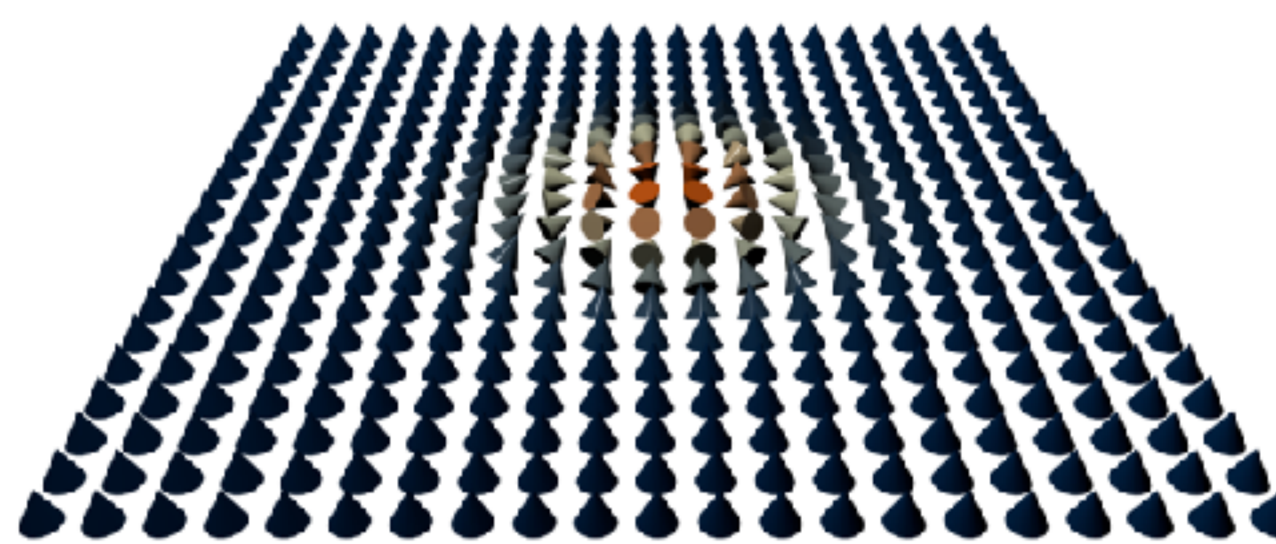
A case study of computational science in Jupyter notebooks

Micromagnetic VRE - Hans Fangohr
Brussels, 26 April 2017



Financial support from OpenDreamKit
Horizon 2020, European Research
Infrastructures project (#676541)
http://opendreamkit.org

Overview

- What is micromagnetics?
- State-of-the-art micromagnetics simulation tool
- Beyond state-of-the art: micromagnetic VRE
- Summary

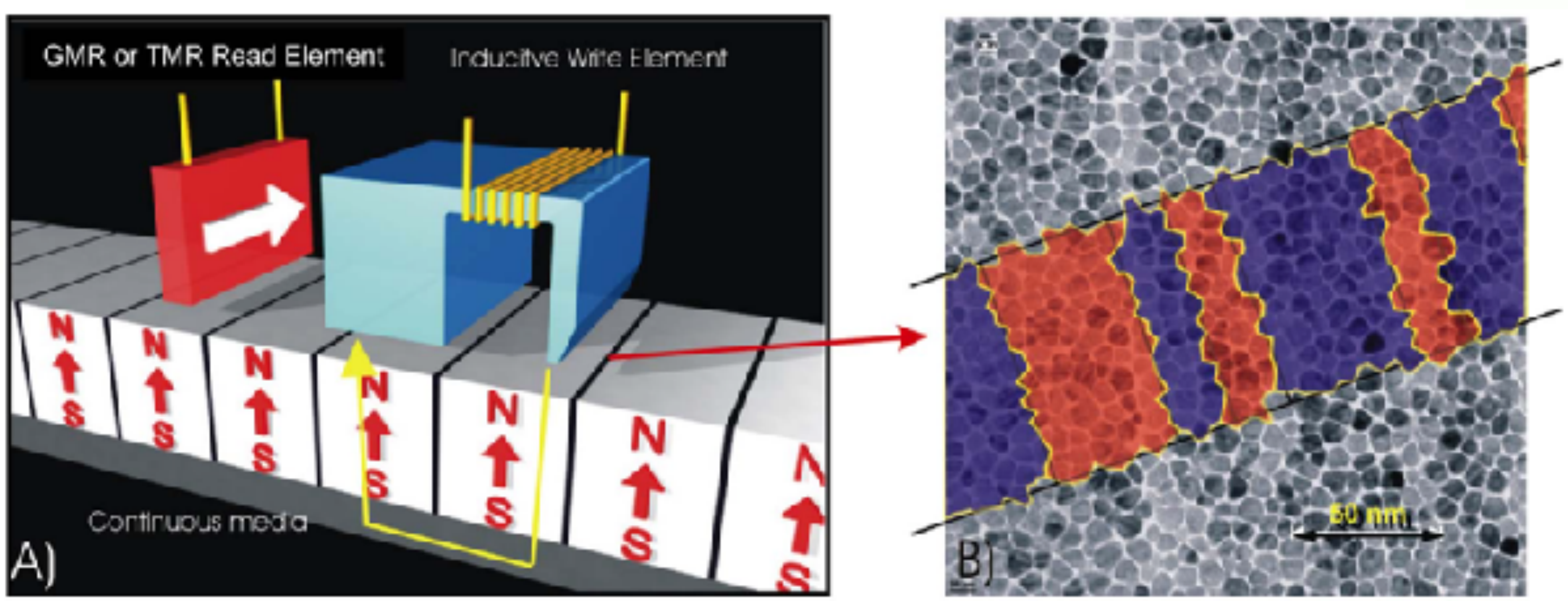
What is micromagnetics ?

- magnetism at small length scales, typically nanometre to micrometre

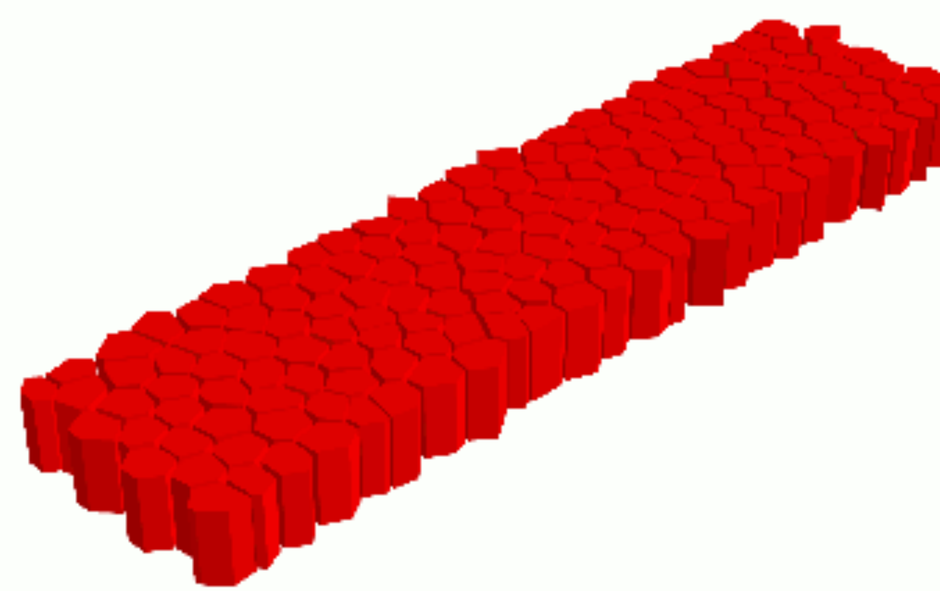
Why magnetic nanostructures?

