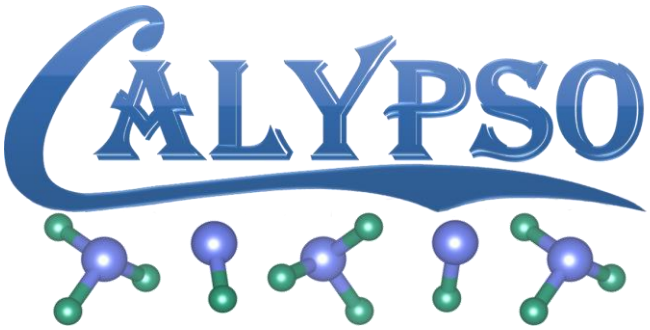


User's Guide

CALYPSO version 5.0

October 20, 2016

Description

The CALYPSO user's guide describes how to run and use various features of the structural prediction program CALYPSO. This guide shows the capabilities of the program, how to use these capabilities, the necessary input files and formats, and how to run the program both on uniprocessor machines and in parallel.

http://www.calypso.cn

CALYPSO License

Structural Prediction software, CALYPSO, is available free of charge for non-commercial use by individuals, academic or research institutions, upon completion and submission of the online registration form available from the CALYPSO web site *http://www.calypso.cn/getting-calypso/*. Commercial use of the CALYPSO software requires A CALYPSO COMMERCIAL LICENSE. Commercial use includes: (1) integration of all or part of the Software into a product for sale, lease or license by or on behalf of Licensee to third parties, or (2) distribution of the Software to third parties that need it to commercialize product sold or licensed by or on behalf of Licensee. These requests can be directed to Prof. Yanming Ma by email: mym@calypso.cn.

Registration Requirements

In completing the online registration form, individuals may register in their own name or with their institutional or corporate affiliations. Registration information must include name, title, e-mail and mailing address of a person. Please sign the CALYPSO license and send the scanned copy to calypso@calypso.cn. The CALYPSO code will be sent.

Contact Information

The best contact way is to send correspondence by email to calypso@calypso.cn or to: Professor Yanming Ma (mym@calypso.cn) or Dr. Yanchao Wang (wyc@calypso.cn)

Mailing address:
Prof. Yanming Ma
State Key Lab of Superhard Materials, Jilin University
130012, Changchun, P. R. China
Phone: +86-431-85168276
FAX: +86-431-85168276
Personal webpage: http://mym.calypso.cn

Contents

1. Introduction

CALYPSO is a swarm-intelligence based structure prediction method and its same-name computer software. The approach requires only chemical compositions of given compounds to predict stable or metastable structures at certain external conditions (e.g., pressure). The method can also be used to inversely design multi-functional materials (e.g., superhard materials, electrides, optical materials, etc). The CALYPSO package is protected by the Copyright Protection Center of China with the Registration No. 2010SR028200 and Classification No. 61000-7500.

1.1 Meaning of CALYPSO

CALYPSO is a short name of "**C**rystal structure **A**na**LY**sis by **P**article **S**warm **O**ptimization". It was originally designed to predict 3-dimensional (3D) "crystal structures". Now, CALYPSO has a more generalized meaning of "structure" prediction, able to deal with structures ranging from 0D to 1D, 2D, and 3D.

"CALYPSO" (with all capitalized letters) is the only name in the field of structure prediction. But the word "Calypso" has diverse meanings. Calypso is the name of one of the Nereids (sea nymphs) in Greek mythology. Calypso also refers to companies, music, places, etc. Have a look at Wikipedia (http://en.wikipedia.org/wiki/Calypso).

CALYPSO structure prediction software takes the advantage of structure evolution via PSO algorithm, one of swarm intelligence schemes. However, many other efficient structure-dealing techniques (e.g., symmetry constraints, bond characterization matrix, introduction of random structures per generation, etc.) were also implemented in CALYPSO. We found that all these techniques implemented are equivalently important for the structure searching efficiency. It is therefore more appropriate to name the developed structure prediction method as a "CALYPSO" method.

1.2 Why PSO?

As an unbiased global optimization method, PSO is inspired by the choreography of a bird flock and can be viewed as a distributed behavior algorithm that performs multidimensional search (see, e.g., Kennedy & Eberhart 1995). PSO is metaheuristic as it makes few or no assumptions about the solutions and can search very large spaces of candidate solutions (dubbed as particles) by moving them in the search-space based on efficient algorithms over the particle's position and velocity.

We quote from website of http://www.swarmintelligence.org.
PSO has been successfully applied in many research and application areas. It is demonstrated that PSO can get better results in a faster, cheaper way compared with other methods.

Another reason that PSO is attractive is that there are few parameters to adjust. One version,

with slight variations, works well in a wide variety of applications. PSO has been used for approaches across a wide range of applications, as well as for specific applications focused on a specific requirement."

1.3 History of PSO on Structure Prediction

Although PSO algorithm has been employed to various optimization problems, the application of PSO in structure prediction started only recently. It was attempted for isolated systems (small clusters and molecules) by Call, Zubarev & Boldyrev in 2007. However, this effort did not lead to any practical application.

The CALYPSO team independently initialized the idea of applying PSO algorithm into structure prediction in 2006 (Ma and Wang) before Call *et al*'s work and made the first application of PSO algorithm into structure prediction of extended systems (e.g., 3D crystals by Wang, Lv, Zhu & Ma in 2010, 2D layers by Luo *et al*, in 2011 and Wang et al, in 2012, 2D surface reconstruction by Lu et al, in 2014, 2D atoms adsorbed on layer materials by Gao et al., in 2015). Structure searching efficiencies of isolated systems have been substantially improved by the CALYPSO team (Lv, Wang, Zhu & Ma in 2012), where the success of this application has been backed up with the introduction of various efficient techniques (e.g., bond characterization matrix for fingerprinting structures, symmetry constraints on structure generation, etc**.**).

2. CALYPSO Program

2.1 Program Features

- Predictions of the energetically stable/metastable structures at given chemical compositions and external conditions (e.g., pressure) for 0D nanoparticles or clusters (section 3.3), 2D layers (section 3.2) and its atom adsorption (see, section 3.8), 2D surface reconstructions (section 3.6), and 3D crystals (section 3.1).
- Functionality-driven design of novel functional materials, e.g., superhard materials (section 3.7), electrides, and optical materials (section 3.9) with desirable hardness values, electron localization functions, and energy band gap, respectively.
- Options for the structural evolutions using global or local PSO techniques.
- Structure predictions with automatic variation of chemical compositions (section 3.5).
- Incorporation of various structure constraints, e.g., fixed rigid molecules (section 3.4), fixed cell parameters, fixed space group or fixed atomic positions (section 3.10).
- X-ray diffraction data assisted structural prediction (section 3.11).
- Prediction of transition states in solids (section 3.12).
- CALYPSO is currently interfaced with VASP, CASTEP, Quantum Espresso, GULP, SIESTA, FPLO, Gaussian, CP2K, LAMMPS, and ABACUS codes for local geometrical optimization and total-energy calculations. Its interface with other codes can also be implemented by users' request.
- It is written in Fortran 90 and memory is allocated dynamically.

2.2 Major Techniques

The success of CALYPSO method is on account of its efficient integration of several major structure dealing techniques.

1. Structural evolution through PSO algorithm. PSO is best known for its ability to overcome large barriers of energy landscapes by making use of the smart structure self-learning within swarm intelligence algorithm. Both global and local PSO algorithms have been implemented. The global PSO has the advantage of fast convergence, while local PSO is good at avoiding structure premature ready for dealing with complex systems.
2. Symmetry constraints on generation of random structures to ensure the creation of physically feasible structure, reduce searching space, and enhance the structural diversity during evolution.
3. Two structural characterization techniques for eliminating similar structures, and partitioning energy surfaces for local PSO structure searches.
   (i) bond characterization matrix technique

(ii) atom-centered symmetrical function technique

4. Introduction of random structures per generation with controllable percentage to enhance structural diversity during evolution.

5. Interface to a number of local structural optimization codes varying from highly accurate DFT methods to fast semi-empirical approaches that can deal with large systems. Local structural optimization is the most time-consuming part of CALYPSO structure prediction. This is a must process since it enables the reduction of noise of energy surfaces and the generation of physically justified structures.

CALYPSO routinely provides:

- Crystal structure prediction (section 3.1)
- 2-dimensional layer structure prediction (section 3.2)
- Clusters or nanoparticles structure prediction (section 3.3)
- Molecular crystal structure prediction (section 3.4)
- Surface reconstruction structure prediction (section 3.6)
- Inverse structural design of superhard materials (section 3.7)
- Structural design of 2D material with atoms adsorption (section 3.8)
- Inverse structural design of optical materials (section 3.9)
- X-ray diffraction data assisted structural prediction (section 3.11)
- Prediction of transition states in solids (section 3.12)

For more details on the methodologies and formalisms of CALYPSO, please read the references cited below.

References:

◆ **CALYPSO Software:**

Yanchao Wang, Jian Lv, Li Zhu, and Yanming Ma*

CALYPSO: A Method for Crystal Structure Prediction, **Comput. Phys. Commun.** 183, 2063 (2012).

◆ **Crystal Structure Prediction:**

Yanchao Wang, Jian Lv, Li Zhu and Yanming Ma*

Crystal structure prediction via particle-swarm optimization, **Phys. Rev. B** 82, 094116 (2010).

◆ **Cluster Structure Prediction:**

Jian Lv, Yanchao Wang, Li Zhu, and Yanming Ma*

Particle-Swarm Structure Prediction on Clusters, **J. Chem. Phys.** 137, 084104 (2012).

◆ **Two-Dimensional Layer Structure Prediction:**

1. Xinyu Luo, Jihui Yang, Hanyu Liu, Xiaojun Wu, Yanchao Wang, Yanming Ma, Su-Huai Wei, Xingao Gong, and Hongjun Xiang

Predicting Two-Dimensional Boron-Carbon Compounds by the global optimization method. **J.**

Am. Chem. Soc. 133, 16285(2011).

2. Yanchao Wang, Maosheng Miao, Jian Lv, Li Zhu, Ketao Yin, Hanyu Liu, and Yanming Ma* *An effective Structure Prediction Method for Layered Materials Based on 2D Particle Swarm Optimization Algorithm*, **J. Chem. Phys.** 137, 224108 (2012).

- **Inverse Design of Superhard Materials:**

Xinxin Zhang, Yanchao Wang, Jian Lv, Chunye Zhu, Qian Li, Miao Zhang, Quan Li and Yanming Ma*

First-Principles Structural Design of Superhard Materials, **J. Chem. Phys.** 138, 114101 (2013).

- **Surface Reconstruction Structure Prediction:**

Shaohua Lu, Yanchao Wang, Hanyu Liu, Maosheng Miao and Yanming Ma*

Self-assembled ultrathin nanotubes on diamond (100) surface, **Nat. Commun.** 5, 3666 (2014).

- **Structural design of 2D material with atoms adsorption:**

Bo Gao, Xuecheng Shao, Jian Lv, Yanchao Wang* and Yanming Ma*

Structure Prediction of Atoms Adsorbed on Two-Dimensional Layer Materials: Method and Applications, **J. Phys. Chem. C** 119, 20111 (2015).

2.3 Bug Report

The CALYPSO package has been thoroughly tested for numerous systems by the CALYPSO team and other users, and has been progressively improved by adding new features and eliminating bugs. We would greatly appreciate comments, suggestions and criticisms by the users of CALYPSO; in case of bug report, the users can contact the authors and send a copy of both input and output by E-mail to the Ma group (calypso@calypso.cn).

2.4 Compilation

For CALYPSO installation, basic UNIX knowledge is required. The users should be acquainted with the tar, gzip, awk, sed, and make commands of the UNIX environment.

Currently, CALYPSO runs on Mac OS X, and Linux. Please download the different version of CALYPSO according to your machine.

For Mac OS X, please download the version of CALYPSO_MAC_5.0.tar.gz

For 32-bit version of Linux, please download the version of CALYPSO_x86.tar.gz

For 64-bit version of Linux, please download the version of CALYPSO_x64.tar.gz

Installing the program is simple. Please use the following commands.

```
$ tar -zvxf Calypso_*.tar.gz
$ cd CALYPSO_*
```

The execution file (calypso.x) can be found in the **bin** fold.

2.5 Execution of CALYPSO

A fast way to test your installation of CALYPSO code can be found in the directory of "Tests". Please simply run the commands:

```
$ cd CALYPSO_*/Tests
```

```
$ cp CALYPSO_*/bin/calypso.x .
$ ./calypso.x >  caly.log &
```

Other examples are provided in the "path-to-package/Examples" directory. This directory contains basic input.dat and shell scripts for running CALYPSO. It is advisable to create independent directory for each job, making a clean and neat environment.

We describe here in details an example of cubic boron nitride "**BN**" on how to run structure prediction by CALYPSO. Here, VASP code was used for geometric structure optimization and total energy calculations. Please go to your working directory and do:

```
$ mkdir BN
$ cd BN
$ cp path-to-package/Examples/BN/* ./
```

The following necessary files should appear in the working directory:

(1) **input.dat**: The CALYPSO input file

(2) **calypso.x:** The executable file for structure prediction

(3) **submit.sh:** Executable shell scripts to submit a job for geometry optimization

(4) **INCAR_* and POTCAR:** The VASP input files

Now you are ready to run the program:

```
$ ./calypso.x > caly.log &
```

After a successful run of the program, several VASP output files and a new directory named as "**results"** will be generated in your working directory.

2.6 Basic inputs and outputs

2.6.1 Input.dat

The main input file named as **input.dat**, contains all necessary parameters for the structure prediction. The file consists of input tags that can be given in any order, or be omitted while the default values are used. Below we offer a quick view of the syntax of the tags:

- The general syntax is **"tag labels" = "value1 value2 value3 …".** For matrix input, it starts with @tag and ends with @End. Values that are not specified in the **input.dat** file are assigned as default values. Input values should be separated by space.
- The labels are case sensitive.
- All text following the # character is taken as comment.
- Logical values can be given as T (or True), or F (or False).

Below are brief descriptions on necessary input parameters.

SystemName = *string*: A string for description of the targeted system (max. 40 characters).
Defualt value: CALYPSO

NumberOfSpecies = *integer*: Number of different atomic (or chemical) species in the system. For example, it is "2" for MgB_2.
Default value: There is no default. You must provide the number.

NameOfAtoms = *string1 string2 string3 …*: Elemental symbols of each atomic species separated by space. Taking MgB_2 and $MgSiO_3$ as examples, one will write "**NameOfAtoms** = Mg B" and "**NameOfAtoms** = Mg Si O", respectively.
Default value: There is no default. You must define it.

NumberOfAtoms = *integer1 integer2 integer3 …*: Number of atoms for each chemical species per formula unit. Taking MgB_2 and $MgSiO_3$ as examples, one writes "**NumberOfAtoms** = 1 2" and "**NumberOfAtoms** = 1 1 3", respectively.
Default value: There is no default.

NumberOfFormula = *integer*1 *integer*2: Defining the number of formula units in the simulation cell. *integer1* and *integer2* are the minimal and maximal number of formula units used in the simulation cell, respectively. If *integer1* equals to *integer2*, then only single choice of formula unit will be adopted. For example, "**NumberOfFormula** = 4 4" indicates one single run of 4 formula units in your simulation, while "**NumberOfFormula** = 1 4" indicates that four separate structure prediction runs for 1, 2, 3, and 4 formula units, respectively, will be performed.

Default value: 1 4

Volume = *real*: The volume (in unit of angstrom^3) per formula unit. If you cannot provide a good estimation on the volume, please use the default value. The program will automatically generate an estimated volume by using the ionic radii of given atoms.

Default value: 0

@DistanceOfIon

$real_{11}$ $real_{12}$ $real_{13}$ …
$real_{21}$ $real_{22}$ $real_{23}$ …
….

