

A quick introduction to Quantum Espresso

Simone di Cataldo

September 27, 2019

Contents

1	Preamble	3
2	Density Functional Theory	3
2.1	What can we calculate with DFT?	5
2.2	What can't we calculate with DFT?	5
3	The Quantum Espresso distribution	6
3.1	How to install	6
3.2	Packages	7
4	Running a DFT calculation	8
4.1	Understanding the calculation	8
4.1.1	Energy Cutoff	9
4.1.2	Pseudopotential	10
4.1.3	Brillouin zone sampling	10
4.1.4	Phonon calculations	10
5	Input files, explained	11
5.1	The must-do calculation (ground state charge density)	11
5.1.1	Example, method 1	12
5.1.2	Example, method 2	13
5.2	Band structure	15
5.2.1	Example, part 1	16
5.2.2	Example, part 2	17
5.3	Density of states	18
5.3.1	Example, part 1	19
5.3.2	Example, part 2	20
5.4	Phonons	21
5.4.1	Example, part 1 - calculate dynamical matrices	22
5.4.2	Example, part 2 - calculate interatomic force constant	22
5.4.3	Example, part 3 - calculate the phonon density of states	22
5.4.4	Example, part 4 - calculate the phonon dispersions	22

1 Preamble

This short introduction is meant to give other students a basic understanding in Density Functional Theory (DFT) and Quantum Espresso (QE). Although there are a lot of resources on DFT online, these are often either too superficial, or too in depth. Here we will try to find a compromise between the two, and provide guidelines to get a calculation done with some understanding of what is happening inside the "DFT black box". It would be impossible to give an exhaustive explanation, for which I'll recommend references in the text.

I strongly recommend to anyone interested in this topic the introductory lecture of prof. Paolo Giannozzi (one of the fathers of Quantum Espresso), which you can find on YouTube (https://youtu.be/1AH2pki_jDPg) [8]. It lasts about one hour and gets you through the theory very quickly. It covers many important aspects, as good as it can be done in one hour. You may also be interested in the second video of the channel (Density Functional Perturbation Theory, used for phonon calculations) and in the third video (specific implementation of DFT in Quantum Espresso)

An in-depth treatment of basic DFT is also given in *A Primer in Density Functional Theory* [7]. It is way too long for a simple project, but it can be a useful reference. The two founding papers for DFT (Hohenberg and Kohn theorem, and Kohn and Sham theorem) are also listed in bibliography [10, 11].

2 Density Functional Theory

Density Functional Theory (DFT) is by far the most popular quantum mechanical modeling method for solid state physics. The main reason is its quite good accuracy compared to its computational cost. As suggested by its name, in DFT all the relevant physical quantities can be expressed as a *functional* of the electron *density*.

To understand why this is an advantage, let's consider the electronic Hamiltonian in a solid, within the Born-Oppenheimer (BO) approximation. Since we only care about the electrons, we neglect the ion-ion interaction and the ionic kinetic energy.

$$H = - \sum_i \frac{\hbar^2 \nabla_i^2}{2m} - \sum_{iJ} \frac{Z_J e^2}{4\pi\epsilon_0 |\vec{r}_i - \vec{R}_J|} + \frac{1}{2} \sum_{i \neq j} \frac{e^2}{4\pi\epsilon_0 |\vec{r}_i - \vec{r}_j|} \quad (1)$$

where \vec{r} is the position of the electron and \vec{R} is the position of the ion. The three terms on the right hand side of the equations are the electronic kinetic energy, the ion-electron interaction and the electron-electron interaction, respectively. As it will occur later, it is useful to mention now that we can think of the ion-electron interaction as if the fixed ions generate an external potential, which acts on the electrons.

If we want to describe our N-electron system then we have to solve the Schrödinger's equation

$$H\Psi = E\Psi \quad (2)$$

where Ψ is the wavefunction of the N-electron system, and E is its total energy. Now, Ψ has to describe N electrons, with $N \approx 10^{23}$, and each electron has 3 degrees of freedom, so Ψ depends on $3N$ variables. Instead if we can express the total energy E as a functional of the electron density $n(\vec{r})$, we have $E = E[n(\vec{r})]$, where n is a function of only three variables. The first Hohenberg and Kohn theorem states that, in fact, this is possible [10]. From an analytical perspective the problem is not simpler, because the explicit functional dependence of E from n is not known, but from a numerical point of view, things are incredibly simpler.

In particular, the second Hohenberg and Kohn theorem states that *for a constant number of electrons N, the electron density $n(\vec{r})$ which minimizes $E[n]$ has to be the ground state electron density of the N-electron system* [10]. This implies that we can start from an arbitrary $n_0(\vec{r})$ and follow a minimization procedure until we find the minimum of $E[n]$. When E is minimized, n will be the correct ground-state electron density ¹.

So far we realized that it is possible to express the total energy of the system as a functional of the electron density, and a variational principle guarantees that we can find a (unique) solution for the energy of the system, but we have no idea on how this could be done. The next step is to observe that the Hohenberg and Kohn theorems do not have any requirements on the Hamiltonian of the system, i.e. they are also valid for an Hamiltonian without electron-electron interaction. This property is exploited by Kohn and Sham, which arrive to the following conclusion: *it is possible to represent the ground state density $n(\vec{r})$ of the interacting N-electron system (with Hamiltonian (1)) as the ground state density of a non-interacting system of N electrons in a fictitious external potential, called self-consistent potential (v_{scf}), constructed in such a way that the correct ground state density is reproduced* [11]. This conclusion represents an immense advantage, because whenever electrons are non-interacting we can decouple the Schrödinger's equation in N independent equations (also called Kohn&Sham equations), solve them one by one to obtain the single particle orbitals (also called Kohn&Sham orbitals) and from those calculate the ground state charge density.

$$\left\{ -\frac{\hbar^2 \nabla^2}{2m} + v_{scf}(\vec{r}) \right\} \psi_i(\vec{r}) = \epsilon_i \psi_i(\vec{r}) \quad (3)$$

Once the ψ are known, I can construct the ground state electron density by summing over the square modulus of the non-interacting Kohn&Sham (KS) ψ , i.e. $n(\vec{r}) = \sum_{i=1}^N |\psi_i(\vec{r})|^2$. Note that n is the same for both the interacting and non-interacting system!

