

DCoreのインストール方法

吉見 一慶

東京大学物性研究所 特任研究員 (PCoMS PI)

ソフトウェア高度化推進チーム

14:10~14:40

1. DCoreのインストール概要

2. MateriApps LIVE!の準備

3. 簡易デモ

14:40~15:00

4. ISSPスパコンでの利用

DCoreのインストール方法

吉見 一慶

東京大学物性研究所 特任研究員 (PCoMS PI)

ソフトウェア高度化推進チーム

1. DCoreのインストール概要

2. MateriApps LIVE!の準備

3. 簡易デモ

4. ISSPスパコンでの利用

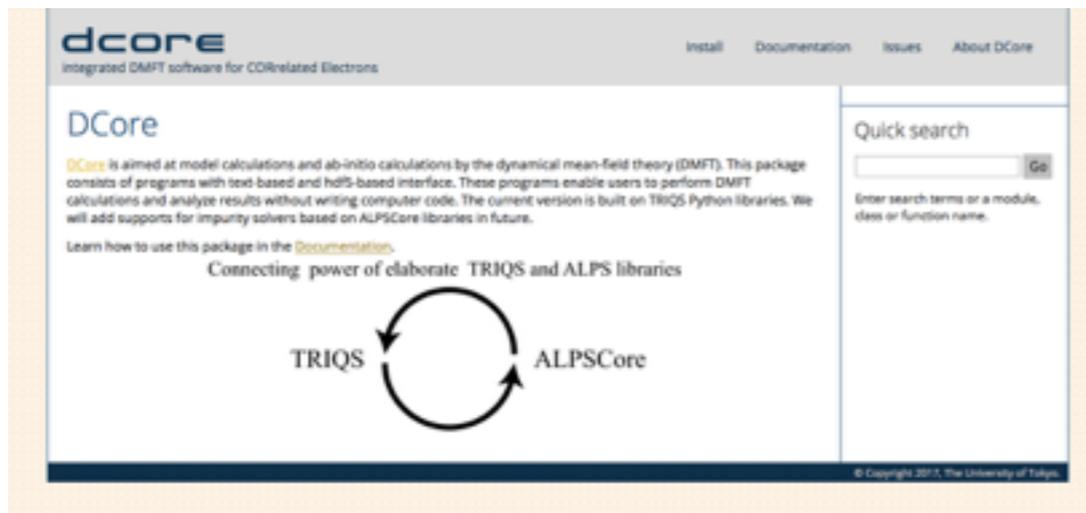
1-1. DCoreの取得方法

1. 検索方法：「MateriApps DCore」で検索



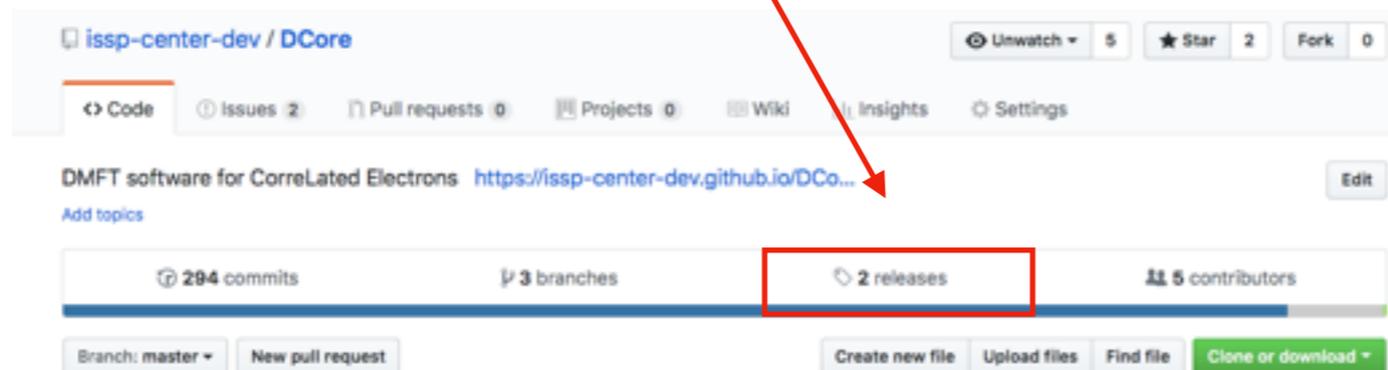
➡ 「公式サイト」をクリック

2. 説明文のDCoreをクリック



➡ GitHubページへ

3. GitHubページのreleasesをクリック (慣れていない人はclone)



➡ Releaseページから
DCoreをダウンロード

1-2. DCoreに必要な環境

1. 必要なライブラリをインストール (全て必要)

- **TRIQS libraries** (checkout tag 1.4.2 on master branch)
<https://triqs.github.io/triqs/master/index.html>
- **TRIQS/DFTTools** (checkout commit d005756 on master branch)
https://triqs.github.io/dft_tools/master

2. ソルバーをインストール (使用するものだけでOK)

- **TRIQS/Hubbard-I solver**
<https://github.com/TRIQS/hubbardi>
- **TRIQS/cthyb** (checkout tag 1.4.2 on master branch)
<https://triqs.github.io/cthyb/master/index.html>
- **ALPSCore/cthyb**
<https://github.com/ALPSCore/CT-HYB>
 - triqs_interface
https://github.com/shinaoka/triqs_interface

1-3. DCoreのインストール

1. DCoreのフォルダへ移動しbuildディレクトリを作成

```
$ mkdir dcore.build && cd dcore.build
```

2. cmakeでdcoreを作成

```
$ cmake -DTRIQS_PATH=path_to_triqs ../
```

3. dcoreのコンパイル後にテスト

```
$ make  
$ make test  
$ make install
```

以上で基本的なインストール作業は終了！

DCoreのインストール方法

吉見 一慶

東京大学物性研究所 特任研究員 (PCoMS PI)

ソフトウェア高度化推進チーム

1. DCoreのインストール概要

2. MateriApps LIVE!の準備

3. 簡易デモ

4. ISSPスパコンでの利用

2-1. 事前準備 (VirtualBoxの利用版)

