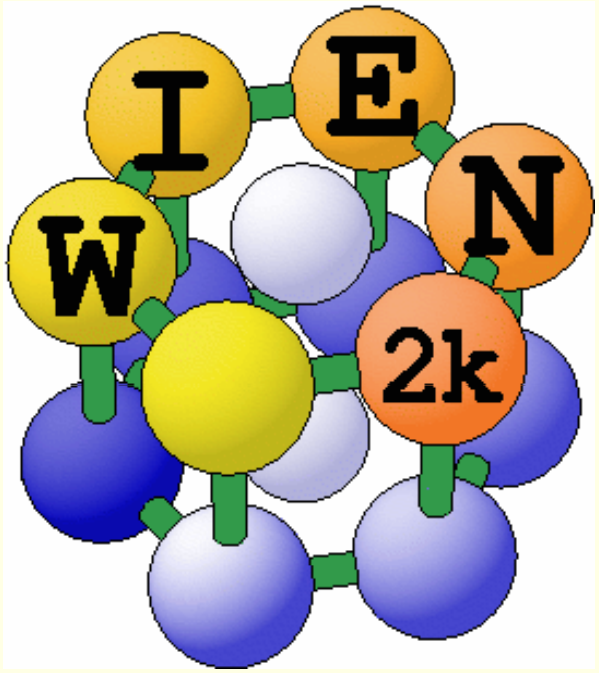
WIEN2k software package



WIEN97: ~500 users
WIEN2k: ~1250 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

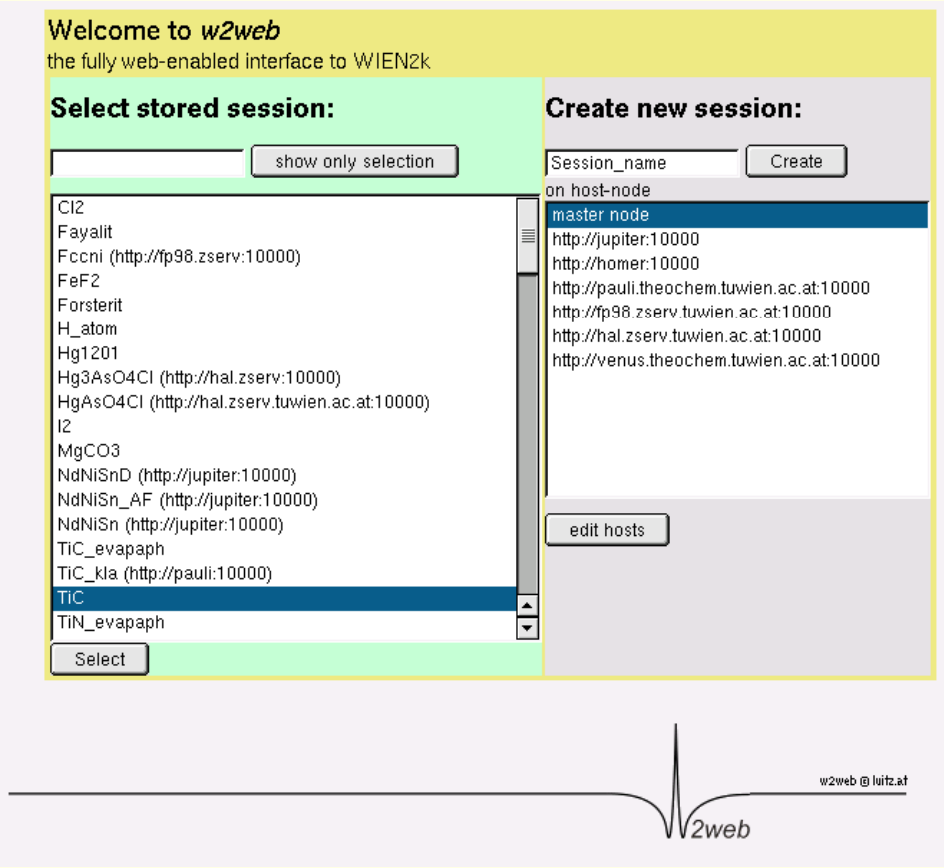
http://www.wien2k.at

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 in 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

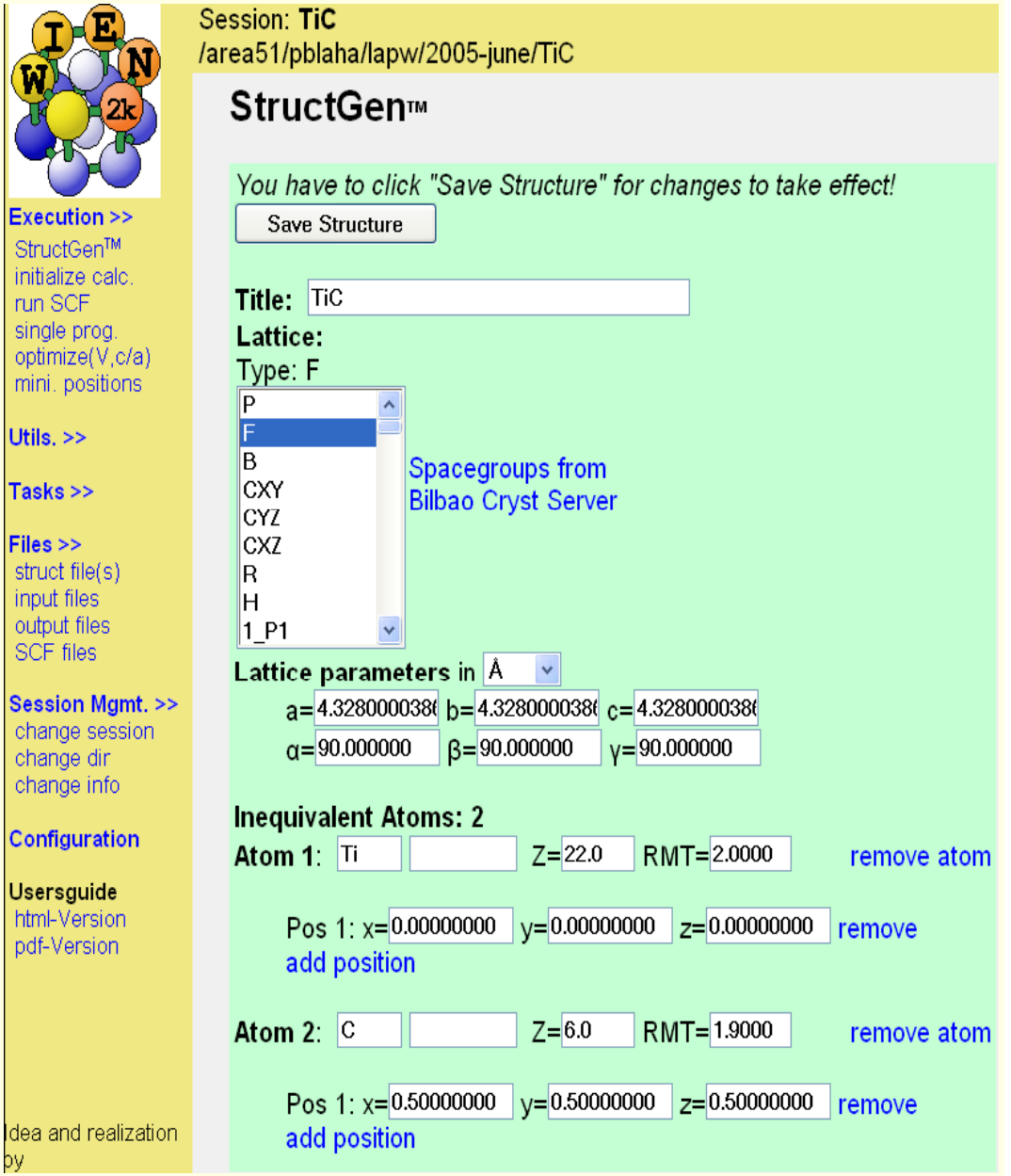
w2web: the web-based GUI of WIEN2k

- 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*
    - firefox http://fp98.zserv:10000
  - *create a new session on the desired host (or select an old one)*



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*



Spacegroup $P4_2/mnm$

Structure given by:

spacegroup

lattice parameter

positions of atoms (basis)

Rutile TiO_2:

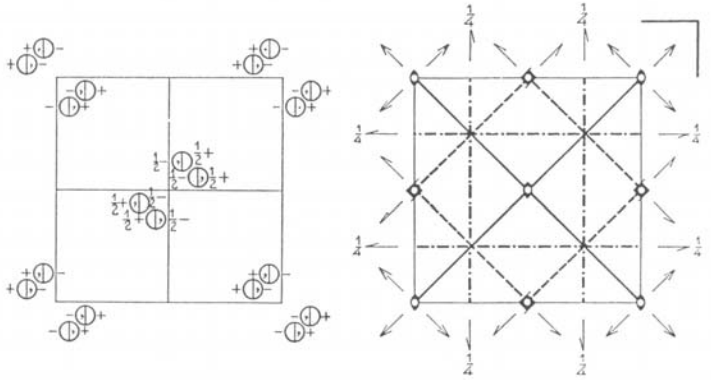
$P4_2/mnm$ (136)

a=8.68, c=5.59 bohr

Ti: (0,0,0)

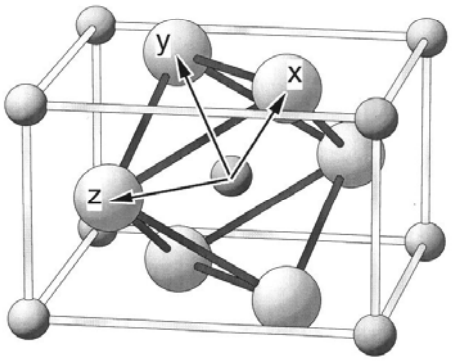
O: (0.304,0.304,0)

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



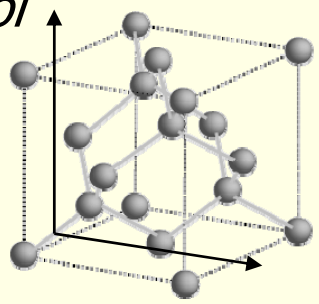
Origin at centre (*mmm*)

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

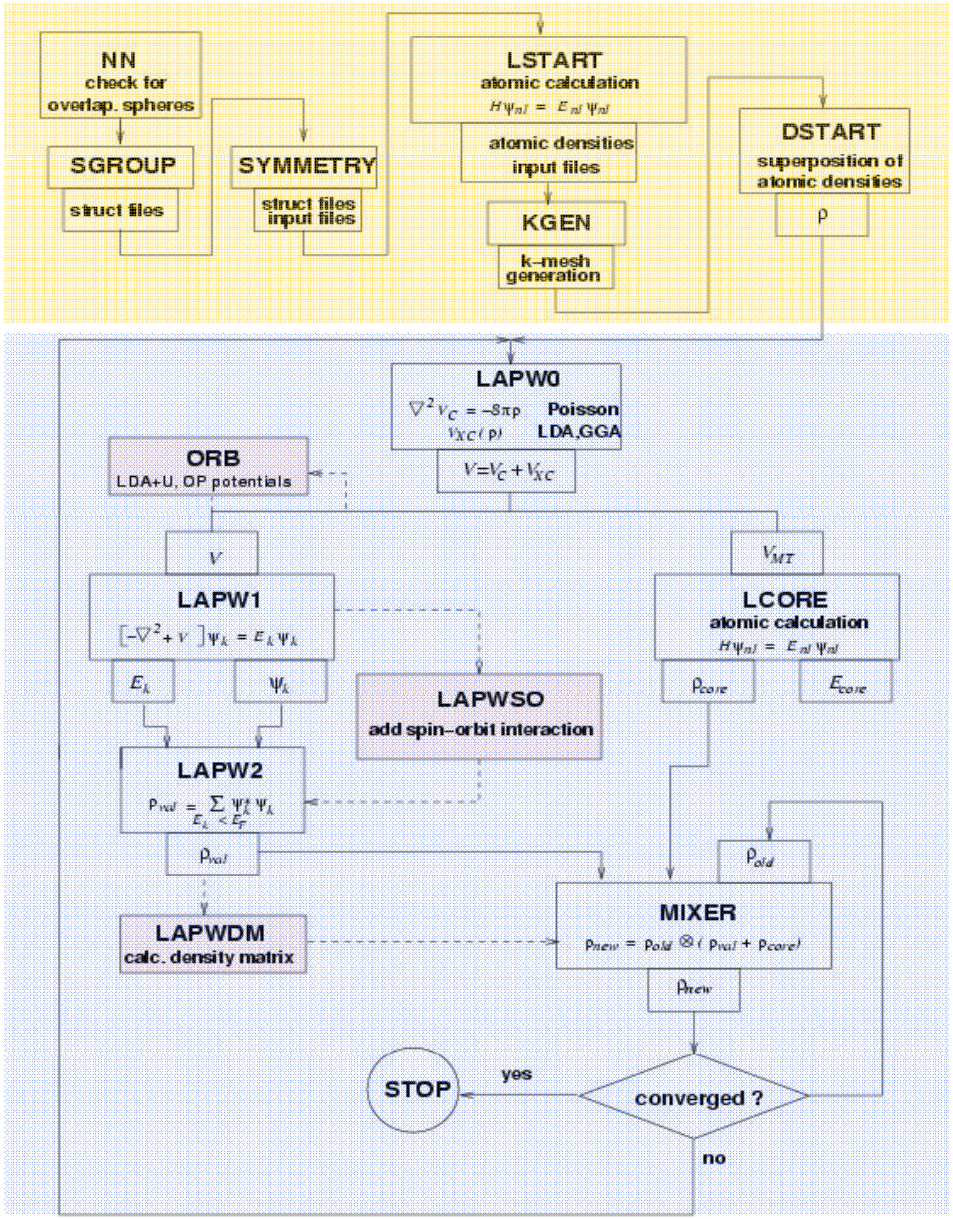
- Specify:
  - *Number of nonequivalent atoms*
  - *lattice type (P, F, B, H, CXY, CXZ, CYZ) or spacegroup symbol*
    - if existing, you must use a SG-setting with inversion symmetry:
      - Si: ±(1/8,1/8,1/8), not (0,0,0)+(1/4,1/4,1/4)!
  - *lattice parameters a,b,c (in Å or bohr)*
  - *name of atoms (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 (d) - 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, RMT equal for same atoms*
- „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)*
-it /-it0	*iterative diagonalization (large speedup)*
-p	*parallel calculation (needs .machines file)*
-so	*add spin-orbit (only after „init_so")*
Spacegroups without inversion use automatically lapw1c, lapw2c (case.in1c,in2c)	

