FACULTY OF SCIENCES

DEPARTMENT OF SOLID STATE SCIENCES
DYNAMAT GROUP

MUMAX3-WORKSHOP SESSION 3

Dr. Jonathan Leliaert,
Dr. Jeroen Mulkers

GHENT UNIVERSITY

IEEE MAGNETICS

NVIDIA

SPINTRONICS

SCHEDULE

Monday 08/31, 6PM-8PM CET

Session 1: general introduction to micromagnetics in mumax3

Session 2: mumax3 ecosystem, workflow and a first simulation

Monday 09/07, 6PM-7:30PM CET

Session 3: basic examples

>homework

Monday 09/14, 6PM-7:30PM CET

Session 4: advanced features and more extensive examples

GHENT UNIVERSITY

TABLE OF CONTENTS

GHENT UNIVERSITY

INTRODUCTION
HOW TO RUN MUMAX3

Windows/linux PC with NVIDIA GPU

Preferred option for small simulations and to learn about mumax3

Web GUI should work out of the box

Windows/linux server with NVIDIA GPUs

Preferred option for big simulations, or large batches of simulations

Setting up the Web GUI might be possible but requires some knowledge on network ports

Google collaboratory session

To fiddle around with mumax3 if you do not have access to an NVIDIA GPU

Free for gmail users, no Web GUI

https://colab.research.google.com/github/JeroenMulkers/mumax3-tutorial/blob/master/mumax3.ipynb

See session 2 of this workshop for more information on how to run mumax3

https://mumax.ugent.be/mumax3-workshop

GHENT UNIVERSITY

INTRODUCTION

ABOUT THE EXAMPLES

- The examples can either be downloaded from:

  https://mumax.ugent.be/mumax3-workshop

- The examples are written for mumax3.10
- Ordered from easy to hard
- Typical execution times for the examples are given for the NVIDIA Titan RTX

PHASE DIAGRAM: STANDARD PROBLEM 3

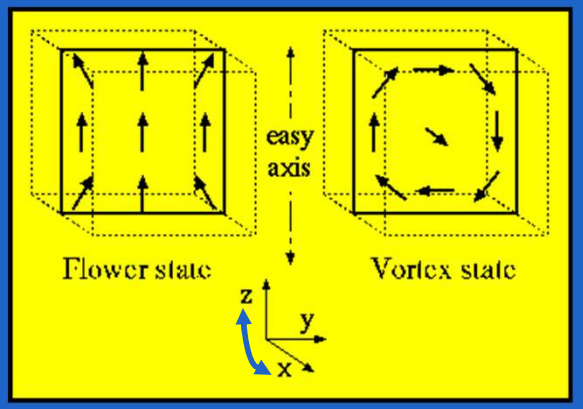
System

Ferromagnetic cube with a uniaxial anisotropy

$$K_u = 0.1\,K_m = 0.1\frac{\mu_0 M_s^2}{2}$$

Task

Calculate the quasi uniform domain limit of a cubic magnetic particle. This is the size *L* of equal energy for the flower state and the vortex state.



Input file: session3_example1a.txt
Titan RTX: 52 s

Input file: session3_example1b.txt
Titan RTX: 15 s

GHENT UNIVERSITY

STANDARD PROBLEM 3

USING UNITLESS PARAMETERS

The total energy density of the ferromagnetic cube with an easy axis along the x direction and uniaxial anisotropy constant $K_u = 0.1\ K_m = 0.1\ (\mu_0 M_s^2/2)$

$$\varepsilon = A(\nabla \boldsymbol{m})^2 - K_u m_x^2 - \frac{\mu_0 M_s^2}{2} \boldsymbol{m}(\boldsymbol{r}) \cdot \int \tilde{N}(\boldsymbol{r} - \boldsymbol{r}') \boldsymbol{m}(\boldsymbol{r}') dV$$

$\frac{\varepsilon}{K_m} \rightarrow \varepsilon$

$l_{ex}\nabla \rightarrow \nabla$

Let's express energy density in terms of the magnetostatic energy density $K_m = \mu_0 M_s/2$ and distances in terms of the exchange length $l_{ex} = \sqrt{A/K_m}$

$$\varepsilon = (\nabla \boldsymbol{m})^2 - 0.1 m_x^2 - \int \boldsymbol{m}(\boldsymbol{r}) \tilde{N}(\boldsymbol{r} - \boldsymbol{r}') \boldsymbol{m}(\boldsymbol{r}') dV$$

Now we can use the following values for the material parameters to solve the specified problem

$$A = 1 \qquad K_u = 0.1 \qquad M_s = \sqrt{2/\mu_0}$$

GHENT UNIVERSITY

STANDARD PROBLEM 3

```
N := 64
setgridsize(N,N,N)

Lmin := 8.0
Lmax := 9.0
Lstep := 0.05

L := Lmin
setcellsize(L/N,L/N,L/N)

Msat = sqrt(2/mu0)
Aex = 1.0
Ku1 = 0.1
anisU = vector(1,0,0)

m = vortex(1,-1)

tableadd(E_total)
tableaddvar(L,"L","")

for L=Lmin ; L<=Lmax; L+=Lstep {
    setcellsize(L/N,L/N,L/N)
    minimize()
    tablesave()
}
```

Setting the grid size

```
N := 64
setgridsize(N,N,N)
```

Notes:

- Total number of cells: $64 \times 64 \times 64 = 262144$
- Grid dimension size $64 = 2^6$ is optimal for the FFT algorithm
- Fixed grid dimensions, cell size will be used to vary the cube size

GHENT UNIVERSITY

STANDARD PROBLEM 3

```
N := 64
setgridsize(N,N,N)

Lmin := 8.0
Lmax := 9.0
Lstep := 0.05

L := Lmin
setcellsize(L/N,L/N,L/N)

Msat = sqrt(2/mu0)
Aex = 1.0
Ku1 = 0.1
anisU = vector(1,0,0)

m = vortex(1,-1)

tableadd(E_total)
tableaddvar(L,"L","")

for L=Lmin ; L<=Lmax; L+=Lstep {
    setcellsize(L/N,L/N,L/N)
    minimize()
    tablesave()
}
```

Defining the cube size range

```
Lmin := 8.0
Lmax := 9.0
Lstep := 0.05

L := Lmin
setcellsize(L/N,L/N,L/N)
```

Cell sizes small enough?

$\frac{L}{N}$: $0.125\, l_{ex} \rightarrow 0.141\, l_{ex}$ 

GHENT UNIVERSITY

STANDARD PROBLEM 3

```
N := 64
setgridsize(N,N,N)

Lmin := 8.0
Lmax := 9.0
Lstep := 0.05

L := Lmin
setcellsize(L/N,L/N,L/N)

Msat = sqrt(2/mu0)
Aex = 1.0
Ku1 = 0.1
anisU = vector(1,0,0)

m = vortex(1,-1)

tableadd(E_total)
tableaddvar(L,"L","")

for L=Lmin ; L<=Lmax; L+=Lstep {
    setcellsize(L/N,L/N,L/N)
    minimize()
    tablesave()
}
```

The unitless material parameters

```
Msat = sqrt(2/mu0)
Aex = 1.0
Ku1 = 0.1
anisU = vector(1,0,0)
```

GHENT UNIVERSITY

STANDARD PROBLEM 3

```
N := 64
setgridsize(N,N,N)

Lmin := 8.0
Lmax := 9.0
Lstep := 0.05

L := Lmin
setcellsize(L/N,L/N,L/N)

Msat = sqrt(2/mu0)
Aex = 1.0
Ku1 = 0.1
anisU = vector(1,0,0)

m = vortex(1,-1)

tableadd(E_total)
tableaddvar(L,"L","")

for L=Lmin ; L<=Lmax; L+=Lstep {
    setcellsize(L/N,L/N,L/N)
    minimize()
    tablesave()
}
```

Setting the initial state

```
m = vortex(1, -1)
```

To study the flower state, replace this with a uniform initial state with $\vec{m}$ along the easy axis (but slightly canted to break symmetry)

```
m = uniform(1.0, 0, 0.01)
```

GHENT UNIVERSITY

STANDARD PROBLEM 3

```
N := 64
setgridsize(N,N,N)

Lmin := 8.0
Lmax := 9.0
Lstep := 0.05

L := Lmin
setcellsize(L/N,L/N,L/N)

Msat = sqrt(2/mu0)
Aex = 1.0
Ku1 = 0.1
anisU = vector(1,0,0)

m = vortex(1,-1)

tableadd(E_total)
tableaddvar(L,"L","")

for L=Lmin ; L<=Lmax; L+=Lstep {
    setcellsize(L/N,L/N,L/N)
    minimize()
    tablesave()
}
```

Scheduling the output

```
tableadd(E_total)
tableaddvar(L,"L","")
```

GHENT UNIVERSITY

STANDARD PROBLEM 3

```
N := 64
setgridsize(N,N,N)

Lmin := 8.0
Lmax := 9.0
Lstep := 0.05

L := Lmin
setcellsize(L/N,L/N,L/N)

Msat = sqrt(2/mu0)
Aex = 1.0
Ku1 = 0.1
anisU = vector(1,0,0)

m = vortex(1,-1)

tableadd(E_total)
tableaddvar(L,"L","")

for L=Lmin ; L<=Lmax; L+=Lstep {
    setcellsize(L/N,L/N,L/N)
    minimize()
    tablesave()
}
```

Sweeping the cube's size

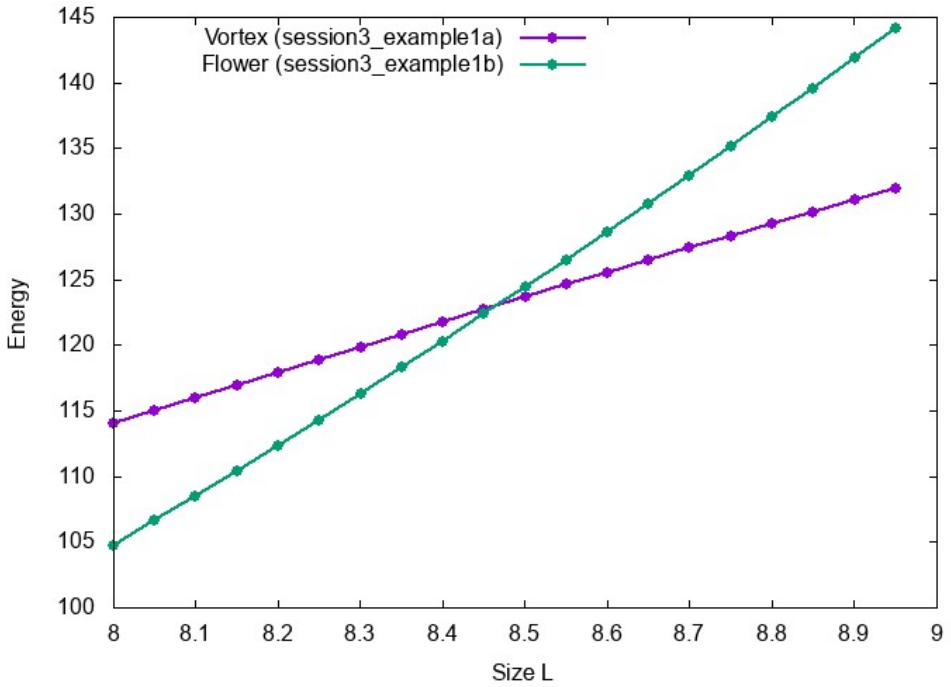
```
for L=Lmin ; L<=Lmax; L+=Lstep {
    setcellsize(L/N,L/N,L/N)
    minimize()
    tablesave()
}
```

