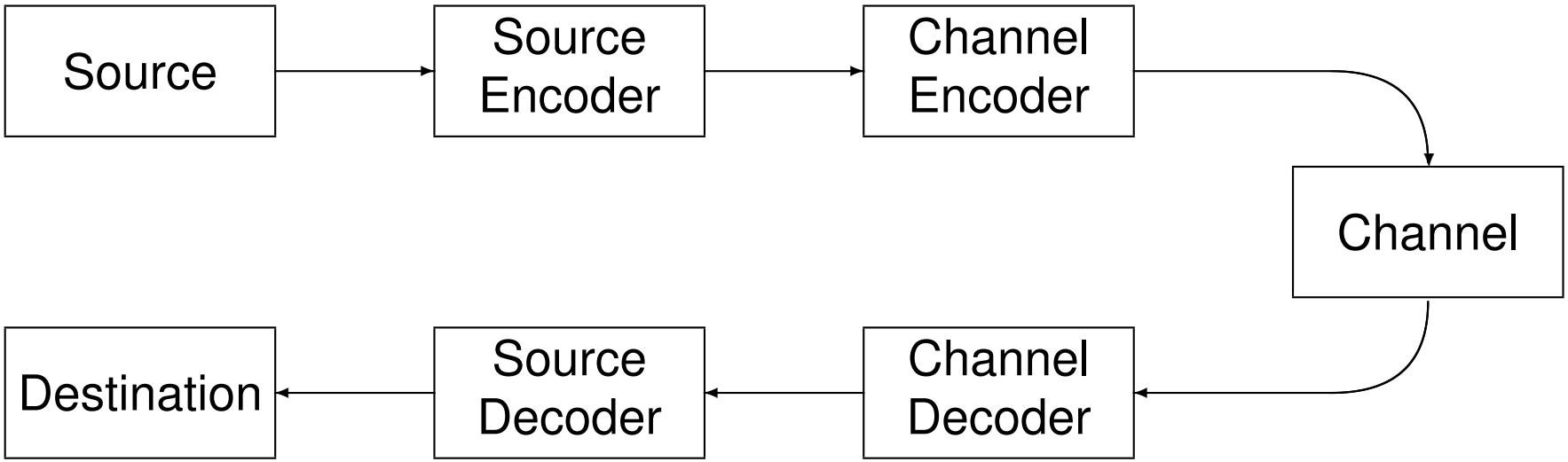
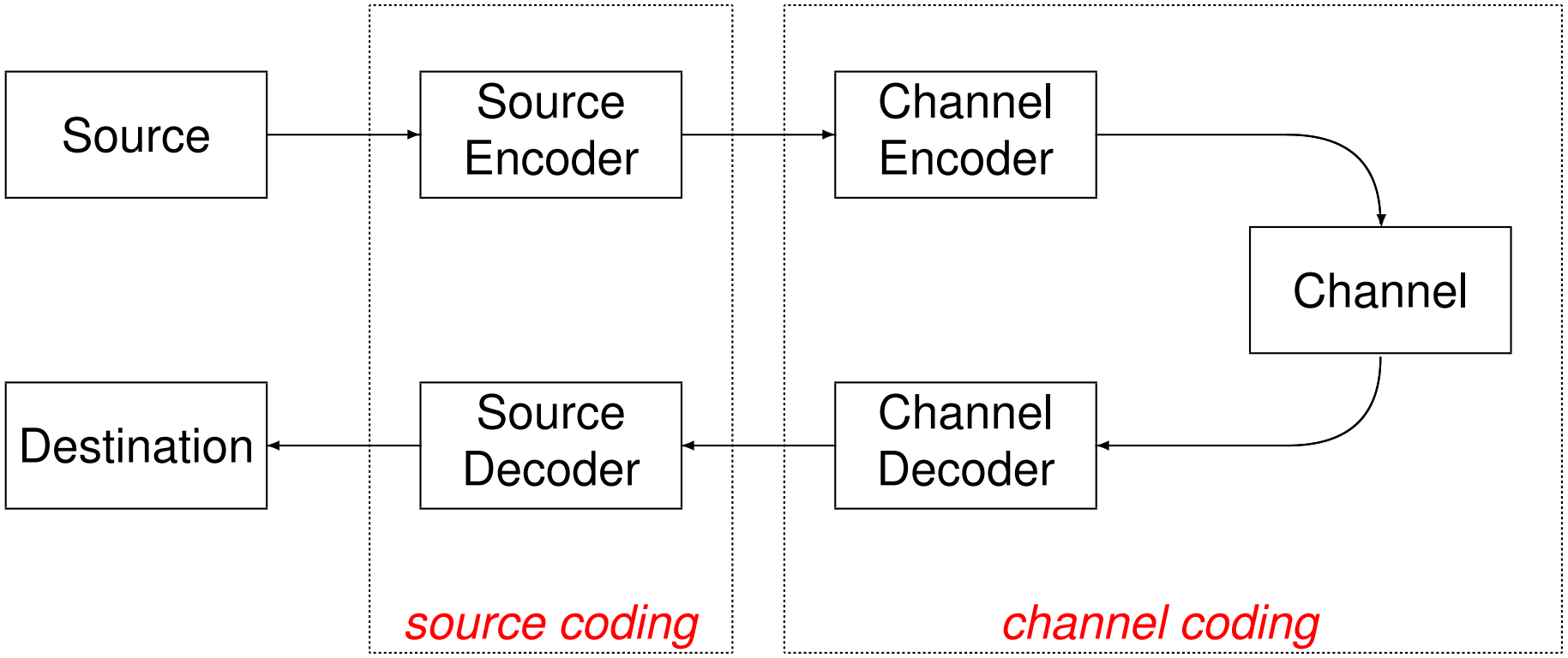
1. Introduction to Channel Coding

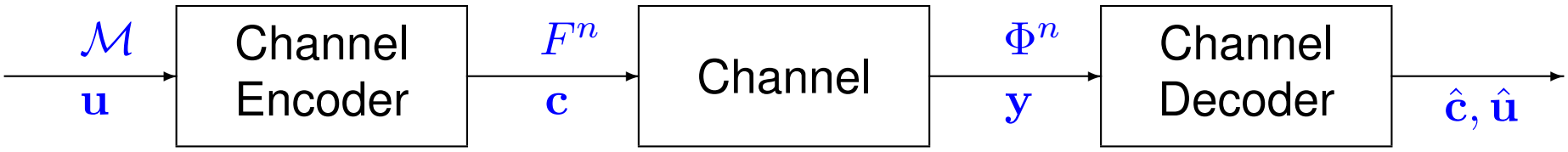
Communication System



Communication System



Channel Coding



Discrete probabilistic channel: (F, Φ, Prob)

- F: finite *input alphabet*, Φ: finite *output alphabet*
- Prob: conditional probability distribution

$$\text{Prob}\{\, \mathbf{y} \text{ received} \mid \mathbf{x} \text{ transmitted} \,\} \quad \mathbf{x} \in F^m, \quad \mathbf{y} \in \Phi^m, \quad m \geq 1$$

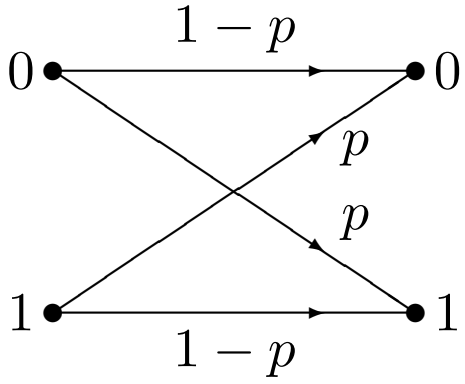
- $\mathbf{u}$: *message word* $\in \mathcal{M}$, set of M possible messages
- $\mathbf{c} \in F^n$: *codeword*, $\mathcal{E} : \mathbf{u} \xrightarrow{1-1} \mathbf{c}$ *encoding*, $\mathcal{C} = \{\mathcal{E}(\mathbf{u}) \mid \mathbf{u} \in \mathcal{M}\}$ *code*
- $\mathbf{y} \in \Phi^n$: *received word*
- $\hat{\mathbf{c}}, \hat{\mathbf{u}}$: *decoded codeword, message word*, $\mathbf{y} \longrightarrow \hat{\mathbf{c}}$ $(\longrightarrow \hat{\mathbf{u}})$ *decoding*

Code Parameters

$\mathcal{C} = \mathcal{E}(\mathcal{M}) \subseteq F^n, \quad |\mathcal{C}| = M$

- n: *code length*
- $k = \log_{|F|} M = \log_{|F|} |\mathcal{C}|$: *code dimension*
- $R = \frac{k}{n}$: *code rate* ≤ 1
- $r = n - k$: *code redundancy*
- We call $\mathcal{C}$ an (n, M) (*block*) *code* over F

Example: Memoryless Binary Symmetric Channel (BSC)



BSC(p)
$p =$ *crossover probability*

(we can assume $p \leq 1/2$)

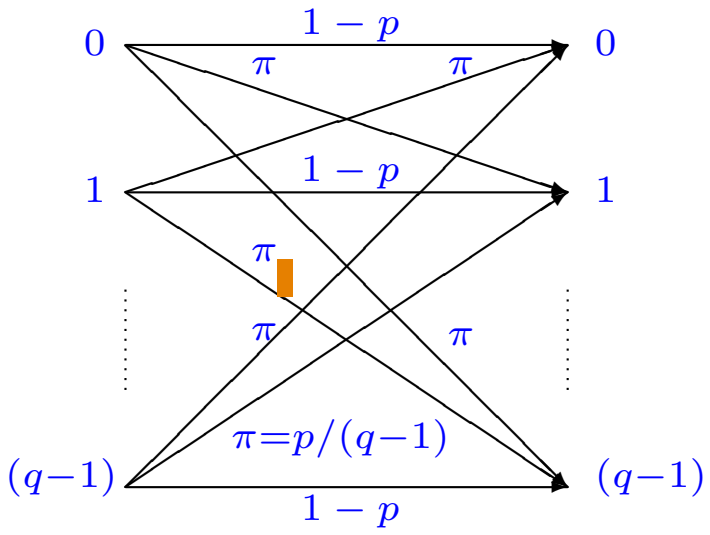
- $F = \Phi = \{0, 1\}$
- $\text{Prob}(0|1) = \text{Prob}(1|0) = p, \quad \text{Prob}(0|0) = \text{Prob}(1|1) = 1 - p$
- For $\mathbf{x} \in F^n, \ \mathbf{y} \in \Phi^m$,

$$\text{Prob}\{\, \mathbf{y} \text{ received} \mid \mathbf{x} \text{ transmitted} \,\} = \prod_{j=1}^{n} \text{Prob}(\, y_j \mid x_j \,) \ = \ p^t (1-p)^{n-t},$$

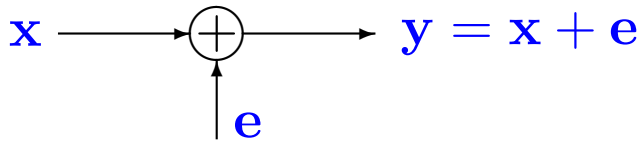
where $t = |\{\ j \mid y_j \neq x_j \ \}|$ (number of errors)

Memoryless q-ary Symmetric Channel (QSC)

- $F = \Phi, \quad |F| = q \geq 2$
- $\text{Prob}(y \mid x) = \begin{cases} 1-p, & x = y, \\ p/(q-1), & x \neq y, \end{cases} \qquad x, y \in F.$
- Assume F is an abelian group,
  e.g.: $\{0, 1, \ldots, q-1\}$ with addition mod q



- *Additive channel* (operating in the group F^n)



- $\mathbf{e} = \mathbf{y} - \mathbf{x}$: *error word* *statistically independent* of $\mathbf{x}$

$$\mathbf{e} = [\, 0 \ldots 0, \boxed{e_{i_1}}, 0 \ldots 0, \boxed{e_{i_2}}, 0 \ldots 0, \boxed{e_{i_t}}, 0 \ldots 0 \,]$$

$i_1, i_2, \ldots, i_t$: *error locations* $\qquad e_{i_1}, e_{i_2}, \ldots, e_{i_t}$: *error values* ($\neq 0$)

The Hamming Metric

- *Hamming distance*

  For single-letters $x, y \in F$: $\mathrm{d}(x,y) = \begin{cases} 0, & x = y, \\ 1, & x \neq y. \end{cases}$

  For vectors $\mathbf{x}, \mathbf{y} \in F^n$: $\mathrm{d}(x,y) = \sum_{j=0}^{n-1} \mathrm{d}(x_j, y_j)$

  number of locations where the vectors differ

- The Hamming distance defines a *metric*:
  - $\mathrm{d}(\mathbf{x}, \mathbf{y}) \geq 0$, with equality if and only if $\mathbf{x} = \mathbf{y}$
  - Symmetry $\mathrm{d}(\mathbf{x}, \mathbf{y}) = \mathrm{d}(\mathbf{y}, \mathbf{x})$
  - Triangle inequality: $\mathrm{d}(\mathbf{x}, \mathbf{y}) \leq \mathrm{d}(\mathbf{x}, \mathbf{z}) + \mathrm{d}(\mathbf{z}, \mathbf{y})$

- *Hamming weight* $\mathrm{wt}(\mathbf{e}) = \mathrm{d}(\mathbf{e}, \mathbf{0})$ *number of nonzero entries*

- When F is an abelian group, $\mathrm{d}(\mathbf{x}, \mathbf{y}) = \mathrm{wt}(\mathbf{x} - \mathbf{y})$

Minimum Distance

- Let $\mathcal{C}$ be an (n, M) code over F, $M > 1$

$$d = \min_{\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C} \,:\, \mathbf{c}_1 \neq \mathbf{c}_2} \mathrm{d}(\mathbf{c}_1, \mathbf{c}_2)$$

  is called the *minimum distance* of $\mathcal{C}$

- We say that $\mathcal{C}$ is an (n, M, d) code

- **Example:** $\mathcal{C} = \{000,\ 111\}$ is the $(3, 2, 3)$ *repetition code* over $F_2 = \{0, 1\}$,

  dimension: $k = 1$, rate: $R = 1/3$

  in general, $\mathcal{C} = \{00\ldots0,\ 11\ldots1\}$: $(n, 2, n)$ repetition code, $R = 1/n$

- **Example:** $\mathcal{C} = \{000, 011, 101, 110\}$ is the $(3, 4, 2)$ *parity code* of dimension $k = 2$ and rate $R = 2/3$ over F_2

  in general, $\mathcal{C} = \{\ (x_0, x_1, \ldots, x_{n-2}, \sum_{i=0}^{n-2} x_i)\ \}$, $(n, 2^{n-1}, 2)$ over F_2

Decoding

- $\mathcal{C} : (n, M, d)$ over F, used on channel $S = (F, \Phi, \text{Prob})$
- A *decoder* for $\mathcal{C}$ on S is a function

$$\mathcal{D} : \Phi^n \longrightarrow \mathcal{C}.$$

- *Decoding error probability* of $\mathcal{D}$ is

$$P_{\text{err}} = \max_{\mathbf{c} \in \mathcal{C}} P_{\text{err}}(\mathbf{c}) \ ,$$

where

$$P_{\text{err}}(\mathbf{c}) = \sum_{\mathbf{y} \,:\, \mathcal{D}(\mathbf{y}) \neq \mathbf{c}} \text{Prob}\{\, \mathbf{y} \text{ received} \mid \mathbf{c} \text{ transmitted} \,\} \ .$$

goal: *find encoders (codes) and decoders that make* P_{err} *small*

Decoding example

- **Example:** $\mathcal{C} = (3, 2, 3)$ binary repetition code, channel $S = \text{BSC}(p)$

  Decoder $\mathcal{D}$ defined by

$$\mathcal{D}(000) = \mathcal{D}(001) = \mathcal{D}(010) = \mathcal{D}(100) = 000$$

$$\mathcal{D}(011) = \mathcal{D}(101) = \mathcal{D}(110) = \mathcal{D}(111) = 111$$

  Error probability

$$\begin{aligned} P_{\text{err}} &= P_{\text{err}}(000) = P_{\text{err}}(111) = \tbinom{3}{2}p^2(1-p) + \tbinom{3}{3}p^3 \\ &= 3p^2 - 3p^3 + p^3 = p - p(1-p)(1-2p) \,. \end{aligned}$$

- $P_{\text{err}} < p$ for $p < 1/2 \;\Rightarrow$ *coding improved message error probability*

  but information rate is $1/3$!

  In general, for the repetition code, we have $P_{\text{err}} \to 0$ *exponentially* (prove!), but $R = 1/n \to 0$ as $n \to \infty$ *— can we do better?*

goal: *find encoders (codes) and decoders that make* P_{err} *small with minimal decrease in information rate*

Maximum Likelihood and Maximum a Posteriori Decoding

- $\mathcal{C} : (n, M, d)$, channel $S : (F, \Phi, \text{Prob})$.

  Maximum likelihood decoder (MLD):

$$\mathcal{D}_{\text{MLD}}(\mathbf{y}) = \arg\max_{\mathbf{c} \in \mathcal{C}} \text{Prob}\{\ \mathbf{y} \text{ received} \mid \mathbf{c} \text{ transmitted}\ \},\ \forall \mathbf{y} \in \Phi^n$$

  With a fixed tie resolution policy, $\mathcal{D}_{\text{MLD}}$ is well-defined for $\mathcal{C}$and S.

