# 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

$$E^{cohes.}_{A_xB_y} = E^{crystal} - xE^{atom}_{A} - yE^{atom}_{B}$$

- $E^{crystal}$: 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 distorted FCC box (identical RMT, RKmax, 1 k-point, spinpolarized)

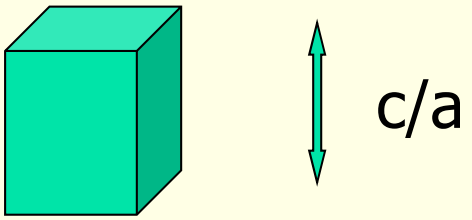
# Structural optimizations:

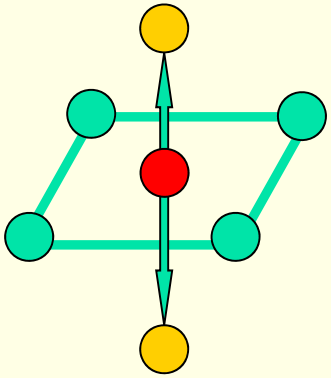
- Lattice parameters, volume, c/a ratio only via total energies:
  - *x optimize: creates a series of "struct" files + script "optimize.job"*
    - select volume or c/a, …
    - select number of cases and desired changes in volume (in % of $V_0$)
  - *edit optimize.job*
    - adapt to your need: change / uncomment various lines, eg.:
      - select different convergence parameters, parallelization, more iterations (-i 40)
      - modify "save_lapw" line (with more specific names)
      - replace "run_lapw" by "runsp_lapw" or add options (-min –fc 1 –orb)
  - *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.

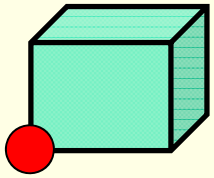
# Symmetry:

- 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

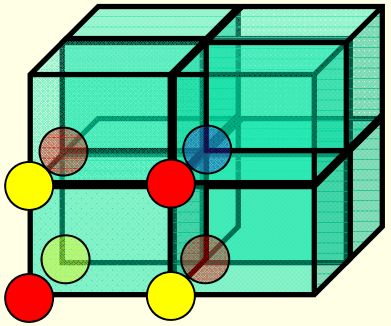


# Supercells (impurities, vacancies, alloys)



2x2x2 = 8 atoms

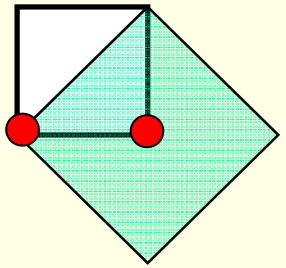


(0,0,0) P➔ 8 atoms (0,0,0) (.5,0,0) (.5,.5,0) (.5,.5,.5) (0,.5,0) (.5,0,.5) (0,0,.5) (0,.5,.5)

| | | | | |
|---|---|---|---|---|
| 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 !!! (otherwise sgroup will restore your original struct file)
  - *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!!!

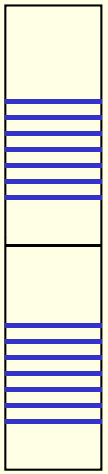
# Structeditor (by R.Laskowski)

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

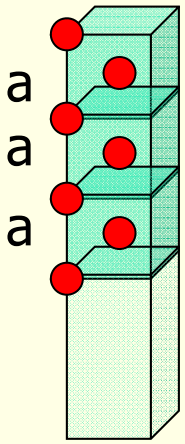
```
octave
s=loadstruct("GaN.struct")
# make an orthorhombic supercell and visualize it
a=[1 0 0; 1 1 0; 0 0 2]
sout=makesupercell (s,a);
showstruct(sout);
# save it as test.struct
savestruct (sout,"test.struct");
# get help on all commands
helpstruct
```

# 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)

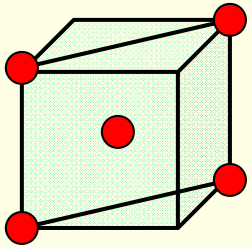
shift to ➔ inversion

(.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

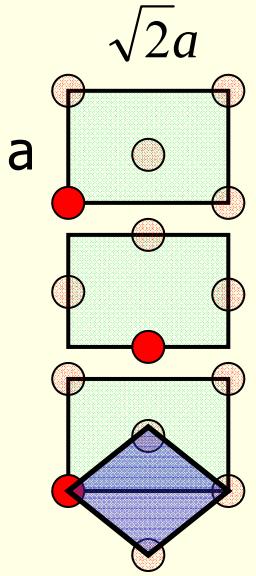
## bcc (110):



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

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

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

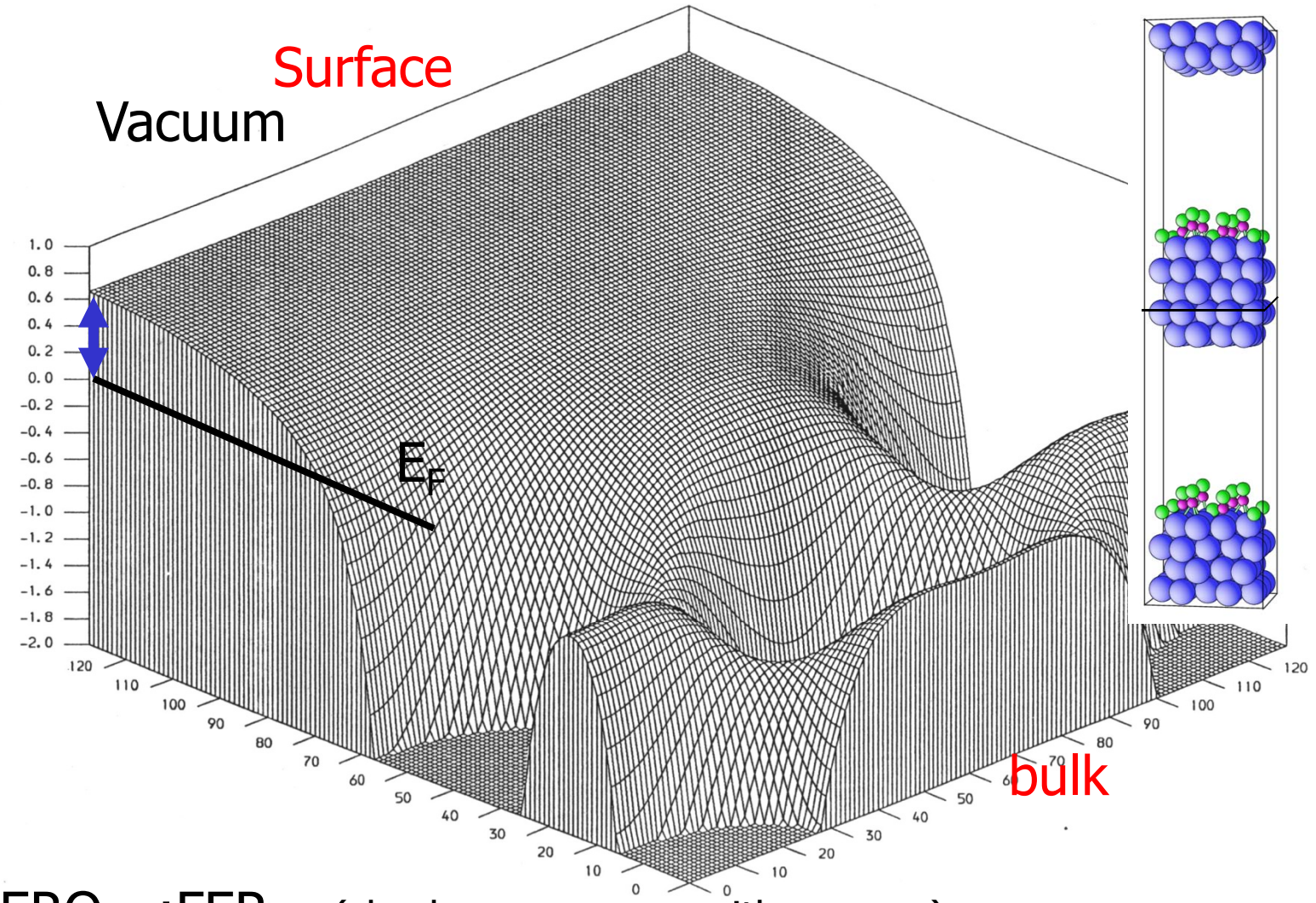


# Work function

**Work function**

potential

supercell




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 $\alpha$:

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

  - *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

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

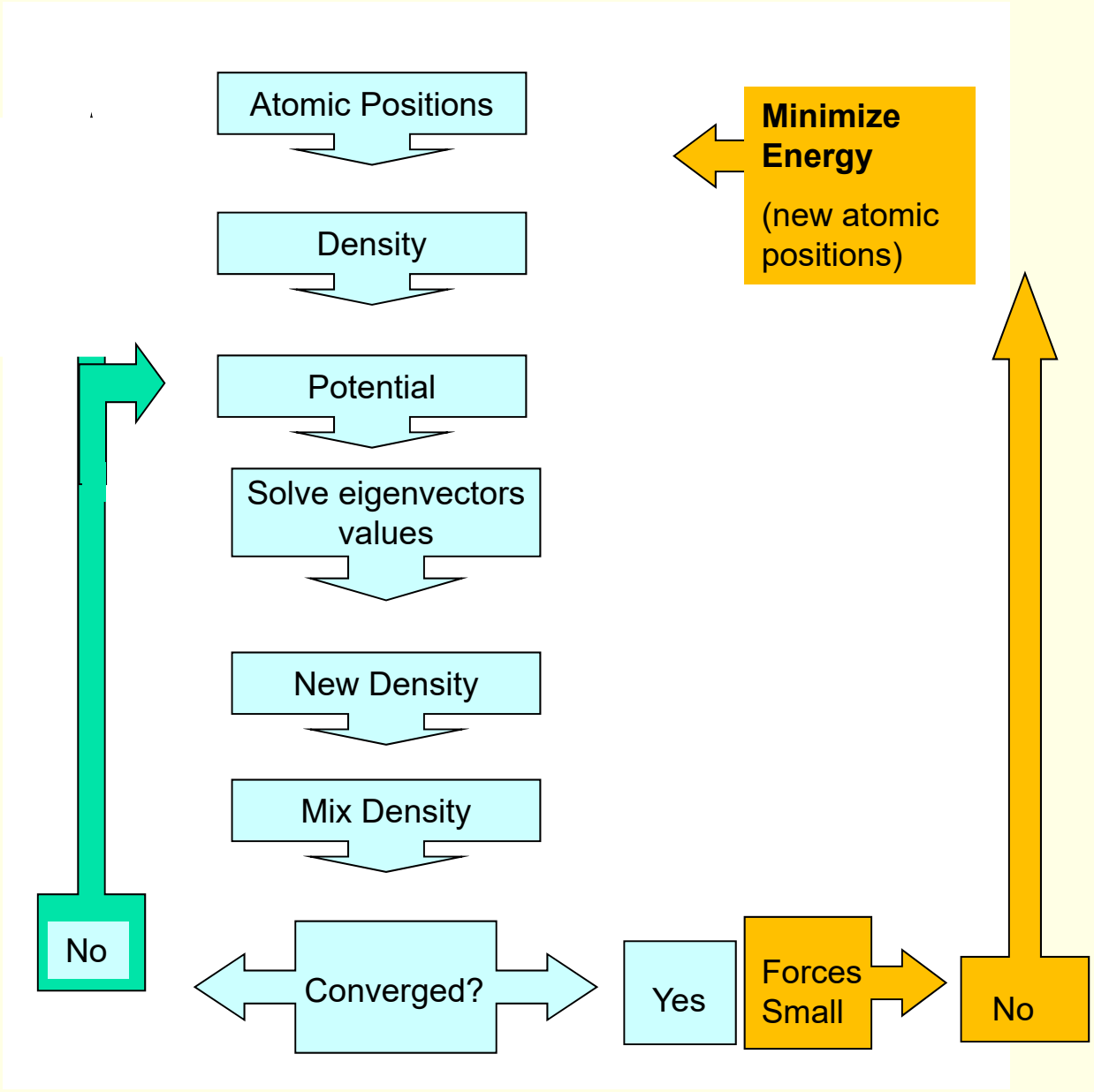
$$\left[(K^2-\varepsilon_i)\oint \phi_{K'}^*(r)\phi_K(r)\,dS_\alpha - i(K-K')\left\langle\phi_{K'}\middle|H-\varepsilon_i\middle|\phi_K\right\rangle_\alpha\right]$$

- 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 :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)



