INFORMATION THEORY AND APPLICATIONS IN COMPUTER SCIENCE

Lecture 6 | 19 October 2026

Student notes

6 Coding bounds and a catalogue of constructions

Random coding gives short encodings whose codewords are far apart. Counting arguments limit how short such encodings can be. Explicit codes then expose different ways to realize distance: parity checks, polynomial evaluation, and composition. Our main algebraic construction is Reed–Solomon; its large alphabet motivates concatenation and Justesen's binary construction.

Conventions and prerequisites. Logarithms are base two unless a base is displayed, and $h_2(p) = -p\log p - (1-p)\log(1-p)$ with continuous endpoint values. We use finite fields, rank–nullity, polynomial division over a field, the union bound, and elementary entropy. A finite-field representation and its arithmetic are assumed available. All codes and decoders are fixed and public during communication. Random choices in an existence proof are design choices.

6.1 The model: distance against arbitrary corruptions

A q-ary code is a subset $C \subseteq \Sigma^n$, where $|\Sigma| = q \geq 2$. For $|C| = M \geq 2$, define

$$d_H(x,y) = |\{i : x_i \neq y_i\}|, \qquad d(C) = \min_{c \neq c'} d_H(c,c'), \qquad R = \frac{\log_q M}{n}, \quad \delta = \frac{d(C)}{n}.$$

The rate counts q-ary message symbols per transmitted q-ary symbol. The bit rate is $R\log q$.

Linear codes. Let q be a prime power, so that a field $\mathbb{F}_q$ of size q exists. A code $C \subseteq \mathbb{F}_q^n$ is *linear* if it contains zero and is closed under linear combinations:

$$ax + by \in C \qquad \text{for all } x, y \in C \text{ and } a, b \in \mathbb{F}_q.$$

Equivalently, C is a vector subspace of $\mathbb{F}_q^n$. If $\dim C = k$ and $d(C) = d$, its parameters are written $[n,k,d]_q$. A choice of k basis vectors identifies messages in $\mathbb{F}_q^k$ with codewords, so $|C| = q^k$ and $R = k/n$. Putting those basis vectors in the rows of a $k \times n$ matrix G gives a *generator matrix*: a row-vector message u is encoded as uG.

The *Hamming weight* is $\|x\|_0 = |\{i : x_i \neq 0\}|$. For linear codes, $d_H(c,c') = \|c - c'\|_0$ and differences are codewords, so

$$d(C) = \min_{0 \neq c \in C} \|c\|_0.$$

The adversary knows the code and the transmitted word and may change any allowed coordinates. The guarantee must hold for every word and every permitted corruption pattern. This differs from averaging over the independent noise of a specified memoryless channel.

Proposition 6.1 (What distance guarantees)**.** *Let t, s be nonnegative integers. A code of minimum distance d detects up to $d-1$ substitutions. It uniquely recovers from every pattern of at most t substitutions and at most s marked erasures if and only if $2t + s < d$. In particular, it corrects $\lfloor (d-1)/2 \rfloor$ arbitrary substitutions or $d-1$ erasures.*

Proof. Detection. Suppose $c \in C$ is transmitted and a nonzero number of substitutions produces $y \in \Sigma^n$ with $1 \leq d_H(c,y) \leq d-1$. If y were a codeword, it would be a distinct codeword closer

than d to c, contradicting minimum distance. Thus testing whether $y \in C$ detects the corruption. This is a correctness argument; it does not give an efficient membership test.

Recovery when $2t + s < d$. Erased positions carry a special symbol $\perp \notin \Sigma$, so the receiver knows their set $E = \{i : y_i = \perp\}$, with $|E| \le s$. A substitution is an incorrect symbol at an unerased position. Consider all codewords consistent with the substitution budget:

$$\mathcal{L}(y, E) = \{c \in C : |\{i \notin E : c_i \neq y_i\}| \le t\}\,.$$

The transmitted word lies in this set. We show that two distinct codewords cannot both lie in it. For $c, c' \in \mathcal{L}(y, E)$, put

$$A = \{i \notin E : c_i \neq y_i\}, \qquad B = \{i \notin E : c'_i \neq y_i\}.$$

Both sets have size at most t. At an unerased position where $c_i \neq c'_i$, the received symbol cannot equal both, so that position belongs to $A \cup B$. Therefore

$$\{i : c_i \neq c'_i\} \subseteq E \cup A \cup B, \qquad d_H(c, c') \le |E| + |A| + |B| \le s + 2t < d.$$

Distinct codewords have distance at least d, so $c = c'$. The candidate set consequently contains exactly the transmitted word. An exhaustive-search decoder that returns this unique candidate always succeeds under the stated budgets. No decoding-time bound is asserted.

Why $2t + s < d$ *is necessary.* Suppose instead that $2t + s \ge d$. Choose codewords $c \neq c'$ at distance exactly d, and let $D = \{i : c_i \neq c'_i\}$. Erase a set $E \subseteq D$ of size $e = \min\{s, d\}$. The number of remaining disagreements is $m = d - e \le 2t$: if $s \ge d$, it is zero; otherwise it is $d - s \le 2t$. Partition $D \setminus E = A \sqcup B$ with $|A|, |B| \le t$. Such a partition exists because $m \le 2t$. Define one received word by

$$y_i = \begin{cases} \perp, & i \in E, \\ c'_i, & i \in A, \\ c_i, & \text{otherwise.} \end{cases}$$

If c was sent, this observation has substitutions precisely on A. If c' was sent, it has substitutions precisely on B. The erased set is the same in both cases, and both explanations obey the budgets. A decoder receives identical data in the two cases and cannot always return both distinct transmitted words. This proves necessity, including pure erasures and pure substitutions.

Finally, setting $s = 0$ gives $2t < d$, equivalently $t \le \lfloor (d-1)/2 \rfloor$; setting $t = 0$ gives $s \le d - 1$. □

6.2 How much redundancy is necessary, and how much suffices?

For binary linear codes the redundancy is $r = n - k$. More generally it is $n - \log_2 M$. We compare lower bounds that every code obeys with an upper bound on the redundancy needed by some code. This convention explains the direction of the *random-code upper bound*: it upper-bounds necessary redundancy, equivalently lower-bounding achievable rate. For background on these bounds, see [1, Chapters 1 and 4].

Let the volume of a Hamming ball be

$$V_q(n, t) = \sum_{j=0}^{t} \binom{n}{j} (q-1)^j, \qquad 0 \le t \le n.$$

The count chooses the changed positions and their replacement symbols.

