

Supplement to the TB-LMTO-ASA (version 47) program

Florent Boucher*, O. Jepsen, and O. K. Andersen
Max-Planck-Institut für Festkörperforschung, Heisenbergstr. 1
D-70569 Stuttgart, Federal Republic of Germany

*Permanent address

UMR C 6502 - Institut des Matériaux de Nantes, 2 rue de la Houssinière
44322 Nantes Cedex 03, France

E-mail: boucher@cnsr-immn.fr

This program package calculates COHP from self-consistent TB-LMTO-ASA potentials and wave functions. It includes all standard TB-LMTO-ASA programs, but some of them have been modified and three new programs added:

lmincohp.run to generate the "COHP" input file

lmcohp.run to perform the COHP calculation

gnucohp.run to generate the command and data files suitable for Gnuplot.

Few options have been added to the original package.

(last update: october 1997)

I - New options

Category OPTIONS

11. Token CHARGE= of cast logical (used by *lm.run* and *lmbnd.run*)

The charge density can be calculated either from the full BZ (with *lm.run*) or with only one (or few) k point(s) (with *lmbnd.run*). If the latter option is chosen, the k points used in the calculations are those defined in the category SYML. For NQ=1 the point Q1 is used. All the options in the category PLOT are still valid with *lmbnd.run*. The option CHARWIN=T can be used now to look the character of some bands above the Fermi level (with *lmbnd.run* only). In the case of charge calculations with *lmbnd.run*, the files BNDS and EIGN are not overwritten.

18. Token COHP= of cast logical (used by *lm.run* only)

T: Informations for COHP calculations (class of atoms for the different bonds) are read from the file "COHP" by *lm.run* and the data needed for COHP calculations (on site overlap matrix, off-diagonal Hamiltonian matrix and eigen vectors) are written on disk after the diagonalisation procedure. These data are saved in the files "COVL", "CHAM" and "CVEC", respectively. They correspond to the β representation (that is, after down folding (O.K. Andersen, Z. Pawłowska, and O. Jepsen, *Phys. Rev. B* **34**, 5253 (1986))).

19. Token RMES=of cast logical (used by *lmbnd.run* with FATBAND=T and *lmctl.run* with WRIBAS=T)

T: Imposes *lmbnd.run* to disregard all empty spheres when writing the EIGN file for the fat bands. This option saves disk space. When used with WRIBAS=T, *lmctl.run* doesn't write the ES positions in the "cstruc" file.

20 - 21. Token BEGBND and ENDBND= of cast integer (used by *lm.run* with FS=T)

For the generation of files by *lm.run* for Fermi surface plot. Only the bands between BEGBND and ENDBND are selected. This option saves disk space. In the default mode (BEGBN=0 and ENDBND=0) all the bands are used.

Category PLOT

7. Token FORMAT=of cast integer

Output files (with extension ".xsgi") suitable for IrisExplorer are produced with FORMAT=4. Several IrisExplorer maps (Rho, ELF, Rho and ELF, Ball and Stick, Fermi Surface) have been included in the LMTO package, in the directory IrisExplorer.

Category CHARGE

8. Token UNOCCUP= of cast logical (used by *lm.run*)

The character of some bands above the Fermi level can be obtained from a full BZ integration using the option UNOCCUP=T. In this case, the energy range of interest is defined with the option CHARWIN=T and with the two values EMIN and EMAX.

Category COHP

1. Token ONLY_H= of cast logical (used by *lmcohp.run*)

T: Only the real space Hamiltonian matrix is calculated and no COHP calculation is done. This option can be used to calculate only the Slater-Koster TB parameters.

2. Token f(N(E))= of cast logical (used by *lmcohp.run*)

F: COHP is calculated as a function of energy

T: COHP is calculated as a function of the number of states. ICOHP(N) is obtained at mesh points through cubic

spline interpolation and COHP(N) is calculated by numerical derivation of ICOHP(N).

3. Token WGAUSS= of cast integer (used by *lmcohp.run* if $f(N(E))=T$)

Before the numerical derivation of ICOHP(N), a Gaussian filter is applied. WGAUSS defines the width (in point unit) of the Gaussian function used to smooth ICOHP(N): $\text{Gauss}(WGAUSS+1) = 10^{-4}$. If WGAUSS=0, no filter is applied.

