User Manual

Version 2.0.0, Oct 28, 2024

Edit by

Zhaocheng Pan

Zhuoye Hu

Ruifeng Zhang

M I C I D

Materials Integrated Computation and Intelligent Design

Content

1 Introduction

VASP, an abbreviation of Vienna Ab-initio Simulation Package[1], stands as a renowned software suite crafted by the Hafner group at the University of Vienna. It is widely utilized for conducting first-principles calculations and executing ab initio molecular dynamics simulations. It is one of the most extensively utilized commercial software in materials computation and simulation, remains a preferred choice due to its high computational accuracy, efficiency, and flexibility for different systems and computational goals. Nevertheless, its lack of a straightforward graphical user interface poses a challenge for beginners.

In most cases, VASP users rely on third-party programs such as ASE, P4VASP, VASPView, VESTA, etc., for visualization, or use custom scripts and programs like VTST, VASPMO, qvasp, lev00, etc. Finding and successfully using the ideal processing script can still be difficult for new users. In recent years, the vaspkit[2] program has partially solved this problem to a great extent, as its powerful pre- and post-processing functions are very user-friendly and convenient for beginners. Nevertheless, vaspkit tends to focus more on simple and general user interaction controls, lacking support for the convenience of high-throughput computational design.

In this context, VASPMATE, a VASP companion program designed for simple and efficient high-throughput process design, was born. VASPMATE features a very user-friendly and powerful pre- and post-processing functionality specifically tailored for VASP software. It runs on Linux, Mac, or Windows systems, is written in C/C++, has a small size, strong portability, and is easy to operate. Additionally, VASPMATE can be easily integrated as a high-throughput flowchart into our integrated computational platform, SPaMD[3], thereby facilitating the design of graphical user interfaces for various high-throughput computational modules, such as ABAND and AEDOS, which are used for high-throughput first principles calculations of band structure and density of states.

2 Configuration and installation

Get the VASPMATE installation package

1. Download directly from the MICID website Link:

http://micid.top/software/vaspmate

2. Send registration form to zrfcms@buaa.edu.cn for the Baidu.com link.

3. Directly from github download, the link is as follows:

https://github.com/zrfcms/VASPMATE

Please send bugs and suggestions to zrfcms@buaa.edu.cn

>>>>>>>>>>>>>>>>>>> Directly run VASPMATE <<<<<<<<<<<<<<<<<<<

We have prepared the latest version of VASPMATE that can be used directly on Linux, MacOS, and Windows systems. If you do not intend to modify the existing code, we **strongly recommend** using it directly instead of recompiling. The specific usage are as follows:

Using on a Linux system

>>tar -xvf VASPMATE-x.x.x.tar

>>cd VASPMATE-x.x.x

>>cd bin

>> ./VASPMATE

At this time, if the terminal displays the following message, it indicates that VASPMATE can be used normally.

```
erye@erye-virtual-machine:/media/erye/erye1$ VASPMATE
***---------------------------------------VASPMATE Version 2.0.0 (2024.6.23)---------------------------------------***
***     An integrated user-interface program for high-throughput first principles computations through VASP code.    ***
***             Copyright[c] 2022-2024, Beihang University by Zhaocheng Pan, Zhuoye Hu and Ruifeng Zhang.             ***
***                             Please send bugs and suggestions to zrfcms@buaa.edu.cn                                ***
***-------------------------------------------------------------------------------------------------------------------***
```

For ease of use, it is recommended to copy VASPMATE to the default search path of the system, such as ~/bin, or add the absolute path of VASPMATE to the environment variables as follows:

>>pwd   #get the current directory path

>>echo ‘export PATH= the current directory path:$PATH’ >> ~/.bashrc

>>source ~/.bashrc

After completing the above configuration, the VASPMATE command is readily available in the terminal from any folder.

Using on a MacOS system

>>tar -xvf VASPMATE-x.x.x.tar

>>cd VASPMATE-x.x.x

>>cd bin

>> ./VASPMATE_mac

If the terminal displays the following message, it indicates that VASPMATE can be used normally.

```
erye@EryedeMac MacOS % ./VASPMATE
VASPMATE Version 2.0.0 (2024.2.19)
An integrated user-interface program for high-throughput first principles computations throu
gh VASP code.
Copyright[c] 2022-2024, Beihang University, by Zhaocheng Pan, Zhuoye Hu and Ruifeng Zhang
Please send bugs and suggestions to zrfcms@buaa.edu.cn
erye@EryedeMac MacOS %
```

Using on a Windows system

Next, we will demonstrate the usage in the Windows Shell terminal.

>>tar -xvf VASPMATE-x.x.x.tar

>>cd VASPMATE-x.x.x

>>cd bin

>> .\VASPMATE_win.exe

If the terminal displays the following message, it indicates that VASPMATE can be used normally.

```
PS E:\cygwin64\home\hzy\VASPMATE_daily_dilil\bin\Windows> .\VASPMATE.exe
VASPMATE Version 2.0.0 (2024.2.19)
An integrated user-interface program for high-throughput first principles computations through VASP code.
Copyright[c] 2022-2024, Beihang University, by Zhaocheng Pan, Zhuoye Hu and Ruifeng Zhang
Please send bugs and suggestions to zrfcms@buaa.edu.cn
PS E:\cygwin64\home\hzy\VASPMATE_daily_dilil\bin\Windows>
```

In particular, if you want to move VASPMATE to another folder, please make sure to copy the three additional *.dll* files to that folder, as shown below.

名称	修改日期	类型	大小
cyggcc_s-seh-1.dll	2023/6/5 23:21	应用程序扩展	74 KB
cygstdc++-6.dll	2023/6/5 23:21	应用程序扩展	1,904 KB
cygwin1.dll	2023/11/29 20:20	应用程序扩展	2,886 KB
VASPMATE.exe	2024/6/2 15:10	应用程序	4,515 KB

>>>>>>>>>>>>>>>>>>> Compiling VASPMATE <<<<<<<<<<<<<<<<<<<<

As an open-source software, VASPMATE allows users to modify the source code and recompile it for use. Below, we provide compilation methods for different systems.

Please note:

1. As a reminder, VASPMATE already provides a ready-to-use version. If you do not need to modify the source code, we do not recommend recompiling. Although we have compiled and adapted VASPMATE for various computers, we cannot guarantee that unexpected compilation issues will not occur.

2. VASPMATE compilation relies on the fftw library (which we have already included in the folder). If you encounter any errors related to fftw during the compilation process, please visit the fftw official website to find solutions.

https://www.fftw.org/

Compiling on a Linux system

It is assumed that gcc and g++ compilers are already installed by default on Linux systems:

```
>>tar -xvf VASPMATE-x.x.x.tar
>>cd VASPMATE-x.x.x.
>>make
```

After completing the above operations, if the following interface appears, it means the compilation is successful:

```
SPG_link.o  ./src/Evolutionary_src/Evolutionary_main.o  ./src/help_src/help.o -O
2 -std=c++0x -static -I./spglib  -I./ -I./libmsym -L./lib -I./fftw/include -L./f
ftw/lib -lfftw3 -lspg -lmsym -lstdc++
/usr/bin/ld: ./src/VASPMATE_src/potcar.o: in function `pot_merge(char const*, ch
ar*, std::vector<std::__cxx11::basic_string<char, std::char_traits<char>, std::a
llocator<char> >, std::allocator<std::__cxx11::basic_string<char, std::char_trai
ts<char>, std::allocator<char> > > >)':
potcar.cpp:(.text+0x419): warning: Using 'getpwuid' in statically linked applica
tions requires at runtime the shared libraries from the glibc version used for l
inking
Compiling VASPMATE
mv VASPMATE ./bin/
Compile successfully!
erye@erye-virtual-machine:~/VASPMATE_daily$
```

The installed VASPMATE is saved in the bin folder. Other settings and methods of use are the same as for direct use.

Compiling on a MacOS system

It is assumed that gcc and g++ compilers are already installed by default on MacOS systems:

>>tar -xvf VASPMATE-x.x.x.tar

>>cd VASPMATE-x.x.x.

>>make

After completing the above operations, if the following interface appears, it means the compilation is successful:

```
ion/QB/QB_dump.o  ./include/Energy_Minimization/QB/QB_function.o  ./include/Energy_Minimizat
ion/QB/QB_neighbor.o  ./include/Energy_Minimization/QB/QB_network.o  ./include/Energy_Minimi
zation/QB/QB_pbc.o  ./include/Energy_Minimization/QSPG/QSPG_func.o  ./include/Energy_Minimiz
ation/QSPG/QSPG_link.o  ./src/Evolutionary_src/Evolutionary_main.o  ./src/help_src/help.o -O
2 -std=c++0x  -I./spglib  -I./ -I./libmsym -L./lib -I./fftw/include        -L./fftw/lib -lf
ftw3 -lspg -lmsym -lstdc++
ld: warning: ignoring duplicate libraries: '-lc++'
Compiling VASPMATE
mv VASPMATE ./bin/
Compile successfully!
erye@EryedeMac VASPMATE_daily %
```

The installed VASPMATE is saved in the bin folder. Other settings and methods of use are the same as for direct use.

Compiling on a Windows system

If you need to use VASPMATE in a Windows environment, please install Cygwin software first. (Note: When installing Cygwin software, please also download the corresponding libraries for gcc, g++, libfltk, cmake, and make under Cygwin.) Then, navigate to the "home" folder under the installation path and copy the VASPMATE-x.x.x.tar file to this folder.

Launch the Cygwin64 Terminal program, and in the new window, enter:

>>tar -xvf VASPMATE-x.x.x.tar

>>cd VASPMATE-x.x.x.

>>make

If the following interface appears, it means the compilation is successful:

The installed VASPMATE.exe is saved in the "bin" folder. Other settings and methods of use are the same as for direct use (please ensure that *VASPMATE.exe* and the other three *.dll* files are in the same folder when using).

3 Syntax and Functionality

This manual will introduce the basic usage of VASPMATE command line, which currently includes generating input files (INCAR, POSCAR, POTCAR, KPOINTS) required for VASP operation, performing various operations and analyses on input structures and parameters, retrieving the required data from VASP output files (such as CONTCAR, OUTCAR, OSZICAR, XDATCAR, CHGCAR, WAVECAR, IBZKPT, EIGENVAL, PROCAR, DOSCAR, etc.), and conducting analysis, reprocessing, and property derivation. The general syntax format and rules of VASPMATE are as follows:

General Syntax:

VASPMATE <--mode/--module> (inputfile) (outputfile) <-option/-parameter> [list/value]

Description:

In this command line format, --mode/--module represents the label for the operational mode/ functional module, prefixed with "--", such as --prim, --cif2pos, --band, --dos. For different operational modes/ functional modules, VASPMATE provides corresponding default input filenames inputfile, such as INPOS and INCIF, and default output filenames outputfile, such as NEWPOS and NEWCIF. Note: When the user only provides one filename, and the operational mode/functional module requires both an input file and an output file, the filename corresponds to inputfile, while outputfile adopts the default filename. For example, VASPMATE --cif2pos INCIF is equivalent to VASPMATE --cif2pos INCIF NEWPOS. -option/-parameter represents the options/parameters label required for the operational mode/functional module, prefixed with "-", such as -opt (replace opt with the option label), -par (replace par with the parameter label), etc. List/Value represents the specific option list/parameter value. Note: When the VASPMATE command line only requires one type of option/parameter label, and both input and output filenames are readily provided, the label <-option/-parameter> can be omitted, and the option list/parameter value [list/value] can be listed directly after the filename. For example, VASPMATE --super

INPOS SUPERPOS -par 4 4 1 is equivalent to VASPMATE --super INPOS SUPERPOS 4 4 1. If the default input and output filenames are used after the operational mode/functional module label <--mode/--module>, the option/variable label <-option/-parameter> cannot be omitted. For example, VASPMATE --super INPOS SUPERPOS -par 4 4 1 is equivalent to VASPMATE --super -par 4 4 1, while VASPMATE --super 4 4 1 (XXX) is incorrect. Suggestion: When being uncertain whether the option/parameter label can be omitted, it is recommended to use the option/parameter label to avoid command line matching and assignment errors.

Pasted below is the outline of the operational modes and functional modules implemented in VASPMATE for reference (it can be viewed via VASPMATE --help). To be noted that there would be more options/parameters required for each operational mode or functional module, which will be discussed in more detail in the later sections.

```
erye@erye-virtual-machine:~/桌面$ VASPMATE --h
**************************************************************************************************
***                             The syntax format and rules for VASPMATE:                      ***
***                VASPMATE <--mode/--module> (inputfile) (outputfile) <-option/-parameter> [list/value]  ***
**************************************************************************************************
***-----------------------POSCAR-------------------------|*|------------------------INCAR------------------------***
*** VASPMATE --prim         # Generate Primitive cell     |*| VASPMATE --i/inp -option  # Load Templates              ***
*** VASPMATE --unit         # Generate Unit cell          |*|          option = rlx/stc/nsc/mds/sar/mag/opt/soc/hse/ ***
*** VASPMATE --super        # Generate Super cell         |*|          vdw/sic/ecc/bca/elf/fpm/dfp/neb/tsd/dos/pbs/  ***
*** VASPMATE --ieee         # Generate IEEE cell          |*|          bbs/bcd/scd/pcd/tlp/esp/wfn/tdm/los/...       ***
*** VASPMATE --vol          # Get cell volume             |*| VASPMATE --ia/id/irm/irp  # keyword/parameter revision ***
*** VASPMATE --cell         # Get cell info               |*| VASPMATE --is/istd # keyword/parameter standardization ***
*** VASPMATE --num          # Get atom number             |*| VASPMATE --iu/ldau # LDA+U setting                      ***
*** VASPMATE --atom         # Get atom info               |*| VASPMATE --im/imag # Magnetic moment setting            ***
*** VASPMATE --symm         # Get symmetry                |*| VASPMATE --iv/ivdw # VDW+D setting                      ***
*** VASPMATE --pos2cif/cif2pos   # Transform CIF&POS      |*|-----------------------KPOINTS-----------------------***
*** VASPMATE --fixc/fixa/fixe    # Fix atomic positions   |*| VASPMATE --k/km/kpt/kmesh  # Set KPOINTS by Input     ***
*** VASPMATE --cartes/direct     # Format Transformation  |*| VASPMATE --ka               # Auto set KPOINTS by KPPRA ***
*** VASPMATE --sortc/sorte       # Sort atom order        |*| VASPMATE --kv               # Auto set KPOINTS by KSPAC ***
*** VASPMATE --movc/movd         # Move atom position     |*|------------------------POTCAR-----------------------***
*** VASPMATE --proj/redef   # Cell Projection/Orientation |*| VASPMATE --pot         # Generate POTCAR via POSCAR   ***
*** VASPMATE --affine/alias # Cell Deformation            |*| VASPMATE --pote        # Generate POTCAR via Element  ***
**************************************************************************************************
***------------------------------------------Module-related------------------------------------------***
*** VASPMATE --std3d/std2d  # Standardize POSCAR for band |*| VASPMATE --3dka/3dkv/3dkm  # 3D band kpoints          ***
*** VASPMATE --ka3d/ka2d    # Create KPATH for band       |*| VASPMATE --3dbs            # 3D band derivation       ***
*** VASPMATE --band         # Band-Structure related      |*| VASPMATE --db              # Database Creation        ***
*** VASPMATE --dos          # DOS-related                 |*| VASPMATE --db2js           # Database Transformation  ***
*** VASPMATE --bader        # Bader-charge-related        |*| VASPMATE --thermo          # Thermo correction        ***
*** VASPMATE --neb          # NEB-related                 |*| VASPMATE --enth            # Formation enthalpy       ***
*** VASPMATE --vcd          # Charge-density-related      |*| VASPMATE --elas/elae       # Elastic Property         ***
*** VASPMATE --pcd          # Partial-charge-density      |*| VASPMATE --opti            # Optical Property         ***
*** VASPMATE --wfn          # Wave-function-related       |*| VASPMATE --magn            # Magnetic Property        ***
*** VASPMATE --fska/fskv/fskm  # Fermi surface kpoints    |*| VASPMATE --sto             # Stochastics Model        ***
*** VASPMATE --fsxd/fs         # Fermi surface derivation |*| VASPMATE --evo             # Evolutionary Model       ***
**************************************************************************************************
```

3.1 VASP basic operation

3.1.1 POSCAR

The POSCAR file contains the detailed information of crystal lattice, atomic coordinates, atomic constraints, and atomic velocities (MD). The following commands operate directly on the POSCAR file. Note that the parameters in parentheses represent the default arguments when they are not specified. For example, VASPMATE --prim implies that it will generate a primitive cell from the default INPOS file and save the newly created primitive cell to the PRIMPOS file by default.

Mode: **cif2pos 和 pos2cif**

Syntax:

VASPMATE --cif2pos file1(INCIF) file2(NEWPOS)

VASPMATE --pos2cif file1(INPOS) file2(NEWCIF)

Description:

These modes are used for the conversion between CIF format and POSCAR format. For cif2pos, file1 represents the filename in CIF format, and file2 represents the filename in POSCAR format; for pos2cif, it's the opposite. The default input file, file1, is INCIF or INPOS, and the output file, file2, is NEWPOS or NEWCIF. If the input file cannot be found, it will print a prompt message: No input file is found!

Examples:

VASPMATE --pos2cif

VASPMATE --pos2cif INPOS

VASPMATE --pos2cif INPOS NEWCIF

VASPMATE --cif2pos

VASPMATE --cif2pos INCIF

VASPMATE --cif2pos INCIF NEWPOS

Mode: **vol**

Syntax:

VASPMATE --vol file(INPOS)

Description:

This mode is used to quickly obtain the unit cell volume of a POSCAR-type file. The default input file, file, is INPOS, and the output information will be displayed below the terminal command line by default. If you need to save the output information to a specific file, you can use the output redirection symbol ">>" in the terminal to redirect the output information to the specified file, such as: VASPMATE --vol >> VOLCAR. If the input file cannot be found, it will print a prompt message: No input file is found!

Examples:

VASPMATE --vol INPOS

VASPMATE --vol

Mode: **num**

Syntax:

VASPMATE --num file(INPOS)

Description:

This mode is designed to quickly retrieve the total number of atoms in the unit cell of a POSCAR-type file. The default input file is INPOS. The output information will be displayed below the terminal command line by default. If users wish to save the output information to a specific file, they can utilize the output redirection symbol ">>" in the terminal to redirect the output to the designated file. For instance, VASPMATE --num >> NUMCAR. If the input file cannot be located, a prompt message will be printed: "No input file is found!"

Examples:

VASPMATE --num INPOS

VASPMATE --num

Mode: **cell**

Syntax:

VASPMATE --cell file(INPOS)

Description:

This mode is intended for quickly obtaining the unit cell structure information (lattice constants, angles between cell axes, and cell vectors) of a POSCAR-type file. The default input file is INPOS. The output information will be displayed below the terminal command line by default and saved in the CELLCAR file. If the input file cannot be found, a prompt message will be printed: "No input file is found!"

Examples:

VASPMATE --cell INPOS

VASPMATE --cell

Mode: **atom**

Syntax:

VASPMATE --atom file(INPOS)

Description:

This mode is designed for quickly retrieving the unit cell atomic information (element types and their quantities, coordinate types, and atomic coordinates) from a POSCAR file. The default input file is INPOS. The output information will be displayed below the terminal command line by default and saved in the ATOMCAR file. If the input file cannot be located, a prompt message will be printed: "No input file is found!"

Examples:

VASPMATE --atom INPOS

VASPMATE --atom

Mode: **prim**

Syntax:

VASPMATE --prim file1(INPOS) file2(PRIMPOS)

Description:

A primitive cell is the smallest repeating unit of a crystal structure, which can fully reflect the chemical structural characteristics of the internal atoms or ions distributed in three-dimensional space. This mode is primarily used to generate the primitive cell form of the current unit cell. The default input file, file1, is INPOS, and the output file, file2, is PRIMPOS. If the input file cannot be found, a prompt message will be printed: "No input file is found!"

Examples:

VASPMATE --prim

VASPMATE --prim INPOS

VASPMATE --prim INPOS PRIMPOS

Mode: **unit**

Syntax:

VASPMATE --unit file1(INPOS) file2(UNITPOS)

Description:

A conventional unit cell can better reflect the characteristics or symmetry of the crystal structure in many cases. This mode is mainly used to generate the conventional unit cell form of the current cell. The default input file, file1, is INPOS, and the output file, file2, is UNITPOS. If the input file cannot be found, a prompt message will be printed: "No input file is found!"

Examples:

VASPMATE --unit

VASPMATE --unit INPOS

VASPMATE --unit INPOS UNITPOS

Mode: **super**

Syntax:

VASPMATE --super file1(INPOS) file2(SUPEPOS) (-np) [sn1 sn2 sn3]

Description:

This mode will construct a repeated supercell of the current unit cell. The lattice vectors of the supercell are integer multiples of the lattice vectors of the input unit cell, where sn1, sn2, and sn3 represent the expansion factors. Please note that sn1, sn2, and sn3 must be integers and greater than or equal to 1. The default input file, file1, is INPOS, and the output file, file2, is SUPEPOS. If the input file cannot be found, a prompt message will be printed: "No input file is found!"

Examples:

VASPMATE --super 4 4 1 (XXX)

VASPMATE --super -np 4 4 1

VASPMATE --super INPOS SUPEPOS 4 4 1

VASPMATE --super INPOS SUPEPOS -np 4 4 1

Mode: **symm**

Syntax:

VASPMATE --symm file(INPOS)

Description:

Symmetry is the most fundamental characteristic of crystal structures. Using this operation will output basic symmetry information of the crystal, such as space group, point group, crystal system, etc. This mode is primarily used to analyze the symmetry of current cell. The default input file is INPOS. The output information will be printed below the terminal command line by default and saved in the SYMMCAR file. If the input file cannot be found, a prompt message will be printed: "No input file is found!"

```
*********************Find Symmetry******************
International Tables: 164
International: P-3m1
Hall symbol: -P 3 2"
Holohedry: Trigonal
Crystal System: -3m
Number: 164
Long Name: P -3 2/m 1
Schoenflies Names: D3d
****************************************************
```

Examples:

VASPMATE --symm

VASPMATE --symm INPOS

Mode: **affine**[4]

Syntax:

VASPMATE --affine file1(INPOS) file2(AFFPOS) -txx/-tyy/-tzz [strain]

VASPMATE --affine file1(INPOS) file2(AFFPOS) -sxy/-syz/-szx [strain]

VASPMATE --affine file1(INPOS) file2(AFFPOS) -pxy/-pyz/-pzx [strain]

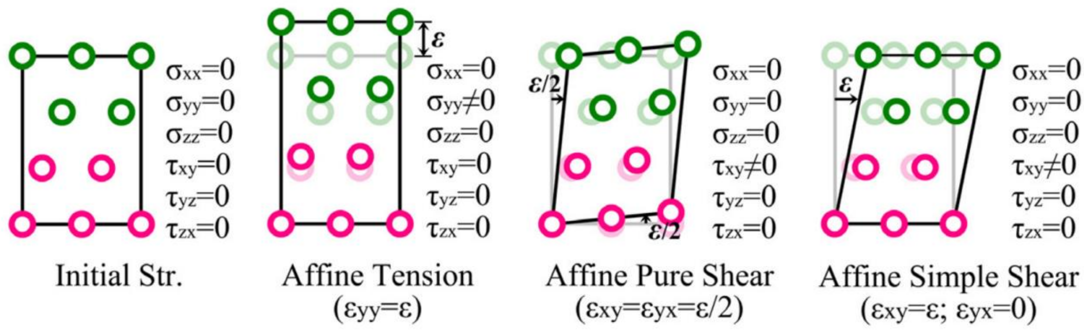
VASPMATE --affine file1(INPOS) file2(AFFPOS) -tension [xx/yy/zz] [init_strain step_length step_num]

VASPMATE --affine file1(INPOS) file2(AFFPOS) -simshear [xy/yz/zx] [init_strain step_length step_num]

VASPMATE --affine file1(INPOS) file2(AFFPOS) -purshear [xy/yz/zx] [init_strain step_length step_num]

Description:

These modes require the presence of the INPOS file in Direct (fractional) coordinates by default and the generated output file is AFFPOS. The flags -txx/-tyy/-tzz correspond to tension deformation, -sxy/-syz/-szx are for simple shear deformation, while -pxy/-pyz/-pzx stand for the pure shear deformation. The parameter [strain] corresponds to the related strain component of ε_{ij}, e.g. ε_{ij}=0.01 for 1%.



In the case of the flags -tension and -simshear/-purshear, they correspond to the uniaxial tension (tension), the simple shear (simshear) and the pure shear (purshear) for a series of deformations. The deformation options are divided into xx/yy/zz and xy/yz/zx, where the former xx/yy/zz correspond to the uniaxial tension modes, while the latter xy/yz/zx for shear mode. The parameter init_strain represents the initial strain value, the parameter step_length denotes the strain increment step, and the parameter step_num is the number of strain steps. By default, it generates AFFPOS_* files, where * represents the current strain value. For shear deformations of xy/yz/zx, they correspond to the deformation matrix components: ε_{xy} and ε_{yx}; ε_{yz} and ε_{zy}; ε_{zx} and ε_{xz}. Note: Since they are multiple option flags, the option flag -tension, -simshear and -purshear cannot be omitted.

$$R^{\text{def}} = R^{\text{ini}} \cdot (I + \varepsilon)$$

$$\varepsilon_{\text{tensile}} = \begin{bmatrix} \varepsilon_1 & 0 & 0 \\ 0 & \varepsilon_2 & 0 \\ 0 & 0 & \varepsilon_3 \end{bmatrix}$$

Where $\varepsilon_1 = \varepsilon_{xx}$, $\varepsilon_2 = \varepsilon_{yy}$, $\varepsilon_3 = \varepsilon_{zz}$ are in Voigt notation.

$$\varepsilon_{\text{pure}} = \begin{bmatrix} 0 & \frac{1}{2}\varepsilon_6 & \frac{1}{2}\varepsilon_5 \\ \frac{1}{2}\varepsilon_6 & 0 & \frac{1}{2}\varepsilon_4 \\ \frac{1}{2}\varepsilon_5 & \frac{1}{2}\varepsilon_4 & 0 \end{bmatrix} \quad \text{and} \quad \varepsilon_{\text{simple}} = \begin{bmatrix} 0 & \varepsilon_6 & \varepsilon_5 \\ 0 & 0 & \varepsilon_4 \\ 0 & 0 & 0 \end{bmatrix}$$

Where $\varepsilon_4 = \varepsilon_{yz} + \varepsilon_{zy}$, $\varepsilon_5 = \varepsilon_{zx} + \varepsilon_{xz}$, $\varepsilon_6 = \varepsilon_{xy} + \varepsilon_{yx}$ are in Voigt notation.

