Cif2x Documentation

リリース *1.0-alpha*

ISSP, University of Tokyo

2023 年 12 月 28 日

目次

第 1 章

概要

1.1 cif2x とは?

近年、機械学習を活用した物性予測や物質設計 (マテリアルズインフォマティクス) が注目されています。機械学習の精度は、適切な教師データの準備に大きく依存しています。そのため、迅速に教師データを生成するためのツールや環境の整備は、マテリアルズインフォマティクスの研究進展に大きく貢献すると期待されます。

cif2x は、cif ファイルから第一原理計算用の入力ファイルを生成するツールです。入力パラメータを雛形として、物質の種類や計算条件によって変わる箇所を結晶構造データなどから構成します。特定の計算条件に応じた複数の入力ファイルを生成することが可能です。現在は、 VASP 、Quantum ESPRESSO 、OpenMX に対応し、将来的には AkaiKKR にも対応する予定です。

1.2 ライセンス

1.3 開発貢献者

本ソフトウェアは以下の開発貢献者により開発されています。

- ver.1.0-alpha (2023/12/28 リリース)
  - 開発者
    - 吉見 一慶 (東京大学 物性研究所)
    - 青山 龍美 (東京大学 物性研究所)
    - 本山 裕一 (東京大学 物性研究所)

        * 福田 将大 (東京大学 物性研究所)
        * 福島 鉄也 (産業技術総合研究所)
        * 井戸 康太 (東京大学 物性研究所)
        * 笠松 秀輔 (山形大学 学術研究院 (理学部主担当))
        * 是常 隆　(東北大学大学院理学研究科)
    - プロジェクトコーディネーター
        * 尾崎 泰助 (東京大学 物性研究所)

1.4 コピーライト

1.5 動作環境

以下の環境で動作することを確認しています。

- Ubuntu Linux + python3

第 2 章

インストールと基本的な使い方

必要なライブラリ・環境

HTP-tools に含まれる第一原理計算入力ファイル生成ツール cif2x を利用するには、以下のプログラムとライブラリが必要です。

- python 3.x
- pymatgen モジュール
- ruamel.yaml モジュール
- f90nml モジュール
- qe-tools モジュール
- numpy モジュール
- pandas モジュール
- monty モジュール

ソースコード配布サイト

- GitHub リポジトリ

ダウンロード方法

git を利用できる場合は、以下のコマンドで cif2x をダウンロードできます。

```
$ git clone https://github.com/issp-center-dev/cif2x.git
```

インストール方法

cif2x をダウンロード後、以下のコマンドを実行してインストールします。cif2x が利用するライブラリも必要に応じてインストールされます。

```
$ cd ./cif2x
$ python3 -m pip install .
```

実行プログラム cif2x がインストールされます。

ディレクトリ構成

```
.
|-- LICENSE
|-- README.md
|-- pyproject.toml
|-- docs/
|   |-- ja/
|   |-- tutorial/
|-- src/
|   |-- cif2x/
|       |-- __init__.py
|       |-- main.py
|       |-- cif2struct.py
|       |-- struct2qe.py
|       |-- qe/
|       |   |-- __init__.py
|       |   |-- calc_mode.py
|       |   |-- cards.py
|       |   |-- content.py
|       |   |-- qeutils.py
|       |   |-- tools.py
|       |-- struct2vasp.py
|       |-- struct2openmx.py
|       |-- openmx/
|       |   |-- __init__.py
|       |   |-- vps_table.py
|       |-- utils.py
|-- sample/
```

基本的な使用方法

cif2x は第一原理計算プログラムのための入力ファイルを生成するツールです。入力パラメータを雛形として、物質の種類や計算条件によって変わる箇所を結晶構造データなどから構成します。現在は Quantum ESPRESSO, VASP, および OpenMX の入力ファイル形式に対応しています。

1. 入力パラメータファイルの作成

cif2x を使用するには、まず、生成する入力ファイルの内容を記述したパラメータファイルを YAML 形式で作成します。詳細についてはファイルフォーマットの章を参照してください。