@End

Minimal interatomic distances (in unit of angstrom) in a format of $\boldsymbol{n \times n}$ matrix. The rank $\boldsymbol{n}$ of the matrix is determined by the "**NumberOfSpecies".** If we take MgB_2 as an example with "**NumberOfSpecies** = 2", the $\mathbf{2 \times 2}$ matrix is defined as

@DistanceOfIon

d_{11} d_{12}

d_{21} d_{22}

@End

Here, d_{11} and d_{12} define the minimal Mg-Mg and Mg-B distances, respectively, while d_{21} and d_{22} define the minimal B-Mg and B-B distances, respectively.

Default value: 0.7 Å

Ialgo =*integer*: It defines which PSO algorithm should be adopted in the simulation.

1: global PSO algorithm

2: local PSO algorithm

3: ABC algorithm with symmetry

Default value: 2

ICode =*integer*: It defines which code should be used for local structure optimization during the structure prediction.

1: VASP

2: SIESTA

3: GULP

4: PWSCF

5: CASTEP

6: CP2K

7: Gaussian

8: DFTB+

9: LAMMPS

Default value: 1

NumberOfLocalOptim=*integer*: It defines how many times a structure should be optimized once it is generated during structure prediction. The reason why we optimize the structure by several times is because the generated structures are often far from their local minima. A single fine structure optimization typically leads to a non-converged calculation. Our tests indicated that three or four-steps optimizations of one structure are the best solution. During the course of these multiple optimizations, the optimization degree should increase gradually: coarse➔medium ➔fine. If one uses VASP for structure optimization, three or four input INCAR files should be therefore provided. The same multiple-optimization procedure applies also to the use of other optimization codes. More details can be found in examples.

Default value: 4

PsoRatio =*real*: This variable defines what percentage of the structures per generation should be produced by PSO. The rest of structures will then be randomly generated with symmetry

constraints.
Default value: 0.6

PopSize =*integer*: The population size, i.e., the total number of structures per generation. Normally, a larger population size is needed for a larger system. Very large population size should be used for simulations of automatic variation of chemical compositions.
Default value: 30

Kgrid =*real*1 *real*2: The precision of the *K*-point sampling for local geometric optimization for VASP, Quantum-Esspresso and DFTB+ codes. The Brillouin zone sampling uses a grid of spacing $2\pi\times$Kgrid $Å^{-1}$. *Real*1 controls the precision of the first two or three local optimizations, and the *real*2 with denser K-points controls the last optimization. The smaller value gives finer optimization results. Note that this setting of Kgrid is only applicable to the uses of VASP, DFTB+ and Quantum-Esspresso codes. For uses of other *ab initio* codes (e.g., CASTEP, CP2K, and Gaussian, etc.), please prepare appropriate Kgrid settings in their input files.
Default value: 0.12 0.06

Command =*string*: The command to submit structure optimization jobs in your computer.
Default value: submit.sh.

MaxStep =*integer*: The maximum number of generations to be executed for the entire structure prediction simulation. Typically, a larger number of generations are needed for a larger system.
Default value: 50

PickUp=*logical*: If this variable is set as **True**, structure prediction will start from a specified generation (see PickStep) where a previous structure prediction job was unexpectedly interrupted.
Default value: False

PickStep=*integer*: At which generation the previously interrupted calculation should be re-started.
Default value: There is no default. If PickUp=True, you must define this variable.

MaxTime=*integer*: The running wall-time limit (in unit of seconds) of one single structure optimization. If the optimization time goes beyond MaxTime, the current structure optimization will be terminated.
Default value: 7200

LMC=*logical*: This variable determines whether the Metropolis criterion should be applied during the PSO structure evolution. For cluster structure prediction, it is highly recommended to set this variable as "True".
Default value: False.

The parameters for structural prediction of two-dimensional layers ####### ################## ***Examples are given in section 3.2***

2D=logical: when this tag is set as "**True**", structure prediction of 2-dimensional layer will be performed.
Default value: False

LFilm=logical: when this tag is set as "**True**", structure prediction of 2-dimensional thin film will be performed.
Default value: False

Thickness=real: The thickness of thin film (in unit of angstrom).
Default value: There is no default. You must supply this variable.

Area=real: The area (in unit of angstrom^2) per formula unit. If you cannot provide a good estimation on the area, please use the default value. The program will automatically generate an estimated area by using the ionic radii of given atoms.
Default value: 0.0

MultiLayer=integer: It defines the number of layers you would like to design.
Default value: 1

DeltaZ=real: The distortion value (in unit of angstrom) along the ***c*** direction, i.e., perpendicular to the ***a-b*** plane of the layer. If this parameter is set to be zero, then a strict flat layer will be designed. Otherwise, a buckled layer is designed.
Default value: 0.2

LayerGap=real: The gap between two layers, i.e., the inter-layer distance (in unit of angstrom). If this parameter is set as zero, then one single layer will be designed.
Default value: 5.0

VacuumGap=real: This variable defines the separations (in unit of angstrom) between the designed single layer (or multi-layers) and its nearest-neighboring periodic images. This value should be large enough to ensure that interactions between the designed layer and its

nearest-neighboring periodic images are negligible.

Default value:10.0

@LayerType

integer$_{11}$ *integer*$_{12}$ *integer*$_{13}$…

integer$_{21}$ *integer*$_{22}$ *integer*$_{23}$**..**

…

@End

Design of multi-layers materials with desirable compositions in a format of ***m*×*n*** matrix. The row (***m***) and column (***n***) ranks of the matrix are determined by the "**MultiLayer**" and "**NumberOfSpecies"**, respectively. For each row (i.e., each layer), the matrix column values are defined as the number of atoms for each chemical species as listed in the "**NameOfAtoms".**

We take the design of double layers of B-C-N materials as an example. Here we have "**MultiLayer = 2**", "**NumberOfSpecies=3",** and "**NameOfAtoms= B C N".** If we purposely design the materials with the first layer having 6 C and 4 N atoms and the second layer having 3 B, 4 C, and 7 N atoms. The **2×3** matrix can be written as follows.

@LayerType

0 6 4

3 4 7

@End

Default value: There is no default. You must define these numbers.

LAtom_Dis=real: The minimal interatomic distance in unit of angstrom.

Default value: 1.0

########################## ***End*** ##########################

################ ***The parameters for cluster structure prediction*** ###############

################## ***Examples are given in section 3.3*** ##################

Cluster =*logical*: when this tag is set as "**True**", a nanocluster structure prediction is performed.

Default value: False

Vacancy=*real*1 *real*2 *real*3: The isolated cluster is placed into an orthorhombic box where the periodic boundary condition is applied. This variable defines the separations (in unit of angstrom) between the studied cluster and its nearest-neighboring periodic images. It should be large enough to ensure that interactions between the studied cluster and its nearest-neighboring images are negligible.

Default value: 10.0 10.0 10.0

For cluster structure prediction, we do not recommend the use of VASP for the structural optimization for large systems since computationally VASP calculations are very expensive. **We recommend using SIESTA, CP2K, and Gaussian codes when cluster sizes are larger than 10 atoms.**

########## ***End*** ##########

The Parameters for structure prediction with rigid Molecules

########## ***Examples are given in section 3.4*** ##########

MOL=*logical*: when this tag is set as "True", structure prediction with fixed rigid molecules will be performed. This is a useful technique for structure design of molecular systems, especially when rigid molecules are known constituent of certain structures. Note that a file named as MOL is needed to define the rigid molecule. Please refer to examples (section 3.4) for settings of MOL.

Default value: False

NumberOfTypeMolecule=*integer*: The number of different types of molecules in the simulation cell.

Default value: There is no default. You must supply this variable.

NumberOfMolecule=*integer1 integer2…*: The number of molecules for different molecular species.

Default value: There is no default. You must supply this variable.

DistOfMol= *real*: The minimum distance (in unit of angstrom) between two rigid molecules.

Default value: 1.5

########## ***End*** ##########

The parameters for structure prediction with variational stoichiometry

########## ***Examples are given in section 3.5*** ##########

VSC=*logical*: If this tag is set as "**True**", structure prediction of automatic variation of chemical compositions will be performed. This technique is designed to explore all possible stoichiometries for given binary systems (e.g., A_xB_y system) at once. However, one has to take his/her own risk for the use of this technique. The search space has been significantly enlarged due to the existence of large number of possible stoichiometries. We highly recommend separate simulations with fixed stoichiometries for confirmation of their results.

Default value: False

MaxNumAtom=*integer*: The maximal number of atoms allowed in the simulation cell.

Default value: 20

@CtrlRange

integer$_{11}$ integer$_{12}$

integer$_{21}$ integer$_{22}$

@End

Defining the compositional range for each type of constituent atom in the binary A_xB_y system. For atom A, *x* varies from *integer*$_{11}$ to *integer*$_{12}$, while for atom B, *y* varies from *integer*$_{21}$ *to integer*$_{22}$.

Default value:

@CtrlRange

1 6

1 6

@End

########################## ***End*** ##########################

########## ***The parameters for surface reconstruction prediction*** ##########

################### ***Examples are given in section 3.6*** #################

LSurface=***logical***: The flag specifies whether surface reconstruction prediction will be performed.

Default value: False

SurfaceThickness=***real***: This variable (in unit of angstrom) specifies the thickness of surface reconstruction, and it should be set as a value slightly larger than the double distance of two adjacent atomic layers in bulk materials.

Default value: 3.0

Substrate=***string***: This variable allows the users to define their own substrates. "***string***" is the name of the user defined substrate file, which contains the structure data of the substrate (i.e., lattice parameters and atomic positions). Here, both POSCAR and ***cif*** file formats are supported. For POSCAR file format, users can use '*Selective dynamics'* tag to define the relaxable atomic layers, otherwise all the atoms in the substrate will remain un-relaxed during surface reconstruction calculations. For ***cif*** file format, users need to insert '_selective*'* tag beneath '_atom_site_fract_z', and then include 'T' after atomic positions to define the relaxable atomic layers, while 'F' for the unrelaxable bulk. Please refer to examples (section 3.6) for more details. By defining this variable to 'Automatic' or 'Auto', the substrate will be generated automatically from its bulk structure. See below *controlling parameters* for generation of *substrate* based on its bulk structure.

Default value: SUBSTRATE.surf.

@SurfaceAtoms

***string*$_{11}$ *integer*$_{12}$**

***string*$_{21}$ *integer*$_{22}$**

…

@End

Defining the reconstructed surface region with desirable compositions in a format of ***m*×*2*** matrix. The row (***m***) rank of the matrix is determined by **NumberOfSpecies**. For each row, the matrix contains two columns. The first column (***string***) is the elemental symbol of each chemical species, followed by the number of atoms for such a chemical species (***integer***). Please refer to the Examples for more detailed setting of these parameters.

Default value: There is no default. You must supply this variable.

SpaceSaving=***logical***: If this parameter is set as "**True**", the output files for structure relaxation will be deleted. Since the output files for structure relaxation are very large, it is desirable to save the storage space in hard driver by deleting these redundant files. However, one might set this parameter as "*False*" only in case of debugging the results.

Default value: True

ECR=***logical***: If this parameter is set as "True", only those initial structures that meet electron counting rule are accepted. For more information on electron counting rule, please refer to the paper by Lu *et al.*, [Nat. Commun. 5, 3666 (2014)]. Note that ECR only works for semiconductors with large band gaps.

Default value: False

The controlling parameters below are used to generate substrate from bulk crystal structure ONLY when ***"substrate"*** *tag is set as* ***"Automatic" (or "Auto"*** *for short****).***

CifFilePath = *string.* This tag specifies the name of crystal structure (in *CIF* format), whose surface structure prediction is to be performed.

Default vale: There is no default value.

MillerIndex = *h k l.* This variable defines the targeted surface as determined by the Miller indices h, k, and l for doing surface reconstruction calculations. For a known bulk structure, once the Miller indices are defined, CALYPSO will automatically generate the idea surface structure denoted by lattice vectors (a_1, a_2) ready for performing surface reconstruction simulations. See below tag of "MatixNotation" for how to define a reconstructed surface denoted by lattice vectors (b_1, b_2). CALYPSO also supports the use of 4-numbered (***h k i l***) Miller index for hexagonal and rhombohedral lattice systems. Here, the variable should be

defined as **MillerIndex = *h k i l.***

Default value: There is no default value.

@MatrixNotation

***interger*$_{11}$ *interger*$_{12}$**

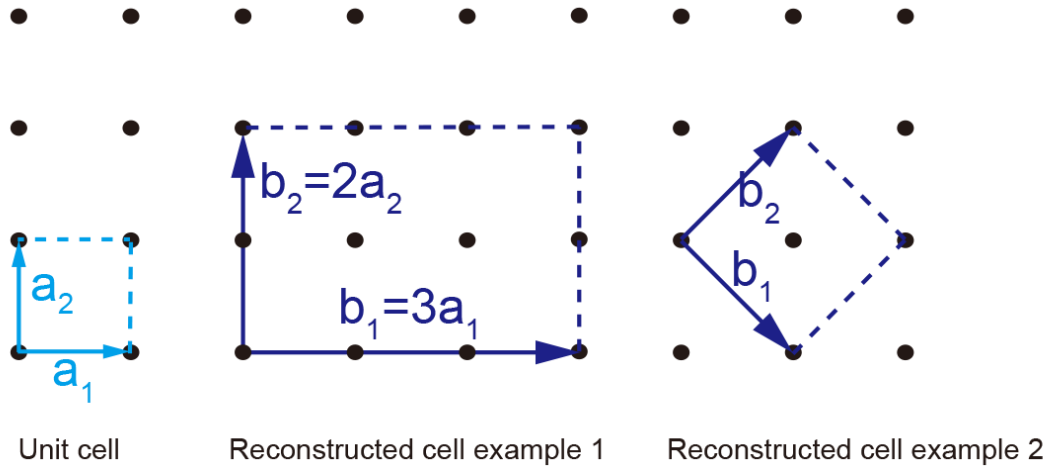
***interger*$_{21}$ *interger*$_{22}$**

@End

This 2×2 matrix is used to define the reconstructed surface. Reconstructed surface lattice vectors (b_1, b_2) can be obtained via multiplying this matrix by the ideal surface lattice vectors (a_1, a_2).

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} integer11 & integer12 \\ integer21 & integer22 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

For example, for the reconstructed surface 1, the 2×2 matrix is written as $\begin{matrix} 3 & 0 \\ 0 & 2 \end{matrix}$, for the reconstructed surface 2, the 2×2 matrix is written as $\begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix}$.



Default value: there is no default value for this variable.

SlabVacuumThick = ***real***. This parameter (in units of angstrom) defines the vacuum region, where the whole slab is separated from its periodic images.

Default value: 10.0

SlabTopMost = *string*. Users can control the topmost layer of the substrate that contains multi-elements via modifying this parameter. For example, for a binary system of TiO_2, this parameter should be defined as “Ti” or “O”. For single-element system, this parameter can be neglected.

Default value: CALYPSO

SlabNumLayers = ***integer.*** This parameter controls the number of layers in the substrate.

Default value: 6

NumRelaxedLayers = *integer.* This parameter controls the number of relaxable layers in the substrate, and it must be smaller than **SlabNumLayers**.
Default value: 2

CapBondsWithH= *logical.* If this parameter is set as "True", the dangling bonds in the bottom side of the slab will be passivated via pseudo-hydrogen atoms.
Default value: True

############################## ***End*** ##########################

########## ***The parameters for inverse design of superhard materials*** ########
################## ***Example is given in section 3.7*** ##################
Hardness= *logical*: when this tag is set as "True", structure design of superhard materials will be performed. Here, we introduce an inverse-design scheme where "hardness" is used as the fitness function, instead of "energy" in a standard structure prediction. After the simulation of CALYPSO inverse-design of superhard materials, one can plot out hardness values versus energy map to dig out the candidate superhard structures possessing simultaneously high hardness and low energies. For more details, please refer to the paper by Zhang *et al*, J. Chem. Phys. 138, 114101 (2013).
Default value: False
############################## ***End*** ##############################

The parameters for atom or molecule adsorption of 2D layer materials
################## ***Examples are given in section 3.8*** ##################
Adsorption= ***logical***: When this flag is set as "True", the prediction of 2D materials with atom or molecule adsorption will be performed.
Default value: False

AdsorptionStyle= ***integer***: It determines which method for generation of adsorption structures should be adopted in the simulation cell.

