CS250/EE387 - LECTURE 15 - FOLDED RS CODES

AGENDA

⓪ The story so far

① FRS codes

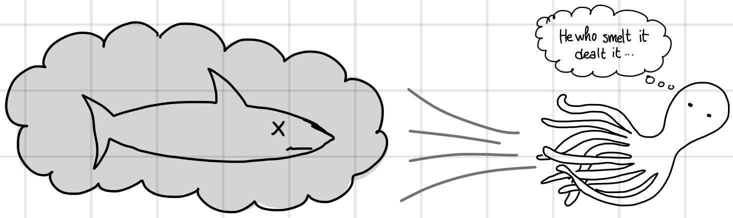
  Ⓐ DEFINITION

  Ⓑ FIRST PASS

  Ⓒ NEXT PASS

TODAY'S OCTOPUS FACT

Octopuses can shoot (sometimes toxic) ink to defend against predators.



⓪ THE STORY SO FAR:

- REED-SOLOMON CODES! AWESOME!
- LIST-DECODING CAPACITY: $\rho = 1 - R$
- GURUSWAMI-SUDAN: Can efficiently list-decode RS codes up to $\rho = 1 - \sqrt{R}$
- QUESTION: Can we efficiently list-decode RS codes up to $\rho = 1 - R$?

  ANSWER: **NO** or **PROBABLY NOT**, depending on the RS code.

  If eval pts $= \mathbb{F}_q$ [Ben-Sasson, Kopparty, Radhakrishnan]

  It's as hard as discrete log. [Cheng, Wan]

- :(

Today we will see the happy ending to the story, via...

FOLDED REED-SOLOMON CODES.

These were the first codes known to achieve capacity with explicit algs [Guruswami, Rudra 2008ish] and since then there have been several other constructions.

