
Pymatflow Docs

Release 0.0.2

DeqiTang

Mar 29, 2022

Contents

1	Tutorials	3
2	Manuals	9
3	For Chinese User	21
4	Indices and tables	23

The Python Materials WorkFlow (Pymatflow) is an workflow simplifier for reserach on Materials Science by means of Ab initio simulation. Pymatflow provides module interface to several popular open source Ab initio programs along with many useful command line tools which can help setting up, post-processing the simulations. A basic usage of Pymatflow (via module interaface) is like this:

```
>>> from pymatflow.qe import static_run
>>> task = static_run()
>>> task.get_xyz("C02.xyz")
>>> params = {
>>>     "encutwfc":60,
>>>     "degauss": 0.001
>>> }
>>> task.set_params(params)
>>> task.scf(runopt="genrun", auto=0)
```

The above code will read the xyz structure file C02.xyz and do a basic scf calculation on it. Note, to make sure `task.get_xyz()` run normally we must specify cell parameters in the second line of C02.xyz manually, in format like this:

```
3
cell: 10 0 0 | 0 10 0 | 0 0 10
C  0.0000000 0.000000 0.000000
O -1.1600000 0.000000 0.000000
O  1.1500000 0.000000 0.000000
```


1.1 Tutorials (English)

Sorry! The content is still on the way!

1.2 中文版教程

目前关于 Pymatflow 的模块接口使用的教程暂未完成。用户现在如何使用已经编写好的命令行工具来用于日常的研究当中，了解使用方式的最佳方式是使用案例 (Examples)。

1.2.1 命令行脚本使用示例（部分为公众号文章链接）

Phonopy+Quantum ESPRESSO 计算固体声子谱 (Pymatflow 简化版)

本教程将简单介绍如何使用 Pymatflow 简单地实现利用 Phonopy+Quantum ESPRESSO 实现对固体声子谱的计算。

Phonopy 通过体系的原子间力常数来计算声子谱，而力常数的计算需要借助于第一性原理程序，目前非常多的 DFT 程序都得到了支持，比如 VASP、QE、Abinit、CP2K 等。

Pymatflow 工具将使用 Phonopy 的过程进行了封装，能够方便地利用 Phonopy 结合其它 DFT 程序，比如 QE、CP2K、Abinit 来快速进行声子谱的计算。

使用流程

- 准备好 P 势文件和 xyz 结构文件，注意第二行需要按照一定方式注明 cell 参数。
- 运行命令提交计算

案例演示

最近待在家里，切身体验柴米油盐酱醋茶的滋味。为此这次的研究对象就选取每天都能尝到的 NaCl 好了。这里的 NaCl 为常见的面心立方结构，空间群为 225，结构如图所示：

准备 xyz 类型的结构文件 (nacl.xyz) 如下：

```
8
cell:  5.6916936  0.0000000  0.0000000 |  0.0000000  5.6916936  0.0000000
↪ |  0.0000000  0.0000000  5.6916936
Na      0.0000000  0.0000000  0.0000000
Na      0.0000000  2.8458468  2.8458468
Na      2.8458468  0.0000000  2.8458468
Na      2.8458468  2.8458468  0.0000000
Cl      0.0000000  0.0000000  2.8458468
Cl      0.0000000  2.8458468  0.0000000
Cl      2.8458468  0.0000000  0.0000000
Cl      2.8458468  2.8458468  2.8458468
```

然后在服务器上运行以下命令：

```
qe-phonopy.py -f nacl.xyz --ecutwfc 60 --kpoints-mp 3 3 3 0 0 0 --supercell-n
1 1 1
```

然后你就可以静静等待任务运行结束。然后运行后处理程序。下面解释一下参数的意义：

- -f 指定结构文件
- -kpoints-mp 指定 Monkhorst-Pack 的 K 点方案
- -supercell-n 是通过 pymatflow 传递给 Phonopy 的超胞参数，对应于 Phonopy 的 -dim 参数

下面便是结果：

声子谱：

这次的内容就到这里了，后面我还会继续更新 Pymatflow 的使用案例，并继续整理文档，欢迎使用和讨论，有意参与开发者或者有任何建议者可以联系我。

更多文章

[Band Structure](#)

[Phonopy+QE](#)

