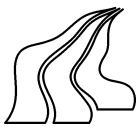
Quasi-Cyclic Codes

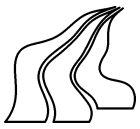
Represented by Gröbner Bases

Aalborg University

Department of Mathematical Sciences
Thomas H. Skjærbæk • MAT6 • 1. Feb. 2010 – 15. Jun. 2010



Aalborg University
Department of Mathematical Sciences



TITLE:

Quasi-Cyclic Codes Represented by Gröbner Bases

PROJECT PERIOD:

From 1. February 2010

To 15. June 2010

AUTHOR:

Thomas Hassing Skjærbæk

Supervisor:

Diego Ruano

Olav H. Geil

COPIES: 5

PAGES: 83

SYNOPSIS:

We will begin by defining modules and submodules. We will show that submodules are a generalization of ideals, and we will generalize the Gröbner basis theory from ideals to submodules. Some of the basic theory about linear codes will be consider, and we will consider the cyclic codes. Quasi-cyclic codes, which are a generalization of the cyclic codes, will be defined and studied. We will show that we can use the Gröbner basis theory for modules on the quasi-cyclic codes to find a generating set for these.
We will give a method to decode Reed-Solomon codes by using the theory about modules and Gröbner basis, and we will give an algorithm for converting a Gröbner basis with respect to one monomial order to a Gröbner basis with respect to another monomial order.
In the last chapter we will consider a decoding method for quasi-cyclic codes that uses their Gröbner basis representation, and we will discuss this method's weaknesses. We will then restrict ourself to a specific set of quasi-cyclic codes and give an algorithm for decoding these.

Danish Summary

I Kapitel 1 vil vi introducere moduler and submoduler i en kommutativ ring R^m, hvor R for det meste er en polynomiumsring $k[x_1, \ldots, x_n]$. Submoduler er en generalisering af idealer i R, og det vil derfor være naturligt at generalisere teorien omkring Gröbner baser for ideals til Gröbner baser for submoduler. For at kunne gøre dette får du brug for at define monomiale ordner for R^m, and vi vil betragte to af de mest almindelige. En divisions algoritme for R^m vil blive introduceret, med hvilken vi vil være i stand til at dividere et element $\boldsymbol{f} \in R^m$ med et sæt af elements $(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s)$. Efter vi formelt har defineret Gröbner baser for submoduler i R^m, vil vi vise at givet en Gröbner base for et submodul $M \in R^m$ og et element $\boldsymbol{f} \in M$, at vi kan bestemme om $\boldsymbol{f} \in M$. I det sidste afsnit af dette kapitel vil vi betragte endnu en egenskab af Gröbner baserne. Vi vil vise at givet et sæt af generatorer $(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s)$ for et submodul $M \subseteq R^m$, så vil vi være i stand til at finde et sæt af generatorer for syzygy-modulet $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s) \subseteq R^s$.

I Kapitel 2 vil vi betragte noget fundamental teori omkring lineære koder. Vi vil betragte de cykliske koder og vise at disse kan betragtes som idealer i kvotient ringen $\mathbb{F}_q[x]/\langle x^n - 1\rangle$. Quasi-cykliske koder, some er en generalisering af cykliske koder, vil også blive introduceret. Vi vil vise at quasi-cykliske coder af længde $n = lm$ kan repræsenteres af submoduler i $R^l/\langle x^m - 1\rangle$. Det følger, at det vil være naturligt at repræsentere generatorerne af de quasi-cykliske coder som Gröbner baser, og vi vil bevise en sætning, som viser strukturen af disse Gröbner baser.

En speciel klasse af cykliske koder vil blive betragtet i Kapitel 3, nemlig Reed-Solomon koderne. Vi vil give en metode til at dekode Reed-Solomon koderne, som benytter noget af den teori, som vi har fra moduler og Gröbner baser. Til sidst i dette kapitel vil vi give en algorithm til at konvertere en givet Gröbner baser for et eller andet modul med hensyn til en monomial orden til en Gröbner base for det samme modul men med hensyn til en anden monomial orden.

I det sidste kapitel af dette speciale vil vi betragte dekodning af quasi-cykliske

coder, hvor vi vil benytte deres Gröbner base repræsentation. Vi vil se at i denne generelle situation, hvor der er mere en én generator, så vil denne metode fejle, hvis blot en blok indeholder for mange fejl. Vi vil derfor begrænse os til at betragte quasi-cykliske koder genereret af en generator. Den generelle situation her har stadig en svaghed, som vi vil diskutere, før vi begrænser os yderligere til at betragte quasi-cykliske koder med én generator af en specifik form. Vi vil give en algoritme til at dekode disse, og vi vil vise at denne algorithm er meget effektiv, hvis vores modtagne ord mangler større dele.

Preface

This thesis is written by Thomas Hassing Skjærbæk on the MAT6-semester at Aalborg University in the period February 1st 2010 to June 15th 2010.

This thesis is written in the main field of Discrete Mathematics, and it is a continuation of the MAT5-semester project written in the fall of last year about Gröbner bases for ideals in the polynomial ring. Throughout the thesis the computer algebra system Singular (http://www.singular.uni-kl.de/) has been used to do calculations.

Aalborg, 15th of June 2010.

Thomas Hassing Skjærbæk

Contents

Introduction

In Chapter 1 we will introduce modules and submodules of a commutative ring R^m, where we will mostly work over the polynomial ring $R = k[x_1, \ldots, x_n]$. Submodules are a generalisation of ideals of R, whereby it will be natural to generalize the theory of Gröbner bases for ideals to Gröbner bases for modules. For this we will need to define monomial orders for R^m, and we will consider two of the most common. A division algorithm for R^m will be introduced, which will allow us to divide an element $\boldsymbol{f} \in R^m$ with a set of elements $(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s)$. After the Gröbner bases for submodules of R^m have been formally defined, we will solve the Submodule Membership Problem; that is, we will show that given an element $\boldsymbol{f} \in R^m$ and a submodule $M \subseteq R^m$ we can use a Gröbner basis to determine if $\boldsymbol{f} \in M$. The last section of this chapter will be devoted to another property of Gröbner bases, namely the so-called syzygy modules. We will show that given a set of generators $(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s)$ for a submodule $M \subseteq R^m$, we will be able to find a set of generators for the syzygy module $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s) \subseteq R^s$.

In Chapter 2 we will consider some basic theory about linear codes. We will then consider the cyclic codes and show that these can be consider as ideals in the quotient ring $\mathbb{F}_q[x]/\langle x^n - 1\rangle$. Quasi-cyclic codes, which are a generalisation of the cyclic codes, will be introduced. We will show that quasi-cyclic codes of length $n = lm$ can be represented by submodules of $R^l/\langle x^m - 1\rangle$. It follows that it is natural to represent the generators of the quasi-cyclic codes as Gröbner bases, and we will proof a theorem that shows the structure of these Gröbner bases.

A special class of cyclic codes will be consider in Chapter 3, namely the Reed-Solomon codes. We will give a decoding method for the Reed-Solomon codes, which utilizes some of the theory of modules and Gröbner bases. In the end of this chapter we will give an algorithm for converting a given Gröbner basis for some submodule with respect to one order to a reduced Gröbner basis for the same submodule but with respect to another order.

In the last chapter of this thesis we will consider decoding of quasi-cyclic

codes, where we will use their Gröbner basis representation. We will see that in the general case with more than one generator that this method will fail if we have just one block with too many errors. We will therefore restrict ourself to considering 1-generator quasi-cyclic codes. The general case of these still have a weakness, which we will discuss before restricting ourself to 1-generator quasi-cyclic codes, where the generator has a specific structure. We will give an algorithm for decoding these, and we will show that this algorithm is very effective if our received word contains a lot of erasures.

Chapter 1

Modules

In this chapter we will introduce modules and submodules over a commutative ring R. We will see that modules have similarities with ideals, and we will show that ideals are actually just 1-dimensional modules. The Gröbner basis theory from ideals will be generalized to modules. To do this we will need to define monomial orders for R^m and give a division algorithm for R^m. In the last section of this chapter we will consider Syzygy modules, which are a special type of submodules. This chapter is based on [Cox et al., 2005, Section 5.1–5.3] and [Cox et al., 2007, §1–§5 of Chapter 2].

1.1 Modules

We will begin with the formal definition of a module over a commutative ring.

Definition 1.1.1 (Modules)
A module over a commutative ring with unity R (or an R-module) is a set M together with the binary operations addition $(+)$ and scalar multiplication $(\cdot)$ with the following properties:

(i) M is an abelian group under addition; that is, addition in M is associative and commutative, there is an additive identity $\mathbf{0} \in M$, and each element $\boldsymbol{f} \in M$ has an additive inverse $-\boldsymbol{f}$, satisfying $\boldsymbol{f} + (-\boldsymbol{f}) = \mathbf{0}$,

(ii) For all $a \in R$ and all $\boldsymbol{f}, \boldsymbol{g} \in M$ we have $a(\boldsymbol{f} + \boldsymbol{g}) = a\boldsymbol{f} + a\boldsymbol{g}$,

(iii) For all $a, b \in R$ and all $\boldsymbol{f} \in M$ we have $(a + b)\boldsymbol{f} = a\boldsymbol{f} + b\boldsymbol{f}$,

(iv) For all $a, b \in R$ and all $\boldsymbol{f} \in M$ we have $(ab)\boldsymbol{f} = a(b\boldsymbol{f})$,

(v) If 1 is the multiplicative unity in R, then $1\boldsymbol{f} = \boldsymbol{f}$ for all $\boldsymbol{f} \in M$.

The simplest modules are those equal to R^m over R; that is, the ones that consist of all $m \times 1$ matrices with elements of R, and where addition and scalar multiplication is defined, respectively, as

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_m + b_m \end{bmatrix}, \quad c \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} ca_1 \\ ca_2 \\ \vdots \\ ca_m \end{bmatrix},$$

where $a_i, b_i, c \in R$, $1 \leq i \leq m$. Just like we can construct ideals of a ring R as a subset of R, we can construct submodules of R^m as subsets of R^m, which satisfy the conditions in Definition 1.1.1, and which are therefore also modules. More formally, we have the following definition.

Definition 1.1.2 (Submodules)
Let $M \subseteq R^m$ be R-module, and let $N \subseteq R^m$ be a subset of M. If, for any $\boldsymbol{f}, \boldsymbol{g} \in N$ and $a \in R$, we have $a\boldsymbol{f} + \boldsymbol{g} \in N$, then we call N a submodule of M.

As an example of a submodule of R^m let $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s$ be a set of $m \times 1$ matrices. Then the set

$$\langle \boldsymbol{f}_1, \ldots, \boldsymbol{f}_s \rangle = \{a_1\boldsymbol{f}_1 + \cdots + a_s\boldsymbol{f}_s \in R^m | a_i \in R, 1 \leq i \leq s\}$$

of all possible R-linear combinations of these $m \times 1$ matrices is a submodule of R^m. We can generalize this result to any generating set $F = \langle \boldsymbol{f}_1, \ldots, \boldsymbol{f}_i, \ldots \rangle \subseteq M$, where M is any R-module, by constructing a submodule N of M as the collection of all $\boldsymbol{f} \in M$ that can be written as

$$\boldsymbol{f} = \sum_{i=1}^{\infty} a_i \boldsymbol{f}_i,$$

with $a_i \in R$, where finitely many $a_i \neq 0$, and $\boldsymbol{f}_i \in F$. To see that N is an R-module, note that if $\boldsymbol{f}, \boldsymbol{g} \in N$ and $a \in R$, then $\boldsymbol{f} = \sum_{i=1}^{\infty} a_i \boldsymbol{f}_i$ and $\boldsymbol{g} = \sum_{i=1}^{\infty} b_i \boldsymbol{f}_i$ such that

$$\begin{aligned} a\boldsymbol{f} + \boldsymbol{g} &= a \sum_{i=1}^{\infty} a_i \boldsymbol{f}_i + \sum_{i=1}^{\infty} b_i \boldsymbol{f}_i \\ &= \sum_{i=1}^{\infty} (aa_i + b_i) \boldsymbol{f}_i, \end{aligned}$$

where $aa_i + b_i \in R$. Thus, $a\boldsymbol{f} + \boldsymbol{g} \in N$ and N is a submodule of M by Definition 1.1.2.

If the ring R is a field k, then the definition of an R-module is the same as that of a vector space over k. If, however, R is a polynomial ring $k[x_1, \ldots, x_n]$, then the modules can exhibit behavior different from vector spaces. We illustrate this in the following example.

Example 1.1.3

Let $R = k[x, y, z]$ be a polynomial ring, and $M = \langle \boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3 \rangle \subseteq R^3$, where

$$\boldsymbol{f}_1 = \begin{bmatrix} y \\ -x \\ 0 \end{bmatrix}, \boldsymbol{f}_2 = \begin{bmatrix} z \\ 0 \\ -x \end{bmatrix}, \boldsymbol{f}_3 = \begin{bmatrix} 0 \\ z \\ -y \end{bmatrix},$$

be an R-module, and consider the 1×3 matrix $A = [x\ y\ z]$. Then it is seen that $M = \ker A = \{\boldsymbol{f} \in R^3 | A\boldsymbol{f} = 0\}$. Since $A\boldsymbol{f}_i = 0$ for $i = 1, 2, 3$ the $\subseteq$ inclusion follows. To see the other inclusion consider $\boldsymbol{f} = [g_1\ g_2\ g_3]^T$ such that $A\boldsymbol{f} = g_1x + g_2y + g_3z$. The following relations give the desired inclusion:

$$\begin{aligned} g_1 = 0 &\Rightarrow g_2 = z \cdot \text{something}, g_3 = y \cdot \text{something}, \\ g_2 = 0 &\Rightarrow g_1 = z \cdot \text{something}, g_3 = x \cdot \text{something}, \\ g_3 = 0 &\Rightarrow g_1 = y \cdot \text{something}, g_2 = x \cdot \text{something}, \end{aligned}$$

and note that if g_1, g_2, g_3 are all nonzero, then it follows that

$$\begin{aligned} g_1 &= y \cdot \text{something} + z \cdot \text{something}, \\ g_2 &= x \cdot \text{something} + z \cdot \text{something}, \\ g_3 &= x \cdot \text{something} + y \cdot \text{something}. \end{aligned}$$

The generating set $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3\}$ is minimal in the sense that $M \neq \langle \boldsymbol{f}_i, \boldsymbol{f}_j \rangle$, $1 \leq i < j \leq 3$, since $\boldsymbol{f}_k \notin \langle \boldsymbol{f}_i, \boldsymbol{f}_j \rangle$ for $k \neq i, j$. Now, note that $z\boldsymbol{f}_1 - y\boldsymbol{f}_2 + x\boldsymbol{f}_3 = 0$, which shows that the generating vectors is linear dependent. This property that a minimal generating set it not linearly independent is not possible with any vector space. □

Modules over R is closely related to ideals in R. In fact, ideals are just 1-dimensional R-modules, which can be easily verified. If $M \subseteq R$ is an R-module, then M is an ideal in R. Too see this note that if $\boldsymbol{f} = \sum_{i=1}^{\infty} a_i \boldsymbol{f}_i \in M$ and $\boldsymbol{g} = \sum_{i=1}^{\infty} b_i \boldsymbol{f}_i \in M$, where a_i, b_i are nonzero polynomials for a finite set of $i \in \{1, \ldots, \infty\}$, then $\boldsymbol{f} + \boldsymbol{g} = \sum_{i=1}^{\infty} (a_i + b_i)\boldsymbol{f}_i \in M$. If further $h \in R$,

then $h\boldsymbol{f} \in M$, since this just correspondent to scalar multiplication, which modules are closed under. Hence, M satisfies the conditions for an ideal in R. That an ideal is an R-module follows directly from the same argumentation. We will now show how to define modules in the computer algebra program Singular.

Example 1.1.4

We first define the ring we will be working over. Here we will use $\mathbb{Q}[x, y, z]$ over the monomial order $\succeq_{\text{TOP}}$ over the lex order (which we define in Section 1.2). We will then define the module used in Example 1.1.3.

```
>ring R=0,(x,y,z),(lp,c);
>vector f1=[y,-x,0];
>vector f2=[z,0,-x];
>vector f2=[0,z,-y];
>module M=f1,f2,f3;
>print(M);
y, z, 0,
-x,0, z,
0, -x,-y
```

Note that in Singular we define rows as vectors. Since the elements of the module is defined as vectors, we can easily do addition and scalar multiplication.

```
>vector f4=f1+f2+f3;
>print(f4);
[y+z,-x+z,-x-y]
>vector f5=(x-2z)*f1;
>print(f5);
[xy-2yz,-x2+2xz]
```

□

In Example 1.1.3 we saw one of the main differences between vector spaces and modules, namely that a minimal generating set for a module is not necessarily linearly independent. In a vector space a minimal generating set is called a basis and is always linearly independent and, thus, minimal in the sense that it contains the minimal amount of vectors to generate the whole space.

Definition 1.1.5 (A Basis)
A basis is a minimal generating set where the generators are linear independent.

If R is the polynomial ring $k[x_1, \ldots, x_n]$, then any module over R that requires more than a single generator cannot have a generating set F which is linearly independent, since any two polynomials $\boldsymbol{f}_1, \boldsymbol{f}_2 \in F$ satisfy the non-trivial linear dependence relation $\boldsymbol{f}_2\boldsymbol{f}_1 - \boldsymbol{f}_1\boldsymbol{f}_2 = \mathbf{0}$. We will distinguish between the two types of bases by referring to a basis in $k[x_1, \ldots, x_n]$ as an ideal basis, and the usual linearly independent basis as a module basis. The following proposition states when a module does have a module basis.

Proposition 1.1.6
Let M be an R-module. A set $F = \{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n\} \subseteq M$ is a module basis for M if and only if every element $\boldsymbol{f} \in M$ can be written uniquely in the form

$$\boldsymbol{f} = a_1\boldsymbol{f}_1 + \cdots + a_n\boldsymbol{f}_n, \tag{1.1}$$

where $a_1, \ldots, a_n \in R$.

Proof
Suppose that $F = \{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n\}$ is a module basis for M, and let $\boldsymbol{f} \in M$. Since F is a basis it spans M and, thus, we can write $\boldsymbol{f}$ as in Equation (1.1). Now, suppose Equation (1.1) is not unique. Then there exists $b_1, \ldots, b_n \in R$ such that

$$\boldsymbol{f} = b_1\boldsymbol{f}_1 + \cdots + b_n\boldsymbol{f}_n.$$

Subtracting this equation from Equation (1.1) yields

$$0 = (a_1 - b_1)\boldsymbol{f}_1 + \cdots + (a_n - b_n)\boldsymbol{f}_n,$$

but since $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n$ are linearly independent it follows that $a_i - b_i = 0$ for $i = 1, \ldots, n$. Hence, $a_i = b_i$ for every i, and Equation (1.1) is unique.

