

# 1. Overview

The **Fat\_Band\_QE.m** code is designed to process the output files from Quantum ESPRESSO (QE) PDOS calculations and generate fat band structure plots. It produces two types of plots:

- **Color-Variation Plot:** Orbital contributions are shown with varying colors and the point size is proportional to the PDOS value.
- **Fixed-Color Plot:** Each orbital is assigned a fixed color while the point size varies with the PDOS value.

This tool is useful for visualizing the orbital character of bands, helping researchers interpret electronic structure calculations.

Make an input file with the required field and flags and save it as text file. Run the **Fat\_Band\_QE** code. Example: `Fat_Band_QE(input_file.txt)`

## 1.1 Prerequisites

Before running the code, ensure that you have completed the following QE calculations:

### 1. Self-consistent Field (SCF) Calculation:

- Command example:

Linux: `pw.x < Ga2O3.scf.in > Ga2O3.scf.out`

Windows: `pw.exe < Ga2O3.scf.in > Ga2O3.scf.out`

### 2. Band Structure Calculation:

- Command example:

Linux: `pw.x < Ga2O3.band.in > Ga2O3.band.out`

Windows: `pw.exe < Ga2O3.band.in > Ga2O3.band.out`

### 3. Projected Density of States (PDOS) Calculation:

- Command example:

Linux: `projwfc.x < Ga2O3.pdos.in > Ga2O3.pdos.out`

Windows: `projwfc.exe < Ga2O3.pdos.in > Ga2O3.pdos.out`

## 1.2. File-Naming:

Every file must be named with your material's prefix, followed by the corresponding extension for the calculation type. For example, if your prefix is Ga2O3, the files should be named exactly as follows:

- **SCF Calculation:**

- Input file: `Ga2O3.scf.in`
- Output file: `Ga2O3.scf.out`

- **Band Calculation:**

- Input file: `Ga2O3.band.in`
- Output file: `Ga2O3.band.out`

- **PDOS Calculation:**

- Input file: `Ga2O3.pdos.in`

- Output file: Ga203.pdos.out

## 2. Input Format

There are two field `WRITE_PDOS` and `PLOT_PDOS`. It is possible to run both field at one input file or run them separately. Make sure you run `WRITE_PDOS` before `PLOT_PDOS`. Each field end with `/` like quantum espresso input file. Each flag ends with a comma(,). It is a must. Command can be written after `!` mark. Matrix must be written in single line. Each row is separated by semicolon (;) and full matrix should be enclosed by square bracket []. See `Want_States` & `xy_limit` usage. Cell type input also must be written in one line and each cell is separated by ; or , and full cell is enclosed by curly bracket {}.

```
&WRITE_PDOS
prefix="Ga203",           ! Mandatory: Material prefix used in your QE files
inputdir=".",             ! Mandatory: Directory containing the QE output files
Nspin_Type="spin_up_down", ! Mandatory
Delimiter="band", ! Mandatory: Specify the input from which kpoints is being read.
outdir=".",               ! Optional: Output directory for generated files
Overwrite=1,              ! Optional: 0 or 1 (whether to overwrite existing files)
Efermi=1,                 ! Optional: 0 or 1 (1 fermi is shifted to 0eV.0 fermi is kept as it is.)
/

&PLOT_PDOS
prefix="Ga203",           ! Mandatory if &WRITE_PDOS field is not present
inputdir="./Ga203.fat_band", ! Mandatory if &WRITE_PDOS field is not present;
Delimiter="band",         ! Specify from which input kpoints is being read.
Want_States=[1 2 9 10;3 6 11 0; 4 7 0 0;5 8 13 14], ! Mandatory
Legend_Details={"\itC_s"; "\itB_py"; "\itB_px"; "\itB_pz"}, ! Optional:
outdir=".",               ! Optional: Output directory for plots
Band_Limit=[1 202],       ! Optional: Specify band limits as [min_band, max_band]
Color_Matrix=[0 0 1; 1 0 0; 0 1 0; 1 1 0], ! Colors assigned to each orbital
Sorting_Type="ascend",     ! Optional: "ascend" or "descend" for sorting bands
Weight_Multiplier = 150,   ! Optional:
Justify_Zero = 0.02,       ! Optional: Small value adjustment for zero weights
Draw_Fermi = 1,            ! Optional: 0 or 1 (whether to draw the Fermi level)
Aspect_Ratio = [0.6 1 1],  ! Optional: Aspect ratio of the figure
xy_limit=[NaN, NaN; NaN, NaN], ! Optional: X and Y axis limits (use NaN for auto-
scaling)
Interpoint=100,           ! Optional: Positive integer to adjust interpolation points
Figure_Spec={"Times", "bold", 12}, ! Optional: Figure font specifications
/
```

### 2.1 Flag Details:

#### &WRITE\_PDOS Flags

This block is responsible for reading and processing the QE output files to extract PDOS information. Each flag here plays a specific role:

- **prefix (mandatroy)**  
*Definition:* A string that identifies your material.

*Usage:* All input and output files must start with this prefix (e.g., Ga2O3).