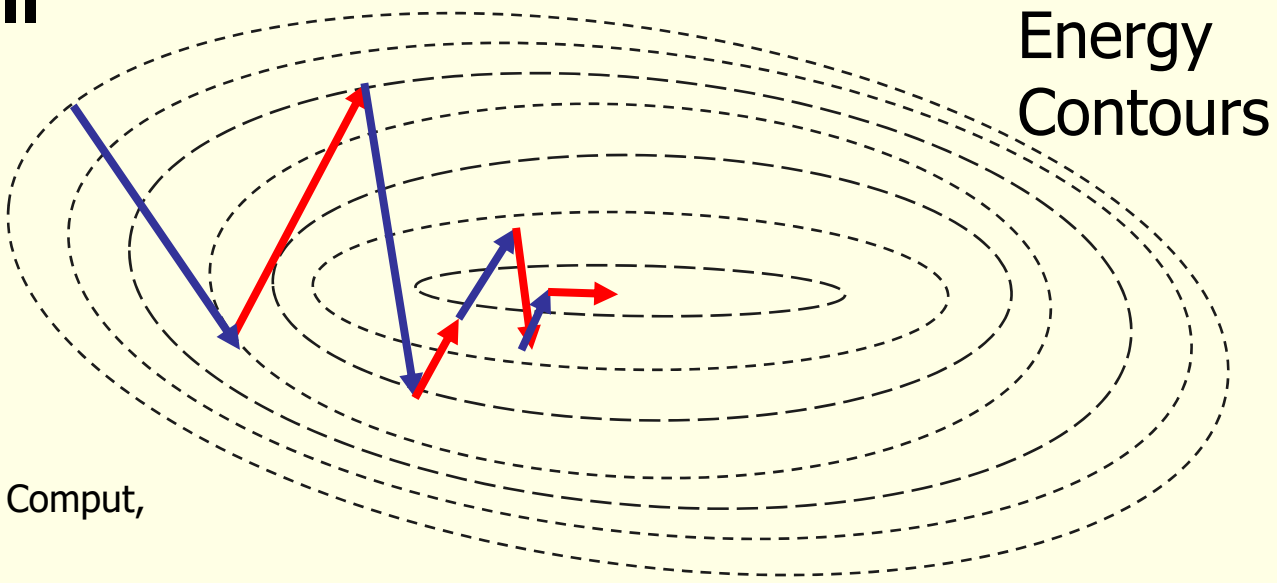

**Traditional way:**

- 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

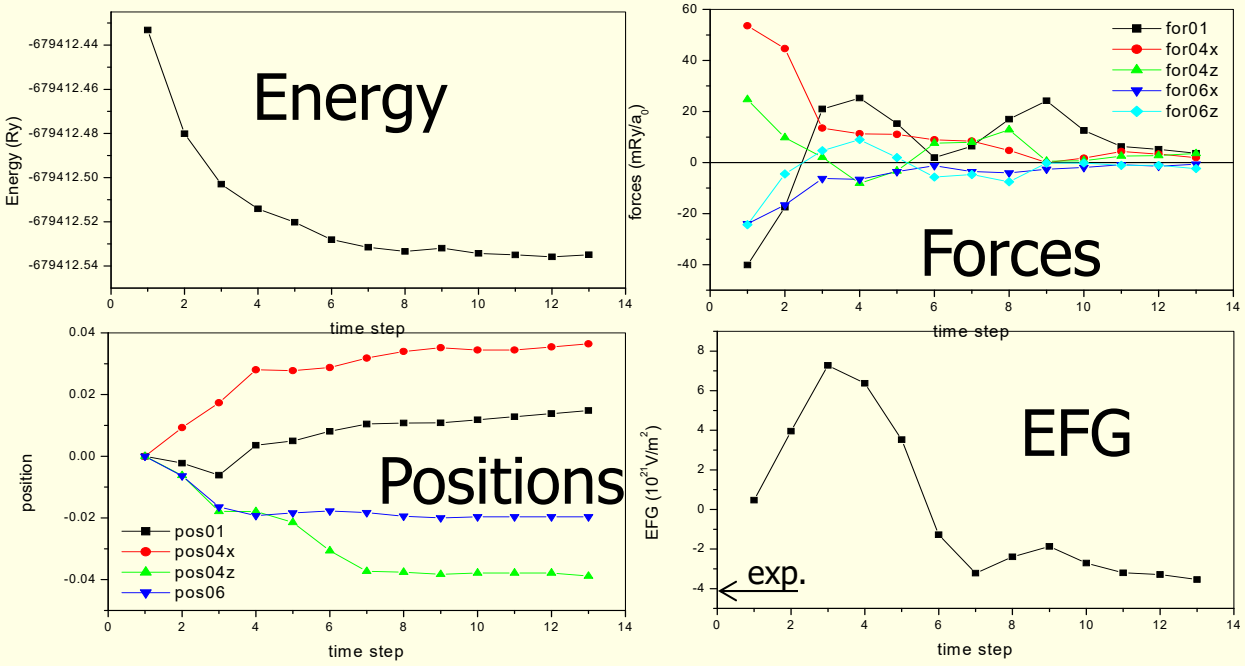
# Structural optimization of internal parameters using "PORT"

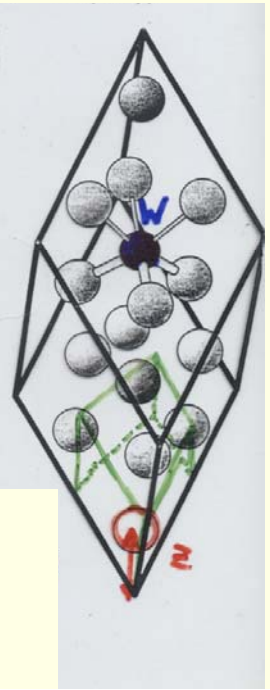
- /home/pblaha/tio2> min_lapw [-p -it -sp] [-j "run -fc 1 -p -it"] [-NI]
  - *performs scf-cycle for fixed positions*
  - *get forces and move atoms along forces (building an approximate Hessian) and writing a new case.struct file*
  - *extrapolate density (case.clmsum)*
  - *perform next scf cycle and 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

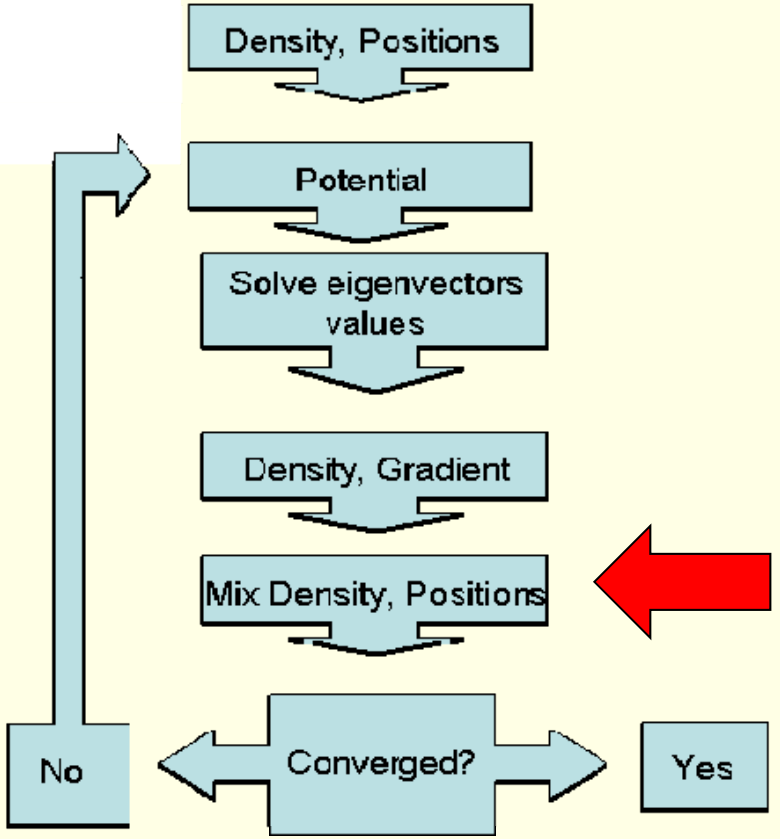
## W impurity in Bi (2x2x2 supercell: $Bi_{15}W$)



