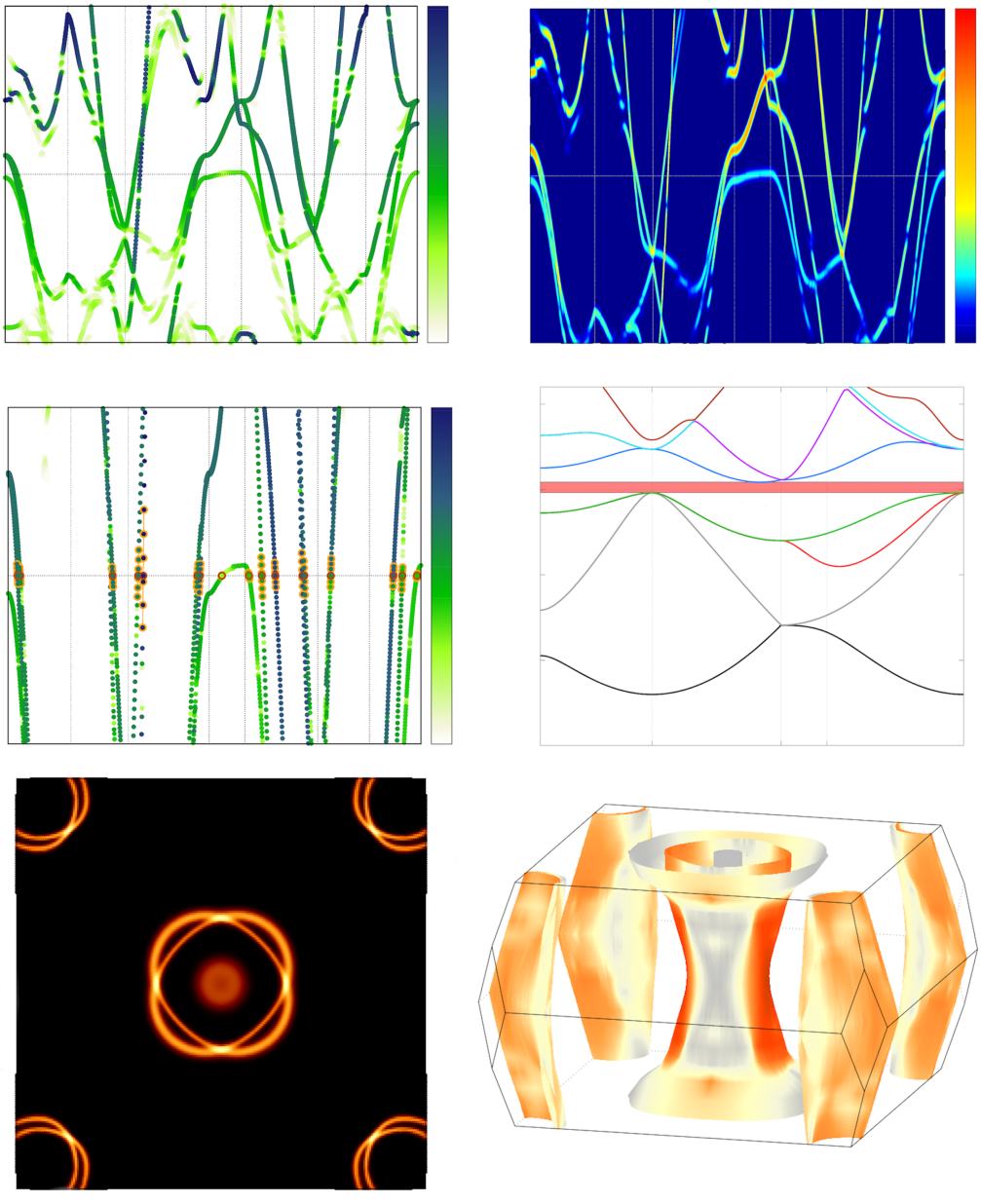
Manual for the **bands4vasp** post-processing package

David Dirnberger

July 30, 2025



Contents

1 About bands4vasp

The bands4vasp post-processing package is exclusively build for the analysis and visualisation of band structure and especially unfolding calculations from VASP. Bands4vasp is visualizing the band structure or the effective band structure (EBS) of static calculations, it can calculate the spectral function of this band structure or of the associated Fermi surface. Bands4vasp can also visualize the nlm- and site projected wave function character of each orbital and it provides a plot with the bandindex colour-coded band structure. But the core part of bands4vasp is an algorithm which automatically determines the Fermi vectors of the calculated structure. Because of the huge variety of band structures, this algorithm doesn't find allways all Fermi vectors on the first run. For such cases bands4vasp provides several parameters which helps to identify all bands and facilitate thereby the calculation of all Fermi vectors. There is a Fermi surface mode, which automatically project this Fermi vectors onto the surface, where the VASP calculations were done. There are also some pre-processing routines, for preparing the VASP files for this kind of Fermi surface calculation. Bands4vasp is written in FORTRAN, it uses Gnuplot for the visualisation and a Bash-environment brings all together.

