

# HΦの演習問題 (解説)

三澤 貴宏

東京大学物性研究所 特任研究員 (PCoMS PI)



例題1: spin 1/2 dimer (full diag)

例題2: spin 1/2 chain (Lanczos)

例題3: J1-J2 Heisenberg model(Lanczos,TPQ)

例題4: Kitaev model (Lanczos,TPQ)

例題5: Hubbard chain (Lanczos,TPQ)

好きなものからやって下さい

ほとんどlaptop PCでできるはずですが(TPQはちょっと重いですが...)

もちろん、自分のやりたい別の課題もやっても OKです



Basic exercise !

いくつかのサンプルは以下のサイトにあります  
<https://github.com/yomichi/HPhi-samples>

# 例題 1: 反強磁性 $S = 1/2$ Heisenberg dimer

- 最も簡単な量子系のひとつである 反強磁性<sup>2</sup>  $S = 1/2$  Heisenberg dimer model
  - $\mathcal{H} = \vec{S}_1 \cdot \vec{S}_2$
  - ただひとつの基底状態とみつつの縮退した励起状態とがある
    - 基底状態は全スピン  $S_{\text{tot}} = 0$  で、エネルギーは  $E = -3/4$
    - 励起状態は全スピン  $S_{\text{tot}} = 1$  で、エネルギーは  $E = 1/4$
- 最初的一步としてこれを HPhi で解く

---

<sup>2</sup>以下特筆なければすべて反強磁性模型です

# 例題 1: $S = 1/2$ Heisenberg dimer

## ■ 入力ファイル SpinHalf.def

```
1 model = "SpinGC"  
2 lattice = "chain_lattice"  
3 method = "FullDiag"  
4 2S = 1  
5 L = 2  
6 J = 0.5
```

## ■ これを HPhi の Standard モードで処理するだけ

- SpinGC の GC は Grand Canonical
- $J = 0.5$  なのは周期的境界条件のためボンドが 2 本あるから

```
1 > HPhi -s SpinHalf.def
```

余裕があれば,  $S=1, S=3/2...$ などをやってみましょう ( $2S=2$ などとするだけ)

$E_{\min}=-S(S+1), E_{\max}=S^2$ になるはず

## 例題 2: $S = 1/2$ Heisenberg chain

### ■ 入力ファイル StdFace.def

```
1 model = "Spin"  
2 lattice = "chain_lattice"  
3 method = "Lanczos"  
4 2S = 1  
5 2Sz = 0  
6 J = 1.0  
7 L = 12
```

- Spin は Canonical
- 2Sz は 計算したい状態の全スピンの z 成分の 2 倍
- 今回は低エネルギー状態のみでいいので Lanczos 法を使う

```
1 > HPhi -s StdFace.def
```

## 例題 2-A: $S = 1/2$ Heisenberg chain

- $S = 1/2$  Heisenberg chain
  - $\mathcal{H} = \sum_{i=1}^L \vec{S}_i \cdot \vec{S}_{i+1}$
  - 周期境界  $S_{L+1} = S_1$  を課す (以下同様)
  - 基底状態は朝永・Luttinger 流体で、ギャップレス
  - 励起は線形分散を示す
  - 有限系では状態の取れる波数が離散化されるため、エネルギーギャップが開く
    - 分散が線形なので、 $\Delta \propto 1/L$  となる<sup>3</sup>
- 有限系のエネルギーギャップを HPhi で計算する

---

<sup>3</sup>実際には対数補正がかかる

## 例題 2: $S = 1/2$ Heisenberg chain

### ■ 入力ファイル StdFace.def

```
1 model = "Spin"  
2 lattice = "chain_lattice"  
3 method = "Lanczos"  
4 2S = 1  
5 2Sz = 0  
6 J = 1.0  
7 L = 12
```

- Spin は Canonical
- 2Sz は 計算したい状態の全スピンの z 成分の 2 倍
- 今回は低エネルギー状態のみでいいので Lanczos 法を使う

```
1 > HPhi -s StdFace.def
```

## 例題 2: $S = 1/2$ Heisenberg chain

- 今回も大量の出力とともに一瞬で計算が終わる（はず）
- 必要な情報は `output/zvo_Lanczos_Step.dat` に出力されている<sup>4</sup>

```
1 > tail -n 3 output/zvo_Lanczos_Step.dat
2 stp=42 -5.3873909174 -5.0315434037
   -4.7773893336 -4.5693744101
3 stp=44 -5.3873909174 -5.0315434037
   -4.7773893337 -4.5693744108
4 stp=46 -5.3873909174 -5.0315434037
   -4.7773893337 -4.5693744108
```

- これは各 “Lanczos step” で計算した「ちいさい固有値」が幾つか並んでいる
  - デフォルト設定では第一励起エネルギーの収束をチェックしている
- これの左 2 つの差がエネルギーギャップ

<sup>4</sup>一応ログ出力にもほぼ同じのが出ている



## 例題 2: $S = 1/2$ Heisenberg chain

### データ抽出の自動化

- 一回だけなら電卓に手入力でもいいけれど、超絶面倒くさいので自動化すべき
- `tail` と `awk` でワンライナーを書くのが楽<sup>5</sup>

```
1 > tail -n 1 output/zvo_Lanczos_Step.dat  
  | awk '{print $3-$2}'  
2 0.355848
```

- 長さ  $L$  を変えながらエネルギーギャップ  $\Delta$  を何度も計算して、ギャップの長さ依存性を見てみよう
  - ノート PC、特に仮想マシンでは  $L = 18, 20$  ぐらいで止めておくのが賢明
    - それ以上のサイズは自習時間にがんばってください
  - $L$  を変えて何度も計算するという操作も自動化を試みてみよう
    - → 次のページ

<sup>5</sup>自分の手に馴染んでいるなら何使ってもいいです

# 例題 2: $S = 1/2$ Heisenberg chain

## 入力の自動化

- 今回はひとつのパラメータを変えるだけなのでシェルで事足りる
- 共通となるパラメータをくくりだして保存しておく (StdFace.common)

```
1 model = "Spin"
2 lattice = "chain_lattice"
3 method = "Lanczos"
4 2S = 1
5 2Sz = 0
6 J = 1.0
```

- 別々の  $L$  毎にこの共通ファイルをコピーして、足りないパラメータを追記すればよい

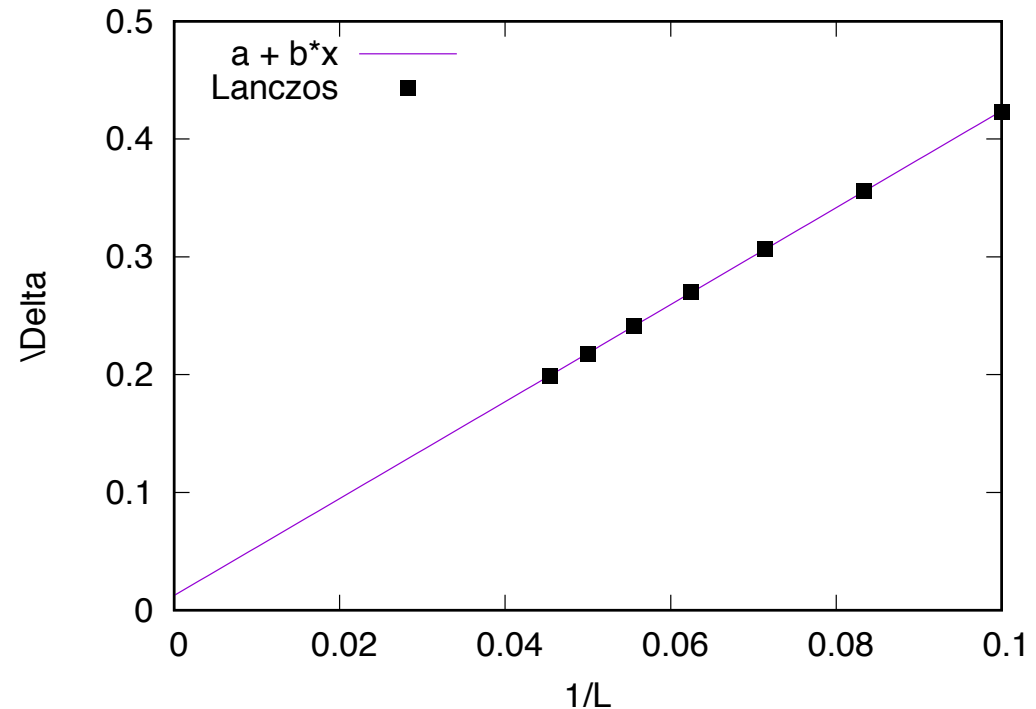
```
1 rm -f res.dat
2 for L in 10 12 14 16; do
3   cp StdFace.common StdFace.def
4   echo "L=$L" >> StdFace.def
5   HPhi -s StdFace.def
6   gap=$((tail -n1 \
7     output/zvo_Lanczos_Step.dat | \
8     awk '{print $3-$2}'))
9   echo $L $gap >> res.dat
10 done
```

- ヒアドキュメントを使うのもアリ。自分の好きなようにやりましょう

# 例題 2: $S = 1/2$ Heisenberg chain

結果

- $S = 1/2$  Heisenberg chain はギャップの閉じるぎりぎりのところであり、 $\log L$  の対数補正が現れるので結構数値解析が難しい



- $S = 1/2$  XY chain ( $J_z = 0, J_x = 1.0$ ) ではもっと綺麗にかける
  - → 練習問題
- $S=1$  Heisenberg chain にするとギャップが見えるはず (Haldane gap)  
→ 練習問題 [input file で  $2S=2$  とするだけ]

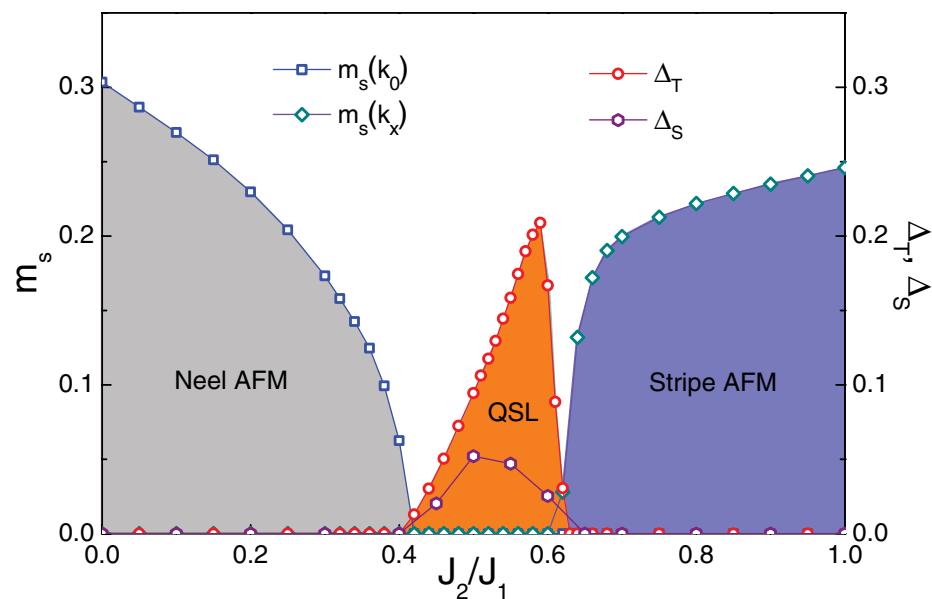
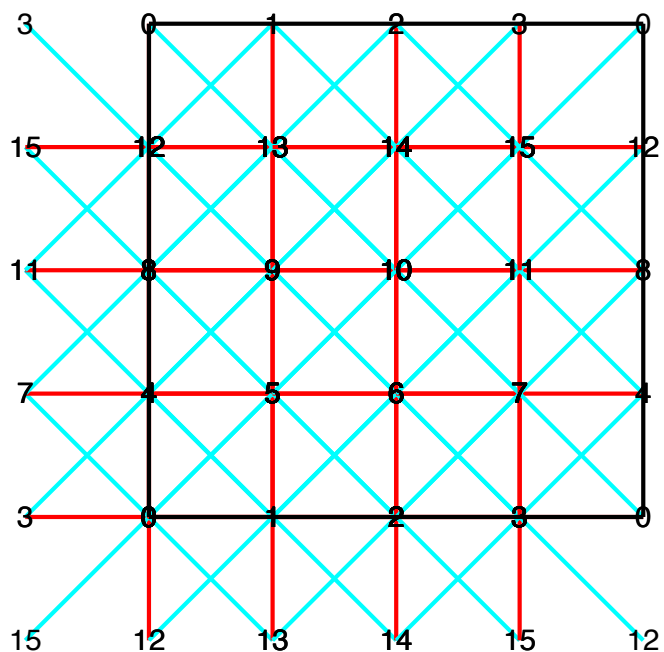


