A User Guide for *SLUSCHI*

Solid and Liquid in Ultra Small Coexistence with Hovering Interfaces

1 Introduction

SLUSCHI is a fully automated code which calculates melting points based on first-principles molecular dynamics simulations, with interface to the first-principles code VASP. Starting from the crystal structure of a solid (which the user inputs), *SLUSCHI* will automatically build a supercell of a proper size, prepare solid-liquid coexistence, and then employ the small-cell coexistence method to calculate the melting temperature. *SLUSCHI* is applicable to a wide variety of materials, thanks to the fact that density functional theory calculations are highly generalizable.

Please refer to Section 6 for more details.

2 Installation

With `sluschiX_XX.tar.gz` in the current directory, type

```
gunzip sluschiX_XX.tar.gz
tar -xvf sluschiX_XX.tar
```

where `X_XX` is the current version number. These commands create a directory called `sluschi`, which contains the whole package. For future reference, I'll call the whole access path to this directory `sluschi`.

Type

```
cd sluschi
```

and this is the main directory where *SLUSCHI* will be running. To install, type

```
cd src
make
```

This will compile the `.f90` files and generate the corresponding `.x` executables.

`sluschi` is the directory where the code will be running. Change the input files to specify the system of interest. To launch another melting point calculation, simply duplicate `sluschi` and modify the input files accordingly.

3 Input files

Files colored in red are required.

- Standard VASP Input files
  - `INCAR` is the file that run VASP MD. Please note that the following tags are deactivated.
    - ISMEAR is automatically set to -1 to enforce Fermi distribution.
    - SIGMA is set to the value k_BT to reflect the electronic temperature.
    - NSW is set to 80. The isobaric-isothermal ensemble (NPT) is realized by adjustment of cell shape every 80 MD steps of canonical ensemble (NVT). This strategy helps stabilize cell shape, compared to normal barostat.
    - SMASS is set to 0, which determines the thermomass. With this setting, the interaction between the system and the thermal bath oscillates at a period of 40 MD steps.
    - TEBEG and TEEND are set to the sample temperature.
    - NBANDS and POTIM are adjusted based on previous MD results (user may turn it off by setting `adj_potim=0`).

- POSCAR provides the unit-cell atomic structure of the solid. We recommend using the primitive unit-cell to maximize performance.
- POTCAR must be in accordance with the `POSCAR`.
- `KPOINTS` is not required. The k-mesh is specified by the `kmesh` tag in the `job.in` file and generated automatically.
- jobsub is a job script that runs VASP on a cluster. For example, VASP jobs are submitted by

  `qsub jobsub` or `sbatch jobsub`

  The user needs to replace this file with his/her own VASP job script. Please make sure that the wall time limit is long enough (typically 2 hours on 32 cores, but it may vary).
- `jobsub_gamma` (optional) is the Γ-point version of the VASP job script. In many cases, especially for insulators, running VASP with the Γ-point along is already accurate enough to capture the melting point. Under such circumstances, providing this file saves the computer cost by a factor up to 50%.
- `KPOINTS_gamma` (optional) is employed to melt the solid and prepare solid-liquid coexisting configurations. Since this process does not require high accuracy, running Γ-point version VASP (if `jobsub_gamma` is also provided) saves computer cost.

- job.in The tags in `job.in` are described in detail below. Important tags are colored in red.
  - temp: Take a guess at the melting point (in Kelvin). This temperature is used to calculate thermal expansion, and it is usually the lower boundary in the melting point search. Avoid entering a value that is much higher than the true melting point. If you have no clue what the melting point is, enter 500.
  - press: Enter pressure in unit of kbar. For ambient atmospheric pressure, enter 0.
  - confident: If you are confident about the melting point you entered, set it to 1. Otherwise, 0. With `confident=1`, the code will employ a smaller step (100 instead of 800 K) in temperature search.
  - error: Tell *SLUSCHI* how accurate you want the melting point calculation to be. *SLUSCHI* will continue sampling until the standard error is smaller than this value, based on its melting point fitting. Reasonable values for `error` range from 30 to 100 (in Kelvin). Avoid entering small numbers, which will render computer cost skyrocket.
  - kmesh: *SLUSCHI* adopts the automatic k-mesh generation in VASP (see http://cms.mpi.univie.ac.at/vasp/vasp/Automatic_k_mesh_generation.html). *SLUSCHI* will generate a `KPOINTS` file as

    ```
    Automatic mesh
    0
    Auto
    $kmesh
    ```

    The lower `kmesh`, the less dense the k-points and the lower the computer cost. Typically, set `kmesh` to 10-20 for insulators and `kmesh` to 30-40 for metals. This is generally safe to guarantee convergence in energies and forces. Higher temperatures usually allow smaller values of `kmesh`, because the broader electronic distribution (due to the high electronic temperature) smooths out the states in the Brillouin zone.
  - radius: When *SLUSCHI* builds a supercell from the solid structure of the input, this tag determines the distance between the periodic images and thus the size of the supercell. Based on the unit-cell provided, the code will find a supersell approximately $2a \times a \times a$ with angles 60°, 90° and 90° respectively, where a is specified by `radius` (in Å) and the solid-liquid interface is perpendicular to the first dimension. This cell shape theoretically minimizes the interaction among periodic images and hence the finite-size effect.

