Entities in scientific workflows (`mammos-entity`)

Hans Fangohr

Professor of Computational Modelling
Max Planck Institute for the Structure and Dynamics of Matter, Hamburg, Germany
University of Southampton, Southampton, United Kingdom
https://fangohr.github.io

Funded by
the European Union

Outline

1. Introduction
2. Quantities: units in software (`astropy.units` / `mammos_units`)
3. Entities: more metadata with `mammos-entity`
4. Workflow: from raw data to self-explaining data in file
5. Summary

1. Introduction

One line summary

Could someone else understand your saved data without asking you about it?

One line summary

Could someone else understand your saved data without asking you about it?

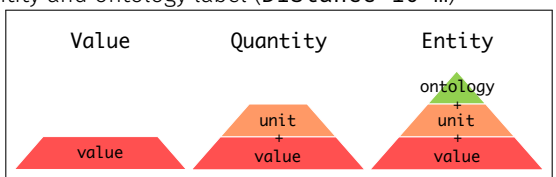
Often, there is metadata missing.

Session goals

1. At the end of this session you should know about the following terms:
   - value (`10`)
   - unit (`m`)
   - quantity = value and unit (`10 m`)
   - ontology label (`Distance`)
   - entity = quantity and ontology label (`Distance 10 m`)

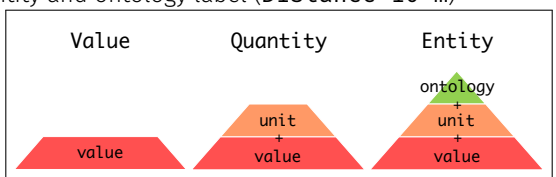
Session goals

1. At the end of this session you should know about the following terms:
   - value (`10`)
   - unit (`m`)
   - quantity = value and unit (`10 m`)
   - ontology label (`Distance`)
   - entity = quantity and ontology label (`Distance 10 m`)



Session goals

1. At the end of this session you should know about the following terms:
   - value (`10`)
   - unit (`m`)
   - quantity = value and unit (`10 m`)
   - ontology label (`Distance`)
   - entity = quantity and ontology label (`Distance 10 m`)



2. You understand in principle that you can use software tools to:
   - convert values with units (`10 m` → `0.01 km`) with `astropy.units`
   - create entities using `mammos-entity`
   - create entity collections
   - save entity collections to files