GHENT UNIVERSITY

STANDARD PROBLEM 3

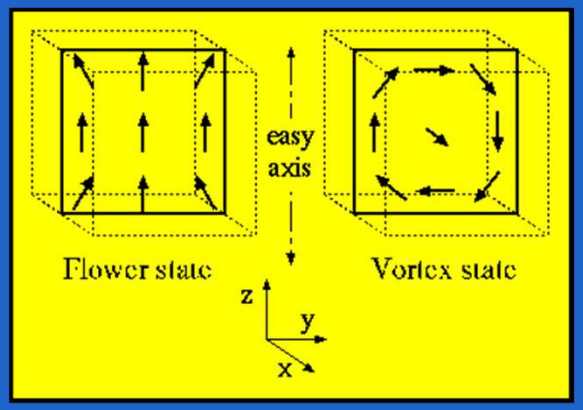
table.txt

# t (s)	mx ()	my ()	mz ()	E_total (J)	L ()
0	-3.73E-09	-8.73E-10	-0.41020608	113.99219	8
0	-4.47E-08	-5.47E-09	-0.40383163	114.98763	8.05
0	-1.49E-08	-7.39E-09	-0.39705437	115.976906	8.1
0	-2.24E-08	-1.06E-08	-0.39031976	116.96014	8.15
0	-4.47E-08	-8.56E-09	-0.3839808	117.93747	8.2
0	-2.98E-08	-9.20E-09	-0.377673	118.90891	8.25
0	-1.30E-08	-1.14E-08	-0.37151143	119.87465	8.3
0	-1.49E-08	-1.09E-08	-0.36549017	120.834694	8.35
0	-3.35E-08	-4.19E-09	-0.35949343	121.78918	8.4
0	-2.98E-08	-6.29E-09	-0.3538565	122.73815	8.45
0	-2.61E-08	-4.37E-09	-0.34811968	123.68168	8.5
0	-2.42E-08	-3.38E-09	-0.3427369	124.619865	8.55
0	-7.45E-09	-3.55E-09	-0.33735725	125.55271	8.6
0	-3.73E-08	-4.66E-09	-0.33208716	126.48035	8.65
0	-2.61E-08	-4.07E-09	-0.32683557	127.40276	8.7
0	-7.45E-09	-5.24E-09	-0.32190114	128.32005	8.75
0	-2.42E-08	-3.61E-09	-0.31686217	129.23222	8.8
0	-3.73E-08	-5.88E-09	-0.3121314	130.1394	8.85
0	-3.73E-08	-4.07E-09	-0.307299	131.04153	8.9
0	-7.45E-09	-7.86E-09	-0.3026616	131.93867	8.95



GHENT UNIVERSITY

STANDARD PROBLEM 3



What have we learned?

- How to use for loops in mumax3 scripts
- How to study phase diagrams
- How to work with unitless parameters

GHENT UNIVERSITY

SKYRMION RACETRACK

Skyrmion driven by spin transfer torques (STT) in a chiral ferromagnetic strip [1]

System: Co/Pt strip

- Ferromagnetic
- Perpendicular Magnetic Anisotropy (PMA)
- Interfacially-induced Dzyaloshinskii-Moriya interaction (iDMI)
- Uniform spin-polarized current along the length of the strip

Input file: session3_example2.txt
Titan RTX: 139s

GHENT UNIVERSITY

[1] **Nucleation, stability and current-induced motion of isolated magnetic skyrmions in nanostructures**
J. Sampaio, V. Cros, S. Rohart, A. Thiaville, A. Fert
Nature Nanotechnology **volume 8**, pages 839–844 (2013)

SKYRMION RACETRACK

```
// Skyrmion racetrack

setgridsize(256,64,1)
setcellsize(1e-9,1e-9,1e-9)
setpbc(4,0,0)

Msat = 580e3
Aex = 15e-12
Dind = 3.0e-3
Ku1 = 0.8e6
AnisU = vector(0,0,1)
alpha = 0.1

m = neelskyrmion(-1, 1).transl(-40e-9,0,0)
minimize()

Pol = 0.4
xi = 0.2
j = vector(-1e12,0,0) // A/m2

autosave(m,4e-10)
tableAutosave(1e-11)
tableAdd(ext_bubblepos)

run(10e-9)
```

Setting the mesh

```
setgridsize(256,64,1)
setcellsize(1e-9,1e-9,1e-9)
setpbc(4,0,0)
```

Note on the cell size:

In chiral ferromagnets, the magnetization might vary on a much smaller length scale than the exchange length!

When choosing the cell size make sure that maxAngle is small enough during the simulation (let's say <0.35rad)

In this example `print(maxAngle)` returns 0.26rad → cell size OK

GHENT UNIVERSITY

SKYRMION RACETRACK

```
// Skyrmion racetrack

setgridsize(256,64,1)
setcellsize(1e-9,1e-9,1e-9)
setpbc(4,0,0)

Msat = 580e3
Aex = 15e-12
Dind = 3.0e-3
Ku1 = 0.8e6
AnisU = vector(0,0,1)
alpha = 0.1

m = neelskyrmion(-1, 1).transl(-40e-9,0,0)
minimize()

Pol = 0.4
xi = 0.2
j = vector(-1e12,0,0) // A/m2

autosave(m,4e-10)
tableAutosave(1e-11)
tableAdd(ext_bubblepos)

run(10e-9)
```

The material parameters[1]

```
Msat = 580e3
Aex = 15e-12
Dind = 3.0e-3
Ku1 = 0.8e6
AnisU = vector(0,0,1)
alpha = 0.1
```

GHENT UNIVERSITY

[1] **Nucleation, stability and current-induced motion of isolated magnetic skyrmions in nanostructures**
J. Sampaio, V. Cros, S. Rohart, A. Thiaville, A. Fert
Nature Nanotechnology **volume 8**, pages 839–844 (2013)

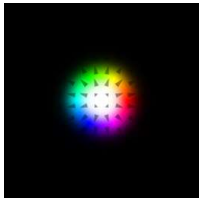
SKYRMION RACETRACK

```
// Skyrmion racetrack

setgridsize(256,64,1)
setcellsize(1e-9,1e-9,1e-9)
setpbc(4,0,0)

Msat = 580e3
Aex = 15e-12
Dind = 3.0e-3
Ku1 = 0.8e6
AnisU = vector(0,0,1)
alpha = 0.1

m = neelskyrmion(-1, 1).transl(-40e-9,0,0)
minimize()

Pol = 0.4
xi = 0.2
j = vector(-1e12,0,0) // A/m2

autosave(m,4e-10)
tableAutosave(1e-11)
tableAdd(ext_bubblepos)

run(10e-9)
```

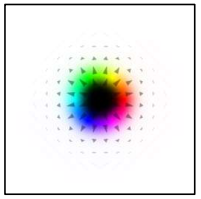
The initial state:

```
m = neelskyrmion(-1, 1).transl(-40e-9,0,0)
minimize()
```

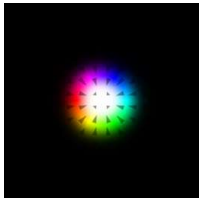
Neelskyrmion(1,1)



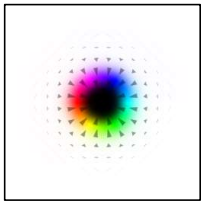
Neelskyrmion(1,-1)



Neelskyrmion(-1,1)



Neelskyrmion(-1,-1)



GHENT UNIVERSITY

SKYRMION RACETRACK

```
// Skyrmion racetrack

setgridsize(256,64,1)
setcellsize(1e-9,1e-9,1e-9)
setpbc(4,0,0)

Msat = 580e3
Aex = 15e-12
Dind = 3.0e-3
Ku1 = 0.8e6
AnisU = vector(0,0,1)
alpha = 0.1

m = neelskyrmion(-1, 1).transl(-40e-9,0,0)
minimize()

Pol = 0.4
xi = 0.2
j = vector(-1e12,0,0)

autosave(m,4e-10)
tableAutosave(1e-11)
tableAdd(ext_bubblepos)

run(10e-9)
```

Applying a spin polarized current:

```
Pol = 0.4
xi = 0.2
j = vector(-1e12,0,0)
```

GHENT UNIVERSITY

SKYRMION RACETRACK

```
// Skyrmion racetrack

setgridsize(256,64,1)
setcellsize(1e-9,1e-9,1e-9)
setpbc(4,0,0)

Msat = 580e3
Aex = 15e-12
Dind = 3.0e-3
Ku1 = 0.8e6
AnisU = vector(0,0,1)
alpha = 0.1

m = neelskyrmion(-1, 1).transl(-40e-9,0,0)
minimize()

Pol = 0.4
xi = 0.2
j = vector(-1e12,0,0)

autosave(m,4e-10)
tableAutosave(1e-11)
tableAdd(ext_bubblepos)

run(10e-9)
```

Output scheduling:

```
autosave(m,4e-10)
tableAutosave(1e-11)
tableAdd(ext_bubblepos)
```

GHENT UNIVERSITY

SKYRMION RACETRACK

```
// Skyrmion racetrack

setgridsize(256,64,1)
setcellsize(1e-9,1e-9,1e-9)
setpbc(4,0,0)

Msat = 580e3
Aex = 15e-12
Dind = 3.0e-3
Ku1 = 0.8e6
AnisU = vector(0,0,1)
alpha = 0.1

m = neelskyrmion(-1, 1).transl(-40e-9,0,0)
minimize()

Pol = 0.4
xi = 0.2
j = vector(-1e12,0,0)

autosave(m,4e-10)
tableAutosave(1e-11)
tableAdd(ext_bubblepos)

run(10e-9)
```

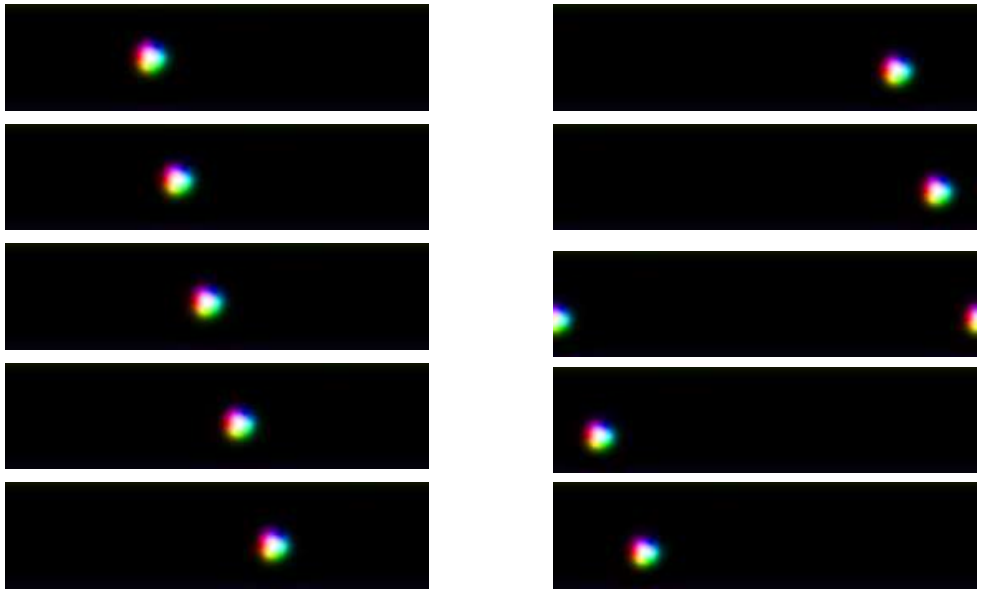
Solving the LLG equation

```
Run(10e-9)
```