微信公众号



欢迎关注微信公众号“生材有道”，可以订阅更多资讯。

使用 Pymatflow 辅助 Quantum ESPRESSO 计算能带

疫情期间，我们要积极响应国家、社会的号召待乖乖在家里。学校暂时肯定回不去了，待在家里我们也可以好好学习哦。今天我将自己陆陆续续花了半年利用课余时间用 Python 写的一个 DFT 计算流程管理工具 Pymatflow 整理并发布到了 PyPI 上。

Pymatflow 本质上是一个为开源软件如 Quantum ESPRESSO、CP2K、SIESTA 等自动生成输入文件以及将任务在本地运行或者提交到服务器运行的工具。

为了向大家介绍该工具，我这里以一个实际的能带结构计算的例子来展示如何使用 Pymatflow:

这里假设已经在服务器上安装了 Pymatflow，然后你的服务器的作业调度器是 PBS。具体的安装办法是:

```
pip install pymatflow --user
```

注意要将 `~/local/bin` 加入到你的 `PATH` 中，如果你是系统管理员，也可以直接:

```
sudo pip install pymatflow
```

然后准备好 xyz 格式的结构文件，这里以 MoS2 作为模拟对象。

结构文件 (mos2.xyz) 内容如下:

```
6
cell:  3.1922380  0.0000000  0.0000000 | -1.5961190  2.7645592  0.0000000
↪ |  0.0000000  0.0000000  13.3782944
Mo     1.5961190  0.9215197  3.3445736
Mo     0.0000000  1.8430395  10.0337208
S      1.5961190  0.9215197  8.4695574
S      0.0000000  1.8430395  4.9087370
S      1.5961190  0.9215197  11.5978843
S      0.0000000  1.8430395  1.7804102
```

需要注意的是为了让 Pymatflow 能够正确读取结构, 你需要在 xyz 结构文件的第二行指定体系的晶胞参数, 就像上述内容中所示。

然后就开始进行静态 SCF 的计算, 将 mos2.xyz 文件与对应的势能文件放在一个目录下, 然后运行一下命令:

```
qe-scf.py -f mos2.xyz -d mos2-band --ecutwfc 80 --kpoints-mp 3 3 3 0 0 0
--runopt=gen --auto=3
```

等待上述任务完成后执行一下命令进行 nscf 的计算:

```
qe-nscf.py -f mos2.xyz -d mos2-band --ecutwfc 80 --kpoints-mp 5 5 5 0 0 0
--runopt=gen --auto=3
```

然后你需要准备一个能带高对称点路径文件, 示例如下:

```
11
0.000000 0.000000 0.000000 #GAMMA 15
0.000000 0.500000 0.000000 #Z 15
0.000000 0.500000 0.500000 #D 15
0.000000 0.000000 0.500000 #B 15
0.000000 0.000000 0.000000 #GAMMA 15
-0.500000 0.000000 0.500000 #A 15
-0.500000 0.500000 0.500000 #E 15
0.000000 0.500000 0.000000 #Z 15
-0.500000 0.500000 0.000000 #C_2 15
-0.500000 0.000000 0.000000 #Y_2 15
0.000000 0.000000 0.000000 #GAMMA 15
```

等 nscf 计算结束后, 执行以下命令进行能带结构的计算:

```
qe-bands.py -f mos2.xyz -d mos2-band --ecutwfc 80 --crystal-b-file kpath.txt
--runopt=gen --auto=3
```

支持能带结构计算就完成了, 运行以下命令进行作图:

```
post-qe-bands.py -d mos2-band
```

得到结果如下:

该工具在我的日常学习工作中扮演者比较重要的角色, 可以很大程度提高我的工作效率 (当然撇去开发

时间不算)。另外需要注意的是此工具的使用需要使用者对自己在做什么有清晰的认识，具体的计算参数的设置可以通过命令行参数传递给程序。

目前 Pymatflow 对 QE、CP2K 的结构弛豫、晶胞参数优化、SCF、能带计算、过渡态计算等有较好的支持，我也会继续增加功能和进行维护，感兴趣的朋友也可以联系我一起维护。由于才整理好代码部分，文档还未整理完全，目前会通过此公众号发布一些使用案例来和大家交流。有好的建议欢迎发送邮件到 `pymatflow@163.com`

