ERROR-CORRECTING CODE
\[ C \subseteq \Sigma^n \quad \Sigma \text{ alphabet} \]
\( (n, k, d)_q \text{ - code} \)
- block length \( n \)
- message length \( k(= \log_q |\Sigma|) \)
- minimal distance \( d = \Delta(C) \)
- alphabet size \( q = |\Sigma| \)

Linear code: \( \Sigma^n = \mathbb{F}_q^n \quad C \subseteq \mathbb{F}_q^n \quad \) linear subspace

**Generator matrix**
\[ G \in \mathbb{F}_q^{k \times n} \]
Rows \( G \) = basis of \( C \)
\[ C = \{ x G \mid x \in \mathbb{F}_q^k \} \]
(can think of \( x \) as the actual messages)

**Parity check matrix**
\[ H \in \mathbb{F}_q^{n \times (n-k)} \]
\[ C = \{ y \in \mathbb{F}_q^n \mid y H = 0 \} \]

Both are compact representations of code \( C \)
Computationally (essentially) equivalent
HAMMING CODE

∀ C ∈ {0, 1}^n

[2^l - 1, 2^l - l - 1, 3]_2 - code

Code with parity check matrix H
having rows

1, 2, ..., 2^l - 2, 2^l - 1

written in binary

Add \( t = O(\log k) \) extra bits to
message to get distance 3. \( \iff \) correct
a single error.

This is necessary

\[ \mathcal{B}_q(r, r) = \{ x \in \mathbb{F}_q^n | \Delta(x, y) \leq r \} \]

Hamming ball. Volume of this ball

\[ \text{Vol}_q(r, n) = \sum_{i=0}^{r} \binom{n}{i} (q - 1)^i \]

THEOREM 6 (HAMMING)

If an \( (n, k, d)_q \) code exists, then

\[ q^k \cdot \text{Vol}_q(\lfloor \frac{d-1}{2} \rfloor, n) \leq q^n \]

Codes that meet this bound are called perfect

Hamming codes are perfect.

Only two other codes perfect.
so Hamming bound rules out

\((n, n-d+1, d)\_2\) - code already

for \(d=3\) in many cases.

But why did we ask only for \(k=n-d+1\)?

Why not be even more ambitious (for
large alphabets, say)? Because this is
an upper bound regardless of alphabet

**THEOREM 8 (SINGLETON BOUND)**

If \(C\) is an \((n, k, d)\_q\) - code,
then \(d \leq n-k+1\).

**Proof**

Let \(\Sigma\) alphabet of \(C\). Consider

projection maps \(\Pi: \Sigma^n \rightarrow \Sigma^{k-1}\) that projects
every word \(\in \Sigma^n\) to its first \(k-1\)
coordinates.

Since \(|\text{Range}(\Pi)| = q^{k-1}\)

\(\quad |C| = q^k\)

There exist \(x, y \in C\) with \(x \neq y\), such that
\(\Pi(x) = \Pi(y)\), i.e., \(x\) and \(y\) agree in
first \(k-1\) coordinates.

Then \(\Delta(C) \leq \Delta(x,y) \leq n-(k-1)\) \(\Box\)

**DEF 9**

An \((n, k, d)\_q\) - code \(C\) is a MAXIMUM DISTANCE

SEPARABLE (MDS) CODE if \(C\) meets the

Singleton bound, i.e., if \(k = n-d+1\)

\((d = n-k+1)\)
MDS codes and perfect codes are incomparable. (i.e. \exists non-perfect MDS codes and \exists non-MDS perfect codes)

We will next study an important family of MDS codes

**REED-SOLOMON CODES (RS CODES)**
 Introduced in [Reed & Solomon 1959]
 Constructed using univariate polynomials.
 Relies heavily on following basic property

**FACT 10 (DEGREE MANTRA*)**
 Two distinct polynomials \( p_1, p_2 \in \mathbb{F}_q[x] \)
 of degree strictly less than \( k \) agree in strictly less than \( k \) points in \( \mathbb{F}_q \). That is, there are at most \( k-1 \) distinct points \( x \in \mathbb{F}_q \)
 such that \( p_1(x) = p_2(x) \).