1. Interesting complex system with tuneable parameters and experiments

2. Applications include
   - magnetic data storage (hard disk)
   - cancer diagnostics and therapy
   - low energy magnetic logic (spintronics, skyrmionics)

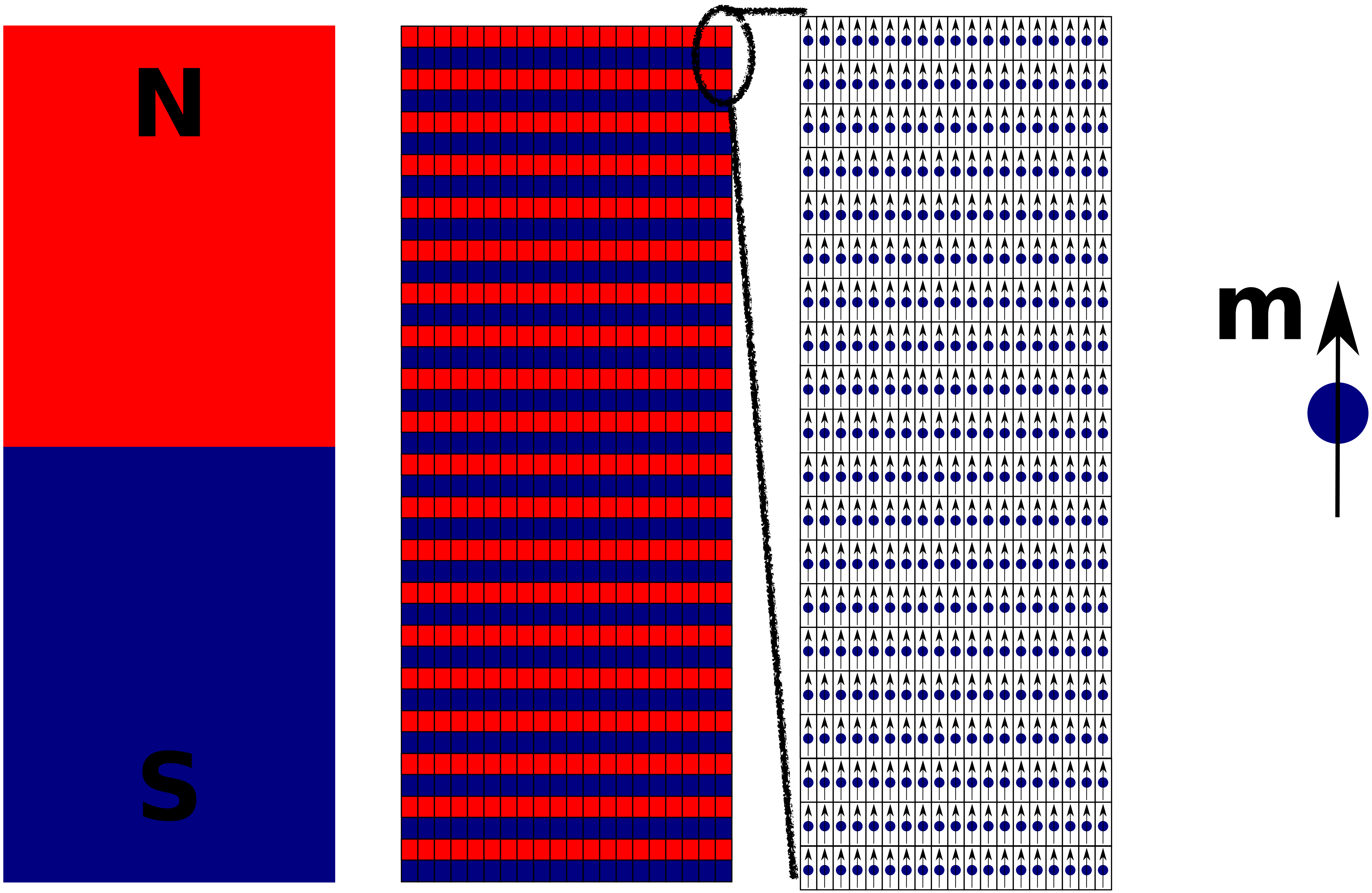


E. Dobisz et. al., Proceedings of IEEE 96, 1836 (2008)



Curtis & Fangohr (2011)

