Electronic Density of States and Energy Bands Plotter (dosbandsplot.py) USER GUIDE

Andrés Megías Toledano
University of Extremadura

Version 1.3
October 2020

1. File dosbandsplot.py

The script dosbandsplot.py, written in Python 3, can represent electronic bands and densities of states (DOS) calculated by the software Quantum ESPRESSO. The script needs the modules os, sys, ast, copy, glob, pathlib and configparser, from the Python's standard library, and also the libraries NumPy and Matplotlib.

In order to run the code for a file of bands or densities of states (or both), it is necessary to specify some parameters in a configuration file. For running the code one has to write the following in the command line, being in the folder of the file dosbandsplot.py:

```
python3 dosbandsplot.py <path>  .
```

The argument <path> is the path of the configuration file; if one does not especify it, the path ./config.txt will be used, that is, the code will search in the current folder (./) the configuration file with name config.txt. If the file dosbandsplot.py is not inside the current folder, but in a folder with path <folder>, one should write <folder>/dosbandsplot.py instead of dosbandsplot.py. Moreover, if the file dosbandsplot.py is included in a folder of executables files, the term python3 can be removed from the command line, without needing to specify that folder.

2. Configuration file

The configuration file is a plain text document with the writing formating style defined by the configparser module. Here one can specify the parameters of the script dosbandsplot.py by defining different variables. These variables are divided in four distinct blocks: the preamble, the common options, the bands mode options, and the density of states mode options.

In order to specify each of this sections, one has to write in a single line, in capital letters and enclosed in brackets the title of each section. Then one can write the definitions of the variables, on different lines, using a colon (:) or the equals sign (=); for example: variable = value. The name of the variables is case-insensitive. For the logical variables, the possible values are yes/no, on/off and true/false, being case-insensitive as well. For the rest, the possible values will depend on the characteristics of each variable. In the case of a text variable, it value will have to be written between simple quotation marks ('). If the definition a variable is not written, it default value will be used (see section 3), which in the case of logical variables is always the negative value (no/off/false).

2.1. Preamble

The title of this block is PREAMBLE. Here the working mode of the code is selected, that is, it is declared if the code will analyse electronic bands or densities of states (or both). The logical variables bands mode and density of states mode determine the working mode.[1] For example, to activate only the bands mode one can write:

```
[PREAMBLE]
bands mode = yes
density of states mode = no
```

As it has been said, if a logical variable is not defined, it will be used the negative value (no). Hence, in the previous example the line density of states mode = no could have been omitted. To sum up, the only two variables of this block are:

- **bands mode.** Determines whether the bands mode is used or not (yes/no).
- **density of states mode.** Determines whether the density of states mode is used or not (yes/no).

2.2. Common options

The title of this sections is COMMON OPTIONS. The variables defined in this block are the ones which are common to both working modes. They consist of some graphic options and parameters, and variables related to energy. Bellow are all the variables of this block:

- **save image.** To save or not an image of the created figure (yes/no); if so, another variable can be declared:
  - **image format.** Format of the created image (for instance, 'png' or 'pdf').
- **font.** Typographic font used in the plots.
- **font size.** Size of the font used in the plots, in points (pt).
- **figure dimensions.** Figure/s dimensions (width × height), in inches (").
- **frame width.** Width of the lines of the plot frame, in points (pt).
- **plot scale.** Scale of the plot symbols and lines with respect to the default sizes.
- **zero energy.** Value used as zero energy, if specified; if so, the rest of the variables that refer to the energy must be expressed in the new resulting scale.
- **energy lines.** Energy values to be marked with a dashed line; for example: energy lines = [0, 2].
- **energy limits.** Limits for the energy axis, in the correspondent units. One of the limits can be unespecified by writing two simple quotation marks (''), in which case the minimun value (for the lower limit) or the maximum (for the upper) will be used; for

1. Note the presence of blank spaces in the variable names.

example: `energy limits = [3, '']`. For any of the limits it can be specified a value equal to the maximum or the minimum plus or minus a quantity `x` by writing `'val+x'` or `'val-x'`, with `val = min, max`; for example: `energy limits = ['max-6', '']`. Plus, numbers can also be written between simple quotation marks; for example: `energy limits = ['3', '']`.

- **energy label.** Label of the energy axis.
- **energy units.** Units used for energy.
- **energy range priority.** In case that the bands mode is also activated and some of the energy limits (or both) is not specified, this variable determines if the energy range which will be used to take the limit (or limits) will be the range of the energy bands (`'bands'`), that of the densities of states (`'density of states'`), the intersection of both (`'intersection'`) or their union (`'union'`).