2. 結晶構造ファイルと擬ポテンシャルファイルの配置

   対象となる物質の結晶構造を記述したファイルを用意します。ファイル形式は CIF または pymatgen が扱える POSCAR や xfs 形式に対応しています。

   Quantum ESPRESSO の場合、利用する擬ポテンシャルファイルと、CSV 形式のインデックスファイルを配置します。擬ポテンシャルファイルの配置先などは入力パラメータファイル内に指定します。

   VASP の場合、擬ポテンシャルファイルの格納場所を `~/.config/.pmgrc.yaml` ファイルに記述するか環境変数にセットします。入力パラメータファイル内で指定することもできます。

3. コマンドの実行

   作成した入力パラメータファイルおよび結晶構造データファイルを入力として cif2x プログラムを実行します。Quantum ESPRESSO 用の入力ファイルを生成する場合はターゲットオプションに `-t QE` を指定します。VASP の場合は `-t VASP`, OpenMX の場合は `-t OpenMX` を指定します。

```
$ cif2x -t QE input.yaml material.cif
```

第 3 章

チュートリアル

第一原理計算入力ファイル生成ツール cif2x を使うには、入力パラメータファイルと結晶構造データおよび擬ポテンシャルファイルを用意した後、プログラム cif2x を実行します。現在は Quantum ESPRESSO, VASP, OpenMX の入力ファイル生成に対応しています。以下では、 `docs/tutorial/cif2x` ディレクトリにある Quantum ESPRESSO 向けサンプルを例にチュートリアルを実施します。

3.1 入力パラメータファイルを作成する

入力パラメータファイルには、第一原理計算プログラムに与える入力ファイルの内容を記述します。

以下に入力パラメータファイルのサンプルを記載します。このファイルは YAML 形式のテキストファイルで、結晶構造データに対するオプションの指定や、出力する第一原理計算入力ファイルの内容を記述します。仕様の詳細については **ファイルフォーマット** の章を参照してください。

YAML フォーマットでは、 `keyword: value` の辞書形式でパラメータを記述します。 `value` には数値や文字列などのスカラー値や、複数の値を `[ ]` または箇条書きの形式で列挙するリスト型、または辞書型を入れ子にすることができます。

```
structure:
  use_ibrav: false
  tolerance: 0.05

optional:
  pseudo_dir: ./pseudo
  pp_file: ./pseudo/pp_psl_pbe_rrkjus.csv

tasks:
  - mode: scf
    output_file: scf.in
```

(次のページに続く)

(前のページからの続き)

```
  output_dir: scf
  template: scf.in_tmpl
  content:
    namelist:
      control:
        prefix: pwscf
        pseudo_dir:
        outdir: ./work
      system:
        ecutwfc:
        ecutrho:
    CELL_PARAMETERS:
    ATOMIC_SPECIES:
    ATOMIC_POSITIONS:
      option: crystal
    K_POINTS:
      option: automatic
      grid: [8,8,8]
```

入力パラメータファイルは structure, optional, tasks のブロックから構成されます。structure は結晶構造データに関するオプションを指定します。optional は擬ポテンシャルに関する global な設定などを行います。

tasks は出力する第一原理計算入力ファイルの内容を指定します。一連の計算に対応して複数のファイルを生成できるよう、tasks は配列の値を取ります。各出力について、計算内容は mode パラメータで指定します。SCF 計算の scf や NSCF 計算の nscf に対応するほか、一般的な出力を行う任意の出力モードを指定できます。ファイルは output_dir および output_file で指定するファイルに書き出されます。

出力内容は content に記載します。Quantum ESPRESSO の入力ファイルは、 &keyword で始まる Fortran90 の namelist 形式と、K_POINTS などのキーワードで始まり空行で分割される cards と呼ばれるブロックからなります。content には namelist と cards を入れ子の辞書形式で指定します。いくつかの例外を除いて、指定された内容が基本的にはそのまま入力ファイルに書き出されます。値が空欄のキーワードは、結晶構造データなどから求めた値が代入されます。

