Calculation of crystal field parameters with Wannier functions: manual

P. Novák
Institute of Physics of ASCR, Cukrovarnická 10, 162 53 Prague 6, Czech Republic

(Dated: March 12, 2014)

PACS numbers:
Keywords:

This is manual and documentation of CFP calculation. Illustration for $\mathrm{NdCoO_3}$ system (inputs and outputs of the steps I - VI) is included in RECFP directory. The files are in subdirectory Nth_step/*case*; *case* $\equiv$ ndcoo3. For steps I and II consult WIEN2k userguide.

I. FIRST STEP: CALCULATION WITH f STATES IN CORE

This is standard WIEN2k nonspin-polarized calculation.
Start with init_lapw, but first case.struct must be created.
WIEN has several possibilities how to do it:

- *makestructure*
- using *cif* file
- using *w2web*
- modifying some of the examples in *SRC_example_struct_files*

For the unit cell parameters it is usually better to use experimental values, though WIEN can also be applied (userguide sect. 5.3). Internal lattice parameters may be optimized (required in case of an impurity) using WIEN *mini* program (userguide sect. 5.3). Atomic radius of R should be as large as possible, at least ~ 2 a.u.
With *case.struct*, the *case.inst* (input to the initial atomic program) is to be created by:
instgen
Proceed with
init_lapw
Usually we choose GGA for exchange-correlation potential (options 11, 13 or 19) and -6 Ry for energy to separate core and valence states. (in some cases this energy has to be increased). The following inputs created by init_lapw must be modified:

- in *case.in*1 (*case.in*1*c* in systems without the inversion symmetry, which leads to the complex eigenvectors) expansion energy of R(f) states should be put high ~ 3 Ry, and step put equal to 0.
- in *case.inc* f states of R should be included and potential well ~ 1.5 Ry inserted for the R atom.
- in *case.in*2 (*case.in*2*c* in systems without the inversion symmetry) number of electrons must be reduced by actual number of R(f) electrons, which were put in the core. Also we increase as a rule $GMAX$ (last line) from 12 to 16.

When the *init_lapw* calculates k-list and the script ask you:
Shift of k-mesh allowed. Do you want to shift: (0=no, 1=shift)
enter 0.
Proceed with:
run_lapw -cc 0.00001 -p
cc charge convergency, *p* parallel execution.
Well converged results should be stored in separate directory e.g. *fcore*.

II. SECOND STEP: f STATES INCLUDED IN VALENCE STATES

The valence states, which will be allowed to hybridize with R(f) should be selected. In oxides these states include as a rule oxygen $2p$ and $2s$, other states which might hybridize with R(f) could be added. Shift energy Δ for these selected states should be chosen (in oxide perovskites choosing Δ = -0.6 Ry for oxygen 2s, 2p gave good results). In accord with this selection orbital potential *case.vorb* should be created and *case.in*1 (*case.in*1*c* for systems without inversion) modified.

- modification of *case.in*1 (*case.in*1*c*):
  - leave out all semicore states like R(5s, 5p) etc., providing this does not lead to ghostbands.
  - For states selected as valence, put reasonable value for expansion energy (0.3 Ry for oxygen 2p and 2s).
  - For f states expansion energy should be find out, as described below.
  - States which are not allowed to hybridize should have high expansion energy $\sim$ 3 Ry and step 0.
- orbital potential *case.vorb*
  orbital potential is a diagonal matrix and it should be created by hand. For selected valence states put Δ on the diagonal. Typical value of Δ for oxygen $2s, 2p$ states is between -0.3 and -0.7 Ry. It may be necessary to put large positive energy Δ' (e.g. 6 Ry) on diagonal elements of states which are not allowed to hybridize (e.g. Co 3d). To ease the construction of *case.vorb* look in directory RECFP/2nd_step/vorb_help.

