ASSIGNMENT 1: GRAPH PLOTTING

E. ARNOLD, M-S. LIU, R. PRABHU & C.S. RICHARDS
THE UNIVERSITY OF CAMBRIDGE

Written in XeLaTeX

CODING CONCEPTS

INTRODUCTION

In this tutorial we will introduce some useful MATLAB functions for directory organisation, and we will plot a close-packed plane.

GOOD CODING STYLE

In general, any written code should be easily understandable. If one does not adhere to writing code in good style, it is effectively useless to anyone else; this prevents them from using it for constructive purposes, regardless of how specialised their knowledge base is. Try to focus on the clarity of the code - instead of computation speed, attempt to make the code natural, and easy to follow. Do not worry about speed at this stage! It will always be the case that data handling scripts can be made faster, even after several versions have been created. Optimisation is a task we will eventually opt to achieve. However, we recommend that this is pursued as a secondary task, once a more basic working solution has been created. Optimising a script is much easier once there is something to base it on.

What can we consider "good coding style"? Some introductory comments on this matter include:

- Use meaningful and descriptive variable names. Do not use meaningless single character variable names (as physicists are notorious for doing), unless it is obvious, throughout the program, what the variable refers to. Place a comment on the line above the definition of any variable: allowing yourself, and others, to read the code with ease.
- Include comments that explain the logic and goal of each section of code. For example, when defining a function, describe what the function returns, and what input it takes.
- Include comments to justify the calculations used. This will make everyone happier.
- Write functions to represent sections of code used more than once, rather than repeating the code on numerous occassions. This will make the code easier to follow.

In the next section we will demonstrate some built-in MATLAB functions. Thse can be used to navigate file paths, and to call user-defined functions.

CREATING FOLDERS

When working with MATLAB, you may wish to "call" a function or a script which is not part of the default MATLAB installation. What does this mean? It provides us with a means to avoid the need to rewrite an existing section of code - we can save it as a function. For most practical purposes, it is useful to save an existing function in the same directory as the program, and then to use an in-line command to run the function. How do we specify what file to specify a function from? If it is stored in the same directory as the program being written, the task is simple: in the relevant block of code, it is only necessary to include the file name. Otherwise, should the function not be stored in the same directory, MATLAB must become aware of where it is stored. How do we add these directories to MATLAB's list of folders? The command "addpath" is used.

Example

The following general script checks whether a directory exists. If it does not exist, then the directory is created.

```
%'scripts_path' is a variable, storing the path as a string
%~ is the logical negation operator
%'mkdir' creates folder using the input string as the path
%'addpath' adds the path to the list that matlab searches
%'dir' tells the programme that the path points to a folder
if ~exist(scripts_path,'dir')
    mkdir(scripts_path);
end
addpath(script_path);
```

There are many more system functions available in the native MatLab installation. Try using the library documentation to explore some of them.

PROBLEMS

1. Write a script, similar to the one above, creating a folder containing a file named "auxCls.m".

PLOTTING LATTICES

PREREQUISITES

The chapter on *Crystal Structures* from the *Theory Handbook*.

INTRODUCTION

In our study of the structures of crystals, we will usually encounter the so-called "lattice points" which describe the positions of atoms. These are abstract entities, and require thought; plotting them is impractical, as they only consist of a single set of coordinates. In this tutorial, we will explore ways of making plots that act as useful graphics for the positions of atoms in a crystal lattice.

MATLAB'S MESHGRID

Later in this tutorial, we will plot lattice points. In order to plot these points, we must first generate the positions of the lattice points. How do we do this? Part of the task requires using the function `meshgrid`. This is a matrix manipulation function, taking two vectors $\mathbf{x}$ and $\mathbf{y}$, and returning two matrices $\mathbf{X}$ and $\mathbf{Y}$. These are given by:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x} & \rightarrow \\ \mathbf{x} & \rightarrow \\ \vdots & \vdots \\ \mathbf{x} & \rightarrow \end{pmatrix};$$

$$\mathbf{Y} = \begin{pmatrix} \mathbf{y} & \mathbf{y} & \cdots & \mathbf{y} \\ \downarrow & \downarrow & \cdots & \downarrow \end{pmatrix},$$

which can be interpreted with ease. The number of rows in one matrix is equal to the number of rows in the other. In addition, each row in $\mathbf{X}$ is the vector $\mathbf{x}$, and each column in $\mathbf{Y}$ is the vector $\mathbf{y}$. From the equations above, it should be clear how each is constructed.

