Probing dark matter models with DarkPACK: present state and future developments

Marco Palmiotto

[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

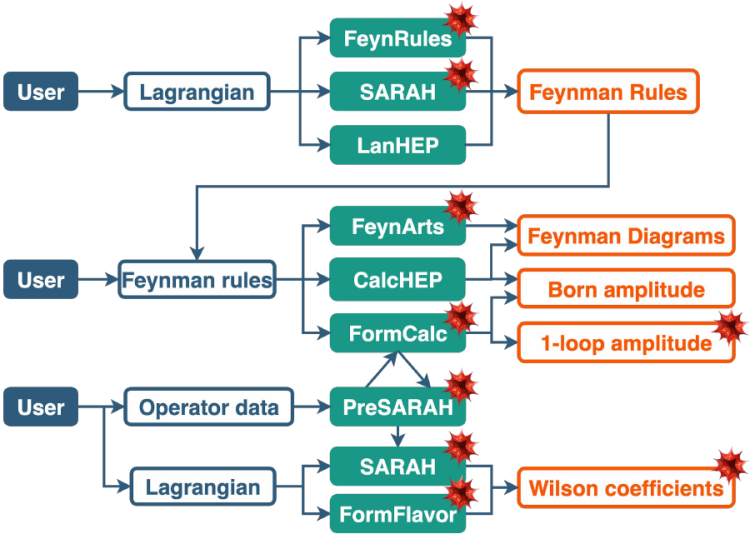
- Cross sections
- Decay rates
- Matrix elements

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

$$\text{Lagrangian density} \rightarrow \underset{\textbf{Unique}}{\text{amplitudes, } \ldots} \rightarrow \text{DM observables}$$

DarkPack's phylosophy

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

Unique

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

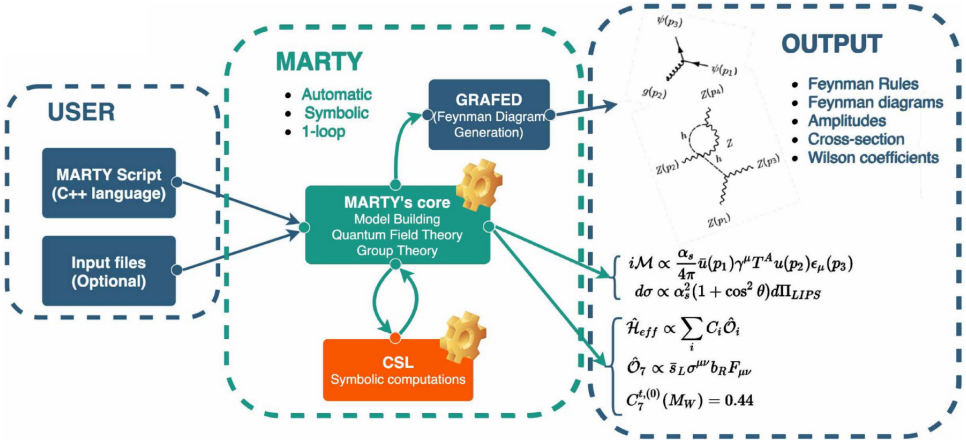
Modular

- 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

- An example **model 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

- An example **model 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

- An example **model 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 **add-ons**: the folder `auxiliary_library` 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

- An example **model 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 **add-ons**: the folder `auxiliary_library` 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 ← existence of many other tools
- Performance check ← lots of particles and Feynman rules

