1 Runwien user's and reference guide

Authors: Alberto Otero-de-la-Roza and Víctor Luaña

Contact: alberto@carbono.quimica.uniovi.es

Address: Departamento de Química Física y Analítica, Universidad de Oviedo, Julián Clavería 8, 33007 Oviedo, Spain

Version: 1.0.11

Date: 2008-10-09

Contents

2 User's Guide

2.1 Introduction

Developed by K. Schwartz, P. Blaha and their coworkers, the WIEN2K [WIEN2k] electronic structure package performs Full Potential Linear Augmented Plane Wave (FP-LAPW) calculations on solids. The package is made of a large number of Fortran and C codes that exchange information through a set of ASCII and binary data files. A collection of shell scripts is used to drive the main tasks. The user is expected to work through a web interface that greatly simplifies the trouble of creating and formatting the several input files needed for any calculation. However, we have felt the need of a tool capable of processing large series of calculations, such as a potential energy surface exploration. Therefore, we have devised a new approach, better suited for large scale production runs: runwien.

Runwien is a new, script-driven, interface for the WIEN2K package. The user is expected to write a single file describing the system to be calculated and the tasks to perform and runwien is in charge of the ordered execution of the many programs integrated in the WIEN2K bundle. The input file can be quite short and simple because runwien knows default values for many of the internal parameters governing the electronic structure calculation. On the other hand, direct control of every relevant parameter is also available. Exploring the convergence of the results with respect to the internal parameters is facilitated because a grid instead of a fixed value can be used for one or more of the parameters. Similarly, a potential energy surface can be explored by using a grid on one or more of the independent variables that control the system geometry. The execution concludes with the production of a flexible summary of the most useful results.

This user manual is divided into two main parts. The first part introduces runwien through the discussion of a selected set of examples. The second part consists of a formal description of the orders implemented in runwien and some of the main mechanisms used in programming the interpreter.

2.2 Obtaining and installing runwien

See README.

2.3 First steps: Calculations on hcp Be

Our first example corresponds to a calculation of hcp Be at the experimental NPT (normal pressure and temperature) geometry. We use a large GMAX=20.0 value, as recommended for light elements, and fix an appropriate muffin tin radius (RMT). Let us take a look to the "be.wien" input:

```
#runwien#
# Comment lines must start with a # as the first non-blank character.
# Blank lines, spaces, tabs and upper/lower caps are ignored

general
  # Crystal description: H corresponds to a hexagonal lattice.
  # Check the lattice types in the WIEN2K user guide.
  lattice H
  equiv list Be
```

[WIEN2k] P. Blaha, K. Schwarz, G.K.H.. Madsen, D. Kvasnicka, and J. Luitz, "WIEN2k, An Augmented Plane Wave + Local Orbitals Program for Calculating Crystal Properties", (Karlheinz Schwarz, Techn. Universität Wien, Austria), 2001, ISBN 3-9501031-1-2. http://www.wien2k.at/

```
    0.66666667 0.33333333 0.75000000
    0.33333333 0.66666667 0.25000000
  end equiv list
  relativistic yes
  cell parameters 4.3210478 4.3210478 6.7714557 90 90 120
  rmt 1        1.8
  gmax         20.0
end general

# Use default values for the initialization and prescf sections
initialization default
prescf default

scf
  # Increase the default number of scf iterations and ask for
  # a convergence of 0.00001 Ry in the energy.
  max iterations 50
  energy conv 0.00001
end scf

synopsis
 exhaustive
end synopsis
```

This input is divided into sections, each one corresponding to a main part in a WIEN2K calculation. There are at least four sections, required for all meaningful jobs, that must be entered into a precise order:

- The **general** section is used to enter a reference geometry for the crystal (lattice lengths and angles, space group symmetry, atoms in the main cell, etc) and give values to the main parameters for the FP-LAPW method (muffin tin spheres, number of sample points in the reciprocal cell, etc). The user can ask here for an exploration of one or more of the model parameters but the crystal geometry is held fixed.
- The **initialization** section is used to decide low level but important details of the FP-LAPW model. The exchange and correlation functional, core-valence separation energy, local orbital special treatments, etc. are defined in this section. The default values are correct in many cases.
- The **prescf** section is in charge of preparing the sampling of k-waves and the initial electron density. The default values are usually enough for the corresponding parameters.
- The **scf** section controls the self-consistent iteration process. In our example we have decided to increase the maximum iteration cycles to 50 instead of the default 30. The convergence in the energy up to 10 microRydberg is actually the default.

The remaining sections in the input are all optional:

Printrho: Tasks related to the electron density and several interaction potentials.

Dosplot: Calculation and plot of the Density of States (DOS) and similar tasks.

Rxplot: Print the calculated x-ray emission or absorption spectra.

Bandplot: Plot the band structure.

Kdos: Combined plot of the band structure and the density of states.

Aim: Perform an Atoms in Molecules, i.e. Bader, analysis of the crystal electron density using Sofo et al. code included in WIEN2k.

Critic: Perform an Atoms in Molecules, i.e. Bader, analysis of the crystal electron density using Martín Pendás et al. critic code [POL2008].

Sweep: Produces a Potential Energy Surface by means of the variation of one or more geometrical parameter: cell lengths and angles, internal atomic positions, or some combinations of these.

Gibbs: Use the energy versus volume data to determine the Equation of State (EOS) and produce an estimation of thermal effects using the quasiharmonic Debye model implemented in Blanco et al. gibbs code [BFL2004].

Elastic: Drive the calculation of the elastic constants of the crystal.

Free: Used to determine the energy of free atoms under the same calculation conditions used for a crystal.

Synopsis: Flexible tool for the creation of reports on the calculation results.

Following WIEN2k usual practices, the runwien input file should reside in its own directory, and the directory name provides the root name for the large abundance of files created during the calculation. In other words, let us assume the calculation will reside in the directory:

```
#ascii#
~alberto/calc/wien/tests/be/
```

Then, the runwien input should be called “be.wien”. The calculation starts by invoking the runwien interpreter:

```
#ascii#
cd ~alberto/calc/wien/tests/be/
runwien.awk be.wien
```

Runwien will start writing status information to stdout (you can redirect it to a file). After a while the calculation will end and the file ’synopsis.out’ will contain a summary of the run. In the case of our example:

```
#ascii#
General information
---------------------------------------------------------------------
title   : runwien.awk lapw calculation, v.1.0.5
root    : be
pwd     : /home/alberto/calc/wien/tests/be
machine : xxxx
date    : jue nov 29 17:34:35 CET 2007

Sections run and time information
---------------------------------------------------------------------
general section run, time (s) = 0
initialization section run, time (s) = 1
prescf section run, time (s) = 0
scf section run, time (s) = 23
total run time (s) = 24

Structural information
---------------------------------------------------------------------
```

[POL2008] A. Otero-de-la-Roza, M. A. Blanco, A. Martín Pendás and Víctor Luaña, “Critic: a new program for the topological analysis of solid-states electron densities”, (to be published).

[BFL2004] M. A. Blanco, E. Francisco, and V. Luaña, “GIBBS: isothermal-isobaric thermodynamics of solids from energy curves using a quasi-harmonic Debye model”, Comput. Phys. Commun. **158** (2004) 57--72.

```
Mol. formula : Be1
Mol. mass    : 9.012182
a / bohr     : 4.32104800
b / bohr     : 4.32104800
c / bohr     : 6.77145600
alpha / deg  : 90.00000000
beta / deg   : 90.00000000
gamma / deg  : 120.00000000
c/a          : 1.56708650
c/b          : 1.56708650
b/a          : 1.00000000
volume/bohr^3: 109.49413924
Mol. volume  : 54.74706962
space group  : P63/mmc
lattice      : H
system       : hexagonal
nonequivalent atom list in .struct (and position of a representative
   atom):
  atom 1 is Be mult=2 x=0.66666667 y=0.33333333 z=0.75000000

Additional information
----------------------------------------------------------------------
Number of electrons in unit cell : 8
Valence electrons in unit cell   : 4
Core electrons in unit cell      : 4
Core | val | total e- for atom   : [Be] 2 | 2 | 4
Core | val el. conf. for atom    : [Be] 1S(2) | 2S(2)
Core leaking (e-) for atom       : [Be] 1.08300e-03
Atomic energy (lstart) [Be]      : -29.2657470
Total in vacuo energy (lstart)   : -29.2657470

Calculation fixed parameters
----------------------------------------------------------------------
spinpolarized               : no
relativistic first density  : yes
general, npt                : 781
general, rmt (atom 1)       : 1.80000
general, r0                 : 0.00010000
general, rkmax              : 7.00000
general, lmax               : 10
general, lnsmax             : 4
general, gmax               : 20.000
general, mix                : 0.40000
general, kpts               : 1000
initialization, potential   : ggapbe96
initialization, ecoreval /ry: -6.000
initialization, emin / ry   : -9.00000000
initialization, emax / ry   : 2.00000000
scf, commands used          :  -i 50 -ec 0.00001

Only one structure, no variable parameters
Energy*/ry  = -29.5397615000
E fermi /ry = 0.50969
```

```
No critic data available
...(Rest of summary skipped)...
```

As this output clearly reveals, there is a large number of internal parameters that runwien knows about and that the user can keep as their defaults of can control at wish. Occasionally the user may want to explore the influence of some of these parameters on the results or just optimize the best calculation conditions. This is facilitated because runwien provides the possibility of entering a list and a range of values for some of the parameters. For instance:

```
#runwien#
rmt 1 1.70,1.75,1.80
```

will perform three different calculations using the enlisted three different values for the muffin tin radius of Be (non-equivalent atom number 1). The same purpose could be achieved using a range:

```
#runwien#
rmt 1 1.70/1.80/0.05
```

where the three values of the range are the starting and ending values and the step (1 by default), respectively. A list of ranges and values is also possible:

```
#runwien#
rmt 1 1.7,1.800/1.900/0.025
```

The actual value of the parameter can even be given as a increase or decrease of the value given by WIEN2k:

```
#runwien#
rmt 1 -10%,+15%
```

where -10% means ten percent *less* than the default or recommended value, and +15% fifteen percent more. The internal parameters that can receive this treatment of using a list or range (percent values only work for rmt) include:

gmax: Maximum reciprocal vector for the fourier expansion of the electron density.

kpts: Number of k-points in the whole first brillouin zone.

lmax: Maximum angular moment for the spherical harmonics expansion in the basis functions.

lnsmax: Maximum angular moment for the spherical harmonics used to calculate matrix elements with non muffin tin functions.

mix: Mixing factor of the previous densities used on the self-consistent calculation.

npt: Number of radial mesh points. It must be an odd number. 381 is the recommended value for LDA and 781 for GGA calculations.

r0: First radial mesh point: a logarithmic scale is used, that starts in 0.0001 bohr by default.

rkmax: Product rmt times kmax, where kmax is the maximum k used for the plane waves set.

rmt: Muffin tin radius in bohr. Each non-equivalent atom in the crystal can have a different value, although difficult convergence problems can arise if the crystal includes atoms with very different rmt's.

The above list represent some of the most influential parameters on the final LAPW results. The exploration of the convergence of the total energy with respect to rkmax, for instance, is a quite common task when we start working on a new crystal. Similarly, the rmt radius needs to be adapted when high pressure phases are involved or we are studying the topological properties of the electron density.

The exploration can involve two or more parameters at once. For instance:

```
#runwien#
rmt 1 1.8/2.0/0.5
kpts 2000,5000,10000
rkmax 7.0/9.0/0.5
```

would ask for 75 different calculations in which five different values of rmt are combined with three values for kpts and with five values for rkmax. This multidimensional search can easily give rise to a large number of calculations. Runwien provides two different mechanisms to reduce this dimensional escalade. The “also” keyword can be used to perform two or more independent explorations of parameters. For instance:

```
#runwien#
   rmt 1 1.8/2.0/0.5
   kpts 5000
   rkmax 8.0
also
   rmt 1 1.9
   kpts 2000,5000,10000
   rkmax 8.0
also
   rmt 1 1.9
   kpts 5000
   rkmax 7.0/9.0/0.5
```

will produce three independent one dimensional scans. On the other hand, two or more parameters can be linked to vary in a correlated way:

```
#runwien#
link rmt rkmax
rmt 1 1.8/2.0/0.5
rkmax 7.0/9.0/0.5
```

In this example, there will be just five different calculations where both variables will change at the same time. Of course, the linked variables should have the same number of values.

2.4 Potential energy surface (PES) calculations

A very important feature of runwien is its ability to calculate a large amount of points in a potential energy surface with a minimal effort. The section in charge of this task is called *sweep* and requires that the general, initialization, prescf and scf sections be executed first. The sweep section works by creating a subdirectory for each of the required geometries and preparing a “.wien” file on each directory. The runwien script will be applied afterwards to all those new input files unless the user asks otherwise. All the structures created by the sweep section share the same WIEN2k parameters (which can include a list or range of values for some of them) but vary in their cell constant values. We will refer to this mechanism as **sweep structures** from now on.

Let us see an example. From the calculations in the previous section we have determined the best WIEN2k parameters to use on the hcp Be crystal. Some more tricks that our next input uses are a precise set of linearization energies and the best set of basis functions for the whole PES:

```
#runwien#
general
   lattice H
   equiv list Be
      0.66666667 0.33333333 0.75000000
      0.33333333 0.66666667 0.25000000
   end equiv list
   relativistic yes
   cell parameters 4.3210478 4.3210478 6.7714557 90 90 120
   # Reduce rmt to avoid muffin collision on compression
   rmt 1 -10%
   gmax   20.0
   rkmax  7.5
   kpts   7500
end general
initialization
   ecoreval -10.0
   orbitals 1   .24243  0
     0    0.054     0.002 CONT 0
     0   -6.813     0.002 CONT 0
     1    0.261     0.002 CONT 0
   end orbitals
   # LM list for spherical harmonics expansion of charge density
   # inside the muffin tin sphere. You probably will have to
   # recompile WIEN2K with some augmented dimension parameter.
   # If this is your case, you can simply remove this line.
   lm list 1
      0 0  2 0  3 3  4 0  5 3  6 0  6 6
      7 3  8 0  8 6  9 3  9 9  10 0 10 6
   end lm list
end initialization
prescf default
scf
   do none
   # Another runwien feature: you can select which structures are
   # going to be calculated. The "do none" order forbids the
   # calculation of the structures defined in the general section
   # (a single one in this case). The structures defined in the
   # next sweep part will be the do ones being calculated.
   max iterations 50
   energy conv 0.0001
end scf
sweep
   with v 80,85,90,95,100,105,110
   # keep c/a = sqrt(8/3) as in an *ideal* hcp structure
   with c/a 1.632993
   print energy vs v
end sweep
synopsis default
```

The **with** instruction in the sweep section indicates the geometrical variables that are going to be explored. The cell lengths and angles can be explored in this way but, as the example shows, the user can ask for a variation of the cell volume, the ratio between two cell lengths, the internal position of an atom, etc. On the other hand, runwien avoids those changes that are not compatible with the crystal symmetry. For instance, an instruction like:

```
#runwien#
with b 4.0/5.0/0.2
```

will be simply ignored for an hexagonal crystal.

The sweep exploration can involve changes in two or more independent variables to produce a multidimensional energy surface:

```
#runwien#
with v 80/100/5
with c/a 1.60/1.80/0.05
```

Two or more independent scans can be produced in sequence:

```
#runwien#
with a 4.20/4.40/0.05
with c 6.70/6.80/0.05
also
with a 4.10/4.50/0.05
with c 6.75
also
with a 4.30
with c 6.60/6.90/0.05
```

A significant limitation of the current sweeping mechanism is that the crystallographic coordinates of the atoms within the cell cannot be changed. The sweeping mechanism is thus limited to the cell parameters.

2.5 Continuing and completing a calculation

There are many occasions, specially when exploring large PES, when you will not do all the calculations in a row. For instance, your initial set of geometries does not bracket the energy minimum appropriately and you want to add points to the grid, or some kind of problem has stopped your run in the middle of a calculation, or simply, you want to distribute your calculations on a collection of machines and, afterwards, put back together for the final summary. Whatever the circumstance, runwien provides a method for stopping and resuming calculations by using checkpoint files. These files are automatically compiled during the run and form a kind of “black box” of the calculation. The LOADCHECK order allows the user to load all this information back into runwien. This CHECK/LOADCHECK mechanism can also be useful if runwien run is interrupted. If that happens, you can manually clean the point which was being calculated and restart from that point.

Let us examine an example of this recovery mechanism using, again, the hcp Be PES as a test case. The calculation will be divided into three main parts: first, create the sweep tree structure and input files; then run the runwien jobs under the sweep subdirectory manually; and then load back the results into a general runwien. The .wien file for the first step would be:

```
#runwien#
general
  lattice H
  equiv list Be
    0.66666667 0.33333333 0.75000000
    0.33333333 0.66666667 0.25000000
  end equiv list
  relativistic yes
  cell parameters 4.3210478 4.3210478 6.7714557 90 90 120
  rmt 1       -10%
  # This time, rmt is set to 10% less than the recommended value.
  # Otherwise, muffin tin spheres would collide on compression.
  gmax         20.0
  rkmax        7.5
  kpts         7500
end general
initialization
  ecoreval -10.0
  orbitals 1   .24243  0
    0    0.054     0.002 CONT 0
    0   -6.813     0.002 CONT 0
    1    0.261     0.002 CONT 0
  end orbitals
  lm list 1
    0 0  2 0  3 3  4 0  5 3  6 0  6 6
    7 3  8 0  8 6  9 3  9 9  10 0 10 6
  end lm list
end initialization
prescf default
scf
  do none
  max iterations 50
  energy conv 0.0001
end scf
sweep
  do none
  with v 50,60,70,80,90,100,110,120
  with c/a 1.632993
end sweep
```

Let us assume that this file is “be-pes.wien” and it is located in a directory called “be-pes”. The calculation is performed by issuing the order:

```
#ascii#
runwien.awk be-pes.wien
```

The calculation ends in a few seconds. According to our input, runwien creates a collection of subdirectories and input files, but the “DO NONE” orders in the SCF and SWEEP sections prevents any lengthy calculation to actually take place. The subdirectories created are:

```
#ascii#
./be-pes-check
./be-pes1
./be-pes-sweep
./be-pes-sweep/be-pes-sweep1
./be-pes-sweep/be-pes-sweep2
./be-pes-sweep/be-pes-sweep3
./be-pes-sweep/be-pes-sweep4
./be-pes-sweep/be-pes-sweep5
./be-pes-sweep/be-pes-sweep6
./be-pes-sweep/be-pes-sweep7
./be-pes-sweep/be-pes-sweep8
```

The <be-pes-check> directory will contain the “.check” status reports, essential to the stop-and-continue mechanism. The <be-pes1> directory contains the input files to perform the single WIEN2k calculation required by the GENERAL section. Finally, the eight subdirectories in <be-pes-sweep> correspond to the eight cell volumes asked for in the SWEEP section. Each subdirectory contains a “.wien” file that can be run by hand to produce the calculation of the corresponding point. On a Beowulf cluster, for instance, mpiexec or a similar job launcher can be used to run each sweep point on a different core, thus achieving a parallel execution. When all the 8 points have been calculated, the information is taken back into runwien with a new <be-pes.wien>:

```
#runwien#
loadcheck reread
synopsis default
```

LOADCHECK reads all the information stored in the checkfiles (structure, etc.) and reread instructs runwien to check if actual information is different from checkfile's information. This is the case, for we want the energies and properties of sweep points loaded. Having read all this information, runwien pretty-prints a synopsis containing all the data you have manually calculated. The <synopsis.out> of the run is:

```
#ascii#
General information -----------------------------------------------
title   : runwien.awk lapw calculation, v.1.0.5
root    : temp
pwd     : /home/alberto/temp
machine : xxxx
date    : mie oct 24 00:07:22 CEST 2007

Sections run and time information ----------------------------------
general section run, time (s) = 0
initialization section run, time (s) = 1
prescf section run, time (s) = 0
scf section run, time (s) = 0
sweep section run, time (s) = 1507
total run time (s) = 1508

Structural information ---------------------------------------------
Mol. formula : Be1
Mol. mass    : 9.012182
a / bohr     : 4.32104800
b / bohr     : 4.32104800
c / bohr     : 6.77145600
alpha / deg  : 90.00000000
beta / deg   : 90.00000000
gamma / deg  : 120.00000000
c/a          : 1.56708650
c/b          : 1.56708650
b/a          : 1.00000000
volume/bohr^3: 109.49413924
Mol. volume  : 54.74706962
space group  : P63/mmc
lattice      : H
system       : hexagonal
nonequivalent atom list in .struct (and position of a representative
   atom):
  atom 1 is Be mult=2 x=0.66666667 y=0.33333333 z=0.75000000

Additional information ---------------------------------------------
Number of electrons in unit cell : 8
```

```
Valence electrons in unit cell   : 8
Core electrons in unit cell      : 0
Core | val | total e- for atom   : [Be]  | 4 | 4
Core | val el. conf. for atom    : [Be]  | 1S(2)2S(2)
Core leaking (e-) for atom       : [Be] 0.00000e+00
Atomic energy (lstart) [Be]      : -29.2657470
Total in vacuo energy (lstart)   : -29.2657470

Calculation fixed parameters -----------------------------------------
spinpolarized               : no
relativistic first density  : yes
general, npt                : 781
general, rmt (atom 1)       : 1.88100
general, r0                 : 0.00010000
general, rkmax              : 7.50000
general, lmax               : 10
general, lnsmax             : 4
general, gmax               : 20.000
general, mix                : 0.40000
general, kpts               : 7500
initialization, potential   : ggapbe96
initialization, ecoreval /ry: -10.000
initialization, emin / ry   : -13.00000000
initialization, emax / ry   : 2.00000000
scf, commands used          :  -i 50 -ec 0.0001

Only one structure, no variable parameters
No scf data available...
No critic data available

Structural grid (sweep) results -------------------------------------
Fixed parameters --
alpha / deg   : 90
beta / deg    : 90
gamma / deg   : 120
c / a         : 1.63299
c / b         : 1.63299
b / a         : 1

Variable parameters --
number of structures : 7
 a / bohr b / bohr c / bohr     v/bohr^3 it. warning    energy / ry    efermi / ry   time /s pl
 3.83877  3.83877  6.26868      40.0000   5      --    -29.5183990    0.724170000        153 0.0
 3.91713  3.91713  6.39665      42.5000   5      --    -29.5266380    0.680700000        217 0.0
 3.99248  3.99248  6.51969      45.0000   5      --    -29.5323950    0.641240000        225 0.0
 4.06508  4.06508  6.63825      47.5000   5      --    -29.5361680    0.605240000        230 0.0
 4.13519  4.13519  6.75273      50.0000   5      --    -29.5383420    0.572460000        227 0.0
 4.20299  4.20299  6.86345      52.5000   5      --    -29.5392095    0.542130000        226 0.0
 4.26867  4.26867  6.97071      55.0000   5      --    -29.5390180    0.514280000        228 0.0

Output, index, plot and gnuplot files ------------------------------
* .out files
 - scfsummary.out
 - temp-sweep/temp-sweep1/runwien.out
 - temp-sweep/temp-sweep2/runwien.out
 - temp-sweep/temp-sweep3/runwien.out
 - temp-sweep/temp-sweep4/runwien.out
 - temp-sweep/temp-sweep5/runwien.out
 - temp-sweep/temp-sweep6/runwien.out
 - temp-sweep/temp-sweep7/runwien.out
 - sweepsummary.out
 - temp-sweep/temp-sweep1/synopsis.out
 - temp-sweep/temp-sweep2/synopsis.out
 - temp-sweep/temp-sweep3/synopsis.out
 - temp-sweep/temp-sweep4/synopsis.out
 - temp-sweep/temp-sweep5/synopsis.out
 - temp-sweep/temp-sweep6/synopsis.out
 - temp-sweep/temp-sweep7/synopsis.out
 - synopsis.out
* .index files
 - temp.index
 - temp-sweep/temp-sweep.index
* .ps files
* .gnuplot files
```

2.6 Calculation of properties

The runwien programming paradigm is based on clearness and modularity. Once the SCF process has converged, the calculation of many crystal properties can be achieved by running a collection of independent property sections: *dosplot* for the Density of States, *bandplot* for the Band of Structure analysis, *critic* for the electron density topology, and so on. The modularity of runwien makes it easy to add or remove sections from the ".wien" input file.

