connect/setup of w2web environment

- login on to your PC with the **username/pw** you got during registration (remember the name, you may want to connect also from your laptop)
- only at the first login (or after advise due to an update):
  - *cp /home/nfs3/smr3929/setup.sh ~/.bash_profile*
  - *logout and login again*
- start w2web on the default port 7890 and a firefox browser using:
  - *w2web*
    - the first time you have to enter **user-id/pw** (use the same as for your login), for all other questions hit „enter“
  - *firefox nodename:port as indicated on the screen*
    - If you later need to kill (and restart) w2web (because of X-display or updates), use: kill_w2web; w2web
    - if your computer has been shutdown, restart w2web
- Start with the exercises.

Exercises:

- In the following you find some suggestions for exercises, which teach you various tasks one may perform with WIEN2k. Each exercise has a small query at the beginning and you should be able to answer them afterwards. We will discuss the exercises at the very end.

- New WIEN2k users should start with the first basic exercises (1-6), covering: structure generation, initialization, scf-cycle, bandstructure, DOS, electron density, structure optimization, supercell generation, surfaces, parallelization, spin polarization

- Later on, choose examples of your interest as there are probably more exercises than you can do here.
- Of course, with growing experience you can also try a personal example.

- Please note, that often "calculational parameters" are set to "minimal cpu-time" instead of "fully converged calculations".

- Do not use such small values for final results and publications without convergence checks !! Usually init –prec 2 gives converged results.

Content

- 1) Basic tasks with w2web
- 2) Volume optimization
- 3) Optimization of free atomic positions
- 4) ferromagnetism (Fe)
- 5) antiferromagnetism (Cr)
- 6) commandline interface, supercells, surfaces, structeditor
- 7) adsorption energies, free atoms
- 8) band gaps: mBJ, mGGA and hybrid-DFT
- 9) phase stability of CsCl with various DFT approximation
- 10) optimization of hexagonal or tetragonal (orthorhombic) structures
- 11) van der Waals crystals (PBE-D3/4, nonlocal-vdW)
- 12) Xspec (XANES)
- 13) DFT+U, spin-orbit coupling
- 14) exchange parameter J in NiO
- 15) optical properties
- 16) valence photoelectron spectra, renormalized DOS
- 17) phonons
- 18) parallelization
- 19) atoms-in-molecules
- 20) Hyperfine interactions (EFG, NMR)
- 21) boltztrap2
- 22) wannier functions, Berry charges, topology, fold2Bloch, mstar

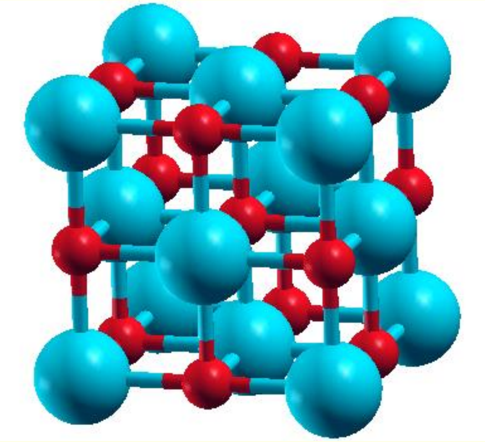
Exercise 1: Getting started:

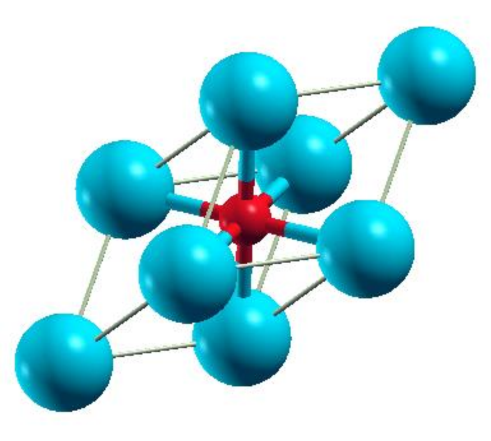
- Learn basic operations/tasks with the w2web interface of WIEN2k.
- (structure generation, initialization, scf cycle, DOS, electron density with Xcrysdens, bandstructure)

- Questions:
  - *How many iterations did you need for scf ?*
  - *How long takes a single scf-cycle ?*
  - *How many k-points did you actually use in the FBZ and IBZ ?*
  - *What are your RMTs and RKMAX ?*
  - *Which states dominate at: -15 eV: -8 to -2.5 eV: -2.5 to EF:*
  - *EF to 2.5 eV: 2.5 to 6 eV:*
  - *At what energy are the semi-core Ti-3s and Ti-3p states ?*
  - *Is Ti positively or negatively charged ? Why ?*
  - *Has Ti a larger eg or t2g occupation ?*

Exercise 1: Getting started:

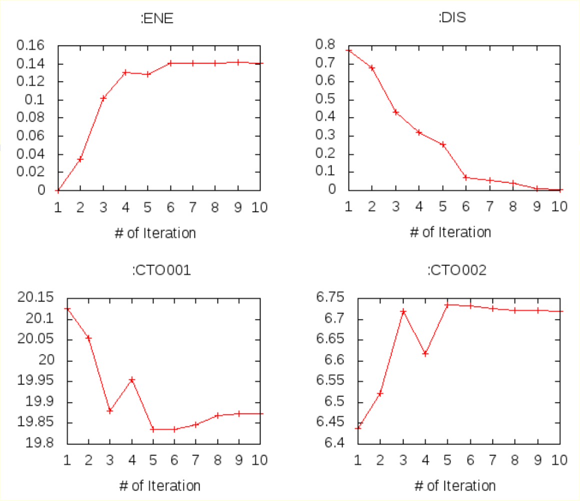
- i) After w2web has been started, connect with firefox to w2web (nodename:port)
- iv) Try the "quick-start" example for **TiN** (similar to TiC in the UG)
  - *create new session named "TiN", "create" and "select" the suggested directory.*
  - *Generate structure (a=4.235 Ang; reduce RMT by 1%)*
  - *view structure with Xcrysden (switch primitive / conventional cell)*



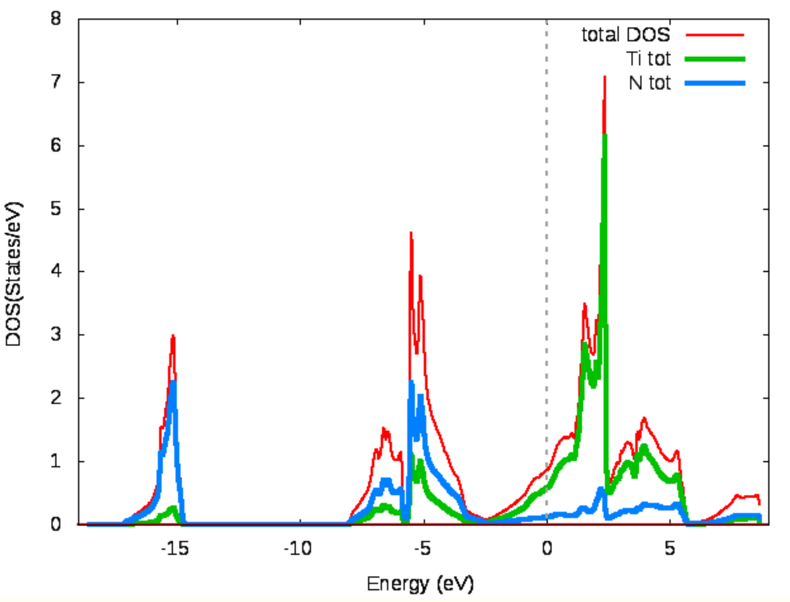


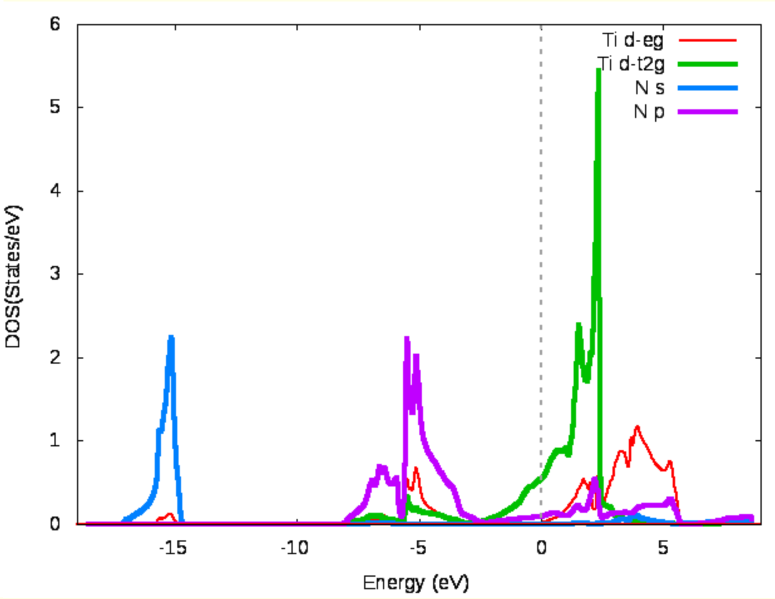
  - *initialize (fast mode; use defaults)*
  - *scf-cycle (run_lapw); use defaults; monitor "STDOUT" (reload in reverse order)*
  - *check "dayfile" (in utils)*

TiN continued



- *utilities: analyse*
  - (:ENE, :DIS, :CTO), graphically
- *utilities: save_lapw (use as save-name: "TiN_exp_pbe_prec1")*
- ***DOS**: (do the **necessary** steps)*
- *(plot 7 cases: total + Ti-tot + N-tot and Ti-eg + Ti-t2g + N-s + N-p)*

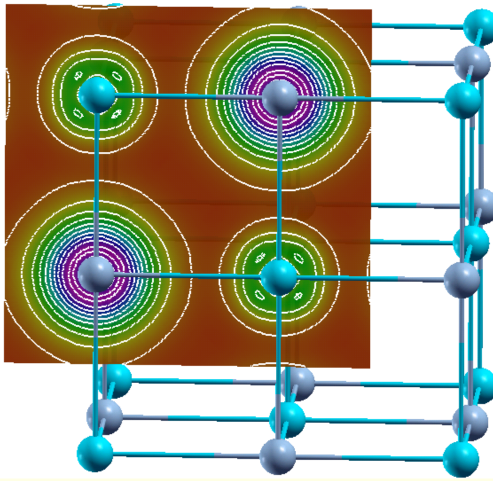




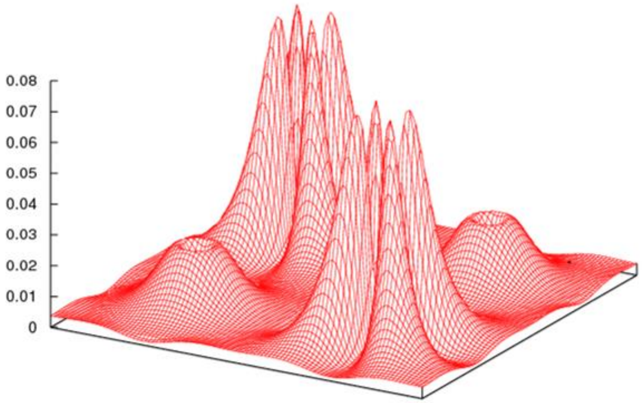
TiN continued …

- *electron density (use xcrysden to select the (100) plane), view it in xcrysden and rhoplot to "understand contour and 3D-plots")*
  - **valence** density (without semicore: check TiN.scf1 to find a EMIN which truncates the Ti-3s,3p states, rerun x lapw2 with this EMIN); compare the density around Ti with TiC (UG)
  - **difference** density (observe "charge transfer" and "t_{2g}-anisotropy" around Ti)
  - densities of the "**N-p**" and "**occupied Ti-d-band**" (get the corresponding E-intervals from DOS-plots (in Ry!) and use these energies in the "x lapw2" step; observe the e_g and t_{2g} asymmetry around Ti and the different N-p "weights", explain the chemical bonding

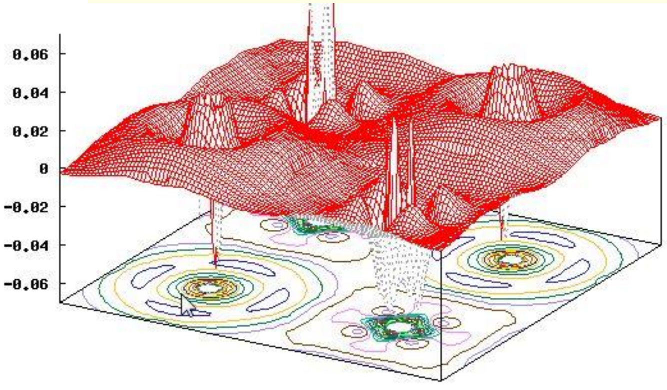
valence ρ



Ti-d band

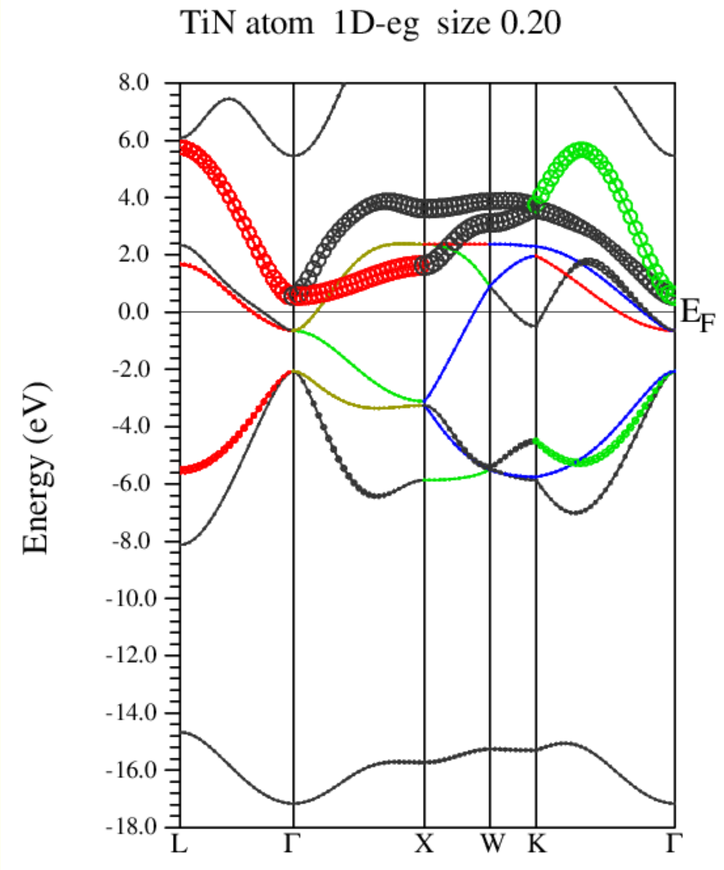


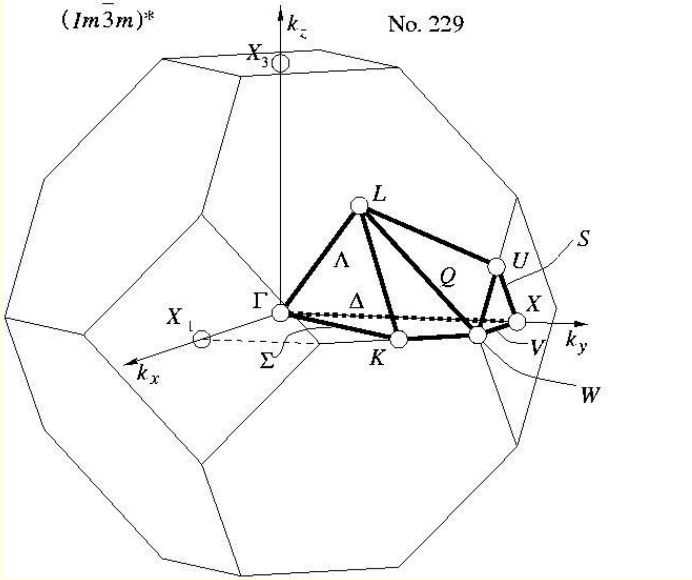
difference density



TiN continued

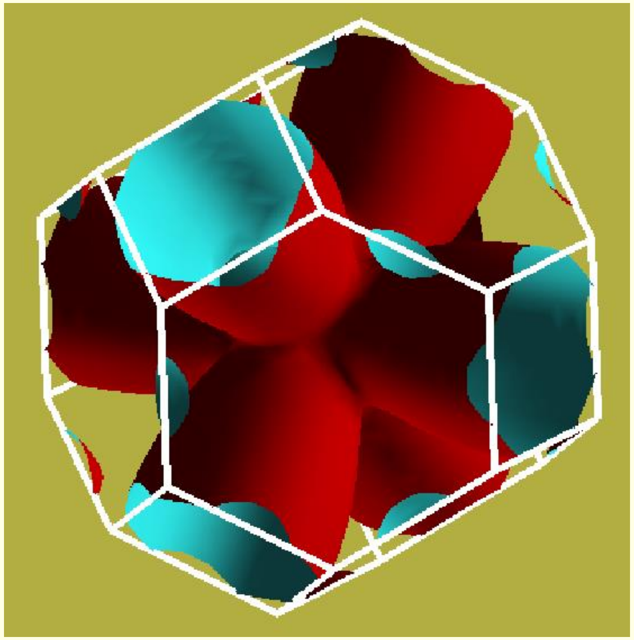
- bandstructure (along L-Gamma-X-W-K-Gamma with “character plotting”)
  - *use xcrysden (100 points, save as „TiN.klist_band“)*
  - *identify “t2g-” and “eg-” bands (fat band plots) by setting atom/column in TiN.insp*





TiN continued …

- *Fermi surfaces*
  - open a terminal, change into the TiN directory and issue:
  - `xcrysden --wien_fermisurface .`
    - choose a good k-mesh (eg. 10000 points);
    - plot the FS for all bands (**9**, 10,11) which cross E_F and compare to band structure



Exercises 2: lattice parameter of TiC

- Volume optimization in WIEN2k
- learn how to check computational parameters for converged results (reusing prev. calculations)
- Questions: (the exp. lattice parameter is 4.328 Ang, complete the Table
  - *What prec do you need for 0.01 Ang accuracy, what for B0 with 1% ?*

prec	**0**	**1**	**2**	**3**
Rkmax/K-points				
time for scf / cycles				
sigma of fit				
a_0 (Ang)/B_0				

  - *What could we change to get better agreement with experiment ?*

- Note: different properties (here a_0, B_0) may require different parameters.
- Note: different types of atoms, insulator/metal and system size need different parameters:
  - *H: RKmax > 2.5; sp-elements: RKmax > 6; d-elements: RKmax > 7; f-elements: RKmax > 8; (see our faq-page)*
  - *1 atom/cell, insulator/metal: 100-1000 / 1000-10000 k-points or more*
  - *For N atoms/cell you can reduce the k-mesh by a factor N*
- Note: For a good B_0 you need 7 volumes around V_0 with delta-V 1%

Exercises 2: lattice parameter of TiC

- TiC (fcc, **a=4.328** Ang, **setrmt 4%**)

- a) initialize in fast mode with **LDA, -prec 0** (on purpose bad values!), click on Utils/edit .machines and save (2 localhosts)
- b) run x optimize: specify
  - *run command: run_lapw –ec 0.00001 –cc 0.001 -p*
  - *save-filename: lda_prec0*
  - *volume opt. and generate 6 structures (-12, -9, -6, -3, 0, 3% change) (because of LDA we expect 1-2% smaller lattice parameter (3-8% in volume) than experiment)*
- c) run optimize.job, plot the results (using *lda_prec0)

- d) init –prec 1, LDA, -nodstart
- e) edit "optimize.job": **Uncomment** the "**cp line**" and **"comment clmextrapol"**, change the save statement:
  - *cp ${i}_lda_prec0.clmsum TiC.clmsum # Previously converged densities may save a lot of scf cycles.*
  - *# clmextrapol ...*
  - *save_lapw ${i}_lda_prec1*
- f) repeat step c) (plot the results for "*lda_prec1")

- g) repeat steps d-f for prec 2 and 3

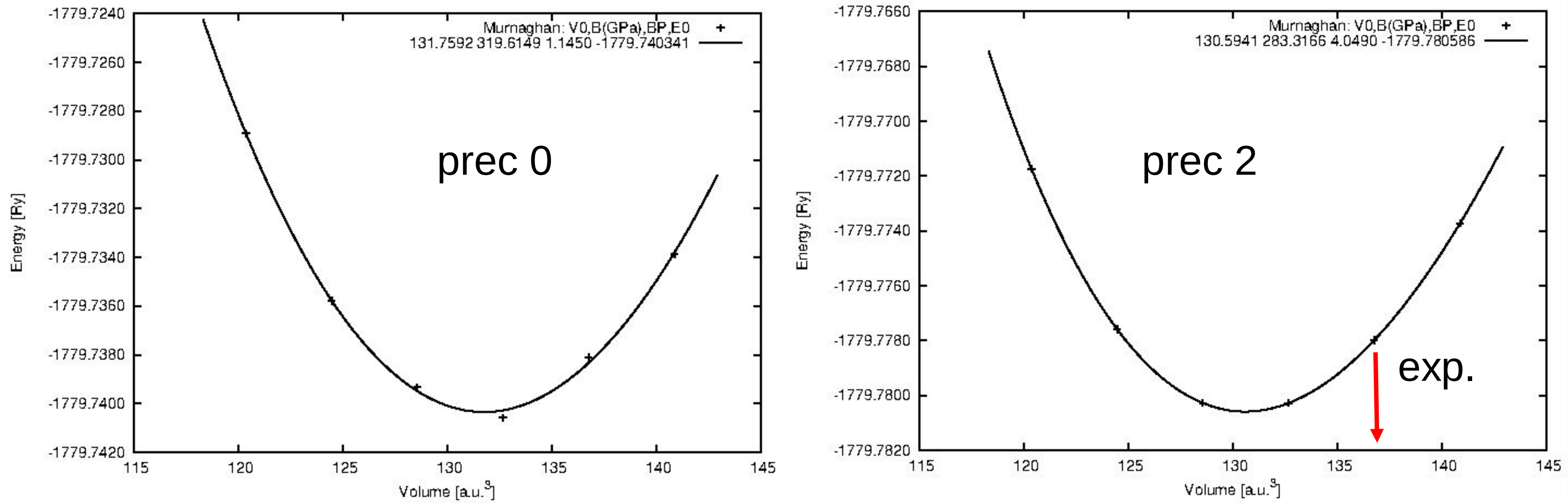
Volume optimization for TiC

- Open a terminal, change into the TiC directory (cd ~/WIEN2k/TiC)

- grepline :ene '*.scf' 1 # observe the "WARNING" in some cases
- grepline :WAR '*prec3.scf' 2 # QTL-B warnings for Ti-d (atom 1, l=2)
- grepline :e2_0001 '*prec3.scf' 1 # the E-param is jumping for 2 cases, because it could find E-top and thus sets E-d into the middle of the (mostly unoccupied) d-band. HDLOs (prec 2 and 3) compensate this. This is in part the reason for large sigma and changes in B_0

- grepline :rkm '*0.0*.scf' 1 # observe the increase of NMAT, why is the reported Rkmax smaller than what is in the corresponding case.in1 file ?

- grepline :ene '*0.0*scf' 1 # observe the convergence of the energy with RKmax



Exercise 3: optimization of positions in Mg(OH)$_2$

- **learn how to optimize internal atomic positions ("structure optimization").** In principle this should be done in every case, where forces are above ~5 mRy/bohr.
- Often, atomic positions from DFT are more accurate than experiment (powder XRD, light atoms), while lattice parameters are certainly not.
- learn how to use the UG for "arrows" plots

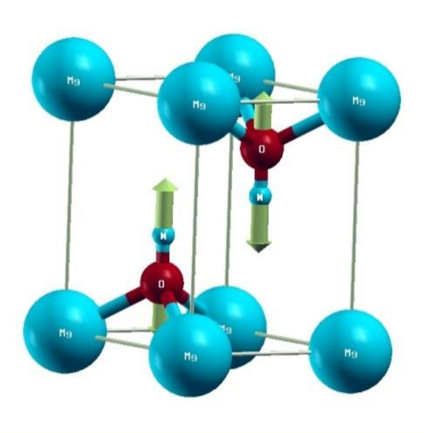
- questions:
  - *what was your Rkmax and # of k-points ?*
  - *what are the initial and final (relaxed) forces ?*
  - *how many iterations did you need to optimize positions?*
  - *what are the final z-coordinates for O: and H: atoms?*
  - *what are the relaxed O-H and Mg-O distances ?*

Exercise 3: optimization of positions in $Mg(OH)_2$

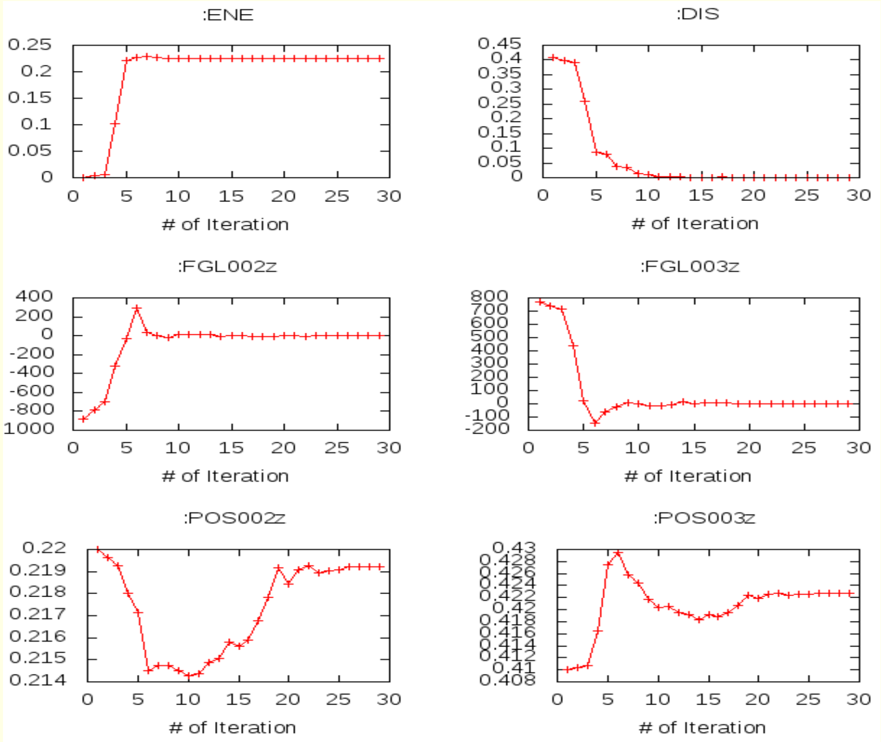
- P-3m1 (164), a=b=3.15 c=4.77 Å γ=120°; Mg(0,0,0)   O(1/3,2/3,**0.22**) H(1/3,2/3,**0.41**);   RMT: reduce by 7%
- init_lapw –prec 1n
- scf cycle with force convergence (-fc 1)
- edit **case.scf** (and find final :FGL002z, :FGL003z); save case_initial
- minimization using MSR1a (second option from „mini positions") or activate MSR1a button in „scf"): it executes:
  - ***run -min –fc 1 –cc 0.001 –ec 0.0001***
    - -min sets MSR1a in case.inm, (sometimes a crude scf cycle to come closer to „Born-Oppenheimer" surface is necessary (run -fc 10)
- analyze **case.scf** and find out how many scf cycles you needed
    - :ENE :FGL002z :POS002z :FGL003z :POS003z :DIS
- save_lapw case_final
- use the „arrows" utility to display

  initial forces and final relaxations

  (search the UG: „arrows")



$Mg(OH)_2$ continue



- **check convergence:**
  - *init_lapw –prec 2n -nodstart*
  - *run_lapw –fc 1 (check and compare your forces afterwards)*

Exercise 4: spin-polarized calculations

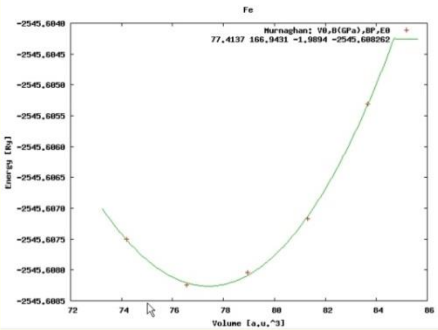
- Learn how to run magnetic calculations in WIEN2k
- Learn how magnetic moments and band width change with volume (pressure)

- questions:
  - *What is the corresponding Rkmax for "–prec 1"*
  - *What is the theoretical lattice parameter:*
  - *What is the Fe moment in PBE at the experimental and theoretical volume:*
  - *How do the moments and band width change with volume*

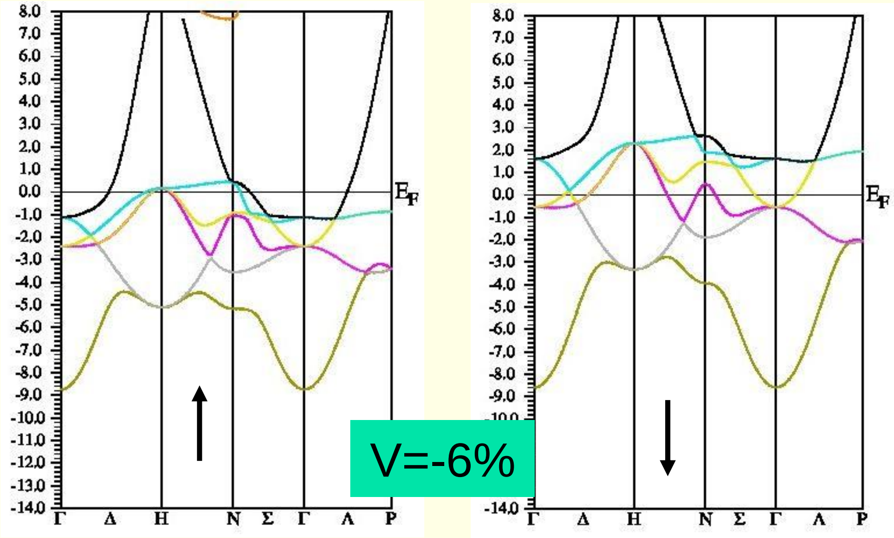
Volume	**-6**	**-3**	**0**	**3**	**6**
MMT					
$\Gamma-\Gamma$ (eV)					
H-H (eV)					

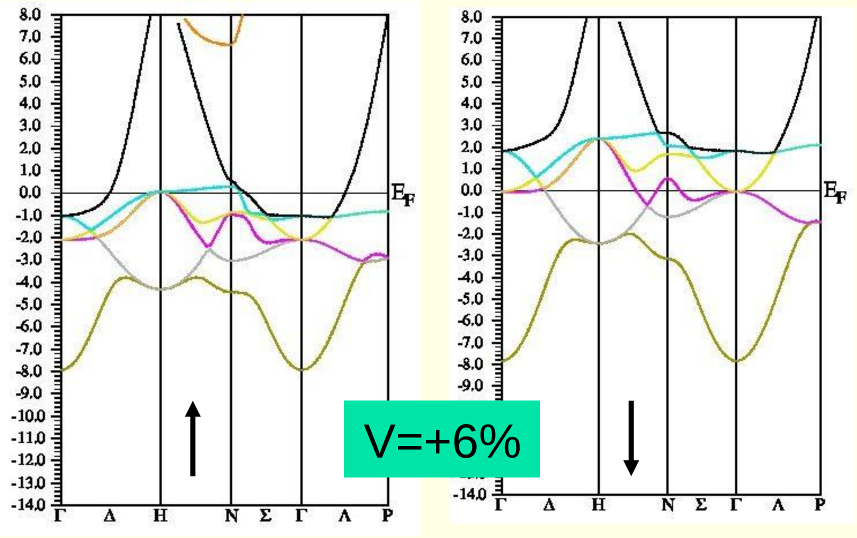
Exercise 4: spin-polarized calculations

- Magnetism: bcc Fe (a_0=2.86 Å), setrmt: 3%;
  - *init_lapw: spin-polarization, -prec 1*
  - *do a volume optimization (-6, -3, 0, 3, 6 %)* *(specify* ***runsp –ec 0.00001*** *and* ***prec1****)*
    - check equilibrium volume, :MMT as function of volume



  - *compare bandstructure and DOS for large/small volumes (use restore_lapw for desired volume; adjust EF in case.insp)*





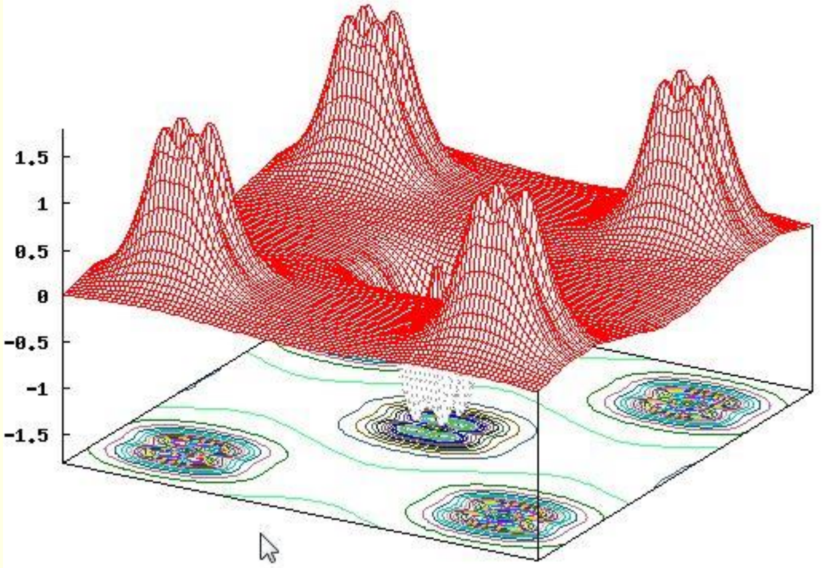
Exercise 5: antiferromagnetic calc.

- learn how to do antiferromagnetic calculations
- learn how to compare 2 different calculations (energies, moments)
- learn how to plot spin densities

- Questions:
  - *is FM or AFM Cr more stable? (:ENE)*
  - *is FM Cr stable at all ? check moments (MMI001:)*

Exercise 5: antiferromagnetic calc.

- Antiferromagnetism: bcc Cr (a_0=2.885 Å) (use **prec 1**; **-cc 0.001**)
  - *try 2 different calculations (in two different sessions (directories)):*
  - *ferromagnetic solution (bcc cell with 1 Cr)*
  - *antiferromagnetic calculation (P cell with Cr1 and Cr2 (at 0.5,0.5,0.5))*
    - do the **individual-mode** initialization procedure and choose up/dn for the two Cr atoms when creating case.inst using the "instgen" button (or run **instgen -ask** before **init_lapw -sp** on the command line)
    - for afminput your symmetry operation is "identity+(0.5,0.5,0.5)"
  - *plot spin-densities in the (110) planes*
    - do both spins before plotting:
      - x lapw1 –up; x lapw1 –dn; x lapw2 –up; x lapw2 -dn
    - observe "spatial localization"
    - t_{2g}-asymmetry



Exercise 6: Creation of supercells

- Learn how to run WIEN2k from the command line
- learn how to create simple (x supercell) and complicated (structure editor with octave) supercells of various sizes

- questions:
  - *MgO supercells: how many **non-equivalent** atoms do you have after x supercell and in the final super.struct ? (16: /); (32: /); (64: /); (128: /)*
  - *What has nn/sgroup done ?*
  - *MgO (001) surface:*
    - What has sgroup done ??
    - How many **total** (/) and **non-equivalent** (/) atoms and how many **atoms/layer** (/) do you have **before/after** sgroup ?
    - Do you have inversion symmetry ?
    - check the forces of unrelaxed scf-file. Which atoms will relax in which direction (into or out of the surface) ?
    - How much have the surface and sub-surface atoms finally relaxed ?

