WARNING: This is a presentation PDF and does not have animations. For a MacOS Keynote presentation, ask Marijan.

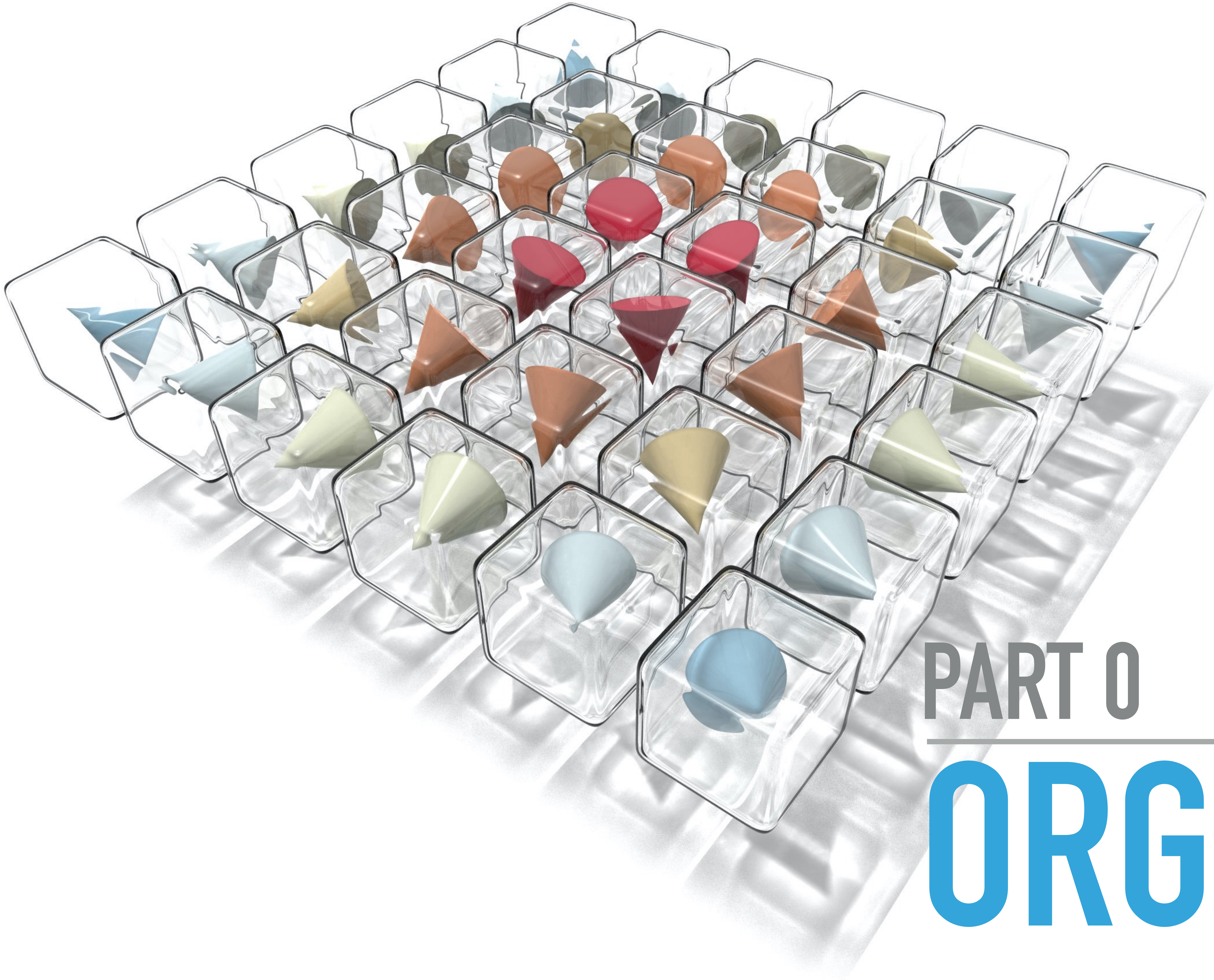
MARIJAN BEG[1], ONDREJ HOVORKA[1], AND HANS FANGOHR[1,2]

MICROMAGNETICS WITH UBERMAG

[1]University of Southampton, Highfield, SO17 1BJ Southampton, United Kingdom
[2]European XFEL GmbH, Holzkoppel 4, 22869 Schenefeld, Germany

Engineering and Physical Sciences Research Council

Last updated: 24 April 2020



PART 0

ORGANISATION

WHAT DO I NEED FOR THE WORKSHOP?

- The **workshop repository** is going to be updated regularly and it is available at:

  https://github.com/ubermag/workshop

  - It contains all the slides, tutorials, exercises, and up-to-date information about the workshop.
- Tutorials and exercises can be **run in the cloud** using Binder.
  - You **do not need to install anything** and no files will be created on your machine.
  - Binder can be accessed by **Binder badge** in the workshop repository:

    launch binder

  - Starting Binder can take a few minutes, so **please be patient**.
- **WARNING:** The most recent update of Safari web browser on MacOS sometimes does not interpret the colours well in 3D interactive plots and **Google Chrome is recommended**.

SESSIONS

- The workshop is divided into **3 sessions**:
  1. The basics of micromagnetics, Jupyter, and Python
     - **Friday 24 April 2020, 14:00-15:30** (UK time)
  2. Micromagnetics with Ubermag
     - **Wednesday 29 April 2020, 14:00-15:30** (UK time)
  3. Data analysis and visualisation with Ubermag
     - **Monday 4 May 2020, 14:00-15:30** (UK time)
- **Follow-up sessions**, depending on the requests and feedback can be organised after the workshop.

ZOOM

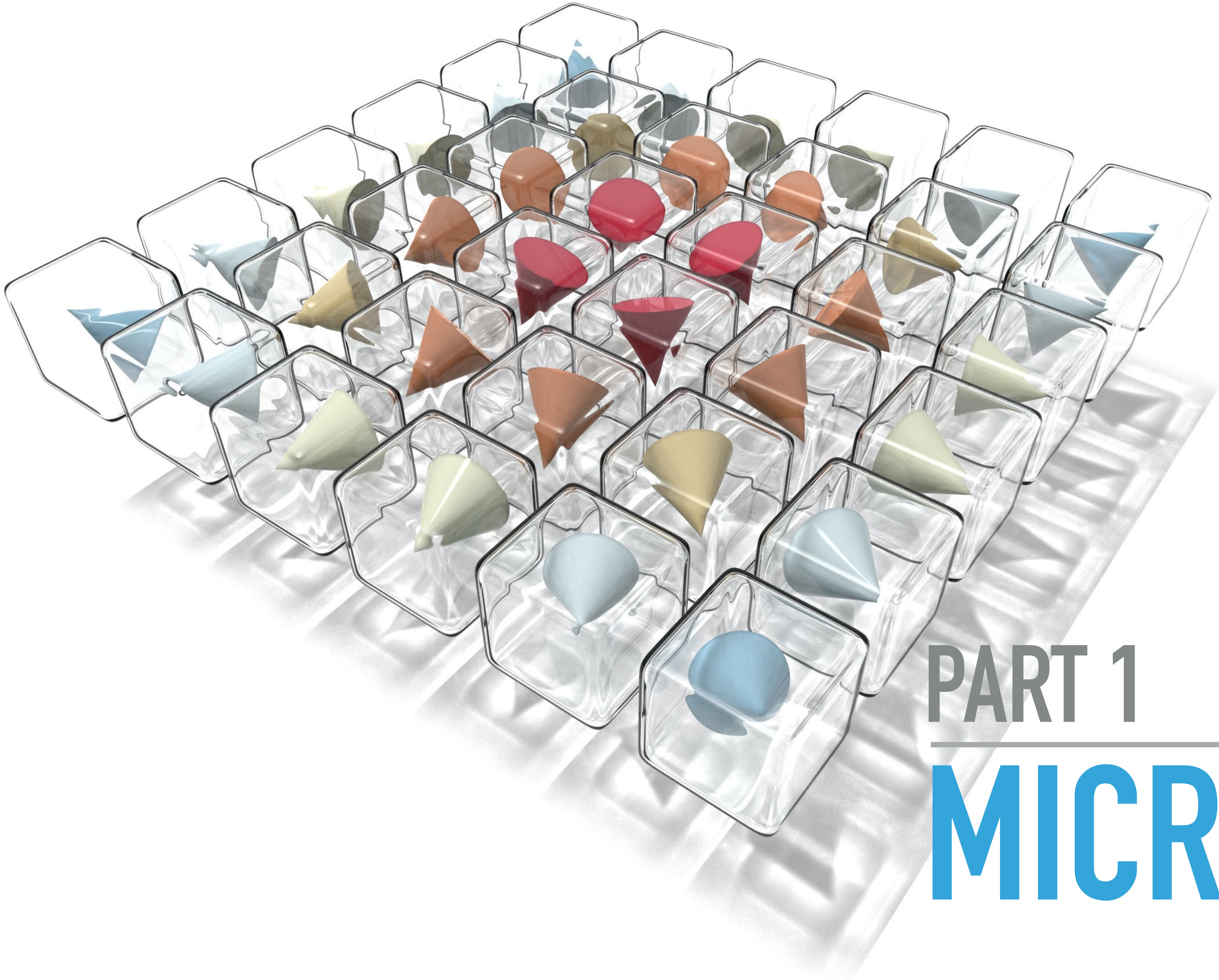
- We are going to use **Zoom** (https://zoom.us).
  - Please **install Zoom** before the workshop.
  - You do not have to create an account with Zoom.
  - The **link to access the workshop** is available in the workshop repository.
- When you join the session, **your camera and microphone are disabled by default**.
  - Please feel free to **turn them on if you have any questions**.
- Please **interrupt me at any time**.

FURTHER NOTES

- We are going to have **people from outside the Skyrmion Project** who will be joining us for the workshop at some stage (UK, USA, Sweden, Canada, India, China…)
    - Reasonable session times to cover different time zones
- All the materials are publicly available, and you are more then welcome to contribute. However, **please be careful what you share.**
- If you already attended our MMM, Intermag, or ICM conference tutorials, although there is some overlap, Ubermag always changes and there will be **a lot of new material**.

THE PLAN FOR SESSION 1

- The aim of the workshop is to gain some **basic understanding of micromagnetics** and to be **able to run useful simulations** in the end.
- Our assumption is **no previous experience** in micromagnetics, Python, or Jupyter.
- In Session 1, we are going to **cover only the very basics**, so that everybody should be able to follow sessions 2 and 3 (the most important things happens later).
- We are aware that most of you are **already familiar** with some or all of the topics today.

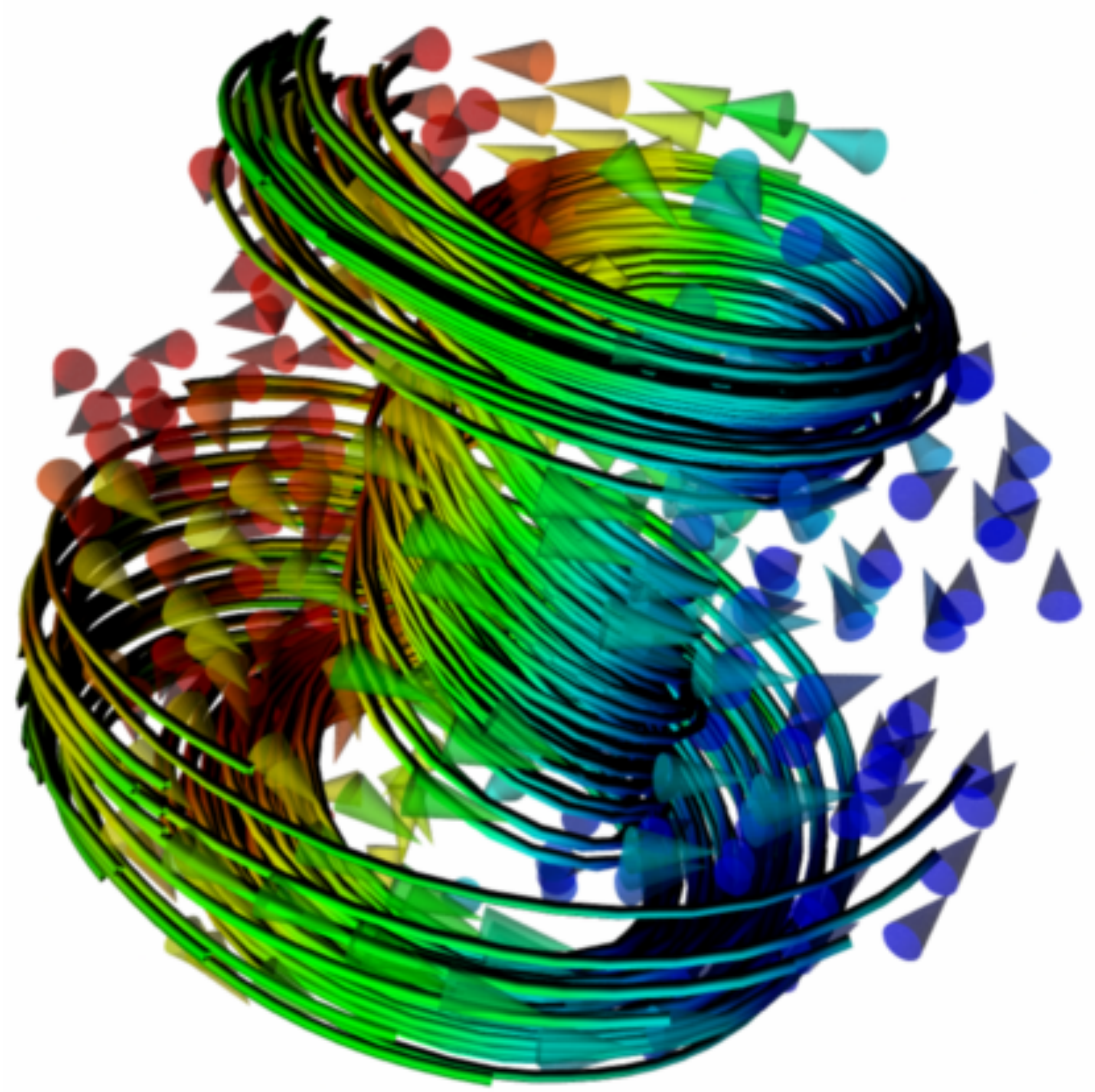


PART 1

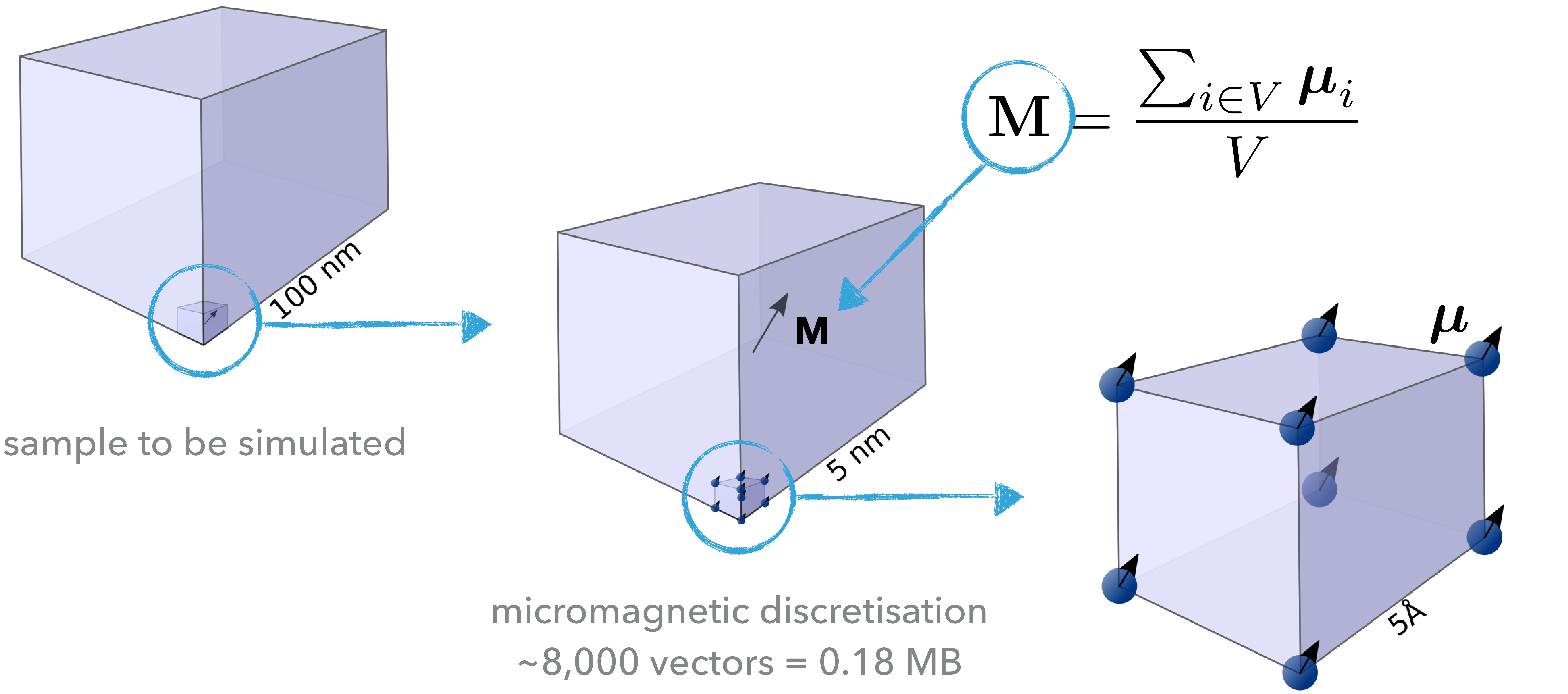
MICROMAGNETICS

MICROMAGNETICS

"… is the study, modelling, and simulation of magnetic materials and their behaviour at the nanometer scale."