- *Maximum a posteriori (MAP) decoder*:

$$\mathcal{D}_{\text{MAP}}(\mathbf{y}) = \arg\max_{\mathbf{c} \in \mathcal{C}} \text{Prob}\{\ \mathbf{c} \text{ transmitted} \mid \mathbf{y} \text{ received}\ \},\ \ \forall \mathbf{y} \in \Phi^n$$

  But,

$$\begin{aligned} &\text{Prob}\{\ \mathbf{c} \text{ transmitted} \mid \mathbf{y} \text{ received}\ \} \\ &= \text{Prob}\{\ \mathbf{y} \text{ received} \mid \mathbf{c} \text{ transmitted}\ \} \cdot \frac{\text{Prob}\{\ \mathbf{c} \text{ transmitted}\ \}}{\text{Prob}\{\ \mathbf{y} \text{ received}\ \}} \end{aligned}$$

  $\Longrightarrow$ MLD and MAP are the same when $\mathbf{c}$ is *uniformly distributed*

MLD on the BSC

- $\mathcal{C} : (n, M, d)$, channel $S : \text{BSC}(p)$

$$\text{Prob}\{\, \mathbf{y} \text{ received} | \ \mathbf{c} \text{ transmitted} \,\} = \prod_{j=1}^{n} \text{Prob}\{\, y_j \text{ received} \mid c_j \text{ transmitted} \,\}$$

$$= \ p^{\mathrm{d}(\mathbf{y},\mathbf{c})}(1-p)^{n-\mathrm{d}(\mathbf{y},\mathbf{c})} = (1-p)^n \cdot \left(\frac{p}{1-p}\right)^{\mathrm{d}(\mathbf{y},\mathbf{c})},$$

where $\mathrm{d}(\mathbf{y}, \mathbf{c})$ is the Hamming distance. Since $p/(1-p) < 1$ for $p < 1/2$, for all $\mathbf{y} \in F_2^n$ we have

$$\mathcal{D}_{\text{MLD}}(\mathbf{y}) = \arg\min_{\mathbf{c} \in \mathcal{C}} \mathrm{d}(\mathbf{y}, \mathbf{c})$$

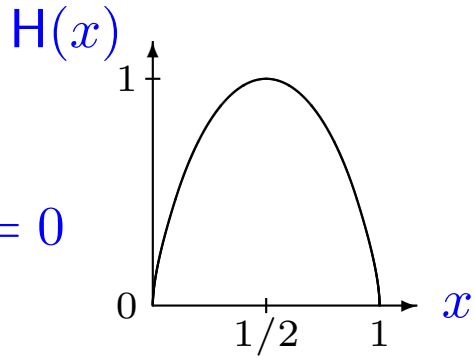
$$\boxed{\mathcal{D}_{\text{MLD}} = \textit{nearest-codeword decoder}}$$

- True also for $\text{QSC}(p)$ whenever $p < 1 - 1/q$

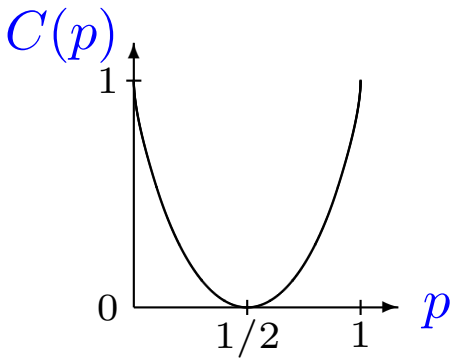
Capacity of the BSC

- *Binary entropy function* $\mathsf{H} : [0,1] \to [0,1]$

$$\mathsf{H}(x) = -x log_2 x - (1-x)\log_2(1-x), \quad \mathsf{H}(0) = \mathsf{H}(1) = 0$$



- *Capacity* of $\mathsf{BSC}(p)$ is given by $C(p) = 1 - \mathsf{H}(p)$



- A special case of the *capacity of a probabilistic channel*, as defined by Shannon (1948)

Shannon Coding Theorems for the BSC

Theorem. (Shannon Coding Theorem for the BSC – 1948.) *Let $S = BSC(p)$ and let R be a real number in the range $0 \leq R < C(p)$. There exists an infinite sequence of (n_i, M_i) block codes over F_2, $i = 1, 2, \cdots$, such that $(\log_2 M_i)/n_i \geq R$ and, for MLD for those codes (with respect to S), the probability $P_{\text{err}} \to 0$ as $i \to \infty$.*

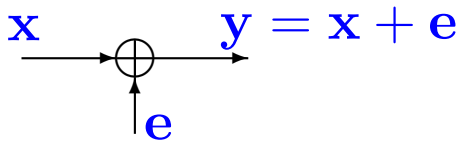
Proof. By a *random coding* argument. *Non-constructive!* ∎

Theorem. (Shannon Converse Coding Theorem for the BSC – 1948.) *Let $S = BSC(p)$ and let $R > C(p)$. Consider any infinite sequence $\{\mathcal{C}_i : (n_i, M_i)\}$ of block codes over F_2, $i = 1, 2, \cdots$, such that $(\log_2 M_i)/n_i \geq R$ and $n_1 < n_2 < \cdots < n_i < \cdots$. Then, for any decoding scheme for $\{\mathcal{C}_i\}$ (with respect to S), the probability $P_{\text{err}} \to 1$ as $i \to \infty$.*

Proof. (Loose argument.)

Error Correction

$$\mathbf{e} = [0 \ldots 0, e_{i_1}, 0 \ldots 0, e_{i_2}, 0 \ldots 0, e_{i_t}, 0 \ldots 0]$$



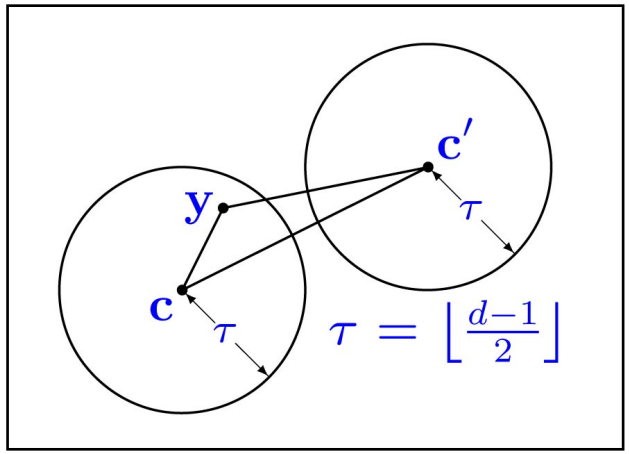
$i_1, i_2, \ldots, i_t$: *error locations* $\quad$ $e_{i_1}, e_{i_2}, \ldots, e_{i_t}$: *error values* ($\neq 0$)

- *Full error correction*: the task of recovering all $\{i_j\}$ and $\{e_{i_j}\}$ given $\mathbf{y}$

Theorem. *Let $\mathcal{C}$ be an (n, M, d) code over F. There is a decoder $\mathcal{D} : F^n \to \mathcal{C}$ that recovers correctly every pattern of up to $\lfloor (d-1)/2 \rfloor$ errors for every channel $S = (F, F, \mathsf{Prob})$.*

Proof. Let $\mathcal{D}$ be a nearest-codeword decoder.
Use triangle inequality.□

Theorem is tight: *For every $\mathcal{D}$ there is a codeword $\mathbf{c} \in \mathcal{C}$ and $\mathbf{y} \in F^n$ such that $\mathrm{d}(\mathbf{y}, \mathbf{c}) \leq \lfloor (d+1)/2 \rfloor$ and $\mathcal{D}(\mathbf{y}) \neq \mathbf{c}$.*



Error Correction Examples

- Binary $(n, 2, n)$ repetition code. Nearest-codeword decoding corrects up to $\lfloor (n-1)/2 \rfloor$ errors (take majority vote).

- Binary $(n, 2^{n-1}, 2)$ parity code cannot correct single errors: $(11100\ldots0)$ is at distance 1 from codewords $(11000\ldots0)$ and $(10100\ldots0)$

Error Detection

- Generalize the definition of a decoder to $\mathcal{D} : F^n \to \mathcal{C} \cup \{\text{'E'}\}$, where 'E' means *"I found errors, but don't know what they are"*

  Theorem. *Let $\mathcal{C}$ be an (n, M, d) code over F. There is a decoder $\mathcal{D} : F^n \to \mathcal{C} \cup \{\text{'E'}\}$ that detects (correctly) every pattern of up to $d-1$ errors.*

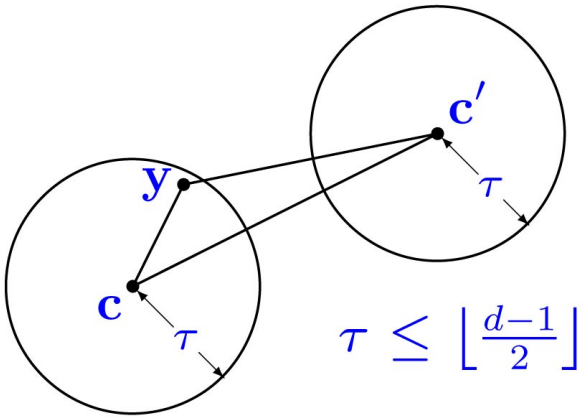
  Proof. $\mathcal{D}(\mathbf{y}) = \begin{cases} \mathbf{y} & \text{if } \mathbf{y} \in \mathcal{C} \\ \text{'E'} & \text{otherwise} \end{cases}$.

  Example: Binary $(n, 2^{n-1}, 2)$ parity code can detect single errors (a single bit error maps an even parity word to an odd parity one)

Combined correction/detection

- **Theorem.** *Let τ and σ be nonnegative integers such that $2\tau + \sigma \leq d-1$. There is a decoder $\mathcal{D} : F^n \to \mathcal{C} \cup \{\text{'E'}\}$ such that*
  - *if the number of errors is τ or less, then the errors will be recovered correctly;*
  - *otherwise, if the number of errors is $\tau + \sigma$ or less, then they will be detected.*

Proof. $\mathcal{D}(\mathbf{y}) = \begin{cases} \mathbf{c} & \text{if there is } \mathbf{c} \in \mathcal{C} \text{ such that } \mathrm{d}(\mathbf{y}, \mathbf{c}) \leq \tau \\ \text{'E'} & \text{otherwise} \end{cases}$.

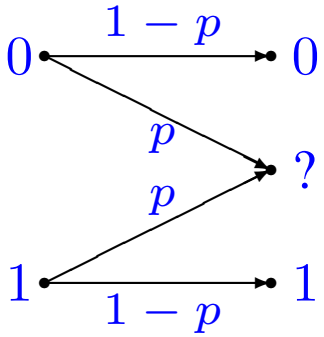


Erasure Correction

- *Erasure*: an error of which we know the *location* but not the *value*

$$[\, y_1 \cdots y_{i_1-1},\ ?\ , y_{i_1+1} \cdots y_{i_2-1},\ ?\ , y_{i_2+1} \cdots,\ ?\ , y_{i_t+1} \cdots y_n\,]$$

- *Erasure channel*: $S = (F, \Phi, \mathsf{Prob})$ with $\Phi = F \cup \{?\}$.



Theorem. *Let $\mathcal{C}$ be an (n, M, d) code over F and let $\Phi = F \cup \{?\}$. There is a decoder $\mathcal{D} : \Phi^n \to \mathcal{C} \cup \{\text{'E'}\}$ that recovers every pattern of up to $d-1$ erasures.*

Proof. On $\rho \leq d-1$ erasures, try all $|F|^\rho$ vectors that coincide with $\mathbf{y}$ in non-erased locations. Find unique codeword, if any. Otherwise, fail.

Combined correction/erasure/detection

- **Theorem.** *Let $\mathcal{C}$ be an (n, M, d) code over F and let $S = (F, \Phi, \text{Prob})$ be a channel with $\Phi = F \cup \{?\}$. For each number ρ of erasures in the range $0 \leq \rho \leq d-1$, let $\tau = \tau_\rho$ and $\sigma = \sigma_\rho$ be nonnegative integers such that $2\tau + \sigma + \rho \leq d-1$ .There is a $\mathcal{D} : \Phi^n \to \mathcal{C} \cup \{\text{'E'}\}$ such that*
  - *if the number of errors (excluding erasures) is τ or less, then all the errors and erasures will be recovered correctly;*
  - *otherwise, if the number of errors is $\tau + \sigma$ or less, then the decoder will return* 'E'*.*

- Full error correction "costs" twice as much as detection or erasure correction. Price list:
  - full error to correct: requires 2 units of distance
  - erasure to correct: requires 1 unit of distance
  - full error to detect: requires 1 unit of distance

- How does distance "cost" translate to redundancy "cost"?

Summary

- (n, M, d) *code* over alphabet F:

$$\mathcal{C} \subseteq F^n, \quad |\mathcal{C}| = M, \quad d = \min_{\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}, \mathbf{c}_1 \neq \mathbf{c}_2} \mathrm{d}(\mathbf{c}_1, \mathbf{c}_2)$$

- $k = \log_{|F|} M$: *code dimension*
  $r = n - k$: *code redundancy*
  $R = k/n$: *code rate*
- *Maximum likelihood decoding*: $\hat{\mathbf{c}} = \arg\max_{\mathbf{c}\in\mathcal{C}} \text{Prob}\{\mathbf{y} \text{ received} \mid \mathbf{c} \text{ sent}\}$
- For QSC, equivalent to $\hat{c} = \arg\min_{\mathbf{c}\in\mathcal{C}} \mathrm{d}(\mathbf{y}, \mathbf{c})$ *nearest codeword decoding*

Summary

- Shannon: there are sequences of codes $\mathcal{C}_i(n_i, M_i)$ that allow $P_{\text{err}}(\mathcal{C}_i) \stackrel{i\to\infty}{\to} 0$ while keeping $R_i \geq R > 0$, as long as $R < C$, where C is a number that depends solely on the channel (*channel capacity*)

  Error-free communication is possible at positive information rates

  (he just didn't tell us how to implement this in practice)

- Maximum likelihood decoding may be too complex: sometimes we need to settle for less

- If $2\tau + \rho + \sigma \leq d - 1$, an (n, M, d) code can
  - correct ρ *erasures* and τ *full errors*
  - *detect* between $\tau + 1$ and $\tau + \sigma$ errors (in addition to ρ erasures)

- Challenges: how to find good codes (codes with large d), how to represent them compactly, how to encode, how to decode

2. Linear Codes

Linear Codes

- Assume $\mathbb{F}$ can be given a *finite* (or *Galois*) *field* structure
  - $|\mathbb{F}| = q$, where $q = p^m$ for some prime number p and integer $m \geq 1$. We denote such a field by $\mathbb{F}_q$ or $\mathrm{GF}(q)$
  - Example: $\mathbb{F}_2$ with $\mathrm{XOR}, \mathrm{AND}$ operations
- $\mathcal{C} : (n, M, d)$ over $\mathbb{F}_q$ is called *linear* if $\mathcal{C}$ is a linear sub-space of $\mathbb{F}^n$ over $\mathbb{F}$
  - $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C},\ a_1, a_2 \in \mathbb{F} \ \Rightarrow\ a_1\mathbf{c}_1 + a_2\mathbf{c}_2 \in \mathcal{C}$
- A linear code $\mathcal{C}$ has $M = q^k$ codewords, where $k = \log_q M$ is the dimension of $\mathcal{C}$ as a linear space over $\mathbb{F}$
- $r = n - k$ is the redundancy of $\mathcal{C}$, $R = k/n$ its rate
- We use the notation $[n, k, d]$ to denote the parameters of a linear code

Generator Matrix

- A *generator matrix* for a linear code $\mathcal{C}$ is a $k \times n$ matrix G whose rows form a basis of $\mathcal{C}$
- **Example:** $G = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \quad \hat{G} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$

  are *both* generators of the $[3, 2, 2]$ parity code over $\mathbb{F}_2$
- In general, the $[n, n-1, 2]$ parity code over any F is generated by

$$G = \left(\begin{array}{c|c} I & \begin{matrix} -1 \\ -1 \\ \vdots \\ -1 \end{matrix} \end{array} \right),$$

  where I is the $(n-1) \times (n-1)$ identity matrix
- What's G for the repetition code? $G = (1\ 1\ \ldots\ 1)$

Minimum Weight

- For an $[n, k, d]$ code $\mathcal{C}$,

$$\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C} \implies \mathbf{c}_1 - \mathbf{c}_2 \in \mathcal{C}\,, \text{ and } \mathsf{d}(\mathbf{c}_1, \mathbf{c}_2) = \mathsf{wt}(\mathbf{c}_1 - \mathbf{c}_2)\,.$$

Therefore,

$$d = \min_{\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}\,:\, \mathbf{c}_1 \neq \mathbf{c}_2} \mathsf{d}(\mathbf{c}_1, \mathbf{c}_2) = \min_{\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}\,:\, \mathbf{c}_1 \neq \mathbf{c}_2} \mathsf{wt}(\mathbf{c}_1 - \mathbf{c}_2) = \min_{\mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}} \mathsf{wt}(\mathbf{c})\,.$$

$\Rightarrow$ *minimum distance is the same as minimum weight for linear codes*

- Recall also that $\mathbf{0} \in \mathcal{C}$ and $\mathsf{d}(\mathbf{c}, \mathbf{0}) = \mathsf{wt}(\mathbf{c})$

Encoding Linear Codes

- Since $\mathrm{rank}(G) = k$, the map $\mathcal{E} : \mathbb{F}^k \to \mathcal{C}$ defined by

$$\mathcal{E} : \mathbf{u} \mapsto \mathbf{u}G$$

  is 1-1, and can serve as an encoding mechanism for $\mathcal{C}$.

