases into the ROSETTA package. In this way, phenomenological studies using an effective field theory framework can be straightforwardly performed.

1 Introduction

The start of a second LHC experimental era raises new hopes to detect physics beyond the standard model (BSM). The high energy of the experiment increases the chances of a direct discovery of new physics resonances, while a combination of high energy and high luminosity favors the possible observation of new phenomena via standard model (SM) precision tests. Interestingly the latter offers a complementary and model-independent tool for BSM searches if the results are interpreted in the context of an effective field theory (EFT). The EFT indeed captures in a general way the low-energy effects of heavy new physics from a bottom-up perspective. More precisely, it systematically organizes possible departures from the SM as an expansion in the energy at which the processes of interest occur over the (high) new physics scale, and simultaneously provides a dictionary to

interpret these departures in the context of explicit BSM models.

Given the SM field content (including a single Higgs doublet), assuming baryon and lepton number conservation, flavor universality and a linear realization of the electroweak symmetry, the leading effects implied by an EFT description consist of dimension-six operators that are supplemented to the SM Lagrangian. At this order, 59 (76 real) new independent coefficients [1,2][1] capture all possible deformations from the SM. Despite this large number of new free parameters, important classes of observables (e.g., Higgs production and decay or Z-pole observables) depend on a much smaller subset of parameters [4–9]. Owing to that, the EFT approach is not only useful for parameterizing BSM searches but is also testable per se by looking at correlations among the expected signatures.

Another important aspect of the EFT approach is the choice of the operator basis, so that a given physical effect could be modeled by different combinations of operators at a fixed order in the EFT expansion. This well-known fact is related to the possibility of redefining the SM fields in such a way that the zeroth order Lagrangian in the EFT expansion (i.e., the SM Lagrangian) is unaltered, while combinations of the first-order operators (i.e., dimension-six operators) proportional to the SM equations of motion can be eliminated up to subleading higher-dimensional effects. For this reason, different complete and non-redundant operator bases have been proposed in the literature, sharing the same physical predictions but having different advantages. The most popular choices include the so-called *Warsaw* basis [2], SILH (strongly interacting light Higgs) basis [10,11] and BSM *pri-*

a e-mail: k.mimasu@sussex.ac.uk

---

[1] Relaxing flavor universality, the number of independent dimension-six operators grows to 2499 [3].

Table 1 Main features of the different EFT basis choices discussed in this document

Basis	Underlying gauge symmetry	Fields used in the Lagrangian
Warsaw, SILH	$SU(3)_C \times SU(2)_L \times U(1)_Y$	Gauge-eigenstates
BSM primaries, Higgs	$SU(3)_C \times SU(2)_L \times U(1)_Y$	Mass-eigenstates
Higgs/BSM characterisation	$SU(3)_C \times U(1)_{EM}$	Mass-eigenstates

maries basis [6,12,13]. The Warsaw basis represents the first set of non-redundant operators that has been proposed and is particularly appropriate for comparisons with BSM theories that modify the interactions of the SM fermions. In contrast, the SILH basis has been designed to capture the effects of universal theories where new physics mostly couples to the SM bosons. Finally, the BSM primaries basis is more suitable for a bottom-up approach since it is formulated in terms of mass-eigenstates and has a more transparent connection to measurable quantities, its operators being aligned with physical observables.

Given these multiple viewpoints, it is cumbersome to express the experimental results in a basis-independent manner that can be readily interpreted in any of the above-mentioned frameworks. On the other hand, different bases may be convenient for particular applications, either because they facilitate the comparison with a given class of BSM theories or simply because different experimental analyses look more transparent in a specific basis. For instance, the Warsaw basis contains an apparent blind direction with respect to the electroweak precision tests [6,14], which introduces large theoretical correlations among all LEP constraints. As a result, the bounds on the strength of the dimension-six interactions appear less transparent [15]. The SILH basis has a similar drawback yielding a correlation between LEP2 and LHC constraints, while the downside of the BSM primaries basis lies in the comparison with explicit BSM models that is complicated. The ROSETTA package that we present in this paper has been designed to explicitly solve such problems by allowing for a straightforward translation between different EFT languages.

In addition to translating, another important goal of the ROSETTA program is to provide a platform for communication with Monte Carlo event generators, no matter which EFT basis is chosen. To achieve this, we have implemented in ROSETTA the *Higgs* basis for EFT operators that has been recently designed by the LHC Higgs Cross Section working group (LHCHXSWG) [16]. This proposal, built on the BSM primaries basis (see Ref. [13]), combines two ingredients. First, all possible operators of dimension up to six are written in terms of the SM mass-eigenstates

$$\Delta\mathcal{L}^{(\text{mass})} = \sum_i \frac{c_i}{\Lambda^{d_i}} \mathcal{O}_i \left(G_\mu^a, W_\mu^\pm, Z_\mu, A_\mu, h, t, b, \nu_\tau, \tau, \ldots\right), \quad (1)$$

where the operators $\mathcal{O}_i$ have a mass dimension ranging from two to six. The dimensionless coefficients c_i are then suppressed by an appropriate power d_i of the high-energy scale Λ, with $d_i = -2, \ldots, 2$. We refer the ensemble of operators included in the resulting Lagrangian, which is in spirit very similar to the Higgs characterisation Lagrangian of Ref. [17], as the BSM *characterisation* (BSMC) Lagrangian. Due to the lack of manifest $SU(2)_L \times U(1)_Y$ invariance, the BSMC Lagrangian is associated with a larger number of independent coefficients compared to the Warsaw, SILH or BSM primaries bases. For this reason, the second ingredient defining the Higgs basis consists of relations among the c_i coefficients that restore the full $SU(3)_C \times SU(2)_L \times U(1)_Y$ symmetry. As summarized in Table 1, the BSM primaries and Higgs Lagrangians are both of the form of $\Delta\mathcal{L}^{(\text{mass})}$, but they additionally include constraints among the different Wilson coefficients that render the Lagrangian invariant under the electroweak symmetry. In contrast, the Warsaw and SILH basis Lagrangians are directly written in terms of the SM gauge-eigenstates,

$$\Delta\mathcal{L}^{(\text{gauge})} = \sum_{i'} \frac{c_i'}{\Lambda^2} \mathcal{O}_{i'} \left(G_\mu^a, W_\mu^i, B_\mu, \Phi, Q_L, u_R, d_R, L_L, e_R\right), \quad (2)$$

and are manifestly symmetric under the electroweak symmetry group.

We have implemented the mass basis Lagrangian $\Delta\mathcal{L}^{(\text{mass})}$ into FEYNRULES [18][2] and tuned the output format of ROSETTA so that the translation maps an EFT Lagrangian given in a specific basis to $\Delta\mathcal{L}^{(\text{mass})}$ and generates an output file that is compatible with the FEYNRULES implementation. As a consequence, any high-energy physics tool (and in particular any Monte Carlo event generator) that is interfaced to FEYNRULES can be employed within the context of any EFT basis of operators that is included in ROSETTA.

With the advent of automated next-to-leading order (NLO) accurate Monte Carlo event generation software, it is important that ROSETTA remains flexible enough to eventually provide compatibility with this new generation of tools. Recent progress has been made on the theory side both in implement-

---

ing the linear dimension-six description discussed above in the FEYNRULES framework [20] and in calculating the renormalisation group (RG) evolution of the full set of operators and their mutual mixing [3,21–23]. In the former case, ROSETTA can simply be extended to provide an output compatible with the eventual NLO model implementation, analogously to the BSMC Lagrangian. The latter case of evaluating the RG running effects, while being a slightly separate issue, highlights a key feature of our tool, given that the calculation of these effects has only been performed in the original Warsaw basis of Ref. [2]. The framework provided by ROSETTA allows for the application of these results in any desired basis. Although the initial version of the software does not explicitly deal with these effects, its translation functionality can already be used in their context and we plan for future versions to incorporate RG evolution of the SM EFT Wilson coefficients.

The remainder of this paper is organized as follows. In Sect. 2, we describe the basic functionalities of ROSETTA and how to make use of the program. Sect. 3 is dedicated to an example of usage of ROSETTA in which we focus on new physics Higgs couplings to the SM bosons. We express them in different bases and detail the output that is provided by ROSETTA. Our work is summarized in Sect. 4.