1: Random generations of structures

2: Generating structures with fixed positions of adatom

Default value: 1

NumberOfTypeAtom= ***integer***: The number of different types of adatoms in the simulation cell.
Default value: There is no default. You must supply this variable.

BothSide= ***logical*****:** when this tag is set as "True", the adatoms are adsorbed on both sides of a 2D layer. Otherwise, the adatoms are adsorbed on single side of a 2D layer.

Default value: False

@Adatoms

string*$_{11}$** ***integer*$_{12}$** ***(integer*$_{13}$*)

string*$_{21}$** ***integer*$_{22}$** ***(integer*$_{23}$*)

…

@End

Defining the adatoms in a format of ***m*×2** for **BothSide**=F or ***m*×3** matrix for **BothSide**=T. The row (***m***) rank of the matrix is determined by **NumberOfTypeAtom**. When **BothSide**=F, the matrix contains two columns for each line. The first column (***string***) is the elemental symbol of each adatom, followed by the number of adatoms in the simulation cell (***integer***). We take the 4×4 supercell of graphene adsorbing eight hydrogen atoms as an example. The **1×2** matrix is defined as

@Adatoms

H **8**

@End

Here, H is the elemental symbol of adatom. **8** indicates that 8 H atoms are adsorbed for 4×4 supercell of graphene.

When **BothSide**=T, the matrix contains three columns for each line. The first column (***string***) is the elemental symbol of each adatom. The second column is the number of adatoms on one side. The third column is the number of adatoms on the other side. We take the 4×4 supercell of graphene adsorbing eight hydrogen atoms as an example. The **1×3** matrix is defined as

@Adatoms

H **4** **4**

@End

Here, H is the elemental symbol of adatom. **4** indicates that 4 H atoms are adsorbed on one side of 4×4 supercell of graphene. **4** indicates that 4 H atoms are adsorbed on the other side of 4×4 supercell of graphene.

Default value: There is no default. You must supply this variable.

@SuperCell

***interger*$_{11}$** ***interger*$_{12}$**

***interger*$_{21}$** ***interger*$_{22}$**

@End

This 2×2 matrix is used to define the substrate. Whose lattice vectors can be obtained via multiplying this matrix by the ideal lattice vectors.

Default value: there is no default value for this variable.

RangeOfZAxis= ***real*1 *real*2**: Defining the range of distances between the 2D layer and adatoms. ***real*1** and ***real*2** specify the maximal and minimal distance, respectively.
Default value: 1.7 1.2

################################## ***End*** ##############################

Parameters for design of optical materials with desirable band gap
################## ***Example is given in section 3.9*** ##################
BandGapDesign = ***logical***: When this flag is set as "True", inverse design of optical materials with desirable band gap will be performed.
Default value: False

TarBandGap = ***real.*** This parameter defines the desirable band gap.
############################## ***End*** ##############################

################ ***The parameters for special constraints*** ################
################## ***Examples are given in section 3.10*** ##################
SpeSpaceGroup= *integer*1 *integer*2**:** Defining the space groups ranging from integer1 to integer2 for generation of structures. If *integer*1 equals to *integer*2, structure generations will be confined to this specified space group. This option is particularly useful when one tries to perform a fixed space group calculation. For a general structure prediction, please use the default value. All 230-space groups will be allowed for generation of structures.
Default value: 1 230

FixCell=*logical*: If this tag is set as "True", the structure prediction with fixed cell parameters will be performed. Note that a file named as **cell.dat** is needed to define the fixed cell parameters. Please refer to examples (section 3.8) for settings of **cell.dat**.
Default value: False

FixAtom= logical**:** If this tag is set as "True", the structure prediction with fixed atomic positions will be performed. Note that a file named as **cell.dat** is needed to define the fixed atomic positions. Please refer to examples (section 3.8) for settings of **cell.dat**.
Default value: False
########################## ***End*** ##########################

Parameters for X-ray diffraction data assisted structural prediction
################## ***Example is given in section 3.11*** ##################
LXRD= *logical*: When this flag is set as "True", X-ray diffraction data assisted structural

prediction will be performed and a file named as **XRD.dat** containing the experimental XRD data should be provided.
Default value: False

WaveLength= *real*. This parameter defines the wavelength used in the experimental XRD measurement.
Default value: There is no default. You must supply this variable.

RangeOf2Theta= *real*1 *real*2**:** Defining the range of 2θ angles from *real*1 to *real*2 for comparison with the experimental XRD data.
Default value: There is no default. You must supply this variable.

StandardPeakPosition = *real*1 *real*2**:** Defining the position of characteristic peak of experimental XRD data in the 2θ range from *real*1 to *real*2. The peak of experimental XRD with maximum intensity is selected as the characteristic peak.
Default value: There is no default. You must supply this variable.
################################ ***End*** ##############################

Parameters for Prediction of Transition States in Solids
################## ***Example is given in section 3.12*** ##################
LTranState = *logical*: When this flag is set as "True", prediction of transition states in solids will be performed. Within this module, only VASP code can be used to perform local structure optimization at the current stage. Note that a file named as **IF_struct.dat** is needed to provide structural information for initial and final solid states. Please refer to examples (section 3.12) for settings of **IF_struct.dat**.
Default value: False

NumberOfImages= *integer*. This parameter defines the number of images, which are used to sample the transition state by the modified nudged elastic band method, for each trial transition path. For the sake of simplicity, this parameter should be divided evenly by **PopSize**. For example, if the **PopSize** is set as 30, **NumberOfImages** can be chosen as 3, 5, or 6. With our modified nudged elastic band method, the number of images for sampling the transition state can be chosen as small as 3 for eight-atom silicon system. A slightly larger **NumberOfImages** is expected for a more complex system.
Default value: There is no default. You must supply this variable.

############################## ***End*** ##############################

The input files (**INCAR_***) and pseudopotential file (**POTCAR)** are needed for local

structure optimization using **VASP**.

The input files **(sinput_*)** and pseudopotential files **(*.psf)** are needed for local structure optimization **using SIESTA.**

The input files **(ginput_*)** are needed for local structure optimization **using GULP.**

The input files **(cp2k.inp_*)** are needed for local structure optimization **using CP2K.**

2.6.2 CALYPSO Outputs

All the major output files are listed in the folder of "**results**":

CALYPSO_input.dat: Backup of the initial input file.

similar.dat: It contains the geometrical structure parameters of predicted structures.

pso_ini_*: It includes the information of the initial structures of the *-th generation.

pso_opt_*: It includes the enthalpy data and structural parameters of the geometrically optimized structures of the *-th generation.

pso_sor_*: The enthalpy data sorted in ascending order of the *-th generation.

struct.dat: Necessary information for all predicted structures (the space group number, the volume, the number of atoms, etc.).

2.6.3 Analysis of Results

CALYPSO calculations typically generate a large number of structures. It is necessary to devise a versatile tool for data analyses. We here develop a **CALYPSO_ANALYSIS KIT (CAK),** allowing automatic structure analysis.

2.6.3.1 Installation of CAK

CAK should be installed on a Linux or a Mac OS X system. Structure analysis through CAK tool needs the use of mathematic lib of **numpy** package in Python. You might need to link the path of Python to the variable 'PYBIN' in your Makefile. Installation of CAK is simple.

```
$ cd CALYPSO_ANALYSIS_KIT
$ make
```

Then add "source /path/caly.sh" in .bashrc.

2.6.3.2 The CAK Commands

Please go to the directory "**results**" and type '**cak.py**'.

```
$ cd "path-to-calculation/results"
$ cak.py
```

An output file named as '**Analysis_Output.dat**' will be generated. By default, Analysis_Output.dat contains space-group and enthalpy information of 50 energetically best

structures. Note that space groups of structures are determined with a tolerance of 0.1 Å by default. Below we show several more advanced analyzing options.

$cak.py -a
All the structures generated by CALYPSO will be analyzed and output.

$cak.py **-n** *integer*
The specified number of best structures will be analyzed and output.

$cak.py **-t** *real*
One of the specified values ranging from 0.01 to 1.0 Å will be used as the tolerance for symmetry analyses of given structures. The default value is set as 0.1.

$cak.py **-m** '*real*1 *real*2 … *real*n'
Allowing multi-tolerance values (separated by space) for symmetry analyses of given structures.

$cak.py **--cif -m** '*real*1 *real*2 … *real*n'
The structure files with .**cif** format will be generated in directories of '**dir_n**' (*n* indicates the tolerance value for symmetry analysis).

$cak.py **--vasp -m** '*real*1 *real*2 … *real*n'
The VASP POSCAR files containing structure information will be generated in directories '**dir_n**' (*n* indicates the tolerance value for symmetry analysis).

$cak.py **--pri –vasp/cif**
The primitive cells of structures will be output.

$cak.py **-p**
Plotting the figure of lowest enthalpy as a function of generations.

$cak.py **--hard**
The structures will be output in the descending order of hardness.

2.6.3.3 The Output Files

(1) ***Analysis_Output.dat***

At least three columns are present in this file by default. The first column is the index of the structures sorted by enthalpies in the ascending order (the numbers in bracket indicate the

structure number out of all structures generated in the CALYPSO calculation). The second column shows the enthalpy data. The corresponding space-group numbers as obtained by the desirable tolerance value for the structures are listed in the third column.

(2) ***Convexhull.dat***

This file contains two columns. The first column shows various stoichiometries for a given binary system. The second column presents the lowest enthalpy for each stoichiometry.

(3) ***plot.dat***

This file contains two columns. The first column shows the generation number, while the second gives the lowest enthalpy for each generation.

*(4)****UCell_m_n.vasp***

This file contains the structure data in conventional cell in the POSCAR format of VASP. *m* and *n* indicate the enthalpy ranking number and the space group number of the structure, respectively.

(5) ***PCell_m_n.vasp***

This file contains the structure data in primitive cell in the POSCAR format of VASP. *m* and *n* indicate the enthalpy ranking number and the space group number of the structure, respectively.

(7) ***m_n.cif***

This file contains the structure data in conventional cell in the **.cif** format. *m* and *n* indicate the enthalpy ranking number and the space group number of the structure, respectively.

(8) ***m_n_p.cif***

This file contains the structure data within a primitive unit cell in the **.cif** format. *m* and *n* indicate the enthalpy ranking number and the space group number of the structure, respectively.

3. Examples

3.1 Crystal Structure Prediction

This section is to show the examples for the three-dimensional crystal structure prediction using CALYPSO code. We take the structure prediction of cubic boron nitride (cBN) as an example. Here, VASP code was used for geometry optimization and enthalpy calculations.
The CALYPSO input file of '**input.dat**' and VASP input files of **INCAR_*** and pseudopotential file of **POTCAR** are needed.

The following files should be present in the working directory:

Files	**Description**
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO containing controllable key parameters
INCAR_*	Input files for VASP
POTCAR	Pseudopotential file for VASP
submit.sh	Job submission file

a) The **calypso.x** can be downloaded in the webpage of http://www.calypso.cn according to different operating systems.

b) The parameters in the **input.dat** are shown below.

```
########### The Parameters of CALYPSO #############
# A string of one or several words contain a descriptive
# name of the system (max. 40 characters).
SystemName = BN
# Number of different atomic species in the simulation,
# where it has two type elements (B and N).
NumberOfSpecies = 2
# Element symbols of the different chemical species.
NameOfAtoms = B N
# Number of atoms for each chemical species in one formula unit.
NumberOfAtoms = 1 1
```

```
# The range of formula unit per cell in your simulation.
NumberOfFormula = 1 1
# The volume of 1 f.u. unit=angstrom^3.
Volume = 20.0
# Minimal distances between atoms of each chemical species.
@DistanceOfIon
 1.0 0.8
 0.8 0.8
@End
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
# The proportion of the structures generated by PSO
# against the new structures in each generation.
PsoRatio = 0.6
# The population size. Normally, it is larger for larger systems.
PopSize = 30
# It determines which local optimization program
# should be used in the simulation.
ICode = 1
# The Number of local optimization for each structure.
NumberOfLocalOptim = 4
# It is the k-point sampling solution in local optimization.
Kgrid = 0.12 0.06
# The script for submitting local optimization calculations.
Command = sh submit.sh
# The Max step for iteration
MaxStep =30
#################### End Parameters ######################
```

c) We strongly suggest that you use the multi-optimization strategy for structure relaxation. For VASP, the number of **INCAR_*** (INCAR_1, INCAR_2, …) files is in accordance with the setting tag **`NumberOfLocalOptim`**. In this example, four INCAR files are used for structure relaxation. **INCAR_1** and **INCAR_2** files are used to perform very coarse structure relaxation with the fixed volume, and **INCAR_3** and **INCAR_4** files are used to perform full structure relaxation (i.e., variable lattice parameters, variable volumes, and variable atomic coordinates) with medium and accurate precision, respectively.

```
# INCAR_1:
SYSTEM = local optimization
PREC = LOW
EDIFF = 3e-2
IBRION = 2
ISIF = 2
NSW = 40
ISMEAR = 0 ; SIGMA = 0.05
POTIM = 0.50
LWAVE = FALSE
LCHARG = FALSE
ISTART = 0
PSTRESS = 3000
EDIFFG = -4e-2
```

```
# INCAR_2:
SYSTEM = local optimization
PREC = Normal
EDIFF = 2e-2
IBRION = 2
ISIF = 4
NSW = 40
ISMEAR = 0
SIGMA = 0.05
POTIM = 0.2
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 3000
EDIFFG = -4e-1
```

```
# INCAR_3:
SYSTEM = local optimization
PREC = Normal
EDIFF = 2e-4
IBRION = 1
ISIF = 3
NSW = 40
ISMEAR = 0 ; SIGMA = 0.05
POTIM = 0.1
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 3000
EDIFFG = 2e-3
```

```
# INCAR_4:
SYSTEM = local optimization
ENCUT = 600
EDIFF = 1e-5
IBRION = 2
ISIF = 3
NSW = 80
ISMEAR = 0 ; SIGMA = 0.05
POTIM = 0.1
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 3000
EDIFFG = 2e-3
```

d) POTCAR should be provided. **ATTENTION!!!** The order of elements in POTCAR must be identical to the element order in the setting tag of **`NameOfAtoms`**.

e) '**submit.sh**' is the job submission file for the VASP calculations.
Here is an example of submit.sh:

```
#!/bin/sh
mpdboot
mpiexec -n 12 ~/bin/vasp.4.6 > vasp.log 2> /dev/null
```

Once all input files are ready, you can simply type

```
$ ./calypso.x > caly.log &
```

to execute the CALYPSO in the sequential mode. Or you can write this command into the **pbs** script and submit it in a queue system.

CALYPSO run will generate the "**results**" folder in the current directory you have specified. The **"results"** folder contains all the outputs files of CALYPSO. To analyze the results, please simply type the following commands:

```
$ cd results
$ cak.py  --cif
```

The index of the structures sorted by enthalpies in the ascending order will be present in the Analysis_Output.dat file, and all predicted structure files by CALYPSO with cif formats are shown in the **dir_0.1** directory. Please see section 2.6.3 for more information about the analyses of CALYPSO results.

3.2 Two-Dimensional Structure Prediction

This section is to show the examples for the two-dimensional structure prediction using CALYPSO code. Here, VASP code was used for geometry optimization and enthalpy calculations. The CALYPSO input file of '**input.dat**' and VASP input files of **INCAR_*** and pseudopotential file of **POTCAR** are needed.

