MICRO-435

Exercises W13: Magnetic Nanostructures

Simulation of NML at physical level

In these lab exercises, you will learn how to run low-level "physical" simulations of magnetic circuits. Several software packages are available for free to simulate magnetic dynamics (OOMMF, MAGPAR, NMAG, MUMAX). The software you will use is called OOMMF, and it was developed at the National Institute of Standards and Technology (NIST). You can download it for free at *http://math.nist.gov/oommf/* (on UNIX systems, it must be installed compiling the source code, whereas, on Windows, a pre-built .exe file is available). OOMMF is the fastest micromagnetic simulator on CPU, MUMAX is much faster but requires a GPU. OOMMF uses finite-elements solvers and simulates the dynamic behavior of any magnetic material of any shape, therefore it can simulate NanoMagnet Logic (NML) circuits accurately. **Since OOMMF is based on FEM simulations, the system is divided into a mesh, thus magnets are not considered single-domain magnets.** Remember it during the lab to correctly analyze the wire behavior.

Physical Simulations of a NML wire

Since OOMMF is a free software, advanced functions are not available, and to successfully launch a simulation requires a substantial dose of handwork. Unfortunately, this is the price you pay with many free, open-source software. To launch the software locate the installation path and double click on **oommf.tcl**, see Figure 1.
After opening OOMMF, a small window will appear with a white box inside: check on it. In the enlarged window (see Figure 2) check the white box between *micronanosys* and *Your-User-Name*. The OOMMF main window will be enlarged, showing all the options available. A short description follows.

- *mmArchive*. Utility used to save simulation data.
- *mmDataTable*. Utility used to save data on tables.
- *mmDisp*. Integrated environment to show simulation results on a graphic form.
- *mmGraph*. Utility to generate 2D plots from simulation results.
- *mmProbEd*. Utility to define a simulation parameters for the old 2D simulation engine.
- *mmSolve2D*. Old 2D simulation engine.
- *Oxsii*. 3D simulation engine.

For this lab exercise you will use only the *Oxsii* and the *mmDisp* utilities.

