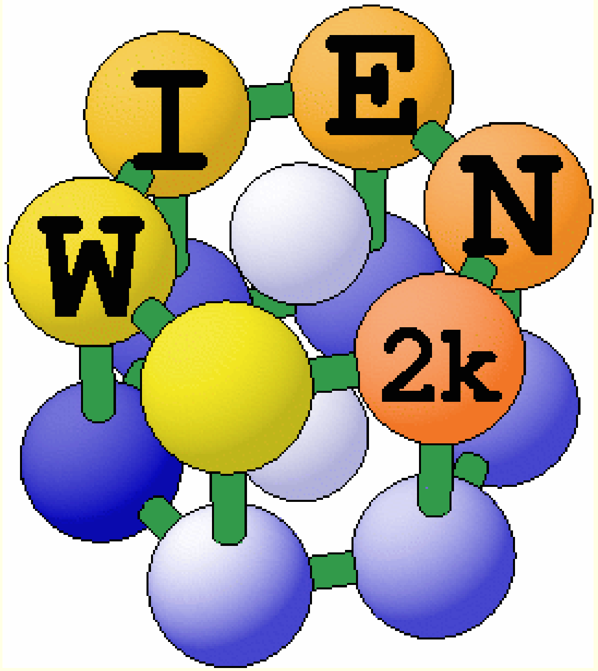
WIEN2k software package



WIEN97: ~500 users
WIEN2k: ~960 users

An Augmented Plane Wave Plus Local Orbital
Program for Calculating Crystal Properties

Peter Blaha
Karlheinz Schwarz
Georg Madsen
Dieter Kvasnicka
Joachim Luitz

November 2001
Vienna, AUSTRIA
Vienna University of Technology

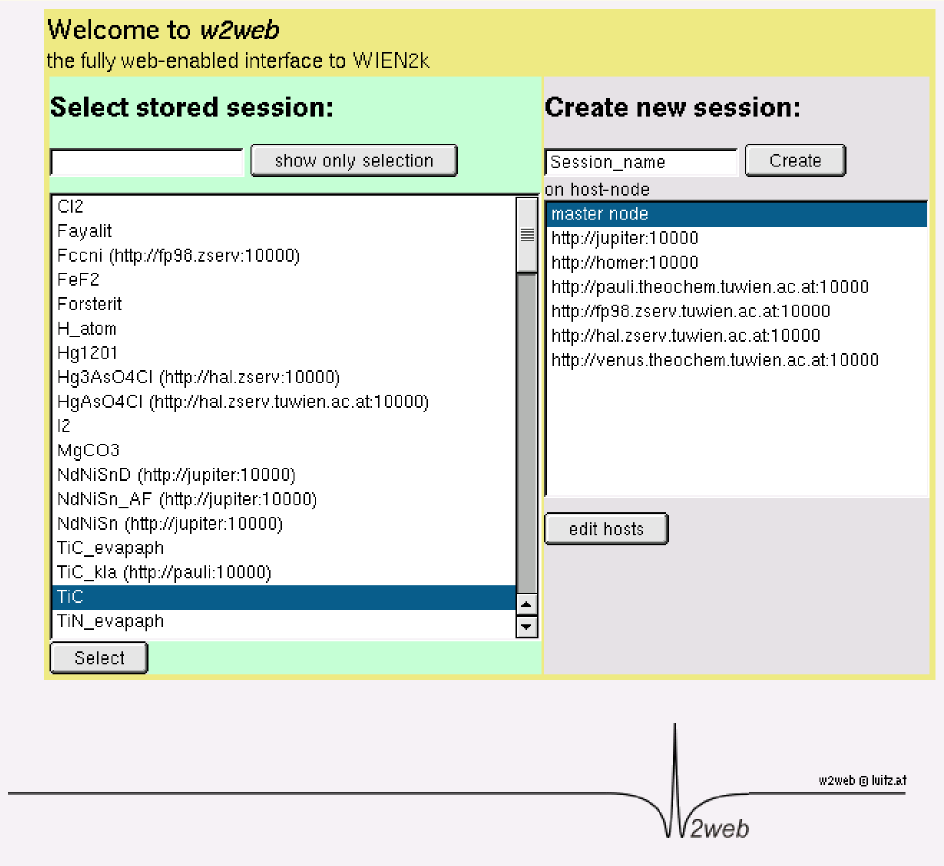
http://www.wien2k.at

General remarks on WIEN2k

- WIEN2k consists of many independent F90 programs, which are linked together via C-shell scripts.
- Each „case“ runs in his own directory ./case
- The „master input“ is called case.struct
- Initialize a calculation: init_lapw
- Run scf-cycle: run_lapw (runsp_lapw)
- You can run WIEN2k using any www-browser and the w2web interface, but also at the command line of an xterm.
- Input/output/scf files have endings as the corresponding programs:
  - *case.output1…lapw1; case.in2…lapw2; case.scf0…lapw0*
- Inputs are generated using STRUCTGEN(w2web) and init_lapw

w2web: the web-based GUI of WIEN2k

- Based on www
  - *WIEN2k can be managed remotely via w2web*
- Important steps:
  - *start w2web on all your hosts*
    - login to the desired host (ssh)
    - w2web (at first startup you will be asked for username/password, port-number, (master-)hostname. creates ~/.w2web directory)
  - *use your browser and connect to the (master) host:portnumber*
    - mozilla http://fp98.zserv:10000
  - *create a new session on the desired host (or select an old one)*



Welcome to *w2web*

the fully web-enabled interface to WIEN2k

Select stored session:

Cl2
Fayalit
Fccni (http://fp98.zserv:10000)
FeF2
Forsterit
H_atom
Hg1201
Hg3AsO4Cl (http://hal.zserv:10000)
HgAsO4Cl (http://hal.zserv.tuwien.ac.at:10000)
I2
MgCO3
NdNiSnD (http://jupiter:10000)
NdNiSn_AF (http://jupiter:10000)
NdNiSn (http://jupiter:10000)
TiC_evapaph
TiC_kla (http://pauli:10000)
TiC
TiN_evapaph

Create new session:

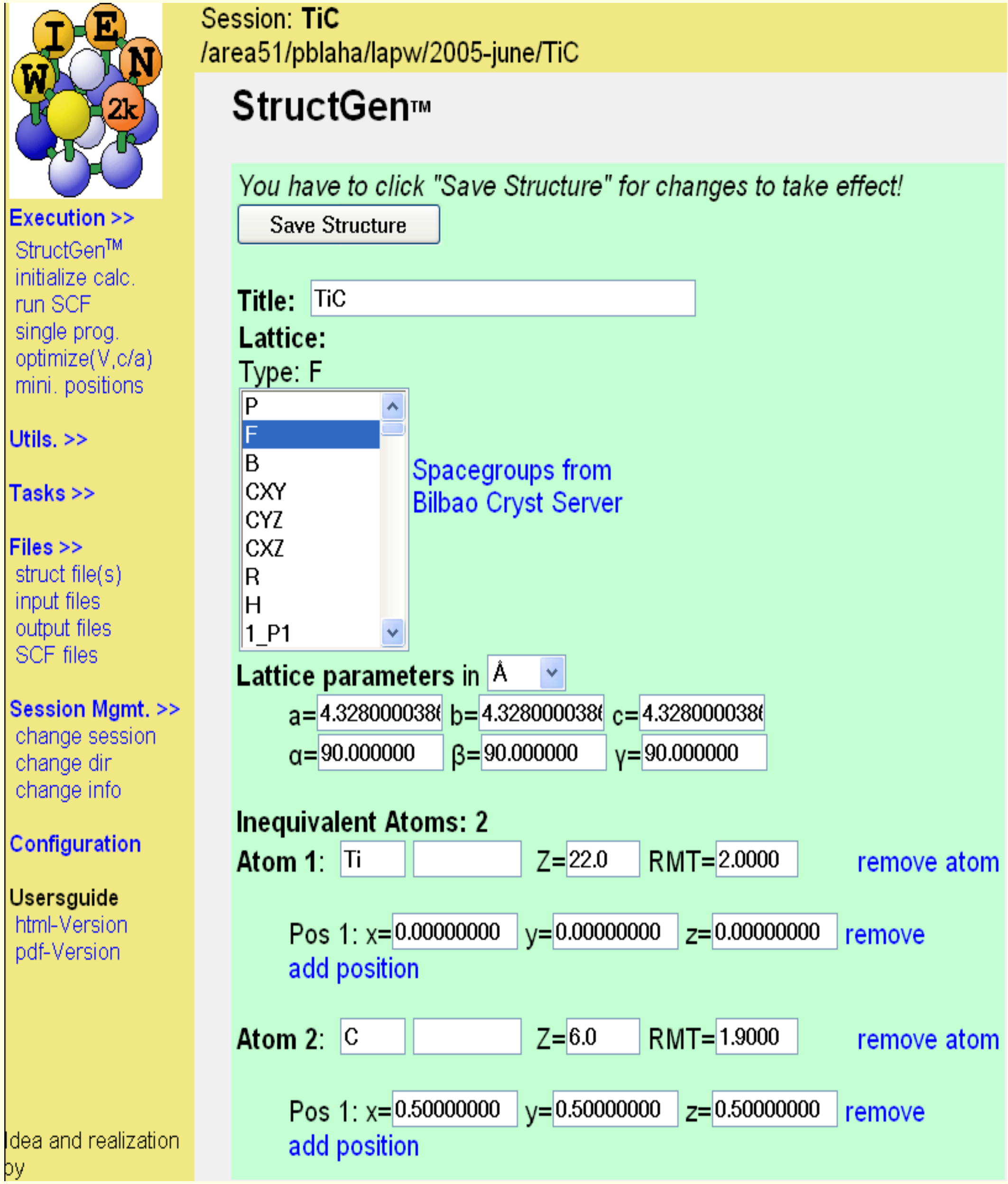
Session_name

on host-node

master node
http://jupiter:10000
http://homer:10000
http://pauli.theochem.tuwien.ac.at:10000
http://fp98.zserv.tuwien.ac.at:10000
http://hal.zserv.tuwien.ac.at:10000
http://venus.theochem.tuwien.ac.at:10000

w2web GUI (graphical user interface)

- Structure generator
  - *spacegroup selection*
  - *import cif file*
- step by step initialization
  - *symmetry detection*
  - *automatic input generation*
- SCF calculations
  - *Magnetism (spin-polarization)*
  - *Spin-orbit coupling*
  - *Forces (automatic geometry optimization)*
- Guided Tasks
  - *Energy band structure*
  - *DOS*
  - *Electron density*
  - *X-ray spectra*
  - *Optics*



Session: **TiC**
/area51/pblaha/lapw/2005-june/TiC

StructGen™

You have to click "Save Structure" for changes to take effect!

Save Structure

Title: TiC

Lattice:

Type: F

Lattice parameters in Å

a=4.328000038 b=4.328000038 c=4.328000038

α=90.000000 β=90.000000 γ=90.000000

Inequivalent Atoms: 2

Atom 1: Ti Z=22.0 RMT=2.0000 remove atom

Pos 1: x=0.00000000 y=0.00000000 z=0.00000000 remove

add position

Atom 2: C Z=6.0 RMT=1.9000 remove atom

Pos 1: x=0.50000000 y=0.50000000 z=0.50000000 remove

add position

Spacegroup $P4_2$/mnm

Structure given by:

spacegroup

lattice parameter

positions of atoms

(basis)

Rutile TiO_2:

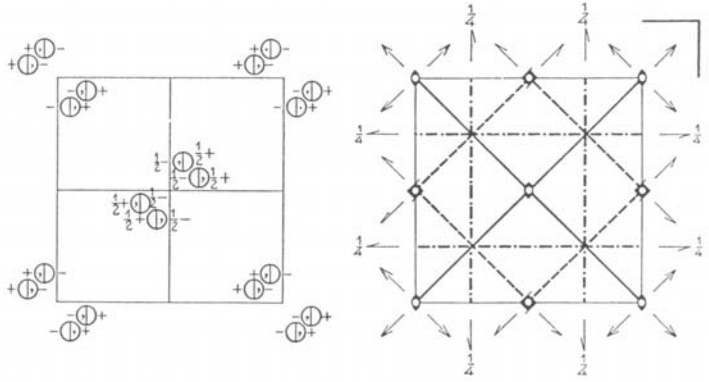
$P4_2$/mnm (136)

a=8.68, c=5.59 bohr

Ti: (0,0,0)

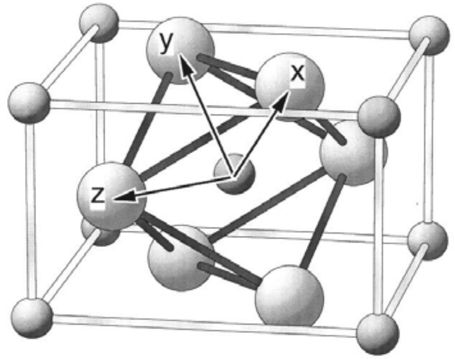
O: (0.304,0.304,0)

$P4_2/mnm$ D_{4h}^{14} No. 136 $P\,4_2/m\,2_1/n\,2/m$ $4/m\,m\,m$ Tetragonal



Origin at centre (*mmm*)

Number of positions, Wyckoff notation, and point symmetry			Co-ordinates of equivalent positions				Conditions limiting possible reflections
16	*k*	1	x,y,z; $x,y,\bar{z}$; y,x,z; $y,x,\bar{z}$;	$\bar{x},\bar{y},z$; $\bar{x},\bar{y},\bar{z}$; $\bar{y},\bar{x},z$; $\bar{y},\bar{x},\bar{z}$;	$\frac{1}{2}+x,\frac{1}{2}-y,\frac{1}{2}+z$; $\frac{1}{2}+x,\frac{1}{2}-y,\frac{1}{2}-z$; $\frac{1}{2}+y,\frac{1}{2}-x,\frac{1}{2}+z$; $\frac{1}{2}+y,\frac{1}{2}-x,\frac{1}{2}-z$;	$\frac{1}{2}-x,\frac{1}{2}+y,\frac{1}{2}+z$; $\frac{1}{2}-x,\frac{1}{2}+y,\frac{1}{2}-z$; $\frac{1}{2}-y,\frac{1}{2}+x,\frac{1}{2}+z$; $\frac{1}{2}-y,\frac{1}{2}+x,\frac{1}{2}-z$.	General: *hkl*: No conditions *hk*0: No conditions 0*kl*: $k+l=2n$ *hhl*: No conditions
8	*j*	*m*	x,x,z; $x,x,\bar{z}$;	$\bar{x},\bar{x},z$; $\bar{x},\bar{x},\bar{z}$;	$\frac{1}{2}+x,\frac{1}{2}-x,\frac{1}{2}+z$; $\frac{1}{2}+x,\frac{1}{2}-x,\frac{1}{2}-z$;	$\frac{1}{2}-x,\frac{1}{2}+x,\frac{1}{2}+z$; $\frac{1}{2}-x,\frac{1}{2}+x,\frac{1}{2}-z$.	Special: as above, plus no extra conditions
8	*i*	*m*	$x,y,0$; $y,x,0$;	$\bar{x},\bar{y},0$; $\bar{y},\bar{x},0$;	$\frac{1}{2}+x,\frac{1}{2}-y,\frac{1}{2}$; $\frac{1}{2}+y,\frac{1}{2}-x,\frac{1}{2}$;	$\frac{1}{2}-x,\frac{1}{2}+y,\frac{1}{2}$; $\frac{1}{2}-y,\frac{1}{2}+x,\frac{1}{2}$.	
8	*h*	2	$0,\frac{1}{2},z$; $\frac{1}{2},0,z$;	$0,\frac{1}{2},\bar{z}$; $\frac{1}{2},0,\bar{z}$;	$0,\frac{1}{2},\frac{1}{2}+z$; $\frac{1}{2},0,\frac{1}{2}+z$;	$0,\frac{1}{2},\frac{1}{2}-z$; $\frac{1}{2},0,\frac{1}{2}-z$.	*hkl*: $h+k=2n$; $l=2n$
4	*g*	*mm*	$x,\bar{x},0$;	$\bar{x},x,0$;	$\frac{1}{2}+x,\frac{1}{2}+x,\frac{1}{2}$;	$\frac{1}{2}-x,\frac{1}{2}-x,\frac{1}{2}$.	
4	*f*	*mm*	$x,x,0$;	$\bar{x},\bar{x},0$;	$\frac{1}{2}+x,\frac{1}{2}-x,\frac{1}{2}$;	$\frac{1}{2}-x,\frac{1}{2}+x,\frac{1}{2}$.	
4	*e*	*mm*	$0,0,z$;	$0,0,\bar{z}$;	$\frac{1}{2},\frac{1}{2},\frac{1}{2}+z$;	$\frac{1}{2},\frac{1}{2},\frac{1}{2}-z$.	
4	*d*	$\bar{4}$	$0,\frac{1}{2},\frac{1}{4}$;	$\frac{1}{2},0,\frac{1}{4}$;	$0,\frac{1}{2},\frac{3}{4}$;	$\frac{1}{2},0,\frac{3}{4}$.	
4	*c*	2/*m*	$0,\frac{1}{2},0$;	$\frac{1}{2},0,0$;	$0,\frac{1}{2},\frac{1}{2}$;	$\frac{1}{2},0,\frac{1}{2}$.	
2	*b*	*mmm*	$0,0,\frac{1}{2}$;	$\frac{1}{2},\frac{1}{2},0$.			
2	*a*	*mmm*	$0,0,0$;	$\frac{1}{2},\frac{1}{2},\frac{1}{2}$.			