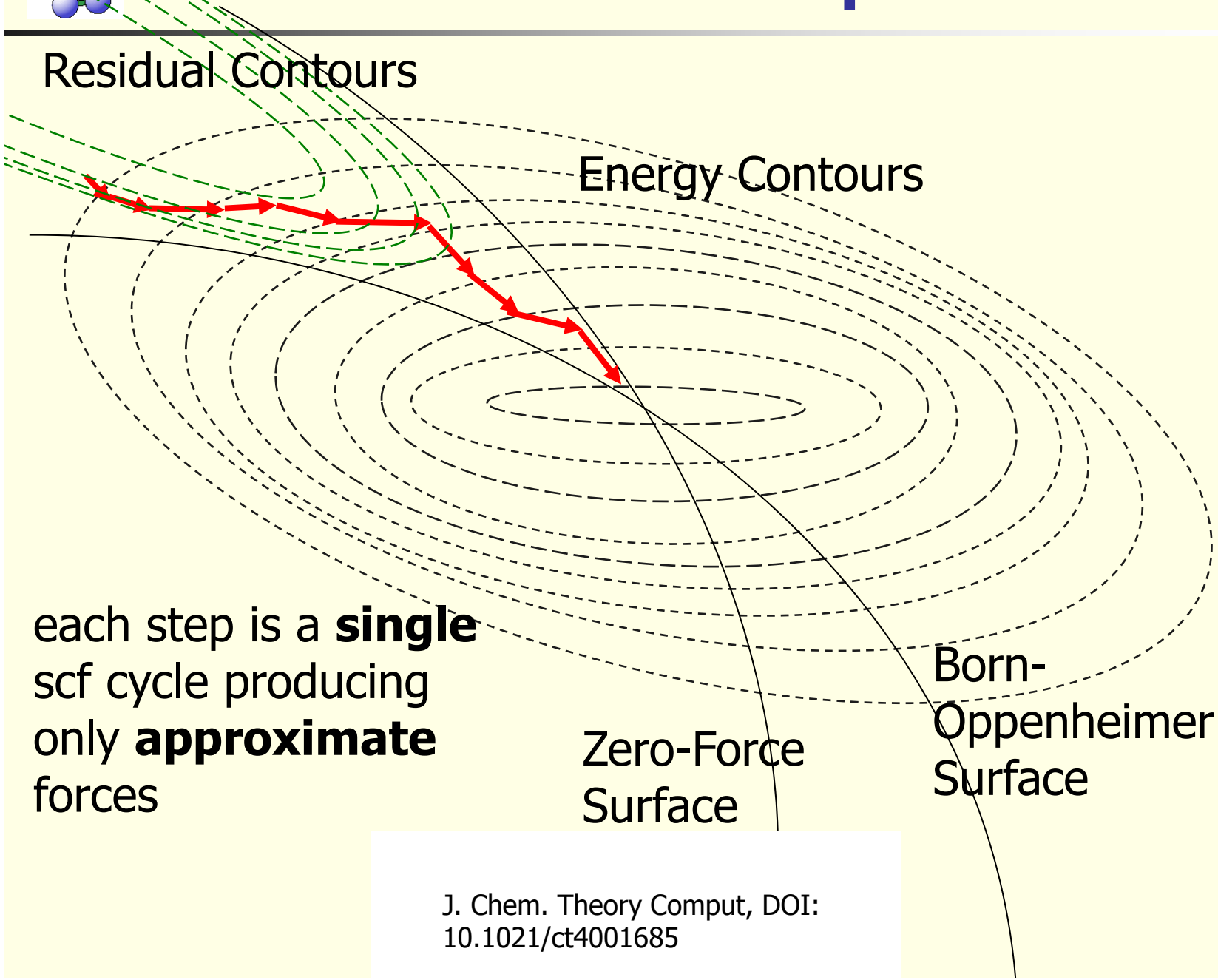
# Alternative method: **Fused Loop**

- 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




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.

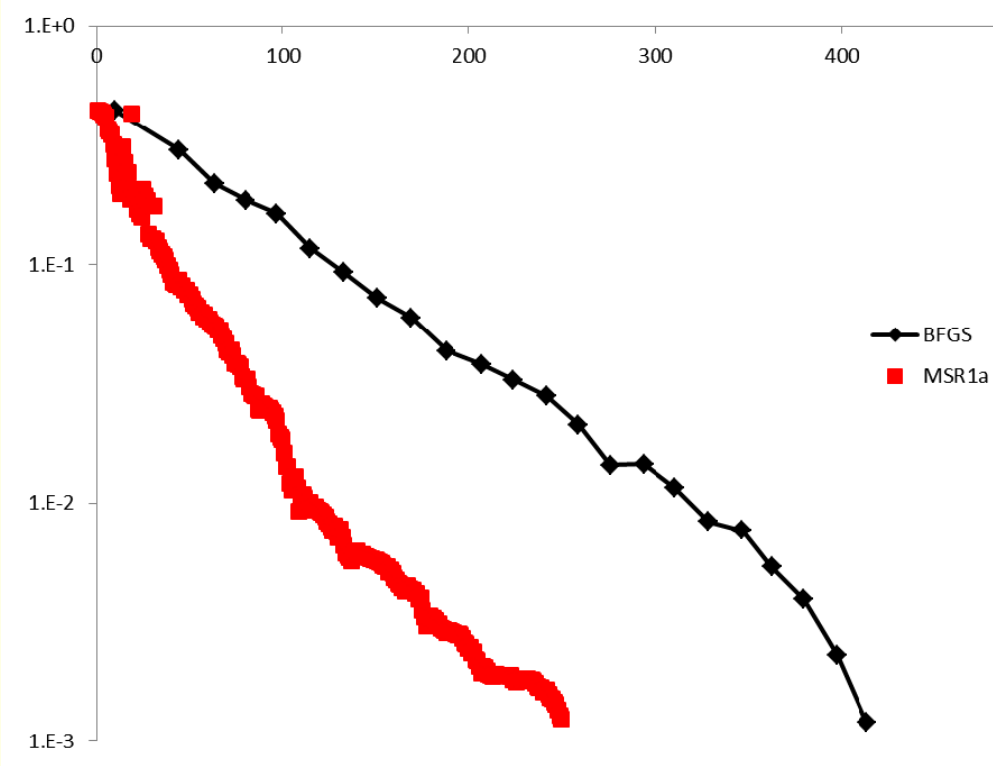
- Generalizable to multisecant method (better,

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

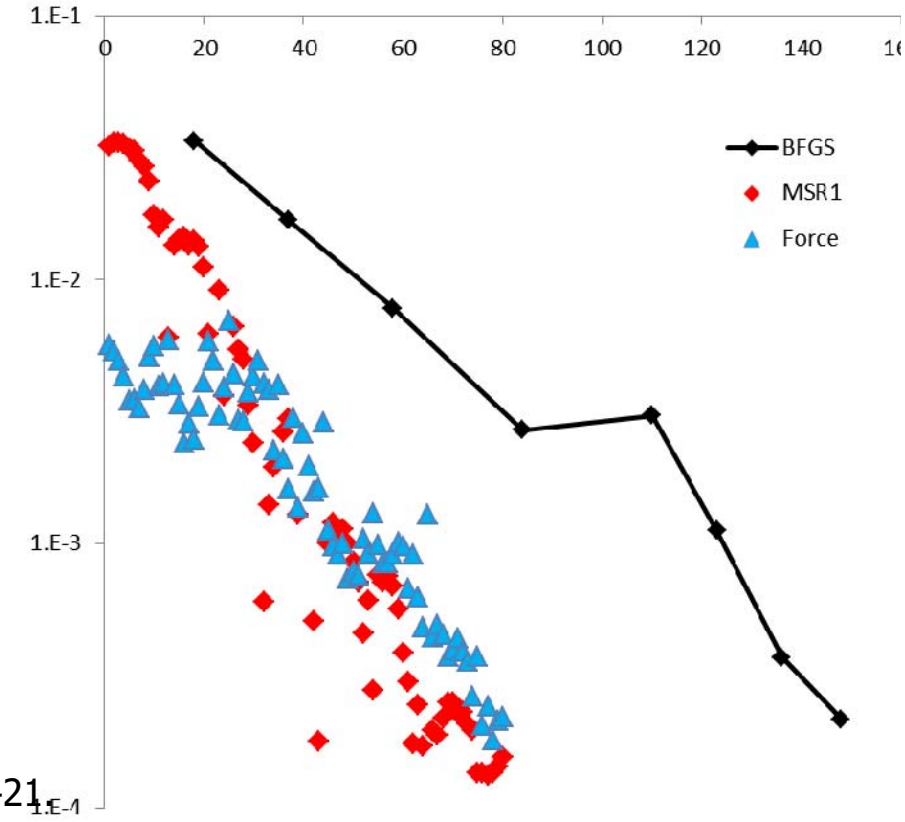
# Comparison of the 2 methods

## Larger Problems:

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




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




Lyudmila V. Dobysheva (2011)

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

# Structural optimization of internal parameters using "**MSR1a**"

- run_lapw **–min** -fc 1.0 -cc 0.001 -ec 0.0001 [-it -noHinv -p ]
- modifies case.inm and sets „**MSR1a**"