FINITE-DIFFERENCE DISCRETISATION



$$\mathbf{M} = \frac{\sum_{i \in V} \boldsymbol{\mu}_i}{V}$$

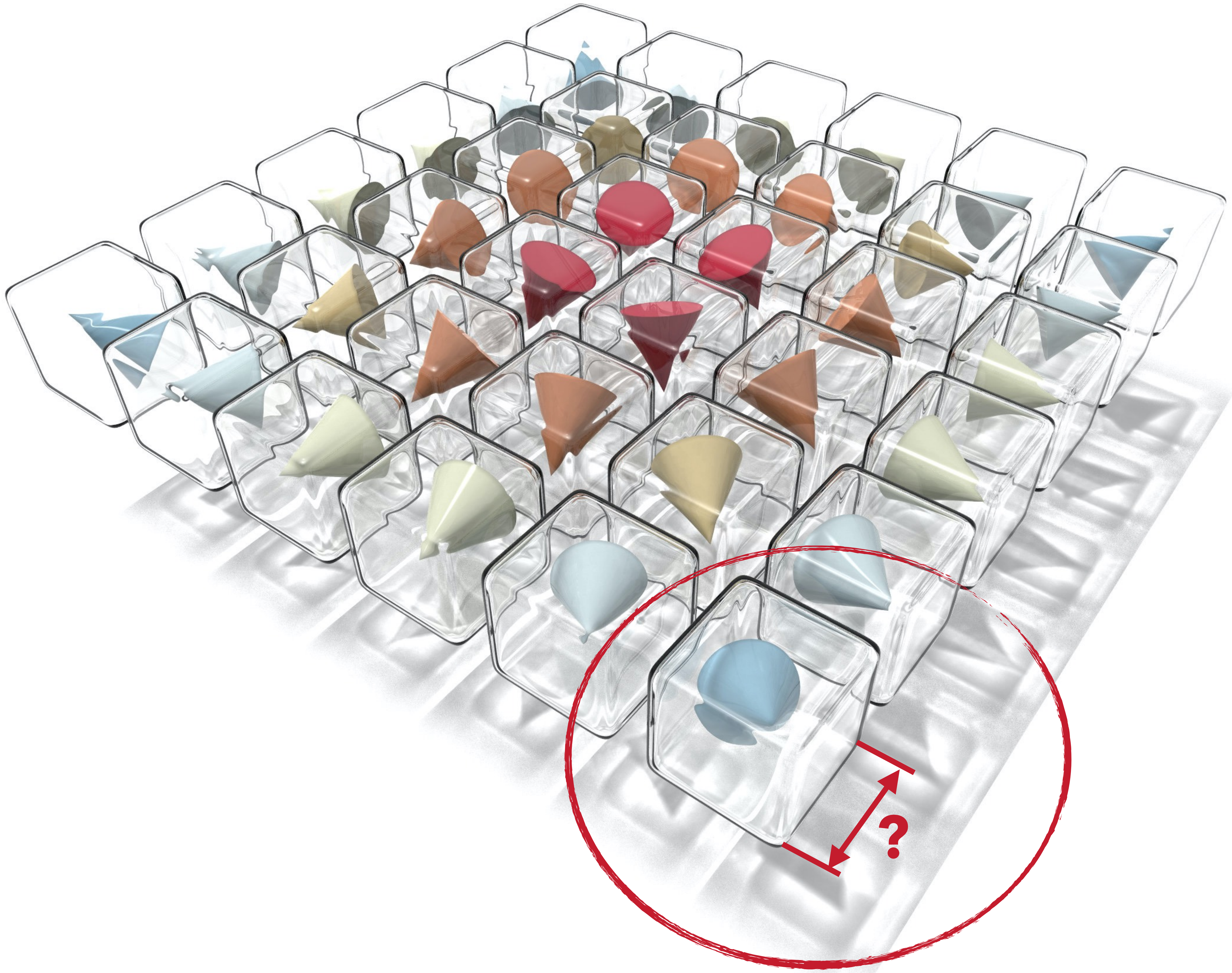
sample to be simulated

micromagnetic discretisation
~8,000 vectors = 0.18 MB

atomistic discretisation
~ 8,000,000 vectors ~ 183 MB

HOW DO I CHOOSE THE RIGHT DISCRETISATION?

- It must be…
  - large enough to ignore the atomic structure of the material (individual magnetic moments), but …
  - small enough to resolve magnetic configurations like domain walls, vortices, skyrmions etc.



MAGNETISATION

- Magnetisation is **the main "unknown"** we want to compute using micromagnetics.
- A single vector can be associated to each discretisation cell
- Continuous **vector field**

$$\mathbf{M} = \mathbf{M}(\mathbf{r}, t)$$

- It is generally the **function of both space and time**.

MICROMAGNETIC ASSUMPTIONS

1. Magnetisation is **differentiable** (continuous and slowly changing) in respect to both space and time.
2. The **norm of magnetisation is constant** - both space and time invariant.
   - zero temperature

$$\mathbf{m}(\mathbf{r}) = \frac{\mathbf{M}(\mathbf{r})}{M_\mathrm{s}} \qquad M_\mathrm{s} = |\mathbf{M}|$$

normalised magnetisation

saturation magnetisation

$$|\mathbf{m}| = 1$$

$$m_x, m_y, m_z \in [-1, 1]$$

ENERGY EQUATION (HAMILTONIAN)

- The sum of individual energy density terms

scalar field

$$w(\mathbf{m}) = w_1(\mathbf{m}) + w_2(\mathbf{m}) + w_3(\mathbf{m}) + \cdots = \sum_i w_i(\mathbf{m})$$

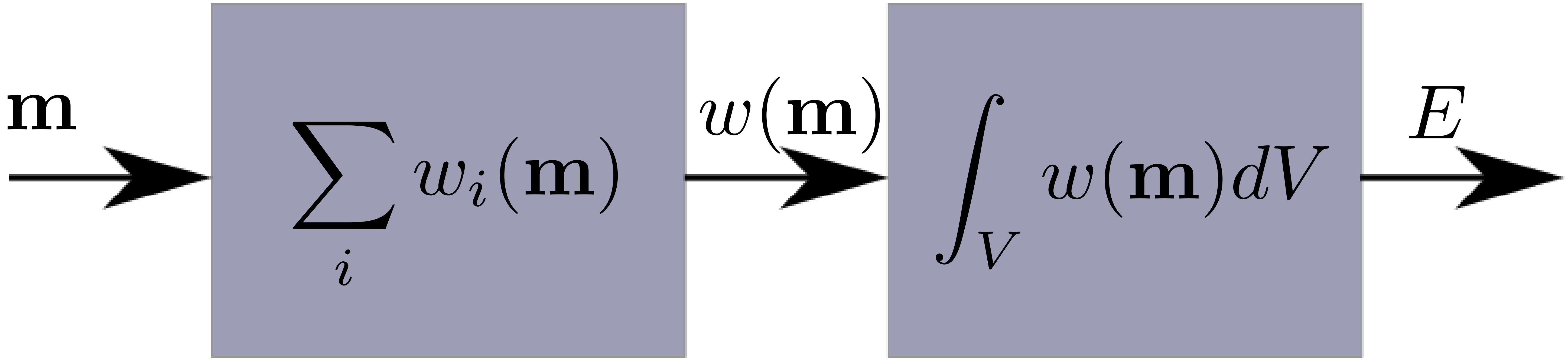
?

user-defined

- Energy

scalar

$$E = \int_V w(\mathbf{m}) dV$$

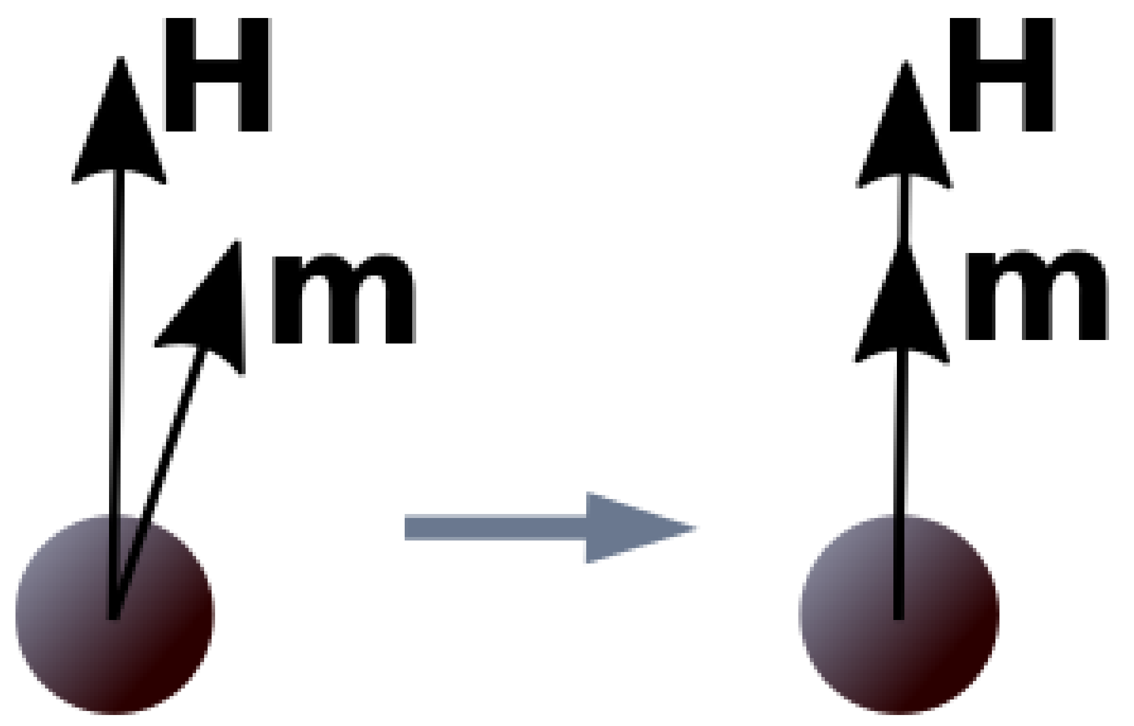


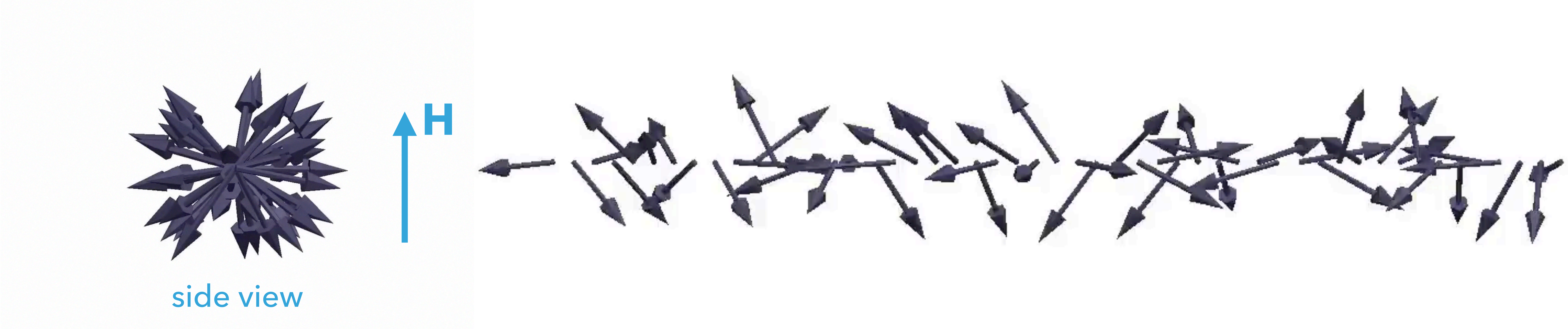
ZEEMAN

- Aligns magnetisation **m** **parallel to the external magnetic field** **H**.
- Parameter: **H**
- **B** vs. **H**

$$w_\mathrm{z} = -\mu_0 \mathbf{M} \cdot \mathbf{H} = -\mu_0 M_\mathrm{s} \mathbf{m} \cdot \mathbf{H}$$

