

## **MARTY, a new C++ framework for automated symbolic calculations in Beyond the Standard Model physics**

---

**Grégoire Uhlich,<sup>a,\*</sup> Farvah Mahmoudi<sup>a,b,c</sup> and Alexandre Arbey<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, 103 boulevard Saint-Michel, 75005 Paris, France*

*E-mail: [g.uhlich@ipnl.in2p3.fr](mailto:g.uhlich@ipnl.in2p3.fr)*

Theoretical calculations Beyond the Standard Model (BSM) constitute a challenge for high energy physicists, but are necessary when searching for New Physics. The predictions of a BSM scenario need to be compared with experimental data and the Standard Model values in order to identify the model that fits better what we observe in particle colliders. BSM predictions require very involved and error prone calculations of amplitudes, cross-sections and Wilson coefficients. Calculations at the one-loop level are often necessary for these quantities since some phenomenologically important processes may not occur at tree-level, such as Flavor Changing Neutral Currents (FCNC) in flavor physics that vanish at the tree-level. One-loop calculations have to be done analytically which is very time consuming and in practice rarely done for general BSM models. Here we present *MARTY*, a public and open-source C++ code. *MARTY* is the very first independent program able to calculate automatically amplitudes, squared amplitudes and Wilson coefficients at the one-loop level for general BSM models. This type of calculations requires a computer algebra system and could only be done, up to now, using Mathematica, a commercial and closed software for symbolic manipulations. *MARTY* does not rely on Mathematica since it re-implements its own computer algebra system also in C++, called *CSL*. *MARTY* will considerably ease BSM studies as by automating the analytical calculations the main task of the user would be the model building part. Once interfaced with other phenomenological codes in particle physics, *MARTY* will be incredibly efficient to make detailed predictions for general BSM models automatically.

*40th International Conference on High Energy physics - ICHEP2020  
July 28 - August 6, 2020  
Prague, Czech Republic (virtual meeting)*

---

\*Speaker

## 1. Introduction

Calculations in physics Beyond the Standard Model (BSM) have always been a challenge, especially at the one-loop order. Transition amplitudes, differential cross-sections, Wilson coefficients, etc. need to be evaluated in order to make predictions for physical observables. Then statistical comparisons of the predictions against the SM and experimental values allow us to discriminate between BSM models and constrain their parameters.

Calculation at the one-loop level is often necessary, as some physically interesting processes only appear at this order. This is the case of Flavor Changing Neutral Currents (FCNC) in flavor physics, that generally require one-loop level values for Wilson coefficients. At the one-loop level, automatic calculations cannot be performed numerically, and the very large number of terms together with renormalization processes require symbolic computations, as one would do by hand.

Up to now – at the one-loop level – symbolic calculations could be done mainly by hand or using Mathematica [1], a commercial and closed software dedicated to this task. Several packages written with Mathematica have implemented high-energy physics features. FeynRules [2] computes Feynman rules from a BSM Lagrangian, from which FormCalc [3] can derive tree-level and one-loop quantities such as transition amplitudes or differential cross-sections. Packages such as FormFlavor [4] or FlavorKit [5, 6] can also calculate Wilson coefficients using the FormCalc machinery.

Many efforts have been made to develop independent computer algebra systems for high-energy physics. LanHEP [7], CompHEP [8] and CalcHEP [9] automate together calculations from the Lagrangian using their own symbolic computation framework. However Mathematica packages are still the only ones to provide one-loop calculations and Wilson coefficients in BSM scenarios.

With *MARTY* [10] we introduce a modern solution to this issue. It is a free and open-source code fully written in modern C++ (2017 standard) which implements the theoretical BSM machinery required to calculate amplitudes, squared amplitudes and Wilson coefficients up to the one-loop order. *MARTY* does not depend on any other framework since it has its own C++ Symbolic computation Library (*CSL*) to manipulate mathematical expressions. For all the reasons cited above, *MARTY* is unique and could be easily extended in the future to more models and calculations.

