

Introduction to WIEN2k package

Y. Melikhov

Group of X-ray Spectroscopy and Microanalysis,
Laboratory of X-ray and Electron Microscopy Research,
Institute of Physics, PAS



Acknowledgements



ALBA Synchrotron Light facility located in Cerdanyola del Vallès, Spain



This project is co-funded by the European Union within the framework of the EAgle Project No. 316014 Capacities/Research Potential FP7-REGPOT-2012-2013-1



Centre for the Computational Science at Interdisciplinary Centre for Mathematical and Computational Modelling, Warsaw, Poland



The program package WIEN2k allows to perform electronic structure calculations of solids using density functional theory (DFT).
First Polish workshop and the 22-nd workshop in the history of the WIEN2k code: "WIEN2k and SPECTROSCOPY: HANDS-ON WORKSHOP" which was held at Institute of Physics PAS, Warsaw, Poland in Fall 2014.



Group of X-ray Spectroscopy and Microanalysis, Laboratory of X-ray and Electron Microscopy Research, Institute of Physics, PAS:
Dr. Iraida Demchenko, Mr. Yevgen Syryanyy, Dr. Pawel Rejmak

Acknowledgements

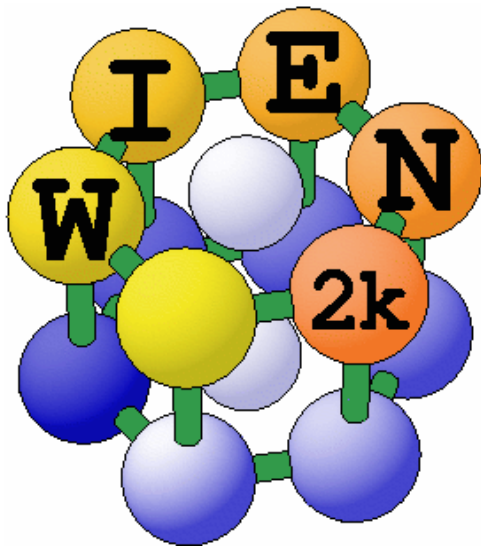


The program package WIEN2k allows to perform electronic structure calculations of solids using density functional theory (DFT).

First Polish workshop and the 22-nd workshop in the history of the WIEN2k code: “WIEN2k and SPECTROSCOPY: HANDS-ON WORKSHOP” which was held at Institute of Physics PAS, Warsaw, Poland in Fall 2014.

The material used in this lecture is based (almost entirely) on the following lectures delivered during “WIEN2k and SPECTROSCOPY: HANDS-ON WORKSHOP” which was held at Institute of Physics PAS, Warsaw, Poland in Fall 2014:

- “[Introduction to WIEN2k package](#)” by P. Blaha
- “[Installation of WIEN2k and parallelization](#)” by P. Blaha
- “[Large scale applications with WIEN2k](#)” by P. Blaha
- “[DFT and the concepts of \(L\)APW+lo](#)” by K. Schwarz
- “[Magnetism \(FM, AFM, FSM\)](#)” by K. Schwarz



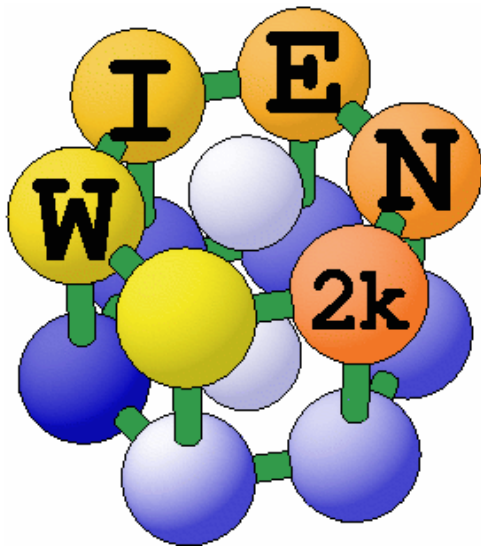
WIEN97: ~500 users
WIEN2k: ~2350 users

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

**Peter Blaha
Karlheinz Schwarz
Georg Madsen
Dieter Kvasnicka
Joachim Luitz**

November 2001
Vienna, AUSTRIA
Vienna University of Technology

<http://www.wien2k.at>



WIEN2k: ~2350 users

Current developers of WIEN2k code

Peter Blaha
Karlheinz Schwarz
Robert Laskowski
Laurence Marks
Fabien Tran
Elias Assmann

Current version of WIEN2k code

WIEN2k_14.2

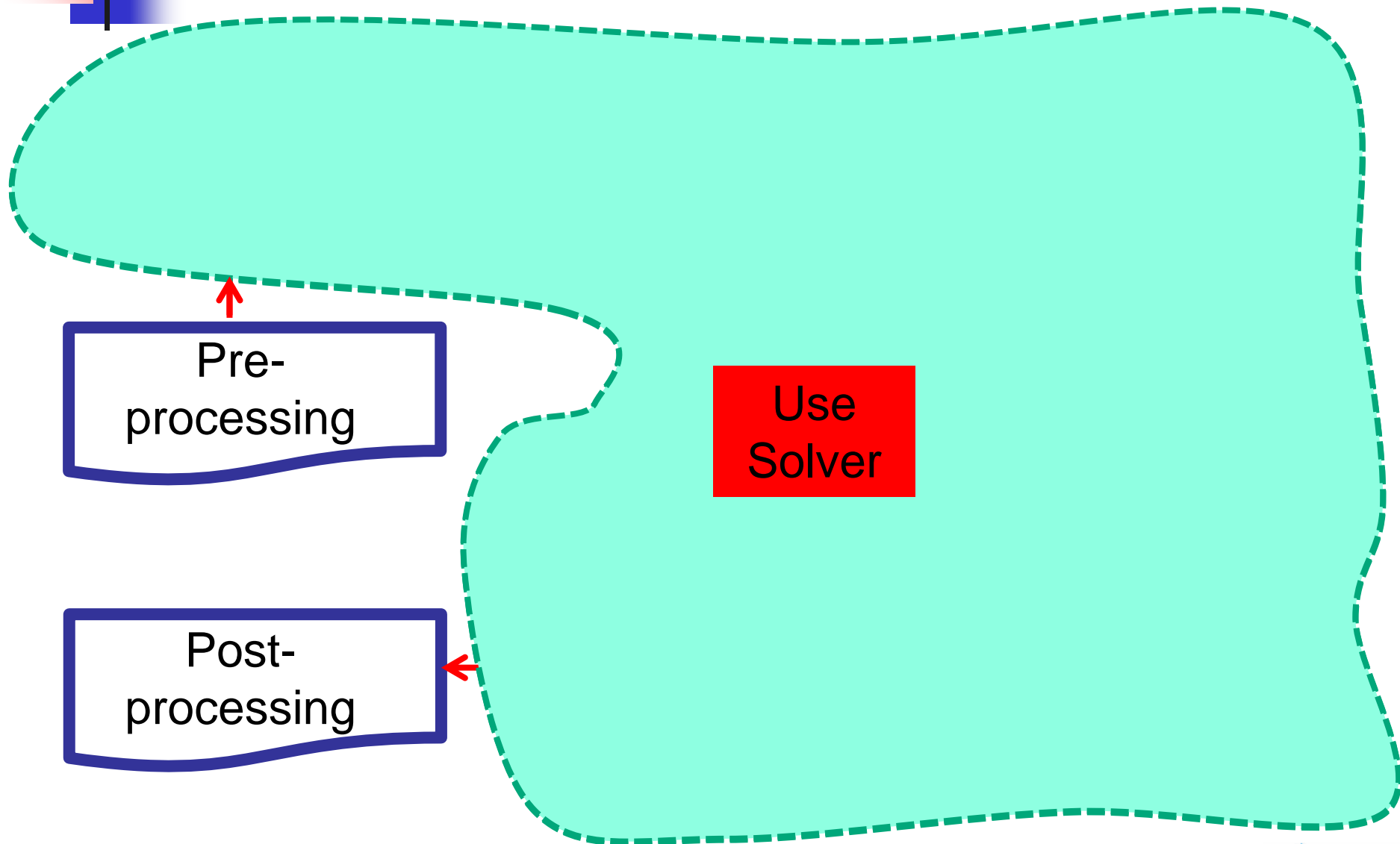
<http://www.wien2k.at>



General remarks on WIEN2k

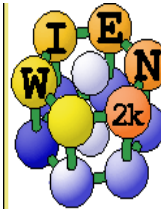
- WIEN2k consists of many independent F90 programs, which are linked together via C-shell scripts.
- You can run WIEN2k using any www-browser and the w2web interface, but also at the command line in terminal
- Each „case“ runs in his own directory `./case`
- The „master input“ is called `case.struct`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