$$\mathbf{M} = M_\mathrm{s} \mathbf{m}$$



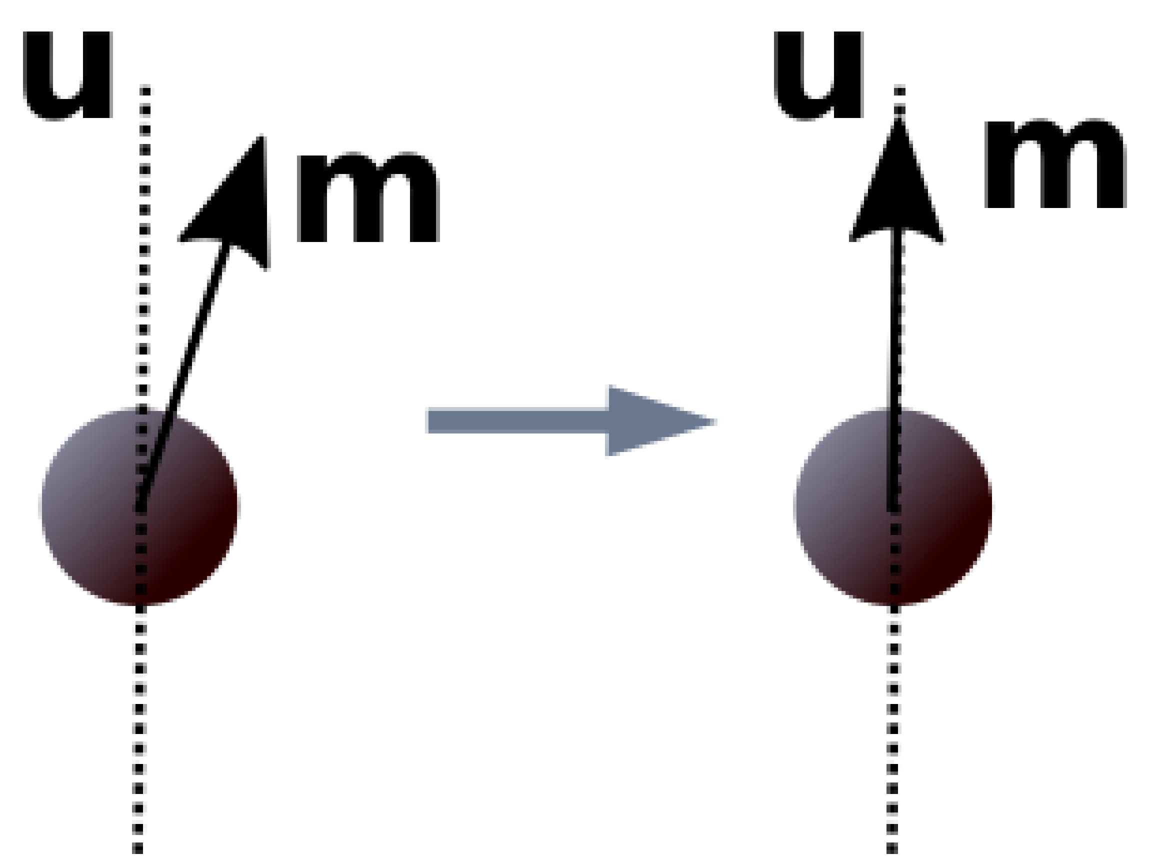


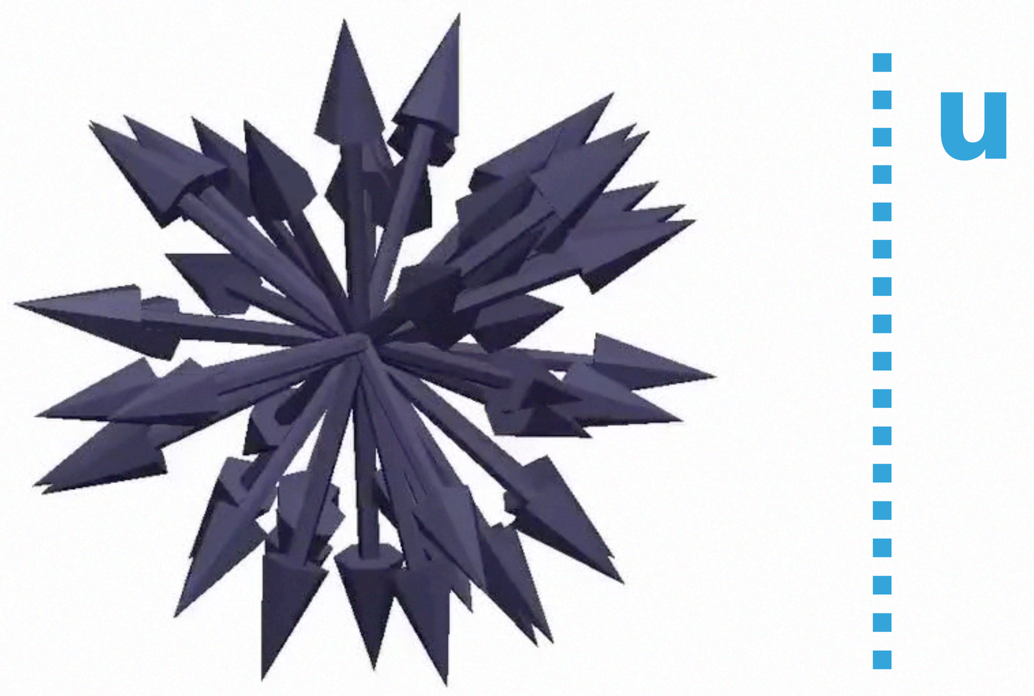
UNIAXIAL ANISOTROPY

- Aligns magnetisation **parallel or antiparallel to the anisotropy axis u**.
- Parameters: K, $\mathbf{u}$

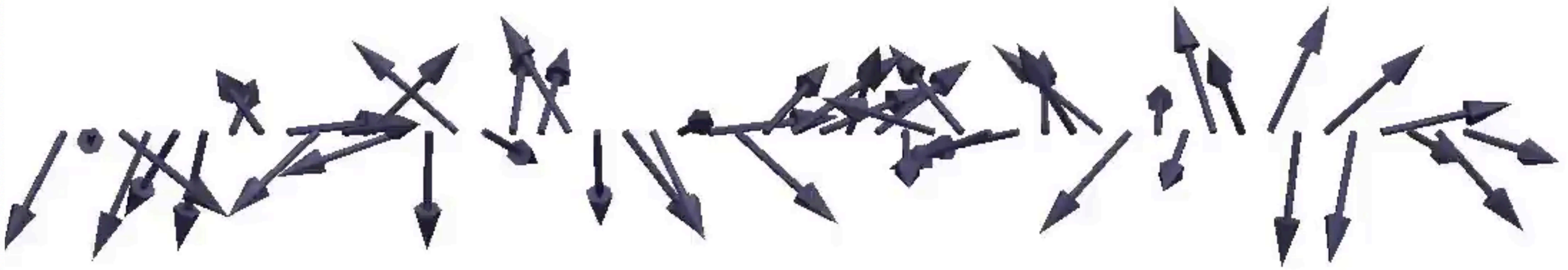
$$w_\mathrm{a} = -K(\mathbf{m}\cdot\mathbf{u})^2$$

Be careful about the sign of K!





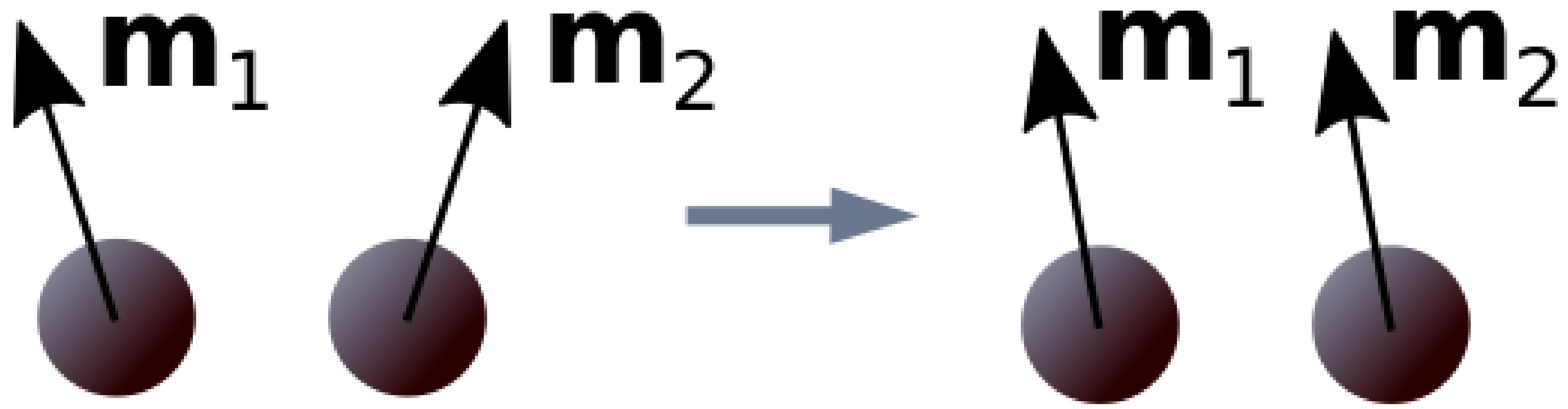
side view

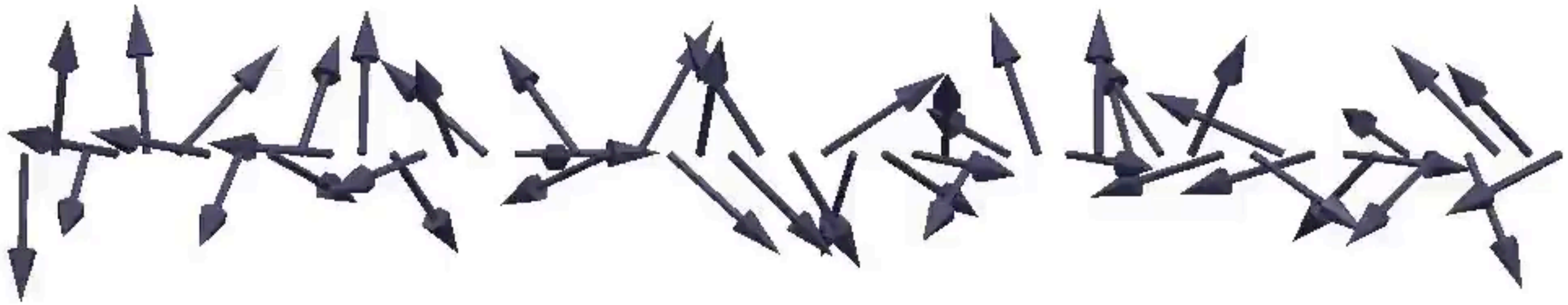


EXCHANGE

- Aligns all magnetic moments **parallel to each other**.
- Parameter: *A*

$$w_{\text{ex}} = A[(\nabla m_x)^2 + (\nabla m_y)^2 + (\nabla m_z)^2] = A\mathbf{m}\cdot\nabla^2\mathbf{m} = A(\nabla\mathbf{m})^2$$

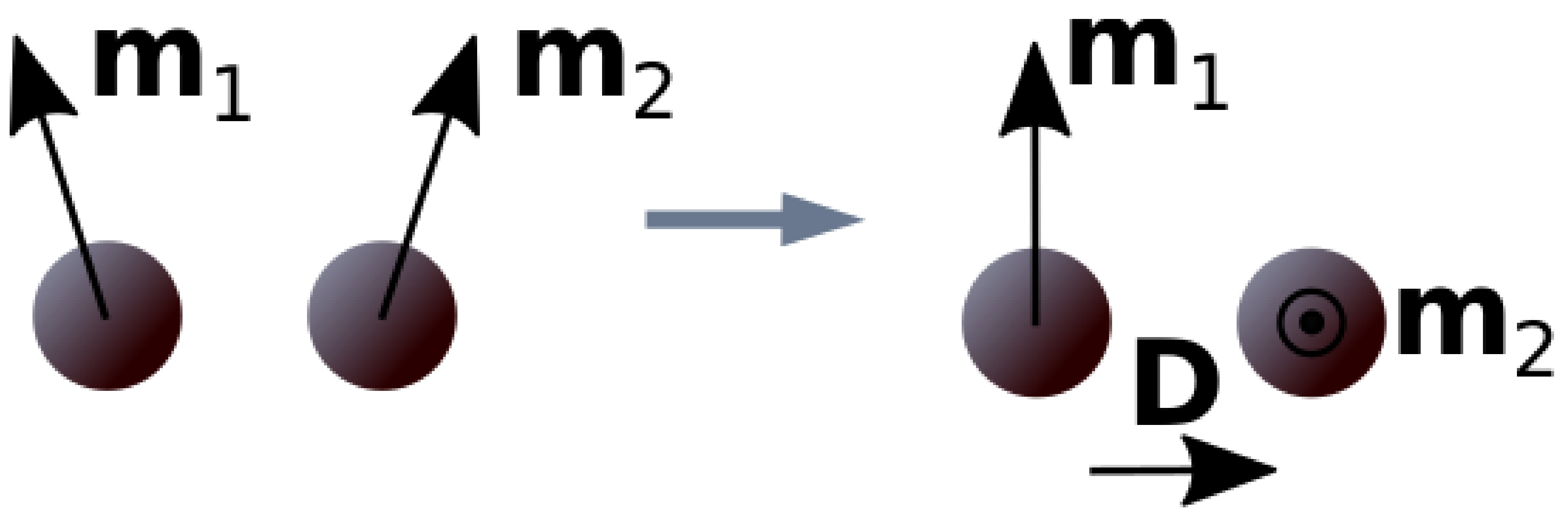


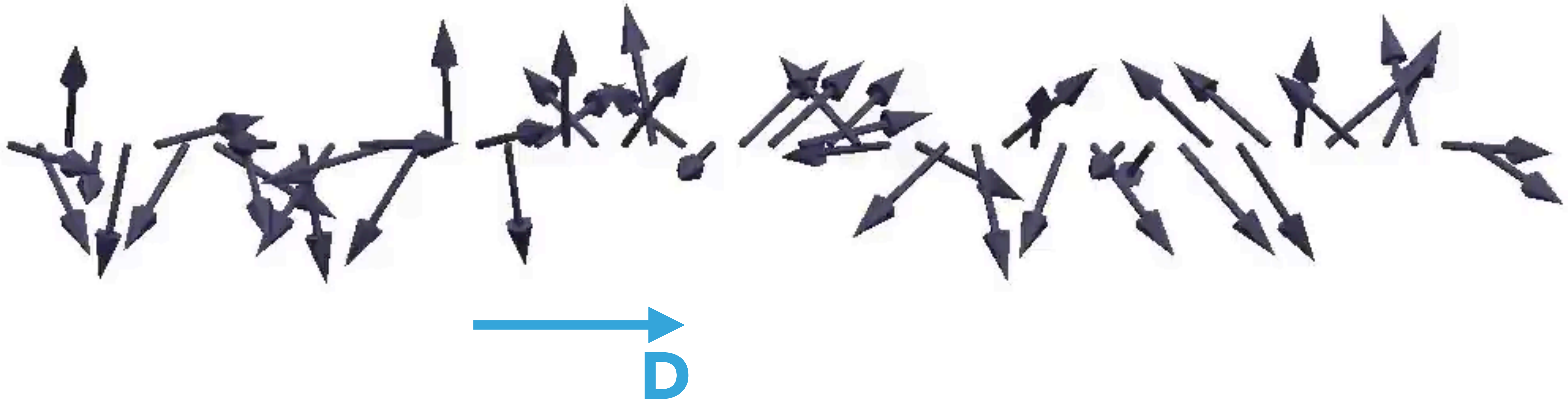


DZYALOSHINSKII-MORIYA (DM)

- Aligns neighbouring magnetic moments **perpendicular to each other**.
- Parameters: *D*, crystallographic class

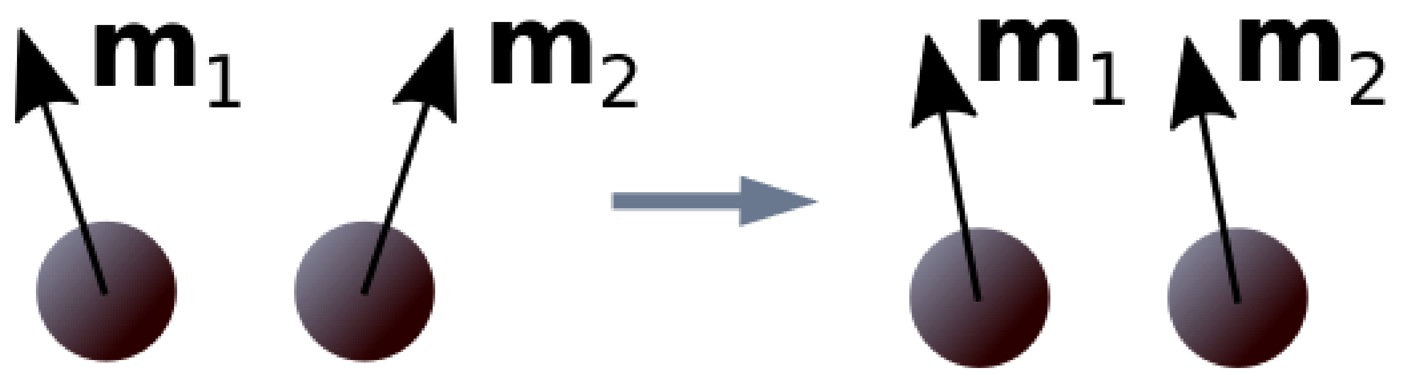
$$w_{\rm dmi} = D\mathbf{m}\cdot(\nabla\times\mathbf{m})$$



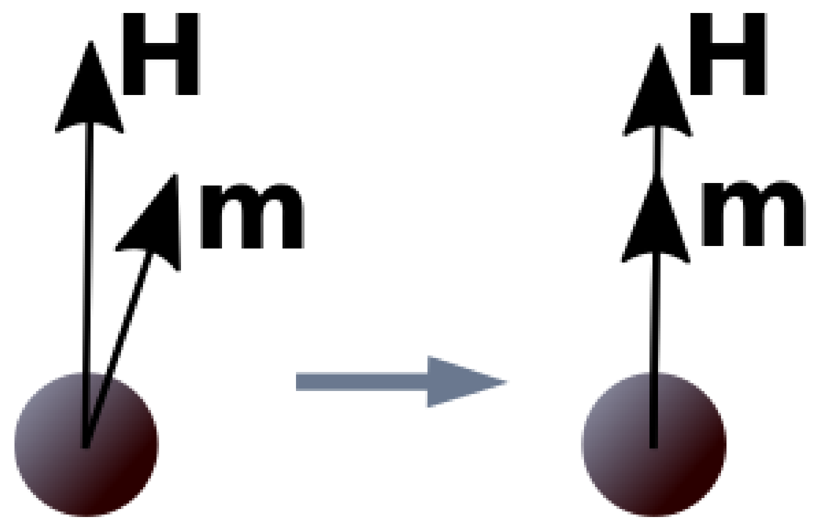


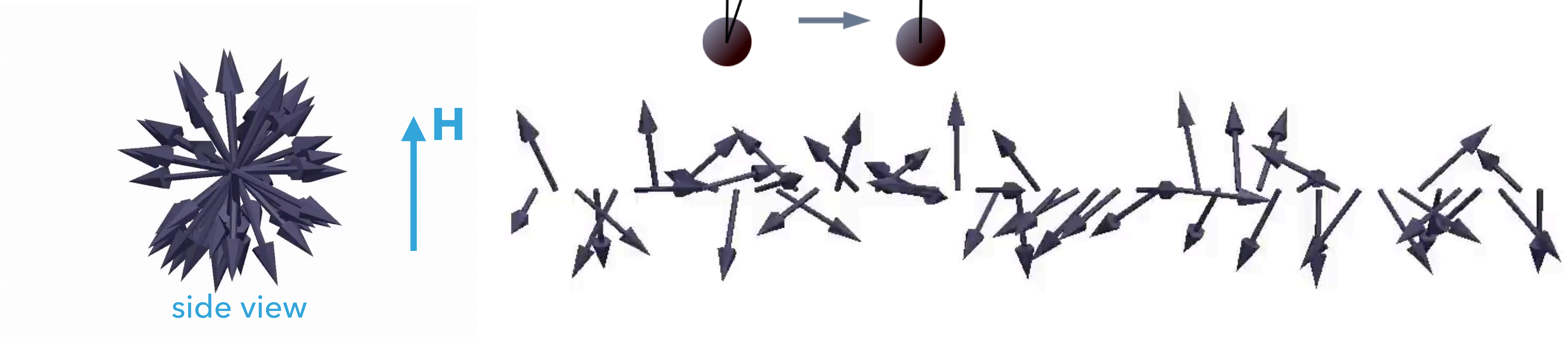
EXCHANGE AND ZEEMAN

- Exchange aligns all spins parallel to each other with no preferential direction.



