DarkPACK: A modular software to compute BSM squared amplitudes for particle physics and dark matter observables

Marco Palmiotto (m.palmiotto@ip2i.in2p3.fr)

[2211.10376 M.P., A. Arbey, N. F. Mahmoudi]

Université Claude Bernard Lyon 1, France
Institut de Physique des 2 Infinis

Motivations

Problem

How to verify if a given BSM model can describe some dark matter observables

Motivations

Problem

How to verify if a given BSM model can describe some dark matter observables

At the very basis, we need

- Cross sections
- Decay rates
- Matrix elements

Motivations

Problem

How to verify if a given BSM model can describe some dark matter observables

At the very basis, we need

- Cross sections
- Decay rates
- Matrix elements

maybe at 1 loop

Motivations

Problem

How to verify if a given BSM model can describe some dark matter observables

At the very basis, we need

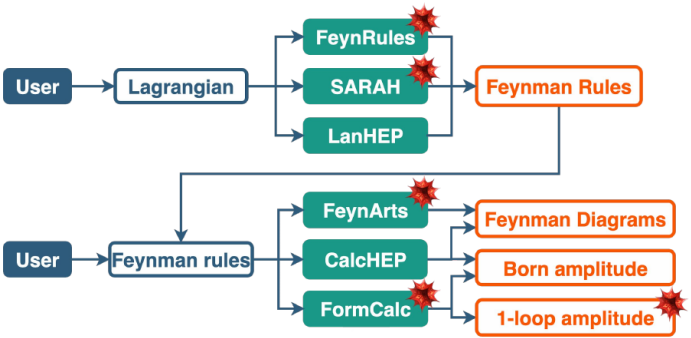
- Cross sections
- Decay rates
- Matrix elements

maybe at 1 loop

To compute

- Relic density
- Direct and indirect detection observables

Some solutions



- Many codes are required
- Several passages of input
- Mathematica dependencies

DarkPack's phylosophy

DarkPACK is conceived to have a **unique** and **modular** workflow

Unique

Lagrangian density $\rightarrow$ amplitudes, ... $\rightarrow$ DM observables

DarkPack's phylosophy

DarkPACK is conceived to have a **unique** and **modular** workflow

Unique

Lagrangian density $\rightarrow$ amplitudes, ... $\rightarrow$ DM observables

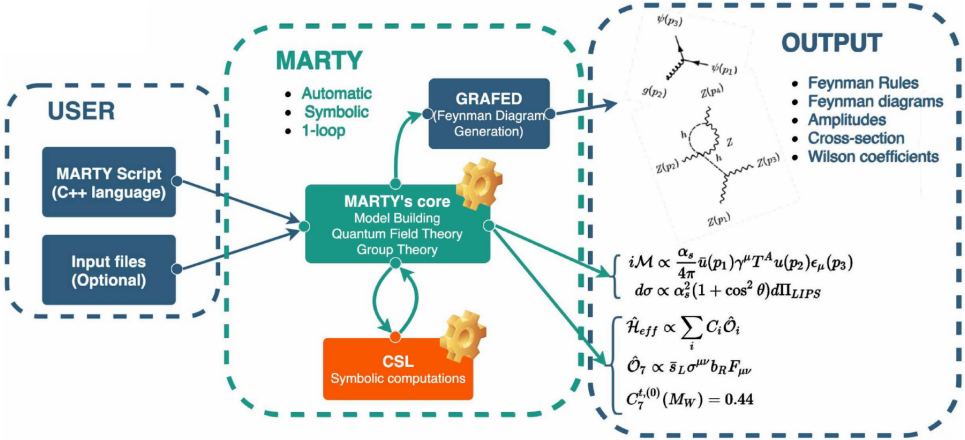
Modular

- Possibility of stopping at any point of the chain...
- ...to link it with external software
- More ease in writing custom functionalities $\leftarrow$ Object-oriented structure

MARTY

website: `https://marty.in2p3.fr`

manual: 2011.02478



MARTY

With MARTY the user can

- Write a Lagrangian symbolically in a C++ source file
  - By defining the **gauge symmetries** of the model

MARTY

With MARTY the user can

- Write a Lagrangian symbolically in a C++ source file
  - By defining the **gauge symmetries** of the model
  - By defining the **fields** of the model

MARTY

With MARTY the user can

- Write a Lagrangian symbolically in a C++ source file
  - By defining the **gauge symmetries** of the model
  - By defining the **fields** of the model
  - By adding **potential** terms

MARTY

With MARTY the user can

- Write a Lagrangian symbolically in a C++ source file
  - By defining the **gauge symmetries** of the model
  - By defining the **fields** of the model
  - By adding **potential** terms
  - By performing **SSB** if that's in the model

MARTY

With MARTY the user can

- Write a Lagrangian symbolically in a C++ source file
  - By defining the **gauge symmetries** of the model
  - By defining the **fields** of the model
  - By adding **potential** terms
  - By performing **SSB** if that's in the model
- **Symbolically** get quantities such as
  - $\sum \overline{|M|^2}, \Gamma$
  - Wilson coefficients
  - Feyman diagrams

