

Lithium ELNES with WIEN2k

September 10, 2018

Contents

1	Tools	1
2	Setup	2
3	Convergence	5
3.1	Cell parameters:	6
3.2	K point and RKMax convergence	7
4	TELNES3	7
4.1	Monopole effects	9
5	Core Hole	9
6	Common Errors/Issues Encountered in Wien2k	11
6.1	Setting RMT/RKMax	11
6.2	NN in Optimization	11
6.3	GMax Value less than Gmin	11

1 Tools

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

- Wien2k - If you are reading this guide, 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"
- Critic2: <https://github.com/aoterodelaroza/critic2>. Download the zip from GitHub, unzip where you want to install it eg ~/Programs or something), install (using dnf/apt-get) autoconf, automake, and your favourite flavour of fortran. Then run the 4 commands from the readme: "autoreconf -i", "./configure", "make", and "(sudo) make install". You may or may not need the sudo for the last one.

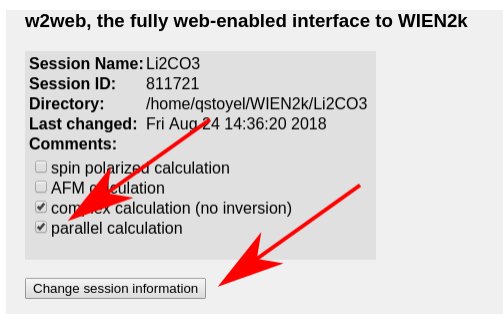
2 Setup

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:



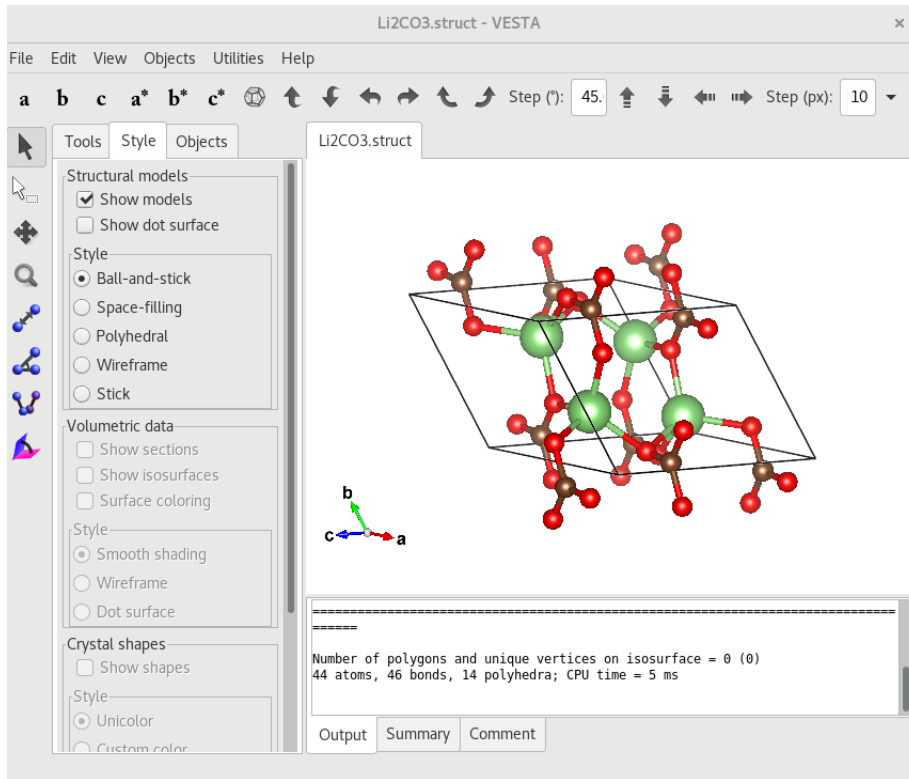
Create/change a working directory, and change the session information for parallel calculation.



