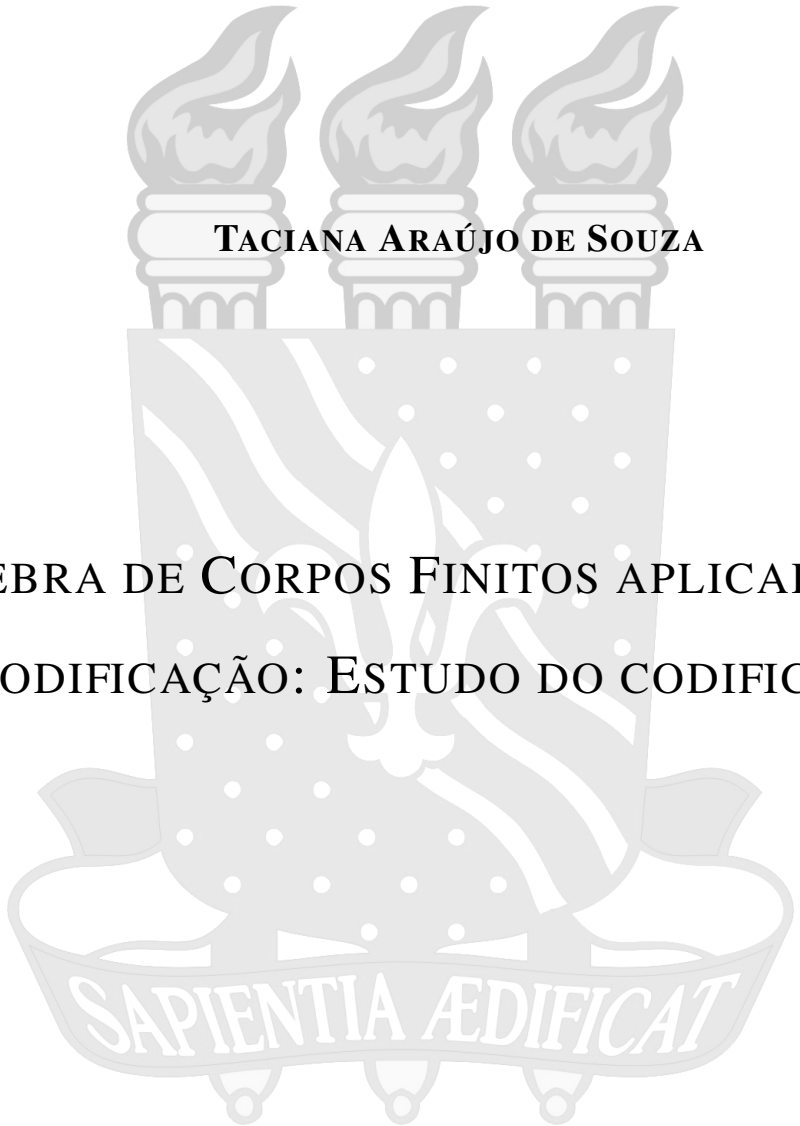
UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
DEPARTAMENTO DE MATEMÁTICA
CURSO DE BACHARELADO EM MATEMÁTICA



TACIANA ARAÚJO DE SOUZA

ÁLGEBRA DE CORPOS FINITOS APLICADA À TEORIA DA CODIFICAÇÃO: ESTUDO DO CODIFICADOR BCH.

JOÃO PESSOA, ABRIL DE 2012.

TACIANA ARAÚJO DE SOUZA

ÁLGEBRA DE CORPOS FINITOS APLICADA À TEORIA DA CODIFICAÇÃO: ESTUDO DO CODIFICADOR BCH.

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Bacharelado em Matemática da Universidade Federal da Paraíba como requisito para obtenção do título de Bacharel em Matemática.

ORIENTADOR: LUIZ GUEDES CALDEIRA, DR.

João Pessoa, Abril de 2012.

UNIVERSIDADE FEDERAL DA PARAÍBA

CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA

DEPARTAMENTO DE MATEMÁTICA

CURSO DE BACHARELADO EM MATEMÁTICA

Taciana Araújo de Souza

Álgebra de Corpos Finitos aplicada à Teoria da Codificação: Estudo do codificador BCH.

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Bacharelado em Matemática da Universidade Federal da Paraíba como requisito para obtenção do título de Bacharel em Matemática.

Prof. Eduardo Gonçalves dos Santos, Docteur.
Coordenador do Curso de Bacharelado em Matemática

Banca Examinadora:

Luiz Guedes Caldeira, Dr.
Orientador
Instituto Federal de Ciência e Tecnologia da Paraíba

Hélio Pires de Almeida, Dr.
Universidade Federal da Paraíba

Eduardo Gonçalves dos Santos, Dr.
Universidade Federal da Paraíba

DEDICATÓRIA

Dedico aos meus pais, Gabriel Francisco de Souza e Maria do Socorro Araújo, que são meus melhores amigos e que me apoiaram incondicionalmente na realização deste sonho.

AGRADECIMENTOS

Primeiramente agradeço a Deus pelo dom da vida, e aos meus pais, Gabriel Francisco de Souza e Maria do Socorro Araújo, aos quais devo minha vida, minha formação e grande parte de meus princípios.

Ao meu orientador, Dr. Luiz Guedes Caldeira, que foi incumbido da árdua tarefa de me conduzir nesse trabalho, a quem devo muito pela paciência, compreensão e disponibilidade em me orientar.

Ao coordenador do curso de bacharelado em Matemática, Dr. Eduardo Gonçalves dos Santos, pelo conhecimento que me passou durante os anos da graduação, além da paciência e compreensão nos momentos de dificuldades que tive durante o curso.

Aos bons amigos Thiago Machado (Thiagão), Eudes Mendes e Sheldon, amigos desde o início do curso, Rubicely e Gérsica, grandes amigas que ganhei ao longo do curso e quero agradecer especialmente aos amigos Suelen e Luan, que me ajudaram muito na execução deste trabalho.

Aos professores do curso de graduação em Matemática da UFPB que foram de grande importância na minha formação.

À minha querida avó Antônia Lopes de Souza, que me ensinou o valor da humanidade, respeito, e do carinho, que compreendeu mesmo com saudade minha ausência em alguns momentos.

Aos meus irmãos Alessandra, Arthur e Ellis Souza, que me deram amor e carinho me apoiando em todas as etapas da minha formação.

A todas as pessoas que, de alguma forma, contribuíram para a minha formação e que, portanto, indiretamente, viabilizaram a execução deste trabalho.

TACIANA ARAÚJO DE SOUZA

João Pessoa, 11 de Abril de 2012.

Resumo do Trabalho de Conclusão de Curso apresentado à UFPB como parte dos requisitos necessários para a obtenção do título de Bacharel em Matemática.

ÁLGEBRA DE CORPOS FINITOS APLICADA À TEORIA DA CODIFICAÇÃO: ESTUDO DO CODIFICADOR BCH.

Taciana Araújo de Souza

Abril/2012

Orientador: Luiz Guedes Caldeira, Dr.
Área de Concentração: Matemática Aplicada
Palavras-chaves: Teoria da codificação, Códigos corretores de erros, Códigos BCH
Número de páginas: X + 80

Este trabalho tem como objetivo estudar a aplicação da Matemática na Teoria da Codificação, através do estudo dos códigos corretores de erros BCH. Os códigos BCH são construídos utilizando a estrutura algébrica de corpos finitos e são caracterizados por operações sobre polinômios. A fim de compreender o funcionamento dos códigos BCH, iremos fazer uma breve introdução ao estudo de anéis, corpos, classes residuais, anéis de polinômios, corpos de Galois e espaços vetoriais. A seguir introduziremos os códigos de bloco lineares e os códigos cíclicos e, finalmente, será introduzida a classe especial de códigos cíclicos denominada BCH.

Abstract of coursework presented to UFPB as a partial fulfillment of the requirements for the degree of Bachelor of Mathematics

FINITE FIELD ALGEBRA APPLIED TO CODING THEORY: BCH ENCODER ANALYSIS

Taciana Araújo de Souza

April/2012

Supervisor: Luiz Guedes Caldeira, Dr.
Area of Concentration: Math Applications
Keywords: Coding Theory, Error Correcting Codes, BCH Codes
Number of pages: X +80

This work aims to study the application of abstract mathematics in Coding Theory, through the study of error correcting codes and a sepcial class denoted BCH. The BCH code building uses algebraic structure of finite fields and are characterized by operations on polynomials. In order to understand the operation of the BCH codes, we will make a brief introduction to the study of rings, fields, congruence relations on the integers, polynomial rings, Galois fields, and vector spaces. Next, we'll be introduced the linear block codes and cyclic codes, and ultimately a special class of cyclic codes known as BCH.

SUMÁRIO

1 INTRODUÇÃO **1**

2 ESTRUTURA ALGÉBRICA DOS CÓDIGOS DE BLOCO **4**

2.1 Anéis 4

2.2 Classes Residuais 8

2.2.1 Classes Residuais dos Inteiros 10

2.3 Ideais de um anel 11

2.4 Corpos Finitos 12

2.5 Anéis de Polinômios 15

2.6 Construção do Corpo de Galois $GF(2^m)$ 19

2.6.1 Propriedades Básicas do Corpo de Galois $GF(2^m)$ 23

2.7 Espaços Vetoriais 26

2.7.1 Transformações Lineares 28

3 CÓDIGOS CORRETORES DE ERROS **30**

3.1 Códigos de Blocos Lineares 31

3.2 Distância Mínima de um Código de Bloco 39

3.3 Códigos Cíclicos 42

3.3.1 Codificação de Códigos Cíclicos 49

3.3.2 Cálculo da Síndrome e Detecção de Erro 52

3.3.3 Decodificação de Códigos Cíclicos 53

4 CÓDIGOS BCH **56**

4.1 Códigos BCH Binários 57

4.1.1 Códigos Cíclicos Definidos por Anulamento 58

4.2 Decodificação de Códigos BCH 63

4.2.1 Algoritmo iterativo para encontrar o polinômio localizador do erro $\sigma(X)$ . . . 67

4.2.2 Método de Chien para determinação dos números localizadores do erro 71

4.2.3 Desempenho dos Códigos BCH 72

5 CONSIDERAÇÕES FINAIS **77**

BIBLIOGRAFIA **79**

LISTA DE TABELAS

2.1 Adição em $\mathbb{Z}_2$. 10
2.2 Multiplicação em $\mathbb{Z}_2$. 10
2.3 Adição módulo-7. 15
2.4 Multiplicação módulo-7. 15
2.5 Adição em $\mathbb{F}_4 = K[X]_{P(X)}$ 18
2.6 Multiplicação em $\mathbb{F}_4 = K[X]_{P(X)}$ 18
2.7 Tabela logarítmica de Zech do corpo $GF(2^4)$. 23
2.8 Representação para elementos de $GF(2^4)$ gerado por $P(X) = 1 + X + X^4$. 25

3.1 Código de Hamming $(7, 4)$ 37

4.1 Passos do algoritmo de Berlekamp para um código BCH binário 68
4.2 Passos do algoritmo de Berlekamp para o código BCH binário do Exemplo 4.4. . . . 70

LISTA DE FIGURAS

3.1 Elementos de sistema de comunicação digital 30

4.1 Desempenho do código BCH(63,k,d) num canal AWGN 74
4.2 Desempenho do código BCH(31,k,d) num canal AWGN 75
4.3 Desempenho do código BCH(255,k,d) num canal AWGN 76

CAPÍTULO 1

INTRODUÇÃO

Desde a publicação da "Teoria Matemática da Comunicação"pelo engenheiro e matemático Claude Elwood Shannon, em 1948 [1], o mundo passou por uma transição tecnológica, também chamada de revolução digital, e vimos uma crescente mudança do uso de tecnologias analógicas de comunicação e armazenamento da informação para o uso de tecnologias digitais.

Nesse contexto surgiu a necessidade do desenvolvimento dos códigos corretores de erros, que são utilizados sempre que é preciso transmitir ou armazenar uma informação digitalizada de forma confiável, a fim de garantirmos que será possível recuperar a mensagem original.

De acordo com Haykin [2], a comunicação envolve implicitamente a informação transmitida de um ponto a outro por uma sucessão de processos, tais como: a geração de um sinal de mensagem; a descrição desse sinal de mensagem por meio de símbolos elétricos, auditivos ou visuais; a codificação desses símbolos em uma forma apropriada à transmissão por um meio físico; a transmissão desses símbolos até o destino; a decodificação; reprodução dos símbolos originais e recriação do sinal de mensagem original, com uma degradação da qualidade. Esta degradação da qualidade ocorre devido às várias imperfeições no sistema e, em grande parte, essas imperfeições ocorrem no meio de transmissão, que chamaremos de canal.

A codificação de canal, ou codificação para controle de erros, é a codificação dos sinais de informação com o objetivo de diminuir a taxa de erro de símbolo e/ou de bit durante a transmissão dos mesmos através de um canal de comunicação.

Os primeiros códigos corretores de erros foram propostos por Hamming em 1950 [3]. No entanto, os chamados códigos de Hamming são capazes de corrigir apenas um único erro e, portanto, tem pouca aplicação prática atualmente. Em 1957, Prange [4] propôs pela primeira vez os códigos cíclicos. De modo independente, Hocquenghem [5], em 1959, e Bose e Chaudhuri [6], em 1960, desenvolveram uma classe de códigos cíclicos capazes de corrigir erros múltiplos. Essa classe é chamada de códigos BCH, que é o objeto do nosso estudo neste trabalho. Ainda em 1960, foi descoberta uma subclasse dos códigos BCH não binários, por Reed e Solomon [7], denominada simplesmente por RS, com grande capacidade de correção de erros.

Atualmente, os códigos BCH são uma classe de códigos largamente utilizada em diversas aplicações práticas, como por exemplo, gravação de CD's e DVD's, TV digital e transmissão via satélite. Os códigos BCH apresentam diversas vantagens em relação a outros tipos de códigos por sua estrutura matemática bem definida, que permite facilidade de implementação em hardware. Por outro lado, os profissionais formados em Matemática, tanto na modalidade licenciatura quanto no bacharelado, são constantemente questionados sobre quais as aplicações práticas do que estudam. Para um matemático a Matemática não precisa se justificar, pois estudamos pelo prazer de desvendá-la, descobrir padrões, construir modelos, demonstrar teoremas e resolver problemas. Contudo, neste trabalho pretendemos mostrar a importância da Matemática para o desenvolvimento dos códigos corretores de erros.

Este trabalho está organizado da seguinte forma. No Capítulo 2 será feita uma breve introdução à Álgebra abstrata, introduzindo os conceitos de anéis, classes residuais, anéis de polinômios e corpos finitos, incluindo a contrução do corpo de Galois $GF(2^m)$, e destacando algumas propriedades deste

corpo que são relevantes no estudo dos codificadores de bloco. Além disso, introduziremos o conceito de espaço vetorial e de transformações lineares.

No Capítulo 3 será feita uma breve introdução à codificação para o controle de erros, enfatizando os códigos de bloco lineares. Inicialmente, descreveremos os elementos que compõem um sistema de comunicação digital, destacando o codificador de canal, que é utilizado para introduzir redundâncias na informação original a fim de tornar possível a detecção e correção de erros no receptor. Será analisada a estrutura dos códigos de bloco lineares, e introduziremos o conceito de distância mínima de um código de bloco, que é um parâmetro importante para determinar a capacidade de detecção e correção do código. Além disso, será destacada uma classe muito importante dos códigos de bloco, os códigos cíclicos.

No Capítulo 4 será apresentada uma subclasse importante de códigos cíclicos, os códigos BCH propostos em [5, 6], destacando-se os códigos BCH binários. Em seguida, será descrito o algoritmo de decodificação de Berlekamp, cujos detalhes podem ser consultados em [8]. Finalmente, no Capítulo 5 tem-se as principais conclusões deste trabalho.

CAPÍTULO 2

ESTRUTURA ALGÉBRICA DOS CÓDIGOS DE BLOCO

Algumas classes de códigos de bloco especiais, dentre eles o BCH, são construídos utilizando a estrutura algébrica de corpos finitos e são caracterizados por polinômios geradores. Assim, a fim de compreender a estrutura e funcionamento de um código de bloco BCH vamos introduzir algumas estruturas algébricas básicas e construir o corpo de Galois $GF(2^m)$.

2.1 Anéis

Definição 2.1. Um *anel* é um conjunto não-vazio A munido de duas operações binárias

$$\begin{array}{rcl} +: A \times A & \to & A \\ (a,b) & \mapsto & a+b \end{array} \qquad \text{e} \qquad \begin{array}{rcl} \cdot : A \times A & \to & A \\ (a,b) & \mapsto & a \cdot b \end{array}$$

chamadas, respectivamente, de adição e multiplicação tais que valem as seguintes propriedades:

1. $(a+b)+c = a+(b+c), \quad \forall a,b,c \in A$.
2. Existe um elemento neutro, chamado *zero*, denotado por 0 tal que $a+0=0+a=a$, $\forall a \in A$.
3. Existência de um elemento *inverso* para a adição: Dado $a \in A$, existe um elemento chamado

simétrico de a e denotado por $-a$, tal que $a + (-a) = -a + a = 0$.

4. $a + b = b + a, \quad \forall a, b \in A$.

5. $(a \cdot b) \cdot c = a \cdot (b \cdot c), \quad \forall a, b, c \in A$.

6. $a \cdot (b + c) = a \cdot b + a \cdot c$ e $(b + c) \cdot a = b \cdot a + c \cdot a, \quad \forall a, b, c \in A$.

Notação: $(A, +, \cdot)$ denotará um anel A com as operações $+$ e $\cdot$.

Seja $(A, +, \cdot)$ um anel, se existe um elemento denotado por $1 \in A$ tal que $a \cdot 1 = 1 \cdot a = a$, $\forall a \in A$ diremos que A é um *anel com unidade* 1.

Se um anel $(A, +, \cdot)$ satisfaz $a \cdot b = b \cdot a$, $\forall a, b \in A$ é dito *anel comutativo*.

A partir daqui consideraremos como anel A um *anel comutativo com unidade*.

Exemplo 2.1. $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ e $\mathbb{C}$ munidos com as operações de adição e multiplicação são exemplos de anéis. No entanto, o conjunto $\mathbb{N} = \{1, 2, 3, \ldots\}$ munido com operações de adição e multiplicação dos inteiros não forma um anel pois não existem simétricos dos elementos, nem elementro neutro para a adição.

Proposição 2.1. Seja A um anel temos que $a \cdot 0 = 0$, $\forall a \in A$.

Demonstração: Para $a \in A$ temos $a \cdot 0 = a \cdot (0 + 0) = a \cdot 0 + a \cdot 0$, somando-se $-(a \cdot 0)$ à igualdade $a \cdot 0 = a \cdot 0 + a \cdot 0$ temos:

$$\begin{aligned} -(a \cdot 0) + a \cdot 0 &= (-(a \cdot 0) + a \cdot 0) + a \cdot 0 \\ 0 &= 0 + a \cdot 0. \end{aligned}$$

Logo $a \cdot 0 = 0$. ∎

Proposição 2.2. Seja um conjunto não-vazio A munido com a operação $*$, com elemento neutro e. Temos que:

1. O elemento neutro e é único.

2. Se a operação $*$ é associativa e um elemento $a \in A$ possui um elemento inverso, esse inverso é único.

Demonstração:

(1.) Suponhamos, por absurdo, que existam dois elementos neutros e_1 e e_2. Então devemos ter:

$$e_2 = e_1 * e_2 = e_1$$

Daí, $e_1 = e_2$.

(2.) Dado $a \in A$ suponhamos, por absurdo, que existam b_1 e b_2 tais que $b_1 * a = a * b_1 = e$ e $a * b_2 = b_2 * a = e$, temos que $b_2 = b_2 * e = b_2 * (a * b_1) = (b_2 * a) * b_1 = e * b_1 = b_1$.

■

Um anel comutativo com unidade A será chamado de *domínio de integridade* se satisfaz a seguinte condição:

$$\forall a, b \in A, \quad a \neq 0 \ \ e \ \ b \neq 0 \Rightarrow a \cdot b \neq 0.$$

Exemplo 2.2. $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ e $\mathbb{C}$ munidos com as operações de adição e multiplicação usuais são domínios de integridade.

Definição 2.2. Um elemento $a \in A$, onde $(A, +, \cdot)$ é um anel com unidade, será dito *invertível* se existir um elemento $b \in A$ tal que $a \cdot b = 1$. E b é dito *inverso* de a.

Exemplo 2.3. Em $\mathbb{Z}$ os únicos elementos invertíveis são 1 e -1.

Definição 2.3. Um anel comutativo com unidade onde todo elemento não nulo é invertível é chamado de *corpo*.

