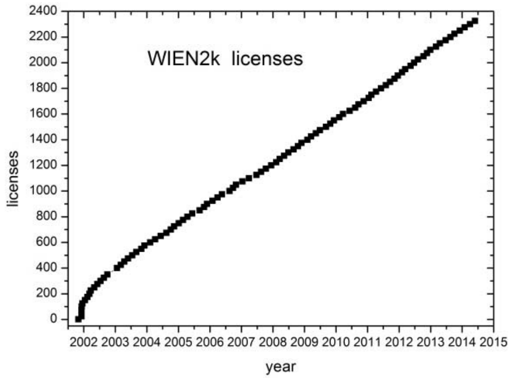
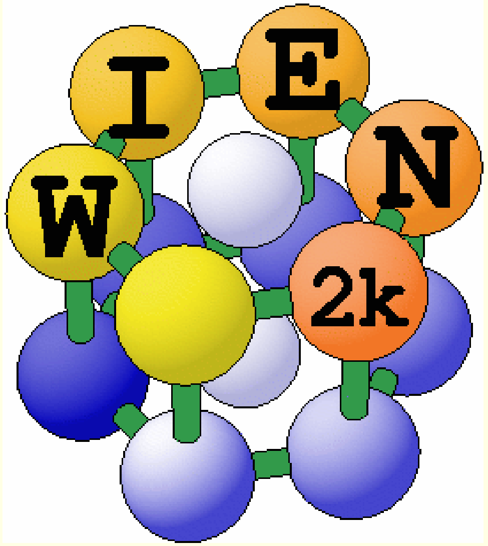
WIEN2k software package





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

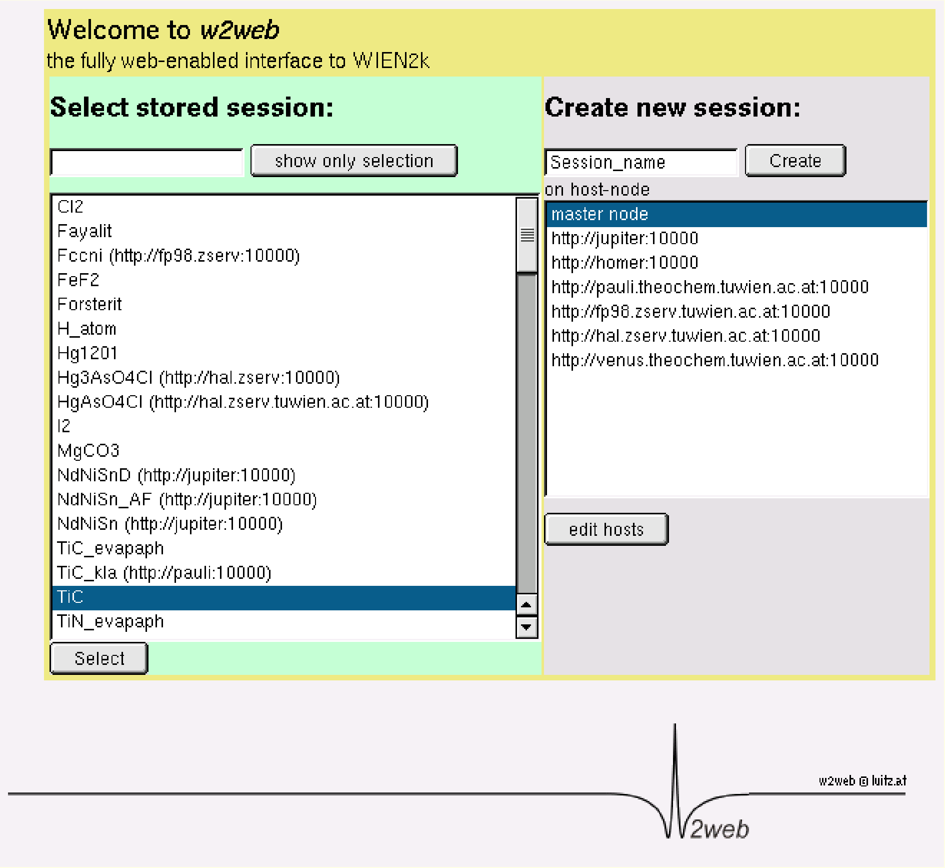
WIEN97: ~500 users
WIEN2k: ~2350 users

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)*



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:

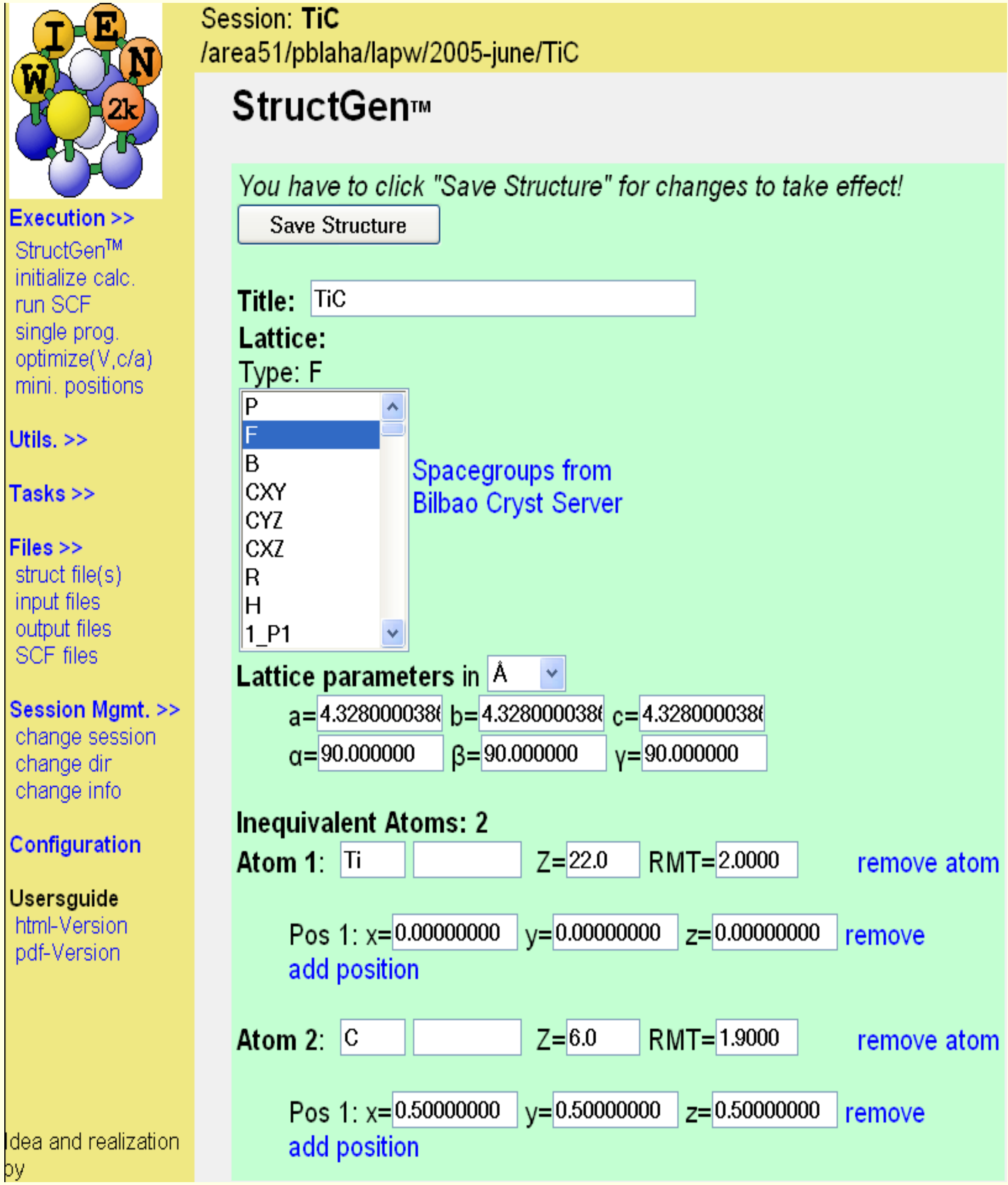
Session_name

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 or xyz 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!

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

Atom 2: C Z=6.0 RMT=1.9000
Pos 1: x=0.50000000 y=0.50000000 z=0.50000000

Spacegroup $P4_2/mnm$

Structure given by:

spacegroup

lattice parameter

positions of atoms (basis)

Rutile TiO_2:

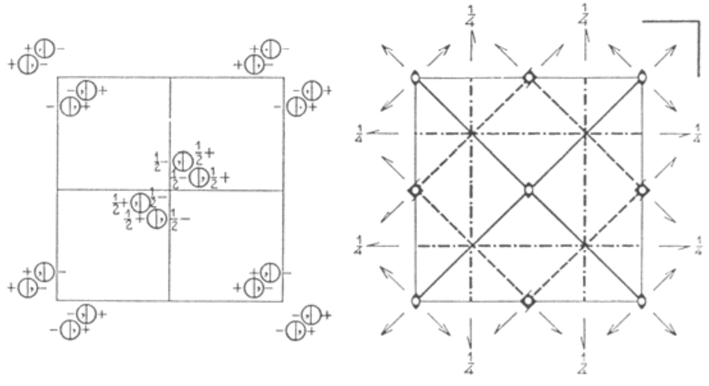
$P4_2/mnm$ (136)

a=8.68, c=5.59 bohr

Ti: (0,0,0)

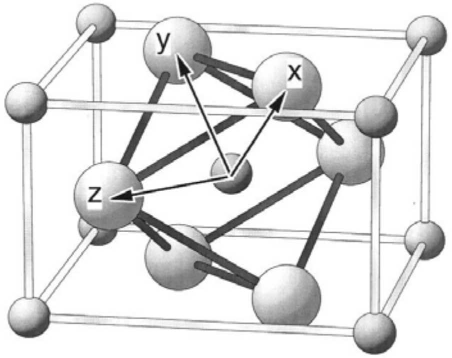
O: (0.304,0.304,0)

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



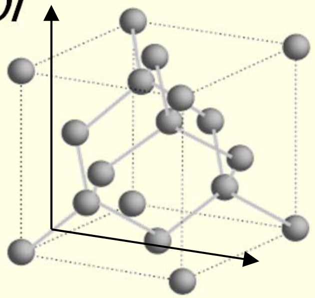
Origin at centre (*mmm*)

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

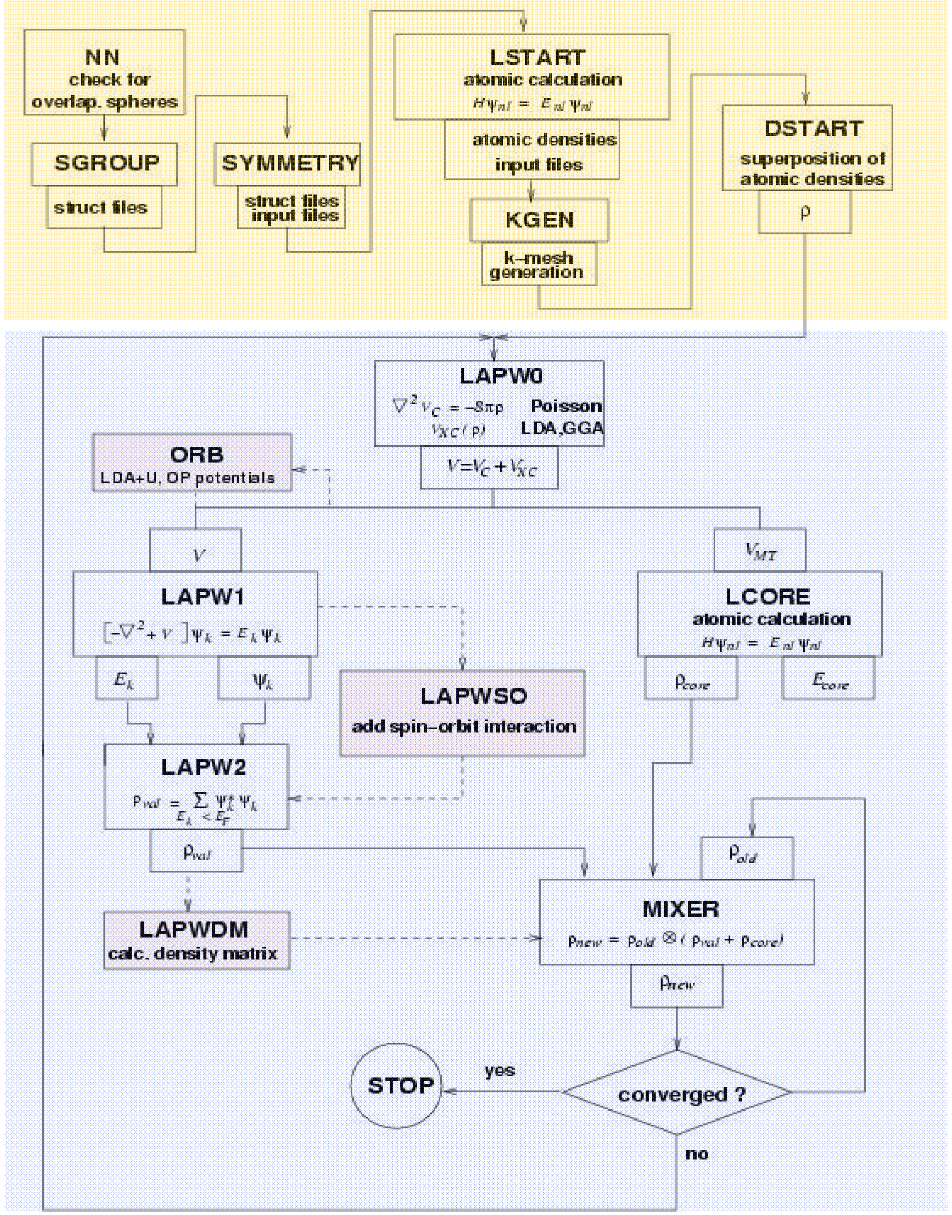
- Specify:
    - *Number of nonequivalent atoms*
    - *lattice type (P, F, B, H, CXY, CXZ, CYZ) or spacegroup symbol*
        - if existing, you must use a SG-setting with inversion symmetry:
            - Si: ±(1/8,1/8,1/8), not (0,0,0)+(1/4,1/4,1/4)!
    - *lattice parameters a,b,c (in Å or bohr)*
    - *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*
- „set RMT and continue“: (specify proper “reduction” of NN-distances)
    - *non-overlapping „as large as possible“ (saves time), but not larger than 2.5 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
  - *step-by-step or batch 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*



RKMAX

- The convergence criterion in APW is the product of R_{MT}.Kmax

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

- http://www.wien2k.at/reg_user/faq/rkmax.html
- medium quality convergence for **smallest** atom:

Rkmax	Element
3.0	H
4.5	Li
5.0	Be, B, Si
5.5	C, P
6.0	N, S
6.5	O, Cl, Na, K, Rb, Cs, Mg, Ca, Sr, Ba, Al
7.0	F
7.5	Sc-Cr, Ga-Br, Y-Mo
8.0	Mn-Zn, Ru-Cd, In-I, La, Ce, Hf-Re
8.5	Os-At, Pr-Lu, Ac-Lr

- basis set scales with $RKmax^3$
- cputime scales with N_{PW}^3

- increasing Rkmax by 10 %

  ⇨ doubles cputime

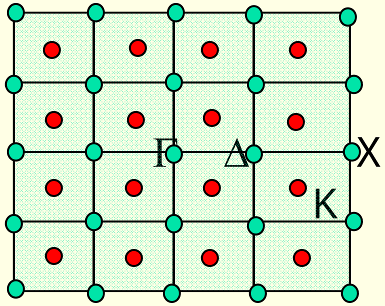
START with **SMALL** Rkmax (relaxation), **increase/test** later

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/TEMPS 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)
  - *automatically "adds inversion"*
    - time inversion holds and E(k) = E(-k)
    - except in magnetic spin-orbit calculations (`x –so kgen`; uses case.ksym file)
    - `x –fbz kgen` (generates „full mesh" in BZ)
  - *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/relaxations*
  - *continue **later** with **finer** mesh*
    - mesh was good if nothing changes and scf terminates after few (3) iterations
  - *use 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
```

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 (-it1,-it2 , -noHinv)* *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 :GAP :CTO001 :NTO001 :QTL001
    - :FOR002: 2.ATOM 19.470 0.000 0.000 19.470
    - :FGL002: 2.ATOM 13.767 13.767 0.000 total forces
    - :LAT :VOL :POSxxx

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

- „QTL-B“ value too large - STOP (or :WARN): “ghostbands”
  - *identify for which **eigenvalue**, **atom** and ℓ it happens, check E_F (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 are adapted automatically but may need changes for
      - surfaces, molecules (negative EF) or heavy elements (EF often larger than 1.0)
    - add a local orbital (or adjust its energy)
  - *if QTL-B occurs for an atom with large RMT, reduce RMT*
    - this may happen for larger RKMAX („numerical linear dependency“)
- scf-cycle diverges (grep :DIS case.scf):
  - *check structure (most likely a wrong structure caused divergence);*
  - *reduce mixing in case.inm slightly; rm *.broyd* case.scf; x dstart*
  - *check E-parameters (see above), check :NEC01*

case.in1

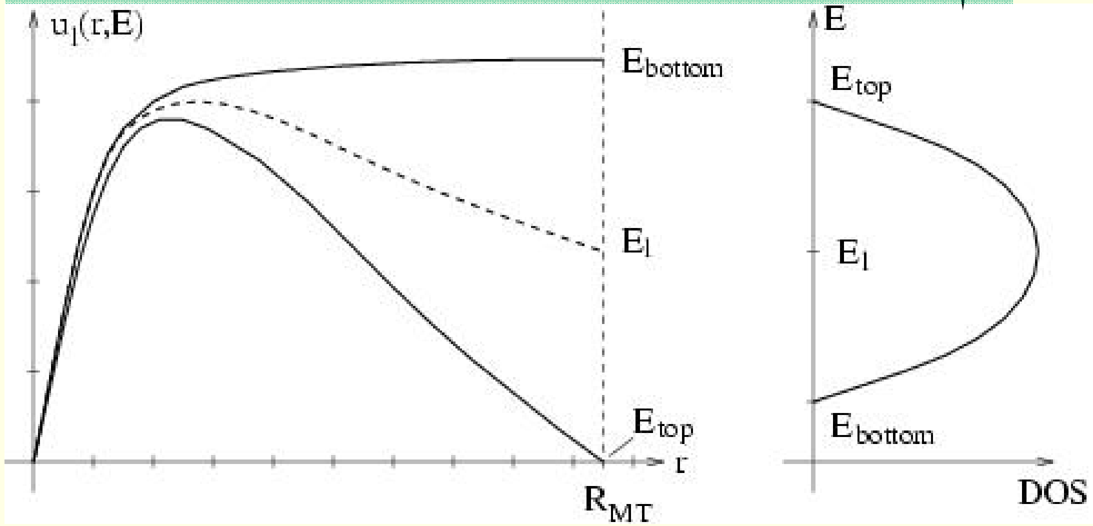
set E_l to E_F-0.2 Ry

```
WFFIL    EF=0.634                      (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
K-VECTORS FROM UNIT:4  -7.0   1.5   16   emin/emax; nband
```

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

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

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



case.klist, case.in2