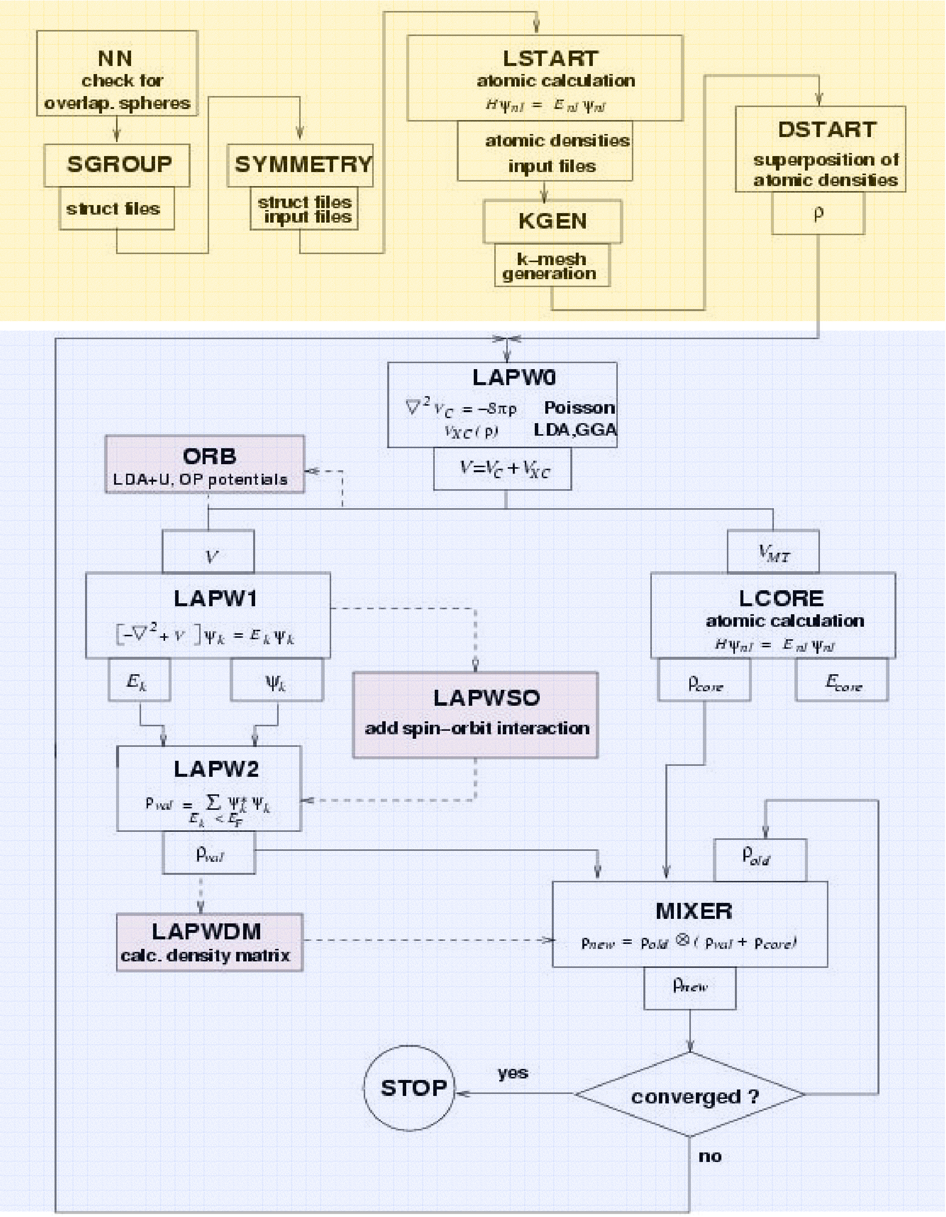


Structure generator

- Specify:
  - *Number of nonequivalent atoms*
  - *lattice type (P, F, B, H, CXY, CXZ, CYZ) or spacegroup symbol*
    - if existing, you must use a SG-setting with inversion symmetry:
      - Si: ±(1/8,1/8,1/8), not (0,0,0)+(1/4,1/4,1/4)!
  - *lattice parameters a,b,c (in Å or bohr)*
  - *name of atoms (Si) and fractional coordinates (position)*
    - as numbers (0.123); fractions (1/3); simple expressions (x-1/2,...)
    - in fcc (bcc) specify just one atom, not the others in (1/2,1/2,0; ...)
- „save structure "
  - *updates automatically Z, r0, equivalent positions and generates case.inst*
- „set RMT and continue": (specify proper "reduction" of NN-distances)
  - *non-overlapping „as large as possible" (saves time), but not larger than 3 bohr*
  - *RMT for sp-elements 10-20 % smaller than for d (f) elements*
  - *largest spheres not more than 50 % larger than smallest sphere*
  - *Exception: H in C-H or O-H bonds: RMT~0.6 bohr (RKMAX~3-4)*
  - *Do not change RMT in a „series" of calculations*
- „save structure – save+cleanup"

Program structure of WIEN2k

- init_lapw
  - *initialization*
  - *symmetry detection (F, I, C-centering, inversion)*
  - *input generation with recommended defaults*
  - *quality (and computing time) depends on k-mesh and R.Kmax (determines #PW)*
- run_lapw
  - *scf-cycle*
  - *optional with SO and/or LDA+U*
  - *different convergence criteria (energy, charge, forces)*
- save_lapw tic_gga_100k_rk7_vol0
  - *cp case.struct and clmsum files,*
  - *mv case.scf file*
  - *rm case.broyd* files*



scf-cycle

- run_lapw [options] (for nonmagnetic cases)
  - *-ec 0.0001* *convergence of total energy (Ry)*
  - *-cc 0.0001* *convergence of charge distance (e^-)*
  - *-fc 1.0* *convergence of forces (mRy/bohr)*
  - *-p* *parallel calculation (needs .machines file)*
  - *-so* *add spin-orbit*
  - *Spacegroups without inversion use automatically lapw1c, lapw2c (case.in1c,in2c)*
  - *If scf-cycle diverges (grep :DIS case.scf): check struture; reduce mixing in case.inm; rm *.broyd* case.scf; x dstart*
- case.scf: master output file, contains history of the scf-cycle
  - *most information is stored with some „labels“ (grep :label case.scf)*
    - :ENE :DIS :FER :CTO001 :NTO001 :QTL001
    - :FOR002: 2.ATOM 19.470 0.000 0.000 19.470
    - :FGL002: 2.ATOM 13.767 13.767 0.000
    - :LAT :VOL :POSxxx

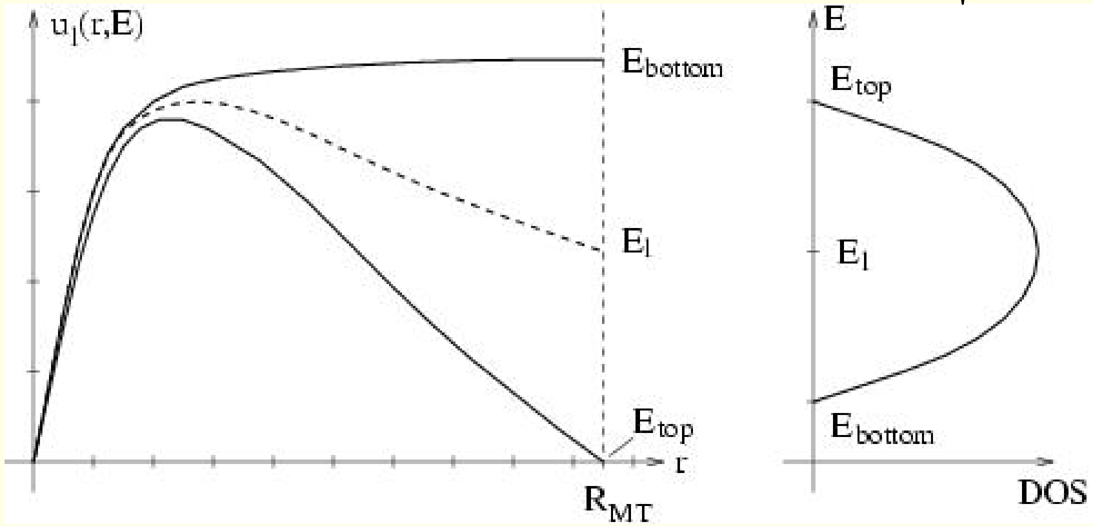
case.in1

```
WFFIL                          (WFPRI, SUPWF)
  7.00    10   4              (R-MT*K-MAX; MAX L IN WF, V-NMT
 0.30      5   0         global E-param with N other, napw
0      0.30        0.000 CONT 1       Es
0     -3.72        0.005 STOP 1       Es-LO with search
1     -2.07        0.010 CONT 1       Ep    with search
1      0.30        0.000 CONT 1       Ep-LO
2      0.30        0.010 CONT 1       0/1…LAPW/APW+lo
```

$$\Psi = \sum_{K_n}^{KMAX} c_{K_n} e^{iK_n r}$$

$$\Phi_{K_n} = \sum_{l}^{l\max} A_{lm} u_l(E_l, r) Y_{lm}$$

$$H^{NS}_{n,m} = \left\langle \Phi_l \middle| V^{NS}_{LM} \middle| \Phi_{l'} \right\rangle$$



case.in1 (cont.), case.in2

```
K-VECTORS FROM UNIT:4     -7.0     1.5     emin/emax window
GAMMA           0    0    0   40   1.0  IX, IY, IZ, IDIV, WEIGHT
                1    0    0   40   6.0
...
X              40    0    0   40   3.0
END
```

case.in2:

```
TOT                              (TOT,FOR,QTL,EFG,FERMI)
-9.0 16.0     0.50 0.05          EMIN, NE, ESEPARMIN, ESEPAR0
TETRA    0.000                   (GAUSS,ROOT,TEMP,TETRA,ALL    eval)
 0 0   4 0   4 4   6 0   6 4
 0 0   4 0   4 4   6 0   6 4
 14.              GMAX(for small H set it to 20-24)
FILE              FILE/NOFILE  write recprlist
```

$$\rho(r)=\sum_{\boxed{LM}}\rho_{LM}(r)Y_{LM}(\hat{r}) \qquad \rho(r)=\sum_{G}^{\boxed{GMAX}}\rho_G e^{iGr}$$

run_lapw -ql 0.05 -in1new 1

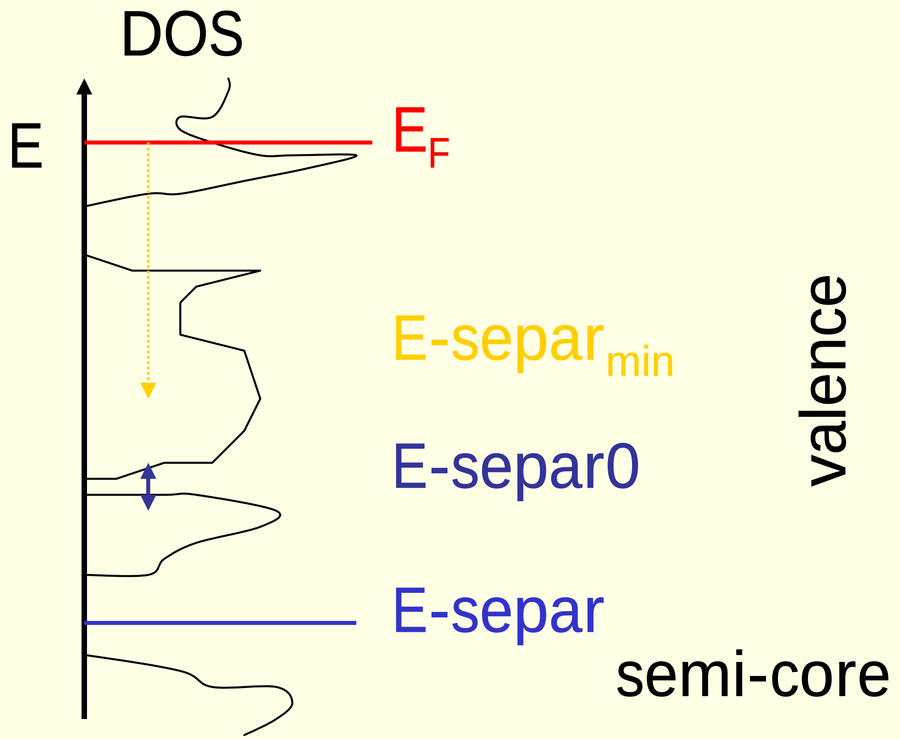
- Alternative case.in1 file produced by write_in1:

case.scf2:

```
Energy to separate semicore and valencestates:   0.34941
:FER  : F E R M I - ENERGY(TETRAH.M.)=   0.79528
       Q-s-low E-s-low   Q-p-low E-p-low   Q-d-low E-d-low
:EPL01:1.9813 -2.6852    5.5892 -1.1099    0.0873  0.0910
       Q-s-hi  E-s-hi    Q-p-hi  E-p-hi    Q-d-hi  E-d-hi
:EPH01:0.0668  0.5152    0.1752  0.6174    1.0614  0.6181
```

