
abICS Documentation

リリース **0.1**

abICS's developer team

2019 年 12 月 09 日

目次

第 1 章	ablICS とは？	1
1.1	概要	1
1.2	開発者	1
1.3	バージョン履歴	1
1.4	ライセンス	2
1.5	コピーライト	2
第 2 章	インストール方法	3
2.1	ダウンロード	3
2.2	必要なライブラリ・環境	3
2.3	ディレクトリ構成	3
2.4	インストール	4
第 3 章	使用方法	7
3.1	概要	7
第 4 章	ファイルフォーマット	9
4.1	[replica] セクション	9
4.2	[solver] セクション	11
4.3	[observer] セクション	12
4.4	[config] セクション	12
第 5 章	アルゴリズム	17
第 6 章	謝辞	19
第 7 章	お問い合わせ	21

第 1 章

abICS とは？

1.1 概要

abICS は、不規則系で配置サンプリングを実行するためのソフトウェアフレームワークであり、金属や酸化物合金などの多成分固体システムに特に重点を置いています。現在は, Quantum Espresso, VASP および aenet を使用することができ、ver.1.0 からは OpenMX もサポートする予定です。

1.2 開発者

abICS は以下のメンバーで開発しています。

- ver. 0.1
 - 笠松 秀輔 (山形大学 学術研究院 (理学部主担当))
 - 本山 裕一 (東京大学 物性研究所)
 - 吉見 一慶 (東京大学 物性研究所)
 - 山本 良幸 (東京大学 物性研究所)
 - 杉野 修 (東京大学 物性研究所)
 - 尾崎 泰助 (東京大学 物性研究所)

1.3 バージョン履歴

2019/12/9 ver.0.1 をリリース.

1.4 ライセンス

本ソフトウェアのプログラムパッケージおよびソースコード一式は GNU General Public License version 3 (GPL v3) に準じて配布されています。

1.5 コピーライト

© 2019- The University of Tokyo. All rights reserved.

本ソフトウェアは 2019 年度 東京大学物性研究所 ソフトウェア高度化プロジェクトの支援を受け開発されており、その著作権は東京大学が所持しています。

第 2 章

インストール方法

2.1 ダウンロード

abICS のソースコードは [GitHub page](#) からダウンロードできます。

```
$ git clone https://github.com/issp-center-dev/pymc-dev
```

2.2 必要なライブラリ・環境

- python3
- numpy
- scipy
- toml (for parsing input files)
- mpi4py (for parallel tempering)
- pymatgen (for parsing vasp I/O)
- qe-tools (for parsing QE I/O)

2.3 ディレクトリ構成

abICS のディレクトリ構成は以下のようになっています. `examples/standard` には簡易ファイルで実行可能なサンプルが, `examples/expert` には `python` モジュールを直接用いて作成されたサンプルがあります. `python` モジュールは `py_mc` ディレクトリ以下に一式格納されています.

```
- examples
  - standard
```

```
- spinel
  - QE
  - vasp
  - openmx
- sub-lattice
  - QE
  - vasp
  - openmx
- expert
  - ising2D
  - 2D_hardcore
  ...
- make_wheel.sh
- py_mc
  - pymc.py
  - python モジュール
- test
```

2.4 インストール

2.4.1 安定版

PyPI に登録されているため、pip を利用してインストール可能です。

```
$ pip install abacus
```

インストールディレクトリを変更したい場合には、`--user` オプションもしくは `--prefix = DIRECTORY (`
`DIRECTORY` にインストールしたいディレクトリを指定) オプションを指定してください:

```
$ pip install --user abacus
```

ライブラリの他に、スクリプト `abacus` がインストールされます。

2.4.2 ユーザ改良版

1. `wheel` ファイルを作成します.

```
$ ./make_wheel.sh
```

2. 作成されたファイルを使用して以下のようにインストールします.

```
$ pip install dist/abacus-*.whl
```

インストールディレクトリを変更したい場合には、`--user` オプションもしくは `--prefix = DIRECTORY (`
`DIRECTORY` にインストールしたいディレクトリを指定) オプションを指定してください:


```
$ pip install --user dist/abacus-\*.whl
```


第 3 章

使用方法

3.1 概要

3.1.1 入力ファイルの準備

3.1.2 実行

ここで指定するプロセス数はレプリカ数と同じである必要があります。

(MPI_Comm_spawn に由来する様々な事項 (実際に割り当てられるプロセスの話や mpiexec のオプションなど) をここかどこかに記載する必要がある)

```
$ mpiexec -np 2 abics input.toml
```