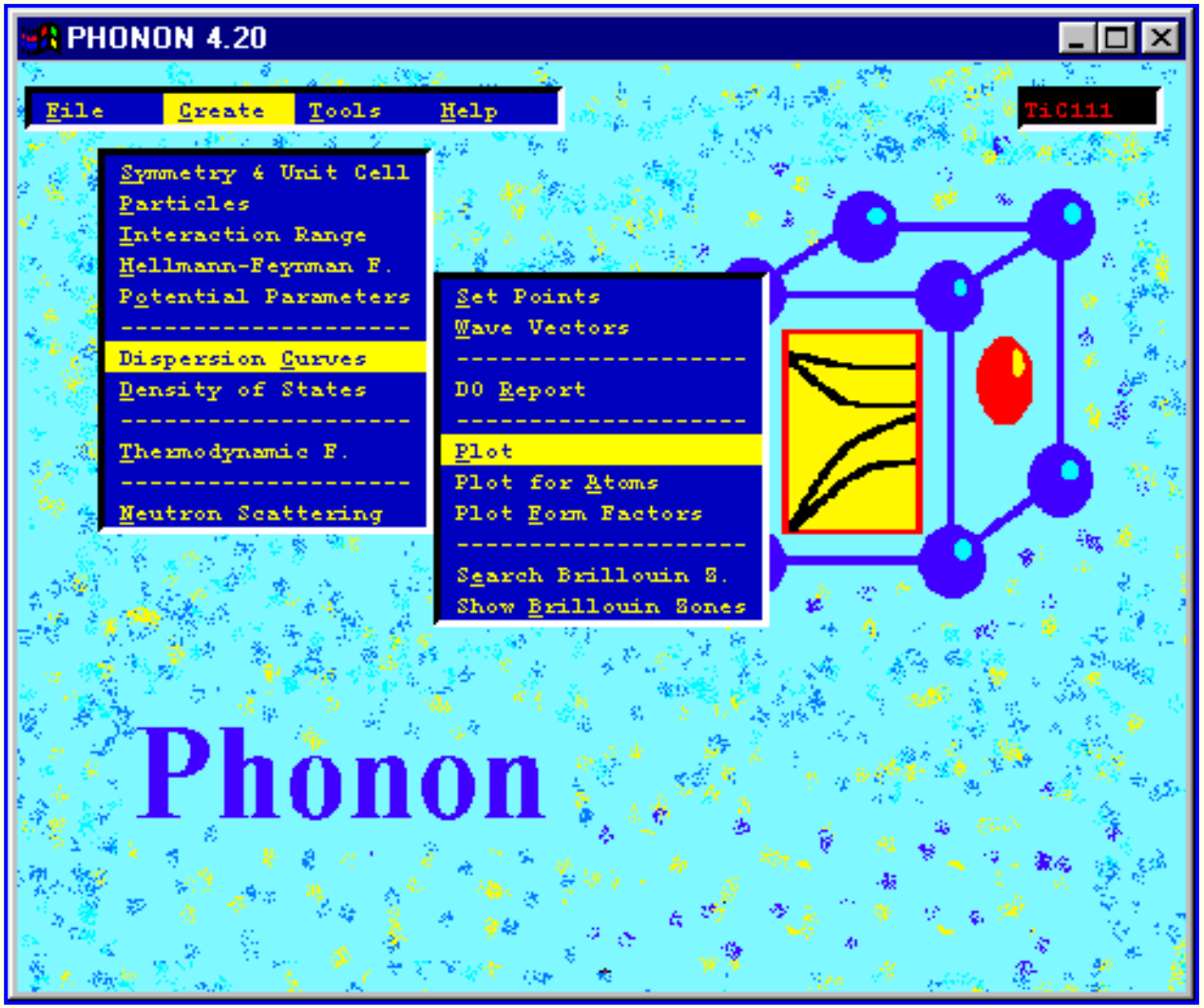
- This runs ONE big scf-calculations optimizing the density and the positions (forces towards zero) simultaneously (may need hundreds of iterations).

- 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).

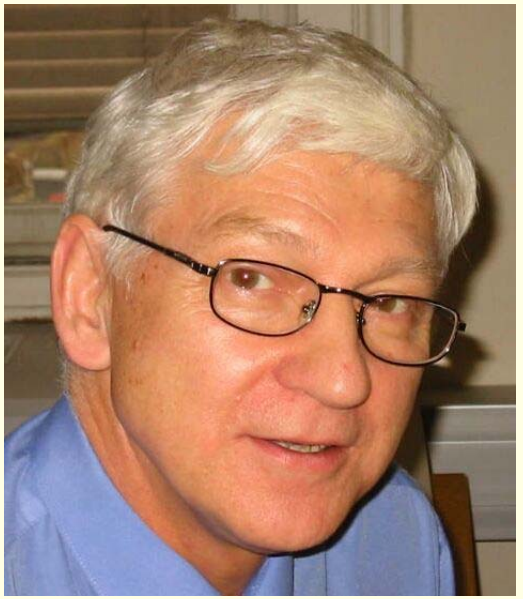
- quite efficient, recommended method, still under development by L.Marks (Northwestern Univ).

# 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
(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

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

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

The dynamical matrix is defined as

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

**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

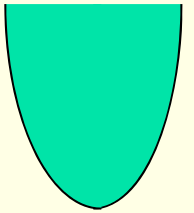
$$\mathbf{F}(\mathbf{n},\mu) = -\partial E/\partial \mathbf{R}(\mathbf{n},\mu)$$

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.

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



n,m: cells
μ,ν: atoms

## CUMMULANT FORCE CONSTANTS

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

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

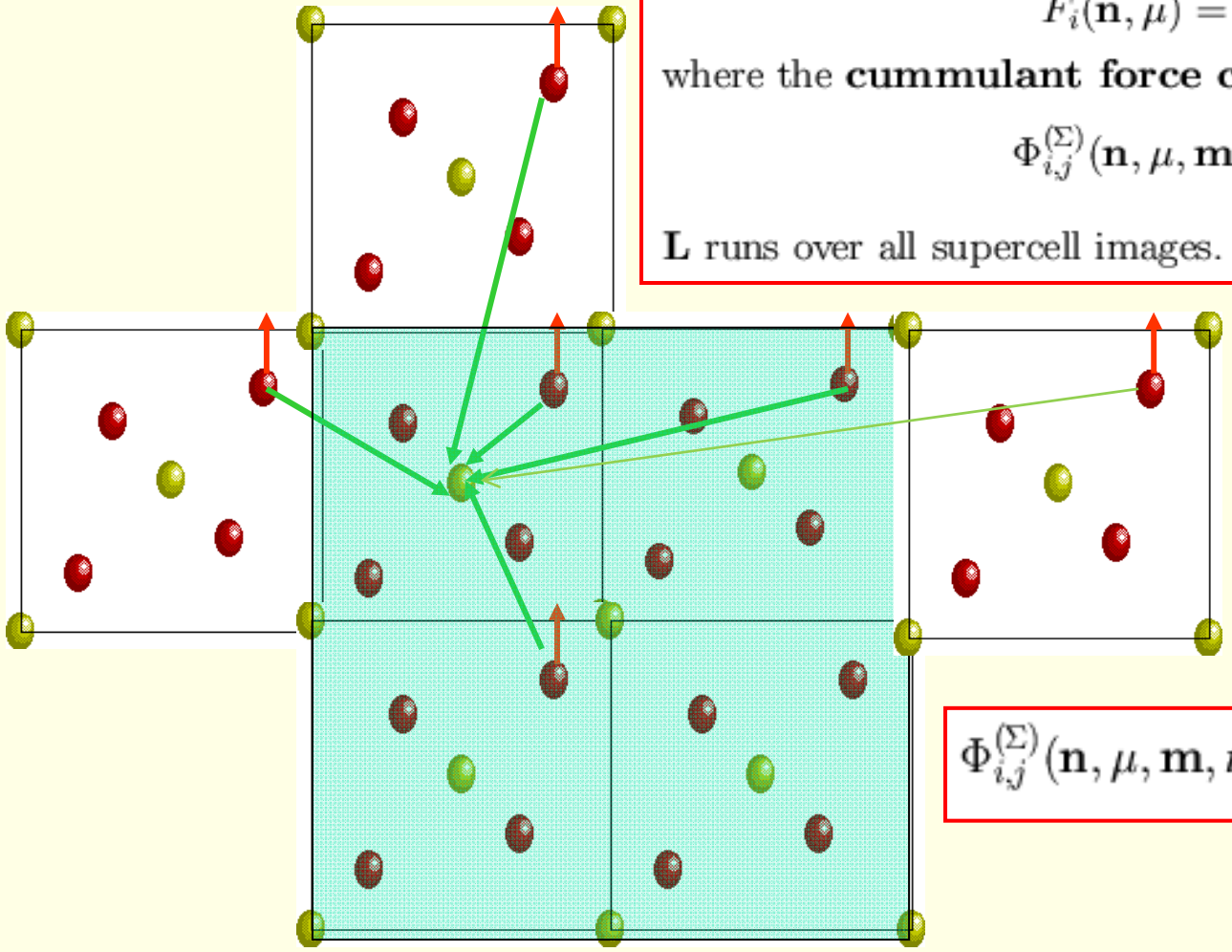
$\mathbf{L} = (L_a, L_b, L_c)$ are the indices of supercell lattice constants.
or