Basic exercise 2 !

# 例題3: J1-J2ハイゼンベルク模型

$$H = J_1 \sum_{\langle i,j \rangle} S_i S_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} S_i S_j$$

最近接 J1, 次最近接 J2



PRB 86, 024424(2012)

lattice.gpで描画可能

J2/J1~0.5で非磁性の  
基底状態(スピン液体?)

# 例題3: J1-J2ハイゼンベルク模型

$$H = J_1 \sum_{\langle i,j \rangle} S_i S_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} S_i S_j$$

1. Lanczosでエネルギーを計算 (サイズ4×4位)
2. TPQで比熱を計算(J2/J1~0.5でどうなるか?)
3. (発展)余裕があればスピン相関も計算してみましよう

```
L = 4  
W = 4  
model = "Spin"  
method = "Lanczos"  
lattice = "square lattice"  
J = 2.0  
J' = 1.0  
2Sz = 0
```

# 例題3: 基底状態の答え

$J_1$ - $J_2$  Heisenberg model,  $N_s=4\times 4$ ,  $J_1=2.0$

E. Dagotto and A. Moreo, PRB (R) 39 , 4744 (1989)

TABLE I. Ground-state energy ( $E_0$ ) and first excited-state energy ( $E_1$ ) per site (both singlets with zero momentum) of the 2D Heisenberg model with frustration as a function of  $J_2$  on a  $4\times 4$  lattice. The error is in the last digit.