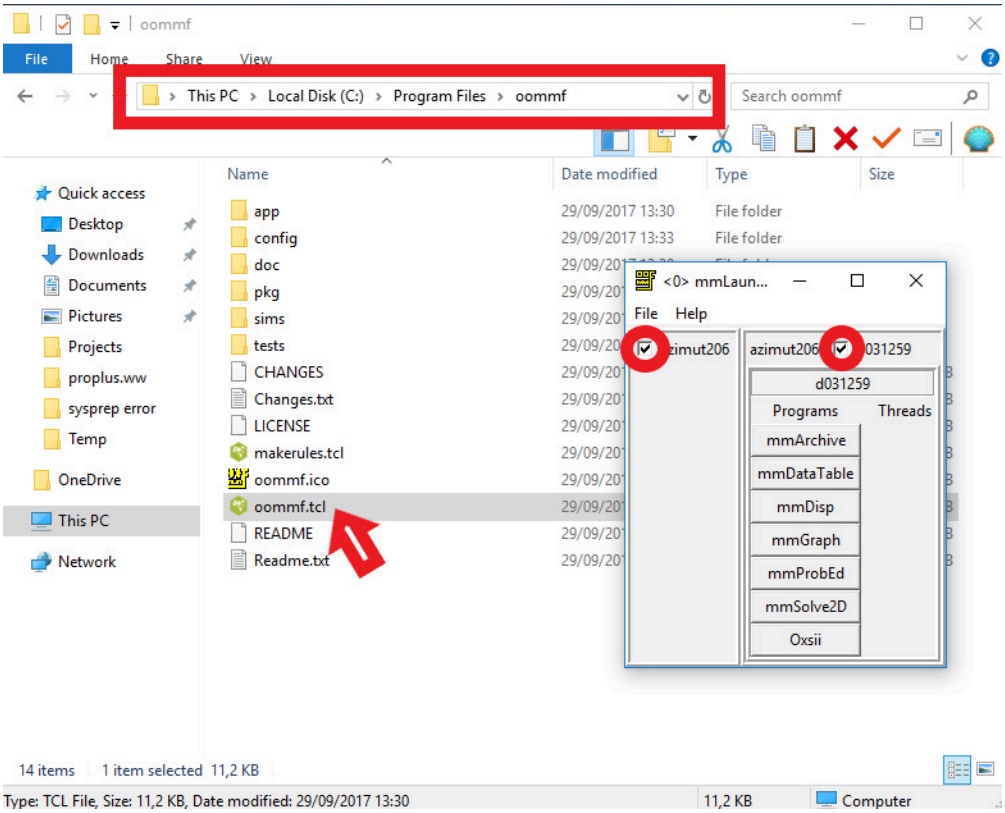

Figure 1: Oommf installation folder

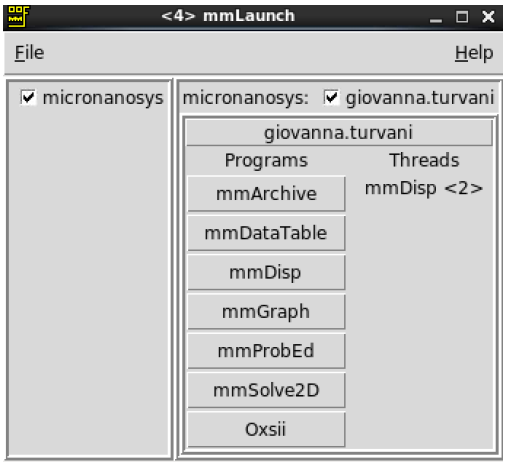

Figure 2: Oommf First Run

NMLOOMMF_focus1: How to run a simulation using OOMMF

The simulation process in OOMMF follows several steps:

1. Geometry creation and problem definition;
2. Launching the simulation;
3. Results visualization.

As an example and to take confidence with the tools, you will simulate a simple circuit. The circuit is composed of one input magnet and a wire of 4 magnets. Magnet sizes are 50x100x20 nm^3 while the separation distance is 20 nm. Magnets are made of permalloy ($Ni_{80}Fe_{20}$), a widely used material. The simulation macro-steps are highlighted in Figure 3.

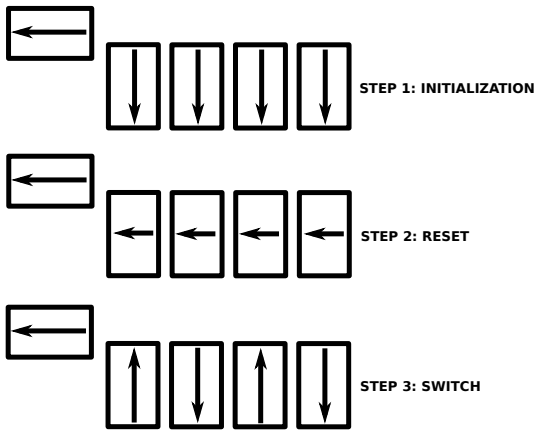

Figure 3: Basic simulation.

Magnets are firstly initialized, forcing their magnetization vector alongside the Y-axis pointing down. The choice of this initial state is made for the sake of simplicity. It is worth underlining that this is not the minimum energy state. Instead, the horizontal magnet is initialized along the X-axis since it is an input. Then an external magnetic field of 75 kA/m is applied to all magnets. This magnetic field forces the magnetization vector alongside the X-axis in an unstable state. Finally, the magnetic field is removed, and the magnets switch according to the input magnet. The required configuration file for this simulation is provided. The structure of the input file is described in the final section of this text.

To launch a simulation follow these steps.

- Click on the 3D solver *Oxsii*.
- Check the white box near *Oxsii* $< n >$, a new window should appear.
- In the new window select *File* $\Rightarrow$ *Load*.