Finding out f states expansion energy E_0:
run *lapw*1 in Γ point only with above *case.in*1*c* and $E_0 \sim$ 0.3 Ry, step 0.001 Ry (modify *case.klist* file by putting 'END' as a second line).
x lapw1 -orb
Find in the *lapw*1 output *case.output*1 whether WIEN found the f states (it should be seven closely spaced levels) lying in the gap. If WIEN did not find f states or their energy is evidently false, change E_0, or/and change Δ.

III. THIRD STEP: w2w AND wannier90 CALCULATIONS

Proceed according to *w2w* and *wannier*90 usersguides. The necessary programs can be downloaded from
wien2k.ac.at/regular users/Unsupported goodies/wien2wannier and
wannier90.org

The following sequence of commands may be used:

step	command	explanation
1	x kgen -fbz cp case.klist case.klist_w90	creates k mesh over whole Brillouin zone
2	x lapw1 -orb (-p)	solve the eigenproblem. -p parallel execution giving n vectors.
3	join_vectorfiles (-c) case n	from n vectors make single file.
4	write_w2win case	write input for w2win.
5	write_win case	write input for wannier90.
6	wannier90.x -pp case	preliminary wannier90 run.
7	write_w2wdef case	write definition file for w2w.
8	w2wr (w2wc) w2w.def	run w2w (w2wc for complex eigenvectors).
9	wannier90.x case	wannierization.

Hints to step 4:

- write_w2win offers form and sequence of the Wannier seed functions. Do not change this form and sequence, as they are used in the next step to calculate CFP.
- write_w2win asks for first and last index of f eigenfunction. These indexes are best to find in *case.scf*1 - lapw1 output of step 2. They should form a group of seven closely spaced levels, which are well separated from the valence and conduction bands. If this is not the case, Δ must be changed.

After finishing step 9, inspect *case.wout*. If the wannier90 converged to correct results, the f functions spreads should be of order of 1 or smaller, depending on how large is the hybridization.

IV. FOURTH STEP: POSTPROCESSING OF wannier90 RESULTS

The program *bkq*, necessary for this step is in:
RECFP/4th_step/programs

wannier90 most important result for CFP calculation is the hamiltonian transformed to Wannier basis. It is saved in file *case_hr.dat*.

Program *bkq* is used to determine the crystal field parameters. This program needs two subroutines: *trafo.f*, *transf.f* and the *lapack* library. The compilation is made using the *Makefile*. In addition *bkq* also needs explicit form of spherical tensors. These were extracted from modified output of *lanthanide* package (see next section for *lanthanide* description) and written on file *Cqk*. This file may be found in RECFP/4th_step

To calculate the CFP:

```
cp Cqk fort.90
cp case_hr.dat fort.7
bkq<case.inbkq>case.outbkq
cp fort.8 case.cfp
```

The input *case.inbkq* consists of single line:
mult iat inhr ipr iprt iprf
mult: multiplicity of R atom (number of equivalent R atoms)
iat: index of atom for which CFP will be calculated
inhr=0 for older wannier90 versions, inhr=1 for more recent versions.
inhr is required because of different format of *case_hr.dat* in older and newer versions of wannier90.
ipr,iprt,iprf printing options for main program, trafo and transf. The bigger are the values of printing options the longer is the output.
Example:
4 1 0 2 0 0

The program first extracts the local $4f$ hamiltonian $\hat{h}_{loc}$ (7 x 7 hermitean matrix) from *case_hr.dat* ($fort.7$). Next $\hat{h}_{loc}$ is expanded in a series of spherical tensor operators $\hat{C}_q^{(k)}$:

$$\hat{h}_{loc} = \sum_{k=2,4,6} \sum_{q=-k}^{k} B_q^{(k)} \hat{C}_q^{(k)}. \tag{1}$$

