VASP2KP

TopMat: www.topmat.org

VASP2KP home page: http://www.vasp2kp.com

Theoretical basis of VASP2KP: http://www.vasp2kp.com/theory/

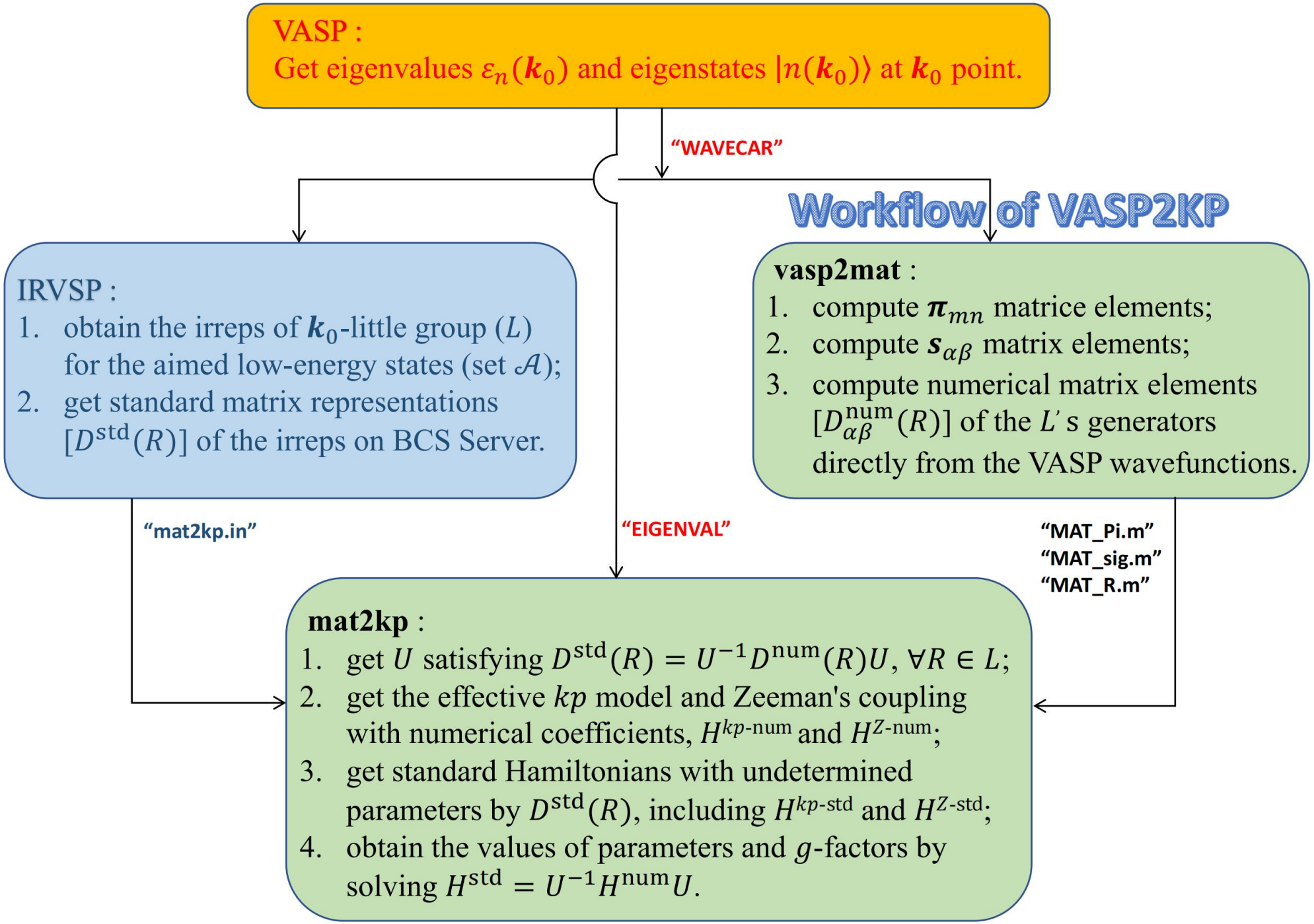
VASP2KP source code: https://github.com/zjwang11/VASP2KP

VASP2KP python package: https://pypi.org/project/VASP2KP

VASP2KP

vasp2mat : compute matrix representations of the generalized momentum operator $\hat{\pi}$, spin operator $\hat{s}$, time reversal operator $\hat{T}$ and crytalline symmetry operators $\hat{R}$ on the DFT wavefunctions. The matrix elements of the operators are derived comprehensively and computed correctly within the Projector Augmented Wave method.

mat2kp : obtain the unitary transformation U that rotates the DFT basis towards the standard basis, and then automatically compute the kp parameters and g-factors. The obtained effective masses and g-factors are important and comparable with experimental observations.


