

mollerチュートリアル

CCMSハンズオン: moller講習会

2024年10月18日 @物性研A614+Zoom

東京大学物性研究所 附属物質設計評価施設

ソフトウェア開発・高度化チーム

チュートリアル

- ▶ mollerのサンプルは物性研データリポジトリにあります。

- ▶ Moller Gallery

<https://datarepo.mdcl.issp.u-tokyo.ac.jp/repo/38>

- ▶ ダウンロード方法

```
$ git clone https://isspns-gitlab.issp.u-tokyo.ac.jp/http-tools-dev/moller-gallery.git
```

- ▶ 今回のチュートリアルのサンプルファイルは物性研システムBに置いてあります。

- ▶ /home/issp/materiapps/oneapi_compiler_classic-2023.0.0--openmpi-4.1.5/moller/moller-gallery/
(実際は1行)

- ▶ 実行手順は各チュートリアルの README を参照してください。

チュートリアル一覧

1. moller/simple

- ▶ mollerのシンプルな実行例

2. moller/dsqss/AFH-chain

- ▶ DSQSS (量子モンテカルロ法) を用いた反強磁性Heisenberg鎖の計算

3. moller/hphi/AFH-chain

- ▶ $H\Phi$ (厳密対角化) を用いた反強磁性Heisenberg鎖の計算

4. moller+cif2x/Basic_usage/kpoints

- ▶ QE 計算における K点数依存性の評価

5. moller+cif2x/Basic_usage/cutoff_energy

- ▶ OpenMX 計算におけるカットオフ依存性の評価

Appendix 1. クラスタ計算機で実行するには

Appendix 2. cif2x のインストール

1. moller/simple

- ▶ 内容
 - ▶ "hello world." を書き込んだファイル result.txt を作成する。
 - ▶ result.txt に "hello world again." を追記する。
 - ▶ 出力先は dataset-0001/ ~ dataset-0020/

1. moller/simple

- ▶ サンプルファイル
 - ▶ input.yaml
 - ▶ moller の入力ファイル。実行プラットフォーム、実行内容などを記述。
 - ▶ make_inputs.sh
 - ▶ データディレクトリ dataset-0001 ~ dataset-0020 と、リストファイル list.dat を作成する。

1. moller/simple

▶ 入力ファイルの例

```
name: testjob
description: Sample task file

platform:
  system: ohtaka
  queue: i8cpu
  node: 1
  elapsed: 00:10:00

prologue:
  code: |
    source /home/issp/materiapps/
oneapi_compiler_classic-2023.0.0--
openmpi-4.1.5/parallel/
parallelvars-20210622-1.sh

jobs:
  hello:
    description: hello world
    node: [1,1]
    run: |
      echo "hello world." > result.txt

  hello_again:
    description: hello world again
    node: [1,1]
    run: |
      echo "hello world again." >> result.txt
```

1. moller/simple

- ▶ 実行例

- ▶ moller をセットアップする

```
$ source /home/issp/materiapps/oneapi_compiler_classic-2023.0.0--  
openmpi-4.1.5/moller/mollervars.sh
```

- ▶ 入力ファイル input.yaml からジョブスクリプト job.sh を作成する

```
$ moller -o job.sh input.yaml
```

- ▶ 出力先ディレクトリと list.dat を作成する

```
$ sh make_inputs.sh
```

1. moller/simple

- ▶ 実行例

- ▶ バッチジョブを投入する

```
$ sbatch job.sh list.dat
```

- ▶ リストファイルを省略した場合はデフォルト値 (list.dat) が使われる

- ▶ 実行状況を確認する

```
$ squeue
```

```
$ moller_status input.yaml list.dat
```

1. moller/simple

- ▶ 実行例

- ▶ dataset-0001/ ~ dataset-0020/ に result.txt が作成され、

```
hello world.  
hello world again.
```

が書き込まれていれば OK。

```
$ moller_status input.yaml list.dat
```

```
| job          | hello      | hello_again |  
|-----|-----|-----|  
| dataset-0001 | o         | o         |  
| dataset-0002 | o         | o         |  
| dataset-0003 | o         | o         |  
| dataset-0004 | o         | o         |  
| dataset-0005 | o         | o         |  
| ...         |           |           |  
| dataset-0020 | o         | o         |
```