**Proof** \( p_1 - p_2 \) is a non-zero polynomial of degree \( < k \) and hence has \( < k \) roots. \( \blacksquare \)

For ease of exposition, describe RS codes by encoding function

\[ \text{RS}_{q,n,k} \text{ code} \]

\[ q = \text{alphabet size} = |\mathbb{F}_q| \]
\[ n = q^k \text{ block length} \]
\[ k \leq n \text{ message length} \]

*) As referred to by Ryan O'Donnell
RS_{q,n,k} is constructed as follows:

1. Generate field $\mathbb{F}_q$ explicitly (so that we can work in it—we'll ignore this issue).

2. Pick $n$ distinct elements $x_1, \ldots, x_n \in \mathbb{F}_q$ (requires $n \leq q$).

3. Let the message $\mathbf{c} \in \mathbb{F}_q^k$ be $c_0, c_1, \ldots, c_{k-1} \in \mathbb{F}_q$ and identify it with polynomial $P(x) = \sum_{j=0}^{k-1} c_j x^j$.

4. Encode $\mathbf{c} = P(x)$ by evaluation at all $x_i$, i.e., the code-word is $E(P(x)) = (P(x_1), P(x_2), \ldots, P(x_n))$.

Clearly, $RS_{q,n,k}$ is a linear code where

$$x \cdot E(P(x)) = E(x \cdot P(x))$$

$$E(P(x)) + E(Q(x)) = E((P+Q)(x))$$

Also clear that block length $= n$, message length $= k$.

Need to argue distance

**Proposition 11** $RS_{q,n,k}$ is an $[n,k,n-k+1]_q$-code.

**Proof**

Note that this meets the Singleton bound, and so is best possible. Hence $d \leq n-k+1$, and we only need to prove $d \geq n-k+1$. 
Suppose we have two distinct codewords $E(P(x))$ and $E(Q(x))$ with $\deg(P(x))$, $\deg(Q(x)) < k$. By the Degree Mismatch, they agree on at most $k-1$ points of $\mathbb{F}_q$ implies $P(x)$ and $Q(x)$ disagree on at least $n-k+1$ of the points $x_1, \ldots, x_n$.

We have the following theorem:

**Theorem 12**

For every prime power $q$ and every pair $k, n \in \mathbb{N}$ such that $k \leq n \leq q$, there exists an $[n, k, n-k+1]_q$ code.

RS codes are possibly the most commonly used codes in practice. Used on CDs and DVDs, they can be decoded efficiently—will talk about this later.

How? This is a code over $\mathbb{F}_q$ for $q$ large. Embed $\log_2 q$ bits as element $x \in \mathbb{F}_q$. Yields $(n^\frac{\log_2 q}{\log_2 q}, k \log_2 q, n-k+1)_2$ code. Not too bad code, but smarter constructions could do better ($= \text{get larger distance}$).
But... Error correction will be much better if errors occur in bursts. Even if \( \log_2 q \) bits are corrupted, as long as they are adjacent, this is 1-2 positions.

So this code is good at correcting burst errors. And apparently, typical errors on storage devices tend to happen in bursts.

Bottleneck of RS codes: \( q > n \)

Can get around this by using multivariate polynomials.

**REED-MULLER CODES (RM CODES)**

Look at total degree of multivariate polynomials.

\[ x^2 y^3 z + x^n z \] has total degree 6

**RM\(_m\), \( q \) code**

Alphabet \( \mathbb{F}_q \) where \( q \) is a prime power.

Total degree of polynomials \( t \)

Multivariate polynomials over \( x_1, x_2, \ldots, x_n \)

Two cases depending on relation between \( t \) and \( q \)
Case 1 \( l < q \)

Message sequence of coefficients

\[
(M_{i_1 \ldots i_m}) \ i_1 + \ldots + i_m \leq l
\]

Represents polynomial

\[
M(x_1, \ldots, x_m) = \sum_{i_1 + \ldots + i_m \leq l} M_{i_1 \ldots i_m} x_1^{i_1} \ldots x_m^{i_m}
\]

Codeword

\[
(M(x)) \ x \in \mathbb{F}_q^m
\]

Block length \( n = q^m \)

Message length \( \# \) \( m \)-long sequences of non-negative integers that sum to at most \( l \)

\[
= \binom{m + l}{m}
\]

Example \( m = 3 \), \( l = 4 \)

Place \( m + l \) balls

0 0 0 0 0 0 0 0 0

Colour \( m \) balls

0 0 0 0 0 0 0 0 0

\( i_j \) = \# white balls between \((j-1)\)st and \(j\)th coloured ball

\[
i_1 = 2 \quad i_2 = 0 \quad i_3 = 1
\]

\[
\sum_j i_j = 3 \leq 4
\]
What about distance?

