Hubbard Model

Garrett Stoll

December 2021

1 HubbardModel_func.py

HubbardModel_func.py sets up a Hubbard Model, using lists of complex numbers as the possible states (1's represent up spin and i's represent down spin). The function get_basis returns the list of all possible states and get_hamiltonian uses this list to return the Hamiltonian. Also present is a helper function for get_hamiltonian, called possible_states, it finds the off diagonals of the Hamiltonian. Functions get_hamiltonian and possible_states suffixed with _cyclic, treat the end sites as neighbors.

$$H = -t \sum_{<i,j>,\sigma} c^{\dagger}_{i\sigma} c_{j\sigma} + U \sum_{i} n_{i\downarrow} n_{i\uparrow} - \mu \sum_{i} (n_{i\downarrow} - n_{i\uparrow})$$

$$Heis = \frac{J}{4} \sum_{<i,j>} [2(c^{\dagger}_{i,\uparrow} c_{i,\downarrow} c^{\dagger}_{j,\downarrow} c_{j,\uparrow} + c^{\dagger}_{i,\downarrow} c_{i,\uparrow} c^{\dagger}_{j,\uparrow} c_{j,\downarrow} + (n_{i,\uparrow} - n_{i,\downarrow})(n_{j,\uparrow} - n_{j,\downarrow}]$$

$$| < \psi_f | e^{-i\tau_0 H_0(\Delta\tau)} ... e^{-i\tau_N H_n(\Delta\tau)} | \psi_i > |^2$$

2 TimeEvolution.py

Probability is defined as the probability of the system to be in a singly occupied state. Heisenberg value is defined as the overlap with the Heisenberg eigenvectors.

TimeEvolution.py sets up a half filled Hubbard system with constant spin (setup_model) and conducts a heuristic combinatorial search algorithm (find_solution) to find a combination of time evolution operators that minimize/maximize the U, probability, or Heisenberg values. The search algorithm creates a tree from the package anytree then allows this tree to grow by applying all time operators for the next time step (test_timeOps) and taking a step in the direction of the top five operators that minimize U or maximize probability or heisenberg value; this choice is indicated by the parameter uph_grow. This process is repeated

five times and then the tree is trimmed (reduced) to the solution that has the lowest average U value or highest average probability or Heisenberg value; this choice is indicated by the parameter uph_trim. This solution can then be plotted with plot_solution with the parameter, uph, indicating U, probability, or Heisenberg as the y-axis. There are also functions to get a list of time operators (get_timeOps) and to get the Heisenberg matrix (get_heisenberg) with helper function, exchanged_states.