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

Marco Palmiotto

Institut de Physique des 2 Infinis
Lyon, France

20 July 2024

DarkPACK: motivations

Problem

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

DarkPACK: 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

DarkPACK: 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

DarkPACK: motivations

Problem

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

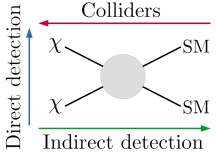
At the very basis, we need:

- Cross sections
- Decay rates
- Matrix elements

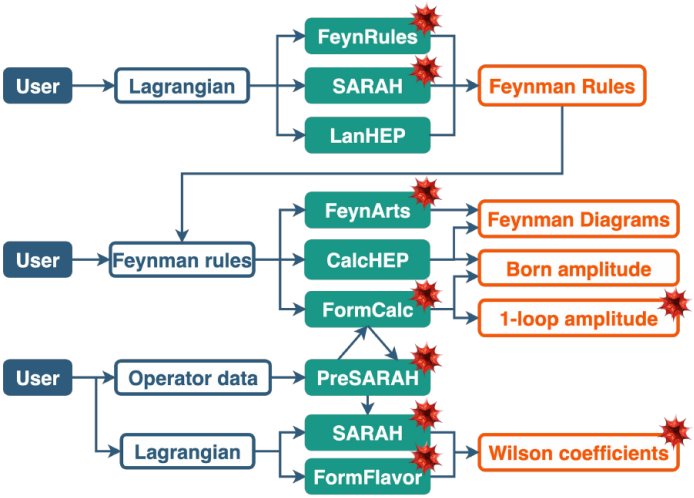
maybe at 1 loop

To compute:

- Relic density
- Direct and indirect detection observables
- Collider observables

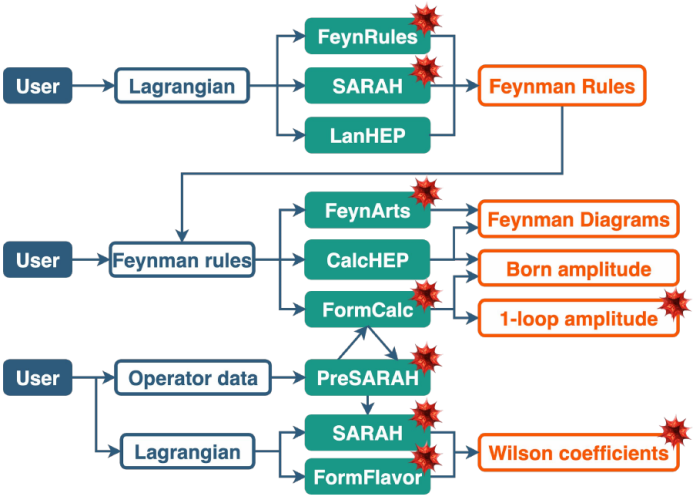


Some solutions



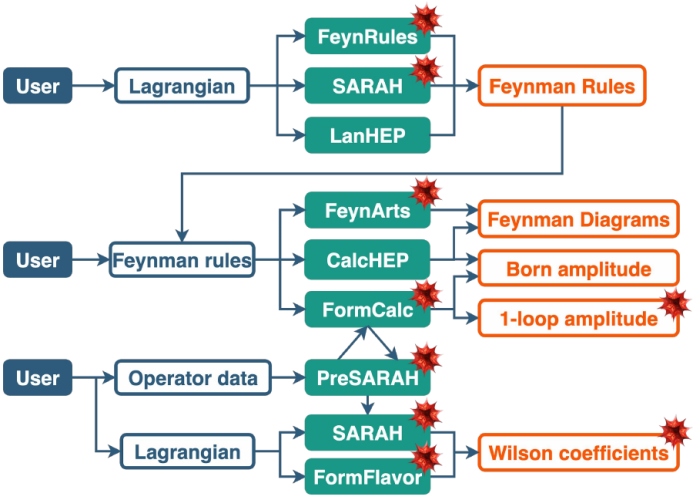
- Many codes are required

Some solutions



- Many codes are required
- Several passages of input

Some solutions



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

DarkPack's philosophy

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

Unique

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

DarkPack's philosophy

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

$$\text{Lagrangian density} \xrightarrow[\textbf{Unique}]{} \text{amplitudes, } \ldots \xrightarrow[\textbf{Modular}]{} \text{DM observables}$$