Examples:

VASPMATE --affine -txx 0.01 # ε_{xx}=0.01

VASPMATE –affine INPOS -sxy 0.01 # ε_{xy}=0.01

VASPMATE --affine INPOS -pxy 0.01 # ε_{xy}=0.01

VASPMATE --affine -simshear xy 0 0.1 10

VASPMATE --affine INPOS -purshear xy 0 0.1 10

VASPMATE --affine -tension xx 0 0.1 10

VASPMATE --affine INPOS AFFPOS -tension xx 0 0.1 10

Mode: **alias**[4]

Syntax:

VASPMATE --alias file1(INPOS) file2(ALIPOS) -txx/-tyy/-tzz [strain] [position]

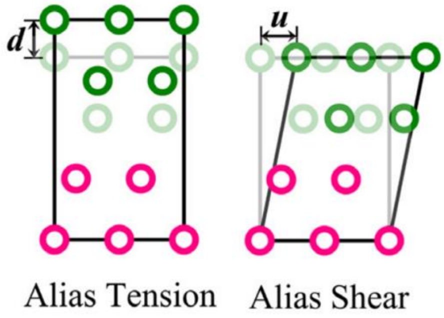
VASPMATE --alias file1(INPOS) file2(ALIPOS) -sxy/-syz/-szx [strain1] [strain2] [position]

VASPMATE --alias file1(INPOS) file2(ALIPOS) -tensi [xx/yy/zz] [istart iend ispac] [position]

VASPMATE --alias file1(INPOS) file2(ALIPOS) -shear [xy/yz/zx] [istart1 iend1 ispac1] [istart2 iend2 ispac2] [position]

Description:

These modes require the presence of the INPOS file in Cartesian coordinates by default and the generated output file is ALIPOS. The flag -txx/-tyy/-tzz correspond to tension deformation, while -sxy/-syz/-szx are for simple shear deformation. The parameter [strain] is specified according to the displacement in unit of Angstrom: *d* stands for the tension displacement and *u* is for the shear displacement (see the attached Figure).



In the case of the flags of -tensi and -shear, they correspond to uniaxial tension (tension) and simple shear (shear) for a series of displacements. For the tension (-tensi) deformation, the options [xx/yy/zz] represents the deformation direction, the parameter [istart] is the initial strain in that direction, [iend] is the final strain, and [ispac] is the strain step size for each increment. The total strain value for each step is istart + ispac * k, where k = 0, 1, 2, 3, ... until istart + ispac * k > iend. Similarly, for the shear (-shear) deformation, the options [xy/yz/zx] represent the deformation plane, and they correspond to the deformation matrix components: ε_{yz} and ε_{zx}; ε_{xy} and ε_{zx}; ε_{xy} and ε_{yz} respectively. The parameters [istart1] and [istart2] determine the initial strains in the two directions of the plane, [iend1] and [iend2] are the final strains in those two directions, and [ispac1] and [ispac2] are the steps for each increment in the two directions. The total strain values for each step are istart1+i×ispac1 and istart2+j×ispac2, where i=0,1,2,3…, j=0,1,2,3… until istart1+i×ispac1>iend1, istart2+j×ispac2>iend2. The parameter [position] specifies the position of the separation interface in Cartisian coordinate (UNIT: Angstrom), corresponding to their respective deformation modes. For example, the tension along the xx direction corresponds to the position along the x-axis, while the shear on the xy plane corresponds to the position along the z-axis.

Tip: After using alias deformation operations, it is often necessary to fix the atomic positions. For example, during shear, all atoms perpendicular to the deformation plane need to be relaxed, while the atoms in other two directions need to be fixed. At this time, one can use the fix mode to uniformly relax or fix atomic coordinates.

These modes generates ALIPOS_*(tensi) or ALIPOS_*_*(shear) files by default, where * represents the current strain values. Note: Since they are multiple option flags, the flags -tensi/-shear cannot be omitted.

Examples:

VASPMATE --alias -txx 0.5 #The displacement is 0.5 Angstrom

VASPMATE --alias INPOS -sxy 1 1 2 #1 Angstrom, 0.5 Angstrom, position=2A

VASPMATE --alias -tensi xx 0 10 1 0 # from 0 A to 10 A with step of 1A

VASPMATE --alias INPOS -shear xy 0 10 1 1 10 2 5 # position= 5A

VASPMATE --alias INPOS ALIPOS -shear xy 1 10 1 1 10 1 5 # position= 5A

Mode: **proj**

Syntax:

VASPMATE --proj file1(INPOS) file2(PROJPOS) -rot [rotx roty rotz]

VASPMATE --proj file1(INPOS) file2(PROJPOS) -ind [pvh pvk pvl] [uvu uvv uvw]

VASPMATE --proj file1(INPOS) file2(PROJPOS) -mat [mat11 mat12 mat13 mat21 mat22 mat23 mat31 mat32 mat33]

Description:

These modes project a specific direction of the given crystal lattice (i.e., the direction in which strain is applied) to align parallel to an axial vector (such as y) facilitating the application of uniaxial tensile deformation; or it projects the normal of the crystal lattice's slip plane to be parallel to one axial vector (such as x) and the shear direction parallel to another axial vector (such as y), to facilitate the application of shear deformation for a specific slip system.

The option -rot specifies the rotation method and requires three rotation angles: rotx, roty, rotz.

The option -ind is the crystallographic index method, demanding a reciprocal lattice vector perpendicular to the real-space plane (pvh, pvk, pvl), defined as [pvh, pvk, pvl]*, and a lattice vector [uvu, uvv, uvw]. It is mandatory that the lattice vector [uvu, uvv, uvw] lies in the plane (pvh, pvk, pvl), ensuring that pvh·uvu + pvk·uvv + pvl·uvw

= 0. Note that pvh, pvk, pvl, uvu, uvv, and uvw must be integers.

The option -mat defines the generalized projection method, where mat is a normalized orthogonal 3x3 matrix M. This matrix satisfies the relation R = R'· M with the original atomic basis vector matrix R and the projected atomic basis vector matrix R'. This matrix can be obtained from the visualization program VESTA.

The default input file for this mode is INPOS, and the output file is PROJPOS. Note: Multiple option flags can be used here, and one cannot omit the option flags -rot/-ind/-mat.

Examples:

VASPMATE --proj -rot 30.00 0.00 30.00

VASPMATE --proj INPOS -rot 30.00 0.00 30.00

VASPMATE --proj -ind 1 1 1 1 1 -2

VASPMATE --proj INPOS PROJPOS -ind 1 1 1 1 1 -2

VASPMATE --proj -mat 0.5774 0.5774 0.5774 0.4082 0.4082 -0.8165 -0.7071 0.7071 0.0000

Mode: **redef**

Syntax:

VASPMATE --redef file1(INPOS) file2(REDEPOS) (-par) [vect11 vect12 vect13 vect21 vect22 vect23 vect31 vect32 vect33]

VASPMATE --redef file1(INPOS) file2(REDEPOS) (-par) [vect11 vect12 vect13 vect21 vect22 vect23 vect31 vect32 vect33] [a xx b xy]

Description:

This mode redefines the current cell according to the specified three crystallographic vectors: [vect11 vect12 vect13 vect21 vect22 vect23 vect31 vect32 vect33] and then determine the crystal orientation based on the last four parameters: [a xx b xy]. In the last four parameters, a and b are the cell axis of the redefined cell, which take the symbols of a, b or c, xx and xy correspond to the Cartesian coordinate axis with xx=[x, y, z] and plane xy=[xz, xy, yz] respectively (note that these are all in lowercase). For example, the symbols "a x b xy" indicate that the new cell axis a will be parallel to the x-axis of Cartesian coordinate system, and the cell axis b will be in the xy plane. The default input file1 is INPOS, and the output file2 is REDEPOS. Note: The flag -par can be omitted if both input and output filenames are provided. Conversely, if the input and output filenames are not listed (i.e., the default filenames are used or only one filename is provided), the flag -par cannot be omitted.

Examples:

VASPMATE --redef -par 1 -1 0 1 1 -2 2 2 2 a x b xy

VASPMATE --redef INPOS -par 1 -1 0 1 1 -2 2 2 2 a x b xy

VASPMATE --redef INPOS REDEPOS 1 -1 0 1 1 -2 2 2 2 c x a yz

VASPMATE --redef INPOS REDEPOS -par 1 -1 0 1 1 -2 2 2 2 c x a yz

VASPMATE --redef -par 1 -1 0 1 1 -2 1 1 1

Mode: **ieee**

Syntax:

VASPMATE --ieee file1(INPOS) file2(IEEEPOS)

Description:

This mode converts a 2D or 3D structure into the IEEE format as required by reference[5]. The default input file, file1, is INPOS, and the output file, file2, is IEEEPOS. If the input file cannot be located, a prompt message will be printed: "No input file is found!"

Examples:

VASPMATE --ieee

VASPMATE --ieee INPOS

VASPMATE --ieee INPOS IEEEPOS

Mode: **fix**

Syntax:

VASPMATE --fixc file1(INPOS) file2(FIXPOS) (-par) [axis] [m n] [fix-tag]

VASPMATE --fixa file1(INPOS) file2(FIXPOS) (-par) [number] [fix-tag]

VASPMATE --fixe file1(INPOS) file2(FIXPOS) (-par) [element1] [fix-tag] / [element2] [fix-tag] / …

VASPMATE --ufix file(INPOS)

Description:

These modes are primarily used for fixing atomic positions according to requirements and is divided into three types of fixing methods. The first mode, --fixc, fixes atoms based on the components of their coordinates; the parameter [axis] specifies the selected cell axis a, b, or c, and the parameters [m n] are the minimum and maximum values along cell axis, respectively. Atoms whose axis coordinate component lies between [m n] are fixed in a certain direction, for example, F F F. The second mode, --fixa, fixes atoms based on their atomic order number, with the last three parameters being T (True) or F (False). The third mode, --fixe, fixes atoms based on the element type; the parameters of [element1] and [element2] are the names of elements, with the last three parameters being T or F. The final mode, --ufix, clean any fixes, allowing all atoms to relax during simulation. The default input file/file1 is INPOS, and the output file2 is FIXPOS.

Examples:

VASPMATE --fixc c 2 10 F F F (XXX)

VASPMATE --fixc -par c 2 10 F F F

VASPMATE --fixc INPOS FIXPOS c 2 10 F F F

VASPMATE --fixc INPOS FIXPOS -par c 0 0.5 F F F

VASPMATE --fixa INPOS FIXPOS 2-5 F F T

VASPMATE --fixa INPOS -par 2-5 F F T

VASPMATE --fixe INPOS FIXPOS B F F T / N F F F

VASPMATE --fixe INPOS FIXPOS -par B F F T / N F F F

VASPMATE --ufix INPOS

Mode: **direct or cartes**

Syntax:

VASPMATE --cartes file1(INPOS) file2(NEWPOS)

VASPMATE --direct file1(INPOS) file2(NEWPOS)

Description:

These modes are primarily used for the conversion of fractional and Cartesian coordinates in a POSCAR file. The mode flag --cartes represents the conversion of the current structure into Cartesian coordinates, while the flag --direct represents the conversion of the current structure into fractional coordinates. This mode does not require specifying the initial structure's coordinate format; VASPMATE will automatically determine it. The default input file1 is INPOS, and the output file2 is NEWPOS.

Examples:

VASPMATE --cartes

VASPMATE --direct INPOS NEWPOS

Mode: **sortc/sorte**

Syntax:

VASPMATE --sortc file1(INPOS) file2(SORTPOS) (-par) [axis]

VASPMATE --sorte file1(INPOS) file2(SORTPOS) (-par) [element-list]

Description:

These modes are primarily used for sorting atoms in a unit cell. The mode flag "--sortc" is used to rearrange the order of different types of atoms in the POSCAR file based on the magnitude of their coordinate components along a selected cell axis, with the default one of c-axis. The mode flag "--sorte" is used to rearrange different types of atoms in the POSCAR file according to the order specified in the element-list, with the default in alphabetical order. The default input file1 is INPOS, and the output file2 is SORTPOS.

Examples:

VASPMATE --sortc c (XXX)

VASPMATE --sortc -par c

VASPMATE --sortc INPOS SORTPOS c

VASPMATE --sorte -par B N

VASPMATE --sorte INPOS SORTPOS B N

VASPMATE --sorte -par Ti B N

Mode: **movc/movd**

Syntax:

VASPMATE --movc file1(INPOS) file2(MOVEPOS) (-par) [axis] [min max] [dx dy dz]

VASPMATE --movd file1(INPOS) file2(MOVEPOS) (-par) [axis] [min max] [dx dy dz]

Description:

These modes are used for directional movement of atoms in the unit cell. The mode flag --movc/--movd represent movement distances relative to Cartesian coordinates (cartes) and fractional coordinates (direct), respectively. The parameters of dx, dy, and dz represent the distances moved in the three directions along cell axis a, b and c. The atoms to be moved are selected based on their coordinate components falling within the range [min, max] along the cell axis a, b or c. It's important to note that the values of these coordinate components should be consistent with the selected mode --movc/--movd. The default input file is INPOS, and the output file is MOVEPOS.

Examples:

VASPMATE --movc c 0 10 1 2 3 (xxx)

VASPMATE --movc -par c 0 10 1 2 3

VASPMATE --movd -par c 0 1 0.1 0.2 0.3

VASPMATE --movd -par c 0 1 0.1 0.2 0.3

VASPMATE --movd INPOS MOVEPOS c 0 1 0.1 0.2 0.3

VASPMATE --movd INPOS MOVEPOS -par c 0 1 0.1 0.2 0.3

Mode: **vel/velocity**

Syntax:

VASPMATE --vel/velocity file(INPOS) (-T) [temperature]

Description:

This mode is used to set the velocity of each atom for AIMD simulations according to the specified temperature based on the Maxwell-Boltzmann distribution. The parameter flag -T represent the temperature in unit of Kelvins with the default value of 300K. The default input file is INPOS, and the output file is VELOPOS.

Examples:

VASPMATE --vel

VASPMATE --vel INPOS -T 300

VASPMATE --vel -T 300

3.1.2 INCAR

Mode: **incar parameter files**

Syntax:

VASPMATE --i/--inp [option-list]

Description:

For ease of use, VASPMATE provides INCAR template files with default parameters for different computational tasks. The option names provided by [option-list] are abbreviated commands for computational tasks. The generated file format is incar_*… Please refer to the table below for detailed information.

Commands	Type	Computational tasks
rlx/lar	Basic	Lattice & Atomic Relaxation
ssc/stc	Basic	Static Self-consistent Calculation
nsc	Basic	Non-Self-consistent Calculation
mds	Basic	Molecular Dynamic Simulation
sar	Basic	Standard Atomic Relaxation
mag/mpc	Optional	Magnetic Property Calculation
opt/opc	Optional	Optical Property Calculation
soc	Optional	Spin-Orbit Coupling Calculation
hse	Optional	Hybrid function HSE06
vdw	Optional	VDW Correction DFT+D3
sic	Optional	Self-Interaction Correction LDA+U

ecc	Full	Elastic Constants Calculation
bca	Optional	Bader Charge Analysis
elf	Optional	Electron Localization Function
fpm	Optional	Frozen Phonon Method
dfp	Optional	Density functional Perturbation
neb	Full	Nudged Elastic Band method
dim/tss	Full	Transition State Searching: Dimer
dos	Optional	Density of State
pbs	Full	Projected Band Structure
bbs	Full	Basic Band Structure
bcd	Optional	Basic Charge Density
scd	Optional	Spin Charge Density
pcd	Optional	Partial Charge Density
tlp	Optional	Total Local Potential
esp	Optional	Electrostatic Potential
wfn	Optional	Wave Function Calculation
tdm	Optional	Transition Dipole Moment
los	Optional	Linear Optical Spectrums

Notice that the Optional Types should be used together with basic Types.

Examples:

VASPMATE --i rlx

VASPMATE --inp rlx

VASPMATE --i stc sic hse

Mode: **incar parameter revision**

Syntax:

VASPMATE --ia [keyword] [value] / [keyword] [value] / …] (--i_app or --i_append)

VASPMATE --id [keyword-list] (--i_del or --i_delete)

VASPMATE --irm [keyword-list] (--i_rem or --i_remove)

VASPMATE --irp [keyword] [value] / [keyword] [value] / …] (--i_rep or --i_replace)

Description:

This mode is primarily used for the on-site modification of INCAR file, and it requires the presence of an INCAR file in the current directory. The options --ia and --irp are used to modify the value of a corresponding keyword in INCAR and append a corresponding value at the end of the file, respectively. The option --id is used to delete the line containing a specific keyword, while --irm is used to comment out the line where the keyword is located. The keyword refers to the various parameters provided by INCAR, and the value corresponds to the parameter value of the keyword, which can be a single or multiple numbers and strings separated by spaces. For --ia and --irp, if multiple parameters are appended consecutively, they need to be separated by the delimiter "/".

Examples:

VASPMATE --ia EINT 2 5 / LORBIT 11

VASPMATE --id NSW ENCUT

VASPMATE --irm NSW ENCUT

VASPMATE --irp EINT 2 5 / LORBIT 11

Mode: **incar parameter standardization**

Syntax:

VASPMATE --is/--istd (--i_std or --i_standard)

Description:

This mode is primarily used for standardizing the INCAR file, which requires the presence of an INCAR file in the current directory. Standardization operations include removing duplicate keyword lines, deleting commented keyword lines, rearranging and blocking keyword lines (such as initialization of ISTART, ICHARG, LWAVE, and LCHARG; file input/output; structural relaxation; electronic self-consistency; magnetism-related; optics-related; energy-band-related; DOS-related; hybrid functional-related; VDW-related; LDA+U-related, etc.). By default, the standardized INCAR is saved in the STDINP file.

Examples:

VASPMATE --is

VASPMATE –istd

Mode: **Default on-site Coulomb interaction (LDA+U) parameters**

Syntax:

VASPMATE --iu/ldau (INPOS) (-a)

VASPMATE --iu/ldau (INPOS) (-t) [Table]

Description:

This mode is used to automatically set the corresponding INCAR parameters for VASP calculations that require LDA+U correction based on the element types in INPOS, such as LDAL, LDAU, LDAJ, and other parameters. The -t/-tab flag indicates the use of user-defined element U values, where [Table] is a custom file (such as ldau.table) that stores the default U values for elements. The content format is [element L U J], for example, Mo 2 2.4 0. In this case, VASPMATE will prioritize to use the parameters provided in the file. If the -t flag is not specified or is specified as -a, VASPMATE will use the default provided U values. The default input file is INPOS, and the INCAR modified based on LDA+U is saved in the NEWINP file.

Examples:

VASPMATE --iu

VASPMATE --iu -a

VASPMATE --iu INPOS -a

VASPMATE --ldau -t ./ldau.table

VASPMATE --ldau INPOS ./ldau.table

VASPMATE --ldau INPOS -t ./ldau.table

Mode: **Default magnetic moment parameters**

Syntax:

VASPMATE --im/imag (INPOS) (-a) (-clm/-afm/-sfm/-sfmw/-nlm)

VASPMATE --im/imag (INPOS) -t [Table] (-clm/-afm/-sfm/-sfmw/-nlm)

Description:

This mode is designed to automatically set the corresponding INCAR parameters for VASP calculations that require adding magnetic moments based on the element types in INPOS, such as ISPIN, MAGMOM, and other parameters. The -t parameter allows users to customize the magnetic moments of elements, where [Table] is a custom file (e.g., magmom.table) that stores the default magnetic moments for elements. The content format is [element magmom], for instance, Fe 2.00. In this case, VASPMATE will prioritize to use the parameters provided in the file. If not specified or specified as -a, VASPMATE will adopt its default parameters. The magnetic reference formula is as follows:

$$MAGMOM = \begin{cases} NIONS * 1.0 \ \ ISPIN = 2 \\ 3 * NIONS * 1.0 \ LNONCOLLINE = .TRUE. \end{cases}$$

Note: When using a custom file [Table], the -t flag cannot be omitted (this operation mode is not a single option flag). The -clm flag is used to automatically set the magnetic moments in the INCAR file based on INPOS for collinear magnetic calculations. By default, the configured INCAR is saved in the NEWINP file. For antiferromagnetic and ferrimagnetic calculations, based on INPOS, a series of supercell named POS_afm_001(002...) and INCAR files named incar_afm_001(002...), ferrimagnetic state (the -sfm flag represents setting based on elements) incar_sfm_001(002...), and ferrimagnetic state (the -sfmw flag represents setting based on the Wyckoff sites of atoms) incar_sfmw_001(002...) are generated. When generating a series of INCAR files, a magnetic moment setting list file MAGMOM_list is also created to note the correspondence with the newly generated INCAR files. For non-collinear magnetic calculations, based on INPOS, a series of supercell named POS_afm_001(002...) and INCAR files named incar_nlm_001(002...) are generated, similar to the antiferromagnetic calculations above, but requiring more INCAR configurations for different magnetic states.

Examples:

VASPMATE --im

VASPMATE --im -clm

VASPMATE --im -a

VASPMATE --im -a -clm

VASPMATE --im INPOS -a -clm

VASPMATE --imag -t ./mag.table

VASPMATE --imag -t ./mag.table -clm

VASPMATE --imag INPOS -t ./mag.table -clm

VASPMATE --im -afm

Mode: **Default VDW dispersion correction parameters**

Syntax:

VASPMATE --iv/ivdw (INPOS) (-a)

VASPMATE --iv/ivdw (INPOS) (-t) (d3b/d2/d3z/b86/b88/pbe/rpbe)

Description:

This mode is used to determine whether a structure belongs to a 2D system based on structural information in INPOS, such as z-axis height, atomic spacing, and vacuum layer thickness. It then decides whether to add VDW correction accordingly. By default, the corresponding INCAR parameter is set to [IVDW = 12]. The INCAR with added VDW correction is saved in the NEWINP file by default.

Examples:

VASPMATE --iv

VASPMATE --iv -a

VASPMATE --iv d3z (XXX)

VASPMATE --iv -t d3z

VASPMATE --iv INPOS d3z

VASPMATE --iv INPOS -t d3z

VASPMATE --ivdw INPOS -t d3z

3.1.3 KPOINTS

This section will introduce commonly used methods for generating and setting

KPOINTS. For the energy band calculations on specific high-symmetry point paths, please refer to the subsequent chapters on energy band calculations.

Mode: **According to INPUT or POSCAR**

Syntax:

VASPMATE --k/km/kpt/kmesh (-par) [k1 k2 k3] [kmesh] [kscheme]

VASPMATE --ka file(INPOS) (-par) [kppra] [kscheme]

VASPMATE --kv file(INPOS) (-par) [kspac] [kscheme]

Description:

This mode is used to create the KPOINTS file required for non-self-consistent energy band calculations, with the default output file being NEWKPT. In the VASPMATE program, there are three specified methods for generating KPOINTS: KMESH, KPPRA, and KSPAC.

VASPMATE includes the general KMESH method for determining KPOINTS, which involves manually setting the values of k1, k2, and k3. Note: The KMESH setting method (--k/km/kpt/kmesh) does not require an input file. The KPPRA setting method (--ka) requires an input file, INPOS. It is a method that automatically sets the k-point grid while maintaining a constant k-point density. VASPMATE strictly calculates the reciprocal lattice vector ratio, and the number of k-points on each reciprocal lattice vector axis is proportional to $|\boldsymbol{a_1} \cdot (\boldsymbol{a_2} \times \boldsymbol{a_3})|/|\boldsymbol{a_2} \times \boldsymbol{a_3}|$ where $\boldsymbol{a_2}$ and $\boldsymbol{a_3}$ are the other two reciprocal vectors. The KSPAC setting method (--kv) also requires the INPOS input file. It is another method that automatically sets the k-point grid to make the grid as uniform as possible. The number of k-points on the three reciprocal lattice vectors is calculated using the formula $|a_i|/KSPAC$. The Kscheme specifies the type of grid used, which is typically divided into Gamma shift ("Gamma") and Monkorst-Pack ("MP").

For each setting option, VASPMATE provides default parameters. Even without setting the sprinkling accuracy or grid type, it can still generate a relatively accurate k-point file based on the default parameters, improving the flexibility of use. For KPPRA and KSPAC, the INPOS input file is required, and the suggested default parameter values are 1000 and 0.5, respectively (for energy band and density of states calculations, it is recommended to increase the accuracy appropriately). The generated k-point file are named NEWKPT (to facilitate distinguishing, for calculating the energy band structure on the high-symmetry point path, it is named NEWKPATH, referring to the subsequent energy band calculation section).

Examples:

VASPMATE --k 7 7 1 G

VASPMATE --k -par 7 7 1 G

VASPMATE --kpt -par 7 7 1 G

VASPMATE --ka 8000 G (XXX)

VASPMATE --ka INPOS 8000 G

VASPMATE --ka INPOS -par 8000 G

VASPMATE --kv 0.5 G (XXX)

VASPMATE --kv -par 0.5 G

VASPMATE --kv INPOS 0.5 G

3.1.4 POTCAR

This section will introduce the operation of selecting and merging the appropriate type of POTCAR according to the requirements.

Mode: **According To File**

Syntax:

VASPMATE --pot file(INPOS)

VASPMATE --pot file(INPOS) -type (-suff postfix1 postfix2…)

VASPMATE --pot file(INPOS) -t type (-suff postfix1 postfix2…)

(type = LDA, GGA, PBE)

(suffix/postfix = s, d, h, sv, pv, GW)

Description:

VASPMATE will extract and merge POTCAR files from the pseudopotential library based on the element types in INPOS to generate NEWPOT. However, this requires the user to copy the POTCAR files for the corresponding elements to the current directory, or create a file named .potpath in the ~/ directory and enter the path to the pseudopotential library in the following format (note the case sensitivity):

```
PBE_PATH = /vol-th/home/zrfcms1/POTPBE54
```

VASPMATE supports to merge three types of pseudopotential: PBE, LDA, and GGA. The default type is the PBE pseudopotential without a suffix. The "postfix" refers to the suffix options for the pseudopotential, such as s, h, sv, pv, etc. Note: Since this operation mode is not a single option tag, the tag -t cannot be omitted.

Examples:

VASPMATE --pot

VASPMATE --pot PBE (XXX)

VASPMATE --pot -PBE

VASPMATE --pot -t PBE

VASPMATE --pot INPOS PBE (XXX)

VASPMATE --pot INPOS -t PBE

VASPMATE --pot -PBE -suff s

VASPMATE --pot -t PBE -suff sv GW

VASPMATE --pot INPOS -PBE -suff sv GW

VASPMATE --pot INPOS -t PBE -suff sv GW

Mode: **According To Elements**

Syntax:

VASPMATE --pote -type [elem1 elem2...] (-suff [postfix1 postfix2…])

VASPMATE --pote -t [type] -e [elem1 elem2...] (-suff [postfix1 postfix2…])

(type = LDA, GGA, PBE)

(elem = B, N, C …)

(postfix = s, d, h, sv, pv, GW)

Description:

In this mode, VASPMATE will generate corresponding POTCAR files (NEWPOT) not only based on the element types in the POSCAR file but also according to user-specified elements. Similarly, users need to copy the POTCAR files for the corresponding elements to the current directory, or create a file named .potpath in the ~/ directory and enter the path to the pseudopotential library. The default type is the PBE pseudopotential. Note: Since this mode is not a single option tag, the tags -t/-type and -e cannot be omitted.

