Definitions

RDM: $\rho_{ij} = \langle \Psi | c_i^\dagger c_j | \Psi \rangle$

von Neumann entropy: $S = -\operatorname{Tr}[\rho \ln \rho] = -\sum_i \lambda_i \ln \lambda_i$

λ_i are the eigenvalues of ρ.

Possible questions:

Is the entropy extensive?

Do quantum chemistry methods and QMC agree on the amount of entropy in H chains? (how far do we have to push them)

Does the entropy converge to the exact result at the same rate as the energy? For different methods?

Plan

1) Hydrogen chain: r = [1.4,2.0, 3.0]. Number of atoms =
2) **Data generation**: Compute 1-particle density matrices and energy for DMC and CCSD.
    1) CCSD convergence:
        1) basis
    2) DMC convergence
        1) Use HCI to select determinants. Converge with HCI tolerance
        2) timestep: .02, .01 (see previous graphs for convergence). Settled on 0.02
3) Analysis: Diagonalize density matrix, make plots on the right.

We were having trouble with bias in the entropy due to statistical noise in QMC.

Today: I found a way to diagonalize the density matrix from QMC with minimal bias.

Key final plots:

1) Energy vs entropy as a function of basis for CCSD and # determinants for DMC (checks convergence)
2) Entropy versus system size (checks size consistency.)

Test Data

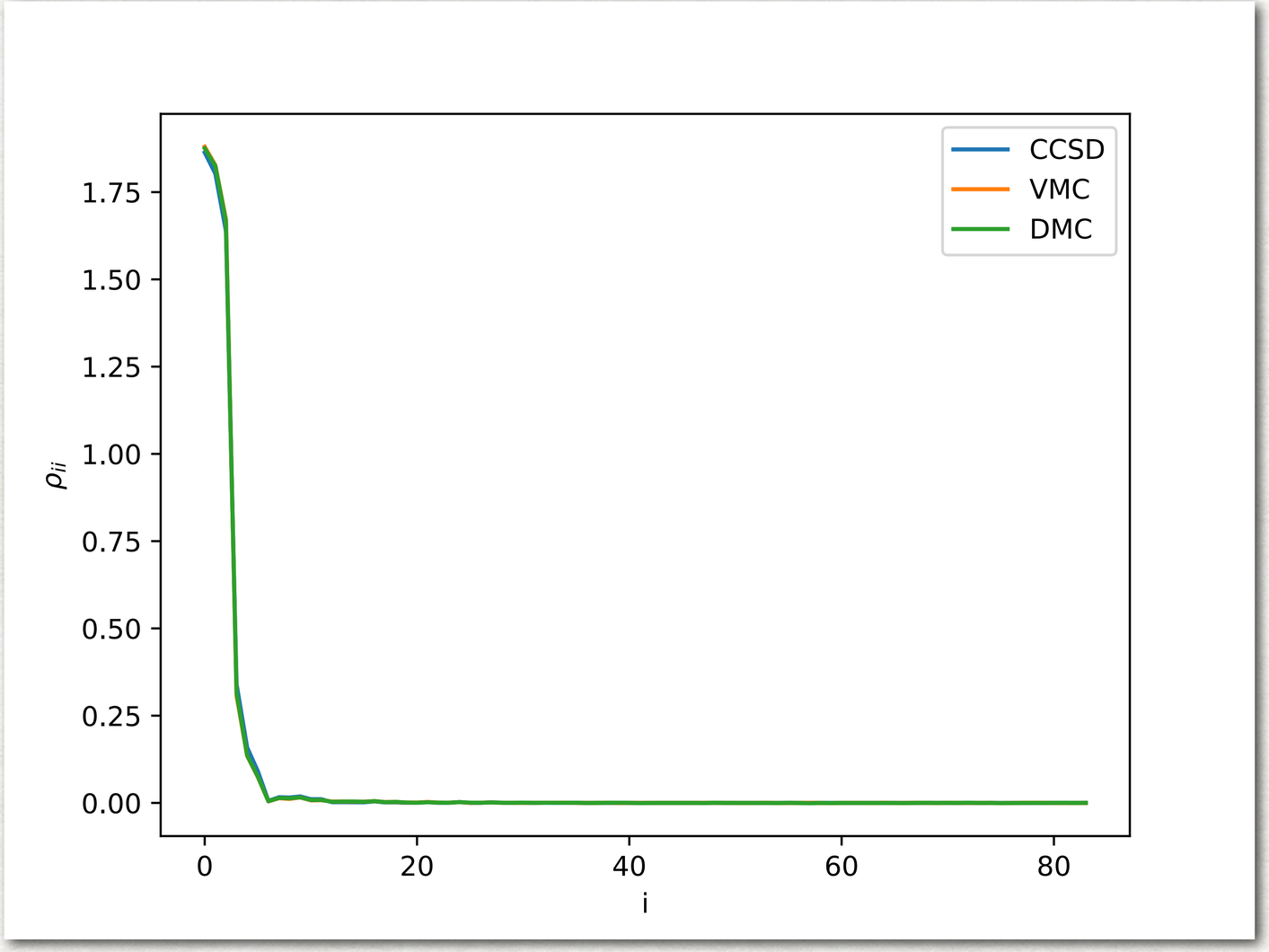
6 hydrogen atoms, r=3.0 Bohr. All done with ccecp.