We will show some of the post-scf possibilities that runwien offers using the popular WIEN2k user's guide example: TiC. A commented <TiC.wien> input file can be as follows:

```
#runwien#
general
  # title keyword sets the global title. Try to avoid
  # "exotic" characters here, like ' or \
  title TiC calculation, chapter 3 in wien guide

  # lattice is fcc
  lattice F

  cell parameters 8.178735 8.178735 8.178735 90 90 90
  # There are two non-equivalent atoms: Ti and C, occupying the
  # positions of a rock-salt structure
  equiv list Ti
    0 0 0
  end equiv list
  equiv list C
    0.5 0.5 0.5
  end equiv list

  # Explore a small change from the recommended rmt values.
  # The rmt of both atoms will be linked, giving rise to a 1D scan.
  link rmt1 rmt2
  rmt 1 -3%, -2%, -1%
  rmt 2 -3%, -2%, -1%

  gmax 15.0
  r0 0.00005
  kpts 1000
end general
initialization
  xcpotential ggapbe96
  energymax 2.0
end initialization
prescf default
scf
  # Use charge convergence criterion
  charge conv 0.0001
end scf
do 1
# Run the following sections (except sweep) only for structure
# number 1, i.e. for the one with -3% rmt. The *do* and *except*
# keywords provide a rich mechanism for this kind of selection.

printrho
  # Plot the electron valence density in a plane.
  rho valrho
  # Other options: total, deformation, spin and atomic densities;
  # coulomb, exchange-correlation and total potential.

  # Energy cutoff for the bands adding to the valence density
  energymin -1.0
  # Plane specification: corners and number of grid points
  origin -1 -1 0 4
  xend -1 3 0 4
  yend 3 -1 0 4
  npt 100 100
  # Number of unit cells added up on each direction
  nshells 3 2 3
  # Cap the electron density fixing a min and max value in (e/bohr^3)
  zmin -0.075
  zmax 0.3
end printrho

dosplot
  # Density of States plot (total and projected).
  # Use eV (Rydberg is the default) and fix plot limits
  plotunits ev
  plotxmin -12.5 ev
  plotxmax 3.0 ev
  # DOS list as set in the .qtl file (see WIEN2k manual).
  dos list
    0 1 tot
    1 4 Ti d
    1 5 Ti eg
    1 6 Ti t2g
    2 2 C s
    2 3 C p
  end dos list
end dosplot

rxplot
  # X-Ray emission or absorption spectra.
  # Extend the eigenvalue window to 2.5 Ry (use lapw1). As a general
  # rule, each time this value is set in lapw1 to calculate a
  # property, it is reset at the end of the section to its original
  # value.
  in1maxenergy 2.5
  # Calculating absorption spectra for Ti, n=2, l=1.
  atom 1
  n 2
  l 1
  # x-limits and dx for the plot.
  plotxmin -2.0
  de 0.02
  plotxmax 15.0
end rxplot

bandplot default
# Plot the band structure with all the default values.

critic
  # Topological analysis of the electron density.
  # Get the critical points using Newton's search and 3-level
  # recursive division of the irreducible wedge of the wigner-seitz
  # cell (IWS)
  newton 1e-15
  iws 3
end critic

sweep
  # Calculate 5 points: from -10% to +10% of the volume of the
  # reference structure which, in this case, is the first one
  # indicated in the general section.
  with v -10/10/5%
  print energy vs v
end sweep

synopsis
  exhaustive
end synopsis
```

2.7 A complex case: ferrocene

Let us examine now a complex case: ferrocene. The structure can be found in one of the WIEN2K distribution example .struct files. Runwien offers the possibility of loading a structure from a .struct file, as well as from a .cif file (through the use of the cif2struct utility). To do that, we create a directory for the calculation (.../ferrocene/) and copy the original struct file into /ferrocene/orig-ferrocen.struct. The ferrocene.wien file contains:

```
#runwien#
general
  loadstruct orig-ferrocen.struct
  # Load structural info from existing .struct
  kpts 1
  rkmax 3.0
  lmax 12
  lnsmax 5
  gmax 23.0
end general
initialization
  xcpotential ggapbe96
end initialization
prescf default
scf
  max iterations 100
  energy conv 0.00001
  itdiag 6
  # Require iterative diagonalization with a full diagonalization
  # each 6 scf cycles. This value has no effect in newer WIEN2k
  # versions.
end scf
synopsis default
```

For the sake of computational simplicity, only one k-point will be used. The synopsis.out generated by runwien is:

```
#ascii#
General information
-----------------------------------------------------------------------
title   : Ferrocen     (Molecule)                              <=
root    : temp
pwd     : /home/alberto/temp
machine : xxxx
date    : mie oct 24 02:22:37 CEST 2007

Sections run and time information
-----------------------------------------------------------------------
general section run, time (s) = 0
initialization section run, time (s) = 4
prescf section run, time (s) = 124
scf section run, time (s) = 3043
total run time (s) = 3171

Structural information
-----------------------------------------------------------------------
Mol. formula : Fe1C2C2C2C2C2H2H2H2H2H2
Mol. mass    : 186.031400
a / bohr     : 15.00000000
b / bohr     : 15.00000000
```

```
c / bohr     : 15.00000000
alpha / deg  : 90.00000000
beta / deg   : 90.00000000
gamma / deg  : 90.00000000
c/a          : 1.00000000
c/b          : 1.00000000
b/a          : 1.00000000
volume/bohr^3: 3375.00000000
Mol. volume  : 3375.00000000
space group  : P-1
lattice      : P
system       : cubic
structure loaded from .struct file : ferrocen.struct
nonequivalent atom list in .struct (and position of a representative
   atom):
  atom 1 is Fe mult=1 x=0.00000000 y=0.00000000 z=0.00000000
  atom 2 is C mult=2 x=0.08686660 y=0.12560000 z=0.20900000
  atom 3 is C mult=2 x=0.90739070 y=0.12142770 z=0.20900000
  atom 4 is C mult=2 x=0.85589760 y=0.94944650 z=0.20900000
  atom 5 is C mult=2 x=0.00354910 y=0.84732850 z=0.20900000
  atom 6 is C mult=2 x=0.14629600 y=0.95619730 z=0.20900000
  atom 7 is H mult=2 x=0.16444000 y=0.23976270 z=0.20566670
  atom 8 is H mult=2 x=0.82278690 y=0.23048250 z=0.20566670
  atom 9 is H mult=2 x=0.72603630 y=0.90268330 z=0.20566670
  atom 10 is H mult=2 x=0.00789420 y=0.70937250 z=0.20566670
  atom 11 is H mult=2 x=0.27884260 y=0.91769900 z=0.20566670

Additional information
---------------------------------------------------------------------
Number of electrons in unit cell : 96
Valence electrons in unit cell   : 64
Core electrons in unit cell      : 32
Core | val | total e- for atom   : [C] 2 | 4 | 6
Core | val el. conf. for atom    : [C] 1S(2) | 2S(2)2P(2)
Core leaking (e-) for atom       : [C] 4.11000e-04
Core | val | total e- for atom   : [Fe] 12 | 14 | 26
Core | val el. conf. for atom    : [Fe] 1S(2)2S(2)2P(6)3S(2) |
       3P(6)3D(6.5)4S(1.5)
Core leaking (e-) for atom       : [Fe] 3.47700e-03
Core | val | total e- for atom   : [H]  | 1 | 1
Core | val el. conf. for atom    : [H]  | 1S(1)
Core leaking (e-) for atom       : [H] 0.00000e+00
Atomic energy (lstart) [Fe]      : -2545.1312760
Atomic energy (lstart) [ C]      : -75.6207230
Atomic energy (lstart) [ H]      : -0.9665040
Total in vacuo energy (lstart)   : -3311.0035460

Calculation fixed parameters
---------------------------------------------------------------------
spinpolarized               : no
relativistic first density  : yes
general, npt                : 381
general, rmt (atom 1)       : 2.10000
```

```
general, rmt (atom 2)       : 1.25000
general, rmt (atom 3)       : 1.25000
general, rmt (atom 4)       : 1.25000
general, rmt (atom 5)       : 1.25000
general, rmt (atom 6)       : 1.25000
general, rmt (atom 7)       : 0.82000
general, rmt (atom 8)       : 0.82000
general, rmt (atom 9)       : 0.82000
general, rmt (atom 10)       : 0.82000
general, rmt (atom 11)       : 0.82000
general, r0                 : 0.00050000
general, rkmax              : 3.00000
general, lmax               : 12
general, lnsmax             : 5
general, gmax               : 23.000
general, mix                : 0.40000
general, kpts               : 1
initialization, potential   : ggapbe96
initialization, ecoreval /ry: -6.000
initialization, emin / ry   : -9.00000000
initialization, emax / ry   : 2.00000000
scf, commands used          :  -i 100 -ec 0.00001 -it 6

Only one structure, no variable parameters
Energy*/ry  = -3318.8725220000
E fermi /ry = -0.13297
No critic data available

Output, index, plot and gnuplot files
----------------------------------------------------------------------
* .out files
 - scfsummary.out
 - synopsis.out
* .index files
 - temp.index
* .ps files
* .gnuplot files
```

As can be seen, a very complicated structure can be calculated easily by using runwien adequately. Next steps towards ferrocene electronic structure could include examining kpts and rkmax effect on energy and properties, optimization of rmt, etc. All this steps can be synthesized in just a few lines of input, and all the output generated by them is extracted, sorted and presented in a comfortable way.

2.8 Full WIEN2k calculation in sodium chloride

Next, a calculation of the sodium chloride crystal is presented in full detail. The geometry of the unit cell of NaCl can be retrieved from a public crystallography database such as the COD (crystallography open database, http://www.crystallography.net/) or, alternatively, inserted by hand. In our example, we have chosen to download the cif file from the COD, and store it as nacl.cif in the nacl/ directory, which is to hold our calculations:

```
#ascii#
$ mkdir nacl
$ cd nacl
```

In a simple crystal such as NaCl, it is reasonable to start with an all-default calculation up to the SCF cycle. This requires a nacl.wien file in nacl/ containing:

```
#runwien#
general
 loadcif nacl.cif
end general
initialization default
prescf default
scf default
synopsis default
```

The job is run using:

```
#ascii#
$ runwien.awk nacl.wien
```

In approximately one minute, the calculation ends. You can check that the geometry of the unit cell is the correct one by running xcrysden on the struct file:

```
#ascii#
$ xcrysden --wien_struct nacl1/nacl1.struct
```

Also, you can readily examine the .struct file and check that everything is in order. The synopsis.out file contains some useful information. Some important things to check are that the actual cell geometry corresponds to your expectations; possible core leakings and other warnings from the SCF process and the correct core/valence assignment:

```
#ascii#
Core | val el. conf. for atom : [Cl] 1S(2)2S(2)2P(6) | 3S(2)3P(5)
Core | val el. conf. for atom : [Na] 1S(2) | 2S(2)2P(6)3S(1)
```

In our case, an energy was obtained but runwien.awk indicates the presence of ghost bands:

```
#ascii#
Energy*/ry = -1248.1317470000
E fermi /ry = -0.13756
Warnings : (5)
No critic data available
(5) Ghost bands.
```

The scf output file contains more details on the problem:

```
#ascii#
:WARN : QTL-B value eq.  5.08 in Band of energy -0.27651 ATOM= 2 L=1
:WARN : You should change the E-parameter in case.in1 or use -in1new switch
```

These data indicates that a further adjustment of the basis set is needed. To this end, remove the all-default calculation:

```
#ascii#
$ rm -rf nacl*/ *.index *.out
```

Change the wien file in order to let WIEN2k calculate an appropriate basis set in the third cycle:

```
#runwien#
general
 loadcif nacl.cif
end general
```

```
initialization default
prescf default
scf
 new in1 3
end scf
synopsis default
```

Execute it:

```
#ascii#
$ runwien.awk nacl.wien
```

Note that letting WIEN2k generate a good basis set does not always work, and sometimes you will be forced to do it by hand. However, it works fine in NaCl, rendering a correct energy and no ghost band warnings:

```
#ascii#
Energy*/ry = -1248.1369720000
E fermi /ry = -0.13097
No critic data available
```

The basis set information is contained in the nacl1/nacl1.in1 file, and reads:

```
#ascii#
WFFIL        (WFPRI, SUPWF)
7.5 10 4 (R-MT*K-MAX; MAX L IN WF, V-NMT
-.43099 4 0 (GLOBAL E-PARAMETER WITH n OTHER CHOICES, global APW/LAPW)
 0  -0.23100   0.00000CONT 1
 0  -3.71800   0.00200CONT 1
 1  -0.19700   0.00000CONT 1
 1  -1.65900   0.00000CONT 1
-.43099 3 0 (GLOBAL E-PARAMETER WITH n OTHER CHOICES, global APW/LAPW)
 0  -0.21700   0.00000CONT 1
 0  -1.03900   0.00000CONT 1
 1  -0.18700   0.00000CONT 1
K-VECTORS FROM UNIT:4      -9.0       2.0   emin/emax window
```

which translates almost without modification to a new wien input file (nacl.wien):

```
#runwien#
general
 loadcif nacl.cif

 rmt 1 2.0,2.25,2.5
 rmt 2 2.0,2.25,2.5
 kpts 2000
 rkmax 7.5
 also
 kpts 2000/10000/2000
 rkmax 7.5
 also
 kpts 2000
 rkmax 7.5/10.0/0.5
end general
initialization
 lm list 1
  0 0 4 0 4 4 6 0 6 4 8 0 8 4 8 8 10 0 10 4 10 8
 end lm list
 lm list 2
  0 0 4 0 4 4 6 0 6 4 8 0 8 4 8 8 10 0 10 4 10 8
 end lm list
 orbitals 1 -.43099 0
  0 -0.231 0.000 CONT 1
  0 -3.718 0.002 STOP 1
  1 -0.197 0.000 CONT 1
  1 -1.659 0.000 CONT 1
 end orbitals
 orbitals 2 -.43099 0
  0 -0.217 0.000 CONT 1
  0 -1.039 0.000 CONT 1
  1 -0.187 0.000 CONT 1
 end orbitals
end initialization
prescf default
scf default
critic
 iws 2
 newton 1d-15
end critic
synopsis default
```

Before proceeding with the calculation, we will try to determine the best values for some other WIEN2k parameters. Concretely, rmt, the number of k-points and RKmax. The output of the nn program (nacl1/nacl1.outputnn) contains:

```
#ascii#
RMT( 1)=2.50000 AND RMT( 2)=2.50000
SUMS TO 5.00000 LT. NN-DIST= 5.32956
```

So rmt must be significantly smaller than 2.5 for both atoms as we want to explore the potential energy surface. In our example, a grid of 3x3 values of rmt for Na and Cl atoms was selected. The block:

```
#runwien#
rmt 1 2.0,2.25,2.5
rmt 2 2.0,2.25,2.5
kpts 2000
rkmax 7.5
```

calculates all the 3x3=9 possibilities found by combining the three values for the rmt of Na (2.0, 2.25 and 2.5) with the corresponding three possibilities for the Cl atom (also 2.0, 2.25 and 2.5). The kpts and rkmax parameters are fixed at 2000 and 7.5 respectively for all the 9 structures. The keyword:

```
#runwien#
also
```

starts a new grid, independent of the previous one. The keywords:

```
#runwien#
kpts 2000/10000/2000
rkmax 7.5
```

in this new block vary kpts from 2000 up to 10000 in steps of 2000 (amounting to 5 points), while rkmax is fixed to 7.5. Note that the also keyword causes this 1D parameter grid to be independent from the 2D rmt grid described above. Here, rmt for both atoms assumes its default value (2.5). Finally, some values of rkmax are also studied using the input:

```
#runwien#
also
kpts 2000
rkmax 7.5/10.0/0.5
```

Again, the also keyword uncouples the rkmax variation from the rest of the parameter grids.

The initialization section comes next in the input file. Our interest in this example is in the topological analysis of the electron density of NaCl. To that end, the LM list for the expansion of the density in the muffin tins is extended up to LMAX = 10:

```
#runwien#
lm list 1
 0 0 4 0 4 4 6 0 6 4 8 0 8 4 8 8 10 0 10 4 10 8
end lm list
lm list 2
 0 0 4 0 4 4 6 0 6 4 8 0 8 4 8 8 10 0 10 4 10 8
end lm list
```

Note that this LM list most likely will require you to recompile WIEN2k with an augmented LMAX2 parameter. However, for the sake of clarity, you can omit the lm list specification in the input file (defaults to LMAX = 6). The basis set specification follows, almost directly copied from the in1 file of our previous calculation. The prescf and scf sections are run with the default values and, finally, the localization of the critical points for all the calculated systems is requested (the critic program is needed for this step). Along with the energy, we are going to use the electron density flatness as a criterion to assess the adequacy of the WIEN2k parameters. The electron density flatness [flatness] is a measure of the planarity of the electron density in the interstitial (sometimes the term 'planarity' is used instead of 'flatness', but they are equivalent). It is calculated as the ratio between the global minimum of the electron density and the maximum density at the bond critical points. Run this calculation after removing the old output files:

```
#ascii#
$ rm -rf nacl*/ *.index *.out
$ runwien.awk nacl.wien
```

In approximately one hour, the calculation finishes. The part of the synopsis.out file relevant to us is:

```
#ascii#
Calculation variable parameters
---------------------------------------------------------------------
number of structures : 20
num   rmt 1 rmt 2 rkmax   kpts basis it. warning    energy*/ ry   \
 efermi / ry      time /s planarity morsesum                      topology
1     2.00  2.00  7.50   2000   296   9      --  -1248.1409550    \
   -0.05629          164  0.154138        0 2[4,4](0)|2[24,24]|1[48]|1[8]
2     2.25  2.00  7.50   2000   296   9      --  -1248.1415330    \
   -0.06395          146  0.156605        0 2[4,4](0)|2[24,24]|1[48]|1[8]
3     2.50  2.00  7.50   2000   296   9      --  -1248.1409760    \
   -0.07045          146  0.157923        0 2[4,4](0)|2[24,24]|1[48]|1[8]
4     2.00  2.25  7.50   2000   296   9      --  -1248.1412770    \
   -0.07409          145  0.153908        0 2[4,4](0)|2[24,24]|1[48]|1[8]
5     2.25  2.25  7.50   2000   182   9      --  -1248.1398230    \
```

[flatness] P. Mori-Sánchez, A. Martín-Pendás, and Víctor Luaña, "A classification of covalent, ionic and metallic solids based on the electron density", J. Am. Chem. Soc. 124 (2002) 14721--14723.

```
   -0.08988          114  0.154612         0 2[4,4](0)|2[24,24]|1[48]|1[8]
6     2.50  2.25  7.50   2000   182   9      --  -1248.1400320    \
   -0.09598          115  0.155934         0 2[4,4](0)|2[24,24]|1[48]|1[8]
7     2.00  2.50  7.50   2000   296   7      --  -1248.1407630    \
   -0.09291          114  0.155285         0 2[4,4](0)|2[24,24]|1[48]|1[8]
8     2.25  2.50  7.50   2000   182   9      --  -1248.1399670    \
   -0.10967          113  0.156159         0 2[4,4](0)|2[24,24]|1[48]|1[8]
9     2.50  2.50  7.50   2000   150   9      --  -1248.1388390    \
   -0.12442          101  0.158959         0 2[4,4](0)|2[24,24]|1[48]|1[8]
10    2.50  2.50  7.50   2000   150   9      --  -1248.1388390    \
   -0.12442          101  0.158959         0 2[4,4](0)|2[24,24]|1[48]|1[8]
11    2.50  2.50  7.50   4000   150   9      --  -1248.1388410    \
   -0.12441          139  0.159010         0 2[4,4](0)|2[24,24]|1[48]|1[8]
12    2.50  2.50  7.50   6000   150   9      --  -1248.1388490    \
   -0.12438          229  0.158995         0 2[4,4](0)|2[24,24]|1[48]|1[8]
13    2.50  2.50  7.50   8000   150   9      --  -1248.1388460    \
   -0.12439          268  0.159012         0 2[4,4](0)|2[24,24]|1[48]|1[8]
14    2.50  2.50  7.50  10000   150   9      --  -1248.1388450    \
   -0.12440          310  0.158970         0 2[4,4](0)|2[24,24]|1[48]|1[8]
15    2.50  2.50  7.50   2000   150   9      --  -1248.1388390    \
   -0.12442           98  0.158959         0 2[4,4](0)|2[24,24]|1[48]|1[8]
16    2.50  2.50  8.00   2000   182   9      --  -1248.1397580    \
   -0.11914          109  0.157354         0 2[4,4](0)|2[24,24]|1[48]|1[8]
17    2.50  2.50  8.50   2000   194   9      --  -1248.1402350    \
   -0.11442          119  0.157093         0 2[4,4](0)|2[24,24]|1[48]|1[8]
18    2.50  2.50  9.00   2000   272   9      --  -1248.1404990    \
   -0.11021          134  0.157530         0 2[4,4](0)|2[24,24]|1[48]|1[8]
19    2.50  2.50  9.50   2000   296   8      --  -1248.1406700    \
   -0.10617          133  0.158164         0 2[4,4](0)|2[24,24]|1[48]|1[8]
20    2.50  2.50 10.00   2000   344   7      --  -1248.1408050    \
   -0.10160          132  0.158840         0 2[4,4](0)|2[24,24]|1[48]|1[8]
```

This information can be plotted in order to select the adequate parameters.

The first 9 structures correspond to the 2D rmt grid mentioned above. Varying the rmt of any of the two atoms affects the energy in the mRy scale. The effect is similar on the flatness of the electron density, with difference near 0.001. The electron density flatness is relatively insensitive to (reasonable) variations in the muffin tin radius of atoms. The cause for this is that the flatness is defined as a ratio of densities, both of them corresponding to points that are, almost always, located in the interstitial region. Both energy and flatness plots against the rmt of both atoms do not show any clear trend. As we are interested in PES exploration, we will go on with rmt = 2 for both atoms to avoid muffin tin collisions.

The NaCl crystal is highly ionic, so saturation of k-points is achieved easily. In this calculation, it makes no major difference whether the system is calculated using 2000 or 10000 k-points as both energy and flatness remain unchanged. Concerning RKmax, the flatness seems quite insensitive to a variation in this parameter. The energy decreases on increasing RKmax, and a minimum is not observed. Contrary to kpts, rkmax is not a variational parameter and an increase in energy usually indicates problems related to linear dependencies in the basis set. We have selected kpts = 2000 and rkmax = 7.5, in order to keep the computational cost low, but a better rkmax is needed if the results are to be meaningful.

The selected parameters correspond to the first structure studied in the parameter exploration. Just for checking, we can calculate the density of states, band structure, and kdos (K-points and DOS) for it. To do this, rewrite the wien input file:

```
#runwien#
loadcheck
do 1
dosplot default
bandplot default
kdos
 kdos 1
end kdos
synopsis default
```

The previous calculation is loaded with loadcheck, and only the first structure is calculated with the do 1 keyword. Next, the density of states, spaghetti and kdos plots are requested, all with the default options. The calculation produces the neat plots:

```
#ascii#
nacl01/nacl01.dosband.1.eps
nacl01/nacl01.dosplot.1.ps
nacl01/nacl01.spaghetti.ps
```

Once the best calculation conditions have been selected, we advance to the determination of the energy against the cell volume. We select a reasonable set of volumes around the equilibrium volume, and make runwien.awk calculate the chosen points by rewriting the wien input file:

```
#runwien#
loadcheck
sweep
  rkmax 7.5
  kpts 2000
  rmt 1 2.0
  rmt 2 2.0
  with v 900/1500/100
  dosplot
  end dosplot
  bandplot
  end bandplot
  kdos
   kdos 1
  end kdos
  critic
    iws 2
    newton 1e-15
  end critic
end sweep
synopsis default
```

This input file commands runwien.awk to vary the cell volume from 900 to 1500 in steps of 100 atomic units. Note that the information about the system is loaded in with loadcheck, so it is not necessary to run general, etc. again. In addition to the topological analysis (critic), the density of states (dosplot), band structure (bandplot) and merged DOS and bandstructure (kdos) plots are issued. The WIEN2k parameters have values that correspond exactly to the first structure calculated in the parameter exploration. Therefore, you can alternatively use the input:

```
#runwien#
loadcheck
sweep
  reference struct 1
  with v 900/1500/100
  dosplot
  end dosplot
  bandplot
  end bandplot
  kdos
   kdos 1
  end kdos
  critic
    iws 2
    newton 1e-15
  end critic
end sweep
synopsis default
```

which does exactly the same thing as the former. The wien file is processed with the command:

```
#ascii#
$ runwien.awk nacl.wien
```

Let us assume that, for whatever reason, the calculation is killed (oh no!) in the middle of the sweep run. Specifically, the fourth point is affected:

```
#ascii#
[...]
[info|sweep] Calculating sweep structure #3...
[info|sweep] Extracting information from structure #3...
[info|sweep] Marking as done...
[info|sweep] Writing checkpoints...
[info|sweep] Calculating sweep structure #4...
Killed
```

To check the actual point at which the calculation was killed, you can examine the stdout as above, the 'sweep-script' temporary file created by runwien.awk or simply execute runwien.awk on the input:

```
#runwien#
loadcheck reread
synopsis default
```

This produces an output:

```
#ascii#
Variable parameters --
number of structures : 7
 a / bohr b / bohr c / bohr     v/bohr^3 it. warning    energy / ry \
    efermi / ry   time /s planarity morsesum                      topology
 9.65489  9.65489  9.65489     225.0000   9      --  -1248.1042110  \
   0.098260000       129 0.4215360       24    2[4,4](0)|1[24]|1[48]|1[8]
10.00000 10.00000 10.00000     250.0000   9      --  -1248.1258250  \
         0.036130000       138 0.1632730        0 2[4,4](0)|2[24,24]|1[48]|1[8]
10.32280 10.32280 10.32280     275.0000   9      --  -1248.1370460  \
        -0.013030000       154 0.1575500        0 2[4,4](0)|2[24,24]|1[48]|1[8]
10.62659 10.62659 10.62659     300.0000  10      --  -1248.1407190  \
        -0.052370000         0 0.0000000      n/a               n/a
10.91393 10.91393 10.91393     325.0000               --- not done ---
11.18689 11.18689 11.18689     350.0000               --- not done ---
11.44714 11.44714 11.44714     375.0000               --- not done ---
```

that indicates that the fourth sweep point is in the middle of the calculation and that the last three points were not calculated at all. To continue the interrupted job, use an input file:

```
#runwien#
loadcheck
sweep
  reference struct 1
  do 4/7
  dosplot
  end dosplot
  bandplot
  end bandplot
  kdos
   kdos 1
  end kdos
  critic
    iws 2
    newton 1e-15
  end critic
end sweep
synopsis default
```

and run it. Note that we have removed the 'with' line and included the 'do 4/7' to continue the calculation. Runwien.awk will then rewrite the wien files in the sweep directory tree with exactly the same data, and calculate the missing points. After running the input file, the result is:

```
#ascii#
Variable parameters --
number of structures : 7
 a / bohr b / bohr c / bohr     v/bohr^3 it. warning    energy / ry \
   efermi / ry   time /s planarity morsesum                      topology
 9.65489  9.65489  9.65489     225.0000   9      --  -1248.1042110  \
  0.098260000       128 0.4215360       24    2[4,4](0)|1[24]|1[48]|1[8]
10.00000 10.00000 10.00000     250.0000   9      --  -1248.1258250  \
  0.036130000       141 0.1632730        0 2[4,4](0)|2[24,24]|1[48]|1[8]
10.32280 10.32280 10.32280     275.0000   9      --  -1248.1370460  \
 -0.013030000       171 0.1575500        0 2[4,4](0)|2[24,24]|1[48]|1[8]
10.62659 10.62659 10.62659     300.0000  10      --  -1248.1407190  \
 -0.052370000       191 0.1542220        0 2[4,4](0)|2[24,24]|1[48]|1[8]
10.91393 10.91393 10.91393     325.0000   9      --  -1248.1412300  \
 -0.084610000       178 0.1508880        0 2[4,4](0)|2[24,24]|1[48]|1[8]
11.18689 11.18689 11.18689     350.0000   9      --  -1248.1391160  \
 -0.111400000       190 0.1479630        0 2[4,4](0)|2[24,24]|1[48]|1[8]
11.44714 11.44714 11.44714     375.0000   9      --  -1248.1352460  \
 -0.133760000       201 0.1450710        0 2[4,4](0)|2[24,24]|1[48]|1[8]
```

The DOS, spaghetti and KDOS plots are generated in:

```
#ascii#
nacl-sweep/nacl-sweep?/nacl-sweep?1/*.{ps,eps}
```

The data of energy can be readily graphed manually, and also using the runwien.awk input file:

```
#runwien#
loadcheck
sweep
```

```
  print energy vs v
end sweep
synopsis default
```

that generates the nacl-sweep.print1.eps file in the root directory of the calculation.

