

New Developments in FEYNRULES

A. Alloul

FEYNRULES: AA, N.D. Christensen, C. Degrande, C. Duhr, B. Fuks

IPHC - UNIVERSITÉ DE STRASBOURG

May 16-21, 2013

- 1 Basic features of FEYNRULES
- 2 New developments in version 1.8
- 3 Conclusion

Need for automatization

✱ Standard Model $\equiv$ effective theory.

- ⇒ **Hierarchy** problem
- ⇒ **Neutrino** oscillations
- ⇒ **Dark** matter and energy
- ⇒ ...

✱ LHC is running

- ⇒ Deviations from Standard Model?
- ⇒ New signatures?
- ⇒ Properties of the Higgs-like boson
- ⇒ ...



Need for theoretical predictions

✱ Develop **new** models

- ⇒ Write Lagrangian
- ⇒ Compute several quantities (decay widths, mass and mixing matrices ...)

✱ **Implement** in Monte Carlo (MC) tools

- ⇒ Many available tools (CALCHEP, MADGRAPH, PYTHIA, SHERPA...)
- ⇒ Every tool has its **dedicated** file format
- ⇒ Every tool has its **advantages / disadvantages**

Starting from your **model**, FEYNRULES provides all the **necessary** ingredients to generate the **model file** for you beloved MC tool(s)

How it works

- ✧ **Mathematica** package
 - ⇒ Christensen, Duhr (CPC'09), AA, Christensen, Degrande, Duhr, Fuks (prep)
 - ⇒ Version 1.6 and 1.8 *beta* downloadable from:  `feynrules.irmp.ucl.ac.be`
- ✧ Derives **Feynman rules** from a Lagrangian.

- ✧ **Exports** them to MC tools through interfaces.
 - ⇒ Available interfaces to
 - CALCHEP ■ FEYNARTS ■ MADGRAPH ■ SHERPA ■ UFO ■ WHIZARD

How it works

- ✱ **Mathematica** package
 - ⇒ Christensen, Duhr (CPC'09), AA, Christensen, Degrande, Duhr, Fuks (prep)
 - ⇒ Version 1.6 and 1.8 *beta* downloadable from:  feynrules.irmp.ucl.ac.be
- ✱ Derives **Feynman rules** from a Lagrangian.
 - ⇒ Implement your model
 - Gauge group and parameter declarations
 - ⇒ Lorentz and gauge local invariance required.
 - Field definitions
 - ⇒ scalar, fermion, vector, tensor, ghost, superfield, spin-3/2.
 - ⇒ Existing database (many models already **available**)
 - SM & simple extensions
 - Supersymmetry
 - Extra-dimensions
 - Strongly coupled and effective theories
- ✱ **Exports** them to MC tools through interfaces.
 - ⇒ Available interfaces to
 - CALCHEP ■ FEYNARTS ■ MADGRAPH ■ SHERPA ■ UFO ■ WHIZARD

How it works: the QCD Lagrangian

From your idea to the Monte Carlo events:

✱ Define the model

Gauge symmetries

Particles

Parameters

Lagrangian

```
M$GaugeGroups = {  
  SU3C == {  
    Abelian -> False,  
    CouplingConstant -> gs,  
    GaugeBoson -> G,  
    StructureConstant -> f,  
    Representations -> {T,Colour},  
    SymmetricTensor -> dSUN }}
```

How it works: the QCD Lagrangian

From your idea to the Monte Carlo events:

✱ Define the model

Gauge symmetries

Particles

Parameters

Lagrangian

```
M$ClassesDescription == {
  V[1] == {  ClassName -> G,
    SelfConjugate -> True,
    Indices -> {Index[Gluon]},
    Mass -> 0,
    Width -> 0,
    ParticleName -> "g",
    PDG -> 21,
    PropagatorLabel -> "G",
    PropagatorType -> C,
    PropagatorArrow -> None,
    FullName -> "G"},
  F[1] == {  ClassName -> q,
    ClassMembers -> {d,u,s,c,b,t},
    FlavorIndex -> Flavour,
    SelfConjugate -> False,
    Indices -> {Index[Flavour], Index[Colour]},
    Mass -> {MQ, MD, MU, MS, MC, MB, MT},
    Width -> {WQ, 0, 0, 0, 0, 0, WT },
    ParticleName -> {"d", "u", "s", "c", "b", "t"},
    AntiParticleName -> {"d~", "u~", "s~", "c~", "b~", "t~"},
    PDG -> {1, 2, 3, 4, 5, 6},
    PropagatorLabel -> {"d", "u", "s", "c", "b", "t"},
    PropagatorType -> Straight,
    PropagatorArrow -> Forward}}
```