cc.chk: 1-RDM from CCSD and CCSD(T) in MO basis, triple zeta basis
h6_3.0_vtz_dmc_hci0.02_0_large_0_12800_0.02.chk: 1-RDM from DMC in MO basis
h6_3.0_vtz_vmc_hci0.02_0_large_0_12800.chk: 1-RDM from DMC in MO basis

VMC optimization was performed for up to 12800 configurations. Confirmed that the energy difference between 6400 and 12800 configurations was less than 1 mHa.

Average error bars on the 1-RDMs:
DMC: 0.000288
VMC: 0.00018

Comparison of Density Matrices



All 3 methods get very similar RDMs.

The entropies should be similar, I think.

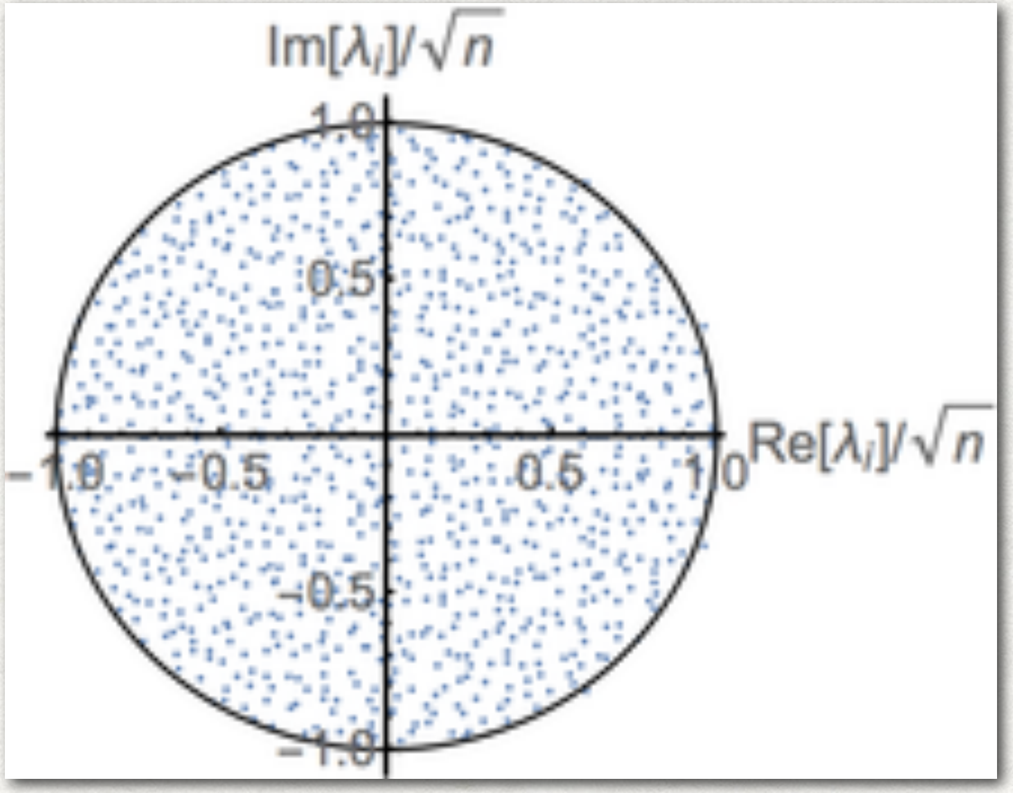
Stochastic errors are not visible here.