また、template として入力ファイルの雛形を指定することもできます。 template に指定したファイルの内容と content のデータを合わせたものを入力データとして扱います。同じキーワードのデータが存在する場合は content の指定が優先されます。従って、既存の入力ファイルを元に必要な箇所を入力パラメータファイルで上書きする使い方が可能です。上記の例では次のファイル (scf.in_tmpl) を template として取り込み、カットオフと CELL_PARAMETER, ATOMIC_SPECIES, ATOMIC_POSITIONS, K_POINT の箇所を結晶構造等から決めます。ecutwfc と ecutrho が空欄で上書きされていることに留意してください。

```
&control
    calculation = 'scf'
    prefix = 'pwscf'
    pseudo_dir = './pseudo'
    outdir = './work'
    tstress = .true.
    tprnfor = .true.
/

&system
    ibrav = 0
    nat = 7
    ntyp = 3
    ecutwfc = 36.0
    ecutrho = 180.0
    occupations = 'smearing'
    smearing = 'm-p'
    degauss = 0.01
    noncolin = .true.
    nspin = 2
/

&electrons
    missing_beta = 0.1
    conv_thr = 1e-08
/
```

3.2 第一原理計算入力ファイルを生成する

入力パラメータファイル (input.yaml) と結晶構造データ (Co3SnS2_nosym.cif) を入力として cif2x を実行します。

```
$ cif2x -t QE input.yaml Co3SnS2_nosym.cif
```

予め必要な擬ポテンシャルのファイルを `./pseudo` ディレクトリに配置し、擬ポテンシャルのインデックスファイルを `./pseudo/pp_psl_pbe_rrkjus.csv` に作成しておきます。

`cif2x` を実行すると Quantum ESPRESSO 用の入力ファイルが生成され出力されます。出力先は入力パラメータファイル内のパラメータで指定するディレクトリ (`output_dir`) およびファイル (`output_file`) です。この例では `./scf/scf.in` に SCF 計算用の入力ファイルが書き出されます。

3.3 パラメータセットを指定する

入力パラメータ内の値をいくつか変えながら一連の入力ファイルを生成したいことがあります。例えばカットオフの値や k 点の数を変えて収束性を評価するなどの場合です。入力パラメータには値のリストや範囲を指定することができ、値の組み合わせごとに個別のディレクトリを作成して入力ファイルを生成します。パラメータセットの指定は特別な構文 `${...}` を用います。

```
content:
  K_POINTS:
    option: automatic
    grid:   ${ [ [4,4,4], [8,8,8], [12,12,12] ] }
```

例えば上記のように K_POINTS を指定すると、 grid の値が [4,4,4], [8,8,8], [12,12,12] の入力ファイルがそれぞれ 4x4x4/, 8x8x8/, 12x12x12/ サブディレクトリ内に作成されます。

第 4 章

コマンドリファレンス

4.1 cif2x

第一原理計算のための入力ファイルを生成する

書式:

```
cif2x [-v][-q] -t target input_yaml material.cif
cif2x -h
cif2x --version
```

説明:

input_yaml に指定した入力パラメータファイルと material.cif に指定した結晶構造データを読み込み、第一原理計算プログラム用の入力ファイルを生成します。現在は Quantum ESPRESSO, VASP, OpenMX に対応しています。以下のオプションを受け付けます。

- `-v`

  実行時に表示されるメッセージを冗長にします。複数回指定すると冗長度が上がります。

- `-q`

  実行時に表示されるメッセージの冗長度を下げます。 `-v` の効果を打ち消します。複数回の指定が可能です。

- `-t` *target*

  対象となる第一原理計算プログラムを指定します。 *target* として指定可能な値は以下のとおりです。大文字小文字は区別しません。

  - `QE`, `espresso`, `quantum_espresso`: Quantum ESPRESSO 向け入力ファイルを生成します。
  - `VASP`: VASP 向け入力ファイルを生成します。