## 2. Design

*MARTY* is composed of three modules (see Figure 1). *CSL* is a pure symbolic calculation library, containing no physics. It includes features allowing to store and manipulate mathematical expressions, in particular with indicial tensors (such as  $\gamma$ -matrices  $\gamma_{\alpha\beta}^{\mu}$ ). Then the physics core of *MARTY* implements model building utilities and all the group theory and quantum field theory features needed to perform one-loop calculations for BSM physics. Finally, *GRAFED* (Generating and Rendering Application for FEynman Diagrams) generates automatically Feynman diagram layouts, and displays them via a C++/Qt [11] application<sup>1</sup>.

Every calculation in *MARTY* is automatic, symbolic, and at the one-loop level. Possible outputs are Feynman rules (from a Lagrangian), transition amplitudes, differential cross-sections, and Wilson coefficients. Inputs have to be given in a *MARTY script* written in C++ language.

<sup>1</sup>*GRAFED* uses only the free and open-source part of Qt.



calculated. These quantities may be fully simplified automatically at tree-level, or at one-loop with at most five external particles. One-loop momentum integrals, once reduced symbolically to scalar quantities, are evaluated numerically using the dedicated Fortran library LoopTools [13].

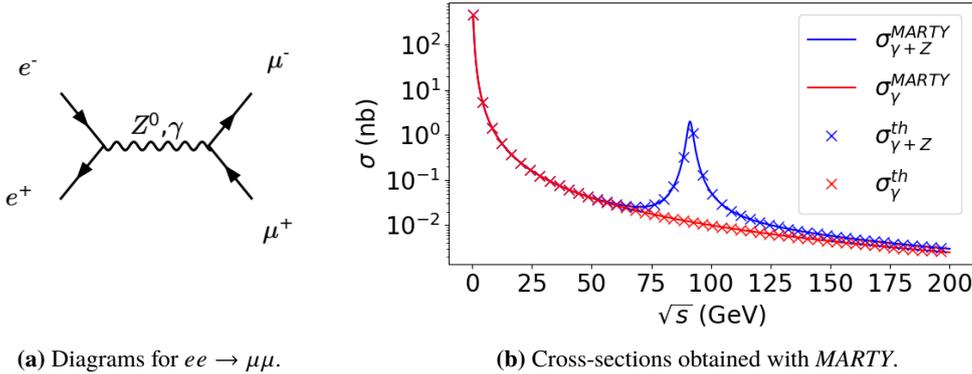
For the first release of *MARTY*, only Leading Order (LO) contributions are included. One-loop results from *MARTY* can be used when the tree-level quantities vanish, such as FCNC processes in the SM. One-loop corrections to tree-level processes will be treated in a next version. These will include renormalization (couplings, masses, fields), Renormalization Group Equations (RGEs) and operator mixings for Wilson coefficients.

*MARTY* generates C++ code automatically, allowing the user to perform phenomenological analyses. Any mathematical expression (*MARTY*'s output typically) may be written as numerical functions in a C++ library, taking as parameter every symbol that has no specified value. These libraries are also built automatically and may be called from a C++ program or directly using Python to facilitate further analysis.

## 4. Validation

### 4.1 Cross-section at the tree-level

Figure 2a shows the two diagrams contributing to the  $e^+e^- \rightarrow \mu^+\mu^-$  process at tree-level in the SM. They are both s-channels, with a photon or  $Z$ -boson exchange. The cross-section is thus shaped around two resonances: The photon resonance ( $m_\gamma = 0$  GeV) and the  $Z$  one ( $M_Z \approx 91.2$  GeV). The full cross-section calculation has been done with *MARTY*. The bare output is the differential cross-section. The phase space integration is not done automatically but is straightforward to implement in Python from *MARTY*'s output, a numerical function taking external momenta as parameters. Results are compared to analytical formulas calculated by hand, with or without the  $Z$  contribution. This is shown in Figure 2b.