General steps to solve



w2web GUI

- **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
- **The rest**
 - Use terminal



Execution >>

StructGen™
initialize calc.
run SCF
single prog.
optimize(V,c/a)
mini. positions

Utils. >>

Tasks >>

Files >>

struct file(s)
input files
output files
SCF files

Session Mgmt. >>

change session
change dir
change info

Configuration

Usersguide

html-Version
pdf-Version

Idea and realization
by

Session: TiC

/area51/pblaha/lapw/2005-june/TiC

StructGen™

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

Save Structure

Title: TiC

Lattice:

Type: F

P
F
B
CXY
CYZ
CXZ
R
H
1_P1

Spacegroups from
Bilbao Cryst Server

Lattice parameters in Å

a=4.328000038 b=4.328000038 c=4.328000038
 $\alpha=90.000000$ $\beta=90.000000$ $\gamma=90.000000$

Inequivalent Atoms: 2

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

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

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

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

Structure

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) $2a$

O: (0.304,0.304,0)

Wyckoff position: $x, x, 0$

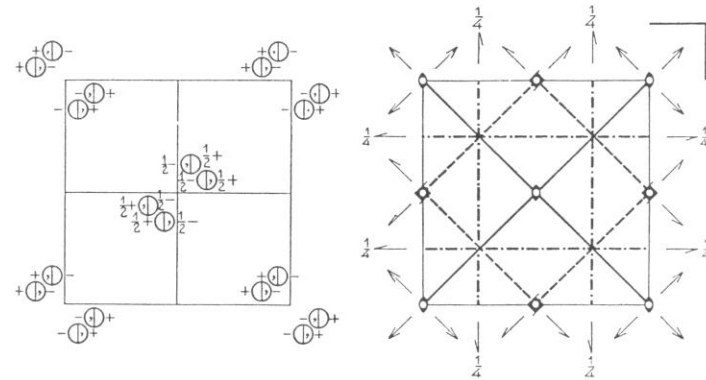
$4f$

$P4_2/mnm$
 D_{4h}^{14}

No. 136

$P4_2/m 2_1/n 2/m$

$4/m m m$ Tetragonal



Origin at centre (mmm)

Number of positions,
Wyckoff notation,
and point symmetry

Co-ordinates of equivalent positions

Conditions limiting
possible reflections

General:

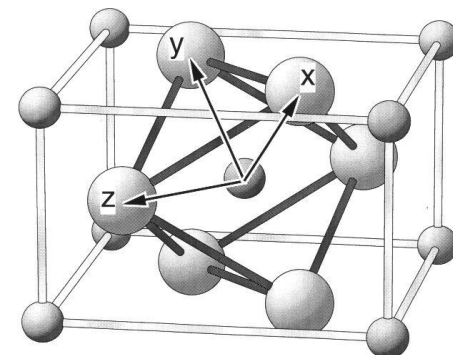
hkl : No conditions
 $hk0$: No conditions
 $Ok l$: $k+l=2n$
 hhl : No conditions

Special: as above, plus

no extra conditions

hkl : $h+k=2n$; $l=2n$

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} ; $\bar{y}, \bar{x}, \bar{z}$; $\frac{1}{2}+y, \frac{1}{2}-x, \frac{1}{2}-z$; $\frac{1}{2}-y, \frac{1}{2}+x, \frac{1}{2}-z$.
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} ; $\bar{x}, \bar{x}, \bar{z}$; $\frac{1}{2}+x, \frac{1}{2}-x, \frac{1}{2}-z$; $\frac{1}{2}-x, \frac{1}{2}+x, \frac{1}{2}-z$.
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	$0, \frac{1}{2}, z$; $0, \frac{1}{2}, \bar{z}$; $0, \frac{1}{2}, \frac{1}{2}+z$; $0, \frac{1}{2}, \frac{1}{2}-z$; $\frac{1}{2}, 0, z$; $\frac{1}{2}, 0, \bar{z}$; $\frac{1}{2}, 0, \frac{1}{2}+z$; $\frac{1}{2}, 0, \frac{1}{2}-z$.
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: $\pm(1/8, 1/8, 1/8)$, 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 ($x-1/2, \dots$)
 - in fcc (bcc) specify just one atom, not the others in $(1/2, 1/2, 0; \dots)$

■ „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: $RMT \sim 0.6$ bohr ($RKMAX \sim 3-4$)
- Do not change RMT in a „series” of calculations, RMT *equal* for *same* atoms

■ „save structure – save+cleanup“

Program structure of WIEN2k

■ `init_lapw`

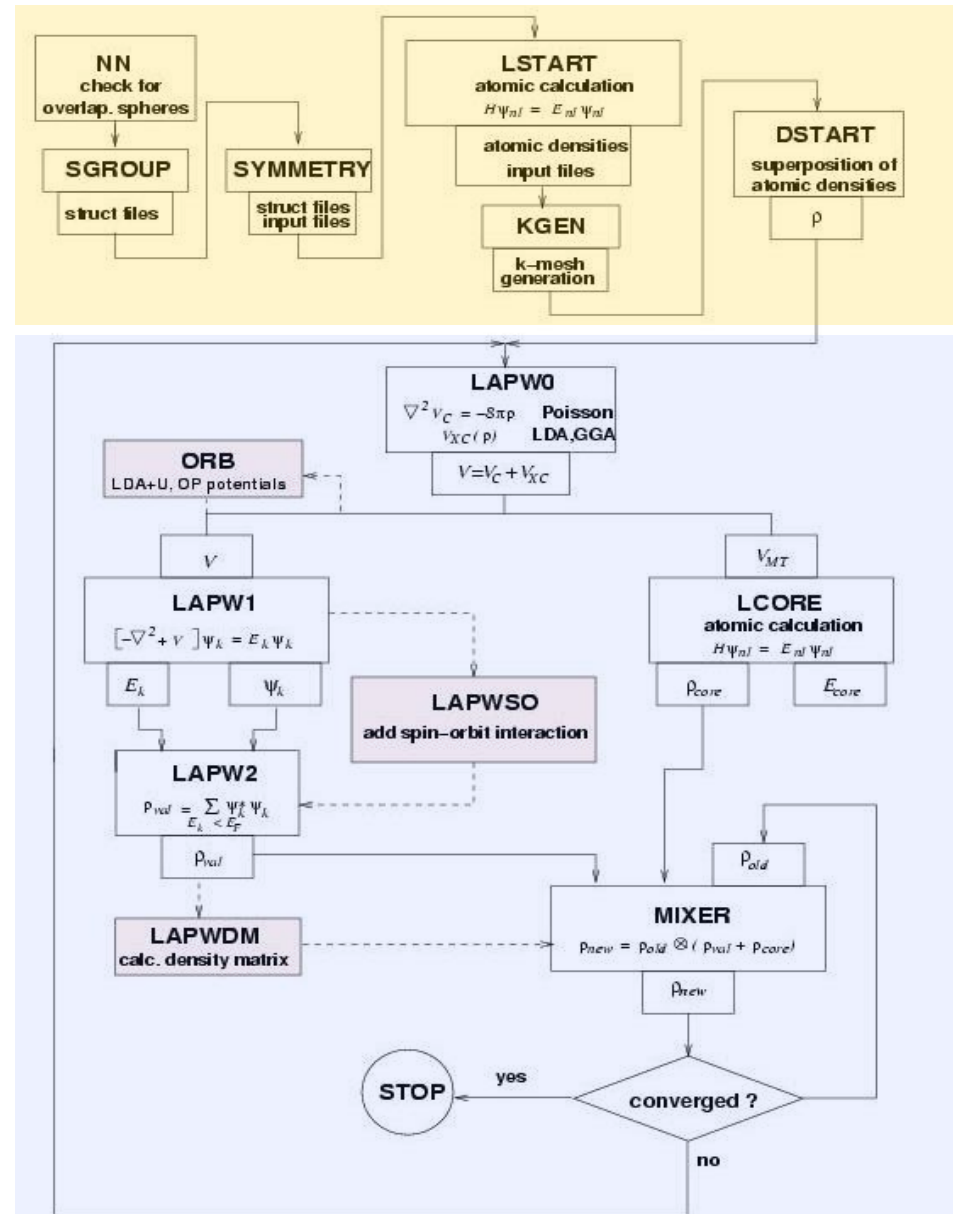
- *step-by-step or batch initialization*
- *symmetry detection (F, I, C-centering, inversion)*
- *input generation with recommended defaults*
- *quality (and computing time) depends on **k-mesh** and **R.Kmax** (determines #PW)*

