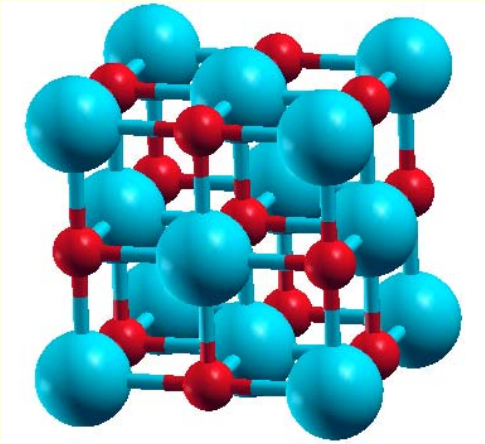
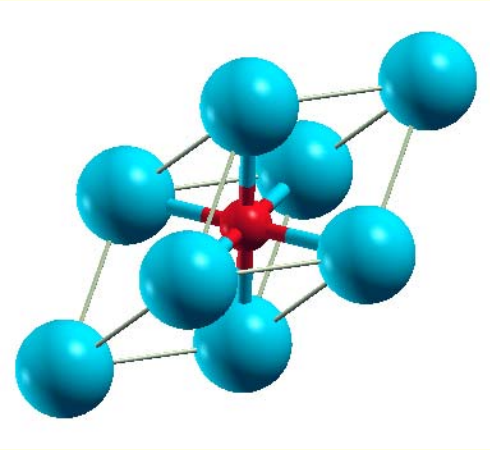
# Exercises:

- In the following you find some suggestions for exercises, which teach you various tasks one may perform with WIEN2k.
- New WIEN2k users should start with the first basic exercises (1-5), covering: structure generation, initialization, scf-cycle, bandstructure, DOS, electron density, structure optimization, supercell generation, spin polarization
- Later on, choose examples of your interest as there are probably more exercises than you can do here.
- 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 !!

# Exercise 1: Getting started:

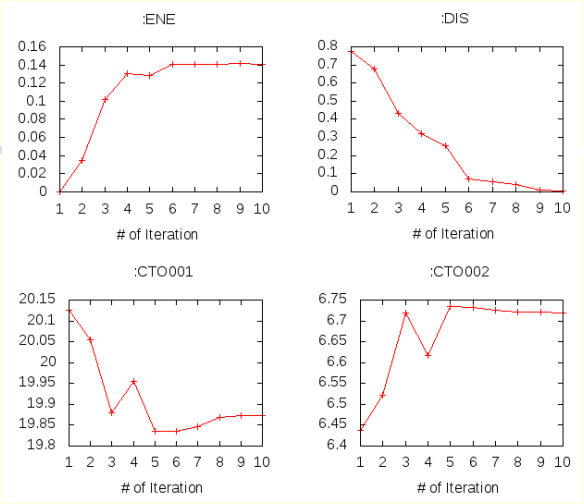
- i) Open a terminal window (skip points i-iii if done before)
- ii) Start w2web
- iii) Connect with firefox to w2web as indicated on the screen of ii)
- iv) Try the "quick-start" example for **TiN** (similar to TiC in the UG)
  - *create new session named "TiN", "create" and "select" the suggested directory.*
  - *Generate structure (a=4.235 Ang; reduce RMT by 1%)*
  - *view structure with Xcrysden (switch primitive / conventional cell)*