To show the other way, suppose that any $\boldsymbol{f} \in M$ can be written uniquely as Equation (1.1). This means that $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n$ spans M, and we just need to show that this set is linearly independent. For this, consider the equation

$$0 = a_1\boldsymbol{f}_1 + \cdots + a_n\boldsymbol{f}_n.$$

Since this equation is unique, it follows that $a_1 = \cdots = a_n = 0$, and, thus, $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n$ is linearly independent. Hence, $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n$ is a module basis for M. ■

Modules that do have a module basis are given a special name.

Definition 1.1.7 (Free Module)
Let M be an R-module. M is said to be a free module if M has a module basis.

For instance, when $M = R^m$ we always have the standard basis with the elements

$$\boldsymbol{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \boldsymbol{e}_m = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Naturally, every module over a field is a free module, since it is equivalent to a vector space.

In general, it can be difficult to determine whether a submodule of R^m is free. The next theorem by Quillen and Suslin, which we will not prove, says that when a submodule $M = \ker A$ over $k[x_1, \ldots, x_n]$, where $A = [f_1 \ \cdots \ f_m]$, then it is free.

Theorem 1.1.8 (Quillen-Suslin)
Let $R = k[x_1, \ldots, x_n]$ and suppose that $f_1, \ldots, f_m \in R$ are polynomials that generate all of R – that is, $\langle f_1, \ldots, f_m \rangle = \langle 1 \rangle = R$. Then the module M of all solutions $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m)^T \in R^m$ of the linear equation

$$f_1 \boldsymbol{X}_1 + \cdots + f_m \boldsymbol{X}_m = \boldsymbol{0}$$

is free.

We now consider homogeneous R-linear equations of the form

$$a_1 \boldsymbol{f}_1 + \cdots + a_t \boldsymbol{f}_t = \boldsymbol{0},$$

where $a_i \in R$, $1 \leq i \leq t$, and $\boldsymbol{f}_i$, $1 \leq i \leq t$, are elements of some submodule in R^m. In the next proposition we will prove that the set of all t-tuples $(a_1, \ldots, a_t)$ satisfying the above equation is a submodule of R^t.

Proposition 1.1.9
Let $(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$ be an ordered t-tuple of elements $\boldsymbol{f}_i \in M$. The set of all $(a_1, \ldots, a_t)^T \in R^t$ such that $a_1 \boldsymbol{f}_1 + \cdots + a_t \boldsymbol{f}_t = \boldsymbol{0}$ is an R-submodule of R^t called the (first) syzygy module of $(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$ and denoted $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$.

Proof

We want to prove that $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$ is closed under addition and scalar multiplication. Let $(a_1, \ldots, a_t)^T, (b_1, \ldots, b_t)^T \in \mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$, and let $c \in R$. Then

$$\begin{aligned} a_1\boldsymbol{f}_1 + \cdots + a_t\boldsymbol{f}_t &= \mathbf{0}, \\ b_1\boldsymbol{f}_1 + \cdots + b_t\boldsymbol{f}_t &= \mathbf{0}. \end{aligned}$$

Now, multiply the first equation by c and add it to the second,

$$(ca_1 + b_1)\boldsymbol{f}_1 + \cdots + (ca_t + b_t)\boldsymbol{f}_t = \mathbf{0}.$$

That is, we also have $((ca_1+b_1), \ldots, (ca_t+b_t))^T \in \mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$, and, thus, $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$ is a submodule of R^t. ■

We can represent a submodule $M = \langle \boldsymbol{f}_1, \ldots, \boldsymbol{f}_t \rangle \subseteq R^t$ with a so-called presentation matrix. A presentation matrix for M is any matrix whose columns span $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$. For instance, if the linear relations

$$\begin{aligned} a_{11}\boldsymbol{f}_1 + \cdots + a_{t1}\boldsymbol{f}_t &= \mathbf{0}, \\ &\vdots \\ a_{1s}\boldsymbol{f}_1 + \cdots + a_{ts}\boldsymbol{f}_t &= \mathbf{0}, \end{aligned}$$

generate $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$, then a presentation matrix for M is

$$\begin{bmatrix} a_{11} & \ldots & a_{1s} \\ \vdots & \ddots & \vdots \\ a_{t1} & \ldots & a_{ts} \end{bmatrix}.$$

Example 1.1.10

Consider the submodule $M \subseteq R^3$ defined in Example 1.1.3, $M = \langle \boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3 \rangle$, where

$$\boldsymbol{f}_1 = \begin{bmatrix} y \\ -x \\ 0 \end{bmatrix}, \boldsymbol{f}_2 = \begin{bmatrix} z \\ 0 \\ -x \end{bmatrix}, \boldsymbol{f}_3 = \begin{bmatrix} 0 \\ z \\ -y \end{bmatrix}.$$

We can use Singular to calculate the syzygy module $\mathrm{Syz}(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3) \subseteq R^3$.

```
>ring R=0,(x,y,z),(lp,c);
>vector f1=[y,-x,0];
```

```
>vector f2=[z,0,-x];
>vector f2=[0,z,-y];
>module M=f1,f2,f3;
>print(syz(M));
z,
-y,
x
```

Thus, we have the linear equation

$$z\boldsymbol{f}_1 - y\boldsymbol{f}_2 + x\boldsymbol{f}_3 = \mathbf{0},$$

which generates $\mathrm{Syz}(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3)$. A presentation matrix is given as

$$\begin{bmatrix} z \\ -y \\ x \end{bmatrix}.$$

□

We also need to define the quotient of a module.

Definition 1.1.11 (The Quotient of M by N)
Let N be a submodule of M, and let

$$[\boldsymbol{f}] = \{\boldsymbol{g} \in M | \boldsymbol{g} - \boldsymbol{f} \in N\} = \boldsymbol{f} + N$$

denote the set of all elements of M equivalent to $\boldsymbol{f}$, called an equivalent class or equivalent coset of M. The quotient of M by N, denoted M/N, is the set of all equivalent classes in M.

$$M/N = \{[\boldsymbol{f}] | \boldsymbol{f} \in M\}.$$

Operations in M/N are defined as

$$\begin{aligned} [\boldsymbol{f}] + [\boldsymbol{g}] &= [\boldsymbol{f} + \boldsymbol{g}], \\ a[\boldsymbol{f}] &= [a\boldsymbol{f}], \end{aligned}$$

where $\boldsymbol{f}, \boldsymbol{g} \in M/N$ and $a \in R$.

The quotient of M by N is an R-module and the operations in M/N are well-defined. To see that the operations are well-defined, let $\boldsymbol{f}' \in [\boldsymbol{f}]$ and

$\boldsymbol{g}' \in [\boldsymbol{g}]$. Then $\boldsymbol{f}' = \boldsymbol{f} + \tilde{\boldsymbol{f}}$ and $\boldsymbol{g}' = \boldsymbol{g} + \tilde{\boldsymbol{g}}$ for some $\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{g}} \in N$. Since

$$\boldsymbol{f}' + \boldsymbol{g}' = (\boldsymbol{f} + \tilde{\boldsymbol{f}}) + (\boldsymbol{g} + \tilde{\boldsymbol{g}}) = (\boldsymbol{f} + \boldsymbol{g}) + (\tilde{\boldsymbol{f}} + \tilde{\boldsymbol{g}}),$$

where $\tilde{\boldsymbol{f}} + \tilde{\boldsymbol{g}} \in N$, we have $[\boldsymbol{f}' + \boldsymbol{g}'] = [\boldsymbol{f} + \boldsymbol{g}]$. Now, let $a \in R$. Then

$$a\boldsymbol{f}' = a(\boldsymbol{f} + \tilde{\boldsymbol{f}}) = a\boldsymbol{f} + a\tilde{\boldsymbol{f}},$$

where $a\tilde{\boldsymbol{f}} \in N$, and, thus, $[a\boldsymbol{f}'] = [a\boldsymbol{f}]$. Hence, the operations defined in Definition 1.1.11 are well-defined. To see that M/N is an R-module, let $[\boldsymbol{f}], [\boldsymbol{g}] \in M/N$ and let $a \in R$. Then

$$a[\boldsymbol{f}] + [\boldsymbol{g}] = [a\boldsymbol{f} + \boldsymbol{g}],$$

and since $a\boldsymbol{f} + \boldsymbol{g} \in M$ we have $[a\boldsymbol{f} + \boldsymbol{g}] \in M/N$, and, thus, M/N is an R-module. The zero element of M/N, $[\mathbf{0}]$, is a set that can be represented by any element of the submodule N.

It is natural to define mappings that preserve some given structures, and we will, thus, consider R-module homomorphisms.

Definition 1.1.12 (R-module Homomorphism)
An R-module homomorphism between two R-modules M and N is an R-linear map between M and N, $\phi : M \to N$, such that for all $\boldsymbol{f}, \boldsymbol{g} \in M$ and all $a \in R$ we have

$$\phi(a\boldsymbol{f} + \boldsymbol{g}) = a\phi(\boldsymbol{f}) + \phi(\boldsymbol{g}).$$

The homomorphism ϕ is called an isomorphism if it is both one-to-one and onto. If ϕ is an isomorphism, then M and N are said to be isomorphic, written $M \cong N$.

As an example of a natural homomorphism we have the map between a module M and the quotient M/N, $N \subseteq M$, given by $\phi(\boldsymbol{f}) = [\boldsymbol{f}]$ for every $\boldsymbol{f} \in M$.

Proposition 1.1.13
Suppose that A is an $l \times m$ matrix with entries in R, and suppose that A is a presentation matrix for two different R-modules M and N. Then

(i). M and N are isomorphic as R-modules,

(ii). M (and, hence, N) is isomorphic to R/AR^m where AR^m denotes the image im$A$ *of R^m under multiplication by A.*

Proof

For part (i) note that since A is a presentation matrix for M, then there exists generators $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_l$ for M such that the columns of A generate $\text{Syz}(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_l)$. Since A is also a presentation matrix for N, there exists generators $\boldsymbol{n}_1, \ldots, \boldsymbol{n}_l$ for N such that the columns of A generate $\text{Syz}(\boldsymbol{n}_1, \ldots, \boldsymbol{n}_l)$. Let $\phi : M \to N$ be the homomorphism defined by $\phi(\boldsymbol{m}_i) = \boldsymbol{n}_i$, so $\phi(\sum_{i=1}^{l} c_i \boldsymbol{m}_i) = \sum_{i=1}^{l} c_i \boldsymbol{n}_i$ for some $c_i \in R$. ϕ is clearly onto, since M and N consist of the same number of generators and every $\boldsymbol{n}_i$ is just mapped from $\boldsymbol{m}_i$. To see that ϕ is also one-to-one, let $\sum_{i=1}^{l} c_i \boldsymbol{m}_i = \sum_{i=1}^{l} d_i \boldsymbol{m}_i$ for some $d_i \in R$. Using ϕ on both sides of the equality gives

$$\sum_{i=1}^{l} c_i \boldsymbol{n}_i = \phi\left(\sum_{i=1}^{l} c_i \boldsymbol{m}_i\right) = \phi\left(\sum_{i=1}^{l} d_i \boldsymbol{m}_i\right) = \sum_{i=1}^{l} d_i \boldsymbol{n}_i, \tag{1.2}$$

which proves the one side. Now, suppose $\sum_{i=1}^{l} c_i \boldsymbol{n}_i = \sum_{i=1}^{l} d_i \boldsymbol{n}_i$. Then Equation (1.2) shows that we also have $\phi(\sum_{i=1}^{l} c_i \boldsymbol{m}_i) = \phi(\sum_{i=1}^{l} d_i \boldsymbol{m}_i)$, which proves that ϕ is well-defined and, thus, one-to-one. Since ϕ is both onto and one-to-one, it is an isomorphism, and $M \cong N$.

To prove part (ii), note that since A is an $l \times m$ matrix, then AR^m is a submodule of R^l generated by the columns of A. The quotient R^l/AR^m is generated by the set $(\boldsymbol{e}_1 + AR^m, \ldots, \boldsymbol{e}_l + AR^m)$, where $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_l$ are the standard basis vectors of R^l. Consider the equation

$$\begin{aligned} \mathbf{0} &= c_1(\boldsymbol{e}_1 + AR^m) + \cdots + c_l(\boldsymbol{e}_l + AR^m) \\ &= (c_1\boldsymbol{e}_1 + \cdots + c_l\boldsymbol{e}_l) + (c_1 + \cdots + c_l)AR^m \end{aligned}$$

for some $c_1, \ldots, c_l \in R$. That is, $(c_1, \ldots, c_l)^T \in \text{Syz}(\boldsymbol{e}_1 + AR^m, \ldots, \boldsymbol{e}_l + AR^m)$ and $(c_1, \ldots, c_l)^T \in \text{Syz}(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_l)$. This is true if and only if $(c_1, \ldots, c_l)^T \in AR^m$. This means that $(c_1, \ldots, c_l)^T$ is spanned by the columns of A. It follows that A is a presentation matrix for R^l/AR^m, and since A is also a presentation matrix for M and N, we must have $M \cong N \cong R^l/AR^m$. ■

In the next section we will generalize some of known theory from ideals, such as monomial orders and Gröbner basis. As with ideals, we will consider the membership problem to determine when a given element in R^m is an element of submodule in R^m.

1.2 Monomial Orders and Gröbner Bases for Modules

In this section R will be the polynomial ring $k[x_1, \ldots, x_n]$. We will generalize the theory of monomial orders and Gröbner basis from ideals to submodules in R^m, and we will consider the following problems:

(i). (Submodule Membership) Given a submodule $M \subseteq R^m$ and $\boldsymbol{f} \in R^m$, determine if $\boldsymbol{f} \in M$.

(ii). (Syzygies) Given an ordered s-tuple of generators $(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s)$ of an R-module M over R^m, find a set of generators for the module $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s) \subseteq R^s$. In other words, find a presentation matrix for M.

A monomial $\boldsymbol{m}$ is an element of the form $x^\alpha \boldsymbol{e}_i$ for some i, where $\alpha \in \mathbb{N}_0^n$. Every element $\boldsymbol{f} \in R^m$ can be written uniquely as a k-linear combination of monomials $\boldsymbol{m}_i$,

$$\boldsymbol{f} = \sum_{i=1}^{n} c_i \boldsymbol{m}_i,$$

where $c_i \in k$, $c_i \neq 0$, and is called a coefficient. Every element $c_i \boldsymbol{m}_i$ of the sum is called a term.

Example 1.2.1
Let $R^3 = (k[x, y])^3$. Then

$$\begin{aligned}
\boldsymbol{f} &= \begin{bmatrix} 3x^2y^2 + y^2 + 2 \\ 2x^4 \\ xy^2 - 5x \end{bmatrix} \\
&= 3\begin{bmatrix} x^2y^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} y^2 \\ 0 \\ 0 \end{bmatrix} + 2\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 2\begin{bmatrix} 0 \\ x^4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ xy^2 \end{bmatrix} - 5\begin{bmatrix} 0 \\ 0 \\ x \end{bmatrix} \\
&= 3x^2y^2\boldsymbol{e}_1 + y^2\boldsymbol{e}_1 + 2\boldsymbol{e}_1 + 2x^4\boldsymbol{e}_2 + xy^3\boldsymbol{e}_3 - 5x\boldsymbol{e}_3,
\end{aligned}$$

which is a k-linear combination of monomials. □

If $\boldsymbol{m} = x^\alpha \boldsymbol{e}_i$ and $\boldsymbol{n} = x^\beta \boldsymbol{e}_j$ are monomials in R^m, then we say that $\boldsymbol{n}$ divides $\boldsymbol{m}$ if and only if $i = j$ and x^β divides x^α, and we define the quotient

$\boldsymbol{m}/\boldsymbol{n} = x^\alpha/x^\beta = x^{\alpha-\beta} \in R$. We define the least common multiple of $\boldsymbol{m}$ and $\boldsymbol{n}$, $\mathrm{LCM}(\boldsymbol{m}, \boldsymbol{n})$, to be the least common multiple of x^α and x^β times $\boldsymbol{e}_i$ whenever $\boldsymbol{m}$ and $\boldsymbol{n}$ contains the same basis element $\boldsymbol{e}_i$, otherwise we define $\mathrm{LCM}(\boldsymbol{m}, \boldsymbol{n}) = \boldsymbol{0}$. The greatest common divisor, $\mathrm{GCD}(\boldsymbol{m}, \boldsymbol{n})$, is defined as the greatest common divisor of x^α and x^β times $\boldsymbol{e}_i$ if $\boldsymbol{m}$ and $\boldsymbol{n}$ contain the same standard basis element $\boldsymbol{e}_i$, otherwise $\mathrm{GCD}(\boldsymbol{m}, \boldsymbol{n}) = \boldsymbol{0}$.

If a submodule $M \subseteq R^m$ can be generated by a set of monomials, we say that M is a monomial submodule.

Definition 1.2.2 (Monomial Submodules)
A submodule $M \subseteq R^m$ is called a monomial submodule if it can be generated by monomials. That is, if there exists a subset $A \subseteq \mathbb{N}_0^n$ such that every element of M can be written in the form $\sum_{\alpha \in A, 1 \leq i \leq m} h_\alpha x^\alpha \boldsymbol{e}_i$, where $h_\alpha \in R$.

Monomial submodules are closely related to monomial ideals as we will show in the next proposition, but first we need a few lemmas.

Lemma 1.2.3
Let $M \subseteq R^m$ be a monomial submodule. A monomial $x^\beta \boldsymbol{e}_j$, $\beta \in \mathbb{N}_0^n$, lies in M if and only if $x^\beta \boldsymbol{e}_j$ is divisible by $x^\alpha \boldsymbol{e}_i$ for some $\alpha \in A$.

Proof
Note that for $x^\beta \boldsymbol{e}_j$ to be divisible by some $x^\alpha \boldsymbol{e}_i$, we need to have $i = j$. Hence, we will only consider monomials $x^\alpha \boldsymbol{e}_i$ where $i = j$. Now, if x^β is divisible by some x^α, then $x^\beta \boldsymbol{e}_j \in M$ by definition.