Within suitable approximations on the dependence of the total energy on the electron density (especially for the so-called exchange-correlation energy),

¹The first Hohenberg and Kohn theorem guarantees that the solution is unique

the form of $v_{scf}(\vec{r})$ can be constructed starting from the ionic positions and the electron density $n(\vec{r})$. Note that $n(\vec{r})$ is both a required ingredient to construct the Hamiltonian, and one of the results we are interested into! The solution has to be reached using a *self-consistent* algorithm: we start from a trial density $n_0(\vec{r})$, construct $v_{scf}(\vec{r})$, solve (3), calculate $n_1(\vec{r})$ and so on, until our results do not change from one cycle to another (i.e. we achieved "self consistency"). The self consistent (SCF) loop is the central loop in any DFT code.

2.1 What can we calculate with DFT?

DFT can be used to calculate the *ground state* energy of a system, and the eigenvalues ϵ_i of the single-particle equation give the dispersion relationship $\epsilon(\vec{k})$, as will be clarified later.²

Moreover, a perturbative approach to DFT (called Density Functional Perturbation Theory) can be used to calculate the n-th order derivatives of the total energy with respect to a perturbation, which gives quantities such as the forces acting on each atom, pressure, phonon frequencies, and so on. It makes use of the Hellman-Feynman theorem for first derivatives and its extension, the 2n+1 theorem, for higher order derivatives.

In short, we can reliably calculate:

- Total energy of a given structure
- Band structure, density of states
- Forces acting on each atom, stress tensor
- Elastic properties
- Phonon frequencies (within the harmonic approximation)
- Electric polarizability, Raman and Infrared activity
- Electron-phonon coupling, superconducting T_c

2.2 What can't we calculate with DFT?

DFT is a ground-state theory, therefore it fails to calculate time-dependent properties, such as excited states or response to time-dependent fields. There is a complicated extension to the theory which does that (Time-dependent Density Functional Theory).

It is very difficult to describe *localized* states within DFT. Atoms with localized *d* or *f* orbitals such as some transition metals, actinides, and lanthanides, can be problematic. Calculations with these elements should be carried out carefully, and results should be examined critically.

²From a formal point of view there is no direct correspondence between the eigenvalues of the single particle equations and the eigenvalues of the full, interacting equation. However there is very often a good correspondence, and DFT bands are nowadays accepted as a good estimate of the real bands.

3 The Quantum Espresso distribution

Quantum Espresso is *an integrated suite of Open-Source computer codes for electronic-structure calculations and materials modeling at the nanoscale. It is based on density-functional theory, plane waves, and pseudopotentials.* [2].

We will worry about plane waves and pseudopotentials in section 4.1. For now it is enough to say that Quantum Espresso is an ensemble of programs which can be used to calculate the properties listed in 2.1. The installation can be a bit tricky if you are not familiar with bash. The following instructions for installation will work for most Linux distributions and Mac OSX. For the latter you first need to install the command line tools (go to <https://developer.apple.com/downloads/index.action>, register and download command line tools).

3.1 How to install

1. Go to <https://gitlab.com/QEF/q-e/tags>
2. Download the latest version by clicking on the cloud with a down arrow (.zip or .tar format)
3. Unpack the downloaded folder
4. With the terminal, change directory into the newly created folder (e.g. `cd /home/Downloads/q-e-qe-X.X.X/`)
5. I recommend installing the parallel environment OpenMPI. Although this is not strictly necessary, it allows you to use multiple processors to speed up the calculations.
6. Execute "configure" script by typing `./configure`. This does most of the work for you, as it checks whether you have the necessary libraries.
7. If a library or a compiler is missing, it will give an error
8. Copy the name of the missing library on google and find a package which contains it (Stack Exchange is your friend)
9. Linux systems: type `sudo apt-get update`, then `sudo apt-get upgrade`, then `sudo apt-get install NAME_OF_PACKAGE`
10. Max OSX systems: you may need to install macports or homebrew, and then do `sudo port install NAME_OF_PACKAGE` or `brew install NAME_OF_PACKAGE`
11. Once all the necessary libraries are there, the script will finish with success
12. Execute the command `make all`
13. Again, if you do not have "make" installed, you can install it via apt-get (Linux) or homebrew (OSX)
14. The code will be compiled, and executables copied into the `q-e-qe-X.X.X/bin/` folder. In order to get access to the executables from anywhere, you have to add them to the `$PATH` variable

15. Open the file ".bashrc" or ".bash_profile" in your home folder with any text editor. This is a hidden file, so you need to open it from the terminal, typing "vim /home/.bashrc" or "emacs /home/.bashrc"
16. Add the line "export PATH=\$PATH:/path/to/qe-folder/bin/" (e.g. export PATH=\$PATH:/home/Downloads/q-e-qe-X.X.X/bin) to your .bashrc or .bash_profile folder
17. Close and re-open the terminal (it will force it to load the new settings)
18. Try to type "pw.x" and press enter. If you did everything correctly, the program pw.x should start (you can kill it with ctrl+c)

3.2 Packages

There are a lot of executables inside the q-e-qe-X.X.X/bin/, here's a short (not exhaustive) list, with a short description. More details on how to use each code are described in section 4.