  - `OpenMX`: OpenMX 向け入力ファイルを生成します。

- `input_yaml`

  入力パラメータファイルを指定します。形式は YAML format です。

- `material.cif`

  結晶構造データファイルを指定します。形式は CIF の他、pymatgen で扱える形式のファイルを指定できます。

- `-h`

  ヘルプを表示します。

- `--version`

  バージョン情報を表示します。

第 5 章

ファイルフォーマット

5.1 入力パラメータファイル

入力パラメータファイルでは、cif2x で第一原理計算入力ファイルを生成するための設定情報を YAML 形式で記述します。本ファイルは以下の部分から構成されます。

1. structure セクション: 結晶構造データの扱いについてのオプションを記述します。
2. optional セクション: 擬ポテンシャルファイルの指定や、YAML の参照機能を利用する場合のシンボル定義を行います。
3. tasks セクション: 入力ファイルの内容を記述します。

5.1.1 structure

`use_ibrav` (デフォルト値: `false`)

結晶構造の入力に Quantum ESPRESSO の `ibrav` パラメータを利用します。 `true` の場合、格子のとり方を Quantum ESPRESSO の convention に合うように変換します。入力ファイルにはあわせて格子に関するパラメータ a, b, c, `cosab`, `cosac`, `cosbc` が (必要に応じて) 書き出されます。

`tolerance` (デフォルト値: 0.01)

`use_ibrav = true` の場合に、再構成した Structure データと元データとの一致を評価する際の許容度を指定します。

`supercell` (デフォルト値: なし)

supercell を設定する場合に supercell のサイズを $[n_x, n_y, n_z]$ で指定します。

5.1.2 optional

第一原理計算プログラムごとに必要な global な設定を行います。記述する内容は以下の各節に記述します。

5.1.3 tasks

入力ファイルの内容を記述します。複数の入力ファイルに対応するため、`tasks` には各入力ファイルごとのブロックからなるリスト形式をとります。各ブロックに記述する項目は以下のとおりです。

`mode` (Quantum ESPRESSO)

入力ファイルの計算内容を指定します。現時点では Quantum ESPRESSO の pw.x 向けに `scf` と `nscf` に対応しています。対応していない mode については、 `content` の内容をそのまま出力します。

`output_file` (Quantum ESPRESSO)

出力ファイル名を指定します。

`output_dir`

出力先のディレクトリ名を指定します。デフォルト値はカレントディレクトリです。

`content`

出力内容を指定します。Quantum ESPRESSO の場合は `namelist` ブロックに namelist データ (`&system` や `&control` など) を記述し、 `K_POINTS` 等の card データを個別のブロックとして記述します。card データはパラメータをとるものがあります。

`template` (Quantum ESPRESSO)

`template_dir` (VASP)

出力内容のテンプレートファイルを指定します。指定がない場合はテンプレートを利用しません。このファイルの内容を `content` に追加します。同じデータがある場合は後者を優先します。

5.1.4 パラメータセット指定

入力パラメータには値のリストや範囲を指定することができ、値の組み合わせごとに個別のディレクトリを作成して入力ファイルを生成します。パラメータセットの指定には特別な構文 `${...}` を用います。指定方法は次の通りです:

- リスト `${[ A, B, ... ]}`

  パラメータセットを Python のリストとして列挙します。各項目はスカラー値やリストを指定できます。

- 範囲指定 `${range(N)}`, `${range(start, end, step)}`

パラメータの範囲を指定します。それぞれ 0〜N-1, start〜end を step 刻み (step を省略した場合は 1) です。N, start, end, step は int または float です。

5.2 Quantum ESPRESSO 向けパラメータ

optional セクションおよび tasks セクションの content に記載する内容について、Quantum ESPRESSO 固有の内容を記述します。現時点では pw.x の scf および nscf に対応しています。

5.2.1 optional セクション

pp_file

元素種と擬ポテンシャルを対応付ける CSV 形式のインデックスファイルを指定します。ファイルの書式は、元素種、擬ポテンシャルファイルのタイプ、nexclude、orbitals です。例:

```
Fe,pbe-spn-rrkjus_psl.0.2.1,4,spd
```

