FeynRules

Lecture II

Claude Duhr

MadGraph School 2013, Beijing, 22-26/05/13

FeynRules: the basics

- In the previous lecture we learned the basic usage of FeynRules:

  ➡ Implement a model.

  ➡ Compute Feynman rules.

  ➡ Use the interfaces.

- While this basic usage is in principle enough to implement any model, doing so can be quite tedious in practise.

Supersymmetric models

- Example I: Supersymmetric models.
  - ➡ SUSY models are very compact when written in superspace notation.
  - ➡ Matrix element generators require Feynman rules given for the component fields.
  - ➡ Thus, FeynRules requires the Lagrangian for the component fields.
  - ➡ The component field Lagrangian can be extremely complicated!

Masses

- Example II: Mass spectra
  - ➡ In most BSM scenarios, the mass matrices are not diagonal, but we need to diagonalize the mass matrices.
  - ➡ For simple models, this can be done analytically, and the analytic formulas can be implemented as internal parameters (cf. tutorial).
  - ➡ In most models, the diagonalization cannot be performed analytically (cf. 6x6 squark mixing matrix).
  - ➡ Could use Mathematica to numerically diagonalize matrices, but want to avoid to rerun FeynRules for every benchmark point.

Decay rates

- Example III: Widths and branching ratios
  - ➡ Matrix element generators require the widths of the particles.
  - ➡ Need to recompute all the widths for every benchmark point (some generators do this on the fly).
  - ➡ Cannot use analytic formulas for the width as internal parameters, as allowed channels are benchmark dependent.
  - ➡ Want to avoid to rerun FeynRules for every benchmark point.

FeynRules

- Aim of the lecture: Give an introduction to the Mathematica package FeynRules.

- Lecture I: The basics.

  How to implement a model and compute its Feynman rules.

- Lecture II: Advanced topics.
  - ➡ SUSY
  - ➡ Computing two-body decays.
  - ➡ Spectrum generation with ASperGe.
  - ➡ Towards NLO.

Superfields in FeynRules

The lifecycle of SUSY pheno

- Example: SUSY model

$$\mathcal{L} = \Phi^\dagger e^{-2gV} \Phi_{|_{\theta^2\bar\theta^2}} + \frac{1}{16 g^2 \tau_{\mathcal{R}}} \mathrm{Tr}(W^\alpha W_\alpha)_{|_{\theta^2}} + \frac{1}{16 g^2 \tau_{\mathcal{R}}} \mathrm{Tr}(\bar W_{\dot\alpha} \bar W^{\dot\alpha})_{|_{\bar\theta^2}}$$
$$+ W(\Phi)_{|_{\theta^2}} + W^\star(\Phi^\dagger)_{|_{\bar\theta^2}} + \mathcal{L}_{\rm soft}$$

- Very easy 'theory description'
  - ➡ Choose a gauge group (+ additional internal symmetries).
  - ➡ Choose the matter content (= chiral superfields in some representation).
  - ➡ Write down the most general superpotential.
  - ➡ Write down the soft-SUSY breaking terms.
  - ➡ (+ check validity of the model)

The lifecycle of SUSY pheno

- Example: SUSY model

$$\mathcal{L} = \Phi^\dagger e^{-2gV}\Phi_{|_{\theta^2\bar\theta^2}} + \frac{1}{16g^2\tau_{\mathcal{R}}}\mathrm{Tr}(W^\alpha W_\alpha)_{|_{\theta^2}} + \frac{1}{16g^2\tau_{\mathcal{R}}}\mathrm{Tr}(\bar W_{\dot\alpha}\bar W^{\dot\alpha})_{|_{\bar\theta^2}}$$
$$+ W(\Phi)_{|_{\theta^2}} + W^\star(\Phi^\dagger)_{|_{\bar\theta^2}} + \mathcal{L}_{\rm soft}$$

- 'Monte Carlo description'
  - ➡ Express superfields in terms of component fields.
  - ➡ Express everything in terms of 4-component fermions (beware of the Majoranas!).
  - ➡ Express everything in terms of mass eigenstates.
  - ➡ Integrate out D and F terms.
  - ➡ Implement vertices one-by-one (beware of factors of *i*, *etc*!)

Supersymmetric models

- FeynRules allows to use the superfield formalism for supersymmetric theories.
- The code then
  - ➡ expands the superfields in the Grassmann variables and integrates them out.
  - ➡ Weyl fermions are transformed into 4-component spinors.
  - ➡ auxiliary fields are integrated out.
- As a result, we obtain a Lagrangian that can be exported to matrix element generators!

Supersymmetric models

- Example: SUSY QCD
  - ➡ 1 octet vector superfield $V^a = (\tilde{g}^a, G^a_\mu, D^a)$
  - ➡ 1 triplet left-handed chiral superfield $Q^i_L = (\tilde{q}^i_L, \chi^i, F^i_L)$
  - ➡ 1 triplet right-handed chiral superfield $Q^i_R = (\tilde{q}^i_R, \bar{\xi}^i, F^i_R)$

- The physical spectrum contains
  - ➡ a gauge boson, the gluon
  - ➡ two complex triplet scalars
  - ➡ an octet Majorana fermion
  - ➡ a triplet Dirac fermion, the quark $q^i = (\chi^i, \bar{\xi}^i)$

Supersymmetric models

- Interactions (almost) entirely fixed by SUSY

$$Q_L^\dagger\, e^{-2g_s V}\, Q_L + Q_R^\dagger\, e^{-2g_s V}\, Q_R + \frac{1}{8g_s^2}\mathrm{Tr}(W^\alpha W_\alpha) + \frac{1}{8g_s^2}\mathrm{Tr}(\overline{W}_{\dot\alpha}\overline{W}^{\dot\alpha})$$
$$+W(Q_L, Q_R^\dagger) + W^\star(Q_L^\dagger, Q_R)$$

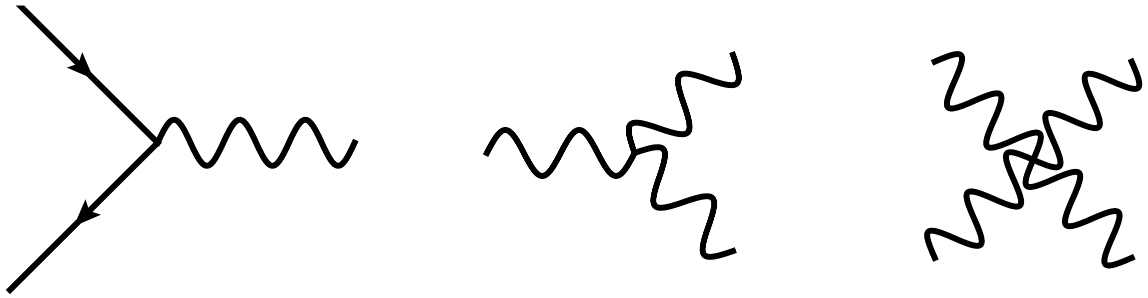
- The gauge sector is already rather complicated in terms of component fields...