Other classes are available:

- ⇒ Weyl fermions **W[]**
- ⇒ Scalars **S[]**
- ⇒ Chiral superfields **CSF[]**
- ⇒ Vector superfields **VSF[]**

How it works: the QCD Lagrangian

From your idea to the Monte Carlo events:

✱ Define the model

Gauge symmetries

Particles

Parameters

Lagrangian

$$g_s = \sqrt{4\pi\alpha_s}$$



```
M$Parameters == {
  aS == {
    ParameterType -> External,
    BlockName -> SMINPUTS,
    OrderBlock -> 3,
    Value -> 0.1184,
    InteractionOrder -> {QCD,2},
    TeX -> Subscript[["Alpha"],s],
    Description -> "Strong coupling
constant at the Z pole"
  },
  gs == {
    ParameterType -> Internal,
    Value -> Sqrt[4*Pi*aS],
    InteractionOrder -> {QCD,1},
    TeX -> Subscript[g,s],
    ParameterName -> G,
    Description -> "Strong coupling
constant" }}}
```


How it works: the QCD Lagrangian

From your idea to the Monte Carlo events:

✱ Define the model

Gauge symmetries

Particles

Parameters

Lagrangian

$$\mathcal{L}_{QCD} = -\frac{1}{4} G_{\mu\nu}^a G_a^{\mu\nu} + \sum_f \left[\bar{q}_f (i\not{D} + g_s \not{G}^a T^a - m_f) q_f \right]$$

```
LQCD :=
-1/4 FS[G,mu,nu,aa] FS[G,mu,nu,aa] +
I*qbar.Ga[mu].DC[q,mu] - MQ[f]*qbar[s,f,c].q[s,f,c];
```

How it works: the QCD Lagrangian

From your idea to the Monte Carlo events:

- * Define the model ✓
- * Load the model

```
<<FeynRules`;  
LoadModel["modelfile.fr"];
```

How it works: the QCD Lagrangian

From your idea to the Monte Carlo events:

- * Define the model ✓
- * Load the model ✓
- * Several commands available

⇒ General commands

- Hermitian conjugate: `HC[q]`
- Flavor expansion: `ExpandIndices[LQCD, FlavorExpand -> Generation]`
- Check hermiticity `CheckHermiticity[LQCD]`
- Feynman Rules: `FeynmanRules[LQCD]`
- ...

How it works: the QCD Lagrangian

From your idea to the Monte Carlo events:

- * Define the model ✓
- * Load the model ✓
- * Several commands available
 - ⇒ General commands ✓
 - ⇒ **Supersymmetry** dedicated commands:
 - Get components of a superfield `SF2Components[superfield]`
 - Kinetic terms of a vector superfield `VSFKineticTerms[superfield]`
 - ...

How it works: the QCD Lagrangian

From your idea to the Monte Carlo events:

- * Define the model ✓
- * Load the model ✓
- * Several commands available
 - ⇒ General commands ✓
 - ⇒ **Supersymmetry** dedicated commands ✓
- * Interfaces
 - ⇒ UFO: `WriteUFO[LQCD]`
 - ⇒ FEYNARTS: `WriteFeynArtsOutput[LQCD]`
 - ⇒ CALCHEP: `WriteCHOutput[LQCD]`
 - ⇒ WHIZARD: `WriteW0Output[LQCD]`
 - ⇒ ...