To find $B_q^{(k)}$ we make use of the fact that $\hat{C}_q^{(k)}$ form complete orthogonal set of operators in the subspace of $4f$ states. Then

$$B_q^{(k)} = \frac{1}{n_{k,q}} \sum_{i=1}^{7} \sum_{j=i}^{7} h_{loc}^{i,j} C_q^{(k),(i,j)}, \tag{2}$$

where $n_{k,q}$ is the normalizing factor:

$$n_{k,q} = \sum_{i=1}^{7} \sum_{j=i}^{7} C_q^{(k),(i,j)} C_q^{(k),(i,j)}. \tag{3}$$

Using above equations *bkq* calculates all independent CFP having odd k (three real, twelve complex). Understandably those $B_q^{(k)}$ not allowed by the symmetry are equal to zero. File *case.cfp* ($fort.8$) contains crystal field parameters in cm^{-1} written in the format that is used as the input file for 'REcfp' (or 'lanthanide').

Important:

- CFP are referred to the coordinate system in which the WIEN2k calculations (steps 1, 2) were performed.
- CFP with odd k are not calculated by *bkq*. The reason is that matrix elements of corresponding $\hat{C}_q^{(k)}$ in the f subspace are zero.

V. FIFTH STEP: USING REcfp PACKAGE

Program for this step is in:
RECFP/5th_step/program/SRC_REcfp)

Program 'REcfp' (modified version of 'lanthanide'[1]) calculates eigenenergies and eigenvectors of atom (ion) with f^n (also p^n and d^n, treated without three body corrections) electron configuration. The hamiltonian includes:

- free ion terms (e-e repulsion, spin-orbit coupling).
- Zeeman interaction
- Crystal field interaction

On input it requires

- 19 parameters of free ion hamiltonian.
- Magnitude and direction of external magnetic field.
- Crystal field parameters.

Original version of 'lanthanide' may be obtained from the CPC library. 'lanthanide' is written in *C*. At present 'lanthanide' is used only when eigenvectors in $|L, S, J, M_J\rangle$ are needed, as 'REcfp' does not provide this possibility yet.

'REcfp' compared to 'lanthanide' makes the calculation more flexible: calculation may be repeated with changed one-particle terms (Zeeman or crystal field) without the necessity to repeat the CPU demanding calculation of two and three body terms. Also the outputs are more user friendly. *REcfp* source code is a hybrid of *C* and fort90.

In most cases we used the parameters of free ion hamiltonian as determined by Carnall et al.[2]. For Nd and Er we checked that if other sets of parameters are used, the effect on the multiplet splitting and magnetism is only marginal. These parameters in format used as input for 'REcfp' (or 'lanthanide') are in
RECFP/5th_step/RE_param.

REcfp execution

```
cp case.cfp fort.8
cp R.inp fort.7
REcfp<case.incfp>case.outcfp
cp fort.66 case.energy
```

File R.inp (R=Ce ...... Yb) are in RECFP/5th_step/RE_param. Explanation of R.inp

```
electrons l RANGE STORE    number of electrons  L  0 0
R2 R4 R6 S2 S4 S6          put always  1. 1. 1. 0. 0. 0.
F2 F4 F6 xi                Slater integrals, s-o coupling constant [cm-1]
Alpha Beta Gamma           Trees parameters  [cm-1]
T2,T3,T4,T6,T7,T8          Casimir parameters [cm-1]
M0 M2 M4 P2 P4 P6          Judd parameters [cm-1]
Bx By Bz                   external field [T]
Bex_x Bex_y Bex_z          exchange field [T] treated as external field, but
                           acts only on the spin
```

For meaning of Trees, Casimir and Judd parameters see[1,2]
Input to *REcfp* case.incfp

line 1	nel,l,nstates,nv
line 2	ipr,iprd,iprm,iprb,iprcfp
line 3	chpar,nchange
	if chpar=1
line 4	ncomp,(comp(i),db(i),i=1,ncomp)
	if chpar=2 nchange lines each with $\vec{B}_{ext}$ is read from fort.4