Assume that $x^\beta \boldsymbol{e}_j \in M$. We can write x^β as $\sum_{\alpha \in A} h_\alpha x^\alpha$, where $h_\alpha \in R$. Expand the right-hand side as a linear combination of monomials. The left-hand side consists only of a single monomial x^β so all the monomials on the right-hand side with a degree different from x^β must cancel out. This means that the right-hand side can be written as the sum $\sum_{\alpha \in A} h'_\alpha x^\alpha$, $h'_\alpha \in R$. Now, $h'_\alpha x^\alpha = c_\alpha x^\beta$, where $c_\alpha \in k, c_\alpha \neq 0$, for some α and thus $x^\beta = (c_\alpha^{-1} h'_\alpha) x^\alpha$, which shows that x^α divides x^β. ■

We will also give Dickson's Lemma for ideals, since we will need this in the proof for the following proposition.

Lemma 1.2.4 (Dickson's Lemma)
Let $I = \langle x^\alpha | \alpha \in A \rangle \subseteq k[x_1, \ldots, x_n]$, where A is a subset of $\mathbb{N}_0^n$, be a monomial ideal. Then I can be written in the form $I = \langle x^{\alpha(1)}, \ldots, x^{\alpha(s)} \rangle$, where

$\alpha(1), \ldots, \alpha(s) \in A \subseteq \mathbb{N}_0^n$. *That is, I can be finitely generated.*

Proof

We will prove this by induction. For $n = 1$, I is generated by the monomials $\{x^\alpha | \alpha \in A \subseteq \mathbb{N}_0\}$. Let β be the smallest element of A, such that $\beta \leq \alpha$ for all $\alpha \in A$. Every x^α can be divided by x^β, and it follows that I can be generated by x^β, $I = \langle x^\beta \rangle$.

Now, suppose the theorem is true for $n - 1, n > 3$. As the nth variable we will be using y to make it more clear. The monomials in $k[x_1, \ldots, x_{n-1}, y]$ can then be written as $x^\alpha y^m$, where $\alpha \in \mathbb{N}_0^{n-1}$ and $m \in \mathbb{N}_0$. We want to find generators for the monomial ideal $I \subseteq k[x_1, \ldots, x_{n-1}, y]$. Let $J = \langle x^\alpha | x^\alpha y^m \in I, m \geq 0 \rangle$ be the monomial ideal in $k[x_1, \ldots, k_{n-1}]$ where $x^\alpha y^m \in I$ for some $m \geq 0$. The induction hypothesis implies that J must be finitely generated, say, $J = \langle x^{\alpha(1)}, \ldots, x^{\alpha(s)} \rangle$. J can be understood as the projection of I: $k[x_1, \ldots, x_{n-1}, y] \to k[x_1, \ldots, x_{n-1}]$. By definition $x^{\alpha(i)} y^{m_i} \in I$ for $1 \geq i \geq s$. Let $m = \max\{m_1, \ldots, m_s\}$. Next, define the "slices" J_l of I generated by the monomials x^β such that $x^\beta y^l \in I$, and consider the list

$$
\begin{aligned}
J_0 &: x^{\alpha_0(1)}, \ldots, x^{\alpha_0(s_0)}, \\
J_1 &: x^{\alpha_1(1)} y, \ldots, x^{\alpha_1(s_1)} y, \\
&\vdots \\
J_{m-1} &: x^{\alpha_{m-1}(1)} y^{m-1}, \ldots, x^{\alpha_{m-1}(s_{m-1})} y^{m-1}, \\
J = J_m &: x^{\alpha(1)} y^m, \ldots, x^{\alpha(s)} y^m.
\end{aligned}
$$

By the induction hypothesis, every J_l has a finite generating set, say, $J_l = \langle x^{\alpha_l(1)}, \ldots, x^{\alpha_l(s_l)} \rangle$. The claim is that I is generated by the monomials in the above list; that is, $I = \langle J_0 \cup J_0 y \cup \cdots \cup J_{m-1} y^{m-1} \cup J y^m \rangle$.

Every monomial in I is divisible by a monomial in the list. To see this, let $x^\alpha y^p \in I$. If $p \geq m$, then $x^\alpha y^p$ is divisible by some $x^{\alpha(i)} y^m$ by the construction of J. If $p < m$, then $x^\alpha y^p$ will be divisible by some $x^{\alpha_p(i)} y^p$ by the construction of J_p. Thus, the above monomials generate an ideal having the same monomials as I, and, therefore, these ideals must be the same.

To finish the proof, we switch back to using the variables $x_1, \ldots, x_n$, such that $I = \langle x^\alpha | \alpha \in A \rangle \subseteq k[x_1, \ldots, x_n]$. What we need to show is that the finite set of generators can be chosen from the list x^α, $\alpha \in A$. We have already showed that $I = \langle x^{\beta(1)}, \ldots, x^{\beta(s)} \rangle$ for some $x^{\beta(i)} \in I$. It follows that each $x^{\beta(i)}$ is divisible by some $x^{\alpha(i)}$. Thus, we can interchange $x^{\beta(i)}$ with $x^{\alpha(i)}$. If we do this for every generator of $I = \langle x^{\beta(1)}, \ldots, x^{\beta(s)} \rangle$ we end up with $I = \langle x^{\alpha(1)}, \ldots, x^{\alpha(s)} \rangle$, which is the desired form. ■

We are now ready to state the next proposition, which says that every monomial submodule can be finitely generated.

Proposition 1.2.5
Let $M \subseteq R^m$ be a monomial submodule.

(i). Let $\boldsymbol{f} \in R^m$. Then $\boldsymbol{f} \in M$ if and only if every term of $\boldsymbol{f}$ lies in M.

(ii). Every monomial submodule of R^m is generated by a finite set of monomials.

(iii). Every infinite ascending chain $M_1 \subseteq M_2 \subseteq \dots$ of monomial submodules of R^m must stabilize. That is, there exists an N such that $M_N = M_{n+1} = \cdots = M_{N+l} = \dots$ for all $l \geq 0$.

(iv). Let $\{\boldsymbol{m}_1, \dots, \boldsymbol{m}_t\}$ be a set of monomial generators for M, and let $\epsilon_1, \dots, \epsilon_t$ denote the standard basis vectors in R^t. Let $\boldsymbol{m}_{ij} = \mathrm{LCM}(\boldsymbol{m}_i, \boldsymbol{m}_j)$. The syzygy module $\mathrm{Syz}(\boldsymbol{m}_1, \dots, \boldsymbol{m}_t)$ is generated by the syzygies $\sigma_{ij} = (\boldsymbol{m}_{ij}/\boldsymbol{m}_i)\epsilon_i - (\boldsymbol{m}_{ij}/\boldsymbol{m}_j)\epsilon_j$, for all $1 \leq i < j \leq t$ ($\sigma_{ij} = 0$ unless $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$ contain the same standard basis vector in R^m).

Proof
If every term of $\boldsymbol{f}$ lies in M, then $\boldsymbol{f} \in M$ by definition. Now, suppose $\boldsymbol{f} \in M$. Then $\boldsymbol{f}$ can be written as a linear combination of monomials $\boldsymbol{f} = \sum_{\alpha \in A, i} h_\alpha x^\alpha \boldsymbol{e}_i$, $h_\alpha \in R$. Expand the right-hand side of this equation as we did in the proof for Lemma 1.2.3 such that $\boldsymbol{f}$ is k-linear combination of monomials. We need to show that these monomials lie in M. Since all of the monomials by construction are multiples of some $x^\alpha \boldsymbol{e}_i$, $\alpha \in A$, it follows from Lemma 1.2.3 that they all lie in M, and, thus, every term of $\boldsymbol{f}$ lies in M.

For part (ii) let M be a monomial submodule of R^m. Let $M_i = M \cap R\boldsymbol{e}_i$ for each i, which is also a monomial submodule of R^m. To see this, let $\boldsymbol{f}, \boldsymbol{g} \in M_i$. Then $\boldsymbol{f} = f\boldsymbol{e}_i, \boldsymbol{g} = g\boldsymbol{e}_i \in M$. Since M is a submodule of R^m it satisfies the module conditions, and since addition and scalar multiplication do not change the standard basis $\boldsymbol{e}_i$, the module conditions are also satisfied for M_i. Every element of M_i is of the form $f\boldsymbol{e}_i$, which means that we can write $M_i = I_i\boldsymbol{e}_i$ for some monomial ideal $I_i \subseteq R$. By Dickson's Lemma the monomial ideal I_i can be generated by a finite set of monomials $x^{\alpha(i1)}, \dots, x^{\alpha(id_i)}$. Now, since

$$x^{\alpha(11)}\boldsymbol{e}_1, \dots, x^{\alpha(1d_1)}\boldsymbol{e}_1,$$

$$\vdots$$
$$x^{\alpha(m1)}\boldsymbol{e}_m, \ldots, x^{\alpha(md_m)}\boldsymbol{e}_m$$

generate M, it follows that M can be generated by a finite set of monomials.

For part (iii) let $M = \bigcup_{i=1}^{\infty} M_i$, which is also a monomial submodule of R^m. To see this, first note that the zero element is in every M_i and, thus, also in M. Let $\boldsymbol{f}, \boldsymbol{g} \in M$, such that $\boldsymbol{f} \in M_i$ and $\boldsymbol{g} \in M_j$ for some i, j. Since the submodules form an ascending chain we can assume that $M_i \subseteq M_j$, and, thus $\boldsymbol{f} \in M_j$. Since M_j is a submodule and, thus, satisfies the module conditions, and since $\boldsymbol{f}, \boldsymbol{g}$ was chosen arbitrarily, the module conditions are also satisfied by M. Then by part (ii) M has a finite generating set, say $\{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_t\}$. For some $i \in \{1, \ldots, t\}$ we must have $\boldsymbol{m}_i \in M_j$. Let M_{j_i} denote the smallest submodule containing $\boldsymbol{m}_i$, $\boldsymbol{m}_i \in M_{j_i}$. Hence, we have $\boldsymbol{m}_1 \in M_{j_1}, \ldots, \boldsymbol{m}_t \in M_{j_t}$. Let $N = \max\{j_1, \ldots, j_t\}$ such that $M_N = \langle \boldsymbol{m}_1, \ldots, \boldsymbol{m}_t \rangle$. Then

$$\langle \boldsymbol{m}_1, \ldots, \boldsymbol{m}_t \rangle = M_N \subseteq M_{N+1} \subseteq \cdots \subseteq M = \langle \boldsymbol{m}_1, \ldots, \boldsymbol{m}_t \rangle.$$

Hence, an infinite ascending chain of submodules will stabilize.

To prove part (iv) let $(a_1, \ldots, a_t)^T$ be a syzygy on a set of monomials $(m_1, \ldots, m_t)$, such that

$$\mathbf{0} = a_1\boldsymbol{m}_1 + \cdots + a_t\boldsymbol{m}_t.$$

Consider the expansion of this expression in terms of the standard basis in R^m,

$$\mathbf{0} = f_1\boldsymbol{e}_1 + \cdots + f_m\boldsymbol{e}_m.$$

We note that we must have $f_1 = \cdots = f_n = 0$, and, thus, we can split up the syzygy $(a_1, \ldots, a_t)^T$ in subsets of the monomials containing $\boldsymbol{e}_i$ for each i. Let $\{\boldsymbol{n}_1, \ldots, \boldsymbol{n}_s\} \subseteq \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_t\}$ be the monomials containing $\boldsymbol{e}_i$ for some i,

$$\boldsymbol{n}_1 = x^{\alpha_1}\boldsymbol{e}_i, \ldots, \boldsymbol{n}_s = x^{\alpha_s}\boldsymbol{e}_i.$$

If $(b_1, \ldots, b_s)^T$ is a syzygy of $\{\boldsymbol{n}_1, \ldots, \boldsymbol{n}_s\}$, then $\mathbf{0} = b_1\boldsymbol{n}_1 + \cdots + b_s\boldsymbol{n}_s$ or, equivalently,

$$0 = b_1x^{\alpha_1} + \cdots + b_sx^{\alpha_s}.$$

The terms of this expression with the same multidegree must also sum up to zero, or, in other words, the coefficients of the terms with the same multidegree must sum to zero. Thus, we can split up the syzygy $\{\boldsymbol{n}_1, \ldots, \boldsymbol{n}_s\}$ in subsets

$$(c_1 x^{\alpha-\alpha_1}, \ldots, c_s x^{\alpha-\alpha_s})^T, \quad c_1, \ldots, c_s \in k,$$

where $c_1 + \cdots + c_s = 0$. This syzygy is called a homogeneous syzygy, and can also be split up in sets where all entries in the syzygy are zero except for two. To see this, consider an example where we let $s = 3$. Then a syzygy can be written as

$$(c_1 x^{\alpha-\alpha_1}, c_2 x^{\alpha-\alpha_2}, c_3 x^{\alpha-\alpha_3})^T$$

with $c_1 + c_2 + c_3 = 0$. We can split this syzygy as

$$((c_1 + c_3)x^{\alpha-\alpha_1}, c_2 x^{\alpha-\alpha_2}, 0)^T + (-c_3 x^{\alpha-\alpha_1}, 0, c_3 x^{\alpha-\alpha_3})^T.$$

We note that $((c_1+c_3)x^{\alpha-\alpha_1}, c_2 x^{\alpha-\alpha_2})^T = -c_2(x^{\alpha-\alpha_1}, -x^{\alpha-\alpha_2})^T$ is a syzygy on the pair $x^{\alpha_1}, x^{\alpha_2}$ and $(-c_3 x^{\alpha-\alpha_1}, c_3 x^{\alpha-\alpha_3})^T = -c_3(x^{\alpha-\alpha_1}, x^{\alpha-\alpha_3})^T$ is a syzygy on the pair $x^{\alpha_1}, x^{\alpha_3}$.

This splitting works for any s; that is, for any s every homogeneous syzygy can be written as syzygies between pairs of monomials. Let x^α, x^β be two monomials, and let x^γ be a multiple of these. Then $\mathrm{Syz}(x^\alpha, x^\beta) = (x^{\gamma-\alpha}, x^{\gamma-\beta})^T$ is a monomial times

$$\sigma = \left(\mathrm{LCM}(x^\alpha, x^\beta)/x^\alpha, \mathrm{LCM}(x^\alpha, x^\beta)/x^\beta\right)^T.$$

To sum up, we split the whole syzygy $\mathrm{Syz}(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_t)$ up in to syzygies $\mathrm{Syz}(0, \ldots, 0, c_\alpha x^\alpha, 0, \ldots, 0, c_\beta x^\beta, 0, \ldots, 0)\boldsymbol{e}_i$ for every i. If we let $\boldsymbol{m}_{ij} = \mathrm{LCM}(\boldsymbol{m}_i, \boldsymbol{m}_j)$, then these are all generated by

$$\sigma_{ij} = (\boldsymbol{m}_{ij}/\boldsymbol{m}_i)\epsilon_i - (\boldsymbol{m}_{ij}/\boldsymbol{m}_j)\epsilon_j.$$

■

Note that in part (i) that for every term of $\boldsymbol{f}$ to lie in $M = \langle \boldsymbol{m}_1, \ldots, \boldsymbol{m}_t \rangle$, every term must be divisible by some $\boldsymbol{m}_i$. Hence, as with monomial ideals, the submodule membership problem is easy to solve for monomial submodules. Thus, we have $\boldsymbol{f} \in M$ if and only if the remainder of $\boldsymbol{f}$ on division by the basis of M is zero.

Just as with ideals, before introducing the Gröbner bases for modules we need to define a monomial ordering of the monomials in R^m, and we need a division algorithm on elements of R^m. We will then be able to extend Buchberger's Algorithm to R^m.

Definition 1.2.6 (Monomial Ordering)
A monomial ordering $\succeq$ on $R^m = (k[x_1, \ldots, x_n])^m$ is a total order on the set of monomials such that

(i). for every pair of monomials $\boldsymbol{m}, \boldsymbol{n} \in R^m$ with $\boldsymbol{m} \succeq \boldsymbol{n}$, we have $x^\alpha \boldsymbol{m} \succeq x^\alpha \boldsymbol{n}$ for every monomial $x^\alpha \in R$,

(ii). $\succeq$ is a well-ordering; that is, every nonempty subset has a smallest element under $\succeq$.

We will consider two different families of monomial orders on R^m here that are both an extension of the monomial orderings on R, namely the TOP extension (term-over-position) and the POT extension (position-over-term). Recall the lexicographic order for R:

Definition 1.2.7 (Lexicographic Order for $x_1 > \cdots > x_n$ ($\succeq_{\text{lex}}$))
Let $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_n)$ in $\mathbb{N}_0^n$. We say that $\alpha \succeq_{\text{lex}} \beta$ in $\mathbb{N}_0^n$ and $x^\alpha \succeq_{\text{lex}} x^\beta$ in $k[x_1, \ldots, x_n]$, when the leftmost nonzero entry in the vector difference $\alpha - \beta \in \mathbb{Z}^n$ is positive.

If not anything else is specified, we will use the extensions on this order.

Definition 1.2.8 (Monomial Orderings on R^m with $\boldsymbol{e}_1 > \boldsymbol{e}_2 > \ldots$)
Let $\succeq_R$ be any monomial order on R and $\boldsymbol{e}_i > \boldsymbol{e}_j$ whenever $i < j$.

(i). (TOP extension of $\succeq_R$ – $\succeq_{\text{TOP}}$) We say that $x^\alpha \boldsymbol{e}_i \succeq_{\text{TOP}} x^\beta \boldsymbol{e}_j$ if $x^\alpha \succeq_R x^\beta$, or if $x^\alpha = x^\beta$ and $i \leq j$.

(ii). (POT extension of $\succeq_R$ – $\succeq_{\text{POT}}$) We say that $x^\alpha \boldsymbol{e}_i \succeq_{\text{POT}} x^\beta \boldsymbol{e}_j$ if $i < j$, or if $i = j$ and $x^\alpha \succeq_R x^\beta$.

To see that the $\succeq_{\text{TOP}}$ order is a monomial order note that we have $x^\alpha \boldsymbol{e}_i \succeq_{\text{TOP}} x^\beta \boldsymbol{e}_j$ whenever $x^\alpha \succeq_R x^\beta$. Since $\succeq_R$ is a monomial order we also have $x^\gamma x^\alpha \succeq_R x^\gamma x^\beta$ for some $x^\gamma \in R$, which means that $x^\gamma x^\alpha \boldsymbol{e}_i \succeq_{\text{TOP}} x^\gamma x^\beta \boldsymbol{e}_j$. If $x^\alpha = x^\beta$, then the condition is clearly also satisfied, since we only consider the standard basis vectors in this situation. For the $\succeq_{\text{POT}}$ order we have

$x^\alpha \boldsymbol{e}_i \succeq_{\mathrm{POT}} x^\beta \boldsymbol{e}_j$ whenever $i < j$, which is the same situation as that just discussed for $\succeq_{\mathrm{TOP}}$. If $i = j$, then $x^\alpha \boldsymbol{e}_i \succeq_{\mathrm{POT}} x^\beta \boldsymbol{e}_j$ whenever $x^\alpha \succeq_R x^\beta$, which we can argue the same way as we did with $\succeq_{\mathrm{TOP}}$. The well-ordering of both $\succeq_{\mathrm{TOP}}$ and $\succeq_{\mathrm{POT}}$ follows directly from $\succeq_R$ being a well-ordering.

