

Equilibrium Phases in the 1-D Hubbard Model

Code Documentation

PHYS 416

Johann Gan

April 11, 2019

1 Contents

This document lists overall program descriptions and usage instructions. For other code details, see docstrings and comments in the actual Python files. For a high-level description of the methods used, see the “Methods” section in the main report.

2 ed.py

Sets up the Hamiltonian matrix in the number basis, and numerically solves the eigenvalue problem for the band structure (energy levels). The band structure is plotted in an energy diagram, along with the Fermi level (as a dashed red line) for reference. For even numbers of sites, the central (conduction) band is highlighted in orange, and the plot is zoomed in on the conduction band to see finer details (both the zoomed and original plots are saved).

Plots are saved in the following file format:

“fermi_hubbard_bands_n{n sites}_U{U value}_t{t value}.png”
“fermi_hubbard_bands_zoomed_n{n sites}_U{U value}_t{t value}.png” (even)

Configuration is at the start of the main script (near the bottom after all the function definitions). To use, simply run after setting the desired system parameters:

- n is the number of sites. Should be small ($\lesssim 7$).
- t is the hopping amplitude.
- U is the interaction strength.
- μ is the chemical potential.

Periodic boundary conditions are used. The program exploits the fact that the Hubbard Hamiltonian conserves total particle number, and solves for the energies by diagonalizing a bunch of subspaces where each subspace has a fixed total particle number. This greatly reduces the problem size. The central band is identified using a hierarchical clustering method.

3 energy_curves.py

Not an actual program. A module containing supporting functions for mean-field calculations of energies and magnetizations, assuming different phases (ferromagnetic or antiferromagnetic) in the system.

4 mft_compressibility.py

Computes and plots the $\rho(\mu)$ and $d\rho/d\mu$ curves under different Hubbard parameters U and t . Each data point (ρ, μ) is color-coded by the phase in which the state exists. Blue is for ferromagnetic, green is for paramagnetic, and red is for antiferromagnetic. A Monte Carlo sampling method for initial guesses is used along with a mean-field calculation, so results are not guaranteed to be converged (though convergence failure is rare). Any unconverged results are demarcated in the plots by dashed red lines.

Plots are saved in the following file format:

“fermi_hubbard_compressibility_U{U value}_t{t value}.png”.

Configuration is at the start of the main script (near the bottom after all the function definitions). To use, simply run after setting the desired system parameters:

- n is the number of sites. Should be relatively large to get a good finite-sized approximation of the chain.
- t is the hopping amplitude.
- U is the interaction strength.
- mus is an array of the chemical potentials for which to compute the corresponding density and compressibility.

5 mft_phase.py

Computes and plots the $U/t - \rho$ magnetic phase diagram for the Hubbard model. On the first plot, points on a 2D grid of U/t and ρ values are colored blue, green, or red for whether that point corresponds to a ferromagnetic, paramagnetic, or antiferromagnetic state, respectively. On the second plot, the same 2D grid points are colored according to the magnitude of the appropriate order parameter of its phase: magnetization for ferromagnetism, staggered magnetization for antiferromagnetism, and zero for paramagnetism.

Plots are saved in the file: “fermi_hubbard_phase_diagram.png”.

Configuration is at the start of the main script (near the bottom after all the

function definitions). To use, simply run after setting the desired system parameters:

- *U_over_t_max* is the highest value of U/t that will be computed and plotted on the phase diagram. The minimum value will always be zero.
- *U_over_t_npoints* is the number of discrete points to compute and plot between $U/t = 0$ and *U_over_t_max*.
- *nparticle_res* is the grid spacing of total particle number (density is computed as $\rho = n_{\text{particles}}/n_{\text{sites}}$). Must be even for the antiferromagnetic calculations to work.
- *nsites* is the number of sites. Should be relatively large to get a good finite-sized approximation of the chain. Note that particle number will be incremented from 0 all the way up to $2 * nsites$ in steps of *nparticle_res*, so a very large number of *nsites* will lead to a longer runtime.

Note: Depending on the number of phase-space points that need to be computed, the program may take a long time to complete.