The Hadamard phase of orthogonal matrices: solutions

These and future solutions are very concise but try to fully address all the content in the questions. Use office hours if you need more detail.

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.

   Solution: Since the orthogonal matrices form a group, for any element U we have $UU^{-1} = U^{-1}U = 1$. The statement "U has orthogonal rows" is the matrix equation $UU^T = 1$. This identifies U^T as U^{-1}, and by the first remark we know $U^TU = 1$, that is, the columns are orthogonal.

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.

   Solution:

$$\begin{bmatrix} 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

   For $n \geq 3$, the total row count, and the three orthogonalities of the first three columns, give us the following four equations:

$$\begin{aligned} n_1 + n_2 + n_3 + n_4 &= n \\ n_1 + n_2 - n_3 - n_4 &= 0 \\ n_1 - n_2 + n_3 - n_4 &= 0 \\ n_1 - n_2 - n_3 + n_4 &= 0. \end{aligned}$$

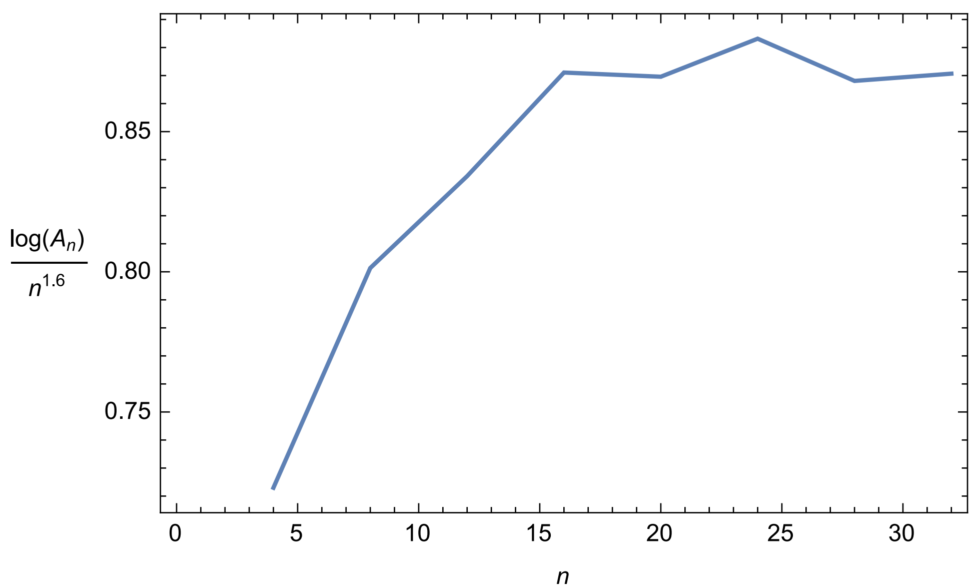
   Add these to get $4n_1 = n$, so n is a multiple of 4.

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.

   "Solution": OEIS A206711 gives the counts for orders $4, \ldots, 32$:

$$\{A_4, A_8, \ldots\} = \{768, 4954521600, \ldots\}.$$

   The best hint of extensive entropy I found was by ignoring the symmetry group and having "volume" scale as $n^{1.6}$:



   Note that $\log|G_n| \sim 2n \log n$, where $|G_n|$ is the order of the symmetry group, would be a subdominant correction.

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.

Solution: This is a property of the geometry of p-norms:

$$\|x\|_p = (|x_1|^p + \cdots + |x_m|^p)^{1/p}.$$

In the positive orthant the constant-norm surfaces are spheres for $p = 2$ that become flat planes at $p = 1$ and concave surfaces for $p < 1$. The smallest constant-p-norm surface, for $p < 2$, that touches the constant 2-norm surface, does so when

$$|x_1| = |x_2| = \cdots = |x_m|. \tag{1}$$

The orthogonal matrices lie on a 2-norm surface in dimension $m = n^2$. By property (1), the p-norm, for $p < 2$, is maximized when all the matrix elements have the same magnitude, that is, when the orthogonal matrix is Hadamard.

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

   Solution: I used a local minimizer in *Mathematica*, and many random starting points, and always obtained

$$\frac{1}{3}\begin{bmatrix} -1 & 2 & 2 \\ 2 & -1 & 2 \\ 2 & 2 & -1 \end{bmatrix}$$

   or a symmetry related matrix. That's not a proof but compelling evidence that this is the unique solution for $n = 3$. A proper proof is probably quite involved.

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).

Solution:

```
void rowrot(int i,int j,double a)
  {
  eold=0.;
  for(k=0;k<n;++k)
    {
    u[k]=h[i][k];
    v[k]=h[j][k];

    eold+=-fabs(u[k])-fabs(v[k]);
    }

  c=cos(a);
  s=sin(a);

  enew=0.;
  for(k=0;k<n;++k)
    {
    h[i][k]=c*u[k]+s*v[k];
    h[j][k]=-s*u[k]+c*v[k];

    enew+=-fabs(h[i][k])-fabs(h[j][k]);
    }

  if(exp(-beta*(enew-eold))>urand())
    {
    energy+=enew-eold;
    accept++;
    }
  else
    for(k=0;k<n;++k)
      {
      h[i][k]=u[k];
      h[j][k]=v[k];
      }
  }
```