ADSORPTION SITES

When an atom adsorbs onto a surface, it can do so at many different sites. These can be easily summarised:

- Top sites: these are the positions directly above the atoms on the top layer of atoms. While they are not marked on the diagram, they would fall exactly above any of the atoms on the top layer.
- Bridge sites: these are the positions above the centre of the lines connecting two adjacent lattice points.
- Substitutional sites: when an atom is absent from the periodic arrangement of the surface layer, an adsorbate may take its place. These would be located in substitutional sites.
- Hollow sites: ordinarily, there is a periodic stacking arrangement of close-packed planes in a crystal structure. This stacking sequence is different for hexagonal close-packed (HCP) and face-centred cubic (FCC), which leads to their distinct bulk structures. Whilst FCC can be described with the stacking sequence "ABCABC", HCP is the more simple "ABAB" (the letters used here have no special meaning). Do not overthink this! It states that one stacking sequence is periodic with period 2, and the other has period 3. Consequently, the positions of atoms in the plane above the upmost plane, known as hollow sites, are in different positions for the two types of lattice. In both cases, hollow sites sit at the intersticial site where an atom in the next plane would be.

Top sites can be considered as the local maxima of potential well, while the hollow sites are the minima, and bridge sites the saddle points. This will be useful for future tutorials.

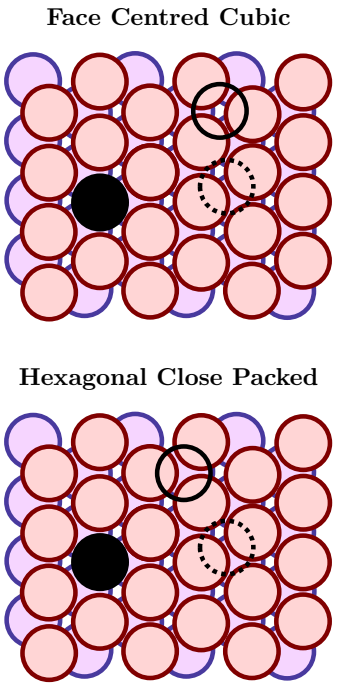

Figure 1 | The types of sites in FCC and HCP. Due to the different stacking sequence in the solids, the hollow sites are different for the type of lattice. The solid unfilled circles represent hollow sites. The dashed unfilled circles represent bridge sites. The filled circles represent substitutional sites.

AZIMUTHS

In this tutorial, we will only be considering the so-called "close-packed planes" of a crystal structure. In a close-packed plane, all adjacent atoms in a plane appear to touch each other, forming a hexagonal pattern. For convenience, it is very easy to label basis vectors in a plane: these are $\langle 01 \rangle$ and $\langle 10 \rangle$. These can be assigned arbitrarily to any of the axes in figure 2.

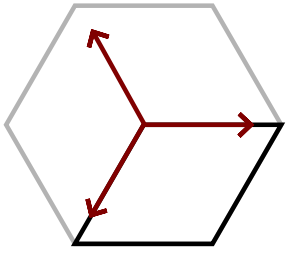

Figure 2 | Potential axes for the indexing of a close-packed surface. The two basis vectors can be assigned to any two of the above, as they form a complete basis.

TASKS

1. Derive the basis vectors for a close-packed plane. Give these in terms of the cartesian basis vectors.
2. Write a function that creates the basis vectors for a close packed plane.
3. Write a function that generates the reciprocal basis vectors for the

real space basis above.

4. Write a function that finds all lattice points in a certain interval (*e.g.* finding all lattice points between -10 and 10, or -5 and 5).
5. Consider the Ru(0001) surface. This is a close-packed plane. Draw a circle on each lattice point, and then draw the unit cell.
6. Change the colours of the circles. First plot them in black, then blue, then red. This is aimed to demonstrate how plotting structures with more than one type of atom can be easily done.
7. Plot triangles instead of circles. Then switch back to circles.
8. Derive the positions of the top sites, bridge sites and hollow sites on a face-centred cubic lattice. Plot each of these, on the same graph, using a different shape.
9. Attach a legend to your graph. Indicate the direction of the basis vectors on your graph.

Tips

1. For task 2, try using the "`meshgrid`" function to speed up the calculations. This is given in the solutions to this task.
2. For task 5, use the functions that have been previously written. Do not write them again!
3. For task 5, the figure 3 is a comic resembling what your output may look like.

SUMMARY

In this tutorial, we have explored how to visually represent the close-packed planes from two lattice types. This skill of visualisation will be useful throughout your further studies.

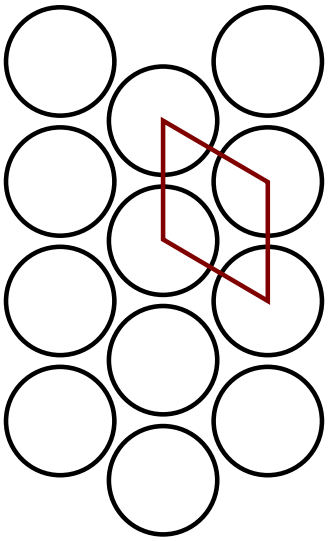

Figure 3 | Ru(0001) lattice structure. The unit cell is drawn on.