

Mumax3 lesson1

MICROMAGNETISM

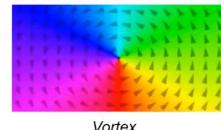
In ferromagnets, neighboring magnetic moments have the tendency to align

↓ *Continuum approx.*



Magnetization can be described by a continuous vector field

$$\mathbf{M}(\mathbf{r}, t) = M_s(\mathbf{r}) \underbrace{\mathbf{m}(\mathbf{r}, t)}_{\text{central quantity of interest}}$$



- picosecond time scale
- 1nm – 1μm length scale



5

The physics is fully described by the total magnetic energy functional:

$$E[\mathbf{m}] = \int_V \left\{ \begin{array}{lll} A(\nabla \mathbf{m})^2 & - \mu_0 \mathbf{M} \cdot \mathbf{H}_{ext} & - \frac{\mu_0}{2} \mathbf{M} \cdot \mathbf{H}_{demag} \\ Exchange & Zeeman & \end{array} + \right. \begin{array}{l} Crystal anisotropy \\ Dzyaloshinskii-Moriya \\ Magneto-elasticity \\ Higher-order exchange \\ ... \end{array} \right\} d^3\mathbf{r}$$



Two main objectives



Time integration dynamics

- Spin waves
- Domain wall motion
- Spin transfer torques
- Vortex excitation
-

Energy minimization statics

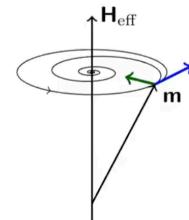
- Stable magnetic states
- Hysteresis curves
- Phase diagrams
- Domain wall profiles
- ...

The magnetization dynamics is described by the Landau-Lifshitz-Gilbert (LLG) equation

$$\dot{\mathbf{m}} = -\frac{\gamma}{1+\alpha^2} \left[\underbrace{\mathbf{m} \times \mathbf{H}_{eff}}_{\text{precession}} + \underbrace{\alpha \mathbf{m} \times (\mathbf{m} \times \mathbf{H}_{eff})}_{\text{damping}} \right]$$

with

$$\mathbf{H}_{eff} = -\frac{1}{\mu_0 M_s} \frac{\delta E}{\delta \mathbf{m}}$$



Computation of the magnetization dynamics:

$$E[\mathbf{m}, t] \rightarrow H_{eff}(\mathbf{r}) \xrightarrow[\square]{\text{LLG}} \mathbf{m}(\mathbf{r}, t)$$

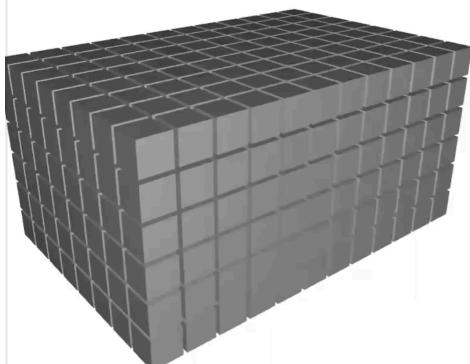
Mumax3's scripting language is a subset of golang

```
// saturation magnetization
Msat = 5e6

// declare new variable
Freq := 1e9

for ( i:=0; i<10; i++){
    print(i)
}

if 1+8 == 9 {
    print("Of course 1+8=9")
}
```



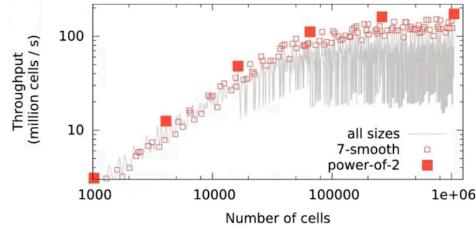
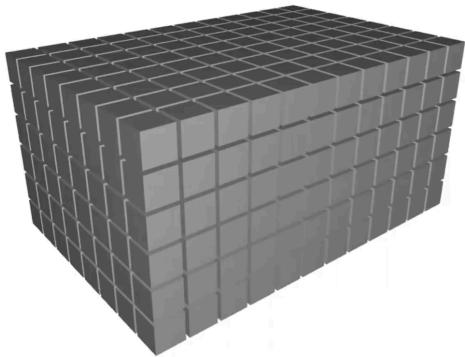
- Rectangular simulation box (origin in the center)
- Single regular rectangular grid
- Uniform magnetization inside cell $\mathbf{m}_{ijk} = \mathbf{m}(x_i, y_j, z_k)$
- Cell size < exchange length

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

- PBC values are number of virtual repetitions of the simulation box used to calculate dipolar interactions

```
setpbc(4,0,0)
```

The cuda fft library (used for the computation of the demag field) is highly optimized for grid size dimensions with small prime factors.



