$\mathcal{H}\Phi$2.0の新機能

局所最適ブロック共役勾配法による複数固有状態計算
相関関数のフーリエ変換ユーティリティー
シフト双共役勾配法による励起スペクトル計算

物性研究所 物質設計評価施設
大型計算機室ソフトウェア高度化推進チーム
河村光晶

2017/6/29
$\mathcal{H}\Phi$講習会

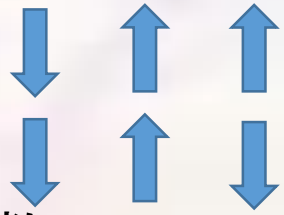
目次

- 新機能1:複数固有状態計算
  - 背景
  - 局所最適ブロック共役勾配(LOBCG)法
  - 計算例
- 新機能2:静的相関関数のフーリエ変換
- 新機能3:励起スペクトル計算
  - 背景
  - シフト双共役勾配(Shifted BiCG)法
  - 計算例
- まとめ

新機能1:複数固有値計算

背景

$\mathcal{H}\Phi$のターゲット　　格子多体系



2^N次元のヒルベルト空間(Nはサイト数)

直接法(全対角化)	O[$(2^N)^2$]のメモリ O[$(2^N)^3$]の計算時間

反復法 (最大・最小・絶対値最大、その他)	O(2^N)のメモリ O(N × 2^N × 反復回数)の計算時間

例/べき乗法

$\boldsymbol{x} := \hat{H}\boldsymbol{x}$ ⇄ $\boldsymbol{x} := \boldsymbol{x}/|\boldsymbol{x}|$

固有値の絶対値の大きい状態に収束していく

これまでの$\mathcal{H}\Phi$

べき乗法 → Lanczos法

最大固有値と最小固有値を計算する

Krylov部分空間:$\mathrm{span}\left(\left|\Phi^{(0)}\right\rangle, \hat{H}\left|\Phi^{(0)}\right\rangle, \hat{H}^2\left|\Phi^{(0)}\right\rangle, \cdots, \hat{H}^M\left|\Phi^{(0)}\right\rangle\right)$の部分空間ハミルトニアン

各反復でハミルトニアンをかける時に直交化をする

それ以前のベクトルを持っておく必要なし(部分空間ハミルトニアンンが3重対角)

$$H_{\mathrm{sub}} = \begin{pmatrix} & & 0 \\ & \ddots & \\ 0 & & \end{pmatrix}$$

これを対角化して固有値を求める

最大・最小近辺の固有値も計算できるが・・・

固有ベクトルを計算するときには・・・

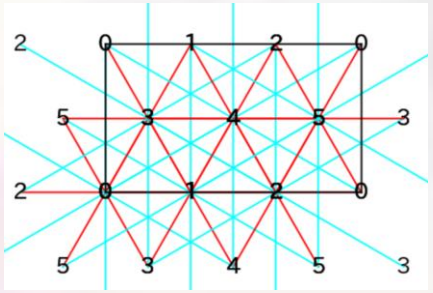
Lanczos法(固有値) → Lanczos法(固有ベクトル) → 逆反復法(固有ベクトルの高精度化)

これまでの$\mathcal{H}\Phi$の問題点

例/三角格子ハバード模型(6サイト)

```
model = "Hubbard“
lattice = "triangular"
a0w = 3
a0l = 0
a1w = -1
a1l = 2
method = "Lanczos"
t = 1.0
U = 4.0
nelec = 6
2Sz = 0
```

```
$ gnuplot lattice.gp
```



```
method = “FullDiag"
```

```
i=     0 Energy=-10.848286
i=     1 Energy= -8.315735
i=     2 Energy= -8.315735
i=     3 Energy= -8.237733
i=     4 Energy= -7.728182
i=     5 Energy= -7.703537
i=     6 Energy= -7.703537
i=     7 Energy= -7.554153
i=     8 Energy= -7.554153
略
```

```
 LanczosStep  E[1] E[2] E[3] E[4] Target:E[3] E_Max/Nsite
  stp = 2 -0.3548292899 12.4881600985 xxxxxxxxxx xxxxxxxxx  0.0000000000 xxxxxxxxx
  stp = 4 -6.0683873495 1.8080229826 10.2313320721 18.5822029997 10.2313320721 3.0970338333
  stp = 6 -8.1000584754 -3.5362664145 2.5187038512 9.0952060082 2.5187038512 3.3988254807
略
  stp = 64 -10.8482855887 -10.7899137901 -8.3157354869 -8.2377334064 -8.3157354869 3.8136933537
  stp = 66 -10.8482855887 -10.8465909598 -8.3157354869 -8.2377334064 -8.3157354869 3.8136933537
```

数値誤差で直交性が崩れて偽の縮退が出来る

縮退している状態数までは分からない

局所最適ブロック共役勾配法

Locally Optimal Block Conjugate Gradient (LOBCG) method

A. V. Knyazev, SIAM J. Sci. Compute. 23, 517 (2001).

山田進 他, 日本計算工学会論文集, 20060027 (2006).