```
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
```

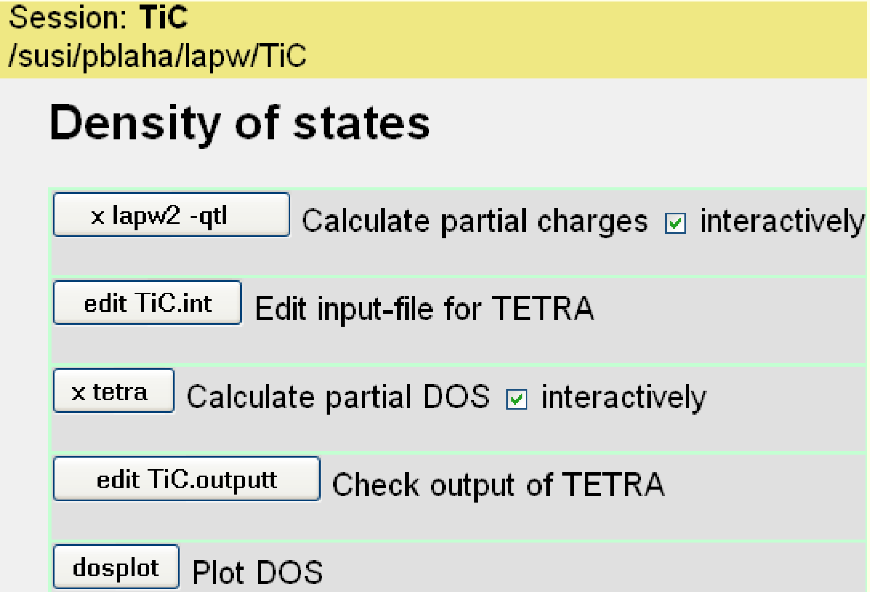
$$\rho(r) = \sum_{\boxed{LM}} \rho_{LM}(r) Y_{LM}(\hat{r}) \qquad \rho(r) = \sum_{G}^{\boxed{GMAX}} \rho_G e^{iGr}$$

Properties with WIEN2k - I

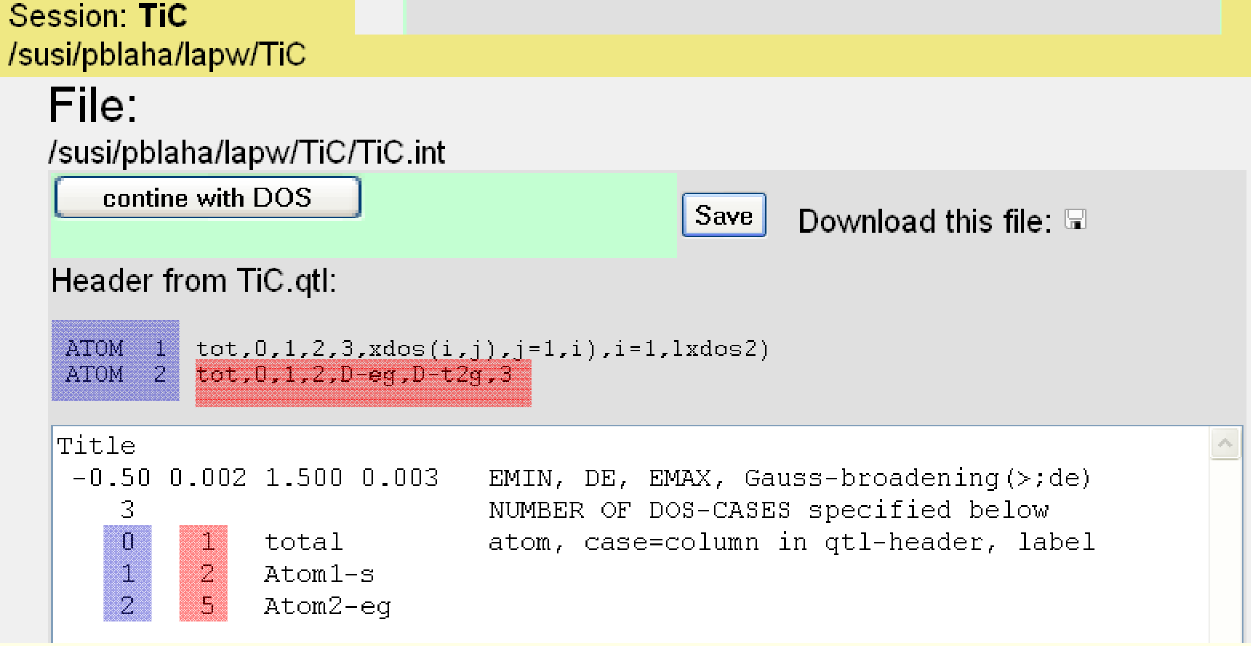
- Energy bands
  - *classification of irreducible representations*
  - *´character-plot´ (emphasize a certain band-character)*
- Density of states
  - *including partial DOS with l and m- character (eg.* p_x *,* p_y *,* p_z *)*
- Electron density, potential
  - *total-, valence-, difference-, spin-densities,* 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*

partial charges “qtl” + DOS

- be sure to have case.vector 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


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

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

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
```

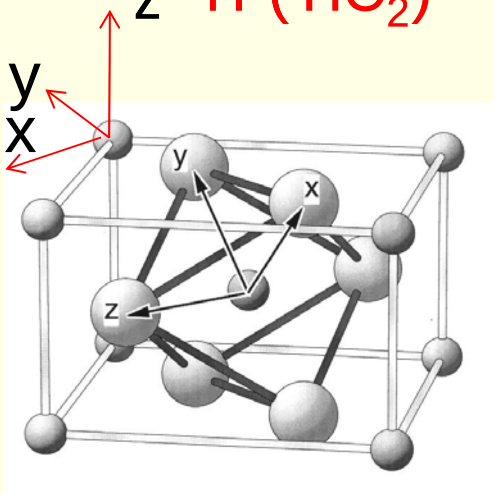
```
Title
 -0.50 0.002 1.500 0.003   EMIN, DE, EMAX, Gauss-broadening(>;de)
    3                      NUMBER OF DOS-CASES specified below
    0    1   total         atom, case=column in qtl-header, label
    1    2   Atom1-s
    2    5   Atom2-eg
```

partial charges:

- local rotation matrix:
  - *transfers z (y) into highest symmetry*
  - *reduces terms in LM series*
  - *"chemical" interpretation*
    - p_x is different from p_y

$$\begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Ti (TiO_2)



  - *see case.struct and case.outputs*

- x qtl (instead of x lapw2 -qtl)
  - ***f-**orbitals*
  - *qtls for **different coordinate system** (eg."octahedral" in TiO_2)*
  - *relativistic basis (**$p_{1/2}$-$p_{3/2}$** or **$d_{3/2}$-$d_{5/2}$** splitting in so calculation)*
  - *for angular dependend TELNES (ISPLIT 88, 99)*

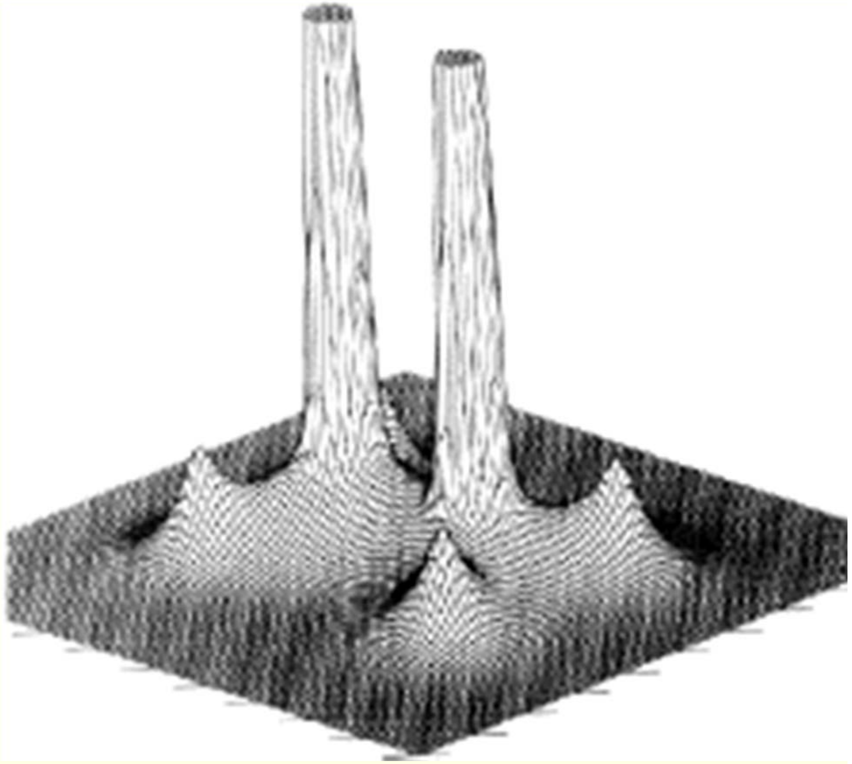
Properties with WIEN2k - I

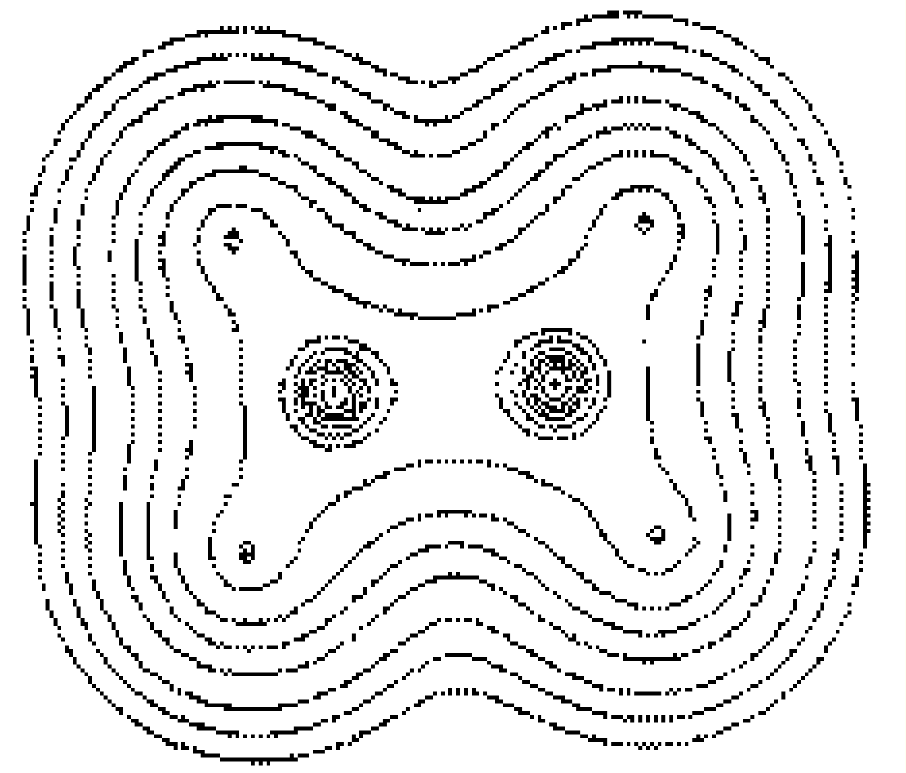
- Energy bands
  - *classification of irreducible representations*
  - *´character-plot´ (emphasize a certain band-character)*
- Density of states
  - *including partial DOS with l and m- character (eg.* p_x , p_y , p_z *)*
- Electron density, potential
  - *total-, valence-, difference-, spin-densities,* 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*
  - *NMR chemical shifts*

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)

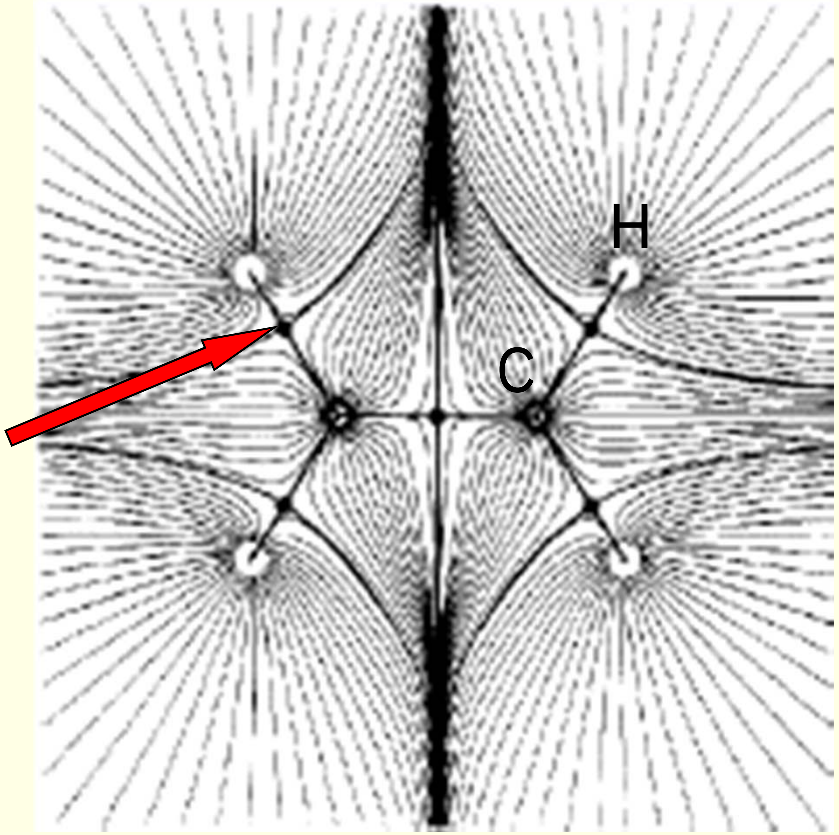
Electron density of C_2H_4





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)

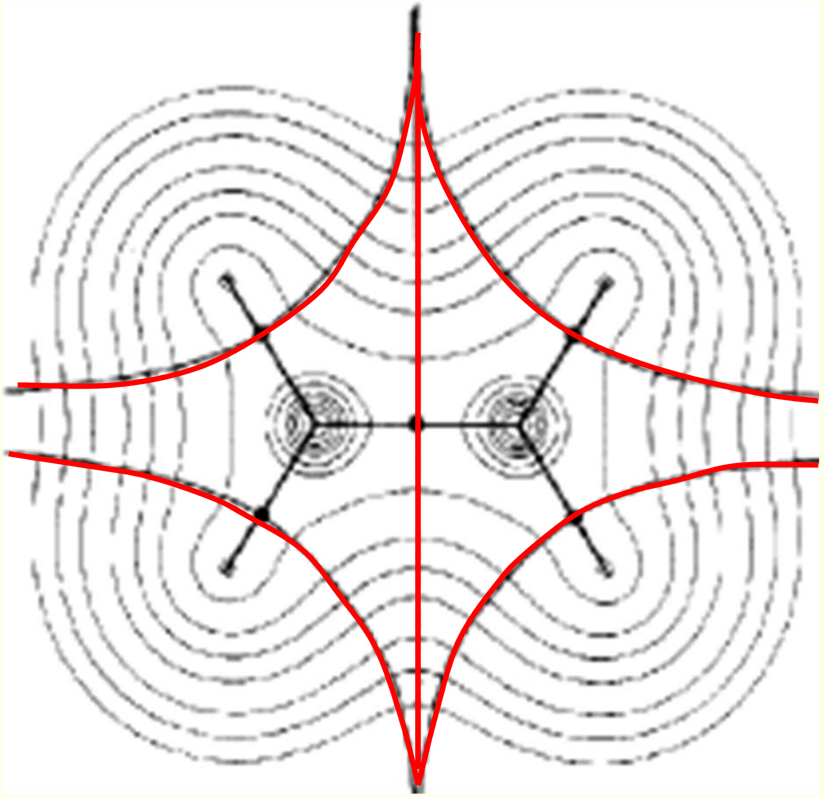


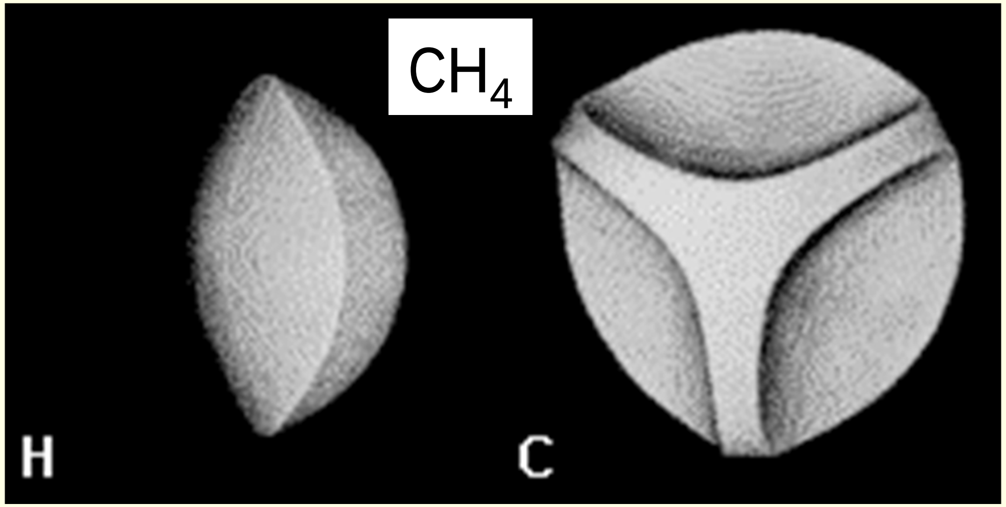
trajectories of constant $\nabla\rho$
originating at CPs in C_2H_4

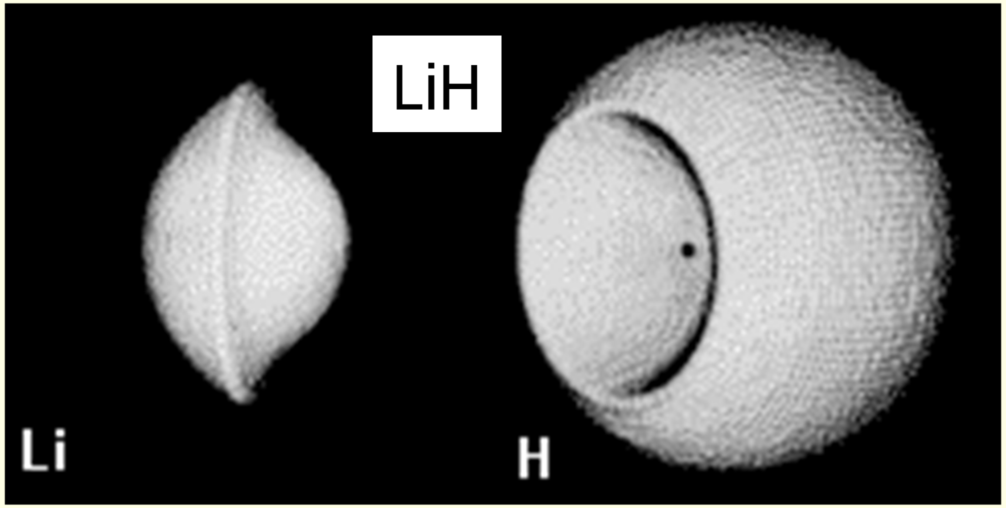
AIM-III

- “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

- example of BN/Ni with “difference” to free atoms,
- workfunction shift
- Bader analysis of some inorganic compounds:

	ρ(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)

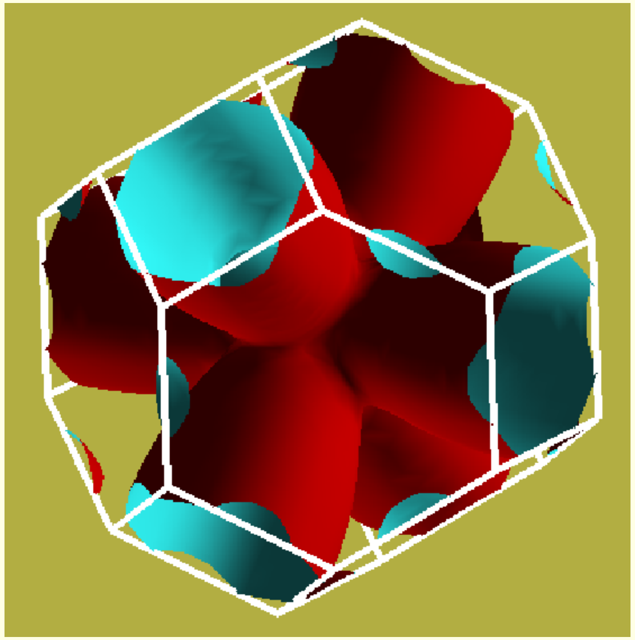
:PC x, y, z, λ_1, λ_2, λ_3, ch, laplacian, rho

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, hexagonal, and tetragonal cells*
  - *Phonons via supercells*
    - interface to PHONON (K.Parlinski) – bands, DOS, thermodynamics, neutrons
    - interface to PHONOPY (A. Togo)
      - http://www.wien2k.at/reg_user/unsupported
- 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)

Fermi surfaces

- **`xcrysden --wien_fermisurface tin.struct`**
  - choose a good k-mesh (eg. 10000 points); (DON'T CHANGE to UNIT 5 !!!)
  - plot the FS for all bands which cross E_F and compare to band structure



- *for 2D plots there is also a WIEN2k-tool „fsgen“ (see UG)*
- *SKEAF (www.wien2k.at/reg_users/unsupported): quantum oszillations*

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, hexagonal, and tetragonal cells*
  - *Phonons via supercells*
    - interface to PHONON (K.Parlinski) – bands, DOS, thermodynamics, neutrons
    - interface to PHONOPY (A. Togo)
      - http://www.wien2k.at/reg_user/unsupported
- 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)

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)

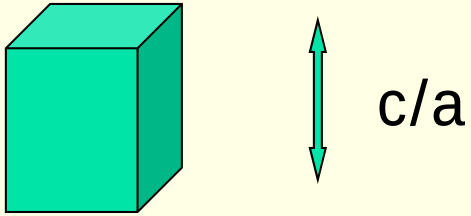
Structural optimizations:

- 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” (with more 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*

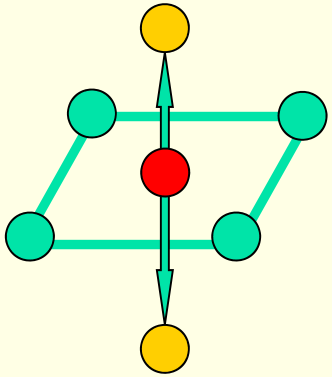
  - *combinations of volume and c/a are possible: 2Doptimize*
    - “x optimize” always uses case_initial.struct (if present)
    - do a “volume” optimization to create case_vol_xx.struct files
    - copy the respective case_vol_xx.struct file to case_initial.struct
    - x optimize with “c/a” for this 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 ...