➔ case.in1:

```
WFFIL         (WFPRI, SUPWF)
  7.00         10     4
 .49528   6    0
 0     0.515     0.000 CONT 1
 0    -2.685     0.000 CONT 1
 1     0.617     0.000 CONT 1
 1    -1.110     0.000 CONT 1
 2     0.618     0.000 CONT 1
 2     0.091     0.000 CONT 1
 ...
```



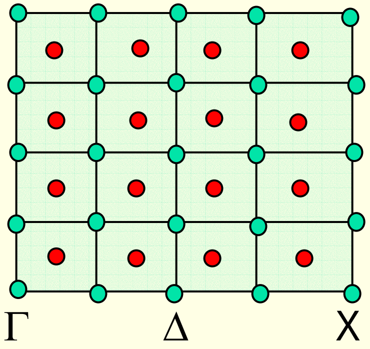
BZ integration, “FERMI”-methods

- Replace the “integral” of the BZ by a finite summation on a mesh of “k-points”

$$\rho(r) = \sum_{n}^{E_n<E_F} \int \psi_{k,n}^{*}\, \psi_{k,n}\, d^3k \quad = \quad \sum_{k,n} w_{k,n}\, \psi_k^{*} \psi_k$$

- weights $w_{k,n}$ depend on k and bandindex n (occupation)
  - *for full “bands” the weight is given by “symmetry”*
    - w(Γ)=1, w(x)=2, w(Δ)=4, w(k)=8

    ⟹ shifted “Monkhorst-Pack” mesh



  - *for partially filled bands (metals) one must find the Fermi-energy (integration up to NE) and determine the weights for each state $E_{k,n}$*
    - linear tetrahedron method (TETRA, eval=999)
    - linear tetrahedron method + “Bloechl” corrections (TETRA)
    - “broadening methods”
      - gauss-broadening (GAUSS 0.005)
      - temperature broadening (TEMP 0.005)
  - *broadening useful to damp scf oszillations, but dangerous (magnetic moment)*

k-mesh generation

- x kgen (generates k-mesh and reduces to irreducible wedge using symmetry)
  - *always "add inversion" except in magnetic spin-orbit calculations*
    - time inversion holds and E(k) = E(-k)
  - *always "shift" the mesh for scf-cycle*
    - gaps often at Γ ! (might not be in your mesh)
  - *small unit cells and metals require large k-mesh (1000-100000)*
  - *large unit cells and insulators need only 1-10 k-points*
  - *use at first a fairly coarse mesh for scf*
  - *continue later with finer mesh*
    - mesh was good if nothing changes and scf terminates after few (3) iterations
  - *use an even finer meshes for DOS, spectra, optics,…*

Program execution:

- All programs are executed via the „master“ shell-script „x“:

  x lapw2 –up –c

- This generates a „def“ file: lapw2.def

```
5,'tin.in2c',         'old',      'formatted'
6,'tin.output2up',    'unknown','formatted'
8,'tin.clmvalup',     'unknown','formatted'
10,'./tin.vectorup','unknown','unformatted'
```

- and executes: lapw2c lapw2.def
- All WIEN2k-shell scripts have long and short names:
  - *x_lapw; runsp_lapw, runfsm_lapw → x; runsp; runfsm*
- All scripts have a „help“ switch „-h“, which explains flags and options (without actually execution)

  x –h *x lapw1 -h*

Getting help

- *_lapw –h „help switch“ of all WIEN2k-scripts
- help_lapw:
  - *opens usersguide.pdf; Use ^f keyword to search for an item („index“)*
- html-version of the UG: ($WIENROOT/SRC_usersguide/usersguide.html)
- http://www.wien2k.at/reg_user
  - *FAQ page with answers to common questions*
  - *Update information: When you think the program has an error, please check newest version*
  - *Textbook section: DFT and the family of LAPW methods by S.Cottenier*
  - *Mailing-list:*
    - subscribe to the list (always use the same email)
    - full text search of the „digest“ (your questions may have been answered before)
    - posting questions: Provide sufficient information, locate your problem (case.dayfile, *.error, case.scf, case.outputX).
    - „My calculation crashed. Please help.“ This will most likely not be answered.

Properties with WIEN2k - I

- Energy bands
  - *classification of irreducible representations*
  - *´character-plot´ (emphasize a certain band-character)*
- Density of states
  - *including partial DOS with l and m- character (eg. p_x , p_y , p_z)*
- Electron density, potential
  - *total-, valence-, difference-, spin-densities, ρ of selected states*
  - *1-D, 2D- and 3D-plots (Xcrysden)*
  - *X-ray structure factors*
  - *Bader´s atom-in-molecule analysis, critical-points, atomic basins and charges ($\nabla\rho.\vec{n}=0$)*
  - *spin+orbital magnetic moments (spin-orbit / LDA+U)*
- Hyperfine parameters
  - *hyperfine fields (contact + dipolar + orbital contribution)*
  - *Isomer shift*
  - *Electric field gradients*

Properties with WIEN2k - II

- Total energy and forces
  - *optimization of internal coordinates, (MD, BROYDEN)*
  - *cell parameter only via E_{tot} (no stress tensor)*
  - *elastic constants for cubic cells*
  - *Phonons via supercells*
    - interface to PHONON (K.Parlinski) – bands, DOS, thermodynamics, neutrons
- Spectroscopy
  - *core level shifts*
  - *X-ray emission, absorption, electron-energy-loss (with core holes)*
    - core-valence/conduction bands including matrix elements and angular dep.
  - *optical properties (dielectric function in RPA approximation, JDOS including momentum matrix elements and Kramers-Kronig)*
  - *fermi surface (2D, 3D)*

Properties with WIEN2k - III

- New developments (in progress)
  - *non-collinear magnetism (available on request: www.wien2k.at)*
  - *transport properties (Fermi velocities, Seebeck, conductivity, thermoelectrics, ..) (G.Madsen's BotzTrap code)* www.chem.au.dk/~webuorg/new/groups/gm/gm.html)
  - *non-linear optics (available on request)*
  - *Bethe-Salpeter equation (Excitons)*
  - *Compton profiles*
  - *linear response (phonons, E-field) (C.Ambrosch-Draxl)*
  - *stress tensor (C.Ambrosch-Draxl)*
  - *approximate HartreeFock (+Hybrid functionals), exact exchange, GW*
  - *grid-computing*

Cohesive energy

$$E_{A_xB_y}^{cohes.} = E^{crystal} - xE_A^{atom} - yE_B^{atom}$$

- $E^{crystal}$: scalar-relativistic valence (or approx. SO)

- E^{atom} : LSTART: fully-relativistic➔ inconsistent description

  ➔ for heavier elements (2nd row):

  supercell with one atom in a ~30 bohr FCC box (identical RMT, RKmax, 1 k-point, spinpolarized)

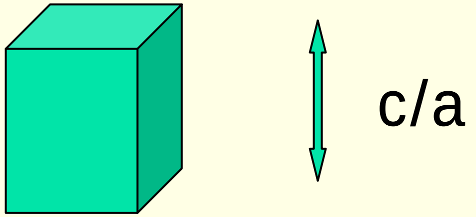
Structural optimizations:

- Lattice parameters, volume, c/a ratio only via total energies:
  - *x optimize: create a series of “struct” files + script “optimize.job”*
    - select volume or c/a, ...
    - select number of cases and desired changes in volume (in % of V_0)
  - *edit optimize.job*
    - adapt to your need: change / uncomment various lines, eg.:
      - select different convergence parameters, parallelization, more iterations (-i 40)
      - different “save_lapw” (into a directory with specific names)
      - replace “run_lapw” by “runsp_lapw” or min_lapw –I –j “run_lapw –I –fc 1”
  - *execute optimize.job*
  - *plot (analyse) the results*

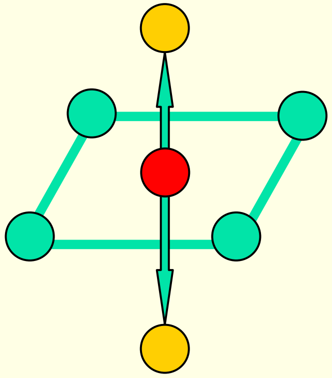
  - *combinations of volume and c/a are possible:*
    - “x optimize” always uses case_initial.struct (if present)
    - do a “volume” optimization to create case_vol_xx.struct files
    - copy the respective case_vol_xx.struct file to case_initial.struct
    - x optimize with “c/a” for this particular volume and proceed as above.

Symmetry:

- WIEN „preserves“ symmetry:
  - *c/a optimization of „cubic“ TiC:*
    - change c lattice parameter in TiC.struct (tetragonal distortion, #sym.op=0)
    - init_lapw
    - change c back to cubic
    - x optimize ...



  - *„Jahn-Teller“ distortion:*
    - when you start with a perfect octahedra, you will never get any distortion
    - ➔start with slightly distorted positions



Total energies and atomic forces

(Yu et al.; Kohler et al.)

- Total Energy:
  - *Electrostatic energy*
  - *Kinetic energy*
  - *XC-energy*

$$U[\rho] = \tfrac{1}{2}\int d^3\vec{r}\ \rho(\vec{r})V_{es}(\vec{r}) + \tfrac{1}{2}\sum_\alpha Z_\alpha V_{es}^\alpha(\vec{r})$$

$$T[\rho] = \sum_i n_i \varepsilon_i - \int d^3\vec{r}\ \rho(\vec{r})V_{eff}(\vec{r})$$

$$E_{xc}[\rho] = \int d^3\vec{r}\ \rho(\vec{r})\varepsilon_{xc}(\vec{r})$$

- Force on atom α:

$$\vec{F}^\alpha = \frac{-dE_{tot}}{d\vec{R}_\alpha} = F_{HF}^\alpha + F_{core}^\alpha + F_{val}^\alpha$$

  - *Hellmann-Feynman-force*
  - *Pulay corrections*
    - Core
    - Valence
- expensive, contains a summation of matrix elements over all occupied states

$$F_{HF}^\alpha = Z_\alpha \sum_{m=-1}^{1} \lim_{r_\alpha \to 0} \frac{V_{1m}^{es}(r_\alpha)}{r_\alpha} \nabla_\alpha \left[r_\alpha Y_{1m}(\hat{r})\right]$$

$$F_{core}^\alpha = -\int \rho_{core}(r)\nabla_\alpha V_{eff}(r)\, d\vec{r}$$

$$F_{val}^\alpha = \int_\alpha V_{eff}(r)\nabla_\alpha \rho_{val}(r)\, d\vec{r} + \sum_{k,i} n_i \sum_{K,K'} c_i^*(K')c_i(K) \times$$

$$\left[(K^2 - \varepsilon_i)\oint \phi_{K'}^*(r)\phi_K(r)\, dS_\alpha - i(K-K')\langle \phi_{K'} | H - \varepsilon_i | \phi_K \rangle_\alpha\right]$$

Optimization of internal parameters using “forces”

- Forces only for “free” structural parameters:
  - *NaCl: (0,0,0), (0.5,0.5,0.5) : all positions fixed by symmetry*
  - *TiO_2: Ti (0,0,0), O (u,u,0): one free parameter (u,x,y,z)*
- Forces are only calculated when using:
  - *run_lapw –fc 1.0   (mRy/bohr)*
    - grep :for002 case.scf
      - 200.
      - -130.
      - 140.
      - 135   only F_{HF} + F_{core}
      - 120
      - 122   forces converging
      - 121
      - -12.3 F_{HF} + F_{core} + F_{val}, only this last number is correct
- Forces are usefull for
  - *structural optimization (of internal parameters)*
  - *phonons*

Structural optimization of internal parameters

- /home/pblaha/tio2> min_lapw -h
  - *OPTIONS:*
    - -j JOB -> job-file JOB (run_lapw -I -fc 1. -i 40)
    - -p -> does a k-point parallel calculation
    - -sp -> does a spin-polarized calculation (runsp_lapw)
    - -NI/-I -> without/with initialization of input-files (after inM changes)
    - -i NUMBER -> max. NUMBER (50) of structure changes
  - *CONTROL FILES:*
    - .minstop stop after next structure change
- /home/pblaha/tiO2> cat tio2.inM
    - PORT 2.0 #(NEW1, NOSE, MOLD, tolf (a4,f5.2))
    - 1.0 1.0 1.0 0.7 # Atom1
    - 1.0 1.0 1.0 0.7 # Atom2 (NEW1: 1,2,3:delta, 4:eta(1=MOLD))
- monitor minimization in file case.scf_mini
  - *contains last iteration of each geometry step*
  - *If minimizations gets stuck or oscillates:*
    - change deltas, rm case.tmpM

Optimization of atomic posistions (E-minimization via forces)

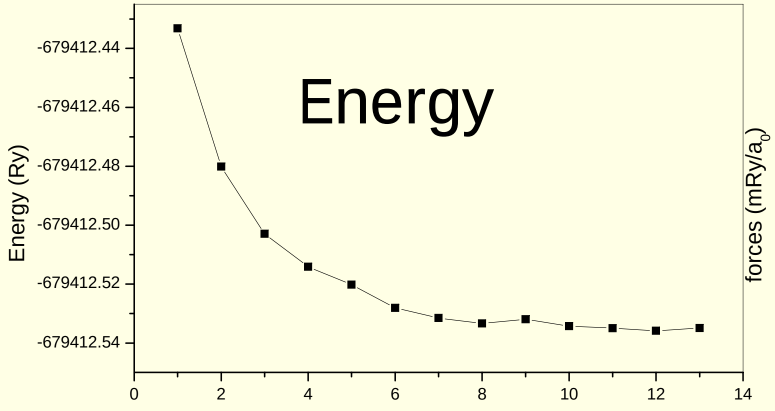
- damped Newton mechanics scheme

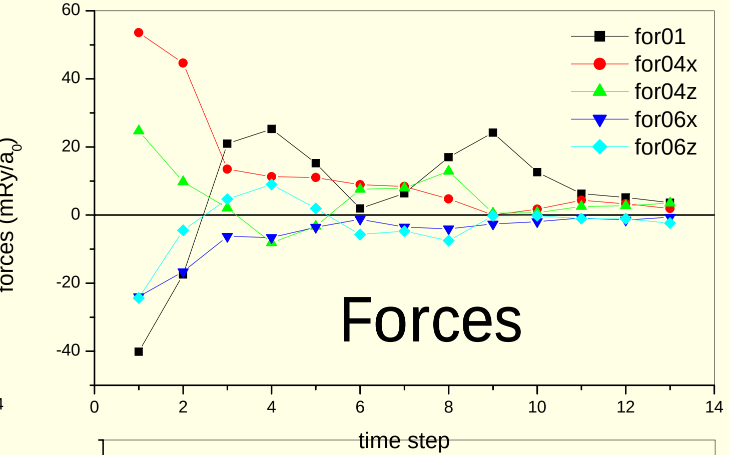
(at the moment not very user friendly, input adjustment by hand)

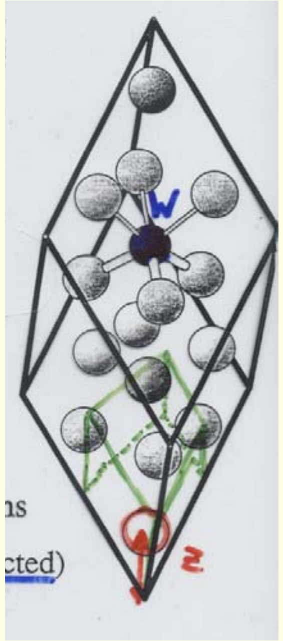
$$\vec{F}_i = -\frac{\partial E}{\partial \vec{R}_i}$$