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 #ideally pick a multiple of kpoint number
1:localhost #so don't use 5 cores for 13 k points...
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 Li_2CO_3 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.

```

Terminal
File Edit View Search Terminal Help
qstoyel@Li2C03$ init_lapw
next is setrat
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,iiz 3 3 27.818223000000000 27.
818223000000000 35.986764000000001
ATOM 1 Li ATOM 8 0
RMT( 1)=1.63000 AND RMT( 8)=1.27000
SUMS TO 2.90000 LT. NN-DIST= 3.63663
ATOM 2 Li ATOM 8 0
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 (up, the no spin case), unless you have a transition metal. Select GGA PBE as you XC potential, again, unless you have reason to suspect otherwise. 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 $\sim -3.7\text{Ry}$, so try with -3.5Ry to make sure they will be treated as core states 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, the Lithium muffin tins need to be bigger. Ideally, they should be 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, try Li=1.8, C=1.2, O=1.22 and try init_lapw again, if that still didn’t work (lstart still had leakage errors), keep going until it does, in this case RMT’s of Li=2.0, C=1.14, O=1.22. These RMT values are too different ($\text{RMT}_{\text{Li}} \approx 2\text{RMT}_{\text{C}}$) and will cause

errors further along. The solution here is to simply accept the leakage (ignore the warnings) and acknowledge that it might be unavoidably causing artifacts. Alternatively, pick a nicer structure to analyze.

- **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** Now that the lithium 1S states are well contained, cre states can be selected based on containment instead of energy meaning, the higher energy states (2S, P) of other elements can still be treated as valence. Entering 0.995 should be sufficient here, but make sure to verify that the lithium states are still core states.
- **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 10 4 (R-MT*K-MAX; MAX L IN WF, V-NMT)
0.30 1 0 (GLOBAL E-PARAMETER WITH n OTHER CHOICES, global APW/LAPW)
0 0.30 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 initially be taken for fast convergence, in this case I chose RkMax=7.0, and 8 k points (Li2CO3 is an insulator, for metals 1000 k points is a good starting point).

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

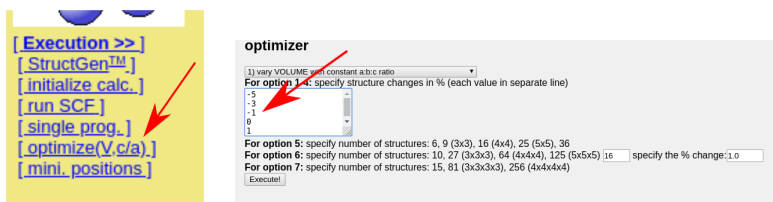
Assuming there were no warnings in the final run through init_lapw, we can now start convergence.

3 Convergence

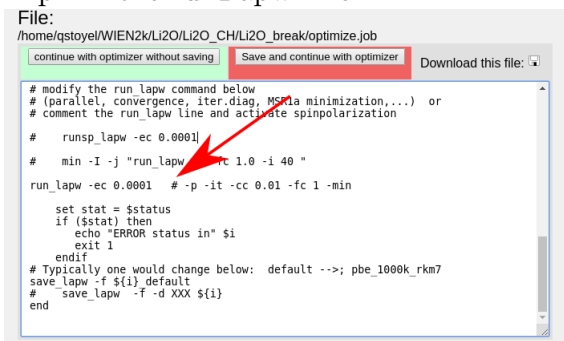
Ideally, you want to converge everything regarding your simulation. Typically this is cell parameters, k points and Rkmax. The first step is to just make sure the calculation converges, running it with a small Rkmax and few kpoints and making sure it finishes.

3.1 Cell parameters:

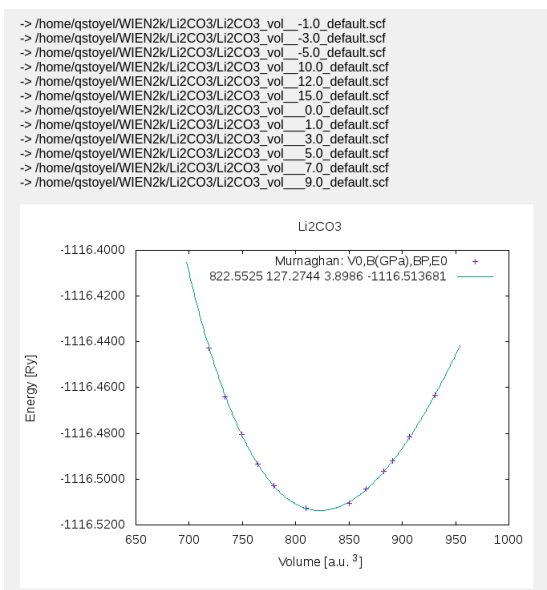
This process works best from W2Web, as described in the tutorials and the muffin tins from setrmt reduced by a healthy percentage ($\sim 5\text{-}10\%$) to avoid nn errors. As a number (5-11+) of calculations are run in this process, a low number of k points and RKMax values is ideal here. For Li_2CO_3 , I used 16 kpoints and an RKmax of 7. In the x “optimize” tab, choose what you want to optimize, the first option (volume) works well, unless you have suspicions otherwise. Enter a range of values of test volumes, see picture:



Also make sure to edit “optimize.job” to enable parallization by moving the “#” to after the “-p” in the run_lapw line:



You can then “run optimize.job” from w2web. This job is okay to run in the background, so you can close the browser and run it overnight as well. When it is done, plot the “Energy vs Volume,” which should give you something like this:



In the case of Li₂CO₃ and the original cif from materials project, quite a range of volume options were needed to locate the minimum. The graph indicates that a 3-4% increase should correspond to the optimized structure. To use this structure, search the case directory for all the struct files and rename the appropriate one to case.struct, for Li₂CO₃ this was “Li₂CO₃_vol___3.0.struct → Li₂CO₃.struct”. Alternatively, cut and paste the lattice parameters from this file into the w2web structgen tool. Finally, rerun init_lapw and readjust sphere sizes as necessary to account for the increase (or decrease) in cell size.

3.2 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. Generally k points are easier to converge, so start with them and then move on to RKMax. The procedure for converging both of these values is:

- set/increase kpoints or RKMax, either by re-running “x kgen” or editing case.in1 and case.in1.st.
- Run the scf cycle using “run_lapw -p -NI”, the NI flag means it will continue from where the previous calculation left off which saves time. I would still run your final choice from scratch though.
- Check the energy in case.scf. To do this, find it in “scf files” on w2web and use ctrl+f in your browser to search the file for “:ene” , which should appear in a line that looks like:

```
:ENE : ***** TOTAL ENERGY IN Ry = -1116.50733157
```

There will be one of these lines for each scf cycle, so find the last one in the document and note the energy.

- loop through the first 3 steps until the energy no longer changes significantly when you increase the kpoints/RKMax. A table is useful here to track these effects:

kpoints	RKMax	Energy
8	7.0	-1116.5073
16	7.0	-1116.4971
32	7.0	-1116.4975
32	8.0	-1116.5038
32	9.0	-1116.5045

4 TELNES3

Once the calculation is converged, ELNES can be calculated. There is a good deal of useful information and a complete description of the input file in the wien2k userguide, which

should solve most issues. Again, there is a large list of parameters that must be set for in order to obtain reasonable results. It is also worth converging Kpoints and RKMax against the spectra as well.

The majority of the important parameters need to be set in the **case.innes** file, which is easiest through w2web. Choose the right atom (in this case Li1) for the edge, and the right atomic numbers:

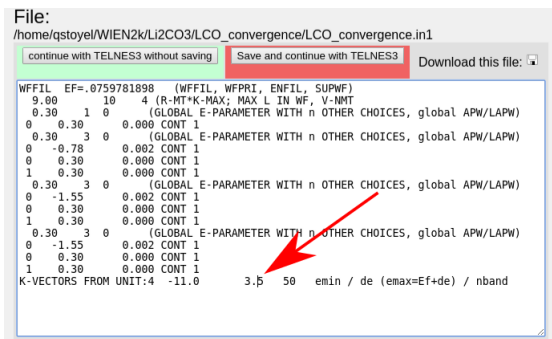
Edge	n	l
K	1	0
L1	2	0
L23	2	1
M45	3	2

Next, set the edge onset, edge values can be found at http://www.kayelaby.npl.co.uk/atomic_and_nuclear_physics/4_2/4_2_1.html as well as at a number of other locations. Set the beam energy to it's correct value, same goes for the collection and convergence angles, although TELNES is relatively robust to these: eg 5mrad produces very similar results to 1 mrad.

Set the energy grid to a large range of values, eg -20-50eV so you can see all of the features that might appear. The defaults for the remaining values should be fine.

In addition to the case.innes parameters, increase the number of kpoints, to at least double, or $10\times$, so that there is less doubt about this being converged, use x kgen for this.

Increase the upper energy limit in case.in1 from 1.5 to $\sim 2-3.5$, see picture. This value defines how many higher energy states are included in Ry ($1 \text{ Ry} \approx 13.6 \text{ eV}$, $1.5 \text{ Ry} \approx 20 \text{ eV}$). Therefore, to obtain the correct ELNES for features more than 20eV from the onset, this should be increased to match.



x lapw1 -p, x qtl -telnes -p, and x telnes3 can then all be run in succession. These are relatively slow commands, so to avoid waiting on them, run all three in sequence from the command line: `$ x lapw1 -p && x qtl -telnes -p && x telnes3`.

Once telnes3 is finished, edit case.inb and play with the spectrometer broadening on the last line, before clicking “x broadening” to generate the final spectra. To experiment

with different broadening values, just repeat these steps. Sometimes w2web gets stuck and stops re-broadening the spectra, in which case, just delete the case.broadspec file and run “x broadening” again. Plot the spectrum in a new tab (ctrl+click on “plot”) for easy comparison to new spectra without having to save it. Rerun all the TELNES steps, but with a higher/lower number of kpoints to confirm that that is not effecting the spectra. At this point you can also uncheck the “calculate and write DOS” checkboxes in case.innes, which allow you to rerun telnes without needing to rerun lapw1 each time (unless you change the k points/case.in1 file).

