HW1

June 5, 2024

0.1 Finite Fields

Defines the finite field $\mathbb{F}_4$, with or without explicit a minimal polynomial, and lists the finite field elements.

[1]:
```
F4.<w> = GF(4)
F4
```

[1]: Finite Field in w of size 2^2

[2]:
```
for a in F4:
    print(a, a.polynomial().list())
```

```
0 []
w [0, 1]
w + 1 [1, 1]
1 [1]
```

[3]:
```
F4.<w> = GF(2^2, modulus=x^2 + x + 1)
F4
```

[3]: Finite Field in w of size 2^2

[4]:
```
for a in F4:
    print(a, a.polynomial().list())
```

```
0 []
w [0, 1]
w + 1 [1, 1]
1 [1]
```

Defines the Frobenius automorphism.

[5]:
```
Frob4 = F4.frobenius_endomorphism()
Frob4
```

[5]: Frobenius endomorphism w |--> w^2 on Finite Field in w of size 2^2

[6]:
```
Frob4(w), w^2
```

[6]: (w + 1, w + 1)

0.2 Example 1

[7]:
```
G1 = matrix(F4,[[1,0,0,1,w,w],[0,1,0,w,1,w],[0,0,1,w,w,1]])
G1
```

[7]:
```
[1 0 0 1 w w]
[0 1 0 w 1 w]
[0 0 1 w w 1]
```

Python indexing starts with 0.

[8]:
```
G1[0,:]
```

[8]:
```
[1 0 0 1 w w]
```

[9]:
```
G1[0,:] + G1[1,:] + G1[2,:]
```

[9]:
```
[1 1 1 1 1 1]
```

0.3 Exercise 1

Check numerically that for the code $\mathcal{C}$ of Example 1, $M_2(\mathcal{C}) = 3$.

[10]:
```
def Frobv(v):
    '''
        input: matrix row
        output: list
    '''

    # Frobenius
    Frob = (v.base_ring()).frobenius_endomorphism()

    sv = []
    for a in v[0]:
        sv.append(Frob(a))

    return sv
```

[11]:
```
G1[0,:],Frobv(G1[0,:])
```

[11]:
```
([1 0 0 1 w w], [1, 0, 0, 1, w + 1, w + 1])
```

[12]:
```
b1 = G1[0,:] + G1[1,:] + G1[2,:]
b2 = G1[0,:]
sb1 = Frobv(b1)
```

```
sb2 = Frobv(b2)

B = matrix(F4,[b1.list(),b2.list(), sb1, sb2])
B,B.rank()
```

[12]:
```
(
[    1     1     1     1     1     1]
[    1     0     0     1     w     w]
[    1     1     1     1     1     1]
[    1     0     0     1 w + 1 w + 1], 3
)
```

[13]:
```
C = []
for a in F4:
    for b in F4:
        for c in F4:
            C.append(matrix([a,b,c])*G1)
```

[14]:
```
len(C),C[0]
```

[14]:
```
(64, [0 0 0 0 0 0])
```

[15]:
```
for b1 in C[1:]:
    for b2 in C[1:]:
        # subcodes of dimension 2
        D = matrix(F4,[b1.list(),b2.list()])
        if D.rank() == 2:
            # applies Frobenius
            sb1 = Frobv(b1)
            sb2 = Frobv(b2)

            B = matrix(F4,[b1.list(),b2.list(), sb1, sb2])

            if B.rank() < 3:
                print(b1,b2)
```

0.4 Exercise 2

Use sage to reproduce Example 5, that is : * construct $\mathbb{F}_{5^4}$ (the same minimal polynomial may be used to double check the results but this is not critical to the exercise) * compute the factorization of $x^3-1, x^3-2, x^4-2, x^4-4$ * use x^3-1 to construct to cyclic codes of rank weight 1 * use x^3-2 to construct two cyclic codes of rank weight 1, and a cyclic code of rank weight at least 2.