*Example:*

`prefix="Ga2O3",`

- **inputdir (mandatory)**

*Definition:* Directory path where the QE files are stored.

*Usage:* Set this to the location of your `prefix.scf.in/out`, `prefix.band.in/out`, and `prefix.pdos.in/out` files.

*Example:* `inputdir="."`, indicates the current directory.

- **Nspin\_Type (mandatory)**

*Definition:* Specifies the spin treatment used in your calculation.

*Usage:* Acceptable values are:

- `"no_soc"` for non-spin-polarized calculations, pseudopotential type: scalar relativistic and `nspin=1`, `lspinorb=.false`, `noncolin=.false` in QE pw calculation
- `"soc"` when spin-orbit coupling is included, pseudopotential type: full relativistic and `nspin=4`, `lspinorb=.true`, `noncolin=.true` in QE pw calculation
- `"spin_up_down"` when spin-polarized calculation, LSDA (magnetization along z axis), `nspin=2`

- **Delimiter (mandatory)**

*Definition:* Identifies from which input files the kpoints are being read.

*Usage:* This flag helps the code determine where the kpoints tick are being used in the plot. See 2.2 for more details. Options include:

- `"scf"` for self-consistent field calculations,
- `"nscf"` for non-self-consistent field,
- `"band"` for band structure calculations. *Example:*  
`Delimiter="band",`

- **Outdir (optional)**

*Default:* `outdir="."`,

*Definition:* Directory path where the processed PDOS data (typically in .MAT format) will be stored.

*Usage:* It's optional; if not specified, the current directory is used.

*Example:* `outdir="."`,

- **Overwrite (optional)**

*Default:* `Overwrite=0`,

*Definition:* Controls whether existing output files should be overwritten.

*Usage:* Use 1 to allow overwriting and 0 to prevent it.

*Example:* `Overwrite=1`,

- **Efermi (optional)**

*Default:* `Efermi=1`,

*Definition:* Determines whether the Fermi energy is to be shifted to 0eV or not.

*Usage:* Set to 1 if you want to see the Fermi energy at 0eV, or 0 otherwise.

*Example:* Efermi=1,

## &PLOT\_PDOS Field Flags

This block is dedicated to generating the fat band plots using the processed data. Each flag customizes a different aspect of the plotting procedure:

- **prefix (mandatroy)**  
*Definition:* Identifies your material and should match the one in the &WRITE\_PDOS block.  
*Usage:* It is mandatory if the &WRITE\_PDOS block is not present in the input file.  
*Example:* prefix="Ga2O3",
- **inputdir**  
*Definition:* The directory where the .MAT files (produced by &WRITE\_PDOS) are stored.  
*Usage:* Often, this will be the same as the outdir specified in the &WRITE\_PDOS block, appended with a subdirectory named <prefix>.fat\_band.  
*Example:* inputdir="./Ga2O3.fat\_band",
- **Delimiter (mandatroy)**  
*Definition:* Same as &WRITE\_PDOS field.  
*Usage:* Use the same options as in the &WRITE\_PDOS block: "scf", "nscf", or "band".  
*Example:* Delimiter="band",
- **Want\_States (mandatroy)**  
*Definition:* A matrix that selects specific orbital or state indices to be plotted.  
*Usage:*
  - Each group separated by ; corresponds to a different orbital or group of orbitals.
  - The values indicate the state indices to be visualized.
  - Zero padding is required if the number of states in each orbital group is not the same.
  - Should be written in one single line. *Example:*  
Want\_States=[1 2 9 10;3 6 11 0; 4 7 0 0;5 8 13 14],
- **Legend\_Details (opional)**  
*Default:* Legend\_Details={"A"; "B";  
"C"; 'Location'; 'bestoutside'; 'Times'; 'bold';10},  
*Definition:* A cell array of strings that provides labels for each orbital in the plot legend.  
*Usage:*
  - One can use LaTeX formatting for styling (e.g., italics).
  - It is optional; if omitted, default legend labels {"A"; "B"; "C"} are used. Number of legend labels are equal to number of orbital groups used in Want\_States.
  - Legend position, font name, formatting, size can also be defined. *Example:*  
Legend\_Details={"\itC\_s"; "\itB\_py";  
"\itB\_px"; "\itB\_pz"; 'Location'; 'bestoutside'; 'Times'; 'bold';10},  
Or  
Legend\_Details={"\itC\_s"; "\itB\_py"; "\itB\_px"; "\itB\_pz"; 'Position'; [1  
0.5 0.5]; 'Times'; 'bold';10},
- **outdir (opional)**

*Default:* `outdir="."`

*Definition:* Specifies the output directory where the generated plots will be saved.

*Usage:* It is optional; if not specified, plots are saved in the current directory. *Example:* `outdir="."`,

- **Band\_Limit (optional)**

*Default:* `Band_Limit=[1 nbnd]`,

*Definition:* An optional two-element vector defining the range of bands to plot.

*Usage:*

- The first element is the minimum band number, and the second is the maximum band number you want to see in the plot.

*Example:* `Band_Limit=[1 202]`,

- **Color\_Matrix (optional)**

*Default:* `Color_Matrix=hsv(no of orbital groups in Want_States)`

*Definition:* A matrix where each row specifies an RGB color for the corresponding orbital.