- 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

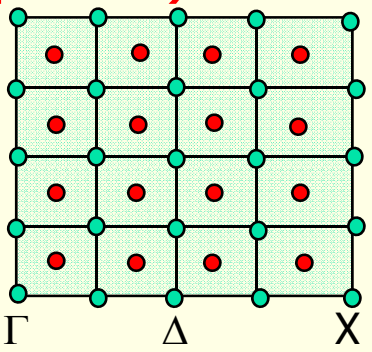
  - :LAT :VOL :POSxxx

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.

most common problems

- scf-cycle diverges (grep :DIS case.scf):
  - *check structure;*
  - *reduce mixing in case.inm; rm *.broyd* case.scf; x dstart*
- „QTL-B" value too large - STOP
  - *identify for which eigenvalue, atom and ℓ it happens, check EF (case.scf2, case.output2)*
  - *identify the corresponding linearization energies in case.scf1*
  - *change the corresponding linearization energy in case.in1*
    - compare and check with :EPL and :EPH lines in case.scf2
    - default E-parameters may need changes for
      - surfaces (EF often negative) or heavy elements (EF often larger than 1.0)
    - use „–in1new 1" switch
  - *if QTL-B occurs for an atom with large RMT, reduce RMT*
    - this may happen for larger RKMAX („numerical linear dependency")

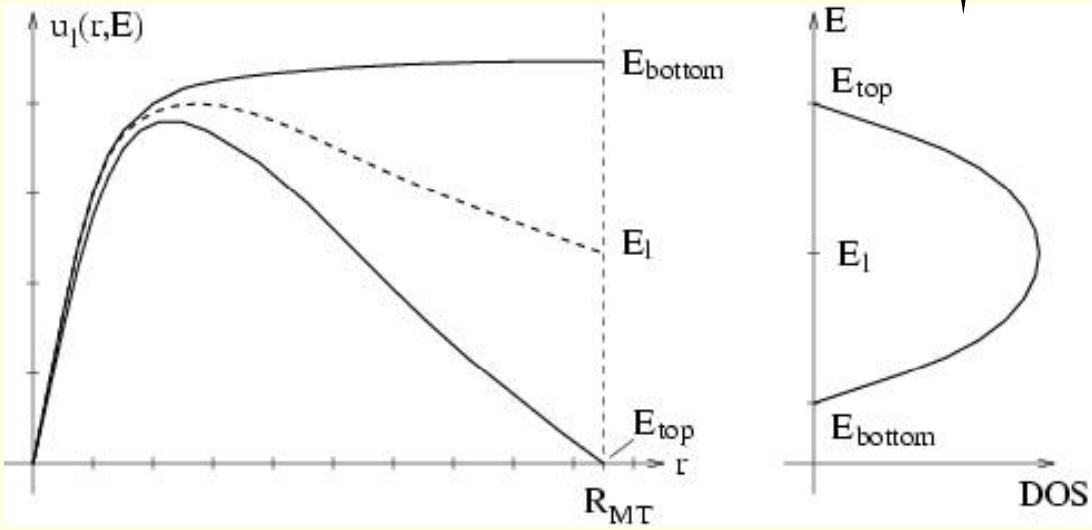
case.in1

```
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
```

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

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

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



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
```

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

run_lapw -ql 0.05 -in1new 1

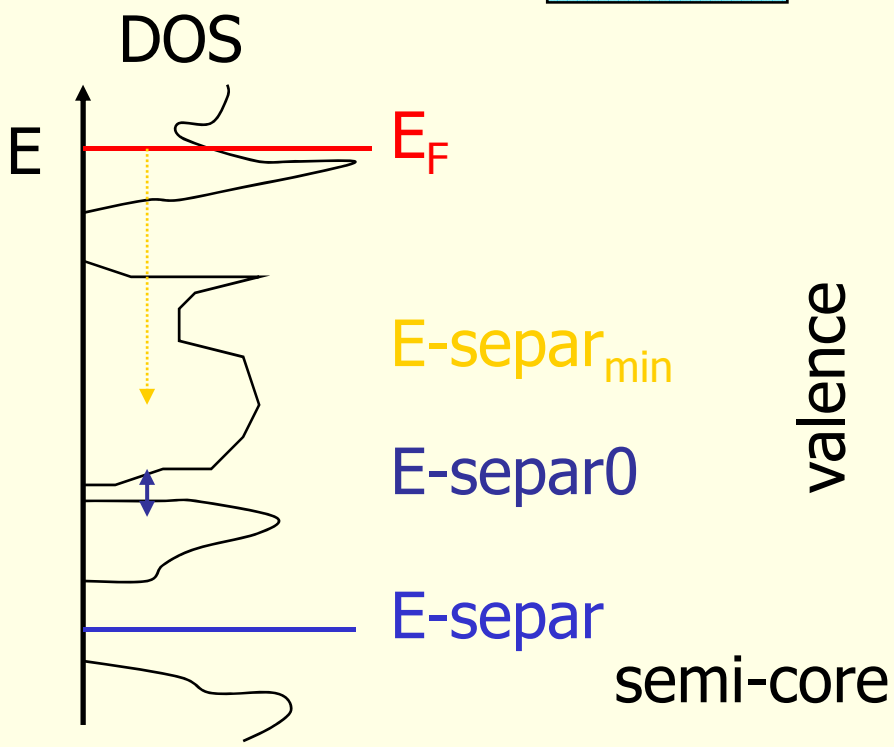
- Alternative case.in1 file produced by write_in1:

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
 ...
```



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*

DOS

- be sure to have eigenvectors on a dense tetrahedral mesh after a scf calculation
  - *eventually:*
    - x kgen
    - edit case.in1 (larger Emax)
    - x lapw1
- case.outputt
  - *integrated DOS*
- case.dos1ev (3ev)
  - *text-file for plotting*
  - *E-zero at E_F*

Session: **TiC**
/susi/pblaha/lapw/TiC

Density of states

x lapw2 -qtl	Calculate partial charges ☑ interactively
edit TiC.int	Edit input-file for TETRA
x tetra	Calculate partial DOS ☑ interactively
edit TiC.outputt	Check output of TETRA
dosplot	Plot DOS

Session: **TiC**
/susi/pblaha/lapw/TiC

File:
/susi/pblaha/lapw/TiC/TiC.int

contine with DOS Save Download this file:

Header from TiC.qtl:

```
 ATOM   1  tot,0,1,2,3,xdos(i,j),j=1,i),i=1,lxdos2)
 ATOM   2  tot,0,1,2,D-eg,D-t2g,3
```

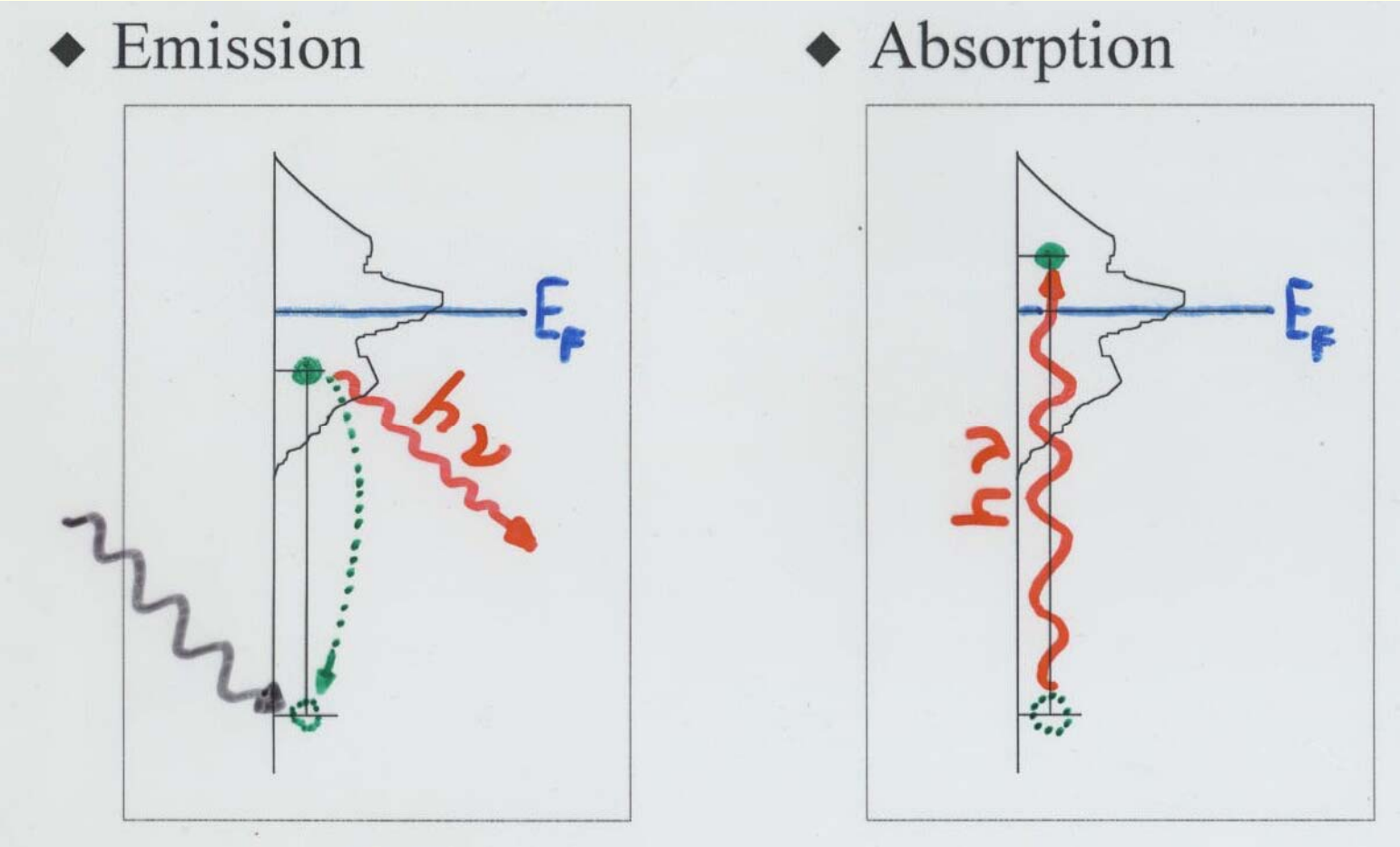
```
Title
 -0.50 0.002 1.500 0.003   EMIN, DE, EMAX, Gauss-broadening(>;de)
    3                      NUMBER OF DOS-CASES specified below
    0    1   total         atom, case=column in qtl-header, label
    1    2   Atom1-s
    2    5   Atom2-eg
```

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
    - interface by G.Madsen to PHON (D.Alfe)
      - http://www.chem.au.dk/~webuorg/new/groups/gm/gm.html
- 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 (using XcrysDen)*

X-ray emission and absorption spectra

XES **XAS**

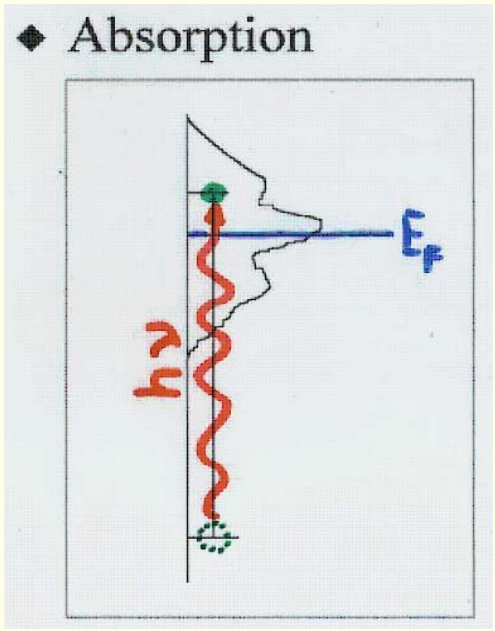


conduction bands

valence bands

core state

XAS: X-ray absorption spectra



dipole operator

$$I \propto \nu^3 \left|\langle \Psi_{val} \left| r \right| \Psi_{core} \rangle\right|^2 \chi_l^A(\epsilon)$$

ℓ-like PDOS (atom A)

$$\frac{I_{n'l'}}{\nu^3} = \sum_l W_{ll'} M_A^2(l, n'l', \epsilon) \chi_l^A(\epsilon) \delta(\epsilon - E_{core}, h\nu)$$

energy conservation

$\Delta\ell = \pm 1$

dipole-section rule

Radial transition probability:

- from a core state with q.n. n´ℓ´on atom A
- to an ℓ-like conduction band state inside the atomic sphere A
- within the dipole approximation

