Writing an Exact Diagonalization (ED) Routine for the Hubbard Model Hamiltonian

Jixun Ding

Jan 25, 2019

The Hubbard model Hamiltonian (when only including nearest neighbor hopping), written in second quantization notation is

$$H = -t \sum_{\langle ij \rangle \sigma} \left[c^\dagger_{j\sigma} c_{i\sigma} + c^\dagger_{i\sigma} c_{j\sigma} \right] - \mu \sum_{i\sigma} n_{i\sigma} + U \sum_i \left(n_{i\uparrow} - 1/2 \right) \left(n_{i\downarrow} - 1/2 \right)$$

where i, j label spatial lattice sites, $\sigma = \{\uparrow, \downarrow\}$ labels electron spin, and the number operator $n_{i\sigma} = c^\dagger_{i\sigma} c_{i\sigma} = \{0, 1\}$ counts the number of electrons in any particular [site, spin] combination state.

As an example, think of a 1-D chain of $N = 4$ lattice sites. At each site, we can have 4 possible occupation states: empty, one $\uparrow$ electron, one $\downarrow$ electron, or a pair of $\uparrow\downarrow$ electrons. This means that the dimension of the full Hilbert space is $4^N = 256$. If there are $N = 8$ lattice sites, then the dimension of the full Hilbert space is $4^N = 65536$. The dimension of the full Hilbert space grows exponentially in N.

The idea behind "exact diagonalization" (ED) is the same as what we do in every introductory quantum course. We

1. Find a suitable complete orthonormal basis of the Hilbert space,
2. Write down the matrix representation of H in this basis, and
3. (Partially) diagonalize the Hamiltonian to find the energy eigenkets and eigenvalues. Usually, only the ground state wavefunction and energy is desired.

So really, this method is more suitably called "diagonalization". People use the word "exact", presumably to distinguish this method from other methods of working with the Hubbard model which don't explicitly find the energy eigenkets and eigenvalues.

1 Basis

In order to reduce the size of Hilbert space we have to work with, we observe that H as defined above commutes with the number operators

$$n_\uparrow = \sum_i n_{i\uparrow} \qquad n_\downarrow = \sum_i n_{i\downarrow}$$

which counts the total number of up electrons and total number of down electrons in the system, respectively.

It is a general fact that: given a symmetry of the system, i.e. an operator A that commutes with H, the Hamiltonian will not mix states from different eigenspaces of A. Hence, we can divide the Hilbert space into domains labeled by fixed $(n_\uparrow, n_\downarrow)$ numbers, and study each domain separately. Each domain has dimension

$$\binom{N}{n_\uparrow} \cdot \binom{N}{n_\downarrow}$$

The largest of these domains has dimension $\binom{N}{N/2} \cdot \binom{N}{N/2}$. For $N = 4$, this number is $\binom{4}{2} \cdot \binom{4}{2} = 36$.

The conservation of pair $(n_\uparrow, n_\downarrow)$ is equivalent to the conservation of pair $(n_e = n_\uparrow + n_\downarrow, S_z = (n_\uparrow - n_\downarrow)/2)$, the total number of electrons and the total spin-z component. So for $N = 4$, the pairs $(n_\uparrow = 3, n_\downarrow = 2)$ and $(n_e = 5, S_z = 1/2)$ label the same Hilbert space.

We are typically interested in the zero-spin, half-filling domain, which is labeled by $(n_\uparrow = N/2, n_\downarrow = N/2)$ or $(n_e = N, S_z = 0)$. Hence subsequently I will assume that we are working in this space.

Consider the $N = 4$ system again. Some basis vectors in our Hilbert space are

$$v = |\uparrow\ \uparrow\ -\ -\rangle \otimes |\downarrow\ -\ \downarrow\ -\rangle$$
$$u = |-\ \uparrow\ \uparrow\ -\rangle \otimes |-\ -\ \downarrow\ \downarrow\rangle$$

To work with this basis, we need to define a total order. The conventional method is to translate the electron positions to bit patterns, and order bit patterns by the integer values they correspond to. For our example, this makes a table (sometimes called a **Lin Table**) that looks like:

$\uparrow$ electron index i	bit pattern	integer value	$\downarrow$ electron index j	bit pattern	integer value
0	0011	3	0	0011	3
1	0101	5	1	0101	5
2	0110	6	2	0110	6
3	1001	9	3	1001	9
4	1010	10	4	1010	10
5	1100	12	5	1100	12

Which allows us to create a total ordering $n = 6i + j$. Looking back, our example basis vectors are in fact

$$v = |i = 5, j = 4\rangle = |n = 34\rangle$$
$$u = |i = 2, j = 0\rangle = |n = 12\rangle$$

2 Matrix Representation

To write down the matrix representation of H, we need to find out how H acts on basis vectors. The k-th column of H corresponds to the result of $H|k\rangle$.

The action of each term in H are:

- U term: count double occupancy, i.e. bitwise AND.
- μ term: count total particle number, which in our case is N.
- Hopping term: left or right shift of single bits, along with sign changes when electrons hop across the periodic boundary and reappear on the other side.