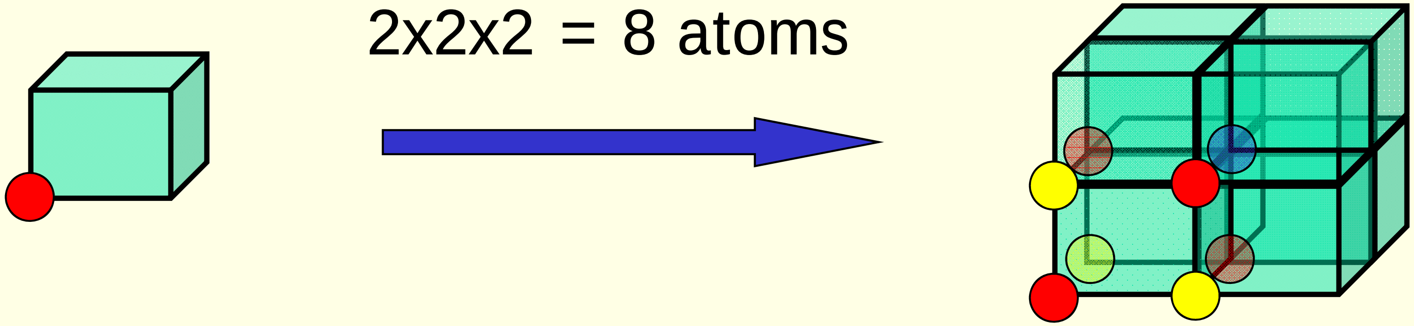


  - *„Jahn-Teller“ distortion:*
    - when you start with a perfect octahedra, you will never get any distortion
    - ➔start with slightly distorted positions



Supercells

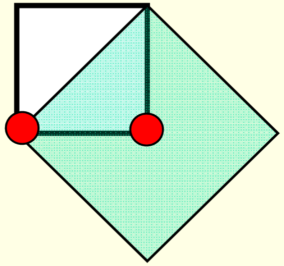
2x2x2 = 8 atoms



(0,0,0)	P→ 8 atoms	(0,0,0)	(.5,0,0) (0,.5,0) (0,0,.5)	(.5,.5,0) (.5,0,.5) (0,.5,.5)	(.5,.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)

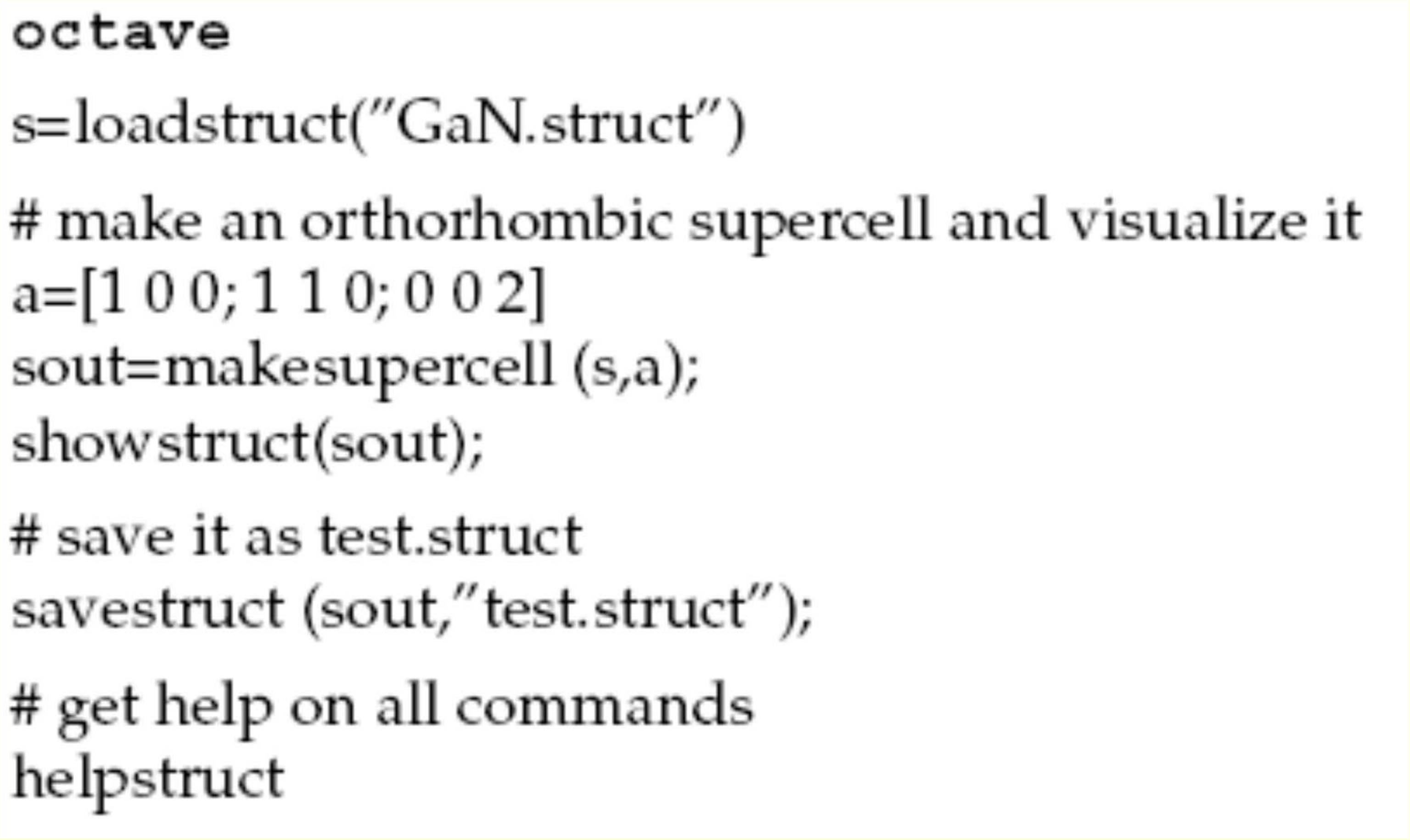


Supercells

- 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) or*
  - *label at least 1 atom (core-holes, specific magnetic order; change “Fe” to “Fe1”; this tells the symmetry-programs that Fe1 is NOT a Fe atom!!)*
- At present „supercell“ works only along unit-cell axes!!!

Structeditor (by R.Laskowski)

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

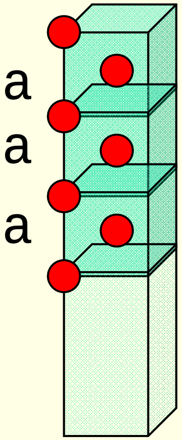


```
octave
s=loadstruct("GaN.struct")
# make an orthorhombic supercell and visualize it
a=[1 0 0; 1 1 0; 0 0 2]
sout=makesupercell (s,a);
showstruct(sout);
# save it as test.struct
savestruct (sout,"test.struct");
# get help on all commands
helpstruct
```

Surfaces

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

bcc (001) 7 layers:



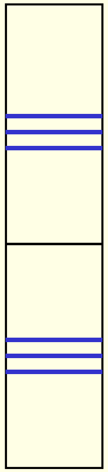
(0 0 6z)
(.5 .5 5z)
(0 0 4z)
(.5 .5 3z)
(0 0 2z)
(.5 .5 z)
(0 0 0)

shift to
➔
inversion

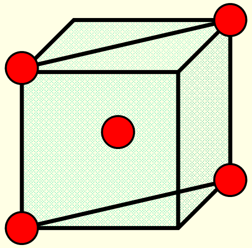
(.5 .5 +/-3z)
(0 0 +/-2z)
(.5 .5 +/-z)
(0 0 0)

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

z= a/2c

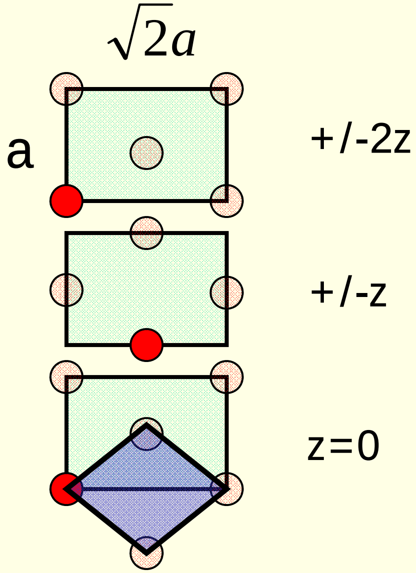


bcc (110):



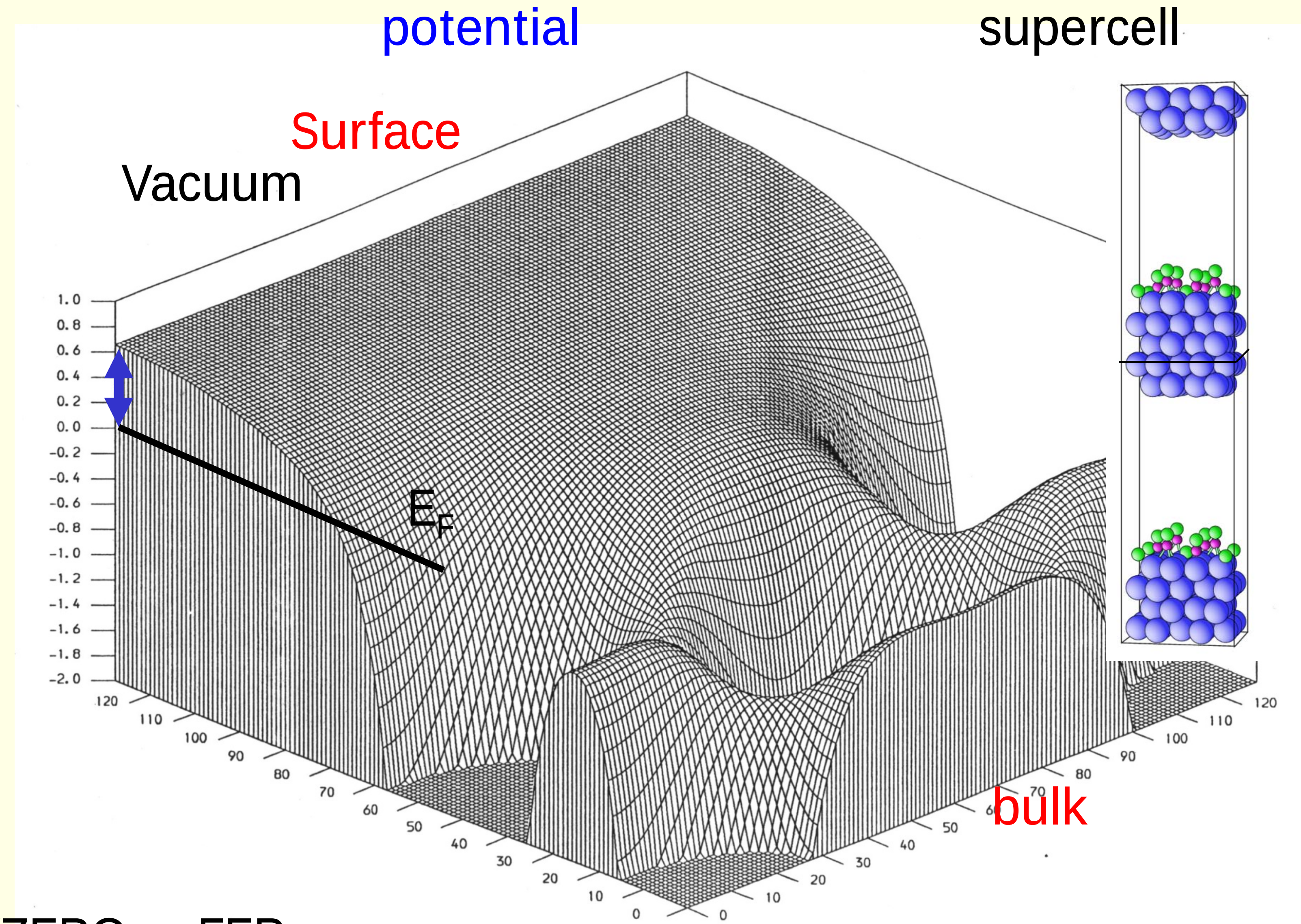
orthorhombic CXY-lattice: a, $\sqrt{2}a$, c

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



Work function

Work function



WF= :VZERO - :FER (check convergence with vacuum)

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

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

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

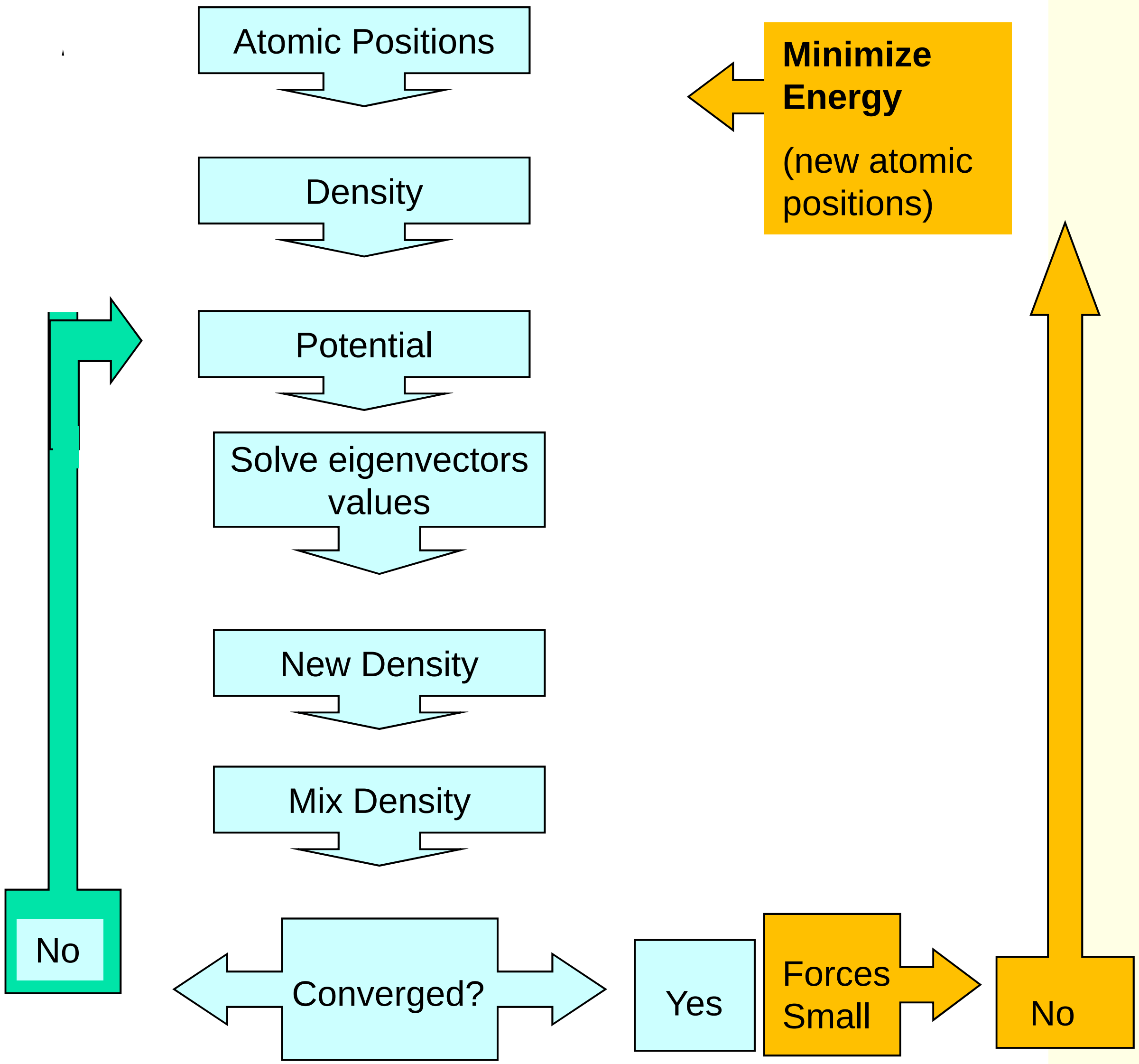
$$F_{val}^{\alpha} = \int_{\alpha} V_{eff}(r)\nabla_{\alpha}\rho_{val}(r)\, d\vec{r} + \sum_{k,i} n_i \sum_{K,K'} c_i^{*}(K')c_i(K) \times$$

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

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 “-fc”:
  - *run_lapw –fc 1.0 (mRy/bohr)*
    - grep :fgl002 case.scf
      - 200. partial
      - -130. partial
      - 140. partial
      - 135 partial only $F_{HF} + F_{core}$
      - 120 partial
      - 122 partial forces converging
      - 121 partial → changes “TOT” to “FOR” in case.in2
      - -12.3 **total** $F_{HF} + F_{core} + F_{val}$, only this last number is correct
- Forces are useful for
  - *structural optimization (of internal parameters)*
  - *phonons*

Structure optimization (atomic positions)


Atomic Positions

Density

Potential

Solve eigenvectors values

New Density

Mix Density

Converged?