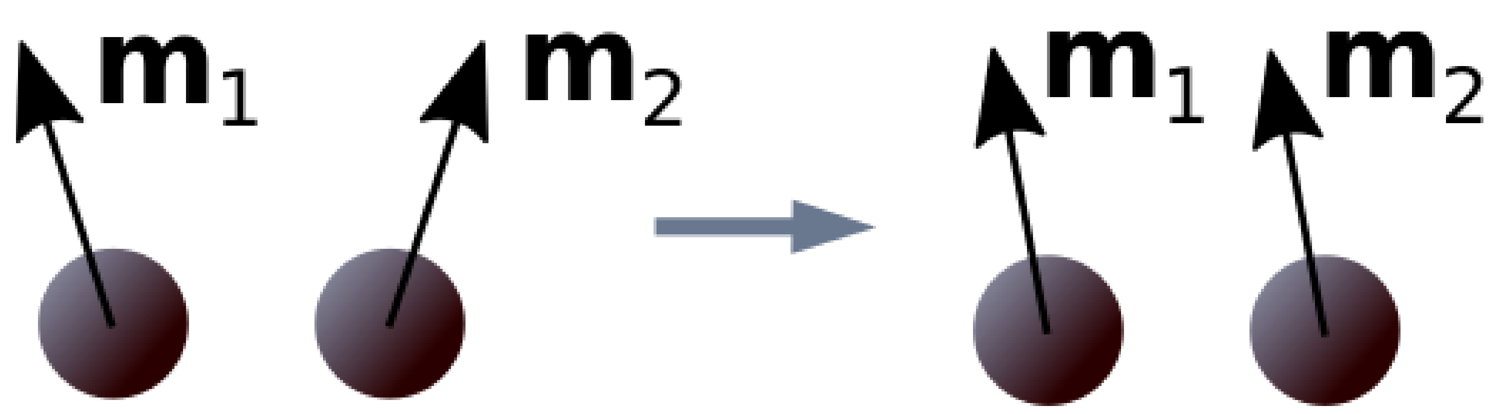
- Zeeman dictates the direction of magnetisation.



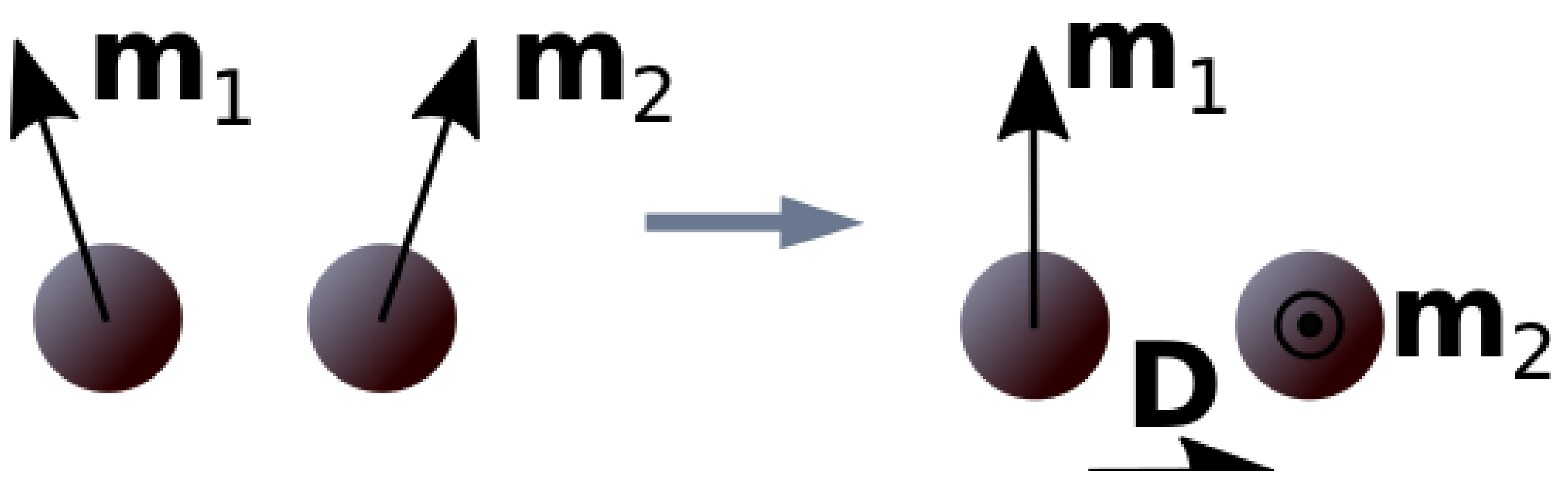


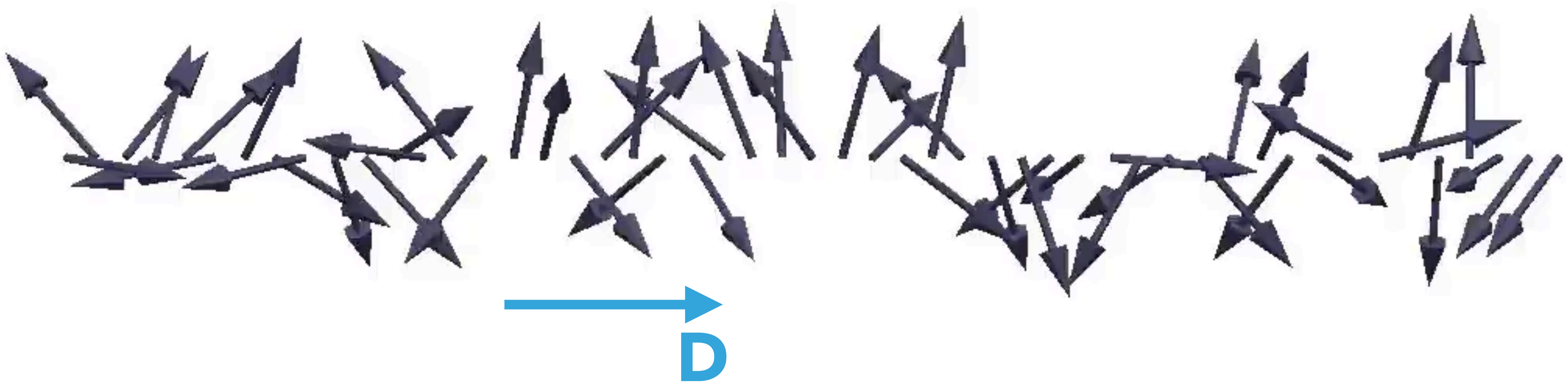
EXCHANGE AND DMI

- Exchange aligns all spins parallel to each other with no preferential direction.

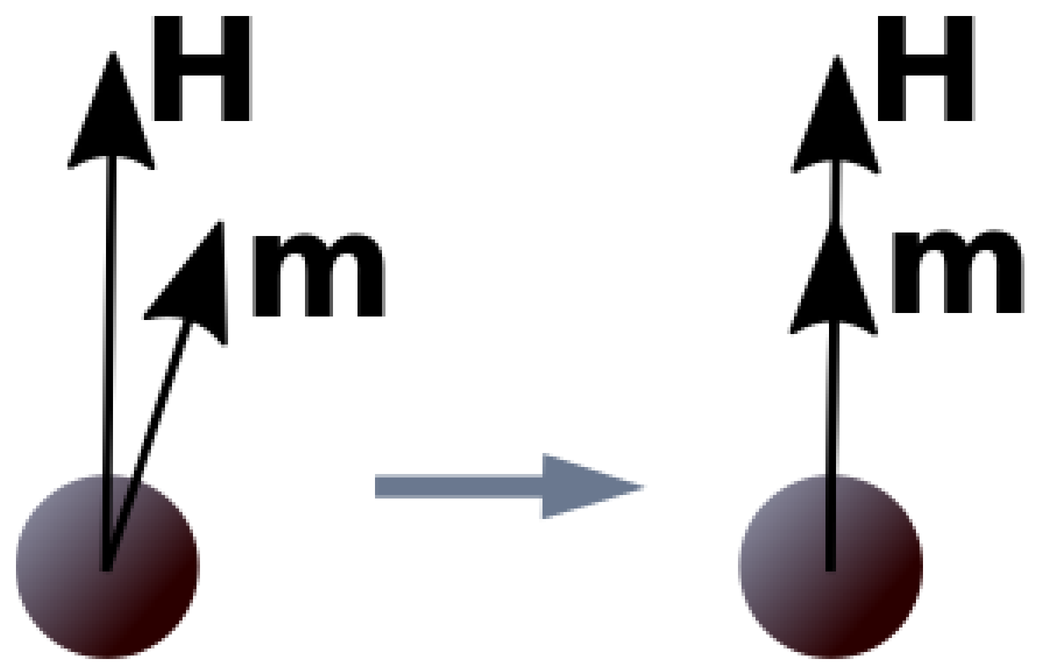


- DMI wants neighbouring spins perpendicular to each other.