2 Rosetta

The aim of ROSETTA is to provide a modular and flexible package for EFT basis translation and communication with event generation tools. The primary framework which ROSETTA has been designed to translate into is the phenomenological effective Lagrangian, $\Delta\mathcal{L}^{(\text{mass})}$, which will be explicitly defined in Sect. 3.1. The motivation for this choice lies in the availability of an implementation within the FEYNRULES framework [18], to be downloaded from the FEYNRULES model repository [24], which ensures the link with event generators and high-energy physics programs [25,26].

The most basic functionality of ROSETTA is to map a chosen set of input parameters (the Wilson coefficients in a specific basis choice) onto the BSMC coefficients such that the output can be employed within tools relying on a BSMC basis description. As long as the input format respects the conventions sketched in Sect. 2.2 and that are inspired by the Supersymmetry Les Houches Accord (SLHA) [27,28], the user may define his/her own map to the BSMC coefficients (or to any other basis implementation) and proceed with event generation using the accompanying FEYNRULES implementation. This highlights one of the key features of ROSETTA, the possibility to easily define one's own input basis and directly use it in the context of many programs via the translation functionality of ROSETTA. The strength of this approach is that it is much simpler than developing from

scratch new modules for existing tools in the context of a new basis. To this end, ROSETTA not only enables the translation of an EFT basis into the BSMC Lagrangian, but also allows for translations into any of the other bases included in the package, i.e., currently the Higgs, Warsaw and SILH bases. Translations between these three bases in any direction are possible, so that the addition of a new basis by the user only requires the specification of translation rules to any one of the three core bases. One is subsequently able to indirectly translate the new basis into any of the other two bases, as well as into the BSMC Lagrangian. The details of how one can implement a new basis in ROSETTA are discussed in Sect. 2.4.

2.1 Getting started with ROSETTA

The latest release of ROSETTA can be obtained from

```
http://rosetta.hepforge.org
```

The package contains a PYTHON executable named `translate`, an information file named `README` and two directories, a first folder (named `Cards`) collecting example input files and a second folder (named `Rosetta`) including the source code of ROSETTA. The executable takes as input an SLHA-style parameter file with the coefficients of the dimension-six operators associated with a particular basis. Information on the format of such an input file can be found in Sect. 2.2. The execution of the `translate` script from a shell yields the generation of an output parameter file where all parameters are this time the coefficients of the dimension-six operators associated with a specified new basis, the default choice being the BSMC Lagrangian. The tool can be used by typing in

```
./translate PARAMCARD.dat OPTIONS
```

where `PARAMCARD.dat` is the name of the SLHA-style input file and `OPTIONS` stands for optional arguments. The latter could consist of one or more of the following choices that will modify the behavior of the program.

`-h` or `–help`	This displays a help message and exits the program.
`-o` or `–output`	This allows for the specification of the name of the output file, that is by default `PARAMCARD_new.dat`.
`-s` or `–silent`	The program suppresses warnings and takes the default answer to any question that may have to be asked to the user.
`-t` or `–target`	This allows for providing the name of the basis into which the translation occurs, the default being `bsmc`

3	+1.18400e-01	# aS
4	+9.11876e+01	# MZ
25	+1.25000e+02	# MH

and the other acceptable choices being `higgs`, `silh` or `warsaw`.

-w or –overwrite	This allows the program to overwrite any pre-existing output file.
-e or –ehdecay	This allows to use the interface with the eHDECAY program [11] for the calculation of the Higgs boson width and branching fractions. See Sect. 2.5.2.
-f or –flavor	This allows to specify the treatment of the flavor structure relevant for the fermionic operators, the default being `general` and the other acceptable choices being `universal` and `diagonal`. See Sect. 2.5.1.
-d or –dependent	This allows the program to also write out any dependent parameters calculated by the translation function to the output file.

On run time, ROSETTA starts by performing several checks on the input parameters and verifies the consistency of the input file with respect to the specifications of the internal basis implementation. In this way, any missing SM inputs (with respect to the requirements included in the `required_inputs` and `required_masses` attributes of the basis class, see Sect. 2.3) can be included using the value provided in the Particle Data Group (PDG) review [29], while any missing coefficient associated with an operator that is present in the basis (and thus declared in the `independent` attribute of the basis class, see Sect. 2.3) can be included with a zero value.

Once the translation is achieved, ROSETTA outputs a new parameter file that is by default named PARAMCARD_new. dat. This file contains the values of all parameters relevant for the target basis and also includes the necessary modifications to the input parameters, such as the W-boson mass that may depend on some dimension-six operator coefficients.

2.2 Input files and their handling in ROSETTA

ROSETTA requires input parameters to be given under the form of a file encoded in a format similar to the SLHA one detailed in Refs. [27,28]. Parameters are grouped into blocks and each parameter is identified inside its own block by one or more integer numbers called counters. For instance, the SM inputs necessary for the definition of the SILH basis would read

```
BLOCK SMINPUTS #
   1     +1.27916e+02 # aEWM1
   2     +1.16638e-05 # Gf
```

where the different entries respectively correspond to the inverse of the electromagnetic coupling constant (`aEWM1`), the Fermi constant (`Gf`), the strong coupling constant (`aS`), the Z-boson mass (`MZ`) and the Higgs boson mass (`MH`). Inspired by the usual SLHA conventions, all masses are also collected into a block called `MASS` where the counters correspond to the PDG identifiers of the particles [29]. Furthermore, matrix quantities receive a block of their own with counters specifying the position inside the matrix. In this way, a single block would be needed to encode, for instance, the c_{Hud} coefficients associated with the $\mathcal{O}_{Hud}$ operator of the Warsaw basis that is defined by

$$\mathcal{O}_{Hud} = -i\left[\bar{u}\gamma^{\mu}d\right]\left[\tilde{\Phi}^{\dagger}D_{\mu}\Phi\right].\tag{3}$$

In this expression, u and d denote the $SU(2)_L$ singlets of right-handed up-type and down-type quark fields, respectively, and Φ and $D_{\mu}\Phi$ stand for the weak doublet of Higgs fields and its gauge-covariant derivative. In flavor space, the c_{Hud} coefficients take the form of a matrix, implemented in the input file as

```
BLOCK WBxHud
1 1 0.1e+00 # cHud11
1 2 0.0e+00 # cHud12
1 3 0.0e+00 # cHud13
2 1 0.0e+00 # cHud21
2 2 0.1e+00 # cHud22
2 3 0.0e+00 # cHud23
3 1 0.0e+00 # cHud31
3 2 0.0e+00 # cHud32
3 3 0.1e+00 # cHud33
```

The block name contains information on the basis (`WB`) and on the considered operator (`Hud`). Sample parameter files for all core bases can be found in the `Cards` directory shipped with the program. Within those files, we have adopted the above block naming scheme. The name of each block starts with two letters identifying the basis (`BC`, `HB`, `SB` and `WB` for the BSMC, Higgs, SILH and Warsaw bases respectively) that are followed by a separator (`x`), and ends with the name of the considered coefficient as it is defined in the LHCHXSWG proposal for an EFT basis choice [16]. In the case of EFT operators independent of fermions, the related (non-matrix) coefficients are collected in different blocks as a function of the Lorentz structure of the operators. For instance, the `SBxV2H2` block will include all operators of the SILH basis containing two occurrences of the Higgs field and two occurrences of the gauge fields. Their ordering follows their order of appearance in the LHCHXSWG proposal. The imaginary

parts of all parameters are stored in corresponding blocks whose names are prefixed with the IM tag.

The ROSETTA package contains built-in methods for dealing with an SLHA-like structure, and these methods have all been implemented in the Rosetta/SLHA.py file. When an input file is read, the parser included in the SLHA.py file recognizes the existing BLOCK and DECAY structures of the input file and stores them as instances of the SLHA.NamedBlock, SLHA.NamedMatrix and of the SLHA.Decay classes. These are dictionary-like objects that can be assigned, indexed and iterated over as regular PYTHON dictionaries. An SLHA.NamedBlock object reflects the information embedded in an SLHA block so that it possesses a name attribute and stores values associated with integer keys as well as a mapping from the integer keys to the parameter names. In this way, parameters can be accessed by indexing either their integer key or their name. Similarly, SLHA.NamedMatrix objects function analogously but operate with a pair of indices for indexing. An SLHA.Decay object contains an integer attribute PID that is the PDG identifier of the particle whose decays are described by the considered block, as well as a total attribute allowing for the storage of the total width. Individual decay channels are then indexed by tuples of PDG codes associated with the decay products, and the stored values are the branching fractions. Finally, the SLHA.py file also includes the definition of the SLHA.Card class that serves as a container for a collection of instances of the above objects. The implementation of any basis in ROSETTA therefore requires the user to provide definitions for the blocks and parameters to be specified in the input file that will be read into an SLHA.Card instance belonging to that basis class. More practical information and examples are given in Sects. 2.3 and 2.4.