    - Where would you add a Pt atom ?

Exercise 6: Creation of supercells

- These exercises should be done WITHOUT w2web in a terminal window !
- **creation of basic structure: MgO**
- cd ~/WIEN2k; mkdir MgO; cd MgO
- makestruct (and type in the following information). It creates **init.struct**
  - *MgO: lattice type: **F**, a= **8.051** bohr (theoretical a_0 with PBE. Surfaces should be done with theor. a_0)*
  - *Mg (0,0,0), O (0.5,0.5, 0.5); 3% reduction of RMTs*
- cp init.struct MgO.struct
- view the structure using: xcrysden --wien_struct init.struct
- **16-atom supercell**
- mkdir super-16; cd super-16; cp ../MgO.struct super-16.struct; x supercell (use **super-16.struct**, select **2x2x2** and **F-cell):**
- cp super-16_super.struct super-16.struct
- edit super-16.struct and mark first Mg atom as "**Mg1**"
- x nn and if :WARNINGs appear do the next line:
  - *cp super-16.struct_nn super-16.struct; and repeat the "x nn" step above*
- x sgroup and view super-16.outputsgroup (no errors, but gives you a spacegroup)
  - *view the structure with xcrysden. (xcrysden --wien_struct super-16.struct) .Now you would be ready to run i**nit_lapw ....***

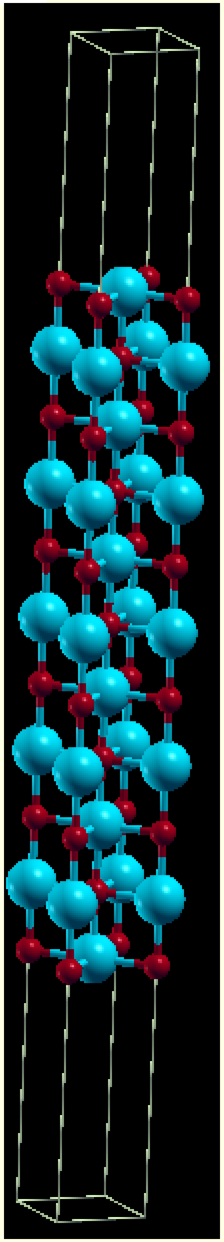
Exercise 6: Creation of supercells (cont.)

- **32, 64 and 128-atom supercells** (as above, but with B, P cell or 4x4x4-F)
- cd ..; mkdir super-32; cp MgO.struct super-32/super-32.struct; cd super-32
- do the steps as before ….

- Instead of labelling “Mg1”, one could also **remove** an atom (vacancy) or **replace** an atom by another (impurity).
- In one case, „forget‟ to label Mg1. What will sgroup do ??
- PS: Replacing atoms is better done in w2web, because this will also update the radial mesh (heavier atoms get smaller r0). (change **name** of atom AND **remove Z** !!)

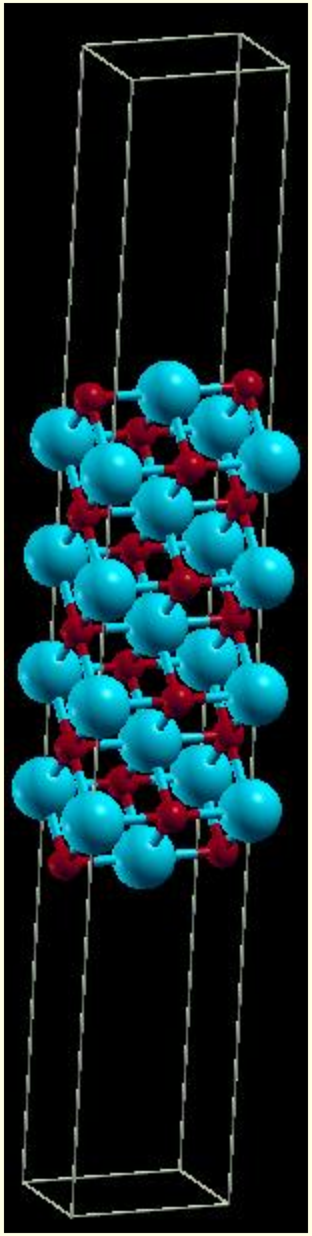
Exercise 6: Creation of surface slabs, relax it

- **(001) surface with 7 layers:**
- cd ~/WIEN2k/MgO; mkdir 001; cp MgO.struct 001; cd 001
- x supercell (use **MgO.struct**, 1x1x3, 30 bohr vacuum in z; repeat atom at 0:yes)
- cp MgO_super.struct 001.struct
- setrmt -r 3; cp 001.struct_setrmt 001.struct # reduce RMT
- xcrysden --wien_struct MgO_super.struct & (leave it open for comparison)
- x sgroup and view 001.outputsgroup (it creates a new (smaller) structure)
- cp 001.struct_sgroup 001.struct
- xcrysden --wien_struct 001.struct (Compare !
- init_lapw -prec 1n # 2D-BZ ! Which Fermi method has been selected ?
- run_lapw –fc 3 # observe the forces in scf-file
- save_lapw unrelaxed
- run_lapw –min –fc 1 # minimizes forces by optimizing positions
  - *Observe the progress of the minimization using: grep :FR 001.scf; grep :APOS001 001.scf; grep :APOS007 001.scf*
- save_lapw relaxed



Exercise 6: Creation of supercells (structeditor)

- **(110) surface of MgO with 9 layers:** (using the **structeditor)**
- cd ~/WIEN2k/MgO; mkdir 110-O; cp MgO.struct 110-O; cd 110-O
- octave (you can use repeat-key arrow-up !)
  - *helpstruct* *# list all possible commands*
  - *a=loadstruct(“MgO.struct”);*
  - *ac=makeconventional(a);* *# convert F into P cell*
  - *help makesurface* *# explains the syntax*
  - *sr=makesurface(ac, [1 1 0], 1, 20., 30.);*
  - *showstruct(sr)* *# check out the number of layers and repeat the **sr=makesurface** command with larger thickness until you get 9 layers. How do you get an O-atom at the origin ?*
  - *savestruct(sr,”super.struct”)*
  - *quit*
- xcrysden --wien_struct super.struct &
- x sgroup –f super and view super.outputsgroup
- cp super.struct_sgroup 110-O.struct
- xcrysden --wien_struct 110-O.struct
  - *what has sgroup done ?? how many total and non-equivalent atoms and how many **atoms/layer** do you have before/after sgroup ? Do you have inversion symmetry ?*



exercise 7: adsorption energies

- learn how to calculate adsorption energies (Pd/MgO(001))
- calculate „free" atoms (or molecules)
- make calculations with „consistent" Rkmax
- use „top" (exit with „q") to observe memory and cpu-usage

- questions:
- How many scf cycles did you need for relaxation:
- How much memory is used in lapw1, how many cores are used
- What are the relaxed Pd-O and O-Mg distances
- What happened to the O atom beneath the Pd ?
- Where are the Pd s,p and d bands in relation to EF and the MgO bands ?
- Where are the Mg and O-surface-atom bands in relation to the bulk bands ?
- How large is the adsorption energy ?
- What is the electronic configuration of the free Pd atom

- Note: A good calculation should use -prec 2n

Pd/MgO(001)

- make a directory ~/WIEN2k/MgO/Pd-001, copy 001.struct into the new directory
- add **2 equivalent** atoms manually (you can either edit the struct file, but this is much easier in w2web, since the struct file is position dependent !!) at a suitable starting position, eg. (0,0,+/-z) (2 atoms to keep inversion symmetry !!)
  - *where would you add two Pt atoms ?*
  - *at what distance (a bad starting position will require VERY long relaxation time and could force you to use bad RMTs) ?*
  - *increase the Pd RMT by about 10%   (why ?)*
- x nn
  - *check your struct file using xcrysden and Pd-001.outputnn (distances)*
- init_lapw –prec 1n; run_lapw –fc 10   # preconverge forces
- open another window and execute: top   # observe memory and cpu usage
- run_lapw –min –fc 1 –NI
- calculate the band structure (most easily in w2web) and identify the different bands

Pd atom

- make a directory ~/WIEN2k/MgO/Pd-atom, create a struct file: (FCC, 30 bohr, Pd at origin with RMT as in Pd/MgO(001), ignore the errors in setrmt)
- Checkout Rkmax of Pd-001.scf as well as all RMTs. The „consistent" Rkmax is:
  - *Rkmax(Pd) = Rkmax(Pd-001) * RMT(Pd) / RMT(O)*
- init_lapw -sp –rkmax xxx -numk 1   # 1 k-point, spin-polarization
- runsp_lapw –it   # iterative diagonalization

- calculate the adsorption energy
  - $E^{ads} = (E(Pd\text{-}001) - E(001) - 2 * E(Pd)) / 2$

- The Pd-001 structure could also serve as base for a bigger supercell (for instance 2x2x1) to simulate reduced "coverage" instead of a full monolayer.

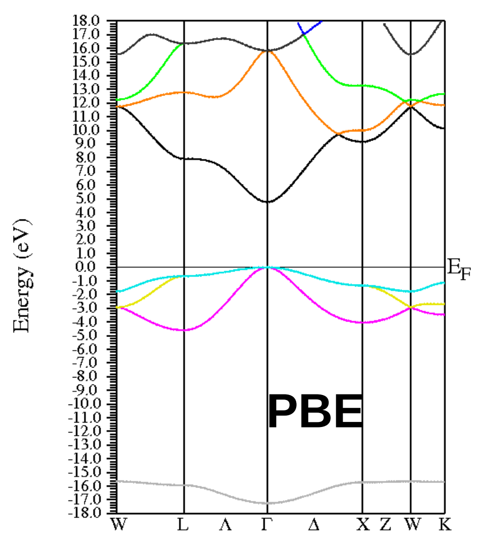
Exercise 8: band gaps of MgO

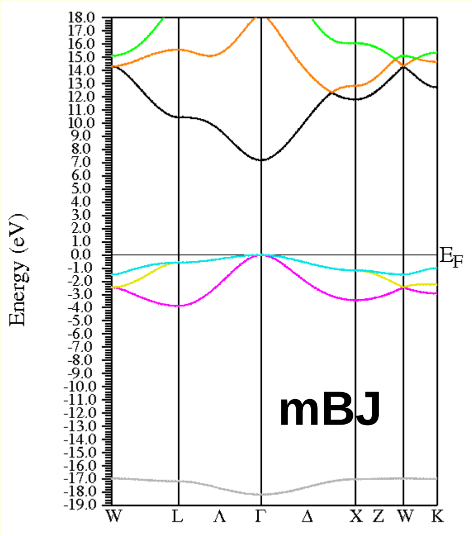
- Learn how to do TB-mBJ and m-GGA calculations
- Learn how to do hybrid-DFT calculations

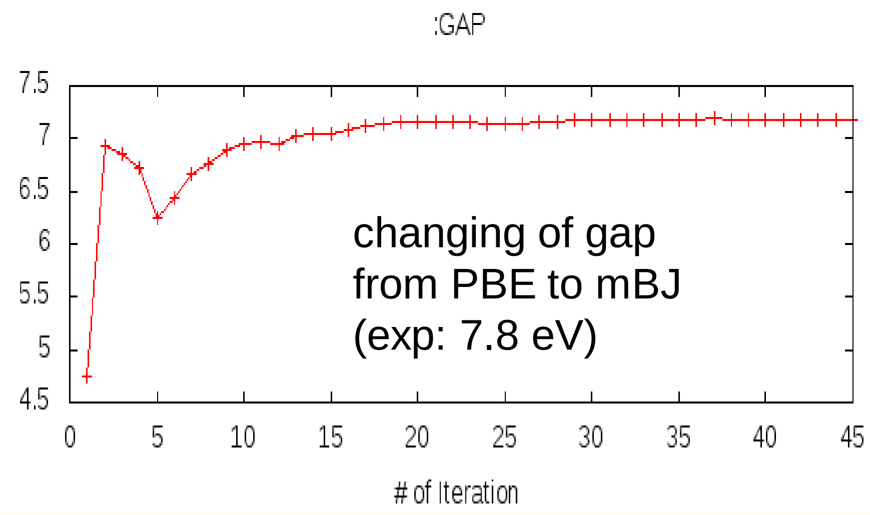
- Question:
  - *How many scf-iterations did you need for PBE, mBJ. KTBM_GAP and hybrid-DFT*
  - *What is the time for a hybrid calculation*
  - *What is the gap in PBE: mBJ: TASK KTBM_GAP: hybrid-DFT: (exp: 7.8 eV)*
  - *At what energy is the*
    O-2s band (at Γ):

Exercise 8: band gaps of MgO

- MgO (NaCl, a=7.96 bohr; default initialization for nonmetal; scf-cycle)
  - *PBE: check the gap (:GAP from "anaylysis"),*
    - plot a band structure in PBE (E-range from -19 to 18 eV)
  - *TB-mBJ:*
    - save the PBE calculation, execute:
      - init_mbj_lapw (in utils) „phase 1" of the initialization (see also in the UG 4.5.9)
      - run_lapw -NI -i 1
      - rm *.bro*
      - init_mbj_lapw „phase 2", use original mBJ parameters
    - run scf cycle
    - monitor the change of the :GAP
    - plot a band structure (fcc) and compare with PBE
    - save the mBJ calculation







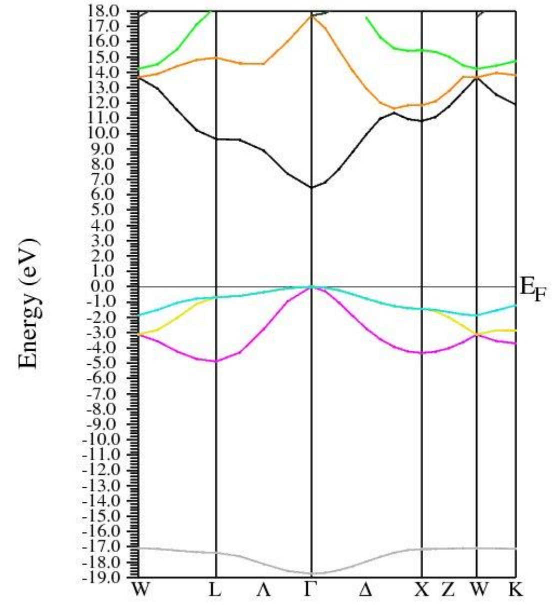
changing of gap
from PBE to mBJ
(exp: 7.8 eV)

Exercise 8: bandgap of MgO using meta-GGAs

- setup a meta-GGA calculation
  - *click on utils/init_mgga_lapw # follow the instructions and select TASK*
- run the scf cycle, check the gap, save
- do the same with KTBM_GAP

Exercise 8: continued …

- Perform a hybrid-DFT calculation using YS-PBE0
  - create a new case, perform a PBE calculation and save the results.
  - the setup for hybrid-calculations can be made in w2web (Utils/init_hf_lapw), or in a terminal-window using „init_hf_lapw". (More details are given in the UG 4.5.8)
    - Select NBAND=12 (case.inhf), why ??
    - and a 6x6x6 / 6x6x6 k-point mesh (no reduction)
  - scf cycle with **-hf -p** (insert 4 lines with **1:localhost** into **.machines**)
    - we do this in k-parallel since hybrids are expensive, for larger systems we could also use a „reduced" HF-k-mesh, see UG
  - monitor the change of the :GAP (only every 2nd value is from HF !), use „scfmonitor :gap" on the command line.
  - plot a band structure:
    - only the k-mesh selection can be done in w2web, then open a terminal and change into the proper directory
    - run_bandplothf_lapw -p
    - x spaghetti # creates case.insp
    - edit case.insp (insert E_F and increase the plotting energy range).
    - x spaghetti -hf -p



Exercise 9: influence of DFT approximation

- learn how to use the **command line interface**
- learn: use various DFT functionals, learn about the stress tensor, check accuracy

- compare NaCl and CsCl phases for CsCl ! (gap, a_0, energy difference) using PBEsol, PBE, non-scf and scf SCAN, KTBM-8 meta-GGAs.

	a_0^{CsCl}(Å)	**a_0^{NaCl}(Å)**	**$E^{CsCl} - E^{NaCl}$ (mRy)**	**gap (eV)**
PBESOL				
PBE				
Non-scf SCAN				---
SCAN				
KTBM_8				
experiment	4.12	---		~ 8.4

optimize CsCl using PBE in **CsCl** structure

- mkdir CsCl; cd CsCl; mkdir CsCl; mkdir NaCl; cd CsCl
- makestruct
  - *P lattice, a0= 4.12 Ang; Cs: (.0 0 0), Cl: (.5 .5 .5), 5% reduction*
- cp init.struct CsCl.struct
- init_lapw –prec 1n
- x optimize –job "run_lapw –ec 0.000001" –save pbe_prec1n
- ./optimize.job
- eplot –a pbe_prec1n # note the scattered plot, we have to increase accuracy
- cp CsCl_initial.struct CsCl.struct; init_lapw –prec 2n # will change RMT to 2.35
- x optimize –job "run_lapw –ec 0.000001" –save pbe_prec2 # we cannot use the previous struct files
- init_mgga # creates case.inm_tau for later mGGA calculations
- ./optimize.job
- eplot –a pbe_prec2
  - *note a_0, E_0 and gap at a_{exp}*

optimize CsCl using PBE in **NaCl** structure

- cd ../NaCl; makestruct   # since we do not know what a0 is, we estimate it as
  - *$a_{NaCl} \sim a_{CsCl}$*1.6   ($1.6^3 \sim 4$ - NaCl cube contains 4 fu)*
  - *F lattice, a0= 6.6 Ang (=12.47 bohr); Cs: (.0 0 0), Cl: (.5 .5 .5), 5% reduction*
- cp init.struct NaCl.struct
- we will use the stress tensor to estimate a better a_0 (non-relativistic !)
- mkdir stress; cd stress; cp ../NaCl.struct stress.struct
- init_lapw –prec 2n
- run_lapw –str 0.1; # this changes to NREL; tail *scf, since the pressure is large we increase a
- save 12.47; edit stress.struct and set a to 12.97; init –prec 2n; run_lapw –str 0.1
- save 12.97; edit stress.struct and set a to 13.47; init –prec 2n; run_lapw –str 0.1
  - *from the pressure you should get a good estimate of the **NREL** a_0*
- cd .. ; edit NaCl.struct and put your estimated a_0; init –prec 2n
- init_mgga   # creates case.inm_tau for later mGGA calculations
- x optimize –job "run_lapw –ec 0.000001" –save pbe_prec2n
- ./optimize.job
- eplot –a pbe_prec2n   # compare E_0 to that of the CsCl structure

calculate both phases with other DFT approx.

- PBESOL:
  - *change in case.in0   XC_PBE to   XC_PBESOL*
  - *change the save name in optimize.job to pbesol_prec2n*
  - *./optimize.job;   eplot –a pbesol_prec2n*
- non-scf meta-GGA calculation: (if you did init_mgga before)
  - *change in case.in0   XC_PBESOL to   XC_SCAN*
  - *change the save name in optimize.job to nonscf-scan_prec2n*
  - *./optimize.job;   eplot –a nonscf-scan_prec2n*

- meta-GGA calculations:
  - *init_mgga   # select SCAN;*
  - *edit optimize.job*
    - change the save name in optimize.job to scan_prec2n
    - comment the „clmextrapol-line"
    - uncomment and change:
      - cp ${i}_prec2n_pbe.clmsum CsCl.clmsum
      - cp ${i}_prec2n_pbe.tausum CsCl.tausum
- ./optimize.job
- eplot –a scan_prec2
- Do the same with the KTBM_8 meta-GGA

Exercise 10: optimize a+c (hex or tet lattices)

- learn how to use 2Doptimize_lapw and optimize_abc_lapw
- lern how to run a job in background

- Optimize lattice parameters a and c and internal position x for rhombohedral Sb

	a (bohr)	c (bohr)	x
optimize_abc			
2Doptimize			
experiment	8.127	21.207	0.2336

rhombohedral Sb

- mkdir Sb; cd Sb; makestruct
  - *R-3m; 4.30070,11.22200; Sb: (.2336,.2336,.2336)*
  - *set RMT to 2.35*
- mkdir optimize_abc; cp init.struct optimize_abc/optimize_abc.struct; cd optimize_abc
- init –prec 2n
- optimize_abc -h   # checkout the parameters
- since the job may take longer, we send it in the background and detach from terminal using **nohup ... &**. It will continue even after logout.
- nohup optimize_abc -d 1 -j "run_lapw -I -fc 1. -min -ec 0.000005" &
  - *the output from the screen is sent to* ***nohup.out****, which you can always monitor. Also use* ***top*** *to „see" the running jobs .*
  - *note the final a, c and x-position of Sb*
- mkdir pbe; mv optimize_abc_save step* pbe
  - *you could now select a different XC-functional or change accuracy setup and run optimize_abc again*

alternative optimization using 2Doptimize

- mkdir 2D; cp init.struct 2D/2D.struct; cd 2D
- init –prec 2n
- set2D_lapw
  - *choose defaults; edit 2Doptimze.job (change savename and run command)*
- nohup 2Doptimize.job &
- ana2D_lapw
- for a better determination of the position x, you should rerun run_lapw –min for the fitted lattice parameters

Exercise 11: van der Waal crystals

- optimize h-BN using various approximations (PBE, PBE-D3/4, nl-VDW, R2SCAN)
- get structure from cif-file; (a_0=4.732 c_0=12.583)
- learn how to optimize a 2D-vdW crystal, DFT-D3/4, nonlocal-vdW, scripting, gnuplot

- Questions:
- what is the energy scale for a and c variations

	a_0 (bohr)	**c_0 (bohr)**
PBE		
R2SCAN		
PBE-D3		
PBE-D4		
rev-vdw-DF2		

- PS: For layer binding you would need to run BN at large c

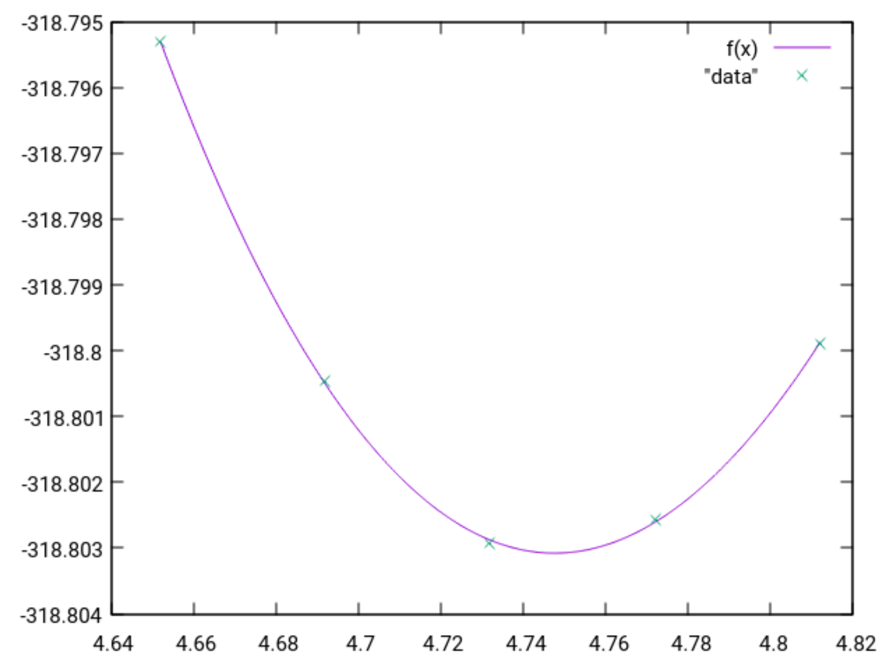
BN: optimize a

- mkdir BN; Download the cif file http://www.wien2k.at/Depository/BN-ICSD_CollCode241875.cif and copy it into BN
- x cif2struct –f BN-ICSD_CollCode241875; cp BN-ICSD_CollCode241875.struct BN.struct
- setrmt –r 3; cp BN.struct_setrmt BN.struct
- # optimize a, create your own script (if you know bash, you can also create a bash script):
- mkdir a; cp BN.struct a/a.struct; cd a; init –prec 1n
- vi job and insert:
    - *#!/bin/csh -f*
    - *x dstart -super* *# creates superposed density for current struct file*
    - *foreach i (4.652 4.692 4.732 4.772 4.812)*
    - *cp $i.struct a.struct*
    - *clmextrapol_lapw* *# extrapolates density to new struct file*
    - *run_lapw -cc 0.00001 -ec 0.000001*
    - *save ${i}_pbe_prec1*
    - *end*

- create 5 struct files 4.652.struct , … , and change a and b to the corresponding values
- chmod +x job; job
- grepline :ene ' *pbe_prec1.scf ' 1
- grepline :ene ' *pbe_prec1.scf ' 1 -s | cut -c1-5,66- > data

fit and plot via gnuplot

- vi fitparam
    - *a0=4.74*
    - *e0=-318.8*
    - *a1=1*
    - *a2=1*
- gnuplot
    - *f(x)=e0+a1*(x-a0)**2 + a2*(x-a0)**3*
    - *fit f(x) 'data' via 'fitparam'* *# note the fitted a0*
    - *plot 'data'*
    - *replot f(x)*
    - *quit*



optimize c (with optimal a_0)

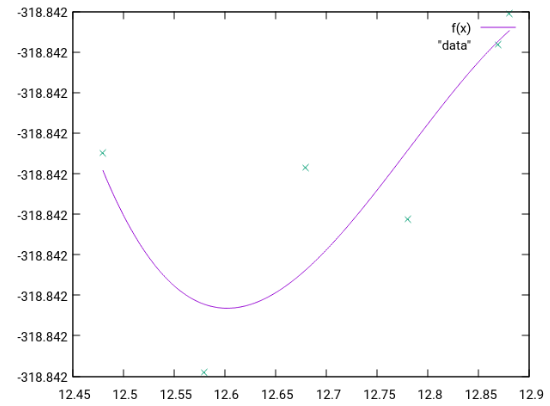
- mkdir c; cp job c; cp a.struct c/c.struct; cd c
- edit c.struct and put optimal a,b; cp c.struct 12.58.struct
- create 12.48, 12.68, 12.78 and 12.88.struct using
- sed -e "s/12.58/12.48/“ c.struct > 12.48.struct; …
- edit job and adapt it
- init –nometal
- ./job
- analyse, fit and plot similar as for a_0

repeat for PBE-D3/D4

- *cd ../a; x dftd3; x dftd4 # create necessary input*
- *edit job and:*
  - replace clmextrapol line by: cp ${i}_pbe_prec1.clmsum a.clmsum
  - add –dftd3 to run_lapw command
  - modify save-name
- *./job # and analyse as before*

- *cd ../c; x dftd3; x dftd4 # create necessary input*
- *edit job*
- *insert proper a_0 into the 5 structfiles, either with an editor or*
  - sed -i -e "s/4.748/4.738/g" 12.48.struct # and the same for the other 4 struct files
- *./job # and analyse as before*

- *What do you suggest to get better data ????*



nl-vdW with **rev-vdW-DF2**

- Check out the UG (nonlocal vdW functionals, sect. 4.5.16)
- cp $WIENROOT/SRC_templates/case.innlvdw a.innlvdw
- edit this file and insert: Kernel-type 1, Z_{ab}=-1.887
- edit a.in0 and modify the first line to
  - *TOT XC_GGA_X_B86_R EC_LDA VC_LDA*
- edit job and add -nlvdw to the run_lapw line

- Repeat the steps as before in the a and c directories

r2SCAN meta-GGA

- do the steps as before in the a and c directories
- restore the „first"_pbe_prec1 case to generate proper tau
- `init_mgga; run –i 1; rm *.bro*`
- `init_mgga` and select **r2SCAN**
- edit job (remove extra switch in run_lapw) and run the steps as before.

Exercise 12: Mg K-XAS in MgO

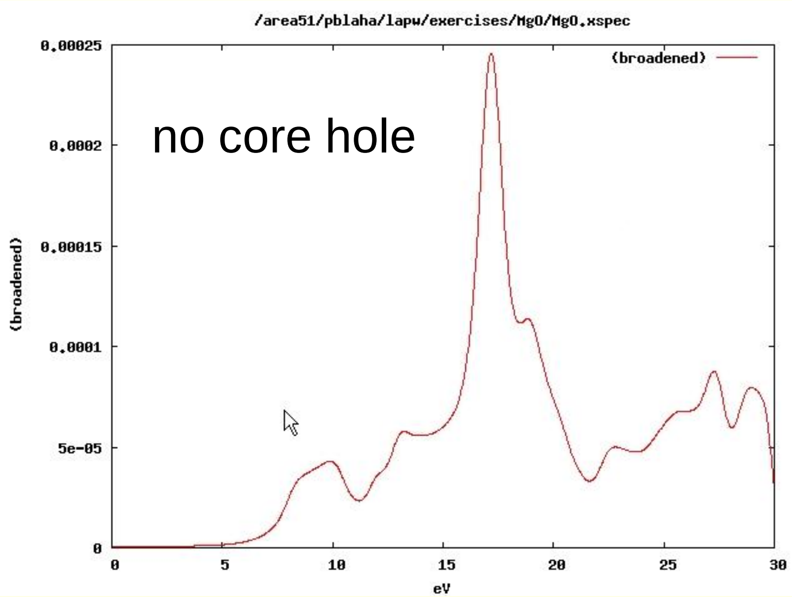
- Learn how to calculate XANES spectra (ELNES is very similar) in w2web
- Learn how to do a supercell calculation with a core-hole

- Questions:
  - *compare core-hole and no-core-hole calculations with experiment (see lecture notes)*

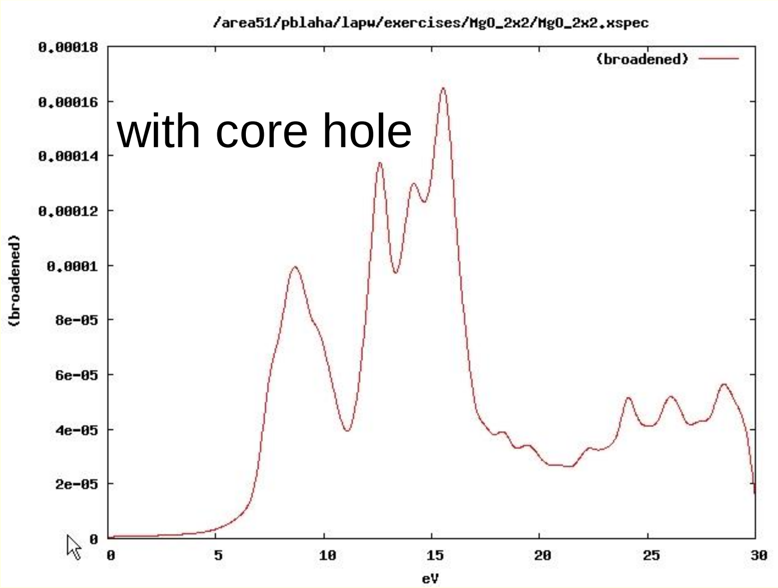
- PS: Usually larger supercells should be used.
- In metallic systems, half- or no corehole is often better (screening)

Exercise 12: Mg K-XAS in MgO

- MgO (NaCl structure, a=7.96 bohr; default initialization with –prec 1; scf-cycle)
  - *XSPEC task: larger EMAX in MgO.in1; select in MgO.inxs: Mg-K ABS from 0-30 eV, vary broadening)*
- Supercells: MgO 2x2x2 FCC-supercell for core-hole simulation
  - *create new "session", copy MgO.struct into new directory*
  - *x supercell; (specify proper struct-filename, 2x2x2, F-lattice)*
  - *cp supercell-struct file to correct name "case.struct"; "**label**" 1st atom (Mg → Mg1)*
  - *init_lapw*
  - *edit case.inc (remove a core electron from 1st atom)*
  - *edit case.in2 (add one valence electron)*
  - *run_lapw*
  - *edit case.in2 (remove extra valence electron)*
  - *XSPEC task for Mg-K XAS (see above)*



Mg-K XAS



Exercise 13: LDA+U calculations (command line)

- Learn how to do DFT+U calculations for correlated electrons
- Learn how to create a more complicated supercell for AFM structure
- Learn how to plot DOS (on the command line)
- Learn how to do a magnetic calculation with spin-orbit coupling
- Learn how to calculate orbital moments

- Questions:
  - *compare DOS (total, Ni1, Ni2, O) for all calculations*
  - *compare gaps (exp: 4eV),spin and orbital moments, occupied Ni-d band position*

scheme	**gap (eV)**	**spin m**	**orbital m**	**Ni-3d position**
PBE				
PBE+U				
PBE+U+so				
mBJ				

Exercise 13: LDA+U calculations (command line)

- NiO: NaCl structure, A-type AFM along [111]:
  - *This exercise should be done WITHOUT w2web in a terminal window !*
  - *cd ~/WIEN2k; mkdir NiO; cd NiO*
  - *makestruct* *# this creates init.struct*
    - NiO: F-lattice, a=b=c=4.195 Ang, α=β=γ=90°
    - Ni (0, 0, 0), O (0.5, 0 , 0)
    - setrmt 3%
- NiO supercell for A-type AFM along [111] (alternating ferromagnetic planes
  - *octave*
    - helpstruct # lists all structeditor commands
    - help loadstruct # gives help for specific command
    - s=loadstruct(“init.struct”) # load fcc NiO struct file
    - sc=makeprimitive(s); # convert FCC into primitive R cell
    - sr=makesupercell(sc,[1 1 0; 0 1 1; 1 0 1]); # create supercell (110 vectors)
    - showstruct(sr) # view the structure
    - savestruct(sr,“NiO.struct“)
    - quit

Exercise 13: LDA+U calculations

- *edit NiO.struct and label atoms „Ni1" and „Ni2" (use overwrite mode, don't „insert" 1 and 2 !!!) # this labels spin-up/dn Ni atoms*
- *x sgroup # determines SG, makes O atoms equivalent*
- *cp NiO.struct_sgroup NiO.struct # use the new struct file*
- *xcrysden --wien_struct NiO.struct # visualize and understand the structure*

  # switch conventional/primitive cell; R-cell with (a/√2; a/√2; 2 a √3)
- *instgen -ask # generates non-default NiO.inst: put Ni1: up; Ni2:dn; O: nm*
- *init_lapw -sp # initialize with all defaults (check init_lapw -h)*
- *runsp_lapw # scf-calc. with all defaults (PBE)*
- *save_lapw NiO_pbe # save the calculation*
- *x lapw2 -up -qtl; x lapw2 -dn -qtl # calculate partial charges for DOS*
- *configure_int -b total 1 tot,d 2 tot,d 3 tot,s,p end # configure which DOS to calc.,*
  - also quite covenient to use without batch mode
- *x tetra -up; x tetra -dn # calc. DOS*
- *dosplot2 -up # plot DOS*
- *save_lapw -dos NiO_pbe # save the DOS*

Exercise 13: LDA+U calculations

- GGA+U calculation
  - *init_orb -orb # generate default input files*
    - use U=5eV, J=0 for Ni-d; search the UG to understand case.inorb/indm
  - *runsp_lapw -orb #scf-calc. with PBE+U*
  - *save_lapw NiO_pbe+u*
  - *calculate and save the DOS as above ….*
- GGA+SO calculations (M=[111], without relativistic LO, Emax=5.0)
  - *initso_lapw*
    - select M=[111], no relativistic LO, Emax=5.0, spin-polarization and accept the new struct file.
  - *runsp -so -orb*
  - *x lapwdm -up -so # for orbital moment :orb001 in NiO.scfdmup*
  - *save_lapw NiO_pbe+u+so*
  - *calculate and save DOS*

Exercise 13: LDA+U calculations

- TB-mBJ calculation (without SO):
  - *restore NiO_pbe # restore pbe calculation*
  - *init_mbj_lapw # and follow the instructions on the screen*
  - *run the scf cycle and calculate the DOS*

