Tutorial to using Quantum ESPRESSO with the espresso.py module

Zhongnan Xu

14-8-2014

Contents

1 Introduction

The purpose of tutorial is instruct one how to perform first principles, density functional theory (DFT) calculations efficiently with the QUANTUM-ESPRESSO program along with the python wrapper `espresso.py`. The purpose of this guide, as of now, is *not* instructional for...

1. Physics behind implementations of DFT
2. Setting up specific atoms objects for calculations
3. Calculating physical or thermodynamic properties from DFT

Instead, we will go over some of the nuts and bolts of successfully running DFT calculations. These topics include

1. Writing code for submitting, checking, and reading PWSCF and DOS calculations
2. Some parameter optimization tests
3. Dealing with common errors

Therefore, it assumed that user has had some experience with ASE, and the explanations of constructing the atoms object will not be given.

2 Installation

The espresso.py module requires both QUANTUM-Espresso and the Atomic Simulation Environment (ASE) to be installed and working on the system. These are both free and can be found at http://www.quantum-espresso.org/ and https://wiki.fysik.dtu.dk/ase/, respectively. All python modules required are the same as listed for ASE.

Once the espresso.py module has been downloaded, just add this line to your path.

```
export PYTHONPATH=path/to/espresso/espresso:$PYTHONPATH
```

Note, the espresso folder within the overall espresso.py module folder (also named espresso) is the path that must be added.

3 Performing calculations

3.1 PWSCF calculations

The most common calculation you will be performing are Plane Wave Self-Consistent Field (PWSCF) calculations. In these calculations, we specify the atomic coordinates, pseudopotentials, and parameters, and it calculates for us important thermodynamic quantities such as the total energy, force, and stresses. After a successful calculation, we can then extract the data.

Note, performing this calculation does not give us human readable information on the electronic structure besides basic information on band occupancies. Obtaining useful electronic structure information for the construction of band structures and density of states (DOS) require followup calculations from programs within the QUANTUM-Espresso module and will be summarized in following sections. However, it all starts with the pwscf command.

3.1.1 The simplest possible example: H in a box

One of the simplest systems to calculate (though not necessarily the fastest) is an atom in a box. Because of the large size of the box and sharp gradients in the electronic density due to it being an atomic state, electronic convergence can be surprisingly slow. However, this is by far the easiest system to imagine, so we start with this to show the basics of running commands.

The beginning of the script will always start the same

```
from espresso import * # First import the module
```

This command imports all of the commands you will ever need to run QUATNUM-Espresso calculations. This will be imported pretty early on.

After this, we need to construct our atoms object. This is done below. The H atom is at the (0, 0, 0) coordinate, and the cell is a cuboid with edge lengths 8, 9, and 10 Å long. We need a cuboid to break symmetry, which is required for finding the ground states of gaseous atoms. The box needs to be large enough to minimize neighbor interactions.

```
from ase.atoms import Atoms, Atom # Import the atoms object from ASE

atoms = Atoms([Atom('H', (0, 0, 0))],
              cell = (8, 9, 10))
```

Now we run the calculation.

```
with Espresso('examples/output/H',          # With respect to the directory this script
                                            # is in, this is the directory where the
                                            # calculation will be taking place. The module
                                            # will automatically make the folders necessary.
                                            # Just assure the folder doesn't exist, and if it
```

```
                                           # does, that it’s empty

              atoms=atoms,                 # This is where we put in the atoms object

              ecutwfc=60.0, ecutrho=600.0, # These are the kinetic energy cutoff parameters
                                           # These values determine heavily the convergence
                                           # of your calculation and therefore the time and
                                           # accuracy of your calculation. You should perform
                                           # convergence tests before performing large amounts
                                           # of studies.

              kpts=(1, 1, 1),              # This is how many kpoints in the x, y, and z
                                           # direction of the unit cell. Similar to ecutwfc
                                           # and ecutrho, the more kpoints the more converged
                                           # and expensive. Testing is recommended.

              occupations=’smearing’,      # This is to determing the smearing at electrons
                                           # at the fermi level. Typically we do smearing.

              smearing=’gauss’,            # The type of smearing we want. Typically its gauss
                                           # for insulators and mp (methfessel-paxton) for
                                           # metals.

              degauss=0.01) as calc:       # The width of the smearing. Will dicuss this value
                                           # later.
    calc.calculate()
```

Performing this calculation will return one of four things.

1. The exception `EspressoSubmitted`

   This means your job was successfully submitted! Whether its correctly running or not is a different story, and examples of this will be covered in later sections. If you submit job for the first time in an empty or nonexistent directory, you should always receive this exception.

2. The exception `EspressoRunning`
   This means your job is either queued or running. If you catch this, nothing was modified in any of your input files.

3. The exception `EspressoNotConverged`
   This means the job has finished, but for some reason or another the calculation is not converged. Besides directly looking at the calculation directory yourself, the `espresso.py` module does contain some info on what happened and what you should do. Coverage on troubleshooting techniques will be later sections.

4. Nothing
   This means your calculation was submitted earlier, has finished, and was converged.

To catch these exceptions, one would re-write calculation line (calc.calculate). Mine typically look like this.

```
    try:
        calc.calculate()
        print calc.espressodir, ’Complete’
    except (EspressoSubmitted, EspressoRunning):
        print calc.espressodir, ’Running’
    except (EspressoNotConverged):
        print calc.espressodir, ’Not Converged’
```

In the code above, I have it print either ’Complete’, ’Running’, or ’Not Converged’ so I know what the status is. The object calc.espressodir is just a string that’s the same as the relative directory path you’ve entered above. In this script, it would just be ’examples/output/H’. The complete code is below and can be run directly. The downloadable python script can be found in the attachment below.

```
from espresso import * # First import the module
from ase.atoms import Atoms, Atom # Import the atoms object from ASE
```

```

atoms = Atoms([Atom('H', (0, 0, 0))],
              cell = (8, 9, 10))

with Espresso('output/H',                  # With respect to the directory this script
                                           # is in, this is the directory where the
                                           # calculation will be taking place. The module
                                           # will automatically make the folders necessary.
                                           # Just assure the folder doesn't exist, and if it
                                           # does, that it's empty

              atoms=atoms,                 # This is where we put in the atoms object

              ecutwfc=60.0, ecutrho=600.0, # These are the kinetic energy cutoff parameters
                                           # These values determine heavily the convergence
                                           # of your calculation and therefore the time and
                                           # accuracy of your calculation. You should perform
                                           # convergence tests before performing large amounts
                                           # of studies.

              kpts=(1, 1, 1),              # This is how many kpoints in the x, y, and z
                                           # direction of the unit cell. Similar to ecutwfc
                                           # and ecutrho, the more kpoints the more converged
                                           # and expensive. Testing is recommended.

              occupations='smearing',      # This is to determing the smearing at electrons
                                           # at the fermi level. Typically we do smearing.

              smearing='gauss',            # The type of smearing we want. Typically its gauss
                                           # for insulators and mp (methfessel-paxton) for
                                           # metals.

              degauss=0.01) as calc:       # The width of the smearing. Will dicuss this value
                                           # later.
    try:
        calc.calculate()
        print calc.espressodir, 'Complete'
    except (EspressoSubmitted, EspressoRunning):
        print calc.espressodir, 'Running'
    except (EspressoNotConverged):
        print calc.espressodir, 'Not Converged'
```

3.2 DOS calculations

4 Parameter optimization

5 Troubleshooting