

# Lithium ELNES with WIEN2k

August 31, 2018

## 1 Tools

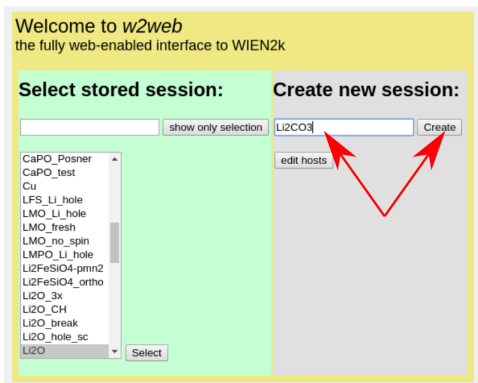
Here's a list of software I use to run simulations. I've put some installation instructions for linux as needed.

- Wien2k -You either already have it installed somewhere, or should figure out how to do that before buying a license.
- VESTA: <http://jp-minerals.org/vesta/en/download.html>. Download the .rpm file, install it with your package manager, eg. "sudo apt-get vesta...rpm"

## 2 Setup

We need a couple things to get started for ELNES simulations. Foremost is a crystal structure. You can either get this from the literature, XRD, or alternatively Materials Project: <https://materialsproject.org/>. Download a cif (the primitive cell typically) or enter the coordinates directly into the wien2k struct gen tool.

Make a new Wien2k session:



Create/change a working directory, and change the session information for parallel calculation.

#### w2web, the fully web-enabled interface to WIEN2k

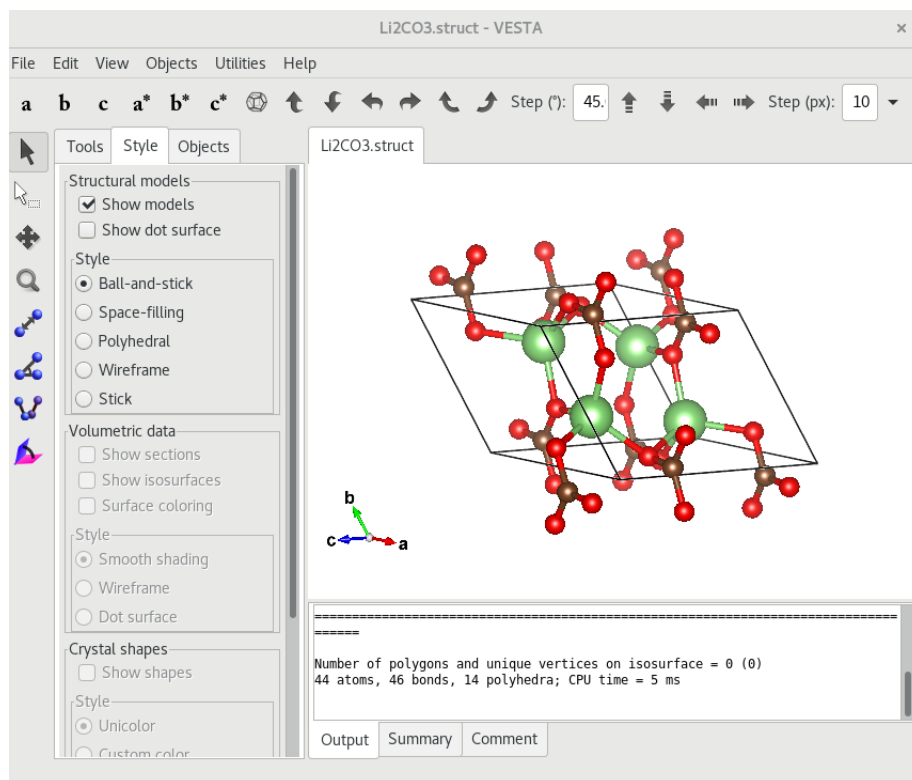
Session Name: Li2CO3  
Session ID: 811721  
Directory: /home/qstoyel/WIEN2k/Li2CO3  
Last changed: Fri Aug 24 14:36:20 2018  
Comments:  
☐ spin polarized calculation  
☐ AFM calculation  
☒ complex calculation (no inversion)  
☒ parallel calculation

[Change session information](#)

You will also need to make a “.machines” file which you can either steal from one of my directories, look in the user guide and make your own, cut and paste the one from below, or wait for w2web to automatically generate one at some point after it inevitably crashes on something. Sample .machines file:

```
#####  
#This is a valid .machines file  
#  
granularity:1  
1:localhost #as many of these lines as you want cpu cores running  
1:localhost  
1:localhost  
1:localhost  
1:localhost  
1:localhost  
1:localhost
```

Next, go make a struct file, with struct gen, either by importing the cif, or entering the positions manually. Use VESTA (drag n’ drop the .struct file) to make sure that the structure is what you’d expect. In our Li2CO3 case this looks like:



Now we need to initialize the case. You can do this in w2web, but we are going to get our hands dirty anyways, so I like to use the command line for this and see what's actually going on. Run “init\_lapw” in the case directory.