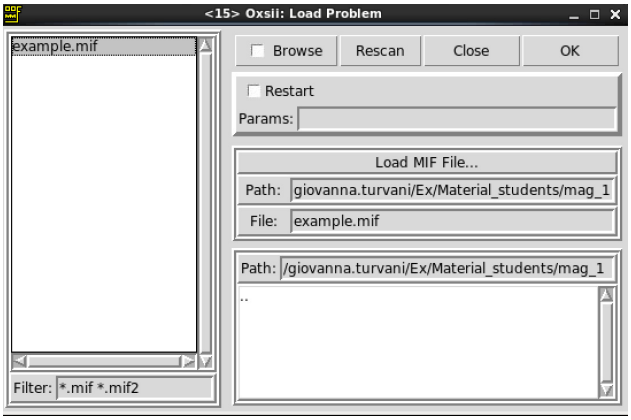

Figure 4: .mif file loading

- Navigate through the folders and double click on the file *example.mif*, a new window *mmDisp* should appear.
- In the *Oxsii* window, among the outputs click on *Oxs_MinDriver::Magnetization*.

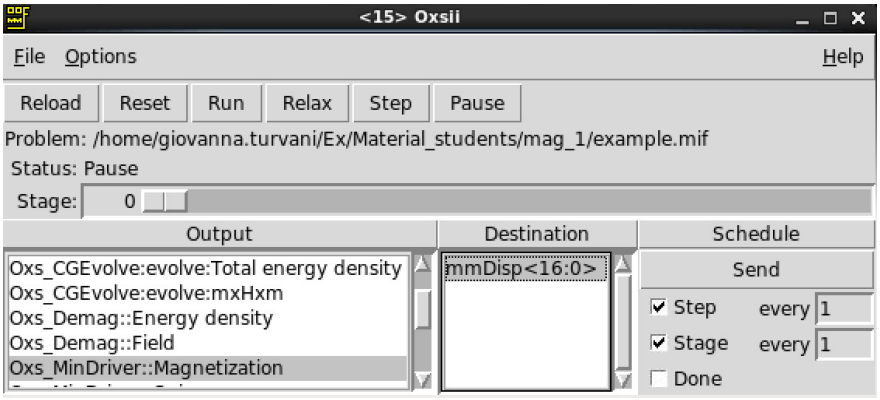

Figure 5: The *Oxsii* window

- In the *Destination* sub-window, a new option *mmDisp<n:x>* should appear, click on it.
- In the *Schedule* sub-window, check both *Step* and *Stage* and then click on *Send*.
- In the *mmDisp* window, the circuit should now appear. Each magnet is identified by a set of arrows representing magnetization vectors.
- Enlarge the *mmDisp* window until it fills the screen, then select *View* ⇒ *Fill Dysplay*. This will adapt the arrow sizes automatically.