M本ベクトルを求めるとすると、各ステップごとに

固有ベクトル$|\Phi_1\rangle, |\Phi_2\rangle, \cdots |\Phi_M\rangle$

残差ベクトル $\{|r_i\rangle = H|\Phi_i\rangle - \varepsilon_i|\Phi_i\rangle\}$

共役勾配ベクトル $|p_1\rangle, |p_2\rangle, \cdots |p_M\rangle$

で部分空間ハミルトニアン(3M次元)を作って
それを対角化

エネルギーの低い方からM個のベクトルを次のステップの近似固有ベクトルとする。

全ての残差ベクトルのノルムがしきい値以下になるまで繰り返す。

計算時間・メモリは求める固有ベクトルの数に比例する

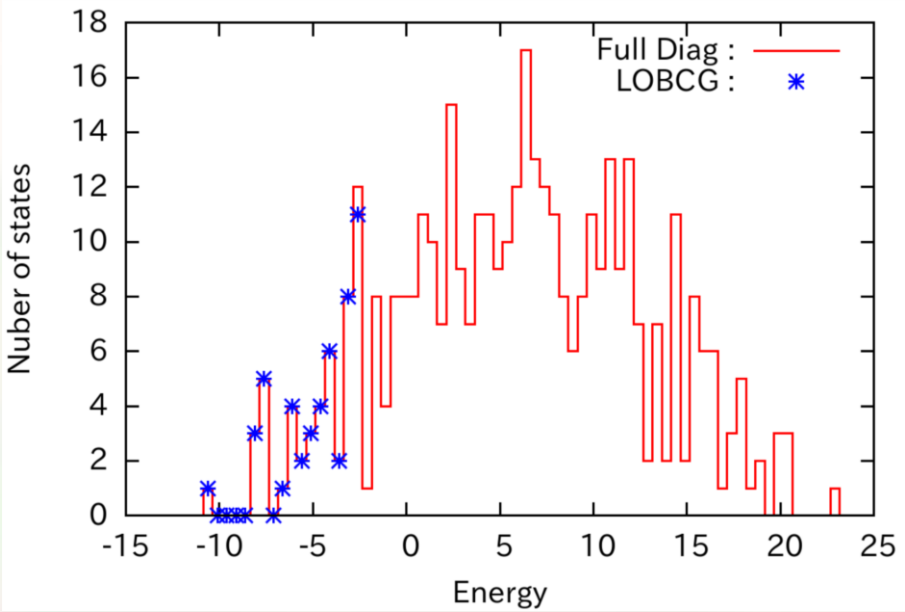
最小固有値だけ計算することも可能

計算例

例/三角格子ハバード模型(6サイト)

```
model = "Hubbard“
lattice = "triangular"
a0w = 3
a0l = 0
a1w = -1
a1l = 2
method = “CG"
t = 1.0
U = 4.0
nelec = 6
2Sz = 0
LanczosEPS = 8
Exct = 5
```

```
Exct = 50
```




```
Step Residual-2-norm   Threshold      Energy
  1   7.04783e+00     6.33060e-04   5.51419e+00   6.33060e+00   5.96198e+00   5.91659e+00   5.84129e+00
  2   6.81182e+00     3.19811e-04   1.75785e+00   2.62971e+00   2.86870e+00   2.99176e+00   3.19811e+00
  3   4.82958e+00     3.85028e-04  -3.85028e+00  -2.95489e+00  -2.01928e+00  -1.62988e+00  -1.21946e+00
  4   3.82792e+00     6.84264e-04  -6.84264e+00  -5.73852e+00  -4.84241e+00  -4.29262e+00  -3.96856e+00
略
110   1.44101e-03     1.08483e-03  -1.08483e+01  -8.31574e+00  -8.31574e+00  -8.23773e+00  -7.72818e+00
111   1.24017e-03     1.08483e-03  -1.08483e+01  -8.31574e+00  -8.31574e+00  -8.23773e+00  -7.72818e+00
112   1.02433e-03     1.08483e-03  -1.08483e+01  -8.31574e+00  -8.31574e+00  -8.23773e+00  -7.72818e+00
```

