The Guruswami-Sudan Algorithm.

- Berlekamp-Massey algorithm, algebraic in approach, is probably the most famous decoding algorith for RS codes.
- ~~Our~~ The GS algorithm is, however, particularly interesting because it can decode RS codes often much beyond the conventional error-correcting bounds ($\sim d/2$).
- Typically, we are guaranteed to sucessfully decode $t_0 = \lfloor (d-1)/2 \rfloor$ errors. (Recall our discusion with spheres).
- However, some spheres could have larger radius and still not intersect other spheres.
- In many situations, the probability that by increasing the radius of sphere ~~and~~ we would decode words incorrectly can be very small

which gives us the ability to correct upto $t_{GS} = n - 1 - \lfloor \sqrt{n(k-1)} \rfloor$ errors without making the probability of incorrectly decoding prohibitively large.

Rough Sketch of Algorithm

- We have transmitted a codeword $c = (f(\alpha^0), f(\alpha^1), \ldots f(\alpha^{n-1}))$ and we have received the word $r = (\beta_0, \beta_1, \ldots \beta_{n-1})$.

The original codeword was found by evaluating polynomial $f(z)$ with deg $< k$ at the n non zero field elements, $\alpha^0, \alpha^1, \ldots \alpha^{n-1}$.

So, By definition, coefficients in polynomial $f(z)$ correspond directly to components of original vector in F_q^n which we wish to encode.

Given γ, the algorithm indicates that we can find a polynomial $p(x)$ so that:

$$|\{i : p(\alpha^i) \neq \beta_i\}| \leq t_m$$

where, t_m is number of errors we can decode and,

$$t_m \geq t_0 = \lfloor d/2 \rfloor .$$

We have the variable m which specifies the multiplicity to be used in the interpolation step.

- We can divide the algorithm in two part:

1) The Interpolation step:
Given $\gamma = (\beta_0, \beta_1, \ldots \beta_{n-1})$, we construct a two-variable polynomial

$$Q(x, y) = \sum_{i,j} a_{i,j} x^i y^j$$

So that Q has a zero of multiplicity m at each point

(α^i, β_i) and the $(1, k-1)$ weighted degree of $Q(x,y)$ is as small as possible.

2) The factorization step:

We then find all factors of $Q(x,y)$ of the form $y - p(x)$ where $p(x)$ is a polynomial of degree $(k-1)$ or less. The list of all these factors is

$$L = \{p_1(x), p_2(x) \ldots\ldots - p_L(x)\}$$

L is a set of polynomials corresponding with a vector or codeword in F_q^n. It can be:

a) The transmitted codeword.

b) Plausible codewords with hamming distance less than (or equal) t_m from y.

c) Implausible codewords with distance $> t_m$ from y.

If less than t_m errors have occured, L will include original word. If there is only one codeword (i.e. polynomial) in L, then we are done. If there are more than one codeword in L then we must decide which one we should decode.

Some Background Theory:

Monomial Ordering:

- For a two variable poly $Q(x,y) \in k[x,y]$,

$$Q(x,y) = \sum_{i,j \geq 0} a_{ij} x^i y^j$$

- It will be helpful to have a one-dimensional ordering on the monomials. We denote the set of all monomials of two variables as:

$$M[x,y] = \{x^i y^j : i,j \geq 0\}$$

- Note that $M[x,y]$ is isomorphic to $\mathbb{N}^2$ under the bijection

$$(x^i y^j) \longmapsto (i,j).$$

- Define a monomial ordering '$<$' over a set M.

If $a, b, c \in M^2$, then:

1) $a_1 = b_1$ & $a_2 \leq b_2 \Rightarrow (a_1, a_2) < (b_1, b_2)$ where $\bar{a} = (a_1, a_2)$ & $\bar{b} = (b_1, b_2)$

2) If a & b are distinct then, $a < b$ or $b < a$

$\Rightarrow a \leq b \Rightarrow a+c \leq b+c$.

- Now, we will use weighted degree orderings.

Let $\omega = (u, v)$ where u & v are non negative integers and at least one is non zero.

- Def: For a fixed $\omega = (u, v)$,

$$\deg_\omega x^i y^j = ui + vj$$

Eg: Say $\omega_1 = (1, 3)$; $\omega_2 = (0, 1)$.

$\therefore \deg_{\omega_1} x^2y = 2\times1+3 = 5$, $\deg_{\omega_2} x^2y = 2\times0+1 = 1$,

& $\deg_{\omega_1} x = 1$, $\deg_{\omega_2} x = 0$,

& $\deg_{\omega_2} x^2 = 2$, $\deg_{\omega_2} x^2 = 0$.

Note that $\deg_{\omega_2} x^2 = \deg_{\omega_1} x$ ... and so these are not ~~lexico~~ lexicographical orderings.

- Define: ω-lex order as

$$x^{i_1} y^{j_1} < x^{i_2} y^{j_2}.$$

If either $ui_1 + vj_1 < ui_2 + vj_2$

or, $ui_1 + vj_1 = ui_2 + vj_2$ and $i_1 < i_2$.

(Also for ω-revlex, $i_1 > i_2$)..

- Eg: $\omega = (1,3)$

$x < x^2 < y < x^3 < xy < x^4 < x^2y < x^2 < y^2 < x^3$ ...

ω-lex

$x < x^2 < x^3 < y < x^4 < xy < x^5 < x^2y < x^6 < x^3y < y^2 < \ldots$

ω-revlex

- $\omega = (1,1) \rightsquigarrow$ Graded-lex (grlex) and reverse graded-lex (grevlex).

$\omega = (0,1) \rightsquigarrow$ y-ordering.

- We can define a monomial ordering as

$1 = \phi_0(x,y) < \phi_1(x,y) < \ldots$,

ϕ_k corresponds to a unique monomial.

Then $Q(x,y) = \sum_{j=0}^{J} a_j \phi_j(x,y)$.

Under this ordering, the leading monomial (LM) is

$$LM(Q) = \phi_j(x, y).$$

So, $\deg Q(x, y) = \deg LM(Q)$

$$= \deg \phi_j(x, y).$$

So rank of $Q(x, y)$ is J.

- We will also define $\text{Ind}(\phi) = k$ such that $\phi_k = x^i y^j$ where $\phi_0 < \phi_1 < \cdots$.
- We now define the following notation,

$A(k, v) = \text{Ind}(x^k)$ &

$B(k, v) = \text{Ind}(y^L)$

where we use $(1, v)$-revlex order.

- ∵ x^k is first monomial of $(1,v)$-degree k & y^L is last monomial of $(1,v)$-degree vL,

$$A(k,v) = \left|\{(i,j) : i + vj < k\}\right|$$

$$B(L,v) = \left|\{(i,j) : i + vj \le Lv\}\right| - 1$$

- For $k \ge 0$, let $r = k \bmod v$.

Then, $$A(k,v) = \frac{k^2}{2v} + \frac{k}{2} + \frac{r(v-r)}{2v}$$

$$B(L,v) = \frac{vL^2}{2} + \frac{(v+2)L}{2}.$$

- The ~~first~~ second one can be proved by induction. (~~Not so~~ Trivial)
For the first one we can apply Euler's summation formula. (Not so trivial).

- Now, let us denote the degree of polynomial as

$$D(u, v; J) = \max \{ \deg_{u,v} \phi_j(x,y) : j = 0,1 \ldots J \}$$

(where $Q(x,y) = \sum_{j=0}^{J} q_j \phi_j(x,y)$).

- Now we state upper bounds for the $(0,v)$-degree and y-degree of $Q(x,y)$ as

$$\deg_{1,v} Q(x,y) \leq D(1, v; J).$$

and $\deg_{0,1} Q(x,y) \leq D(0,1; J)$

- Now consider $A = \{ 0 = a_0 < a_1 < a_2 < \ldots \}$ a sequence of integers and a real number $x \geq 0$.
We want the greatest integer in A say n that if is $\leq x$.
We refer to its index as 'the rank of apparition' of

x with respect to A, denoted as $r_A(x) = k$ such that $a_k \leq x < a_{k+1}$.

