Unfolding in VASP

David Dirnberger,[†] Georg Kresse,[†,‡] Cesare Franchini,[†,¶] and Michele Reticcioli[*,†]

†*University of Vienna, Faculty of Physics and Center for Computational Materials Science, Vienna, Austria*

‡*VASP Software GmbH, Vienna, Austria*

¶*Dipartimento di Fisica e Astronomia, Università di Bologna, 40127 Bologna, Italy*

E-mail: michele.reticcioli@univie.ac.at

Eigenstate Unfolding

The analysis of the electronic properties in DFT calculations adopting large unit cells gets complicated by the shrinking of the Brillouin zone (BZ) and the consequent folding of the eigenstates in the reciprocal space. These intricate supercell states can be unfolded back into the larger BZ of the primitive cell by applying the unfolding technique. The latest version of the unfolding algorithm is currently included in a patch package for VASP (version 6.2.1)[1]. Basic features of unfolding are included in the VASP code by default: the new features included only in the patch package are marked in red.

Theoretical background. The option *LKPROJ*=.TRUE. in the INCAR files activates the calculation of the projection $P_{\boldsymbol{K}m}(\boldsymbol{k}_i)$ of the supercell eigenstates $|\boldsymbol{K}m\rangle$ on the primitive cell eigenstates $|\boldsymbol{k}n\rangle$:

$$P_{\boldsymbol{K}m}(\boldsymbol{k}_i) = \sum_{\{\boldsymbol{g}\}} |C_{m,\boldsymbol{g}+\boldsymbol{k}_i}|^2 \ , \tag{1}$$

where the eigenstates $|\boldsymbol{K}m\rangle$ are expanded in terms of a plane-wave basis set with coefficients $C_{m,\boldsymbol{K}}$; m and n denote energy band indices at vectors $\boldsymbol{K}$ and $\boldsymbol{k}_i$ in the reciprocal space of the supercell and primitive cell, respectively. This projection represents the amount of Bloch character of the states $|\boldsymbol{k}_i n\rangle$ contributing to $|\boldsymbol{K}m\rangle$, which allows for a direct connection between the reciprocal space of the supercell and the primitive cell.

Requirements. An Unfolding calculation requires:

- Usual requirements for standard band-structure calculations (e.g., WAVECAR/CHGCAR file from a previously converged self-consistent calculation obtained for the supercell).
- The POSCAR and POSCAR.prim files defining the structure of the supercell and primitive cell, respectively. Supercell and primitive cell must satisfy the equation $\mathbf{A} = \mathbf{Ma}$, where all m_{ij} elements of the transformation matrix M are integer numbers.

[1] Unfolding Patch Package for VASP available at https://github.com/QuantumMaterialsModelling/UnfoldingPatch4vasp. The associated bands4vasp toolkit is available at https://github.com/QuantumMaterialsModelling/bands4vasp.

Note: the POSCAR.prim file can be replaced by the INTMUL option in the INCAR file (option available only for the patched VASP).

- The options *LKPROJ*=.TRUE. and NCORE=1 in the INCAR file.
- A set of supercell $\boldsymbol{K}$ points in the KPOINTS file. The primitive cell $\boldsymbol{k}_i$ points connected to every $\boldsymbol{K}$ by linear combinations of reciprocal supercell lattice vectors $\boldsymbol{G}$ are automatically calculated.
  In the patched VASP, a user friendlier approach is available: if a 'p' or 'P' character is present on the 3$^{\text{rd}}$ line of the KPOINTS file, then (the internal variable KPOINT_PROJECTION_PC is set to true and) the coordinates in the KPOINTS file are interpreted as $\boldsymbol{k}$ points of the primitive cell and automatically projected on the supercell $\boldsymbol{K}$ points.

Note: A complete list of options for the Unfolding scheme is described in Section "Eigenstate Unfolding: INCAR flags".

Output files. The dedicated output files of the Unfoldig schemes are the PRJCAR file (storing the $P_{\boldsymbol{K}m}(\boldsymbol{k}_i)$ Bloch characters), and (only for the patch package) the PROCAR.prim and CONTCAR.prim optional files.