How it works: the QCD Lagrangian

From your idea to the Monte Carlo events:

- * Define the model ✓
- * Load the model ✓
- * Several commands available
 - ⇒ General commands ✓
 - ⇒ **Supersymmetry** dedicated commands ✓
- * The UNIVERSAL FEYNRULES OUTPUT interface:
 - ⇒ A python module to be linked to **any** code
 - ⇒ **All** model information is included
 - ⇒ **No restriction** on the vertices (e.g., Lorentz and color structure)
 - ⇒ Used by **MADGRAPH 5**, **MADANALYSIS 5**, **GoSAM** and **HERWIG++**

How it works: the QCD Lagrangian

From your idea to the Monte Carlo events:

- * Define the model ✓
- * Load the model ✓
- * Several commands available
 - ⇒ General commands ✓
 - ⇒ **Supersymmetry** dedicated commands ✓
- * Interfaces ✓

For more details see:

- 📖 **FeynRules - Feynman rules made easy**, Christensen, Duhr, CPC'09
- 📖 **A superspace module for the FeynRules package**, Duhr, Fuks, CPC'11
- 📖 **A Comprehensive approach to new physics simulations**, Christensen, de Aquino, Degrande, Duhr, Fuks, Herquet, Maltoni, Schumann, E.P.J C71 '11
- 📖 **UFO - The Universal FeynRules Output**, Degrande, Duhr, Fuks, Grellscheid, Mattelaer, Reiter, CPC '12
- 📖 **Introducing an interface between WHIZARD and FeynRules**, Christensen, Duhr, Fuks, Reuter, Speckner, E.P.J. C72 '12
- 📖 **Beyond the Minimal Supersymmetric Standard Model: from theory to phenomenology**, Fuks, Int.J.Mod.Phys. A27 '12

- 1 Basic features of FEYNRULES
- 2 New developments in version 1.8**
- 3 Conclusion

Spin-3/2 Rarita-Schwinger fields

* Spin-3/2 fields

- ⇒ Described first by Rarita & Schwinger in 1941.
- ⇒ Appears in some BSM theories (Gravitino in SUSY, e.g)

* In FEYNRULES $\left\{ \begin{array}{l} \text{Lagrangian's dimensionality } \textbf{unrestricted.} \\ \textbf{Complete} \text{ superspace module.} \end{array} \right\}$ **Everything is there !**

* Two **new classes** for field declaration

- ⇒ For two-component fermions `RW`
- ⇒ For four-component fermions `R`

* Goldstino / Gravitino

- ⇒ Spontaneous supersymmetry breaking $\Rightarrow$ massless fermion.
- ⇒ Its interactions given by **supercurrent** $\epsilon \cdot J^\mu + \bar{\epsilon} \cdot \bar{J}^\mu = \frac{\partial \mathcal{L}}{\partial (\partial_\mu X)} \delta_\epsilon X - K^\mu$
- ⇒ FEYNRULES dedicated command `Supercurrent[1c,1v,1w,sp,mu]`
 1c: Chiral Lagrangian, 1v: Vector Lagrangian, 1w: Superpotential, sp,mu: Spin & Lorentz indices.

* UFO and CALCHEP interfaces **adapted**

 **Simulating spin-3/2 particle production at colliders,**
Christensen, de Aquino, Deutschmann, Duhr, Fuks, Garcia-Cely, Mattelaer, Mawatari, Oehl, Takaesu,
In preparation

Beta version available

Decays package

* Tree-Level two-body decay widths

- ⇒ Are now **automatically** computed for any model

```
verts = FeynmanRules[LQCD];  
vertsexp = FlavorExpansion[verts];  
results = ComputeWidths[vertsexp];
```

- ⇒ Everything is **analytical**

* Phase-space closed channels included

- ⇒ No information on the (numerical values of the) spectrum at this level
- ⇒ Benchmark scenario **independent**

* Information passed to the UFO `WriteUFO[LQCD, AddDecays -> True]`

- ⇒ **Flexible**: closed formulas (NLO, n-body, BSM) can be included.