- Applying elementary row operations and possibly reordering coordinates, we can bring G to the form

$$G = \left(\begin{array}{c|c} I & A \end{array} \right) \qquad \textit{systematic generator matrix,}$$

  where I is a $k \times k$ identity matrix, and A is a $k \times (n-k)$ matrix.

$$\mathbf{u} \mapsto \mathbf{u}G = \left(\mathbf{u} \mid \mathbf{u}A \right) \qquad \textit{systematic encoding.}$$

- In a systematic encoding, the *information symbols* from $\mathbf{u}$ are transmitted 'as is,' and $n-k$ *check symbols* (or *redundancy* symbols, or *parity* symbols) are appended.

Parity Check Matrix

- Let $\mathcal{C} : [n, k, d]$. A *parity-check matrix (PCM)* of $\mathcal{C}$ is an $r \times n$ matrix H such that for all $\mathbf{c} \in \mathbb{F}^n$,

$$\mathbf{c} \in \mathcal{C} \qquad \Longleftrightarrow \qquad H\mathbf{c}^T = \mathbf{0}\,.$$

  - $\mathcal{C}$ is the (right) kernel of H in $\mathbb{F}^n$. Therefore,

$$\mathrm{rank}(H) = n - \dim\ker(H) = n - k$$

  - We will usually have $r = \mathrm{rank}(H) = n - k$ (no superfluous rows)

- For a generator matrix G of $\mathcal{C}$, we have

$$HG^T = 0 \Rightarrow GH^T = 0, \quad \text{and} \quad \dim\ker(G) = n - \mathrm{rank}(G) = n - k = r$$

- If $G = (\ I \mid A\ )$, then $H = (-A^T \mid I\ )$ is a (systematic) parity-check matrix.

Dual Code

- The *dual* code of $\mathcal{C} : [n, k, d]$ is

$$\mathcal{C}^{\perp} = \{ \mathbf{x} \in F^n \ : \ \mathbf{x}\mathbf{c}^T = 0 \ \ \forall \mathbf{c} \in \mathcal{C} \},$$

  or, equivalently

$$\mathcal{C}^{\perp} = \{ \mathbf{x} \in F^n \ : \ \mathbf{x}G^T = \mathbf{0} \}.$$

- $(\mathcal{C}^{\perp})^{\perp} = \mathcal{C}$
- G and H of $\mathcal{C}$ reverse roles for $\mathcal{C}^{\perp}$:

$$\mathcal{C} : \left\{ \begin{array}{ccc} G & = & H^{\perp} \\ H & = & G^{\perp} \end{array} \right\} \ : \ \mathcal{C}^{\perp}.$$

- $\mathcal{C}^{\perp}$ is an $[n, n-k, d^{\perp}]$ code over $\mathbb{F}$

Examples

- $H = (1\ 1\ \ldots\ 1)$ is a PCM for the $[n, n-1, 2]$ parity code, which has generator matrix

$$G = \left(\begin{array}{c|c} & -1 \\ I & -1 \\ & \vdots \\ & -1 \end{array} \right) .$$

  On the other hand, H generates the $[n, 1, n]$ repetition code, and G is a check matrix for it $\Rightarrow$ *parity and repetition codes are dual*.

- $[7, 4, 3]$ *Hamming code* over $\mathbb{F}_2$ is defined by

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} , \qquad G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} .$$

- $GH^T = 0$ can be verified by direct inspection

Minimum Distance and H

- **Theorem.** *Let H be a PCM of $\mathcal{C} \neq \{\mathbf{0}\}$. The minimum distance of $\mathcal{C}$ is the largest integer d such that every subset of $d-1$ columns in H is linearly independent.*

  Proof. There is a codeword $\mathbf{c}$ of weight t in $\mathcal{C}$ if and only if there are t l.d. columns in H (those columns that correspond to non-zero coordinates of $\mathbf{c}$). □

- **Example:** Code $\mathcal{C}$ with

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} .$$

  All the columns are different $\Rightarrow$ every 2 columns are linearly independent $\Rightarrow\ d \geq 3$.

  On the other hand, $H \cdot [1\,1\,1\,0\,0\,0\,0]^T = \mathbf{0} \ \Rightarrow\ d = 3$.

The Binary Hamming Code

- The m-th *order Hamming code* $\mathcal{H}_m$ over $\mathbb{F}_2$ is defined by the $m \times (2^m - 1)$ PCM

$$H_m = [\, \mathbf{h}_1 \; \mathbf{h}_2 \; \ldots \; \mathbf{h}_{2^m-1} \,],$$

where $\mathbf{h}_i$ is the length-m (column) binary representation of i.

Theorem. *$\mathcal{H}_m$ is a $[2^m - 1, 2^m - 1 - m, 3]$ linear code.*

Proof. $[n, k]$ parameters are immediate. No two columns of H_m are l.d. $\Rightarrow d \geq 3$. On the other hand, $\mathbf{h}_1 + \mathbf{h}_2 + \mathbf{h}_3 = \mathbf{0}$ for all m. $\square$

The q-ary Hamming Code

- The m-th order Hamming code $\mathcal{H}_{q,m}$ over $\mathbb{F} = \mathbb{F}_q,\ q \geq 2$, has PCM $H_{q,m}$ consisting of all distinct nonzero m-columns $\mathbf{h} \in \mathbb{F}_q^m$ *up to scalar multiples*, e.g.

$$\mathbf{h} \in H_{q,m} \ \Rightarrow\ a\mathbf{h} \notin H_{q,m}\ \ \forall a \in \mathbb{F}_q - \{1\}.$$

Theorem. *$\mathcal{H}_{q,m}$ is an $[n, n-m, 3]$ code with*

$$n = \frac{q^m - 1}{q - 1}$$

Proof. As before, no two columns of $H_{q,m}$ are multiples of each other, i.e. dependent. One the other hand, there are l.d. triplets of columns. □

Cosets and Syndromes

- Let $\mathbf{y} \in \mathbb{F}^n$. The *syndrome* of $\mathbf{y}$ (with respect to a PCM H of $\mathcal{C}$) is defined by
$$\mathbf{s} = H\mathbf{y}^T \in \mathbb{F}^{n-k}.$$
The set
$$\mathbf{y} + \mathcal{C} = \{\mathbf{y} + \mathbf{c} \ : \ \mathbf{c} \in \mathcal{C}\}$$
is a *coset* of $\mathcal{C}$ (as an additive subgroup) in $\mathbb{F}^n$.

- If $\mathbf{y}_1 \in \mathbf{y} + \mathcal{C}$, then
$$\mathbf{y}_1 - \mathbf{y} \in \mathcal{C} \Rightarrow H(\mathbf{y}_1 - \mathbf{y})^T = \mathbf{0} \Rightarrow H\mathbf{y}_1^T = H\mathbf{y}^T$$
$\Rightarrow$ *The syndrome is invariant for all* $\mathbf{y}_1 \in \mathbf{y} + \mathcal{C}$.

- Let $F = F_q$. Given a PCM H, there is a 1-1 correspondence between the q^{n-k} cosets of $\mathcal{C}$ in $\mathbb{F}^n$ and the q^{n-k} possible syndrome values (H is full-rank $\Rightarrow$ all values are attained).

Syndrome Decoding of Linear Codes

- $\mathbf{c} \in \mathcal{C}$ is sent and $\mathbf{y} = \mathbf{c} + \mathbf{e}$ is received on an additive channel
- $\mathbf{y}$ and $\mathbf{e}$ are in the same coset of $\mathcal{C}$
- Nearest-neighbor decoding of $\mathbf{y}$ calls for finding the closest codeword $\mathbf{c}$ to $\mathbf{y}$ $\Rightarrow$ find a vector $\mathbf{e}$ of *lowest weight* in $\mathbf{y}+\mathcal{C}$: a *coset leader*.
  - *coset leaders need not be unique* (when are they?)
- Decoding algorithm: upon receiving $\mathbf{y}$
  - compute the syndrome $\mathbf{s} = H\mathbf{y}^T$
  - find a coset leader $\mathbf{e}$ in the coset corresponding to $\mathbf{s}$
  - decode $\mathbf{y}$ into $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$
- If $n - k$ is (very) small, a table containing one leader per coset can be pre-computed. The table is indexed by $\mathbf{s}$.
- In general, however, syndrome decoding appears exponential in $n - k$. In fact, it has been shown to be NP-hard.

Decoding the Hamming Code

1. Consider $\mathcal{H}_m$ over $\mathbb{F}_2$. Given a received $\mathbf{y}$,

$$\mathbf{s} = H_m \mathbf{y}^T$$

is an m-tuple in $\mathbb{F}_2^m$, $n = 2^m - 1$, $m = n - k$

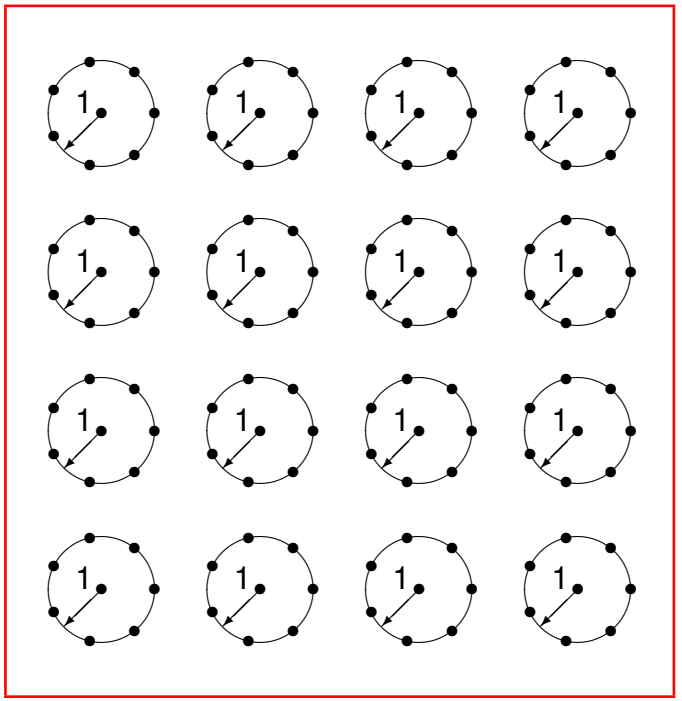
2. if $\mathbf{s} = \mathbf{0}$ then $\mathbf{y} \in \mathcal{C} \Rightarrow \mathbf{0}$ is the coset leader of $\mathbf{y} + \mathcal{C}$

3. if $\mathbf{s} \neq \mathbf{0}$ then $\mathbf{s} = \mathbf{h}_i$ for some $1 \leq i \leq 2^m - 1 \Rightarrow$

$$\mathbf{e}_i = [0, \quad 0, \quad \ldots, \quad 0, \quad \underset{\substack{\Uparrow \\ i}}{1}, \quad 0, \quad \ldots, \quad 0]$$

is the coset leader of $\mathbf{y} + \mathcal{C}$, since

$H_m \mathbf{y}^T = \mathbf{s} = \mathbf{h}_i = H_m \mathbf{e}_i$, $\mathbf{y} \notin \mathcal{C}$, and $\mathrm{wt}(\mathbf{e}_i) = 1$.



- *every word in $\mathbb{F}_2^n$ is at distance at most 1 from a codeword*
- *spheres of radius 1 around codewords are disjoint and cover $\mathbb{F}_2^n$:* ***perfect code***

steps 1–3 above describe a *complete decoding algorithm* for $\mathcal{H}_m$, $\forall m$

Deriving Codes from Other Codes

- *Adding an overall parity check.* Let $\mathcal{C}$ be an $[n, k, d]$ code with some odd-weight codewords. We form a new code $\hat{\mathcal{C}}$ by appending a 0 at the end of even-weight codewords, and a 1 at the end of odd-weight ones.
  - $\hat{\mathcal{C}}$ is an $[n+1, k, d+1]$ code. Every codeword in $\hat{\mathcal{C}}$ has even weight.
  - **Example:** The $[7, 4, 3]$ binary Hamming code can be extended to an $[8, 4, 4]$ code with PCM

$$\hat{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

corrects any pattern of 1 error, and detects any pattern of 2.

- *Expurgate by throwing away codewords.* E.g., select subset of codewords satisfying an independent parity check.
  - **Example:** Selecting the even-weight sub-code of the $[2^m - 1, 2^m - 1 - m, 3]$ Hamming code yields a $[2^m - 1, 2^m - 2 - m, 4]$ code.
- *Shortening by taking a cross-section.* Select all codewords $\mathbf{c}$ with, say, $c_1 = 0$, and eliminate that coordinate (can be repeated for more coordinates). An $[n, k, d]$ code yields an $[n-1, k-1, \geq d]$ code.

3. Bounds on Code Parameters

The Singleton Bound

- The *Singleton bound*.

  Theorem. *For any* (n, M, d) *code over an alphabet of size* q*,*

  $$d \leq n - (\log_q M) + 1 \ .$$

  Proof. Let $\ell = \lceil \log_q M \rceil - 1$. Since $q^\ell < M$, there must be at least two codewords that agree in their first ℓ coordinates. Hence, $d \leq n - \ell$. $\square$

- For linear codes, we have $d \leq n = k + 1$.

- $\mathcal{C} : (n, M, d)$ is called *maximum distance separable (MDS)* if it meets the Singleton bound, namely $d = n - (\log_q M) + 1$.

MDS Code Examples

- Trivial and semi-trivial codes
  - $[n, n, 1]$ whole space $\mathbb{F}_q^n$, $[n, n-1, 2]$ parity code, $[n, 1, n]$ repetition code
- *Normalized generalized Reed-Solomon (RS) codes*

  Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be *distinct* elements of $\mathbb{F}_q$, $n \leq q$. The RS code has PCM

$$H_{\mathrm{RS}} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \ldots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \ldots & \alpha_n^{n-k-1} \end{pmatrix}.$$

Theorem. *Every Reed-Solomon code is MDS.*

Proof. Every $(n-k) \times (n-k)$ sub-matrix of H_{RS} has a nonsingular *Vandermonde* form. Hence, every $(n-k)$ columns of H_{RS} are l.i. $\Rightarrow d \geq n - k + 1$. $\square$

The Sphere-Packing Bound

- The *sphere* of center $\mathbf{c}$ and radius t in $\mathbb{F}_q^n$ is the set of vectors at Hamming distance t or less from $\mathbf{c}$. Its *volume* (cardinality) is

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

Theorem. [The sphere-packing (SP) bound)] *For any* (n, M, d) *code over* $\mathbb{F}_q$*,*

$$M \cdot V_q(n, \lfloor (d-1)/2 \rfloor) \leq q^n \ .$$

Proof. Spheres of radius $t = \lfloor (d-1)/2 \rfloor$ centered at codewords must be disjoint. □

- For a linear $[n, k, d]$ code, the bound becomes $V_q(n, \lfloor (d-1)/2 \rfloor) \leq q^{n-k}$. For $q = 2$,

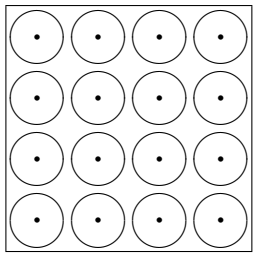
$$\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i} \leq 2^{n-k}$$

Perfect Codes

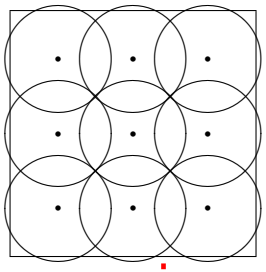
- A code meeting the SP bound is said to be *perfect*.
- Known perfect codes:
  - $[n, n, 1]$ whole space $\mathbb{F}_q^n$,
  - $[n, 1, n]$ repetition code for n odd
  - $\mathcal{H}_{q,m}$, q any GF size, $m \geq 1$
  - the $[23, 12, 7]$ binary and $[11, 6, 5]$ ternary *Golay* codes

In a well-defined sense, this is it!!!
Any perfect code must have parameters identical to one of the above

- Perfect *packing* codes are also perfect *covering codes*



packing



covering

application

The Gilbert-Varshamov bound

- The Singleton and SP bounds set *necessary* conditions on the parameters of a code. The following is a *sufficient* condition:

Theorem. [The Gilbert-Varshamov (GV) bound] *There exists an* $[n, k, d]$ *code over the field* $\mathbb{F}_q$ *whenever*

$$V_q(n-1, d-2) < q^{n-k}.$$

Proof. Construct, iteratively, an $(n-k) \times n$ PCM where every $d-1$ columns are l.i., starting with an identity matrix, and adding a new column in each iteration. Assume we've gotten $\ell-1$ columns. There are at most $V_q(n-1, d-2)$ linear combinations of $d-2$ or less of these columns. As long as $V_q(\ell-1, d-2) < q^{n-k}$, we can find a column we can add without creating a dependence of $d-1$ columns. □

Theorem. *Let*

$$\rho = \frac{q^k - 1}{q-1} \cdot \frac{V_q(n, d-1)}{q^n} .$$

Then, a random $[n, k]$ *code has minimum distance* d *with* $\mathsf{Prob} \geq 1 - \rho$.