An example of the action of the hopping term, which mixes basis states:

$$v = |34\rangle = |1100\rangle \otimes |1010\rangle \qquad \Rightarrow \qquad -t\big(|1010\rangle \otimes |1010\rangle - |0101\rangle \otimes |1010\rangle \tag{1}$$
$$+ |1100\rangle \otimes |1100\rangle + |1100\rangle \otimes |1001\rangle + |1100\rangle \otimes |0110\rangle - |1100\rangle \otimes |0011\rangle\big) \tag{2}$$
$$= -t(|28\rangle - |10\rangle + |35\rangle + |33\rangle + |32\rangle - |30\rangle) \tag{3}$$

Line (2) is the result of $\uparrow$ electron hopping, and line (3) is the result of $\downarrow$ electron hopping. Combining the action of all three terms and applying the sum action to each basis vector produces the matrix representation of H in the ordered basis we have defined.

This H matrix has some convenient properties:

- If we let t, U, μ be real, as is usually the case, the matrix H is real & symmetric.
- Each column of H contains at most $2(n_\uparrow + n_\downarrow) + 1 = 2N + 1$ nonzero elements, so it is very sparse.

3 (Partial) Diagonalization

Due to the above properties of matrix H and due to the fact that we are usually only interested in studying the ground state and the first few excited states of H, we use a method called the **Lanczos** (pronounced lan-chose) **algorithm**[1], a sort of modified power iteration, to perform the diagonalization. The important properties of this algorithm is that it:

- Works for complex Hermitian or real symmetric matrices.
- Is very fast for sparse matrices.
- Is very fast when finding a small number of extreme eigenvalues, but may be slow if you ask for full diagonalization.

The naive form of the algorithm suffers from numerical instability, so we usually don't implement the diagonalization ourselves, but instead use a linear algebra package like ARPACK[2], which implements the Implicitly Restarted Lanczos Method (IRLM). Interfacing to ARPACK in python is handled by the `scipy.sparse` package.

4 Sanity Checks

For the case $N = 4, n_\uparrow = n_\downarrow = 2$, setting $t = 1, U = 5, \mu = 2$ produces a H matrix that looks like:

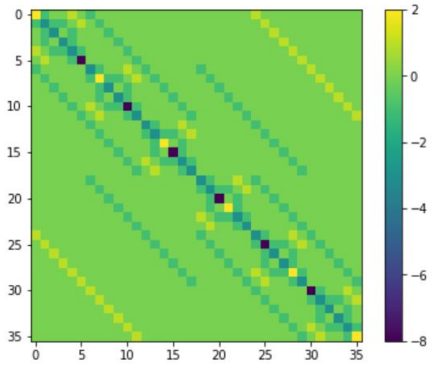

Figure 1: Plot of H matrix for $N = 4, n_\uparrow = n_\downarrow = 2$, and setting $t = 1, U = 5, \mu = 2$.

For the case $N = 6, n_\uparrow = n_\downarrow = 3$, the sparsity pattern of the H matrix looks as below. Compare Fig. 2 to plot on group website[3].

[1] `https://en.wikipedia.org/wiki/Lanczos_algorithm`
[2] `https://www.caam.rice.edu/software/ARPACK/`
[3] `https://web.stanford.edu/group/photontheory/computational_methods.html`

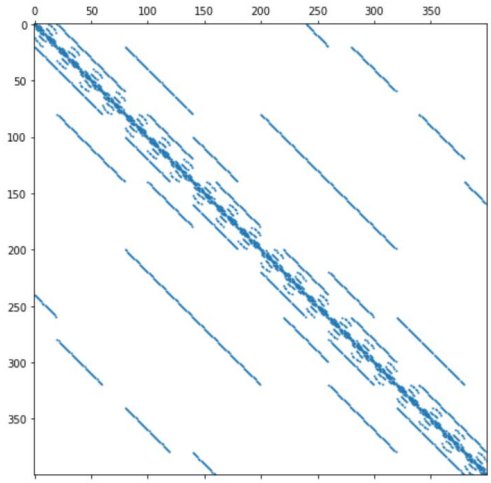

Figure 2: Sparsity pattern of H matrix for $N = 6, n_\uparrow = n_\downarrow = 3$.

//TODO: GS-energy vs U/t plot.
//TODO: XAS plot

5 A few more comments

//TODO: timing benchmark
Most of the time is spent on constructing the sparse matrix in coordinate format[4].
When using ED in real research problems, diagonalization is performed via several machines in parallel (see PARPACK and Python MPI 4). State of the art is approximately a 4-by-4 square lattice.
//TODO: Check SoA.
The advantages of ED:

- Conceptually straightforward. This is especially useful for pedagogical purposes.
- Produces the ground state wavefunction, which gives us complete knowledge of the state of the system at $T = 0\,K$, from which we can calculate any expectation value we desire.

The disadvantages of ED:

- Expensive in computation time and memory requirement, and hence limited to working with < 20 sites even with parallelization.
- Results obtained on small clusters must be extrapolated if we are modeling a large system.
- Worsens as we add next nearest neighbor, next-next nearest neighbor interactions...This is because the Lanczos algorithm relies on the sparsity of the Hamiltonian matrix.
- Cannot produce expectation values of observables at higher temperatures, where many states mix. This is because the Lanczos method is only good at producing a few extreme eigenkets.

Some of the disadvantages of ED are avoided by determinant quantum Monte Carlo (DQMC), but the DQMC method has its own issues. See other tutorials.

[4] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.coo_matrix.html`

References

[1] H. Fehske, R. Schneider, and A. Weisse, eds., *Computational Many-Particle Physics*, Springer-Verlag Berlin Heidelberg, 2008, ch. 18.