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

where the **cummulant force constant** is

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

$\mathbf{L}$ 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):

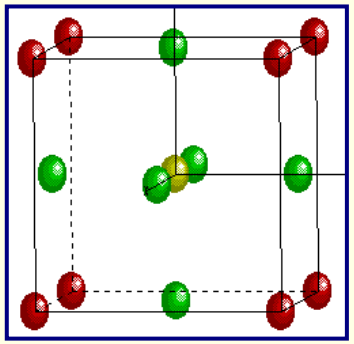
$$exp\{-2\pi i \mathbf{k}_s \cdot \mathbf{L}\} = 1$$

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.

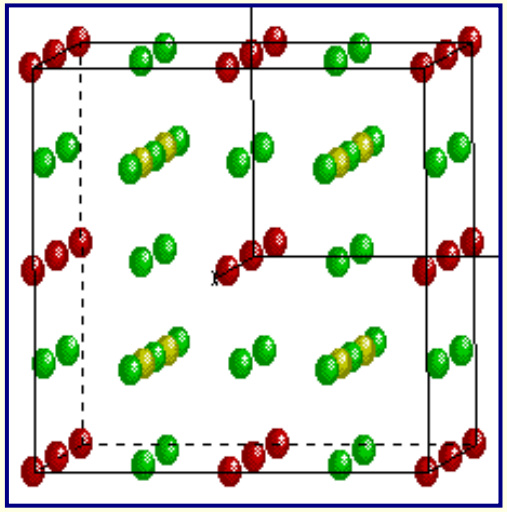
# Exact wave vectors

1x1x1



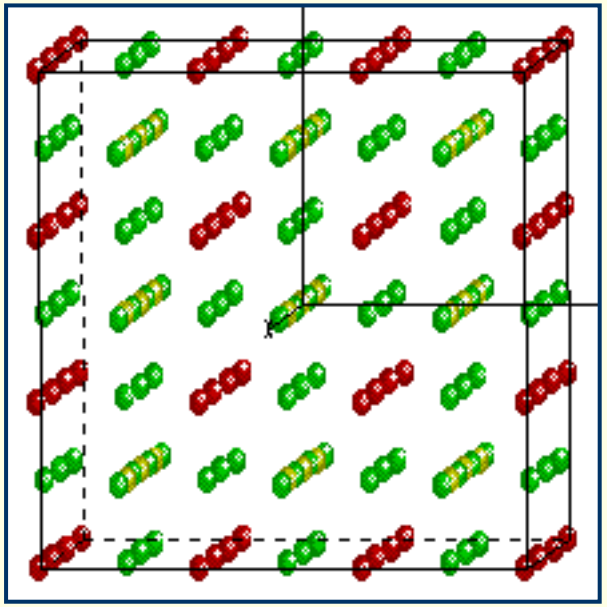
Exact: Γ

2x2x2

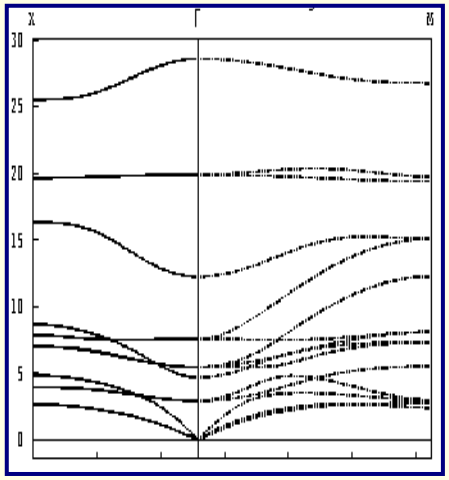


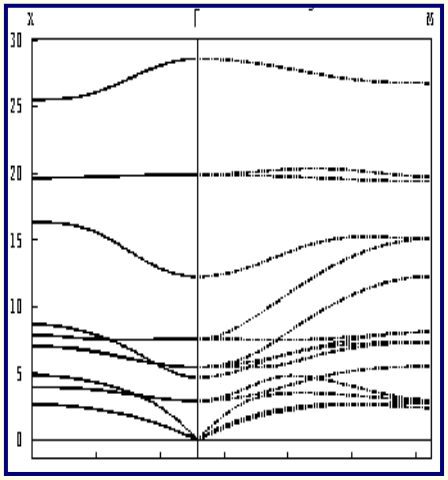
Exact: Γ, X, M, R

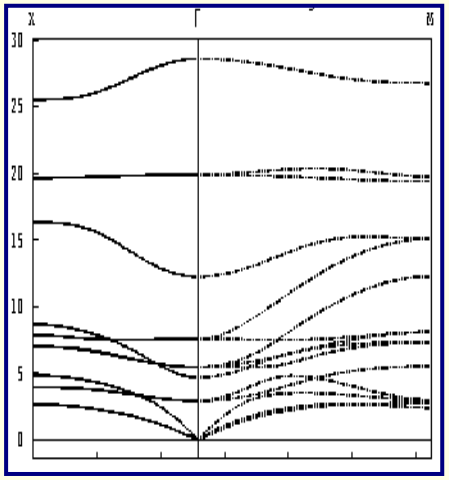
3x3x3



Exact: Γ X Γ M ⇩ Γ ⇩



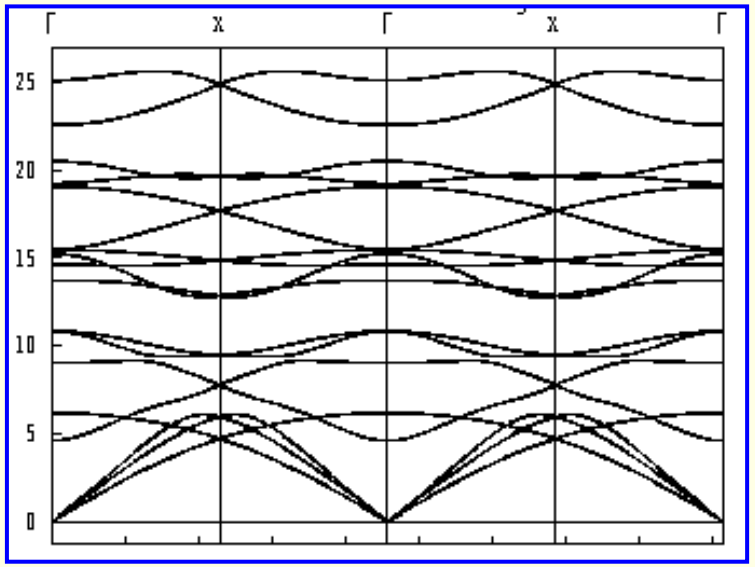



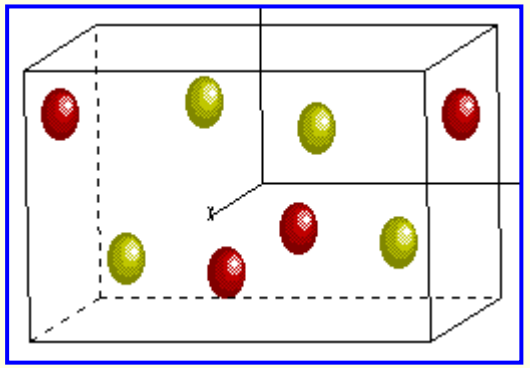
# Phonon dispersions + density of states