Theorem 6.2 (Singleton and Hamming lower bounds on redundancy)**.** *Every $(n, M, d)_q$ code satisfies*

$$M \le q^{n-d+1} \quad \textit{(Singleton bound)},$$
$$MV_q(n, \lfloor (d-1)/2 \rfloor) \le q^n \quad \textit{(Hamming bound)}.$$

Consequently, a binary code carrying $k = \log_2 M$ bits and correcting every t substitutions has

$$\boxed{\underbrace{n - k \ge 2t}_{\textit{Singleton}}, \qquad \underbrace{n - k \ge \log V_2(n, t)}_{\textit{Hamming}}.}$$

Proof. Delete any $d-1$ coordinates. The remaining strings identify the codewords injectively, giving the first inequality. For the second, radius-$\lfloor (d-1)/2 \rfloor$ balls around different codewords are disjoint by the triangle inequality, and all lie in Σ^n. A code correcting t arbitrary substitutions must have $d \ge 2t+1$; substitute this into both inequalities. □

The Gilbert–Varshamov (GV) existence bound is due to Gilbert (1952) for general codes and Varshamov (1957) for linear codes [1, §4.6]. We use its standard random-generator proof [1, §4.2.2].

Theorem 6.3 (Gilbert–Varshamov bound: random linear-code form)**.** *For integers $1 \le k \le n$ and $1 \le d \le n$, if*

$$(2^k - 1)\,\frac{V_2(n, d-1)}{2^n} < 1,$$

there exists a binary linear length-n, dimension-k code with distance at least d. In particular, the simpler condition

$$\boxed{n - k > \log V_2(n, 2t)}$$

suffices to correct t arbitrary substitutions, when $2t \le n-1$.

Proof. Choose all entries of a $k \times n$ matrix G independently and uniformly in $\mathbb{F}_2$. For each fixed nonzero message $u \in \mathbb{F}_2^k$, the coordinates of uG are independent fair bits: each column has a fair inner product with u, and the columns are independent. Therefore

$$\Pr_G[\|uG\|_0 \le d-1] = 2^{-n} V_2(n, d-1).$$

Union-bound over the $2^k - 1$ nonzero messages. With positive probability every nonzero message has weight at least d. Since $d \ge 1$, this also excludes a nonzero kernel and proves that the encoder is injective. Fix one successful matrix. The argument supplies existence and a short matrix description, but no efficient test for success or efficient decoder. □

The linear structure matters to this proof: we counted nonzero *differences* through nonzero messages. A direct union bound over all pairs in a completely unstructured random codebook has about M^2 terms and loses a factor of two in the resulting exponent.[1]

Saving randomness with a Toeplitz matrix. The same proof works with a random *Toeplitz matrix* $G_{ij} = a_{j-i}$ (constant along diagonals), using $k+n-1$ independent fair bits $a_{1-k}, \dots, a_{n-1}$ instead of kn. For fixed $u \ne 0$, let i_0 be its first nonzero index. Condition on seeds with index below $1 - i_0$ and expose the outputs $(uG)_1, \dots, (uG)_n$ in order: output j contains the fresh fair seed a_{j-i_0} with coefficient one, plus only lower-indexed seeds. Hence uG is uniform on $\mathbb{F}_2^n$, and the same GV union bound applies; codewords for different messages need not be independent.

[1] Typical random linear codes also explain why GV is a natural benchmark. For a binary random generator matrix, distinct nonzero messages u, v are linearly independent, so uG and vG are independent uniform vectors. If Z counts nonzero messages with $\|uG\|_0 < \delta n$ and $\mu = \mathbb{E}Z$, then $\mathrm{Var}(Z) \le \mu$. For fixed $R > 1 - h_2(\delta)$, the volume estimate below makes μ grow exponentially, and Chebyshev gives $\Pr[Z = 0] \le 1/\mu \to 0$. Thus typical random linear codes cannot beat the GV rate by a fixed positive amount. This is evidence about the random ensemble, not a proof of optimality among all codes.

The entropy form. For an integer $0 \le t \le n/2$, with $\alpha = t/n$, we have[2]

$$\frac{2^{nh_2(\alpha)}}{n+1} \le V_2(n,t) \le 2^{nh_2(\alpha)}. \tag{6.1}$$

Fix $0 < \delta < 1/2$. For every $0 < R < 1 - h_2(\delta)$ and all sufficiently large n, random linear coding gives dimension $\lfloor Rn \rfloor$ and distance at least $\lceil \delta n \rceil$.[3] Conversely, for any binary family with $d_n/n \to \delta$, Hamming and Singleton imply

$$\limsup R_n \le \min\{1 - h_2(\delta/2),\, 1 - \delta\}.$$

Thus, at the same relative distance, the elementary existence and impossibility bounds leave a gap.[4] For an error fraction $0 < \tau < 1/4$, they give the asymptotic comparison

$$\underbrace{R < 1 - h_2(2\tau)}_{\text{sufficient for some code}}, \qquad \underbrace{\limsup R_n \le 1 - h_2(\tau)}_{\text{necessary for every family}}.$$

The doubled radius comes from separating two possible transmitted words. These are adversarial-error bounds; the BSC capacity $1 - h_2(\tau)$ from Lecture 5 addresses a different noise model.

A further obstruction: Plotkin. Plotkin's bound counts how much pairwise separation the n coordinates can supply. In one coordinate, the codewords split into a zero group and a one group; only pairs from opposite groups differ there. Balanced groups maximize this contribution. Summing the contributions limits the minimum distance [4].

Theorem 6.4 (Binary Plotkin bound)**.** *Let $C \subseteq \{0,1\}^n$ have $M \ge 2$ codewords and minimum distance $d > n/2$. Then*

$$\boxed{M \le \left\lfloor \frac{2d}{2d-n} \right\rfloor.} \tag{6.2}$$

No linearity assumption is needed.

Proof. List the codewords as $c^{(1)}, \dots, c^{(M)}$ and count the total distance over *unordered* pairs:

$$S = \sum_{1 \le i < \ell \le M} d_H(c^{(i)}, c^{(\ell)}).$$

[2]For the upper bound, use the entropy-counting method of Lecture 2, Section 2.6. Let X be uniform on the ball about 0, put $p_i = \Pr[X_i = 1]$, and set $\bar{p} = n^{-1}\sum_i p_i = \mathbb{E}\|X\|_0/n \le \alpha$. The chain rule, conditioning-reduces-entropy, concavity of h_2, and its monotonicity on $[0, 1/2]$ give

$$\log V_2(n,t) = H(X) \le \sum_i h_2(p_i) \le n h_2(\bar{p}) \le n h_2(\alpha).$$

For the lower bound when $t > 0$, t is a mode of Binomial$(n, t/n)$: the consecutive-mass ratio $\frac{n-j}{j+1}\frac{t}{n-t}$ is greater than one for $j < t$ and less than one for $j \ge t$. Its mass at t is therefore at least $1/(n+1)$, so $V_2(n,t) \ge \binom{n}{t} \ge 2^{nh_2(\alpha)}/(n+1)$. The case $t = 0$ is immediate.