Example 1.2.9
Let $\succeq_R$ be the lexicographic order $\succeq_{\mathrm{lex}}$ and consider the monomials from Example 1.2.1. With $\succeq_{\mathrm{POT}}$ we get the following monomial ordering:

$$\begin{bmatrix} x^2y^2 \\ 0 \\ 0 \end{bmatrix} \succeq_{\mathrm{POT}} \begin{bmatrix} y^2 \\ 0 \\ 0 \end{bmatrix} \succeq_{\mathrm{POT}} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \succeq_{\mathrm{POT}} \begin{bmatrix} 0 \\ x^4 \\ 0 \end{bmatrix} \succeq_{\mathrm{POT}} \begin{bmatrix} 0 \\ 0 \\ xy^2 \end{bmatrix} \succeq_{\mathrm{POT}} \begin{bmatrix} 0 \\ 0 \\ x \end{bmatrix}.$$

Now consider $\succeq_{\mathrm{TOP}}$,

$$\begin{bmatrix} 0 \\ x^4 \\ 0 \end{bmatrix} \succeq_{\mathrm{TOP}} \begin{bmatrix} x^2y^2 \\ 0 \\ 0 \end{bmatrix} \succeq_{\mathrm{TOP}} \begin{bmatrix} 0 \\ 0 \\ xy^2 \end{bmatrix} \succeq_{\mathrm{TOP}} \begin{bmatrix} 0 \\ 0 \\ x \end{bmatrix} \succeq_{\mathrm{TOP}} \begin{bmatrix} y^2 \\ 0 \\ 0 \end{bmatrix} \succeq_{\mathrm{TOP}} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

□

With a monomial ordering in place we can extend the definitions of the leading coefficient, leading monomial, and leading term of a polynomial to R^m.

Definition 1.2.10
Let $\succeq$ be a monomial ordering on R^m, and write $\boldsymbol{f} \in R^m$ as a sum of terms

$$\boldsymbol{f} = \sum_{i=1}^{t} c_i \boldsymbol{m}_i, \quad c_i \in k,$$

with $\boldsymbol{m}_1 \succeq \boldsymbol{m}_2 \succeq \ldots \succeq \boldsymbol{m}_t$. We define

$$\begin{aligned} \mathrm{LC}_\succeq(\boldsymbol{f}) &= c_1, \\ \mathrm{LM}_\succeq(\boldsymbol{f}) &= \boldsymbol{m}_1, \\ \mathrm{LT}_\succeq(\boldsymbol{f}) &= c_1\boldsymbol{m}_1, \\ \mathrm{multideg}_\succeq(\boldsymbol{f}) &= (\alpha_1, \ldots, \alpha_n) \subseteq \mathbb{N}_0^n \text{ if } \boldsymbol{m}_1 = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \boldsymbol{e}_i. \end{aligned}$$

We will give an example to show these.

Example 1.2.11
Let $\boldsymbol{f} \in (k[x,y])^3$ be given by

$$\boldsymbol{f} = \begin{bmatrix} 3x^2y^2 + y^2 + 2 \\ 2x^4 \\ xy^2 - 5x \end{bmatrix}.$$

Then by Definition 1.2.10 we have

$$\begin{aligned}
&\mathrm{LC}_{\succeq_{\mathrm{POT}}}(\boldsymbol{f}) = 3, && \mathrm{LC}_{\succeq_{\mathrm{TOP}}}(\boldsymbol{f}) = 2,\\
&\mathrm{LM}_{\succeq_{\mathrm{POT}}}(\boldsymbol{f}) = \begin{bmatrix} x^2y^2 \\ 0 \\ 0 \end{bmatrix}, && \mathrm{LM}_{\succeq_{\mathrm{TOP}}}(\boldsymbol{f}) = \begin{bmatrix} 0 \\ x^4 \\ 0 \end{bmatrix},\\
&\mathrm{LT}_{\succeq_{\mathrm{POT}}}(\boldsymbol{f}) = 3\begin{bmatrix} x^2y^2 \\ 0 \\ 0 \end{bmatrix}, && \mathrm{LT}_{\succeq_{\mathrm{TOP}}}(\boldsymbol{f}) = 2\begin{bmatrix} 0 \\ x^4 \\ 0 \end{bmatrix},\\
&\mathrm{multidegree}_{\succeq_{\mathrm{POT}}}(\boldsymbol{f}) = (2,2), && \mathrm{multidegree}_{\succeq_{\mathrm{TOP}}}(\boldsymbol{f}) = (4,0).
\end{aligned}$$

If we want to use these monomial orders in Singular we do as follows.

```
>ring R1=0,(x,y),(c,lp); //POT order over lex
>vector f=[3x2y2+y2+2,2x4,xy2-5x];
>lead(f); //LT(f)
[3x2y2]
>ring R2=0,(x,y),(lp,c); //TOP order over lex
>vector f=imap(R1,f);
>lead(f);
2x4*gen(2)
```

Note that if the last entries in the vector is zero, then Singular does not write these. Also note that with the $\succeq_{\mathrm{TOP}}$ order Singular uses $\mathrm{gen} * (2)$ which is the standard basis $\boldsymbol{e}_2$. If we used print(lead(f)) we would get [0,2x4]. □

We can now introduce the Division Algorithm for R^m.

Theorem 1.2.12 (Division Algorithm for R^m)
Fix a monomial ordering on R^m and let $F = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s)$ be an ordered s-tuple of elements of R^m. Then every $\boldsymbol{f} \in R^m$ can be written as

$$\boldsymbol{f} = a_1\boldsymbol{f}_1 + \cdots + a_s\boldsymbol{f}_s + \boldsymbol{r}, \tag{1.3}$$

where $a_i \in R$, $\boldsymbol{r} \in R^m$, $\mathrm{LT}(a_i\boldsymbol{f}_i) \leq \mathrm{LT}(\boldsymbol{f})$ for all i. Furthermore, either $\boldsymbol{r} = \boldsymbol{0}$ or $\boldsymbol{r}$ is a k-linear combination of monomials none of which is divisible

by any of $\mathrm{LM}(\boldsymbol{f}_1), \ldots, \mathrm{LM}(\boldsymbol{f}_s)$*, and* $\boldsymbol{r}$ *is called the remainder of* $\boldsymbol{f}$ *on division by* F*.*

Algorithm 1.2.13 (Division Algorithm)
Input
$\boldsymbol{f} \in R^m$
$\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s \in R^m$

Output
$a_1, \ldots, a_s \in R$ for (1.3)
$\boldsymbol{r} \in R^m$ for (1.3)

Initialize
$a_1 := 0, \ldots, a_s := 0$
$\boldsymbol{r} := \boldsymbol{0}$
$\boldsymbol{p} := \boldsymbol{f}$

Loop
WHILE $\boldsymbol{p} \neq \boldsymbol{0}$ DO
    $i := 1$
    divisionoccurred := false
    WHILE $i \leq s$ AND divisionoccurred = false DO
        IF $\mathrm{LT}(\boldsymbol{f}_i)$ divides $\mathrm{LT}(\boldsymbol{p})$ THEN
            $a_i := a_i + \mathrm{LT}(\boldsymbol{p})/\mathrm{LT}(\boldsymbol{f}_i)$
            $\boldsymbol{p} := \boldsymbol{p} - (\mathrm{LT}(\boldsymbol{p})/\mathrm{LT}(\boldsymbol{f}_i))\boldsymbol{f}_i$
            divisionoccurred := true
        ELSE
            $i := i + 1$
    IF divisionoccurred=false THEN
        $\boldsymbol{r} := \boldsymbol{r} + \mathrm{LT}(\boldsymbol{p})$
        $\boldsymbol{p} := \boldsymbol{p} - \mathrm{LT}(\boldsymbol{p})$

Proof
To prove that every $\boldsymbol{f} \in R^m$ can be written in the form

$$\boldsymbol{f} = a_1\boldsymbol{f}_1 + \cdots + a_s\boldsymbol{f}_s + \boldsymbol{p} + \boldsymbol{r},$$

where $\boldsymbol{p}$ is defined as in the algorithm, by using the Division Algorithm, we will show that this holds at every step of the algorithm. We will prove this by induction. With the initial values where $a_1, \ldots, a_s, \boldsymbol{r}$ are all zero, this is clearly true. Now, suppose it is true for some step in the algorithm. Two things can occur in the next step. If it is a division step, then some $\mathrm{LT}(\boldsymbol{f}_i)$

divides $\mathrm{LT}(\boldsymbol{p})$ and the equality

$$a_i \boldsymbol{f}_i + \boldsymbol{p} = (a_i + \mathrm{LT}(\boldsymbol{p})/\mathrm{LT}(\boldsymbol{f}_i))\boldsymbol{f}_i + (\boldsymbol{p} - (\mathrm{LT}(\boldsymbol{p})/\mathrm{LT}(\boldsymbol{f}_i))\boldsymbol{f}_i)$$

shows that the value does not change, and, thus, $\boldsymbol{f}$ can still be written in the desired form. If the next step is not a division step, then it is a remainder step, and both $\boldsymbol{p}$ and $\boldsymbol{r}$ will be changed, while the sum $\boldsymbol{p} + \boldsymbol{r}$ will stay the same, since

$$\boldsymbol{p} + \boldsymbol{r} = (\boldsymbol{p} - \mathrm{LT}(\boldsymbol{p})) + (\boldsymbol{r} + \mathrm{LT}(\boldsymbol{p})),$$

and $\boldsymbol{f}$ still has the desired form. The algorithm will terminate when $\boldsymbol{p} = \boldsymbol{0}$, and at that point $\boldsymbol{f}$ is in the form of Equation (1.3). In the algorithm we only add terms to $\boldsymbol{r}$ when they are divisible by none of the $\mathrm{LT}(\boldsymbol{f}_i)$, so $a_1, \ldots, a_s, \boldsymbol{r}$ all have the desired properties when the algorithm terminates.

We need to show that the algorithm eventually terminates. This follows from the fact that the multidegree of $\boldsymbol{p}$ drops at each step of the algorithm. In a division step $\boldsymbol{p}$ is redefined as

$$\boldsymbol{p}' = \boldsymbol{p} - \frac{\mathrm{LT}(\boldsymbol{p})}{\mathrm{LT}(\boldsymbol{f}_i)}\boldsymbol{f}_i.$$

Since

$$\begin{aligned}
\mathrm{LT}(\boldsymbol{p}') &= \mathrm{LT}(\boldsymbol{p}) - \mathrm{LT}\left(\frac{\mathrm{LT}(\boldsymbol{p})}{\mathrm{LT}(\boldsymbol{f}_i)}\boldsymbol{f}_i\right) \\
&= \mathrm{LT}(\boldsymbol{p}) - \frac{\mathrm{LT}(\boldsymbol{p})}{\mathrm{LT}(\boldsymbol{f}_i)}\mathrm{LT}(\boldsymbol{f}_i) \\
&= \boldsymbol{0},
\end{aligned}$$

the leading term of p is canceled, and, thus, the multidegree must drop. If the step is a remainder step, then $\boldsymbol{p}$ is redefied as

$$\boldsymbol{p}' = \boldsymbol{p} - \mathrm{LT}(\boldsymbol{p}).$$

Clearly, the leading term is also canceled in this situation, and the multidegree must drop. Now, since $\succeq$ is a well-ordering, it follows that the multidegree of $\boldsymbol{p}$ must eventually be zero where the algorithm terminates.

The last thing we need to proof is that $\mathrm{LT}(a_i \boldsymbol{f}_i) \leq \mathrm{LT}(\boldsymbol{f})$ for all i. Since every term of a_i is of the form $\mathrm{LT}(\boldsymbol{p})/\mathrm{LT}(\boldsymbol{f}_i)$ for some value of $\boldsymbol{p}$, we have $\mathrm{LT}(\boldsymbol{p}) = \mathrm{LT}(a_i \boldsymbol{f}_i)$, but since the multidegree of $\boldsymbol{p}$ drops, we must have $\mathrm{LT}(a_i \boldsymbol{f}_i) = \mathrm{LT}(\boldsymbol{p}) \leq \mathrm{LT}(\boldsymbol{f})$, where equality is true with the initial value $\boldsymbol{f} = \boldsymbol{p}$. ■

To show how the Division Algorithm works we will show an example.

Example 1.2.14
Let

$$\boldsymbol{f} = (5xy^2 - y^{10} + 3, 4x^3 + 2y, 16x)^T \in (k[x,y])^3,$$

and let

$$\begin{aligned}\boldsymbol{f}_1 &= (xy + 4x, 0, y^2)^T,\\ \boldsymbol{f}_2 &= (0, y-1, x-2)^T.\end{aligned}$$

We will use the $\succeq_{\mathrm{POT}}$ order and divide $\boldsymbol{f}$ with $(\boldsymbol{f}_1, \boldsymbol{f}_2)$.

Step 1: We write our initial values:

$$\begin{aligned}a_1 &:= 0,\\ a_2 &:= 0,\\ \boldsymbol{r} &:= \mathbf{0},\\ \boldsymbol{p} &:= \boldsymbol{f} = (5xy^2 - y^{10} + 3, 4x^3 + 2y, 16x)^T.\end{aligned}$$

Step 2: Notice that $\mathrm{LT}(\boldsymbol{f}_1) = xy\boldsymbol{e}_1$ divides $\mathrm{LT}(p) = 5xy^2\boldsymbol{e}_1$, and that $\mathrm{LT}(p)/\mathrm{LT}(f_1) = 5y$, so

$$\begin{aligned}a_1 &:= 5y,\\ a_2 &:= 0,\\ \boldsymbol{r} &:= \mathbf{0},\\ \boldsymbol{p} &:= (5xy^2 - y^{10} + 3, 4x^3 + 2y, 16x)^T - (4y)(xy + 4x, 0, y^2)^T\\ &= (-20xy - y^{10} + 3, 4x^3 + 2y, 16x - 5y^3)^T.\end{aligned}$$

Step 3: Now $\mathrm{LT}(\boldsymbol{f}_1) = xy\boldsymbol{e}_1$ still divides $\mathrm{LT}(p) = -20xy\boldsymbol{e}_1$, and that $\mathrm{LT}(p)/\mathrm{LT}(f_1) = -20$, so

$$\begin{aligned}a_1 &:= 5y - 20,\\ a_2 &:= 0,\\ \boldsymbol{r} &:= \mathbf{0},\\ \boldsymbol{p} &:= (-20xy - y^{10} + 3, 4x^3 + 2y, 16x - 5y^2)^T - (-20)(xy + 4x, 0, y^2)^T\\ &= (80x - y^{10} + 3, 4x^3 + 2y, 16x - 5y^3 + 20y^2)^T.\end{aligned}$$

Step 4-7: Neither $\mathrm{LT}(\boldsymbol{f}_1) = xy\boldsymbol{e}_1$ nor $\mathrm{LT}(\boldsymbol{f}_2) = y\boldsymbol{e}_2$ divides $\mathrm{LT}(\boldsymbol{f}) = 80x\boldsymbol{e}_1$, so $80x\boldsymbol{e}_1$ goes to the remainder. Note that the next few steps are remainder steps, so we will skip these.

$$\begin{aligned} a_1 &:= 5y - 20, \\ a_2 &:= 0, \\ \boldsymbol{r} &:= (80x - y^{10} + 3, 4x^3, 0)^T, \\ \boldsymbol{p} &:= (0, 2y, 16x - 5y^3 + 20y^2)^T. \end{aligned}$$

Step 8: Now $\mathrm{LT}(\boldsymbol{f}_1) = xy\boldsymbol{e}_1$ does not divide $\mathrm{LT}(p) = 2y\boldsymbol{e}_2$, but $\mathrm{LT}(\boldsymbol{f}_2) = y\boldsymbol{e}_2$ does, so

$$\begin{aligned} a_1 &:= 5y - 20, \\ a_2 &:= 2, \\ \boldsymbol{r} &:= (80x - y^{10} + 3, 4x^3, 0)^T, \\ \boldsymbol{p} &:= (0, 2y, 16x - 5y^3 + 20y^2)^T - (2)(0, y - 1, x - 2)^T \\ &= (0, 2, 14x - 5y^3 + 20y^2 + 4)^T. \end{aligned}$$

The last steps are all remainder steps, so

$$\begin{aligned} a_1 &:= 5y - 20, \\ a_2 &:= 2, \\ \boldsymbol{r} &:= (80x - y^{10} + 3, 4x^3 + 2, 14x - 5y^3 + 20y^2 + 4)^T, \\ \boldsymbol{p} &:= (0, 0, 0)^T \end{aligned}$$

We conclude that we can write $\boldsymbol{f}$ as

$$\boldsymbol{f} = (5y - 20)\boldsymbol{f}_2 + 2\boldsymbol{f}_2 + (80x - y^{10} + 3, 4x^3 + 2, 14x - 5y^3 + 20y^2 + 4)^T.$$

We can also use Singular to perform the divisions. We will need to define the 2-tuple $(\boldsymbol{f}_1, \boldsymbol{f}_2)$ as a module in Singular.

```
>ring R=0,(x,y),(c,lp);
>vector f=[5xy2-y10+3,4x3+2y,16x];
>vector f1=[xy+4x,0,y2];
>vector f2=[0,y-1,x-2];
>module M=f1,f2;
>division(f,M);
[1]:
   _[1,1]=5y-20
```

```
   _[1,2]=2
[2]:
   _[1]=[80x-y10+3,4x3+2,14x-5y3+20y2+4]
[3]:
   _[1,1]=1
```

[1]_[1,1] is a_1, [1]_[1,2] is a_2, and [2]_[1] is the remainder $\boldsymbol{r}$. [3] is some number we need to multiply with $\boldsymbol{f}$, such that $[3] \cdot \boldsymbol{f} = a_1\boldsymbol{f}_1 + a_2\boldsymbol{f}_2 + \boldsymbol{r}$. Usually, [3] is just 1. □

We are now ready to define Gröbner bases for modules.

