

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



Though it can be used on Linux, Mac and Windows, in this post I will show you how to install and use it on Linux-UBUNTU.

Downloading:

Go to [this link](#) and download the package, or download it using the terminal by giving the following command.

```
sudo apt-get install quantum-espresso
```

 (You can skip step 1 if you install it this way)

Installing:

Step 1:

To use Quantum Espresso, you need to extract(uncompress) the zip/tar(compressed) file that you downloaded(You don't need to uncompress it if you used the install command).

To do that just *right click* on the compressed file, and select 'Extract'.

Or you can do it using the terminal by using the following command:

```
tar -zxvf espresso-5.0.2.tar.gz
```

NOTE: Make sure you write the correct file name and version `espresso-5.0.2.tar.gz`. It will probably be different for you.

Once installation is done, a directory will be created with many files and other directories.

Step 2:

Now an important thing to note is that the Quantum Espresso needs a working fortran-95 compiler (most "fortran-90" compilers are ok, "g77" is not). If you don't have it installed, the rest of the steps may not work. You can try performing the next step, if it works then you are good to go, else you need a Fortran compiler.

You can download and install a Fortran compiler by executing the following command in the terminal:

```
sudo apt-get install gfortran
```

Step 3:

Once, you have done the installation, you need to change your directory to the directory of the Quantum Espresso software, and run the following terminal command:

```
./configure
```

You can change the directory by using the command `cd` so if your copy of Quantum Espresso is in a directory called 'qe-6.1' which is in the Home directory, then you can use: `cd qe-6.1`

"configure" is a wrapper, calling the "install/configure" script, that tries to guess your machine and to choose compilation and linking options accordingly. "install/configure" is generated from "install/configure.ac" using the

complex but well-known Unix utility “autoconf”. The result of “configure” is a file called “make.sys” containing the compilation and linking options. You may want to have a look at it to verify what “configure” thinks about your system. Several options to “configure” may (and sometimes, have to) be specified (see below).

If you get a message “architecture xxx not recognized”, specify option “ARCH=...”, such as e.g. (for a BlueGene machine):

```
./configure ARCH=ppc64-bg
```

If “configure” selects a compiler you don’t like (or one that doesn’t like Quantum ESPRESSO) specify option “F90=...” or “MPIF90=...”, e.g.:

```
./configure F90=gfortran
```

If everything is fine you should get a bunch of (mostly obscure and irrelevant) messages but no error. Read the last lines: you may need to understand them if something goes wrong at compilation stage, or if you need to boost performances.

Once this is done, you can compile the packages you will be needing. If you need to perform Plane Wave SCF calculation then you need to compile the ‘pw’ package. To do that run the following command:

```
make pw
```

“make” is another complex but standard unix utility that compiles what is needed in the way it is needed and in the order in which is needed (in Fortran-90 you need to compile modules before programs that use them, for instance). The configuration files for “make” are either coming with the package or are generated by “configure”. If everything goes well, executables will appear in “bin/”:

```
ls bin/*.x
```

You can compile all the available packages at once by running:

```
make all
```

Step 4:

Once you’re done with the above steps, it means you now have the necessary tools to perform a *pw scf* calculation. To perform such a calculation, you need an input file for Quantum Espresso. You can download the sample input file from this link. Once you have the file, save it in a directory of your choice. For the sake of this tutorial, let’s say you save it in a directory called ‘QE’ inside the Quantum Espresso installation directory.

Now, create a directory called ‘temp’ inside the directory that contains the input file (in our example the directory is ‘QE’).

Now let me tell you a little about the input file that we have here. In this file, we are basically trying to find out some of the properties for a *Cu* atom, like the *total energy*, etc.

The input file is a really important file and you will need to learn quite a few things to be able to build one yourself. There are a lot of things that need explaining there, but I guess I will leave that for another post, and in this tutorial we will, just focus on running our first *pw scf* calculation. For the sake of the tutorial you can download the input file that I will be using from here or you can just copy the following and save it in a file called *in*.

```
&control
prefix=''
outdir='temp'
pseudo_dir = '.',
/
&system
ibrav= 2, celldm(1) =6.824, nat= 1, ntyp= 1,
ecutwfc =30.0,
occupations='smearing', smearing='mp', degauss=0.06
```

```

/  

&electrons  

/  

ATOMIC_SPECIES  

Cu 63.546 Cu.pbesol-dn-kjpaw_psl.0.2.UPF  

ATOMIC_POSITIONS  

Cu 0.00 0.00 0.00  

K_POINTS automatic  

8 8 8 0 0 0
    
```

Step 5:

Now the last thing you need to do is to get the *pseudopotential* for your atom.

If you open the input file using a text editor, you will notice the field: `pseudo_dir = ' '`, inside the `&control` section.

Inside the single quotes you need to specify the directory of your pseudopotential. But before that you first need to download one from [here](#). Once you go to the link you can click on any atom to get a list of it's available pseudopotentials. For this tutorial, we need a particular potential for Cu. So click on Cu(29) in the periodic table shown. Now you will see a list of a variety of potentials. Just search for this one : `'Cu.pbesol-dn-kjpaw_psl.0.2.UPF'` and download it.

Step 6:

```
'/home/manas/qe-6.1/bin/pw.x' <in>out
```

Once the execution is done, you will notice a file named out created in the directory with the input file.

Open and you can find information regarding the energies, how many iterations it took to converge, and more.