NiO cont…

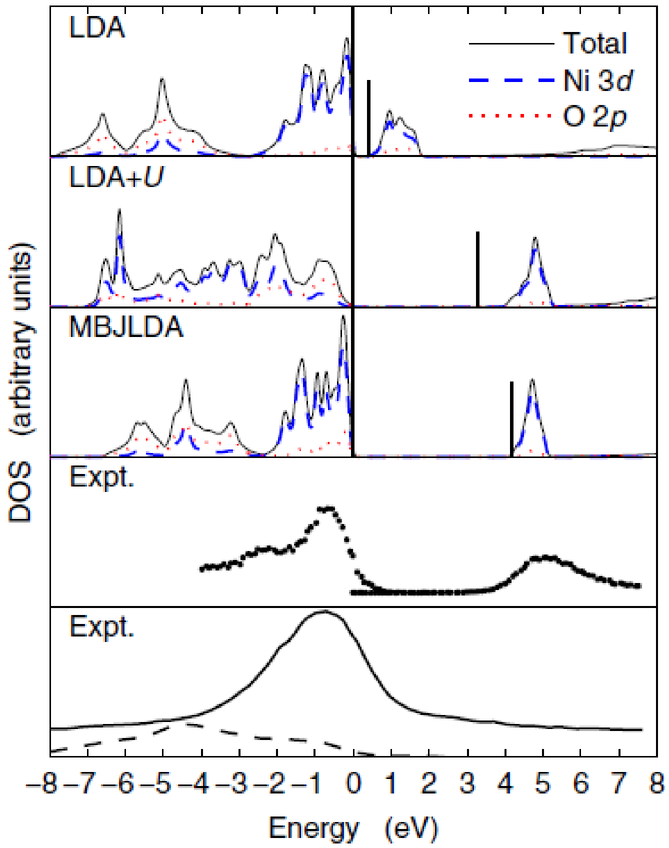
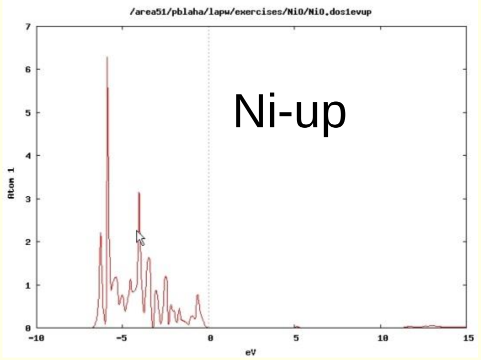
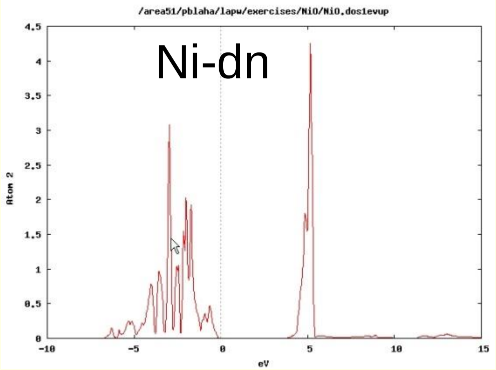


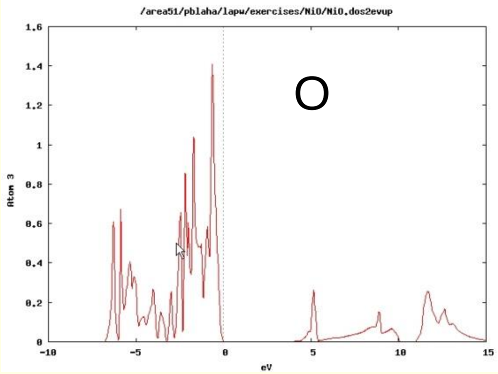
FIG. 2 (color online). DOS of NiO. The vertical bars indicate the end of the fundamental band gap which starts at $E = 0$ eV. The panels labeled "Expt." show photoelectron [25] (upper panel) and XES [33] [lower panel, Ni (solid line) and O (dashed line) spectra] measurements.

from Tran,Blaha, PRL 102, 226401 (2009)

GGA+U







Exercise 14: calculate the exchange parameter J in NiO

- Learn how to estimate the magnetic coupling J
- Learn 2 different approaches to extract J:
  - ○ (A) Energy-mapping analysis based on two ordered spin states (FM and AFM)
  - ○ (B) Energy-mapping analysis based on four ordered spin states (uu, dd, ud and du)

Ref. Dalton Trans., 2013, 42, 823

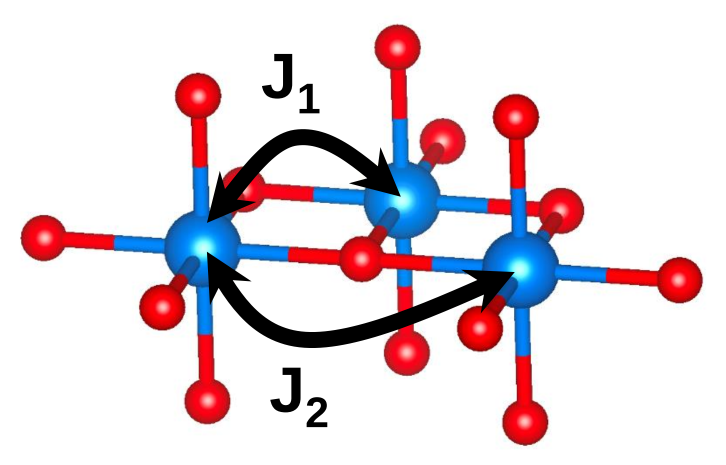
Spin hamiltonian: $\hat{H}_{spin} = \sum_{i<j} J_{ij}\vec{S}_i\vec{S}_j$

(A) $J_{ij} = (E_{FM} - E_{AFM})/2N$

with N: number of J_{ij}

(B) $J_{ij} = \dfrac{E_{uu} + E_{dd} - E_{ud} - E_{du}}{4S^2}$

with S: Total spin of Ni in NiO



J_1: nearest-neighbors exchange coupling

J_2: next nearest-neighbors exchange coupling

Exp. Values from neutron scattering:

J_1 = -1.37 meV and J_2 = +19.01 meV

FM: J < 0 **AFM: J > 0**

Exercise 14: calculate the exchange parameter J in NiO

- Questions:
  - *Compare gaps between FM and AFM calculations in the approach (A)*
  - *Estimate J_2 using the approach (A) using PBE and PBE+U*
  - *Estimate J_2 using the approach (B) using PBE+U and compare to (A)*

scheme	**J_2 (PBE) (meV)**	**J_2 (PBE+U) (meV)**
Method A		
Method B	-------------	

  - *Estimate J_1 using the approach (B) using PBE+U and compare J_1 and J_2 theoretical values to experimental ones.*

Exercise 14: Magnetic exchange (command line)

- NiO: Estimation of J_2 from approach A (FM magnetic order)
  - *This exercise should be done WITHOUT w2web in a terminal window !*
  - *The AFM calculation has been already done in PBE and PBE+U. Let's now do the FM calculation.*
  - *cd ~/WIEN2k; mkdir NiO_FM; cd NiO_FM*
  - *cp ../NiO/NiO.struct NiO_FM.struct # Use the same struct than AFM*
- GGA calculation
  - *instgen –ask # generates non default NiO.inst : put Ni1: up ; Ni2:up; O: nm*
  - *init_lapw –sp*
  - *run_lapw # scf calc . with all defaults (PBE)*
  - *save_lapw NiO_FM_pbe # save the calculation*
- GGA+U calculation
  - *init_orb -orb # generate default input files*
    - use U=5eV, J=0 for Ni-d; search the UG to understand case.inorb/indm
  - *runsp_lapw -orb #scf-calc. with PBE+U*
  - *save_lapw NiO_FM_pbe+u*

Exercise 14: Magnetic exchange (command line)

- NiO: Estimation of J_2 from approach B (creation of a supercell)
  - *This exercise should be done WITHOUT w2web in a terminal window !*
  - *We first generate the supercell allowing to isolate one Ni-Ni dimer for which we will calculate the magnetic exchange parameter J.*
  - *cd ~/WIEN2k; mkdir Mapping_NiO; cd Mapping_NiO; mkdir J2_uu; cd J2_uu*
  - *cp ../../NiO/init.struct .   # start from NaCl structure created in exercise 13*

  - *x supercell (use init.struct, 3x2x2, P-lattice)*
  - *cp init_super.struct J2_uu.struct*
  - *edit J2_uu.struct and label atoms „Ni1“ and „Ni2“(use overwrite mode, don‘t „insert“ 1 and 2 !!!)*
  - *x sgroup.   # Check in uu.struct_sgroup that the space group in now P4mm*
  - *cp J2_uu.struct_sgroup J2_uu.struct*
  - *instgen –ask # generates non default NiO.inst : put Ni: up; O: nm*

Exercise 14: Magnetic exchange (command line)

- NiO: J_2 from approach B (creation of the magnetic orders dd and du)
  - *cd ..; mkdir J2_dd; cd J2_dd # create the directory for **down-down magnetic order***
  - *cp ../J2_uu/J2_uu.struct J2_dd.struct*
  - *instgen –ask   # put Ni: up (except Ni1 :dn and Ni2 :dn); O: nm*

  - *cd ..; mkdir J2_du; cd J2_du # create the directory for **down-up magnetic order***
  - *cp ../J2_uu/J2_uu.struct J2_du.struct*
  - *instgen –ask   # put Ni: up (except Ni1 :dn); O: nm*

  - *Why is it not necessary to calculate ud magnetic order?*

  - *run these 3 calculations (uu, dd and du) using PBE+U as in exercise 13:*
  - *init_lapw –sp -nometal*
  - *init_orb –orb   (U=5 eV)*
  - *runsp_lapw –orb*

Exercise 14: Magnetic exchange (command line)

- NiO: Estimation of J_1 from approach B (creation of a supercell)
  - *Let’s redo the same approach to estimate J_1 now.*
  - *We first generate the supercell allowing to isolate the related Ni-Ni dimer.*
  - *cd ~/WIEN2k/Mapping_NiO; mkdir J1_uu; cd J1_uu*
  - *cp ../../NiO_AFM/init.struct .   # start from NaCl structure created in exercise 13*

  - *x supercell (use init.struct, 2x2x2, P-lattice)*
  - *cp init_super.struct J1_uu.struct*
  - *edit J1_uu.struct and label atom 1 „Ni1“ and atom 9 „Ni2“(use overwrite mode, don‘t „insert“ 1 and 2 !!!)*
  - *x sgroup.   # Check in uu.struct_sgroup that the space group in now Amm2*
  - *cp uu.struct_sgroup uu.struct*
  - *instgen –ask # generates non default NiO.inst : put Ni: up; O: nm*

Exercise 14: Magnetic exchange (command line)

- NiO: J_1 from approach B (creation of the magnetic orders dd and du)
  - *cd ..; mkdir J1_dd; cd J1_dd # create the directory for **down-down magnetic order***
  - *cp ../J1_uu/J1_uu.struct J1_dd.struct*
  - *instgen –ask # put Ni: up (except Ni1 :dn and Ni8 :dn); O: nm*

  - *cd ..; mkdir J1_du; cd J1_du # create the directory for **down-up magnetic order***
  - *cp ../J1_uu/J1_uu.struct J1_du.struct*
  - *instgen –ask # put Ni: up (except Ni1 :dn); O: nm*

  - *run these 3 calculations (uu, dd and du) using PBE+U as in exercise 13:*
  - *init_lapw –sp -nometal*
  - *init_orb –orb (U=5eV)*
  - *runsp_lapw –orb -ec 0.00001 # because of small E-differences*

Exercise 15: optical properties

- Learn how to calculate optical properties (in the single particle approx.)
- Consider k-point convergence and relativistic effects

- Question:
  - *How many k-points do you need to converge optics for Al:*
  - *What is the plasma frequency in Al ?*

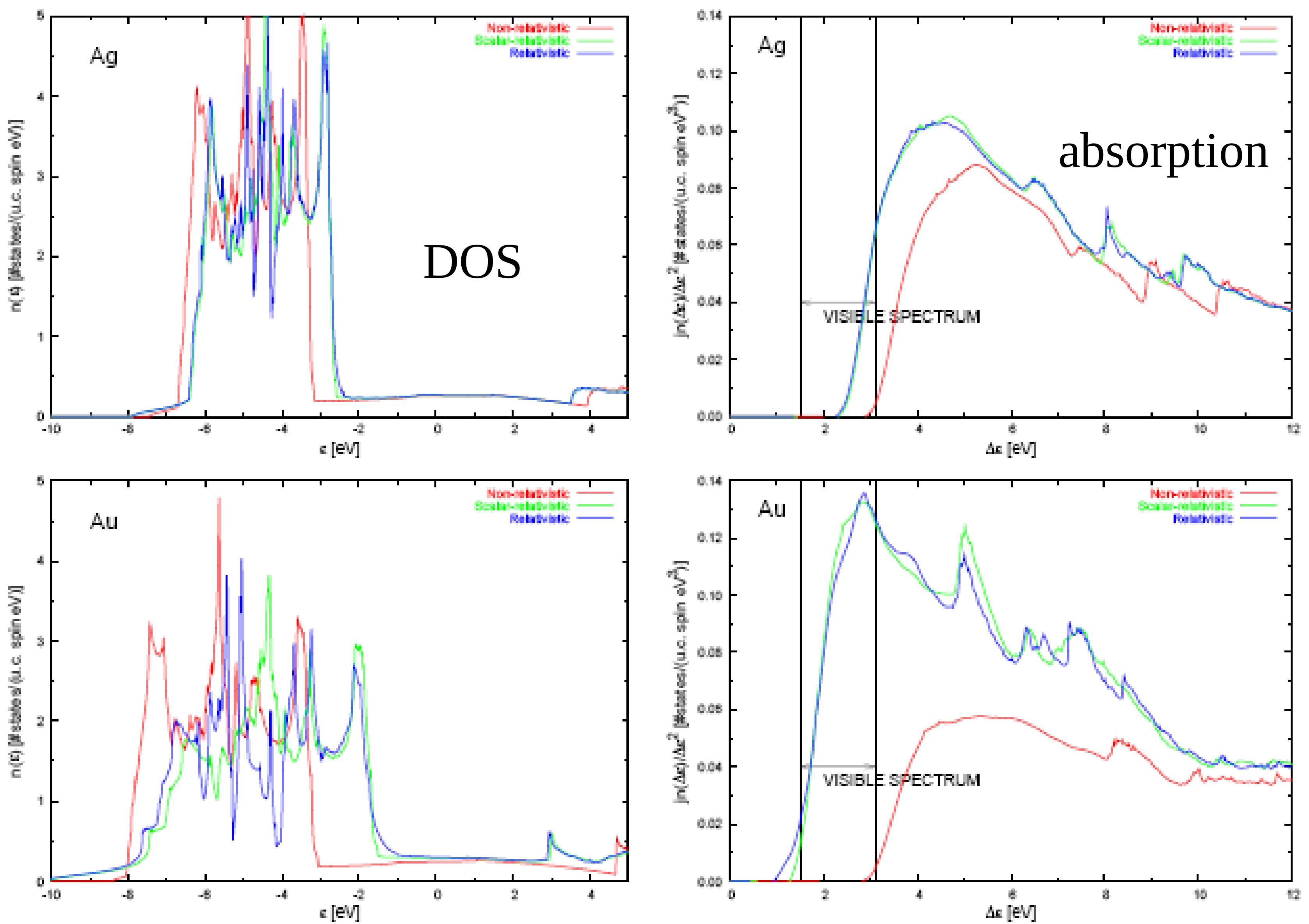
  - *What is the "optical gap" in Ag and Au:*

case	NREL	RELA	RELA+SO
Ag			
Au			

Exercise 15: optical properties

- Optical properties: fcc Al (either in w2web or on the command line)
  - *a_0=4.05 Å ; init_lapw ; run_lapw*
  - *calculate optics (as described in the optics lecture, compare with the Al - Fig.)*
    - calculate plasma frequency and dielectric function
    - check your results with respect to k-mesh and repeat with different k-meshes
      - x kgen (increase k-mesh=k-mesh$_{old}$*2), Emax=5.0 in Al.in1
      - x lapw1 ; x lapw2 –fermi
      - x optic, x joint, x kram # this creates default inputs. Edit them for your needs
      - x optic, x joint (with SWITCH=6, Plasmafrequency in Al.outputjoint)
      - x joint (with SWITCH=4), x kram
      - opticplot
- Optical properties: fcc Ag and Au (both have a_0=4.08 Å)
    - NREL calc. (change RELA to NREL in case.struct after init_lapw), do the scf and optics (save nrel; save nrel –optic)
    - scalar-relativistic calc. (NREL → RELA), do the scf and optics (save rela)
    - include spin-orbit: init_so; run_lapw -so (case.inso **without RLOs** since optic does not support RLOs; put large Emax in case.in1); optics (save rela-so)

Ag and Au: a relativistic effect



Exercise 16: Valence PES of PbO_2

- Learn valence photoelectron spectroscopy calculations with WIEN2k
- Learn how to renormalize the PDOS and decompose the interstital DOS

- Questions: Compare spectra with lecture notes !
  - *Note the change of cross sections with excitation energy. Fill the following table:*

Energy	**Pb-6s**	**Pb-6p**	**Pb-5d**	**O-2s**	**O-2p**
56					
1487					
7799					

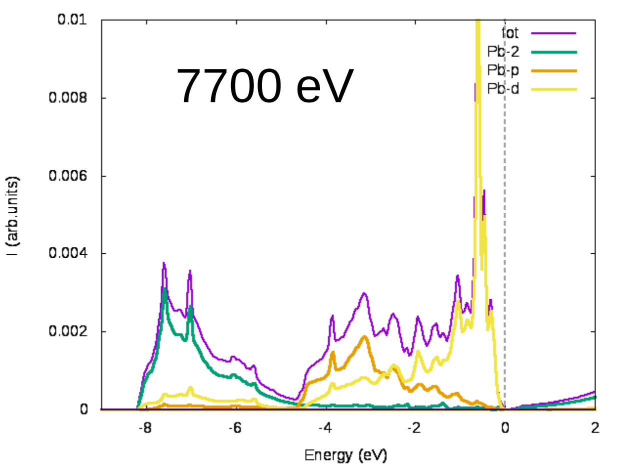
  - *Note the different localization of the atomic orbitals. Put the **fraction of charge** into the Table:*

	Pb-6s	**Pb-6p**	**Pb-5d**	**O-2s**	**O2p**
lstart	0.503				
PES					

  - *Note the difference in the PDOS and renormalized PDOS*
  - *PS: The cross section of Pb-5d is a bit overestimated, since the Pb-d states close to EF have already 6d character (smaller cross section)*

Exercise 16: Valence PES of PbO_2

- PbO2: SG 136 (P42/mnm), a= 9.37, c=6.42 bohr; Pb(0,0,0), O(.307,.307,0)
- mkdir ~/WIEN2k/PbO2; cd ~/WIEN2k/PbO2
- makestruct; cp init.struct PbO2.struct
- init; run
- x lapw2 -qtl
- configure_int with total, Pb-s,p,d and O-s,p DOS
- x tetra
- x pes
  - *Run and then plot it for 3 different excitation energies: 56,*
  - *Press "enter" for all other questions*



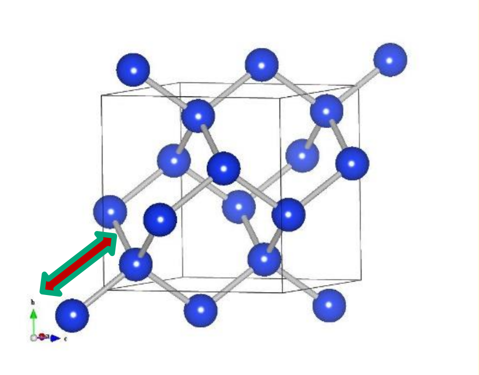
- x broadening –pes (2 times)
- Plot the PDOS, the renormalized PDOS and the PES (for all 3 energies).
- dosplot2 # dosplot2 -ren # dosplot2 –pes # dosplot2 –pesb
- Nicer plots are obtained with broadened spectra (-pesb) when comparing with experiment.

Exercise 17: Phonons of Si and $SrTiO_3$

- Learn about phonons “by hand”
- phonon calculations with WIEN2k and Phonopy

- Questions:
  - *Γ phonon in Si (exp: 15.53THz)*
    - Displacement   force (mRy/bohr)   total energy (Ry)
    - -0.005
    - 0
    - 0.005
    - force constant k (mRy/bohr^2):
    - frequency (THz):
    - frequency (THz) from phonopy:
  - *Why are there imaginary frequencies at M and R in SrTiO3 ?? What does it mean ??*
  - *What would you have to do to calculate the LO-TO splitting at Γ ?*

Phonons in Si



- Calculate optical Γ phonon is Si:
  - *FCC, a0=5.43 Å, Si at ±(.125+x,.125+x,.125+x)*
  - *use: PBE, 500 k,* ***-fc 0.02 -cc 0.001***
  - *initialize with x=-0.005, run scf, save,*
  - *change x to +0.005, run scf*
  - *due to anharmonicity of the vibration determine the force constant k from the forces (convert to Ry/bohr) using:*

$$\Delta E = \frac{k}{2}u^2 + 4k_{xyz}\left(\frac{u}{\sqrt{3}}\right)^3, \qquad -F = ku + \frac{4}{\sqrt{3}}k_{xyz}u^2$$

with $u_{111}=\sqrt{3}\ x\ a_0$ *and* F_{111} *from (:FOR)*

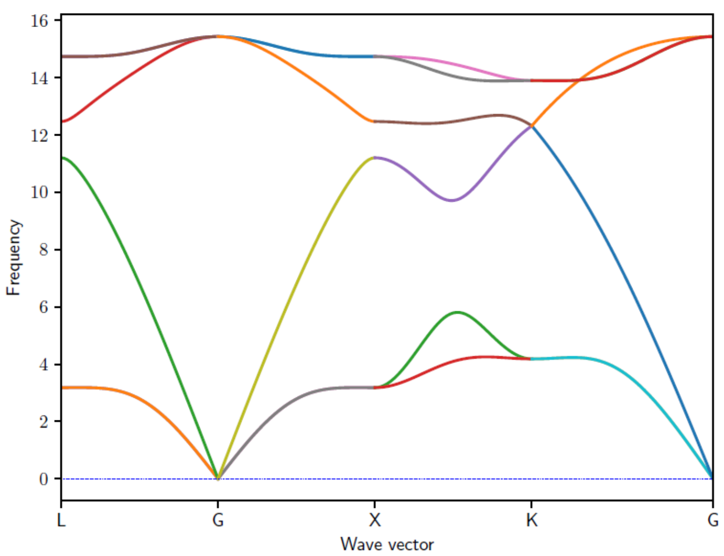
  - *from k calculate the frequency (exp: 15.53 THz)*
  - $\omega[THz] = 108.9\sqrt{\frac{k}{M}}$ *with k[Ry/bohr*^{2}*] and mass M=28.0855*
  - *conversion:* $\sqrt{\frac{13.6\ eV\ 1.6\ 10^{-19}\ J/eV}{1.66\ 10^{-27}\ kg}}\ /\ 0.529177\ 10^{-10}\ m\ *\ 241.8\ THz$

Phonons of Si using Phonopy (without w2web)

- makestruct
  - *Si, F cell, a=b=c=5.43 Ang, α=β=γ=90°; Si: (.125,.125,.125);(.875,.875,.875)*
  - *setrmt 3%*
- x supercell (init.struct, 1x1x1, P-lattice) # phonopy can handle only P
- cp init_super.struct Si-phonon.struct
- edit Si-phonon.struct # label all atoms as Si1,2,3,4
- init_lapw –prec 1n
- phonopy --wien2k -c Si-phonon.struct -d --dim="2 2 2"
- mkdir 1;
- cp Si-phonon.structS-001.in 1/1.struct
  - *cd 1*
  - *init_lapw –prec 1n*
  - *run_lapw -fc 0.02* *# (optionally use some parallelization)*
  - *cp 1.scf ..*
  - *cd ..*

Phonons continued

- phonopy --wien2k -f 1.scf
- create band.conf with editor, containing the following information:
  - *ATOM_NAME = Si*
  - *DIM = 2 2 2*
  - *PRIMITIVE_AXIS = 0.5 0.5 0.0   0.0 0.5 0.5   0.5 0.0 0.5*
  - *BAND = 0.5 0.5 0.5   0 0 0 0.5 0 0   0.5 0.5 0   0 0 0*
  - *BAND_LABELS = L G X K G*
  - *BAND_CONNECTION = .TRUE.*
- phonopy --wien2k -c Si-phonon.struct band.conf -p
- phonopy --wien2k -c Si-phonon.struct band.conf -p -s # creates band.pdf
- your result should be like:



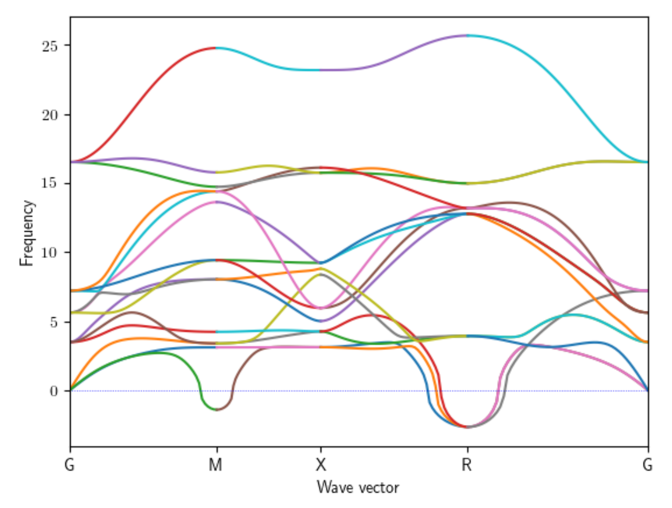
- check in band.yaml the Γ-frequency and compare with it with your previous result

Exercise 17: Phonons of $SrTiO_3$

- cd ~/WIEN2k; mkdir SrTiO3; cd SrTiO3;
- makestruct
  - *$SrTiO_3$: SG 221(P m-3m), a=b=c=7.38 bohr, α=β=γ=90°*
  - *Sr (0.5, 0.5, 0.5), Ti (0, 0, 0), O (0.5, 0 , 0)*
  - *setrmt 3%*
- cp init.struct SrTiO3.struct
- init_lapw –prec 1n
- phonopy --wien2k -c SrTiO3.struct -d --dim="2 2 2"
- mkdir 1; mkdir 2; mkdir 3
- Copy SrTiO3.structS-001, SrTiO3.structS-002 and SrTiO3.structS-003 in 1,2 and 3
- Do the same for directories 1, 2 & 3 (open 3 windows and do it in parallel):
  - *cd 1*
  - *mv SrTiO3.structS-001 1.struct*
  - *init_lapw -prec 1n*
  - *run_lapw -fc 0.1 –cc 0.0001*
  - *cp 1.scf ..*

Phonons continued

- phonopy --wien2k -f 1.scf 2.scf 3.scf
- create band.conf with editor, containing the following information:
  - *ATOM_NAME = Sr Ti O*
  - *DIM = 2 2 2*
  - *PRIMITIVE_AXIS = 1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0*
  - *BAND = 0 0 0 1/2 1/2 0 0 1/2 0 1/2 1/2 1/2 0 0 0*
  - *BAND_LABELS = G   M X R G*
  - *BAND_CONNECTION = .TRUE.*
- phonopy --wien2k -c SrTiO3.struct band.conf –p
- save with: phonopy --wien2k -c SrTiO3.struct band.conf –p –s #(band.pdf)
- your result should be like:



Exercise 18: Parallelization

- These tests should be done in the SrTiO3/1-directory (phonon exercise). We use this larger example (40 atoms/cell, no inversion) to test the different parallelization realized in WIEN2k.
- **OMP parallelism (shared memory only, on one node):**
  - *Is used automatically, if the environment variable OMP_NUM_THREADS is set (4 in our setup). You can change it with:*
    - export OMP_NUM_THREADS=N   or
    - edit .machines and insert lines like: omp_global:N (omp_lapw0/1/2:N)
  - *Run* `x lapw0`, `x lapw1`, *and* `x lapw2` *with different number of threads and fill out the table below. For more info of different parts: tail *put0; grep horb *put1; tail -15 *put2*
  - *In a second terminal keep a top command running to keep track of the peak CPU usage! (timing and performance depends on your actual hardware (cores, memory-access, frequency). Here we have a 6 core Intel I7 cpu.)*

	# threads = 1		**# threads = 2**		**# threads = 4**		**# threads = 8**	
program	time (s)	% CPU	time (s)	% CPU	time (s)	% CPU	time (s)	% CPU
lapw0								
lapw1								
lapw2								

Exercise 18: Parallelization

- **k-point parallelism:** (works also on a cluster of PCs with shared filesystem)
  - *x kgen; 30 ; 1 → 6 k-points in IBZ*
  - *needs a .machines file (we have 6 k-points and 6 cores, so up to 6 parallel jobs make sense)*

```
1:localhost
1:localhost
1:localhost
omp_global:1    #  or 2
```

  - *Calculations are started with **x lapw1/2 (–p)***
  - *Run* `lapw1` *and* `lapw2` *serial or in parallel with omp_global 1 and 2, fill out the table below and compare the times.*

# parallel jobs	**1**	**1 omp2**	**3**	**3 omp2**	**6**	**6 omp2**
program	time (s)	time (s)	time (s)	time (s)	time (s)	time (s)
`lapw1`						
`lapw2`						

- **MPI parallelism:** (not installed here. Distributed matrices, useful for VERY large problems (more than 64 atoms/cell, NMAT > 10000) and a large computer cluster with infiniband network (up to several 100 cores))

Exercise 19: Baders AIM analysis

- Learn how to calculate “Atoms in Molecules” properties (bond critical points, atomic basins, atomic charges) with the aim and the critic2 (unsupported software) tool
- Learn to get /RMT independent) magnetic moments/atom for compounds
- Questions:
  - *How many different bond- ring- and cage - critical points do we have in TiN and TiC?*
  - *What are the Ti (N,C) Bader charges and atomic volumes in TiN and TiC using aim and critic2 /100/200/300)?*

	Vol - aim	**Vol-crit 100/200/300**	**Q(Ti) aim**	**Q(Ti)-(crit) 100/200/300**	**Q(X) aim**	**Q(X)-crit 100/200/300**
Ti(TiN)						
Ti(TiC)						
N(TiN)						
C(TiC)						

- *What are the magnetic moments in NiO from :MMI and Bader analysis ?*

Exercise 19: Baders AIM analysis

- Calculate this in TiN and TiC (exercise 1 and 2)
  - *cd ~/WIEN2k/TiN   # we already have a scf density*
  - *x aim   # creates a default* ***TiN.inaim*** *(bond critical points)*
  - *x aim   # executes aim with the default input*
  - *extractaim   TiN.outputaim   # extracts BCP and puts them into* ***critical_points_ang***
  - *cat critcal_points_ang*

  - *e TiN.inaim   # copy the „SURF" part at the beginning*
  - *x aim; tail TiN.outputaim*
  - *e TiN.inaim   # change to atom 2*
  - *x aim; tail TiN.outputaim*

- repeat this in TiC (restore the best calculation for vol=0.0 first !)

Exercise 19: Baders AIM analysis

- Calculate the magnetic moments of Ni and O in AFM NiO in GGA and GGA+U
  - *go back to the previous NiO exercise*
  - *restore_lapw ….. # restore the corresponding calculation*
  - *x aim –dn # creates case.inaim*
  - *edit case.inaim # put surface part on top*
  - *x aim –dn # get moment at the bottom of case.outputaim**
  - *repeat for the O atom*
  - *repeat for GGA+U*
  - *compare :MMIxxx with the Bader moments*

Exercise 20: Ga-NMR in $BaGa_4$

- Learn how to calculate Hyperfine interaction spectra (NMR, Mössbauer, PAC)
- Learn how to calculate Electric field gradients (EFG) in solids
- Learn how to calculate NMR Chemical shifts, Knight shifts and dipolar corrections

- Questions: Estimate σ_{ref} and complete the following table:

case	**V_{zz}(exp)**	**V_{zz}(th)**	**δ_{iso}(exp)**	**δ_{iso}(th)**	**σ_{orb}**	**σ_c**	**σ_{sd}**
Ga1	5.99		3010				
Ga2	1.20		840				

- R.Laskowski et al., J. Phys. Chem. C 2017, 121, 753−760

Exercise 20: Ga-NMR in BaGa$_4$

- This exercise should be done WITHOUT w2web in a terminal window !
- *cd ~/WIEN2k; mkdir BaGa4; cd BaGa4*
- *makestruct* (and type in the following information)
  - *BaGa$_4$: SG 139 (I4/mmm), a,a,c= 4.566 4.566 10.775 Ang*
  - *Ba (0,0, 0), Ga1(0.5,0, 0.75), Ga2 (0,0, 0.38)*
- *cp init.struct BaGa4.struct*
- *init_lapw -numk 10000 –fermit 0.004*
- edit .machines (insert 2 lines with 1:localhost)
- *run_lapw -p -fc 1 -cc 0.0001; save_lapw rkm7_10k_4mry*
- *tail *scf* and verify that the forces are “small” (no struct opt. necessary)
- *x_nmr_lapw -mode in1* (and view the resulting *in1c_nmr file)
- *x_nmr_lapw –p –metal*
  - *check BaGa4.outputnmr_integ for* σ_{iso} *of both Ga atoms*
  - *grep :EFG002 *scf*
  - *grep :EFG003 *scf*

- * These calc. will take some time, continue next page while run/x_nmr is running

NMR continued ..