更多文章

[Band Structure](#)

[Phonopy+QE](#)

微信公众号



欢迎关注微信公众号“生材有道”，可以订阅更多资讯。

1.2.2 注意事项

在使用 Pymatflow 命令行工具自动进行第一性原理模拟输入文件的生成、任务提交和结果后处理的过程中，如果涉及到能带、声子谱计算等涉及到倒空间高对称点路径的模拟，用户可统一通过 `--kpath` 或者 `--kpath-file` 提供高对称点路径信息 (推荐使用后者)。 `--kpath-file` 用于指定文件包含有高对称点路径，文件格式如下：

```
5
0.000000 0.000000 0.000000 #GAMMA 15
```

(continues on next page)

(continued from previous page)

```
0.500000 0.000000 0.000000 #X |  
0.000000 0.500000 0.000000 #Y 15  
0.000000 0.000000 0.000000 #GAMMA 15  
0.000000 0.000000 0.500000 #Z
```

其中，第一行是一个整数指定一共有多少个高对称点。后面的行用于指定高对称点的坐标信息符号信息(必须大写)以及连接情况。如果某一个高对称点与其后的高对称点是相连的，那么每一行的最后应当是一个整数用于指定两个高对称点连线中间的 k 点个数。如果一个高对称点与其后的高对称点断开，则在该行末尾用 `|` 指定。

如图中第一个 GAMMA 点通过 15 个点连接到了 X 点，然后 X 点与其后的 Y 点处于断开状态。

2.1 Pymatflow 用户手册

Pymatflow 是一个第一性原理模拟的工作流自动化管理软件，有两种工作模式。其一是终端命令行模式，目的是减少用户进行 DFT 模拟所需要的输入文件准备和结果后处理时间，让用户在服务器集群上通过一行命令就可以提交任务和进行结果的提取。其二是使用 **Pymatflow** 提供的 API 进行高通量计算，这部分的使用需要用户具有一定的编程能力。目前支持的计算程序有 Abinit、Quantum ESPRESSO、CP2K、SIESTA、VASP

由于个人精力有限，本手册内容并不完善，目前主要对命令行工具 **matflow**、**postflow** 和 **structflow** 的使用方式进行了初步介绍，以方便用户试用。

2.1.1 安装 Pymatflow

```
~$ pip install pymatflow
```

目前推荐的方式是在服务器中使用 **pip** 安装已发布的 **pymatflow**。

如果自己不是系统管理员，没有 root 权限：

```
~$ pip install pymatflow --user
```

并保证 `~/local/bin` 在环境变量 **PATH** 之中。

pymatflow 支持本地与远程服务器交互操作，但是需要做一定配置，后续做补充。

使用须知

首先用户使用本软件需要对结果自行负责，Pymatflow 不保证结果的正确性。需要用户具有一定的背景知识，能够鉴别计算任务的合理性。

1. 结构文件

可以通过 `--cif`、`--xsd`、`--xsf` 或者 `--xyz` 三个参数之一传递结构文件路径给 `matflow` 或者 `post-flow`。用户需要注意的是这四个参数一次只能设置一个。如果是进行过渡态计算，则通过 `--images` 参数输入结构。

1. 赝势文件

`matflow` 提供了 `--pot` 选项供用户选择赝势文件，有三种设置方式。第一种是指定一个文件夹，该文件夹中需要包含所有的赝势文件，并且建议用户在该文件夹中仅存放赝势文件，因为目前 `matflow` 会将所有该文件夹下的内容都拷贝到当前路径。其二是默认形式，即不需要指定 `--pot` 选项，但用户需要保证所有赝势文件都在当前路径下。第三种方式是 `--pot auto`，此时 `matflow` 会自动会你准备好赝势文件。

对于第三种方式，我们需要对服务器进行一下配置，将赝势库都准备在用户家目录下面。各个程序的赝势存放位置分别是：`~/ .pot-vasp`、`~/ .pot-qe`、`~/ .pot-siesta`、`~/ .pot-abinit`。需要注意的是目前对 CP2K 的支持不需要指定赝势，默认只会使用 `DZVP-MOLLOPT-SR-GTH` 基组和 `GTH-PBE` 类型的赝势。