It is \((1 - \frac{d}{q})^n\)

Follows from generalization of Degree Manna

**Schwartz-Zippel (DeMillo-Lipton) Lemma 13**

A non-zero polynomial \(f \in \mathbb{F}_q[x_1, \ldots, x_n]\)
of total degree \(d\) is zero on at most
\(a \cdot \frac{d}{q}\) fraction of the points in \(\mathbb{F}_q^n\)

For different messages \(M \neq M'\)
\(M - M'\) is a non-zero polynomial of
total degree \(d \Rightarrow \text{zero on at most } m \cdot \frac{d}{q} \text{ points.}\)

Encodings of \(M\) and \(M'\) differ in
\((1 - \frac{d}{q})^n\) points.

**Proof of Schwartz-Zippel**

Induction over \(m\). Note that for
random \(a_1, \ldots, a_m \in \mathbb{F}_q\)
\(Pr[f(a_1, \ldots, a_m) = 0] \geq 1 - \frac{d}{q}\)

**Base case:** if \(f\) univariate, Degree Manna
says at most \(d\) roots

\(Pr[f(a) \neq 0] \geq \frac{q - d}{q} = 1 - \frac{d}{q}\)
Inductive step

Write \( f(x_1, \ldots, x_m) = \sum_{i=0}^{d} x_i^i f_i(x_2, \ldots, x_m) \)

\( f \) is non-zero, so some \( f_i \neq 0 \)

Fix largest \( i^* \) so to \( f_i \neq 0 \)

By Hoeffding

\[
\Pr \left[ \text{fix}(a_2, \ldots, a_m) \neq 0 \right] \geq 1 - \frac{d - i^*}{q}
\]

If \( \text{fix}(a_2, \ldots, a_m) \neq 0 \), then

\( f(x_1, a_2, \ldots, a_m) \) is a non-zero \( i^* \)-variate poly

of degree \( i^* \), and so is \( 0 \) for at most \( i^* \) values of \( x_1 \). Hence

\[
\Pr \left[ f(a_1, \ldots, a_m) \neq 0 \right] \geq \Pr \left[ f(a_1, \ldots, a_m) \neq 0 \mid \text{fix}(a_2, \ldots, a_m) \neq 0 \right] \cdot \Pr \left[ \text{fix}(a_2, \ldots, a_m) \neq 0 \right]
\]

\[
\geq \left( 1 - \frac{i^*}{q} \right) \left( 1 - \frac{d - i^*}{q} \right)
\]

\[
\geq 1 - \frac{d}{q}.
\]
Case 2: total degree \( l > q \)

Messages: Polynomials of total degree \( \leq l \)
Individual degree \( \leq l - 1 \) in each variable

\[
S(m, l, q) := \{ i_1 \ldots i_m \mid \sum_{j} i_j \leq l \ 0 \leq i_j < q \}
\]

\[
K(m, l, q) := \left| S(m, l, q) \right|
\]

Messages

\[
M(x_1, \ldots, x_m) = \sum_{i \in S(m, l, q)} m_{i} x_1^{i_1} x_2^{i_2} \cdots x_m^{i_m}
\]

Codeword

\[
( M(x) )_{x \in \mathbb{F}_q^m}
\]

Block length \( q^m \)

Message length \( K(m, l, q) \)

Distance

Write \( l = a(q - 1) + b \) \( a \leq m \) \( b < q - 1 \)

Then distance is

\[
\geq q^{m-a} \left( 1 - \frac{b}{q} \right)
\]
Lemma 14

If \( f \in \mathbb{F}_q[x_1, \ldots, x_m] \) has individual degree at most \( s \) in each variable and has total degree \( d = sk+r \), then \( f \) is non-zero on at least \( \left(1 - \frac{s}{q}\right)^k \left(1 - \frac{r}{q}\right) \) fraction of points in \( \mathbb{F}_q^m \).

Proof (Exercise)

Example A multilinear polynomial (individual degree \( s \)) of total degree \( k \) is non-zero on at least \( 1 - k \) fraction of \( \mathbb{F}_2^m \).

Sample parameters for case 2 (as in Reed-Muller paper)

\( q = 2, \ l < m \)

Then \( K(m, l, 2) = \text{Vol}_2(l, m) \geq \binom{m}{l} \)

Distance \( \geq 2^{m-l} \)

\( RM_{m, l, 2} \) is a \( [2^m, \binom{m}{l}, 2^{m-l}]_2 \) code
Let's talk about randomness. Already saw several times that randomness can be very helpful.

Examples:

1. Good Ramsey graphs hard to construct. Random graphs (with right parameters) excellent properties.

2. Expander graphs complicated to construct. Random graphs are super-good expanders.

3. Will talk later about what it means that an error correcting code is (asymptotically) good. Tons of research has gone into code construction. Cannot beat properties of random code over $\mathbb{F}_2 = \{0, 1\}$.