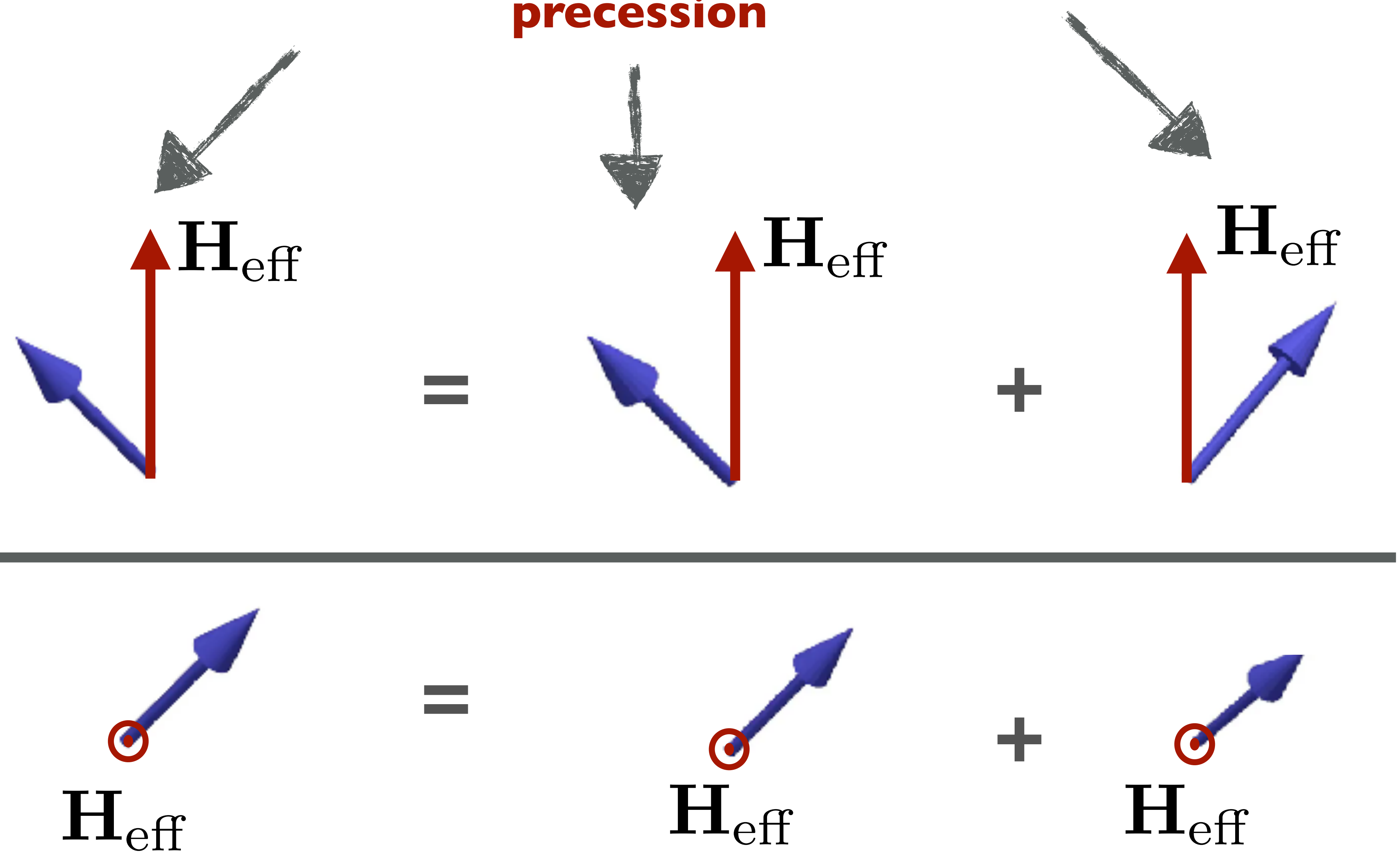
Magnetic moment



Magnetisation dynamics

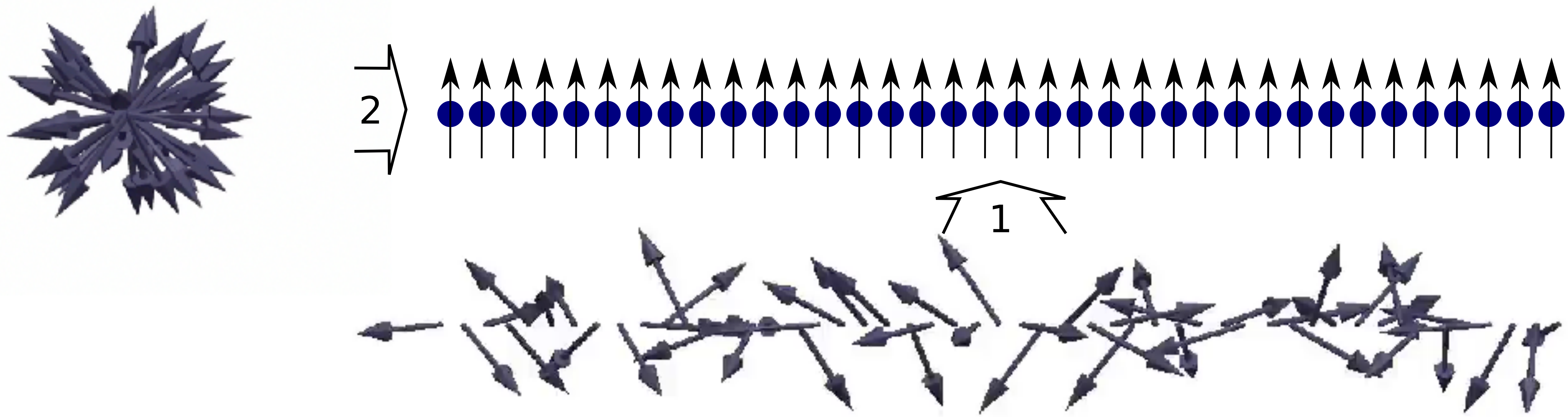
- Landau-Lifshitz-Gilbert (LLG) equation

$$\frac{\partial \mathbf{m}}{\partial t} = \underbrace{\gamma^* \mathbf{m} \times \mathbf{H}_{\text{eff}}}_{\text{precession}} + \overbrace{\alpha \mathbf{m} \times \frac{\partial \mathbf{m}}{\partial t}}^{\text{damping}}$$