■ `run_lapw`

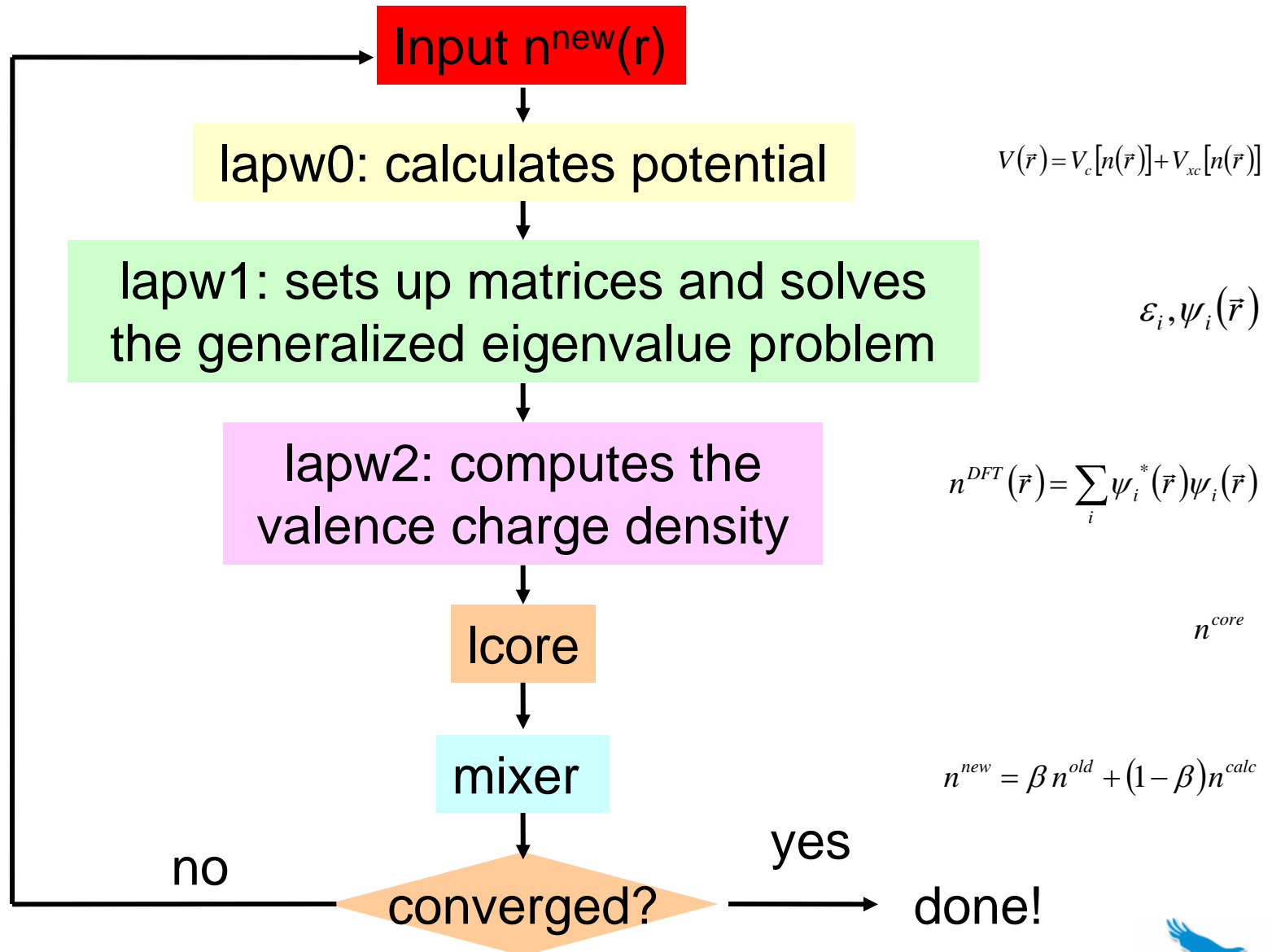
- *scf-cycle*
- *optional with SO and/or LDA+U*
- *different convergence criteria (energy, charge, forces)*

■ `save_lapw tic_gga_100k_rk7_vol0`

- *cp case.struct and clmsum files,*
- *mv case.scf file*
- *rm case.broyd* files*

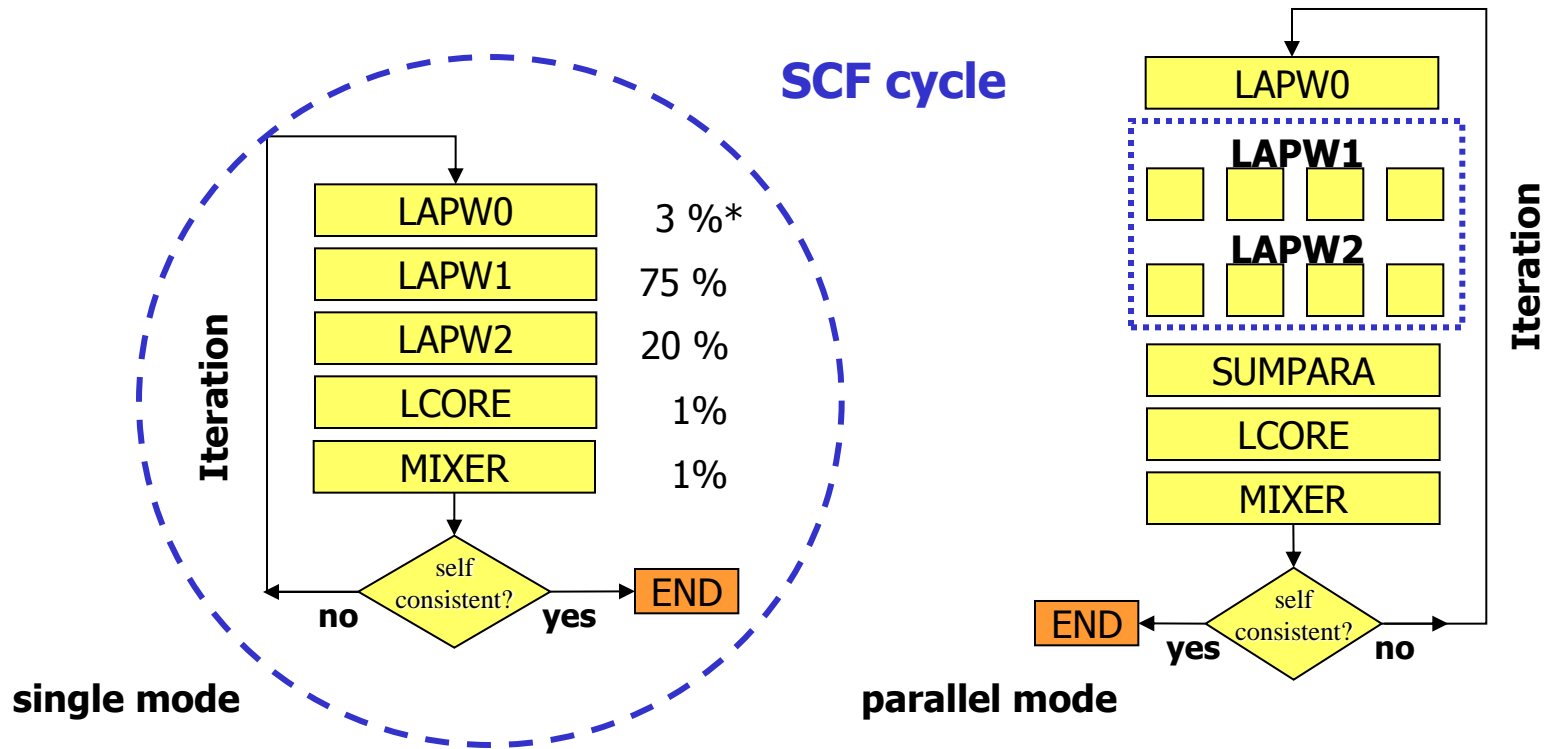


Flow Chart of WIEN2k (SCF)



