Entanglement in the quantum Heisenberg antiferromagnet

(Sethna, "Entropy, Order Parameters, and Complexity", ex. XXX)

Here we introduce the quantum Heisenberg antiferromagnet, and use it to explore how entropy, temperature, and equilibration can emerge through the entanglement of two portions of a large system -- closely related to the eigenstate thermalization hypothesis. We saw in Entanglement of two spins that ignoring part of a system can take a quantum pure state into a mixture of states on the remaining subsystem; this should remind you of our derivation of the canonical ensemble from a microcanonical system divided into subsystem and bath (Section 6.1, Fig. 6.1). This analogy becomes much more powerful with a larger system, a one-dimensional chain of spin 1/2 particles.

The one-dimensional Heisenberg antiferromagnet has Hamiltonian

$$H = \sum S_m \cdot S_{m+1}$$

where we have set the strength of the coupling J to 1 -- positive, and hence favoring antiparallel spins. Here the quantum spins S=(σX,σY,σZ) have spin 1/2, and are written in terms of the Pauli matrices

σx=((0 1),(1 0)) σy=((0 i), (–i 0)) σz=((1 0),) (0–1))

Let us begin with an analytical calculation of the Hamiltonian and the eigenstates for Nspins=2, considered already in Entanglement of two spins. We work in the four-dimensional σz basis

{|↑1⟩⟨↑2||↑1⟩⟨↓2||↓1⟩⟨↑2||↓1⟩⟨↓2|}

(a) Show analytically that

$$H2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Find the eigenvalues and eigenstates for this Hamiltonian. Is the ground state the triplet or the singlet? Does this make sense for an antiferromagnet? (Hint: The spin S1 commutes with the kets |↑2⟩ and |↓2⟩ and vice-versa.)

Implementing this calculation elegantly on the computer demands that we understand how the single-spin σ operators and the dot product Sm·Sm+1 act on the entire 2^{Nspins}-dimensional Hilbert space. The fact that they commute with the parts of the wavefunction involving other spins says that they act as identity matrices I(M) on those parts of the Hilbert space. That is, σx[1] for the first spin needs to be promoted to σx[1]$\otimes I\left(2^{\text{Nspins}-1}\right)$, and σx[2] for the second needs to be turned into I(2) $\otimes$ σx[1] $\otimes$ $I\left(2^{\text{Nspins}-2}\right)$, \dots

(b) Implement this numerically for the two-spin system. Calculate the Heisenberg Hamiltonian, and

verify the answer of part (a). (Hint: Many modern programming languages have support for tensor data structures. These efficient routines will be important in later steps, so use them here.)

```
σX = ...;
σY = {{0, -I}, {I, 0}};
σZ = ...;

tensorProduct[M1_, M2_] := ArrayFlatten[TensorProduct[M1, M2]]
σx[1] = tensorProduct[σX, IdentityMatrix[2]];
σx[2] = ...;
σy[1] = ...;

Heisenberg2 = σx[1].σx[2] + ...
MatrixForm[Heisenberg2]
```

In this exercise, we shall discuss how pure energy eigenstates states in a system AB become mixed states when we split the system into a subsystem A and a bath B, and study the properties of these mixed states. We shall index operators acting on the subsystem A with Latin letters i, j, operators on the bath B with Greek letters α, β, and operators on the total system AB with capital letters I, J, or sometimes with pairs of indices iα, jβ.

(c) If $\rho_{i\alpha,j\beta}$ is the density matrix for the whole system AB, show analytically that the sum $\rho_{i\alpha,j\alpha}$ gives the reduced density matrix for the subsystem (e.g., as defined in 'Entanglement of two spins').

We can use the two-spin problem of part (a) to preview the rest of the exercise, in a context where you know the answer from 'Entanglement of two spins'. Here we view the first spin as the the 'subsystem' A, and the second spin as the 'bath' B.