Supersymmetric models

- Interactions (almost) entirely fixed by SUSY

$$Q_L^\dagger\, e^{-2g_s V}\, Q_L + Q_R^\dagger\, e^{-2g_s V}\, Q_R + \frac{1}{8g_s^2}\mathrm{Tr}(W^\alpha W_\alpha) + \frac{1}{8g_s^2}\mathrm{Tr}(\overline{W}_{\dot\alpha}\overline{W}^{\dot\alpha})$$
$$+W(Q_L, Q_R^\dagger) + W^\star(Q_L^\dagger, Q_R)$$

- The gauge sector is already rather complicated in terms of component fields...

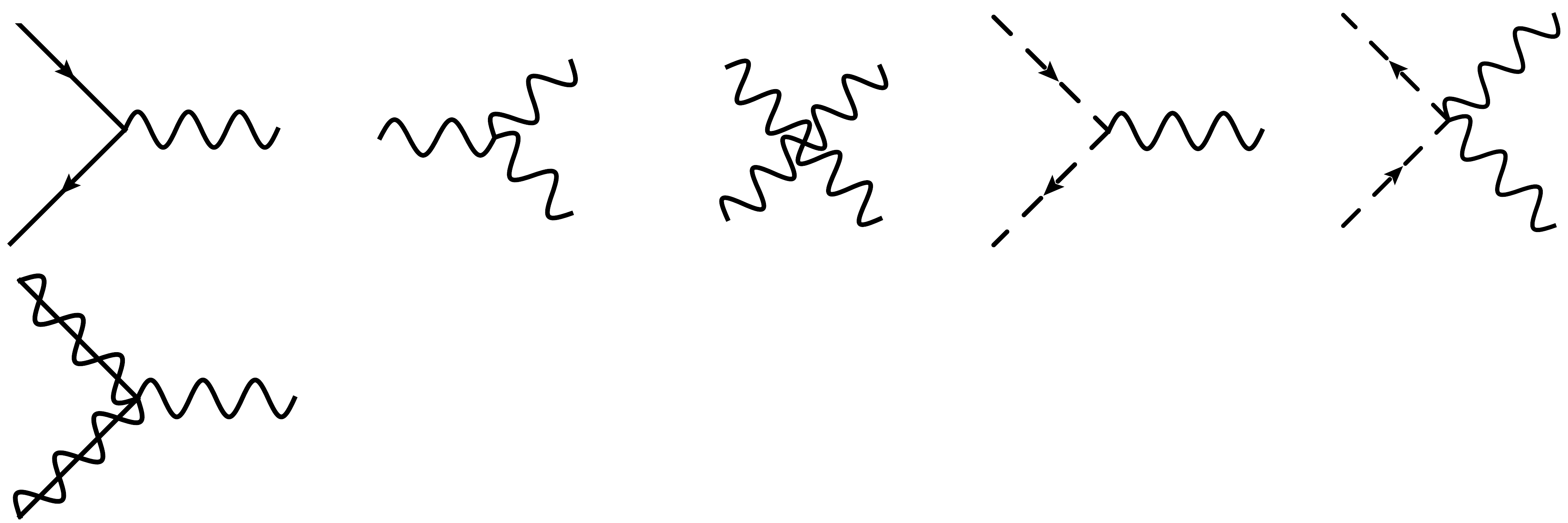


Supersymmetric models

- Interactions (almost) entirely fixed by SUSY

$$Q_L^\dagger\, e^{-2g_sV}\, Q_L + Q_R^\dagger\, e^{-2g_sV}\, Q_R + \frac{1}{8g_s^2}\mathrm{Tr}(W^\alpha W_\alpha) + \frac{1}{8g_s^2}\mathrm{Tr}(\overline{W}_{\dot\alpha}\overline{W}^{\dot\alpha})$$
$$+W(Q_L, Q_R^\dagger) + W^\star(Q_L^\dagger, Q_R)$$

- The gauge sector is already rather complicated in terms of component fields...

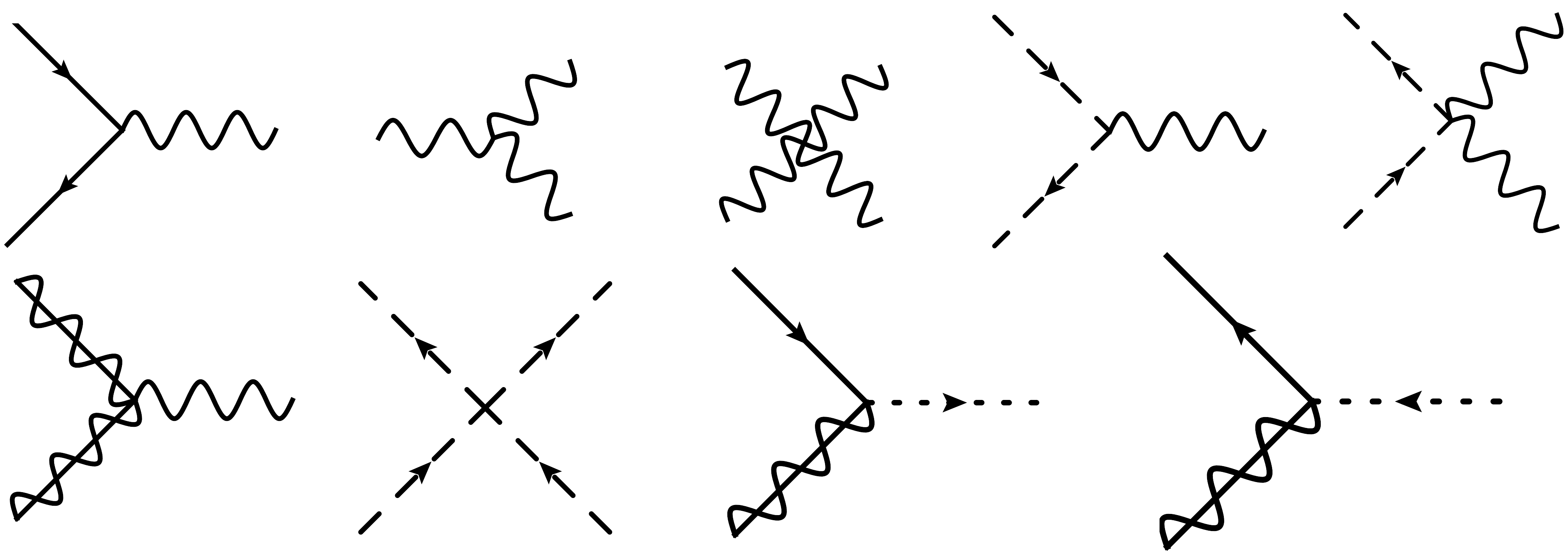


Supersymmetric models

- Interactions (almost) entirely fixed by SUSY

$$Q_L^\dagger\, e^{-2g_s V}\, Q_L + Q_R^\dagger\, e^{-2g_s V}\, Q_R + \frac{1}{8g_s^2}\mathrm{Tr}(W^\alpha W_\alpha) + \frac{1}{8g_s^2}\mathrm{Tr}(\overline{W}_{\dot\alpha}\overline{W}^{\dot\alpha})$$
$$+W(Q_L, Q_R^\dagger) + W^\star(Q_L^\dagger, Q_R)$$

- The gauge sector is already rather complicated in terms of component fields...