1. 高对称点路径

在对晶体材料的体相进行研究时，常会用到倒空间这个强有力的工具。在使用 `Pymatflow` 的过程中，所有涉及倒空间高对称点路径的任务都以一个规范的形式进行管理。你可以通过 `--kpath-manual` 或者 `--kpath-file` 向 `matflow` 传入高对称点路径。

使用 `--kpath-manual` 时，需要在其后按照一定格式手动输入路径：

```
--kpath-manual '0.000000 0.000000 0.000000 GAMMA 15' '0.500000 0.000000 0.000000 X|
↪|' '0.000000 0.500000 0.000000 Y 15' '0.000000 0.000000 0.000000 GAMMA 15' '0.
↪000000 0.000000 0.500000 Z|' '0.500000 0.500000 0.500000 R 15' '0.000000 0.
↪000000 0.000000 GAMMA 15' '0.000000 0.500000 0.500000 T|' '0.500000 0.000000 0.
↪500000 U 15' '0.000000 0.000000 0.000000 GAMMA 15' '0.500000 0.500000 0.000000 V
↪15'
```

上面的参数给出的高对称点路径是： $\Gamma - X|Y - \Gamma - Z|R - \Gamma - T|U - \Gamma - V$ 。

Notes:

- 如果一个 K 点的最后一个符号为一个整数，那么它将通过该整数个点连接到下一个 K 点。
- 如果一个 K 点的最后一个符号为 `|`，那么它与后面的 K 点处于断开状态。
- `kx`，`ky`，`kz` 是倒空间 K 点坐标的分数坐标。

此外，用户可以将高对称点路径信息准备在文件中，然后将文件路径通过 `--kpath-file` 传递给 `Py-matflow`。文件格式如下：

```

11
0.000000 0.000000 0.000000 #GAMMA 15
0.500000 0.000000 0.000000 #X |
0.000000 0.500000 0.000000 #Y 15
0.000000 0.000000 0.000000 #GAMMA 15
0.000000 0.000000 0.500000 #Z |
0.500000 0.500000 0.500000 #R 15
0.000000 0.000000 0.000000 #GAMMA 15
0.000000 0.500000 0.500000 #T |
0.500000 0.000000 0.500000 #U 15
0.000000 0.000000 0.000000 #GAMMA 15
0.500000 0.500000 0.000000 #V 15

```

文件的第一行制定了文件后面给出的行数。

每一行对应于 `kpath` 数据结构中的一个元素：

- 行中前三个元素对应于 `K` 点的 `x`、`y` 和 `z` 坐标
- 行中第四个元素对应于 `K` 点的符号 (需要是大写)
- 行中第五个元素对应于 `K` 点与其后 `K` 点的连接关系，意义和上述 `kpath` 数据结构中一致

用户往往需要通过 `--kpath-file` 参数向 `Pymatflow` 的命令行工具 (如 `matflow`、`post-flow`、`structflow`) 指定包含有高对称 `K` 点的文件路径。要注意这里同样坐标必须是分数坐标。注意 `--kpath-manual` 的优先级高于 `--kpath-file`。

1. 环境变量

用户需要为 `Pymatflow` 指定所用程序的环境变量，比如将下列内容加入到你的 `~/.bashrc` 中：

```

export PMF_VASP_STD=/path/to/your/vasp_std
export PMF_VASP_NCL=/path/to/your/vasp_ncl
export PMF_VASP_GAM=/path/to/your/vasp_gam
export PMF_CP2K=/path/to/your/cp2k.[popt|psmp...]
export CP2K_DATA_DIR=/path/to/your/cp2k/data

```

然后通过下面的命令使其立即生效：

```
~$ source ~/.bashrc
```

2.1.2 matflow 简介

`matflow` 提供了计算任务的自动化生成和提交功能。使用方式归结为一句话就是“主命令 + 子命令 + 可选参数”。主命令当然就是 `matflow` 了，至于子命令，目前一共有 5 个，即：

- `abinit`
- `qe`
- `cp2k`

- siesta
- vasp

使用时也就代表了命令使用到的计算器。用户可以通过 `matflow -h` 查看使用帮助。如果想要具体了解某一个计算器提供的可选参数和使用帮助需要加上子命令选项，比如命令 `matflow vasp -h` 可以查看 `vasp` 对应的帮助文档。下面我们通过一个例子来看如何在服务器上使用 `matflow` 来辅助第一性原理计算。

