Postprocessing of the quasiparticle energies to obtain the GW band structure

We are going to use the YAMBO post-processor *ypp* to process the band structure. Apply the following set of commands to extract the required data to plot the GW band structure. For more information about *ypp* run levels please go to: http://www.yambo-code.org/input_file/ypp/ypp_bands.php

The major requirement of the *ypp* post-processor is that we need to remove the time-reversal (TR) symmetry in the k-points generated in the ground state calculation, before we plot the GW band-structure interpolation. We are going to remove the TR symmetry and then save the data in a newly created directory named "FixSymm."

First apply the following command to generate the band interpolation input file. This command will generate the input file (ypp.in) and immediately open it in vi editor. You do not need to change anything just save the vi file and exit.

```
$ ./ypp -s b
```

Now apply the following command to remove the TR symmetry.

```
$ ./ypp -y
```

This command will generate the following file and immediately open it in vi editor. Untag the TR symmetry (remove the # sign in the line marked in red from the file below), then save and exit the vi file.

```
#   __   __     _       __  __          ____       U  ___  u
#   \ \ / / U  /"\  U u |" \/ "| u U | __") u     \/"_ \/
#    \ V /    \/ _ \/   \| |\/| |/    \|  _ \/      | | | |
#   U_|"|_u  / ___ \     | |  | |      | |_) | .-,_| |_| |
#     |_|   /_/   \_\    |_|  |_|      |____/   \_)-\___/
# .-,//|(_    \\    >>  <<,-,,-.     _|| \\_        \\
#  \_) (__) (__)  (__)  (./  \.)    (__) (__)      (__)
#
# GPL Version 4.2.1 Revision 110. (Based on r.14778 h.7b4dc3
#                         MPI Build
#                http://www.yambo-code.org
#
fixsyms                          # [R] Reduce Symmetries
#RmAllSymm                       # Remove all symmetries
RmTimeRev                       # Remove Time Reversal
```

Now apply the following command to generate the new directory named “FixSymm.”

```
$ ./ypp
```

At the end of this command it will generate the “FixSymm” directory with new data inside the “MoS2.save” directory. Now we can perform the band interpolation inside the FixSymm directory,

Apply the following command inside the FixSymm directory to initialize YAMBO and to test everything runs well,

```
$ ../yambo
```

You should have the following run-time output at the end,

```
.
.
.
.
.
E) 02s(X)
 <11s> SE indexes |########################################| [100%] 02s
(E) 02s(X)
 <16s> [04] Timing Overview
 <16s> [05] Game Over & Game summary
```

Now apply the following command to generate the band interpolation input file. This command will generate the following input file (ypp.in) and immediately open it in vi editor.

```
$ ../ypp -s b -V qp
```

Update the section marked in red accordingly, save the vi file and exit.

```
#
# _____ _____    _     _____   _____ _______     ___
# |_  _||_  _|  / \    |_  \ / _||_  _ \  ."  `.
#  \ \ / /   / _ \     |  \/  |   | |_) | /  .-.  \
#   \ \/ /   / ___ \    | |\  /| |   |  __". | |  | |
#   _| |_  _/ /  \ \_  _| |_\/_| |_ _| |__) |\  `-" /
#  |_____||____| |____||_____||_____||_____/  `.___."
#
```

```
#
#
# GPL Version 4.2.1 Revision 110. (Based on r.14778 h.7b4dc3
#                     MPI Build
#             http://www.yambo-code.org
#
electrons                # [R] Electrons (and holes)
bnds                     # [R] Bands
cooIn= "rlu"             # Points coordinates (in) cc/rlu/iku/alat
BANDS_steps=20           # Number of divisions
% INTERPGrid
-1 |-1 |-1 |                     # Interpolation BZ Grid
%
ShellFac= 20.00000       # The bigger it is a higher number of shells is used
GfnQPdb= "E < SAVE/ndb.QP"           # [EXTQP G] Database
GfnQP_N= 1               # [EXTQP G] Interpolation neighbours
% GfnQP_E
 0.000000 | 1.000000 | 1.000000 |       # [EXTQP G] E parameters  (c/v) eV|adim|adim
%
GfnQP_Z= ( 1.000000 , 0.000000 )       # [EXTQP G] Z factor  (c/v)
GfnQP_Wv_E= 0.000000   eV      # [EXTQP G] W Energy reference  (valence)
% GfnQP_Wv
 0.00     | 0.00     | 0.00     |       # [EXTQP G] W parameters  (valence) eV| 1|eV^-1
%
GfnQP_Wv_dos= 0.000000 eV      # [EXTQP G] W dos pre-factor  (valence)
dos                      # [R] DOS
GfnQP_Wc_E= 0.000000   eV      # [EXTQP G] W Energy reference  (conduction)
% GfnQP_Wc
 0.00     | 0.00     | 0.00     |       # [EXTQP G] W parameters  (conduction) eV| 1 |eV^-1
%
GfnQP_Wc_dos= 0.000000 eV      # [EXTQP G] W dos pre-factor  (conduction)
%QPkrange                #  generalized Kpoint/Band indices
  1| 78|  10| 16|
%
%BKpts                   # K points of the bands circuit
0.00000 |0.00000 |0.00000 |
0.66666 |-0.33333 |0.00000 |
0.50000 |0.00000 |0.00000 |
0.00000 |0.00000 |0.00000 |
%
```

The option “-V qp” is to include the Quasi-particle correction.

Modify the line GfnQPdb= "none" to GfnQPdb= "E < SAVE/ndb.QP" where ndb.QP is the database containing the quasi-particle corrections.