- pw.x
- It is by far the main and most important: it executes the self consistent calculation to find the ground state charge density. It can also minimize forces to find the optimal structure from a given set of atomic positions. The results of this calculation are the starting point for any other calculation.
- dos.x
- A small script which sums over the eigenvalues obtained by pw.x to get the density of states
- bands.x
- Another small script which collects the output of a special pw.x calculation, and produces a data file to plot the bands
- ph.x
- A code which uses Density Functional Perturbation Theory to calculate the dynamical matrix at different \vec{q} points in the Brillouin zone (BZ). The dynamical matrices are used by other executables to calculate the phonon dispersions. It can also calculate the electron-phonon coupling.
- q2r.x
- A small script which transforms the dynamical matrices (reciprocal space) to the interatomic force constant matrices (real space). It is an intermediate step to calculate the phonon dispersion relationship
- matdyn.x
- A small script which takes the interatomic force constant matrix and interpolates them on a given path in the BZ. Used to calculate the phonon dispersions.

- `pp.x`
- A post processing tool to plot various observables (charge density, wave functions, self-consistent potential and so on)
- `plotband.x`
- A useful tool to collect the result of a band structure calculation and transform it into a gnuplot-readable file

I also recommend the installation of a visualization software. Xcrysden is the most integrated with Quantum Espresso, and can be installed by just typing `"sudo apt-get install xcrysden"` (Linux) or `"sudo port install xcrysden"` (Mac OSX, once macports is installed).

To visualize the structure open an output file of QE by typing `"xcrysden -pwo output_file.out"`

Another very nice visualization software is VESTA (<http://jp-minerals.org/vesta/en/>). It is less integrated with Quantum Espresso, but it has much better graphical options to make good-looking structures.

4 Running a DFT calculation

Running a calculation requires an input file for each package, containing information on the configuration of ions (unit cell, atomic positions) plus some other numerical parameters, and a pseudopotential file for each element in the input file. I recommend reading the next section ("Understanding the calculation") to get a feeling of what each of these parameters mean. If you don't care and you just want to crunch some numbers, I have provided an example for each calculation. The examples show calculations for the cubic phase of copper.

Once the input file is ready, execute the command

```
"pw.x < input_file.in > output_file.out"
```

The following command reads the file `"input_file.in"` and redirects it to `pw.x` as if you were typing it in the terminal. Then, the standard output is redirected to a file called `"output_file.out"`, so that you can read it.

If you set up a multiprocessor environment you can type

```
"mpirun -np N pw.x input_file.in > output_file.out"
```

where `N` is the number of cores in your machine (typically 2 or 4).

4.1 Understanding the calculation

In section 2 we saw that with the Kohn and Sham picture we can represent our system of `N` interacting electrons in terms of `N` noninteracting electrons in an external, self consistent potential, and that this gives rise to a Schrödinger's equation that can be solved numerically. There are some extra details which are needed to understand all the parameters that one has to set up into the input file.

First and foremost, we consider all of our DFT calculations to be done in a solid, i.e. a system with discrete translational invariance. This implies that

we can use the Bloch theorem. The advantages of using the Bloch theorem are two:

1. The Hamiltonian is diagonal in \vec{k}
2. All the calculations can be restricted to a single unit cell

Using Bloch theorem, equation 3 is expressed in the Bloch basis, and defines the dispersion relationship $\epsilon_{\vec{k}}$

$$\left(-\frac{\hbar^2}{2m}\left(\frac{1}{i}\Delta + \vec{k}\right)^2 + v_{scf}(\vec{r})\right) u_{\vec{k}}(\vec{r}) = \epsilon_{\vec{k}} u_{\vec{k}}(\vec{r}) \quad (4)$$

where \vec{k} is a wave vector belonging to the first Brillouin zone and v_{scf} and $u_{\vec{k}}(\vec{r})$ are lattice-periodic functions. This equation should look familiar. If it is not, I recommend you to read chapter 8 of Ashcroft and Mermin's "Solid State Physics" [4]. This equation is saying that not only we can solve a single particle equation, we can also solve it for different values of \vec{k} , as the Hamiltonian is already diagonal in \vec{k} . For each given \vec{k} we will get a set of $\epsilon_i(\vec{k})$, which is our dispersion relationship.

4.1.1 Energy Cutoff

Equation 4 is still a second order differential equation, which needs to be solved numerically. The easiest way to do it is the following: instead of directly solving the solution, we can expand the wave function $u_{\vec{k}}(\vec{r})$ as a sum of basis set functions, weighed with a coefficient. Then the correct solution is found if one finds the right coefficients, i.e. if one solves a linear system, which is much easier to do in a machine. In principle one has to consider an *infinite* sum of basis set functions. In practice, one has to *truncate* the expansion, including enough basis functions to describe all the physics correctly. In Quantum Espresso the expansion is done in terms of *plane waves*, which have several advantages.

In the code, the number of plane waves included in the calculation is expressed as a *cutoff energy* (ecutwfc in the input file). The higher the cutoff energy, the more plane waves (but also the slower the calculation). The optimal cutoff mostly depends on the element and the size of the cell, and little on the chemical environment. In order to check which cutoff is right for your system, you should run a calculation with different values of the energy cutoff, until the total energy *converges*, i.e. the difference between the total energy, calculated with two different cutoff energies is smaller than the required accuracy (typically 1 meV/atom) The total energy cutoff depends on the elements included in your calculations, and it is bound by the element which requires the highest cutoff. With a cutoff of 60-80 Ry you should be safe for all elements, but I do recommend checking for convergence, as with some elements need a much lower cutoff.

4.1.2 Pseudopotential

The number of plane waves required can be reduced dramatically by the use of a *pseudopotential*. In short, a pseudopotential is a modification of the ionic potential which allows one to greatly reduce the number of plane waves needed, without changing the chemical properties of the atoms. There is an incredibly large variety of pseudopotentials, which mostly has to do with the method used to generate it. The potentials recommended on the website of QE are *usually* all reliable, but unless you know what you're doing *use them at your own risk*.

I recommend the *Optimized Norm Conserving Vanderbilt* (ONCV) potentials. You can find a database in the .UPF format (used by Quantum Espresso) at this address http://www.quantum-simulation.org/potentials/sg15_oncv/ with almost all elements. These potentials are rather costly (they require a high energy cutoff) but they are very robust.