Finally, we want to obtain a estimation of the thermodynamic properties of the NaCl system by using the quasiharmonic Debye model implemented in gibbs. However, we feel that 7 points is not enough data for this task so we would like to refine the potential energy surface by adding new points intertwined with the previous grid. Additionally, it turns out that we have at our disposal a new batch calculation system so we do not want runwien.awk to calculate the new points serially. To this end, we write a new input file:

```
#runwien#
loadcheck
sweep
  do none
  with v 950/1450/100
end sweep
synopsis default
```

and execute runwien.awk on it. Due to the 'do none' keyword, no actual calculation is performed, but the directory and the wien input file for the new points (8 to 13) are generated. The user is in charge of running these points manually, either using a queue system or by any other means. For example, for point 8:

```
#ascii#
$ cd nacl-sweep/nacl-sweep08/
$ runwien.awk nacl-sweep08.wien
```

When the points from 8 to 13 are calculated, the data can be read back into runwien.awk and the energy plotted against the atomic volume using the input file:

```
#runwien#
loadcheck reread
sweep
  print energy vs v
end sweep
synopsis default
```

which compiles the information in all the structures, either manually or automatically run:

```
#ascii#
Variable parameters --
number of structures : 13
 a / bohr b / bohr c / bohr     v/bohr^3 it. warning    energy / ry\
  efermi / ry   time /s planarity morsesum                      topology
 9.65489  9.65489  9.65489     225.0000   9      --  -1248.1042110 \
 0.098260000       128 0.4215360       24    2[4,4](0)|1[24]|1[48]|1[8]
10.00000 10.00000 10.00000     250.0000   9      --  -1248.1258250 \
 0.036130000       141 0.1632730        0 2[4,4](0)|2[24,24]|1[48]|1[8]
10.32280 10.32280 10.32280     275.0000   9      --  -1248.1370460 \
-0.013030000       170 0.1575500        0 2[4,4](0)|2[24,24]|1[48]|1[8]
10.62659 10.62659 10.62659     300.0000  10      --  -1248.1407190 \
-0.052370000       191 0.1542220        0 2[4,4](0)|2[24,24]|1[48]|1[8]
10.91393 10.91393 10.91393     325.0000   9      --  -1248.1412300 \
-0.084610000       177 0.1508880        0 2[4,4](0)|2[24,24]|1[48]|1[8]
```

```
11.18689 11.18689 11.18689     350.0000   9      --  -1248.1391160 \
-0.111400000       189 0.1479630        0 2[4,4](0)|2[24,24]|1[48]|1[8]
11.44714 11.44714 11.44714     375.0000   9      --  -1248.1352460 \
-0.133760000       201 0.1450710        0 2[4,4](0)|2[24,24]|1[48]|1[8]
 9.83048  9.83048  9.83048     237.5000  10      --  -1248.1166690 \
 0.065370000       195 0.0000000      n/a                          n/a
10.16396 10.16396 10.16396     262.5000   9      --  -1248.1324490 \
 0.010100000       196 0.0000000      n/a                          n/a
10.47690 10.47690 10.47690     287.5000  10      --  -1248.1390750 \
-0.033630000       248 0.0000000      n/a                          n/a
10.77217 10.77217 10.77217     312.5000  10      --  -1248.1413790 \
-0.069280000       257 0.0000000      n/a                          n/a
11.05209 11.05209 11.05209     337.5000   9      --  -1248.1404140 \
-0.098650000       258 0.0000000      n/a                          n/a
11.31851 11.31851 11.31851     362.5000   9      --  -1248.1373680 \
-0.123110000       270 0.0000000      n/a                          n/a
```

The file nacl-sweep.print1.eps shows that the energy curve is smooth. With this extended set of data, we can run the gibbs program using the runwien.awk input:

```
#runwien#
loadcheck
gibbs
 temperature 0,278.15
end gibbs
synopsis default
```

The gibbs program is run in the subdirectory nacl-gibbs/. It writes the output file nacl.outputgibbs. The energy against volume data is fitted using a numerical procedure involving polynomials of increasing order. The Debye temperature is calculated from the static bulk modulus as obtained from the fit. The thermodynamic properties of the crystal at zero and room temperature are plotted with the above input, rendering (excerpt):

```
#ascii#
RESULTS AT P=0 FOR ALL TEMPERATURES
===================================
    T(K) V(bohr3) G(kJ/mol) U(kJ/mol) S(J/mol K) Cv(J/mol K)
-------------------------------------------------------------
    0.00   322.39   -602.20      4.83    0.00000     0.00000
  278.15   335.62   -612.96     14.36   76.39760    48.19218
```

Each point calculated in sweep is stored under the corresponding subdirectory in nacl-sweep/. The index file nacl-sweep/nacl-sweep.index contains the information relating the unit cell geometry with the numbering. These subdirectory contain also runwien.awk jobs. Thus a detailed analysis of one of these points is always possible using runwien.awk.

3 Reference guide

3.1 General conventions

WIEN2k calculations take place in a directory that is referred to as <root>. Most of the many files involved in the calculation are named as "<root>.<something>". Following this convention, the runwien input file must be called "<root>.wien".

A runwien run can give rise to many WIEN2k calculations, either because the user wants to explore the effect of modifying some model parameter or because a scan of the geometry has been called for. Every one of those calculations is done in a particular subdirectory, that is automatically created by the runwien script. We will refer to the calculations as “cases” and to their subdirectories as <case>.

The runwien input is divided into several sections, some of them mandatory and some used only when the user requires the calculation of some particular properties or techniques. Most of the runwien orders are particular to a given section, so the structure of the present guide reflects the structure of the input file. Before listing sections and keywords, there are some general caveats and advices you have to take into account when using runwien:

- Any line whose first non-blank character is '#' is interpreted as a comment and subsequently ignored. Blank lines are ignored too. Tab characters are equivalent to blanks.
- Input is case-insensitive (except for titles, file names, and similar). We will use CAPITALS in this reference to indicate runwien keywords. Other conventions used in describing runwien orders include:
  - optional keywords and data are enclosed in square brackets: “[something]”.
  - alternative settings are separated by vertical bars: “op1 | op2”.
  - most keywords expect data in the form of an integer or real number, or a character string. The type of data will be indicated by appending a suffix to the data name: “.i” for an integer, “.r” for a real, and “.s” for a string.
  - some keywords can receive as input a range of values. The range can be formed as any of: value | min/max | min/max/step, where the value, min, max, and step can be integer (range.i will be used to stress this character) or real values (range.r), depending on the particular keyword on which they appear. A range like “1/10” corresponds to {1,2,3, ..., 10} whereas “0/5/0.5” is equivalent to {0, 0.5, 1, ..., 5}.
- The crystal geometry can be read in from several sources. As a general rule, but in particular when loading from a .cif file, care must be taken with precision in the atom coordinates: a 1/3 carelessly written as 0.3333 will give rise to an incorrect Wyckoff position and to the wrong multiplicity. Use 0.333333333 and similar to make sure this does not happen. In addition, .cif files downloaded from databases use to have ms-dos newline format and need to be converted before being used with runwien. One more warning: be careful with atom names; crystallographers tend to assign names to atoms such as “O-h1”, “Ca2+” and the like. These names will confuse runwien and WIEN2k.
- Stdout of (almost) every program is stored as <root>.<nameofprogram>.err or similar. Error files coming from the scf process are assigned the name <root>.<nameofprogram>.error.
- Try to keep the structure and keyword list without tabs and extra blanks. This will make your life easier later on (see the description of RMT in the GENERAL section).

3.2 Global keywords and environments

```
#runwien#
EXIT
CLEAN [WIEN | FULL]
UNDO <section>
LOADCHECK [REREAD]
SYSTEM order.s
DO [range.i | ALL | NONE | NEW | OLD]
EXCEPT [range.i]
PARALLEL machines.s
DUPLICATE range.i
```

These keywords and environments must appear outside any section.

EXIT Terminates the runwien execution immediately.

CLEAN [WIEN | FULL] Remove some files from the <root> directory and the <cases> subdirectories. Runwien **must** know the number of directories to clean, so this keyword must come after a LOADCHECK, a GENERAL or a SWEEP section[1]. In general, only those files whose name start with <root> or <case> are involved in the cleaning. So, if you want to keep your own files safe from the cleaning procedures, just name them differently.

The default cleaning removes the "<case>.vector*" and "<case>.help*" files.

The WIEN option uses clean_lapw WIEN2K utility script to remove a large collection of files, usually not required after the SCF convergence has been achieved.

The FULL option removes a large list of files from all the directories involved. In fact the only <root> and <case> files kept will be: ".clmsum", ".clmup", ".clmdn", ".struct", ".in1", ".in2", ".inm", ".klist", ".outputst", ".scf", ".dayfile", ".incritic", ".outputcritic", ".ingibbs", ".outputgibbs", ".ps", ".eps" and ".pdf". In other words, the FULL cleaning keeps the input and output files and the intermediate files needed to run the final property calculations.

UNDO <section> Undo a section, removing its ".check" and all its information. Currently, only SWEEP, CRITIC, ELASTIC, FREE and GIBBS sections can be undone[2].

LOADCHECK [REREAD] Loads a saved calculation from the ".check" files. If the REREAD keyword is present, runwien does not rely on the ".check" information and tries to determine as much as possible from all the existing files. This way, externally calculated sweep or general structures can be loaded. REREAD does rely on a set of minimal variables: general and sweep iterations, number of non-equivalent atoms, and the like, which should never vary. Some unimportant variables like the printing level in kgen or the ones related to printrho, dosplot, etc. are plainly ignored. Be however careful when rereading a structure which has only gone through the general section: its ".struct" file is usually badly set (the struct file is heavily modified in the initialization section). When loading from a previous calculation, you must be aware that non-variable parameters can *not* be changed. That is, changing from energy conv 0.0001 to energy conv 0.00001 is not allowed and, in fact, will stop the calculation if tried. Also, specifying the same value as the one it is stored in .check is redundant, but valid in general. There is, however, one exception to this: environments (such as equiv list, etc.) can not be specified when loading checkpoints.

SYSTEM order.s Execute an external OS order. This is prone to produce any malware action in a ".wien" file inherited from an untrusted source. Due to this potential security risk, this command is disabled by default. Should you want to use this order, uncomment the corresponding line in the beginning of runwien (variable const_allow_system).

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] The DO/EXCEPT pair can be used to select a subset of calculations. The default, in absence of DO/EXCEPT orders, is to perform only the calculation of new structures, avoiding the cases already declared as done within the ".check" status files. The apparition of a

[1] Before cleaning runwien needs to determine the value of two important internal variables: general_iterations and sweep_iterations. They are calculated in general section and sweep section respectively. Alternatively, they can be read from the checkfiles in the course of the LOADCHECK task. Both variables are used to determine the number and the names of the <case> directories involved in the run. Runwien prints to the standard output the cleaning operations done, so you can check if everything has been correctly done.

[2] The remaining sections are GENERAL, INITIALIZATION, PRESCF, SCF, PRINTRHO, DOSPLOT, RXPLOT, BANDPLOT, KDOS, AIM, and SYNOPSIS. You can run PRINTRHO, DOSPLOT, RXPLOT, BANDPLOT, AIM and SYNOPSIS several times, each new run overwriting the previous files. You can not undo GENERAL, INITIALIZATION, PRESCF and SCF, but you can rerun the old structures with LOADCHECK / DO OLD or similar (see below). However, it is not possible to change their parameters. Of course, you can always resort to manual execution and rereading the results.

DO/EXCEPT set outside any section changes globally the default from that point on. Contrarily, a DO/EXCEPT declared within a section range will only affect locally to that particular section.

As an exception, the ELASTIC, FREE, SWEEP, GIBBS, and SYNOPSIS sections use particular DO/EXCEPT mechanisms:

- ELASTIC ignores the DO/EXCEPT options, but the deformations required to calculate some elastic constants can be skipped with the NOSEND order.
- The FREE section is only affected by the local DO/EXCEPT declaration, with a particular syntax allowing element symbols (See the FREE section keywords).
- The SWEEP section uses only the local DO/EXCEPT declarations and ignores the global values.
- The DO/EXCEPT list has no meaning within the GIBBS and SYNOPSIS tasks.

The DO/EXCEPT orders can be qualified with:

range.i: A range of integers indicating the cases to include or exclude. The range can be any combination of: a list of integers (e.g. 1,3,9); a first/last pair (e.g. 2/10); a first/last/step triplet (e.g. 2/10/2).

ALL: Include all, new and old, cases.

NONE: Exclude all cases.

NEW: Include only the new cases. This is the global default.

OLD: Include just the old cases.

In the next example:

```
#runwien#
 general
    ...
    lmax 1/20
    # This involves 20 points or structs
    ...
 end general
 initialization default
 # Change default behavior for next sections:
 do 4/10
 except 9
 # Run only points: 4, 5, 6, 7, 8, and 10
 prescf default
 scf default
 critic
    do 3
    # Run critic task only on point 3
 end critic
```

the GENERAL and INITIALIZATION sections will affect to 20 points, the PRESCF and SCF steps to 6 cases, and the CRITIC task to a single case.

Default: calculate only the NEW structures.

PARALLEL machines.s Information about the details of a run using parallelization is done through a .machines file in WIEN2k (see WIEN2k user's guide for details). The PARALLEL keyword indicates the location of the .machines file and commands runwien to run the calculations in parallel, using the -p flag. The machines file is copied to the directory where the calculations are actually carried out. The file is also passed down to the elastic, free and sweep sections.

The method of generating .machines files in a batch system job using a script (some examples are given in the WIEN2k FAQ) can also be applied, as long as there is a pointer to the newly generated .machines file.

If no machines.s is indicated, the parallelization is explicitly negated, canceling the effect of previous PARALLEL orders. Note that PARALLEL related variables are not stored in the check files, so this keyword must be indicated in a runwien execution using LOADCHECK if it is intended to run parallelized.

Default: serial calculations.

DUPLICATE range.i Duplicate the calculations indicated in the range.i. The numbers parsed from range.i are treated sequentially. For each structure number, increment the structure counter and copies the whole directory, renaming its contents. The checkfiles are automatically updated.

3.3 General section

3.3.1 Syntax

```
#runwien#
GENERAL
  LOADCIF filecif.s
  LOADSTRUCT filestruct.s
  LATTICE lattice_letter.s
  EQUIV LIST environment
    x.r  y.r  z.r
  END EQUIV LIST
  SPGLIST spg.s [RHOMB]
    at.s x.r y.r z.r
    ...
  END SPGLIST
  CELL PARAMETERS  a.r b.r c.r  alpha.r beta.r gamma.r
  TITLE runtitle.s
  SPINPOLARIZED {YES|NO}
  RELATIVISTIC {YES|NO}
  NPT [atom.i|*] {range.i | AUTO} [,...]
  RMT [atom.i|*] {range.r | AUTO} [,...]
  R0 [atom.i|*] {range.r | AUTO} [,...]
  RKMAX {AUTO | range.r} [,...]
  LMAX  {AUTO | range.i} [,...]
  LNSMAX  {AUTO | range.i} [,...]
  GMAX {AUTO | range.i} [,...]
  MIX {AUTO | range.r} [,...]
  KPTS {AUTO | range.i} [,...]
  ALSO
  LINK var1.s var2.s [var3.s ... ]
  DO [range.i | ALL | NONE | NEW | OLD]
  EXCEPT [range.i]
  LDA+U {SIC|AMF|HMF}
   iat1.i {l1.i|l1.s} [defu.r defj.r]
   iat2.i {l2.i|l2.s} [defu.r defj.r]
   ...
  END LDA+U
  U [line.i] urange.r
  J [line.i] urange.r
END GENERAL
```

The general section is, perhaps, the most important one in runwien. It reads the characteristic calculation variables (structural and WIEN2k), verifies needed parameters are set, parses expressions in the .wien input, loads info from external files, creates header in stdout and whole .index file, creates the iteration counter and the directory tree structure and starts the .struct files that will be completed later with data from the initialization section.

If a previous runwien calculation must be loaded, use LOADCHECK. Otherwise, runwien will issue an error when it finds the old directories and exit. The loaded structures become "old" and are not messed with, unless required by an do or except keyword. Also, you can not change non-variable parameters from the old calculation such as cell constants, atomic positions, etc. Should you want to do this, start another runwien job in a different directory.

The creation of structures using variables parameters is fairly easy. These variable parameters are npt, rmt, r0, rkmax, lmax, lnsmax, gmax, mix and kpts (see below for a definition). The user can vary one of this parameters with a line like:

```
#runwien#
rkmax 7.5,8.0,8.5
```

Thus generating three structures, each with a different rkmax value, but all with the same values for the rest of the options. If the user adds another variable parameter:

```
#runwien#
rkmax 7.5,8.0,8.5
kpts  1000,2000
```

There will be 6 structures, whose values will be those of the direct product of both sets.

There is a mechanism to avoid the coupling of variables: the also keyword. When an also keyword appears, further variable parameter specifications will be independent of the previous. For example:

```
#runwien#
rkmax 7.5,8.0,8.5
also
kpts  1000,2000
```

will only create **five** structures: 3 with rkmax = 7.5, 8.0 and 8.5 and the default value for kpts and 2 with kpts = 1000, 2000 and the default value for rkmax.

It is important to keep in mind that when an ALSO order is found, all the variables are reset to their defaults. That is, if you set the mixing parameter (mix) to 0.05 and then write an ALSO, structures created after that point will have the default value of mix (0.40) and not 0.05.

There is another way to couple variables: the LINK keyword. This keyword accepts two or more variables. When the structures are created, these variables run simultaneously over their respective values. For example:

```
#runwien#
link lmax lnsmax
lmax 10,11,12
lnsmax 4,5,6
```

will create **three** structures, with values of (lmax,lnsmax) = (10,4) , (11,5) , (12,6). The variables keep linked even when an ALSO keyword is found. All the variables linked must have the same number of values, and runwien will issue an error if this requirement is not met.

For a calculation to start and to be correctly run, all necessary data must be provided through input. There is a relatively small set of indispensable parameters: lattice type, equivalent atom list and positions in the unit cell and cell parameters. This information can be provided by:

- Indicating LATTICE, EQUIV list and CELL PARAMETERS in the general section.

- Indicating SPGLIST and CELL PARAMETERS in the general section.
- Loading this information from a .cif file, with the LOADCIF keyword.
- Loading it from a .struct file, using the LOADSTRUCT keyword.
- Loading it from a check file (generated in a previous run), with LOADCHECK or LOADCHECK REREAD.

3.3.2 Keywords and environments

LATTICE lattice_letter.s Crystal Bravais lattice, in WIEN2k naming convention. That is, P for primitive, F for face-centered, B for body-centered, H for hexagonal, etc. See WIEN2k user's guide for details.

R lattices must be handled with special care. In a R lattice, the cell parameters are expected to be those of an hexagonal cell, while the atomic positions are given in a rhombohedral setup. The conversion between these two is uniquely defined, and documented in the WIEN2k user's guide.

Compulsory parameter!

EQUIV LIST environment Formed as

```
EQUIV LIST symbol.s
    x.r y.r z.r
    ...
END EQUIV LIST
```

List of equivalent positions for a non-equivalent atom in the unit cell, in crystallographic units. Symbol is the symbol for the atom (case insensitive). The symbol representing an atom uses the same conventions as WIEN2k: the identity of the atom is determined from the first two (one) letters in the name. Further additions to the atom symbol are allowed, and they are passed down to the struct file.

Several equiv list environments are allowed, each of them corresponding to a new non-equivalent atom.

Compulsory parameter!

SPGLIST environment Formed as

```
SPGLIST spg.s [RHOMB]
    at.s x.r y.r z.r
    ...
END SPGLIST
```

The spglist environment determines the crystal geometry completely, along with the cell parameters. It accepts the space group label (spg.s) and the list of non-equivalent atoms, given as the atomic symbol (at.s) and the crystallographic coordinates of a representative (x.r y.r and z.r). The atoms given in the body of spglist are automatically replicated and written to the struct file. Runwien calls the WIEN2k's spacegroup utility to find the rest of equivalent atoms in the unit cell. Note the difference in behaviour to that of EQUIV LIST: only one atom for each non-equivalent atom type is required to be present in the environment.

In the case of crystals that may be described using either hexagonal or rhombohedral cells, two options are available. The first (and default) is writing the cell parameters and atomic positions in the hexagonal setup. The other, is to write both cell parameters and atomic coordinates using the rhombohedral cell. In the latter case, the keyword RHOMB must be passed to SPGLIST.

The appropriate conversion for working with WIEN2k is done by Runwien. The struct file may end up having a 'H' lattice (hexagonal cell parameters and atomic coordinates) or a 'R' lattice (hexagonal cell parameters but rhombohedral atomic coordinates, with a conversion matrix defined in WIEN2k user's guide). The choice depends on the symbol in the list below.

In some cases, the convention (as consigned in the International Tables for Crystallography) indicates two different cell origins for the same crystal. This usually happens in the space groups where the position of maximum local symmetry in the unit cell does not correspond to the points where the inversion center is located. In such cases, a second origin is defined at these points.

In WIEN2k, it is computationally cheaper to place the origin at a point where inversion symmetry is present, so that the real versions of the code are used. Therefore, in cases where ambiguity could arise, spacegroup assumes the second origin has been selected.

The space group label follows the format accepted by spacegroup, which is also used in the w2web interface. The exhaustive list of space group labels follows. The lattice of the struct file generated using the corresponding space group is also indicated. Note that there are more 'Spacegroup internal label's than 'Short label's. This allows the user to specify the orientation of a given privileged axis in the crystal.