[3]As of October 2026, the GV curve remains the best known general asymptotic lower bound on binary rate at fixed relative distance. Jiang–Vardy [2] improve the GV code-size bound by a factor of order n for fixed $0 < \delta \le 0.499$; this adds only $O(\log n/n)$ to the rate and does not change its limiting exponent. For larger alphabets the situation differs: AG codes beat the q-ary GV curve for some relative distances over fixed square fields $q \ge 49$ [16, §4.2]. Thus typical random-code behavior alone does not establish optimality.

[4]The Hamming upper bound is known to be loose. If $R^*(\delta)$ denotes the best asymptotic binary rate, the first McEliece–Rodemich–Rumsey–Welch (MRRW) bound gives

$$R^*(\delta) \le h_2\Big(\tfrac{1}{2} - \sqrt{\delta(1-\delta)}\Big)$$

[1, §8.2]. At $\delta = \frac{1}{2} - \varepsilon$, GV gives an achievable rate $\Theta(\varepsilon^2)$, whereas this upper bound is $O(\varepsilon^2 \log(1/\varepsilon))$ as $\varepsilon \downarrow 0$. An August 2026 preprint of Alrabiah–Guruswami [3] reports strict improvements over both MRRW bounds using classical–quantum channels as a proof tool for ordinary binary codes. The binary GV optimality question remains open.

Every pair has distance at least d, and there are $M(M-1)/2$ pairs. Consequently,

$$S \geq \frac{M(M-1)}{2}\, d.$$

Now count the same quantity by coordinates. If coordinate j contains a_j ones among the M codewords, it contains $M - a_j$ zeros. A pair differs there exactly when it takes one word from each group. There are $a_j(M - a_j)$ such unordered pairs, each counted once. Thus

$$S = \sum_{j=1}^{n} a_j(M - a_j) \leq \sum_{j=1}^{n} \frac{M^2}{4} = \frac{nM^2}{4},$$

where $a_j(M - a_j) = M^2/4 - (a_j - M/2)^2 \leq M^2/4$. Combining the two counts and dividing by $M/4 > 0$ gives

$$2d(M-1) \leq nM, \qquad M(2d-n) \leq 2d.$$

The hypothesis $d > n/2$ makes $2d - n$ positive, so division is valid. Since M is an integer, taking the floor proves the bound. □

What the bound says about rate. For any fixed $\varepsilon > 0$, a family with $d/n \geq 1/2 + \varepsilon$ satisfies

$$M \leq \frac{2(d/n)}{2(d/n) - 1} = 1 + \frac{1}{2(d/n) - 1} \leq 1 + \frac{1}{2\varepsilon}.$$

Its number of messages is bounded independently of n, so its binary rate $\log M/n$ tends to zero. Thus a fixed excess of relative distance above $1/2$ forces the rate to vanish.

The boundary $d = n/2$ requires a separate step. The denominator in (6.2) is zero at this boundary, so we cannot substitute into that formula. Suppose n is even and $d = n/2$. Keep the codewords having the more frequent value in a chosen coordinate; there are $M' \geq M/2$ of them. Delete that coordinate. All retained words agreed there, so their pairwise distances are unchanged. Their new length is $n' = n - 1$, and their distance is at least $d = n/2 > n'/2$. Apply the same counting argument with this lower bound d to obtain

$$M' \leq \frac{2d}{2d - n'} = \frac{n}{n - (n-1)} = n, \qquad M \leq 2M' \leq 2n.$$

If $M' = 1$, the last bound is immediate. Hence rate also vanishes for codes with distance exactly half their length: here it is at most $\log(2n)/n$. The Hadamard family below has relative distance $1/2$ and vanishing rate.

6.3 Hamming codes: exact packing and an efficient decoder

Hamming's construction [5] makes the syndrome itself identify the corrupted coordinate. Let $H \in \mathbb{F}_2^{r \times n}$ have full row rank. The code $C = \{c \in \mathbb{F}_2^n : Hc^{\mathsf{T}} = 0\}$, also written $\ker H$, has dimension $n - r$. We call H a *parity-check matrix*: each row gives one linear constraint on a codeword. If $y = c + e$, the syndrome is

$$Hy^{\mathsf{T}} = He^{\mathsf{T}}.$$

It depends only on the error. The minimum distance of $\ker H$ is the least number of dependent columns of H: a nonzero kernel vector is exactly a nontrivial dependence among its support columns.

Theorem 6.5 (The binary Hamming family)**.** *For each integer $r \geq 2$, take as columns of H_r every nonzero vector of $\mathbb{F}_2^r$ exactly once. Then*

$$C_r = \ker H_r \quad \textit{has parameters} \quad [2^r - 1,\, 2^r - 1 - r,\, 3]_2.$$

It corrects any one-bit error by computing the syndrome and, if it is nonzero, flipping the unique position with that column.

Proof. The unit vectors occur among the columns, so H_r has rank r. No column is zero and no two are equal, excluding weights one and two from the kernel. Two distinct columns sum to a third nonzero column, so a weight-three codeword exists. For a single error at position i, the syndrome is precisely column i; no error gives syndrome zero. □

Encoding can also be systematic. Put the unit-vector columns first, so $H_r = [I_r \mid A]$. For a message $u \in \mathbb{F}_2^{n-r}$, transmit

$$c = (Au, u)$$

in column-vector notation, since $Au + Au = 0$ in $\mathbb{F}_2^r$. Syndrome computation uses $O(rn)$ bit operations; a table identifies the corrupted coordinate. Outside the one-error promise a nonzero syndrome still names a column, and the decoder can return the wrong codeword.

These codes attain the Hamming bound exactly:

$$|C_r| V_2(2^r - 1, 1) = 2^{2^r - 1 - r}\, 2^r = 2^{2^r - 1}.$$

Nevertheless, $R \to 1$ while $d/n = 3/(2^r - 1) \to 0$. An *asymptotically good* family over a fixed alphabet instead has unbounded length, rate bounded below by a positive constant, and relative distance bounded below by a positive constant.

Application. Protecting stored or transmitted words against a single-bit corruption: a small parity-check syndrome locates the error directly [5].

Finite geometry of the parity-check columns. A point of the projective space $\mathrm{PG}(r-1, 2)$ is a one-dimensional subspace of $\mathbb{F}_2^r$. Each such subspace has exactly one nonzero vector, so the columns of H_r list all these points. Two distinct columns $h_i = a$ and $h_j = b$ span the two-dimensional vector space $\{0, a, b, a + b\}$. Its three nonzero vectors are the three points of a projective line, and the third is another column, $h_k = a + b$. Consequently

$$H_r(e_i + e_j + e_k)^{\mathsf{T}} = a + b + (a + b) = 0.$$

The three positions are distinct, so this is a weight-three codeword. No column is zero and no two columns coincide, excluding weights one and two. Figure 1 therefore gives a geometric explanation of the minimum distance being exactly three.

