

## HyperIso: A general BSM calculator for flavour observables

---

**Théo Reymermier,<sup>a,\*</sup> Niels Fardeau<sup>a</sup> and Farvah Mahmoudi<sup>a,b,c</sup>**

<sup>a</sup>*Université de Lyon, Université Claude Bernard Lyon 1, CNRS/IN2P3, Institut de Physique des 2 Infinis de Lyon, UMR 5822, F-69622, Villeurbanne, France*

<sup>b</sup>*Theoretical Physics Department, CERN, CH-1211 Geneva 23, Switzerland*

<sup>c</sup>*Institut Universitaire de France (IUF), 75005 Paris, France*

*E-mail: [theo.reymermier@cern.ch](mailto:theo.reymermier@cern.ch), [n.fardeau@ip2i.in2p3.fr](mailto:n.fardeau@ip2i.in2p3.fr), [mahmoudi@in2p3.fr](mailto:mahmoudi@in2p3.fr)*

HYPERISO is a refactored and expanded version of the flavour code SUPERISO allowing for efficient calculations of flavour observables. While SUPERISO was dedicated to Standard Model (SM), Two Higgs Doublet Model (THDM) and several Supersymmetry (SUSY) models, HYPERISO now implements a transparent interface with MARTY (a public tool to perform analytical QFT calculations) to extend SUPERISO's observable calculation routines to generic Beyond the Standard Model (BSM) scenarii. HYPERISO also implements a Python binding and Graphical User Interface (GUI) to improve user accessibility and ease of use.

*The European Physical Society Conference on High Energy Physics (EPS-HEP2025)  
7-11 July 2025  
Marseille, France*

---

\*Speaker

## 1. Introduction

In the search for new physics BSM, flavour observables play a central role in constraining possible extensions of the SM. Processes such as B- and K-meson decays serve as sensitive probes of potential new-physics effects. As experimental precision continues to sharpen at LHCb and Belle II, there is a growing demand for theory tools that can accommodate intricate scenarios, support quantitative inference and deliver robust predictions of the various BSM models.

Existing packages, like SUPERISO [1–4], have enabled substantial progress in this program. However, their coverage is typically tied to specific models, such as THDM or SUSY-like models, which makes extensions to novel theories or alternative effective descriptions cumbersome. In addition, many of the available interfaces are still relatively rigid, limiting usability across the different communities from phenomenologists to experimental analysts.

We introduce HYPERISO, a next-generation framework that succeeds and extends SUPERISO to evaluate flavour observables across arbitrary theoretical models, thanks to the MARTY framework [5, 6], while performing statistical analyses (e.g  $\chi^2$  evaluations). HYPERISO leverages the MARTY framework to compute (and match to the UV theory) Wilson coefficients automatically, allowing users to use any new BSM models, with the only requirement that new particles be above  $\sim 100$  GeV. A high-performance C++ core conducts the computations and is complemented by two interfaces (C++ and a Python format). A GUI is also available to broaden accessibility for users without much knowledge in coding.

HYPERISO accepts Les Houches Accord (LHA) [7] as inputs and integrates smoothly with existing high-energy physics tools, facilitating spectrum generation, parameters's scans and global fits.

We will describe in this write-up the software's design and implementation, the comparison between HYPERISO and SUPERISO on classical models such as THDM, and show the capabilities of HYPERISO on new models, without hardcoded Wilson coefficients.

## 2. Theory and Software Architecture

### 2.1 Theoretical setup

In order to separate the long-distance physics from the short-distance physics (containing the BSM dependence), we work within the Weak Effective Theory (WET), in which the heavy degrees of freedom, like the top quark,  $W^\pm$ , Z and H bosons, are integrated out. The new particles from the BSM scenarios are also integrated out. The resulting low-energy dynamics are encoded in an operator basis with Wilson coefficients  $C_i(\mu)$  evolved to the relevant scale for each observable.

For  $b \rightarrow s$  transitions, the effective Hamiltonian takes the standard form :

$$\mathcal{H}_{eff}^{b \rightarrow s} = -\frac{4G_F}{\sqrt{2}} V_{tb} V_{ts}^* \sum_{i=1}^{10} C_i(\mu) O_i(\mu) + \text{h.c.} \quad (1)$$

Perturbative QCD corrections are organized as :

$$C_i(\mu) = \sum_{j=0}^{\infty} \left( \frac{\alpha_s(\mu)}{4\pi} \right)^{(j)} C_i^j(\mu) \quad (2)$$

The running of the Wilson coefficients is done either in the standard operator basis or the traditional operator basis (both are available in HyperIso).