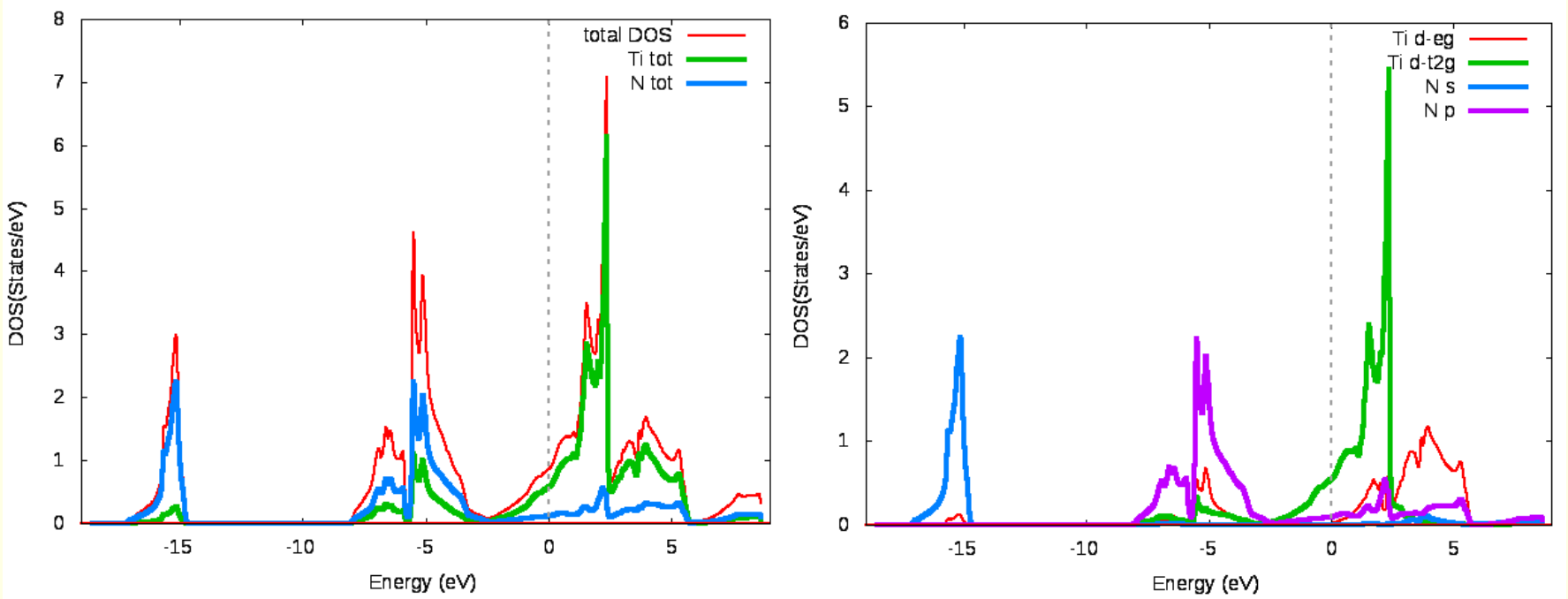
  - *initialize (init_lapw -b); use defaults*
  - *scf-cycle (run_lapw); use defaults; monitor "STDOUT" and "dayfile"*
    - How many iterations did you need ? How long took a single scf-iteration ?

# TiN continued




- *utilities: analyse*
  - (:ENE, :DIS, :CTO) graphically

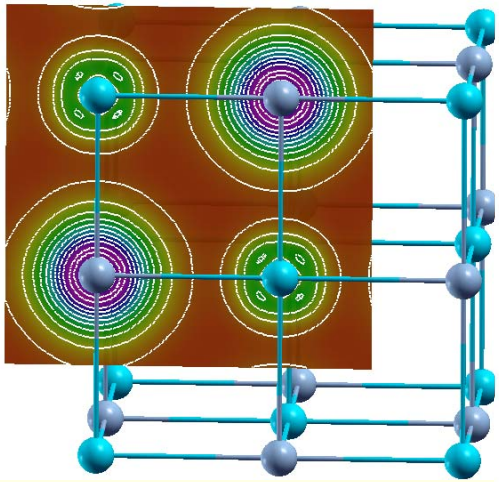
- *utilities: save_lapw (use as save-name: "TiN_exp_pbe_rk7_1000k")*
- *DOS (plot 7 cases: total + Ti-tot + N-tot and Ti-eg + Ti-t2g + N-s + N-p)*



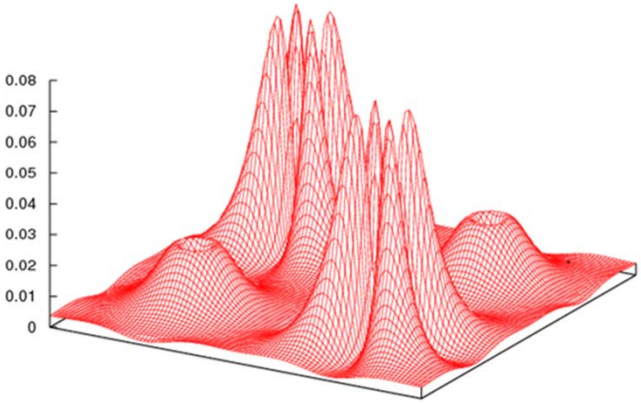
# TiN continued ...

- *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); 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