- **Theorem:** With v fixed, define sequences $\{a_k = A(k, v)\}$ and $\{b_L = B(L, v)\}$. Then,

$$D(1, v : J) = r_A(J).$$

$$D(0, 1 : J) = r_B(J).$$

Proof: Observe that x^k is the first monomial of $(1, v)$-degree k and y^L is first monomial of $(0, 1)$-degree L.

- For $v \geq 1$, $k \geq 0$,

$$\frac{k^2}{2v} \leq A(k,v) \leq \frac{(k+v/2)^2}{2v}$$

(can be proved easily from $A = \frac{k^2}{2v} + \frac{k}{2} + \frac{v(v-2)}{2v}$)

- For $v \geq 1$, $J \geq 0$,

$$\left\lfloor \sqrt{2vJ} - \frac{v}{2} \right\rfloor \leq r_A(J) \leq \lfloor \sqrt{2vJ} \rfloor - 1$$

- For $v \geq 1$, $J \geq 0$

$$r_B(J) = \left\lfloor \sqrt{\frac{2J}{v} + \left(\frac{v+2}{2v}\right)^2} - \left(\frac{v+2}{2v}\right) \right\rfloor$$

So, $$\left\lfloor \sqrt{\frac{2J}{v}} - \frac{v+2}{2v} \right\rfloor \leq r_B(J) \leq \left\lfloor \sqrt{\frac{2J}{v}} \right\rfloor$$

- The above two results can be proved using the previous results and the following lemma:

If $a_k = f(k)$, $f(x)$ is a cts increasing function of $x > 0$

then, $r_A = \lfloor f^{-1}(x) \rfloor$.

Also, if $g(k) \le a_k \le f(k)$,
$f(x)$ & $g(x)$ are cts increasing

then $\lfloor f^{-1}(x) \rfloor \le r_A(x) \le \lfloor g^{-1}(x) \rfloor$

Proof: Let $k = r_A(x)$.

Then, $g(k) \le a_k \le x < a_{k+1} \le f(k+1)$.

So, $\underbrace{k}_{r_A(x)} \le g^{-1}(x)$ & $f^{-1}(x) < \underbrace{k}_{r_A(x)} + 1$

$\therefore f^{-1}(x) - 1 < r_A(x) < g^{-1}(x)$

Since $r_A(x)$ is an integer, we get our result.

Multiplicity of Zero.

- We say than $Q(x,y) = \sum a_{i,j} x^i y^j \in F[x,y]$, has a zero of multiplicity of order m at $(0,0)$ and write.

$$ord(\varphi : 0,0) = m.$$

if $Q(x,y)$ involves no term of total degree less than m.
i.e. $a_{i,j} = 0$ if $i+j < m$.

- Similarly, $ord(Q : \alpha, \beta) = m$ if $Q(x+\alpha, y+\beta)$ has a zero of order m at $(0,0)$.

- Hasse derivatives:

$$Q(x,y) = \sum_{i,j} a_{i,j} x^i y^j \in F[x,y].$$

for any $(\alpha, \beta) \in F^2$, we

$$Q(x+\alpha, y+\beta) = \sum_{r,s} Q_{r,s}(\alpha,\beta) x^r y^s$$

where

$$Q_{r,s}(x,y) = \sum_{i,j} \binom{i}{r}\binom{j}{s} a_{ij} x^{i-r} y^{j-s}$$

$$\forall r,s \text{ s.t. } 0 \le r+s < m$$

which is called (r,s)th Hasse / mixed partial derivative of $Q(x,y)$.