The following files should be present in the working directory:

Files	Description
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO containing controllable key parameters
INCAR_*	Input files for VASP
POTCAR	Pseudopotential for VASP
submit.sh	Job submission file

a) The **calypso.x** can be downloaded according to different operating systems.

b) The following parameters in **input.dat** are shown for the B_3N_2 two-dimensional system.

```
# A string of one or several words contain a descriptive
# name of the system (max. 40 characters).
SystemName = B3N2
# Number of different atomic species in the simulation.
NumberOfSpecies = 2
# Element symbols of the different chemical species.
NameOfAtoms = B N
# Number of atoms for each chemical species in one formula unit.
NumberOfAtoms =  3 2
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
# The proportion of the structures generated by PSO.
PsoRatio = 0.6
# The population size.
PopSize = 20
# It determines which local optimization program
# should be interfaced in the simulation.
ICode= 1
# The Number of local optimization for each structure.
NumberOfLocalOptim= 3
```

```
# The precision of the K-point sampling for local optimization
Kgrid = 0.1 0.07
# The command to perform local optimization calculation.
Command = sh submit.sh
# The Max step for iteration
MaxStep =50
###### The Parameters for 2D structure prediction ########
# If True, a 2D structure prediction is performed.
2D = T
# The number of layers
MultiLayer = 2
# The Area of 2D system
Area = 14
# The distortion value along the C axis
DeltaZ = 0
# The gap between two layers
LayerGap=5
# The vacuum gap between the top surface of the slab and the top lattice,
# and between the bottom surface of the slab and the bottom lattice.
VacuumGap=5
# The number atoms for each layer
@LayerType
1 1
2 1
@End
# Minimal distance between atoms of each chemical species.
# Unit is in angstrom.
LAtom_Dis = 1.4
#######################END 2D Parameters ########################
```

b) We strongly suggest that you use the multi-stage strategy for structure relaxation. For VASP, the number of **INCAR_*** (INCAR_1, INCAR_2, …) is in accord with the setting tag of `NumberOfLocalOptim`. In this example, we use three INCAR files for structure relaxation. **INCAR_1** and **INCAR_2** are used to perform very coarse structure relaxation keeping the volume fixed, **INCAR_3** is used to perform full structure relaxation with accurate precision.

```
# INCAR_1
SYSTEM = B3N2
PREC = LOW
EDIFF = 3e-3
IBRION = 2
ISIF = 2
NSW = 50
ISMEAR = 1
SIGMA = 0.2
POTIM = 0.50
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 0.001
EDIFFG = 4e-2
```

```
# INCAR_2
SYSTEM = B3N2
PREC = LOW
EDIFF = 2e-3
IBRION = 2
ISIF = 4
NSW = 50
ISMEAR = 1
SIGMA = 0.2
POTIM = 0.200
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 0.001
EDIFFG = 1e-2
```

```
# NCAR_3
SYSTEM = B3N2
PREC = Accurate
ENCUT = 400
EDIFF = 2e-4
IBRION = 2
ISIF = 3
NSW = 60
ISMEAR = 1
SIGMA = 0.2
POTIM = 0.10
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 0.001
EDIFFG = 1e-4
```

c) POTCAR should be provided. **ATTENTION!!!** The order of elements in POTCAR must be identical to the element order in the setting tag of **`NameOfAtoms`**.

d) '**submit.sh**' is the job submission file for performing the VASP calculations.
Here is an example of submit.sh:

```
#!/bin/sh
mpdboot
mpiexec -n 12 ~/bin/vasp.4.6 > vasp.log 2> /dev/null
```

Once all input files are ready, you can just type

```
$ ./calypso.x > caly.log &
```

to execute the CALYPSO in the sequential mode. Or you can write this command into the pbs script and submit it.

CALYPSO will generate the "**results**" folder, which contains all outputs files. To analyze the results of CALYPSO, just type following command:

```
$ cd results
$ cak.py  --cif
```

The index of the structures sorted by enthalpies in the ascending order is presented in Analysis_Output.dat file, and these newly predicted structure files with cif format are shown in **dir_0.1** directory. Please see section 2.6.3 for further information about the analysis of CALYPSO results.

3.3 Cluster Structure Prediction

This section is to show the examples for the cluster structure prediction using CALYPSO code. In section 3.3.1, an example for boron clusters that composed of 6 atoms (B_6) is presented; The Gaussian09 code was used for geometry optimization and energy calculations. In section 3.3.2, an example for Ti_4 clusters is given and the local optimization is performed by VASP. Finally, an example for B_{12} clusters is provided in section 3.3.3; The CP2K code was used for geometry optimization and energy calculations.

3.3.1 Tutorial for B_6 clusters

In this tutorial, a CALYPSO structure search for B_6 clusters is presented. The local structure optimizations are performed by Gaussian09 code, so you have to install the Gaussian09 in your machine.

The following files should be present in the working directory:

Files	Description
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO containing controllable key parameters
gsinput_*	Input files for Gaussian
submit.sh	Job submission file

a) The **calypso.x** can be downloaded according to different operating systems.

b) The parameters in **input.dat** are shown below

```
######## The Basic Parameters of CALYPSO ##########
# A string of one or several words contain a descriptive
# name of the system (max. 40 characters).
SystemName = B6Cluster
# Number of different atomic species in the simulation.
NumberOfSpecies = 1
# Element symbols of the different chemical species.
NameOfAtoms = B
# Number of atoms for each chemical species in one formula unit.
NumberOfAtoms =  6
# The range of formula unit per cell in your simulation.
NumberOfFormula = 1 1
```

```
# Minimal distance between atoms of each chemical species.
@DistanceOfIon
 1.7
@End
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
# The proportion of the structures generated by PSO.
PsoRatio = 0.8
# The population size.
PopSize = 20
# It determines which local optimization program
# should be interfaced in the simulation.
ICode= 7
# The Number of local optimization for each structure.
NumberOfLocalOptim= 2
# The command to perform local optimization calculation.
Command = sh submit.sh
# The Max step for iteration
MaxStep =50
# If True, the metropolis rule is used during iterations
LMC= T
############ End Basic Parameters ##################
###### The Parameters For Cluster structure prediction #########
# If True, a cluster structure prediction is performed.
Cluster= T
# The Vacancy length for 3 dimensional space
Vacancy = 15 15 15
################ End Cluster Parameters ###################
```

c) We strongly suggest that you use the multi-stage strategy for structure relaxation. For Gaussian, the number of **gsinput_*** (gsinput_1 and gsinput_2) is in accord with the setting tag `NumberOfLocalOptim`. In this example, we use two gsinput_* files for structure relaxation. The **gsinput_1** is used to perform very coarse structure relaxation, and **gsinput_2** is used to perform structure relaxation with medium precision. During the structure relaxation, **gsinput_*** files will be renamed as **gsinput**, and atomic coordinates will be automatically attached by CALYPSO.

```
%chk=calypso.chk
%nprocs=12
%Mem=12GB
#p RHF/PBEPBE/3-21G
SCF=(fermi,MaxCycle=565,conver=4)
OPT=(Redundant,MaxCycle=50,MaxStep=29,loose)
NoSymm

dr38 Geometry Optimization

0 1
```

```
%chk=calypso.chk
%nprocs=12
%Mem=12GB
#p RHF/PBEPBE/3-21G
SCF=(xqc,MaxCycle=565,conver=5)
OPT=(Cartesian,MaxCycle=1500,MaxStep=29,loose)
GEOM=(checkpoint)
NoSymm

dr38 Geometry Optimization

0 1
there is an empty line!!!
```

d) '**submit.sh**' is the job submission file for the Gaussian calculation.

Here is an example of **submit.sh**:

```
#!/bin/sh
g09 < gsinput > gsoutput
```

Once all the input files are ready, you can just type

```
$ ./calypso.x > caly.log &
```

to execute the CALYPSO in the sequential mode. Or you can write this command into the pbs script and submit it.

CALYPSO will generate the "**results**" folder, which contains all most the output files. To analyze the results, just type following command:

```
$ cd results
$ cak.py  -cif
```

The index of the structures sorted by energies in the ascending order is presented in **Analysis_Output.dat** file, and these newly predicted structure files with cif format are shown in **dir_origin** directory. Please see section 2.6.3 for further information about the analysis of CALYPSO results.

3.3.2 Tutorial for Ti_4 clusters

In this tutorial, a CALYPSO structure search for Ti_4 clusters is presented. The local structural optimizations are performed by VASP, so you have to install the VASP code in your machine. All the reference files can be found in "**Path-to-CALYPSO/Example/Ti4_VASP"**.

The following files should be present in the working directory:

Files	Description
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO containing controllable key parameters
INCAR_*	Input files for VASP
POTCAR	Pseudopotential for VASP
submit.sh	Job submission file

a) The **calypso.x** can be downloaded according to different operating systems.

b) The parameters in **input.dat** are shown below.

```
##### The Basic Parameters of CALYPSO #########
# A string of one or several words contain a descriptive
# name of the system (max. 40 characters).
SystemName = Ti
# Number of different atomic species in the simulation.
NumberOfSpecies = 1
# Element symbols of the different chemical species.
NameOfAtoms = Ti
# Number of atoms for each chemical species in one formula unit.
NumberOfAtoms =  4
# The range of formula unit per cell in your simulation.
NumberOfFormula = 1 1
# Minimal distance between atoms of each chemical species.
@DistanceOfIon
 2.6
@End
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
```

```
# The proportion of the structures generated by PSO.
PsoRatio = 0.8
# The population size.
PopSize = 30
# It determines which local optimization program
# should be interfaced in the simulation.
ICode= 1
# The Number of local optimization for each structure.
NumberOfLocalOptim= 1
# The command to perform local optimization calculation.
Command = sh submit.sh
# The Max step for iteration
MaxStep =50
# If True, the metropolis rule is used during iterations
LMC= T
#########################End Basic Parameters ###################
###### The Parameters For Cluster structure prediction #########
# If True, a cluster structure prediction is performed.
Cluster= T
# The Vacancy length for 3 dimensional space
Vacancy = 12 12 12
####################End           Cluster           Parameters
#######################
```

c) In this example, geometry optimization is performed by VASP code and only one input file for VASP (named as "**INCAR_1**") is needed.

```
SYSTEM = Ti
PREC = Accurate
ENCUT = 300
EDIFF = 1e-4
IBRION = 2
ISIF = 2
NSW = 1000
ISMEAR = 0
SIGMA = 0.0001
POTIM = 0.100
LCHARG = FALSE
LWAVE = FALSE
EDIFFG = -0.01
ISPIN = 2
ISYM = 0
```

d) POTCAR should be provided by the user.

e) '**submit.sh**' is the job submission file for performing the VASP calculations.
Here is an example of **submit.sh**:

```
#!/bin/sh
mpdboot
mpiexec -n 12 ~/bin/vasp.4.6 > vasp.log 2> /dev/null
```

Once all the input files are ready, you can just type

```
$ ./calypso.x > caly.log &
```

to execute the CALYPSO in the sequential mode. Or you can write this command into the pbs script and submit it.

CALYPSO will generate the "**results**" folder, which contains all most the output files. To analyze the CALYPSO results, just type following command:

```
$ cd results
$ cak.py  --cif
```

The index of the structures sorted by energies in the ascending order is presented in **Analysis_Output.dat** file, and these newly predicted structure files with cif format are shown in **dir_origin** directory. Please see section 2.6.3 for further information about the analysis of CALYPSO results.

3.3.3 Tutorial for B_{12} clusters

In this tutorial, a CALYPSO structure search for B_{12} clusters is presented. The local structural optimizations are performed by CP2K, so you have to install the CP2K code in your machine. All the reference files can be found in "**Path-to-CALYPSO/Example/B12_cp2k"**.

The following files should be present in the working directory:

Files	Description
calypso.x	The executable file for running CALYPSO program
EMSL_BASIS_SETS	Basis sets files for CP2K
POTENTIAL	Pseudopotential files for CP2K
input.dat	The input file of CALYPSO containing controllable key parameters
cp2k.inp_*	Input files for CP2K
submit.sh	Job submission file

a) The **calypso.x** can be downloaded according to different operating system.

b) The following parameters in **input.dat** are shown for the B_{12} cluster.

```
######### The Basic Parameters of CALYPSO ##############
# A string of one or several words contain a descriptive
# name of the system (max. 40 characters).
SystemName = B
# Number of different atomic species in the simulation.
NumberOfSpecies = 1
# Element symbols of the different chemical species.
NameOfAtoms = B
# Number of atoms for each chemical species in one formula unit.
NumberOfAtoms =  12
# The range of formula unit per cell in your simulation.
NumberOfFormula = 1 1
# Minimal distance between atoms of each chemical species.
@DistanceOfIon
 1.7
@End
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
# The proportion of the structures generated by PSO.
PsoRatio = 0.8
```

```
# The proportion of the structures generated by PSO.
PsoRatio = 0.8
# The population size.
PopSize = 30
# It determines which local optimization program
# should be interfaced in the simulation.
ICode= 6
# The Number of local optimization for each structure.
NumberOfLocalOptim= 1
# The command to perform local optimization calculation.
Command = sh submit.sh
# The Max step for iteration
MaxStep =50
# If True, the metropolis rule is used during iterations
LMC= T
#########################End Basic Parameters ###################

###### The Parameters For Cluster structure prediction #########
# If True, a cluster structure prediction is performed.
Cluster= T
# The Vacancy length for 3 dimensional space
Vacancy = 12 12 12
####################End Cluster Parameters ######################
```

c) In this example, geometry optimization is performed by CP2K code and only one input file for CP2K (named as "**cp2k.inp_1**") is needed. During the structure relaxation, **cp2k.inp_1** file will be renamed as **cp2k.inp**, and atomic coordinates will be automatically inserted into this file (behind the marker line "#########").

```
&FORCE_EVAL
  METHOD QS
  &DFT
    BASIS_SET_FILE_NAME EMSL_BASIS_SETS
    POTENTIAL_FILE_NAME POTENTIAL
    &MGRID
      CUTOFF 250
      REL_CUTOFF 50
    &END MGRID
       &QS
      EPS_DEFAULT 1.0E-10
```

```
      &END QS
    &POISSON
       PSOLVER MT
    &END POISSON
    &SCF
      SCF_GUESS ATOMIC
      &OT ON
        MINIMIZER DIIS
      &END OT
      MAX_SCF  100
      EPS_SCF     1.0E-5
      &PRINT
        &RESTART OFF
        &END
      &END
    &END SCF
    &XC
      &XC_FUNCTIONAL PBE
      &END XC_FUNCTIONAL
    &END XC
  &END DFT
  &SUBSYS
########
    &KIND B
      BASIS_SET 6-311Gxx
      POTENTIAL GTH-PBE-q3
    &END KIND
  &END SUBSYS
&END FORCE_EVAL
&GLOBAL
  PROJECT cp2k
  RUN_TYPE GEOMETRY_OPTIMIZATION
  PRINT_LEVEL LOW
  &TIMINGS
     THRESHOLD 0.000001
  &END
&END GLOBAL
&MOTION
  &GEO_OPT
    OPTIMIZER BFGS
    MAX_ITER  1000
    MAX_FORCE 0.00045
    RMS_FORCE 0.0003
  &END GEO_OPT
&END MOTION
```

c) The users should provide the "**EMSL_BASIS_SETS**" and "**POTENTIAL**" files, which contain the basis set and pseudopotential for running CP2K code.

d) '**submit.sh**' is the job submission file for performing the CP2K calculations.
Here is an example of submit.sh:

```
#!/bin/sh
mpirun.lsf cp2k.popt -i cp2k.inp > out.cp2k 2>&1
```

Once all the input files are ready, you can just type

```
$ ./calypso.x > caly.log &
```

to execute the code in the sequential mode. Or you can write this command into the pbs script and submit it.

CALYPSO will generate the "**results**" folder in current directory, which contains all most the outputs files of CALYPSO. To analyze the results, just type following command:

```
$ cd results
$ cak.py  --cif
```

The index of the structures sorted by energies in the ascending order is presented in **Analysis_Output.dat** file, and these newly predicted structure files with cif format are shown in **dir_orign** directory. Please see section 2.6.3 for further information about the analysis of CALYPSO results.

3.4 Molecular Structure Prediction

This section is to show the example for the structure prediction with fixed rigid molecules using CALYPSO code. Here, SIESTA code was used for geometry optimization and enthalpy calculations. The CALYPSO input file of '**input.dat**', rigid molecular structure information file of "**MOL**", SIESTA input files of **sinput_*** and pseudopotential file of ***.psf** are needed.

