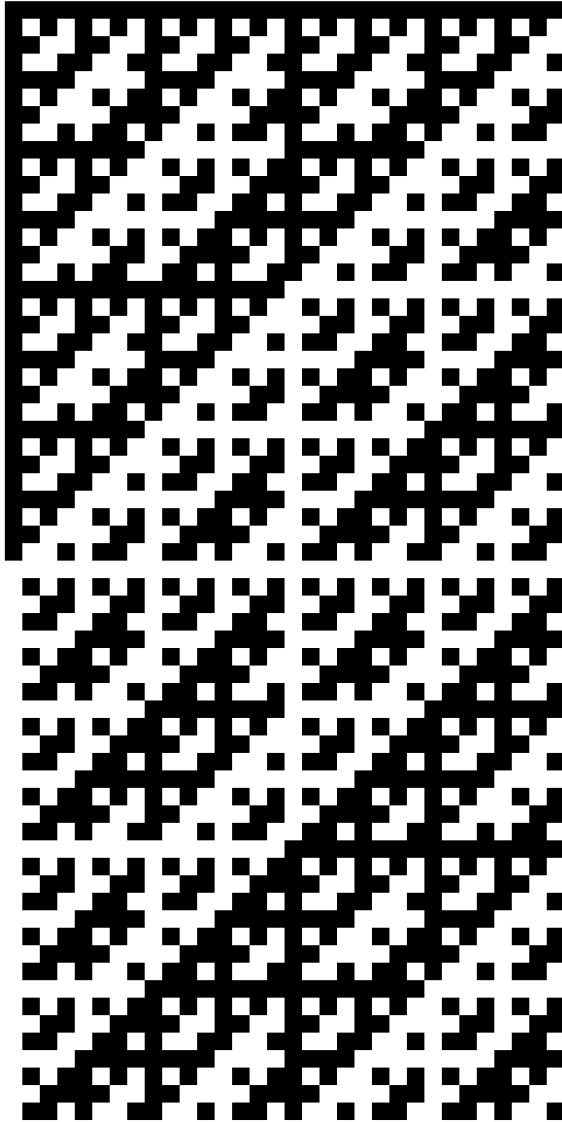


Hadamard code



Matrix of the Punctured Hadamard code (32, 6, 16) for the Reed-Muller code (1, 5) of the NASA space probe Mariner 9

The **Hadamard code** is an error-correcting code that is used for error detection and correction when transmitting messages over very noisy or unreliable channels. In 1971, the code was used to transmit photos of Mars back to Earth from the NASA space probe **Mariner 9**.^[1] Because of its unique mathematical properties, the Hadamard code is not only used by engineers, but also intensely studied in coding theory, mathematics, and theoretical computer science. The Hadamard code is named after the French mathematician **Jacques Hadamard**. It is also known under the names **Walsh code**, **Walsh fam-**

Figure 1 shows a 16x16 grid of 256 cells, each containing a 4-bit Gray code. The grid is organized into four 8x8 blocks. The first block (rows 1-8, columns 1-8) shows the 8-bit Gray code for the first 8 bits. The second block (rows 1-8, columns 9-16) shows the 8-bit Gray code for the next 8 bits. The third block (rows 9-16, columns 1-8) shows the 8-bit Gray code for the next 8 bits. The fourth block (rows 9-16, columns 9-16) shows the 8-bit Gray code for the last 8 bits. The grid is labeled with (A) through (ABCD) on the left and top.

XOR operations
Here the white fields stand for 0
and the red fields for 1

ily,^[2] and **Walsh–Hadamard code**^[3] in recognition of the American mathematician **Joseph Leonard Walsh**.

The Hadamard code is an example of a **linear code** over a **binary alphabet** that maps messages of length k to codewords of length 2^k . It is unique in that each non-zero codeword has a **Hamming weight** of exactly 2^{k-1} , which implies that the **distance** of the code is also 2^{k-1} . In standard **coding theory notation** for **block codes**, the Hadamard code is a $[2^k, k, 2^{k-1}]_2$ -code, that is, it is a **linear code** over a **binary alphabet**, has **block length** 2^k , **message length** (or dimension) k , and **minimum distance** $2^k/2$. The block length is very large compared to the message length, but on the other hand, errors can be corrected even in extremely noisy conditions. The **punctured Hadamard code** is a slightly improved version of the Hadamard code; it is a $[2^k, k+1, 2^{k-1}]_2$ -code and thus has a slightly better **rate** while maintaining the relative distance of $1/2$, and is thus preferred in practical applications. The punctured Hadamard code is the same as the first order **Reed–Muller code** over the binary alphabet.^[4]