Space group labels		
Short label	Spacegroup internal label	Associated lattice
p1	p1	P
p-1	-p1	P
p2	p2x	P
p2	p2y	P
p2	p2z	P
p21	p2xa	P
p21	p2yb	P
p21	p2zc	P
c2	c2x	CXY
a2	a2y	CYZ
b2	b2z	CXZ
b2	b2x	CXZ
c2	c2y	CXY
a2	a2z	CYZ
pm	p-2x	P
pm	p-2y	P
pm	p-2z	P
pc	p-2xc	P
pa	p-2ya	P
pb	p-2zb	P
pb	p-2xb	P
pc	p-2yc	P
pa	p-2za	P
pn	p-2xbc	P

pn	p-2yac	P
pn	p-2zab	P
cm	c-2x	CXY
am	a-2y	CYZ
bm	b-2z	CXZ
bm	b-2x	CXZ
cm	c-2y	CXY
am	a-2z	CYZ
cc	c-2xc	CXY
aa	a-2ya	CYZ
bb	b-2zb	CXZ
bb	b-2xb	CXZ
cc	c-2yc	CXY
aa	a-2za	CYZ
p2/m	-p2x	P
p2/m	-p2y	P
p2/m	-p2z	P
p21/m	-p2xa	P
p21/m	-p2yb	P
p21/m	-p2zc	P
c2/m	-c2x	CXY
a2/m	-a2y	CYZ
b2/m	-b2z	CXZ
b2/m	-b2x	CXZ
c2/m	-c2y	CXY
a2/m	-a2z	CYZ
p2/c	-p2xc	P
p2/a	-p2ya	P
p2/b	-p2zb	P
p2/b	-p2xb	P
p2/c	-p2yc	P
p2/a	-p2za	P
p2/n	-p2xbc	P
p2/n	-p2yac	P
p2/n	-p2zab	P
p21/c	-p2xca	P
p21/a	-p2yab	P
p21/b	-p2zbc	P
p21/b	-p2xba	P
p21/c	-p2ycb	P
p21/a	-p2zac	P
p21/n	-p2xabc	P
p21/n	-p2yabc	P

p21/n	-p2zabc	P
c2/c	-c2xc	CXY
a2/a	-a2ya	CYZ
b2/b	-b2zb	CXZ
b2/b	-b2xb	CXZ
c2/c	-c2yc	CXY
a2/a	-a2za	CYZ
p222	p2z;2x	P
p2221	p2zc;2x	P
p2122	p2xa;2y	P
p2212	p2yb;2z	P
p21212	p2z;2xab	P
p22121	p2x;2ybc	P
p21221	p2y;2zca	P
p212121	p2zac;2xab	P
c2221	c2zc;2x	CXY
a2122	a2xa;2y	CYZ
b2212	b2yb;2z	CXZ
c222	c2z;2x	CXY
a222	a2x;2y	CYZ
b222	b2y;2z	CXZ
f222	f2z;2x	F
i222	i2z;2x	B
i212121	i2zac;2xab	B
pmm2	p2z;-2x	P
p2mm	p2x;-2y	P
pm2m	p2y;-2z	P
pmc21	p2zc;-2x	P
p21ma	p2xa;-2y	P
pb21m	p2yb;-2z	P
pcm21	p2zc;-2y	P
p21am	p2xa;-2z	P
pm21b	p2yb;-2x	P
pcc2	p2z;-2xc	P
p2aa	p2x;-2ya	P
pb2b	p2y;-2zb	P
pma2	p2z;-2xa	P
p2mb	p2x;-2yb	P
pc2m	p2y;-2zc	P
pbm2	p2z;-2yb	P
p2cm	p2x;-2zc	P
pm2a	p2y;-2xa	P
pca21	p2zc;-2xac	P

p21ab	p2xa;-2yba	P
pc21b	p2yb;-2zcb	P
pbc21	p2zc;-2ybc	P
p21ca	p2xa;-2zca	P
pb21a	p2yb;-2xab	P
pnc2	p2z;-2xbc	P
p2na	p2x;-2yca	P
pb2n	p2y;-2zab	P
pcn2	p2z;-2yac	P
p2an	p2x;-2zba	P
pn2b	p2y;-2xcb	P
pmn21	p2zac;-2x	P
p21mn	p2xba;-2y	P
pn21m	p2ycb;-2z	P
pnm21	p2zbc;-2y	P
p21nm	p2xca;-2z	P
pm21n	p2yab;-2x	P
pba2	p2z;-2xab	P
p2cb	p2x;-2ybc	P
pc2a	p2y;-2zca	P
pna21	p2zc;-2xn	P
p21nb	p2xa;-2yn	P
pc21n	p2yb;-2zn	P
pbn21	p2zc;-2yn	P
p21cn	p2xa;-2zn	P
pn21a	p2yb;-2xn	P
pnn2	p2z;-2xn	P
p2nn	p2x;-2yn	P
pn2n	p2y;-2zn	P
cmm2	c2z;-2x	CXY
a2mm	a2x;-2y	CYZ
bm2m	b2y;-2z	CXZ
cmc21	c2zc;-2x	CXY
a21ma	a2xa;-2y	CYZ
bb21m	b2yb;-2z	CXZ
ccm21	c2zc;-2y	CXY
a21am	a2xa;-2z	CYZ
bm21b	b2yb;-2x	CXZ
ccc2	c2z;-2xc	CXY
a2aa	a2x;-2ya	CYZ
bb2b	b2y;-2zb	CXZ
amm2	a2z;-2x	CYZ
b2mm	b2x;-2y	CXZ

cm2m	c2y;-2z	CXY
abm2	a2z;-2xb	CYZ
b2cm	b2x;-2yc	CXZ
cm2a	c2y;-2za	CXY
bma2	b2z;-2ya	CXZ
c2mb	c2x;-2zb	CXY
ac2m	a2y;-2xc	CYZ
ama2	a2z;-2xa	CYZ
b2mb	b2x;-2yb	CXZ
cc2m	c2y;-2zc	CXY
bbm2	b2z;-2yb	CXZ
c2cm	c2x;-2zc	CXY
am2a	a2y;-2xa	CYZ
aba2	a2z;-2xab	CYZ
b2cb	b2x;-2ybc	CXZ
cc2a	c2y;-2zca	CXY
bba2	b2z;-2yba	CXZ
c2cb	c2x;-2zcb	CXY
ac2a	a2y;-2xac	CYZ
fmm2	f2z;-2x	F
f2mm	f2x;-2y	F
fm2m	f2y;-2z	F
fdd2	f2z;-2xd	F
f2dd	f2x;-2yd	F
fd2d	f2y;-2zd	F
imm2	i2z;-2x	B
i2mm	i2x;-2y	B
im2m	i2y;-2z	B
iba2	i2z;-2xab	B
i2cb	i2x;-2ybc	B
ic2a	i2y;-2zca	B
ima2	i2z;-2xa	B
i2mb	i2x;-2yb	B
ic2m	i2y;-2zc	B
ibm2	i2z;-2yb	B
i2cm	i2x;-2zc	B
im2a	i2y;-2xa	B
pmmm	-p-2z;-2x	P
pnnn	-p-2zab;-2xbc	P
pccm	-p-2z;-2xc	P
pmaa	-p-2x;-2ya	P
pbmb	-p-2y;-2zb	P
pban	-p-2zab;-2xb	P

pncb	-p-2xbc;-2yc	P
pcna	-p-2yca;-2za	P
pmma	-p-2za;-2xa	P
pbmm	-p-2xb;-2yb	P
pmcm	-p-2yc;-2zc	P
pmam	-p-2ya;-2xa	P
pmmb	-p-2zb;-2yb	P
pcmm	-p-2xc;-2zc	P
pnna	-p-2za;-2xbc	P
pbnn	-p-2xb;-2yca	P
pncn	-p-2yc;-2zab	P
pnan	-p-2ya;-2xbc	P
pnnb	-p-2zb;-2yca	P
pcnn	-p-2xc;-2zab	P
pmna	-p-2zac;-2x	P
pbmn	-p-2xba;-2y	P
pncm	-p-2ycb;-2z	P
pman	-p-2yab;-2x	P
pnmb	-p-2zbc;-2y	P
pcnm	-p-2xca;-2z	P
pcca	-p-2za;-2xac	P
pbaa	-p-2xb;-2yba	P
pbcb	-p-2yc;-2zcb	P
pbab	-p-2ya;-2xab	P
pccb	-p-2zb;-2ybc	P
pcaa	-p-2xc;-2zca	P
pbam	-p-2z;-2xab	P
pmcb	-p-2x;-2ybc	P
pcma	-p-2y;-2zca	P
pccn	-p-2zab;-2xac	P
pnaa	-p-2xbc;-2yba	P
pbnb	-p-2yca;-2zcb	P
pbcm	-p-2zc;-2xb	P
pmca	-p-2xa;-2yc	P
pbma	-p-2yb;-2za	P
pcmb	-p-2yb;-2xc	P
pcam	-p-2zc;-2ya	P
pmab	-p-2xa;-2zb	P
pnnm	-p-2z;-2xn	P
pmnn	-p-2x;-2yn	P
pnmn	-p-2y;-2zn	P
pmmn	-p-2zab;-2xa	P
pnmm	-p-2xbc;-2yb	P

pmnm	-p-2yca;-2zc	P
pbcn	-p-2zn;-2xab	P
pnca	-p-2xn;-2ybc	P
pbna	-p-2yn;-2zca	P
pcnb	-p-2yn;-2xac	P
pcan	-p-2zn;-2yba	P
pnab	-p-2xn;-2zcb	P
pbca	-p-2zac;-2xab	P
pcab	-p-2yab;-2xac	P
pnma	-p-2zac;-2xn	P
pbnm	-p-2xba;-2yn	P
pmcn	-p-2ycb;-2zn	P
pnam	-p-2yab;-2xn	P
pmnb	-p-2zbc;-2yn	P
pcmn	-p-2xca;-2zn	P
cmcm	-c-2zc;-2x	CXY
amma	-a-2xa;-2y	CYZ
bbmm	-b-2yb;-2z	CXZ
bmmb	-b-2yb;-2x	CXZ
ccmm	-c-2zc;-2y	CXY
amam	-a-2xa;-2z	CYZ
cmca	-c-2zac;-2x	CXY
abma	-a-2xba;-2y	CYZ
bbcm	-b-2ycb;-2z	CXZ
bmab	-b-2yab;-2x	CXZ
ccmb	-c-2zbc;-2y	CXY
acam	-a-2xca;-2z	CYZ
cmmm	-c-2z;-2x	CXY
ammm	-a-2x;-2y	CYZ
bmmm	-b-2y;-2z	CXZ
cccm	-c-2z;-2xc	CXY
amaa	-a-2x;-2ya	CYZ
bbmb	-b-2y;-2zb	CXZ
cmma	-c-2za;-2x	CXY
abmm	-a-2xb;-2y	CYZ
bmcm	-b-2yc;-2z	CXZ
bmam	-b-2ya;-2x	CXZ
cmmb	-c-2zb;-2y	CXY
acmm	-a-2xc;-2z	CYZ
ccca	-c-2za;-2xac	CXY
abaa	-a-2xb;-2yba	CYZ
bbcb	-b-2yc;-2zcb	CXZ
bbab	-b-2ya;-2xab	CXZ

cccb	-c-2zb;-2ybc	CXY
acaa	-a-2xc;-2zca	CYZ
fmmm	-f-2z;-2x	F
fddd	-f-2zuv;-2xvw	F
immm	-i-2z;-2x	B
ibam	-i-2z;-2xab	B
imcb	-i-2x;-2ybc	B
icma	-i-2y;-2zca	B
ibca	-i-2zac;-2xab	B
icab	-i-2yab;-2xac	B
imma	-i-2zac;-2x	B
ibmm	-i-2xba;-2y	B
imcm	-i-2ycb;-2z	B
imam	-i-2yab;-2x	B
immb	-i-2zbc;-2y	B
icmm	-i-2xca;-2z	B
p4	p4	P
p41	p41	P
p42	p4c	P
p43	p43	P
i4	i4	B
i41	i41b	B
p-4	p-4	P
i-4	i-4	B
p4/m	-p4	P
p42/m	-p4c	P
p4/n	-p4a	P
p42/n	-p4bc	P
i4/m	-i4	B
i41/a	-i4ad	B
p422	p4;2	P
p4212	p4ab;2ab	P
p4122	p41;2c	P
p41212	p43n;2nw	P
p4222	p4c;2	P
p42212	p4n;2n	P
p4322	p43;2c	P
p43212	p41n;2abw	P
i422	i4;2	B
i4122	i41b;2bw	B
p4mm	p4;-2	P
p4bm	p4;-2ab	P
p42cm	p4c;-2c	P

p42nm	p4n;-2n	P
p4cc	p4;-2c	P
p4nc	p4;-2n	P
p42mc	p4c;-2	P
p42bc	p4c;-2ab	P
i4mm	i4;-2	B
i4cm	i4;-2ab	B
i41md	i41b;-2	B
i41cd	i41b;-2c	B
p-42m	p-4;2	P
p-42c	p-4;2c	P
p-421m	p-4;2ab	P
p-421c	p-4;2n	P
p-4m2	p-4;-2	P
p-4c2	p-4;-2c	P
p-4b2	p-4;-2ab	P
p-4n2	p-4;-2n	P
i-4m2	i-4;-2	B
i-4c2	i-4;-2c	B
i-42m	i-4;2	B
i-42d	i-4;2bw	B
p4/mmm	-p4;-2	P
p4/mcc	-p4;-2c	P
p4/nbm	-p4a;-2b	P
p4/nnc	-p4a;-2bc	P
p4/mbm	-p4;-2ab	P
p4/mnc	-p4;-2n	P
p4/nmm	-p4a;-2a	P
p4/ncc	-p4a;-2ac	P
p42/mmc	-p4c;-2	P
p42/mcm	-p4c;-2c	P
p42/nbc	-p4ac;-2b	P
p42/nnm	-p4ac;-2bc	P
p42/mbc	-p4c;-2ab	P
p42/mnm	-p4n;-2n	P
p42/nmc	-p4ac;-2a	P
p42/ncm	-p4ac;-2ac	P
i4/mmm	-i4;-2	B
i4/mcm	-i4;-2c	B
i41/amd	-i4bd;-2	B
i41/acd	-i4bd;-2c	B
p3	p3	H
p31	p31	H

p32	p32	H
r3	r3	R
p-3	-p3	H
r-3	-r3	R
p312	p3;2	H
p321	p3;2“	H
p3112	p31;2#0,0,1/3	H
p3121	p31;2”	H
p3212	p32;2#0,0,1/6	H
p3221	p32;2“	H
r32	r3;2”	R
p3m1	p3;-2“	H
p31m	p3;-2	H
p3c1	p3;-2”c	H
p31c	p3;-2c	H
r3m	r3;-2“	R
r3c	r3;-2”c	R
p-31m	-p3;-2	H
p-31c	-p3;-2c	H
p-3m1	-p3;-2“	H
p-3c1	-p3;-2”c	H
r-3m	-r3;-2“	R
r-3c	-r3;-2”c	R
p6	p6	H
p61	p61	H
p65	p65	H
p62	p62	H
p64	p64	H
p63	p6c	H
p-6	p-6	H
p6/m	-p6	H
p63/m	-p6c	H
p622	p6;2	H
p6122	p61;2#0,0,-1/12	H
p6522	p65;2#0,0,1/12	H
p6222	p62;2#0,0,1/3	H
p6422	p64;2#0,0,1/6	H
p6322	p6c;2#0,0,1/4	H
p6mm	p6;-2	H
p6cc	p6;-2c	H
p63cm	p6c;-2	H
p63mc	p6c;-2c	H
p-6m2	p-6;2	H

p-6c2	p-6c;2	H
p-62m	p-6;-2	H
p-62c	p-6c;-2c	H
p6/mmm	-p6;-2	H
p6/mcc	-p6;-2c	H
p63/mcm	-p6c;-2	H
p63/mmc	-p6c;-2c	H
p23	p2;2;3	P
f23	f2;2;3	F
i23	i2;2;3	B
p213	p2ac;2ab;3	P
i213	i2ac;2ab;3	B
pm-3	-p2;2;3	P
pn-3	-p2ab;2bc;3	P
fm-3	-f2;2;3	F
fd-3	-f2uv;2vw;3	F
im-3	-i2;2;3	B
pa-3	-p2ac;2ab;3	P
ia-3	-i2ac;2ab;3	B
p432	p4;2;3	P
p4232	p4n;2;3	P
f432	f4;2;3	F
f4132	f4d;2;3	F
i432	i4;2;3	B
p4332	p4bdn;2ab;3	P
p4132	p4bd;2ab;3	P
i4132	i4bd;2ab;3	B
p-43m	p-4;2;3	P
f-43m	f-4;2;3	F
i-43m	i-4;2;3	B
p-43n	p-4n;2;3	P
f-43c	f-4c;2;3	F
i-43d	i-4bd;2ab;3	B
pm-3m	-p4;2;3	P
pn-3n	-p4a;2bc;3	P
pm-3n	-p4n;2;3	P
pn-3m	-p4bc;2bc;3	P
fm-3m	-f4;2;3	F
fm-3c	-f4n;2;3	F
fd-3m	-f4vw;2vw;3	F
fd-3c	-f4ud;2vw;3	F
im-3m	-i4;2;3	B
ia-3d	-i4bd;2ab;3	B

Compulsory parameter!

CELL PARAMETERS [ANGSTROM] a.r b.r c.r alpha.r beta.r gamma.r Unit cell parameters in bohr and sexagesimal degrees. If the ANGSTROM keyword appears after CELL PARAMETERS, a.r, b.r and c.r are to be in angstrom.

Compulsory parameter!

TITLE runtitle.s Sets a title for the calculation. This will be used on several places, including titles within gnuplot scripts. Including single or double quotes will interfere severely. To be safe you should stick to the characters: [0..9a..zA..Z_-.,:;].

Default: runwien lapw calculation, v. <runwien-version>.

SPINPOLARIZED {YES|NO} Spin-polarized calculation. Express the system density as a sum of spin-up and spin-down contributions, and calculate each of them separately.

Default: no.

RELATIVISTIC {YES|NO} Relativistic calculation of the core electron density.

Default: yes.

NPT [atom.i|*] {range.i | AUTO} [,...] Number of radial mesh points. It must be an odd number. The WIEN2k manual recommends 381 for LDA calculations, 781 for GGA. The use of a first/last or first/last/step range will start a scan of NPT values. The AUTO keyword selects the default value for NPT.

A NPT order affects to a single atom type (by default to the first atom in the input), indicated by the integer atom.i. The number atom.i is a non-equivalent atom index, pointing to the corresponding EQUIV LIST. The non-equivalent atom index are assigned in order of appearance of the EQUIV LIST environment (1 for the first atom, 2 for the second atom, etc.). If a "*" label replaces the atom.i integer, the NPT order affects all the atoms in the crystal.

An order like "NPT 300 /500/100" will confound the parser into assuming that 300 is the atom number rather than the first value of a range. If your order omits the number of the atom, be careful to avoid white characters within the declared range.

Default: 781 for every atom. If no atom index is indicated, 1 is assumed.

RMT [atom.i|*] {range.r | AUTO} [,...] Muffin tin radius, in bohr, for the indicated non-equivalent atom. The AUTO keyword lets WIEN2k choose the RMT according to the geometry of the system. Other possibilities are:

- a range (first/last or first/last/step) asks runwien to perform a set of calculations with a variable RMT.
- values like x% correspond to multiply the internal RMT value by a factor according to the formula rmt = (auto-value)*(1+x/100). If the % symbol occurs anywhere within a range, each RMT value will be interpreted as a multiplying factor. This is the case of, for instance -3/-4/-5%, which is equivalent to -3%/-4%/-5%.
- a list of values or ranges separated by commas is also admisible.

A RMT order affects to a single atom type (by default to the first atom in the input). An order like "RMT 3 /5/0.5" will confound the parser into assuming that 3 is the atom number rather than the first value of a range. If your order omits the number of the atom, be careful to avoid white characters within the declared range. If a "*" label replaces the atom.i integer, the RMT order affects all the atoms in the crystal.

Default: AUTO (let WIEN2k determine it), if no atom is indicated, 1 is assumed.

R0 [atom.i|*] {AUTO | range.r} [,...] First radial mesh point, in bohr, for the indicated non-equivalent atom . Wien2k manual recommends a value between 0.0005 and 0.00005. Similarly to most orders in this section, AUTO selects the standard WIEN2k value. A list of values, a range or, even, a list of ranges is also acceptable.

A R0 order affects to a single atom type (by default to the first atom in the input), indicated by the integer atom.i. The number atom.i is a non-equivalent atom index, pointing to the corresponding EQUIV LIST. The non-equivalent atom index are assigned in order of appearance of the EQUIV LIST environment (1 for the first atom, 2 for the second atom, etc.). If your order omits the index of the atom (which amounts to applying the R0 variaton to the first atom), be careful to avoid white characters within the declared range. If a "*" label replaces the atom.i integer, the R0 order affects all the atoms in the crystal.

Default: 0.0001. If no atom index is indicated, 1 is assumed.

RKMAX {AUTO | range.r} [,...] Give a value (or a list of values) for the RMT*KMAX product, where KMAX is the maximum k for plane waves and RMT is the smallest of the muffin tin radii of the different types of atoms. This parameter does not appear on the .struct input file, but we have included it within the GENERAL section to let the user explore different values: different RKMAX need different directories and sets of initialization files, even when their .struct are the same. The same reason applies to other parameters included here, like LMAX, LNSMAX, and GMAX.

Default: AUTO (assume WIEN2k proposed value).

LMAX {AUTO | range.i} [,...] Maximum value of l to be used on the spherical harmonics expansion inside the muffin tins. A large value is needed to satisfy continuity of the electron density at the muffin tin surface but there are limitations hard coded in the WIEN2k source.

Default: AUTO (don't change WIEN2k proposed value).

LNSMAX {AUTO | range.i} [,...] Maximum l value for spherical harmonics expansion used to computate matrix elements with non-muffin tin functions.

Default: AUTO (don't change WIEN2k proposed value).

GMAX {AUTO | range.i} [,...] Maximum reciprocal vector used on the fourier expansion of electron density. WIEN2k manual recommends using a large GMAX (25) for systems with short H bonds. GGA calculations also require larger GMAX than the L(S)DA calculations (a value of 14 is typical in this case).

Default: AUTO (don't change WIEN2k proposed value).

MIX {AUTO | range.r} [,...] Mixing factor used in the creation of the new charge density.

Default: AUTO (don't change WIEN2k proposed value).

KPTS {AUTO | range.i} [,...] Number of k-points in the first brillouin zone.

Default: 1000.

ALSO Use this keyword to separate two independent sets of structures. For instance:

```
#runwien#
rkmax 3.0
kpts 1000/5000/2000
also
rkmax 3.2
kpts 1000/5000/2000
```

will explore the effect of the KPTS parameter for two different values of RMT. Every one of the explorations separated by ALSO will be done independently.

Default: --.

LINK var1.s var2.s [var3.s ...] Link the indicated variables so that they vary simultaneously. The linking must be done between variables that receive the same number of values or a mismatch error will be found. The variable names are the same as the keywords described before: RKMAX, GMAX, etc. In the case of RMT, RMT1 corresponds to the first atom, RMT2 to the second, etc. The same syntax is used in NPT (NPT1, NPT2,...) and R0 (R01, R02,...). Note that runwien.awk input is case-insensitive. The linking is global and affects every ALSO block. Several LINK keywords may be issued, in order to set more than one ligature among variables. However, the sets of variables linked must be disjoint.

Default: --.

LOADCIF filecif.s Use the structural data from the indicated cif file to complete the system geometry. Explicitly set data will not be overwriten. For instance:

```
#runwien#
loadcif graphite.cif
cell 4.8 4.8 6.5 90 90 120
```

will keep the cell parameters from the explicit CELL order, but read the rest of the structural data from the 'graphite.cif' file.

The LOADCIF task depends on the cif2struct code distributed with the WIEN2k package. Cif files should be used with caution, particularly those files coming from a external database. In particular:

- Atomic coordinates are usually stored with low precision. This has bad consequences for symmetry determined values like 1/3 and 2/3. A coordinate of 0.3333 instead of 1/3 can produce a spurious atom when the nonequivalent position is replicated by symmetry. The best action in this case is hand editing this file to change these coordinates to something like 0.333333333.

Note that the RMT, NPT and R0 values generated by cif2struct are not meaningful, so they are completely ignored. These parameters need to be set in the general section specification.

Default: --.

LOADSTRUCT filestruct.s Load the information contained in the indicated struct file. The behavior of this order is analogous to that of LOADCIF, that is, values explicitly set are maintained, and the structural information is completed using the struct file.

In this case, the number of parameters to be read is larger: lattice type; list of non-equivalent atoms, positions, and mutiplicities; cell parameters; NPT, R0 and RMT for each atom.

Contrarily to LOADCIF, the RMT, NPT and RO contained in the struct file are read. These values act as default, if the corresponding keyword does not appear in the general section. If you want to recover the WIEN2k value for any of this, use the AUTO value. The strategy of loading all the information from a struct file and then changing the cell parameters is heavily used in the sweep section (see below).

Keep in mind that crystallographic coordinates like 1/3, 1/6, and similar, should be given with a large precision to prevent ghost duplicacies of atoms when applying the symmetry operators.

Default: --.

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] The DO/EXCEPT pair is used to select a subset of calculations to run when more that one calculation is implied from the input data. The behavior of this orders is described in the GLOBAL section.

LDA+U environment Formed as

```
LDA+U {SIC|AMF|HMF}
    iat1.i {l1.i|l1.s} [defu.r defj.r]
    iat2.i {l2.i|l2.s} [defu.r defj.r]
    ...
END LDA+U
```

Includes an orbital dependent correction to the exchange-correlation potential. This method allows LDA (or GGA) calculations to successfully reproduce the properties of the so-called 'strongly correlated systems'. Common systems of this kind are transition metal oxides and rare earth elements and compounds. For more information, see the WIEN2k user's guide and references therein.

The first keyword in the environment selects the flavor of LDA+U:

- SIC: Self-Interaction Correction (Anisimov et al., 1993).
- AMF: Around the Mean Field (Czyzyk et al., 1994)
- HMF: Hubbard model in Mean Field (Anisimov et al., 1991).

Each line in the LDA+U environment activates the U correction for a particular atom and atomic symmetry (l). The first field corresponds to the non-equivalent atom index in the unit cell. The second field selects the angular momentum quantum number l, either by its value (0, 1, 2,..) or by the corresponding letter (s, p, d,...). The optional values defu.r and defj.r set the default values for U and J for the atom and l specified in the line.

The U (Coulomb on-site parameter) and J (exchange parameter) characterize the LDA+U correction. These parameters are treated in runwien through the keywords U and J, that can generate series of calculations in the same way as RMT, NPT, R0, ... See below for more information.

U [line.i] urange.r Coulomb parameter for LDA+U environment line line.i. Runwien accepts a range of U parameters in urange.r. Each of the U in the range generates a new structure, much in the same way as the other calculation parameters listed above, providing for an easy way of evaluating the effect of U on the system.

If no line.i is indicated, the first line is assumed. This syntax, however, forces the range urange.r to be a single field (i.e. without spaces or tabs). If line.i='*', the U values apply to all the lines in the LDA+U environment.

Default: defu.r, if it was set for line line.i in the LDA+U environment. Otherwise, 0.

J [line.i] jrange.r Exchange parameter for LDA+U environment line line.i. Runwien accepts a range of J parameters in jrange.r. Each of the J in the range generates a new structure, much in the same way as the other calculation parameters listed above, providing for an easy way of evaluating the effect of J on the system.

If no line.i is indicated, the first line is assumed. This syntax, however, forces the range jrange.r to be a single field (i.e. without spaces or tabs). If line.i='*', the J values apply to all the lines in the LDA+U environment.

Default: defj.r, if it was set for line line.i in the LDA+U environment. Otherwise, 0.

3.3.3 Requirements of the general section

The GENERAL section does not require previous sections to be run. In fact, this is, almost surely, the first section you will be running. It needs, however, some essential variables: lattice, cell parameters and non-equivalent list of atoms. This variables can be set either manually (via the correspondent keywords), reading a .struct, reading a .cif or loading a previous calculation.

3.4 Initialization section

3.4.1 Syntax

```
#runwien#
INITIALIZATION [DEFAULT]
```

```
    XCPOTENTIAL [LSDA | GGAPBE96 | GGAPW91 | GGAWC06]
    ECOREVAL core_energy.r
    ENERGYMIN min_energy.r
    ENERGYMAX max_energy.r
    NNFACTOR nn_dist.r
    ORBITALS neq_atom.i global_energy.r global_apw.i
      orb_l.i orb_energy.r orb_var.i {CONT|STOP} apw.i
      ...
    END ORBITALS
    LM LIST neq_atom.i
      l1 m1 l2 m2 l3 m3 ...
      lk mk lk+1 mk+1 ...
      ...
    END LM LIST
    LM LIST neq_atom.i lmax.i
    FERMI {ROOT | TEMP | TEMPS | GAUSS | TETRA | ALL} value.r
    DO [range.i | ALL | NONE | NEW | OLD]
    EXCEPT [range.i]
  END INITIALIZATION
```

Generates all the necessary input files for the calculation. Runs in sequence a host of programs related to symmetry, initial density generation,... see WIEN2k manual for more information about the process. Also, completes the struct file with the options required by the user.

