MuMax³
GPU-accelerated micromagnetism

Jeroen Mulkers*¹,², Jonathan Leliaert¹

*²jeroen.mulkers@gmail.com
¹DyNaMat group, Department of Solid State Sciences, Ghent University, Belgium
²Department of Physics, Antwerp University, Belgium

Universiteit
Antwerpen

 GHENT UNIVERSITY
1. Intro micromagnetism
2. What is MuMax³
3. How to use MuMax³
4. Live Demo
5. Examples
1. Micromagnetism

Magnetisation described by a continuous vector field

\[ \vec{M}(\vec{r}, t) = M_{sat} \vec{m}(\vec{r}, t) \]

on the picosecond time scale and submicrometer length scale

Quasi-uniform state

Vortex

Néel Skyrmion

Néel Skyrmion
1. Micromagnetism

Magnetization in continuum approx.
\[ \mathbf{M}(x, y) = M_{\text{sat}} \mathbf{m}(x, y) \]

Local effective field
\[ \mathbf{H}_{\text{eff}} = -\frac{\delta E_{\text{free}}}{\delta \mathbf{m}} \]

Landau-Lifshitz-Gilbert equation for dynamics
\[ \dot{\mathbf{m}} = -\frac{\gamma}{1 + \alpha^2} \left[ \mathbf{m} \times \mathbf{H}_{\text{eff}} + \alpha \mathbf{m} \times (\mathbf{m} \times \mathbf{H}_{\text{eff}}) \right] \]
1. Micromagnetism

Interactions present in MuMax$^3$

In effective field:
- Exchange interaction
- Anisotropy
- Zeeman interaction
- Demagnetization
- Interfacially induced DMI
- Bulk DMI
- Thermal field
- Custom interaction

Extra LLG terms:
- Spin transfer torques

Not (yet) in MuMax$^3$
- Magnetoelastic effects
- ...

Jeroen.mulkers@gmail.com
2. What is MuMax³

- Finite-difference based micromagnetic simulation package
- Developed at DyNaMat (UGent) by Arne Vansteenkiste
- Active community groups.google.com/forum/#!forum/mumax2
- Well documented mumax.github.io
- Open source (GPLv3) github.com/mumax/3
- Mainly written in golang
- CUDA C kernels for heavy lifting
3. How to use MuMax$^3$

- Web interface with Mumax$^3$ running locally or remotely
- MuMax$^3$ scripting language
- Golang program  
  
  ```go
  import github.com/mumax/3/engine
  
  setgridsize(128, 32, 4)
  setcellsize(5e-9, 5e-9, 5e-9)
  Msat = 860e3
  Aex = 13e-12
  alpha= 0.2
  m=uniform(1, 1, 0)
  
  f := 1e9 // 1GHz
  A := 0.01 // 10mT
  B_ext = vector(0.1, A*sin(2*pi*f*t), 0)
  
  run(10e-9)
  ```
3. How to use MuMax³
Basic concepts

- MuMax³'s scripting language is subset of golang
- MuMax³ scripts have the .mx3 file extension

```plaintext
// saturation magnetisation
Msat = 5e6

// declare new variable
Freq := 1e9

// External time dependent field
B_ext = vector(0,0,0.1*sin(2*pi*Freq*t))

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

if 1+8 == 9 {
  print("Of course 1+8=9")
}
```
3. How to use MuMax$^3$: Basic concepts

- Scripting
- Grid and cells
- Shapes
- Regions
- Output
- Run/Minimize

- MuMax$^3$ uses finite differences on a regular rectangular grid
- Single magnetic moment in every cell (possibly zero)
- Grid dimensions preferably $2^n$
- Cell size < exchange length
- PBC values are number of virtual repetitions of the simulation box used to calculate dipolar interactions

```plaintext
setcellsize(4e-9,4e-9,4e-9)
setgridsize(128,64,4)
setpbc(2,4,0)
```
2. What is MuMax³

- Finite-difference based micromagnetic simulation package
- Developed at DyNaMat (UGent) by Arne Vansteenkiste
- Active community [groups.google.com/forum/#!forum/mumax2](groups.google.com/forum/#!forum/mumax2)
- Well documented [mumax.github.io](mumax.github.io)
- Open source (GPLv3) [github.com/mumax/3](github.com/mumax/3)
- Mainly written in golang
- CUDA C kernels for heavy lifting
3. How to use MuMax³:
Basic concepts

- Scripting
- Grid and cells
- **Shapes**
- Regions
- Output
- Run/Minimize

- Does not depend on the grid
- Default location: center of universe (0,0,0)
- Object methods to translate, rotate, add, intersect, ...

```plaintext
myCirc := circle(100e-9)
myRect := rect(60e-9,60e-9).rotZ(pi/4).transl(30e-9,0,0)
pacman := myCirc.Sub(myRect)
```

- To outline the geometry of your magnet: `setgeom(pacman)` sets magnetization to zero in cells outside the shape
3. How to use MuMax³
Basic concepts

- Scripting
- Grid and cells
- Shapes
- Regions
- Output
- Run/Minimize

- Each cell is assigned to a single region
- 256 regions in total (index 0 → 255)
- Each region can have its own set of material parameters
- A region can be defined by a shape

```c++
// Circular region with large Msat
Msat = 500e3 // A/m
myCirc := circle(20e-9)
Defregion(1,myRect)
Msat.setRegion(1,1000e3)
```
3. How to use MuMax³:
Basic concepts

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 voor scalar and vector fields
  ```
  save(Edens_total)
  saveas(Edens_total,"edens.ovf")
  autosave(Edens_total 1e-10) // write peridically
  ```

- Scripting
- Grid and cells
- Shapes
- Regions
- Output
- Run/Minimize
3. How to use MuMax³:
Basic concepts

• Scripting
• Grid and cells
• Shapes
• Regions
• Output

• Run/Minimize

Most important solver settings

SetSolver(5) // 5th order Dormand-Prince
fixdt = 1e-14 // if 0: adaptive

Solve LLG equation for certain period of time, number of steps, or until condition is met

run(time)
steps(100)
runWhile(condition)

Relaxation using LLG without precession term

relax()

Steepest gradient method to minimize the energy

minimize()
4. Live demo
5. Examples
vortex eigenmodes

Mumax3-fft for time→frequency Fourier transform
5. Examples
Artificial spin ice

MFM image generation
5. Examples
Skyrmion racetrack

- Moving skyrmion due to STT
- Spatially dependent interfacially induced DMI

$$D_{in} = 0.8D_c$$

5. Examples
Modelling exchange bias

- Define custom interaction (in script) for a negative exchange coupling between layers to create a compensated antiferromagnet

\[ J_{I} \]
\[ \delta t_{AFM} \]

Any questions?