VASP :
Get eigenvalues $\varepsilon_n(\boldsymbol{k}_0)$ and eigenstates $|n(\boldsymbol{k}_0)\rangle$ at $\boldsymbol{k}_0$ point.
"WAVECAR"
Workflow of VASP2KP
IRVSP :
1. obtain the irreps of $\boldsymbol{k}_0$-little group (L) for the aimed low-energy states (set $\mathcal{A}$);
2. get standard matrix representations $[D^{\mathrm{std}}(R)]$ of the irreps on BCS Server.
vasp2mat :
1. compute $\boldsymbol{\pi}_{mn}$ matrice elements;
2. compute $\boldsymbol{s}_{\alpha\beta}$ matrix elements;
3. compute numerical matrix elements $[D^{\mathrm{num}}_{\alpha\beta}(R)]$ of the L's generators directly from the VASP wavefunctions.
"mat2kp.in"
"EIGENVAL"
"MAT_Pi.m"
"MAT_sig.m"
"MAT_R.m"
mat2kp :
1. get U satisfying $D^{\mathrm{std}}(R) = U^{-1}D^{\mathrm{num}}(R)U$, $\forall R \in L$;
2. get the effective kp model and Zeeman's coupling with numerical coefficients, $H^{kp\text{-num}}$ and $H^{Z\text{-num}}$;
3. get standard Hamiltonians with undetermined parameters by $D^{\mathrm{std}}(R)$, including $H^{kp\text{-std}}$ and $H^{Z\text{-std}}$;
4. obtain the values of parameters and g-factors by solving $H^{\mathrm{std}} = U^{-1}H^{\mathrm{num}}U$.

Installation of vasp2mat (based on vasp.6.4.X)

1. Prepare the source file folder of vasp.6.4.X and renamed it as ‘vasp.6.4’.

2. Download the patch ‘vasp2mat.6.4-patch-1.0.1.sh’ from https://github.com/zjwang11/VASP2KP

3. Put the patch and ‘vasp.6.4’ in the same folder.

```
(base) [szhang@alpha100 try]$ ls
vasp2mat.6.4-patch-1.0.1.sh  vasp.6.4
```

4. Run the patch ‘vasp2mat.6.4-patch-1.0.1.sh’.

$ bash vasp2mat.6.4-patch-1.0.1.sh

If vasp2mat is compiled successfully, the information below will be output

```
Finishing installing vasp2mat!
The path of vasp2mat:
```

The path of vasp2mat will be shown above.

Installation of vasp2mat (based on vasp.5.3.X)

1. Prepare the source file folder of vasp.5.3.X and renamed it as ‘vasp.5.3’, and prepare the library file ‘vasp.5.lib’.

2. Download the patch ‘vasp2mat.5.3-patch-1.0.1.sh’ from https://github.com/zjwang11/VASP2KP

3. Put ‘vasp.5.3’, ‘vasp.5.lib’ and ‘vasp2mat.5.3-patch-1.0.1.sh’ in the same folder.

4. Compile ‘vasp.5.lib’.

5. Run the patch ‘vasp2mat.5.3-patch-1.0.1.sh’.

$ bash vasp2mat.5.3-patch-1.0.1.sh

If vasp2mat is compiled successfully, the information below will be output

The path of vasp2mat will be shown above.

Installation of mat2kp

1. Install python package ‘kdotp-generator’.

$ pip install kdotp-generator

2. Install python package ‘VASP2KP’.

$ pip install VASP2KP

3. Download the patch ‘mat2kp’ from https://github.com/zjwang11/VASP2KP

4. (optional) Revise the first line in ‘mat2kp’ to the path of the python Interpreter.

1. Transform POSCAR to the standard form (for irvsp).

```
$ phonopy --symmetry --tolerance 0.01 -c POSCAR
$ POS2ABR > ABR.out
```

then POSCAR_std is the standard POSCAR.

2. Run VASP to do self-consistence field calculation.

3. Run VASP to do band calculation to obtain WAVECAR and EIGENVAL at the specific K-point.

Notice:
INCAR: to construct the kp model, the parameter NBANDS in the band calculation input file INCAR should be set to a large number, and 'LWAVE=.TRUE.' should be set to obtain WAVECAR!

KPOINTS: It is better to set only 1 kpoint to construct the kp model.

3. Run VASP to do band calculation to obtain WAVECAR and EIGENVAL at the specific K-point.

Example: Bi2Se3-Γ

KPOINTS

```
Bi2Se3
1
Reciprocal lattice
0   0   0   1.0
```

INCAR

```
ISTART = 1
ICHARG = 11
LWAVE = .TRUE.
LSORBIT = .TRUE.
MAGMOM = 15*0.0
NBANDS = 600
.....(other parameters)
```