Defining superfields

```
VSF[1] == { ClassName   -> GSF,
            GaugeBoson  -> G,
            Gaugino     -> gow,
            Indices     -> {Index[Gluon]}},

CSF[1] == { ClassName    -> QL,
             Chirality   -> Left,
             Weyl        -> qLw,
             Scalar      -> QLs,
             Indices->{Index[Colour]}},
```

$$V^a = (\tilde{g}^a, G^a_\mu, D^a)$$

$$Q^i_L = (\tilde{q}^i_L, \chi^i, F^i_L)$$

- The component fields are defined separately.
- Auxiliary F and D fields could be added, but can be left out, and are created on the fly.

Using superfields

```
WS = ...
SL = VSFKineticTerms[] + CSFKineticTerms[] + WS + HC[WS];
```

- A set of functions allows to transform the superspace action into a component field Lagrangian.
  - ➡ SF2Components: expansion in the Grassmann parameters
  - ➡ ThetaThetabarComponent etc.: selects the desired coefficient in the Grassmann expansion.
  - ➡ SolveEqMotionF/SolveEqMotionD: solves the equations of motion for the F and D terms.
  - ➡ WeylToDirac: Transforms Weyl fermions into 4-component fermions.

Using superfields

$$-4\, f_{\mathrm{Gluon\$1},\mathrm{Gluon\$2},\mathrm{Gluon\$3}}\, f_{\mathrm{Gluon\$3},\mathrm{Gluon\$4},\mathrm{Gluon\$5}}\, G_{\mathrm{mu\$1},\mathrm{Gluon\$1}}\, G_{\mathrm{mu\$1},\mathrm{Gluon\$4}}\, G_{\mathrm{mu\$2},\mathrm{Gluon\$2}}\, G_{\mathrm{mu\$2},\mathrm{Gluon\$5}}\, \mathrm{gs}^4 +$$

$$16\, \partial_{\mathrm{mu\$2}}\left(G_{\mathrm{mu\$1},\mathrm{Gluon\$1}}\right) f_{\mathrm{Gluon\$1},\mathrm{Gluon\$2},\mathrm{Gluon\$3}}\, G_{\mathrm{mu\$1},\mathrm{Gluon\$2}}\, G_{\mathrm{mu\$2},\mathrm{Gluon\$3}}\, \mathrm{gs}^3 + 8\, i\, \bar{\mathrm{go}}_{\mathrm{r\$28685},\mathrm{Gluon\$2}} \cdot \mathrm{go}_{\mathrm{r\$28698},\mathrm{Gluon\$1}}\, f_{\mathrm{Gluon\$1},\mathrm{Gluon\$2},\mathrm{Gluon\$3}}\, G_{\mathrm{mu\$1},\mathrm{Gluon\$3}}\, \gamma^{\mathrm{mu\$1}} \cdot P_{-\mathrm{r\$28685},\mathrm{r\$28698}}\, \mathrm{gs}^3 -$$

$$8\, i\, \bar{\mathrm{go}}_{\mathrm{r\$28698},\mathrm{Gluon\$1}} \cdot \mathrm{go}_{\mathrm{r\$28685},\mathrm{Gluon\$2}}\, f_{\mathrm{Gluon\$1},\mathrm{Gluon\$2},\mathrm{Gluon\$3}}\, G_{\mathrm{mu\$1},\mathrm{Gluon\$3}}\, \gamma^{\mathrm{mu\$1}} \cdot P_{+\mathrm{r\$28698},\mathrm{r\$28685}}\, \mathrm{gs}^3 - 8\, \partial_{\mathrm{mu\$2}}\left(G_{\mathrm{mu\$1},\mathrm{Gluon\$1}}\right)^2 \mathrm{gs}^2 +$$

$$8\, \partial_{\mathrm{mu\$2}}\left(G_{\mathrm{mu\$1},\mathrm{Gluon\$1}}\right) \partial_{\mathrm{mu\$1}}\left(G_{\mathrm{mu\$2},\mathrm{Gluon\$1}}\right) \mathrm{gs}^2 + G_{\mathrm{mu\$1},\mathrm{Gluon\$1}}\, G_{\mathrm{mu\$1},\mathrm{Gluon\$2}}\, \mathrm{QLs}_{\mathrm{Colour\$1}}\, \mathrm{QLs}^{\dagger}_{\mathrm{Colour\$2}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2},\mathrm{Colour\$3}}\, T^{\mathrm{Gluon\$2}}_{\mathrm{Colour\$3},\mathrm{Colour\$1}}\, \mathrm{gs}^2 +$$

$$G_{\mathrm{mu\$1},\mathrm{Gluon\$1}}\, G_{\mathrm{mu\$1},\mathrm{Gluon\$2}}\, \mathrm{QRs}_{\mathrm{Colour\$1}}\, \mathrm{QRs}^{\dagger}_{\mathrm{Colour\$2}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2},\mathrm{Colour\$3}}\, T^{\mathrm{Gluon\$2}}_{\mathrm{Colour\$3},\mathrm{Colour\$1}}\, \mathrm{gs}^2 + 4\, i\, \bar{\mathrm{go}}_{\mathrm{r\$28679},\mathrm{Gluon\$1}} \cdot \partial_{\mathrm{mu\$1}}\left(\mathrm{go}_{\mathrm{r\$28692},\mathrm{Gluon\$1}}\right) \gamma^{\mathrm{mu\$1}} \cdot P_{-\mathrm{r\$28679},\mathrm{r\$28692}}\, \mathrm{gs}^2 -$$

$$4\, i\, \partial_{\mathrm{mu\$1}}\left(\bar{\mathrm{go}}_{\mathrm{r\$28682},\mathrm{Gluon\$1}}\right) \cdot \mathrm{go}_{\mathrm{r\$28695},\mathrm{Gluon\$1}}\, \gamma^{\mathrm{mu\$1}} \cdot P_{-\mathrm{r\$28682},\mathrm{r\$28695}}\, \mathrm{gs}^2 - 4\, i\, \partial_{\mathrm{mu\$1}}\left(\bar{\mathrm{go}}_{\mathrm{r\$28692},\mathrm{Gluon\$1}}\right) \cdot \mathrm{go}_{\mathrm{r\$28679},\mathrm{Gluon\$1}}\, \gamma^{\mathrm{mu\$1}} \cdot P_{+\mathrm{r\$28692},\mathrm{r\$28679}}\, \mathrm{gs}^2 +$$