Setup of DarkPACK

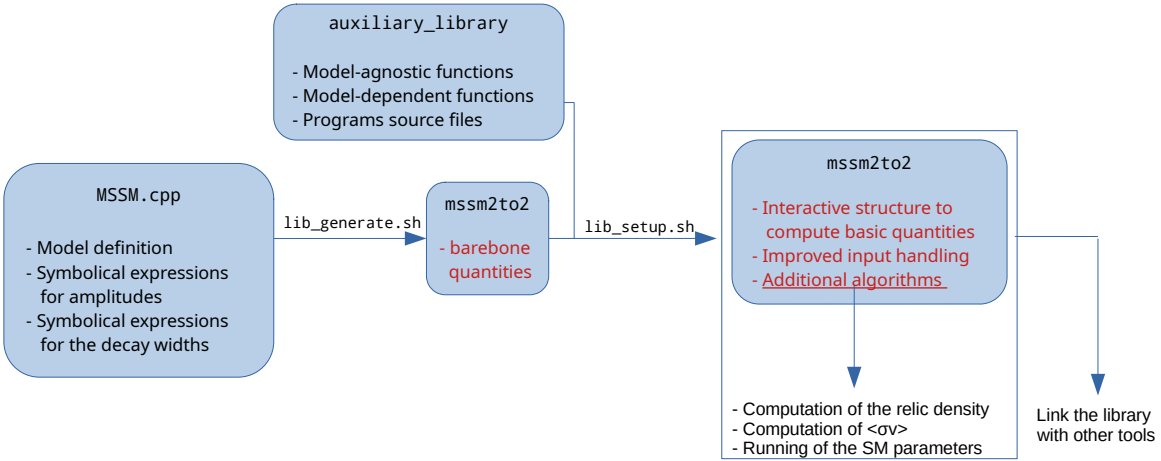
It relies on two script

- `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

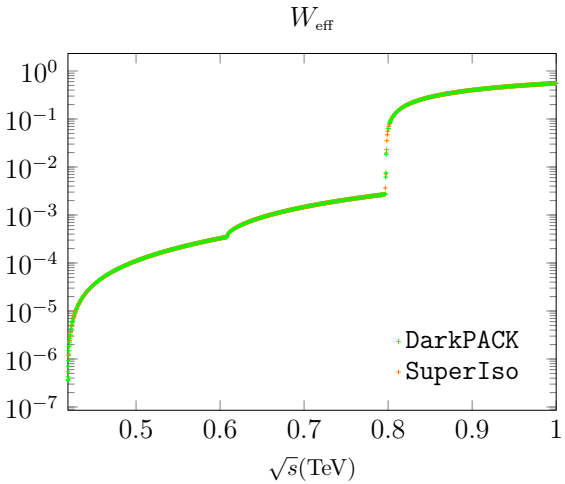
Detailed instructions on the scripts can be found in the `README.md`

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


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

Capabilities

Observables:

- $\sum |M|^2$, Γ $\rightarrow$ up to 1-loop (LO)
- W_{eff}, $\langle \sigma v \rangle$ $\rightarrow$ improved stability at low T
- Ωh^2 $\rightarrow$ from `SuperIso Relic`
  well-tested, reliable in MSSM, NMSSM

Implementation:

- user-friendly
- unique and modular framework
- native parallelisation $\leftarrow$ avoiding global variables
  $\rightarrow$ good portability
- no external dependencies, but the ones of `MARTY`

Capabilities

Observables:

- $\sum |M|^2$, Γ $\rightarrow$ up to 1-loop (LO)
- W_{eff}, $\langle \sigma v \rangle$ $\rightarrow$ improved stability at low T
- Ωh^2 $\rightarrow$ from `SuperIso Relic`
  well-tested, reliable in MSSM, NMSSM

Implementation:

- user-friendly
- unique and modular framework
- native parallelisation $\leftarrow$ avoiding global variables
  $\rightarrow$ good portability
- no external dependencies, but the ones of `MARTY`

released MSSM

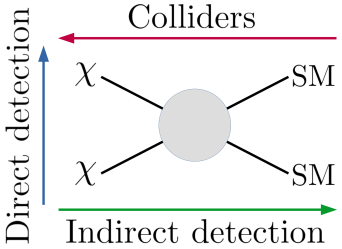
- performance
- consistency

release of new models

- stability
- ease of use

Development roadmap

- Releasing new models
- Improving the model-agnostic algorithms
- More general forms of the Boltzmann equation
  → Solving a system of equations: one for every species
  → Supporting models with multiple DM candidates
  → Considering more general scenarios, i.e. freeze-in
- Native functions for direct searches
  → `MARTY` provides Wilson coefficients
- Native functions for indirect searches
  → required amplitudes already provided
  → already possible to link it with external software
- Improving portability with UFO files
  → such a feature is among the future developments of `MARTY`



Conclusions

Today DarkPACK allows to

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

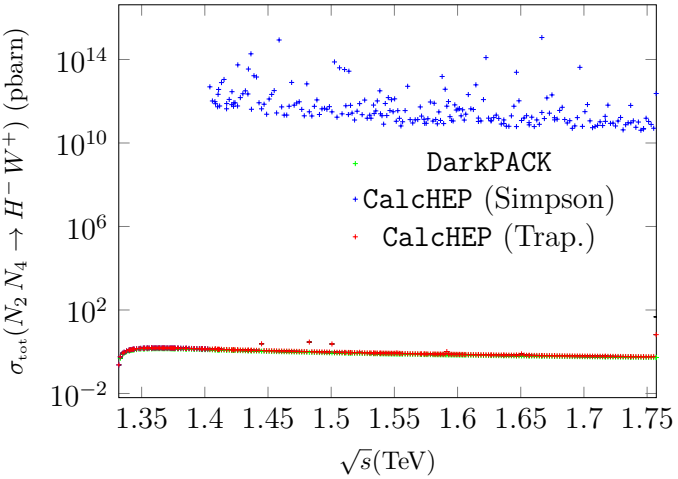
$\rightarrow$ validated in the MSSM

Currently, we are working on the implementation of a new model
Next, we will

- Release the source code for the new model
- Follow the development roadmap
- use DarkPack to verify if specific NP models can help to explain DM observables

Thank you for the attention!

Simpson rule vs trapezoidal rule pt. 1



Simpson rule vs trapezoidal rule pt. 2