第 4 章

ファイルフォーマット

abICS の入力ファイルは、以下の 4 つのセクションから構成されます。

1. [replica] セクション

レプリカ数や温度の幅、モンテカルロステップ数など、レプリカ交換モンテカルロ部分のパラメータを指定します。

2. [solver] セクション

ソルバーの種類 (VASP, QE, ...)、ソルバーへのパス、不変な入力ファイルのあるディレクトリなど（第一原理計算）ソルバーのパラメータを指定します。

3. [observer] セクション

取得する物理量の種類などを指定します。

4. [config] セクション

合金の配位などを指定します。

以下、順に各セクションの詳細について説明します。

4.1 [replica] セクション

レプリカ数や温度の幅、モンテカルロステップ数など、レプリカ交換モンテカルロ部分のパラメータを指定します。以下のようなファイルフォーマットをしています。

```
[replica]
nreplicas = 3
nprocs_per_replica = 1
kTstart = 500.0
kTend = 1500.0
nsteps = 5
```

```
RXtrial_frequency = 2
sample_frequency = 1
print_frequency = 1
```

4.1.1 入力形式

keyword = value の形式でキーワードとその値を指定します. また, #をつけることでコメントを入力することができます (それ以降の文字は無視されます).

4.1.2 キーワード

- 温度に関する指定

- kTstart

- 形式 : float 型 (>0)

- 説明 : 初期温度を与えます.

- kTend

- 形式 : float 型 (>0)

- 説明 : 計算を終了する最終温度を与えます.

- nsteps

- 形式 : int 型 (自然数)

- 説明 : 温度の分割数を指定します.

- レプリカに関する指定

- nprocs_per_replica

- 形式 : int 型 (自然数)

- 説明 : レプリカに対するプロセス数を指定します. デフォルト値 = 1.

- nreplicas

- 形式 : int 型

- 説明 : レプリカ数を指定します.

- その他

– RXtrial_frequency

形式 : int 型 (自然数)

説明 : レプリカ交換を何モンテカルロステップごとに行うかを指定します. デフォルト値 = 1

– sample_frequency

形式 : int 型 (自然数)

説明 : 物理量測定を何モンテカルロステップごとに行うかを指定します. デフォルト値 = 1

– print_frequency

形式 : int 型 (自然数)

説明 : 測定した物理量を何モンテカルロステップごとに保存するかを指定します. デフォルト値 = 1

4.2 [solver] セクション

ソルバーの種類 (VASP, QE, …)、ソルバーへのパス、不変な入力ファイルのあるディレクトリなど (第一原理計算) ソルバーのパラメータを指定します. 以下のようなファイルフォーマットをしています.

```
[solver]
type = 'vasp'
path = './vasp'
base_input_dir = './baseinput'
perturb = 0.1
```

4.2.1 入力形式

keyword = value の形式でキーワードとその値を指定します. また, #をつけることでコメントを入力することができます (それ以降の文字は無視されます).

4.2.2 キーワード

- type

形式 : str 型

説明 : ソルバーの種類 (OpenMX, QE, VASP) を指定します.

- path

形式 : str 型

説明：ソルバーへのパスを指定します。

- `base_input_dir`

形式：str 型

説明：ベースとなる入力ファイルへのパスを指定します。

- `perturb`

形式：float 型

説明：対称性が良い構造を入力にしてしまうと、構造最適化が鞍点で止まってしまうがちである。これを避けるため、各原子をこのパラメータに比例するようにランダムに変位させたものを初期構造とする。0.0 あるいは `false` に設定することも可能。デフォルト値 = 0.0。

4.3 [observer] セクション

取得する物理量の種類などを指定します。以下のようなファイルフォーマットをしています。

```
[observer]
type = 'default'
```

4.3.1 入力形式

`keyword = value` の形式でキーワードとその値を指定します。また、`#`をつけることでコメントを入力することができます (それ以降の文字は無視されます)。