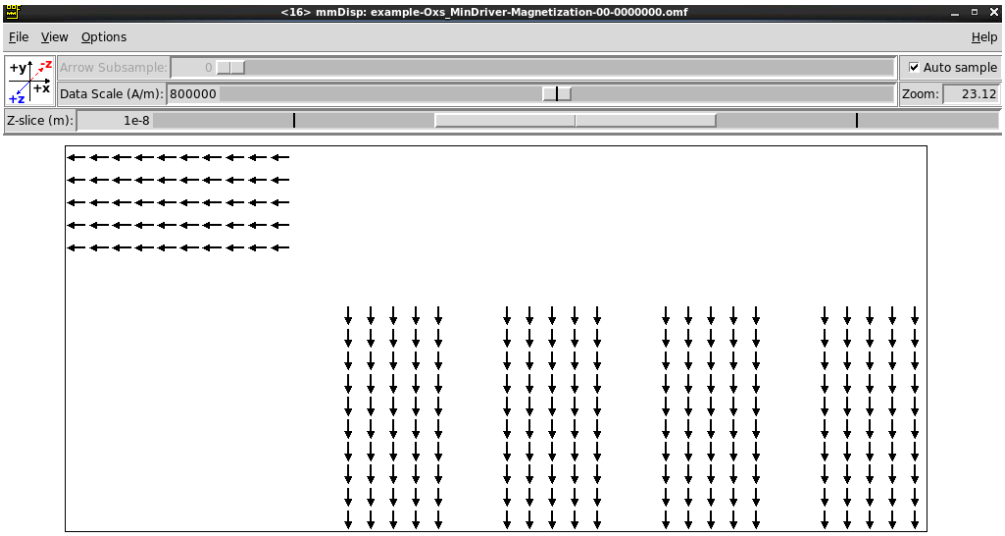

Figure 6: Sample set-up

- By moving the slice *Data Scale (A/m)* you can change the arrow sizes, whereas the number of arrows can be changed moving the slider *Arrow Subsample*. These functionalities are helpful, especially with big circuits, to identify the various magnets better.

- Run the simulation by clicking *Run* in the *Oxsii* window, you can now see runtime the evolving state of the circuits in the *mmDisp* window.

NMLOOMMF_focus1_Q1: Explain the magnetic behavior of the simulated wire.

NMLOOMMF_focus2: Magnetic field strength

The magnetic field strength is linked to the magnets geometry and position. A magnetic field too weak cannot force magnets into the reset state, so they may not switch correctly. At the same time, to reduce power consumption, it is essential to use the lowest possible value of the magnetic field. This exercise aims to find the lowest value of the magnetic field that assures a successful switching. **Read file *LAB3_theory.pdf* to understand how to modify the file to increase the magnetic field strength.**

NMLOOMMF_focus2_Q2: Copy and rename the *example.mif* configuration file. Decrease the magnetic field strength gradually. What is the minimum value of the magnetic field that assures a correct switching?

OPTIONAL_BASE_EX3: Magnets sizes

The magnetic field strength is linked to the magnets aspect ratio. Reducing the magnets aspect ratio reduces the magnetic field strength, whereas increasing the aspect ratio increases the magnetic field strength. However, changing magnet sizes also affects the dynamic switching behavior of magnets.

Copy and rename the *example.mif* configuration file. Increase the height of the magnets to 150 nm (remember to adjust the input magnet position to keep the same distance with the first magnets of the wire).

OPTIONAL_BASE_EX3_Q3: What is the minimum magnetic field required to assure a successful switching?

Copy and rename the *example.mif* configuration file. Decrease the height of the magnets to 80 nm (remember to adjust the input magnet position to keep the same distance with the first magnets of the wire). Then,

OPTIONAL_BASE_EX3_Q4: What is the minimum magnetic field required to assure a successful switching

Copy and rename the *example.mif* configuration file. Continue to decrease the height of the magnets until the wire still behaves correctly (remember to adjust the input magnet position to keep the same distance with the first magnets of the wire).

OPTIONAL_BASE_EX3_Q5: Can you understand, from the simulation, why it does not work when the aspect ratio is too small?

NMLOOMMF_focus4: Magnet distances

The distance among magnets is an important parameter because higher distances simplify the fabrication process. However, changing the distance among magnets affects both the magnetic field strength and the magnets dynamic behavior. **Read file *LAB3_theory* to understand how to modify the distance among magnets.**

NMLOOMMF_focus4_Q6: Copy and rename the *example.mif* configuration file. Reduce distances among magnets to 10 nm (horizontal distance, also for the driver). Simulate the circuit with a magnetic field strength of 75 kA/m and 100 kA/m. Concerning the transient and regime operations, how does the behaviour change? What are the consequences, in your opinion, at the circuit level? (e.g. power consumption).

NMLOOMMF_focus4_Q7: Increases distances by 10 nm each time until you find the maximum distance that allows the circuit to work as intended, keep the magnetic field strength to 75 kA/m. What is the maximum working distance?

NMLOOMMF_focus4_Q8: When you find the maximum distance (previous question), further increase the distance and try to increase the magnetic field strength up to 150 kA/m. Does the circuit begin to work again?

NMLOOMMF_focus5: Wire length

The number of magnets that can be chained is limited, not only for thermal noise but also depending on the applied magnetic field and magnetic dynamics.