Tip: try to use grid size dimensions which are '7-smooth'

Example of good and bad grid size dimensions:

$$190 = 2 \cdot 5 \cdot 19 \quad 191 = 191 \quad 192 = 2^6 \cdot 3$$

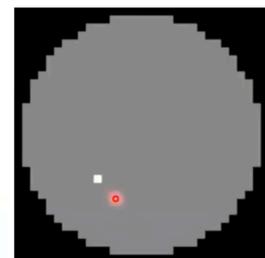


- Material parameters are assigned to the 256 regions.
- Material parameters can be functions of time
- There are vector and scalar material parameters
- Material parameters are predefined, they can not be created

```
// Assigning to a material parameter sets a value in all regions:  
Msat = 800e3  
AnisU = vector(1, 0, 0)  
  
// When regions are defined, they can also be set region-wise:  
Msat.SetRegion(0, 800e3)  
Msat.SetRegion(1, 540e3)  
  
// Material parameters can be functions of time as well:  
f := 500e6  
Ku1 = 500 * sin(2*pi*f*t)
```

- 256 regions in total (index 0 → 255)
- Each cell is assigned to a single region (default region id is 0)
- Each region has its own set of material parameters
- Two ways to set the region id in cells:
 1. Set region id of a single cell
 2. Set region id of all cells in a shape

```
SetGridSize(32,32,1)  
SetCellSize(1,1,1)  
  
// Set region id of cells in a circle to 1  
DefRegion(1, circle(30))  
  
// Set region id of cell (10,10,0) to 2  
DefRegionCell(2, 10, 10, 0)  
  
Save(regions)
```



- An excitation is a regional material parameter
- Additionally, one can add an arbitrary number of time- and space-dependent vector fields of the form :

$$g(x_i, y_j, z_k) * f(t)$$

Jeroen Mulkers

```
B_ext = vector(0,0,1)
B_ext.Add(LoadFile("antenna.ovf"), sin(2*pi*f*t))
B_ext.removeExtraTerms()
```

•

INITIAL MAGNETIZATION

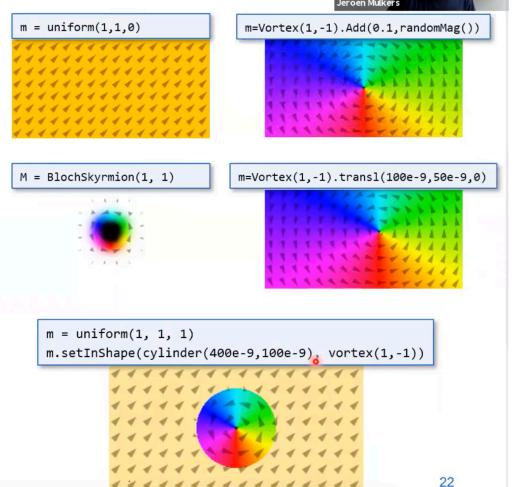
Config

```
Uniform(mx, my, mz)
RandomMag()
RandomMagSeed(seed)
TwoDomain(
    mx1, my1, mz1,
    mx2, my2, mz2,
    mx3, my3, mz3)
Vortex(circ, pol)
AntiVortex(circ, pol)
VortexWall(mxLeft, mxRight, circ, pol)
NeelSkyrmion(charge, pol)
BlochSkyrmion(charge, pol)
Conical(kVec, coneDir, coneAngle)
Helical(kVec)
```



Config methods

```
Transl(dx,dy,dz)
Scale(sx,sy,sz)
Add(ratio, config)
RotZ(angle)
```



Jeroen Mulkers

22

OUTPUT



3 output media:

- log file for input, logging and printing
- table.txt (t, mx, my, mz, ...)

```
tableadd(E_total)
tableaddvar(myVar,"myVar","unit")
tablesave() // write single line
tableautosave(1e-12) // write periodically
```

- .ovf files for scalar and vector fields

```
save(Edens_total)
saveas(Edens_total,"edens.ovf")
autosave(Edens_total, 1e-10) // write periodically
```