Note that,

$$Q_{r,s}(\alpha,\beta) = \text{Coeff}_{x^r y^s} Q(x+\alpha, y+\beta)$$

~~Proof: By Binomial expansio~~

Proof: By binomial expansion of $Q(x+\alpha, y+\beta) = \sum_{i,j} a_{ij}(x+\alpha)^i (y+\beta)^j$

- $Q(x,y) = \sum_{r,s} Q_{r,s}(\alpha,\beta)(x-\alpha)^r (y-\beta)^s$.
- Theorem: The polynomial $Q(x,y)$ has a zero of order m at (α,β) iff.

$$Q_{r,s}(\alpha,\beta) = 0 \quad \forall \ r \text{ and } s \text{ s.t. } 0 \le r+s < m.$$

Proof: $\text{ord}(Q;\alpha,\beta) \ge m$ iff $Q(x+\alpha, y+\beta)$ has a zero of order m at $(0,0)$. But by the above results, $Q(x+\alpha, y+\beta)$ has a zero of order m at $(0,0)$ iff $Q_{r,s}(\alpha+\beta) = 0$ for all $0 \le r+s < m$.

Interpolation Theorem

- Sent vector → $(f(\alpha^0), f(\alpha^1), \ldots f(\alpha^{n-1}))$
  Recieved → $(\beta_0, \beta_1, \ldots \beta_{n-1})$.
  (may contain errors).

We need to find the word corresponding to $f(x)$.

- Find a list of polynomials so that when we evaluate them at field elements, we are close to recieved vector.

First we will need to build a two-variable polynomial over F_q by interpolating on pts
$\{(\alpha^0, \beta_0), (\alpha^1, \beta_1), \ldots (\alpha^{n-1}, \beta_{n-1})\}$

Interpolation is simply finding a function (in fact polynomial) which matches a set of data points.

- The Interpolation Theorem:
  Let $m(\alpha,\beta) : (\alpha,\beta) \in F_q^2$ be a multiplicity function and let $\phi_0 < \phi_1 < \dots$ be an arbitrary monomial order. Then there exists a nonzero polynomial $Q(x,y)$ of the form $Q(x,y) = \sum_{i=0}^{C} a_i \cdot \phi_i(x,y)$ — (A)
  where $C = \sum_{\alpha,\beta} \binom{m(\alpha,\beta)+1}{2}$
  which has a zero of multiplicity $m(\alpha,\beta)$ at $(x,y) = (\alpha,\beta)$ for all $(\alpha,\beta) \in F_q^2$.

Proof:

$Q(x,y)$ has a zero of multiplicity m at (α,β) if and only if

$Q_{r,s}(\alpha,\beta)=0 \;\forall\; (r,s)$ such that

$$0 \le r+s < m(\alpha,\beta).$$

Now, there are $\binom{m(\alpha,\beta)+1}{2}$ choices for (r,s) where and from the Hasse derivative, each such choice imposes one homogeneous linear constraint on the coefficients a_i. In total, there are C such linear constraints imposed on $(+1)$ coefficients $a_0, a_1, \ldots a_C$. So there must be at least one non zero solution to this set of equations

which corresponds to a non-zero polynomial $Q(x,y)$ of the form (A) with required multiplicities.

- **Corollary:**

For any $(1,v)$, there exists a non-zero polynomial $Q(x,y)$ with the required zero multiplicities whose $(1,v)$-degree is strictly less than $\sqrt{2vC}$.

Proof: Make $\{\phi_j(x,y)\}_j$ to be $(1,v)$-weighted order.

So, $\deg_{1,v} Q(x,y) \le \max\{\deg_{1,v}\phi_j(x,y) : j=0,\dots C\}$

$= \deg_{1,v}\phi_C(x,y) = \gamma_A(C)$

where $A = (a_{ik})$ is sequence such (x^k) for $(1,v)$-weighted order.

But $\gamma_A \le \sqrt{2vJ}$ (From previous result)

Factorization Theorem

Q-score: We define Q-score of a function $f \in F[x]$ in relation to $Q(x,y) \in F[x,y]$ as

$$Sq(f) = \sum_{\alpha \in F} ord(Q : \alpha, f(\alpha))$$

Q score represents how close $(f(\alpha^0), f(\alpha^1), \ldots, f(\alpha^{n-1}))$ is to $(\beta_0, \beta_1, \ldots \beta_{n-1})$ for any f we choose.