The current release of HYPERISO supports an important set of flavour physics observables, the main decays (in which all the observables are calculated) are presented in Tab.1.

$$\begin{array}{c|c|c|c|c}
 B \rightarrow X_{s,d}\gamma & B \rightarrow X_s\ell\ell & B \rightarrow K^{(*)}\ell\ell & B \rightarrow K^*\gamma & B \rightarrow D^{(*)}\ell\nu \\
 B_s \rightarrow \phi\ell\ell & B_{s,d} \rightarrow \ell\ell & B_u \rightarrow \ell\nu & & \\
 K \rightarrow \pi\nu\bar{\nu} & K_{L,S} \rightarrow \mu\mu & K \rightarrow \pi ll & K \rightarrow l\nu & \\
 M_0 - \bar{M}_0 & D_{s,d} \rightarrow \ell\nu & \Lambda_b \rightarrow \Lambda ll & & 
 \end{array}$$

**Table 1:** List of decays available within HYPERISO.

All relevant B decays are available, but also K and D decays are provided. Other observables like the anomalous magnetic moment (g-2) can also be computed within HYPERISO.

## 2.2 Architecture, dataflow and interfaces

HYPERISO is the direct evolution of SUPERISO. It retains a C++ core while adopting a hexagonal (port-and-adapters) architecture to decouple physics logic from input/output from each part (Wilson coefficients calculation, Observables calculation, LHA read/write, etc.). This design reduces coupling between components and makes it straightforward to swap implementation details, for instance, exchanging the Wilson coefficients provider (hard-coded ones, MARTY inputs or user inputs), or adding new observables/models without refactoring the rest of the stack. The architecture of HYPERISO is presented in Fig.1.

The Core of HYPERISO can be divided in four principal layers. The **PhysicalModel** layer, in which the short-distance elements are calculated (Wilson coefficient Matching, RGE). The **BusinessLogic** layer where the long-distance contributions (observables) are evaluated. The **Data/IO** layer which deals with configuration, inputs and result serialisation and finally the **Application** layer, where the Command Line Interface (CLI), API and GUI orchestrates the end-to-end workflows.

Default inputs (SM/QCD constants, CKM, hadronic parameters) are provided via JSON and trace back to PDG values [8]. YAML files can override any entry to test alternative conventions or recent measurements (user responsibility for any result given by non-default values). the LHA files supply model-specific parameters, which by design override JSON defaults when the same quantity is specified. This layered precedence makes reproducibility and controlled studies straightforward.

The interfacing with MARTY is done using a special interface, developed inside HYPERISO which can create and run C++ scripts from templates, which extract the Wilson coefficients as CSV, and can then be used inside HYPERISO for the RGE and observable calculations.

The frontends exist in multiple formats, such as a classical C++ format, a CLI (for simple calculations), also a pybind11-based [9] Python bindings allow the user to directly drive HYPERISO from a Python environment. A GUI uses the Python binding to create a "no-code" version of HYPERISO, with an interactive window.