In the section “QPkrange”, we ask the program to plot GW band numbers 10 to 16 (b10 - b11).

“BANDS_steps=20” defines the number of divisions between k-points. We define the preferred k-point path under the section “BKpts”.

Now run the *ypp* code to interpolate the GW band structure.

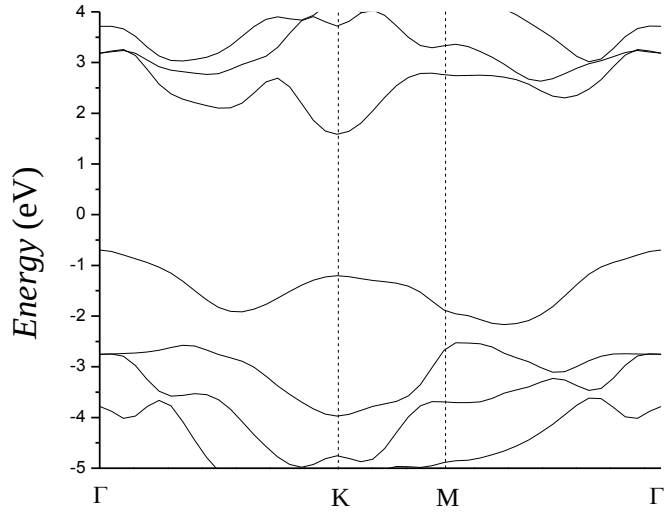
```
$ ../ypp
```

You will find the necessary output data points to plot the band structure in the file named “o.bands_interpolated” as shown below,

```
#  __  __   ________   ___ __ __    _______   ______
# /_/\/_/\ /_______/\ /__//_//_/\ /_______/\ /_____/\
# \ \ \ \ \\::: _  \ \\::\| \| \ \\::: _  \ \\::_ \ \
#  \:\_\ \ \\::(_)  \ \\:.      \ \\::(_)  \/_\:\ \ \ \
#   \::::_\/ \:: __  \ \\:.\-/\  \ \\::  _  \ \\:\ \ \ \
#     \::\ \  \:.\ \  \ \\. \  \  \ \\::(_)  \ \\:\_\ \ \
#      \__\/   \__\/\__\/ \__\/ \__\/ \_______\/ \_____\/
#
#
# GPL Version 4.2.1 Revision 110. (Based on r.14778 h.7b4dc3
#                        MPI Build
#                http://www.yambo-code.org
#
#     |k|          b10          b11          b12          b13
b14          b15          b16          kx           ky           kz
#
      0.000000    -3.782562    -2.753197    -2.753098    -0.696117
3.182787     3.183222     3.716361     0.000000     0.000000     0.0000
00
       0.03490     -3.88096     -2.74582     -2.74852     -0.72613
3.22668      3.21217      3.71416      0.03333     -0.01667      0.0000
0
       0.06981     -4.01496     -2.79262     -2.73862     -0.79446
3.25758      3.23814      3.66399      0.06667     -0.03333      0.0000
0
       0.10471     -3.97347     -2.97577     -2.72625     -0.86923
3.13814      3.18156      3.51600      0.10000     -0.05000      0.0000
0
       0.13962     -3.78489     -3.24896     -2.70882     -0.94485
2.86789      3.04885      3.30870      0.13333     -0.06667      0.0000
0
       0.17452     -3.66134     -3.47975     -2.67282     -1.03529
2.57580      2.91722      3.13149      0.16667     -0.08333      0.0000
0
```

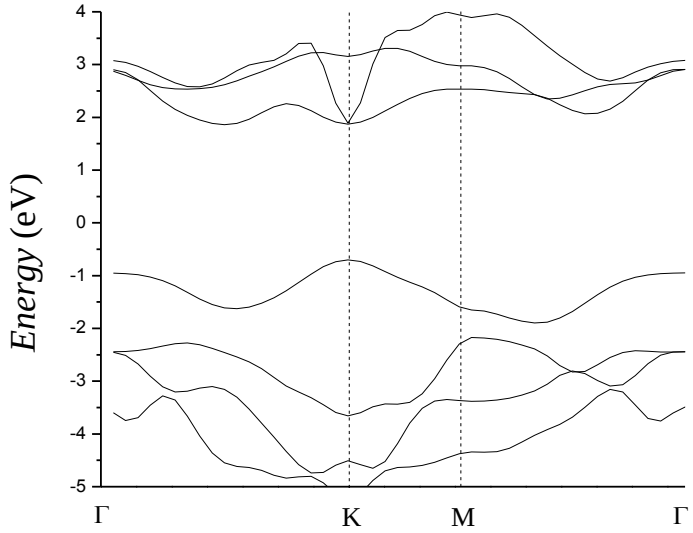
```
--More--(53%)
```

Finally, you can plot "|k| vs "b10 through b16" columns using a preferred graphing tool such as excel, origin, etc…, to plot the GW band structure as follows,



GW Band structure of MoS_2 single layer with *ecutwfc*=10Ry and k-points mesh 12×12×1.

From the following GW band structure plot you can see how the calculation is sensitive to the *ecutwfc* parameter, when it is changed from 10Ry to 30Ry.



GW Band structure of MoS_2 single layer with *ecutwfc*=30Ry and k-points mesh 12×12×1.

If you are interested in further details of convergence of parameters for publication quality for MoS_2, please refer to: Molina-Sánchez *et* al, Phys. Rev. **B 88**, 045412 (2013), **https://arxiv.org/abs/1306.4257**.