Exact Diagonalization Notes

Tsung-Han Lee

May 13, 2015

1 Hamiltonians and Full diagonalization:

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

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 represntation is chosen in the convention $c^{\dagger}_{1\uparrow}...c^{\dagger}_{N\uparrow}c^{\dagger}_{1\downarrow}...c^{\dagger}_{N\downarrow}|0>$, 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 $|0, 1, -1, 2>$. The basises is build based on two symmetries: the total charge (N) and the total spin S_Z. The quantum number is represnted 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 $r = 1...400$ 400 basises. Noted that in the note, the real S_Z need to be devided by 2, because the code's S_Z is set to integer for the convinience of building basis. Similarily, 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 convinience 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, $A_{1\uparrow}(\omega)$.

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

$\sum_{Q,S_Z,r} |_{diag} < 0,0,0|c^{\dagger}_{1\uparrow}|Q, S_Z, r >_{diag} |^2\delta(\omega - E_{Q,S_Z,r} + E_{0,0,0})$

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

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

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

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

${}_{diag} < 0, 0, 0|c_{1\uparrow}|1, 1, r >_{diag(1\times M)} = U^T_{0,0(1\times L)} \cdot {}_{old} < 0, 0, r'|c_{1\uparrow}|1, 1, r'' >_{old(L\times M)}$
$\cdot U_{1,1(M\times M)}$

The dimension for $U^T_{0,0}$ 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 ${}_{diag} < 0, 0, 0|c_{1\uparrow}|1, 1, r >_{diag}$, the square of it is called spectral weight, which is printed out in "spectralWeightsXXXXX.dat". Pluging in the energies $E_{1,1,r}$ and the corresponding spectralweight $|_{diag} < 0, 0, 0|c_{1\uparrow}|1, 1, r >_{diag}|^2$ into $A_{1\uparrow}(\omega)$ we obtain the electron part of the spectral function. The hole part $|_{diag} < 0, 0, 0|c_{1\uparrow}| - 1, -1, r >_{diag}|^2$ can be calculated similarily. We can check that:

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

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:

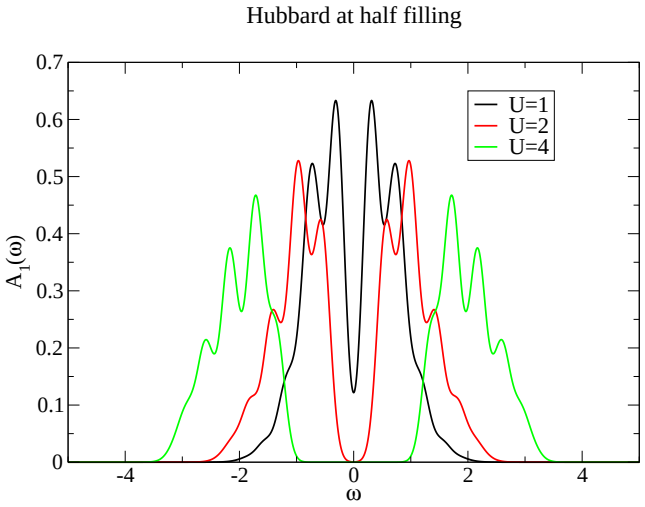

Figure 1: $\mu = -U/2$, $t = -0.5$, and broadening is 0.2.

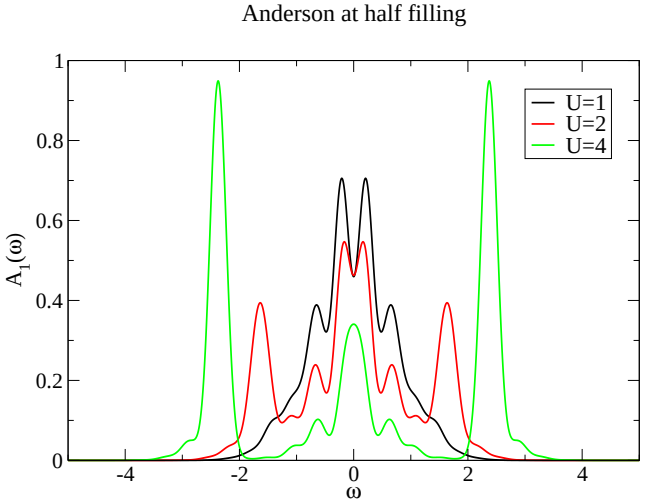

Figure 2: $\mu = -U/2$, $t = -0.5$, and broadening is 0.2.

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. $(|u_0>, H|u_0>, H^2|u_0> ... H^N|u_0>)$. 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 $H|u_n>$, matrix multiplication which can be done parallely using openmp.

The Lanczos algorithm is stated below:

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

2. Generate the Lanczos basis using the relation

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

where

$$a_n = \frac{<u_n|H|u_n>}{<u_n|u_n>}$$

and

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

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

3. Storing the resulting tridiagonal matrix

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

in to two vectors, then diagonalize the matrix by pass the vectors $(a_0, a_1, ..., a_n)$ and $(b_1, b_2, ..., b_n)$ to the LAPCAK routine dstev.

4. Compare the ground state energies $|E_{g.s,N} - E_{g.s,N-1}| < \epsilon$. If the condition is satisfied, we obtain theconvered 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 $H|u_n>$ operation. Here we breifly 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 $|u_n>$, $|u_{n+1}>$, and $|u_{n-1}>$. Now, we focus on the term

$$H|u_n>= H \sum_{Q,S_Z,r} c_{Q,S_Z,r}|Q,S_Z,r>=$$

$$\sum_{Q'',S''_Z,r''} |Q'',S''_Z,r''><Q'',S''_Z,r''|H \sum_{Q',S'_Z,r'} |Q',S'_Z,r'><Q',S'_Z,r'| \sum_{Q,S_Z,r} c_{Q,S_Z,r}|Q,S_Z,r>$$

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.

$$H|u_n>= \sum_{Q,S_Z,r} \sum_{r'} <Q,S_Z,r|H|Q,S_Z,r'> c_{Q,S_Z,r'}|Q,S_Z,r> \qquad (1)$$

So $H|u_n>$ is a vector with a coefficient, $\sum_{r'} <Q,S_Z,r|H|Q,S_Z,r'> c_{Q,S_Z,r'}$, for each basis $|Q,S_Z,r>$. 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, $N >= 10$, 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 consumming. It is possible to parallelize the $H|u_n>$ 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

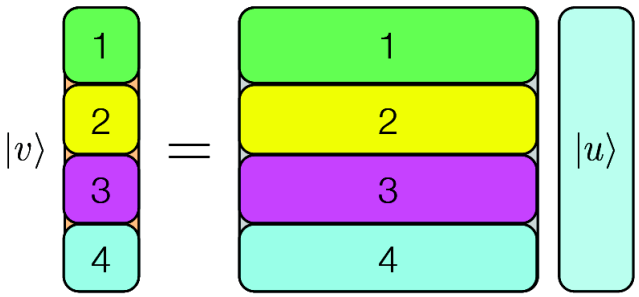

Figure 3: From Dr. Andreas Lauchli lecture powerpoints.