Specifically, the programs run are setrmt_lapw (sets the rmt, if not given by the user), instgen_lapw, nn, sgroup, symmetry and lstart. Also, prepares .in1 .in2 and .inm files.

```
#runwien#
initialization default
```

Runs initialization assuming all default parameters. It is equivalent to:

```
#runwien#
initialization
end initialization
```

3.4.2 Keywords and environments

XCPOTENTIAL [LSDA | GGAPBE96 | GGAPW91 | GGAWC06] Chooses one of the available exchange-correlation potentials.

Default: ggapbe96

ECOREVAL core_energy.r Initial core-valence separation energy, in Ry. This energy value is compared to the results of the atomic calculation and core and valence states are thus assigned. This assignment does not vary during the scf cycle, unlike the band energies.

Default: -6.0

ENERGYMIN min_energy.r Lower energy cutoff for eigenvalue search (.in1 energy min).

Default: auto (wien2k determines which is best)

ENERGYMAX max_energy.r Higher energy cutoff for eigenvalue search (.in1 energy max).

Default: auto

NNFACTOR nn_dist.r Search for nearest neighbors (nn) in structure up to a distance of nnfactor * nn-distance.

Default: 2

ORBITALS environment The syntax of this environment corresponds to:

```
#runwien#
ORBITALS neq_atom.i global_energy.r global_apw.i
  orb_l.i orb_energy.r orb_var.i {CONT|STOP} apw.i
  ...
END ORBITALS
```

Orbital list, in the same format as accepted in .wien (but, of course, free format). <neq-atom> is the number of the non-equivalent atoms for which the basis is specified. global-energy is the default linearization energy and global-apw = 1 if apw method basis function is to be used; 0 if lapw. First <orb-l> appearing in the list refers to local orbitals + plane waves basis functions for that l. Subsequent orb-l will refer to local orbitals. orb-energy is the linearization energy; orb-var > 0 commands wien to search for a good linearization energy and if stop is set, stop in case it is not found. Finally, use <apw> = 1 in case you want an apw basis function type; 0 in case of lapw. See wien's manual for more information about basis.

Default: set by wien.

LM LIST environment The syntax of this environment corresponds to:

```
#runwien#
LM LIST neq_atom.i
  l1 m1 l2 m2 l3 m3 ...
  lk mk lk+1 mk+1 ...
  ...
END LM LIST
```

Specify LM lattice harmonics expansion of charge density. See wien's manual for L M pairs adequate for each structure.

Be careful with dimensioning parameters in WIEN2k code. By default, L > 6 require redimensioning LMAX2 in aim, dstart, lapw0 and lapw2. This redimensioning is required for some of the tests included in this distribution!

The dimensioning parameter for the maximum L value is higher (up to 10) in recent versions of WIEN2k, so redimensioning is not necessary.

If there are atoms of very low local symmetry (1, m, etc.) and the LM list is written up to LMAX = 10, it may happen that an additional parameter has to be redimensioned, depending on your choice of lnsmax. This parameter is NGAU, in the param.inc file of the lapw1 program, that controls the maximum number of gaunt coefficients calculated. The default value for NGAU is 2350. We have found a value of 2650 enough for an atom in local symmetry 1 with a LM list up to ten and lnsmax = 5.

Default: set by wien.

LM LIST neq_atom.i lmax.i Same as above, but the LM pairs are automatically generated by runwien.awk. This involves parsing the output of the sgroup program (outputsgroup), reading the local symmetry of each center, and generating the appropriate list of LMs up to a maximum L of lmax.i for the non-equivalent atom neq_atom.i.

This form of LM LIST can not be used in cases where two equivalent atoms in the unit cell are forced into non-equivalence (see, for instance, antiferromagnetic Cr, test023). In such cases, use the LM LIST environment above. The specific list of LM pairs for each local symmetry is shown below.

LM pairs up to LMAX = 10	
Loc sym.	LM list

1	0 0 1 0 1 1 -1 1 2 2 2 1 2 0 -2 1 -2 2 3 3 3 2 3 1 3 0 -3 1 -3 2 -3 3 4 4 4 3 4 2 4 1 4 0 -4 1 -4 2 -4 3 -4 4 5 5 5 4 5 3 5 2 5 1 5 0 -5 1 -5 2 -5 3 -5 4 -5 5 6 6 6 5 6 4 6 3 6 2 6 1 6 0 -6 1 -6 2 -6 3 -6 4 -6 5 -6 6 7 7 7 6 7 5 7 4 7 3 7 2 7 1 7 0 -7 1 -7 2 -7 3 -7 4 -7 5 -7 6 -7 7 8 8 8 7 8 6 8 5 8 4 8 3 8 2 8 1 8 0 -8 1 -8 2 -8 3 -8 4 -8 5 -8 6 -8 7 -8 8 9 9 9 8 9 7 9 6 9 5 9 4 9 3 9 2 9 1 9 0 -9 1 -9 2 -9 3 -9 4 -9 5 -9 6 -9 7 -9 8 -9 9 10 10 10 9 10 8 10 7 10 6 10 5 10 4 10 3 10 2 10 1 10 0 -10 1 -10 2 -10 3 -10 4 -10 5 -10 6 -10 7 -10 8 -10 9 -10 10
-1	0 0 -2 2 -2 1 2 0 2 1 2 2 -4 4 -4 3 -4 2 -4 1 4 0 4 1 4 2 4 3 4 4 -6 6 -6 5 -6 4 -6 3 -6 2 -6 1 6 0 6 1 6 2 6 3 6 4 6 5 6 6 -8 8 -8 7 -8 6 -8 5 -8 4 -8 3 -8 2 -8 1 8 0 8 1 8 2 8 3 8 4 8 5 8 6 8 7 8 8 10 10 10 9 10 8 10 7 10 6 10 5 10 4 10 3 10 2 10 1 10 0 -10 1 -10 2 -10 3 -10 4 -10 5 -10 6 -10 7 -10 8 -10 9 -10 10
2	0 0 1 0 2 0 2 2 -2 2 3 0 3 2 -3 2 4 0 4 2 -4 2 4 4 -4 4 5 0 5 2 -5 2 5 4 -5 4 6 0 6 2 -6 2 6 4 -6 4 6 6 -6 6 8 0 8 2 -8 2 8 4 -8 4 8 6 -8 6 8 8 -8 8 10 0 10 2 -10 2 10 4 -10 4 10 6 -10 6 10 8 -10 8 10 10 -10 10
m	0 0 1 1 -1 1 2 0 2 2 -2 2 3 1 -3 1 3 3 -3 3 4 0 4 2 -4 2 4 4 -4 4 5 1 -5 1 5 3 -5 3 5 5 -5 5 6 0 6 2 -6 2 6 4 -6 4 6 6 -6 6 7 1 -7 1 7 3 -7 3 7 5 -7 5 7 7 -7 7 8 0 8 2 -8 2 8 4 -8 4 8 6 -8 6 8 8 -8 8 9 1 -9 1 9 3 -9 3 9 5 -9 5 9 7 -9 7 9 9 -9 9 10 0 10 2 -10 2 10 4 -10 4 10 6 -10 6 10 8 -10 8 10 10 -10 10
2/m	0 0 -2 2 2 0 2 2 -4 4 -4 2 4 0 4 2 4 4 -6 6 -6 4 -6 2 6 0 6 2 6 4 6 6 -8 8 -8 6 -8 4 -8 2 8 0 8 2 8 4 8 6 8 8 -10 10 -10 8 -10 6 -10 4 -10 2 10 0 10 2 10 4 10 6 10 8 10 10
222	0 0 2 0 2 2 3 0 -3 2 4 0 4 2 4 4 5 0 -5 2 -5 4 6 0 6 2 6 4 6 6 7 0 -7 2 -7 4 -7 6 8 0 8 2 8 4 8 6 8 8 9 0 -9 2 -9 4 -9 6 -9 8 10 0 10 2 10 4 10 6 10 8 10 10
mm2	0 0 1 0 2 0 2 2 3 0 3 2 4 0 4 2 4 4 5 0 5 2 5 4 6 0 6 2 6 4 6 6 7 0 7 2 7 4 7 6 8 0 8 2 8 4 8 6 8 8 9 0 9 2 9 4 9 6 9 8 10 0 10 2 10 4 10 6 10 8 10 10
mmm	0 0 2 0 2 2 4 0 4 2 4 4 6 0 6 2 6 4 6 6 8 0 8 2 8 4 8 6 8 8 10 0 10 2 10 4 10 6 10 8 10 10
4	0 0 1 0 2 0 3 0 4 0 -4 4 4 4 5 0 -5 4 5 4 6 0 -6 4 6 4 7 0 -7 4 7 4 8 0 -8 4 8 4 -8 8 8 8 9 0 -9 4 9 4 -9 8 9 8 10 0 10 -4 10 4 10 -8 10 8
-4	0 0 2 0 3 2 -3 2 4 0 -4 4 4 4 5 2 -5 2 6 0 -6 4 6 4 7 2 -7 2 7 6 -7 6 8 0 -8 4 8 4 -8 8 8 8 9 2 -9 2 9 6 -9 6 10 0 -10 4 10 4 -10 8 10 8
4/m	0 0 2 0 4 0 -4 4 4 4 6 0 -6 4 6 4 8 0 -8 4 8 4 -8 8 8 8 10 0 -10 4 10 4 -10 8 10 8
422	0 0 1 0 2 0 3 0 4 0 4 4 5 0 -5 4 5 4 6 0 6 4 7 0 -7 4 7 4 8 0 8 4 8 8 9 0 -9 4 -9 8 10 0 10 4 10 8
4mm	0 0 1 0 2 0 3 0 4 0 4 4 5 0 5 4 6 0 6 4 7 0 7 4 8 0 8 4 8 8 9 0 9 4 9 8 10 0 10 4 10 8
-42m	0 0 2 0 -3 2 4 0 4 4 -5 2 6 0 6 4 -7 2 -7 6 8 0 8 4 8 8 -9 2 -9 6 10 0 10 4 10 8
-4m2	0 0 2 0 -3 2 4 0 4 4 -5 2 6 0 6 4 -7 2 -7 6 8 0 8 4 8 8 -9 2 -9 6 10 0 10 4 10 8
4/mmm	0 0 2 0 4 0 4 4 6 0 6 4 8 0 8 4 8 8 10 0 10 4 10 8

3	0 0 1 0 2 0 3 0 3 3 -3 3 4 0 4 3 -4 3 5 0 5 3 -5 3 6 0 6 3 -6 3 6 6 -6 6 7 0 7 3 -7 3 7 6 -7 6 8 0 8 3 -8 3 8 6 -8 6 9 0 9 3 -9 3 9 6 -9 6 9 9 -9 9 10 0 10 3 -10 3 10 6 -10 6 10 9 -10 9
-3	0 0 2 0 4 0 -4 3 4 3 6 0 -6 3 6 3 -6 6 6 6 8 0 -8 3 8 3 -8 6 8 6 10 0 -10 3 10 3 -10 6 10 6 -10 9 10 9
312	0 0 1 0 2 0 3 0 -3 3 4 0 4 3 5 0 -5 3 6 0 6 3 6 6 7 0 -7 3 -7 6 8 0 8 3 8 6 9 0 -9 3 -9 6 -9 9 10 0 10 3 10 6 10 9
321	0 0 1 0 2 0 3 0 -3 3 4 0 4 3 5 0 -5 3 6 0 6 3 6 6 7 0 -7 3 -7 6 8 0 8 3 8 6 9 0 -9 3 -9 6 -9 9 10 0 10 3 10 6 10 9
31m	0 0 1 0 2 0 3 0 3 3 4 0 4 3 5 0 5 3 6 0 6 3 6 6 7 0 7 3 7 6 8 0 8 3 8 6 9 0 9 3 9 6 9 9 10 0 10 3 10 6 10 9
3m1	0 0 1 0 2 0 3 0 3 3 4 0 4 3 5 0 5 3 6 0 6 3 6 6 7 0 7 3 7 6 8 0 8 3 8 6 9 0 9 3 9 6 9 9 10 0 10 3 10 6 10 9
-31m	0 0 2 0 4 0 4 3 6 0 6 3 6 6 8 0 8 3 8 6 10 0 10 3 10 6 10 9
-3m1	0 0 2 0 4 0 4 3 6 0 6 3 6 6 8 0 8 3 8 6 10 0 10 3 10 6 10 9
6	0 0 1 0 2 0 3 0 4 0 5 0 6 0 -6 6 6 6 7 0 -7 6 7 6 8 0 -8 6 8 6 9 0 -9 6 9 6 10 0 -10 6 10 6
-6	0 0 2 0 3 3 -3 3 4 0 5 3 -5 3 6 0 6 6 7 3 -7 3 8 0 8 6 9 0 9 3 -9 3 9 9 -9 9 10 0 10 6
6/m	0 0 2 0 4 0 6 0 -6 6 6 6 8 0 -8 6 8 6 10 0 -10 6 10 6
622	0 0 1 0 2 0 3 0 4 0 5 0 6 0 6 6 7 0 -7 6 8 0 8 6 9 0 -9 6 10 0 10 6
6mm	0 0 1 0 2 0 3 0 4 0 5 0 6 0 6 6 7 0 7 6 8 0 8 6 9 0 9 6 10 0 10 6
-6m2	0 0 2 0 3 3 4 0 5 3 6 0 6 6 7 3 8 0 8 6 9 3 9 9 10 0 10 6
-62m	0 0 2 0 3 3 4 0 5 3 6 0 6 6 7 3 8 0 8 6 9 3 9 9 10 0 10 6
6/mmm	0 0 2 0 4 0 6 0 6 6 8 0 8 8 10 0 10 8
23	0 0 4 0 4 4 6 0 6 4 -3 2 6 2 6 6 -7 2 -7 6 8 0 8 4 8 8 -9 2 -9 6 -9 4 -9 8 10 0 10 4 10 8 10 2 10 6 10 10
m-3	0 0 4 0 4 4 6 0 6 4 6 2 6 6 8 0 8 4 8 8 10 0 10 4 10 8 10 2 10 6 10 10
432	0 0 4 0 4 4 6 0 6 4 8 0 8 4 8 8 -9 4 -9 8 10 0 10 4 10 8
-43m	0 0 4 0 4 4 6 0 6 4 -3 2 -7 2 -7 6 8 0 8 4 8 8 -9 2 -9 6 10 0 10 4 10 8
m-3m	0 0 4 0 4 4 6 0 6 4 8 0 8 4 8 8 10 0 10 4 10 8

FERMI {ROOT | TEMP | TEMPS | GAUSS | TETRA | ALL} value.r Method to find fermi energy. See WIEN2k manual for details on each of the method. Value is a method-specific characteristic number.

Default: set by WIEN2k (tetra 0.000). In case no value is provided, use 0.000.

SGROUP Use the struct file proposed by sgroup. This struct file contains the optimum geometrical parameters (cell lengths, origin, etc.) for the calculation. However, sometimes the user may be interested in overriding this behavior. For instance, if he wants to calculate a system with non-equivalent atoms of the same type at symmetry equivalent positions (supercells, etc.). An example of this is bcc Cr, where both atoms in the unit cell are non-equivalent (allows antiferromagnetic ordering):

```
#runwien#
equiv list Cr 1
 0.0 0.0 0.0
end equiv list
```

```
equiv list Cr 2
 0.5 0.5 0.5
end equiv list
```

If sgroup is run on the struct file generated by this input, both atoms are made equivalent. Due to these reasons, copying the sgroup struct file is disabled by default.

Default: do not use it.

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] The DO/EXCEPT pair is used to select a subset of calculations to run when more that one calculation is implied from the input data. The behavior of this orders is described in the GLOBAL section.

Default: Use the local DO/EXCEPT pair if present; otherwise use the global DO/EXCEPT pair if present; if neither is defined calculate only new structures.

3.4.3 Requirements of the initialization section

Initialization can not be run unless general has been run or loaded.

3.5 Prescf section

3.5.1 Syntax

```
#runwien#
PRESCF
  KGENOUTPUT {LONG | SHORT}
  KGENSHIFT {YES | NO}
  NICE nice_level.i
  DO [range.i | ALL | NONE | NEW | OLD]
  EXCEPT [range.i]
END PRESCF
```

Runs programs which are the last step before scf calculation. These include kgen (generates plane waves) and dstart (initial density).

In kgen, inversion is always added to the k-point list if asked.

```
#runwien#
prescf default
```

Runs prescf section assuming default values for all variables.

3.5.2 Keywords and environments

KGENOUTPUT {LONG | SHORT} Select the printing level of kgen output. This is asked for only in older versions of WIEN2k (< 20080421)

Default: short

KGENSHIFT {YES | NO} If it is allowed, shift k-point mesh away from high symmetry directions.

Default: yes

NICE nice_level.i Process priority for dstart (can take quite long). Child processes can not have a lower nice than runwien itself.

Default: 0 (increase to decrease priority).

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] The DO/EXCEPT pair is used to select a subset of calculations to run when more that one calculation is implied from the input data. The behavior of this orders is described in the GLOBAL section.

Default: Use the local DO/EXCEPT pair if present; otherwise apply the global DO/EXCEPT pair if present; if neither is defined calculate only new structures.

3.5.3 Requirements of the prescf section

General and initialization sections are required for prescf to run correctly.

3.6 Scf section

3.6.1 Syntax

```
#runwien#
SCF [DEFAULT]
   MAX ITERATIONS maxit.i
   CHARGE CONV cconv.r
   ENERGY CONV econv.r
   FORCE CONV fconv.r
   NICE nice_value.i
   REUSE {CHAIN|FIXED n.i|FIRST|ANY|PATH path.s}
   ITDIAG [cyc.i]
   NEW IN1 cyc.i
   MINI [commandline.s]
   NOSUMMARY
   DO [range.i | ALL | NONE | NEW | OLD]
   EXCEPT [range.i]
END SCF
```

Runs the scf cycle. By default, a summary will be generated in the root directory, called scfsummary.out, containing some relevant information about the scf process.

Notice that the simple

```
#runwien#
scf default
```

will invoke scf assuming default values on all variables.

3.6.2 Keywords and environments

MAX ITERATIONS maxit.i Maximum number of cycles.

Default: 30

CHARGE CONV cconv.r Charge convergence criterion.

Default: none

ENERGY CONV econv.r Energy convergence criterion.

Default: 0.00001, unless one of cc, ec or fc is indicated.

FORCE CONV fconv.r Force convergence criterion.

Note that if any of energy conv, force conv or charge conv are set, they constitute the scf stop criterion. If **none** of them are set, then the default value is energy conv, 0.00001 Ry.

Default: none

NICE nice_value.i Process priority for run_lapw. Child processes can not have a lower nice than runwien itself.

Default: 0 (increase to decrease priority).

REUSE {CHAIN|FIXED n.i|FIRST|ANY|PATH path.s} Use the converged density of a previously calculated structure as starting point, replacing the superposition density generated by dstart.

There are 5 modes of REUSE. In CHAIN, each general section structure tries to reuse the converged density of the preceding structure. In FIXED, one given structure (number n.i) is selected and consistently used as starting point. FIRST is equivalent to FIXED 1, i. e., use always the density of the first structure. If ANY is used, runwien tries to find a directory among the calculated structures that contains a valid clmsum and struct files. Finally, a path may be given directly with the keyword PATH, pointing to the location of a calculation.

With independence of the method employed in determining the path containing the initial density, runwien has three levels of modification of the clmsum file. If the struct file of the source calculation differs in any structural parameter (cell parameters or atomic positions) from the actual calculation, a extrapolation is carried out using the clmaddsub utility (only available in recent versions of WIEN2k). This process involves subtracting the old superposition density (stored in the .clmsum.atomic file) and adding the new one.

Second, if the structural parameters are the same but the struct files of both calculations differ in npt, rmt or r0 for any atom, a interpolation of the radial grid is performed, using the WIEN2k's clminter program.

Finally, if the struct files contain the same information, the clmsum file is simply copied, and effectively replaces the superposition density as starting point for the SCF cycle.

In older versions of runwien (<= 14.12.2007), the extrapolation of the charge density is substituted by a mere renormalization.

It is interesting to note that REUSE may lead to some drastic shortening of the computational cost of your calculations, specially if the old densities correspond correspond to a good set of parameters. Therefore, routine usage of this keyword should be considered.

Default: no reuse.

ITDIAG [cyc.i] Use iterative diagonalization and do a complete diagonalization each cyc.i cycles. In versions of WIEN2k >= 13.08.2007, cyc.i has no effect.

Analogously to REUSE, ITDIAG may impose dramatic cuts in the computational cost of the calculation.

Default: no it. diag. cyc.i = 3, if applicable.

NEW IN1 cyc.i Generate a new basis set in cycle <cyc>.

Default: no

MINI [commandline.s] Perform an internal parameter optimization using the mini program (through the min_lapw script). The commandline.s optional field is appended to min_lapw in the calling command (min_lapw <commandline.s>). If no commandline.s is found, -j 'run*_lapw scfcmdline.s' is used, where scfcmdline.s are the arguments for the run*_lapw script. This keyword is provided as a temporary solution. It will be improved in later versions.

Default: no internal optimization.

NOSUMMARY Don't run the summarizing script. This script will generate a scfsummary.out file in root directory, containing information about the results of the scf process.

Default: false

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] The DO/EXCEPT pair is used to select a subset of calculations to run when more that one calculation is implied from the input data. The behavior of this orders is described in the GLOBAL section.

Default: Use the local DO/EXCEPT pair if present; otherwise apply the global DO/EXCEPT pair if present; if neither is defined calculate only new structures.

The DO and EXCEPT orders are frequently used the SCF section: for instance to repeat the costly SCF process on a particular geometry that has failed to converge, while keeping intact the geometries already converged. As a matter of fact, WIEN2k internal orders, like run_lapw, can be run manually if necessary.

Default: if no do or except was specified in scf and global, calculate only new structures. If there is a global do or except, use it.

3.6.3 Requirements of the scf section

General, initialization and prescf sections are required.

3.7 Spinorbit section

3.7.1 Syntax

```
#runwien#
SPINORBIT [DEFAULT]
   DIRECTION h.i k.i l.i
   ADDLO at.i [el.r de.r]
   EXCLUDE at1.i at2.i ...
   NEWEMAX emax.r
   NEWKPTS kpts.r
   MAX ITERATIONS maxit.i
   CHARGE CONV cconv.r
   ENERGY CONV econv.r
   FORCE CONV fconv.r
   NICE nice_value.i
   ITDIAG [cyc.i]
   NEW IN1 cyc.i
   DO [range.i | ALL | NONE | NEW | OLD]
   EXCEPT [range.i]
END SPINORBIT
```

Re-runs the scf cycle including the spin-orbit interaction through a second-variational method (see WIEN2k user's guide for details). This section is run right after the scf cycle, using the converged orbitals as the starting point for the new scf.

The command

```
#runwien#
spinorbit default
```

will invoke spinorbit assuming default values on all variables.

In spinpolarized cases, a reduction in the symmetry of the system may occur. Note that the new struct file generated by symmetso is adopted, but it is not parsed, so runwien will still use the old structural information. Also, in cases where LDA+U is being used, the .inorb (input for the orb program) and .indm (input for the lapwdm program) are *not* adapted. This shortcomings may be fixed in later versions of runwien.

3.7.2 Keywords and environments

DIRECTION h.i k.i l.i Chooses the direction of magnetization.

Default: 0 0 1

ADDLO at.i [el.r de.r] Add an LO with a p1/2 radial function to the non-equivalent atom at.i, with energy el.r. If de.r is not zero, the energy is automatically adjusted, with steps de.r.

Default: el.r = -4.97, de.r = 0.005

EXCLUDE at1.i at2.i ... Exclude the spin-orbit correction for the non-equivalent atoms at1.i, at2.i,...

Default: --.

NEWEMAX emax.r This keyword allows the user to change the upper limit of the eigenvalue window calculated in lapw1 by writing a new emax.r in the .in1(c) file.

Default: --.

NEWKPTS {kpts.r|kpts.r%} In spinpolarized cases, where symmetry may be reduced, regenerate the k-list with a number of k-points equal to kpts.r in the whole 1BZ. If the value provided contains the % character, the kpts.r value will be interpreted as a percentage of the original kpts for each of the structures run in the spinorbit section.

Default: 100% (same number of kpts as in the original).

MAX ITERATIONS maxit.i Maximum number of cycles.

Default: the value provided by the user in the SCF section, otherwise 30.

CHARGE CONV cconv.r Charge convergence criterion.

Default: none.

ENERGY CONV econv.r Energy convergence criterion.

Default: none.

FORCE CONV fconv.r Force convergence criterion.

The default convergence criterion is the same as in the SCF section, or energy conv 0.00001 if it was not set.

Default: none

NICE nice_value.i Process priority for run_lapw. Child processes can not have a lower nice than runwien itself.

Default: 0 (increase to decrease priority).

ITDIAG [cyc.i] Use iterative diagonalization and do a complete diagonalization each cyc.i cycles. In versions of WIEN2k >= 13.08.2007, cyc.i has no effect.

Default: no it. diag. cyc.i = 3, if applicable.

NEW IN1 cyc.i Generate a new basis set in cycle <cyc>.

Default: --.

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] The DO/EXCEPT pair is used to select a subset of calculations to run when more that one calculation is implied from the input data. The behavior of this orders is described in the GLOBAL section.

Default: Use the local DO/EXCEPT pair if present; otherwise apply the global DO/EXCEPT pair if present; if neither is defined calculate only new structures.

3.7.3 Requirements of the spinorbit section

General, initialization, prescf and scf sections are required.