2.3. Bands mode options

The title of this sections is BANDS MODE. In the bands mode the code shows a plot of the electronic bands storaged in a data file generated by Quantum ESPRESSO. Moreover, an interactive mode is activated, in which the coordinates of the plot points will be shown on the terminal by clicking on them. Below are the variables of this block:

- **folder.** path of the folder containing the bands file.
- **file.** name of the bands file.
- **wavevector points.** Name and coordinates (in the proper units) of the limits of the different tracks of the reciprocal spaces where the bands are calculatede, using for each point a Python dictionary structure, and separating them by commas; for example: `wavevector points = {'Γ': (0, 0, 0)}, {'L': (0.5, 0.5, 0.5)}`.
- **wavenumber offset.** This value is used in the determination of the values of the wavenumber of the original data which correspond to the points in the reciprocal space specified in the variable `wavevector points` (excluding the initial and final ones); by default it has a value of 0, but in some cases it should be used a value of 1.[2]
- **number of empty bands.** Number of empty bands in the data file. If it is specified, the occupied and empty bands will be plotted with different colors, and the script will calculate the minimum of the conduction band and the maximum of the valence band.
- **plot title.** Title of the bands plot.
- **wavevector label.** Label of the reciprocal space axis.
- **plot colors.** Array with the three colors used to plot the bands; the first one is the color used if the variable `number of empty bands` is not declared, and the second and

2. This deppends on the Quantum ESPRESSO calculation. It is convenient to firstly try without the comma and, if it is observed in the bands plot that the delimiting points are not correct, use a value of 1.

third ones are used for the occupied and empty bands, respectively, if number of empty bands is declared.

2.4. Density of states mode options

The title of this section is DENSITY OF STATES MODE. In the density of states mode the code shows a plot of the density of states of the projections specified in the configuration file, selected from the files generated by Quantum ESPRESSO. Moreover, those selections can be exported to a new data file. In case the bands mode is also activated, the code will plot on the same window the electronic bands and the densities of states. Below are the variables used in this block.

- **folder.** Path of the folder containing the density of states files.
- **files prefix.** Prefix of the density of states files (name prior to .pdos_atm…).
- **export densities of states.** To create or not a text file with the plotted densities of states (yes/no); here the the limits used for the plot are not used.
- **total density of states.** To plot or not the total density of states file (yes/no).
- **spin-orbit.** Presence or not of the spin-orbit effect (yes/no).
- **energy zoom.** To create or not a second plot centered in an energy value (yes/no); if so, the two following variable can be specified:
  - **energy zoom parameters.** Parameters of the energy zoom, expresed in a Python dictionary; in particular, the central energy ('center') and the amplitude ('amplitude'). For example: `energy zoom parameters = {'center': 0, 'amplitude': 2}`.
- **density of states limits.** Limits for the density of states axis, in the correspondent units. One of the limits can be unespecified by writing two simple quotation marks (''), in which case the minimun value (for the lower limit) or the maximum (for the upper) will be used; for example: `density of states limits = [10, '']`. It is possible to allow the script to choose a proper value for the upper limit by writing 'auto'; for example: `density of states limits = [10, 'auto']`. Plus, numbers can also be written between simple quotation marks; for example: `density of states limits = ['10', '']`.
- **projections.** Partial projections of the density of states wanted to plot. The way of declaring this variables is explaining with detail in section 2.5.
- **plot title.** Title of the density of states plot.
- **density of states label.** Label of the density of states axis.
- **density of states units.** Units used for the density of states.
- **plot colors.** Array with the colors used to represent the partial densities of states, in the order stated by the variable projections. If there are more densities of states than specified colors, these will be repeated.

2.5. Selection of the partial densities of states

The variable projections determines the projections which will be selected from the original data files. It is a Python tuple with several subelements, which firstly has a total of N Python lists. Each of these lists has various elements: it has a total of M tuples, and it also has a text variable that we will call mode, which will be explained later. Hence, the structure is:

$$\texttt{projections} = (\begin{bmatrix} \texttt{parameters}_{11}, & \cdots & \texttt{parameters}_{1M}, & \texttt{mode} \\ \vdots & \ddots & \vdots & \vdots \\ \texttt{parameters}_{N1}, & \cdots & \texttt{parameters}_{NM}, & \texttt{mode} \end{bmatrix}) \quad ,$$

where $\texttt{parameters}_{ij}$, with $i = 1, \ldots, N$ and $j = 1, \ldots, M$, are the lists of atomic parameters of the projections, that are actually Python tuples. These parameters refer to the classification of the Quantum ESPRESSO files of densities of states.

The name of these files has a certain format, which varies slightly depending on the existence of spin-orbit coupling. If not, the format is:

<prefix>.pdos_atm#<atom-num>(<element>)_wfc#<function-num>(<orbital>) .