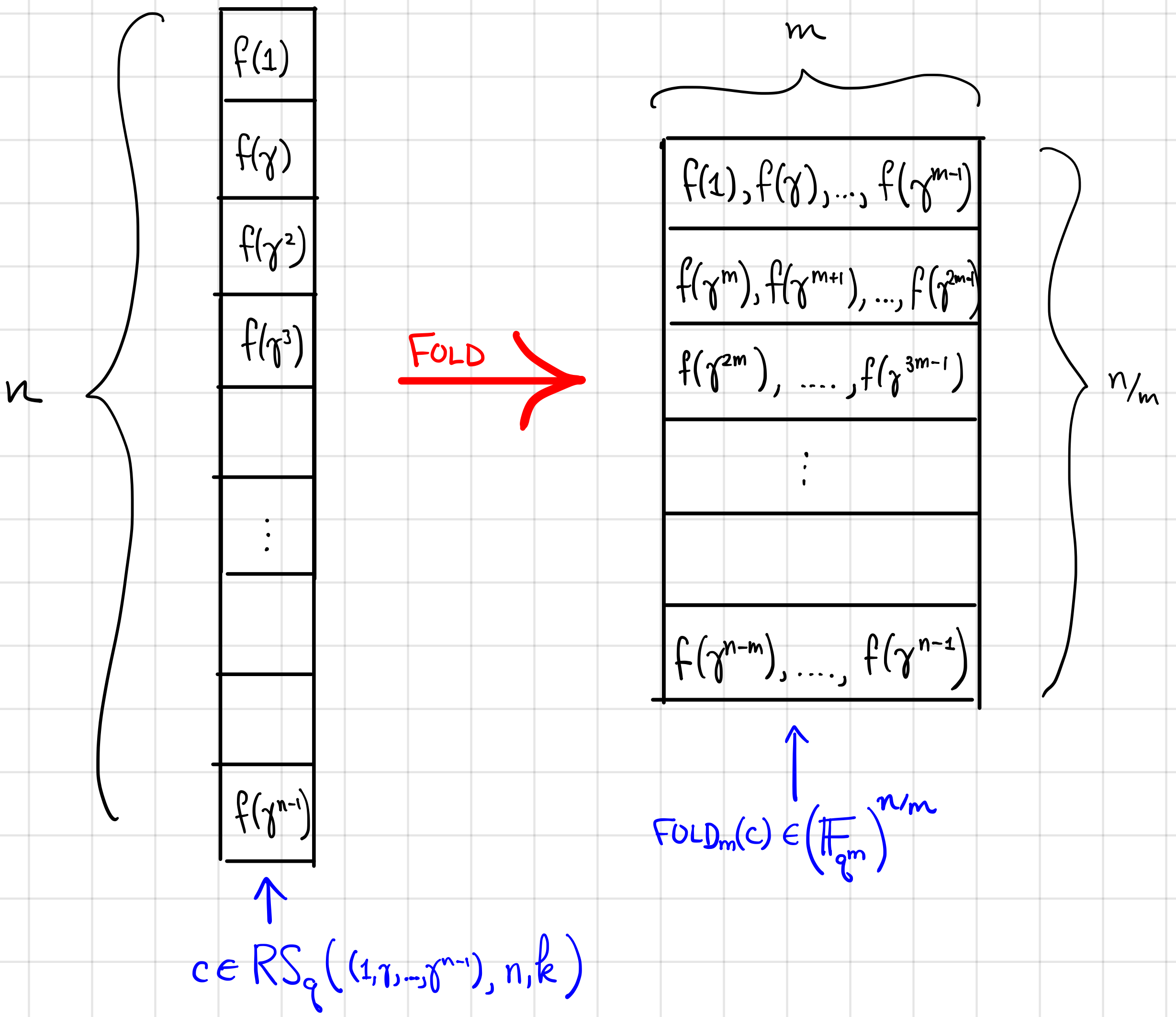
1 Folded Reed Solomon Codes

A Definition.

A **Folded RS code** is obtained by "folding" RS codes:

Start with an RS code with eval pts $1, \gamma, \gamma^2, \gamma^3, \dots, \gamma^{n-1}$, where $n=q-1$, γ is a primitive element of $\mathbb{F}_q$

Suppose that $m \mid n$. Then consider the following operation on a codeword from the RS code:



Define the m-folding $\mathrm{FOLD}_m(\mathcal{C}')$ of a code $\mathcal{C}'$ as $\{\mathrm{FOLD}_m(c) : c \in \mathcal{C}'\}$.

DEF. Let $C' = RS_q((1, \gamma, \dots, \gamma^{n-1}), n, k)$, where $n = q-1$ and $\gamma \in \mathbb{F}_q$ is a primitive element. Let $m | n$.

The **FOLDED RS CODE** corresponding to C' is $\mathrm{FOLD}_m(C')$.

The length of $\mathrm{FOLD}_m(C')$ is $N = n/m$

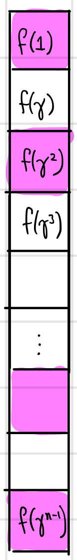
The alphabet is $\Sigma = \mathbb{F}_q^m$.

NOTE: Since folding just shuffles around the symbols, the rate does not change. Formally,

$$\mathrm{Rate}(C) = \frac{\log_{q^m}|C|}{N} = \frac{\log|C|}{\frac{n}{m}\cdot\log(q^m)} = \frac{\log|C|}{n\log(q)} = \frac{\log|C'|}{n\log(q)} = \frac{\log_q|C'|}{n} = \mathrm{Rate}(C')$$

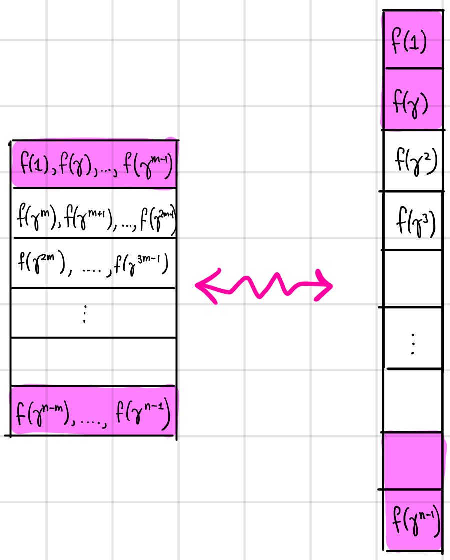
Why should this be a good idea? It "restricts" the power of the adversary:

WORLD 1: Bad guy gets to corrupt a p-fraction of an RS codeword.



Bad guy has more freedom – they can corrupt any set of symbols he wants from the original codeword.

WORLD 2: Bad guy gets to corrupt a p-fraction of a folded RS codeword.



Here, much less freedom. Only certain error patterns in the original codeword are allowed.