Normally, Hadamard codes are based on **Sylvester's construction of Hadamard matrices**, but the term “Hadamard code” is also used to refer to codes constructed from arbitrary **Hadamard matrices**, which are not necessarily of Sylvester type. In general, such a code is not linear. Such codes were first constructed by **R. C. Bose and S. S. Shrikhande** in 1959.^[5] If n is the size of the Hadamard matrix, the code has parameters $(n, 2n, n/2)_2$, meaning it is a not-necessarily-linear binary code with $2n$ code-

words of block length n and minimal distance $n/2$. The construction and decoding scheme described below apply for general n , but the property of linearity and the identification with Reed–Muller codes require that n be a power of 2 and that the Hadamard matrix be equivalent to the matrix constructed by Sylvester’s method.

The Hadamard code is a **locally decodable** code, which provides a way to recover parts of the original message with high probability, while only looking at a small fraction of the received word. This gives rise to applications in **computational complexity theory** and particularly in the design of **probabilistically checkable proofs**. Since the relative distance of the Hadamard code is $1/2$, normally one can only hope to recover from at most a $1/4$ fraction of error. Using **list decoding**, however, it is possible to compute a short list of possible candidate messages as long as fewer than $\frac{1}{2} - \epsilon$ of the bits in the received word have been corrupted.

In **code division multiple access** (CDMA) communication, the Hadamard code is referred to as Walsh Code, and is used to define individual **communication channels**. It is usual in the CDMA literature to refer to codewords as “codes”. Each user will use a different codeword, or “code”, to modulate their signal. Because Walsh codewords are mathematically **orthogonal**, a Walsh-encoded signal appears as **random noise** to a CDMA capable mobile **terminal**, unless that terminal uses the same codeword as the one used to encode the incoming **signal**.^[6]

1 History

Hadamard code is the name that is most commonly used for this code in the literature. However, in modern use these error correcting codes are referred to as Walsh–Hadamard codes.

There is a reason for this:

Jacques Hadamard did not invent the code himself, but he defined **Hadamard matrices** around 1893, long before the first **error-correcting code**, the **Hamming code**, was developed in the 1940s.

The Hadamard code is based on Hadamard matrices, and while there are many different Hadamard matrices that could be used here, normally only **Sylvester’s construction of Hadamard matrices** is used to obtain the codewords of the Hadamard code.

James Joseph Sylvester developed his construction of Hadamard matrices in 1867, which actually predates Hadamard’s work on Hadamard matrices. Hence the name *Hadamard code* is not undisputed and sometimes the code is called *Walsh code*, honoring the American mathematician Joseph Leonard Walsh.

A Hadamard code was used during the 1971 **Mariner 9** mission to correct for picture transmission errors. The data words used during this mission were 6 bits long,

which represented 64 **grayscale** values.

Because of limitations of the quality of the alignment of the transmitter at the time (due to Doppler Tracking Loop issues) the maximum useful data length was about 30 bits. Instead of using a **repetition code**, a [32, 6, 16] Hadamard code was used.

Errors of up to 7 bits per word could be corrected using this scheme. Compared to a **5-repetition code**, the error correcting properties of this Hadamard code are much better, yet its rate is comparable. The efficient decoding algorithm was an important factor in the decision to use this code.

The circuitry used was called the “Green Machine”. It employed the **fast Fourier transform** which can increase the decoding speed by a factor of three. Since the 1990s use of this code by space programs has more or less ceased, and the Deep Space Network does not support this error correction scheme for its dishes that are greater than 26 m.

2 Constructions

While all Hadamard codes are based on Hadamard matrices, the constructions differ in subtle ways for different scientific fields, authors, and uses. Engineers, who use the codes for data transmission, and **coding theorists**, who analyse extremal properties of codes, typically want the **rate** of the code to be as high as possible, even if this means that the construction becomes mathematically slightly less elegant.

On the other hand, for many applications of Hadamard codes in **theoretical computer science** it is not so important to achieve the optimal rate, and hence simpler constructions of Hadamard codes are preferred since they can be analyzed more elegantly.

2.1 Construction using inner products

When given a binary message $x \in \{0, 1\}^k$ of length k , the Hadamard code encodes the message into a codeword $\text{Had}(x)$ using an encoding function $\text{Had} : \{0, 1\}^k \rightarrow \{0, 1\}^{2^k}$. This function makes use of the **inner product** $\langle x, y \rangle$ of two vectors $x, y \in \{0, 1\}^k$, which is defined as follows:

$$\langle x, y \rangle = \sum_{i=1}^k x_i y_i \bmod 2.$$

Then the Hadamard encoding of x is defined as the sequence of *all* inner products with x :

$$\text{Had}(x) = \left(\langle x, y \rangle \right)_{y \in \{0, 1\}^k}$$

As mentioned above, the *punctured* Hadamard code is used in practice since the Hadamard code itself is somewhat wasteful. This is because, if the first bit of y is zero, $y_1 = 0$, then the inner product contains no information whatsoever about x_1 , and hence, it is impossible to fully decode x from those positions of the codeword alone. On the other hand, when the codeword is restricted to the positions where $y_1 = 1$, it is still possible to fully decode x . Hence it makes sense to restrict the Hadamard code to these positions, which gives rise to the *punctured* Hadamard encoding of x ; that is, $\text{pHad}(x) = (\langle x, y \rangle)_{y \in \{1\} \times \{0,1\}^{k-1}}$.

2.2 Construction using a generator matrix

The Hadamard code is a linear code, and all linear codes can be generated by a generator matrix G . This is a matrix such that $\text{Had}(x) = x \cdot G$ holds for all $x \in \{0,1\}^k$, where the message x is viewed as a row vector and the vector-matrix product is understood in the *vector space* over the *finite field* \mathbb{F}_2 . In particular, an equivalent way to write the inner product definition for the Hadamard code arises by using the generator matrix whose columns consist of *all* strings y of length k , that is,

$$G = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ y_1 & y_2 & \dots & y_{2^k} \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}.$$

where $y_i \in \{0,1\}^k$ is the i -th binary vector in *lexicographical order*. For example, the generator matrix for the Hadamard code of dimension $k = 3$ is:

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

The matrix G is a $(k \times 2^k)$ -matrix and gives rise to the *linear operator* $\text{Had} : \{0,1\}^k \rightarrow \{0,1\}^{2^k}$.

The generator matrix of the *punctured* Hadamard code is obtained by restricting the matrix G to the columns whose first entry is one. For example, the generator matrix for the punctured Hadamard code of dimension $k = 3$ is:

$$G' = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

Then $\text{pHad} : \{0,1\}^k \rightarrow \{0,1\}^{2^{k-1}}$ is a linear mapping with $\text{pHad}(x) = x \cdot G'$.

For general k , the generator matrix of the punctured Hadamard code is a *parity-check matrix* for the *extended Hamming code* of length 2^{k-1} and dimension $2^{k-1} - k$

, which makes the punctured Hadamard code the *dual code* of the extended Hamming code. Hence an alternative way to define the Hadamard code is in terms of its parity-check matrix: the parity-check matrix of the Hadamard code is equal to the generator matrix of the Hamming code.

2.3 Construction using general Hadamard matrices

Generalized Hadamard codes are obtained from an n -by- n *Hadamard matrix* H . In particular, the $2n$ codewords of the code are the rows of H and the rows of $-H$. To obtain a code over the alphabet $\{0,1\}$, the mapping $-1 \mapsto 1, 1 \mapsto 0$, or, equivalently, $x \mapsto (1-x)/2$, is applied to the matrix elements. That the minimum distance of the code is $n/2$ follows from the defining property of Hadamard matrices, namely that their rows are mutually orthogonal. This implies that two distinct rows of a Hadamard matrix differ in exactly $n/2$ positions, and, since negation of a row does not affect orthogonality, that any row of H differs from any row of $-H$ in $n/2$ positions as well, except when the rows correspond, in which case they differ in n positions.

To get the punctured Hadamard code above with $n = 2^{k-1}$, the chosen Hadamard matrix H has to be of Sylvester type, which gives rise to a message length of $\log_2(2n) = k$.

3 Distance

The distance of a code is the minimum *Hamming distance* between any two distinct codewords, i.e., the minimum number of positions at which two distinct codewords differ. Since the Walsh–Hadamard code is a *linear code*, the distance is equal to the minimum *Hamming weight* among all of its non-zero codewords. All non-zero codewords of the Walsh–Hadamard code have a *Hamming weight* of exactly 2^{k-1} by the following argument.