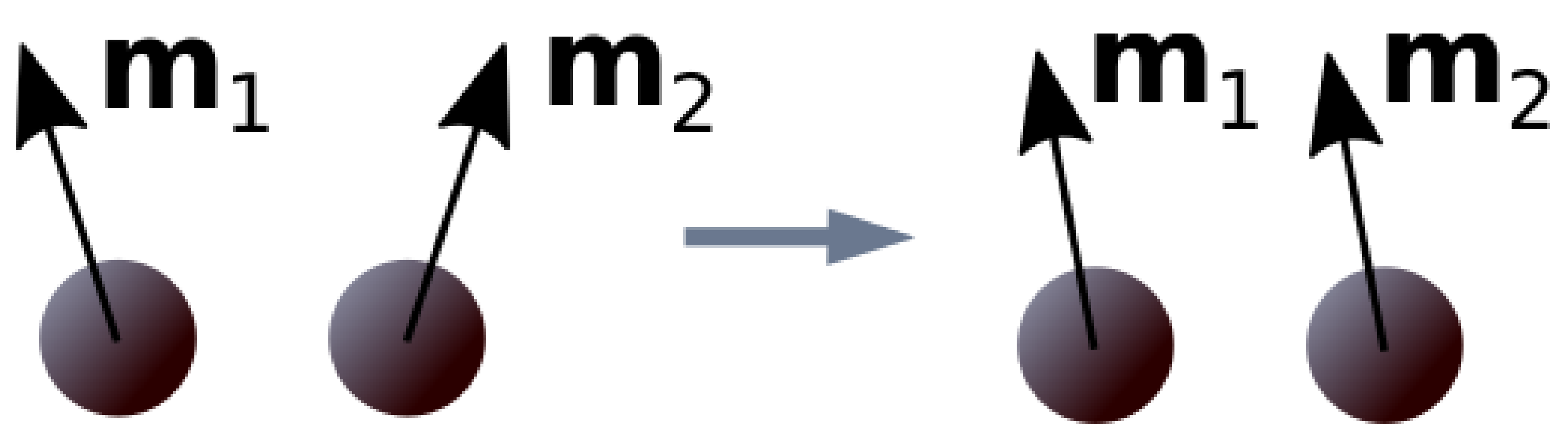




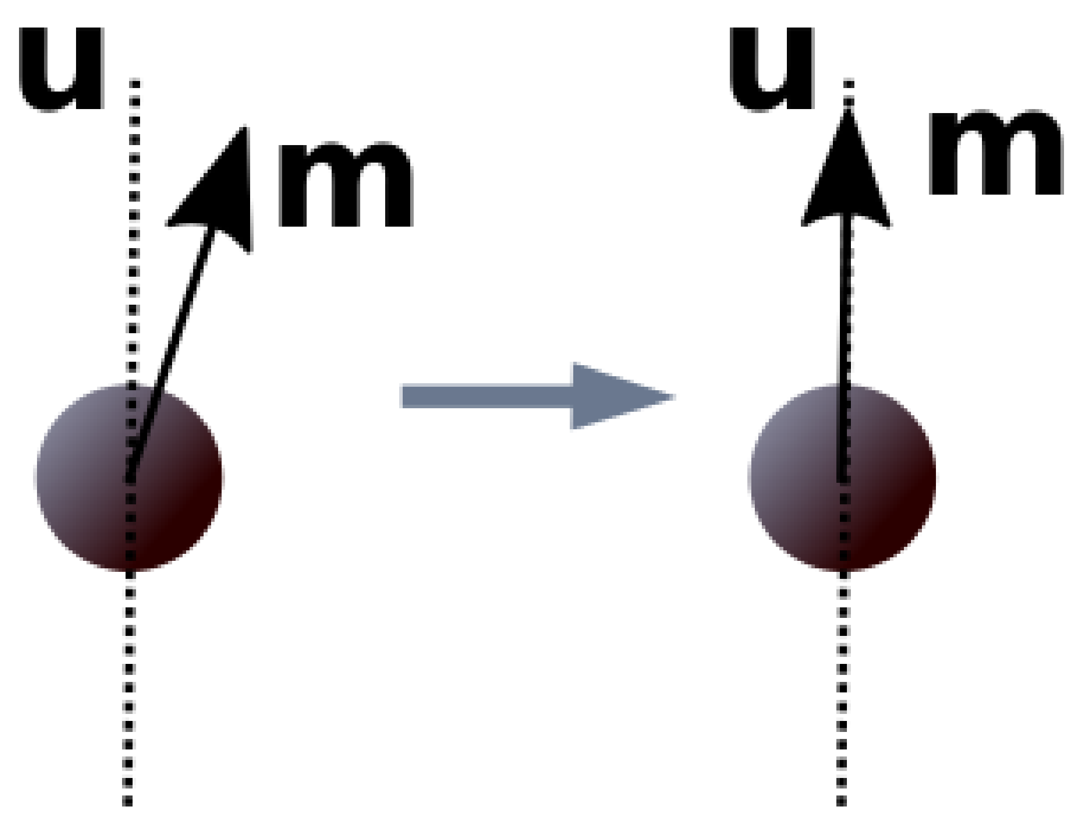
MORE COMPLICATED CASE



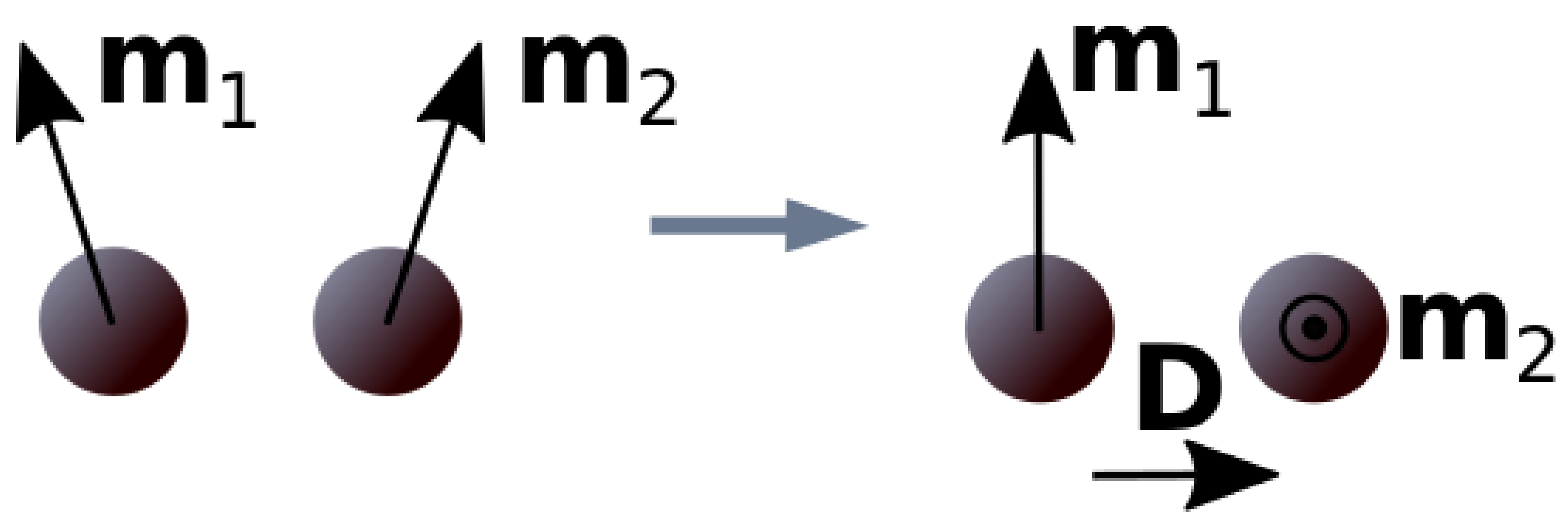
$$w_{\mathrm{z}} = -\mu_0 M_{\mathrm{s}} \mathbf{m} \cdot \mathbf{H}$$



$$w_{\mathrm{ex}} = A(\nabla \mathbf{m})^2$$

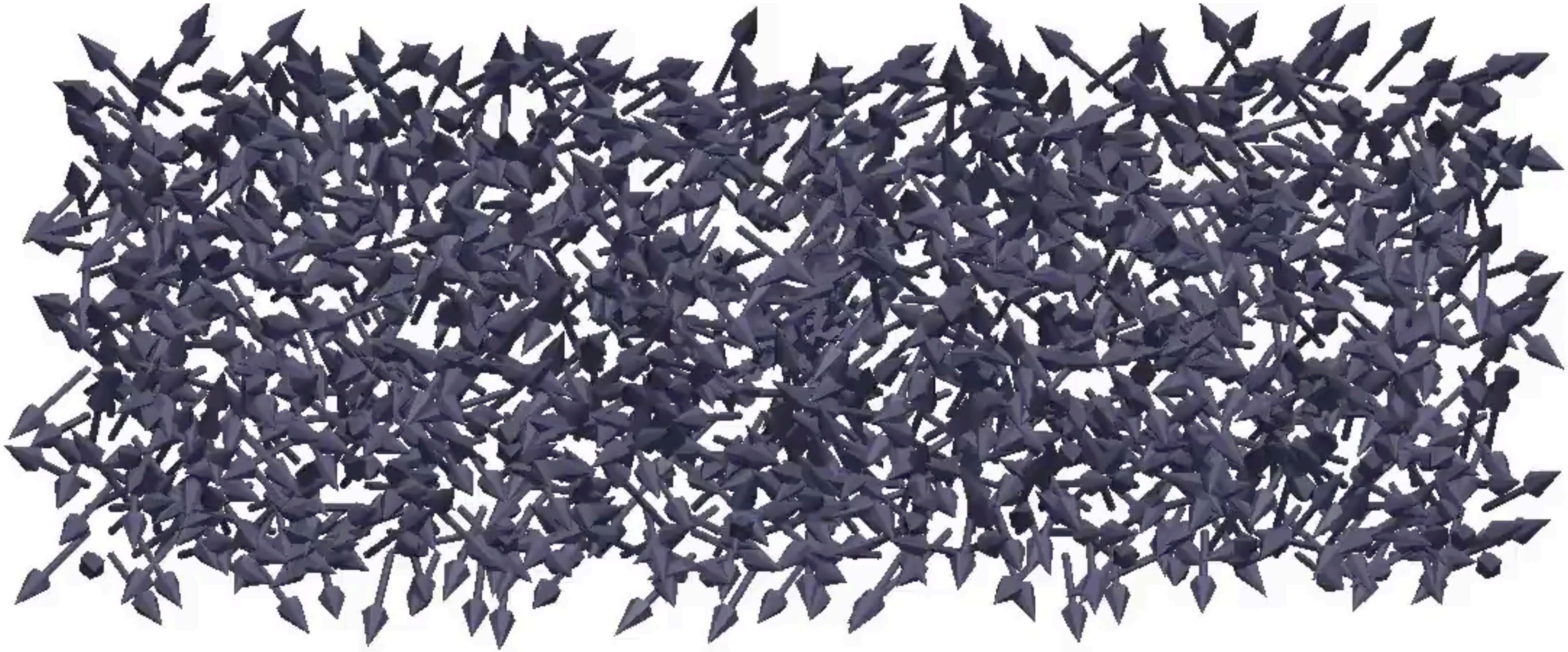


$$w_{\mathrm{a}} = -K(\mathbf{m} \cdot \mathbf{u})^2$$



$$w_{\mathrm{dmi}} = D\mathbf{m} \cdot (\nabla \times \mathbf{m})$$

2D SAMPLE

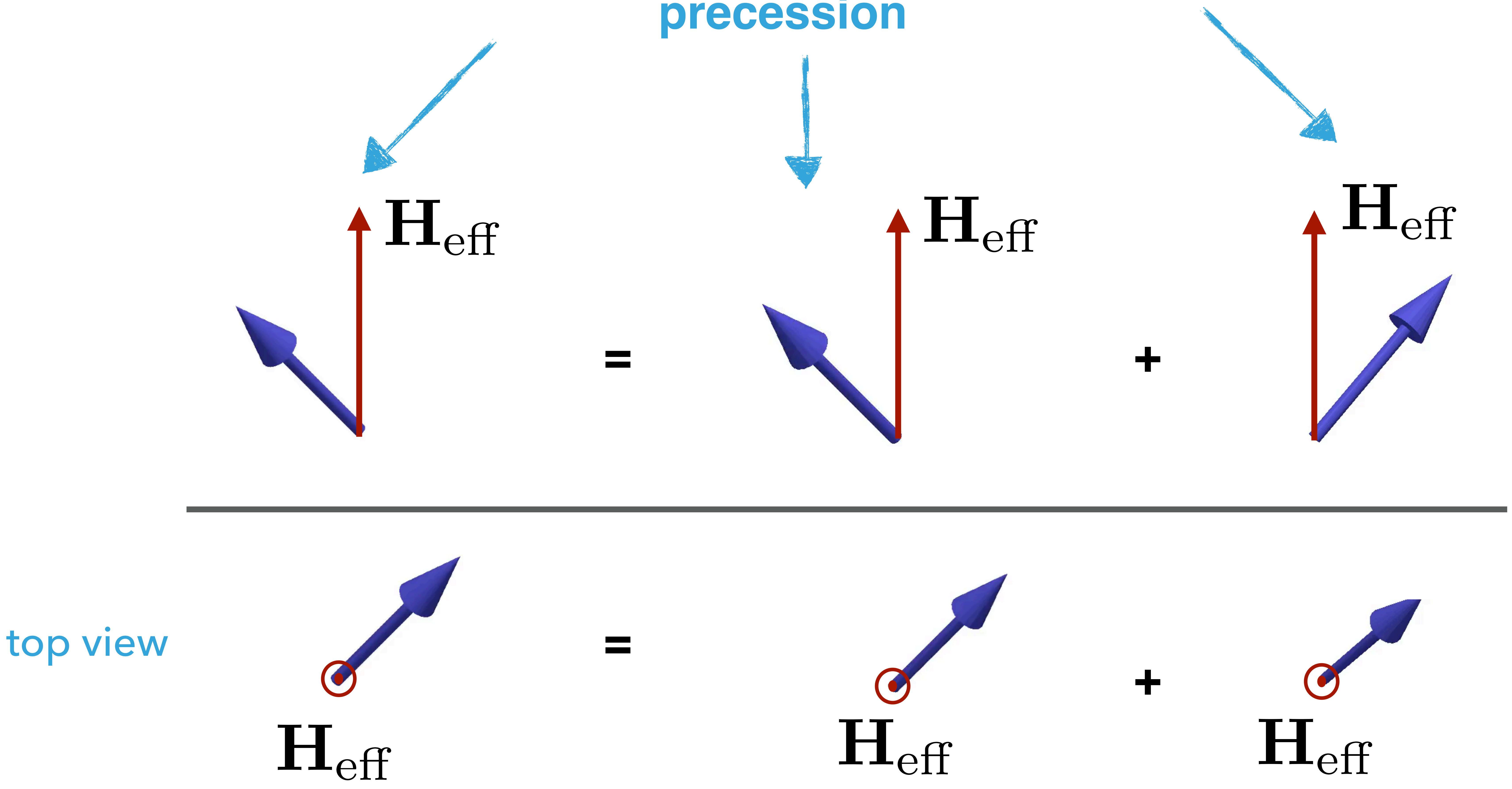


OVERVIEW OF DIFFERENT ENERGY TERMS

- **Zeeman**
  - **H** (vector)
- **Uniaxial Anisotropy**
  - *K* (scalar) *and* **u** (vector)
  - be careful about the sign of *K* (easy axis vs. easy plane)
- **Exchange**
  - A (scalar)
  - antiferromagnetic exchange (A<0) not justifiable
- **DMI**
  - D (scalar)
  - The form of DMI term depends on the DMI vector
- **Cubic anisotropy**
  - *K, u1, u2*
  - To be discussed next time.
- **Demagnetisation (magnetostatic)**
  - To be discussed next time

DYNAMICS EQUATION

$$\frac{d\mathbf{m}}{dt} = \underbrace{-\gamma_0^* \mathbf{m} \times \mathbf{H}_{\text{eff}}}_{\text{precession}} + \overbrace{\alpha \mathbf{m} \times \frac{d\mathbf{m}}{dt}}^{\text{damping}} \qquad (\gamma_0 = \mu_0 \gamma)$$



DYNAMICS EQUATION

$$\frac{d\mathbf{m}}{dt} = -\gamma_0^* \mathbf{m} \times \mathbf{H}_{\text{eff}} + \alpha \mathbf{m} \times \frac{d\mathbf{m}}{dt}$$

$$\Downarrow$$

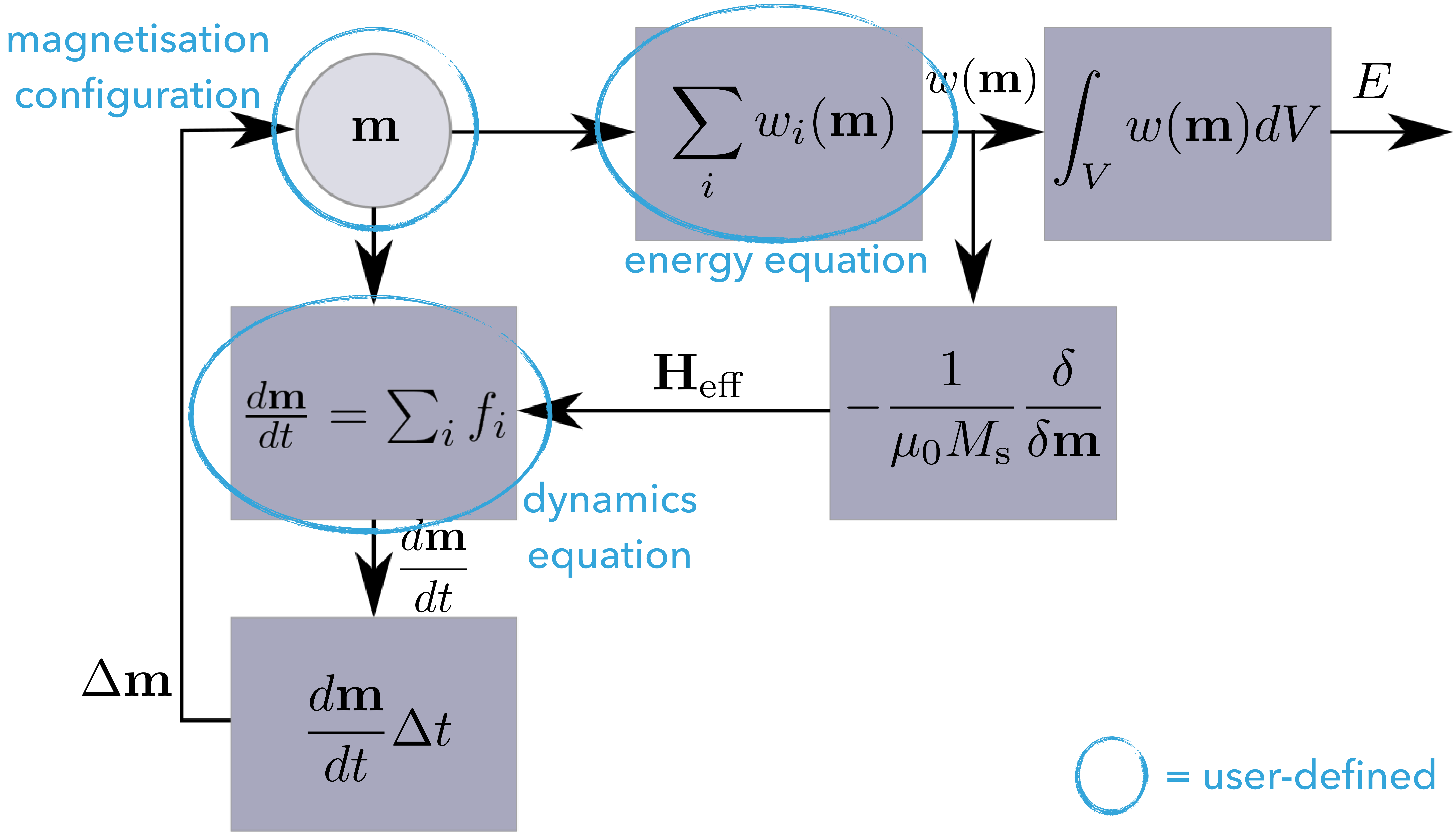
$$\frac{d\mathbf{m}}{dt} = -\frac{\gamma_0^*}{1+\alpha^2} \mathbf{m} \times \mathbf{H}_{\text{eff}} - \frac{\gamma_0^* \alpha}{1+\alpha^2} \mathbf{m} \times (\mathbf{m} \times \mathbf{H}_{\text{eff}})$$

$$\frac{d\mathbf{m}}{dt} = f_1(\mathbf{m}, \mathbf{H}_{\text{eff}}) + f_1(\mathbf{m}, \mathbf{H}_{\text{eff}}) + \cdots = f(\mathbf{m}, \mathbf{H}_{\text{eff}})$$

- Effective field

$$\mathbf{H}_{\text{eff}} = -\frac{1}{\mu_0 M_{\text{s}}} \frac{\delta w(\mathbf{m})}{\delta \mathbf{m}}$$