GHENT UNIVERSITY

SKYRMION RACETRACK

```
mumax3-convert -png skyrmionracetrack.out/*.ovf
```

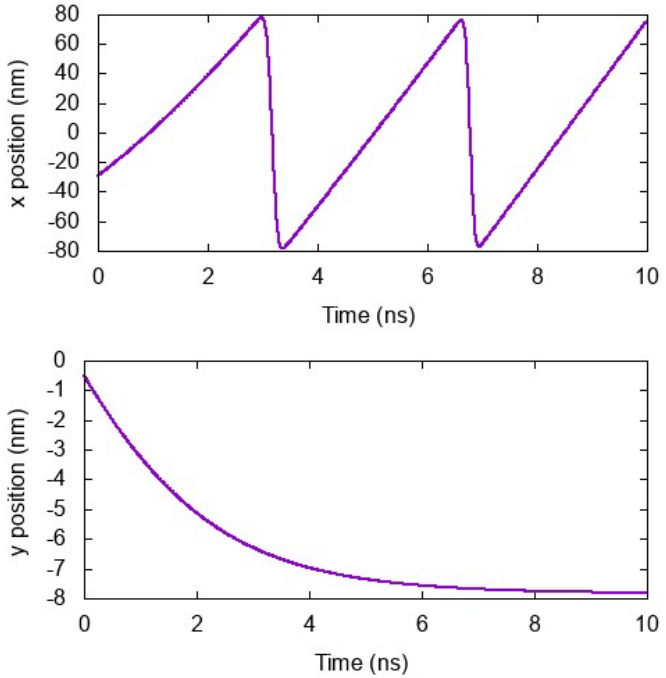


GHENT UNIVERSITY

SKYRMION RACETRACK

table.txt

# t (s)	mx ()	my ()	mz ()	ext_bubbleposx (m)	ext_bubbleposy (m)	ext_bubbleposz (m)
0	1.46E-11	1.86E-09	-0.96588	-2.90E-08	-4.99E-10	0
1.00E-11	-0.00011191	-2.78E-05	-0.96588	-2.88E-08	-5.31E-10	0
2.00E-11	-1.95E-05	5.34E-06	-0.96587	-2.85E-08	-5.61E-10	0
3.00E-11	-9.36E-05	-1.93E-05	-0.96587	-2.82E-08	-5.93E-10	0
4.00E-11	-3.82E-05	-7.81E-06	-0.96587	-2.79E-08	-6.25E-10	0
5.00E-11	-7.69E-05	-1.55E-05	-0.96587	-2.76E-08	-6.56E-10	0
6.00E-11	-4.99E-05	-1.33E-05	-0.96587	-2.74E-08	-6.88E-10	0
7.00E-11	-6.81E-05	-1.61E-05	-0.96588	-2.71E-08	-7.19E-10	0
8.00E-11	-5.55E-05	-1.68E-05	-0.96588	-2.68E-08	-7.50E-10	0
9.00E-11	-6.35E-05	-1.79E-05	-0.96588	-2.65E-08	-7.82E-10	0
1.00E-10	-5.81E-05	-1.91E-05	-0.96588	-2.62E-08	-8.13E-10	0
1.10E-10	-6.13E-05	-2.03E-05	-0.96588	-2.60E-08	-8.44E-10	0
1.20E-10	-5.89E-05	-2.13E-05	-0.96588	-2.57E-08	-8.74E-10	0
1.30E-10	-6.03E-05	-2.26E-05	-0.96588	-2.54E-08	-9.05E-10	0
1.40E-10	-5.90E-05	-2.36E-05	-0.96589	-2.51E-08	-9.36E-10	0
1.50E-10	-5.96E-05	-2.49E-05	-0.96589	-2.48E-08	-9.67E-10	0
1.60E-10	-5.89E-05	-2.59E-05	-0.96589	-2.45E-08	-9.97E-10	0
1.70E-10	-5.91E-05	-2.71E-05	-0.96589	-2.42E-08	-1.03E-09	0
1.80E-10	-5.87E-05	-2.82E-05	-0.96589	-2.39E-08	-1.06E-09	0
1.90E-10	-5.87E-05	-2.94E-05	-0.9659	-2.37E-08	-1.09E-09	0
2.00E-10	-5.84E-05	-3.05E-05	-0.9659	-2.34E-08	-1.12E-09	0



GHENT UNIVERSITY

SKYRMION RACETRACK

What have we learned?

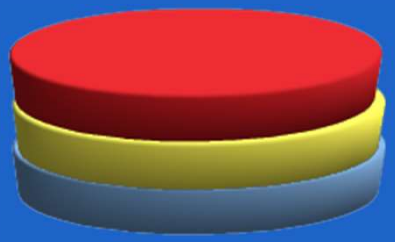
- How to simulate spin transfer torques (Zhang-Li model)
- How to simulate chiral ferromagnets (iDMI)

TABLE OF CONTENTS

GHENT UNIVERSITY

SLONCZEWSKI SPIN TRANSFER TORQUE

Fixed layer
Spacer layer
Free layer



Input file: session3_example3.txt

Titan RTX: 7 s

System

ellipse-shaped spin-torque MRAM stack

Task

Switching the MRAM bit using a spin polarized current

GHENT UNIVERSITY

https://mumax.github.io/examples.html#example14

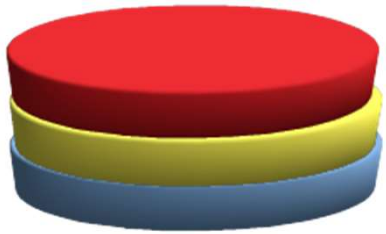
SLONCZEWSKI STT

```
// geometry
sizeX := 160e-9
sizeY := 80e-9
sizeZ := 5e-9

Nx := 64
Ny := 32
Nz := 1

setgridsize(Nx, Ny, Nz)
setcellsize(sizeX/Nx, sizeY/Ny, sizeZ/Nz)
setGeom(ellipse(sizeX, sizeY))
```

```
// set up free layer
Msat  = 800e3
Aex   = 13e-12
alpha = 0.01
m     = uniform(1, 0, 0)
```

Ellipse-shaped spin-torque MRAM stack consisting of a fixed layer, spacer and free layer.
Only the free layer magnetization is explicitly modeled, so we use a 2D grid.

Fixed layer
Spacer layer
Free layer



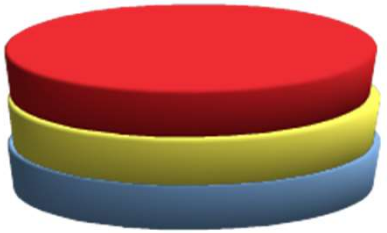
GHENT UNIVERSITY

SLONCZEWSKI STT

```
// set up fixed and spacer layer parameters
lambda       = 1
Pol          = 0.5669
epsilonprime = 0

angle := 20 * pi/180
px := cos(angle)
py := sin(angle)
fixedlayer = vector(-px, -py, 0)
```

```
// send current
Jtot := 0.008            // total current in A
area := sizeX*sizeY*pi/4
jc   := Jtot / area       // current density in A/m2
J = vector(0, 0, jc)
```

Fixed layer
Spacer layer
Free layer



J is the current density, along the z-axis in this case.

SLONCZEWSKI STT

Slonczewski spin-transfer torque

$$\tau = \beta \frac{\epsilon - \alpha\epsilon'}{1+\alpha^2} \boldsymbol{m} \times (\boldsymbol{m}_P \times \mathbf{m}) - \beta \frac{\epsilon' - \alpha\epsilon}{1+\alpha^2} \boldsymbol{m} \times \boldsymbol{m}_P$$

with

$$\beta = \frac{j_z \hbar}{M_s e\, d}$$

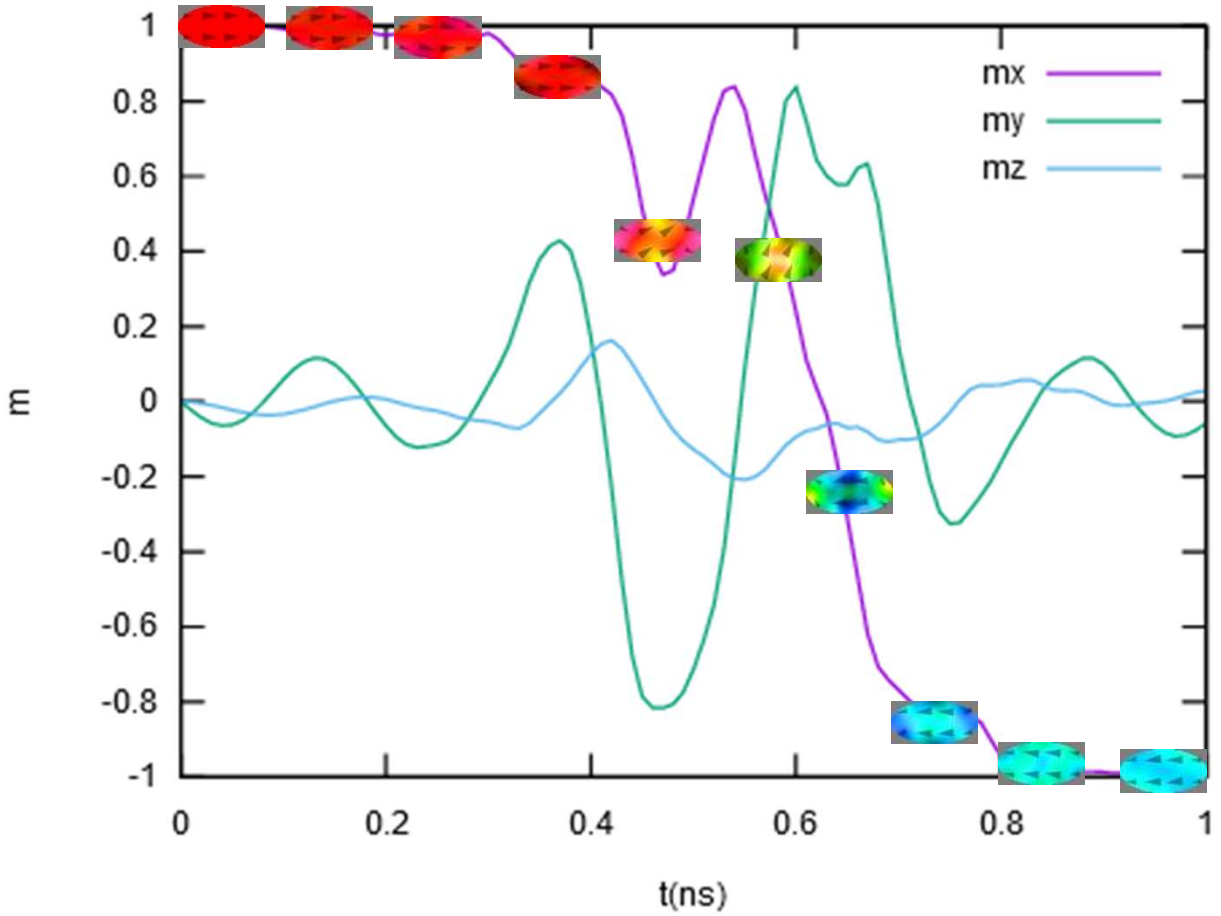
$$\epsilon = \frac{P\Lambda^2}{(\Lambda^2+1) + (\Lambda^2-1)(\boldsymbol{m}\cdot\boldsymbol{m}_P)}$$

Regional Material Parameters	
Pol	Electrical current polarization
Lambda	Slonczewski Λ parameter
EpsilonPrime	Slonczewski secondairy STT term ε'
alpha	Damping parameter
Msat	Saturation magnetization (A/m)
FreeLayerThickness	Slonczewski free layer thickness (if set to zero (default), then the thickness will be deduced from the mesh size) (m)
Excitation	
J	Electrical current density (A/m2)
FixedLayer	Slonczewski fixed layer polarization
Output Quantities	
STTorque	Spin-transfer torque/γ0 (T)
Other functionalities	
DisableSlonczewskiTorque	Disables Slonczewski torque (default=false)
FixedLayerPosition	Position of the fixed layer: FIXEDLAYER_TOP, FIXEDLAYER_BOTTOM (default=FIXEDLAYER_TOP)

GHENT UNIVERSITY

SLONCZEWSKI STT

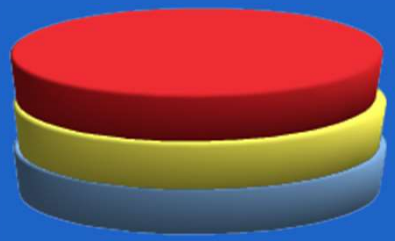
```
// schedule output & run
autosave(m, 100e-12)
autosnapshot(m, 100e-12)
tableautosave(10e-12)
run(1e-9)
```



GHENT UNIVERSITY

SLONCZEWSKI SPIN TRANSFER TORQUE

Fixed layer
Spacer layer
Free layer



WHAT HAVE WE LEARNED?