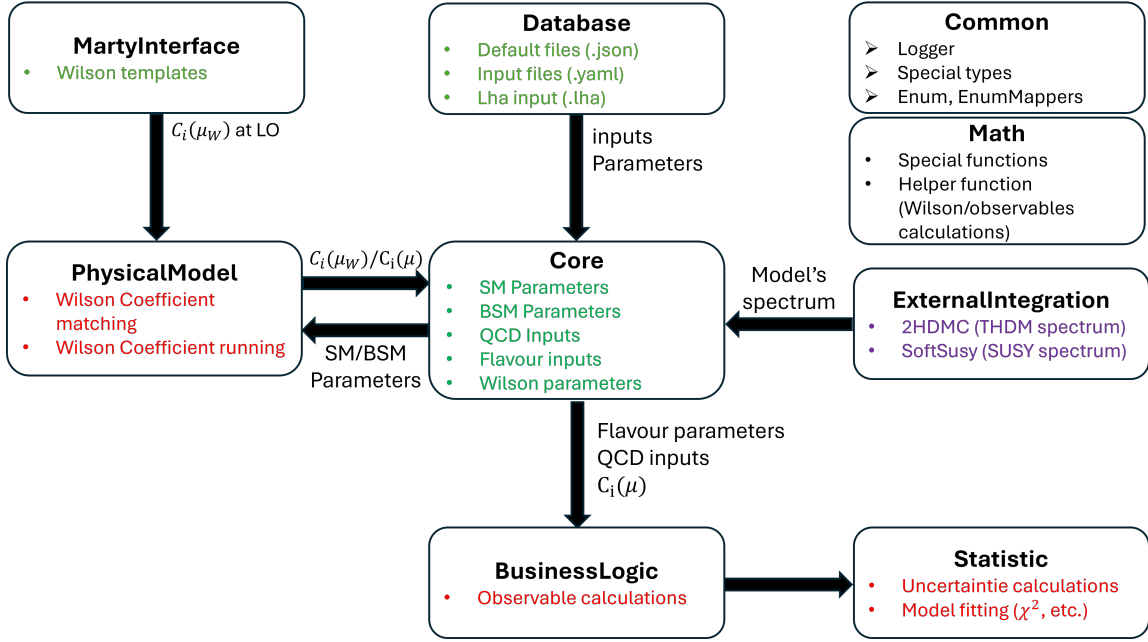


Figure 1: Architecture of HyperIso.

### 3. Results

We report two complementary validations. A legacy comparison against SUPERISO for implemented models (SM, THDM, SUSY) and an internal cross-check contrasting HYPERISO's hard-coded path (legacy of SUPERISO) with the MARTY-driven path on representative scenarios. All runs use identical input parameters and renormalization conventions.

#### 3.1 Legacy comparison with SuperIso

For models like SM, THDM and SUSY, we reproduced the hard-coded methods of SUPERISO, which perform calculations up to NNLO. Even for new models where HYPERISO needs MARTY to run, one might use the SM contributions up to NNLO within HyperIso's core for more accurate results. Results of the comparison with SUPERISO for the matching scale Wilson coefficients calculations are summarized in Tab.2. THDM was tested in type 2, and the MSSM as an example for SUSY. We used the same LHA input for both SUPERISO and HYPERISO. Other type of SUSY models were also tested to validate HYPERISO's calculations. We also tested the running (up to NNLO) of the coefficients which shows 0% difference, as it contains few parameters and is model-independent.

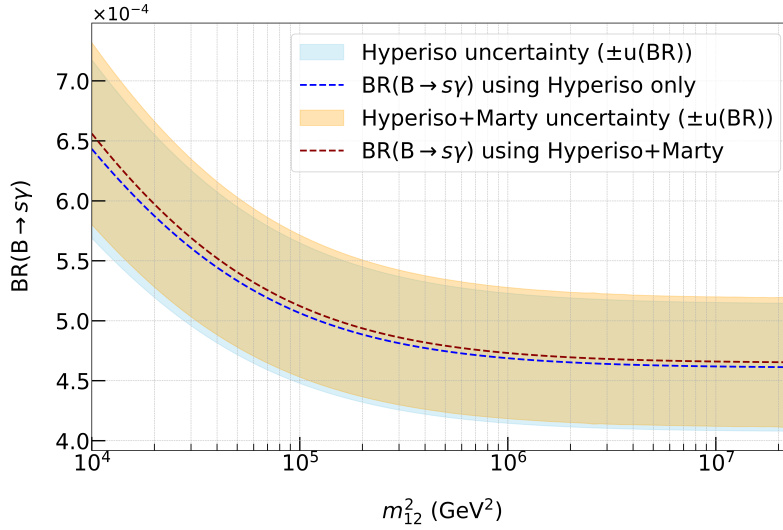
#### 3.2 HYPERISO hard-coded vs MARTY-driven path

After making sure the calculations within HYPERISO were correct, we also tested MARTY's result compared to the hard-coded version. Fig.2 presents an example for the  $b \rightarrow s\gamma$  calculation while varying the  $m_{12}^2$  parameters, using both MARTY for the Wilson coefficients calculations or the hard-coded versions. We are able to reproduce all SuperIso's results using HYPERISO with

