Implementation and validation of the MSSM in FeynRules.

Benjamin Fuks (IPHC Strasbourg)

In collaboration with C. Duhr (UCL), N. Christensen (MSU) & MadGraph people.

GDR SUSY meeting @ LAL (Orsay)
December 04, 2008

Outline

1. Motivation
2. FeynRules
   - What is FeynRules?
   - Example: the QCD Lagrangian
3. Implementation of the general MSSM
   - The model
   - Validation
4. Summary - outlook

Outline

1. Motivation
2. FeynRules
   - What is FeynRules?
   - Example: the QCD Lagrangian
3. Implementation of the general MSSM
   - The model
   - Validation
4. Summary - outlook

Introduction to leading order automated tools

- One of the LHC purposes: which model of new physics is the correct one?
  - We need data [which are hopefully coming ~~next year~~ in the next-to-leading years].
  - We need theoretical predictions for all BSM models.

Confront data and theory

- But...
  - Often we have to calculate more than 1.000 (even 10.000) diagrams.
  - The help of automated tools is mandatory.

- Tools zoology
  - CalcHEP/CompHEP [Pukhov *et al.* (1999); Boss *et al.* (2004)].
  - FeynArts/FormCalc [Hahn (1999,2001)].
  - Herwig [Corcella *et al.* (2001); Bahr *et al.* (2008)].
  - MadGraph/MadEvent [Alwall *et al.* (2007); Maltoni, Stelzer (2003)].
  - Sherpa [Gleisberg *et al.* (2004)].
  - Whizard/Omega [Moretti *et al.* (2001); Kilian *et al.* (2007)].
  - ...

Implementing new physics models in diagram calculators

- Why using several programs?
  - * Each has its own strengths and weaknesses.
  - * Golden project: simultaneous implementation of a new model.
  - * Compare the results.
- Implementing a new model $\equiv$ writing a list of Feynman rules.
  - * Often one vertex at a time.
  - * Tedious and error prone process.
  - * Each program has its own conventions.
  - * Validation and bug corrections.
  - * E.g. the general MSSM without four-scalars vertices: $\mathcal{O}(1000)$ vertices.

FeynRules [Duhr and Christensen (2008)]

- * **Automization.**
- * Mathematica-based package calculating Feynman rules from a Lagrangian.
- * Generating a model file appropriate for each program.

Outline

1 Motivation

2 FeynRules
- What is FeynRules?
- Example: the QCD Lagrangian

3 Implementation of the general MSSM
- The model
- Validation

4 Summary - outlook

FeynRules in details

- Input:
  * Particles and fields.
  * Gauge groups.
  * Parameters (masses, coupling constants, mixing matrices,...).
  * The Lagrangian.
- Processing in Mathematica a list of generic vertices.
- Re-processing the list to:
  * A TEX-file.
  * A FeynArts/FormCalc model file.
  * A MadGraph/MadEvent model file.
  * A CalcHep/CompHep model file.
  * A Sherpa model file.

Is your favourite code missing?
$\Rightarrow$ **Contact us !**

The FeynRules philosophy

* **Theorist-friendly environment to develop new models**:
  Mathematica-based.
* **Filling the gap between model building and collider phenomenology.**
  1) Lagrangian → FeynRules → model file for your Monte Carlo code.
  2) Monte Carlo code → phenomenology.
* **Avoid separate implementations of a model on different programs.**
  FeynRules does it for you!
* **Exploit the strengths of the different programs!**

Example: the QCD Lagrangian - parameters

- Parameters of the model:

Parameters

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

- * Contains all the information needed by the Monte Carlo codes.
- * Contains the TEX-form required to write the TEXfile.

Example: the QCD Lagrangian - gauge group

- The gauge group $SU(3)_C$:

Gauge group

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

* We have defined the gauge boson G as the gluon field.
* We have associated the parameter gs as the QCD coupling constant.
* The structure functions, the representations, ... are defined.

Example: the QCD Lagrangian - particles

- The quark fields:

Particle list

```
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  -> {"q", "d", "u", "s", "c", "b", "t"},
    PropagatorType   -> Straight,
    PropagatorArrow  -> Forward}
```

* Organized in classes ⇒ implicit summations ≡ compact Lagrangian.
* Contains all the information needed by the Monte Carlo codes.

Example: the QCD Lagrangian - the Lagrangian

- The QCD Lagrangian is

$$\mathcal{L}_{\rm QCD} = -\frac{1}{4} G^a_{\mu\nu} G^{a\mu\nu} + \bar{q}_f \left(i\partial\!\!\!/ - m_f + g_s G\!\!\!/^a T^a \right) q_f \,,$$

where we are summing over the quark flavours.

Lagrangian

```
LQCD = -1/4 * FS[G, mu, nu, a] * FS[G, mu, nu, a] +

  I*qbar.Ga[mu].del[q, mu] - MQ[f] * qbar[s,f,c].q[s,f,c] +

  gs * G[mu,a] * qbar.Ga[mu].T[a].q
```

* The gluon strength tensor is automatically defined with the gauge group.
* Implicit summations over the flavours.
* Easy debugging.

Example: the QCD Lagrangian - results

- We obtain the Feynman rules

Results

```
FeynmanRules[LQCD, FlavorExpand->False]

    Vertex 1
    Particle 1 : Vector , G
    Particle 2 : Dirac , q†
    Particle 3 : Dirac , q
    Vertex:
```

$$i\, g_s\, \gamma^{\mu_1}_{s_2,s_3}\, \delta_{f_2,f_3}\, T^a_{m_2,m_3}$$

  * Explicit flavour expansion possible.
  * We would have then six vertices, one for each flavour.

Outline

1. Motivation
2. FeynRules
   - What is FeynRules?
   - Example: the QCD Lagrangian
3. Implementation of the general MSSM
   - The model
   - Validation
4. Summary - outlook

Implementation of the general MSSM in FeynRules

- A general version of the MSSM is considered.
  - * All possible mixings in the scalar sector.
    $\Rightarrow 6 \times 6$ flavour violating mixing matrices.
  - * All possible complex phases.
  - * One exception (presently): CP violating Higgs mixing.
- *Usual MSSM* limit easily taken.
- We want a Lagrangian as easy as possible.
  - * Partially written in the interaction basis.
  - * Partially written in the mass basis.
- Model parameters.
  - * Follow a SLHA-2-like format, the SLHA-FR format.
  - * Provided with a C++ translator from SLHA1/2 to SLHA-FR.

Validation (1)

- Analytical cross sections: FeynArts/FormCalc interface.
  - * Check of the FC-produced formulas with litterature.
  - ✔ All $2 \longrightarrow 2$ SUSY particle pair hadroproduction processes.
  - ✘ $2 \longrightarrow 3$ processes: FormCalc-6.0 issue.
- MadGraph/MadEvent
  - * Limit of the usual MSSM.
  - * Numerical check versus the MadGraph-Stock MSSM model.
  - * MG-Stock was validated by the CATPISS collaboration [Hagiwara *et al.* (2006)].
  - ✔ 320 decay widths.
  - ✔ 456 $2 \longrightarrow 2$ SUSY particle pair production processes.
  - ☛ $2 \longrightarrow 3$ processes: ongoing work (good for the signs in the vertices).

22.10.2008, IPHC Strasbourg

✌ **Our first milestone: 776 succesfully tested processes !**

Validation (2)

- Some examples:

Process	MG-FR	MG-Stock	Result
b,t~>sd1,su1~	3.9273×10^{-1}	3.9192×10^{-1}	OK: 0.206675%
b,t~>sd1,su6~	3.6715×10^{-1}	3.675×10^{-1}	OK: 0.0952381%
b,t~>sd2,su1~	4.2427×10^{-1}	4.2506×10^{-1}	OK: 0.185856%
b,t~>sd2,su6~	4.7632×10^{-1}	4.7523×10^{-1}	OK: 0.229363%
b,t~>x1-,n1	5.6383×10^{-4}	5.6449×10^{-4}	OK: 0.11692%
b,t~>x1-,n2	5.9582×10^{-3}	5.9638×10^{-3}	OK: 0.0938999%
b,t~>x1-,n3	5.2845×10^{-3}	5.2925×10^{-3}	OK: 0.151157%
b,t~>x1-,n4	6.5567×10^{-3}	6.5586×10^{-3}	OK: 0.0289696%
b,t~>x2-,n1	2.2335×10^{-3}	2.235×10^{-3}	OK: 0.0671141%
b,t~>x2-,n2	5.8572×10^{-3}	5.8536×10^{-3}	OK: 0.0615006%
b,t~>x2-,n3	2.3739×10^{-2}	2.3737×10^{-2}	OK: 0.00842566%
b,t~>x2-,n4	2.1151×10^{-2}	2.1143×10^{-2}	OK: 0.0378376%
b,t~>h-,h1	6.9053×10^{-3}	6.8954×10^{-3}	OK: 0.143574%
b,t~>h-,h2	1.392×10^{-3}	1.3915×10^{-3}	OK: 0.0359324%
b,t~>h-,h3	1.3642×10^{-3}	1.3656×10^{-3}	OK: 0.102519%
b,t~>z,h-	1.2562×10^{-2}	1.2558×10^{-2}	OK: 0.0318522%