3.8 Free section

3.8.1 Syntax

```
#runwien#
FREE
  ATOM atname.s
  GLOBAL
  REFERENCE STRUCT n.i
  CELL a.r
  SPINPOLARIZED {YES | NO}
  NPT r.r
  RMT r.r
  R0 r.r
  RKMAX rkmax.r
  LMAX lmax.i
  LNSMAX lnsmax.i
  GMAX gmax.r
  MIX lambda.r
  XCPOTENTIAL {LSDA | GGAPBE96 | GGAPW91 | GGAWC06}
  ECOREVAL ecore.e
  ENERGYMIN emin.r
  ENERGYMAX emax.r
  ORBITALS neq_atom.i global_energy.r global_apw.r
    orb_l.i orb_energy.r orb_var.i {CONT | STOP} apw.i
    ...
  END ORBITALS
  LM LIST neq_atom.i
    l1 m1 l2 m2 l3 m3 ...
    lk mk lk+1 mk+1 ...
    ...
  END LM LIST
  FERMI {ROOT | TEMP | TEMPS | GAUSS | TETRA | ALL} value.r
  MAX ITERATIONS maxit.i
  CHARGE CONV cconv.r
  ENERGY CONV econv.r
  FORCE CONV fconv.r
  ITDIAG [cyc.i]
  NEW IN1 cyc.i
  CLEAN
  NOSEND
  NICE nice.i
  DO [ALL | NONE | NEW | OLD] [atname1.s [atname2.s ...]]
  EXCEPT [atname1.s [atname2.s ...]]
END FREE
```

The calculation of the free in-vacuo energy of an atom using WIEN2k is a delicate matter. Following the recommendations of the WIEN2k mailing list, this section tries to obtain a value for the in-vacuo energy of all the atoms composing a given reference system (one of the general structures) by calculating a fcc structure with only one atom type, only 1 k-point and very large cell parameter. The atomic in-vacuo (free) energies are then used to calculate the total in-vacuo energy of the system. This “free energy” may be used in the gibbs section, as reference structure and to calculate the dissociation energy of the crystal.

The calculations required for this task are stored in <root>-free-<at> where <root> is the root for the calculation and <at> the atom name.

```
#runwien#
free default
```

Runs free section assuming default values for all variables.

3.8.2 Keywords and environments

ATOM atname.s Atom to which the following keywords apply. Free section calculates the in-vacuo energy of every atom type in the molecular formula. However, this task can be worked out with different calculation levels for different atoms. With atom <atname>, you indicate that the following lines apply only to the atom whose symbol is <atname>, until runwien finds another atom keyword or a global keyword (see below).

Orders appearing before any ATOM specification (i.e. right after the free keyword) will apply to all the atom types.

Default: --.

GLOBAL After this keyword is read, all the input refers to all the atom types.

For example, if you are calculating CaTiO3 and wish to set up a cell parameter of 20.0 for Ca, 21.0 for Ti and 22.0 for O, you can write:

```
#runwien#
cell 20.0
atom Ti
cell 21.0
atom O
cell 22.0
```

Default: --.

REFERENCE STRUCT n.i Use the <n>th general structure to fill up the variables which were not set in the input file.

Default: the first general structure which was run through the general section.

CELL a.r Cell parameter for the fcc conventional cell used in the free energy calculation. Keep this value **high**, but not too much. Values between 20.0 and 25.0 are just right. See WIEN2k mailing list for reference.

Default: 25.0

SPINPOLARIZED {YES | NO} Perform a spinpolarized or non-spinpolarized calculation for the selected atom.

Default: reference structure.

NPT r.r Number of radial mesh points. Must be an odd number. Wien2k manual recommends 381 for lda calculations, 781 for gga. Please, note that the argument for npt is *not* a list of values, i.e., npt is fixed throughout all the sweep run. "auto" can be used and it will be passed into local .wien files.

Default: reference structure.

RMT r.r Muffin tin radius, in bohr. Note that this rmt does not require the non-equivalent atom: the system only has one atom!

Default: reference structure.

R0 r.r First radial mesh point, in bohr. Wien2k manual recommends a value between 0.0005 and 0.00005.

Default: reference structure.

RKMAX rkmax.r rmt * Kmax, where Kmax is the maximum k for plane waves and rmt is the smallest muffin tin radius.

Default: reference structure.

LMAX lmax.i Maximum l value for spherical harmonics expansion inside muffin tins. Required in initialization, same as above.

Default: reference structure.

LNSMAX lnsmax.i Maximum l value for spherical harmonics expansion used to computate matrix elements with non-muffin tin functions. Required in initialization, same as above.

Default: reference structure.

GMAX gmax.r Maximum reciprocal vector for fourier expansion of electron density. Wien2k manual recommends using a large gmax (25) for systems with short H bonds. Also, a larger gmax for gga calculations (14) than for lsda potentials. Required in initialization, same as above.

Default: reference structure.

MIX lambda.r Mixing factor used in the creation of the new charge density.

Default: reference structure.

XCPOTENTIAL {LSDA | GGAPBE96 | GGAPW91 | GGAWC06} Chooses one of the available exchange-correlation potentials.

Default: reference structure.

ECOREVAL ecore.e Initial core-valence separation energy, in Ry.

Default: reference structure.

ENERGYMIN emin.r Lower energy cutoff for eigenvalue search.

Default: reference structure.

ENERGYMAX emax.r Higher energy cutoff for eigenvalue search.

Default: reference structure.

ORBITALS environment The syntax of this environment corresponds to:

```
#runwien#
ORBITALS neq_atom.i global_energy.r global_apw.i
  orb_l.i orb_energy.r orb_var.i {CONT|STOP} apw.i
  ...
END ORBITALS
```

Orbital list, in the same format as accepted in .wien (but, of course, free format). <neq-atom> is the number of the non-equivalent atoms for which the basis is specified. global-energy is the default linearization energy and global-apw = 1 if apw method basis function is to be used; 0 if lapw. First <orb-l> appearing in the list refers to local orbitals + plane waves basis functions of that l. Subsequent orb-l will refer to local orbitals. orb-energy is the linearization energy; orb-var > 0 commands wien to search for a good linearization energy and if stop is set, stop in case it is not found. Finally, use <apw> = 1 in case you want an apw basis function type; 0 in case of lapw. See WIEN2k's manual for more information about basis.

Default: reference structure.

LM list The syntax of this environment corresponds to:

```
#runwien#
LM LIST neq_atom.i
  l1 m1 l2 m2 l3 m3 ...
  lk mk lk+1 mk+1 ...
  ...
END LM LIST
```

Specify LM lattice harmonics expansion of charge density. See wien's manual for L M pairs adequate for each structure.

Default: reference structure.

FERMI {ROOT | TEMP | TEMPS | GAUSS | TETRA | ALL} value.r Method to find fermi energy. See WIEN2k's manual for details on each of the method. Value is a method-specific characteristic number.

Default: reference structure.

MAX ITERATIONS maxit.i Maximum number of cycles.

Default: reference structure.

CHARGE CONV cconv.r Charge convergence criterion.

Default: reference structure.

ENERGY CONV econv.r Energy convergence criterion.

Default: reference structure.

FORCE CONV fconv.r Force convergence criterion.

Note that reference values are only used if all three criteria are not given.

Default: reference structure.

ITDIAG [cyc.i] Use iterative diagonalization and do a complete diagonalization each cyc.i cycles. In versions of WIEN2k >= 13.08.2007, cyc.i has no effect.

Default: --. cyc.i = 3.

NEW IN1 cyc.i Generate a new basis set in cycle <cyc>.

Default: --.

CLEAN Append this clean line to the wien input file for the selected atom.

Default: --.

NOSEND Create the wien input file but do not run the atom for which this keyword applies (atom / global mechanism).

Default: --.

NICE nice.i Process priority for every section of local .wien. Note that runwien child processes can not have a lower nice than runwien itself.

Default: --.

DO [ALL | NONE | NEW | OLD] [atname1.s [atname2.s ...]] Used together with the EXCEPT order below.

EXCEPT [atname1.s [atname2.s ...]] As a default, the FREE section will obtain the in vacuo energy for the atoms in the crystal molecular formula that have not been calculated before. The DO/EXCEPT orders can be used to modify this behavior. Notice that the syntax of the order in this section differs slightly from their counterpart in other sections. Here, a list of atom name substitutes the range of integer indices. On the other hand, the FREE section is not affected by the general DO/EXCEPT, only by the local options.

Default: calculate only the new atoms. The global DO/EXCEPT pair has no effect.

3.8.3 Requirements of the free section

General, initialization and prescf sections are required.

3.9 Elastic section

3.9.1 Syntax

```
#runwien#
ELASTIC [DEFAULT]
  REFERENCE STRUCT n.i
  DEFORMATION [POINTS n.i] [MAXLENGTH mlenght.r] [MAXANGLE mangle.r]
  POLYORDER ord.i
  FIXMIN {YES | NO}
  TERM1 {YES | NO}
  NPT [atom.i] n.i
  RMT [atom.i] radius.r
  R0 [atom.i] r.r
  RKMAX r.r
  LMAX l.i
  LNSMAX l.i
  GMAX g.r
  MIX lambda.r
  KPTS k.r
  LDA+U {SIC|AMF|HMF}
   iat1.i {l1.i|l1.s} u.r j.r
   iat2.i {l2.i|l2.s} u.r j.r
   ...
  END LDA+U
  XCPOTENTIAL [LSDA | GGAPBE96 | GGAPW91 | GGAWC06]
  ECOREVAL ecore.r
  ENERGYMIN emin.r
  ENERGYMAX emax.r
  ORBITALS neq_atom.i global_energy.r global_apw.r
    orb_l.i orb_energy.r orb_var.i {CONT|STOP} apw.i
    ...
  END ORBITALS
  LM LIST neq_atom.i
    l1 m1  l2 m2  l3 m3 ...
    lk mk  lk+1 mk+1 ...
    ...
  END LM LIST
  FERMI {ROOT | TEMP | TEMPS | GAUSS | TETRA | ALL} value.r
```

```
    MAX ITERATIONS miter.i
    CHARGE CONV cconv.r
    ENERGY CONV econv.r
    FORCE CONV fconv.r
    ITDIAG [cyc.i]
    NEW IN1 cyc.i
    NICE nice.i
    REUSE {CHAIN|REFERENCE|PATH|DETECT}
    NOREUSE
    MINI [commandline.s]
    SPINORBIT
     ...
    END SPINORBIT
    CLEAN [WIEN | FULL]
    NOSEND
  END ELASTIC
```

The elastic sections allows the user to calculate the elastic constants of a general structure. This section expands the capabilities of WIEN2k by providing a means for calculating the elastic constants of cubic and hexagonal systems . The calculations run automatically but it is also possible to do the deformation points manually and load the results afterwards.

All the elastic section work is done in the <root>-elastic directory, where <root> is the root of the calculation. In this directory there is a subdirectory for each deformation: 3 for cubic systems (volume, tetragonal and rhombohedral deformations) and 5 for hexagonal systems (volume, a and b, c, alpha and beta, and gamma deformations). These deformations are annotated by a six-digit code, where a 0 represents the original cell parameter and a 1 a varying cell parameter. Thus, for example, 111000 represents the volume deformation while 000001 means that gamma is the only cell parameter that is varied.

Each deformation directory contains several subdirectories, each corresponding to one point in the deformation. Runwien avoids having a different symmetry for the unstrained point by adding it up a tiny deformation so as to give it the same symmetry as the rest. Note that **the struct file generated by sgroup is used in the deformation points**. This is planned to change in future versions.

Once all the energies of all the deformed geometries are known, runwien proceeds to calculate the elastic constants. First, the finite lagrangian strain tensor is calculated from the cell parameter deformations. Then, gnuplot scripts (with name fit-xxxxxx.gnuplot) are used to fit the data and extract the second derivative of the energy. Finally, this information is used to calculate the elastic constants and evaluate the Born stability conditions. The energy information and exact procedure and formulas used are written to the file <root>-elastic/elasticsummary.out.

The LOADCHECK REREAD option is useful in this context. Using this keyword, the elastic calculations can be manually and the results read in into runwien.awk. The elastic summary, fit scripts, graphs and elastic constants are regenerated. There is, however, one exception to this: every file with the name of a target fit script (fit-xxxxxx.gnuplot, where xxxxxx is the deformation identifier) will be recognized and used for the fitting, instead of being overwritten.

The results of the elastic section are printed in a synopsis run. The scripts and data employed to generate the elastic constants and some guidance about the process are stored in <root>-elastic.

The DO/EXCEPT mechanism is not implemented in elastic section.

For the elastic section to proceed to the calculation of the elastic constants, all the energies for all the deformations must be known.

The easiest way of calling this section is accepting all default conventions and parameters

```
#runwien#
elastic default
```

Note that an accurate determination of the elastic constants requires the inner deformation of atomic positions to be taken into account. This 'inner strain' can be calculated by using the mini optimization program, through the mini keyword of runwien.awk. This option, however, is a temporary solution and a further development of this feature in runwien.awk is planned. Should this keyword fail to calculate the inner strain, you can fall back to running mini manually and rereading the results in runwien.awk.

3.9.2 Keywords and environments

REFERENCE STRUCT n.i General section's reference structure for calculations. This structure provides default values in case they are not user-provided.

Default: the first which was run through general section.

DEFORMATION [POINTS n.i] [MAXLENGTH mlenght.r] [MAXANGLE mangle.r] Selects the number of points calculated in each deformation, the maximum length spanned in a, b and c cell parameter's deformations and the max angle spanned in alpha, beta, gamma cell parameter's deformations. There is nothing wrong about an even number of points, but it seems more sensible to choose and odd number, for you are interested in the second derivative at zero strain. The mlength parameter is given as a fraction of the original cell length. For example, if mlength = 0.05, the deformation in the a parameter will comprise the interval [a*(1-mlength), a*(1+mlength)]. The mangle value is the maximum distance the deformation reaches from the original angle cell parameter, in sexagesimal degrees. For example, if mangle = 5, the deformed alpha interval for a cubic system will be [alpha-5, alpha+5].

If an odd number of points is used, the central point will be slightly displaced from the zero strain position, to ensure that all points are calculated with the same, low, symmetry. This displacement is 0.001 bohr in distances and 0.005 degrees in angles.

The POINTS, MAXLENGHT and MAXANGLE keywords may exchange positions.

Default: 7 points, 0.05 in maxlength and 5.0 in maxangle.

POLYORDER ord.i Order of the fitting polynomial. The only valid orders are 2, 4, 6 and 8.

Default: order 4.

FIXMIN {YES | NO} If the user activates the fixmin (fixmin yes), then the point around which the polynomial is expanded is fixed to the zero-strain value (0.0 in general, -0.5 for the hexagonal monoclinic deformation). Otherwise, the equilibrium point is optimized in the fitting process.

Default: no.

TERM1 {YES | NO} The term1 keyword instructs runwien to include in the polynomial the 1st grade term (a1 * (x-x0)). Including this term is equivalent to the variation of x0 allowed by FIXMIN NO. If term1 is selected, then FIXMIN is forcibly set to YES.

Default: no.

NPT [atom.i] n.i Number of radial mesh points for non-equivalent atom with index atom.i. Must be an odd number. Wien2k manual recommends 381 for lda calculations, 781 for gga. Please, note that the argument for npt is *not* a list of values, i.e., npt is fixed throughout all the sweep run. “auto” can be used and it will be passed into local .wien files.

Auto keyword will make runwien accept runwien's default value. Npt value will not change from one structure to the other, but, generally speaking, auto keyword should be avoided in elastic section, as all the points must have the same calculation level.

All this comments are valid for general section's varying parameters in sweep section (rmt, r0, ...).

Default: reference structure.

RMT [atom.i] radius.r Muffin tin radius, in bohr, for non-equivalent atom with index atom.i.

Default: reference structure, if no atom is indicated, 1 is assumed.

R0 [atom.i] r.r First radial mesh point, in bohr, for non-equivalent atom with index atom.i. WIEN2k manual recommends a value between 0.0005 and 0.00005.

Default: reference structure.

RKMAX r.r rmt * Kmax, where Kmax is the maximum k for plane waves and rmt is the smallest muffin tin radius.

Default: reference structure.

LMAX l.i Maximum l value for spherical harmonics expansion inside muffin tins. Required in initialization, same as above.

Default: reference structure.

LNSMAX l.i Maximum l value for spherical harmonics expansion used to computate matrix elements with non-muffin tin functions. Required in initialization, same as above.

Default: reference structure.

GMAX g.r Maximum reciprocal vector for fourier expansion of electron density. Wien2k manual recommends using a large gmax (25) for systems with short H bonds. Also, a larger gmax for gga calculations (14) than for lsda potentials. Required in initialization, same as above.

Default: reference structure.

MIX lambda.r Mixing factor used in the creation of the new charge density.

Default: reference structure.

KPTS k.r Number of k-points in the whole first brillouin zone.

Default: reference structure.

LDA+U environment Formed as

```
LDA+U {SIC|AMF|HMF}
      iat1.i {l1.i|l1.s} u.r j.r
      iat2.i {l2.i|l2.s} u.r j.r
      ...
END LDA+U
```

Includes an orbital dependent correction to the exchange-correlation potential. This method allows LDA (or GGA) calculations to successfully reproduce the properties of the so-called 'strongly correlated systems'. Common systems of this kind are transition metal oxides and rare earth elements and compounds. For more information, see the WIEN2k user's guide and references therein.

The first keyword in the environment selects the flavor of LDA+U:

- SIC: Self-Interaction Correction (Anisimov et al., 1993).
- AMF: Around the Mean Field (Czyzyk et al., 1994)
- HMF: Hubbard model in Mean Field (Anisimov et al., 1991).

Each line in the LDA+U environment activates the U correction for a particular atom and atomic symmetry (l). The first field corresponds to the non-equivalent atom index in the unit cell. The second field selects the angular momentum quantum number l, either by its value (0, 1, 2,..) or by the corresponding letter (s, p, d,...).

The U (Coulomb on-site parameter) and J (exchange parameter) characterize the LDA+U correction. These parameters are set using the u.r and j.r values respectively.

Default: reference structure.

XCPOTENTIAL [LSDA | GGAPBE96 | GGAPW91 | GGAWC06] Chooses one of the available exchange-correlation potentials.

Default: reference structure.

ECOREVAL ecore.r Initial core-valence separation energy, in Ry.

Default: reference structure.

ENERGYMIN emin.r Lower energy cutoff for eigenvalue search.

Default: reference structure.

ENERGYMAX emax.r Higher energy cutoff for eigenvalue search.

Default: reference structure.

ORBITALS environment The syntax of this environment corresponds to:

```
#runwien#
ORBITALS neq_atom.i global_energy.r global_apw.i
  orb_l.i orb_energy.r orb_var.i {CONT|STOP} apw.i
  ...
END ORBITALS
```

Orbital list, in the same format as accepted in .wien (but, of course, free format). <neq_atom> is the number of the non-equivalent atoms for which the basis is specified. <global_energy> is the default linearization energy and <global_apw> = 1 if apw method basis function is to be used; 0 if lapw. First <orb_l> appearing in the list refers to local orbitals + plane waves basis functions of that l. Subsequent <orb_l> will refer to local orbitals. <orb_energy> is the linearization energy; <orb_var> > 0 commands wien to search for a good linearization energy and if stop is set, stop in case it is not found. Finally, use <apw> = 1 in case you want an APW basis function type; 0 for a LAPW function. See the WIEN2k manual for more information about bases.

Default: reference structure.

LM LIST environment The syntax of this environment corresponds to:

```
#runwien#
LM LIST neq_atom.i
  l1 m1 l2 m2 l3 m3 ...
  lk mk lk+1 mk+1 ...
  ...
END LM LIST
```

Specify LM lattice harmonics expansion of charge density. See wien's manual for L M pairs adequate for each structure.

Default: reference structure.

FERMI {ROOT | TEMP | TEMPS | GAUSS | TETRA | ALL} value.r Method to find fermi energy. See WIEN2k's manual for details on each of the method. <Value> is a method-specific characteristic number.

Default: reference structure.

MAX ITERATIONS miter.i Maximum number of cycles.

Default: reference structure.

CHARGE CONV cconv.r Charge convergence criterion.

Default: reference structure.

ENERGY CONV econv.r Energy convergence criterion.

Default: reference structure.

FORCE CONV fconv.r Force convergence criterion. Note that the default values only will be used if all three criteria are undefined.

Default: reference structure.

ITDIAG [cyc.i] Use iterative diagonalization and do a complete diagonalization each cyc.i cycles. In versions of WIEN2k >= 13.08.2007, cyc.i has no effect.

Default: --. cyc.i = 3

NEW IN1 cyc.i Generate a new basis set in cycle <cyc>.

Default: --.

NICE nice.i Process priority for every section of local .wien.

Runwien child processes can not have a lower nice than runwien itself.

Default: reference structure's scf_nice.

REUSE {CHAIN|REFERENCE|PATH path.s|DETECT} The REUSE keyword allows the user to control the reutilization of converged densities from one deformation point to the other in a elastic calculation. The CHAIN keyword makes each deformation use the converged density of the previous one. The first structure of each deformation type is calculated starting from the superposition of atomic densities.

With REFERENCE, each structure extrapolates the charge density of the selected reference structure of the general section. In a way similar to SCF section, PATH allows the user to use consistently a given calculation as starting point from every strained structure. Finally, if DETECT is read, runwien searches for existing clmsum and struct files in the directories of the elastic section, and uses the density files if found.

Default: DETECT.

NOREUSE Reusing the converged densities can lead to large time savings. However, problems can arise, specially in the deformations involving lowering of symmetry. The NOREUSE keyword allows the user to avoid the reutilization of densities of previous points completely.

Default: --.

MINI [commandline.s] Perform an internal parameter optimization using the mini program (through the min_lapw script). The mini line is passed down to all the wien input files in the elastic calculation. This is a temporary solution to the problem of including the inner strain in the elastic constant calculation.

Default: no internal optimization.

SPINORBIT environment The syntax of this environment corresponds to:

```
#runwien#
SPINORBIT
  ...
END SPINORBIT
```

Add a spinorbit section in wien input file to all the structures in the elastic calculation

The lines included in the body of the environment are passed verbatim to the corresponding wien input files.

The information in this environment (and the following) is not parsed: runwien will not be able to tell you if it is right or wrong until it is calculating the elastic structures.

CLEAN [WIEN | FULL] Append a 'clean [wien|full]' at the end of each deformation point .wien. This will make each point in elastic section clean itself after being calculated. For details on clean keyword, see above (global).

NOSEND Create the wien input files but do not run the calculation of the elastic constants. Useful if you plan to do it manually and then reading it back with a LOADCHECK REREAD.

Default: --.

3.9.3 Requirements of the elastic section

General, initialization and prescf sections are required.

3.10 Printrho section

3.10.1 Syntax

```
#runwien#
PRINTRHO
  RHO {TOTALRHO|VALRHO|SPIN|ATOMIC|DEFORM|VTOTAL|VCOUL|VXC}
  ENERGYMIN emin.r
  ORIGIN x.r y.r z.r den.r
  XEND x.r y.r z.r den.r
  YEND x.r y.r z.r den.r
  NSHELLS nx.i ny.i nz.i
  NPT nx.i ny.i
  ZMIN zmin.r
  ZMAX zmax.r
  DC dc.i
  TYPE {3D | C}
  SCALE {NORMAL | LOG}
  NOLABELS
  DO [range.i | ALL | NONE | NEW | OLD]
  EXCEPT [range.i]
END PRINTRHO
```

Using a general structure, prints its electron density, valence electron density, deformation density, spin density, atomic (promolecular density), coulomb potential, exchange-correlation potential and total potential in an arbitrary plane. This section extends WIEN2k's default utilities by providing a way to print planes defined by non-orthogonal axis. Additionally, the scripts employed are saved as .gnuplot scripts, so the user can modify them to fit his/her particular needs.

The gnuplot script is directly applicable to the data contained in the structure subdirectory.

The resulting .ps files are placed in the structure's subdirectory.

```
#runwien#
printrho default
```

Same as above but assuming defaults on all variables.

3.10.2 Keywords and environments

RHO {TOTALRHO|VALRHO|SPIN|ATOMIC|DEFORM|VTOTAL|VCOUL|VXC} Type of potential/-density plot. Totalrho means total (valence + core) density. Valrho, valence density. Spin, spin density, only allowed in spinpolarized calculations. Atomic, density as superposition of atomic densities calculated in initialization. Deform, deformation density. Vtotal, total potential. Vcoul, coulomb potential. Vxc, exchange-correlation potential.

Default: total density plot.

ENERGYMIN emin.r Lower energy cutoff for the processing of electron density, in Ry. Choose ecoreval (see above) in order to calculate only valence densities.

Default: total density (energymin value in .in2 file)

ORIGIN x.r y.r z.r den.r Origin of the plane. x, y and z must be integers. den is the denominator. The final point will be x/den, y/den, z/den. Crystallographic units.

Default: 0 0 0 1

XEND x.r y.r z.r den.r X end of the plane. Same syntax as above. Crystallographic units.

Default: 1 0 0 1

YEND x.r y.r z.r den.r Y end of the plane. Same syntax as above. Crystallographic units.

Default: 0 1 0 1

NSHELLS nx.i ny.i nz.i Number of unit cell shells in each direction which will be used to compute the electron density.

Default: 3 3 3

NPT nx.i ny.i Number of points in each axis.

Default: 100 100

ZMIN zmin.r Lowest bound for electron density graph, in atomic units (e/ bohr^3). This parameter will be passed down to a gnuplot script, so a value like '*' is allowed, meaning automatic fit to the lower bound of the function.

Default: *

ZMAX zmax.r Upper bound for electron density graph, in atomic units (e- / bohr^3). This parameter will be passed down to a gnuplot script, so '*' value is allowed, meaning automatic fit to the upper bound of the function.

Default: *

DC dc.i Number of contour lines used.

Default: 20

TYPE {3D | C} Choose between a 3d representation of the electron density or a 2d contour plot.

Default: 3d

SCALE {NORMAL | LOG} Select a "normal" or logarithmic representation of the plotted scalar function.

Default: normal

NOLABELS Do not include labels on the plot

Default: false

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] As a default, printrho will be run only on the new points, skipping those points already examined. The DO/EXCEPT orders can be used to modify this behavior.

Default: Use the local DO/EXCEPT specification if available. Use the global DO/EXCEPT data otherwise. In absence of user orders do the printrho task only on new points.

3.10.3 Requirements of the printrho section

This section requires general, initialization, prescf and scf sections.