2 The Algorithm

First bands4vasp reads in all data inbetween a given energy interval and a Bloch character (if present) bigger than the given threshold. These values can be defined in the INPAR file (see section INPAR - The bands4vasp input file). The next step is to evaluate the spectral function (1) and write it in *banddata.specfun.dat*, if **SPECFUN** is activated in INPAR.

$$A(\boldsymbol{k}, E) = \sum_m P_{\boldsymbol{K}m}(\boldsymbol{k})\delta(E_m - E) \tag{1}$$

If this is done, it compresses the data depending on the given parameters in the INPAR file. For a specific state n with the energy E_n the algorythm picks all states inbetween the energy interval $[E_n - EDIF; E_n + EDIF]$ and they will be compressed to a new weighted averaged state (2), than the procedure starts again with the next energy state. Let $\{E\}_{i=1,\dots,d}$ be a set of energy states inbetween the interval $[E_1 - EDIF; E_1 + EDIF]$.

$$\tilde{E} = \frac{\sum_i P_i \cdot E_i}{\sum_i P_i} \tag{2}$$

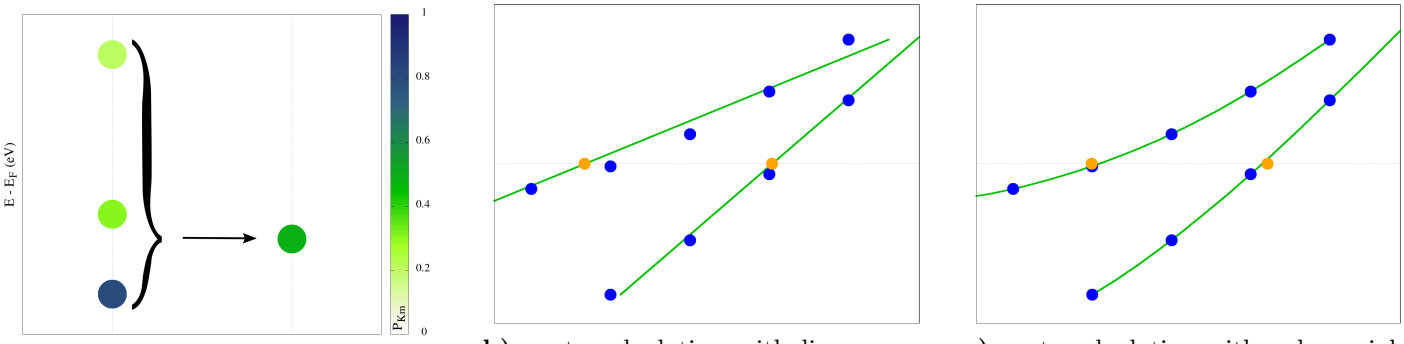
Where $\tilde{E}$ is the weighted averaged energy of the band center of mass, E_i is the energy and P_i the Bloch character of state i. Also the orbital- and Bloch character have to be weighted with the following relations:

$$\tilde{P} = \frac{\sum_i P_i(E_\Delta - \mid \tilde{E} - E_i \mid)}{\sum_i (E_\Delta - \mid \tilde{E} - E_i \mid)} \tag{3}$$

$$\tilde{Oc} = \frac{\sum_i Oc_i(E_\Delta - \mid \tilde{E} - E_i \mid)}{\sum_i (E_\Delta - \mid \tilde{E} - E_i \mid)} \tag{4}$$

$$E_\Delta = max(\{E\}_i) - min(\{E\}_i) \tag{5}$$

Here E_Δ is the maximal energy interval of all the energy states which will be compressed to one band center of mass. These compression of the eigenstates is schematically shown in the sketch 1a).
At this point bands4vasp writes the raw data and as well the processed data in the files:
Bandstructure, *EBSbloch*, *EBSorbit*, and *Bandindexplot*.
If **ROOTSCALC** is set to true (default), bands4vasp scans the bands for roots, which are associated with Fermi vectors. To find the Fermi roots there are many different parameters to distinguish the bands and indentify the states which belong to the same band. How to use these paramters is described in section 3.3. Once it founds a band crossing the Fermi level, the intersection point is either calculated with linear regression (**LPOLY**=.FALSE.) or with polynomial interpolation (**LPOLY**=.TRUE.) and the regula falsi method. The linear regression yield to reliable results for localy straight bands and the roots can be calculated exactly with an analytical algorithm [see figure 1 b)]. The polynomial interpolation is a good choice, if the bands are very sinuous around Fermi. The degree of the polynomial is given by the number of points which are included in the calculation $\Rightarrow\ 2 \cdot NPOINTS - 1$. If you set for example **NPOINTS**=3, there will be 3 points on each side of Fermi level or 6 in total. This will lead