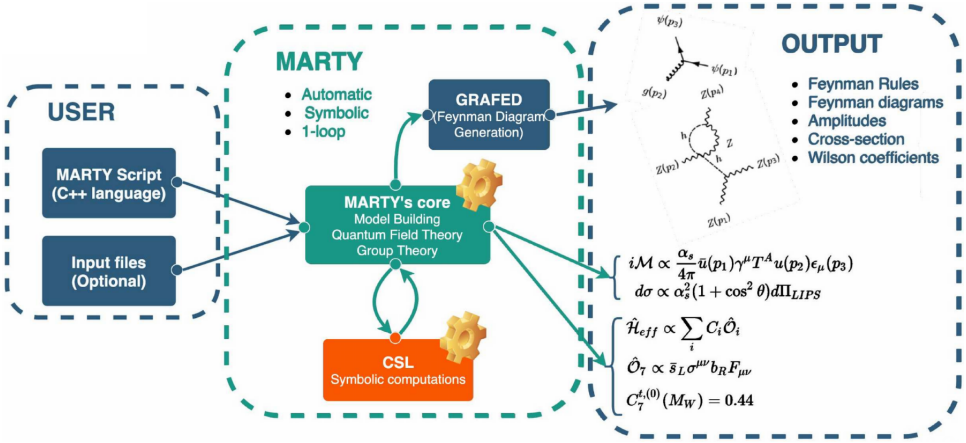
- Possibility of stopping at any point of the chain...
- ...to link it with external software
- More ease in writing custom functionalities $\leftarrow$ Object-oriented structure

References: (M.P., A.Arbey, N.F.Mahmoudi, CPC) user manual
(M.P., thesis) full reference

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 → up to 1 loop level
  - Feyman diagrams
- Output those results in a **numerical** C++ library

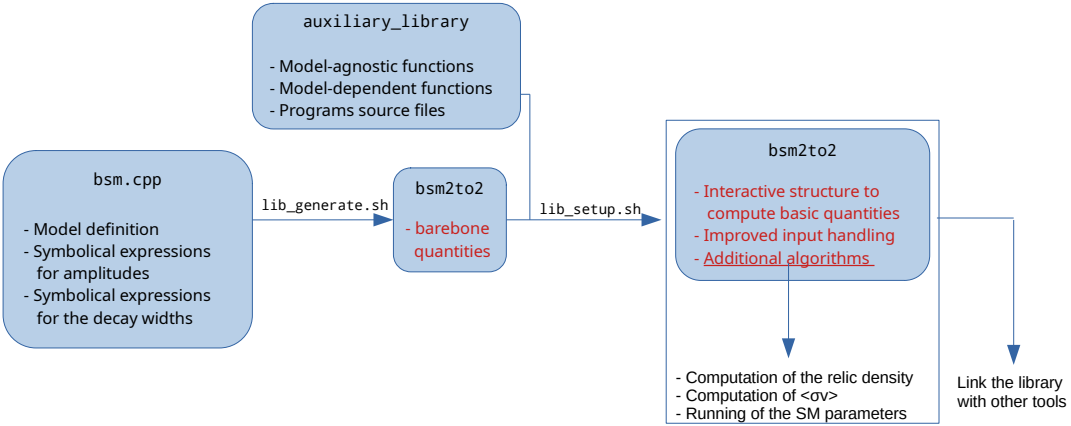
Content of DarkPACK

DarkPACK and its documentation can be downloaded at

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

(2211.10376 M.P., A.Arbey, N.F.Mahmoudi)

How it works


auxiliary_library
- Model-agnostic functions
- Model-dependent functions
- Programs source files
bsm.cpp
- Model definition
- Symbolical expressions for amplitudes
- Symbolical expressions for the decay widths
lib_generate.sh
bsm2to2
- barebone quantities
lib_setup.sh
bsm2to2
- 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

Capabilities

Observables:

- $\sum |M|^2$, Γ $\rightarrow$ @LO if $\leq$ 1-loop
- W_{eff}, $\langle \sigma v \rangle$ $\rightarrow$ improved stability at low T
- Ωh^2 $\rightarrow$ from `SuperIso Relic`
  $\rightarrow$ well-tested, reliable in MSSM, NMSSM
  $\rightarrow$ allows for **modified cosmology**

Capabilities

Observables:

- $\sum |M|^2$, Γ $\rightarrow$ @LO if $\leq$ 1-loop
- W_{eff}, $\langle \sigma v \rangle$ $\rightarrow$ improved stability at low T
- Ωh^2 $\rightarrow$ from `SuperIso Relic`
  $\rightarrow$ well-tested, reliable in MSSM, NMSSM
  $\rightarrow$ allows for **modified cosmology**

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 $\rightarrow$ @LO if $\leq$ 1-loop
- W_{eff}, $\langle \sigma v \rangle$ $\rightarrow$ improved stability at low T
- Ωh^2 $\rightarrow$ from `SuperIso Relic`
  $\rightarrow$ well-tested, reliable in MSSM, NMSSM
  $\rightarrow$ allows for **modified cosmology**

Implementation:

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

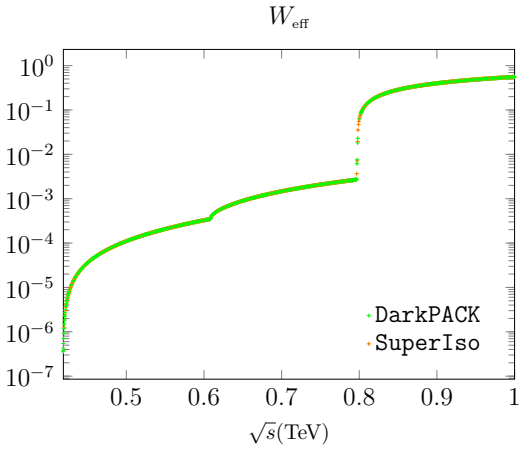
released MSSM
and "scalar" model

- performance
- consistency

release of new models

- stability
- ease of use

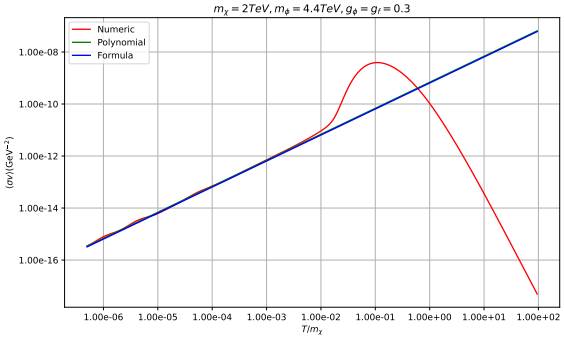
Example: W_{eff} in the MSSM



Example: $\langle \sigma v \rangle$ in the “scalar” model

$$\mathcal{L} \supset -g_\chi \phi \bar{\chi} \chi + \sum_{f \in \{\text{SMfermions}\}} \frac{y_f}{\sqrt{2}} g_f \phi \bar{f} f$$

- ϕ parity-even scalar mediator
- χ Dirac fermion



Questions we would like to answer

- What is the impact of **modified cosmology** on DM relic density?
- Can sampling the parameter space be easier if we know more about each species' abundance?
- How much do thermodynamical assumption influence the DM abudance?

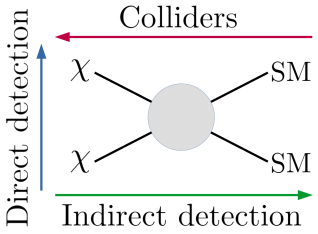
Questions we would like to answer

- What is the impact of **modified cosmology** on DM relic density?
- Can sampling the parameter space be easier if we know more about each species' abundance?
- How much do thermodynamical assumption influence the DM abudance?