Flow Chart of WIEN2k (SCF)

- individual FORTRAN programs linked by shell-scripts
- the output of one program is input for the next
- lapw1/2 can run in parallel on many processors



* fraction of total computation time

FP-(L)APW- lo

FP-(L)APW- lo method

For potential use:

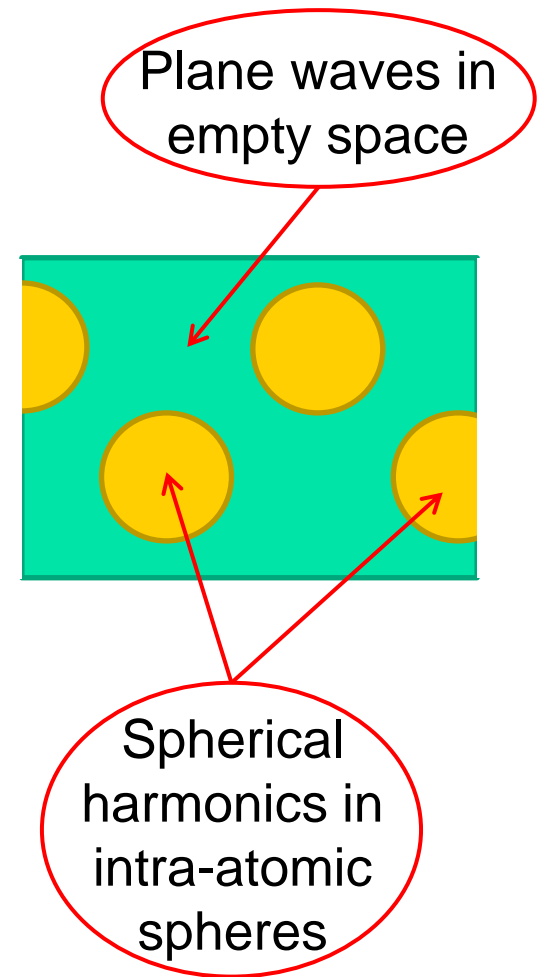
- FP, i.e. Full Potential

For basis functions one can use:

- LAPW (plus LO),
- APW- lo (plus LO),
- or mixed

Practically, the numbers which control the accuracy of the solution are:

- The cut-off for the plane waves: RK_{max}
- The cut-off for the angular functions L_{max}



RK_{MAX} parameter

- The convergence criterion in APW is the product of $R_{MT} \cdot K_{max}$
- http://www.wien2k.at/reg_user/faq/rkmax.html
- medium quality convergence for **smallest** atom:

- basis set scales with RK_{max}^3
- cputime scales with N_{PW}^3
- increasing Rk_{max} by 10 %
→ doubles cputime

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

START with **SMALLer** Rk_{max} (relaxation)

increase/test later

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	<i>convergence of total energy (Ry)</i>
■ -cc 0.0001	<i>convergence of charge distance (e⁻)</i>
■ -fc 1.0	<i>convergence of forces (mRy/bohr)</i>
■ -it (-it1,-it2 , -noHinv)	<i>iterative diagonalization (large speedup)</i>
■ -p	<i>parallel calculation (needs .machines file)</i>
■ -so	<i>add spin-orbit (only after „init_so“)</i>
■ <i>Spacegroups without inversion use automatically lapw1c, lapw2c (case.in1c,in2c)</i>	

■ 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.
- **Lecture notes from previous WIEN2k workshops**

case.in0 file

```
TOT      XC_PBE      ! MULT/COUL/EXCH/POT /TOT ;   VXC-SWITCH
NR2V      IFFT  8      ! R2V  EECE/HYBR IFFT  LUSE
  30  30 108  4.00  1 ! min IFFT-parameters, enhancement factor, iprint
0 0.0      (#of FK in E-field expansion, EFELD (Ry))
```

Functional	Authors	Year	indxc (case.in0)
LDA	Dirac, Slater ...	1930 - ...	XC_LDA (5)
GGA:			
PBE	Perdew et al	1996	XC_PBE (13)
WC	Wu, Cohen	2005	XC_WC (11)
PBEsol	Perdew et al.	2007	XC_PBESOL (19)
HTBS	Haas et al.	2011	EX_HTBS VX_HTBS EC_PBE VC_PBE (46)
TB-mBJ*	Tran, Blaha	2009	XC_MBJ; EX_GRR VX_GRR (28,50)
revTPSS**	Perdew et al.	2009	XC_REVTPSS (29)
MGGA_MS2**	Sun et al.	2013	XC_MGGA_MS 0.504 0.146 4.0

* only a potential ($E_{xc} = \text{LDA}$)

** only E_{xc} ($V_{xc} = \text{PBE}$)

case.in1 file

set E_f to $E_F - 0.2$ Ry

WFFIL

EF=0.634

(WFPRI, SUPWF)

Every atom

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

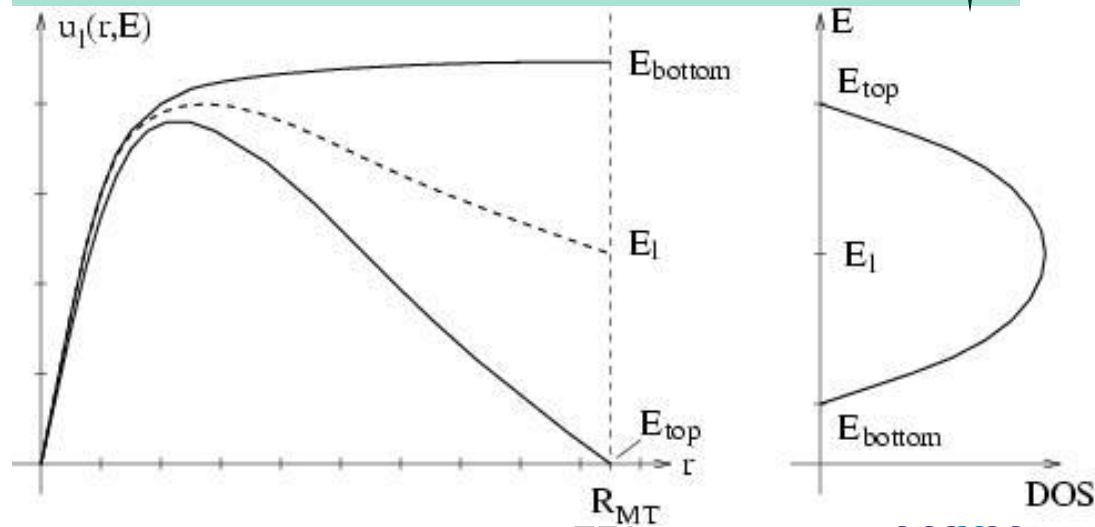
16

emin/emax; nband

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

$$\Phi_{K_n} = \sum_l^{lmax} A_{lm} u_l(E_l, r) Y_{lm}$$

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



case.klist, case.in2 files

```

■ GAMMA          0      0      0      40      1.0      IX, IY, IZ, IDIV, WEIGHT
■                1      0      0      40      6.0
■ ...
■ X              40      0      0      40      3.0
■ END
  