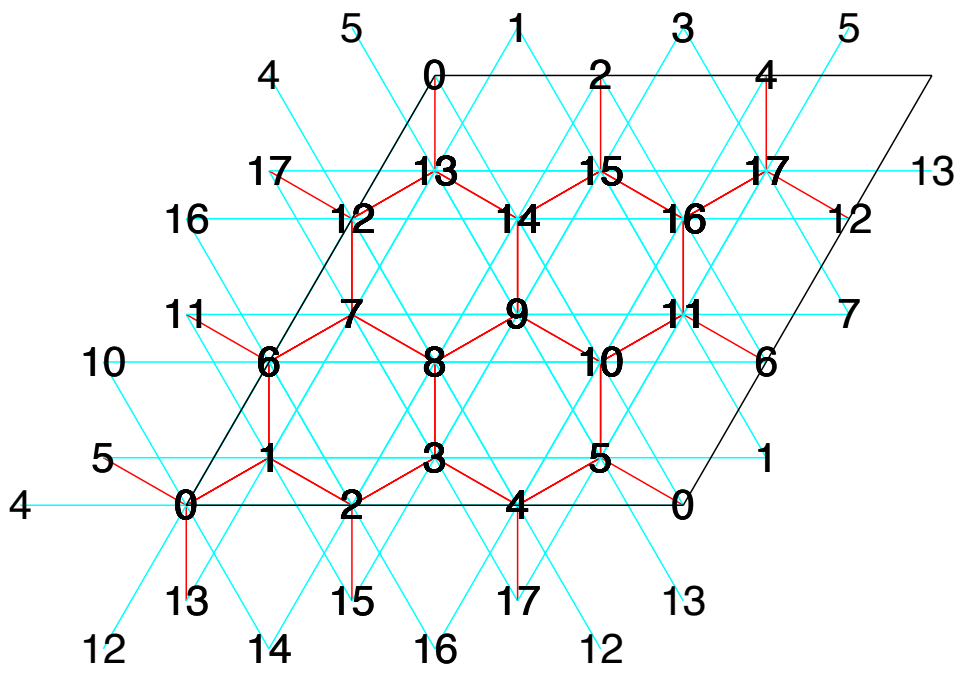
$J_2$	$E_0$	$E_1$
0.950	-1.065978	-1.0160
1.100	-1.047189	-1.0254
1.150	-1.047183	-1.0307
1.200	-1.051792	-1.0380
1.325	-1.089305	-1.0804
1.400	-1.127716	-1.1169
1.500	-1.188546	-1.1691
1.600	-1.254670	-1.2233
1.750	-1.358437	-1.3072

今ならPCで数秒で計算できる。

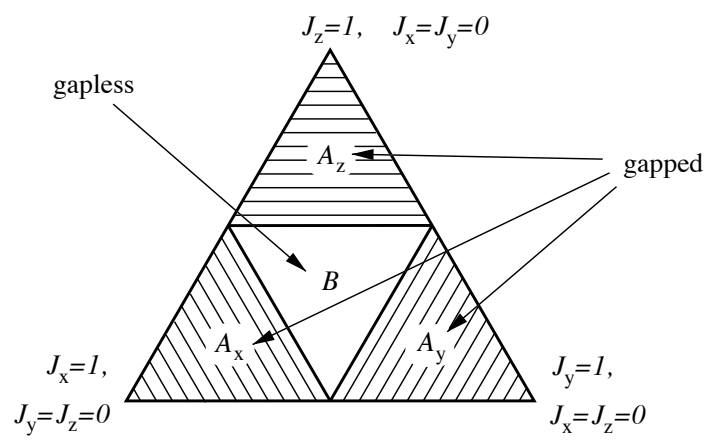
# 例題4: Kitaev model

$$H = -J_x \sum_{x\text{-bond}} S_i^x S_j^x - J_y \sum_{y\text{-bond}} S_i^y S_j^y - J_z \sum_{z\text{-bond}} S_i^z S_j^z$$

3方向のそれぞれが  
Jx, Jy, Jzで相互作用



## 相図



Annals of Physics 321, 2-111 (2016)

lattice.jpで描画可能

可解模型→スピン液体



# 例題4: Kitaev model

$$H = -J_x \sum_{x\text{-bond}} S_i^x S_j^x - J_y \sum_{y\text{-bond}} S_i^y S_j^y - J_z \sum_{z\text{-bond}} S_i^z S_j^z$$

1. Lanczosでエネルギーを計算 (サイズ18サイト位)
2. TPQで比熱を計算: マヨラナ粒子の兆候がみえるか?
3. (発展)次近接のスピン相関が厳密に0を確認
4. (発展)ハイゼンベルク項をたすとどうなるか?
5. (発展)磁場をかけて磁化の温度依存性から帯磁率が計算可能

```
W = 3
L = 3
model = "SpinGC"
method = "Lanczos"
lattice = "Honeycomb"
J0x = -1.0
J0y = 0.0
J0z = 0.0
J1x = 0.0
J1y = -1.0
J1z = 0.0
J2x = 0.0
J2y = 0.0
J2z = -1.0
```