(d) Select the singlet eigenstate, and normalize it if necessary. Generate the pure-state density matrix, and reshape it into the four index tensor $\rho_{i\alpha,j\beta}$. Trace over the bath as in part (c), and verify that the reduced density matrix ρA_{ij} describes an unpolarized spin. Calculate the entropy by taking the suitable matrix trace.

```
{vals, vecs} = Eigensystem[Heisenberg2]
ψsinglet = vecs[[...]] / Sqrt[...]

ρPure2 = tensorProduct[ψsinglet, Conjugate[ψsinglet]];
MatrixForm[ρPure2]
ρAB2FourIndex = ArrayReshape[ρPure2, {2, 2, 2, 2}];
MatrixForm[ρAB2FourIndex]
ρA = TensorContract[ρAB2FourIndex, {2, 4}];
MatrixForm[ρA]
-Tr[ρA MatrixLog[...]]
Log[2.]
```

To generate the Heisenberg Hamiltonian for multiple spins, we can save steps by noting that we already know the Hamiltonian for two spins from part (a). So the term Sm · Sm+1 in our Heisenberg Hamiltonian becomes

$I(2^{m-1}) \otimes \mathrm{H2} \otimes I(2^{\mathrm{Nspins}-(m+1)})$

(e) Use this to write a function that returns the Heisenberg Hamiltonian as a 2Nspins×2Nspins matrix. Check, for Nspins=2 it returns H2 from part (a).
Check for Nspins=3 its eigenvalues are (−4,−4,2,2,2,2,0,0), and for Nspins=4 that its distinct eigenvalues are $\left\{-3-2\sqrt{3},\ -1-2\sqrt{2},\ 3,\ -1+2\sqrt{2},\ -1,\ -3+2\sqrt{3}\right\}$.

```
Heisenberg[nSpins_] :=
 Sum[tensorProduct[tensorProduct[IdentityMatrix[...], Heisenberg2],
   IdentityMatrix[...]], {m, 1, nSpins - 1}]
```

In[•]:=
```
Heisenberg[3] // MatrixForm
```

In[•]:=
```
Eigenvalues[Heisenberg[3]]
Eigenvalues[Heisenberg[4]]
```

It is important for the speed of matrix operations to use real numbers and not exact constants for larger matrices. We can do this by converting Heisenberg2 into a real matrix.

In[•]:=
```
Heisenberg2 = 1.0 Heisenberg2
Eigenvalues[Heisenberg[3]] // Chop
Eigenvalues[Heisenberg[4]]
```

We shall work with a system of Nspins=NAB=10 spins in the chain; we shall primarily study a subsystem with NA=4 spins, so the bath has NB=NAB−NA=6 spins. We shall use an eigenstate ψ of H with NAB spins to calculate the reduced density matrix ρA for NA spins, to investigate the entanglement between A and the bath B, to calculate the entanglement entropy, and to illustrate eigenstate thermalization. For the latter, we want an energy that is lower than average, but not near zero.

(f) Create H for 10 spins. Find its energy eigenvalues and eigenstates, and (if necessary) sort them in increasing order of their energy. Pick the energy eigenstate ψ of the full system that is 1/4 the way from the bottom (the $K = 2^{\mathrm{NAB}-3}$ entry). Calculate the pure density matrix ρpure, reshape it into the four

index tensor $\rho AB_{i\alpha,j\beta}$, and trace over the bath to give the reduced density matrix ρAij. Check that ρA has trace one (as it must), and calculate $Tr[(\rho A)^2]$. Is it is a mixed state?

```
nAB = 10;
HamAB = Heisenberg[nAB];
{EAB, ψAB} = Transpose[Sort[Transpose[Eigensystem[HamAB]]]];
K = ...;
ψK = ψAB[[K]];
EK = EAB[[...]]
nA = 4;
ρPure = tensorProduct[..., Conjugate[ψK]];
ρAB = ArrayReshape[ρPure, {2^nA, 2^(nAB - nA), 2^nA, 2^(nAB - nA)}];
ρA = TensorContract[..., ...];
Tr[...]
Tr[... . ...]
```

The entanglement entropy between A and B for a pure state ψ of AB is the entropy of the reduced density matrix of A.

(g) Calculate the entanglement entropy $S=-Tr\rho A \log\rho A$. Check that it has the same entropy as subsystem B. Write a loop over NA ranging through all values from zero to NAB, and plot S as a function of NA for our particular eigenstate ψ. Where is the entanglement entropy largest? Explain why it goes to zero for the two endpoints.