No

Yes

Minimize Energy

(new atomic positions)

Forces Small

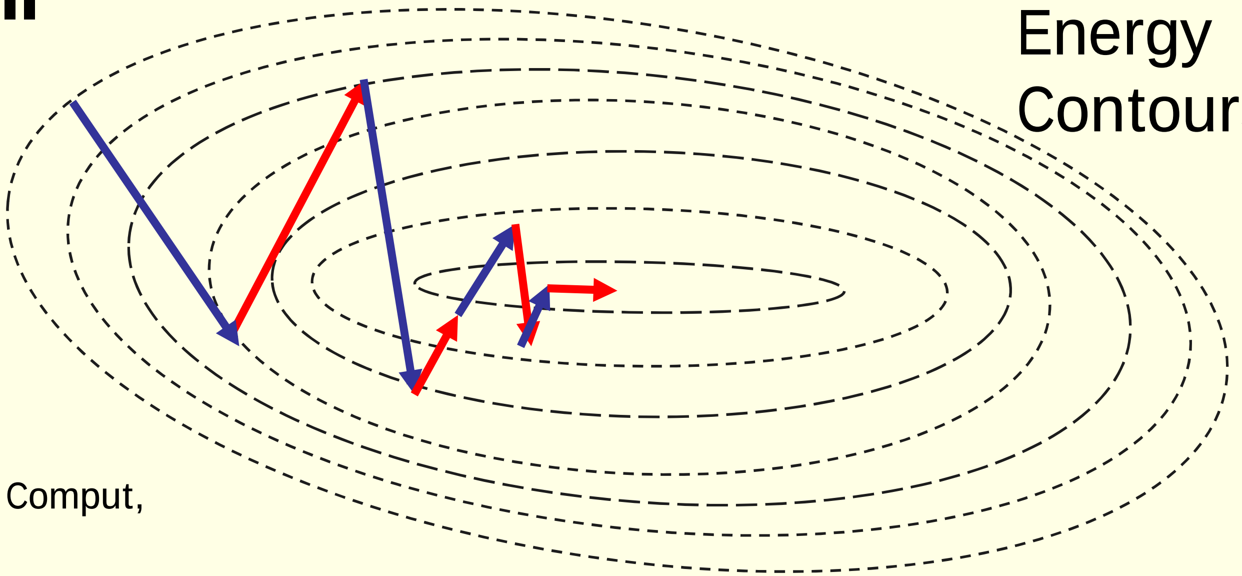
No

Traditional way:

- Inner loop: obtain fixed-point for given atom positions
- Outer loop: optimize atomic positions

Current algorithms

- Calculate SCF mapping, time T_0
- Broyden expansion for fixed-point problem, self-consistent density, N_{SCF} iterations
- BFGS is most common for optimizing the atomic positions (Energy), N_{BFGS}
- Time scales as $N_{SCF} * N_{BFGS} * T_0$



each step is a **full** scf calculation producing **accurate** forces

L.D.Marks: J. Chem. Theory Comput, DOI: 10.1021/ct4001685

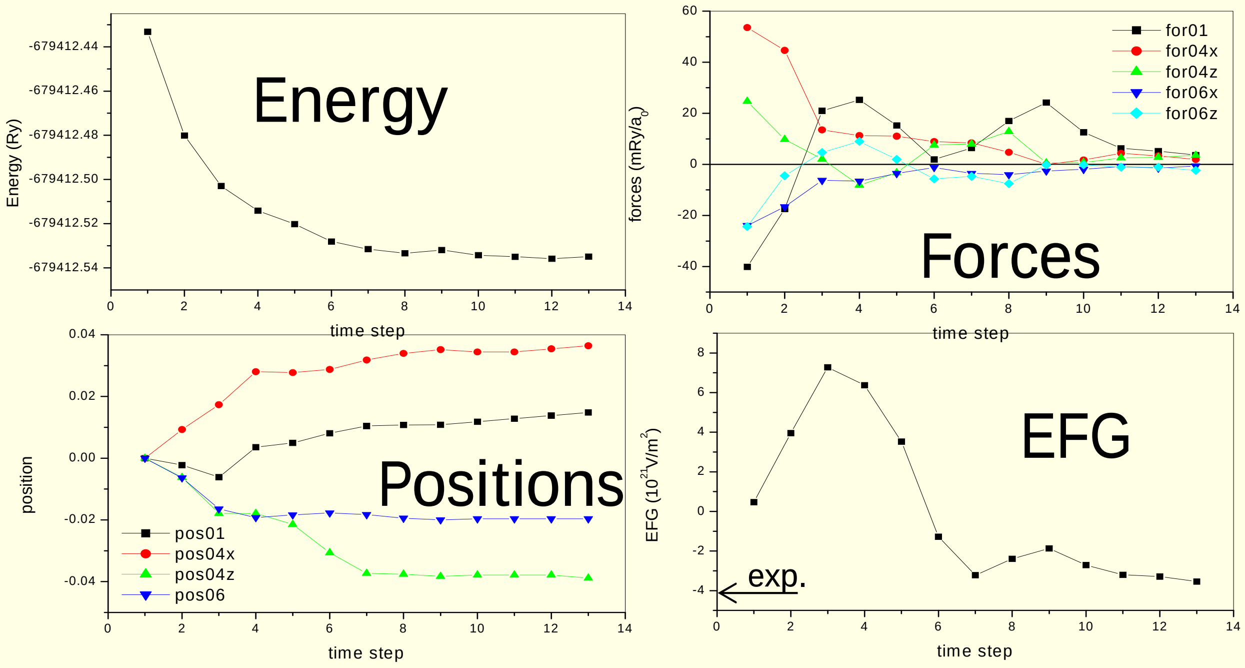
Structural optimization of internal parameters using “PORT”

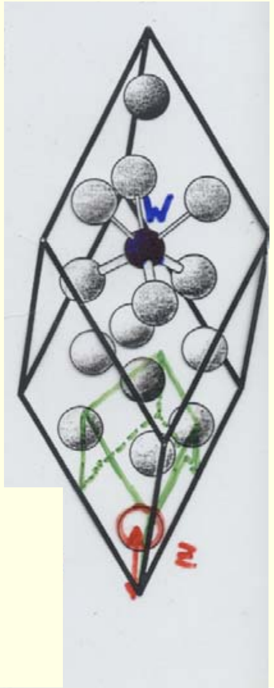
- /home/pblaha/tio2> min_lapw [-p -it -sp] [-j “run -fc 1 -p -it”] [-NI]
  - *performs scf-cycle for fixed positions*
  - *get forces and move atoms along forces (building an approximate Hessian) and writing a new case.struct file*
  - *extrapolate density (case.clmsum)*
  - *perform next scf cycle and loop until forces are below „tolf“*
  - *CONTROL FILES:*
    - .minstop   stop after next structure change
- tio2.inM (generated automatically by “pairhess” at first call of min_lapw)
    - 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 !)*
    - grep :ENE case.scf_mini
    - grep :FGLxxx case.scf_mini   (:POSxxx)

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 and construct approx. Hessian)
  - 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: $Bi_{15}W$)

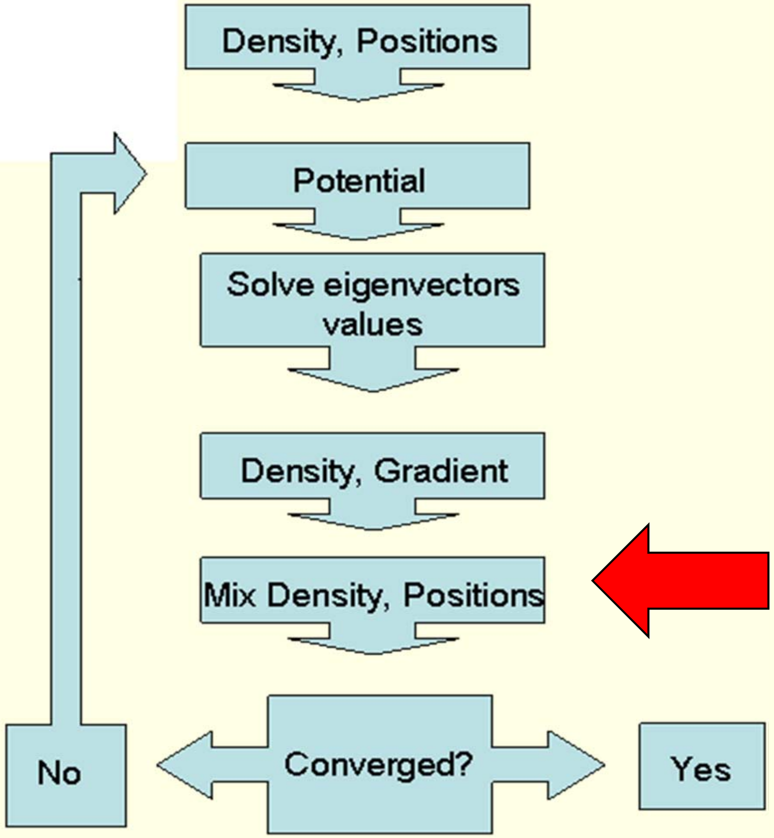




Alternative method: **Fused Loop**

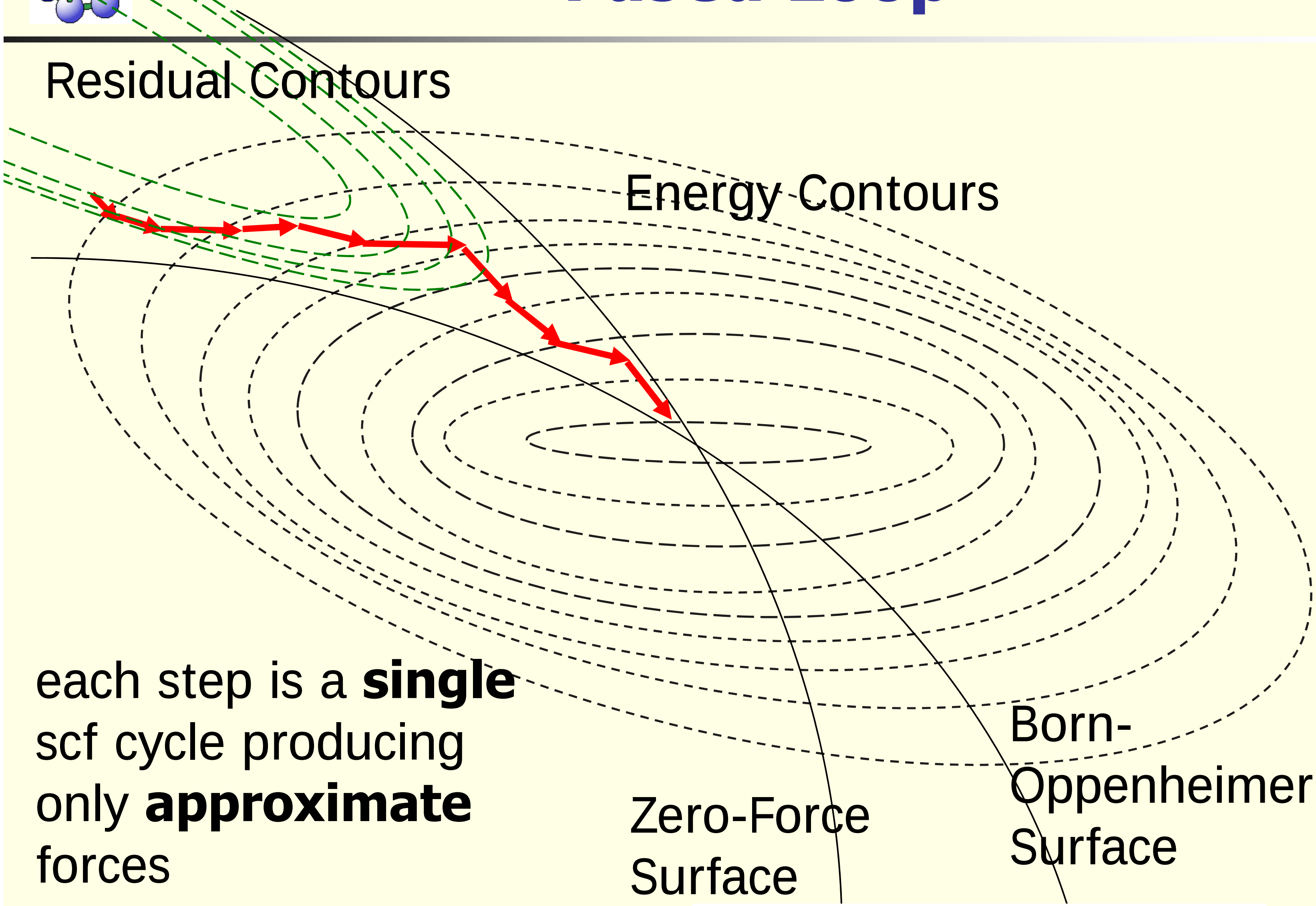
- Treat the **density** and **atomic positions** ***all*** at the same time.

- No restrictions to “special” cases, general algorithm has to work for insulators, metals, semiconductors, surfaces, defects, hybrids....
- Few to no user adjustable parameters



L.D. Marks, J. Chem. Theory Comput, DOI: 10.1021/ct4001685

Fused Loop



Residual Contours

Energy Contours

each step is a **single** scf cycle producing only **approximate** forces

Zero-Force Surface

Born-Oppenheimer Surface

J. Chem. Theory Comput, DOI: 10.1021/ct4001685

Broyden Fixed-Point Methods

- Solve $(\rho(r,x)-F(\rho(r,x)),G)=0$
- $s_k = (\rho,x)_{k+1} - (\rho,x)_k; y_k = (F(\rho,x),G)_{k+1} - (F(\rho,x),G)_k$
- Broyden's "Good Method"

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)s_k^T}{s_k^T s_k} \qquad H_{k+1} = H_k + \frac{(s_k - H_k y_k)s_k^T}{s_k^T y_k}$$

- Broyden's "Bad Method"

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)y_k^T}{y_k^T y_k}$$

C.G. Broyden, A Class of Methods for Solving Nonlinear Simultaneous Equations, Mathematics of Computation, 19 (1965) 577-593.

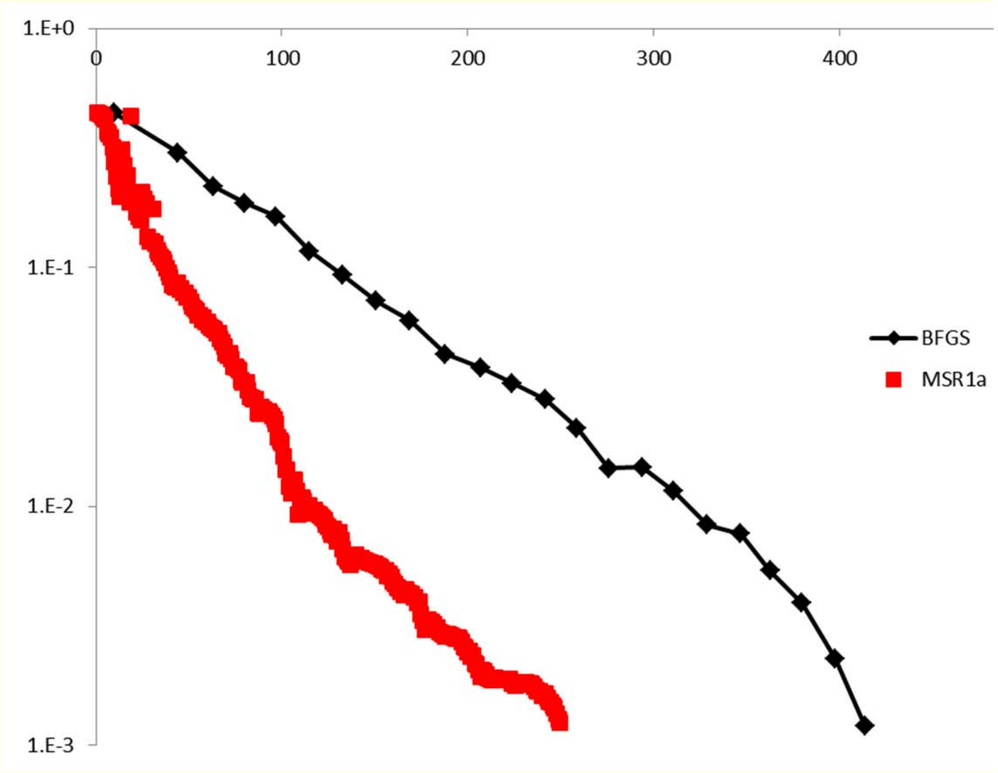
- Generalizable to multisecant method (better,

L.D.Marks: J. Chem. Theory Comput, DOI: 10.1021/ct4001685

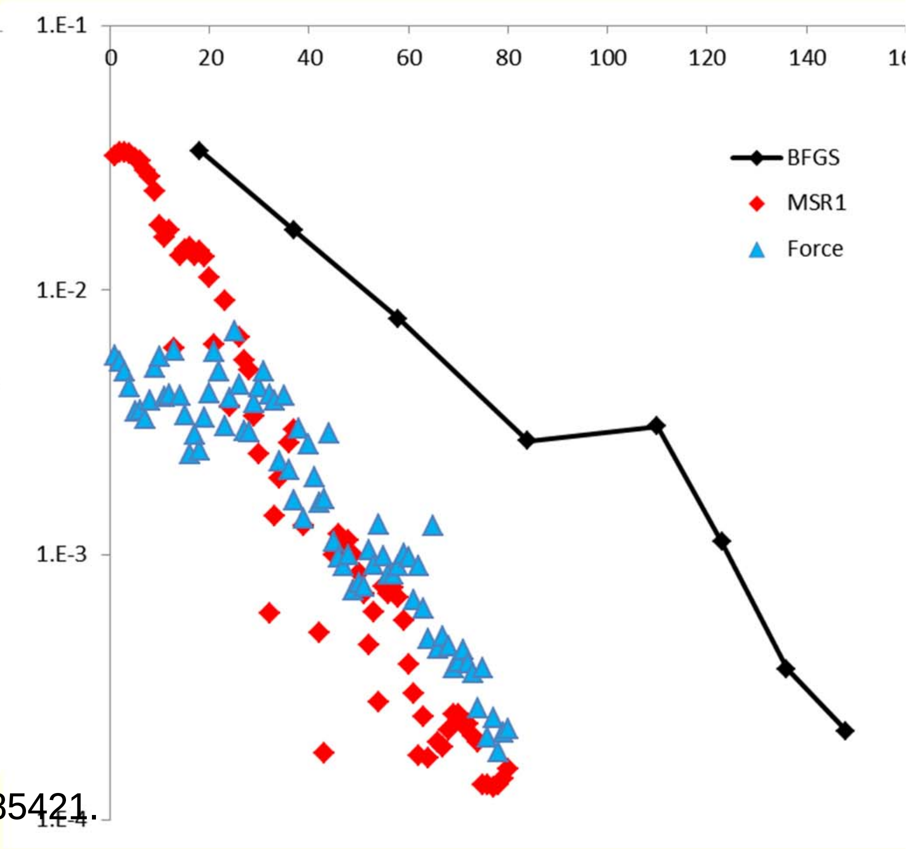
Comparison of the 2 methods

Larger Problems:

52 atoms, MgO (111)+H_2O 108 atoms AlFe



J. Ciston, A. Subramanian, L.D. Marks, PhRvB, 79 (2009) 085421.



Lyudmila V. Dobysheva (2011)

J. Chem. Theory
Comput, DOI:
10.1021/ct4001685

Structural optimization of internal parameters using “**MSR1a**”

- edit case.inm and set „**MSR1a**“
- run_lapw -fc 1.0 -cc 0.001 -ec 0.0001 [-it -noHinv -p]

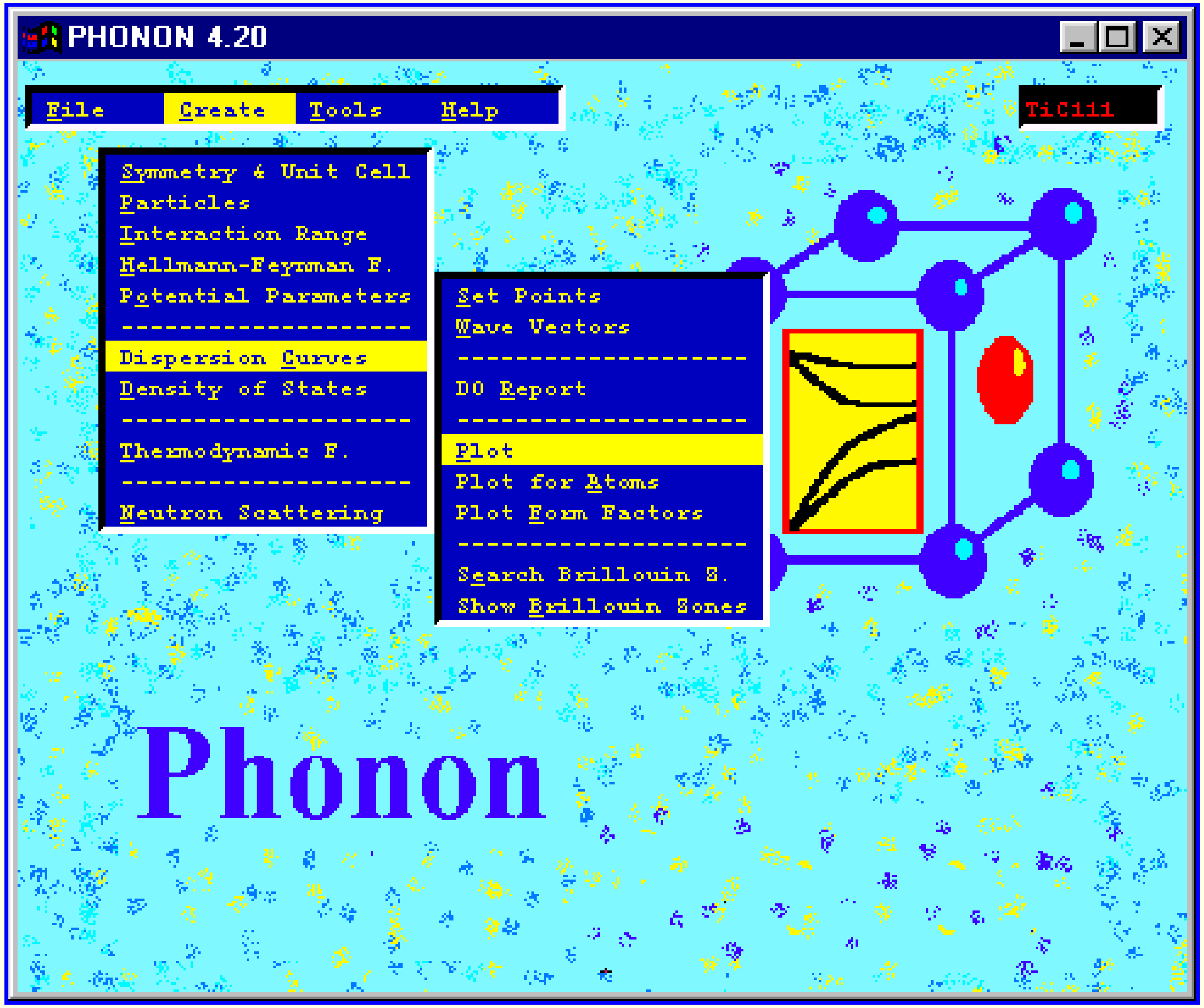
- This runs ONE big scf-calculations optimizing the density and the positions (forces towards zero) simultaneously (may need hundreds of iterations).