計算時間・メモリは求める固有ベクトルの数に比例する

ベクトルの書き出し&再スタート

スパコンの時間制限等

```
a0w = 3
a0l = 0
a1w = -1
a1l = 2
model = "Hubbard"
method = "CG“
lattice = "triangular"
t = 1.0
U = 4.0
nelec = 6
2Sz = 0
LanczosEPS = 8
Lanczos_max = 10
restart = "RestartSave"
```

近似固有ベクトルを書き出してプログラム終了。

```
 Start:  Input vector.
FileOpenError: ./output/tmpvec_set0_rank_0.dat.
A file of Inputvector does not exist.
Start from scratch.
  initial_mode=1 (random): iv = -1 i_max=400 k_exct =1

  Step    Residual-2-norm      Threshold      Energy
     1      6.39768e+00      5.51419e-04      5.51419e+00
略
     9      2.10106e+00      9.94784e-04     -9.94784e+00
    10      1.40169e+00      1.04545e-03     -1.04545e+01

######  End  : Calculate Lanczos EigenValue.  ######

  Start:  Output vector.
  End  :  Output vector.
  LOBPCG is not converged in this process.
```

※ファイルサイズと書き出しにかかる時間に注意！

読み込んで次の計算のイニシャルゲスとして使う(この入力ファイルをそのまま使う)。
この場合は2回リスタートすると収束する。

※共役勾配ベクトル(履歴に依存)の情報がリセットされるため、
中断しない時よりトータルの反復回数はやや増える。

残差の減少の加速ーLOBPCG法ー

解の収束≒残差のノルム$|\langle r|r\rangle|^2$がある程度小さくなる。$|r\rangle = \left(\hat{H} - \varepsilon\right)|\Phi\rangle$

残差をなるべく速く減少させたい。

Locally Optimal Block Preconditioned Conjugate Gradient method

A. V. Knyazev, SIAM J. Sci. Compute. 23, 517 (2001).

山田進 他, 日本計算工学会論文集, 20060027 (2006).

前処理

相似変換により、反復法が収束しやすい行列に変えてから計算し、逆変換して戻す。

効果的な前処理の方法は行列の性質(由来)に依存する。

もっとも単純な前処理、対角スケーリング(Point Jacobi)を使ってみる。

src/CalcByLOBPCG.c 330行目

```
int do_precon = 0;//If = 1, use preconditioning (experimental)
```

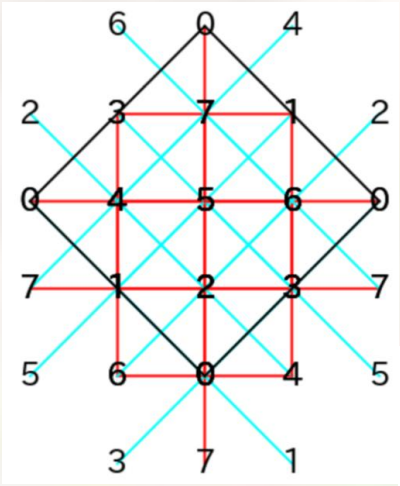
```
int do_precon = 1;//If = 1, use preconditioning (experimental)
```