$$4\, i\, \bar{\mathrm{go}}_{\mathrm{r\$28695},\mathrm{Gluon\$1}} \cdot \partial_{\mathrm{mu\$1}}\left(\mathrm{go}_{\mathrm{r\$28682},\mathrm{Gluon\$1}}\right) \gamma^{\mathrm{mu\$1}} \cdot P_{+\mathrm{r\$28695},\mathrm{r\$28682}}\, \mathrm{gs}^2 - i\, \partial_{\mathrm{mu\$1}}\left(\mathrm{QLs}^{\dagger}_{\mathrm{Colour\$1}}\right) G_{\mathrm{mu\$1},\mathrm{Gluon\$1}}\, \mathrm{QLs}_{\mathrm{Colour\$2}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$1},\mathrm{Colour\$2}}\, \mathrm{gs} +$$

$$i\sqrt{2}\, \bar{q}_{\mathrm{r\$28675},\mathrm{Colour\$1}} \cdot \mathrm{go}_{\mathrm{r\$28688},\mathrm{Gluon\$1}}\, P_{+\mathrm{r\$28675},\mathrm{r\$28688}}\, \mathrm{QLs}_{\mathrm{Colour\$2}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$1},\mathrm{Colour\$2}}\, \mathrm{gs} + i\, \partial_{\mathrm{mu\$1}}\left(\mathrm{QRs}^{\dagger}_{\mathrm{Colour\$1}}\right) G_{\mathrm{mu\$1},\mathrm{Gluon\$1}}\, \mathrm{QRs}_{\mathrm{Colour\$2}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$1},\mathrm{Colour\$2}}\, \mathrm{gs} -$$

$$i\sqrt{2}\, \bar{q}_{\mathrm{r\$28689},\mathrm{Colour\$1}} \cdot \mathrm{go}_{\mathrm{r\$28676},\mathrm{Gluon\$1}}\, P_{-\mathrm{r\$28689},\mathrm{r\$28676}}\, \mathrm{QRs}_{\mathrm{Colour\$2}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$1},\mathrm{Colour\$2}}\, \mathrm{gs} + i\, \partial_{\mathrm{mu\$1}}\left(\mathrm{QLs}_{\mathrm{Colour\$1}}\right) G_{\mathrm{mu\$1},\mathrm{Gluon\$1}}\, \mathrm{QLs}^{\dagger}_{\mathrm{Colour\$2}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2},\mathrm{Colour\$1}}\, \mathrm{gs} -$$

$$i\sqrt{2}\, \bar{\mathrm{go}}_{\mathrm{r\$28677},\mathrm{Gluon\$1}} \cdot q_{\mathrm{r\$28690},\mathrm{Colour\$1}}\, P_{-\mathrm{r\$28677},\mathrm{r\$28690}}\, \mathrm{QLs}^{\dagger}_{\mathrm{Colour\$2}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2},\mathrm{Colour\$1}}\, \mathrm{gs} - i\, \partial_{\mathrm{mu\$1}}\left(\mathrm{QRs}_{\mathrm{Colour\$1}}\right) G_{\mathrm{mu\$1},\mathrm{Gluon\$1}}\, \mathrm{QRs}^{\dagger}_{\mathrm{Colour\$2}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2},\mathrm{Colour\$1}}\, \mathrm{gs} +$$

$$i\sqrt{2}\, \bar{\mathrm{go}}_{\mathrm{r\$28691},\mathrm{Gluon\$1}} \cdot q_{\mathrm{r\$28678},\mathrm{Colour\$1}}\, P_{+\mathrm{r\$28691},\mathrm{r\$28678}}\, \mathrm{QRs}^{\dagger}_{\mathrm{Colour\$2}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2},\mathrm{Colour\$1}}\, \mathrm{gs} + \bar{q}_{\mathrm{r\$28687},\mathrm{Colour\$2}} \cdot q_{\mathrm{r\$28700},\mathrm{Colour\$1}}\, G_{\mathrm{mu\$1},\mathrm{Gluon\$1}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2},\mathrm{Colour\$1}}\, \gamma^{\mathrm{mu\$1}} \cdot P_{-\mathrm{r\$28687},\mathrm{r\$28700}}\, \mathrm{gs} -$$

$$\bar{q}_{\mathrm{r\$28699},\mathrm{Colour\$1}} \cdot q_{\mathrm{r\$28686},\mathrm{Colour\$2}}\, G_{\mathrm{mu\$1},\mathrm{Gluon\$1}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$1},\mathrm{Colour\$2}}\, \gamma^{\mathrm{mu\$1}} \cdot P_{+\mathrm{r\$28699},\mathrm{r\$28686}}\, \mathrm{gs} + \frac{1}{2}\, \partial_{\mathrm{mu\$1}}\left(\mathrm{QLs}_{\mathrm{Colour\$1}}\right) \partial_{\mathrm{mu\$1}}\left(\mathrm{QLs}^{\dagger}_{\mathrm{Colour\$1}}\right) +$$

$$\frac{1}{2}\, \partial_{\mathrm{mu\$1}}\left(\mathrm{QRs}_{\mathrm{Colour\$1}}\right) \partial_{\mathrm{mu\$1}}\left(\mathrm{QRs}^{\dagger}_{\mathrm{Colour\$1}}\right) - \frac{1}{4}\, \partial_{\mathrm{mu\$1}}\left(\partial_{\mathrm{mu\$1}}\left(\mathrm{QLs}^{\dagger}_{\mathrm{Colour\$1}}\right)\right) \mathrm{QLs}_{\mathrm{Colour\$1}} - \frac{1}{4}\, \partial_{\mathrm{mu\$1}}\left(\partial_{\mathrm{mu\$1}}\left(\mathrm{QLs}_{\mathrm{Colour\$1}}\right)\right) \mathrm{QLs}^{\dagger}_{\mathrm{Colour\$1}} - \frac{1}{4}\, \partial_{\mathrm{mu\$1}}\left(\partial_{\mathrm{mu\$1}}\left(\mathrm{QRs}^{\dagger}_{\mathrm{Colour\$1}}\right)\right) \mathrm{QRs}_{\mathrm{Colour\$1}} -$$

$$\frac{1}{4}\, \partial_{\mathrm{mu\$1}}\left(\partial_{\mathrm{mu\$1}}\left(\mathrm{QRs}_{\mathrm{Colour\$1}}\right)\right) \mathrm{QRs}^{\dagger}_{\mathrm{Colour\$1}} + \frac{1}{32}\, \mathrm{QLs}_{\mathrm{Colour\$1\$28637}}\, \mathrm{QLs}_{\mathrm{Colour\$1\$28639}}\, \mathrm{QLs}^{\dagger}_{\mathrm{Colour\$2\$28637}}\, \mathrm{QLs}^{\dagger}_{\mathrm{Colour\$2\$28639}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2\$28637},\mathrm{Colour\$1\$28637}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2\$28639},\mathrm{Colour\$1\$28639}} +$$

$$\frac{1}{32}\, \mathrm{QLs}_{\mathrm{Colour\$1\$28639}}\, \mathrm{QLs}^{\dagger}_{\mathrm{Colour\$2\$28639}}\, \mathrm{QRs}_{\mathrm{Colour\$1\$28638}}\, \mathrm{QRs}^{\dagger}_{\mathrm{Colour\$2\$28638}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2\$28638},\mathrm{Colour\$1\$28638}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2\$28639},\mathrm{Colour\$1\$28639}} +$$

