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
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 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*
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)
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
The convergence criterion in APW is the product of $R_{MT} \cdot K_{max}$

$$\Psi = \sum_{K_n}^{K_{MAX}} C_{K_n} e^{iK_n r}$$

http://www.wien2k.at/reg_user/faq/rkmax.html

medium quality convergence for smallest atom:

- basis set scales with $R K_{max}^3$
- cputime scales with $N_{PW}^3$
- increasing $R k_{max}$ by 10 % doubles cputime

START with SMALL $R k_{max}$ (relaxation), increase/test later
BZ integration, “FERMI’’-methods

- Replace the “integral” of the BZ by a finite summation on a mesh of “k-points”
  \[ \rho(r) = \sum_n \int_{E_n < E_F} \psi_{k,n}^* \psi_{k,n} \, d^3k = \sum_{k,n} w_{k,n} \psi_{k,n}^* \psi_{k,n} \]

- weights \( w_{k,n} \) depend on \( k \) and bandindex \( n \) (occupation)
  - for full “bands” the weight is given by “symmetry”
    - \( w(\Gamma) = 1, w(x) = 2, w(\Delta) = 4, w(k) = 8 \)
  - 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 oscillations, 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 $\Gamma$ ! (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_lapw x_lapw2 -up -orb`

- 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 laspw1 -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)

<table>
<thead>
<tr>
<th>:ENE</th>
<th>:DIS</th>
<th>:FER</th>
<th>:GAP</th>
<th>:CTO001</th>
<th>:NTO001</th>
<th>:QTL001</th>
</tr>
</thead>
<tbody>
<tr>
<td>:FOR002:</td>
<td>2.ATOM</td>
<td>19.470</td>
<td>0.000</td>
<td>0.000</td>
<td>19.470</td>
<td></td>
</tr>
<tr>
<td>:FGL002:</td>
<td>2.ATOM</td>
<td>13.767</td>
<td>13.767</td>
<td>0.000</td>
<td>total forces</td>
<td></td>
</tr>
<tr>
<td>:LAT</td>
<td>:VOL</td>
<td>:POSxxx</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 (correct number of ε)
case.in1

<table>
<thead>
<tr>
<th>WFFIL</th>
<th>EF=0.634</th>
<th>(WFPRI, SUPWF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.00</td>
<td>10 4</td>
<td>(R-MT*K-MAX; MAX L IN WF, V-NMT)</td>
</tr>
<tr>
<td>0.30</td>
<td>5 0</td>
<td>global E-param with N other, napw</td>
</tr>
<tr>
<td>0 0.30</td>
<td>0.000</td>
<td>CONT 1 Es</td>
</tr>
<tr>
<td>0 -3.72</td>
<td>0.005</td>
<td>STOP 1 Es-LO with search</td>
</tr>
<tr>
<td>1 -2.07</td>
<td>0.010</td>
<td>CONT 1 Ep with search</td>
</tr>
<tr>
<td>1 0.30</td>
<td>0.000</td>
<td>CONT 1 Ep-LO</td>
</tr>
<tr>
<td>2 0.30</td>
<td>0.010</td>
<td>CONT 1 0/1..LAPW/APW+lo</td>
</tr>
</tbody>
</table>

K-VECTORS FROM UNIT: 4

\[
\Psi = \sum_{K_n}^{K_{MAX}} c_{K_n} e^{iK_n r}
\]

\[
\Phi_{K_n} = \sum_{l}^{l_{max}} A_{lm} u_{l}(E, r) Y_{lm}
\]

\[
H_{n,m}^{NS} = \langle \Phi_{l}^{LM} | V_{LM}^{NS} | \Phi_{l}^{l} \rangle
\]

set \( E/ \) to \( E_F-0.2 \) Ry
HDLOs: case.in1

- f (d) wavefunctions have a large E-dependency in cases with large RMT
- For high precision calculations extend the basis set with a HDLO (high derivative LO):

\[
\Phi_{K_n} = \sum_l A_{lm}(K_n)u_l(E_l, r)Y_{lm} \\
\phi_{l,\text{atom}} = (A_{lm}u_{lm} + B_{lm}\dot{u}_l)Y_{lm} \\
\phi_{l,\text{atom}} = (A_{lm}u_{lm} + C_{lm}\ddot{u}_l)Y_{lm}
\]