Um corpo F que contém um corpo K, tal que as operações de F, quando restritas a K, coincidam com as operações de K, é chamado de *extensão* de K. Neste caso, dizemos que K é um *subcorpo*

de F. Denotaremos por F/K ou pelo diagrama

$$F$$

$$|$$

$$K$$

Neste caso, K é dito *corpo básico* ou corpo fundamental da extensão.

Exemplo 2.4. $\mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$, neste caso $\mathbb{C}$ é uma extensão de $\mathbb{R}$, $\mathbb{C}$ é uma extensão de $\mathbb{Q}$ e $\mathbb{R}$ é uma extensão de $\mathbb{Q}$.

Definição 2.4. Dado A um domínio de integridade, podemos definir o *corpo das frações* de A como sendo o conjunto:

$$Q(A) = \left\{ \frac{a}{b}; a, b \in A \text{ e } b \neq 0 \right\}$$

munido com as operações de adição e multiplicação:

$$\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd} \quad \text{e} \quad \frac{a}{b} \cdot \frac{c}{d} = \frac{a \cdot c}{b \cdot d}$$

respectivamente.

Vejamos a seguir alguns conceitos relacionados a divisibilidade dos elementos contidos em anéis.

Definição 2.5. Dados um anel A e dois elementos $a, b \in A$, diremos que a *divide* b, se existir um elemento $c \in A$, tal que $b = a \cdot c$, e denotaremos por $a|b$. Podemos dizer que a *é divisor de* b ou que b *é múltiplo de* a.

Definição 2.6. Sejam $a, b \in A$, dizemos que a é um *associado* de b se existir um elemento invertível $u \in A$ tal que $a = u \cdot b$.

Definição 2.7. Seja A um anel e $a \in A$ um elemento não invertível, diremos que a é *irredutível* se os únicos divisores de a são os seus associados e os elementos invertíveis de A.

Um elemento não invertível $a \in A$ será dito redutível quando não for irredutível.

Definição 2.8. Dado o anel A, e $a \in A\backslash\{0\}$ tal que a não é invertível, a é dito um *elemento primo* se

$$\forall\, b, c \in A, \quad a|b \cdot c \Rightarrow a|b \text{ ou } a|c.$$

Sejam A um domínio de integridade, $a, b \in A$ tais que $a \neq 0$ ou $b \neq 0$, e $d \in A$ será dito *máximo divisor comum (MDC)* de a e b se as seguintes condições forem satisfeitas:

1. $d|a$ e $d|b$.
2. $\forall c \in A$; $c|a$ e $c|b \Rightarrow c|d$.

Dois elementos num anel são *primos entre si* se os únicos divisores comuns desses elementos são invertíveis.

Sejam A um anel e $a, b \in A$, m é um *mínimo múltiplo comum* (MMC) de a e b se as seguintes condições forem satisfeitas:

1. $a|m$ e $b|m$.
2. $\forall c \in A$, se $a|c$ e $b|c$, então $m|c$.

2.2 Classes Residuais

Uma relação R entre pares de elementos de um conjunto A é dita *relação de equivalência*, se ela é reflexiva, simétrica e transitiva, ou seja, dados $a, b, c \in A$ devemos ter:

1. aRa. Reflexiva
2. Se aRb, então bRa. Simétrica
3. Se aRb e bRc, então aRc. Transitiva

Notação: $\sim$ denotará uma relação de equivalência.

Definição 2.9. Seja $\sim$ uma relação de equivalência em um conjunto A e $a \in A$. Definimos a *classe de equivalência* $\overline{a}$ do elemento a em relação a $\sim$ o conjunto $\overline{a} = \{b \in A; a \sim b\}$.

Sejam A um anel e $a, b, m \in A$, diremos que *a* é *congruente* a *b* módulo *m*, se $m|(b-a)$ e denotaremos por

$$a \equiv b \bmod m. \tag{2.1}$$

Proposição 2.3. Sejam $a, b, c, d \in A$ e $(A, +, \cdot)$ um anel. Então:

1. $a \equiv a \bmod m$.
2. Se $a \equiv b \bmod m$, então $b \equiv a \bmod m$.
3. Se $a \equiv b \bmod m$ e $b \equiv c \bmod m$, então $a \equiv c \bmod m$.
4. Se $a \equiv c \bmod m$ e $b \equiv d \bmod m$, então $a + b \equiv c + d \bmod m$ e $a \cdot b \equiv c \cdot d \bmod m$.

As propriedades $1, 2$, e 3 mostram que a relação de congruência é uma relação de equivalência e a propriedade 4 nos mostra que esta relação é compatível com as operações de adição e multiplicação do anel A.

Demonstração: Vamos demonstrar apenas a última propriedade, pois as demais são imediatas. Suponha que $a \equiv c \bmod m$ e $b \equiv d \bmod m$, assim temos $m|(a-c)$ e $m|(b-d)$

$$a \cdot b - c \cdot d = a \cdot (b-d) + d \cdot (a-c).$$

Como $(b-d)$ e $(a-c)$ são múltiplos de m, $\exists \lambda, \mu \in A$ tais que $(a-c) = m \cdot \lambda$ e $(b-d) = m \cdot \mu$, então:

$$\begin{aligned} a \cdot (b-d) + d \cdot (a-c) &= a \cdot m \cdot \lambda + d \cdot m \cdot \mu = m \cdot (a\lambda + d\mu) \Rightarrow \\ a \cdot b - c \cdot d &= m \cdot (a\lambda + d\mu). \end{aligned}$$

■

A *classe residual* de um elemento $a \in A$, módulo *m*, é o conjunto

$$\overline{a} = \{b \in A; b \equiv a \bmod m\} = \{a + m\lambda; \lambda \in A\}.$$

Podemos definir $m \cdot A = \{m \cdot \lambda; \lambda \in A\}$ e assim temos:

$$\overline{a} = a + mA.$$

onde a é chamado de *representante* da classe residual $\overline{a}$.

Vamos definir A_m como sendo o conjunto de todas as classes residuais em A módulo m.

O conjunto A_m munido das operações $\overline{a} + \overline{b} = \overline{a+b}(\text{ mod } m)$ e $\overline{a} \cdot \overline{b} = \overline{a \cdot b}(\text{ mod } m)$ é um anel, com $\overline{0}$ e $\overline{1}$, respectivamente, os elementos neutros para a adição e multiplicação.

Dentre as classes residuais, estamos interessados nas classes residuais dos inteiros que definiremos a seguir.

2.2.1 Classes Residuais dos Inteiros

Seja $m \in \mathbb{N} - \{0\}$, então podemos definir o conjunto

$$\mathbb{Z}_m = \{\overline{0}, \overline{1}, \ldots, \overline{(m-1)}\}$$

em que, se $i, j = 0, 1, \ldots, m-1$ com $i \neq j$, então $\overline{i} \neq \overline{j}$.

Exemplo 2.5. $\mathbb{Z}_2 = \{\overline{0}, \overline{1}\}$ com as operações definidas abaixo é um anel

Tabela 2.1: Adição em $\mathbb{Z}_2$.

+	$\overline{0}$	$\overline{1}$
$\overline{0}$	$\overline{0}$	$\overline{1}$
$\overline{1}$	$\overline{1}$	$\overline{0}$

Tabela 2.2: Multiplicação em $\mathbb{Z}_2$.

$\cdot$	$\overline{0}$	$\overline{1}$
$\overline{0}$	$\overline{0}$	$\overline{0}$
$\overline{1}$	$\overline{0}$	$\overline{1}$

Além disso, como $\overline{1}$ é o único elemento não nulo de $\mathbb{Z}_2$ e é invertível, temos que $\mathbb{Z}_2$ é um corpo, também chamado do corpo binário.

Proposição 2.4. Seja $\overline{a} \in \mathbb{Z}_m$, $\overline{a}$ é invertível $\Leftrightarrow MDC(a, m) = 1$.

Demonstração: Suponha $\overline{a}$ invertível $\Rightarrow \exists b \in \mathbb{Z}$ tal que $\overline{a} \cdot \overline{b} = 1 \Rightarrow \overline{a \cdot b} = 1 \Rightarrow a \cdot b \equiv 1 \bmod m \Rightarrow$ $\exists \lambda \in \mathbb{Z}$ tal que $1 = a \cdot b + \lambda \cdot m$. Como $MDC(a, m)|a$ e $MDC(a, m)|m$ temos que existem s e

$t \in \mathbb{Z}_m$ tais que $a = MDC(a,m)\cdot t$ e $m = MDC(a,m)\cdot s \Rightarrow 1 = MDC(a,m)\cdot[t\cdot b + \lambda \cdot s] \Rightarrow$ $MDC(a,m)|1 \Rightarrow MDC(a,m) = 1$.

Agora suponha $MDC(a,m) = 1 \Rightarrow \exists b, c \in \mathbb{Z}_m$ tais que $b\cdot a + c\cdot m = 1 \Rightarrow b\cdot a \equiv 1 \bmod m \Rightarrow$ $\overline{b}\cdot\overline{a} = \overline{a\cdot b} = \overline{1} \Rightarrow \overline{a}$ é invertível. ■

Proposição 2.5. O anel $(\mathbb{Z}_m, +, \cdot)$ é um corpo se, e somente se, m é um número primo.

Demonstração: $\mathbb{Z}_m$ é um corpo $\Leftrightarrow$ todos os elementos de $\mathbb{Z}_m$ são invertíveis, mas pela Proposição 2.4, isto é o mesmo que $MDC(1,m) = MDC(2,m) = \ldots = MDC(m-1,m) = 1$, isto significa que m é primo. ■

2.3 Ideais de um anel

Definição 2.10. Um subconjunto I não vazio de um anel A é um ideal de A se:

i. $\forall a, b \in I, a + b \in I$;

ii. $\forall a \in I$ e $\forall c \in A$, $ca \in I$ e $ac \in I$.

Note que um ideal I sempre contém o elemento zero de A, pois dado um elemento qualquer não nulo $a \in I$, temos $0 = 0a \in I$.

Também é possível observar que $I = 0$ e $I = A$ são ideais de A.

Definição 2.11. Seja A um anel comutativo com unidade e $a \in A$, então o conjunto $I(a) = \{ca; c \in A\}$ é um ideal de A, chamado de *ideal principal* gerado por a.

Em geral, se $a_1, \cdots, a_n \in A$, então o conjunto

$$I = I(a_1, \cdots, a_n) = \{c_1a_1 + \cdots + c_na_n; c_1, \cdots, c_n \in A\}$$

é um ideal de A. Os elementos $a_1, \cdots, a_n$ são chamados de geradores de I.

2.4 Corpos Finitos

De acordo com Hefez e Villela [9], grande parte da teoria de códigos baseia-se na álgebra linear sobre corpos finitos. Portanto, vejamos algumas propriedades importantes desses corpos a fim de compreender a estrutura dos códigos corretores de erros.

Definição 2.12. Seja K um corpo finito com elemento unidade 1. Considere o conjunto

$$\Lambda_K = \{n \in \mathbb{N};\ n \cdot 1 = 0\} \subset \mathbb{N}.$$

A *característica* do corpo finito K é definida como o inteiro positivo λ

$$\lambda = \min \Lambda_K = \min\{n \in \mathbb{N};\ n \cdot 1 = 0\}. \tag{2.2}$$

Proposição 2.6. Seja K um corpo finito, então λ é um número primo.

Demonstração: Suponha que λ não seja primo, então existem $\lambda_1,\ \lambda_2 \in \mathbb{Z}$ tais que $1 < \lambda_1 < \lambda$ e $1 < \lambda_2 < \lambda$, com $\lambda = \lambda_1\lambda_2$. Logo,

$$0 = \lambda \cdot 1 = (\lambda_1 \cdot \lambda_2) \cdot 1 = \lambda_1(\lambda_2 \cdot 1) = (\lambda_1 \cdot 1)(\lambda_2 \cdot 1).$$

Como todo corpo é um domínio de integridade, então

$$(\lambda_1 \cdot 1)(\lambda_2 \cdot 1) = 0 \Leftrightarrow \lambda_1 \cdot 1 = 0 \ \text{ ou } \ \lambda_2 \cdot 1 = 0.$$

Mas isso é uma contradição, pois, λ é o menor inteiro positivo tal que $\lambda \cdot 1 = 0$.

■

Definição 2.13. Seja $\alpha \in K^*$, onde $K^* = K \setminus \{0\}$ e um corpo finito, define-se a *ordem* do elemento α como sendo o menor inteiro n tal que $\alpha^n = 1$.

Definição 2.14. Seja G um conjunto não-vazio munido da operação binária $* : G \times G \to G$. Dizemos que o par $(G, *)$ é um *grupo* se são válidas as seguintes propriedades:

a. $a * (b * c) = (a * b) * c, \quad \forall a, b, c \in G.$

b. $\exists e \in G$ tal que $a * e = e * a = a, \quad \forall a \in G.$

c. $\forall a \in G, \exists b \in G$ tal que $a * b = b * a = e$ e este elemento b é dito *inverso* de a.

Os corpos finitos também são chamados de *corpos de Galois* [1]. No Exemplo 2.5 vimos que o corpo $\mathbb{Z}_2$ é chamado *corpo binário*, que é um corpo de Galois, e podemos denotá-lo por $GF(q)$ para $q = 2$, teremos $GF(2)$.

Definição 2.15. Sejam A e B dois anéis (ou corpos). Uma função $f : A \to B$ será chamada *homomorfismo* se, para todos os elementos $a, b \in A$, valem as seguintes condições:

(i) $f(a + b) = f(a) + f(b)$.

(ii) $f(a \cdot b) = f(a) \cdot f(b)$.

(iii) $f(1) = 1$.

Definição 2.16. Um homomorfismo bijetor de corpos será chamado de *isomorfismo*. Dois corpos serão ditos *isomorfos* se existir um isomorfismo entre eles.

Teorema 2.1. Seja K um corpo finito com característica $\lambda = p$, onde p é um número primo. Então, K contém um subcorpo isomorfo a $\mathbb{Z}_p$ (que ainda denotaremos por $\mathbb{Z}_p$). Em particular, K tem p^n elementos para algum número natural n.

Teorema 2.2. Seja α um elemento não-nulo do corpo finito $GF(q)$. Então $\alpha^{q-1} = 1$.

Demonstração: Sejam $b_1, b_2, \ldots, b_{q-1}$ os $q-1$ elementos não-nulos de $GF(q)$. Como $GF(q)$ é um corpo, sabemos que a operação de multiplicação é fechada, pois a multiplicação de dois elementos de

[1] Evariste Galois nasceu em 25 de outubro de 1811 na cidade de Bourg-la-Reine, na França. Filho do republicano Nicolas-Gabriel Galois e de Adelaide Marie Demante, que foi responsável por sua educação até os 12 anos de idade. Em 1823 iniciou sua educação formal no Liceu de Louis-le-Grand, em Paris. Em 1829 publicou seu primeiro artigo sobre a solução de equações algébricas. Ficou conhecido pelo seu estudo da insolubilidade das equações de graus superiores a quatro ao criar um objeto, conhecido hoje como "Grupo de Galois", que possibilita investigar se um polinômio qualquer pode ter suas soluções encontradas por meio de radicais ou não. Galois faleceu no dia 31 de maio de 1832, aos 20 anos de idade, após um duelo.

$GF(q)$ resulta em um elemento do próprio $GF(q)$. Então, os $q-1$ elementos $\alpha \cdot b_1, \alpha \cdot b_2, \ldots, \alpha \cdot b_{q-1}$ são não-nulos e temos:

$$(\alpha \cdot b_1)(\alpha \cdot b_2) \cdot \ldots \cdot (\alpha \cdot b_{q-1}) = b_1 \cdot b_2 \cdot \ldots \cdot b_{q-1}$$

$$\alpha^{q-1}(b_1 \cdot b_2 \cdot \ldots \cdot b_{q-1}) = b_1 \cdot b_2 \cdot \ldots \cdot b_{q-1}.$$

Como por hipótese $\alpha \neq 0$ e $b_1 \cdot b_2 \cdot \ldots \cdot b_{q-1} \neq 0$ devemos ter:

$$\alpha^{q-1} = 1.$$

■

Teorema 2.3. Seja α um elemento não-nulo do corpo finito $GF(q)$. Seja n a ordem de α. Então n divide $q-1$.

Demonstração: Suponha que n não divide $q-1$. Dividindo $q-1$ por n, obtemos $q-1 = k \cdot n + r$, onde $0 < r < n$. Então $\alpha^{q-1} = \alpha^{kn+r} = \alpha^{kn} \cdot \alpha^r = (\alpha^n)^k \cdot \alpha^r = 1^k \cdot \alpha^r = \alpha^r$. Portanto, $\alpha^{q-1} = 1$ se $\alpha^r = 1$, mas isto é impossível pois $0 < r < n$, e n é o menor inteiro tal que $\alpha^n = 1$. Logo, n divide $q-1$.

■

Definição 2.17. Seja o corpo finito $GF(q)$, um elemento não-nulo α de $GF(q)$ é dito *primitivo* se a ordem de α for $q-1$, ou seja, se $GF(q)^* = \{1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\}$.

Portanto, as potências do elemento primitivo geram todos os elementos não-nulos de $GF(q)$.

Teorema 2.4. Todo corpo finito possui elementos primitivos.

Uma demonstração deste teorema pode ser consultada em [9]. Contudo, vejamos a seguir um exemplo que ilustra este fato.

Exemplo 2.6. Considere o corpo finito $GF(7)$ com as operações definidas nas tabelas abaixo

Tabela 2.3: Adição módulo-7.

$+$	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

Tabela 2.4: Multiplicação módulo-7.

$\cdot$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

Se tomarmos as potências do elemento 3 usando a tabela da multiplicação obtemos:

$3^1 = 3.$

$3^2 = 3 \cdot 3 = 2 \bmod 7.$

$3^3 = 3 \cdot 3^2 = 6 \bmod 7.$

$3^4 = 3 \cdot 3^3 = 4 \bmod 7.$

$3^5 = 3 \cdot 3^4 = 5 \bmod 7.$

$3^6 = 3 \cdot 3^5 = 1 \bmod 7.$

Então a ordem do elemento 3 é 6, e como $q = 7$ temos $n = q - 1 \Rightarrow 6 = 7 - 1$. Portanto, 3 é um elemento primitivo do $GF(7)$.

2.5 Anéis de Polinômios

Definição 2.18. Sejam A um anel e X uma indeterminada. Definimos o *polinômio* $P(X)$ com coeficientes em A na indeterminada X como

$$P(X) = \sum_{i=0}^{n} a_i X^i = a_0 + a_1 X + \ldots + a_n X^n \tag{2.3}$$

em que $n \in \mathbb{Z}^+$ e $a_i \in A, \forall i = 0, 1, \ldots, n$.

Dados os polinômios $P(X) = a_0 + a_1X + \ldots + a_nX^n$ e $Q(X) = b_0 + b_1X + \ldots + b_mX^m$, podemos dizer que $P(X) = Q(X)$ se $a_i = b_i$, para todo i. Seja $A[X] = \{P(X);\ a_i \in A; \forall i = 1, \ldots, n\}$, ou seja, $A[X]$ é o conjunto de todos os polinômios na indeterminada X com coeficientes em A. Note que $A \subset A[X]$. Considere $P(X)$ e $Q(X)$ definidos acima, podemos definir as seguintes operações:

$$P(X) + Q(X) = \sum_{i=0}^{\max\{n,m\}} (a_i + b_i)X^i \tag{2.4}$$

$$P(X) \cdot Q(X) = \sum_{i=0}^{n+m} c_iX^i \tag{2.5}$$

onde $c_i = \sum_{j=0}^{i} a_j \cdot b_{i-j}X^i$.

O conjunto $A[X]$ munido das operações definidas acima é um anel.

Se $P(X) = 0 + 0X + 0X^2 + \ldots + 0X^n$ indicaremos $P(X) \equiv 0$ e dizemos que o polinômio é *identicamente nulo* sobre A. Se $P(X) = a$, com $a \in A$, então $P(X)$ é o polinômio *constante* a. Se $P(X) \in A[X]$ tal que $a_n \neq 0$ e $a_j = 0\ \forall j > n$ dizemos que n é o grau de $P(X)$ e denotaremos por $\partial P(X) = n$.