Lots of codes are near the GV bound. But it's very hard to find them!

Asymptotic Bounds

- **Def.:** *relative distance* $\delta = d/n$
- We are interested in the behavior of δ and $R = \log_q M$ as $n \to \infty$.
- Singleton bound: $d \le n - \lceil \log_q M \rceil + 1 \;\Rightarrow\; \boxed{R \le 1 - \delta + o(1)}$
- For the SP and GV bounds, we need estimates for $V_q(n,t)$
- **Def.:** *symmetric q-ary entropy function* $\mathsf{H}_q : [0,1] \to [0,1]$

$$\mathsf{H}_q(x) = -x \log_q x - (1-x) \log_q (1-x) + x \log_q (q-1) \;,$$

  - $\mathsf{H}_q(0) = 0,\ \mathsf{H}_q(1) = \log_q(q-1)$, strictly $\cap$-convex, $\max = 1$ at $x = 1 - 1/q$
  - coincides with $\mathsf{H}(x)$ when $q = 2$

Asymptotic Bounds (II)

Lemma. *For* $0 \leq t/n \leq 1-(1/q)$,

$$V_q(n,t) = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leq q^{n\mathsf{H}_q(t/n)} \ .$$

Lemma. *For integers* $0 \leq t \leq n$,

$$V_q(n,t) \geq \binom{n}{t} (q-1)^t \geq \frac{1}{\sqrt{8t(1-(t/n))}} \cdot q^{n\mathsf{H}_q(t/n)} \ .$$

Theorem. [Asymptotic SP bound] *For every* $(n, q^{nR}, \delta n)$ *code over* $\mathbb{F}_q$,

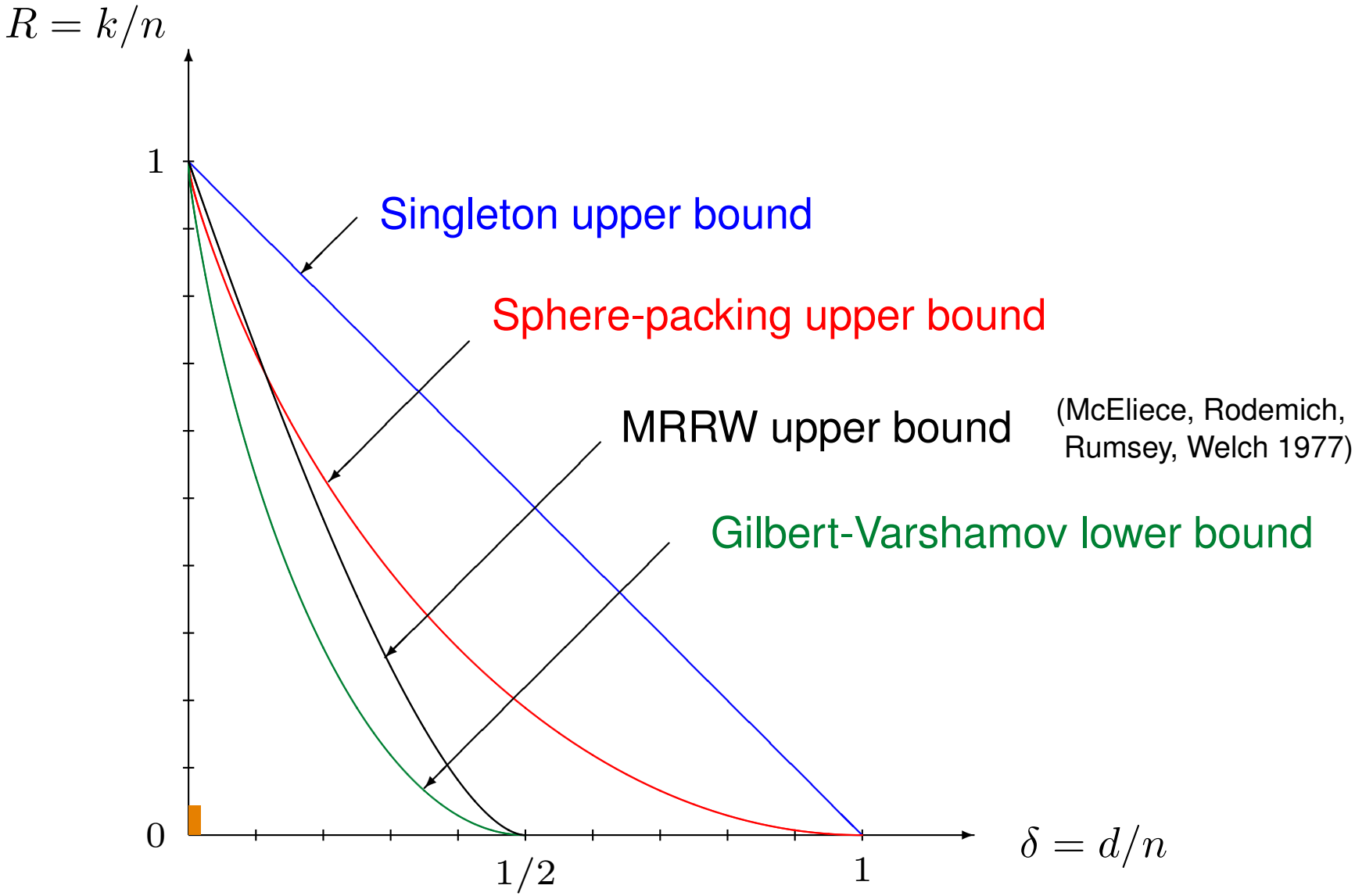
$$\boxed{R \leq 1 - \mathsf{H}_q(\delta/2) + o(1) \ .}$$

Theorem. [Asymptotic GV bound] *Let* $n,\ nR,\ \delta n$ *be positive integers such that* $\delta \in (0, 1-(1/q)]$ *and*

$$\boxed{R \leq 1 - \mathsf{H}_q(\delta) \ .}$$

Then, there exists a linear $[n, nR, \geq \delta n]$ *code over* Fq.

Plot of Asymptotic Bounds



4. Brief Review of Finite Fields

Finite Field Basics

- For a prime p, $\mathbb{F}_p$ (or $\mathrm{GF}(p)$) denotes the ring of integers mod p
- $\mathbb{F}_p$ is a (*finite, or Galois*) field
  - For every integer $a \in \{1, 2, \ldots, p-1\}$, we have $\gcd(a, p) = 1$. By Euclid's algorithm, there exist integers s and t such that $s \cdot a + t \cdot p = 1$. The integer s, taken modulo p, is the multiplicative inverse of a in the field $\mathbb{F}_p$.

    Example: Inverse of 16 modulo 41:

$$
\begin{array}{rcrcrcrcr}
9 & = & 41 & - & 2 \cdot 16 & = & 1 \cdot 41 & - & 2 \cdot 16 \\
7 & = & 16 & - & 1 \cdot 9 & = & -1 \cdot 41 & + & 3 \cdot 16 \\
2 & = & 9 & - & 1 \cdot 7 & = & 2 \cdot 41 & - & 5 \cdot 16 \\
1 & = & 7 & - & 3 \cdot 2 & = & -7 \cdot 41 & + & 18 \cdot 16
\end{array}
$$

$\Rightarrow \quad 18 = 16^{-1}$ in $\mathbb{F}_{41}$

- **Proposition.** *Let $\mathbb{F}$ be a finite field, let $q = |\mathbb{F}|$, and let $a \in \mathbb{F}$. Then, $q \cdot a = \underbrace{a + a + \cdots + a}_{q} = 0$ and $a^q = a$.*

  Proof. By Lagrange's theorem on the additive group $\mathbb{F}^+$ and the multiplicative group $\mathbb{F}^*$ of $\mathbb{F}$, and the fact that $0^{|\mathbb{F}|} = 0$. □

Field Characteristic

- Let 1 be the identity in $\mathbb{F}^*$. The *characteristic* $\mathrm{char}(\mathbb{F})$ of $\mathbb{F}$ is the order of 1 in the *additive* group $\mathbb{F}^+$, if finite. Otherwise, $\mathrm{char}(\mathbb{F}) = 0$.
  - For a finite field $\mathbb{F}$, we always have $\mathrm{char}(\mathbb{F}) > 0$.
  - **Examples:** $\mathrm{char}(\mathbb{F}_7) = 7$, $\mathrm{char}(\mathbb{Q}) = \mathrm{char}(\mathbb{R}) = 0$. Consider $\mathbb{K} = \mathbb{F}_2(x)$, the field of rational functions with coefficients in $\mathbb{F}_2$. Then, $\mathrm{char}(K) = 2$ even though $\mathbb{K}$ is infinite.

- **Proposition.** *If $\mathrm{char}(\mathbb{F}) > 0$ then it is a prime p. $\mathbb{F}$ then contains a sub-field isomorphic to $\mathbb{F}_p$.*

- **Proposition.** *Let $\mathbb{F}$ be a finite field, let $a, b \in \mathbb{F}$, and let $p = \mathrm{char}(\mathbb{F})$. Then $(a+b)^p = a^p + b^p$.*

  Proof. The binomial coefficient $\binom{p}{i} = \frac{p(p-1)(p-2)\cdots(p-i+1)}{i!}$ is a multiple of p for $0 < i < p$. □

Polynomials

- For a field $\mathbb{F}$ and indeterminate x,
  - $\mathbb{F}[x]$: ring of polynomials in x, with coefficients in $\mathbb{F}$. This is an *Euclidean* ring: degree, divisibility, division with reminder, GCD, etc. are well defined and "behave" as we're used to over the reals
  - The *extended Euclidean algorithm* can be applied to elements of $\mathbb{F}[x]$, and for $a, b \in \mathbb{F}[x]$, not both zero, we have polynomials $s(x), t(x)$ such that

$$s(x) \cdot a(x) + t(x) \cdot b(x) = \gcd(a(x), b(x))$$

- $P(x) \in \mathbb{F}[x]$ is called *irreducible* if

$$\deg(P(x))>0 \text{ and } P(x)=a(x)b(x) \Rightarrow \deg(a(x))=0 \text{ or } \deg(b(x))=0$$

  - **Example:** irreducibles over $\mathbb{F}_2$
    degree 1: $x, x+1$ degree 3: $x^3+x+1,\ x^3+x^2+1$
    degree 2: x^2+x+1 degree 4: $x^4+x+1,\ x^4+x^3+1,\ x^4+x^3+x^2+x+1$
  - $\mathbb{F}[x]$ is a *unique factorization domain* (factorization into irreducible polynomials is unique up to permutation and scalar multiples).

Arithmetic Modulo an Irreducible Polynomial

- Let $\mathbb{F}$ be a field and $P(x)$ an *irreducible* polynomial of degree $h \geq 1$.
- The ring of residue classes $\mathbb{F}[x]$ modulo $P(x)$ is denoted $\mathbb{F}[x]/\langle P(x) \rangle$.
  - Let $\mathbb{F}[x]_n =$ set of polynomials of degree $< n$ in x over $\mathbb{F}$
  - $\mathbb{F}[x]/\langle P(x) \rangle$ can be represented by $\mathbb{F}[x]_h$ with arithmetic mod $P(x)$.

Theorem. $\mathbb{F}[x]/\langle P(x) \rangle$ *is a field.* □

- This theorem, and the one saying $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ is a field, are special cases of the same theorem on Euclidean rings.
- As with integers, inverses are found found using the Euclidean algorithm: $\gcd(a(x), P(x)=1 \Rightarrow \exists s(x), t(x)\colon\ s(x)a(x)+t(x)P(x)=1 \Rightarrow s(x)$ is a multiplicative inverse of $a(x)$ in $\mathbb{F}[x]/\langle P(x) \rangle$.

Example: Inverse of x^2 modulo x^3+x+1 over $\mathbb{F}_2$ (recall that $z = -z$).

$$
\begin{aligned}
x+1 &= x^3+x+1 + x\cdot x^2 &&= 1\cdot(x^3+x+1) + x\cdot(x^2)\\
x &= x^2 + x\cdot(x+1) &&= x\cdot(x^3+x+1) + (x^2+1)\cdot(x^2)\\
1 &= (x+1) + x &&= (x+1)\cdot(x^3+x+1) + (x^2+x+1)\cdot(x^2)
\end{aligned}
$$

$\Rightarrow \quad x^2+x+1 = (x^2)^{-1}$ in $\mathbb{F}_2[x]/\langle x^3+x+1 \rangle$

Extension Fields

- A field $\mathbb{K}$ is an *extension field* of a field $\mathbb{F}$ if $\mathbb{F}$ is a *sub-field* of $\mathbb{K}$
- $\mathbb{K}$ is a vector space over $\mathbb{F}$. The dimension $[\mathbb{K} : \mathbb{F}]$ of this vector space is referred to as the *extension degree* of $\mathbb{K}$ over $\mathbb{F}$.
  - If $[\mathbb{K} : \mathbb{F}]$ is finite, $\mathbb{K}$ is called a *finite extension* of F. A finite extension is not necessarily a finite field: $\mathbb{C}$ is a finite extension of $\mathbb{R}$.
  - $\mathbb{F}[x]/\langle P(x)\rangle$ is an extension of degree h of $\mathbb{F}$, where $h = \deg(P)$.
  - When $\mathbb{F}_q$ is a finite field with q elements, $\mathbb{F}[x]/\langle P(x)\rangle$ has q^h elements.
  - If $|\mathbb{F}| = q$, and $\mathrm{char}(\mathbb{F}) = p$, then $q = p^m$ for some integer $m \geq 1$.

Finite Field Example

- $\mathbb{F} = \mathbb{F}_2,\ P(x) = x^3 + x + 1$. Let $[f(x)]$ represent the residue class $\{g(x) \in \mathbb{F}_2[x] \ : \ g(x) \equiv f(x) \pmod{P(x)}\}$.

Elements of $\mathbb{F}_8 = \mathbb{F}_2[x]/\langle P(x)\rangle$ and their inverses

element	*inverse*
0	–
1	1
$[x]$	$[x^2+1]$
$[x+1]$	$[x^2+x]$
$[x^2]$	$[x^2+x+1]$
$[x^2+1]$	$[x]$
$[x^2+x]$	$[x+1]$
$[x^2+x+1]$	$[x^2]$

Examples:

- $[x]\cdot[x^2+x] = [x^3+x^2] = [x^2+x+1]$
- $[x^2+1]\cdot[x^2] = [x^4+x^2]$ $= [x^2+x+x^2] = [x]$

Facts (for general $\mathbb{F}$ and $P(x)$:

- *The element $[x] \in \mathbb{F}[x]/\langle P(x)\rangle$ is a root of $P(x)$.*
- Denote $\alpha = [x]$. Then, $\mathbb{F}[x]/\langle P(x)\rangle$ is isomorphic to $\mathbb{F}(\alpha)$.
- If $\deg(P(x))=h$, then $\{1, \alpha, \alpha^2, \ldots, \alpha^{h-1}\}$ is a basis of $\mathbb{F}(\alpha)$ over $\mathbb{F}$.

Finite Field Example

- $\mathbb{F} = \mathbb{F}_2,\ P(x) = x^3 + x + 1$. Let $[f(x)]$ represent the residue class $\{g(x) \in \mathbb{F}_2[x] \ : \ g(x) \equiv f(x) \pmod{P(x)}\}$.

Elements of $\mathbb{F}_8 = \mathbb{F}(\alpha)$
and their inverses

element	*inverse*
0	–
1	1
α	$\alpha^2 + 1$
$\alpha + 1$	$\alpha^2 + \alpha$
α^2	$\alpha^2 + \alpha + 1$
$\alpha^2 + 1$	α
$\alpha^2 + \alpha$	$\alpha + 1$
$\alpha^2 + \alpha + 1$	α^2

Examples:

- $\alpha \cdot (\alpha^2 + \alpha) = \alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1$
- $\alpha^2 + 1 \cdot \alpha^2 = \alpha^4 + \alpha^2$
  $= \alpha^2 + \alpha + \alpha^2 = \alpha$

Facts (for general $\mathbb{F}$ and irreducible $P(x)$):

- *The element* $[x] \in \mathbb{F}[x]/\langle P(x)\rangle$ *is a root of* $P(x)$.
- Denote $\alpha = [x]$. Then, $\mathbb{F}[x]/\langle P(x)\rangle$ is isomorphic to $\mathbb{F}(\alpha)$.
- If $\deg(P(x))=h$, then $\{1, \alpha, \alpha^2, \ldots, \alpha^{h-1}\}$ is a basis of $\mathbb{F}(\alpha)$ over $\mathbb{F}$.

