Documentation for the `cmp.py` script

1 Introduction

The program in this project is written in the open source programming language Python, using the standard library, along with the NumPy and Matplotlib packages, with PyQt5 being used for the GUI. The versions used in development was

- Python v3.6.5
- NumPy v1.14.3
- Matplotlib v2.2.2
- PyQt 5.9.2

The program should work with newer versions, though maybe not if they are major overhauls (say going from Numpy 1.x.y to 2.x.y). You can find instructions on how to install the different packages on their respective websites.

It is however recommended that you install Anaconda, which is a Python distribution that also includes both NumPy, Matplotlib and PyQt5, along with a wealth of other packages useful for scientific computing. The Jupyter package, especially, is useful, as it allows the user to write a bunch of small scripts (called cells) in a single file (called a notebook), and run the cells individually and quickly, while sharing the workspace between cells (i.e., variables persist between cells). A quick intro to Jupyter notebooks can be found at this link.

2 Installation

As recommended we will install the Anaconda distribution. Download the software from here: `https://www.anaconda.com/download/`. Install the software, and when asked if you want to include Anaconda to the path (or something along those lines) say: "Yes". This will allow you to start Python from your command line/terminal with the command `python`.

When Anaconda has been installed, you need the files from the project, these can be found on GitHub, at `https://github.com/NikolaiNielsen/Bachelor` (Or a direct download link).

Extract the code from the archive. The files `cmp.py`, `lattices.py`, `scattering.py`, `band_structure.py` and `gui.py` are all necessary, while `test.ipynb` includes some code examples.

To run the code you can do one of a couple of things:

- Start Jupyter notebook (run the command `jupyter notebook` from the command line/terminal), navigate to the directory where the code is located, start the test notebook (`test.ipynb`) and write your code there.
- Navigate, using your command line/terminal, to the directory where the code is located and run the command `python`. This creates an interactive python session where you can type in each command manually.

The reason why you need to navigate to the folder containing the code is because otherwise Python will not know where to look for the package, when you run the first line in the usage section. Python first checks the current working directory and next any other folders added to `$PYTHONPATH`. If your current working directory is the directory containing the code, then python can correctly import the package.