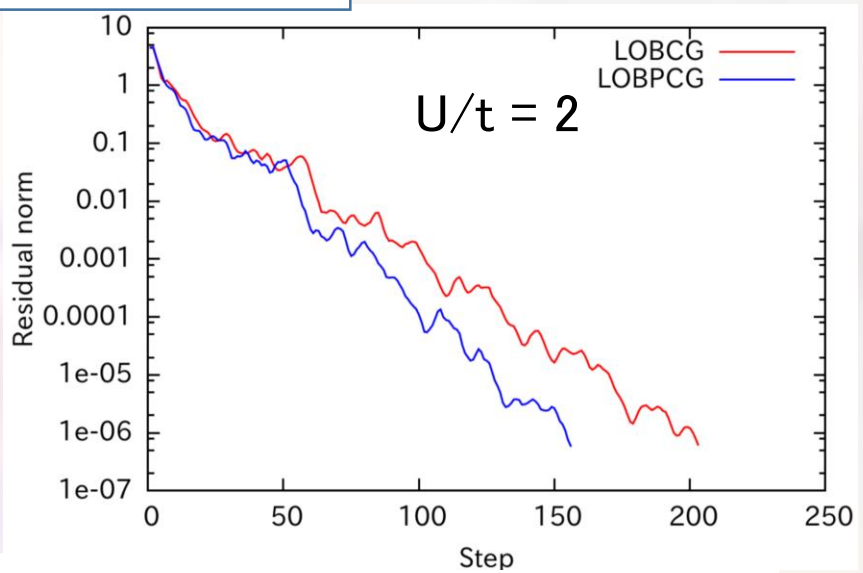
計算結果

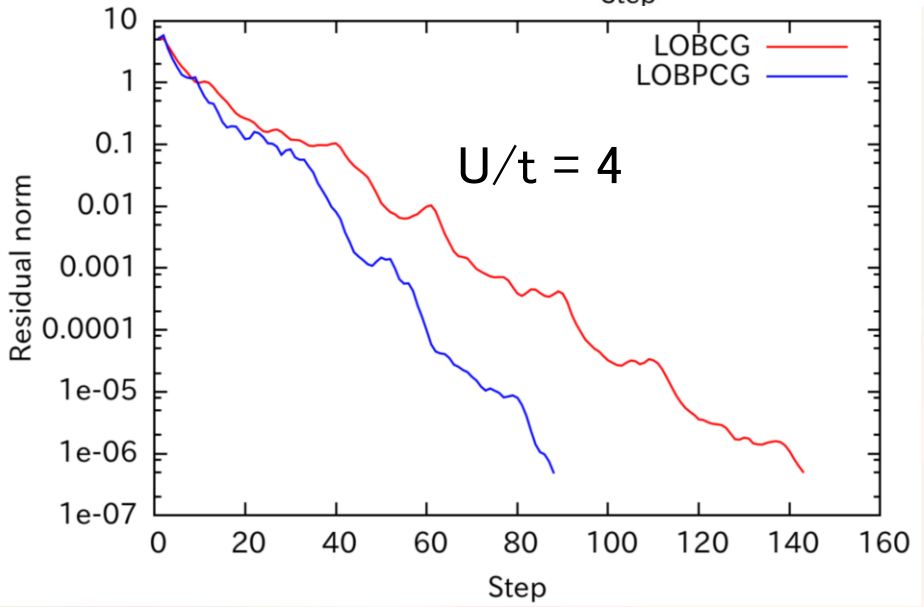
Output/zvo_Lanczos_Step.dat

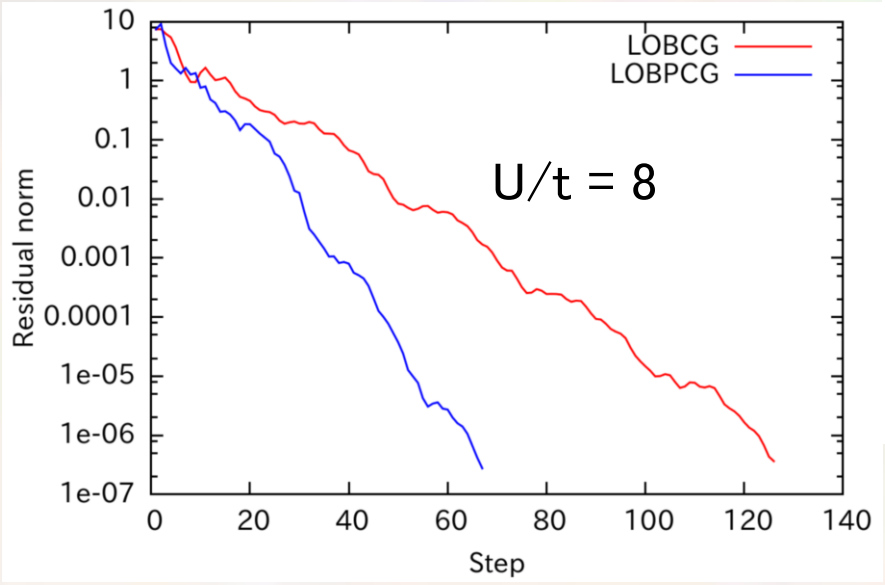
例/正方格子ハバード模型(8サイト)

```
a0w = 2
a0l = 2
a1w = -2
a1l = 2
model = "Hubbard"
method = "CG“
lattice = “square"
t = 1.0
U =
nelec = 8
2Sz = 0
```