4.1.3 Brillouin zone sampling

In the previous section we saw that equation 4 can be solved for a given \vec{k} . Many electronic and thermodynamic properties of the solid, however, depend on the dispersion relationship $\epsilon_{\vec{k}}$ over the whole Brillouin zone (i.e. for all possible values of \vec{k} in the first BZ). In principle there are infinitely many \vec{k} . In practice what the code does is sampling the first BZ (which is a 3D solid) with a discrete grid.

The number of points needed in the grid depends on several factors, and it is not obvious to decide how many to use. In particular, not many k-points are needed for the ground state charge density, but a lot are needed for the density of states. In the next sections we'll see some guidelines on a case-by-case basis.

In order to reduce the number of \vec{k} points needed, the code typically introduces a *smearing* on the occupations, such as the Fermi function at $T \neq 0$, or a Gaussian broadening. Imagine that for every sum involving δ functions, we sum over a Gaussian instead. This helps to greatly speed up convergence. Values between 0.01 and 0.08 Ry are reasonable for the self-consistent calculation. The larger the smearing, the less the \vec{k} points needed, and the faster the calculation. Values too large, however, may give wrong results, especially in semiconductors where the conduction band may become occupied. As a first guess, try with 0.05 Ry.

4.1.4 Phonon calculations

Phonon calculations are possible within an extension of DFT called Density Functional Perturbation Theory (DFPT). The process is, in principle, quite simple, and it was outlined during the course. First, the equilibrium positions of the atoms have to be calculated. Then, using the harmonic approximation, the linear spring constants are calculated. For an in-depth overview I recommend you ref. [5], for details on the calculations see section 5.4.

5 Input files, explained

Note: you can also find the input files here [3].

5.1 The must-do calculation (ground state charge density)

This is the first step for any calculation you want to do. You need to set the structure in your input file, then a few more parameters, and run the pw.x code. A complete reference of all the parameters for the input file can be found at https://www.quantum-espresso.org/Doc/INPUT_PW.html. Also have a look at the sample input files that follow each section **Note:** in the input files the exclamation mark is used for comments.

The most starting point is setting up the atomic positions correctly. Follow this workflow, and don't give up if you cannot make it at first try (it takes some practice).

1. Look up the structure in a reference, you will need the space group and the Wyckoff positions. The ".cif" files also contain this information.
2. Look up in Wikipedia which Bravais lattice corresponds to your space group (https://en.wikipedia.org/wiki/List_of_space_groups). For instance, space group 71 (or Immm) is a body-centered orthorhombic. Note: for our purpose it is important to check whether it is simple, face-centered, etc.
3. Look up in the input guide (https://www.quantum-espresso.org/Doc/INPUT_PW.html) at the keyword "ibrav" which number corresponds to your lattice type
4. The "ibrav" instructions also contains useful information on which cell parameters you will need to input (e.g. for a cubic you only need to input one lattice parameter, for a tetragonal you need to input two, for a monoclinic you need three lattice parameters and an angle, and so on)
5. Provide the cell parameters in the input file. The three axes are called a , b and c , and the angles between them α , β and γ
6. **Note:** The format might be not intuitive. The first cell parameter (celldm(1)) has to be in Bohr. Then celldm(2) and celldm(3), if needed, have to be expressed as the ratio b/a and c/a , respectively. Finally, celldm(4), celldm(5) and celldm(6), if needed, are the cosine of the angles α , β and γ
7. Add the atomic coordinates (see below)

There are mostly two methods to provide the atomic coordinates. Usually in literature only the *inequivalent* positions of atoms are given, and the full structure has to be reconstructed by symmetry operations (or a program that does it for you). The same is true also for crystallographic information files (.cif). The first method is to directly add the space group in the input file ("space_group = ###"), and then only give the inequivalent positions for the atoms, with their Wyckoff position. The second method is to use a software

like VESTA to reconstruct the cell, and then export it in a format where all the coordinates are explicitly given (such as ".vasp" extension), and the cell is described by the three lattice vectors. With symmetric structures the first method is usually very easy and fast, while the second is more automatic for structures with low symmetry.

Note: in the input file you have to specify the "number of atoms" in the cell. Actually, when you use the first method you need to specify only the number of atoms given in input (which is going to be less, as the others are generated by symmetry!)

In the output file of the self consistent calculation, an estimate for the Fermi energy will be given. Look into the output file (or into the standard output) for the keyword "Fermi" (capital F is important). Alternatively, if you redirected the standard output to a file, you can type from the terminal "grep 'Fermi' output_file_name". Note that this number makes sense only in a metal. In an insulator the code will still print a "Fermi energy", which has **no** meaning, not even the chemical potential.

5.1.1 Example, method 1

In a paper the information on the crystal structure would look like this ([9, 6])

Atom	Space group: $225 (Fm - 3m)$ Wyckoff position	Coordinates
Cu	4a	(0,0,0)

The position of the atoms, given this symmetry, do not require to specify any coordinates (there is no free parameter). This can be checked by visiting again the Bilbao Crystallographic Server at the subpage for Wyckoff positions (http://www.cryst.ehu.es/cryst/get_wp.html). The page requires the input of the space group, and returns all the Wyckoff positions for that space group, with the specification of which coordinates are fixed by symmetry, and which have to be given. See a sample input file in the next page.

5.1.2 Example, method 2

We can reconstruct the above crystal structure in a program like VESTA, and export it into the ".vasp" format, which looks as follows:

The first three vectors are the lattice vectors (in Å). Then, there is a line expressing to which element each line refers to, and last there are the atomic coordinates, in units of the lattice vectors. Note that in this case the number of atoms written in the input file reflects the number of atoms in the unit cell. The input file is below.

```

&control

calculation = 'scf'

outdir = './output'

pseudo_dir = '/home/simonedicaldo/QE-pseudopotentials/'
prefix = 'Cu'

tstress = .true.
tprnfor = .true.

forc_conv_thr = 1e-5
&end

&system
ibrav = 2

celldm(1) = 6.90
space_group = 225

nat = 1
ntyp = 1
! nbnd = 16

ecutwfc = 120

occupations = 'smearing'

smearing = 'mp'
degauss = 0.02
&end
&electrons
conv_thr = 1.D-8
&end
&ions

ion_dynamics = 'bfgs'
&end
&cell

cell_dynamics = 'bfgs'
press = 0

&end
ATOMIC_SPECIES
Cu 63.54 Cu_ONCV_PBE_sr.upf

ATOMIC_POSITIONS crystal_sg

Cu 4a

```

! Input file is divided in sections, dependent variables affected. This mostly affects file calculation type.