Final state effects:
supercell with (partial) core hole on one of the atoms

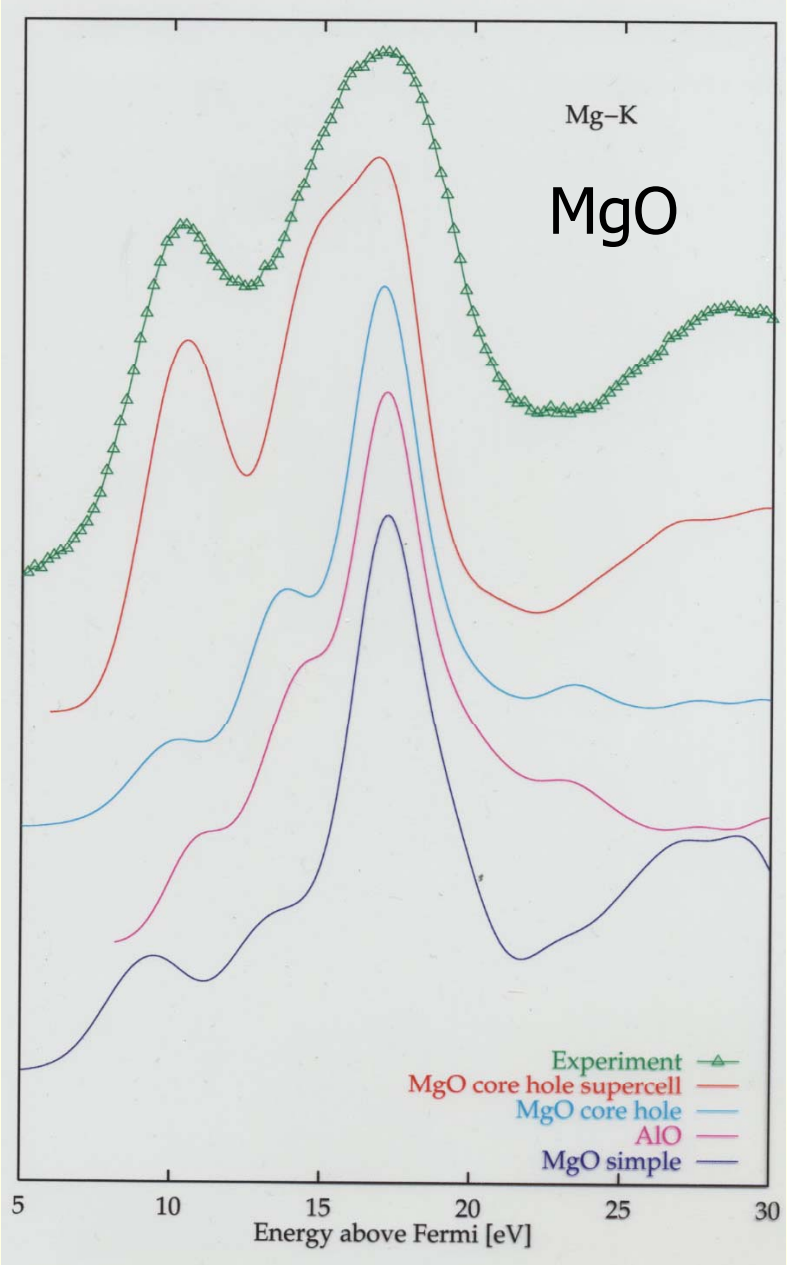
"Final state rule":

"Final state" determines the spectrum:

Emission spectroscopy:
Final state has filled core, but valence hole. This is usually well screened, thus one "sees" the groundstate.

Absorption spectroscopy:
"hole" in core state, but additional e^- in conduction band. Core-hole has large effect on the spectrum (static approximation)

➔ 2x2x2 supercell calculation, with core hole in one of the Mg atoms. This allows the conduction state to relax (adjust to the larger effective nuclear charge), but also to have screening from the environment.

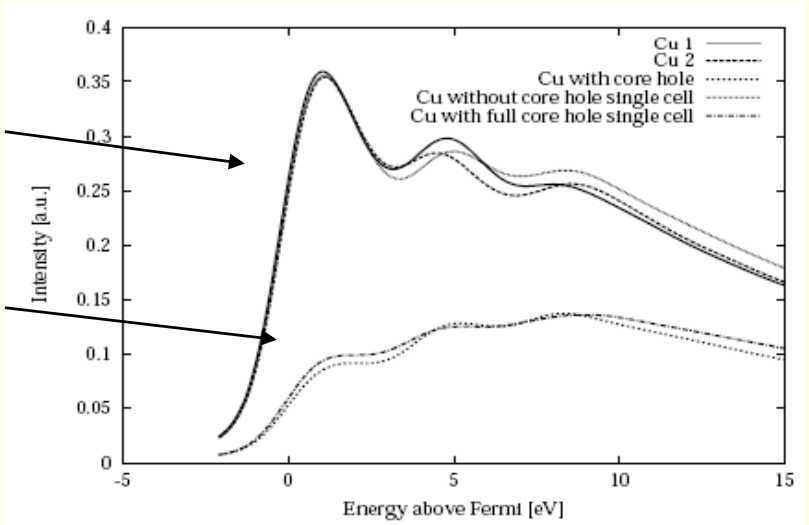


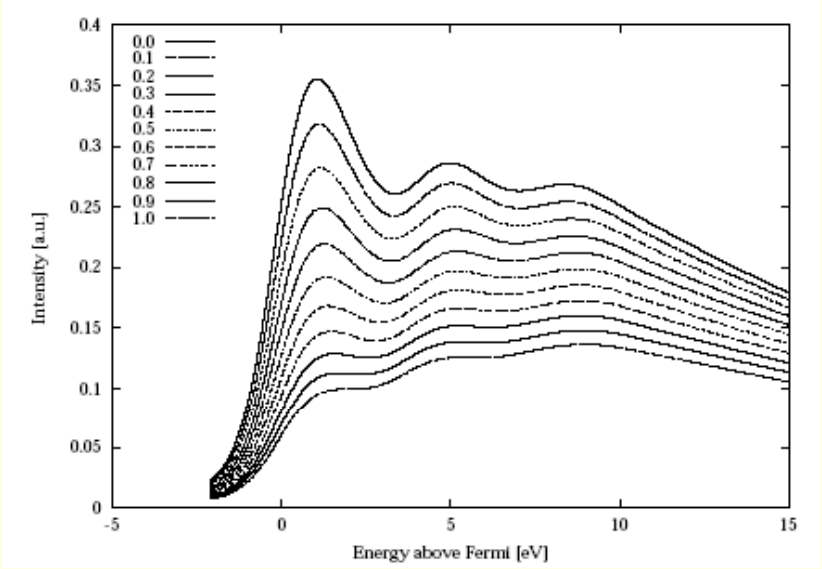
Partial core hole screening in the Cu L3 edge

- J.Luitz et al., Eur. Phys. J. B **21**, 363{367 (2001)

without corehole

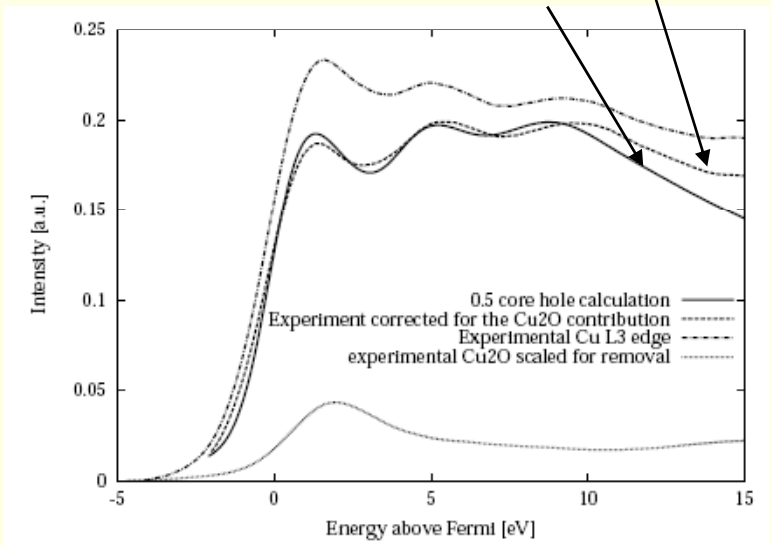
with corehole

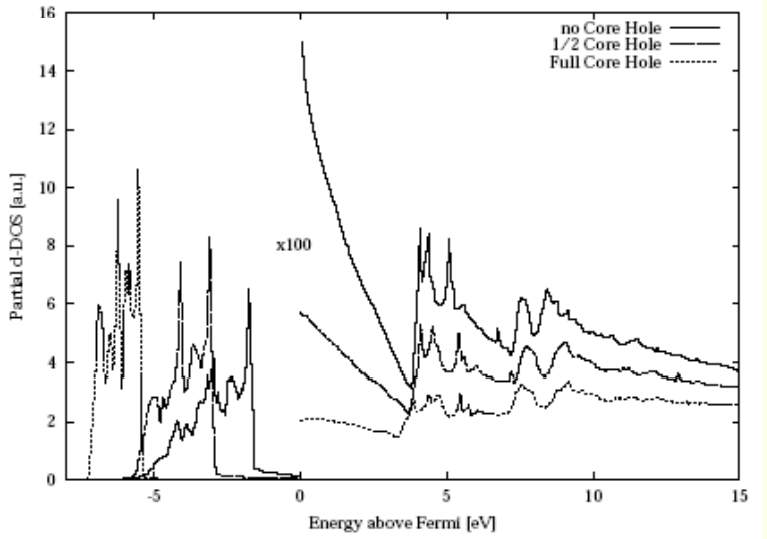




experiment

0.5e corehole





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 (M.Scheffler, FH Berlin)*
  - *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)

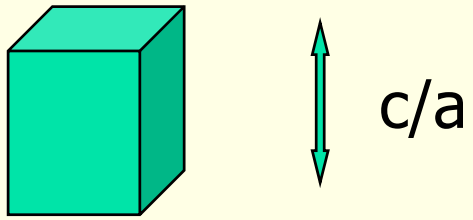
Structural optimizations:

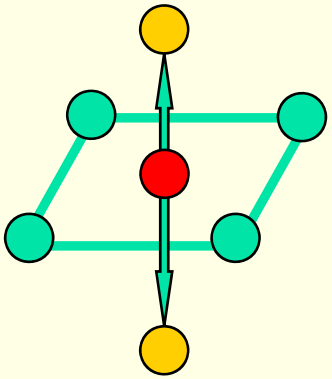
- Lattice parameters, volume, c/a ratio only via total energies:
  - *x optimize: creates a series of "struct" files + script "optimize.job"*
    - select volume or c/a, …
    - select number of cases and desired changes in volume (in % of V_0)
  - *edit optimize.job*
    - adapt to your need: change / 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*

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

  - *Pulay corrections*
    - Core

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

    - Valence

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

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

- expensive, contains a summation of matrix elements over all occupied states

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 "-fc":
  - *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 → changes "TOT" to "FOR" in case.in2
      - -12.3 $F_{HF} + F_{core} + F_{val}$, only this last number is correct
- Forces are useful for
  - *structural optimization (of internal parameters)*
  - *phonons*

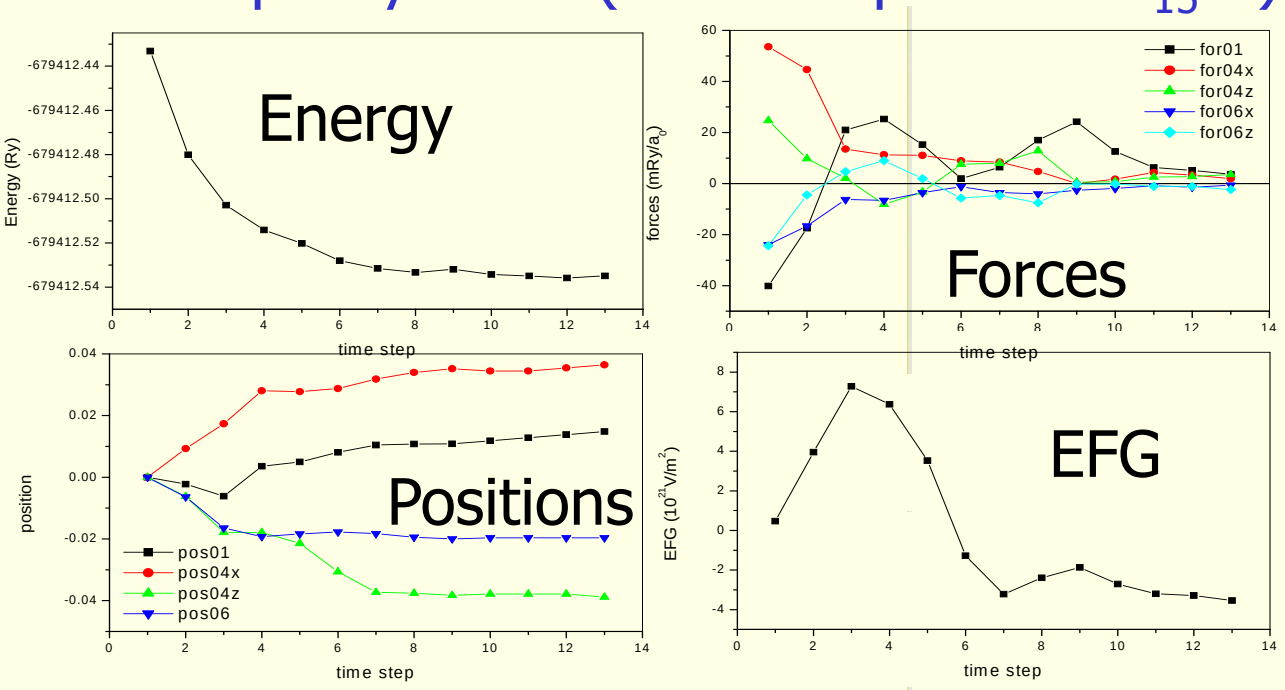
Structural optimization of internal parameters