Definition 1.2.15 (Gröbner Bases)
Let $M \subseteq R^m$ be a submodule, and let $\succeq$ be a monomial ordering.

(i). Denote by $\langle \mathrm{LT}(M) \rangle$ the monomial submodule generated by he leading terms of all $\boldsymbol{f} \in M$ with respect to $\succeq$.

(ii). A finite set $G = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s\} \subseteq M$ is called a Gröbner basis for M if $\langle \mathrm{LT}(M) \rangle = \langle \mathrm{LT}(\boldsymbol{g}_1), \ldots, \mathrm{LT}(\boldsymbol{g}_s) \rangle$.

We will later show that every submodule has a Gröbner basis. First we will show how to find a Gröbner basis in Singular.

Example 1.2.16
We will use Singular to calculate a Gröbner basis for the submodule $M = \langle \boldsymbol{f}_1, \boldsymbol{f}_2 \rangle$, where

$$\boldsymbol{f}_1 = (xy + 4x, 0, y^2)^T,$$
$$\boldsymbol{f}_2 = (0, y - 1, x - 2)^T.$$

We will use the $\succeq_{\mathrm{POT}}$ order.

```
>ring R=0,(x,y),(c,lp);
>vector f1=[xy+4x,0,y2];
>vector f2=[0,y-1,x-2];
>module M=f1,f2;
>std(M);
_[1]=[0,y-1,x-2]
_[2]=[xy+4x,0,y2]
```

We see that $(\boldsymbol{f}_1, \boldsymbol{f}_2)$ is already a Gröbner basis for M.

As another example, consider the same module, but let us calculate a Gröbner basis with respect to $\succeq_{\mathrm{TOP}}$.

```
>ring R=0,(x,y),(lp,c);
>vector f1=[xy+4x,0,y2];
>vector f2=[0,y-1,x-2];
>module M=f1,f2;
>std(M);
_[1]=x*gen(3)+y*gen(2)-gen(2)-2*gen(3)
_[2]=xy*gen(1)+4x*gen(1)+y2*gen(3)
```

Thus, a Gröbner basis for M with respect to $\succeq_{\mathrm{TOP}}$ is also given by $(\boldsymbol{f}_1, \boldsymbol{f}_2)$. □

Just like with ideals, then the remainder on division of some $\boldsymbol{f} \in R^m$ by a Gröbner basis is uniquely determined, as we will proof in the next proposition.

Proposition 1.2.17
Let $M = \langle \boldsymbol{g}_1, \ldots, \boldsymbol{g}_s \rangle \subseteq R^m$ be a submodule generated by the Gröbner basis $G = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s\} \subseteq M$, and let $\boldsymbol{f} \in R^m$. Then the remainder $\boldsymbol{r} \in R^m$ on division of $\boldsymbol{f}$ by G is uniquely determined.

Proof
The Division Algorithm gives

$$\boldsymbol{f} = a_1\boldsymbol{g}_1 + \cdots + a_s\boldsymbol{g}_s + \boldsymbol{r},$$

where no term of $\boldsymbol{r}$ is divisible be any $\mathrm{LT}(\boldsymbol{g}_i)$. Define $\boldsymbol{g} = a_1\boldsymbol{g}_1 + \cdots + a_s\boldsymbol{g}_s \in M$. We will prove that $\boldsymbol{f}$ can be uniquely written as $\boldsymbol{f} = \boldsymbol{g} + \boldsymbol{r}$; that is, the remainder is uniquely determined, $\boldsymbol{r} = \boldsymbol{f} - \boldsymbol{g}$.

Suppose $\boldsymbol{f}$ can also be written as $\boldsymbol{f} = \boldsymbol{g}' + \boldsymbol{r}'$, where $\boldsymbol{r} \neq \boldsymbol{r}'$. Then $\boldsymbol{r} - \boldsymbol{r}' = \boldsymbol{g}' - \boldsymbol{g} \in M$, and $\mathrm{LT}(\boldsymbol{r} - \boldsymbol{r}') \in \langle \mathrm{LT}(M) \rangle = \langle \mathrm{LT}(\boldsymbol{g}_1), \ldots, \mathrm{LT}(\boldsymbol{g}_s) \rangle$ by the definition of Gröbner bases. This means that $\mathrm{LT}(\boldsymbol{r} - \boldsymbol{r}')$ is divisible by some $\mathrm{LT}(\boldsymbol{g}_i)$, but this cannot be true because of the definition of a remainder. Thus, $\boldsymbol{r} - \boldsymbol{r}' = \boldsymbol{0}$, and it follows that $\boldsymbol{r}$ must be uniquely determined. ■

As we can easily solve the ideal membership problem with Gröbner bases, we can immediately solve the module membership problem with the use of Gröbner bases.

Proposition 1.2.18
Let G be a Gröbner basis for a submodule $M \subseteq R^m$, and let $\boldsymbol{f} \in M$. Then $\boldsymbol{f} \in M$ if and only if the remainder on division by G is zero.

Proof
If the remainder on division of $\boldsymbol{f}$ by G is zero, then by definition $\boldsymbol{f} \in M$. Conversely, suppose $\boldsymbol{f} \in M$. Then $\boldsymbol{f}$ can be written $\boldsymbol{f} = \boldsymbol{f} + \boldsymbol{0}$. Since this expression is unique by Proposition 1.2.17, it follows that the remainder of $\boldsymbol{f}$ on division by G is zero. ■

It follows that in Example 1.2.14, $\boldsymbol{f} = (5xy^2 - y^{10} + 3, 4x^3 + 2y, 16x)^T \notin \langle \boldsymbol{f}_1, \boldsymbol{f}_2 \rangle$, where $\boldsymbol{f}_1 = (xy + 4x, 0, y^2)^T$ and $\boldsymbol{f}_2 = (0, y - 1, x - 2)^T$, since we saw in Example 1.2.16 that $(\boldsymbol{f}_1, \boldsymbol{f}_2)$ is a Gröbner basis for $\langle \boldsymbol{f}_1, \boldsymbol{f}_2 \rangle$.

It is important to note that a Gröbner basis does not need to be a module basis; that is, the set of generators that is the Gröbner basis do not need to be linearly independent. However, as with ideals, Gröbner bases do exist for all submodules of R^m.

Theorem 1.2.19
Let $M \subseteq R^m$ be a submodule, and fix a monomial order $\succeq$. Then M has a Gröbner basis with respect to this monomial order.

Proof
Consider the monomial submodule $\langle \mathrm{LT}(M) \rangle$. By Proposition 1.2.5 $\langle \mathrm{LT}(M) \rangle$ can be generated by a finite number of generators. That is, there exists a set $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s \in M$ such that $\langle \mathrm{LT}(M) \rangle = \langle \mathrm{LT}(\boldsymbol{g}_1), \ldots, \mathrm{LT}(\boldsymbol{g}_s) \rangle$. It follows that $\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s\} \subseteq M$ is a Gröbner basis for M. ■

An application of Gröbner bases is the fact that any infinite ascending chain of submodules must eventually stabilize.

Theorem 1.2.20 (The Ascending Chain Condition)
Let $M_1 \subseteq M_2 \subseteq \ldots$ be an ascending chain of submodules over R^m. Then the chain will stabilize for some $N \geq 1$; that is, there exists an $N \geq 1$ such that

$$\cdots \subseteq M_{N-1} \subseteq M_N = M_{N+1} = M_{N+2} = \ldots.$$

Proof
We already proved this for monomial submodules in Proposition 1.2.5 on page 23. In that proof we only used the monomial property to say that every submodule is finitely generated. By Theorem 1.2.19 every submodule of R^m

has a Gröbner basis, which is a finite generating set. The theorem follows directly by combining these two facts, since at every expansion $M_{i+1} \supseteq M_i$ we have at least one $\boldsymbol{g} \in M_{i+1}$, where $\mathrm{LT}(\boldsymbol{g})$ is not divisible by any $\mathrm{LT}(\boldsymbol{g}_i)$, where $\boldsymbol{g}_i$ are the generators in the Gröbner basis for M_i. But this means that $\langle \mathrm{LT}(M) \rangle$ is growing, which we have shown will eventually stabilize. ■

To be able to calculate a Gröbner basis for a submodule we need a way to determine when a basis if a Gröbner basis. As with ideals this is done with the so-called $\boldsymbol{S}$-elements.

Definition 1.2.21 (The $\boldsymbol{S}$-element)
Fix a monomial ordering on R^m, and let $\boldsymbol{f}, \boldsymbol{g} \in R^m$. The $\boldsymbol{S}$-element of $\boldsymbol{f}$ and $\boldsymbol{g}$, denoted $\boldsymbol{S}(\boldsymbol{f}, \boldsymbol{g})$, is the following element of R^m. Let $\boldsymbol{m} = \mathrm{LCM}(\mathrm{LT}(\boldsymbol{f}), \mathrm{LT}(\boldsymbol{g}))$. Then

$$\boldsymbol{S}(\boldsymbol{f}, \boldsymbol{g}) = \frac{\boldsymbol{m}}{\mathrm{LT}(\boldsymbol{f})}\boldsymbol{f} - \frac{\boldsymbol{m}}{\mathrm{LT}(\boldsymbol{g})}\boldsymbol{g}.$$

Note that if $\boldsymbol{f}$ and $\boldsymbol{g}$ are elements of a module M, then $\boldsymbol{S}(\boldsymbol{f}, \boldsymbol{g}) \in M$, since it is a linear combination of $\boldsymbol{f}$ and $\boldsymbol{g}$. To illustrate the definition we will show an example.

Example 1.2.22
We want to calculate the $\boldsymbol{S}$-element of $\boldsymbol{f}$ and $\boldsymbol{g}$, where

$$\begin{aligned} \boldsymbol{f} &= [xy + 1, x^2, 0]^T, \\ \boldsymbol{g} &= [x^3y^3, x - 1, y^2]^T, \end{aligned}$$

with respect to the $\succeq_{\mathrm{POT}}$ order. First note that $\boldsymbol{m} = \mathrm{LCM}(xy\boldsymbol{e}_1, x^3y^3\boldsymbol{e}_1) = x^3y^3$. Thus, we have

$$\begin{aligned} \boldsymbol{S}(\boldsymbol{f}, \boldsymbol{g}) &= \frac{x^3y^3}{xy}[xy + 1, x^2, 0]^T - \frac{x^3y^3}{x^3y^3}[x^3y^3, x - 1, y^2]^T \\ &= [x^2y^2, x^4y^2 - x + 1, -y^2]^T. \end{aligned}$$

We can easily use Singular to calculate the $\boldsymbol{S}$-element.

```
>ring R=0,(x,y),(lp,c);
>vector f=[xy+1,x2,0];
>vector g=[x3y3,x-1,y2];
>LIB "teachstd.lib"; //this loads the function spoly among others
```

```
>spoly(f,g);
[x2y2,x4y2-x+1,-y2]
```

□

Before giving Buchberg'er Criterion we need the following lemma.

Lemma 1.2.23
Let $\boldsymbol{f} = \sum_{i=1}^{s} c_i \boldsymbol{f}_i$, *where* $c_i \in k$ *and* $\boldsymbol{f}_i \in R^m$, *and suppose that* $\text{multideg}(\boldsymbol{f}_i) = \delta \in \mathbb{N}_0^n$ *for all* i. *If* $\text{multideg}(\boldsymbol{f}) < \delta$, *then* $\boldsymbol{f}$ *is a* k*-linear combination of the* $\boldsymbol{S}$*-elements* $\boldsymbol{S}(\boldsymbol{f}_j, \boldsymbol{f}_k)$ *for* $1 \leq j, k \leq s$. *Furthermore,* $\text{multideg}(\boldsymbol{S}(\boldsymbol{f}_j, \boldsymbol{f}_k)) < \delta$ *for each* j, k.

Proof
Let $d_i = \text{LC}(\boldsymbol{f}_i)$ such that $\text{LC}(c_i \boldsymbol{f}_i) = c_i d_i$. Since the multidegree of each $c_i \boldsymbol{f}_i$ is δ, and their sum have a multidegree strictly smaller than δ, the sum of the leading coefficients must cancel out,

$$\sum_{i=1}^{s} c_i d_i = 0. \tag{1.4}$$

Now, define $\boldsymbol{p}_i := \boldsymbol{f}_i / d_i$, and note that $\text{LC}(\boldsymbol{p}_i) = 1$. Consider the telescoping sum

$$\begin{aligned} \boldsymbol{f} &= \sum_{i=1}^{s} c_i \boldsymbol{f}_i = \sum_{i=1}^{s} c_i d_i \boldsymbol{p}_i \\ &= c_1 d_1 (\boldsymbol{p}_1 - \boldsymbol{p}_2) + (c_1 d_1 + c_2 d_2)(\boldsymbol{p}_2 - \boldsymbol{p}_3) + \dots \\ &\quad + (c_1 d_1 + \cdots + c_{s-1} d_{s-1})(\boldsymbol{p}_{s-1} - \boldsymbol{p}_s) + (c_1 d_1 + \cdots + c_s d_s)\boldsymbol{p}_s. \end{aligned} \tag{1.5}$$

The assumption $\text{LT}(\boldsymbol{f}_i) = d_i x^\delta \boldsymbol{e}_i$ implies that $\text{LCM}(\text{LT}(\boldsymbol{f}_j), \text{LT}(\boldsymbol{f}_k)) = x^\delta \boldsymbol{e}_i$ whenever $\text{LT}(\boldsymbol{f}_j)$ and $\text{LT}(\boldsymbol{f}_k)$ have the same standard basis $\boldsymbol{e}_i$, and

$$\begin{aligned} \boldsymbol{S}(\boldsymbol{f}_j, \boldsymbol{f}_k) &= \frac{x^\delta \boldsymbol{e}_i}{\text{LT}(\boldsymbol{f}_j)} \boldsymbol{f}_j - \frac{x^\delta \boldsymbol{e}_i}{\text{LT}(\boldsymbol{f}_k)} \boldsymbol{f}_k \\ &= \frac{x^\delta \boldsymbol{e}_i}{d_j x^\delta \boldsymbol{e}_i} d_j \boldsymbol{p}_j - \frac{x^\delta \boldsymbol{e}_i}{d_k x^\delta \boldsymbol{e}_i} d_k \boldsymbol{p}_k \\ &= \boldsymbol{p}_j - \boldsymbol{p}_k. \end{aligned}$$

From Equation (1.4) we have $(c_1 d_1 + \cdots + c_s d_s)\boldsymbol{p}_s = \boldsymbol{0}$, so Equation (1.5) yields

$$\boldsymbol{f} = c_1 d_1 \boldsymbol{S}(\boldsymbol{f}_1, \boldsymbol{f}_2) + (c_1 d_1 + c_2 d_2)\boldsymbol{S}(\boldsymbol{f}_2, \boldsymbol{f}_3) + \dots$$

$$+ (c_1 d_1 + \cdots + c_{s-1} d_{s-1}) \boldsymbol{S}(\boldsymbol{f}_{s-1}, \boldsymbol{f}_s),$$

which has the desired form. Since every $\boldsymbol{p}_i$ has multidegree δ, the sum $\boldsymbol{p}_j - \boldsymbol{p}_k$ must have a multidegree strictly smaller than δ for every j, k, and it follows that $\boldsymbol{S}(\boldsymbol{f}_j, \boldsymbol{f}_k)$ must also have a multidegree strictly smaller than δ for every j, k. ■

We can now state Buchberger's Criterion for submodules, which states when a given basis is a Gröbner basis. We will denote $\overline{\boldsymbol{S}(\boldsymbol{f}, \boldsymbol{g})}^G$ the remainder of $\boldsymbol{S}(\boldsymbol{f}, \boldsymbol{g})$ on division by G.

Theorem 1.2.24 (Buchberger's Criterion for Submodules)
A set $G = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s\} \subseteq R^m$ is a Gröbner basis for the submodule $M \subseteq R^m$ it generates if and only if the remainder on division by G of $\boldsymbol{S}(\boldsymbol{g}_i, \boldsymbol{g}_j)$ is $\mathbf{0}$ for all i, j.

Proof
If G is a Gröbner basis for the submodule $M \subseteq R^m$, then it follows from Proposition 1.2.18 that the remainder of $\boldsymbol{S}(\boldsymbol{g}_i, \boldsymbol{g}_j)$ on division by G is zero since $\boldsymbol{S}(\boldsymbol{g}_i, \boldsymbol{g}_j) \in M$ by construction.

To show the other way, suppose $\overline{\boldsymbol{S}(\boldsymbol{g}_i, \boldsymbol{g}_j)}^G = \mathbf{0}$ for every i, j, and let $\boldsymbol{f}$ be any nonzero element in M. If we can prove that $\mathrm{LT}(\boldsymbol{f}) \in \langle \mathrm{LT}(\boldsymbol{g}_1), \ldots, \mathrm{LT}(\boldsymbol{g}_s) \rangle$, then it follows that G is a Gröbner basis for M.