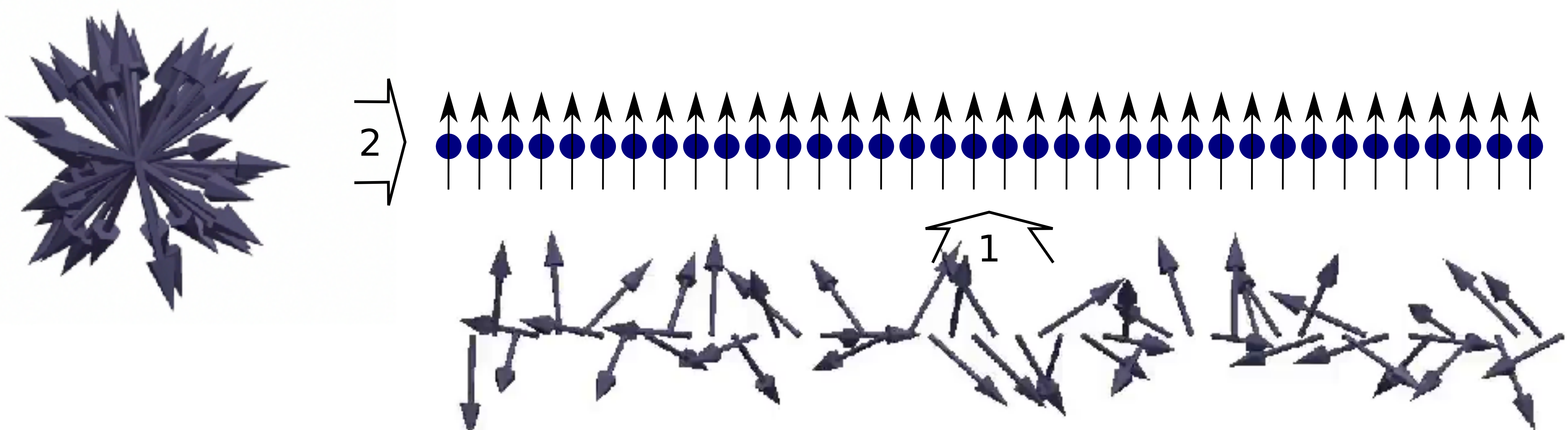


Different types of physics

A: Align the magnetic moment to an external field

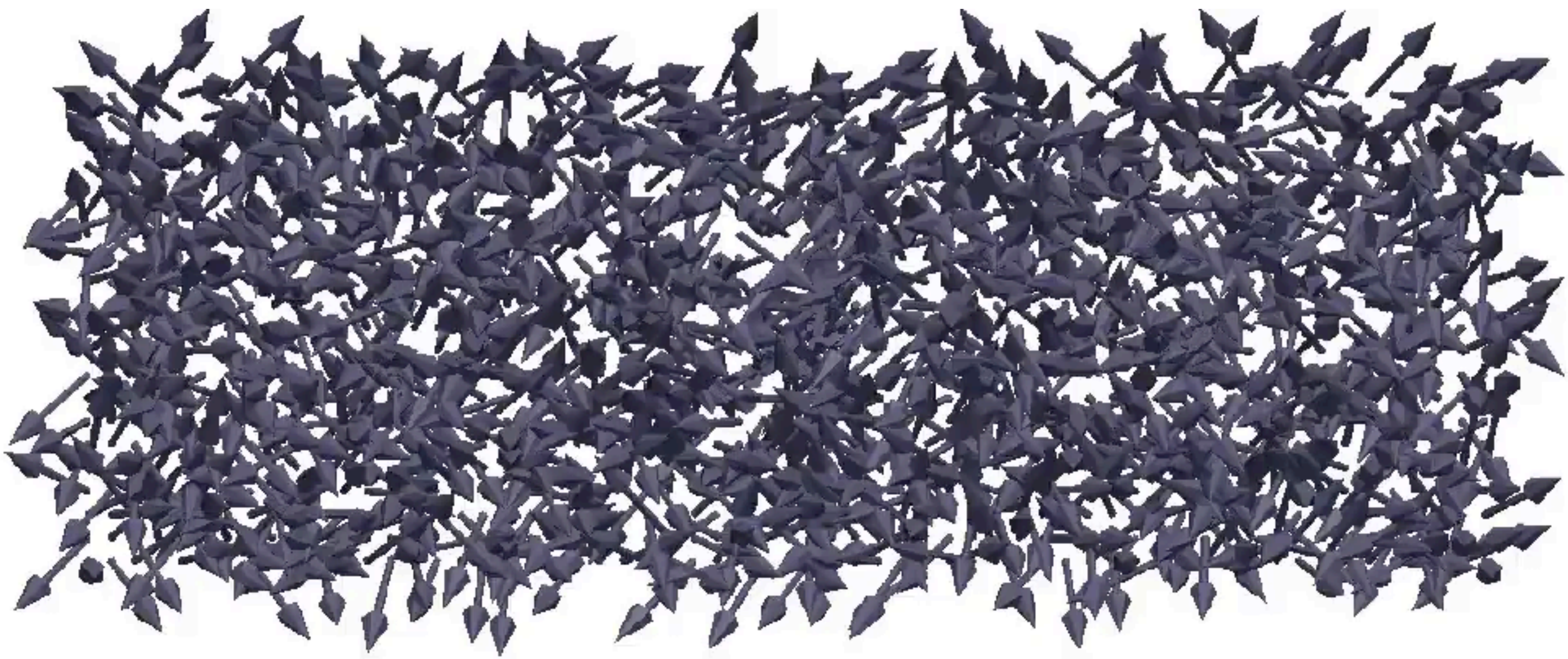


B: Align all magnetic moments to be parallel



More complicated case

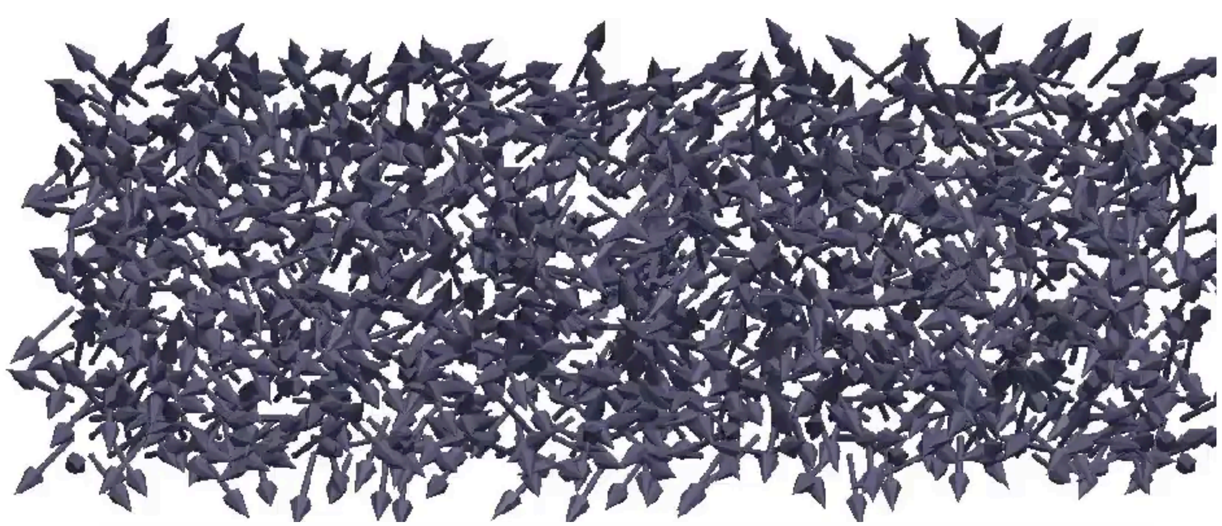
- Two-dimensional sample.
- Four interactions included (Exchange, Zeeman, Anisotropy, Dzyaloshinskii-Moriya energy (DMI))

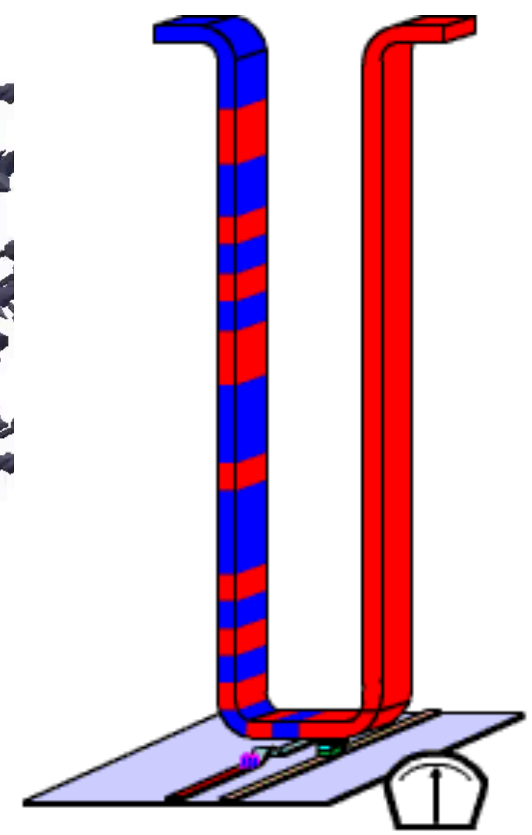


Computational magnetism important

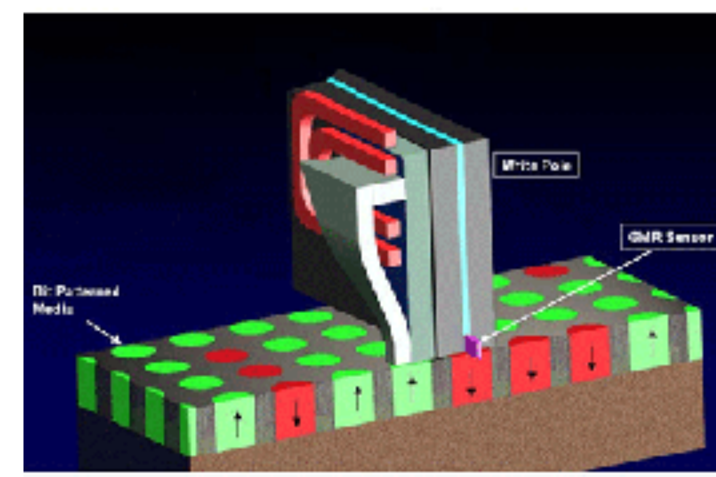
- The number of problems that can be solved analytically is very limited.

- Experimental techniques do not provide enough spatial and temporal resolution.





Parkin, Science, 320, 190 (2008)