[16]:
```
F54.<ww> = GF(5^4, modulus = x^4 + 4*x^2 + 4*x +2)
F54
```

[16]:
```
Finite Field in ww of size 5^4
```

[17]:
```
R.<y> = PolynomialRing(F54)
factor(y^3-1), ww^104, ww^520
```

[17]:
```
((y + 4) * (y + 2*ww^3 + 2*ww^2 + 2*ww + 3) * (y + 3*ww^3 + 3*ww^2 + 3*ww + 3),
 2*ww^3 + 2*ww^2 + 2*ww + 3,
 3*ww^3 + 3*ww^2 + 3*ww + 3)
```

[18]:
```
factor(y^3-2), ww^364, ww^572
```

[18]:
```
((y + 2) * (y + ww^3 + ww^2 + ww + 4) * (y + 4*ww^3 + 4*ww^2 + 4*ww + 4),
 4*ww^3 + 4*ww^2 + 4*ww + 4,
 ww^3 + ww^2 + ww + 4)
```

[19]:
```
factor(y^4-2), ww^39, ww^195, ww^351, ww^507
```

[19]:
```
((y + ww^3 + ww^2 + 4*ww) * (y + 2*ww^3 + 2*ww^2 + 3*ww) * (y + 3*ww^3 + 3*ww^2
+ 2*ww) * (y + 4*ww^3 + 4*ww^2 + ww),
 2*ww^3 + 2*ww^2 + 3*ww,
 4*ww^3 + 4*ww^2 + ww,
 3*ww^3 + 3*ww^2 + 2*ww,
 ww^3 + ww^2 + 4*ww)
```

[20]:
```
factor(y^4-4), ww^78, ww^234, ww^390, ww^546
```

[20]:
```
((y + ww^3 + ww^2 + ww) * (y + 2*ww^3 + 2*ww^2 + 2*ww) * (y + 3*ww^3 + 3*ww^2 +
3*ww) * (y + 4*ww^3 + 4*ww^2 + 4*ww),
 4*ww^3 + 4*ww^2 + 4*ww,
 3*ww^3 + 3*ww^2 + 3*ww,
 ww^3 + ww^2 + ww,
 2*ww^3 + 2*ww^2 + 2*ww)
```

We consider $x^3 - 1$, which will give cyclic codes of length 3. Then $g(x) = x + 4$ generates a cyclic code, and $x + 4$ is the codeword $(4, 1, 0)$ which has rank weight 1.

We consider $x^3 - 2$. There are 3 factors: $x + 2$, $x - w^{364}$, $x - w^{572}$. Then $x^3 - 2$ generates a first cyclic code of rank weight 1.

[21]:
```
(y-ww^364)*(y-ww^572)
```

[21]:
```
y^2 + 2*y + 4
```

This shows that $x - w^{364}$ has also a rank of 1. Now consider $g(x) = (x + 2)(x - w^{364})$.

[22]:
```
(y-ww^364)*(y+2)
```

[22]:
```
y^2 + (ww^3 + ww^2 + ww + 3)*y + 2*ww^3 + 2*ww^2 + 2*ww + 2
```

This code has dimension 1, so codewords are all multiples of

$$(2w^3 + 2w^2 + 2w + 2, w^3 + w^2 + w + 3, 1).$$

Since one coefficient is already in the ground field, no (non-zero) multiple of this codeword will have all coefficients in the group field.

0.5 Exercise 3

Illustrate Proposition 5, namely construct a cyclic code of length 6 over $\mathbb{F}_{q^m}$ (feel free to choose the field extension) such that $(x-1)$ does not divide its generator polynomial, and exhibit a codeword in $\mathbb{F}_q$.

[23]:
```
F72.<u> = GF(7^2)
S.<xx> = PolynomialRing(F72)
factor(xx^6-1)
```

[23]:
```
(xx + 1) * (xx + 2) * (xx + 3) * (xx + 4) * (xx + 5) * (xx + 6)
```

[24]:
```
(xx + 1) * (xx + 2) * (xx + 3) * (xx + 4) * (xx + 5)
```

[24]:
```
xx^5 + xx^4 + xx^3 + xx^2 + xx + 1
```

[25]:
```
(xx + 1) * (xx + 2) , (xx + 3) * (xx + 4) * (xx + 5)
```

[25]:
```
(xx^2 + 3*xx + 2, xx^3 + 5*xx^2 + 5*xx + 4)
```

Then $g(x) = x^2 + 3x + 2$ generates a cyclic code which has a codeword of rank 1, since for $c(x) = x^3 + 5x^2 + 5x + 4$, we have $g(x)c(x)$ corresponding to the codeword $(1, 1, 1, 1, 1, 1)$.