The following files should be present in the working directory:

Files	Description
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO containing controllable key parameters
sinput_*	Input files for SIESTA
MOL	Rigid molecular structure information with Z_Matrix format
*.psf	Pseudopotential for SIESTA
submit.sh	Job submission file

a) The **calypso.x** can be downloaded according to different operating systems.

b) The following parameters in **input.dat** are shown for the CH_4 system.

```
########### The Basic Parameters of CALYPSO #############
# A string of one or several words contain a descriptive
# name of the system (max. 40 characters).
SystemName = CH4
# Number of different atomic species in the simulation.
NumberOfSpecies = 2
# Element symbols of the different chemical species.
NameOfAtoms = C H

# Number of atoms for each chemical species in one formula unit.
NumberOfAtoms = 1 4
```

```
a# The range of formula unit per cell in your simulation.
NumberOfFormula = 1 1
# The volume per formula unit. Unit is in angstrom^3.
Volume=60.0
# Minimal distance between atoms of each chemical species.
@DistanceOfIon
1.0  1.0
1.0 1.0
@End
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
# The proportion of the structures generated by PSO.
PsoRatio = 0.6
# The population size. Normally, it has a larger number for larger
systems.
PopSize = 20
# It determines which local optimization program
# should be interfaced in the simulation.
ICode= 2
# The Number of local optimization for each structure.
NumberOfLocalOptim= 3
# The precision of the K-point sampling for local optimization
Kgrid = 0.1 0.07
# The command to perform local optimization calculation.
Command = sh submit.sh
# The Max step for iteration
MaxStep =50
##########################End Basic Parameters ###################

###### The Parameters For Molecular structure prediction ########
# If True, a molecular structure prediction is performed.
Mol=T
# The number of type molecule
NumberOfTypeMolecule=1
# The number of molecule for each type
NumberOfMolecule=2
# The minimum distance between two rigid molecules
DistOfMol =1.5
######################### End Molecular Parameters ###############
```

c) MOL file contains the rigid molecular structure information with Z_Matrix format. The Z_matrix is a way to represent a system built of atoms with internal coordinates. It provides a description of each atom in a molecule in terms of its atomic type, bond length, bond angle,

and dihedral angle. The detailed descriptions about Z-Matrix can be found at http://en.wikipedia.org/wiki/Z-matrix_(chemistry).

Here we take CH4 as an example

```
5
1 0 0 0 0.0 0.0 0.0 1 1 1
2 1 0 0 1.044 0.0 0.0 0 1 1
2 1 2 0 1.044 109.48 0.0  0 0 1
2 1 2 3 1.044 109.48 120  0 0 0
2 1 2 3 1.044 109.48 120  0 0 0
```

The first line gives the total number of the atoms in the rigid molecular structure information; the following lines give the internal coordinates of atoms.

Nspecie I J K R A T ifr ifa ift

Nspecie(integer) indicates the species number of the atom.

I, J, K(*integer*): atomic order in molecule is used to define the internal coordinates for current atomic coordinates.

R(*real*): bond length is made by current atom with respect to I atom.

A(*real*): bond angle is made by current atom with respect to I and J atoms.

T(*real*): dihedral angle is made by current atom with respect to I, J and K atoms.

Ifr, ifa, ift(*integer*): flags that indicate whether r, a, and t should be varied during local structural relaxation. 0 for fixed and 1 for varying.

d) We strongly suggest that you use the multi-stage strategy for structure relaxation. For SIESTA, the number of **sinput_*** (sinput_1, sinput_2,…) is in accord with the setting tag of `NumberOfLocalOptim`. In this example, we use three SIESTA input files (**sinput_1**, **sinput_2**, **sinput_3**, and **sinput_4**) for structure relaxation. **sinput_1** and **sinput_2** are used to perform very coarse structure relaxation, while **sinput_3** is used to perform full structure relaxation with accurate precision.

```
# sinput_1
SystemName siesta
SystemLabel siesta
NumberOfSpecies 2
NumberOfAtoms  10
%block ChemicalSpeciesLabel
1    6 C
2    1 H
%endblock ChemicalSpeciesLabel
PAO.BasisSize   SZ
kgrid_cutoff    8.0 Ang
MeshCutoff     80 Ry
PAO.EnergyShift 0.02 Ry
XC.functional         GGA
XC.authors            PBE
MaxSCFIterations       100
DM.MixingWeight        0.150
DM.Tolerance          1.d-4
DM.NumberPulay         5
# Relaxation, smearing, etc.
ElectronicTemperature   3000 K
MD.TypeOfRun          cg
Optim.Broyden         .true.
MD.VariableCell        .true.
MD.ConstantVolume      .false.
MD.MaxForceTol         1d-3 eV/Ang
MD.NumCGsteps          5
Use-Save-CG           .true.
Use-Save-XV           .true.
MD.Broyden.Initial.Inverse.Jacobian 0.20
MD.RemoveIntramolecularPressure .true.
MD.TargetPressure      100.0 GPa
ZM.ForceTolLen 0.04 eV/Ang
ZM.ForceTolAng 0.0001 eV/deg
ZM.MaxDisplLen 0.1 Ang
ZM.MaxDisplAng 20.0 deg
AtomicCoordinatesFormat NotScaledCartesianAng
%include Zmatrix.data
%block MM.Potentials
1 1 C6 16.292 0.5
2 2 C6  0.735 0.5
1 2 C6  3.185 0.5
%endblock MM.Potentials
```

```
# sinput_2
SystemName siesta
SystemLabel siesta
NumberOfSpecies 2
NumberOfAtoms   10
%block ChemicalSpeciesLabel
1    6 C
2    1 H
%endblock ChemicalSpeciesLabel
PAO.BasisSize   SZ
%endblock PAO.BasisSizes
kgrid_cutoff    8.0 Ang
MeshCutoff     80 Ry
PAO.EnergyShift 0.01 Ry
XC.functional         GGA
XC.authors            PBE
MaxSCFIterations       100
DM.MixingWeight        0.150
DM.Tolerance          1.d-5
DM.NumberPulay         5
ElectronicTemperature   3000 K
MD.TypeOfRun          cg
Optim.Broyden         .true.
MD.VariableCell        .true.
MD.ConstantVolume      .false.
MD.MaxForceTol        2d-5 eV/Ang
MD.NumCGsteps         40
Use-Save-CG           .true.
Use-Save-XV           .true.
MD.Broyden.Initial.Inverse.Jacobian 0.20
MD.RemoveIntramolecularPressure .true.
MD.TargetPressure      100.000010 GPa
ZM.ForceTolLen 0.04 eV/Ang
ZM.ForceTolAng 0.0001 eV/deg
ZM.MaxDisplLen 0.1 Ang
ZM.MaxDisplAng 20.0 deg
AtomicCoordinatesFormat NotScaledCartesianAng
%include Zmatrix.data
%block MM.Potentials
1 1 C6 16.292 1.392
2 2 C6  0.735 2.025
1 2 C6  3.185 1.649
%endblock MM.Potentials
```

```
# sinput_3
SystemName siesta
SystemLabel siesta
NumberOfSpecies 2
NumberOfAtoms   10
%block ChemicalSpeciesLabel
1    6 C
2    1 H
%endblock ChemicalSpeciesLabel
PAO.BasisSize   DZ
kgrid_cutoff    8.0 Ang
MeshCutoff     80 Ry
PAO.EnergyShift 0.002 Ry
XC.functional         GGA
XC.authors            PBE
MaxSCFIterations       100
DM.MixingWeight        0.150
DM.Tolerance          1.d-5
DM.NumberPulay         5
ElectronicTemperature   3000 K
MD.TypeOfRun          cg
Optim.Broyden         .false.
MD.VariableCell        .true.
MD.ConstantVolume      .false.
MD.MaxForceTol         2d-5 eV/Ang
MD.NumCGsteps          60
Use-Save-CG           .true.
Use-Save-XV           .true.
MD.RemoveIntramolecularPressure .true.
MD.TargetPressure      100.000010 GPa
ZM.ForceTolLen 0.04 eV/Ang
ZM.ForceTolAng 0.0001 eV/deg
ZM.MaxDisplLen 0.1 Ang
ZM.MaxDisplAng 20.0 deg
AtomicCoordinatesFormat NotScaledCartesianAng
%include Zmatrix.data
%block MM.Potentials
1 1 C6 16.292 1.392
2 2 C6  0.735 2.025
1 2 C6  3.185 1.649
%endblock MM.Potentials
```

e) The psedopotential ***.psf** (C.psf and H.psf in this case) should be provided by the user.

f) '**submit.sh**' is the job submission file for performing the SIESTA calculation.
Here is an example of submit.sh:

```
#!/bin/sh
siesta < siesta.fdf > siesta.out
```

Note: the SIESTA output file should be named as "siesta.out".

Once all the input files are ready, you can just type

```
$ ./calypso.x > caly.log &
```

to execute the CALYPSO code in the sequential mode. Or you can write this command into the pbs script and submit it.

CALYPSO will generate the "**results**" folder, which contains all the output files of CALYPSO. To analyze the CALYPSO results, just type following command:

```
$ cd results
$ cak.py  --cif
```

The index of the structures sorted by enthalpies in the ascending order is presented in Analysis_Output.dat file, and these newly predicted structure files with cif format are shown in **dir_0.1** directory. Please see section 2.6.3 for further information about the analysis of CALYPSO results.

3.5 Variable Stoichiometry Structure Prediction

This section is to show the example for variable stoichiometry structure prediction using CALYPSO code. Here, VASP code was used for geometry optimization and enthalpy calculations. The CALYPSO input file of '**input.dat**', VASP input files of **INCAR_*** and pseudopotential file of **POTCAR** are needed.

The following files should be present in the working directory:

File:	Description
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO containing controllable key parameters
INCAR_*	Input files for VASP
POTCAR	Pseudopotential for VASP
submit.sh	Job submission file

a) The **calypso.x** can be downloaded according to different operating systems.

b) The following parameters in **input.dat** are shown for the LiH system.

```
########### The Parameters of CALYPSO #############
# A string of one or several words contain a descriptive
# name of the system (max. 40 characters).
SystemName = LiH
# Number of different atomic species in the simulation,
# where it has two type elements (B and N).
NumberOfSpecies = 2
# Element symbols of the different chemical species.
NameOfAtoms = Li H
```

```
# Number of atoms for each chemical species in one formula unit.
NumberOfAtoms = 1 1
# The range of formula unit per cell in your simulation.
NumberOfFormula = 1 1
# The volume of 1 f.u. unit=angstrom^3.
Volume = 5.0
# Minimal distances between atoms of each chemical species.
@DistanceOfIon
 1.0 0.8
 0.8 0.8
@End
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
# The proportion of the structures generated by PSO
# against the new structures in each generation.
PsoRatio = 0.6
# The population size. Normally, it is larger for larger systems.
PopSize = 30
# It determines which local optimization program
# should be used in the simulation.
ICode = 1
# The Number of local optimization for each structure.
NumberOfLocalOptim = 4
# It is the k-point sampling solution in local optimization.
Kgrid = 0.12 0.06
# The script for submitting local optimization calculations.
Command = sh submit.sh
# The Max step for iteration
MaxStep =30
# If True, Variational Stoichiometry structure prediction is performed
VSC=T
VSCEnergy= 0 0
# The Max Number of Atoms in unit cell
MaxNumAtom=20
# The Variation Range for each type atom
CtrlRange= 1 1 1 7
##################### End Parameters ######################
```

c) We strongly suggest that you use the multi-stage strategy for structure relaxation. For VASP, the number of **INCAR_*** (INCAR_1, INCAR_2, …) is in accord with the setting tag of `NumberOfLocalOptim`. In this example, we use four INCAR files for structure relaxation. **INCAR_1** and **INCAR_2** are used to perform very crude structure relaxation keeping the volume fixed, **INCAR_3** is used to perform full structure relaxation with medium precision,

while **INCAR_4** is used to perform very accurate calculation.

```
# INCAR_1:
SYSTEM = local optimization
PREC = LOW
EDIFF = 3e-2
IBRION = 2
ISIF = 2
NSW = 40
ISMEAR = 0 ; SIGMA = 0.05
POTIM = 0.50
LWAVE = FALSE
LCHARG = FALSE
ISTART = 0
PSTRESS = 3000
EDIFFG = -4e-2
```

```
# INCAR_2:
SYSTEM = local optimization
PREC = Normal
EDIFF = 2e-2
IBRION = 2
ISIF = 4
NSW = 40
ISMEAR = 0
SIGMA = 0.05
POTIM = 0.2
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 3000
EDIFFG = -1e-2
```

```
# INCAR_3:
SYSTEM = local optimization
PREC = Normal
EDIFF = 2e-4
IBRION = 1
ISIF = 3
NSW = 40
ISMEAR = 0 ; SIGMA = 0.05
POTIM = 0.1
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 3000
EDIFFG = 1e-3
```

```
# INCAR_4:
SYSTEM = local optimization
ENCUT = 600
EDIFF = 1e-5
IBRION = 2
ISIF = 3
NSW = 80
ISMEAR = 0 ; SIGMA = 0.05
POTIM = 0.1
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 3000
EDIFFG = 1e-4
```

d) POTCAR should be provided by the user. The order of element of POTCAR must be the same with the setting tag of `NameOfAtoms`.

e) '**submit.sh**' is the submission job file for performing the VASP calculations.
Here is an example of submit.sh:

```
#!/bin/sh
mpdboot
mpiexec -n 12 ~/bin/vasp.4.6 > vasp.log 2> /dev/null
```

Once all input files are ready, you can just type

```
$ ./calypso.x > caly.log &
```

to execute the CALYPSO code in the sequential mode. Or you can write this command into the pbs script and submit it.

CALYPSO will generate the "**results**" folder, which contains all the output files of CALYPSO. To analyze the results, just type following command:

```
$ cd results
$ cak.py  --cif
```

The index of the structures sorted by enthalpies in the ascending order is presented in ./dir_Li*H*/Analysis_Output.dat file, and these newly predicted structure files with cif format are shown in **dir_0.1** directory.

3.6 Surface Structure Prediction

This section introduces the surface structure prediction module. It includes a description of the required input files, a typical run of the program, and result analysis. The most important input files are: **input.dat** (control parameters), **SUBSTRATE.surf** (the structure files of substrate, both cif and VASP POSCAR format are supported), and input files for **VASP/DFTB+**. Most of the output files reside in ***results*** directory. Note that this part of **CALYPSO** package is currently in the early stage of development, there might be large changes on the control parameters and output files in the following versions. For surface structure prediction, DFTB+ and VASP package are supported for local optimization by now. This section will give a general introduction to the files used to run the program in the first place, then a relatively detailed description of each files, followed by two example runs of the program via VASP and DFTB+ packages.

3.6.1 Diamond (111) surface reconstruction prediction

This subsection will introduce how to predict the (111) surface reconstruction of diamond using VASP as the local relaxation code. In this example run, the substrate will be generated automatically from crystal structures. To perform surface structure prediction, one needs to create a directory and prepare the following files:

File:	Description
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO specifying the controllable key parameters
INCAR_*	Input files for VASP
POTCAR	Pseudopotential for VASP
submit.sh	Job submission file

a) The **calypso.x** can be downloaded according to operating system.

b) input.dat: This file specifies all the control parameters that tell the program how to do the job of global surface structure search. Most of the parameters have default values and do not need to be changed. Note that the Substrate tag is specified to '***Automatic***', which means the substrate will be generated from the bulk crystal.