4. Run IRVSP to obtain the standard irreps.

Input files: WAVECAR, OUTCAR

$ irvsp -sg <space group number> -nb <start band number> <end band number>

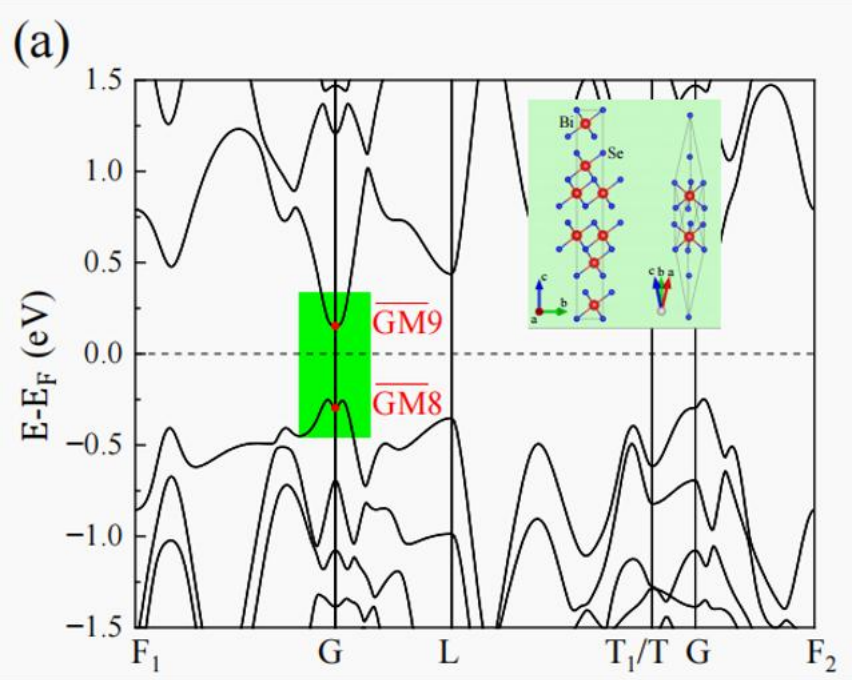
Example: Bi2Se3-Γ, bands: 47-50

$ irvsp -sg 166 -nb 47 50

Output: tqc.txt

Computed bands: 47 - 50
GM: GM8 (2); GM9 (2); [4]

The irreps of the bands are GM8 and GM9.



5. Run vasp2mat to obtain the generalized momentum matrix.

Input files: POSCAR, POTCAR, INCAR, KPOINTS, WAVECAR, INCAR.mat

$ vasp2mat > vasp2mat.out 2> vasp2mat.err

INCAR.mat

```
&vmat_para
    vmat = 11
    vmat_name = 'Pi'
    vmat_k = <k-point number>
    bstart=<start band number>,  bend=<end band number>
/
```

Notice: When using vasp2mat to calculate matrices, MPI parallel computing is not allowed!

5. Run vasp2mat to obtain the generalized momentum matrix.

Notice:
POSCAR, POTCAR, KPOINTS file should be the same as the step 4.
INCAR file should be the same as the step 4, and revise ‘LWAVE = .TRUE.’ to ‘LWAVE = .FALSE.’, ‘ICHARG = 11’ to ‘ICHARG = 1’.
To construct kp model, the energy band range of generalized momentum matrix should be large.
For instance, if we want to calculate the kp model of the bands 47-50, we can set ‘bstart=1, bend=600’. (‘bend’ should not be larger than NBANDS in INCAR)

Output: MAT_Pi.m

A matlab source file, which contains the generalized momentum matricies along x, y and z directions. This source file can be directly imported into matlab.

generalized momentum matrix $\hat{\boldsymbol{\pi}} = \hat{\boldsymbol{p}} + \frac{1}{2mc^2}(\hat{\boldsymbol{s}} \times \nabla V(\boldsymbol{r}))$

5. Run vasp2mat to obtain the generalized momentum matrix.

Example: Bi2Se3-Γ

KPOINTS

```
Bi2Se3
1
Reciprocal lattice
0   0   0   1.0
```

INCAR

```
ISTART = 1
ICHARG = 1
LWAVE = .FALSE.
LSORBIT = .TRUE.
MAGMOM = 15*0.0
NBANDS = 600
......(other parameters)
```

INCAR.mat

```
&vmat_para
    vmat = 11
    vmat_name = 'Pi'
    vmat_k = 1
    bstart=<start band num>,  bend=<end band num>
/
```

5. Run vasp2mat to obtain generalized momentum matrices.

If SOC is not considered in your system, you should computed momentum matrix instead of generalized momentum matrices.

INCAR.mat
for momentum matrix

```
&vmat_para
    vmat = 7
    vmat_name = 'Pi'
    vmat_k = <k-point number>
    bstart=<start band num>,  bend=<end band num>
/
```

$$\hat{\boldsymbol{\pi}} = \hat{\boldsymbol{p}} + \frac{1}{2mc^2}(\hat{\boldsymbol{s}} \times \nabla V(\boldsymbol{r}))$$

6. Run vasp2mat to obtain spin matrices.

Input files: POSCAR, POTCAR, INCAR, KPOINTS, WAVECAR, INCAR.mat

$ vasp2mat > vasp2mat.out 2> vasp2mat.err

INCAR.mat

```
&vmat_para
    vmat = 10
    vmat_name = 'sig'
    vmat_k = <k-point number>
    bstart=<start band num>,  bend=<end band num>
/
```

Output: MAT_sig.m

A matlab source file, which contains the spin matricies along x, y and z directions. This source file can be directly imported into matlab.

When ‘LNONCOLLINEAR=.False.’, spin matrices can not be computed.