Three special blocks named BASIS, SMINPUTS and MASS must always be present. The first and only element of the BASIS block refers the name of the basis into which RO-SETTA must read the input file and informs the program on the other blocks it should look for, based on the structure specified in the implementation of that particular basis. This name should be a single unique string with no spaces. The next two mandatory blocks consist of conventional input blocks specifying the values of the SM inputs and of the particle masses. The set of required inputs will depend on the specifications in the corresponding basis implementation. Moreover, the user can optionally specify the value of the elements of the CKM matrix by setting their real and imaginary parts within the VCKM and IMVCKM blocks. If absent, the information of the PDG review [29] is used by ROSETTA. All extra blocks and decay structures are stored, left unchanged and passed to the output file unless the user demands to use the eHDECAY program, which will overwrite any existing decay information on the Higgs.

2.3 Structure of ROSETTA

ROSETTA is a PYTHON package containing the implementation of a Basis class equipped with several utility functions for reading, processing and writing SLHA-style parameter files. Working implementations of bases are derived from this class and only require a small amount of information specifying the block structure of the EFT parameters, the required SM inputs and a series of translation functions to other existing basis implementations. In order to be able to define a new basis class, we describe in this section the properties of the Basis objects.

The Basis class has a number of intrinsic data members that should be defined in order to get a working implementation of an EFT basis. These consist of the independent, required_inputs and required_masses attributes already mentioned in Sect. 2.1, together with the name, blocks and flavored members of the class.

name	Unique string identifier for the basis implementation, e.g., higgs, bsmc, silh or warsaw for the core bases shipped with the package.
independent	List of strings containing the names of the independent EFT operator coefficients of the basis. These are expected to be present in the input parameter file.
required_inputs	Set of integers containing the SLHA counters of the required SM inputs. See Table 2 for the complete list of those allowed in ROSETTA.
required_masses	Set of integers containing the PDG identifiers of the particles whose masses are required as input and that are not included in required_inputs.
blocks	Dictionary with the non-matrix SLHA block names as keys and lists of coefficients stored in that block as values.
flavored	Dictionary with matrix SLHA block names as keys. The values are other dictionaries with the keywords kind, domain and cname as keys. This describes the properties of the matrices.

In the case of the definition of a matrix block, the self-explanatory possible values for the keyword kind are symmetric, hermitian and general and those related to the keyword domain are real and complex. The prop-

Table 2 Identifying counters of the SMINPUTS block with the SM parameters allowed to be used within ROSETTA. This generalizes the SLHA standards where the Higgs mass is ignored [28]. The internal names used by ROSETTA are also given

Counter	Parameter	ROSETTA name
1	The inverse of the electromagnetic coupling constant α^{-1}	aEWM1
2	The Fermi constant G_F	Gf
3	The strong coupling constant α_s	aS
4	The Z-boson mass m_Z	MZ
5	The bottom quark mass m_b	MB
6	The top quark mass m_t	MT
7	The tau lepton mass m_τ	MTAU
25	The Higgs boson mass m_H	MH

erties of the ensuing matrix object will depend on the choice of these keywords. The name to be given to the individual EFT coefficients are derived from the value of the keyword cname. Conventionally, the real and imaginary components are prefixed with the letters R and I respectively, while the position (i, j) in the matrix is referred to by a suffix ixj. A complex parameter comes with a prefix C.

Once an input file is read, an instance of the SLHA.Card class that can be accessed via the card member of the basis class is created and populated with the information provided as input. The content of the mandatory MASS and SMINPUTS blocks is exported to data members of the basis class named mass and inputs that can then be used for accessing the SM parameters, while the CKM matrix is stored into the ckm container of the basis class. In the ROSETTA framework, the EFT operator coefficients are implemented as elements of the relevant basis class and can be accessed via standard PYTHON methods. For instance, all the coefficients associated with a basis object named MyBasis could be printed, together with their values, by coding

```
for k, v in MyBasis.items():
print k, v
```

In addition, a direct accessor to each EFT operator coefficient is created from its name, which facilitates the implementation of the translation functions that in general extensively reference individual parameter values. This however assumes that there are no duplicate parameter names in the SLHA-like input file, which nevertheless leads to a program exception. There are hence multiple ways to access a given parameter. For example, a parameter A stored as the third element of a block MyBlock that is part of the definition of a basis MyBasis could be equally accessed as

```
MyBasis['A']
MyBasis.card['A']
MyBasis.card.blocks['MyBlock']['A']
MyBasis.card.blocks['MyBlock'][3]
```

In the lines above, the parameter A is respectively accessed from the MyBasis object, from the SLHA.Card instance associated with the current basis and from the SLHA.NamedBlock object associated with the MyBlock block (using either the parameter name or the counter as an index).

2.4 Implementing a new basis

One of the important features of ROSETTA is the intended ease with which a user can define a new basis class to suit his/her specific physics needs. In the context of an ultraviolet complete model, he/she may be interested in the phenomenological consequences of a particular high-scale scenario in the EFT framework. Imagining that he/she has derived all dimension-six Wilson coefficients in a particular basis, ROSETTA could be used to map these coefficients to the FEYN-RULES effective Lagrangian implementation in the mass-eigenstate basis so that the collider phenomenology of such a scenario could be investigated. This task is realized by implementing a new basis in ROSETTA and by connecting the new basis input parameters to those of one of the existing core basis implementations.

Alternatively, the user may have developed a particular resource performing a useful analysis or calculation in a non-standard basis choice. The corresponding basis implementation in ROSETTA with a translation to one of the core bases could then allow one to use this tool in the context of all other existing basis implementations in ROSETTA and therefore greatly widen its scope. The eHDECAY feature of RO-SETTA is an example of this, as it works with a set of operators corresponding to the SILH basis. Via ROSETTA, eHDECAY is now available for calculations in the SILH, Warsaw and Higgs bases, as well as in any additional basis that may be implemented in the future.

In this section, we provide an example that outlines the basic requirements for implementing a new basis in RO-SETTA. We also refer the reader to the file Rosetta/TemplateBasis.py which serves as a concrete toy example that can be used as a template for creating a new basis class as well as the core basis implementations for more complete realizations.

All ROSETTA basis classes inherit from the mother class Basis implemented in the Rosetta/internal/Basis.py file. This class contains all the machinery necessary for reading, writing and translating so that a new basis implementation solely demands the user to create a new file that must be saved in the Rosetta directory and

that includes the declaration of a `Basis` subclass. The user has then to define the class attributes described in Sect. 2.3. First, it is essential that the name of the basis class matches the filename in which it is saved minus the extension in order to ensure a proper running of the program. Second, the `independent`, `blocks` and `flavored` attributes of the class define the input parameters of the basis and their desired SLHA-like structure, while the `required_inputs` and `required_masses` lists are specified according to the needs of the translation functions that are planned to be implemented. One can also specify a `dependent` attribute to explicitly define a particular parameter as dependent. For instance, the following code refers to the implementation of a new basis class called `MyBasis` and has been included in the file `Rosetta/MyBasis.py`.

```
from internal import Basis
class MyBasis(Basis.Basis):
  name = 'mybasis'
  independent=['A','B','one','two','MYxMAT']
  dependent = ['Cmat3x3']
  blocks = {'letters':['A','B','C'],
      'numbers':['one','two','three']}
  flavored = {'MYxMAT':{'kind':'hermitian',
                        'domain':'complex',
                        'cname':'mat'}}
  required_inputs = {1,2,4}
  required_masses = {24,25,6}
```

