2HDM Meeting.

Flavour Physics tools:

Part II: Flavorkit

Víctor Martín Lozano
victor.lozano@desy.de

Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Flavorkit is a flavour physics tool based on SARAH and SPheno developed by Avelino Vicente.

- Observables calculator
- Analytical and numerical computations
- Analytical expression of the Wilson coefficients
- It allows new observables regardless the BSM model.

Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Spirit of flavorkit.

Step 1: Consideration of an effective Lagrangian that includes the operators relevant for the flavor observable.

$$\mathcal{L}_{eff} = \sum_i C_i \mathcal{O}_i \,.$$

The Lagrangian includes a list of higher-dimensional operators and the Wilson coefficients can be induced either at tree or 1-loop levels and include the SM and BSM contributions in this way:

$$C_i = C_i^{\mathrm{SM}} + C_i^{\mathrm{BSM}}$$

Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Spirit of flavorkit.

Step 2: The Wilson coefficients are computed diagrammatically, taking into account all possible tree-level and 1-loop topologies leading to the operators.

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

Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Example: $\mathrm{BR}(\mu \to e\gamma)$ in the SM + RH-neutrinos

At leading order we only have dipole interactions, $\ell_\alpha \to \ell_\beta \gamma$,

$$\mathcal{L}_{\ell\ell\gamma}^{\mathrm{dipole}} = ie\, m_{\ell_\alpha}\, \bar{\ell}_\beta \sigma^{\mu\nu} q_\nu \left(K_2^L P_L + K_2^R P_R \right) \ell_\alpha A_\mu + \mathrm{h.c.}$$

Flavorkit.

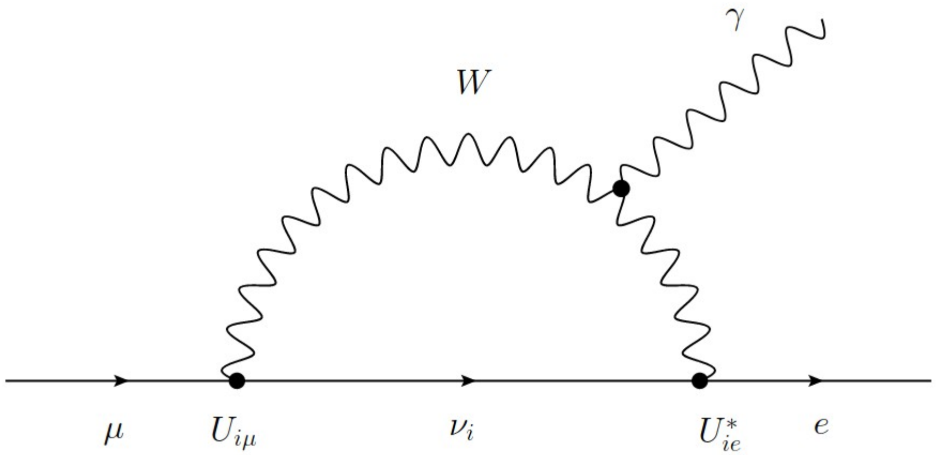
https://sarah.hepforge.org/FlavorKit.html

Example: $\mathrm{BR}(\mu \to e\gamma)$ in the SM + RH-neutrinos

At leading order we only have dipole interactions, $\ell_\alpha \to \ell_\beta \gamma$,

$$\mathcal{L}_{\ell\ell\gamma}^{\text{dipole}} = ie\, m_{\ell_\alpha}\, \bar{\ell}_\beta \sigma^{\mu\nu} q_\nu \left(K_2^L P_L + K_2^R P_R\right) \ell_\alpha A_\mu + \text{h.c.}$$

The information of the model is encoded in $K_2^{L,R}$ and they are obtained after summing up all Feynman diagrams contributing at a given loop level.



Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Example: $\mathrm{BR}(\mu \to e\gamma)$ in the SM + RH-neutrinos

At leading order we only have dipole interactions, $\ell_\alpha \to \ell_\beta \gamma$,

$$\mathcal{L}_{\ell\ell\gamma}^{\text{dipole}} = ie\, m_{\ell_\alpha}\, \bar{\ell}_\beta \sigma^{\mu\nu} q_\nu \left(K_2^L P_L + K_2^R P_R \right) \ell_\alpha A_\mu + \text{h.c.}$$

The information of the model is encoded in $K_2^{L,R}$ and they are obtained after summing up all Feynman diagrams contributing at a given loop level.

$$K_2^L = \frac{G_F}{2\sqrt{2}\pi^2} m_\mu \sum_i \lambda_{i\mu} \lambda_{ie}^* (F_1 + F_2)$$

$$K_2^R = \frac{G_F}{2\sqrt{2}\pi^2} m_e \sum_i \lambda_{i\mu} \lambda_{ie}^* (F_1 - F_2)$$

λ_{ij} Pontecorvo-Maki-Nakagara-Sakata entries, F_1 and F_2 are loop funtions.

Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Example: $\mathrm{BR}(\mu \to e\gamma)$ in the SM + RH-neutrinos

After computing the Wilson coefficients $K_2^{L,R}$, we can write the contribution in the BSM model:

$$\Gamma\left(\ell_\alpha \to \ell_\beta \gamma\right) = \frac{\alpha m_{\ell_\alpha}^5}{4}\left(|K_2^L|^2 + |K_2^R|^2\right)$$

If we have a new model and we need a prediction for $\mathrm{BR}(\mu \to e\gamma)$ the main task is to compute the Wilson coefficients. However, this could be a hard task to do.

Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Usage:

If we want to compute flavour observables in a specific BSM model computing the Wilson coefficients to have predictions, one needs:

- Information of the vertices, mass matrices.
- RGEs to get the running parameters at the considered scale.
- Expressions to compute the operators.
- Formulas to obtain the observables from the operators.

Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Usage:

If we want to compute flavour observables in a specific BSM model computing the Wilson coefficients to have predictions, one needs:

- Information of the vertices, mass matrices.
- RGEs to get the running parameters at the considered scale.
- Expressions to compute the operators.
- Formulas to obtain the observables from the operators.

In order to obtain all this information Flavorkit makes use of SARAH and SPheno.

Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Basic Usage:

Included by default. Already included in SARAH when creating a new model. The observables included are:

Lepton flavor	Quark flavor
$\ell_\alpha \to \ell_\beta \gamma$	$B^0_{s,d} \to \ell^+ \ell^-$
$\ell_\alpha \to 3\,\ell_\beta$	$\bar{B} \to X_s \gamma$
$\mu - e$ conversion in nuclei	$\bar{B} \to X_s \ell^+ \ell^-$
$\tau \to P\,\ell$	$\bar{B} \to X_{d,s} \nu\bar{\nu}$
$h \to \ell_\alpha \ell_\beta$	$B \to K \ell^+ \ell^-$
$Z \to \ell_\alpha \ell_\beta$	$K \to \pi \nu \bar{\nu}$
	$\Delta M_{B_{s,d}}$
	ΔM_K and ε_K
	$P \to \ell \nu$