Note que não está definido o grau do polinômio identicamente nulo e que ∂ pode ser interpretada como uma função de $A[X]$ em $\mathbb{N}$ da seguinte forma

$$\begin{aligned} \partial\colon\ A[X] - \{0\} &\to \mathbb{N} \\ P(X) &\mapsto \partial P(X) \end{aligned}$$

Seja K um corpo e $K[X]$ o domínio dos polinômios sobre K na indeterminada X. Sejam $F(X), G(X) \in K[X]$. Suponha que o grau de $G[X]$ é diferente de zero, efetuando a divisão de $F(X)$ por $Q(X)$ obtemos um único par de polinômios sobre $Q(X), R(X) \in K[X]$, chamados de *quociente* e *resto*, respectivamente, tais que:

$$F(X) = Q(X) \cdot G(X) + R(X) \tag{2.6}$$

onde ou $R(X) = 0$ ou $\partial R(X) < \partial G(X)$.

Esta expressão é conhecida como Algoritmo de Euclides da Divisão.

Exemplo 2.7. Vamos dividir $f(X) = 1 + X + X^4 + X^5 + X^6$ por $g(X) = 1 + X + X^3$, onde $f(X)$ e $g(X)$ são polinômios sobre $GF(2)$.

$$\begin{array}{l|l} X^6 + X^5 + X^4 + \quad +X + 1 & \underline{X^3 + X + 1} \\ \underline{+X^6 \quad + X^4 + X^3 \qquad\qquad} & X^3 + X^2 \\ \quad X^5 + \quad +X^3 + \quad +X + 1 & \\ \underline{\quad X^5 + \quad +X^3 + X^2 \qquad} & \\ \qquad\qquad\qquad X^2 + X + 1 & \end{array}$$

$\Rightarrow f(X) = g(X)(X^3 + X^2) + (X^2 + X + 1)$.

Definição 2.19. Seja K um corpo e $P(X) \in K[X]$ tal que $\partial P(X) \geq 1$. Dizemos que $P(X)$ é um polinômio *irredutível* sobre K se, toda vez que $P(X) = G(X)H(X)$, tal que $G(X), H(X) \in K[X]$, então temos $G(X) = a$ constante em K ou $H(X) = b$ constante em K. Se $P(X)$ não for irredutível sobre K, dizemos que $P(X)$ é *redutível* sobre K.

Definição 2.20. Seja K um corpo e $K[X]$ o anel dos polinômios sobre K. Dado $P(X) \in K[X]$. Dizemos que $P(X)$ é *mônico* quando o coeficiente do termo de mais alto grau for $1 \in K$.

Podemos definir as classes residuais de $A = K[X]$ módulo um polinômio não constante e mônico $m = P(X)$ de grau n da seguinte forma:

$$A_m = K[X]_{P(X)} = \{[R(X)]; R(X) \in K[X] \text{ com } R(X) = 0 \text{ ou } \partial R(X) < n\}. \qquad (2.7)$$

Dados $R_1(X), R_2(X) \in K[X]$, com $\partial R_1(X) < n$ e $\partial R_2(X) < n$, tais que $R_1(X) \neq R_2(X)$, então $[R_1(X)] \neq [R_2(X)]$.

Teorema 2.5. O anel $K[X]_{P(X)}$ é um corpo se, e somente se, o polinômio $P(X)$ é irredutível.

Este teorema nos fornece um método prático pra construir corpos finitos, como poderemos observar no exemplo a seguir.

Exemplo 2.8. Sejam $K = \mathbb{Z}_2$ e $P(X) = X^2 + X + 1$. $P(X)$ é um polinômio irredutível de grau 2 em $K[X]$. Podemos construir um corpo $\mathbb{F}_4$ dado por: $\mathbb{F}_4 = K[X]_{P(X)} = \{[0], [1], [X], [1 + X]\}$ com as seguintes operações:

Tabela 2.5: Adição em $\mathbb{F}_4 = K[X]_{P(X)}$

$+$	$[0]$	$[1]$	$[X]$	$[1+X]$
$[0]$	$[0]$	$[0]$	$[0]$	$[0]$
$[1]$	$[0]$	$[1]$	$[X]$	$[1+X]$
$[X]$	$[0]$	$[X]$	$[1+X]$	$[1]$
$[1+X]$	$[0]$	$[1+X]$	$[1]$	$[X]$

Tabela 2.6: Multiplicação em $\mathbb{F}_4 = K[X]_{P(X)}$

$\cdot$	$[0]$	$[1]$	$[X]$	$[1+X]$
$[0]$	$[0]$	$[1]$	$[X]$	$[1+X]$
$[1]$	$[1]$	$[0]$	$[1+X]$	$[X]$
$[X]$	$[X]$	$[1+X]$	$[0]$	$[1]$
$[1+X]$	$[1+X]$	$[X]$	$[1]$	$[0]$

A partir do Teorema 2.5 temos um método para construir corpos finitos. Dados $K = \mathbb{Z}_p$, onde p é um número primo positivo e $P(X) \in K[X]$ um polinômio irredutível com $\partial P(X) < n$, então $K[X]_{P(X)}$ é formado pelas classes de polinômios em $K[X]$ e o corpo $K[X]_{P(X)}$ tem p^n elementos.

Proposição 2.7. Qualquer polinômio irredutível sobre $GF(2)$ de grau m divide $X^{2^m-1} + 1$.

Definição 2.21. Um polinômio irredutível $P(X)$ de grau m é dito *primitivo* se o menor inteiro positivo n para o qual $P(X)$ divide $X^n + 1$ é $n = 2^m - 1$.

Exemplo 2.9. O polinômio $P(X) = X^4 + X + 1$ divide $X^{15} + 1$ mas não divide nenhum polinômio $X^n + 1$ para $1 \leq n < 15$. Portanto, $P(X)$ é um polinômio primitivo.

É possível construir códigos corretores de erros com símbolos de qualquer corpo de Galois $GF(p)$, onde p é um número primo ou uma potência de algum número primo. No entanto, por razões de ordem prática, nos sistemas de transmissão digital ou sistemas de armazenamento, são utilizados códigos com símbolos no corpo binário $GF(2)$ e nas suas extensões $GF(2^m)$, onde $m \in \mathbb{Z}, m > 1$[10].

Portanto, na seção seguinte iremos mostrar um método pra construir um corpo de Galois com 2^m elementos $(m > 1)$ em $GF(2)$.

2.6 Construção do Corpo de Galois $GF(2^m)$

Considere os elementos $0, 1 \in GF(2)$ e um novo símbolo α. Então, definimos a multiplicação " $\cdot$ " da seguinte forma:

$$\begin{aligned} 0 \cdot \alpha^j &= \alpha^j \cdot 0 = 0 \\ 1 \cdot \alpha^j &= \alpha^j \cdot 1 = \alpha^j \\ \alpha^i \cdot \alpha^j &= \alpha^j \cdot \alpha^i = \alpha^{i+j} \end{aligned}$$

onde $i, j = 0, 1, 2, \ldots$.

Formamos assim o conjunto $F = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^j, \ldots\}$ onde a operação multiplicação " $\cdot$ " está definida. Note que o elemento 1 pode ser denotado por α^0.

Precisamos impor uma condição sobre o elemento α para que o conjunto F tenha apenas 2^m elementos e seja fechado sob a operação de multiplicação definida.

Seja $P(X)$ um polinômio primitivo de grau m sobre $GF(2)$. Assuma que $P(\alpha) = 0$, isto é, α é uma raiz de $P(X)$. Temos, por definição que $P(X)$ divide $X^{2^m-1} + 1$ temos: $X^{2^m-1} + 1 = Q(X)P(X)$. Substituindo X por α temos:

$$\begin{aligned} \alpha^{2^m-1} + 1 &= Q(\alpha)P(\alpha) \\ \alpha^{2^m-1} + 1 &= Q(\alpha) \cdot 0 \\ \alpha^{2^m-1} + 1 &= 0. \end{aligned}$$

Adicionando 1 de ambos os lados desta equação e usando a adição módulo-2, obtemos:

$$\alpha^{2^m-1} = 1.$$

Portanto, α é um elemento primitivo, e obtemos o conjunto F^* finito e que contém os seguintes elementos:

$$F^* = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^m-2}\}.$$

É fácil mostrar que F^* é fechado sob a multiplicação " $\cdot$ " definida para F. E podemos definir a adição " $+$ " em F^*, mas para isto precisamos definir a expressão polinomial para os elementos de F^*.

O elemento $0 \in F^*$ será representado pelo *polinômio identicamente nulo* e os elementos diferentes de zero $\alpha^0, \alpha^1, \ldots, \alpha^{2^m-2} \in F^*$ serão representados por $2^m - 1$ polinômios diferentes de zero de α sobre $GF(2)$ com grau menor ou igual a $m - 1$:

$$\alpha^i = a_{i0} + a_{i1}\alpha + a_{i2}\alpha^2 + \ldots + a_{i,m-1}\alpha^{m-1} \tag{2.8}$$

onde $0 \leq i < 2^m - 1$.

Nesta representação a operação adição pode ser definida usando-se certas tabelas chamadas de tabelas logarítmicas de Zech que podemos construir da seguinte maneira:

Se $i \leq j$, temos que $\alpha^i + \alpha^j = \alpha^i(1 + \alpha^{j-i})$. Então, se soubermos para cada r determinar o inteiro $Z(r)$ tal que $1 + \alpha^r = \alpha^{Z(r)}$, obtemos:

$$\alpha^i + \alpha^j = \alpha^i \cdot \alpha^{z(j-i)}. \tag{2.9}$$

Assim, formamos as tabelas de Zech com os valores de $Z(r)$ para $1 \leq r \leq q - 2$. E para efetuar a operação de adição em F, escolhemos um elemento primitivo de F^* e utilizamos as tabelas de Zech.

Temos que F^* é fechado sob a adição " $+$ " definida deste modo e F^* é um grupo comutativo com a adição "$+$". Além disso, os elementos diferentes de zero de F^* formam um grupo comutativo

com a operação de multiplicação " $\cdot$ " que definimos anteriormente.

Usando a operação polinomial para os elementos de F^* e como os polinômios satisfazem a lei da distributividade então a multiplicação em F^* é distributiva sob a adição em F^*. Portanto, o conjunto F^* munido com as operações de adição e multiplicação definidas acima é um corpo de Galois com 2^m elementos, $GF(2^m)$.

Seja $a_0 + a_1\alpha + a_2\alpha^2 + \ldots + a_{m-1}\alpha^{m-1}$ a representação polinomial de um elemento β do corpo $GF(2^m)$. Para o estudo de codificadores existe uma representação mais apropriada dos elementos β. Vamos representar o elemento β por uma sequência ordenada de m componentes chamada m-*upla*, como segue:

$$(a_0, a_1, a_2, \ldots, a_{m-1}) \tag{2.10}$$

onde os m componentes são exatamente os m coeficientes da representação polinomial. Note que $a_i \in GF(2), 0 \leq i < m-1$.

Exemplo 2.10. Seja $GF(2^4) = GF(2)[X]_{[X^4+X+1]}$ e considere o elemento primitivo $\alpha = [X] \in GF(2^4)$. Vamos determinar a tabela de Zech de $GF(2^4)$. Note que $X^4 + X + 1$ é irredutível sobre $GF(2)$. Se α é um elemento primitivo, então $\alpha^{15} = 1$. Sabemos que $\alpha^i = a_{i0} + a_{i1}\alpha + a_{i2}\alpha^2 + \cdots + a_{i,m-1}\alpha^{m-1}$ para $0 \leq i < m-1$, então para $m = 4$ temos:

$$\alpha^i = a_{i0} + a_{i1}\alpha + a_{i2}\alpha^2 + a_{i3}\alpha^3$$

onde $a_{i0}, a_{i1}, a_{i2}, a_{i3} \in GF(2)$. Portanto, cada elemento do corpo $GF(2^4)$ pode ser escrito na base $\{1, \alpha, \alpha^2, \alpha^3\}$ e sabemos que α é raiz do polinômio $X^4 + X + 1 \Rightarrow P(\alpha) = \alpha^4 + \alpha + 1 = 0 \Rightarrow \alpha^4 = \alpha + 1$. Usando estes fatos obtemos as seguintes identidades:

$$\alpha^4 = \alpha + 1 \tag{2.11}$$

$$\alpha^5 = \alpha^4 \cdot \alpha = (\alpha + 1) \cdot \alpha = \alpha^2 + \alpha \tag{2.12}$$

$$\alpha^6 = \alpha^5 \cdot \alpha = (\alpha^2 + \alpha) \cdot \alpha = \alpha^3 + \alpha^2 \tag{2.13}$$

$$\alpha^7 = \alpha^6 \cdot \alpha = (\alpha^3 + \alpha^2) \cdot \alpha = \alpha^4 + \alpha^3 = \alpha^3 + \alpha + 1 \tag{2.14}$$

$$\alpha^8 = \alpha^7 \cdot \alpha = (\alpha^3 + \alpha + 1) \cdot \alpha = \alpha^4 + \alpha^2 + \alpha = \alpha^2 + 1 \tag{2.15}$$

$$\alpha^9 = \alpha^8 \cdot \alpha = (\alpha^2 + 1) \cdot \alpha = \alpha^3 + \alpha \tag{2.16}$$

$$\alpha^{10} = \alpha^9 \cdot \alpha = (\alpha^3 + \alpha) \cdot \alpha = \alpha^4 + \alpha^2 = \alpha^2 + \alpha + 1 \tag{2.17}$$

$$\alpha^{11} = \alpha^{10} \cdot \alpha = (\alpha^2 + \alpha + 1) \cdot \alpha = \alpha^3 + \alpha^2 + \alpha \tag{2.18}$$

$$\alpha^{12} = \alpha^{11} \cdot \alpha = (\alpha^3 + \alpha^2 + \alpha) \cdot \alpha = \alpha^4 + \alpha^3 + \alpha^2 = \alpha^3 + \alpha^2 + \alpha + 1 \tag{2.19}$$

$$\alpha^{13} = \alpha^{12} \cdot \alpha = \alpha^4 + \alpha^3 + \alpha^2 + \alpha = \alpha^3 + \alpha^2 + 1 \tag{2.20}$$

$$\alpha^{14} = \alpha^{13} \cdot \alpha = \alpha^4 + \alpha^3 + \alpha = \alpha^3 + 1 \tag{2.21}$$

$$\alpha^{15} = \alpha^{14} \cdot \alpha = \alpha^4 + \alpha = 1 \tag{2.22}$$

Dessas identidades obtemos a tabela de Zech, pois de (2.11), tem-se $Z(1) = 4$ e $Z(4) = 1$. De (2.15), obtemos $Z(2) = 8$ e $Z(8) = 2$. De (2.21), obtemos $Z(3) = 14$ e $Z(14) = 3$.

Somando 1 a ambos os lados das igualdades $(2.12), (2.13), (2.14)$ e (2.18) e comparando com $(2.17), (2.20), (2.16)$ e (2.19) respectivamente, obtemos

$$\alpha^5 + 1 = \alpha^2 + \alpha + 1 = \alpha^{10} \Rightarrow Z(5) = 10 \text{ e } Z(10) = 5.$$

$$\alpha^6 + 1 = \alpha^3 + \alpha^2 + 1 = \alpha^{13} \Rightarrow Z(6) = 13 \text{ e } Z(13) = 6.$$

$$\alpha^7 + 1 = \alpha^3 + \alpha = \alpha^9 \Rightarrow Z(7) = 9 \text{ e } Z(9) = 7.$$

$$\alpha^{11} + 1 = \alpha^3 + \alpha^2 + \alpha + 1 = \alpha^{12} \Rightarrow Z(11) = 12 \text{ e } Z(12) = 11.$$

Assim formamos a Tabela 2.7. Agora, podemos efetuar a adição de quaisquer dois elementos do corpo $GF(2^4)$ como, por exemplo,

$$\alpha^5 + \alpha^9 = \alpha^5 \cdot \alpha^{Z(4)} = \alpha^5 \cdot \alpha^1 = \alpha^6.$$

Tabela 2.7: Tabela logarítmica de Zech do corpo $GF(2^4)$.

r	Z(r)
1	4
2	8
3	14
4	1
5	10
6	13
7	9
8	2
9	7
10	5
11	12
12	11
13	6
14	3

Na álgebra comum sabemos que um polinômio com coeficientes reais pode não ter raízes reais e ter raízes complexas, e vimos pelo Exemplo 2.4 que o corpo dos números complexos é uma extensão do corpo dos números reais. Do mesmo modo, podemos ter um polinômio sobre $GF(2)$ que não possui raízes em $GF(2)$ mas terá raízes em alguma extensão $GF(2^m)$. Vejamos a seguir alguns resultados envolvendo essas raízes.

2.6.1 Propriedades Básicas do Corpo de Galois $GF(2^m)$

Proposição 2.8. Os $2^m - 1$ elementos diferentes de zero de $GF(2^m)$ formam todas as raízes de $X^{2^m-1} + 1$.

Definição 2.22. Seja $\beta \in GF(2^m)$, definimos o *polinômio minimal* de β como sendo o polinômio $\phi(X)$ de menor grau sobre $GF(2)$ tal que $\phi(\beta) = 0$. Este polinômio $\phi(X)$ é irredutível.

Exemplo 2.11. O polinômio minimal de $0 \in GF(2^m)$ é $\phi(X) = X$, pois $\phi(\beta) = 0$.

Teorema 2.6. Sejam $F(X)$ um polinômio sobre $GF(2)$ e $\phi(X)$ o polinômio minimal do elemento β de um corpo. Se β é uma raiz de $F(X)$, então $F(X)$ é divisível por $\phi(X)$.

Demonstração: Dividindo $F(X)$ por $\phi(X)$, obtemos:

$$F(X) = Q(X)\phi(X) + R(X)$$

onde $\partial R(X) < \partial\phi(X)$ ou $R(x) \equiv 0$.Substituindo β em $F(X)$ e sabendo que $F(\beta) = \phi(\beta) = 0$ temos:

$$F(\beta) = Q(\beta)\phi(\beta) + R(\beta) \Rightarrow 0 = Q(\beta) \cdot 0 + R(\beta) \Rightarrow R(\beta) = 0.$$

Se $R(X) \neq 0, R(X)$ seria um polinômio de menor grau que $\phi(X)$ que tem como raiz β. No entanto, isto é uma contradição pois $\phi(X)$ é o polinômio minimal de β. Portanto, $R(X) \equiv 0$ e $\phi(X)$ divide $F(X)$.

■

Definição 2.23. Sejam $F(X)$ um polinômio sobre $GF(2)$ e β um elemento pertencente a uma extensão do corpo $GF(2)$. Se β é uma raiz de $F(X)$, então para algum $l \geq 0, \beta^{2^l}$ também é raiz de $F(X)$. Este elemento β^{2^l} é chamado de *conjugado* de β.

Teorema 2.7. Seja $\phi(X)$ o polinômio minimal de um elemento $\beta \in GF(2^m)$. Seja e o menor inteiro tal que $\beta^{2^e} = \beta$. Então,

$$\phi(X) = \prod_{i=0}^{e-1}(X + \beta^{2^i}). \tag{2.23}$$

Tabela 2.8: Representação para elementos de $GF(2^4)$ gerado por $P(X) = 1 + X + X^4$.

Representação em potências	Representação polinomial	Representação 4-upla
0	0	(0 0 0 0)
1	1	(1 0 0 0)
α	α	(0 1 0 0)
α^2	α^2	(0 0 1 0)
α^3	α^3	(0 0 0 1)
α^4	$1 + \alpha$	(1 1 0 0)
α^5	$\alpha + \alpha^2$	(0 1 1 0)
α^6	$\alpha^2 + \alpha^3$	(0 0 1 1)
α^7	$1 + \alpha + \alpha^3$	(1 1 0 1)
α^8	$1 + \alpha^2$	(1 0 1 0)
α^9	$\alpha + \alpha^3$	(0 1 0 1)
α^{10}	$1 + \alpha + \alpha^2$	(1 1 1 0)
α^{11}	$\alpha + \alpha^2 + \alpha^3$	(0 1 1 1)
α^{12}	$1 + \alpha + \alpha^2 + \alpha^3$	(1 1 1 1)
α^{13}	$1 + \alpha^2 + \alpha^3$	(1 0 1 1)
α^{14}	$1 + \alpha^3$	(1 0 0 1)

Exemplo 2.12. Considere o corpo de Galois $GF(2^4)$ dado pela Tabela 2.8. Seja $m = 4$, o polinômio $P(X) = 1 + X + X^4$ é um polinômio primitivo sobre $GF(2)$, de acordo com o Exemplo 2.9.