This snippet of code specifies the declaration of the basis class `MyBasis` whose unique string identifier is given by `mybasis`. The independent parameters to be read from an input SLHA-like file are defined to be A, B, one and two and are assumed to be organized into the two SLHA blocks LETTERS and NUMBERS. A flavored matrix, MYxMAT, is also present and deemed to be an independent input parameter *except* for its (3,3) component that is explicitly included within the `dependent` attribute of the basis class. The translation methods to be implemented require the knowledge of six SM masses and parameters that must be specified via the `required_inputs` and `required_masses` attributes of the basis class. In our case, the electroweak inputs α^{-1}, G_F and m_Z are connected to the SMINPUTS block of the SLHA-like input structure, while the W-boson, Higgs boson and top quark masses are connected to its MASS block. The extra parameters C and three are dependent parameters that the user has to define in terms of the independent and SM parameters (see below). The non-SM part of an illustrative input file could be

```
BLOCK BASIS
 1 mybasis # input basis
BLOCK LETTERS
 1 8.6e-2 # A
 2 0.002  # B
```

```
BLOCK NUMBERS
 1 1.5e-2 # one
 2 2.8e-3 # two
BLOCK MYxMAT
 1 1 3.4e-2 # Rmat1x1
 1 2 7.8e-5 # Rmat1x2
 1 3 5.2e-4 # Rmat1x3
 2 2 5.6e-3 # Rmat2x2
 2 3 3.3e-3 # Rmat2x3
BLOCK IMMYxMAT
 1 2 9.9e-3 # Imat1x2
 1 3 1.9e-4 # Imat1x3
 2 3 4.6e-3 # Imat2x3
```

while its SM part would include the SMINPUTS and MASS blocks with values for the six above-mentioned SM inputs, as well as the two blocks related to the CKM matrix in the case where one would be interested in using non-default values for its elements. Only the relevant elements of MYxMAT need be provided given that it is declared to be Hermitian, and the (3,3) element is left unspecified as it is a dependent parameter.

The dependent parameters are evaluated via a method named `calculate_dependent()` that must be provided by the user. Continuing with the example above, we include in the new basis class implementation the code

```
def calculate_dependent(self):
  self['C']=(self['A']+self['B'])/2.
  self['three']=(self['one']-self['two'])/2.
  self['MYxMAT'][3,3]=10.*self['MYxMAT'][2,2]
```

This imposes that the C parameter is defined as the mean of the A and B parameters, that the three parameter equals half of the difference of the one and two parameters and that the (3,3) entry of the MYxMAT matrix is equal to 10 times the value of its (2,2) entry.

When executed, ROSETTA begins with the reading of the input file and next calls the `calculate_dependent()` method for evaluating the remaining basis parameters. RO-SETTA finally performs the translation to another basis by using the translation methods defined by the user. Their implementation requires the use of a `translation` decorator with an argument that refers to the name of the target basis and that must match a basis implementation contained in the `Rosetta` directory. For example, we could link the `mybasis` basis above to the Warsaw basis by implementing

```
@Basis.translation('warsaw')
def mytranslation(self, wbasis):
  a_EW = 1./self.inputs['aEWM1']
  wbasis['cWW'] = a_EW*self['C']
  wbasis['WBxHpl'][1,1] = self['two']
  return wbasis
```

Translation functions such as the `mytranslation(...)` one above take an instance of the target basis class as their only argument and return it after setting its parameter values. Relations involving (matrix) parameters with a flavor structure should be performed in a flavor general way, as discussed in Sect. 2.5.1.

If modifications to the SM input parameters need to be made (i.e., the `mass` and `inputs` attributes of the basis class), the function `modify_inputs()` must be implemented similarly to the `calculate_dependent()` method. The following example defines a shift of the W-boson mass by the `A` parameter,

```
def modify_inputs(self):
    self.mass[24] = self.mass[24] + self['A']
```

In general, the user-defined functions may require the evaluation of parameters such as the weak and hypercharge gauge couplings or the electroweak mixing angle. The choice of relations (e.g., tree- or loop-level) to be used to consistently derive these parameters from the inputs is left to the user. In the core bases provided with ROSETTA, the `calculate_inputs()` method relies on tree-level relations to deduce all the SM parameters.

Having defined a basis class according to these specifications, ROSETTA is able to detect the presence of the basis implementation and to automatically construct possible translation paths to other existing bases from the user-defined translation functions. The recognition of the implemented basis by ROSETTA is also reflected in the help message accompanying the `translate` script, the name of the new basis appearing as one of the possibilities for the target basis option.

2.5 Additional features

2.5.1 Flavor structure of the fermionic operators

In the general case, each matrix block of the input file includes one entry for each possible flavor assignment of the corresponding operator. The flavor option of the `translate` executable introduced in Sect. 2.1 allows the user to make assumptions on the flavor structure of the operators so that ROSETTA reads input files and generates output files with a simplified block structure (unless the BSMC basis is used as it requires all coefficients to be specified). Setting this option will act on *all* of the matrix parameters declared in the `flavored` attribute of a basis class implementation. The flavor option can be fixed either to `universal` where all matrices of operator coefficients are proportional to the identity or to `diagonal` where only their flavor-diagonal elements are retained. In the `universal` case one is allowed to define matrix blocks containing only the (1,1)

element while in the diagonal case, all three diagonal elements must be provided. Sample input files can be found in the `Cards` directory of the program. In the definitions of the three core bases, the flavor-symmetry-breaking Yukawa-like operators are normalized by the masses of the fermions such that the `universal` flavor option will lead to a minimally flavor-violating (MFV) structure where the physical effects of the coefficients are scaled by the corresponding fermion masses [30]. For example, in the Higgs basis, these Yukawa-like terms are written as:

$$\Delta \mathcal{L}_{\text{Yuk}} = \sqrt{m_f^i m_f^j} \delta y_f^{ij} \bar{f}^i \left(\cos \phi^{ij} - i \gamma_5 \sin \phi^{ij} \right) f^j. \quad (4)$$

The corresponding normalizations are also used for the Warsaw and SILH basis implementations in ROSETTA to simplify the translations and also the possibility of encoding MFV into any EFT description. The same argument applies to the dipole operators, O_{fV}, as well as the O_{Hud} operator mentioned in Sect. 2.2. The former set of operators breaks the flavor symmetry in an identical way to the Yukawa-like operators and will hence receive the same $\sqrt{m_f^i m_f^j}$ normalization. In the latter case, the flavor structure of the operator requires two Yukawa insertions as it is composed of right-handed quarks only. Moreover, being a charged-current operator, the MFV construction requires the insertion of the CKM matrix such that the operator is normalized as

$$\mathcal{O}_{Hud} = -i m_u^i m_d^j V_{CKM}^{ij} \left[\bar{u}^i \gamma^\mu d^j \right] \left[\tilde{\Phi}^\dagger D_\mu \Phi \right]. \quad (5)$$

Since this particular operator is unique and maps to a single operator in all of the other core bases, the corresponding translations remain unaffected. This normalization is however not the same as the one chosen in Ref. [16]. Users should therefore bear in mind these normalizations which have been chosen to single out operators that explicity break the flavor symmetry of the Lagrangian. That being said, they are merely a convenient way for the user to implement MFV and can be worked around if the user so desires.

ROSETTA recognizes coefficients by their names so that the naming of the elements of the matrix coefficients must respect the conventions described in the previous section for their real (an `R` prefix) and imaginary (an `I` prefix) parts, and for their position (i, j) inside the matrix (an `ixj` suffix). Implementing translations from flavored parameters should ideally always be done in the most general case such that the various flavor options work correctly. To this aim, basic matrix algebra operations have been implemeted in the `internal/matrices.py` module of ROSETTA. The available functions are `matrix_mult`, `matrix_add`, `matrix_sub` and `matrix_eq` and correspond to matrix multiplication, addition, subtraction and assignment respectively. They can be used to assign values to a matrix SLHA block according to the result of a

specific operation between two other matrix SLHA blocks. These functions require two mandatory arguments for the objects between which the operation should be performed and a third optional argument specifying the matrix block to which the result of the operation should be assigned. For instance, `matrix_mult(M1,M2,M3)` assigns to the matrix `M3` the result of the multiplication of the matrices `M1` and `M2`. If the `M3` argument is omitted, a generic matrix object is returned such that matrix utility functions can be combined. The `matrix_eq(M1,M2)` method is the only exception. It takes two mandatory arguments `M1` and `M2` and allows for the assignment of the elements of the `M1` matrix to the `M2` matrix.