Here <prefix> is the text which identifies the Quantum ESPRESSO calculation; <atom-num> is the number of the atom according to its ordination on the crystal lattice; <element> is the element of that atom; <function-num> is the number of the wave function (or functions) associated to this data file with respect to all the functions calculated for this specific atom of the lattice; and <orbital> is the type of orbital of this wave function (or functions). For example:

Si_pDOS.dat.pdos_atm#1(Si)_wfc#2(p) .

In case of having the spin-orbit effect, the format would be:

<prefix>.pdos_atm#<atom-num>(<element>)_wfc#<function-num>(<orbital>_j<momentum>) .

Here <momentum> would be the total angular momentun number. For example:

Si_pDOS.dat.pdos_atm#1(Si)_wfc#2(p_j0.5) .

Each of these data files contains several columns. The first one are the energy values; the second one is the partial density of states corresponding to the wave functions which has the parameters listed in the file name; and the rest of columns are the partial densities of states separated according to the orientation quantum number, related to the angular momentum (orbital angular momentum in case of not having the spin-orbit effect, total angular momentum in caso of having that effect). Therefore, the second column is the sum of the partial densities of states of the following columns.

Going back to the variable projections, on each list of atomic parameters of the projection one can specify the atom number (<atom-num>), the element (<element>), the orbital (<orbital>), the total angular momentum number (<momentum>), if it is the case, and besides that, the value of the orientation quantum number (redefined from 1 to the number of values following the order of the data file), which we will call <orientation>. Hence, each list

of atomic parameters woukd have the following structure, without and with spin-orbit effect, respectively:

```
parameters = (<atom-num>, <element>, <orbital>, <orientation>)   ,
parameters = (<atom-num>, <element>, <orbital>, <momentum>, <orientation>)   .
```
[3]

For example:

```
parameters = (1, 'Si', 'p', 1)   ,
parameters = (1, 'Si', 'p', 0.5, 1)   .
```

Once knowing this, here are some more options. If one does not want to specify a field value, two simple quotations marks must be written (''). For example:

```
projections = ([('', 'Se', 'p', ''), mode])   .
```

If one wants to specify several values of a field, an abbreviated writing style can be used, writing the different values separated by a plus sign (+). For example:

```
projections = ([('', 'Se', 's+p', ''), mode])   ,
```

which would be equivalent to:

```
projections = ([('', 'Se', 's', ''), ('', 'Se', 'p', ''), mode])   .
```

In the case of a numeric field, the expression should be enclosed in single quotation marks ('); for instance, '1+2'.

As for the variable mode, it determines the mode of selection of the projections. It has two possible values: mode = 'ind' (individual mode) and mode = 'sum' (sum mode). In the individual mode, each of the lists of atomic parameters (parameters$_1$, ..., parameters$_N$) will be plotted individually, with diferent colors, and if the densities of states are exported to a text file, each of them will be written in a different column. Plus, if the abbreviated writing style has been used, each of the possible combination of states will be plot separatedly. On the contrary, in the sum mode, whether the abbreviate writing style is used or more than one list of atomic parameters are used (parameters$_1$, ..., parameters$_N$), all of these states will be added up, plotting that sum and exporting it in just one column. In the case of introducing only one list of parameters without using the abbreviated writing style, the mode is irrelevant.

Lastly, for each element added to the variable projections, the resulting set of densities of states (depending on the number of lists of atomic parameters used (parameters$_1$, ..., parameters$_N$) and the mode used (mode = 'ind' or mode = 'sum')) will be plotted separatedly. For example:

```
projections = ([('', 'Se', 's+p', ''), 'ind'])
```

would be equivalent to:

3. The elements of the list containing letters must be written between simple quotation marks (').

```
projections = ([('', 'Se', 's', ''), 'ind'],
               [('', 'Se', 'p', ''), 'ind'])   ,
```

and would also be equivalent to:

```
projections = ([('', 'Se', 's', ''), ('', 'Se', 'p', ''), 'ind'])   .
```

3. Default variable values

Below are the default values of the variables of dosbandsplot.py, that is, the values that they will take if they are not declared in the configuration file.

```
[PREAMBLE]
bands mode = no
density of states mode = no
[COMMON OPTIONS]
save image = no
image format = 'jpg'
font = 'DejaVu Sans'
font size = 14
figure dimensions = [7, 7]
frame width = 1.2
plot scale = 1
zero energy = 0
energy lines = []
energy limits = ['', '']
energy label = 'energy (eV)'
energy units = 'eV'
energy range priority = 'union'
[BANDS MODE]
folder = './'
file = 'bands.dat'
wavevector points = ()
wavenumber offset = 0
number of empty bands = ''
plot title = 'Energy bands'
wavevector label = 'wavevector'
plot colors = [(0.5, 0.5, 0.5), (0.3, 0.3, 0.8), (0.8, 0.3, 0.3)]
[DENSITY OF STATES MODE]
folder = './'
files prefix = 'pDOS.dat'
export densities of states = no
total density of states = no
spin-orbit = no
energy zoom = no
```

```
energy zoom parameters = {'center': 0, 'amplitude': 3}
density of states limits = ['', '']
projections = ()
plot title = 'Partial densities of states'
density of states label = 'density of states (/eV)'
density of states units = '/eV'
plot colors = ['chocolate', 'darkgreen', 'deepskyblue', 'tab:pink', 'tab:brown',
               'blueviolet', 'orangered', 'tab:olive', 'cadetblue']
```