擬ポテンシャルファイルのファイル名は Fe.pbe-spn-rrkjus_psl.0.2.1.UPF に対応します。

cutoff_file

擬ポテンシャルファイルとカットオフを対応付ける CSV 形式のインデックスファイルを指定します。ファイルの書式は、擬ポテンシャルファイル、ecutwfc の値、ecutrho の値 です。

pseudo_dir

擬ポテンシャルファイルを格納するディレクトリ名を指定します。カットオフの値を擬ポテンシャルファイルから取得する場合に使用します。Quantum ESPRESSO の pseudo_dir パラメータとは独立に指定します。

5.2.2 content

namelist

- structure セクションの use_ibrav パラメータに応じて、 &system の格子情報の指定が上書きされます。
  - use_ibrav = false: ibrav は 0 にセットされます。また、格子パラメータに関する a, b, c, cosab, cosac, cosbc, celldm は削除されます。
  - use_ibrav = true: ibrav は結晶構造データから取得された Bravais 格子のインデックスがセットされます。また、Structure データは基本格子のとり方など Quantum ESPRESSO の convention に合わせて再構成されます。

- nat (原子数) および ntyp (元素種の数) は結晶構造データから取得される値で上書きされます。
- カットオフ ecutwfc および ecutrho の情報は、パラメータの値が空欄の場合は擬ポテンシャルファイルから取得します。

CELL_PARAMETERS

- structure セクションの use_ibrav パラメータが true の場合は出力されません。false の場合は格子ベクトルが出力されます。単位は angstrom です。
- 格子ベクトルの情報は結晶構造データから取得されます。 data フィールドに 3x3 の行列を直接指定した場合はその値が用いられます。

ATOMIC_SPECIES

- 原子種・原子量・擬ポテンシャルファイル名のリストを出力します。
- 原子種の情報は結晶構造データから取得されます。擬ポテンシャルのファイル名は pp_list で指定する CSV 形式のインデックスファイルを参照します。
- data フィールドに必要なデータを指定した場合はその値が用いられます。

ATOMIC_POSITIONS

- 原子種と原子座標 (fractional coordinate) のリストを出力します。
- ignore_species に原子種または原子種のリストを指定した場合、その原子種については if_pos の値が 0 にセットされます。MD や構造最適化の際に使われます。
- data フィールドに必要なデータを指定した場合はその値が用いられます。

K_POINTS

- k 点の情報を出力します。 option に出力タイプを指定します。
  - gamma: Γ 点を用います。
  - crystal: メッシュ状の k 点のリストを出力します。メッシュの指定は grid パラメータまたは vol_density や k_resolution から導出される値が用いられます。
  - automatic: k 点のメッシュを指定します。メッシュの指定は grid パラメータまたは vol_density や k_resolution から導出される値が用いられます。 シフトの指定は kshifts パラメータを参照します。
- メッシュの指定は以下の順序で決定されます。
  - grid パラメータの指定。grid の値は n_x, n_y, n_z の配列またはスカラー値 n です。後者の場合は $n_x = n_y = n_z = n$ と仮定します。
  - vol_density パラメータから自動導出。

  - k_resolution パラメータから自動導出。k_resolution のデフォルトは 0.15 です。

- data フィールドに必要なデータを指定した場合はその値が用いられます。

5.3 VASP 向けパラメータ

optional セクションおよび tasks セクションの content に記載する内容について、VASP 固有の内容を記述します。

5.3.1 optional

擬ポテンシャルのタイプや格納場所を指定します。

pymatgen では、擬ポテンシャルファイルを PMG_VASP_PSP_DIR/*functional*/POTCAR. *element* (.gz) または PMG_VASP_PSP_DIR/*functional*/ *element* /POTCAR から取得します。PMG_VASP_PSP_DIR はディレクトリの指定で、設定ファイル ~/.config/.pmgrc.yaml に記載するか、同名の環境変数に指定します。また、*functional* は擬ポテンシャルのタイプで、 POT_GGA_PAW_PBE や POT_LDA_PAW などが決められています。

pseudo_functional

擬ポテンシャルのタイプを指定します。タイプの指定と上記の *functional* の対応は pymatgen 内のテーブルに定義され、 PBE → POT_GGA_PAW_PBE, LDA → POT_LDA_PAW などのようになっています。

