Hands on DarkPACK: next upgrade and future goals

Marco Palmiotto

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

07/10/2024

Where are we?

- DarkPACK’s primary goal is to compute the observables related to **dark matter** from a given model

Where are we?

- DarkPACK’s primary goal is to compute the observables related to **dark matter** from a given model
- This includes relic density, direct and indirect detection, and collider observables

Where are we?

- DarkPACK's primary goal is to compute the observables related to **dark matter** from a given model
- This includes relic density, direct and indirect detection, and collider observables
- Guiding principles: user-friendliness, ease of implementation of new models

Where are we?

- DarkPACK's primary goal is to compute the observables related to **dark matter** from a given model
- This includes relic density, direct and indirect detection, and collider observables
- Guiding principles: user-friendliness, ease of implementation of new models
- How we made it: upgrading the numerical library of MARTY

Next upgrade: new build system

handmade scripts → **cmake**

Next upgrade: new build system

handmade scripts → **cmake**

What's been done so far:

- proving a **devcontainer**
  → no need to install MARTY locally

Next upgrade: new build system

handmade scripts → **cmake**

What’s been done so far:

- proving a **devcontainer**
  → no need to install MARTY locally
- now it is possible to use **multiple models** at once

Next upgrade: new build system

handmade scripts → **cmake**

What's been done so far:

- proving a **devcontainer**
  → no need to install MARTY locally
- now it is possible to use **multiple models** at once
- easier development and usage thanks to the possibility of using debuggers in IDEs

Next upgrade: new build system

handmade scripts → **cmake**

What's been done so far:

- proving a **devcontainer**
  → no need to install MARTY locally
- now it is possible to use **multiple models** at once
- easier development and usage thanks to the possibility of using debuggers in IDEs

What's left to do:

- MARTY as git **submodule**
  → easy to install locally
- Implement **unitary tests**

Next upgrades

Models' parameter scan:

Next upgrades

Models' parameter scan:

- new input handling improved for scans

Next upgrades

Models' parameter scan:

- new input handling improved for scans
- example files for the *scalar* model

Next upgrades

Models' parameter scan:

- new input handling improved for scans
- example files for the *scalar* model
- release of the $SU(2)$ flavour model as a real-case example

Next upgrades

Models' parameter scan:

- new input handling improved for scans
- example files for the *scalar* model
- release of the $SU(2)$ flavour model as a real-case example

Performance upgrades:

Next upgrades

Models' parameter scan:

- new input handling improved for scans
- example files for the *scalar* model
- release of the $SU(2)$ flavour model as a real-case example

Performance upgrades:

- **unitary tests**

Next upgrades

Models' parameter scan:

- new input handling improved for scans
- example files for the *scalar* model
- release of the $SU(2)$ flavour model as a real-case example

Performance upgrades:

- **unitary tests**
- **numerical precision** optimisation
  → non-convergent integrals problem
- **memory** profiling optimizations
- **parallelisation** on CPU with SYCL and openmp

N2 upgrades

Within the GAMBIT collaboration:

- Embedding MARTY and DarkPACK into GUM
  → support for GUM model files for import/export of a model

N2 upgrades

Within the GAMBIT collaboration:

- Embedding MARTY and DarkPACK into GUM
  → support for GUM model files for import/export of a model
- NLO amplitudes' calculation in MARTY
  → implementation of RGEs

N2 upgrades

Within the GAMBIT collaboration:

- Embedding MARTY and DarkPACK into GUM
  → support for GUM model files for import/export of a model
- NLO amplitudes' calculation in MARTY
  → implementation of RGEs
- library format file for MARTY and DarkPACK
  → "dynamical" management of the libraries

N3 upgrades

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

- Developing a temperature-dependent approach
- Implementing the algorithm for the equation
  → Dropping the mutual kinetic equilibrium hypothesis
  → Studying freeze-in scenarios
  → Studying the evolution of the density of each species separately

Conclusions

- There are lots of goals for DarkPACK, in the years to come
- Scans on the scalar model are the current goal, to pass then to the $SU(2)$ flavour model immediately next
- User-friendliness remains a key point for the development
- Upgrades will involve all aspects of the code:
    - Performance
    - Features
    - Refactoring

Thank you for the attention

Models

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

The $SU(2)$ flavour model:

$$\mathcal{L} \supset i\overline{\chi}(\partial\!\!\!/ + iT^a \tilde{W}^\mu_a \gamma_\mu)\chi \; + \; M_\chi \bar{\chi}\chi$$