Roots of Polynomials

- **Proposition.** *A polynomial of degree $n \geq 0$ over a field $\mathbb{F}$ has at most n roots in any extension of $\mathbb{F}$.*
- **Proposition.** *Let $\mathbb{F}$ be a finite field. Then, $x^{|\mathbb{F}|} - x = \prod_{\beta \in \mathbb{F}} (x - \beta)$.*
- **Proposition.** *Let $\mathbb{F} = \mathbb{F}_q$, let $P(x)$ be an irreducible polynomial of degree h over $\mathbb{F}$. Let α be a root of $P(x)$. Then, $\alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{h-1}}$ are also roots of $P(x)$.*

  Proof. Recall that $a^q = a$ for all $a \in \mathbb{F}$. Thus, $0 = P(\alpha)^q = \left(\sum_{i=0}^{h} P_i \alpha^i\right)^q = \sum_{i=0}^{h} P_i^q \alpha^q = \sum_{i=0}^{h} P_i \alpha^q = P(\alpha^q)$. □

  - $\{ \alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{h-1}} \}$ is the set of *all* roots of P; therefore, $P(x) = \prod_{i=0}^{h-1} (x - \alpha^{q^i})$.
  - $\varphi : x \mapsto x^q$ is called a *Frobenius* mapping. $\{\varphi^i\}_{i=0}^{h-1}$ are *automorphisms* of $\mathbb{F}(\alpha)$ that fix $\mathbb{F}$. They form the *Galois group* of $[\mathbb{F}(\alpha) : \mathbb{F}]$.
  - $\mathbb{F}(\alpha)$ is the *splitting field* of $P(x)$.
  - $P(x)$ is the *minimal polynomial* of α.

Primitive Elements

- **Theorem.** *Let $\mathbb{F}$ be a finite field. Then, $\mathbb{F}^*$ is a* cyclic *group.* □

- A generator α of the cyclic group $\mathbb{F}^*$ is called a *primitive* element. For such an element, we have

$$\mathbb{F}^* = \{\, \alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{|\mathbb{F}|-2} \,\}.$$

  - $\mathcal{O}(\beta)$ will denote the multiplicative *order* of $\beta \in \mathbb{F}^*$.
  - clearly, if $|\mathbb{F}| = q$, then $\mathcal{O}(\beta) \mid (q-1)$, and, for a primitive element α, $\mathcal{O}(\alpha) = q-1$.
  - if $\beta = \alpha^k$ then $\mathcal{O}(\beta) = (q-1)/\gcd(q-1, k)$.
  - $\mathbb{F}$ has $\phi(q-1)$ primitive elements, where ϕ is the *Euler totient function*.

- Let $\mathbb{F}$ be a finite field, $P(x)$ an irreducible polynomial of degree h over $\mathbb{F}$, and α a root of $P(x)$. $P(x)$ is called a *primitive polynomial* if α is a primitive element of $\mathbb{F}(\alpha)$.
  - A primitive polynomial is irreducible.

Characterization of Finite Fields

Let $\mathbb{F}$ be a finite field with $|\mathbb{F}| = q$.

- $q = p^n$ for some prime p and integer $n \geq 1$.
  - p is the characteristic of F.
- Let $Q(x) = x^{q^h} - x$, $n \geq 1$. The roots of $Q(x)$ in its splitting field form an extension field $\mathbb{K}$ of $\mathbb{F}$, with $[\mathbb{K} : \mathbb{F}] = h$.

> *There is a finite field of size q for all q of the form $q = p^n$, p prime, $n \geq 1$. All finite fields of size q are isomorphic.*

The *unique* (up to isomorphism) field of size $q = p^n$ is denoted $\mathbb{F}_q$ or $\mathrm{GF}(q)$.

- There are irreducible polynomials and primitive polynomials of any degree ≥ 1 over $\mathbb{F}_q$.

Finite Fields: Summary

- There is a *unique* finite field $\mathbb{F}_q$, of size q, for each q of the form $q = p^m$, where p is prime and $m \geq 1$.
- When p is prime, $\mathbb{F}_p$ can be represented as the integers $\{0, 1, \ldots, p-1\}$ with arithmetic modulo p.
- When $q = p^m,\ m > 1$, $\mathbb{F}_q$ can be represented as $\mathbb{F}_p[x]_m$ (polynomials of degree $< m$ in $\mathbb{F}_p[x]$) with arithmetic modulo an irreducible polynomial $P(x)$ of degree m over $\mathbb{F}_p$: $\mathbb{F}_q \sim \mathbb{F}_p[x]/\langle P(x) \rangle$
  - $\mathbb{F}_q$ is an *extension* of degree m of $\mathbb{F}_p$
  - here, p can be a prime or itself a power of a prime
  - $P(x)$ has a root α in $\mathbb{F}_q$, $\alpha \sim [x] \in \mathbb{F}_p[x]_m$
  - $\alpha, \alpha^p, \alpha^{p^2}, \ldots, \alpha^{p^{m-1}}$ are *all* the roots of $P(x)$; all are in $\mathbb{F}_q$
  - *All* irreducible polynomials of degree m over $\mathbb{F}_p$ have *all* their roots in $\mathbb{F}_q$
- Every finite field $\mathbb{F}_q$ has a *primitive* element α: $\mathbb{F}_q = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{q^m-2}\}$
  - the minimal polynomial $P(x)$ of a primitive element α is a *primitive polynomial*
  - every primitive polynomial is irreducible, but *not every irreducible is primitive*

Finite Field Example: $\mathbb{F}_{16}$

α is a root of $P(x) = x^4 + x + 1 \in \mathbb{F}_2[x]$ (primitive).

i	α^i	*binary* 0123	*min poly*
–	0	0000	x
0	1	1000	$x+1$
1	α	0100	x^4+x+1
2	α^2	0010	x^4+x+1
3	α^3	0001	$x^4+x^3+x^2+x+1$
4	$\alpha+1$	1100	x^4+x+1
5	$\alpha^2+\alpha$	0110	x^2+x+1
6	$\alpha^3+\alpha^2$	0011	$x^4+x^3+x^2+x+1$
7	$\alpha^3+\alpha+1$	1101	x^4+x^3+1
8	α^2+1	1010	x^4+x+1
9	$\alpha^3+\alpha$	0101	$x^4+x^3+x^2+x+1$
10	$\alpha^2+\alpha+1$	1110	x^2+x+1
11	$\alpha^3+\alpha^2+\alpha$	0111	x^4+x^3+1
12	$\alpha^3+\alpha^2+\alpha+1$	1111	$x^4+x^3+x^2+x+1$
13	$\alpha^3+\alpha^2+1$	1011	x^4+x^3+1
14	α^3+1	1001	x^4+x^3+1

- if $\beta = \alpha^i$, $0 \leq i \leq (q-2)$, we say that i is the *discrete logarithm* of β to base α.

Examples:

- $(\alpha^2+\alpha)\cdot(\alpha^3+\alpha^2) = \alpha^5 \cdot \alpha^6 = \alpha^{11} = \alpha^3 + \alpha^2 + \alpha$
- $(\alpha^3+\alpha+1)^{-1} = \alpha^{-7} = \alpha^8 = \alpha^2 + 1$
- $\log_\alpha(\alpha^3+\alpha^2+1) = 13$

The Number of Irreducible Polynomials

- The *Möbius function* is defined as follows: let $n = \prod_{i=1}^{s} p_i^{e_i}$ be the prime factorization of $n \in \mathbb{Z}_{>0}$. Then,

$$\mu(n) = \begin{cases} 1 & \text{if } n = 1 \\ (-1)^s & \text{if } e_i = 1 \text{ for } 1 \le i \le s \\ 0 & \text{otherwise.} \end{cases}$$

Proposition. *Let $\mathcal{I}(n,q)$ denote the number of monic irreducible polynomials of degree n over $\mathbb{F}_q$. Then,*

$$\mathcal{I}(n,q) = \frac{1}{n} \sum_{m \mid n} \mu(m) q^{n/m} .$$

Proof. Let $\mathcal{F}_{m,q}(x) = \prod f(x), \quad \deg(f) = m, \ f(x)$ irreducible over $\mathbb{F}_q$.

$$x^{q^n} - x = \prod_{m|n} \mathcal{F}_{m,q} \quad \Longrightarrow \quad q^n = \sum_{m|n} m\mathcal{I}(m,q)$$

Use the Möbius inversion formula. □

Application: Double-Error Correcting Codes

- We can rewrite the PCM of the $[2^m-1, 2^m-1-m, 3]$ binary Hamming code $\mathcal{H}_m$ over $\mathbb{F}_2$ as
$$H = (\alpha_1 \ \alpha_2 \ \ldots \ \alpha_{2^m-1}) \ ,$$
where α_j ranges over all the nonzero elements of $\mathbb{F}_2$.

- **Example:** Let $m=4$ and α a root of $P(x)=x^4+x+1$. We take $\alpha_j=\alpha^{j-1}$, and
$$H_4 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} .$$

- A vector $\mathbf{c} = (c_1 \ c_2 \ \ldots \ c_n)$ is a codeword of $\mathcal{H}_m$ iff
$$H_m \mathbf{c}^T = \sum_{j=1}^{n} c_j \alpha_j = 0.$$

- If there is exactly one error, the error vector is of the form $\mathbf{e}_i = [0^{i-1} \, 1 \, 0^{n-i}]$, and the syndrome is $s = H_m \mathbf{e}_i^T = \alpha_j$. The syndrome gives us the error location directly.

Application: Double-Error Correcting Codes (II)

- What if there are two errors? Then, we get $\mathbf{e} = \mathbf{e}_i + \mathbf{e}_j$, and

$$s = \alpha_i + \alpha_j, \quad \text{for some } i, j,\ 1 \le i < j \le n,$$

which is insufficient to solve for α_i, α_j. *We need more equations ...*

- Consider the PCM

$$\hat{H}_m = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{2^m-1} \\ \alpha_1^3 & \alpha_2^3 & \dots & \alpha_{2^m-1}^3 \end{pmatrix} .$$

Syndromes are of the form

$$\mathbf{s} = \begin{pmatrix} s_1 \\ s_3 \end{pmatrix} = \hat{H}_m \mathbf{y}^T = \hat{H}_m \mathbf{e}^T .$$

Assume that the number of errors is at most 2.

- Case 1: $\mathbf{e} = 0$ (no errors). Then, $s_1 = s_3 = 0$.
- Case 2: $\mathbf{e} = \mathbf{e}_i$ for some $i,\ 1 \le i \le n$ (one error). Then,

$$\begin{pmatrix} s_1 \\ s_3 \end{pmatrix} = \hat{H}_m \mathbf{e}^T = \begin{pmatrix} \alpha_i \\ \alpha_i^3 \end{pmatrix} ;$$

namely, $s_3 = s_1^3 \neq 0$, and the error location is the index i such that $\alpha_i = s_1$.

Application: Double-Error Correcting Codes (III)

- Case 3: $\mathbf{e} = \mathbf{e}_i + \mathbf{e}_j$ for some $i, j, \quad 1 \leq i < j \leq n$ (two errors).

$$\begin{pmatrix} s_1 \\ s_3 \end{pmatrix} = \hat{H}_m \mathbf{e}^T = \begin{pmatrix} \alpha_i + \alpha_j \\ \alpha_i^3 + \alpha_j^3 \end{pmatrix} .$$

Since $s_1 = \alpha_i + \alpha_j \neq 0$, we can write

$$\frac{s_3}{s_1} = \frac{\alpha_i^3 + \alpha_j^3}{\alpha_i + \alpha_j} = \alpha_i^2 + \alpha_i \alpha_j + \alpha_j^2 .$$

Also,

$$s_1^2 = (\alpha_i + \alpha_j)^2 = \alpha_i^2 + \alpha_j^2 .$$

We add the two equations, and recall the definition of s_1 to obtain

$$\frac{s_3}{s_1} + s_1^2 = \alpha_i \alpha_j \; ; \qquad s_1 = \alpha_i + \alpha_j .$$

In particular, $\alpha_i \alpha_j \neq 0 \quad \Rightarrow \quad s_3 \neq s_1^3$, separating Case 3 from Cases 1–2. It follows that α_i and α_j are the roots of the following quadratic equation in x over $\mathbb{F}_{2^m}$:

$$x^2 + s_1 x + \frac{s_3}{s_1} + s_1^2 = 0 .$$

Overall, we have a decoding algorithm for up to two errors.

Two-error correcting BCH code.

Application: Double-Error Correcting Codes (IV)

- **Example:** As before, $\mathbb{F} = \mathbb{F}_{16}$, and α is a root of $P(x) = x^4 + x + 1$.

$$\hat{H}_4 = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} \end{pmatrix}$$

and, in binary form,

$$\hat{H}_4 = \left(\begin{array}{cccccccccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{array}\right).$$

For this code, we know

- $k \geq 15 - 8 = 7$ (in fact, the dimension is exactly 7)
- $d \geq 5$ (in fact, $d = 5$)

Variations on the Double-error Correcting Code

- Add an overall parity bit

$$\hat{H}_4 = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} & 0 \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

  For this code, we know

  - $n = 16$
  - $k = 7$ (same number of words)
  - $d = 6$
  - corrects 2 errors, detects 3

- Expurgate words of odd weight

$$\bar{H}_4 = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

  - $n = 15,\ k = 6,\ d = 6$: corrects 2 errors, detects 3

5. Reed-Solomon Codes

Generalized Reed-Solomon Codes

- Let $\alpha_1, \alpha_2, \ldots, \alpha_n$, $n < q$, be distinct nonzero elements of $\mathbb{F}_q$, and let $v_1, v_2, \ldots, v_n$ be *nonzero* elements of $\mathbb{F}_q$ (not necessarily distinct). A *generalized Reed-Solomon (GRS)* code is a linear $[n, k, d]$ code $\mathcal{C}_{\mathrm{GRS}}$ with PCM

$$H_{\mathrm{GRS}} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \ldots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \ldots & \alpha_n^{n-k-1} \end{pmatrix} \begin{pmatrix} v_1 & & & \\ & v_2 & & 0 \\ & & \ddots & \\ 0 & & & v_n \end{pmatrix}.$$

α_j: *code locators* (distinct), v_j:*column multipliers* ($\neq 0$)

Reminder. $\mathcal{C}_{\mathrm{GRS}}$ is an MDS code, namely, $d = n - k + 1$.

Theorem. *The dual of a GRS code is a GRS code.*

Proof. Show that G_{GRS} can have the same form as H_{GRS}, with k rows, the same locators, and a different choice of multipliers $\{v_i'\}$. □

Distinguished Classes of GRS Codes

- *Primitive GRS codes:* $n = q-1$ and $\{\alpha_1, \alpha_2, \ldots, \alpha_n\} = F^*$; usually $\alpha_i = \alpha^{i-1}$ for a primitive $\alpha \in \mathbb{F}$.
- *Normalized GRS codes:* $v_j = 1$ for all $1 \le j \le n$.
- *Narrow-sense GRS codes:* $v_j = \alpha_j$ for all $1 \le j \le n$.
- Allowing one $\alpha_i = 0$ (column $[1\,0\,\ldots\,0]^T$, not in narrow sense GRS):
  (singly) extended GRS code $\Rightarrow n \le q$
- Allowing one $\alpha_i = \infty$ (column $[0\,\ldots\,0\,1]^T$, not in narrow sense GRS):
  (doubly) extended GRS code $\Rightarrow n \le q+1$

Example: Let $v_1, v_2, \ldots, v_n$ be the column multipliers of a primitive GRS code. We verify that the dual GRS code has column multipliers α_j/v_j. Let α be a primitive element of $\mathbb{F}$. For $v'_j = \alpha_j/v_j$, a typical entry in the postulated $G \cdot H^T$ is

$$G_i \cdot H_h^T = \sum_{\ell=1}^{n} v_\ell v'_\ell \alpha_\ell^{i+h} = \sum_{\ell=1}^{n} \alpha_\ell^{i+h+1} = \sum_{\ell=1}^{n} \alpha^{(\ell-1)(i+h+1)} = \frac{\alpha^{n(i+h+1)} - 1}{\alpha^{i+h+1} - 1} = 0$$

with $0 \le i \le k-1,\ 0 \le h \le n-k-1$ and, thus, $0 \le i+h+1 \le n-1$ (recall that $\mathcal{O}(\alpha) = n$).

$$\Rightarrow \text{ (normalized primitive GRS)}^{\perp} = \text{ (narrow-sense primitive GRS)}.$$

GRS Encoding as Polynomial Evaluation

- For $\mathbf{u} = (u_0\, u_1\, \ldots\, u_{k-1}\,)$, let $u(x) = u_0 + u_1 x + u_2 x^2 + \cdots + u_{k-1} x^{k-1}$. Then,

$$\begin{aligned}
\mathbf{c} &= \mathbf{u}\, G_{\mathrm{GRS}} = \mathbf{u} \cdot \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \ldots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \ldots & \alpha_n^{k-1} \end{pmatrix} \begin{pmatrix} v_1' & & & \\ & v_2' & & 0 \\ & & \ddots & \\ 0 & & & v_n' \end{pmatrix} \\
&= [\, v_1' u(\alpha_1)\; v_2' u(\alpha_2)\; \ldots\; v_n' u(\alpha_n)\,]
\end{aligned}$$

- Minimum distance now follows from the fact that a polynomial of degree $\leq k-1$ cannot have more than $k-1$ roots in $\mathbb{F}_q \Rightarrow \mathsf{wt}(\mathbf{c}) \geq n-k+1$.