4.1 Monopole effects

Because the lithium 1S state is quite delocalized, it is susceptible to monopole effects. These cause artifacts in the spectra resulting from non orthogonal states at the muffin tin boundary. Fortunately, it is easy to check for them. Return to the case.innes file and change the interaction order to 0.

The screenshot shows the TELNES web interface. The title is "Lithium K edge in Li2CO3 no hole". The atom is set to "1: Li1". The edge is "use n and l (n=1 l=0)". The edge onset is 55.00 eV, and the beam energy is 300 keV. The energy grid is from -20.0000 eV to 50.0000 eV in steps of 0.0500 eV. The collection solid angle is 5.00 mrad, and the convergence solid angle is 1.87 mrad. The spectrometer broadening is 0.50 eV. The Q-mesh is NR=5, NT=2. In the "Advanced settings" section, the "Interaction order" is set to 0. The "Initialisation" section has checkboxes for "Calculate DOS", "write DOS", "Calculate rotation matrices", and "write rotation matrices", all of which are checked. The "Verbosity" is set to "basic", and "File headers" is set to "Write headers (default)". The "Interaction potential" is set to "relativistic (recommended)". The "Q-grid" is set to "U" (uniform). The "Final state selection rule" is set to "L=1 +/- 1 (default)". The "Extend potential beyond Rmt" checkbox is checked. The "Set Fermi energy manually" checkbox is also checked. The "Read core state wavefunction" and "Read final state wavefunctions" checkboxes are checked. The "Calculate DOS only" checkbox is checked. The "NBOT" is set to 200. A "save" button is at the bottom.

Telnes now only calculates the monopole component, look in case.outputtelnes to confirm this. Compare the monopole contribution to the full spectra, it should be much smaller ($\sim 100\times$). If it is not, set the interaction order to 1 to enforce dipole selection and use that for all future spectra.

5 Core Hole

To introduce a core hole, start an entirely new case, and copy only the struct file. Open a terminal in the new directory and run “x supercell.” Enter the struct file name, size of the supercell ($2\times 2\times 2$ is generally enough), no vacuum or shift. Open the new file (case_super.struct)

in vesta, and make sure it looks right. The most important part of this process is to somehow isolate a single lithium atom so only one core hole gets inserted into the cell:

VS

Inequivalent Atoms: 12			
Atom 1:	Li	Z=3.0	RMT=2.0000
Pos 1:	x=0.19885150	y=0.16104400	z=0.05090650
Atom 2:	Li	Z=3.0	RMT=2.0000
Pos 1:	x=0.80114850	y=0.33895600	z=0.05090650

Inequivalent Atoms: 4			
Atom 1:	Li	1	Z=3.0 RMT=2.0000
Pos 1:	x=0.19885150	y=0.16104400	z=0.05090650
Pos 2:	x=0.80114850	y=0.33895600	z=0.05090650
Pos 3:	x=0.80114850	y=0.83895600	z=0.94909350
Pos 4:	x=0.19885150	y=0.66104400	z=0.94909350

When the struct file is ready, initialize using the converged parameters (RMT, RKMax) from the single cell case, and decrease the k points according to the size of the supercell. After init_lapw, insert a core hole into the cell by editing case.inc and case.inm:

GNU nano 2.5.3	File: Li2C03 hole sc.inc	Modified
1 0.00	0 NUMBER OF ORBITALS (EXCLUDING SPIN), SHIFT, IPRINT	