- How to use Skonczewski spin transfer torques
- Mumax3 only explicitly models the free layer

GHENT UNIVERSITY

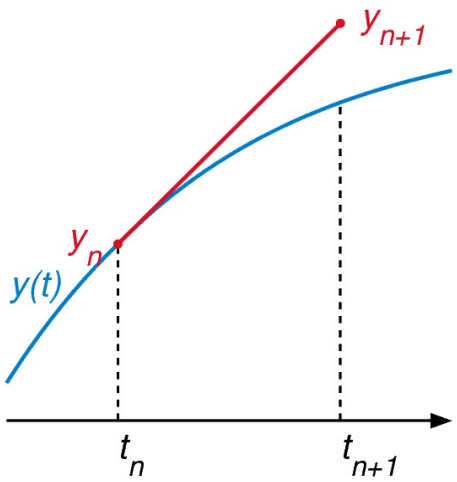
INTERMEZZO

UNDER THE HOOD: SOLVERS

GHENT UNIVERSITY

HOW DOES A SOLVER WORK?

Simple example: Euler method



$$\frac{dy}{dt} = f(t, y)$$

LLG equation

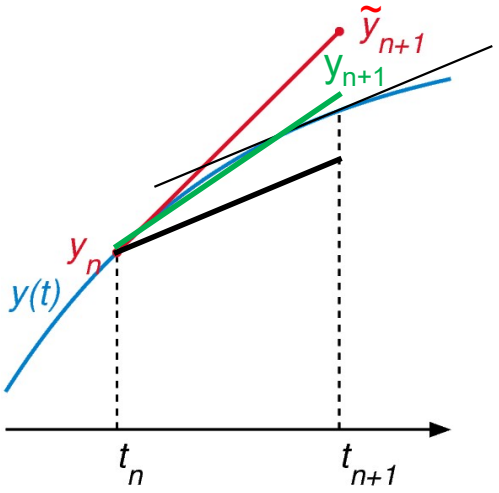
$$\dot{\boldsymbol{m}} = -\frac{\gamma}{1+\alpha^2}\left[\boldsymbol{m} \times \boldsymbol{H}_{eff} + \alpha\, \boldsymbol{m} \times (\boldsymbol{m} \times \boldsymbol{H}_{eff})\right]$$

$$y_{n+1} = y_n + hf(t, y_n)$$

GHENT UNIVERSITY

HOW DOES A SOLVER WORK?

Second example: Heun method



$$\frac{dy}{dt} = f(t, y)$$

$$\tilde{y}_{n+1} = y_n + hf(t, y_n)$$

$$y_{n+1} = y_n + \frac{h}{2}\left(f(t, y_n) + f(t + h, \tilde{y}_{n+1})\right)$$

GHENT UNIVERSITY

Figure adapted from wikimedia commons, by kelam, CC-BY-SA-4.0 license

RUNGE-KUTTA METHODS

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i,$$

$$\begin{aligned}
k_1 &= f(t_n, y_n),\\
k_2 &= f(t_n + c_2 h, y_n + h(a_{21} k_1)),\\
k_3 &= f(t_n + c_3 h, y_n + h(a_{31} k_1 + a_{32} k_2)),\\
&\vdots\\
k_s &= f(t_n + c_s h, y_n + h(a_{s1} k_1 + a_{s2} k_2 + \cdots + a_{s,s-1} k_{s-1})).
\end{aligned}$$

$$\begin{array}{c|cccc}
0 & & & & \\
c_2 & a_{21} & & & \\
c_3 & a_{31} & a_{32} & & \\
\vdots & \vdots & & \ddots & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} \quad b_s
\end{array}$$

GHENT UNIVERSITY

https://en.wikipedia.org/wiki/Runge-Kutta_methods

NOTATION: BUTCHER TABLEAU

Euler method

First order convergence

0	
	1

$$y_n$$

$$y_{n+1} = y_n + hf(t, y_n)$$

Heun method

Second order convergence

0		
1	1	
	$\frac{1}{2}$	$\frac{1}{2}$

$$y_n$$

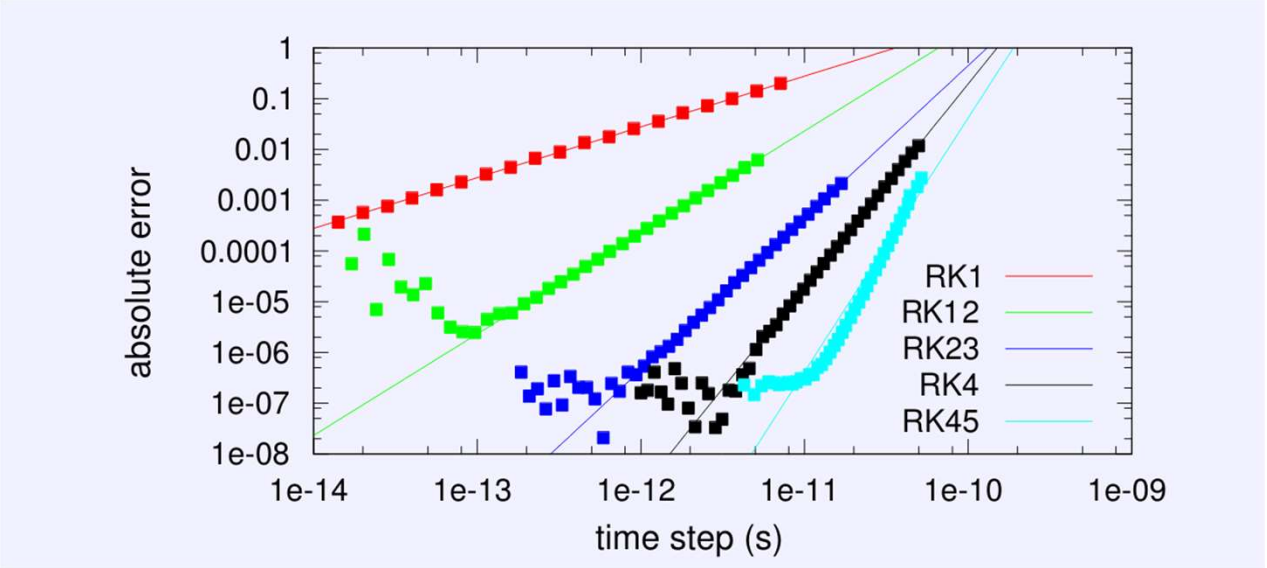
$$\tilde{y}_{n+1} = y_n + hf(t, y_n)$$

$$y_{n+1} = y_n + \frac{h}{2}\left(f(t, y_n) + f(t + h, \tilde{y}_{n+1})\right)$$

GHENT UNIVERSITY

https://en.wikipedia.org/wiki/Runge-Kutta_methods

ERROR CONVERGENCE



Absolute error on a single spin after precessing without damping for one period in a 100 mT field, as function of the different solvers' time steps. The errors follow 1st, 2nd, 3rd, 4th or 5th order convergence for the respective solvers down to a limit set by the single precision arithmetic.

GHENT UNIVERSITY

ADAPTIVE TIME STEPPING

Euler method

First order convergence

0	
	1

$$y_n$$

$$y_{n+1} = y_n + hf(t, y_n)$$

Heun method

Second order convergence

0		
1	1	
	$\frac{1}{2}$	$\frac{1}{2}$
	1	0

$$y_n$$

$$\tilde{y}_{n+1} = y_n + hf(t, y_n)$$

$$y_{n+1} = y_n + \frac{h}{2}\left(f(t, y_n) + f(t + h, \tilde{y}_{n+1})\right)$$

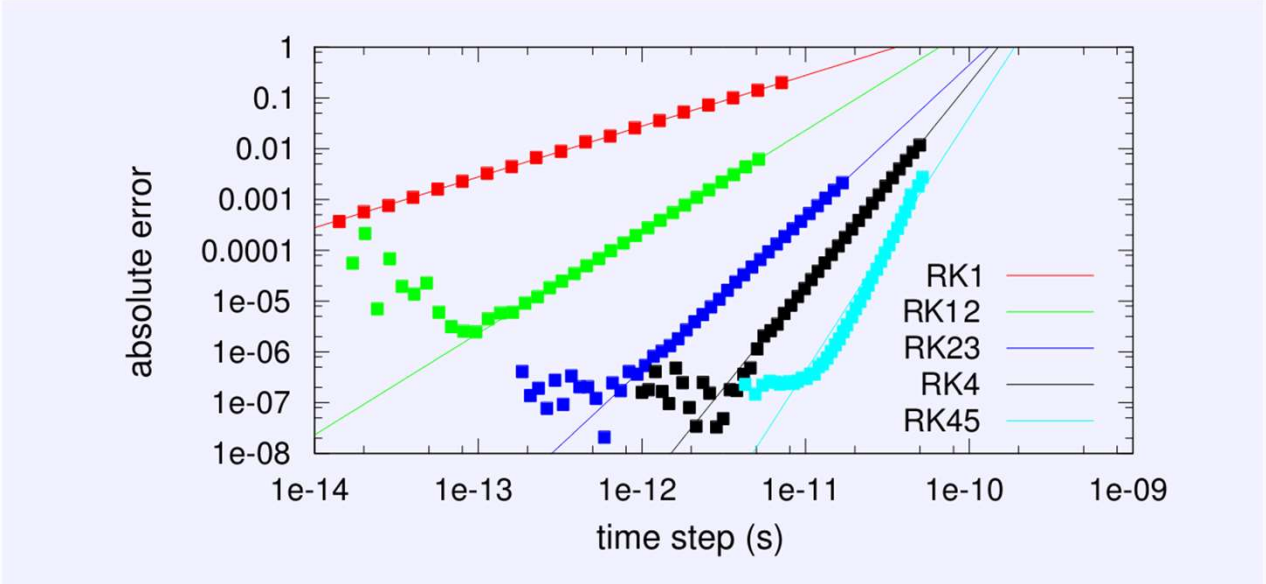
$$y_{n+1} = y_n + hf(t, y_n)$$

Difference between both solutions is measure for the error!