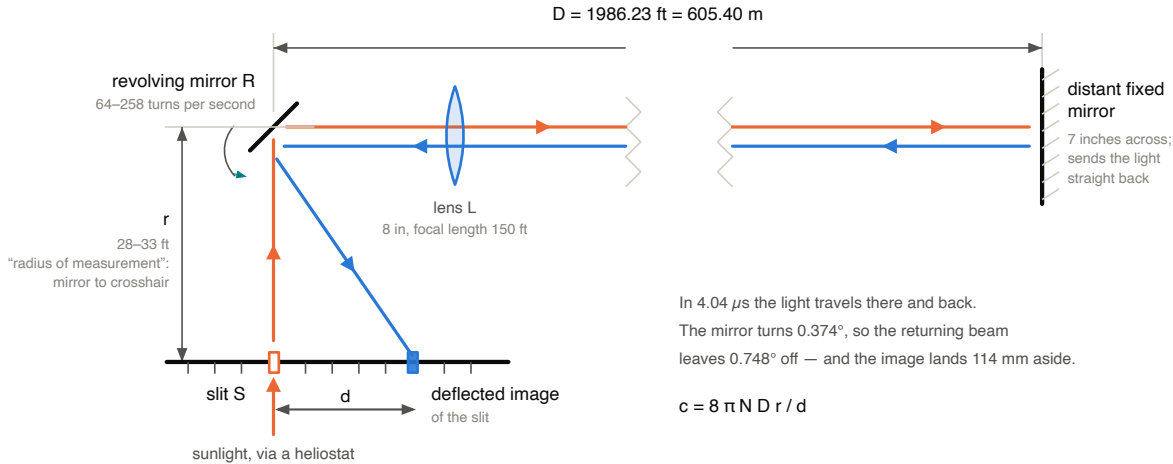
A data file

Raw data from a study by Michelson to determine the speed of light *c* (published 1879, Gutenberg#11753)

```
257.36  28.672  114.55  0.99614
257.52  28.655  114.56  0.99614
257.52  28.647  114.50  0.99614
193.14  28.647   85.84  0.99598
193.14  28.650   85.89  0.99598
257.42  28.650  114.53  0.99614
257.39  28.658  114.47  0.99614
257.39  28.658  114.46  0.99614
257.39  28.662  114.47  0.99614
257.39  28.660  114.50  0.99614
257.39  28.678  114.53  0.99614
257.29  28.685  114.52  0.99614
257.29  28.685  114.54  0.99614
257.45  28.690  114.74  0.99614
257.45  28.690  114.70  0.99614
257.45  28.690  114.68  0.99614
257.49  28.172  112.56  0.99614
257.49  28.172  112.56  0.99614
257.49  28.172  112.57  0.99614
257.42  28.178  112.56  0.99614
257.42  28.178  112.56  0.99614
```

Experiment Michelson 1879 (speed of light)

Plan view, looking down. Angles are exaggerated; the real deflection is 0.75°.



In 4.04 μs the light travels there and back.

The mirror turns 0.374°, so the returning beam leaves 0.748° off — and the image lands 114 mm aside.

$$c = 8 \pi N D r / d$$

Reproduce key result from raw data

Raw data from a study by Michelson (from 1879, Gutenberg#11753)

Task: use the data to reproduce the key result (speed of light c).

257.36	28.672	114.55	0.99614

Reproduce key result from raw data

Raw data from a study by Michelson (from 1879, Gutenberg#11753)

Task: use the data to reproduce the key result (speed of light c).

257.36	28.672	114.55	0.99614

- raw data not sufficient to reproduce key result

Reproduce key result from raw data

Raw data from a study by Michelson (from 1879, Gutenberg#11753)

Task: use the data to reproduce the key result (speed of light c).

```
257.36   28.672   114.55   0.99614
```

Units

#	unit
1	revolutions per second
2	feet
3	turns of a micrometer screw
4	millimetres

Reproduce key result from raw data

Raw data from a study by Michelson (from 1879, Gutenberg#11753)

Task: use the data to reproduce the key result (speed of light c).

```
257.36   28.672   114.55   0.99614
```

Meaning and Units

#	meaning	unit
1	rotation rate of a revolving mirror	revolutions per second
2	distance from mirror to scale	feet
3	screw turns from slit to deflected image	turns of a micrometer screw
4	how far one turn of that screw moves the crosshair	millimetres

Reproduce key result from raw data

Raw data from a study by Michelson (from 1879, Gutenberg#11753)

Task: use the data to reproduce the key result (speed of light c).

```
257.36   28.672   114.55   0.99614
```

Name, meaning and Units

#	name	meaning	unit
1	N	rotation rate of a revolving mirror	revolutions per second
2	r	distance from mirror to scale	feet
3	turns	screw turns from slit to deflected image	turns of a micrometer screw
4	pitch	how far one turn of that screw moves the crosshair	millimetres

Reproduce key result from raw data

Raw data from a study by Michelson (from 1879, Gutenberg#11753)

Task: use the data to reproduce the key result (speed of light *c*).

```
257.36   28.672   114.55   0.99614
```

Name, meaning and Units

#	name	meaning	unit
1	N	rotation rate of a revolving mirror	revolutions per second
2	r	distance from mirror to scale	feet
3	turns	screw turns from slit to deflected image	turns of a micrometer screw
4	pitch	how far one turn of that screw moves the crosshair	millimetres
5	d	derived: separation of slit and deflected image	
6	c	derived: speed of light	

Strategy to reproduce key result of paper

- 100 rows with data
- Convert raw numbers into SI units
- Compute derived entities $d = \texttt{turns} * \texttt{pitch}$ and $c = 8\pi NDr/d$
- Use constant $D = 1986.23$ feet, separation of revolving and fixed mirror (from paper)
- Average the 100 values for c

Reproduce the key result of the paper

Demo in notebook `1-compute-speed-from-entities.ipynb` (interactive notebook on binder)

- Use a data file that is metadata rich (`data/michelson-1879-entities.csv`)
- created with `mammos-entity`
- Quick demo to show approach — not explained yet