Seja $\beta = \alpha^3 \in GF(2^4)$. Os conjugados de β são

$$\beta^2 = (\alpha^3)^2 = \alpha^6.$$

$$\beta^{2^2} = (\alpha^3)^{2^2} = \alpha^{12}.$$

$$\beta^{2^3} = (\alpha^3)^{2^3} = \alpha^{24}.$$

Usando os dados da Tabela 2.8 e sabendo que $2^m - 1 = 2^4 - 1 = 15$ elementos não-nulos de $GF(2^4)$ formam as raízes de $X^{2^m-1} + 1$ temos $\alpha^{15} + 1 = 0 \Rightarrow \alpha^{15} = 1$. Usando este fato temos:

$$\beta^{2^3} = \alpha^{24} = \alpha^{15} \cdot \alpha^9 = 1 \cdot \alpha^9 = \alpha^9.$$

O polinômio minimal de $\beta = \alpha^3$ é:

$$\begin{aligned}
\phi(X) &= (X+\alpha^3)(X+\alpha^6)(X+\alpha^{12})(X+\alpha^9) \\
\phi(X) &= [X^2+\alpha^3X+\alpha^6X+\alpha^9][X^2+\alpha^{12}X+\alpha^9X+\alpha^{21}] \\
\phi(X) &= [X^2+(\alpha^3+\alpha^6)X+\alpha^9][X^2+(\alpha^{12}+\alpha^9)X+\alpha^{21}] \\
\phi(X) &= X^4+(\alpha^2+\alpha^8)X^3+(\alpha^6+\alpha^{10}+\alpha^9)X^2+(\alpha^{17}+\alpha^8)X+\alpha^{15}
\end{aligned}$$

onde

$$\alpha^2+\alpha^8 = \alpha^2+1+\alpha^2 = 1$$

$$\alpha^6+\alpha^{10}+\alpha^9 = \alpha^2+\alpha^3+1+\alpha+\alpha^2+\alpha+\alpha^3 = 1$$

$$\alpha^{17}+\alpha^8 = \alpha^{15}\cdot\alpha^2+\alpha^8 = 1\cdot\alpha^2+1+\alpha^2 = 1$$

$\phi(X) = X^4+X^3+X^2+X+1$.

Veremos com detalhes no Capítulo 3 que um código de bloco é um subespaço vetorial. Portanto, precisamos definir os conceitos de espaço vetorial e subespaço, assim como a noção de transformação linear.

2.7 Espaços Vetoriais

Definição 2.24. Sejam um corpo K, cujos elementos serão chamados de escalares, e um conjunto V, cujos elementos serão chamados de vetores. Diremos que V é um *espaço vetorial* sobre K, ou um *K-espaço vetorial*, se existir a operação de adição em V

$$\begin{aligned}
+: \quad V\times V &\longrightarrow V \\
(v,w) &\mapsto v+w
\end{aligned}$$

e se existir a operação de multiplicação dos elementos de V por escalares,

$$\begin{aligned}
\cdot: \quad K\times V &\longrightarrow V \\
(\lambda,v) &\mapsto \lambda v
\end{aligned}$$

satisfazendo as seguintes propriedades:

1. $(u+v)+w = u+(v+w), \quad \forall u, v, w \in V$. Associatividade da adição

2. $u+v = v+u, \quad \forall u, v \in V$. Comutatividade da adição

3. Existe um elemento *neutro* $0 \in V$ tal que

$$u + 0 = u.$$

4. Dado um elemento $u \in V$, existe um elemento inverso $-u$, chamado *simétrico* de u, tal que,

$$u + (-u) = 0.$$

5. Dados $\lambda, \mu \in K$ e $u \in V$, vale

$$(\lambda + \mu) \cdot u = \lambda \cdot u + \mu \cdot u.$$

6. Dados $\lambda \in K$ e $u, v \in V$, vale

$$\lambda \cdot (u + v) = \lambda \cdot u + \lambda \cdot v.$$

7. Dados $\lambda, \mu \in K$ e $u \in V$, vale

$$(\lambda \cdot \mu) \cdot u = \lambda \cdot (\mu \cdot u).$$

8. Para todo $u \in V$,

$$1 \cdot u = u,$$

onde 1 é a unidade de K.

Exemplo 2.13. $\mathbb{R}$*-espaços vetoriais* $\mathbb{R}^n$ *e* $\mathbb{C}$*-espaços vetoriais* $\mathbb{C}^n$.

Definição 2.25. Um *subespaço vetorial* de um K-espaço vetorial V é um subconjunto não vazio W de V, que, com as operações de adição e multiplicação por escalares de V, que é também um K-espaço vetorial.

Podemos afirmar que um subconjunto não vazio W de um espaço vetorial V é um subespaço vetorial se é satisfeita a seguinte condição:

$$\forall u, v \in W, \quad \forall \lambda \in K, \quad u + \lambda \cdot v \in W. \tag{2.24}$$

Se V um K-espaço vetorial, dados $v_1, \ldots, v_n \in V$ dizemos que $v_1, \ldots, v_n$ são *linearmente independentes*, se for satisfeita a seguinte relação

$$\lambda_1 v_1 + \ldots + \lambda_n v_n = 0 \Rightarrow \lambda_1 = \ldots = \lambda_n = 0 \tag{2.25}$$

com $\lambda_1, \ldots, \lambda_n \in K$. E, caso contrário, $v_1, \ldots, v_n$ são ditos *linearmente dependentes*.

Diremos que um subconjunto $B \subset V$ *gera* V quando todo elemento de V puder ser escrito na forma

$$\lambda_1 v_1 + \ldots + \lambda_n v_n \tag{2.26}$$

com $v_1, \ldots, v_n \in B$ e $\lambda_1, \ldots, \lambda_n \in K$.

Quando um subconjunto $B \subset V$ gera V e os elementos de qualquer subconjunto finito de B forem linearmente independentes sobre K, diremos que B é uma *base* de V. O número de elementos de uma base será chamado *dimensão* de V sobre K e denotado por $\dim_K V$.

2.7.1 Transformações Lineares

Definição 2.26. Sejam V e W dois K-espaços vetoriais. Diremos que uma função $T : V \to W$ é uma *transformação linear* quando for satisfeita a seguinte condição:

$$\forall u, v \in V, \quad \forall \lambda \in K, \quad T(u + \lambda v) = T(u) + \lambda T(v) \tag{2.27}$$

Seja $T : V \to W$ uma transformação linear. Definimos o *núcleo* de T como sendo o K-subespaço vetorial de V definido por

$$\ker T = \{v \in V; T(v) = 0\}.$$

E, definimos a *imagem* de T como sendo o subespaço vetorial de W definido por

$$Im\, T = \{T(v); v \in V\}.$$

Teorema 2.8 (Teorema do Núcleo e da Imagem). Sejam V e W espaços vetoriais de dimensão finita sobre um corpo K. Para toda transformação linear $T : V \to W$ tem-se

$$\dim_K V = \dim_K \ker T + \dim_K Im\, T. \tag{2.28}$$

Dada uma transformação linear $T : V \to W$ entre espaços vetoriais de dimensão finita e uma base $v_1, \ldots, v_n$ de V e uma base $w_1, \ldots, w_n$ de W, temos que

$$T(v_i) = \lambda_{i1} w_1 + \lambda_{i2} w_2 + \ldots + \lambda_{im} w_m; \quad i = 1, \ldots, n. \tag{2.29}$$

Se $v = x_1 v_1 + \ldots + x_n v_n$, então $(x_1, \ldots, x_n)$ são as *coordenadas* de v na base $v_1, \ldots, v_n$. Note que é possível exibir uma representação matricial para a transformação T, pois as coordenadas de $w = T(v)$ na base $w_1, \ldots, w_m$ de W são dadas pelo produto matricial

$$(x_1, \ldots, x_n) \cdot \begin{pmatrix} \lambda_{11} & \lambda_{12} & \ldots & \lambda_{1m} \\ \lambda_{21} & \lambda_{22} & \ldots & \lambda_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{n1} & \lambda_{n2} & \ldots & \lambda_{nm} \end{pmatrix}. \tag{2.30}$$

Assim, concluímos a fundamentação matemática básica necessária para introduzir a Teoria dos códigos corretores de erros. No próximo capítulo vamos introduzir a estrutura dos códigos de bloco lineares, que são a classe de códigos mais utilizada na prática [9]. Em seguida, serão abordados os códigos cíclicos, que formam uma subclasse de códigos lineares com grande aplicação prática devido a sua estrutura cíclica, o que permite implementar bons algoritmos de codificação e decodificação.

CAPÍTULO 3

CÓDIGOS CORRETORES DE ERROS

Em um sistema de comunicação uma mensagem é transmitida por um canal de comunicação o qual está sujeito a interferências, comumente chamadas de ruído, e por esse motivo são necessárias medidas para garantir a confiabilidade e qualidade dessa transmissão.

Os ruídos fazem com que a mensagem recebida seja diferente da mensagem enviada e, para a transmissão da informação com confiabilidade, existe a necessidade de desenvolver métodos capazes de detectar e corrigir erros[11]. Daí surge a codificação para o controle de erros, que envolve o uso de um codificador de canal no transmissor e um algoritmo de decodificação no receptor.

A comunicação é o processo de transferência de uma informação de uma fonte em um ponto para um usuário no destino[2]. Considere o sistema de comunicação digital constituído pelos seguintes elementos: fonte, codificador de canal, canal, decodificador de canal e usuário, dado pela Figura 3.1.

Figura 3.1: Elementos de sistema de comunicação digital

A fonte de informação pode ser de natureza analógica ou digital. Quando for de natureza

analógica como, por exemplo, sinais de voz, essa informação precisa ser digitalizada e transformada em símbolos discretos. Em seguida, a informação é codificada no codificador de canal introduzindo redundâncias na informação para que possíveis erros introduzidos no canal ruidoso possam ser detectados no decodificador de canal.

A unidade de informação denomina-se bit (uma contração de binary digit - dígito binário). A taxa de código é a razão entre a quantidade de bits que entram num codificador e a quantidade de bits que saem do codificador, denotada por R. E definimos a capacidade de um canal C como sendo a medida em bits por uso de canal.

Shannon [1] mostrou que se um canal tem capacidade C e uma fonte tem uma taxa de código R, então existe uma técnica de codificação apropriada tal que os símbolos produzidos pela fonte podem ser transmitidos pelo canal com uma probabilidade de erro arbitrariamente pequena.

Os códigos gerados pelo codificador de canal são chamados códigos corretores de erros e podem ser classificados basicamente em códigos de bloco e códigos convolucionais [10]. Essa classificação é baseada na presença ou não de memória nos codificadores, assim, os códigos de bloco são ditos sem memória e os códigos convolucionais são ditos com memória, pois um determinado bit codificado depende de um ou mais bits de informação anteriores combinados linearmente.

Neste trabalho abordamos os códigos de blocos lineares, enfatizando o estudo dos códigos cíclicos, através do estudo do codificador BCH [5, 6], a ser discutido detalhadamente no Capítulo 4.

3.1 Códigos de Blocos Lineares

Uma informação a ser transmitida ou armazenada de forma digital, por razões práticas, é codificada em dígitos binários 0 e 1, portanto, iremos discutir os códigos de bloco com símbolos do corpo binário $GF(2)$.

Em um codificador de bloco, a sequência de informação binária é segmentada em blocos de mensagens com k bits, denotados por $u = (u_0, u_1, \ldots, u_{k-1})$, onde $u_i \in GF(2), i = 0, 1, \ldots, k-1$, assim teremos 2^k possíveis mensagens. Cada mensagem u é transformada em uma palavra-código v com n bits. Os $n - k$ bits introduzidos na mensagem u são chamados bits de verificação de paridade que são a redundância utilizada para o decodificador identificar se houve erros durante a transmissão e, se possível, corrigi-los.

Seja K um corpo finto com q elementos. Temos, portanto, para cada número natural n, um K-espaço vetorial K^n de dimensão n.

Definição 3.1. Um código de bloco de comprimento n e 2^k palavras código é dito *código linear* $\mathcal{C} \subset K^n$, denotado por (n, k), quando $\mathcal{C}$ for subespaço de dimensão k de K^n.

Seja (n, k) um código linear e seja $\{g_0, g_1, \ldots, g_{k-1}\}$ uma de suas bases, portanto, todo elemento do código se escreve de modo único na forma

$$v = u_0 g_0 + u_1 g_1 + \ldots + u_{k-1} g_{k-1} \tag{3.1}$$

onde $u_i, i = 0, 1, \ldots, k-1$ são coordenadas da mensagem $u = (u_0, u_1, \ldots, u_{k-1})$ de comprimento k.

Os vetores da base $\{g_0, g_1, \ldots, g_{k-1}\}$ formam a *matriz geradora* do código G de ordem $k \times n$.

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix} \tag{3.2}$$

onde $g_i = (g_{i0}, g_{i1}, \ldots, g_{i,n-1})$ para $0 \leq i < k$. Seja K um corpo finto. Considere a transformação

linear definida por:

$$T : K^k \longrightarrow K^n$$

$$u \longmapsto u \cdot G$$

Se $u = (u_0, u_1, \ldots, u_{k-1})$ é a mensagem original, então cada palavra código será dada por:

$$v = u \cdot G \tag{3.3}$$

$$v = (u_0, u_1, \ldots, u_{k-1}) \cdot \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix}$$

$$v = u_0 g_0 + u_1 g_1 + \ldots + u_{k-1} g_{k-1}.$$

Exemplo 3.1. Considere o corpo finito $K = GF(2)$. O código de bloco linear $\mathcal{C} \subset GF(2)^7$ de dimensão 4, denotado por $(7, 4)$, que é chamado de código de Hamming [3], cuja matriz geradora é dada por:

$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Seja $u = (1\ 1\ 0\ 1)$ a mensagem de entrada no codificador, então a palavra-código correspondente será:

$$v = u \cdot G$$

$$v = (1\,1\,0\,1) \cdot \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$v = (0\,0\,0\,1\,1\,0\,1).$$

Os códigos de Hamming formam uma classe de códigos lineares da forma $(2^m - 1, 2^m - 1 - m)$, para $m > 2$, que foram desenvolvidos em 1950 por Richard W. Hamming [3]. Tais códigos são bastante eficientes na correção de erros simples, ou seja, corrigem apenas um único erro. Na próxima seção veremos com detalhes como determinar a capacidade de detecção e correção de erros de um código a partir do conceito de distância mínima.

Os códigos de bloco lineares cujos k bits da mensagem original permanecem inalterados na palavra-código são chamados de códigos *sistemáticos*. Podemos observar um código de bloco sistemático no Exemplo 3.1.

$$u = (1\,1\,0\,1) \longrightarrow v = (0\,0\,0\,\underbrace{1\,1\,0\,1}_{u}).$$

Um código de bloco linear sistemático (n, k) é completamente determinado pela matriz geradora G de ordem $k \times n$ da forma padrão:

$$G = \left[\begin{array}{cccc|cccc} p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & \cdots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-,n-k-1} & 0 & 0 & \cdots & 1 \end{array}\right] \tag{3.4}$$

onde $p_{ij} \in GF(2), \forall\, 0 \leq i < k$ e $\forall\, 0 \leq j < n-k$. Assim,

$$G = [P \mid I_k]$$

onde I_k é a matriz identidade $k \times k$ e $P = (p_{ij})_{k \times n-k},\ p_{ij} \in GF(2)$. A partir de G, com algumas manipulações algébricas , extraímos uma matriz $H = [I_{n-k}\ P^T]$, chamada de *matriz de verificação de paridade*, que chamaremos simplesmente de *matriz de paridade*.

A matriz de paridade H é utilizada no processo de detecção e correção de erros, pois dada uma n-upla v podemos afirmar que v é uma palavra código de um código (n, k) gerado pela matriz $G = [P \mid I_k]$ se, e somente se, $v \cdot H^T = 0$. Portanto, a partir da matriz H podemos identificar quando uma mensagem v pertence ou não ao código.

Considere um código linear (n, k) com matriz geradora G e matriz de paridade H. Sejam $v = (v_0, v_1, \ldots, v_{n-1})$ uma palavra código transmitida por um canal ruidoso e $r = (r_0, r_1, \ldots, r_{n-1})$ um vetor recebido no decodificador. Devido aos ruídos do canal de transmissão o vetor recebido r pode ser diferente de v. Então, podemos dizer que essa diferença é o *padrão de erro* dado por:

$$e = r + v = (e_0, e_1, \ldots, e_{n-1}). \tag{3.5}$$

Note que $e_i = 1$ implica que houve erro na i-ésima coordenada e, caso contrário, teremos $e_i = 0$, pois estamos considerando as operações no corpo de Galois $GF(2)$. Além disso, podemos reescrever $r = v + e$. No processo de decodificação, ao receber uma mensagem r o decodificador precisa primeiramente determinar se existem ou não erros de transmissão, para isto, o decodificador calcula a *síndrome* de r:

$$s = r \cdot H^T = (s_0, s_1, \ldots, s_{n-k-1}). \tag{3.6}$$

Se r é uma palavra código, então $s = r \cdot H^T = 0$. Nesse caso não existem erros, ou seja, $e = 0$. No entanto, a recíproca não é verdadeira. É possível haver erros que não possam ser detectados, isto

é, $s = 0$ mas r não é a palavra código transmitida. Isto ocorre quando temos um padrão de erro não-nulo e que pertence ao código. Nesse caso $r = v + e$ é a soma de duas palavras código e temos $s = r \cdot H^T = 0$. Portanto, estamos diante de um *erro de decodificação*.

Podemos afirmar que $s \neq 0 \Leftrightarrow r$ contém algum erro. Além disso, um fato importante que podemos observar é que a síndrome depende apenas do padrão de erro, e não da palavra-código transmitida, pois $s \cdot H^T = (v + e) \cdot H^T = v \cdot H^T + e \cdot H^T$, mas como v é palavra-código então $v \cdot H^T = 0$. Logo, $s = e \cdot H^T$.

Assim, vimos que é possível detectar erros introduzidos no canal de transmissão. Contudo, nem todos os padrões de erro podem ser corretamente decodificados. A capacidade de correção de erros de um código de bloco depende diretamente de sua distância mínima, que definiremos na próxima seção.

Exemplo 3.2. Considere o código de Hamming $(7, 4)$ dado pela Tabela 3.1 cuja matriz geradora é dada por:

$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Se $u = (1\ 0\ 1\ 1)$ é uma mensagem a ser codificada, a palavra-código é dada por:

$$v = u \cdot G = (1\ 0\ 1\ 1) \cdot \left[\begin{array}{ccc|cccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

$$v = (1\ 0\ 0\ 1\ 0\ 1\ 1).$$

Tabela 3.1: Código de Hamming $(7, 4)$

Mensagens	Palavras-código
0 0 0 0	0 0 0 0 0 0 0
1 0 0 0	1 1 0 1 0 0 0
0 1 0 0	0 1 1 0 1 0 0
1 1 0 0	1 0 1 1 1 0 0
0 0 1 0	1 1 1 0 0 1 0
1 0 1 0	0 0 1 1 0 1 0
0 1 1 0	1 0 0 0 1 1 0
1 1 1 0	0 1 0 1 1 1 0
0 0 0 1	1 0 1 0 0 0 1
1 0 0 1	0 1 1 1 0 0 1
0 1 0 1	1 1 0 0 1 0 1
1 1 0 1	0 0 0 1 1 0 1
0 0 1 1	0 1 0 0 0 1 1
1 0 1 1	1 0 0 1 0 1 1
0 1 1 1	0 0 1 0 1 1 1
1 1 1 1	1 1 1 1 1 1 1

Note que a matriz G está na forma sistemática:

$$G = [P \mid I_4].$$

Logo, a matriz de paridade pode ser facilmente encontrada, pois sabemos que é da forma:

$$H = [I_3 \mid P^T]$$

$$H = \left[\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right].$$

Seja $v = (1\,0\,0\,1\,0\,1\,1)$ a palavra-código transmitida e $r = (1\,0\,0\,1\,0\,0\,1)$ o vetor recebido.

Para determinar se houve erro na transmissão, o decodificador calcula a síndrome de r, dada por

$$s = r \cdot H^T$$

$$s = (1\,0\,0\,1\,0\,0\,1) \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = (1\,1\,1).$$

Como $s \neq 0$, então r não pertence ao código de bloco linear $(7, 4)$ dado pela Tabela 3.1. Para encontrar o padrão de erro basta usar o fato que $s = e \cdot H^T$ e temos que:

$$(1\,1\,1) = (e_0\ e_1\ e_2\ e_3\ e_4\ e_5\ e_6) \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \Rightarrow$$

$$(1\,1\,1) = (e_0 + e_3 + e_5 + e_6 \quad e_1 + e_3 + e_4 + e_5 \quad e_2 + e_4 + e_5 + e_6).$$

O padrão de erro com menor número de coordenadas diferente de zero, ou seja, que tem o menor número de erros e que satisfaz a equação acima é o vetor $e = (0\ 0\ 0\ 0\ 0\ 1\ 0)$. Considerando um canal binário simétrico (BSC), no qual a probabilidade de transmitir um dígito errado é igual para os dígitos 1 e 0, então o vetor $e = (0\ 0\ 0\ 0\ 0\ 1\ 0)$ é o vetor erro mais provável.