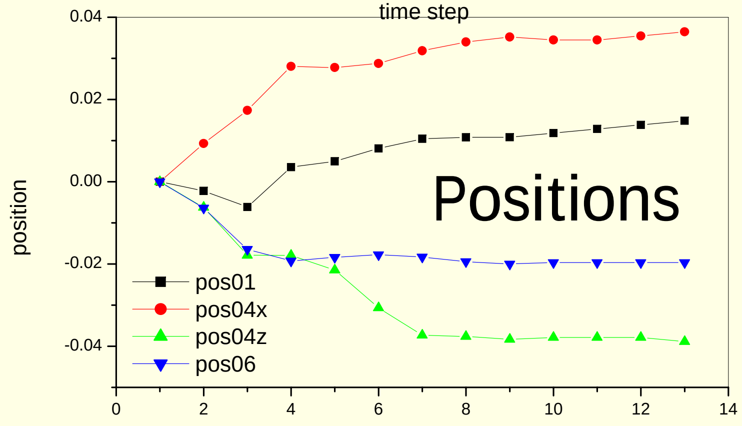
- new quite efficient quasi-Newton (Broyden) scheme

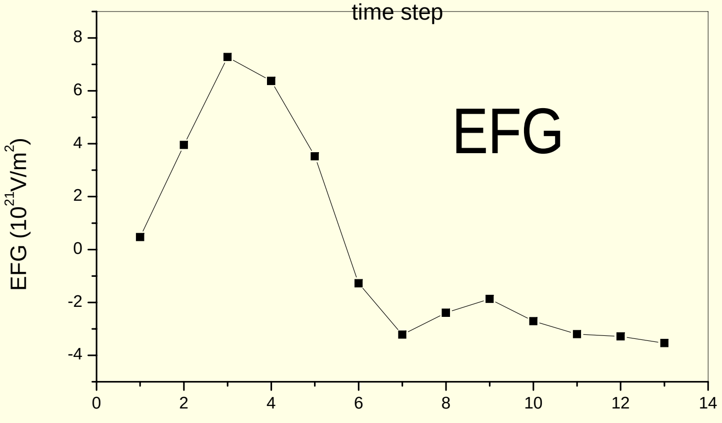
W impurity in Bi (2x2x2 supercell: $Bi_{15}W$)



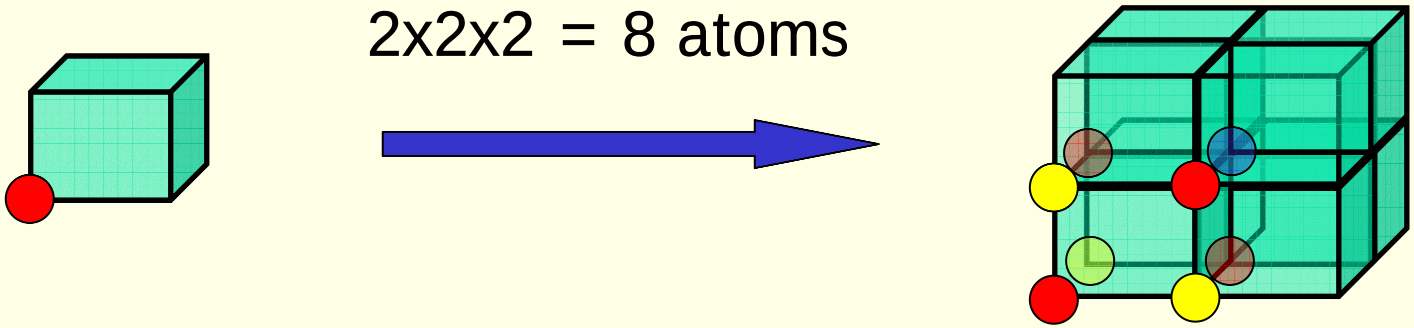








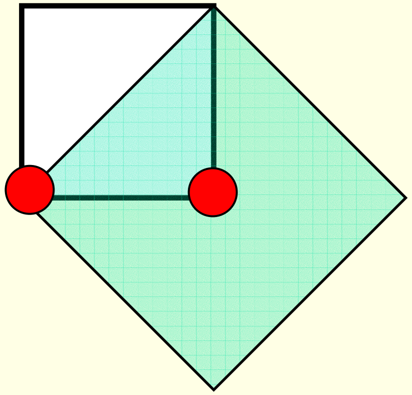
Supercells


2x2x2 = 8 atoms

(0,0,0)	P➔ 8 atoms	(0,0,0)	(.5,0,0)	(.5,.5,0)	(.5,.5,.5)
			(0,.5,0)	(.5,0,.5)	
			(0,0,.5)	(0,.5,.5)	
	B➔ 4 atoms	yes	yes	no	no
	F➔ 2 atoms	yes	no	no	yes

4x4x4 supercells: P (64), B (32), F (16) atoms

Supercells

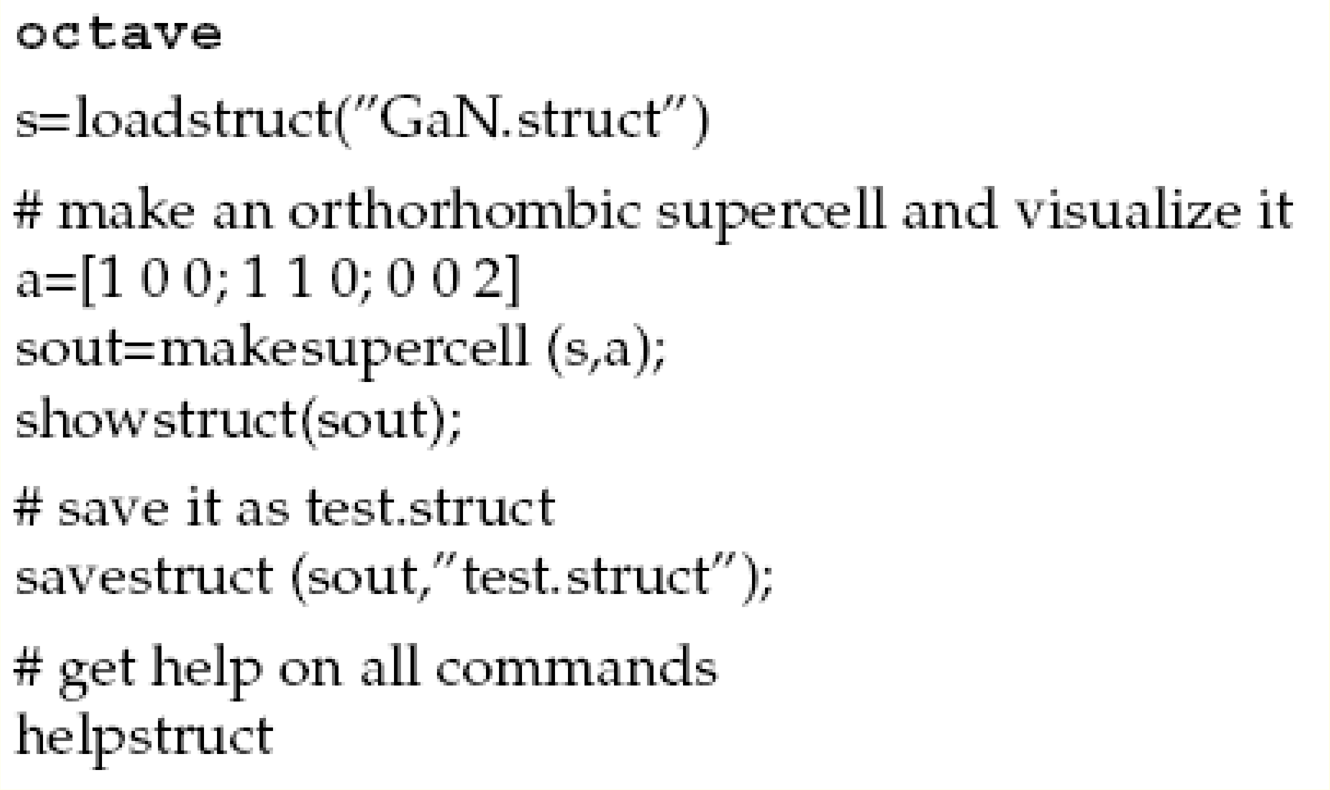


$\sqrt{2}x\sqrt{2}$ supercells (1 ➔ 2 atoms)

- Program „supercell“:
  - *start with „small“ struct file*
  - *specify number of repetitions in x,y,z (only integers, e.g. 2x2x1)*
  - *specify P, B or F lattice*
  - *add „vacuum“ for surface slabs (only (001) indexed surfaces)*
  - *shift all atoms in cell*
- You must break symmetry!!!
  - *replace (impurities, vacancies) or displace (phonons) at least 1 atom*
- At present „supercell“ works only along unit-cell axes!!!

Structeditor (by R.Laskowski)

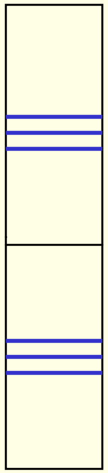
- requires octave (mathlab) and opendx (visualization)
- allows complex operations on struct-files



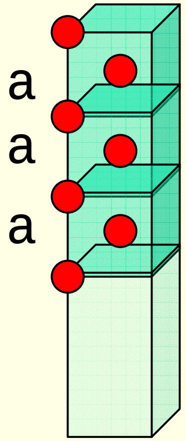
```
octave
s=loadstruct("GaN.struct")
# make an orthorhombic supercell and visualize it
a=[1 0 0; 1 1 0; 0 0 2]
sout=makesupercell (s,a);
showstruct(sout);
# save it as test.struct
savestruct (sout,"test.struct");
# get help on all commands
helpstruct
```

Surfaces

- 2D-slabs with finite number of layers with „vacuum“ in 3rd dimension



bcc (001) 7 layers:



(0 0 6z)
(.5 .5 5z)
(0 0 4z)
(.5 .5 3z)
(0 0 2z)
(.5 .5 z)
(0 0 0)

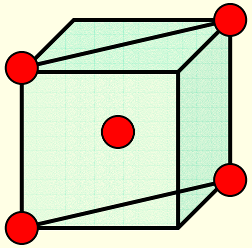
shift to
➔
inversion

(.5 .5 +/-3z)
(0 0 +/-2z)
(.5 .5 +/-z)
(0 0 0)

with lattice parameters:
a, a, c=(3a+15-20bohr vacuum)

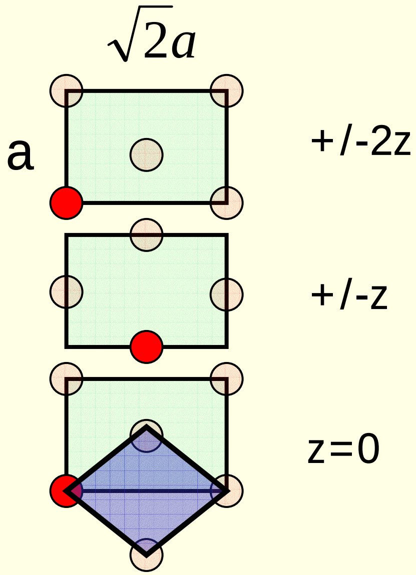
z= a/2c

bcc (110):

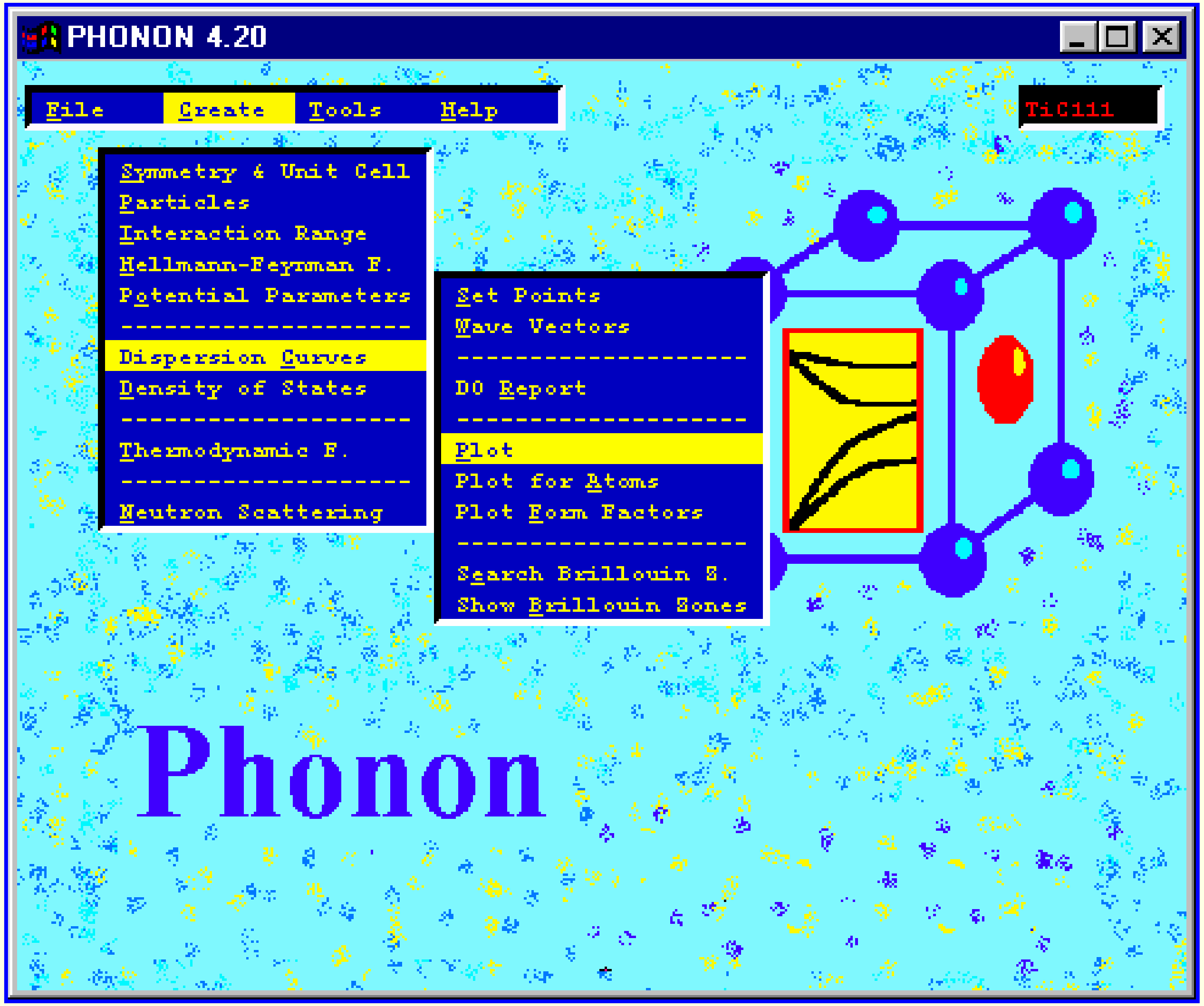


orthorhombic CXY-lattice: a, $\sqrt{2}a$, c

(0 0 0) z=a/$\sqrt{2}a$ c
(0 .5 +/-z)
(0 0 +/-2z)