Reproduce the key result of the paper

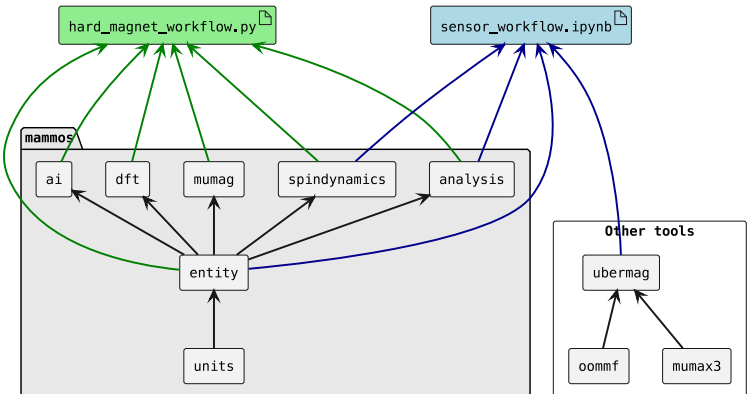
Demo in notebook `1-compute-speed-from-entities.ipynb` (interactive notebook on binder)

- Use a data file that is metadata rich (`data/michelson-1879-entities.csv`)
- created with `mammos-entity`
- Quick demo to show approach — not explained yet

- ✓ Metadata includes
  - units
  - (ontology) labels & descriptions
- ✓ Automatic execution of calculations with units
- ✓ Compatibility with (Python) data analysis ecosystem (`pandas`)

MaMMoS (MAgnetic Multiscale MOdelling Suite) project

EU-funded MaMMoS project produces open source software [1]:



Today we focus on `mammos-units` (≈ `astropy.units`) and `mammos-entity` only.

[1] "Towards a MAgnetic Multiscale MOdelling Suite (MaMMoS)," 10.1109/TMAG.2026.3689962 (2026).

- https://mammos-project.github.io/mammos
- https://mammos-project.github.io/mammos/examples

Outlook

- ☒ `1-compute-speed-from-entities.ipynb`

  Given saved entity data from the Michelson experiment, we compute derived entities.
- ☐ `2-units.ipynb` — Introduction to computing with units.
- ☐ `3-entities.ipynb` — Introduction to creating and using an entity.
- ☐ `3-entity-collection.ipynb` — Introduction to entity collections (allows saving and reading entity data).
- ☐ `4-raw-to-entitycollection.ipynb` — Notebook that shows how to convert raw numbers into entity data (creates the data file that is used by notebook `1-compute-speed-from-entities.ipynb`).

Outlook

- ☒ `1-compute-speed-from-entities.ipynb`
  Given saved entity data from the Michelson experiment, we compute derived entities.
- ☐ `2-units.ipynb` — Introduction to computing with units.
- ☐ `3-entities.ipynb` — Introduction to creating and using an entity.
- ☐ `3-entity-collection.ipynb` — Introduction to entity collections (allows saving and reading entity data).
- ☐ `4-raw-to-entitycollection.ipynb` — Notebook that shows how to convert raw numbers into entity data (creates the data file that is used by notebook `1-compute-speed-from-entities.ipynb`).

Electronic versions of the notebooks are available from `https://s.gwdg.de/ptjpe4`.

2. Quantities: units in software (`astropy.units` / `mammos_units`)

Python tools — roadmap

- Python 3
- `numpy`: process vectors, matrices, tensors
- `pandas`: dealing with tabular data
- `astropy.units`: dealing with units
- `jupyter` notebook: interactive document in browser
- `mammos-entity`: ontology labels, entity collections, storing data sets

Installation `mammos-entity`

Pixi:

- install pixi (from `https://pixi.prefix.dev/latest/installation/`)
- find suitable subdirectory
- `pixi init .` # create pixi installation in this directory
- `pixi add astropy jupyterlab mammos-entity`
- `pixi shell` # activate this pixi environment
- `jupyter-lab`

Advanced (venv):

- assume you have python installed already
- `python -m venv entities` # create virtual environment
- `source entities/bin/activate` # activate virtual environment
- `pip install astropy jupyterlab mammos-entity`
- `jupyter-lab`

Calculating with units: Astropy.units (mammos-units)

Demo in notebook `2-units.ipynb` (interactive notebook on binder)

Summary units

Lessons units

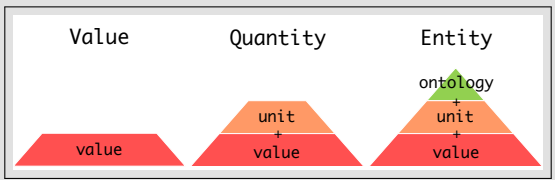
- software can compute with units
- avoids (human) mistakes
- and thus speeds up data processing

Summary units

Lessons units

- software can compute with units
- avoids (human) mistakes
- and thus speeds up data processing

What next?