- **setrmt.** Sets the muffin tin size on the atoms. Reduce the sphere size by 0% using either old or new scheme and accept, it doesn't matter much as we are going to change this all later.
- **nn.** Checks for overlapping muffin tins. Enter “2.0”, close the first file and use the new NN file if it suggests it, run nn with 2.0 again, look at how much “wiggle room” you have on the spheres, see pic.

```

File Edit View Search Terminal Help
qstoyel@Li2CO3$ init_lapw
next is setrmt
Automatic determination of RMTs. Please specify the desired RMT reduction
compared to almost touching spheres.
Typically, for a single calculation just hit enter, for force minimization
use 1-5; for volume effects you may need even larger reductions.

Enter reduction in %
0
Use old or new scheme (o/n)
n
specify nn-bondlength factor: (usually=2) [and optionally dlimit, dstmax (about
1.d-5, 20)]
DSTMAX: 20.000000000000000
iix,iy,iiz      3      3      3      27.818223000000000      27.
818223000000000      35.986764000000001

ATOM 1 Li      ATOM 8 O
RMT( 1)=1.63000 AND RMT( 8)=1.27000
SUMS TO 2.90000 LT. NN-DIST= 3.63663

ATOM 2 Li      ATOM 8 O
RMT( 2)=1.63000 AND RMT( 8)=1.27000
SUMS TO 2.90000 LT. NN-DIST= 3.63663
  
```

- **sgroup**. Verifies the space group. Again, accept any changes the program makes, these steps are all about reducing the cell symmetries to what they should be. Again, the files can be largely ignored at this point, with the exception of indication of a Bravais lattice change, in which case just take the new struct file. If so, nn and sgroup will run again with “nice” results.
- **symmery**. Generates all the symmetry operations. Run it and continue (enter “c”)
- **lstart**. Set spin state, pick your XC kernel, define cutoff between core and valence states. Accept default spins, unless you have a transition metal, and and GGA PBE as you XC potential. Picking the energy is the most important part of this first run init\_lapw, we want to try to get the Li 1s states to be treated as core states. They typically have energies of  $\sim -3.5\text{Ry}$ , so try with that first, and look in case.outputst (the file that pops up), for the following lines for the lithium atom and look at the 1S states:

	E-up (Ry)	E-dn (Ry)	Occupancy		q/sphere	core-state
1S	-3.801947	-3.785288	1.00	1.00	0.9859	T
1S	-3.801947	-3.785288	1.00	1.00	0.9859	T
2S	-0.236699	-0.003313	1.00	0.00	0.0468	F
2S	-0.236699	-0.003313	1.00	0.00	0.0468	F

These indicate the core states (T/F), their energy levels (in this case  $\sim -3.8\text{eV}$ ) and how much the electrons in these states live in the muffin tins (0.9859). As this is less than 1, it means a lot of 1S lithium electron is leaking out of the muffin tins, which is why there should now be all kinds of warnings popping up. So go ahead and “ctrl-c” out of init\_lapw.

To fix the leakage problem, we need to make the Lithium muffin tins bigger. We want to make them just big enough to hold all of the 1S electrons, without making them too different from the other muffin tins, as the larger this difference, the harder things get to calculate/converge. In this case we try Li=1.8, C=1.2, O=1.22 and try init\_lapw again.

- **setrmt**: Setrmt will try to reset the muffin tins to the defaults, make sure to discard these (enter d)
- **nn** Make sure you don’t get errors, and that everything is as tight as it can be, in this case the Oxygen-carbon spacing is the limiting factor.
- **sgroup** Should run fine.
- **symmery** Should run fine.
- **lstart** Rewrite the old file, note increase in electron containment for Lithium (0.9922) in case.outputst. Go back and loop through increasing the muffin tins a little more (1.85,1.9,2.0) and checking the containment until it gets above 0.995 and the warnings stop.

- **kgen** Set the RkMax value in case.in1.st:

```

GNU nano 2.5.3 File: Li2CO3.in1.st
#FFIL EF= 0.50000 (WFFIL, WFPRI, ENFIL, SUPWF)
7.00 16 4 (R-MT*K-MAX; MAX L IN WF, V-NWT)
0.30 1 0 (GLOBAL E-PARAMETER WITH n OTHER CHOICES, global APW/LAPW)
0 0.00 0.000 CONT 1
0.30 3 0 (GLOBAL E-PARAMETER WITH n OTHER CHOICES, global APW/LAPW)
0 -0.78 0.002 CONT 1
0 0.30 0.000 CONT 1
1 0.30 0.000 CONT 1
0.30 3 0 (GLOBAL E-PARAMETER WITH n OTHER CHOICES, global APW/LAPW)
0 -1.55 0.002 CONT 1
0 0.30 0.000 CONT 1
1 0.30 0.000 CONT 1
0.30 3 0 (GLOBAL E-PARAMETER WITH n OTHER CHOICES, global APW/LAPW)
0 -1.55 0.002 CONT 1
0 0.30 0.000 CONT 1
1 0.30 0.000 CONT 1
K-VECTORS FROM UNIT:4 -11.0 1.5 101 emin / de (emax=Ef+de) / nband

```

and then pick a k\_point number. Both of these values should be taken for fast convergence, in this case I chose RkMax=7.0, and 16 k points.