$$\frac{1}{32}\, \mathrm{QLs}_{\mathrm{Colour\$1\$28637}}\, \mathrm{QLs}^{\dagger}_{\mathrm{Colour\$2\$28637}}\, \mathrm{QRs}_{\mathrm{Colour\$1\$28640}}\, \mathrm{QRs}^{\dagger}_{\mathrm{Colour\$2\$28640}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2\$28637},\mathrm{Colour\$1\$28637}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2\$28640},\mathrm{Colour\$1\$28640}} +$$

$$\frac{1}{32}\, \mathrm{QRs}_{\mathrm{Colour\$1\$28638}}\, \mathrm{QRs}_{\mathrm{Colour\$1\$28640}}\, \mathrm{QRs}^{\dagger}_{\mathrm{Colour\$2\$28638}}\, \mathrm{QRs}^{\dagger}_{\mathrm{Colour\$2\$28640}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2\$28638},\mathrm{Colour\$1\$28638}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2\$28640},\mathrm{Colour\$1\$28640}} -$$

$$\frac{1}{16}\, \mathrm{QLs}_{\mathrm{Colour\$1}}\, \mathrm{QLs}_{\mathrm{Colour\$1\$28641}}\, \mathrm{QLs}^{\dagger}_{\mathrm{Colour\$2}}\, \mathrm{QLs}^{\dagger}_{\mathrm{Colour\$2\$28641}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2},\mathrm{Colour\$1}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2\$28641},\mathrm{Colour\$1\$28641}} -$$

$$\frac{1}{16}\, \mathrm{QLs}_{\mathrm{Colour\$1}}\, \mathrm{QLs}^{\dagger}_{\mathrm{Colour\$2}}\, \mathrm{QRs}_{\mathrm{Colour\$1\$28642}}\, \mathrm{QRs}^{\dagger}_{\mathrm{Colour\$2\$28642}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2},\mathrm{Colour\$1}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2\$28642},\mathrm{Colour\$1\$28642}} -$$

$$\frac{1}{16}\, \mathrm{QLs}_{\mathrm{Colour\$1\$28643}}\, \mathrm{QLs}^{\dagger}_{\mathrm{Colour\$2\$28643}}\, \mathrm{QRs}_{\mathrm{Colour\$1}}\, \mathrm{QRs}^{\dagger}_{\mathrm{Colour\$2}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2},\mathrm{Colour\$1}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2\$28643},\mathrm{Colour\$1\$28643}} -$$

$$\frac{1}{16}\, \mathrm{QRs}_{\mathrm{Colour\$1}}\, \mathrm{QRs}_{\mathrm{Colour\$1\$28644}}\, \mathrm{QRs}^{\dagger}_{\mathrm{Colour\$2}}\, \mathrm{QRs}^{\dagger}_{\mathrm{Colour\$2\$28644}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2},\mathrm{Colour\$1}}\, T^{\mathrm{Gluon\$1}}_{\mathrm{Colour\$2\$28644},\mathrm{Colour\$1\$28644}} + \frac{1}{2}\, i\, \bar{q}_{\mathrm{r\$28680},\mathrm{Colour\$1}} \cdot \partial_{\mathrm{mu\$1}}\left(q_{\mathrm{r\$28693},\mathrm{Colour\$1}}\right) \gamma^{\mathrm{mu\$1}} \cdot P_{-\mathrm{r\$28680},\mathrm{r\$28693}} -$$

$$\frac{1}{2}\, i\, \partial_{\mathrm{mu\$1}}\left(\bar{q}_{\mathrm{r\$28683},\mathrm{Colour\$1}}\right) \cdot q_{\mathrm{r\$28696},\mathrm{Colour\$1}}\, \gamma^{\mathrm{mu\$1}} \cdot P_{-\mathrm{r\$28683},\mathrm{r\$28696}} - \frac{1}{2}\, i\, \partial_{\mathrm{mu\$1}}\left(\bar{q}_{\mathrm{r\$28694},\mathrm{Colour\$1}}\right) \cdot q_{\mathrm{r\$28681},\mathrm{Colour\$1}}\, \gamma^{\mathrm{mu\$1}} \cdot P_{+\mathrm{r\$28694},\mathrm{r\$28681}} + \frac{1}{2}\, i\, \bar{q}_{\mathrm{r\$28697},\mathrm{Colour\$1}} \cdot \partial_{\mathrm{mu\$1}}\left(q_{\mathrm{r\$28684},\mathrm{Colour\$1}}\right) \gamma^{\mathrm{mu\$1}} \cdot P_{+\mathrm{r\$28697},\mathrm{r\$28684}}$$

Model database

**We encourage model builders writing
order to make them useful to a comm
FeynRules model database, please se**

- claude.duhr@durham.ac.uk
- neil@hep.wisc.edu
- fuks@cern.ch

Available models

Standard Model
Simple extensions of the SM (9)
Supersymmetric Models (4)
Extra-dimensional Models (4)
Strongly coupled and effective field theories (4)
Miscellaneous (0)

Model	Contact
MSSM	B. Fuks
NMSSM	B. Fuks
RPV-MSSM	B. Fuks
R-MSSM	B. Fuks

Summary

- FeynRules allows to use the superfield formalism for supersymmetric theories.
    - ➡ expands the superfields in the Grassmann variables and integrates them out.
    - ➡ Weyl fermions are transformed into 4-component spinors.
    - ➡ auxiliary fields are integrated out.
- Upshot: You implement a SUSY model using compact superspace notation, and FeynRules takes care of the signs!

Interlude:
SLHA files

SLHA

- For SUSY theories there is a standardized format for numerical values of parameter sets
  - ➡ SUSY Les Houches Accord (SLHA)
  - ➡ Allows for easy communication between different codes, e.g., spectrum generators and MC codes.
- Many ME generators have adopted/extended the SLHA format as the default format for numerical input parameters (=external parameters).
  - ➡ Allows to change the numerical input parameters at run-time.
  - ➡ FeynRules can read/write SLHA files.

SLHA

- Parameters are grouped into blocks.

```
Block SMINPUTS      # Standard Model inputs
    1       1.32506980E+02   # alpha_em(MZ)(-1) SM MSbar
    2       1.16639000E-05   # G_Fermi
    3       1.18000000E-01   # alpha_s(MZ) SM MSbar
    4       9.11880000E+01   # Z mass (as input parameter)
```

- New blocks can be added by the user
  - ➡ Numerical values must be real.
  - ➡ SMINPUTS should always be defined.
  - ➡ Some blocks are mandatory.

SLHA

- The masses and widths of all massive particles are defined in the MASS and DECAY blocks:

```
Block MASS     #  Mass spectrum (kinematic masses)
#      PDG       Mass
       24     8.04190000E+01   # W       mass
       25     1.25000000E+02   # H       mass

DECAY       25    5.75308848E-03   # H   width
#       BR          NDA       ID1      ID2
    8.27451012E-02   2         4       -4   # BR( H -> c  cbar  )
    7.17809696E-01   2         5       -5   # BR( H -> b  bbar  )
```

- NB: Parameters in SLHA file must not be independent.
  - ➡ E.g., widths are always dependent.
  - ➡ Must however make sure that the parameters are consistent!