GHENT UNIVERSITY

https://en.wikipedia.org/wiki/Runge-Kutta_methods

PERFORMANCE



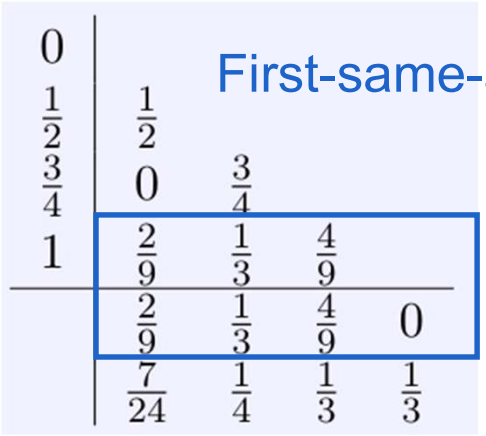
solver	Euler	Heun	RK3	RK4	Dormand-Prince	Fehlberg56	Fehlberg67
$\frac{\#\text{evaluations}}{\text{step}}$	1	2	3	4	6	8	13

GHENT UNIVERSITY

IMPORTANT MUMAX SOLVERS

Bogacki-Shampine solver

```
Setsolver(3)
//used during relax()
//Third order convergence with
//embedded second order method
```



$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{3}{4} & 0 & \frac{3}{4} & & \\
1 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & \\
\hline
 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & 0 \\
 & \frac{7}{24} & \frac{1}{4} & \frac{1}{3} & \frac{1}{3}
\end{array}
$$

GHENT UNIVERSITY

IMPORTANT MUMAX SOLVERS

Bogacki-Shampine solver

```
Setsolver(3)
//used during relax()
//Third order convergence with
//embedded second order method
```

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{3}{4} & 0 & \frac{3}{4} & & \\
1 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & \\
\hline
 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & 0 \\
 & \frac{7}{24} & \frac{1}{4} & \frac{1}{3} & \frac{1}{3}
\end{array}
$$

First-same-as-last (FSAL)

Dormand-Prince solver

```
Setsolver(5)
//default for dynamics
//Fifth order convergence with
//embedded fourth order method
```

$$
\begin{array}{c|ccccccc}
0 & & & & & & & \\
\frac{1}{5} & \frac{1}{5} & & & & & & \\
\frac{3}{10} & \frac{3}{40} & \frac{9}{40} & & & & & \\
\frac{4}{5} & \frac{44}{45} & -\frac{56}{15} & \frac{32}{9} & & & & \\
\frac{8}{9} & \frac{19372}{6561} & -\frac{25360}{2187} & \frac{64448}{6561} & -\frac{212}{729} & & & \\
1 & \frac{9017}{3168} & -\frac{355}{33} & \frac{46732}{5247} & \frac{49}{176} & -\frac{5103}{18656} & & \\
1 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & \\
\hline
 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & 0 \\
 & \frac{5179}{57600} & 0 & \frac{7571}{16695} & \frac{393}{640} & -\frac{9209}{339200} & \frac{187}{2100} & \frac{1}{40}
\end{array}
$$

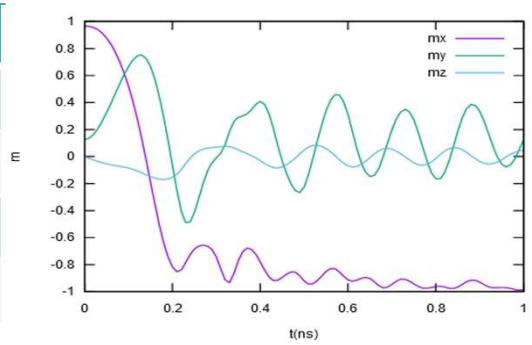
solver	Euler	Heun	RK3	RK4	Dormand-Prince	Fehl-berg56	Fehl-berg67
#evaluations / step	1	2	3 =4-1	4	6 =7-1	8	13

GHENT UNIVERSITY

WHICH ONE IS BEST?

```
//Standard problem 4 with tiny cellsize
SetGridsize(512, 128, 1)
SetCellsize(500e-9/512, 125e-9/128, 3e-9)
Msat  = 800e3
Aex   = 13e-12
alpha = 0.02

m = uniform(1, .1, 0)
relax()

B_ext = vector(-24.6E-3, 4.3E-3, 0)
run(1e-9)
```

solver	#steps	Time (s)
2	120740	83.4
3	**62148**	**39.27**
5	59912	69.61
6	54123	78.17

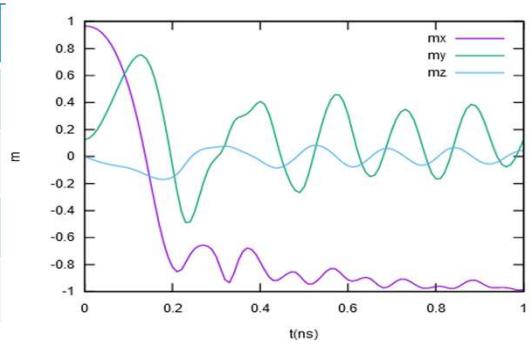


GHENT UNIVERSITY

WHICH ONE IS BEST?

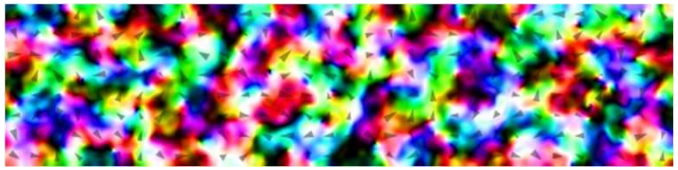
```
//Standard problem 4 with tiny cellsize
SetGridsize(512, 128, 1)
SetCellsize(500e-9/512, 125e-9/128, 3e-9)
Msat  = 800e3
Aex   = 13e-12
alpha = 0.02

m = uniform(1, .1, 0)
relax()

B_ext = vector(-24.6E-3, 4.3E-3, 0)
run(1e-9)
```

solver	#steps	Time (s)
2	120740	83.4
3	**62148**	**39.27**
5	59912	69.61
6	54123	78.17




```
SetGridsize(512, 128, 1)
SetCellsize(500e-9/512, 125e-9/128, 3e-9)
Msat  = 800e3
Aex   = 13e-12
alpha = 0.02

m = randommag()


B_ext = vector(-24.6E-3, 4.3E-3, 0)
run(2e-10)
```

solver	#steps	Time (s)
2	460177	451
3	26573	42.26
5	**7785**	**24.58**
6	7004	32



GHENT UNIVERSITY

IT DEPENDS,
BUT THE DEFAULTS ARE GOOD

UNDER THE HOOD: SOLVERS

WHAT HAVE WE LEARNED?

- Basics of Runge-Kutta methods
- How does adaptive time stepping work
- Default mumax3 solvers offer good performance

THERMAL GRADIENT DRIVEN DOMAIN WALL

System

Transverse domain wall in permalloy nanostrip

Task

Calculate the motion of the domain wall driven by a thermal gradient over the nanostrip

Input file: session3_example4.txt

Titan RTX: 28 min

Magnetization

Thermal field

GHENT UNIVERSITY

THERMAL GRADIENT DRIVEN DOMAIN WALL

```
SetMesh(256, 16, 1, 5e-09, 5e-09, 20e-09, 0, 0, 0)
Msat = 800e3
Aex = 1e-11
alpha = 0.1
```

```
SetGridsize(256, 16, 1)
SetCellsize(5e-9, 5e-9, 20e-9)
SetPBC(0,0,0)
```

```
//define regions
for i :=1;i<198;i++{
    defregion(i,xrange(-640e-9+(i+1)*6.4e-9,-640e-9+(i+2)*6.4e-9))
}
```


```
//define thermal gradient and set thermal seed
for i :=1;i<200;i++{
    temp.setregion(i,200+i)
}
Thermseed(12345)
```

0K| 200K ----------> 400K |0K

Slide 46

JM1 [@Jonathan Leliaert] eerste keer dat setmesh expliciet gebruikt wordt in de workshop. Dus je zal tijdens het presenteren hier wel een woordje uitleg bij moeten geven
Jeroen Mulkers, 9/6/2020

JL1 doe ik
Jonathan Leliaert, 9/6/2020

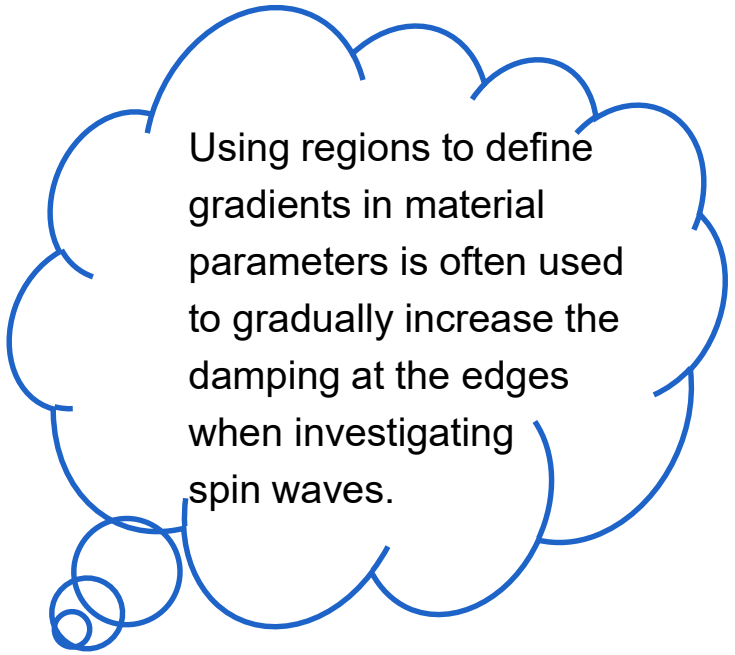
THERMAL GRADIENT DRIVEN DOMAIN WALL

```
SetMesh(256, 16, 1, 5e-09, 5.e-09, 20e-09, 0, 0, 0)
Msat = 800e3
Aex = 1e-11
alpha = 0.1
```

```
//define regions
for i :=1;i<198;i++{
    defregion(i,xrange(-640e-9+(i+1)*6.4e-9,-640e-9+(i+2)*6.4e-9))
}
```


```
//define thermal gradient and set thermal seed
for i :=1;i<200;i++{
    temp.setregion(i,200+i)
}
Thermseed(12345)
```