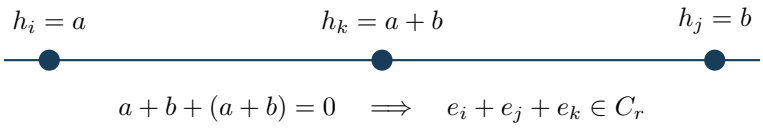


Figure 1: The three points of a projective line over $\mathbb{F}_2$ give a three-column dependence. This is finite projective geometry; the drawing does not represent Euclidean vector addition.

6.4 A catalogue: what each family is designed to do

The parameters below describe distance. Efficient decoding is a separate property. In the binary rows, n denotes the displayed blocklength.

Family	Alphabet / length	Rate	Relative distance
Hamming	2 / $2^r - 1$	$1 - r/n$	$3/n$
Hadamard	2 / 2^m	m/n	$1/2$
Reed–Muller	2 / 2^m	$2^{-m} \sum_{j=0}^{r} \binom{m}{j}$	2^{-r}
Reed–Solomon	q / $n \le q$	k/n	$(n-k+1)/n$
Long code	2 / 2^{2^m}	m/n	$1/2$
LDPC	2 / n	$1 - \operatorname{rank}(H)/n$	Depends on H

Hadamard: all linear measurements. For $a \in \mathbb{F}_2^m$, list $a \cdot x$ at every $x \in \mathbb{F}_2^m$. This is a linear $[2^m, m, 2^{m-1}]_2$ code. A nonzero linear form takes each value equally often: translating x by a vector v with $a \cdot v = 1$ pairs opposite values. The code has excellent relative distance and very small rate. Its local correction and list-decoding connections are the focus of Lecture 7. *Application.* Reconstructing a vector from noisy linear measurements underlies the Goldreich–Levin hard-core predicate and its connection to pseudorandomness [6].

Reed–Muller: all low-degree polynomial measurements. For $0 \le r \le m$, evaluate every multilinear polynomial of total degree at most r on $\mathbb{F}_2^m$. The parameters are

$$\mathrm{RM}(r, m) = \left[2^m, \sum_{j=0}^{r} \binom{m}{j}, 2^{m-r} \right]_2 .$$

The reading section proves both dimension and distance. The degree-one family includes a constant term, giving dimension $m + 1$; the Hadamard convention above uses only homogeneous linear forms and has dimension m. These are related codes with different message spaces. *Application.* Polynomial-evaluation codes support program self-correction and local recovery from a corrupted function table. The polynomial-correction results of Gemmell–Sudan use larger-field Reed–Muller variants [7]; see also [1, Chapters 9 and 13].

Long code: every Boolean measurement. A message is $a \in \{0,1\}^m$. Coordinates are indexed by all Boolean functions $f : \{0,1\}^m \to \{0,1\}$, and the coordinate indexed by f equals $f(a)$. There are 2^{2^m} coordinates and 2^m messages. For distinct a, b, exactly half of all functions have $f(a) \neq f(b)$, proving the relative distance in the table. Its huge length makes it useful as an encoding for local consistency tests. *Application.* PCPs and hardness-of-approximation reductions use the interpretation of a codeword as all Boolean measurements of one hidden label [8].

LDPC: many short parity constraints. A low-density parity-check code is $\ker H$ for a sparse binary matrix H [9], usually with a bounded number of ones per row and column as length grows. Its Tanner graph has a variable node for each coordinate, a check node for each row, and an edge for each nonzero entry. Each check requires the XOR of its neighboring variables to be zero. Sparse checks allow local message-passing procedures.

Peeling on an erasure channel. The receiver knows which bits are erased, and the remaining bits are correct. In each check, move the known bits to the right-hand side:

$$\bigoplus_{i \in U(a)} x_i = s_a, \qquad s_a = \bigoplus_{j \in N(a) \setminus U(a)} x_j,$$

where $N(a)$ is the neighbor set of check a and $U(a)$ consists of its still-erased neighbors. If $U(a) = \{i\}$, the check determines $x_i = s_a$. Recover that bit, remove its variable node and incident edges from the residual graph, and update every neighboring check b by $s_b \leftarrow s_b \oplus x_i$. Repeat. Each recovered value is forced by a parity equation, so induction shows that every peeling step is correct. Figure 2 illustrates the update.

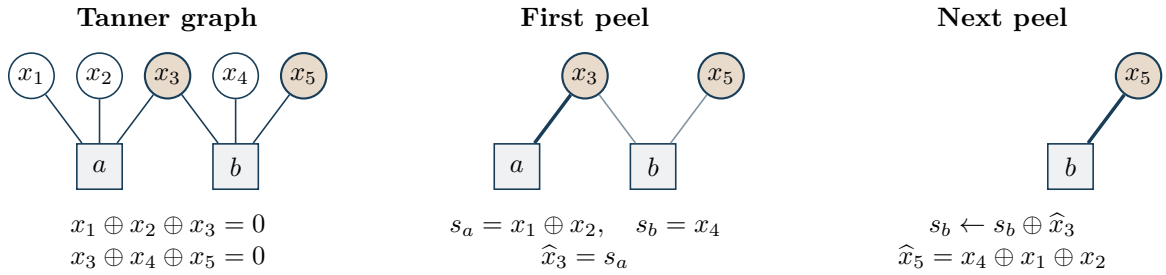


Figure 2: A Tanner graph and two peeling steps. Circles are variables and squares are parity checks; shaded circles mark the erased bits x_3, x_5. After known bits are absorbed, check a has one erased neighbor. Recovering and removing x_3 leaves check b with one erased neighbor.

If erasures remain but each adjacent check has at least two erased neighbors, peeling stalls: the remaining erased set is a *stopping set*. Thus sparsity alone does not establish positive relative distance, a decoding threshold, or capacity. Such conclusions require properties of the chosen graph and an analysis of the chosen decoder. *Application.* Channel coding with many simple, parallelizable message-passing updates: the sparse graph keeps each decoding round inexpensive [9].

More algebraic and graph constructions. The following families extend the catalogue along three useful axes: restrict an extension-field code to a smaller alphabet, change the evaluation space, or impose graph structure on the parity checks. The definitions and parameter statements are retained in the extended catalogue in the further-reading section.