2. moller/dsqss/AFH-chain

- ▶ 内容

- ▶ 量子多体問題の経路積分モンテカルロ法ソフトウェア DSQSS を用いて、反強磁性ハイゼンベルク鎖の磁気感受率の温度依存性を計算する
- ▶ 異なるパラメータ M , L , T の計算を並列に実行する

2. moller/dsqss/AFH-chain

- ▶ 1つのパラメータに対する計算
 - ▶ 入力ファイル例
 - ▶ job.sh (右図)
 - ▶ std.toml (下図)
 - ▶ 実行例

```
$ sbatch job.sh
```

```
#!/bin/sh
#SBATCH -p i8cpu
#SBATCH -N 1
#SBATCH -n 8
#SBATCH -t 00:30:00

source /home/issp/materiapps/
oneapi_compiler_classic-2023.0.0--
openmpi-4.1.5/dsqss/dsqssvars.sh

dla_pre std.toml
srun dla param.in
```

```
[hamiltonian]
model = "spin"
M = 1
Jz = -1.0
Jxy = -1.0
h = 0.0
```

```
[lattice]
lattice = "hypercubic"
dim = 1
L = 8
bc = true
```

```
[parameter]
nset = 10
ntherm = 1000
ndecor = 1000
nmcs = 10000
wvfile = "wv.xml"
dispfile = "disp.xml"
beta = 0.5
```