```

case.in2:

```

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

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

$$\rho(r) = \sum_G^{GMAX} \rho_G e^{iGr}$$

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*

- x qtl (instead of x lapw2 -qtl)

Session: TiC
/susi/pblaha/lapw/TiC

Density of states

x lapw2 -qtl

Calculate partial charges ☒ interactively

edit TiC.int

Edit input-file for TETRA

x tetra

Calculate partial DOS ☒ interactively

edit TiC.outputt

Check output of TETRA

dosplot

Plot DOS


Session: TiC
/susi/pblaha/lapw/TiC

File:

/susi/pblaha/lapw/TiC/TiC.int

continue with DOS

Save

Download this file: 

Header from TiC.qtl:

ATOM	1	tot,0,1,2,3,xdos(i,j),j=1,i,i=1,1xdos2)
ATOM	2	tot,0,1,2,D-eg,D-t2g,3

Title

-0.50 0.002 1.500 0.003
3

EMIN, DE, EMAX, Gauss-broadening(>;de)
NUMBER OF DOS-CASES specified below
atom, case=column in qtl-header, label

0	1	total
1	2	Atom1-s
2	5	Atom2-eg



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*
($\nabla \rho \cdot \vec{n} = 0$)
- *spin+orbital magnetic moments (spin-orbit / LDA+U)*

■ Hyperfine parameters

- *hyperfine fields (contact + dipolar + orbital contribution)*
- *Isomer shift*
- *Electric field gradients*
- *NMR chemical shifts*



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)

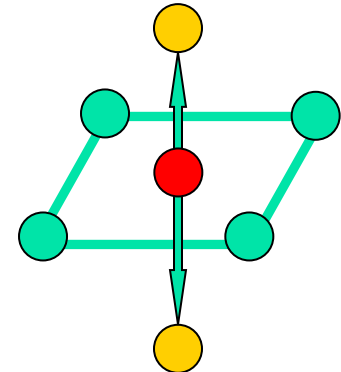
Properties with WIEN2k – III

■ advanced topics and developments

- *non-collinear magnetism (available on request: www.wien2k.at)*
- *transport properties (Fermi velocities, Seebeck, conductivity, thermoelectrics, ..): G. Madsen's BotzTrap code*
(see http://www.wien2k.at/reg_user/unsupported)
- *Bethe-Salpeter equation (for excitons, R.Laskowski, C.Ambrosch-Draxl)*
- *GW (M.Scheffler, Hong Jiang)*
- *Hartree-Fock (+Hybrid DFT-functionals)*
- *Berry phases (BerryPI by O.Rubel et al. (http://www.wien2k.at/reg_user/unsupported))*
- *NMR – Chemical shifts*
- *– Quadrupole splittings (also for Mössbauer spectroscopy)*

Structural optimization – I

- **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 add switches ("-min, -fc 1 -cc 0.01")
 - *execute optimize.job*
 - *plot (analyse) the results*
- **WIEN „preserves" symmetry:**
 - „Jahn-Teller" distortion:
 - when you start with a perfect octahedra, you will never get any distortion
 - → start with slightly distorted positions



Structural optimization – II

- Forces only for “free” structural parameters:
 - *NaCl: (0,0,0), (0.5,0.5,0.5) : all positions fixed by symmetry*