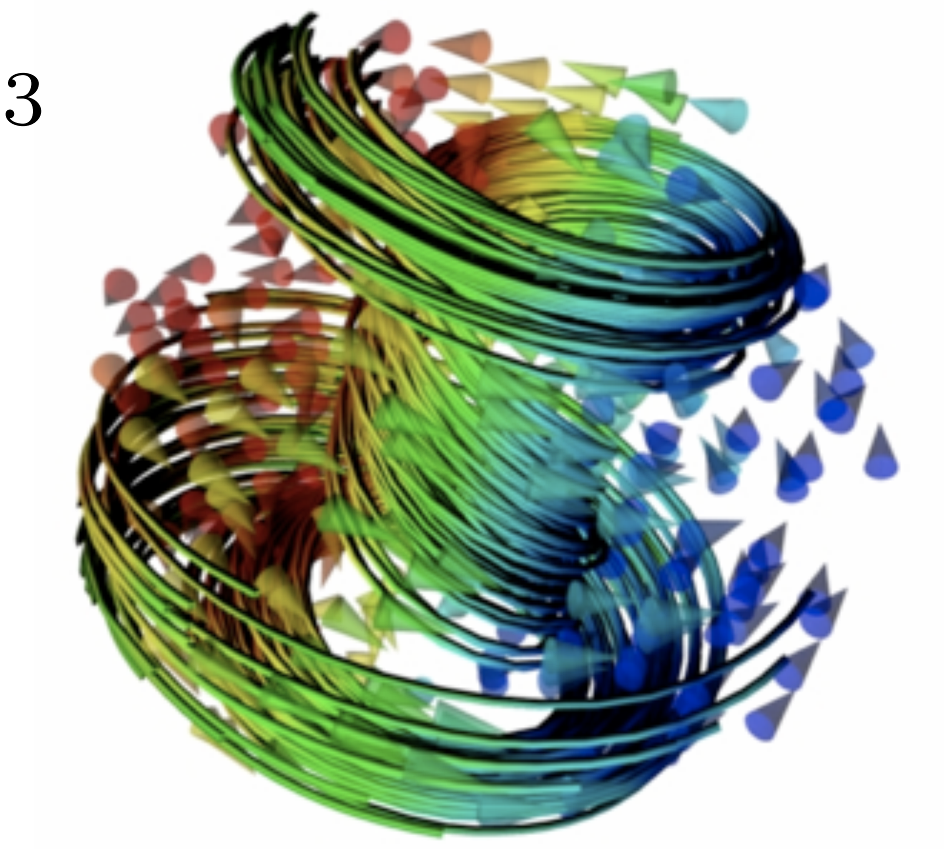


Bit-patterned media (Seagate)

Micromagnetic model

- Coarse graining to go from atoms to continuous magnetisation, known as *micromagnetic model*
- Magnetisation in sample V is described by a continuous *vector field* $\mathbf{m}(\mathbf{r})$:

$$\mathbf{m} : V \mapsto \mathbb{R}^3 \qquad V \subset \mathbb{R}^3$$



- We have an equation of motion

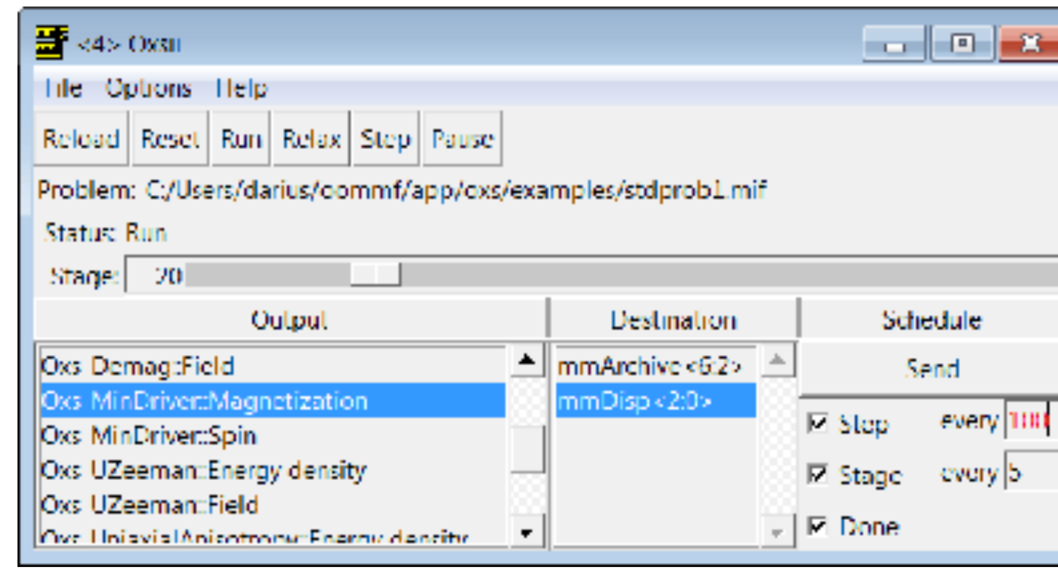
$$\frac{\partial \mathbf{m}}{\partial t} = \mathbf{f}(\mathbf{m})$$

- $\mathbf{f}$ is complicated, involves PDEs

State of the art micromagnetic simulation tool

Object Oriented MicroMagnetic Framework (OOMMF)

- Probably the most widely used simulation tool
- Developed at NIST, USA
- Cited over 2200 times in scientific publications
- Written in C++, some Tcl glue / interface