2. moller/dsqss/AFH-chain

- ▶ moller による一括計算

- ▶ M, L, T を変えたパラメータセットを作成する

```
$ bash make_inputs.sh
```

- ▶ moller でジョブスクリプトを作成する

```
$ moller -o job.sh input.yaml
```

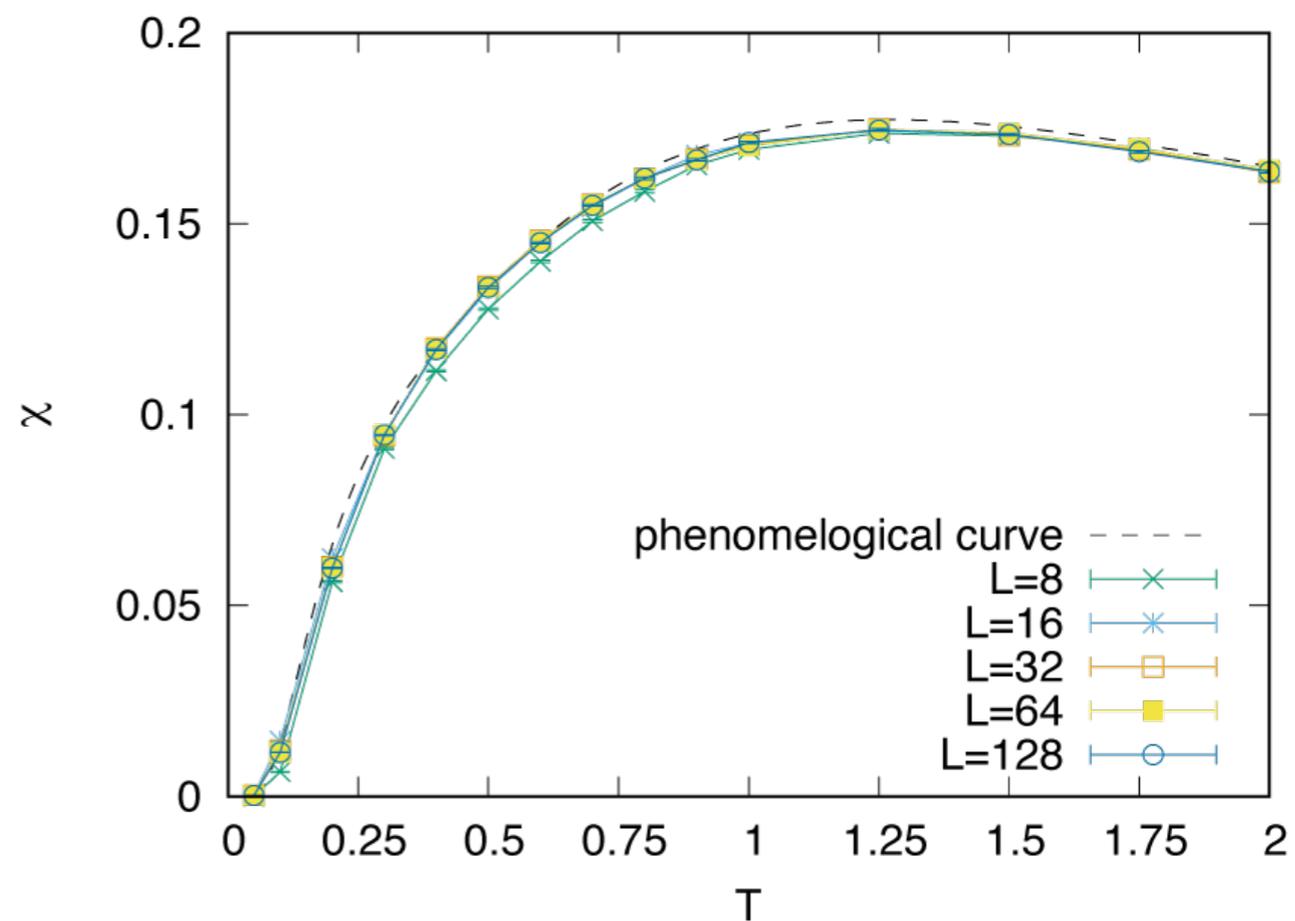
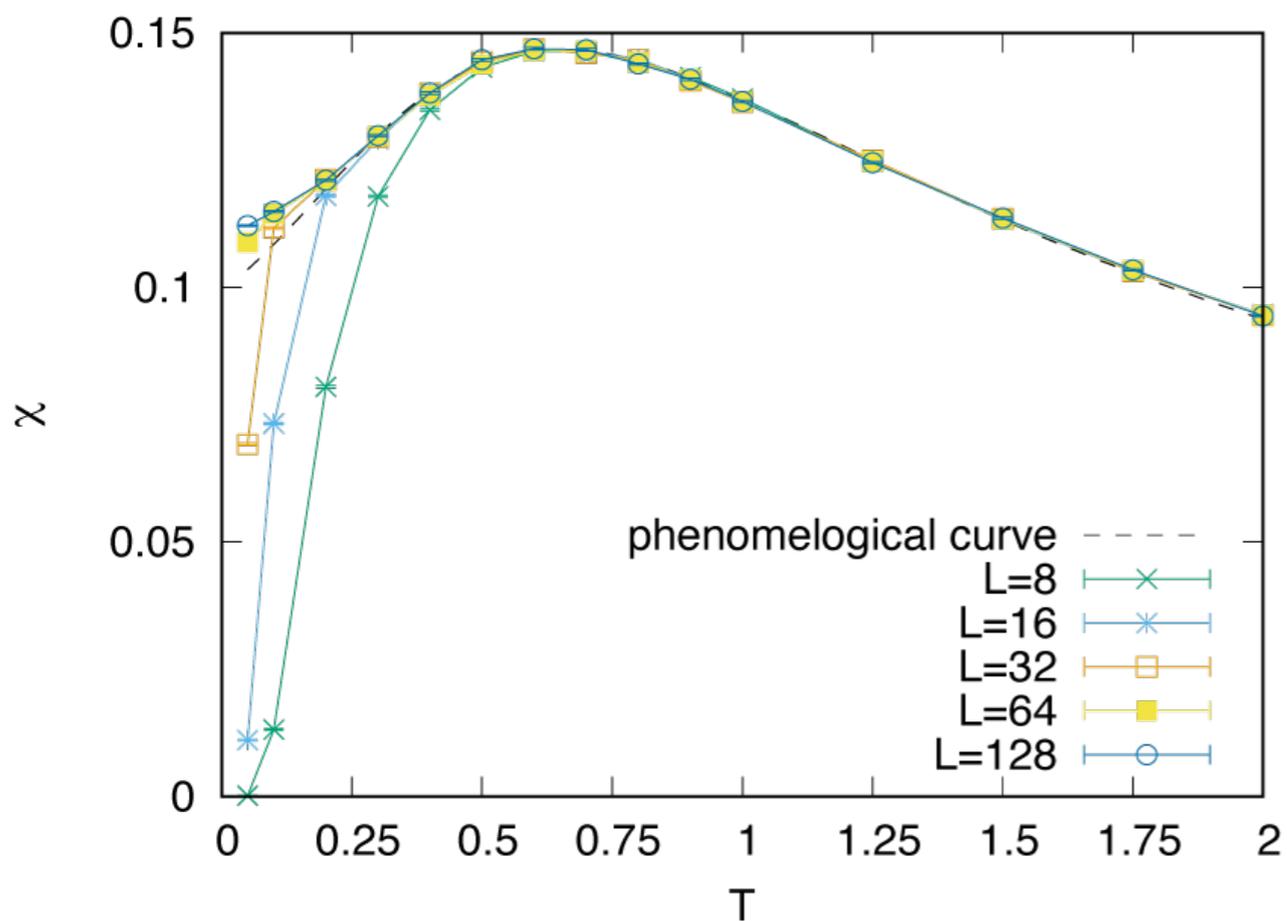
- ▶ バッチジョブを投入する

```
$ sbatch job.sh list.dat
```