- **Theorem:**

Suppose $f(x) \in F[x]$ of degree less than N, $Q(x,y) \in F[x,y]$ and $Sq(f) > \deg_x Q$

Then $y - f(x)$ is a factor of $Q(x,y)$.

Proof:

Let $Q(x,y) = \sum_{i,j} a_{i,j} x^i y^j$

then $Q(x, f(x))$ is a polynomial in x

$$Q(x, f(x)) = \sum_{i,j \geq 0} a_{ij} x^i f(x)^j$$

Now we will require the following results

1) If $f(x) \in F[x]$ of degree less than v then $\deg Q(x, f(x)) \leq \deg_{1,v} Q(x,y)$.

2) $Q(x, f(x)) = 0$ iff $(y - f(x)) \mid Q(x,y)$.

3) If $\operatorname{ord}(Q : \alpha, \beta) = k$ and $f(\alpha) = \beta$.

then, $(x-\alpha)^k \mid Q(x, f(x))$

So, $\prod_{\alpha \in F} (x-\alpha)^{ord(Q:\alpha, f(\alpha))}$

divides $Q(x, f(x))$ (By ③).

But by ①, degree of $Q(x, f(x))$ is at most $deg_{1,v} Q(x,y)$ and degree of $\prod_{\alpha \in F} (x-\alpha)^{ord(Q:\alpha, f(\alpha))}$ is $S_Q(f)$.

Thus, if $S_Q(f) > deg_{1,v} Q$. then $Q(x, f(x)) = 0$ and therefore from ②, $y - f(x)$ divides $Q(x,y)$.

GRS Decoder

- $k(f, \beta) = |\{i : f(\alpha_i) = \beta_i\}|$

  $D(f, \beta) = |\{i : f(\alpha_i) \neq \beta_i\}|$

  $\left(= n - k(f, \beta)\right)$

(k is number of places they agree and D is disagree).

- $\binom{n}{2} = n\left(\frac{m+1}{2}\right) = \frac{nm(m+1)}{2}$.

  which will be the upper range of two-variable polynomial we will construct.

- $k_m(n, k) = \min\{k : A(mk, k) \geq \binom{mk}{2}\}$

  $= 1 + \lfloor r_A(C)/m \rfloor$

- $A_m(n, k) = n - k_m(n, k)$

  $= n - 1 - \lfloor r_A(C)/m \rfloor$

$$L_m(n,k) = \max\{L : B(L,v) \le (k,n)\}$$
$$= \partial_B(C).$$

- Also we have the following bounds,

$$\left\lfloor \sqrt{vn\frac{m+1}{m}} - \frac{v}{2m} \right\rfloor + 1 \le k_m \le \left\lfloor \sqrt{\frac{vn(m+1)}{m}} \right\rfloor - 1$$

$$n - \left\lfloor \sqrt{vn \cdot \frac{m+1}{m}} \right\rfloor \le d_m \le n - 1 - \left\lfloor \sqrt{vn\frac{m+1}{m}} - \frac{v}{2m} \right\rfloor$$

$$L_m = \left\lfloor \sqrt{\frac{n}{v} m(m+1) + \left(\frac{v+2}{2v}\right)^2} - \frac{v+2}{2v} \right\rfloor < \left(m+\frac{1}{2}\right)\sqrt{\frac{n}{v}}$$

The GS(m) decoder:

It creates a nonzero two-variable polynomial of the form.

$$Q(x,y) = \sum_{j=0}^{c(n,m)} a_j \phi_j(x,y)$$

where $\phi_0 < \phi_1 < \dots$ is a $(1,v)$-weighted monomial order such that $Q(x,y)$ has a zero of order m at each of the n points (α_i, β_i).

The output of the algorithm is the list of y-roots of $Q(x,y)$ i.e. $L = \{f(x) \in F[x] : (y - f(x)) \mid Q(x,y)\}$

Theorem: The output list L contains every polynomial of degree $\leq v$ such that $k(f,\beta) \geq k_m$.
Furthermore, the number of polynomials in the list is at most L_m.