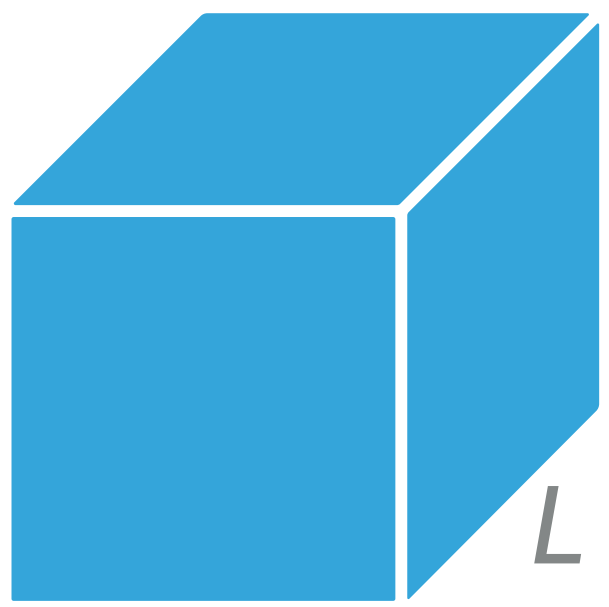
(OVERSIMPLIFIED) MICROMAGNETIC SIMULATOR

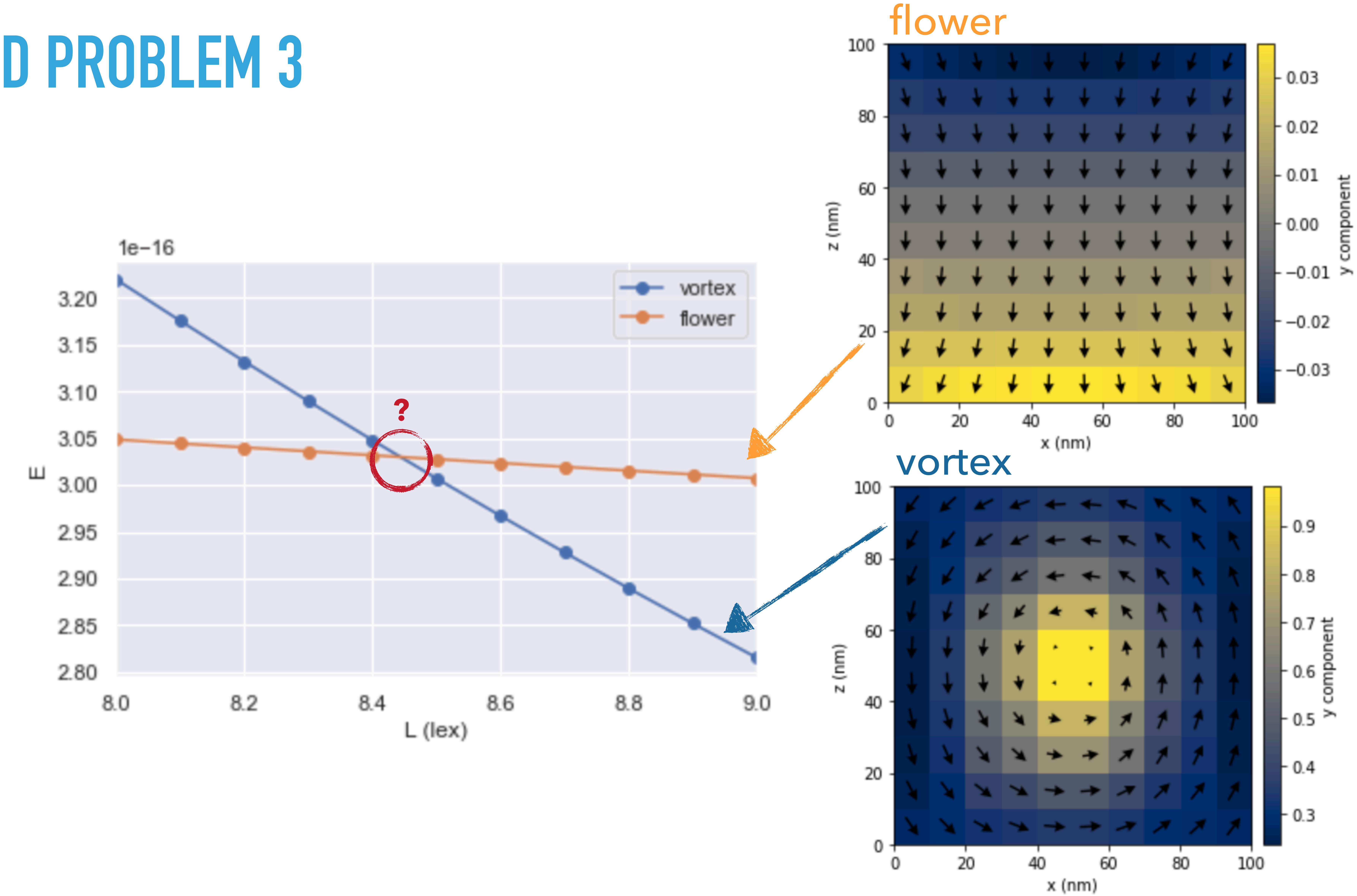


PART 2

COMPUTATIONAL WORKFLOW

CASE STUDY: STANDARD PROBLEM 3



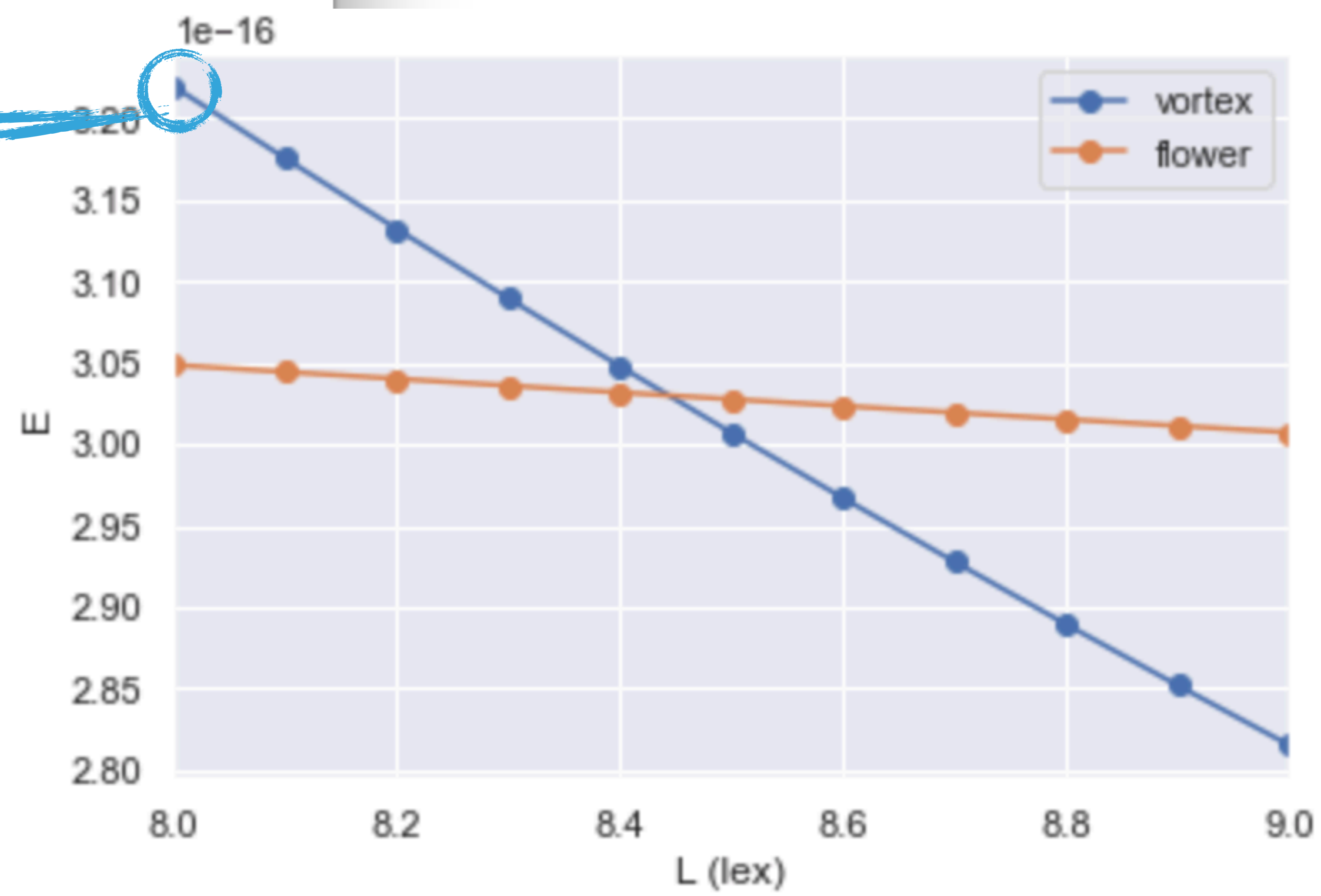


Research question: For what cube edge length *L*, vortex and flower states have the same energy?

STEP 1: WRITE CONFIGURATION FILE

my_project — IPython: Users/mb4e10 — emacs -nw stdprob3.mif — 95×37

```
# MIF 2.1
# MIF Example File: stdprob3.mif
# Description: Sample problem description for muMAG Standard Problem #3

set pi [expr {4*atan(1.0)}]
set mu0 [expr {4*$pi*1e-7}]

Parameter seed 0
RandomSeed $seed  ;# Initialize seed to {} to get a seed
## value from the system clock.


#######################################################################
# Simulation parameters

Parameter L 8    ;# Cube dimension, in units of exchange length
Parameter N 32   ;# Number of cells along one edge of cube

Parameter initial_state "vortex" ;# Initial state should be
## one of "uniform", "vortex", "canted", "cantedvortex", "twisted",
## "random" or "file <filename>"; in the last case <filename> is the
## name of a file to use as the initial configuration.

Parameter stop 1e-3

#######################################################################
# Auxiliary variables:

# Work out Ms so magnetostatic energy density, Km=0.5*mu0*Ms^2,
# is 1e6 J/m^3
set Km 1e6
set Ms [expr {sqrt(2*$Km/$mu0)}]

# Arbitrarily set cube dimension to 100 nm, and compute cellsize and
# exchange length based on parameters L and N.
-uu-:---F1  stdprob3.mif   Top L1     (Fundamental)------------------------------------------
```