- ▶ 結果を集計してプロットする

2. moller/dsqss/AFH-chain

▶ 実行例



3. moller/hphi/AFH-chain

- ▶ 内容

- ▶ 量子格子模型を厳密対角化法により解くソフトウェア HΦ を用いて、反強磁性ハイゼンベルク鎖の励起ギャップ Δ のシステムサイズ依存性を計算する
- ▶ 異なるパラメータ L の計算を並列に実行する

3. moller/hphi/AFH-chain

▶ 実行手順

1. データセットを作成する

```
$ bash ./make_inputs.sh
```

2. moller を使ってジョブスクリプトを作成する

```
$ moller -o job.sh input.yaml
```

3. バッチジョブを実行する

```
$ sbatch job.sh list.dat
```

4. 実行状況を確認する

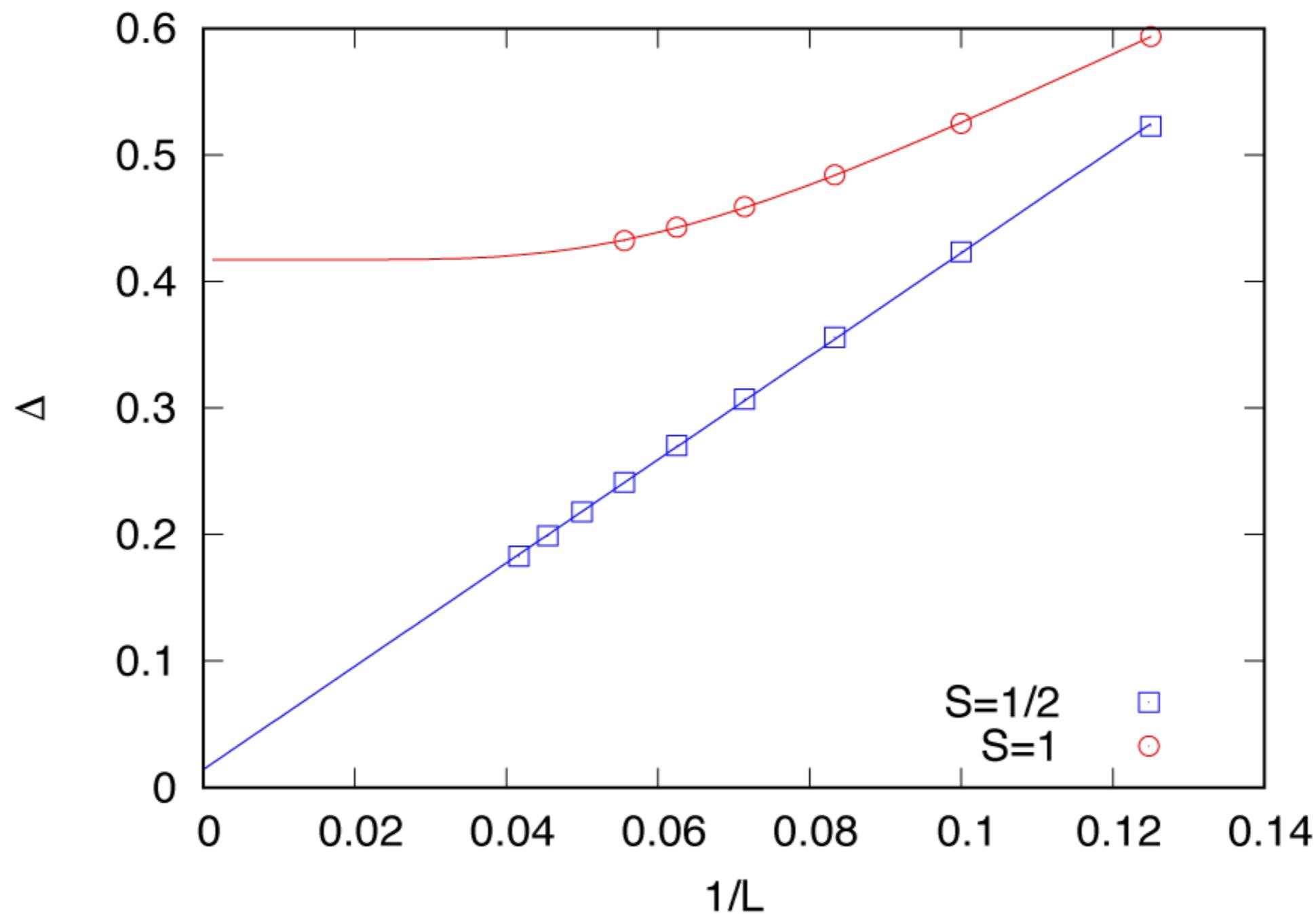
```
$ moller_status input.yaml list.dat
```