4. Token CLASS1, CLASS2= of cast integer (used by *lmincohp.run*)

The class of the source atoms and of the target atoms to select particular bonds.

5. Token DIMIN, DIMAX= of cast double (used by *lmincohp.run*)

The distance interval (given in atomic unit) to search the bonds between the atoms of CLASS1 and the atoms of CLASS2. Several lines with tokens 4 and 5 can be used.

gnudos.run

Now it is possible to plot both the DOS and IDOS curves on the same plot. For this, at least the latest 3.5 version of gnuplot has to be used (beta version pre 3.6). Specific options of this gnuplot version are also used by the other *gnu* interfaces. Beta 3.5 versions are available on <ftp://cmpl1.phys.soton.ac.uk/> and the gnuplot 3.5 beta 315 is included in the package in the directory Gnuplot.

II - Running the TB-LMTO-ASA-COHP programs

In the following sections, it is assumed that the full self-consistent calculations have already been performed. In order to work with the minimall basis set, one should downfold all the LMTO of the empty spheres.

Step 1 - Generation of the "COHP" file

The "COHP" ASCII file contains all the parameters necessary to define the bonds (to be described below). It is used by the programs *lm.run* and *lmcohp.run*. The file is automatically generated by the program *lmincohp.run* according to the CTRL file parameters (see new category COHP, token 4 and 5). After its creation, it can be edited to suppress some bonds (equivalent bonds for instance). If the logical IACTIV=T, a possibility to write/not write bond informations in the "COHP" file is proposed by the program *lmincohp.run*. An example of a "COHP" file is given at the end of this document. The bond length is reported in atomic unit and in Angstrom. The positions of the source atoms correspond to the positions IBAS1 in the "CTRL" file and the positions of the target atoms are obtained by applying

the given primitive unit cell translations (TRANSLATIONS=) to the positions IBAS2 of the "CTRL" file. The bond directions are indicated by VECTOR (given in units of ALAT and in Cartesian coordinates). The information in the field ORBITALS= is written according to the downfolding indices of each atom class. Only the orbitals with IDXDN=1 are kept.

Step 2 - Generation of data for COHP calculations

The program *lm.run* has to be run with the option COHP=T. Only the values IBAS1 and IBAS2 in the "COHP" file are used to save the relevant data on disk (Hamiltonian matrix, overlap matrix and eigen vectors). At least one "bndasa" calculation is needed to save all expected values.

Step 3 - COHP calculations

The program *lmcohp.run* is run with the option ONLY_H=F to calculate the COHP or the option ONLY_H=T to calculate the real space Hamiltonian matrix only. In the tetrahedron integration method used for COHP calculation, the energy range and energy mesh points defined in the DOS category are used. The local Cartesian coordinate system can be changed for each atom. Rotations can be given with angles (EULANG=.true.) or a new orientation can be given by means of a new z-axis and an axis to define the new yz-plane (of course, this last vector should not be parallel to z). Normalization is not required. In the latter case, EULANG is set to .false. and the values of "EULER ANGLES=" are ignored. One has also the possibility to treat all orbitals listed under the bond description, or to make any linear combination of the orbitals for either the source atoms or the target atoms, or both. A typical COHP file where one is interested in the bonding between two sp^3 hybrid orbitals in the diamond structure would be:

```
SOURCE:
  ALL= .false.          EULANG= .true.
  EULER ANGLES= 0.000000 0.000000 0.000000
  ZAXIS=          0.000000 0.000000 1.000000
  YAXIS=          0.000000 1.000000 0.000000
  ORBITAL=
  + 1 (           s) * COEF.= 1
  + 2 (           py) * COEF.= 1
  + 3 (           pz) * COEF.= 1
  + 4 (           px) * COEF.= 1
TARGET:
  ALL= .false.          EULANG= .true.
  EULER ANGLES= 0.000000 0.000000 0.000000
  ZAXIS=          0.000000 0.000000 1.000000
  YAXIS=          0.000000 1.000000 0.000000
  ORBITAL=
  + 1 (           s) * COEF.= -1
  + 2 (           py) * COEF.= 1
  + 3 (           pz) * COEF.= 1
  + 4 (           px) * COEF.= 1
```

Normalization of the hybrid orbitals is not required. The sign of the coefficients depends upon the direction of the bond and upon the sign of $\pm\sqrt{\Delta}$ in the "CTRL" file.