Mass diagonalization with ASperGe

Mass matrices

- In general, the mass matrices appearing inside the Lagrangian are not diagonal, but they need to be diagonalized.
- For simple models, this can be done analytically (cf. tutorial).
- For complicated models, this needs to be done numerically.

Mass matrices

- In general, the mass matrices appearing inside the Lagrangian are not diagonal, but they need to be diagonalized.
- For simple models, this can be done analytically (cf. tutorial).
- For complicated models, this needs to be done numerically.
- Example:

$$\frac{1}{2}\partial_\mu \phi_i \partial^\mu \phi_i - \frac{1}{2}\phi_i \mathcal{M}_{ij}\phi_j$$

$$\Phi_i = U_{ij}\phi_j \qquad\qquad U^T \mathcal{M} U = D$$

➡ For small matrices this can be done analytically.

➡ For ‘large’ matrices, need numerical diagonalization.

ASperGe

- For the MSSM, there are many tools that allow to perform this task.
  - ➡ General idea: obtain SLHA file that contains the whole mass spectrum, + mixing matrices.

ASperGe

- For the MSSM, there are many tools that allow to perform this task.
  - ➡ General idea: obtain SLHA file that contains the whole mass spectrum, + mixing matrices.
- Something similar can be done for generic models!
  - ➡ ASperGe = Automatic Spectrum Generator

    [Alloul, de Causmaecker, d'Hondt, Fuks, Rausch de Traubenberg]
- ASperGe has two parts:
  - ➡ Mathematica (embedded into FR): extract mass matrices form Lagrangian.
  - ➡ C++: Numerical diagonalization of the matrices.

ASperGe

[Alloul, de Causmaecker, d'Hondt, Fuks, Rausch de Traubenberg]

- Mixing relations can be specified in the model file:

$$\begin{pmatrix} A_\mu \\ Z_\mu \end{pmatrix} = U_w \begin{pmatrix} B_\mu \\ W^3_\mu \end{pmatrix}$$

```
Mix["AZmix"] == {
  MassBasis     -> {A, Z},
  GaugeBasis    -> {B, Wi[3]},
  MixingMatrix -> UW,
  BlockName     -> WEAKMIX
}
```

```
GaugeBasis = MixingMatrix . MassBasis
```

- Mass matrix can be extracted from Lagrangian:

```
ComputeMassMatrix[ L ]
```

ASperGe

[Alloul, de Causmaecker, d'Hondt, Fuks, Rausch de Traubenberg]

- ASperGe generates from FeynRules a C++ code that allows to diagonalize the mass matrices

```
WriteASperGe[ L ]
```

- Once generated, the C++ code can be used standalone!

  ➡ No need to return to Mathematica every time!

  ➡ Input: SLHA file without masses and mixing.

  ➡ Output: SLHA file with masses and mixing.

Computing widths and branching ratios

Widths

- Similar to the masses, the widths need to be given as numerical inputs in the SLHA input files.
- Some MC codes need also the branching ratios in order to decay the particles.

```
DECAY     25    5.75308848E-03  # H  width
#     BR          NDA      ID1      ID2
   8.27451012E-02   2         4       -4  # BR( H -> c  cbar  )
   7.17809696E-01   2         5       -5  # BR( H -> b  bbar  )
```

- However, widths and branching ratios are not independent parameters, so their value cannot be chosen freely.

  ➡ User needs to compute them separately.

Widths

- Solution 1: Use ME generators to compute all the widths, and then update the parameter input file.
  - ➡ Some codes even do this on the fly!
- Downside: This procedure must be repeated for every parameter set!

Widths

- Solution 1: Use ME generators to compute all the widths, and then update the parameter input file.
  - ➡ Some codes even do this on the fly!
- Downside: This procedure must be repeated for every parameter set!
- Solution 2: In many cases the two-body decays are dominant.
  - ➡ Two-body decays are easy to compute analytically.

$$\Gamma_{1\to 2} = \frac{1}{2m}\int d\Phi_2\,|\mathcal{M}_{1\to 2}|^2 = \frac{1}{2m}|\mathcal{M}_{1\to 2}|^2\,\text{Vol(phase space)}$$

- ➡ Two-body decays are constants!

The decay module

$$\Gamma_{1\to 2} = \frac{1}{2m}\int d\Phi_2\, |\mathcal{M}_{1\to 2}|^2 = \frac{1}{2m}|\mathcal{M}_{1\to 2}|^2\,\mathrm{Vol(phase\ space)}$$

The decay module

$$\Gamma_{1\to 2} = \frac{1}{2m}\int d\Phi_2 \, |\mathcal{M}_{1\to 2}|^2 = \frac{1}{2m}|\mathcal{M}_{1\to 2}|^2 \, \mathrm{Vol(phase\ space)}$$

- In FeynRules, we have
  - ➡ all the three-point vertices,
  - ➡ a high-level computer algebra system.
- That's all we need to get the two-body decays!

The decay module

$$\Gamma_{1\to 2} = \frac{1}{2m}\int d\Phi_2\, |\mathcal{M}_{1\to 2}|^2 = \frac{1}{2m}|\mathcal{M}_{1\to 2}|^2\, \mathrm{Vol(phase\ space)}$$

- In FeynRules, we have
  - ➡ all the three-point vertices,
  - ➡ a high-level computer algebra system.
- That's all we need to get the two-body decays!

```
vertices = FeynmanRules[ L ];

decays = ComputeDecays[ vertices ];
```

- All two-body partial widths are computed analytically, and stored in some internal format.

The decay module

- The partial widths can be output in the UFO format, and be used when generating a process.

The decay module

- The partial widths can be output in the UFO format, and be used when generating a process.

```
Decay_H = Decay(name = 'Decay_H',
                particle = P.H,
                partial_widths = {
                         (P.b,P.b__tilde__):'3*MH**2*yb**2',
                         (P.ta__minus__,P.ta__plus__):'MH**2*ytau**2',
                         (P.c,P.c__tilde__):'3*MH**2*yc**2',
                         (P.t,P.t__tilde__):'3*MH**2*yt**2'})
```

The decay module

- The partial widths can be output in the UFO format, and be used when generating a process.

```
Decay_H = Decay(name = 'Decay_H',
          particle = P.H,
          partial_widths = {
                   (P.b,P.b__tilde__):'3*MH**2*yb**2',
                   (P.ta__minus__,P.ta__plus__):'MH**2*ytau**2',
                   (P.c,P.c__tilde__):'3*MH**2*yc**2',
                   (P.t,P.t__tilde__):'3*MH**2*yt**2'})
```

- NB: All possible analytic formulas are output, independently whether they are kinematically allowed!