coefs	SM			THDM			SUSY		
	SI	HI	relative error %	SI	HI	relative error %	SI	HI	relative error %
$\mu_W = O(M_W)$									
$C_1$	1.58e-1	1.58e-1	0	0	0	0	-5.59e-7	-5.59e-7	0
$C_2$	1.00	1.00	0	0	0	0	0	0	0
$C_3$	-2.93e-4	-2.93e-4	0	-3.72e-8	-3.72e-8	0	4.93e-7	4.93e-7	1.83e-4
$C_4$	-3.25e-3	-3.25e-3	4.25e-2	9.30e-6	9.30e-6	7.50e-5	4.00e-6	4.00e-6	2.24e-4
$C_5$	3.39e-5	3.39e-5	1.20e-2	1.40e-8	1.40e-8	1.95e-4	-4.44e-8	-4.44e-8	2.03e-4
$C_6$	6.35e-5	6.35e-5	0	2.63e-8	2.63e-8	0	-8.33e-8	-8.33e-8	1.08e-4
$C_7$	-2.11e-1	-2.11e-1	3.10e-2	8.01e-4	8.00e-4	3.86e-2	8.46e-4	8.45e-4	4.75e-2
$C_8$	-1.15e-1	-1.15e-1	2.19e-2	8.70e-4	8.70e-4	2.40e-5	-1.01e-2	-1.01e-2	6.71e-4
$C_9$	1.93	1.93	3.50e-2	-2.78e-4	-2.78e-4	3.09e-4	5.88e-3	5.88e-3	2.62e-2
$C_{10}$	-4.17	-4.17	7.36e-2	-4.89e-3	-4.89e-3	1.60e-5	5.72e-3	5.72e-3	3.03e-4
$CQ_1$	-2.60e-4	-2.60e-4	1.54e-3	-3.83e-4	-3.83e-4	7.84e-4	-4.30e-4	-4.30e-4	7.81e-3
$CQ_2$	4.86e-4	4.85e-4	4.13e-4	6.37e-6	6.37e-6	2.51e-2	1.91e-4	1.91e-4	1.70e-2
$CP_7$	-4.24e-3	-4.24e-3	0	0	0	0	2.19e-4	2.19e-4	0
$CP_8$	-2.22e-3	-2.22e-3	0	0	0	0	1.04e-5	1.04e-5	2.88e-3
$CP_9$	0	0	0	0	0	0	-2.97e-6	-2.97e-6	1.55e-2
$CP_{10}$	0	0	0	0	0	0	7.00e-5	7.00e-5	0
$CPQ_1$	0	0	0	0	0	0	-6.93e-6	-6.93e-6	5.77e-4
$CPQ_2$	0	0	0	0	0	0	-6.93e-6	-6.93e-6	1.44e-4

**Table 2:** Comparaison SUPERIso (SI) vs HYPERIso (HI) using relative error (%) for the Wilson coefficients calculations at matching scale  $O(M_W)$ .

MARTY. Differences arise from approximations made by HYPERIso (and SUPERIso) which are not considered by MARTY.



**Figure 2:**  $b \rightarrow sy$  as a function of  $m_{12}^2$  in the THDM type 2 using HYPERIso (with/without MARTY).

### 3.3 Conclusions

HYPERIso is a software package based on SuperIso for flavour-physics computations. It is model-independent at its core thanks to MARTY, making it very efficient for constraining many BSM models. It is also able to reproduce all SUPERIso's calculations up to NNLO. New studies on different models ( $Z'$ , leptiquarks, etc.) have begun with HYPERIso and will yield new results soon.

## References

- [1] F. Mahmoudi, *SuperIso: A program for calculating the isospin asymmetry of  $B \rightarrow K^* \gamma$  in the MSSM*, *Comput. Phys. Commun.* **178** (2008) 745–754 [arXiv:0710.2067 [hep-ph]].
- [2] F. Mahmoudi, *SuperIso v3.0: A program for calculating flavor physics observables in 2HDM and supersymmetry*, *Comput. Phys. Commun.* **180** (2009) 1579–1613 [arXiv:0808.3144 [hep-ph]].
- [3] F. Mahmoudi, *SuperIso v3.0, flavor physics observables calculations: Extension to NMSSM*, *Comput. Phys. Commun.* **180** (2009) 1718–1719.
- [4] S. Neshatpour and F. Mahmoudi, *Flavour Physics Phenomenology with SuperIso*, *PoS Comp-Tools2021* (2022) 010 [arXiv:2207.04956 [hep-ph]].
- [5] G. Uhlich, F. Mahmoudi and A. Arbey, *MARTY – Modern ARTificial Theoretical physicist: A C++ framework automating theoretical calculations beyond the Standard Model*, *Comput. Phys. Commun.* **264** (2021) 107928 [arXiv:2011.02478].
- [6] G. Uhlich, F. Mahmoudi and A. Arbey, *Automatic extraction of one-loop Wilson coefficients in general BSM scenarios using MARTY-1.4*, *PoS EPS-HEP2021* (2022) 507 [arXiv:2110.14515 [hep-ph]].
- [7] B. C. Allanach *et al.*, *SUSY Les Houches Accord 2*, *Comput. Phys. Commun.* **180** (2009) 8–25 [arXiv:0801.0045 [hep-ph]].
- [8] S. Navas *et al.* [Particle Data Group Collaboration], *Review of Particle Physics*, *Phys. Rev. D* **110** (2024) 030001.
- [9] W. Jakob, J. Rhinelander and D. Moldovan, *pybind11 — Seamless operability between C++11 and Python*, <https://github.com/pybind/pybind11> (2016).