A concrete example can be found in the `calculate_dependent()` function included in the Higgs basis implementation. The deviations of the W-boson couplings to the weak doublet of left-handed quark fields $\delta g_L^{W_q}$ are related to those of the Z-boson couplings to the individual left-handed up-type and down-type quarks $\delta g_L^{Z_u}$ and $\delta g_L^{Z_d}$ via the CKM matrix V_{CKM},

$$\delta g_L^{W_q} = \delta g_L^{Z_u} \cdot V_{\mathrm{CKM}} - V_{\mathrm{CKM}} \cdot \delta g_L^{Z_d} . \tag{6}$$

The ROSETTA implementation of this relation makes use of a combination of the `matrix_mult` and `matrix_sub` method,

```
matrix_sub(
  matrix_mult(HB['HBxdGLzu'], HB.ckm),
  matrix_mult(HB.ckm, HB['HBxdGLzd']),
  HB['HBxdGLwq']
)
```

where HB is an instance of the Higgs basis class. The third argument of the `matrix_sub` method allows one to assign the result of the matrix subtraction to the elements of the `HBxdGLwq` matrix block. Matrix blocks also come with the `T()` and `dag()` methods for transposing and Hermitian conjugation.

2.5.2 Interface to *e*HDECAY

In order to calculate dimension-six operator contributions to the Higgs boson width and branching ratios, ROSETTA includes an interface to the eHDECAY program [11]. It can be switched on by executing the `translate` script with the eHDECAY option (see Sect. 2.1). In order to use this feature, the path to a local installation of eHDECAY on the user system should be specified in the `Rosetta/config.txt` file, next to the `eHDECAY_dir` keyword, and a (possibly indirect) translation linking the basis of interest to the SILH basis should exist. If so, the translation will be performed, eHDECAY will be run internally and an SLHA decay

block for the Higgs boson will be appended to the output file.

Since the SILH basis description adopted in eHDECAY assumes the MFV paradigm, an additional layer of translation is internally performed by ROSETTA to render its internal SILH basis implementation MFV-compliant. Details can be found in Sect. 3.4.

3 Mapping different EFT basis choices

In this section, we discuss the BSMC Lagrangian containing redundant parameters that is the default basis which ROSETTA has been designed to translate into. We explain the relations with the non-redundant Higgs, Warsaw and SILH bases and focus on a subset of operators connected to single Higgs production at the LHC to provide examples of usage of ROSETTA.

3.1 The BSMC Lagrangian and the Higgs basis

To study the Higgs boson properties in detail at the next LHC runs, the LHCHXSWG has proposed a parameterization of anomalous interactions of the SM fermions, gauge bosons and the Higgs boson allowing both for a transparent linking to physical observables and for an easy implementation in Monte-Carlo event generators [16]. The framework is that of a general effective Lagrangian defined in the mass-eigenstate basis, where all kinetic terms are canonically normalized and diagonal, and where all mass terms are diagonal. Moreover, the tree-level relations between the gauge couplings and the usual electroweak input parameters (G_F, $\alpha(0)$, m_Z) are the same as in the SM. In such a frame, i.e., in the BSMC Lagrangian, the coefficients of the interaction terms in the Lagrangian are related in an intuitive way to quantities observable in experiment, and any parameter in the effective Lagrangian can be measured.

The BSMC Lagrangian captures all physics effects that may arise in the presence of lepton-number and baryon-number conserving dimension-six operators beyond the SM. However, it is more general than a basis defined before electroweak symmetry breaking as it contains more free parameters. This is because the $SU(3)_C \times SU(2)_L \times U(1)_Y$ gauge symmetry *linearly* realized at the level of an operator basis implies relations between different couplings of the effective Lagrangian defined after electroweak symmetry breaking. The latter indeed only respects the $SU(3)_C \times U(1)_{EM}$ symmetry. In particular, the charged and neutral gauge boson interactions are related, as are those with zero, one and two Higgs bosons. These relations are *not* implemented at the level of the BSMC Lagrangian so that it may be used to study more general theories such as when the electroweak

symmetry is non-linearly realized or when some operators of dimension greater than six are included.

The *Higgs* basis has been proposed as a convenient parameterization of another non-redundant dimension-six EFT basis. In this approach, the relations (that hold in any non-redundant dimension-six basis of EFT operators) between different couplings of the BSMC Lagrangian required by a linearly realized $SU(2)_L \times U(1)_Y$ local symmetry are enforced. Furthermore, the Higgs basis has been defined by choosing a specific subset of independent parameters from all couplings of the BSMC Lagrangian. The choice of the independent couplings is motivated by their direct connection to observables constrained by electroweak precision and Higgs studies. This approach is similar to the one introduced in Ref. [6], except that a different subset of couplings has been chosen, and the number of independent couplings is the same as for any basis of non-redundant dimension-six operators. Moreover, there exists a linear one-to-one invertible transformation between the independent couplings of the Higgs basis and the Wilson coefficients in any basis. The remaining BSMC Lagrangian couplings are all dependent parameters that can be expressed in terms of the independent ones.

The BSMC Lagrangian is displayed in Ref. [16], up to four-fermion terms and interactions involving five or more fields. Here, to illustrate the relationship between the BSMC and other bases, we focus on a part of the Lagrangian describing the CP-even interactions of the Higgs boson with two SM gauge bosons. After denoting by G_μ^a, $W_\mu^\pm$, Z_μ, A_μ and h the gluon, the W-boson, the Z-boson, the photon and the Higgs boson fields, respectively, the relevant part of the Lagrangian reads

$$
\begin{aligned}
\Delta\mathcal{L}_h = \frac{h}{v}\Big[& 2\delta c_w m_W^2 W_\mu^+ W^{-\mu} + \delta c_z m_Z^2 Z_\mu Z^\mu \\
& + c_{gg}\frac{g_s^2}{4} G_{\mu\nu}^a G_a^{\mu\nu} + c_{ww}\frac{g^2}{2} W_{\mu\nu}^+ W^{-\mu\nu} \\
& + c_{zz}\frac{g^2}{4c_\theta^2} Z_{\mu\nu}Z^{\mu\nu} + c_{z\gamma}\frac{gg'}{2} Z_{\mu\nu}A^{\mu\nu} \\
& + c_{\gamma\gamma}\frac{g'^2 c_\theta^2}{4} A_{\mu\nu}A^{\mu\nu} + c_{w\square}g^2\left(W_\mu^- \partial_\nu W^{+\mu\nu} + \text{h.c.}\right) \\
& + c_{z\square}g^2 Z_\mu \partial_\nu Z^{\mu\nu} + c_{\gamma\square}gg' Z_\mu \partial_\nu A^{\mu\nu}\Big].
\end{aligned}
\tag{7}
$$

In our notation, c_θ (s_θ) stands for the cosine (sine) of the electroweak mixing angle, v for the vacuum expectation value of the neutral component of the Higgs doublet Φ, and g_s, g and g' are the strong, weak and hypercharge coupling constants. Moreover, we have introduced the field strength tensors of the gauge bosons that we define as

$$
\begin{aligned}
V_{\mu\nu} &= \partial_\mu V_\nu - \partial_\nu V_\mu \quad \text{for} \quad V = W^\pm, Z \text{ and } A, \\
G_{\mu\nu}^a &= \partial_\mu G_\nu^a - \partial_\nu G_\mu^a + g_s f^a{}_{bc} G_\mu^b G_\nu^c,
\end{aligned}
\tag{8}
$$

in which $f^a{}_{bc}$ are the structure constants of $SU(3)_C$.