We will prove (or at least sketch):

THM. Let $1 \le s \le m$. Let $\mathcal{C}$ be an m-folded RS code of rate R. Then $\mathcal{C}$ is (p, L)-list-decodable for

$$p = \frac{s}{s+1}\left(1 - \left(\frac{m}{m-s+1}\right)\cdot R\right)$$

with list size $L = q^s$. Moreover, the list is contained in an affine subspace of dimension s.

Choose $m = 1/\varepsilon^2$, $s = 1/\varepsilon$. Then $L = q^{1/\varepsilon}$, and

$$\frac{s}{s+1}\left(1 - \left(\frac{m}{m-s+1}\right)R\right) = \frac{1/\varepsilon}{1/\varepsilon + 1}\cdot\left(1 - \left(\frac{1/\varepsilon^2}{1/\varepsilon^2 - 1/\varepsilon + 1}\right)\cdot R\right)$$

$$= \frac{1}{1+\varepsilon}\cdot\left(1 - \left(\frac{1}{1-\varepsilon+\varepsilon^2}\right)R\right)$$

$$= 1 - R - O(\varepsilon)$$

So folded RS codes will give us capacity-achieving list-decodable codes:

- Rate R
- $p = 1 - R - O(\varepsilon)$

This trade-off is optimal.

- $L = q^{1/\varepsilon} = (N/\varepsilon^2)^{1/\varepsilon}$ ← This is NOT optimal... should probably be $1/\varepsilon$.
- $|\Sigma| = q^{1/\varepsilon^2} = (N/\varepsilon^2)^{1/\varepsilon^2}$ ← This is also NOT optimal... we'd like it to be constant.

Lower bound on the list size is wide open

Note: In recent work it was shown that in fact the list size IS constant! [Kopparty-RonZewi-Saraf-W. 18]
But we won't talk about this...

Subsequent work has given codes which get basically all of the desiderata, although there are still many open questions.

(B) FIRST PASS: Let's ignore that parameter "s."

Here is the decoding algorithm. It has a familiar outline.

FOLDED RS DECODER TAKE I.

Input:
$$y = \left[\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{m-1} \end{pmatrix}, \begin{pmatrix} y_m \\ \vdots \\ y_{2m-1} \end{pmatrix}, \ldots, \begin{pmatrix} y_{n-m} \\ \vdots \\ y_{n-1} \end{pmatrix} \right] \in \left(\mathbb{F}_q^m \right)^N$$

as well as an agreement parameter t, and a parameter D, and $k \le n$.

Output: A list $\mathcal{L}$ containing all polynomials $f(X) \in \mathbb{F}_q[X]$ of degree $< k$, so that for at least t values of $i \in [0, N-1]$,

$$\begin{pmatrix} f(\gamma^{mi}) \\ \vdots \\ f(\gamma^{mi+m-1}) \end{pmatrix} = \begin{pmatrix} y_{mi} \\ \vdots \\ y_{mi+m-1} \end{pmatrix}$$

STEP 1. (Interpolation)

Find a nonzero polynomial

$$Q(X, Y_1, Y_2, \ldots, Y_m) = A_0(X) + A_1(X) \cdot Y_1 + \cdots + A_m(X) \cdot Y_m$$

with $\deg(A_0) \le D + k - 1$, and $\deg(A_i) \le D$ for $i = 1, \ldots, m$.
so that

$$Q\Big(\underbrace{\gamma^{mi}}_{\substack{X \text{ takes on} \\ \text{the "first" eval} \\ \text{pt for that symbol}}}, \underbrace{y_{mi}, y_{mi+1}, \ldots, y_{mi+m-1}}_{Y_1, \ldots, Y_m \text{ take on the symbol } \begin{pmatrix} y_{mi} \\ \vdots \\ y_{mi+m-1} \end{pmatrix}} \Big) = 0 \quad \forall\, i \in \{0, \ldots, N-1\}.$$

ctd...

Folded RS decoder Take 1 continued...

STEP 2. (Root-Finding)

Find all the polys f so that

$$Q(X, f(X), f(\gamma X), \ldots, f(\gamma^{m-1} X)) \equiv 0,$$

and return them.

This alg. is not our final algorithm, and it only gets $p = \left(\frac{m}{m+1}\right)(1 - m \cdot R)$.
However, the main ideas will come through by analyzing this (easier) version, so let's start there.