Here is an example **input.dat** for diamond (111) surface reconstruction prediction.

```
LSurface    = T  # Surface reconstruction predictions
ICode       = 1  #VASP
Kgrid       = 0.2
PopSize     = 20
MaxStep     = 30
PsoRatio    = 0.6
SurfaceThickness = 2.0
#|atomic symbol|count|
@SurfaceAtoms
C    4
@End
#-------------------------------------------------
#Following parameters are used to build
#the surface from bulk crystal info reconstruction symmetry
Substrate  = Automatic
@MatrixNotation
2    0
0    1
@End
UseCifFile = T
CifFilePath = dia-vasp.cif
MillerIndex = 1 1 1
SlabDepth = 0.0
SlabNumLayers = 6
NumRelaxedLayers = 2
CapBondsWithH = T
```

c) VASP input files: For VASP users, these files must present in the current work directory: **INCAR_1, INCAR_2, POTCAR-*,** and **submit.sh**. **INCAR_1** and **INCAR_2** are the central input files for VASP. The structures will be relaxed by a set of relatively low precision control parameters in **INCAR-1** first; then the coarsely relaxed structures will be further relaxed by a set of relatively high-precision parameters in **INCAR_2**. Users do NOT need to cat **POTCAR** of each atomic species together; the program will do the job automatically. However, one has to copy **POTCAR** for each element to this directory and rename it to "POTCAR-*" format, while the star (*) stands for the name of the element. Taking surface structures constructed by C and H atoms for example, one needs to copy **POTCAR** of C and H to this directory and rename them to **POTCAR-C** and **POTCAR-H,** respectively. For pseudo-hydrogens with fractional valence electrons, such as Z=0.75, its POTCAR should be renamed to **POTCAR-H.75**.

```
# INCAR_1:
PREC   = LOW
EDIFF  = 3e-2
EDIFFG = -4e-2
IBRION = 2
ISIF   = 0
NSW    = 45
ISMEAR = 0 ; SIGMA = 0.05
POTIM  = 0.050
LWAVE  = FALSE
LCHARG = FALSE
ISTART = 0
```

```
# INCAR_2:
ENCUT  = 500
EDIFF  = 1e-4
EDIFFG = 1e-3
IBRION = 2
ISIF   = 0
NSW    = 100
ISMEAR = 0; SIGMA = 0.05
POTIM  = 0.0300
LCHARG = FALSE
LWAVE  = FALSE
ISTART = 0
```

d) submit.sh: It is the script for local relaxation via VASP. Here is an example of submit.sh:

```
#!/bin/sh
mpiexec -n 12 /share/apps/vasp/vasp.5.2 >vasp.log 2>&1
```

Once all the input files are ready, you may run the structure prediction via the following command under unix/linux operating system, or alternatively, put this execution command into a job submision script.

```
$ ./calypso.x >  caly.log &
```

The result analysis process of surface structure prediction is similar to the crystal structure predictions. Please see section 2.3.1 for more information. You may run the following commands:

```
$ cd results
$ cak.py --cif
```

You will see the surface formation enthalpy order of predicted structures in Analysis_Output.dat file, and these newly predicted structure files with cif format can be found in dir_0.1 directory. Please see section 2.6.3 for further information about the result analysis.

3.6.2 Hydrogenated diamond (100) surface reconstruction

Example to perform hydrogenated diamond (100) surface reconstruction prediction using predefined substrate file and DFTB+ local relaxation will be demonstrated in this subsection. Files needed to run this example are listed below:

File/Directory:	Description
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO specifying the controllable key parameters

SUBSTRATE.surf	Substrate file (cif and POSCAR formatted are supported)
DFTB+ input files	dftb_in.hsd_PRE_1, dftb_in.hsd_PRE_2, SK files
submit.sh	A script to run VASP or DFTB+.

a) The **calypso.x** can be downloaded according to different operating systems.

b) input.dat contains all the parameters to run the surface structure predictions. In this example, we show how to control the program to adopt user specified substrate and use DFTB+ code to relax structures locally. Here is an example **input.dat**:

```
LSurface    = T  # Surface reconstruction predictions
ICode       = 8  #DFTB+
Kgrid       = 0.2
PopSize     = 20
MaxStep     = 30
PsoRatio    = 0.6
SurfaceThickness = 2.0
@SurfaceAtoms #|atomic symbol|count|
C    4
H    4
@End
Substrate  = SUBSTRATE.surf
```

c) SUBSTRATE.surf is the user defined substrate files. The surface will be generated on top of this substrate. Both VASP and CIF files can be recognized by CALYPSO. Note that this file is optional. One can set the **Substrate** control parameter in **input.dat** to '***Auto***' or'***Automatic***' in order to generate substrate automatically via crystal info. Please see previous example for more information. For VASP format, chemical symbols should be inserted just before the line of the numbers of atoms.

```
Diamond-100
1.00000
   5.054258   0.000000   0.000000
   0.000000   5.054258   0.000000
   0.000000   0.000000  19.467375
   H   C
   8  24
Selective dynamics
Direct
 0.10783118     0.00000000     0.21796252   F   F   F
 0.10783118     0.50000000     0.21796252   F   F   F
 0.60783118     0.00000000     0.21796252   F   F   F
 0.60783118     0.50000000     0.21796252   F   F   F
 0.39216882     0.00000000     0.21796252   F   F   F
 0.39216882     0.50000000     0.21796252   F   F   F
 0.89216882     0.00000000     0.21796252   F   F   F
 0.89216882     0.50000000     0.21796252   F   F   F
 0.25000000     0.25000000     0.48632006   T   T   T
 …
```

Besides, CIF format is also supported. However, one needs to add _selective tag to specify which atoms are allowed to relax. Please see the following example file:

```
data_Surf
_amcsd_formula_title         "H8C48"
_symmetry_Int_Tables_number   1
_cell_length_a              5.052205
_cell_length_b              5.052205
_cell_length_c             19.824233
_cell_angle_alpha           90.000000
_cell_angle_beta            90.000000
_cell_angle_gamma           90.000000

loop_
_space_group_symop_operation_xyz
x,y,z
loop_
_atom_site_label
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
_selective
C       0.00000000  0.25000000  0.74778343   T
C       0.00000000  0.75000000  0.74778343   T
C       0.50000000  0.25000000  0.74778343   T
C       0.50000000  0.75000000  0.74778343   T
C       0.25000000  0.25000000  0.70273190   T
C       0.25000000  0.75000000  0.70273190   T
C       0.75000000  0.25000000  0.70273190   T
C       0.75000000  0.75000000  0.70273190   T
C       0.25000000  0.50000000  0.65768037   F
C       0.75000000  0.00000000  0.65768037   F
C       0.25000000  0.00000000  0.65768037   F
 …
```

d) DFTB+ input files: hsd formated input files are needed to relax structures: **dftb_in.hsd_PRE_1, dftb_in.hsd_PRE_2**. Control parameters line between #### will be filled automatically via the program and users shouldn't change these values. Besides, users should provide Slater-Koster (SK) files for DFTB+ calculation (Some SK files can be found in this link: http://www.dftb.org/parameters/).

```
Geometry = GenFormat {
 <<<"dftb.gen"
}

Driver = ConjugateGradient {
  ####
  MovedAtoms
  ####
  MaxForceComponent = 1E-3
  MaxSteps = 1000
  OutputPrefix = "geom.out"
}

Hamiltonian = DFTB {
  SCC = Yes
  MaxSCCIterations = 500
  SCCTolerance= 1E-4
  SlaterKosterFiles =
Type2FileNames {
    Prefix = "./"
    Separator = "-"
    Suffix = ".skf"
    LowerCaseTypeName = No
  }


  ####
  MaxAngularMomentum
  ####

  Mixer = Broyden {
    MixingParameter = 0.2
  }
  Filling = Fermi {
    Temperature [Kelvin] =
300.0
  }

  ####
  KPointsAndWeights
  ####
}

Options {}
ParserOptions {
  ParserVersion = 4
}
```

```
Geometry = GenFormat {
 <<<"dftb.gen"
}

Driver = ConjugateGradient {
  ####
  MovedAtoms
  ####
  MaxForceComponent = 1E-4
  MaxSteps = 1000
  OutputPrefix = "geom.out"
}

Hamiltonian = DFTB {
  SCC = Yes
  MaxSCCIterations = 300
  SCCTolerance= 1E-5
  SlaterKosterFiles =
Type2FileNames {
    Prefix = "./"
    Separator = "-"
    Suffix = ".skf"
    LowerCaseTypeName = No
  }


  ####
  MaxAngularMomentum
  ####

  Mixer = Broyden {
    MixingParameter = 0.2
  }
  Filling = Fermi {
    Temperature [Kelvin] =
300.0
  }

  ####
  KPointsAndWeights
  ####
}

Options {}
ParserOptions {
  ParserVersion = 4
}
```

e) '**submit.sh**' is the submission job file for perform the DFTB+ calculation.

Here is an example of submit.sh:

```
#!/bin/sh

export OMP_NUM_THREADS=4

/share/apps/dftb+ >dftb.log 2>&1
```

Once all the input files are ready, you may run the structure prediction via the following command, or alternatively, put this execution command into a job submit script.

```
$ ./calypso.x >  caly.log &
```

The result analysis process of surface structure prediction is similar to the crystal structure predictions. Please see section 2.3.1 for more information. You may run the following commands:

```
$ cd results
$ cak.py --cif
```

You will see the surface formation enthalpy order of predicted structures in Analysis_Output.dat file, and these newly predicted structure files with cif format can be found in dir_0.1 directory. Please see section 2.6.3 for further information about the result analysis.

3.7 Design of Superhard Materials

This section is to show the examples for design of superhard materials using CALYPSO code. Here, VASP code is used for geometry optimization and enthalpy calculations. The CALYPSO input file of '**input.dat**', VASP input files of **INCAR_*** and pseudopotential file of **POTCAR** are needed.

The following files should be present in the working directory:

File:	Description
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO containing controllable key parameters
INCAR_*	Input files for VASP
POTCAR	Pseudopotential for VASP
submit.sh	Job submission file

a) The **calypso.x** can be downloaded according to different operating systems.

b) The following parameters in **input.dat** are shown for the carbon system.

```
####### The Basic Parameters of CALYPSO #######
# A string of one or several words contain a descriptive
# name of the system (max. 40 characters).
SystemName = Carbon
# Number of different atomic species in the simulation.
NumberOfSpecies = 1
# Element symbols of the different chemical species.
NameOfAtoms = C
# Number of atoms for each chemical species in one formula unit.
NumberOfAtoms =  1
```

```
a# The range of formula unit per cell in your simulation.
NumberOfFormula = 4 4
# The volume per formula unit. Unit is in angstrom^3.
Volume=14.0
# Minimal distance between atoms of each chemical species.
@DistanceOfIon
 1.30
@End
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
# The proportion of the structures generated by PSO.
PsoRatio = 0.6
# The population size. Normally, it has a larger number for larger
systems.
PopSize = 30
# It determines which local optimization code should be
# interfaced in the simulation.
ICode= 1
# The Number of local optimization for each structure.
NumberOfLocalOptim= 4
# The precision of the K-point sampling for local optimization
Kgrid = 0.12 0.06
# The command to perform local optimization calculation.
Command = sh submit.sh
# The Max step for iteration
MaxStep =50
############ End Basic Parameters ##########

############ The Parameters for hardness #########
# If True, Hard structure prediction is performed
Hardness= T
```

c) We strongly suggest that you use the multi-stage strategy for structure relaxation. For VASP, the number of **INCAR_*** (INCAR_1, INCAR_2, …) is in accord with the setting tag of `NumberOfLocalOptim`. In this example, we use four INCAR files for structure relaxation. **INCAR_1** and **INCAR_2 are used to** perform very coarse structure relaxation keeping the volume fixed. **INCAR_3** is used to perform full structure relaxation with medium precision, and **INCAR_4** is used to perform very accurate calculations.

```
# INCAR_1:
SYSTEM = local optimization
PREC = LOW
EDIFF = 3e-2
IBRION = 2
ISIF = 2
NSW = 40
ISMEAR = 0 ; SIGMA = 0.05
POTIM = 0.50
LWAVE = FALSE
LCHARG = FALSE
ISTART = 0
PSTRESS = 0.001
EDIFFG = -4e-2
```

```
# INCAR_2:
SYSTEM = local optimization
PREC = Normal
EDIFF = 2e-2
IBRION = 2
ISIF = 4
NSW = 40
ISMEAR = 0
SIGMA = 0.05
POTIM = 0.2
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 0.001
EDIFFG = -1e-2
```

```
# INCAR_3:
SYSTEM = local optimization
PREC = Normal
EDIFF = 2e-4
IBRION = 1
ISIF = 3
NSW = 40
ISMEAR = 0 ; SIGMA = 0.05
POTIM = 0.1
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 0.001
EDIFFG = 1e-3
```

```
# INCAR_4:
SYSTEM = local optimization
ENCUT = 600
EDIFF = 1e-5
IBRION = 2
ISIF = 3
NSW = 80
ISMEAR = 0 ; SIGMA = 0.05
POTIM = 0.1
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 0.001
EDIFFG = 1e-4
```

d) POTCAR is provided by the user. The order of element must be the same with the setting tag of `NameOfAtoms`.

e) '**submit.sh**' is the submission job file for performing the VASP calculations.
Here is an example of submit.sh:

```
#!/bin/sh
mpdboot
mpiexec -n 12 ~/bin/vasp.4.6 > vasp.log 2> /dev/null
```

Once all the input files are ready, you can just type

```
$ ./calypso.x > caly.log &
```

to execute the CALYPSO code in the sequential mode. Or you can write this command into the pbs script and submit it.

CALYPSO will generate the "**results**" folder, which contains all outputs files of CALYPSO. To analyze the results, just type following command:

```
$ cd results
$ cak.py  --cif –hard
```

The index of the structures sorted by hardness in the descending order is presented in Analysis_Output.dat file, and these newly predicted structure files with cif format are shown in dir_0.1 directory. Please see section 2.6.3 for further information about the analysis of CALYPSO results.

3.8 Structure prediction of atom or molecule adsorption of 2D layer material

This section is to show the example for the structure prediction of hydrogenated graphene. Here, DFTB+ code was used for geometry optimization and energy calculations. The CALYPSO input file of '**input.dat**', 2D substrate structure information file of '**SUB.dat**', DFTB+ input files of **dftb_in_*.hsd** and Slater-Koster (SK) files of ***.skf** are needed.

File:	Description
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO containing controllable key parameters
SUB.dat	2D substrate structure information
Dftb_in_*.hsd	Input files for DFTB+
*.skf	SK files for DFTB+
submit.sh	Job submission file

a) The **calypso.x** can be downloaded according to different operating systems.

b) The following parameters in **input.dat** are shown for the hydrogenated graphene system.

```
############ The Basic Parameters of CALYPSO#################
# A string of one or several words contain a descriptive
# name of the system (max. 40 characters).
SystemName = Hydrogenated-graphene
# Number of different atomic species in the simulation.
NumberOfSpecies = 2
# Element symbols of the different chemical species.
NameOfAtoms = C H
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
# The proportion of the structures generated by PSO.
PsoRatio = 0.6
# It determines which local optimization method should be interfaced
#in the simulation.
ICode=  8
# The Number of local optimization for each structure.
NumberOfLocalOptim = 2
# The precision of the K-point sampling for local optimization
Kgrid = 0.1  0.06
# The command to perform local optimization calculation on your
computer.
Command = sh submit.sh
# The population size. Normally, it has a larger number for larger
#systems.
PopSize = 30
# the max step of iteration
MaxStep = 30
#######The Parameters For 2D Material With Adsorption ##########
# If True, a prediction of 2D Material With Adsorption is performed.
Adsorption = T
# It determines which method of generating structures should be
#adopted.
AdsorptionStyle = 2

#number of kinds of adatoms

NumberOfTypeAtom = 1
# If true, the prediction is performed on both sides of 2D materials.
BothSide = T
```

```
# The information of adatom. The specific number:
# Name of atomic element| Numbers of adtoms
@Adatoms
H 2 2
@end
# The number of supercell
@SuperCell
2 0
0 2
@end
# The range of length of bond
RangeOfZAxis = 1.4 1.2

############ End of adsorption prediction ######################
```

c) SUB.dat contains the 2D substrate structure information. Here we take graphene as an example

```
graphene
1.0
   2.4594000    0.0000000    0.0000000
  -1.2297000    2.1298750    0.0000000
   0.0000000    0.0000000   15.0000000
  C
  2
Direct
 0.66667  0.33333  0.50000
 0.33333  0.66667  0.50000
points
 2
 0.66667  0.33333  0.50000
 0.33333  0.66667  0.50000
```

The structure information is given in the format of POSCAR. If **AdsorptionStyle** is set to 2 in **input.dat.** The adsorption site is needed. The line below the word "**points**" gives numbers of sites in the cell. The next few lines give the direct coordinates of adsorption sites.

d) DFTB+ input files: hsd formated input files are needed to relax structures: **dftb_in_1.hsd, dftb_in_2.hsd**. Besides, users should provide Slater-Koster (SK) files for DFTB+ calculation (Some SK files can be found in this link: http://www.dftb.org/parameters/).

```
Geometry = GenFormat {
 <<<"dftb.gen"
}

Driver = ConjugateGradient {
  MaxForceComponent = 1E-3
  MaxSteps = 1000
  OutputPrefix = "geom.out"
  LatticeOpt = Yes
}

Hamiltonian = DFTB {
  SCC = Yes
  MaxSCCIterations = 500
  SCCTolerance= 1E-4
  SlaterKosterFiles =
Type2FileNames {
    Prefix = "./"
    Separator = "-"
    Suffix = ".skf"
    LowerCaseTypeName = No
  }


  MaxAngularMomentum {
    C = p
    H = s
  }

  Mixer = Broyden {
    MixingParameter = 0.2
  }
  Filling = Fermi {
    Temperature [Kelvin] =
300.0
  }

  ####

}

Options {}
ParserOptions {
  ParserVersion = 4
}
#####
```

```
Geometry = GenFormat {
 <<<"dftb.gen"
}

Driver = ConjugateGradient {
  MaxForceComponent = 1E-4
  MaxSteps = 1000
  OutputPrefix = "geom.out"
  LatticeOpt = Yes
}

Hamiltonian = DFTB {
  SCC = Yes
  MaxSCCIterations = 300
  SCCTolerance= 1E-5
  SlaterKosterFiles =
Type2FileNames {
    Prefix = "./"
    Separator = "-"
    Suffix = ".skf"
    LowerCaseTypeName = No
  }


  MaxAngularMomentum {
    C = p
    H = s
  }

  Mixer = Broyden {
    MixingParameter = 0.2
  }
  Filling = Fermi {
    Temperature [Kelvin] =
300.0
  }

  ####

}

Options {}
ParserOptions {
  ParserVersion = 4
}
#####
```

e) '**submit.sh**' is the submission job file for perform the DFTB+ calculation.