Validation (3)

- Some examples (cont'd):

Process	MG-FR	MG-Stock	Result
tau-,vt~>sl1-,sv1~	1.2619×10^{-2}	1.261×10^{-2}	OK: 0.0713719%
tau-,vt~>sl6-,sv1~	5.7656×10^{-2}	5.7647×10^{-2}	OK: 0.0156123%
tau-,vt~>x1-,n1	2.2155×10^{-2}	2.2162×10^{-2}	OK: 0.0315856%
tau-,vt~>x1-,n2	2.0793×10^{-2}	2.0788×10^{-2}	OK: 0.0240523%
tau-,vt~>x1-,n3	2.0889×10^{-3}	2.0851×10^{-3}	OK: 0.182245%
tau-,vt~>x1-,n4	3.3175×10^{-3}	3.3171×10^{-3}	OK: 0.0120587%
tau-,vt~>x2-,n1	2.0102×10^{-3}	2.011×10^{-3}	OK: 0.0397812%
tau-,vt~>x2-,n2	3.3299×10^{-3}	3.3284×10^{-3}	OK: 0.0450667%
tau-,vt~>x2-,n3	2.5187×10^{-2}	2.5226×10^{-2}	OK: 0.154602%
tau-,vt~>x2-,n4	2.2665×10^{-2}	2.2631×10^{-2}	OK: 0.150236%
tau-,vt~>h-,h1	2.1315×10^{-6}	2.1436×10^{-6}	OK: 0.564471%
tau-,vt~>h-,h2	4.7254×10^{-3}	4.7229×10^{-3}	OK: 0.0529336%
tau-,vt~>h-,h3	4.716×10^{-3}	4.729×10^{-3}	OK: 0.2749%
tau-,vt~>w-,h1	7.3062×10^{-3}	7.3161×10^{-3}	OK: 0.135318%
tau-,vt~>w-,h2	8.1658×10^{-3}	8.1661×10^{-3}	OK: 0.00367372%
tau-,vt~>w-,h3	8.1923×10^{-3}	8.1858×10^{-3}	OK: 0.0794058%
tau-,vt~>z,h-	1.3762×10^{-2}	1.377×10^{-2}	OK: 0.0580973%

Validation (4)

- Some examples (cont'd):

Process	MG-FR	MG-Stock	Result
g,a>su1,su1~	5.9465×10^{-2}	5.9485×10^{-2}	OK: 0.0336219%
g,a>su2,su2~	4.6418×10^{-2}	4.6371×10^{-2}	OK: 0.101356%
g,a>su3,su3~	4.636×10^{-2}	4.6361×10^{-2}	OK: 0.00215699%
g,a>su4,su4~	4.561×10^{-2}	4.5558×10^{-2}	OK: 0.11414%
g,a>su5,su5~	4.5588×10^{-2}	4.555×10^{-2}	OK: 0.0834248%
g,a>su6,su6~	4.4111×10^{-2}	4.4146×10^{-2}	OK: 0.0792824%
g,a>sd1,sd1~	1.2235×10^{-2}	1.2236×10^{-2}	OK: 0.00817261%
g,a>sd2,sd2~	1.1686×10^{-2}	1.1693×10^{-2}	OK: 0.0598649%
g,a>sd3,sd3~	1.1659×10^{-2}	1.1662×10^{-2}	OK: 0.0257246%
g,a>sd4,sd4~	1.1659×10^{-2}	1.1665×10^{-2}	OK: 0.0514359%
g,a>sd5,sd5~	1.1298×10^{-2}	1.1294×10^{-2}	OK: 0.035417%
g,a>sd6,sd6~	1.1286×10^{-2}	1.1278×10^{-2}	OK: 0.0709346%

Outline

1 Motivation

2 FeynRules
- What is FeynRules?
- Example: the QCD Lagrangian

3 Implementation of the general MSSM
- The model
- Validation

4 Summary - outlook

Summary - Outlook

- FeynRules.
  * Mathematica-based package computing Feynman rules from a Lagrangian.
  * Generic output.
  * Generating model file feeding ~~some~~ as many MC codes as possible. [contact us to add yours].
  * The model library is getting bigger and bigger. [contact us to add your favourite one].
- The general MSSM model.
  * The implementation is achieved.
  * The validation is ongoing.
- FeynArts/FormCalc.
  * Find and fix the bug in FormCalc-6.0.
  * Check additional $2 \to 3$ processes.
- MadGraph/MadEvent.
  * Check the $2 \to 3$ SUSY particle production processes.
  * Check the general MSSM $\Rightarrow$ vs. XSUSY [BenjF & Herrmann (in preparation)].
- CalcHEP/CompHEP and Sherpa validation.