The Lagrangian above contains ten real coupling parameters that are all independent in the BSMC picture. However, if $\Delta\mathcal{L}_h$ originates from an EFT with dimension-six operators, only six of these couplings are independent and the remaining four can be expressed in terms of these six and of the SM parameters. In the Higgs basis, δc_z, c_{gg}, c_{zz}, $c_{z\gamma}$, $c_{\gamma\gamma}$ and $c_{z\square}$ are chosen as independent parameters and the four remaining couplings are calculated as

$$
\begin{aligned}
\delta c_w &= \delta c_z + 4\delta m, \\
c_{ww} &= c_{zz} + 2s_\theta^2 c_{z\gamma} + s_\theta^4 c_{\gamma\gamma}, \\
c_{w\square} &= \frac{g^2 c_{z\square} + g'^2 c_{zz} - (g^2-g'^2)s_\theta^2 c_{z\gamma} - g^2 s_\theta^4 c_{\gamma\gamma}}{g^2-g'^2}, \\
c_{\gamma\square} &= \frac{2g^2 c_{z\square} + (g^2+g'^2)c_{zz} - (g^2-g'^2)c_{z\gamma} - g^2 s_\theta^2 c_{\gamma\gamma}}{g^2-g'^2}.
\end{aligned}
\tag{9}
$$

In the first of these relations, δm denotes the shift of the W-boson mass that is possibly induced by the presence of higher-dimensional operators and that we normalize as

$$
\Delta\mathcal{L}_{\text{mass}} = 2\delta m \frac{g^2 v^2}{4} W_\mu^+ W^{-\mu}.
\tag{10}
$$

An input file (that we name `HiggsBasis.dat` in our example) describing this part of the Higgs basis Lagrangian would be of the form

```
BLOCK BASIS
   1  higgs # basis
BLOCK HBxh
   1   1.00000e-01 # dCz
   2   1.00000e-01 # Cgg
   3   1.00000e-01 # Caa
   4   1.00000e-01 # Cza
   5   1.00000e-01 # Czz
   6   1.00000e-01 # Czbx
```

It includes, in addition to the blocks above, the SM parameters as well as vanishing values for all other EFT coefficients. In order to export this setup to the BSMC Lagrangian, we use ROSETTA by typing in a shell

```
./translate HiggsBasis.dat
```

ROSETTA first calculates all dependent coefficients and next generates an output file named `HiggsBasis_new.dat` given in the framework of the BSMC Lagrangian. This file contains in particular values for the four δc_w, c_{ww}, $c_{w\square}$ and

$c_{\gamma\Box}$ dependent parameters, the corresponding output block being, according to Eq. (9),

```
BLOCK BCxh
   1  +1.00000e-01  # dCw
   1  +1.00000e-01  # dCw
   2  +1.00000e-01  # dCz
   3  +1.52190e-01  # Cww
   4  +1.00000e-01  # Cgg
   5  +1.00000e-01  # Caa
   6  +1.00000e-01  # Cza
   7  +1.00000e-01  # Czz
   8  +1.56506e-01  # Cwbx
   9  +1.00000e-01  # Czbx
  10  +3.41838e-01  # Cabx
```

In addition, the `HiggsBasis_new.dat` file also includes extra non-vanishing coefficients that are linked to the six independent parameters δc_z, c_{gg}, $c_{\gamma\gamma}$, $c_{z\gamma}$, c_{zz} and $c_{z\Box}$ by gauge invariance. For instance, a di-Higgs coupling to two gluonic field strength tensors is present,

```
BLOCK BCxhh
   4   1.00000e-01  # cgg2
```

3.2 The Warsaw basis

The ten interaction terms of the $\Delta\mathcal{L}_h$ Lagrangian introduced in Sect. 3.1 can be seen as generated by six independent operators of the Warsaw basis,

$$
\begin{aligned}
&\Delta\mathcal{L}_h^{\mathrm{W}}\\
&= \frac{1}{v^2}\Big[c_{GG}\frac{g_s^2}{4}\,\Phi^\dagger\Phi\,G_{\mu\nu}^a G_a^{\mu\nu} + c_{WW}\frac{g^2}{4}\,\Phi^\dagger\Phi\,W_{\mu\nu}^i W_i^{\mu\nu}\\
&\quad + c_{WB}gg'\,\Phi^\dagger\sigma_i\Phi\,W_{\mu\nu}^i B^{\mu\nu} + c_{BB}\frac{g'^2}{4}\,\Phi^\dagger\Phi\,B_{\mu\nu}B^{\mu\nu}\\
&\quad + c_H\,\partial_\mu\big[\Phi^\dagger\Phi\big]\partial^\mu\big[\Phi^\dagger\Phi\big] + c_T\big[\Phi^\dagger\overleftrightarrow{D}_\mu\Phi\big]\big[\Phi^\dagger\overleftrightarrow{D}^{\,\mu}\Phi\big]\Big].
\end{aligned}
$$
(11)

In this expression, we have introduced the Pauli matrices σ_i, the Hermitian derivative operator,

$$
\Phi^\dagger\overleftrightarrow{D}_\mu\Phi = \Phi^\dagger(D_\mu\Phi) - (D_\mu\Phi^\dagger)\Phi,
$$
(12)

the gauge-covariant derivative and the hypercharge and weak field strength tensors

$$
\begin{aligned}
D_\mu\Phi &= \Big(\partial_\mu - \frac{i}{2}g\sigma_k W_\mu^k - \frac{i}{2}g'B_\mu\Big)\Phi,\\
W_{\mu\nu}^i &= \partial_\mu W_\nu^i - \partial_\nu W_\mu^i + g\epsilon^i{}_{jk}W_\mu^j W_\nu^k,\\
B_{\mu\nu} &= \partial_\mu B_\nu - \partial_\nu B_\mu.
\end{aligned}
$$
(13)

The six Wilson coefficients c_{GG}, c_{WW}, c_{WB}, c_{BB}, c_H and c_T appearing in $\Delta\mathcal{L}_h^{\mathrm{W}}$ are related to the ten couplings in the effective Lagrangian $\Delta\mathcal{L}_h$ as

$$
\begin{aligned}
\delta c_w &= -c_H - \frac{4g^2 g'^2}{g^2-g'^2}c_{WB} + \frac{4g^2}{g^2-g'^2}c_T - \frac{3g^2+g'^2}{g^2-g'^2}\delta v,\\
\delta c_z &= -c_H - 3\delta v,\\
c_{gg} &= c_{GG},\\
c_{ww} &= c_{WW},\\
c_{zz} &= \frac{g^4 c_{WW} + 4g^2 g'^2 c_{WB} + g'^4 c_{BB}}{(g^2+g'^2)^2},\\
c_{z\gamma} &= \frac{g^2 c_{WW} - 2(g^2-g'^2)c_{WB} - g'^2 c_{BB}}{g^2+g'^2},\\
c_{\gamma\gamma} &= c_{WW} + c_{BB} - 4c_{WB},\\
c_{w\Box} &= \frac{2}{g^2-g'^2}\big[g'^2 c_{WB} - c_T + \delta v\big],\\
c_{z\Box} &= -\frac{2}{g^2}\big[c_T - \delta v\big],\\
c_{\gamma\Box} &= \frac{2}{g^2-g'^2}\big[(g^2+g'^2)c_{WB} - 2c_T + 2\delta v\big].
\end{aligned}
$$
(14)

Here, δv is defined by

$$
\delta v = \frac{1}{2}\Big[(c'_{H\ell})_{11} + (c'_{H\ell})_{22}\Big] - \frac{1}{4}(c_{\ell\ell})_{1221}
$$
(15)

and summarizes the dependence on the additional Warsaw basis operators,[3]

$$
\frac{i}{v^2}c'_{H\ell}\big[\bar{\ell}\sigma_i\gamma_\mu\ell\big]\big[\Phi^\dagger\sigma^i\overleftrightarrow{D}^\mu\Phi\big] + \frac{1}{v^2}c_{\ell\ell}\big[\bar{\ell}\gamma_\mu\ell\big]\big[\bar{\ell}\gamma^\mu\ell\big].
$$
(16)

Starting from the Higgs basis example of Sect. 3.1 where all independent parameters are fixed to 0.1, we employ Ro-SETTA to invert the relations of this section and calculate the numerical values of the Warsaw basis coefficients included in $\Delta\mathcal{L}_h^{\mathrm{W}}$ that would yield the same $\Delta\mathcal{L}_h$ Lagrangian. Typing in a shell

```
./translate HiggsBasis.dat -t warsaw
```

we obtain an output file where several non-zero EFT coefficients can be found. The numerical value of those on which we focus here can be extracted from the generated file,

```
BLOCK BASIS
   1   warsaw # translated basis
BLOCK WBxH4D2 #
   1    -1.98704e-01 # cH
```

---

[3] These operators contribute to the muon decay at the tree level. Taking this into account leads to a shift between the measured Fermi constant and the vacuum expectation value of the Higgs field, which motivates the notation δv.

```
    2     +1.18790e-02 # cT
BLOCK WBxV2H2 #
    1     +1.00000e-01 # cGG
    2     +1.52190e-01 # cWW
    3     +5.45943e-03 # cBB
    4     +1.44124e-02 # cWB
```

In our benchmark scenario, the δv shift is vanishing.