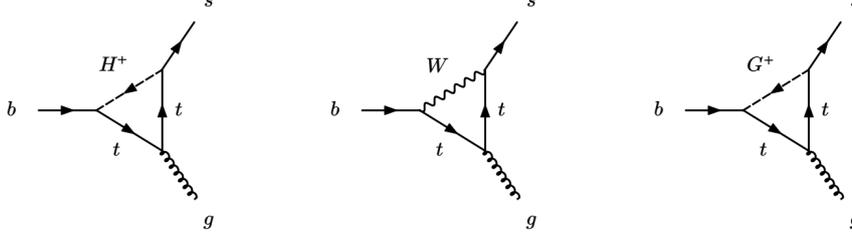
**Figure 2:**  $\sigma^{\text{SM}}(e^+e^- \rightarrow \mu^+\mu^-)$  at tree-level. Results with (blue) and without (red) the  $Z$  boson for *MARTY*'s output (solid lines) and analytical values (crosses) match perfectly. The mean relative error is  $\langle \Delta \rangle \approx 10^{-6}$ . The  $Z$  resonance can be seen around  $\sqrt{s} = M_Z = 91.2$  GeV.

The  $Z$ -resonance appears clearly, with a maximum value  $\sigma_{peak} \approx 2$  nb. The symbolic part of the calculation (from the Lagrangian to the squared amplitude) is performed automatically by

*MARTY*. From the squared amplitude to the final cross-section value, purely numerical computations are performed and no theoretical subtleties are left. This part is deferred to the user and can be done either in C++ or in Python using libraries generated by *MARTY*.

#### 4.2 Wilson coefficient at the one-loop level

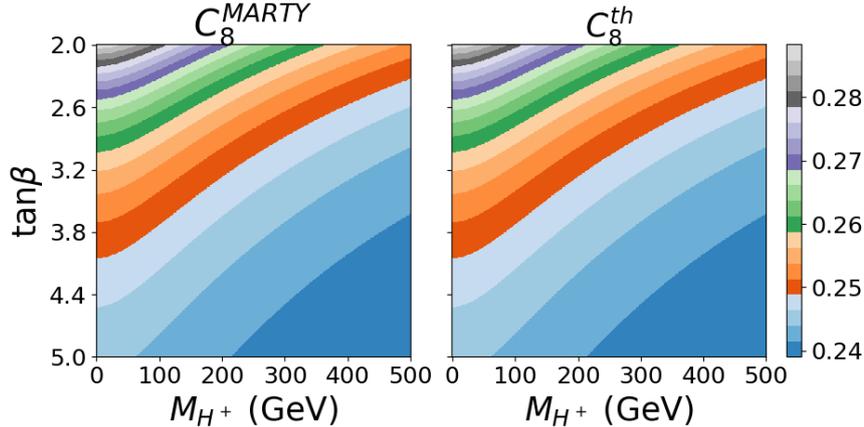
A more complicated example is the FCNC  $b \rightarrow sg$  process. We consider here non-primed coefficients only<sup>3</sup>.



**Figure 3:** Examples of contributions to the Wilson coefficients  $C_8$  in the 2HDM. Diagrams are generated with *GRAFED*.  $G^+$  is the Goldstone boson of  $W$ , and there are in addition charged Higgs  $H^+$  contributions specific to the 2HDM model.

The Leading Order (LO) is at the one-loop level and we perform the calculation on-shell keeping 1-Particle Reducible (1PR) diagrams. Calculation is done with *MARTY*, and the results are compared to the analytical formulas taken from the Superiso manual [14].