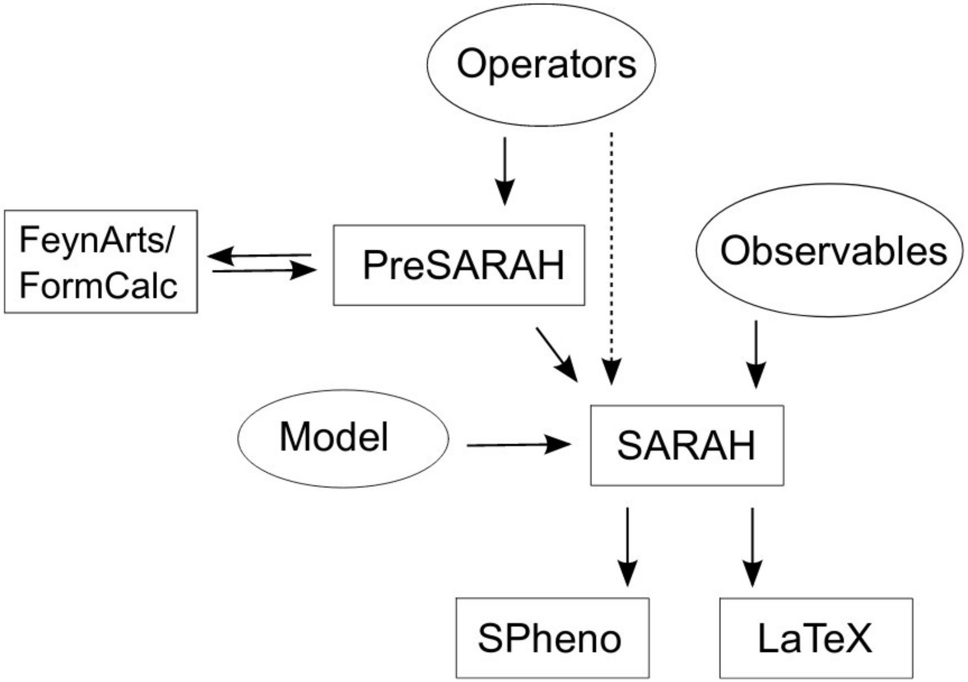
Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Advanced Usage:

The observable of study is not included in the previous list.

One has to use PreSARAH that is related with other tools (FeynArts/FormCalc) and produce the code for SPheno.



Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Advanced Usage:

-Name of the process

-Considered process: 4 fermions, 2 fermions 1 scalar, 2 fermions 1 vector

-External fields: Charged lepton, neutrino, down/up quark, scalar Higgs, pseudoscalar Higgs, Z/W boson

Flavorkit.

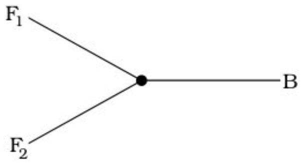
https://sarah.hepforge.org/FlavorKit.html

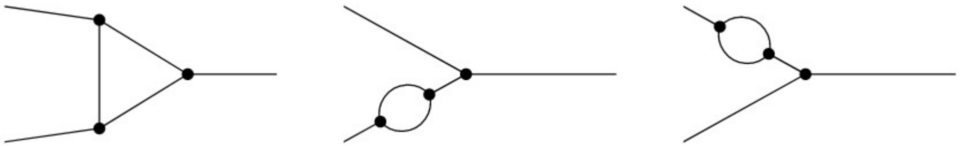
Advanced Usage:

-Name of the process

-Considered process: 4 fermions, 2 fermions 1 scalar, 2 fermions 1 vector

-External fields: Charged lepton, neutrino, down/up quark, scalar Higgs, pseudoscalar Higgs, Z/W boson





Flavorkit.

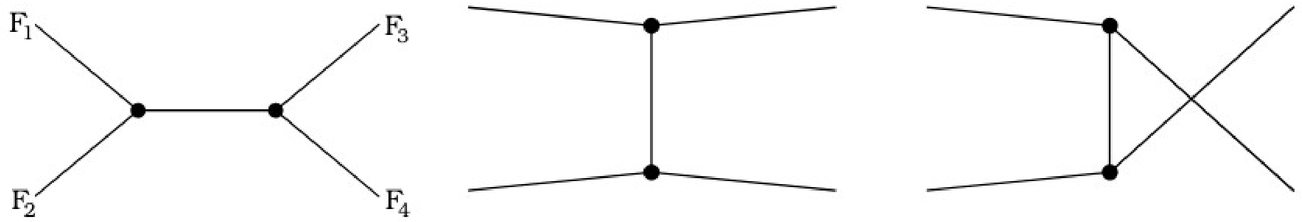
https://sarah.hepforge.org/FlavorKit.html

Advanced Usage:

-Name of the process

-Considered process: 4 fermions, 2 fermions 1 scalar, 2 fermions 1 vector

-External fields: Charged lepton, neutrino, down/up quark, scalar Higgs, pseudoscalar Higgs, Z/W boson



Flavorkit.

