FeynRules Tutorial

Claude Duhr
IPPP/Durham

January 13, 2010

1 Introduction

The main topic of this tutorial is FEYNRULES, a MATHEMATICA package that allows to derive Feynman rules from a Lagrangian and to implement them into various HEP tools in an automated way. In particular, during this tutorial we will

- implement a simple model of Dark Matter into FEYNRULES and derive its Feynman rules.
- export the model to MICROMEGAS to compute the DM relic density.
- export the model to MADGRAPH to study its LHC phenomenology.

2 The Model

The model we will consider consists in the Standard Model (SM) augmented by a single singlet scalar field S. For S to be a viable Dark Matter (DM) candidate it must be stable, or in other words there must be a symmetry ("R-parity") that prevents S from decaying. We choose here the minimal scenario and we postulate a $\mathbb{Z}_2$ symmetry such that S is $\mathbb{Z}_2$-odd whereas all SM particles are $\mathbb{Z}_2$-even. the Lagrangian for this model then takes the very simple form,

$$\mathcal{L} = \mathcal{L}_{SM} + \mathcal{L}_{DM}\,, \tag{1}$$

where $\mathcal{L}_{SM}$denotes the SM Lagrangian, and the Dark Matter sector is,

$$\mathcal{L}_{DM} = \frac{1}{2}\partial_\mu S\,\partial^\mu S - \frac{1}{2}\left(m_S^2 + \lambda_{SH}\,v^2\right)S^2 + \lambda_S S^4 + \lambda_{SH} S^2\,\Phi^\dagger\Phi\,, \tag{2}$$

where v denotes the vev of the SM Higgs field

$$\Phi = \left(\phi^\pm, \frac{H + i\phi^0 + v}{\sqrt{2}}\right)^T \tag{3}$$

and m_S denotes the mass of the scalar S. m_S, λ_S and λ_{SH} are free parameters of the model. Here we choose for the sake of the example $m_S = 70$ GeV and $\lambda_S = \lambda_{SH} = 0.02$.

3 The FeynRules implementation

The next step is to implement the model (*i.e.*, the Lagrangian) into FeynRules. Since our model consists in the simple addition of a new particle that leaves the SM sector of the model unchanged, we do not need to change anything to $\mathcal{L}_{SM}$ and we can concentrate solely on $\mathcal{L}_{DM}$.

Implementing a new model in FeynRules consists in three steps:

1. Define the new particle.
2. Define the new parameters.
3. Enter the Lagrangian into Mathematica.

.

3.1 Defining new particles and parameters: the FeynRules model file

New particles and parameters are defined in the so-called model-file, a text-file that contains lists of all the particles and parameters, together with their properties (*e.g.*, mass of a particles, numerical value of a parameter).

- Go to the directory `~/FeynRules1.4.3/Models/YETI10`,

  ```
  cd ~/FeynRules1.4.3/Models/yeti10
  ```

  In this directory you find a text file called `SM.fr`, which contains all the particle and parameter definitions for the SM. This file does not need to be changed, but we need to create an additional file where we define in a similar way the new particle S and the new parameters m_S, λ_S and λ_{SH}.

- **Preparing the new model file:** Next, open a new text file called `ScalarDM.fr` in the working directory. We start be including some information on our new model. The name of the new model can be entered via the `M$ModelName` variable,

  ```
  M$ModelName = "Scalar_Dark_Matter";
  ```

  Note that this variable also sets the name of all output file produced by FeynRules. You can include some more information about the model for future via the `M$Information` variable,

  ```
  M$Information = {
               Authors      -> {"C. Duhr"},
               Version      -> "1.0",
               Date         -> "13. 01. 2010",
               Institutions -> {"Durham / IPPP"},
               Emails       -> {"claude.duhr@durham.ac.uk"}};
  ```

- **Defining a new particle:** New particles are included in a list called `M$ClassesDescription`. In our case, this list only contains the new scalar particle,

```
M$ClassesDescription = {
     S[4] == {
           ClassName     -> sc,
           SelfConjugate -> True,
           Mass          -> {Msc, 70},
           ParticleName  -> "~sc"}
          }
```

This defines a new scalar (`S`) with name `sc`[1]. The number in square brackets is just a counter that numbers the scalar particles in the model (in this case it is 4, since the SM already contains three scalars: the higgs boson and the two pseudo-Goldstone bosons). The scalar is furthermore real (`SelfConjugate -> True`) and has a mass of 70 GeV (`Mass -> {Msc, 70}`). Note that in this way we already defined the parameter m_S that appears in the Lagrangian. A comment is in order about the option `ParticleName`. The `ClassName` of the particle refers to the name used inside Mathematica. The `ParticleName` on the other hand however refers to the name (a string) to be used later on in MicrOmegas and MadGraph. The reason for this is that Micromegas requires the names of all $\mathbb{Z}_2$-odd particles to start with a ~.