! Which type of calculation has to be performed (self consistent field)

! Path to the folder to store the output files. All calculations will have to read into this folder. Make sure that the path is correct

! It's a label for the calculation. Has to be unique when you read into the output folder in a new calculation.

! Calculates and prints the stress tensor (if .true.)

! Calculates and prints the forces acting on atoms (if .true.)

! Explained above. 6 is for simple tetragonal, 1 for fcc and so on

! Units for celldm(1) are Bohr (1 Bohr \approx 0.529 Å)

! Space group number. Needed in order to generate coordinates using Wyckoff positions

! Number of atoms in the atomic coordinates

! How many different atomic species there are

! You can change the number of bands in the calculation. Be careful, there have to be a few more than the number of electrons

! Very important! Cutoff energy on the wavefunction expansion.

! Using a smearing on the occupations so to converge with less k points

! Type of smearing

! Width of smearing (in Ry)

! Convergence threshold on scf calculation

! Ignored in scf calculations, only used in "vc-relax" calculations

! Ignored in scf calculations, only used in "vc-relax" calculations

! Pressure to be applied to the cell for hydrostatic calculations. It is in kilobar (1 kB = 0.1 GPa)

! The crystal_sg command is used to determine the atomic positions from Wyckoff positions

! Since the position of Cu is determined from the Wyckoff positions, no coordinates are needed

```

New structure
1.0
3.5999999046  0.0000000000  0.0000000000
0.0000000000  3.5999999046  0.0000000000
0.0000000000  0.0000000000  3.5999999046
Cu
4
Direct
0.0000000000  0.0000000000  0.0000000000
0.0000000000  0.5000000000  0.5000000000
0.5000000000  0.0000000000  0.5000000000
0.5000000000  0.5000000000  0.0000000000

```

5.2 Band structure

In order to calculate the bands in a way that can give a nice plot, one has to calculate the eigenvalues of the equation (4) for a specific set of values of \vec{k} . In particular, we want to construct a path in \vec{r} space which connects all the high symmetry points of the Brillouin zone, and then calculate $\epsilon_{\vec{k}}$ for this path. Note that this approach can **not** come out of a self-consistent calculation, because here we are only sampling a line into a 3D space. What we can do is

1. Perform a self-consistent calculation to get the correct ground state charge density and self-consistent potential (see 5.1)
2. Solve equation (4) with the potential obtained in the previous calculation on the desired path in \vec{k} space
3. This is done by using "pw.x", selecting "bands" as the type of calculation, and giving the path in the Brillouin zone as input
4. Collect the result of the previous calculation into a data file
5. Done by the simple script "bands.x". It produces a file with the extension ".gnu" which contains the linearized path (in units of $\frac{2\pi}{a}$) in \vec{k} space on the first column, and the corresponding energy (in eV) on the second column.

The high-symmetry points in the Brillouin zone depend on the space group of our system. Once the coordinates of these points are known, the path is automatically generated by QE by using a special option in the k-points setting (`crystal_b`, for crystal coordinates), which only requires *the coordinates* of the special points and *the number of intermediate points* between each special point. These coordinates can be obtained by visiting the Bilbao Crystallographic Server (http://www.cryst.ehu.es/cryst/get_kvec.html) and entering the space group number of our system. Here the coordinates and names of the high-symmetry points are given, as well as a 3D representation of the BZ. As a rule of thumb, you want to figure out a path which visit all the high-symmetry points via high-symmetry lines, without passing along the same line twice.

5.2.1 Example, part 1

First run a self-consistent calculation (see "The must-do calculation", section 5.1). Then run pw.x again with the "bands" option (solve Schrödinger's equation on a given path in reciprocal space). A sample input file is in the next page

5.2.2 Example, part 2

Then collect the result running the script "bands.x" The output, "Cu.bands.gnu" contains the plottable data!

5.3 Density of states

Mathematically the energy dependent density of states is obtained from the dispersion relationship $\epsilon(\vec{k})$ using the formula

$$dos(\epsilon) = \int \frac{d^3k}{(2\pi)^3} \delta(\epsilon - \epsilon(\vec{k})) \quad (5)$$

The above formula is mathematically sound, but not directly applicable in a computational algorithm. In fact, the sum on the left is obtained as an *infinite sum* of *infinitely narrow* functions (the δ functions). In practice one either substitutes the δ functions with a function of finite width (like a gaussian), or uses an interpolation method (the tetrahedron method). For the purpose of calculating the density of state, the tetrahedron method is by far more efficient, although it still needs a large number of \vec{k} points in order to give a smooth density of states.

There is a trick which allows one to use a very fine grid of \vec{k} points without too much computational effort. The trick exploits the fact that the self-consistent potential and the ground state charge density need a relatively small number of \vec{k} points to converge. It works as follows

1. Perform a self-consistent calculation to get the correct ground state charge density and self-consistent potential (see 5.1). Here you do not need a lot of \vec{k} points. As a result, you will obtain the correct self-consistent potential and charge density.
2. Perform a non-self-consistent calculation, which starts from the potential obtained in the previous step. Now you will use the tetrahedron method, and a large number of \vec{k} points. This can be done using the "pw.x" code, selecting a calculation of type "nscf", occupation "tetrahedra", and a fine \vec{k} grid
3. Collect the results using the "dos.x" script. It outputs a dos file, which is a three columns file containing the energy on the first column, the density of states (states/eV) on the second column, and the integral of the density of states on the third column.