The decay module

- The partial widths can be output in the UFO format, and be used when generating a process.

```
Decay_H = Decay(name = 'Decay_H',
          particle = P.H,
          partial_widths = {
                   (P.b,P.b__tilde__):'3*MH**2*yb**2',
                   (P.ta__minus__,P.ta__plus__):'MH**2*ytau**2',
                   (P.c,P.c__tilde__):'3*MH**2*yc**2',
                   (P.t,P.t__tilde__):'3*MH**2*yt**2'})
```

- NB: All possible analytic formulas are output, independently whether they are kinematically allowed!
  - ➡ Some channels might be open for some benchmark scenarios but not for other
  - ➡ Channels depend on spectrum.

Towards NLO

Towards NLO

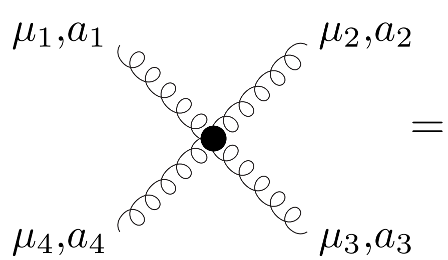
- At the current stage FeynRules can
  - ➡ compute Feynman rules.
  - ➡ compute two-body partial widths.
  - ➡ extract and diagonalize mass matrices (vis ASperGe).
- This is enough to cover large parts of BSM phenomenology at tree-level.

Towards NLO

- At the current stage FeynRules can
  - ➡ compute Feynman rules.
  - ➡ compute two-body partial widths.
  - ➡ extract and diagonalize mass matrices (vis ASperGe).
- This is enough to cover large parts of BSM phenomenology at tree-level.
- For next-to-leading order (NLO), tree-level Feynman rules are not enough!
  - ➡ UV counterterms.
  - ➡ 'R2 vertices' (depending on the NLO ME generator).

R2 vertices

- All the automatized NLO codes are based, in one way or another, on some unitary-based approach.
- Unitarity, however, does not provide everything, but misses the rational pieces (without cuts).
- Some can be obtained, others (R2) need a different approach.
- R2 vertices can be obtained via effective tree-level Feynman rules.



$$= -\frac{ig^4 N_{col}}{96\pi^2} \sum_{P(234)} \Bigg\{ \Bigg[\frac{\delta_{a_1a_2}\delta_{a_3a_4} + \delta_{a_1a_3}\delta_{a_4a_2} + \delta_{a_1a_4}\delta_{a_2a_3}}{N_{col}}$$

$$+ 4\, Tr(t^{a_1}t^{a_3}t^{a_2}t^{a_4} + t^{a_1}t^{a_4}t^{a_2}t^{a_3})\,(3 + \lambda_{HV})$$

$$- Tr(\{t^{a_1}t^{a_2}\}\{t^{a_3}t^{a_4}\})\,(5 + 2\lambda_{HV}) \Bigg] g_{\mu_1\mu_2} g_{\mu_3\mu_4}$$

$$+12 \frac{N_f}{N_{col}} Tr(t^{a_1}t^{a_2}t^{a_3}t^{a_4}) \left(\frac{5}{3} g_{\mu_1\mu_3} g_{\mu_2\mu_4} - g_{\mu_1\mu_2} g_{\mu_3\mu_4} - g_{\mu_2\mu_3} g_{\mu_1\mu_4} \right) \Bigg\}$$

[Draggiotis, Garzelli, Papadopoulos, Pittau;
Garzelli, Malamos, Pittau]

What are the R_2 rational terms?

$$\bar{A}(\bar{q}) = \frac{1}{(2\pi)^4} \int d^d\bar{q} \frac{\bar{N}(\bar{q})}{\bar{D}_0 \bar{D}_1 \ldots \bar{D}_{m-1}}, \qquad \bar{D}_i = (\bar{q} + p_i)^2 - m_i^2$$

$$\bar{N}(\bar{q}) = N(q) + \widetilde{N}(\tilde{q}, q, \epsilon)$$

where $\bar{X}$ lives in d dimension, X in 4, $\widetilde{X}$ in ϵ.

R_2 definition

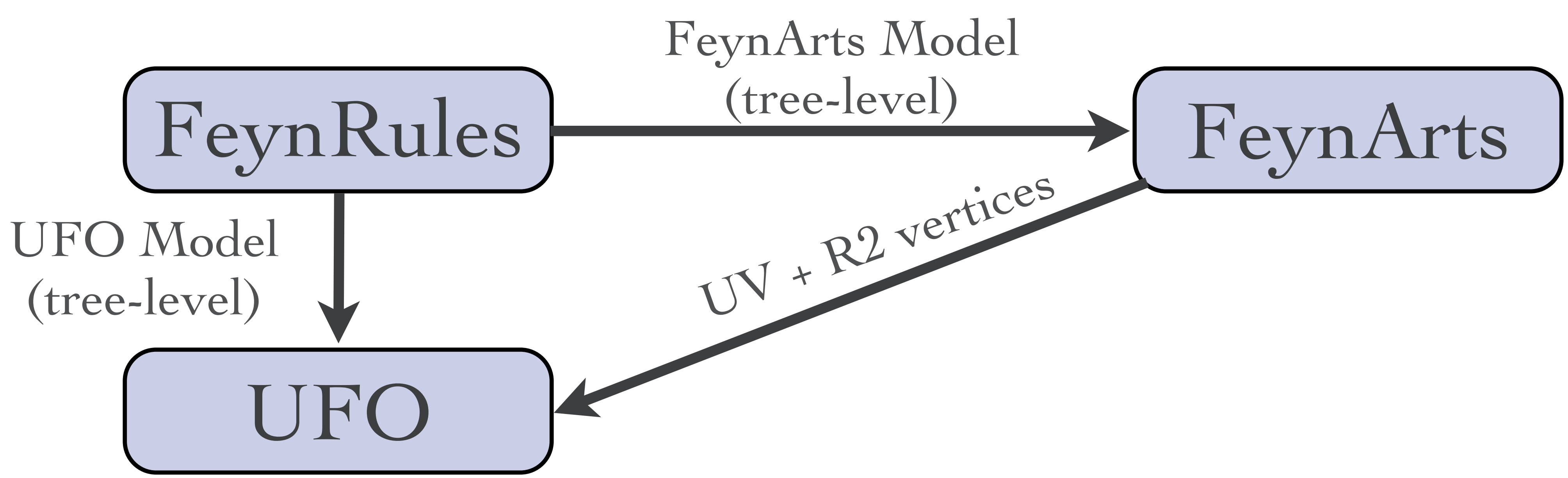
$$R_2 \equiv \frac{1}{(2\pi)^4} \int d^d\bar{q} \frac{\widetilde{N}(\tilde{q}, q, \epsilon)}{\bar{D}_0 \bar{D}_1 \ldots \bar{D}_{m-1}}$$

Finite (< 4 legs) set of vertices computed once for all!

FR@NLO

- **Upshot:** Additional information needed for NLO can be obtained specific loop integrals.
  - ➡ can be computed once and for all for every model.

- **Idea:** Use the FeynRules interfaced to FeynArts to generate and compute these loop integrals:



FR@NLO

- Automated BSM@NLO will become possible (at least for large classes of models)!

FR@NLO

- Automated BSM@NLO will become possible (at least for large classes of models)!