- ・ インストール作業

1. Virtual Boxのインストール

<https://www.virtualbox.org/wiki/Downloads>

2. MateriApps Live!のインストール(今回はUSBで配布)

<https://sourceforge.net/projects/materiappslive/>

から最新のMateriApps Live!をインストール。

インストール後、ovaファイルをダブルクリックする

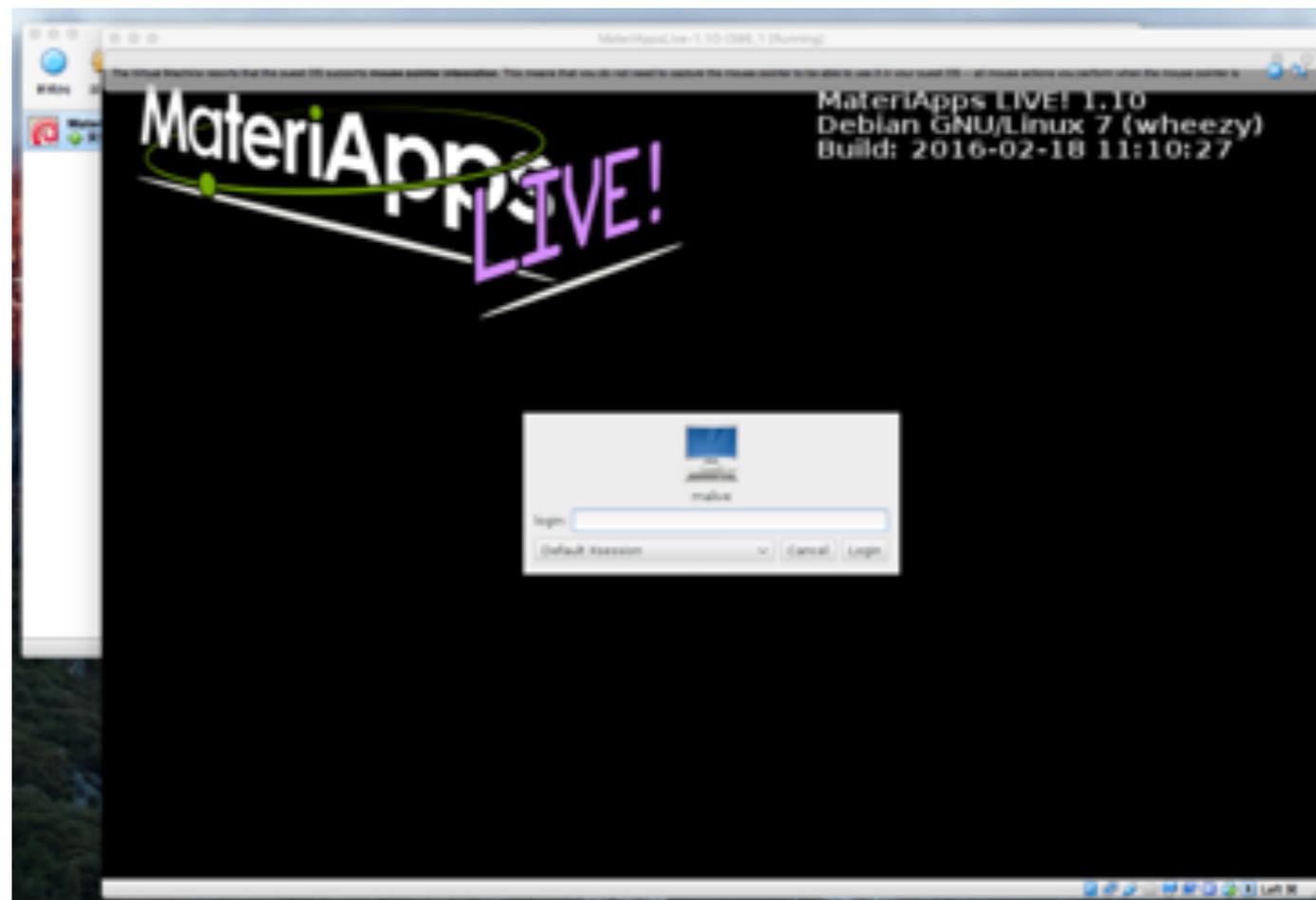
と、VirtualBoxへインポートされます。

2-2. 共有フォルダの設定

- ・ おすすめ：手元の環境とVirtual Box間でファイルのやり取りをできるように共有フォルダの設定を行います
 1. 「設定」 - 「共有フォルダー」 タブを開き、「+」をクリックします。
 2. 「フォルダーのパス」の右側の「v」をクリックし、「その他」を選択します。
 3. 「自動マウント」をチェックし「OK」を選択。
 4. 仮想マシンを起動すると、/mnt/sf_…の下に2で選択したフォルダが表示されます。

2-3. 起動確認

- MateriAppsLive-2.0b2-amd64をダブルクリック



login: user、 Password: live

2-4. キーボードの設定

- 日本語キーボード用に環境変更します。
 1. 「System Tools」 - 「LXTerminal」 を選択します。
 2. 以下のコマンドを入力します。

```
$ setxkbmap -layout jp
```

2-5. サンプルファイルのコピー

- DCoreのサンプルファイル一式は
`/usr/share/doc/dcore/doc/tutorial`
にあります。これらをコピーして使用します。
- ここではdcoreフォルダを作成してそこにコピーします。
 1. dcoreフォルダの作成と移動
`$ mkdir dcore && cd dcore`
 2. サンプルファイルのコピー
`$ cp -r /usr/share/doc/dcore/doc/tutorial .`

DCoreのインストール方法

吉見 一慶

東京大学物性研究所 特任研究員 (PCoMS PI)