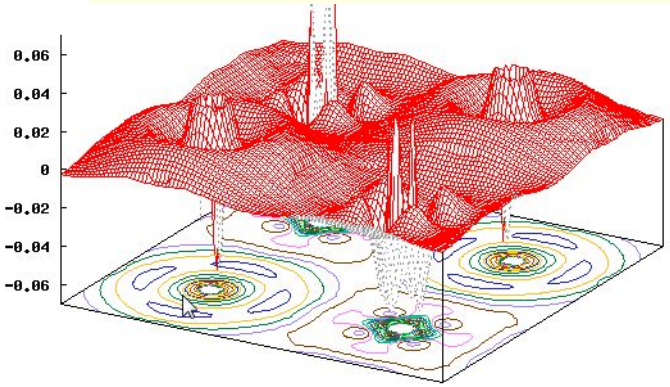
valence ρ



Ti-d band

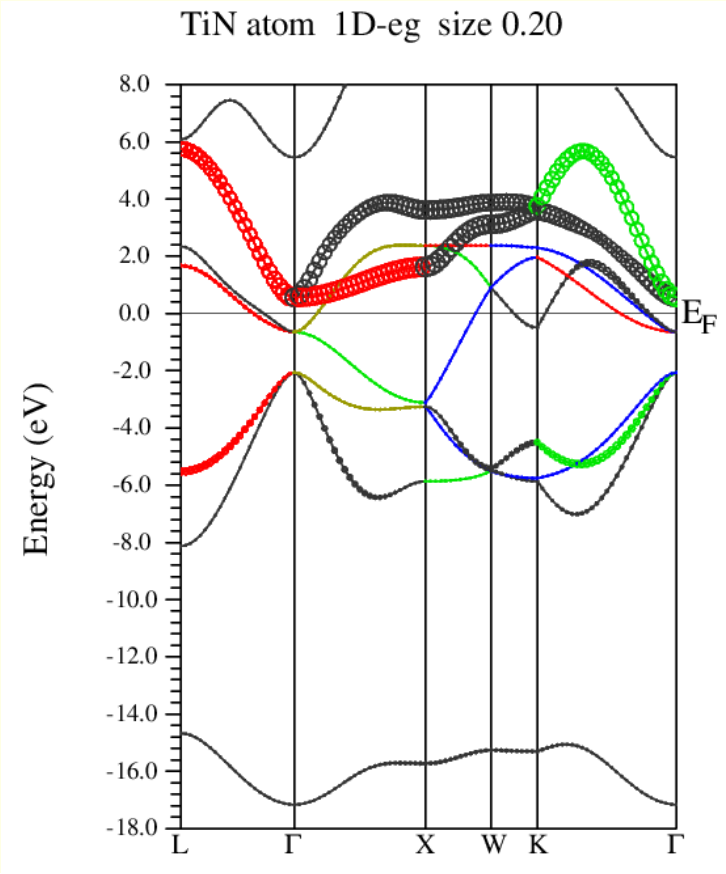


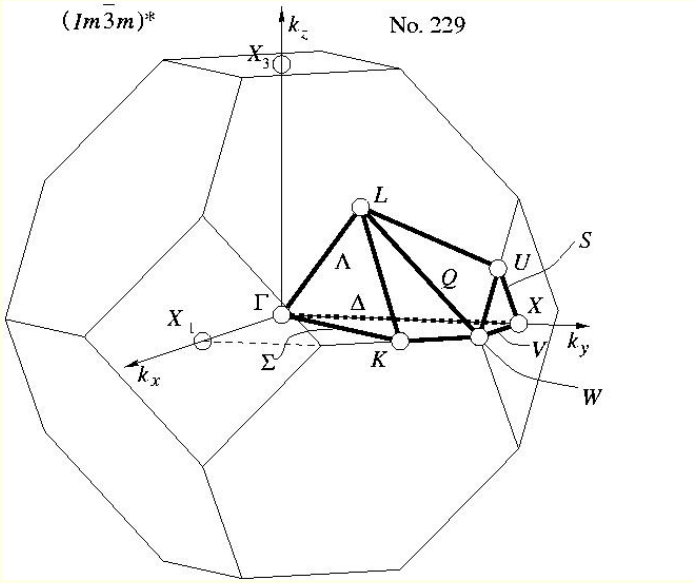

difference density



# TiN continued

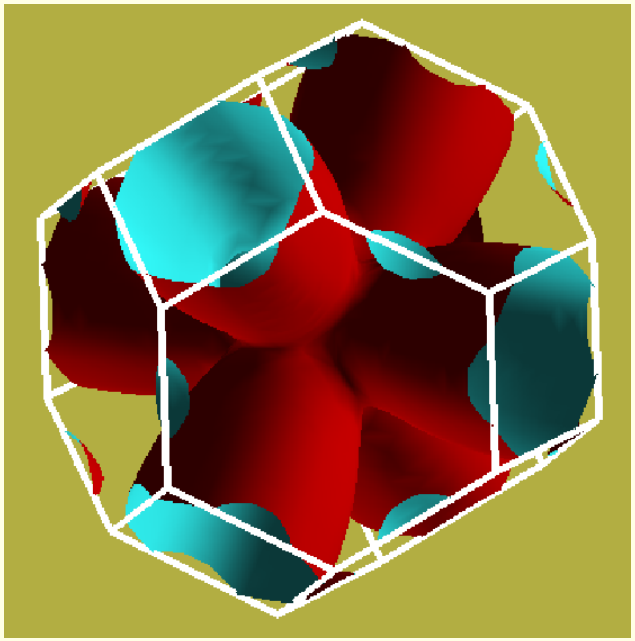
- bandstructure (along L-Gamma-X-W-K-Gamma with “character plotting”)
  - *use xcrysden (save as „xcrysden.klist“; select „from xcrysden“ in next step and click generate k-mesh )*
  - *identify “t2g-” and “eg-” bands (fat band plots)*