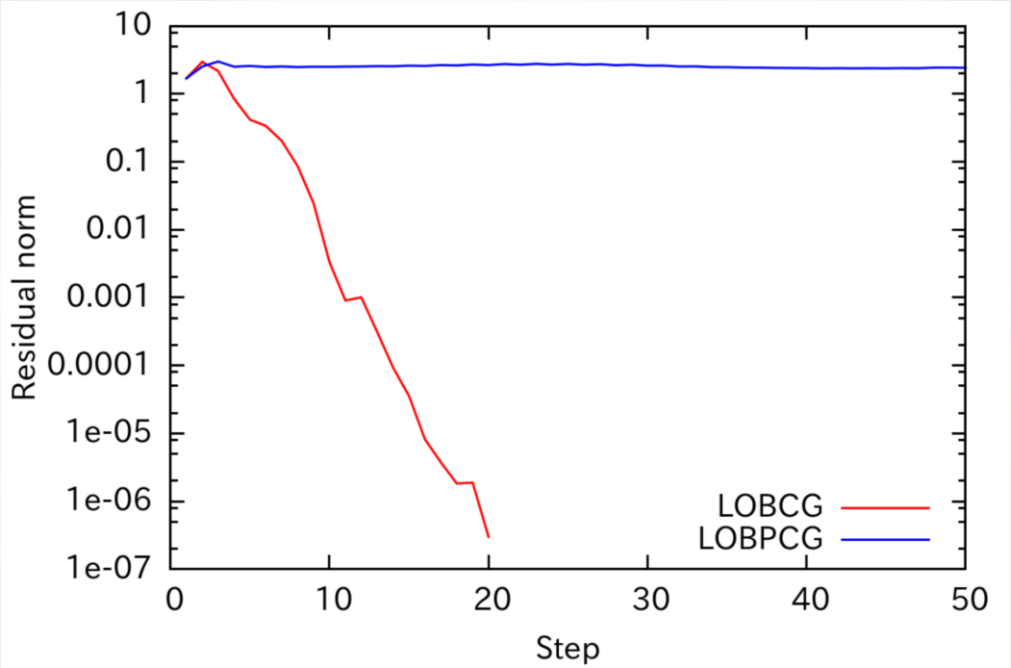




ハイゼンベルグ模型 (U/t = ∞)はどうか？

ハイゼンベルグ模型

```
a0w = 2
a0l = 2
a1w = -2
a1l = 2
model = “Spin"
method = "CG“
lattice = “square"
J = 1.0
2Sz = 0
```



効果的な前処理の方法は行列の性質(由来)に依存する。

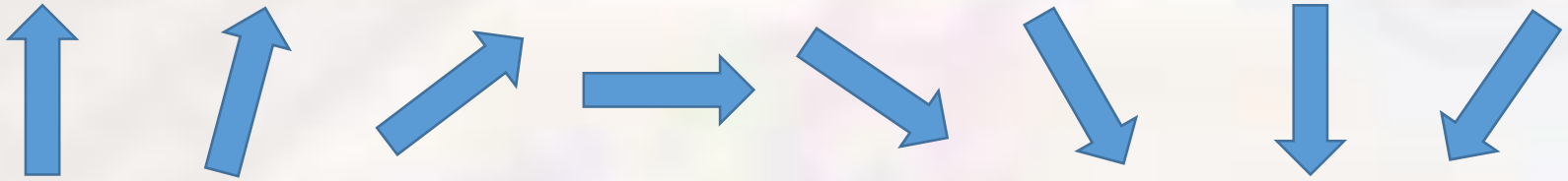
対角スケーリング: 対角項が大きいときに有効

色々試してみると、ハバードモデルだけやる分には良さそう。

スピン系では収束しなくなることがある。もしくは、速くならない。

新機能2:静的相関関数のフーリエ変換

長距離(と言っても厳密対角化レベル)の相関を調べる。



$$\langle \hat{A}_k^\dagger A_k \rangle = \frac{1}{N_{\mathrm{Cell}}} \sum_{ij}^{N_{\mathrm{site}}} e^{i\mathbf{k}\cdot(\mathbf{R}_i - \mathbf{R}_j)} \langle \hat{A}_i^\dagger A_j \rangle$$