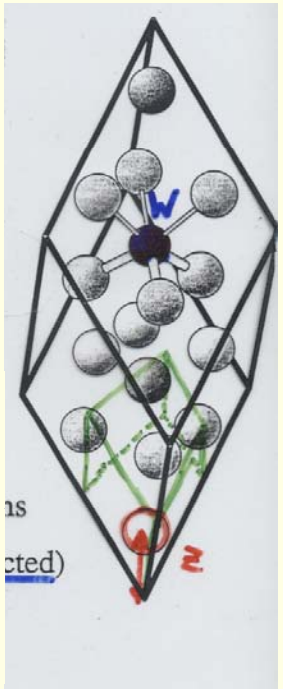
- /home/pblaha/tio2> min_lapw -h
  - *OPTIONS:*
    - -p -> does a k-point parallel calculation
    - -it -> use iterative diagonalization
    - -sp -> does a spin-polarized calculation (runsp_lapw)
    - -NI -> without initialization of input-files (continue after a "crash")
    - -i NUMBER -> max. NUMBER (50) of structure changes
    - -j JOB -> job-file JOB (run_lapw -I -fc 1. -i 40)
  - *CONTROL FILES:*
    - .minstop stop after next structure change
- /home/pblaha/tiO2> cat tio2.inM
    - PORT 2.0 #(NEW1, NOSE, MOLD, tolf (a4,f5.2))
    - 0.0 1.0 1.0 1.0 # Atom1 (0 will constrain a coordinate)
    - 1.0 1.0 1.0 1.0 # Atom2 (NEW1: 1,2,3:delta_i, 4:eta (1=MOLD, damping))
- monitor minimization in file case.scf_mini
  - *contains last iteration of each geometry step*
  - *each step N is saved as case_N.scf (overwritten with next min_lapw !)*

Optimization of atomic posistions (E-minimization via forces)

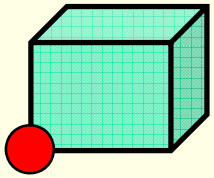
- damped Newton mechanics scheme (NEW1: with variable step)
- quite efficient quasi-Newton (PORT) scheme
  - minimizes E (using forces as gradients)
  - If minimizations gets stuck or oscillates: (because E and F_i are inconsistent):
    - touch .minstop; min –nohess (or rm case.tmpM .min_hess)
    - improve scf-convergence (-ec), Rkmax, k-mesh, …
    - change to NEW1 scheme

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

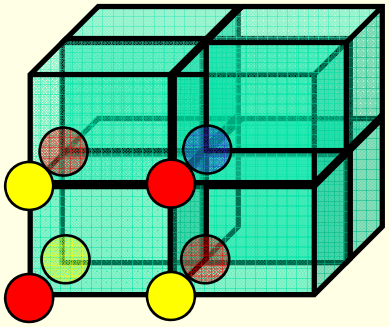




Supercells



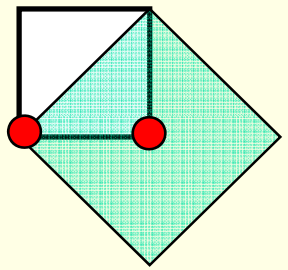
2x2x2 = 8 atoms



(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

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



Supercells

- 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!!!

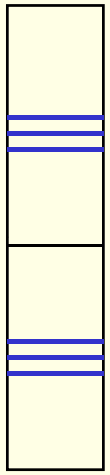
Structeditor (by R.Laskowski)

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

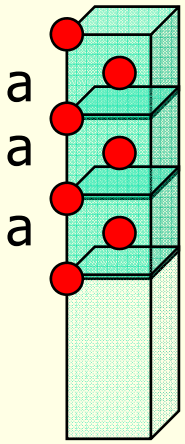
```
octave
s=loadstruct("GaN.struct")
# make an orthorhombic supercell and visualize it
a=[1 0 0; 1 1 0; 0 0 2]
sout=makesupercell (s,a);
showstruct(sout);
# save it as test.struct
savestruct (sout,"test.struct");
# get help on all commands
helpstruct
```

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)

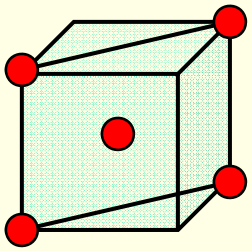
shift to ➔ inversion

(.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

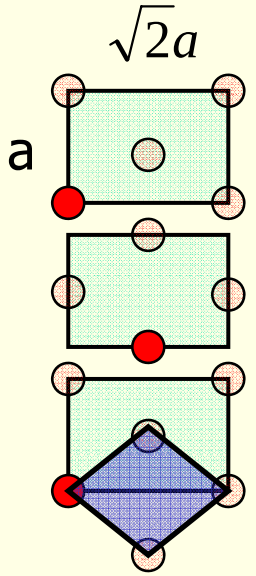
bcc (110):



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

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

z=a/$\sqrt{2}a$ c

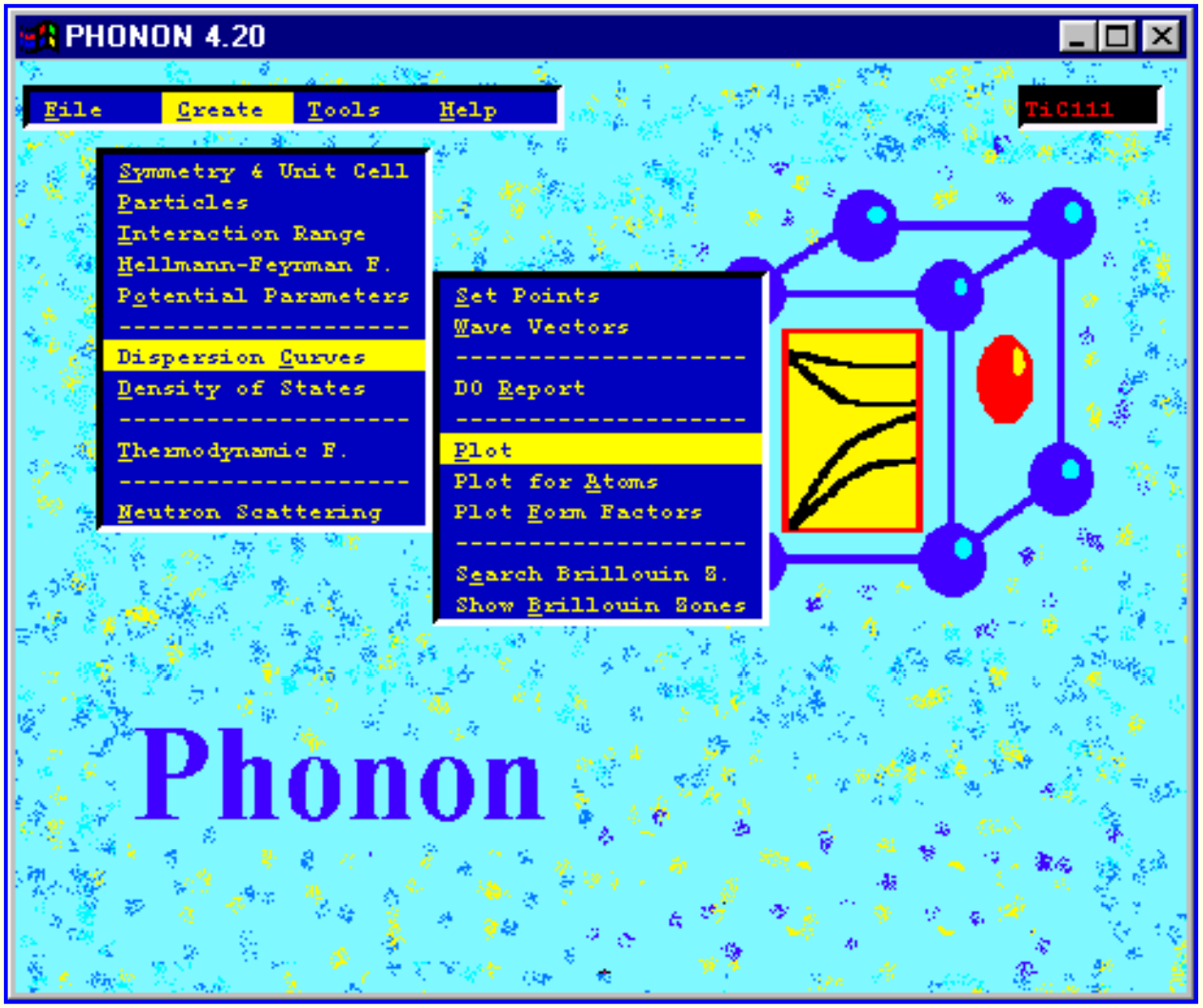


+/-2z

+/-z

z=0

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

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

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 $\frac{\partial E}{\partial \mathbf{R}_i(\mathbf{n},\mu)}\,|_o = 0$.

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

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

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

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

Master equation of direct method.

CUMMULANT FORCE CONSTANTS

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

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

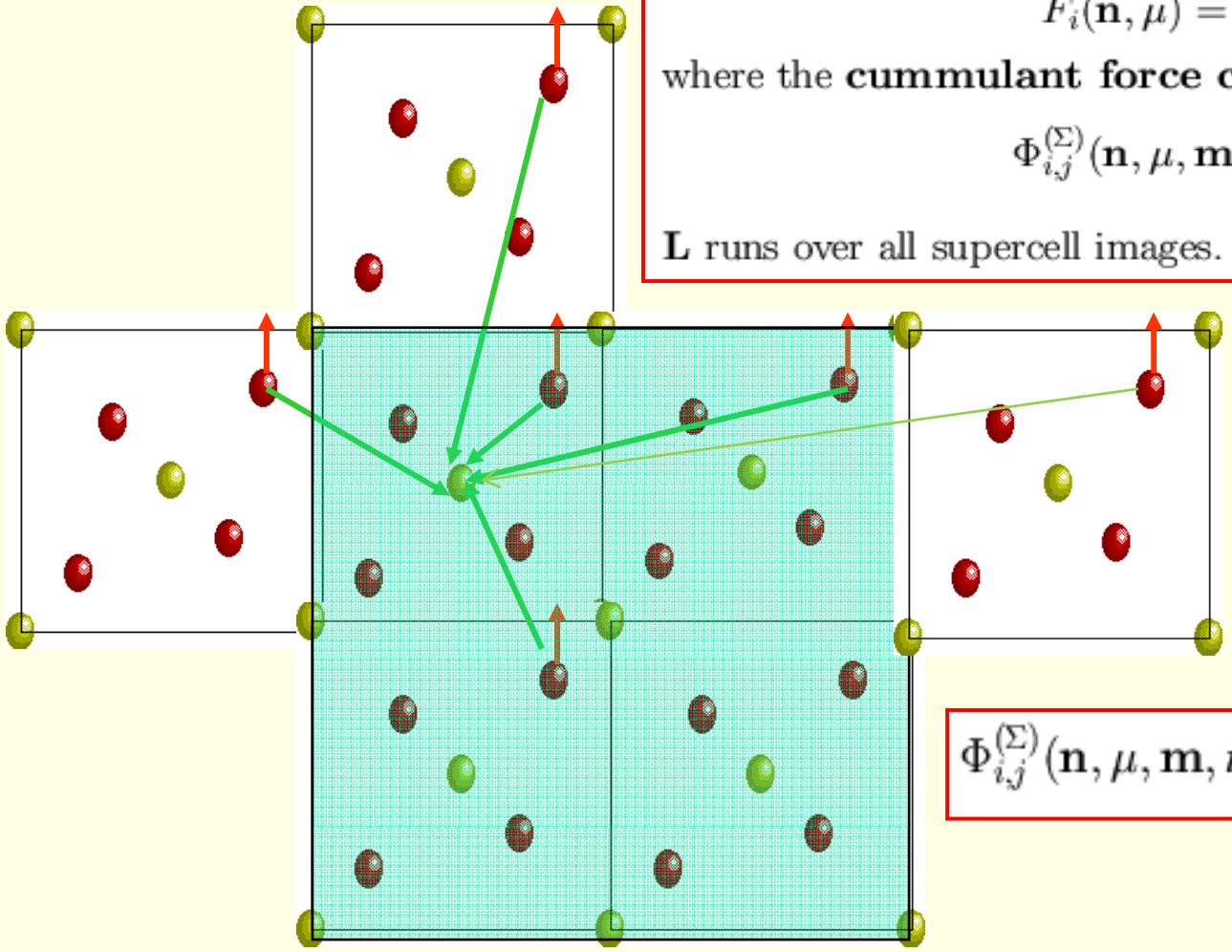
$\mathbf{L} = (L_a, L_b, L_c)$ are the indices of supercell lattice constants.
or

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

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.



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

Supercell 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):

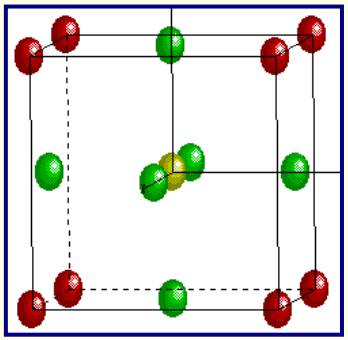
$$exp\{-2\pi i \mathbf{k}_s \cdot \mathbf{L}\} = 1$$

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.