Examples:

VASPMATE --pote B N C (XXX)

VASPMATE --pote -PBE B N C

VASPMATE --pote -t PBE -e B N C

VASPMATE --pote -PBE B N C -suff sv GW

VASPMATE --pote -t GGA -e B N C -suff sv GW

VASPMATE --pote -GGA -e B N C -suff sv GW

3.1.5 Input Consistency Check

Mode: **file check**

Syntax:

VASPMATE --check (-in) (inp/pos/kpt/pot)

Description:

VASPMATE makes a judgment on whether the current conditions meet the requirements for submitting a task for VASP calculation based on the files in the current folder.

Since VASP does not automatically check whether the element types in the POSCAR file match those in the POTCAR file during calculation, even if the pseudopotential of the H element is used when calculating graphene, VASP will not report an error but continue the calculation. However, the results are definitely inconsistent. VASPMATE automatically checks if the current pseudopotential elements are consistent and promptly prints an error: "Attention: Element is not consistently matched!"

VASPMATE checks if the four files required for VASP operation exist in the current folder: INCAR, POSCAR, KPOINTS, and POTCAR. If any of these files are missing or empty, it provides the following prompt: "Attention: No/Empty file is found!", where file=INCAR/POSCAR/KPOINTS/POTCAR.

VASPMATE also checks for the necessary input files based on the control parameters in INCAR: when ISTART equals 1, 2, or 3, it checks for the presence of WAVECAR and alerts "Attention: ISTART = 1, But No WAVECAR is found!" if missing. Similarly, when ICHARG equals 11, 12, or 13, it verifies the existence of CHGCAR and warns "Attention: ICHARG = 11, But No CHGCAR is found!" if absent.

VASPMATE performs consistency checks based on INCAR control parameters, such as: (1) consistency between DFT+U keywords LDAUL/LDAUU/LDAUJ and POSCAR elements, (2) selectivity check for DFT+D keywords IVDW and LUSE_VDW, (3) structural optimization and static self-consistent parameter consistency checks, like ISIF=3 conflicting with NSW=0, (4) consistency between the magnetic keyword MAGMOM and POSCAR elements, and (5) non-self-consistent parameter consistency checks, where ISTART=1 and ICHARG=11 require reading WAVECAR and CHGCAR files.

Examples:

VASPMATE --check

VASPMATE --check -in

VASPMATE --check -in inp

VASPMATE --check -in pos

VASPMATE --check inp

VASPMATE --check pos

VASPMATE --check kpt

VASPMATE --check pot

3.1.6 Output Convergence Check

Mode: **Convergence check**

Syntax:

VASPMATE --check (-out) (rlx/stc/mds)

Description:

This mode is used for convergence checks after VASP calculation tasks are completed, focusing primarily on structural optimization (rlx) and static self-consistent field (scf) calculations. The generated results will be stored in the file named "Converge.data".

For structural optimization, VASPMATE determines whether the calculation uses an energy convergence criterion or a force convergence criterion based on the EDIFFG parameter in the OUTCAR file. In cases where force convergence is successful, i.e., when EDIFFG is less than 0, VASPMATE will print the following message to judge if the convergence criterion is met by the largest force on the structure during the calculation process being lower than the absolute value of EDIFFG.

1 >> Energy Converged!

2 >>Force Converged!

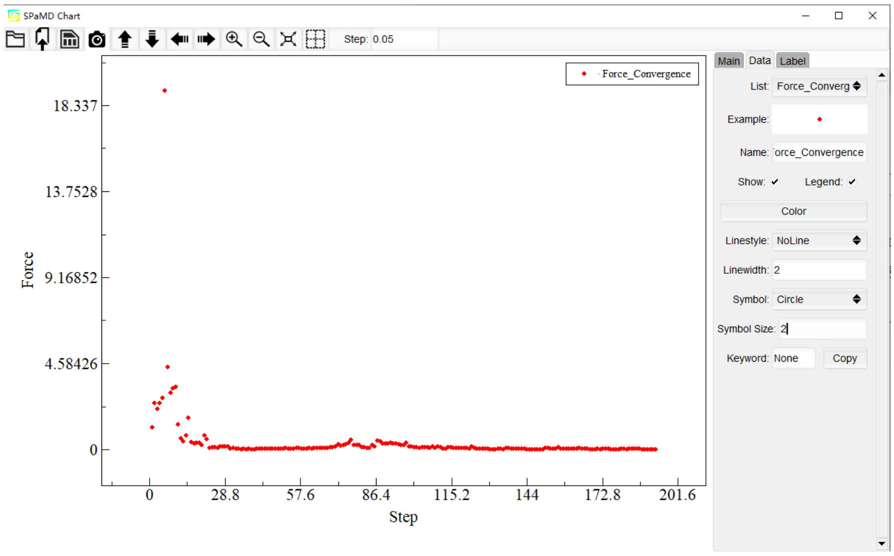
3 >> Energy not Converged!

4 >> Force not Converged!

Calculation details:

```
EDIFFG = -0.020000 in this calculation!
EDIFFG < 0, VASPMATE will calculate force convergence!
The calculation force result has successfully converged!
Written "Force_Convergence.spco" file!
```

It also generates a file named "Force_Convergence.spco" for import into SPaMD to observe the variation of structural forces with ionic steps, as shown below. (Note that the force on fixed atoms is uniformly set to zero).

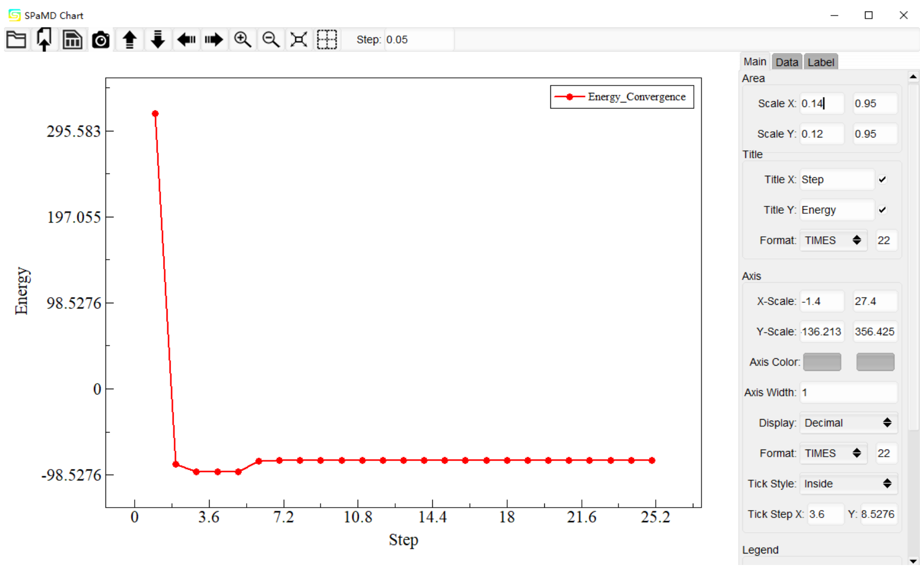


For energy convergence, that is when EDIFFG is greater than 0, VASPMATE will output the following message to determine if the convergence criterion is met by the energy difference between the last two ionic steps being lower than EDIFFG.

Calculation details:

```
EDIFFG = 1E-05 in this calculation!
EDIFFG > 0, VASPMATE will calculate Energy convergence!
The calculation energy result has successfully converged!
Written "Energy_Convergence.spco" file!
```

Additionally, it generates an "Energy_Convergence.spco" file for importing into SPaMD to observe the change in system energy with ionic steps, as shown below.



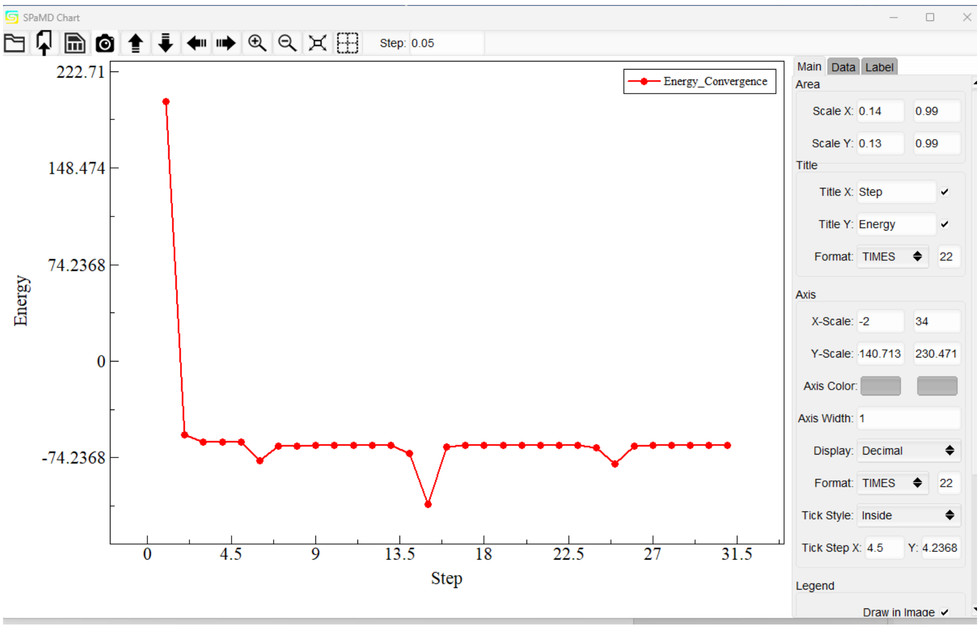
For static self-consistent field calculations, VASPMATE initially reads the EDIFF parameter from the OUTCAR file and outputs the energy variation with each electronic step, determining the convergence criterion by ensuring that the energy difference between the last two electronic steps is less than EDIFF.

```
EDIFF = 1.00000e-07 in this calculation!
The calculation energy result has successfully converged!
Written "Energy_Convergence.spco" file!
```

The generated "Energy_Convergence.spco" file can also be imported into SPaMD to observe the variations of system energy with electronic steps.

For molecular dynamics simulation, VASPMATE first reads the EDIFF parameter in OUTCAR and reads the XDATCAR file to output the energy variation with each time step.

The generated Energy_Convergence.spco file can also be imported into SPaMD to observe the system energy variation over time steps, as shown below.



Examples:

VASPMATE --check -out

VASPMATE --check -out rlx

VASPMATE --check -out stc

VASPMATE --check -out mds

VASPMATE --check rlx

VASPMATE --check stc

VASPMATE --check mds

3.1.7 Calculation Clean

Mode: **files clean**

Syntax:

VASPMATE --clean (-ssc/-nsc/-rlx/-mds/-dos/-band/-chg/-elf)

Description:

After each DFT calculation, this mode will be used make clean by removing the unnecessary files in the current folder. The flag -ssc is used for cleaning after the static self-consistent calculation is performed, -nsc is used for the non-self-consistent calculation, -rlx is used for the geometry optimization and atomic relaxation, -mds is for the MD simulations. If there is no parameter after the flag "--clean", VASPMATE will clean all the unnecessary files except for INCAR, POSCAR, KPOINTS, POTCAR and bash script.sh (.ksh).

Examples:

VASPMATE --clean

VASPMATE --clean -rlx

VASPMATE --clean -mds

VASPMATE --clean -dos

3.2 Band structure calculation

The calculation of energy band structure requires the preparation of a normalized primitive cell and a high-symmetry K-point path (K-PATH) in the irreducible Brillouin zone. VASPMATE generates the normalized primitive cell required for energy band calculation through the following command.

Mode: Standard primitive cell

Syntax:

VASPMATE --std3d file1(INPOS) file2(STD3POS)

VASPMATE --std2d file1(INPOS) file2(STD2POS)

Description:

In the process of calculating the energy band structure, it is often necessary to convert the current unit cell into a normalized primitive cell and generate a high-symmetry K-point path in the Brillouin zone based on it. This mode is mainly used to generate the normalized primitive cell form of the current unit cell, and on this basis, generate a high-symmetry K-point path for energy band structure drawing. In this command, 3d indicates that the calculation is for a three-dimensional (3D) structure, and 2d indicates that the calculation is for a two-dimensional (2D) structure. For two-dimensional structures, VASPMATE standardizes the 2D structure according to the following rules: (1) Add a vacuum layer normal to the z-axis direction, (2) Place the two-dimensional structure at the center position along the z-coordinate. The default input file, file1, is INPOS, and the output file, file2, is STD3POS/STD2POS. If the input file cannot be found, a prompt message will be printed: “No input file is found!”

Examples:

VASPMATE --std3d

VASPMATE --std3d INPOS

VASPMATE --std3d INPOS STD3POS

Selecting high-symmetry K-points and connection paths in the irreducible Brillouin zone for the normalized primitive cell is critical in calculating the energy band structure. The traditional approach involves obtaining high-symmetry points in the first Brillouin zone of the crystal reciprocal space through the SeeK-Path website or Material Studio software. Then, high-density K-points on the path between high-symmetry points are generated through interpolation to obtain the required KPOINTS. However, these operations are often relatively cumbersome and unsuitable for high-throughput computation. VASPMATE integrates an algorithm consistent with SeeK-Path[6] to analyze the high-symmetry points of crystals and provides a suitable set of K-PATH for energy band calculations. After testing, VASPMATE can generate high-symmetry point paths that are completely consistent with SeeK-Path for 24 variant structures of the seven crystal systems.

3.2.1 Basic Band Structure Using PBE Functional

The following command line format is used to generate a high-symmetry K-point path for basic energy band calculation by means of PBE functional.

Mode: K-PATH of standardized primitive cell

Syntax:

VASPMATE --ka3d file(INPOS) (-par) [value]

VASPMATE --ka2d file(INPOS) (-par) [value]

Description:

Using this functionality, VASPMATE will generate a NEWKPATH file based on the normalized primitive cell corresponding to INPOS. This file contains the high-symmetry point path, which can be directly copied and used as KPOINTS. Additionally, a HIGH_SYMMETRY_POINT file will be created, which stores information about high-symmetry points and the recommended K-path. The parameter [value] represents the default number of points distributed along each path. Furthermore, if INPOS is not a normalized primitive cell, This functionality will also generate a normalized primitive cell named STD3POS/STD2POS (Note: This primitive cell is equivalent to the normalized primitive cell generated through --std3d/--std2d as described earlier).

The content of the NEWKPATH file is as follows:

```
K-Path Generated by VASPMATE
    20
Line-Mode
Reciprocal
      0.000000     0.000000     0.000000    GAMMA
      0.500000     0.500000     0.500000    T

      0.500000     0.500000     0.500000    T
      0.734877     0.265123     0.500000    H_2

      0.500000    -0.265123     0.265123    H_0
      0.500000     0.000000     0.000000    L

      0.500000     0.000000     0.000000    L
      0.000000     0.000000     0.000000    GAMMA

      0.000000     0.000000     0.000000    GAMMA
      0.382562    -0.382562     0.000000    S_0

      0.617438     0.000000     0.382562    S_2
      0.500000     0.000000     0.500000    F

      0.500000     0.000000     0.500000    F
      0.000000     0.000000     0.000000    GAMMA
```

The content of the HIGH_SYMMETRY_POINT file is as follows:

```
High - symmetry points(in fractional coordinates).
      0.000000     0.000000     0.000000  GAMMA
      0.500000     0.500000     0.500000  T
      0.500000     0.000000     0.000000  L
      0.000000    -0.500000     0.000000  L_2
      0.000000     0.000000    -0.500000  L_4
      0.500000     0.000000     0.500000  F
      0.500000     0.500000     0.000000  F_2
      0.382562    -0.382562     0.000000  S_0
      0.617438     0.000000     0.382562  S_2
      0.382562     0.000000    -0.382562  S_4
      0.617438     0.382562     0.000000  S_6
      0.500000    -0.265123     0.265123  H_0
      0.734877     0.265123     0.500000  H_2
      0.734877     0.500000     0.265123  H_4
      0.500000     0.265123    -0.265123  H_6
      0.382562    -0.265123     0.382562  M_0
      0.617438     0.265123     0.617438  M_2
      0.734877     0.382562     0.382562  M_4
      0.617438     0.617438     0.265123  M_6
      0.382562     0.382562    -0.265123  M_8
suggest path : GAMMA-T-H_2|H_0-L-GAMMA-S_0|S_2-F-GAMMA
```

Examples:

VASPMATE --ka3d -par 20

Next, we introduce the post-processing analysis of VASPMATE for energy band calculation. All data have set the Fermi level to 0. It's worth noting that the Fermi level should be based on the value from the static self-consistent calculation. Therefore, if you want to obtain an accurate Fermi level, one may execute the following command after structural optimization or static self-consistency to extract the Fermi level from DOSCAR for subsequent data processing of energy band:

VASPMATE --dos -efermi >> FERMI_LEVEL

During the process to extract the basic energy band structure, if VASPMATE does not find the FERMI_LEVEL file, it will perform energy band analysis using the Fermi level from the DOSCAR file in the current folder and print the following prompt message: "Attention: Please use the Fermi level in self-consistent calculation!" However, the results obtained in this way are often inaccurate. Below are various methods for extracting the data of basic energy band structure.

Mode: **Basic Band Structure**

Syntax:

VASPMATE --band -b

Description:

This functionality requires the following files to be present in the current folder: CONTCAR provides the information of structure, OUTCAR provides the basic output information, PROCAR provides the information of band weight, EIGENVAL provides the information of energy eigenvalues and K-points, the KPOINTS file provides the information of K-PATH , and the DOSCAR file provides the information of Fermi level. The output files include: *BAND.dat* and *BAND_REFORMATTED.dat*, which store the band information; *KLABELS*, which saves the symbols and positions of high-symmetry points on the band diagram; and *Fermi_Energy*, which records the information of Fermi level. These files can be directly used for plotting the energy band structure in Origin or with the built-in analyzer in SPaMD.

The BAND.dat and *BAND_REFORMATTED.dat* files contain the band information, and their contents are as follows:

```
#K-Path(1/A) Energy-Level(eV)
# NKPTS & NBANDS:  140   48
# Band-Index    1
 0.000000   -17.572077
 0.040558   -17.570388
 0.081116   -17.565358
 0.121675   -17.557113
 0.162233   -17.545714
 0.202791   -17.531449
 0.243349   -17.514537
 0.283907   -17.495348
 0.324466   -17.474306
 0.365024   -17.451885
 0.405582   -17.428610
 0.446140   -17.405026
 0.486698   -17.381823
 0.527256   -17.359717
 0.567815   -17.339357
 0.608373   -17.321463
 0.648931   -17.306706
 0.689489   -17.295712
 0.730047   -17.288932
 0.770606   -17.286634
 0.770606   -17.286634
 0.800588   -17.285113
 0.830570   -17.280551
 0.860552   -17.272973
 0.890534   -17.262418
 0.920517   -17.248923
 0.950499   -17.232565
 0.980481   -17.213415
 1.010463   -17.191566
 1.040445   -17.167154
 1.070428   -17.140311
 1.100410   -17.111204
 1.130392   -17.079983
 1.160374   -17.046889
 1.190356   -17.012127
```

The *KLABELS* file stores the symbols and coordinate positions of high-symmetry points on the band diagram, and its content are shown below:

```
K-Label    Coordinate of high-symmetry k-point in band-structure plots
    GAMMA    0.000000
    T    0.770606
    H_2|H_0 1.340267
    L    1.958396
    GAMMA    2.690696
    S_0|S_2 3.593656
    F    3.878486
    GAMMA    4.735345


*Tips: Tip: Please label each high symmetry point in NEWKPATH file. Otherwise, they will be identified as 'XX' in KLABELS file
```

The *Fermi_Energy* file saves the information of Fermi level, and its content is pasted below for reference:

```
#The Fermi_Energy is read from DOSCAR.The data in TODS.dat has shifted efermi to 0 ev.
8.331273
```

Examples：

VASPMATE --band -b

Mode: **Projected Band-Structure for Each Atom**

Syntax:

VASPMATE --band -a

Description:

This functionality requires the following files to be present in the current folder: CONTCAR provides the information of structure, OUTCAR provides basic output information, PROCAR provides the information of band weight, EIGENVAL provides the information of energy eigenvalues and K-points, the KPOINTS file provides the information of K-PATH , and the DOSCAR file provides the information of Fermi level. The option will output the projected band structure of each atom to the files named *PBAND_A*.dat*, where * represents the atomic index. Each file includes the projection information of one atom, along with the weights for each angular momentum: s, py, pz, px, dxy, dyz, dz2, dxz, dx2+y2, and tot. The first column represents the length of the K-path in $Å^{-1}$, the second column is the band energy, the next columns are the projections of the lm orbitals onto this band, and the final column is the total projection of the atom onto this band.

```
#K-Path        Energy      s      py      pz      px      dxy      dyz      dz2      dxz      dx2      tot
# NKPTS & NBANDS:  140  48
# Band-Index   1
 0.000000   -17.572077  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 0.040558   -17.570388  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 0.081116   -17.565358  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 0.121675   -17.557113  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 0.162233   -17.545714  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 0.202791   -17.531449  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 0.243349   -17.514537  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 0.283907   -17.495348  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 0.324466   -17.474306  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 0.365024   -17.451885  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 0.405582   -17.428610  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 0.446140   -17.405026  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 0.486698   -17.381823  0.042000   0.005000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.527256   -17.359717  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.567815   -17.339357  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.608373   -17.321463  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.648931   -17.306706  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.689489   -17.295712  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.730047   -17.288932  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.770606   -17.286634  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.770606   -17.286634  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.800588   -17.285113  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.830570   -17.280551  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.860552   -17.272973  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.890534   -17.262418  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.920517   -17.248923  0.042000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.950499   -17.232565  0.041000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.048000
 0.980481   -17.213415  0.041000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 1.010463   -17.191566  0.040000   0.006000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 1.040445   -17.167154  0.040000   0.007000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.047000
 1.070428   -17.140311  0.039000   0.007000   0.000000   0.001000   0.000000   0.000000   0.000000   0.000000   0.000000   0.046000
 1.100410   -17.111204  0.038000   0.007000   0.000000   0.001000   0.000000   0.000000   0.000000   0.000000   0.000000   0.046000
 1.130392   -17.079983  0.038000   0.007000   0.000000   0.001000   0.000000   0.000000   0.000000   0.000000   0.000000   0.046000
 1.160374   -17.046889  0.037000   0.008000   0.000000   0.001000   0.000000   0.000000   0.000000   0.000000   0.000000   0.045000
```

Examples：

VASPMATE --band -a

***Mode:* Projected Band-Structure for Each Element**

Syntax:

VASPMATE --band -e

Description:

This functionality requires the following files to be present in the current folder: CONTCAR provides the information of structure, OUTCAR provides basic output information, PROCAR provides the information of band weight, EIGENVAL provides the information of energy eigenvalues and K-points, the KPOINTS file provides the information of K-PATH , and the DOSCAR file provides the information of Fermi level. The option exports the projected band structure of each element to files named *PBAND_*.dat*, where * stands for the element symbol. Each file contains the projection details of one element, along with the weights for each angular momentum: s, py, pz, px, dxy, dyz, dz2, dxz, dx2+y2, and tot. The first column represents the length of the K-path in $Å^{-1}$, the second column denotes the band energy, the succeeding columns represent the projections of the lm orbitals onto this band, and the final column is the total projection of one element onto this band.

Examples：

VASPMATE --band -e

Mode: **Projected Band-Structure for Selected Atoms & Elements**

Syntax:

VASPMATE --band -s [atom-index] [element-index]

Description:

This functionality requires the following files to be present in the current folder: CONTCAR provides the information of structure, OUTCAR provides basic output information, PROCAR provides the information of band weight, EIGENVAL provides the information of energy eigenvalues and K-points, the KPOINTS file provides the information of K-PATH, and the DOSCAR file provides the information of Fermi level. This option exports the projected band structure of selected atoms and elements to files named *PBAND_A*.dat* and *PBAND_B*.dat*, where * represents the selected atom index. All generated files, such as *PBAND_A1.dat*, *PBAND_B2.dat*, and so on, can be used for plotting in Origin. Additionally, the output file named KLABELS is used to mark the positions of high-symmetry points on the graph.

Examples：

VASPMATE --band -s 1-4 B N

Mode: **Projected Band-Structure for Selected Atoms**

Syntax:

VASPMATE --band -sa [atom-index]

Description:

This functionality requires the following files to be present in the current folder: CONTCAR provides the information of structure, OUTCAR provides basic output information, PROCAR provides the information of band weight, EIGENVAL provides the information of energy eigenvalues and K-points, the KPOINTS file provides the information of K-PATH, and the DOSCAR file provides the information of Fermi level. This option exports the projected band structure of selected atoms to files named *PBAND_A*.dat*, where * denotes the chosen atom index. Each file contains the projection details for the selected atom, along with weights for each angular momentum: s, py, pz, px, dxy, dyz, dz2, dxz, dx2+y2, and tot. The first column represents the length of the K-path in Å^{-1}, the second column denotes the band energy, the following columns embody the projections of the lm orbitals onto this band, and the final column is the total projection of the selected atom onto this band.

Examples：

VASPMATE --band -sa 1

VASPMATE --band -sa 2-5

***Mode:* Projected Band-Structure for Selected Elements**

Syntax:

VASPMATE --band -se [element-index]

Description:

This functionality requires the following files to be present in the current folder: CONTCAR provides the information of structure, OUTCAR provides basic output information, PROCAR provides the information of band weight, EIGENVAL provides the information of energy eigenvalues and K-points, the KPOINTS file provides the information of K-PATH , and the DOSCAR file provides the information of Fermi level. This option exports the projected band structure for selected elements to files named *PBAND_B*.dat*, where B represents the chosen element name and * denotes the corresponding atom index within that element. All generated files, such as *PBAND_B1.dat*, *PBAND_B2.dat*, and so on, can be used for plotting in Origin. Additionally, KLABELS is used to mark the positions of high-symmetry points on the graph.

Examples：

VASPMATE --band -se B

VASPMATE --band -se B N

Mode: Projected Band-Structure for Multiple Atoms or Elements

Syntax:

VASPMATE --band -m/-ma/-me [atom-index]/[element-index]

Description:

This functionality requires the following files to be present in the current folder: CONTCAR provides the information of structure, OUTCAR provides basic output information, PROCAR provides the information of band weight, EIGENVAL provides the information of energy eigenvalues and K-points, the KPOINTS file provides the information of K-PATH , and the DOSCAR file provides the information of Fermi level. This option exports the projected band structures for selected atoms and elements to a file named *PBAND_SUM_A1-4_B-N.dat*, where the selected atoms and elements can be specified in a free format. This is helpful in studying the layered bands and comparing the surface bands with the internal ones.

Examples：

VASPMATE --band -m 1-4 B N

VASPMATE --band -ma 1-4

VASPMATE --band -me B N

***Mode:* Sum of Projected Band for Selected Orbitals of Selected Atoms/Elements**

Syntax:

VASPMATE --band -o/-oa/-oe [atom-index]/[element-index] [orbit-index]

Description:

This functionality requires the following files to be present in the current folder: CONTCAR provides the information of structure, OUTCAR provides basic output information, PROCAR provides the information of band weight, EIGENVAL provides the information of energy eigenvalues and K-points, the KPOINTS file provides the information of K-PATH , and the DOSCAR file provides the information of Fermi level. The function exports projected band structure data for selected atoms/elements and their corresponding orbitals to the file *PBAND_SUM.dat*. Orbit types include s, py, pz, px, dxy, dyz, dz2, dxz, dx2+y2 (f1-f7). The "all" option can be used to select all orbitals. Please note that only one combination of atom + orbit can be performed accurately; consecutive additions may lead to inaccurate results.