1,-1,1	(N,KAPPA,OCCUP)	
1 0.00	0 NUMBER OF ORBITALS (EXCLUDING SPIN), SHIFT, IPRINT	
1,-1,2	(N,KAPPA,OCCUP)	
1 0.00	0 NUMBER OF ORBITALS (EXCLUDING SPIN), SHIFT, IPRINT	
1,-1,2	(N,KAPPA,OCCUP)	
1 0.00	0 NUMBER OF ORBITALS (EXCLUDING SPIN), SHIFT, IPRINT	
1,-1,2	(N,KAPPA,OCCUP)	
1 0.00	0 NUMBER OF ORBITALS (EXCLUDING SPIN), SHIFT, IPRINT	
1,-1,2	(N,KAPPA,OCCUP)	
1 0.00	0 NUMBER OF ORBITALS (EXCLUDING SPIN), SHIFT, IPRINT	
1,-1,2	(N,KAPPA,OCCUP)	
1 0.00	0 NUMBER OF ORBITALS (EXCLUDING SPIN), SHIFT, IPRINT	
1,-1,2	(N,KAPPA,OCCUP)	

In case.inc, a hole is inserted by changing the occupancy of an orbital. Each set of lines corresponds to the core states of each atom defined in the struct file. This is why it is important to be able to treat the lithium states as core states, otherwise it would not be possible to insert a hole. This is also why it is important to break the symmetry in the unit cell as if not this would insert holes into multiple atoms. In the case.inm file, the excited electrons need to be added to the background charge:

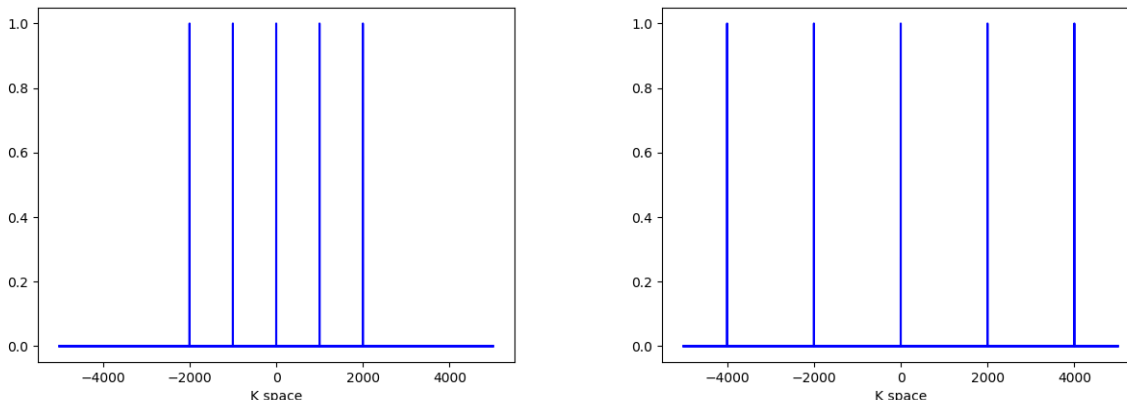
GNU nano 2.5.3	File: Li2C03 hole sc.inm
MSR1	-1.00 YES (BROYD/PRATT, BG charge (-1 for core hole), norm)
0.20	mixing FACTOR for BROYD/PRATT scheme
1.00 1.00	PW and CLM-scaling factors
9999 8	idum, HISTORY

The value is negative as this operation adds charge, not electron number. If for whatever reason, symmetry cannot be broken in the unit cell, this value would be increased according to the multiplicity of the excited atom. Don't do this though, as it defies the point of using a supercell. Once the background charge is added, execute run_lapw -p.

6 Common Errors/Issues Encountered in Wien2k

6.1 Setting RMT/RKMax

The RKMax value is a little obscure and takes some getting used to. It is defined as the Muffin Tin Radii \times Maximum K point vector. The maximum k point corresponds to the highest frequency plane wave used in the calculation.



Two case, both with 5 kpoints and different RKMax's. The plot on the left depicts a smaller RKMax relative to the one on the right.

Increasing RKMax allows for higher frequency (more precise) plane waves to be used in the basis set. These high frequency terms are however more computationally costly (CPU requirements scale as RKMax^3) and less essential for describing large features. This is why the RKMax includes the Muffin Tin radius: large atoms only require lower frequency plane waves. Each atom has an effective RKMax which is relative to the ratio of its radius and that of the smallest atom. Choosing very different muffin tin radii for different elements leads to issues as it becomes more difficult to align the surface features of each atom.

Increasing RKMax arbitrarily does not solve the problem either, as: 1. “approximate numerical linear dependency” occurs and 2. Calculations become prohibitively expensive. RKMax should at most be between 9-10.

6.2 NN in Optimization

Crashes the first scf cycle almost immediately, due to overlapping muffin tins resulting from a decreased cell size. Solution: decrease all muffin tin sizes before running “x optimize”.

6.3 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.