- Status: Email by V. Hirschi yesterday: agreement in MG5 with FR@NLO SM model for (NLO QCD only)

d d~ > w+ w- g	g g > h h t t~	g g > h t t~	u d~ > h t b~	u u~ > w+ w- b b~
u u~ > d d~	d g > d g	d~ u~ > d~ u~	g u~ > g u~	g g > d d~
g g > t t~	g g > g g	d~ d > g a	u~ u > g z	e+ e- > d d~
d u~ > w- g	d~ d > a g g	d~ d > z g g	d~ d > z z g	d~ u > w+ g g
g g > a t t~	g g > g t t~	g g > w- d~ u	g g > z t t~	s s~ > a z g
u u~ > w+ w- z	u u~ > z z z			

➡ We are on the right track!

Galileo

Galileo

[Christensen, Setzer, Stefanus]

- At this point, one of the biggest bottlenecks for implementing a model into FeynRules is writing the model file.
- In principle:
  - ➡ Fields / particle content.
  - ➡ Symmetries.
  - ➡ Numerical value of the input parameters.
- In practise: This can be quite complicated still (although it is already much easier then before).

Galileo

[Christensen, Setzer, Stefanus]

- At this point, one of the biggest bottlenecks for implementing a model into FeynRules is writing the model file.
- In principle:
  - ➡ Fields / particle content.
  - ➡ Symmetries.
  - ➡ Numerical value of the input parameters.
- In practise: This can be quite complicated still (although it is already much easier then before).
- Solution: A code that turns symmetries and fields into a Lagrangian!

Galileo

[Christensen, Setzer, Stefanus]

- Idea:
    - ➡ User specifies fields and symmetry groups.
    - ➡ Code generates all Lagrangian term (up to a certain dimension.
    - ➡ Output of FeynRules model file.
- Code not ready yet, but I can give you some snapshots...

Galileo

[Christensen, Setzer, Stefanus]

I | G | F | Config

	U(1)	+G
e_L	-1	
e_R	-1	
+F		

Lagrangian

$-\frac{1}{4}A_{\mu\nu}A^{\mu\nu}+i\overline{e_L}\gamma_\mu D^\mu e_L+$
$i\overline{e_R}\gamma_\mu D^\mu e_R$
$m_0\overline{e_R}e_L+$
$m_0^*\overline{e_L}e_R$

U(1)

[Christensen]

Galileo

[Christensen, Setzer, Stefanus]

I | G | F | Config

	SU(3)	SU(2)	U(1)	+G
Q_L	3	2	$\frac{1}{6}$	
u_R	3	1	$\frac{2}{3}$	
d_R	3	1	$\frac{-1}{3}$	
L_L	1	2	$\frac{-1}{2}$	
e_R	1	1	-1	
Φ	1	2	$\frac{1}{2}$	
+F				

Lagrangian

$-\frac{1}{4}G_{\mu\nu}G^{\mu\nu}+\theta_0 G_{\mu\nu}\widetilde{G}^{\mu\nu}+-\frac{1}{4}W_{\mu\nu}W^{\mu\nu}+\theta_1 W_{\mu\nu}\widetilde{W}^{\mu\nu}+-\frac{1}{4}B_{\mu\nu}B^{\mu\nu}+i\overline{Q_L}\gamma_\mu D^\mu Q_L+i\overline{u_R}\gamma_\mu D^\mu u_R+$

$i\overline{d_R}\gamma_\mu D^\mu d_R+i\overline{L_L}\gamma_\mu D^\mu L_L+i\overline{e_R}\gamma_\mu D^\mu e_R+D_\mu\Phi D^\mu\Phi^*$

$\mu_{r2}\Phi\Phi^*$

$\lambda_{r3}\Phi\Phi\Phi^*\Phi^*+$

$\lambda_{r4}\Phi\Phi\Phi^*\Phi^*$

$y_5\Phi\overline{u_R}Q_L+y_5^*\Phi^*\overline{Q_L}u_R+y_6\Phi^*\overline{d_R}Q_L+y_6^*\Phi\overline{Q_L}d_R+y_7\Phi^*\overline{e_R}L_L+$

$y_7^*\Phi\overline{L_L}e_R$

[Christensen]

Galileo

[Christensen, Setzer, Stefanus]

I | G | F | Config

	SU(3)	SU(2)	U(1)	+G
Q_L	3	2	$\frac{1}{6}$	
u_R	3	1	$\frac{2}{3}$	
d_R	3	1	$\frac{-1}{3}$	
L_L	1	2	$\frac{-1}{2}$	
e_R	1	1	-1	
Φ	1	2	$\frac{1}{2}$	
+F				

Lagrangian

$-\frac{1}{4} G_{\mu\nu} G^{\mu\nu} + \theta_0 G_{\mu\nu} \widetilde{G}^{\mu\nu} + -\frac{1}{4} W_{\mu\nu} W^{\mu\nu} + \theta_1 W_{\mu\nu} \widetilde{W}^{\mu\nu} + -\frac{1}{4} B_{\mu\nu} B^{\mu\nu} + i \overline{Q_L} \gamma_\mu D^\mu Q_L + i \overline{u_R} \gamma_\mu D^\mu u_R +$
$i \overline{d_R} \gamma_\mu D^\mu d_R + i \overline{L_L} \gamma_\mu D^\mu L_L + i \overline{e_R} \gamma_\mu D^\mu e_R + D_\mu \Phi D^\mu \Phi^*$

$\mu_{r2} \Phi \Phi^*$

$\lambda_{r3} \Phi \Phi \Phi^* \Phi^* +$
$\lambda_{r4} \Phi \Phi \Phi^* \Phi^*$

$y_5 \Phi \overline{u_R} Q_L + y_5^* \Phi^* \overline{Q_L} u_R + y_6 \Phi^* \overline{d_R} Q_L + y_6^* \Phi \overline{Q_L} d_R + y_7 \Phi^* \overline{e_R} L_L + y_7^* \Phi \overline{L_L} e_R + y_8 \Phi \Phi \overline{L_L^c} L_L + y_8^* \Phi^* \Phi^* \overline{L_L} L_L^c +$
$y_9 \Phi \Phi \overline{L_L^c} L_L + y_9^* \Phi^* \Phi^* \overline{L_L} L_L^c$

[Christensen]

Galileo

[Christensen, Setzer, Stefanus]

- Current status:
  - ➡ Supports any semi-simple compact Lie algebra (symmetry).
  - ➡ Supports fields of spin 0, 1/2, 0 + superfields.
  - ➡ Automatically generates the Lagrangian.
  - ➡ GUI wrapper.

- To do:
  - ➡ Symmetry breaking.
  - ➡ Mass diagonalization + rotation physical basis.
  - ➡ Export model to FeynRules.

Summary

- FeynRules:
  - ➡ Computes Feynman rules from a Lagrangian for large classes of models.
  - ➡ Interfaces to many ME generators.
  - ➡ Superfield formalism.
  - ➡ Mass diagonalization + ASperGe.
  - ➡ Two-body decays.

- Try it out on your favorite model!