Examples：

VASPMATE --band -oa 1-3 B N

VASPMATE --band -oa 1-3 s px py

VASPMATE --band -oe B all

Mode: **Band index & K Point**

Syntax:

VASPMATE --band -id/-index

Description:

This functionality requires the following files to be present in the current folder: CONTCAR provides the information of structure, OUTCAR provides basic output information, PROCAR provides the information of band weight, EIGENVAL provides the information of energy eigenvalues and K-points, the KPOINTS file provides the information of K-PATH , and the DOSCAR file provides the information of Fermi level. This functionality is used to output the band indices, the energy level, the high-symmetry K-point numbers, and the high-symmetry K-path near the Fermi level (five bands above and below) after the current VASP calculation. Detailed and complete information can be found in the output file *BANDKPTS_INFO*, which facilitates the real-time viewing and the selection of corresponding band indices (Fermi level) and K-point numbers (symbols). Note: The information can be read either solely from the

OUTCAR file or from the IBZKPT and EIGENVAL files.

```
# 5 energy bands near the Fermi level.

# Band-Index    4
 12.875643      -3.545947
 12.837888      -3.558394
 12.800134      -3.595239
 12.762380      -3.655020
```

Examples:

VASPMATE --band -id

Mode: **Band Gap**

Syntax:

VASPMATE --band -bg

Description:

This functionality requires the following files to be present in the current folder: CONTCAR provides the information of structure, OUTCAR provides basic output information, PROCAR provides the information of band weight, EIGENVAL provides the information of energy eigenvalues and K-points, the KPOINTS file provides the information of K-PATH , and the DOSCAR file provides the information of Fermi level. This option is used to extract more information from the band structure calculations, including Band Gap, VBM (Valence Band Maximum), CBM (Conduction Band Minimum), Highest and Lowest Occupied Bands, Location of VBM & CBM, and whether it is a direct bandgap material. Note: The above information is only relevant for insulator or semiconductor crystals with a bandgap greater than 0. For metallic crystals, VBM and CBM are not meaningful, and only "Band Character: Metallic" information will be provided.

```
------------------------ Band Information ------------------------
             Band Gap (eV):   0.556672
            Band Character:   Indirect
         Fermi Energy (eV):   6.052482
        VBM Eigenvalue(eV):   5.759581
        CBM Eigenvalue(eV):   6.316252
    Highest-Occupied Band:           4
     Lowest-Occupied Band:           5
          Location of VBM:   0.000000   0.000000   0.000000
          Location of CBM:   0.421053   0.000000   0.421053
-------------------------------------------------------------------
```

Examples:

VASPMATE --band -bg

Mode: **Effective Mass**

Currently, VASPMATE only supports effective mass calculations for non-charged and non-magnetic semiconductors or insulators. For materials that have already undergone band structure calculations, there is no need to re-sample points on the corresponding high-symmetry path, and the band data can be directly fitted. This method not only ensures accuracy but is also suitable for degenerate cases at the top of the valence band and the bottom of the conduction band. This option requires the following files to be present in the current folder: CONTCAR provides the information of structure, OUTCAR provides the basic output information, PROCAR provides the information of band weight, EIGENVAL provides the information of energy eigenvalues and K-points, the KPOINTS file provides the information of K-PATH , and the DOSCAR file provides the information of Fermi level.

Syntax:

VASPMATE --band -em [point] [direct]…(-nb [band-index]) (-np [fitting-point])

Description:

This functionality is used to calculate the effective mass of semiconductors or insulators. Here, [point] represents the symbol of a high-symmetry point, such as G, K, M, etc. The parameter [direct] indicates the direction at that high-symmetry point, divided into l (left) and r (right). The parameters [point] and [direct] can be used consecutively. The parameter [band-index] corresponds to the band index of the valence band maximum. It can be specified using -nb, but by default, VASPMATE will automatically search for it. Note that if you want to use a high-symmetry point that was not used in the original calculation, you can simply append the coordinates and the name of the high-symmetry point at the end of the KPOINTS file. The parameter [fitting-point] represents the number of fitting points, which defaults to 6 and can be specified using the keyword -np (Tip: To improve the accuracy of quadratic curve fitting, it is recommended to select fewer points around the extremum, with a suggestion of 3-6 points).

Examples：

VASPMATE --band -em K l K r

VASPMATE --band -em G l M r -nb 5 -np 4

Besides the aforementioned method in calculating effective mass from the existing band data, there is a more straightforward approach. This method consists of two parts: preprocessing and post-processing. VASPMATE performs the calculation which involves the re-sampling points at equal intervals along the specified high-symmetry path. Then, it fits the calculated energy eigenvalues at each point with the k-path using a quadratic curve and determines the effective mass of electrons/holes at that high-symmetry point. In this process, VASPMATE will fit all data points starting from the initial point and selecting the minimum R-squared value as the fitting criterion. This ensures a minimal information loss to a greatest extent.

Syntax:

VASPMATE --em

Description:

This method to calculate effective mass relies on the EMC.in control file, which contains the following information: (1) Processing mode, (2) KMESH sampling mode, (3) Brillouin zone sampling accuracy (similar to the KSPAC parameter), (4) Number of sampling points on the high-symmetry path, (5) Sampling interval on the high-symmetry path, (6) Path direction between high-symmetry points. Below is the content of the preprocessing EMC.in file used when calculating the effective mass of MoS2.

```
1   # "1" for pre-process (generate KPOINTS), "2" for post-process(calculate m*)
G
0.21
6       # number of points for quadratic function fitting.
0.015   # k-cutoff, unit Å-1.
2       # number of tasks for effective mass calculation
0.333333 0.3333333 0.000 0.000 0.000 0.000 K->Γ # Specified two K-points and direction
0.333333 0.3333333 0.000 0.500 0.000 0.000 K->M # Specified two K-points and direction
```

When using VASPMATE --em as described above, VASPMATE will generate an EMCKPT file based on the information in the EMC.in file, which is then copied as the KPOINTS file for VASP calculation. After the calculation is complete, change the calculation mode on the first line to 2, leaving the remaining unchanged. Using VASPMATE --em to perform analysis will output an Effective_Mass file containing the effective mass. As shown below, the calculated effective mass for MoS2 is 0.45, and those of holes is 0.54, which are very close to the values of 0.44 and 0.54 reported by Kormányos[14] and others, respectively.

```
+ -------------------Summary------------------- +
Fitting Point: 6
Band Index:            LUMO = 10    HOMO = 9
Effective-Mass (in m0)  Electron (R2)   Hole (R2)
K-Path Index 1: K->Γ    0.454757 (0.999998) -0.546734 (1.000000)
K-Path Index 2: K->M    0.488905 (0.999967) -0.625759 (0.999928)
+ -------------------Summary------------------- +
```

Examples:

VASPMATE --em   (need control file: EMC.in)

【Procedure】The basic band structure calculation using VASPMATE is as follows:

(1) **【Structural Optimization】**After constructing the structure in POSCAR, perform normal structural optimization. Below lists a series of commands in preparing the structural optimization. Afterwards, copy the optimized CONTCAR to INPOS for next step.

cp POSCAR INPOS

VASPMATE --i rlx   # Generate incar_rlx for structural relaxation

cp incar_rlx INCAR

VASPMATE --pot -PBE   # Generate NEWPOT according to INPOS

cp NEWPOT POTCAR

VASPMATE --ka -par 4000   # Generate NEWKPT according to INPOS

cp NEWKPT KPOINTS

(2) **【Static Preparation】** Based on the optimized structure, use VASPMATE --std3d to generate the standardized primitive cell file STD3POS from INPOS, use VASPMATE --ka -par 4000 to generate the K-point sampling file NEWKPT from INPOS, and use VASPMATE --inp stc to generate the incar_stc file. Copy them to POSCAR, KPOINTS, and INCAR respectively.

(3) **【Static Self-Consistent Calculation】**Perform a single-point static self-consistent calculation to obtain CHGCAR and WAVECAR, which will be used for subsequent non-self-consistent band structure calculations.

(4) 【Band Preparation】Use VASPMATE --ka3d -par 20 (where 20 is the number of sampling points) to generate the K-point path file NEWKPATH for band structure calculation based on INPOS, and copy it to KPOINTS. Use VASPMATE --inp nsc to generate the incar_nsc file for non-self-consistent calculation and copy it to INCAR. Note: Non-self-consistent calculations require ISTART=1 and ICHARG=11.

(5) **【Band Calculation】**Perform a non-self-consistent calculation by reading the CHGCAR and WAVECAR generated from the previous static self-consistent calculation.

(6) **【Band Extraction】**Use VASPMATE --band -b to extract band structure. It will generate the files of Band.dat and Band_Reformated.dat to save band structure, the KLABELS file to save the symbols and coordinate positions of high-symmetry points on the band structure, and the Fermi_Energy file to save Fermi level.

(7) **【Band Analysis】**Use Origin or SPaMD to draw the band structure and analyze the high-symmetry K points of interest.

(8) **【Band Analysis】**Use VASPMATE --band -em G l G r to calculate the effective mass on both sides of the Gamma point.

(9) **【Band Analysis】**Use VASPMATE --band -bg to extract other information from band structure such as Band Gap, VBM, CBM, Highest-Occupied band & Lowest-Occupied Band, Location of VBM & CBM, and whether it is a direct band gap.

3.2.2 Exact Band Structure Using HSE Functional

After generating the KPOINTS file for HSE band structure calculations, the exact band structure can be obtained directly from a static self-consistent calculation (without the need for a non-self-consistent calculation). The following command line format is used to generate a high-symmetry K-point path for exact band structure calculation by means of HSE functional.

Syntax:

VASPMATE --kahse (INPOS) (-par) [kppra] [resolution] [kscheme]

VASPMATE --kvhse (INPOS) (-par) [kspac] [resolution] [kscheme]

VASPMATE --kmhse (-par) k_1 k_2 k_3 [resolution] [kscheme]

Description:

Compared to the generation of k-points for PBE band structure calculation, the HSE calculation introduces a parameter called "resolution" for the density of points along the high-symmetry path. The default value is 0.05. The generated file NEWKPT saves the k-point information for the HSE functional calculation which can be directly used for VASP calculations. Note: The k-points for exact band structure calculations by means of HSE functional include the SCF process, so the obtained Fermi level is accurate, and there is no need to prepare an additional FERMI_LEVEL file as in PBE functional calculations.

```
9   9   9   85  89  7   15  10  12  14  17  5   16          # Parameters to Generate KPOINTS (Do NOT Edit This Line)
    174
Reciprocal latticen
    0.000000    0.000000    0.000000    1
    0.111111    0.000000    0.000000    6
    0.222222    0.000000    0.000000    6
    0.333333    0.000000    0.000000    6
    0.444444    0.000000    0.000000    6
    0.111111    0.111111    0.000000    6
    0.222222    0.111111    0.000000    12
    0.333333    0.111111    0.000000    12
    0.444444    0.111111    0.000000    12
    -0.444444   0.111111    0.000000    12
    -0.333333   0.111111    0.000000    12
    -0.222222   0.111111    0.000000    12
    -0.111111   0.111111    0.000000    6
    0.222222    0.222222    0.000000    6
    0.333333    0.222222    0.000000    12
    0.444444    0.222222    0.000000    12
    -0.444444   0.222222    0.000000    12
    -0.333333   0.222222    0.000000    12
    -0.222222   0.222222    0.000000    6
    0.333333    0.333333    0.000000    6
    0.444444    0.333333    0.000000    12
    -0.444444   0.333333    0.000000    12
    -0.333333   0.333333    0.000000    6
    0.444444    0.444444    0.000000    6
    -0.444444   0.444444    0.000000    6
    0.111111    0.111111    0.111111    2
    0.222222    0.111111    0.111111    6
    0.333333    0.111111    0.111111    6
    0.444444    0.111111    0.111111    6
    -0.444444   0.111111    0.111111    6
    -0.333333   0.111111    0.111111    6
    -0.222222   0.111111    0.111111    6
    -0.111111   0.111111    0.111111    6
    0.222222    0.222222    0.111111    6
    0.333333    0.222222    0.111111    12
```

Examples:

VASPMATE --kahse INPOS 8000 0.05 G

VASPMATE --kahse INPOS -par 8000 0.05 G

VASPMATE --kvhse INPOS -par 0.5 0.05 G

VASPMATE --kmhse 7 7 1 0.05 G

VASPMATE --kmhse -par 7 7 1 0.05 G

For the post-processing of exact band structure calculations by means of HSE functional, the command format is similar to that of basic band structure calculations, with only slight differences in labeling. Specific input formats can be referred to the details of basic band calculations by means of PBE functional.

Syntax:

VASPMATE --band -hb

VASPMATE --band -ha

VASPMATE --band -he

VASPMATE --band -hs [atom-index] [element-index]

VASPMATE --band -hsa [atom-index]

VASPMATE --band -hse [element-index]

VASPMATE --band -hm/-hma/-hme [atom-index]/[element-index]

VASPMATE --band -ho/-hoa/-hoe [atom-index]/[element-index]

VASPMATE --band -hbg

Description:

Refer to the corresponding descriptions of options and parameters for post-processing of band structure by means of PBE functional in preceding section.

Examples：

VASPMATE --band -hb

VASPMATE --band -ha

VASPMATE --band -he

VASPMATE --band -hs 1-4 B N

VASPMATE --band -hsa 1-4

VASPMATE --band -hse B N C

VASPMATE --band -hm/-hma/-hme 1-4 B N

VASPMATE --band -ho/-hoa/-hoe 1-3 s px py

VASPMATE --band -hbg

【Procedure】The exact band structure calculation using VASPMATE is as follows:

(1) **【Structure Optimization】** Similar to the basic band calculation, use the PBE functional to optimize the crystal structure. Note: The structure optimization uses the PBE functional, while the HSE functional is only used for electronic structure calculations. Afterwards, copy CONTCAR to INPOS for next step.

(2) **【Static Preparation】** Based on the optimized structure, use VASPMATE --std3d to generate a standardized primitive cell file STD3POS from INPOS, use VASPMATE --ka -par 4000 to generate a K-point sampling file NEWKPT from INPOS, and use VASPMATE --inp stc to generate an incar_stc file. Copy these files to POSCAR, KPOINTS, and INCAR, respectively.

(3) **【Static Calculation】** Perform a single-point static self-consistent calculation to obtain CHGCAR and WAVECAR, which will be used for subsequent exact energy band calculations by means of HSE functional.

(4) **【Band Preparation】** Use VASPMATE --ka3d -par 20 (where 20 is the number of sampling points) to generate the K-point path file NEWKPATH and the high-symmetry point file HIGH_SYMMETRY_POINTS for band structure calculations based on INPOS.

(5) **【Band Preparation】** Use VASPMATE --kahse INPOS 8000 0.05 G to generate the NEWKPT file required for exact band calculations by merging the K-point path file NEWKPATH.

(6) **【Band Preparation】** (Optional Step) Perform a self-consistent calculation by means of PBE functional using the newly generated NEWKPT, and then read the wave function for the next self-consistent calculation by means of HSE functional. This is usually done to reduce computational time.

(7) **【Band Preparation】** Alternatively, skip the previous step and directly perform the exact band calculation by means of HSE functional. Note: use VASPMATE --i pbs hse to generate the incar_pbs_hse file for the exact band calculation (projected band structure requires setting LORBIT=10 or 11). The file should contain the following information:

LORBIT=11 or 10

LHFCALC=.TRUE.

AEXX=0.25

HFSCREEN=0.2

TIME=0.4

(8) 【**Band Calculation**】 Copy NEWKPT to KPOINTS and incar_stc_hse to INCAR. Perform the exact band calculation to generate band-related files by means of HSE functional: CONTCAR provides structural information, DOSCAR provides the information of Fermi level, OUTCAR provides basic output information, PROCAR provides the information of band weight, EIGENVAL provides the information of energy level, and KPOINTS provides the information of K-PATH .

(9) 【**Band Extraction**】 Use VASPMATE --band -he to extract the exact band structure calculated by HSE functional. This will generate the projected band structure for each element in the file PBAND_*.dat, where * represents the element symbol. Each file includes the projection for one element, as well as the weight on each angular momentum: s, py, pz, px, dxy, dyz, dz2, dxz, dx2+y2, and tot. The first column is the length of the K-path in Å^{-1}, the second column is the band energy, the following columns are the projections of the lm orbitals on this band, and the last column is the total projection of the element on this band. The KLABELS file saves the symbols and coordinate positions of high-symmetry points on the band structure.

(10) 【**Band Analysis**】 Use Origin or SPaMD to draw the projected band structure.

3.3 Density of States Calculation

In the case of quasi-continuous distribution of electronic energy levels, the ratio between the number of quantum states ΔZ in the energy range from E to E + ΔE and the energy difference ΔE is called the density of states (DOS). The density of states is closely related to the band structure and provides another visualization of the band structure. Many density of states analyses correspond one-to-one with band structure analyses, and many terms are also similar to those used in band structure analysis. Because the density of states is more intuitive, it is more widely used than the band structure analysis. Many properties of solids, such as electronic specific heat, absorption and emission of light and X-rays, are related to the density of states.

It is relatively simple to process density of states calculations through VASPMATE. In the INCAR file, LORBIT is set to 10 or 11, and the other DOS-related parameters need be provided for a better visualization, such as the number of energy points NEDOS and the energy range EMIN/EMAX. The command VASPMATE --inp stc dos may be used to generate the corresponding INCAR file: incar_stc_dos. During the VASP calculation of the density of states, by projecting the density of states onto specific orbitals or atoms, the partial density of states (PDOS) and local density of states (LDOS) can be obtained for the orbital analysis and compositional analysis, respectively. When LORBIT=10 is set in the INCAR file, the projected orbitals corresponds to s, p, and d (f). When LORBIT=11, the projected orbitals are further subdivided into s, px, py, pz, dxy, dyz, dxz, dx2+y2, dz2 (f1-f7).

VASPMATE provides powerful post-processing analysis capabilities to extract the density of states. When using This functionality, all data will be set to zero by subtracting the Fermi level.

Mode: **Density of States**

Syntax:

Syntax	*Description*
VASPMATE --dos -t [none]	Get the total DOS
VASPMATE --dos -a [none]	Output the projected DOS for each atom to separate file.
VASPMATE --dos -e [none]	Output the projected DOS for each element to separate file.
VASPMATE --dos -s [atom-index]	Output the projected DOS for selected atoms/element to separate file.
VASPMATE --dos -sa [atom-index]	Output the projected DOS for selected atoms to separate file.
VASPMATE --dos -se [element-index]	Output the projected DOS for selected elements to separate file.
VASPMATE --dos -m/-me/-ma [select-list]	Output the sum of projected DOS for selected multiple atoms/elements
VASPMATE --dos -o/-oe/-oa [atom&orbit-list]	Output the sum of projected DOS for selected atoms and orbitals
VASPMATE --dos -bc	Output the Band Center(s,p,d,f) or (s,px,py,pz,dxy,dyz,dxz,fn...)

Description:

This functionality requires the existence of the following files in the current folder:

CONTCAR provides the structural information, OUTCAR provides the basic output information, and DOSCAR provides the information of electronic density of states and Fermi level.

The relevant files output by This functionality include: TDOS.dat (*PDOS_A*.dat*, *PDOS_.dat, PDOS_SUM_A*.dat, PDOS_SUM_*.dat). The *PDOS_SUM_A1-5_N.dat* file stores energy and selected atomic orbital projection information. The first line of PDOS_USER.dat corresponds to the command VASPMATE --dos -o 1-2 s N px py. The SELECT_ATOMS_LIST file stores the information of atom selection. PDOS_USER.dat corresponds to VASPMATE --dos -oa 1-4 s px 5-9 all. The Band_Center file stores the s, p, and d band center.

Examples：(Files starting with 'I' are the results of DOS integration, with the structure B_6N and a total of 14 atoms.)

Input command	*Output files*
VASPMATE --dos -t	TDOS.dat & ITDOS.dat
VASPMATE --dos -a	PDOS_A1.dat & IPDOS_A1.dat ~ PDOS_A9.dat & IPDOS_A9.dat
VASPMATE --dos -e	PDOS_B.dat & IPDOS_B.dat & PDOS_N.dat & IPDOS_N.dat
VASPMATE --dos -s 1-5	PDOS_A1.dat & IPDOS_A1.dat ~ PDOS_A5.dat & IPDOS_A5.dat
VASPMATE --dos -m 1-5 N	PDOS_SUM_A1-5_N.dat & IPDOS_SUM_A1-5_N.dat SELECT_ATOMS_LIST
VASPMATE --dos -oa 1-4 s px	PDOS_USER.dat
VASPMATE --dos -bc all	Band_Center

TDOS.dat(PDOS_A.dat,PDOS_*.dat,PDOS_SUM_A*_*.dat,PDOS_SUM_*_*.dat files all have the same format and can be used for plotting graphs in Origin.)*

#Energy	TDOS
-18.876479	0.000000
-18.858479	0.000000
-18.840479	0.000000
-18.823479	0.000000
-18.805479	0.000000
-18.787479	0.000000
-18.769479	0.000000
-18.751479	0.000000
-18.733479	0.000000
-18.715479	0.000000
-18.698479	0.000000
-18.680479	0.000000
-18.662479	0.000000
-18.644479	0.000000
-18.626479	0.000000
-18.608479	0.000000
-18.590479	0.000000
-18.573479	0.000000
-18.555479	0.000000

PDOS_SUM_A1-5_N.dat (saves the energy and the selected atomic projected orbital information; the selected atomic information can be viewed in the SELECT_ATOMS_LIST)

#Energy	s	py	pz	px	dxy	dyz	dz2	dxz	dx2	tot	
-18.876479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.858479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.840479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.823479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.805479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.787479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.769479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.751479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.733479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.715479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.698479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.680479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.662479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.644479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.626479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.608479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.590479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.573479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.555479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.537479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.519479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.501479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.483479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.465479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.448479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.430479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.412479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.394479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.376479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.358479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.340479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.323479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.305479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.287479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.269479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.251479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-18.233479	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

PDOS_USER.dat (the first line corresponds to the command VASPMATE --dos -o 1-2 s N px py)

```
#Energy 1-2_s   N_px&py
-18.876479  0.000000    0.000000
-18.858479  0.000000    0.000000
-18.840479  0.000000    0.000000
-18.823479  0.000000    0.000000
-18.805479  0.000000    0.000000
-18.787479  0.000000    0.000000
-18.769479  0.000000    0.000000
-18.751479  0.000000    0.000000
-18.733479  0.000000    0.000000
-18.715479  0.000000    0.000000
-18.698479  0.000000    0.000000
-18.680479  0.000000    0.000000
-18.662479  0.000000    0.000000
-18.644479  0.000000    0.000000
-18.626479  0.000000    0.000000
-18.608479  0.000000    0.000000
-18.590479  0.000000    0.000000
-18.573479  0.000000    0.000000
-18.555479  0.000000    0.000000
-18.537479  0.000000    0.000000
-18.519479  0.000000    0.000000
-18.501479  0.000000    0.000000
-18.483479  0.000000    0.000000
-18.465479  0.000000    0.000000
-18.448479  0.000000    0.000000
-18.430479  0.000000    0.000000
-18.412479  0.000000    0.000000
-18.394479  0.000000    0.000000
-18.376479  0.000000    0.000000
```

SELECT_ATOMS_LIST (information on the selection of atoms; in the graphic, all atoms are selected)

```
ATOMS_ID ATOM_LABEL  X_POSITION    Y_POSITION    Z_POSITION    SELECTED?
    1    B1   0.796745     0.328767     0.796745     T
    2    B2   0.203255     0.671233     0.203255     T
    3    B3   0.796745     0.796745     0.328767     T
    4    B4   0.203255     0.203255     0.671233     T
    5    B5   0.328767     0.796745     0.796745     T
    6    B6   0.671233     0.203255     0.203255     T
    7    B7   0.999036     0.664694     0.999036     T
    8    B8   0.000964     0.335306     0.000964     T
    9    B9   0.999036     0.999036     0.664694     T
    10   B10  0.000964     0.000964     0.335306     T
    11   B11  0.664694     0.999036     0.999036     T
    12   B12  0.335306     0.000964     0.000964     T
    13   N1   0.626852     0.626852     0.626852     T
    14   N2   0.373148     0.373148     0.373148     T
```

PDOS_USER.dat(VASPMATE --dos -oa 1-4 s px 5-9 all)

```
#Energy 1-4_s&px    5-9_all
-18.887084  0.000000    0.000000
-18.869084  0.000000    0.000000
-18.851084  0.000000    0.000000
-18.834084  0.000000    0.000000
-18.816084  0.000000    0.000000
-18.798084  0.000000    0.000000
-18.780084  0.000000    0.000000
-18.762084  0.000000    0.000000
-18.744084  0.000000    0.000000
-18.726084  0.000000    0.000000
-18.709084  0.000000    0.000000
-18.691084  0.000000    0.000000
-18.673084  0.000000    0.000000
-18.655084  0.000000    0.000000
-18.637084  0.000000    0.000000
-18.619084  0.000000    0.000000
-18.602084  0.000000    0.000000
-18.584084  0.000000    0.000000
-18.566084  0.000000    0.000000
-18.548084  0.000000    0.000000
-18.530084  0.000000    0.000000
-18.512084  0.000000    0.000000
-18.494084  0.000000    0.000000
```

Band_Center (stores the s, p, d band center data)

```
#   Atom_ID   s-band-center    p-band-center    d-band-center   (in units of eV)
SPIN-UP all 49.639715   51.622606   25.861222
SPIN-DW all 41.392813   53.453747   27.228907

#   Atom_ID   s    py    pz    px    dxy    dyz    dz2    dxz    dx2    (in units of eV)
SPIN-UP all 49.639715   51.695426   51.599076   51.572309   25.025426   24.530076   26.772782   26.027049   26.819471
SPIN-DW all 49.639715   53.600855   53.439957   53.316472   26.138259   25.684847   28.493581   27.208352   28.452890


Remark:
Energy window for integration is from -12.597178 to 76.665822.
Band Center is respect to Fermi level, i.e., E_F = 0 eV.
```

【Procedure】The calculation of projected density of states using VASPMATE is as follows:

(1) **【Structure Optimization】** Optimize the structure of POSCAR using the PBE functional. After obtaining CONTCAR, copy it to INPOS.

(2) **【Static Preparation】** Execute VASPMATE --kpa 8000 to generate a standard KPOINTS file named NEWKPT based on the INPOS file. Copy NEWKPT to KPOINTS.

(3) **【Static Preparation】** Use VASPMATE --inp stc dos to quickly generate the corresponding INCAR file named incar_stc_dos. Copy incar_stc_dos to INCAR. Note: In INCAR, ensure that LORBIT is set to 10 or 11, and set the DOS-related parameters such as the number of energy points (NEDOS) and the energy range (EMIN/EMAX).