另外为了减少输入，用户可以使用 `mflow` 代替 `matflow`。

2.1.3 matflow 命令工具使用示例

这里以一个简单的优化 `LiH` 立方晶胞参数的例子，展示如何使用 `Pymatflow` 来加速计算的准备工作。

首先，准备好 `LiH` 的晶体结构文件 `lih.cif`。然后在对应路径运行一下命令：

```
~$ matflow vasp -r 2 --cif lih.cif --encut 300 --ibrion 2 --isif 2 --kpoints-mp 3 3 3 0 0 0 -d lih-cubic
```

即可在指定的 `lih-cubic` 文件夹中自动生成输入文件，并提交任务到服务器中，要注意的是这里默认提交的是 `PBS` 类型的任务。目前支持的服务器只有吕梁天河二号和 `PBS` 集群，或者以单机模式运行。具体使用参见 `--server` 参数的帮助文档。至于结果的提取由 `postflow` 工具提供。目前 `matflow` 对 `vasp` 支持的计算类型包括：静态计算 (`scf`、`nscf`、`bands`)、结构优化、立方晶胞参数优化、六方晶胞参数优化、四方晶胞参数优化、结合 `VTST` 进行过渡态计算、`VASP` 自带声子谱计算、结合 `Phonopy` 进行声子谱计算。

2.1.4 postflow 简介

`postflow` 对 `Pymatflow` 的后处理进行了部分封装，目前还未完全成熟。可能有部分选项无法使用。这里展示一下通常的使用流程。比如对上述的 `LiH` 晶胞参数优化任务的后处理，只需要运行命令：

```
~$ postflow vasp -r 2 -d lih-cubic
```

就可以进行结果的提取，生成的文件统一在 `lih-cubic/post-processing` 目录下。

目前 `Pymatflow` 的后处理功能主要还是由 `API` 提供，但是往 `postflow` 移植的工作正在进行中，后续会完善。

2.1.5 structflow 简介

`structflow` 主要提供了常见的结构文件转换，以及进行固定原子结构优化等功能，其中结构文件转换得益于 `ASE` 项目的 `ase.io` 的帮助。这部分内容会在后续的文章中进行介绍

2.1.6 使用 Pymatflow 命令行的通式

2.1.7 matflow 之 VASP 计算器

VASP 静态计算

```
~$ matflow vasp -r 0 --cif YOUR_STRUCTURE.cif -d DIRECTORY --encut VALUE --ediff_
↪VALUE --kpath-file KPATH_FILE_PATH --kpoints-mp VALUE
```

参数 `-r 0` 表示静态计算，其行为是 `scf`、`nscf`、`bands` 一起进行计算。其中需要注意 `-d` 参数不能指定为当前路径，否则命令会抛出警告并退出。后处理命令如下：

```
~$ postflow vasp -r 0 -d DIRECOTRY --kpath-file KPATH_FILE_PATH --kpoints-mp VALUE
```

VASP 结构优化

```
~$ matflow vasp -r 1 --cif YOUR_STRUCTURE.cif -d DIRECTORY --encut VALUE --ediff_
↪VALUE --ediffg VALUE --ibirion VALUE --isif VALUE --kpoints-mp VALUE
```

参数 `-r 1` 表示进行结构优化。其中需要注意 `-d` 参数不能指定为当前路径，否则命令会抛出警告并退出。后处理命令如下：

VASP 晶胞参数优化

```
~$ matflow vasp -r [2|3|4] --cif YOUR_STRUCTURE.cif -d DIRECTORY --encut VALUE --
↪ediff VALUE --kpoints-mp VALUE
```

参数 `-r 2` 是进行 cubic 晶胞的 cell 参数的优化，`-r 3` 进行 hexagonal 晶胞的 cell 参数的优化，`-r 4` 进行 tetragonal 晶胞的 cell 参数的优化。其中需要注意 `-d` 参数不能指定为当前路径，否则命令会抛出警告并退出。后处理命令如下：

```
~$ postflow vasp -r [2|3|4] -d DIRECOTRY
```