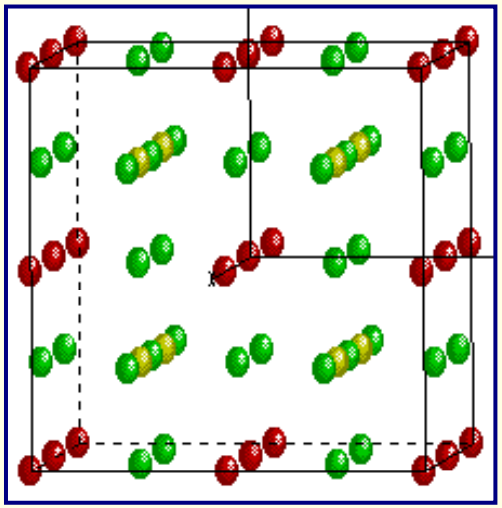
Exact wave vectors

1x1x1



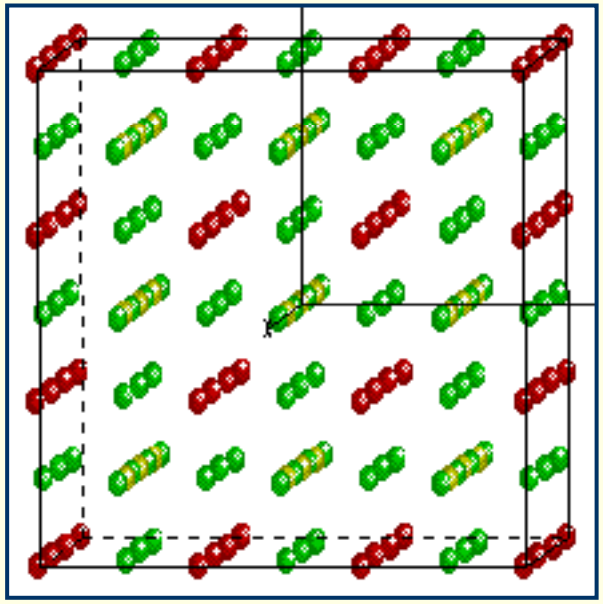
Exact: Γ

2x2x2

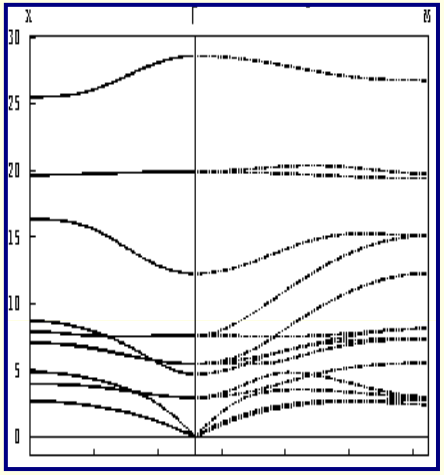


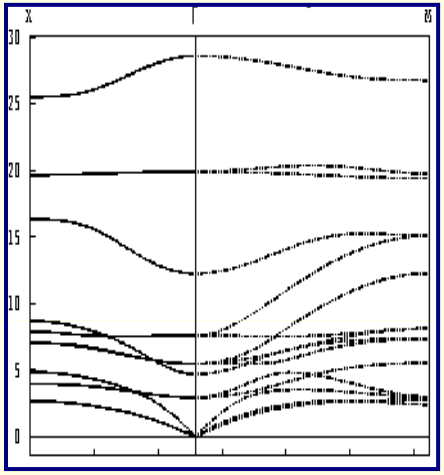
Exact: Γ, X, M, R

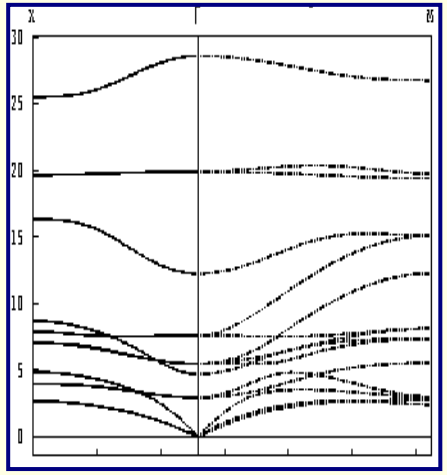
3x3x3



Exact: Γ X Γ M ⇩ Γ ⇩



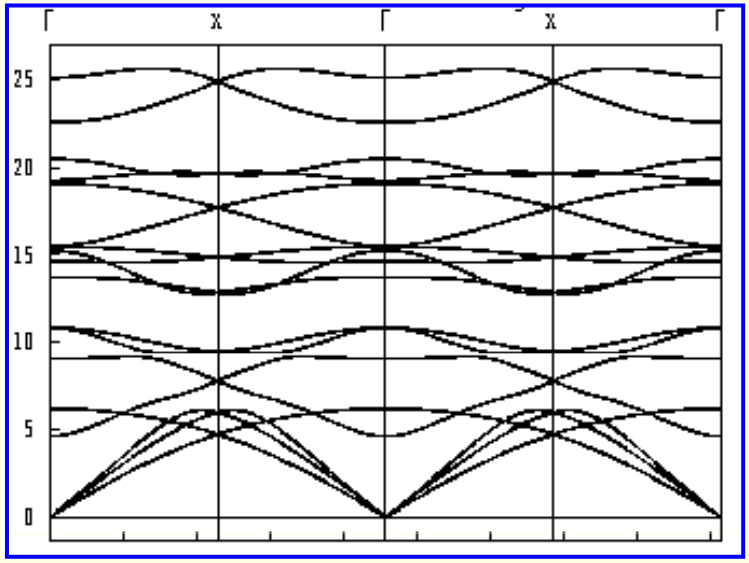




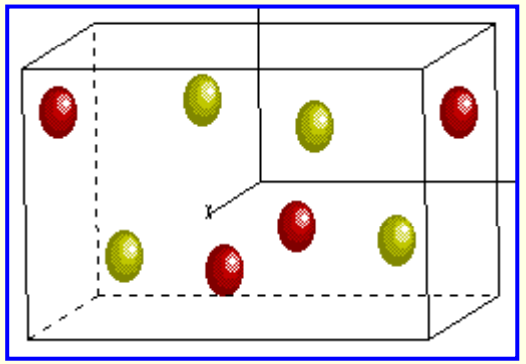
Phonon dispersions + density of states

Frequency

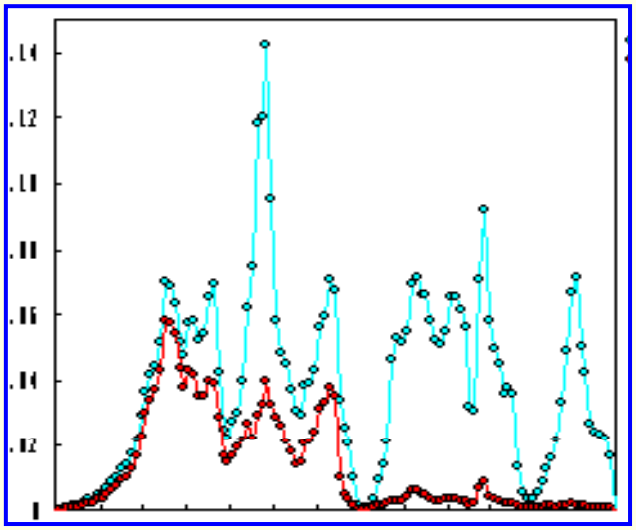
ω



GeO_2 P4_2/mnm

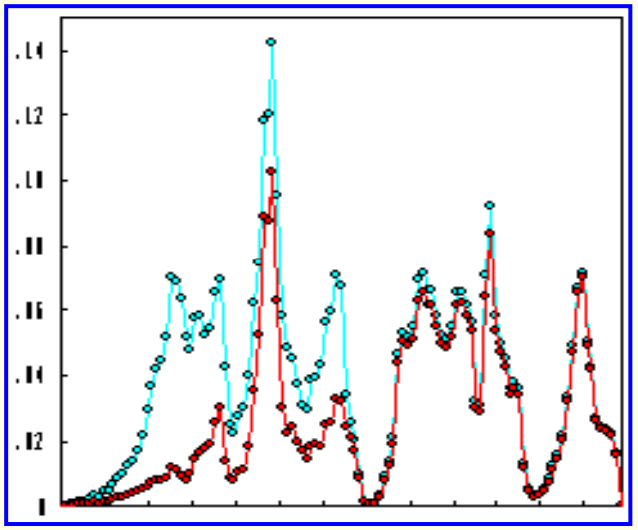


Total + Germanium



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:

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

Heat capacity C_v:

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

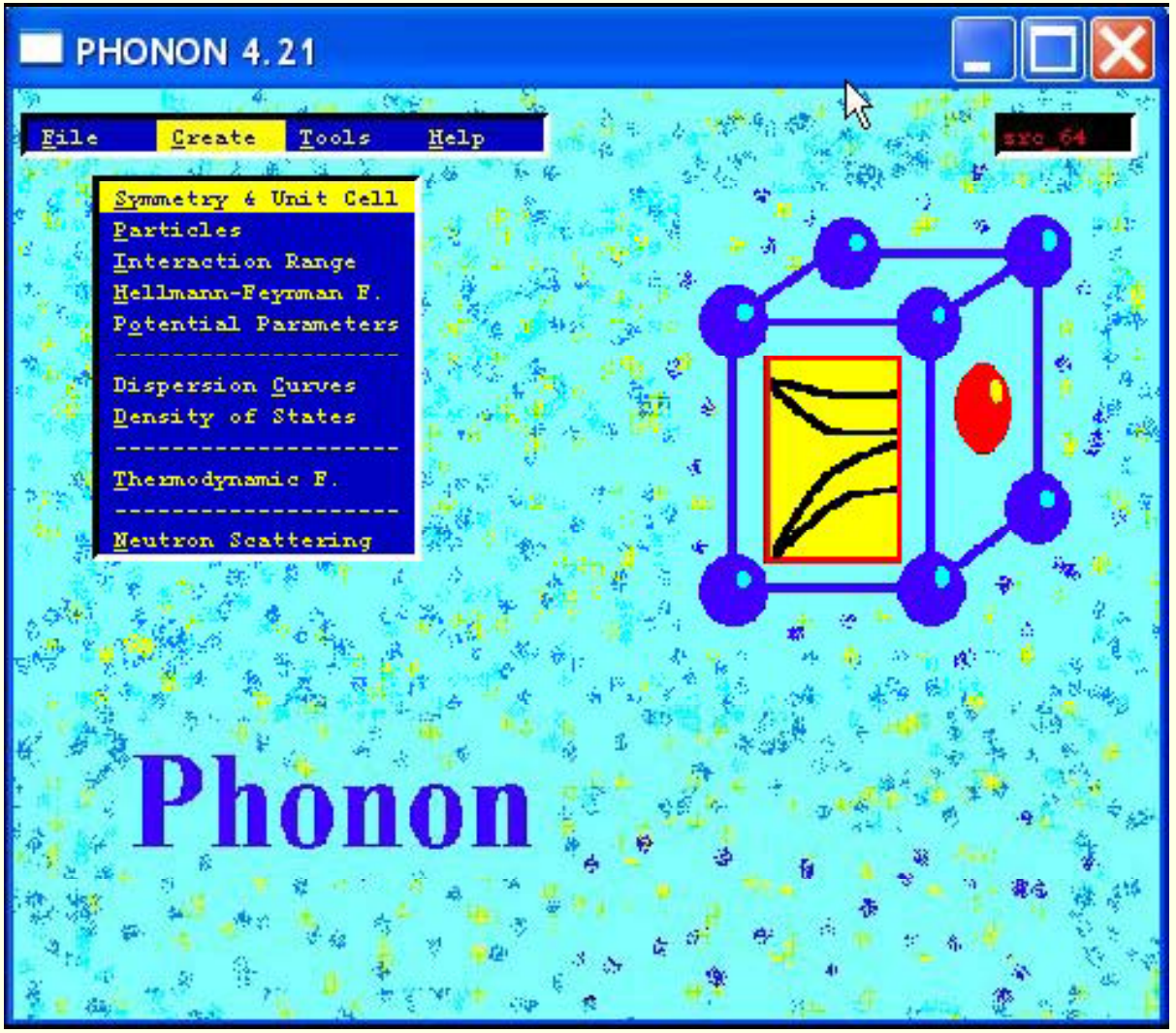
Thermal displacements:

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

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

PHONON-I

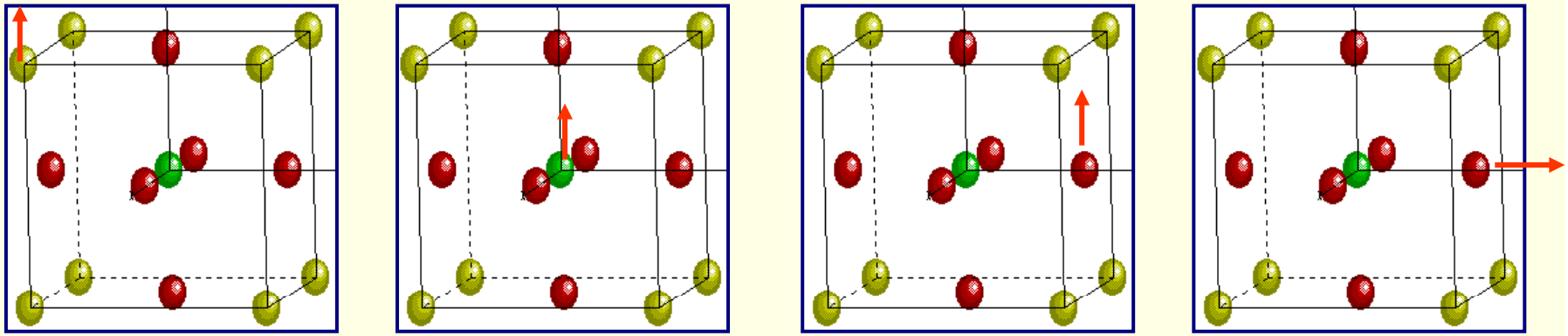
- PHONON
  - *by K.Parlinski (Crakow)*
  - *runs under MS-windows*
  - *uses a „direct" method to calculate Force-constants with the help of an ab initio program*
  - *with these Force-constants phonons at arbitrary k-points can be obtained*
- Define your spacegroup
- Define all atoms