- Decoding as *noisy interpolation*: reconstruct $u(x)$ from $(k+2t)$ noisy evaluations $u(\alpha_1)+e_1,\, u(\alpha_2)+e_2, \ldots, u(\alpha_{k+2t})+e_{k+2t}$, possible if at most t evaluations are corrupted.

Conventional Reed-Solomon Codes

- *Conventional Reed-Solomon (RS)* code: GRS code with $n|(q-1)$, $\alpha \in \mathbb{F}^*$ with $\mathcal{O}(\alpha) = n$,

$$\begin{aligned} \alpha_j &= \alpha^{j-1}, \quad 1 \le j \le n, \\ v_j &= \alpha^{b(j-1)}, \quad 1 \le j \le n. \end{aligned}$$

- *Canonical PCM* of a RS code is given by

$$H_{\rm RS} = \begin{pmatrix} 1 & \alpha^b & \dots & \alpha^{(n-1)b} \\ 1 & \alpha^{b+1} & \dots & \alpha^{(n-1)(b+1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{b+d-2} & \dots & \alpha^{(n-1)(b+d-2)} \end{pmatrix} \qquad (\text{\# rows} = d-1 = n-k)$$

- $\mathbf{c} \in \mathcal{C}_{\rm RS} \iff H_{\rm RS}\mathbf{c}^T = \mathbf{0} \iff c(\alpha^\ell) = 0, \ \ell = b, b+1, \dots, b+d-2.$
- $\alpha^b, \alpha^{b+1}, \dots, \alpha^{b+d-2}$: *roots* of $\mathcal{C}_{\rm RS}$
- $g(x) = (x-\alpha^b)(x-\alpha^{b+1})\cdots(x-\alpha^{b+d-2})$: *generator polynomial* of $\mathcal{C}_{\rm RS}$

RS Codes as Cyclic codes (another polynomial characterization)

- $\mathbf{c} \in \mathcal{C}_{\mathrm{RS}} \iff c(\alpha^{\ell}) = 0, \quad \ell = b, b+1, \ldots, b+d-2$
- $g(x) = (x - \alpha^{b})(x - \alpha^{b+1}) \cdots (x - \alpha^{b+d-2})$

  Therefore, $\mathbf{c} \in \mathcal{C}_{\mathrm{RS}} \iff g(x) | c(x)$ and

$$\mathcal{C}_{\mathrm{RS}} = \{\, u(x)g(x) \;:\; \deg(u) < k \,\} \subseteq \mathbb{F}_q[x]_n$$

- Every root of $g(x)$ is also a root of $x^n - 1 \quad \Rightarrow \quad g(x) \,|\, x^n - 1$.
- $\mathcal{C}_{\mathrm{RS}}$ is the *ideal* generated by $g(x)$ in the ring $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$
- RS codes are *cyclic*: $c(x) \in \mathcal{C}_{\mathrm{RS}} \Rightarrow xc(x) \bmod (x^n - 1) \in \mathcal{C}_{\mathrm{RS}}$, or

$$\mathbf{c} = [\, c_0\, c_1\, \ldots\, c_{n-1} \,] \in \mathcal{C}_{\mathrm{RS}} \Rightarrow [\, c_{n-1}\, c_0\, c_1\, \ldots\, c_{n-2} \,] \in \mathcal{C}_{\mathrm{RS}}$$

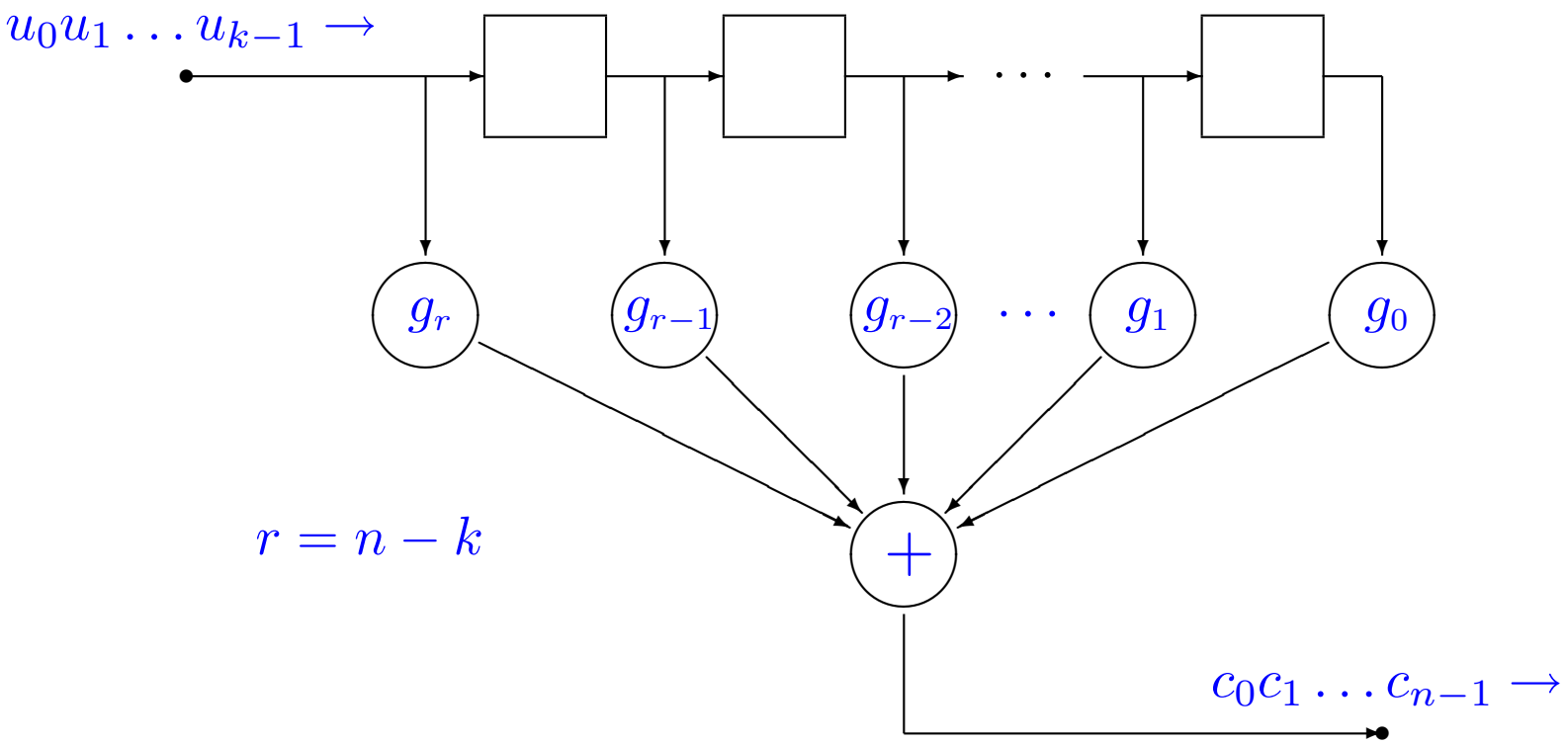
- Distinguished RS codes
  - Primitive RS: $n = q - 1$, α primitive element of $\mathbb{F}_q$
  - Narrow-sense RS: $b = 1$
  - Normalized RS: $b = 0$

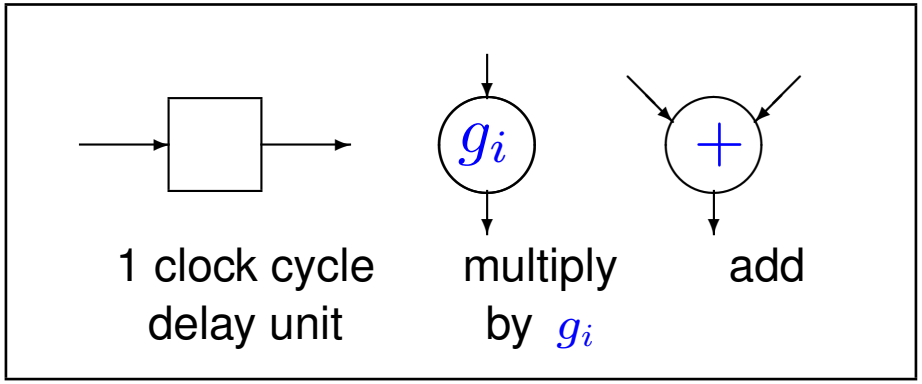
Encoding RS codes

- We can encode GRS codes as any linear code: $\mathbf{u} \mapsto \mathbf{u}\, G_{\mathrm{GRS}}$
- In the polynomial interpretation of RS codes: $u(x) \mapsto u(x)g(x)$, corresponding to a non-systematic generator matrix

$$G = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & & \\ & g_0 & g_1 & \cdots & g_{n-k} & \\ 0 & & \ddots & \ddots & \cdots & \ddots \\ & & & g_0 & g_1 & \cdots & g_{n-k} \end{pmatrix} \qquad (g_{n-k} = 1)$$

Non-systematic Encoding Circuit





Systematic Encoding of RS Codes

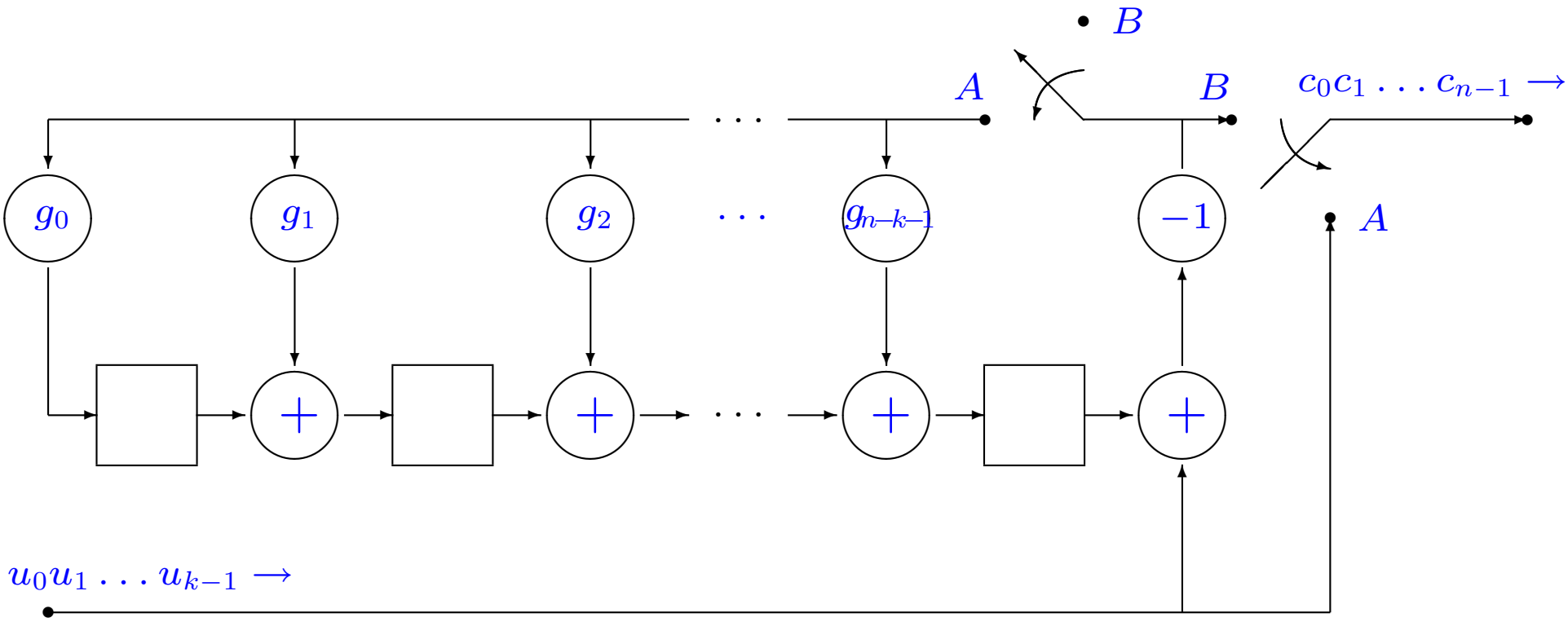
- For $u(x) \in \mathbb{F}_q[x]_k$, let $r_u(x)$ be the unique polynomial in $\mathbb{F}_q[x]_{n-k}$ such that
$$r_u(x) \equiv x^{n-k}u(x) \pmod{g(x)}$$

- Clearly, $x^{n-k}u(x) - r_u(x) \in \mathcal{C}_{\mathrm{RS}}$

- The mapping $\mathcal{E}_{\mathrm{RS}} : u(x) \mapsto x^{n-k}u(x) - r_u(x)$ is a *linear, systematic* encoding for $\mathcal{C}_{\mathrm{RS}}$

$$\begin{array}{r}
\left[\begin{array}{cccccccc} u_{k-1} & u_{k-2} & \dots & u_0 & 0 & 0 & \dots & 0 \end{array}\right] \\
-\left[\begin{array}{cccccccc} 0 & 0 & \dots & 0 & r_{n-k-1} & r_{n-k-2} & \dots & r_0 \end{array}\right] \\
\hline
\left[\begin{array}{cccccccc} c_{n-1} & c_{n-2} & \dots & c_{n-k} & c_{n-k-1} & c_{n-k-2} & \dots & c_0 \end{array}\right]
\end{array}$$

Systematic Encoding Circuit



- Switches
  - at A for k cycles
  - at B for $n-k$ cycles

- Register contents: $R_0(x) = 0$;

$$\begin{aligned} R_\ell(x) &= xR_{\ell-1}(x) + x^r u_{k-\ell} \\ &= x^{n-k} \sum_{i=1}^{\ell} u_{k-i} x^{\ell-i} \bmod g(x) \\ \ell &= 1, 2, \ldots, k\,. \end{aligned}$$

6. Decoding Generalized Reed-Solomon Codes

Decoding Generalized Reed-Solomon Codes

- We consider $\mathcal{C}_{\mathrm{GRS}}$ over $\mathbb{F}_q$ with PCM

$$H_{\mathrm{GRS}} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{d-2} & \alpha_2^{d-2} & \dots & \alpha_n^{d-2} \end{pmatrix} \begin{pmatrix} v_1 & & & \\ & v_2 & & 0 \\ & & \ddots & \\ 0 & & & v_n \end{pmatrix}$$

  with $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{F}_q^*$ distinct, and $v_1, v_2, \dots, v_n \in \mathbb{F}_q^*$

- Codeword $\mathbf{c}$ transmitted, word $\mathbf{y}$ received, with error vector

$$\mathbf{e} = (e_1\ e_2\ \dots\ e_n) = \mathbf{y} - \mathbf{c}$$

- $J = \{\kappa\ :\ e_\kappa \neq 0\}$ set of *error locations*

- We describe and algorithm that correctly decodes $\mathbf{y}$ to $\mathbf{c}$, under the assumption $|J| \leq \frac{1}{2}(d-1)$.