Step 4 - COHP plotting

The program *lmcohp.run* generates an ASCII file ("COPL" file) which can be used by *gnucohp.run*. In the first lines of the file are stored:

1: the *x*-scale: ESCALE for energy with $f(N(E))=F$ or NSCALE for number of states with $f(N(E))=T$

2: the number of COHPs, number of spins, number of points, min/max for the *x*-axis, position of the Fermi level

3 \rightarrow ncohp+2: the COHP labels

The data are presented in column format with, for each point, the *x*-scale value and then the COHP and ICOHP in the following way:

((cohp(np,i,ns),icohp(np,i,ns), $i = 1, \text{ncohp}$), $ns = 1, \text{nspin}$)

Example of "COHP" file

```

NUMBER OF COHPs= 2
If ALL is .true., all the orbitals will be included else
the hybrid orbital defined with COEF. will be used. The
normalization is done automatically.
Euler angles are used to change the local coordinate system.
They are given in unit of PI (PI -> 1.0). If EULANG is false
the new system is defined with two axes.
COHP= 1
Bond between Fe1 and O : 4.09999 a.u. -> 2.16962 Angstr.
TITLE=Fe_1/O_3_tr(-1,0,0) : 2.169 Ang.
SOURCE: ATOM=Fe1 IBAS1= 1
TARGET: ATOM=O IBAS2= 3 TRANSLATIONS= -1 0 0
VECTOR= -.50000 .00000 .00000
SOURCE:
ALL= .true. EULANG= .true.
EULER ANGLES= 0.000000 0.000000 0.000000
ZAXIS= 0.000000 0.000000 1.000000
YAXIS= 0.000000 1.000000 0.000000
ORBITAL=
+ 5 ( dxy ) * COEF.= 1.000000
+ 6 ( dyz ) * COEF.= 1.000000
+ 7 ( d3z^2-1 ) * COEF.= 1.000000
+ 8 ( dxz ) * COEF.= 1.000000
+ 9 ( dx^2-y^2 ) * COEF.= 1.000000
TARGET:
ALL= .true. EULANG= .true.
EULER ANGLES= 0.000000 0.000000 0.000000
ZAXIS= 0.000000 0.000000 1.000000
YAXIS= 0.000000 1.000000 0.000000
ORBITAL=
+ 1 ( s ) * COEF.= 1.000000
+ 2 ( py ) * COEF.= 1.000000
+ 3 ( pz ) * COEF.= 1.000000
+ 4 ( px ) * COEF.= 1.000000
COHP= 2
Bond between Fe1 and O : 4.09999 a.u. -> 2.16962 Angstr.
SOURCE: ATOM=Fe1 IBAS1= 1
TARGET: ATOM=O IBAS2= 3 TRANSLATIONS= 0 -1 0
VECTOR= .00000 -.50000 .00000
TITLE=Fe_1/O_3_tr(0,-1,0) : 2.169 Ang.
SOURCE:
ALL= .true. EULANG= .true.
EULER ANGLES= 0.000000 0.000000 0.000000
ZAXIS= 0.000000 0.000000 1.000000
YAXIS= 0.000000 1.000000 0.000000
ORBITAL=
+ 5 ( dxy ) * COEF.= 1.000000
+ 6 ( dyz ) * COEF.= 1.000000
+ 7 ( d3z^2-1 ) * COEF.= 1.000000
+ 8 ( dxz ) * COEF.= 1.000000
+ 9 ( dx^2-y^2 ) * COEF.= 1.000000
TARGET:
ALL= .true. EULANG= .true.
EULER ANGLES= 0.000000 0.000000 0.000000
ZAXIS= 0.000000 0.000000 1.000000
YAXIS= 0.000000 1.000000 0.000000
ORBITAL=
+ 1 ( s ) * COEF.= 1.000000
+ 2 ( py ) * COEF.= 1.000000
+ 3 ( pz ) * COEF.= 1.000000
+ 4 ( px ) * COEF.= 1.000000

```

END