Logo, a palavra-código decodificada será:

$$
\begin{aligned}
v^* &= r + e \\
v^* &= (1\,0\,0\,1\,0\,0\,1) + (0\,0\,0\,0\,0\,1\,0) \\
v^* &= (1\,0\,0\,1\,0\,1\,1).
\end{aligned}
$$

3.2 Distância Mínima de um Código de Bloco

Considere um canal de transmissão no qual a probabilidade de receber uma palavra código sem erros é maior do que a probabilidade de recebermos com um erro, que deve ser maior do que a de recebermos com dois erros, e assim por diante. Além disso, considere que neste canal todas as palavras código são igualmente prováveis de serem transmitidas. Então o receptor decide qual a palavra código foi efetivamente transmitida adotando a regra da máxima semelhança, ou seja, o receptor decide como palavra código válida a que mais se assemelha à palavra recebida, na qual se requer que tenha ocorrido o menor número de erros. A fim de determinar o número de erros que podem ser detectados e se esses erros podem ser corrigidos precisamos definir um parâmetro muito importante, denominado *distância mínima*.

Definição 3.2. Seja $v = (v_0, v_1, \ldots, v_{n-1}) \in V^n$ um espaço vetorial sobre $GF(2)$. O *peso de Hamming* de v, denotado por $w(v)$ é definido como o número de bits diferentes de zero em v.

Exemplo 3.3. Seja $v = (1\,0\,0\,1\,0\,1\,0)$, então o peso de Hamming de v é $w(v) = 3$.

Definição 3.3. Considere $v, v' \in V^n$ um espaço vetorial sobre $GF(2)$. A *distância de Hamming* entre v e v' denotada por $d(v, v')$ é definida como o número de coordenadas em que v e v' diferem, isto é,

$$d(v, v') = |\{i;\ v_i \neq v'_i,\ 0 \leq i \leq n-1\}|.$$

Definição 3.4. Dado um código $\mathcal{C} \subset V^n$, então a *distância mínima* de $\mathcal{C}$ denotada por $d_{\min}$ é dada

por:

$$d_{\min} = min\{d(v, v'); v, v' \in \mathcal{C}, v \neq v'\}.$$

E o *peso mínimo* do código $\mathcal{C}$ é denotado por $w_{\min}$ é dado por:

$$w_{\min} = \min\{w(v); v \in \mathcal{C}, v \neq 0\}.$$

A distância de Hamming é uma *métrica*, também chamada de *métrica de Hamming*, portanto, valem as seguintes propriedades para $v, v', v'' \in V^n$:

(i) $d(v, v') \geq 0$ e $d(v, v') = 0 \Leftrightarrow v = v'$.

(ii) $d(v, v') = d(v', v)$.

(iii) $d(v, v') \leq d(v, v'') + d(v'', v')$. (Desigualdade triangular)

Seja V^n um espaço vetorial sobre $GF(2)$. Podemos definir os conceitos de bola e esfera em V^n, tal como é feito em qualquer espaço métrico.

Definição 3.5. Dado $x \in V^n$ e $s \in \mathbb{Z}_+^*$, chama-se *bola* de centro em x e de raio s ao conjunto

$$B(x, s) = \{v \in V^n : d(v, x) \leq s\}$$

e *esfera* de centro em x e raio s, ao conjunto

$$S(x, s) = \{v \in V^n : d(v, x) = s\}.$$

Agora, podemos estabelecer um importante teorema referente à detecção e correção de erros, pois ao receber uma mensagem r podemos detectar se ela contém erros e decidir se r pertence ou não ao código $\mathcal{C}$. Para isto, consideraremos o canal BSC, e assim o critério de correção será substituir o elemento r pelo elemento $v \in \mathcal{C}$ mais próximo de r.

Teorema 3.1. Seja $\mathcal{C}$ um código com distância mínima d. Então $\mathcal{C}$ pode corrigir até $\mathcal{K} = \frac{d-1}{2}$ e detectar até $d - 1$ erros.

Demonstração: Seja $v \in \mathcal{C}$ e suponha que ele foi recebido como outro elemento r, com $t \leq d-1$ erros. Como o número t de erros acontecidos é precisamente a distância de Hamming de v a r temos que $d(v,r) \leq d-1 < d$. Como d é a distância mínima de $\mathcal{C}$, então $r \notin \mathcal{C}$ e, portanto, o erro pode ser detectado.

Suponhamos ainda que o número t de erros é menor que $\mathcal{K}$. Considere a bola $B(r,\mathcal{K})$ de centro em r e raio $\mathcal{K}$. Como $d(v,r) = t \leq \mathcal{K} \Rightarrow v \in B(r,\mathcal{K})$.

Afirmação 3.1. v é o único elemento de $\mathcal{C}$ contido na bola $B(r,\mathcal{K})$.

De fato, se existisse outro elemento $v' \in \mathcal{C}$ tal que $v' \in B(r,\mathcal{K})$ teríamos:

$$d(v,v') \leq d(v,r) + d(r,v') \leq 2\mathcal{K} < d$$

mas isto é uma contradição, pois d é a distância mínima do código $\mathcal{C}$. Logo, v é o elemento de $\mathcal{C}$ mais próximo de r e é possível corrigir o erro. ■

Proposição 3.1. (Cota de Singleton) Os parâmetros (n,k,d) de um código linear satisfazem à desigualdade

$$d \leq n - k + 1$$

Exemplo 3.4. Considere o código de Hamming dado pela Tabela 3.1. Vamos calcular a distância de Hamming entre a palavra-código $v = (1\ 1\ 0\ 1\ 0\ 0\ 0)$ e algumas palavras-código:

$d(1\ 1\ 0\ 1\ 0\ 0\ 0\ ,0\ 1\ 1\ 0\ 1\ 0\ 0) = 4$

$d(1\ 1\ 0\ 1\ 0\ 0\ 0\ ,1\ 1\ 1\ 0\ 0\ 1\ 0) = 3$

$d(1\ 1\ 0\ 1\ 0\ 0\ 0\ ,1\ 0\ 0\ 0\ 1\ 1\ 0) = 4$

$d(1\ 1\ 0\ 1\ 0\ 0\ 0\ ,0\ 0\ 1\ 0\ 1\ 1\ 1) = 7$

Calculando a distância de Hamming entre todas as palavras-código obtemos $d_{\min} = 3$. Este fato ocorre para qualquer código de Hamming. Portanto, pelo Teorema 3.1 podemos verificar que todo código de Hamming é capaz de detectar até 2 erros e corrigir no máximo 1 erro.

Assim, podemos observar que os códigos de Hamming formam um classe de códigos capazes de corrigir apenas erros simples. Contudo, existe a necessidade de desenvolver códigos eficientes na correção de erros múltiplos. Além disso, o fato dos códigos de bloco lineares serem completamente descritos por sua matriz geradora e que, para verificar se uma determinada palavra pertence ao código é utilizada uma matriz de paridade, implicam num alto custo computacional. De fato, conforme aumenta-se a dimensão da matriz aumentamos a custo computacional da decodificação.

No entanto, observa-se que esse custo pode ser reduzido se pudermos descrever o código utilizando o mínimo de informação armazenada. Assim, surgem os códigos cíclicos, que podem ser representados por polinômios. Tais códigos são importantes por serem facilmente implementados em hardware, utilizando registradores de deslocamento e, além disso, porque podemos construir códigos cíclicos capazes de corrigir erros múltiplos como, por exemplo, os códigos BCH.

3.3 Códigos Cíclicos

Os códigos cíclicos formam uma subclasse importante dos códigos de bloco lineares que foram desenvolvidos inicialmente por Eugene Prange em 1957 [4]. Desde seu desenvolvimento surgiram várias classes de códigos cíclicos, dentre elas a classe dos códigos BCH, que serão discutidos com detalhes no próximo capítulo.

Definição 3.6. Seja K um corpo finito. Um código linear $\mathcal{C} \subset K^n$ é chamado de *código cíclico* se para todo $v = (v_0, v_1, \ldots, v_{n-1}) \in \mathcal{C}$ o vetor $v^{(i)} = (v_{n-i}, v_{n-i+1}, \ldots, v_{n-1}, v_0, v_1, \ldots, v_{n-i-1}) \in \mathcal{C}$.

Exemplo 3.5. $C_1 = \{0000, 1010, 0101, 1111\}$ é um código cíclico. $C_2 = \{0000, 1001, 0110, 1111\}$ não é um código cíclico.

Consideraremos o corpo binário $K = GF(2)$. Defina R_n como sendo o anel das classes residuais

em $K[X]$ módulo $X^n + 1$, isto é,

$$R_n = K[X]_{(X^n+1)}$$

. Um elemento de R_n é um conjunto da forma:

$$[f(X)] = \{f(X) + g(X)(X^n + 1); g(X) \in K[X]\}$$

e a adição e multiplicação são definidas, respectivamente, por:

$$[f_1(X)] + [f_2(X)] = [f_1(X) + f_2(X)]$$

$$[f_1(X)] \cdot [f_2(X)] = [f_1(X) \cdot f_2(X)]$$

Note que R_n munido com a multiplicação por escalares $\lambda \in K$, definida por

$$\lambda[f(X)] = [\lambda \cdot f(X)]$$

é um K-espaço vetorial de dimensão n com base $1, [X], \cdots, [X^{n-1}]$. Observe que R_n é isomorfo a K^n através da transformação linear:

$$T : K^n \longrightarrow R_n$$

$$(a_0, \cdots, a_{n-1}) \longmapsto [a_0 + a_1X + \cdots + a_{n-1}X^{n-1}]$$

Portanto, dada uma palavra código v podemos reescrevê-la como um polinômio

$$v(X) = v_0 + v_1X + \ldots + v_{n-1}X^{n-1}$$

e assim cada palavra código correspondente a um polinômio com $\partial v(X) \leq n-1$, chamado *polinômio código*.

Teorema 3.2. Um subespaço C de K^n é um código cíclico se, e somente se, $T(C)$ é um ideal de R_n.

Teorema 3.3. Dado um código cíclico C, existe um $v \in C$ tal que $C =< v >$.

As demonstrações do Teorema 3.2 e do Teorema 3.3 podem ser consultadas em [9].

Como podemos observar os códigos cíclicos apresentam a estrutura algébrica de um ideal, que simplifica a implementação desses códigos. Veremos a seguir algumas características importantes dos códigos cíclicos decorrentes deste fato.

O polinômio código correspondente a palavra código $v^{(i)}$ é

$$v^{(i)}(X) = v_{n-i} + v_{n-i+1}X + \cdots + v_{n-1}X^{i-1} + v_0X^i + v_1X^{i+1} + \cdots + v_{n-i-1}X^{n-1}.$$

Podemos observar uma relação algébrica interessante entre $v(X)$ e $v^{(i)}(X)$. Multiplicando $v(X)$ por X^i e fazendo algumas manipulações algébricas, obtemos:

$$\begin{aligned} X^i v(X) &= v_0X^i + v_1X^{i+1} + \cdots + v_{n-i-1}X^{n-1} + \cdots + v_{n-1}X^{n+i-1} \\ X^i v(X) &= (v_{n-i} + v_{n-i+1}X + \cdots + v_{n-1}X^{i-1})(X^n + 1) + v^{(i)}(X) \\ X^i v(X) &= q(X)(X^n + 1) + v^{(i)}(X) \end{aligned}$$

ou ainda

$$v^{(i)} = X^i v(X) \bmod (X^n + 1). \tag{3.7}$$

Exemplo 3.6. Seja $v = (1\ 1\ 0\ 1)$ para $n = 4$ considere um deslocamento de ordem 3 da palavra-código v. Então teremos:

$$v(X) = 1 + X + X^3$$

$$X^3 v(X) = X^3 + X^4 + X^6.$$

Dividindo $X^3 v(X)$ por $X^4 + 1$ temos:

$$\begin{array}{lll} X^6 \quad + X^4 + X^3 & & |\underline{X^4 + 1} \\ \underline{X^6 \qquad\qquad + X^2} & & X^2 + 1 \\ \quad X^4 + X^3 + X^2 & & \\ \underline{\quad X^4 \qquad\qquad + 1} & & \\ \quad X^3 + X^2 \quad + 1 & & \end{array}$$

Logo, $X^3 v(X) = (X^2 + 1)(X^4 + 1) + (X^3 + X^2 + 1)$. Daí, $v^{(3)}(X) = X^3 + X^2 + 1 = 1 + X^2 + X^3$, que corresponde a palavra-código $v^{(3)} = (1\ 0\ 1\ 1)$. Note que $v^{(3)}$ pode ser obtida deslocando-se $v = (1\ 1\ 0\ 1)$ três vezes.

A fim de determinar quais as condições necessárias para gerar códigos cíclicos vejamos alguns teoremas importantes a seguir.

Teorema 3.4. O polinômio código não nulo de grau mínimo em um código cíclico $\mathcal{C}$ é único.

Demonstração: Seja $g(X) = g_0 + g_1 X + \cdots + g_{r-1} X^{r-1} + X^r$ não nulo de grau mínimo r, tal que $g(x) \in \mathcal{C}$. Suponhamos, por absurdo, que $g(X)$ não seja único. Então, existe outro polinômio código de grau r, $g'(X) = g'_0 + g'_1 X + \cdots + g'_{r-1} X^{r-1} + X^r$. Como $\mathcal{C}$ é linear $g(X) + g'(X) \in \mathcal{C}$, isto é,

$$\begin{aligned} g(X) + g'(X) &= (g_0 + g'_0) + (g_1 + g'_1)X + \cdots + (g_{r-1} + g'_{r-1})X^{r-1} + (1+1)X^r \\ &= (g_0 + g'_0) + \cdots + (g_{r-1} + g'_{r-1})X^{r-1} \in \mathcal{C}. \end{aligned}$$

Se $g(X) + g'(X) \neq 0$, implica que existe um polinômio código não nulo de grau menor do que r. Isto é um absurdo. Logo, $g(X) + g'(X) = 0$. Isto implica que $g(X) = g'(X)$. Portanto, $g(X)$ é único.

■

Teorema 3.5. Seja $g(X) = g_0 + g_1 X + \cdots + g_{r-1} X^{r-1} + X^r$ um polinômio código não nulo de grau mínimo em um código cíclico $\mathcal{C}$. Então, o coeficiente g_0 deve ser igual a 1.

Demonstração: Suponha que $g_0 = 0$, então

$$\begin{aligned} g(X) &= g_1X + g_2X^2 + \cdots + g_{r-1}X^{r-1} + X^r \\ &= X(g_1 + g_2X + \cdots + g_{r-1}X^{r-2} + X^{r-1}). \end{aligned}$$

Se deslocarmos $g(X)$ ciclicamente $n-1$ vezes, obtemos um polinômio código não nulo $g_1 + g_2X + \cdots + X^{r-1}$ com grau menor que r. Mas isto é uma contradição, pois assumimos que $g(X)$ é o polinômio código de grau mínimo. Portanto, $g_0 \neq 0$.

■

Teorema 3.6. Seja $G(X) = 1 + g_1X + \cdots + g_{r-1}X^{r-1} + X^r$ o polinômio código de grau mínimo num código cíclico (n, k). Um polinômio binário de grau menor ou igual a $n-1$ é um polinômio código se, e somente se, for múltiplo de $g(X)$.

Demonstração: Seja $v(X)$ um polinômio $\partial v(X) \leq n-1$. suponha que $v(X)$ é múltiplo de $g(X)$. Então,

$$\begin{aligned} v(X) &= (u_0 + u_1X + \cdots + u_{n-r-1}X^{n-r-1})g(X) \\ &= u_0g(X) + u_1Xg(X) + \cdots + u_{n-r-1}X^{n-r-1}g(X). \end{aligned}$$

Note que da Equação (3.7) obtemos os deslocamentos cíclicos de $g(X)$:

$$\begin{aligned} Xg(X) &= g^{(1)}(X) \\ X^2g(X) &= g^{(2)}(X) \\ \vdots &\quad \vdots \\ X^{(n-r-1)}g(X) &= g^{(n-r-1)}(X). \end{aligned}$$

Como $\mathcal{C}$ é linear, a soma de duas palavras códigos deve pertencer ao código. Então, podemos dizer que $v(X) = u_0g(X) + u_1Xg(X) + \cdots + u_{n-r-1}X^{n-r-1}g(X) \in \mathcal{C}$. Com isso, mostramos que $v(X) \in \mathcal{C}$. Resta mostrar que $v(X)$ é múltiplo de $g(X)$.

Dividindo $v(X)$ por $g(X)$ obtemos:

$$v(X) = u(X)g(X) + b(X) \tag{3.8}$$

onde $b(X) \equiv 0$ ou $\partial b(X) < \partial g(X)$. Reescrevendo a equação obtemos:

$$b(X) = v(X) + u(X)g(X). \tag{3.9}$$

Segue-se da primeira parte do teorema que $v(X)$ e $u(X)g(X) \in \mathcal{C}$, $b(X)$ também deverá pertencer a $\mathcal{C}$, pois $\mathcal{C}$ é linear. Se $b(X) \neq 0$, então é um polinômio de grau menor do que o grau de $g(X)$. Absurdo, pois supomos $g(X)$ o polinômio não nulo de grau mínimo em $\mathcal{C}$. Assim, $b(X) \equiv 0$. Isto prova a segunda parte do teorema, e $v(X)$ é múltiplo de $g(X)$.

■

Teorema 3.7. Em um código cíclico (n, k), existe um único polinômio de grau $n - k$.

$$g(X) = 1 + g_1X + g_2X^2 + \cdots + g_{n-k-1}X^{n-k-1} + X^{n-k}.$$

O número de polinômios binários de grau menor ou igual a $n - 1$ que são múltiplos de $g(X)$ é 2^{n-r}. Pelo teorema anterior temos que estes polinômios formam todos os polinômios do código cíclico (n, k). Como existem 2^k polinômios em (n, k), então $2^{n-r} = 2^k \Rightarrow n-r = k \Rightarrow r = n-k$. Portanto, o polinômio de grau mínimo, não-nulo, em um código cíclico (n, k) é da seguinte forma:

$$g(X) = 1 + g_1X + g_2X^2 + \cdots + g_{n-k-1}X^{n-k-1} + X^{n-k}.$$

Todo polinômio código é múltiplo de $g(X)$, e todo polinômio binário de grau menor ou igual a $n - 1$, que é múltiplo de $g(X)$, é um polinômio código.

Sejam os coeficientes de $u(X) = u_0 + u_1X + \cdots + u_{k-1}X^{k-1}$ os k dígitos de informação a ser codificada e $v(X)$ o respectivo polinômio código. Um código cíclico (n, k) pode ser completamente caracterizado pelo seu polinômio de grau mínimo, não-nulo, $g(X)$, pois qualquer polinômio código pode ser obtido multiplicando $u(X)$ por $g(X)$.

$$v(X) = u(X)g(X). \tag{3.10}$$

O polinômio $g(X)$ é chamado *polinômio gerador do código* e o $\partial g(X)$ é igual ao número de dígitos de paridade do código.

Teorema 3.8. O polinômio gerador $g(X)$ de um código cíclico (n, k) é um fator de $X^n + 1$.

Demonstração: Multiplicando $g(X)$ por X^k resulta num polinômio de grau n. Dividindo $X^k g(X)$ por $X^n + 1$, obtemos:

$$X^k g(X) = (X^n + 1) + g^{(k)}(X) \tag{3.11}$$

onde $g^{(k)}(X)$ é o resto, e é um polinômio código obtido deslocando $g(X)$ ciclicamente k vezes. Assim, $g^{(k)}(X)$ é múltiplo de $g(X)$, ou seja, $g^{(k)}(X) = a(X)g(X)$. Da Equação 3.7 obtemos:

$$X^n + 1 = \{X^k + a(X)\}g(X).$$

Portanto, $g(X)$ é um fator de $X^n + 1$.

■

Agora, considerando os resultados apresentados, somos capazes de determinar quando, para algum n, k, existe um código cíclico (n, k). Para isto, vejamos o seguinte teorema.

Teorema 3.9. Se $g(X)$ é um polinômio com $\partial g(X) = n - k$ e é um fator de $X^n + 1$, então $g(X)$ gera um código cíclico (n, k).

Demonstração: Considere os k polinômios $g(X), Xg(X), X^2g(X), \ldots, X^{k-1}g(X)$, todos com grau menor ou igual a $n - 1$. Combinando esses polinômios linearmente temos:

$$\begin{aligned} v(X) &= u_0 g(X) + u_1 X g(X) + \cdots + u_{k-1} X^{k-1} g(X) \\ &= (u_0 + u_1 X + \cdots + u_{k-1} X^{k-1}) g(X). \end{aligned}$$

Note que $v(X)$ é também um polinômio com $\partial v(X) \leq n - 1$ e é múltiplo de $g(X)$. Existem um total de 2^k polinômios desses e formam um código linear (n, k).