- **Defining new parameters:** We already defined the mass m_S when we defined the new particle, but we still need to define the two new coupling constants λ_S and λ_{SH} by including them into the list `M$Parameters` as shown:

```
M$Parameters = {
     lS == {
          Value -> 0.02,
          InteractionOrder -> {QED, 2},
          Description -> "DM scalar couplings"},

     lSH == {
          Value -> 0.02,
          InteractionOrder -> {QED, 2},
          Description -> "DM - Higgs coupling"}
}
```

`Description` can be any string, and only serves as a documentation. `InteractionOrder` is an additional option required by MadGraph. For now it suffices to now that this defines λ_S and λ_{SH} as "quartic couplings of electroweak strength".

- Finally, save the text file as `ScalarDM.fr` in the current working directory.

3.2 Running FeynRules

- **Loading FeynRules:** Open an empty Mathematica notebook, and load the FeynRules package via the command:

---

[1] Note that we cannot call the new particle S, because the symbol `S` is already used to refer to scalars in general in FeynRules.

```
$FeynRulesPath = SetDirectory["~/FeynRules1.4.3"];

<<FeynRules‘

SetDirectory[$FeynRulesPath <> "/Models/YETI10"];
```

- **Loading the model files:** Once the FeynRules package is loaded, we have to read in the model files with the particle and parameter definitions:

  ```
  LoadModel[ "SM.fr", "ScalarDM.fr" ];
  ```

- **Implementing a Lagrangian:** We are now ready to enter the Lagrangian of the Model. The SM part of the Lagrangian is already contained in `SM.fr` and is stored in the variables `LGauge`, `LFermions`, `LHiggs` and `LGhost`.

  Let us now turn to the new sector described by the Lagrangian in Eq. (2). The Lagrangian can be entered into Mathematica in the following way:

Textbook expression	Mathematica expression
$\mathcal{L}_{DM} =$	`LDM :=`
$\frac{1}{2}\partial_\mu S\,\partial^\mu S$	`1/2 del[Sc, mu] del[Sc,mu]`
$-\frac{1}{2}\left(m_S^2 + \lambda_{SH}\,v^2\right) S^2$	`-1/2 (Msc^2 + lSH v^2) Sc^2`
$+\lambda_S S^4$	`+lS Sc^4`
$+\lambda_{SH} S^2\,\Phi^\dagger\Phi$	`+lSH Sc^2 Phibar.Phi`

  `Phi` and `Phibar` are predefined in the SM model file and denote the Higgs doublet of Eq. (3).

- **Running FeynRules:** We are now ready to run FeynRules. First we can perform some basic sanity checks on the Lagrangian we just implemented, *e.g.*, checking the hermiticity of the Lagrangian. This can be done using the `CheckHermiticity` command as shown,

  ```
  CheckHermiticity[ LDM ];
  ```

  After the Lagrangian passed this check, we may have a look at the interaction vertices involving the new particle by using the `FeynmanRules` command:

  ```
  FeynmanRules[ LDM ];
  ```

3.3 Running the interfaces

We have now achieved a complete FeynRules implementation of our model, and we can now output the model to various matrix element generators to study the phenomenology of the model. In this tutorial we want to use MicrOmegas to compute the DM relic density as well as MadGraph to compute the DM production cross section in a specific channel at the LHC.

Since MicrOmegas is based on CalcHep, we can implement our model into MicrOmegas by asking FeynRules to write a CalcHep output by means of the corresponding interface. Since CalcHep work in Feynman gauge, we first switch to this particular gauge via,

```
FeynmanGauge = True;
```

This command tells FeynRules to include all the ghost and pseudo-Goldstone vertices when computing the interaction vertices. The computation of the vertices and the writing of the CalcHep model files is done via the command

```
WriteCHOutput[ LGauge, LFermions, LHiggs, LYukawa, LGhost, LDM ];
```

FeynRules now computes all the interaction vertices associated with the Lagrangians and produces a set of output files, saved in the directory `Scalar_Dark_Matter-CH`, that can be directly read into CalcHep and MicrOmegas.

Similarly, the input files for MadGraph can be generated via the command:

```
FeynmanGauge = False;

WriteMGOutput[ LGauge, LFermions, LHiggs, LYukawa, LDM ];
```

where we turned the switch to Feynman gauge off because MadGraph only works in unitary gauge. The output files for MadGraph are saved in the directory `Scalar_Dark_Matter_MG`.