ソフトウェア高度化推進チーム

1. DCoreのインストール概要
2. MateriApps LIVE!の準備
- 3. 簡易デモ**
4. ISSPスパコンでの利用

基本的な計算の流れ

入力ファイルの作成
(テキストインターフェイス)

第一原理計算から導出したパラメータを使用する場合、
Wannier90形式のファイルも用意
(hopping, Interaction)

dcore実行用 : dcore_pre
入力ファイル(hdf5形式)の作成

選択されたsolverに応じた
hdf5形式の入力ファイルを作成

DMFT計算 : dcore
(出力結果 : hdf5形式)

選択されたsolverを用いたDMFT計算
(TRIQSで用いられているフォーマットで結果出力)
Green関数、自己エネルギーが出力される。

計算後処理 : dcore_post
(出力形式 : テキスト)

出力された結果の加工(主に解析接続)やグラフ表示。
テキスト形式での出力もする。

入力ファイル一覧

全5つのblockから構成される

[model] : 模型に関する設定
格子
軌道の数・種類
電子数
相互作用の種類
相互作用の大きさ

[system] : 系に関する設定
虚時間の分点の数
松原振動数の分点の数
逆温度
化学ポテンシャル
etc...

[impurity_solver] : ソルバーの設定
TRIQS/hubbard-I, TRIQS/cthyb,
ALPS/cthyb

[control] : ソルバーの計算条件の設定
次のステップに進む際のmixingパラメータ
DMFT-loopの最大ループ数
再計算フラグ

[tool] : ポスト処理時の設定
最大・最小実振動数
k点の始点・終点と分点数
振動数の虚部のシフト量
etc.

ref.) <https://issp-center-dev.github.io/DCore/reference/input.html>

計算例: single-band 2D Hubbard model (1)

<https://issp-center-dev.github.io/DCore/tutorial/square/square.html>

dmft_square.ini

```
[model]
seedname = square
lattice = square
norb = 1
nelec = 1.0
t = -1.0
kanamori = [(2.0, 0.0, 0.0)]

[system]
beta = 40.0
nk = 8
n_iw = 1000
prec_mu = 0.001

[impurity_solver]
name = TRIQS/hubbard-I

[control]
max_step = 7

[tool]
broadening = 0.4
nnode = 4
knode = [(G,0.0,0.0,0.0),(X,0.5,0.0,0.0),(M,0.5,0.5,0.0),(G,0.0,0.0,0.0)]
nk_line = 100
omega_max = 6.0
omega_min = -5.0
Nomega = 400
```

格子
軌道数
電子数
ホッピング
相互作用
逆温度
不純物ソルバー
自己無撞着計算の回数
 $A(k,\omega)$ を計算するパス

計算例: single-band 2D Hubbard model (2)

<https://issp-center-dev.github.io/DCore/tutorial/square/square.html>

0. サンプルディレクトリへ移動

```
> cd tutorial/square
```

1. 入力ファイルからHDF5ファイルを作成

```
> dcore_pre dmft_square.ini
```

```
dmft_square.ini → square.h5
```

2. 自己無撞着計算

```
> dcore dmft_square.ini
```

```
▶ dmft_square.ini → square.out.h5  
▶ square.h5
```