a) weighted average of the eigenstates

b) roots calculation with linear regression

c) roots calculation with polynomial interpolation and the regula falsi method

Figure 1: Schematic representations of the methods used for the automatic calculation of the Fermi vector

to a polynomial of degree 5. To calculate the roots, bands4vasp uses the regula falsi method [see figure 1 c)]. This is in comparison to the exact analytical solution of the linear regression a disadvantage. Nevertheless it always depends on the shape of the bands and the number of included points near the Fermi level, which method is the best choice, but the most reliable method is the linear regression with a small amount of considered points and therefore it is the default method. The orbital- and Bloch characters are averaged with the same principle as discribed for the compression [see Eq. (2),(3),(4)]. These calculated Fermi roots are transformed to a 3-dimensional Fermi vector and written in the *FERMIROOTS.dat* file.

Bands4vasp comes with the possibility for an automatic determination of the Fermi surface. If Fermi surface mode is activated with −−fermi option , it will first figure out the orthonormal basis vectors of the surface, where the calcualations were done. The basis set is used for the transformation of the Fermi vectors into terms of the surface. Now it checks if the points of symmetry (**SYMPOINT1**/**SYMPOINT2**) are defined in INPAR and are part of that surface. After all coordinates are transformed into terms of the surface and reflected, if the point(s) of symmetry are set, all data are written in the file *Fermisurface.dat* and as well the data of the Fermi surface spectral function in the file *Fermisurface.specfun.dat*, if activated.
In the case that no Fermi roots were found, bands4vasp calculates the energy gap around the Fermi level. This is done by taking the energy distance from the lowest energy over the Fermi level to the highest energy under Fermi. In that case the files *Fermisurface.dat* and *Fermisurface.specfun.dat* and as well the *Fermisurface** image are not created.

3 In- and output files

3.1 VASP-files (input)

There are 3 typs of files bands4vasp is able to read:

- The **PRJCAR** file for unfolding calculations in VASP.
- The **PROCAR** file for nlm- and site projected wave function character of each eigenvalue.
- The **PROCAR.prim** file, analogous to PROCAR, but for unfolding calculations.

Additionally bands4vasp needs the Fermi energy and the lattice of the structure. The Fermi energy can be defined either in the INPAR file or extract from a OUTCAR file. For unfolding calculations the lattice is extracted from PRJCAR file. If it is not an unfolding calculation the lattice will extracted from the POSCAR file.
Note: Unfolding calculations needs to be done in the line mode and for multiple sized super cells with the 'p' mode. As it is shown in this example KPONTS file:

```
automatic radial sampling of k-points
100
p
line
rec
#makepath
0 0 0
0.5 0 0
0 0.5 0
```

For more information see the vasp wiki.

3.2 bands4vasp output files

Bands4vasp provides a lot of different information, which are derived from VASP files.

Datafiles - The datafiles are stored in './bands4vasp_data/', except *FERMIROOTS.dat*.

- *FERMIROOTS.dat* contains all information of the evaluated Fermi vectors and some calculation specific information. This file is stored in the execution directory.
- *banddata.dat* contains the raw data of the whole band structure.
- *banddata.slim.dat* contains the processed data of the band structure.
- *banddata.specfun.dat* contains all information of the spectral function from the band structure.
- *Fermisurface.dat* contains all information of the evaluated Fermi surface.
- *Fermisurface.spec.dat* contains all information of the evaluated Fermi surface spectral function.
- *autognuplot_bands4vasp.gnu* is the gnuplot file for generating all images.

Images - The images are stored in './bands4vasp_img/', except the image of interest.

- *Bandsturcture.** shows the bands inbetween the energy interval defined by **EDELTA1**.
- *EBSbloch.** shows the bands inbetween the energy interval defined by **EDELTA1**/**EDELTA2** with respect to Bloch character.
- *EBSorbit.** shows the bands inbetween the energy interval defined by **EDELTA1** with respect to orbital character and if present a variable pointsize proportional to the Bloch character.
- *Bandstructure.spec./EBSbloch.spec.* shows the spectral function inbetween the energy interval defined by **EDELTA1** with respect to Bloch character.
- *Bandindexplot* shows the bandstructure with colour-coded bandindex occuring in the VASP files.
- *Fermisurface.spec.* shows the Fermi surface spectral function.
- *Fermisurface* shows the derived Fermi surface from the Fermi vectors.
- *Fermisurface_bloch* shows the derived Fermi surface with the calculated avaraged Bloch characters.
- *Fermisurface_orbit* shows the derived Fermi surface with the calculated avaraged orbital characters.