- 2 0.30 0.010 CONT 1 APW+lo
- 2 0.30 0.010 CONT 2 HDLO

- F.Karsai et al., CPC 220, 230 (2017)
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 40 44 46 0 64
  0 0 40 44 46 0 64
  14. GMAX (for small H set it to 20-24)
- FILE FILE/NOFILE write recprlist

\[ \rho(r) = \sum_{LM} \rho_{LM}(r) Y_{LM}(\hat{r}) \]

\[ \rho(r) = \sum_{G} \rho_{G} e^{iGr} \]
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, $\rho$ 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
    ($\nabla \rho \cdot \vec{n} = 0$)
  - spin+orbital magnetic moments (spin-orbit / LDA+U)
- Hyperfine parameters
  - hyperfine fields (contact + dipolar + orbital contribution)
  - Isomer shift
  - Electric field gradients
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
  ▫ $x$ lapw2 –qtl
    $\Psi_n^*\Psi_n=1=q_{out} + \sum_{t}^{\alpha} \sum_{l} q_{t,l}$
  ▫ case.outputt
    ▪ integrated DOS
  ▫ case.dos1ev (3ev)
    ▪ text-file for plotting
    ▪ $E$-zero at $E_F$
partial charges:

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

- see *case.struct* and *case.outputs*

- **x qtl** (instead of x lapw2 -qt)
  - *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 dependent 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, \( \rho \) 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 (\( \nabla \rho \cdot \mathbf{n} = 0 \))*
    - 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 C₂H₄
Bonds are characterized by „critical points“, where $\nabla \rho = 0$

- density maximum: (3,-3); 3 negative curvatures $\lambda$, (at nucleus or non-NM)
- bond CP: (3,-1): 2 negative, 1 positive $\lambda$ (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 $C_2H_4$
"Atoms" are regions within a zero-flux surface $\nabla \rho \cdot \hat{n} = 0$

$\rho$ of $C_2H_4$ with zero-flux lines defining atomic basins
- example of BN/Ni with “difference” to free atoms,
- workfunction shift
- Bader analysis of some inorganic compounds:

<table>
<thead>
<tr>
<th></th>
<th>$\rho$(e/A$^3$)</th>
<th>$\Delta\rho$(e/A$^5$)</th>
<th>Q (e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cl$_2$</td>
<td>1.12</td>
<td>-6.1</td>
<td>-</td>
</tr>
<tr>
<td>I$_2$</td>
<td>0.48</td>
<td>-0.9</td>
<td>-</td>
</tr>
<tr>
<td>TiC</td>
<td>0.51</td>
<td>1.8</td>
<td>1.7</td>
</tr>
<tr>
<td>TiN</td>
<td>0.47</td>
<td>3.9</td>
<td>1.7</td>
</tr>
<tr>
<td>TiO</td>
<td>0.43</td>
<td>5.8</td>
<td>1.5</td>
</tr>
<tr>
<td>KCl</td>
<td>0.08</td>
<td>1.2</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Cl$_2$ more covalent than I$_2$

Cl$_2$ more covalent then I$_2$

more ionic, but less charge?

less ionic then TiC?
You must have a “good” scf-density (case.clmsum)

- no core leakage, LMs up to L=8-10 in case.in2

SURF
1
20 0.0 1.570796327
20 0.0 0.785398163
0.07 1.0 4
1.65 0.1
3 3 3
IRHO
WEIT
30
END
---------------------

CRIT
1
ALL
3 3 3
END

extractaim_lapw: ➔ critical_points_ang (converted units)
:PC x, y, z, \lambda_1, \lambda_2, \lambda_3, ch, laplacian, rho
Properties with WIEN2k - II