look_at_data.py

Literature: Random Matrix

If a $N \times N$ matrix is filled with random variables $A_{ij} \sim N(0,\sigma)$

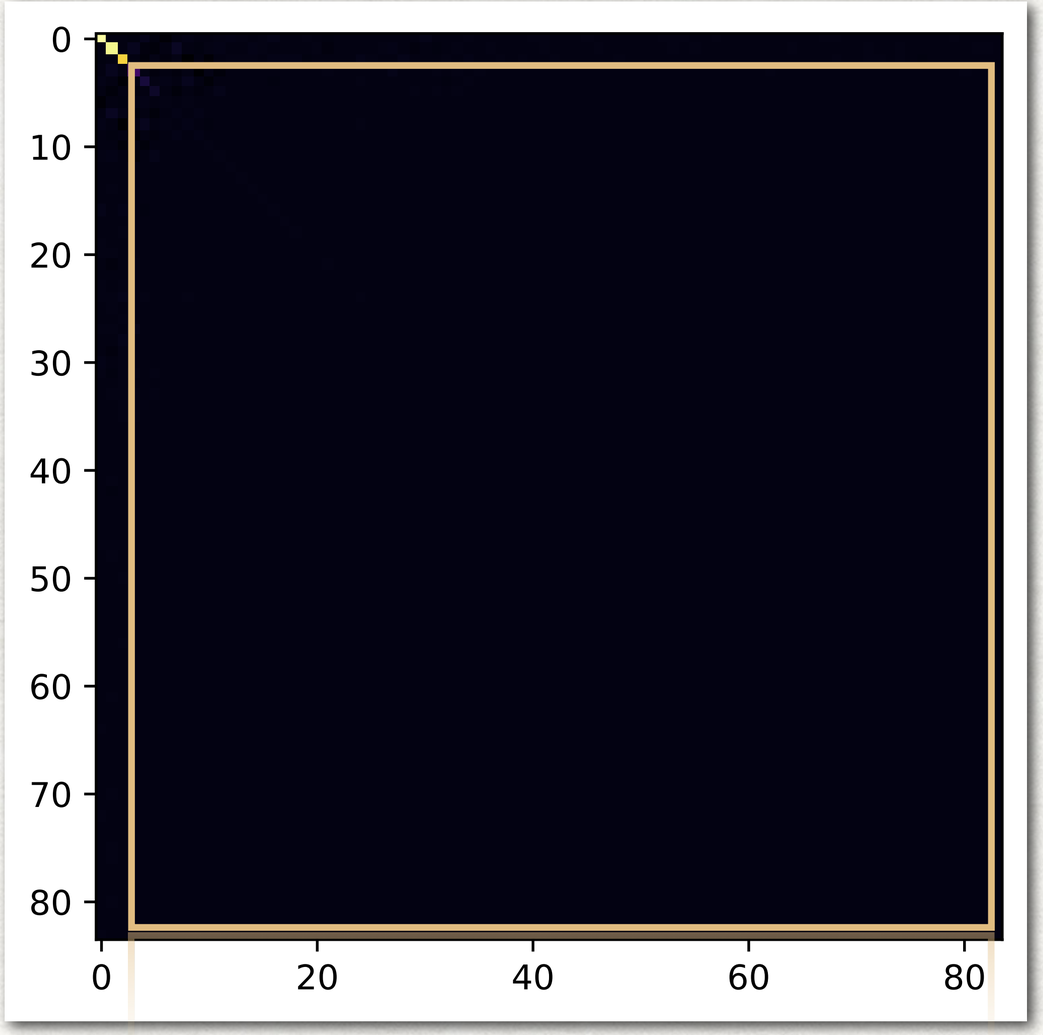
Then the eigenvalue spectrum has a circular distribution with radius $\sigma\sqrt{N}$