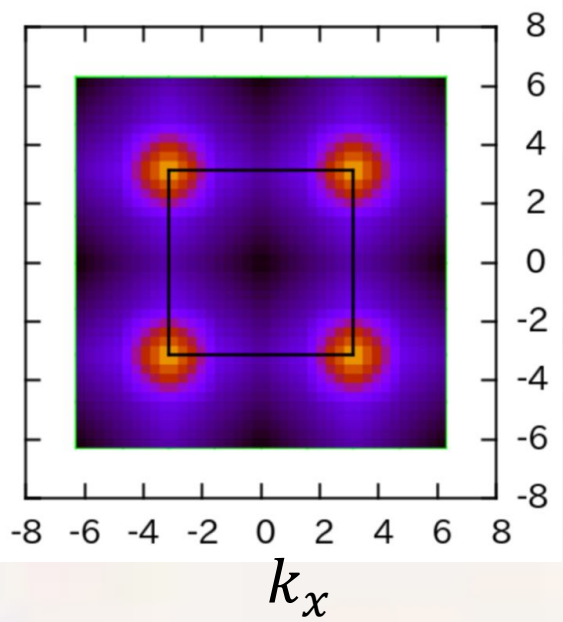
ユーティリティ・プログラムとドキュメントはHΦ本体と別にある。

tool/fourier : フーリエ変換をするプログラム
tool/corplot : 3次元プロットをするプログラム
doc/userguid.html : からマニュアルを閲覧できる

計算結果

例/正方格子ハイゼンベルグ模型(16サイト)
samples/Standard/Spin/HeisenbergSquare/

```
$ パス/HPhi –s StdFace.def
$ パス/fourier namelist.def geometry.dat
$ パス/corplot output/zvo_corr.dat
```




```
#####  Plot Start  #####

  Please specify target number from below (0 or Ctrl-C to exit):

  Real Part Without ErrorBar
    [ 1] Up-Up [ 2] Down-Down [ 3] Density-Density [ 4] SzSz [ 5] S+S- [ 6] S.S
  Imaginary Part Without ErrorBar
    [11] Up-Up [12] Down-Down [13] Density-Density [14] SzSz [15] S+S- [16] S.S
  Real Part With ErrorBar
    [21] Up-Up [22] Down-Down [23] Density-Density [24] SzSz [25] S+S- [26] S.S
  Imaginary Part With ErrorBar
    [31] Up-Up [32] Down-Down [33] Density-Density [34] SzSz [35] S+S- [36] S.S

 Target : 6 （と打ってEnter）
```

Lanczos,LOBCG,TPQ,FullDaig(,mVMC)それぞれで使える

既知の問題点
corplot内でgnuplotを呼び出しているが、
4.4より前のバージョンのgnuplotでは描画できない。

制限
S>1/2のスピンには対応していない。

新機能2:励起スペクトル計算

背景　中性子散乱、ARPES、その他応答

$$G_{ij}(\omega) = \langle \Phi_0 \left| \hat{A}_i^{\dagger} \left(\omega - \hat{H}\right)^{-1} \hat{A}_j \right| \Phi_0 \rangle$$　HPhiでは i=j

シフト(双)共役勾配法

Shifted Bi-Conjugate Gradient (BiCG) method

A. Frommer, Computing 70, 87 (2003).

S. Yamamoto, *et al.*, JPSJ 77, 114713 (2008).

$|b\rangle = \hat{A}_j |\Phi_0\rangle$

$$\left(\omega_n - \hat{H}\right)|x_n\rangle = |b\rangle$$

$$G_{ii}(\omega_n) = \langle b | x_n \rangle$$

Krylov部分空間: $\mathrm{span}\left(|b\rangle, \left(\omega_n - \hat{H}\right)|b\rangle, \left(\omega_n - \hat{H}\right)^2 |b\rangle, \cdots, \left(\omega_n - \hat{H}\right)^M |b\rangle\right)$は$\omega_n$によらず共通化できる。

残差$|r_n^{(l)}\rangle = \left(\omega_n - \hat{H}\right)\left|x_n^{(l)}\right\rangle - |b\rangle$を全ての$\omega_n$について平行にするアルゴリズム。

ω_nの数をどれだけ増やしても1個の振動数の計算と所要時間はほぼ同じ。

シフト(双)共役法ライブラリ:Kω

$$(\omega_n - \hat{H})|x_n\rangle = |b\rangle$$

典型的にはω_nは複素数

共役勾配(CG)法(エルミート行列)

