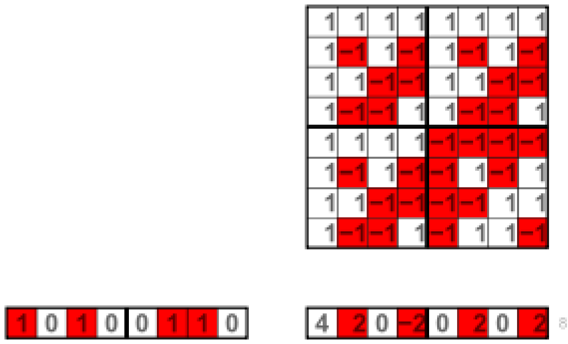
WIKIPEDIA

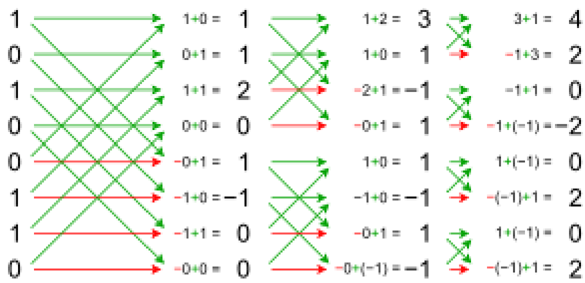
Hadamard transform

The **Hadamard transform** (also known as the **Walsh–Hadamard transform**, **Hadamard–Rademacher–Walsh transform**, **Walsh transform**, or **Walsh–Fourier transform**) is an example of a generalized class of Fourier transforms. It performs an orthogonal, symmetric, involutive, linear operation on 2^m real numbers (or complex numbers, although the Hadamard matrices themselves are purely real).



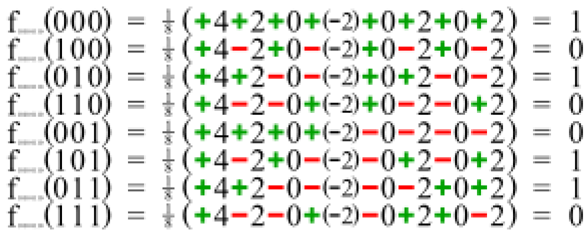
The product of a Boolean function and a Walsh matrix is its Walsh spectrum:[1]
(1,0,1,0,0,1,1,0) × H(8) = (4,2,0,−2,0,2,0,2)

The Hadamard transform can be regarded as being built out of size-2 discrete Fourier transforms (DFTs), and is in fact equivalent to a multidimensional DFT of size $2 \times 2 \times \cdots \times 2 \times 2$.[2] It decomposes an arbitrary input vector into a superposition of Walsh functions.



Fast Walsh–Hadamard transform, a faster way to calculate the Walsh spectrum of (1,0,1,0,0,1,1,0).

The transform is named for the French mathematician Jacques Hadamard, the German-American mathematician Hans Rademacher, and the American mathematician Joseph L. Walsh.



The original function can be expressed by means of its Walsh spectrum as an arithmetical polynomial.

Contents

Definition

The Hadamard transform H_m is a $2^m \times 2^m$ matrix, the Hadamard matrix (scaled by a normalization factor), that transforms 2^m real numbers x_n into 2^m real numbers X_k. The Hadamard transform can be defined in two ways: recursively, or by using the binary (base-2) representation of the indices n and k.

Recursively, we define the 1 × 1 Hadamard transform H_0 by the identity $H_0 = 1$, and then define H_m for $m > 0$ by:

$$H_m = \frac{1}{\sqrt{2}} \begin{pmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{pmatrix}$$

where the $1/\sqrt{2}$ is a normalization that is sometimes omitted.

For $m > 1$, we can also define H_m by:

$$H_m = H_1 \otimes H_{m-1}$$

where $\otimes$ represents the Kronecker product. Thus, other than this normalization factor, the Hadamard matrices are made up entirely of 1 and −1.

Equivalently, we can define the Hadamard matrix by its (k, n)-th entry by writing

$$k = \sum_{i=0}^{m-1} k_i 2^i = k_{m-1} 2^{m-1} + k_{m-2} 2^{m-2} + \cdots + k_1 2 + k_0$$

and

$$n = \sum_{i=0}^{m-1} n_i 2^i = n_{m-1} 2^{m-1} + n_{m-2} 2^{m-2} + \cdots + n_1 2 + n_0$$

where the k_j and n_j are the binary digits (0 or 1) of k and n, respectively. Note that for the element in the top left corner, we define: $k = n = 0$. In this case, we have:

$$(H_m)_{k,n} = \frac{1}{2^{\frac{m}{2}}} (-1)^{\sum_j k_j n_j}$$

This is exactly the multidimensional $2 \times 2 \times \cdots \times 2 \times 2$ DFT, normalized to be unitary, if the inputs and outputs are regarded as multidimensional arrays indexed by the n_j and k_j, respectively.

Some examples of the Hadamard matrices follow.

$$H_0 = +1$$

$$H_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

(This H_1 is precisely the size-2 DFT. It can also be regarded as the Fourier transform on the two-element *additive* group of **Z**/(2).)

$$H_2 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$$H_3 = \frac{1}{2^{\frac{3}{2}}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

$$(H_n)_{i,j} = \frac{1}{2^{n/2}} (-1)^{i \cdot j}$$

where $i \cdot j$ is the bitwise dot product of the binary representations of the numbers i and j. For example, if $n \geq 2$, then $(H_n)_{3,2} = (-1)^{3\cdot 2} = (-1)^{(1,1)\cdot(1,0)} = (-1)^{1+0} = (-1)^1 = -1$, agreeing with the above (ignoring the overall constant). Note that the first row, first column element of the matrix is denoted by $(H_n)_{0,0}$.