Seja $v(X) = v_0 + v_1 X + \cdots + v_{n-1} X^{n-1}$ um polinômio código neste código. Multiplicando $v(X)$ por X obtemos

$$\begin{aligned} Xv(X) &= v_0X + v_1X^2 + \cdots + v_{n-2}X^{n-1} + v_{n-1}X^n \\ &= v_{n-1}(X^n + 1) + (v_{n-1} + v_0X + \cdots + v_{n-2}X^{n-1}) \\ &= v_{n-1}(X^n + 1) + v^{(1)}(X) \end{aligned}$$

onde $v^{(1)}(X)$ é um deslocamento cíclico de $v(X)$. Uma vez que $Xv(X)$ e $X^n + 1$ são divisíveis por $g(X), v^{(1)}(X)$ deve ser divisível por $g(X)$. Assim, $v^{(1)}(X)$ é um múltiplo de $g(X)$ e é uma combinação linear de $g(X), Xg(X), \ldots, X^{k-1}g(X)$. Então, $v^{(1)}(X)$ é um múltiplo de $g(X)$ e é uma combinação linear de $g(X), Xg(X), \ldots, X^{k-1}g(X)$. Então, $v^{(1)}(X)$ é também um polinômio código. Pela Definição 3.6 o código linear gerado por $g(X), Xg(X), \ldots, X^{k-1}g(X)$ é um código cíclico (n, k).

■

3.3.1 Codificação de Códigos Cíclicos

Como vimos na Seção 3.1 o processo de codificação de um código de bloco linear é descrito pela Equação 3.3. Portanto, precisamos agora definir a forma da matriz geradora para um código cíclico.

Considere o código cíclico (n, k) com polinômio gerador

$$g(X) = 1 + g_1X + \cdots + g_{n-k-1}X^{n-k-1} + X^{n-k}.$$

Mostramos que o código (n, k) é gerado pelos k polinômios códigos $g(X), Xg(X), \ldots, X^{k-1}g(X)$ em $\mathcal{C}$. Portanto, a matriz geradora do código será dada por:

$$G = \begin{bmatrix} 1 & g_1 & g_2 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & g_1 & \cdots & g_{n-k-1} & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & g_{n-k-2} & g_{n-k-1} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & g_1 & g_2 & \cdots & 1 \end{bmatrix}. \tag{3.12}$$

Exemplo 3.7. Considere o código de Hamming $(7, 4)$ com polinômio gerador $g(X) = 1 + X + X^3$. A matriz geradora será:

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

É interessante notar que para o processo de decodificação precisamos determinar a matriz de paridade H. No entanto, a determinação da matriz H se torna fácil se tivermos a matriz G na forma sistemática.

Considere a matriz G definida pela Equação (3.12), que nem sempre está na forma sistemática. Dividindo X^{n-k+i} por $g(X)$ para $i = 0, 1, \ldots, k-1$, obtemos:

$$X^{n-k+i} = a_i(X)g(X) + b_i(X)$$

onde $b_i(X)$ é o resto da divisão e podemos escrever:

$$b_i(X) = b_{i0} + b_{i1}X + \cdots + b_{i,n-k-1}X^{n-k-1}.$$

Como $b_i(X) + X^{n-k+i}$ para $i = 0, 1, \ldots, k-1$ são múltiplos de $g(X)$, então são polinômios códigos. Arranjando estes k polinômios códigos como linhas de uma matriz $k \times n$, obtém-se a matriz G na forma sistemática:

$$G = \begin{bmatrix} b_{00} & b_{01} & b_{02} & \cdots & b_{0,n-k-1} & 1 & 0 & \cdots & 0 \\ b_{10} & b_{11} & b_{12} & \cdots & b_{1,n-k-1} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ b_{k-1,0} & b_{k-1,1} & b_{k-1,2} & \cdots & b_{k-1,n-k-1} & 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{3.13}$$

A *matriz de paridade* correspondente será:

$$H = \begin{bmatrix} 1 & 0 & \cdots & 0 & b_{00} & b_{10} & b_{20} & \cdots & b_{k-1,0} \\ 0 & 1 & \cdots & 0 & b_{01} & b_{11} & b_{21} & \cdots & b_{k-1,1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & b_{0,n-k-1} & b_{1,n-k-1} & b_{2,n-k-1} & \cdots & b_{k-1,n-k-1} \end{bmatrix}. \tag{3.14}$$

Exemplo 3.8. Considere o código cíclico $(7, 4)$ com polinômio gerador $g(X) = 1 + X + X^3$. Dividindo X^3, X^4, X^5 e X^6 por $g(X)$ obtemos:

$$\begin{aligned} X^3 &= g(X) + (1 + X). \\ X^4 &= Xg(X) + (X + X^2). \\ X^5 &= (X^2 + 1)g(X) + (1 + X + X^2). \\ X^6 &= (X^3 + X + 1)g(X) + (1 + X^2). \end{aligned}$$

Daí, obtemos os polinômios códigos:

$$\begin{aligned} v_0(X) &= b_0(X) + X^3 = 1 + X + X^3. \\ v_1(X) &= b_1(X) + X^4 = X + X^2 + X^4. \\ v_2(X) &= b_2(X) + X^5 = 1 + X + X^2 + X^5. \\ v_3(X) &= b_3(X) + X^6 = 1 + X^2 + X^6. \end{aligned}$$

Logo, a matriz geradora na forma sistemática será:

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

e a matriz de verificação de paridade será:

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

3.3.2 Cálculo da Síndrome e Detecção de Erro

Seja $r = (r_0, r_1, \ldots, r_{n-1})$ o vetor recebido, vimos anteriormente que para um código linear determinar se um vetor recebido é um vetor código é preciso calcular a síndrome $s = r \cdot H^T$, onde H é a matriz de paridade.

Tome um código cíclico (n, k) na forma sistemática, a síndrome pode ser facilmente calculada. Considere o vetor recebido como um polinômio $r(X)$ com $\partial r(X) \leq n - 1$.

$$r(X) = r_0 + r_1 X + \cdots + r_{n-1} X^{n-1}.$$

Dividindo $r(X)$ por $g(X)$, obtemos:

$$r(X) = a(X)g(X) + s(X).$$

O resto será $s(X)$ com $\partial s(X) \leq n - k - 1$. Então os $n - k$ coeficientes de $s(X)$ formam a síndrome s. Pela Equação (3.6) temos que se $s(X) \equiv 0$ então $r(X)$ é um polinômio código.

Exemplo 3.9. Considere o código cíclico $(7, 4)$ gerado por $g(X) = 1 + X + X^3$ e $r(X) = x^2 + X^4 + X^5$.

$$
\begin{array}{ll}
X^5 + X^4 + \quad\; + X^2 & \;|\underline{X^3 + X + 1} \\
\underline{X^5 \quad\; + X^3 + X^2} & \quad X^2 + X + 1 \\
\quad X^4 + X^3 & \\
\underline{\quad X^4 \quad\; + X^2 + X} & \\
\quad X^3 + X^2 + X & \\
\underline{\quad\; X^3 \quad\; + X + 1} & \\
\quad\; X^2 \quad\; + 1 \quad \longrightarrow s(X) &
\end{array}
$$

$$r(X) = (1 + X + X^2)(1 + X + X^3) + (1 + X^2)$$

$$s(X) = 1 + X^2 \Rightarrow s = (1\ 0\ 1).$$

3.3.3 Decodificação de Códigos Cíclicos

A estrutura matemática bem definida, assim como as propriedades algébricas e geométricas dos códigos cíclicos são um fator importante na decodificação, pois utilizando corretamente essas propriedades é possível simplificar o circuito de decodificação.

O processo de decodificação de códigos cíclicos pode ser implementado utilizando os mesmos três passos da decodificação de códigos lineares, pois pela Definição 3.6 todo código cíclico é também um código linear. Assim, calcula-se a síndrome, verifica-se a qual padrão de erro esta síndrome está associada e é feita a correção de erro.

A estrutura cíclica de um código cíclico permite decodificar em série um vetor recebido

$$r(X) = r_0 + r_1 X + \cdots + r_{n-1} X^{n-1}.$$

Cada dígito recebido pode ser decodificado utilizando o mesmo circuito. Após calcular $s(X)$ o circuito decodificador verifica se essa síndrome corresponde a um dos padrões de erro

$$e(X) = e_0 + e_1 X + \cdots + e_{n-1} X^{n-1}$$

com um erro na posição de maior ordem, X^{n-1}, isto é, $e_{n-1} = 1$. Se $s(X)$ não corresponde a um padrão erro com $e_{n-1} = 1, r(X)$ e $s(X)$ são deslocados ciclicamente de uma posição simultaneamente. Obtemos

$$r^{(1)}(X) = r_{n-1} + r_0 X + \cdots + r_{n-2} X^{n-1}.$$

Agora, o mesmo circuito decodificador irá verificar se $s^{(1)}(X)$ correspondente a um padrão de erro na posição X^{n-1}.

Se a síndrome $s(X)$ de $r(X)$ corresponde a um padrão de erro com um erro na posição X^{n-1}, isto é, $e_{n-1} = 1$, o dígito r_{n-1} está errado e pode ser corrigido através da soma $r_{n-1} + e_{n-1}$. Após a correção obtemos:

$$r_1(X) = r_0 + r_1 X + \cdots + r_{n-2} X^{n-2} + (r_{n-1} + e_{n-1}) X^{n-1}.$$

Assim, o efeito do dígito errado na mensagem é retirado da síndrome $s(X)$ e teremos

$$s_1(X) = X^{n-1} + s(X).$$

Agora, deslocando-se ciclicamente e simultaneamente $r_1(X)$ e $s_1(X)$ de uma posição obtemos:

$$r_1^{(1)}(X) = (r_{n-1} + e_{n-1}) r_0 X + r_1 X^2 + \cdots + r_{n-2} X^{n-1}.$$

A síndrome $s_1^{(1)}(X)$ de $r_1^{(1)}(X)$ será o resto da divisão de $X[s(X) + X^{n-1}]$ por $g(X)$. Como os restos da divisão de $Xs(X)$ e X^n por $g(X)$ são, respectivamente, $s^{(1)}(X)$ e 1, tem-se

$$s_1^{(1)}(X) = s^{(1)}(X) + 1.$$

O processo continua com decodificação dos demais dígitos recebidos $r_{n-2}, r_{n-3}, \ldots, r_0$ que é feita de modo análogo a decodificação de r_{n-1}. Quando um erro é detectado e corrigido, o seu efeito na síndrome é removido. O decodificador para após n deslocamentos e se no fim do processo não tivermos $s(X) \equiv 0$, então existe um padrão de erro não corrigível.

De acordo com Hefez e Villela [9] o fato de muitas características dos códigos cíclicos serem traduzidas por operações sobre polinômios em uma indeterminada torna os algoritmos de codificação e decodificação desses códigos bastante eficientes. No entanto, a determinação da distância mínima desses códigos é complexa e isso os torna pouco úteis na prática.

No próximo capítulo veremos que é possível construir uma subclasse de códigos cíclicos para os quais podemos obter a priori cotas inferiores para as distâncias mínimas.

CAPÍTULO 4

CÓDIGOS BCH

Durante os anos 50 foram encontrados muitos resultados importantes da Teoria da informação e da Teoria da codificação. Nessa época, surgiram os primeiros códigos de bloco desenvolvidos por Hamming em 1950 [3]. Desse período até 1957 as técnicas de codificação de códigos de bloco então existentes envolviam operações com matrizes e vetores e recorriam a tabelas de decodificação. Em 1957, Eugene Prange [4] publicou um trabalho onde apresentou uma estrutura algébrica útil para os codificadores de bloco, mostrando a estrutura cíclica de alguns códigos de bloco já existentes, e surgem assim os códigos cíclicos.

Conforme vimos na Seção 3.2 a distância mínima de um código determina quantos erros podem ser detectados e corrigidos no decodificador. Além disso, verificamos no capítulo anterior que os códigos cíclicos constituem uma importante classe de códigos lineares por apresentarem uma estrutura algébrica especial que permite uma caracterização polinomial dos elementos do código. Este fato simplifica os processos de codificação e decodificação, pois proporciona o desenvolvimento de algoritmos computacionais eficientes para este fim.

Contudo, a determinação da distância mínima para os códigos cíclicos não é uma tarefa simples. Nesse sentido, em 1959, Hocquenghem [5] e, de modo independente, Bose e Chaudhuri [6], em

1960, desenvolveram os códigos BCH binários através de generalizações dos códigos de Hamming.

Os códigos BCH constituem uma família de códigos cíclicos definidos através de um conjunto conveniente de raízes de seus polinômios geradores, que faz com que esses códigos possuam uma cota inferior obtida a priori para suas distâncias mínimas. Tais códigos são uma generalização dos códigos de Hamming capazes de corrigir erros múltiplos.

A estrutura cíclica desses códigos foi demonstrada por Peterson em 1960 [12], que desenvolveu o primeiro algortimo de decodificação para os códigos BCH binários. Em 1961, Gorenstein e Zieler [13] generalizaram os códigos BCH binários para códigos de p^m símbolos (p primo) e aperfeiçoaram o algoritmo descoberto por Peterson em 1960 [12].

Em 1965, Berlekamp [8] desenvolveu um algoritmo iterativo que tornou mais eficiente o algoritmo de Peterson. Mais tarde, Massey [14, 15] aprimorou o algoritmo de Berlekamp, que passou a ser conhecido como algoritmo iterativo de Berlekamp-Massey. Desde então, foram desenvolvidos vários algoritmos que aperfeiçoaram o algoritmo de Peterson, no entanto, os mais eficientes são o algoritmo iterativo de Berlekamp-Massey e o algoritmo desenvolvido por Chien [16]. Portanto, tais algoritmos serão apresentados neste trabalho a fim de compreender a decodificação para códigos BCH binários.

4.1 Códigos BCH Binários

Os códigos BCH binários formam uma subclasse importante de códigos cíclicos cujos elementos pertencem ao corpo de Galois $GF(2^m)$ e são representados por polinômios geradores sobre $GF(2)$. Neste trabalho trataremos apenas dos códigos BCH binários, pois a extensão para códigos não binários é considerada um processo simples e direto onde basta substituir $GF(2)$ por $GF(q)$ e $GF(2^m)$ por $GF(q^m)$.

Dado um número qualquer inteiro $m \geq 3$ e $t < 2^{m-1}$, existe um código BCH(n, k, d) capaz de

corrigir t erros em um bloco com $n = 2^m - 1$ dígitos com os seguintes parâmetros:

Comprimento do bloco: $n = 2^m - 1$.

Número de dígitos de paridade: $n - k \leq mt$.

Distância mínima: $d_{\min} \geq 2t + 1$.

Considerando um corpo finito K, mostramos no capítulo anterior que é possível construir códigos cíclicos $C \subset K^n$ com polinômios geradores $g(X)$ divisores de $X^n - 1$. Agora veremos que é possível descrevermos um código através das raízes de $g(X)$ em alguma extensão do corpo de base K, que são as chamadas *raízes da unidade*.

Uma raiz n-ésima da unidade num corpo F é uma raiz em F do polinômio $X^n - 1$. No entanto, nem sempre é possível fatorar o polinômio $X^n - 1$ como o produto de fatores lineares em $F[X]$.

Teorema 4.1. Seja K um corpo finito com q elementos e, n, um inteiro primo com q. Então, existem uma extensão F de K e um elemento $\gamma \in F$ tais que

$$X^n - 1 = (X - \gamma^0)(X - \gamma^1)(X - \gamma^2) \cdots (X - \gamma^{n-1}),$$

com $\gamma^0, \gamma^1, \cdots, \gamma^{n-1}$ dois a dois distintos.

Tomando o menor m tal que $q^m = 1$, então o corpo $GF(q^m)$ será o menor corpo onde $X^n - 1$ se fatora como produto de fatores lineares. Nesse caso, $GF(q^m)$ é o corpo das raízes de $X^n - 1$. Portanto, um elemento primitivo de $GF(q^m)$ gera todas as raízes n-ésimas da unidade.

Uma raiz n-ésima da unidade que não é raiz m-ésima da unidade para nenhum $m < n$ será chamada raiz n-ésima *primitiva* da unidade. No teorema acima, se n e q são primos entre si, então γ é uma raiz n-ésima primitiva da unidade.

4.1.1 Códigos Cíclicos Definidos por Anulamento

Os códigos cíclicos podem ser definidos por condições de anulamento. Essa caracterização é de grande importância pois facilita a determinação da sua matriz teste de paridade.

Dado um código cíclico $C \subset K^n$, onde $K = GF(q)$ e n e q são primos entre si. Seja T um isomorfismo dado por:

$$T : K^n \longrightarrow R_n$$

$$(a_0, \cdots, a_{n-1}) \longmapsto [a_0 + a_1 X + \cdots + a_{n-1} X^{n-1}]$$

Pelo Teorema 3.2 o código cíclico C pode ser visto como um ideal $T(C) = I([g(X)])$ no anel $R_n = K[X]_{(X^n-1)}$, onde $g(X) \in K[X]$ é um divisor de $X^n - 1$. Seja F um corpo finito que contém K, sobre o qual o polinômio $X^n - 1$ se fatora em fatores lineares mônicos distintos. Sejam $\theta_1, \cdots, \theta_r$ as raízes de $g(X)$ em F, que são portanto duas a duas distintas.

Proposição 4.1. Com as notações e condições acima, temos que:

$$T(C) = I([g(X)]) = \{[f(X)] \in R_n; f(\theta_1) = \cdots = f(\theta_r) = 0\}.$$

Demonstração: Note que $[f(X)] \in I([f(X)])$ se, e somente se, existe $c(X)$ em $K[X]$ tal que $[f(X)] = [c(X)][g(X)]$, o que é equivalente a dizer que existe $d(X) \in K[X]$ tal que $f(X) = d(X)g(X)$. Sabendo que $g(X)$ divide $X^n - 1$ e que $\theta_1, \cdots, \theta_r$ são as raízes de $g(X)$ em F, então $f(\theta_i) = 0, \forall i = 1, \cdots, r$. ∎

A partir desse resultado podemos contruir códigos cíclicos com um conjunto de raízes n-ésimas da unidade, de modo que teremos uma cota inferior para a sua distância mínima. Essa subclasse de códigos cíclicos é chamada de BCH.

Teorema 4.2. Seja $K = GF(q)$ e, n, um inteiro maior do que 1 e primo com q. Seja $F = GF(q^m)$ um corpo onde $X^n - 1$ se decompõe em fatores lineares, e seja $\gamma \in F$ uma raiz n-ésima primitiva da unidade. Seja $\phi_i(X)$ o polinômio mínimo de γ^i e C um código cíclico com polinômio gerador

$$g(X) = mmc\{\phi_1(X), \phi_2(X), \ldots, \phi_{\delta-1}(X)\}.$$

com $\delta \leq n$. Então, a distância mínima de C é pelo menos δ e sua dimensão é pelo menos $n - m(\delta - 1)$, onde $m = dim_K F$.

Portanto, tomando $\delta - 1 = 2t$, definiremos agora os códigos BCH binários da seguinte forma.

Definição 4.1. Seja α um elemento primitivo em $GF(2^m)$ e seja $\phi_i(X)$ o polinômio mínimo de α^i. Então o código BCH de distância designada δ é um código cíclico gerado pelo polinômio de menor grau $g(X)$ sobre $GF(2)$ cujas raízes são

$$\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}. \tag{4.1}$$

onde $\delta - 1 = 2t$.

Assim, $g(X)$ deve ser o mínimo múltiplo comum (MMC) de $\phi_1(X), \phi_2(X), \ldots, \phi_{2t}(X)$, isto é,

$$g(X) = mmc\{\phi_1(X), \phi_2(X), \ldots, \phi_{2t}(X)\}. \tag{4.2}$$

Se $i \in \mathbb{Z}$ é par, pode ser expresso como um produto da seguinte forma:

$$i = j \cdot 2^l,$$

onde j é um número ímpar e $l \geq 1$. Então, $\alpha^i = (\alpha^j)^{2^l}$ é o conjugado de α^j, e portanto, α^i e α^j têm o mesmo polinômio mínimo, isto é,

$$\phi_i(X) = \phi_j(X).$$

Portanto, qualquer potência par de α na sequência (4.2) tem o mesmo polinômio de alguma potência ímpar de α na mesma sequência. Assim, podemos simplificar a Equação (4.2), ou seja,

$$g(X) = mmc\{\phi_1(X), \phi_3(X), \ldots, \phi_{2t-1}(X)\}. \tag{4.3}$$

Como cada polinômio $\phi_i(X)$ tem $\partial\phi_i(X) \leq m$, então $\partial g(X) \leq mt$.