- `mkdir spin; cp BaGa4.struct spin/spin.struct; cp .machines spin; cd spin`
- `instgen_lapw –nm; init_lapw –b –sp –numk 10000 –fermit 0.004`
- `runsp_c_lapw –p;   save non-magnetic`
- `cp $WIENROOT/SRC_templates/case.vorbup_100T spin.vorbup` (and for dn)
- `runsp_lapw –p –orbc –cc 0.000005;  save_lapw rkm7_10k_4mry`
- `grepline :hff002/3 ´rkm*scf´ 3` σ_c=HFF*1000 *(for a 100T field)*
- compare with experiment: (F.Haarmann et al. Chem. Eur. J. 2011, 17, 7560 – 7568)
- `cp $WIENROOT/SRC_templates/case.indm spin.indm;`
- edit *indm: set for atom 2 and 3, three ℓ values, ℓ=1,2,3; and 3 5 in last line
- `x lapwdm –p –up/dn; cat *scfdmup/dn;` $\sigma_{sd}=(tot_{up}-tot_{dn})*10000$
- $\sigma_{tot}(th)=\sigma_{iso}-\sigma_c-\sigma_{sd}$ *and then* $\delta_{iso}(th)=(\sigma_{tot}(th)-\sigma_{ref}(th))$
- *Estimate* σ_{ref}*(th) to obtain “best” agreement with exp. (usually one would either calculate the reference compound or do several Ga compounds)*
- *In metals the results are* ***very*** *sensitive to k-mesh and smearing and careful convergence are necessary (up to 1 000 000 k-points).*

exercise 21: boltztrap2

- Learn how to install Botztrap2 using pip
- Learn some basic python

- Questions:

- Inspect the relevant data of TiCoSb and characterize the DFT calculation.
  - *What rkmax, k-mesh and functional was used for the scf calculation.*
  - *What k-mesh was used for the energy file*
- Plot the band structure
- Plot the transport coefficients as a function of the chemical potential μ at 300 K and 1000 K. At what μ are the maxima ?

boltztrap2

- install botztrap and some examples:
  - *pip install BoltzTraP2*
  - *git clone https://gitlab.com/sousaw/BoltzTraP2.git*
  - *cd BoltzTraP2*
  - *tar -xf data.tar.xz*
  - *inspect the files in data/TiCoSb*
  - *cd examples/*
  - *python3 TiCoSb.py*
  - *edit TiCoSb.py and change the temperature from 300 to 1000 K*
  - *python3 TiCoSb.py*

  - *You may also explore the commandline interface btp2*
  - *checkout the doku/wiki at the gitlab page*

Exercise 22: wannier90+BerryPi+fold2Bloch+mstar

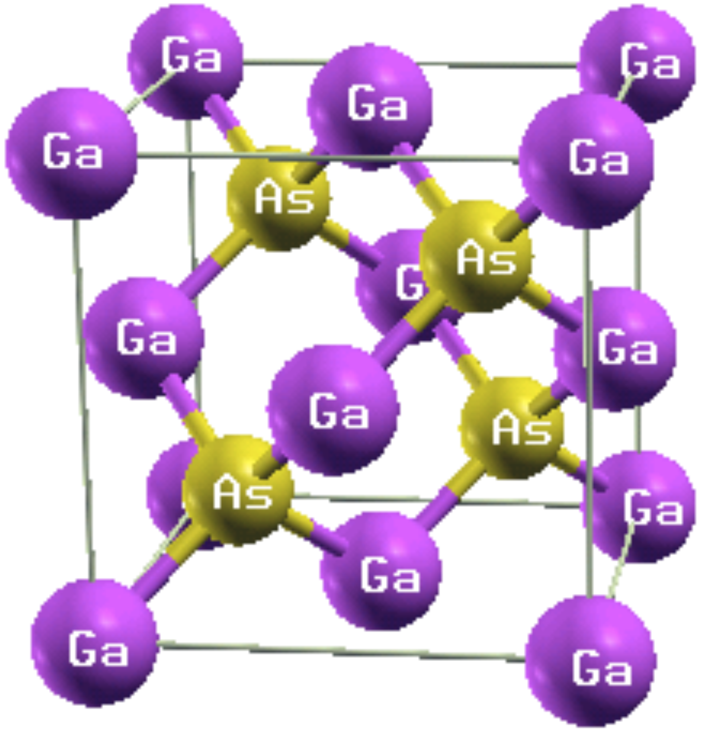
- Create **wannier** functions using **w2w** and **wannier90**
- Calculate **Born** effective charges using **BerryPi**
- Calculate **topological** properties (**Weyl-points, Cern numbers**) using **BerryPi**
- **Backfold** bandstructures from supercells using **fold2Bloch**
- **mstar:** calculate band masses in semiconductors

GaAs -- MLWF

Special thanks to Elias Assmann (TU Graz)
for the generous help in
preparation of this tutorial

YouTube video: https://youtu.be/R4c1YHDh3GE

1. Wien2k SCF



Create a tutorial directory, e.g.

```
$ mkdir GaAs-MLWF
```

Create the structure file using the following parameters:

2 atoms per primitive unit cell (Ga, As)

Lattice "F" = f.c.c.

Lattice parameters $a_0 = b_0 = c_0$ = 10.683 Bohr

Positions: "0 0 0" for Ga and "1/4 1/4 1/4" for As; RMT's - automatic

You can use `xcrysden` to view the structure

```
$ xcrysden --wien_struct GaAs-MLWF.struct
```

Initialize Wien2k calculation (LDA, ~600 k-points ≡ 8x8x8 mesh)

```
$ init_lapw -b -vxc 5 -numk 600
```

Run regular SCF calculation using default convergence criteria

```
$ run_lapw
```

After SCF cycle is completed (~8 iterations). We proceed with the band structure

Prepare the list of k points to be used for the band structure plot (`GaAs-MLWF.klist_band` file) using `xcrysden`

```
xcrysden File > Open Wien2k
    > Select k-path
```

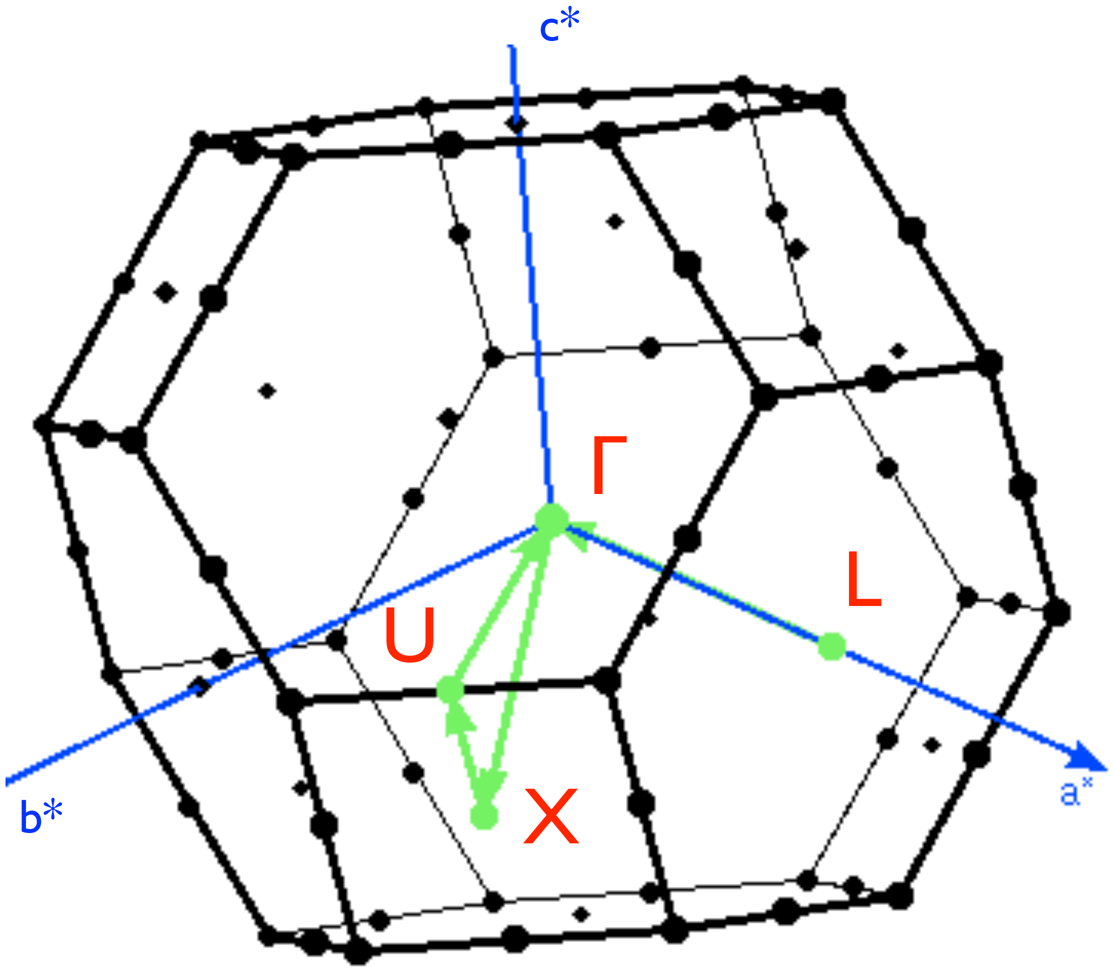
Select points L(1/2 0 0), Γ(0 0 0), X(1/2 1/2 0), (5/8 5/8 1/4), Γ

Set the total of 100 k-points along the path.

Save the list as `GaAs-MLWF.klist_band`

Solve eigenvalue problem on the k-path

```
$ x lapw1 -band
```



Get the Fermi energy

```
$ grep :FER *scf
```

For the band structure plot we will use the web interface (w2web). Create a new session and navigate to the current work directory.

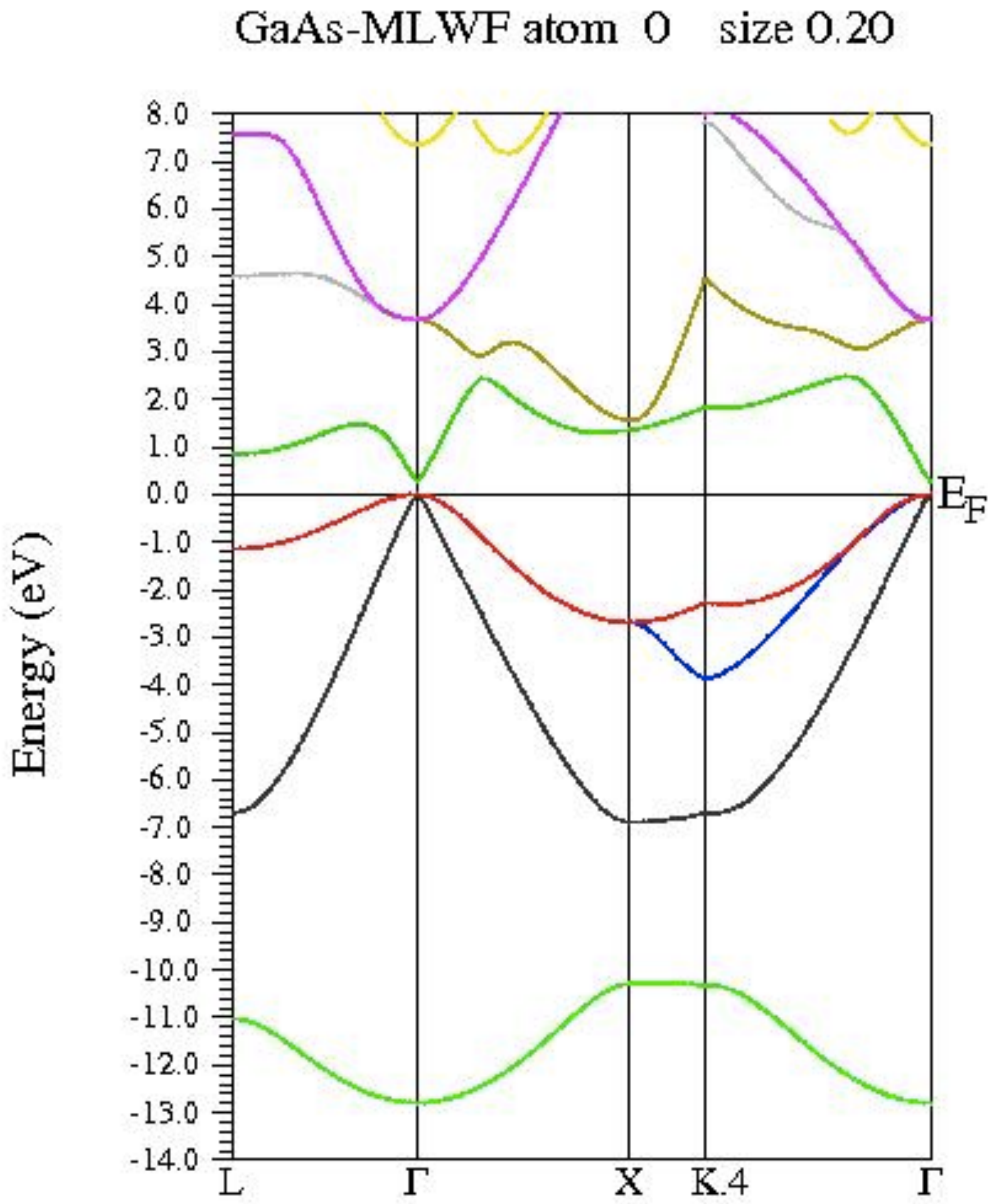
w2web **Tasks > Bandstructure**

w2web **Select**
“Edit GaAs-MLWF.insp”,
insert the Fermi energy,
save

w2web **x spaghetti**

w2web **plot band structure**

Your band structure will be similar to the one shown on the right.
Our aim is to construct Wannier functions that reproduce this band structure including valence and some conduction bands.



Before we proceed it is useful to determine the band indices for the region of interest

```
$ grep :BAN *scf2
:BAN00004:   4   -2.275995   -2.275442  2.00000000
:BAN00005:   5   -2.275995   -2.275261  2.00000000
:BAN00006:   6   -0.784989   -0.777234  2.00000000
:BAN00007:   7   -0.777234   -0.772519  2.00000000
:BAN00008:   8   -0.777234   -0.772371  2.00000000
:BAN00009:   9   -0.772654   -0.770305  2.00000000
:BAN00010:  10   -0.771540   -0.769389  2.00000000
:BAN00011:  11   -0.625463   -0.437790  2.00000000
:BAN00012:  12   -0.191045    0.314670  2.00000000
:BAN00013:  13    0.029025    0.314670  2.00000000
:BAN00014:  14    0.067108    0.314670  2.00000000
:BAN00015:  15    0.335212    0.647926  0.00000000
:BAN00016:  16    0.429016    0.720389  0.00000000
:BAN00017:  17    0.585553    1.053515  0.00000000
:BAN00018:  18    0.585553    1.053515  0.00000000
:BAN00019:  19    0.853994    1.118413  0.00000000
```

d-orb. of As and Ga (do not participate in bonding)

4 bonding + 4 antibonding orbitals (8 in total for sp3 bonding x 2 atoms)

⇧ Emin (Ry)

⇧ Emax

⇧ occupancy

2. Construction of Wannier functions

Prepare a separate directory

```
$ prepare_w2wdir GaAs-MLWF GaAs-WANN
$ cd GaAs-WANN
```

Initialize Wien2Wannier

```
$ init_w2w
```

Select 8x8x8 k-mesh (unshifted);

energy range (eV) -13 10 (this is not very critical);

band indices [Nmin Nmax] 11 18 (see the previous page);

for the projection we choose “1:s,p” and “2:s,p” (1 = Ga, 2 = As)

Get the vector file on the full Brillouin zone mesh

```
$ x lapw1
```

Compute matrix elements needed for Wannier90

```
$ x w2w
```

Run Wannier90

```
$ x wannier90
```

Verify the output

spread $\langle \Delta \mathbf{r}^2 \rangle$

⇓

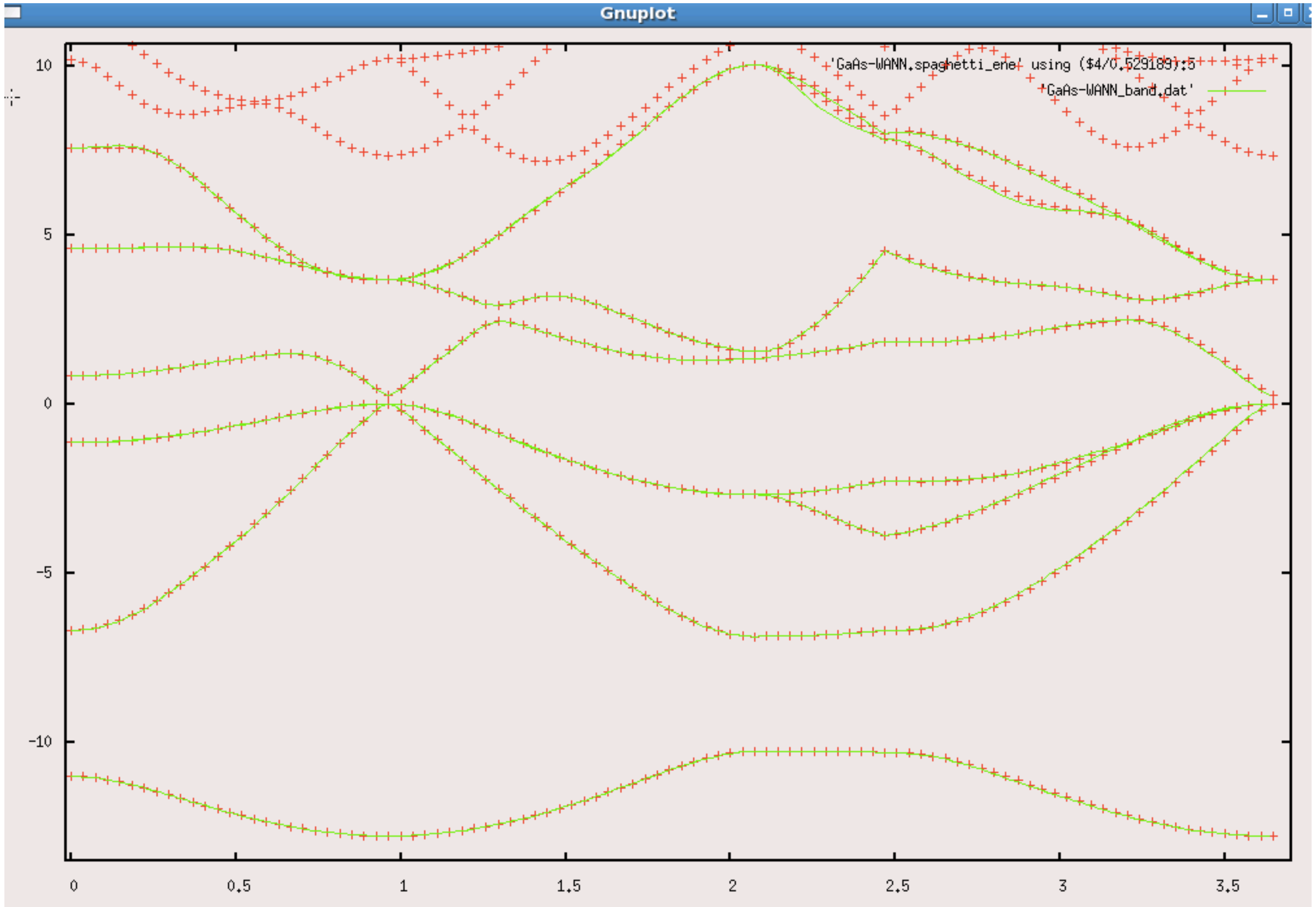
```
$ less GaAs-WANN.wout
...
Final State
  WF centre and spread    1  (  0.000000, -0.000000, -0.000000 )     1.91981243
  WF centre and spread    2  ( -0.000000, -0.000000,  0.000000 )     5.86945318
  WF centre and spread    3  ( -0.000000,  0.000000, -0.000000 )     5.86945318
  WF centre and spread    4  (  0.000000,  0.000000,  0.000000 )     5.86945318
  WF centre and spread    5  (  1.413299, -1.413299, -1.413299 )     1.61179550
  WF centre and spread    6  (  1.413301, -1.413300, -1.413300 )     3.82462632
  WF centre and spread    7  (  1.413300, -1.413301, -1.413300 )     3.82462632
  WF centre and spread    8  (  1.413300, -1.413300, -1.413301 )     3.82462632
...
```

There you can see the position and spread of the WF's, how they changed in the course of convergence. WF's 1-4 are all positioned at the origin (atom 1), WF's 5-8 are centred at the 2nd atom (please check the coordinates)

3. Post-processing

Plot the band structure

```
$  gnuplot
gnuplot> plot 'GaAs-WANN.spaghetti_ene' using
   ($4/0.529189):5, 'GaAs-WANN_band.dat' with lines
```



+ original Wien2k band structure

– Band structure computed from Wannier functions

Plotting WF's (can take a while). Get the template of an input file

```
$ cp $WIENROOT/SRC_templates/case.inwplot GaAs-WANN.inwplot
```

Edit "GaAs-WANN.inwplot"

Select origin "`-1  -1  -1  1`" and axis x, y, z
" 1 -1 -1 1"
"-1 1 -1 1"
"-1 -1 1 1"
grid point mesh: 30 30 30
"1" for the Wannier function index

Compute the 1st Wannier function on the mesh chosen

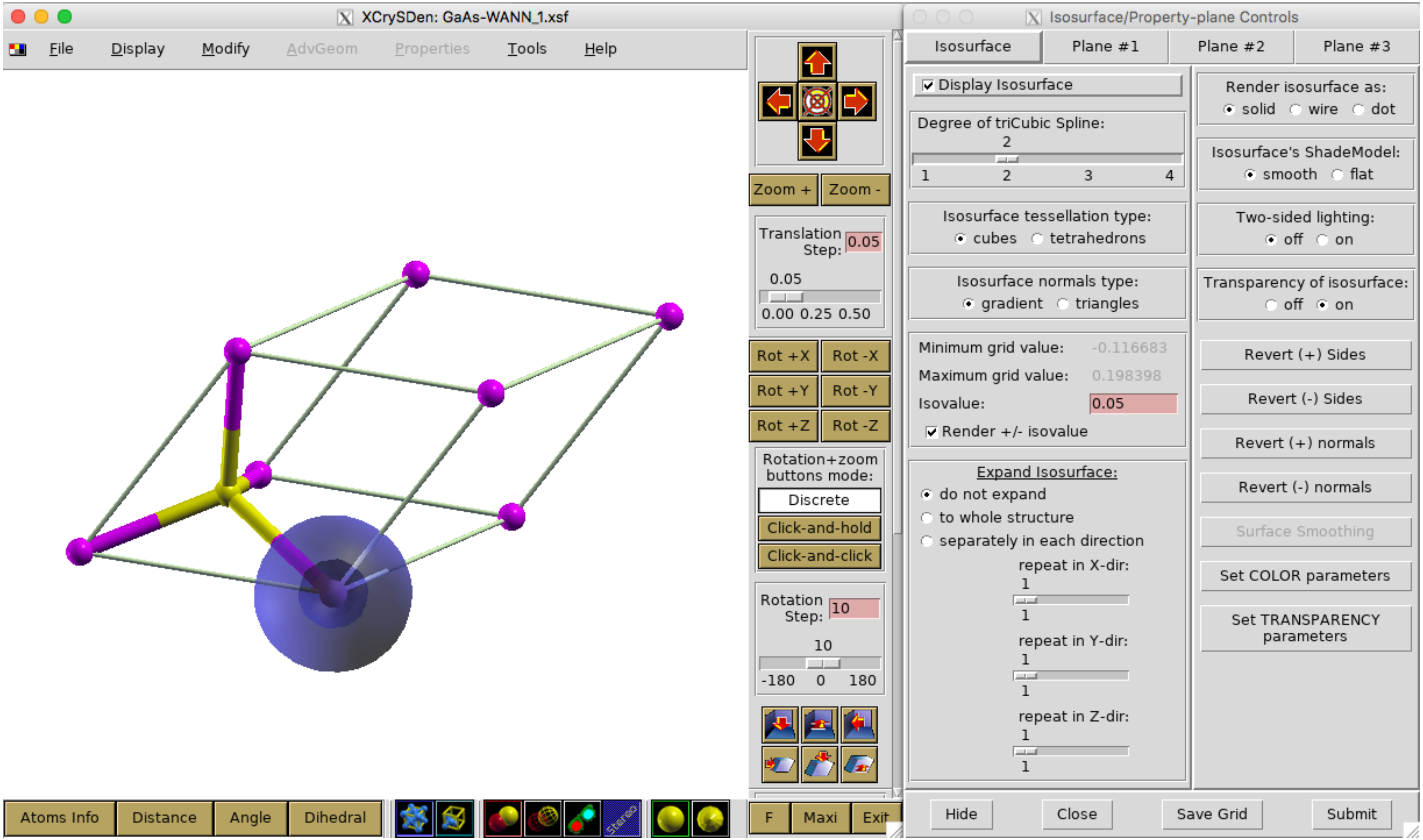
```
$ x wplot -wf 1
```

If you need to plot any other WF's (2, 3, etc), just edit the option.

Convert the output of `wplot` into `xcrysden` format for plotting.

```
$ wplot2xsf
```

Visualize with `xcrysden` (instructions on the next page)



Translation Step: 0.05

0.05

0.00 0.25 0.50

Rotation+zoom buttons mode: Discrete

Rotation Step: 10

10

-180 0 180

☑ Display Isosurface

Degree of triCubic Spline: 2

1 2 3 4

Isosurface tessellation type: ● cubes ○ tetrahedrons

Isosurface normals type: ● gradient ○ triangles

Minimum grid value: -0.116683

Maximum grid value: 0.198398

Isovalue: 0.05

☑ Render +/- isovalue

Expand Isosurface:

● do not expand

○ to whole structure

○ separately in each direction

repeat in X-dir: 1

1

repeat in Y-dir: 1

1

repeat in Z-dir: 1

1

Render isosurface as: ● solid ○ wire ○ dot

Isosurface's ShadeModel: ● smooth ○ flat

Two-sided lighting: ● off ○ on

Transparency of isosurface: ○ off ● on

```
$  xcrysden --xsf GaAs-WANN_1.xsf
```

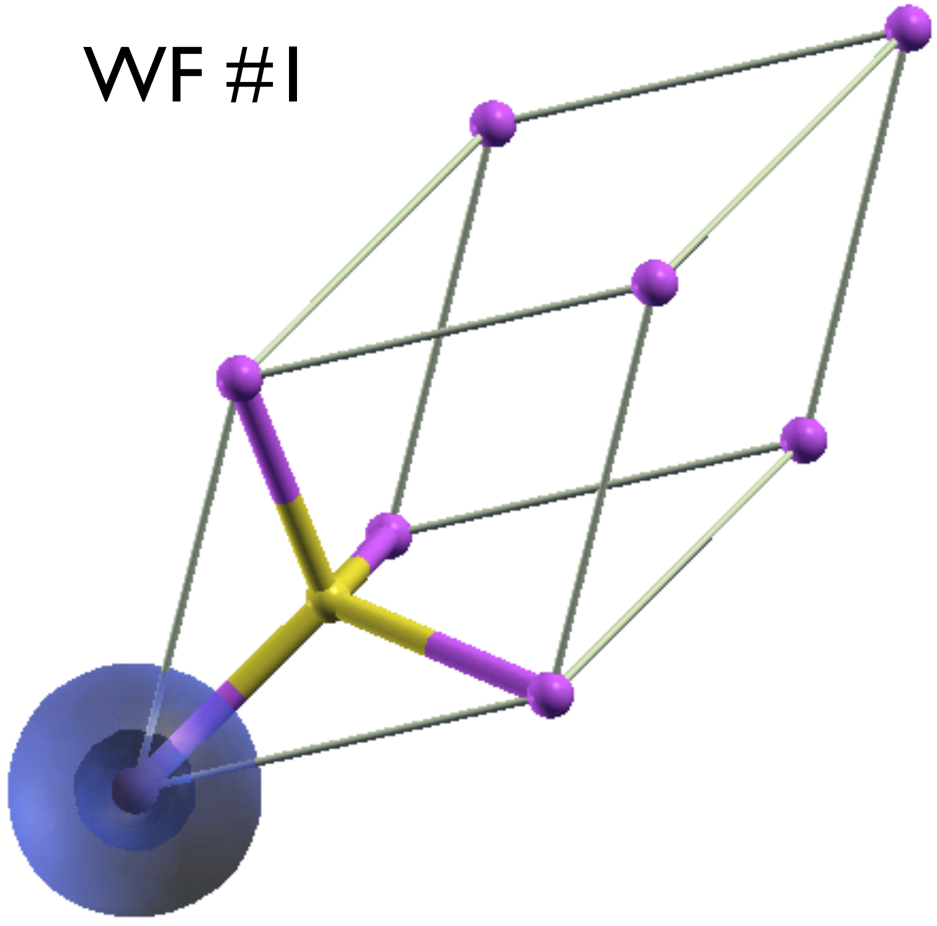
`xcrysden` `Tools > Data Grid > OK`

Set "Isovalue:" and check box "render +/- isovalue" (see screenshot on the previous page)

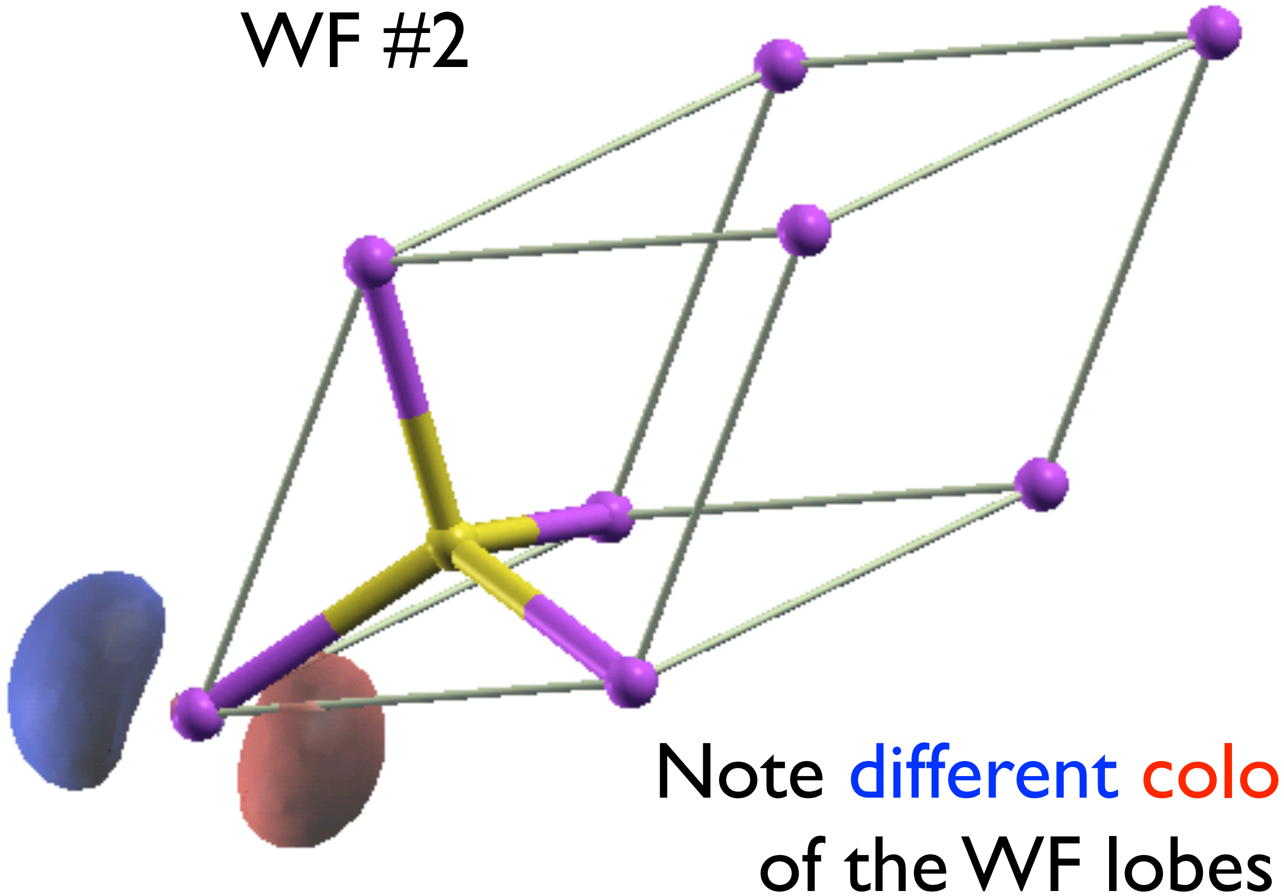
Play with the settings. You will get a spherical (s-like) WF centred at the origin.

The second WF resembles a p-orbital (you can get it by editing "GaAs-WANN.inwplot", re-run "`x wplot`" and "`wplot2xsf`"). The new file should be called **`GaAs-WANN_2.xsf`**

WF #1



WF #2



Note different colours of the WF lobes

Wannier Hamiltonian (similar to LCAO)

```
$   less GaAs-WANN_hr.dat
...
    0     0     0     1     1   -4.324045    0.000000
    0     0     0     2     1   -0.000000   -0.000000
    0     0     0     3     1   -0.000002    0.000000
    0     0     0     4     1   -0.000001    0.000000
    0     0     0     5     1   -1.472536    0.000000
    0     0     0     6     1   -1.157628    0.000000
    0     0     0     7     1   -1.157628    0.000000
    0     0     0     8     1   -1.157628    0.000000
...
```

Home unit cell

$\langle s_l|$

$|s_l\rangle$

Matrix element (eV)

$\langle s_l|H|s_l\rangle = E_{sl}$

no imag. part of the matrix element

no on-site hopping between different orbitals

Determine on site energies E_s and E_p for Ga and As and compare them to those suggested by Harrison (note: only their relative differences are important). Ga-p electrons correspond to the "0 0 0 2 2" line, As-s can be found at "0 0 0 4 4", etc. Results for the energy differences from WF are approximately 20-30% greater that Harrison's data. They should not agree exactly as WFs include matrix elements beyond the first nearest neighbour.