The method discussed above is much faster, as equation (4) on the fine \vec{k} grid *only needs to be solved once!*

5.3.1 Example, part 1

First run a self-consistent calculation (see "The must-do calculation", section 5.1). Then, run pw.x again, with the option "nscf", using the tetrahedron method for occupations, and a much denser \vec{k} grid.

5.3.2 Example, part 2

Once the non self consistent calculation has finished, run the "dos.x" script to collect the results. See the sample input file in the next page. This will produce a plottable file called "Cu.dos.dat".

5.4 Phonons

The calculation of phonon dispersions requires several steps and, in general, it takes much longer than a self-consistent calculation. It is articulated in the following steps:

1. Check that the ions are in their equilibrium positions! Forces should be close to zero, stress tensor should be diagonal, and all diagonal components should be the same (homogeneous pressure). You may want to use the 'relax' option (see 5.1). Since we want to perturb the atomic positions w.r.t. equilibrium, we first want to make sure that the atomic positions are at equilibrium, i.e. each component of the force acting on atoms is of the order of (or less than) 1 meV/atom.
2. Calculate the ground state charge density with the equilibrium positions of the ions (see 5.1).
3. Calculate the dynamical matrices in reciprocal space on a grid (using ph.x).³ In this step the code performs a small displacement of the atoms, and calculates the resulting perturbation on the potential and on the charge density. Within linear response theory (see [5]) the dynamical matrices are calculated. The strength of QE lies in the fact that the dynamical matrices are calculated directly in reciprocal space, therefore there is no need of constructing a supercell.
4. Calculate the force constants from reciprocal to real space (using q2r.x), which simply means that a Fourier transform is performed, from the dynamical matrix (reciprocal space), to the force constant matrix (real space)
5. Interpolate on a dense grid in reciprocal space to calculate phonon density of states (using matdyn.x)
6. Interpolate on a line in reciprocal space to calculate phonon dispersion relationship (using matdyn.x)

The last two points are the so-called Fourier interpolation. Imagine we have a function $F(\vec{q})$ which is calculated in reciprocal space on a finite grid \vec{q}_i . We can calculate the inverse-Fourier transform $f(\vec{r}) = \frac{1}{(2\pi)^3} \sum_{i=1}^N F(\vec{q}_i) e^{i\vec{q}_i \vec{r}}$. Even if our grid contains a few points, we can perform this Fourier transform for an arbitrarily dense \vec{r} grid. Then if we want to know $F(\vec{q})$ at a specific \vec{q} , we just need to calculate the Fourier transform $F(\vec{q}) = \sum_{\vec{r}} f(\vec{r}) e^{-i\vec{q} \vec{r}}$. This kind of interpolation is much more efficient than directly interpolating in reciprocal space, so that reasonable phonon dispersions can be obtained with very small grids.

The dynamical matrices calculated by ph.x are the Fourier transform of the force constant matrix (the linear spring system between atoms in real space). This allows one to calculate the perturbations to the atomic positions without

³**Note:** this grid has nothing to do with the \vec{k} grid of the scf calculation, except that it samples the same Brillouin zone.

the need of a supercell to describe long-wavelength modes. In order to get the phonon dispersions one uses the tool `q2r.x` to transform these dynamical matrices into the force constant matrix (which is defined in real space). Then the force constant matrix can be either read by an external tool, or read by the built-in tools of Quantum Espresso. The tool `matdyn.x` reads the interatomic force constant matrix to calculate the dynamical matrix at a given point in reciprocal space. The points can be defined on a fine grid (to calculate the phonon density of states) or on a path between high-symmetry points in the Brillouin zone (to plot the phonon dispersions).

5.4.1 Example, part 1 - calculate dynamical matrices

First perform a self-consistent calculation (see 5.1). Try to optimize the parameters so that the self consistent calculation is fast (less than two or three minutes) or the phonon calculation will take a very long time.

Execute the code `ph.x`

5.4.2 Example, part 2 - calculate interatomic force constant

Execute the code `q2r.x` (it reads the files in the folder where it is executed).

An explanation of the contents of the `fc` file can be found at [1]. This is only needed if you want to use your own tools to read it.

5.4.3 Example, part 3 - calculate the phonon density of states

Now we can read the interatomic force constant file to calculate the phonon density of states. Run `matdyn.x` with the flag `'dos = .true.'`. A file named `"matdyn.dos"` will be produced. The first column contains the energies (in cm^{-1}), while the second contains the dos ($\text{states}/\text{cm}^{-1}$).

5.4.4 Example, part 4 - calculate the phonon dispersions

This produces a file named `"Cu.phbands.gp"` which contains a gnuplot-friendly data structure which you can plot. **Note:** the first column contains the energies (in cm^{-1}), while the `n` following columns contain a phonon branch each.

If one or more of your modes display *negative* frequencies, it means that your system is dynamically *unstable*. This may have a few different causes. It might be because of an actual physical instability, but more likely your calculation is not converged (try using a larger smearing) or the atoms in your unit cell are not at their equilibrium positions (check that the forces between atoms are less than 10^{-4} Ry/au and that the stress tensor is diagonal and all diagonal elements are equal).

6 Example: result

Using the input files and the pseudopotentials provided, you should in the end obtain this band structure and density of states. This plot was made using `xmgrace`, but you can use any plotting tool you're comfortable with.

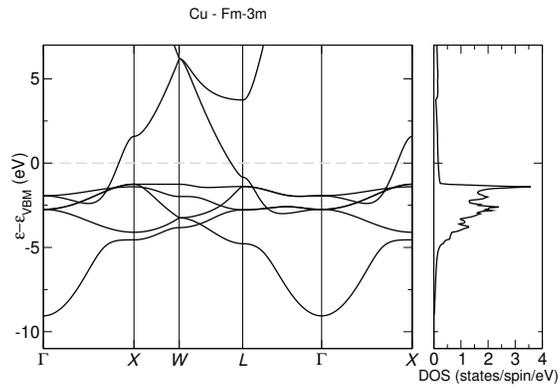


Figure 1: Left panel: band structure calculation for the face centered cubic phase of Cu. Right panel: density of states.