Since $\boldsymbol{f} \in M$, we can write $\boldsymbol{f}$ as

$$\boldsymbol{f} = \sum_{i=1}^{s} a_i \boldsymbol{g}_i, \quad a_i \in R. \tag{1.6}$$

First note that we must have

$$\text{multidegree}(\boldsymbol{f}) \leq \max\{\text{multidegree}(a_i \boldsymbol{g}_i)\}, \tag{1.7}$$

since, otherwise, the multidegree of the left-hand side of Equation (1.6) would be strictly lower then the right-hand side. Now, denote by $m(i)$ the multidegree of $a_i \boldsymbol{g}_i$, and let $\delta = \max\{m_1, \ldots, m_s\}$. Thus, we have multidegree$(\boldsymbol{f}) \leq \delta$. Since monomial orders are well-orderings, it is possible to find an expression for $\boldsymbol{f}$ that minimizes δ. We want to prove that multidegree$(\boldsymbol{f}) = \delta$ when δ is minimal, since we then can conclude that $\mathrm{LT}(\boldsymbol{f}) \in \langle \mathrm{LT}(\boldsymbol{g}_1), \ldots, \mathrm{LT}(\boldsymbol{g}_s) \rangle$, which is what we want. Suppose that multidegree$(\boldsymbol{f}) < \delta$, and rewrite $\boldsymbol{f}$ as

$$\boldsymbol{f} = \sum_{m(i)=\delta} a_i \boldsymbol{g}_i + \sum_{m(i)<\delta} a_i \boldsymbol{g}_i$$

$$= \sum_{m(i)=\delta} \mathrm{LT}(a_i)\boldsymbol{g}_i + \sum_{m(i)=\delta} (a_i - \mathrm{LT}(a_i))\boldsymbol{g}_i + \sum_{m(i)<\delta} a_i\boldsymbol{g}_i. \tag{1.8}$$

Note that multidegree$((a_i - \mathrm{LT}(a_i))\boldsymbol{g}_i) < \delta$, and, thus, this sum must also have a multidegree strictly smaller than δ. By our assumption the sum $\sum_{m(i)=\delta} \mathrm{LT}(a_i)\boldsymbol{g}_i$ must also have a multidegree strictly smaller than δ. Write $\mathrm{LT}(a_i) = c_i x^{\alpha(i)}$, $c_i \in k$, such that

$$\sum_{m(i)=\delta} \mathrm{LT}(a_i)\boldsymbol{g}_i = \sum_{m(i)=\delta} c_i x^{\alpha(i)}\boldsymbol{g}_i. \tag{1.9}$$

This sum can be written as a linear combination of $\boldsymbol{S}$-elements according to Lemma 1.2.23. First note that if $\mathrm{LT}(\boldsymbol{g}_j)$ and $\mathrm{LT}(\boldsymbol{g}_k)$ have different standard basis, then $\boldsymbol{S}(x^{\alpha(j)}\boldsymbol{g}_j, x^{\alpha(k)}\boldsymbol{g}_k) = \boldsymbol{0}$. Therefore, suppose their standard basis are the same, say $\boldsymbol{e}_i$, then

$$\begin{aligned}
\boldsymbol{S}(x^{\alpha(j)}\boldsymbol{g}_j, x^{\alpha(k)}\boldsymbol{g}_k) &= \frac{x^\delta \boldsymbol{e}_i}{x^{\alpha(j)}\mathrm{LT}(\boldsymbol{g}_j)} x^{\alpha(j)}\boldsymbol{g}_j - \frac{x^\delta \boldsymbol{e}_i}{x^{\alpha(k)}\mathrm{LT}(\boldsymbol{g}_k)} x^{\alpha(k)}\boldsymbol{g}_k \\
&= \frac{x^\delta \boldsymbol{e}_i}{x^{\gamma_{jk}}\boldsymbol{e}_i} \left(\frac{x^{\gamma_{jk}}\boldsymbol{e}_i}{\mathrm{LT}(\boldsymbol{g}_j)}\boldsymbol{g}_j - \frac{x^{\gamma_{jk}}\boldsymbol{e}_i}{\mathrm{LT}(\boldsymbol{g}_k)}\boldsymbol{g}_k \right) \\
&= x^{\delta-\gamma_{jk}}\boldsymbol{S}(\boldsymbol{g}_j, \boldsymbol{g}_k),
\end{aligned}$$

where $x^{\gamma_{jk}}\boldsymbol{e}_i = \mathrm{LCM}(\mathrm{LT}(\boldsymbol{g}_j), \mathrm{LT}(\boldsymbol{g}_k))$. Equation (1.9) can, thus, be written as

$$\sum_{m(i)=\delta} \mathrm{LT}(a_i)\boldsymbol{g}_i = \sum_{j,k} c_{jk} x^{\delta-\gamma_{jk}} \boldsymbol{S}(\boldsymbol{g}_j, \boldsymbol{g}_k), \quad c_{ijk} \in k. \tag{1.10}$$

The initial assumption was that the remainder of $\boldsymbol{S}(\boldsymbol{g}_j, \boldsymbol{g}_k)$ on division by G is zero, so the Division Algorithm yields

$$\boldsymbol{S}(\boldsymbol{g}_j, \boldsymbol{g}_k) = \sum_{i=1}^{s} a_{ijk}\boldsymbol{g}_i, \quad a_{ijk} \in R.$$

We know that

$$\mathrm{multideg}(a_{ijk}\boldsymbol{g}_i) \leq \mathrm{multideg}(\boldsymbol{S}(\boldsymbol{g}_j, \boldsymbol{g}_k))$$

for every i, j, k. Now, consider the equation

$$x^{\delta-\gamma_{jk}}\boldsymbol{S}(\boldsymbol{g}_j, \boldsymbol{g}_k) = \sum_{i=1}^{s} b_{ijk}\boldsymbol{g}_i,$$

where $b_{ijk} = x^{\delta-\gamma_{jk}}a_{ijk} \in R$. Then

$$\text{multideg}(b_{ijk}\boldsymbol{g}_i) \leq \text{multideg}(x^{\delta-\gamma_{jk}}\boldsymbol{S}(\boldsymbol{g}_j, \boldsymbol{g}_k)) < \delta. \tag{1.11}$$

We can now rewrite Equation (1.10) as

$$\sum_{m(i)=\delta} \text{LT}(a_i)\boldsymbol{g}_i = \sum_{j,k} c_{jk}\left(\sum_{i=1}^{s} b_{ijk}\boldsymbol{g}_i\right) = \sum_{i=1}^{s} a_i'\boldsymbol{g}_i,$$

where $\text{multideg}(a_i'\boldsymbol{g}_i) < \delta$ by Equation (1.11). If we again consider Equation (1.8), where we substitute $\sum_{m(i)=\delta}\text{LT}(a_i)\boldsymbol{g}_i$ with $\sum_{i=1}^{s}\text{LT}(a_i')\boldsymbol{g}_i$, then we note that $\boldsymbol{f}$ is still written as a linear combination of the $\boldsymbol{g}_i$'s, but where every summand have a multidegree strictly smaller than δ, which is a contradiction of δ being the smallest multidegree. Thus, equality must hold in Equation (1.7). It follows that $\text{LT}(\boldsymbol{f}) \in \langle \text{LT}(\boldsymbol{g}_1), \ldots, \text{LT}(\boldsymbol{g}_s)\rangle$, since when $\text{multideg}(\boldsymbol{f}) = \max\{\text{multideg}(a_i\boldsymbol{g}_i)\}$, then no cancelation of leading terms occur in Equation (1.8), and, thus, G is a Gröbner basis for M. ■

Consider the following example.

Example 1.2.25

Let $M = \langle \boldsymbol{f}_1, \boldsymbol{f}_2\rangle \subseteq (k[x,y])^3$, where $\boldsymbol{f}_1, \boldsymbol{f}_2$ are given as in Example 1.2.14,

$$\boldsymbol{f}_1 = (xy + 4x, 0, y^2)^T,$$
$$\boldsymbol{f}_2 = (0, y-1, x-2)^T.$$

We already saw in Example 1.2.16 that $G = (\boldsymbol{f}_1, \boldsymbol{f}_2)$ is a Gröbner basis for M. Since $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ have leading term in different standard basis, it follows that $\boldsymbol{S}(\boldsymbol{f}_1, \boldsymbol{f}_2) = \boldsymbol{0}$. If we add

$$\boldsymbol{f}_3 = \boldsymbol{f}_1 + \boldsymbol{f}_2 = (xy + 4x, y-1, x + y^2 - 2)^T$$

to G as a redundant generator, we have

```
>ring R=0,(x,y),(lp,c);
>LIB "teachstd.lib";
>vector f1=[xy+4x,0,y2];
>vector f2=[0,y-1,x-2];
>vector f3=f1+f2;
>vector s=spoly(f1,f3);
[0,-y+1,-x+2]
>module M=f1,f2,f3;
>division(s,M)[2]; //returns only the remainder
_[1]=0
```

Thus, we have

$$\boldsymbol{S}(\boldsymbol{f}_1, \boldsymbol{f}_3) = (0, -y+1, -x+2)^T,$$

but

$$\overline{\boldsymbol{S}(\boldsymbol{f}_1, \boldsymbol{f}_3)}^{G} = \mathbf{0}.$$

We clearly also have $\boldsymbol{S}(\boldsymbol{f}_2, \boldsymbol{f}_3) = \mathbf{0}$. It follows from Theorem 1.2.24 that $(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3)$ is a Gröbner basis for M. □

We can now extend Buchberger's Algorithm for ideals to submodules. With this algorithm we will be able to add extra generators to a given basis for a submodule to produce a Gröbner basis for this submodule.

Theorem 1.2.26 (Buchberger's Algorithm for Submodules)
Let $F = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t) \subseteq R^m$, and fix a monomial order on R^m. The following algorithm computes a Gröbner basis G for $M = \langle F \rangle$.

Algorithm 1.2.27
Input
$F = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$ where $\boldsymbol{f}_i \in R^m$
a monomial order $\succeq$

Output
a Gröbner basis $G = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s)$, $s \geq t$, for $M = \langle F \rangle$ with respect to $\succeq$

Initialize
$G := F$

Loop
WHILE $G \neq G'$ DO
    $G' := G$
    FOR each pair $\{i, j\}$, $i < j$ DO
        $\boldsymbol{S}_{ij} := \boldsymbol{S}(\boldsymbol{f}_i, \boldsymbol{f}_j)$
        IF $\boldsymbol{r}_{ij} = \overline{\boldsymbol{S}_{ij}}^{G'} \neq 0$ THEN
            t:=t+1
            $\boldsymbol{f}_t := r_{ij}$
            $G := G + \{\boldsymbol{f}_t\}$

Proof
First note that each step of Buchberger's Algorithm certainly gives a basis

for M, since we start with a basis to which we just add more elements, and since $\boldsymbol{S}_{ij} \in M$, then $\overline{\boldsymbol{S}_{ij}}^{G'} \in M$. The algorithm terminates when $\overline{\boldsymbol{S}_{ij}}^{G'} = \mathbf{0}$ for every i, j, and it follows from Buchberger's Criterion that G is a Gröbner basis for M. Hence, we just need to prove that the algorithm eventually terminates. After each loop of the while loop G will consist of the old basis $G =: G'$ and the nonzero $\boldsymbol{r}_{ij}$, and, thus,

$$\langle \mathrm{LT}(G') \rangle \subsetneq \langle \mathrm{LT}(G) \rangle$$

while $G' \neq G$, since $\boldsymbol{r}_{ij}$ is a remainder on division by G', $\mathrm{LT}(\boldsymbol{r}_{ij})$ is not divisible by the leading terms of any element of G'. Hence, $\mathrm{LT}(\boldsymbol{r}_{ij}) \notin \langle \mathrm{LT}(G') \rangle$, but $\mathrm{LT}(\boldsymbol{r}_{ij}) \in \langle \mathrm{LT}(G) \rangle$. Through the while loop we will, therefore, get a strictly ascending chain of submodules, and by the Ascending Chain Condition, Theorem 1.2.20, this chain will eventually stabilize. Thus, $\langle \mathrm{LT}(G') \rangle = \langle \mathrm{LT}(G) \rangle$ will eventually occur, which implies that $G' = G$ and the algorithm terminates. ■

Minimal and Reduced Gröbner bases are defined as for ideals.

Definition 1.2.28 (Minimal and Reduced Gröbner Bases)
Let $G \subseteq R^m$ be a Gröbner basis for a submodule $M \subseteq R^m$. A minimal Gröbner basis is a Gröbner basis G such that

(i). $\mathrm{LC}(\boldsymbol{g}) = 1$ for all $\boldsymbol{g} \in G$,

(ii). for all $\boldsymbol{g} \in G$, $\mathrm{LT}(\boldsymbol{g}) \notin \langle \mathrm{LT}(G - \{\boldsymbol{g}\}) \rangle$.

Furthermore, if

(i). for all $\boldsymbol{g} \in G$ no monomial of $\boldsymbol{g}$ lies in $\langle \mathrm{LT}(G - \{\boldsymbol{g}\}) \rangle$,

G is called a reduced Gröbner basis for M

In the last section of this chapter, we will consider the second problem stated in the beginning of this chapter. That is, we want to develop a method for finding a set of generators for a syzygy module $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s)$ given a generating set $(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_s)$ for some submodule of R^m.

1.3 Syzygy Modules

We will in this section show one of the properties of Gröbner basis for modules. R will again denote the polynomial ring $k[x_1, \ldots, x_n]$. Solving the Syzygy problem from Section 1.2 will allow us to find a presentation matrix for any submodule of R^m for which we know the generators.

Theorem 1.3.1 (Schreyer's Theorem)
Let $G = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$ be a Gröbner basis for some module over R^m with respect to any monomial order $\succeq$, and let $\varepsilon_1, \ldots, \varepsilon_s$ denote the standard basis vectors in R^s. Define

$$\begin{aligned} \boldsymbol{m}_{ij} &= \mathrm{LCM}(\mathrm{LT}(\boldsymbol{g}_i), \mathrm{LT}(\boldsymbol{g}_j)) \in R^m, \\ \boldsymbol{a}_{ij} &= a_{ij1}\varepsilon_1 + \cdots + a_{ijs}\varepsilon_s \in R^s, \end{aligned}$$

and

$$\boldsymbol{s}_{ij} = \frac{\boldsymbol{m}_{ij}}{\mathrm{LT}(\boldsymbol{g}_i)}\varepsilon_i - \frac{\boldsymbol{m}_{ij}}{\mathrm{LT}(\boldsymbol{g}_j)}\varepsilon_j - \boldsymbol{a}_{ij}.$$

Then the set $\{\boldsymbol{s}_{ij} | 1 \leq i, j \leq s\}$ forms a Gröbner basis for the syzygy module $M = \mathrm{Syz}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$ with respect to a monomial order $\succeq_G$ defined as follows: $x^\alpha \varepsilon_i \succeq_G x^\beta \varepsilon_j$ if $\mathrm{LT}(x^\alpha \boldsymbol{g}_i) \succeq \mathrm{LT}(x^\beta \boldsymbol{g}_j)$ in R^m, or if $\mathrm{LT}(x^\alpha \boldsymbol{g}_i) = \mathrm{LT}(x^\beta \boldsymbol{g}_j)$ and $i < j$.

Proof
That $\succeq_G$ is a monomial order follows directly from $\succeq$ being a monomial order.

Since $\boldsymbol{S}(\boldsymbol{g}_i, \boldsymbol{g}_j) = -\boldsymbol{S}(\boldsymbol{g}_j, \boldsymbol{g}_i)$ it suffices to consider $i < j$. We want to show that

$$\mathrm{LT}_{\succeq_G}(\boldsymbol{s}_{ij}) = \frac{\boldsymbol{m}_{ij}}{\mathrm{LT}(\boldsymbol{g}_i)}\varepsilon_i. \tag{1.12}$$

Since we only consider $i < j$, we have

$$\frac{\boldsymbol{m}_{ij}}{\mathrm{LT}(\boldsymbol{g}_i)}\varepsilon_i > \frac{\boldsymbol{m}_{ij}}{\mathrm{LT}(\boldsymbol{g}_j)}\varepsilon_j.$$

By construction we have

$$\mathrm{LT}(\boldsymbol{S}(\boldsymbol{g}_i, \boldsymbol{g}_j)) \geq \mathrm{LT}(a_{ijl}\boldsymbol{g}_l)$$

for all $1 \leq l \leq s$, and by definition

$$\mathrm{LT}\left(\frac{\boldsymbol{m}_{ij}}{\mathrm{LT}(\boldsymbol{g}_i)}\boldsymbol{g}_i\right) > \mathrm{LT}(\boldsymbol{S}(\boldsymbol{g}_i, \boldsymbol{g}_j)).$$

Thus,

$$\mathrm{LT}\left(\frac{\boldsymbol{m}_{ij}}{\mathrm{LT}(\boldsymbol{g}_i)}\varepsilon_i\right) > \mathrm{LT}(a_{ijl}).$$

which proves Equation (1.12). Now, let

$$\boldsymbol{f} = \sum_{i=1}^{s} f_i\varepsilon_i \in M,$$

and let $\mathrm{LT}_{\succeq_G}(f_i\varepsilon_i) = m_i\varepsilon_i$ for some term m_i appearing in f_i. Furthermore, let $\mathrm{LT}_{\succeq_G}(\boldsymbol{f}) = m_v\varepsilon_v$ for some v, and set

$$\boldsymbol{s} = \sum_{u\in S} m_u\varepsilon_u,$$

where $S = \{u | m_u\mathrm{LT}(\boldsymbol{g}_u) = m_v\mathrm{LT}(\boldsymbol{g}_v)\}$. Since $\boldsymbol{f} \in M = \mathrm{Syz}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$ we have $\boldsymbol{s} \in \mathrm{Syz}(\mathrm{LT}(\boldsymbol{g}_u)|u \in S)$. By Proposition 1.2.5 on page 23 part (iii) we know that $\boldsymbol{s}$ is an element of the syzygy module over R^s generated by the

$$\sigma_{uw} = \frac{\boldsymbol{m}_{uw}}{\mathrm{LT}(\boldsymbol{g}_u)}\varepsilon_u - \frac{\boldsymbol{m}_{uw}}{\mathrm{LT}(\boldsymbol{g}_w)}\varepsilon_w,$$

where $u < w$ are elements of S. Then it follows from Equation (1.12) that $\mathrm{LT}_{\succeq_G}(\boldsymbol{s})$ is divisible by $\mathrm{LT}_{\succeq_G}(\boldsymbol{s}_{ij})$ for some $i < j$, which means that the set $\{\boldsymbol{s}_{ij}|1 \leq i, j \leq s\}$ forms a Gröbner basis for M with respect to the $\succeq_G$ order.

■

The theorem shows how to find a Gröbner basis for the syzygy module $\mathrm{Syz}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$ over R^s with respect to the $\succeq_G$ order given a Gröbner basis $G = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$ with respect to any monomial order for some submodule over R^m. We will extend this result to a complete solution to the syzygy problem, and, thus, we will be able to find a generating set for a syzygy module given any set of generators for a submodule of R^m.

Let $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t \in R^m$ be a set of generators for a submodule $M = \langle \boldsymbol{f}_1, \ldots, \boldsymbol{f}_t \rangle$, and let $G = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$ be a Gröbner basis for M. Denote by F the $m \times t$ matrix with $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t$ as columns, and G the $m \times s$ matrix with $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s$ as columns. Then there exists a $t \times m$ matrix A such that $FA = G$, and a $s \times m$ matrix B such that $GB = F$.

Lemma 1.3.2

Let $G = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$ be some Gröbner basis for some submodule $M = \langle \boldsymbol{f}_1, \ldots, \boldsymbol{f}_t \rangle \subseteq R^m$, and let A and B be matrices such that $G = FA$ and $F = GB$, where $F = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$. If $\boldsymbol{s} \in R^s$ is an element of $\mathrm{Syz}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$, then $A\boldsymbol{s}$ is an element of $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$. Similarly, if $\boldsymbol{t} \in R^t$ is an element of $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$, then $B\boldsymbol{t}$ is an element of $\mathrm{Syz}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$. Furthermore, each column of the matrix $I_t - AB$ defines an element of $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$.