For spin polarized calculations the extansion .spin1. and .spin2. are added to the filenames.

**These images have also a processed version including technical details (for example, the interpolation calculated to determine the Fermi wave vectors). This is indicated by the extension .processed.*

3.3 INPAR - The bands4vasp input file

bands4vasp is controlled via a set of parameters stored in a file named **INPAR**. If no `INPAR` file is present in the working directory at run time, `bands4vasp` falls back to its built-in defaults. A template containing these default settings is supplied in the installation directory and can be copied to the current directory via:

```
$ b4vasp --inpar
```

For transparency, every run writes the parameters actually used to an **OUTPAR** file in the working directory. Note that certain parameters only take effect when a specific VASP file type is selected; these cases are flagged with the following three symbols:

- • for PRJCAR files
- ○ for PROCAR files
- ⊙ for PROCAR.prim files

General control parameters

• ○ ⊙ **EFERMI** - sets the source for the Fermi energy, default is from ./OUTCAR. It is possible to set a directory which includes the OUTCAR file (e.g. /home/calculation/), to set the Fermi energy explicitly (e.g. 0.4), or by setting the keyword 'bands' to take the Fermi energy from the OUTCAR files of each banddata respectively.

• ○ ⊙ **EDELTA1** – Energy window (in eV) used for the raw-data plots *EBSbloch*, *EBSorbit*, *EBSbloch.spec*, *Bandstructure.spec*, and *Bandindexplot*.
Single value ΔE: interval is symmetric around E_F, i.e. $[-\Delta E, +\Delta E]$.
Two values E_{min} E_{max}: lower and upper bounds are set individually (relative to E_F).
If the bounds are given in reverse order ($E_{min} > E_{max}$) they are swapped internally.
Typical range: ±0.1–5 eV.

• ○ ⊙ **EDELTA2** – Energy window (in eV) applied to the *processed* plots. Only eigenstates within this range are considered when computing the Fermi vectors. Typical choices mirror those of **EDELTA1**; however, narrowing **EDELTA2** directly tightens the set of states entering the Fermi vector determination, potentially reducing computational load while focusing on the most relevant energy window.

•⊙ **EDIF** – Energy–diffusion tolerance (in eV) used during the effective–band–structure construction. For each k-point, all eigenstates whose energies differ by less than **EDIF** are merged and represented as a single effective state in the processed data (see Section 2). A larger **EDIF** reduces the number of distinct effective states but can smear fine spectral features; a smaller value preserves detailed structure at the expense of a more *fragmented* effective band structure. Typical range: 0.001–0.10 eV.

•⊙ **BAVERAGE** – Governs how Bloch characters are combined when multiple eigenstates contribute to a single effective state. This choice influences the entire effective band construction and therefore propagates to all *processed* plots and to the calculation of Fermi vectors.
`.TRUE.` : the *weighted average* of the Bloch character is stored, yielding a normalised value suitable for direct comparison between bands.
`.FALSE.` : the individual Bloch characters are *summed*, retaining information on the total spectral weight per state.

○⊙ **OAVERAGE** – Governs how orbital characters are combined when multiple eigenstates contribute to a single effective state. This choice influences the entire effective band construction and therefore propagates to all *processed* plots and to the calculation of Fermi vectors.
`.TRUE.` : the *weighted average* of the orbital character is stored, yielding a normalised value suitable for direct comparison between bands.
`.FALSE.` : the individual orbital characters are *summed*, retaining information on the total spectral weight per state.

•⊙ **BLOCH_THRESHOLD** – Minimum Bloch-character criterion. Eigenstates whose Bloch character falls below this threshold are discarded from all subsequent analyses and plots.

•⊙ **DBLOCH** – Maximum permissible difference in Bloch character for neighbouring eigenstates to be treated as belonging to the *same* effective band during the construction of the *effective* band structure. If the absolute Bloch-character contrast between two proximate states does not exceed **DBLOCH**, the states are merged into a single track.

A larger **DBLOCH** value yields smoother, less fragmented effective bands but risks over-merging distinct branches; a smaller value preserves fine band splitting at the expense of fragmented tracks and potentially noisier Fermi vector extraction. Typical range: 0.05–0.30, depending on the material's band dispersion and the desired balance between continuity and fidelity.

○⊙ **ODISTINCT** – Maximum allowable difference in *orbital* character between neighbouring eigenstates for them to be regarded as the *same* effective band during the generation of the effective band structure. For each orbital, two proximate states are merged into one trajectory only if their orbital weights differ by less than **ODISTINCT**. A larger **ODISTINCT** value yields smoother, less fragmented effective bands but risks over-merging distinct branches; a smaller value preserves fine band splitting at the expense of fragmented tracks and potentially noisier Fermi vector extraction. Recommended range: 0.05–0.50 , adjustable according to the material's orbital hybridisation and the desired trade-off between smoothness and fidelity.