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

Phonons:

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



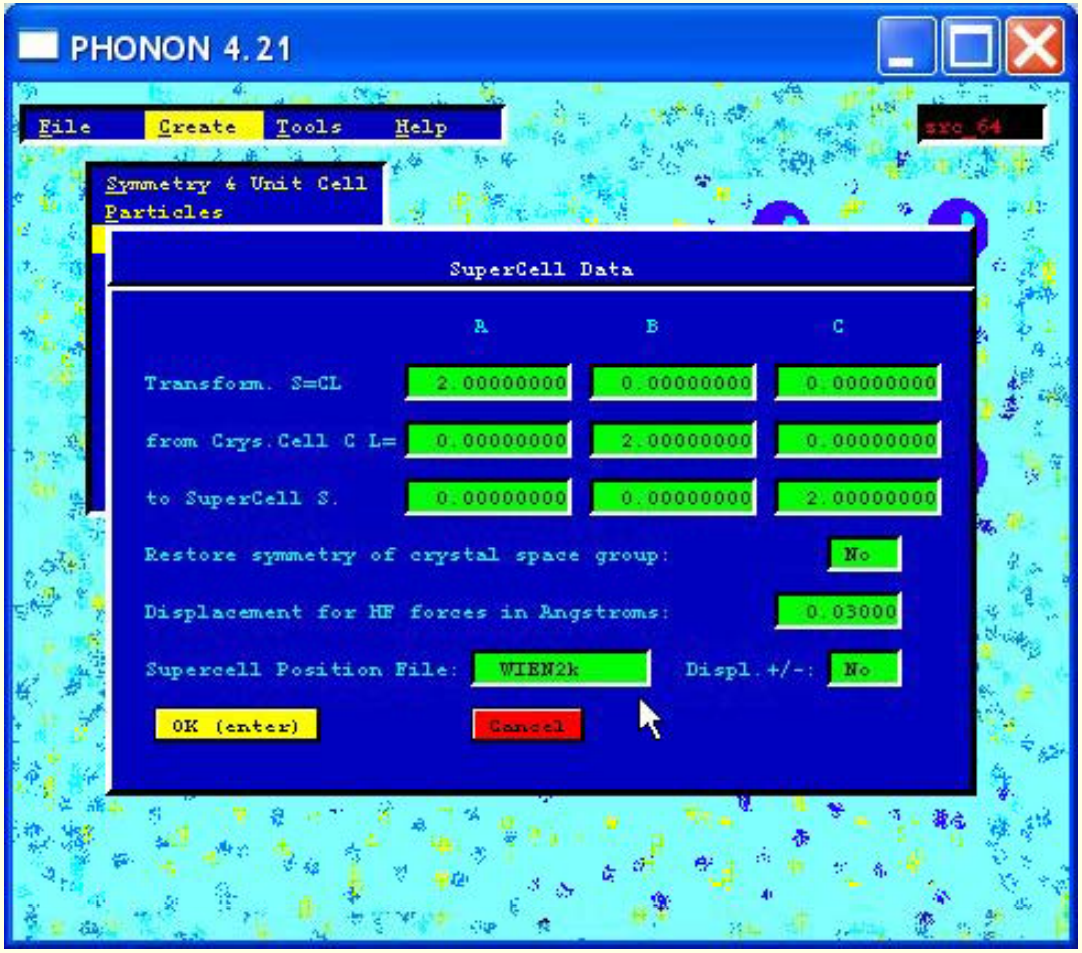
(Displacement pattern for cubic perovskite)

- *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

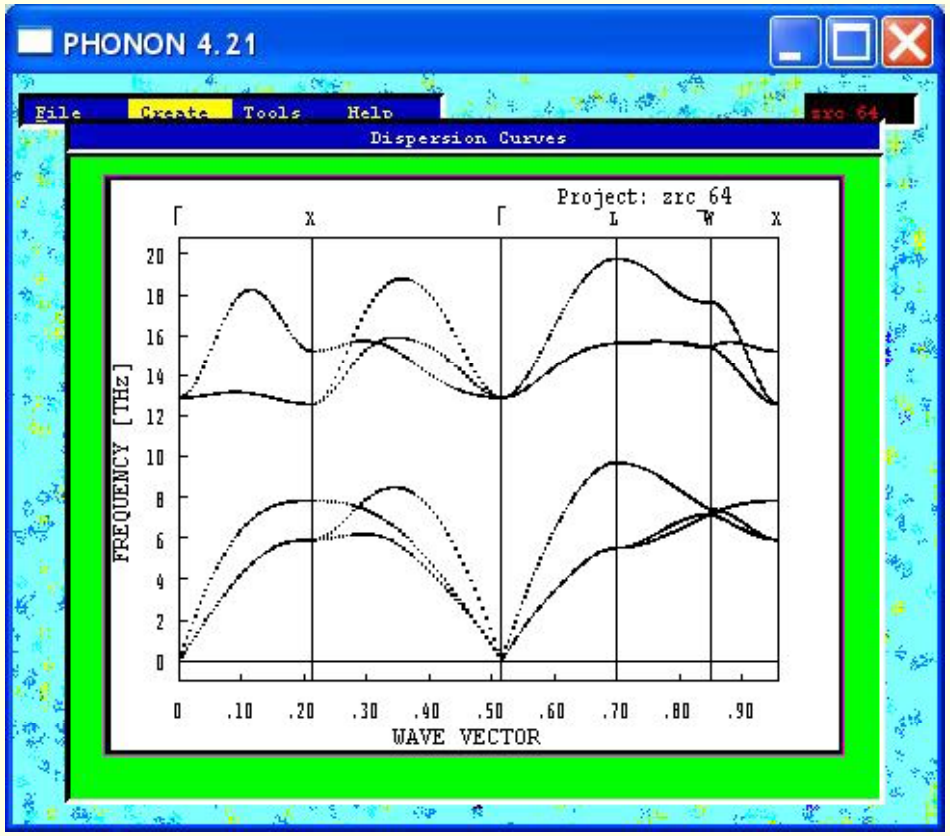
- Define an interaction range (supercell)
  - *create displacement file*
  - *transfer case.d45 to Unix*
- Calculate forces for all required displacements
  - *init_phonon_lapw*
    - for each displacement a case_XX.struct file is generated in an extra directory
    - runs nn and lets you define RMT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



- 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

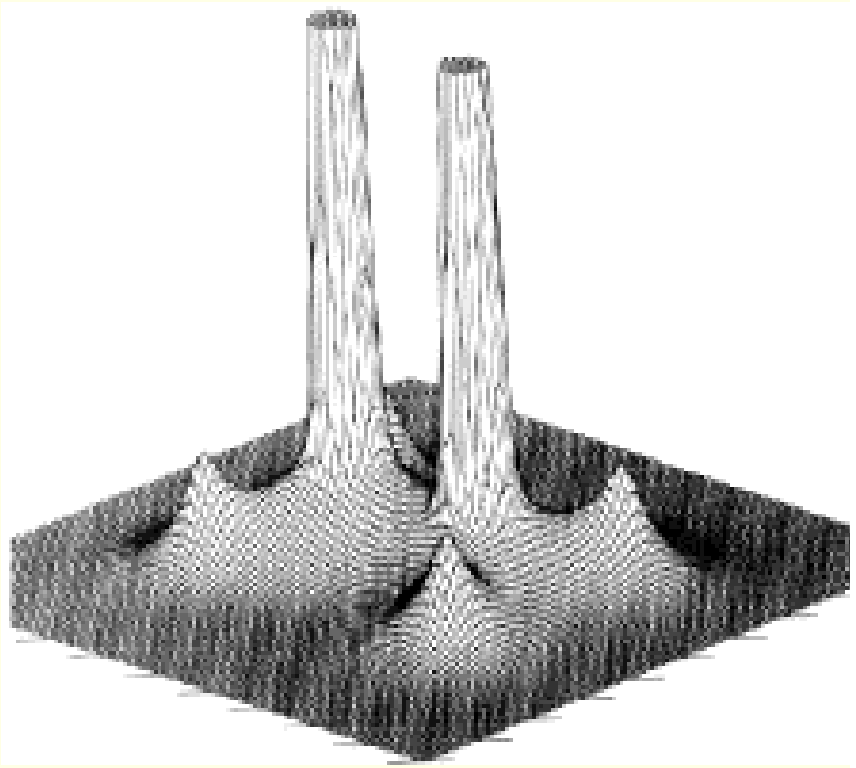
- analyze_phonon_lapw
  - *reads the forces of the scf runs*
  - *generates „Hellman-Feynman" file case.dat and a „symmetrized HF-file case.dsy (when you have displacements in both directions)*
    - check quality of forces:
    - sum F_x should be small (0)
    - abs(F_x) should be similar for +/- displacements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

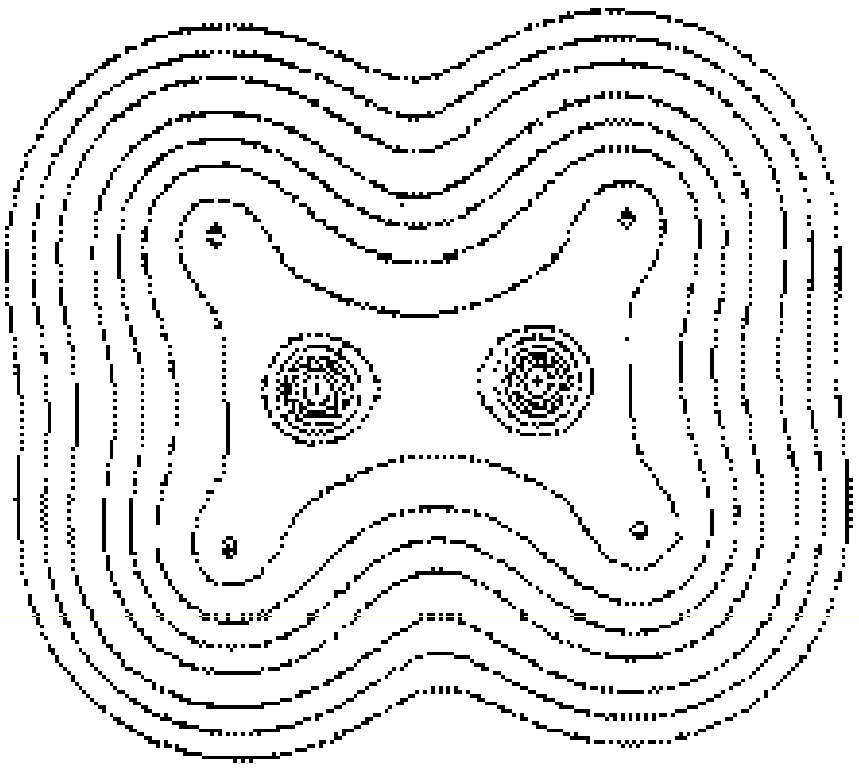


Atoms in Molecules

- 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



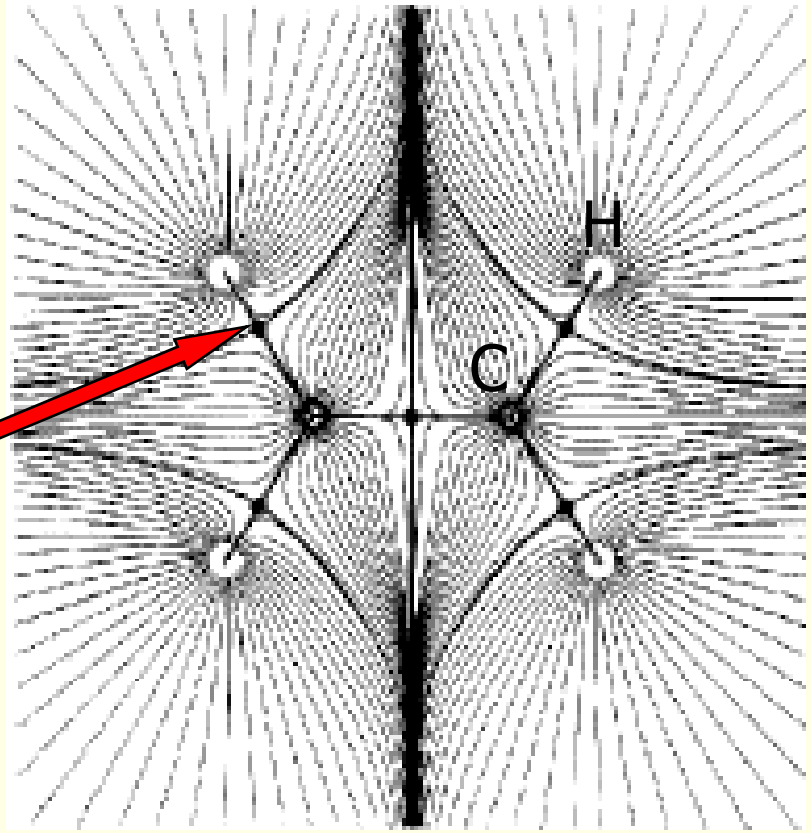


AIM-II

- Bonds are characterized by „critical points", where $\nabla\rho = 0$

  - density maximum: (3,-3); 3 negative curvatures λ, (at nucleus or non-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