As usual, we need to do three things.

i. Set parameters
ii. Show that we can find such a Q ← As usual, via linear algebra
iii. Show that STEP 2 is a good idea. ← As usual, low-deg. polys don't have many roots.
iv. Show that we can do STEP 2 efficiently. ← Usually this one follows since we can factor polys, but now we'll switch it up a bit.

i. Suppose that $t \geq N\left(\frac{1}{m+1} + mR\left(\frac{m}{m+1}\right)\right)$ ← This will give the value of $p = 1 - t/N$ that we claimed.

Let $D = \left\lfloor \frac{N-k+1}{m+1} \right\rfloor$

ii. There are $\underbrace{D+k}_{\text{Coeffs of } X^j \text{ that show up in } A_0} + \underbrace{m(D+1)}_{\text{coeffs of } Y_i \cdot X^j \text{ that show up in each } A_i} = (m+1)(D+1) + k - 1$ coefficients.

Our choice of N ensures that this is $\#\text{coeffs} > N = \#\text{constraints}$, so we can find the desired Q.

iii Let $R(X) = Q(X, f(X), f(\gamma X), \ldots, f(\gamma^{m-1}X))$

$$= \underbrace{A_0(X)}_{\deg\ D+k-1} + \underbrace{f(X)\cdot A_1(X)}_{\substack{\deg\ D+k-1 \\ [\text{Since } \deg(f)<k \\ \deg(A_1)\le D]}} + \cdots + \underbrace{f(\gamma^{m-1}X)\cdot A_m(X)}_{\deg\ D+k-1}$$

So $\deg(R) \le D+k-1$.

Suppose that $\mathrm{Fold}_m(f)$ agrees with y in at least t places. Then R has at least t roots, so $R\equiv 0$ as long as

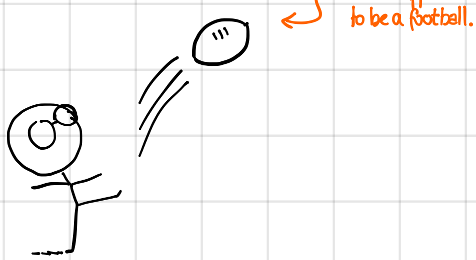
$$(\#\text{roots}) > \deg(R)$$
$$t > D+k-1 = \left\lfloor \frac{N-k+1}{m+1} \right\rfloor + k-1.$$

So it would be enough if
$$t \ge \frac{N-k}{m+1} + k$$
$$= \frac{N}{m+1} + \frac{mk}{m+1}$$
$$= N\left(\frac{1}{m+1} + \left(\frac{m}{m+1}\right)\cdot R\right)$$ which is what we chose.

Therefore, if f agrees w/ y too much, we <u>will</u> return it. ✓

iv. How do we actually find this list? Is it small?

PUNT. (we'll come back to this).



(C) FIXING UP the FIRST PASS.

We will tweak things to get MORE ROOTS in our interpolating poly.

Suppose that
$$\begin{pmatrix} f(\gamma^{mi}) \\ \vdots \\ f(\gamma^{mi+m-1}) \end{pmatrix} = \begin{pmatrix} y_{mi} \\ \vdots \\ y_{mi+m-1} \end{pmatrix}$$

We were using the root of Q at

$$Q(\gamma^{mi}, f(\gamma^{mi}), \ldots, f(\gamma^{mi+s-1}), f(\gamma^{mi+s}), \ldots, f(\gamma^{mi+m-1})) \equiv 0.$$

But we could get MORE roots if we did something like this:

Make Q have fewer vars so this makes sense.

$$Q(\gamma^{mi}, f(\gamma^{mi}), \ldots, f(\gamma^{mi+s-1})) \equiv 0$$

$$Q(\gamma^{mi+1}, f(\gamma^{mi+1}), \ldots, f(\gamma^{mi+s})) \equiv 0$$

$$\ddots$$

$$Q(\gamma^{mi+m-s}, f(\gamma^{mi+m-s}), \ldots, f(\gamma^{mi+m-1})) \equiv 0$$

Basically, we take a sliding window across each symbol and turn that into a constraint.

There's a trade-off here:
- more agreement! ← So we get better "t" for a given "D."
- more constraints... ← So D has to be smaller in order to be able to find Q.

But this trade-off turns out to be beneficial!

Folded RS Decoder Take II

The changes from TAKE I are highlighted in RED.

Input: $y = \left[\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{m-1} \end{pmatrix}, \begin{pmatrix} y_m \\ \vdots \\ y_{2m-1} \end{pmatrix}, \ldots, \begin{pmatrix} y_{n-m} \\ \vdots \\ y_{n-1} \end{pmatrix} \right] \in \left(\mathbb{F}_q^m\right)^N$

as well as an agreement parameter t, and a parameter D, and $k \leq n$.
and a parameter $s \leq m$.