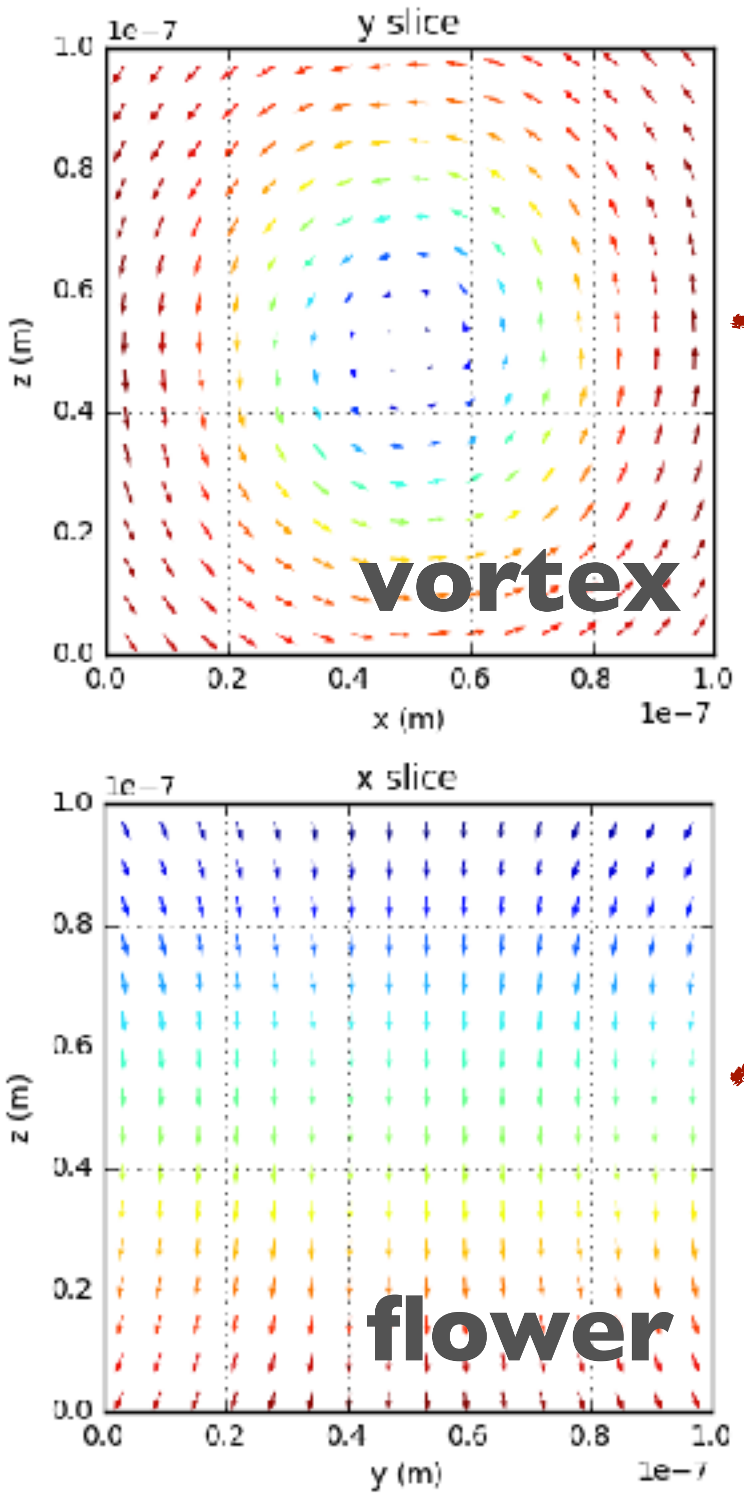


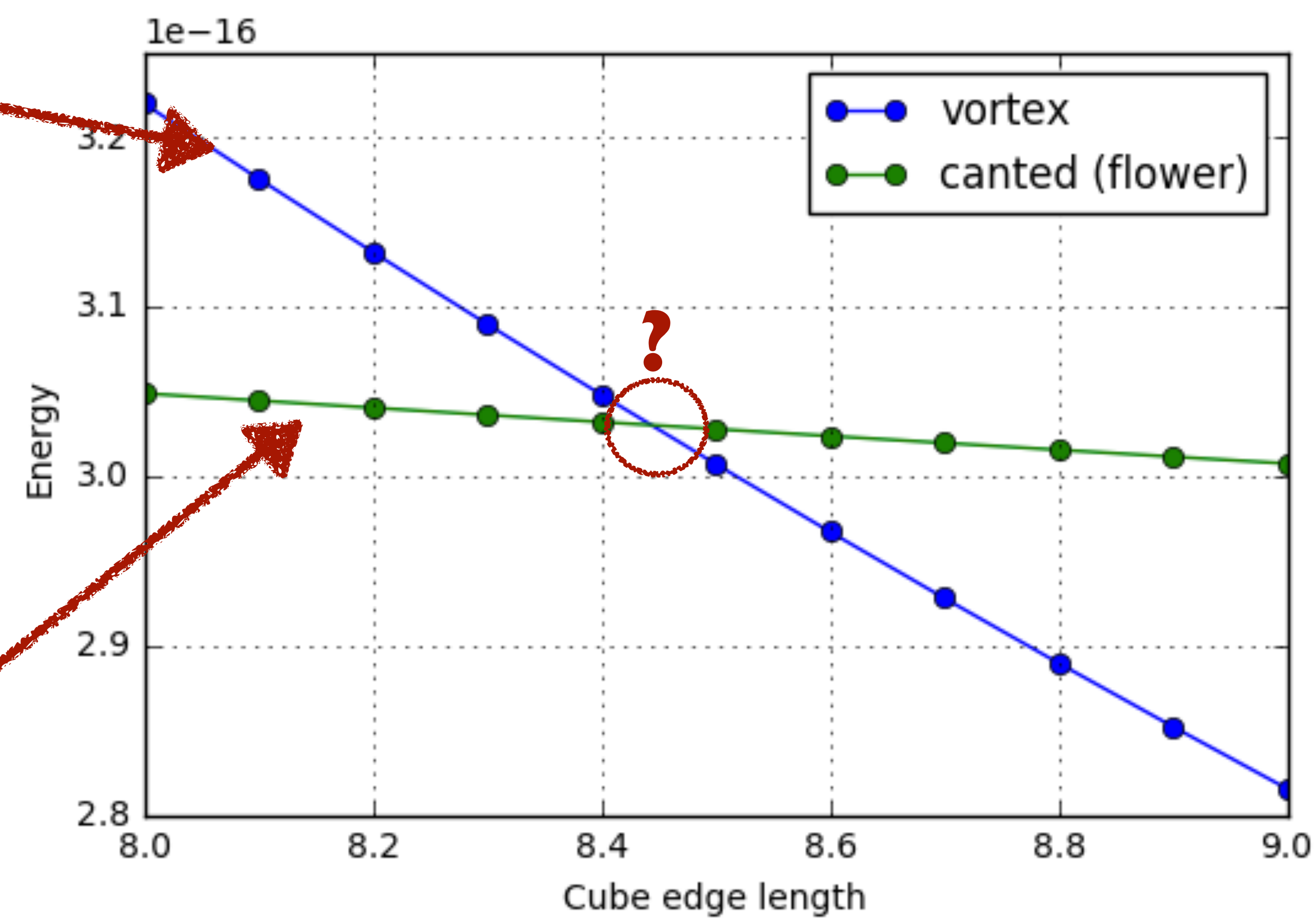
GUI

Tcl config file

Research workflow example

For what cube edge length have vortex and flower states the same energy?

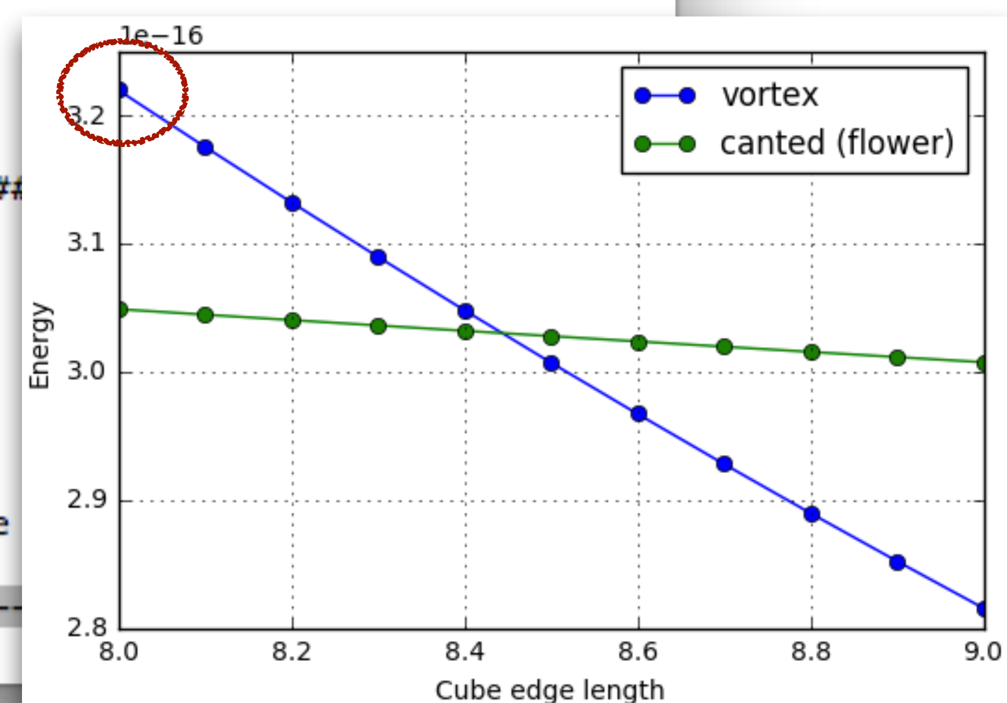




Step 1: write simulation configuration

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
# exchange length based on parameters L and N.
-uu-:---F1  stdprob3.mif    Top L1     (Fundamental)-------------
```



Step 1: write simulation configuration

examples — emacs stdprob3.mif — 95×37

```
      set vx [expr {$a11*$nvx+$a12*$nvy+$a13*$nvz}] ;# Rotate, backside
      set vy [expr {$a21*$nvx+$a22*$nvy+$a23*$nvz}]
      set vz [expr {$a31*$nvx+$a32*$nvy+$a33*$nvz}]
      return [list $vx $vy $vz]
   }

   proc CantedVortexBaseCompute { x y z } {
      set normsq [expr {$x*$x+$y*$y}]
      if {$normsq <= 0.0125} {
         return [list [expr {0.125*rand()}] [expr {0.125*rand()}] 1.0]
      }
      return [list [expr {-1*$y}] $x 0]
   }
}