# 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
## Testing accuracy: RKmax and k-points

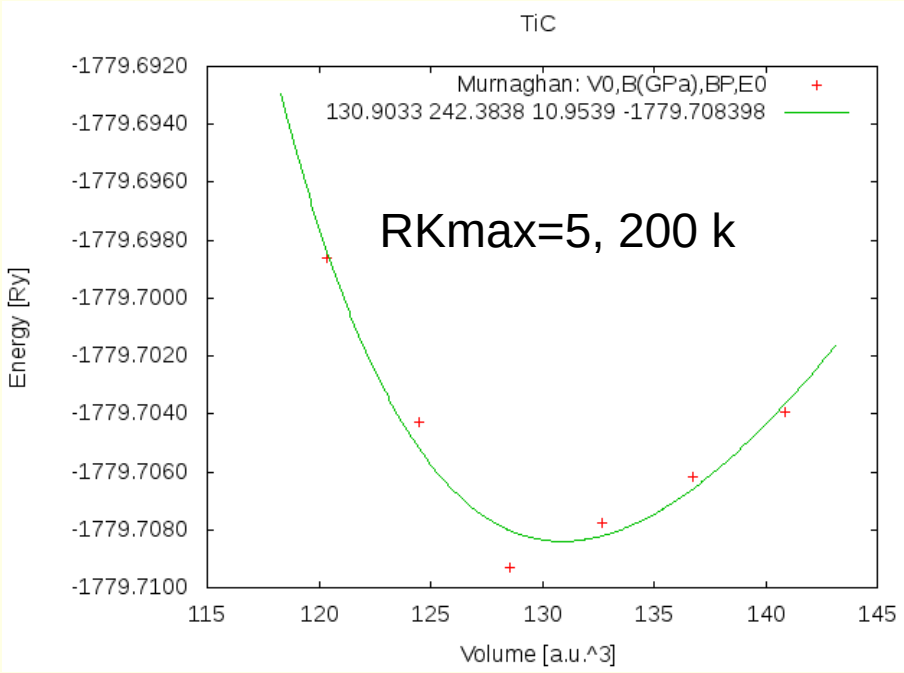
- TiC (fcc, **a=4.328** Ang, **setrmt 4%**)
- a) initialize in expert mode with **LDA, RKmax=5, 200 k-points** (bad values, on purpose !!)
- b) run x optimize and generate 6 structures (-12, -9, -6, -3, 0, 3% volume change)
  - *(because of LDA we expect 1-2% smaller lattice parameter (3-8% in volume) than experiment)*
- c) edit "optimize.job". Modify the "run_lapw" and "save_lapw" commands to:
  - *run_lapw –cc 0.001 –ec 0.00001*
  - *save_lapw ${i}_default_rkm5_200k*
- d) run optimize.job, plot the results (using *rkm5_200k)
- e) set **RKMAX=6.5** in TiC.in1 and x kgen with **1000k**
- f) edit "optimize.job". **Uncomment** the "**cp line**" and "**comment clmextrapol**" modify:
  - *cp ${i}_default_rkm5_200k.clmsum TiC.clmsum # Using previously converged densities saves a lot of CPU time!!*
  - *# clmextrapol ...*
  - *save_lapw ${i}_default_rkm6.5_1000k*
- g) repeat step d) (plot the results for "*_rkm6.5_1000k")

