Due date: Thursday, September 12

The Hadamard phase of orthogonal matrices

In this first homework assignment we will get you up to speed on material that will be useful for our class research project. Much of this can be found online but you are encouraged to work out everything from scratch.

1. An $n \times n$ *Hadamard matrix* H has only ± 1 elements and orthogonal rows. Prove that the columns are also orthogonal. *Hint:* Relate the transpose to the inverse and the use the equality of right/left inverses.

2. Write down examples of 1×1 and 2×2 Hadamard matrices. Show that all higher $n \times n$ Hadamard matrices must have n divisible by 4. *Hint:* By multiplying rows by -1, as necessary, you can arrange to have all rows start with $+++$, $++-$, $+-+$, or $+--$. Say there are $n_1, \ldots, n_4$ rows with those starts. Now use orthogonality of the first three columns to find relations among these integers.

3. Go to sequence A206711 of OEIS to get counts of Hadamard matrices up to $n = 32$. By analyzing these numbers, speculate about the *entropy* of Hadamard matrices. You might want to first factor out the symmetry group (flipping the sign of any row or column, or any row/column permutation, gives another Hadamard). Entropy in thermodynamics is extensive, but in this case it's not clear what takes the place of "volume" — is it n or n^2, or something else? There's no "correct" answer to this problem, its aim is just to get you thinking.

4. By rescaling, $H \to H/\sqrt{n}$, we turn H into a standard orthogonal matrix (rotation matrix). This is the normalization we will be working with from now on. The 2×2 case is the "Hadamard gate" of quantum computing. Our focus now shifts from the discrete set of Hadamard matrices to *probability distributions on the continuous group of orthogonal matrices*. We will use the symbol U for elements of the group, and define the Hamiltonian

$$\mathcal{H}_n(U) = -\sqrt{n} \sum_{i=1}^{n} \sum_{j=1}^{n} |U_{ij}|^{\alpha},$$

where $\alpha > 0$ is a parameter. Note that $\mathcal{H}_n(U)$ is a constant (trivial) for $\alpha = 2$. Unless stated otherwise, we will usually take $\alpha = 1$. From a computational perspective this is the least expensive way of having a cusp at zero (incentivizing matrix elements to select a sign). Prove that the ground states of $\mathcal{H}_n$, for $\alpha < 2$, are Hadamard matrices for those n where Hadamard matrices exist.

5. Find a ground state for $\mathcal{H}_3$ with $\alpha = 1$.

6. We will study the Gibbs probability density

$$\exp\left(-\beta\, \mathcal{H}_n(U)\right)$$

and discover there is a low temperature *Hadamard phase*. This density is defined over the entire group and we use the uniform measure on the group. We will use the Metropolis-Hastings method to sample the distribution of orthogonal matrices and study their phase behavior. The part in these computations that dominates the work/time is the elementary Markov-chain update, $U \to U'$. To generate updates we will use *Givens rotations*, applied to pairs of rows or columns, by angle θ. Write a well optimized piece of code in a low-level language that implements Givens rotations. In addition to creating U' efficiently, the code should also be efficient in computing the corresponding change in $\mathcal{H}_n$. Your code does not need to run by itself, that is, you do not need to declare variables and allocate memory. In a future assignment you will see to these details and wrap your "Markov chain engine" in a form that can be called in the language of your choice (including python).