Here is an example of submit.sh:

```
#!/bin/sh

export OMP_NUM_THREADS=4

/share/apps/dftb+ >dftb.log 2>&1
```

Once all the input files are ready, you may run the structure prediction via the following command, or alternatively, put this execution command into a job submit script.

```
$ ./calypso.x >  caly.log &
```

The result analysis process of surface structure prediction is similar to the crystal structure predictions. Please see section 2.3.1 for more information. You may run the following

commands:

```
$ cd results
$ cak.py --cif
```

You will see the surface formation enthalpy order of predicted structures in Analysis_Output.dat file, and these newly predicted structure files with cif format can be found in dir_0.1 directory. Please see section 2.6.3 for further information about the result analysis.

3.9 Design of Optical Materials with Desirable Electronic Band Gap

This section is to show the examples for design of optical materials with desirable electronic band gap using CALYPSO code. We take the structure prediction of carbon (C) as an example. Here, VASP code was used for geometry optimization and enthalpy calculations. The CALYPSO input file of '**input.dat**' and VASP input files of **INCAR_*** and pseudopotential file of **POTCAR** are needed.

The following files should be present in the working directory:

Files	Description
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO containing controllable key parameters
INCAR_*	Input files for VASP
POTCAR	Pseudopotential for VASP
submit.sh	Job submission file

a) The **calypso.x** can be downloaded according to different operating systems.

b) The parameters in **input.dat** are shown below.

```
########### The Parameters of CALYPSO #############
# A string of one or several words contain a descriptive
# name of the system (max. 40 characters).
SystemName = C
# Number of different atomic species in the simulation,
# where it has one type elements (C).
NumberOfSpecies = 1
# Element symbols of the different chemical species.
NameOfAtoms = C
# Number of atoms for each chemical species in one formula unit.
NumberOfAtoms = 1
#If True, the prediction structure with target band gap is performed
Band_edge = T
#The required band gap
TarBandGap = 2.5
```

```
# The range of formula unit per cell in your simulation.
NumberOfFormula = 4  4
# The volume of 1 f.u. unit=angstrom^3.
Volume = 15.0
# Minimal distances between atoms of each chemical species.
@DistanceOfIon
 1.0
 1.0
@End
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
# The proportion of the structures generated by PSO
# against the new structures in each generation.
PsoRatio = 0.6
# The population size. Normally, it is larger for larger systems.
PopSize = 30
# It determines which local optimization program
# should be used in the simulation.
ICode = 1
# The Number of local optimization for each structure.
NumberOfLocalOptim = 4
# It is the k-point sampling solution in local optimization.
Kgrid = 0.12 0.06 0.04
# The script for submitting local optimization calculations.
Command = sh submit.sh
# The Max step for iteration
MaxStep =30
#################### End Parameters ######################
```

c) We strongly suggest that you use the multi-stage strategy for structure relaxation. For VASP, the number of **INCAR_*** (INCAR_1, INCAR_2, …) is one larger than the setting tag `NumberOfLocalOptim`. In this example, we use four INCAR files for structure relaxation and one INCAR files for calculating band gap. **INCAR_1** and **INCAR_2** are used to perform very coarse structure relaxation keeping the volume fixed, **INCAR_3** is used to perform full structure relaxation with medium precision, **INCAR_4** is used to perform very accurate calculations, and **INCAR_META** is used to perform very accurate band gap calculations.

```
# INCAR_1:
SYSTEM = local optimization
PREC = LOW
EDIFF = 3e-2
IBRION = 2
ISIF = 2
NSW = 40
ISMEAR = 0 ; SIGMA = 0.05
POTIM = 0.50
LWAVE = FALSE
LCHARG = FALSE
ISTART = 0
PSTRESS = 0.001
EDIFFG = -4e-2
```

```
# INCAR_2:
SYSTEM = local optimization
PREC = Normal
EDIFF = 2e-2
IBRION = 2
ISIF = 4
NSW = 40
ISMEAR = 0
SIGMA = 0.05
POTIM = 0.2
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 0.001
EDIFFG = -4e-1
```

```
# INCAR_3:
SYSTEM = local optimization
PREC = Normal
EDIFF = 2e-4
IBRION = 1
ISIF = 3
NSW = 40
ISMEAR = 0 ; SIGMA = 0.05
POTIM = 0.1
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 0.001
EDIFFG = 2e-3
```

```
# INCAR_4:
SYSTEM = local optimization
ENCUT = 400
EDIFF = 1e-5
IBRION = 2
ISIF = 3
NSW = 80
ISMEAR = 0 ; SIGMA = 0.05
POTIM = 0.1
LWAVE = FALSE
LCHARG = FALSE
PSTRESS = 0.001
EDIFFG = 2e-3
```

```
# INCAR_META:
SYSTEM = META
PREC = Accurate
ENCUT = 400
EDIFF = 1e-6
IBRION = -1
ISMEAR = -5 ; SIGMA = 0.05
METAGGA = MBJ
LASPH = .TRUE.
LMINTAU = .TURE.
NELM = 80
NEDOS = 800
```

d) POTCAR should be provided by the user. The order of element in POTCAR must be the same with the setting tag of `NameOfAtoms`. Meta-GGA calculations require POTCAR files containing information on the kinetic energy density of the core-electrons. To check the pseudopotential file (POTCAR), please type:

```
grep kinetic POTCAR
```

This should yield at least the following lines (for each element on the file):

```
kinetic energy-density
mkinetic energy-density pseudized
```

and for PAW datasets with partial core corrections:

```
kinetic energy density (partial)
```

e) '**submit.sh**' is the job submission file for performing the VASP calculations and we must use the vasp.5.* (vasp.4* version is not support META-GGA).
Here is an example of submit.sh:

```
#!/bin/sh
mpdboot
mpiexec -n 12 ~/bin/vasp.5.2.2 > vasp.log 2> /dev/null
```

Once all the input files are ready, you can simply type

```
$ ./calypso.x >  caly.log &
```

to execute the CALYPSO in the sequential mode. Or you can write this command into the pbs script and submit it.

CALYPSO will generate the "**results**" folder in current directory, which contains all most the outputs files of CALYPSO. To analyze the results, just type following command:

```
$ cd results
$ cak.py --cif
```

The index of the structures sorted by band gap in the ascending order is presented in Analysis_Output.dat file, and these newly predicted structure files with cif format are shown in **dir_0.1** directory. Please see section 2.6.3 for further information about the analysis of CALYPSO results.

3.10 Crystal Structure Prediction with Fixed Cell Parameters or Atomic Positions

This section is to show the examples for the three-dimensional crystal structure prediction with fixed cell parameters or partial atomic positions using CALYPSO code. Here, VASP code is used for geometry optimization and enthalpy calculations. The CALYPSO input file of '**input.dat**' and VASP input files of **INCAR_*** and pseudopotential file of **POTCAR** are needed.

The following files should be presented in the working directory:

File:	Description
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO containing controllable key parameters
cell.dat	The input file contains fixed cell parameters and atomic positions
INCAR_*	Input files for VASP
POTCAR	Pseudopotential for VASP
submit.sh	submission job file

a) The **calypso.x** can be downloaded according to different operating systems.

b) The parameters in **input.dat** are shown as below.

```
######## The Basic Parameters of CALYPSO #######
SystemName = SiC
# Number of different atomic species in the simulation.
NumberOfSpecies = 2
# Element symbols of the different chemical species.
NameOfAtoms =  C Si
# Number of atoms for each chemical species in one formula unit.
NumberOfAtoms = 4 4
# The range of formula unit per cell in your simulation.
NumberOfFormula = 1 1
```

```
# Minimal distance between atoms of each chemical species. Unit is in
angstrom.
@DistanceOfIon
 1.2 1.4
 1.4 1.4
@End
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
# The proportion of the structures generated by PSO.
PsoRatio = 0.6
PopSize = 5
ICode= 1
NumberOfLbest=4
# The Number of local optimization for each structure.
NumberOfLocalOptim= 3
# The precision of the K-point sampling for local optimization
Kgrid = 0.12 0.06
Command = sh submit.sh
# If True, the fixed the  cell parameters
FixCell= T
# If True, the fixed the partial atomic positions
FixAtom= T
```

c) The **cell.dat** file contains the fixed information of cell parameters and partial atomic positions. In this example, the atomic positions of C are fixed. The first three lines indicate the lattice matrix. The forth line is the number of atoms to be fixed followed with the fixed coordinates of atomic positions.

```
4.4   0.0   0.0
0.0   4.4   0.0
0.0   0.0   4.4
4   0
0.0   0.0   0.0
0.5   0.5   0.0
0.5   0.0   0.5
0.0   0.5   0.5
```

d) We strongly suggest that you use the multi-stage strategy for structure relaxation. For VASP, the number of **INCAR_*** (INCAR_1, INCAR_2, …) is in accord with the setting tag of `NumberOfLocalOptim`. In this example, we use three INCAR files for structure

relaxation. **INCAR_1** and **INCAR_2** are used to perform very crude structural relaxations with fixed volume, and **INCAR_3** is used to perform full structure relaxation with accurate precision.

```
# INCAR_1:
SYSTEM =
optimisation
PREC = LOW
EDIFF = 3e-2
IBRION = 2
SYMPREC=1e-3
ISIF = 2
NSW = 45
ISMEAR = 1
SIGMA = 0.2
POTIM = 0.050
PSTRESS = 0.01
EDIFFG = -4e-2
```

```
# INCAR_2:
SYSTEM            =
optimisation
PREC = Normal
EDIFF = 2e-2
IBRION = 2
SYMPREC=1e-3
ISIF = 2
NSW = 55
ISMEAR = 1
SIGMA = 0.2
POTIM = 0.100
PSTRESS = 0.01
EDIFFG = -4e-1
```

```
# INCAR_3:
SYSTEM =
optimisation
PREC = accurate
ENCUT = 520.0
EDIFF = 1e-5
IBRION = 2
ISIF = 3
NSW = 60
ISMEAR = 1
SIGMA = 0.2
POTIM = 0.05
PSTRESS = 0.01
EDIFFG = 1e-3
```

e) POTCAR should be provided by the user. The order of element of POTCAR must be the same with the setting tag of `NameOfAtoms`.

f) '**submit.sh**' is the job submission file for VASP calculations.
Here is an example of submit.sh:

```
#!/bin/sh
mpdboot
mpiexec -n 12 ~/bin/vasp.4.6 > vasp.log 2> /dev/null
```

Once all the input files are ready, you can simply type

```
$ ./calypso.x > caly.log &
```

to execute the CALYPSO code in the sequential mode. Or you can write this command into the pbs script and submit it.

CALYPSO will generate the "**results**" folder, which contains all the output files. To analyze the CALYPSO results, just type following command:

```
$ cd results
$ cak.py  --cif
```

The index of the structures sorted by enthalpies in the ascending order is presented in Analysis_Output.dat file, and these newly predicted structure files with cif format are shown in **dir_0.1** directory. Please see section 2.6.3 for further information about the analysis of CALYPSO results.

3.11 Structural Prediction via X-ray Diffraction Data

This section is to show the example for the structural prediction of ZnO via X-ray diffraction data using CALYPSO code. Here, VASP code was used for geometry optimization and enthalpy calculations. The CALYPSO input file of '**input.dat**' and VASP input files of **INCAR_*** and pseudopotential file of **POTCAR** are needed.

The following files should be present in the working directory:

Files	**Description**
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO containing controllable key parameters
INCAR_*	Input files for VASP
POTCAR	Pseudopotential file for VASP
submit.sh	Job submission file
XRD.data	The experimental XRD data

a) The **calypso.x** can be downloaded according to different operating systems.

b) The parameters in the **input.dat** are shown below.

```
########### The Parameters of CALYPSO #############
# A string of one or several words contain a descriptive
# name of the system (max. 40 characters).
SystemName = ZnO
# Number of different atomic species in the simulation,
# where it has two type elements (B and N).
NumberOfSpecies = 2
# Element symbols of the different chemical species.
NameOfAtoms =Zn  O
# Number of atoms for each chemical species in one formula unit.
NumberOfAtoms = 1 1
```

```
# The range of formula unit per cell in your simulation.
NumberOfFormula = 2 2
# The volume of 1 f.u. unit=angstrom^3.
Volume = 25.0
# Minimal distances between atoms of each chemical species.
@DistanceOfIon
 1.0  1.0
 1.0  1.0
@End
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
# The proportion of the structures generated by PSO
# against the new structures in each.
PsoRatio = 0.6
# The population size. Normally, it is larger for larger systems.
PopSize = 30
# It determines which local optimization program
# should be used in the simulation.
ICode = 1
# The Number of local optimization for each structure.
NumberOfLocalOptim = 3
# It is the k-point sampling solution in local optimization.
Kgrid = 0.1 0.07
# The script for submitting local optimization calculations.
Command = sh submit.sh
# The Max step for iteration
MaxStep =30
#If True, a structure prediction via XRD is performed.
LXRD= T
# The wavelength of X-ray
WaveLength=1.54056
# The step of the experimental XRD data.
StepOf2Theta= 0.02
# The 2θ ranges of the experimental XRD data.
RangeOf2Theta= 20.0 80.0
# The position of standard peak of experimental XRD data.
StandardPeakPosition= 35.8 36.8
#################### End Parameters ######################
```

c) The **XRD.dat** file contains the information of the experimental XRD data. The values of the first row and second row are 2θ and intensities of XRD, respectively.

```
20.00      38.3014
20.02      38.3014
20.04      38.3014
……
```

d) We strongly suggest that you use the multi-optimization strategy for structure relaxation. For VASP, the number of **INCAR_*** (INCAR_1, INCAR_2, …) files is in accordance with the setting tag **NumberOfLocalOptim**. In this example, four INCAR files are used for structure relaxation. **INCAR_1** and **INCAR_2** files are used to perform very coarse structure relaxation with the fixed volume, and **INCAR_3** file is used to perform full structure relaxation, respectively.

```
# INCAR_1:
SYSTEM =
optimisation
PREC = LOW
EDIFF = 1e-2
IBRION = 2
ISIF = 2
NSW = 45
ISMEAR = 0
SIGMA = 0.05
KSPACING =0.5
POTIM = 0.050
PSTRESS = 0.001
EDIFFG = -4e-1
```

```
# INCAR_2:
SYSTEM            =
optimisation
PREC = Normal
EDIFF = 2e-3
IBRION = 2
ISIF = 4
NSW = 55
ISMEAR = 0
SIGMA = 0.05
KSPACING =0.5
POTIM = 0.100
PSTRESS = 0.001
EDIFFG = -4e-1
```

```
# INCAR_3:
SYSTEM =
optimisation
PREC = accurate
ENCUT = 520.0
EDIFF = 1e-4
IBRION = 2
ISIF = 3
NSW = 200
ISMEAR = 0
SIGMA = 0.05
KSPACING =0.2
POTIM = 0.05
PSTRESS = 0.01
```

e) POTCAR should be provided. **ATTENTION!!!** The order of elements in POTCAR must be identical to the element order in the setting tag of **`NameOfAtoms`**.

f) '**submit.sh**' is the job submission file for the VASP calculations.
Here is an example of submit.sh:

```
#!/bin/sh
mpdboot
mpiexec -n 12 ~/bin/vasp.4.6 > vasp.log 2> /dev/null
```

Once all input files are ready, you can simply type

```
$ ./calypso.x > caly.log &
```

to execute the CALYPSO in the sequential mode. Or you can write this command into the **pbs** script and submit it in a queue system.