(4) **【Static Self-Consistent Calculation】** Perform a single-point static self-consistent calculation to determine the density of states.

(5) **【Density of States Extraction】** Use VASPMATE --dos -e to extract the projected density of states for each element, such as PDOS_B.dat, IPDOS_B.dat, PDOS_N.dat, and IPDOS_N.dat. Note: In INCAR, when LORBIT=10, the projected orbitals are s, p, d(f); when LORBIT=11, the projected orbitals are further subdivided into s, px, py, pz, dxy, dyz, dxz, dx2+y2, dz2(f1-f7).

(6) **【Density of States Analysis】** Use VASPMATE --dos -bc all to calculate the center positions of each orbital, such as (s, p, d, f) or (s, px, py, pz, dxy, dyz, dxz, fn...).

3.4 Bader Charge Calculation

Bader proposed the use of zero-flux surfaces of charge density to divide atoms. A zero-flux surface is an ideal position to divide atoms using a two-dimensional plane where the charge density perpendicular to it is at its minimum. VASP provides the functionality to write out the core charge from PAW calculations. This functionality is achieved by adding LAECHG=.TRUE. to the INCAR file, where the core charge will be written to AECCAR0 and the valence charge to AECCAR2. These two charge density files need to be summed for Bader analysis. This analysis requires a very fine FFT grid, so the total charge is kept constant by continuously increasing the value of NG(X,Y,Z). VASPMATE provides a function to sum the two charge density files, as follows:

Mode: **Combine Charge Density**

Syntax:

VASPMATE --bader -comb (AECCAR0) (AECCAR2) (factor1) (factor2)

Description:

This functionality is used to add the charge densities from two CHGCAR files, with factor1 and factor2 as proportionality coefficients. If not specified, they both default to 1. The output charge density is AECCAR0*factor1 + AECCAR2*factor2, and the output file is named as CHGCAR_SUM. If the two charge density files being merged are not consistent, VASPMATE will output an error message: "Attention: Two CHGCAR files are not consistent!". This functionality requires that two files necessary for Bader charge analysis have already been generated using VASP calculations: AECCAR0 provides the core electron density, while AECCAR2 provides the valence electron density.

Examples:

VASPMATE --bader -comb

VASPMATE --bader -comb AECCAR0 AECCAR2

VASPMATE --bader -comb AECCAR0 AECCAR2 1.1 0.9

Mode: **Calculate Atomic Charge**

Syntax:

VASPMATE --bader -d/-derive/-calc

Description:

This functionality extracts the coordinates and charges of atoms from the Bader analysis result file ACF.dat, and calculates the net charge of atoms by comparing them with the ZVAL values in the POTCAR file. This functionality requires the presence of both ACF.dat (generated by the Bader program) and the POTCAR file in the current directory. The default output file is named *Bader_Charge*.

```
#Calculate the net charge, generated by VASPMATE.

#No.  Element      X             Y             Z         Bader_Charge      ZVAL       Net_Charge
#---------------------------------------------------------------------------------------------------
  1      N      3.584158      0.000000      0.000000      0.000000      5.000000      5.000000
  2      N      3.584158      1.773982      1.773982      0.000000      5.000000      5.000000
  3      C      1.838467      1.782347     -0.008365      3.603271      4.000000      0.396729
  4      C      1.838467     -0.008365      1.782347      3.603271      4.000000      0.396729
  5      C      2.682618      0.878626      2.669337      5.717231      4.000000     -1.717231
  6      C      2.682618      2.669337      0.878626      5.717231      4.000000     -1.717231
  7      B      0.903878      0.886991      0.886991      6.679498      3.000000     -3.679498
  8      B      0.903878      2.660972      2.660972      6.679498      3.000000     -3.679498
#---------------------------------------------------------------------------------------------------
```

Examples:

VASPMATE --bader -calc

【Procedure】The Bader charge calculations using VASPMATE is as follows:

(1) **【Structure Optimization】** Perform normal structure optimization on the initial structure. Copy the optimized CONTCAR to INPOS.

(2) **【Structure Optimization】** Use VASPMATE --ka 4000 to generate a NEWKPT file based on INPOS, and copy it to KPOINTS. Use VASPMATE --inp stc bca to generate incar_stc_bca, and copy it to INCAR. Alternatively, the static calculation INCAR may be modified directly as follows:

LCHARG = .TRUE. # Output the required charge density file CHGCAR

LAECHG = .TRUE. # Output the corrected core and valence charge files AECCAR0 and AECCAR2

(3) **【Static Self-Consistent Calculation】** Perform a single-point static self-consistent

calculation to generate the charge density files AECCAR0 and AECCAR2.

(4) 【**Charge Preparation**】Use VASPMATE --bader -comb AECCAR0 AECCAR2 to combine the charge densities and obtain the merged file CHGCAR_sum.

(5) 【**Charge Calculation**】Perform the Bader charge population analysis using bader CHGCAR -ref CHGCAR_sum, which will generate three files for further analysis: ACF.dat, BCF.dat, and AtomVolumes.dat.

(6) 【**Charge Analysis**】Use VASPMATE --bader -calc to calculate the charge of each atom, which will generate the Bader_Charge file.

3.5 Transition State Search via NEB

Nudged Elastic Band (NEB) is a commonly used method for searching transition states and reaction paths. In NEB calculations to search for a reasonable transition state, a supposed path containing a certain number of image structures needs to be constructed between initial and final structures, and then the transition state is obtained by means of various optimization algorithms. By considering the intrinsic issue of the NEB method in locating the transition state, Henkelman proposed the CI-NEB method which provides another definition of the force acting on the point with the highest energy. In CI-NEB method, this point is not subject to the spring force from adjacent points, preventing it from being pulled away from the transition state. Additionally, the sign of the potential energy component at this point parallel to the path direction is reversed, prompting it to climb towards the transition state in the direction of increasing energy. The CI-NEB method requires only fewer points, such as a total of 5 or even 3 points including the initial and final states, to accurately locate the transition state, making it one of the most efficient methods for finding transition states.

VASPMATE provides the following functions for NEB calculations of transition state energy barrier: (1) Comparing the similarity between the initial and final structures; (2) Calculating the number of interpolation points based on the difference between the initial and final structures, and automatically generating intermediate state structures; (3) After the calculation is completed for each intermediate structure, the OUTCAR is analyzed and transition state is saved to the corresponding file.

Mode: **Structure Similarity**

Syntax:

VASPMATE --neb -sim file1(INIPOS) file2(FINPOS)

Description:

This functionality is primarily used to compare the structural differences between the initial and final structures, and to suggest the interpolated points. The default input files are INIPOS and FINPOS.

Examples:

VASPMATE --neb -sim

VASPMATE --neb -sim INIPOS FINPOS

Mode: **Insert Points**

Syntax:

VASPMATE --neb -ins (number) (-line/-idpp)

Description:

This functionality is used to create intermediate states for NEB calculations based on the recommended interpolation values or a specified interpolation number (number >= 1). The number of interpolation points depends on the return value from the previous similarity check, and generally, it can be taken as (similarity return value / 0.8). This functionality requires four files in the current folder: INIPOS, INIOUT, FINPOS, and FINOUT, which represent the initial state structure POSCAR, the initial state calculation OUTCAR, the final state structure POSCAR, and the final state calculation OUTCAR, respectively. VASPMATE will create folders from the initial state to the final state according to the number of interpolation points, named 00, 01, 02, and so on. The -line/-idpp tags are used to select different interpolation methods: LINE (Linear Interpolation) and IDPP (Image Dependent Pair Potential). The default method is LINE. The folders for the initial and final states contain their corresponding POSCAR and OUTCAR files (the OUTCAR files corresponding to the initial and final states are copied to the corresponding folders for subsequent data analysis), while the intermediate state folders contain their corresponding POSCAR files.

Examples:

VASPMATE --neb -ins

VASPMATE --neb -ins 5 -idpp

Mode: **Output Data**

Syntax:

VASPMATE --neb -out

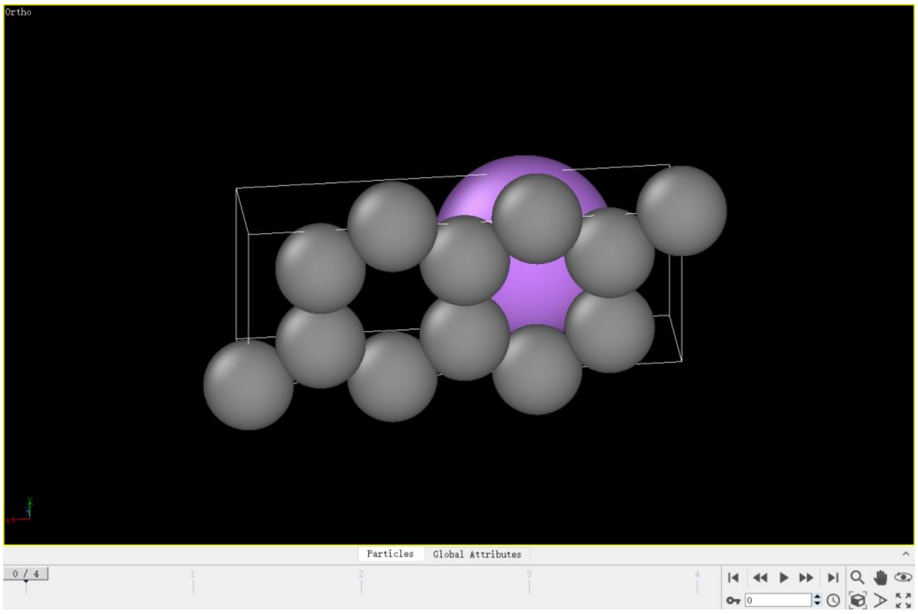
Description:

After using VASPMATE --neb -ins to complete the interpolation, the intermediate state structures will be generated (each intermediate state folder should contain the corresponding CONTCAR and OUTCAR files). At this point, This functionality can be used to extract and analyze the calculated data, with the output files being neb.dat and movie.xyz.

The content of the neb.dat file is as follows: the first column represents the structure number, the second column represents the distance (the calculation result of the distance between two adjacent structures), the third column represents the total energy, the fourth column represents the relative energy with the initial state energy as the reference value, and the fifth column represents the force (the force along the NEB path). Note: There seems to be a discrepancy in the column description provided, as there are five columns but only four mentioned initially. It is assumed that the fourth column description was a typo and have added a fifth column description for the force, which is typically included in NEB analysis.

Number	Distance	Energy	Relative Energy	Force
0	0.000000	-111.885923	0.000000	0.000000
1	0.625837	-111.765665	0.120258	-0.248757
2	1.252108	-111.680083	0.205840	-0.001109
3	1.876502	-111.763987	0.121936	0.247196
4	2.500640	-111.885900	0.000023	0.000000

The movie.xyz file can be used for animations in VMD or viewed in OVITO（jmol）to observe the entire process. Below is the snapshot of importing the movie.xyz file for the LiC_{12} structure into OVITO during NEB calculation.



Examples:

VASPMATE --neb -out

【Procedure】The CINEB transition state search using VASPMATE is as follows:

(1) **【Structure Optimization】** Optimize the initial and final state structures.

(2) **【Transition State Preparation】** Use VASPMATE --neb -sim to check the similarity between the two optimized structures (two CONTCAR files) by calculating the root mean square distance between corresponding atoms in the initial and final states. If the return value is less than 5 Angstroms, the next step can generally be proceeded.

(3) **【Transition State Preparation】** Use VASPMATE --neb -ins to insert intermediate state points. The number of interpolation points depends on the return value of the previous similarity check, and generally, the number of interpolation points can be taken as the return value multiplied by 1.2. If the interpolation is successful, folders such as ./00, ./01, ./02, ..., ./0N will be automatically generated. Here, 00 represents the initial state, and 0N represents the final state. The file names inside the folders are all POSCAR.

(4) **【Transition State Preparation】** Copy the corresponding OUTCAR files of the initial and final states into their respective folders for subsequent data analysis.

(5) **【Transition State Preparation】** Use VASPMATE --inp neb to generate the INCAR_neb file and copy it to INCAR. Since the transition state needs structure optimization, the modifications can be made based on the INCAR file for structure optimization:

- EDIFF=1E-7 # Calculation accuracy requirement (Note: For rough convergence, it can be appropriately relaxed to 1E-5)
- EDIFFG = -0.03 # Structural optimization convergence accuracy (-0.05 can be used for difficult-to-converge systems)
- IOPT = 1 # Optimization algorithm, 7, 2, or 1 are recommended, 0 represents the algorithm provided by VASP
- IBRION = 3, POTIM = 0 # Tags to activate the VTST optimization algorithm
- ICHAIN=0 # Enable the NEB method
- LCLIMB = .TRUE. # CI-NEB
- IMAGES = 1 # Number of interpolation points
- SPRING = -5 # Spring force constant, -5 is the default value

The following parameters are read from the INCAR file.

(IOPT = 0) Use VASP optimizers specified from IBRION (default)
(IOPT = 1) LBFGS = Limited-memory Broyden-Fletcher-Goldfarb-Shanno
(IOPT = 2) CG = Conjugate Gradient
(IOPT = 3) QM = Quick-Min
(IOPT = 4) SD = Steepest Descent
(IOPT = 7) FIRE = Fast Inertial Relaxation Engine

(6) **【Transition State Search】** Use VASP to perform transition state search optimization.

(7) **【Transition State Extraction】** Use VASPMATE --neb -out to extract and analyze the generated data, with the output files being neb.dat and movie.xyz. The neb.dat file provides information related to the final minimum energy path, while the movie.xyz file may be used for checking the reasonableness of the interpolation points.

3.6 Calculation of Differential Charge Density

VASPMATE provides powerful and simple post-processing capabilities for the differential charge density calculations, with the generated file in CHGCAR format. Charge Density Difference (CDD) is one of the important tools for studying electronic structure, as it provides an intuitive information of electronic distribution, thereby facilitating the analysis of the nature of chemical bonds.

Differential charge density is mainly divided into the following categories:

(1) Subtracting the charge densities of two or more constituent structures from those of compound structure, as follows:

$$\Delta\rho = \rho_{AB} - \rho_A - \rho_B$$

(2) The difference in charge density before and after a self-consistent calculation, also known as deformation charge density, is as follows:

$$\Delta\rho = \rho\left(AB_{self-consistent}\right) - \rho(AB_{atomic})$$

(3) Subtracting the charge density of one electronic state from another, is as follows:

$$\Delta\rho = \rho(AB_{state1}) - \rho(AB_{state2})$$

Note: In all charge density calculations, it is essential to maintain consistency in cell parameters, atomic positions, and grid mesh. Additionally, for files in the CHGCAR format, such as the electrostatic potential (LOCPOT), the electron localization function (ELFCAR), and the partial charge density (PARCHG), the differences can also be calculated using the same method.

Mode: **Charge Density Split**

Syntax:

```
VASPMATE --vcd -split file(CHGCAR)
```

Description:

When using the spin polarization parameter (ISPIN=2), the CHGCAR file can be used to calculate the total charge density and the spin charge density. The default output files are CHGTOT.vasp, CHGSPIN.vasp, CHGSPIN_UP.vasp, and CHGSPIN_DW.vasp, which can be visualized in VESTA or SPaMD softwares.

Examples:

VASPMATE --vcd -split

Mode: **Charge Density Summation**

Syntax:

VASPMATE --vcd -sum [File_list]

Description:

This functionality is used to quickly calculate the sum of multiple charge densities. [File_list] is the list of CHGCAR files that need to be processed, and the -sum command is used to sum multiple charge densities. The default output file is CHGSUMM.vasp, which can visualized in VESTA or SPaMD softwares. If the charge densities are inconsistent, VASPMATE will print an error message: "Attention: The CHGCAR files are not consistent!"

Examples:

VASPMATE --vcd -sum CHGCAR_C CHGCAR_O

Mode: **Charge Density Difference**

Syntax:

VASPMATE --vcd -diff [File_list]

Description:

This functionality is used to quickly calculate the difference between multiple charge density files. [File_list] represents the list of CHGCAR files to be processed. The first CHGCAR file listed after the -diff command is the charge density to be subtracted from, while the remaining CHGCAR files are the charge densities to be subtracted. The default output file is named CHGDIFF.vasp, which can be visualized using VESTA or SPAMD. If the charge densities are not consistent, VASPMATE will print an error message: "Attention: The CHGCAR files are not consistent!"

Examples:

VASPMATE --vcd -diff CHGCAR_CO CHGCAR_C CHGCAR_O

【Procedure】The calculation of spin charge density using VASPMATE is as follows:

(1) **【Structure Optimization】** Perform normal structure optimization on the initial structure. Copy CONTCAR to INPOS.

(2) **【Static Preparation】** Based on the optimized unit cell, use VASPMATE --ka -par 4000 to generate a K-point sampling file NEWKPT from INPOS, and use VASPMATE --inp scd to generate the incar_scd file (Note: Enable the spin polarization parameter: ISPIN=1). Copy these files to KPOINTS and INCAR, respectively.

(3) **【Static Self-Consistent Calculation】** Perform a single-point static self-consistent calculation to obtain CHGCAR.

(4) **【Charge Density Extraction】** Use VASPMATE --vcd -split to calculate the total charge density and spin charge density based on CHGCAR. The output files are CHGTOT.vasp, CHGSPIN.vasp, CHGSPIN_UP.vasp, and CHGSPIN_DW.vasp.

(5) **【Charge Density Analysis】** Use VESTA or SPaMD to visualize the total charge density and spin charge density.

3.7 Calculation of Band Edge Charge Density

The calculations of charge densities corresponding to the valence band maximum (VBM) and conduction band minimum (CBM) play a critical role to illustrate the contributions of specific atoms to the VBM and CBM. In systems with surface states, it is also often necessary to calculate the charge density to underline the electronic mechanism. VASPMATE provides a convenient solution to get the band edge charge density.

Firstly, the structural optimization and static self-consistent calculations need to be performed on the target structure in sequence to obtain CHGCAR and WAVECAR files. Then, the band edge charge density can be calculated in the following three modes:

(1) Specify the band index and K-point number; (2) Specify the energy range [E1, E2]; (3) Specify the region near the Fermi level ([EF-EINT, EF]). Note that VASPMATE provides a default pcd template to generate INCAR file with a simple modification for the aforementioned three modes.

Mode: **Partial Charge Density**

Syntax:

VASPMATE --pcd -ib/-ik/-en/-ef [value]

Description:

For the three modes mentioned above, four options (-ib/-ik/-en/-ef) are provided to modify the selected energy band, the selected K-point, the energy range, and the width near the Fermi level. Note that the modification should correspond to the parameters in INCAR. Among them, -ik and -ib correspond to the first mode and can be used consecutively (see example). Note: One may use VASPMATE --band -id to view the band index and K-point number near the Fermi level, or directly obtain the required K-point number and band index by viewing the OUTCAR file (or viewing the IBZKPT and EIGENVAL files). The data contained in the IBZKPT file is the number of K-points in the irreducible Brillouin zone, arranged in order. When generating K-points in automatic mode, VASP will automatically generate a simplified K-point matrix, which is stored in the IBZKPT file and can be directly copied to the KPOINTS file for use. The EIGENVAL file is an eigenvalue file that provides the eigenvalues of a series of energy bands for each selected K-point. Using the INCAR file generated by this mode, VASP will generate a PARCHG file, which corresponds to the partial charge density in the same format of CHGCAR. The following figure shows the incar_pcd_ik file generated by the command VASPMATE --pcd -ik, corresponding to the first mode.

```
#Decomposed Charge Density
ISTART = 1        (Job: 0 - new  1 - cont  2 - samecut)
ICHARG = 1        (Read charge : 1 - file 2 - atom 10 - const)
LPARD = .TRUE.        (Activate decomposed charge density)
LSEPB = .TRUE.        (Separately write PARCHG.nb by every band or not)
LSEPK = .TRUE.        (Separately write PARCHG.nk by every kpoint or not)

# Method I : Partial Charge for the specified BANDSand KPOINTS
IBAND = 20 21 22 23   (Set this parameters manually)
KPUSE = 1 2 3 4       (Set this parameters manually)

# * *********Notes * ***********
# (1) Copy IBZKPT as KPOINTS for static calculation,
# (2) Band structure calculation.
```

Examples:

VASPMATE --pcd -ik 1 2 3 4

VASPMATE --pcd -ib 1 2 3 4

VASPMATE --pcd -ik 1 2 -ib 3 4

VASPMATE --pcd -en -10 -5

VASPMATE --pcd -ef -1

【Procedure】The calculation of band edge charge density using VASPMATE is as follows:

(1) **【Structure Optimization】** After constructing the initial structure POSCAR, perform normal structure optimization. Copy the optimized CONTCAR to INPOS.

(2) **【Static Preparation】** Based on the optimized cell, use VASPMATE --std3d to generate a standard primitive cell STD3POS from INPOS, use VASPMATE --ka -par 4000 to generate a K-point sampling file NEWKPT from INPOS, and use VASPMATE --inp stc to generate the incar_stc file. Copy these files to POSCAR, KPOINTS, and INCAR, respectively.

(3) **【Static Self-Consistent Calculation】** Perform a single-point static self-consistent calculation to obtain CHGCAR and WAVECAR, which will be used for subsequent non-self-consistent band structure calculations.

(4) **【Band Preparation】** Use VASPMATE --ka3d -par 20 (where 20 is the number of sampling points) to generate a K-point path NEWKPATH for band structure calculation based on INPOS, and copy it to KPOINTS.

(5) 【**Band Preparation**】 Use VASPMATE --inp nsc to generate the incar_nsc file for non-self-consistent calculation, and copy it to INCAR. Note: For non-self-consistent calculations, set ISTART=1 and ICHARG=11.

(6) 【**Band Calculation**】 Perform a non-self-consistent calculation by reading the WAVECAR and CHGCAR generated from the previous static self-consistent calculation.

(7) 【**Band Extraction**】 Use VASPMATE --band -b to extract band structure. This will generate Band.dat and BAND_REFORMATTED.dat files to save band structure, and the KLABELS file to save the symbols and coordinate positions of high-symmetry points on the band structure.

(8) 【**Band Analysis**】 Use VASPMATE --band -id to view the band index (-ib) and high-symmetry K-point number (-ik) near the Fermi level, or directly open the OUTCAR file (or IBZKPT and EIGENVAL files) to check the required K-point number and band index.

(9) 【**Charge Preparation**】 Use VASPMATE --pcd -ib xxx -ik xxx to generate the incar_pcd_ibxxx_ikxxx file corresponding to the band index and K-point number, and copy it to INCAR.

(10) 【**Charge Self-Consistent Calculation**】 Perform a single-point static self-consistent calculation, and the generated PARCHG file in which stores the partial charge density in the same format of CHGCAR.

3.8 Visual Analysis of Real Space Wave Functions

VASPMATE can extract the plane wave coefficients of Kohn-Sham (KS) orbitals from the WAVECAR file and output the real-space wave functions for visualization. Users only need to specify the corresponding k-point number and band index. Note: VASPMATE can only be used to extract the wave function of a specified single K-point and a single energy band. If an accumulated wave function information is needed for multiple K-points or within a certain energy range, one needs to use the band edge charge density method.

Mode: **Wave Function Visualization**

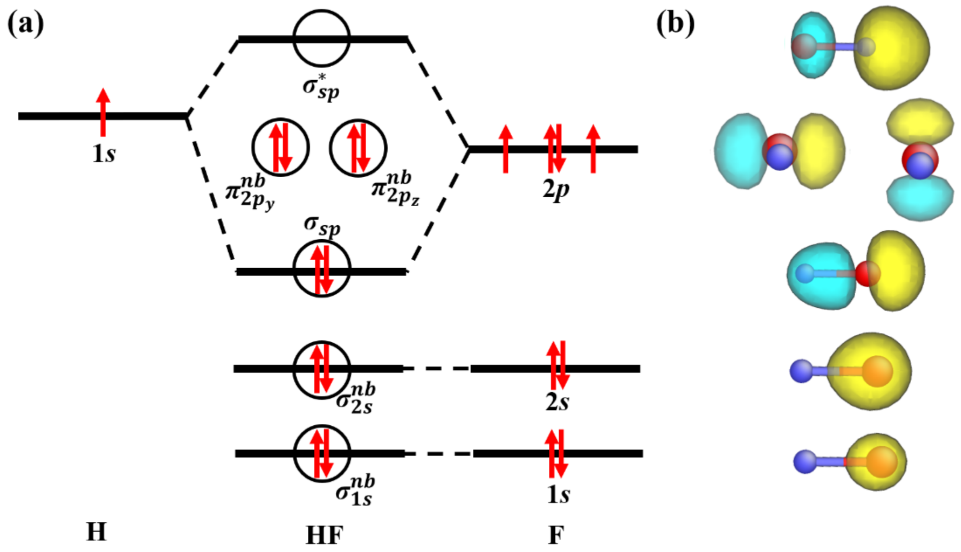
Syntax:

VASPMATE --wfn -ik [kpoint_index] -ib [band_index]

Description:

This functionality allows for quick and convenient retrieval of wave function in real space from the WAVECAR file based on user-specified K-point number and/or band index. Users can freely select the K-point number and band index for visualization using the -ik and -ib parameters. If not specified, both default to the minimum value of 1. Note: Users can use VASPMATE --band -id to view the band index and K-point number near the Fermi level, or directly check the OUTCAR file (or examine the IBZKPT and EIGENVAL files) to obtain the required K-point number and band index. The IBZKPT file contains the number of K-points in the irreducible Brillouin zone. When generating K-points in automatic mode, VASP will generate a reduced K-point matrix stored in the IBZKPT file, which can be directly copied to the KPOINTS file. The EIGENVAL file is the eigenvalue file, which provides eigenvalues for a series of energy bands at various selected K-points. The output filenames are wfn_Bxxx_Kxxx_REAL.vasp and wfn_Bxxx_Kxxx_IMAG.vasp, representing the real and imaginary parts of the wave function, respectively. For ISPIN=2 cases, UP and DW files will be output separately. The output files can be visualized in VESTA or SPaMD softwares. This functionality requires the following files to be present in the current folder: CONTCAR provides the structure information, and WAVECAR provides the main wave function information (in binary format). The header information of the WAVECAR file includes the spin state SPIN, the number of K-points NKPTS, the number of energy bands NBANDS, the cutoff energy ENCUT, and the maximum G value in reciprocal space (GX, GY, and GZ).

The figure below shows the calculated molecular orbital energy levels of HF, indicating that the combination of H and F atomic orbitals forms six molecular orbitals of HF, and the corresponding orbital energies gradually increase from bottom to top.



Examples:

VASPMATE --wfn

VASPMATE --wfn -ik 10 -ib 10

【Procedure】The calculation of real-space wave functions using VASPMATE is as follows:

(1) **【Structure Optimization】** After building a structure in POSCAR, perform normal structure optimization. Copy the optimized CONTCAR to INPOS.