The rows of the Hadamard matrices are the Walsh functions.

Computational complexity

The Hadamard transform can be computed in $n \log n$ operations ($n = 2^m$), using the fast Hadamard transform algorithm.

Quantum computing applications

In quantum information processing the Hadamard transformation, more often called **Hadamard gate** in this context (cf. quantum gate), is a one-qubit rotation, mapping the qubit-basis states $|0\rangle$ and $|1\rangle$ to two superposition states with equal weight of the computational basis states $|0\rangle$ and $|1\rangle$. Usually the phases are chosen so that we have

$$H = \frac{|0\rangle + |1\rangle}{\sqrt{2}}\langle 0| + \frac{|0\rangle - |1\rangle}{\sqrt{2}}\langle 1|$$

in Dirac notation. This corresponds to the transformation matrix

$$H_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

in the $|0\rangle, |1\rangle$ basis, also known as the computational basis. The states $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ are known as $|+\rangle$ and $|-\rangle$ respectively, and together constitute the polar basis in quantum computing.

Many quantum algorithms use the Hadamard transform as an initial step, since it maps m qubits initialized with $|0\rangle$ to a superposition of all 2^m orthogonal states in the $|0\rangle, |1\rangle$ basis with equal weight.

Notably, computing the quantum Hadamard transform is simply the application of a Hadamard gate to each qubit individually because of the tensor product structure of the Hadamard transform. This simple result means the quantum Hadamard transform requires *log* n *operations, compared to the classical case of* n *log* n *operations.*

Hadamard gate operations

$$\begin{aligned} H(|0\rangle) &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle =: |+\rangle \\ H(|1\rangle) &= \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle =: |-\rangle \\ H\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) &= \frac{1}{2}\Big(|0\rangle + |1\rangle\Big) + \frac{1}{2}\Big(|0\rangle - |1\rangle\Big) = |0\rangle \\ H\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) &= \frac{1}{2}\Big(|0\rangle + |1\rangle\Big) - \frac{1}{2}\Big(|0\rangle - |1\rangle\Big) = |1\rangle \end{aligned}$$

One application of the Hadamard gate to either a 0 or 1 qubit will produce a quantum state that, if observed, will be a 0 or 1 with equal probability (as seen in the first two operations). This is exactly like flipping a fair coin in the standard probabilistic model of computation. However, if the Hadamard gate is applied twice in succession (as is effectively being done in the last two operations), then the final state is always the same as the initial state. This would be like taking a fair coin that is showing heads, flipping it twice, and it always landing on heads after the second flip.

Other applications

The Hadamard transform is also used in data encryption, as well as many signal processing and data compression algorithms, such as JPEG XR and MPEG-4 AVC. In video compression applications, it is usually used in the form of the sum of absolute transformed differences. It is also a crucial part of Grover's algorithm and Shor's algorithm in quantum computing. The Hadamard transform is also applied in experimental techniques such as NMR, mass spectrometry and crystallography. It is additionally used in some versions of locality-sensitive hashing, to obtain pseudo-random matrix rotations.

See also

- Fast Walsh–Hadamard transform
- Pseudo-Hadamard transform
- Haar transform
- Generalized distributive law

External links

- Ritter, Terry (August 1996). "Walsh-Hadamard Transforms: A Literature Survey" (http://www.ciphersbyritter.com/RES/WALHAD.HTM).
- Akansu, A.N.; Poluri, R. (July 2007). "Walsh-Like Nonlinear Phase Orthogonal Codes for Direct Sequence CDMA Communications" (http://web.njit.edu/~akansu/PAPERS/Akansu-Poluri-WALSH-LIKE2007.pdf) (PDF). *IEEE Transactions on Signal Processing*. **55** (7): 3800–6. doi:10.1109/TSP.2007.894229 (https://doi.org/10.1109%2FTSP.2007.894229).
- Pan, Jeng-shyang Data Encryption Method Using Discrete Fractional Hadamard Transformation (http://www.freepatentsonline.com/y2009/0136023.html) (May 28, 2009)
- Lachowicz, Dr. Pawel. Walsh–Hadamard Transform and Tests for Randomness of Financial Return-Series (http://www.quantatrisk.com/2015/04/07/walsh-hadamard-transform-python-tests-for-randomness-of-financial-return-series/) (April 7, 2015)
- Beddard, Godfrey; Yorke, Briony A. (January 2011). "Pump-probe Spectroscopy using Hadamard Transforms" (http://www1.chem.leeds.ac.uk/People/GSB/Hadamard_for_web.pdf) (PDF).
- Yorke, Briony A.; Beddard, Godfrey; Owen, Robin L.; Pearson, Arwen R. (September 2014). "Time-resolved crystallography using the Hadamard transform" (http://www.nature.com/nmeth/journal/vaop/ncurrent/full/nmeth.3139.html). *Nature Methods*. **11**: 1131–1134. doi:10.1038/nmeth.3139 (https://doi.org/10.1038%2Fnmeth.3139). PMC 4216935 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4216935). PMID 25282611 (https://www.ncbi.nlm.nih.gov/pubmed/25282611).

References

1. Compare Figure 1 in Townsend, W. J.; Thornton, M. A. "Walsh Spectrum Computations Using Cayley Graphs". CiteSeerX 10.1.1.74.8029 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.74.8029).
2. Kunz, H.O. (1979). "On the Equivalence Between One-Dimensional Discrete Walsh-Hadamard and Multidimensional Discrete Fourier Transforms" (http://doi.ieeecomputersociety.org/10.1109/TC.1979.1675334). *IEEE Transactions on Computers*. **28** (3): 267–8. doi:10.1109/TC.1979.1675334 (https://doi.org/10.1109%2FTC.1979.1675334).