NMLOOMMF_focus5_Q9: Copy and rename the *example.mif* configuration file. Add a fifth magnet to the chain. Does the circuit behave correctly?

Stabilizing the magnetization of magnets is necessary for correctly propagating signals. However, unlike horizontal wires, nearest neighbor dipole fields in vertical wires oppose the magnetization direction, hindering hard axis stability. Typically, the field strength needed to stabilize horizontal wires does not ensure a successful logic propagation in vertical wires.

A solution to this instability is to place blocks of magnetic material with easy axes always parallel to the applied clocking field direction, producing a static bias along a nanomagnets hard axis. These "helper blocks" provide an additional Zeeman energy term to the nanomagnet, acting to keep nanomagnets magnetized along their hard axis until the signal reaches it. Helper blocks do not change the "logic" behavior of the circuit and can thus be used to avoid signal propagation errors. Also, they reduce external field requirements making it possible to send a signal over long distances and create copies of itself (fanout).

In this exercise, you will use a helper block at the end of the NML wire as a sort of "termination" of the line to assure correct signal propagation.

NMLOOMMF_focus5_Q10: Try to add a so-called "helper block", a magnet with a width of 200 nm and a height of 100 nm. Dispose it at the same height as the fifth magnet and next to it (Figure 7). Compare the two simulations (with and without the helper block). How do you explain

the difference in the behavior?

NMLOOMMF_focus5_Q11: Increase the number of magnets by adding one magnet each time (keeping the helper block at the end). What happens if you increase the number of elements, do you think there is a maximum number of elements that can be chained without errors in the signal propagation?

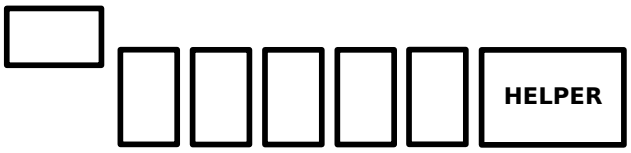

Figure 7: Helper block.

NMLOOMMF_focus6: Vertical wires

The target of this exercise is to simulate a vertical wire and to see how the magnetic field and helper blocks affect circuits behavior. To define the configuration file for this part use as a template the example.mif file given.

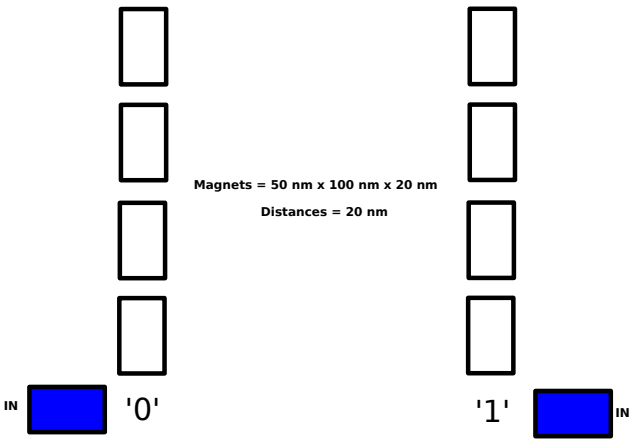

Figure 8: Vertical wire without helper blocks

1. Copy and rename the example.mif configuration file.
2. Change the geometry according to Figure 8.
3. Simulate the circuit with a magnetic field of 75kA/m. **Question 1:** Does it works correctly?
4. Modify the circuit adding helper block according to Figure 9 and simulate it, with both logic 0 and 1, using a magnetic field of 75kA/m.

   NMLOOMMF_focus6_Q12: How the circuit behaves?

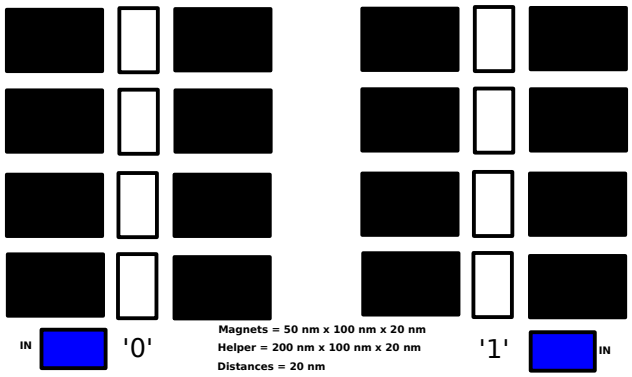

Figure 9: Vertical wire with helper blocks

NMLOOMMF_focus7: Majority voter

Simulate the ideal majority voter, which structure is shown in Figure 10.