 - *TiO₂: 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*

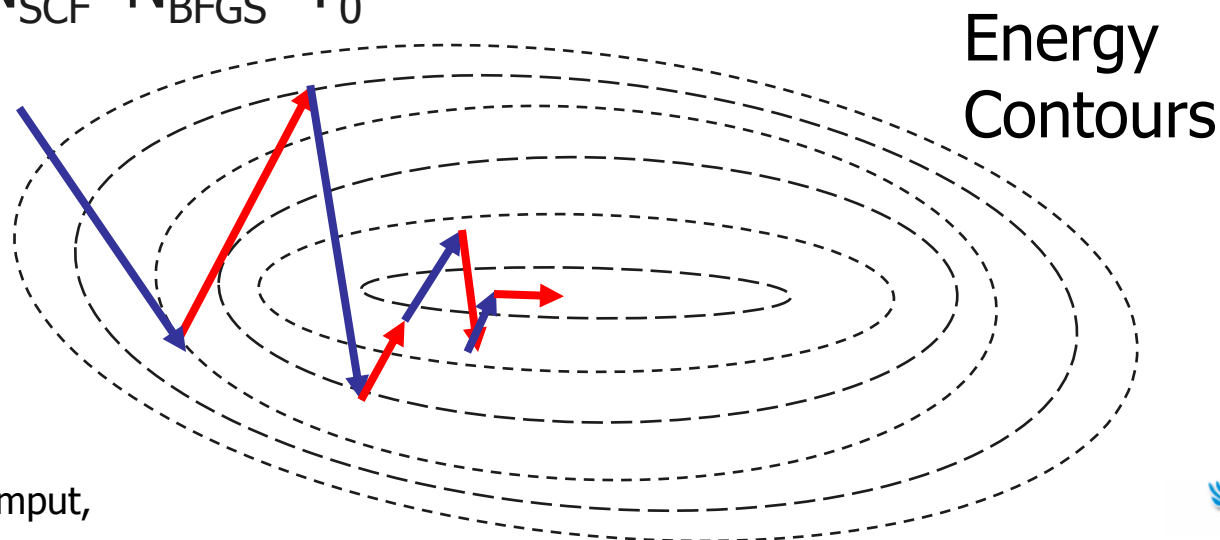
■ 200.	partial	
■ -130.	partial	
■ 140.	partial	
■ 135	partial	only $F_{\text{HF}} + F_{\text{core}}$
■ 120	partial	
■ 122	partial	forces converging
■ 121	partial	→ changes “TOT” to “FOR” in case.in2
■ -12.3	total	$F_{\text{HF}} + F_{\text{core}} + F_{\text{val}}$, only this last number is correct
- Forces are useful for
 - *structural optimization (of internal parameters)*
 - *phonons*

Structural optimization – III

Straightforward way (PORT, NEW1)

- 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 $N_{\text{SCF}} * N_{\text{BFGS}} * T_0$

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



Structural optimization – IV

Straightforward way (PORT, NEW1)

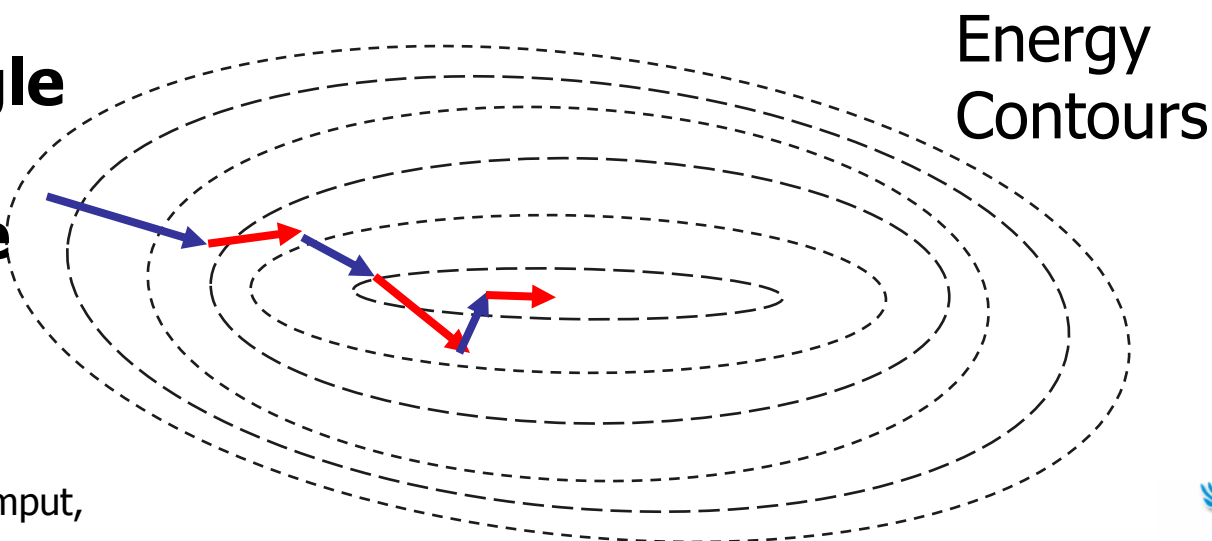
- `/home/pblaha/tio2> min_lapw [-p -it -sp] [-j "run -fc 1 -p -it"] [-NI]`
 - *performs scf-cycle for fixed positions*
 - *get forces and move atoms along forces (building an approximate Hessian) and writing a new case.struct file*
 - *extrapolate density (case.clmsum)*
 - *perform next scf cycle and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

Structural optimization – V

Alternative way (MSR1)

- Treat the **density** and **atomic positions** *all* at the same time
- Approximate forces are produced but each SCF cycle
- Takes more SCF cycles than for ordinary convergence but finds the optimized parameters faster

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





Structural optimization – VI

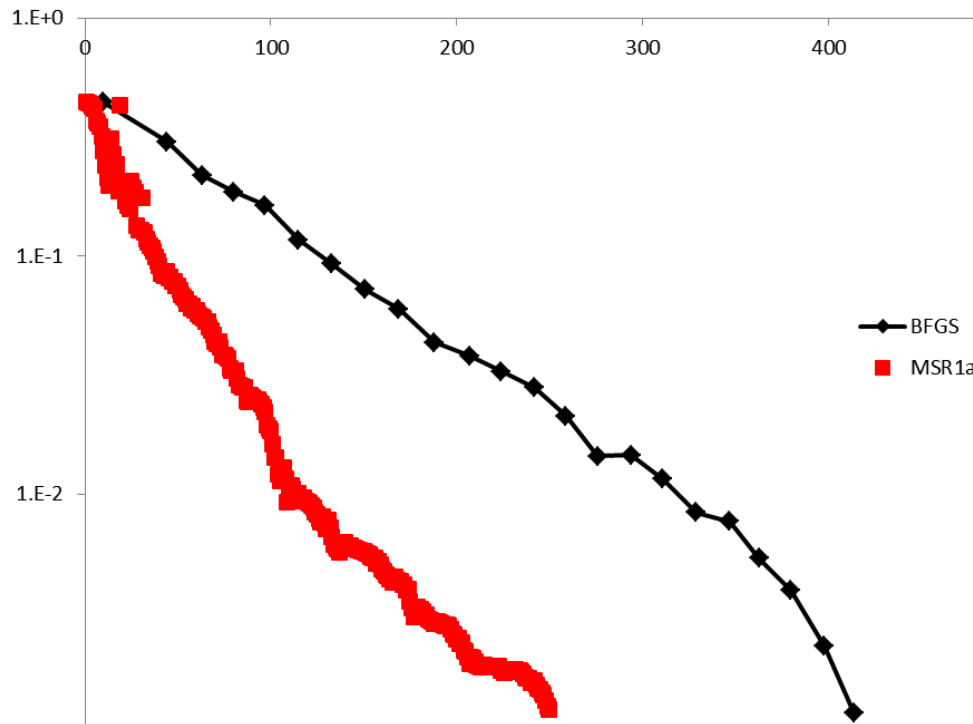
Alternative way (MSR1)

- `run_lapw -min -fc 1.0 -cc 0.001 -ec 0.0001 [-it -noHinv -p]`
 - *this sets edit „**MSR1a**“ in `case.inm`*
- 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).

Structural optimization – VII

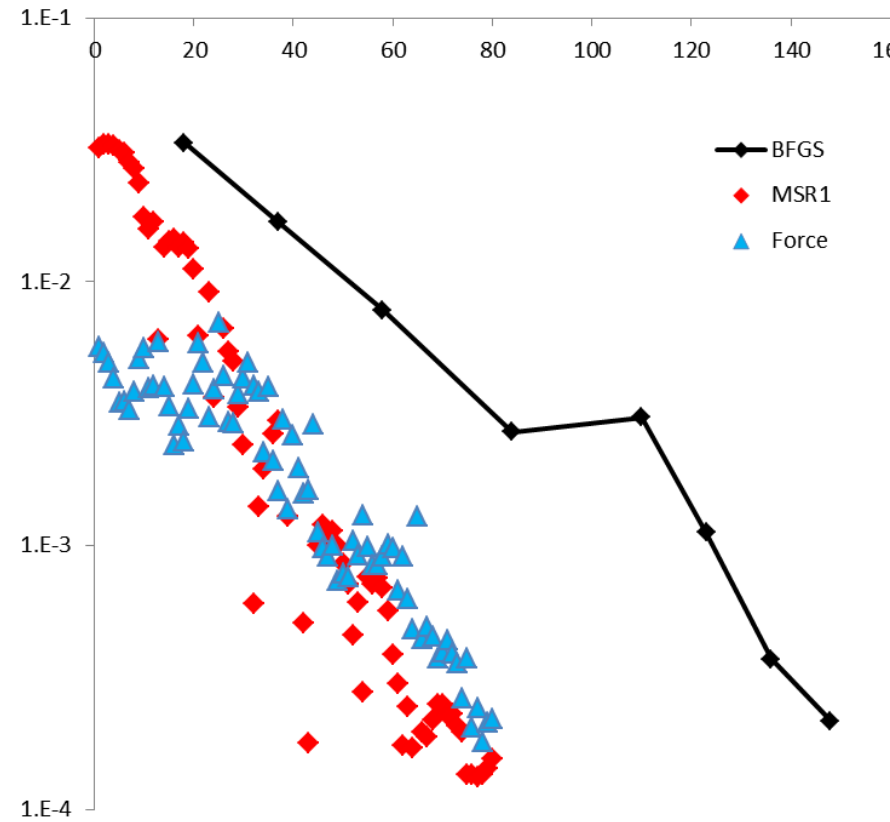
Comparison of the 2 methods

52 atoms, MgO (111)+H₂O



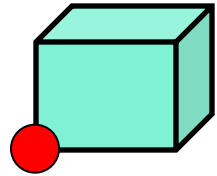
J. Ciston, A. Subramanian, L.D. Marks, PhRvB, 79 (2009) 085421.

108 atoms AlFe

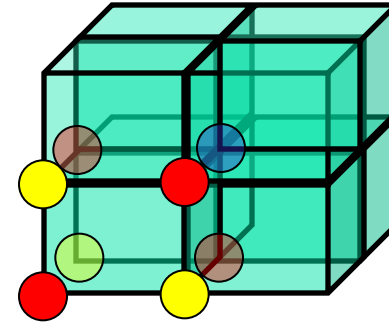


Lyudmila V. Dobysheva (2011)

Supercells – I



$2 \times 2 \times 2 = 8$ 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

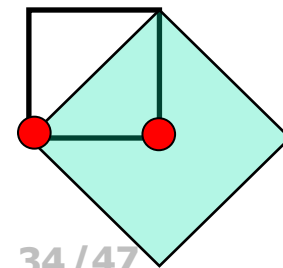
no

no

yes

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

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





Supercells – II

- 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
```