3. 自己無撞着計算の収束を確認

```
> dcore_check dmft_square.ini
```

4. 物理量解析

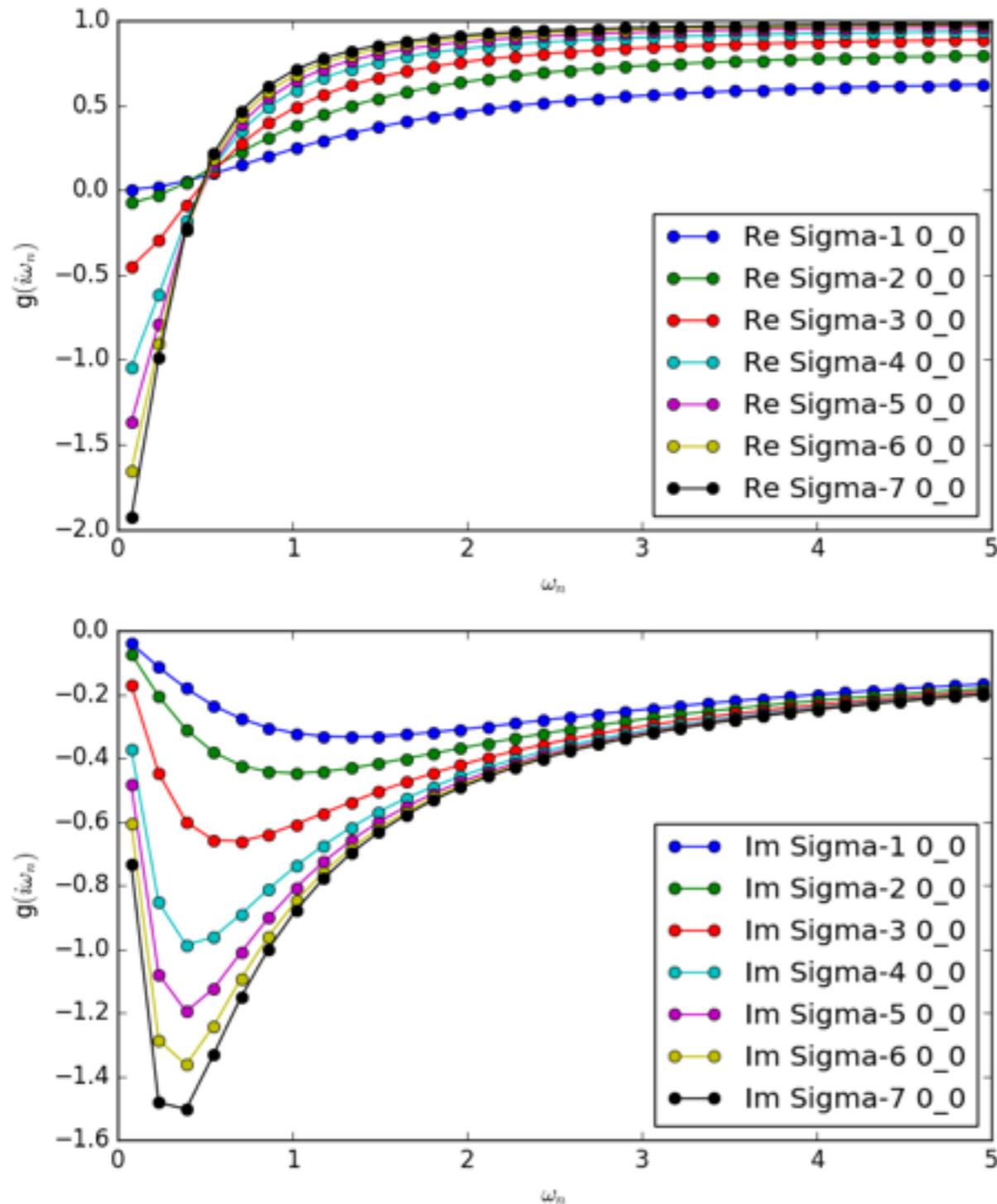
```
> dcore_post dmft_square.ini
```

計算例: single-band 2D Hubbard model (3)