Proof:

$$\deg_{1,v} Q(x,y) \leq \max\{\deg_{1,v} \phi_i(x,y) : i = 0, \ldots\}$$
$$= \gamma_A(C).$$

So any polynomial $f(x)$ of degree $e \leq v$ such that $mult(f,\beta) > \gamma_A(C)$ will be a y-root of $Q(x,y)$.

So, if $k(f,\beta) \geq 1 + \lfloor \gamma_A(C)/m \rfloor = k_m$, $f(x)$ will be on the list.

Also, y-degree of $Q(x,y) \leq \gamma_B((m,m)) = L_m$ since number of y-roots can't exceed its y degree, output list contains at most L_m polynomials.

- So, if $\leq k_m$ errors occurs in $S(m)$ decoded would include the correct word in L.

Note that as m increases, we may decode more errors but we need to perform more computation.

Decoding Algorithm

We are given n, r, field F_q and multiplicity m.
The decoder recieves the input:
$\beta = (\beta_0, \beta_1, \dots \beta_{n-1}) \in F_q^n$.

1/ We build $Q(x,y)$ with zeroes of multiplicity $m(\alpha,\beta) = m$ at points $(\alpha^0, \beta_0), (\alpha^1, \beta_1) \dots (\alpha^{n-1}, \beta_{n-1})$ and at every other point, $m(\alpha,\beta) = 0$. Note that m is directly related to number of errors we can correct. Also we have singularity at these points.
We build $Q(x,y)$ as per the interpolation theorem

$$Q(x,y) = \sum_{i=0}^{c} a_i \phi(x,y)$$

where,

$$C = \sum_{\alpha,\beta} \binom{m(\alpha,\beta)+1}{2}$$

$$= n \binom{m+1}{2} = \frac{nm(m+1)}{2}.$$

and,

$q(x,y)$ is $(1, r-1)$-revlex monomial ordering.

Now we have the structure of the polynomial but not the coefficients i.e a_i's.

However, we know that $Q(x,y)$ will have multiplicities at the above points if and only if

$$Q_{r,s}(\alpha,\beta) = \sum_{i,j} \binom{i}{r}\binom{j}{s} a_{i,j} x^{i-r} y^{j-s} = 0$$

$$\forall r, s \text{ s.t. } 0 \le r+s < m$$

Now since there are $\binom{m+1}{2}$ equations defined by each point and there are n points for a total of $c = n\binom{m+1}{2}$ equations.

Since we have c independent equations and $c+1$ unknowns (the coefficients). and system is homogeneous, there is at least one non-trivial solution to this system of equations. So voila, we can now construct our function $Q(x, y)$.

2.7 Now, we take polynomials $f(x) \in F_q[x]$ with degree less than k and find their Q-score.

(Recall that $S_Q(f) = \sum_{\alpha \in F} ord(Q(\alpha, f(\alpha))$ and it is a measure of how close $f(\alpha^0), f(\alpha^1) \ldots f(\alpha^{n-1})$ is to $(\beta_0, \beta_1, \ldots \beta_{n-1})$.)

If less than t_m errors have occurred, one of these represents the original data. The Q-score gives the number of points at which the function $f(x)$ agrees with the one which we interpolated to attain $Q(x, y)$.

Now, we compare Q score of each polynomial $f(x)$ to the $(1, v)$-weight of $Q(x, y)$:

$$\deg_{1,v} Q(x,y) = \deg_{1,v} \phi_c(x,y).$$
$$= \deg_{1,v} x^I y^J = I + vJ.$$

For each $f(x) \in F_q[x]$ with degree less than k, if $S_Q(f) > I + vJ$, we add $f(x)$ to the list L.

So if less than t_m errors have occured then list L is guaranteed to include the original word which was sent.

The algorithm returns L as its result!

A Example

We will work though a complete example from start to finish over the very simple $(3, 2, 2)$ RS code described in Section 3. Since $n = q - 1$, we will be working over the field $\mathbb{F}_4$. The code is not guaranteed to decode any errors, since the conventional error-correction bound is $\lfloor \frac{d-1}{2} \rfloor = 0$. However, we will find that if 1 error occurs we can narrow down the result to a list of three codewords. The multiplicity, m, is used throughout the decoding algorithm; for our example we will set $m = 2$.