7. Run vasp2mat to obtain the representation matrices of generators of the little group at the specific-K point.

Input files: POSCAR, POTCAR, INCAR, KPOINTS, WAVECAR, INCAR.mat

$ vasp2mat > vasp2mat.out 2> vasp2mat.err

INCAR.mat

```
&vmat_para
 ! operator-----------------------------------
 vmat = 12
 vmat_name = '<operator's name>'
 vmat_k = <k-point number>
 bstart=<start band num>,  bend=<end band num>
 ! rotation-----------------------------------
 rot_n(:) = <The vector of the rotational axis of the symmetry operator.>
 rot_alpha = <The rotation angle in degrees.>
 rot_det = 1/-1                    ! The determinant of the O(3) matrix of the symmetry operator.
 rot_tau(:) = <The space translation part of the symmetry operator.>
 rot_spin2pi = .false.
 time_rev = .true./.false.         ! Whether the operator has a time-reversal component
/
```

7. Run vasp2mat to obtain the representation matrices of generators of the little group at the specific-K point.

Example: Bi2Se3-Γ

INCAR.mat

```
&vmat_para
 ! operator-----------------------------------
 vmat = 12
 vmat_name = 'C3z'                    !/   'C2x'   /   'P'   /   'T'
 vmat_k = 1
 bstart=47, bend=50
 ! rotation-----------------------------------
 rot_n(:) = 0 0 1                   !/   1 0 0   /   0 0 1   /   0 0 1
 rot_alpha = 120                     !/   180   /   0   /   0
 rot_det = 1                        !/   1   /   -1   /   1
 rot_tau(:) = 0 0 0
 rot_spin2pi = .false.
 time_rev = .false.                    !/   .false.   /   .false.   /   .true.
/
```

8. Put all matrix files obtained in steps 5-7 and EIGENVAL obtained in step 3 in a file folder.

Directory Structure
----<file folder name>
| | EIGENVAL
| | MAT_Pi.m
| | MAT_sig.m
| | MAT_***.m

Notice:
1. EIGENVAL is obtained in step 3 but not in steps 5-7. When computing the matrices, vasp2mat will also generate EIGENVAL, but it is empty so that it can not be the input file of mat2kp!
2. To run mat2kp, vmat_name of generalized momentum matrices (or momentum matrices) should be set to ‘Pi’, and that of spin matrices should be set to ‘sig’. vmat_name of representation matrices can be set arbitrarily.

8. Put all matrix files obtained in steps 5-7 and EIGENVAL obtained in step 3 in a file folder.

Example: Bi2Se3-Γ

Directory Structure

```
Bi2Se3
 |  mat2kp.in
 |
 └──GMmat
 |    |  EIGENVAL
 |    |  MAT_Pi.m
 |    |  MAT_sig.m
 |    |  MAT_C3z.m
 |    |  MAT_C2x.m
 |    |  MAT_P.m
 |    |  MAT_T.m
```

9. Create the input file ‘mat2kp.in’ for mat2kp.

Necessary parameter : Symmetry, python dictionary.

a part of mat2kp.in

```
Symmetry = {
‘<operator1’s name>’ : {
'rotation_matrix': <O(3) matrix of operator1>,
‘repr_matrix’: <representation matrix of operator1>,
‘repr_has_cc’: True/False   #Whether the operator has a time-reversal component
},
‘<operator2’s name>’ : {
'rotation_matrix': <O(3) matrix of operator2>,
‘repr_matrix’: <representation matrix of operator2>,
‘repr_has_cc’: True/False   #Whether operator2 has a time-reversal component
},
...
‘<operatorN’s name>’ : {
'rotation_matrix': <O(3) matrix of operatorN>,
‘repr_matrix’: <representation matrix of operatorN>,
‘repr_has_cc’: True/False   #Whether operatorN has a time-reversal component
}
}
```

9. Create the input file ‘mat2kp.in’ for mat2kp and write the symmetry information.

Necessary parameter : Symmetry, python dictionary.

Notice:

1. the matrix should be input as:

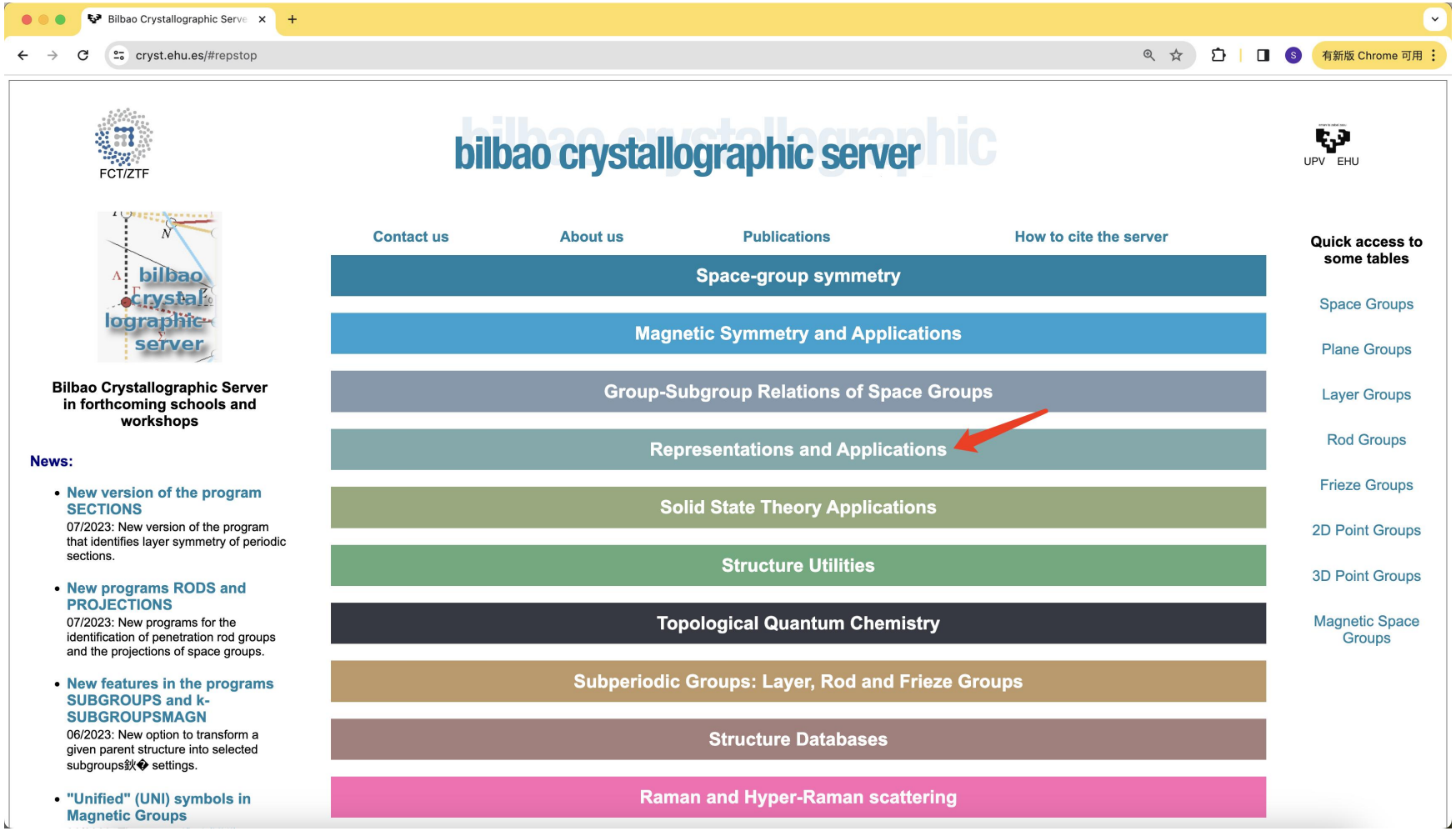
```
Matrix([
[0,-Rational(1,2)-I*sqrt(3)/2,0,0],
[Rational(1,2)-I*sqrt(3)/2,0,0,0],
[0,0,0,-Rational(1,2)-I*sqrt(3)/2],
[0,0,Rational(1,2)-I*sqrt(3)/2,0]
])
```

The fraction a/b should be written as Rational(a,b), imaginary unit should be written as I.

2. Math functions sin, cos, sqrt and tan can be directly used.
3. If the representation is composed by many irreps, matrices are better to be written as $D^1(R)\bigoplus D^2(R)\bigoplus \ldots \bigoplus D^N(R)$, where irreps are arranged in order of their corresponding energies from low to high($E^1 < E^2 < \ldots < E^N$).
4. operators’ name should be the same in vasp2mat!

10. Copy standard representation from BCS Server.

BCS Server: https://www.cryst.ehu.es/#repstop



10. Copy standard representation from BCS Server.

Example: Bi2Se3-166-Γ

Computed bands: 47 - 50
GM: GM8 (2); GM9 (2); [4]

https://www.cryst.ehu.es/cgi-bin/cryst/programs/corepresentations_out.pl?super=166.98&vecfinal=GM

	$\overline{\text{GM8}}$	$\overline{\text{GM9}}$
$\{C_{3z}\|0,0,0\}$	$\begin{pmatrix} e^{-\frac{\pi i}{3}} & 0 \\ 0 & e^{\frac{\pi i}{3}} \end{pmatrix}$	$\begin{pmatrix} e^{-\frac{\pi i}{3}} & 0 \\ 0 & e^{\frac{\pi i}{3}} \end{pmatrix}$
$\{C_{2x}\|0,0,0\}$	$\begin{pmatrix} 0 & e^{-\frac{2\pi i}{3}} \\ e^{-\frac{\pi i}{3}} & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & e^{-\frac{2\pi i}{3}} \\ e^{-\frac{\pi i}{3}} & 0 \end{pmatrix}$
$\{P\|0,0,0\}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$
$\{T\|0,0,0\}$	$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\mathcal{K}$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\mathcal{K}$