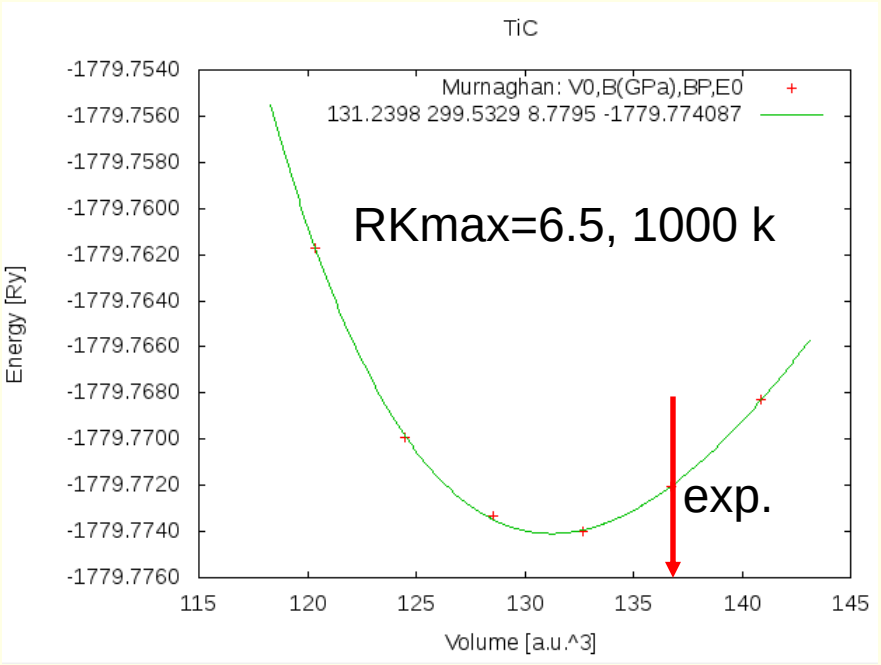
- *Find out how RKmax and k-points lead to smooth/non-smooth curves. Estimate good values and compare in particular B and BP (Bulkmodulus and its volume derivative). Fully converged results would require RKmax=8 - 9 , 10000 k and 10 volumes with $\Delta V$=1%.*
- *You may also do this with another XC-potential (eg. PBEsol) and will see a very large effect ...*

- Remember: Depending on the specific property you want to calculate (just a DOS, or Energy-Volume curves, or EFG, or structure optimization with forces,..) and the desired accuracy, the types of atoms, insulator/metal and system size you may need different RKmax and k-point samplings:
  - *H: RKmax > 2.5; sp-elements: RKmax > 5; d-elements: RKmax > 6; f-elements: RKmax > 7; (see our faq-page)*
  - *1 atom/cell, metal: 1000-10000 k-points or more*
  - *1 atom/cell, insulator: 100-1000 k-points or more*
  - *For N atoms/cell you can reduce the k-mesh by a factor N*