3. Entities: more metadata with `mammos-entity`

Two ways to capture metadata – and why both are hard

Human-readable notes and descriptions: a paper or README.md

- easy to start with,
- but the author and data generator may forget to write down all context

Two ways to capture metadata – and why both are hard

Human-readable notes and descriptions: a paper or README.md

- easy to start with,
- but the author and data generator may forget to write down all context

Machine-readable data: Structured metadata following a standard

- can be checked and processed automatically.
- examples for metadata standards: NeXus files, CIF, email headers, ...
- but ...need to agree on a standard!

Ontologies for materials science provide a metadata standard

EMMO — Elementary Multiperspective Material Ontology

- a top-level ontology for materials, models, characterisation, data, and units.
- `https://emmo-repo.github.io`

Ontologies for materials science provide a metadata standard

EMMO — Elementary Multiperspective Material Ontology

- a top-level ontology for materials, models, characterisation, data, and units.
- `https://emmo-repo.github.io`

MagMO — Magnetic Materials Ontology

- extends EMMO with concepts specific to magnetic materials, e.g. `SpontaneousMagnetization`, `CurieTemperature`.
- `https://github.com/MaMMoS-project/MagneticMaterialsOntology`

Ontologies for materials science provide a metadata standard

EMMO — Elementary Multiperspective Material Ontology

- a top-level ontology for materials, models, characterisation, data, and units.
- `https://emmo-repo.github.io`

MagMO — Magnetic Materials Ontology

- extends EMMO with concepts specific to magnetic materials, e.g. `SpontaneousMagnetization`, `CurieTemperature`.
- `https://github.com/MaMMoS-project/MagneticMaterialsOntology`

`mammos-entity`

- Entities from EMMO and MagMO are available through the `mammos-entity` Python package.
- provides additional free-text descriptions of entities
- supports reading and writing of collections of entities

mammos-entity: demo entity

Demo in `3-entities.ipynb` (interactive notebook on binder)

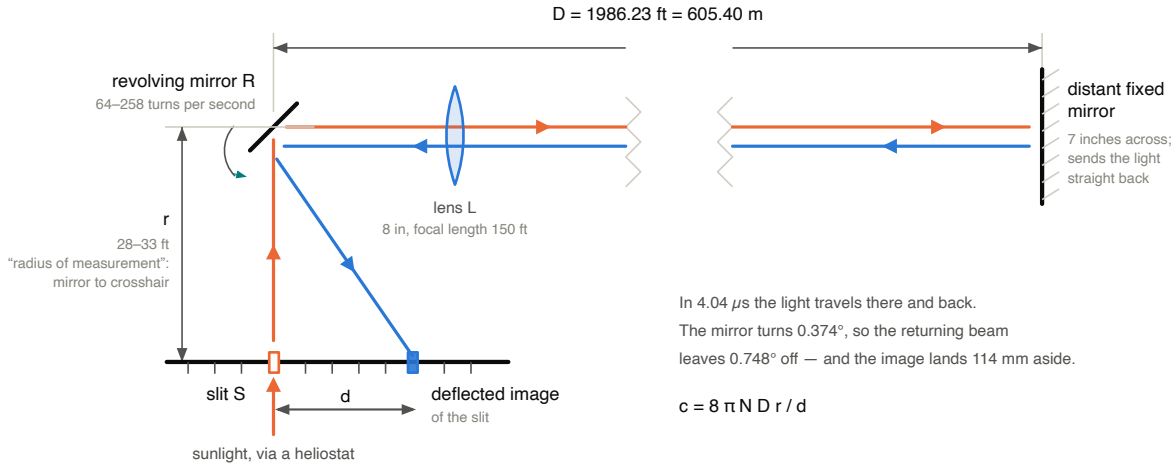
mammos-entity: demo entity collection

Demo in `3-entity-collection.ipynb` (interactive notebook on binder)

4. Workflow: from raw data to self-explaining data in file

Experiment Michelson 1879 (speed of light)

Plan view, looking down. Angles are exaggerated; the real deflection is 0.75°.



In 4.04 μs the light travels there and back.