Output: A list $\mathcal{L}$ containing all polynomials $f(X) \in \mathbb{F}_q[X]$ of degree $< k$, so that for at least t values of $i \in [0, N-1]$,

$$\begin{pmatrix} f(\gamma^{mi}) \\ \vdots \\ f(\gamma^{mi+m-1}) \end{pmatrix} = \begin{pmatrix} y_{mi} \\ \vdots \\ y_{mi+m-1} \end{pmatrix}$$

STEP 1. (Interpolation)

Find a nonzero polynomial

$$Q(X, Y_1, Y_2, \ldots, Y_s) = A_0(X) + A_1(X) \cdot Y_1 + \cdots + A_s(X) \cdot Y_s$$

with $\deg(A_0) \leq D + k - 1$, and $\deg(A_i) \leq D$ for $i = 1, \ldots, s$ so that

$$Q(\gamma^{im+j}, y_{im+j}, \ldots, y_{im+j+s-1}) = 0 \quad \forall\ 0 \leq i < N \quad \forall\ 0 \leq j \leq m-s.$$

STEP 2. (Root-Finding)

Find all the polys f so that

$$Q(X, f(X), f(\gamma X), \ldots, f(\gamma^{s-1} X)) \equiv 0,$$

and return them.

Again, we go through our steps i, ii, iii, iv.

This is what t should be to get the result we claimed at the beginning.

i. SET PARAMETERS: Suppose

$$D = \left\lfloor \frac{N(m-s+1) - k + 1}{s+1} \right\rfloor, \quad t > \left\lfloor \frac{D + k - 1}{m - s + 1} \right\rfloor$$

ii. We can find Q. FUN EXERCISE! (exactly the same as usual)

iii. STEP 2 is a good idea.

OUTLINE:

FUN EXERCISE! Fill in the details.

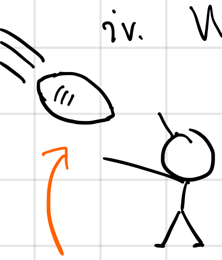
- If $\mathrm{FOLD}_m(f)$ agrees w/ y in $\geq t$ places, then

$$R(X) := Q(X, f(\gamma X), \ldots, f(\gamma^{s-1} X))$$

has at least $\underbrace{t}_{t \text{ agreement pts}}(\underbrace{m-s+1}_{\#\text{shifts per agreement pt}})$ roots.

- The degree of $R(X)$ is $< t(m-s+1)$ with our choices of parameters.

iv. We can efficiently find all polys f so that $Q(X, f(X), \ldots, f(\gamma^{s-1}X)) \equiv 0$.



This is supposed to be the football we punted earlier.

The way we would normally do this is argue that since the degree of Q is small there can't be too many f's.
That breaks down a bit here since each instance of f is a little different.

Instead, we'll go back to LINEAR ALGEBRA.

Here, we will just sketch how this goes.
See Chapter 14 of ESSENTIAL CODING THEORY for details.

iv. continued.

We want to find all f so that

$$(*) \quad R(X) := Q(X, f(X), f(\gamma X), \dots, f(\gamma^{s-1} X)) \equiv 0.$$

Say that $f(X) = f_0 + f_1 X + f_2 X^2 + \dots + f_{k-1} X^{k-1}$

We can write $(*)$ as a giant linear equation in the coefficients $f_0, f_1, \dots, f_{k-1}$.

QUESTION: What is the constant term in $R(X)$?
That is, if $R(X) = \sum_j r_j X^j$, what is r_0?

ANSWER 1: $r_0 = 0$, since $R \equiv 0$.