Frequency

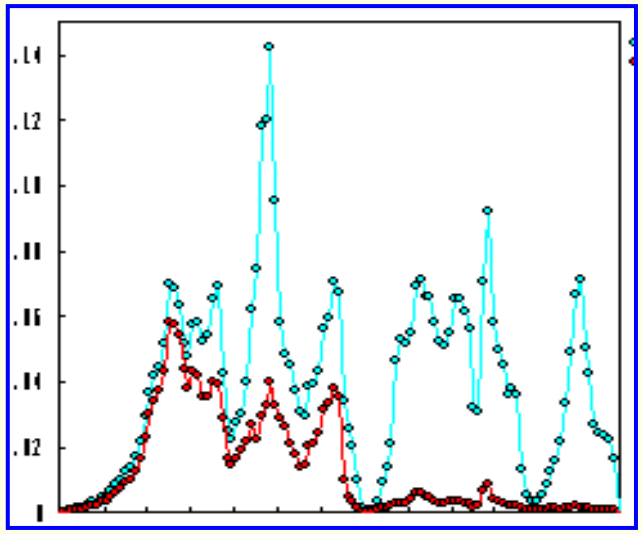
ω




$GeO_2$ P4_2/mnm

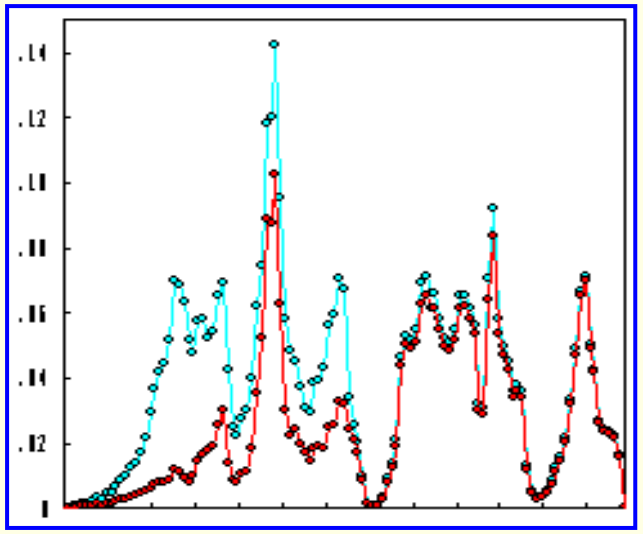


Total + Germanium



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:

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

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 $C_v$:

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

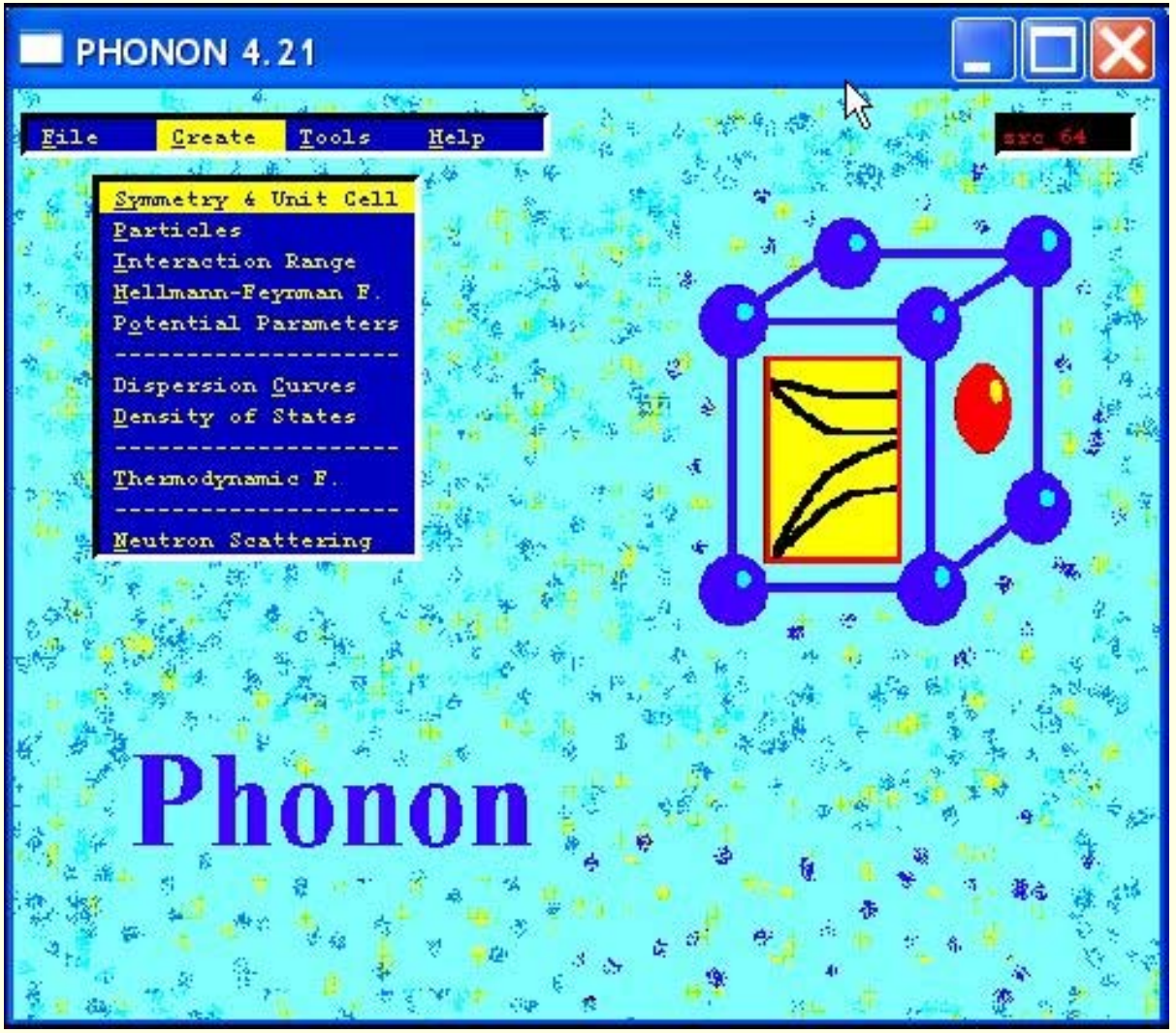
Thermal displacements:

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

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

# PHONON-I

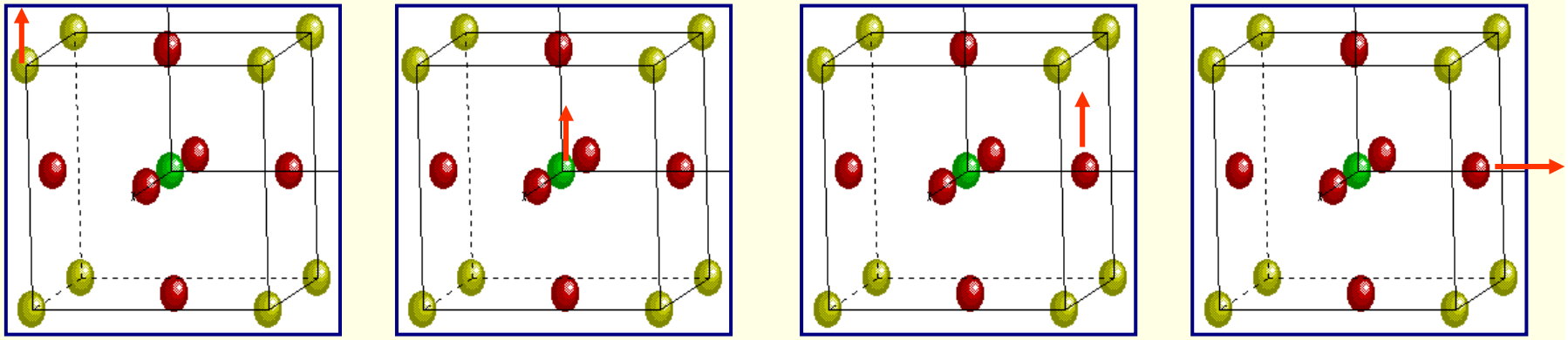
- PHONON
  - *by K.Parlinski (Crakow)*
  - *Linux or MS-windows*
  - *uses a „direct" method to calculate Force-constants with the help of an ab initio program*
  - *with these Force-constants phonons at arbitrary k-points can be obtained*
- Define your spacegroup
- Define all atoms




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

# Phonons:

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



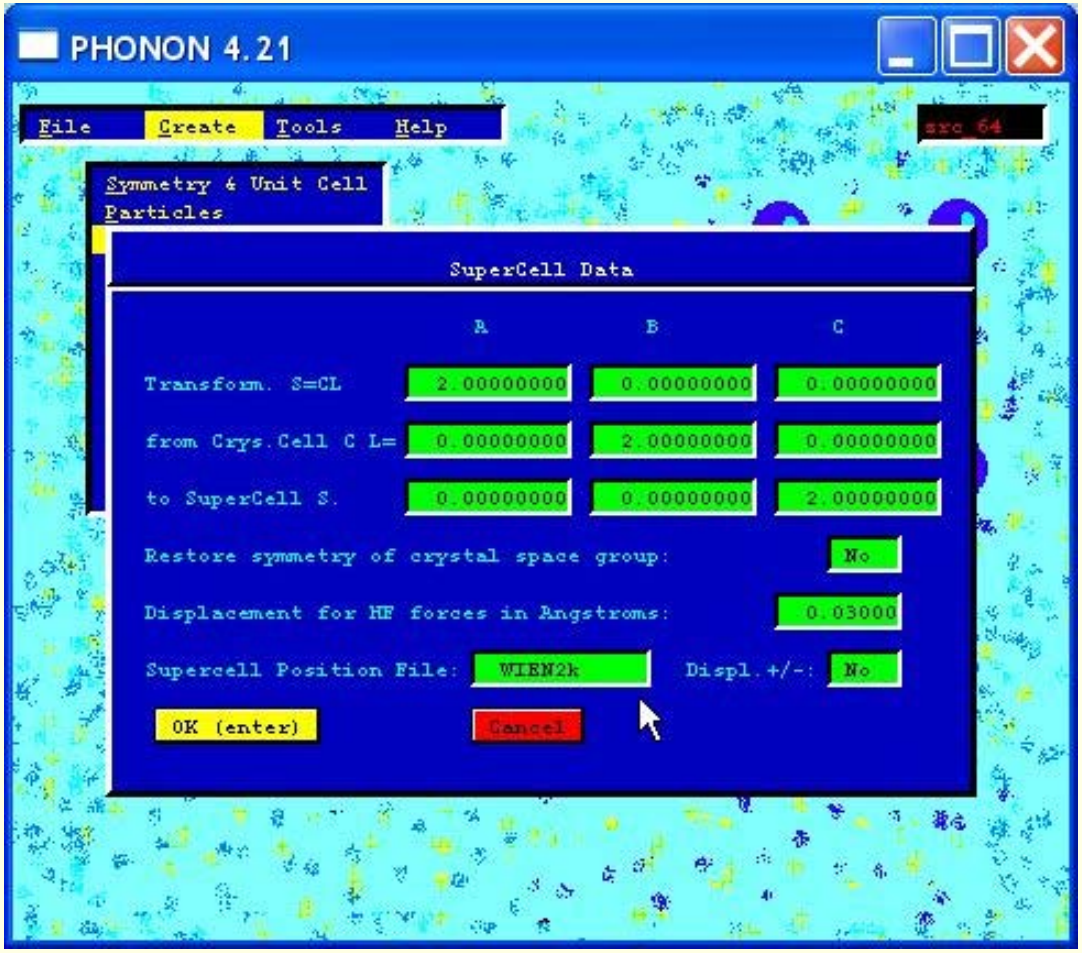
(Displacement pattern for cubic perovskite)

- *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

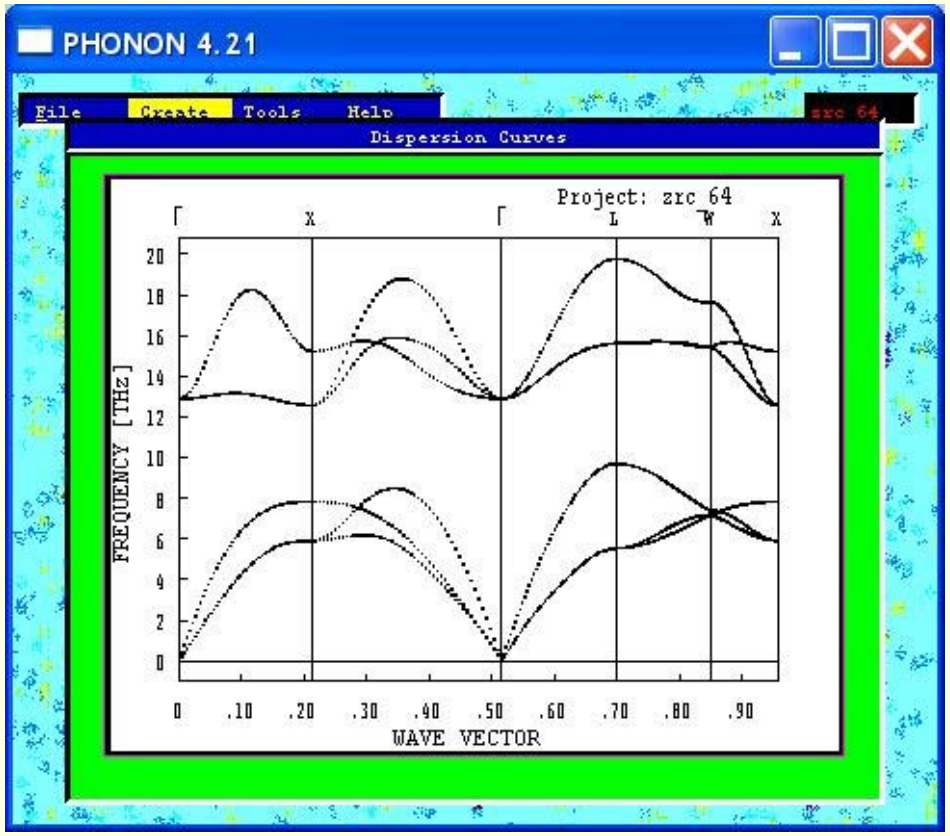
- Define an interaction range (supercell)
  - *create displacement file*
  - *transfer case.d45 to Unix*
- Calculate forces for all required displacements
  - *init_phonon_lapw*
    - for each displacement a case_XX.struct file is generated in an extra directory
    - runs nn and lets you define RMT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




- 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.
- free energies at T>0; quasiharmonic approximation

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