- *Remember: Always test your **specific property** for convergence !!*

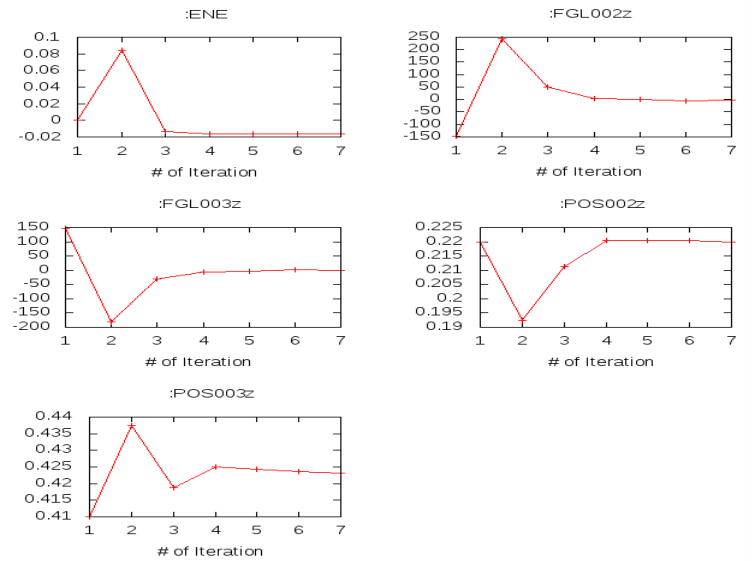
# Volume optimization for TiC

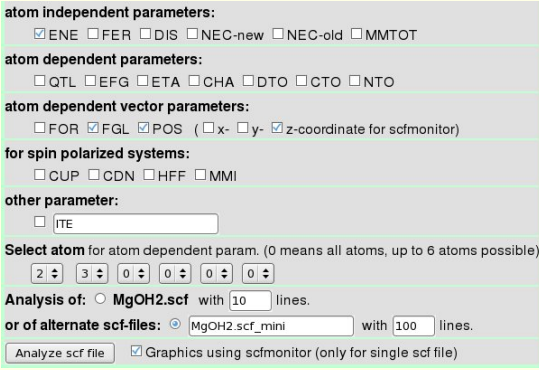


# Exercise 3: optimization of positions in $Mg(OH)_2$

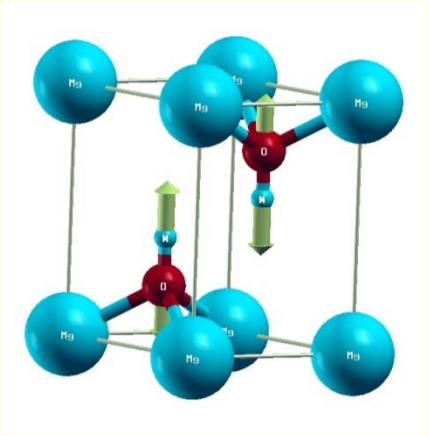
- **create two "cases" (directories) for PORT and MSR1a optimization**
  - *initialize both cases (or copy after init one case to the other and use „rename_files")*
  - *P-3m1 (164), a=b=3.15 c=4.77 Å γ=120°; Mg(0,0,0) O(1/3,2/3,**0.22**) H(1/3,2/3,**0.41**); RMT: reduce by 7%*
  - *init_lapw -b –numk 100 –rkmax 3*
- **minimization using PORT:**
  - *min_lapw (or „mini-positions in w2web)*
  - *save_lapw case_relaxed_rkm3*
  - *analyze **case.scf_mini***
    - :ENE :FGL002z :POS002z :FGL003z :POS003z
  - *Find out how many scf cycles you needed*
    - grepline :ITE '*scf' 1 (in terminal)
- **check RKMAX convergence:**
  - *increase RKMAX to 3.5 (case.in1)*
  - *run –fc 1 (and check your forces)*

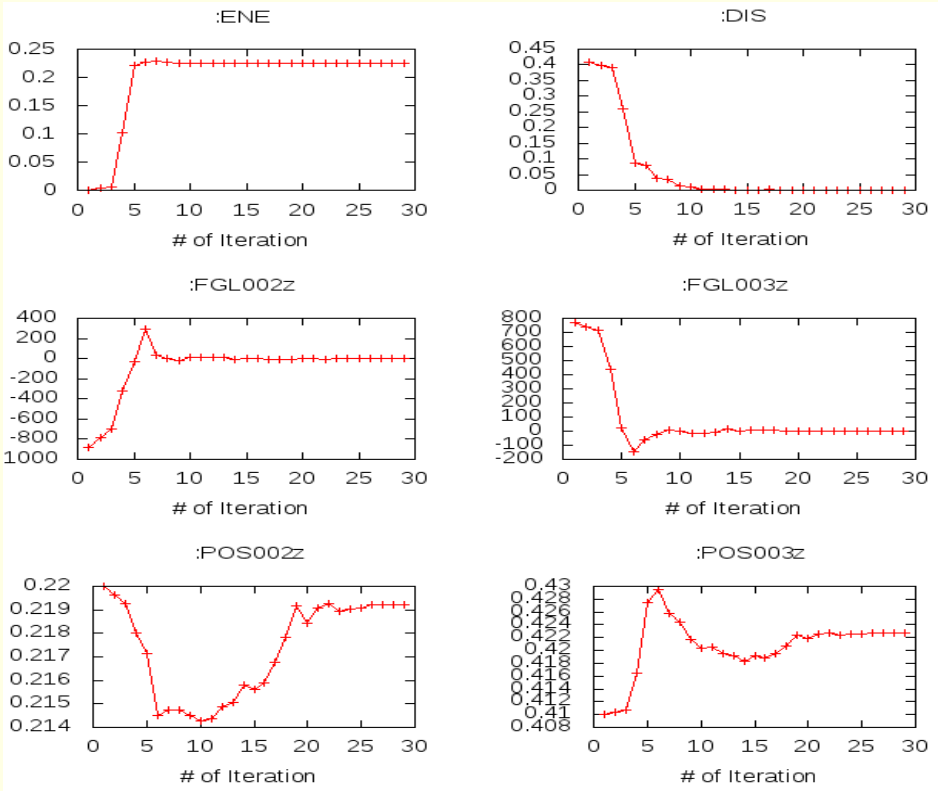


# $Mg(OH)_2$ continue

- minimization using MSR1a:
  - ***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 20)
  - *analyze* ***case.scf*** and find out how many scf cycles you needed
    - :ENE :FGL002z :POS002z :FGL003z :POS003z :ITE
  - *save_lapw case_final*
  - *use the „arrows“ utility to display initial forces and final relaxations (see UG p.195)*





# Exercise 4: Creation of supercells

- These exercises should be done WITHOUT w2web in a terminal window !
- **creation of basic structure: MgO**
- mkdir super; cd super;
- makestruct (and type in the following information). It creates **init.struct**
  - *MgO: lattice type: F, a= 7.96 bohr*
  - *Mg (0,0,0), O (0.5,0.5, 0.5)*
- cp init.struct super.struct
- view the structure using: xcrysden --wien_struct init.struct

- **16-atom supercell**
- x supercell (use **super.struct**, select **2x2x2** and **F-cell):**
- cp super_super.struct super.struct
- edit super.struct and mark first Mg atom as "**Mg1**"
- x nn and if :WARNINGs appear do the next line:
  - *cp super.struct_nn super.struct; and repeat the "x nn" step above*
- x sgroup and view super.outputsgroup (no errors, but gives you a spacegroup)
  - *how many non-equivalent atoms do you have now ? view the structure with xcrysden. Now you would be ready to run* ***init_lapw -b ...., but we just save it using cp super.struct super_16.struct***