3.3 SILH basis

We now consider the case where all operators included in the $\Delta\mathcal{L}_h$ Lagrangian of Sect. 3.1 are induced by a set of operators of the SILH basis,

$$
\Delta\mathcal{L}_h^{\rm S} = \frac{1}{v^2}\Big[s_{GG}\frac{g_s^2}{4}\,\Phi^\dagger\Phi\, G_{\mu\nu}^a G_a^{\mu\nu} + s_{BB}\frac{g'^2}{4}\,\Phi^\dagger\Phi\, B_{\mu\nu}B^{\mu\nu}
$$
$$
+ s_W\frac{ig}{2}\big[\Phi^\dagger\sigma^i\overleftrightarrow{D}_\mu\Phi\big]\partial_\nu W_i^{\mu\nu} + s_B\frac{ig'}{2}\big[\Phi^\dagger\overleftrightarrow{D}_\mu\Phi\big]\partial_\nu B^{\mu\nu}
$$
$$
+ i\,s_{HW}\,g\,\big[D_\mu\Phi^\dagger\sigma^i D_\nu\Phi\big]W_i^{\mu\nu} + i\,s_{HB}\,g'\,\big[D_\mu\Phi^\dagger D_\nu\Phi\big]B^{\mu\nu}
$$
$$
+ s_{2W}\,D^\mu W_{\mu\nu}^i D_\rho W_i^{\rho\nu} + s_{2B}\,\partial^\mu B_{\mu\nu}\partial_\rho B^{\rho\nu}
$$
$$
+ s_H\,\partial_\mu\big[\Phi^\dagger\Phi\big]\partial^\mu\big[\Phi^\dagger\Phi\big] + s_T\big[\Phi^\dagger\overleftrightarrow{D}_\mu\Phi\big]\big[\Phi^\dagger\overleftrightarrow{D}^\mu\Phi\big]
$$
$$
+ i\,s'_{H\ell}\big[\bar\ell\sigma_i\gamma_\mu\ell\big]\big[\Phi^\dagger\sigma^i\overleftrightarrow{D}^\mu\Phi\big]\Big]. \tag{17}
$$

The ten Wilson coefficients included in $\Delta\mathcal{L}_h$ can be rewritten in terms of the eleven parameters appearing in $\Delta\mathcal{L}_h^{\rm S}$ as

$$
\delta c_w = -s_H - \frac{g^2 g'^2(s_W+s_B+s_{2W}+s_{2B})}{g^2-g'^2} - \frac{4g^2}{g^2-g'^2}s_T
$$
$$
\qquad + \frac{3g^2+g'^2}{g^2-g'^2}\delta v,
$$
$$
\delta c_z = -s_H - 3\delta v,
$$
$$
c_{gg} = s_{GG},
$$
$$
c_{ww} = -s_{WW},
$$
$$
c_{zz} = -\frac{g^2 s_{HW} + g'^2 s_{HB} - g'^2 s_\theta^2 s_{BB}}{g^2+g'^2}, \tag{18}
$$
$$
c_{z\gamma} = -\frac{s_{HW}-s_{HB}}{2} - s_\theta^2 s_{BB},
$$
$$
c_{\gamma\gamma} = s_{BB},
$$
$$
c_{w\Box} = \frac{1}{2}s_{HW} + \frac{g^2(s_W+s_{2W}) + g'^2(s_B+s_{2B}) - 4s_T + 4\delta v}{2(g^2-g'^2)},
$$
$$
c_{z\Box} = \frac{g^2(s_W+s_{2W}+s_{HW}) + g'^2(s_B+s_{2B}+s_{HB}) - 4s_T + 4\delta v}{2g^2},
$$
$$
c_{\gamma\Box} = \frac{s_{HW}-s_{HB}}{2} + \frac{g^2(s_W+s_{2W}) + g'^2(s_B+s_{2B}) - 4s_T + 4\delta v}{g^2-g'^2},
$$

where $\delta v = \frac{1}{2}(s'_{H\ell})_{22}$. ROSETTA can be used to extract the numerical values of the independent SILH parameters by inverting the above relations. Adopting the benchmark scenario of Sect. 3.1 where all the relevant Higgs basis independent parameters have been fixed to 0.1, we type in a shell

```
./translate HiggsBasis.dat -t silh
```

so that we can extract all the required SILH coefficients from the generated output file,

```
BLOCK BASIS
    1    silh # translated basis
BLOCK SBxH4D2 #
    1     -1.00000e-01 # sH
    2     +0.00000e+00 # sT
BLOCK SBxV2H2 #
    1     +1.00000e-01 # sGG
    2     +1.00000e-01 # sBB
    3     +4.65203e-01 # sW
    4     -4.65203e-01 # sB
    5     -1.52190e-01 # sHW
    6     +9.45406e-02 # sHB
    7     +0.00000e+00 # s2W
    8     +0.00000e+00 # s2B
```

3.4 Yukawa-like operators

An important difference between the definitions of the SILH and Warsaw bases provided in the LHCHXSWG proposal [16] and their original descriptions lies in the forms of the Yukawa-like operators,

$$
\Delta\mathcal{L}_{\rm Yuk} = \frac{(c_f)_{ij}}{v^2}\left(\Phi^\dagger\Phi\right)\bar F_L^i\Phi f_R^j, \tag{19}
$$

where F_L and f_R denote a generic weak doublet of left-handed fermions and a generic weak singlet of right-handed fermions respectively. In the original Warsaw basis definition, these Yukawa-like operators take the above form. In the LHCHXSWG proposal (on which ROSETTA is based), these operators have been redefined in a way allowing one to decouple their contributions to the fermion masses (that are extracted from appropriate measurements and thus fixed), as well as to simplify the implementation of MFV,

$$
\Delta\mathcal{L}'_{\rm Yuk} = -\frac{\sqrt{m_i m_j}}{v}\frac{(c'_f)_{ij}}{v^2}\left(\Phi^\dagger\Phi - \frac{v^2}{2}\right)\bar F_L'^i\Phi f_R'^j, \tag{20}
$$

where the primes denote fields taken in the mass eigenbasis. The Wilson coefficients c_f and c'_f are related by unitary transformations U_L and U_R that map the field gauge eigenbasis to the mass eigenbasis with the would-be mass modifications absorbed into the diagonalized Yukawa matrices Y_f^D,

$$
c'_f = \frac{v}{\sqrt{m_i m_j}}U_L^\dagger c_f U_R \quad\text{and}\quad Y_f^D = U_L^\dagger Y_f U_R + \frac{c_f}{2}. \tag{21}
$$

In the original SILH basis description, an additional assumption of minimal flavor violation is included, such that the flavor structure is taken aligned with the Yukawa matrices,

$$\Delta \mathcal{L}_{\text{Yuk}}^{\text{MFV}} = (Y_f)_{ij} \frac{c_f^{\text{MFV}}}{v^2} \left(\Phi^\dagger \Phi \right) \bar{F}_L^i \Phi f_R^j. \qquad (22)$$

The Wilson coefficients c_f^{MFV} are therefore proportional to the identity matrix in flavor space and are thus universal. Thanks to the convenient normalizations, they are now trivially related linearly to those of the Warsaw and SILH basis descriptions of the LHCHXSWG proposal by

$$(c_f)_{ii} = \frac{v}{m_i} Y_f^D c_f^{\text{MFV}} = \sqrt{2} c_f^{\text{MFV}}. \qquad (23)$$

These relations are used internally for the eHDECAY interface of ROSETTA, which takes SILH basis input parameters assuming the MFV convention of Eq. (22). In order to consistently use eHDECAY, ROSETTA translates these coefficients from the alternative version of the SILH basis detailed in Ref. [16]. As a consequence, a general flavor structure cannot be employed when making use of the eHDECAY interface. Although it is in principle possible to input different values for the c_c^{MFV}, c_b^{MFV}, c_t^{MFV}, c_μ^{MFV} and c_τ^{MFV} parameters (referred in Ref. [11] as $\bar{c}_c$, etc.) when running eHDECAY on its own, large deviations from non-universality in these coefficients consist of a significant departure from the MFV paradigm and should not be used for complete consistency within ROSETTA.

