

Qbasis Manual

Zhentaο Wang

September 1, 2017

Contents

1	Introduction	2
2	Model representation	3
2.1	Matrix representation of local Hilbert Space	3
2.2	Lin table	5
3	Finite size cluster	6
4	Symmetry	7
4.1	Translation symmetry	7
4.2	Weisse table	7
5	Eigenvalue problem	8
5.1	Lanczos algorithm	8
5.1.1	Description	8
5.1.2	Implementation	9
6	Code examples	11
6.1	Exact diagonalization	11

Chapter 1

Introduction

Chapter 2

Model representation

2.1 Matrix representation of local Hilbert Space

Fermi-Hubbard Model

$$\mathcal{H} = -t \sum_{\langle ij \rangle} \sum_{\sigma} (c_{i\sigma}^{\dagger} c_{j\sigma} + h.c.) + U \sum_i n_{i\uparrow} n_{i\downarrow}. \quad (2.1.1)$$

Local Hilbert space is 4-dimensional:

$$\{|0\rangle, |\uparrow\rangle, |\downarrow\rangle, |\uparrow\downarrow\rangle\}. \quad (2.1.2)$$

In this basis:

$$c_{\uparrow} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (2.1.3a)$$

$$c_{\downarrow} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (2.1.3b)$$

From these two operators as input, the code is able to automatically derive the following operators:

$$c_{\uparrow}^{\dagger} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (2.1.4a)$$

$$c_{\downarrow}^{\dagger} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}, \quad (2.1.4b)$$

$$n_{\uparrow} = \begin{pmatrix} 0 & & & \\ & 1 & & \\ & & 0 & \\ & & & 1 \end{pmatrix}, \quad (2.1.4c)$$

$$n_{\downarrow} = \begin{pmatrix} 0 & & & \\ & 0 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}. \quad (2.1.4d)$$

t-J Model

$$\mathcal{H} = -t \sum_{\langle ij \rangle} \sum_{\sigma} (c_{i\sigma}^{\dagger} c_{j\sigma} + h.c.) + J \sum_{\langle ij \rangle} \left[\frac{S_i^+ S_j^- + S_i^- S_j^+}{2} + S_i^z S_j^z - \frac{1}{4} n_i n_j \right], \quad (2.1.5)$$

where

$$S_i^+ = c_{i\uparrow}^{\dagger} c_{i\downarrow}, \quad (2.1.6a)$$

$$S_i^- = c_{i\downarrow}^{\dagger} c_{i\uparrow}, \quad (2.1.6b)$$

$$S_i^z = \frac{1}{2} (c_{i\uparrow}^{\dagger} c_{i\uparrow} - c_{i\downarrow}^{\dagger} c_{i\downarrow}). \quad (2.1.6c)$$

Local Hilbert space is 3-dimensional:

$$\{|0\rangle, |\uparrow\rangle, |\downarrow\rangle\}. \quad (2.1.7)$$

In this basis:

$$c_{\uparrow} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (2.1.8a)$$

$$c_{\downarrow} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (2.1.8b)$$

and all other derived operators can be derived automatically by the code.

Spinless Fermion

$$\mathcal{H} = -t \sum_{\langle ij \rangle} (c_i^{\dagger} c_j + h.c.) + V_1 \sum_{\langle ij \rangle} n_i n_j. \quad (2.1.9)$$

Local Hilbert space is 2-dimensional:

$$\{|0\rangle, |1\rangle\}. \quad (2.1.10)$$

In this basis:

$$c = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}. \quad (2.1.11)$$

Bose-Hubbard Model

$$\mathcal{H} = -t \sum_{\langle ij \rangle} (b_i^\dagger b_j + h.c.) + \frac{U}{2} \sum_i n_i (n_i - 1) \quad (2.1.12)$$

Local Hilbert space restricted to at most N_{max} bosons:

$$\{|0\rangle, |1\rangle, \dots, |N_{max}\rangle\}. \quad (2.1.13)$$

In this basis:

$$b = \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \ddots & \sqrt{N_{max}} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (2.1.14)$$

and other operators should be automatically derived.

2.2 Lin table

Chapter 3

Finite size cluster

Chapter 4

Symmetry

4.1 Translation symmetry

4.2 Weiss table

Chapter 5

Eigenvalue problem

5.1 Lanczos algorithm

5.1.1 Description

The m -step Lanczos algorithm performs a factorization in the Krylov subspace:

$$HV = VT + b_m \mathbf{v}_m \mathbf{e}_m^T, \quad (5.1.1)$$

where H is the $N \times N$ matrix of our problem, $V = (\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_{m-1})$ is $N \times m$ matrix formed by the basis in the Krylov subspace, T is the tridiagonal Hessenberg matrix

$$T = \begin{pmatrix} a_0 & b_1 & & & \\ b_1 & a_1 & \ddots & & \\ & \ddots & \ddots & b_{m-1} & \\ & & b_{m-1} & a_{m-1} & \end{pmatrix}, \quad (5.1.2)$$

and $\mathbf{e}_m^T = (0, 0, \dots, 0, 1)$ has only one non-zero element.

Note that \mathbf{v}_i are orthonormal, i.e.

$$V^\dagger V = 1. \quad (5.1.3)$$

We can diagonalize the Hessenberg matrix T , giving eigen-pair (θ_i, \mathbf{s}_i) satisfying

$$T \mathbf{s}_i = \mathbf{s}_i \theta_i, \quad (5.1.4)$$

With such information, we can form the Ritz pair (θ_i, \mathbf{y}_i) which is a good approximation of the eigenvalues and eigenvectors of the original problem H , where

$$\mathbf{y}_i \equiv V \mathbf{s}_i. \quad (5.1.5)$$

To estimate the error of the eigenvalues, let's first roughly assume \mathbf{y}_i is indeed the true eigenvector with eigenvalue E_i , such that $H \mathbf{y}_i = E_i \mathbf{y}_i$. Then $\|H \mathbf{y}_i - \theta_i \mathbf{y}_i\| =$

$|E_i - \theta_i|$, which shows that $\|H\mathbf{y}_i - \theta_i\mathbf{y}_i\|$ is a good estimate of the error of eigenvalue E_i . Now we relax the assumption about y_i , and calculate $\|H\mathbf{y}_i - \theta_i\mathbf{y}_i\|$ directly:

$$\begin{aligned}\|H\mathbf{y}_i - \theta_i\mathbf{y}_i\| &= \|HV\mathbf{s}_i - V\mathbf{s}_i\theta_i\| = \|HV\mathbf{s}_i - VT\mathbf{s}_i\| \\ &= \|(HV - VT)\mathbf{s}_i\| = \|b_m\mathbf{v}_m\mathbf{e}_m^T\mathbf{s}_i\| \\ &= |b_m\mathbf{e}_m^T\mathbf{s}_i|,\end{aligned}\tag{5.1.6}$$

which shows that we can use $|b_m\mathbf{e}_m^T\mathbf{s}_i|$ as the error estimation for the Lanczos procedure.

5.1.2 Implementation

Algorithm 5.1 Simple Lanczos algorithm to obtain extreme eigenvalue, 3-vector version

Require: Space for 3 vectors in RAM: $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$

Require: An array of pointers $\{\tilde{\mathbf{v}}_j \rightarrow \mathbf{v}_{j \bmod 3}\}$

Require: \mathbf{v}_0 normalized, $\mathbf{v}_1 = \mathbf{v}_2 = 0$, $b_0 = 0$.

```

1: for  $m = 1, 2, \dots$  do
2:    $\tilde{\mathbf{v}}_m = -b_{m-1}\tilde{\mathbf{v}}_{m-2}$  (copy from  $\tilde{\mathbf{v}}_{m-2}$ )
3:    $\tilde{\mathbf{v}}_m = H\tilde{\mathbf{v}}_{m-1} + \tilde{\mathbf{v}}_m$ 
4:    $a_{m-1} = (\tilde{\mathbf{v}}_m, \tilde{\mathbf{v}}_{m-1})$ 
5:    $\tilde{\mathbf{v}}_m = \tilde{\mathbf{v}}_m - a_{m-1}\tilde{\mathbf{v}}_{m-1}$ 
6:    $b_m = \|\tilde{\mathbf{v}}_m\|$ 
7:    $\tilde{\mathbf{v}}_m = \tilde{\mathbf{v}}_m/b_m$ 
8:   calculate  $\{\theta_i, s_i\}$ 
9:   for the smallest  $\theta_i$ , calculate  $|b_m\mathbf{e}_m^T\mathbf{s}_i|$ . Stop if small enough
10: end for
```

Algorithm 5.2 Simple Lanczos algorithm to obtain extreme eigenvalue, 2-vector version

Require: Space for 2 vectors in RAM: $\mathbf{v}_0, \mathbf{v}_1$

Require: An array of pointers $\{\tilde{\mathbf{v}}_j \rightarrow \mathbf{v}_{j \bmod 2}\}$

Require: \mathbf{v}_0 normalized, $\mathbf{v}_1 = 0$, $b_0 = 0$.

```

1: for  $m = 1, 2, \dots$  do
2:    $\tilde{\mathbf{v}}_m = -b_{m-1}\tilde{\mathbf{v}}_{m-2}$  (rescale itself, since  $\tilde{\mathbf{v}}_m$  and  $\tilde{\mathbf{v}}_{m-2}$  overlap in RAM)
3:    $\tilde{\mathbf{v}}_m = H\tilde{\mathbf{v}}_{m-1} + \tilde{\mathbf{v}}_m$ 
4:    $a_{m-1} = (\tilde{\mathbf{v}}_m, \tilde{\mathbf{v}}_{m-1})$ 
5:    $\tilde{\mathbf{v}}_m = \tilde{\mathbf{v}}_m - a_{m-1}\tilde{\mathbf{v}}_{m-1}$ 
6:    $b_m = \|\tilde{\mathbf{v}}_m\|$ 
7:    $\tilde{\mathbf{v}}_m = \tilde{\mathbf{v}}_m/b_m$ 
8:   calculate  $\{\theta_i, s_i\}$ 
9:   for the smallest  $\theta_i$ , calculate  $|b_m\mathbf{e}_m^T\mathbf{s}_i|$ . Stop if small enough
10: end for
```

In Fig. 5.1.1 we show the evolution of Ritz values and three possible stopping criteria:

1. $|b_m e_m^T s_i|$.
2. Relative change of the ground state energy: $|(\theta_0^m - \theta_0^{m-1})/\theta_0^m|$.
3. Relative change of the 1st excited state energy: $|(\theta_1^m - \theta_1^{m-1})/\theta_1^m|$.

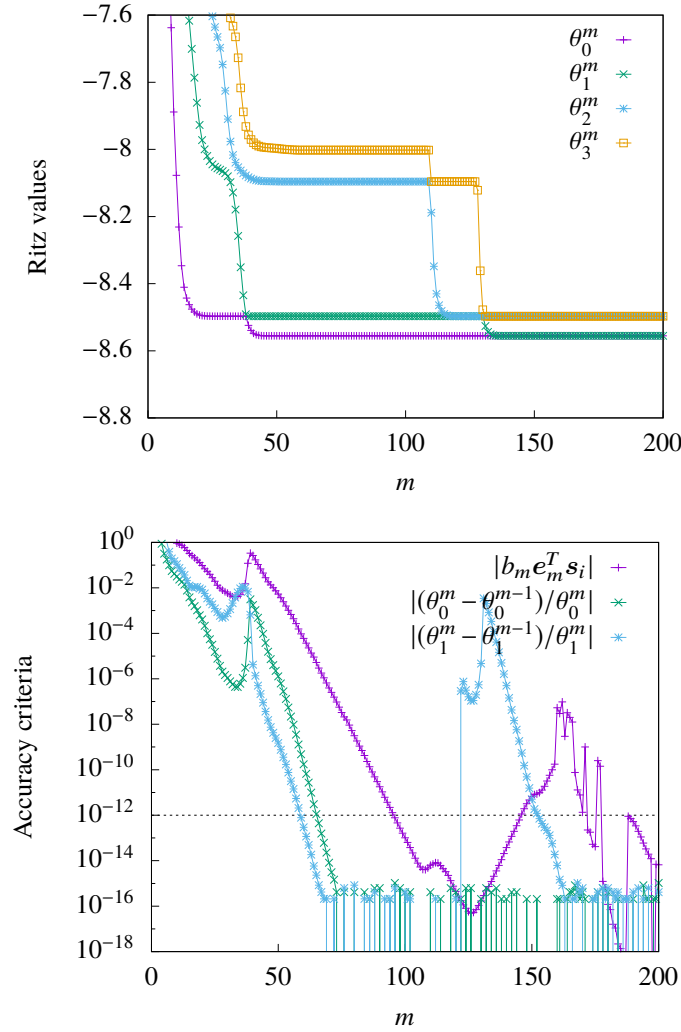


Figure 5.1.1: Evolution of Ritz values for spin- $\frac{1}{2}$ Heisenberg model on 4×4 triangular lattice, with periodic boundary condition and $S_z^{\text{total}} = 0$, $J = 1$. Starting vector is chosen as $(1, 1, 1, \dots, 1)$.

Chapter 6

Code examples

6.1 Exact diagonalization