Family	Defining idea	Application / reason to use it
BCH	Consecutive roots in an extension field	Multiple-error correction over a small alphabet [1]
Algebraic-geometric (AG)	Evaluate functions on a curve	Fixed-alphabet coding; secret sharing and distributed storage [16]
Classical binary Goppa	Rational parity checks, restricted to binary words	Hidden algebraic decoding in McEliece-style cryptography [17]
Expander / Tanner	Local constraints on an expanding graph	Provable distance and fast adversarial-error decoding [18]
Folded Reed–Solomon	Bundle consecutive RS symbols	List decoding beyond the unique-decoding regime [19]
Fountain / LT	Generate a variable number of coded packets	Rateless recovery on packet-erasure channels [20]

AG codes are also called *geometric Goppa codes*. The classical binary Goppa family in the table uses extension-field parity checks and binary subfield restriction. Distinguishing these two uses of "Goppa" makes the algebraic catalogue easier to navigate.

6.5 Reed–Solomon: distance from the root bound

The polynomial-evaluation construction is due to Reed and Solomon [10]. Fix a finite field $\mathbb{F}_q$, distinct points $\alpha_1, \dots, \alpha_n \in \mathbb{F}_q$, and integers $1 \le k \le n \le q$. A message consists of the k coefficients of a polynomial f with $\deg f < k$. Encode by

$$\mathrm{RS}(f) = \big(f(\alpha_1), \dots, f(\alpha_n)\big).$$

The governing fact is that a nonzero degree-s polynomial over a field has at most s distinct roots. A root a gives a factor $X - a$ by division with remainder; factoring one root and inducting proves the fact.

Theorem 6.6 (Reed–Solomon parameters)**.** *The evaluation code has parameters $[n, k, n-k+1]_q$ and attains Singleton.*

Proof. Evaluation is linear. A degree-less-than-k polynomial vanishing at all $n \ge k$ points must be zero, so the map is injective and has dimension k. The difference of two distinct messages has at most $k-1$ roots, giving distance at least $n-k+1$. The polynomial $\prod_{j=1}^{k-1}(X - \alpha_j)$ has exactly $k-1$ zeros among the evaluation points and attains this distance against zero. For $k = 1$ the empty product is the nonzero constant one, so the same argument applies. □

Erasure decoding by interpolation. Any k surviving values suffice. At distinct surviving points $\beta_1, \dots, \beta_k$, with values $v_1, \dots, v_k$, recover

$$f(X) = \sum_{i=1}^{k} v_i \prod_{j \ne i} \frac{X - \beta_j}{\beta_i - \beta_j}. \tag{6.3}$$

Each denominator is nonzero. The ith basis polynomial is one at β_i and zero at the other surviving points. The expression has degree less than k and the required values; the root bound proves uniqueness. This corrects $n - k$ erasures.

Unknown errors: encode their locations in a polynomial. Suppose $y_i = f(\alpha_i)$ except at at most t positions and $2t \le n - k$. If the error positions were known, the polynomial $E_0(X) = \prod_{i:\, y_i \ne f(\alpha_i)}(X - \alpha_i)$ would vanish there. The equations $N(\alpha_i) = y_i E(\alpha_i)$ become linear in the coefficients when we introduce $N = fE$. This is the *Berlekamp–Welch decoder*, also called Welch–Berlekamp [11]. We use its interpolation presentation from [12, §3.3] and [1, Algorithm 12.1.1]; the former credits Gemmell–Sudan [7] for the streamlined presentation.

Theorem 6.7 (Berlekamp–Welch unique decoding)**.** *Fix $t \ge 0$ with $2t \le n - k$. Solve for E, N satisfying*

$$E \text{ monic of degree } t, \qquad \deg N \le k + t - 1, \qquad N(\alpha_i) = y_i E(\alpha_i) \quad (1 \le i \le n).$$

If no solution exists, return failure. Otherwise divide N by E; return the quotient only if the remainder is zero, its degree is less than k, and its evaluation disagrees with y in at most t positions. This recovers the message whenever at most t errors occurred, using $O(n^3)$ field operations by Gaussian elimination and elementary polynomial arithmetic.

Proof. If the actual number of errors is $s \le t$, multiply E_0 by any monic degree-$(t-s)$ polynomial to obtain a monic degree-t polynomial E. Then $N = fE$ satisfies every equation: at errors E vanishes, and elsewhere $y_i = f(\alpha_i)$. Thus a solution exists even when $s < t$.

For *any* solution, $N - fE$ has degree at most $k + t - 1$ and vanishes at all $n - s$ correct positions. Since

$$n - s \ge n - t \ge k + t > k + t - 1,$$

it is identically zero. Monicity excludes $E = 0$, so division returns exactly f and validation accepts. Finally, writing $E = X^t + \sum_{j<t} e_j X^j$ and $N = \sum_{j<k+t} n_j X^j$ gives $k + 2t \le n$ unknown coefficients and n affine linear equations. The stated operations solve them, divide, and check the result. □

Applications. Erasure protection for stored or transmitted blocks exploits recovery from any k surviving symbols. Polynomial evaluation and interpolation also supply the algebraic layer of Shamir's threshold secret-sharing scheme [13].

Why binary expansion is insufficient. For $q = 2^m$, represent each field symbol in a fixed $\mathbb{F}_2$ basis. Reed–Solomon then has binary length mn and dimension mk, but one nonzero field-symbol difference may change only one bit. The immediate binary distance guarantee is merely $n - k + 1$, giving relative distance $(n - k + 1)/(mn)$, which tends to zero when m grows. The root-bound argument counts different *symbols*. An inner binary code is needed to turn a different symbol into many different bits.

6.6 Concatenation and the Justesen construction

Concatenation was introduced by Forney [14]. Let an outer code over an alphabet of size 2^m have length N, size 2^{mk}, and distance D. At coordinate i, choose an injective linear inner map $\phi_i : \mathbb{F}_2^m \to \mathbb{F}_2^\ell$. Concatenation replaces each outer symbol by its ℓ-bit inner encoding.

Proposition 6.8 (Product parameters)**.** *The concatenated code has length $N\ell$, size 2^{mk}, and rate*

$$R = \frac{k}{N}\frac{m}{\ell}.$$

If every inner code has minimum distance at least d, its distance is at least Dd and its relative distance is at least $(D/N)(d/\ell)$.

Proof. Inner injectivity preserves the number of messages. Distinct outer words differ in at least D blocks, and each differing block contributes at least d bit differences. Dividing dimension and distance by length gives the formulas. □

Application. Convert a large-alphabet algebraic code into a binary transmission scheme while designing the inner and outer decoders separately [14].