https://en.wikipedia.org/wiki/Circular_law

Ginibre. J. Math. Phys. 6 440 https://doi.org/10.1063%2F1.1704292

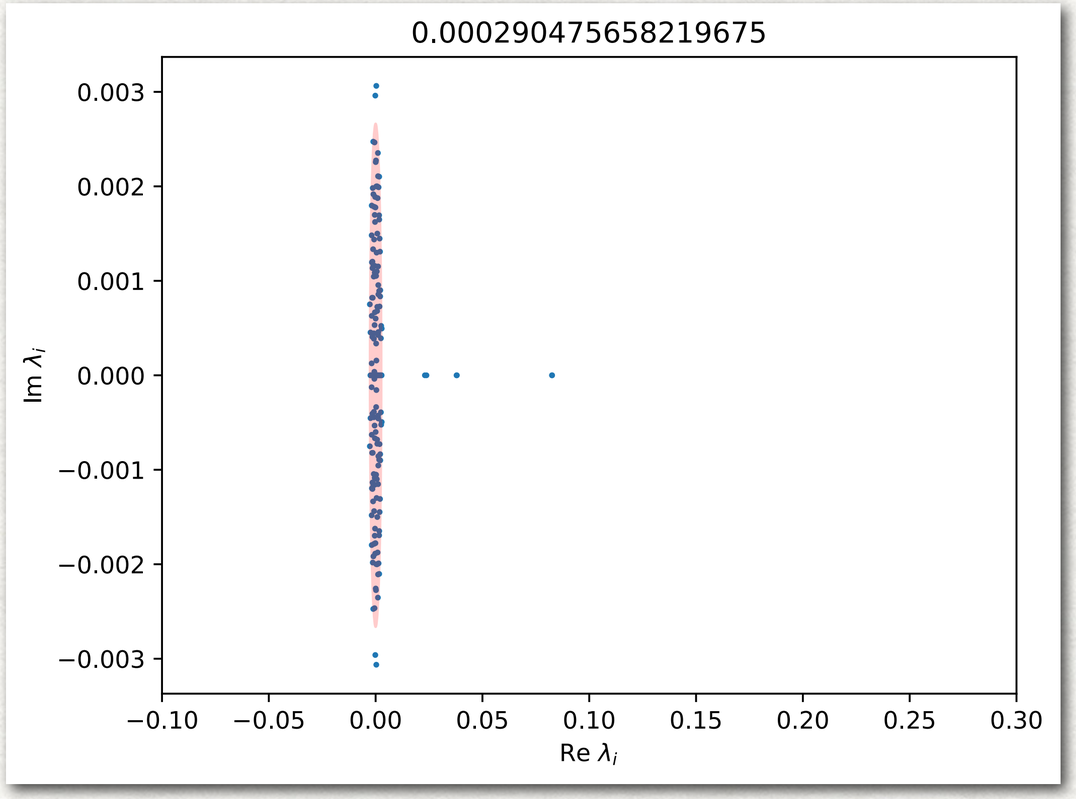
Problem



look_at_data.py

Most of the density matrix is basically zero.

So there ends up being a circle distribution from the random matrix



visualize_circle_distribution.py

Test Setup

Start with coupled cluster RDM.