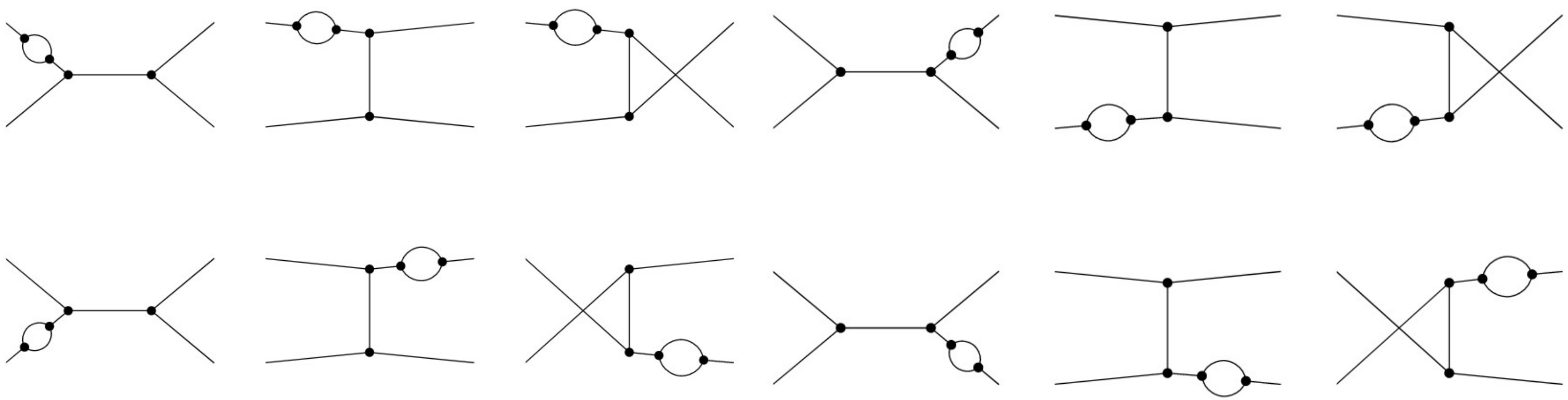
https://sarah.hepforge.org/FlavorKit.html

Advanced Usage:

-Name of the process

-Considered process: 4 fermions, 2 fermions 1 scalar, 2 fermions 1 vector

-External fields: Charged lepton, neutrino, down/up quark, scalar Higgs, pseudoscalar Higgs, Z/W boson



Flavorkit.

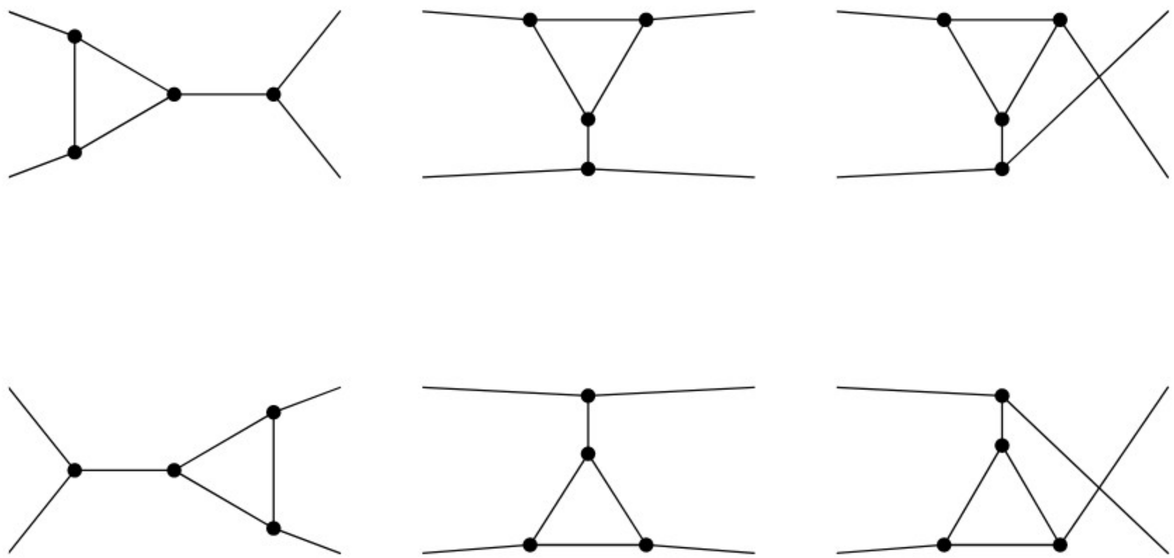
https://sarah.hepforge.org/FlavorKit.html