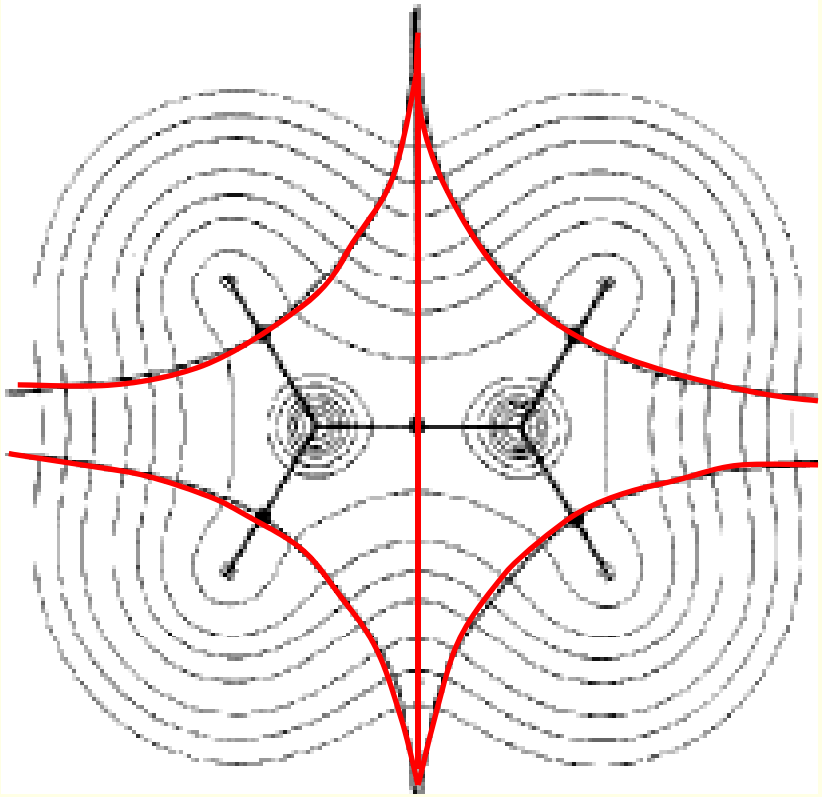


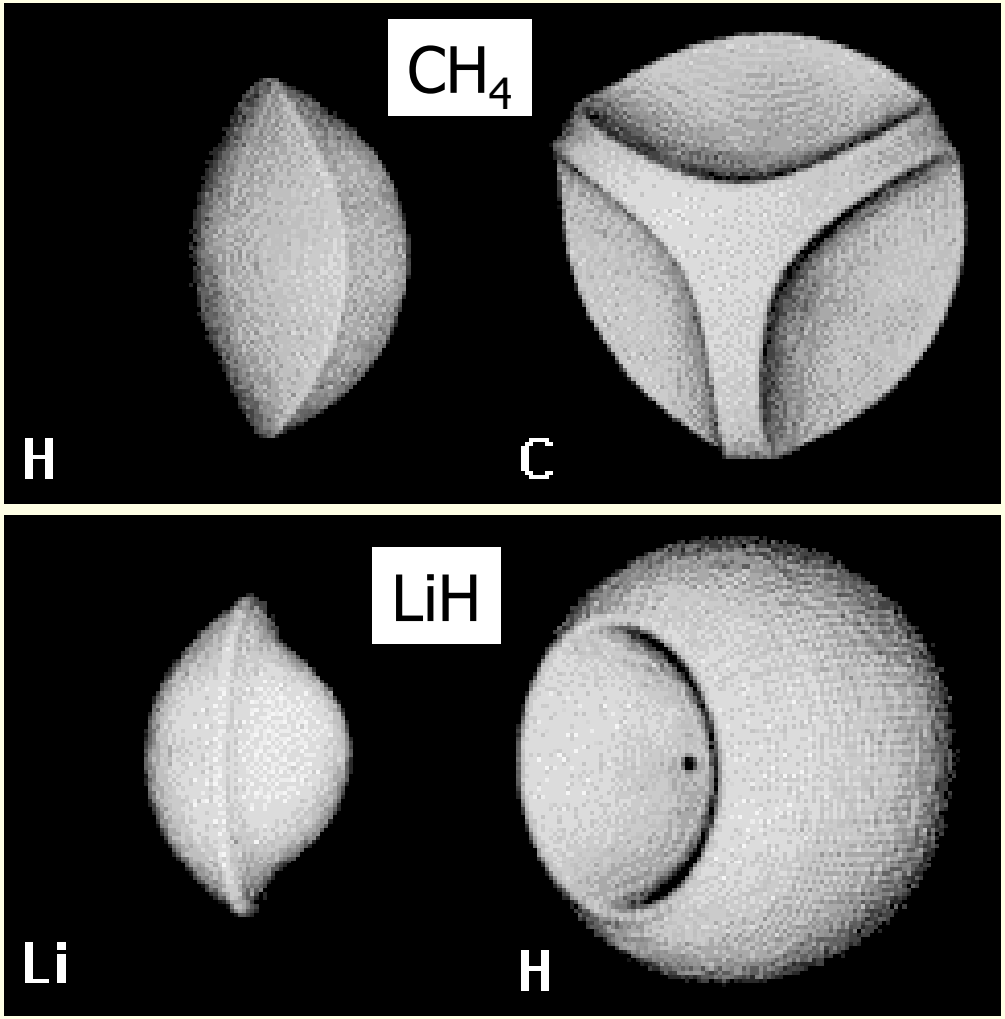
trajectories of constant $\nabla\rho$ originating at CPs in C_2H_4

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
  - *Intel Core-2 dual-core processors with fast memory bus (2-4 Gb memory, 1Gbit-net, SATA disks), Quad-cores ?*
  - *10 atom cells on 128Mb PC / 100 atom cells require 2 Gb RAM*
  - *installation support for many platforms + compiler*

- 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 (ifort, gfortran), BLAS-library (mkl, gotolib), 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_08.tar and examples (executables)
- Uncompress and expand all files using:
  - *tar –xvf wien2k_08.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; gotolib, ATLAS-BLAS)

Compilation

- most common: Intels Fortran compiler (not free for academic)
  - ***ifort*** *9.x,* ***10.0***
    - 8.0 buggy ?, 10.1 buggy ?
    - needs unlimited "stacksize" (limit stacksize=unlimited, otherwise: "segmentation fault")
    - dynamic linking necessary (depends on ifort version, requires system and compiler libraries at runtime, may need $LD_LIBRARY_PATH)
  - *ifc (7.1) ,* ***gfortran***
    - -static linking possible
    - for ifc: patch for 2 externals, see faq-web page or mailing-list digest
  - ***mkl 9.x;*** *mkl 10.0 (buggy ?);* ***gotolib,*** *acml-lib, ATLAS-BLAS*
    - -L/opt/intel/mkl/lib –lmkl_em64t → libmkl_em64t.so
  - *compiler/linker options depend on compiler version + Linux-version !!*
    - -FR (free format) -lguide –lpthread -pthread

userconfig_lapw

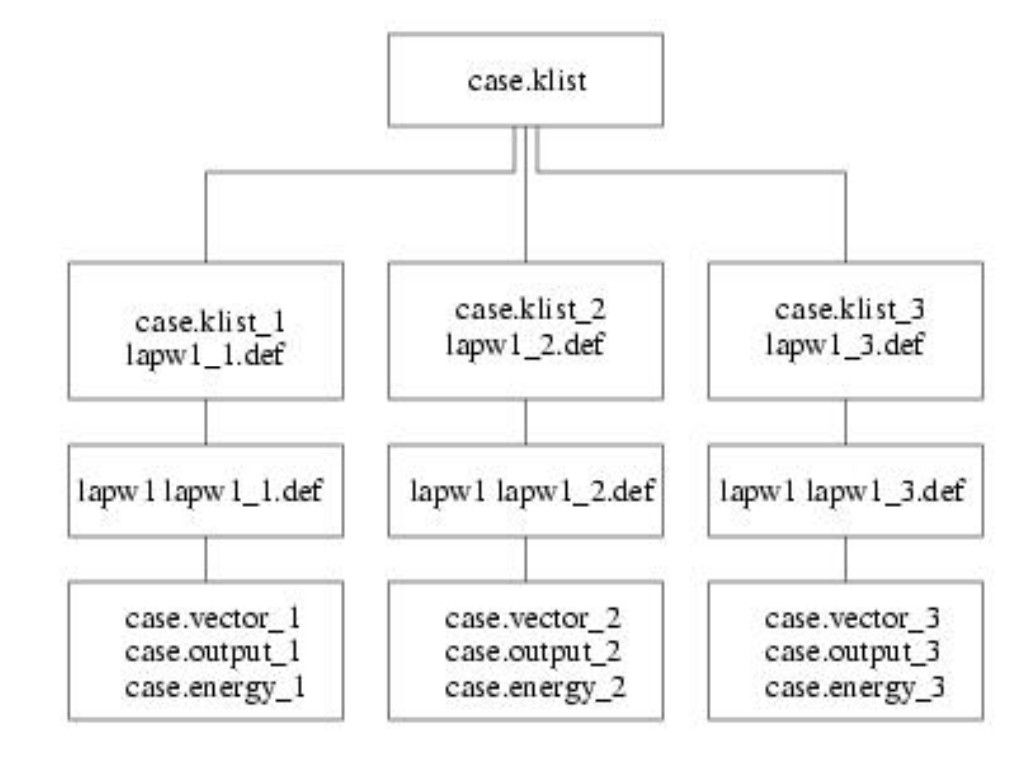
- Every user should run userconfig_lapw
  - *support for tcsh and bash*
  - *sets PATH to $WIENROOT, sets variables and aliases*
    - $WIENROOT, $SCRATCH, $EDITOR, $PDFREADER, $STRUCTEDIT_PATH
    - pslapw: ps –ef | grep lapw
    - lsi: ls –als *.in* lso: ls -als *.output*
    - lss: *.scf* lsc: *.clm*
    - limit stacksize unlimited
    - $OMP_NUM_PTHREADS (for mkl+dual core); $LD_LIBRARY_PATH
- w2web: acts as webserver on a userdefined (high) port.
  - *define user/password and port. (http://host.domain.xx:5000)*
  - *behind firewall create a „ssh-tunnel": ssh -fNL 2000:host:2000 user@host*
  - *~/.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)

k-point Parallelization

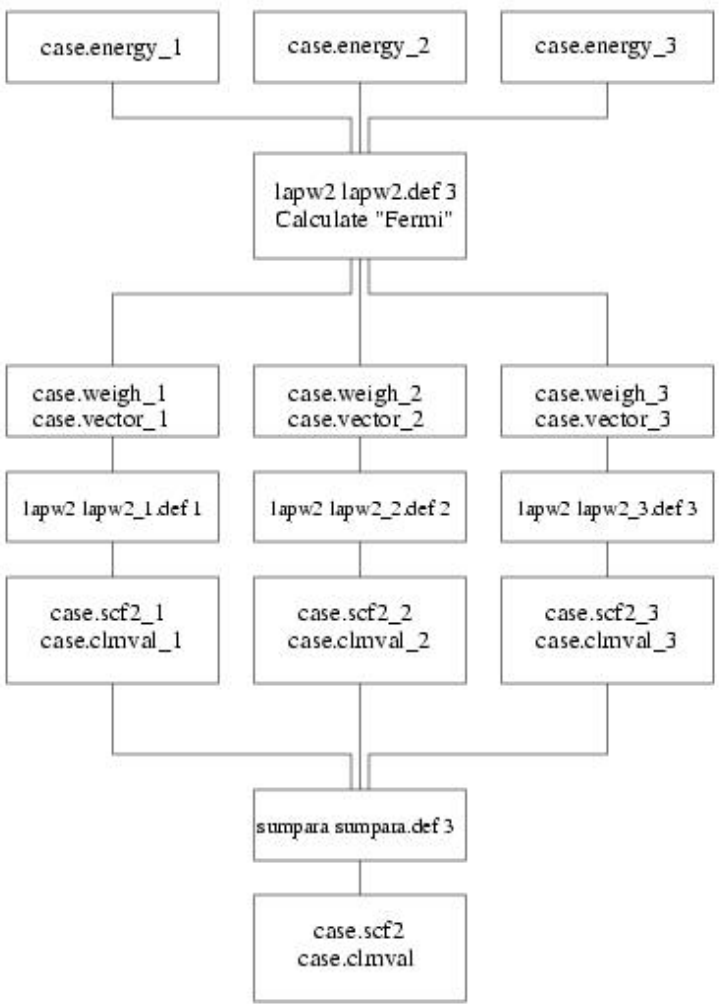
- very efficient parallelization even on loosely coupled PCs (slow network): lapw1+lapw2
  - *common NFS filesystem (files must be accessible with the same path on all machines; use /host1 as data-directory on host1)*
  - *rsh/ssh without password (.rhosts; private/public keys)*
    - ssh-keygen –t rsa
    - append .ssh/authorized_keys on remote host with id_rsa.pub of local host
    - .machines file:
      - 1:host1 (speed:hostname)
      - 2:host2
      - granularity:1 (1:10k+10k; 3: 3+3+3+3+3+3+rest ➔load balancing, not with $SCRATCH, -it
      - extrafine:1 (rest in junks of 1 k)
    - testpara (tests distribution); run_lapw –p
  - *case must fit into memory of one PC !*
  - *high NFS load: use local $SCRATCH directory (only with commensurate k-points/hosts)*

Flow of parallel execution

lapw1para



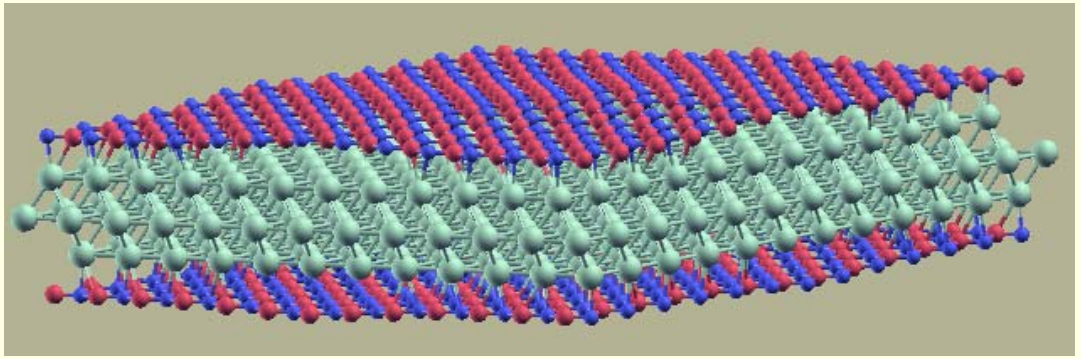
lapw2para



fine-grain mpi-parallelization

- for bigger cases (> 30 atoms)
- faster network (Gbit, 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!!!
- simultaneous k-point and mpi-parallelization possible

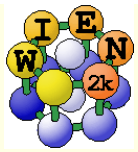
- *BN/Rh(111) nanomesh:*



  - unitcell with 1150 atoms (556 non-equiv.)
  - NMAT=45000-60000; 64 cpus, 2h / iteration

WIEN2k_08.1

- new version with largely improved performance
  - *lapw0: speed-up (non-spinpolarized)*
  - *lapw1 + lapw2: large speedup,*
    - new iterative diagonalization (5x faster, run_lapw –it)
  - *largely improved mpi-versions*
    - perfect scaling with number of cpus for lapw0_mpi (atoms), lapw1_mpi (PW, memory, HAMILT+HNS "scales"; diagonalization not "perfect") and lapw2_mpi (atoms + PW)
    - Gbit network "enough"
    - unitcell with 1150 atoms (556 non-equiv.): BN/Rh(111)
      - NMAT=45000-60000; 64 cpus, 2h / iteration
  - *mixer (L.Marks): 30-50% less scf-cycles, very stable, "real" convergence (no "pseudo-convergence" due to small mixing)*
  - *min_lapw: charge extrapolation for new positions (30-70% less iterations)*


I
E
W
N
2k


TU
WIEN

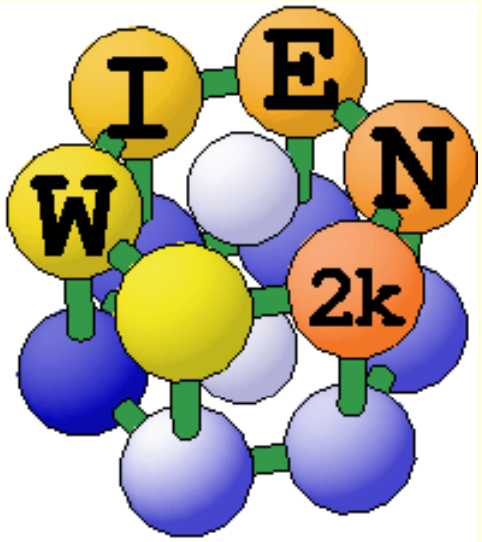
AURORA

Development of W2GRID

DI Johannes Schweifer

Project 8

9th Aurora plenary meeting (Strobl)



A computational task within W2GRID

user

computational task (run_lapw)

GridClient

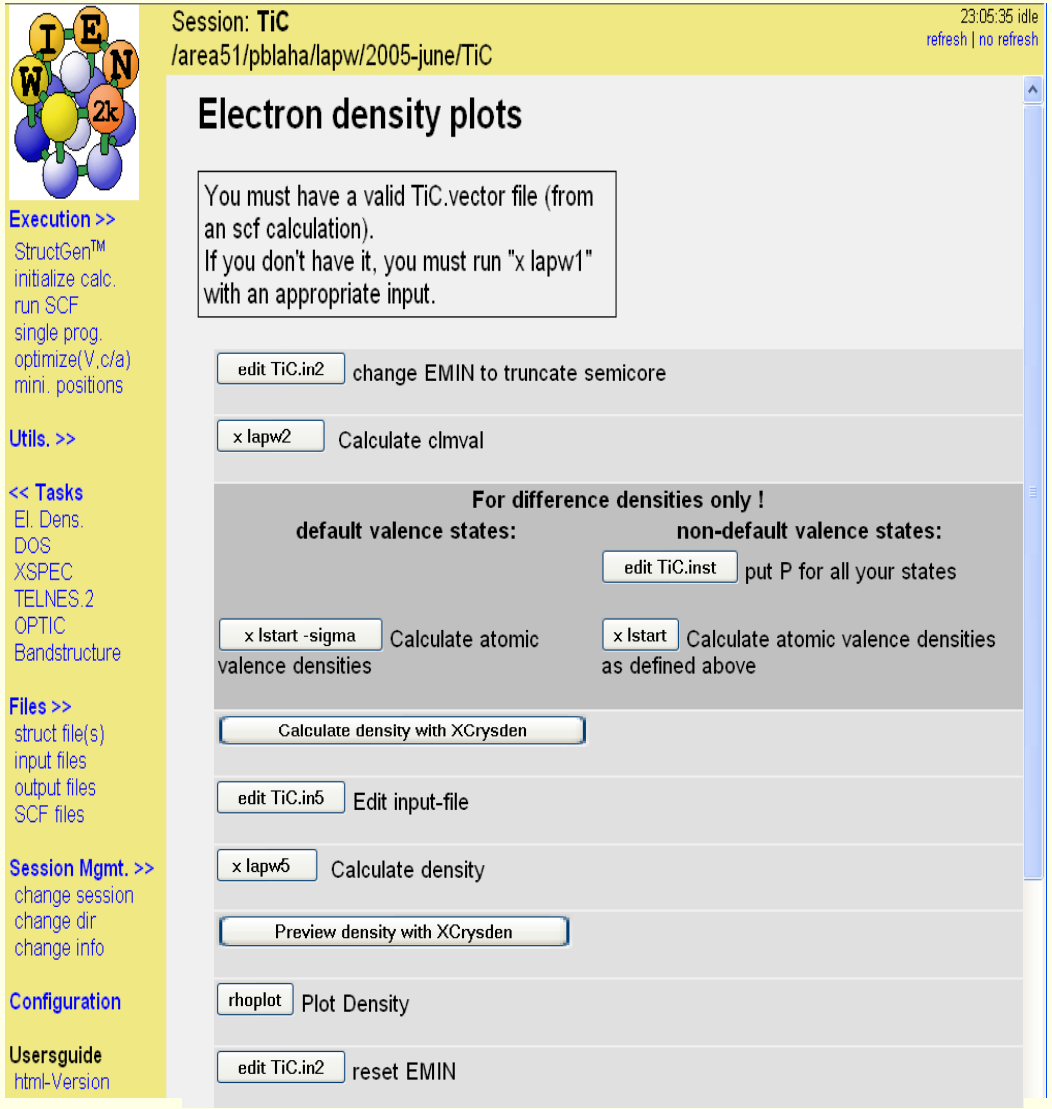
1) analysis of job (memory, time, nodes)
2) checking all hosts (load)
3) selection of suitable hosts
4) automatic distribution of tasks to „best" hosts
5) checking of running jobs, control files
6) returns output data

platform-dependent execution

GridServer on local workstations, clusters or supercomputers

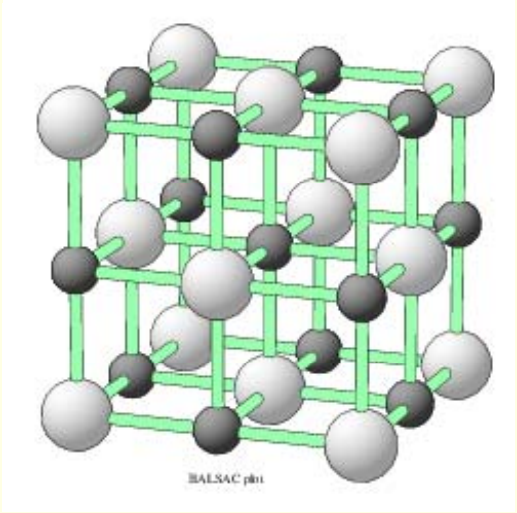
Task for electron density plot

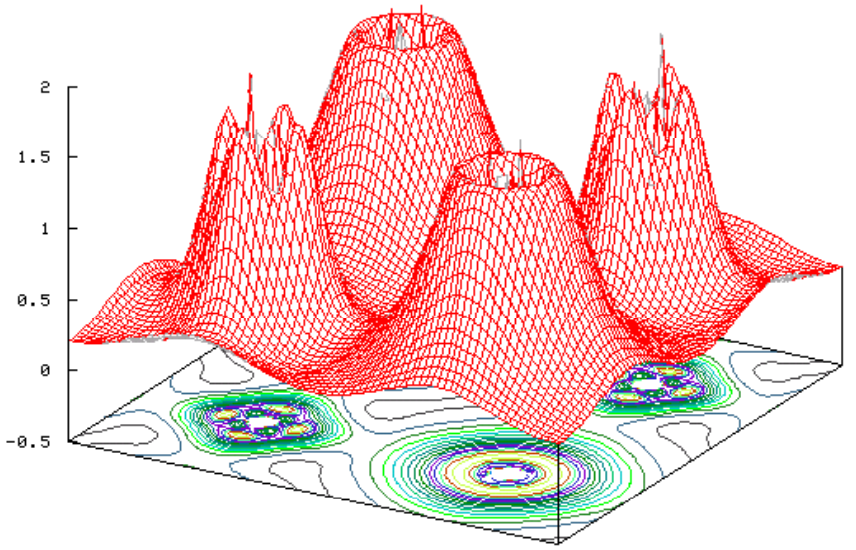
- A task consists of
  - *a series of steps*
  - *that must be executed*
  - *to generate a plot*
- For electron density plot
  - *select states by E-window in case.in2 (e.g. valence e⁻: Ti-3d,4s, C-2s,2p)*
  - *for difference densities make sure you calculate the same states for the free atoms*
  - *select plane for plot (do not put an atom at the corner or edges)*
  - *generate 3D or contour plot with gnuplot or Xcrysden (Tone.Kokalj@ijs.si)*
  - *reset EMIN in case.in2*



TiC electron density

- NaCl structure (100) plane
- Valence electrons only
- plot in 2 dimensions
- Shows
  - *charge distribution*
  - *covalent bonding*
    - between the Ti-3d and C-2p electrons
  - e_g/t_{2g} *symmetry*





Download hardcopy in PostScript format

Select plot type: 3D-plot ◉ Contur-plot ○ with labels ☐

Min -0.5 Max 2 Delta 0.1

plot electron density

edit TiC.in2 reset EMIN

band structure

Session: **TiC**
/susi/pblaha/lapw/TiC

Band structure

Generate k-mesh using XCrysden (save klist as xcrysden.klist)

fcc | create TiC.klist_band

x lapw1 -band Calculate Eigenvalues ☑ interactively

needed only for continuous lines in the plot (not for non-symmorphic spacegroups)!

x irrep Calculate irreducible representations ☑ interactively

for band character plots only!

x lapw2 -band -qtl Calculate partial charges ("qtl"-file) ☑ interactively

edit TiC.insp Insert correct EF

x spaghetti Calculate bandstructure

plot bandstructure Plot bandstructure

- Note: spaghetti may fail if you have an old (incompatible) case.irrep file

case.insp

Session: **TiC**
/susi/pblaha/lapw/TiC

File:

/susi/pblaha/lapw/TiC/TiC.insp

contine with bandstructure | Save | Download this file:

Header from TiC.qtl and possible FERMI energies:

```
 ATOM   1  tot,0,1,2,3,xdos(i,j),j=1,i),i=1,lxdos2)
 ATOM   2  tot,0,1,2,D-eg,D-t2g,3

/susi/pblaha/lapw/TiC/TiC.scf: EF (TETRAH.M.)=   0.74210
/susi/pblaha/lapw/TiC/TiC_save.scf: EF (TETRAH.M.)=   0.74250
/susi/pblaha/lapw/TiC/TiC_vol_-10.0.scf: EF (TETRAH.M.)=   0.84297
/susi/pblaha/lapw/TiC/TiC_vol__-5.0.scf: EF (TETRAH.M.)=   0.79064
/susi/pblaha/lapw/TiC/TiC_vol__10.0.scf: EF (TETRAH.M.)=   0.65771
/susi/pblaha/lapw/TiC/TiC_vol___0.0.scf: EF (TETRAH.M.)=   0.74249
/susi/pblaha/lapw/TiC/TiC_vol___5.0.scf: EF (TETRAH.M.)=   0.69841
```

```
### Figure configuration
 5.0   3.0                          # paper offset of plot
10.0  15.0                          # xsize,ysize [cm]
 1.0    4                           # major ticks, minor ticks
 1.0    1                           # character height, font switch
 1.1    2    4                      # line width, line switch, color swit
### Data configuration
-1.0  1.0  1                       # energy range, energy switch (1:Ry, 2
1      0.74250                      # Fermi switch,  Fermi-level (in Ry u
1   999                            # number of bands for heavier plotting
2     2    0.8                     # jatom, jtype, size  of heavier plott
```