- **Dstart**: make sure to pick non spin polarized, unless you have reason to believe otherwise (is there a transition metal in your sample?)

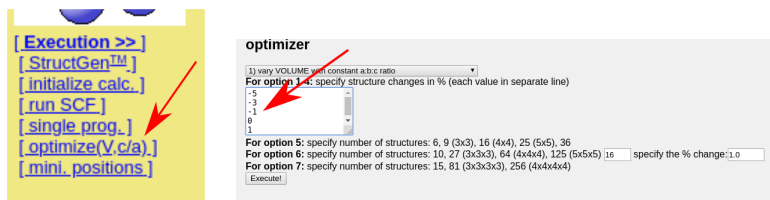
Assuming there were no warnings in the final run through `init.lapw`, we can now start convergence.

## 3 Convergence

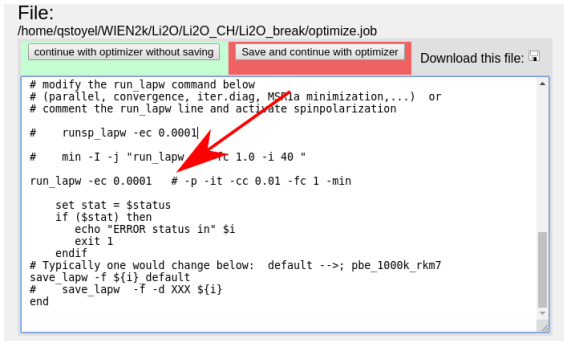
Ideally, you want to converge everything regarding your simulation. Typically this is cell parameters, k points and Rkmax. The first step is to just make sure the calculation converges, running it with a small RKmax and few kpoints and making sure it finishes.

### 3.1 Cell parameters:

This process works best from W2Web, as described in the tutorials. You might need to make your muffin tins smaller to avoid nn errors. You can do this with the muffin tins we picked out above, or just use the ones **setrmt** suggests and reduce them by 5-10%. As you need run a number (5-11+) of calculations, you want to use low k points and RKMax values, for Li2CO3, I used 64 kpoints and an RKmax of 7 initially. In the x “optimize” tab, choose what you want to optimize, the first option (volume) is typically sufficient, unless you have suspicions otherwise. Enter a range of values of test volumes, see picture:



Also make sure to edit “optimize.job” to enable parallelization by moving the “#” to after the “-p” in the `run.lapw` line:

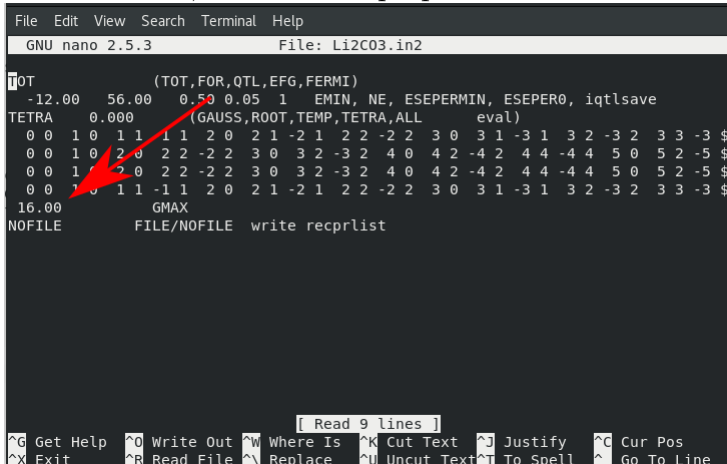


You can then “run optimize.job” from w2web.

## 4 Common Errors Thrown by Code

### 4.1 GMax Value less than Gmin

Occurs in **dstart**, fix is to bump up the Gmax value in case.in2 from 12.00 to 14.00 or 16.00.



### 4.2 K point and RKMax convergence

To converge these parameters, again start with very low values and then increase them, checking the total energy to determine when they are converged. I like to start with k points and then move on to RKMax. To bump up the values, you want to either rerun “x kgen” and enter a larger k point value, or edit case.in1 and case.in1\_st, and increase the RkMax

- setrmt
- nn
- sgrouop
- symmery
- lstart