ANSWER 2: Okay, let's compute it. Recall that

$$R(X) = Q(X, f(X), \dots, f(\gamma^{s-1} X)) = A_0(X) + f(X) \cdot A_1(X) + \dots + f(\gamma^{s-1} \cdot X) \cdot A_s(X),$$

and so

$$r_0 = R(0)$$

Here, $A_i(X) = \sum_j a_{ij} X^j$

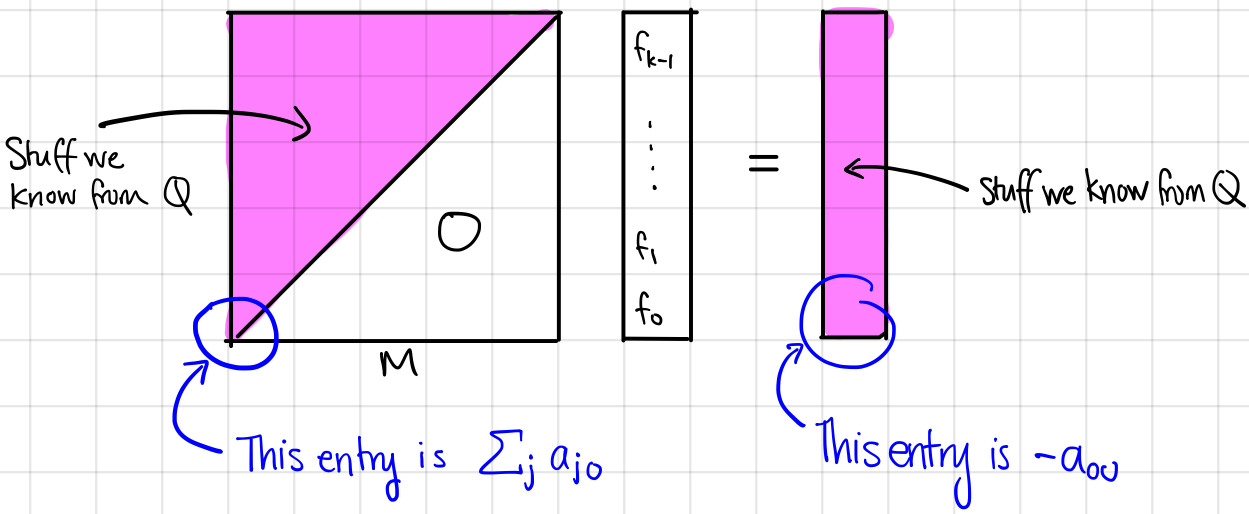
$$= A_0(0) + f(0) \cdot A_1(0) + \dots + f(0) \cdot A_s(0)$$

$$= a_{00} + \sum_{j=1}^{s} a_{j0} \cdot f_0$$

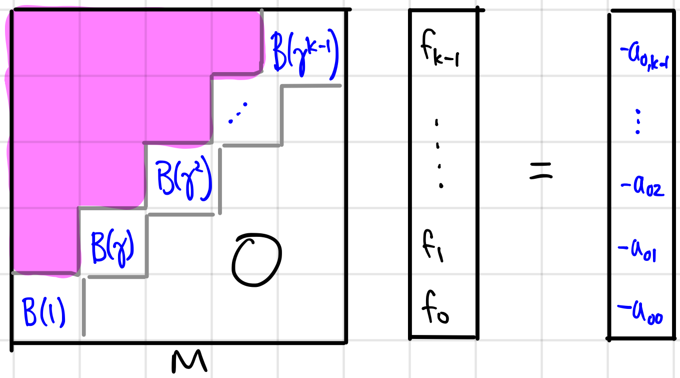
$$= a_{00} + f_0 \left(\sum_{j=1}^{s} a_{j0} \right)$$

So actually we know f_0! $\quad f_0 = \left(\frac{-a_{00}}{\sum_{j=1}^{s} a_{j0}} \right).$

It turns out that we can keep doing this, and moreover it turns out that the linear system that we get is triangular.



Even better, we can exactly figure out what goes on the diagonal:



Where $B(X) = a_{10} + a_{20} X + a_{30} X^2 + \cdots + a_{s,0} X^{s-1}$

B has at most $s-1$ roots

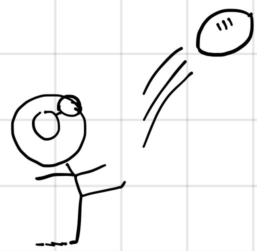
$\Rightarrow$ M has at most $s-1$ O's on the diagonal

$\Rightarrow$ $\dim(\mathrm{Ker}(M)) \le s-1$

$\Rightarrow$ There are at most q^{s-1} solutions to this linear system.

That's what we wanted!

So, modulo the details, we've proved the THM from the beginning of the lecture.



Check out ESSENTIAL CODING THEORY, Chapter 14, for the details!

QUESTIONS to PONDER

1. Work out the details for part (iv)
2. Can you get a smaller list size for FRS codes?
3. Can you extend this algorithm do do list recovery?