RUN/RELAX/MINIMIZE



- Solving the LLG equation (time integration)

```
run(timeperiod)
steps(100)
runWhile(condition)
```

- Minimizing the energy

```
relax() // LLG without precession
Minimize() // steepest descent[1]
```

```
// WARNING: ADVANCED SETTINGS
// In most cases, these settings can be ignored

// Set the solver:
// 1:Euler, 2:Heun, 3:Bogaki-Shampine, 4: Runge-Kutta(RK45),
// 5:Dormand-Prince(the default), 6:Fehlberg, -1:Backward Euler
SetSolver(5)

// set timestep
fixdt = 0 // if 0 (default): use adaptive timestep

// Advanced settings for adaptive timestep (default values are given)
Headroom = 0.8 // headroom dt correction
MaxDt = 0 // if 0, no maximal timestep
MinDt = 0 // if 0, no minimal timestep
MaxErr = 1e-5 // maximum allowed error/step

// Advanced settings for minimizer (default values are given)
MinimizerSamples = 10 //Number of max dM for convergence check
MinimizerStop = 1e-6 //Stopping max dM for Minimize
```



Effective field terms

- Demagnetization
- Exchange
- Anisotropy
- Dzyaloshinskii-Moriya
- External field
- Thermal field
- Custom field

Spin transfer torques

- Zhang-Li STT
- Slonczewski STT

Demagnetization energy density

$$\epsilon = -\frac{\mu_0}{2} M_s \mathbf{m} \cdot \mathbf{H}_{demag}$$

Regional Material Parameters	
Msat	Saturation magnetization (A/m)
NoDemagSpins	Disable magnetostatic interaction per region (default=0, set to 1 to disable).

Output Quantities	
B_demag	Magnetostatic field (T)
Edens_demag	Magnetostatic energy density (J/m3)
E_demag	Magnetostatic energy (J)

Other functionalities	
EnableDemag	Enables/disables demag (default=true)
SetPBC	Sets the number of repetitions in X,Y,Z to create periodic boundary conditions. The number of repetitions determines the cutoff range for the demagnetization.
DemagAccuracy	Controls accuracy of demag kernel



Exchange energy density

$$\varepsilon = A (\nabla \mathbf{m})^2$$

Harmonic mean for inter-region exchange coupling (default behavior)

$$\frac{A}{M_s} = 2 \frac{\frac{A_1}{M_{s_1}} \frac{A_2}{M_{s_2}}}{\frac{A_1}{M_{s_1}} + \frac{A_2}{M_{s_2}}}$$

Regional Material Parameters	
Aex	Exchange stiffness (J/m)
Msat	Saturation magnetization (A/m)

Output Quantities	
B_exch	Exchange field (T)
Edens_exch	Total exchange energy density (including DMI) (J/m3)
E_exch	Total exchange energy (including DMI) (J)
MaxAngle	Maximum angle between exchanged coupled spins (rad)

Other functionalities	
Ext_InterExchange	Sets exchange coupling between two regions
Ext_ScaleExchange	Re-scales exchange coupling between two regions

Zeeman energy density

$$\varepsilon = -M_s \mathbf{m} \cdot \mathbf{B}_{ext}$$

Regional Material Parameters	
Msat	Saturation magnetization (A/m)

Excitation	
B_ext	Externally applied field(T)

Output Quantities	
B_ext	Externally applied field(T)
Edens_zeeman	Zeeman energy density (J/m3)
E_zeeman	Zeeman energy (J)

Interfacially-induced DMI energy density

$$\varepsilon = D [m_z (\nabla \cdot \mathbf{m}) - (\mathbf{m} \cdot \nabla) m_z]$$

Bulk DMI energy density

$$\varepsilon = D \mathbf{m} \cdot (\nabla \times \mathbf{m})$$

Note:
Only single DMI type allowed at once

Regional Material Parameters	
Dind	Interfacially-induced DMI strength (J/m2)
Dbulk	Bulk DMI strength (J/m2)
Msat	Saturation magnetization (A/m)

Output Quantities	
B_exch	Exchange field (including DMI) (T)
Edens_exch	Total exchange energy density (including DMI) (J/m3)
E_exch	Total exchange energy (including DMI) (J)

Other functionalities	
Ext_InterDind	Sets Dind coupling between two regions
Ext_ScaleDind	Re-scales Dind coupling between two regions
OpenBC	Use open boundary conditions (default=false, use Neumann BC). This setting is only relevant for DMI.