以下の pseudo_dir を指定した場合は pymatgen の流儀を無視して擬ポテンシャルの格納ディレクトリを探します。

psuedo_dir

擬ポテンシャルの格納ディレクトリを指定します。擬ポテンシャルファイルのファイル名は pseudo_dir/POTCAR. *element* (.gz) または pseudo_dir/*element*/POTCAR です。

5.3.2 tasks

テンプレートファイルは、 template_dir で指定するディレクトリ内に INCAR, KPOINTS, POSCAR, POTCAR ファイルを配置します。ファイルがない項目は無視されます。

5.3.3 content

incar

- INCAR ファイルに記述するパラメータを列挙します。

kpoints

- `type`

  KPOINTS の指定方法を記述します。以下の値に対応しています。タイプによりパラメータが指定可能なものがあります。詳細は pymatgen.io.vasp のマニュアルを参照してください。

  - `automatic`

    parameter: `grid`

  - `gamma_automatic`

    parameter: `grid`, `shift`

  - `monkhorst_automatic`

    parameter: `grid`, `shift`

  - `automatic_density`

    parameter: `kppa`, `force_gamma`

  - `automatic_gamma_density`

    parameter: `grid_density`

  - `automatic_density_by_vol`

    parameter: `grid_density`, `force_gamma`

  - `automatic_density_by_lengths`

    parameter: `length_density`, `force_gamma`

  - `automatic_linemode`

    parameter: `division`, `path_type` (HighSymmKpath の path_type に対応)

5.4 OpenMX 向けパラメータ

optional セクションおよび tasks セクションの content に記載する内容について、OpenMX 固有の内容を記述します。

5.4.1 optional

data_path

擬原子軌道および擬ポテンシャルのファイルを格納するディレクトリを指定します。入力ファイルの DATA.PATH パラメータに対応します。

5.4.2 content

precision

擬原子軌道を OpenMX マニュアル 10.6 章の Table 1, 2 にしたがって選択します。 quick, standard, precise のいずれかの値を取ります。デフォルト値は quick です。

第 6 章

拡張ガイド

6.1 Quantum ESPRESSO の mode を追加する

Quantum ESPRESSO の計算モードへの対応を追加するには、`src/cif2x/qe/calc_mode.py` の `create_modeproc()` 関数に mode と変換クラスの対応付けを記述します。

```
def create_modeproc(mode, qe):
    if mode in ["scf", "nscf"]:
        modeproc = QEmode_pw(qe)
    else:
        modeproc = QEmode_generic(qe)
    return modeproc
```

mode ごとの変換機能は `QEmode_base` の派生クラスとしてまとめられています。このクラスは `update_namelist()` で namelist ブロックの更新と、`update_cards()` で cards ブロックのデータ生成を行います。現在は pw.x の scf および nscf に対応する `QEmode_pw` クラスと、変換せずそのまま出力する `QEmode_generic` クラスが用意されています。

```
class QEmode_base:
    def __init__(self, qe):
    def update_namelist(self, content):
    def update_cards(self, content):
```

namelist については、空欄の値を結晶構造データ等から生成して代入するほか、格子パラメータなど Structure から決まる値や、他のパラメータとの整合性をとる必要のある値を強制的にセットする場合があります。処理内容はモードごとに個別に対応します。

cards ブロックについては、card の種類ごとに関数を用意し、card 名と関数の対応付けを `card_table` 変数に列挙します。基底クラスの `update_cards()` では、card 名から対応する関数を取得して実行し、card の情報を更新します。もちろん、全く独自に `update_cards()` 関数を作成することもできます。

```
self.card_table = {
    'CELL_PARAMETERS': generate_cell_parameters,
    'ATOMIC_SPECIES': generate_atomic_species,
    'ATOMIC_POSITIONS': generate_atomic_positions,
    'K_POINTS': generate_k_points,
}
```

card ごとの関数は `scr/cif2x/qe/cards.py` にまとめられており、関数名は `generate_{card 名}` としています。この関数は card ブロックのパラメータを引数に取り、card 名、option、data フィールドからなる辞書データを返します。