# 例題5: Hubbard chain

$$H = -t \sum_{\langle i,j \rangle} (c_{i\sigma}^\dagger c_{j\sigma} + \text{h.c.}) + U \sum_i n_{i\uparrow} n_{i\downarrow}$$

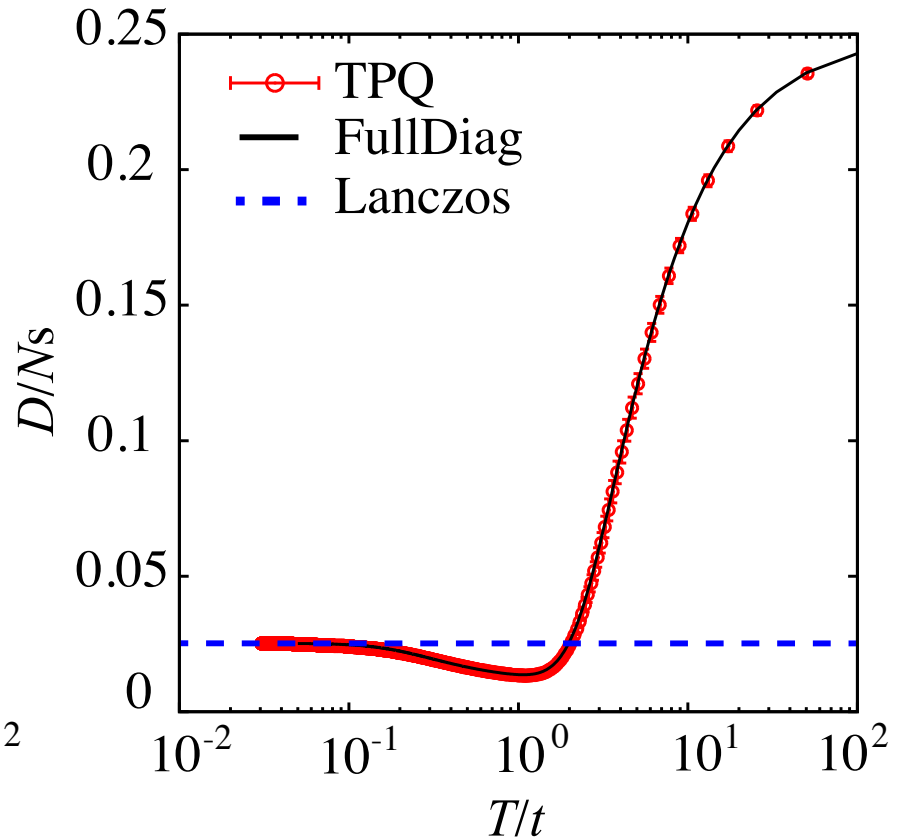
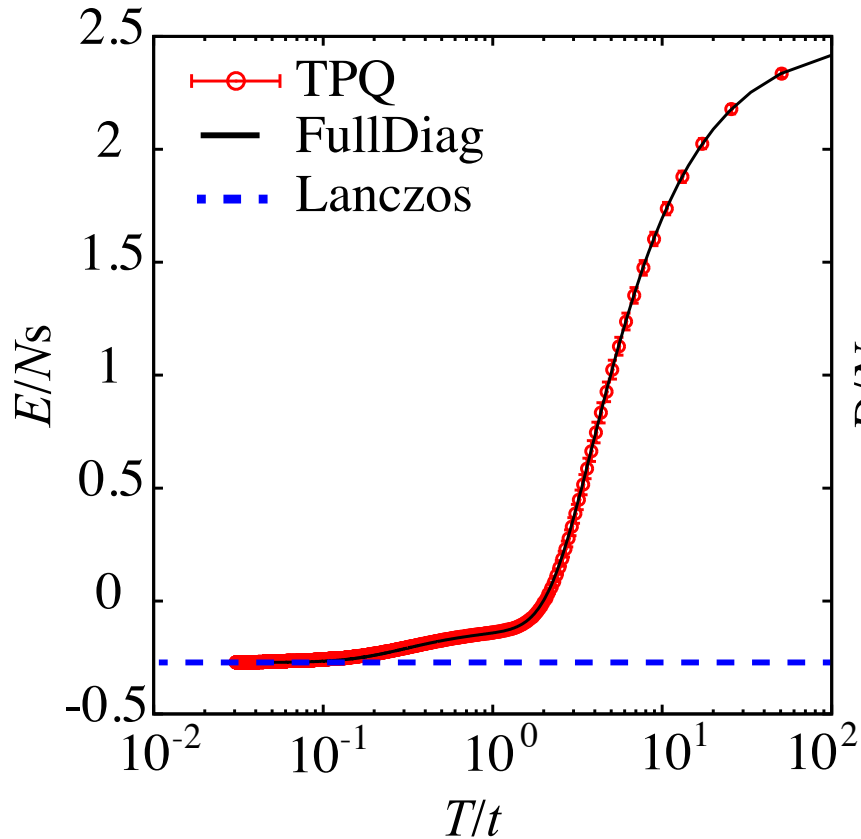
1. Lanczosでエネルギー・二重占有度を計算 (サイズ8サイト位)
2. TPQで比熱・二重占有度を計算:
3. (発展) 全対角化でアンサンブル平均を計算してTPQと比較

```
L = 8
model = "FermionHubbard"
method = "Lanczos"
lattice = "chain"
t = 1.0
U = 8.0
nelec = 8
2Sz = 0
```