- Monitor: :ENE and :FR (av. and max forces, movements)

- it continues until all :FR quantities are below „tolf“ (case.inM) and switches then automatically to MSR1 for a final charge optimization (with fixed positions).

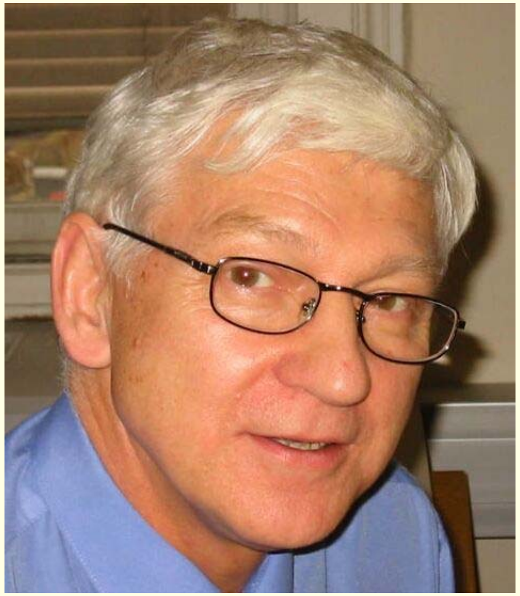
- quite efficient, recommended method, still under development by L.Marks (Northwestern Univ).

Calculations of Phonons: The Direct Method



WIEN2k + Phonon

Copyright by K.Parlinski



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

alternatively use A.Togo`s PHONOPY code +Wien2k-interface
(see www.wien2k.at/unsupported)

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

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

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

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

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

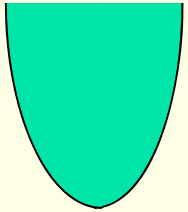
$$\mathbf{F}(\mathbf{n},\mu) = -\partial E/\partial\mathbf{R}(\mathbf{n},\mu)$$

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.

$$V = \frac{1}{2}kx^2$$



n,m: cells

μ,ν: atoms

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.

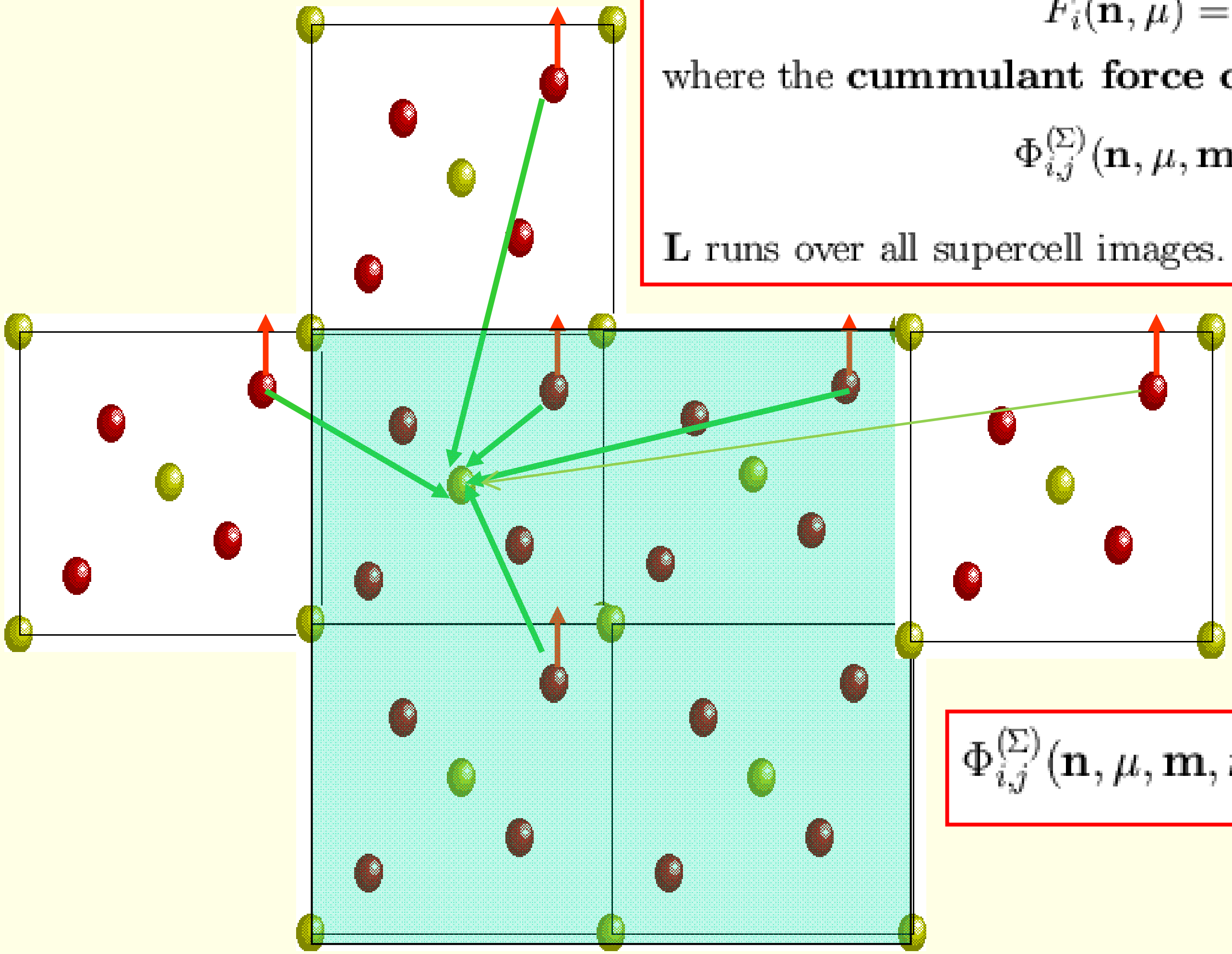
or

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

where the **cummulant force constant** is

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

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 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):

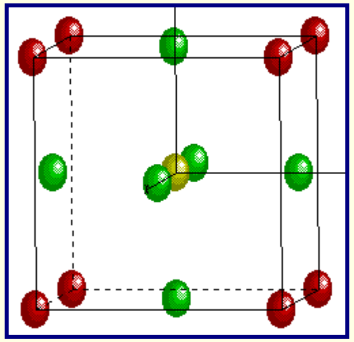
$$exp\{-2\pi i\mathbf{k}_s\cdot\mathbf{L}\} = 1$$

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.

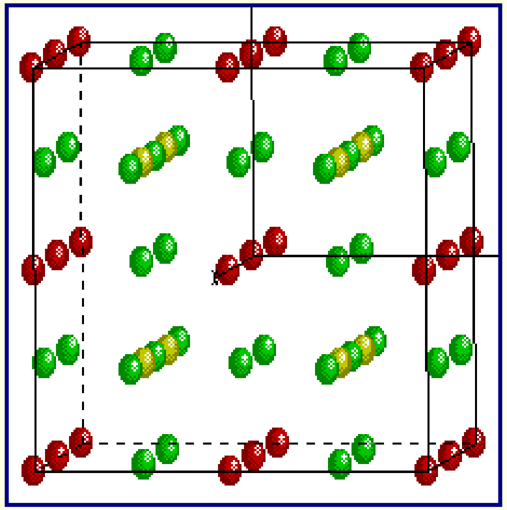
Exact wave vectors

1x1x1



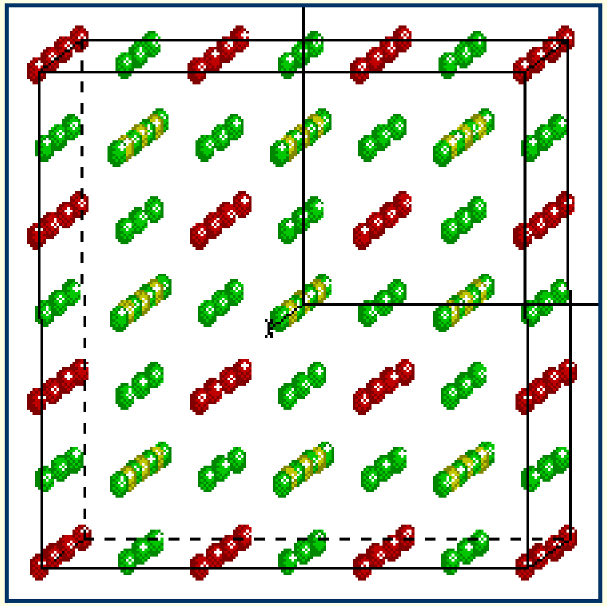
Exact: Γ

2x2x2

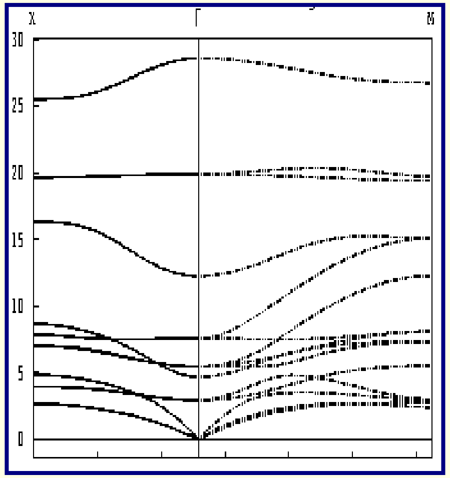


Exact: Γ, X, M, R

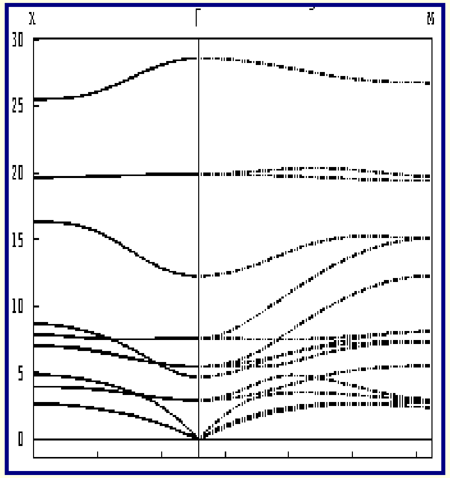
3x3x3



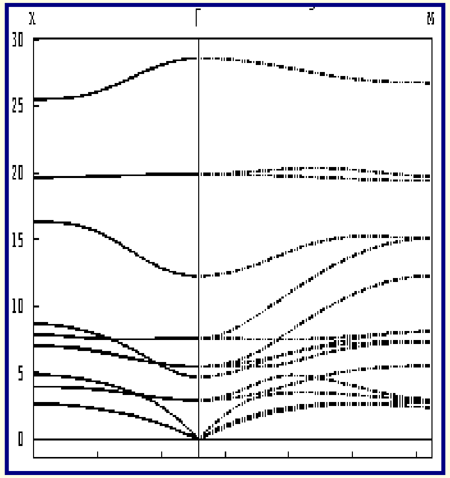
Exact: Γ



X Γ M



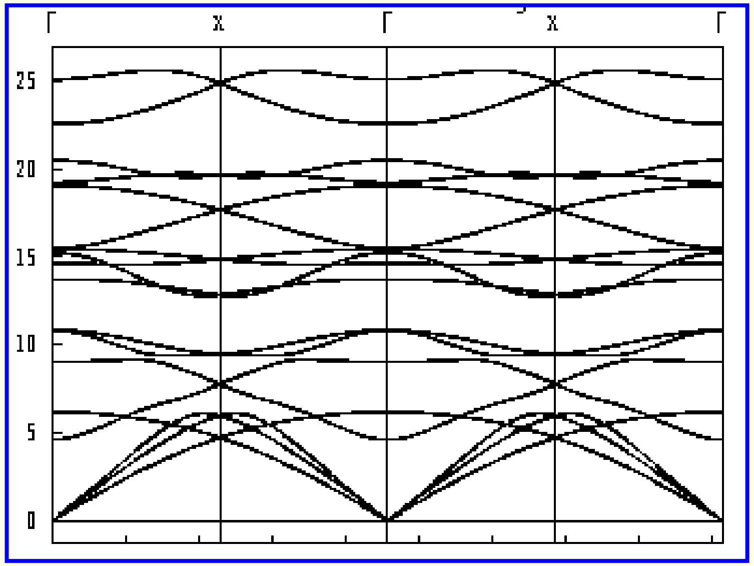
↓ Γ ↓



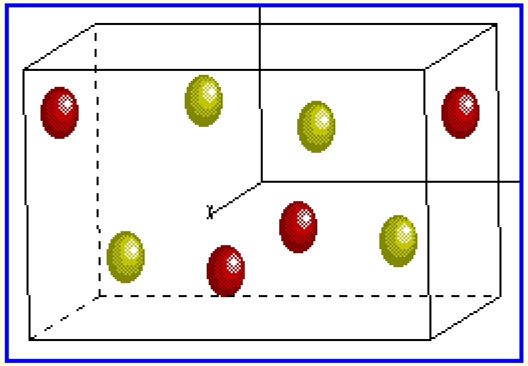
Phonon dispersions + density of states

Frequency

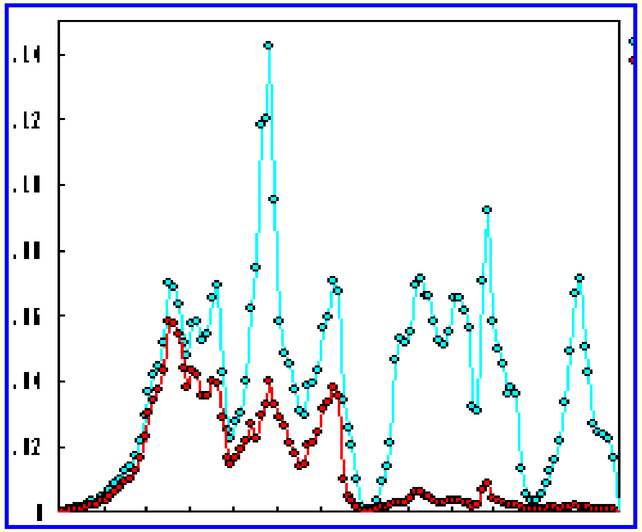
ω



GeO$_2$ P4_2/mnm

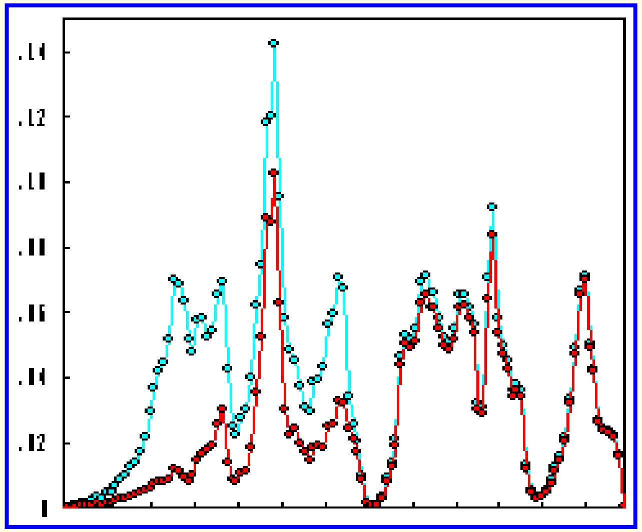


Total + Germanium



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:

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

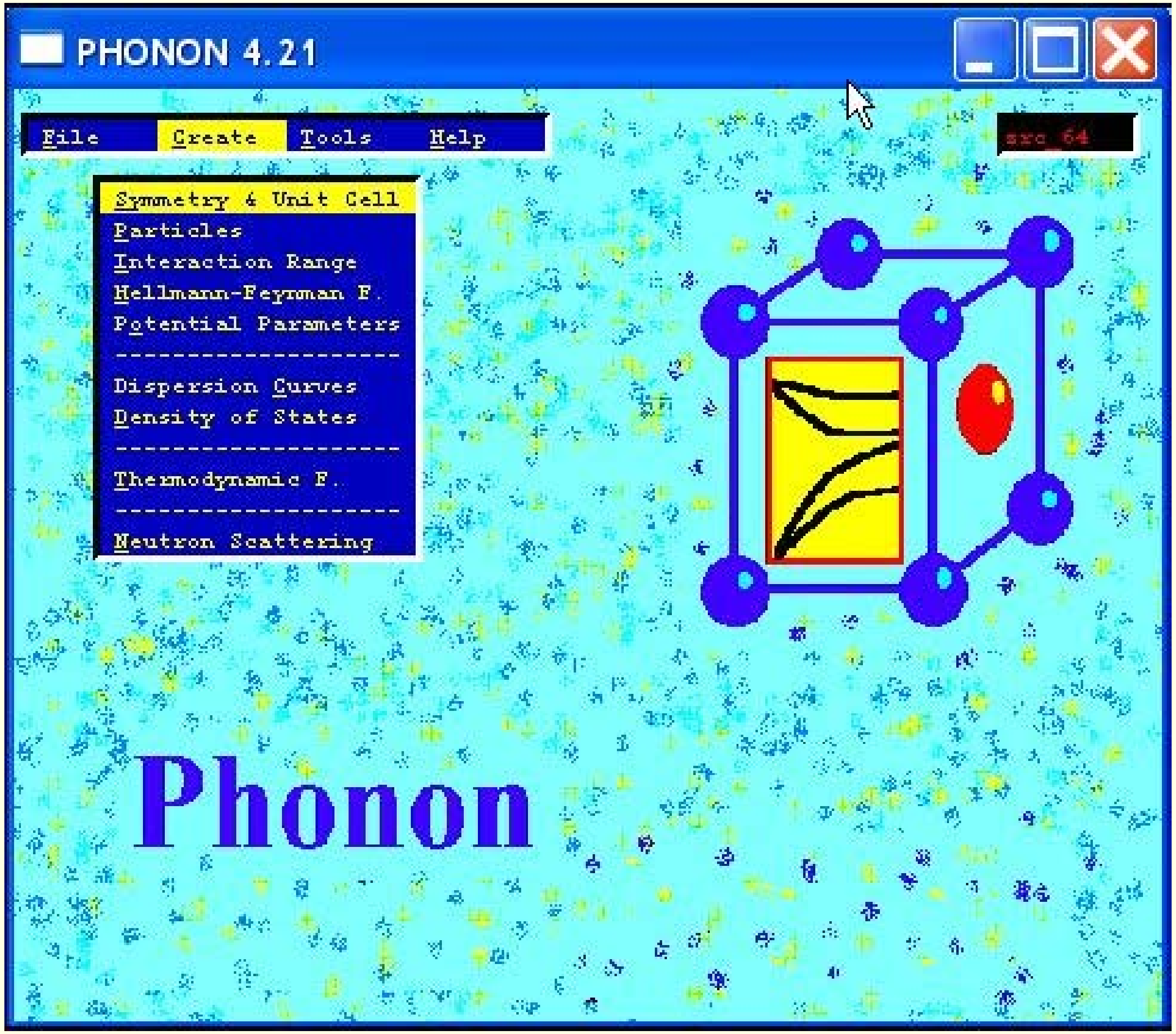
Thermal displacements:

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

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

PHONON-I

- PHONON
  - *by K.Parlinski (Crakow)*
  - *Linux or 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*

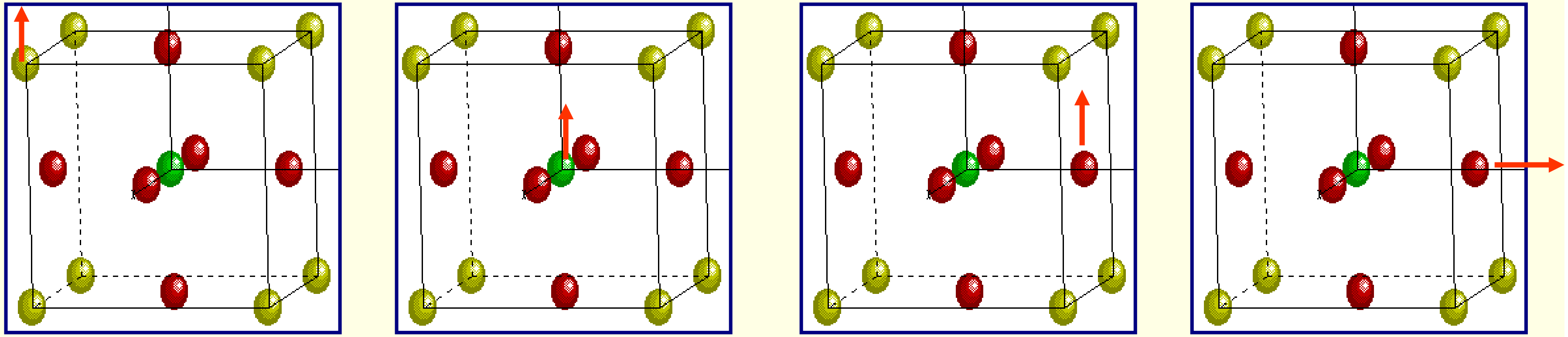


- Define your spacegroup
- Define all atoms

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

Phonons:

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



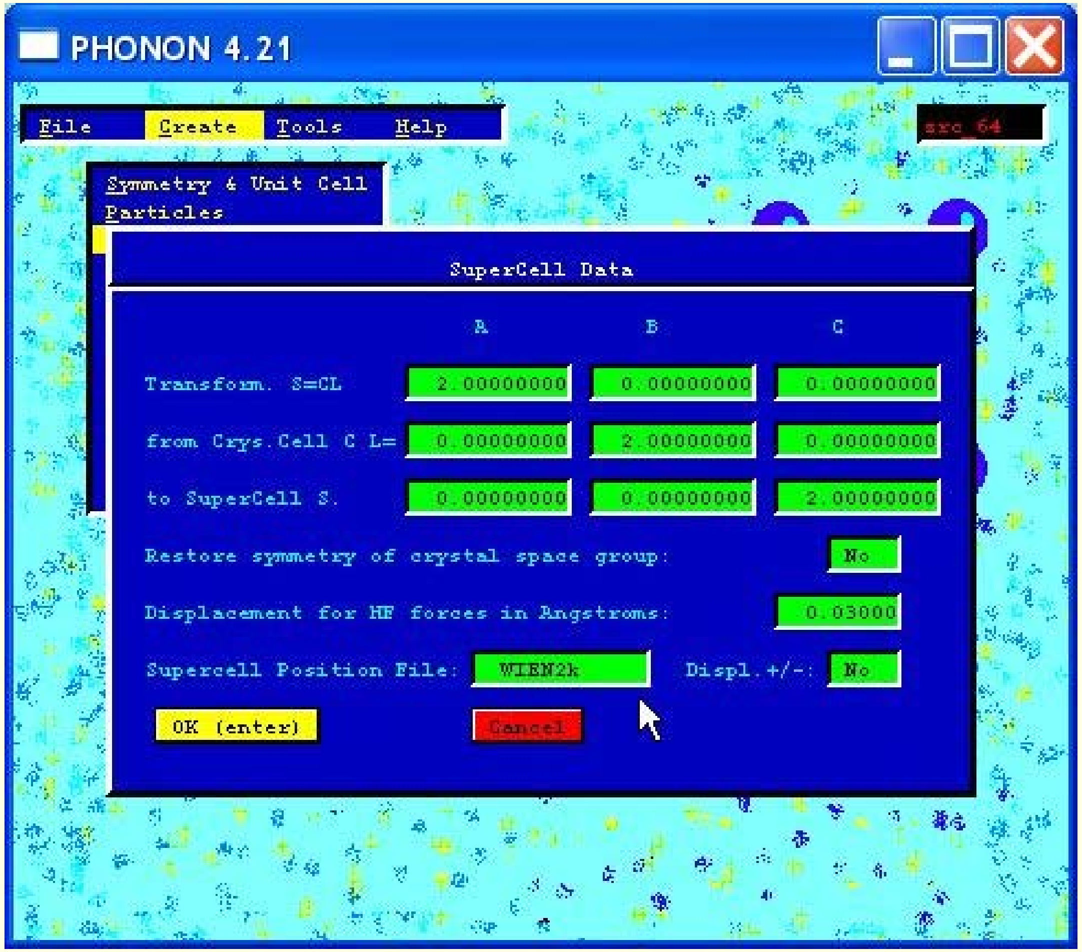
(Displacement pattern for cubic perovskite)

- *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



	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

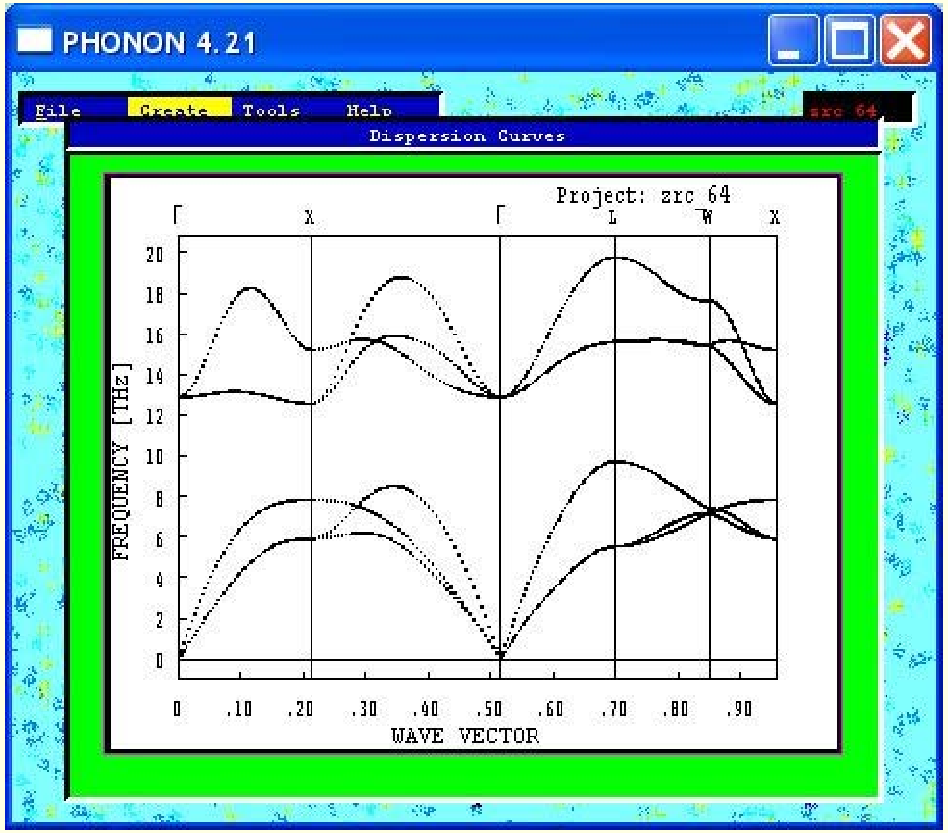
Displacement for HF forces in Angstroms: 0.03000

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

- 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

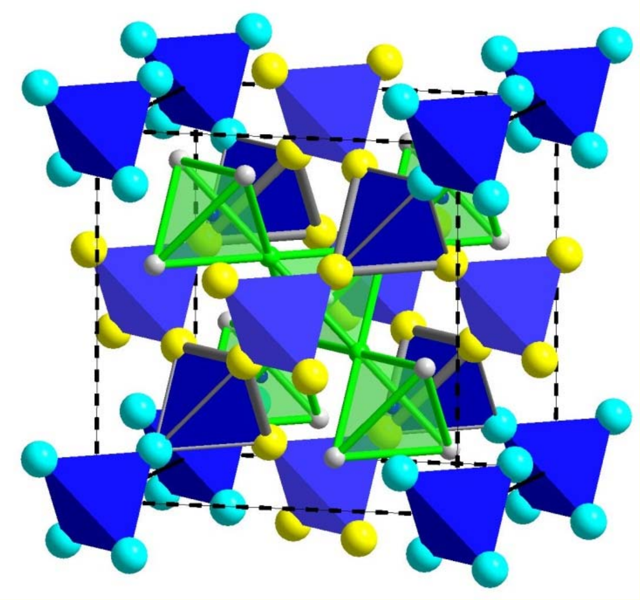
- analyze_phonon_lapw
  - *reads the forces of the scf runs*
  - *generates „Hellman-Feynman“ file case.dat and a „symmetrized HF-file case.dsy (when you have displacements in both directions)*
    - check quality of forces:
    - sum F_x should be small (0)
    - abs(F_x) should be similar for +/- displacements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

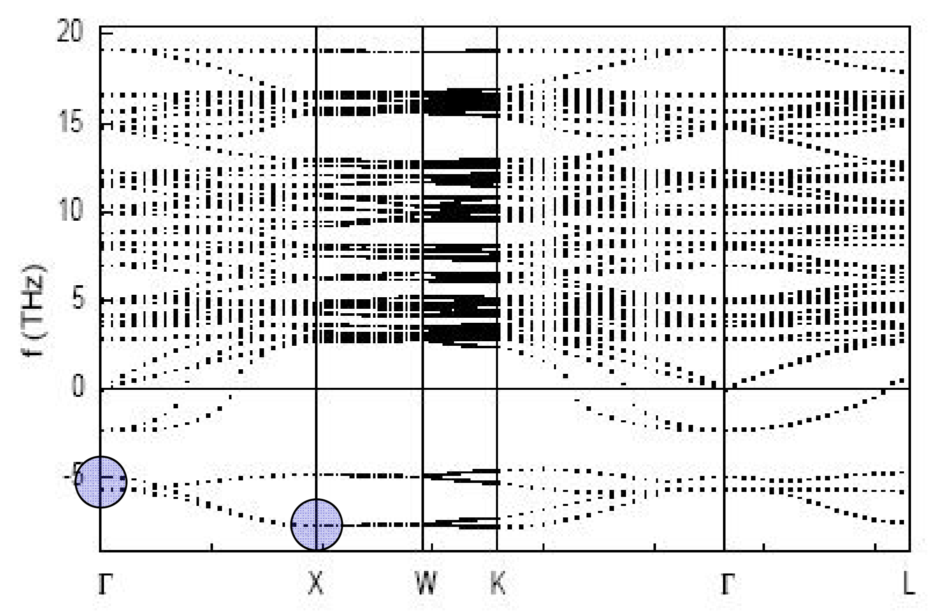


Applications:

- phonon frequencies (compare with IR, raman, neutrons)
- identify dynamically unstable structures, describe phase transitions, find more stable (low T) phases.

Pyrochlore structure of $Y_2Nb_2O_7$: strong phonon instabilities → phase transition





Wannier functions

from E. Assmann, TU Vienna

needs **wannier90** !

Wannier90: A Tool for Obtaining Maximally-Localised Wannier Functions
A. A. Mostofi, J. R. Yates, Y.-S. Lee, I. Souza, D. Vanderbilt and N. Marzari
Comput. Phys. Commun. **178**, 685 (2008) [http://wannier.org]

Bloch functions - Wannier functions

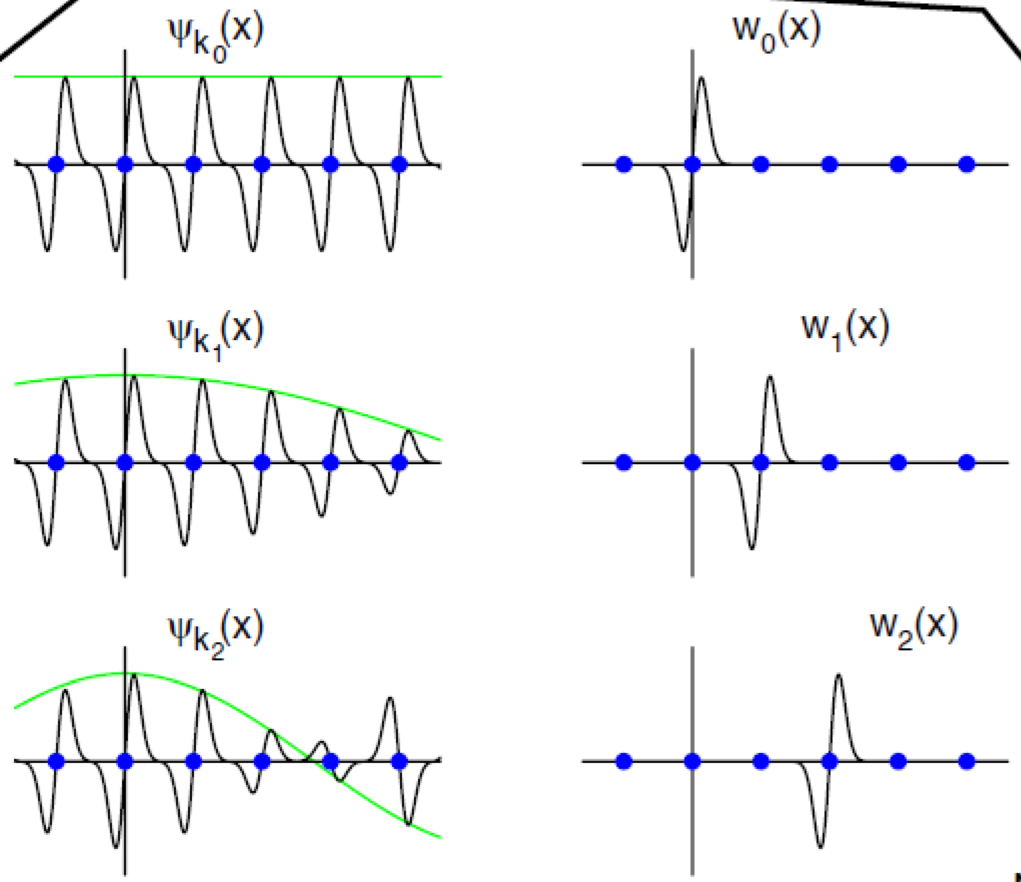
Bloch functions

$$\psi_{n\mathbf{k}}(\mathbf{r}) = u_{n\mathbf{k}}(\mathbf{r})e^{i\mathbf{k}\cdot\mathbf{r}},$$

Wannier functions

$$|\mathbf{R}n\rangle = \frac{V}{(2\pi)^3}\int_{\mathrm{BZ}} d\mathbf{k}\, e^{-i\mathbf{k}\cdot\mathbf{R}}|\psi_{n\mathbf{k}}\rangle.$$

Indexed by the wave vector



Indexed by the lattice vector in real space

Both sets: complete and orthonormal

Marzari et al.:
Rev. Mod. Phys. (2012)

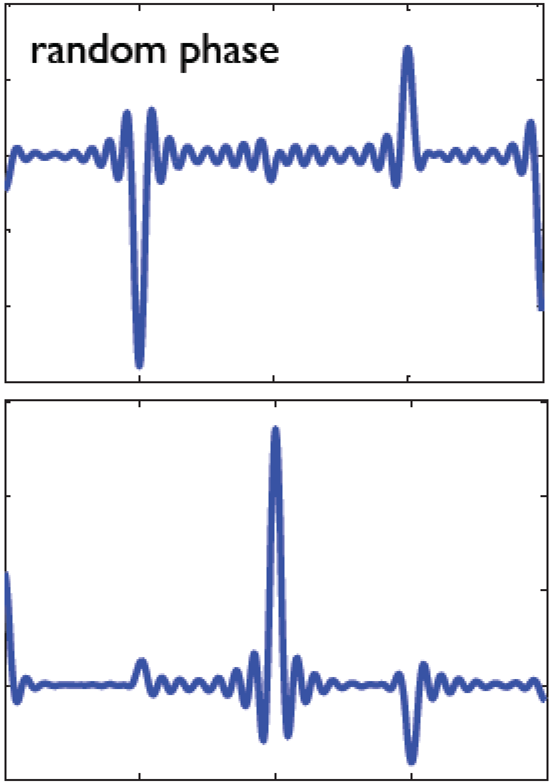
Maximally Localized Wannier Functions

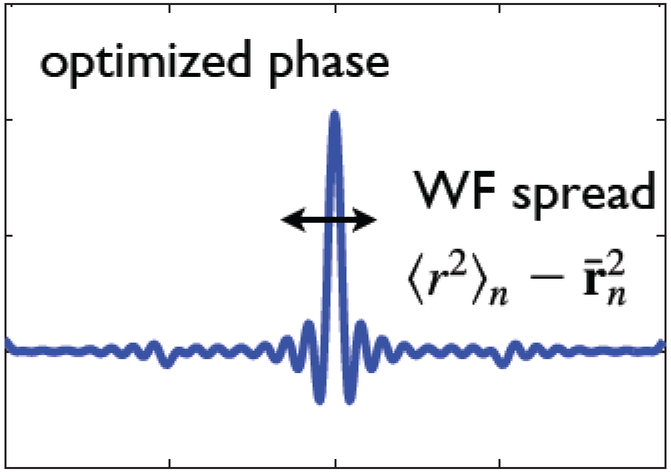
Bloch functions (more precisely):

$$\psi_{n\mathbf{k}}(\mathbf{r}) = u_{n\mathbf{k}}(\mathbf{r})\ e^{i\mathbf{k}\cdot\mathbf{r}}\ e^{i\phi_{n\mathbf{k}}}$$

gauge freedom

(does not change the physical description of the system)