• ○ ⊙ **EGAP** – defines the maximal allowed energy difference between states at neighboring k-points. Two energy states are considered part of the same band if their energy difference is below this threshold. EGAP is used during the identification of Fermi crossings and the construction of fitted Fermi paths. Additionally, it serves as a scaling base for the **DGRAD** parameter, which controls the allowed deviation from the extrapolated energy trend. A smaller EGAP leads to stricter energy matching, while a larger value allows greater flexibility in band tracking across the Brillouin zone.

• ○ ⊙ **GRADIENTD** – sets the tolerance for deviations from the local energy gradient during the construction of fitted Fermi paths. It is used as a scaling factor that defines an acceptable energy window around each interpolated energy state based on the estimated gradient and the parameter **EGAP**. If the next candidate point lies outside this tolerance window, it is rejected. Lower values of GRADIENTD enforce stricter gradient continuity, while higher values allow smoother tolerance for band curvatures and numerical deviations.

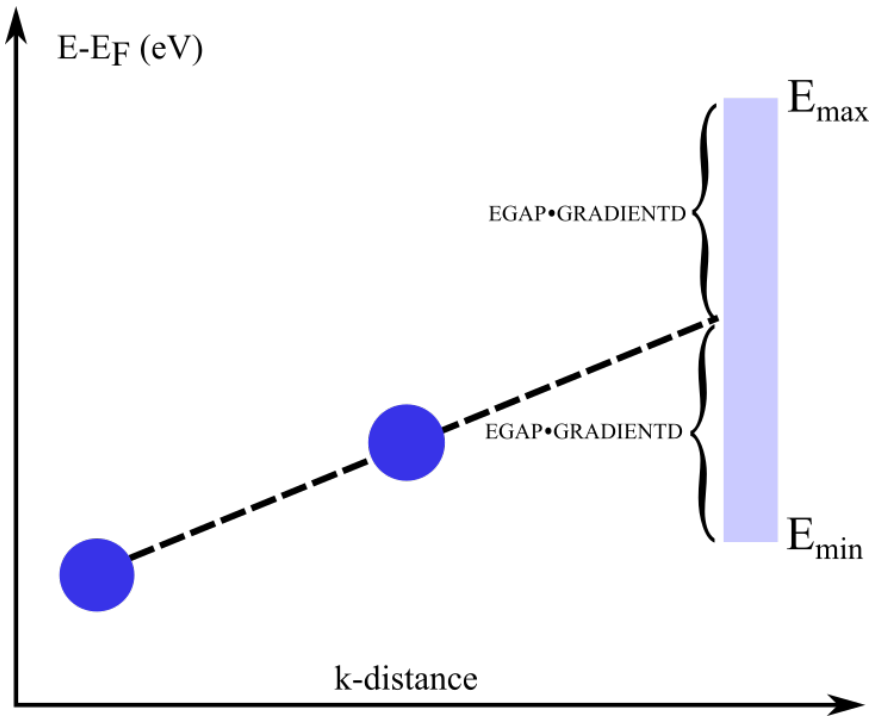


Figure 2: Sketch for demonstration of the meaning from **GRADIENTD**

• ○ ⊙ **NPOINTS** – Sets the number of eigenstates *above* and *below* the Fermi level that are taken into account when constructing each Fermi vector. For every k-point, a total of $2 \times$**NPOINTS** states (symmetrically

distributed around E_F) are examined to determine crossings and interpolate the effective band structure. Larger values provide a broader energetic context and can improve robustness in systems with dense band manifolds. Smaller values focus strictly on the immediate vicinity of E_F. Recommended values: 1-10.

• ○ ⊙ **LPOLY** - If LPOLY is set to .TRUE. a polynomial interpolation with the degree 2·NPOINTS will be used for calculating the Fermi vector. If it is set to .FALSE. (default), the coordinates of the Fermi vectors are calculated by linear regression.

• ○ ⊙ **REGULAPREC** - sets the accuracy for the Regula falsi method, which is used for the polynomial interpolation.

○⊙ **PLOTORB** - sets the orbital number for the orbital plots in order of appearence in PROCAR[.prim] file. If it's set to 0 (default) all orbitals will be considerd and visualized in *EBSorbit_ALL*. For any negative value bands4vasp calculates the total amount of all orbitals (*EBSorbit_tot*).

• ○ ⊙ **BNDDIFF** – Controls whether band crossings between states with *different* band indices are allowed to merge into a single trajectory when constructing Fermi vectors. `.TRUE.` (*default*) : crossings originating from bands with different indices are treated as belonging to one band.
`.FALSE.` : only crossings between states sharing the *same* band index are merged; distinct indices remain separate.