# 例題5: Hubbard chain (有限温度計算)

FullDiag, TPQ, Lanczosの比較

Hubbard model,  $L=8$ ,  $U/t=8$ , half filling,  $S_z=0$



3つの手法は互いに一致

# その他の例

HPhi/samples/Standard/

にHubbard模型、Heisenberg模型、Kitaev模型、近藤格子模型のStandard modeのStdFace.defがあります。

StdFace.defを適宜変えて遊んでみてください。

注意：

-サイズは大きくし過ぎないこと

(PCだとspin 1/2で24site, Hubbardで12サイトくらいが限界)

-Lanczosはサイズが小さいと不安定です

(数千次元以上を推奨)

-TPQはサイズが小さいとうまくいきません

(数千次元以上を推奨)

# 物性研スパコンを使ってみよう!

## 物性研システムB (sekirei)

✓ fat node: 1node (40 cores) 1TBのメモリ、  
2nodesまで使用可 → ~2TB

✓ cpu node: 1node (24cores) 120GBのメモリ、  
144nodesまで使用可 → ~17TB

潤沢なメモリ: spin 1/2 36 sites, Hubbard 18 sitesまでなら  
比較的簡単に計算可能 (5-10年程度前の最先端の計算)

TPQ法を使えば有限温度計算も可能!

お試しクラス(A) : 随時受付(100ポイント以下)

通常クラス(B,C,E): 年2回(6月,12月)受付

緊急クラス(D) : 随時受け付け,ただ緊急性が必要

# 物性研スパコンを使ってみよう!

物性研システムB (sekirei)にはインストール済み  
以下のようなスクリプトをサブミットするだけで、  
大規模並列計算が可能

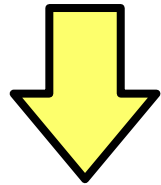
```
#!/bin/sh
#QSUB -queue F144cpu
#QSUB -node 128
#QSUB -mpi 128
#QSUB -omp 24
#QSUB -place pack
#QSUB -over false
#PBS -l walltime=24:00:00
#PBS -N HPhi
cd ${PBS_O_WORKDIR}
source /home/issp/materiapps/HPhi/HPhivars.sh
mpijob HPhi -s StdFace.def
```



Expert mode !