*Usage:*

- The number of rows should match the number of orbital groups in `Want_States`.
- Customize these values to change the appearance of the plot.

*Example:*

`Color_Matrix=[0 0 1; 1 0 0; 0 1 0; 1 1 0]`,

- **Sorting\_Type (optional)**

*Default:* `Sorting_Type="ascend"`,

*Definition:* Determines the order in which bands are sorted for plotting.

*Usage:*

- Acceptable values are "ascend" for ascending order and "descend" for descending order. *Example:* `Sorting_Type="ascend"`

- **Weight\_Multiplier (optional)**

*Default:* `Weight_Multiplier=100`,

*Definition:* A scaling factor applied to the PDOS values to adjust the point sizes in the plot.

*Usage:*

- It should be a natural number. *Example:* `Weight_Multiplier = 150`,

- **Justify\_Zero (optional)**

*Default:* `Justify_Zero = 0.0001`,

*Definition:* A small adjustment factor used to modify zero or near-zero PDOS weights.

*Usage:*

- This ensures that even very low PDOS contributions are visually represented.

*Example:* `Justify_Zero = 0.02`,

- **Draw\_Fermi (optional)**

*Default:* `Draw_Fermi = 0`,

*Definition:* Indicates whether a horizontal line at the Fermi energy should be drawn on the plot.

*Usage:*

- Set to 1 to draw the Fermi level, or 0 to omit it. *Example:* `Draw_Fermi = 1`,

- **xy\_limit (optional)**

*Default:* `xy_limit = [1 nbnd; minimum_energy_of_band  
maximum_energy_of_band],`

*Definition:* Specifies manual limits for the x-axis and y-axis.

*Usage:*

- It is a 2×2 matrix: the first row for the x-axis [`xmin`, `xmax`]. `xmin` & `xmax` are positive integers. `xmin` can be as low as 1. `xmax` can be as high as `nbnd` and the second row for the y-axis [`ymin`, `ymax`]. `ymin` and `ymax` are energy range. Can be any real number.
- Use NaN for any value you wish to auto-scale. *Example:*  
`xy_limit=[NaN, NaN; NaN, NaN],`

- **Interpoint (optional)**

*Default:* `Interpoint = 0,`

*Definition:* Determines the number of interpolation points used for interpolating kpoints in the plot.

*Usage:*

- If not defined or kept as `Interpoint = 0`, then no interpolation of kpoints.
- If defined as any positive integers then kpoints are interpolated to give a smoother curve. A higher number can lead to smoother curves. *Example:* `Interpoint=100,`

- **Figure\_Spec (optional)**

*Default:* `Figure_Spec={"Times"; "bold"; 12},`

*Definition:* Customizes the appearance of the plot's text by specifying font attributes.

*Usage:*

- Typically includes the font name, style, and size in a cell array. *Example:*  
`Figure_Spec={"Times"; "bold"; 12},`

- **Aspect\_Ratio (optional)**

*Default:* `Aspect_Ratio = [1 1 1],`

*Definition:* Customized the size of the plot.

*Usage:*

- Defined as vector. *Example:*  
`Aspect_Ratio = [0.5 1 1],`

## 2.2 Some comments on Kpoints definition in prefix.scf/nscf/band.in file

Below is a revised version with additional clarity and detail:

The code is designed to read k-points from any of the supported input types. It supports every k-point format that Quantum ESPRESSO (QE) recognizes.

- **SCF Files:**

In an SCF input file, k-points are usually generated automatically. If you set the delimiter to "scf", the plot will display only two k-point ticks: one for the first point (labeled A) and one for the last point (labeled B). Note that using `Delimiter="scf"` is not recommended due to its limited detail.

- **PDOS Calculations:**

When calculating the projected density of states (PDOS):

- If the PDOS is generated after a calculation with `calculation="nscf"`, then use `Delimiter="nscf"`,

- If the PDOS is generated after a calculation with `calculation="bands"`, then use `Delimiter="band"`,
- **Custom K-Points Naming:**
  - If the k-points are defined using the `tpiba_b` method, the tick names are read directly from the `band.in` file.
  - If the k-points are defined with the `crystal_b` method, the symmetry point names are set by default to A, B, C, etc. However, you can override these default names by adding a comment with the desired symmetry name.

**Example for crystal\_b type k-points:**

```
K_POINTS {crystal_b}
9
0.0000 0.0000 0.0000 10 ! Γ
0.2662 0.2662 0.0000 1 ! C
-0.2662 0.7338 0.0000 10 ! C2
-0.5000 0.5000 0.0000 10 ! Y2
-0.5000 0.5000 0.5000 10 ! M2
-0.2580 0.7419 0.0000 1 ! D
0.2580 0.2580 0.0000 10 ! D2
0.0000 0.0000 0.5000 10 ! A
0.0000 0.5000 0.5000 10 ! L2
```

The same approach applies for other crystal-type k-point definitions. In summary, the code offers flexibility in handling various k-point inputs from QE and allows for custom labeling based on the input file type or user comments.