A good (n, M, d)-code should allow:

1. fast transmission.
2. transmission of wide variety of messages.
3. to correct many errors.

These will be possible if we have small n, large M and d. But these are, quite intutively also, conflicting aims. We generally aim for maximizing M, given n and d.
We denote by $A_q(n, d)$ the largest value of M such that q-ary (n, M, d)-code exists.

Theorem 0.1. (i) $A_q(n, 1) = q^n$. (ii) $A_q(n, n) = q$.
Proof. (i) $d = 1$ reqiures the words to be *distinct* only. Therefore, $(n, M, 1)$-code $= (F_q)^n \implies M = A_q(n, 1) = q^n$
(ii) $d = n$ implies symbols appearing at any fixed position in the code must be all different $\implies A_q(n, d) \leq q$. But there exists q-ary repetition code of length n, $\begin{bmatrix} 00\ldots0 \\ 11\ldots1 \\ \vdots \\ qq\ldots q \end{bmatrix}$, hence $A_q(n, n) = q$. □

Definition 0.2. Two q-ary codes are said to be ***equivalent*** if one can be obtained from another by combinations of operations of the types, i.e.(i) permuations of the positions of the code. (ii) permutations of the symbols appearing at a fixed position.

The second point means assigning a permutation function f on the *alphabet*, and applying f in a particular column of the code. Distances between operators remain same in these operations, so *equivalent* codes have the same parameteres (n, M, d) and therefore, will correct the same number of errors.

Lemma 0.3. Any q-ary (n, m, d)-code is equivalent to a (n, M, d)-code containing the vector $\mathbf{0} = 00\ldots0$.

Proof. For any codeword $\mathbf{x} = x_1, x_2 \cdots x_n$ in the code, for each $x_i \neq 0$ applying the permutation

$$\begin{pmatrix} 0 & x_i & j & \\ \downarrow & \downarrow & \downarrow & \forall\ j \neq 0, x_i \\ x_i & 0 & j & \end{pmatrix}$$

to the symbols of position i, will give the desired (n, M, d)-code. □

Now taking F_2 to be the set $\{0, 1\}$. Let $\mathbf{x} = x_1x_2\cdots x_n,\ \mathbf{y} = y_1y_2\cdots y_n \in (F_2)^n$, we define the following operations, namely, the ***sum*** $\mathbf{x} + \mathbf{y}$ is the vector in $(F_2)^n$ defined as

$$\mathbf{x} + \mathbf{y} = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)$$

while the ***intersection*** $\mathbf{x} \cap \mathbf{y}$ is a vector in $(F_2)^n$ defined as

$$\mathbf{x} \cap \mathbf{y} = (x_1y_1, x_2y_2, \ldots, x_ny_n)$$

The terms $x_i + y_i$ and x_iy_i are calculated modulo 2.
The ***weight*** of a vector $\mathbf{x}$ in $(F_2)^n$, denoted $w(\mathbf{x})$, is the number of 1s in $\mathbf{x}$.

The proofs of the following two lemmas are ommited as they are quite easy to determine once one gets the lemma.

Lemma 0.4. If $\mathbf{x},\ \mathbf{y} \in (F_2)^n$, then $d(\mathbf{x}, \mathbf{y}) = w(\mathbf{x} + \mathbf{y})$.

Lemma 0.5. If $\mathbf{x},\ \mathbf{y} \in (F_2)^n$, then

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{x}) + w(\mathbf{y}) - 2w(\mathbf{x} \cap \mathbf{y}).$$

Theorem 0.6. Suppose d is odd. Then a binary (n, M, d)-code exists if and only if $(n+1, M, d+1)$-code exists.

Proof. $\implies$: Suppose C is a binary (n, M, d)-code and $\mathbf{x} = x_1x_2\cdots x_n$ be one of it's codewords. Define $x_{n+1} = \sum_{i=1}^{n} x_i$. Let $\hat{C}$ be a $n+1$ length code, given by

$$\hat{C} = \{\hat{\mathbf{x}} \mid \hat{\mathbf{x}} = x_1x_2\cdots x_nx_{n+1} \forall\, \mathbf{x} \in C\}$$