- **Total energy and forces**
  - optimization of internal coordinates, (MD, BROYDEN)
  - cell parameter only via $E_{\text{tot}}$ (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)
  - 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](http://www.wien2k.at/reg_users/unsupported)): quantum oscillations
Properties with WIEN2k - II

- **Total energy and forces**
  - optimization of internal coordinates, (MD, BROYDEN)
  - cell parameter only via $E_{\text{tot}}$ (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](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

\[ E_{A_xB_y}^{\text{cohes.}} = E^{\text{crystal}} - x E^{\text{atom}}_A - y E^{\text{atom}}_B \]

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

- \( E^{\text{atom}} \): LSTART: fully-relativistic ➔ inconsistent description
  ➔ for heavier elements (2nd row):
    supercell with one atom in a \(~30\) bohr distorted FCC box
    (identical RMT, equivalent 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

$\sqrt{2} \times \sqrt{2}$ 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:**

\[
\begin{array}{ccc}
(0 & 0 & 6z) & (0.5 & 0.5 & +/3z) \\
(0.5 & 0.5 & 5z) & (0 & 0 & +/-2z) \\
(0 & 0 & 4z) & shift to & (0.5 & 0.5 & +/-z) \\
(0.5 & 0.5 & 3z) & inversion & (0 & 0 & 0) \\
(0 & 0 & 2z) & & z = a/2c \\
(0.5 & 0.5 & z) & & \\
(0 & 0 & 0) & & \\
\end{array}
\]

**bcc (110):**

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

\[
\begin{array}{ccc}
(0 & 0 & 0) & z = a/\sqrt{2}a \\
(0.5 & 0 & +/-z) & & \\
(0 & 0 & +/-2z) & & \\
\end{array}
\]
Work function

potential

supercell

Surface

Vacuum

Work function

\[ \text{WF} = \text{VZERO} - \text{FER} \quad \text{(check convergence with vacuum)} \]
Total energies and atomic forces

(Yu et al.; Kohler et al.)

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

- **Force on atom α:**

- **Hellmann-Feynman-force**
  - Pulay corrections
    - Core
    - Valence

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

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

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

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

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

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

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

\[
F_{val}^\alpha = \int \rho_{val}(r) \nabla_\alpha V_{eff}(r) \, d\vec{r} + \sum_i n_i \sum_{K,K'} c_i^*(K')c_i(K) \times
\]

\[
\left[ (K^2 - \varepsilon_i) \int \phi_{K'}^*(r)\phi_K(r) \, dS_\alpha - i(K-K') \langle \phi_K, [H - \varepsilon_i] \phi_K \rangle_\alpha \right]
\]
Optimization of internal parameters using “forces”

- Forces only for “free” structural parameters:
  - NaCl: (0,0,0), (0.5,0.5,0.5) : all positions fixed by symmetry
  - TiO$_2$: Ti (0,0,0), O (u,u,0): one free parameter (u,x,y,z)

- Forces are only calculated when using “-fc”:
  - run_lapw –fc 1.0 \( \text{(mRy/bohr)} \)
    - grep :fgl002 case.scf
      - 200. partial
      - -130. partial
      - 140. partial
      - 135 partial \( \text{only } F_{HF} + F_{core} \)
      - 120 partial
      - 122 partial \( \text{forces converging} \)
      - 121 partial \( \rightarrow \text{changes “TOT” to “FOR” in case.in2} \)
      - -12.3 total \( F_{HF} + F_{core} + F_{val} \), only this last number is correct

- Forces are useful for
  - structural optimization \( \text{(of internal parameters)} \)
  - phonons
Structure optimization (atomic positions)

Traditional way:

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

- Atomic Positions
- Density
- Potential
  - Solve eigenvectors values
  - New Density
  - Mix Density
- Minimize Energy (new atomic positions)
- Converged?
  - Yes: Forces Small
  - No: No
- Yes: Converged?
Traditional algorithm:

- 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 $N_{SCF} \times N_{BFGS} \times T_0$

Each step is a **full** scf calculation producing **accurate** forces.