10. Copy standard representation from BCS Server.

Example: Bi2Se3-166-Γ

	$\overline{\text{GM8}}$	$\overline{\text{GM9}}$
$\{C_{3z}\|0,0,0\}$	$\begin{pmatrix} e^{-\frac{\pi i}{3}} & 0 \\ 0 & e^{\frac{\pi i}{3}} \end{pmatrix}$	$\begin{pmatrix} e^{-\frac{\pi i}{3}} & 0 \\ 0 & e^{\frac{\pi i}{3}} \end{pmatrix}$
$\{C_{2x}\|0,0,0\}$	$\begin{pmatrix} 0 & e^{-\frac{2\pi i}{3}} \\ e^{-\frac{\pi i}{3}} & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & e^{-\frac{2\pi i}{3}} \\ e^{-\frac{\pi i}{3}} & 0 \end{pmatrix}$
$\{P\|0,0,0\}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$
$\{T\|0,0,0\}$	$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\mathcal{K}$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\mathcal{K}$

a part of mat2kp.in

```
Symmetry = {
'C3z' : {'rotation_matrix':Matrix([[-Rational(1,2), -sqrt(3)/2,0],[sqrt(3)/2, -Rational(1,2), 0],[0, 0, 1]]),
'repr_matrix':Matrix([[Rational(1,2)-I*sqrt(3)/2,0,0,0],[0,Rational(1,2)+I*sqrt(3)/2,0,0],
[0,0,Rational(1,2)-I*sqrt(3)/2,0],[0,0,0,Rational(1,2)+I*sqrt(3)/2]]),
'repr_has_cc': False},
'C2x' : {'rotation_matrix': Matrix([[1, 0, 0],[0, -1, 0],[0, 0, -1]]),
'repr_matrix':Matrix([[0,-Rational(1,2)-I*sqrt(3)/2,0,0],[Rational(1,2)-I*sqrt(3)/2,0,0,0],
[0,0,0,-Rational(1,2)-I*sqrt(3)/2],[0,0,Rational(1,2)-I*sqrt(3)/2,0]]),
'repr_has_cc': False},
'P' : {'rotation_matrix': Matrix([[-1,0,0],[0, -1, 0],[0, 0, -1]]),
'repr_matrix':Matrix([[1,0,0,0],[0,1,0,0],[0,0,-1,0],[0,0,0,-1]]),
'repr_has_cc': False},
'T' : {'rotation_matrix': eye(3),# Identity Matrix
'repr_matrix': Matrix([[0,1,0,0],[-1,0,0,0],[0,0,0,-1],[0,0,1,0]]),
'repr_has_cc': True}
}
```

11. Set the optional parameters in mat2kp.in.

a part of mat2kp.in

```
# optional parameters
vaspMAT = '<the path of the file folder created in step 8, containing matrices and EIGENVAL>'
order = <Order of the kp model : 2 (default) or 3>
kpmodel = <Whether to compute Hkp: 0 or 1 (default)>
gfactor = <Whether to compute HZ: 0 or 1 (default)>
print_flag = <Where to output results: 1 (screen) or 2 (files, default)>
log = <Whether to output log files: 0 or 1 (default)>
repr_split = <If you use BCS Server's irreps (False) or not (True). Default: False.>
```

Notice:

1. Optional parameter 'vaspMAT' is recommended to set.
The default value is 'mat'.
2. Optional parameters 'print_flag', 'repr_split' and 'log' are not recommened to change.
3. Optional parameters 'order' should not be higher than 3.

9. Create the input file ‘mat2kp.in’ for mat2kp and write the symmetry information.
10. Copy standard representation from BCS Server.
11. Set the optional parameters in mat2kp.in.

Example: Bi2Se3-166-Γ, ‘mat2kp.in’ file

```
Symmetry = {
'C3z' : {'rotation_matrix':Matrix([[-Rational(1,2), -sqrt(3)/2,0],[sqrt(3)/2, -Rational(1,2), 0],[0, 0, 1]]),
'repr_matrix':Matrix([[Rational(1,2)-I*sqrt(3)/2,0,0,0],[0,Rational(1,2)+I*sqrt(3)/2,0,0],
[0,0,Rational(1,2)-I*sqrt(3)/2,0],[0,0,0,Rational(1,2)+I*sqrt(3)/2]]),'repr_has_cc': False},
'C2x' : {'rotation_matrix': Matrix([[1, 0, 0],[0, -1, 0],[0, 0, -1]]),
'repr_matrix':Matrix([[0,-Rational(1,2)-I*sqrt(3)/2,0,0],[Rational(1,2)-I*sqrt(3)/2,0,0,0],
[0,0,0,-Rational(1,2)-I*sqrt(3)/2],[0,0,Rational(1,2)-I*sqrt(3)/2,0]]),'repr_has_cc': False},
'P' : {'rotation_matrix': Matrix([[-1,0,0],[0, -1, 0],[0, 0, -1]]),
'repr_matrix':Matrix([[1,0,0,0],[0,1,0,0],[0,0,-1,0],[0,0,0,-1]]),'repr_has_cc': False},
'T' : {'rotation_matrix': eye(3),'repr_matrix': Matrix([[0,1,0,0],[-1,0,0,0],[0,0,0,-1],[0,0,1,0]]),'repr_has_cc': True}
}
vaspMAT = '../Bi2Se3/GMmat' # the path: to read eigenvalues, Pi, s, and R matrices in this folder.
order = 2 # Order of the kp model : 2 (default) or 3.
print_flag = 2 # Where to output results: 1 (screen) or 2 (files, default).
kpmodel = 1 # Whether to compute Hkp: 0 or 1 (default).
gfactor = 1 # Whether to compute HZ: 0 or 1 (default).
log = 1 # Whether to output log files: 0 or 1 (default).
repr_split = False # If you use BCS Server’s irreps (False) or not (True).
```