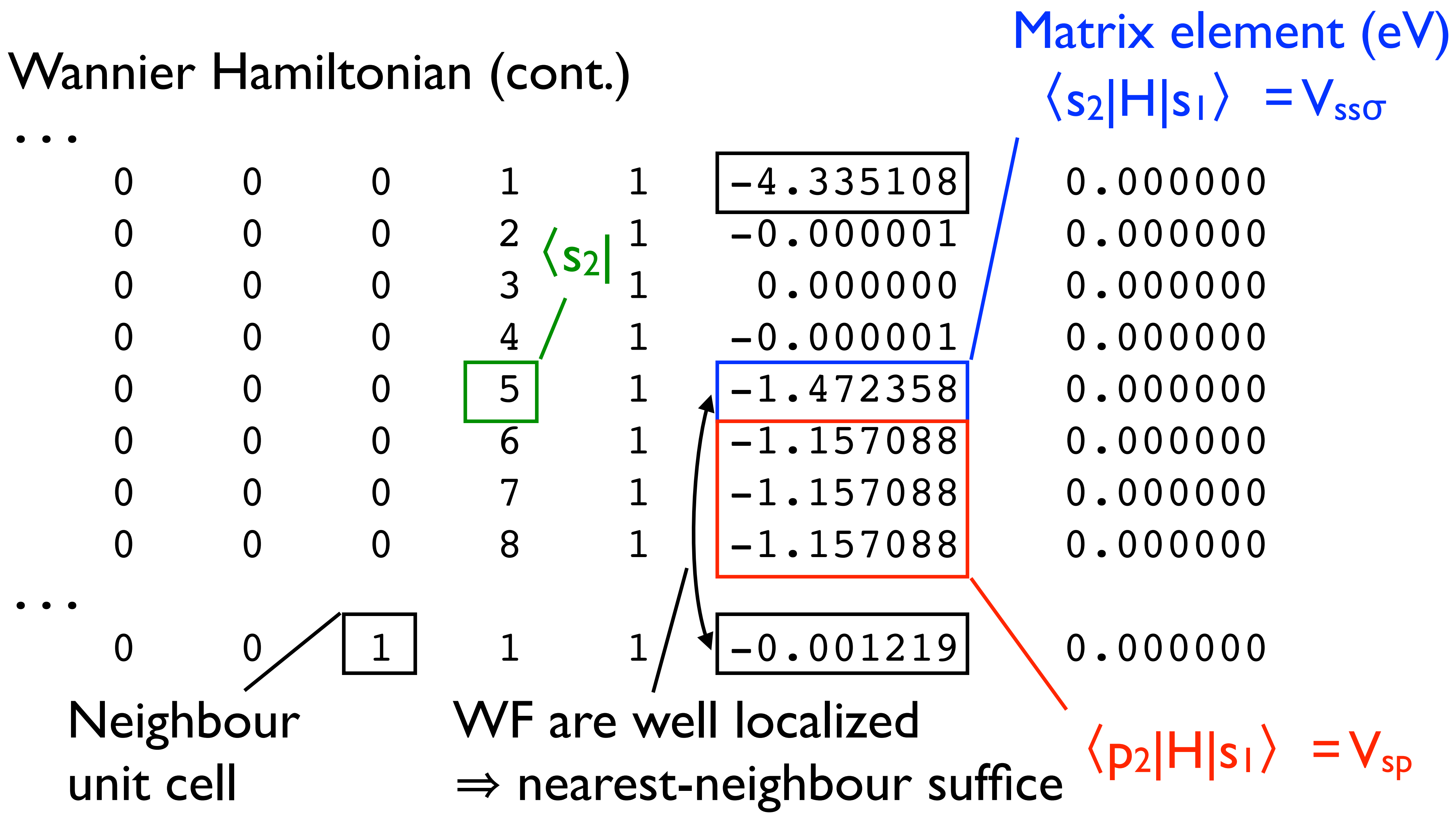
From Harrison's solid state tables:

$E_p(\text{Ga}) - E_s(\text{Ga}) = 5.9$ eV

$E_p(\text{As}) - E_s(\text{As}) = 9.9$ eV

$E_p(\text{Ga}) - E_p(\text{As}) = 3.3$ eV

Wannier Hamiltonian (cont.)



```
...
    0    0    0    1    1   -4.335108    0.000000
    0    0    0    2    1   -0.000001    0.000000
    0    0    0    3    1    0.000000    0.000000
    0    0    0    4    1   -0.000001    0.000000
    0    0    0    5    1   -1.472358    0.000000
    0    0    0    6    1   -1.157088    0.000000
    0    0    0    7    1   -1.157088    0.000000
    0    0    0    8    1   -1.157088    0.000000
...
    0    0    1    1    1   -0.001219    0.000000
```

Matrix element (eV)

$\langle s_2|H|s_1\rangle = V_{ss\sigma}$

$\langle s_2|$

Neighbour unit cell

WF are well localized
⇒ nearest-neighbour suffice

$\langle p_2|H|s_1\rangle = V_{sp}$

Now you have all information required to build your *ab initio* TB sp3 Hamiltonian (Yu & Cardona)

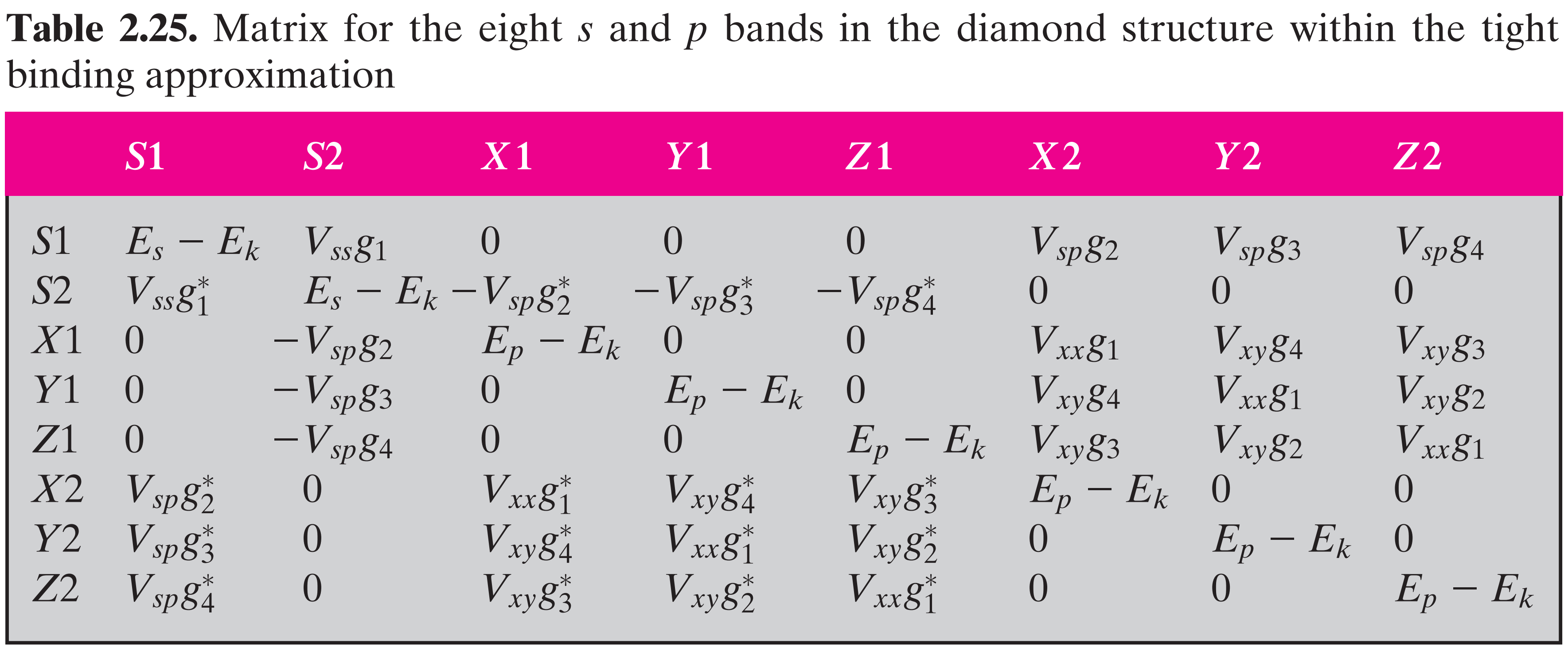


Table 2.25. Matrix for the eight *s* and *p* bands in the diamond structure within the tight binding approximation

	$S1$	$S2$	$X1$	$Y1$	$Z1$	$X2$	$Y2$	$Z2$
$S1$	$E_s - E_k$	$V_{ss}g_1$	0	0	0	$V_{sp}g_2$	$V_{sp}g_3$	$V_{sp}g_4$
$S2$	$V_{ss}g_1^*$	$E_s - E_k$	$-V_{sp}g_2^*$	$-V_{sp}g_3^*$	$-V_{sp}g_4^*$	0	0	0
$X1$	0	$-V_{sp}g_2$	$E_p - E_k$	0	0	$V_{xx}g_1$	$V_{xy}g_4$	$V_{xy}g_3$
$Y1$	0	$-V_{sp}g_3$	0	$E_p - E_k$	0	$V_{xy}g_4$	$V_{xx}g_1$	$V_{xy}g_2$
$Z1$	0	$-V_{sp}g_4$	0	0	$E_p - E_k$	$V_{xy}g_3$	$V_{xy}g_2$	$V_{xx}g_1$
$X2$	$V_{sp}g_2^*$	0	$V_{xx}g_1^*$	$V_{xy}g_4^*$	$V_{xy}g_3^*$	$E_p - E_k$	0	0
$Y2$	$V_{sp}g_3^*$	0	$V_{xy}g_4^*$	$V_{xx}g_1^*$	$V_{xy}g_2^*$	0	$E_p - E_k$	0
$Z2$	$V_{sp}g_4^*$	0	$V_{xy}g_3^*$	$V_{xy}g_2^*$	$V_{xx}g_1^*$	0	0	$E_p - E_k$

This tutorial was verified using

Wien2k 19.1
Wannier90 2.1.0
python 2.7.14
xcrysden 1.5.60

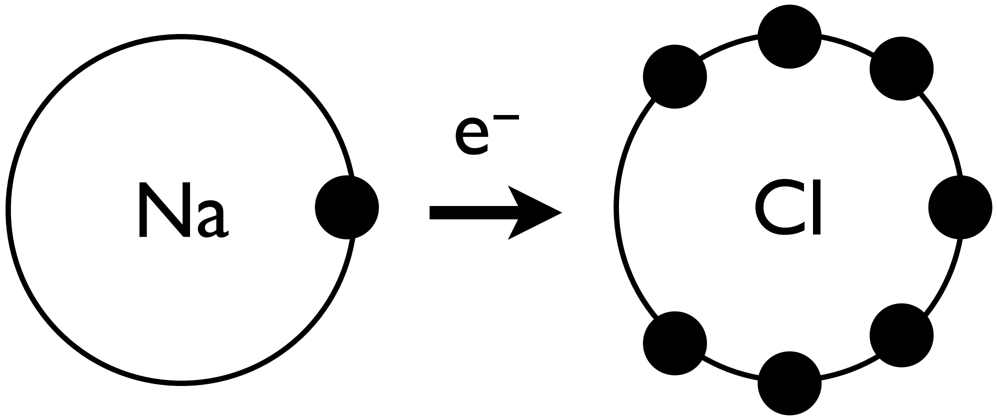
Born effective charge of GaN

+ + BerryPI

YouTube video: https://youtu.be/hLI9nKf35tA

Background

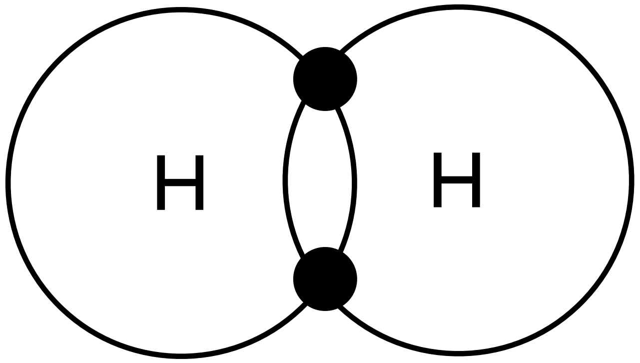
Ionic bond



$Z^* = +1$

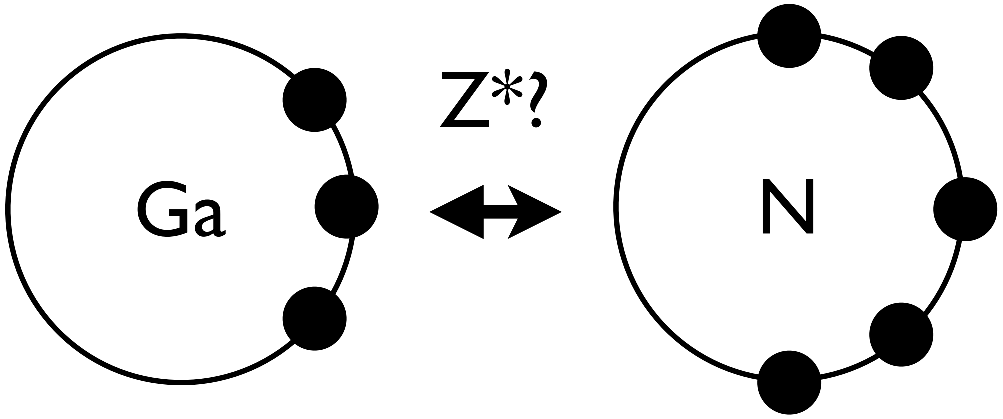
$Z^* = -1$

Covalent bond



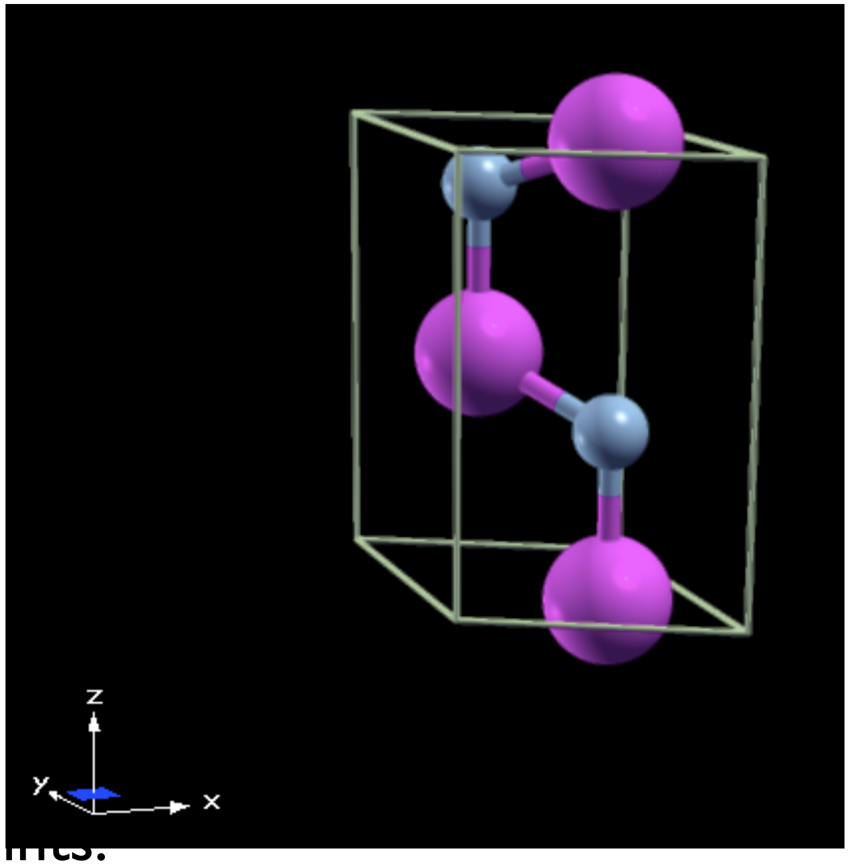
$Z^* = 0$

Mixed



Instructions

w2web Construct a structure file (`../GaN-W/GaN-W.struct`)



4-atoms (2-Ga, 2-N) per unit cell
Hexagonal lattice "H", α = β = 90°, γ = 120°
Cell size (Bohr): a = b = 5.963131; c = 9.722374
Coordinates:

Ga	(2/3	1/3	0)
Ga	(1/3	2/3	1/2)
N	(2/3	1/3	0.376393)
N	(1/3	2/3	0.876393)

Since we will introduce a displacement later,
a small (~3%) RMT reduction is needed.

Initialize SCF calculation with GGA-PBE, RKMAX=7, 300 k points.
The energy separation between core and valence electrons
should be set at -8 Ry to include Ga-3p as valence electrons (otherwise a core leakage warning will be printed).

```
$   init_lapw -b -vxc 13 -ecut -8 -rkmax 7 -numk 300
```

It is worth to verify the structure with XCrysDen

Perform regular SCF calculation

```
$   run_lapw
```

Run Berry phase calculation

```
$ berrypi -k 8 8 4 | tee log
```

Note the ionic and electronic phases along Z-axis (wrapped [-π…+π])

Save the calculation

```
$ save_lapw -d Lambda0
```

Introduce small displacement on N-atoms

w2web Edit Z-coordinate of N atoms by adding 0.001 to the equilibrium value of the fractional coordinate u_z. Since nitrogen has 2 equivalent positions, both need to be updated. Think what is the reason for the need to shift both atoms in this case.

Repeat initialization, SCF and Berry phase calculation steps (note "-a" option used in order to update the log file, not overwrite)

```
$ init_lapw -b -vxc 5 -ecut -8 -rkmax 7 -numk 300
$ run_lapw
$ berrypi -k 8 8 4 | tee -a log
```

option "-a" will append output to the file instead of overriding it

Evaluate the total Berry phase for each of two calculations performed

$$\phi = \phi_{\mathrm{el}} + \phi_{\mathrm{ion}}$$

For the electronic phase use "Berry phase (rad) [-pi … +pi] spin up+dn" value. For the ionic phase use "Total ionic phase wrap. (rad)" value.

Evaluate the phase change

$$\Delta\phi = \phi(\mathrm{perturbed}) - \phi(\mathrm{unperturbed})$$

It should be about -0.034 rad. Compute the effective charge Z* of Nitrogen in GaN using a Berry phases and the "shortcut" expression

$$Z^*_{ii} = \frac{\Delta\phi_i}{2\pi\Delta u_i}$$

Here Δ*u* is the displacement in fractional coordinates. The equation applies to the case of one atom displaced. In our case, we need to take into account that 2 N-atoms were shifted.

Compare computed Z* with the literature value of -2.73 [Volume 44D of the series Landolt-Börnstein - Group III Condensed Matter pp 420-423, "GaN: effective charge, dielectric constants" by D. Strauch]

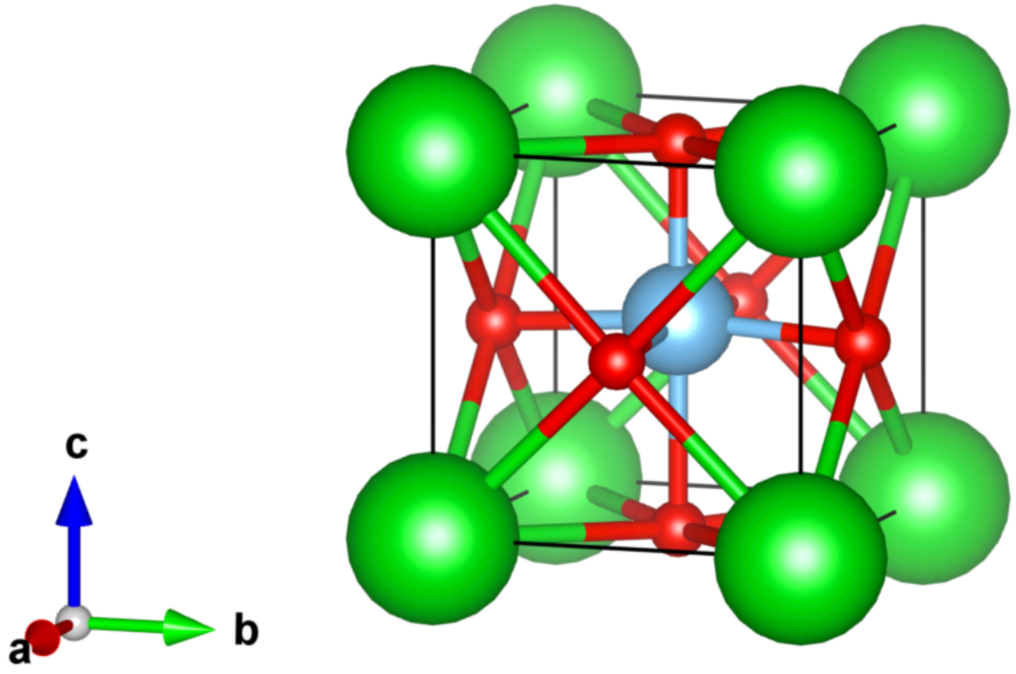
Spontaneous polarization $BaTiO_3$

WIEN2k + + BerryPI

Work flow

w2web Create a new session “BaTiO3ncm” with the working directory “…/BaTiO3/BaTiO3ncm”

w2web Construct a structure file (we begin with the lowest-symmetry non-centrosymmetric structure fi



4-atoms (Ba, Ti & 2O) per unit cell
Body-centred tetragonal lattice (spacegroup #99 P4mm
Cell size (Bohr): a = b = 7.547566; c = 7.626934
Angles: α = γ = β = 90°
Coordinates:
Ba (0 0 0)
Ti (1/2 1/2 0.51517436)
O (1/2 1/2 0.97356131)
O (1/2 0 0.48343742)
[You will see 1 additional equivalent atom after you save the structure O (0 1/2 0.48343742)]

Xcrysden Visualize the structure and compare to the one shown here

w2web Initialize SCF calculation (GGA-PBE) with a small (~3%) RMT reduction, RKMAX=7, and 230 k-points in the whole BZ.

Perform regular SCF calculation with default convergence parameters (~12 iterations). You are welcome to specify additional convergence criteria using `-ec` and `-cc` flags. It is the user's responsibility to check the convergence with respect to the quantity of interest (polarization in this case).

```
$   run_lapw
```

Run Berry phase calculation with a 6x6x6 k mesh. Note the “TOTAL POLARIZATION (C/m2)” along Z.

```
$    berrypi -k 6 6 6
```

A typical mistake is to take the calculated polarization as the spontaneous polarization. The spontaneous polarization is measured with respect to a reference structure, which is a centrosymmetric structure in this case.

w2web Create another session “BaTiO3cm” with the working directory “…/BaTiO3/BaTiO3cm”

Copy files to the new directory while you are still in “…/BaTiO3/BaTiO3ncm” folder

```
$    cp * ../BaTiO3cm
```

Change to the newly created folder

```
$    cd ../BaTiO3cm
```

Rename all BaTiO3ncm.* files to BaTiO3cm.* files

```
$    rename_files BaTiO3ncm BaTiO3cm
```

w2web Edit the “BaTiO3cm” structure file without changing the symmetry operations (do not change RMT radii)

Coordinates:
Ba (0 0 0)
Ti (1/2 1/2 1/2)
O1 (1/2 1/2 0)
O2 (1/2 0 1/2)
(0 1/2 1/2)

Restore original k-mesh taking into account the symmetry with 230 k-points (shifted)

```
$    x kgen
```

Initialize the electron density for the new structure. (We do not want to fully initialize the calculation in order to preserve the former symmetry operations.)

```
$    x dstart
```

Perform standard SCF calculation

```
$    run_lapw
```

Run Berry phase calculation with a 6x6x6 k mesh. Note the "TOTAL POLARIZATION (C/m2)" along Z.

```
$    berrypi -k 6 6 6
```

Spontaneous polarization is obtained by taking a difference in polarization between distorted and the reference structures, i.e., P(BaTiO3ncm) and P(BaTiO3cm):

$$P_s = \left|P_{\text{non-centr.}} - P_{\text{centr.}}\right|$$

You should get Ps ~ 0.31 C/m^2. The experimental value is Ps = 0.26 C/m^2 for the tetragonal phase of $BaTiO_3$ at T ~ 180 K [H. H. Wieder, Phys. Rev. **99**, 1161 (1955)].

Chirality of Weyl points in TaAs semimetal

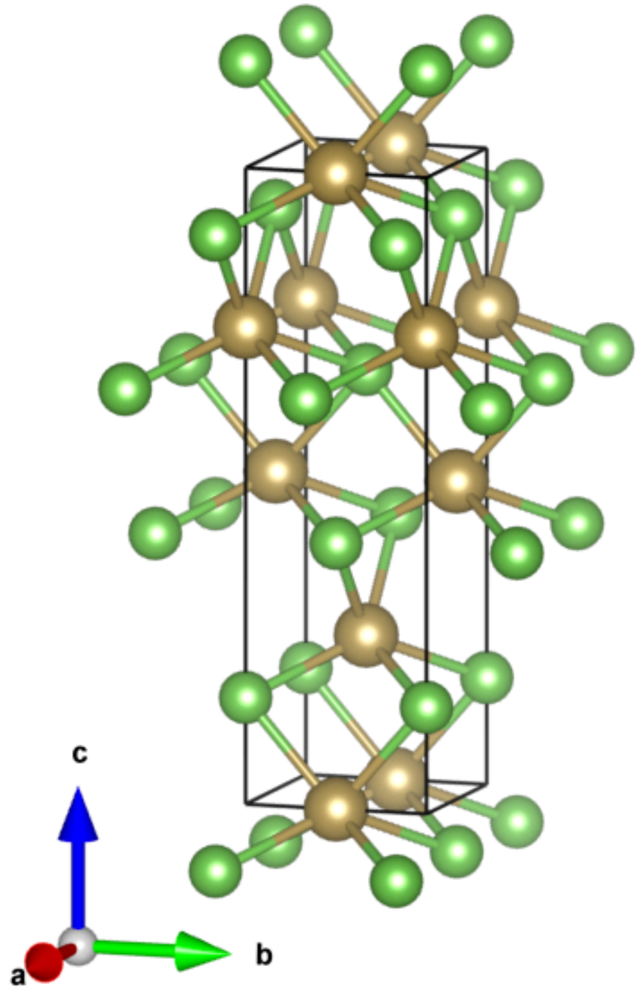
+

+

BerryPI
(WloopPHI.py)

Workflow

w2web Construct a structure file



2-atoms (Ta & As) per unit cell
Body-centred tetragonal lattice (spacegroup #109 $I4_1md$)
Cell size (Bohr): a = b = 6.494611; c = 22.004349
Angles: $\alpha = \gamma = \beta = 90°$
Coordinates:
Ta (0 0 3/4)
As (0 0 0.1677)
[You will see 2 additional equivalent atoms after you save the structure
Ta (0 1/2 0) and As (0 1/2 0.4177)]

Initialize SCF calculation (GGA-PBE) with a small (~3%) RMT reduction, RKMAX=7, and 300 k-points in the whole BZ.

Xcrysden Visualize the structure and compare to the one shown here

Perform regular SCF calculation with energy and change convergence of 0.0001 and 0.001, respectively (~9 iterations)

```
$    run_lapw -ec 0.0001 -cc 0.001
```

Save calculation, then initialize spin-orbit calculation (SOC) using all default parameters, and run SCF-SOC (~6 iterations)

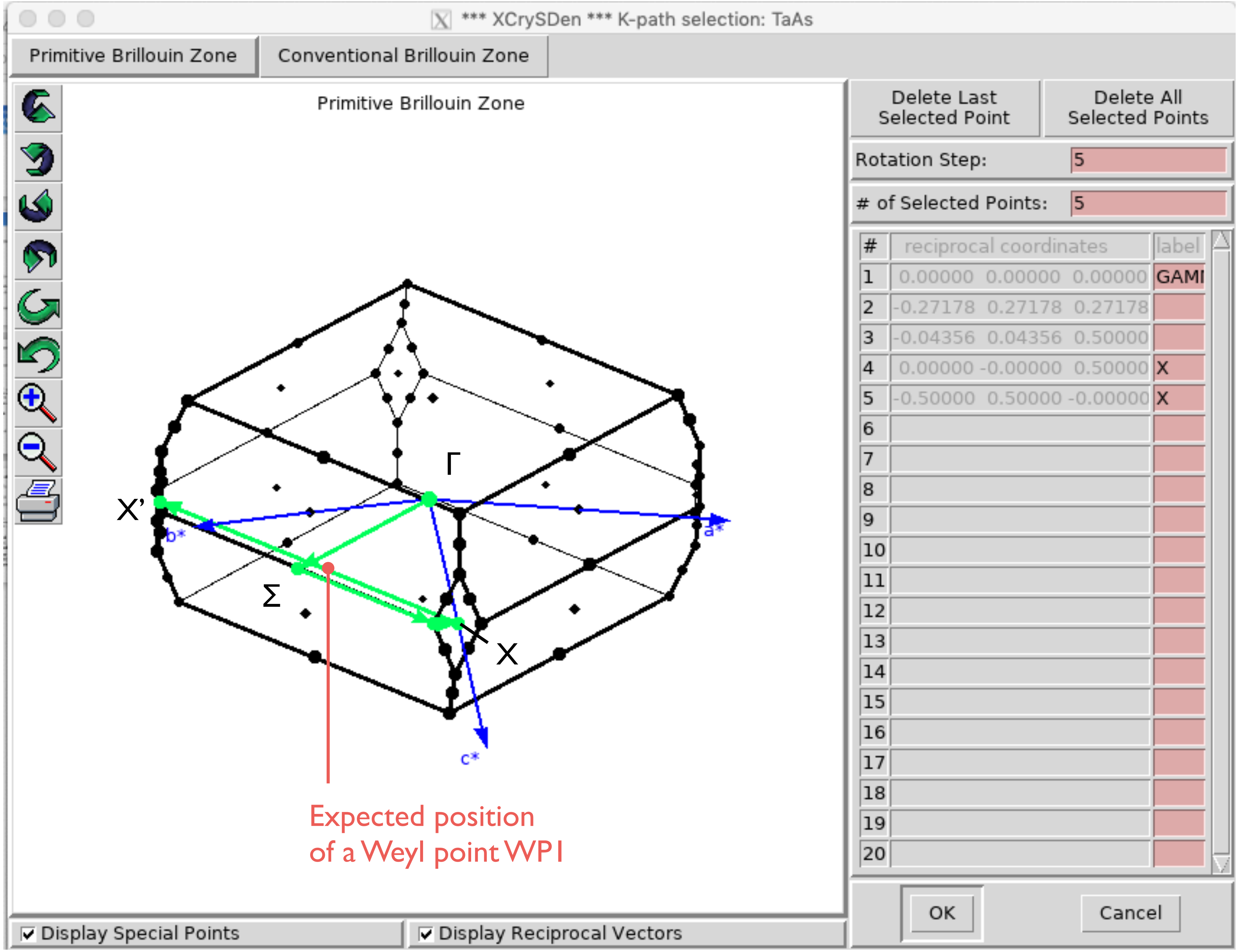
```
$    save_lapw -d noSOC

$    init_so_lapw

$    run_lapw -ec 0.0001 -cc 0.001 -so
```

Check the band gap in case.scf file. It should be about 0.087 eV

Xcrysden File > Open WIEN2k > Select k-path for band structure plot that goes through points listed on the screenshot, ask for 600 points along the path and save the k-point list as "case.klist_band". It is always good to check the case.klist_band file and make sure there are no "******" entries resulted from an insufficient format.



Expected position of a Weyl point WP1

Rotation Step: 5

# of Selected Points: 5

#	reciprocal coordinates	label
1	0.00000 0.00000 0.00000	GAM
2	-0.27178 0.27178 0.27178	
3	-0.04356 0.04356 0.50000	
4	0.00000 -0.00000 0.50000	X
5	-0.50000 0.50000 -0.00000	X
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		

☑ Display Special Points

☑ Display Reciprocal Vectors

Recalculate eigenvalues on the k-path selected

```
$   x lapw1 -band
```

```
$   x lapwso
```

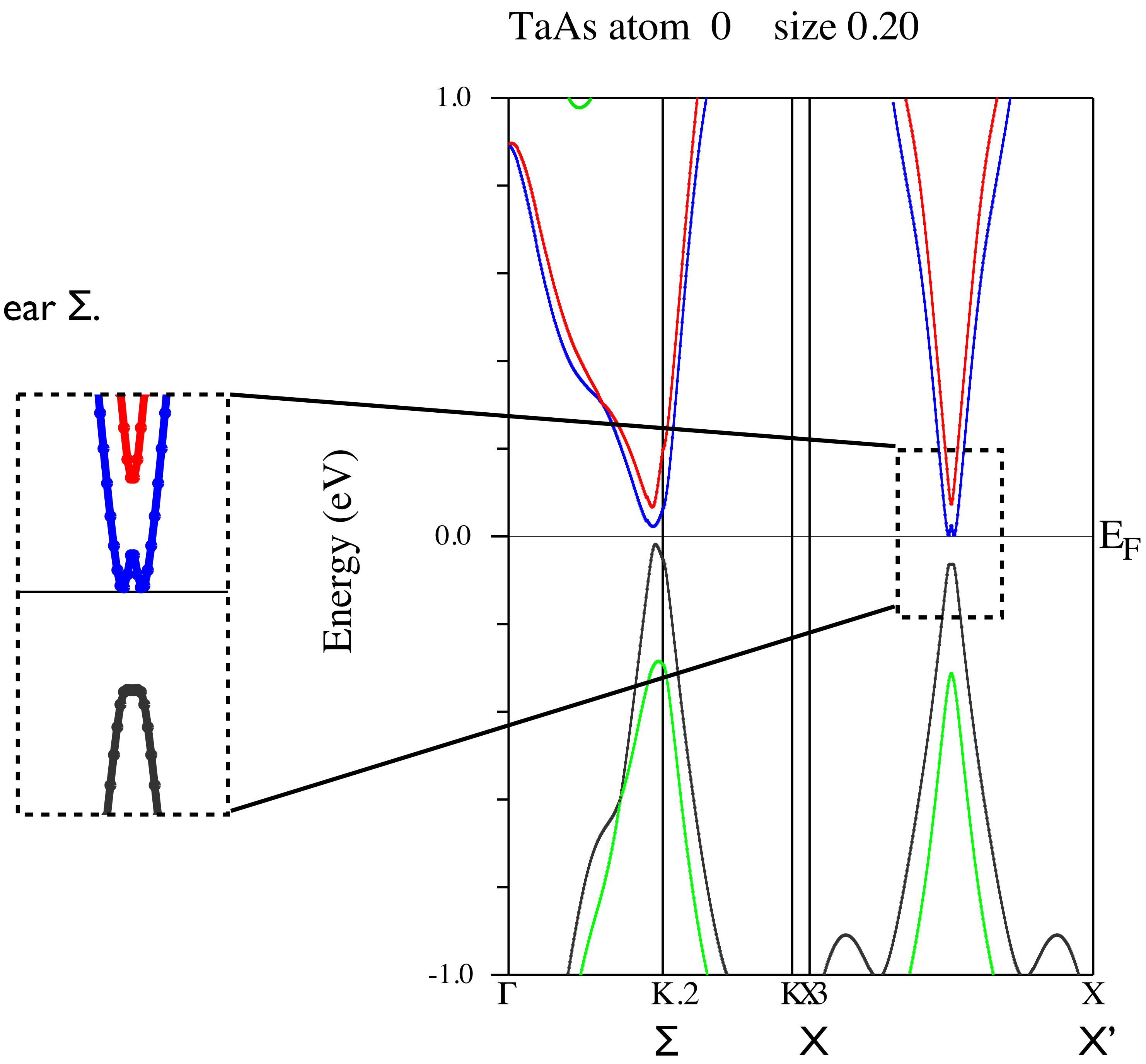
w2web Go to the "Bandstructure" menu and edit the case.insp file:
insert the Fermi energy of 0.801 Ry and set the energy range for plotting at [-1.0, 1.0] eV

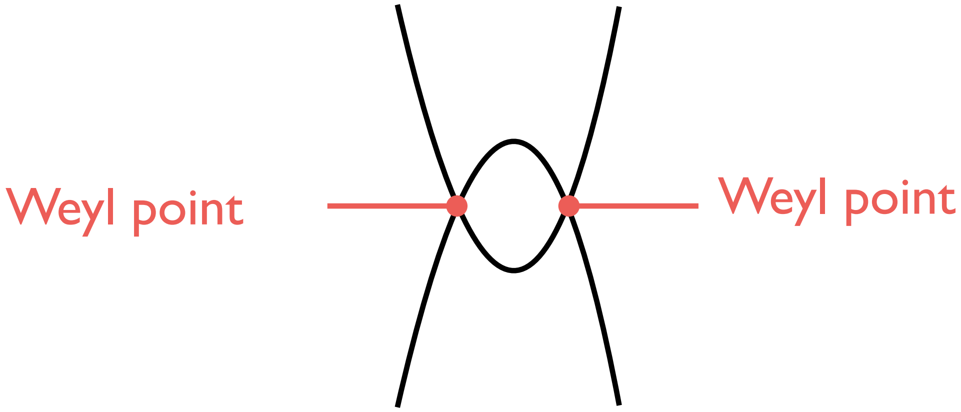
Run "spaghetti" with -so option

```
$   x spaghetti -so
```

w2web Plot the band structure. You will notice the band almost vanish as you approach Weyl points.

One set of Weyl points (nodes) is located near Σ. It is difficult to design the k-path such that it goes *exactly* through the Weyl point. It is because Weyl points usually do not coincide with high-symmetry points of BZ and do not lie on high-symmetry directions. (See figure at the end of this tutorial.)



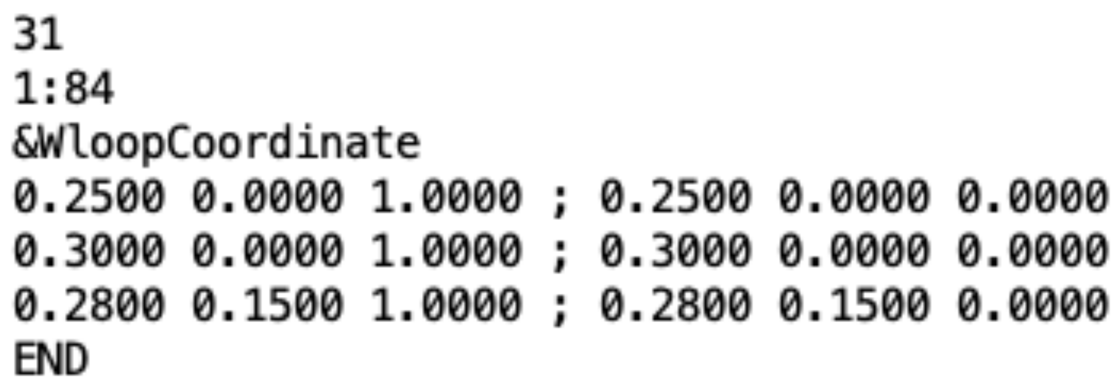


Next we calculate a Berry phase on a series of closed Wilson loops. It corresponds to the Berry flux through the loop.