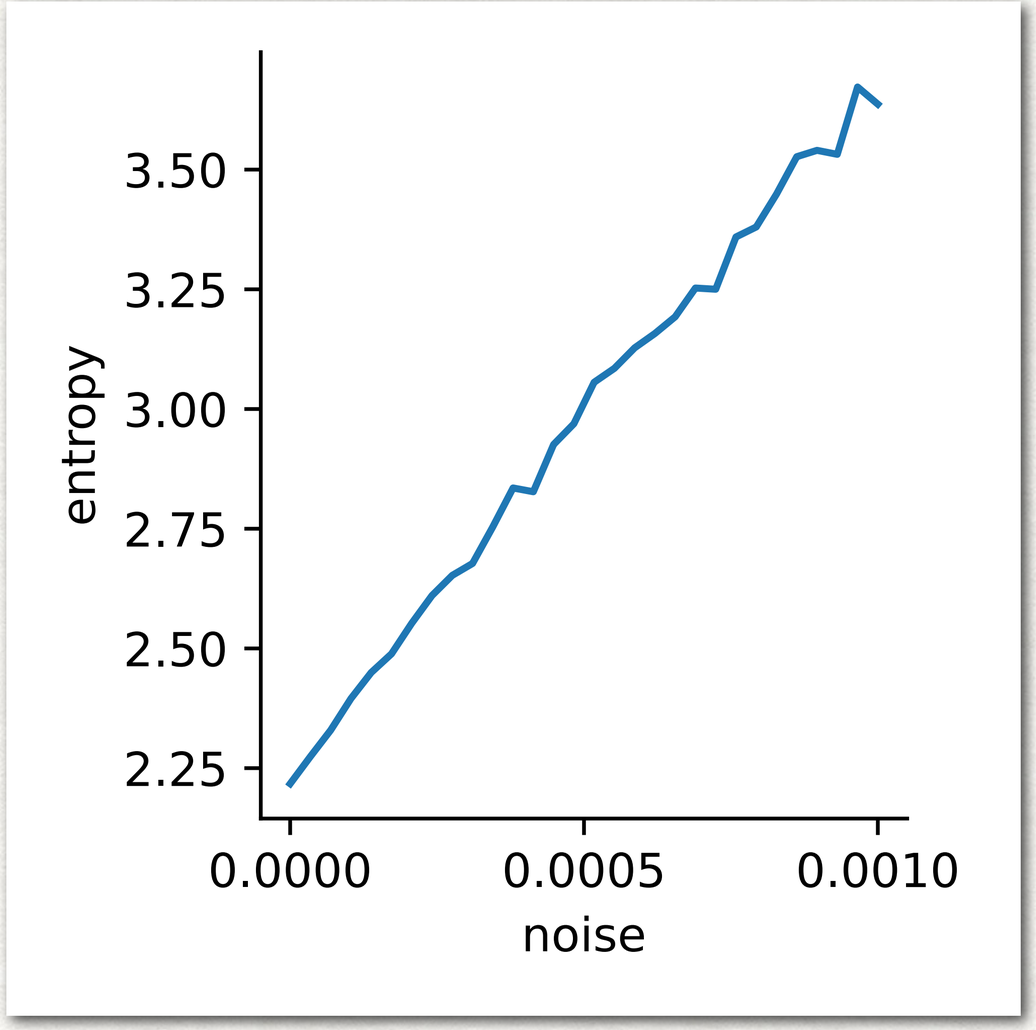
Add $N(0,\sigma)$ to each variable.

Compute the entropy as a function of σ.

Baseline method:
Diagonalize the matrix.
Take real part of eigenvalues.
Sum $-\sum_i \lambda_i \ln \lambda_i$ over eigenvalues greater than 0.

σ goes up to 0.001 since that's a bit above our error bars.

Entropy is significantly biased with even a little noise.



Strategies:

baseline: diagonalize then remove non-positive eigenvalues

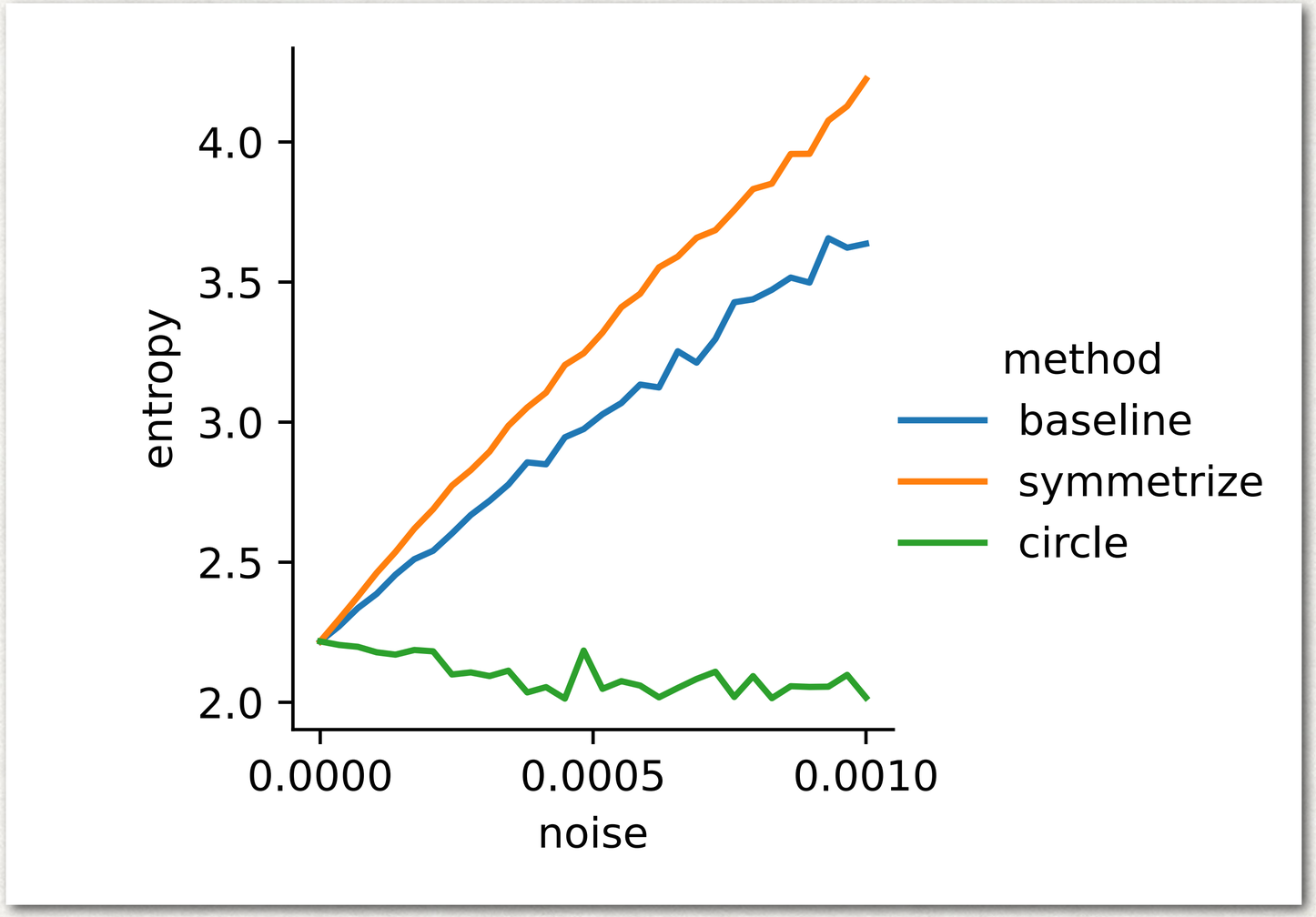
symmetrize: (A + A.T)/2.0 then diagonalize.

circle: (compute_entropy_aggressive)