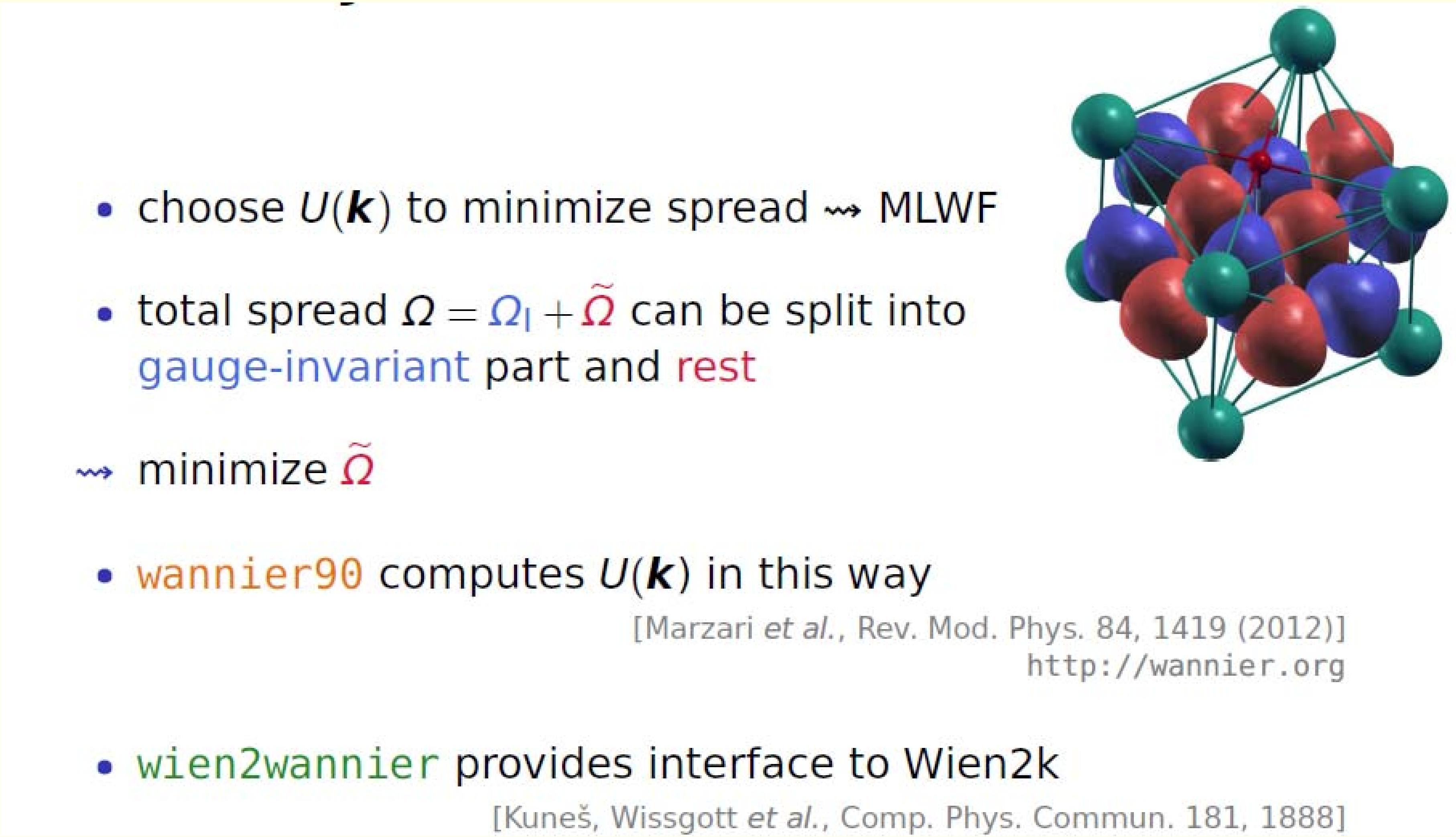


$$|w\,n\boldsymbol{R}\rangle = \frac{V}{(2\pi)}\int_{\mathrm{BZ}} d\boldsymbol{k} \sum_m e^{i\boldsymbol{k}\boldsymbol{R}}\, U_{mn}(\boldsymbol{k})\, |\psi\, m\boldsymbol{k}\rangle$$

$$\Omega = \sum_n [\langle \mathbf{0}n|r^2|\mathbf{0}n\rangle - \langle \mathbf{0}n|\mathbf{r}|\mathbf{0}n\rangle^2] = \sum_n [\langle r^2\rangle_n - \bar{\mathbf{r}}_n^2]$$

choose U(k) to minimize Ω

maximally localized wannier functions



- choose $U(\boldsymbol{k})$ to minimize spread ⇝ MLWF
- total spread $\Omega = \Omega_{\mathrm{I}} + \widetilde{\Omega}$ can be split into gauge-invariant part and rest

⇝ minimize $\widetilde{\Omega}$

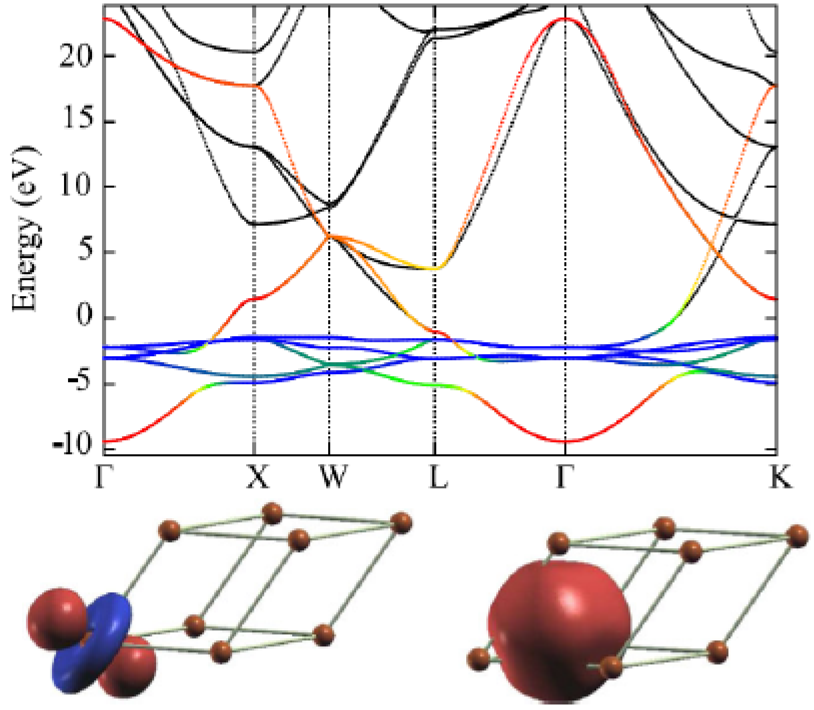
- `wannier90` computes $U(\boldsymbol{k})$ in this way

[Marzari *et al.*, Rev. Mod. Phys. 84, 1419 (2012)]
`http://wannier.org`

- `wien2wannier` provides interface to Wien2k

[Kuneš, Wissgott *et al.*, Comp. Phys. Commun. 181, 1888]

Disentanglement



from Marzari *et al.*

fcc-Cu, 5 *d*-like WF,
2 interstitial *s*-like WF

```
num_bands = 12
num_wann  =  7
```

- other bands may cross target manifold

⇝ must select bands to Wannierize

  - $V(\boldsymbol{k})\ \big(J(\boldsymbol{k}) \times N\big)$

- selection determines Ω_I

⇝ minimize also Ω_I

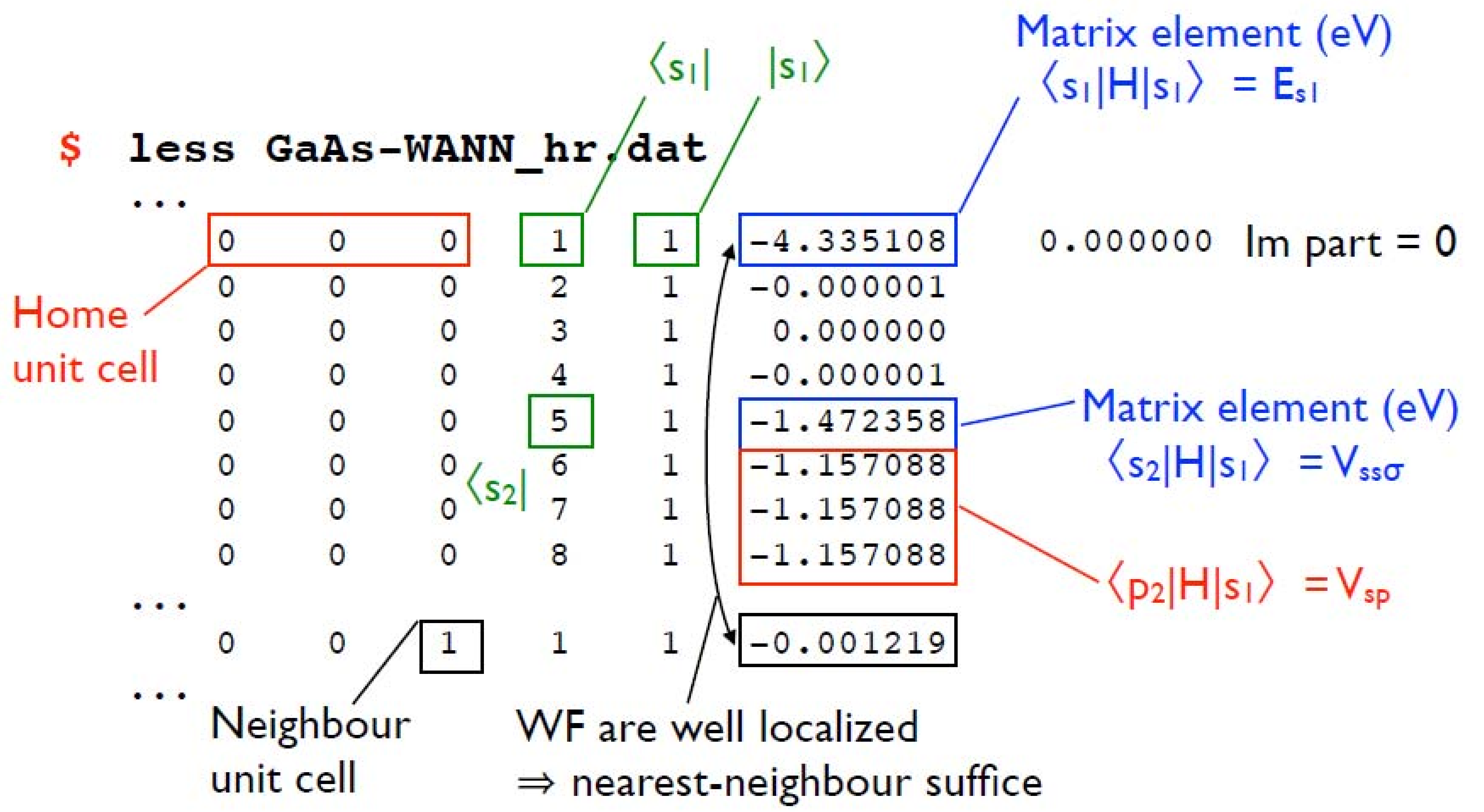
MLWF: applications

- analysis of chemical bonding
- electric polarization and orbital magnetization
  - ⇝ Oleg Rubel's talk
- Wannier interpolation $\mathcal{K} \to \mathcal{G}$

$$H(\boldsymbol{k})|_{\mathcal{K}} \xrightarrow{\mathcal{F}} H(\boldsymbol{R})|_{\mathcal{K}^{-1}} \xrightarrow{\mathcal{F}^{-1}} H(\boldsymbol{k})|_{\mathcal{G}}$$

- Wannier functions as basis functions
  - tight-binding model $H(\boldsymbol{k}) = U^{+}(\boldsymbol{k})\,\epsilon(\boldsymbol{k})\,U(\boldsymbol{k})$
  - ⇝ realistic dynamical mean-field theory (DMFT)

Wannier functions as tight-binding basis



```
$  less GaAs-WANN_hr.dat
   ...
      0      0      0      1      1   -4.335108      0.000000
      0      0      0      2      1   -0.000001
      0      0      0      3      1    0.000000
      0      0      0      4      1   -0.000001
      0      0      0      5      1   -1.472358
      0      0      0      6      1   -1.157088
      0      0      0      7      1   -1.157088
      0      0      0      8      1   -1.157088
   ...
      0      0      1      1      1   -0.001219
   ...
```

$\langle s_1|$ $|s_1\rangle$

$\langle s_2|$

Matrix element (eV) $\langle s_1|H|s_1\rangle = E_{s1}$

Im part = 0

Home unit cell

Matrix element (eV) $\langle s_2|H|s_1\rangle = V_{ss\sigma}$

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

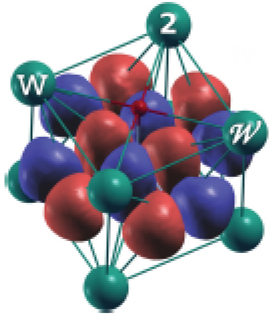
Neighbour unit cell

WF are well localized

⇒ nearest-neighbour suffice

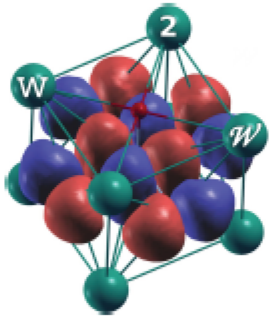
workflow using wien2wannier

- Regular SCF calculation
- Band structure plot



- Initialize wien2wannier (`init_w2w`):
  - select bands, init. projections, # of WF (`case.inwf` file)
  - projected band structure "bands_plot_project" (`case.win` file)
  - additional options related to entanglement (`case.win` file)
- Compute overlap matrix element M_{mn}, A_{mn} (`x w2w`)

- Perform Wannierization (`x wannier90`):
  - position of Wannier centers and spreads (`case.wout` file)
  - Wannier hamiltonian (`case_hr.dat` file)



- Initialize plotting, select plotting range, r-mesh (`write_inwplot`)
- Evaluate WF on the r-mesh selected (`x wplot`)
- Convert the output of `wplot` into `xcrysden` format for plotting (`wplot2xsf`)

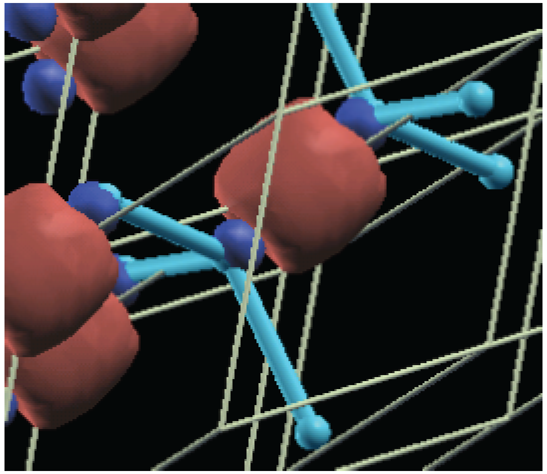
- Plot WF

[Wissgott, Kuneš *et al.* Phys. Rev. B 85, 205133]

Relation to Polarization

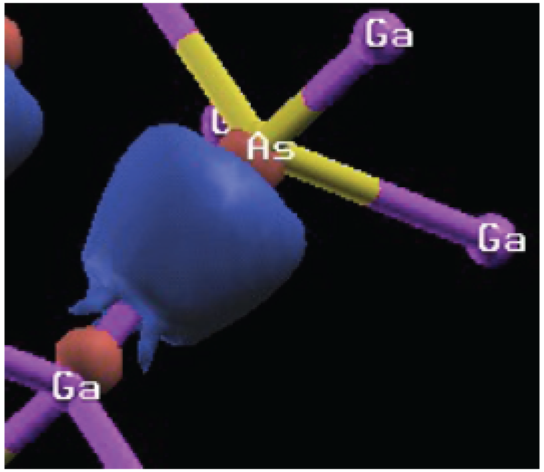
Bond-centered WF

Si



symmetric

GaAs



non-symmetric

$+\ Z_{\mathrm{As}}$

$-$ Wannier center q_e

$+\ Z_{\mathrm{Ga}}$

$$\mathbf{P} = \frac{e}{V}\left(\sum_{\tau} Z_{\tau}\mathbf{r}_{\tau} - \sum_{n}\mathbf{r}_{n}\right)$$

Ionic part

Electronic part

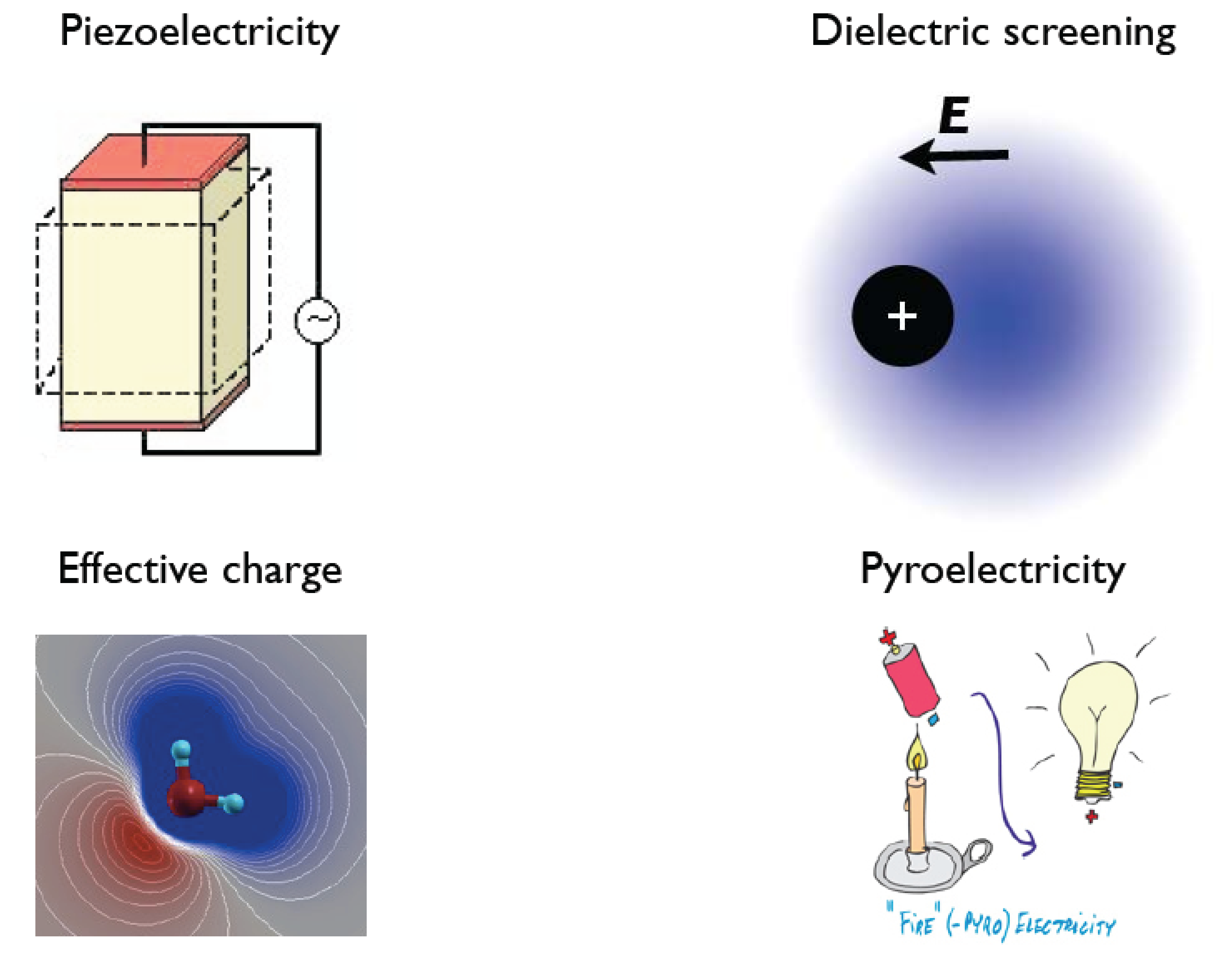
King-Smith & Vanderbilt, Phys. Rev. B **47**, 1651 (1993)

BerryPi

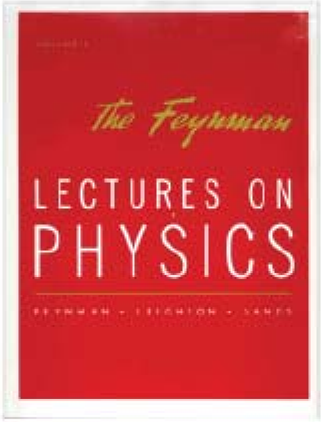
Macroscopic polarization

from O. Rubel (Lakehead University)
needs wien2wannier, Python 2.7.x, numpy
Comput.Phys.Commun. 184, 647 (2013)

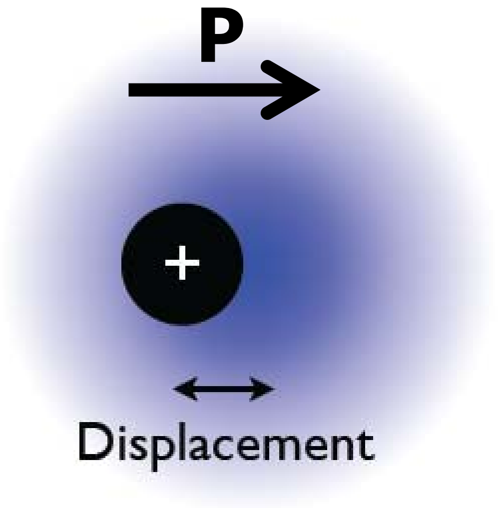
Materials properties related to Polarization

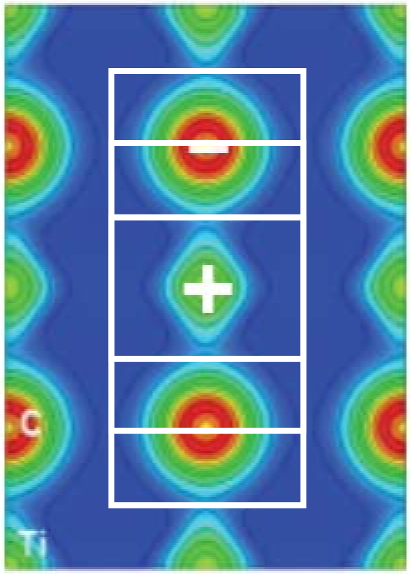


What is Polarization ?