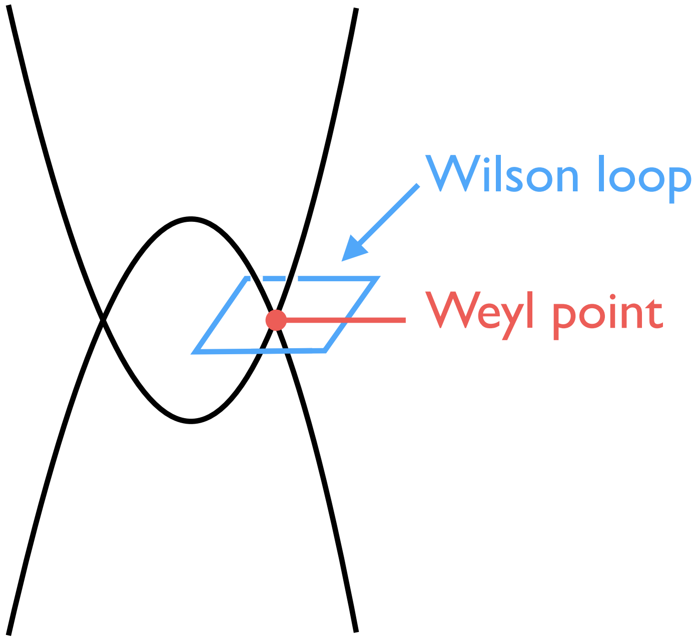
Determine the number of occupied bands in TaAs.scf file (it should be 84).

```
$   grep :BAN *scf
```

Using your favourite editor create the input file `Wloop.in`. within the "case" folder that instructs BerryPI to calculate the cumulative Berry phase for bands 1-84 (all occupied) for a series of Wilson loops. The loops are defined by 3 k-points in the reciprocal space. Reciprocal coordinates of the first loop are [0.25 0 1; 0.30 0 1; 0.28 0.15 1]. Coordinates of the last loop are [0.25 0 0; 0.30 0 0; 0.28 0.15 0]. Intermediate loops (31 in total) will be created by interpolating between k-point coordinates of the first and the last loop.



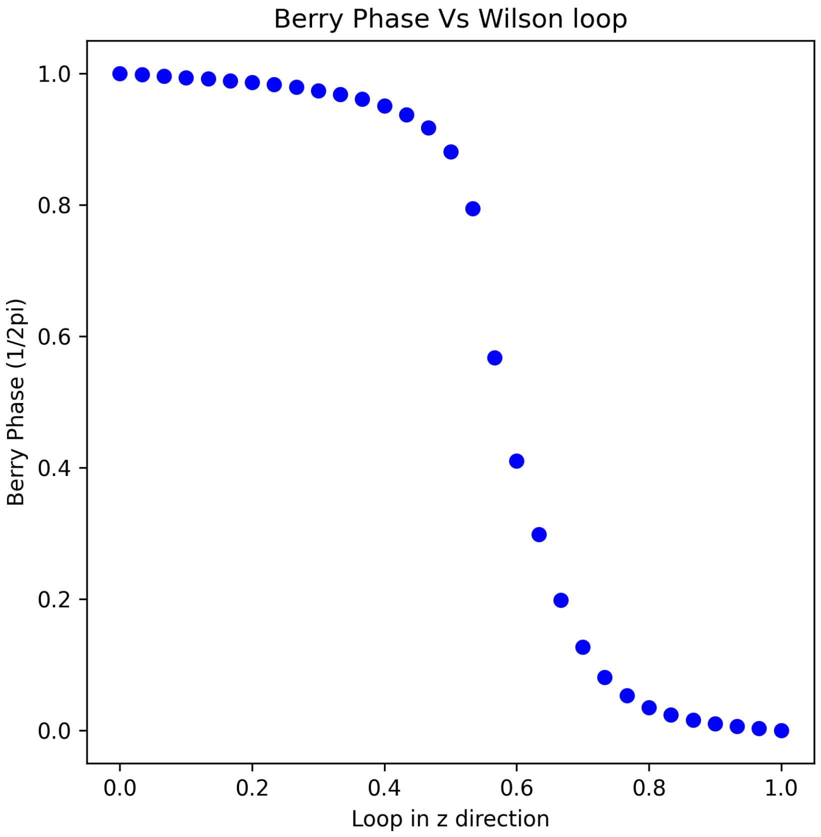
```
31
1:84
&WloopCoordinate
0.2500 0.0000 1.0000 ; 0.2500 0.0000 0.0000
0.3000 0.0000 1.0000 ; 0.3000 0.0000 0.0000
0.2800 0.1500 1.0000 ; 0.2800 0.1500 0.0000
END
```

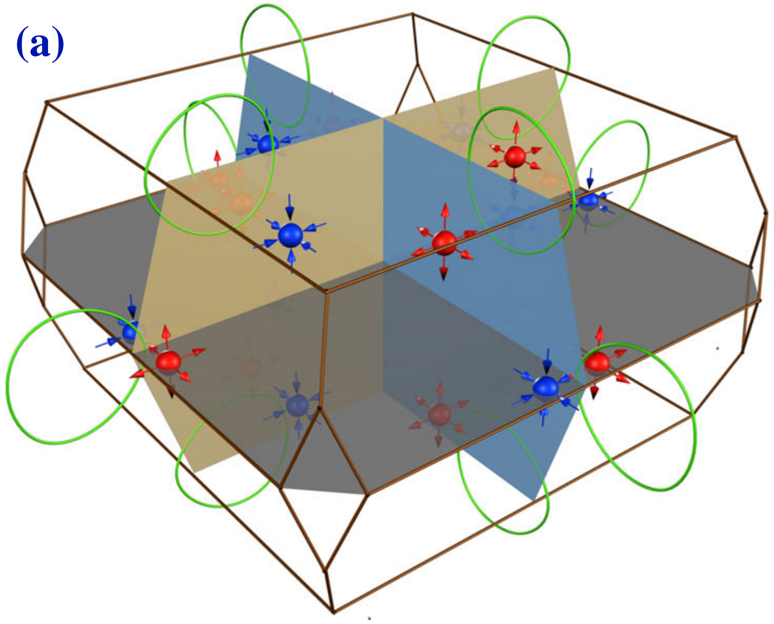


Run berry phase calculation via an auxiliary Python script `WloopPHI.py` (the SOC is implied).

```
$   python $WIENROOT/SRC_BerryPI/BerryPI/WloopPHI.py Wloop.in
```

The file PHI.dat contains the output table with Berry phases. The Berry phase makes a 2π revolution (see next page) as the loop traverses from the $k_z = 0$ plane to the $k_z = 1$ plane.





Weng et al.,
Physical Review X 5, 011029 (2015)

Based on the graph above, determine k_z coordinate of the Weyl point.

What will happen with the graph if the chirality of the Weyl point will change to opposite?

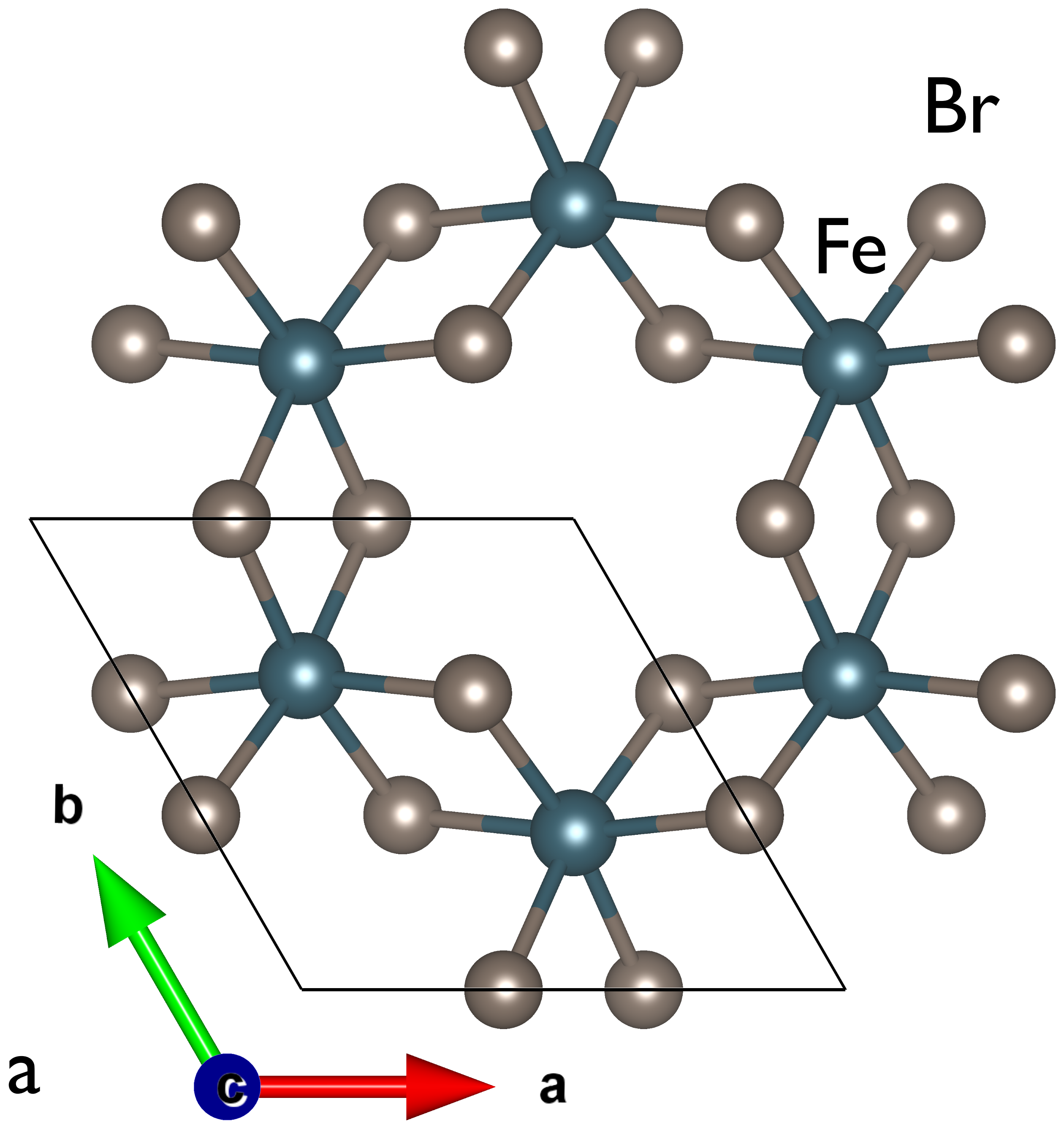
What will happen with the graph if the chirality of the Weyl point will double or triple?

What will happen with the graph if two Weyl points are encountered along the way?

Chern topological invariant and Berry curvature maps of $FeBr_3$ and MoS_2

+BerryPI (CherN.py)

Workflow: Chern number of 2D-$FeBr_3$



w2web Construct a structure file (spin-polarization: yes, 3% RMT reduction)

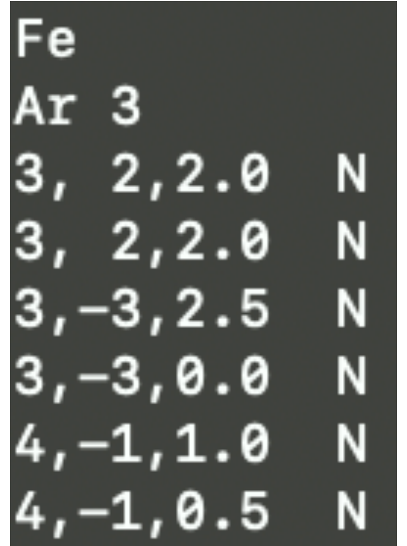
2-atoms (Fe & Br)
(spacegroup #162 P-$\overline{31}$/m)
Cell size (Å): a = b = 6.29708; c = 19.82860
Angles: $\alpha = \beta = 90°$ $\gamma = 120°$
Coordinates:
Fe (1/3 2/3 1/2)
Br (0.62895 0.0 0.43062)
[You will see 6 additional equivalent atoms after you save the structure
Fe (2/3 1/3 1/2) and 5 Br]

Xcrysden Visualize the structure and compare to the one shown here. It should be noted that a vacuum (>10 Å) is added in the direction normal to the monolayer to avoid inter-layer interaction.

Generate the FeBr3.inst with the initial electronic configuration and spin occupation

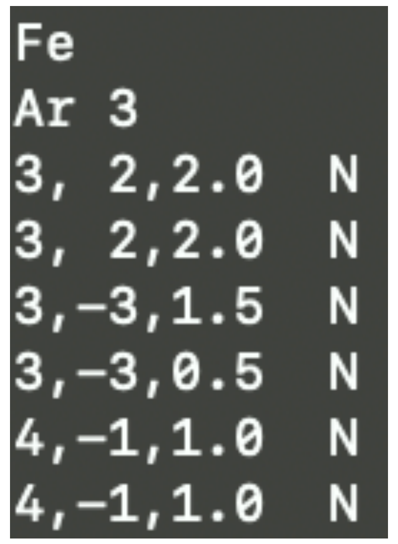
```
$     instgen_lapw
```

Modify the .inst file to have a predominant up (or down) spin on the Fe (d-orbitals) to give an initial magnetic moment.



```
Fe
Ar 3
3, 2,2.0  N
3, 2,2.0  N
3,-3,2.5  N
3,-3,0.0  N
4,-1,1.0  N
4,-1,0.5  N
```

→



```
Fe
Ar 3
3, 2,2.0  N
3, 2,2.0  N
3,-3,1.5  N
3,-3,0.5  N
4,-1,1.0  N
4,-1,1.0  N
```

Initialize SCF spin-polarized calculation with default values (PBE)

```
$     init_lapw -b -sp
```

Generate 9 x 9 x 1 mesh

```
$     x kgen; 0 ; 9 9 1
```

Perform regular SCF spin-polarized calculation with energy and change convergence of 0.0001 and 0.001, respectively (~17 iterations with 4 core k-point parallelization).

```
$     runsp_lapw -ec 0.0001 -cc 0.001 -p
```

Prepare `.machines` file that contains these 4 lines

```
1:localhost
1:localhost
1:localhost
1:localhost
```

Save calculation, then initialize spin-orbit calculation (SOC) using all default parameters, and run SCF-SOC

```
$     save_lapw -d noSOC

$     init_so_lapw;

$     x kgen -so; 0; 9 9 1

$     runsp_lapw -ec 0.0001 -cc 0.001 -so -p

$     save_lapw -d SOC
```

Xcrysden File > Open WIEN2k > Select k-path for band structure plot that goes through points listed on the screenshot, ask for 600 points along the path and save the k-point list as "case.klist_band". It is always good to check the case.klist_band file and make sure there are no "******" entries resulted from an insufficient format.

Recalculate eigenvalues on the k-path selected

```
$        x lapw1 -band
```

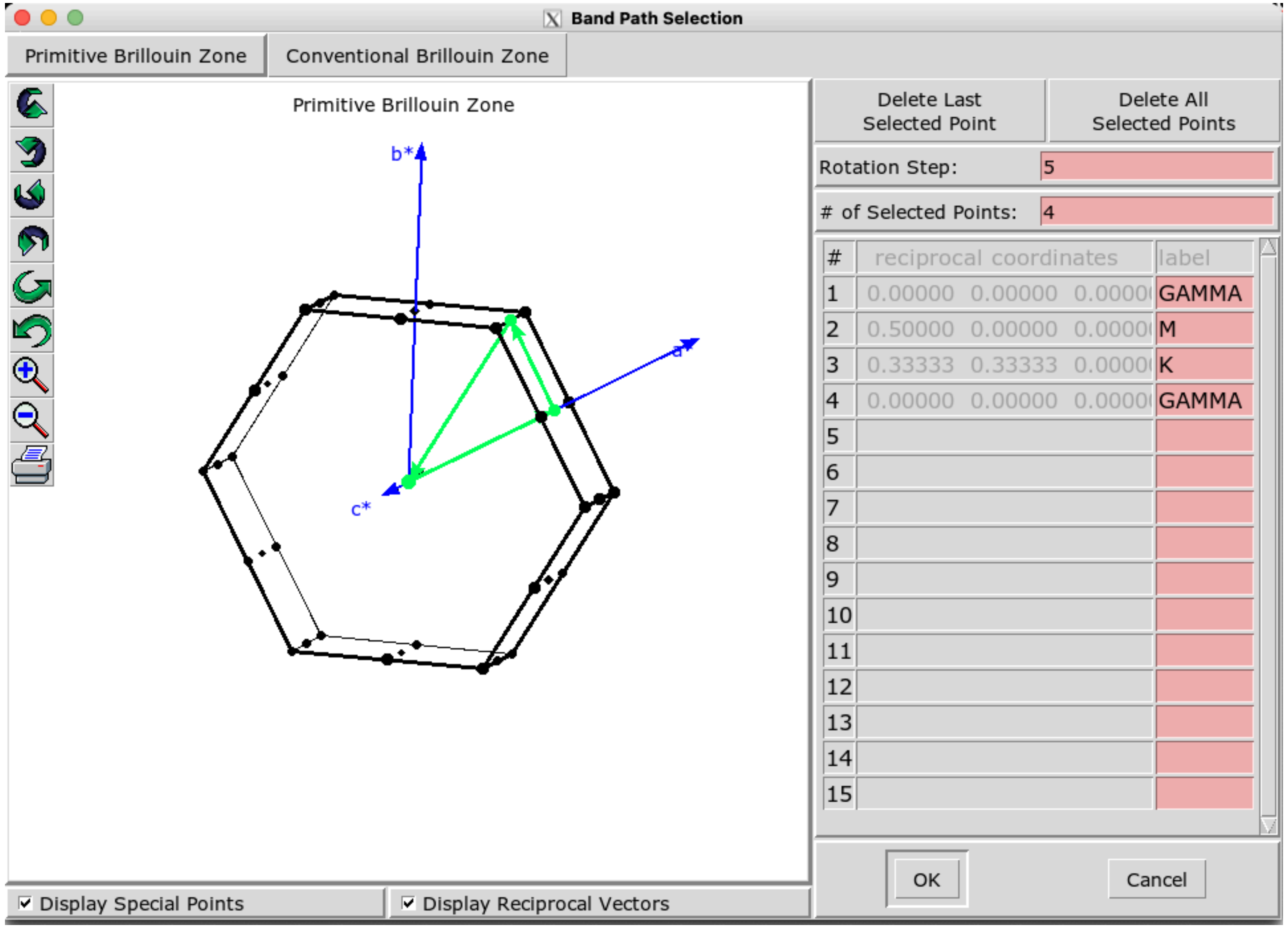
```
$        x lapwso
```

w2web Go to the "Bandstructure" menu and edit the case.insp file:

insert the Fermi energy and set the energy range for plotting at [-1.0, 1.0] eV Run "spaghetti" with -so option

```
$        x spaghetti
```

```
$        x spaghetti -so
```



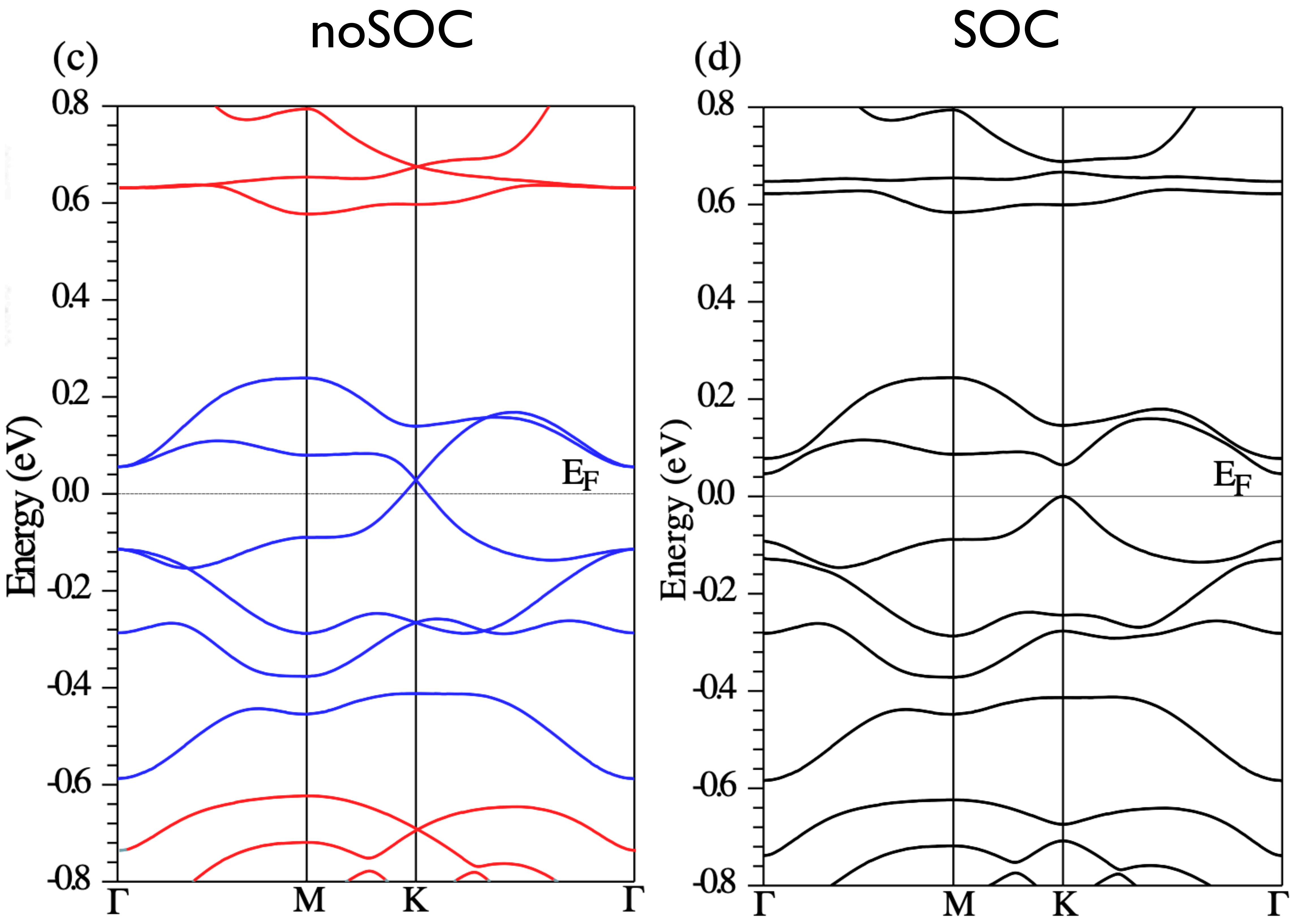
Rotation Step: 5

# of Selected Points: 4

#	reciprocal coordinates	label
1	0.00000 0.00000 0.0000	GAMMA
2	0.50000 0.00000 0.0000	M
3	0.33333 0.33333 0.0000	K
4	0.00000 0.00000 0.0000	GAMMA
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

w2web Plot both the non-SOC and SOC band structures.

It can be seen that there is a Spin-Orbit coupling induced gap opening of the Dirac cone.



Next we calculate the Chern number on the z=0 plane of the BZ.

Determine the number of occupied bands in FeBr3.scf (should be 130)

```
$       grep :BAN SOC/*scf
```

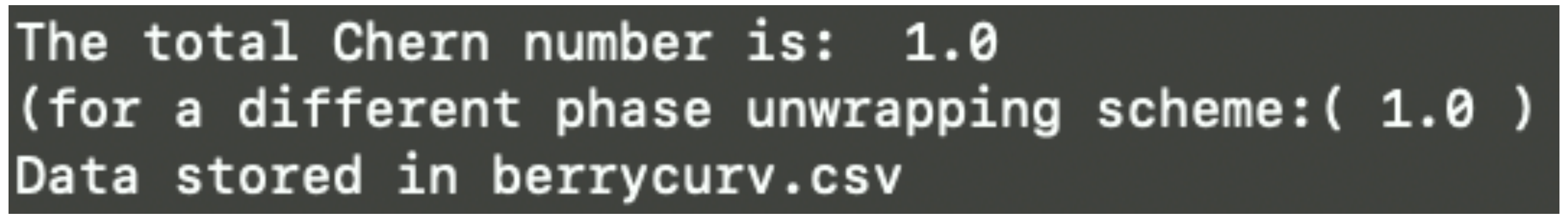
Modify the inputs for the CherN.py script (copy from $WIENROOT/SRC_BerryPI/BerryPI the CherN.py file) :

```
bands = [1,130] # band range
n_1 = 4          # discretization by (n_1 -1) in the 1 direction
n_2 = 4          # discretization by (n-2 - 1) in the 2 direction
plane_dir = 3    # direction normal to the plane  (1 or 2 or 3)
plane_height = 0.0          # value of the constant plane
boundary = [0, 1.0 ,  0 , 1.0] #boundary selection: ex if plane_dir = 3 -> [1min,1max,2min,2max]
spinpolar = True            # spin polarized
orbital = False             # additional orbital potential
parallel = True             # parallel calculation
```

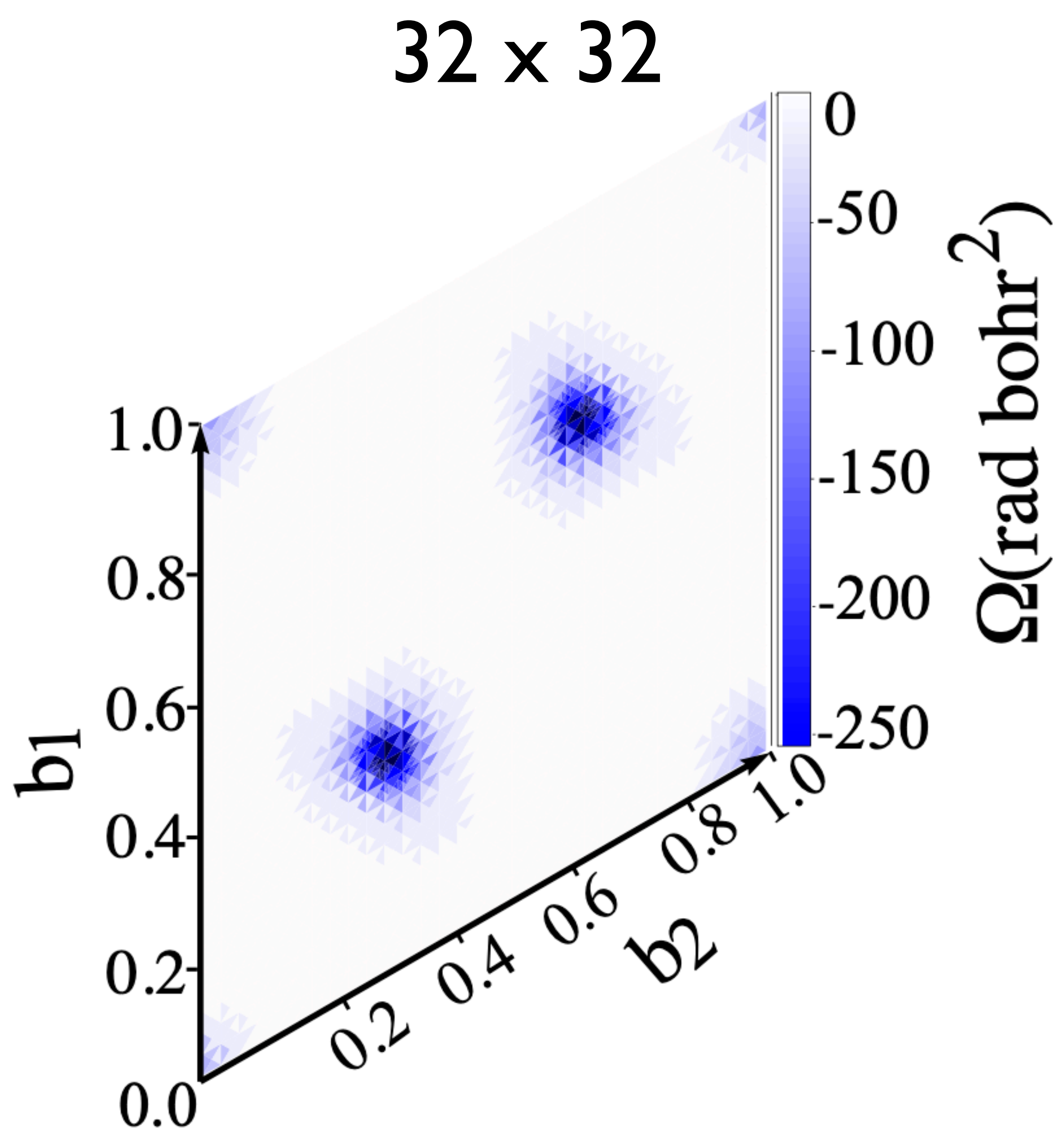
Run CherN.py

```
$        python3 CherN.py
```

The obtained result for the Chern topological invariant should be 1, which means there is one conduction edge state. The berry curvature map is presented in the berrycurv.pdf file; a better quality image can be obtained with higher discretization values.



```
The total Chern number is:  1.0
(for a different phase unwrapping scheme:( 1.0 )
Data stored in berrycurv.csv
```

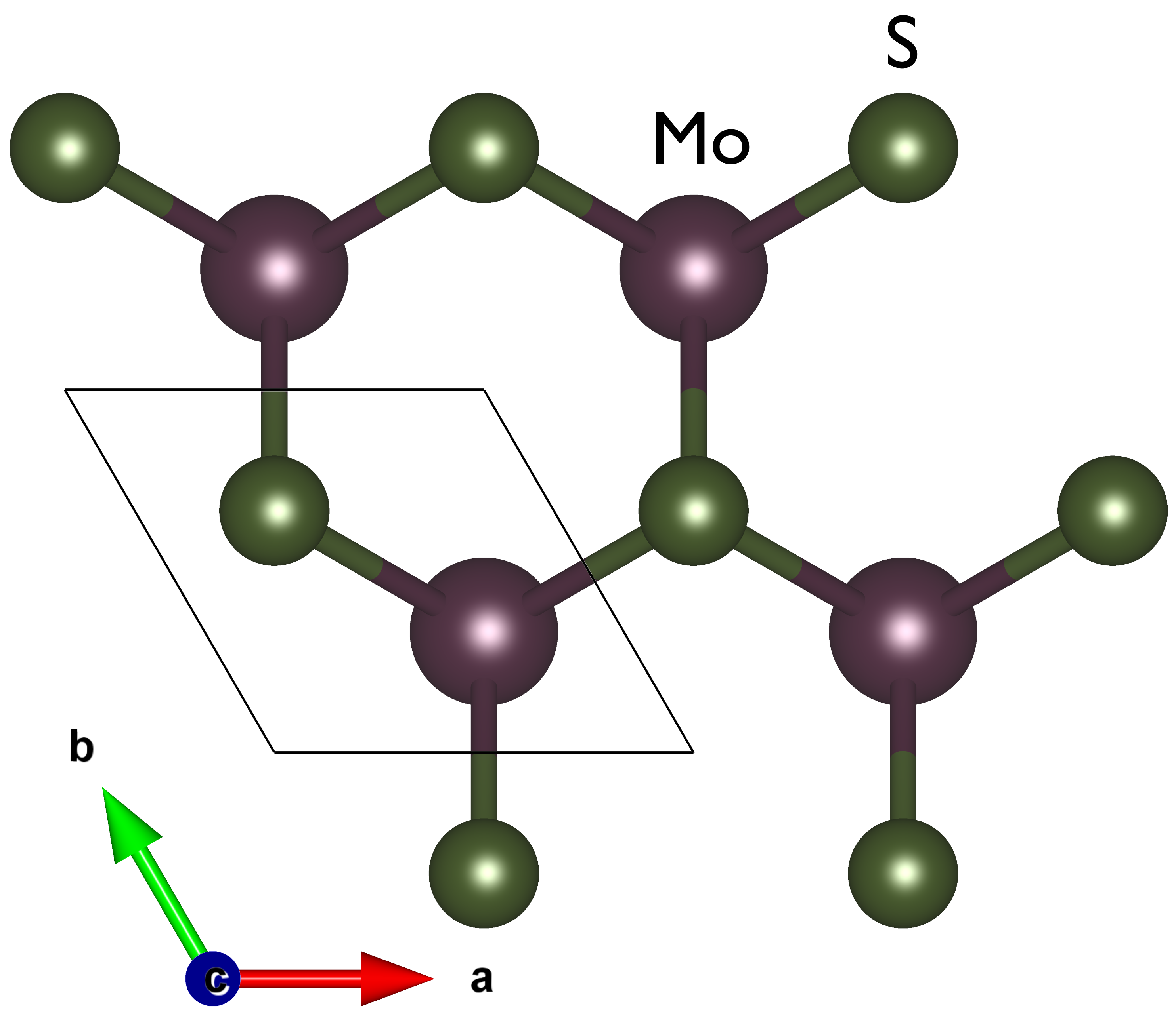


Workflow: Berry curvature map of 2D-MoS_2

w2web Construct a structure file (3% RMT reduction)

2-atoms (Mo & S)
(spacegroup #187 $P\bar{6}m2$)
Cell size (Å): a = b = 3.16; c = 12.29
Angles: $\alpha = \beta = 90°$ $\gamma = 120°$
Coordinates:

Mo (0 0 0)
S (2/3 1/3 0.87100)
[You will see 1 additional equivalent atom after you save the structure
S (2/3 1/3 0.12900)]



Xcrysden Visualize the structure and compare it to the one shown here.

It should be noted that a vacuum (>10 Å) is added in the direction normal to the monolayer to avoid inter-layer interaction.

Initialize SCF-SOC calculation with default values (GGA-PBE)

```
$     init_lapw -b

$     init_so_lapw
```

Generate 9 x 9 x 1 mesh

```
$     x kgen -so; 0; 9 9 1
```

run SCF-SOC calculation

```
$     run_lapw -ec 0.0001 -cc 0.001 -so
```

Save calculation

```
$     save_lapw -d MoS2-SOC
```

Xcrysden File > Open WIEN2k > Select k-path for band structure plot that goes through points listed on the screenshot, ask for 600 points along the path and save the k-point list as "case.klist_band". It is always good to check the case.klist_band file and make sure there are no "******" entries resulted from an insufficient format.