12. Run mat2kp.

$ mat2kp > mat2kp.out 2> mat2kp.err

mat2kp.out

```
Low-energy bands: [...]
Energies of these bands: [...]
Number of low-energy bands: ...
Number of IrReps: ...
IrReps' dimensions: [...]
Loading D^std from "mat2kp.in" successfully!
Traces of the matrix representations (D^num) obtained by vasp2mat are given below:
IrRep  Dim      ...
     1     ...
    ...
K is the complex conjugation for anti-unitary operations.
Loading D^num from "MAT_*.m" successfully!
Find the unitary transformation U successfully (U^-1 D^num U= D^std) !
Finish downfolding processes!
Finish constructing kp invariant Hamiltonian of order 0!
...
Finish constructing kp invariant Hamiltonian in "kp-parameters.out"!
Finish constructing Zeeman's coupling in "g-factors.out"!
Congratulations! The computation of VASP2KP has finished!!!
```

12. Run mat2kp.

Notice:
1. mat2kp should be run in the directory where the mat2kp.in file is located.
2. If you set ‘print_flag = 1’, all the output will be printed to the screen (mat2kp.out) but not *.out file.
3. If you use VASP2KP<=1.0.2, mat2kp.out will be empty.
4. Select as few generators as possible and try to align their rotation axes along the xyz axes as much as possible, otherwise it may require a long computation time!

12. Run mat2kp.

Example: Bi2Se3-166-Γ, ‘mat2kp.out’ file

mat2kp.out

```
Low-energy bands: [47 48 49 50]
Energies of these bands: [4.665438 4.665438 5.114245 5.114245]
Number of low-energy bands: 4
Number of IrReps: 2
IrReps' dimensions: [2, 2]
Loading D^std from "mat2kp.in" successfully!
Traces of the matrix representations (D^num) obtained by vasp2mat are given below:
IrRep  Dim         C3z          C2x           P           T
   1    2   1.00+0.00i   -0.00+0.00i    2.00-0.00i    0.00+0.00i K
   2    2   1.00+0.00i   -0.00+0.00i   -2.00-0.00i    0.00+0.00i K
K is the complex conjugation for anti-unitary operations.
Loading D^num from "MAT_*.m" successfully!
Find the unitary transformation U successfully (U^-1 D^num U= D^std) !
Finish downfolding processes!
Finish constructing kp invariant Hamiltonian of order 0!
Finish constructing kp invariant Hamiltonian of order 1!
Finish constructing kp invariant Hamiltonian of order 2!
Finish constructing kp invariant Hamiltonian in "kp-parameters.out"!
Finish constructing Zeeman's coupling in "g-factors.out"!
Congratulations! The computation of VASP2KP has finished!!!
```

13. Organize output files to obtain results.

Output files: kp-parameters.out, g-factors.out

The matrix of kp invariant Hamiltonian

kp-parameters.out

values of kp parameters

The matrix of Zeeman’s coupling

g-factors.out

values of g-factors

kp Hamiltonian
========== Result of kp Hamiltonian ==========
Matrix([[...]])
Parameters:
a1 = ...;
...
Error of the linear least square method: ...
Sum of absolute values of numerical zero elements: ...

Zeeman's coupling
========== Result of Zeeman's coupling ==========
mu_B/2*Matrix([[...]])
Parameters:
g1 = ...;
...
Error of the linear least square method: ...
Sum of absolute values of numerical zero elements: ...

13. Organize output files to obtain results.

Example: Bi2Se3-166-Γ

kp-parameters.out

```
kp Hamiltonian
========== Result of kp Hamiltonian ==========
Matrix([[a1 + a2 + c1*(kx**2 + ky**2) + c2*(kx**2 + ky**2) + c3*kz**2 + c4*kz**2, 0, -
I*b2*kz, -b1*(kx*(sqrt(3) + 3*I) + ky*(3 - sqrt(3)*I))/3], [0, a1 + a2 + c1*(kx**2 + ky**2) +
c2*(kx**2 + ky**2) + c3*kz**2 + c4*kz**2, b1*(kx*(sqrt(3) - 3*I) + ky*(3 + sqrt(3)*I))/3,
I*b2*kz], [I*b2*kz, b1*(kx*(sqrt(3) + 3*I) + ky*(3 - sqrt(3)*I))/3, a1 - a2 + c1*(kx**2 + ky**2) -
c2*(kx**2 + ky**2) + c3*kz**2 - c4*kz**2, 0], [-b1*(kx*(sqrt(3) - 3*I) + ky*(3 + sqrt(3)*I))/3, -
I*b2*kz, 0, a1 - a2 + c1*(kx**2 + ky**2) - c2*(kx**2 + ky**2) + c3*kz**2 - c4*kz**2]])
Parameters:
a1 = 4.8898 ;
a2 = -0.2244 ;
b1 = -3.238 ;
b2 = 2.5562 ;
c1 = 19.5842 ;
c2 = 44.4746 ;
c3 = 1.8117 ;
c4 = 9.5034 ;
Error of the linear least square method: 3.93e-06
Sum of absolute values of numerical zero elements: 6.47e-02
```