0K| 200K ----------> 400K |0K



THERMAL GRADIENT DRIVEN DOMAIN WALL

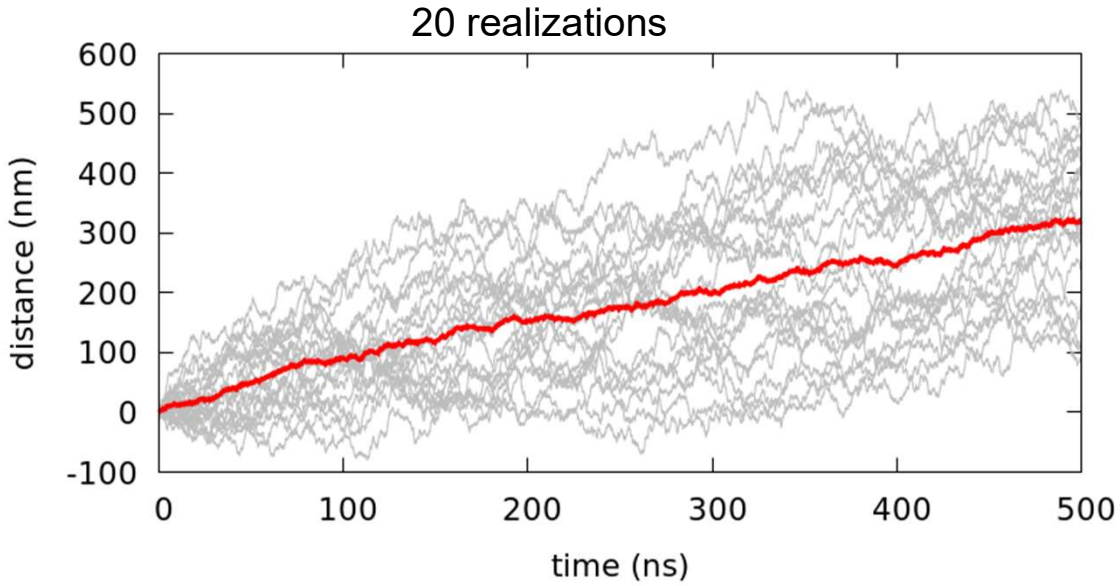
```
m=twodomain(1,0,0,  0,1,0,  -1,0,0)
relax()
```

```
//remove edge charges
BoundaryRegion := 0
MagLeft        := 1
MagRight       := -1
ext_rmSurfaceCharge(BoundaryRegion, MagLeft, MagRight)
relax()
```

GHENT UNIVERSITY

THERMAL GRADIENT DRIVEN DOMAIN WALL

```
//schedule output
tableadd(ext_dwxpos)
tableautosave(1e-10)

//set solver with adaptive timestepping
setsolver(5)
fixdt=0

run(5e-7)
```

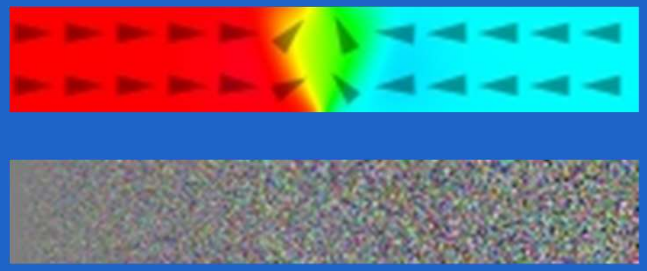


Adaptively time stepping the stochastic Landau-Lifshitz-Gilbert equation at nonzero temperature: Implementation and validation in MuMax³

AIP Advances **7**, 125010 (2017)

GHENT UNIVERSITY

THERMAL GRADIENT DRIVEN DOMAIN WALL



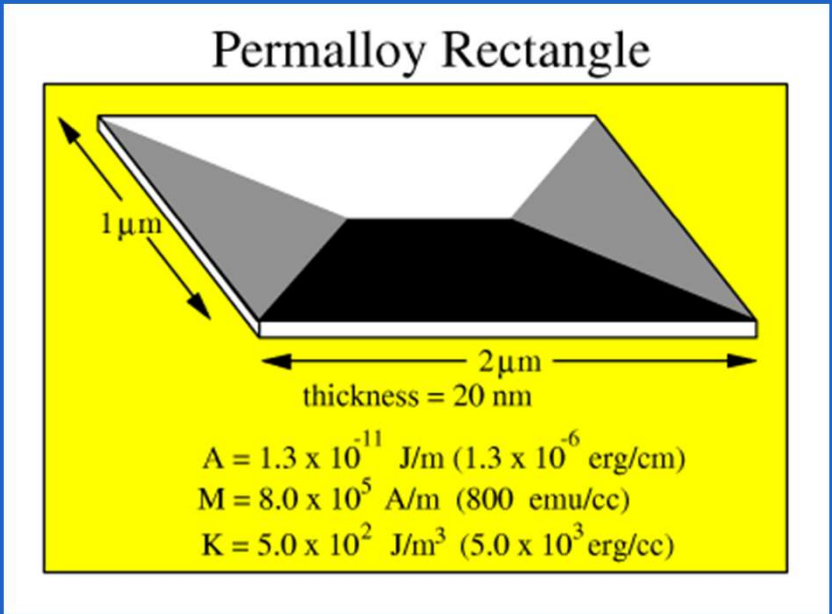
WHAT HAVE WE LEARNED?

- `SetMesh(...)` command
- How to construct gradients in parameters
- How to remove edge charges
- Adaptive time stepping at nonzero temperature

HYSTERESIS: STANDARD PROBLEM 1

Input file: Session3_example5a.txt

Titan RTX: 46 min



System

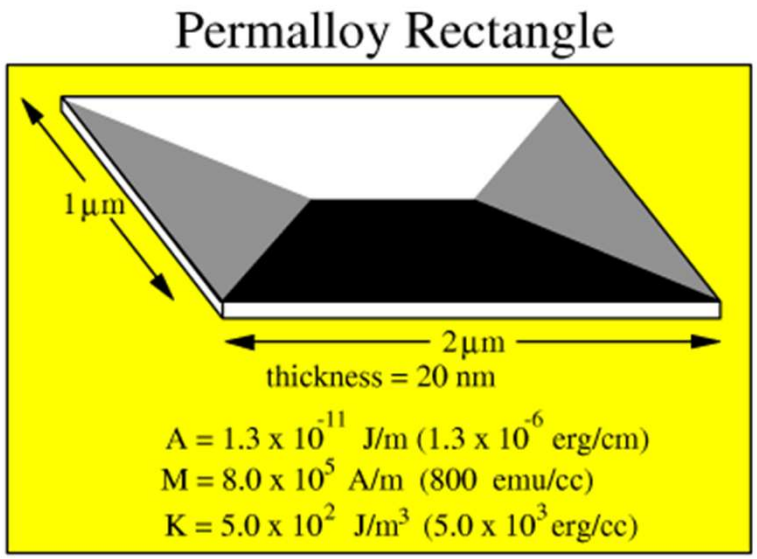
Permalloy rectangle

Task

Calculate the hysteresis loop for an external field parallel to the long axis.

https://www.ctcms.nist.gov/~rdm/mumag.org.html

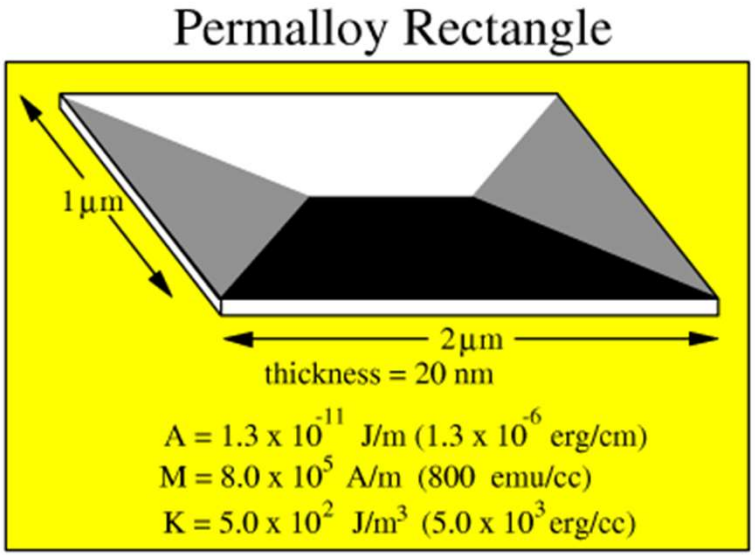
HYSTERESIS:STANDARD PROBLEM 1




```
SetGridsize(192,384, 1)
SetCellsize(1000e-9/192,2000e-9/384,  20e-9)

Msat  = 800e3
Aex   = 13e-12
alpha = 0.02
Ku1=5e2
anisu=vector(0,1,0)
m=uniform(-0.1,-1,0)
```

GHENT UNIVERSITY

HYSTERESIS:STANDARD PROBLEM 1



Task

Calculate the hysteresis loop for an external field parallel to the long axis.

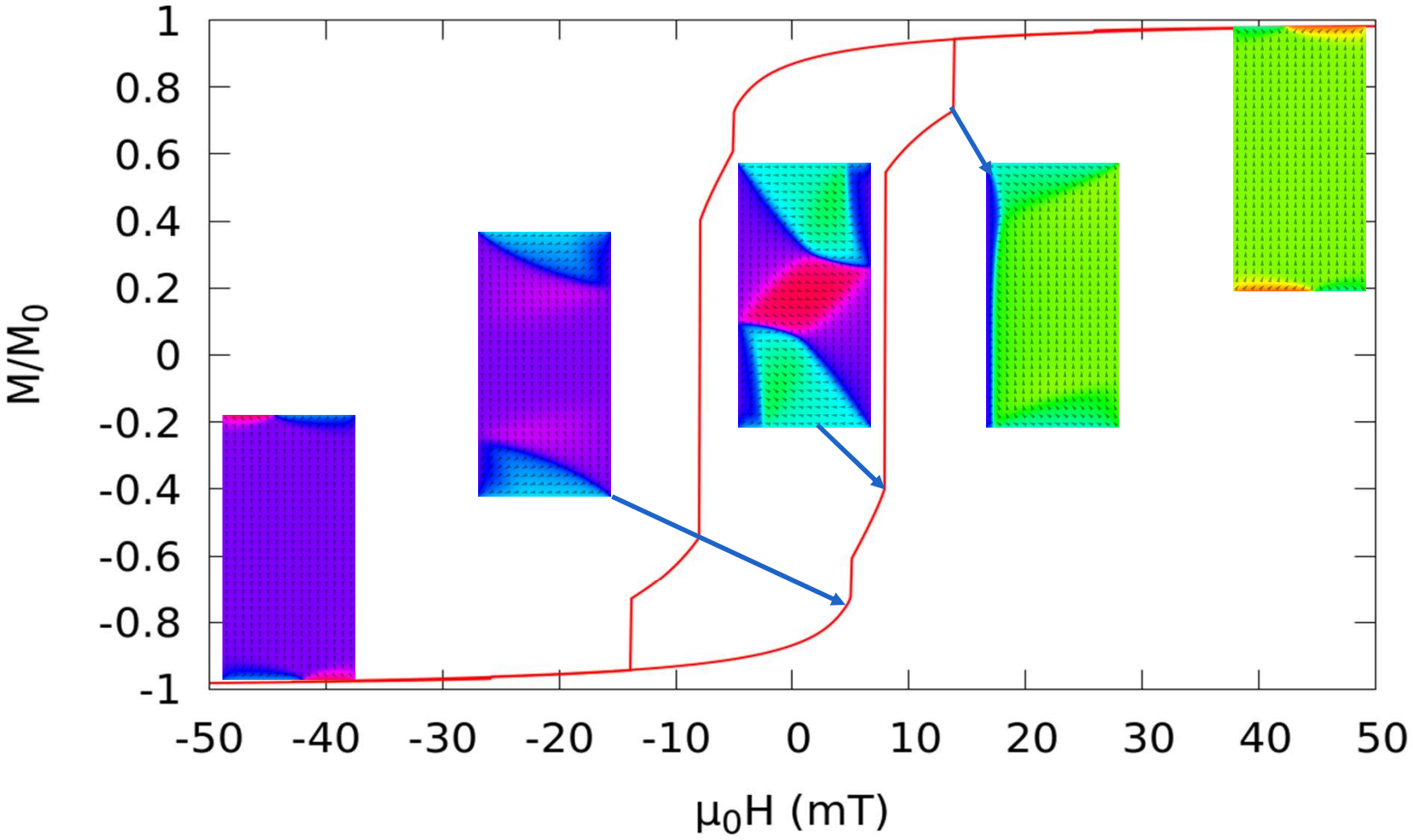
```
angle  :=pi/180   //1 degree
B_ext   = vector( -0.05*sin(angle) , -0.05*cos(angle) , 0)
tableadd(B_ext)
relax()

B_min  :=  -0.05
B_max  :=   0.05
B_step :=   0.0005
```

```
//ramp the field up
for B:=B_min;B<=B_max;B+=B_step{
    B_ext = vector(B*sin(angle),B*cos(angle),  0)
    minimize()
    tablesave()
    snapshot(m)
}
```