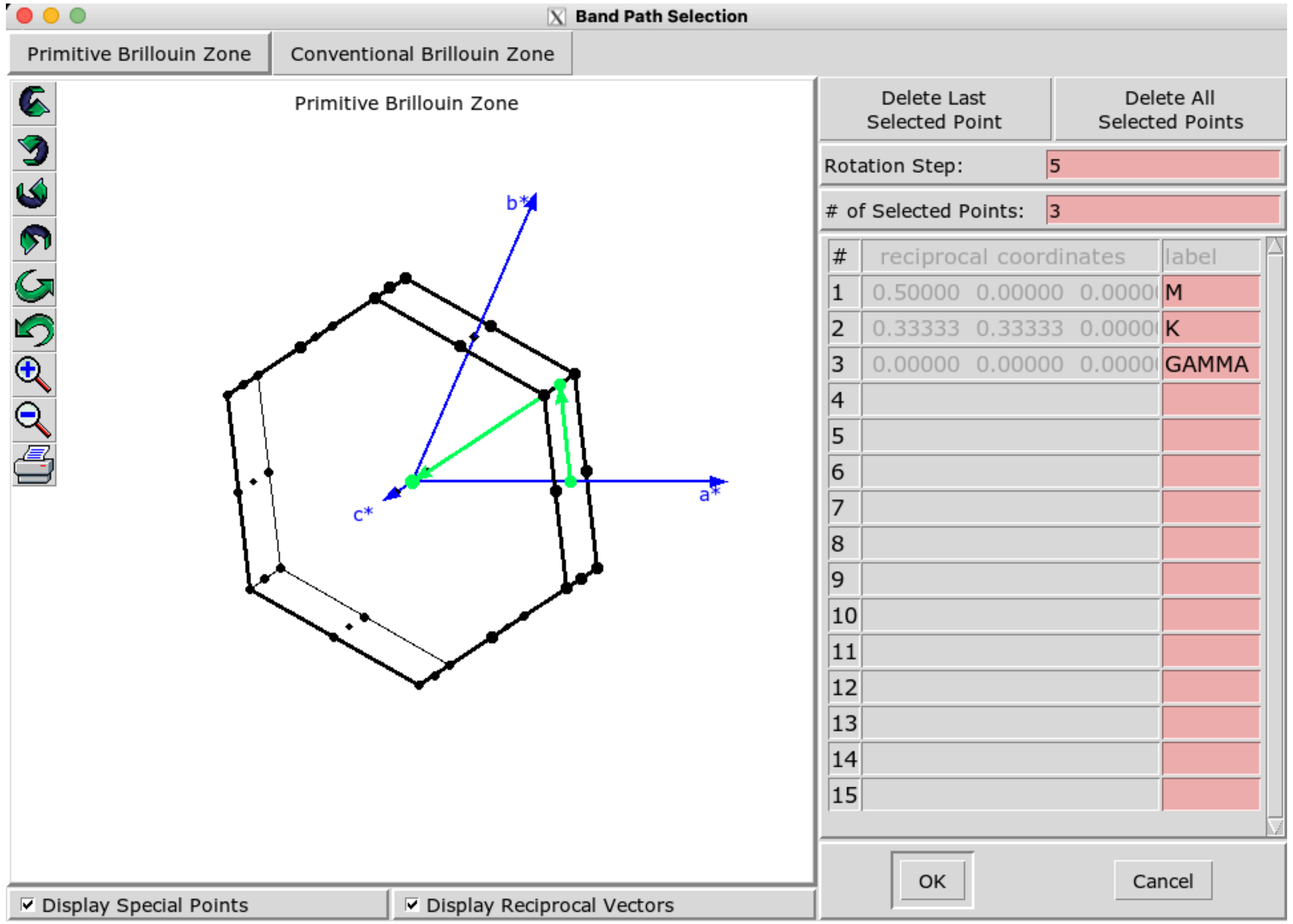
Recalculate eigenvalues on the k-path selected

```
$       x lapw1 -band
$       x lapwso
```

w2web Go to the "Bandstructure" menu and edit the case.insp file:

insert the Fermi energy and set the energy range for plotting at [-1.0, 1.0] eV Run "spaghetti" with -so option

```
$       x spaghetti -so
```



Rotation Step: 5

# of Selected Points: 3

#	reciprocal coordinates	label
1	0.50000 0.00000 0.0000	M
2	0.33333 0.33333 0.0000	K
3	0.00000 0.00000 0.0000	GAMMA
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

☑ Display Special Points
☑ Display Reciprocal Vectors

w2web Plot both the electronic band structure.

Next we calculate the Berry curvature map on the z=0 plane of the BZ.

Determine the number of occupied bands in MoS2.scf (should be 26)

```
$        grep :BAN SOC/*scf
```

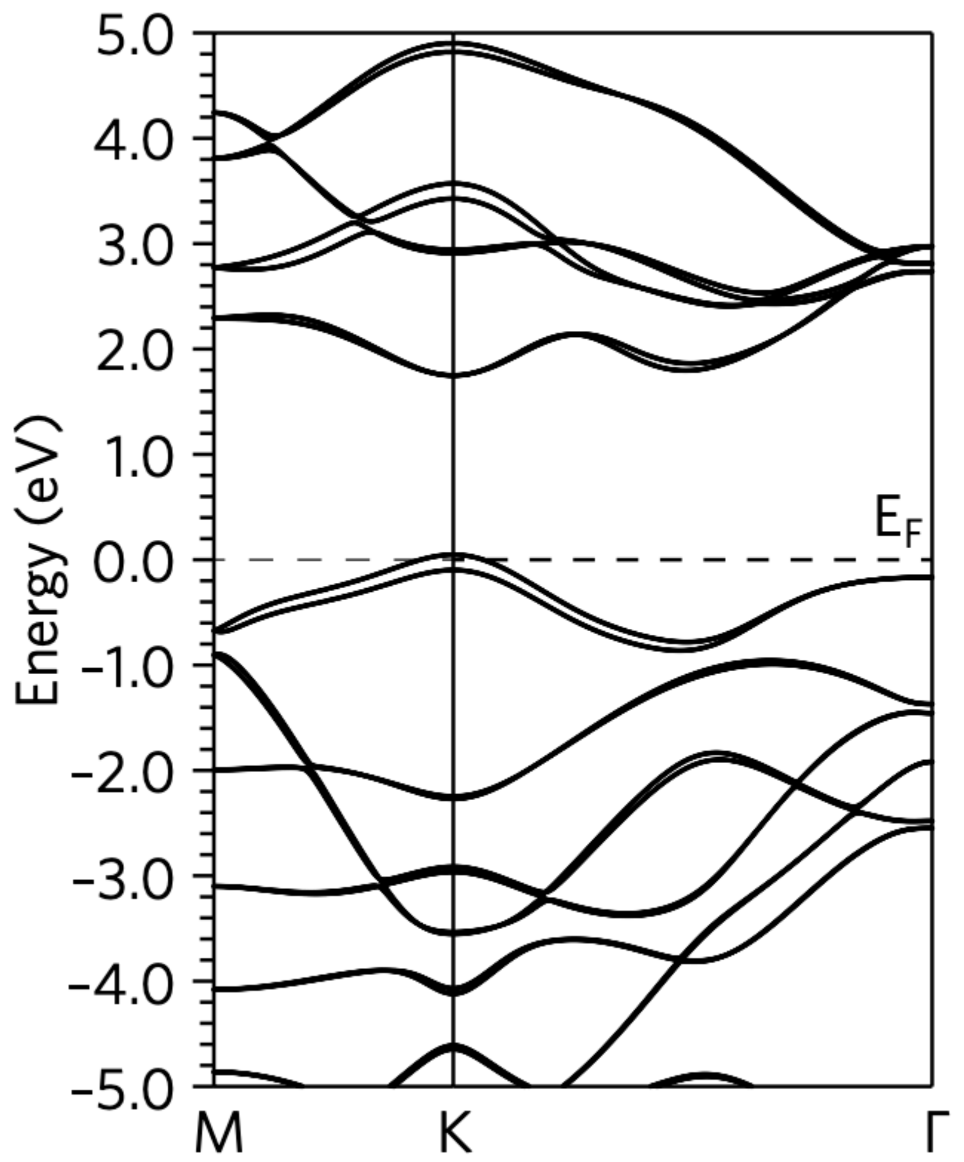
Modify the inputs for the CherN.py script (copy from $WIENROOT/SRC_BerryPI/BerryPI the CherN.py file)

```
bands = [1,26] # band range
n_1 = 6         # discretization by (n_1 -1) in the 1 direction
n_2 = 6         # discretization by (n-2 - 1) in the 2 direction
plane_dir = 3  # direction normal to the plane  (1 or 2 or 3)
plane_height = 0.0              # value of the constant plane
boundary = [0 , 1.0 , 0 , 1.0] #boundary selection
spinpolar = False               # spin polarized
orbital = False                 # additional orbital potential
parallel = False                # parallel calculation
```

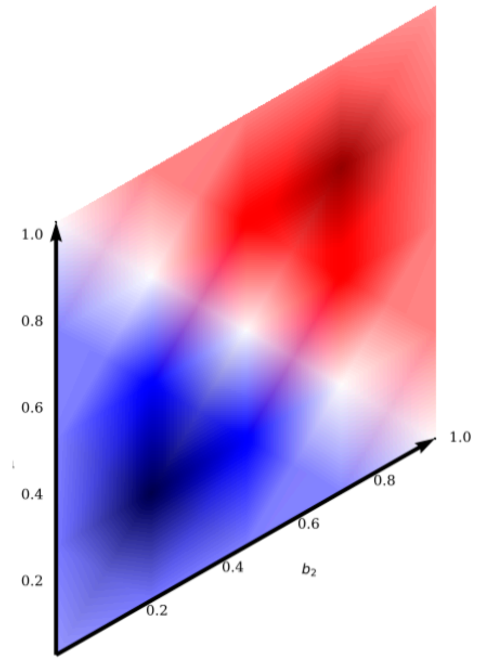
Run CherN.py

```
$        python3 CherN.py
```

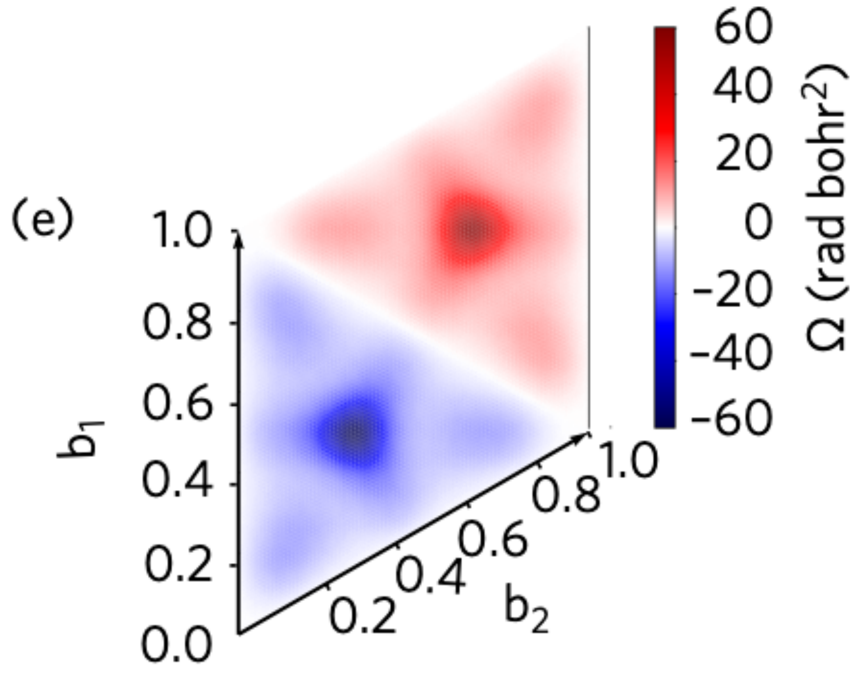
The berry curvature map is presented in the berrycurv.pdf file; a better quality image can be obtained with higher discretization values. The Chern number is 0 as expected as the non-magnetic system preserves the Time-Reversal Symmetry

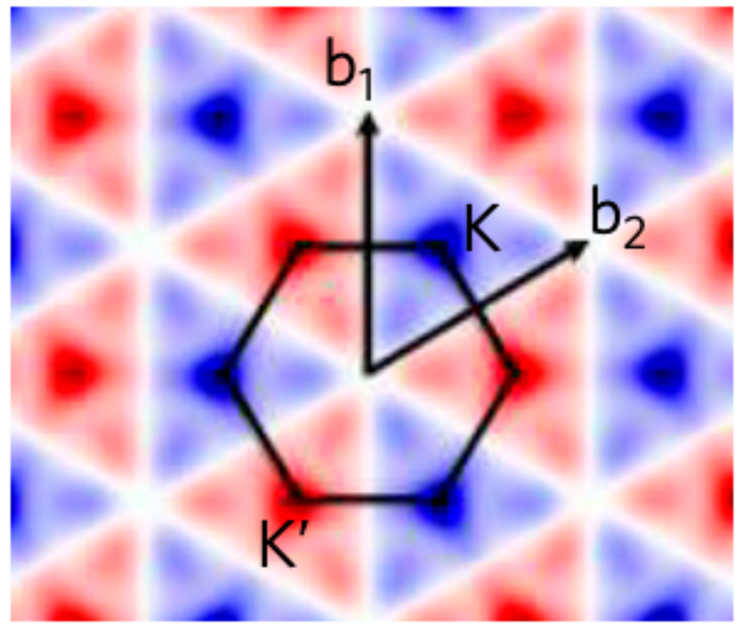


5 x 5



50 x 50





Z_2 topological invariant of Bi_2Se_3

+BerryPI (wcc.py)

Workflow: Hybrid Wannier Charge Centres

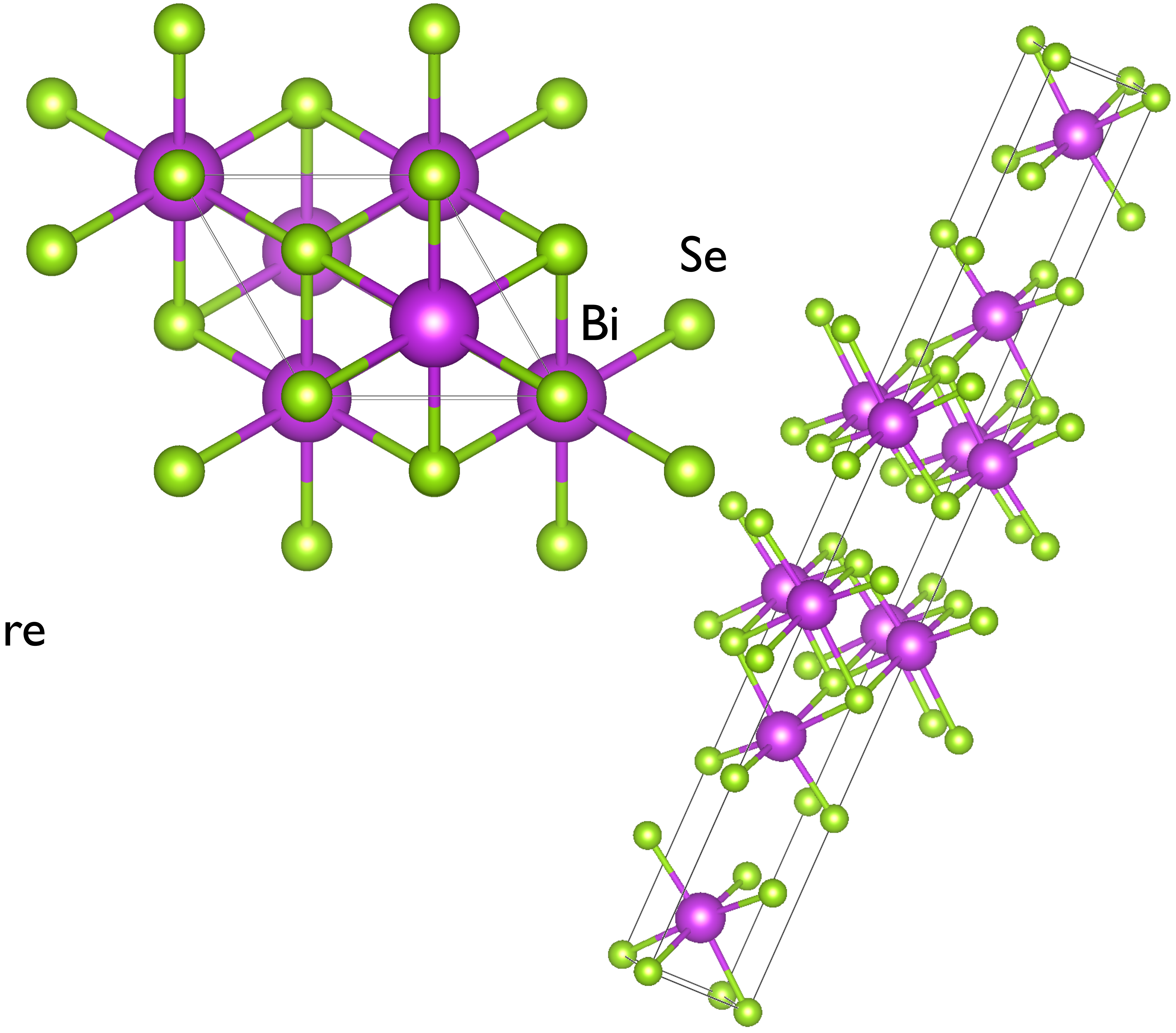
w2web Construct a structure file (3% RMT reduction)

3 Inequivalent-atoms (Bi & Se)
(Spacegroup #166 $R\bar{3}m$)

Cell size (Bohr): a = b = 7.829135; c = 54.114197
Angles: α = β = 90° γ = 120°
Coordinates:

Bi	(0.59920	0.59920	0.59920)
Se	(0.0	0.0	0.0)
Se	(0.78830	0.78830	0.78830)

[You will see 2 additional equivalent atoms after you save the structure Bi (0.0 0.0 0.40080) and 2 Se (0.0 0.0 0.21170)]



Xcrysden Visualize the structure and compare to the one shown here

Initialize SCF-SOC calculation (GGA-PBE)

```
$    init_lapw -b -rkmax 8

$    init_so_lapw
```

Generate 5 x 5 x 5 mesh

```
$     x kgen -so; 0; 5 5 5; 1
```

run SCF-SOC calculation

```
$    run_lapw -ec 0.0001 -cc 0.001 -so
```

Save calculation

```
$    save_lapw -d Bi2Se3-SOC
```

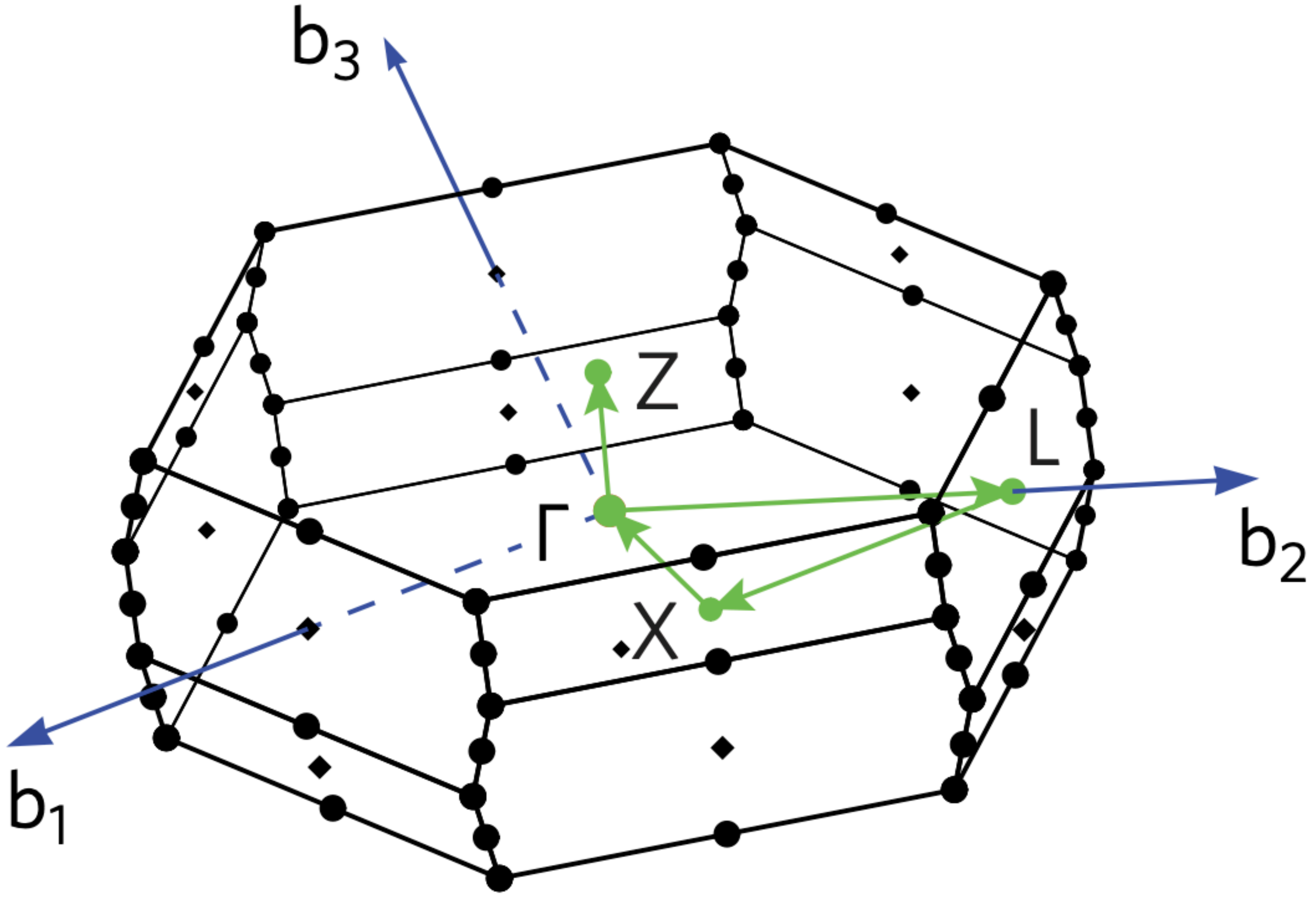
Xcrysden File > Open WIEN2k > Select k-path for band structure plot that goes through points listed on the screenshot, ask for 600 points along the path and save the k-point list as "case.klist_band". It is always good to check the case.klist_band file and make sure there are no "******" entries resulted from an insufficient format.

Recalculate eigenvalues on the k-path selected

```
$      x lapw1 -band
$      x lapwso
```

w2web Go to the "Bandstructure" menu and edit the case.insp file:

insert the Fermi energy and set the energy range for plotting at [-2.0, 2.0] eV Run "spaghetti" with -so option



```
$      x spaghetti -so
```

w2web Plot the electronic band structure.

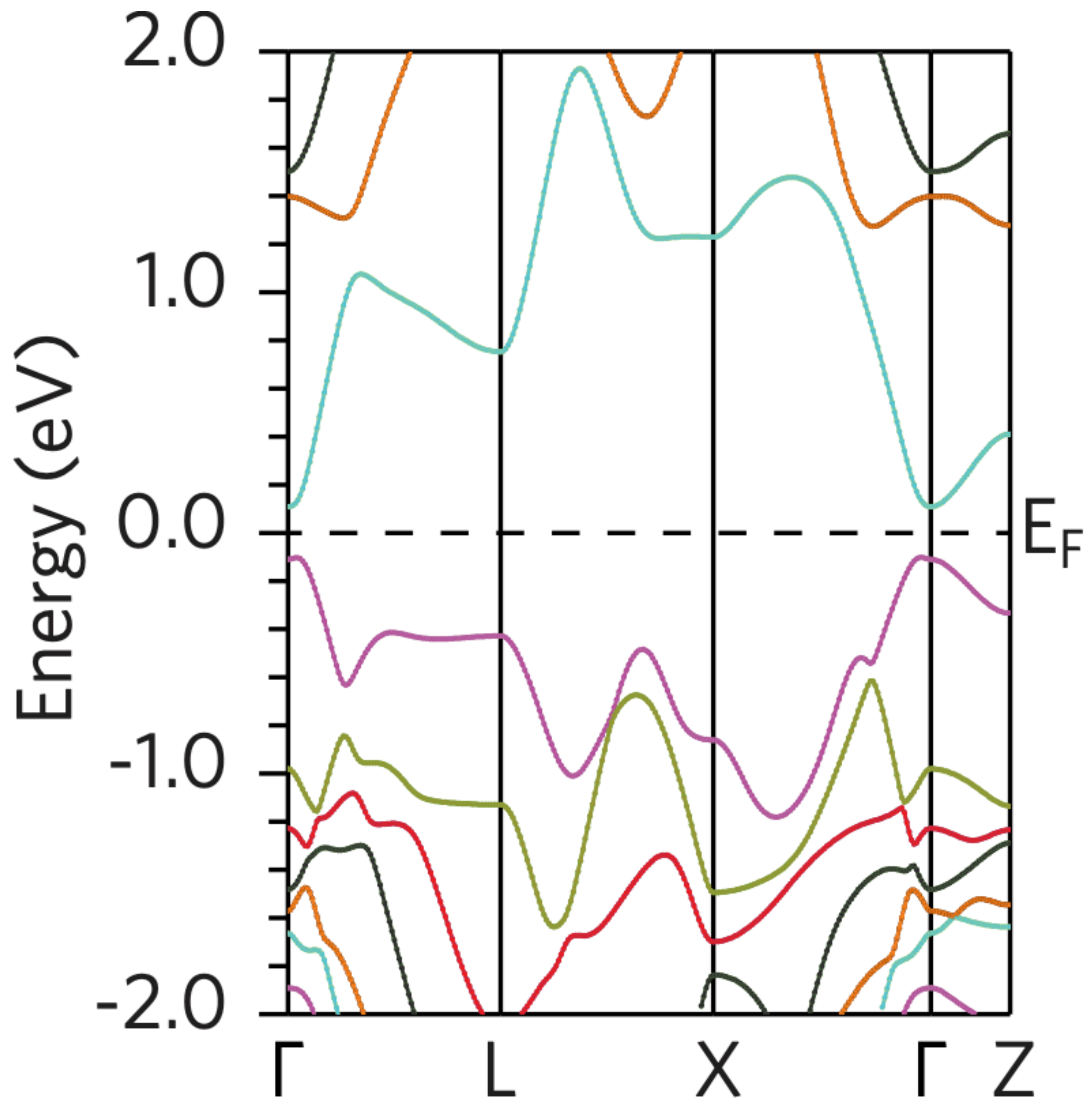
Next we investigate the evolution of Hybrid Wannier Charge Centres between TRIM points in order to differentiate between a topologically trivial and non trivial phase. We select the top group of valence bands, which are separated by a ca. 3 eV gap.

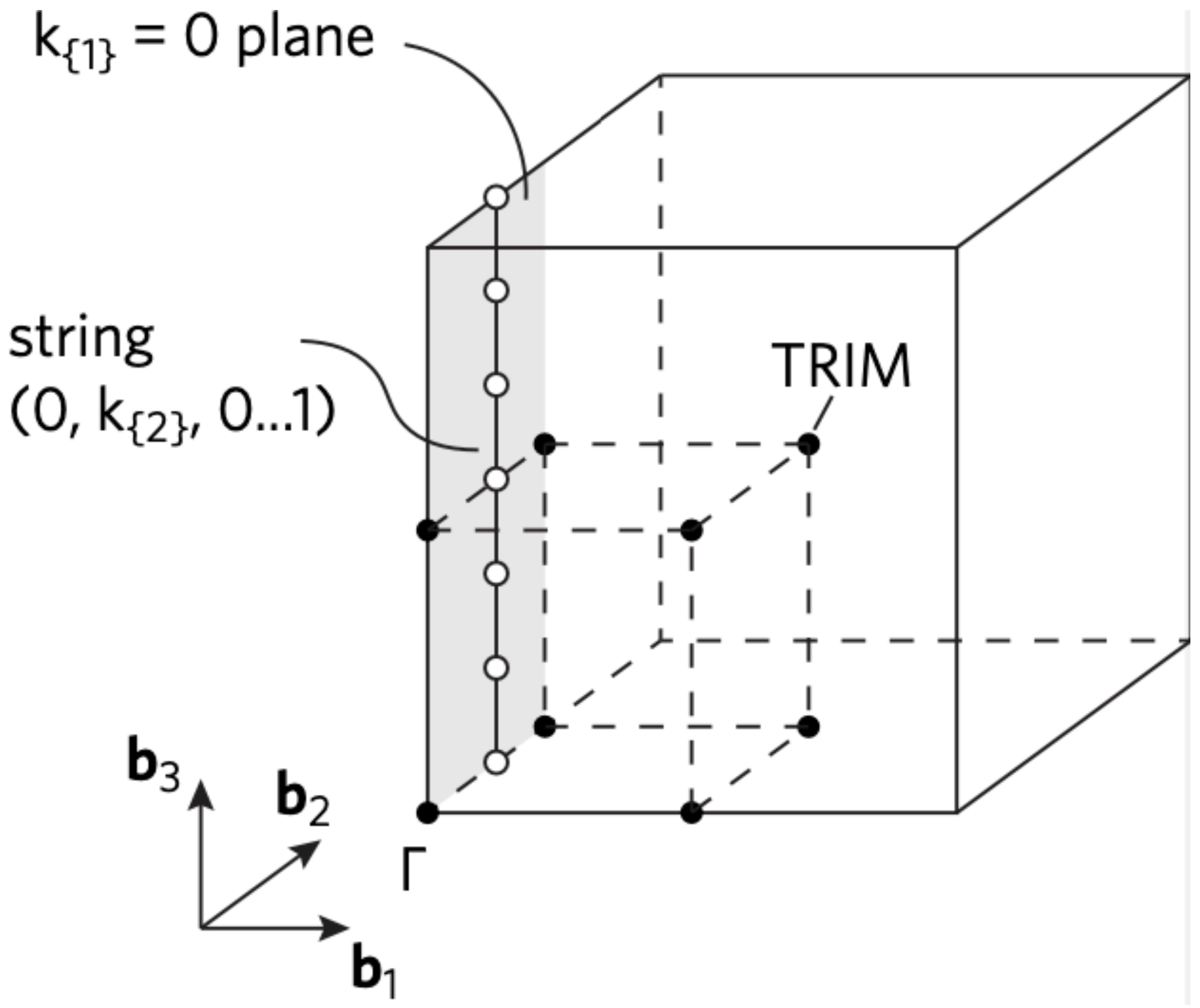
Modify the inputs for the wcc.py script (copy from $WIENROOT/SRC_BerryPI/BerryPI the wcc.py file)

```
kevoldir = 2      # Y, evolution direction (Wilson loops are constructed perpendicularly)
kevol = [0, 0.5]  # start and end in fraction of the corresponding reciprocal lattice
                  # vector G[kevoldir], in this case from Gamma to L (a TRIM point)
nkevol = 20       # discretization intervals (Number of lines constructed)
kwlsndir = 3      # Z, different from kevoldir (Wannierization direction)
nkwlsn = 10       # discretization intervals (Points along the line)
kfix = 0.0        # in fraction of reciprocal lattice vectors G[kfixdir]
bands = [61, 78]
parallel = True   # parallel option [-p]
spinpolar = False # [-sp]
orbital = False   # [-orb]
```

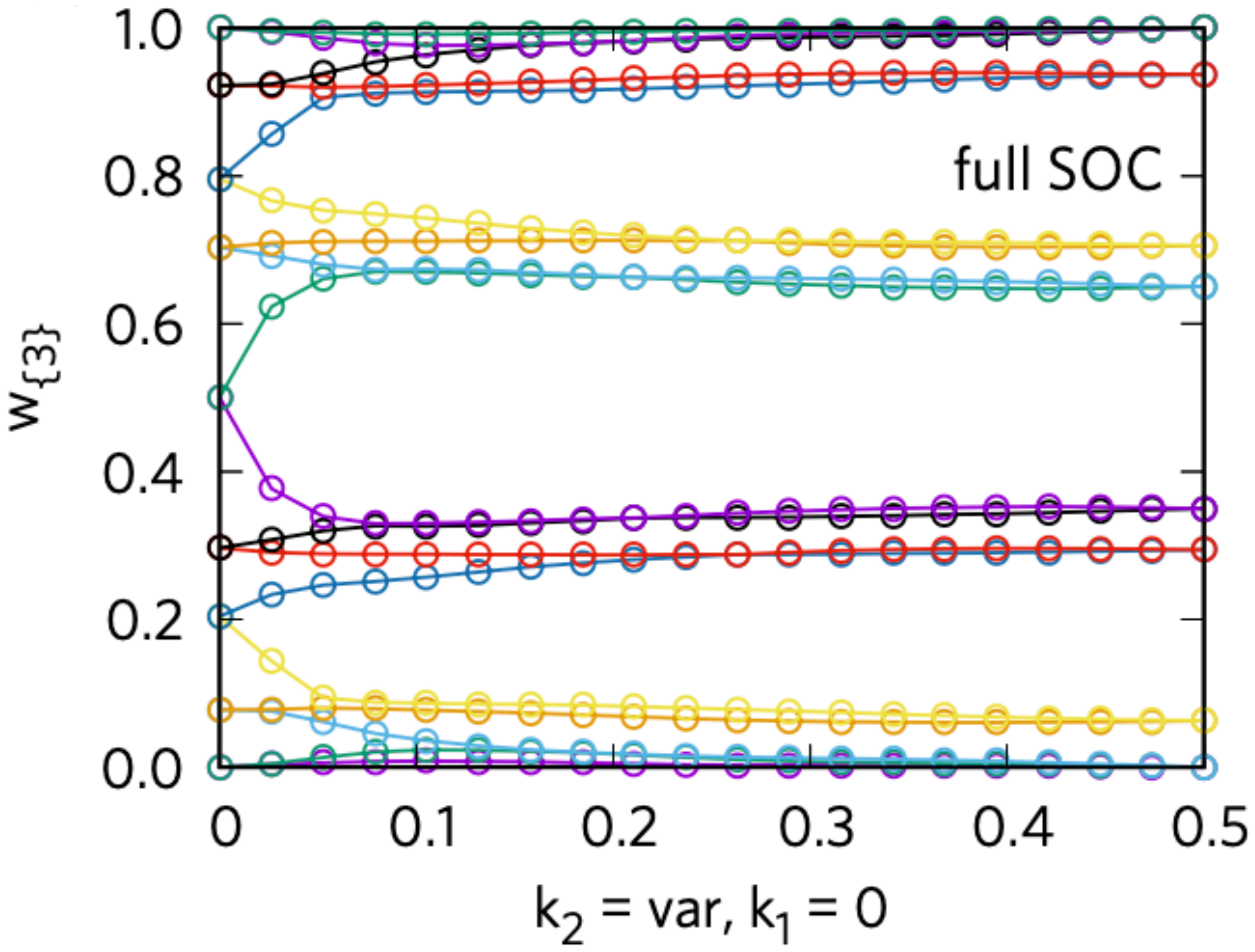
Run wcc.py

```
$        python3 wcc.py
```





The result will be stored in the wcc.csv file, which can be used to graph the HWCC evolution



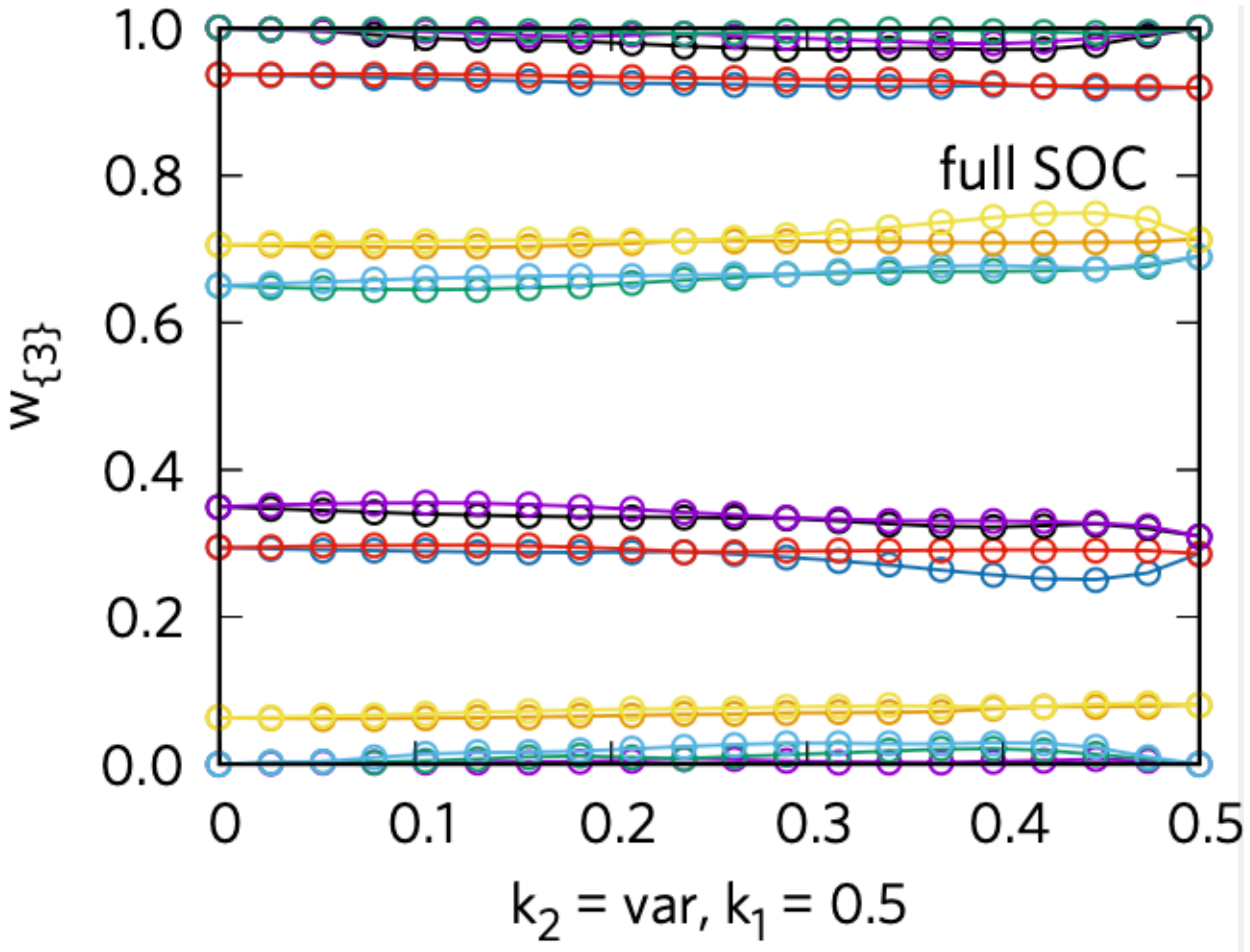
A gapless flow is evidenced, which points to a band inversion at Γ indicating a non trivial topological phase. The gapless flow of HWCCs in an insulator indicates winding of the HWCCs around the BZ, which is intimately connected with an adiabatic Thouless charge pumping in the bulk and leads to metallic edge states in a finite system. This bulk-boundary correspondence is captured by topological invariants. The full set of $\mathbb{Z}2$ topological indices can be inferred from the flow of HWCCs between various TRIM points.

