

convergence testing

The procedure that is sketched in this document is one you should follow for *every* new crystal you want to calculate using any DFT code. Sticking strictly to this procedure guarantees that you will calculate numerically meaningful results, rather than random noise. This is what is meant by making a DFT calculations that is (numerically) *converged*.

Please make sure you read before the document about ‘three major keywords in the QE input file’. The present document is an exercise about what you read there.

You will make this exercise for the [halite mineral](#), also known as rock salt. Start your virtual machine, open a terminal window, descend into your working folder (create a new subfolder for halite), and get a cif file for this structure from a crystallographic database (more details about the latter will follow in the chapter on crystallography):

```
cd
cd workQE
mkdir halite
cd halite
wget http://www.crystallography.net/cod/9008678.cif
cif2cell 9008678.cif -p quantum-espresso -o halite.in
```

Download pseudopotentials for the elements chlorine and sodium:

```
wget http://www.quantum-espresso.org/upf_files/Na.pbe-spn-kjpaw_psl.1.1.0.0.UPF
wget http://www.quantum-espresso.org/upf_files/Cl.pbe-n-kjpaw_psl.1.1.0.0.UPF
```

Add the same kind of extra lines to halite.in as you did for the basic silicon example before. The final file should look like this:

```
*****
#*                               Generated by cif2cell 1.2.10 2018-09-17 14:27                               *
#*   T. Bjorkman, Comp. Phys. Commun. 182, 1183-1186 (2011). Please cite generously. *
#*                               *
#*           Data obtained from COD. Reference number : 9008566                               *
#*                               ()                               *
#*           Wyckoff, R. W. G., Crystal Structures 1, 7-83 (1963)                               *
#*                               *
*****

&CONTROL
  calculation='scf',
  outdir='.',
  prefix='halite',
  pseudo_dir='.',
  verbosity='low',
  tprnfor=.true.,
  tstress=.true.,
/
```

```

&SYSTEM
 ibrav = 0
A =    5.64056
nat = 2
ntyp = 2
ecutwfc=66,
ecutrho=323,
input_dft='pbe',
occupations='smearing',
smearing='mv',
degauss=0.005d0,
/

&ELECTRONS
conv_thr=1d-08,
mixing_beta=0.7d0,
/

CELL_PARAMETERS {alat}
0.5000000000000000    0.5000000000000000    0.0000000000000000
0.5000000000000000    0.0000000000000000    0.5000000000000000
0.0000000000000000    0.5000000000000000    0.5000000000000000

ATOMIC_SPECIES
Na  22.98900  Na.pbe-spn-kjpaw_psl.1.1.0.0.UPF
Cl  35.45150  Cl.pbe-n-kjpaw_psl.1.1.0.0.UPF

ATOMIC_POSITIONS {crystal}
Cl  0.5000000000000000    0.5000000000000000    0.5000000000000000
Na  0.0000000000000000    0.0000000000000000    0.0000000000000000

K_POINTS {automatic}
3 3 3 0 0 0

```

1. choose your exchange-correlation functional

The first choice is an easy one, and does not really affect the numerical convergence. Which approximation to the exchange-correlation functional will we take? This choice will determine in a rather unpredictable way how much your final results will deviate from the (unknown) true result, obtained by using the (unknown) exact exchange-correlation functional.

In this course, we will use always the PBE exchange-correlation functional, unless explicitly noted. This means that in the &SYSTEM block you should have this line:

```
input_dft='PBE',
```

Cross-check that the two pseudopotential files are indeed for the PBE functional as well (we'll see later how to do that systematically, but for now you can verify that the long name of the pseudopotentials mention the string 'pbe').

In the next two steps, we'll find out how to guarantee that you'll find the numerically correct PBE-predictions, and not a value with random noise.

1.5 suggested default for the basis set size

(this is step one-and-a-half, as not every pseudopotential file has the required information to perform this step – in that case, just skip it) Open the two pseudopotential files that you downloaded. Sometimes, the header of this file contains a suggestion for the basis set size for the wavefunctions (ecutwfc) and the basis set size for the density (ecutrho). If this information is present, let's make use of it.

In those pseudopotential files, you'll see these lines:

Na-file:

```
Suggested minimum cutoff for wavefunctions: 66. Ry  
Suggested minimum cutoff for charge density: 323. Ry
```

Cl-file:

```
Suggested minimum cutoff for wavefunctions: 45. Ry  
Suggested minimum cutoff for charge density: 223. Ry
```

This means that $ecutwfc=66$ and $ecutrho=323$ (for both parameters the maximum of the two suggested values) would be a fair choice for the basis set sizes. We'll test it explicitly in step 3, but for now we'll use these values as a start.