A quick visual check of the *Bandindexplot* is recommended to confirm whether the chosen setting yields the desired continuity.

• **KAPPANORM** - If KAPPANORM is set to .TRUE. the Bloch character for each band and K-point from the super cell will be normalized respectively. This works only with PRJCAR files.

○⊙ **SELECTION** - select a specific ion and the associated orbital characters in order of appearence in PROCAR[.prim] file. By giving two numbers seperated by a minus 'n-m', bands4vasp will take ion n up to m and sum the orbital characters together. If the value is 0 (default) all ions are considered.
Note: The Bloch character in FERMIROOTS.dat or in the images always refer to the complete system. If one ion or a group of ions are selected, the *EBSorbit_tot* or the *tot* window in the *EBSorbit_ALL* image and the *tot* orbital character in the FERMIROOTS.dat file representing the Bloch character induced bei the selected group of ions.

• ○ ⊙ **SKIPKPOINT** - if SKIPKPOINT is set to one integer n, the first n k-points will be ignored. By setting two numbers seperated by a minus 'n-m', bands4vasp will ignore k-points n up to m.

• ○ ⊙ **ROOTSCALC** - if ROOTSCALC is set to .FALSE. no Fermi vectors will be calculated.

The following parameters are for ALL filetypes.

Fermisurface

The parameters in the following section are only considered if a fermisurface calculation is done, by passing the option −−fermi (see also section 4.2).

▷ **SYMPOINT1/SYMPOINT2** - These are the symmetry points for the resulting *Fermisurface** plots. If *SYMPOINT1* is set to a 3d-vector, where the values have to be seperated by one space ("SYMPOINT1 = 0.0 0.0 0.0"), the calculated Fermi vectors will be copied to the rotaded positions at rotation angles $[90°; 180°; 270°]$. If *SYMPOINT2* is set to a vector as well, the Fermi vectors will axial reflected with the mirror axis defined by *SYMPOINT1* and *SYMPOINT2*. The symmetry points will only be accept, if they are on the Fermi surface.

▷ **SYMREC** - if SYMREC is set to .TRUE., the coordinates for SYMPOINT1 and SYMPOINT2 are treated in reciprocal coordinates (default). If SYMREC is set to .FALSE., bands4vasp consideres cartesion coordinates.

▷ **FSUREDGEVEC** – controls the output format of the Fermi surface edge vectors. Accepted values are `none`, `reciprocal` (default), and `cartesian`. If set to `none`, no vectors are printed. If set to `reciprocal`, vectors are shown in lattice coordinates. If set to `cartesian`, the 3D Cartesian coordinates (k_x, k_y, k_z) are printed.

▷ **FSURTICS** – if set to `.TRUE.` (default), the in-plane axes and tick marks are displayed in the Fermi surface plots. For constant-kz slices, this corresponds to projected Cartesian coordinates (k_x, k_y). If set to `.FALSE.`, no ticks or coordinate axes are shown.

▷ **FGRID** – controls the number of dashed grid lines in the Fermi surface plots. Accepts integer values from 0 to 100.
`FGRID = 0` (default) disables the grid entirely.
`FGRID = n > 0` draws `n` evenly spaced dashed lines along both the x and y directions, dividing the plot into `n+1` segments per axis.
For example: `FGRID = 1` draws center lines (halves), `FGRID = 2` divides into thirds, etc.

Spectral function

▷ **SPECFUN** - if SPECFUN is set to .TRUE., the spectral function of the bandstructure/fermisurface is calculated and visualized (default = .FALSE.).

▷ **SIGMA** – Energy broadening parameter (in eV) that substitutes the Dirac delta function in the spectral-function evaluation (see Eq. 1). A smaller **SIGMA** sharpens spectral features but may amplify numerical noise; a larger value smooths the spectrum at the cost of energy resolution. Choose **SIGMA** in conjunction with **SPECDELTA** so that the sampling step adequately resolves the broadened peaks. Typical values lie in the range 0.01–0.20 eV for most materials.

▷ **SPECDELTA** – Determines the energy resolution (in eV) for the spectral function evaluation. Smaller values of **SPECDELTA** increase the number of energy states included for each k-point, resulting in higher resolution, increased computation time, and larger **.specfun** files (see Section 3.2).

▷ **SLIMSPEC** – If set to `.TRUE.` (default), only eigenstates with a Bloch character greater than zero are considered for output, reducing data size and computation time. No interpolation is performed in this mode.
If set to `.FALSE.`, all energy states sampled at intervals specified by **SPECDELTA** are stored and plotted, regardless of their Bloch character. This results in significantly larger **.specfun** files and longer computation times. In this mode, the spectral function of the Fermi surface is additionally visualized using gnuplot's built-in `pm3d` interpolation (for visualization only).

