aSuperIso_DM:
a new toolbox for dark matter calculations

Glenn ROBBINS
KBFI, Tallinn, Estonia

In collaboration with Alexandre Arbey, Benjamin Fuks and Nazila Mahmoudi

PASCOS 2019, Manchester, UK
July 4th 2019

An automisation of SuperIso Relic

SuperIso Relic

- Computation of flavour observables
- Relic density within the framework of adjustable cosmological models (dark energy, dark entropy, ...)
- Indirect detection (Fermi-LAT dSphs, AMS-02 antiprotons)
- Direct detection (Xenon1T, LUX, PANDA-X,PICO60...)
- **But specific to the MSSM and NMSSM**

aSuperIso_DM (automated SuperIso for Dark Matter)

A generalisation of SuperIso Relic to any quantum field theory that can be cast under a Lagrangian form and that features a dark matter particle

An automisation of SuperIso Relic

SuperIso Relic

- Computation of flavour observables
- Relic density within the framework of adjustable cosmological models (dark energy, dark entropy, ...)
- Indirect detection (Fermi-LAT dSphs, AMS-02 antiprotons)
- Direct detection (Xenon1T, LUX, PANDA-X,PICO60...)
- **But specific to the MSSM and NMSSM**

aSuperIso_DM (automated SuperIso for Dark Matter)

A generalisation of SuperIso Reli
that can be cast under a Lagrang ... ark
matter particle

Warning:
The following is a preliminary presentation,
the final code and manual is coming soon!

FeynRules Interface

Previously on
SuperIso Relic

Squared Amplitudes for DM (co-)annihilations generated with FormCalc from the (N)MSSM LanHEP model files.

$\rightarrow$ **Flavour-expanded diagrams**:
similar squared amplitudes were calculated multiple times ($\chi\,\chi \to e^+e^-$, $\mu^+\mu^-$, $\tau^+\tau^-$ ...)

$\rightarrow$ Large amount of files and large compilation time

Now on
aSuperIso_DM

Squared amplitudes are generated from a FeynRules model file using Mathematica routines interfaced with FeynRules.

$\rightarrow$ **Diagrams not flavour-expanded**

$\rightarrow$ Small amount of files and reduced compilation time

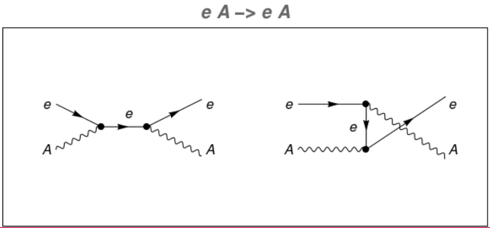
Diagrams

Simple Mathematica interface to obtain and visualise 2→2 processes from a list of Feynman Rules

In[1]:= **diag**=**Diagram22**[**feynmanrules**,{**e**,**A**,**e**,**A**}]

Out[1]={DiagramList[{e,A,e,A},
{{e,A,e,A,m$s},e}, {{e,1},$-ie\gamma_{s_5,s_1}{}^{\mu_2}$,
{A,2},{e, m$s}},{e,3}, $-ie\gamma_{s_3,s_6}{}^{\mu_4}$,{A,4}}},
{{e,A,e,A,m$u,e},{{e,1}, $-ie\gamma_{s_5,s_1}{}^{\mu_4}$,
{A,4},{e,m$u},{A,2}, $-ie\gamma_{s_3,s_6}{}^{\mu_2}$,{e,3}}}]]