Below we can see the phonon dispersions and phonon density of states for the fcc phase. Note that in Γ three frequencies go to zero, as expected.

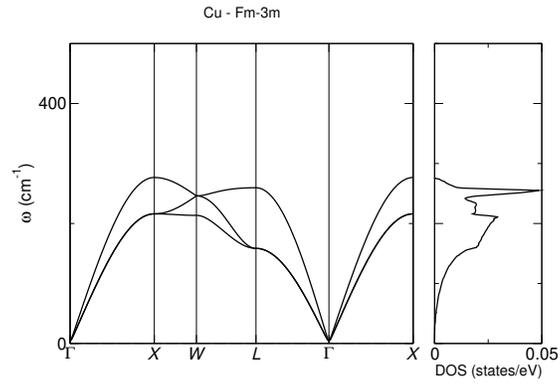


Figure 2: Left panel: Phonon dispersion for the face centered cubic phase of Cu. Right panel: phonon density of states.

References

- [1] Discussion on the interatomic force constant matrix in the qe forum.
- [2] <https://www.quantum-espresso.org>.
- [3] Input files for qe downloadable here.
- [4] N. W. Ashcroft and Mermin. *Solid State Physics*. Saunders College Publishing, 1976.
- [5] Stefano Baroni, Stefano de Gironcoli, Andrea Dal Corso, and Paolo Giannozzi. Phonons and related crystal properties from density-functional perturbation theory. *Reviews of Modern Physics*, 73:515, 2001.
- [6] Robert A. Evarestov and Andrei V. Bandura. First-principles calculations on the four phases of BaTiO_3 . *Journal of Computational Chemistry*, 33:1123–1130, 2012.
- [7] C. Fiolhais, F. Nogueira, and M. Marques. *A Primer in Density Functional Theory*. Springer, 2003.
- [8] Paolo Giannozzi. Introduction to density functional theory (dft).
- [9] J. Harada, T. Pedersen, and Z. Barnea. X-ray and neutron diffraction study of tetragonal barium titanate. *Acta Cryst.*, A26:336, 1970.
- [10] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136(3B):B864–B871, 1964.
- [11] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140(4A):A1133–A1138, 1965.

```

&control

calculation = 'scf'

outdir = './output'

pseudo_dir = '/home/simonedicaldo/QE-pseudopotentials/'
prefix = 'Cu'

tstress = .true.
tprnfor = .true.

forc_conv_thr = 1e-5
&end

&system
ibrav = 0

celldm(1) = 6.90
nat = 4
ntyp = 1
! nbnd = 16

ecutwfc = 120

occupations = 'smearing'

smearing = 'mp'
degauss = 0.02
&end
&electrons
conv_thr = 1.D-8
&end
&ions

ion_dynamics = 'bfgs'
&end
&cell

cell_dynamics = 'bfgs'
press = 0

&end
ATOMIC_SPECIES
Cu 63.54 Cu_ONCV_PBE_sr.upf 27

CELL_PARAMETERS angstrom 3.5999999046 0.0000000000 0.0000000000
0.0000000000 3.5999999046 0.0000000000
0.0000000000 0.0000000000 3.5999999046
ATOMIC_POSITIONS crystal
Cu 0.0000000000 0.0000000000 0.0000000000
Cu 0.0000000000 0.5000000000 0.5000000000

```

! Input file is divided in sections, each section contains variables affected. This most of the time is the calculation type.
! Which type of calculation has been chosen (in this case self consistent field)
! Path to the folder to store the output files. All calculations will have to read this file, so make sure that the path is correct

! It's a label for the calculation, it's used when you read into the output files to identify the calculation.
! Calculates and prints the stress tensor
! Calculates and prints the forces on the ions (if .true.)

! Explained above. 6 is for simple cubic (fcc) and so on
! Units for celldm(1) are Bohr radii
! Number of atoms in the atomic cell
! How many different atomic species
! You can change the number of bands, but be careful, there have to be enough bands for the number of electrons
! Very important! Cutoff energy for the wavefunction expansion.
! Using a smearing on the occupation numbers allows less k points
! Type of smearing
! Width of smearing (in Ry)

! Convergence threshold on the total energy

! Ignored in scf calculations, but used in "vc-relax" calculations

! Ignored in scf calculations, but used in "vc-relax" calculations

! Pressure to be applied to the system during calculations. It is in kilobar (kbar)

! Crystal coordinates

Band structure calculation

```
&control

calculation = 'bands'

outdir = './output'

pseudo_dir = '/home/simonedicaldo/QE-pseudopotentials/'
prefix = 'Cu'

&end
&system
ibrav = 2
celldm(1) = 6.90
space_group = 225

nat = 1
ntyp = 1
! nbnd = 16

ecutwfc = 120

occupations = 'smearing'

smearing = 'mp'
degauss = 0.04
&end
&electrons
conv_thr = 1.D-8
&end
ATOMIC_SPECIES
Cu 65 Cu_ONCV_PBE_sr.upf
ATOMIC_POSITIONS crystal_sg

Cu 4a
K_POINTS crystal_b
6
0 0 0 50
!Gamma 0.5 0 0.5 50
!X 0.5 0.25 0.75 50
!W 0.5 0.5 0.5 50
!L 0 0 0 50
!Gamma 0.5 0 0.5 50
!X
```

! Input file is divided in sections, dependent variables affected. This mostly affects file calculation type.

! Which type of calculation has to be performed (self consistent field)

! Path to the folder to store the output files. Calculations will have to read into this folder.

! It's a label for the calculation. Has to be unique when you read into the output folder in a calculation.