双共役勾配(BiCG)法（非エルミート行列）

この計算をするルーチンをライブラリとして公開している。

$\hat{H}$をかける部分はユーザー(ライブラリを使う人)が自分で作る(リバース・コミュニケーション・インターフェース)。

このライブラリをHΦで使っている。

HΦ以外の公開プログラム・論文等での利用は今のところ確認できていない

計算例(入力ファイル)

例/正方格子ハバード模型(8サイト)

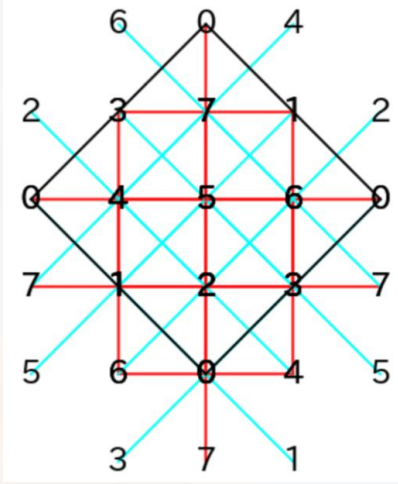
HPhiを2回実行する必要がある。

```
a0W = 2
a0L = 2
a1W = -2
a1L = 2
model = "hubbard"
method = "CG"
lattice = "square"
t = 1.0
t' = 0.5
U = 4.0
2Sz = 0
nelec = 8
LanczosEPS = 8
EigenvecIO = "out"
CalcSpec = "None“
SpectrumType = "SzSz"
SpectrumQW = 0.5
SpectrumQL = 0.5
OmegaMin = -10.0
OmegaMax = 20.0
OmegaIM = 0.2
```

1回目:基底状態を求める

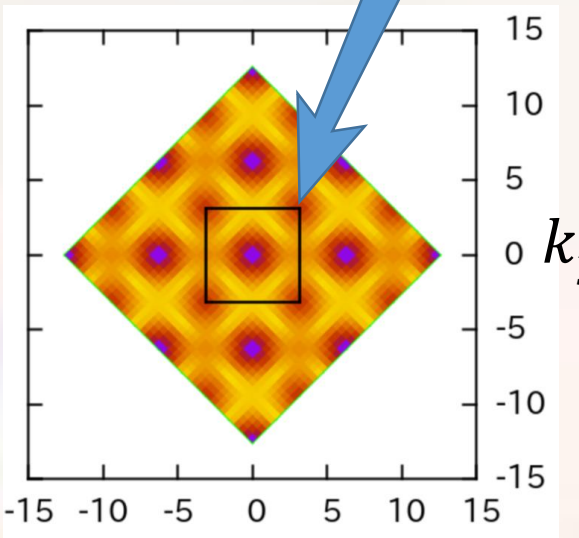
```
a0W = 2
a0L = 2
a1W = -2
a1L = 2
model = "hubbard"
method = "CG"
lattice = "square"
t = 1.0
t' = 0.5
U = 4.0
2Sz = 0
nelec = 8
LanczosEPS = 8
EigenvecIO = "out"
CalcSpec = "Normal"
SpectrumType = "SzSz"
SpectrumQW = 0.5
SpectrumQL = 0.5
OmegaMin = -10.0
OmegaMax = 20.0
OmegaIM = 0.2
```