Proof

Consider the matrix equation $G = FA$ and multiple by $\boldsymbol{t}s \in \mathrm{Syz}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$ on the right. Then $\boldsymbol{0} = G\boldsymbol{s} = FA\boldsymbol{s} = F(A\boldsymbol{s})$, which shows that $A\boldsymbol{s}$ is an element of $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$. Now, consider the matrix equation $F = GB$ and multiply this by $\boldsymbol{t} \in \mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$ on the right. Then $\boldsymbol{0} = F\boldsymbol{t} = GB\boldsymbol{t} = G(B\boldsymbol{t})$, and, thus, $B\boldsymbol{t} \in \mathrm{Syz}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$. To see the last of the lemma, consider

$$0 = F - F = F - FAB = F(I_t - AB),$$

which proves the desired property. ■

We are now ready to show the proposition that solves the general problem of computing syzygy modules for a general ordered t-tuple $F = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$ of elements of R^m.

Proposition 1.3.3

Let $F = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$ be an ordered t-tuple of elements of R^m, and let $G = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$ be a Gröbner basis for $M = \langle F \rangle$ with respect to some monomial order on R^m. Let A and B be matrices such that $G = FA$ and $F = GB$, and let $\{\boldsymbol{s}_{ij} | 1 \leq i, j \leq s\}$ be a Gröbner basis for $\mathrm{Syz}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$, where $\boldsymbol{s}_{ij}$ is defined as in Theorem 1.3.1. Also, denote by $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_t$ the columns of the $t \times t$ matrix $I_t - AB$. Then

$$\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t) = \langle A\boldsymbol{s}_{ij}, \boldsymbol{S}_1, \ldots, \boldsymbol{S}_t \rangle. \tag{1.13}$$

Proof

Since $F(I_t - AB) = 0$, and since $FA\boldsymbol{s}_{ij} = G\boldsymbol{s}_{ij} = \boldsymbol{0}$, it is clear that $\langle A\boldsymbol{s}_{ij}, \boldsymbol{S}_1, \ldots, \boldsymbol{S}_t \rangle \subseteq \mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$. To show the other inclusion, we want to show that every element of $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$ can be written as a linear combination of the $A\boldsymbol{s}_{ij}$ and $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_t$. Let $\boldsymbol{t} \in \mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$ such that $B\boldsymbol{t} \in \mathrm{Syz}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$. Since the $\boldsymbol{s}_{ij}$ generate $\mathrm{Syz}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$, we can write $B\boldsymbol{t}$ as a linear combination of these generators,

$$B\boldsymbol{t} = \sum_{i,j} a_{ij} \boldsymbol{s}_{ij},$$

where $a_{ij} \in R$. Multiplying this equation by A on the left yields

$$AB\boldsymbol{t} = \sum_{i,j} a_{ij} A\boldsymbol{s}_{ij}.$$

Now, consider the equation

$$\begin{aligned} \boldsymbol{t} &= ((I_t - AB) + AB)\boldsymbol{t} \\ &= (I_t - AB)\boldsymbol{t} + \sum_{i,j} a_{ij} A\boldsymbol{s}_{ij}. \end{aligned}$$

Note that $(I_t - AB)\boldsymbol{t}$ is a linear combination of $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_t$. Thus, we have shown that $\boldsymbol{t} \in \langle A\boldsymbol{s}_{ij}, \boldsymbol{S}_1, \ldots, \boldsymbol{S}_t \rangle$, and since $\boldsymbol{t}$ is an arbitrary element of $\mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$, we have proved that $\langle A\boldsymbol{s}_{ij}, \boldsymbol{S}_1, \ldots, \boldsymbol{S}_t \rangle \supseteq \mathrm{Syz}(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t)$. Hence, the equality in Equation (1.13) holds. ■

In the proposition we forced G to be a Gröbner basis. This was only to ensure that the $\boldsymbol{s}_{ij}$ would generate the syzygy module $\mathrm{Syz}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$. We can generalize this proposition to any set of generators for a submodule M if we can find a presentation matrix D for the syzygy module.

Corollary 1.3.4

With the same notation as above, suppose that $G = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$ is any set of generators for $M = \langle F \rangle$, and let D be a presentation matrix for M such that the columns of D generate $\mathrm{Syz}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_s)$. Then the block matrix

$$[AD \; I_t - AB]$$

is a presentation matrix for M with respect to the generating set $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_t$.

Proof

This follows from directly from Proposition 1.3.3. ■

In the next chapter we will consider some of the basic theory about linear codes, and we will introduce the quasi-cyclic codes. We will show that there is a one-to-one correspondence between quasi-cyclic codes of length $n = ml$ and submodules of the quotient ring $(\mathbb{F}_q[x]/\langle x^m - 1 \rangle)^l$.

Chapter 2

Codes

In this chapter we will present some of the basic theory about linear codes. We will consider cyclic codes, so we in Chapter 3 can give a method for decoding a special class of cyclic codes, namely the Reed-Solomon codes. We will also introduce quasi-cyclic codes, which are a generalisation of cyclic codes, and we will show that we can see quasi-cyclic codes as submodules. We will therefore use the Gröbner basis theory from the last chapter to give a way to represent the quasi-cyclic codes. This chapter is based on [Justesen and Høholt, 2000, Sections 1.1-1.2 and 6.1-6.2], [Huffman and Pless, 2003, Sections 1.2-1.4, 4.1-4.2, and 4.5], [Lally, 2000, Section 1.1-1.2 and Chapter 2], and [Lally and Fitzpatrick, 2001, Section 1-2].

2.1 Linear Codes

In this project we will only consider linear codes. Let $\mathbb{F}_q^n$ denote the vector space of all n-tuples over a finite field $\mathbb{F}_q$. An (n, k) linear code $\mathcal{C}$ is a k-dimensional subspace of the vector space $\mathbb{F}_q^n$ with $M = q^k$ elements, $\mathcal{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_M\}$, where $\boldsymbol{c}_i = (c_{i0}, \ldots, c_{i(n-1)})$ are called the codewords of $\mathcal{C}$. The linear property assures that if $\boldsymbol{c}_i, \boldsymbol{c}_j \in \mathcal{C}$ and $f \in \mathbb{F}$, then $f\boldsymbol{c}_i + \boldsymbol{c}_j \in \mathcal{C}$. Further, the codeword with zeroes in every entry is always a codeword in $\mathcal{C}$. One of the most common ways to represent a code is by a generator matrix.

Definition 2.1.1 (Generator Matrix)
A generator matrix G for an (n, k) code $\mathcal{C}$ is a $k \times n$ matrix whose rows form a basis for $\mathcal{C}$ and are, thus, linearly independent.

Since any codeword in $\mathcal{C}$ can be represented as a linear combinations of the rows of the generator matrix G, we can represent each codeword as an

information vector $\boldsymbol{u}$ of k entries. Then

$$\boldsymbol{c} = \boldsymbol{u}G.$$

Example 2.1.2
A $(7, 4)$ binary code $\mathcal{C}$ can have the generator matrix

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

This code consists of $4^2 = 16$ codewords, which are linear combinations of the rows of G. Now, let $\boldsymbol{u} = (1, 0, 1, 0)$ be an information vector. Then

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

is a codeword in $\mathcal{C}$. □

Since every linear combination of the rows of a generator matrix lies in the code, row operations of the generator matrix does not change the code. It is often convenient to write a generator matrix in the form $G = [I_k \ A]$ called the standard form, where I_k is the $k \times k$ identity matrix. The generator matrix G in Example 2.1.2 is in standard form. Another way to represent a code $\mathcal{C}$ is with a parity check matrix.

Definition 2.1.3 (Parity Check Matrix)
A parity check matrix H for an (n, k) code $\mathcal{C}$ is an $(n - k) \times n$ matrix, whose rows are linearly independent, such that

$$\mathcal{C} = \{\boldsymbol{x} \in \mathbb{F}_q^n | H\boldsymbol{x}^T = \boldsymbol{0}\},$$

where $\boldsymbol{x}^T$ denotes the transpose of $\boldsymbol{x}$. Each $H\boldsymbol{x}^T$ is called a parity check.

The parity check matrix is itself a generator for an $(n, n-k)$ code called the dual code and denoted $\mathcal{C}^\perp$,

$$\mathcal{C}^\perp = \{\boldsymbol{x} \in \mathbb{F}_q^n | \boldsymbol{x} \cdot \boldsymbol{c} = 0 \ \forall \boldsymbol{c} \in \mathcal{C}\}.$$

When G is in standard form, a parity check matrix can be found as $H = [-A^T \ I_{n-k}]$.

Example 2.1.4

A parity check matrix for the $(7,4)$ code $\mathcal{C}$ represented by the generator matrix G in Example 2.1.2 is

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

□

To be able to consider the error-correcting capability of a code $\mathcal{C}$, we will consider the minimum distance and weight of a code.

Definition 2.1.5 (Hamming Distance $d(\boldsymbol{x}, \boldsymbol{y})$ and Hamming Weight $w(\boldsymbol{x})$)
The (Hamming) distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, denoted $d(\boldsymbol{x}, \boldsymbol{y})$, is defined as the number of coordinates in which $\boldsymbol{x}$ and $\boldsymbol{y}$ differ.
The (Hamming) weight of a vector is defined to be the number of nonzero coordinates, $w(\boldsymbol{x}) = d(\boldsymbol{x}, \boldsymbol{0})$.

The minimum distance of a code $\mathcal{C}$, denoted d, is the smallest distance between distinct codewords of the code. Since $d(\boldsymbol{x}, \boldsymbol{y}) = w(\boldsymbol{x}-\boldsymbol{y})$, the minimum distance of a code $\mathcal{C}$ is also equal to the smallest weight of any nonzero codeword. We often refer to a code $\mathcal{C}$ with minimum distance d as a $[n, k, d]$ code. The code in Example 2.1.2 is a $[7, 4, 3]$ code. When a codeword $\boldsymbol{c}$ is sent, the received word can contain some errors. If $\boldsymbol{y}$ is the received word, then $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$, where $\boldsymbol{e}$ is an error vector, and the number of errors is given by $w(\boldsymbol{e})$. We would like to be able to determine $\boldsymbol{e}$, but this cannot always be accomplished. Therefore, we call a code t-error correcting if for any two distinct codewords $\boldsymbol{c}_i \neq \boldsymbol{c}_j$, and any two error vectors $\boldsymbol{e}_i, \boldsymbol{e}_j$ of weight $\leq t$, we have $\boldsymbol{c}_i + \boldsymbol{e}_i \neq \boldsymbol{c}_j + \boldsymbol{e}_j$.

Theorem 2.1.6

An (n, k, d) code is t-error correcting if and only if $t < d/2$.

Proof

Suppose $t < d/2$, and let $\boldsymbol{c}_i, \boldsymbol{c}_j$ be two codewords and $\boldsymbol{e}_i, \boldsymbol{e}_j$ be two error vectors of weight $\leq t$ such that $\boldsymbol{c}_i + \boldsymbol{e}_i = \boldsymbol{c}_j + \boldsymbol{e}_j$. But since $\boldsymbol{c}_i - \boldsymbol{c}_j = \boldsymbol{e}_j - \boldsymbol{e}_i$, we have $w(\boldsymbol{e}_j - \boldsymbol{e}_i) = w(\boldsymbol{c}_i - \boldsymbol{c}_j) \leq 2t < d$, which contradicts the fact that d is the minimum distance. Conversely, suppose that $t \geq d/2$, and let $\boldsymbol{c}$ be a codeword of weight d. Change $t + 1$ of the nonzero entries of $\boldsymbol{c}$ to zero to obtain $\boldsymbol{y}$. Then $w(\boldsymbol{y}) = d - (t + 1) < t$, and $d(\boldsymbol{0}, \boldsymbol{y}) < t$, which means that

$\boldsymbol{y}$ is now closer to the $\mathbf{0}$ codeword than it is to $\boldsymbol{c}$. Thus, if $t < d/2$ we might not be able to decode a received word $\boldsymbol{y}$ to the correct codeword $\boldsymbol{c} = \boldsymbol{y} - \boldsymbol{e}$.

■

Since the code in Example 2.1.2 is a $[7, 4, 3]$ code, it is 1-error correcting, meaning if only one error occurred, then we will be able to correct it.

2.2 Cyclic Codes

In this section we will consider a specific class of linear codes, namely the cyclic codes.

Definition 2.2.1 (Cyclic Codes)
A linear code $\mathcal{C}$ of length n over a finite field $\mathbb{F}_q$ is called a cyclic code if for every codeword $c \in \mathcal{C}$ the codeword obtained by a cyclic shift is also a codeword in $\mathcal{C}$. That is,

$$c = (c_0, \ldots, c_{n-1}) \in \mathcal{C} \Rightarrow c' = (c_{n-1}, c_0, \ldots, c_{n-2}) \in \mathcal{C}.$$

Every codeword $c \in \mathcal{C}$ can be represented as a polynomial of degree at most $n-1$. In other words, we can represent the codeword $c = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ in polynomial form as $c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1} \in \mathbb{F}_q[x]$. With this notation a cyclic shift corresponds to multiplication by x modulo $x^n - 1$. It follows from the definition of cyclic codes that if $\mathcal{C}$ is a cyclic code and $c(x) \in \mathcal{C}$, then $xc(x) = c_{n-1} + c_0 x + c_1 x^2 + \cdots + c_{n-2}x^{n-1} \in \mathcal{C}$. With this in mind, there is a bijective correspondence between cyclic codes and ideals of the quotient ring

$$R_n = \mathbb{F}_q[x]/\langle x^n - 1\rangle.$$

It follows that it is natural to define a generator for the code $\mathcal{C}$.

Definition 2.2.2 (Generator for a Cyclic Code)
The generator $g(x)$ for a cyclic code $\mathcal{C}$ is the unique monic polynomial of minimum degree in $\mathcal{C}$.

We have the following theorem about the generator.

Theorem 2.2.3
Let $\mathcal{C}$ be a nonzero cyclic code in R_n. The generating polynomial from Definition 2.2.2 has the following properties:

(i) $g(x)$ *generates* $\mathcal{C}$*; that is,* $\mathcal{C} = \langle g(x) \rangle$,

(ii) $g(x)$ *divides* $x^n - 1$,

Let $k = n - \deg(g(x))$*, and let* $g(x) = \sum_{i=0}^{n-k} g_i x^i$*, where* $g_{n-k} = 1$*. Then*

(iii) the dimension of $\mathcal{C}$ *is* k *and* $\{g(x), xg(x), \ldots, x^{k-1}g(x)\}$ *is a basis for* $\mathcal{C}$,

(iv) every element of $\mathcal{C}$ *is uniquely expressible as a product* $g(x)f(x)$*, where* $f(x) = 0$ *or* $\deg(f(x)) < k$,

(v)

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k} \end{bmatrix}$$

$$\leftrightarrow \begin{bmatrix} g(x) & & & & \\ & xg(x) & & & \\ & & \ddots & & \\ & & & x^{k-1}g(x) \end{bmatrix}$$

is a generator matrix for $\mathcal{C}$,

Proof
Since $\mathcal{C}$ is a nonzero subset of R_n, there exists a polynomial $g(x)$ that is monic and is of minimum degree in $\mathcal{C}$. Let $c(x) \in \mathcal{C}$. Then the Division Algorithm in $\mathbb{F}_q[x]$ gives

$$c(x) = f(x)g(x) + r(x), \tag{2.1}$$

where either $r(x) = 0$ or $\deg(r(x)) < \deg(g(x))$. Since $g(x)$ has minimal degree in $\mathcal{C}$, it follows that $r(x) = 0$, and since $\mathcal{C}$ is an ideal in R_n, we can write $\mathcal{C} = \langle g(x) \rangle$, which proves (i). Part (ii) follows from the fact that

$x^n - 1$ corresponds to the zero codeword in $\mathcal{C}$, which means that we can write $x^n - 1 = f(x)g(x)$ for some $f(x)$.

Now, suppose $\deg(g(x)) = n-k$. By Equation (2.1) we have $c(x) = f(x)g(x)$ whenever $c(x) \in \mathcal{C}$. If $c(x) = 0$ we have $f(x) = 0$, and if $c(x) \neq 0$, $\deg(c(x)) = d < n$ we have $\deg(f(x)) = d - (n-k) < k$, which proves (iv). As a basis for $\mathcal{C}$ we can, thus, choose $\{g(x), xg(x), \ldots, x^{k-1}g(x)\}$, and, thus, we have the generator matrix

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k} \end{bmatrix}.$$

This proves the rest of the theorem. ■

Part (i) of the theorem proves that since a cyclic code is an ideal in R_n, and since a cyclic code can be generated by a single generator, then R_n is a principal ideal ring. Part (iv) of the theorem gives an easy way to determine if a given word $c(x)$ is a codeword in a given cyclic code $\mathcal{C}$, since by (iv) every $c(x) \in \mathcal{C}$ is divisible by $g(x)$.

In Theorem 2.2.3 we showed that the generator polynomial $g(x)$ divides $x^n - 1$, and, thus, there must be a correspondence between the divisors of the polynomial $x^n - 1$ and the generators of the cyclic codes in R_n. We can determine the number of codes in R_n if we know the factorization of $x^n - 1$.

Lemma 2.2.4

Let m denote the number of irreducible divisors of $x^n - 1$ of degree lower than n. The number of codes in R_n is then 2^m.

Proof

We will show this by induction. If $x^n - 1$ is irreducible, then the only codes are the zero code and the code generated by $x^n - 1$. Now, suppose the lemma is true for $m - 1$, and denote the 2^{m-1} generators by $g_1, \ldots, g_{2^{m-1}}$. For the mth irreducible divisor, say g, we then have the generators $g_1, \ldots, g_{2^{m-1}}, gg_1, \ldots, gg_{2^{m-1}}$. Thus, if $x^n - 1$ has m irreducible divisors, then we have $2 \cdot 2^{m-1} = 2^m$ codes. ■

Example 2.2.5

Let $n = 7$. We can factor $x^7 - 1$ in irreducible polynomials as

$$x^7 - 1 = (x+1)(x^3 + x^2 + 1)(x^3 + x + 1),$$

and, thus, $m = 3$. Lemma 2.2.4 says that $\mathcal{R}_7$ has $2^3 = 8$ binary cyclic codes $\mathcal{C}_i$ with generator $g_i(x)$, which we list below:

i	dim	$g_i(x)$
0	0	$1 + x^7$
1	1	$(1 + x^2 + x^3)(1 + x + x^3) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6$
2	3	$(1 + x)(1 + x^2 + x^3) = 1 + x + x^2 + x^4$
3	3	$(1 + x)(1 + x + x^3) = 1 + x^2 + x^3 + x^4$
4	4	$1 + x^2 + x^3$
5	4	$1 + x + x^3$
6	6	$1 + x$
7	7	1

We see that $g(x) = 1 + x^2 + x^3$ generates a binary cyclic code $\mathcal{C}$ in R_7 with dimension 4; that is, $\mathcal{C}$ is a $[7, 4]$ cyclic code over $\mathbb{F}_2$. The generator matrix for $\mathcal{C}$ is

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

This code is the same as the one used in Example 2.1.2 on page 48.