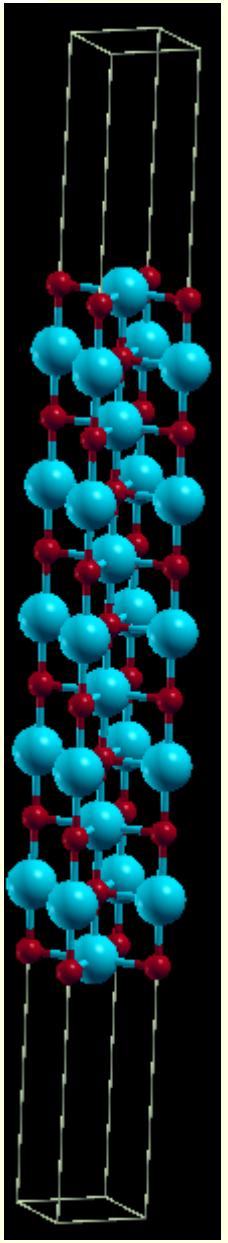
# Exercise 4: Creation of supercells (cont.)

- **32, 64 and 128-atom supercells** (as above, but with B, P cell or 4x4x4-F)
- cp init.struct super.struct
- x supercell (use **super.struct**, ...**):**
- cp super_super.struct super.struct
- edit super.struct and mark first Mg atom as "**Mg1**"
- x nn and if :WARNINGs appear do the next line:
  - *cp super.struct_nn super.struct; and repeat the "x nn" step above*
- x sgroup and view super.outputsgroup (no errors, but gives you a spacegroup)
  - *how many non-equivalent atoms do you have now ? view the structure with xcrysden. Now you would be ready to run i**nit_lapw -b ...., (see eg. lecture on XANES spectroscopy)***
  - *save the structures using **cp super.struct super_32.struct***

- Instead of labelling "Mg1", one could also **remove** an atom (vacancy) or **replace** an atom by another (impurity).
- Replacing atoms is better done in w2web, because this will also update radial meshes. (change **name** of atom AND **remove Z** !!)

# Exercise 4: Creation of surface slabs, relax it

- **(001) surface with 11 layers:**
- mkdir 001, cp init.struct 001/001.struct; cd 001
- x supercell (use **001.struct**, 1x1x5, 30 bohr vacuum in z; repeat atom at top (y))**:**
- cp 001_super.struct 001.struct
- xcrysden --wien_struct 001_super.struct & (leave it open for comparison)
- x sgroup and view 001.outputsgroup (it created a new structure for you)
- cp 001.struct_sgroup 001.struct
- xcrysden --wien_struct 001.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 ?*
  - *save the structure using* ***cp 001.struct start_surface-001.struct***
- init_lapw –b –numk 10 –fermit 0.002 # 2D-BZ !
- run_lapw –fc 10 # observe the forces in scf-file, what relaxation do you expect ?
- save_lapw unrelaxed
- run_lapw –min –fc 1 # minimizes forces by optimizing positions
- while running, edit 001.inM and increase tolf to 5; save_lapw relaxed
  - *How much have the surface and sub-surface atoms relaxed ?*

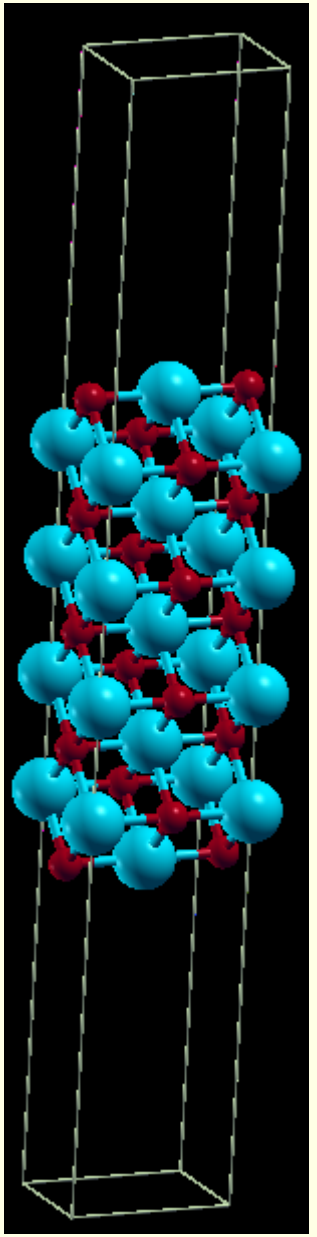


# add-atoms, bigger cells, ...

- If you now want to study **adsorption** of an atom you could simply add **2 equivalent** atoms manually (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 Fe atoms ?*
  - *at what distance ?*
  - *check it out using xcrysden*
- This structure could then serve as base for a bigger supercell (for instance 2x2x1) to simulate reduced “coverage”.