5. 結果を集計してプロットする

```
$ bash ./extract_gap.sh  
$ gnuplot --persist gap.plt
```

3. moller/hphi/AFH-chain

▶ 実行例



4. moller+cif2x/kpoints

- ▶ 内容

- ▶ Quantum ESPRESSO (QE) を用いた計算における k点数依存性を評価する
- ▶ cif2x により、入力ファイル input_cif2x.yaml から QE の入力ファイルを作成する
- ▶ moller を用いて、異なる kpoints パラメータの計算を並列に実行する

4. moller+cif2x/kpoints

▶ 実行手順

0. 擬ポテンシャルファイルを取得する

```
$ cd pseudo  
$ sh fetch-pseudo.sh
```

擬ポテンシャルのテーブルを cif2x のパッケージからコピーする

```
$ cp cif2x/sample/cif2x/qe/pseudo/*.csv .
```

4. moller+cif2x/kpoints

▶ 実行手順

1. cif2x を使って QE の入力ファイルの雛形 scf.in を作成

```
$ cif2x -t QE input_cif2x.yaml mp-126_Pt.cif
```

2. K点数の範囲 kpoints.txt とリストファイル list.dat を作成

```
3 10 1
```

```
kpoint_3
```

```
kpoint_4
```

```
...
```

3. moller を使ってジョブスクリプトを作成

```
$ moller -o job.sh input_moller.yaml
```

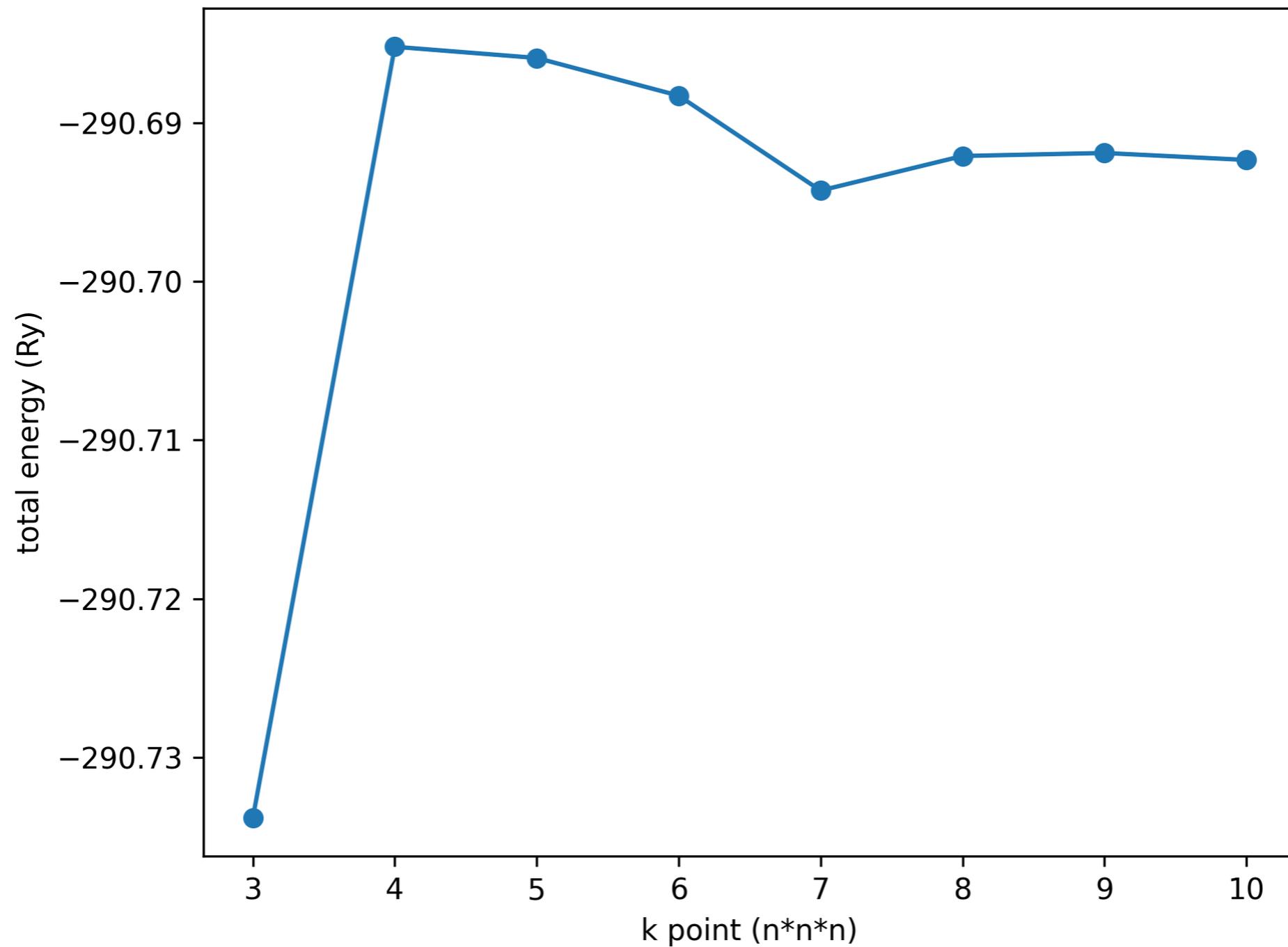
4. バッチジョブを投入

```
$ sbatch job.sh list.dat
```

5. 結果(utot_values.txt)をプロット

4. moller+cif2x/kpoints

▶ 実行例



5. moller+cif2x/cutoff_energy

▶ 内容

- ▶ OpenMX を用いた計算における cutoff パラメータ依存性を評価する
- ▶ cif2x により、入力ファイル input_cif2x.yaml から OpenMX の入力ファイルを作成する
- ▶ moller を用いて、異なる cutoff パラメータの計算を並列に実行する

5. moller+cif2x/cutoff_energy

▶ 実行手順

1. cif2x を使って OpenMX の入力ファイルの雛形 Pt.dat を作成

```
$ cif2x -t openmx input_cif2x.yaml mp-126_Pt.cif
```

2. カットオフの範囲 cutoff.txt とリストファイル list.dat を作成

```
200 700 50
```

```
cutoff_200
```

```
cutoff_250
```

```
...
```