<https://issp-center-dev.github.io/DCore/tutorial/square/square.html>

3. 自己無撞着計算の収束を確認

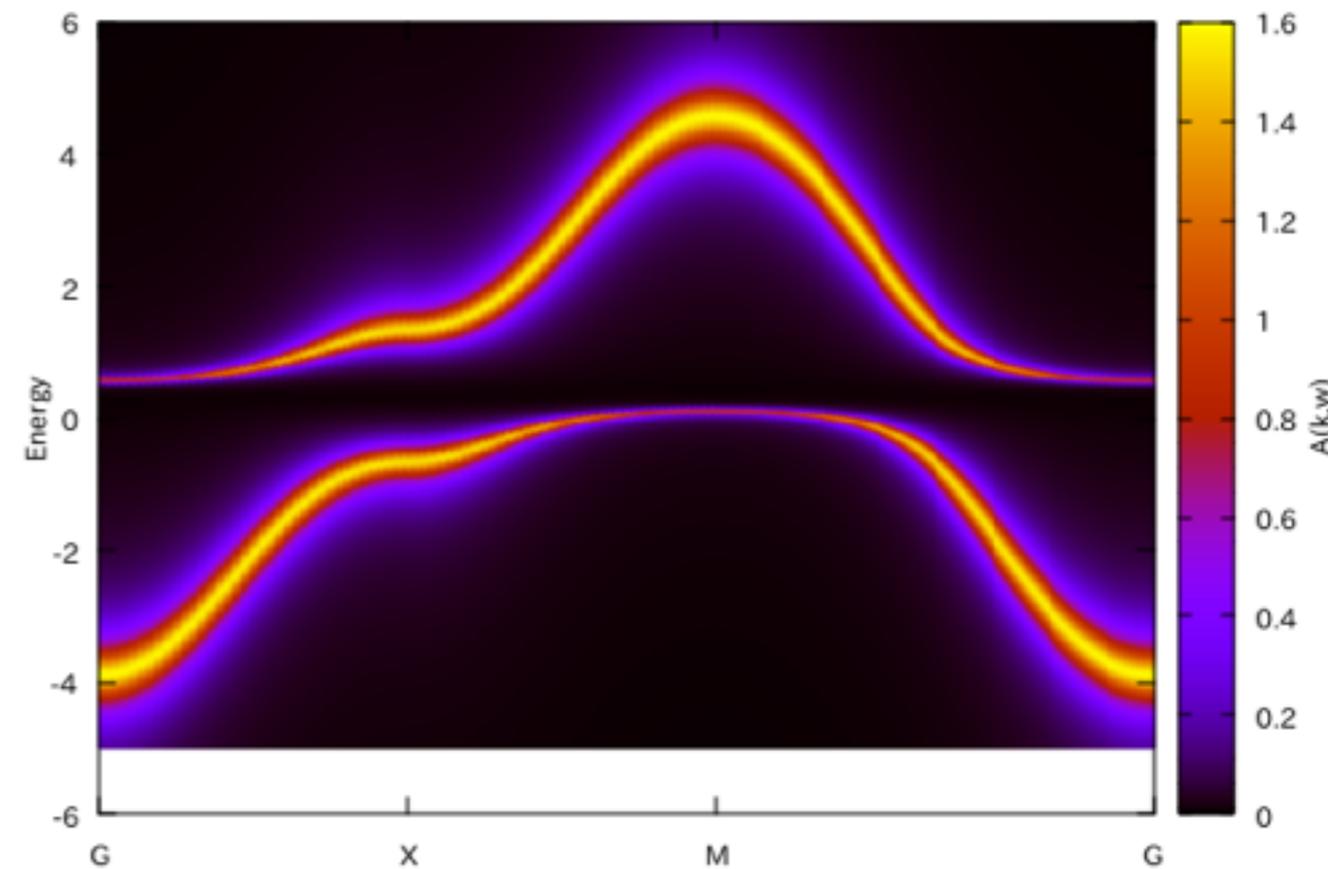
```
> dcore_check dmft_square.ini
```



4. 物理量解析

```
> dcore_post dmft_square.ini
```

```
> gnuplot square_akw.gp
```



DCore: チュートリアル一覧

第一原理計算を出発点にした解析例を用意

1. Quantum ESPRESSOとwannier90を用いた解析例

2. OpenMXとwannier90を用いた解析例

3. スピン軌道相互作用ありの場合の解析例 (単一軌道)

4. スピン軌道相互作用ありの場合の解析例 (多軌道)