1st line
nel is number of electrons
l is orbital quantum number
nstates: if 0, all eigenvalues are calculated, else only *nstates* eigenvalues are calculated
nv: if = 1 only energies are printed on fort.66, if *nv* = 2 also eigenvectors are printed
2nd line
ipr, *iprd*, *iprm*, *iprb*, *iprcfp* printing options in different parts of program. The bigger ipr, the longer output.
3rd line
chpar if $\neq 0$ either magnetic field B_{ext} (*chpar*=1 or 2) or CFP (*chpar*=3) are changed
nchange is number of changes
4th line for *chpar*=1

ncomp number of components of B_{ext} which will be changed
comp(i)=1,2,3 index of component (x,y,z)
db(i) increment of i-th component (starting $B_{ext}(i)$ values are on 7th line of fort.7.)
if *chpar*=2, there is no 4th line, but instead *nchange* lines are read from fort.4, each line with three components $B_{ext,x}, B_{ext,y}, B_{ext,z}$.

For *nv*=1 case.energy consists of *nchange* x *nstates* lines
$i \quad \varepsilon_i\ [cm^{-1}] \quad \varepsilon_i\ [meV] \quad \varepsilon_i - \varepsilon_1\ [meV]$
i is the index, ε_i the energy of the eigenstate. i=1 corresponds to the ground state.

There are five subdirectories $a, b, c, 110, ab\_plane$ in RECFP/5th_step/ndcoo3. In a, b, c and 110 subdirectories calculation of ground multiplet energies in the field $0 \leqq B_{ext} \leqq 1.25$ T (50 steps with increment 0.025 T) and $\vec{B}_{ext}$ along the orthorhombic axes and [110] are given. $\vec{B}_{ext} \parallel [110]$ is needed for calculation of canonical form of tensors in orthorhombic perovskites. The *ab_plane* contains angular dependence in the *ab* plane ($0 \leqq \phi \leqq 182^o$) for B_{ext}= 1 T and the step 1^o. In the latter case *ab_phi_1_T* should be copied on fort.4.

Important note
At the moment *REcfp* program has a limitation for f^n configurations with $4 \leqq n \leqq 11$ as maximum number *nchange* of changes is six. The user must therefore:

- be satisfied with 6 steps or
- run REcfp repeatedly. If this is the case, when executing REcfp do not copy *case*.inp to fort.7, after each REcfp run cp fort.66 to *case.energy_1*, *case.energy_2* etc.
- cp *case.energy_1* fort.21, *case.energy_2* fort.22 etc. and run auxiliary program *ehelp*:
  RECFP/6th_step/programs/ehelp<case.inehelp>case.energy
  case.inehelp consists of single line:
  nfile,nstates,nb
  where *nfile* is number of *REcfp* runs = number of files *case.energy_i*

VI. 6TH STEP: MAGNETIC PROPERTIES

Programs are in RECFP/6th_step/programs

Program *REcfp* yields dependence of the energy levels on external magnetic field. For B_{ext}=0 the splitting of multiplets by the crystal field is obtained. $B_{ext} \neq 0$ then yields dependence of these levels on the magnetic field. From this dependence the $\hat{g}$ tensor and van Vleck susceptibility $\hat{\chi}_{vV}$ tensors may be calculated, including their canonical form. There are several programs that facilitate this task.

Program *bhelp* rewrites the *case.energy* to a form more suitable for further processing, as plotting $\varepsilon_i(B_{ext})$ dependence or determination of $\hat{g}, \hat{\chi}$ tensors. Parameter *nv* in 5th step must be equal to 1 in order that *case.energy* contains only energies, not eigenvectors.

To execute the program:

```
cp case.energy fort.7
bhelp<case.inbhelp>case.outbhelp
cp fort.8 case.E_B
```

case.inbhelp consists of single line:
$n1, nb, nstates, nplot, b0, db, ipr$
$n1$: allows to leave out 1 ... $n1$ levels
nb: number of B_{ext}
nstates number of states calculated for each B_{ext}
nplot plotting option - see below
b0 starting value of B_{ext}
db increment of B_{ext}
ipr printing option.