2回目:スペクトルを求める



$\langle \hat{S}^{\dagger}_{qz} S_{qz} \rangle$

q = M点



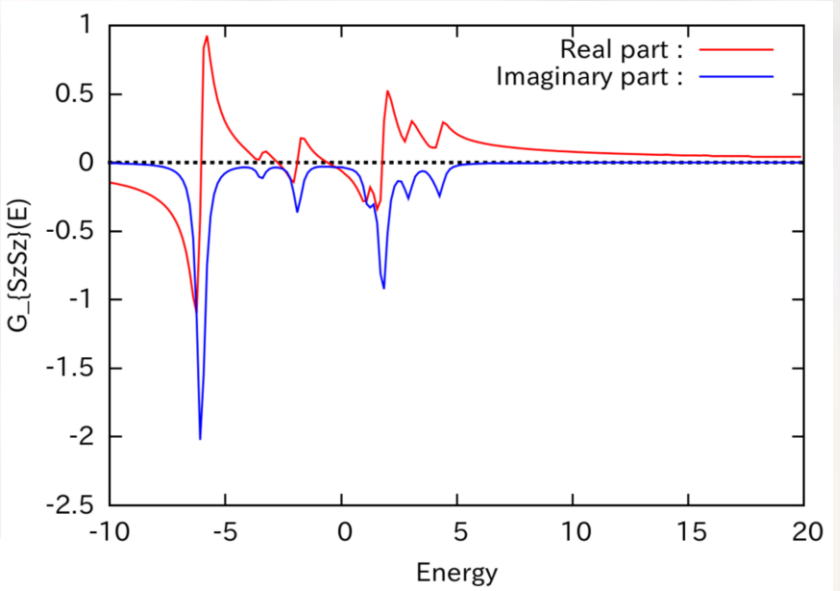
計算結果

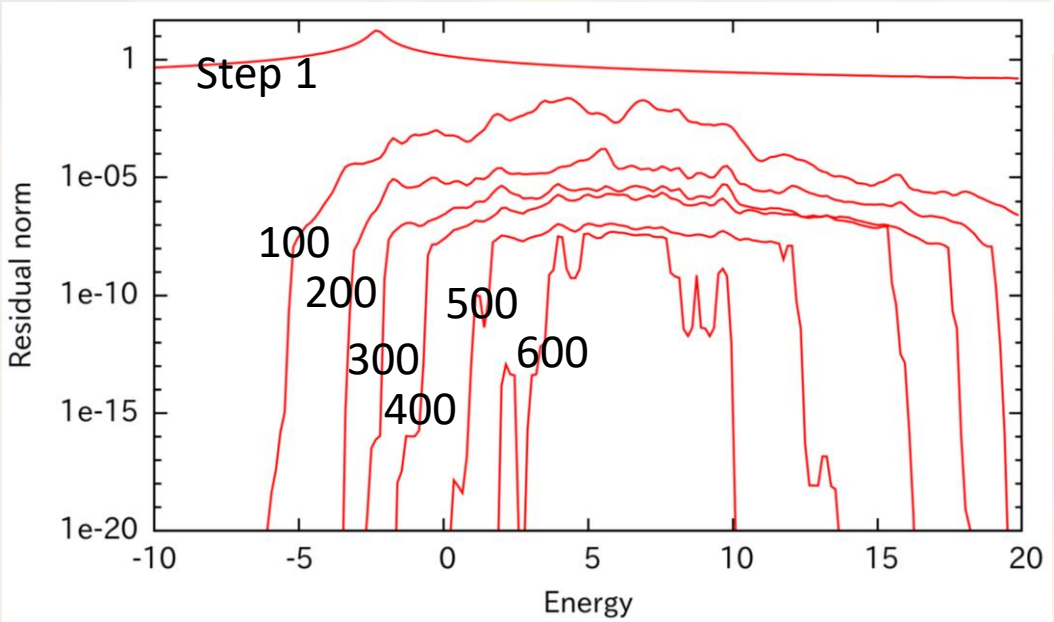
```
#####  Spectrum calculation with BiCG  #####

 Start: Calculate tridiagonal matrix components.

 Iteration   Status   Seed    Residual-2-Norm
       1        0       52    1.785310263663404e+01
       2        0       29    4.341492751424147e+00
       3        0       68    5.104459320308541e+00
略
     627        0      107    5.590599345090712e-08
     628        0      107    1.731462499130794e-08
     629        0      107    8.950365060361546e-09
 End:   Calculate tridiagonal matrix components.
```

output/zvo_DynamicalGreen.dat





output/residual.dat

エネルギーごとの残差の収束

まとめ

1. LOB(P)CG法
   - 従来のLanczos法の入力ファイルから method=“CG”とすれば使える。
   - 固有ベクトルを直接計算する(残差のチェックをすること)
2. 静的相関関数のフーリエ変換
   - スタンダードモードと一緒に使うとすぐに逆格子空間の相関関数が計算できる。
   - エキスパートモードでも、サイトの座標ファイルを自分で用意すれば可能
   - 詳しくはマニュアル参照
3. スペクトル計算
   - 収束判定の閾値はもう少し大きくてもよいかもしれない。
   - residual.datのマニュアルはまだ無い
   - 残差のチェックをすること
- 残差の減少をチェックすること

参考文献: “線形計算の数理”,杉原正顯, 室田一雄