3.11 Dosplot section

3.11.1 Syntax

```
#runwien#
DOSPLOT
  PLOTUNITS {EV|RY}
  ENERGYMIN emin.r [EV]
  ENERGYMAX emax.r [EV]
  DE de.r [EV]
  BROAD broad.r
  DOS LIST
    atom.i descr.r label.s
    ...
  END DOS LIST
  SPIN {MERGE | NEW | NO}
  JOIN A[UP|DN][\*] B[UP|DN][\*] [ C[UP|DN][\*] ... ]
  PLOTXMIN plotxmin.r [EV]
  PLOTXMAX plotxmax.r [EV]
  IN1MAXENERGY in1maxenergy.r [EV]
  DO [range.i | ALL | NONE | NEW | OLD]
  EXCEPT [range.i]
END DOSPLOT
```

Print the density of states plot for general structures. The DOS is generated using tetra, for the selected atom (or interstitial) and angular momentum values. The dosplot section expands the utility set of WIEN2k: the user can join several DOS plots in one, optionally scaled with multiplicity, merge or split spin DOS, etc. Also, as in printrho section, the .gnuplot is saved, so the user can customize it.

```
#runwien#
dosplot default
```

Same as above but assuming defaults on all variables.

3.11.2 Keywords and environments

PLOTUNITS {EV|RY} Units for the DOS representation: electron-volts or rydbergs.

Default: ry

ENERGYMIN emin.r [EV] Lower energy bound for density of states mesh calculation, in Ry. If eV is indicated, runwien will assume <emin> is in eV.

Default: lower bound for lowest energy band. -2.0 Ry if nothing appropriate is found.

ENERGYMAX emax.r [EV] Upper energy bound for density of states mesh calculation, in Ry. If eV is indicated, runwien will assume <emax> is in eV.

Default: upper bound for highest energy band. 1.0 Ry if nothing appropriate is found.

DE de.r [EV] Energy increment in the mesh, in Ry. If eV is indicated, runwien will assume <de> is in eV.

Default: 0.0025 Ry

BROAD broad.r Gauss-broadening factor.

Default: 0.003

DOS LIST environment The syntax of this environment corresponds to:

```
#runwien#
DOS LIST
  atom.i descr.r label.s
  ...
END DOS LIST
```

Sequence of lines, each of them indicating a DOS calculation. Atom represents the atom for which the dos is calculated, 0 for total dos, nneq+1 is the interstitial region, where nneq is the number of non-equivalent atoms. descr refers to the column of the qtl-file; 1 total, 2 s, 3 p... label is an identification text.

Default: 0 1 tot

SPIN {MERGE | NEW | NO} Specifies the treatment of spin-up and spin-down density of states. Merge means dos-up, dos-down and total dos will share the same plot. New, dos-up, dos-down and total dos each of them will occupy an individual plot. No, dos-up and dos-down will be in separate plots and total dos will not be plotted.

The spin keyword can only be used in spinpolarized calculations.

Default: new

JOIN A[UP|DN][*] B[UP|DN][*] [C[UP|DN][*] ...] (In case you are reading the ASCII documentation, * stands for the asterisk character)

Join several DOS in one plot. a, b, c, ... stand for DOS plot numbers, corresponding to line numbers in the dos list environment. up and dn flags are only applicable in spinpolarized calculations, and indicate the DOS spin that is to be used (if none is indicated, then total DOS). * flag instructs runwien to multiply the DOS by the multiplicity of the non-equivalent atom. Note * is not permitted for interstitial (nneq + 1) or total (0) DOS plots. *, up and dn can exchange positions, as long as they appear after the dos line number and they form one field.

Default: --.

PLOTXMIN plotxmin.r [EV] Lower bound for energy in dos graph, in Ry. If eV is indicated, runwien will assume <plotxmin> is in eV. This value will be passed to the gnuplot script, so '*' is acceptable input, meaning that gnuplot automatically fits the graph limit.

Default: *

PLOTXMAX plotxmax.r [EV] Upper bound for energy in dos graph, in Ry. If eV is indicated, runwien will assume <plotxmax> is in eV. This value will be passed to the gnuplot script, so '*' is acceptable input, meaning that gnuplot automatically fits the graph limit.

Default: *

IN1MAXENERGY in1maxenergy.r [EV] Upper bound for extending the eigenvalue calculation using lapw1, in Ry. If eV is indicated, runwien will assume <in1maxenergy> is in eV.

Default: maxenergy in initialization section.

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] As a default, dosplot will be run only on the new points, skipping those points already examined. The DO/EXCEPT orders can be used to modify this behavior.

Default: Use the local DO/EXCEPT specification if available. Use the global DO/EXCEPT data otherwise. In absence of user orders do the dosplot task only on new points.

3.11.3 Requirements of the dosplot section

General, initialization, prescf, scf sections are required.

3.12 Bandplot section

3.12.1 Syntax

```
#runwien#
BANDPLOT [DEFAULT]
  KLIST {TEMPLATE tplfile.s | FILE <file>}
  IN1MAXENERGY in1maxenergy.r [EV]
  DO [range.i | ALL | NONE | NEW | OLD]
  EXCEPT [range.i]
END BANDPLOT
```

Print the band structure (spaghetti plots). Bandplot section uses WIEN2k utility spaghetti to create the plots.

```
#runwien#
bandplot default
```

Same as above but assuming defaults on all variables.

3.12.2 Keywords and environments

KLIST {TEMPLATE tplfile.s | FILE <file>} Enters the k-list file to be used in the spaghetti plot. If the keyword template is found, then file must be one of the template files in WIEN2k root directory ($WIENROOT/SRC_templates), with or without the .klist extension. At the time we are writing this, the available files are bcc.klist, fcc.klist, hcp.klist and simple_cubic.klist. If the user wishes to provide a different k-list file, file keyword is used, with the path and name of the file (with or without k-list).

K-list files are easily created using xcrysden, available at http://www.xcrysden.org .

Default: according to the lattice type, a template file will be selected: bcc.klist for B, fcc.klist for F, simple_cubic.klist for P and hcp.klist for H. If the lattice type is not one of those, an error will be risen and no band plot will be done.

IN1MAXENERGY in1maxenergy.r [EV] Upper bound for the eigenvalue calculation using lapw1, in Ry. If eV is indicated, runwien will assume in1maxenergy.r is in eV.

Default: maxenergy in initialization section.

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] As a default, bandplot will be run only on the new points, skipping those points already examined. The DO/EXCEPT orders can be used to modify this behavior.

Default: Use the local DO/EXCEPT specification if available. Use the global DO/EXCEPT data otherwise. In absence of user orders do the bandplot task only on new points.

3.12.3 Requirements of bandplot section

General, initialization, prescf and scf sections are required.

3.13 Kdos section

3.13.1 Syntax

```
#runwien#
```

```
KDOS
  PLOTXMIN plotxmin.r [RY]
  PLOTXMAX plotxmax.r [RY]
  PLOTYMAX plotymax.r
  KDOS ndos.i[UP|DN] [MERGE|UP|DN]
  DO [range.i | ALL | NONE | NEW | OLD]
  EXCEPT [range.i]
END KDOS
```

Print a diagram combining the band structure spaghetti plots (K) with the density of states (DOS). Use gnuplot version 4.2 or superior for best appearance of the plot.

```
#runwien#
kdos default
```

Same as above but assuming defaults on all variables.

3.13.2 Keywords and environments

PLOTXMIN plotxmin.r [RY] Lower bound for energy in dos graph, in eV. If Ry is indicated, runwien will assume <plotxmin> is in Ry. The '*' character may be used, but the behavior is the same as if no plotxmin were issued.

Default: if a dosplot section was executed in the same run, use its plotxmin. Else, -12.0.

PLOTXMAX plotxmax.r [RY] Upper bound for energy in dos graph, in eV. If Ry is indicated, runwien will assume <plotxmax> is in Ry. The '*' character may be used, but the behavior is the same as if no plotxmax were issued.

Default: if a dosplot section was executed in the same run, use its plotxmax. Else, 8.0.

PLOTYMAX plotymax.r Upper limit for DOS plot.

Default: maximum value for DOS in the selected x-range.

KDOS ndos.i[UP|DN] [MERGE|UP|DN] The kdos keyword creates a kdos diagram. The DOS used is the <ndos> line in the dos list (see dosplot section). If an 'up' (resp. 'dn') is appended to the <dos> identifier, use the spin-up (resp. spin-down) density of states. Also, the band structure diagram may be generated using the spin-up ('up'), spin-down ('dn') or both ('merge') spin contributions to the energy bands by writing the appropriate second field.

Default: no extra kdos plots.

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] As a default, bandplot will be run only on the new points, skipping those points already examined. The DO/EXCEPT orders can be used to modify this behavior.

Default: Use the local DO/EXCEPT specification if available. Use the global DO/EXCEPT data otherwise. In absence of user orders do the bandplot task only on new points.

3.13.3 Requirements of the kdos section

General, initialization, prescf, scf, dosplot and bandplot sections are required.

3.14 Rxplot section

3.14.1 Syntax

```
#runwien#
RXPLOT [DEFAULT]
  ATOM nneq.i
  N n.i
  L l.i
  PLOTXMIN emin.r
  PLOTXMAX emax.r
  DE de.r
  TYPE [EMIS | ABS]
  IN1MAXENERGY emax.r
  DO [range.i | ALL | NONE | NEW | OLD]
  EXCEPT [range.i]
END RXPLOT
```

Prints calculated x-ray emission or absorption spectra for general structures. The rxplot section uses WIEN2k xspec facility to obtain the spectra data, but then creates a gnuplot file to print it. As in previous sections, this gnuplot file is saved in the general structure subdirectory.

```
#runwien#
rxplot default
```

Same as above but assuming defaults on all variables.

3.14.2 Keywords and environments

ATOM nneq.i Calculate the spectra for the indicated non-equivalent atom.

Default: 1

N n.i Quantum number n for orbital involved in electronic absorption or emission.

Default: 1

L l.i Quantum number l for orbital involved in electronic absorption or emission.

Default: 0

PLOTXMIN emin.r Minimum energy for x-ray spectra graph, in eV.

Default: -2.0

PLOTXMAX emax.r Maximum energy for x-ray spectra graph, in eV.

Default: 15.0

DE de.r Energy increment for x-ray spectra graph, in eV.

Default: 0.02

TYPE [EMIS | ABS] Absorption or emission x-ray spectra.

Default: abs

IN1MAXENERGY emax.r Maximum energy for the eigenvalue window calculation in .in1, in Ry.

Default: 10.0 Ry

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] As a default, rxplot will be run only on the new points, skipping those points already examined. The DO/EXCEPT orders can be used to modify this behavior.

Default: Use the local DO/EXCEPT specification if available. Use the global DO/EXCEPT data otherwise. In absence of user orders do the rxplot task only on new points.

3.14.3 Requirements of the rxplot section

General, initialization, prescf and scf sections are required.

3.15 Aim section

3.15.1 Syntax

```
#runwien#
AIM [DEFAULT]
  ATOM nneq.i
  USES {TWO | THREE | ALL | FOUR}
  NSHELLS nx.i ny.i nz.i
  DO [range.i | ALL | NONE | NEW | OLD]
  EXCEPT [range.i]
END AIM
```

Locate critical points in electron density by using the WIEN2k program aim. For now, calculating the basin properties is not supported, nor does the analysis of the output file.

```
#runwien#
aim default
```

Same as above but assuming defaults on all variables.

3.15.2 Keywords and environments

ATOM nneq.i Atom from where the search is started, counting multiplicity.

Default: 1

USES {TWO | THREE | ALL | FOUR} Start the critical point search between pairs of atoms, all three atoms, all two and three atoms or all four atoms respectively.

Default: four

NSHELLS nx.i ny.i nz.i Number of unit cell shells which will be used to compute electron density, in each direction.

Default: 3 3 3

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] As a default, aim will be run only on the new points, skipping those points already examined. The DO/EXCEPT orders can be used to modify this behavior.

Default: Use the local DO/EXCEPT specification if available. Use the global DO/EXCEPT data otherwise. In absence of user orders do the aim task only on new points.

3.15.3 Requirements of the aim section

General, initialization, prescf and scf sections are required.

3.16 Critic section

3.16.1 Syntax

```
#runwien#
CRITIC [DEFAULT]
  LINE {ATOM NNEQ mult.i | x.r y.r z.r}
    .. {ATOM NNEQ mult.i | x.r y.r z.r} npts.i
  NICE nice.i
  INTEGRALS nr.i ntheta.i nphi.i
  BASINPLOT level.i [delta.r [theta.r phi.r]]
  {IWS | NOIWS} [i.i]
  NEWTON [r.r]
  GRDVEC
     FILES rootname.s
     PLANE x0.r y0.r z0.r x1.r y1.r z1.r x2.r y2.r z2.r
     SCALE sx.r sy.r
     PARAM rad1.r rad2.r rad3.r step.r endpt.r proj.i
     ORIG x.r y.r z.r atr.i up.i down.i
     CP id.i up.i down.i
     CPALL
     BCPALL up.i down.i
     RBCPALL bup.i bdown.i rup.i rdown.i
     CHECK
          x.r y.r z.r
          ...
     ENDCHECK
     {RHO|DENSITY|LAP|LAPLACIAN} [LOG|ATAN|BADER]
        .. [nptsu.i nptsv.i niso.i]
  END GRDVEC
  DOPLOT [DENSITY|RHO|GRADMOD|LAPLACIAN|LAP|GRADIENT]
      .. [FORMAT {DEFAULT|GNUPLOT|D2D|D3D|CARTESIAN|3DCUBE|CUBE}]
      .. [FILE rootfile.s]
      .. {1D x0.r y0.r z0.r x1.r y1.r z1.r nu.i
      .. | 2D x0.r y0.r z0.r x1.r y1.r z1.r x2.r y2.r z2.r nu.i nv.i
      .. | 3DC x0.r y0.r z0.r x1.r y1.r z1.r nu.i nv.i nw.i}
  USE {RHO | VALRHO | ATOMIC}
  NOSUMMARY
  DO [range.i | ALL | NONE | NEW | OLD]
  EXCEPT [range.i]
END CRITIC
```

Locate critical points in electron density by using the critic program. Optionally, calculate .basin files, adequate as tessel input for plotting atomic basins. Creates a final summary in criticsummary.out, in root directory.

The critic output file contains much more information than the topology and planarity extracted by runwien to the general summary. This output file, with extension ".outputcritic", is saved to the general structure subdirectory, and can be read/processed/printed by the user. The input file, ".incritic", is also saved and can be edited to continue the topological analysis beyond the routine tasks.

```
#runwien#
critic default
```

Same as above but assuming defaults on all variables.

3.16.2 Keywords and environments

LINE {ATOM NNEQ mult.i | x.r y.r z.r} {ATOM NNEQ mult.i | x.r y.r z.r} npts.i Print the electron density and other properties on a regular grid of <npts> points in the indicated segment. The end points of the segment can be atoms or general positions entered as crystallographic coordinates.

Default: don't calculate any line.

NICE nice.i Set the critic process priority. Some critic tasks (determining the atomic basins and integrating properties in them) require a large cpu time, so it is useful to lower the process priority in this case. Notice that runwien child processes can not have a lower nice than runwien itself.

Default: 0

INTEGRALS nr.i ntheta.i nphi.i Calculate the atomic basin integrals, that currently provide the atomic charge and volume for each basin. Use <nr> radial points, and <ntheta> and <nphi> angular points. The algorithm used in critic is a bisection-like method, where the criterion for bisection is the nuclear cp where the gradient line ends, starting from an arbitrary point.

Default: do not integrate.

BASINPLOT level.i [delta.r [theta.r phi.r]] Plot attraction basin for each of the ncp. Basinplot starts with an octahedron and subdivides it <level> times, if <level> > 0; or starts with a cube and subdivides it -<level> times if <level> < 0. Then it finds the zero-flux surfaces (bisection method) on the rays determined by the resulting polyhedron. Delta is the precision (cryst. coords.) with which the surface is determined in each ray. Occasionally, a ray is traced along a high symmetry line and the bisection method fails to determine the true surface. A small rotation in theta and/or phi (1e-3 or 1e-4, typically) can then be applied to prevent this problem.

Default: do not plot basin files.

{IWS | NOIWS} [i.i] Defines the criteria to be used in the search for critical points. The search will start on a partition of the whole unit cell (option NOIWS) or of the irreducible part of the Wigner-Seitz cell (option IWS). <i> degrees of subdivision will be used in any case.

Default: noiws and i = 3 (or whatever is the current critic default).

NEWTON [r.r] Do an automatic search for the critical points of the electron density using Newton's algorithm. If critic fails to find all the critical points, try first to increase the subdivision level in the IWS or NOIWS order. There are other techniques for this same purpose, but Newton's is the best method on most cases, and it is the only one that can be used through runwien. Edit the critic input file and run critic interactively if you want to explore other possibilities.

Notice that, by default, critic section does not calculate anything. Newton must be indicated in order to find the critical points of the electron density. Also, be careful with exponential numbers with "d" or "q" as the exponential symbol, as AWK does not understand this notation.

Default: r = 1e-15

GRDVEC environment The syntax of this environment corresponds to:

```
#runwien#
GRDVEC
   FILES rootname.s
   PLANE x0.r y0.r z0.r x1.r y1.r z1.r x2.r y2.r z2.r
   SCALE sx.r sy.r
   PARAM rad1.r rad2.r rad3.r step.r endpt.r proj.i
   ORIG x.r y.r z.r atr.i up.i down.i
   CP id.i up.i down.i
   CPALL
   BCPALL up.i down.i
```

```
     RBCPALL bup.i bdown.i rup.i rdown.i
     CHECK
          x.r y.r z.r
          ...
     ENDCHECK
     {RHO|DENSITY|LAP|LAPLACIAN} [LOG|ATAN|BADER]
        .. [nptsu.i nptsv.i niso.i]
  END GRDVEC
```

Include a grdvec environment in the critic input file. Grdvec plots a plane containing the gradient paths originating from a critical point. For a detailed explanation of the syntax, see the critic program documentation.

Default: --.

DOPLOT [DENSITY|RHO|GRADMOD|LAPLACIAN|LAP|GRADIENT] [FORMAT {DEFAULT|GNUPLOT|D2D|D Generalized plot function in critic. Allows plotting of 1d, 2d and 3d representations of density, gradient and laplacian. For an exhaustive description of the input, consult critic's manual.

Default: --.

USE {RHO | VALRHO | ATOMIC} The USE keyword switches the clmsum used in critic input to that of valence density (valrho) and promolecular density (atomic).

Default: rho.

NOSUMMARY Don't generate the critic summary.

Default: --.

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] As a default, critic will be run only on the new points, skipping those points already examined. The DO/EXCEPT orders can be used to modify this behavior.

Default: Use the local DO/EXCEPT specification if available. Use the global DO/EXCEPT data otherwise. In absence of user orders do the critic task only on new points.

3.16.3 Requirements of critic section

Requires general, initialization, prescf and scf sections.

3.17 Sweep section

3.17.1 Syntax

```
#runwien#
SWEEP
  REFERENCE STRUCT n.i
  NPT [atom.i] pt.i
  RMT [atom.i] rad.r
  R0 [atom.i] rad.r
  RKMAX rkmax.r
  LMAX lmax.i
  LNSMAX lnsmax.i
  GMAX gmax.r
  MIX mixfactor.r
  KPTS kpts.i
  LDA+U {SIC|AMF|HMF}
```

```
 iat1.i {l1.i|l1.s} u.r j.r
 iat2.i {l2.i|l2.s} u.r j.r
 ...
END LDA+U
WITH {A|B|C | ALPHA|BETA|GAMMA | V | C/A|C/B|B/A} range.r
ALSO
PRINT [zvar.s VS]  yvar.s  VS  xvar.s
XCPOTENTIAL [LSDA|GGAPBE96|GGAPW91|GGAWC06]
ECOREVAL ecore.r
ENERGYMIN emin.r
ENERGYMAX emax.r
ORBITALS neq_atom.i global_energy.r global_apw.i
  orb_l.i orb_energy.r orb_var.r {CONT|STOP} apw.i
  ...
END ORBITALS
LM LIST neq-atom.i
  l1 m1 l2 m2 l3 m3 ...
  lk mk lk+1 mk+1 ...
  ...
END LM LIST
FERMI {ROOT|TEMP|TEMPS|GAUSS|TETRA|ALL} value.r
MAX ITERATIONS miter.i
CHARGE CONV cconv.r
ENERGY CONV econv.r
FORCE CONV fconv.r
ITDIAG [cyc.i]
NEW IN1 cyc.i
NICE nice.i
REUSE {CHAIN|REFERENCE|PATH|DETECT}
NOREUSE
NOSUMMARY
SPINORBIT item_list.i
  ...
END SPINORBIT
ELASTIC item_list.i
  ...
END ELASTIC
PRINTRHO item_list.i
  ...
END PRINTRHO
DOSPLOT item_list.i
  ...
END DOSPLOT
RXPLOT item_list.i
  ...
END RXPLOT
BANDPLOT item_list.i
  ...
END BANDPLOT
AIM item_list.i
  ...
END AIM
CRITIC item_list.i
```

```
      ...
    END CRITIC
    MINI [commandline.s]
    CLEAN [WIEN | FULL]
    DO [range.i | ALL | NONE | NEW | OLD]
    EXCEPT [range.i]
  END SWEEP
```

Examine the properties of the crystal as one or more cell parameters change their value, independently or not. The sweep calculation inherits, as a default, the cell parameters entered in the general section but these can be overridden manually. The results may be printed to a xy- or xyz-file and plotted out using a gnuplot script.

The sweep section, thus, is runwien's power tool for potential energy surface exploration. This exploration may be done for any arbitrary crystal system. Notice, however, that the reference crystal system and symmetry, entered in the general section, **can not** be modified in the sweep section. For example, if the reference structure is cubic, the cell can't be deformed into anything else than a cube. This can be tricked, however by using a tiny deformation on the reference structure. For instance, use something like 90.0001 on the three cell angles if you want to explore a rhombohedral deformation.

All the properties seen in previous sections (printrho, dosplot, ...) can be calculated for whatever sweep structures you like.

These properties are calculated in a natural way, for the internal working of the sweep structure involves one .wien input file for each structure calculated. Therefore, all the "property section" lines are passed directly to the indicated structures' wien file, and runwien executed on them.

All the sweep calculations are contained in a directory that is called <root>-sweep, where <root> is the root name for the calculation. This directory contains one subdirectory for each structure and an index file. Each subdirectory starts with one wien file. When every wien file is written, runwien batch-calculate all of them unless stated otherwise by the user.

The same considerations for the variation of parameter as in the general section apply in this section. Thus if you request runwien to calculate a = 4.3, 4.4 and c = 6.7, 6.8 ; four structures will be calculated with all the possible combinations. One difference is that in sweep section there is no link option. The also keyword is available too.

At the end of the sweep section, a summarizing script will be run and a summary file (sweepsummary.out) will be created in the root directory, containing the relevant information of the sweep run.

The sweep section has no default orders. Enter explicitly whatever you want it to perform.

3.17.2 Keywords and environments

REFERENCE STRUCT n.i Select which structure from the general section will be used as reference. Any parameter or condition not explicitly set will be inherited from this reference.

Default: the first structure that appears in the general section.

NPT [atom.i] pt.i Number of radial mesh points, for non-equivalent atom index atom.i. Must be an odd number. WIEN2k manual recommends 381 for lda calculations, 781 for gga. Please, note that the argument for npt is *not* a list of values, i.e., npt is fixed throughout all the sweep run. "auto" can be used and it will be passed into local .wien files.

Auto keyword will make runwien accept runwien's default value. Npt value will not change from one structure to the other, but, generally speaking, auto keyword should be avoided in sweep section, as all the points in the potential energy surface must have the same calculation level.

All this comments are valid for general section's varying parameters in sweep section (rmt, r0, ...).

Default: reference structure.

RMT [atom.i] rad.r Muffin tin radius, in bohr, for non-equivalent atom index atom.i.

Default: reference structure, if no atom is indicated, 1 is assumed.

R0 [atom.i] rad.r First radial mesh point, in bohr, for non-equivalent atom index atom.i. WIEN2k manual recommends a value between 0.0005 and 0.00005.

Default: reference structure.

RKMAX rkmax.r rmt * Kmax, where Kmax is the maximum k for plane waves and rmt is the smallest muffin tin radius.

Default: reference structure.

LMAX lmax.i Maximum l value for spherical harmonics expansion inside muffin tins. Required in initialization, same as above.

Default: reference structure.

LNSMAX lnsmax.i Maximum l value for spherical harmonics expansion used to computate matrix elements with non-muffin tin functions. Required in initialization, same as above.

Default: reference structure.

GMAX gmax.r Maximum reciprocal vector for fourier expansion of electron density. Wien2k manual recommends using a large gmax (25) for systems with short H bonds. Also, a larger gmax for gga calculations (14) than for lsda potentials. Required in initialization, same as above.

Default: reference structure.

MIX mixfactor.r Mixing factor used in the creation of the new charge density.

Default: reference structure.

KPTS kpts.i Number of k-points in the whole first brillouin zone.

Default: reference structure.

LDA+U environment Formed as

```
LDA+U {SIC|AMF|HMF}
    iat1.i {l1.i|l1.s} u.r j.r
    iat2.i {l2.i|l2.s} u.r j.r
    ...
END LDA+U
```

Includes an orbital dependent correction to the exchange-correlation potential. This method allows LDA (or GGA) calculations to successfully reproduce the properties of the so-called 'strongly correlated systems'. Common systems of this kind are transition metal oxides and rare earth elements and compounds. For more information, see the WIEN2k user's guide and references therein.

The first keyword in the environment selects the flavor of LDA+U:

- SIC: Self-Interaction Correction (Anisimov et al., 1993).
- AMF: Around the Mean Field (Czyzyk et al., 1994)
- HMF: Hubbard model in Mean Field (Anisimov et al., 1991).

Each line in the LDA+U environment activates the U correction for a particular atom and atomic symmetry (l). The first field corresponds to the non-equivalent atom index in the unit cell. The second field selects the angular momentum quantum number l, either by its value (0, 1, 2,..) or by the corresponding letter (s, p, d,...).