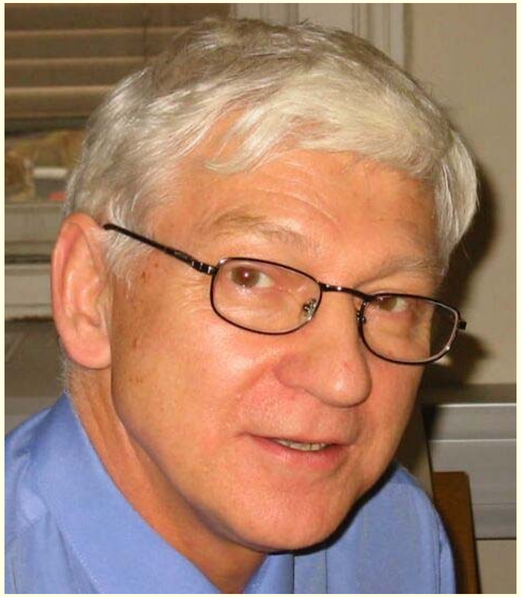


Calculations of Phonons: The Direct Method



WIEN2k + Phonon

Copyright by K.Parlinski



http://wolf.ifj.edu.pl/phonon/

(alternatively use D.Alfe`s PHON code
+W2P-interface from G.Madsen)

THEORY OF DIRECT METHOD

System energy E (at $T = 0$) as a function of atomic positions $\mathbf{R}(\mathbf{n}, \mu)$ is

$$E(\mathbf{R}(\mathbf{n}, \mu), ..\mathbf{R}(\mathbf{m}, \nu), \ldots) = E_o + \frac{1}{2} \sum_{\mathbf{n},\mu,\mathbf{m},\nu} \Phi(\mathbf{n}, \mu, \mathbf{m}, \nu)\mathbf{U}(\mathbf{n}, \mu)\mathbf{U}(\mathbf{m}, \nu)$$

where the *force constant matrix* are

$$\Phi_{i,j}(\mathbf{n}, \mu, \mathbf{m}, \nu) = \left. \frac{\partial^2 E}{\partial \mathbf{R}_i(\mathbf{n},\mu) \partial \mathbf{R}_j(\mathbf{m},\nu)} \right]_o$$

is defined at $\frac{\partial E}{\partial \mathbf{R}_i(\mathbf{n},\mu)} \mid_o = 0$.

The dynamical matrix is defined as

$$\mathbf{D}(\mathbf{k}; \mu, \nu) = \frac{1}{\sqrt{M_\mu M_\nu}} \sum_{\mathbf{m}} \Phi(0, \mu; \mathbf{m}, \nu)\, exp\{-2\pi i \mathbf{k} \cdot [\mathbf{R}(0, \mu) - \mathbf{R}(\mathbf{m}, \nu)]\}$$

$\mathbf{m}$ runs over *all* atoms. Diagonalization of the dynamical matrix

$$\boxed{\omega^2(\mathbf{k}, j)\mathbf{e}(\mathbf{k}, j) = \mathbf{D}(\mathbf{k})\mathbf{e}(\mathbf{k}, j)}$$

gives phonon frequencies $\omega^2(\mathbf{k}, j)$ and polarization vectors $\mathbf{e}(\mathbf{k}, j)$.

Any atomic displacement $\mathbf{U}(\mathbf{m}, \nu)$ generates forces

$$\mathbf{F}(\mathbf{n}, \mu) = -\partial E / \partial \mathbf{R}(\mathbf{n}, \mu)$$

on all other atoms. Hence

$$\boxed{F_i(\mathbf{n}, \mu) = -\Sigma_{\mathbf{m},\nu,j}\, \Phi_{i,j}(\mathbf{n}, \mu, \mathbf{m}, \nu) U_j(\mathbf{m}, \nu)}$$

Master equation of direct method.

CUMMULANT FORCE CONSTANTS

Displace an atom by $\mathbf{U}(\mathbf{m},\nu)$

$$F_i(\mathbf{n},\mu) = -\sum_{\mathbf{L}} \Phi_{i,j}(\mathbf{n},\mu,\mathbf{m}+\mathbf{L}),\nu)U_j(\mathbf{m},\nu)$$

$\mathbf{L} = (L_a, L_b, L_c)$ are the indices of supercell lattice constants.

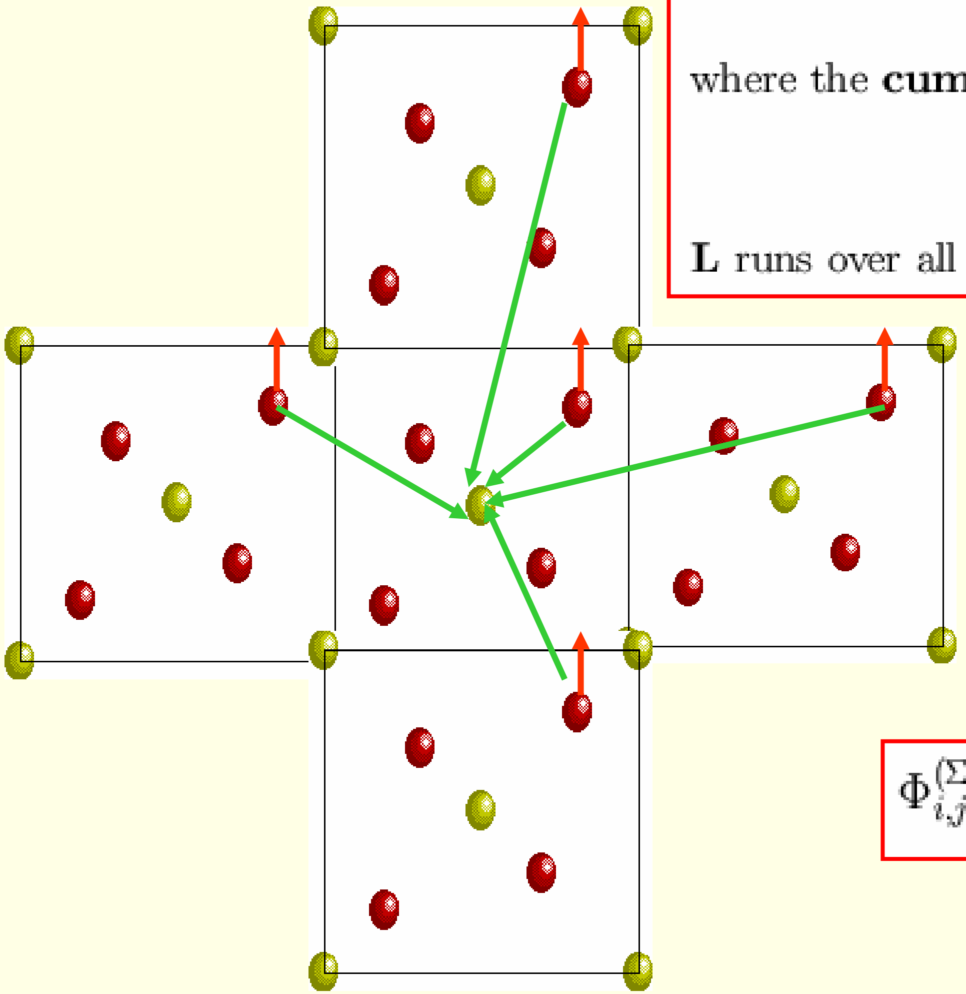
or

$$F_i(\mathbf{n},\mu) = -\Phi^{(\Sigma)}_{i,j}(\mathbf{n},\mu,\mathbf{m},\nu)U_j(\mathbf{m},\nu)$$

where the **cummulant force constant** is

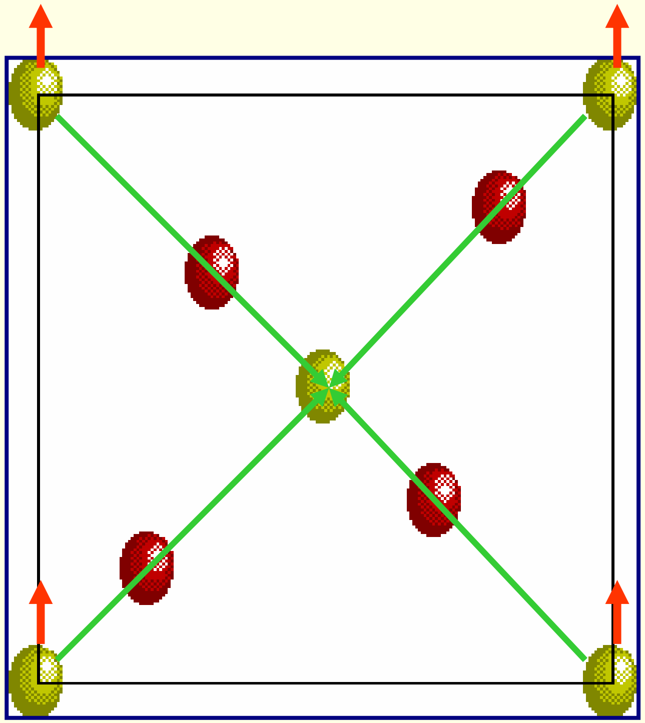
$$\Phi^{(\Sigma)}_{i,j}(\mathbf{n},\mu,\mathbf{m},\nu) = \sum_{\mathbf{L}} \Phi_{i,j}(\mathbf{n},\mu,\mathbf{m}+\mathbf{L},\nu)$$

$\mathbf{L}$ runs over all supercell images.



$$\boxed{\Phi^{(\Sigma)}_{i,j}(\mathbf{n},\mu,\mathbf{m},\nu) = \sum_{\mathbf{L}} \Phi_{i,j}(\mathbf{n},\mu,\mathbf{m}+\mathbf{L},\nu)}$$

Supercell Force Constants



For atoms at supercell surface, edges, corners we define **supercell force constants** (SC)

$$\boxed{\Phi_{i,j}^{(SC)}(\mathbf{n},\mu,\mathbf{m},\nu) = \frac{1}{n_{\mathbf{m}}}\Phi_{i,j}^{(\Sigma)}(\mathbf{n},\mu,\mathbf{m}+\mathbf{L},\nu)}$$

n_m – number of equivalent atoms, on surfaces, edges, or corners of supercell.
n_m = 1 for all inner atoms of supercell

Master equation of direct method

$$\boxed{F_i(\mathbf{n},\mu) = -\sum_{(\mathbf{m},\nu)\in SC}\Phi_{i,j}^{(SC)}(\mathbf{n},\mu,\mathbf{m},\nu)U_j(\mathbf{m},\nu)}$$

Sum (**m**, ν) over atoms within the supercell

Supercell dynamical matrix.   Exact wave vectors.

Conventional dynamical matrix:

$$\mathbf{D}(\mathbf{k};\mu,\nu)=\frac{1}{\sqrt{M_\mu M_\nu}}\sum_{\mathbf{m}}\Phi(0,\mu;\mathbf{m},\nu)\,exp\{-2\pi i\mathbf{k}\cdot[\mathbf{R}(0,\mu)-\mathbf{R}(\mathbf{m},\nu)]\}$$

Supercell dynamical matrix:

$$\mathbf{D}^{(SC)}(\mathbf{k};\mu,\nu)=\frac{1}{\sqrt{M_\mu M_\nu}}\sum_{\mathbf{m}\in SC}\Phi^{(SC)}(0,\mu;\mathbf{m},\nu)\,exp\{-2\pi i\mathbf{k}\cdot[\mathbf{R}(0,\mu)-\mathbf{R}(\mathbf{m},\nu)]\}$$

These two matrices are equal if

$$\mathbf{D}^{(SC)}(\mathbf{k};\mu,\nu)=\mathbf{D}(\mathbf{k};\mu,\nu)$$

- **interaction range** is confined **to interior** of supercell (supercell is big enough)
- wave vector is **commensurate with the supercell** and fulfils the condition (independent of interaction range):

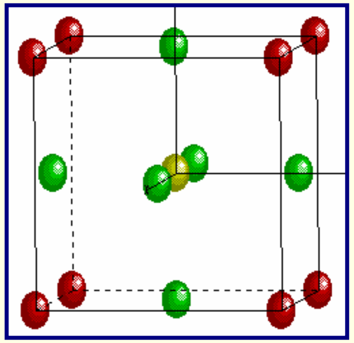
$$exp\{-2\pi i\mathbf{k}_s\cdot\mathbf{L}\}=1$$

At wave vectors $\mathbf{k}_s$ the phonon frequencies are “exact”, provided the **supercell contains the complete list of neighbors.**

Wave vectors $\mathbf{k}_s$ are commensurate with the supercell size.