Plot specific options

▷ **MAKEPLOTS** - if MAKEPLOTS is set to .FALSE. no plots will be created.

▷ **FILEFORMAT** - bands4vasp supports a choice of different file format of the plots. Possible formats are: png (default), pdf, eps.

▷ **BANDINDEXPLOT** - if set to .TRUE. the Bandindexplot will be created (see section 3.2).

▷ **LEAVEPLOTDATA** - if set to .FALSE. the directory './bands4vasp_data/' will be removed after bands4vasp is finished.

Visual parameters

▷ **PATHPOINTS** – Assigns a label to every path stored in the VASP output files `PRJCAR` or `PROCAR[.prim]`. Labels are given as blank-separated tokens in the order in which the paths appear.

A leading `/` before a token requests the corresponding Greek letter (e.g. `/G` → Γ).

Example `PATHPOINTS=/G M A K L` produces the labels Γ, M, A, K, L for the successive path points read from the VASP results.

▷ **LFITPOINTS** - if set to .TRUE. the fitpoints from the linear regression/polynomial interpolation are shown in the plots. By default they are only shown in the processed versions of the plots.

▷ **LLINES** - if set to .TRUE. the graph from the linear regression/polynomial interpolation are shown in the plots. By default they are only shown in the processed versions of the plots.

▷ **LROOTS** - if set to .TRUE. the Fermi roots from the linear regression/polynomial interpolation are shown in the plots. By default they are only shown in the processed versions of the plots. If there are no Fermi roots found, the energy interval of the band gap is show in the plots, only if LROOTS is not set to .FALSE..

▷ **PSFAC** - this factor changes the pointssize in the plots.
⇒ $pointsize = original\_pointsize \times PSFAC$

▷ **BCOLOURS** - is defined by two gnuplot colors seperated by a blank. The colors define the color palette for the Bloch plots. For a list of all accesable colornames open the terminal and enter:
$gnuplot -e 'show palette colornames'

▷ **OCOLOURS** - same as BCOLOURS, but for the orbital plots.

▷ **BACKCOLOUR** - sets the background color of all plots, except the spectral function plots.

Pre-processing parameter

▷ **PREPREVIEW** - if set to .TRUE., an interactive preview of the selected pre-processing method (–pre-lines, –pre-circle, or –pre-surface) is shown, provided that a graphical display is available (i.e. the $DISPLAY environment variable is set). On remote systems, this can be enabled via an X11-forwarding SSH session (e.g., by connecting with ssh -X). If no display is available, a PNG image will be generated instead. If set to .FALSE., no preview will be shown at all. This option requires a working Python environment with matplotlib installed.

4 Getting Started

For the installation *bands4vasp_v< version >.tar.gz* and *install.sh* are needed and can be downloaded here. The installation starts with an execution of the *install.sh* file.

```
$ source ./install.sh
```

4.1 Usage

```
$ b4vasp [OPTION] ... [file]
```

To start a calculation there have to be the following files:

- PROCAR.prim, PROCAR or PRJCAR :: For the data of the bandstructure.
- PRJCAR or POSCAR :: For the information of the lattice. When analysing unfolding calculations it is necessary to have at least one PRJCAR file.
- OUTCAR :: If the Fermi energy is not given in the INPAR file, bands4vasp takes it from OUTCAR.

There are 3 different ways of using bands4vasp:

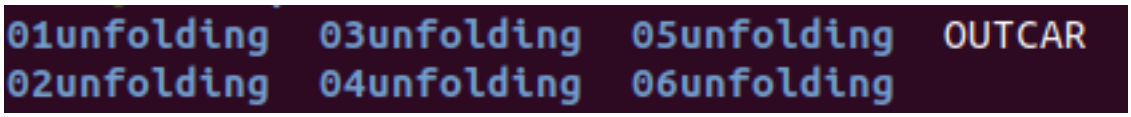
- **Single path** calculations can be done in the directory of the VASP files by entering the command 'b4vasp',

  ```
  $ cd [folder-path]
  $ b4vasp
  ```

  or one can pass the directory.

  ```
  $ b4vasp [folder-path]
  ```

- **Multi path** calculations are possible if there are numbered folders including the VASP files from the band calculation. To pass this kind of structure a %-sign represents the number in the foldername.


  ```
  $ b4vasp %unfolding
  ```

  This will pass all directories with numbers from 1-1000 (also with leading zeros) and the name 'unfolding' $\Rightarrow$ $1unfolding$, $01unfolding$, $001unfolding$, $2unfolding$, ...

- **Fermisurface** calculations need the same structure as the multi path calculations and additionally with option flag –fermi.

  ```
  $ b4vasp --fermi %unfolding
  ```

  To get a reasonable result, the line calculations performed with VASP need to be on one plane, but the geometrical arrangement of the lines is not important.