Syndrome Computation

- First step of the decoding algorithm

$$\mathbf{S} = \begin{pmatrix} S_0 \\ S_1 \\ \vdots \\ S_{d-2} \end{pmatrix} = H_{\mathrm{GRS}}\mathbf{y}^T = H_{\mathrm{GRS}}\mathbf{e}^T$$

$$S_\ell = \sum_{j=1}^{n} y_j v_j \alpha_j^\ell = \sum_{j=1}^{n} e_j v_j \alpha_j^\ell = \sum_{j\in J} e_j v_j \alpha_j^\ell \,, \quad \ell = 0, 1, \ldots, d-2$$

Example: For RS codes, we have $\alpha_j = \alpha^{j-1}$ and $v_j = \alpha^{b(j-1)}$, so

$$S_\ell = \sum_{j=1}^{n} y_j \alpha^{(j-1)(b+\ell)} = y(\alpha^{b+\ell}) \,, \quad \ell = 0, 1, \ldots, d-2 \,.$$

- *Syndrome polynomial*:

$$S(x) = \sum_{\ell=0}^{d-2} S_\ell x^\ell = \sum_{\ell=0}^{d-2} x^\ell \sum_{j\in J} e_j v_j \alpha_j^\ell = \sum_{j\in J} e_j v_j \sum_{\ell=0}^{d-2} (\alpha_j x)^\ell \,.$$

A Congruence for the Syndrome Polynomial

$$S(x) = \sum_{j \in J} e_j v_j \sum_{\ell=0}^{d-2} (\alpha_j x)^\ell \ .$$

- We have

$$\sum_{\ell=0}^{d-2} (\alpha_j x)^\ell \equiv \frac{1}{1 - \alpha_j x} \pmod{x^{d-1}}$$

$$\Rightarrow \quad \boxed{S(x) = \sum_{j \in J} \frac{e_j v_j}{1 - \alpha_j x} \pmod{x^{d-1}}} \qquad \left(\sum_\phi \square = 0 \right)$$

More Auxiliary Polynomials

- *Error locator polynomial (ELP)*

$$\Lambda(x) = \prod_{j \in J} (1 - \alpha_j x) \qquad \left(\prod_{\phi} \square \triangleq 1 \right)$$

- *Error evaluator polynomial (EEP)*

$$\Gamma(x) = \sum_{j \in J} e_j v_j \prod_{m \in J \setminus \{j\}} (1 - \alpha_m x)$$

- $\Lambda(\alpha_\kappa^{-1}) = 0 \qquad \Longleftrightarrow \qquad \kappa \in J$ *roots of EEP point to error locations*
- $\Gamma(\alpha_\kappa^{-1}) = e_\kappa v_\kappa \prod_{m \in J \setminus \{\kappa\}} (1 - \alpha_m \alpha_\kappa^{-1}) \neq 0 \quad \Rightarrow \boxed{\gcd(\Lambda(x), \Gamma(x)) = 1}$
- The degrees of ELP and EEP satisfy

$$\deg \Lambda = |J| \quad \text{and} \quad \deg \Gamma < |J|$$

Of course, we don't know $\Lambda(x)$, $\Gamma(x)$: our goal is to find them

Key Equation of GRS Decoding

- Since $|J| \leq \frac{1}{2}(d-1)$, we have

(1) $\boxed{\deg \Lambda \leq \frac{1}{2}(d-1)}$ and (2) $\boxed{\deg \Gamma < \frac{1}{2}(d-1)}$

- The ELP and the EEP are related by

$$\Gamma(x) = \sum_{j \in J} e_j v_j \prod_{m \in J \setminus \{j\}} (1 - \alpha_m x) = \sum_{j \in J} e_j v_j \frac{\Lambda(x)}{1 - \alpha_j x} = \Lambda(x) \sum_{j \in J} \frac{e_j v_j}{1 - \alpha_j x}$$

$$\Longrightarrow \text{(3)} \quad \boxed{\Lambda(x) S(x) \equiv \Gamma(x) \pmod{x^{d-1}}}$$

(1)+(2)+(3): *key equation of GRS decoding*

- (3) is a set of $d-1$ *linear* equations in the coefficients of Λ and Γ
- $\lfloor \frac{1}{2}(d-1) \rfloor$ equations depend only on Λ (corresponding to x^i, $i \geq \frac{1}{2}(d-1)$)
- we can solve for Λ, find its root set J, and then solve *linear* equations for e_j
- straightforward solution leads to $O(d^3)$ algorithm — we'll present an $O(d^2)$ one

The Extended Euclidean Algorithm

- Euclid for polynomials: given $a(x), b(x)$ over a field $\mathbb{F}$, with $a(x) \neq 0$ and $\deg a > \deg b$, the algorithm computes
  - *remainders* $r_i(x)$, *quotients* $q_i(x)$, and *auxiliary polynomials* $s_i(x)$ and $t_i(x)$

$r_{-1}(x) \leftarrow a(x)$; $r_0(x) \leftarrow b(x)$;
$s_{-1}(x) \leftarrow 1$; $s_0(x) \leftarrow 0$;
$t_{-1}(x) \leftarrow 0$; $t_0(x) \leftarrow 1$;
for ($i \leftarrow 1$; $r_{i-1}(x) \neq 0$; i++) {
  $q_i(x) \leftarrow r_{i-2}(x)$ div $r_{i-1}(x)$;
  $r_i(x) \leftarrow r_{i-2}(x) - q_i(x) r_{i-1}(x)$;
  $s_i(x) \leftarrow s_{i-2}(x) - q_i(x) s_{i-1}(x)$;
  $t_i(x) \leftarrow t_{i-2}(x) - q_i(x) t_{i-1}(x)$;
}

  - Let $\nu =$ largest i such that $r_i \neq 0$. Then, $r_\nu(x) = \gcd(a(x), b(x))$
  - We also know that $s_\nu(x)a(x) + t_\nu(x)b(x) = \gcd(a(x), b(x))$ (often used to compute modular inverses)
  - This is a special case of a more general relation between the sequences

Properties of the Euclidean Algorithm Sequences

Proposition. *The following relations hold:*

(i) *For* $i = -1, 0, \ldots, \nu + 1$*:* $\quad s_i(x)a(x) + t_i(x)b(x) = r_i(x)$

(ii) *For* $i = 0, 1, \ldots, \nu + 1$*:* $\quad \deg t_i + \deg r_{i-1} = \deg a$

Proof. By induction on i. □

Proposition. *Suppose that* $t(x), r(x) \in \mathbb{F}[x] \setminus \{0\}$ *satisfy the following conditions:*

(C1) $\gcd(t(x), r(x)) = 1$
(C2) $\deg t + \deg r < \deg a$
(C3) $t(x)b(x) \equiv r(x) \pmod{a(x)}$

Then, for some $h \in \{0, 1, \ldots, \nu+1\}$ *and a constant* $c \in \mathbb{F}$*, we have*

$$t(x) = c \cdot t_h(x) \quad \textit{and} \quad r(x) = c \cdot r_h(x) \; .$$

Proof. Standard polynomial manipulations, and observing that the sequence $\deg r_i$ is strictly decreasing. □

Solving the Key Equation

- Apply the Euclidean algorithm with
  $a(x) = x^{d-1}$ and $b(x) = S(x)$, to produce
  $\Lambda(x) = c \cdot t_h(x)$ and $\Gamma(x) = c \cdot r_h(x)$
  [the key equation guarantees conditions (C1)–(C3)].

  How do we find h —the stopping index?

Theorem. *The solution to the key equation is unique up to a scalar constant, and it is obtained with the Euclidean algorithm by stopping at the* unique *index* h *such that*

$$\deg r_h < \tfrac{1}{2}(d-1) \leq \deg r_{h-1}$$

Proof. Such an h exists because r_i is strictly decreasing. The degree properties follow from the propositions. □

Finding the Error Values

- *Formal derivatives* in finite fields: $\left[\sum_{i=0}^{s} a_i x^i\right]' = \sum_{i=1}^{s} i a_i x^{i-1}$

  $(a(x)b(x))' = a'(x)b(x) + a(x)b'(x)$ (not surprising)

- For the ELP, we have

$$\Lambda(x) = \prod_{j \in J}(1 - \alpha_j x) \quad \Rightarrow \quad \Lambda'(x) = \sum_{j \in J}(-\alpha_j) \prod_{m \in J \setminus \{j\}} (1 - \alpha_m x)\,,$$

  and, for $\kappa \in J$,

$$\Lambda'(\alpha_\kappa^{-1}) = -\alpha_\kappa \prod_{m \in J \setminus \{\kappa\}} (1 - \alpha_m \alpha_\kappa^{-1}) \quad \text{and} \quad \Gamma(\alpha_\kappa^{-1}) = e_\kappa v_\kappa \prod_{m \in J \setminus \{\kappa\}} (1 - \alpha_m \alpha_\kappa^{-1})$$

- Therefore, for all error locations $\kappa \in J$, we obtain

$$e_\kappa = -\frac{\alpha_\kappa}{v_\kappa} \cdot \frac{\Gamma(\alpha_\kappa^{-1})}{\Lambda'(\alpha_\kappa^{-1})}$$

Forney's algorithm for error values

Summary of GRS Decoding

Input: received word $(y_1\ y_2\ \ldots\ y_n) \in \mathbb{F}_q^n$.

Output: error vector $(e_1\ e_2\ \ldots\ e_n) \in \mathbb{F}_q^n$.

1. *Syndrome computation:* Compute the polynomial $S(x) = \sum_{\ell=0}^{d-2} S_\ell x^\ell$ by

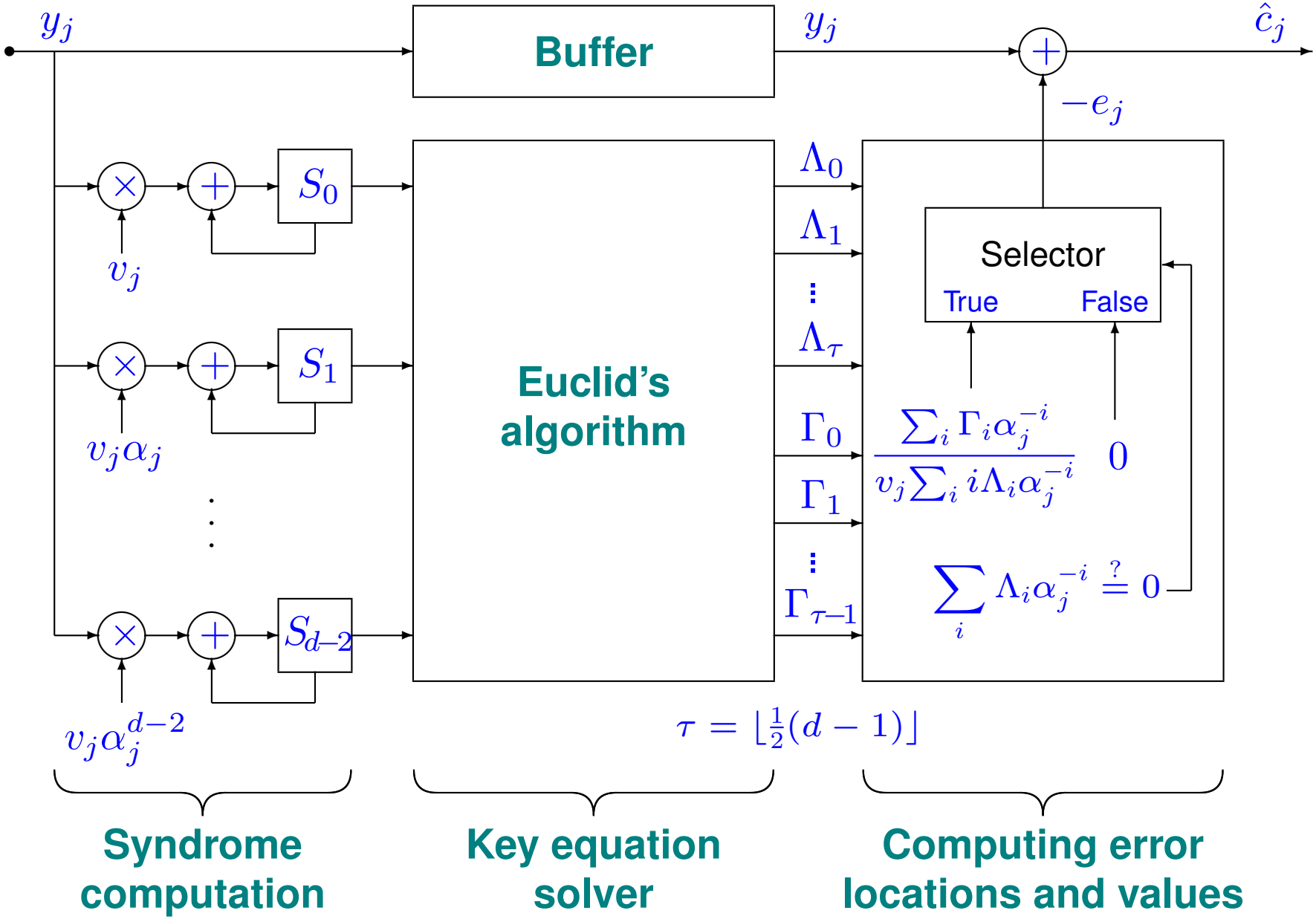
$$S_\ell = \sum_{j=1}^{n} y_j v_j \alpha_j^\ell \ , \quad \ell = 0, 1, \ldots, d-2 \ .$$

2. *Solving the key equation:* Apply Euclid's algorithm to $a(x) \leftarrow x^{d-1}$ and $b(x) \leftarrow S(x)$ to produce $\Lambda(x) \leftarrow t_h(x)$ and $\Gamma(x) \leftarrow r_h(x)$, where h is the smallest index i for which $\deg r_i < \frac{1}{2}(d-1)$.
3. *Forney's algorithm:* Compute the error locations and values by

$$e_j = \begin{cases} -\dfrac{\alpha_j}{v_j} \cdot \dfrac{\Gamma(\alpha_j^{-1})}{\Lambda'(\alpha_j^{-1})} & \text{if } \Lambda(\alpha_j^{-1}) = 0 \\ 0 & \text{otherwise} \end{cases} \quad , \quad j = 1, 2, \ldots, n \ .$$

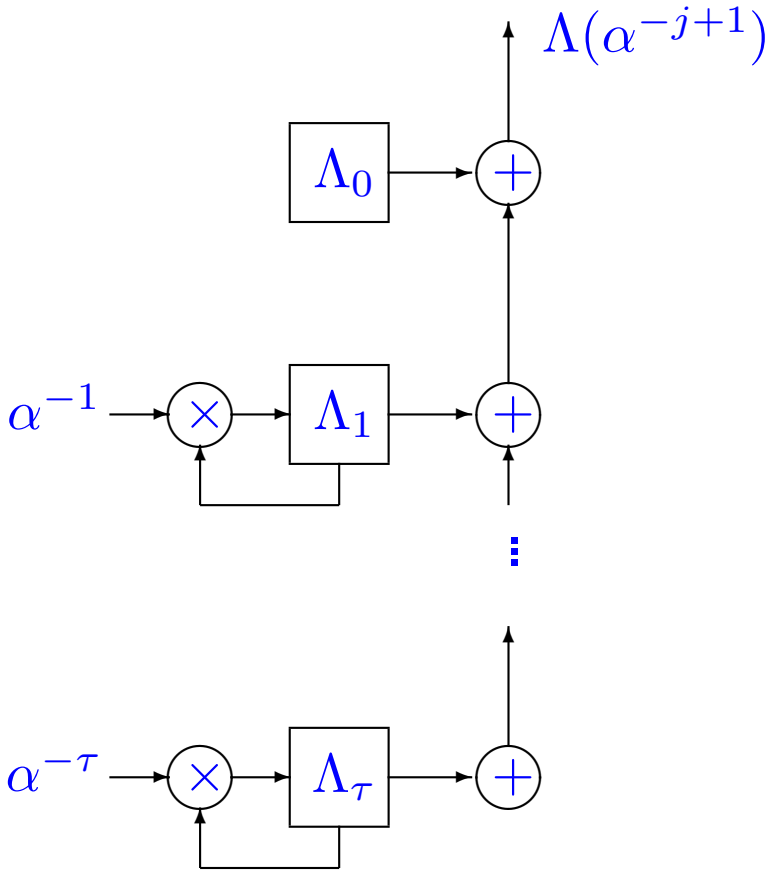
Complexity: 1. $O(dn)$ 2. $O(|J+1|d)$ 3. $O(|J+1|n)$

Schematic for GRS Decoder



Finding Roots of the ELP (RS Codes)

Chien search for RS codes ($\alpha_j = \alpha^{j-1},\ 1 \le j \le n$)



At clock cycle # j, the cell labeled Λ_i contains

$\Lambda_i \alpha^{-i(j-1)},\ 0 \le i \le \tau,\ 1 \le j \le n$

RS Decoding Example

Example

Other Decoding Algorithms

- *Peterson-Gorenstein-Zierler* [1960]: First algorithm for solving the key equation by solving a system of linear equations on $\{\Lambda_i\}$ in terms of $\{S_j\}$ — iterates on the number of errors t, complexity $O(d^4)$.

- *Berlekamp algorithm* [1967] (also referred to as *Berlekamp-Massey* due to a clearer description and improvements by Massey [1969]): first efficient solution of the key equation, using Newton's identities and solving for shortest recurrence that generates the syndrome sequence. Complexity comparable to the Euclidean algorithm.

- *Welch-Berlekamp* [1986]: Solves key equation starting from *remainder syndrome* $y(x) \pmod{g(x)}$, without computing power sums. Akin to continued fractions and Padé approximations. Suitable for soft-decoding.

- *List decoding* Decodes beyond $\tau = \lfloor \frac{1}{2}(d-1) \rfloor$ errors, producing a list of candidate decoded codewords. Very often, the coset leader is unique even beyond τ. Dates back to the '50s, but has gotten recent focus due to elegant and efficient algorithms by Sudan ['97] and others for GRS and AG codes.

- Many other variants: time/frequency interpretations, time-domain decoding [Blahut '83], soft decision etc. ...

7. Codes Related to GRS Codes

Alternant Codes

- Let $\mathbb{F} = \mathbb{F}_q$ and let $\mathcal{C}_{\rm GRS}$ be an $[N, K, D]$ code over $\Phi = \mathbb{F}_{q^m}$. The set of codewords of $\mathcal{C}_{\rm GRS}$ with coordinates in $\mathbb{F}$, is called an *alternant code*, $\mathcal{C}_{\rm alt} = \mathcal{C}_{\rm GRS} \cap \mathbb{F}^N$. For a PCM $H_{\rm GRS}$ of $\mathcal{C}_{\rm GRS}$, we have

$$\mathbf{c} \in \mathcal{C}_{\rm alt} \quad \Longleftrightarrow \quad \mathbf{c} \in \mathbb{F}^N \text{ and } H_{\rm GRS}\mathbf{c}^T = \mathbf{0}.$$

  This is also called a *sub-field sub-code*.∎

- Let $[n, k, d]$ be the parameters of $\mathcal{C}_{\rm alt}$. Clearly, $n = N$, and $d \geq D$; D is called the *designed distance*.

  Each row of $H_{\rm GRS}$ translates to $\leq m$ independent rows over $\mathbb{F}$, so

$$n - k \leq (N - K)m = (D - 1)m \quad \Rightarrow \quad k \geq n - (D - 1)m$$

  ∎

- Decoding: can be done with the same algorithm that decodes $\mathcal{C}_{\rm GRS}$.