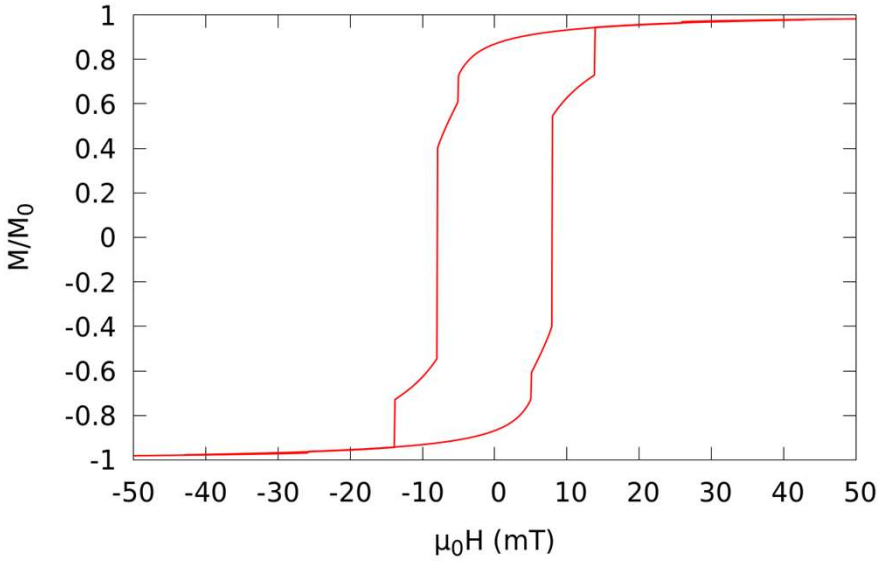
```
//ramp the field down
for B:=B_max;B>=B_min;B-=B_step{
    B_ext = vector( B*sin(angle),B*cos(angle), 0)
    minimize()
    tablesave()
    snapshot(m)
}
```

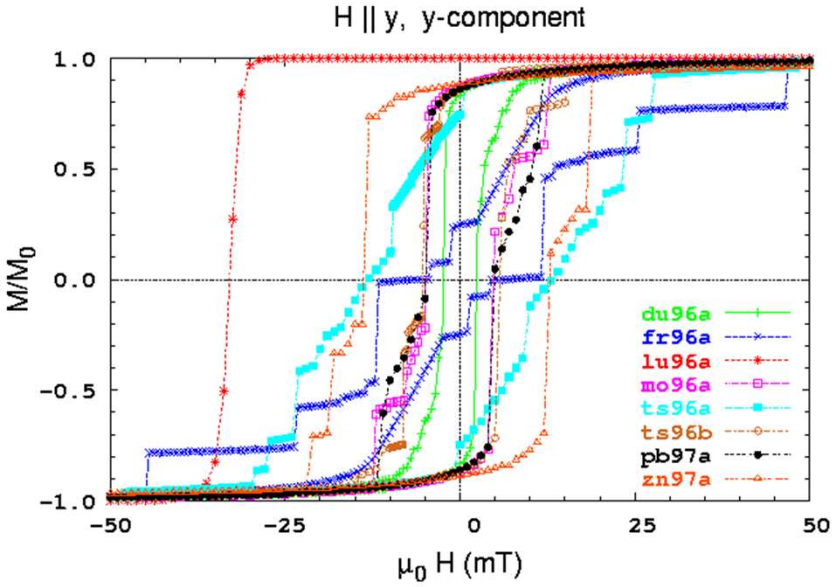
GHENT UNIVERSITY

HYSTERESIS: STANDARD PROBLEM 1



HYSTERESIS: STANDARD PROBLEM 1





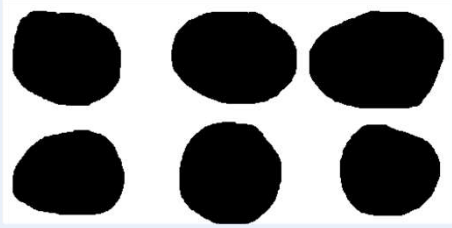
TAKE-HOME MESSAGE:

BE CAREFUL WHEN SIMULATING HYSTERESIS LOOPS!

GHENT UNIVERSITY

HYSTERESIS: COMPLEX GEOMETRY

```
//square thin-film mesh with 1600 nm side length
SetMesh(512, 512, 1, 3.125e-09, 3.125e-09, 20e-09, 0, 0, 0)

//permalloy like material parameters
Msat = 800e3
Aex = 1e-11
alpha = 0.01

//add drawing of real sample
real:= imageShape("drawing.png").scale(1, 0.5, 1.).transl(0, -400e-9, 0)

//define 6 perfect circles
perfect := circle(350e-9).repeat(600e-9, 400e-9, 0).transl(0, 200e-9, 0).intersect(yrange(0, inf))

//set the geometry
setgeom(perfect.add(real))
```

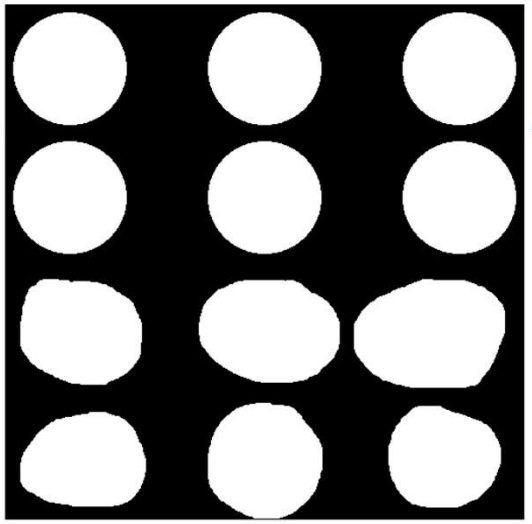


drawing.png

Input file: session3_example5b.txt

Titan RTX: 23 min

geometry

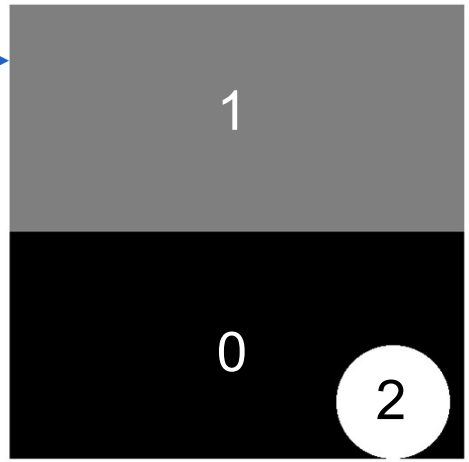


GHENT UNIVERSITY

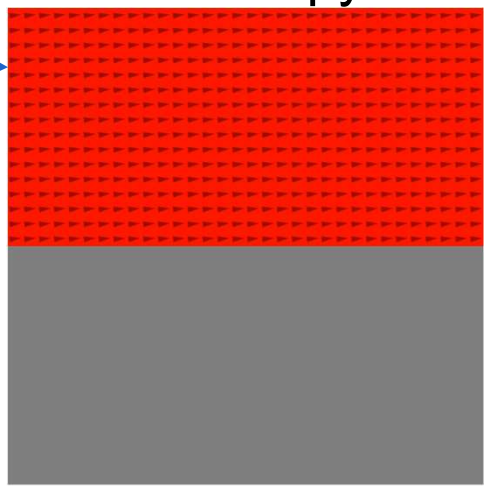
HYSTERESIS: COMPLEX GEOMETRY

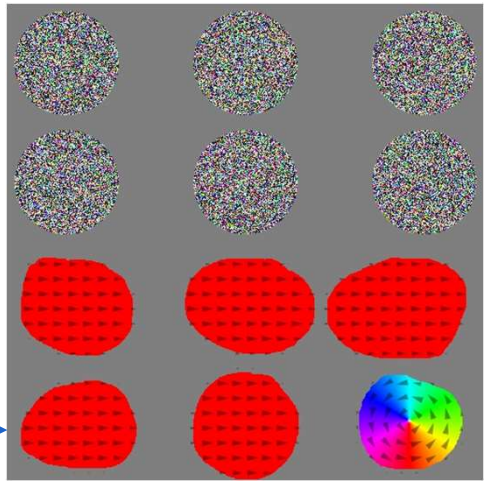
```
//define 2 regions, default region =0
defregion(1, yrange(0,inf))
defregion(2, circle(400e-9).transl(550e-9, -600e-9, 0))


//set material parameters in regions
Ku1.setregion(1,2e4)
anisu.setregion(1,vector(1,0.1,0))

Aex.setregion(2,2e-11)
msat.setregion(2,400e3)

//set initial magnetization in regions or shapes
//default = randommag
m.setinshape(real, uniform(1, 0, 0))
m.setregion(2, vortex(1, 1).transl(550e-9, -600e-9, 0))
```

regions



anisotropy





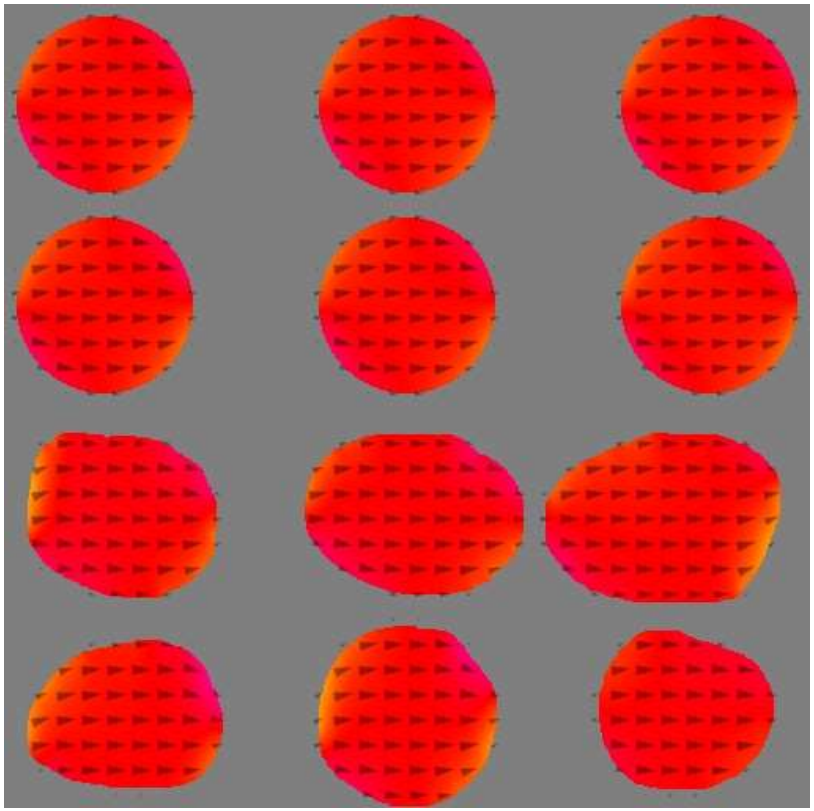
magnetization

GHENT UNIVERSITY

HYSTERESIS: COMPLEX GEOMETRY

```
//relax the magnetization in saturated state
B_ext=vector(0.1,0,0)
tableadd(B_ext)
relax()
```

```
H_min  :=  -0.1
H_max  :=   0.1
H_step :=   0.002

//ramp the field down
for H:=H_max;H>=H_min;H-=H_step{
    B_ext=vector(H,0,0)
    minimize()
    tablesave()
    snapshot(m)
}

//ramp the field up again
for H:=H_min;H<=H_max;H+=H_step{
    B_ext=vector(H,0,0)
    minimize()
    tablesave()
    snapshot(m)
}
```