If $nplot = 0$, file $fort.8$ is created. This file consists of $nb * nstates$ lines, each line containing
B_{ext} [T] $\quad \varepsilon_i$ [meV] index(B_{ext}) $\quad i$ (index of state)
if *nplot*=1, there are also $nfile = nstates/12 + 1$ output files written on fort.41 .... fort.40+nfile. Each of these files contains nb lines and each line consists from up to 13 numbers:
$B_{ext} \quad \varepsilon_i(B_{ext}) \quad i = nstart, nend$

$fort.41$ starts with $i = 1$ and ends with $i = \min(12, nstates)$
if $nstates > 12$ then $fort.42$ starts with $i = 13$ and ends with $i = \min(24, nstates)$, etc.
These files can be copied to *case.plot*1, *case.plot*2 etc.

Program *regr* calculates from $\varepsilon_i(B_{ext})$ dependence the components of $\hat{g}$ and $\hat{\chi}_{vV}$ using the quadratic regresion. The present version calculates $\hat{g}$ and $\hat{\chi}_{vV}$ for up to four directions of $\vec{B}_{ext}$. Taking these directions along three orthorhombic axes and [110] allows to determine in the orthoperovskites all independent tensor components[4,5].
To execute the program:

```
cp a/case.E_B fort.27
cp b/case.E_B fort.28
cp c/case.E_B fort.29
cp 110/case.E_B fort.30
CFP/6th_step/programs/regr<case.inregr>ndcoo3.outregr
mv fort.79 case.E0_g_chi_canonical
mv fort.98 case.g_chi
mv fort.9 case.accuracy
rm fort*
```

Input file *case.inregr* consists of single line
$nbe, nstate, kramers, ncomp, nab, ipr$
nbe number of magnetic fields
$nstate$ number of eigenstates
$kramers = 1$ for Kramers ions, $kramers = 0$ for non Kramers ions
$ncomp$ number of components ($ncomp$=4 if a, b, c, and 110 $\hat{g}$ and $\hat{\chi}_{vV}$ components of are calculated.)
nab: if $nab = 1$ the canonical form of $\hat{g}$ and $\hat{\chi}_{vV}$ in the c plane is calculated.

File *case.E0_g_chi_canonical* consists (for Kramers ions) from
$nstate/2$ lines with
Kramers doublet index, $\quad E(B_{ext} = 0)$ [meV], principal $\hat{g}$ components, principal $\hat{\chi}_{vV}$ components $[\mu_B/T]$

For non Kramers ions there are $nstate$ lines, each line corresponding to one of the singlet state.

File *case.E0_g_chi* for each Kramers doublet (non Kramers singlet) consists of index of state, four values g_a, g_b, g_c, g_{110} and in the next line four values of corresponding values of χ_{vV}.

File *case.accuracy* may be used to check the accuracy of quadratic regression. It consists of $nstate * nbe$ lines
$B_{ext}\ \ \varepsilon_i(B_{ext})$ of REcfp $\ \ \varepsilon_i(B_{ext})$ of regression, $\sum_1^i$ of squares of differences.

Program *temp* calculates temperature dependence of the magnetic moment and susceptibility of R ion. On input it needs the file *case.E0_g_chi_canonical* created by program *regr*. Using he Boltzman statistics it determines from energies and magnetic moments of eigenstates the temperature dependence of R magnetism.