CALYPSO run will generate the "**results**" folder in the current directory you have specified. The **"results"** folder contains all the outputs files of CALYPSO. To analyze the results, please simply type the following commands:

```
$ cd results
$ cak.py  --cif
```

The index of the structures sorted by enthalpies in the ascending order will be presented in the Analysis_Output.dat file, and all predicted structure files by CALYPSO with cif formats are shown in the **dir_0.1** directory. Please see section 2.6.3 for more information about the analyses of CALYPSO results.

3.12 Prediction of Transition States in Solids

This section is to show the example for the prediction of transition state of diamond to β-Sn of Si using CALYPSO code. Here, VASP code was used for enthalpy calculations. The CALYPSO input file of '**input.dat**' and VASP input files of **INCAR_1** and pseudopotential file of **POTCAR** are needed.

The following files should be present in the working directory:

Files	**Description**
calypso.x	The executable file for running CALYPSO program
input.dat	The input file of CALYPSO containing controllable key parameters
INCAR_1	Input files for VASP
POTCAR	Pseudopotential file for VASP
submit.sh	Job submission file
IF_struct.dat	The structures of initial and final states

a) The **calypso.x** can be downloaded according to different operating systems.

b) The parameters in the **input.dat** are shown below.

```
########### The Parameters of CALYPSO #############
# A string of one or several words contain a descriptive
# name of the system (max. 40 characters).
SystemName = Si
# Number of different atomic species in the simulation,
NumberOfSpecies = 1
# Element symbols of the different chemical species.
NameOfAtoms =Si
# Number of atoms for each chemical species in one formula unit.
NumberOfAtoms =8
# The range of formula unit per cell in your simulation.
NumberOfFormula = 1.0 1.0
# The volume of 1 f.u. unit=angstrom^3.
Volume = 35.0
# Minimal distances between atoms of each chemical species.
@DistanceOfIon
 1.0
@End
# It determines which algorithm should be adopted in the simulation.
Ialgo = 2
# The proportion of the structures generated by PSO
PsoRatio = 0.6
# The population size. Normally, it is larger for larger systems.
PopSize = 6
# It determines which local optimization program
ICode = 1
# The Number of local optimization for each structure.
NumberOfLocalOptim = 1
# It is the k-point sampling solution in local optimization.
Kgrid = 0.2 0.2
# The script for submitting local optimization calculations.
Command = sh submit.sh
```

```
The Max step for iteration
MaxStep =30
#If True, prediction of Transition State in Solids is performed.
LTranState= T
# The number of images for each trial path.
NumberOfImages=3
#################### End Parameters ####################
```

c) The **IF_struct.dat** file contains the structural information of initial and final states. Here we take phase transition of diamond to β-Sn in Si as an example. The structural information of initial state is shown in red following the structure of final state. The format of each structure in IF_struct.dat file is the same with that used in VASP code. Specially, the first line is treated as a comment line. The second line provides a universal scaling factor to scale all lattice vectors (the value is set to 1.0 in our module). On the following three lines the three lattice vectors defining the unit cell of the system are given (first line corresponding to the first lattice vector, second to the second, and third to the third). The sixth line supplies the number of atoms per atomic species (one number for each atomic species). The ordering must be consistent with the POTCAR. The seventh line (Direct) specifies the atomic position provided in fractional coordinates. The next lines give the three coordinates for each atom.

```
Si-initial state
   1.00000000000000
     5.466680928324514  0.0000000000000000   0.0000000000000000
     0.0000000000000000 5.4666809283245144   0.0000000000000000
     0.0000000000000000 0.0000000000000000   5.4666809283245144
     8
Direct
  0.2500000000000000  0.2500000000000000  0.2500000000000000
  0.7500000000000000  0.7500000000000000  0.2500000000000000
  0.7500000000000000  0.2500000000000000  0.7500000000000000
  0.2500000000000000  0.7500000000000000  0.7500000000000000
  0.0000000000000000  0.0000000000000000  0.0000000000000000
  0.0000000000000000  0.5000000000000000  0.5000000000000000
  0.5000000000000000  0.0000000000000000  0.5000000000000000
  0.5000000000000000  0.5000000000000000  0.0000000000000000
```

```
Si-final state
   1.00000000000000
     4.685999999999999 0.000000000000000   0.000000000000000
     0.000000000000000 4.696859999999999   0.000000000000000
     0.000000000000000 0.000000000000000   5.169999999999999
   8
Direct
  0.0000000000000000  0.5000000000000000  0.1250000000000000
  0.5000000000000000  0.5000000000000000  0.2500000000000000
  0.0000000000000000  0.0000000000000000  0.0000000000000000
  0.5000000000000000  0.0000000000000000  0.8750000000000000
  0.5000000000000000  0.5000000000000000  0.7500000000000000
  0.0000000000000000  0.0000000000000000  0.5000000000000000
  0.5000000000000000  0.0000000000000000  0.3750000000000000
  0.0000000000000000  0.5000000000000000  0.6250000000000000
```

c) Currently, VASP code is used to structure relaxation. In this example, INCAR file is used for structure relaxation.

```
# INCAR_1:
SYSTEM = scf
PREC = Accurate
NSW=1
IBRION=-1
EDIFF = 1e-3
ISMEAR = 1
SIGMA = 0.2
PSTRESS = 0.001
```

d) POTCAR should be provided. **ATTENTION!!!** The order of elements in POTCAR must be identical to the element order in the setting tag of **`NameOfAtoms`**.

e) '**submit.sh**' is the job submission file for the VASP calculations.
Here is an example of submit.sh:

```
#!/bin/sh
mpdboot
mpiexec -n 12 ~/bin/vasp.4.6 > vasp.log 2> /dev/null
```

Once all input files are ready, you can simply type

```
$ ./calypso.x > caly.log &
```

to execute the CALYPSO in the sequential mode. Or you can write this command into the **pbs** script and submit it in a queue system.

CALYPSO run will generate the "**results**" folder in the current directory you have specified. The **"results"** folder contains all the outputs files of CALYPSO. To analyze the results, please simply type the following commands:

```
$ cd results
$ cak.py  --cif
```

The index of the structures sorted by enthalpies in the ascending order will be presented in the Analysis_Output.dat file, and all predicted structure files by CALYPSO with cif formats are shown in the **dir_0.1** directory. Please see section 2.6.3 for more information about the analyses of CALYPSO results.

4. Special Topic

4.1 The Parallel mode

The parallel mode of CALYPSO enables the geometrical optimization of structures through desirable number of CPUs. The parallel mode of CALYPSO supports torque **pbs** system.

1. The keywords “Parallel” and “NumberOfParallel” are needed for using the parallel mode.
   For example, you can add ‘Parallel = T’ and ‘NumberOfParallel = 4’ to the input.dat. NumberOfParallel defines the number of structure relaxations you want to run in parallel.

2. Modify the **pbs** script for generating the machinefile. The machinefile is a file that contains a list of the possible machines on which you want your MPI program to run.
   For example, you should add `‘cat $PBS_NODEFILE > machinefile’` before the line of executable calypso.x (usually `‘./calypso.x > caly.log’`).

3. Modify the script for vasp execution.
   For example, you can write the submit.sh as below:
   ```
   mpiexec –machine snodefile –n 12 vasp > out.vasp 2 > /dev/null
   ```

4.2 Remote Submission

The remote submission mode will allow the calypso running on a local machine while the VASP is running on the remote cluster. This feature is yet under testing on Torque PBS system at present.

Here is the workflow:

1. You should set the keywords 'RemoteParallel = T' to enable the remote submission mode. And you can define the number of structures you want to relax at the same time by setting the keyword 'NumberOfParallel' which just like in the parallel mode.

2. Modify the submitremote.sh file.

```
server='test@10.60.36.168'
port='22'
```

Please modify the 'server' and the 'port' to adapt your cluster.

3. To set up passwordless connection from local machine to remote cluster, you need to copy the public key from you local machine (directory ~/.ssh/) to the remote cluster. Here is the list of commands you need to execute.
On the local machine:

```
[user@local]$ ssh-keygen
[user@local]$scp ~/.ssh/id_rsa.pub test@cluster.calypso.cn:~/ssh/local.pub
```

On the remote cluster:

```
[test@cluser]$ cd ~/.ssh/
[test@cluser]$ cat local.pub >> authorized_keys
[test@cluser]$ rm local.pub
```

4. Prepare the pbs script 'vasp.pbs' to calculate the vasp. Here is an example.

```
cat vasp.pbs

#!/bin/bash
#PBS -l nodes=1:ppn=6
#PBS -j oe
```

```
#PBS -V
cd $PBS_O_WORKDIR
if [ ! -f ~/.mpd.conf ]; then
/bin/echo "secretword=dfadfs" >> ~/.mpd.conf
/bin/chmod 600 ~/.mpd.conf
fi
# Intel MPI Home
MPI_HOME=/opt/intel/impi/4.0.0.027
# setup Nums of Processor
NP=`cat $PBS_NODEFILE|wc -l`
echo "Numbers of Processors:  $NP"
echo "---------------------------"
# Number of MPD
N_MPD=`cat $PBS_NODEFILE|uniq|wc -l`
echo "started mpd Number: $N_MPD"
echo "---------------------------"
# setup mpi env (em64t)
$MPI_HOME/bin64/mpdboot -r ssh -n $N_MPD -f $PBS_NODEFILE

# running program
for i in 1 2 3
do
cp INCAR_$i INCAR
python writekp.py 0.1
mpiexec -n $NP /share/apps/vasp/vasp.4.6 > log 2 > /dev/null
cp CONTCAR POSCAR
done
cp INCAR_4 INCAR
python writekp.py 0.07
mpiexec -n $NP /share/apps/vasp/vasp.4.6 > log 2 > /dev/null
```

Note: the blue digits can be modified to control the precision of the *k*-mesh sampling for local optimization.

5. When you have prepared the necessary files (such as POTCAR, INCAR_*, vasp.pbs, submitremote.sh), you can execute the calypso.x on your local machine.

4.3 The split mode

The split mode of CALYPSO can provide a more flexible way to relax the structures.

The structural files with POSCAR format (POSCAR_*) are generated by calypso.x. You can relax these structures on any computer clusters or workstations.

Here is the workflow:

1. Enabling the split mode by setting 'Split = T' in the input.dat.
2. Executing the calypso.x to generate the structural files (in POSCAR format) of 'POSCAR_1, POSCAR_2, …' for one particular generation.
3. After optimizing all these structures one by one on all possible computer clusters, please copy the OUTCAR and CONTCAR to "OUTCAR_1 and CONTCAR_1", or "OUTCAR_2 and CONTCAR_2" … and then moves all these files to the directory where you executed calypso.x.
4. Executing calypso.x again to generate the POSCAR_* files for the next generation.
5. To repeat the steps 2-4 until the halting criterion is reached.

5. Selected Publications

1. Reactions of xenon with iron and nickel are predicted in the Earth's inner core

   Nature Chem. 6, 644 (2014)

2. Caesium in high oxidation states and as a p-block element

   Nature Chem. 5, 846 (2013)

3. Semi-metallic Be5C2 monolayer global minimum with quasi-planar pentacoordinate carbons and negative Poisson's ratio

   Nature Commun. 7, 11488 (2016)

4. Self-assembled ultrathin nanotubes on diamond (100) surface

   Nature Commun. 5, 3666 (2014)

5. Pressure-stabilized lithium caesides with caesium anions beyond the -1 state

   Nature Commun. 5, 4861 (2014)

6. High Pressure Partially Ionic Phase of Water Ice,

   Nature Commun. 2, 563 (2011)

7. Tellurium Hydrides at High Pressures: High-Temperature Superconductors

   Phys. Rev. Lett. 116, 057002 (2016)

8. High-energy density and superhard nitrogen-rich B-N compounds

   Phys. Rev. Lett. 115, 105502 (2015)

9. Metallic Icosahedron Phase of Sodium at Terapascal Pressures

   Phys. Rev. Lett. 114, 125501 (2015)

10. Superhard BC3 in Cubic Diamond Structure

    Phys. Rev. Lett. 114, 015502 (2015)

11. High-Pressure Hydrogen Sulfide from First Principles: A Strongly Anharmonic Phonon-Mediated Superconductor

    Phys. Rev. Lett. 114, 157004 (2015)

12. Stacking Principle and Magic Sizes of Transition Metal Nanoclusters Based on Generalized Wulff Construction

    Phys. Rev. Lett. 111, 115501 (2013)

13. Global structural optimization of tungsten borides

    Phys. Rev. Lett. 110, 136403 (2013)

14. Towards direct-gap silicon phases by the inverse band structure design approach

    Phys. Rev. Lett. 110, 118702 (2013)

15. Cagelike Diamondoid Nitrogen at High Pressures

Phys. Rev. Lett. 109, 175502 (2012)

16. Predicted novel high-pressure phases of lithium

Phys. Rev. Lett.106, 015503 (2011)

17. Substitutional Alloy of Bi and Te at High Pressure

Phys. Rev. Lett. 106, 145501 (2011)

18. Novel Superhard Carbon: C-Centered Orthorhombic C8

Phys. Rev. Lett. 107, 215502 (2011)

19. Novel Superhard Carbon: C-Centered Orthorhombic C8

Phys. Rev. Lett. 107, 215502 (2011)

20. Spiral Chain O4 Form of Dense Oxygen

Proc. Natl. Acad. Sci. USA 109, 751 (2012)

21. Gold as a 6p-Element in Dense Lithium Aurides

J. Am. Chem. Soc. 138, 4046 (2016)

22. FeB6 Monolayers: The Graphene-like Material with Hypercoordinate Transition Metal

J. Am. Chem. Soc. 138, 5644 (2016)

23. Anionic Chemistry of Noble Gases: Formation of Mg-NG (NG = Xe, Kr, Ar) Compounds under Pressure

J. Am. Chem. Soc. 137, 14122 (2015)

24. Two-Dimensional Cu2Si Monolayer with Planar Hexacoordinate Copper and Silicon Bonding

J. Am. Chem. Soc. 137, 2757 (2015)

25. Direct Band Gap Silicon Allotropes

J. Am. Chem. Soc. 136, 9826 (2014)

26. Structural Evolution of Carbon Dioxide under High Pressure

J. Am. Chem. Soc. 135, 14167 (2013)

27. Tetragonal Allotrope of Group 14 Elements

J. Am. Chem. Soc. 134, 12362 (2012)

28. Predicted lithium-boron compounds under high pressure

J. Am. Chem. Soc. 134, 18699 (2012)

29. Predicting Two-Dimensional Boron–Carbon Compounds by the Global Optimization Method

J. Am. Chem. Soc. 133, 16285 (2011)

6. Acknowledgements

We gratefully acknowledge the suggestions of Hongjun Xiang, Yunlong Liao, Sabino Maggi and Idoia G. de Gurtubay and the support of all the CALYPSO-ers.