########################################################################

proc Vortex { x y z } {
   set yrad [expr {2.*$y-1.}]
   set zrad [expr {2.*$z-1.}]
   set normsq [expr {$yrad*$yrad+$zrad*$zrad}]
   if {$normsq <= 0.05} {return "1 0 0"}
   return [list 0.0 $zrad [expr {-1*$yrad}]]
}

proc Twisted { x y z } {
   global pi
   set vx 0
   set vy [expr {sin(($pi/2.)*($x-0.5))}]
   set vz [expr {cos(($pi/2.)*($x-0.5))}]
   return [list $vx $vy $vz]
}

switch [string tolower [lindex $initial_state 0]] {
   "uniform" {
-uu-:---F1  stdprob3.mif    56% L143    (Fundamental)-------------------------------------------
```

Step 2: run simulation

my_project — IPython: Users/mb4e10 — -bash ▸ python — 95×37

```
Marijans-MBP:my_project mb4e10$ ls
stdprob3.mif
Marijans-MBP:my_project mb4e10$ tclsh $OOMMFTCL boxsi +fg stdprob3.mif -exitondone 1
Start: "/Users/mb4e10/my_project/stdprob3.mif"
Options: -exitondone 1 -threads 2
Boxsi version 1.2.1.0
Running on: marijans-macbook-pro.local
OS/machine: Darwin/x86_64
User: mb4e10    PID: 72176
Number of threads: 2
Mesh geometry: 32 x 32 x 32 = 32 768 cells
Checkpoint file: /Users/mb4e10/my_project/sp3-vortex-seed0000.restart
Boxsi run end.
Marijans-MBP:my_project mb4e10$ ls
sp3-vortex-seed0000.odt stdprob3.mif
Marijans-MBP:my_project mb4e10$
```

Step 3: extract data from output file

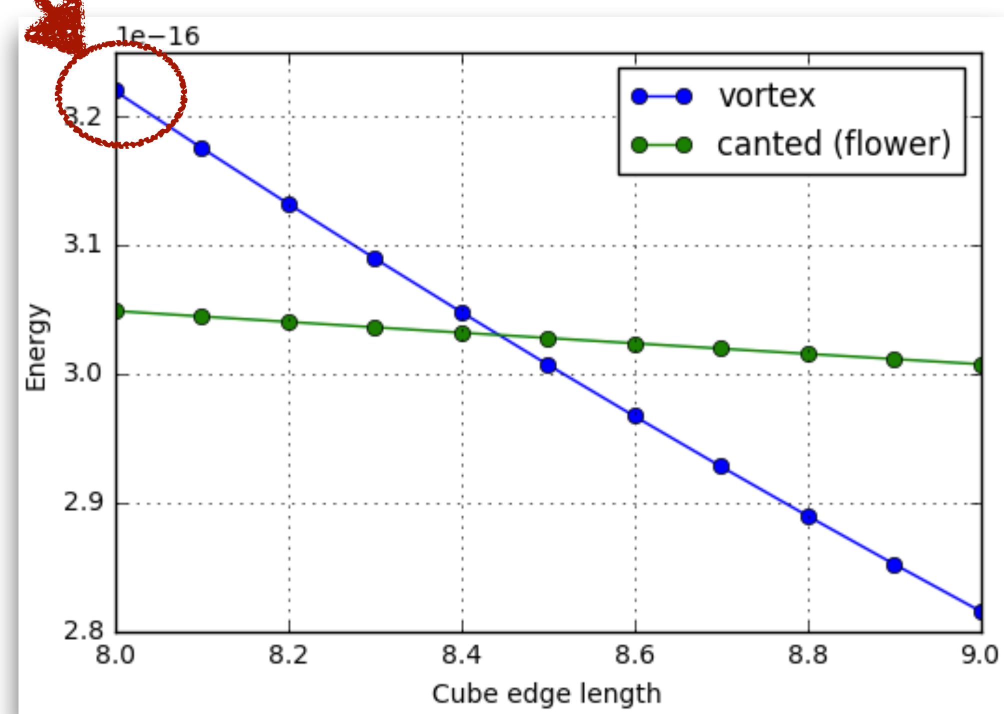
my_project — IPython: Users/mb4e10 — emacs -nw sp3-vortex-seed0000.odt — 95×37