    We recommend use `radius=10` (Å) so that the finite-size effect is small: a smaller `radius` may lead to a considerable error. Neither do we recommend set `radius` larger than 10, as it may significantly increase computer cost.
  - path: This tag tells *SLUSCHI* where the executables are. Simply set it to the full access path of `sluschi/src/`. Do not miss the red part.

- `vaspcmd`: This is the command for job submission, e.g., `qsub` or `sbatch` typically.
- `navg`: This is a parameter for the barostat and it specifies the number of MD steps (here 80×`navg` steps), from which the average pressure is calculated. Default value is 3.
- `factor`: This is another parameter for the barostat and it adjusts the cell shape in response to the average pressure. In detail,

$$L' = (I + P/f)L, \qquad L = \left(\vec{a}, \vec{b}, \vec{c}\right),$$

  where L and L' are the old and new lattice vector matrices ($\vec{a}, \vec{b}, \vec{c}$ are the three 3×1 lattice vectors), I is an 3×3 identity matrix, P is average stress tensor (in kbar) and f is the `factor`. Default value is 10000. In the case that stress/pressure is sensitive to volume (e.g., materials with high elastic modulus or materials under high pressure), use a larger `factor`.
- `adj_potim` and `tgt_nelm`: `adj_potim` can be set to 0 (no) or 1(yes, by default). If `adj_potim` is set to 1, *SLUSCHI* will adjust the `POTIM` tag in `INCAR`, such that electronic structure calculations finish in `tgt_nelm` (`tgt_nelm=4` by default) steps on average. This helps avoid improper `POTIM` settings that are either too large or too small.

- `diff_solid` and `diff_liquid`: *SLUSCHI* distinguishes solids and liquids based on their diffusions. These two tags provide the parameters for this purpose. By default, `diff_solid=0.5` and `diff_liquid=1.0`. If you find that *SLUSCHI* mistakenly labels a coexisting structure as a solid, try decrease the value of `diff_solid`. Similarly, if you find that *SLUSCHI* mistakenly labels a coexisting structure as a liquid, try increase the value of `diff_liquid`.

4 Run

Running *SLUSCHI* is very simple. After you prepare the files described in the previous section, your `sluschi` directory should look like the following.

```
INCAR POSCAR POTCAR job.in jobsub src
```

Optional files such as `KPOINTS_gamma` and `jobsub_gamma` may also be there.

Then copy the main executable `SLUSCHI` to this directory and run it.

```
cp src/SLUSCHI .
./SLUSCHI &
```

It may take from one day up to two weeks for *SLUSCHI* to calculate a melting temperature and to meet the accuracy requirement, depending on system specifications. Please make sure that the executable `SLUSCHI` keeps running in the background.

5 Outputs

The main output file is `SLUSCHI.out`. Please follow its log to find more detailed output files named `log.out` in each directories, e.g.,

```
Dir_OptUnitCell Dir_VolSearch Dir_Melt Dir_CoexRun
```

The final melting temperature is in the file `Dir_CoexRun/MP.out`, as well as printed in the main `SLUSCHI.out` file.

6 How *SLUSCHI* works

SLUSCHI calculates the melting point following a list of steps.

1. The code first runs a quick structure optimization on the unit-cell you provide, under the pressure `$press` you specify. (i.e., ISIF=3 and IBRION=2 in VASP)

2. Based on the optimized unit-cell, the code builds a supercell, such that periodic images are separated approximately by a distance `$radius`.

3. By running isobaric-isothermal (NPT) molecular dynamics, the code roughly determines the thermal expansion of the solid at the estimated melting temperature `$temp` you provided.

4. Two such supercells are put together into a larger supercell. Half of the supercell is frozen, i.e., the atoms are not allowed to move, while the other half is gradually heated in MD until the solid melts. After it, snapshots of half-solid-half-liquid coexistence are collected from MD trajectory.

5. The code finally employs the small-cell coexistence method to calculate the melting point.