4.3.2 キーワード

- `type`

形式：str 型

説明：物理量セットを指定します。

- 「default」

* デフォルト設定です。エネルギーと各座標に入った原子グループを取得します。

4.4 [config] セクション

合金の配位などを指定します。以下のようなファイルフォーマットをしています。


```

[config]
unitcell = [[8.1135997772, 0.0000000000, 0.0000000000],
            [0.0000000000, 8.1135997772, 0.0000000000],
            [0.0000000000, 0.0000000000, 8.1135997772]]
supercell = [1,1,1]

[[config.base_structure]]
type = "O"
coords = [
    [0.237399980, 0.237399980, 0.237399980],
    [0.762599945, 0.762599945, 0.762599945],
    ## 中略
    [0.262599975, 0.262599975, 0.762599945],
]

[[config.defect_structure]]
coords = [
    [0.000000000, 0.000000000, 0.000000000],
    [0.749999940, 0.249999985, 0.499999970],
    ## 中略
    [0.124999993, 0.624999940, 0.124999993],
]

[[config.defect_structure.groups]]
name = 'Al'
# species = ['Al']      # default
# coords = [[[0,0,0]]]  # default
num = 16

[[config.defect_structure.groups]]
name = 'Mg'
# species = ['Mg']      # default
# coords = [[[0,0,0]]]  # default
num = 8

```

4.4.1 入力形式

keyword = values の形式でキーワードとその値を指定します. また, #をつけることでコメントを入力することができます (それ以降の文字は無視されます).

4.4.2 キーワード

- 格子の指定

– unitcell

形式: list 型

説明：格子ベクトル **a, b, c** を, リスト形式で [**a, b, c**] として指定します.

– supercell

形式：list 型

説明：超格子の大きさをリスト形式で [**a, b, c**] 指定します.

- [[config.base_structure]] セクション

type と coords によりモンテカルロ計算で動かさない原子種とその座標を指定します. 原子種が複数ある場合には, 複数の [[config.base_structre]] セクションを指定します.

– type

形式：str 型

説明：原子種を指定します.

– coords

形式：list の list もしくは 文字列

説明：座標を指定します. 3 次元座標を表す 3 要素のリストを N 個（原子の数）だけ並べたリストか, 座標を N 行 3 列に並べた文字列で指定します.

- [[config.defect_structure]] セクション

モンテカルロで更新する原子が入る座標 (coords) と入りうる原子（団） (group) を指定します. Ver. 1.0 では POSCAR や cif からの変換ツールが利用出来るようになる予定です.

– coords

形式：list の list もしくは 文字列

説明：原子が入る座標を指定します. 3 次元座標を表す 3 要素のリストを N 個（原子の数）だけ並べたリストか, 座標を N 行 3 列に並べた文字列で指定します.

– [[config.defect_structure.groups]] セクション

モンテカルロで更新する原子グループの情報を指定します.

* name

形式：str 型

説明：原子グループの名前を指定します.

* species

形式：list 型

説明：原子グループに属する原子種を指定します。デフォルト値は name で指定したものがひとつだけ含まれたリストです。

* coords

形式：list の list もしくは 文字列

説明：原子グループ中の各原子の座標を指定します。3 次元座標を表す 3 要素のリストを N 個（原子の数）だけ並べたリストか、座標を N 行 3 列に並べた文字列で指定します。デフォルト値は $[[0.0, 0.0, 0.0]]$ です。

* num

形式：int 型

説明：この原子グループの数を指定します。

第 5 章

アルゴリズム

T.B.A.

資料をアップする？

第 6 章

謝辞

このソフトウェアの開発は, 様々なプロジェクトとコンピューター資源の提供によりサポートされてきました。この場を借りて感謝します。

- ポスト「京」重点課題 5
- 東京大学物性研究所スーパーコンピュータ共同利用
- 文部科学省卓越研究員事業
- 科学研究費補助金 (No. JP18H05519, No. 19K15287)
- JST CREST (No. JPMJCR15Q3, No. 19K15287)
- NEDO

また, abICS は東京大学物性研究所 ソフトウェア高度化プロジェクト (2019 年度) の支援を受け開発されました。この場を借りて感謝します。

第 7 章

お問い合わせ

abICS に関するお問い合わせはこちらにお寄せください。

- バグ報告

abICS のバグ関連の報告は **GitHub** の **Issues** で受け付けています。

バグを早期に解決するため、報告時には次のガイドラインに従ってください。

- 使用している abICS のバージョンを指定してください。
- インストールに問題がある場合には、使用しているオペレーティングシステムとコンパイラの情報についてお知らせください。
- 実行に問題が生じた場合は、実行に使用した入力ファイルとその出力を記載してください。

- その他

研究に関連するトピックなど **GitHub** の **Issues** で相談しづらいことを問い合わせる際には、以下の連絡先にコンタクトをしてください。

E-mail: `abics-dev__at__issp.u-tokyo.ac.jp` (`_at_`を`@`に変更してください)