! Explained above. 6 is for simple tetragonal

! Units for celldm(1) are Bohr (1 Bohr \approx 0.529 Å)

! Space group number. Needed in order to determine coordinates using Wyckoff positions

! Number of atoms in the atomic coordinates

! How many different atomic species there are

! You can change the number of bands in the calculation, be careful, there have to be a few more than the number of electrons

! Very important! Cutoff energy on the wavefunction expansion.

! Using a smearing on the occupations so to avoid less k points

! Type of smearing

! Width of smearing (in Ry)

! Convergence threshold on scf calculation

! Convergence threshold on electrons calculation

! The crystal_sg command is used to determine atomic positions from Wyckoff positions

```
&bands
prefix = 'Cu'      Prefix which has to match that of the previous calculations
outdir = './output' Output folder in which data has to be read
filband = 'Cu.bands' ! Name of the file with plottable data
&end
```

Non self consistent calculation

```
&control

calculation = 'nscf'

outdir = './output'
pseudo_dir = '/home/simonedicaldo/QE-pseudopotentials/'
prefix = 'Cu'

tstress = .true.
tprnfor = .true.

&end
&system
ibrav = 2
celldm(1) = 6.9
space_group = 225

nat = 1
ntyp = 1
! nbnd = 16

ecutwfc = 120

occupations = 'tetrahedra'

&end
&electrons
conv_thr = 1.D-8
&end
&ions

ion_dynamics = 'bfgs'
&end
&cell

cell_dynamics = 'bfgs'
press = 0

&end
ATOMIC_SPECIES
Cu 63.54 Cu_ONCV_PBE_sr.upf
ATOMIC_POSITIONS crystal_sg

Cu 4a

K_POINTS automatic
32 32 32 0 0 0
```

! Input file is divided in sections, dependent variables affected. This mostly affects file calculation type.

! Which type of calculation has to be performed (self consistent field)

! Path to the folder to store the output files

! It's a label for the calculation. Has to be unique when you read into the output folder in a calculation.

! Calculates and prints the stress tensor (if .true.)

! Calculates and prints the forces acting on atoms (if .true.)

! Explained above. 6 is for simple tetragonal

! Units for celldm(1) are Bohr (1 Bohr \approx 0.529 Å)

! Space group number. Needed in order to determine atomic coordinates using Wyckoff positions

! Number of atoms in the atomic coordinates

! How many different atomic species there are

! You can change the number of bands in the calculation, but be careful, there have to be a few more than the number of electrons

! Very important! Cutoff energy on the wavefunction expansion.

! Using a smearing on the occupations so that the calculation with less k points

! Convergence threshold on scf calculation

! Ignored in scf calculations, only used in "vc-relax" calculations

! Ignored in scf calculations, only used in "vc-relax" calculations

! Pressure to be applied to the cell for hydrostatic calculations. It is in kilobar (1 kB = 0.1 GPa)

! The crystal_sg command is used to determine the atomic positions from Wyckoff positions

! Since the position of Ba is determined using the Wyckoff positions, no coordinates have to be written

! The first three numbers determine how many k points in each direction have to be used.

```

&dos
prefix = 'Cu'
outdir = './output'
Emin = -20 ! Starting energy for integration
Emax = 50 ! Final energy for integration
DeltaE = 0.1 ! Energy grid step
fildos = 'Cu.dos.dat'
&end

```

```

&inputph
outdir = './output'
prefix = 'Cu'
ldisp = .true. ! Calculate dynamical matrices on a grid in reciprocal
                space
fildyn = 'Cu.dyn' ! Prefix for the name of dynamical matrices
nq1 = 2, ! Defining a grid of points in reciprocal space to calculate
           dynamical matrices
nq2 = 2, ! The choice depends on the size and shape of your cell
nq3 = 2, !
&end

```

```

&input
fildyn = 'Cu.dyn' ! Prefix for the name of dynamical matrices
ffrc = 'Cu.fc' ! Name of interatomic force constant matrix file
&end

```

```

&input
ffrc = 'Cu.fc' ! Force constant file
asr = 'crystal' ! Acoustic sum rule to apply to acoustic modes in gamma
dos = .true. ! Flag to calculate density of states
nk1 = 8, nk2 = 8, nk3 = 8 ! Grid to integrate density of states
deltaE = 1 ! Energy step, in cm-1
&end

```

&input	
ffrc = 'Cu.fc' !	Interatomic force constant file
q_in_band_form = .true. !	If this option is true, then the points should be provided below
q_in_cryst_coord = .true.	Note: the points are given in the exact same way as a band structure calculation
ffrq = 'Cu.phbands' !	Name of the output file
&end	
6 !	Number of different points given (in crystal coordinates)
0 0 0 50	!Gamma
0.5 0 0.5 50	!X
0.5 0.25 0.75 50	!W
0.5 0.5 0.5 50	!L
0 0 0 50	!Gamma
0.5 0 0.5 50	!X

References

- [1] Discussion on the interatomic force constant matrix in the qe forum.
- [2] <https://www.quantum-espresso.org>.
- [3] Input files for qe downloadable here.
- [4] N. W. Ashcroft and Mermin. *Solid State Physics*. Saunders College Publishing, 1976.
- [5] Stefano Baroni, Stefano de Gironcoli, Andrea Dal Corso, and Paolo Giannozzi. Phonons and related crystal properties from density-functional perturbation theory. *Reviews of Modern Physics*, 73:515, 2001.
- [6] Robert A. Evarestov and Andrei V. Bandura. First-principles calculations on the four phases of BaTiO_3 . *Journal of Computational Chemistry*, 33:1123–1130, 2012.
- [7] C. Fiolhais, F. Nogueira, and M. Marques. *A Primer in Density Functional Theory*. Springer, 2003.
- [8] Paolo Giannozzi. Introduction to density functional theory (dft).
- [9] J. Harada, T. Pedersen, and Z. Barnea. X-ray and neutron diffraction study of tetragonal barium titanate. *Acta Cryst.*, A26:336, 1970.
- [10] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136(3B):B864–B871, 1964.
- [11] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140(4A):A1133–A1138, 1965.