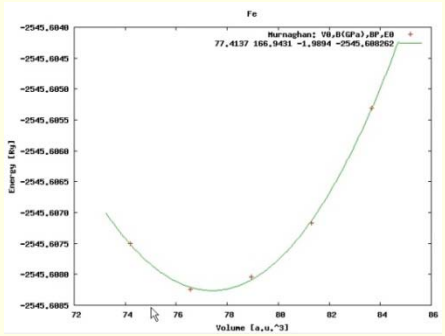
# Exercise 4: Creation of supercells (cont.)

- **(110) surface with 9 layers: (**using the **structeditor)**
- octave (use repeat-key arrow-up !)
  - *helpstruct* *# list all possible commands*
  - *a=loadstruct("init.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 and view super.outputsgroup
- cp super.struct_sgroup super.struct
- xcrysden --wien_struct super.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 ?*
  - *save the structure using* ***cp super.struct super_surface-110.struct***



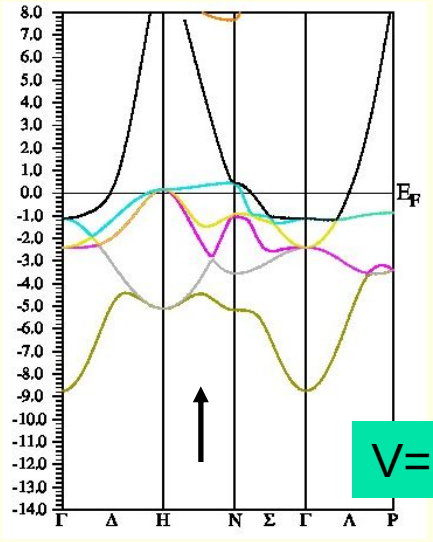
# Exercise 5: spin-polarized calculations

- Magnetism: bcc Fe ($a_0$=2.86 Å)
  - *setrmt: 3%; 5000k; spin-polarization:yes, use RKmax=7, then 8*
  - *do a volume optimization (-6, -3, 0, 3, 6 %)* *(activate runsp_lapw instead of run_lapw !)*
    - check equilibrium volume, :MMT as function of volume

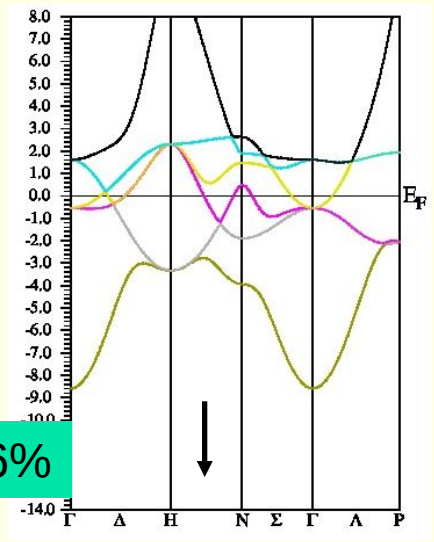


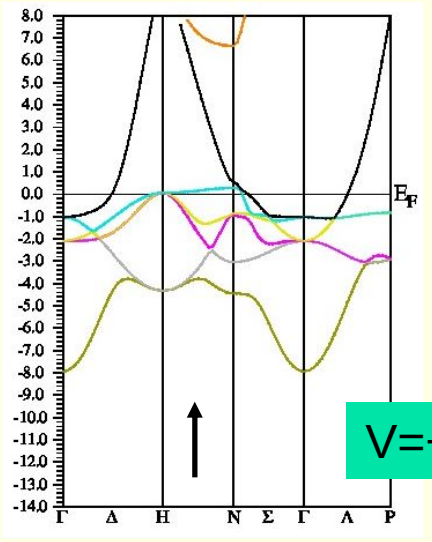

--- MMTOT ---------- in 5 files:
Fe_vol___0.0_rk8_5000k.scf::MMTOT: 2.21
Fe_vol___3.0_rk8_5000k.scf::MMTOT: 2.26
Fe_vol__-3.0_rk8_5000k.scf::MMTOT: 2.16
Fe_vol___6.0_rk8_5000k.scf::MMTOT: 2.31
Fe_vol__-6.0_rk8_5000k.scf::MMTOT: 2.13

- *compare bandstructure and DOS for large/small volumes (restore_lapw for desired volume; x lapw0 "recreates" potentials, adjust EF in case.insp)*




V=-6%




V=+6%