$$\dot{n}_i + 3Hn_i = -\sum_{j=1}^{N}\sum_{a,b}\left[\langle\sigma v_{\text{Møl}}\rangle_{ij\to ab} n_i n_j - \langle\sigma v_{\text{Møl}}\rangle_{ab\to ij} n_a n_b\right] +$$
$$-\sum_{j\neq i}\sum_{a,b}\left[\langle\sigma v_{\text{Møl}}\rangle_{ia\to jb} n_i n_a - \langle\sigma v_{\text{Møl}}\rangle_{jb\to ia} n_j n_b\right] + \qquad \Rightarrow \quad \dot{n} + 3Hn = \langle\sigma v_{\text{Møl}}\rangle\left(n^2 + n_{\text{eq}}^2\right)$$
$$-\sum_{j\neq i}\sum_{a,b}\left[\langle\Gamma_{i\to jab}\rangle\left(n_i - n_i^{\text{eq}}\right) - \langle\Gamma_{j\to iab}\rangle\left(n_j - n_j^{\text{eq}}\right)\right]$$

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 $\langle \sigma v \rangle$, Ωh^2 for coannihilation
- Have a library easy to link with other software
- Have a framework portable and performance-oriented

Conclusions

Today DarkPACK allows to:

- Compute $\sum |M|^2$ and Γ at LO in many NP scenarios
- 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

Conclusions

Today DarkPACK allows to:

- Compute $\sum |M|^2$ and Γ at LO in many NP scenarios → validated in the MSSM
- Compute $\langle \sigma v \rangle$, Ωh^2 for coannihilation → validated in the “scalar” model
- Have a library easy to link with other software
- Have a framework portable and performance-oriented

Conclusions

Today DarkPACK allows to:

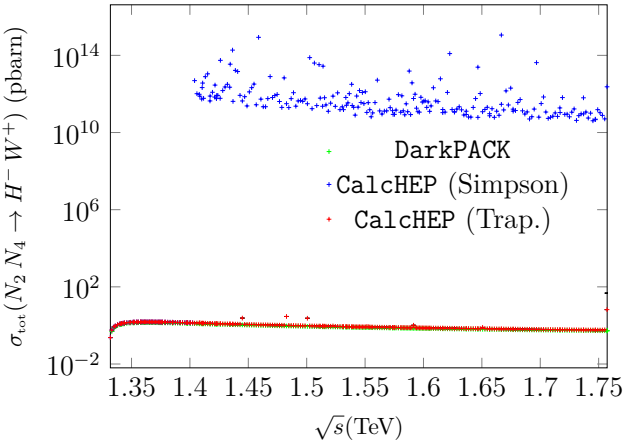
- Compute $\sum |M|^2$ and Γ at LO in many NP scenarios
- Compute $\langle \sigma v \rangle$, Ωh^2 for coannihilation
- Have a library easy to link with other software
- Have a framework portable and performance-oriented

→ validated in the MSSM
→ validated in the “scalar” model

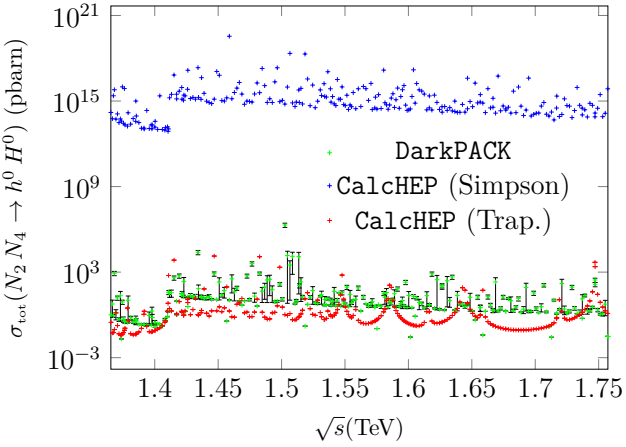
Work in progress: a model with an horizontal flavour symmetry

Thank you for the attention!

Simpson rule vs trapezoidal rule pt. 1



Simpson rule vs trapezoidal rule pt. 2



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

Current features of SuperIso Relic

The current version of the software allows the followings

- Considering **non-thermal** production of **DM**
- Considering **entropy injection**
- Considering **variable dark energy**
- Some freedom in the choice of the **QCD equation of state** by modifying the lattice parameters
- Considering MSSM and NMSSM
- Following the evolution of the density of only the LSP (lightest supersymmetric particle)

Limitations of MARTY

- It works in 4D-Minkowski spacetime
  $\rightarrow$ need to reduce to 4D multi-dimensional theories
- It does not support spin 2 and spin 3/2 fields
- It is not able to do non-perturbative calculations