The mirror turns 0.374°, so the returning beam leaves 0.748° off — and the image lands 114 mm aside.

$$c = 8 \pi N D r / d$$

Raw data (`michelson-1879.dat`)

100 data points:

```
257.36  28.672  114.55  0.99614
257.52  28.655  114.56  0.99614
257.52  28.647  114.50  0.99614
193.14  28.647   85.84  0.99598
193.14  28.650   85.89  0.99598
257.42  28.650  114.53  0.99614
257.39  28.658  114.47  0.99614
257.39  28.658  114.46  0.99614
257.39  28.662  114.47  0.99614
```

In Notebook (michelson-1879.dat)

```
import pandas as pd
pd.read_csv("data/michelson-1879.dat", sep=r"\s+", header=None)
```

	0	1	2	3
0	257.36	28.672	114.55	0.99614
1	257.52	28.655	114.56	0.99614
2	257.52	28.647	114.50	0.99614
3	193.14	28.647	85.84	0.99598
4	193.14	28.650	85.89	0.99598
...	...	...	...	...
95	257.28	33.311	133.00	0.99627
96	192.95	33.205	99.45	0.99606
97	128.63	33.205	66.34	0.99586
98	96.48	33.205	49.76	0.99580
99	64.32	33.205	33.17	0.99574

100 rows × 4 columns

Data with metadata: the entity collection (`michelson-1879-entities.csv`)

```
import mammos_entity as me
data = me.from_csv("data/michelson-1879-entities.csv")
data
```

▼ **EntityCollection** 5 items · N, r, screw_turns, ...

Michelson's 1879 determination of the speed of light: the instrument readings.

A. A. Michelson, "Experimental Determination of the Velocity of Light", Astronomical Papers of the American Ephemeris 1 (1880) 109-145. 100 observations made at the U.S. Naval Academy, Annapolis, 5 June - 2 July 1879. Transcribed via the R package loon.data (michelson_1879); see also MacKay and Oldford, Statistical Science 15(3) 254-278 (2000), doi:10.1214/ss/1009212817.

*Rotating-mirror method. Light travels from the revolving mirror to a fixed mirror and back while the mirror turns; the returned beam is deflected by twice the mirror rotation and read at radial distance r as a displacement d, giving c = 8 * pi * N * D * r / d*

The apparatus constant D is the separation of the revolving and the fixed mirror. It is not part of the observation table and is recorded only here: D = 1986.23 ft = 605.4029 m

The revolving mirror rotates with frequence N. The speed of light is not stored in this file: it is derived from these readings. Note that the light travelled in air, not in vacuum.

N **Frequency** · shape=(100,)

Rotation rate of the revolving mirror, in revolutions per second, timed against a tuning fork.

Data with metadata: as a dataframe (michelson-1879-entities.csv)

```python
import mammos_entity as me
data = me.from_csv("data/michelson-1879-entities.csv")
data.to_dataframe(include_units=True)
```

	N (Hz)	r (m)	screw_turns	screw_pitch (m)	d (m)
0	257.36	8.739226	114.55	0.000996	0.114108
1	257.52	8.734044	114.56	0.000996	0.114118
2	257.52	8.731606	114.50	0.000996	0.114058
3	193.14	8.731606	85.84	0.000996	0.085495
4	193.14	8.732520	85.89	0.000996	0.085545
...	...	...	...	...	...
95	257.28	10.153193	133.00	0.000996	0.132504
96	192.95	10.120884	99.45	0.000996	0.099058
97	128.63	10.120884	66.34	0.000996	0.066065
98	96.48	10.120884	49.76	0.000996	0.049551
99	64.32	10.120884	33.17	0.000996	0.033029

100 rows × 5 columns

How to create the entity file from the raw data?

How to create `data/michelson-1879-entities.csv` from `data/michelson-1879.dat`?

Demo: `4-raw-to-entitycollection.ipynb` (interactive notebook on binder)

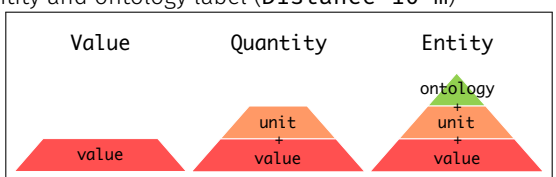
5. Summary

Revisit: Session goals

1. At the end of this session you should know about the following terms:
   - value (`10`)
   - unit (`m`)
   - quantity = value and unit (`10 m`)
   - ontology label (`Distance`)
   - entity = quantity and ontology label (`Distance 10 m`)

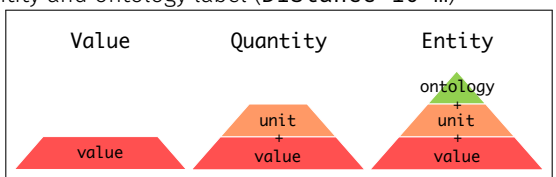
Revisit: Session goals

1. At the end of this session you should know about the following terms:
   - value (`10`)
   - unit (`m`)
   - quantity = value and unit (`10 m`)
   - ontology label (`Distance`)
   - entity = quantity and ontology label (`Distance 10 m`)



Revisit: Session goals

1. At the end of this session you should know about the following terms:
   - value (`10`)
   - unit (`m`)
   - quantity = value and unit (`10 m`)
   - ontology label (`Distance`)
   - entity = quantity and ontology label (`Distance 10 m`)



2. You understand in principle that you can use software tools to:
   - convert values with units (`10 m` → `0.01 km`) with `astropy.units`
   - create entities using `mammos-entity`
   - create entity collections
   - save entity collections to files

Take home messages

- Record any number with units
- Units should *participate in calculations* (also in software)
- *Metadata* in entities provide additional context
  - *Ontology labels* state what a value physically *means*
  - Entities can catch unit mistakes automatically
  - *Descriptions* can provide more detail than ontology labels
- Entity collections can save whole data sets with metadata

Take home messages

- Record any number with units
- Units should *participate in calculations* (also in software)
- *Metadata* in entities provide additional context
  - *Ontology labels* state what a value physically *means*
  - Entities can catch unit mistakes automatically
  - *Descriptions* can provide more detail than ontology labels
- Entity collections can save whole data sets with metadata

Further reading

- this lecture introduced basic ideas (only)
- use documentation, in particular for more special and advanced cases:
  `https://mammos-project.github.io/mammos/examples/mammos-entity`

Common workflow when taking measurements / post-processing data

- start with raw data (such as `michelson-1879.dat`)
- attach units ($\rightarrow$ `Quantity`)
- add metadata ($\rightarrow$ `Entity`)
- compute derived quantities (such as d and/or c)
- represent data set/experiment as *entity collection*
- save to disk (CSV, YAML, HDF5)

Entity gotchas

- what *ontology label* is correct?
  - use `me.search_labels()` and LLM

Entity gotchas

- what *ontology label* is correct?
  - use `me.search_labels()` and LLM
- cannot compute with entities (because the ontology label might be wrong)
  - use `.quantity` for calculations (or `.q`)

Entity gotchas

- what *ontology label* is correct?
  - use `me.search_labels()` and LLM
- cannot compute with entities (because the ontology label might be wrong)
  - use `.quantity` for calculations (or `.q`)
- merging / extending entity collections
  - split entity collection into metadata and pandas dataframe
  - do merge in pandas
  - create new entity collection from metadata and dataframe (see docs)

Entity gotchas

- what *ontology label* is correct?
  - use `me.search_labels()` and LLM
- cannot compute with entities (because the ontology label might be wrong)
  - use `.quantity` for calculations (or `.q`)
- merging / extending entity collections
  - split entity collection into metadata and pandas dataframe
  - do merge in pandas
  - create new entity collection from metadata and dataframe (see docs)
- cannot write CSV file for some entity collections
  - CSV file can only handle tabular data
  - YAML and HDF5 are more flexible
  - use HDF5 for large data sets (binary format + compression)
    - note: can write/read entities in existing HDF5 files

Summary

- Lecture materials at `https://s.gwdg.de/ptjpe4`
- Feedback (good, bad and other) desired — `mammos_entity` still changing quickly: `hans.fangohr@mpsd.mpg.de`

Acknowledgements

S. Holt, M. Lang, S. Pathak, A. Petrocchi, M. Adams, W. Hortschitz, S. Pile, A. Vishina, M. Hasan, G. A. Marchant, T. Colnaghi, C. Winkler, J. Winkler, W. Rigaut, T. Devillers, P. L. Berre, C. Rahi, N. Dempsey, T. G. Woodcock, A. Marek, H. Herper, T. Schrefl.

"Towards a MAgnetic Multiscale MOdelling Suite (MaMMoS)", 10.1109/TMAG.2026.3689962 (2026)

Funding: European Commission (Project 101135546 MaMMoS), `https://mammos-project.github.io`