We performed this calculation with *MARTY* in the 2 Higgs Doublet Model (2HDM) of type IV (Lepton specific). There are SM ( $W$  and Goldstone  $G^+$ ) and charged Higgs ( $H^+$ ) contributions which depend on the charged Higgs mass  $M_{H^+}$  and  $\tan\beta$  that is the angle between the vacuum expectation values of the two Higgs doublets. Examples of contributions are presented in Figure 3. We vary  $M_{H^+}$  in  $[0, 500]$  GeV and  $\tan\beta$  in  $[2, 5]$ . Results are presented in Figure 4, where it can be seen that *MARTY*'s output is in perfect agreement with analytical formulas.



**Figure 4:**  $C_8^{2HDM}(M_W)$  as a function of the charged Higgs mass  $M_{H^+}$  and  $\tan\beta$  for *MARTY* on the left and analytical values on the right [14]. The mean relative error is  $\langle\Delta\rangle \approx 10^{-5}$ .

<sup>3</sup>Primed coefficients are suppressed by  $m_s/m_b$  and are often approximated to zero (when  $m_s$  is set to zero).

## 5. Conclusion and Outlook

We presented *MARTY*, a new C++ framework automating theoretical calculations symbolically in BSM physics. The degree of generality reached by *MARTY* has never been obtained before. It has its own symbolic manipulation library (*CSL*), and automates all theoretical calculations directly from the Lagrangian. Feynman rules, Feynman diagrams, amplitudes, cross-sections, and Wilson coefficients can be obtained in a very large variety of BSM models up to the one-loop level. A full NLO treatment will also be implemented in the near future, which is lacking in currently available codes.

A proof of its capabilities has been demonstrated by a tree-level cross-section calculation in the SM, and a 1-loop Wilson coefficient calculation in the 2HDM. Most of popular BSM models can be built in *MARTY*. The Minimal Supersymmetric extension of the Standard Model (MSSM), extended gauge models, vector-like quarks are examples of possible BSM implementations.

*MARTY* can be very useful for BSM phenomenology. One of its most significant advantages is to be written as a unique code, not depending on any other framework. *MARTY* embeds all aspects of model building and high energy physics calculations in the same C++ framework. This represents a unique opportunity for more collaborations to develop *MARTY* even further, extending it for other types of models, other simplification methods, or even different types of calculations.

## References

- [1] Wolfram Research Inc., Mathematica, <https://www.wolfram.com/mathematica>.
- [2] A. Alloul, N. D. Christensen, C. Degrande, C. Duhr, and B. Fuks, *Comput. Phys. Commun.* **185**, 2250 (2014), arXiv:1310.1921.
- [3] T. Hahn and M. Perez-Victoria, *Comput. Phys. Commun.* **118**, 153 (1999), hep-ph/9807565.
- [4] J. A. Evans and D. Shih, (2016), arXiv:1606.00003.
- [5] F. Staub, *Adv. High Energy Phys.* **2015**, 840780 (2015), arXiv:1503.04200.
- [6] W. Porod, F. Staub, and A. Vicente, *Eur. Phys. J. C* **74**, 2992 (2014), arXiv:1405.1434.
- [7] A. Semenov, *Comput. Phys. Commun.* **180**, 431 (2009), arXiv:0805.0555.
- [8] CompHEP, E. Boos *et al.*, *Nucl. Instrum. Meth. A* **534**, 250 (2004), arXiv:hep-ph/0403113.
- [9] A. Pukhov *et al.*, hep-ph/9908288.
- [10] G. Uhlich, F. Mahmoudi, and A. Arbey, arXiv:2011.02478.
- [11] The Qt Company, Qt open source model, <https://www.qt.io>.
- [12] P. Cvitanović, *Group theory*, Princeton University Press, 2008, ISBN-9780691202983.
- [13] T. Hahn and M. Perez-Victoria, *Comput. Phys. Commun.* **118**, 153 (1999), hep-ph/9807565.
- [14] F. Mahmoudi, *Comput. Phys. Commun.* **180**, 1579 (2009), arXiv:0808.3144.