(Note: ρA will have some zero eigenvalues, which confuses Mathematica's matrix logarithm. We define a function pLogp, which returns zero if p <= 0$ and p log p otherwise, and sum it over the eigenvalues of ρA.)

```
pLogp[p_] := If[p > 0, ..., 0]
S[ρ_] := With[{spectrum = Eigenvalues[ρ]},
  -Sum[..., {i, 1, Length[spectrum]}]]

SA = ...
ρB = TensorContract[..., ...];
SB = ...

ρs = Table[ArrayReshape[ρPure, {...}], {nA, 0, nAB}];
ρAs = Table[TensorContract[ρ, ...], {ρ, ρs}];
Ss = Table[S[...], {ρA, ρAs}] // Chop
ListPlot[Table[{nA, Ss[[nA + 1]]}, {nA, 0, nAB}]]
```

The term 'entanglement' is mutual; A and B are entangled, rather than B has somehow perturbed A. This is not an accident. As you checked numerically, the entanglement entropies of the two subsystems should be the same. (This can be shown using the Schmidt decomposition -- an application of singular value decomposition to density matrices in quantum mechanics).

In statistical mechanics, a large system AB in the microcanonical ensemble at energy E will, when restricted to a relatively small subsystem A, generate an equilibrium thermal ensemble at the corresponding temperature. The eigenstate thermalization hypothesis argues that many quantum systems this to an extreme: for any eigenstate ψ with energy E, the reduced density matrix ρA of the subsystem will converge to a Boltzmann equilibrium thermal ensemble as the system size goes to infinity.

Let us calculate the probability p[k] that our subsystem is in eigenstate ψA[k], p[k]=Tr(|ψA[k]⟩⟨ψA[k]|ρA). We are simulating a rather small system, so fluctuations will be large.

(h) Make a log plot of p[k] vs. EA[k]. Do a nonlinear fit to the predicted form above to find β, and plot the result with your data.

```
HamA = Heisenberg[nA];
{EA, ψA} = Eigensystem[HamA];
ψAKetBras = Table[tensorProduct[ ..., Conjugate[ψ]], {ψ, ψA}];
pψs = Table[Tr[ψKetBra . ...], {ψKetBra, ψAKetBras}];
{A0, β0} = {A, β} /. NonlinearModelFit[Transpose[{EA, pψs}],
    A Exp[-β Ek], {A, β}, Ek, MaxIterations → 50 000]["BestFitParameters"]
Show[{ListLogPlot[Transpose[{EA, pψs}]], LogPlot[
   A0 Exp[-β0 Ek], {Ek, -8, 5}]}]
```

In particular, the reduced density matrix is predicted to be at the temperature of the microcanonical ensemble at the energy E of the original pure state ψ.

(i) Write a function EbarAB(β) returning the average energy of the entire system as a function of β. Take a sampling of eigenstates ψK of the total system, fit pk vs EAk as in part(h), and plot β vs. E along with your prediction β(EbarAB). Do you achieve a rough agreement?

We suggest starting with only a few points, spread over the interval. For example, points starting at K=32 and separated by ΔK=64 will span the range avoiding the endpoints.

```
Clear[ρA, βA, pψs]
ρA[ψ_] := TensorContract[ArrayReshape[tensorProduct[ψ, Conjugate[ψ]],
   {2^nA, 2^(nAB - nA), 2^nA, 2^(nAB - nA)}], {2, 4}]
pψs[ψ_] := Table[Tr[ψAKetBra . ρA[ψ]], {ψAKetBra, ψAKetBras}]
βA[ψ_] :=
 β /. NonlinearModelFit[Transpose[{EA, pψs[ψ]}],
    A Exp[-β Ek], {A, β}, Ek, MaxIterations → 50 000]["BestFitParameters"]
Eβ = Table[{EAB[[K]], βA[ψAB[[K]]]}, {K, 32, 2^nAB, 64}];
data = ListPlot[Eβ]

EbarAB[β_] := Total[EAB Exp[-β EAB]] / Total[ ...]
EbarAB[-0.5]
```

In[•]:=
```
theory = ParametricPlot[{EbarAB[β], β}, {β, -0.8, 0.5}];
Show[{data, theory}]
```