A.1 Definitions

The first step is to find the monomial ordering that we will be using. Since $v = k - 1 = 1$, we will use $(1, 1)$-*revlex*:

$$1 < x < y < x^2 < xy < y^2 < x^3 < x^2y < xy^2 < y^3 < ...$$

By Theorem 4.2, we can find the sequences r_A and r_B:

$$r_A = D(1, 1) = \{0, 1, 1, 2, 2, 2, 3, 3, 3, 3, ...\},$$

$$r_B = D(0, 1) = \{0, 0, 1, 0, 1, 2, 0, 1, 2, 3, ...\}.$$

In Section 4.6 we introduce the following values:

$$C(n, m) = n\binom{m+1}{2} = 9$$

$$t_m(n, k) = n - 1 - \lfloor r_A(C)/m \rfloor = 2 - \lfloor r_A(9)/2 \rfloor = 2 - \lfloor 3/2 \rfloor = 1$$

$$L_m = r_B(C) = r_B(9) = 3$$

A.2 Encoding and Transmission

We will be encoding vectors of the form $(\alpha_0, \alpha_1) \in \mathbb{F}_4^2$ giving us 16 possible words and so each can correspond with a hexadecimal digit. Consider that we want to send the hexadecimal digit $0x7$. In binary, we have $0b0111$, and so we encode the vector $(1, \alpha^2)$. The encoding map in the Section 3 indicates that $(1, \alpha^2) \mapsto (\alpha, 0, \alpha^2)$. This can be established by representing $(1, \alpha^2)$ as a polynomial $f(z) = 1 + (\alpha^2)z$ and evaluating it at the non-zero elements of $\mathbb{F}_4$: $(f(\alpha^0), f(\alpha), f(\alpha^2)) = (\alpha, 0, \alpha^2)$.

To simulate transmitting the encoded word across a noisy channel we will modify the second component of the vector, yielding $\mathbf{r} = (\alpha, 1, \alpha^2)$.

A.3 Decoding

We receive the word $\mathbf{r} = (\alpha, 1, \alpha^2)$. We will closely follow the steps indicated in Section 4.7 to determine a list of possible words. We must build the polynomial $Q(x, y)$ by interpolating at the points $(1, \beta_0)$, (α, β_1),

and (α^2, β_2) where $\mathbf{r} = (\beta_0, \beta_1, \beta_2)$. Each point corresponds to a non-zero element in $\mathbb{F}_4$. In our case the points are: $\{(1, \alpha), (\alpha, 1), (\alpha^2, \alpha^2)\}$. Our goal is to find all polynomials of degree less that k such that when they are evaluated at the non-zero elements of $\mathbb{F}_4$ the result 'closely matches' our received vector $\mathbf{r}$. To do this we build the polynomial $Q(x, y)$ so that it has a zero of multiplicity $m = 2$ at each of the interpolating points. We can write the polynomial $Q(x, y)$ as

$$Q(x, y) = \sum_{i=0}^{C(n,m)} a_i \phi_i(x, y)$$

$$= a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 + a_6 x^3 + a_7 x^2 y + a_8 x y^2 + a_9 y^3$$

Each monomial ϕ_i is the ith monomial in the $(1, 1)$-*revlex* ordering listed above. We must find the coefficients. Theorem 4.9 states that $Q(x, y)$ has a zero of multiplicity m at each of these interpolating points if the following is satisfied:

$$Q_{r,s}(\alpha, \beta) = \sum_{i,j} \binom{i}{r} \binom{j}{s} a_{i,j} x^{i-r} y^{j-s} = 0 \qquad \forall r, s \text{ s.t. } 0 \leq r + s < m(\alpha, \beta) \qquad \forall \alpha, \beta \in \mathbb{F}_4$$

Where $(i, j) \in \{(I, J) : 0 \leq \mathrm{Ind}(x^I, y^J) < C(n, m) = 9\}$.