4. Example of configuration file

Below is an example of configuration file in both bands mode and density of states mode, in addition to the resulting plots (Figure 1).

```
[PREAMBLE]
bands mode = yes
density of states mode = yes
[COMMON OPTIONS]
save image = yes
energy limits = [-1, '']
[BANDS MODE]
folder = './bands'
file = 'bands_Si.dat'
wavevector points = {'X': (0, 0.5, 0.5)}, {'Γ': (0, 0, 0)}, {'L': (0.5, 0.5, 0.5)}
number of empty bands = 30
[DENSITY OF STATES MODE]
folder = './pDOS'
files prefix = 'pDOS_Si.dat'
export densities of states = yes
total density of states = yes
spin-orbit = yes
density of states limits = [0, 'auto']
projections = ([('', '', 's+p', '', ''), 'ind'],
               [('', '', 's+p', '', ''), 'sum'])
```

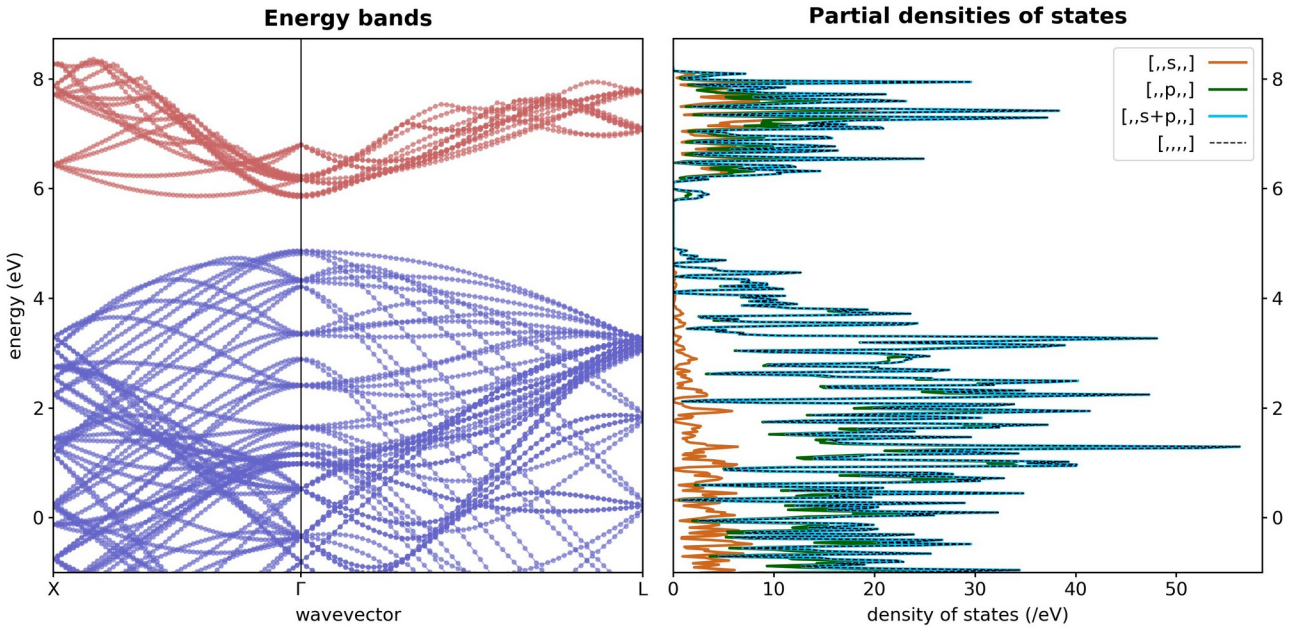

Figure 1. Plots generated by the script `dosbandsplot.py` in the example case. *Left*: Plot of the energy bands. *Right*: Plot of the selected densities of states.

5. Links of interest

The following links may be of interest:

- **Quantum ESPRESSO.**
  https://www.quantum-espresso.org/
- **Python.**
  https://www.python.org/ .
- **Python's standard library.**
  https://docs.python.org/3/library/
- **NumPy.**
  https://numpy.org/
- **Matplotlib.**
  https://matplotlib.org/