Note: If only a folder structure is passed, bands4vasp prefers PROCAR.prim files. If they are not present it looks for PRJCAR files or at least PROCAR files. For specifying the VASP filetype it is possible to pass the filetype directly.

```
$ b4vasp  %unfolding/PROCAR.prim
```

```
$ b4vasp  %unfolding/PRJCAR
```

```
$ b4vasp  %unfolding/PROCAR
```

4.2 Options

Beside the −−fermi option is also the option -rs [−−readsave]. The -rs option reads the raw data obtained from a previous execution of bands4vasp (avoiding to read VASP files, which could be time consuming), thus, it is useful for testing, visualization purposes and adjustment of the parameters to get all Fermi vectors. It can be used in the following way:

- First run a calculation in the normal mode.

  ```
  $ b4vasp %unfolding
  ```

- Now one can modify the INPAR file and run the calculation again, but this time without a path specification, with the optional –fermi option and the option -rs.

  ```
  $ b4vasp --fermi -rs
  ```

CAUTION: There are some parameter, which can not be reset in the −−readsave mode. EDELTA1 and EDELTA2 can only set to intervals which are included in the original EDELTA1 interval. SKIPKPOINT, SELECTION, EFERMI, KAPPANORM and BLOCH_THRESHOLD don't have any influence on the −−readsave mode.

There aro also some pre-processing options available. For all the following pre-processing procedures there has to be a folder, which includes all files for the VASP calculation. The section in the KPOINTS file, where one specifies the coordinates of the k-points, is replaced by the flag '#makepath', followed by coordinates depending on the type of sampling. The entries above the '#makepath' flag are copied in every KPOINTS file followed by the calcuclated coordinates. A KPOINTS file for a radial sampling with the first line from center (0, 0, 0) to (0.5, 0, 0) and the last line from (0, 0, 0) to (0, 0.5, 0) in reciprocal coordinates is shown below:

```
automatic radial sampling of k-points
100
p
line
rec
#makepath
0 0 0
0.5 0 0
0 0.5 0
```

More information about unfolding method with vasp or the KPOINTS file can be found on the vasp wiki page. There are 3 typs of sampling methods:

- **--pre-lines $1 $2** :: prepares a structure of line calculations for each pair of k-points in KPOINTS file after the flag '#makepath'. The following two informations need to be passed:
  $1 ⇒ The directory for the prepared VASP files $2 ⇒ The name for the new created multi-directory

- **--pre-circles $1 $2 $3** :: prepares radial centered sampling. The specification of the radial sampling is done by three coordinates defined in the KPOINTS file, after the flag '#makepath': The first coordinate is the center of the circle. The second coordinate defines the first line of the sampling, starting from the center. The third coordinate defines the last line of the sampling. The sampling is done in a mathematical positiv sense. The following 3 informations need to be passed:
  $1 ⇒ directory for the prepared VASP files $2 ⇒ number of equidistant points on the circle $3 ⇒ name of the new created multidirectory

- **--pre-surface $1 $2 $3** :: prepares a equidistant sampling within a rectangle defined by the coordinates given in the KPOINTS file after the flag '#makepath': The first coordinate is the corner of the rectangle where the calculation starts. The second coordinate defines with the first one, the direction and the length of the line calculations. With the third coordinate the rectangle is defined and as well the translation direction of the line sampling. The following 3 informations need to be passed:
  $1 ⇒ directory for the prepared VASP files $2 ⇒ number of equidistant lines on the surface $3 ⇒ name of the new created multidirectory

With the options `--help` and `--info` the most important informations about bands4vasp is printed.

References

[1] *bands4vasp GitHub page*
https://github.com/QuantumMaterialsModelling/bands4vasp

[2] *Unfolding Patch for VASP GitHub page*
https://github.com/QuantumMaterialsModelling/UnfoldingPatch4vasp

[3] David Dirnberger; Georg Kresse; Cesare Franchini; and Michele Reticcioli. *Electronic State Unfolding for Plane Waves: Energy Bands, Fermi Surfaces, and Spectral Functions.* The journal of physical chemistry C. https://pubs.acs.org/doi/10.1021/acs.jpcc.1c02318

[4] David Dirnberger. *Electronic state unfolding for plane waves, an automatized computational implementation* https://utheses.univie.ac.at/detail/69219#

[5] *VASP wiki for the unfolding method*
https://www.vasp.at/wiki/index.php/LKPROJ

[6] *VASP wiki for the KPOINTS file*
https://www.vasp.at/wiki/index.php/KPOINTS

[7] *VASP wiki for the spd- and site projection*
https://www.vasp.at/wiki/index.php/LORBIT