4 Summary

In this paper, we have introduced the ROSETTA package, a PYTHON program dedicated to the translation of a given EFT basis of independent operators to other viable basis choices. We have also included, in this document, technical details so that users can easily extend the program and implement their own choices of EFT operator basis.

Currently, the program allows the user to translate benchmarks designed in the Higgs, SILH and Warsaw bases into any of these three bases. In addition, the code also expresses any scenario in terms of the BSMC Lagrangian of EFT operators, a basis that has been defined from the Higgs basis after ignoring all relations among the operators that are induced by a linear realization of the electroweak symmetry. A FEYN-RULES implementation allows ROSETTA to be linked to other high-energy physics tools. The relations among the different Wilson coefficients that hold in the context of the Higgs, SILH and Warsaw bases of independent operators have been implemented into ROSETTA so that they are preserved when a setup is exported to the BSMC Lagrangian by the program. This scheme has the strength to be easily generalizable to study different setups providing a description of the Higgs boson properties, such as those with a non-linearly realized electroweak symmetry or including higher-dimensional operators beyond dimension six.

In the future, we believe that translations from one basis to another will allow for broadening the scope and the use of past calculations very relevant for precision Higgs physics. Along these lines, higher-order calculations in QCD performed in the BSMC Lagrangian [31–33] could be used within any given EFT language, and the renormalization group running of the Wilson coefficients, that has been calculated in the SILH basis [34,35] and in the Warsaw bases [3,21,22], could be exported to different bases too.

References

1. W. Buchmuller, D. Wyler, Effective Lagrangian analysis of new interactions and flavor conservation. Nucl. Phys. B **268**, 621 (1986)
2. B. Grzadkowski, M. Iskrzynski, M. Misiak, J. Rosiek, Dimension-six terms in the standard model Lagrangian. JHEP **1010**, 085 (2010). arXiv:1008.4884
3. R. Alonso, E.E. Jenkins, A.V. Manohar, M. Trott, Renormalization group evolution of the standard model dimension six operators III: gauge coupling dependence and phenomenology. JHEP **1404**, 159 (2014). arXiv:1312.2014
4. A. Pomarol, F. Riva, Towards the ultimate SM fit to close in on higgs physics. JHEP **1401**, 151 (2014). arXiv:1308.2803
5. J. Ellis, V. Sanz, T. You, Complete higgs sector constraints on dimension-6 operators. JHEP **1407**, 036 (2014). arXiv:1404.3667
6. R.S. Gupta, A. Pomarol, F. Riva, BSM primary effects, Phys. Rev. **D91**(3), 035001 (2015). arXiv:1405.0181
7. J. Ellis, V. Sanz, T. You, The effective standard model after LHC run I. JHEP **1503**, 157 (2015). arXiv:1410.7703
8. A. Falkowski, F. Riva, Model-independent precision constraints on dimension-6 operators. JHEP **1502**, 039 (2015). arXiv:1411.0669
9. A. Efrati, A. Falkowski, Y. Soreq, Electroweak constraints on flavorful effective theories. JHEP **07**, 018 (2015). arXiv:1503.07872
10. G. Giudice, C. Grojean, A. Pomarol, R. Rattazzi, The strongly-interacting light higgs. JHEP **0706**, 045 (2007). arXiv:hep-ph/0703164
11. R. Contino, M. Ghezzi, C. Grojean, M. Muhlleitner, M. Spira, Effective Lagrangian for a light Higgs-like scalar. JHEP **1307**, 035 (2013). arXiv:1303.3876
12. E. Masso, An effective guide to beyond the standard model physics. JHEP **1410**, 128 (2014). arXiv:1406.6376

13. A. Pomarol, Higgs physics, in *2014 European School of High-Energy Physics (ESHEP 2014) Garderen, The Netherlands, June 18–July 1, 2014*, 2014. arXiv:1412.4410

14. C. Grojean, W. Skiba, J. Terning, Disguising the oblique parameters. Phys. Rev. D **73**, 075008 (2006). arXiv:hep-ph/0602154

15. Z. Han, W. Skiba, Effective theory analysis of precision electroweak data. Phys. Rev. D **71**, 075009 (2005). arXiv:hep-ph/0412166

16. M. Duehrssen-Debling, A. T. Mendes, A. Falkowski, G. Isidori, Higgs basis: proposal for an EFT basis choice for LHC HXSWG

17. P. Artoisenet, P. de Aquino, F. Demartin, R. Frederix, S. Frixione et al., A framework for Higgs characterisation. JHEP **1311**, 043 (2013). arXiv:1306.6464

18. A. Alloul, N.D. Christensen, C. Degrande, C. Duhr, B. Fuks, FeynRules 2.0—a complete toolbox for tree-level phenomenology. Comput. Phys. Commun. **185**, 2250–2300 (2014). arXiv:1310.1921

19. A. Alloul, B. Fuks, V. Sanz, Phenomenology of the Higgs effective Lagrangian via FEYNRULES. JHEP **1404**, 110 (2014). arXiv:1310.5150

20. C. Degrande, B. Fuks, K. Mawatari, K. Mimasu, V. Sanz, in preparation

21. E.E. Jenkins, A.V. Manohar, M. Trott, Renormalization group evolution of the standard model dimension six operators I: formalism and lambda dependence. JHEP **10**, 087 (2013). arXiv:1308.2627

22. E.E. Jenkins, A.V. Manohar, M. Trott, Renormalization group evolution of the standard model dimension six operators II: Yukawa dependence. JHEP **01**, 035 (2014). arXiv:1310.4838

23. B. Henning, X. Lu, H. Murayama, How to use the standard model effective field theory. arXiv:1412.1837

24. http://feynrules.irmp.ucl.ac.be/wiki/BSMCharacterisation

25. N.D. Christensen, P. de Aquino, C. Degrande, C. Duhr, B. Fuks, M. Herquet, F. Maltoni, S. Schumann, A comprehensive approach to new physics simulations. Eur. Phys. J. C **71**, 1541 (2011). arXiv:0906.2474

26. C. Degrande, C. Duhr, B. Fuks, D. Grellscheid, O. Mattelaer et al., UFO—the universal feynrules output. Comput. Phys. Commun. **183**, 1201–1214 (2012). arXiv:1108.2040

27. P.Z. Skands, B. Allanach, H. Baer, C. Balazs, G. Belanger et al., SUSY les houches accord: interfacing SUSY spectrum calculators, decay packages, and event generators. JHEP **0407**, 036 (2004). arXiv:hep-ph/0311123

28. B.C. Allanach et al., SUSY les houches accord 2. Comput. Phys. Commun. **180**, 8–25 (2009). arXiv:0801.0045

29. Particle Data Group Collaboration, K. Olive et al., Review of particle physics. Chin. Phys. **C38**, 090001 (2014)

30. G. D'Ambrosio, G.F. Giudice, G. Isidori, A. Strumia, Minimal flavor violation: an effective field theory approach. Nucl. Phys. B **645**, 155–187 (2002). arXiv:hep-ph/0207036

31. F. Maltoni, K. Mawatari, M. Zaro, Higgs characterisation via vector-boson fusion and associated production: NLO and parton-shower effects, Eur. Phys. J. **C74**(1), 2710 (2014). arXiv:1311.1829

32. F. Demartin, F. Maltoni, K. Mawatari, B. Page, M. Zaro, Higgs characterisation at NLO in QCD: CP properties of the top-quark Yukawa interaction, Eur. Phys. J. **C74**(9), 3065 (2014). arXiv:1407.5089

33. F. Demartin, F. Maltoni, K. Mawatari, M. Zaro, Higgs production in association with a single top quark at the LHC, Eur. Phys. J. **C75**(6) (2015). arXiv:1504.00611

34. J. Elias-Mir, J.R. Espinosa, E. Masso, A. Pomarol, Renormalization of dimension-six operators relevant for the Higgs decays $h \to \gamma\gamma$, γZ. JHEP **08**, 033 (2013). arXiv:1302.5661

35. J. Elias-Miro, J.R. Espinosa, E. Masso, A. Pomarol, Higgs windows to new physics through d=6 operators: constraints and one-loop anomalous dimensions. JHEP **11**, 066 (2013). arXiv:1308.1879