To execute the program:

```
cp case.E0_g_chi_canonical fort.7
temp < case.intemp >case.outtemp
```

File case.intemp consists of three lines:
kramers,nstate
nb,b0,db,dbb
nt,t0,dt

Explanation of file case.intemp
1st line
kramers = 1 for Kramers ions, = 0 for non Kramers ions
nstate is number of eigenstates. It must be the same as in file case.inregr
2nd line
nb is number of B_{ext} values for which the temperature dependence is calculated
b0 is starting value of B_{ext} [T]
db is increment of B_{ext} [T]
dbb is step for calculating derivative dm/dB [T] (typically dbb = 0.1 T)
3rd line nt is number of temperatures
t0 is starting value of temperature
dt is increment of temperature

Case.outtemp contains the input data, followed by nt blocks, each block having nb lines
$T \ \ m_x \ \ m_y \ \ m_z \ \ \chi_x \ \ \chi_x \ \ \chi_x \ \ B_{ext}$,
where m_α, χ_α are magnetic moments in μ_B and susceptibility along principal directions, B_{ext} [T] is external magnetic field.

For convenience
$T \ \ m_x \ \ m_y \ \ m_z \ \ \chi_x \ \ \chi_x \ \ \chi_x \ \ B_{ext}$,
are also written separately for each B_{ext} value on files fort.40+ib, ib=1, .. nb.

Program *tempav* allows to calculate temperature dependence of the magnetic moment and susceptibility of R ion in a polycrystal. It is similar to program *temp*, in addition it performs an averaging of the magnetic moment and susceptibility over angles ϕ ($0 \leqq \phi \leqq \pi$) and θ ($0 \leqq \theta \leqq \pi/2$). Averaging is performed by two dimensional integration over ϕ and $x = \cos\theta$.

To execute the program:

```
cp case.E0_g_chi_canonical fort.7
tempav < case.intempav >case.outtempav
```

File case.intempav consists of four lines:
kramers,nstate
nphi,nx
nb,b0,db,dbb
nt,t0,dt
nphi and nx are numbers of integration points for variables ϕ and x, meaning of remaining three lines is the same as in file case.intemp.

The output files are again analogs of program temp outputs, in particular files fort.40+ib (ib=1...nb) contain nt lines
$T \ \ m_{aver} \ \ \chi_{aver} \ \ B_{ext}$.

VII. OVERVIEW OF STEPS

To understand this manual our papers[3–5] and the userguides to WIEN2k, wien2wannier and wannier90 should be consulted. Below we list the sequence of commands described above.

1. init_lapw
2. run_lapw (-cc 0.00001 -p): f in core
3. x lapw1 -orb (-p): f in valence states
4. wien2wannier: interface WIEN2k and wannier90
5. wannier90: Bloch to Wannier transformation
6. bkq: local hamiltonian determination followed by calculation of CFP
7. REcfp: calculation of multiplets splitting
8. magnetism. bhelp: $\varepsilon_i(B_{ext})$ dependence, regr: determination of $\hat{g}$ and susceptibility tensors, temp and tempav: temperature dependence of magnetism

VIII. MISCELANOUS

If a series of compounds, like R:YAP, R:YAG etc is considered, the calculation following 1st step can be made easier, as *case.in*1, *case.vorb*, *case.klist*, *case.w*2*win* of steps 2 and 3 are the same for all compounds. we thus suggest to create corresponding script.

RECFP contains also subdirectory *auxiliary* containing auxiliary scripts and in subdirectory *programs* also programs. At the moment there are two programs *compare* and *cfp_Delta*. *compare* helps to compare experimental and calculated multiplet splittings, which is nontrivial in case some levels were not detected in the experiment. *cfp_Delta* is a simple program to obtain $CFP(\Delta)$ dependence.

[1] S. Edwardsson and D. Aberg, CPC **133**, 396 (2001).

[2] W.T. Carnall, G.L. Goodman, K. Rajnak, and R.S. Rana, J. Chem. Phys. **90**, 3443 (1989).
[3] P. Novák, K. Knížek, and J. Kuneš, Phys. Rev. B **87**, 205139 (2013).
[4] P. Novák, K. Knížek, M. Maryško, Z. Jirák and J. Kuneš, J. Phys.: Condens. Matter **25**, 446001 (2013).
[5] P. Novák, V. Nekvasil, and K. Knížek, J. Mag. Mag. Mater. **358-359**, 228 (2014).