VASP 过渡态计算 (VTST)

```
~$ matflow vasp -r 5 --images INITIAL.cif FINAL.cif -d DIRECTORY --encut VALUE --
↪ediff VALUE --kpoints-mp VALUE
```

参数 `-r 5` 代表进行过渡态计算。其中 `--images` 用于指定初始和终态结构。

VASP 声子谱计算 (内置 IBRION=[5, 6, 7, 8])

```
~$ matflow vasp -r 6 --cif YOUR_STRUCTURE.cif -d DIRECTORY --encut VALUE --ediff_
↪VALUE --kpoints-mp VALUE --ibrion [5|6|7|8] --supercell-n VALUE
```

参数-r 6 是利用 vasp 内置的算法进行声子谱计算, 需要设置--ibrion 为 5|6|7|8 之一。其中需要注意-d 参数不能指定为当前路径, 否则命令会抛出警告并退出。后处理命令如下:

```
~$ postflow vasp -r 6 --cif YOUR_STRUCTURE.cif -d DIRECTORY --kpath-file KPATH_
↪FILE_PATH
```

VASP 声子谱计算 (Phonopy)

```
~$ matflow vasp -r 7 --cif YOUR_STRUCTURE.cif -d DIRECTORY --encut VALUE --ediff_
↪VALUE --kpoints-mp VALUE --supercell-n VALUE
```

2.1.8 matflow 之 QE 计算器

QE 几何结构优化 (relax)

```
~$ matflow qe -r 1 --cif YOUR_STRUCTURE.cif -d DIRECTORY --ecutwfc VALUE --kpoints-
↪mp VALUE
```

QE 几何结构优化 (vc-relax)

```
~$ matflow qe -r 2 --cif YOUR_STRUCTURE.cif -d DIRECTORY --ecutwfc VALUE --kpoints-
↪mp VALUE
```

QE 晶胞参数优化

```
~$ matflow qe -r [3|4|5] --cif YOUR_STRUCTURE.cif -d DIRECTORY --ecutwfc VALUE --
↪kpoints-mp VALUE
```

参数-r 3 是进行 cubic 晶胞的 cell 参数的优化, -r 4 进行 hexagonal 晶胞的 cell 参数的优化, -r 5 进行 tetragonal 晶胞的 cell 参数的优化。其中需要注意-d 参数不能指定为当前路径, 否则命令会抛出警告并退出。后处理命令如下:

QE 声子谱计算 (DFPT)

```
~$ matflow qe -r 0 --static scf --cif YOUR_STRUCTURE -d DIRECTORY --ecutwfc VALUE -
↪ -kpionts-mp VALUE
~$ # after the above task terminates
~$ matflow qe -r 7 --cif -d DIRECTORY --nq VALUE --tr2-ph VALUE
```

QE 声子谱计算 (Phonopy)

```
~$ matflow qe -r 8 --cif YOUR_STRUCTURE.cif -d DIRECTORY --ecutwfc VALUE --kpoints-
↪ mp VALUE --supercell-n VALUE
```

2.1.9 matflow 之 CP2K 计算器

CP2K 结构优化 (GEO_OPT)

```
~$ matflow cp2k -r 1 --cif YOUR_STRUCTURE.cif -d DIRECTORY --cutoff VALUE --
↪ kpoints-scheme VALUE
```

CP2K 结构优化 (CELL_OPT)

```
~$ matflow cp2k -r 2 --cif YOUR_STRUCUTRE.cif -d DIRECTORY --cutoff VALUE --
↪ kpoints-scheme VALUE
```

CP2K 声子谱计算 (phonopy)

```
~$ matflow cp2k -r 7 --cif YOUR_STRUCTURE.cif -d DIRECTORY --cutoff VALUE --
↪ kpoints-scheme VALUE --supercell-n VALUE
```

CP2K 振动分析 (VIBRATIONAL_ANALYSIS)

```
~$ matflow cp2k -r 8 --cif YOUR_STRUCTURE.cif -d DIRECTORY --cutoff VALUE --
↪ kpoints-scheme VALUE
```

CP2K 晶胞参数优化

```
~$ matflow cp2k -r [3|4|5] --cif YOUR_STRUCTURE.cif -d DIRECTORY --ecutwfc VALUE --  
↪ kpoints-mp VALUE
```