Now $w(\hat{\mathbf{x}})$ is every codeword $\hat{\mathbf{x}}$ in $\hat{C}$, it follows from Lemma 0.5 that $d(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ will be even for all $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ in $\hat{C}$. Hence, $d(\hat{C})$ is even. But $d \leq d(\hat{C}) \leq d+1$, so $d(\hat{C} = d+1$ as d is odd. Therefore, there exists a binary $(n+1, M, d+1)$-code. (This type of operation is called adding overall *parity-check* on a code C)
$\impliedby$: Suppose C is a binary $(n+1, M, d+1)$-code. Let $\mathbf{x}, \mathbf{y} \in C$ such that $d(\mathbf{x}, \mathbf{y}) = d+1$. Now we remove one column where they differ from the whole code C. We are left now left with (n, M, d)-code. Therefore, binary (n, M, d)-code exists. □

Corollary 0.6.1. If d is odd, then $A_2(n+1, d+1) = A_2(n, d)$. Equivalently, if d is even, then $A_2(n, d) = A_2(n-1, d-1)$.

Proof. Follows directly from Theorem 0.6. □

Notion of a ***sphere*** in $(F_q)^n$, natural definition that follows from Hamming distance.

Definition 0.7. For any vector $\mathbf{u}$ in $(F_q)^n$, a ***sphere*** of radius r and centre $\mathbf{u}$, is given by

$$S(\mathbf{u}, r) = \{\mathbf{v} \in (F_q)^n \mid d(\mathbf{u}, \mathbf{v}) \leq r\}$$

Lemma 0.8. A sphere of radius r in $(F_Q)^n$ contains exactly

$$\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{r}(q-1)^r$$

vectors.

Proof. For finding vectors with a distance $d \in \{1, 2, \ldots, r\}$, we can choose d positions from the centre vector in $\binom{n}{d}$ ways, and we have $(q-1)$ choices for each of the positions. So, total vectors at a distance d from the the centre vector are $\binom{n}{d}(q-1)^d$. Then summing over all possible distances gives the result. □

Theorem 0.9 (**Hamming bound** or the **sphere-packing bound**)**.** A q-ary (n, M, d)-code saisfies

$$M\left\{\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{t}(q-1)^t\right\} \leq q^n$$

where $t = \left\lfloor \dfrac{d-1}{2} \right\rfloor$.

Proof. By Corollary **????**, spheres of radius $\leq t$ and centered at the codewords won't overlap, therefore, the sum of elements of all these sphere will not exceed the *cardinality* of $(F_q)^n$. □

Definition 0.10. A code that achieves the *sphere-packing bound*, i.e. equality occurs in Theorem 0.9, is called a ***perfect code***.

Some *trivial* examples of *perfect codes* are binary-repition code, single word codes or the whole set $(F_q)^n$. But the *non-trivial* ones are the ones we are looking to find, as these codes will allow us to transmit/receive messages with very less probable grey region.

Remark. For binary codes, the *Hamming bound* turns out be close to the actual values of $A_2(n, d)$ when $n \geq 2d+1$. It is a weak bound for the case when $n \leq 2d$. For such cases, *Plotkin bound* is better and the following lemmas will lead to the bound.

Lemma 0.11. If there exists a binary (n, M, d)-code, then there exists a binary $(n-1, M', d)$-code such that $M' \geq \frac{M}{2}$. Further, $A_2(n-1, d) \geq \frac{A_2(n,d)}{2}$.

Proof. Let C be the existing (n, M, d)-code then if we partition the set C in two disjoint sets, those ending with 0 and 1, then atleast one of them will have *order* $\geq \frac{M}{2}$ (as the sum of orders is M). Then if we remove the last bit from this larger set then we have our binary $(n-1, M', d)$-code with $M' \geq \frac{M}{2}$. Also, therefore $A_2(n-1, d) \geq \frac{A_2(n,d)}{2}$. □

Lemma 0.12. If C is a binary (n, M, d)-code with $n < 2d$, then

$$M \leq \begin{cases} \dfrac{2d}{2d-n} & \text{if M is even} \\ \dfrac{2d}{2d-n} - 1 & \text{if M is odd} \end{cases}.$$