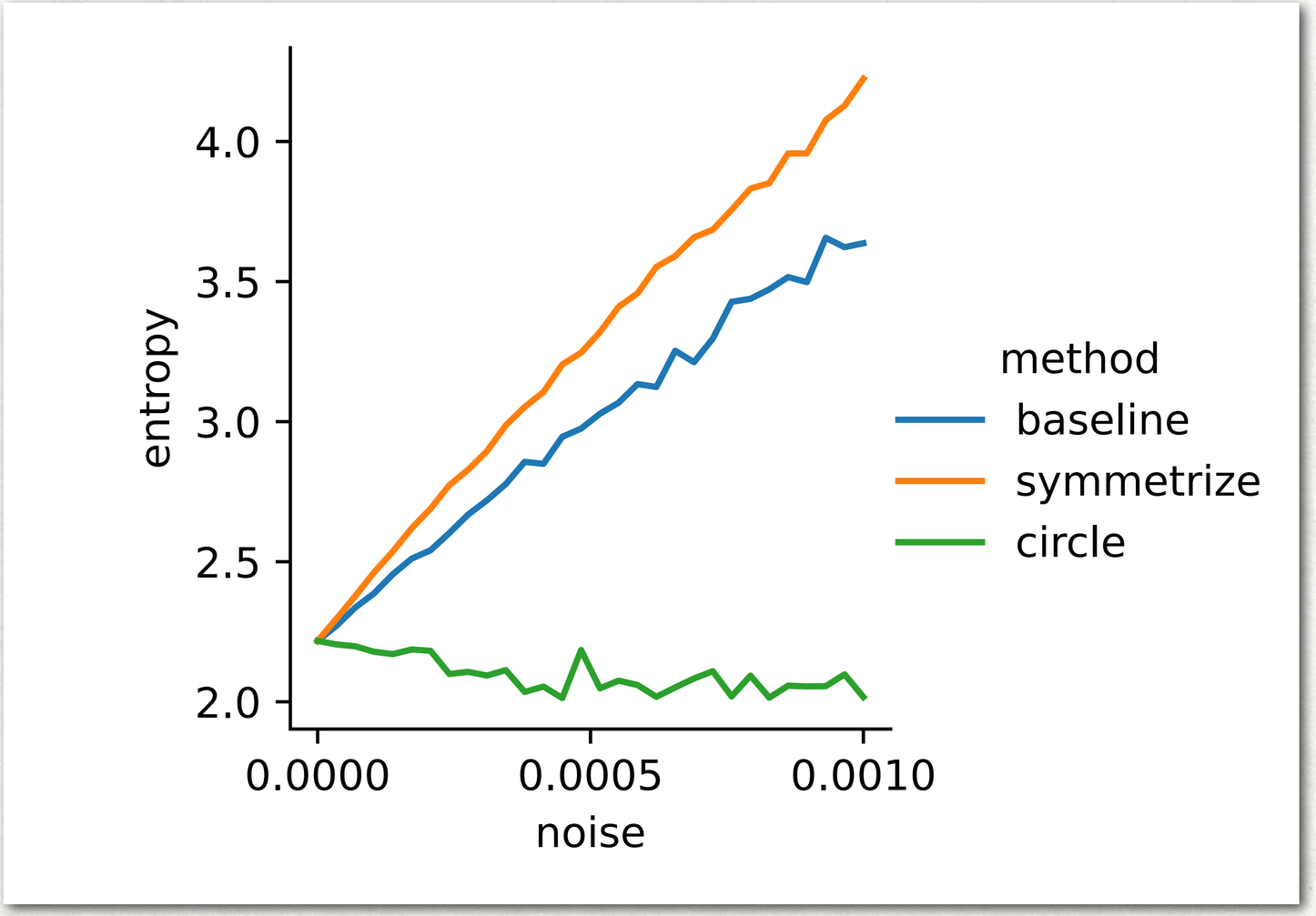
1) Compute circle as $r = \sigma\sqrt{N}$
2) Diagonalize
3) Remove all $|\lambda_i| < r$ from the sum
4) Take real part and only eigenvalues greater than zero.

Implementations in:
extrapolation_methods2.py

plot in:
test_cc_entropy.py



Conclusion



Circle method is much better!

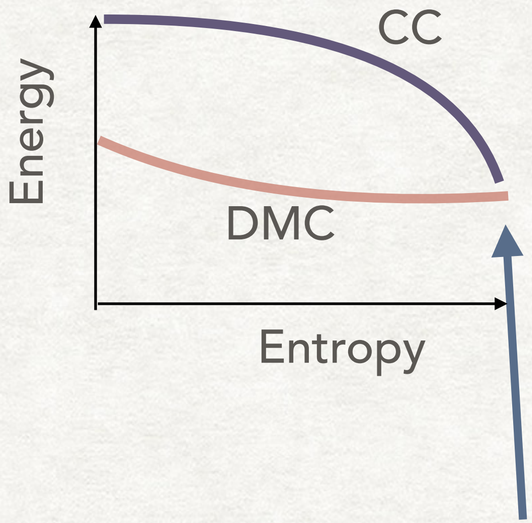
We will use that in the future.

Next Time

Key final plots:

1) Energy vs entropy as a function of basis for CCSD and # determinants for DMC (checks convergence)
2) Entropy versus system size (checks size consistency.)

Attempt to make these plots using the new method. Here's how I think they will look.



Will check to see that CC and DMC end up at the same place