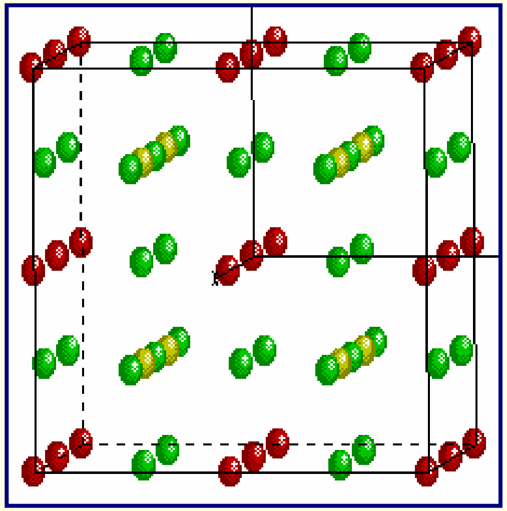
Exact wave vectors

1x1x1



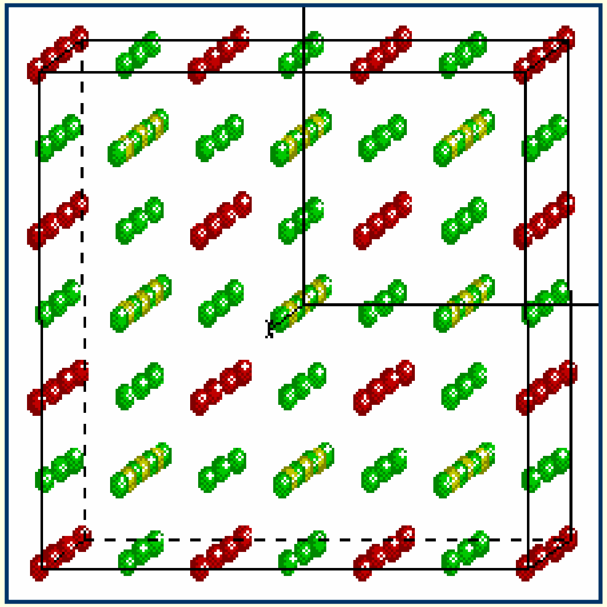
Exact: Γ

2x2x2



Exact: Γ, X, M, R

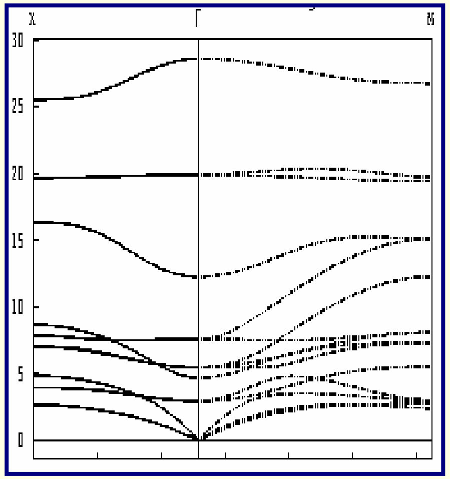
3x3x3

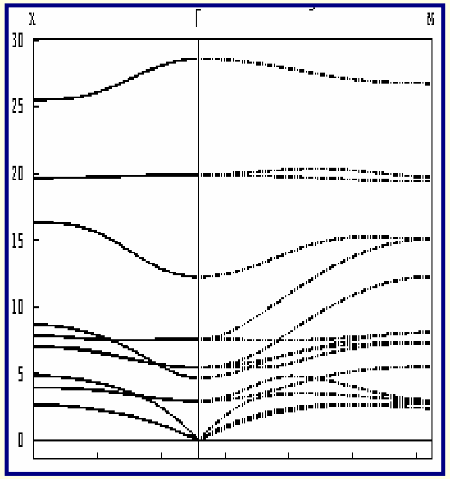


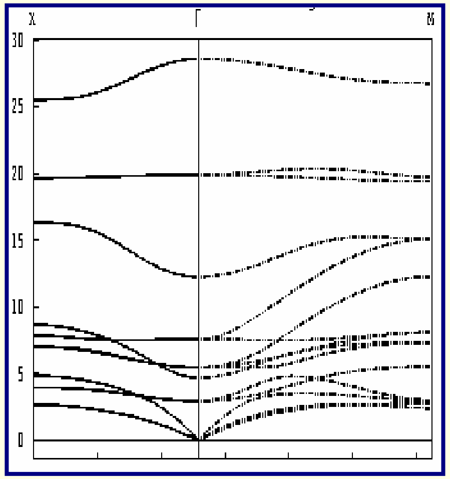
Exact: Γ

X Γ M

↓ Γ ↓



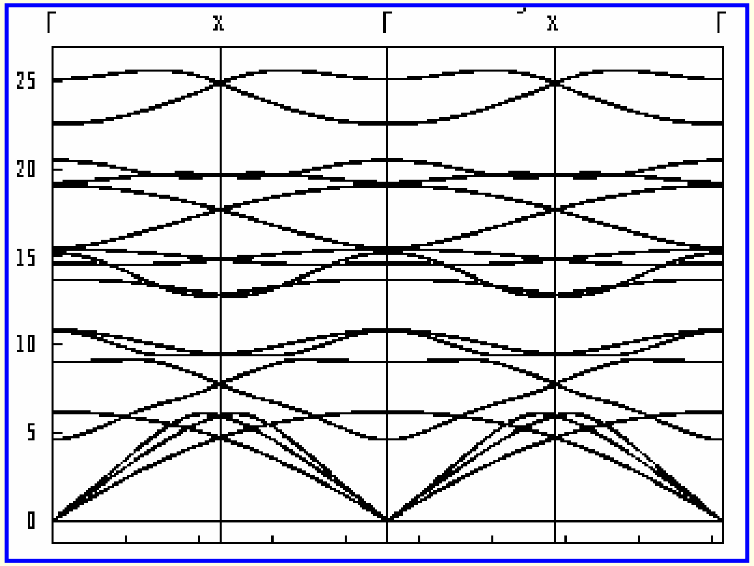




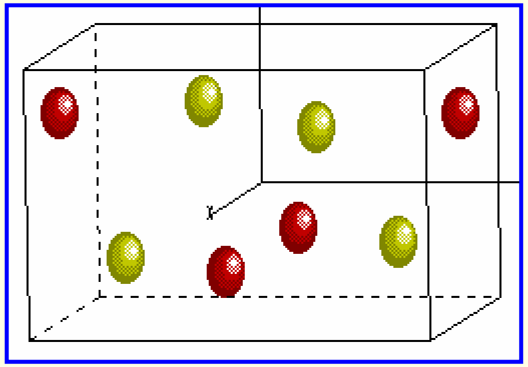
Phonon dispersions + density of states

Frequency

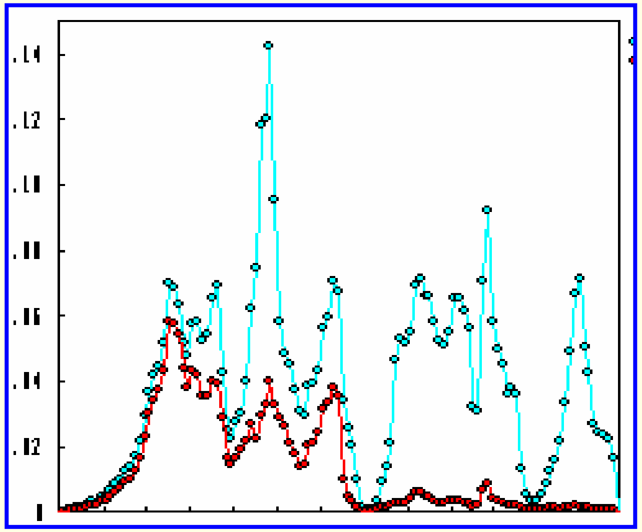
ω



GeO_2 P4_2/mnm

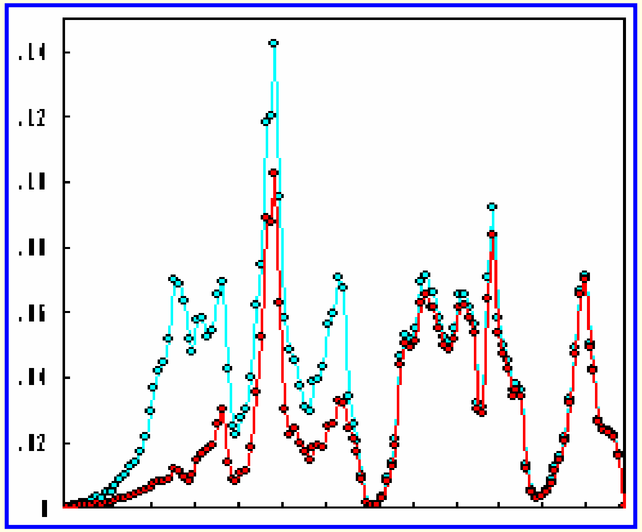


Total + Germanium



ω

Total + Oxygen



ω

Thermodynamic functions of phonon vibrations

Internal energy:

$$E \;= \frac{1}{2} r \int_0^\infty d\omega \, g(\omega) \, (\hbar\omega) \, coth \left(\frac{\hbar\omega}{2k_B T} \right)$$

Free energy:

$$F \;= r k_B T \int_0^\infty d\omega \, g(\omega) \, ln \left[2 sinh \left(\frac{\hbar\omega}{2k_B T} \right) \right]$$

Entropy:

$$S \;= r k_B \int_0^\infty d\omega \, g(\omega) \left\{ \left(\frac{\hbar\omega}{2k_B T} \right) \left[coth \left(\frac{\hbar\omega}{2k_B T} \right) - 1 \right] - ln \left[1 - exp \left(- \frac{\hbar\omega}{k_B T} \right) \right] \right\} (79)$$

Heat capacity C$_v$:

$$C \;= r k_B \int_0^\infty d\omega \, g(\omega) \left(\frac{\hbar\omega}{k_B T} \right)^2 \frac{exp(\frac{\hbar\omega}{k_B T})}{\left[exp\left(\frac{\hbar\omega}{k_B T} \right) - 1 \right]^2}$$

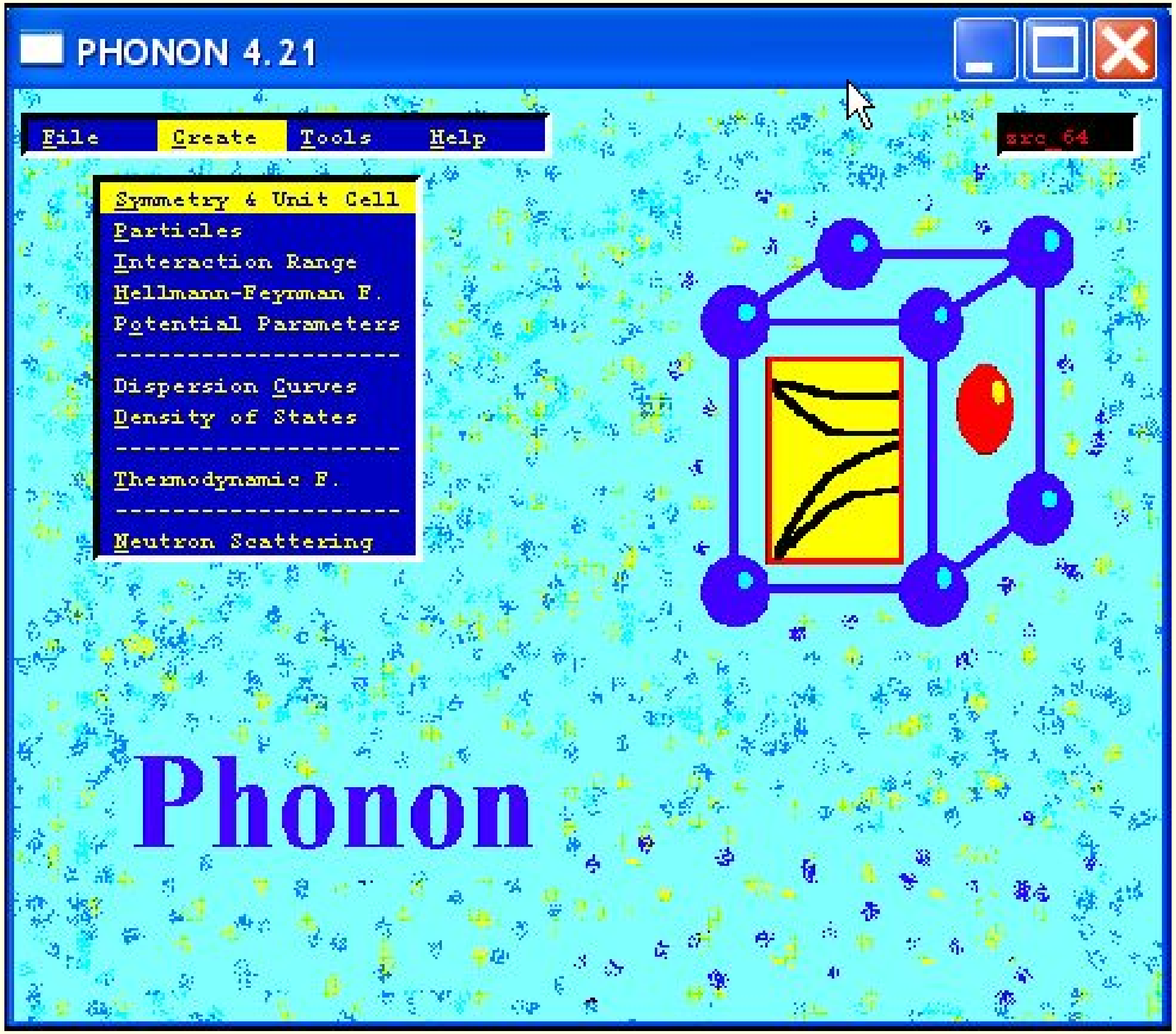
Thermal displacements:

$$B_{ij}(\mu) = < U_i(\mu) U_j(\mu) >$$

$$B_{ii}(\mu) = \frac{\hbar r}{2M_\mu} \int_0^\infty d\omega \, g_{ii,\mu}(\omega) \, \frac{1}{\omega} coth \left(\frac{\hbar\omega}{2k_B T} \right)$$

PHONON-I

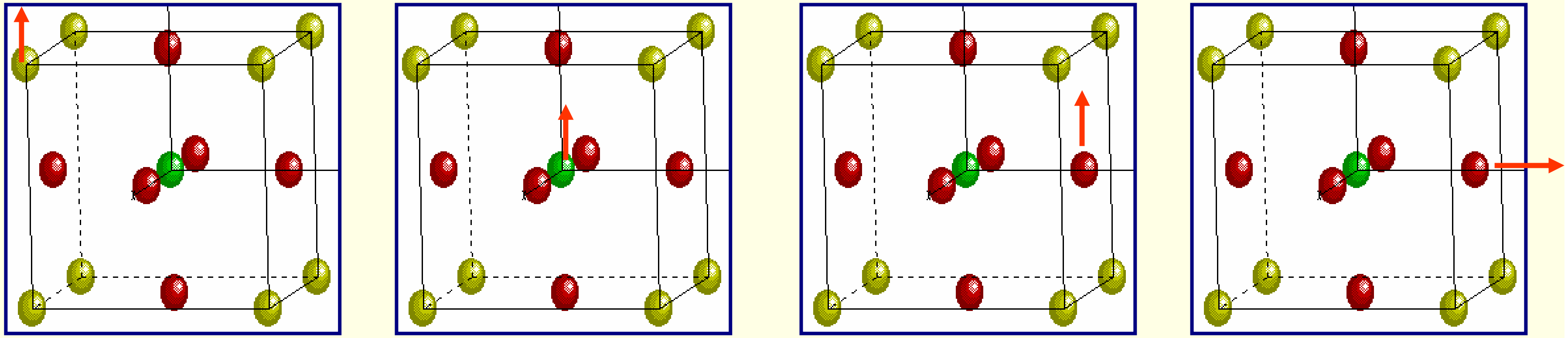
- PHONON
  - *by K.Parlinski (Crakow)*
  - *runs under MS-windows*
  - *uses a „direct“ method to calculate Force-constants with the help of an ab initio program*
  - *with these Force-constants phonons at arbitrary k-points can be obtained*
- Define your spacegroup
- Define all atoms



http://wolf.ifj.edu.pl/phonon/

Phonons:

- *selects symmetry adapted atomic displacements (4 displacements in cubic perovskites)*