4 MicrOmegas

Since the new scalar S we introduced is the only $\mathbb{Z}_2$-odd particle in our model, it is necessarily stable and hence a candidate for Dark Matter. The CalcHep implementation of our model obtained by FeynRules can be used together with MicrOmegas to compute in an automated way the relic density of our DM candidate. The corresponding steps will be described in the following.

- Go to the MicrOmegas home directory:

  ```
  cd ~/micromegas_2.2.CPC.i
  ```

- **Creating a new project:** We first need to create a new working directory for our model. This is done via the shell command

  ```
  ./newProject scalarDM
  ```

  This creates a directory `scalarDM` in the MicrOmegas home directory that contains all the routines needed to compute the relic density for a given model. The information on the model itself (particles, vertices, *etc.*) is however not yet included.

- **Adding a new model:** Go in to the working directory we just created,

  ```
  cd scalarDM
  ```

  To add a new model it is enough to copy the CalcHep files written by FeynRules into the subdirectory `work/models`,

  ```
  cp  -r  ~/FeynRules1.4.3/Models/YETI10/Scalar_Dark_Matter-CH/*   work/models/
  ```

The model is now ready to be used just like all other built-in models.

- **Running** MicrOmegas**:** To run MicrOmegas we need to create an input file for the parameters of the model. Open a new text file `input.par` in your text editor. You can specify in this file the numerical values for the parameters of the DM sector, *e.g.*,

  ```
  Msc   70
  lS    0.02
  lSH   0.02
  ```

  If you leave this file empty, then the default values given in the model file will be used.

  Next, we need to compile the MicrOmegas executable,

  ```
  gmake   main=main.c
  ```

  and we can finally run MicrOmegas via the command

  ```
  ./main  input.par
  ```

  MicrOmegas will now compute the DM annihilation rate for $DMDM \to X$, where DM can be any $\mathbb{Z}_2$-odd particle. By convention, MicrOmegas assumes that the names of all $\mathbb{Z}_2$-odd particles start by `~`. In our case, there is only one such particle, `~sc`, and so the code will compute the annihilation rate for `~sc ~sc -> X`.

- Play with it! You can now play with the new model, analyze the output, change the input parameters in `input.par`, *etc.*

5 MadGraph

MadGraph is a program that allows to generate the tree-level matrix element for a given process, and is hence suitable to study the collider phenomenology of a model. In this tutorial we are interested in the DM production at the LHC, and we will in particular compute the LHC cross section for the process $p\,p \to Z\,S\,S$.

- Go to the MadGraph home directory,

  ```
  cd ~/MG_ME_V4.4.32
  ```

- **Adding a new model:** All models are saved in the subdirectory `Models`, and so we need to copy our FeynRules-generated model to this directory,

  ```
  cp  -r  ~/FeynRules1.4.3/Models/YETI10/Scalar_Dark_Matter_MG   Models/
  ```

  It is possible to test the sanity of the model by executing the sequence of commands

  ```
  cd Models/Scalar_Dark_Matter_MG
  make testprog
  ./testprog
  ```

If the model is correct, then the numerical values of all the parameters and couplings in the model should be printed on the screen.

- **Running MadGraph:** To obtain the cross section for our process with MadGraph, we first need to create a new working directory. Go back to the MadGraph home directory, and make a copy of the directory `Template`,

```
cp  -r  Template  scalarDM
```

The process is entered via the `proc_card.dat`, stored in the `Cards/` subdirectory of scalarDM. Note however that the name of the new particle has slightly changed! MadGraph does not allow a particle name to start with a ~ (which was mandatory for MicrOmegas). FeynRules corrected for this by renaming the particle into `hsc`. The process is then entered as follows in the `proc_card.dat`:

```
# Begin PROCESS      # This is TAG. Do not modify this line

p,p>Z,hsc,hsc   @1   # First Process
QCD=99               # Max QCD couplings
QED=99               # Max QED couplings
end_coup             # End the couplings input

done                 # this tells MG there are no more procs
```

By default, MadGraph always assumes SM processes, so we need to change the model information in the `proc_card.dat` as shown:

```
# Begin MODEL  # This is TAG. Do not modify this line
Scalar_Dark_Matter_MG
# End   MODEL  # This is TAG. Do not modify this line
```

We are now in principle ready to run the code. However, for simplicity, let us change the definition of the proton so that it only contains up and down quarks.

```
# Begin MULTIPARTICLES # This is TAG. Do not modify this line
P uu~dd~g
```