3. moller を使ってジョブスクリプトを作成

```
$ moller -o job.sh input_moller.yaml
```

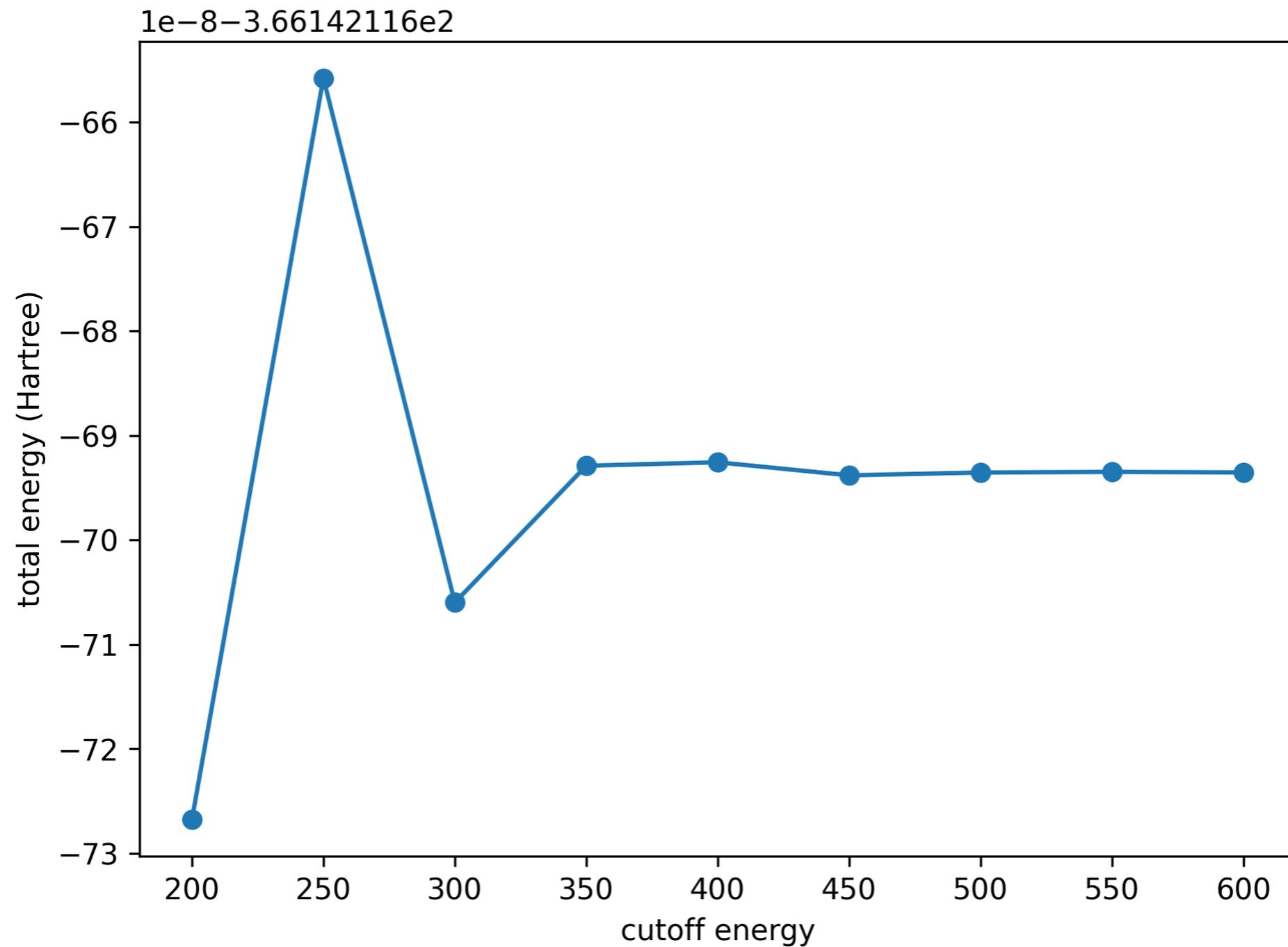
4. バッチジョブを投入

```
$ sbatch job.sh list.dat
```

5. 結果(utot_values.txt)をプロット

5. moller+cif2x/cutoff_energy

▶ 実行例



クラスター計算機で実行するには

- ▶ PBS系のジョブスケジューラ(torqueなど)を使っている場合 (intel mpi)
 - ▶ 入力ファイルの platform セクションの指定:

```
platform:  
  system: pbs  
  core: [ノードあたりのコア数]
```

cif2xのインストール

- ▶ moller+cif2x を ohtaka で使う手順
 - ▶ Python環境と moller をセットアップする

```
$ source /home/issp/materiapps/oneapi_compiler_classic-2023.0.0--  
openmpi-4.1.5/moller/mollervars.sh
```

仮想環境を用意する

```
$ python3 -m venv venv  
$ source venv/bin/activate
```

pip を更新する

```
$ python3 -m pip install pip --upgrade
```

- ▶ cif2x をリポジトリから取得してインストールする

```
$ git clone https://github.com/issp-center-dev/cif2x.git  
$ cd cif2x  
$ python3 -m pip install .
```