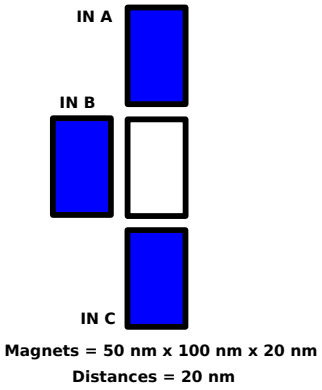

Figure 10: Ideal majority voter structure

1. Open and execute the configuration file (*maj_vot000.mif*).
2. **Question 3: Does it behaves correctly?**
3. Starting from the provided configuration file modify it in order to simulate the circuit with another input pattern (try "010" or "110")). **NMLOOMMF_focus7_Q13:** Does the circuit behaves correctly from a logic point of view?

Geometry creation and problem definition in OOMMF

To simulate a magnetic structure using OOMMF, you must use a configuration file (.mif). **The *example.mif* file can be opened using any text editor. To open it, right-click on it and select *Open with Other Application*. Scroll down the menu and select *gedit* then click on *Open*.**
The configuration file can be divided into three parts:

- Geometry definition;

- Problem description;
- Numerical solver and simulator settings.

Geometry definition

Magnets are created using the command *Specify Oxs_BoxAtlas: name* {}.

```
Specify Oxs_BoxAtlas:input1 {
xrange { 10e-9 110e-9 }
yrange { 120e-9 170e-9 }
zrange { 0 20e-9}
name i1
}
```

For example in this case we are creating a magnet, called *input1*, with a width of 100 (*xrange { 10e-9 110e-9 }, 110 - 10 = 100*), an height of 50nm (*yrange { 120e-9 170e-9 }, 170 - 120 = 50*), and a thickness of 20nm (*zrange { 0 20e-9}, 20 - 0 = 20*). At this magnet a label called *i1* is attached. OOMMF uses an absolute coordinate system, so the X-axis position of the next magnet in the chain starts at 130nm.

```
Specify Oxs_BoxAtlas:mag1 {
xrange { 130e-9 180e-9 }
yrange { 0 100e-9 }
zrange { 0 20e-9}
name m1
}
```

Following the same syntax, the following magnet has a width of 50nm, a height of 100nm, a thickness of 20nm, and is placed 20nm after the input magnet. The relative X-axis displacement can be understood by looking at the *xrange* values of both magnets: the first magnet ends at 110nm while the second magnet starts at 130nm, 130 - 110 = 20. This second magnet is also located in a lower Y-axis position. The relative Y-axis displacement can be understood by comparing the *yrange* values of both magnets: the first magnet starts at 120nm, while the second magnet ends at 100nm. The description of the other three magnets is straightforward.

```
Specify Oxs_BoxAtlas:mag2 {
xrange { 200e-9 250e-9 }
yrange { 0 100e-9 }
zrange { 0 20e-9}
name m2
}

Specify Oxs_BoxAtlas:mag3 {
xrange { 270e-9 320e-9 }
yrange { 0 100e-9 }
zrange { 0 20e-9}
name m3
}

Specify Oxs_BoxAtlas:mag4 {
xrange { 340e-9 390e-9 }
```

```
yrange { 0 100e-9 }
zrange { 0 20e-9}
name m4
}
```