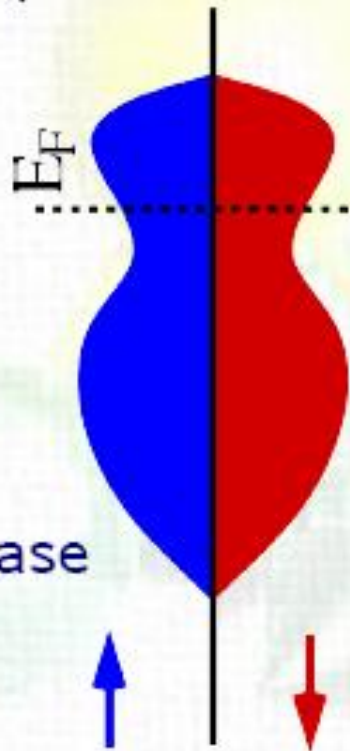
```
helpstruct
```

Calculations with Magnetism – I

- Wien2k can only handle collinear or non-magnetic cases

run_lapw script: DOS

```
x lapw0
x lapw1
x lapw2
x lcore
x mixer
```

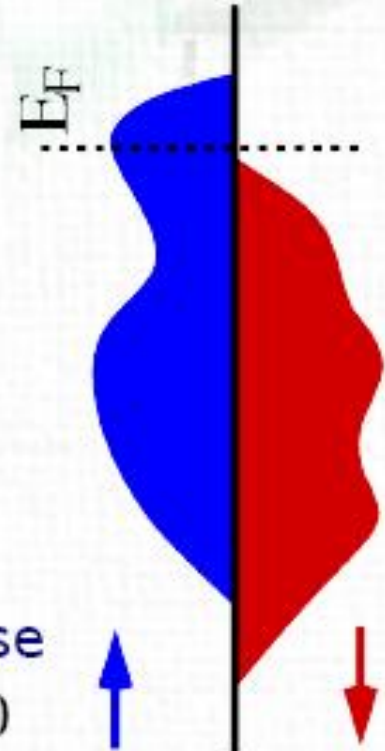


non-magnetic case

$$m = n_{\uparrow} - n_{\downarrow} = 0$$

run_lapw script: DOS

```
x lapw0
x lapw1 -up
x lapw1 -dn
x lapw2 -up
x lapw2 -dn
x lcore -up
x lcore -dn
x mixer
```



magnetic case

$$m = n_{\uparrow} - n_{\downarrow} \neq 0$$



Calculations with Magnetism – II

- *runsp_lapw* script (unconstrained magnetic calc.)
 - runs lapw1/2 for both spins *independently*
 - case.scf contains extra information:
 - grep :MMT case.scf (for total moment)
 - grep :MMI case.scf (for atomic moments)
 - grep :HFF case.scf (for hyperfine fields)
- *runfsm_lapw -m value* (constrained moment calc.)
 - for difficult to converge magnetic cases or simply to constrain a moment (→ 2 Fermi-energies → external magnetic field)
- *runafm_lapw* (anti-ferromagnetic calculation)
 - calculates only spin-up, uses symmetry to generate spin-dn

Calculations with the MBJ potential

1. Prepare the input files for an usual PBE (or LDA) calculation
2. run scf cycle (+ structure optimization,)
3. save_lapw case_pbe
3. init_mbj_lapw (phase 1, creates case.inm_vresp, sets R2V in case.in0)
4. run(sp)_lapw -i 1 (Run one PBE cycle to create case.vresp and case.r2v)
5. rm *.broyd*
6. init_mbj_lapw (phase 2)
 - sets indx=XC_MBJ in case.in0 and generates case.in0_grr with indx=50;
 - select mBJ-parameters, see Koller et al. PRB 85, 155109 (2012)
7. run(sp)_lapw -i 80 (mBJ calculations need more cycles than PBE)

WIEN2k – Computer system requirements – I

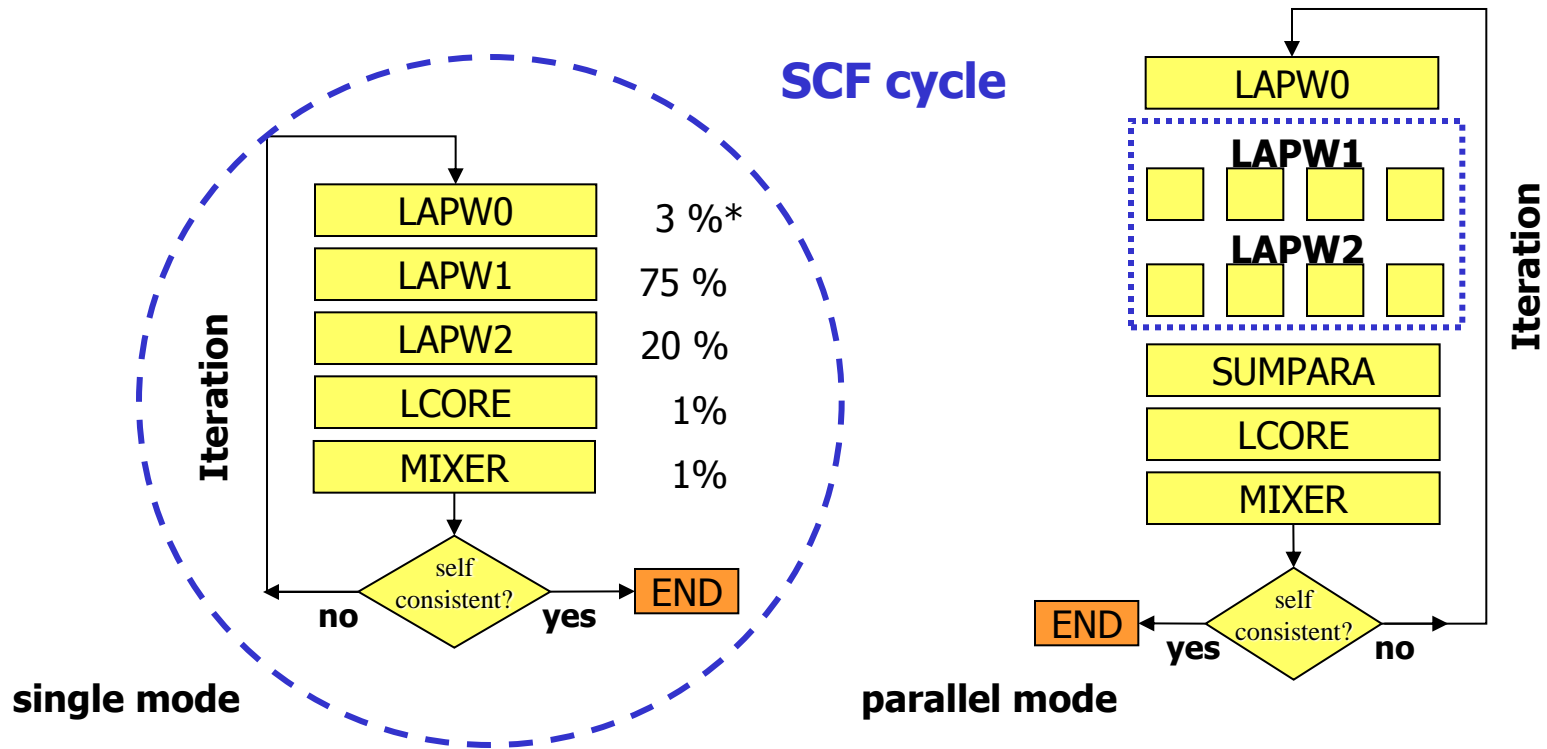
- WIEN2k runs on any **Linux** platform from PCs, Macs, workstations, clusters to supercomputers
 - *Intel **I7** quad (six)-core processors with fast memory bus (1.5-3 Gb/core, Gbit-network, SATA disks). 1000 € /PC,*
 - *with a few such PCs you have a quite powerful cluster (k-parallel)*
 - *60 - 100 atom / cell, requires 2-4 Gb RAM/core*
 - *installation support for many platforms + compiler*
- **Fortran90** (dynamical allocation, modules)
 - *many individual modules, linked together with C-shell or perl-scripts*
- *web-based GUI – w2web (perl)*
- **f90 compiler (ifort, gfortran), BLAS-library (mkl, gotolib), FFTW, perl5, ghostscript (+jpg), gnuplot(+png), tcsh, pdf-reader, www-browser, octave, opendx, python**
- **wannier90, Xcrysden, VESTA, phonopy, boltztrap, ... (unsupported SW)**