Exemplo 4.1. Seja α um elemento primitivo do corpo de Galois $GF(2^4)$ dado pela Tabela 3.1 tem-se

$n = 2^m - 1 \Rightarrow n = 2^4 - 1 = 15$. Os polinômios de $\alpha, \alpha^3, \alpha^5$ são respectivamente,

$$\begin{aligned} \phi_1(X) &= 1 + X + X^4. \\ \phi_3(X) &= 1 + X + X^2 + X^3 + X^4. \\ \phi_5(X) &= 1 + X + X^2. \end{aligned}$$

Da Equação (4.3) concluímos que o código corretor de dois erros é gerado por

$$\begin{aligned} g(X) &= mmc\{\phi_1(X), \phi_3(X)\}. \\ g(X) &= 1 + X^4 + X^6 + X^7 + X^8. \end{aligned}$$

E para o código BCH corretor de 3 erros temos:

$$\begin{aligned} g(X) &= mmc\{\phi_1(X), \phi_3(X), \phi_5(X)\}. \\ g(X) &= 1 + X + X^2 + X^4 + X^5 + X^8 + X^{10}. \end{aligned}$$

Pela definição de um código BCH corretor de $t-$erros e comprimento $n = 2^m - 1$, temos que cada polinômio código $v(X)$ tem $\alpha, \alpha^2, \ldots, \alpha^{2t}$ e seus conjugados como raízes. Portanto, podemos definir estes códigos BCH da seguinte maneira.

Dado $v = (v_0, v_1, \ldots, v_{n-1}) \in GF(2)$, v será uma palavra código se, e somente se, o polinômio

$$v(X) = v_0 + v_1 X + \cdots + v_{n-1} X^{n-1}$$

tem $\alpha, \alpha^2, \ldots, \alpha^{2t}$ como raízes. Uma vez que α^i é uma raiz de $v(X)$ para $1 \leq i \leq 2t$, então $v(\alpha^i) = 0$, isto é,

$$v(\alpha^i) = v_0 + v_1\alpha^i + v_2\alpha^{2i} + \cdots + v_{n-1}\alpha^{(n-1)i} = 0. \tag{4.4}$$

Portanto, a matriz verificação de paridade será da seguinte forma:

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & (\alpha^2)^3 & \cdots & (\alpha^2)^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (\alpha^{2t}) & (\alpha^{2t})^2 & (\alpha^{2t})^3 & \cdots & (\alpha^{2t})^{n-1} \end{bmatrix}. \tag{4.5}$$

Se para algum i e j, α^i é o conjugado de α^j, então $v(\alpha^i) = 0 \Leftrightarrow v(\alpha^j) = 0$. Portanto, podemos omitir a $j-$ésima linha de H e assim reduziremos H:

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & (\alpha^3)^3 & \cdots & (\alpha^3)^{n-1} \\ 1 & (\alpha^5) & (\alpha^5)^2 & (\alpha^5)^3 & \cdots & (\alpha^5)^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (\alpha^{2t-1}) & (\alpha^{2t-1})^2 & (\alpha^{2t-1})^3 & \cdots & (\alpha^{2t-1})^{n-1} \end{bmatrix}. \tag{4.6}$$

Conforme vimos no Capítulo 3, se $v = (v_0, v_1, \ldots, v_{n-1})$ é uma palavra do código BCH corretor de t erros, então temos

$$v \cdot H^T = 0.$$

Portanto, podemos observar que estamos reduzindo o número de entradas da matriz H pela metade. Isso implica diretamente na redução do custo computacional da decodificação, pois uma matriz de paridade com um número menor de entradas reduz o tempo de processamento do algoritmo de decodificação.

Exemplo 4.2. Considere um código BCH com $n = 15$, com capacidade para corrigir dois erros. Logo, tem-se $t = 2$ e $m = 4$.

Sabendo que $n - k \leq mt$, temos $-k \leq mt - n \Rightarrow k \geq n - mt \Rightarrow k \geq 15 - 4 \cdot 2 \Rightarrow k \geq 7$. Tomando $k = 7$ obtemos o código $\mathcal{C}(15, 7)$.

Seja α um elemento primitivo de $GF(2^4)$. Então a matriz de verificação de paridade desse código é:

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} & \alpha^{21} & \alpha^{24} & \alpha^{27} & \alpha^{30} & \alpha^{33} & \alpha^{36} & \alpha^{39} & \alpha^{42} \end{bmatrix}.$$

Pela Tabela 3.1 e como $\alpha^{15} = 1$ podemos reescrever cada entrada da matriz substituindo por seu correspondente $4-$upla:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

4.2 Decodificação de Códigos BCH

Como vimos no Capítulo 3 dada uma mensagem recebida $r(X) = r_0 + r_1X + \cdots + r_{n-1}X^{n-1}$, devido aos ruídos do canal, a mensagem poderá conter erros e teremos:

$$r(X) = v(X) + e(X)$$

onde $v(X)$ é o polinômio código e $e(X)$ o padrão erro. Sabemos que para a decodificação é preciso calcular, primeiramente, a síndrome do padrão erro.

Considere um código BCH (primitivo) corretor de t erros, a síndrome será uma $2t-$upla:

$$S = (S_1, S_2, \ldots, S_{2t}) = r \cdot H^T$$

onde H é a matriz de verificação de paridade, e teremos:

$$S_i = r(\alpha^i) = r_0 + r_1\alpha^i + \cdots + r_{n-1}\alpha^{(n-1)i}$$

para $1 \leq i \leq 2t$.

Para calcular as síndromes dividimos $r(X)$ pelo polinômio mínimo $\phi_i(X)$ de α^i e obtemos:

$$r(X) = a_i(X)\phi_i(X) + b_i(X)$$

onde $b_i(X)$ é o resto e $\partial b_i(X) < \partial \phi_i(X)$. Como $\phi_i(\alpha^i) = 0$ tem-se:

$$S_i = r(\alpha^i) = b_i(\alpha^i). \tag{4.7}$$

Exemplo 4.3. Considere o código BCH corretor de dois erros dado no Exemplo 4.2. Suponha que foi recebido $r(X) = 1 + X^8$. Teremos $2t = 4 \Rightarrow S = (S_1, S_2, S_3, S_4) = r \cdot H^T$.

Os polinômios mínimos de $\alpha, \alpha^2, \alpha^4$ são idênticos

$$\phi_1(X) = \phi_2(X) = \phi_4(X) = 1 + X + X^4.$$

O polinômio mínimo de α^3 é

$$\phi_3(X) = 1 + X + X^2 + X^3 + X^4.$$

Dividindo $r(X)$ por $\phi_1(X)$ e $\phi_3(X)$ obtemos respectivamente

$$b_1(X) = X^2 \quad \text{e} \quad b_3(X) = 1 + X^3.$$

Usando a Equação (4.7), a Tabela 3.1 e substituindo α, α^2 e α^4 em $b_1(X)$, e α^3 em $b_3(X)$ obtemos

$$S_1 = \alpha^2,\ S_2 = \alpha^4,\ S_4 = \alpha^8 \text{ e } S_3 = \alpha^7.$$

Então,

$$S = (\alpha^2, \alpha^4, \alpha^7, \alpha^8).$$

Mostramos no Capítulo 3 que a síndrome calculada a partir do vetor recebido r, na realidade depende apenas do padrão de erro e não da palavra código transmitida v.

Como $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$ são as raízes de cada polinômio código, então $v(\alpha^i) = 0$, para $1 \leq i \leq 2t$. Da Equação (4.7) obtemos:

$$S_i = e(\alpha^i) \tag{4.8}$$

para $1 \leq i \leq 2t$.

Suponhamos que o padrão de erro $e(X)$ tem s erros nas posições $X^{j_1}, X^{j_2}, \ldots, X^{j_s}$, isto é,

$$e(X) = X^{j_1} + X^{j_2} + \cdots + X^{j_s} \tag{4.9}$$

onde $0 \leq j_1 < j_2 < \ldots < j_s < n$. Das Equações (4.8) e (4.9) obtemos:

$$\begin{aligned} S_1 &= \alpha^{j_1} + \alpha^{j_2} + \cdots + \alpha^{j_s} \\ S_2 &= (\alpha^{j_1})^2 + (\alpha^{j_2})^2 + \cdots + (\alpha^{j_s})^2 \\ &\vdots \qquad\quad \vdots \\ S_{2t} &= (\alpha^{j_1})^{2t} + (\alpha^{j_2})^{2t} + \cdots + (\alpha^{j_s})^{2t} \end{aligned} \tag{4.10}$$

onde $\alpha^{j_1}, \alpha^{j_2}, \cdots, \alpha^{j_s}$ são desconhecidos.

Note que encontrando-se um método para resolver estas equações teremos um algoritmo de decodificação dos códigos BCH. Uma vez que encontrarmos $\alpha^{j_1}, \alpha^{j_2}, \cdots, \alpha^{j_s}$ teremos as potências $j_1, j_2, \ldots, j_s$ e saberemos a localização dos erros em $e(X)$.

O sistema de equações (4.10) terá várias soluções possíveis, e cada solução conduz a um padrão de erro diferente, e a solução certa será a solução que conduzir ao padrão de erro com menor número de erros. Em seguida, veremos um procedimento eficiente para determinação de α^{j_l} (com $l = 1, 2, \ldots, s$) a partir das componentes S_i da síndrome. Tome $\beta_l = \alpha^{j_l}, 1 \leq l \leq s$. Chamaremos estes elementos de *números localizadores de erros*, pois eles mostram as posições do erro. Reescrevendo as Equações (4.10), obtemos:

$$\begin{aligned} S_1 &= \beta_1 + \beta_2 + \cdots + \beta_s \\ S_2 &= \beta_1^2 + \beta_2^2 + \cdots + \beta_s^2 \\ &\vdots \qquad\quad \vdots \\ S_{2t} &= \beta_1^{2t} + \beta_2^{2t} + \cdots + \beta_s^{2t} \end{aligned} \tag{4.11}$$

As Equações (4.11) são funções simétricas em $\beta_1, \beta_2, \ldots, \beta_s$, conhecidas como funções simétricas da soma de potências. Para achar as soluções de (4.11) defina o seguinte *polinômio localizador do*

erro [1]:

$$\sigma(X) = (1 + \beta_1 X)(1 + \beta_2 X) \cdots (1 + \beta_s X)$$

$$\sigma(X) = \sigma_0 + \sigma_1 X + \sigma_2 X^2 + \cdots + \sigma_s X^s.$$

As raízes de $\sigma(X)$ são $\beta_1^{-1}, \beta_2^{-1}, \ldots, \beta_s^{-1}$, que são os inversos dos números localizadores de erros. E existe uma relação entre os coeficientes de $\sigma(X)$ e os números localizadores de erro dada pelas equações:

$$\begin{aligned} \sigma_0 &= 1 \\ \sigma_1 &= \beta_1 + \beta_2 + \ldots + \beta_s \\ \sigma_2 &= \beta_1 \cdot \beta_2 + \beta_2 \cdot \beta_3 + \ldots + \beta_{s-1} \cdot \beta_s \\ &\vdots \qquad \vdots \\ \sigma_s &= \beta_1 \cdot \beta_2 \cdot \ldots \cdot \beta_s. \end{aligned} \tag{4.12}$$

As σ_i são ditas *funções simétricas elementares* de β_i. Note que há ainda uma relação entre as σ_i e S_i dadas pelas igualdades de Newton:

$$\begin{aligned} S_1 + \sigma_1 &= 0 \\ S_2 + \sigma_1 \cdot S_1 + 2\sigma_2 &= 0 \\ S_3 + \sigma_1 \cdot S_2 + \sigma_2 \cdot S_1 + 3\sigma_3 &= 0 \\ &\vdots \qquad \vdots \\ S_s + \sigma_1 \cdot S_{s-1} + \cdots + \sigma_{s-1} \cdot S_1 + s \cdot \sigma_s &= 0 \\ S_{s+1} + \sigma_1 \cdot S_s + \cdots + \sigma_{s-1} \cdot S_2 + \sigma_s \cdot S_1 + (s+1) \cdot \sigma_{s+1} &= 0. \end{aligned} \tag{4.13}$$

No caso binário, temos $1 + 1 = 2 = 0$, então:

$$i\sigma_i = \begin{cases} \sigma_i, & \text{para } i \text{ ímpar} \\ 0, & \text{para } i \text{ par.} \end{cases}$$

Os números localizadores de erro $\beta_1, \beta_2, \ldots, \beta_s$ podem ser encontrados determinando as raízes de $\sigma(X)$ se for possível determinar as funções elementares $\sigma_1, \sigma_2, \ldots, \sigma_s$ a partir das Equações (4.13).

As Equações (4.13) têm muitas soluções possíveis, no entanto, estamos interessados na solução de grau mínimo de erros. Se $s \leq t$, $\sigma(X)$ dará o padrão de erro atual. Resumidamente, podemos dizer que o processo de correção de erros dos códigos BCH é descrito pelos três passos:

1. Calcular a síndrome $S = (S_1, S_2, \ldots, S_{2t})$ a partir do polinômio recebido $r(X)$.
2. Determinar o polinômio localizador de erro $\sigma(X)$ a partir da síndrome S.
3. Determinar os números localizadores de erro $\beta_1, \beta_2, \ldots, \beta_s$ a partir das raízes de $\sigma(X)$ e corrigir os erros em $r(X)$.

Esses três passos formam o algoritmo desenvolvido por Peterson [12]. Os passos 1 e 3 são relativamentes simples, enquanto o passo 2 é a parte mais complexa. Essa complexidade é diretamente proporcional ao aumento da capacidade de correção de erros. A fim de simplicar o passo 2 Berlekamp [8] desenvolveu um algoritmo iterativo para encontrar o polinômio localizador de erro, o qual apresentaremos na próxima seção.

4.2.1 Algoritmo iterativo para encontrar o polinômio localizador do erro $\sigma(X)$

O algoritmo iterativo de Berlekamp para encontrar o polinômio localizador do erro será apresentado a seguir sem demonstração formal. Contudo, a demonstração do algoritmo e mais detalhes podem ser consultados em Berlekamp [8].

Primeiramente encontra-se o polinômio de menor grau $\sigma^{(1)}(X)$ cujos coeficientes satisfazem a primeira igualdade de Newton em (4.13). O passo seguinte será testar se os coeficientes de $\sigma^{(1)}(X)$ satisfazem a segunda igualdade de Newton em (4.13), se isso ocorrer tomamos $\sigma^{(2)}(X) = \sigma^{(1)}(X)$. Caso contrário, soma-se a $\sigma^{(1)}(X)$ um termo corretivo a fim de obter $\sigma^{(2)}(X)$ com grau mínimo e

cujos coeficientes satisfaçam as duas primeiras igualdades de Newton em (4.13).

A iteração contínua até se obter $\sigma^{(2t)}(X)$, então toma-se $\sigma(X) = \sigma^{(2t)}(X)$. Este $\sigma(X)$ conduzirá a um padrão de erro $e(X)$, com peso mínimo, que satisfaz as Equações (4.11). Assim, se o número de erros do polinômio recebido $r(X)$ for menor ou igual a t, então $\sigma(X)$ produzirá o padrão de erro verdadeiro.

Para executar a iteração a fim de encontrar $\sigma(X)$ preenchemos sucessivamente a Tabela 4.1, onde $\sigma^{(\mu)}(X)$ é o polinômio de grau mínimo obtido no $\mu-$ésimo passo da iteração, l_μ é o grau de $\sigma^{(\mu)}(X)$ e d_μ é chamada a $\mu-$ésima *discrepância* que definimos a seguir.

Seja o polinômio de grau mínimo determinado no $\mu-$ésimo passo da iteração dado por:

$$\sigma^{(\mu)}(X) = 1 + \sigma_1^{(\mu)} X + \sigma_2^{(\mu)} X^2 + \cdots + \sigma_{l_\mu}^{(\mu)} X^{l_\mu}$$

e cujos coeficientes satisfazem as μ primeiras igualdades de Newton em (4.13), define-se d_μ por

$$d_\mu = S_{\mu+1} + \sigma_1^{(\mu)} S_\mu + \sigma_2^{(\mu)} S_{\mu-1} + \cdots + \sigma_{l_\mu}^{(\mu)} S_{\mu+1-l_\mu}$$

Tabela 4.1: Passos do algoritmo de Berlekamp para um código BCH binário

μ	$\sigma^{(\mu)}(X)$	d_μ	l_μ	$\mu - l_\mu$
-1	1	1	0	-1
0	1	S_1	0	0
1				
2				
$\vdots$				
$2t$				

Com a Tabela 4.1 preenchida até a linha μ podemos construir a linha $\mu + 1$ da seguinte forma:

1. Se $d_\mu = 0$, então $\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X)$.

2. Se $d_\mu \neq 0$, devemos encontrar uma linha ρ, anterior à linha μ tal que $d_\rho \neq 0$ e o valor $\rho - l_\rho$ tenha o maior valor possível. Então $\sigma^{(\mu+1)}(X)$ será dado por

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X) + d_\mu d_\rho^{-1} X^{\mu-\rho} \sigma^{(\rho)}(X) \tag{4.14}$$

que é o polinômio de grau mínimo cujos coeficientes satisfazem as primeiras $\mu + 1$ igualdades de Newton, e

$$l_{\mu+1} = \max(l_\mu, l_\rho + \mu - \rho).$$

Exemplo 4.4. Seja α um elemento primitivo do corpo de Galois $GF(2^4)$ dado pela Tabela 3.1 tal que $1 + \alpha + \alpha^4 = 0$.

Considere o código BCH corretor de dois erros $(15, 7)$ com distância mínima $d_{\min} \geq 5$. Então o polinômio gerador deste código terá peso de Hamming igual a 5 e conforme Exemplo 4.1 é dado por:

$$g(X) = 1 + X^4 + X^6 + X^7 + X^8.$$

Assuma que a palavra código transmitida é

$$v = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

e o vetor recebido é

$$r = (0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0).$$

Então, $r(X) = X^3 + X^5$. E pelo Teorema 2.7 podemos determinar o polinômio minimal dos elementos α, α^2 e α^3:

$$\phi_1(X) = \phi_2(X) = \phi_4(X) = 1 + X + X^4$$

e

$$\phi_3(X) = 1 + X + X^2 + X^3 + X^4.$$

Dividindo $r(X)$ por $\phi_1(X), \phi_2(X)$ e $\phi_3(X)$, respectivamente, obtemos os restos:

$$b_1(X) = b_2(X) = b_4(X) = X + X^2 + X^3$$

e

$$b_3(X) = X + X^2 + X^4.$$

Pela Equação 4.7 obtemos as componentes da síndrome:

$$\begin{aligned} S_1 &= b_1(\alpha) = \alpha^{11}. \\ S_2 &= b_2(\alpha^2) = \alpha^7. \\ S_3 &= b_3(\alpha^3) = \alpha^7. \\ S_4 &= b_4(\alpha^4) = \alpha^{14}. \end{aligned}$$

Agora, usando o algoritmo iterativo descrito anteriormente vamos determinar o polinômio localizador do erro.

Seja $d_0 = S_1 = \alpha^{11} \neq 0$, tome $\rho = -1$, pela Tabela 4.1 temos $\sigma^{(0)}(X) = \sigma^{(-1)}(X) = 1$ e $d_{-1} = 1$. Logo, $\sigma^{(1)}(X) = \sigma^{(0)}(X) + d_0 d_{-1}^{-1} X \sigma^{(-1)}(X)$. Então,

$$\sigma^{(-1)}(X) = 1 + \alpha^{11} X.$$

Seja $d_1 = S_2 + \sigma_1^{(1)} S_1 = \alpha^7 + \alpha^{11}\alpha^{11} = 2\alpha^7 = 0 \Rightarrow \sigma^{(2)}(X) = \sigma^{(1)}(X) = 1 + \alpha^{11} X$. Seja $d_2 = S_3 + \sigma_1^{(2)} S_2 + \sigma_2^{(2)} S_1 = \alpha^7 + \alpha^{11}\alpha^7 + 0 \Rightarrow d_2 = \alpha^4 \neq 0$. Tome $\rho = 0$. Logo,

$$\begin{aligned} \sigma^{(3)}(X) &= \sigma^{(2)}(X) + d_2 d_0^{-1} X^2 \sigma^{(0)}(X) \\ \sigma^{(3)}(X) &= 1 + \alpha^{11} x + \alpha^8 X^2. \end{aligned}$$

Seja $d_3 = S_4 + \sigma_1^{(2)} S_3 + \sigma_2^{(2)} S_2 + \sigma_3^{(2)} S_1 = \alpha^{14} + \alpha^{11}\alpha^7 + \alpha^8\alpha^7 = 2\alpha^{15} = 0 \Rightarrow \sigma^{(4)}(X) = \sigma^{(3)}(X)$. Logo, $\sigma(X) = \sigma^{(4)}(X) = 1 + \alpha^{11} X + \alpha^8 X^2$. Assim, podemos preencher a Tabela 4.2.