```
# 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")
```

Step 4: gather data, and repeat simulations...

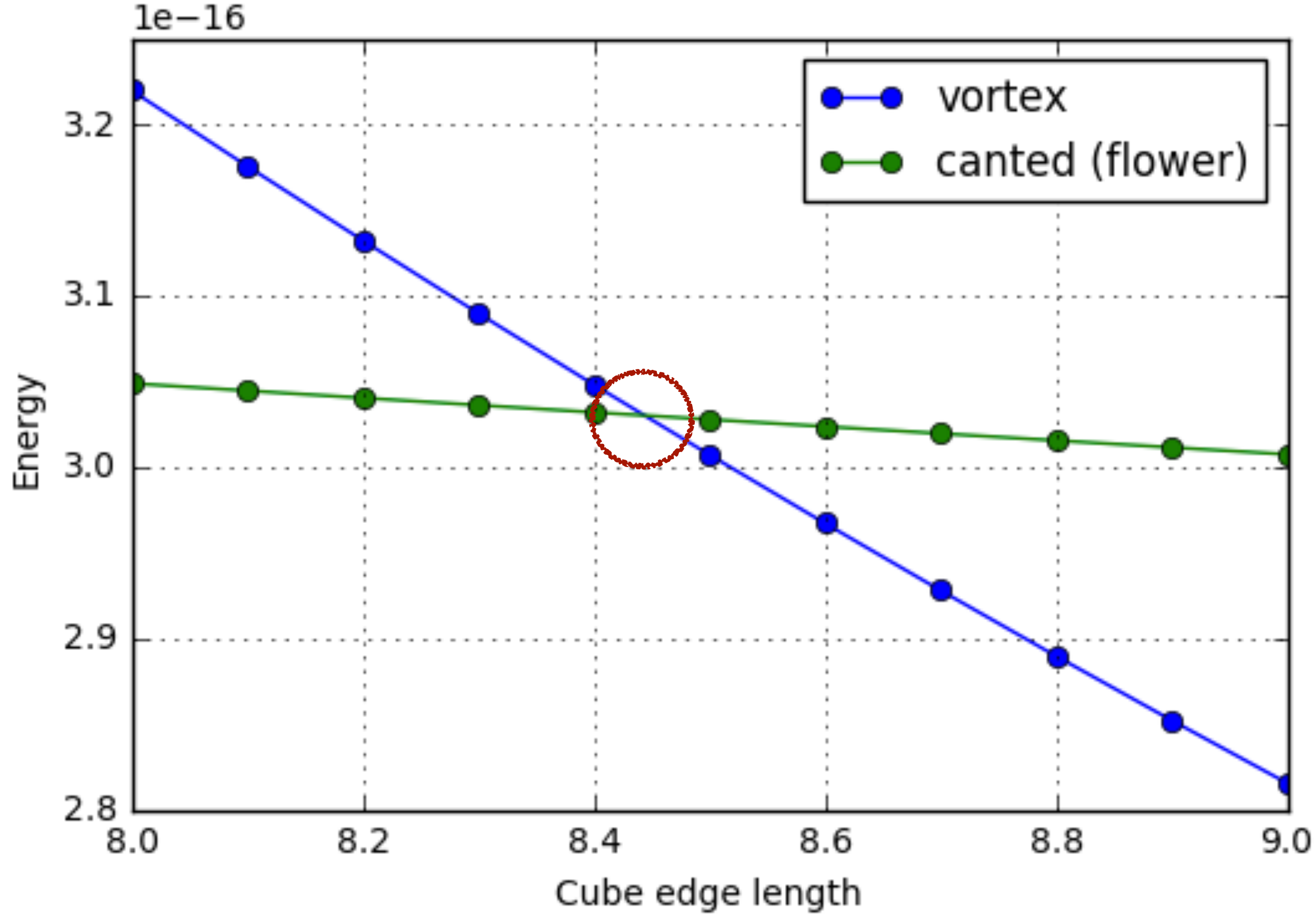
L	flower	vortex
8.0	?	3.23×10^{-16}
8.1	?	?
8.2	?	?
8.3	?	?
8.4	?	?
8.5	?	?
8.6	?	?
8.7	?	?
8.8	?	?
8.9	?	?
9.0	?	?



"Pushing one domino at a time"

Postprocessing

- We plot the data we obtained by running separate plotting scripts or by using some Graphical User Interfaces (Python, MATLAB, Excel, Origin…)



- Find crossing (here at ~8.45).

Issues with (OOMMF) workflow

- Writing config files and extracting data is repetitive, manual process (or requires bash scripting)
- Time consuming; error prone
- Separate post processing and plotting scripts
- Reproducibility?

Jupyter OOMMF

JOOMMF

- Jupyter + OOMMF = JOOMMF
- Micromagnetic Virtual Research Environment (VRE)
- Enable running OOMMF simulations in Jupyter notebook (through Python interface to OOMMF)

Research example (repeated) with Jupyter OOMMF

[Live demo in Notebook: standard_problem3.ipynb,
online at https://github.com/OpenDreamKit/OpenDreamKit.github.io/blob/master/meetings/2017-04-26-ProjectReviewPresentations/joommf/standard_problem3.ipynb]

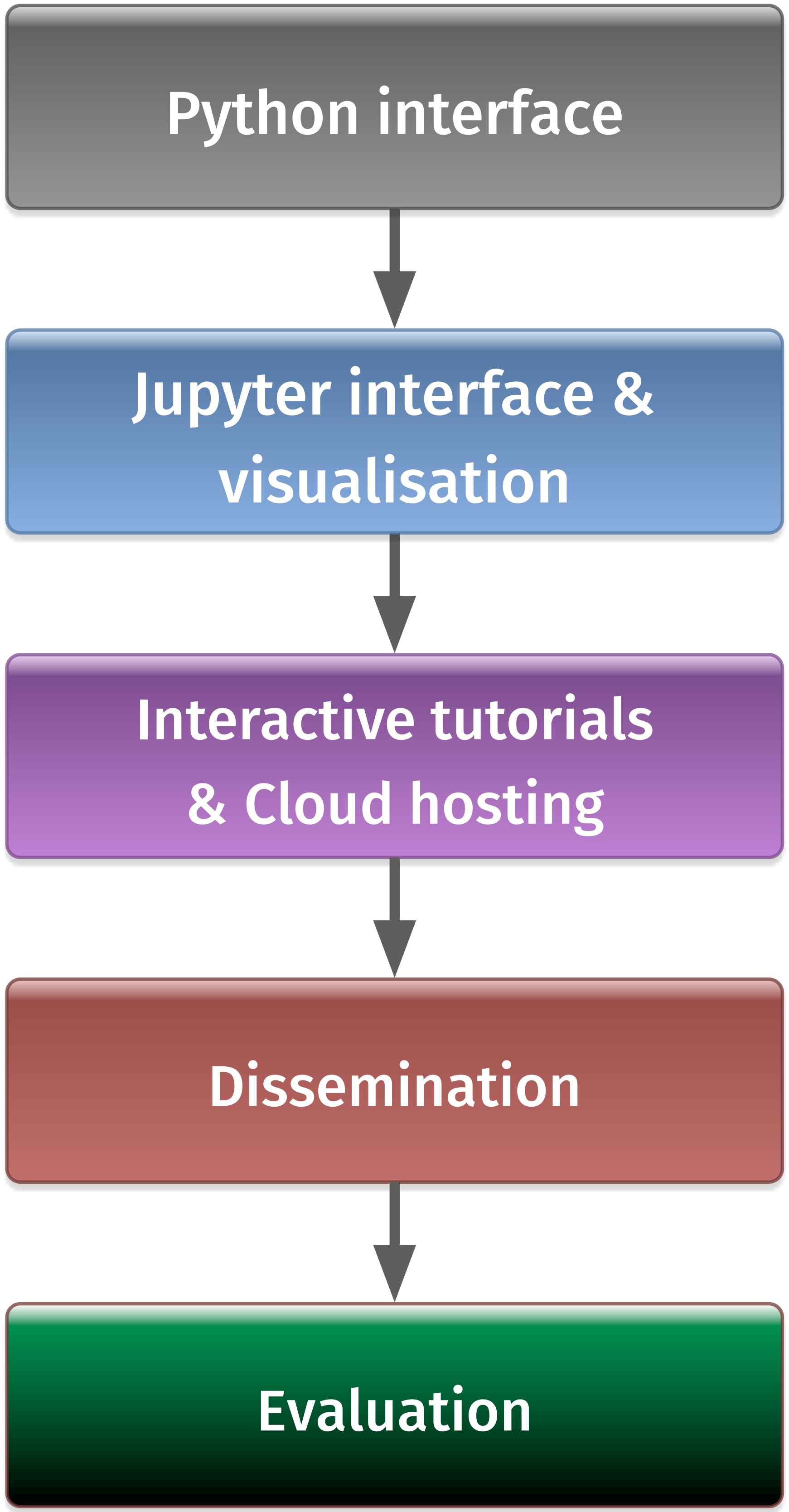
Benefits of JOOMMF

- The entire workflow is contained in a single document, including computation, post processing and visualisation
- Self documenting
- Reproducible: re-execute cells in notebook
- Easy to share & publish

Micromagnetic model integration in VRE

[Live demo in Notebook: micromagneticmodel.ipynb
online at https://github.com/OpenDreamKit/OpenDreamKit.github.io/blob/master/meetings/2017-04-26-ProjectReviewPresentations/joommf/micromagneticmodel.ipynb]

Link to work packages



- WP3 Component architecture — T3.8 - Python
- WP4 User interfaces — T4.11- Jupyter; T4.8 - 3d vis; T4.13 - interactive d; T4.14 - cloud
- WP2 Dissemination — T2.8 - workshops
- WP7 Social Aspects — T7.4 - evaluation

Summary JOOMMF

- Micromagnetic Virtual Research Environment (VRE) allows us to have documentation, models, code, code outputs, in a single file
- Python interface to OOMMF supports component-based approach: can combine OOMMF with the tools from Python ecosystem
- Improved effectiveness and reproducibility: not affordable for individual research groups but enabled by OpenDreamKit
- All open source (joommf.github.io)
- Micromagnetic VRE is specialised VRE built from the VRE Toolkit of OpenDreamKit, and
- Demonstrates how computational mathematics underpins science and engineering

- To cite Jupyter-OOMMF, please use

  Marijan Beg, Ryan A. Pepper, Hans Fangohr:
  User interfaces for computational science: a domain specific language for OOMMF embedded in Python,
  American Institute of Physics, Advances 7, 056025 (2017)
  http://dx.doi.org/10.1063/1.4977225
  also available online https://arxiv.org/abs/1609.07432

- Source code: http://joommf.github.io