2. choose your k-mesh

The number of sampling points is expressed as a three-dimensional grid in reciprocal space ("k-mesh"), and is set in &ELECTRONS block with the K_POINTS keyword. The default is a 1x1x1 mesh. To set a 3x3x3 mesh, use

```
K_POINTS {automatic}  
  3 3 3 0 0 0
```

To use a 5x5x5 mesh, use

```
K_POINTS {automatic}  
  5 5 5 0 0 0
```

The goal is to run several DFT calculations, with increasingly dense k-meshes. You will monitor one or more experimentally observable properties, and see how these evolve as a function of k-mesh

density. As soon as these properties do not depend any longer on the density of the mesh, you have reached the result that corresponds to an ‘infinitely’ dense mesh (i.e. you reached the numerically correct result). These experimental property is then said to be converged with respect to k-mesh density.

The hydrostatic pressure on a given unit cell (we’ll learn later what this means) is an experimentally observable property that is rather sensitive to numerical precision. If this property is converged with respect to k-mesh, many other less sensitive properties will be converged too. Therefore, it is a good idea to monitor the hydrostatic pressure (that’s why we added the `tstress` keyword).

You get a table with the results hereafter. Perform a few of these calculations (or all of them) yourself, to make sure you can reproduce this table. The input file that has been given above is the one that has been used for the 3x3x3 mesh in this series.

k-mesh	Hydrostatic pressure (kbar)	Typical run time (seconds)
1x1x1	243.81	7.44
3x3x3	3.02	11.14
5x5x5	1.23	13.80
7x7x7	1.09	19.14
9x9x9	1.12	28.86
11x11x11	1.13	43.80

From this table, you see that the calculation time increases with increasing k-mesh. That’s why you cannot simply take by default a very dense and safe k-mesh: the calculations would take way too long. If your k-mesh is too coarse (say the 1x1x1 mesh) then your calculation is superfast (8 seconds only), but the predicted hydrostatic pressure is nuts. If you do only this single calculation, you won’t know it is nuts, however. You see this only if you try a denser k-mesh: the predicted pressure changes from 244 to 3 kbar. If you keep increasing the k-mesh density, the hydrostatic pressure stabilizes around 1.1 kbar. To stay on the safe side, let’s select a 7x7x7 mesh as our compromise. With this mesh, you spend 19 seconds to find the converged result. If you would take the 11x11x11 mesh, then you would need 44 seconds to find... the same result. The difference between 19 and 44 seconds might not feel to be much. But if you have a calculation that takes 3 hours or even 3 days, you don’t want to spend 6 hours or 6 days to find exactly the same prediction. If your resources are limited, you might dare to use a 5x5x5 mesh too, which will give you probably a similar prediction in even less time.

Conclusion: it is a good compromise to set the k-mesh to 7x7x7. We’ll continue with that value to select the proper basis set size.

3. choose your basis set size

It is usually OK to keep `ecutrho` as a multiple of `ecutwfc`. We can therefor vary `ecutwfc` first, keeping `ecutrho` at the same factor times `ecutwfc`. Once a good value for `ecutwfc` has been found, the story can be repeated for that multiplication factor. With the suggested values of `ecutwfc=66` and `ecutrho=323`, the multiplication factor is nearly 5. Let’s make it exactly 5 for convenience (that gives

ecutwfc=66 and ecutrho=330, and due to that small change in ecutrho the hydrostatic pressure on the corresponding line in the next table will be slightly different from 1.1 kbar). Keeping the factor 5 fixed, let us try a few values for ecutwfc:

Ecutwfc	Ecutrho	Hydrostatic pressure (kbar)	Typical run time (seconds)
16	80	-5324.4	4.29
26	130	-865.6	6.54
36	180	-268.2	10.54
46	230	-9.0	12.42
56	280	4.3	14.34
66	330	0.9	19.18
76	380	4.4	24.55
86	430	7.1	31.99
96	480	7.8	34.94
...			
200	800	8.0	111.92

The calculation time increases with the value of ecutwfc, hence also for this parameter it is important to select the minimal value that gives you safely the converged result.