This reduces binary construction to finding short good inner codes. Random coding proves their existence, but exhaustive search through all binary generator matrices of size comparable to $m \times m$ takes $2^{\Theta(m^2)}$ trials. When m is comparable to $\log N$, that enumeration is not polynomial in the final blocklength. Justesen's idea [15] is to use a small explicit ensemble of inner codes and tolerate its bad members. We need most inner maps to be good, rather than certify a single map and reuse it everywhere.

Fix $F = \mathbb{F}_{2^m}$, write $q = 2^m$, and fix an $\mathbb{F}_2$-linear identification $[\,\cdot\,] : F \to \mathbb{F}_2^m$. For each $\alpha \in F$, the Wozencraft map is

$$W_\alpha(x) = ([x], [\alpha x]) \in \mathbb{F}_2^{2m}.$$

Each map is linear and injective, with rate $1/2$. Use a Reed–Solomon outer code evaluated at *all* q points, and at the position indexed by α use W_α. Thus encode

$$\boxed{f \longmapsto \big([f(\alpha)], [\alpha f(\alpha)]\big)_{\alpha \in F}, \qquad \deg f < k.} \tag{6.4}$$

This all-field version includes the possibly bad map at $\alpha = 0$; the proof counts it among the exceptions. The usual nonzero-point version has length $2m(q - 1)$ and the same asymptotic tradeoff.

Theorem 6.9 (A binary Justesen family)**.** *Fix $0 < \rho < 1/2$ such that $h_2(\rho) < 1/2$. Put $b_m = V_2(2m, \lfloor 2m\rho \rfloor)$ and $[z]_+ = \max\{z, 0\}$. For $1 \le k \le q$, the map (6.4) gives a binary linear code with length $2mq$, dimension mk, rate $k/(2q)$, and*

$$d \ge [q - k + 1 - b_m]_+ (\lfloor 2m\rho \rfloor + 1), \qquad \frac{b_m}{q} \le 2^{-m(1-2h_2(\rho))}.$$

Consequently, if $k/q \to R_{\text{out}} \in (0, 1)$, then

$$R \to \frac{R_{\text{out}}}{2}, \qquad \liminf \frac{d}{2mq} \ge (1 - R_{\text{out}})\rho > 0.$$

It is an asymptotically good binary family.

The distance mechanism has two parts. At most b_m multipliers have an inner code with distance at most $\lfloor 2m\rho \rfloor$. Two Reed–Solomon words differ at at least $q - k + 1$ positions, so at least $[q - k + 1 - b_m]_+$ of these positions use good inner codes. Each contributes at least $\lfloor 2m\rho \rfloor + 1$ bit differences. The full ensemble count and proof are retained below for further reading.

The final encoder requires polynomially many bit operations in its blocklength using direct polynomial evaluation and field arithmetic. Finite-field representations can be found by exhaustive irreducibility testing in time polynomial in 2^m, which is sufficient for this blocklength claim. No search for a good inner code remains. The proof establishes distance and efficient encoding. A decoder for the full half-distance radius is an additional algorithmic question. The reading section gives a simpler staged decoder with a smaller, but positive, guaranteed error fraction.

Application. When an algorithm or reduction needs an explicit binary encoding with constant rate and tolerance to a positive fraction of adversarial errors, Justesen supplies that interface without searching for a good random inner code [15].

Further reading: additional proofs

Reed–Muller dimension and distance

There are $\sum_{j=0}^{r} \binom{m}{j}$ multilinear monomials of degree at most r. Their evaluation maps are independent. Indeed, write a multilinear polynomial as $f(x', x_m) = g(x') + x_m h(x')$. If its evaluation table is zero, restriction to $x_m = 0$ gives $g = 0$ as a function, and restriction to $x_m = 1$ gives $h = 0$ as a function. Induction in m forces every coefficient to vanish, proving the dimension formula.

For distance, the case $r = m$ is the full function space, with distance one. For $r < m$, induct in m using the same decomposition. If $h = 0$, the table consists of two copies of g and has weight at least $2 \cdot 2^{m-1-r} = 2^{m-r}$. If $h \ne 0$, then $\deg h \le r - 1$ and

$$\|f\|_0 = \|g\|_0 + \|g + h\|_0 \ge \|h\|_0 \ge 2^{(m-1)-(r-1)} = 2^{m-r}.$$

For $r = 0$, only the first case occurs. The monomial $x_1 \cdots x_r$ has weight exactly 2^{m-r}; the empty monomial for $r = 0$ is the constant one. This establishes the claimed exact distance.

The Wozencraft count and full Justesen proof

Call α bad if some nonzero $x \in F$ has $\|W_\alpha(x)\|_0 \le \lfloor 2m\rho \rfloor$. For a fixed nonzero binary pair $(u, v) \in \mathbb{F}_2^m \times \mathbb{F}_2^m$, membership in the image of W_α requires $u = [x] \ne 0$ and

$$\alpha = [\,\cdot\,]^{-1}(v)/[\,\cdot\,]^{-1}(u).$$

Thus a nonzero pair belongs to at most one inner code. Choose one low-weight nonzero witness for each bad multiplier. Their witnesses are all different. There are at most $V_2(2m, \lfloor 2m\rho \rfloor) = b_m$

possible witnesses, so there are at most b_m bad multipliers. Counting the zero vector in the volume only weakens the bound.

Linearity of evaluation, field multiplication, and the basis map makes the concatenated code binary linear. If its output is zero, the first half of every block gives $f(\alpha) = 0$ at all $q \geq k$ points, so $f = 0$. Its binary dimension is therefore mk. For two different polynomials, the root bound gives at least $q - k + 1$ different outer symbols. At most b_m such positions can be bad. At each remaining position, the difference of the two inner words is a nonzero inner word and has weight at least $\lfloor 2m\rho \rfloor + 1$. This proves the finite distance bound in Theorem 6.9.

The entropy upper bound and monotonicity of h_2 on $[0, 1/2]$ give

$$b_m \leq 2^{2mh_2(\lfloor 2m\rho \rfloor/(2m))} \leq 2^{2mh_2(\rho)}, \qquad \frac{b_m}{q} \leq 2^{-m(1-2h_2(\rho))} \longrightarrow 0.$$

Also $(\lfloor 2m\rho \rfloor + 1)/(2m) > \rho$. Divide the finite distance bound by $2mq$ and let $m \to \infty$ to obtain the asymptotic conclusion. Equivalently, for every final binary rate $R \in (0, 1/2)$ and every ρ with $h_2(\rho) < 1/2$, the family has limiting rate R and relative distance at least $(1-2R)\rho$. Allowing ρ to approach $h_2^{-1}(1/2)$ gives the familiar limiting tradeoff. This rate-1/2 inner ensemble establishes final rates below 1/2; no claim about all binary rates is needed here.