Advanced Usage:

-Name of the process

-Considered process: 4 fermions, 2 fermions 1 scalar, 2 fermions 1 vector

-External fields: Charged lepton, neutrino, down/up quark, scalar Higgs, pseudoscalar Higgs, Z/W boson



Flavorkit.

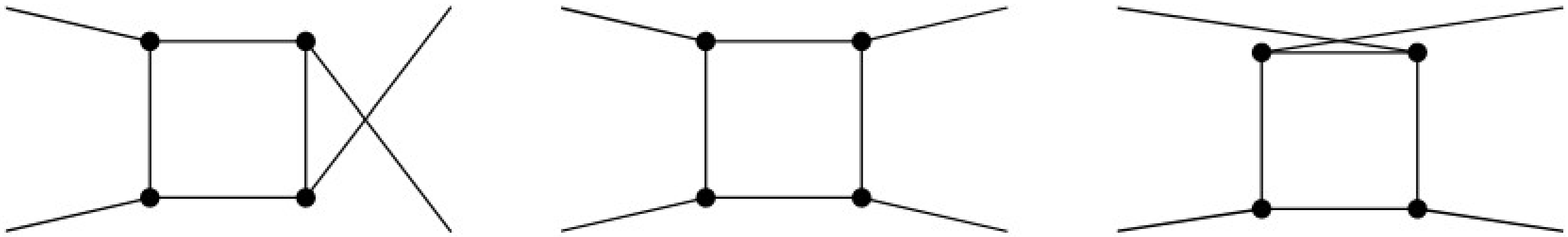
https://sarah.hepforge.org/FlavorKit.html

Advanced Usage:

-Name of the process

-Considered process: 4 fermions, 2 fermions 1 scalar, 2 fermions 1 vector

-External fields: Charged lepton, neutrino, down/up quark, scalar Higgs, pseudoscalar Higgs, Z/W boson



Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Advanced Usage:

Example decay $\ell_\alpha \to \ell_\beta \gamma$ with decay width $\Gamma\left(\ell_\alpha \to \ell_\beta \gamma\right) = \frac{\alpha m_{\ell_\alpha}^5}{4}\left(|K_2^L|^2 + |K_2^R|^2\right)$

LLgGamma.m

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

NeededOperators = {K2L, K2R};

Body = "LLpGamma.f90";
```

Flavorkit.

https://sarah.hepforge.org/FlavorKit.html

Advanced Usage:

Example decay $\ell_\alpha \to \ell_\beta \gamma$

LLgGamma.f90

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

! ------------------------------------------------------------------
! l -> l' gamma
! Observable implemented by W. Porod, F. Staub and A. Vicente
! Based on J. Hisano et al, PRD 53 (1996) 2442 [hep-ph/9510309]
! ------------------------------------------------------------------

Do i1=1,3

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

width=0.25_dp*mf_l(gt1)**5*(Abs(K2L(gt1,gt2))**2 &
           & +Abs(K2R(gt1,gt2))**2)*Alpha

If (i1.eq.1) Then
muEgamma = width/(width+GammaMu)
Elseif (i1.eq.2) Then
tauEgamma = width/(width+GammaTau)
Else
tauMuGamma = width/(width+GammaTau)
End if

End do
```

Thank you for your attention!