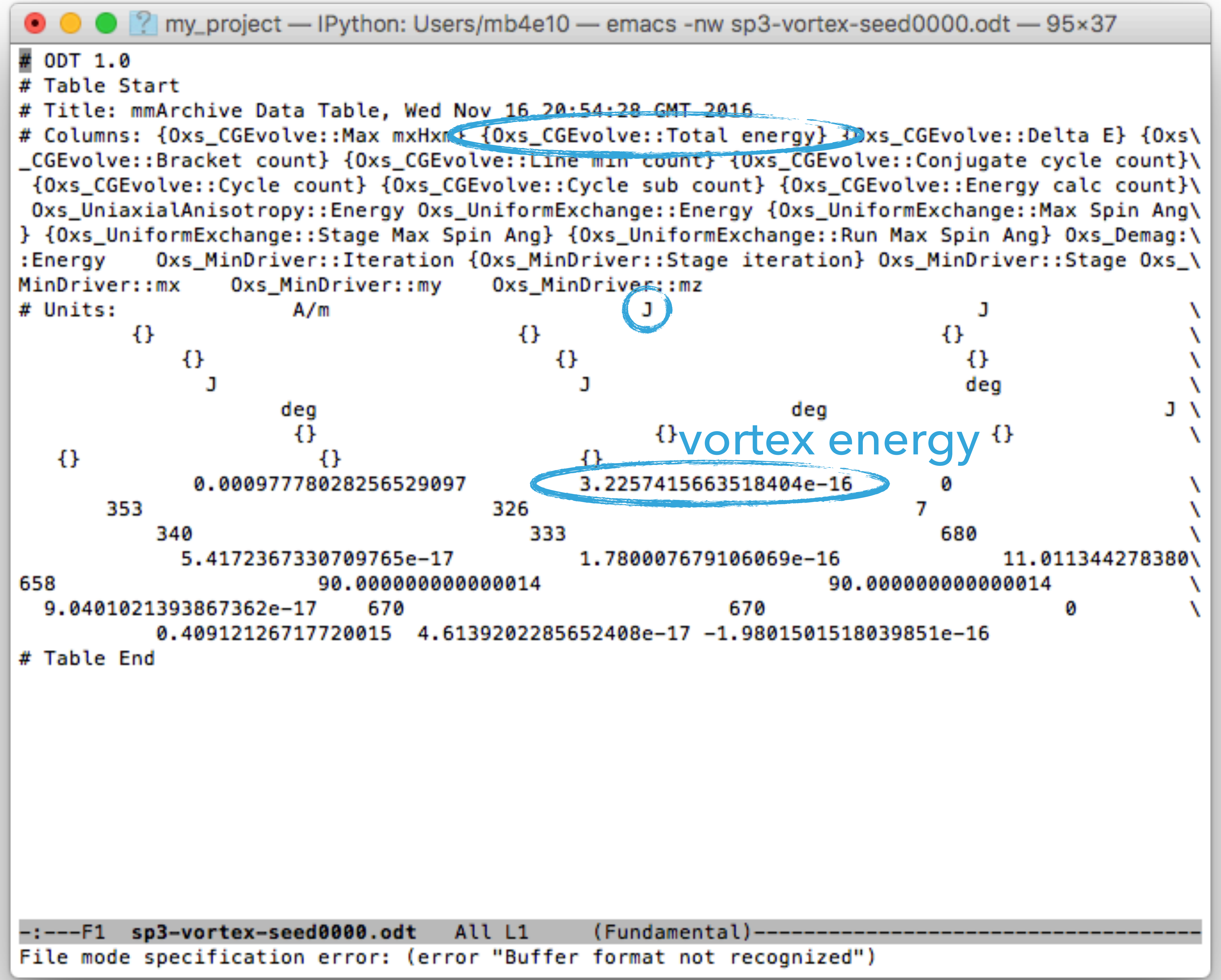
STEP 2: RUN SIMULATION

1. configuration file

2. run OOMMF

3. output file

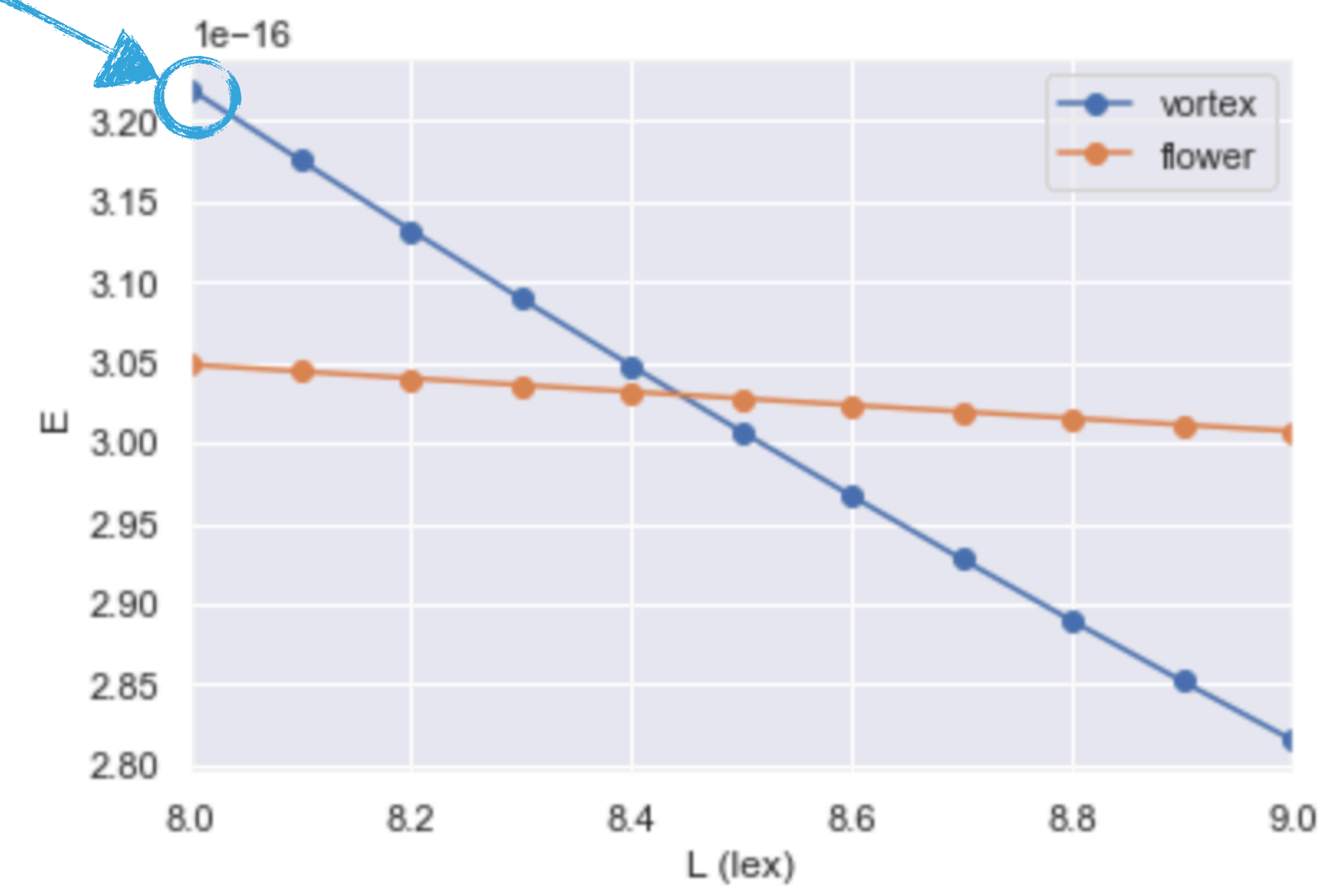
STEP 3: READ RESULTS



```
my_project — IPython: Users/mb4e10 — emacs -nw sp3-vortex-seed0000.odt — 95×37
# ODT 1.0
# Table Start
# Title: mmArchive Data Table, Wed Nov 16 20:54:28 GMT 2016
# Columns: {Oxs_CGEvolve::Max mxHxm} {Oxs_CGEvolve::Total energy} {Oxs_CGEvolve::Delta E} {Oxs\
_CGEvolve::Bracket count} {Oxs_CGEvolve::Line min count} {Oxs_CGEvolve::Conjugate cycle count}\
 {Oxs_CGEvolve::Cycle count} {Oxs_CGEvolve::Cycle sub count} {Oxs_CGEvolve::Energy calc count}\
 Oxs_UniaxialAnisotropy::Energy Oxs_UniformExchange::Energy {Oxs_UniformExchange::Max Spin Ang\
} {Oxs_UniformExchange::Stage Max Spin Ang} {Oxs_UniformExchange::Run Max Spin Ang} Oxs_Demag:\
:Energy    Oxs_MinDriver::Iteration {Oxs_MinDriver::Stage iteration} Oxs_MinDriver::Stage Oxs_\
MinDriver::mx    Oxs_MinDriver::my    Oxs_MinDriver::mz
# Units:              A/m                          J                            J                 \
         {}                              {}                               {}                \
             {}                             {}                               {}                \
              J                             J                              deg                \
                    deg                                         deg                         J \
                     {}                              {}                   {}                \
    {}                    {}                    {}
            0.00097778028256529097          3.2257415663518404e-16           0                 \
       353                              326                              7                 \
           340                              333                              680                \
             5.4172367330709765e-17          1.780007679106069e-16           11.011344278380\
658                 90.000000000000014                      90.000000000000014                 \
  9.0401021393867362e-17   670                              670                       0        \
           0.40912126717720015  4.6139202285652408e-17 -1.9801501518039851e-16
# Table End

-:---F1  sp3-vortex-seed0000.odt    All L1     (Fundamental)-----------------------------------
File mode specification error: (error "Buffer format not recognized")
```

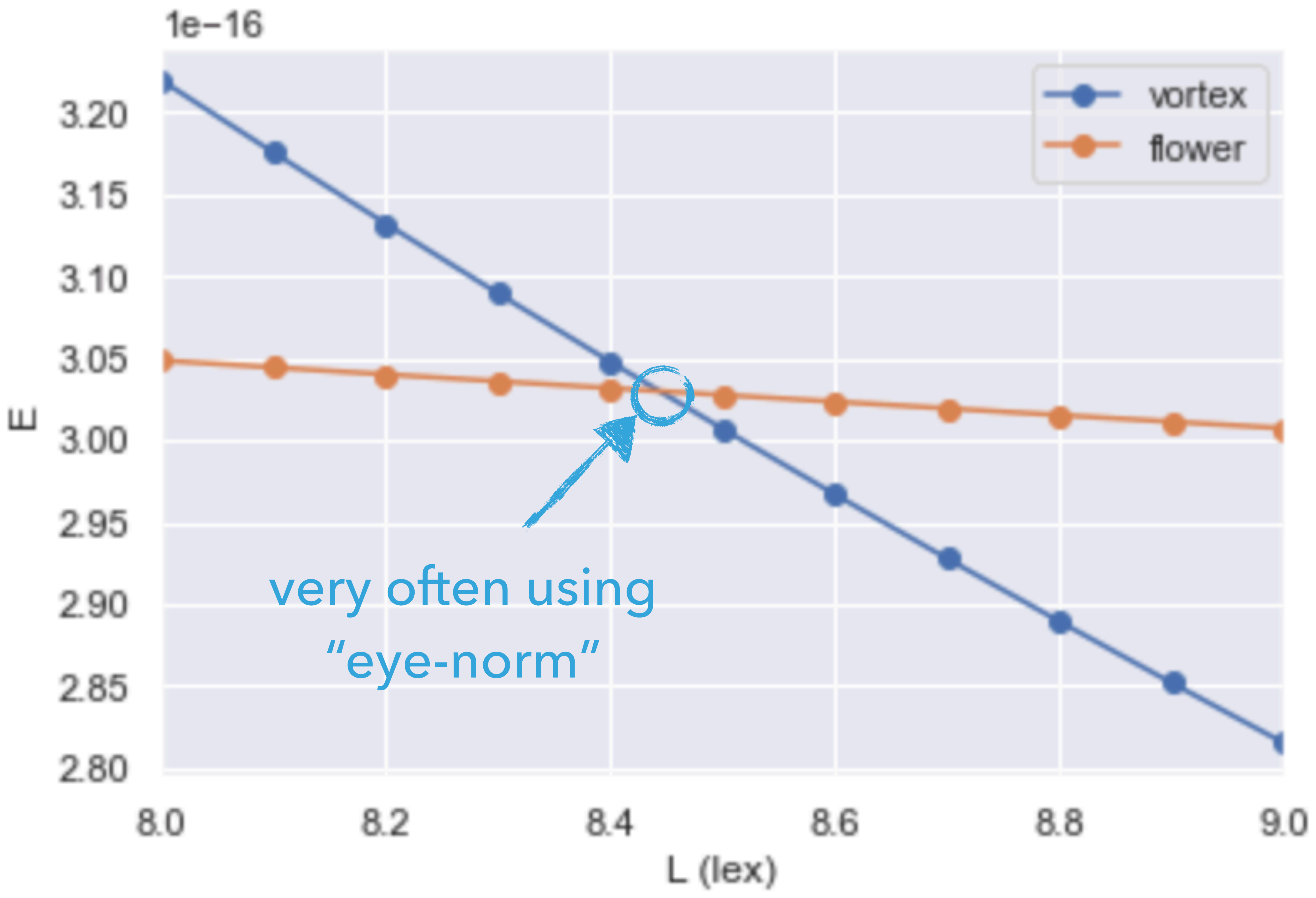
LOOP THROUGH STEPS 1, 2, 3…

L	flower	vortex
8.0	?	3.23×10^{-16}
8.1	?	?
8.2	?	?
8.3	?	?
8.4	?	?
8.5	?	?
8.6	?	?
8.7	?	?
8.8	?	?
8.9	?	?
9.0	?	?

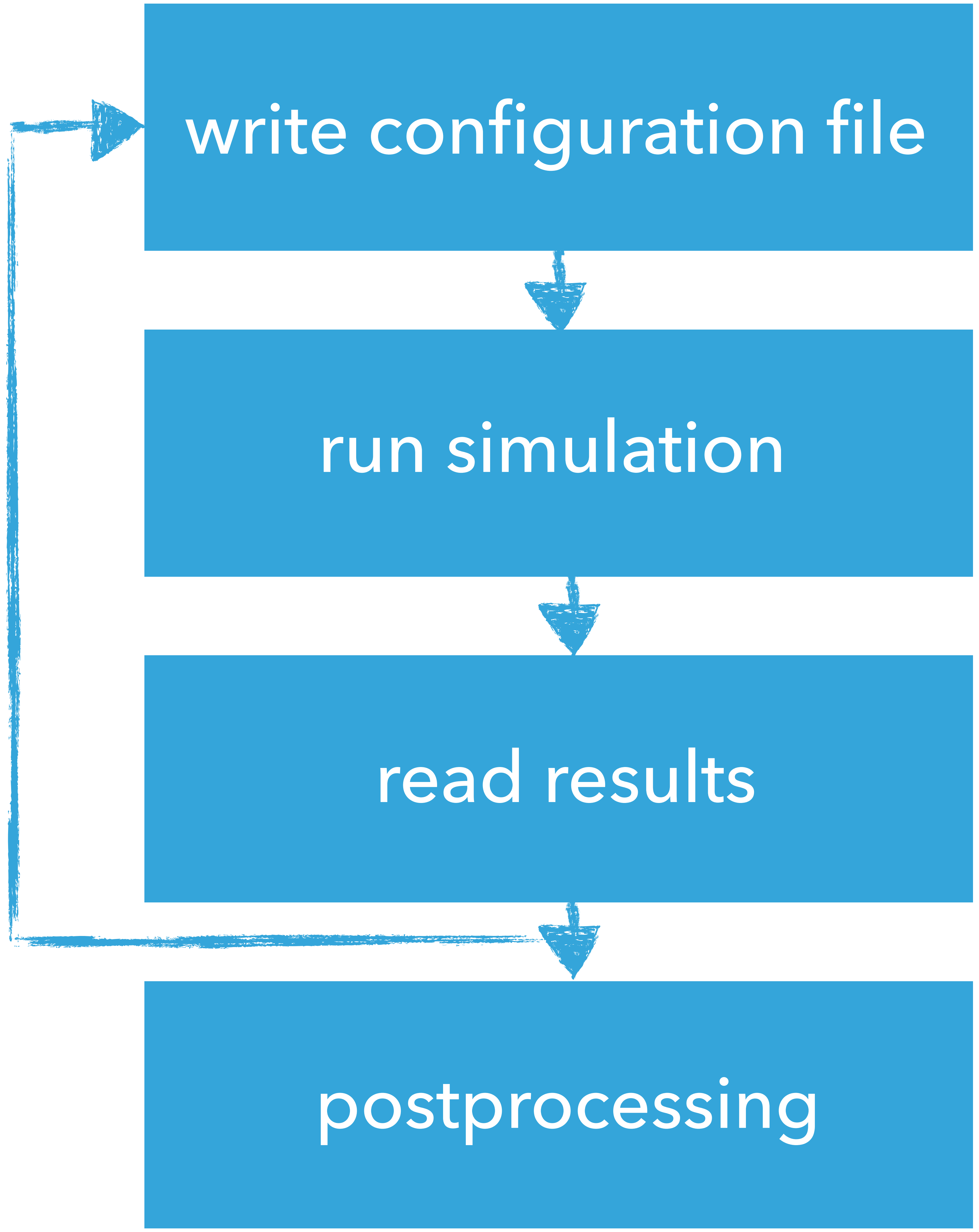


LAST STEP: POSTPROCESSING

- After we obtained all data points, we **plot the results and find crossing**.
- For this step, we often use **separate** plotting scripts or graphical user interface (GUI).

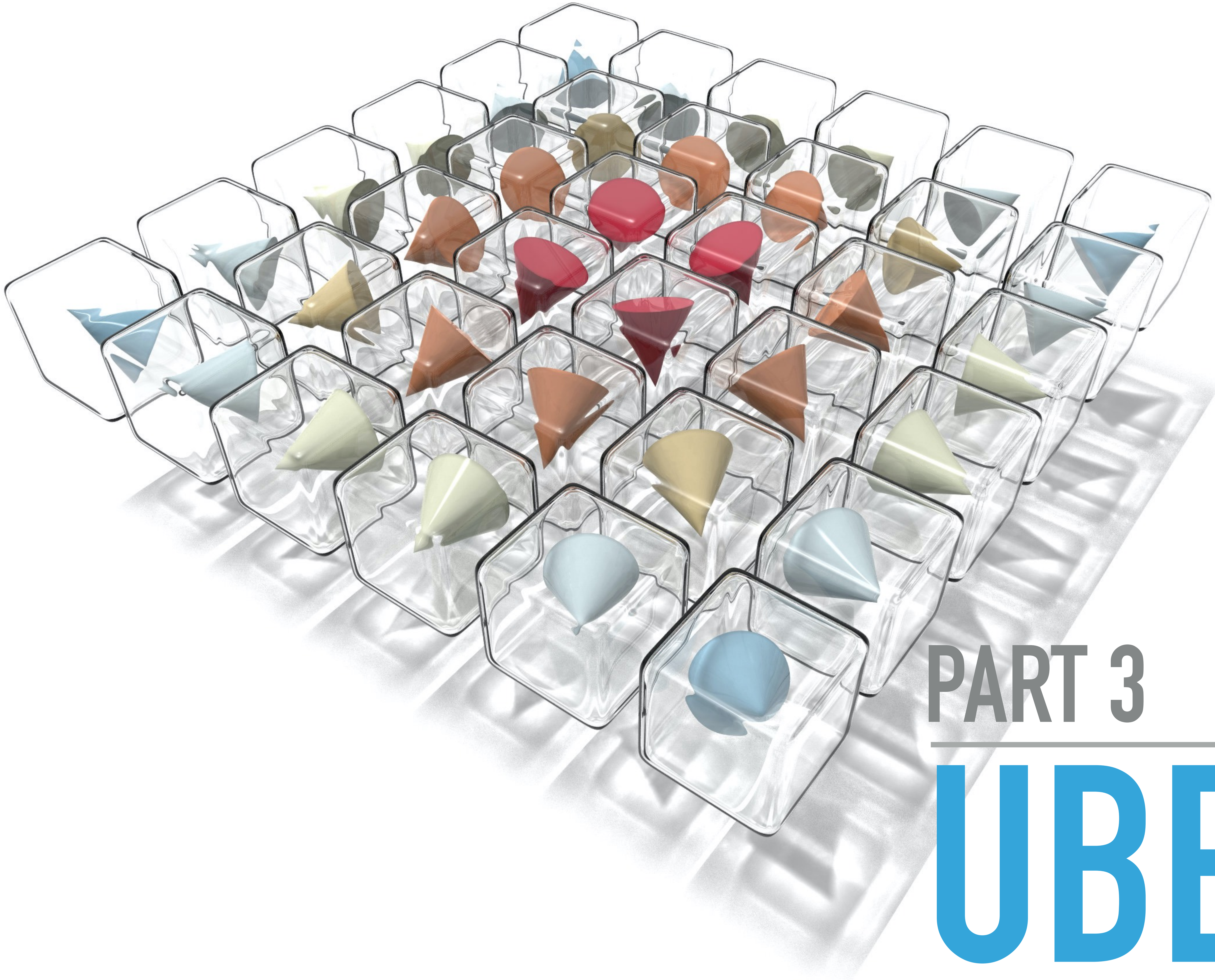


WORKFLOW SUMMARY



WHAT COULD BE THE PROBLEMS WITH THIS WORKFLOW?