5. RESPACKにより求めた有効相互作用を利用した解析例

- [Tutorial with QuantumESPRESSO and wannier90](#)
 - [NiS](#)
 - [SrVO₃](#)
- [Tutorial with OpenMX](#)
 - [SCF computation and Wannier with OpenMX](#)
 - [DMFT calculation](#)
- [Pb with spin-orbit interaction](#)
 - [SCF calculation of Quantum ESPRESSO](#)
 - [Wannierization](#)
 - [DMFT calculation](#)
- [Pb with spin-orbit interaction \(multi-atom with QMC\)](#)
 - [SCF calculation of Quantum ESPRESSO](#)
 - [Wannierization](#)
 - [DMFT calculation](#)
- [Downfolding with RESPACK \(SrVO₃\)](#)
 - [SCF calculation of Quantum ESPRESSO](#)
 - [Wannierization](#)
 - [Dielectric matrix and Effective interaction](#)
 - [DMFT calculation](#)

<https://issp-center-dev.github.io/DCore/documentation.html>

DCoreのインストール方法

吉見 一慶

東京大学物性研究所 特任研究員 (PCoMS PI)

ソフトウェア高度化推進チーム

1. DCoreのインストール概要
2. MateriApps LIVE!の準備
3. 簡易デモ
4. ISSPスパコンでの利用

3-1. sekireiの性能

- Fat ノード (2 ノードまで使用可能)

CPU: Intel Xeon 2.6 GHz (10 cores) ×4

主記憶: DDR4-2133 1 TB (2ノード使用で2TB相当)

- CPU ノード(144 ノードまで使用可能)

CPU: Intel Xeon 2.5 GHz (12 cores) ×2

主記憶: DDR4-2133 128 GB (128ノード使用で16TB相当)

3-2. sekireiを使用するには？(1)

以下の手順で申請すれば利用可能です。

1. 研究代表者の登録
2. 研究課題を申請 (B, C, Eクラスは6月,12月の2回)
3. 利用審査
4. 報告書の提出

利用の流れの詳細は下記URLに記載してありますので、ご参照ください。

<http://www.issp.u-tokyo.ac.jp/supercom/visitor/overview>

3-2. sekireiを使用するには？(2)

小さい計算向けのクラス：Aクラス

Aクラスの概要

■ 申請ポイント：100 ポイント以下

■ 申請回数：半期ごとに1回申請が可能。

ただし、A以外のクラスですでに利用している
研究代表者(グループ)の申請は不可。

■ 報告書は必要なし。

その他申請クラスの詳細については <http://www.issp.u-tokyo.ac.jp/supercom/visitor/about-class> をご参照ください。

3-2. sekireiを使用するには？(3)

100ポイントでどの程度計算可能？

- Fat ノードを 1 ノード 1 日利用：4ポイント消費

→ のべ25日間の使用が可能。

(ポイント消費のルールは ISSP スパコン Webページの「利用案内」 - 「ポイント消費制」に記載)

<http://www.issp.u-tokyo.ac.jp/supercom/visitor/point>

3-3. sekireiで利用可能なソフトウェア

- ・ システムBにプリインストールされているソフトウェア
 - ISSPスパコンページの「利用案内」 - 「インストール済みアプリケーション」に記載
 - プリインストールソフトウェア一覧 (各ソフトウェアの詳細はMateriApps参照)
ISSPが実行方法などのサポートをするもの(黒)、開発者へ問い合わせるもの (青)
1. 第一原理計算関連
 - OpenMX, VASP, [QUANTUM ESPRESSO](#), [RESPACK](#)
 2. 量子格子模型ソルバー関連
 - ALPS, HΦ, mVMC, DSQSS, DCore, [ALPSCore/CT-HYB](#), [TRIQS](#)
 3. 分子動力学関連
 - LAMMPS, [Gromacs](#), [ERmod](#), [feram](#)
 4. その他
 - K ω (Shifted-Krylov), [Rokko](#), Chainer, cuDNN, Julia