* MADGRAPH 5¹

- ⇒ **checks** numerically open channels.
- ⇒ compute **automatically** n-body decay widths

 **Computing decay rates for new physics theories with FEYNRULES and MADGRAPH,**
Duhr, Fuks, Mattelaer, Oeztuerk,
In preparation

Beta version available

¹ CALCHEP calculates also **automatically** widths

Spectrum generator

Problem

- ✱ After generating the model file for your MC-generator, you need
 - ⇒ To calculate the mass matrices: Lengthy and error-prone

Spectrum generator

Problem

- * After generating the model file for your MC-generator, you need
 - ⇒ To calculate the mass matrices: Lengthy and error-prone
 - ⇒ Diagonalize them: No analytical solution for matrices bigger than 4×4

Spectrum generator

Problem

- ✱ After generating the model file for your MC-generator, you need
 - ⇒ To calculate the mass matrices: Lengthy and error-prone
 - ⇒ Diagonalize them: No analytical solution for matrices bigger than 4×4
 - ⇒ Update the model file: Lengthy (several scenarios) and error-prone

Spectrum generator

Problem

- * After generating the model file for your MC-generator, you need
 - ⇒ To calculate the mass matrices: Lengthy and error-prone
 - ⇒ Diagonalize them: No analytical solution for matrices bigger than 4×4
 - ⇒ Update the model file: Lengthy (several scenarios) and error-prone

Solution

- * FEYNRULES will
 - ⇒ extract automatically analytical expressions for mass matrices
 - ⇒ generate automatically a numerical code for the diagonalization
- * The numerical code:
 - ⇒ produces a SLHA-like output

Mass matrices generation with FEYNRULES

Model file simplified

* Field mixing declaration

```
M$MixingsDescription = {  
  Mix["1u"] == {MassBasis -> {A, Z},  
    GaugeBasis -> {B, Wi[3]},  
    MixingMatrix -> UG,  
    BlockName -> WEAKMIX}}
```

⇒ Various options for different fields

Mass matrices generation with FEYNRULES

Model file simplified

- * Field mixing declaration `M$MixingDescription == {Mix["Id"] == {options }}`
- * Vacuum expectation values: `M$vevs == {{phi[2],vev}}`

Mass matrices generation with FEYNRULES

Model file simplified

- * Field mixing declaration `M$MixingDescription == {Mix["Id"] == {options }}`
- * Vacuum expectation values: `M$vevs`
- * Mass matrices, mixing matrices declared **automatically as complex**

Mass matrices generation with FEYNRULES

Model file simplified

- * Field mixing declaration `M$MixingDescription == {Mix["Id"] == {options }}`
- * Vacuum expectation values: `M$vevs`
- * Mass matrices, mixing matrices declared **automatically as complex**

Commands available in FEYNRULES

- * Compute mass matrices: `ComputeMassMatrix[lagr]`
 - ⇒ Possible to only compute a set of mixings with option `Mix -> {"Id"}`
- * Access specific informations with the Id
 - ⇒ `MassMatrix["Id"], GaugeBasis["Id"], MassBasis["Id"]`
- * Summary of all results
 - ⇒ `MixingSummary[]`

Mass matrices generation with FEYNRULES

Model file simplified

- * Field mixing declaration `M$MixingDescription == {Mix["Id"] == {options }}`
- * Vacuum expectation values: `M$vevs`
- * Mass matrices, mixing matrices declared **automatically as complex**

Commands available in FEYNRULES

- * Compute mass matrices: `ComputeMassMatrix[lagr]`
 - ⇒ Possible to only compute a set of mixings with option `Mix -> {"Id"}`
- * Access specific informations with the Id
 - ⇒ `MassMatrix["Id"], GaugeBasis["Id"], MassBasis["Id"]`
- * Summary of all results
 - ⇒ `MixingSummary[]`

Towards a numerical code

- * Write the C++ package: `WriteASperGe[Lagrangian,Mix -> {"Id"}]`
- * Run the package and **import** results: `RunASperGe[]`

ASperGe: Automated Spectrum Generation

✱ ASperGe is a C++ package

- ⇒ Generated **automatically** by FEYNRULES
directory `ModelName_MD`
- ⇒ Contains all the necessary routines to
 - Define the mass matrices
 - **Diagonalize** them
 - Generate an **SLHA**-compliant output file
- ⇒ **Standalone** package
 - Only need GSL and a C++ compiler

```
ACAT2013$ cd MyTestModel
ACAT2013$ make
ACAT2013$ ./ASperGe input.dat output.dat
```

```
Block MASS
# pdg code mass particle
22 0.000000e+00 # A
23 9.180401e+01 # Z
```