The table shows that the initial ecutwfc=66 was not bad: it gives a hydrostatic pressure of 0.9 kbar (or 1.1 kbar with ecutrho=323), which is only a few kbar away from the high-precision value of 8.0 kbar. At least this is much better than what a basis set with ecutwfc=36 would give. Is it worth to spend 112 seconds to find the correct value of 8.0 kbar? The table shows that if we are satisfied with an uncertainty of 1 kbar, we can use ecutwfc=86. In only 32 seconds it gives us a hydrostatic pressure of 7.1 kbar, which is less than 1 kbar away from the correct value of 8.0 kbar. Let's therefore keep the slightly rounded value of ecutwfc=90 from now on.

The last step is to examine the multiplication factor between ecutwfc and ecutrho, which as fixed at 5 so far. Let's examine it from 2 to 10 in some steps:

factor	Ecutrho	Hydrostatic pressure (kbar)	Typical run time (seconds)
2	180	9.11	18.3
3	270	7.88	24.3
4	360	7.43	28.5
5	450	7.56	29.5
6	540	7.70	30.6
7	630	7.48	34.2
...			
10	900	7.61	40.5

The effect is small, and everything from a factor of 4 onwards looks acceptable. Given the very small time difference between a factor 4 and 5, we can stay on the safe side and take ecutrho=450 (factor = 5).

Conclusion

The tests as described above for the k-mesh and basis set size MUST be done for every new type of calculations you will perform. Without such tests, you may either be calculating noise rather than stable values, or you may have precise results but are spending 10 times more computing time than you need. This is called “convergence testing”.

For the halite crystal, a good set of values turns out to be:

```
7x7x7 k-mesh
ecutwfc=90 Ry
ecutrho=450 Ry
```

This ensures a hydrostatic pressure that is converged to about 1 kbar or better. If you're happy with a slightly more noisy value of the pressure, then a k-mesh of 5x5x5, ecutwfc=76 Ry and ecutrho=304 Ry are fine too. The latter calculation will be faster, the former will be more precise. There will always be a trade-off between precision and speed.

Addendum

There is currently work going on at several places to automatize and/or accelerate these convergence testing procedure. We mention here one of these newer tools, that is not yet perfect but may give at least a few hints:

<https://www.ctcms.nist.gov/jarvisml/>

It goes as follows (for the same halite crystal we just dealt with):

Use cif2cell to convert the cif file into the input format for the VASP code:

```
cif2cell 9008678.cif -p vasp -o poscar --vasp-format=5 --vasp-cartesian
nano poscar
```

copy the 2nd line up to the line before ‘cartesian’”

```
1.0
  0.0000000000000000    2.8202800000000000    2.8202800000000000
  2.8202800000000000    0.0000000000000000    2.8202800000000000
  2.8202800000000000    2.8202800000000000    0.0000000000000000
Na Cl
1 1
```

Then do this conversion again, in a slightly different way:

```
cif2cell 9008678.cif -p vasp -o poscar --vasp-format=5
nano poscar
```

copy the text from 'Direct' onwards :

```
Direct
  0.0000000000000000  0.0000000000000000  0.0000000000000000
  0.5000000000000000  0.5000000000000000  0.5000000000000000
```

Glue both blocks of text together, preceded by 'My_3D_material', until you get:

```
My_3D_material
  1.0
  0.000000  2.820280  2.820280
  2.820280  0.000000  2.820280
  2.820280  2.820280  0.000000
Na Cl
  1  1
  direct
  0.00000  0.00000  0.00000
  0.50000  0.50000  0.50000
```

Paste this into the white box on the JARVIS-ML website, and press 'send'.

As output, you will get an estimate of the required k-mesh (1x1x1, quite a bit lower than we tested it should be), an estimate for the required basis set size (energy cut-off, only for the VASP code, not for Quantum Espresso), and even some estimated properties without doing the actual DFT calculation (formation energy, elastic moduli,...)

Again, this is all quite rough and premature, but if you start with a new crystal and you have really no idea about the settings you should use to even start testing, this tool might give you a first hint for the k-mesh at least:

```
Bandgap OptB88vdW (eV): 4.07
Bandgap TBmBJ (eV): 6.41
Cut-off (eV): 594.04
Formation energy/atom (eV): -1.976
Kpoint: 1x1x1
Modulus bulk (GPa): 27.62
Modulus shear (GPa): 17.74
Space-group: Fm-3m
Static Refractive-index OptB88vdW (y): 1.48
Static Refractive-index OptB88vdW (z): 1.36
Static Refractive-index TBmBJ (y): 1.59
Static Refractive-index TBmBJ (z): 1.44
Static refractive-index OptB88vdW (x): 1.48
Static refractive-index TBmBJ (x): 1.49
```