MARTY

With MARTY the user can

- Write a Lagrangian symbolically in a C++ source file
  - By defining the **gauge symmetries** of the model
  - By defining the **fields** of the model
  - By adding **potential** terms
  - By performing **SSB** if that's in the model
- **Symbolically** get quantities such as
  - $\sum \overline{|M|^2}, \Gamma$
  - Wilson coefficients
  - Feyman diagrams

  $\rightarrow$ up to 1 loop level
- Output those results in a **numerical** C++ library

Content of DarkPACK - 1

DarkPACK and its documentation can be downloaded at

`https://gitlab.in2p3.fr/darkpack/darkpack-public`

(2211.10376 Palmiotto, Arbey, Mahmoudi)

Content of DarkPACK - 2

The main content of the DarkPACK consists of

- The file `MSSM.cpp` which uses MARTY to build the MSSM and to do the symbolical manipulations

Content of DarkPACK - 2

The main content of the DarkPACK consists of

- The file `MSSM.cpp` which uses MARTY to build the MSSM and to do the symbolical manipulations
- The folder `mssm2to2`, which contains the barebone numerical libary obtained by compiling and running the model file

Content of DarkPACK - 2

The main content of the DarkPACK consists of

- The file `MSSM.cpp` which uses MARTY to build the MSSM and to do the symbolical manipulations
- The folder `mssm2to2`, which contains the barebone numerical libary obtained by compiling and running the model file
- The folder `auxiliary_library`, which contains
  - The model-agnostic functionalities
  - The model-specific functionalities (in `mssm2to2`)
  - The model-specific source code for the programs (in `script_mssm2to2`)

Content of DarkPACK - 2

The main content of the DarkPACK consists of

- The file `MSSM.cpp` which uses MARTY to build the MSSM and to do the symbolical manipulations
- The folder `mssm2to2`, which contains the barebone numerical libary obtained by compiling and running the model file
- The folder `auxiliary_library`, which contains
  - The model-agnostic functionalities
  - The model-specific functionalities (in `mssm2to2`)
  - The model-specific source code for the programs (in `script_mssm2to2`)

Why the MSSM?

- Numerical tests
- Performance check

Setup of DarkPACK

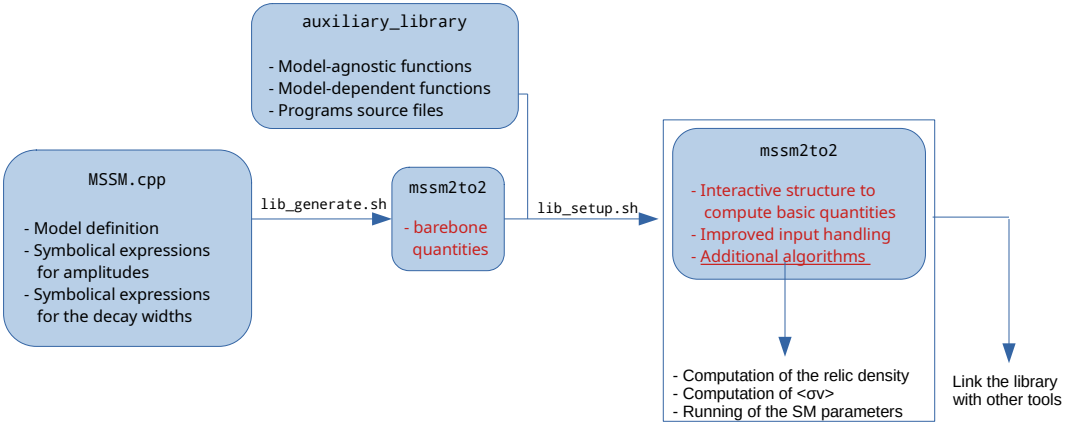
It relies on two script

- `lib_generate.sh` to generate the library
- `lib_setup.sh` to copy the files in `auxiliary_library` in the needed paths and to compile the final library

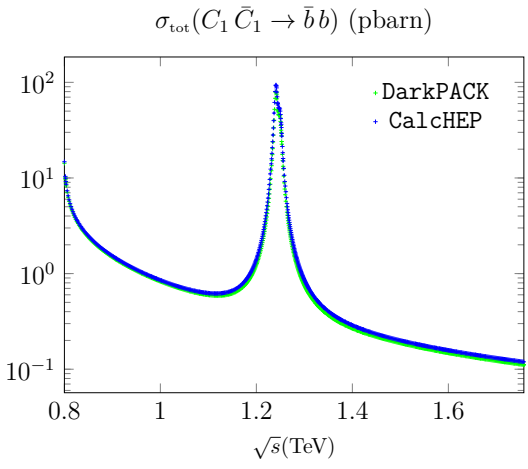
Detailed instructions on the scripts can be found in the `README.md`