 Automated mass spectrum generation for new physics,
AA, D'Hondt, De Causmaecker, Fuks, Rausch de Traubenberg, E.P.J C73 '13

NLO computations

Motivations

- * Leading Order (LO) calculations good for prospectings
 - ⇒ Need for more precise theoretical predictions.
 - ⇒ Next to Leading Order (NLO) is the next step

$$\text{NLO} = \underbrace{\text{LO} + R_2 \text{ vertices} + \text{UV counterterms}} + \text{real emissions} + \text{virtual emissions}.$$

Can be evaluated **once for all** from a Lagrangian

Solution

- * In FEYNRULES

- ⇒ Tree-Level vertices
- ⇒ Lagrangian is there

- * In FEYNARTS

- ⇒ Powerful tool for computing vertices
- ⇒ Interfaced with FEYNRULES

Automatic UV counterterms

Automatic R_2 vertices



UFO@NLO

NLO computations

Motivations

- * Leading Order (LO) calculations good for prospections
 - ⇒ Need for more precise theoretical predictions.
 - ⇒ Next to Leading Order (NLO) is the next step

$$\text{NLO} = \text{LO} + R_2 \text{ vertices} + \text{UV counterterms} + \text{real emissions} + \text{virtual emissions}.$$

Can be evaluated **once for all** from a Lagrangian

Solution

- ✱ In FEYNRULES

- ➡ Tree-Level vertices
- ➡ Lagrangian is there

- * In FEYNARTS

- ➡ Powerful tool for computing vertices
- ➡ Interfaced with FEYNRULES

Automatic UV counterterms ✓✓✓

Automatic R_2 vertices ✓



UFO@NLO

Status legend:

✓✓✓ : Implemented, tested and working.

✓✓✓ : Implemented, almost finished testing.

✓ : Implementation in progress.

Speed improvements: Multicore

✧ Multicore is fully supported

- ⇒ Need to set the boolean `FR$Parallel = True;`
- ⇒ Set the number of cores to use `FR$KernelNumber`
- ⇒ **Second master kernel free**
- ⇒ Speed **improvement!**

```
Model MSSM loaded.
Starting Feynman rule calculation.
Expanding the Lagrangian...
Starting Feynman rule calculation.
Expanding the Lagrangian...
Expansion of indices distributed over 8 kernels.
Collecting the different structures that enter the vertex.
417 possible non-zero vertices have been found -> starting the computation: 417 / 417.
417 vertices obtained.
{81.117552, Null}
```

→ On 1 core 351 s ⇒ gain of **factor 4!**

Conclusion

- ✱ FEYNRULES is a MATHEMATICA package
- ✱ In the last stable version of FEYNRULES:
 - ⇒ Model implementation minimal
 - ⇒ **Superspace** package
 - ⇒ Many interfaces to Monte Carlo tools
 - ⇒ **UNIVERSAL FEYNRULES OUTPUT**
- ✱ The next version's features
 - ⇒ Spectrum generator (with ASPERGE). ✓
 - ⇒ Decay package for $1 \rightarrow 2$ processes. ✓
 - ⇒ Rarita-Schwinger spin-3/2 field implemented. ✓
- ✱ NLO computations and automatic renormalization of the Lagrangian. ✗ next version

Thank you for your attention

NLO in FEYNRULES: Technical details

* Assumptions

- ⇒ Maximum dimension of the operators = 4
- ⇒ Feynman Gauge
- ⇒ $\{\gamma_5, \gamma_\mu\} = 0$
- ⇒ 't Hooft-Veltman scheme
- ⇒ On-shell scheme for the masses and wave functions of massive states
- ⇒ $\overline{\text{MS}}$ otherwise

* Outlook

- ⇒ Effective theories
- ⇒ Other gauge (not Feynman)
- ⇒ ... any suggestion?