MODELLING MATTER

Practical 1 | Atomistic simulations with VAMPIRE

Précis

The aim of this practical is to perform atomistic simulations of a magnetic nanoparticle and calculate the Curie temperature using the VAMPIRE software package. Here you will develop your linux skills and learn how to run a command-line based scientific software package.

Step 1 | Download and compile the VAMPIRE software package

The first step is to download and compile the VAMPIRE software package. Many codes are open source and managed using version controlled software. GitHub is a website that hosts open source software packages using git version control software. The first step involves downloading the source code from the website. On the command line, in your home directory, type:

```
git clone https://github.com/richard-evans/vampire.git
```

This will create a folder called `vampire` and place the source code inside. The next step is to compile the code using make: software that automates the building of more complicated software projects with many files. First change directory into the vampire folder.

```
cd vampire
```

To compile the code type

```
make serial -j 4
```

which will compile the code to run on your computer. Once the code is compiled you can test the code by running the default inputs using

```
./vampire-serial
```

This will run a quick test of the code and print some output on the screen.

Step 2 | Create the input files for a Curie-Temperature simulation

Once the code is compiled and running successfully, the next step is to set up a proper calculation where we want to calculate the Curie temperature of an Iron nanoparticle. VAMPIRE requires two files for each simulation: an ***input*** file and a ***material*** file. In this step we will create a new folder for the simulation and create an input and material file for VAMPIRE to read.

Each simulation should use a different folder to avoid overwriting previous files. First change back to your home directory and then create a new directory called `curie` to store the files for the next simulation.

```
cd ~

mkdir curie

cd curie
```

The next stage is to create files for vampire to read using the touch command (which creates empty files.

```
touch input

touch Fe.mat
```

Step 3 | Create the material file for bulk Fe

The next step is to write the key words in the material text file so that VAMPIRE can read them Use a text editor such as atom, VSCode or gedit to open the material file.

```
gedit Fe.mat
```

The material file specifies the properties of a magnetic material that you would like to simulate. In this case we would like to simulate bulk iron and so we need to find some basic information about the magnetic properties of the system. The parameters for bulk Fe are available from the VAMPIRE methods paper [1] in Table 2 on page 6, alongside other materials such as Co, Ni and Gd.

A detailed list of commands for VAMPIRE is given in the manual, available online

https://vampire.york.ac.uk/resources/vampire-manual.pdf

Along with descriptions of what each of the parameters does. In this simple example we only use a few basic commands, but more complex simulations and materials are possible.

As we would like to simulate bulk iron, enter the following text into the material file `Fe.mat`.

```
#----------------------------------------------------
# VAMPIRE material for bulk Fe
#----------------------------------------------------
#----------------------------------------------------
# Number of Materials
#----------------------------------------------------
material:num-materials = 1
#----------------------------------------------------
# Material 1 iron Generic
#----------------------------------------------------
material[1]:material-name = iron
material[1]:damping-constant = 1.0
material[1]:exchange-matrix[1] = 7.050e-21
material[1]:atomic-spin-moment = 2.22 !muB
material[1]:cubic-anisotropy-constant = 5.65e-25
material[1]:material-element = Fe
```

All VAMPIRE material files must specify a number of materials using

```
material:num-materials = 1
```

which in this example is 1. For more complicated materials this can be larger, and also include different magnetic sublattices. The important magnetic parameters for Monte Carlo simulations are the exchange constant between atoms

```
material[1]:exchange-matrix[1] = 7.050e-21
```

and the magnetic anisotropy, determining the preferred direction of the magnetic moments which in iron has cubic symmetry

```
material[1]:cubic-anisotropy-constant = 5.65e-25
```

The number in the square brackets indicates the material number which is always one in this case as we only have a single material. For systems with more than one material you would normally include exchange interactions between all kinds of atoms, eg for two materials we would have

```
material[1]:exchange-matrix[1] = ...
material[1]:exchange-matrix[2] = ...
material[2]:exchange-matrix[1] = ...
material[2]:exchange-matrix[2] = ...
```

The magnetic moment length depends on the configuration of electronic orbitals, and for Fe we have a moment of 2.2 Bohr magnetons and specified using

```
material[1]:atomic-spin-moment=2.22 !muB
```

Note that all text after a # is a comment and ignored by VAMPIRE allowing you to enter comments. Different units are also supported using the ! character. After saving the

completed file after editing we are now ready to set up the system and kind of VAMPIRE simulation we want to do.