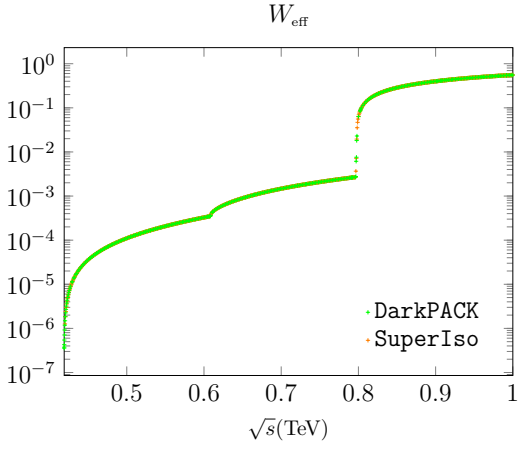
You need to have `MARTY` installed, and define the environmental variable `INSTALLMARTYPATH` as the path where it is built

How it works

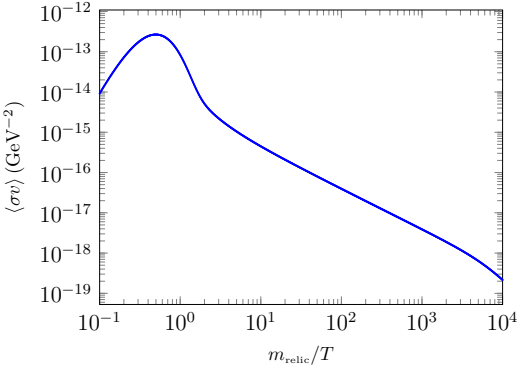

auxiliary_library
- Model-agnostic functions
- Model-dependent functions
- Programs source files
MSSM.cpp
- Model definition
- Symbolical expressions for amplitudes
- Symbolical expressions for the decay widths
lib_generate.sh
mssm2to2
- barebone quantities
lib_setup.sh
mssm2to2
- Interactive structure to compute basic quantities
- Improved input handling
- Additional algorithms
- Computation of the relic density
- Computation of <σv>
- Running of the SM parameters
Link the library with other tools

Some output


$\sigma_{\rm tot}(C_1\,\bar{C}_1 \to \bar{b}\,b)$ (pbarn)
DarkPACK
CalcHEP
10^2
10^1
10^0
10^{-1}
0.8
1
1.2
1.4
1.6
$\sqrt{s}$(TeV)


$W_{\rm eff}$
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
DarkPACK
SuperIso
0.5
0.6
0.7
0.8
0.9
1
$\sqrt{s}$(TeV)

Preliminary results



$$\mathcal{L} \supset -g_\chi \bar{\chi}\chi - \frac{\phi}{v} \sum_{\text{fermions}} g_f m_f \bar{f} f$$

- χ is a Dirac fermion and DM candidate
- ϕ is a scalar mediator

Capabilities

Generating a user-friendly library containing

- $\sum |M|^2$, Γ
- $\langle \sigma v \rangle$ for co-annihilation
- relic density computation

Capabilities

Generating a user-friendly library containing

- $\sum |M|^2$, Γ
- $\langle \sigma v \rangle$ for co-annihilation
- relic density computation

released MSSM

$\surd$ performance

$\surd$ consistency

soon releasing new models

$\surd$ stability

$\surd$ ease of use

Capabilities

Generating a user-friendly library containing

- $\sum |M|^2$, Γ
- $\langle \sigma v \rangle$ for co-annihilation
- relic density computation

released MSSM

$\surd$ performance

$\surd$ consistency

soon releasing new models

$\surd$ stability

$\surd$ ease of use

Performance and portability

- Avoiding global variables
- Parallelization using C++ STL

Present and next development

- Improving the model-agnostic algorithms
- Releasing new models
- Include coscattering
- Solving one Boltzmann equation for every species
- Supporting models with multiple DM candidates
- Native functions to compute direct and indirect detection observables
- Considering more general scenarios, i. e. freeze-in

Conclusions

Today DarkPACK allows to

- Compute $\sum |M|^2$ and Γ at LO in any NP scenario
- Compute $\langle \sigma v \rangle$, Ωh^2 for coannihilation
- Have a library easy to link with other software
- Have a framework portable and performance-oriented

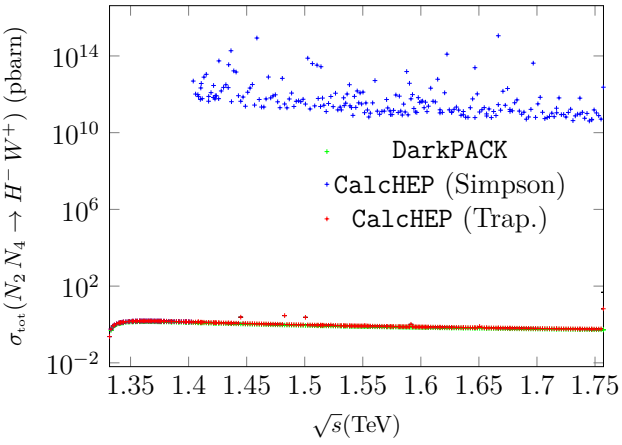
$\rightarrow$ validated in the MSSM

Next, we will

- Validate with a new model and relese it
- Follow the development roadmap
- use DarkPack to see if specific NP models can help to explain DM observables

Thanks for your attention!

Simpson rule vs trapezoidal rule pt. 1



Simpson rule vs trapezoidal rule pt. 2