WARNING: If you have homebrew installed, then you may have the GNU compilers already installed (via brew install gcc) Yet, I found that the libgfortran.a static library was not configured properly (it appeared to be 32bit vs. 64bit). Continuing will install GNU compilers from source, thus you may overwrite the homebrew GNU compilers in /usr/local/ which could lead to problems with any software that has this dependency.

1. Test for some needed commands:
   1. make
   2. git
   3. wget
   4. pip

If you do not have them, don’t worry. You can use the following:

sudo easy\_install <software>

If that does not work, homebrew is another alternative package manger.

1. We need to install some prelimary/dependency software. I typically make a software directory in my home directory to do installs by hand. You can do the same using the following (we first go to the home directory (~), make the software directory, and then change to the software directory) :

cd ~

mkdir software

cd software

1. Install GNU compilers:
   1. Go to <http://hpc.sourceforge.net>
   2. Get the tar archives of gcc and gfortran for you Mac OS:

wget http://prdownloads.sourceforge.net/hpc/gcc-6.3-bin.tar.gz

wget http://prdownloads.sourceforge.net/hpc/gfortran-6.3-bin.tar.gz

* 1. Unpack in root directory for both compilers:

sudo tar –xvzf gcc-6.3-bin.tar.gz –C /

sudo tar –xvzf gfortran-6.3-bin.tar.gz –C /

1. Install dependencies for Quantum Espresso:
2. OpenMPI – An open-source Message-Passing Interface Library. “The way” to allow multiple processor nodes to talk to each other and what is the “glue” all supercomputers. We need to get the latest version from [https:/www.open-mpi.org](https://www.open-mpi.org/) (at time of this document, it is openmpi-2.1.0):
   1. cd ~/software
   2. wget https://www.open-mpi.org/software/ompi/v2.1/downloads/openmpi-2.1.0.tar.gz
   3. tar –xvzf openmpi-2.1.0.tar.gz
   4. cd openmpi-2.1.0
   5. ./configure --prefix=/usr/local
   6. We can compile in parallel using the –j flag with the number of cores to use. I typically use 4:

make –j <# of cpus> all

Go grab a coffee, this takes a while….

* 1. sudo make install

That should be it for installing OpenMPI. Check the install by executing:

mpirun

This should give you something along the lines of:  
--------------------------------------------------------------------------

mpirun could not find anything to do.

It is possible that you forgot to specify how many processes to run

via the "-np" argument.

--------------------------------------------------------------------------

1. FFTW – Fast Fourier Transform library. An optimized library for doing the reciprocal space to real space conversions. Go to <http://www.fftw.org> and get the latest version from (at time of this document, it is fftw-3.3.6-pl2). Similar instructions to installing OpenMPI:
   1. cd ~/software
   2. wget <http://www.fftw.org/fftw-3.3.6-pl2.tar.gz>
   3. tar –xvzf fftw-3.3.6-pl2.tar.gz
   4. cd ffw-3.3.6-pl2
   5. ./configure –prefix=/usr/local
   6. make –j 4
   7. sudo make install
2. LAPACK and BLAS – Optimized pair of linear algebra libraries (LAPACK is built on top of BLAS) used by many computational packages, including many DFT and classical MD packages. We will use the GitHub repository to install the latest version of both packages.:
   1. cd ~/software
   2. git clone <https://github.com/Reference-LAPACK/lapack.git>
   3. cd lapack
   4. make

Time for a 2nd coffee… it conveniently runs the test suite after compiling the packages. This will be the signal for completion. You should get the errors close to 0.00%.

These are the three dependencies we require. The OpenMPI, FFTW, and BLAS+LAPACK packages should be installed in a location that Quantum Espresso can find them during the configuration based on these instructions.

Now, go back to our software directory:

cd ~/software

or cd ..

1. Get latest version of Quantum Espresso. You can go to <http://qe-forge.org/gf/project/q-e/> to see the project. Select the Downloads link on the page to get to the different releases. Using wget you can directly download the 6.1 version tar archive via:

wget http://qe-forge.org/gf/download/frsrelease/240/1075/qe-6.1.tar.gz

1. Install Quantum Espresso:
2. Uncompress the tar archive:

tar –xvzf qe-6.1.tar.gz

1. cd qe-6.1
2. ./configure LIBDIRS="-L/usr/local/lib"
3. To ensure we have linked to the OpenMPI, FFTW and BLAS+LAPACK libraries, we should see something along the following at the end of the ./configure output:

*The following libraries have been found:*

*BLAS\_LIBS= -lblas*

*LAPACK\_LIBS= -llapack -lblas*

*FFT\_LIBS= -lfftw3*

1. Now we should be able to compile Quantum Espresso. We can compile in parallel using the –j flag with the number of cores to use. I typically use 4:

make –j <# of cpus> all

1. Test Quantum Espresso
   1. You can also download the **test-suite** and the **examples**  for Quantum Espresso from the previous download page found via <http://qe-forge.org/gf/project/q-e/>. This is to make sure your installation is working properly and giving accurate results. For the test-suite, go to the directory where you have installed Quantum Espresso. This is where you must run the tests. For version 6.1:
      1. cd ~/software/qe-6.1
      2. wget <http://qe-forge.org/gf/download/frsrelease/240/1073/qe-6.1-test-suite.tar.gz>
      3. tar –xvzf qe-6.1-test-suite.tar.gz
      4. cd test-suite
   2. From here, you can run many different test suites via make. We begin with the serial tests:
      1. make run-tests

Grab lunch while this runs…

* 1. Now, run the parallel tests:
     1. make run-parallel

Run this while you write a rough draft of a paper, prepare a sample, or leave for the end of the day.

NOTE: I did get some FAILED test with the following:

*“Different sets of data extracted from benchmark and test. “*

Nothing to be alarmed about. Looking into it, it appears it is due to a data structure that does not exists for comparison. I believe it is an input/output problem such that data was not saved when it should have been. Thus, it is not in the actual calculation so I ignore these failures for now until I find something to sway me otherwise.

I also had the following for FAILED tests:

*“ERROR: absolute error 4.00e-05 greater than 1.00e-06. (Test: -328.23191. Benchmark: -328.23187.)”*

Yet, you can see the reported error is still within reason. Note the test and if you use something from what this test is reporting, realize you lose accuracy to witht the absolute error reported. (This was for pw\_pawatom, which is a test of the Projector Augmented Wave method for addressing core electrons in a computationally efficient manner).

1. Make it available system-wide
   1. Lastly, you need to add the Quantum Espresso executables to your PATH variable. You can do so by adding them to your bashrc file.
      1. cd ~
      2. Edit the .bashrc file (or make one) and add these lines:

# Quantum ESPRESSo

export QE\_PATH="$HOME/software/qe-6.1"

export PATH="$PATH:$QE\_PATH/bin"

* 1. You can test by issuing the following from anywhere on the command line:
     1. pw.x

If successful, it will start quantum espresso and wait for input file. Kill with Ctrl+C.

Next, installing ASE, adding links for Quantum Espresso, and being able to build systems and run DFT via Quantum Espresso in Python under ASE!