Additional work: To obtain the Z_2 topological invariant

In 3D time-reversal-invariant insulators we need HWCCs for the following sets:

$$k_1 = 0, k_2 = 0 \ldots 0.5, w_{\{3\}} \quad \text{and} \quad k_1 = 0.5, k_2 = 0 \ldots 0.5, w_{\{3\}}, \quad A = (V_1, V'_1)$$
$$k_1 = 0 \ldots 0.5, k_2 = 0, w_{\{3\}} \quad \text{and} \quad k_1 = 0 \ldots 0.5, k_2 = 0.5, w_{\{3\}}, \quad B = (V_1, V'_2)$$
$$k_2 = 0, k_3 = 0 \ldots 0.5, w_{\{1\}} \quad \text{and} \quad k_2 = 0.5, k_3 = 0 \ldots 0.5, w_{\{1\}}. \quad C = (V_1, V'_3)$$

The result from the main tutorial corresponds to the first member of the set A. Computing the second member the following result is obtained



Indicating gaped flow, and a 0 index: A = (1,0)

Results for the remaining two sets B and C are identical due to symmetry of the rhombohedral lattice. The remaining weak indices are B = (1,0) and C = (1,0)

Since the flow of HWCCs is gapless in all directions originating at Γ, we should expect metallic states on *any* surface plane, hence the strong topological insulator. Therefore, the complete topological invariant of Bi_2Se_3 is Z_2 = (1;000)

Reference:

- A.F. Gomez-Bastidas, O. Rubel, Software implementation for calculating Chern and Z2 topological invariants of Crystalline solids with WIEN2k all-electron density functional package, Computer Physics Communications, 292, 2023,108864, https://doi.org/10.1016/j.cpc.2023.108864.

Pre-requisites

This tutorial was verified using:

- WIEN2k 23.3.2
- python 3.10.2
- xcrysden 1.5.60

Python dependencies:

- Numpy
- matplotlib

Resources:
- All in 4 cores not fully employed
- 6GB of maximum memory usage

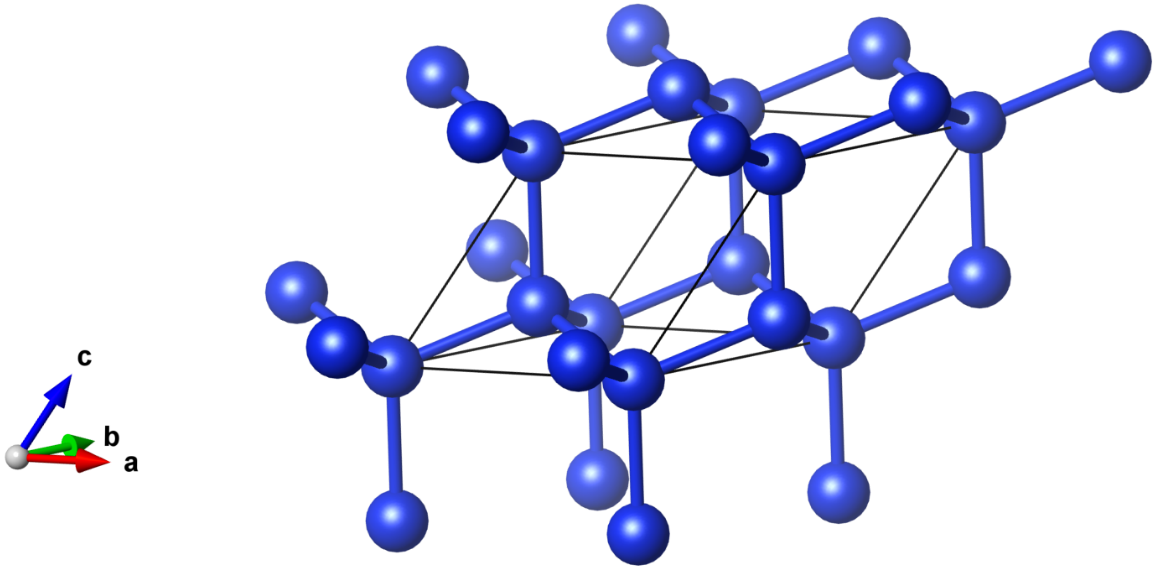
Effective band structure of $Si_{1-x}Ge_x$ alloy

+ fold2Bloch

https://github.com/rubel75/fold2Bloch-Wien2k

Instructions

w2web Construct a primitive lattice of Si



2-atoms per unit cell
Primitive lattice "P", $\alpha = \beta = \gamma = 60^\circ$
Cell size (Ang.): a = b = c = 3.870393 Å
Coordinates:
Si (0 0 0)
Si (1/4 1/4 1/4)

RMTs = 2 Bohrs

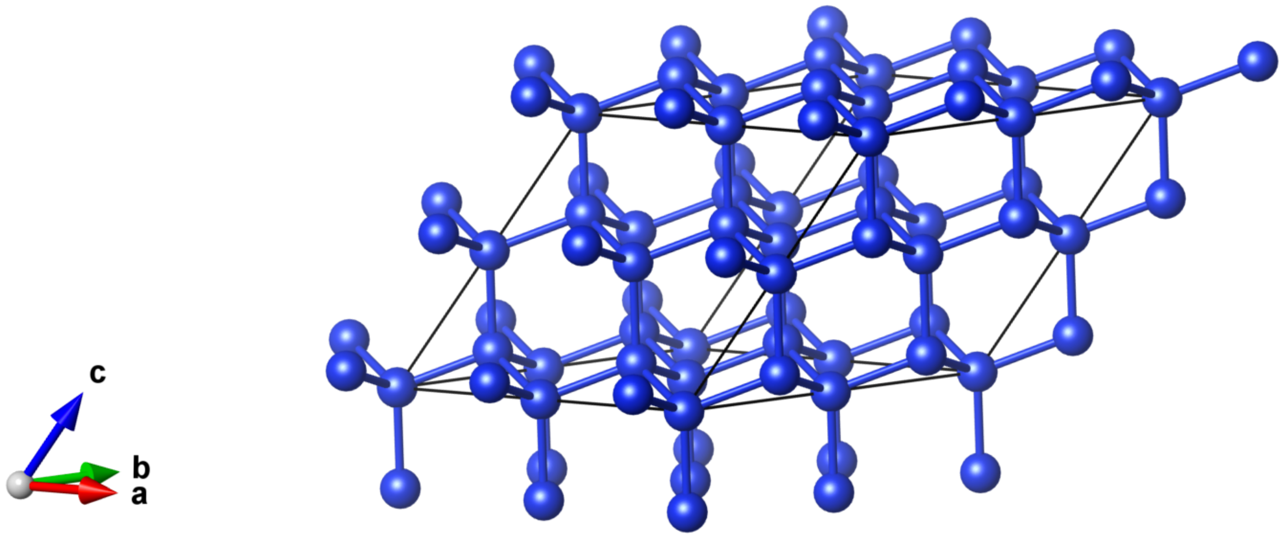
Do **not** initialize calculation.

Create a 2x2x2 supercell: no shifts, P-type lattice, no additional vacuum

```
$   x supercell
```

Rename "[case]_super.struct"
to "[case].struct"

The new supercell should have
16 Si-atoms (2 atoms x 2 x 2 x 2)



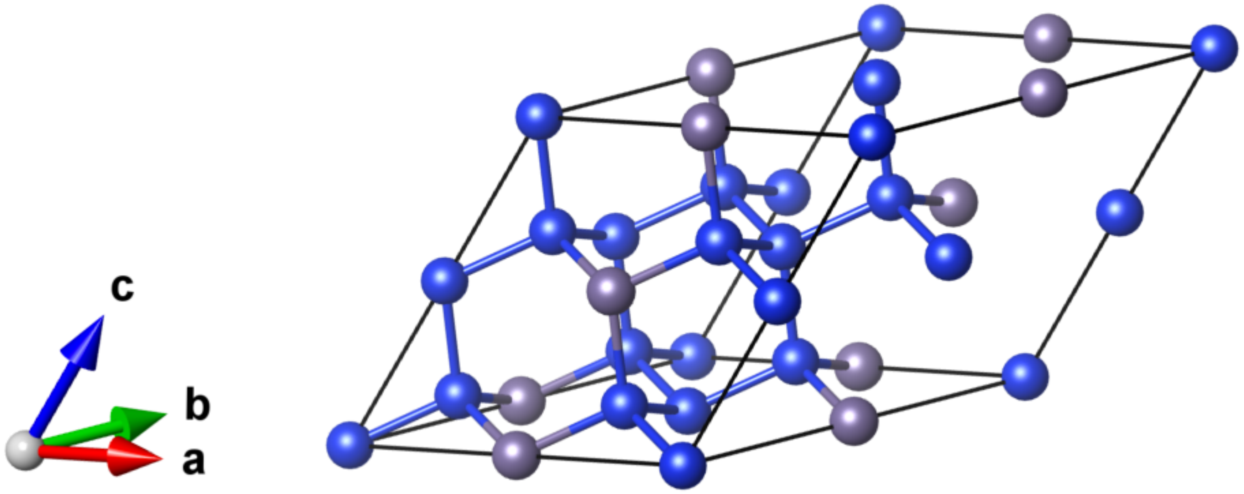
`w2web` Introduce Ge into the structure by switching 3 Si → 3 Ge atoms
The Ge composition will be 3/16 = 0.1875 (*ca.* 19%)

One possibility is to pick any 3 Si-atoms of your choice. But this choice will be "biased" and does not qualify as a random alloy.

Alternatively, we can generate a randomized sequence of integers between 1 and 16 (total number of Si-atoms) using
https://www.random.org/sequences
"**3** **2** **6** 9 13 4 7 16 12 1 5 10 15 14 11 8"
The first 3 numbers are labels of atoms to be switched. Of course every time you will get a different sequence. Thus the arrangement of atoms is not unique.

More elegant solution involves the use of "Special Quasirandom Structures", but it goes beyond this tutorial.

When substituting Si → Ge using StructGen™, make sure to clear up the Z value and label atoms as
Ge 1, Ge 2, Ge 3, Si 1, Si 2, …



Initialize SCF calculation: GGA-PBE, RKmax=6, 3x3x3 k-mesh (shifted)
(note: crude parameters are used for tutorial purposes)

```
$   init_lapw -b … or web interface
```

Run SCF calculation (~10 iterations): iterative diagonalization, energy convergence 0.0001, charge convergence 0.001. It can be faster if you parallelize over k-points and/or take advantage of OMP-parallelization (~10 mins on 4 cores)

```
$   export OMP_NUM_THREADS=4
$   run_lapw -it -ec …  (without -p)
```

or

Prepare `.machines` file that contains these two lines (replace XX with your specific host number)

```
1:localhost
1:localhost
```

```
$   run_lapw -it -p -ec …
```

Save the calculation in "SCF" directory

```
$   save_lapw … or web interface
```

Copy L-Γ-X k-path template (`LGX.klist_band` not a part of the standard Wien2k distribution). Look inside the file. It is not a standard k path used for band structure plots. You will see the path going from L to Gamma to -L and then X to -X. Usually we would consider this path as redundant for a regular band structure plot. But for unfolding to work, we need the path so span the entire width of the Brillouin zone.

```
$   cp $WIENROOT/LGX.klist_band [case].klist_band
```

Recalculate eigenvalues and wave functions for k-points on the path selected. Do not use k-point parallelization here as it will change the workflow of fold2Bloch analysis because multiple [case].vector files will be generated. (Please talk to an instructor if you would like to explore [-p] this option.)

```
$   x lapw1 -band
```

Make sure the vector files are located in the same directory (sensitive to settings of the $SCRATCH variable)

```
$   ls -l [case].vector
```

Perform unfolding of the vector file using the transformation matrix [2 0 0; 0 2 0; 0 0 2], which reflects the size of the supercell

```
$   fold2Bloch [case].vector "'2 0 0:0 2 0:0 0 2'"
```

The newly created file [case].f2b contains 5 columns:
Unfolded KX, KY, KZ, Eigenvalue (Ry), Bloch spectral weight

The Fermi energy can typically be found in [case].scf file (:FER label). However, it can be inaccurate, if the k-point mesh for SCF calculation did not include special points (Γ point in this case).

Copy a Matlab plotting script to the Wien2k work directory
(`ubs_dots.m` is not a part of the standard Wien2k distribution)

```
$   cp $WIENROOT/ubs_dots.m .
```

Edit the plotting script:

KPATH = [1/2 0 0; 0 0 0; 1/2 1/2 0] corresponds to L-Γ-X;

Dp2s = [2 0 0; 0 2 0; 0 0 2] is the transformation matrix between the primitive cell and the supercell;

KLABEL = {'L'; 'G'; ‘X'} are labels for k-points;

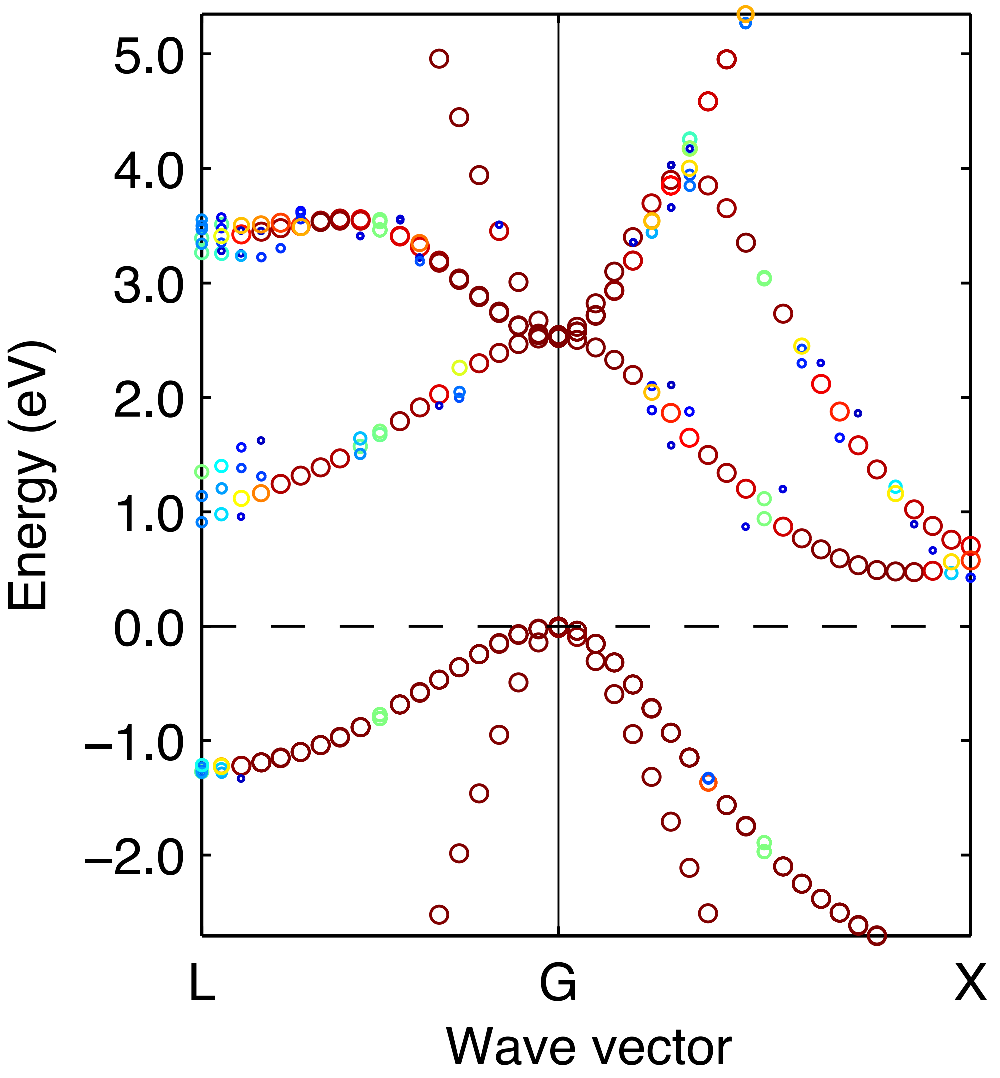
finpt = ‘[case].f2b’ is the input file name;

Ef = 0.400207 is the Fermi energy (Ry);

G = [0.083726 -0.027909 -0.027909;
0.000000 0.078938 -0.039469;
0.000000 0.000000 0.068362]
reciprocal lattice vectors from [case].outputkgen;

Lunch Matab in GUI mode

```
$  matlab
matlab >> ubs_dots
```

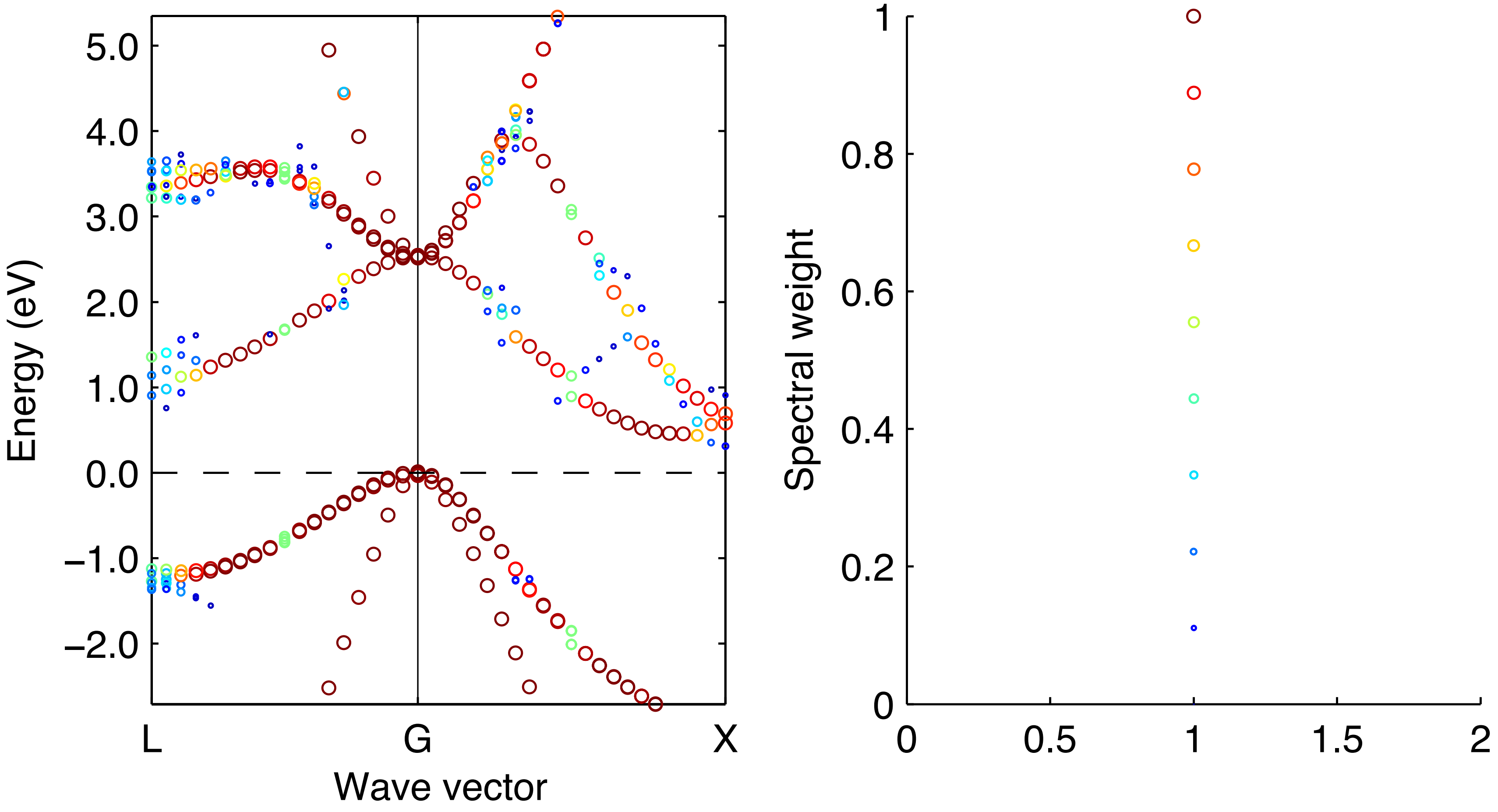


Run plotting script in Matlab and view the effective band structure of the alloy. Note a well-preserved Bloch character at the top of the valence band and bottom of the conduction band in spite of the fact that the structure does not have a zinc-blende symmetry any longer.

When you have time (requires1-2 hrs depending on parallelization and load of machines)…

It should be noted that so far we have ignored static atomic displacements that are present in alloys due to the size mismatch between the additive element (Ge) and the host (Si). Forces in our structure are on the order of 15 mRy/Bohr that requires optimization of atomic positions.

The suggested procedure would be to build a supercell, place the alloying elements, and perform optimization of atomic positions before undertaking infolding. The effective band structure with static atomic displacements included in the calculation is shown below. Can you spot any changes at the band edges?

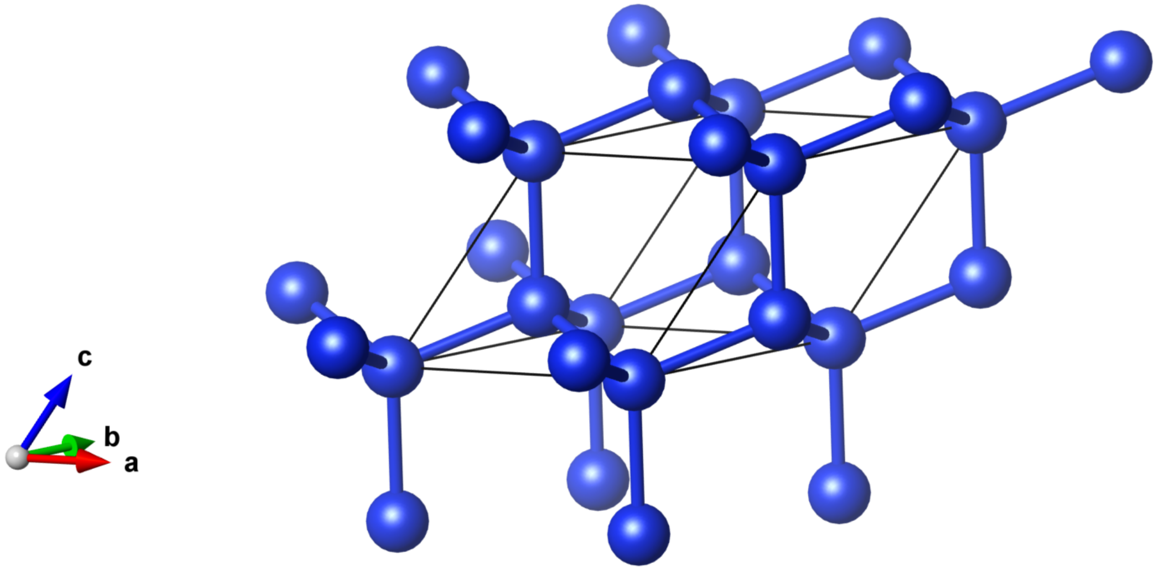


Effective mass in Si from perturbation theory

\+ mstar

https://github.com/rubel75/mstar

Instructions



w2web Construct a primitive lattice of Si

2-atoms per unit cell
Primitive lattice "P", α = β = γ = 60°
Cell size (Ang.): a = b = c = 3.870393 Å
Coordinates:
Si (0 0 0)
Si (1/4 1/4 1/4)

RMTs = 2 Bohrs

Initialize SCF calculation with PBE, RKMAX=7, <u>500 k points</u>:

```
$ init_lapw -b -vxc XX -rkmax YY -numk ZZ
```

It is worth to verify the structure with XCrysDen

Perform regular SCF calculation without SOC (set the energy convergence to 0.0001 Ry and the charge convergence to 0.001e):

```
$ run_lapw -ec XX -cc YY
```

Initialize SOC with default settings (non-magnetic) and follow prompts:

```
$ init_so_lapw
```

Perform SCF calculation with SOC (use the same convergence parameters as without SOC)

```
$ run_lapw -ec XX -cc YY -so
```

Edit the LAPW1 input file `[case].in1` to increase Emax to 10 Ry.

Edit the LAPWSO input file `[case].inso` file to increase Emax to 7 Ry to get more empty states (recall the perturbation theory equation with the sum over all states including the infinite number of empty ones).

Recalculated DFT orbitals (vector files).

```
$ x lapw1
$ x lapwso
```

Fake spin-polarized calculation (needed for optics with SOC).

```
$ ln -s [case].vsp [case].vspup
$ ln -s [case].vsp [case].vspdn
$ ln -s [case].vectorso [case].vectorsoup
```

Get the input file for optics (template):

```
$ cp $WIENROOT/SRC_templates/case.inop [case].inop
```

Edit `[case].inop` to enable writing of momentum matrix elements in `[case].mommat2*` files.

Edit `[case].inop` to extend Emax to 7 Ry.

Execute optics with SOC in the spin-polarized mode:

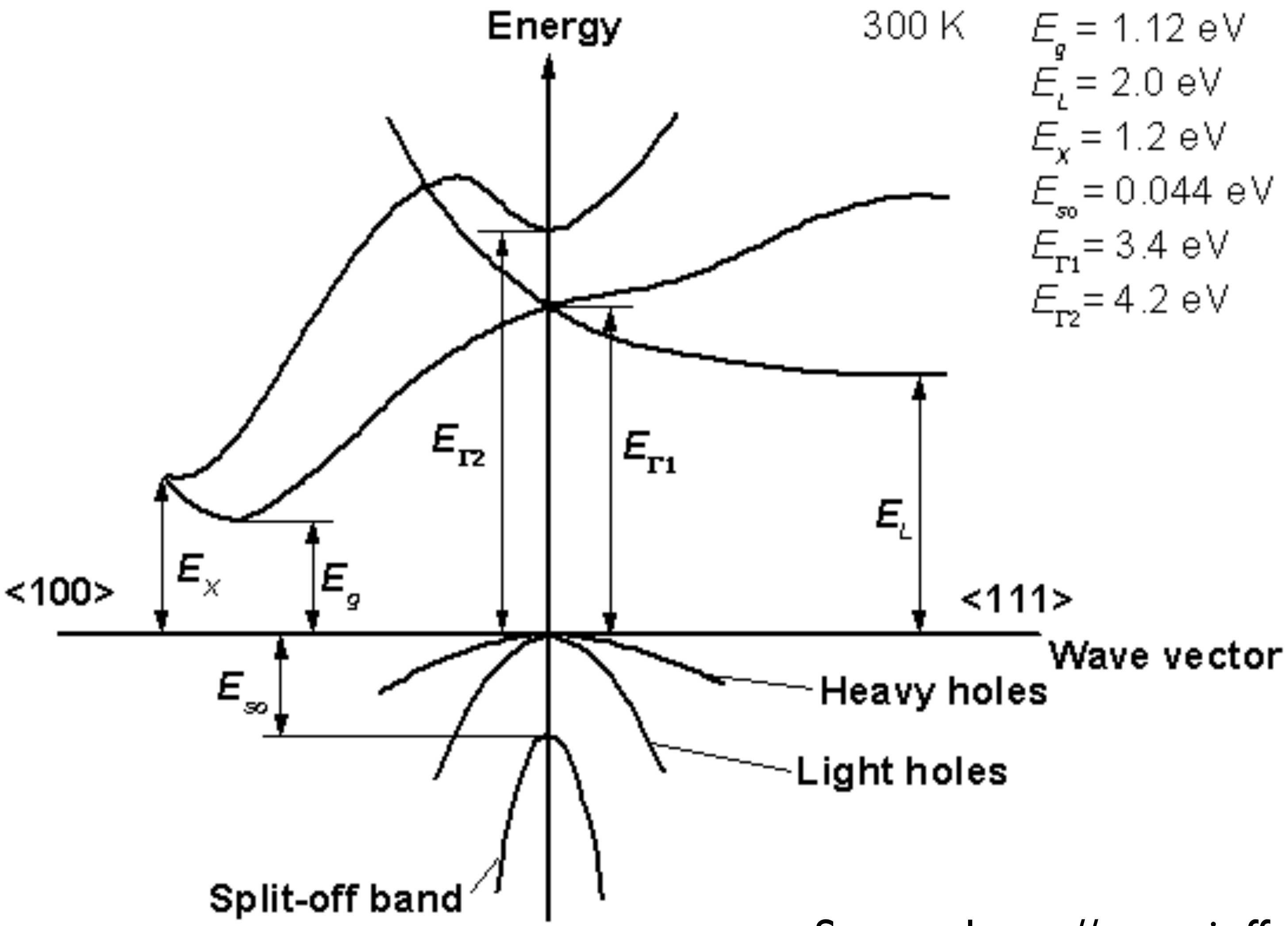
```
$ x optic -so -up
```

Execute mstar pointing to the file with momentum matrix elements and an eigenvalues degeneracy acceptance criterion of $\Delta E \leq 10^{-5}$ Ha:

```
$ /path/to/mstar [case].mommat2up 1e-5
```

Inspect the band occupancy (:BAN in `[case].scf`) and energy eigenvalues (`[case].energyso`) to identify the k-point index for the top of the valence band (it should be at Γ point) and the band indices for the split-off, light-hole, and heavy-hole bands. The band structure is shown on the next page.

Identify the k-point index and the band indices for the conduction band edge (it should be along $\langle k_x, 0, 0 \rangle$ direction).

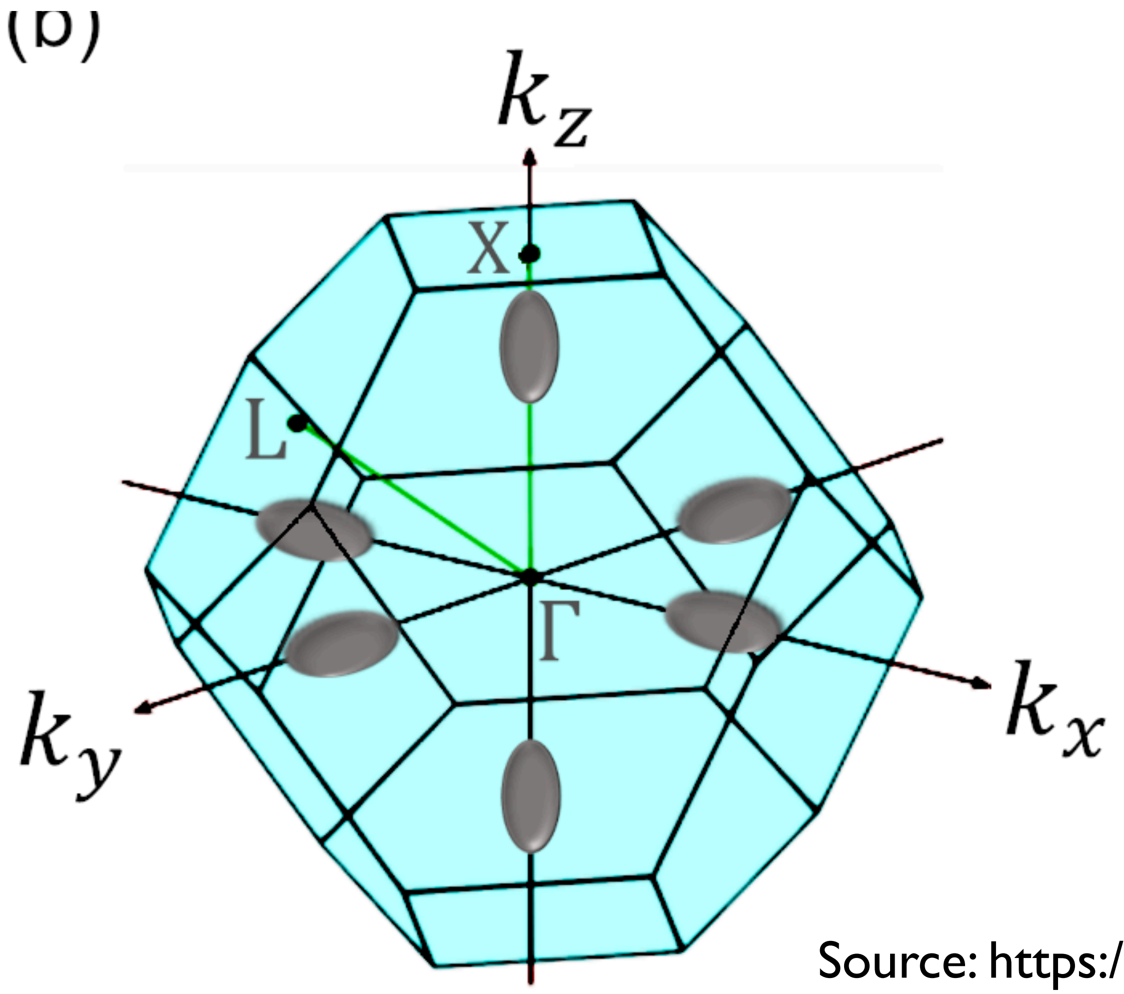


Source: https://www.ioffe.ru/SVA/NSM/Semicond/Si/bandstr.html

Read the inverse effective mass tensor m_0/m^*_{ij} from the file `minv_ij-up.dat` for the k-point index and the band indices of the split-off, light-hole, and heavy-hole bands. The masses of holes along Cartesian $\langle 100 \rangle$ direction should be $m^*_{11,\text{so}} = -0.23$, $m^*_{11,\text{lh}} = -0.19$, and $m^*_{11,\text{hh}} = -0.27$ in units of the electron mass at rest m_0.

Why values of m^*_{11} are negative? What is the meaning of off-diagonal components of the effective mass tensor (m^*_{ij} at $i \neq j$)? Is the mass of holes isotropic? What are the principal components of the effective mass tensor for holes (see file `minv_pr-up.dat`)?

Perform the same analysis for the conduction band edge. You should be able to find longitudinal mass of $0.93m_0$ and the transverse mass of $0.19m_0$. Try to rationalize the result using ellipsoids of equipotential surfaces of the conduction band shown below.



Source: https://docs.nanoacademic.com/qtcad/theory/multivalley_EMT/#id5