$\text{BPP} = "P + possibility to flip random coins"

Is $\text{BPP}$ strictly larger than $P$?

Actually, conventional wisdom is that probably $P = \text{BPP}$.

Line of research "Hardness vs randomness"
Basic idea:

If \( f : \{0,1\}^n \to \{0,1\} \)

hard to compute in \( P \), then for a random \( x \in \{0,1\}^n \)

\((x, f(x))\) will look like random \((n+1)\)-bit string (to poly-time algorithm)

Can stretch this to longer and longer pseudo-random strings

[Impagliazzo-Wigderson '97]

If there are explicit \( n \)-bit functions computable in deterministic time \( 2^{o(n)} \)

that require \( \text{Boolean circuits of size } 2^{-2^{n}} \), then \( P = BPP \)

Such functions are widely believed to exist

This discussion generalizes to Turing machines

Many problems are algebraic (e.g., problems involving polynomials)

Natural to study such problems in algebraic computational model
Canonical example Determinant

X = n\times n matrix with (i,j)th entry x_{ij}.

Determinant is polynomial

\[ \det(X) = \sum_{\pi \in S_n} \text{sgn}(\pi) \prod_{i \leq [n]} x_{i, \pi(i)} \]

S_n group of permutations \( \pi : [n] \to [n] \)

\( \text{sgn}(\pi) = \text{sign of } \pi = (-1)^{\# \text{inversions of } \pi} \)

Can be computed using multiplication and addition (algebraic operations) using Gaussian elimination in time poly(n).

Can also be computed efficiently in parallel on multiple computers (is in NC^2, although we don't need to know what that is).

General fact Any polynomial that can be computed efficiently using addition and multiplication can also be computed (even more) efficiently in parallel.

Time for algebraic computation believed to be false for general Boolean computation.

So extra structural restrictions can help to get stronger bounds.
So algebraic computation interests me.

In particular, we want to talk about algebraic circuits.

\[ (x+y) 5y + 5y(x+z) \]

An algebraic circuit is a DAG:
- leaves/sources labelled by
  - variables \( x, y \)
  - field elements \( x \in \mathbb{F} \)
- internal nodes/gates labelled \( +, * \)

Size of circuit = # edges in DAG
Computes polynomials in the natural way (see example above)

Many Boolean computational tasks can also be "algebraized"
Take problem over \( \mathbb{R}, \mathbb{Z} \) and imbed in \( \mathbb{F}_q \)

Examples
- \( \text{IP} = \text{PSPACE} \)
- Hardness of approximation of NP-complete problems (PCP theorem)
Fundamental problem in algebraic complexity theory

**POLYNOMIAL IDENTITY TESTING (PIT)**

Given an algebraic circuit $C$ computing a polynomial $f(x_1, \ldots, x_n)$, is $f(x^2)$ the identically zero polynomial?

Randomized algorithm:
- Go to large enough field $F$
- Pick random point $x \in F^m$
- If $f(x) = 0$ answer yes, otherwise no.

Schwartz-Zippel lemma says this will work. But can we get a deterministic algorithm?

Two flavours of algorithms

---

**Black-box**

Just try to find out by evaluating circuit on points in $F^m$ as a black box

---

**White-box**

Also analyze structure of circuits (but might not be easy to use this info)
Black box algorithms equivalent to finding HITTING SETS

If \( \mathcal{H} \) is hitting set for class of circuits \( \mathcal{C} \) of certain structure
if \( \forall \) non-zero \( C \in \mathcal{C} \exists \mathcal{H}

s.t. \( C(\mathcal{H}) \neq 0 \)

It can be shown that small hitting sets exist. The problem is to find small explicit hitting sets
for interesting classes of circuits \( \mathcal{C} \)
(general case of unrestricted circuits beyond reach).

PIT is a fundamental problem that turns up in different contexts

Example: Bipartite matching can be reduced to PIT

Take bipartite graph \( G = (U \cup V, E) \) \( |U| + |V| = n \)
Define matrix \( X \) such that

\[
(X)_{ij} = \begin{cases} 
1 & \text{if edge from } i \in U \text{ to } j \in V \\
0 & \text{otherwise}
\end{cases}
\]

\( \det(X) \neq 0 \Rightarrow \exists \text{ perfect matching} \)
This shows that determining whether a graph has a perfect matching can be determined fast in parallel with randomness (is in randomized NC).

This is the only way known to prove this.

Open problem: Construct any non-trivial family set $H$ for this class of determinants.