References and more Resources on Quantum Espresso:

- <http://www.quantum-espresso.org/tutorials/>
- http://www.fisica.uniud.it/~giannozz/QE-Tutorial/tutorial_install.html
- <http://www.fisica.uniud.it/~giannozz/QE-Tutorial/>
- http://www.iiserpune.ac.in/~smr2626/hands_on/week1/jun30/simple_calcs_qe_mastani.pdf
- https://icme.hpc.msstate.edu/mediawiki/index.php/Code:_Quantum_Espresso
- <http://146.141.41.27/Tutorials/day3-QE-I.pdf>
- <http://www.quantum-espresso.org/wp-content/uploads/2013/06/lecture1.pdf>
- http://www.quantum-espresso.org/wp-content/uploads/Doc/INPUT_PW.html
- <http://www.quantum-espresso.org/tools/>



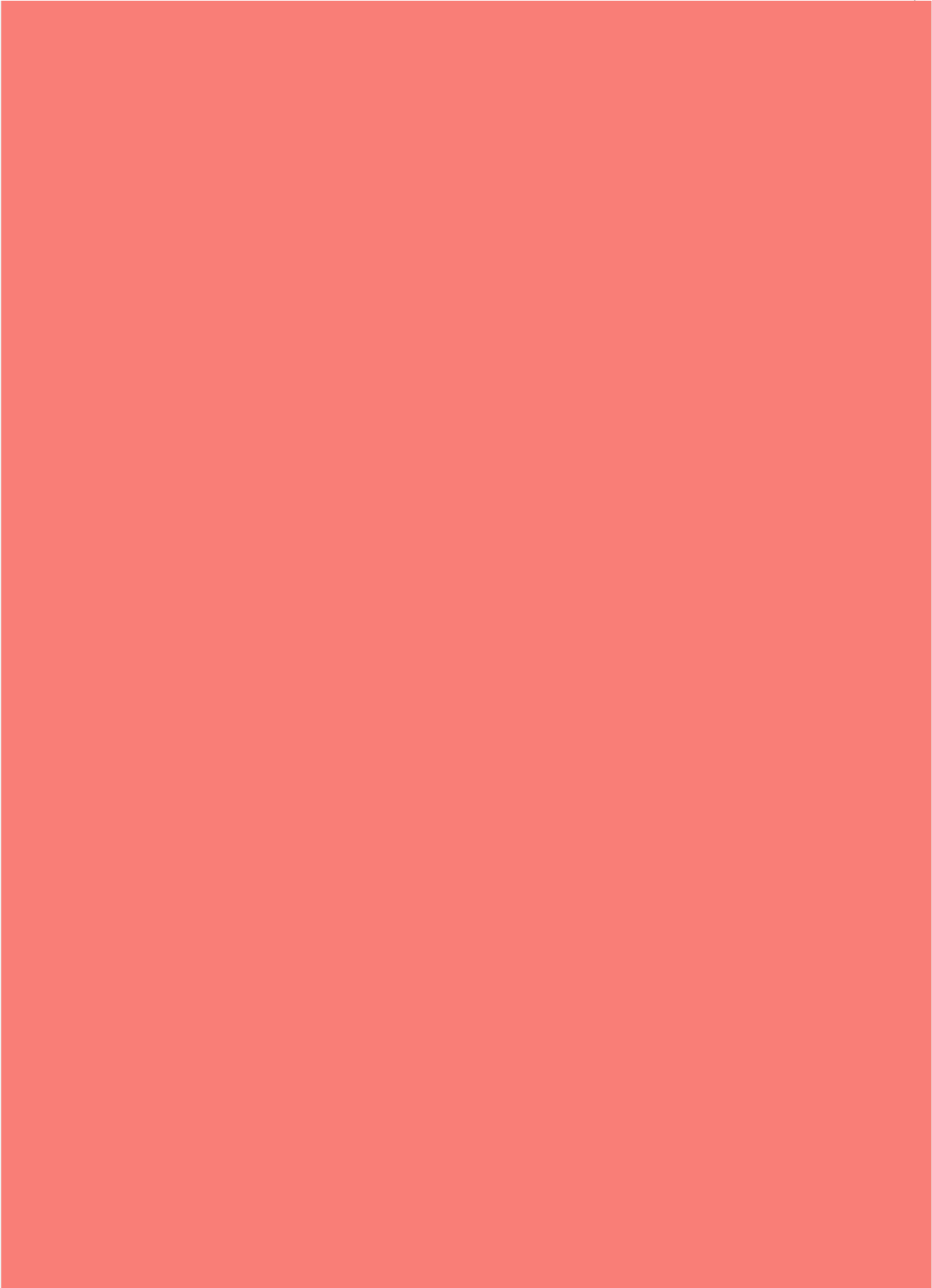
Manas Sharma

I'm a physicist specializing in computational material science with a PhD in Physics from Friedrich-Schiller University Jena, Germany. I write efficient codes for simulating light-matter interactions at atomic scales. I like to develop Physics, DFT, and Machine Learning related apps and software from time to time. Can code in most of the popular languages. I like to share my knowledge in Physics and

Downloading, Installing and an Introduction to QUANTUM ESPRESSO | 4
applications using this Blog and a YouTube channel.

manas.bragitoff.com/







Share this:

[Click to share on Facebook \(Opens in new window\)](#)

[Click to share on Twitter \(Opens in new window\)](#)

[Click to share on WhatsApp \(Opens in new window\)](#)

[Click to share on Pinterest \(Opens in new window\)](#)

[Click to share on Reddit \(Opens in new window\)](#)

[Click to share on LinkedIn \(Opens in new window\)](#)

[Click to email a link to a friend \(Opens in new window\)](#)

[Click to print \(Opens in new window\)](#)

[Click to share on Tumblr \(Opens in new window\)](#)

[Click to share on Pocket \(Opens in new window\)](#)

[Click to share on Telegram \(Opens in new window\)](#)

[wpedon id="7041" align="center"]