References. The implementation of the Unfolding method as included in the patch package is described in:

Electronic State Unfolding for Plane Waves: Energy Bands, Fermi Surfaces, and Spectral Functions, D. Dirnberger, G. Kresse, C. Franchini, and M. Reticcioli, *The Journal of Physical Chemistry C* 23, 12921 (2021), DOI:10.1021/acs.jpcc.1c02318. This version supports the **bands4vasp** external tool for an immediate analysis of unfolding calculations.

Core routines were developed for previous VASP versions, as described in: *Indirect-to-direct gap transition in strained and unstrained* Sn_xGe_{1-x} *alloys*, C. Eckhardt, K. Hummer, and G. Kresse, Physical Review B 89(16), 165201 (2014), DOI:10.1103/PhysRevB.89.165201.

Eigenstate Unfolding: INCAR flags

LKPROJ

LKPROJ= .TRUE. | .FALSE.

Default: *LKPROJ* = .FALSE.

Description: Activates the Unfolding scheme.

INTMUL

INTMUL=[integer array] (9 elements)

$INTMUL = m_{11}\ m_{12}\ m_{13}\ m_{21}\ m_{22}\ m_{23}\ m_{31}\ m_{32}\ m_{33}$

Default: *INTMUL* is not set.

Description: It defines the matrix elements m_{ij} of the transformation matrix **M**, used to calculate the lattice **a** of the primitive cell starting from the supercell **A**, according to **A** = **Ma** , with

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} .$$

The primitive cell lattice is printed out in the POSCAR.prim file. If the POSCAR.prim file already exists (defined by the user), the INTMUL tag is ignored.

Available only in VASP version 6.XXX?? or higher.

SLIM_PRJCAR

SLIM_PRJCAR= .TRUE. | .FALSE.

Default: *SLIM_PRJCAR* = .FALSE.

Description: If true, a lighter version of the PRJCAR file will be produced (see the description of the PRJCAR file).

Available only in VASP version 6.XXX?? or higher.

KPROJ_THRESHOLD

KPROJ_THRESHOLD= [real]

Default: *KPROJ_THRESHOLD* is not set

Description: If this variable is set, the PRJCAR file will contain only values of the Bloch characters $P_{\boldsymbol{K}m}(\boldsymbol{k}_i)$ >*KPROJ_THRESHOLD*. This option is useful to reduce the size of output files.

Eigenstate Unfolding: Input/Output files

PRJCAR

The PRJCAR file stores the output of the Unfodling scheme (written for LKPROJ=.TRUE.).

It has the following format:

- The header contains 4 matrices:

  i) The basis vectors of the primitive cell in reciprocal space, stored by row.

  ii) The basis vectors of the super cell in reciprocal space, stored by row.

  iii) Transformation Matrix $\mathbf{M}$ in real space $\mathbf{A} = \mathbf{Ma}$

  iv) Transformation Matrix $\mathbf{M}'$ in reciprocal space $\mathbf{B} = (\mathbf{M}^{-1})^T\mathbf{b} \;\; \Rightarrow \;\; \mathbf{M}' = (\mathbf{M}^{-1})^T$

- The header contains also the values set for the 'KPOINT_PROJECTION_PC', 'SLIM_PRJCAR' and 'KPROJ_THRESHOLD' variables, and the list of k-points of the primitive cell.

  Example:

```
#KPOINT_PROJECTION_PC=.TRUE.   #SLIM_PRJCAR=.TRUE.  #KPROJ_THRESHOLD=.FALSE.
number of k-points in IBZ of POSCAR.prim:     5
```

kpt	b1	b2	b3	weight
1	0.500000000000	0.500000000000	0.500000000000	1
2	0.375000000000	0.375000000000	0.375000000000	1
3	0.250000000000	0.250000000000	0.250000000000	1
4	0.125000000000	0.125000000000	0.125000000000	1
5	0.000000000000	0.000000000000	0.000000000000	1