A staged decoder and its separate guarantee

Write $a_m = \lfloor 2m\rho \rfloor$ and $t_m = \lfloor a_m/2 \rfloor$. For each block, enumerate all q inner messages, select a nearest inner word, and break ties by a fixed rule. A good inner code has distance at least $a_m + 1$, so it decodes correctly whenever the block has at most t_m bit errors. The at most b_m bad inner positions may always give wrong outer symbols. If the total number of bit errors is E, at most $\lfloor E/(t_m + 1) \rfloor$ additional blocks exceed the good-block radius. The Reed–Solomon outer decoder therefore succeeds whenever

$$b_m + \left\lfloor \frac{E}{t_m + 1} \right\rfloor \leq \left\lfloor \frac{q-k}{2} \right\rfloor.$$

This is a polynomial-time decoder in the final blocklength: it examines q candidates at each of q blocks and then uses the polynomial-time outer decoder. With $k/q \to R_{\text{out}}$, it corrects every error fraction strictly below $\rho(1 - R_{\text{out}})/4$ for all sufficiently large m. The factor 1/4 comes from half-distance decoding at each of two levels. The distance bound itself permits a larger radius for an unrestricted decoder; the two statements must remain separate.

Extended catalogue: definitions and guarantees for further reading

The overview tables are intended for the classroom; this section records the mathematical meaning of the additional entries. The AG formulas invoke the cited algebraic-geometric results. The elementary BCH, binary Goppa, and expander distance arguments are given below.

Subfield subcodes and BCH codes. For a linear code $C \leq \mathbb{F}_{2^m}^n$, the *subfield subcode* $C \cap \mathbb{F}_2^n$ keeps only codewords whose coordinates are binary. Length stays n, and every two retained codewords remain at least $d(C)$ apart; restriction can lower the dimension. Binary expansion instead expresses every field coordinate in m bits and has length mn; these operations have different parameter effects. BCH and classical binary Goppa codes use subfield restriction.

For a primitive binary Bose–Chaudhuri–Hocquenghem (BCH) code, fix $m \geq 2$, $n = 2^m - 1$, a generator γ of $\mathbb{F}_{2^m}^\times$, and an integer $t \geq 1$ with $mt < n$. For $c \in \mathbb{F}_2^n$, indexed by $i = 0, \ldots, n-1$, require

$$\sum_{i=0}^{n-1} c_i \gamma^{ij} = 0 \quad \text{in } \mathbb{F}_{2^m}, \qquad 1 \leq j \leq 2t.$$

These are consecutive-root conditions on the polynomial $c(Z) = \sum_i c_i Z^i$. They define a binary linear code with

$$\boxed{k \geq n - mt, \qquad d \geq 2t + 1.}$$

For dimension, the equation for an even exponent j is the square of the equation for $j/2$, because $c_i^2 = c_i$. Iteration reduces all checks to the t odd exponents up to $2t-1$. Each is at most m binary linear constraints, giving the dimension lower bound. For distance, a putative nonzero word of weight $w \leq 2t$ would make the w columns on its support dependent in the first w checks. That square matrix has entries a_i^j for distinct nonzero support points a_i, $1 \leq j \leq w$. Its determinant is a nonzero product of the a_i and their pairwise differences, by the Vandermonde formula. The dependence is impossible. The case $t = 1$ recovers the Hamming construction. These codes extend its syndrome-based single-error protection to multiple errors; see [1, Exercise 5.11].

AG codes: replace the line by a curve. Let $X/\mathbb{F}_q$ be a smooth projective geometrically irreducible curve of genus g, with distinct rational evaluation points $P_1, \ldots, P_n$. Let G be an effective divisor over $\mathbb{F}_q$, disjoint from these points, of degree s. The Riemann–Roch space $L(G)$ consists of rational functions whose poles are bounded by G, together with zero. Encode

$$f \longmapsto (f(P_1), \ldots, f(P_n)), \qquad f \in L(G).$$

For $2g - 2 < s < n$, Riemann–Roch gives

$$\boxed{k = s + 1 - g, \qquad d \geq n - s, \qquad R + \frac{d}{n} \geq 1 - \frac{g-1}{n}.}$$

The genus-zero construction with poles at infinity is Reed–Solomon. Curves supply longer evaluation sets over a fixed field. A landmark asymptotic result is the Tsfasman–Vlăduţ–Zink bound: for fixed square $q \geq 9$ and positive R, δ satisfying

$$R + \delta = 1 - \frac{1}{\sqrt{q} - 1},$$

there are AG families with limiting lower bounds R on rate and δ on relative distance. These are q-ary statements. See [16, Theorems 21 and 40, and §4.2]. *Applications.* Arithmetic secret sharing and secure multiparty computation use the behavior of evaluation codes under coordinatewise multiplication. AG constructions also give locally recoverable codes for distributed storage; see [16, §§8–9].

Classical binary Goppa codes: algebraic checks over an extension field. Fix $F = \mathbb{F}_{2^m}$, a square-free polynomial $g(Z) \in F[Z]$ of degree $t \geq 1$, and distinct $\alpha_1, \ldots, \alpha_n \in F$ with $g(\alpha_i) \neq 0$. Write $L = (\alpha_1, \ldots, \alpha_n)$ and assume $n > mt$. Define

$$\Gamma(L, g) = \left\{ c \in \mathbb{F}_2^n : \sum_{i=1}^{n} \frac{c_i}{Z - \alpha_i} = 0 \text{ in } F[Z]/(g) \right\}.$$

The denominators are invertible modulo g by the support condition. The parameter guarantees are

$$\boxed{k \geq n - mt, \qquad d \geq 2t + 1.}$$

The quotient ring has dimension t over F, so the condition imposes at most mt binary linear checks. For distance, let a nonzero word have support S of size w, and put $P(Z) = \prod_{i \in S}(Z - \alpha_i)$. Its check is $P'(Z)/P(Z) = 0 \pmod{g}$, so g divides P'. In characteristic two, P' is a square polynomial: only even powers occur, and every coefficient in the finite field has a square root. Since g is square-free, g^2 divides P'. The distinct roots of P make $P' \neq 0$, whence $w - 1 \geq \deg P' \geq 2t$. This proves the distance bound. *Application.* McEliece-style cryptography hides the algebraic decoding description while publishing a code-based encoding or parity-check interface [17]. Its security requires computational assumptions beyond the distance guarantee.