(2) **【Static Preparation】** Based on the optimized structure, use VASPMATE --std3d to generate the standard primitive cell STD3POS from INPOS, use VASPMATE --ka -par 4000 to generate the K-point sampling file NEWKPT from INPOS, and use VASPMATE --inp stc to generate the incar_stc file. Then copy them as POSCAR, KPOINTS, and INCAR, respectively.

(3) **【Static Self-Consistent Calculation】** Perform a single-point static self-consistent calculation to obtain CHGCAR and WAVECAR, which will be used for subsequent non-self-consistent band structure calculations.

(4) **【Band Preparation】** Use VASPMATE --ka3d -par 20 (where 20 is the number of sampling points) to generate the K-point path file NEWKPATH for band structure calculation based on INPOS, and copy it to KPOINTS. Use VASPMATE --inp nsc to generate the incar_nsc file and copy it to INCAR. Note: For non-self-

consistent calculations, set ISTART=1 and ICHARG=11.

(5) 【**Band Calculation**】 Perform a non-self-consistent calculation by reading the WAVECAR and CHGCAR that are generated from the previous static self-consistent calculation.

(6) 【**Band Analysis**】 Use VASPMATE --band -b to extract band structure. The generated Band.dat and BAND_REFORMATTED.dat files will save the band information, while the KLABELS file will save the symbols and coordinate positions of high-symmetry points on the band diagram.

(7) 【**Band Analysis**】 Use VASPMATE --band -id to view the band index -ib and high-symmetry K-point number -ik near the Fermi level, or directly open the OUTCAR file (or IBZKPT and EIGENVAL files) to check the required K-point number and band index.

(8) 【**Wave Function Extraction**】 Use VASPMATE --wfn -ib xxx -ik xxx to generate files corresponding to the band index and K-point number, namely wfn_Bxxx_Kxxx_REAL.vasp and wfn_Bxxx_Kxxx_IMAG.vasp, representing the real and imaginary parts of the wave function, respectively (for spin calculations, UP and DW files will be output separately).

(9) 【**Wave Function Analysis**】 The output files can be visualized in VESTA or SPaMD softwares.

3.9 Visual Analysis of Fermi surface

In condensed matter physics, the Fermi surface is often used to represent the boundary between occupied and unoccupied states. Its shape is determined by the periodicity and symmetry of reciprocal space, as well as the occupation of electronic energy bands. VASPMATE provides the following two functions for Fermi surface calculations: (1) Automatically generate k-points for Fermi surface calculations, and (2) Extract Fermi surface to save in FERMISURFACE.bxsf or FERMISURFACE.frmsf files. These two files can be visualized in SPaMD, XcrySDen[7], and fermisufer[8].

Mode: **Fermi Kpoints**

Syntax:

VASPMATE --fska (INPOS) (-par) [kppra] [kscheme]

VASPMATE --fskv (INPOS) (-par) [kspac] [kscheme]

VASPMATE --fskm (-par) [k1 k2 k3] [kmesh] [kscheme]

Description:

Similar to the standard method for automatically generating k-points, VASPMATE offers three different k-point sampling methods to generate weighted irreducible k-points within the first Brillouin zone (similar to the approach used in HSE calculations). When calculating the Fermi surface, it is recommended to use the --fskv method for uniform k-point sampling to achieve higher accuracy, with a suggested kspac value of 0.05. The generated FERMIKPT file stores the k-point information.

Examples:

VASPMATE --fskv 0.05 G (XXX)

VASPMATE --fskv -par 0.05 G

VASPMATE --fskv INPOS 0.05 G

Mode: **Fermi Surface**

Syntax:

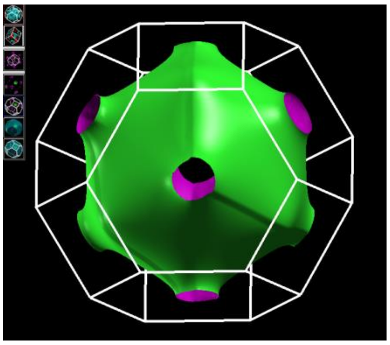
VASPMATE --fsxd -ib [band_index]

VASPMATE --fs -ib [band_index] -io [atom_index] [orbital_index]

Description:

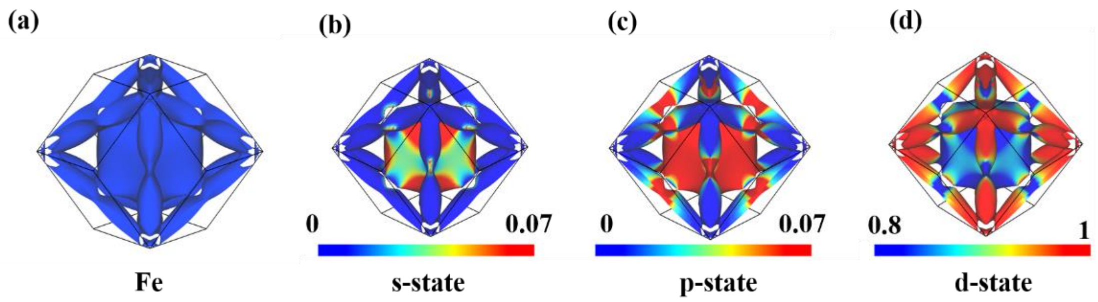
For the post-processing of Fermi surface calculations, VASPMATE provides interfaces for mainstream Fermi surface analysis software such as XcrySDen, fermisurfer, or SPaMD. This functionality requires the presence of the following files in the current folder: CONTCAR provides the structure information, OUTCAR provides the basic output information, EIGENVAL provides the information of energy eigenvalues and K-points, and PROCAR provides the information of band weight. VASPMATE extracts the energy eigenvalue from the EIGENVAL file and assigns it to each K-point based on symmetry. Additionally, by comparing the eigenvalues in the EIGENVAL file, VASPMATE identifies the energy bands that are closest to or cross the Fermi level as the Fermi surface (the Fermi level value can be specified in the FERMI_LEVEL file, or by default, it is read from the DOSCAR file; see the section of density of states calculation for more details). Besides the default mode to select the energy bands near the Fermi level, users can specify the desired band index using the -ib parameter, which can be viewed using VASPMATE --band -id. Now, let's introduce two output methods.

The first command format, --fsxd, indicates that the Fermi surface will be exported in a format recognizable by the XcrySDen program. The image below shows the visualization of the Fermi surface of Cu obtained by importing the FERMISURFACE.bxsf file into XcrySDen or SPaMD.



To be noted that XcrySDen can currently only recognize Fermi surfaces without any additional weights. In contrast, Fermisufer and SPaMD appears to be more comprehensive. VASPMATE can apply different DOS weights from PROCAR to each K-point based on the user-selected atoms and their orbitals, and finally output a data file with corresponding weights for visualization in Fermisufer and SPaMD. The band weights can be selected by the atom_index using the -io tag along with the atomic number or element symbol, and choose the corresponding orbital using the orbital name (note that when LORBIT=11 is set, VASPMATE can still recognize spdf orbitals, and the output value will be the sum of the corresponding split orbitals).

The image below shows the visualization of the Fermi surface of Fe through importing the data file FERMISURFACE.frmsf into SPaMD or fermisufer. From left to right, it shows the Fermi surface without any weights, and with weights applied based on the contributions of the spd orbitals of Fe atoms, respectively.



Examples:

VASPMATE --fsxd

VASPMATE --fsxd -ib 1 2

VASPMATE --fs -io B s dx

VASPMATE --fs -ib 1 2 -io 1 s

【Procedure】The calculation of Fermi surface using VASPMATE is as follows:

(1) **【Structural Optimization】** Perform normal structure optimization on the initial structure. Copy the optimized CONTCAR to INPOS.

(2) **【Static Preparation】** Obtain a higher-precision FERMIKPT file using VASPMATE --fskv -par 0.05 G for uniform k-point sampling, and copy it to KPOINTS.

(3) **【Static Preparation】** Generate the incar_stc file required for a single-point static self-consistent calculation using VASPMATE --i stc, and copy it to INCAR. Note: If the orbital weight needs to be output, set LORBIT=11 in INCAR.

(4) **【Static Self-Consistent Calculation】** Perform a single-point static self-consistent calculation to obtain OUTCAR (basic information), DOSCAR (Fermi level), IBZKPT (K-point path), EIGENVAL (energy eigenvalues), and PROCAR (orbital contribution) files.

(5) **【Fermi Preparation】** Extract energy eigenvalue from the EIGENVAL file and find the band index (-ib xxx) closest to or crossing the Fermi level as the Fermi surface.

(6) **【Fermi Analysis】** Finally, use VASPMATE --fs -ib xxx -io xxx to output the Fermi surface and orbital weight to the corresponding file *FERMISURFACE.frmsf.*

(7) 【**Fermi Analysis**】Use SPaMD or fermisufer to visualize the Fermi surface with the projection of orbital weights.

3.10 Visual Analysis of three-dimensional energy band

For two-dimensional materials, such as graphene, VASPMATE can automatically generate the k-point files by sampling points on the surface of the irreducible Brillouin zone. After processing the VASP calculation with VASPMATE, it can generate the distribution of k-points and energies that can be visualized by Origin or using SPaMD.

Mode: **3dband Kpoints**

Syntax:

VASPMATE --3dka (INPOS) (-par) [kppra] [kscheme]

VASPMATE --3dkv (INPOS) (-par) [kspac] [kscheme]

VASPMATE --3dkm (-par) [k1 k2 k3] [kmesh] [kscheme]

Description:

The k-points for 3D energy bands also continue VASPMATE's sampling style, offering three different sampling methods. For 3D energy band calculations, it is recommended to use the --3dkv uniform sampling method to achieve higher accuracy. The suggested value for kspac is 0.05, and the k-point information is saved in the generated 3dbandKPT file.

Examples:

VASPMATE --3dkv INPOS 0.05 G

VASPMATE --3dkv -par 0.05 G

Mode: **3dband surface**

Syntax:

VASPMATE --3dbs -ib [band_index]

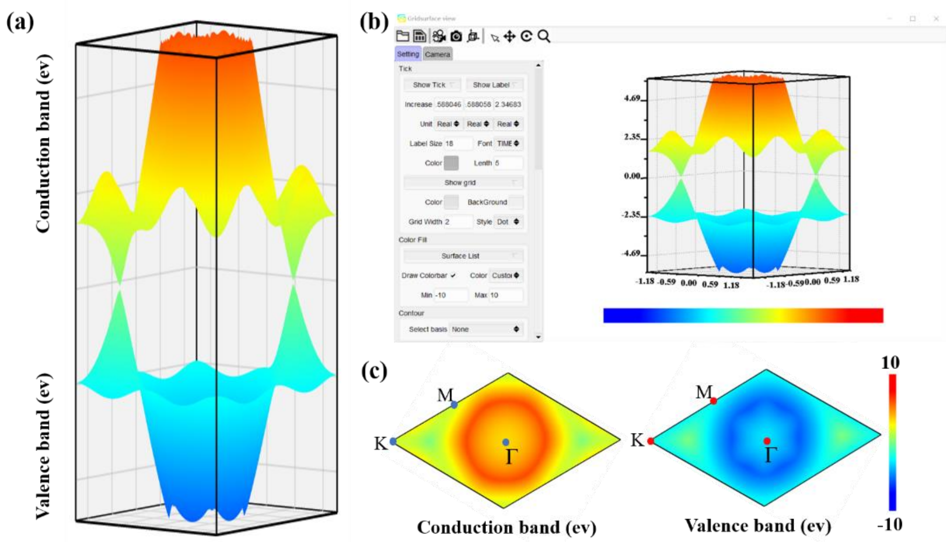
Description:

This functionality requires the following files to be present in the current folder: CONTCAR provides the structure information, OUTCAR provides the basic output information, EIGENVAL provides the information of energy level and band, and KPOINTS provides the x and y coordinate of the reciprocal space k-points. After completing the calculation of 3D energy bands, VASPMATE provides two processing methods. One is to specify the band index, which outputs the corresponding numbered energy band, and multiple bands can be specified at once, controlled by the parameter -ib (-ib specifies the band index near the Fermi level, which can be viewed using VASPMATE --band -id), and they are output to BAND_B*.grd files separately (for systems with spin, they are output to BAND_B*_UP.grd and BAND_B*_DW.grd files respectively). The other method does not need specify a number, and VASPMATE automatically finds the LUMO and HOMO bands and outputs their structures. In addition, VASPMATE will also output two files, KX.grd and KY.grd, which save the x and y coordinates of each k-point in reciprocal space, respectively. By importing the above three files into Origin or SPaMD and using 3D color mapping surface plotting, the following energy band structure diagram can be obtained (the figure below shows the two imported energy bands near the Fermi level).

Examples:

VASPMATE --3dbs

VASPMATE --3dbs -ib 5 6



【 Procedure 】 The calculation of three-dimensional energy bands using VASPMATE is as follows:

(1) 【**Structural Optimization**】 Perform normal structure optimization on the initial structure. Copy the optimized CONTCAR to INPOS.

(2) 【**Static Preparation**】 Use VASPMATE --std3d to generate a standardized primitive cell STD3POS based on INPOS, and use VASPMATE --ka 8000 to generate NEWKPT based on INPOS. Copy STDNPOS to POSCAR and NEWKPT to KPOINTS.

(3) 【**Static Self-Consistent Calculation**】Perform a single-point static self-consistent calculation to obtain CHGCAR and WAVECAR files for subsequent energy band structure calculations.

(4) 【**Energy Band Preparation**】Obtain a higher-precision 3dbandKPT file using VASPMATE --3dkv -par 0.05 G for uniform k-point sampling, and copy it as KPOINTS. Use VASPMATE --bbs to generate an incar_bbs file and copy it as INCAR. Note: For non-self-consistent calculations, set ISTART=1 and ICHARG=11.

(5) 【**Energy Band Calculation**】 Perform a non-self-consistent energy band calculation. VASP outputs the following files: OUTCAR, EIGENVAL, and PROCAR.

(6) 【**Energy Band Analysis**】Use VASPMATE --3dbs -ib xxx to output energy band to BAND_B*.grd files, and also output two files, KX.grd and KY.grd, which save the x and y coordinates of each k-point in reciprocal space, respectively.

(7) 【**Energy Band Analysis**】Import the files into Origin or SPaMD, and use 3D color mapping surface to obtain the energy band structure diagram.

3.11 Automatic Database Construction

3.11.1 Database of sdata and json

At default, VASPMATE can be used to extract the critical input parameters and important output information by parsing the OUTCAR file which are saved to a log.sdata (or log.json) file together with the related files (an SDaMS format or standard

JSON format).

Syntax:

VASPMATE --db file(log.sdata/log.json) (-js)

VASPMATE --db file(log.sdata/log.json) (-b/-a) (-inc/-include) [files (DOSCAR/ EIGENVAL/ PROCAR etc.)] (-js)

Description:

For each VASP calculation, VASPMATE extracts the critical input parameters and important output information by parsing the OUTCAR file. The former includes calculation mode, cutoff energy, functional type, correction method, etc., while the latter includes energy, stress, forces, charge, and magnetic moment, etc. VASPMATE saves these information in a key-value storage format to a log.sdata (or log.json) file together with the related files (an SDaMS format or standard JSON format). By default, the parameter -b corresponds to the inclusion of INCAR, KPOINTS, POSCAR, CONTCAR, and the parameter -a (or no option) will include OSZICAR and OUTCAR in addition to the above four files. The option -inc/-include is used to add more files such as XDATCAR, DOSCAR, CHGCAR, EIGENVAL, PROCAR, etc. To be noted that the user may also use this command together with the script spd.sh to include all related files (located in the VASPMATE installation package directory). The default output filename is log.sdata (or log.json for flag -js). This functionality requires the following files to be present in the current folder: INCAR provides the information of control parameters, KPOINTS provides the information of reciprocal space K-points, POSCAR provides the initial structure information, CONTCAR provides the final structure information, and OUTCAR provides the detailed process information during the VASP run (including not only the existing information from the input files, but also basic information such as symmetry analysis of the calculated system, the K-space sampling points and position coordinates, the information of plane wave basis set, and the nearest neighbor atomic distances. Additionally, it records the calculation information for each step of ion relaxation and electronic self-consistency).

The structure of the OUTCAR file is as follows: VASP version and basic computing environment and resource information; reading INCAR, POTCAR, POSCAR, KPOINTS; nearest neighbor atom and symmetry analysis information; calculation process control parameter information (including default values); unit cell direct space and reciprocal space information and atomic coordinate information; plane wave basis set information, including cutoff energy and number of plane waves; non-local pseudopotential information; information for each electronic step (iteration time and energy information); energy eigenvalue information; atomic force, stress tensor, and unit cell information; system ground state total energy and free energy information; and resource consumption and time statistics for completing the calculation.

Examples:

VASPMATE --db

VASPMATE --db -b -js

VASPMATE --db log.sdata

VASPMATE --db log.sdata -inc DOSCAR

VASPMATE --db log.sdata -inc EIGENVAL/ PROCAR

Syntax:

VASPMATE --db2js file1(log.sdata) file2(log.json)

VASPMATE --js2db file1(log.json) file2(log.sdata)

Description:

This functionality converts the .sdata (.json) data format file (file1) into a common JSON (SDaMS) data format file (file2). The default input filename is log.sdata (log.json), and the output filename is log.json (log.sdata).

Examples:

VASPMATE --db2js log.sdata log.json

VASPMATE --db2js log.sdata

VASPMATE --db2js

VASPMATE --js2db

VASPMATE --js2db log.json

Syntax:

VASPMATE --dbplus file1(log1.sdata) file2(log2.sdata) ... (-js)

Description:

This functionality combine the .sdata data format file2(log2.sdata) ... into file1(log1.sdata), or .json data format file1(log1.json) file2(log2.json) … to file(newlog.json). The default input filename is log1.sdata (log1.json) and log2.sdata (log2.json).

Examples:

VASPMATE --dbplus log1.sdata log2.sdata

VASPMATE –dbplus log1.json log2.json -js

3.11.2 Database of sqlite3

VASPMATE can also be used to extract the critical input parameters and important output information by parsing the OUTCAR file, and then to save them into a SQLITE3 database (a .db file).

Syntax:

VASPMATE --dbs database(vasp.db)

VASPMATE --dbs vasp.db -tables

VASPMATE --dbs vasp.db -schema [table-name]

VASPMATE --dbs vasp.db -table [table-name] -create [keyword] [type]

VASPMATE --dbs vasp.db -table [table-name] -alter [table-name-new]

VASPMATE --dbs vasp.db -table [table-name] -drop

VASPMATE --dbs vasp.db -table [table-name] -set [keyword] [type]

VASPMATE --dbs vasp.db -table [table-name] -select [condition]

VASPMATE --dbs vasp.db -table [table-name] -insert [keyword] [values]

VASPMATE --dbs vasp.db -table [table-name] -delete [condition]

VASPMATE --dbs vasp.db -plus vasp1.db

VASPMATE --dbs vasp.db -table [table-name] -merge [table-name1 table-name2]

Description:

This functionality create the SQLITE database with tables. The default input filename is vasp.db. The flag -tables is used to list all tables, -schema is used to list the structure of a specified table, -table [table-name] is used to specify the table which will be operated on. The flag -create is used to create table, and its default name is vaspinfo, -alter is used to alter table name, -drop is used to drop/delete a table from the database, -set is used to add a column, -select is used to select one or more rows/lines according to the defined condition, -insert is used to insert a row/line, -delete is used to delete one or more rows/lines according to the defined condition. The flag -plus is used to merge the data tables in vasp1.db into vasp.db. If the table names are identical, they will be merged; if they differ, they will be appended. The flag -merge is used to combine the data tables vaspinfo1 and vaspinfo2 into vaspinfo which can be either vaspinfo1 or vaspinfo2, or a new one.

Examples:

VASPMATE --dbs vasp.db

VASPMATE --dbs vasp.db -tables

VASPMATE --dbs vasp.db -schema vaspinfo

VASPMATE --dbs vasp.db -table vaspinfo -alter vaspinfo_new

VASPMATE --dbs vasp.db -table vaspinfo -drop

VASPMATE --dbs vasp.db -table vaspinfo -create id int primary key, name text

VASPMATE --dbs vasp.db -table vaspinfo -create num int, name text

VASPMATE --dbs vasp.db -table vaspinfo -set address, text

VASPMATE --dbs vasp.db -table vaspinfo -select "id<5 AND name='luck'"

VASPMATE --dbs vasp.db -table vaspinfo -insert id,name,address 5,lucky,Beijing

VASPMATE --dbs vasp.db -table vaspinfo -delete "id=5"

VASPMATE --dbs vasp.db -plus vasp1.db

VASPMATE --dbs vasp.db -table vaspinfo -merge vaspinfo1 vaspinfo2

Syntax:

VASPMATE --db2s name.db -table [table_name] (-a/-b) (-inc/include file_name)

Description:

This functionality automatically extracts key data from folders processed by VASP computation and stores it in a specified SQL database (creates one if none exists), and automatically generates a .sdui file recognizable by SDaMS. The default file name is vasp.db, and the default table name is vaspinfo. By default, the parameter -b corresponds to the inclusion of INCAR, KPOINTS, POSCAR, CONTCAR, and the parameter -a (or no option) will include OSZICAR and OUTCAR in addition to the above four files. The option -inc/-include is used to add more files such as XDATCAR, DOSCAR, CHGCAR, EIGENVAL, PROCAR, etc.

Examples:

VASPMATE --db2s

VASPMATE --db2s -b

VASPMATE --db2s -a

VASPMATE --db2s vasp.db

VASPMATE --db2s vasp.db -b

VASPMATE --db2s vasp.db -a

VASPMATE --db2s -table vaspinfo -a

VASPMATE --db2s -table vaspinfo -b

VASPMATE --db2s -table vaspinfo -a -inc CHG CHGCAR

VASPMATE --db2s ~/test/vasp.db -table vaspinfo -a -inc CHG CHGCAR

3.12 Thermodynamic quantity correction

As an excellent first-principles calculation software package, VASP is less than satisfactory in thermodynamic calculations. It can only calculate the electronic energy of the system at 0K without considering the contribution of temperature, which obviously cannot meet the requirement to calculate free energy. To obtain reaction heat, the enthalpy needs to be calculated, and to obtain the internal energy at 0K, ZPE needs to be calculated. In contrast, Gaussian[9] performs better in this regard, but it obviously cannot satisfy users accustomed to VASP, and the calculations are relatively cumbersome. The good news is that Sobereva has developed a comprehensive thermodynamic calculation program called Shermo[10]. By analyzing Gaussian output results, the program calculates translational, rotational, vibrational, and electronic contributions to obtain the internal energy, enthalpy, entropy, free energy, and heat capacity of the system at a given temperature and pressure. Users of VASPMATE are advised to carefully read the Shermo appendix before calculating thermodynamic data, as it provides detailed information on the calculation methods and final contributions of various physicochemical properties, which can greatly help in understanding the output results. However, Shermo still cannot provide direct assistance to VASP users because it cannot recognize VASP output results, and it is not friendly adapted for high-throughput calculations. VASPMATE fills this gap, streamlining parts that may not be interesting to VASP users, and allowing for convenient and efficient thermodynamic correction of VASP output results.

Mode: **Thermo**

Syntax:

VASPMATE --thermo <-parameter> [value]

Description:

This functionality requires the presence of following files in the current folder: CONTCAR provides the structure information, while OUTCAR gives the energy, frequency, and other details. For output results, VASPMATE displays the most interesting aspects of VASP thermodynamic calculations (namely, the corrected thermodynamic quantities) on the screen. These include zero-point energy ($\varepsilon_{ZPE} = h\nu/2$), thermally corrected internal energy ($U(T) = \varepsilon_{ZPE} + \Delta U_{0\to T}$), thermally corrected enthalpy ($H(T) = \varepsilon_{ZPE} + \Delta U_{0\to T} + pV$), and thermally corrected free energy ($G(T) = \varepsilon_{ZPE} + \Delta U_{0\to T} + pV + TS$). Users can utilize the Linux redirection symbol >> to save the output content to a file, facilitating high-throughput extraction of thermodynamic data. Additionally, VASPMATE outputs a detailed breakdown of translational, rotational, vibrational, and electronic contributions to thermodynamics, along with their total sum, providing users with a comprehensive understanding of the entire thermodynamic correction process.

To offer convenient and precise thermodynamic corrections, This functionality provides six control parameters with appropriate default values. The parameters and their brief descriptions are as follows:

(1) -T/-t 300: This parameter sets the thermodynamic temperature in Kelvin, with a default value of 298.15K.

(2) -P/-p 1: This sets the current pressure in atmospheres (atm), where the actual pressure is calculated as p * 101.375KPa. The default is 1 atm.

(3) -s/-sm 1: This parameter determines the spin multiplicity of the current system. It can be simplified as the number of unpaired electrons + 1, with a default value of 1. Users are encouraged to adjust this based on their specific computational system, as an incorrect spin multiplicity can significantly affect the rotational contribution.

(4) -i/-im 0: This sets the motion modes considered during calculations, specifically whether to ignore the contributions of translation and rotation to thermodynamic calculations. The default is 0, meaning all contributions are considered. This setting is particularly useful in calculating the free energy of adsorbed molecules and will be further discussed later.

(5) -f/-lf 1: This parameter configures the low-frequency handling (lowfreq), addressing the treatment of lower frequencies. As lower vibration frequencies contribute more to entropy, very small vibration frequencies can lead to unusual entropy and free energy corrections. Based on Sobereva's approach in Shermo, VASPMATE offers three frequency handling methods: 1) Harmonic approximation (or RRHO), corresponding to -l 0 and is the default setting. However, the standard RRHO model can introduce significant errors in calculating free energy for lower frequencies[11] ,[12], leading to instability. 2) Raising low frequencies, which artificially raises low frequencies to a certain level, improving free energy calculations. The default frequency threshold is 100cm^{-1}. 3) Grimme's entropy interpolation[13], which interpolates between the vibrational entropy calculated under the harmonic approximation (S_{RRHO}) and the free rotor model (S_{FR}). This method provides a smoother transition compared to manually raising

the threshold. For general free energy calculations, it is recommended to enable the -l 1 parameter to reduce the impact of low frequencies. For more flexible systems (such as large molecules with many rotatable bonds or molecular complexes), setting it to -l 2 is advised.

(6) -v/-cv 100: This parameter corresponds to the second frequency handling mode, allowing users to manually set the frequency threshold (critical vibration). The default value is 100cm^{-1}.

Examples:

VASPMATE --thermo -T 300 -p 1 -s 3 -i 1 -l 1 -v 100

VASPMATE --thermo -T 300 -p 1 -s 3 -i 1 -l 1 -v 100 >> THER_INFO.dat

The following two examples of thermodynamic correction of gas phase molecules and thermodynamic correction of adsorbed molecules further introduce the whole process of using.