- The core information of PRJCAR are the $P_{\boldsymbol{K}m}(\boldsymbol{k}_i)$ Bloch character values. These values are grouped per spin, $\boldsymbol{k}_i$ point and energy band index, and reported per every

$\boldsymbol{K}$ point separately. If SLIM_PRJCAR=.TRUE. then the Bloch characters are reported only for $\boldsymbol{K}$ points connected to the $\boldsymbol{k}_i$ point by a linear combination of $\boldsymbol{G}$ vectors of the supercell reciprocal lattice. If SLIM_PRJCAR=.TRUE. is used in combination with the "p" option in the KPOINTS file, then only the $\boldsymbol{k}_i$ points in the KPOINTS file are considered [Recommended Standard Setup].

Example (for the Recommended Standard Setup):

```
spin component:   1
k-point (associated with POSCAR):     1  vkpt:     1.0000000     0.5000000     0.50
band:     1  energy:    -3.8648719  KAPPAsum:  0.1068561E+01
   1:  0.1068561E+01
band:     2  energy:    -2.0080562  KAPPAsum:  0.1655755E+00
   1:  0.1655755E+00
```

PROCAR.prim

The *PROCAR.prim* file is generated, if *LKPROJ* is activated, *LORBIT* >= 10 and the **k**-point conversion "p" flag in the KPOINTS file is set. It retains a PROCAR-like format, with the following two differences:

- The **k**-points represent the primitive cell $\boldsymbol{k}_i$ points.
- The line with the band number, the energy and the occupation of the electronic state contains also the Bloch character $P_{\boldsymbol{K}m}(\boldsymbol{k}_i)$ (identified by the string 'Unfold proj.').

Example of PROCAR.prim file:

```
k-point     1 :   -0.01471685 0.01471685 0.01471685     weight = 0.01000000

band     1 # energy  -46.77371041 # occ.  2.00000000 Unfold proj.:   0.5994299E-01
```

```
ion       s     py     pz     px    dxy    dyz    dz2    dxz  x2-y2    tot
    1  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
```

POSCAR.prim

The *POSCAR.prim* file contains information on the primitive cell lattice structure and, optionally, its atomic basis. It retains the format of POSCAR files.

Importantly, supercell and primitive cell must satisfy the equation $\mathbf{A} = \mathbf{Ma}$, where all m_{ij} elements of the transformation matrix M are integer numbers. If this equation is not satisfied at least within a precision of $\sim 10^{-7}$, VASP calculates a new primitive cell that satisfies the equation and it resembles the cell defined by the user: in this case, this new primitive cell is used in the calculation, it is printed out in the CONTCAR.prim file and a warning gets shown in the standard output.

CONTCAR.prim

The *CONTCAR.prim* file describes the lattice of the primitive cell, which is used for the unfolding calculation. It has the usual POSCAR-like format. It is generated if the INTMUL option is used, or if the POSCAR.prim file defined by the user does not satisfy the equation $\mathbf{A} = \mathbf{Ma}$, where all m_{ij} elements of the transformation matrix M are integer numbers, for the chosen supercell $\mathbf{A}$ and primitive cell $\mathbf{a}$.

Example: B Doped cubic diamond Si

Task

Unfolding calculation for cubic diamond silicon doped with 25% Boron. For standard band-structure calculation of the undoped Cubic Diamond Si see example Cd Si.

Input

POSCAR

```
 doped cubic diamond Si3B1
    5.175
  0.0    1.0    1.0
  0.5    0.0    0.5
  0.5    0.5    0.0
 Si  B
  3  1
 direct
  0.000  0.000  0.000
  0.125  0.250  0.250
  0.625  0.250  0.250
  0.500  0.000  0.000
```

INCAR

```
System = doped Cd Si3B1
NPAR = 1
ICHARG = 11
ISMEAR = 0; SIGMA = 0.1;
LWAVE = .FALSE.
```

```
LORBIT = 11


#Unfolding options
LKPROJ = .TRUE.
INTMUL = 2 0 0 0 1 0 0 0 1
SLIM_PRJCAR = .TRUE.
```