```
File Edit View Search Terminal Help
GNU nano 2.5.3 File: Li2C03.in2

TOT (TOT, FOR, QTL, EFG, FERMI)
-12.00 56.00 0.50 0.05 1 EMIN, NE, ESEPERMIN, ESEPER0, iqtlsave
TETRA 0.000 (GAUSS, ROOT, TEMP, TETRA, ALL eval)
0 0 1 0 1 1 1 1 2 0 2 1 -2 1 2 2 -2 2 3 0 3 1 -3 1 3 2 -3 2 3 3 -3 $
0 0 1 0 2 0 2 2 -2 2 3 0 3 2 -3 2 4 0 4 2 -4 2 4 4 -4 4 5 0 5 2 -5 $
0 0 1 0 2 0 2 2 -2 2 3 0 3 2 -3 2 4 0 4 2 -4 2 4 4 -4 4 5 0 5 2 -5 $
0 0 1 0 2 0 2 2 -2 2 3 0 3 2 -3 2 4 0 4 2 -4 2 4 4 -4 4 5 0 5 2 -5 $
0 0 1 0 1 1 -1 1 2 0 2 1 -2 1 2 2 -2 2 3 0 3 1 -3 1 3 2 -3 2 3 3 -3 $
16.00 GMAX
NOFILE FILE/NOFILE write recprlist

[ Read 9 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

- setrmt
- nn
- sgrouop
- symmery
- lstart