Binary Narrow-Sense Alternant Codes

- Consider $F = \mathbb{F}_2$ and $\mathcal{C}_{\rm GRS}$ over $\mathbb{F}_{2^m}$, with *odd* D and $N \leq 2^m - 1$. For $\mathbf{c} \in \mathbb{F}_2^N$,

$$\mathbf{c} \in \mathcal{C}_{\rm alt} \quad \Longleftrightarrow \quad \sum_{j=1}^{n} c_j \alpha_j^i = 0 \quad \text{for } i = 1, 2, 3, \ldots, D-1 \ .$$

  Over $\mathbb{F}_2$,

$$\sum_{j=1}^{n} c_j \alpha_j^i = 0 \quad \Longleftrightarrow \quad \sum_{j=1}^{n} c_j \alpha_j^{2i} = 0$$

  Therefore, check equations for even values of i are dependent, and the redundancy bound can be improved to

$$n - k \leq \frac{(D-1)m}{2} \ .$$

- A more compact PCM for $\mathcal{C}_{\rm alt}$:

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^3 & \alpha_2^3 & \ldots & \alpha_n^3 \\ \alpha_1^5 & \alpha_2^5 & \ldots & \alpha_n^5 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{D-2} & \alpha_2^{D-2} & \ldots & \alpha_n^{D-2} \end{pmatrix}$$

- Decoding: same as $\mathcal{C}_{\rm GRS}$, but *error values* not needed $\Rightarrow$ simpler key equation alg.

BCH Codes

- *Bose-Chaudhuri-Hocquenghem (BCH)* codes are alternant codes that correspond to conventional RS codes, i.e., for $\mathcal{C}_{\rm RS} : [N, K, D]$ over $\mathbb{F}_{q^m}$, $\mathcal{C}_{\rm BCH} = \mathbb{F}_q^N \cap \mathcal{C}_{\rm RS}$.
- N must divide $q^m - 1 \quad \Rightarrow \quad \gcd(N, q) = 1$. Conversely, if $\gcd(N, q) = 1$, then q belongs to the multiplicative group of the integers modulo N. So, given a code length $n = N$, the smallest possible value of m is the order of q in the multiplicative group modulo N.

Summary of BCH code definition

- Code of length $n \geq 1$ over $\mathbb{F}_q$ such that $\gcd(n, q) = 1$
- $m \geq 1$: [smallest] positive integer s.t. $n | q^m - 1$
- $\alpha \in \mathbb{F}_{q^m}$: element of multiplicative order n
- $D > 0$, b: design parameters

$$\mathcal{C}_{\rm BCH} = \left\{ c(x) \in (\mathbb{F}_q)_n[x] \, : \, c(\alpha^\ell) = 0, \ \ell = b, b+1, \ldots, b+D-2 \right\}$$

BCH Code Example

Example: We design a BCH code of length $n = 85$ over $\mathbb{F}_2$ that can correct 3 errors.

- $m =$ smallest positive integer s.t. $85 | 2^m - 1 \Rightarrow m = 8$.
- $b = 1 \Rightarrow$ narrow-sense
- $D = 7 \Rightarrow$ 3-error correcting
- $n - k \leq (D-1)m/2 = 24$
- resulting $\mathcal{C}_{\rm BCH}$ is $[85, \geq 61, \geq 7]$ over $\mathbb{F}_2$
- Let γ be a primitive element of $\Phi = \mathbb{F}_{2^8}$, $\alpha = \gamma^3$, so that $\mathcal{O}(\alpha) = 85$.
- a 24×85 *binary* PCM of the code can be obtained by representing the entries in H_Φ below as column vectors in $\mathbb{F}_2^8$.

$$H_\Phi = \begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^j & \dots & \alpha^{83} & \alpha^{84} \\ 1 & \alpha^3 & \alpha^6 & \dots & \alpha^{3j} & \dots & \alpha^{79} & \alpha^{82} \\ 1 & \alpha^5 & \alpha^{10} & \dots & \alpha^{5j} & \dots & \alpha^{75} & \alpha^{80} \end{pmatrix}$$

- Let γ be a root of $p(x) = x^8 + x^6 + x^5 + x + 1$ (primitive polynomial). Represent $\mathbb{F}_{2^8} = \mathbb{F}_2(\gamma) \equiv \mathbb{F}_2[x]/\langle p(x) \rangle$. Then,

$$\begin{array}{lllll} \alpha & = & \gamma^3 & & \leftrightarrow [00010000]^T \\ \alpha^3 & = & \gamma^9 & = \gamma + \gamma^2 + \gamma^6 + \gamma^7 & \leftrightarrow [01100011]^T \\ \alpha^5 & = & \gamma^{15} & = \gamma^3 + \gamma^6 + \gamma^7 & \leftrightarrow [00010011]^T \end{array}$$

The second column in $H_{\mathbb{F}_2}$ will be $[00010000\,01100011\,00010011]^T$

BCH Code Example (continued)

- A codeword $\mathbf{c} \in \mathcal{C}_{\mathrm{BCH}}$ satisfies $c(\alpha) = 0$. Therefore,

$$0 = c(\alpha)^2 = \left(\sum_{i=0}^{n-1} c_i x^i\right)^2 = \sum_{i=0}^{n-1} c_i^2 x^{2i} = \sum_{i=0}^{n-1} c_i x^{2i} = c(\alpha^2).$$

  For the same reason, $c(\alpha)=c(\alpha^2)=c(\alpha^4)=c(\alpha^8)=\ldots=c(\alpha^{128})=0 \quad \Rightarrow \quad M_1(x)$, the minimal polynomial of α, divides $c(x)$.

- Similarly for $M_3(x)$, the min. poly. of $alpha^3$, and $M_5(x)$, the min. poly. of α^5.
- Let $g(x) = M_1(x)M_3(x)M_5(x)$. Then,

$$\mathbf{c} \in \mathcal{C}_{\mathrm{BCH}} \quad \Leftrightarrow \quad g(x)|c(x).$$

- $g(x)$ is the *generator polynomial of* $\mathcal{C}_{\mathrm{BCH}}$, which is presented as a *cyclic binary code*.
- In the example, $M_1(x)=x^8+x^7+x^3+x+1$, $M_3(x)=x^8+x^7+x^6+x^4+x^3+x^2+1$, $M_5(x)=x^8+x^5+x^4+x^3+1$.

  $\Rightarrow \; g(x) = x^{24} + x^{17} + x^{16} + x^{15} + x^{14} + x^{12} + x^{11} + x^8 + x^6 + x^5 + x^2 + x + 1.$

BCH Code Example (continued)

- As with RS codes, we have the polynomial (cyclic) interpretation of BCH codes: $u(x) \mapsto c(x)=u(x)g(x)$, with $u(x) \in \mathbb{F}_2[x]$ (a binary polynomial of degree $< k$), corresponding to a non-systematic *binary* generator matrix

$$G = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & & \\ & g_0 & g_1 & \cdots & g_{n-k} & \\ 0 & & \ddots & \ddots & \cdots & \ddots \\ & & & g_0 & g_1 & \cdots & g_{n-k} \end{pmatrix} \qquad (g_{n-k} = 1)$$

- In the example, this representation also implies that $k_{BCH} = 85 - 24 = 61$, the rank of G.
- Codes with dimension better than the bound are obtained when some of the minimal polynomials M_i are of degree less than m.
- such a code would be obtained, for example, if we changed the parameter b so that α^{17} were one of the roots. In this case (no longer a *narrow sense* code), we have

$$O(\alpha^{17}) = 5 \;\Rightarrow\; M_{17}(x) = x^4+x^3+x^2+x+1,$$

i.e.,

$$\alpha^{17} \in \mathbb{F}_{16} \subset \mathbb{F}_{256}.$$

MDS Codes and the MDS Conjecture

- MDS codes satisfy the Singleton bound with equality: $d = n - k + 1$
- GRS codes are MDS
  - there exist GRS codes with parameters $[n, k, n-k+1]$, $1 \le k \le n \le q+1$, for all finite fields $\mathbb{F}_q$
  - recall that lengths $q,\ q+1$ can be obtained using column locators 0 and ∞
  - when $q = 2^m$ and $k = 3$, the matrix

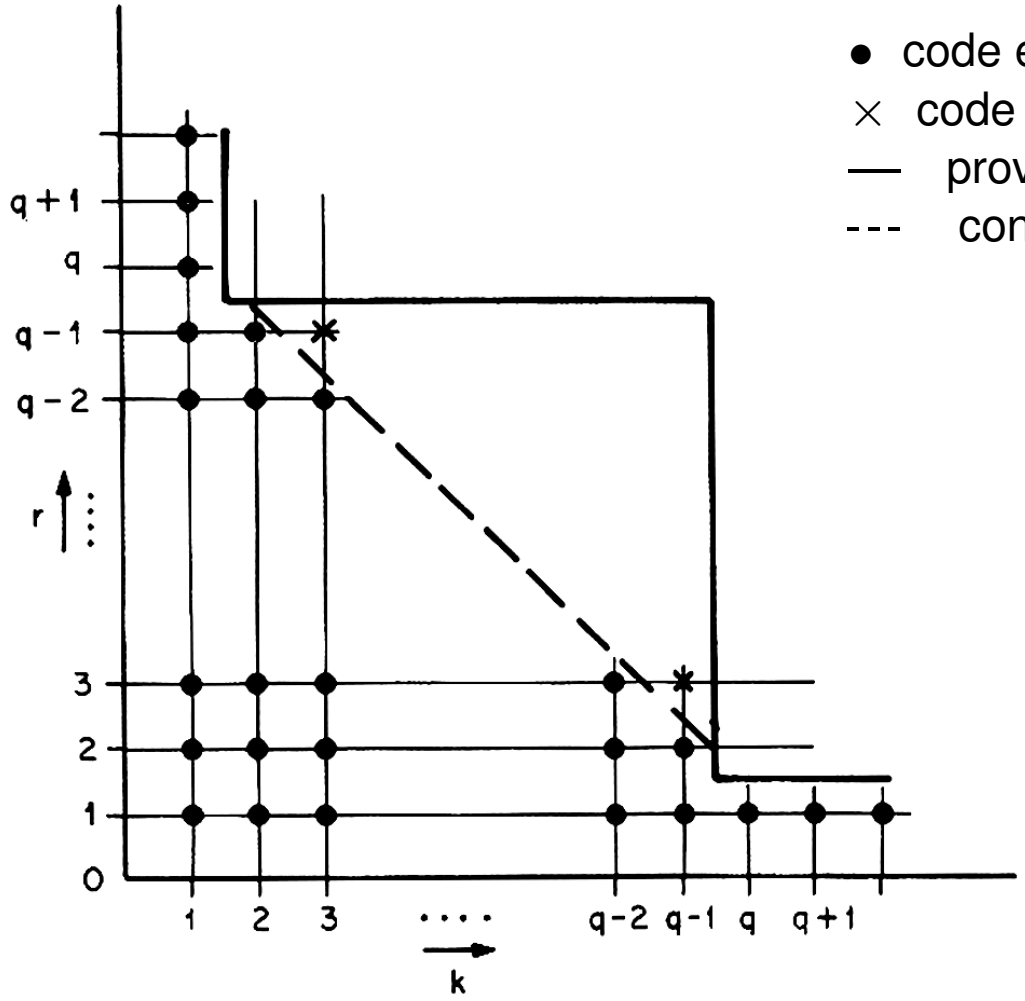
$$\begin{pmatrix} 1 & 1 & \dots & 1 & 1 & 0 & 0 \\ \alpha_1 & \alpha_2 & \dots & \alpha_{q-1} & 0 & 1 & 0 \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_{q-1}^2 & 0 & 0 & 1 \end{pmatrix}$$

    generates a $[q+2, 3, q]$ linear MDS *triply-extended GRS* code; its dual is a $[q+2, q-1, 4]$ MDS code
  - the $[n, n, 1]$ full space, $[n, 1, n]$ repetition code, and the $[n, n-1, 2]$ parity code are MDS, for any length n and field $\mathbb{F}_q$
  - *no other non-trivial MDS codes of length* $> q+1$ *are known*. Let

$$L_q(k) = \max\{\, n \ :\ \exists\, [n, k, n-k+1] \text{ MDS code over } \mathbb{F}_q \,\}.$$

The MDS Conjecture. $L_q(k) = q+1$, *except for the cases listed above.*

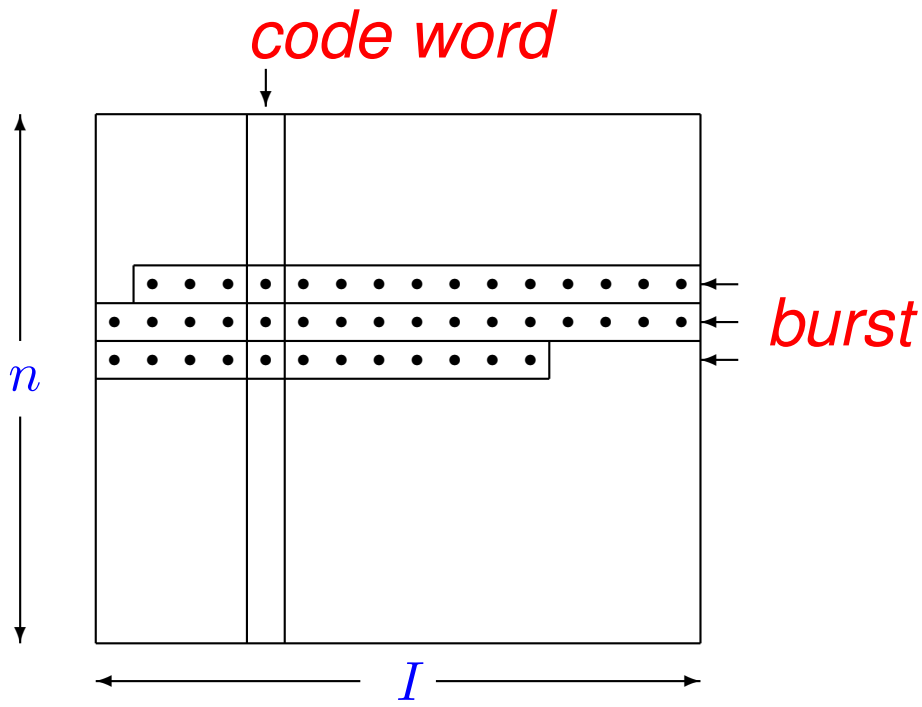
MDS Conjecture Plot



- code exists for all q
- $\times$ code exists iff $q=2^m$
- —— proven bound
- --- conjectured bound

one of the long-standing open problems in coding theory

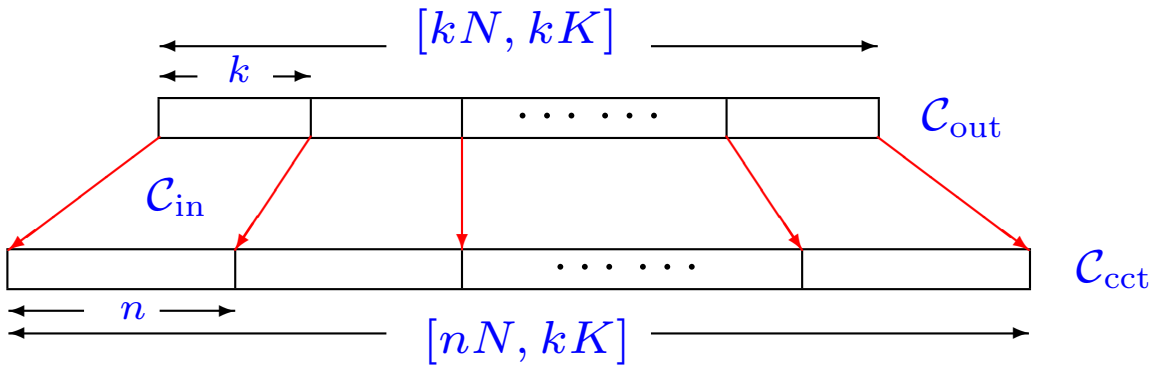
Interleaving and Burst Error Correction



- *Interleaving* spreads bursts of errors among codewords, so that each codeword is affected by a small number of errors.
- Cost: increased *latency*

Concatenated Codes

- Let $\mathcal{C}_{\text{in}}$ be an $[n, k, d]$ code over $\mathbb{F} = \mathbb{F}_q$ (the *inner code*), and let $\mathcal{C}_{\text{out}}$ be an $[N, K, D]$ code over $\Phi = \mathbb{F}_{q^k}$ (the *outer code*)
- Represent Φ as vectors in $\mathbb{F}^k$ using a fixed basis of Φ over $\mathbb{F}$
- A *concatenated code* $\mathcal{C}_{\text{cct}}$ is constructed by replacing each $\mathbb{F}^k$-symbol in $\mathcal{C}_{\text{out}}$ by its mapping to $\mathbb{F}^n$ according to $\mathcal{C}_{\text{in}}$.



- $\mathcal{C}_{\text{cct}}$ has parameters $[nN, kK, \geq dD]$ over F
- $\mathcal{C}_{\text{out}}$ is typically taken to be a GRS code.
  - By letting k grow, we can obtain arbitrarily long codes over $\mathbb{F}_q$, for fixed q. By careful choice of $\mathcal{C}_{\text{in}}$, very good codes can be constructed this way.