The U (Coulomb on-site parameter) and J (exchange parameter) characterize the LDA+U correction. These parameters are set using the u.r and j.r values respectively.

Default: reference structure.

WITH {A|B|C | ALPHA|BETA|GAMMA | V | C/A|C/B|B/A} range.r Grid specification. A "with" line specifies a varying parameter for current crystal system. <range> follows the same <min>/<max>/<step> syntax as the general section. For example, one can vary the parameter of a cubic structure with a line:

```
#runwien#
with a 9.0/12.5/0.5,12.7,12.8
```

You can also use % in the specification if you like. In that case, quantities are calculated against the reference structure's parameters.

A structure of lesser symmetry must have more with keywords added. For example, an hexagonal lattice can vary:

```
#runwien#
with c/a 1.5/1.82/.2
with v 300/500/50,600
```

Please, note all parameters refer to conventional unit cell dimensions and atomic units. Angles are measured in sexagesimal degrees.

For reference structure's crystal system, only certain parameters and parameter rations are allowed by runwien to vary. These are:

- cubic: a or v
- hexagonal: (a and c) or (c/a and v)
- rhombohedric: a and alpha
- tetragonal: a and c
- orthorhombic: a, b and c
- monoclinic: a, b, c and alpha
- triclinic: a, b, c, alpha, beta, gamma

These, and only these, parameters are permitted for a given crystal system. **All other parameter specification different from the above will be ignored**. For example, it is not allowed to vary v and b/a for an hexagonal structure, nor to indicate the geometry of an orthorhombic system using v, c/a and a.

Notice that the volumes entered in the WITH order are conventional cell volumes but the ones in the synopsis and in the summary files are volumes **per unit formula**. Be very careful when copying and pasting from an output to an input!

Default: one structure, equal to the reference structure.

ALSO Used to separate two or more different and independent grid scans. For example:

```
#runwien#
with c/a 1.82
with v 300/500/50
also
with c/a 2.5,2.6
with v 310
```

corresponds to two different scans of the cell volume, the first one having fixed the ratio c/a, and the second one a fixed volume.

Default: none

PRINT [zvar.s VS] yvar.s VS xvar.s Variables to be used in a 2D or 3D "property versus geometry" plot. <zvar> or, if z is missing, <yvar> may be one of energy, efermi, planarity, topology, a, b, c, alpha, beta, gamma, v, c/a, c/b, b/a. Each specification of these keywords generates a file, <root>-sweep.print<n> where <n> is the print line number and <root> the calculation root name. If <z-var> exists, a xyz-file is created (first column x, second column y, third column z) adequate for gnuplot. If only <x-var> and <y-var> are present, an xy-file is created. Finally, if adequate gnuplot scripts are present (and indicated in constant part of runwien's code), they can be run to create the plot.

You can add your own gnuplot scripts into runwien code. To do this, you must add the name of your gnuplot script to the array const_gnuplot[:,:,:] where the dimensions of the array are indicated by a string containing the type of quantity the script is intended for. For example:

```
#ascii#
const_gnuplot["v","c/a","energy"] = "lapw_plot_e_ca_v.gnuplot"
```

The gnuplot script must be available through your PATH and readily executable (it will be called ./gnuplot-script, not gnuplot gnuplot-script).

Default: no prints.

XCPOTENTIAL [LSDA|GGAPBE96|GGAPW91|GGAWC06] Chooses one of the available exchange-correlation potentials.

Default: reference structure.

ECOREVAL ecore.r Initial core-valence separation energy, in Ry.

Default: reference structure.

ENERGYMIN emin.r Lower energy cutoff for eigenvalue search.

Default: reference structure.

ENERGYMAX emax.r Higher energy cutoff for eigenvalue search.

Default: reference structure.

ORBITALS environment The syntax of this environment corresponds to:

```
#runwien#
ORBITALS neq_atom.i global_energy.r global_apw.i
  orb_l.i orb_energy.r orb_var.i {CONT|STOP} apw.i
  ...
END ORBITALS
```

Orbital list, in the same format as accepted in .wien (but, of course, free format). <neq-atom> is the number of the non-equivalent atoms for which the basis is specified. global-energy is the default linearization energy and global-apw = 1 if apw method basis function is to be used; 0 if lapw. First <orb-l> appearing in the list refers to local orbitals + plane waves basis functions of that l. Subsequent orb-l will refer to local orbitals. orb-energy is the linearization energy; orb-var > 0 commands wien to search for a good linearization energy and if stop is set, stop in case it is not found. Finally, use <apw> = 1 in case you want an apw basis function type; 0 in case of lapw. See WIEN2k's manual for more information about basis.

Default: reference structure.

LM LIST environment The syntax of this environment corresponds to:

```
#runwien#
LM LIST neq_atom.i
  l1 m1 l2 m2 l3 m3 ...
  lk mk lk+1 mk+1 ...
  ...
END LM LIST
```

Specify LM lattice harmonics expansion of charge density. See wien's manual for L M pairs adequate for each structure.

Default: reference structure.

FERMI {ROOT|TEMP|TEMPS|GAUSS|TETRA|ALL} value.r Method to find fermi energy. See WIEN2k's manual for details on each of the method. Value is a method-specific characteristic number.

Default: reference structure.

MAX ITERATIONS miter.i Maximum number of cycles.

Default: reference structure.

CHARGE CONV cconv.r Use a charge convergence criterion.

Default: reference structure.

ENERGY CONV econv.r Energy convergence criterion.

Default: reference structure.

FORCE CONV fconv.r Force convergence criterion.

Default: reference structure.

ITDIAG [cyc.i] Use iterative diagonalization and do a complete diagonalization each cyc.i cycles. In versions of WIEN2k >= 13.08.2007, cyc.i has no effect.

Default: reference structure. cyc.i = 3.

NEW IN1 cyc.i Generate a new basis set in cycle <cyc>.

Default: no, even if it is set in reference structure.

NICE nice.i Process priority for every section of local .wien. Runwien child processes can not have a lower nice than runwien itself.

Default: reference structure's scf_nice.

REUSE {CHAIN|REFERENCE|PATH path.s|DETECT} The REUSE keyword allows the user to control the reutilization of converged densities from one deformation point to the other in a sweep calculation. The CHAIN keyword makes each point use the converged density of the previous one. The first structure is calculated starting from the superposition of atomic densities.

With REFERENCE, each structure extrapolates the charge density of the selected reference structure of the general section. In a way similar to SCF section, PATH allows the user to use consistently a given calculation as starting point from every geometry. Finally, if DETECT is read, runwien searches for existing clmsum and struct files in the directories of the sweep section, and uses the density files if found.

Default: CHAIN.

NOREUSE Reusing the converged densities can lead to large time savings. The NOREUSE keyword allows the user to avoid the reutilization of densities of previous points completely.

Default: --.

NOSUMMARY Don't run the summarizing script. This script will generate a sweepsummary.out file in root directory, which contains useful information about the sweep calculation.

Default: no

SPINORBIT environment The syntax of this environment corresponds to:

```
#runwien#
SPINORBIT item_list.i
 ...
END SPINORBIT
```

Add a spinorbit section in the wien input file to some structures. <item_list> is a list of structures where this section will be included. It follows the same syntax as “with” keyword or “rmt”, etc. in general section (using n/m/l, etc.). If no <item_list> is included, all the structures will receive a copy of this code.

The lines included in the body of the environment are passed verbatim to the corresponding wien input files.

The information in this environment (and the following) is not parsed: runwien will not be able to tell if it is right or wrong until it is calculating the sweep structures.

ELASTIC environment The syntax of this environment corresponds to:

```
#runwien#
ELASTIC item_list.i
  ...
END ELASTIC
```

Add a elastic section in wien input file to some structures. <item_list> is a list of structures where this section will be included. It follows the same syntax as “with” keyword or “rmt”, etc. in general section (using n/m/l, etc.). If no <item_list> is included, all the structures will receive a copy of this code.

The lines included in the body of the environment are passed verbatim to the corresponding wien input files.

The information in this environment (and the following) is not parsed: runwien will not be able to tell if it is right or wrong until it is calculating the sweep structures.

Sweep section, combined with elastic section can build up an impressive amount of data! Use this with care.

PRINTRHO environment The syntax of this environment corresponds to:

```
#runwien#
PRINTRHO item_list.i
  ...
END PRINTRHO
```

Add a printrho section in the wien input file on some of the sweep structures. The same considerations described for the elastic environment apply here.

DOSPLOT environment The syntax of this environment corresponds to:

```
#runwien#
DOSPLOT item_list.i
  ...
END DOSPLOT
```

Add a dosplot section in the wien input file on some of the sweep structures. The same considerations described for the elastic environment apply here.

RXPLOT environment The syntax of this environment corresponds to:

```
#runwien#
RXPLOT item_list.i
  ...
END RXPLOT
```

Add a rxplot section in the wien input file on some of the sweep structures. The same considerations described for the elastic environment apply here.

BANDPLOT environment The syntax of this environment corresponds to:

```
#runwien#
BANDPLOT item_list.i
 ...
END BANDPLOT
```

Add a bandplot section in the wien input file on some of the sweep structures. The same considerations described for the elastic environment apply here.

KDOS environment The syntax of this environment corresponds to:

```
#runwien#
KDOS item_list.i
 ...
END KDOS
```

Add a kdos section in the wien input file on some of the sweep structures. The same considerations described for the elastic environment apply here.

AIM environment The syntax of this environment corresponds to:

```
#runwien#
AIM item_list.i
 ...
END AIM
```

Add an aim section in the wien input file on some of the sweep structures. The same considerations described for the elastic environment apply here.

CRITIC environment The syntax of this environment corresponds to:

```
#runwien#
CRITIC item_list.i
 ...
END CRITIC
```

Add a critic section in the wien input file on some of the sweep structures. The same considerations described for the elastic environment apply here.

MINI [commandline.s] Perform an internal parameter optimization using the mini program (through the min_lapw script). The mini line is passed down to all the wien input files in the sweep calculation. This is a temporary solution, and will be modified in later versions.

Default: no internal optimization.

CLEAN [WIEN | FULL] Append a 'CLEAN [WIEN|FULL]' order at the end of each sweep point .wien. This will make each point in sweep section clean itself after being calculated. For details on clean keyword, see description in global section.

Sweep calculations can get **very big**. Apply cleaning if you plan to calculate many structures of a system with a lot of valence electrons.

DO [range.i | ALL | NONE | NEW | OLD] Used together with the EXCEPT order below.

EXCEPT [range.i] As a default, sweep will be run only on the new points, skipping those points already examined. The DO/EXCEPT orders can be used to modify this behavior. At difference from other sections, the global DO/EXCEPT pair do not affect sweep.

Default: Use the local DO/EXCEPT specification if available. Do the sweep task only on new points otherwise.

3.17.3 Requirements of the sweep section

Requires general, initialization, prescf and scf sections.

3.18 Gibbs section

3.18.1 Syntax

```
#runwien#
GIBBS [DEFAULT]
  EOS {NUMERICAL[+{VINET|BIRCH|BCNT}] | VINET | BIRCH | BCNT} value.r
  DEBYE {STATIC|DEBYESTATIC|DEBYEITER|DEBYESTATICBV} value.r
  PRESSURE {AUTO | range.r} [,...]
  TEMPERATURE {AUTO | range.r} [,...]
  FREE ENERGY energy.r
END GIBBS
```

The 'gibbs' code, by Blanco et al [gibbs04], uses the energy versus volume curve and a quasi-harmonic Debye model, to estimate the thermal and pressure effects on the crystal geometry. The equation of state (EOS), bulk modulus, thermal expansion, etc, are calculated in the process. Runwien's GIBBS task uses the energies from a sweep run to prepare the input for the external gibbs code and extract a selection of the most important results.

The minimal code that activates this section is

```
#runwien#
gibbs default
```

It will run gibbs section assuming default values for all variables.

3.18.2 Keywords and environments

EOS {NUMERICAL[+{VINET|BIRCH|BCNT}] | VINET | BIRCH | BCNT} value.r Type of equation of state employed to adjust data. If bcnt or numerical+bcnt, a value is required. See gibbs' manual for more info.

Default: numerical (0 if <value> is not specified).

DEBYE {STATIC|DEBYESTATIC|DEBYEITER|DEBYESTATICBV} value.r Select the method used to compute the debye temperature of the crystal. Valid elections are:

- static: static EOS (no thermal effects).
- debyestatic: get the Debye temperature from computed static bulk moduli.
- debyeiter: get the Debye temperature from computed adiabatic bulk moduli (self-consistent approach).
- debyestaticbv: Debye temperature from static bulk moduli at static equilibrium volume.

<value> is poisson's ratio, only used if debyestatic or debyeiter.

Default: debyestatic.

PRESSURE {AUTO | range.r} [,...] Specify pressures for gibbs. If the list of pressures specified exceeds the possibilities of your data, it will be cut down to an appropriate maximum pressure on gibbs error.

Default: a single pressure, p = 0.

[gibbs04] M. A. Blanco, E. Francisco, and V. Luaña, "GIBBS: isothermal-isobaric thermodynamics of solids from energy curves using a quasi-harmonic Debye model", Comput. Phys. Commun. 158 (2004) 57--72.

TEMPERATURE {AUTO | range.r} [,...] Specify temperatures for gibbs.

Default: one temperature, T = 0.

FREE ENERGY energ.r Value of free in-vacuo energy for the molecular formula. This value will only be used to estimate the formation energy of the crystal. It is possible to run gibbs in absence of this datum.

Runwien will try to use the results of a previous FREE section, if it is available, or the atomic energies estimated from the INITIALIZATION section.

Default: free in-vacuo energy. If free was not run, use the atomic energies estimated on the initialization section.

3.18.3 Requirements of the gibbs section

General, initialization, prescf, scf and sweep sections are required for gibbs to run.

3.19 Synopsis section

3.19.1 Syntax

```
#runwien#
SYNOPSIS
  OUTPUT file.s
  EXHAUSTIVE list_item.i
END SYNOPSIS
```

Synopsis creates a whole runwien run summary and writes it to a specified file. This summary includes general information, calculation times, important input parameters and key output values; all pretty-printed. As it may use information from every section, synopsis is best run at the end, but this is by no means necessary.

Most significant quantities are already included in the output. However, the possibilities for this task are huge. We may add optional information using switches to trigger it in future versions.

As it is common, this section can be run using default values for all variables:

```
#runwien#
synopsis default
```

3.19.2 Keywords and environments

OUTPUT file.s Specify output filename.

Default: synopsis.out

EXHAUSTIVE list_item.i Print exhaustive information for the general section structures indicated in the <list_item>. This list follows the same syntax as "with" in sweep structure and "rmt", ... in general section. Exhaustive information includes not-so-important values like convergence analysis of the scf process, real rkmax used, space filling by muffin tin spheres, etc.

3.19.3 Requirements of the synopsis section

No other section is required for synopsis section. In fact, synopsis will detect if other sections have been run and print a summary according to it.

4 Directory tree for runwien. File locations.

All work runwien performs is constrained to <root> directory, which is where ".wien" file lives. The <root> directory has the same name as the ".wien" file (except for the .wien extension). This is a requirement for runwien to run.

The directory tree for a full exhaustive calculation is as follows:

```
#ascii#
<root>/
 |--<root>.wien                  (main input file)
 |--<root>.index                 (general section structure indexfile)
 |--<root>-sweep.print?          (sweep section print files)
 |--<root>-sweep.print?.eps      (sweep section plots)
 |--*.out                      (summaries (critic,scf,sweep,synopsis))
 |--<root><n>/                   (general section's structure)
 |   ‘--<root><n>.*              (WIEN2k's calculation files, gnuplot
 |                                scripts, ps files, output files)
 |--<root>-free-<at>/           (free section directory for atom <at>)
 |   |--<root>-free.wien         (free section main input file)
 |   ‘--<root>-free1/            (free section calculation files)
 |--<root>-elastic/              (elastic section directory)
 |   ‘--<root>-xxxxxx/           (deformation xxxxxx subdirectory)
 |      ‘--<root><n>/           (point number n in deformation xxxxxx)
 |         |--<root><n>.wien     (deformation point input file)
 |         ‘--<root><n>1/        (deformation point calculation files)
 |--<root>-sweep/                (sweep section directory)
 |   |--<root>-sweep.index       (sweep section indexfile)
 |   ‘--<root>-sweep<n>/         (sweep point local dir)
 |      |--<root>-sweep<n>.wien  (local .wien)
 |      |--<root>-sweep<n>.index (local index file)
 |      |--<root><n>.struct      (reference struct file)
 |      ‘--<root>-sweep<n>1/     (local structure)
 |         ‘--<root>-sweep<n>1.* (local wien's files)
 |--<root>-gibbs/                (gibbs section directory)
 |   |--<root>.ingibbs           (gibbs input file)
 |   ‘--<root>.outputgibbs       (gibbs output file)
 |--<root>-check/                (checkfile directory)
     ‘--*.check                  (checkfiles)
 ‘-- (user files: .cif, .struct, etc.)
```

Postscript files containing the results of the properties sections are stored in their respective structure directories. This also happens for gnuplot scripts which generate them and general output files for WIEN2k programs.

5 Test files

Some twenty full WIEN2k cases are included along with runwien. Mostly, they consist in low-cost simple systems, adequate for testing purposes and for proving the utility of runwien as an interface. The index file (test.index) contains a summary of the type of systems and a hint of what the calculation is about. I lively recommend you to run these tests before trying any real calculation so as to check everything is correctly setup.

Together with the input wien files, our distribution includes the most significant output files (synopsis.out, runwien.out --the log data that runwien prints to stdout--, and some postscript images). Keep our original outputs and compare with the results obtained in your machine to check that your installation works correctly.

Some tests like dosplot tests portray some of the new capabilities of runwien in the determination of electronic properties. You can use them as templates for your calculations if you desire, as they are fairly complete.

The complete list of tests follows. The estimated times refer to a desktop PC (dual core 3.40 GHz, 1 GB RAM) running debian (etch, kernel version 2.6.23.14).

- test001 (TiC) : this is the example in the WIEN2k user's guide. It is a single-point calculation of TiC, plot of valence density, DOS, x-ray spectrum, spaghetti plot, QTAIM analysis and 5 points in the potential energy surface. Estimated time: 7m 23s
- test002 (Ni) : another example from the WIEN2K user's guide. Fcc nickel is calculated mostly with default options. Estimated time: 1m 50s
- test003 (Ti02) : WIEN2K user's guide example. Minimal input, mostly with default options. Estimated time: 3m 36s
- test004 (Be bcc) : an example of the usage of the elastic section in a cubic crystal. The tetragonal deformation is very flat, so there is a good amount of noise in the corresponding energy plot. This affects the accuracy of the elastic constants. Estimated time: 18m 29s
- test005 (Be hcp) : example of usage of elastic in an hexagonal crystal. Estimated time: 2m 29s
- test006 (CaF2) : the cohesive energy of CaF2 is calculated using the free section. Estimated time: 9m 2s
- test007 (Cr hcp) : several examples of the usage of printrho in ferromagnetic hcp Cr. Total, valence, spin, atomic and deformation densities are plotted. Also, total, Coulomb and exchange-correlation potentials. Estimated time: 28m 20s
- test008 (Sr fcc) : example of dosplot. The total, atomic and interstitial and angular contributions to the DOS are plotted. Several joint plots are also calculated, including total = atomic + interstitial ; atomic = s + p + d + f ; and d = dz2 + (dx2y2+dxy) + (dxz+dyz). Estimated time: 7m 53s
- test009 (Cr hcp) : another DOS example, with spinpolarized ferromagnetic hcp Cr. The plots are the analogous to test008, but taken spin contributions into account. Estimated time: 20m 33s
- test010 (Ca fcc) : example of the rxplot section. Estimated time: 1m 54s
- test011 (MgO) : example of bandplot section, for a typically ionic system. Estimated time: 3m 18s
- test012 (Mg hcp) : bandplot section, for a metallic system. Estimated time: 8m 17s
- test013 (Li) : topological analysis of Li using the AIM program. The position of the critical points is determined. Estimated time: 11m 34s
- test014 (Li) : same as test013, but using CRITIC, our own AIM code. Estimated time: 11m 57s
- test015 (Cr hcp) : topological analysis of spinpolarized hcp Cr using CRITIC. Only critical point position is calculated. Estimated time: 39m 5s
- test016 (Be hcp) : example of usage of the sweep section in an hexagonal crystal. A 3x3 grid in (V,c/a) is calculated, then an independent grid in a with fixed c (3 points). This amounts to a total of 12 points. Estimated time: 5m 48s
- test017 (Be bcc) : calculation of the thermodynamic quantities in bcc Be using the GIBBS program. A total of 13 volumes are calculated and passed into GIBBS. Estimated time: 6m 34s

- test018 (Be hcp) : an example of the usage of grids in the general section. The evolution of the topology of the electron density is evaluated against the variation of rmt, rkmax, kpts and mix. Estimated time: 31m 41s
- test019 (C diam) : a two step runwien execution. In a first run, the general, initialization, prescf and scf sections are run. Then, in a second pass, the information is loaded into runwien.awk exploration of rkmax, kpts and rmt is done. Copying the struct file generated by sgroup in the first pass is necessary to shift the origin of the unit cell to a point with inversion symmetry. The sgroup struct file is not necessary in the second pass, for the information is updated right after the copy. Estimated time: 2m 30s + 30m 46s
- test020 (Be hcp) : another two step runwien job. In the first pass, general, initialization, prescf, scf and sweep sections are run. The keyword 'do none' is indicated both in scf and sweep, so that the directories and input files are generated but no actual SCF calculation is done. The user is expected to run the sweep calculations manual. In a second pass, the information is read into runwien using LOADCHECK REREAD and printed with synopsis. Estimated time: 11s + 1h 26m 20s
- test021 (TiC) : the input file generating the example in the article. The TiC rocksalt structure is calculated. Using the converged density, DOS and spaghetti plots are generated. Using these two, the kdos section is used to create a joint DOS-band plot. A topological analysis is performed, including a representation of the laplacian of the electron density in the (001) plane. The elastic constants are calculated using the elastic section. A sweep+gibbs run is done and the isolated atoms energy is calculated using free. Estimated time: 9h 32m
- test022 (ZnO) : an example of a non-centrosymmetric structure. Plots of the total electron density, DOS, band structure, KDOS, RX are generated. Also, topological analysis is carried out with critic. The Morse sum condition on the critical points is not fulfilled, pointing to the need of increased computational parameters (larger LM list, for example). Estimated time: 50m 39s
- test023 (Cr bcc) : antiferromagnetic Cr. This is done using a spinpolarized calculation in bcc, where atoms in (0 0 0) and in (1/2 1/2 1/2) are forcibly made non-equivalent. Estimated time: 6m 34s
- test024 (As) : elemental As. The spglist environment is used in this example. As has a rhombohedral unit cell, which is fed into runwien using the spglist ... rhomb keyword. Estimated time: 2m 39s

6 Programming techniques used in runwien

Runwien is released under a GPL license. This means you can modify the code to fit your needs (provided the resulting code is also under the GPL and you don't remove our names from it). However, in order to do this, you must take into account how runwien could reach the size it has now without ending up in a programming chaos. This was achieved by following some simple programming conventions:

- AWK is a scripting language. It does not have fixed variables types nor local variables so a big deal of care must be taken when creating or removing variables. In runwien every variable has a prefix which delimits its scope. For example, elastic section variables have the prefix elastic (elastic_points, elastic_maxlength, ...). A variable with a section prefix should **never** be modified outside of its section. Some variables are global and receive the global prefix, and can only be written in general and initialization sections. Temporal variables and scratch variables have the temp prefix: you should try to avoid carrying information over sections with them. Constants have the const prefix, and must not be modified.
- Variable naming is a problem in functions. It is extremely important to avoid variable collision when functions call other functions. For this, each function is assigned an integer that corresponds

to its level in the calling tree. Thus, a function that is only called in the main program is level 1, a function that is called in the main program and by level 1 functions is level 2, etc. If n is the level of the function, its variables must have the local<n> prefix.

- Runwien is a modular program. You could, for example, remove a (non-fundamental) section by eliminating the input reading section, the execution section and all the references and related functions. This removal is easier than it sounds. Adding a new section is equally easy. If you plan to do this, use any of the other sections as templates.
- There are some mechanisms your new sections must honor: LOADCHECK / LOADCHECK REREAD / SAVECHECK. Design a checkfile and put into it the variables you want to save if the user decides to work with a two-step process. This usually means result variables for properties sections and almost all variables for general, sweep and the like.
- The interesting results of your section should end up in variables, accessible for the rest of the sections. This is specially important if you want to pass information to synopsis. For example, the energy obtained from the SCF calculation goes into scf_energy[:] which is later read by synopsis. Modify synopsis if you think the quantities obtained are important enough.
- do / except, time control for the section, do / done, global / atom, etc. are mechanisms designed for easy programming and friendly usage. Try to maintain coherence by copying them if you see it fit.

In general, think that a script this size is difficult to maintain unless a good amount of self-discipline is put in it.

If you make some deep/interesting modifications to our code, we would be very grateful to hear from it.

7 In case something goes wrong...

Runwien is provided as a mere tool to automatize tedious tasks. If something goes wrong, runwien usually saves the standard output and standard error of every program it runs in files with .err extension. For example, the output of the nn program's name is <root>.nn.err. Scf programs (lapw0, lapw1, lapw2 and mixer) also produce .error files which are saved as <root>.<progname>.error. The dayfile and the .scf<n> files, together with error files are the main sources you will have to check out if there is a problem in your calculations.

Runwien incorporates a warning system for the SCF cycle (for example, it detects the presence of ghost bands), and preforms some checks on output files to see everything is running smoothly. These checks are the coded result of some of the knowledge accumulated in the WIEN2k mailing list (http://www.wien2k.at/reg_user/inde
but it is by no means complete.

If something goes awry, but you feel you can repair it easily, you can manually do it and reload the calculation into runwien using the LOADCHECK system.

As a final remark, remember this program is provided with **no warranty**, at all. You should always double-check your data for errors.

8 Copyright notice

Generated on: 2008-12-31 00:44:12 CET. Generated by Docutils from reStructuredText source.