Step 4 | Create the input file for a Curie temperature simulation

The next step is to create the input file for the simulation. This controls general properties such as the system size, and what kind of simulation is done by VAMPIRE. Copy the following text into the `input` file

```
#----------------------------------------
# Creation attributes
#----------------------------------------
create:crystal-structure = bcc
create:periodic-boundaries-x
create:periodic-boundaries-y
create:periodic-boundaries-z

#----------------------------------------
# System dimensions
#----------------------------------------
dimensions:unit-cell-size = 2.86 !A
dimensions:system-size-x = 5 !nm
dimensions:system-size-y = 5 !nm
dimensions:system-size-z = 5 !nm

#----------------------------------------
# Material File
#----------------------------------------
material:file = Fe.mat

#----------------------------------------
# Simulation attributes
#----------------------------------------
sim:minimum-temperature = 0.0
sim:maximum-temperature = 1600.0
sim:temperature-increment = 50.0
sim:equilibration-time-steps = 1000
sim:loop-time-steps = 1000
sim:time-steps-increment = 1

#----------------------------------------
# Program and integrator details
#----------------------------------------
sim:program = curie-temperature
sim:integrator = monte-carlo

#----------------------------------------
# data output
#----------------------------------------
output:temperature
output:mean-magnetisation-length
```

```
#----------------------------------------
# screen output
#----------------------------------------
screen:temperature
screen:mean-magnetisation-length
```

The input file specifies a body-centred crystal structure with a lattice constant of a = 2.866 Angstroms and a total system size of (5 nm)^3. The system size is large enough to give sensible statistics but small enough to run the simulation in a few minutes. The material file is loaded with the command

```
material:file = Fe.mat
```

which reads in the material file you generated in step 3. In the simulation attributes we set the minimum and maximum temperature, as the Curie temperature program performs an automatic loop over different temperatures. The temperature increment specifies the temperature steps between these ranges. In this example, the temperature will start at T = 0 K, then increase to 1600 K in steps of 50 K. The `equilibration-time-steps` parameter allows the system to reach thermal equilibrium so that in the data collection phase the system is in thermal equilibrium. The `loop-time-steps` parameter gives the number of Monte Carlo steps done at each temperature, set to 1000 for speed. Production runs would normally use 10,000 - 1,000,000 steps per temperature. The time-steps-increment parameter specifies how many steps between taking statistical samples, in this case only 1.

The program section specifies that we will perform a Curie temperature simulation which automatically cycles through different temperatures and we also specify the Monte Carlo algorithm which is especially efficient for simulating equilibrium properties. The final section specifies the data output. To calculate the Curie temperature, we need to simulate the temperature dependence of the mean magnetisation. Thus the data output needs to include the temperature and mean-magnetisation-length. To check the simulation is running we also output the data to the command line as well as the output file.

Step 5 | Run the Curie temperature simulation

The next step is to run the VAMPIRE software to perform the simulation. Assuming you followed the steps above you can run the code using

```
~/vampire/vampire-serial
```

Running the simulation will generate two new files: a log file with some details about the progress of the simulation and an output file which contains the final data. The simulation will take around 3 minutes to run. Have a break!

Step 6 | Plot the temperature-dependent magnetisation

The next step is to plot the data to see how the average magnetisation length varies with temperature. Gnuplot is a convenient software package for quickly plotting simulation data, as well as producing production quality figures. To use gnuplot to plot the data you need to start the program on the command line by typing

```
gnuplot
```

You can then plot the data using the command

```
plot "output" using 1:2 with points notitle
```

which will launch a new window plotting columns 1 and 2 together with points. You can add axis labels (with units) using

```
set xlabel "Temperature (K)"
set ylabel "Mean magnetisation (|m|)"
```

More information on plotting publication quality figures is available in the lecture notes.