Let $x \in \{0,1\}^k$ be a non-zero message. Then the following value is exactly equal to the fraction of positions in the codeword that are equal to one:

$$\Pr_{y \in \{0,1\}^k} [(\text{Had}(x))_y = 1] = \Pr_{y \in \{0,1\}^k} [\langle x, y \rangle = 1].$$

The fact that the latter value is exactly $1/2$ is called the *random subsum principle*. To see that it is true, assume without loss of generality that $x_1 = 1$. Then, when conditioned on the values of y_2, \dots, y_k , the event is equivalent to $y_1 \cdot x_1 = b$ for some $b \in \{0,1\}$ depending on x_2, \dots, x_k and y_2, \dots, y_k . The probability that $y_1 = b$ happens is exactly $1/2$. Thus, in fact, *all* non-zero

codewords of the Hadamard code have relative Hamming weight $1/2$, and thus, its relative distance is $1/2$.

The relative distance of the *punctured* Hadamard code is $1/2$ as well, but it no longer has the property that every non-zero codeword has weight exactly $1/2$ since the all 1s vector 1^{2^k-1} is a codeword of the punctured Hadamard code. This is because the vector $x = 10^{k-1}$ encodes to $\text{pHad}(10^{k-1}) = 1^{2^k-1}$. Furthermore, whenever x is non-zero and not the vector 10^{k-1} , the random subsum principle applies again, and the relative weight of $\text{Had}(x)$ is exactly $1/2$.

4 Local decodability

A **locally decodable** code is a code that allows a single bit of the original message to be recovered with high probability by only looking at a small portion of the received word.

A code is q -query **locally decodable** if a message bit, x_i , can be recovered by checking q bits of the received word. More formally, a code, $C : \{0, 1\}^k \rightarrow \{0, 1\}^n$, is $(q, \delta \geq 0, \epsilon \geq 0)$ -locally decodable, if there exists a probabilistic decoder, $D : \{0, 1\}^n \rightarrow \{0, 1\}^k$, such that (Note: $\Delta(x, y)$ represents the *Hamming distance* between vectors x and y):

$\forall x \in \{0, 1\}^k, \forall y \in \{0, 1\}^n, \Delta(y, C(x)) \leq \delta n$ implies that $\Pr[D(y)_i = x_i] \geq \frac{1}{2} + \epsilon, \forall i \in [k]$

Theorem 1: The Walsh–Hadamard code is $(2, \delta, \frac{1}{2} - 2\delta)$ -locally decodable for $0 \leq \delta \leq \frac{1}{4}$.

Lemma 1: For all codewords, c in a Walsh–Hadamard code, C , $c_i + c_j = c_{i+j}$, where c_i, c_j represent the bits in c in positions i and j respectively, and c_{i+j} represents the bit at position $(i + j)$.

4.1 Proof of lemma 1

Let $C(x) = c = (c_0, \dots, c_{2^n-1})$ be the codeword in C corresponding to message x .

Let $G = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ g_0 & g_1 & \dots & g_{2^n-1} \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}$ be the generator matrix of C .

By definition, $c_i = x \cdot g_i$. From this, $c_i + c_j = x \cdot g_i + x \cdot g_j = x \cdot (g_i + g_j)$. By the construction of G , $g_i + g_j = g_{i+j}$. Therefore, by substitution, $c_i + c_j = x \cdot g_{i+j} = c_{i+j}$.

4.2 Proof of theorem 1

To prove theorem 1 we will construct a decoding algorithm and prove its correctness.

4.2.1 Algorithm

Input: Received word $y = (y_0, \dots, y_{2^n-1})$

For each $i \in \{1, \dots, n\}$:

1. Pick $j \in \{0, \dots, 2^n - 1\}$ uniformly at random
2. Pick $k \in \{0, \dots, 2^n - 1\}$ such that $j + k = e_i$ where $j + k$ is the bitwise *xor* of j and k .
3. $x_i \leftarrow y_j + y_k$

Output: Message $x = (x_1, \dots, x_n)$

4.2.2 Proof of correctness

For any message, x , and received word y such that y differs from $c = C(x)$ on at most δ fraction of bits, x_i can be decoded with probability at least $\frac{1}{2} + (1 - 2\delta)$.