KPOINTS

```
k-points for bandstructure L-G-X-U K-G
 40
p
line
reciprocal
  0.50000  0.50000  0.50000
  0.00000  0.00000  0.00000

  0.00000  0.00000  0.00000
  0.00000  0.50000  0.50000

  0.00000  0.50000  0.50000
  0.25000  0.62500  0.62500

  0.37500  0.7500   0.37500
  0.00000  0.00000  0.00000
```

Calculation

For an accurate effective band structure, it is recommended to do the following calculation steps:

- Volume relaxation for the lattice constant as in example fcc_Si or one takes a lattice constant from experimental data.
- Relaxation of the internal coordinates of the structure as in example Cd_Si_relaxation.
- Perform a static (NSW=0, IBRION=-1) self consistent calculation to obtain the CHGCAR and WAVECAR files as in example Fcc_Si_DOS.
- Finally, perform the Unfolding calculation. For very large systems and/or for dense sampling of the reciprocal space, it is recommended to divide the path of reciprocal space points in sub-paths and to perform Unfolding calculations separately for every single sub-path.

For the relaxation and the self consistent calculation we used the following KPOINTS file:

```
k-points mesh
 0
Monkhorst-Pack
  5  11  11
  0   0   0
```

The first basisvector B_1 of the doped super cell in reciprocal coordinates has half the length of the other two. To obtain a more or less equally spaced mesh we set also the k-point number in that direction to more or less the half.

Unfolding

For very large systems and/or for dense sampling of the reciprocal space, it is recommended to divide the path of reciprocal space points in sub-paths and to perform Unfolding calculations

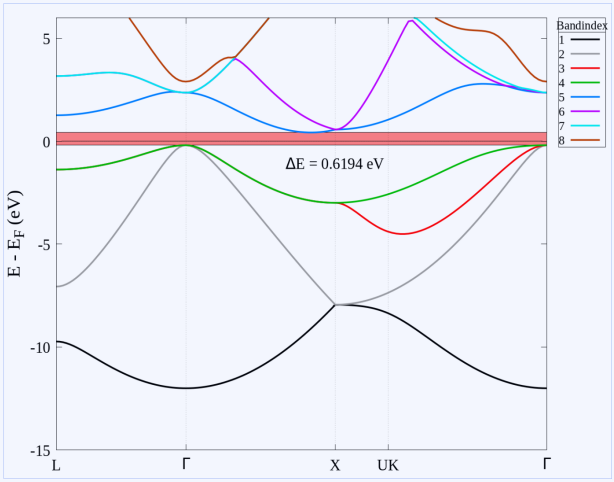
separately for every single sub-path. For every single calculation one needs to prepare a directory with the following files:

- CHGCAR (from the self-consistent calculation)
- WAVECAR (from the self-consistent calculation)
- INCAR
- POSCAR (needs to be the same as used for the self-consistent run)
- KPOINTS
- POTCAR

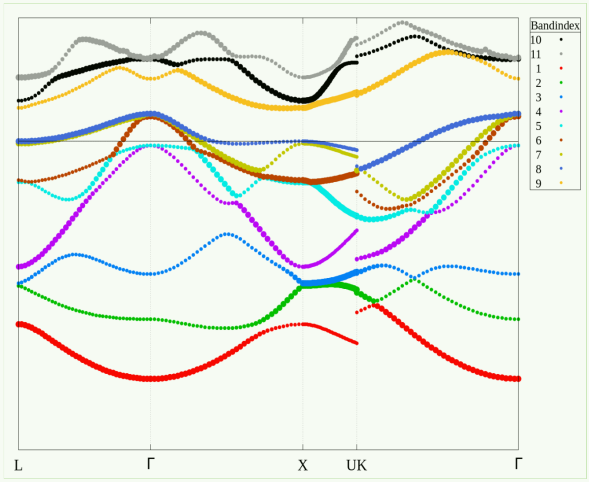
Setup of input files, analysis of output files and production of graphs can be done with the help of the external **bands4vasp** tool.

Effective Band structure

The output of the Unfolding calculation, as obtained by the **bands4vasp** tool is shown in the Figure.



Pure Si diamond



B doped Si diamond