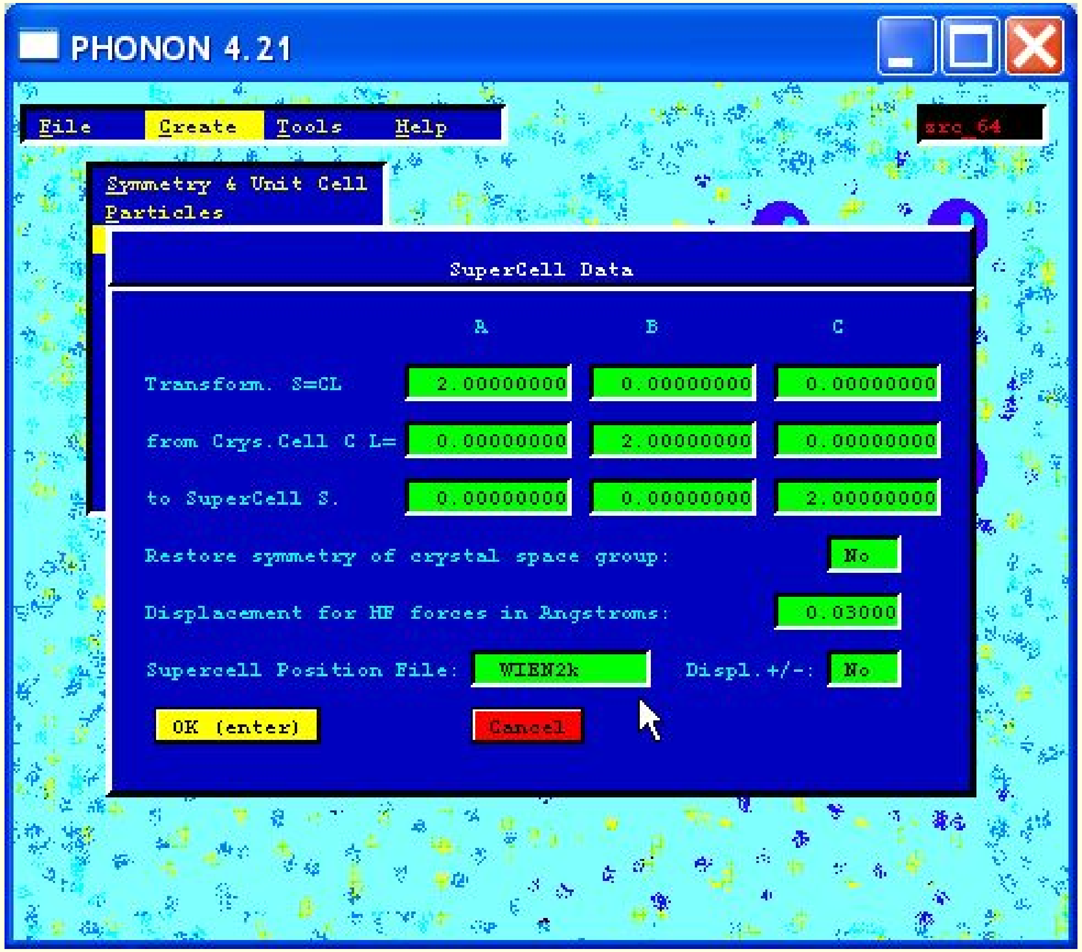
(Displacement pattern for cubic perovskite)

- *select a supercell: (eg. 2x2x2 atom P-type cell)*
- *calculate all forces for these displacements with high accuracy(WIEN2k)*

- *→ force constants between all atoms in the supercell*
- *→ dynamical matrix for arbitrary q-vectors*
- *→ phonon-dispersion ("bandstructure") using PHONON (K.Parlinski)*

PHONON-II

- Define an interaction range (supercell)
  - *create displacement file*
  - *transfer case.d45 to Unix*
- Calculate forces for all required displacements
  - *init_phonon_lapw*
    - for each displacement a case_XX.struct file is generated in an extra directory
    - runs nn and lets you define RMT values like:
      - 1.85 1-16



SuperCell Data

	A	B	C
Transform. S=CL	2.00000000	0.00000000	0.00000000
from Crys.Cell C L=	0.00000000	2.00000000	0.00000000
to SuperCell S.	0.00000000	0.00000000	2.00000000

Restore symmetry of crystal space group: No

Displacement for HF forces in Angstroms: 0.03000

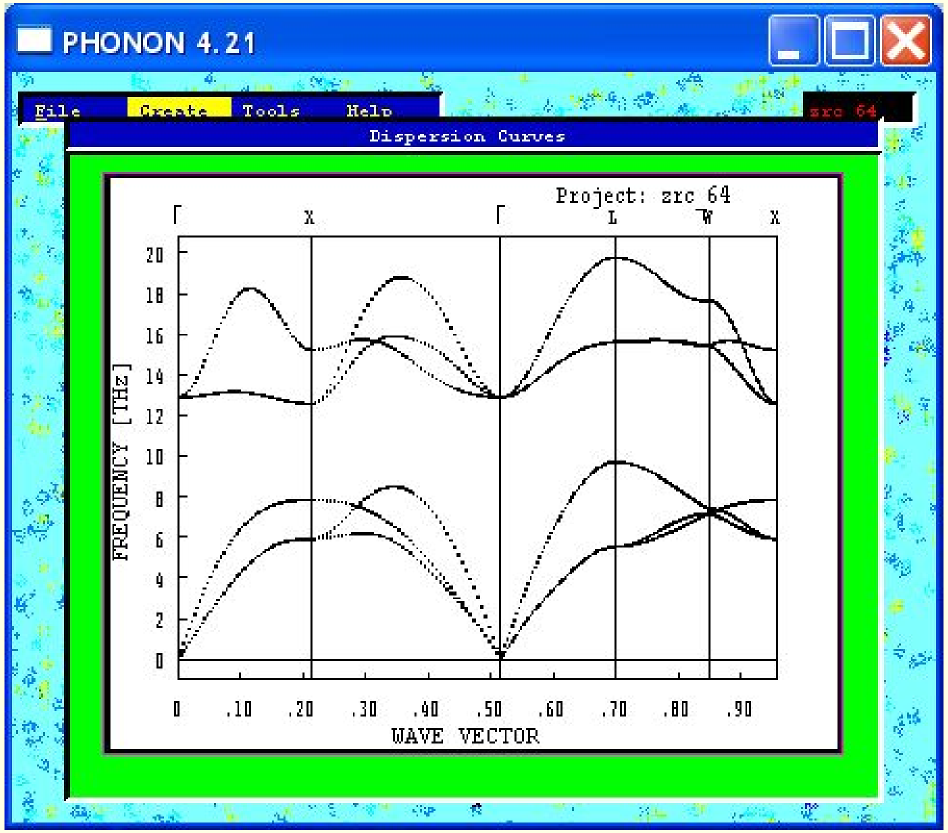
Supercell Position File: WIEN2k Displ.+/-: No

OK (enter) Cancel

- init_lapw: either without symmetry (and then copies this setup to all case_XX)
  or with symmetry (must run init_lapw for all case_XX) (Do NOT use SGROUP)
- run_phonon: run_lapw –fc 0.1 –i 40   for each case_XX

PHONON-III

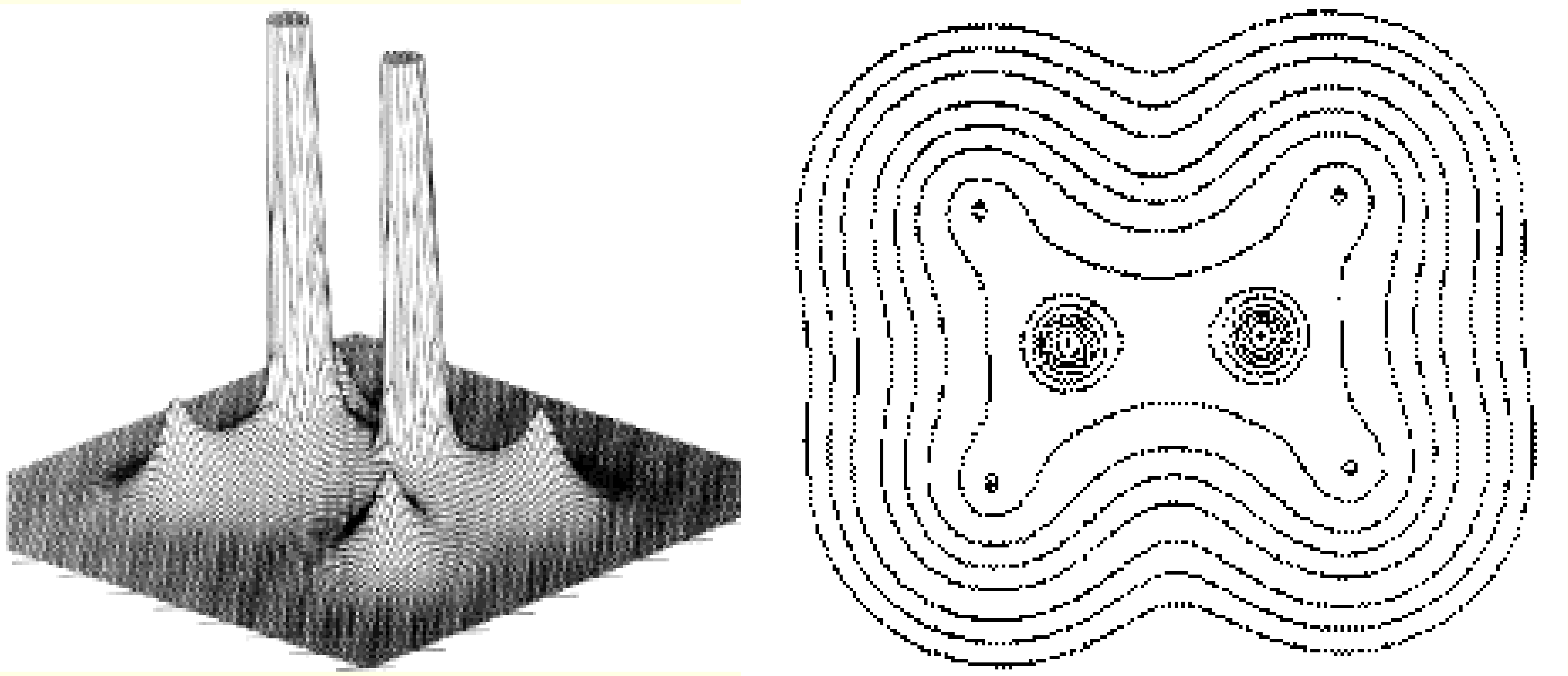
- analyze_phonon_lapw
  - *reads the forces of the scf runs*
  - *generates „Hellman-Feynman“ file case.dat and a „symmetrized HF-file case.dsy (when you have displacements in both directions)*
    - check quality of forces:
    - sum F_x should be small (0)
    - abs(F_x) should be similar for +/- displacements
- transfer case.dat (dsy) to Windows
- Import HF files to PHONON
- Calculate force constants
- Calculate phonons, analyze phonons eigenmodes, thermodynamic functions



Atoms in Molecules

- Theory to characterize atoms and chemical bonds from the topology of the electron density, by R.F.Bader (http://www.chemistry.mcmaster.ca/faculty/bader/aim/aim_0.html)

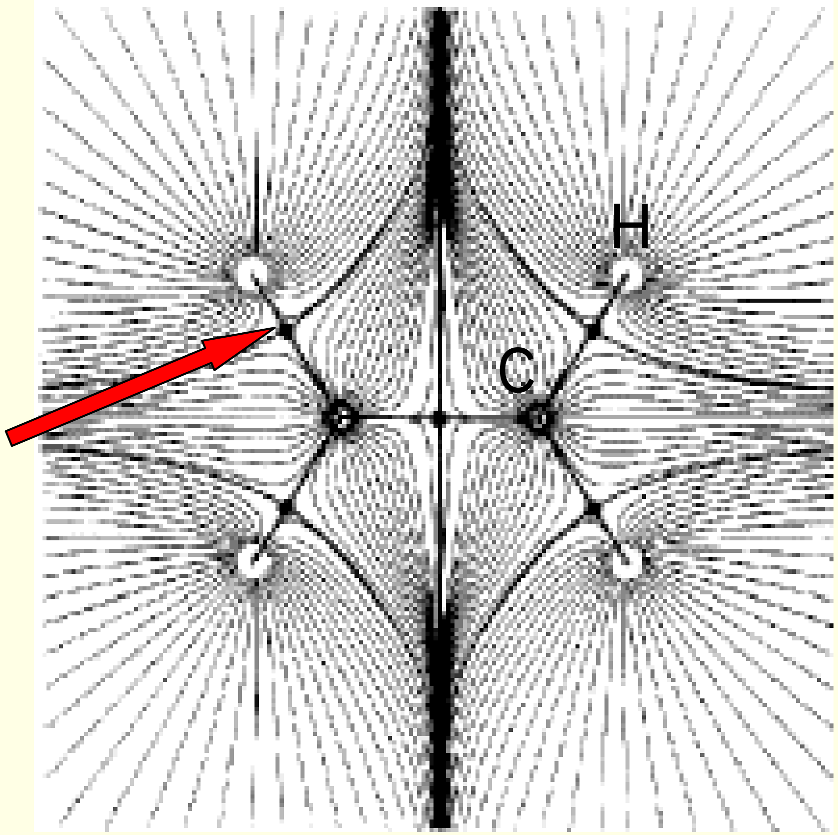
Electron density of C_2H_4



AIM-II

- Bonds are characterized by „critical points“, where $\nabla\rho = 0$
  - •density maximum: (3,-3); 3 negative curvatures λ, (nuclear max, N-NM)
  - •bond CP: (3,-1): 2 negative, 1 positive λ (saddle point)
    - •positive (and large) Laplacian: ionic bond
    - •negative Laplacian: covalent bond
  - •bridge CP: (3,1)
  - •cage CP: (3,3) (minimum)

(3,-1) BCP

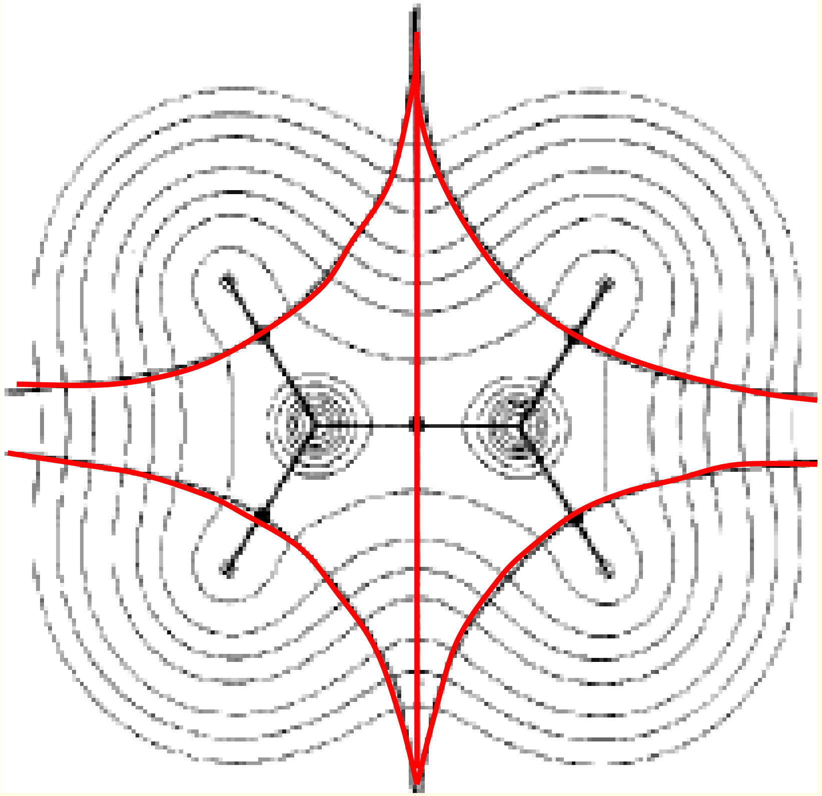


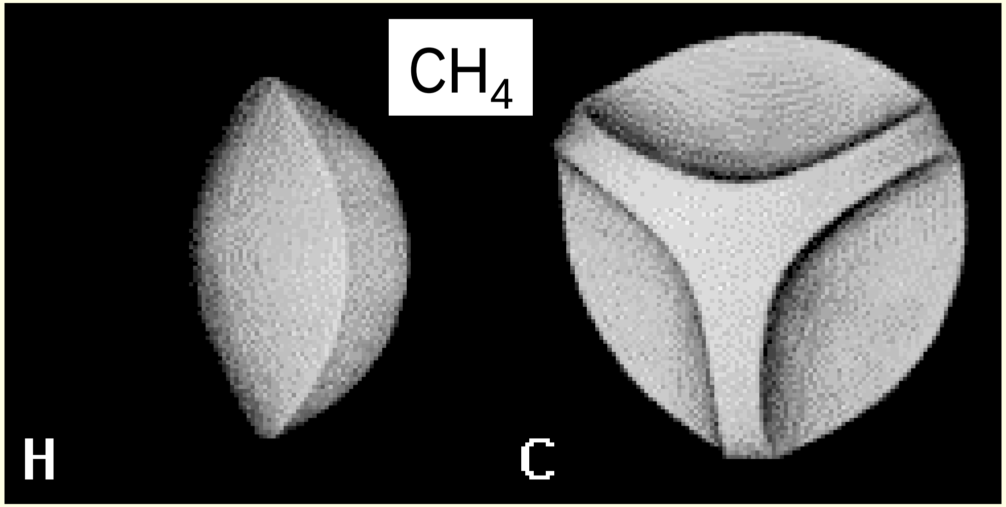
trajectories of constant $\nabla\rho$
originating at CPs in C_2H_4