Tabela 4.2: Passos do algoritmo de Berlekamp para o código BCH binário do Exemplo 4.4.

μ	$\sigma^{(\mu)}(X)$	d_μ	l_μ	$\mu - l_\mu$
-1	1	1	0	-1
0	1	α^{11}	0	0
1	$1 + \alpha^{11} X$	0	1	0 (tome $\rho = -1$)
2	$1 + \alpha^{11} X$	α^4	1	1
3	$1 + \alpha^{11} X + \alpha^8 X^2$	0	2	1 (tome $\rho = 0$)
4	$1 + \alpha^{11} X + \alpha^8 X^2$	-	-	-

Note que α^{10} e α^{12} são as raízes de $\sigma(X)$.

$$\sigma(\alpha^{10}) = 1 + \alpha^{11}\alpha^{10} + \alpha^{8}\alpha^{20} = 1 + \alpha^{6} + \alpha^{13} = 1 + \alpha^{6}\alpha^{Z(7)} = 1 + \alpha^{6}\alpha^{9} = 1 + \alpha^{15} = 0$$

e

$$\sigma(\alpha^{12}) = 1 + \alpha^{11}\alpha^{12} + \alpha^{2}\alpha^{24} = 1 + \alpha^{8} + \alpha^{2} = 1 + \alpha^{2}\alpha^{Z(6)} = 1 + \alpha^{2}\alpha^{13} = 1 + \alpha^{15} = 0.$$

Portanto, obtemos os números localizadores do erro α^5 e α^3, que são os inversos das raízes de $\sigma(X)$. Logo,

$$e(X) = X^3 + X^5.$$

Para obter a palavra transmitida basta somar o polinômio recebido ao padrão do erro $e(X)$:

$$\begin{aligned} v(X) &= r(X) + e(X) \\ v(X) &= (X^3 + X^5) + (X^3 + X^5) \\ v(X) &= 2X^3 + 2X^5 = 0 + 0 = 0 \\ v(X) &\equiv 0. \end{aligned}$$

4.2.2 Método de Chien para determinação dos números localizadores do erro

Determinar os números localizadores do erro é a última etapa da detecção e correção de erros para códigos BCH e, como vimos anteriormente, tais números são exatamente os inversos das raízes de $\sigma(X)$.

Peterson [12] utilizava o método da substituição para encontrar as raízes de $\sigma(X)$ e depois calculava seus inversos. Nesse método testamos cada um dos valores de 1, $\alpha, \alpha^2, \ldots, \alpha^{n-1}$ substituindo no polinômio $\sigma(X)$ para encontrar suas raízes. Contudo, esse processo pode se tornar demorado e, a fim de simplificá-lo, em 1964, Chien [16] apresentou um outro método para determinação dos números localizadores do erro que descreveremos em seguida.

Primeiramente são decodificados os bits de maior ordem, portanto, começamos com r_{n-1}.

Verificamos se α^{n-1} é um número localizador do erro testando se seu inverso α é uma raiz de $\sigma(X)$.

$$\alpha \text{ é raiz de } \sigma(X) \Leftrightarrow 1 + \sigma_1\alpha + \sigma_2\alpha^2 + \cdots + \sigma_v\alpha^v = 0$$

Assim, se α é raiz de $\sigma(X)$, então α^{n-1} é um número localizador do erro e, consequentemente, o dígito r_{n-1} está errado.

Em geral, para decodificar r_{n-l} verificamos se α^{n-l} é um número localizador do erro. Para isto, testamos se seu inverso é raiz de $\sigma(X)$. Sabendo que $\alpha^n = 1$ temos que o inverso de α^{n-l} será $\frac{1}{\alpha^{n-l}} = \frac{\alpha^n}{\alpha^{n-l}} = \alpha^{n-n+l} = \alpha^l$. Portanto, devemos testar se α^l é raiz de $\sigma(X)$.

$$\alpha^l \text{ é raiz de } \sigma(X) \Leftrightarrow 1 + \sigma_1\alpha^l + \sigma_2\alpha^{2l} + \cdots + \sigma_v\alpha^{vl} = 0$$

onde $l = 1, 2, \cdots, n-1$. Assim, se α^l é raiz de $\sigma(X)$, então α^{n-l} é um número localizador do erro e, consequentemente, o dígito r_{n-l} está errado.

4.2.3 Desempenho dos Códigos BCH

Em um sistema de comunicação, ao transmitir uma mensagem por um canal de transmissão precisamos garantir que tal mensagem poderá ser recuperada com qualidade no destino. No entanto, esta mensagem está sujeita à presença de ruídos no canal.

O ruído consiste na alteração de alguma das características do sinal transmitido por efeito de um outro sinal exterior ao sistema de transmissão, ou gerado pelo próprio sistema de transmissão. Estes sinais indesejados são de natureza aleatória e, portanto, não é possível prever o seu valor num instante de tempo futuro.

A fim de analisar o desempenho de um codificador de canal é necessário comparar a relação Sinal-Ruído (SNR-*signal to noise ratio*) com a Taxa de Erro de Bit (BER-*bit error rate*). A SNR é a razão entre o nível máximo de amplitude do sinal transmitido e o nível de amplitude do ruído do sistema. E a BER é a razão entre o número de bits errados e o número de bits transmitidos. A

SNR e a BER estão diretamente relacionados, pois quanto menor o nível de ruído menores serão as distorções e, consequentemente, menores serão as proporções de bits com erro.

O ruído pode ser aditivo, ou seja, soma-se ao sinal, ou multiplicativo, onde o sinal resultante é o produto do sinal transmitido pelo ruído. Uma vez que o ruído é um processo aleatório suas características podem ser descritas através da função densidade de probabilidade da sua amplitude. Diz-se então que o ruído segue uma distribuição Normal (Gaussiana), de Poisson, etc. O ruído diz-se branco quando a sua densidade espectral de potência média é constante a todas as frequências e é dito colorido no caso contrário. Para analisar o efeito do ruído em um sistema de transmissão é possível modelar o canal com o Ruído Branco Aditivo e Gaussiano (AWGN-*additive white gaussian noise*).

A fim de otimizar um sistema de comunicação digital devemos utilizar o mínimo da potência necessária para transmitir um sinal de modo que ainda possamos recuperá-lo no destino. E, para isto, são necessárias técnicas de codificação de canal para garantir a confiabilidade e qualidade da transmissão. Portanto, analisando as curvas BERxSNR podemos determinar quais códigos otimizam o sistema de transmissão, ou seja, quais os códigos BCH que apresentam as menores taxas de erro de bits (BER) a partir de uma relação sinal-ruído (SNR) mínima. Vejamos alguns resultados para um canal AWGN.

Na Figura 4.1 consideramos um canal AWGN e alguns códigos BCH com $n = 63$. Para uma relação sinal-ruído de $8\ dB$ podemos observar que o código $C(63, 10, 13)$ tem uma taxa de erro de bit que se aproxima de 10^{-3}. No entanto, quando aumentamos a distância para $d = 15$ obtemos o código $C(63, 7, 15)$, para o qual a taxa de erro de bit se aproxima de 10^{-4} para a mesma relação sinal-ruído de $8\ dB$. Portanto, o código $C(63, 7, 15)$ é mais eficiente. No entanto, note que se aumentarmos a relação sinal-ruído para valores acima de $10\ dB$ o código $C(63, 24, 7)$ atinge uma taxa de erro de bits próxima de 10^{-4}.

Do mesmo modo nas Figuras 4.2 e 4.3, respectivamente, para $n = 3$ e $n = 255$, podemos

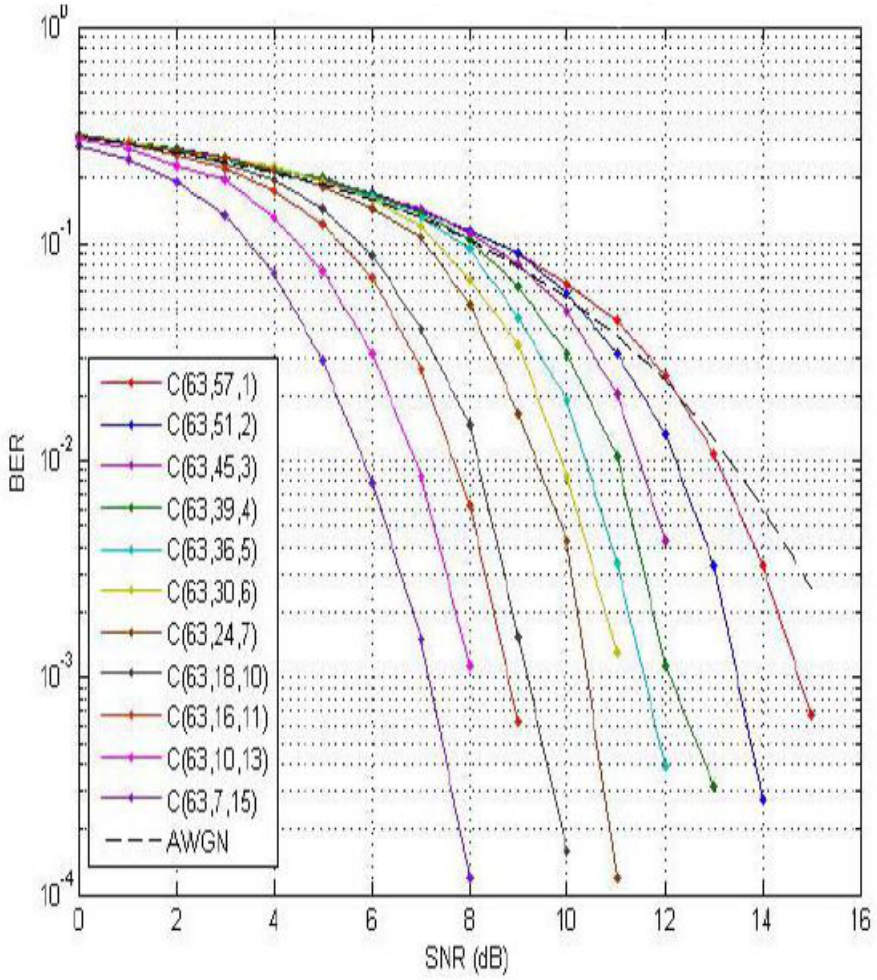


Figura 4.1: Desempenho do código BCH(63,k,d) num canal AWGN

observar que a taxa de erro de bits diminui quando aumentamos a distância mínima do código ou a relação sinal-ruído.

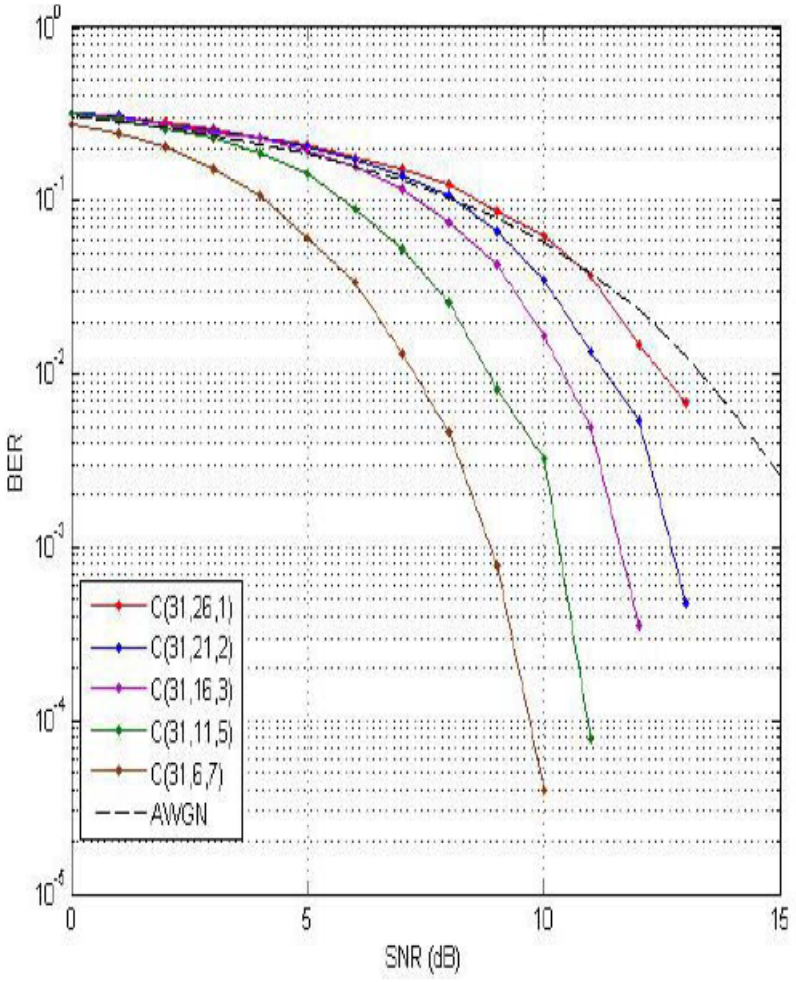


Figura 4.2: Desempenho do código BCH(31,k,d) num canal AWGN

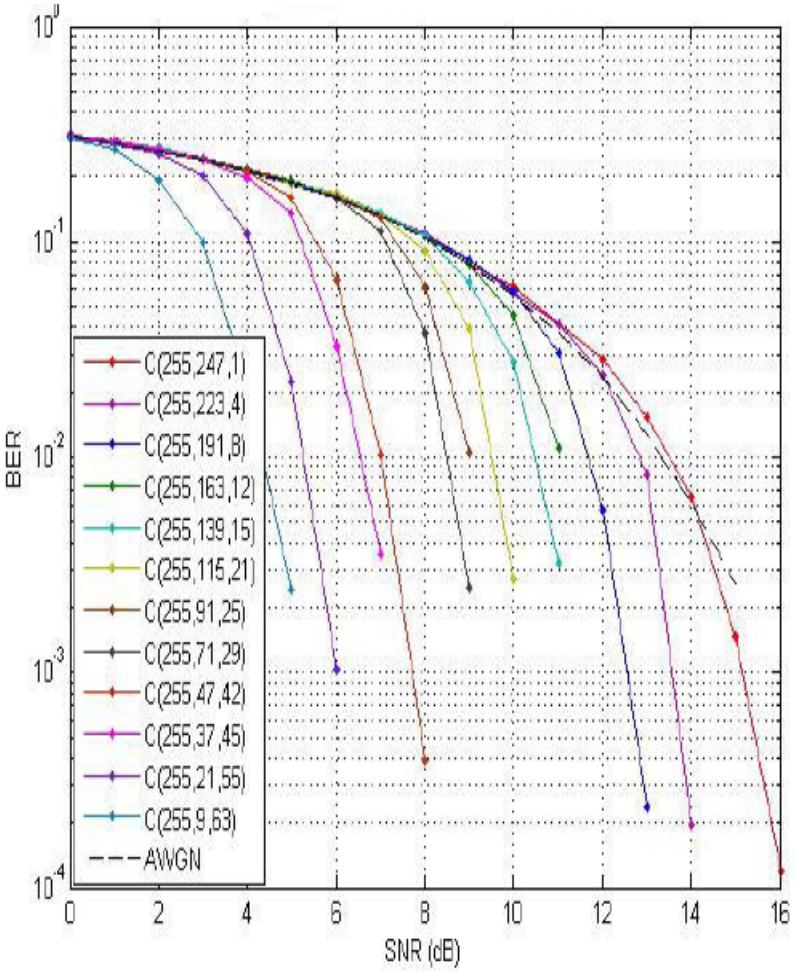


Figura 4.3: Desempenho do código BCH(255,k,d) num canal AWGN

CAPÍTULO 5

CONSIDERAÇÕES FINAIS

Diante da crescente digitalização dos meios de comunicação cada vez mais as pessoas têm acesso aos sistemas de comunicação. Portanto, no nosso dia-a-dia estamos sempre em contato de forma indireta com os Códigos Corretores de Erros, visto que estes são extremamente necessários para garantir a confiabilidade na transmissão ou armazenamento da informação em um sistema de comunicação digital.

Seja quando fazemos um simples telefonema, ou quanto ouvimos um CD, ou assistimos um DVD, estamos em contato direto com alguma informação que foi codificada a fim de corrigir possíveis erros introduzidos no canal de transmissão ou no dispositivo de armazenamento. Além disso, os profissionais formados em Matemática, tanto na modalidade licenciatura quanto no bacharelado, são constantemente questionados sobre quais as aplicações práticas do que estudam. Nesse contexto, surgiu a motivação para este trabalho e, assim, a oportunidade de mostrar a importância da Matemática para o desenvolvimento da codificação de canal.

Para isto foram apresentados vários conceitos de Álgebra tais como: anéis, classes residuais, ideais de um anel, corpos finitos, anéis de polinômios e espaços vetoriais. Vimos que dado um corpo finito K, os códigos de bloco lineares, de um modo geral, são subespaços vetoriais de K^n.

Além disso, mostrou-se que os códigos cíclicos podem ser definidos sob a estrutura de um ideal de um anel, e essa estrutura algébrica implica em várias propriedades relevantes para a eficiência desses códigos. Finalizamos nosso estudo com o caso particular dos códigos BCH, que formam uma subclasse importante de códigos cíclicos, capazes de corrigir erros múltiplos.

BIBLIOGRAFIA

[1] C.E. Shannon.“A mathematical Theory of Communication", Bell System Technical Journal, vol.27, pp. 379-423, July, 1948.

[2] S. Haykin. “Sistemas de Comunicação", Bookman, 4º edição, 2004.

[3] R.W. Hamming. “Error Detecting and Error Correcting Codes", The Bell Systems Technical Journal, vol. XXIX, nº 2, April, 1950.

[4] E. Prange. “Cyclic Error-Correcting Codes in Two Symbols", AFCRC-TN-57, 103, Air Force Cambridge Research Center, Cambridge, Mass., September, 1957.

[5] A. Hocquenghem, “Codes corecteurs d’erreurs", Chiffres, vol. 2, pp. 147-156, 1959.

[6] R.C. Bose and D. K. Ray-Chaudhuri, “On a Class opf Error Correcting Binary Group Codes", Information and Control, vol. 3, pp. 68-79, March 1960.

[7] I. Reed and G. Solomon. “Polynomial codes over certain finite fields", Ioint Society of Industrial and Applied Mathematics Journal 8, n.2, 300-304, 1960.

[8] E. R. Berlekamp, “On Decoding Binary Bose-Chaudhuri-Hocquenghem Codes", IEEE Trans. Inf. Theory, IT-11, pp. 577-580, October, 1965.

[9] A. Hefez e M.L.T. Villela, “Códigos Corretores de Erros", IMPA, Rio de Janeiro, 2008.

[10] . D. J. Costello and S. Lin,“Error Control Coding: Fundamentals and Applications". Englewood Cliffs. New Jersey Prentice Hall, 1983.

[11] C.P. Milies. “Breve introdução à Teoria dos Códigos Corretores de Erros",Colóquio de Matemática da Região Centro-oeste, UFMS. Novembro, 2009.

[12] W.W. Peterson, “Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes", IRE Trans. Inform. Theory, IT-6:459-70, September, 1960.

[13] D. Gorenstein and N. Zierler, “A Class of Cyclic Linear Error-Correcting Codes in p^m Symbols", J. Soc. Ind. Appl. Math., 9: 107-214, June, 1961.

[14] J.L. Massey,“Step-by-step Decoding of the Bose-Chaudhuri-Hocquenghem Codes", IEEE, Transf. Inf. Theory, IT-11, pp. 580-585, October, 1965.

[15] J.L. Massey,“Shift-Registerr Synthesis and BCH Decoding", IEEE, Transf. Inf. Theory, IT-15, pp. 122-127, January, 1969.

[16] R.T Chien,“Cyclic Decoding Procedure for the Bose-Chaudhuri-Hocquenghem Codes", IEEE, Transf. Inf. Theory, IT-10, pp. 357-363, October, 1964.

[17] A. Gonçalves, “Introdução à Álgebra", IMPA, Rio de Janeiro, 2008.

[18] F.U. Coelho e M.L. Lourenço, “Um Curso de Álgebra Linear", 2ª edição, Edusp, São Paulo, 2007.