3.12.1 Thermodynamic correction of gas-phase molecules

Taking thermodynamic correction of H_2O molecules as an example, in molecular frequency calculations, the default value of -i/-im is set to 0, which means considering the thermodynamic contributions of both translational and rotational movements. For linear molecules, the smallest five frequencies are ignored, while for nonlinear molecules, the smallest six frequencies are ignored (this does not directly ignore the contributions of translational and rotational movements, but calculates their contributions to thermodynamic quantities through the partition function of translational and rotational movements. Among them, translational entropy is the main contributor to gas molecular entropy). After using the command VASPMATE --thermo -T 300 -p 1 -s 1 -l 1, the following information will be output on the screen:

```
[roo@localhost freq]$ VASPMATE --thermo -T 300 -p 1 -s 1 -l 1
================Thermo Energy Calculation!================
 Temperature(K): 300.000000
 Pressure(Atm): 1
 Spin multiplicity(Number of Unpaired electron +1): 1
 The treatment of low frequencies: Raising low frequencies.
 Raising lower frequencies to  50.000000 (cm^-1)!
 Molecular Symmetry: C2v
 Zero point energy (ZPE): 13.102430 kcal/mol 0.568175 ev
 Thermal correction to U: 14.893164 kcal/mol 0.645829 ev
 Thermal correction to H: 15.489325 kcal/mol 0.671681 ev
 Thermal correction to G: 1.927806 kcal/mol 0.083598 ev
 Total S                : 189.137994 J/mol/K 0.001960 ev/K
 TS                     : 56741.398078 J/mol 0.588083 ev
 More calculate details please read Thermol_Info.dat file!
==========================================================
```

(1) From the information output by VASPMATE, one may see the input parameters used in the current calculation, such as temperature, pressure, the method for processing low frequencies, and the point group of the molecule. Among various outputs, to obtain the free energy of H_2O, one simply needs to add the G value (i.e., 0.083598 of the same kind) to the final energy calculated by VASP.

(2) Additionally, in the output file THER_INFO.dat, one may first see the information about the computational system and the input values for the calculation, as follows:

```
            ======== Thermo Energy Calculated by VASPMATE ========
                 =========== Calculate Inparameter =========
 Temperature(K): 300.000000
 Pressure(Atm): 1
 Spin multiplicity(Number of Unpaired electron + 1): 1
 The treatment of low frequencies:   Raising low frequencies.
 Raising lower frequencies to  50.000000 (cm^-1)!
Note: Only for translation, U is different to H, and CV is different to CP
```

(3) VASPMATE then outputs the individual contributions of translational, rotational, vibrational, and electronic motions to the thermodynamic quantities.

```
                        ======== Translation ========
Translational q:  1.838161E+30    q/NA: 3.052338E+06
Translational U:  3.741509 kJ/mol 0.894242 kcal/mol
Translational H:  6.235849 kJ/mol 1.490404 kcal/mol
Translational S:  144.932915 J/mol/K  34.639798 cal/mol/K  -TS:-10.391939 kcal/mol
Translational CV: 12.471697 J/mol/K   2.980807 cal/mol/K
Translational CP: 20.786162 J/mol/K   4.968012 cal/mol/K

                        ========= Rotation ========
Rotational q: 4.525796E+01
Rotational U: 3.741509 kJ/mol 0.894242 kcal/mol    =H
Rotational S: 44.169584 J/mol/K   10.556784 cal/mol/K   -TS:  -3.167035 kcal/mol
Rotational CV:    12.471697 J/mol/K   2.980807 cal/mol/K  =CP

                        ======== Vibration ========
Vibrational q(V=0):   1.000496
Vibrational q(bot):   0.000000
Vibrational U(T)-U(0):    0.009412 kJ/mol 0.002250 kcal/mol    =H(T)-H(0)
Vibrational U:     54.829979 kJ/mol    13.104679 kcal/mol    =H
Vibrational S:     0.035495 J/mol/K    0.008483 cal/mol/K   -TS:   -0.002545 kcal/mol
Vibrational CV:    0.238908 J/mol/K    0.057100 cal/mol/K   =CP
Zero-point energy (ZPE):  54.820567 kJ/mol    13.102430 kcal/mol  0.020880 a.u.

                        ========= Electron ========
Electronic q: 1.000000
Electronic U: 0.000000 kJ/mol 0.000000 kcal/mol    =H
Electronic S: 0.000000 J/mol/K    0.000000 cal/mol/K   -TS:   -0.000000 kcal/mol
Electronic CV:    0.000000 J/mol/K    0.000000 cal/mol/K   =CP
```

(4) Finally, it outputs the sum of all contributions and the corrected thermodynamic results. Here, the `Electronic energy` is the electronic energy calculated by VASP, which is -14.219011 eV. By adding the correction to the Gibbs free energy, `G` which is 0.083598 eV, to this value, one obtain the final ‘Sum of electronic energy and thermal correction to G` which is -14.135414 eV, representing the free energy of the H_2O molecule under the current temperature and pressure conditions.

```
                    =========== Total =========
Total q(V=0): 8.323267E+31
Total q(bot): 2.373468E+22
Total q(V=0)/NA:  1.382111E+08
Total q(bot)/NA:  3.941236E-02
Total CV: 25.182302 J/mol/K   6.018715 cal/mol/K
Total CP: 33.496767 J/mol/K   8.005919 cal/mol/K
Total S:  189.137994 J/mol/K  45.205065 cal/mol/K -TS:    -56741.398078 J/mol -0.588083 ev
Zero point energy (ZPE):  54.820567 kJ/mol/K  13.102430 kcal/mol  0.020880 a.u.   0.568175 ev
Thermal correction to U:  62.312997 kJ/mol/K  14.893164 kcal/mol  0.023734 a.u.   0.645829 ev
Thermal correction to H:  64.807337 kJ/mol/K  15.489325 kcal/mol  0.024684 a.u.   0.671681 ev
Thermal correction to G:  8.065938 kJ/mol/K   1.927806 kcal/mol   0.003072 a.u.   0.083598 ev
Electronic energy:    -0.522539 a.u.  -14.219011 ev
Sum of electronic energy and ZPE, namely U/H/G at 0 K:    -0.501659 a.u.  -13.650836 ev
Sum of electronic energy and thermal correction to U: -0.498805 a.u.  -13.573183 ev
Sum of electronic energy and thermal correction to H: -0.497855 a.u.  -13.547331 ev
Sum of electronic energy and thermal correction to G: -0.519467 a.u.  -14.135414 ev

Thanks to Sobereva!(sobereva@sina.com)
More information about Thermol and Gaussian please refer to http://sobereva.com/552 and  https://gaussian.com/thermo/)
```

3.12.2 Thermodynamic correction of adsorbed molecules

In contrast to the calculation of the free energy of gas molecules, the adsorbed molecules form chemical bonds with the substrate, which reduces the degrees of freedom for translation and rotation, thereby limiting the contributions of translation and rotation to entropy and enthalpy. The approach adopted by VASPMATE is to attribute the contributions of translation and rotation to vibrations, meaning that all 3N degrees of freedom for the adsorbed surface molecules are considered for vibrational contributions in the thermodynamic calculations (note that VASPMATE ignores the effects of imaginary frequencies). These operations only require setting the imode parameter to 0 (it is worth noting that when imode is 0, since the contribution of electronic motion is very small at this point, VASPMATE will also ignore it). Taking the adsorption of a PO_2 molecule on a graphite surface as an example, the command VASPMATE --thermo -T 300 -p 1 -s 1 -l 1 -i 1 is entered, and the following results will be obtained on the screen.

```
[roo@localhost ORR]$ VASPMATE --thermo -T 300 -p 1 -s 1 -l 1 -i 1
===============Thermo Energy Calculation!===============
  Temperature(K): 300.000000
  Pressure(Atm): 1
  Spin multiplicity(Number of Unpaired electron +1): 1
  The treatment of low frequencies: Raising low frequencies.
  Raising lower frequencies to  50.000000 (cm^-1)!
  Molecular Symmetry: NULL!
 Warning: Cannot identify point group! Assume rotational symmetry number to be 1!
  Zero point energy (ZPE): 4.481927 kcal/mol 0.194355 ev
  Thermal correction to U: 6.482744 kcal/mol 0.281118 ev
  Thermal correction to H: 6.482744 kcal/mol 0.281118 ev
  Thermal correction to G: 2.885191 kcal/mol 0.125114 ev
  Total S                 : 50.173881 J/mol/K 0.000520 ev/K
  TS                     : 15052.164394 J/mol 0.156005 ev
  More calculate details please read Thermol_Info.dat file!
--------------------------------------------------------------
```

The zero point energy (ZPE) and the corrected results for U, H, and G can be clearly seen. By comparing these data with the output from Shermo, it can be observed that they are in complete agreement, which demonstrates the accuracy of the VASPMATE calculation results.

3.13 Formation enthalpies

The formation enthalpy ΔH for a multicomponent compound $A_xB_yC_z\cdots$ is calculated using the following formula:

$$\Delta H = \frac{1}{x+y+z+\cdots}\left[\, H\left(A_xB_yC_z\right) - xH(A) - \, yH(B) - \, zH(C) - \cdots \,\right]$$

where $H(A_xB_yC_z)$is the total enthalpy of compound $A_xB_yC_z$, and H(A), H(B), H(C) are the corresponding enthalpies of elements A, B, C, respectively. The composition x, y, and z are read from the POSCAR file while the total energy of compound is from the OUTCAR file in the current directory. Note that the lattice stability of elemental allotropy can also be derived when only elemental enthalpy is provided.

Mode: **Formation Enthalpy**

Syntax:

VASPMATE --enth (-ref) [ena enb enc …]

VASPMATE --enth (-ref) [ena enb enc …] (-path) [folder-path]

Description:

The parameters of eha, ehb, ehc are the enthalpies of the stable elemental substances corresponding to elements A, B, and C, respectively. Additionally, to avoid entering the enthalpy values individually, the elemental enthalpies can be saved in an "enthalpy.table" file, with the content formatted as [element enthalpy], for example, "B -6.70392". The -t flag can then be used to load these enthalpy values. VASPMATE will prioritize to use the values provided in the "enthalpy.table" file. VASPMATE calculates the formation enthalpy of the corresponding compound by reading the composition from POSCAR/CONTCAR and the energy from OUTCAR in the current directory. The flag -path is used to specify the folder path to retrieve the calculated enthalpies of all structures. By default, it automatically extracts the enthalpies from all folders with formats such as 0001, 0002, etc., within the folder of the provided path (the default is current folder). This model output the final results to *ENTH_INFO.dat* and *ENTH_INFO*.csv file in default.

Examples:

VASPMATE --enth (using enthalpy.table)

VASPMATE --enth -ref -6.70392 -4.94403

VASPMATE --enth -ref -6.70392 -4.94403 -path .

VASPMATE --enth -path ./BN

VASPMATE --enth (using enthalpy.table)

3.14 Elastic Properties

Elastic properties are fundamental and important quantities to evaluate the reversible response of a material to external loadings. VASPMATE applies a strain matrix to the unit cell, and then the strain energy density and stress can be obtained after a batch of VASP calculations. Based on the generated energy/stress-strain relationships after VASP calculations, it can further obtain the elastic properties of the material, including the stiffness matrix, compliance matrix, Young's modulus, bulk modulus, shear modulus, as well as Poisson's ratio, Pugh's ratio, and Cauchy stress. VASPMATE provides the following three methods for calculating elastic properties: (1) "Stress-Strain" method (AELAS), (2) "Energy-Strain" method (AELAE), and (3) Solid solution projection method (AELAE ss). The calculation results are stored in the *ELAS_INFO.dat* file.

Mode: **Stress-Strain**

Syntax:

```
VASPMATE --elas -generate/-g (strain)
VASPMATE --elas -derive/-d
```

Description:

The -generate/-derive flag is used to do pre-processing for VASP elastic property calculations and the post-processing for the derivation of elastic properties. The parameter (strain) represents the strain magnitude to be applied to the unit cell, which is optional to fill in; if not filled, the default strain steps will be applied: {-0.005, -0.003, -0.001, 0.000, 0.001, 0.003, 0.005}. When VASPMATE performs elastic property calculations, it first standardizes the unit cell based on its space group before proceeding with the next instruction. After the calculations are complete, VASPMATE parses the output files OUTCAR in each folder to obtain the "stress-strain" relationship under different strains. It then fits the material's stiffness matrix using linear least squares method, and subsequently derives other mechanical quantities such as the compliance matrix based on the stiffness matrix. When using the -derive option flag, VASPMATE will output the final results to *ELAS_INFO.dat*.

Examples:

VASPMATE --elas -g

VASPMATE --elas -g -0.00300 -0.00100 0.00100 0.00300

VASPMATE --elas -d

Mode: **Energy-Strain**

Syntax:

VASPMATE --elae -generate/-g (strain)

VASPMATE --elae -derive/-d

Description:

Similar to the "stress-strain" method, in the "energy-strain" method, the -generate/-derive flag is used to do pre-processing for VASP elastic property calculations and the post-processing for the derivation of elastic properties. The parameter (strain) represents the strain magnitude to be applied to the unit cell, is optional to fill in. If not filled, the default strain steps will be applied: {-0.018, -0.015, -0.012, -0.009, -0.006, -0.003, 0.000, 0.003, 0.006, 0.009, 0.012, 0.015, 0.018}. When VASPMATE performs elastic property calculations, it first standardizes the unit cell based on its space group before proceeding with the next instruction. After the calculations are complete, VASPMATE parses the output file OSZICAR in each folder to obtain the "energy-strain" relationship under different strains. It then fits the material's stiffness matrix using quadratic least squares method, and subsequently derives other mechanical quantities such as the compliance matrix based on the stiffness matrix. When using the -derive option flag, VASPMATE will output the final results to *ELAS_INFO.dat*.

Examples:

VASPMATE --elae -g

VASPMATE --elae -g -0.00300 -0.00100 0.00100 0.00300

VASPMATE --elae -d

Mode: **Solid Solution**

Syntax:

VASPMATE --elae ss -g/generate [mode] (strain)

VASPMATE --elae ss -d/derive [mode]

Description:

VASPMATE performs the symmetry analysis before applying corresponding strain matrices based on different symmetries during regular calculations. Directly calculating the elastic properties of solid solution materials can lead to errors in symmetry analysis. Theoretically, it is necessary to perform DFT calculations based on the lowest symmetry, which is the triclinic crystal system, involving the calculation of all 21 independent elastic tensors, which is computationally intensive. Therefore, based on SBP (symmetry-based projection) technology, VASPMATE provides users with a simplified scheme for calculating the elastic properties of solid solution materials. To calculate the elastic constants of solid solutions, the ss option of the energy-strain method needs to be used to enable the solid solution projection mode. The (mode) parameter corresponds to abbreviated commands for projecting the calculated unit cell onto monoclinic, orthorhombic, tetragonal, trigonal, hexagonal, and cubic structures, as follows:

1	Trigonal system
2	Monoclinic system
3	Orthorhombic system
4	Tetragonal system
5	Hexagonal system
6	Cubic system

Examples:

VASPMATE --elae ss -g 6

VASPMATE --elae ss -g 6 -0.00300 -0.00100 0.00100 0.00300

VASPMATE --elae ss -d 6

3.15 Optical Properties

Syntax:

VASPMATE --opti -lop -3d/-2d

Description:

For the option flag -lop, VASPMATE parses the VASP output file vasprun.xml to obtain the real and imaginary parts of the phonon frequency required to calculate the linear optical spectrum (LOP) of the material. These values are saved in the REAL.IN and IMAG.IN files, respectively. The frequency-dependent linear optical spectra include the refractive index, the extinction coefficient, the absorption coefficient, the energy loss function, the reflectivity, etc. Note: For two-dimensional materials, the calculation needs to ensure that the vacuum layer is perpendicular to the z-axis.

Examples:

VASPMATE --opti -lop -3d

VASPMATE --opti -lop -2d

3.16 AIMD Simulation

Syntax:

VASPMATE --mds -nve -ts [steps]

VASPMATE --mds -nvt -T [temperature] -ts [steps]

VASPMATE --mds -npt -T [temperature] -p [pressure] -ts [steps]

VASPMATE --mds -es/-ms

VASPMATE --mds -ns [time step list]

Description:

For the option flag -nve/nvt/npt, VASPMATE will generate the INCAR file for performing MD simulation according to the specified ensembles, i.e. NVE, NVT and NPT. The parameters of -T and -p are used to set the temperature and pressure. The parameter -ts is used to set the total number of time steps. The flag -es/ms is used to parse the VASP output file OSZICAR to obtain the variation of total energy or total magnetic moment with the time steps, which will be saved in ENG_MAG.csv or MAG_INFO.csv. The parameter -ns is used to parse the VASP output file XDATCAR to obtain the configuration corresponds to the specified time steps, and save to the files POSTEP_xxx.vasp.

Examples:

VASPMATE --mds -nvt -T 300 -ts 1000

VASPMATE --mds -npt -T 300 -p 0 -ts 1000

VASPMATE --mds -es

VASPMATE --mds -ns 1 3 4-10

3.17 Random and Evolutionary Structure Generation

To facilitate users in high-throughput structure screening, VASPMATE provides two pathways in generating structures through stochastics and evolutionary algorithms.

3.17.1 Stochastics Algorithm

Mode: **sto**

Syntax:

VASPMATE --sto -atom [natom1 natom2] -elem [element1 element2] -num [file_number] -fav [formula_atom volume] -min [switch (maxstep scale)]

Description:

This functionality is used to randomly generate structures in the POSCAR format using a stochastic algorithm. Parameters [natom1 natom2] correspond to the numbers of respective atoms to be generated. If you need to generate several types of atoms, please separate them with spaces. Parameters [element1 element2] are the names of the elements to be generated. Please ensure that the element names correspond to the atom numbers, and separate different element names with spaces. The parameter file_number represents the number of structure files you wish to generate. The -min flag is used to set whether to perform energy minimization on the generated structures to produce more realistic new structures. If you want to use the energy minimization algorithm, set switch to "0" and enter Maxstep and Scale. If not, set it to "1". By default, This functionality outputs a list of files named POSCAR01, POSCAR02, and so on.

Examples:

VASPMATE --sto -atom 1 2 -elem B C -num 2 -fav 1 0 -min 1

VASPMATE --sto -atom 1 2 -elem B C -num 2 -fav 1 0 -min 0 1000 0.1

3.17.2 Evolutionary Algorithm

Mode: **evo**

Syntax:

VASPMATE --evo file(INPOS) -t [operation] -n [file_number] -min [switch (maxstep scale)]

VASPMATE --evo file1(INPOS1) file2(INPOS2) -t [operation] -n [file_number] -min [switch (maxstep scale)]

Description:

This functionality is used to generate structure files in POSCAR format using an evolutionary algorithm. Parameters file1 and file2 serve as the initial structure files for the structure evolutionary. The -t flag is used to set the type of evolution operation: operation= Stripple/ Permustrain/ Randmove/ Slip/ Twist/ Rotstrain/ crossover. The -n flag is used to specify the number of structures to be generated, denoted by file_number.

The -min flag determines whether to perform energy minimization on the generated structures to produce more realistic new structures. If you want to use the energy minimization algorithm, set switch to "0" and provide values for Maxstep and Scale. If not, set it to "1". By default, This functionality outputs a list of files named POSCAR01, POSCAR02, ..., POSCARn.

Examples:

VASPMATE --evo -t stripple -n 3 -min 1

VASPMATE --evo -t crossover -n 3 -min 1

VASPMATE --evo INPOS -t stripple -n 3 -min 0 1000 0.1

VASPMATE --evo INPOS1 INPOS2 -t crossover -n 3 -min 0 1000 0.1

4 Examples

4.1 Energy band structure of B_6N through PBE

（1）Bash script for energy band calculation

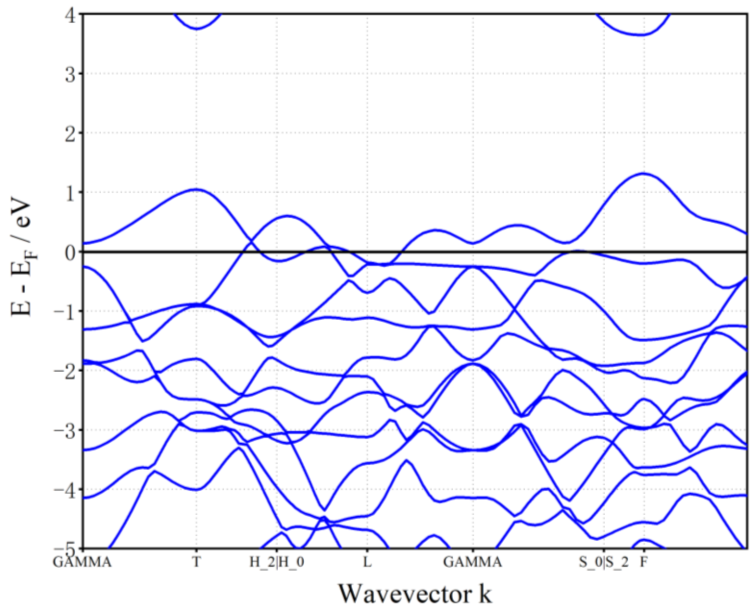
```
#! /bin/bash
VASPMATE_DIR=''
VASP_CAL=''

cp ./structure/* ./INPOS
#>>> create NEWKPATH and PRIMPOS according to POSCAR
${VASPMATE_DIR}/VASPMATE --ka3d -par 20
#>>> create POSCAR file
cp PRIMPOS POSCAR

#-------------------------- Relaxation ------------------------
#>>> create INCAR files: incar_rlx
cp POSCAR INPOS
${VASPMATE_DIR}/VASPMATE --i rlx
cp incar_rlx INCAR
#>>> Change the parameters in INCAR
#${VASPMATE_DIR}/VASPMATE --i_replace ISPIN 2
#${VASPMATE_DIR}/VASPMATE --i_replace ENCUT 400
#>>> create KPOINTS file
${VASPMATE_DIR}/VASPMATE --ka -par 4000
cp NEWKPT KPOINTS
#>>> create POSAR file
cp INPOS POSCAR
#>>> create POTCAR file according to POSCAR
```

```
${VASPMATE_DIR}/VASPMATE --pot -PBE
${VASP_CAL} >> log1.vasp
#--------------------------------------------------------------


#----------------- option(static calculation)------------------
cp CONTCAR INPOS
#>>> create INCAR files: incar_stc
${VASPMATE_DIR}/VASPMATE --i stc
cp incar_stc INCAR
${VASPMATE_DIR}/VASPMATE --irp LCHARG T
#>>> Change the parameters in INCAR
#${VASPMATE_DIR}/VASPMATE --i_replace ISPIN 2
#${VASPMATE_DIR}/VASPMATE --i_replace ICHARG 2
#>>> create KPOINTS file
${VASPMATE_DIR}/VASPMATE --ka -par 4000
cp NEWKPT KPOINTS
#>>> create POSAR file
cp INPOS POSCAR
${VASP_CAL} >> log2.vasp
#--------------------------------------------------------------


#>>> get the Fermi energy from self-consistent calculation
${VASPMATE_DIR}/VASPMATE --dos -efermi >> FERMI_LEVEL


#---------------- band structure calculation ------------------
#>>> create folder for band structure calculation
mkdir band
cd band
cp ../CONTCAR POSCAR
```

```
cp ../POTCAR ./
cp ../CHGCAR ./
cp ../FERMI_LEVEL ./
cp ../NEWKPATH ./
#>>> create KPOINTS file according to NEWKPATH
cp NEWKPATH KPOINTS
#>>> create INCAR file
${VASPMATE_DIR}/VASPMATE --i pbs
cp incar_pbs INCAR
${VASP_CAL} >> log3.vasp


#---------------------- post-processing ----------------------
#>>> Get the band gap
${VASPMATE_DIR}/VASPMATE --band -bg
#>>> Choose output mode
#${VASPMATE_DIR}/VASPMATE --band -b >> vaspmate.log
#${VASPMATE_DIR}/VASPMATE --band -a
#${VASPMATE_DIR}/VASPMATE --band -e
#${VASPMATE_DIR}/VASPMATE --band -s 1-4 N
#${VASPMATE_DIR}/VASPMATE --band -m 1-4 N
#${VASPMATE_DIR}/VASPMATE --band -o 1-3 s px py


#>>> Remove the redundant files
#rm POSCAR KPOINTS* INCAR* POTCAR
#rm CHG* CONTCAR DOSCAR OSZICAR OUTCAR EIGENVAL PCDAT WAVECAR
XDATCAR IBZKPT vasprun.xml
#END
```

（2）The calculated energy band of B_6N is illustrated below based on *Band.txt* and

KLABELS using origin.