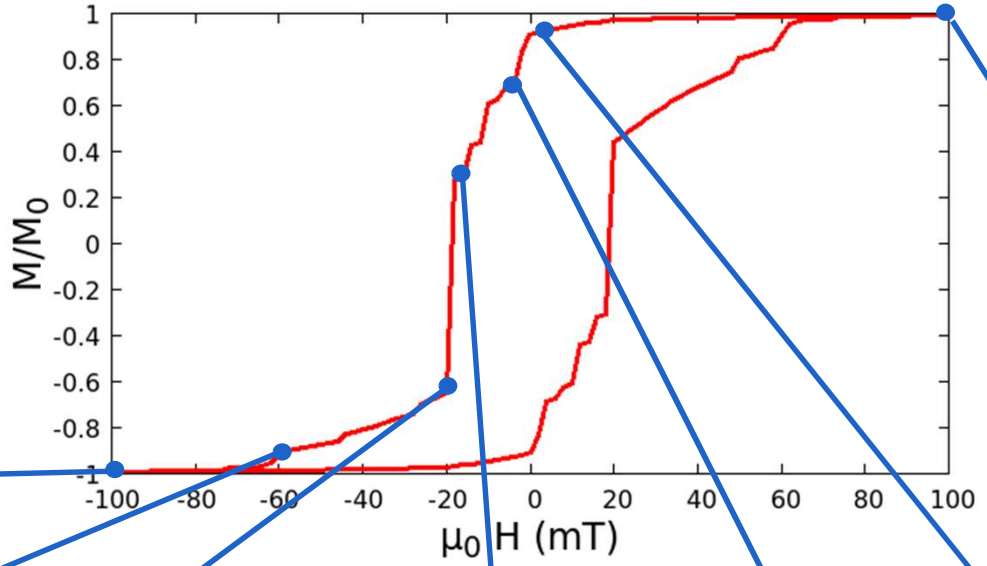


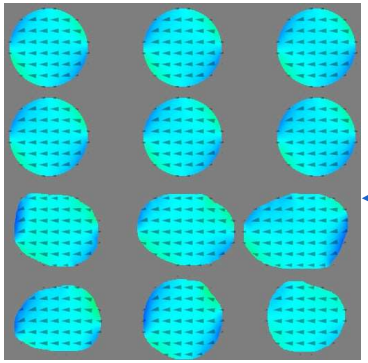
HYSTERESIS: COMPLEX GEOMETRY

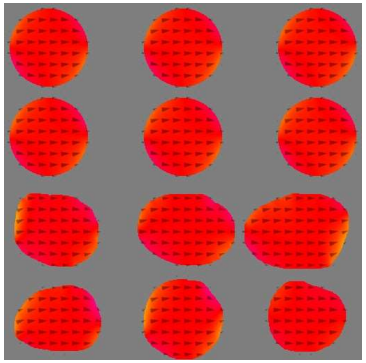
```
//relax the magnetization in saturated state
B_ext=vector(0.1,0,0)
tableadd(B_ext)
relax()
```

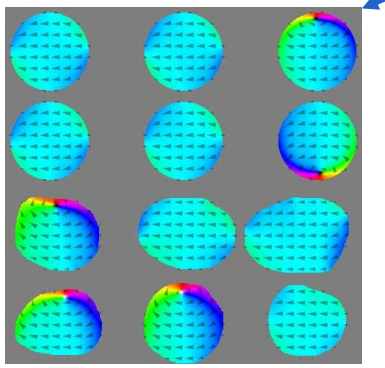
```
//set up the hysteresis loop parameters
B_min  :=  -0.1
B_max  :=   0.1
B_step :=   0.002

//ramp the field down
for B:=B_max;B>=B_min;B-=B_step{
    B_ext=vector(B,0,0)
    minimize()
    tablesave()
    snapshot(m)
}

//ramp the field up again
for B:=B_min;B<=B_max;B+=B_step{
    B_ext=vector(B,0,0)
    minimize()
    tablesave()
    snapshot(m)
}
```

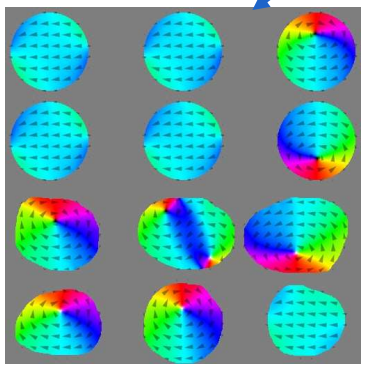
HYSTERESIS: COMPLEX GEOMETRY

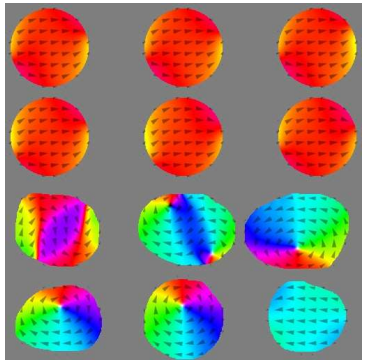


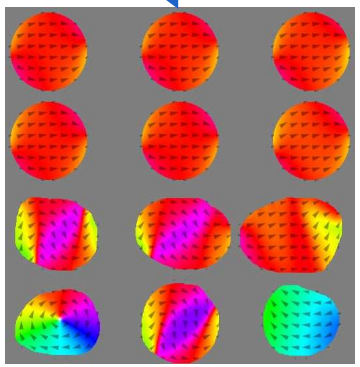


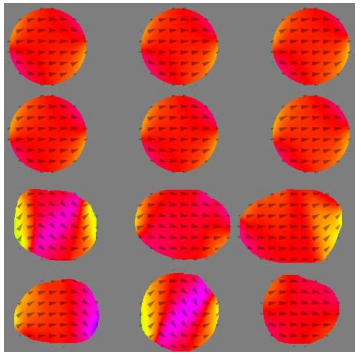








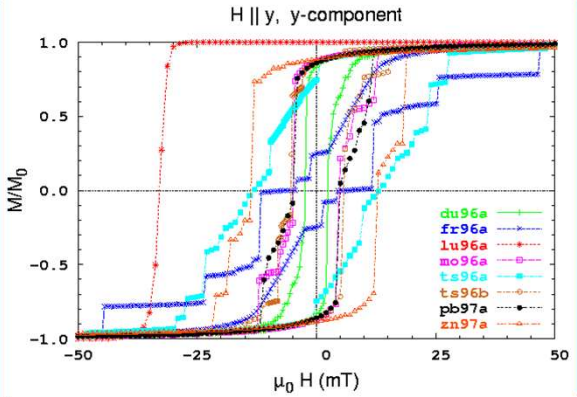


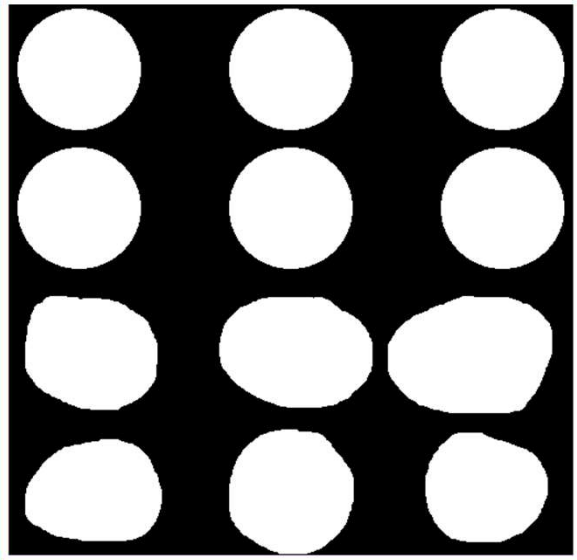


HYSTERESIS EXAMPLES

WHAT HAVE WE LEARNED?

- Be careful when simulating hysteresis loops
- Difference between `relax()` and `minimize()`
- How to construct complex geometries
  - `imageShape("drawing.png")`
  - Predefined shapes and methods (add, transl,…)
- How to set regionwise material parameters
- How to initialize the magnetization
  - `m.setinshape(...)`
  - `m.setregion(...)`





GHENT UNIVERSITY

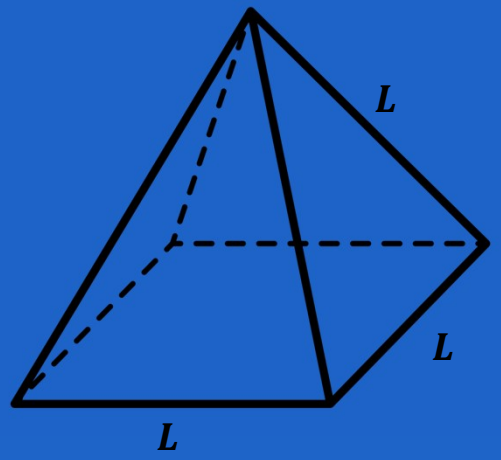
HOMEWORK ASSIGNMENT 1

Consider a ferromagnetic equilateral square pyramid with the following material parameters

- Exchange stiffness $A = 13\ \mathrm{pJ/m}$
- Saturation magnetization $M_{sat} = 800\ \mathrm{kA/m}$

Find the magnetic ground state of this structure for different side lengths L ranging from 50 nm to 100 nm.

Hint: this is exercise is very similar to standard problem 3

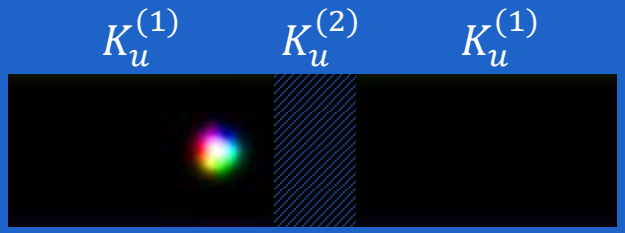


GHENT UNIVERSITY

HOMEWORK ASSIGNMENT 2

Goal: use a local voltage controlled magnetic anisotropy (VCMA) to create a skyrmion gate on a skyrmion racetrack [1]

Use the input script of the skyrmion racetrack as a starting point
Add a region in the center in which the uniaxial anisotropy is increased by 5% (i.e. the VCMA)



$$K_u^{(2)} = K_u^{(1)} + K_u^{VCMA} = 1.05K_u^{(1)}$$

What is the maximal current density for which the region with the VCMA acts as a skyrmion gate?

GHENT UNIVERSITY

[1] **Voltage Controlled Magnetic Skyrmion Motion for Racetrack Memory**
Wang Kang, Yangqi Huang, Chentian Zheng, Weifeng Lv, Na Lei, Youguang Zhang, Xichao Zhang, Yan Zhou & Weisheng Zhao
Scientific reports **volume 6**, pages 23164 (2016)

FACULTY OF SCIENCES

mumax.ugent.be

jonathan.leliaert@ugent.be

jeroen.mulkers@ugent.be

Many thanks to everyone who contributed with a benchmark after last week's session.

More benchmarks remain welcome!

GHENT UNIVERSITY

IEEE MAGNETICS

NVIDIA

SPINTRONICS