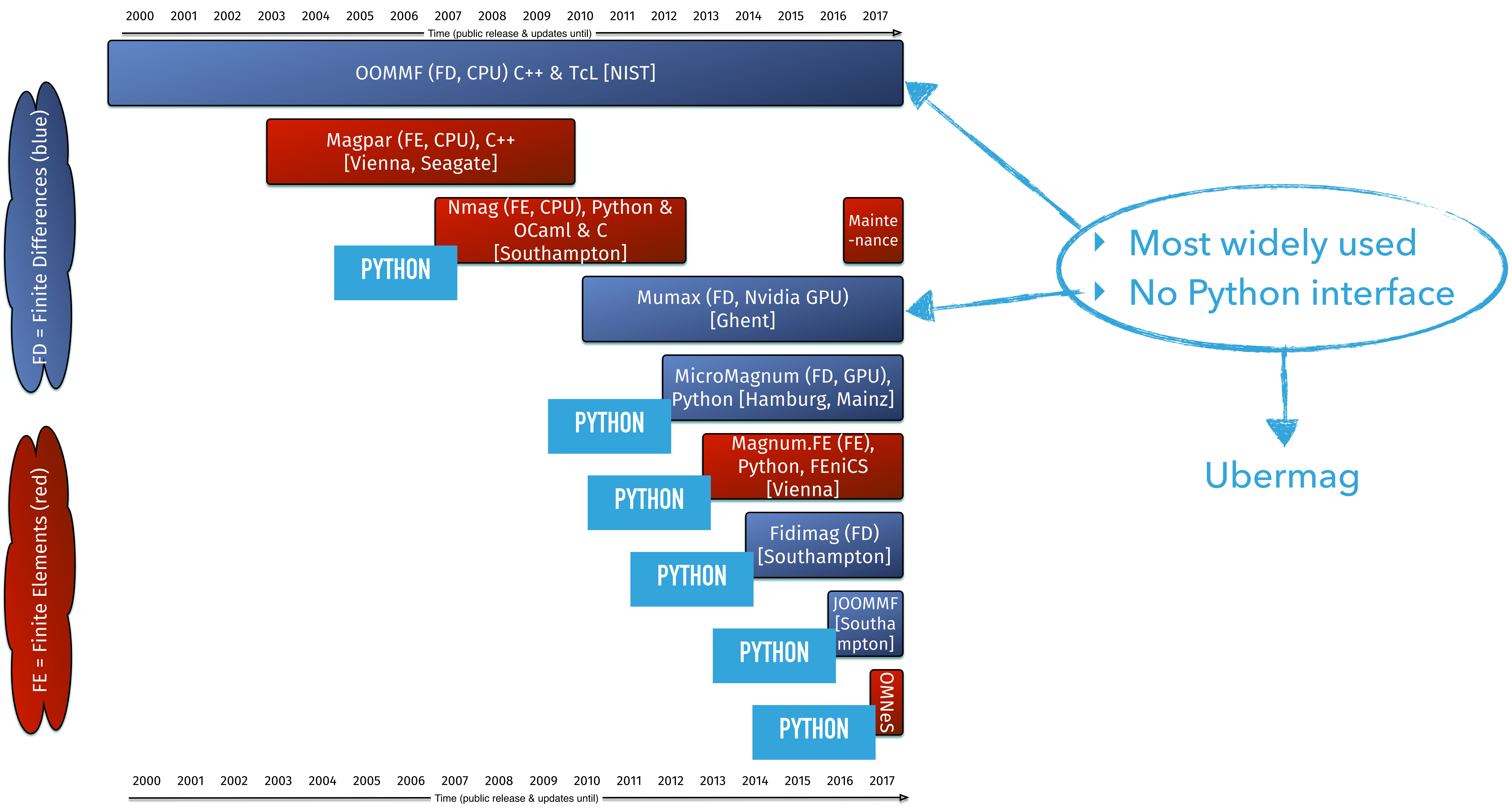
1. **Time** consuming
    - It requires a lot of user input
2. **Keeping log** of all steps that were run and in what order
    - I clicked here, then I changed that, then I fixed that…
3. Separate **postprocessing scripts**
    - Every group has their own scripts with different dependencies
4. **Sharing** the exact workflow
5. **Reproducibility**
6. **Very difficult to automate**
7. Very **steep learning curve**



PART 3

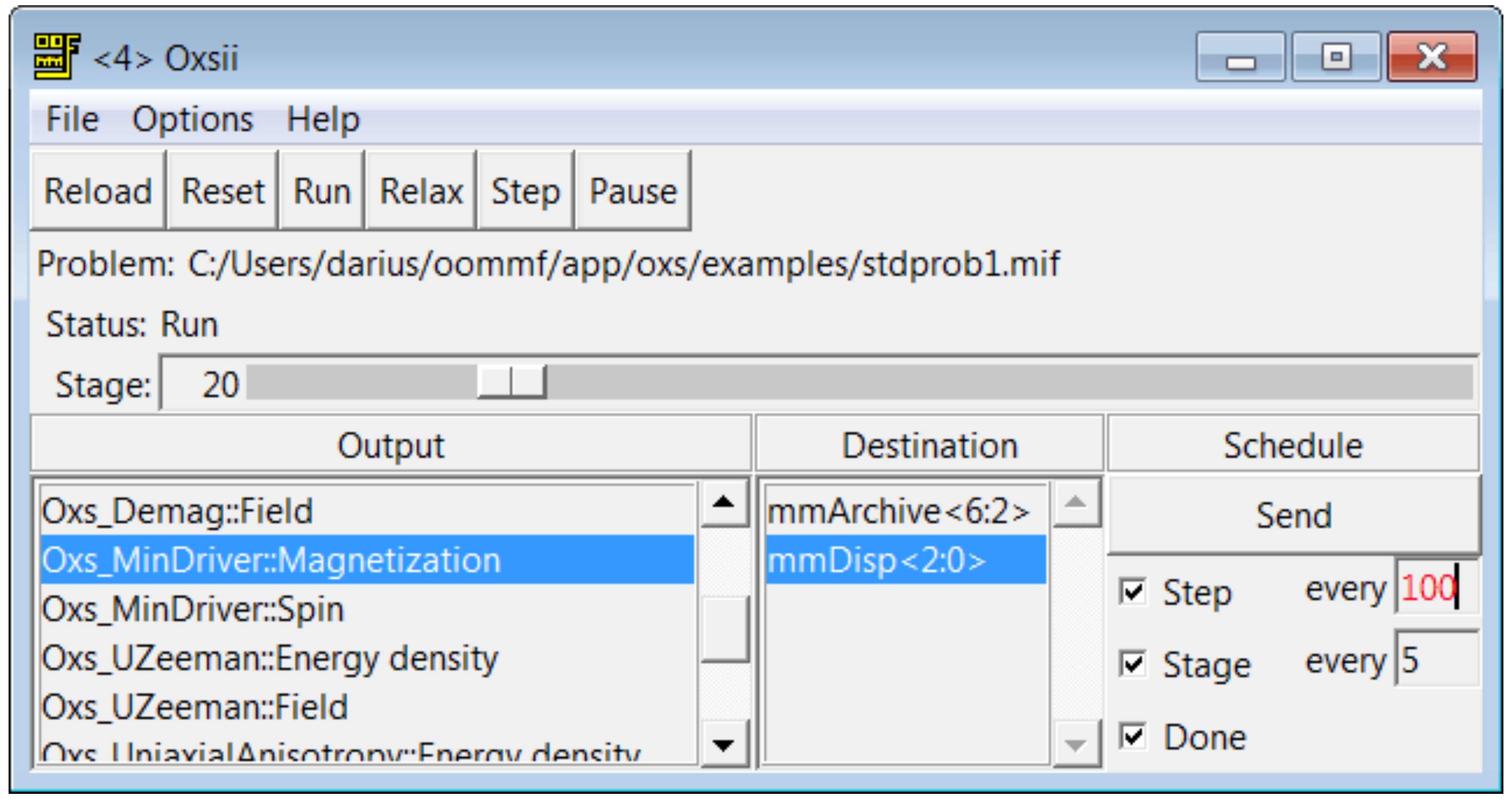
UBERMAG

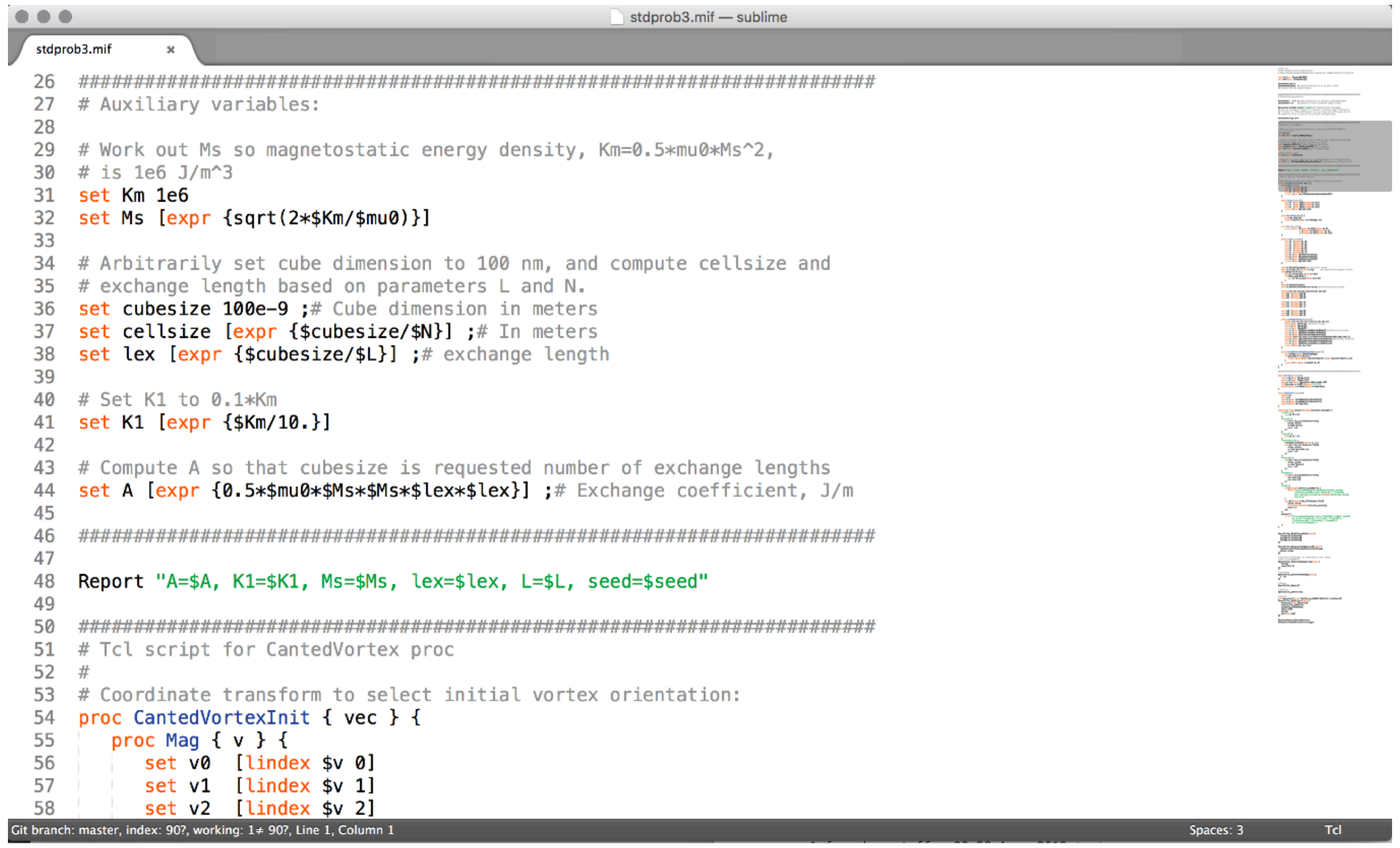
OVERVIEW OF (SOME) MICROMAGNETIC SIMULATORS



WHY DID WE START WITH OOMMF?

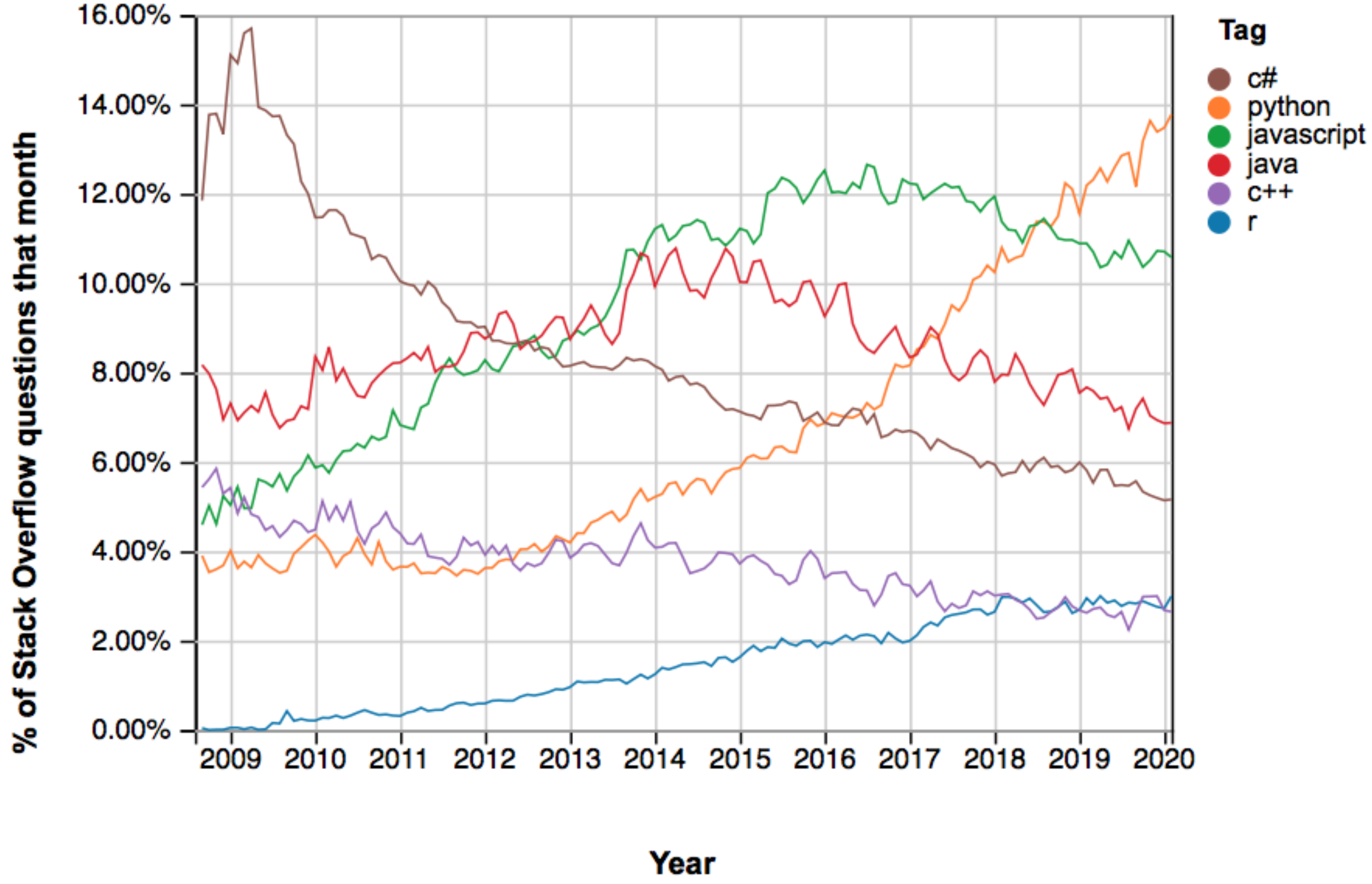
- Probably **the most widely used** micromagnetic simulation tool
- Developed at **NIST**, US, **since ~1998** by Michael Donahue & Don Porter
- **Cited over 2200** times in scientific publications
- Written in **C++** & **Tcl**
- **Finite-difference** code
- Very often **used for comparisons** between codes
- https://math.nist.gov/oommf/





WHY DID WE CHOOSE PYTHON?

- **Modern** programming language
- The language core is **easy to read and easy to learn**
  - Easily accessible to students
- Increasingly popular in **software engineering**
- The most popular in **computational and data science**
- Very **well documented** and well supported
- Interpreted language
- www.python.org



Source: https://towardsdatascience.com

SCIENTIFIC PYTHON ECOSYSTEM

- **numpy**: linear algebra
- **scipy**: numerical analysis
- **matplotlib**: 2d (and some 3d) plotting
- **pandas**: big data for Python
- **scikit-learn**: machine learning
- **Jupyter Notebook**

- No need to reinvent the wheel.

JUPYTER

- **Executable document**
- Text, equations, images, code, and results in a **single document**
- Easily **shared**
- Easily **reproducible**
- Hosted in **web browser**
- Can be **run in the cloud** (Binder)
- www.jupyter.org

UBERMAG

"… provides Python interface to OOMMF and mumax3 (for now), exposes micromagnetic simulations to Python's scientific ecosystem, and embeds them into Jupyter notebook."