13. Organize output files to obtain results.

Example: Bi2Se3-166-Γ

g-factors.out

Zeeman's coupling
========== Result of Zeeman's coupling ==========
mu_B/2*Matrix([[Bz*g3 + Bz*g4, g1*(Bx*(1 - sqrt(3)*I/3) + By*(-sqrt(3)/3 - I)) + g2*(Bx*(1 - sqrt(3)*I/3) + By*(-sqrt(3)/3 - I)), 0, 0], [g1*(Bx*(1 + sqrt(3)*I/3) + By*(-sqrt(3)/3 + I)) + g2*(Bx*(1 + sqrt(3)*I/3) + By*(-sqrt(3)/3 + I)), -Bz*g3 - Bz*g4, 0, 0], [0, 0, Bz*g3 - Bz*g4, g1*(Bx*(1 - sqrt(3)*I/3) + By*(-sqrt(3)/3 - I)) + g2*(Bx*(-1 + sqrt(3)*I/3) + By*(sqrt(3)/3 + I))], [0, 0, g1*(Bx*(1 + sqrt(3)*I/3) + By*(-sqrt(3)/3 + I)) + g2*(Bx*(-1 - sqrt(3)*I/3) + By*(sqrt(3)/3 - I)), -Bz*g3 + Bz*g4]])
Parameters:
g1 = -0.3244 ;
g2 = 5.7610 ;
g3 = -7.8904 ;
g4 = -13.0138 ;
Error of the linear least square method: 6.11e-08
Sum of absolute values of numerical zero elements: 4.12e-03

13. Organize output files to obtain results.

$$H^{\text{eff}}(\boldsymbol{k}, \boldsymbol{B}) = H^{kp} + H^{Z},$$

$$H^{kp} = \begin{pmatrix} D_1 & 0 & -ib_2k_z & -\frac{3i+\sqrt{3}}{3}b_1k_- \\ & D_1 & \frac{\sqrt{3}-3i}{3}b_1k_+ & ib_2k_z \\ & & D_2 & 0 \\ \dagger & & & D_2 \end{pmatrix},$$

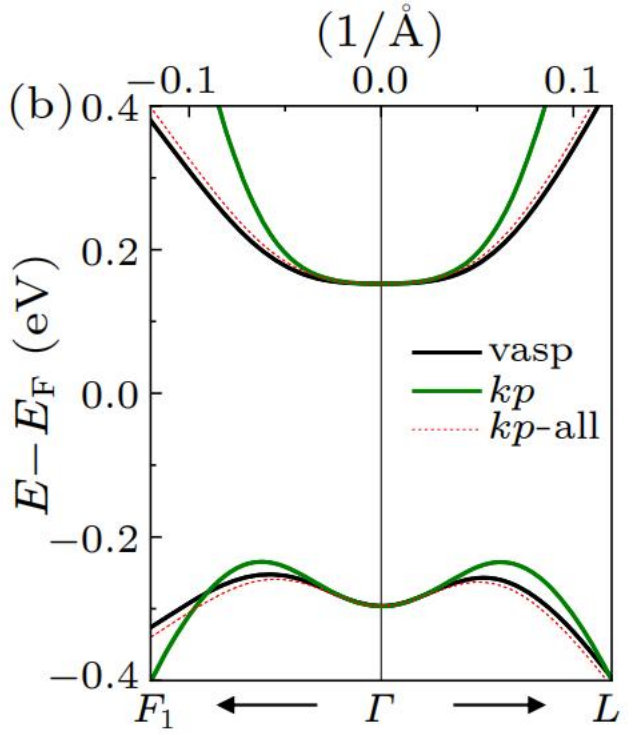
$$H^{Z} = \frac{\mu_{\text{B}}}{2}\begin{pmatrix} h_1^+B_z & h_2^+B_+ & 0 & 0 \\ & -h_1^+B_z & 0 & 0 \\ & & h_1^-B_z & h_2^-B_- \\ \dagger & & & -h_1^-B_z \end{pmatrix},$$

$$D_1 = a_1 + a_2 + (c_1 + c_2)k_+k_- + (c_3 + c_4)k_z^2,$$

$$D_2 = a_1 - a_2 + (c_1 - c_2)k_+k_- + (c_3 - c_4)k_z^2,$$

$$h_1^{\pm} = g_3 \pm g_4,\ h_2^{\pm} = \frac{3-\sqrt{3}i}{3}(g_1 \pm g_2),$$

$$k_{\pm} = k_x \pm ik_y,\ B_{\pm} = B_x \pm iB_y.$$



a_i (eV)	b_i (eV·Å)	c_i (eV·Å^2)	g_i
$a_1 = 4.89$	$b_1 = 3.24$	$c_1 = 19.58$	$g_1 = -0.32$
$a_2 = -0.22$	$b_2 = -2.56$	$c_2 = 44.47$	$g_2 = 5.76$
		$c_3 = 1.81$	$g_3 = -7.90$
		$c_4 = 9.50$	$g_4 = -13.01$

13. Organize output files to obtain results.

Notice:

1. kp parameters in kp-parameters.out are in units of $\text{Å}^n \bullet eV$, i.e., ai are in units of eV, bi in units of $\text{Å} \bullet eV$, ci in units of $\text{Å}^2 \bullet eV$,...
2. g-factors in g-factors.out are in units of 1.