Structural optimization of internal parameters using “PORT”

- `/home/pbla/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 positions (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)

Energy

Forces

Positions

EFG (10$^{-21}$ V/m$^2$)

exp.
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

---

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

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”**
  \[
  B_{k+1} = B_k + \frac{(y_k - B_k s_k) s_k^T}{s_k^T s_k} \quad H_{k+1} = H_k + \frac{(s_k - H_k y_k) s_k^T}{s_k^T y_k}
  \]
- **Broyden’s “Bad Method”**
  \[
  H_{k+1} = H_k + \frac{(s_k - H_k y_k) y_k^T}{y_k^T y_k}
  \]
- Generalizable to multisecant method (better,


Comparison of the 2 methods

Larger Problems:
52 atoms, MgO (111)+H₂O 108 atoms AlFe


Lyudmila V. Dobysheva (2011)
Structural optimization of internal parameters using “MSR1a”

- run_lapw \texttt{-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, \textbf{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 $R(n, \mu)$ is

$$ E(R(n, \mu), \ldots R(m, \nu), \ldots) = E_0 + \frac{1}{2} \sum_{n,\mu,m,\nu} \Phi(n, \mu, m, \nu) U(n, \mu) U(m, \nu) $$

where the force constant matrix are

$$ \Phi_{i,j}(n, \mu, m, \nu) = \left[ \frac{\partial^2 E}{\partial R_i(n, \mu) \partial R_j(m, \nu)} \right]_0 $$

is defined at $\frac{\partial E}{\partial R_i(n, \mu)} \big|_0 = 0$.

The dynamical matrix is defined as

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

$m$ runs over all atoms. Diagonalization of the dynamical matrix

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

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

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

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

on all other atoms. Hence

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

Master equation of direct method.
CUMMULANT FORCE CONSTANTS

Displace an atom by \( U(m, \nu) \)

\[
F_i(n, \mu) = -\sum_L \Phi_{ij}(n, \mu, m + L, \nu)U_j(m, \nu)
\]

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

or

\[
F_i(n, \mu) = -\Phi_{ij}^{(\Sigma)}(n, \mu, m, \nu)U_j(m, \nu)
\]

where the **cummulant force constant** is

\[
\Phi_{ij}^{(\Sigma)}(n, \mu, m, \nu) = \sum_L \Phi_{ij}(n, \mu, m + L, \nu)
\]

\( L \) runs over all supercell images.
Supercell dynamical matrix. Exact wave vectors.

Conventional dynamical matrix:
\[
D(k; \mu, \nu) = \frac{1}{\sqrt{M_\mu M_\nu}} \sum_m \Phi(0, \mu; m, \nu) \exp\{-2\pi i k \cdot [R(0, \mu) - R(m, \nu)]\}
\]

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

These two matrices are equal if
\[
D^{(SC)}(k; \mu, \nu) = D(k; \mu, \nu)
\]

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

\[
\exp\{-2\pi i k_s \cdot L\} = 1
\]

At wave vectors \(k_s\) the phonon frequencies are “exact”, provided the supercell **contains the complete list of neighbors**.

Wave vectors \(k_s\) are commensurate with the supercell size.
Exact wave vectors

1x1x1

Exact: $\Gamma$

2x2x2

Exact: $\Gamma, X, M, R$

3x3x3

Exact: $\Gamma$

X

$\Gamma$

M
Phonon dispersions + density of states

GeO$_2$ P4$_2$/mnm

Frequency $\omega$

Total + Germanium

Total + Oxygen
Thermodynamic functions of phonon vibrations

Internal energy:

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

Free energy:

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

Entropy:

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

Heat capacity $C_v$:

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

Thermal displacements:

$$B_{ij}(\mu) = \langle U_i(\mu) U_j(\mu) \rangle$$

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

  - \(\rightarrow\) **force constants** between all atoms in the supercell
  - \(\rightarrow\) **dynamical matrix** for arbitrary \(q\)-vectors
  - \(\rightarrow\) **phonon-dispersion** ("bandstructure") using **PHONON** (K.Parlinski)
- 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
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)
    - $\text{abs}(F_x)$ 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 $\rightarrow$ phase transition