3-4. sekireiでのソフトウェア実行 (1)

- ・ 事前準備

1. sekireiへのログイン

配布した紙を参考に、端末を開き以下のコマンドを打ってください(MA LIVE!でも可).

```
$ ssh -Y アカウント名@sekirei.issp.u-tokyo.ac.jp
```

→ パスワードを入力

2. 環境構築

DCore関連のソフトウェアを使用できるように環境をコピーします.

```
$ cp /home/t0011/t001100/.bashrc .
```

```
$ source .bashrc
```

3-4. sekireiでのソフトウェア実行 (2)

- ・ システムB sekireiにDCoreと使用するソフトウェア一式は
プリインストール済。
- ・ 各種ファイルの置き場所 (覚書)
 - DCoreのインストール場所
/home/issp/materiapps/triqs
 - 実行ファイルのインストール場所
/home/issp/materiapps/triqs/home/issp/materiapps/triqs/triqs-
d54b4fb8ca206cb23172e945a6523aa003c9d662-1/cxx1y/bin
 - サンプルスクリプトと入力ファイルの場所
/home/issp/materiapps/triqs/doc/tutorial

3-4. sekireiでのソフトウェア実行 (3)

1. 計算環境の準備 (今回ははじめの作業でここは完了済)

```
$ source /home/issp/materiapps/triqs/triqsvars.sh
```

2. 入力ファイルの準備 (今の場合はtutorial/dcore/squareを使用)

```
$ cp -rf /home/issp/materiapps/triqs/doc/tutorial/dcore/square .
```

3. ジョブの投入 (サンプルスクリプトを既に用意してあるのでそれを使用)

```
$ cd square && cp /home/issp/materiapps/triqs/sample_jobscript/dcore.sh .  
$ qsub dcore.sh
```

ref.) Webページ「ソフトウェア高度化」 - 「システムBでの利用」 - 「DCore」

今回はお試し用のキュー(ccms_i18cpu)を使用(普段はデバッグに利用)。

1. ノード数は最大18ノードまで
 2. CPU数は1ノードあたり24
 3. 実行時間は最大30分まで
- 講習会后1週間まで使用できます。

3-4. sekireiでのソフトウェア実行 (4)

ジョブの確認

\$qstat -u アカウント名

dcore.shの中身

```
#!/bin/sh
#QSUB -queue ccms_i18cpu ← キューの指定 (今回はこのキューのみ使用可能)
#QSUB -node 4 ← ノードの個数の指定
#QSUB -mpi 4 ← プロセス数の指定 (MPI)
#QSUB -omp 24 ← スレッド数の指定 (OpenMP)
#QSUB -place pack ← ジョブプロセス CPUコア配置方針
#QSUB -over false ← 要求したコア数以上の並列数のジョブの実行可否
#PBS -l walltime=00:10:00 ← 最大計算時間 (時間 : 分 : 秒)
#PBS -N dcore ← ジョブ名
```

```
## https://issp-center-dev.github.io/DCore/tutorial/square/square.html
```

```
source /home/issp/materiapps/triqs/triqsvars.sh ← 実行環境呼び出し
```

```
cd ${PBS_O_WORKDIR} ← 今のディレクトリへ移動
```

```
mpijob dcore_pre dmft_square.ini
```

```
mpijob dcore dmft_square.ini
```

```
mpijob -np 1 dcore_check dmft_square.ini --output=check.pdf
```

```
mpijob -np 1 dcore_post dmft_square.ini
```

計算実行

実習では**赤い部分**を書き換えて色々とお試しく下さい。

3-5. (補) sekireiでの利用回数測定

対象ソフトウェア：ソフトウェア高度化対象プログラム

プリインストールソフト
(計測用*)

システムB

- XXXXXXXXXXXX
- 並列数
- 高度化ソフトA



ユーザー

- ユーザーID
- 並列数
- 高度化ソフトA

ユーザーIDを暗号化

個人情報は見えない

(*) 利用率を計測しないソフトの選択