Example: energy slices calculation

This file is intended to detail the calculation presented in Sections 4.3.2 of our *BinPo* manuscript. Importantly, we remind you that any *BinPo* component looks for the input parameters at command-line at first, whereas the omitted and more advanced parameters are set by default from the configuration file. Keeping this in mind, we will show how the configuration files are set for each case and what to type from command-line. For a description of all the parameters in the configuration files, see *~/config_files/help_config.md*. Parameters modifiable through the command-line can be checked by means of help command as:

$ python BP-component.py -h

where *BP-component.py* is any *BinPo* component.

Firstly, we will show how the *energy_slices.yaml* is configured.

```
---
ENERGY_SLICES :
        identifier : "sto_01"
        sqrt_kgrid_numbers : 400
        kbox_factor : 0.6
        kbox_shift : [0.0,0.0]
        win_energy_calc : 0.04
        batches : 100
        energy_cut : 0.0
        outfile : "default"
...
```

Now, we will compute an energy slice on a previous SC-calculation under the "run2" identifier, so:

$ python BP-energy_slices.py -id run2 -ec 0.0 -nk 300 -ba 60 -bf 0.7

The energy cut (*ec*) is set to 0.0, indicating that we are computing just the Fermi surface. A k-grid (*nk*) of 300 x 300 points will be used and it will be split in 60 batches (*ba*). The kbox factor parameter (*bf*) is to enlarge or reduce the mapped BZ1. When running this component, we will see the following details:

```
DETAILS :
        Identifier : run2
        Surface : STO(100)
        Number of planes : 40
        K-grid : 300 x 300
                K-box factor : 0.7
                K-shift : (0.0 ,0.0)
        Number of batches : 60
        Temperature : 10 K
        Fermi level : 11.47650 eV
        Energy cut : 0.0 eV
        Output file : run2_ES.dat
```

After the calculation is done, the output file is automatically saved to the *run2* folder. We must plot the energy slice by means of *BP-energy_plot.py* component, whose configuration file (*energy_plot.yaml*) looks like:

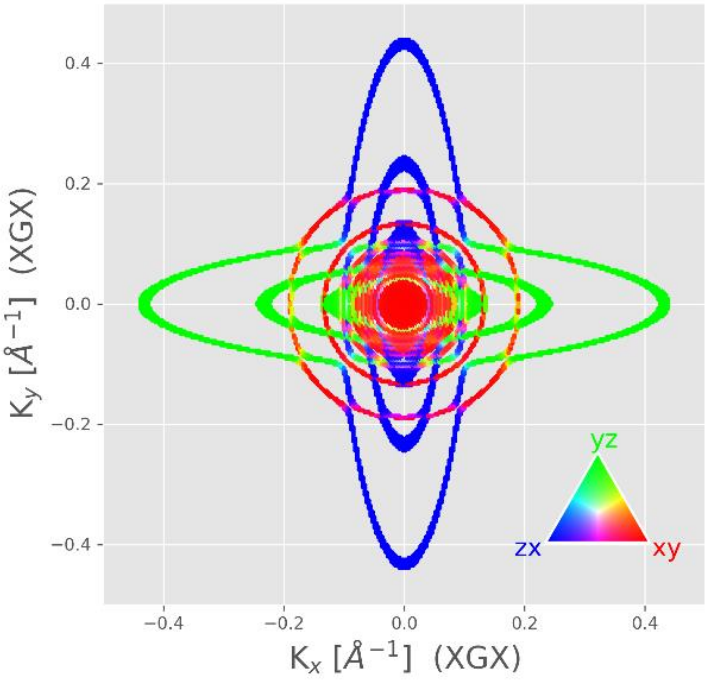
```
---
ENERGY_PLOTTER :
        identifier : "sto_01"
        input_file : "default"
        orbital_char : yes
        color_seq : red,lime,blue
        energy_window : 0.003

        PLOT_ADJUST :
                plotstyle : 'ggplot'
                point_size : 5
                xy_limits : [-0.47, 0.47, -0.47, 0.47]
                fig_size : [6,6]
                axis_adjust : [0.2, 0.15, 0.9, 0.9] #[left, bottom, right, top]
                title : ""
                title_size : 12
        LABELS :
                xlabel : 'K$_x$  ($\AA^{-1}$)'
                xfontsize : 18
                ylabel : 'K$_y$  ($\AA^{-1}$)'
                yfontsize : 18
                ticksize : 12
        COLOR_TRIANGLE:
                proportion : '35%'
                location : 4
                padding : 0.5
                fontsize : 16
        SAVING :
                save_plot : yes
                format : ".png"
                dpi : 300
...
```

So that, we generate the interactive plot by typing:

$ python BP-energy_plot.py -id run2

This is how the output plot looks like:



Now we will compute another energy slice, this time at 50 meV below the Fermi level. This could be done by typing:

$ python BP-energy_slices.py -id run2 -ec -0.05 -nk 300 -ba 60 -bf 0.7

Here, we can see that just the energy cut (*ec*) has been modified. In consequence, the details for this calculation will be now:

```
DETAILS :
        Identifier : run2
        Surface : STO(100)
        Number of planes : 40
        K-grid : 300 x 300
                K-box factor : 0.7
                K-shift : (0.0 ,0.0)
        Number of batches : 60
        Temperature : 10 K
        Fermi level : 11.47650 eV
        Energy cut : -0.05 eV
        Output file : run2_ES.dat
```

Note that, because we did not change the name of the output file, it will be overwritten. We will get the interactive plot by using the *BP-energy_plot.py* component just as before:

$ python BP-energy_plot.py -id run2

This is how the output plot looks like: