

Exact Diagonalization Notes

Tsung-Han Lee

May 24, 2015

1 Hamiltonians and Full diagonalization:

$$H = t \sum_{i=i}^{N-1} (c_{i\sigma}^\dagger c_{i+1\sigma} + c_{i+1\sigma}^\dagger c_{i\sigma}) + \sum_{i=i}^N \epsilon_i c_{i\sigma}^\dagger c_{i\sigma} + \sum_{i=i}^N U_i c_{i\uparrow}^\dagger c_{i\uparrow} c_{i\downarrow}^\dagger c_{i\downarrow}$$

a) Anderson model: $\epsilon_i = \epsilon \delta_{i,1}$ and $U_i = U \delta_{i,1}$. Only the first site has Coulomb repulsion and atomic energy.

b) Hubbard model: $\epsilon_i = \epsilon$ and $U_i = U$.

The basis state representation is chosen in the convention $c_{1\uparrow}^\dagger \dots c_{N\uparrow}^\dagger c_{1\downarrow}^\dagger \dots c_{N\downarrow}^\dagger |0\rangle$, such that the hopping term for the nearest neighbor is always positive. In the code, the basis is represented as 0 (empty), -1 (spin down), 1 (spin up), and 2 (double occupied), so the basis looks like $|0, 1, -1, 2\rangle$. The basis is built based on two symmetries: the total charge (N) and the total spin S_Z . The quantum number is represented as Q and S_Z , each Q and S_Z has subspace r . Normally the highly symmetric state has the largest subspace, e.g. for $N = 4$, $Q = 0$ and $S_Z = 0$ has $r = 1 \dots 400$ 400 bases. Note that in the code, the real S_Z need to be divided by 2, because the code's S_Z is set to integer for the convenience of building basis. Similarly, the real total charge Q need to be added by a total size of the chain N , since it runs from $-N \leq Q \leq N$ for the convenience of building basis.

The full exact diagonalization is performed in each subspace (block diagonal for each Q and S_Z) using LAPACK routine dsyev, and the energies is recorded in "energiesXXXXX.dat".

2 Zero Temperature Spectral Function and Green's function:

Here we discuss the implementation of zero temperature Lehmann representation for spectral function. For simplicity, we calculate the spectral function at the first site with spin up, $A_{1\uparrow}(\omega)$.

$$A_{1\uparrow}(\omega) = \sum_{Q, S_Z, r} |\langle diag < 0, 0, 0 | c_{1\uparrow}^\dagger | Q, S_Z, r \rangle_{diag}|^2 \delta(\omega - E_{0,0,0} + E_{Q,S_Z,r}) + \sum_{Q, S_Z, r} |\langle diag < 0, 0, 0 | c_{1\uparrow}^\dagger | Q, S_Z, r \rangle_{diag}|^2 \delta(\omega - E_{Q,S_Z,r} + E_{0,0,0})$$

Since we know that onlt the state $Q = 1, S_Z = 1$ or $Q = -1, S_Z = -1$, for $c_{1\uparrow}$ and $c_{1\downarrow}^\dagger$ respectly, can survive for the bracket, we only need to focus on their subspace r . Noted that the subscript *diag* means the $|Q, S_Z, r\rangle_{diag}$ is the eigenvector for the energy $E_{Q, S_Z, r}$. Here we focus on the $_{diag} \langle 0, 0, 0 | c_{1\uparrow} | 1, 1, r \rangle_{diag}$ term, the other term can be calculated similarly.

The $_{diag} \langle 0, 0, 0 | c_{1\uparrow} | 1, 1, r \rangle_{diag}$ is not easy to calculate, because each eigenstate $|Q, S_Z, r\rangle_{diag}$ is the mix state of the original basis $|Q, S_Z, r\rangle_{old}$, where $|Q, S_Z, r\rangle_{old}$ is the primitive basis, e.g. $|0, 1, -1, 2\rangle$ for $Q = 0, S_Z = 0$. However, the $_{old} \langle 0, 0, 0 | c_{1\uparrow} | 1, 1, r \rangle_{old}$ is easy to calculate. So we can do a basis transformation:

$$_{diag} \langle 0, 0, 0 | c_{1\uparrow} | 1, 1, r \rangle_{diag} = \sum_{r'} \sum_{r''} _{diag} \langle 0, 0, 0 | 0, 0, r' \rangle_{old} _{old} \langle 0, 0, r' | c_{1\uparrow} | 1, 1, r'' \rangle_{old} \\ _{old} \langle 1, 1, r'' | 1, 1, r \rangle_{diag}$$

The term $_{diag} \langle 0, 0, 0 | 0, 0, r' \rangle_{old}$ and $_{old} \langle 1, 1, r'' | 1, 1, r \rangle_{diag}$ are the orthonormal transformation matrices, which is stored in the code variable "states". Thus, the bracket can be easily evaluate using the simple matrix multiplication. We first evaluate the matrix $_{old} \langle 0, 0, r' | c_{1\uparrow} | 1, 1, r'' \rangle_{old}$ by simple index look up, then multiply it by transform matrix.

$$_{diag} \langle 0, 0, 0 | c_{1\uparrow} | 1, 1, r \rangle_{diag(1 \times M)} = U_{0,0(1 \times L)}^T \cdot _{old} \langle 0, 0, r' | c_{1\uparrow} | 1, 1, r'' \rangle_{old(L \times M)} \cdot U_{1,1(M \times M)}$$

The dimension for $U_{0,0}^T$ is $1 \times L$, where L is the size of subspace $Q = 0, S_Z = 0$. The dimension for $U_{1,1}$ is $M \times M$, where M is the size of subspace $Q = 1, S_Z = 1$. After obtained $_{diag} \langle 0, 0, 0 | c_{1\uparrow} | 1, 1, r \rangle_{diag}$, the square of it is called spectral weight, which is printed out in "spectralWeightsXXXXX.dat". Plugging in the energies $E_{1,1,r}$ and the corresponding spectralweight $|_{diag} \langle 0, 0, 0 | c_{1\uparrow} | 1, 1, r \rangle_{diag}|^2$ into $A_{1\uparrow}(\omega)$ we obtain the electron part of the spectral function. The hole part $|_{diag} \langle 0, 0, 0 | c_{1\uparrow} | -1, -1, r \rangle_{diag}|^2$ can be calculated similarly. We can check that:

$$\sum_r (|_{diag} \langle 0, 0, 0 | c_{1\uparrow} | -1, -1, r \rangle_{diag}|^2 + |_{diag} \langle 0, 0, 0 | c_{1\uparrow} | 1, 1, r \rangle_{diag}|^2) = 1$$

the total spectralweight is 1. Finally, the spectralfunction is outputed as "spectralFunctionXXXXX.dat". The Green's function can be evaluated by the Kramer-Kronig relation:

$$G_{1\uparrow}(\omega) = \int_{-\infty}^{\infty} d\omega' \frac{A(\omega')}{\omega - \omega'}$$

and the output is "greenFunctionXXXXX.dat".

3 Examples:

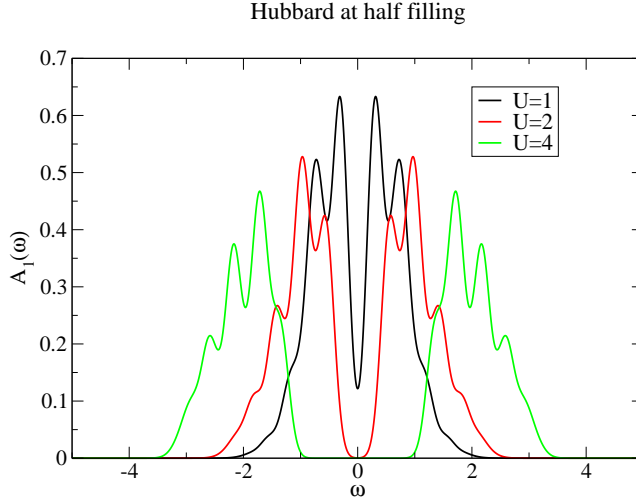


Figure 1: $\mu = -U/2$, $t = -0.5$, and broadening is 0.2.

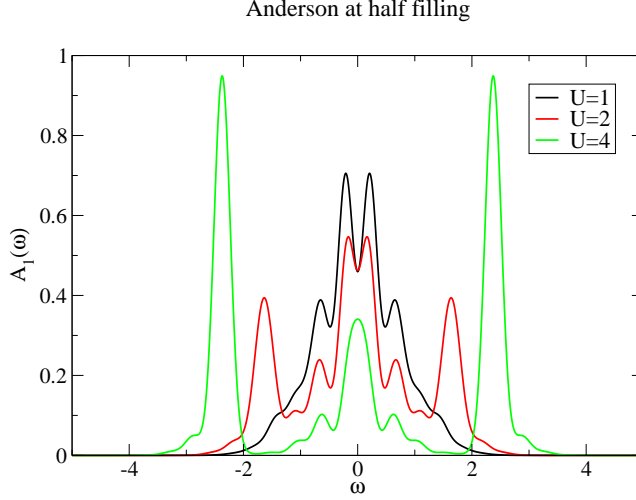


Figure 2: $\mu = -U/2$, $t = -0.5$, and broadening is 0.2.

4 Lanczos Algorithm:

In the Lanczos version of the exact diagonalization, we use the iterative optimization method to minimize the energies in the N dimensional Krylov space,

i.e. ($|u_0\rangle, H|u_0\rangle, H^2|u_0\rangle \dots H^N|u_0\rangle$). The energy converges to the exact ground state energy efficiently with in few ten to 200 iterations. In this algorithm, no full diagonalization ($O(N^3)$) is used. The most time consuming calculation is the $H|u_n\rangle$, matrix multiplication which can be done parallelly using openmp.

The Lanczos algorithm is stated below:

1. Choosing a initial state $|u_0\rangle = \sum_{Q, S_z, r} c_{Q, S_z, r} |Q, S_z, r\rangle$, where the coefficient $c_{Q, S_z, r}$ are chosen randomly using random number generator and $|Q, S_z, r\rangle$ are the original basis in quantum number Q and S_z .
2. Generate the Lanczos basis using the relation

$$|u_{n+1}\rangle = H|u_n\rangle - a_n|u_n\rangle - b_n^2|u_{n-1}\rangle$$

where

$$a_n = \frac{\langle u_n | H | u_n \rangle}{\langle u_n | u_n \rangle}$$

and

$$b_n^2 = \frac{\langle u_n | u_n \rangle}{\langle u_{n-1} | u_{n-1} \rangle}.$$

Here $b_0 = 0$, $|u_{-1}\rangle = 0$, so the first two iterations, $|u_0\rangle$ and $|u_1\rangle$, needs to be done separately.

3. Storing the resulting tridiagonal matrix in normalized lanczos vectors

$$\begin{array}{cccccc} a_0 & b_1 & . & 0 & 0 \\ b_1 & a_1 & b_2 & . & 0 \\ . & b_2 & a_2 & . & . \\ 0 & . & . & . & b_n \\ 0 & 0 & . & b_n & a_n \end{array}$$

in to two vectors, then diagonalize the matrix by pass the vectors (a_0, a_1, \dots, a_n) and (b_1, b_2, \dots, b_n) to the LAPCAK routine dstev. **Note that we expand the matrix in the normalized lanczos vectors $|\phi_n\rangle = \frac{|u_n\rangle}{\langle u_n | u_n \rangle}$, so the off diagonal term is b_n instead of b_n^2 .**

4. Compare the ground state energies $|E_{g.s, N} - E_{g.s, N-1}| < \epsilon$. If the condition is satisfied, we obtain the converged ground state energies. If not, got to 2.

The algorithm is fast and can be done without storing the hamiltonian matrix (Matrix free, no memory problem). The most time consuming part is the $H|u_n\rangle$ operation. Here we briefly discuss the implementation. The logic is basically the same as the full diagonalization in section 1. However, since we only need to store the element a_n and b_n , there is no need to store the entire

hamiltonian. We only need to store the lanczos state $|u_n\rangle$, $|u_{n+1}\rangle$, and $|u_{n-1}\rangle$. Now, we focus on the term

$$H|u_n\rangle = H \sum_{Q,S_Z,r} c_{Q,S_Z,r} |Q, S_Z, r\rangle = \sum_{Q'',S_Z'',r''} |Q'', S_Z'', r''\rangle \langle Q'', S_Z'', r''| H \sum_{Q',S_Z',r'} |Q', S_Z', r'\rangle \langle Q', S_Z', r'| \sum_{Q,S_Z,r} c_{Q,S_Z,r} |Q, S_Z, r\rangle$$

since Q and Q' , S_Z and S_Z' is orthogonal (block diagonal matrix) we can do the matrix multiplication in each quantum number space.

$$H|u_n\rangle = \sum_{Q,S_Z,r} \sum_{r'} \langle Q, S_Z, r| H |Q, S_Z, r'\rangle c_{Q,S_Z,r'} |Q, S_Z, r\rangle \quad (1)$$

So $H|u_n\rangle$ is a vector with a coefficient, $\sum_{r'} \langle Q, S_Z, r| H |Q, S_Z, r'\rangle c_{Q,S_Z,r'}$, for each basis $|Q, S_Z, r\rangle$. The coefficient can be sum up with respect to r' using the same look up method as we construct the hamiltonian matrix, but without storing the matrix elements. Thus, we can save a lot of memory.

5 Parallel:

As the size of the system become large, $N \geq 10$, the hilbert subspace becomes very large even with the symmetry implementation. Although Lanczos method is matrix free (memory free), looping through the hilbert space is still very time consuming. It is possible to parallelize the $H|u_n\rangle$ using simple openmp. The idea is just parallelize the r for loop (matrix row) in Eq.1 using the command “#pragma omp parallel for”. The schemetic representation is as below

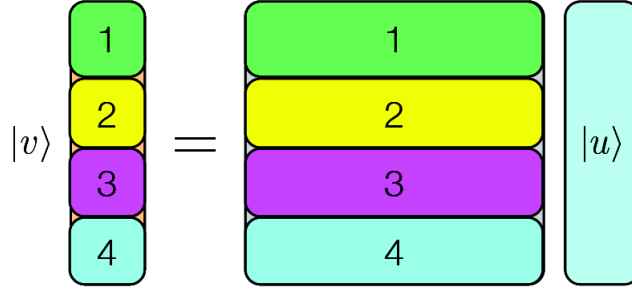


Figure 3: From Dr. Andreas Lauchli lecture powerpoints.

6 Gound State Wavefunction:

After obtain the converged ground state energy, we need to find the ground state wavefunction. While diagonalizing the tridiagonal matrix, we are diagonalized the approximated matrix in the **Normalized** Lanczos basis, i.e. ($|\phi_0\rangle, |\phi_1\rangle, \dots, |\phi_n\rangle$). Thus the approximate ground state wavefunction is in Lanczos basis

$$|\Omega\rangle = \sum_{i=0}^N \langle \phi_i | \Omega \rangle |\phi_i\rangle$$

To calculate the $|\Omega\rangle$ in original basis $|Q, S_Z, r\rangle$, we need to retrace all the $|\phi_n\rangle$ by perfoming the Lanczos algorithm again, and progressively construct the $|\Omega\rangle$

$$|\Omega\rangle = \sum_{Q, S_Z, r} \sum_{i=0}^N \langle \phi_i | \Omega \rangle \langle Q, S_Z, r | \phi_i \rangle |Q, S_Z, r\rangle.$$

The coefficient $\langle \phi_i | \Omega \rangle$ is calculated by diagonalizing the tridiagonal matrix, and the $\langle Q, S_Z, r | \phi_i \rangle$ is calculated by retracing the Lanczos algorithm. If using the unnormalized lanczos vector $|u_i\rangle$, remember to devide by the normalization constant $\langle u_i | u_i \rangle$ for each coefficients.

7 Green's function:

After obtain the ground state wavefunction, we can start calculating the Green's function or other dynamical quantity. We need to start another Lanczos process from the state $c_{i,\uparrow}^\dagger|\Omega\rangle$ (electron) or $c_{i,\uparrow}|\Omega\rangle$ (hole). These states can be easily calculated by look up the ground state wavefunction and the bases in the sector $q = 1$ and $S_Z = 1$ (electron), or $q = -1$ and $S_Z = -1$ (hole), since the ground state loactes at $q = 0$ and $S_Z = 0$.

After obtain the $c_{i,\uparrow}^\dagger|\Omega\rangle$ (electron) and $c_{i,\uparrow}|\Omega\rangle$ (hole), we recalculate the Lanczos $a_{n,e/h}$ and $b_{n,e/h}$ starting from each of these states. We also need the normalization $b_{0,e} = \langle \Omega | c_{i,\uparrow} c_{i,\uparrow}^\dagger | \Omega \rangle$ and $b_{0,h} = \langle \Omega | c_{i,\uparrow}^\dagger c_{i,\uparrow} | \Omega \rangle$ for constructing the Green's function. Once we have $a_{n,e/h}$, $b_{n,e/h}$ and $b_{0,e/h}$, the Green's function can be calculated by this continued fraction recursively,

$$G_{ii}(\omega) = \frac{b_{0,e}^2}{\omega + i\eta + E_{g.s.} - a_{0,e} - \frac{b_{1,e}^2}{\omega + i\eta + E_{g.s.} - a_{1,e} - \frac{b_{2,e}^2}{\omega + i\eta + E_{g.s.} - a_{2,e} - \dots}}} + \frac{b_{0,h}^2}{\omega + i\eta - E_{g.s.} + a_{0,h} - \frac{b_{1,h}^2}{\omega + i\eta - E_{g.s.} + a_{1,h} - \frac{b_{2,h}^2}{\omega + i\eta - E_{g.s.} + a_{2,h} - \dots}}}.$$