By lemma 1, $c_j + c_k = c_{j+k} = x \cdot g_{j+k} = x \cdot e_i = x_i$. Since j and k are picked uniformly, the probability that $y_j \neq c_j$ is at most δ . Similarly, the probability that $y_k \neq c_k$ is at most δ . By the **union bound**, the probability that either y_j or y_k do not match the corresponding bits in c is at most 2δ . If both y_j and y_k correspond to c , then lemma 1 will apply, and therefore, the proper value of x_i will be computed. Therefore, the probability x_i is decoded properly is at least $1 - 2\delta$. Therefore, $\epsilon = \frac{1}{2} - 2\delta$ and for ϵ to be positive, $0 \leq \delta \leq \frac{1}{4}$.

Therefore, the Walsh–Hadamard code is $(2, \delta, \frac{1}{2} - 2\delta)$ locally decodable for $0 \leq \delta \leq \frac{1}{4}$.

5 Optimality


For $k \leq 7$ the linear Hadamard codes have been proven optimal in the sense of minimum distance.^[7]

6 See also

- **Zadoff–Chu sequence** — improve over the Walsh–Hadamard codes

7 Notes

- [1] <http://www.mcs.csueastbay.edu/~{malek/TeX/Hadamard.pdf>
- [2] See, e.g., Amadei, Manzoli & Merani (2002)
- [3] See, e.g., Arora & Barak (2009, Section 19.2.2).
- [4] See, e.g., Guruswami (2009, p. 3).

- [5] Bose, R.C.; Shrikhande, S.S. (1959). “A note on a result in the theory of code construction”. *Information and Control*. **2** (2): 183–194. CiteSeerX 10.1.1.154.2879 . doi:10.1016/S0019-9958(59)90376-6.
- [6] “CDMA Tutorial: Intuitive Guide to Principles of Communications” (PDF). Complex to Real. Retrieved 4 August 2011.
- [7] Jaffe, David B.; Bouyukliev, Iliya, *Optimal binary linear codes of dimension at most seven*

8 References

- Amadei, M.; Manzoli, U.; Merani, M.L. (2002), “On the assignment of Walsh and quasi-orthogonal codes in a multicarrier DS-CDMA system with multiple classes of users”, *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*, **1**, IEEE, pp. 841–5, doi:10.1109/GLOCOM.2002.1188196, ISBN 0-7803-7632-3
- Arora, Sanjeev; Barak, Boaz (2009), *Computational Complexity: A Modern Approach*, Cambridge University Press, ISBN 978-0-521-42426-4
- Guruswami, Venkatesan (2009), *List decoding of binary codes* (PDF)
- Rudra, Atri, “Hamming code and Hamming bound” (PDF), *Lecture notes*

9 Text and image sources, contributors, and licenses

9.1 Text

- **Hadamard code** *Source:* https://en.wikipedia.org/wiki/Hadamard_code?oldid=749301530 *Contributors:* Awaterl, Michael Hardy, Phil Boswell, Sander123, Tea2min, Giftlite, Will Orrick, Eyreland, Grubber, RDBrown, Oli Filth, Ylloh, Chrisahn, AlaiBot, Deflective, Horacelamb, Infrangible, Hpfister, R'n'B, CommonsDelinker, Marcosaedro, EverGreg, Amanda Breckenridge, Mizst, PipepBot, Watchduck, Addbot, Lightbot, Yobot, AnomieBOT, Andrewrp, LilHelpa, J04n, Cpflieger, HRoestBot, RjwilmsiBot, Chris.c.keller, Cleo, BG19bot, Pintoch, Stormmilk, Comp.arch, OccultZone, Frank Klemm, TheSawTooth, Mahmoudmnsor and Anonymous: 22

9.2 Images

- **File:Hadamard-Code.svg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/3/37/Hadamard-Code.svg> *License:* Public domain *Contributors:* Own work *Original artist:* ` Watchduck (a.k.a. Tilman Piesk)`
- **File:Lock-green.svg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/6/65/Lock-green.svg> *License:* CC0 *Contributors:* en:File:Free-to-read_lock_75.svg *Original artist:* User:Trappist the monk
- **File:Multigrade_operator_XOR.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/1e/Multigrade_operator_XOR.svg *License:* Public domain *Contributors:* Own work *Original artist:* ` Watchduck (a.k.a. Tilman Piesk)`

9.3 Content license

- Creative Commons Attribution-Share Alike 3.0