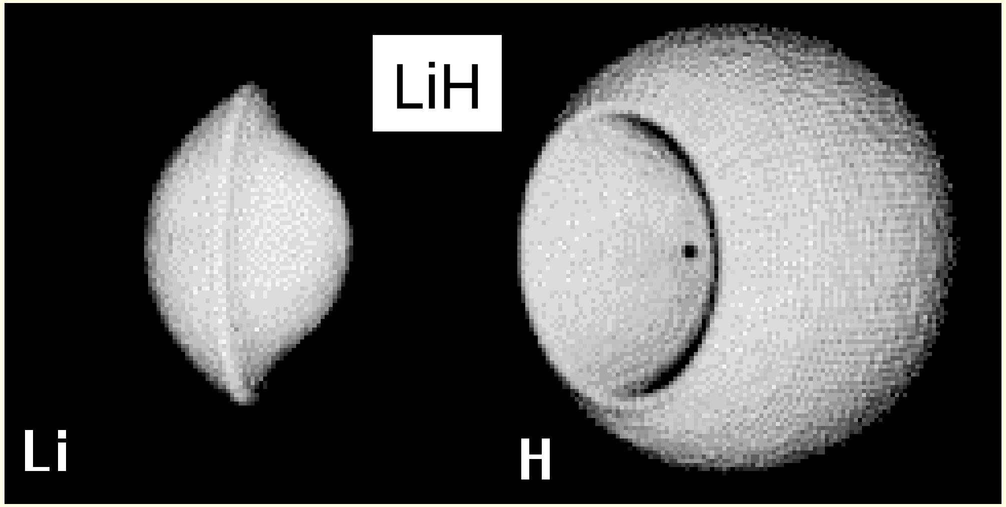
AIM-III

- "Atoms" are regions within a zero-flux surface $\vec{\nabla}\rho \cdot \vec{n} = 0$

ρ of C_2H_4 with zero-flux lines defining atomic basins







AIM-IV

- Bader analysis of some inorganic compounds:

	$\rho(e/A^3)$	$\Delta\rho(e/A^5)$	Q (e)
Cl_2	1.12	-6.1	-
I_2	0.48	-0.9	-
TiC	0.51	1.8	1.7
TiN	0.47	3.9	1.7
TiO	0.43	5.8	1.5
KCl	0.08	1.2	0.6

Cl_2 more covalent
then I_2

more ionic, but less charge?

less ionic then TiC ?

x aim [-c]

- You must have a “good” scf-density (case.clmsum)
  - *no core leakage, LMs up to L=8-10 in case.in2*

```
SURF
1                       atom in center of surface (including MULT)
20 0.0 1.570796327      theta, 20 points, from zero to pi/2
20 0.0 0.785398163      phi, from 0 to pi/4   (depends on symmetry!!)
0.07 1.0 4              step along gradient line, rmin (has reached an atom)
1.65 0.1                initial R for search, step (a.u)
3 3 3                   nshell
IRHO                    "INTEGRATE" rho
WEIT                    WEIT (surface weights are available in case.surf)
30                      30 radial points outside min(RMIN,RMT)
END
---------------------
CRIT
1                       atom around you search for critical points
ALL                     two, three, four, all    (dimers,trimers,....all=2+3)
3 3 3                   nshell
END
```

extractaim_lapw: ➔ critical_points_ang (converted units)

:PC x, y, z, λ_1, λ_2, λ_3, ch, laplacian, rho

WIEN2k- hardware/software

- WIEN2k runs on any Unix/Linux platform from PCs, workstations, clusters to supercomputers
  - *Pentium-IV with fast memory bus (1-2 Gb memory, 100Mbit net, IDE disks)*
  - *10 atom cells on 128Mb PC / 100 atom cells require 1-2 Gb RAM*
  - *installation support for most platforms*

- Fortran90 (dynamical allocation, modules)
  - *real/complex version (inversion)*
  - *many individual modules, linked together with C-shell or perl-scripts*

- *web-based GUI – w2web (perl)*

- f90 compiler, BLAS-library (ifort9+mkl), perl5, ghostscript (+jpg), gnuplot(+png), Tcl/Tk (Xcrysden), pdf-reader, www-browser, octave, opendx

Installation of WIEN2k

- Register via http://www.wien2k.at
- Create your $WIENROOT directory (e.g. ./WIEN2k)
- Download wien2k_06.tar and examples (executables)
- Uncompress and expand all files using:
  - *tar –xvf wien2k_06.tar*
  - *gunzip *.gz*
  - *chmod +x ./expand_lapw*
  - *./expand_lapw*
- This leads to the following directories:

./SRC	*(scripts, ug.ps)*
./SRC_aim	*(programs)*
...	
SRC_templates	*(example inputs)*
...	
SRC_usersguide_html	*(HTML-version of UG)*
example_struct_files	*(examples)*
TiC	

siteconfig_lapw

```
   ****************************************************
   *                      W I E N                     *
   *                 site configuration               *
   ****************************************************
      S    specify a system
      C    specify compiler
      O    specify compiler options, BLAS and LAPACK
      P    configure Parallel execution
      D    Dimension Parameters
      R    Compile/Recompile
      U    Update a package
      L    Perl path (if not in /usr/bin/perl)
      Q    Quit
```

D: define NMATMAX (adjust to your hardware/paging!):
NMATMAX=5000 ➔256Mb (real) or 500Mb (complex)
NMATMAX=10000 ➔ 1Gb (real) ➔ 80-150 atoms/unitcell

Always use „optimized“ BLAS library (ifort+mkl; ATLAS-BLAS, gotolib)

Compilation

- most common: Intels Fortran compiler (free for academic)
  - *ifc (7.1)*
    - -static linking
    - patch for 2 externals, see faq-web page or mailing-list digest
  - *ifort 9.0*
    - 8.0 buggy ?,
    - needs unlimited “stacksize” (limit stacksize=unlimited, otherwise: “segmentation fault”)
    - dynamic linking necessary (requires system and compiler libraries at runtime, may need $LD_LIBRARY_PATH)

  - *mkl 6.1 or higher; ATLAS-BLAS;* ***gotolib***
    - -L/opt/intel/mkl/lib –lmkl_em64t → libmkl_em64t.so
  - *compiler/linker options depend on compiler version + Linux-version !!*

userconfig_lapw

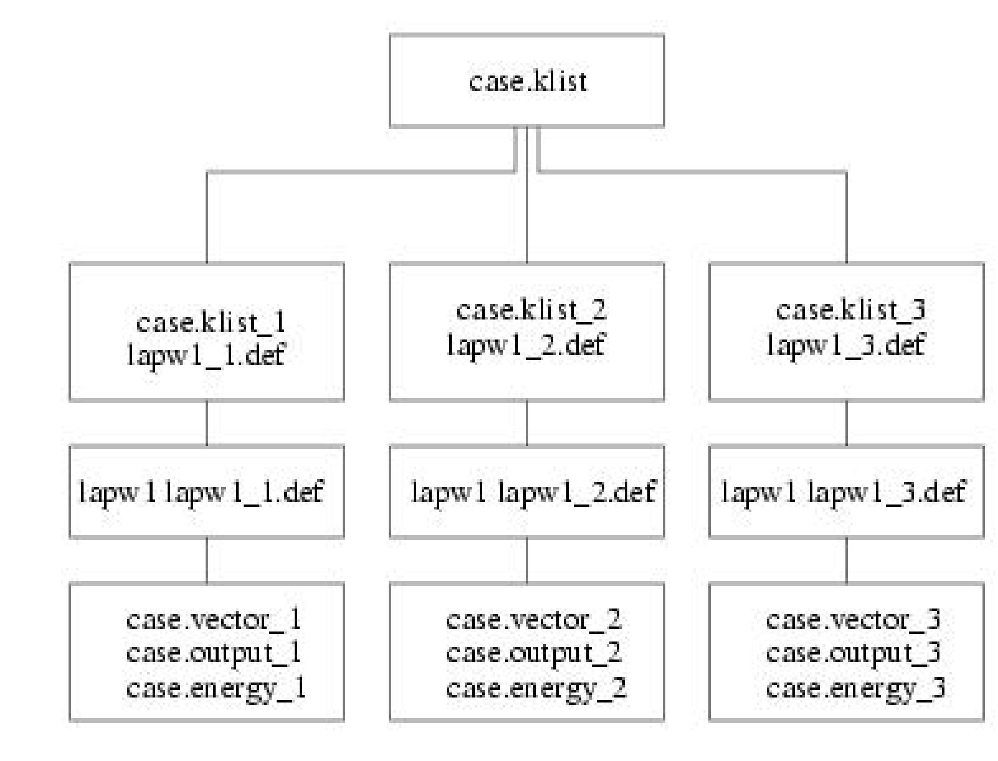
- Every user should run userconfig_lapw
  - *support for tcsh and bash*
  - *sets PATH to $WIENROOT, sets variables and aliases*
    - $WIENROOT, $SCRATCH, $EDITOR, $PDFREADER, $STRUCTEDIT_PATH
    - pslapw: ps –ef | grep lapw
    - lsi: ls –als *.in*
    - lso: *.output*
    - lss: *.scf*
    - lsc: *.clm*
    - limit stacksize unlimited
    - $OMP_NUM_THREADS (for mkl+dual core); $LD_LIBRARY_PATH
- w2web: acts as webserver on a userdefined (high) port.
  - *define „master“ and „slave“ nodes. (master knows all „projects“)*
  - *define user/password and port. (http://host.domain.xx:5000)*
  - *~/.w2web/hostname/conf/w2web.conf: (configuration file)*
    - deny=*.*.*.*
    - allow=128.130.134.* 128.130.142.10
    - define execution types: NAME=commands (eg.: batch=batch < %f)

Parallelization

- k-point parallel on clusters (slow network): lapw1+lapw2
  - *common NFS filesystem (files must be accessable with the same path on all machines)*
  - *rsh/ssh without password (.rhosts; private/public keys)*
    - .machines file:
      - 1:host1 (speed:hostname)
      - 2:host2
      - granularity:1 (1:10k+10k; 3: 3+3+3+3+3+3+rest ➔load balancing)
      - extrafine:1 (rest in junks of 1 k)
    - testpara (tests distribution); run_lapw -p
- fine-grain parallelization for big cases (>50 atoms) and fast network (Myrinet, Infiniband, shared memory machines)
  - *mpi + scalapack*
    - .machines file:
      - 1:host1:4 host2:4 8 mpi-parallel jobs on host1 and host2
      - lapw0:host1:4 host2:4 8 parallel jobs; atom-loops only!!!

Flow of parallel execution

lapw1para



case.klist

case.klist_1
lapw1_1.def

case.klist_2
lapw1_2.def

case.klist_3
lapw1_3.def

lapw1 lapw1_1.def

lapw1 lapw1_2.def

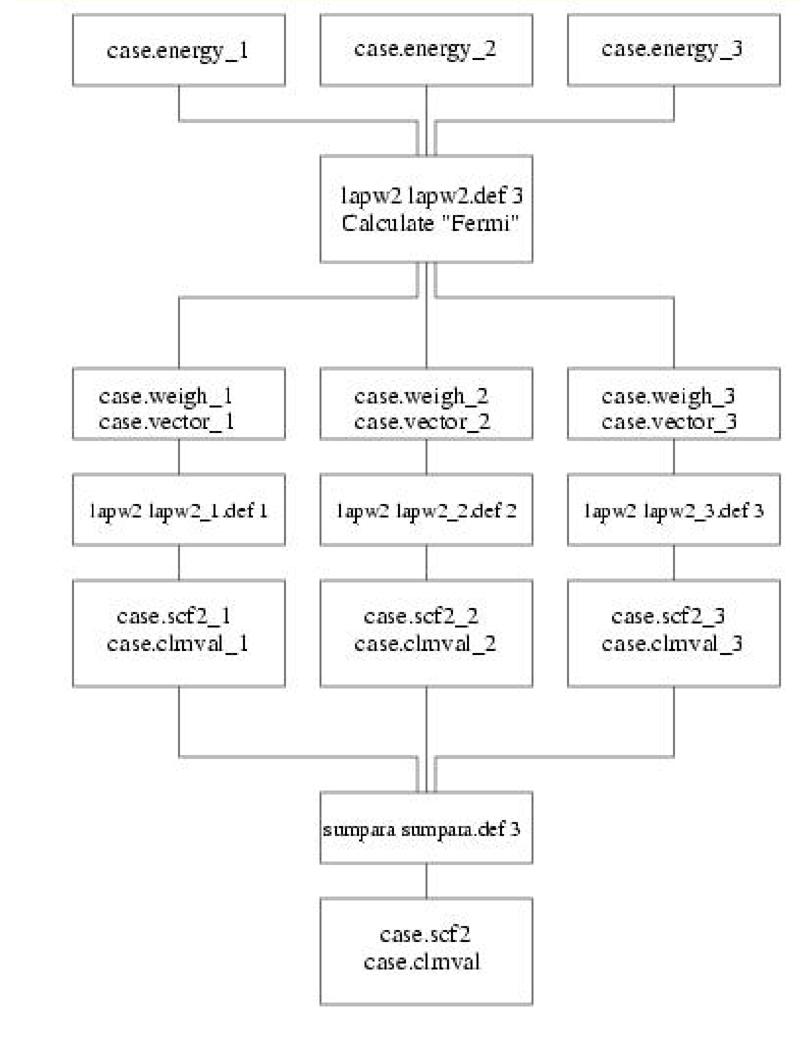
lapw1 lapw1_3.def

case.vector_1
case.output_1
case.energy_1

case.vector_2
case.output_2
case.energy_2

case.vector_3
case.output_3
case.energy_3

lapw2para



case.energy_1

case.energy_2

case.energy_3

lapw2 lapw2.def 3
Calculate "Fermi"

case.weigh_1
case.vector_1

case.weigh_2
case.vector_2

case.weigh_3
case.vector_3

lapw2 lapw2_1.def 1

lapw2 lapw2_2.def 2

lapw2 lapw2_3.def 3

case.scf2_1
case.clmval_1

case.scf2_2
case.clmval_2

case.scf2_3
case.clmval_3

sumpara sumpara.def 3

case.scf2
case.clmval