# How to use HΦ: What is Expert mode ?

HPhi -s StdFace.def



Standard mode:

計算に必要なファイルを**自動生成**

模型を指定するパラメーターファイル (3個)

zInterAll.def, zTrans.def, zlocspn.def

計算条件を指定するパラメーターファイル (2個)

modpara.def, calcmod.def

相関関数を指定するパラメーターファイル (2個)

greenone.def, greentwo.def

+ファイル名を列挙したファイル: namelist.def

**Expert mode:** 計算に必要なファイルを**自分で用意**

→ Standard modeで生成したものを書き換えるのが楽



# How to use HΦ: What is Expert mode ?

**Expert mode:** 計算に必要な以下のファイルを自分で用意  
→ Standard modeで生成したものを書き換えるのが楽

モデルを指定するパラメーターファイル (3個)

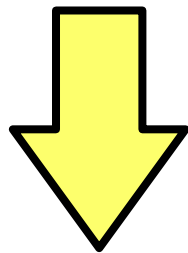
zInterAll.def, zTrans.def, zlocspn.def

計算条件を指定するパラメーターファイル (2個)

modpara.def, calcmod.def

相関関数を指定するパラメーターファイル (2個)

greenone.def, greentwo.def



用意して以下のコマンドを実行

```
HPhi -e namelist.def
```

# How to use HΦ: zInterall.def

## 模型を指定するパラメーターファイルの例

$$H+ = \sum_{i,j,k,l} \sum_{\sigma_1,\sigma_2,\sigma_3,\sigma_4} I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}$$

```
=====
NInterAll
```

```
96
```

相互作用の総数

```
=====zInterAll=====
```

実部

虚部

0	0	0	0	1	0	1	0	0.500000	0.000000
0	0	0	0	1	1	1	1	-0.500000	0.000000
0	1	0	1	1	0	1	0	-0.500000	0.000000
0	1	0	1	1	1	1	1	0.500000	0.000000
0	0	0	1	1	1	1	0	1.000000	0.000000
0	1	0	0	1	0	1	1	1.000000	0.000000

```
...
```

```
i   σ1  j   σ2  k   σ3  l   σ4
```

「任意」の2体の相互作用を指定することができる  
(自動生成するようにしておくとも便利)  
→任意の格子を指定することが出来る

# How to use HΦ: Expert mode

## 模型を指定する簡易版のパラメータファイル

**- CoulombIntra**  $H+ = \sum_i U_i n_{i\uparrow} n_{i\downarrow}$

```
=====  
NCoulombintra 2  
=====  
=====  
=====Exchange=====
```

0	4.0
1	4.0

**-Exchange**  $H+ = \sum_{i,j} J_{ij}^{\text{Ex}} (S_i^+ S_j^- + S_i^- S_j^+)$

```
=====  
NExchange 2  
=====  
=====  
=====Exchange=====
```

0	1	0.5
1	2	0.5

指定が楽&高速化することも  
詳細はマニュアル参照

# HΦを使いこなすには..

- 簡単なファイルの書き出し、読み込み&解析ができると計算後の解析の幅が格段に広がります
- もちろん、通常のプログラミング言語(fortran, C++/Cなど)でも十分できますが、  
スクリプト言語(python, perl, rubyなど)を覚えておくと便利です。sed、awkといったコマンドを使うのも楽ですが、拡張性が低いのが難点です
- 自分で気に入ったものを使うのが一番ですが、  
周りに詳しい人がいる言語を選択するのが無難でしょうか..
- 計算物質科学の分野だと主流はpythonでしょうか..

あとはやりたいこと  
をやってください

- 近藤格子模型
- カゴメ格子, 三角格子...

解析したい模型などの  
相談・質問も大歓迎です!