Since $m(\alpha, \beta) = 0$ for every point except the interpolating points, we will use just those interpolating points. At each of those three points, there are three choices for r and s: $\{(0, 0), (1, 0), (0, 1)\}$ yielding a total of nine equations: $Q_{0,0}(1, \alpha) = 0$, $Q_{0,0}(\alpha, 1) = 0$, $Q_{0,0}(\alpha^2, \alpha^2) = 0$, $Q_{0,1}(1, \alpha) = 0$, $Q_{0,1}(\alpha, 1) = 0$, $Q_{0,1}(\alpha^2, \alpha^2) = 0$, $Q_{1,0}(1, \alpha) = 0$, $Q_{1,0}(\alpha, 1) = 0$, $Q_{1,0}(\alpha^2, \alpha^2) = 0$. The following matrix represents the equations, with each column corresponding to a coefficient, and each row corresponding to one of the previous equations.

$$\begin{bmatrix}
1 & 1 & \alpha & 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 & 0 \\
1 & \alpha & 1 & \alpha^2 & \alpha & 1 & 1 & \alpha^2 & \alpha & 1 & 0 \\
1 & \alpha^2 & \alpha^2 & \alpha & \alpha & \alpha & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & \alpha^2 & 0 \\
0 & 0 & 1 & 0 & \alpha & 0 & 0 & \alpha^2 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & \alpha^2 & 0 & 0 & \alpha & 0 & \alpha & 0 \\
0 & 1 & 0 & 0 & \alpha & 0 & 1 & 0 & \alpha^2 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & \alpha^2 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & \alpha^2 & 0 & \alpha & 0 & \alpha & 0 & 0
\end{bmatrix}$$

In reduced echelon form, we have:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \alpha^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

We have one free coefficient, a_9. Since this is a homogeneous system we have a trivial solution (let $a_9 = 0$) and nontrivial solutions for $a_9 = 1, a_9 = \alpha, a_9 = \alpha^2$. We will let $a_9 = \alpha$. This yields the following values for the coefficients:

$$[\alpha, 0, 0, 0, 1, 0, \alpha, 0, 0, \alpha]$$

So we can write $Q(x, y)$ as:

$$Q(x, y) = (\alpha) + xy + (\alpha)x^3 + (\alpha)y^3$$

Having built $Q(x, y)$, we will find a list of possible sent words: they are the polynomials $f(x)$ such that $y - f(x)$ is a factor of $Q(x, y)$. The Factorization Theorem states that such polynomials will have a *Q-score* $> deg_{1,1}Q(x, y) = 3$. We take each vector in $\mathbb{F}_4^2$ and find the *Q-score* of the corresponding polynomial. If $S_Q(f) > 3$, we add the vector to our list $\mathcal{L}$. Remember that the max size of the list is $L_m = 3$.

elements $\in \mathbb{F}_4^2$	$f(x)$	$S_Q(f)$
$(0, 0)$	0	0
$(0, \alpha)$	αx	2
$(0, \alpha^2)$	$\alpha^2 x$	2
$(0, 1)$	x	2
$(\alpha, 0)$	α	2
(α, α)	$\alpha + \alpha x$	4
(α, α^2)	$\alpha + \alpha^2$	0
$(\alpha, 1)$	$\alpha + 1$	0
$(\alpha^2, 0)$	α^2	2
(α^2, α)	$\alpha^2 + \alpha x$	0
(α^2, α^2)	$\alpha^2 + \alpha^2 x$	0
$(\alpha^2, 1)$	$\alpha^2 + x$	4
$(1, 0)$	1	2
$(1, \alpha)$	$1 + \alpha x$	0
$(1, \alpha^2)$	$1 + \alpha^2 x$	4
$(1, 1)$	$1 + x$	0

So our decoder will return the following list of vectors:

$$\mathcal{L} = \{(\alpha, \alpha), (\alpha^2, 1), (1, \alpha^2)\}$$

As you can see the word that we sent, $(1, \alpha^2)$ is in the list, demonstrating the error-correcting capability of the code. Remember that in this case, we are using a code that conventionally cannot correct any errors. In larger codes the results can be much better than this – yielding exactly the word that was sent.