Once the `proc_card.dat` is ready, we can generate the matrix element for our process by running

```
./bin/newprocess
```

in the `scalarDM` directory. MadGraph now generates a Fortran code for the tree-level matrix element that can be evaluated numerically in a fast and stable way. You can also have a look at the Feynman diagrams that contribute to the process by opening the file `index.html` in a browser and clicking on `Process Information`. Furthermore, the matrix element can also be integrated over phase space running the command

```
./bin/generate_events  0
```

This may take some time depending on the machine. Since we are only interested in the cross section, you can speed up this step by lowering the number of events requested in the `run_card.dat` in the directory `Cards`,

```
  10       = nevents ! Number of unweighted events requested
```

At the end, the result of the integration can be accessed in `index.html` by clicking on `Results and Event Database`.

- **Advanced: Decaying the Z boson:** The Z boson we just produced must of course decay into something. We can for example ask MadGraph to generate the cross section for the same process but where the Z boson decays into an electron-positron pair, $p\,p \to Z\,S\,S \to e^+\,e^-\,S\,S$ by asking int he `proc_card.dat` for the process `p,p>hsc,hsc,(Z>e,e~)`.
- **Advanced: Applying final state cuts:** By default MadGraph applies a set of standard cuts on the final states, *e.g.*,

  $$|p_{T,\text{leptons}}| > 10 \text{ GeV}, \quad |\eta_{\text{leptons}}| < 2.5, \quad E_{T,\text{leptons}} > 0 \text{ GeV}.$$

  The values for the cuts are set in the `run_card.dat` (in the subdirectory `Cards`) via,

  ```
   10  = ptl     ! minimum pt for the charged leptons

    0  = el      ! minimum E for the charged leptons

  2.5  = etal    ! max rap for the charged leptons
  ```

  You can change the values for the cuts and observe how the cross section changes.
- **Advanced: From events to plots:** MadGraph does not only generate the number for the cross section, but also (unweighted) events from which distributions can be obtained. The number of events is set in the `run_card.dat` via

  ```
    10000       = nevents ! Number of unweighted events requested
  ```

  and the generated events are stored in `Events/XX_unweighted_events.lhe.tgz`, where `XX` is the name you gave to the run. The events can be used to make plots. You can go to the MadGraph homepage, `http://madgraph.phys.ucl.ac.be` and click on `Tools` and then on `Plotting Interface (MadAnalysis)`. You can upload your events here and retrieve a set of plots for basic distributions.

6 Advanced: DM pair production via gluon fusion at the LHC

In the previous section we generated the LHC cross section for the production of a pair of DM particles in association with a Z boson. It is also possible to produce directly a pair of DM particles without a

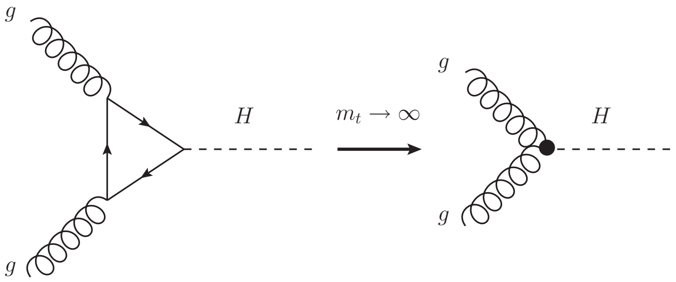

Figure 1: The effective coupling of gluons to the Higgs boson.

vector boson, via a process of the form $p\,p \to H^* \to S\,S$. At the LHC, however, the main production channel for a Higgs boson is not $q\,\bar{q}$ annihilation, but gluon fusion, *i.e.*, a pair of gluon effectively coupling to a Higgs boson via a top quark loop (See Fig. 1). In the large top mass limit, this effective coupling can be described by the higher-dimensional operator

$$\mathcal{L}_{GF} = -\frac{1}{4}\,A_H\,H\,G^a_{\mu\nu}\,G_a^{\mu\nu}\,, \tag{4}$$

where A_H denotes the effective coupling of the gluons to the Higgs boson,

$$A_H = \frac{-g_s^2}{4\pi(3\pi v)}\left[1 + \frac{7}{30}\left(\frac{m_H}{2m_t}\right)^2 + \frac{2}{21}\left(\frac{m_H}{2m_t}\right)^4 + \frac{26}{525}\left(\frac{m_H}{2m_t}\right)^6 + \mathcal{O}\left(\frac{m_H}{2m_t}\right)^8\right]. \tag{5}$$

You can use FeynRules to augment our model by this operator and to implement it in MadGraph to compute the cross section for $p\,p \to H^* \to S\,S$.