Chapter 1

Chapter 5 - Problems

1.1 Problems

Problem 1

For a grain with uniaxial anisotropy in an external field, the direction of magnetization in this grain will be controlled entirely by the uniaxial anisotropy energy density ϵ_a and the magnetic interaction energy ϵ_m. The total energy can be written:

$$\epsilon_{tot} = \epsilon_a + \epsilon_m = K_u \sin^2\theta - \mu_o H M_s \cos(\phi - \theta),$$

where ϕ is the angle of the applied field relative to the easy axis of magnetization and θ is the angle of the moment relative to the easy direction. Show that the flipping field of a grain whose moment is initially antiparallel to the field, i.e. $\phi = 180^\circ$, is given by:

$$H_c = \frac{2K_u}{\mu_o M_s}.$$

Problem 2

In problem, we will begin to use REAL data. The data files used with this book are part of the **PmagPy** distribution, which you should have already downloaded and installed. [If not, follow the full installation instructions for **PmagPy** in the PmagPy Cookbook, and copy the *data_files* directory to a directory dedicated to these problem sets. Also, **PmagPy** is being constantly updated, so you should update your installation with the pip install method before attempting these problems.]

In this problem, you will become familiar with some of the **PmagPy** programs useful to the working paleomagnetist. These programs are designed to work with the Magnetics Information Consortium (MagIC) database (see `http://earthref.org/MAGIC`; see also PmagPy documentation) – a database designed for paleomagnetic and rock magnetic measurements. For now we are just using a few of the programs.

Someone has measured hysteresis loops on a mysterious set of specimens using an alternating gradient force magnetometer. These are contained in the *data_files/Essentials_Examples/Chapter_5* directory. For this problem, we will use the programs **pmag_gui.py** for converting measurement files to the MagIC format, and **hysteresis_magic.py** and **dayplot_magic.py** for visualizing and analyzing the data.

You should make a directory for every project you work on and that includes each homework problem. So create a new directory (say, Problem_5.2). From the command line navigate to your directory and , type **pmag_gui.py**. [**PC users may need to omit the .py termination for all PmagPy programs.**]

Under the 'Import' file menu, choose 'Hysteresis files' and 'Import entire directory'. The click on the 'add' button and select the Chapter_5 folder. There are two characters that distinguish specimens (things that are measured) from samples (things that are taken in the field. The location name can be 'Problem_5.2'. If you look in your project folder (here assumed Problem_5.2), you will find a bunch of new files named IS06a-1.magic and so on. Now click on the 'convert magnetometer files to MagIC format' button on the Pmag GUI main panel and click on 'Skip to next step' button. This will find all the files with the .magic termination. Click on 'OK' to combine these all into a single measurements file for use with **hysteresis_magic.py**. Keep clicking on the 'OK' buttons until the GUI is satisfied. Now exit the GUI and return to your notebook.

a) Run the **hysteresis_magic.py** program from within your notebook using the ! symbol before the command. Start by getting the help message:

```
!hysteresis_magic.py -h
```

Then run the program using these options:

```
!hysteresis_magic.py  -f measurements.txt -sav -fmt png
```

Load the images for the specimen IS06a-2 into the notebook as shown in the template file for this chapter's problems (found in data_files/Essentials_Examples/Notebooks directory you installed before.

Write a detailed figure caption for the three figures for specimen IS06a-2. What is the difference between the red and blue lines? What are the blue squares? What is the "DeltaM" curve?

b) Analyzing all the data in the last problem with **hysteresis_magic** created a file that stored the hysteresis parameters like saturation remanence, coercivity, etc. in a file called *sites.txt* in the project directory. Now run 'dayplot_magic.py' from within your notebook and load up the figures. Write a caption for these plots. How would you interpret them?