As can be clearly seen, the *yrange* and *zrange* values are the same because magnets are aligned both on the Y-axis and Z-axis. The only difference lies in the *xrange* value, each magnet is placed 20nm after the previous magnet. The following command

```
 Specify Oxs_MultiAtlas:struct {
atlas input1
atlas mag1
atlas mag2
atlas mag3
atlas mag4
}
```

creates a simulation object that groups all the previously defined objects. It is used to define conditions globally applied to the whole circuit. The construct

```
 Specify Oxs_RectangularMesh:mesh [subst {
  cellsize {$cellsize $cellsize $cellsize}
  atlas :struct
}]
```

along with the statement

```
Parameter cellsize 10e-9
```

defines the mesh applied. In this case, a cubic mesh with a 10nm side is used. Simulation time depends exponentially on the mesh size: the smaller the mesh, the longer the simulation. Normally, a value of 5nm or less should be used in these simulations, but, for this lab exercise, 10nm is used to have manageable simulation times.

Problem definition

The problem definition statements are used to define magnetic material properties and set simulation conditions, such as initial magnetization and applied magnetic fields.

```
Specify Oxs_UniformExchange {
   A  16.0e-12
}
```

This command defines the exchange constant of the material.

```
Specify Oxs_Demag {}
```

This command enables the calculation of the self-magnetization component for each magnet.

```
Specify Oxs_UZeeman {
  Hrange {
     {    -75e3  0  0   0  0  0   5 }
  }
}
```

This command defines a magnetic field, uniformly applied to the whole circuit. The magnetic field has an initial value of -75kA/m and reaches 0 in 5 simulation steps. Magnetic field initial and final value is defined with three numbers (-75e3 0 0 and 0 0 0) which defines the three components of the vector (x,y,z). As a consequence in this case the magnetic field is applied along the X-axis. Multiple lines can be added to this command to define different waveforms. For example

```
Specify Oxs_UZeeman {
  Hrange {
     {    -75e3  0  0       0  0  0   5 }
     {        0  0  0    75e3  0  0   10 }
  }
}
```

defines a magnetic field with goes from -75kA/m to 0 in 5 simulation steps and then from 0 to 75kA/m in 10 simulation steps.

```
lappend m0_value_list "i1" [list -1 0 0]
lappend m0_value_list "m1" [list 0 -1 0]
lappend m0_value_list "m2" [list 0 -1 0]
lappend m0_value_list "m3" [list 0 -1 0]
lappend m0_value_list "m4" [list 0 -1 0]
```

This command is used to define the initial relative magnetization of each magnet. Each magnetization is divided into three vector components (x,y,z). The value is relative, that is, it can change from -1 to 1. In this case, the input magnet is initialized along the X-axis (vector pointing in the left direction). In contrast, the other magnets are initialized along the Y-axis (vector pointing downward).

```
Specify Oxs_AtlasVectorField:m0 [subst {
   atlas :struct
   norm 1.0
   default_value {1 0 0}
   values { $m0_value_list }
}]
```

This command simply applies the previously defined magnetizations to the magnets. An additional magnetic material parameter is defined in the following construct

```
 Specify Oxs_MinDriver [subst {
 evolver :evolve
 stopping_mxHxm 0.1
 mesh :mesh
 Ms { Oxs_AtlasScalarField {
    atlas :struct
    default_value 8e5
    values {
       universe 0.0
    }
 }}
 m0 :m0
}]
```

The magnetization saturation *Ms* of the permalloy is 8e5A/m.

Numerical solver and simulator settings

These commands are used to set the numerical solver and the simulator. They are described for the sake of completeness, but they are not relevant for this lab exercise.

```
Specify Oxs_CGEvolve:evolve {}
```

This command initializes the numerical solver.

```
Specify Oxs_MinDriver [subst {
 evolver :evolve
 stopping_mxHxm 0.1
 mesh :mesh
 Ms { Oxs_AtlasScalarField {
    atlas :struct
    default_value 8e5
    values {
       universe 0.0
    }
 }}
 m0 :m0
}]
```

This command defines the numerical solver parameters.

```
Destination output mmDisp
Schedule Oxs_MinDriver::Magnetization output Stage 1
```

This command defines which magnetic parameter to show as output (the magnetization in this case) and where to show it, in this case in the OOMMF integrated display environment.