In paricular,

$$A_2(n, d) \leq 2 \left\lfloor \frac{d}{2d-n} \right\rfloor$$

Proof. Let $C = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and T be the $\binom{M}{2} \times n$ matrix whose are vectors $\mathbf{x}_i + \mathbf{x}_j$ for $1 \leq i < j \leq M$. Now in T, number of non-zero entries per row is $d(\mathbf{x}_i, \mathbf{y}_j)$, i.e. greater equal to d, let the total non-zero entries of the matrix be $w(T)$. Thus,

$$\binom{M}{2} d \leq L \implies \frac{M(M-1)}{2} \cdot d \leq w(T)$$

If t_j codewords have 1 in j^{th} position then total number of 1s in j^{th} column of T are

$$t_j(M - t_j) \leq \begin{cases} \frac{M^2}{4} & \text{if M is even} \\ \frac{M^2-1}{4} & \text{if M is odd} \end{cases}$$

Thus,

$$w(T) \leq \begin{cases} \frac{M^2}{4} \cdot n & \text{if M is even} \\ \frac{M^2-1}{4} \cdot n & \text{if M is odd} \end{cases}$$

From above equations:

If M is even:

$$\begin{aligned} \frac{M(M-1)}{2} \cdot d &\leq \frac{M^2}{4} \cdot n \\ 2(M-1)d &\leq Mn \\ \frac{M}{2} &\leq \frac{d}{2d-n} \\ \frac{M}{2} &\leq \left\lfloor \frac{d}{2d-n} \right\rfloor \qquad \left(\text{as } \frac{M}{2} \in \mathbb{Z}\right) \\ M &\leq 2 \left\lfloor \frac{d}{2d-n} \right\rfloor \end{aligned}$$

If M is odd:

$$\begin{aligned} \frac{M(M-1)}{2} \cdot d &\leq \frac{M^2-1}{4} \cdot n \\ 2Md &\leq (M+1)n \\ M &\leq \frac{n}{2d-n} \\ \frac{M+1}{2} &\leq \frac{d}{2d-n} \\ \frac{M+1}{2} &\leq \left\lfloor \frac{d}{2d-n} \right\rfloor \qquad \left(\text{as } \frac{M+1}{2} \in \mathbb{Z}\right) \\ M &\leq 2 \left\lfloor \frac{d}{2d-n} \right\rfloor \end{aligned}$$

□

Theorem 0.13 (**Plotkin Bound**)**.** For a binary (n, M, d)-code:
(i) if d is even and $n < 2d$, then $A_2(n, d) \leq 2 \left\lfloor \frac{d}{2d-n} \right\rfloor$.
(ii) if d is odd and $n < 2d+1$, then $A_2(n, d) \leq 2 \left\lfloor \frac{d+1}{2d+1-n} \right\rfloor$.
(iii) if d is even, then $A_2(2d, d) \leq 4d$.
(iv) if d is odd, then $A_2(2d+1, d) \leq 4d+4$.

Proof. (i) Directly follows from Lemma 0.12.

(ii) From Corollory 0.6.1, we have $A_2(n, d) = A_2(n+1, d+1)$, therefore, $A_2(n, d) = A_2(n+1, d+1) = 2 \left\lfloor \frac{d+1}{2d+1-n} \right\rfloor$(follows from part(i) as $d+1$ is even)

(iii) From Lemma 0.11, $A_2(2d, d) \leq 2A_2(2d-1, d)$. Now, from part(i), we have $A_2(2d-1, d) \leq 2 \left\lfloor \frac{d}{1} \right\rfloor \implies A_2(2d, d) \leq 4d$

(iv) Follows from Corollory 0.6.1 and part(iii).

□

Definition 0.14. A ***balanced-block design*** consists of a set S of v elements, called *points*, and a collection of b subsets of S, called *blocks*, such that for some fixed k, r and λ

1. each blocks contain exactly k points.
2. each point occurs in exactly r blocks.
3. each pair of points occurs together in exactly λ blocks.

We denote this design as a (b, v, k, r, λ)-*design.*

A *balanced block design* satisfy these two basic conditions, namely, (i) $bk = vr$. (ii) $r(k-1) = \lambda(v-1)$. First one results from counting all the points in two ways, points in each block and number of blocks each point is in. Second results from equating the pairs of a particular point in two ways.

Definition 0.15. The ***incidence-matrix*** $A = [a_{ij}]$ of a block design is a $v \times b$ matrix, where rows corresponds to points $x_1, x_2, \ldots, x_v$ of the design while the columns represent the blocks $B_1, B_2, \ldots, B_b$. The matrix is given by

$$a_{ij} = \begin{cases} 1 & \text{if } x_i \in B_j \\ 0 & \text{if } x_i \notin B_j \end{cases}$$

Incidence matrices of block codes are used to generate *perfect codes.*