WIEN2k – Computer system requirements – II

- Installation on a cluster – not for a faint-hearted person
- Read instructions
- recommendation: **Intels Fortran compiler** (includes mkl)
free for non-commercial (but not for academic), www.intel.com
 - *which ifort* → tells you if you can use ifort and which version you have
 - **ifort 14** (vers. 8.0, early 12.x and even some recent versions are *buggy*)
- it does NOT make sense to invest in new hardware but use a „free“ compiler, which is 2 times slower than ifort+mkl
- w2web: acts as **webserver** on a userdefined (high) **port**
 - *behind firewall create a „ssh-tunnel“: `ssh -fNL 2000:host:2000 user@host`*
- Read instructions

Flow Chart of WIEN2k (SCF)

- individual FORTRAN programs linked by shell-scripts
- the output of one program is input for the next
- lapw1/2 can run in parallel on many processors



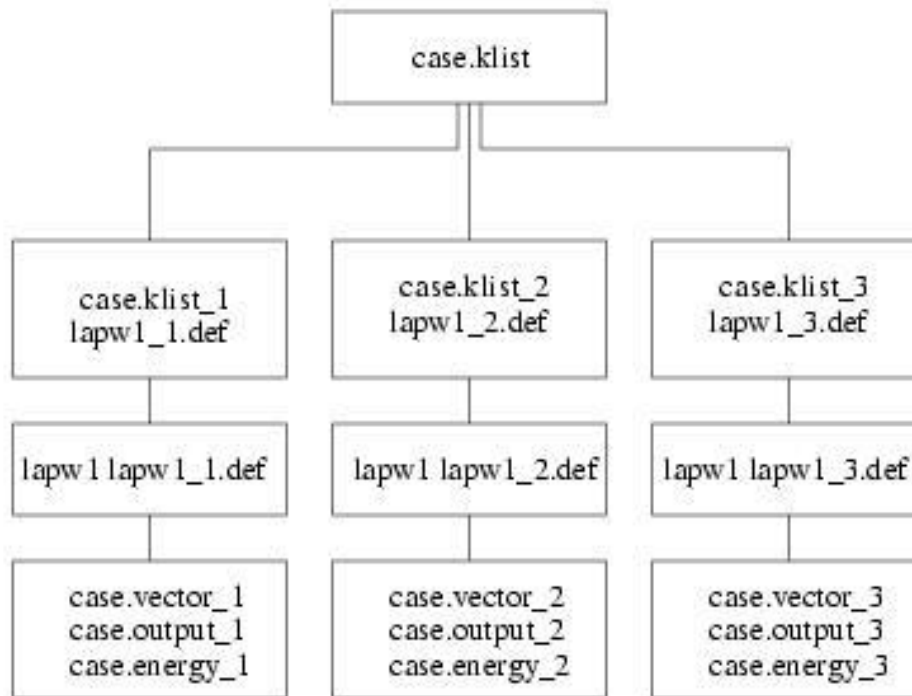
* fraction of total computation time

WIEN2k – k-point parallelization – I

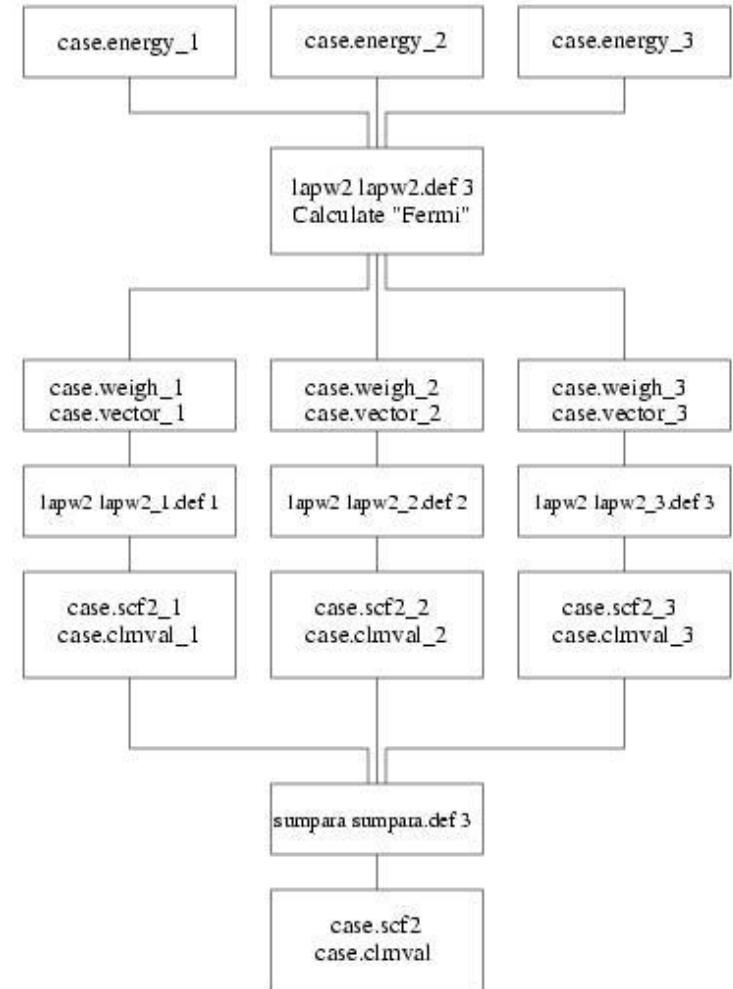
- **very efficient parallelization** even on loosely coupled PCs (**slow** network):
 - **common NFS filesystem** (*files must be accessible with the same path on all machines; use **/host1** as data-directory on **host1***)
 - *.machines file must be created by a user*
 - lapw0:host1 host1 ... (Repeat as many times as processors)
 - 1:host1 (speed:hostname) (Repeat as many times as processors)
 - 2:host2 - Do not use! Only on a cluster with PCs having different speed!
 - granularity:1 - Do not use! Only on a cluster with PCs having different speed!
 - extrafine:1 (rest in junks of 1 k)
 - *case must fit into memory of one PC !*
 - *high NFS load: use local \$SCRATCH directory (only with commensurate k-points/hosts)*
 - **\$OMP_NUM_THREADS=2** (*parallel diag. (mkl) on multi-core CPU*)

WIEN2k – k-point parallelization – II

lapw1para



lapw2para



WIEN2k – batch system

- submit a script to a queuing system (SLURM, PBS, SGE, ...)
- you can only specify total number of cores:
 - **`#$ -pe mpich 32`** *(specify 32 cores, but you don't know the hosts)*
- get the machine names and write **.machines** on the fly:

```
set mpisize_per_k=16
set proclist= `cat $hostfile_tacc`           # this will be different on your computer
set nproc= `cat hostfile_tacc | wc -l`
set i=1
while ($i <= $nproc )
echo -n '1:' >>.machines
@ i1 = $i + $mpisize_per_k
@ i2 = $i1 - 1
echo $proclist[$i-$i2] ':1' >>.machines
set i=$i1
end
echo 'granularity:1' >>.machines
echo 'extrafine:1' >>.machines
```

- you can combine k- and mpi-parallelization (\$mpisize_per_k)
 - *32 cores: 2 k-points, 16 mpi-jobs/k-point*

WIEN2k – iterative diagonalization for big cases

■ run_lapw -p -it -noHinv

cycle 1 (Thu Oct 31 07:20:53 CET 2013) (40/99 to go)

> lapw0 -p (07:20:53) starting parallel lapw0 at Thu Oct 31 07:20:53 2013

----- .machine0 : 64 processors

264.604u 21.742s 0:40.76 702.5% 0+0k 591784+49768io 369pf+0w

> lapw1 -up -p -orb (07:21:34) starting parallel lapw1 at Thu Oct 31

-> starting parallel LAPW1 jobs at Thu Oct 31 07:21:34 CET 2013

running LAPW1 in parallel mode (using .machines)

r09n30 r09n30 r09n30

6.558u 1.796s 29:08.54 0.4% 0+0k 16+520io 0pf+0w

....

cycle 3 (Thu Oct 31 07:50:53 CET 2013) (40/99 to go)

...

> lapw1 -it -up -p -orb -noHinv (09:31:52) starting parallel lapw1 at ...

3.411u 0.908s 14:18.31 0.5% 0+0k 72+536io 0pf+0w

...



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.
- **Lecture notes from previous WIEN2k workshops**