Step 7 | Fit the data to obtain the Curie temperature

While you can estimate the Curie temperature by eye, it is much more helpful to fit a suitable equation to the data to extract the calculated Curie temperature. The temperature-dependent magnetisation is well-fitted by the expression

$$< |m| > (T) = \left(1 - \frac{T}{T_{\mathrm{C}}} \right)^{\beta}$$

where $< |m| >$ is the average magnetisation length, T is the absolute temperature, T_{C} is the Curie temperature and $\beta \approx 1/3$ is the magnetisation critical exponent. To fit this equation to our data in gnuplot we define a fitting function and initial values as follows:

```
m(T) = (1-T/Tc)**beta
Tc = 500.0
beta=0.3333
```

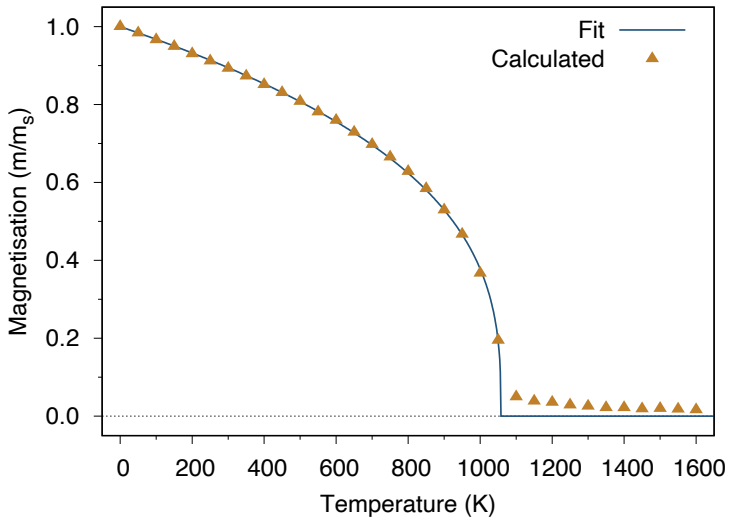
We can then fit this function to our data using the fit command in gnuplot

```
fit m(x) "output" u 1:2 via Tc, beta
```

which will attempt a best fit of the data. However, this fit is poor due to the critical nature of the data close to Tc, causing an average fit over the whole data range. To improve the fit we need a different function below and above T_{C}. Do do this in gnuplot we can use the ternary operator which has the form: condition ? value-if-true : value-if-false. Specifically in gnuplot we define a new fitting function

```
m(T) = T<Tc ? (1-T/Tc)**beta : 0.0
```

Refitting the data yields a much better bit to the data, with example data shown below.



Learning Questions

1) What value of the Curie temperature did you obtain by fitting?

2) How close is this value to the expected value for iron?

3) What physical property of magnets determines the Curie temperature?

4) What effect do you think turning off the periodic boundaries will have on the Curie temperature and why? See if you are correct by trying a simulation by commenting out the following lines in the input file

```
#create:periodic-boundaries-x
#create:periodic-boundaries-y
#create:periodic-boundaries-z
```

Homework

The data can be improved by running more simulation steps for each temperature, and also increasing the number of temperature steps.

How does your estimate of T_C improve with both of these values?

Which is more important - number of time steps or number of temperature steps?

You should also practice producing publication quality figures. Plot your data above with gnuplot but using the pdfcairo terminal to produce a PDF file with

```
set term pdfcairo
```

Ensure you set axis labels with units and select appropriate colours for the points and lines, not the default.

References

[1] Evans et al, Atomistic spin model simulations of magnetic nano materials
J. Phys.: Condens. Matter **26** 103202 (2014)
https://iopscience.iop.org/article/10.1088/0953-8984/26/10/103202/meta