4.2 Energy band structure of B_6N through HSE

（1）Bash script for energy band calculation

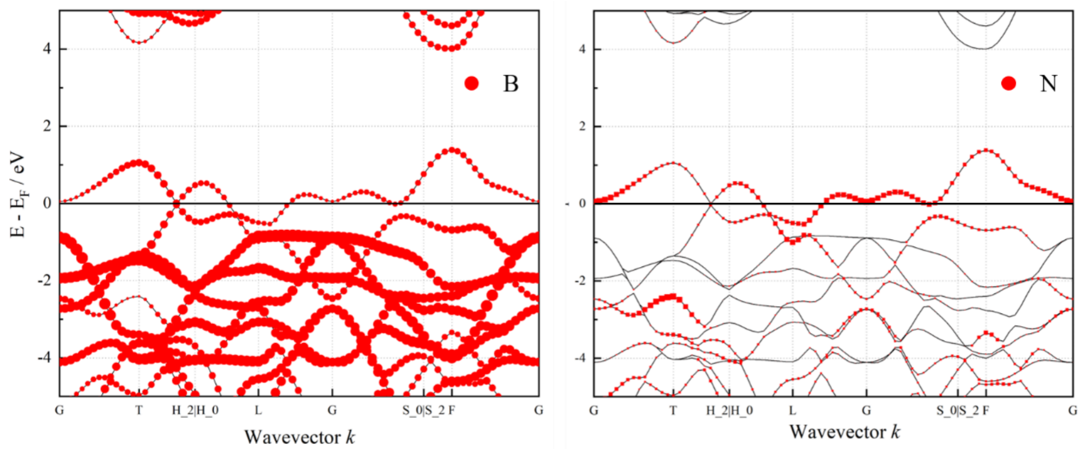
```
#! /bin/bash
VASPMATE_DIR=''
VASP_CAL=''


cp ./structure/* ./INPOS


#>>> create NEWKPATH and PRIMPOS according to POSCAR
${VASPMATE_DIR}/VASPMATE --ka3d -par 20
#>>> create POSCAR file
cp PRIMPOS POSCAR
cp POSCAR INPOS


#------------------------- Relaxation ------------------------
#>>> create INCAR files: incar_rlx
```

```
${VASPMATE_DIR}/VASPMATE --i rlx
cp incar_rlx INCAR
#>>> Change the parameters in INCAR
#${VASPMATE_DIR}/VASPMATE --i_replace ISPIN 2
#${VASPMATE_DIR}/VASPMATE --i_replace ENCUT 400
#>>> create KPOINTS file
${VASPMATE_DIR}/VASPMATE --ka -par 4000
cp NEWKPT KPOINTS
#>>> create POSAR file
cp INPOS POSCAR
#>>> create POTCAR file according to POSCAR
${VASPMATE_DIR}/VASPMATE --pot -PBE
${VASP_CAL} >> log1.vasp
#---------------------------------------------------------------


#------------------ option(static calculation)-----------------
cp CONTCAR INPOS
#>>> create INCAR files: incar_stc
${VASPMATE_DIR}/VASPMATE --i stc
cp incar_stc INCAR
${VASPMATE_DIR}/VASPMATE --irp LCHARG T
#${VASPMATE_DIR}/VASPMATE --i_replace ISPIN 2
#${VASPMATE_DIR}/VASPMATE --i_replace ICHARG 2
#>>> create KPOINTS file
${VASPMATE_DIR}/VASPMATE --ka -par 4000
cp NEWKPT KPOINTS
#>>> create POSCAR file
cp INPOS POSCAR
${VASP_CAL} >> log2.vasp
```

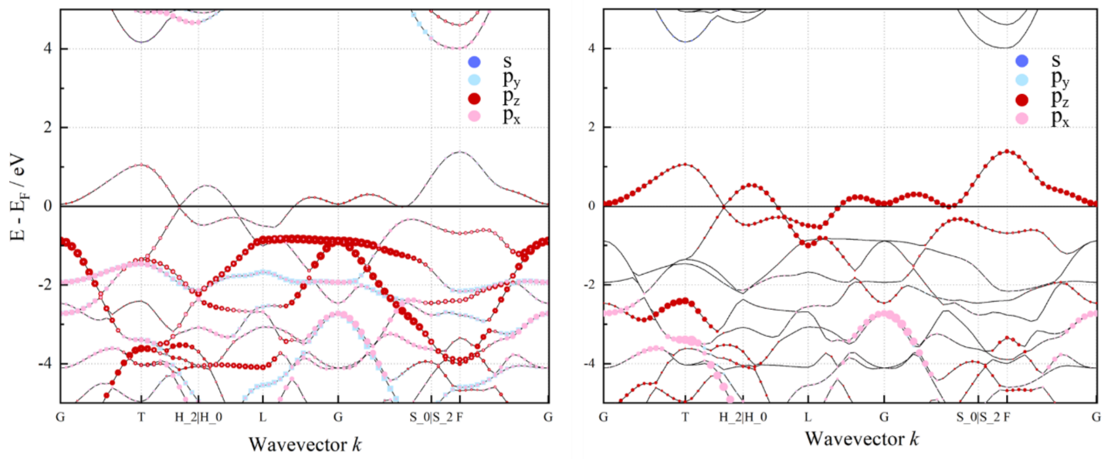
```
#---------------------------------------------------------------


#>>> get the Fermi energy from the self-consistent calculation
${VASPMATE_DIR}/VASPMATE --dos -efermi >> FERMI_LEVEL


#--------------- hse band structure calculation ---------------
#>>> create folder for band structure calculation
mkdir band
cd band
cp ../CONTCAR POSCAR
cp ../POTCAR ./
cp ../CHGCAR ./
cp ../FERMI_LEVEL ./
cp ../NEWKPATH ./
#>>> create KPOINTS file according to NEWKPATH
${VASPMATE_DIR}/VASPMATE --kahse 8000 0.05 G
cp NEWKPT KPOINTS
#>>> create INCAR file
${VASPMATE_DIR}/VASPMATE --i pbs hse
cp incar_pbs_hse INCAR
${VASP_CAL} >> log3.vasp


#---------------------- post-processing ----------------------
#>>> get the band gap
${VASPMATE_DIR}/VASPMATE --band -bg
#>>>Choose output mode
#${VASPMATE_DIR}/VASPMATE --band -hb >> vaspmate.log
#${VASPMATE_DIR}/VASPMATE --band -ha
#${VASPMATE_DIR}/VASPMATE --band -he
```

```
#${VASPMATE_DIR}/VASPMATE --band -hs 1-4 N

#${VASPMATE_DIR}/VASPMATE --band -hm 1-4 N

#${VASPMATE_DIR}/VASPMATE --band -ho 1-3 s px py


#>>> Remove the redundant files

#rm POSCAR KPOINTS* INCAR* POTCAR

#rm CHG* CONTCAR DOSCAR OSZICAR OUTCAR EIGENVAL PCDAT WAVECAR
XDATCAR IBZKPT vasprun.xml

#END
```

（2）The calculated fat band structure of B_6N is illustrated below from *PBAND_B.dat* and *PBAND_N.dat,* which show clearly the contribution of different elements to the energy bands.



To be noted additionally, by selecting the contributions of different orbitals as weights for plotting, the contribution from each orbital in the energy bands can be more distinctly observed. As an illustration, the figure below shows a fat band structure based on the contributions of *s, p_y, p_z*, and p_x orbitals from the B and N elements. It can be seen that the p_z and p_x orbitals play a leading role in bonding for B_6N system.



4.3 Density of states of B_6N through PBE

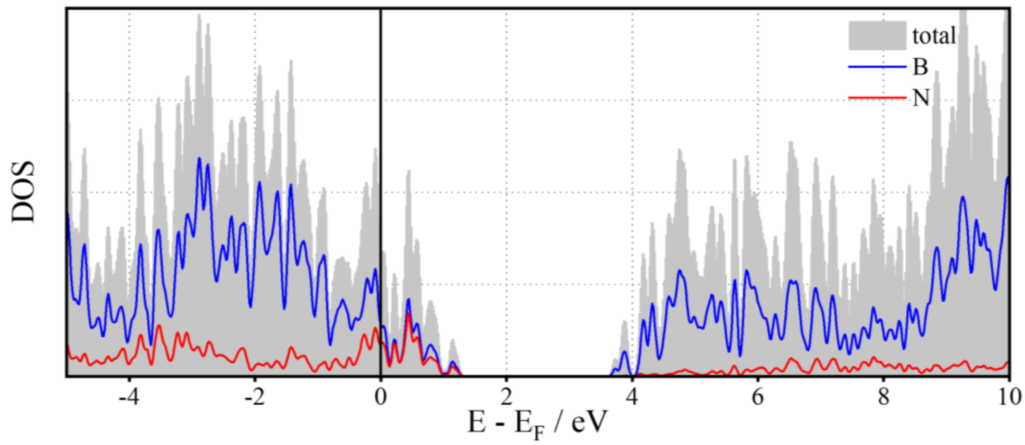
（1）Bash script for energy band calculation

```
#! /bin/bash
VASPMATE_DIR=''
VASP_CAL=''

cp ./structure/* ./INPOS

#>>> create POSCAR file
cp INPOS POSCAR

#------------------------- Relaxation -----------------------
#>>> create INCAR files: incar_rlx
${VASPMATE_DIR}/VASPMATE --i rlx
cp incar_rlx INCAR
#>>> Change the parameters in INCAR
#${VASPMATE_DIR}/VASPMATE --i_replace ISPIN 2
#${VASPMATE_DIR}/VASPMATE --i_replace ENCUT 400
#>>> create KPOINTS file
```

```
${VASPMATE_DIR}/VASPMATE --ka -par 4000
cp NEWKPT KPOINTS
#>>> create POSAR file
cp INPOS POSCAR
#>>> create POTCAR file according to POSCAR
${VASPMATE_DIR}/VASPMATE --pot -PBE
${VASP_CAL} >> log1.vasp
#---------------------------------------------------------------


#------------------ option(static calculation)-----------------
cp CONTCAR INPOS
#>>> create INCAR files: incar_stc
${VASPMATE_DIR}/VASPMATE --i stc
cp incar_stc INCAR
${VASPMATE_DIR}/VASPMATE --irp LCHARG T
#${VASPMATE_DIR}/VASPMATE --i_replace ISPIN 2
#${VASPMATE_DIR}/VASPMATE --i_replace ICHARG 2
${VASPMATE_DIR}/VASPMATE --ka -par 4000
cp NEWKPT KPOINTS
#>>> create POSCAR file
cp INPOS POSCAR
${VASP_CAL} >> log2.vasp
#---------------------------------------------------------------


#---------------------- DOS calculation -----------------------
#>>> DOS calculation
mkdir dos
cd dos
cp ../CONTCAR POSCAR
```

```
cp ../POTCAR ./

cp ../CHGCAR ./

#>>> create KPOINTS file

${VASPMATE_DIR}/VASPMATE --ka -par 4000

cp NEWKPT KPOINTS

#>>> create INCAR file

${VASPMATE_DIR}/VASPMATE --i dos

${VASPMATE_DIR}/VASPMATE --irp ICHARG 11

#${VASPMATE_DIR}/VASPMATE --i_replace LORBIT 11

#${VASPMATE_DIR}/VASPMATE --i_replace NEDOS 1000

cp incar_dos INCAR

${VASP_CAL} >> log3.vasp

#---------------------------------------------------------------


#---------------------- post-processing ----------------------

#>>>Choose output mode

${VASPMATE_DIR}/VASPMATE  --dos -t >> vaspmate.log

#${VASPMATE_DIR}/VASPMATE --dos -a

#${VASPMATE_DIR}/VASPMATE --dos -e

#${VASPMATE_DIR}/VASPMATE --dos -s 1-4 N

#${VASPMATE_DIR}/VASPMATE --dos -m 1-4 N

#${VASPMATE_DIR}/VASPMATE --dos -o 1-3 s px py

#${VASPMATE_DIR}/VASPMATE --dos -bc all


#>>> Remove the redundant files

#rm POSCAR KPOINTS* INCAR* POTCAR

#rm CHG* CONTCAR DOSCAR OSZICAR OUTCAR EIGENVAL PCDAT WAVECAR
XDATCAR IBZKPT vasprun.xml

#END
```

(2) The calculated density of states of B_6N is illustrated below with *TDOS.dat*, *PDOS_B.dat* and *PDOS_N.dat*.



4.4 Density of states of B_6N through HSE

```
#! /bin/bash
VASPMATE_DIR=''
VASP_CAL=''

cp ./structure/* ./INPOS

#>>> create POSCAR file
cp INPOS POSCAR

#-------------------------- Relaxation -----------------------
#>>> create INCAR files: incar_rlx
${VASPMATE_DIR}/VASPMATE --i rlx
cp incar_rlx INCAR
#>>> Change the parameters in INCAR
#${VASPMATE_DIR}/VASPMATE --i_replace ISPIN 2
```

```
#${VASPMATE_DIR}/VASPMATE --i_replace ENCUT 400

#>>> create KPOINTS file

${VASPMATE_DIR}/VASPMATE --ka -par 4000

cp NEWKPT KPOINTS

#>>> create POSAR file

cp INPOS POSCAR

#>>> create POTCAR file according to POSCAR

${VASPMATE_DIR}/VASPMATE --pot -PBE

${VASP_CAL} >> log1.vasp

#-------------------------------------------------------------


#------------------ option(static calculation)-----------------

cp CONTCAR INPOS

#>>> create INCAR files: incar_stc

${VASPMATE_DIR}/VASPMATE --i stc

cp incar_stc INCAR

${VASPMATE_DIR}/VASPMATE --irp LCHARG T

#${VASPMATE_DIR}/VASPMATE --i_replace ISPIN 2

#${VASPMATE_DIR}/VASPMATE --i_replace ICHARG 2

${VASPMATE_DIR}/VASPMATE --ka -par 4000

cp NEWKPT KPOINTS

#>>> create POSCAR file

cp INPOS POSCAR

${VASP_CAL} >> log2.vasp

#------------------------------------------------------------


#---------------------- HSE DOS calculation ------------------

#>>> create folder for DOS calculation

mkdir dos
```

```
cd dos

cp ../CONTCAR POSCAR

cp ../POTCAR ./

cp ../CHGCAR ./

#>>> create KPOINTS file

${VASPMATE_DIR}/VASPMATE --kahse -par 8000 0.05 G

cp NEWKPT KPOINTS

#>>> create INCAR file

${VASPMATE_DIR}/VASPMATE --i dos hse

${VASPMATE_DIR}/VASPMATE --irp ICHARG 11

#${VASPMATE_DIR}/VASPMATE --i_replace LORBIT 11

#${VASPMATE_DIR}/VASPMATE --i_replace NEDOS 1000

cp incar_dos_hse INCAR

${VASP_CAL} >> log3.vasp

#-------------------------------------------------------------


#---------------------- post-processing ----------------------

${VASPMATE_DIR}/VASPMATE  --dos -t >> vaspmate.log

#${VASPMATE_DIR}/VASPMATE --dos -a

#${VASPMATE_DIR}/VASPMATE --dos -e

#${VASPMATE_DIR}/VASPMATE --dos -s 1-4 N

#${VASPMATE_DIR}/VASPMATE --dos -m 1-4 N

#${VASPMATE_DIR}/VASPMATE --dos -o 1-3 s px py

#${VASPMATE_DIR}/VASPMATE --dos -bc all


#>>> Remove the redundant files

#rm POSCAR KPOINTS* INCAR* POTCAR

#rm CHG* CONTCAR DOSCAR OSZICAR OUTCAR EIGENVAL PCDAT WAVECAR
XDATCAR IBZKPT vasprun.xml
```

```
#END
```

4.5 Energy bands and density of states of Fe with spin polarization

（1）Bash script for energy band calculation

```
#! /bin/bash
VASPMATE_DIR=''
VASP_CAL=''

cp ./structure/* ./INPOS

#>>> create NEWKPATH and PRIMPOS according to POSCAR
${VASPMATE_DIR}/VASPMATE --kpt3d 20
#>>> create POSCAR file
cp PRIMPOS POSCAR
cp POSCAR INPOS

#-------------------------- Relaxation -----------------------
#>>> create INCAR files: incar_rlx
${VASPMATE_DIR}/VASPMATE --i rlx
cp incar_rlx INCAR
#>>> Change the parameters in INCAR
${VASPMATE_DIR}/VASPMATE --irp ISPIN 2
#${VASPMATE_DIR}/VASPMATE --i_replace ENCUT 400
#>>> create KPOINTS file
${VASPMATE_DIR}/VASPMATE --ka -par 4000
cp NEWKPT KPOINTS
#>>> create POSAR file
```

```
cp INPOS POSCAR
#>>> create POTCAR file
${VASPMATE_DIR}/VASPMATE --pot -PBE
${VASP_CAL} >> log1.vasp
#---------------------------------------------------------------


#------------------ option(static calculation)-----------------
cp CONTCAR INPOS
#>>> create INCAR files: incar_stc
${VASPMATE_DIR}/VASPMATE --i stc
cp incar_stc INCAR
${VASPMATE_DIR}/VASPMATE --irp LCHARG T
${VASPMATE_DIR}/VASPMATE --irp ISPIN 2
#${VASPMATE_DIR}/VASPMATE --i_replace ICHARG 2
#>>> create KPOINTS file
${VASPMATE_DIR}/VASPMATE --ka -par 4000
cp NEWKPT KPOINTS
#>>> create POSCAR file
cp INPOS POSCAR
${VASP_CAL} >> log2.vasp
#---------------------------------------------------------------


#>>> get the Fermi energy from the self-consistent calculation
${VASPMATE_DIR}/VASPMATE --dos -efermi >> FERMI_LEVEL


#----------------- band structure calculation -----------------
#>>> create folder for band structure calculation
mkdir band
cd band
```

```
cp ../CONTCAR POSCAR
cp ../POTCAR ./
cp ../CHGCAR ./
cp ../NEWKPATH ./
cp ../FERMI_LEVEL ./
#>>> create KPOINTS file according to NEWKPATH
cp NEWKPATH KPOINTS
#>>> create INCAR file
${VASPMATE_DIR}/VASPMATE --i pbs
${VASPMATE_DIR}/VASPMATE --irp ISPIN 2

cp incar_pbs INCAR
${VASP_CAL} >> log3.vasp

#---------------------- post-processing ----------------------
#>>> get the band gap
${VASPMATE_DIR}/VASPMATE --band -bg
#>>>Choose output mode
#${VASPMATE_DIR}/VASPMATE --band -b >> vaspmate.log
#${VASPMATE_DIR}/VASPMATE --band -a
#${VASPMATE_DIR}/VASPMATE --band -e
#${VASPMATE_DIR}/VASPMATE --band -s 1-4 N
#${VASPMATE_DIR}/VASPMATE --band -m 1-4 N
#${VASPMATE_DIR}/VASPMATE --band -o 1-3 s px py

#>>> Remove the redundant files
#rm POSCAR KPOINTS* INCAR* POTCAR
#rm CHG* CONTCAR DOSCAR OSZICAR OUTCAR EIGENVAL PCDAT WAVECAR
XDATCAR IBZKPT vasprun.xml
```

```
#END
```

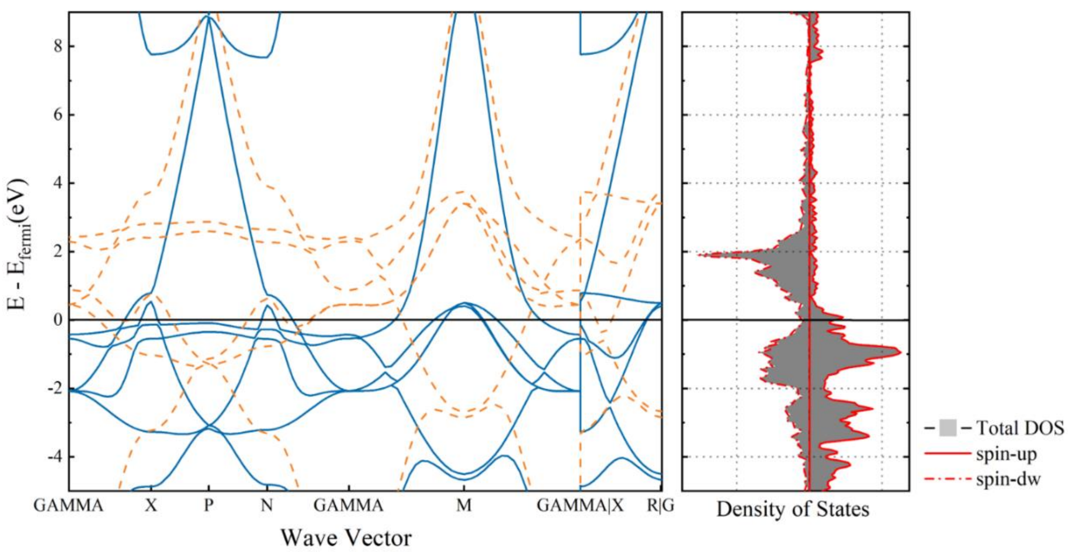
(2) Bash script for density of states calculation

```
#! /bin/bash
VASPMATE_DIR=''
VASP_CAL=''

cp ./structure/* ./INPOS

#>>> create POSCAR file
cp INPOS POSCAR

#-------------------------- Relaxation -----------------------
#>>> create INCAR files: incar_rlx
${VASPMATE_DIR}/VASPMATE --i rlx
cp incar_rlx INCAR
#>>> Change the parameters in INCAR
${VASPMATE_DIR}/VASPMATE --irp ISPIN 2
#${VASPMATE_DIR}/VASPMATE --i_replace ENCUT 400
#>>> create KPOINTS file
${VASPMATE_DIR}/VASPMATE --ka -par 4000
cp NEWKPT KPOINTS
#>>> create POSAR file
cp INPOS POSCAR
#>>> create POTCAR file according to POSCAR
${VASPMATE_DIR}/VASPMATE --pot -PBE
${VASP_CAL} >> log1.vasp
#-------------------------------------------------------------
```

```
#------------------option(static calculation)-----------------
cp CONTCAR INPOS
#>>> create INCAR files: incar_stc
${VASPMATE_DIR}/VASPMATE --i stc
cp incar_stc INCAR
${VASPMATE_DIR}/VASPMATE --irp LCHARG T
${VASPMATE_DIR}/VASPMATE --irp ISPIN 2
#${VASPMATE_DIR}/VASPMATE --i_replace ICHARG 2
#>>> create KPOINTS file
${VASPMATE_DIR}/VASPMATE --ka -par 4000
cp NEWKPT KPOINTS
#>>> create POSCAR file
cp INPOS POSCAR
${VASP_CAL} >> log2.vasp
#---------------------------------------------------------------


#---------------------- dos calculation -----------------------
#>>> create folder for DOS calculation
mkdir dos
cd dos
cp ../CONTCAR POSCAR
cp ../POTCAR ./
cp ../CHGCAR ./
#>>> create KPOINTS file
${VASPMATE_DIR}/VASPMATE --ka -par 4000
cp NEWKPT KPOINTS
#>>> create INCAR file
${VASPMATE_DIR}/VASPMATE --i dos
${VASPMATE_DIR}/VASPMATE --irp ICHARG 11
```

```
${VASPMATE_DIR}/VASPMATE --irp ISPIN 2
#${VASPMATE_DIR}/VASPMATE --i_replace LORBIT 11
#${VASPMATE_DIR}/VASPMATE --i_replace NEDOS 1000
cp incar_dos INCAR
${VASP_CAL} >> log3.vasp
#-------------------------------------------------------------


#---------------------- post-processing -----------------------
#>>>Choose output mode
${VASPMATE_DIR}/VASPMATE  --dos -t >> vaspmate.log
#${VASPMATE_DIR}/VASPMATE --dos -a
#${VASPMATE_DIR}/VASPMATE --dos -e
#${VASPMATE_DIR}/VASPMATE --dos -s 1-4 N
#${VASPMATE_DIR}/VASPMATE --dos -m 1-4 N
#${VASPMATE_DIR}/VASPMATE --dos -o 1-3 s px py
#${VASPMATE_DIR}/VASPMATE --dos -bc all

#>>> Remove the redundant files
#rm POSCAR KPOINTS* INCAR* POTCAR
#rm CHG* CONTCAR DOSCAR OSZICAR OUTCAR EIGENVAL PCDAT WAVECAR
XDATCAR IBZKPT vasprun.xml
#END
```

（3）The calculated band structure and density of states of Fe with spin polarization is illustrated below.


E - E$_{fermi}$(eV)
8
6
4
2
0
-2
-4
GAMMA
X
P
N
GAMMA
M
GAMMA|X
R|G
Wave Vector
Density of States
Total DOS
spin-up
spin-dw

5 High-throughput program

5.1 ABAND and AEDOS

VASPMATE has added high-throughput calculation modules for energy bands and density of states to the integrated platform SPaMD and on-line open-shared platform MICID (www.micid.net) developed by our group. We have designed high-throughput first-principles calculation GUI programs, ABAND and AEDOS, which transform the tedious command lines and script formats into a friendly GUI interactive interface, greatly improving the calculation efficiency of energy bands and density of states. For a detailed introduction to this functionality, please refer to the instructions for AEDOS and ABAND.

5.2 AELAS and AELAE

VASPMATE has incorporated a high-throughput elastic property calculation module into the integrated platform SPaMD and on-line open-shared platform MICID (www.micid.net) developed by our group. We have designed high-throughput first-principles calculation GUI programs, AELAS and AELAE, converting complex command lines and script formats into a user-friendly GUI interactive interface, significantly enhancing the calculation efficiency of elastic properties. For a comprehensive introduction to this functionality, please refer to the manuals for AELAS and AELAE.

5.3 AHULL

VASPMATE has integrated a high-throughput thermodynamic convex hull diagram calculation module into the in-house developed integrated platform SPaMD and on-line open-shared platform MICID (www.micid.net). We have designed a high-throughput first-principles calculation GUI program called AHULL, which transforms complex command lines and script formats into a user-friendly GUI interactive interface. This significantly improves the calculation efficiency of magnetic and optical properties. For a detailed introduction to this functionality, please refer to the AHULL manual.

References

[1] J. Hafner. Ab-initio simulations of materials using VASP: Density-functional theory and beyond. Journal of computational chemistry 29, 2044-2078 (2008).

[2] V. Wang, N. Xu, J.-C. Liu, G. Tang, W.-T. Geng, VASPKIT: A User-Friendly Interface Facilitating High-Throughput Computing and Analysis Using VASP Code, Computer Physics Communications 267, 108033 (2021).

[3] Z. R. Liu , B. N. Yao , R. F. Zhang. SPaMD studio: An integrated platform for atomistic modeling, simulation, analysis, and visualization. Computational Materials Science 210, 111027 (2022).

[4] S. H. Zhang, Z. H. Fu, and R. F. Zhang. ADAIS: Automatic Derivation for Anisotropic Ideal Strengths by means of high throughput first principles computations, Computer Physics Communications 238, 244 (2019).

[5] S. H. Zhang and R. F. Zhang. AELAS: Automatic elastic property derivations via high throughput first-principles computation, Computer Physics Communications 220, 403 (2017).

[6] Y. Hinuma, G. Pizzi, Y. Kumagai, F. Oba, I. Tanaka, Band structure diagram paths based on crystallography, Computational Materials Science 128, 140 (2017).

[7] Kokalj A. XCrySDen—a new program for displaying crystalline structures and electron densities[J]. Journal of Molecular Graphics and Modelling 17, 176-179 (1999).

[8] Kawamura M. FermiSurfer: Fermi-surface viewer providing multiple representation schemes. Computer Physics Communications 239, 197-203 (2019).

[9] M.J. Frisch, G.W. Trucks, H.B. Schlegel, G.E. Scuseria, M.A. Robb, J.R. Cheeseman, G. Scalmani, V. Barone, G.A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A.V. Marenich, J. Bloino, B.G. Janesko, R. Gomperts, B. Mennucci, H.P. Hratchian, J.V. Ortiz, A.F. Izmaylov, J.L. Sonnenberg, Williams, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V.G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J.A.

Montgomery Jr., J.E. Peralta, F. Ogliaro, M.J. Bearpark, J.J. Heyd, E.N. Brothers, K.N. Kudin, V.N. Staroverov, T.A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A.P. Rendell, J.C. Burant, S.S. Iyengar, J. Tomasi, M. Cossi, J.M. Millam, M. Klene, C. Adamo, R. Cammi, J.W. Ochterski, R.L. Martin, K. Morokuma, O. Farkas, J.B. Foresman, D.J. Fox, Gaussian 16 A.03. Wallingford, CT, (2016).

[10] Tian Lu, Qinxue Chen, Shermo: A general code for calculating molecular thermodynamic properties, Computational and Theoretical Chemistry 1200, 113249 (2021)

[11] R.F. Ribeiro, A.V. Marenich, C.J. Cramer, D.G. Truhlar Use of Solution-Phase Vibrational Frequencies in Continuum Models for the Free Energy of Solvation Journal of Physical Chemistry B 115, 14556-14562 (2011).

[12] A.V. Marenich, W. Ding, C.J. Cramer, D.G. Truhlar Resolution of a Challenge for Solvation Modeling: Calculation of Dicarboxylic Acid Dissociation Constants Using Mixed Discrete-Continuum Solvation Models

[13] L. Xu, M.L. Coote. Methods To Improve the Calculations of Solvation Model Density Solvation Free Energies and Associated Aqueous pKa Values: Comparison between Choosing an Optimal Theoretical Level, Solute Cavity Scaling, and Using Explicit Solvent Molecules Journal of Physical Chemistry A 123, 7430-7438 (2019).

[14] A. Kormanyos, G. Burkard, M. Gmitra, J. Fabian, V. Zolyomi, N. D. Drummond and V. Falḱo k.p theory for two-dimensional transition metal dichalcogenide semiconductors 2D Materials 2, 022001 (2015).