We will now assume that in each atom there are charges q separated by a distance δ, so that $q\delta$ is the dipole moment per atom. (We use δ because we are already using d for the plate separation.) If there are N atoms per unit volume, there will be a *dipole moment per unit volume* equal to $Nq\delta$. This dipole moment per unit volume will be represented by a vector, $\boldsymbol{P}$. Needless to say, it is in the direction of the individual dipole moments, i.e., in the direction of the charge





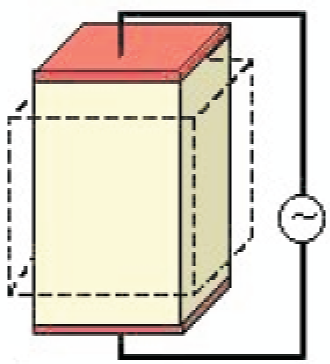
Polarization for periodic solids is undefined

Modern theory of polarization (Berry phase)

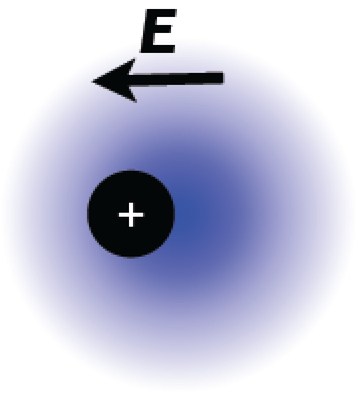
Pioneered by *King-Smith*, *David Vanderbilt* and *Raffaele Resta*

All measurable physical quantities are related to the **change** in polarization!

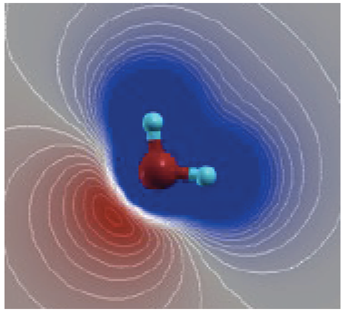
$$\Delta \mathrm{P} = \mathrm{P}^{(0)} - \mathrm{P}^{(1)}$$



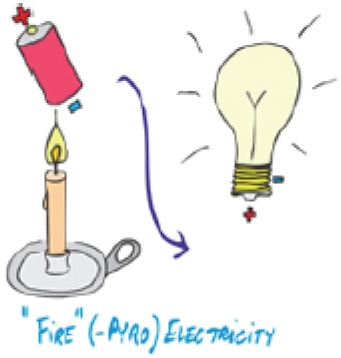
$$\frac{\Delta \mathrm{P}}{\Delta \mathrm{strain}}$$



$$\frac{\Delta \mathrm{P}}{\Delta \mathrm{E}}$$



$$\frac{\Delta \mathrm{P}}{\mathrm{displacement}}$$

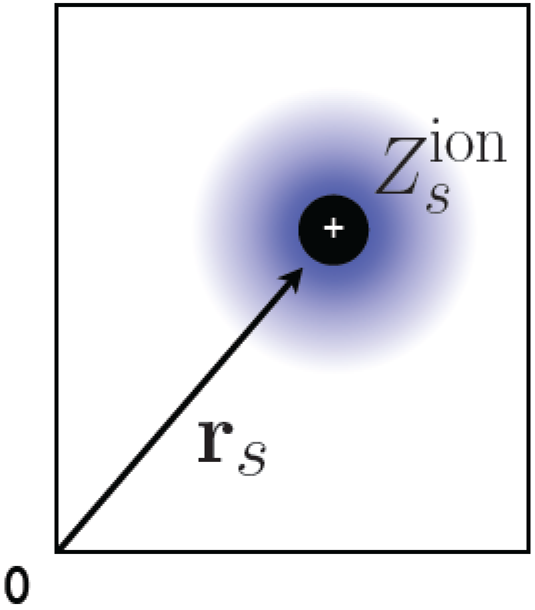


$$\frac{\Delta \mathrm{P}}{\Delta T}$$

Polarization as a transient current

$$\Delta\mathbf{P} = \mathbf{P}^{(1)} - \mathbf{P}^{(0)} = \Omega^{-1}\int dt \int_{\text{cell}} d\mathbf{r}\, \mathbf{j}(\mathbf{r},t)$$

transient current density



$$\mathbf{P} = \mathbf{P}_{\text{ion}} + \mathbf{P}_{\text{el}}$$

$$\mathbf{P}_{\text{ion}} = \frac{e}{\Omega}\sum_{s}^{\text{atoms}} Z_s^{\text{ion}}\mathbf{r}_s$$

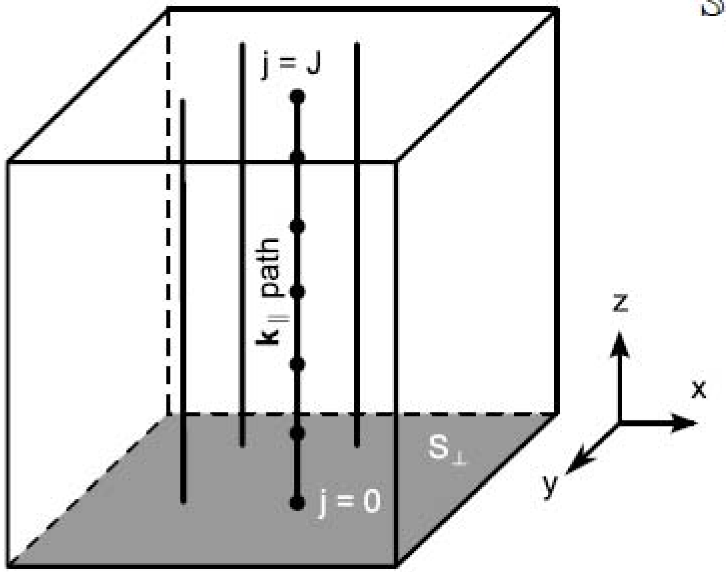
In Wien2k Z_s^{ion} is the core charge

$$-\mathbf{P}_{\text{el}} = \Omega^{-1}\int d\mathbf{r}\,\mathbf{r}\rho(\mathbf{r}) = \Omega^{-1}\sum_{n}^{\substack{\text{occ.}\\ \text{bands}}}\langle\psi_n|\mathbf{r}|\psi_n\rangle \equiv \frac{2ei}{(2\pi)^3}\sum_{n}^{\substack{\text{occ.}\\ \text{bands}}}\int_{\text{BZ}} d\mathbf{k}\,\langle u_{n\mathbf{k}}|\nabla_{\mathbf{k}}|u_{n\mathbf{k}}\rangle$$

King-Smith and David Vanderbilt, Phys. Rev. B **47**, 1651 (1993)

Berry phase

$$\mathrm{d}\varphi_n = -i\langle u_{n\mathbf{k}}|\nabla_{\mathbf{k}}|u_{n\mathbf{k}}\rangle \cdot \mathrm{d}\mathbf{k} = -i\ln\langle u_{n\mathbf{k}}|u_{n(\mathbf{k}+\mathrm{d}\mathbf{k})}\rangle$$



$$\mathbb{S}_{mn}(\mathbf{k}_j,\mathbf{k}_{j+1}) = \langle u_{m\mathbf{k}_j}|u_{n\mathbf{k}_{j+1}}\rangle$$

WIEN2WANNIER

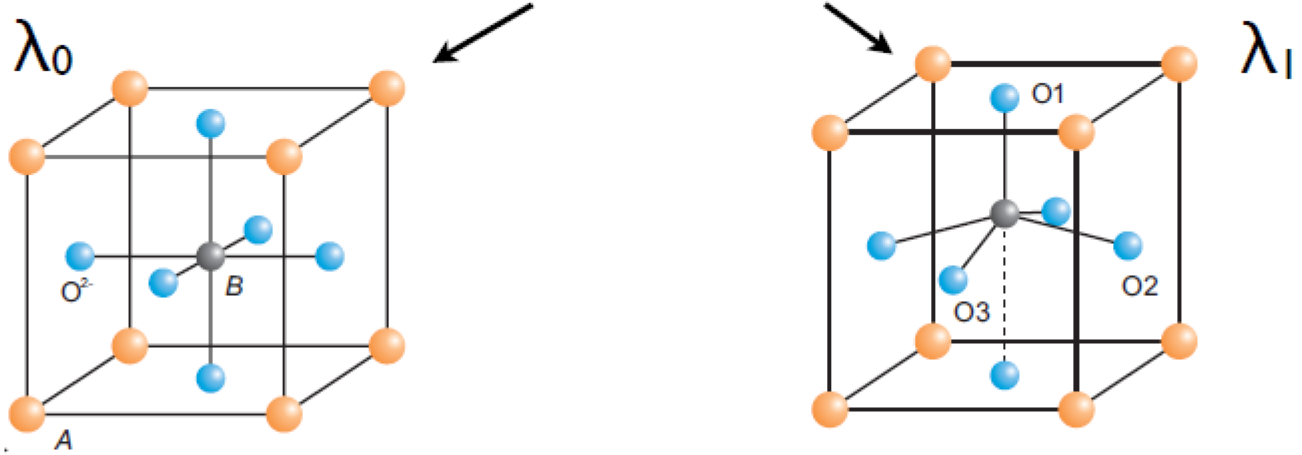
$$\varphi(\mathbf{k}_\parallel) = 2\,\mathrm{Im}\left[\ln\prod_{j=0}^{J-1}\det\mathbb{S}_{M\times M}(\mathbf{k}_j,\mathbf{k}_{j+1})\right]$$

$$\varphi_{\mathrm{el},\alpha} = S_\perp^{-1}\int_{S_\perp}\mathrm{d}S_\perp\,\varphi(\mathbf{k}_\parallel)$$

$$P_\alpha = \frac{e(\varphi_{\mathrm{el},\alpha}+\varphi_{\mathrm{ion},\alpha})}{2\pi\Omega}R_\alpha$$

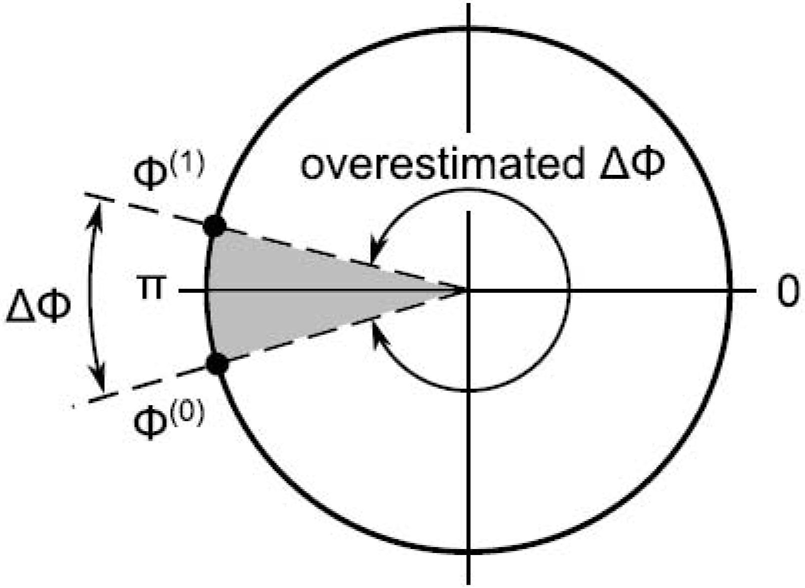
Polarization change for 2 cases:

$$\Delta \mathbf{P} = \mathbf{P}^{(0)} - \mathbf{P}^{(1)}$$



- structure file <u>must</u> preserve the symmetry
- begin with the lowest symmetry (λ_l) case
- copy case λ_l to case λ_0
- edit structure file for case λ_0
- do <u>not</u> initialize calculation (`init_lapw`)
- update density (`x dstart`)
- run SCF cycle (`run_lapw`)
- run `BerryPI`

Uncertainties



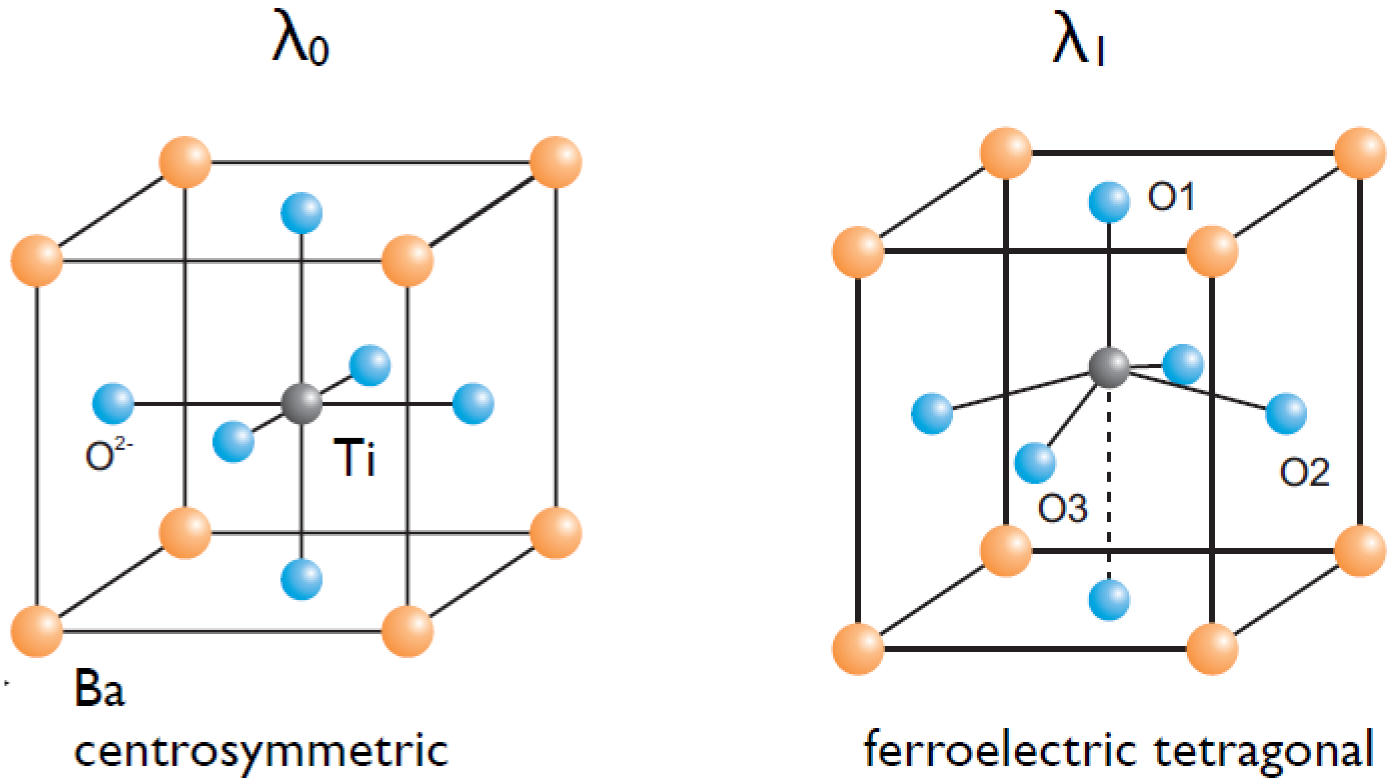
$$P_\alpha = \frac{e(\varphi_{\mathrm{el},\alpha} + \varphi_{\mathrm{ion},\alpha})}{2\pi\Omega} R_\alpha$$

$$\Delta\mathbf{P} = \mathbf{P}^{(0)} - \mathbf{P}^{(1)} \pm \frac{e}{\Omega}\mathbf{R}$$

- cannot determine large polarization difference ~1 C/m^2

Solution: λ_1 ⇨ $\lambda_{1/2}$ ⇨ λ_1

Spontaneous polarization in $BaTiO_3$



$$P_{\mathrm{s}} = P_{\mathrm{nc}} - P_{\mathrm{c}}$$

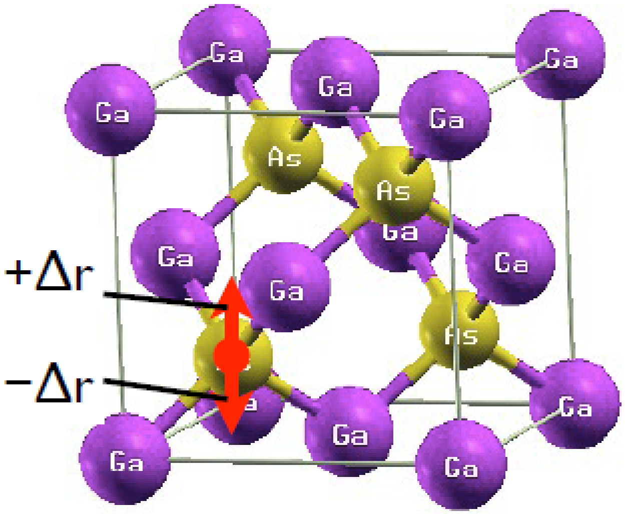
Born effective charges (LO-TO splitting)

Born effective charge is related to polarization

$$\boxed{Z^*_{s,\alpha\beta} = \frac{\Omega}{e}\frac{\partial P_\alpha}{\partial r_{s,\beta}}}$$

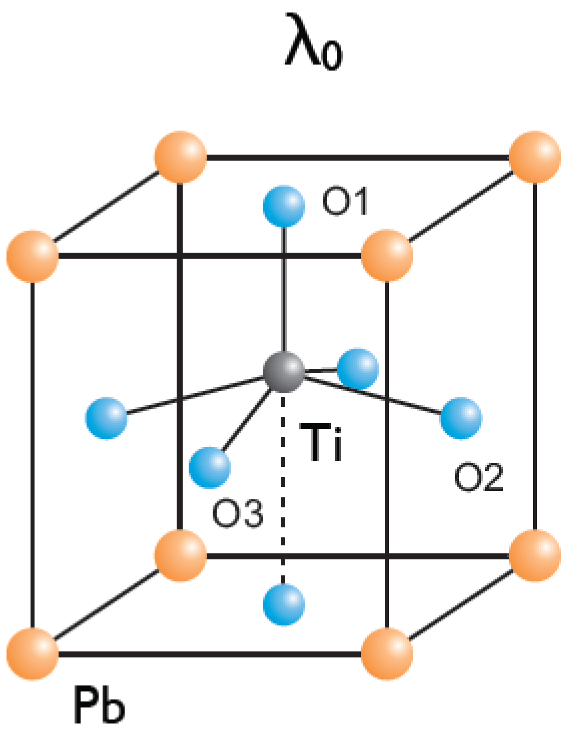
P - Polarization
r - atom position
Ω - unit cell volume
e - elementary charge

Introduce small displacements
±Δr ≪ a_0



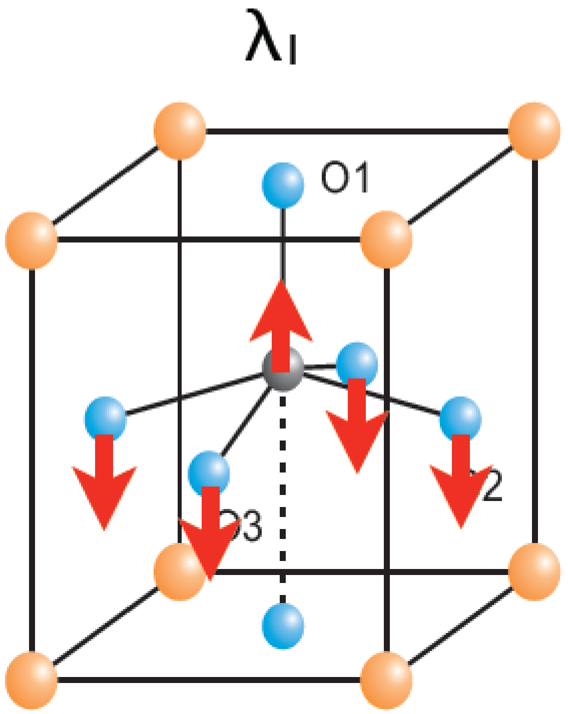
Need to compute the polarization difference between 2 structures:
dP = P(+Δr) − P(−Δr)

Piezoelectricity



tetragonal equilibrium

$a_0 = b_0 \neq c_0$



tetragonal strained

$a_0 = b_0 \neq c_0(1+\varepsilon_3)$