In[2]:= **PaintDiagramList**[**diag**]
Out[2]=



Diagrams

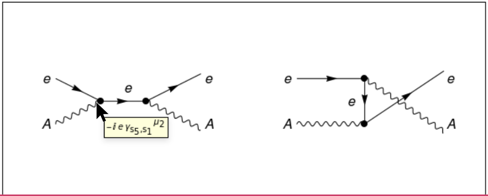
Simple Mathematica interface to obtain and visualise 2→2 processes from a list of Feynman Rules

```
In[1]:= diag=Diagram22[feynmanrules,{e,A,e,A}]

Out[1]={DiagramList[{e,A,e,A},
        {{e,A,e,A,m$s},e}, {{e,1},−ieγ_{s5,s1}^{μ2} ,
        {A,2},{e, m$s}},{e,3}, −ieγ_{s3,s6}^{μ4} ,{A,4}}},
        {{e,A,e,A,m$u,e},{{e,1}, −ieγ_{s5,s1}^{μ4} ,
        {A,4},{e,m$u},{A,2}, −ieγ_{s3,s6}^{μ2} ,{e,3}}}]]

In[2]:= PaintDiagramList[diag]
Out[2]=
```

e A -> e A



Squared Amplitude

`AmpSquared22` is a function to obtain explicit formulae for squared amplitudes, as a function of particle masses and Mandelstam variables S,T, U ($m\$s$, $m\$t$, $m\$u$).
It makes use of abbreviations and sum over propagator generation numbers to gain computation time.
$\Rightarrow$ compared to `FeynCalc` for processes in the Standard Model

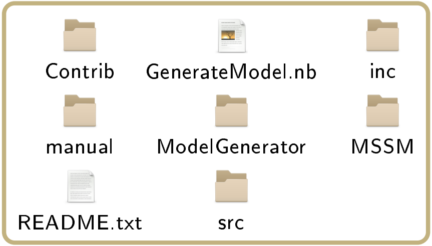
```
In[1]:= AmpSquared22[diag]

Out[1]=
```

$$\frac{2e^4\left(6\text{Me}^8-\text{Me}^4\left(3\text{m\$s}^2+14\text{m\$sm\$u}+3\text{m\$u}^2\right)+\text{Me}^2\left(\text{m\$s}^3+7\text{m\$s}^2\text{m\$u}+7\text{m\$sm\$u}^2+\text{m\$u}^3\right)-\text{m\$sm\$u}\left(\text{m\$s}^2+\text{m\$u}^2\right)\right)}{(\text{Me}^2-\text{m\$s})^2(\text{Me}^2-\text{m\$u})^2}$$

Model generation

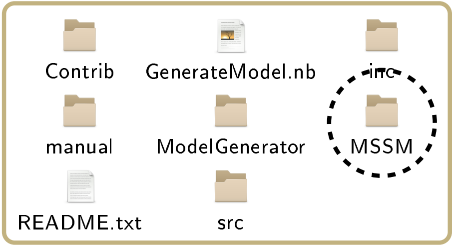
As an example, let's generate the MSSM model:



1. Create a new folder named after your model
2. Move your FeynRules model file to the model directory
3. Open GenerateModel.nb and run every block

Model generation

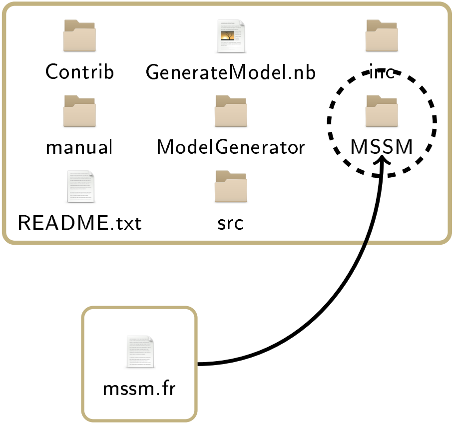
As an example, let's generate the MSSM model:



1. Create a new folder named after your model
2. Move your FeynRules model file to the model directory
3. Open GenerateModel.nb and run every block

Model generation

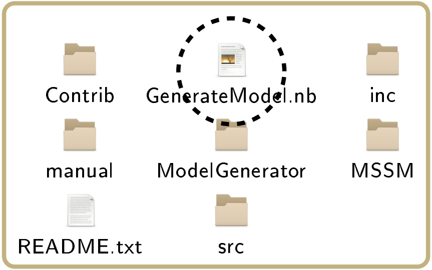
As an example, let's generate the MSSM model:



1. Create a new folder named after your model
2. Move your FeynRules model file to the model directory
3. Open GenerateModel.nb and run every block

Model generation

As an example, let's generate the MSSM model:



1. Create a new folder named after your model
2. Move your `FeynRules` model file to the model directory
3. Open GenerateModel.nb and run every block

Model generation

GenerateModel.nb * - Wolfram Mathematica 11.0

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

```
Quit[]

directory=NotebookDirectory[];
$FeynRulesPath=SetDirectory[directory<>"Contrib/feynrules2.4.68"];
<<FeynRules`;
 SetDirectory[directory<>"Contrib/feynrules2.4.68/Core"];
 <<Decay.m;
SetDirectory[directory<>"ModelGenerator"];
<<DMI.m;
```

Load FeynRules and squared amplitude calculation routines

```
(*This block needs to be modified by the user*)
ModelDirectory = "MSSM"; (*Please write here the directory name containing the FeynRules model file *)
ModelFileName = "mssm.fr"; (*Please write here the FeynRules model file name*)
FR$AmpSquaredParallelize = True; (*Set to True if you want the calculation of square
                     to be parallelized, False otherwise*)
FR$CPDiagrams = True; (*Set to True if the considered model conserves CP-parity.
                In this case, if a diagram is not self-conjugate, the conjugate
                The calculation time is thus significantly reduced.*)
printdiag = False; (*Set to True if you want to print all the 2-2 Feynman diagrams in "diagrams.pdf" *)
```

Block to be modified by the user

```
(*load model*)
directory = directory <> ModelDirectory;
SetDirectory[directory];
LoadModel[ModelFileName];
```

Load model

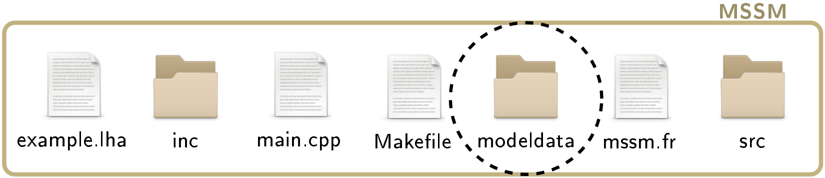
```
GenerateModel[directory, ModelFileName, PrintDiagrams → printdiag];
```

Generate C++ code

Code structure

The Mathematica interface generates:

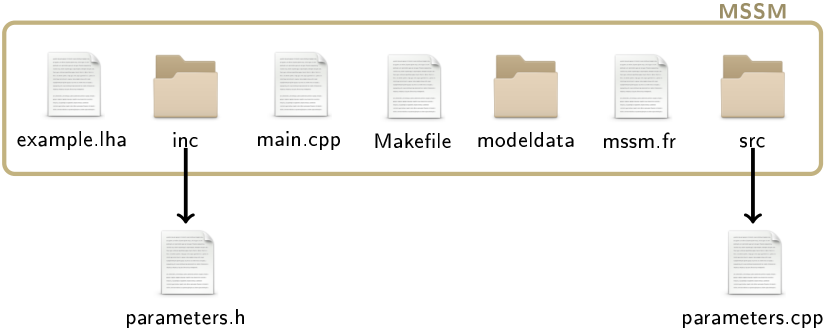
- Lagrangian and Feynman rules
- Model parameter structure definition and initialisation routine
- C++ files for the calculation of (co-)annihilation squared amplitudes, decay widths and branching ratios and DM-quark effective couplings



Code structure

The Mathematica interface generates:

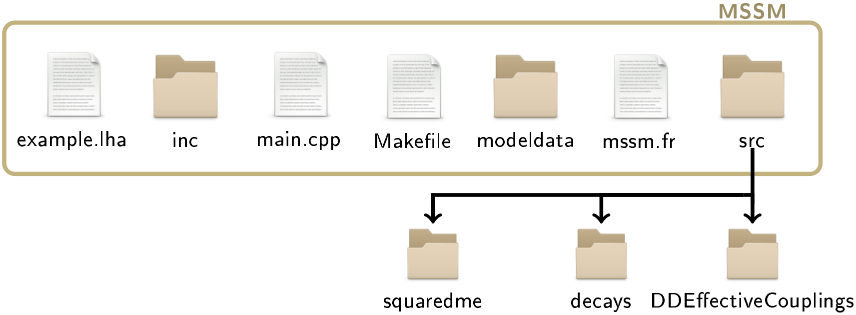
- Lagrangian and Feynman rules
- Model parameter structure definition and initialisation routine
- C++ files for the calculation of (co-)annihilation squared amplitudes, decay widths and branching ratios and DM-quark effective couplings



Code structure

The Mathematica interface generates:

- Lagrangian and Feynman rules
- Model parameter structure definition and initialisation routine
- C++ files for the calculation of (co-)annihilation squared amplitudes, decay widths and branching ratios and DM-quark effective couplings



Relic density

Alternative cosmological scenarios:

- Possibility to chose among several QCD equations of state
- Modification of the expansion rate: Dark density
- Modification of the entropy content:
  - Dark entropy
  - Reheating (radiation entropy)
- Decaying scalar field

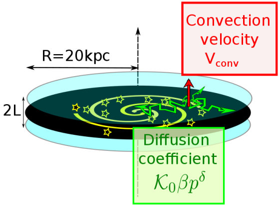
Big Bang Nucleosynthesis constraints:

$\Rightarrow$ using AlterBBN

Indirect detection

AMS-02 antiprotons

- Propagation code by Boudaud et al., interface with USINE coming soon
- **Two-zone diffusion model**



- **Semi-analytical resolution of the propagation equation**
- **Flexible DM halo density profile** (only needs to respect cylindrical symmetry)
- **Flexible propagation parameter values** (Diffusion coefficients, convection velocity...)

Fermi-LAT dwarf spheroidal galaxies (dSphs)

- Likelihoods interpolated from the tabulated likelihoods provided by Fermi-LAT collaboration.
- Flexible sets of dSphs

Direct detection

Spin-Dependent (SD) and spin-independent (SI) cross-sections

- Calculation for non-coloured particles with spin 0, 1/2, 1.
- Similarly to `micrOmegas`, SD and SI contributions to the DM-quarks effective couplings are extracted from the 2-2 DM-quark scattering diagrams by applying projection operators.

Constraints from XENON1T, PANDAX-2, PICO60...

- Flexible DM local density and velocity distriution
- Flexible nucleon and nuclear form factors

Comparison to similar codes

- $\sim$**30 minutes to generate whole MSSM model** on 4 cores, compared to a couple of hours to generate a CalcHep model from the same FeynRules model file (needs flavour-expansion in the latter case).
- Computation time of DM observables similar to SuperIso Relic
- Compilation time reduced to a couple of minutes
- Very close results compared to SuperIso Relic and micrOmegas in the MSSM (comparison to MadDM and in the frame of other models coming soon)

Prospects

- Complete resolution of the relic density Boltzmann equations for **multi-component dark matter**, including **semi-annihilations**.
- **Sommeferfeld Effect** for relic density and indirect detection.
- Models with **spin 3/2 and 2 fields**.
- Implementation of **flavour observables**.

Conclusions

- As supersymmetric scenarios are currently very challenged by expermimental constraints,
  `SuperIso Relic` transforms into `aSuperIso_DM`:
  - works for any particle physics model with a DM candidate respecting a Z2 symmetry
  - compilation time strongly reduced
  - a new way to visualise diagrams and obtain squared amplitude explicit formulae
  - conserves all the features for dark matter specific to `SuperIso Relic`
- **Public code and manual coming very soon!**