Expander codes: expansion forces a large nonzero support. Take a simple bipartite graph with n variable nodes, $r < n$ check nodes, and left degree D. Impose even parity at every check. If both degrees are bounded, this is an LDPC code. Suppose, for an integer $1 \le a < n$, every variable set S with $1 \le |S| \le a$ has

$$|N(S)| > \frac{D}{2}|S|.$$

Then the code has $k \ge n - r$ and $d \ge a + 1$. Indeed, a nonzero codeword supported on such an S would give every adjacent check a positive even number of incident support edges, at least two. Counting edges yields $D|S| \ge 2|N(S)|$, contradicting expansion. If $a/n \ge \alpha > 0$ and $r/n \le 1 - R_0$ for fixed $R_0 > 0$ along unbounded lengths, the family is asymptotically good. This is the structural guarantee missing from sparsity alone.

Efficient decoding is a further claim. Sipser–Spielman construct asymptotically good binary codes with linear-time correction of a constant fraction of arbitrary errors [18]. Their decoding guarantees use additional expansion or local-code hypotheses; the weak distance condition just proved is not by itself a bit-flipping decoder theorem. Tanner codes generalize a parity check to requiring the incident symbols at a check to form a word of a chosen short local code; see [1, Chapter 11].

Two further directions: folded RS and fountain codes. Folded Reed–Solomon codes group consecutive RS evaluations into super-symbols in $\mathbb{F}_q^s$. A corrupted super-symbol is one error in this larger-alphabet model. The construction supports powerful *list decoders*, which return a short list of possible messages and tolerate error levels beyond unique decoding [19]. This is a natural continuation of the Reed–Solomon and Hadamard list-decoding discussions.

Fountain codes address a variable-length erasure problem. LT codes generate coded packets by XORing selected input packets; packet metadata identifies the selected subset. Lost packets remove equations; recovery becomes possible once the received equations determine all input packets. The degree distribution and decoder control how many packets are needed. The number of output packets can grow as needed, giving a rateless transmission interface [20]. Their rate and reliability are analyzed in that erasure model.

Optional: tree codes and interactive coding

This closing topic is outside the classroom route. In an interactive protocol, the next message can depend on a reply that has not yet arrived. The full transcript is therefore unavailable at the start for block encoding. A corrupted reply can change what the parties send next and leave them with inconsistent views of the conversation [22, §§1–2].

A causal code for a growing history. Label every edge of a complete binary tree of depth T by a symbol from a finite output alphabet Γ. Input bits choose a path; the encoder outputs each edge label as it traverses that edge. Write $E(a) \in \Gamma^s$ for the labels produced by a prefix $a \in \{0,1\}^s$. The next label depends only on the input bits seen so far.

For $0 < \alpha < 1$, the *tree-code distance condition* is the following: for every depth $1 \le s \le T$ and every pair of distinct prefixes $a, b \in \{0,1\}^s$ whose first difference is at position j,

$$d_H(E(a)_{j:s}, E(b)_{j:s}) \ge \alpha(s - j + 1).$$

Here $j : s$ denotes coordinates $j, \dots, s$ [22, §2.1]. The guaranteed separation scales with the time since the histories branched. It is not a claim of relative distance α between every pair of length-T outputs. Schulman's existence theorem gives such trees at every depth over an alphabet whose size depends only on α, not on T [21, §II-A, Lemma 1]. Thus the number of transmitted label bits per input bit is bounded independently of the history length. Existence alone supplies no efficient uniform construction or decoder.

Application: simulating a conversation over a noisy link. An interactive coding scheme encodes the parties' evolving histories, continually estimates the other party's history, and repairs disagreement before making further progress. Tree-code distance helps charge a long incorrect history to many channel errors. This is an application of the code inside a simulation protocol; the distance condition alone is not the complete simulation argument.

One consequence of Schulman's theorem is the following. For every deterministic two-party noiseless protocol using at most T bits in a fixed alternating schedule, a deterministic simulation uses a fixed $N = O(T)$ bit transmissions and reproduces its outputs for every input and every corruption of at most ηN bits, for some universal $\eta > 0$. The channels are synchronous, and corruptions are substitutions; local computation is unrestricted [21, Theorem 4]. Synchronous two-way rounds and alternating transmissions differ by only a constant factor here. The result provides constant communication overhead; efficient construction, decoding time, and stronger noise tolerance are separate questions.

For a tutorial treatment, see Gelles' survey [22, §2.1]; its later sections discuss efficient schemes, random versus adversarial noise, and multiparty communication.

Sources and further reading

For code bounds, polynomial codes, and concatenation, see [1]; the exact locators most relevant here are §4.4 (Plotkin), §10.3 (Wozencraft and Justesen), and §12.1, Algorithm 12.1.1 (Berlekamp–Welch). Guruswami's notes [12] give a complementary treatment, especially §2 (Justesen) and §3.3 (Berlekamp–Welch). The all-field Justesen variant and finite rounding terms are proved above; the original construction is [15].

References

[1] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. Online manuscript, version of 26 August 2025.

[2] Tao Jiang and Alexander Vardy. Asymptotic improvement of the Gilbert–Varshamov bound on the size of binary codes. *IEEE Transactions on Information Theory*, 50(8):1655–1664, 2004.

[3] Omar Alrabiah and Venkatesan Guruswami. Binary code rate bounds via classical–quantum channels. arXiv:2608.09347v1, 10 August 2026. Preprint.

[4] Morris Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960.

[5] Richard W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

[6] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. *Proceedings of STOC*, pages 25–32, 1989.

[7] Peter S. Gemmell and Madhu Sudan. Highly resilient correctors for polynomials. Technical Report UCB/CSD-92-661, University of California, Berkeley, 1992.

[8] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998.

[9] Robert G. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962.

[10] Irving S. Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[11] Lloyd R. Welch and Elwyn R. Berlekamp. Error correction for algebraic block codes. U.S. Patent 4,633,470, filed 27 September 1983, issued 30 December 1986.

[12] Venkatesan Guruswami. Notes 7: Justesen codes, Reed–Solomon and concatenated codes decoding. *Introduction to Coding Theory*, Carnegie Mellon University, Spring 2010.

[13] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[14] G. David Forney, Jr. *Concatenated Codes*. MIT Press, 1966.

[15] J. Justesen. A class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652–656, 1972.

[16] Alain Couvreur and Hugues Randriambololona. Algebraic geometry codes and some applications. arXiv:2009.01281, 2020. See §3.1 for AG parameters, §3.2 for classical Goppa codes, and §4.2 for the TVZ bound.

[17] Classic McEliece project. Classic McEliece: specification and reference mathematics. Version of 23 October 2022; see also the project's square-free Goppa-code definition.

[18] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[19] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on Information Theory*, 54(1):135–150, 2008.

[20] Michael Luby. LT codes. *Proceedings of FOCS*, pages 271–280, 2002.

[21] Leonard J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996.

[22] Ran Gelles. Coding for Interactive Communication: A Survey. *Foundations and Trends in Theoretical Computer Science*, 13(1–2):1–157, 2017. Linked author version: 1.3, October 2019.