参数-r 3 是进行 cubic 晶胞的 cell 参数的优化, -r 4 进行 hexagonal 晶胞的 cell 参数的优化, -r 5 进行 tetragonal 晶胞的 cell 参数的优化。其中需要注意-d 参数不能指定为当前路径, 否则命令会抛出警告并退出。

2.1.10 matflow 之 SIESTA 计算器

SIESTA 结构优化

```
~$ matflow siesta -r 1 --cif YOUR_STRUCTURE.cif -d DIRECTORY --meshcutoff VALUE --  
↪ kpoints-mp VALUE
```

SIESTA 晶胞参数优化

```
~$ matflow siesta -r [2|3|4] --cif YOUR_STRUCTURE.cif -d DIRECTORY --ecutwfc VALUE_↵  
↪ --kpoints-mp VALUE
```

参数-r 2 是进行 cubic 晶胞的 cell 参数的优化, -r 3 进行 hexagonal 晶胞的 cell 参数的优化, -r 4 进行 tetragonal 晶胞的 cell 参数的优化。其中需要注意-d 参数不能指定为当前路径, 否则命令会抛出警告并退出。

SIESTA 声子谱计算 (phonopy)

```
~$ matflow siesta -r 5 --cif YOUR_STRUCTURE.cif -d DIRECTORY --meshcutoff VALUE --  
↪ kpoints-mp VALUE --supercell-n VALUE
```

2.1.11 特定类型计算

Pymatflow 提供部分特定类型计算任务, 文档暂无。

2.1.12 structflow 工具

结构格式转换

structflow 支持 cif、xsf、xsd、cube 等结构文件的相互转换。例如, 下面的命令会将 cif 格式的结构文件转换为 xsd 格式。

```
~$ sflow convert -i INPUT_STRUCTURE.cif -o OUTPUT_STRUCTURE.xsd
```

构建 Supercell

```
~$ sflow supercell -i INPUT_STRUCTURE.cif -o OUTPUT_STRUCTURE.cif -n N1  
N2 N3
```

删除晶体结构中的指定元素

```
~$ sflow remove -i INPUT_STRUCTURE.cif -o OUTPUT_STRUCTURE.cif -ele-  
ments ELEMENTS_TO_REMOVE
```

指定移除晶体结构中原子

```
~$ sflow remove -i INPUT_STRUCTURE.cif -o OUTPUT_STRUCTURE.cif -atoms  
ATOMS_TO_REMOVE
```

更改晶体结构中原子的位置

```
~$ sflow move -i INPUT_STRUCTURE.cif -o OUTPUT_STRUCTURE.cif -atoms  
ATOMS_TO_REMOVE -disp LENGTH -direction DIRECTION
```

2.1.13 服务器交互实用工具

- thq.py
- thpull.py
- thcancel.py
- thcmd.py
- threport.py
- sz-cmd.py
- sz-del.py
- sz-q.py
- sz-pull.py

这些工具是本地与远程服务器交互的工具，具体使用需要参数配置，后续补充。

2.1.14 API

文档暂无

2.1.15 官方微信公众平台

Pymatflow “官方微信公众号” “开源科学计算” 目前已经开始试运营。欢迎订阅查看更多关于 Pymatflow 的最新信息和使用教程。



2.1.16 未来

首先欢迎有兴趣的朋友可以一起参与开发。也欢迎用户对程序的使用提出建议，或者提交功能需求。

- 项目地址: <https://gitlab.com/DeqiTang/pymatflow>
- 文档地址: <http://pymatflow.readthedocs.org/>

2.1.17 问题反馈

本文档所有权归属 Pymatflow 项目。对项目有任何问题反馈，请发送邮件至

✉ pymatflow@163.com

Pymatflow 官方微信公众账号”开源科学计算”目前已经开始试运营。欢迎订阅查看更多关于 Pymatflow 的最新信息和使用教程。



往期公众号文章

[使用 Phonopy+CP2K 计算固体声子谱 \(Pymatflow 篇\)](#)

[使用 Pymatflow 辅助 Quantum ESPRESSO 计算能带](#)

[Phonopy+Quantum ESPRESSO 计算固体声子谱 \(Pymatflow 简化版\)](#)

[Something changed?](#)

Indices and tables

- `genindex`
- `modindex`
- `search`