We want to check if the words $c_1(x) = 1 + x^2 + x^5 + x^6$ and $c_2(x) = 1 + x + x^3$ lie in $\mathcal{C} = \langle g(x) \rangle$, where $g(x) = 1 + x^2 + x^3$. We will use Singular to divide $c_1(x)$ and $c_2(x)$ by $g(x)$, respectively.

```
>LIB "redcgs.lib"
>ring R=2,x,lp;
>poly g=1+x2+x3;
>poly c1=1+x2+x5+x6;
>poly c2=1+x+x3;
>pdivi(c1,g)[1];   // the remainder of c1 on division by g
0
>pdivi(c2,g)[2];
x2+x
```

We conclude that $c_1(x)$ is a codeword in $\mathcal{C}$, but that $c_2(x)$ is not. □

For coding it is important to be able to determine the minimum distance of a code, or at least a lower bound, in order to be able to determine the

error-correcting capability of the code. Recall from Section 2.1 that a code of minimum distance d is t-error correcting if and only if $t < 2d$. We will here consider the BCH bound, since the Reed-Solomon codes, which we will study later, utilizes the BCH bound.

Theorem 2.2.6 (BCH Bound)
Let $g(x)$ be a generator polynomial for a cyclic $[n, k, d]$ code over $\mathbb{F}_q$, and suppose that $g(x)$ has among its zeroes $\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+\delta-2}$ for some $b \geq 0$, where $\alpha \in \mathbb{F}_q^m$ is a primitive nth root of unity, and m is the smallest integer such that $n|q^m - 1$. Then $d \geq \delta = \sharp ConsecutiveRoots(g(x)) + 1$.

Definition 2.2.7 (Reed-Solomon Codes)
A code $\mathcal{C}$ with generator polynomial of the form

$$g(x) = (x - \alpha)(x - \alpha^2) \cdots (x - \alpha^n),$$

where $n = q - 1$ and α is a primitive nth root of unity, is called a Reed-Solomon code.

In the next section we will consider quasi-cyclic codes, which are a generalisation of cyclic codes. We will see there there exists a natural correspondence between quasi-cyclic codes and submodules of the quotient ring R_m^l.

2.3 Quasi-Cyclic Codes

We will start this section with the classic definition of a quasi-cyclic code.

Definition 2.3.1 (Quasi-Cyclic Codes (Classic Definition))
A linear block code $\mathcal{C}$ of length $n = ml$ over a finite field $\mathbb{F}_q$ is called a quasi-cyclic code of index l if for every codeword $c \in \mathcal{C}$ there exists a number l such that the codeword obtained by l cyclic shifts is also a codeword in $\mathcal{C}$. That is,

$$c = (c_0, \ldots, c_{n-1}) \in \mathcal{C} \Rightarrow c' = (c_{n-l}, \ldots, c_0, \ldots, c_{n-l-1}) \in \mathcal{C}.$$

In the definition l is defined as the smallest number of cyclic shifts where the

code is invariant. Quasi-cyclic codes are a generalisation of cyclic codes; that is, cyclic codes are quasi-cyclic codes with $l = 1$.

Example 2.3.2

The binary $[6, 3]$ code with generator matrix

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

is a quasi-cyclic code with $l = 2$. To ease the visualization we can write the shifts as blocks,

$$G = \begin{bmatrix} 11 & 01 & 00 \\ 00 & 11 & 01 \\ 01 & 00 & 11 \end{bmatrix}.$$

□

In general, when we consider a generator matrix for a quasi-cyclic code, we do not restrict ourselves to the standard definition of a generator matrix, since we do not require the rows to be linearly independent, and, thus, the dimension of the code is not necessarily equal the number of rows. Since the rowspace of the generator matrix G is equal to the code, we can permutate the rows of the matrix, since this does not change the rowspace. Column permutations do change the rowspace, but we do, however, get an equivalent code, since the length, dimension and weight structure are unchanged.

Example 2.3.3

We will consider Example 2.3.2 again. If we group together columns 1,3,5 and 2,4,6 we get a code of the form

$$\begin{bmatrix} 100 & 110 \\ 010 & 011 \\ 001 & 101 \end{bmatrix}.$$

We notice that this generator matrix consists of two submatrices, and that both these matrices are a 3×3 circulant matrix. □

The generator matrix in Example 2.3.2 gives a so called 1-generator $[6, 3]$ code over $\mathbb{F}_2$ with $l = 2$, $m = 3$, and generator vector (11 01 00). The general form of a 1-generator matrix with generator vector

$$(a_{11}a_{12}\ldots a_{1l} \quad a_{21}a_{22}\ldots a_{2l} \quad \cdots \quad a_{m1}a_{m2}\ldots a_{ml})$$

for a quasi-cyclic code over $\mathbb{F}_q$ of length ml is

$$\begin{bmatrix} a_{11}a_{12}\ldots a_{1l} & a_{21}a_{22}\ldots a_{2l} & \cdots & a_{m1}a_{m2}\ldots a_{ml} \\ a_{m1}a_{m2}\ldots a_{ml} & a_{11}a_{12}\ldots a_{1l} & \cdots & a_{(m-1)1}a_{(m-1)2}\ldots a_{(m-1)l} \\ \vdots & \vdots & \ddots & \vdots \\ a_{21}a_{22}\ldots a_{2l} & a_{31}a_{32}\ldots a_{3l} & \cdots & a_{11}a_{12}\ldots a_{1l} \end{bmatrix},$$

where $a_{ij} \in \mathbb{F}_q$. As we did in Example 2.3.3 we can permutate the generator matrix to get a generator matrix consisting of l circulant submatrices,

$$\begin{bmatrix} C_1 & C_2 & \cdots & C_l \end{bmatrix}, \tag{2.2}$$

where each circulant submatrix is an $m \times m$ matrix of the form

$$C_i = \begin{bmatrix} c_0 & c_1 & \cdots & c_{m-1} \\ c_{m-1} & c_0 & \cdots & c_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & \cdots & c_0 \end{bmatrix} \tag{2.3}$$

with each row being a single cyclic shift of the previous one and $c_i \in \mathbb{F}_q$; that is, the matrix is completely specified by the vector $(c_0, c_1, \ldots, c_{m-1})$. A quasi-cyclic code can have more than one generator vector. A k-generator quasi-cyclic code with the same structure as in Equation (2.2) has the form

$$G = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1l} \\ C_{21} & C_{22} & \cdots & C_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ C_{k1} & C_{k2} & \cdots & C_{kl} \end{bmatrix}, \tag{2.4}$$

with each C_{ij} being of the same form as Equation 2.3. From now on when we refer to a generator matrix for a quasi-cyclic code it will be of the form of Equation (2.4).

Definition 2.3.4 (Quasi-Cyclic Codes)
A linear block code $\mathcal{C}$ with a generator matrix of the form of Equation (2.4) is a quasi-cyclic code.

We note that the circulant matrix has the same structure as the generator matrix for a cyclic code. It follows that if we write the vector $(c_0, c_1, \ldots, c_{m-1})$ in polynomial form as $c(x) = c_0 + c_1 x + \cdots + c_{m-1}x^{m-1}$, such that

$$C_i = \begin{bmatrix} c(x) & \mod (x^m - 1) \\ xc(x) & \mod (x^m - 1) \\ & \vdots \\ x^{m-1}c(x) & \mod (x^m - 1) \end{bmatrix},$$

then there is an isomorphism between $\mathbb{F}_q^m$ and the quotient ring $R = \mathbb{F}_q[x]/\langle x^m - 1\rangle$. If we extend this to the whole generator matrix from Equation (2.2) it can be proven that there exists an isomorphism between $\mathbb{F}_q^{lm}$ and R^l. Because of this isomorphism we can see quasi-cyclic codes $\mathcal{C}$ of index l and length $n = ml$ as an R-submodule of the module R^l. The kernel of the map

$$\phi : (\mathbb{F}_q[x])^l \to R^l$$

is the submodule $\tilde{K}$ in $\mathbb{F}_q[x]$,

$$\begin{aligned}\tilde{K} = ker(\phi) &= \{\boldsymbol{f} \in (\mathbb{F}_q[x])^l | \phi(\boldsymbol{f}) = \mathbf{0} \in R^l\} \\ &= \{\boldsymbol{f} = (f_1, \ldots, f_l) \in (\mathbb{F}_q[x])^l | f_i = k(x^m - 1), k \in \mathbb{F}_q[x], 1 \leq i \leq l\} \\ &= \langle (x^m - 1)\boldsymbol{e}_i, 1 \leq i \leq l\rangle,\end{aligned}$$

where $\boldsymbol{e}_i$, $1 \leq i \leq l$ is the standard basis vectors of $(\mathbb{F}_q[x])^l$. By the Homomorphism Theorem there exists an isomorphism between $(\mathbb{F}_q[x])^l/\tilde{K}$ and R^l, and, thus, there exists an one-to-one correspondence between the submodules $\mathcal{C}$ of R^l and the preimage submodules $\tilde{\mathcal{C}}$ of $(\mathbb{F}_q[x])^l$ containing $\tilde{K}$.

Suppose $\mathcal{C}$ is a k-generator quasi-cyclic code generated by the k elements $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_k$, where $\boldsymbol{r}_i = (r_{i1}, \ldots, r_{il})$. Then its preimage $\tilde{\mathcal{C}}$ is generated by $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_k$ and $(x^m - 1)\boldsymbol{e}_i$, $i = 1, \ldots, l$. Thus, the rows of the matrix

$$\begin{bmatrix} r_{11} & \ldots & r_{1l} \\ \vdots & \ddots & \vdots \\ r_{k1} & \ldots & r_{kl} \\ x^m - 1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & x^m - 1 \end{bmatrix}$$

generate $\tilde{\mathcal{C}}$. We know that every submodule has a minimal Gröbner basis, so denote by $\tilde{G}$ some minimal Gröbner basis with respect to the $\succeq_{\mathrm{POT}}$ order for the submodule $\tilde{\mathcal{C}}$, $\tilde{\mathcal{C}} = \langle \tilde{G}\rangle$. Since $\tilde{K} \subseteq \tilde{\mathcal{C}}$ there exists for every $1 \leq i \leq l$ a $\boldsymbol{g} \in \tilde{G}$ such that $\mathrm{LT}(\boldsymbol{g})$ divides $\mathrm{LT}((x^m - 1)\boldsymbol{e}_i)$, and it follows that the leading monomial of $\boldsymbol{g}$ must be in the ith position. Suppose $\boldsymbol{g}_1, \boldsymbol{g}_2 \in \tilde{G}$ are two element with the leading monomial in the same position, $\mathrm{LM}(\boldsymbol{g}_1) = X\boldsymbol{e}_i$ and $\mathrm{LM}(\boldsymbol{g}_2) = Y\boldsymbol{e}_i$. Since $\tilde{G}$ is a minimal Gröbner basis, and since $\mathrm{LM}(\boldsymbol{g}_1)$ or $\mathrm{LM}(\boldsymbol{g}_2)$ must be divisible by the other, it follows that $X = Y$ and $\tilde{G}$ must contain exactly l elements each with leading monomial in a different position.

By reordering the elements of $\tilde{G}$ we may assume that $\tilde{G}$ is of the triangular form

$$\tilde{G} = \begin{bmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \\ \vdots \\ \boldsymbol{g}_l \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1l} \\ 0 & g_{22} & \cdots & g_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_{ll} \end{bmatrix}, \tag{2.5}$$

where $g_{ii} \neq 0$. In the following theorem we will proof the structure of the reduced Gröbner basis for $\tilde{\mathcal{C}}$.

Theorem 2.3.5
Let $\tilde{\mathcal{C}}$ be a submodule of $(\mathbb{F}_q[x])^l$ containing $\tilde{K}$. Then $\tilde{\mathcal{C}}$ has a reduced Gröbner basis with respect to the $\succeq_{\mathrm{POT}}$ order of the form of Equation (2.5) where

(i). g_{ii} divides x^m-1, and if $\boldsymbol{f} \in \tilde{\mathcal{C}}$ has leading monomial in the ith position, then $\mathrm{LM}(\boldsymbol{f})$ is divisible by $g_{ii}\boldsymbol{e}_i$,

(ii). $\deg(g_{ji}) < \deg(g_{ii}) \leq m$ for $j < i$,

(iii). if $g_{ii} = x^m - 1$, then $\boldsymbol{g}_i = (x^m - 1)\boldsymbol{e}_i$,

(iv). the $\mathbb{F}_q$-dimension of $(\mathbb{F}_q[x])^l/\tilde{\mathcal{C}}$ is $\sum_{i=1}^{l} \deg(g_{ii})$.

Proof
Let $\tilde{G}$ be a reduced Gröbner basis with the triangular form of Equation (2.5). We have already discussed above that since $\tilde{K} \subseteq \tilde{\mathcal{C}}$, then g_{ii} must divide $x^m - 1$, so we just need to show that every element of $\tilde{\mathcal{C}}$ with leading polynomial in the ith position is also divisible by g_{ii}. This is straight forward, since when some element $\boldsymbol{f} \in \tilde{\mathcal{C}}$ has leading monomial in the ith position, then it must be generated by $\boldsymbol{g}_i$, and it follows that $\mathrm{LT}(\boldsymbol{f})$ divides $g_{ii}\boldsymbol{e}_i$.

For part (ii) note that since $\tilde{G}$ is a reduced Gröbner basis, no monomial of $\boldsymbol{g}_j \in \tilde{G}$ lies in $\langle \mathrm{LT}(G - \{\boldsymbol{g}_j\}) \rangle$, which means that no monomial of $\boldsymbol{g}_j$ is divisible by any $\mathrm{LT}(\boldsymbol{g}_i)$. Thus, $\mathrm{LM}(g_{ii})$ does not divide g_{ji} for each i, $1 \leq j < i \leq l$. This is true if and only if $\deg(g_{ii}) > \deg(g_{ji})$ for each i, j, $1 \leq j < i \leq l$. From part (i) we know that g_{ii} divides $x^m - 1$, so $\deg(g_{ii}) \leq m$.

To prove part (iii) suppose that $g_{ii} = x^m - 1$ such that $\boldsymbol{g}_i = (0, \ldots, 0, x^m - 1, g_{i,i+1}, \ldots, g_{il}) \in \tilde{G}$. Since $(x^m - 1)\boldsymbol{e}_i \in \tilde{\mathcal{C}}$ we also have $\boldsymbol{f} = \boldsymbol{g}_i - (x^m - 1)\boldsymbol{e}_i = (0, \ldots, 0, 0, g_{i,i+1}, \ldots, g_{il}) \in \tilde{\mathcal{C}}$. By Part (i) $\mathrm{LT}(\boldsymbol{f})$ is divisible by $g_{i+1,i+1}\boldsymbol{e}_{i+1}$. Since $\mathrm{LT}(\boldsymbol{f}) = \mathrm{LM}(g_{i,i+1})$, we have $\deg(g_{i,i+1}) \geq \deg(g_{i+1,i+1})$, but this is a

contradiction of part (ii), and it follows that $g_{i,i+1}, \ldots, g_{il} = 0$, and, thus, $\boldsymbol{g}_i = (x^m - 1)\boldsymbol{e}_i$.

The $\mathbb{F}_q$-dimension of $(\mathbb{F}_q[x])^l/\tilde{\mathcal{C}}$ is equal for the amount of monomials in the footprint of $(\mathbb{F}_q[x])^l/\tilde{\mathcal{C}}$. That is, the number of monomials $\boldsymbol{X}$ in $(\mathbb{F}_q[x])^l$ where no $\mathrm{LM}(\boldsymbol{g}_i)$ divides $\boldsymbol{X}$. Since every $\boldsymbol{g}_i$ has their leading monomial in a different position we know that one won't be a part of another. Then, since the leading monomial of $\boldsymbol{g}_i$ has degree $\deg(g_{ii})$ it follows that the whole dimension is $\sum_{i=1}^{l} \deg(g_{ii})$. ■

From part (iv) of this theorem we are able to determine the dimension of the code $\mathcal{C}$.

Proposition 2.3.6
The dimension k of the code $\mathcal{C}$ with a Gröbner basis of the form of Theorem 2.3.5 is given by

$$k = lm - \sum_{i=1}^{l} \deg(g_{ii}).$$

Proof
Consider the equation

$$\begin{aligned} k &= \dim(\mathcal{C}) \\ &= \dim(\tilde{\mathcal{C}}/\langle x^m - 1\rangle) \\ &= \dim(\tilde{\mathcal{C}}) - \dim(\langle x^m - 1\rangle) \\ &= \big(\dim((\mathbb{F}_q[x])^l) - \dim(\langle x^m - 1\rangle)\big) - \big(\dim((\mathbb{F}_q[x])^l) + \dim(\tilde{\mathcal{C}})\big) \\ &= \dim((\mathbb{F}_q[x])^l/\langle x^m - 1\rangle) - \dim((\mathbb{F}_q[x])^l/\tilde{\mathcal{C}}). \end{aligned}$$

In part (iv) of Theorem 2.3.5 we determined the codimension of $\tilde{\mathcal{C}}$, and from the same reasoning as in the proof of that theorem we have $\dim((\mathbb{F}_q[x])^l/\langle x^m - 1\rangle) = \sum_{i=1}^{l} m$. Thus,

$$k = \sum_{i=1}^{l} m - \sum_{i=1}^{l} \deg(g_{ii}) = ml - \sum_{i=1}^{l} \deg(g_{ii}).$$

■

Note that when we map a Gröbner basis $\tilde{G}$ for $\tilde{\mathcal{C}}$ to a Gröbner basis G for $\mathcal{C}$ we remove every $\boldsymbol{g}_i \in \tilde{G}$, where $\boldsymbol{g}_i = (x^m - 1)\boldsymbol{e}_i$, since $\boldsymbol{g}_i$ is then mapped to

the zero element in R^l. We can therefore also write the dimension k of $\mathcal{C}$ as

$$k = \sum_{\boldsymbol{g}_i \in G} (l - \deg(g_{ii})).$$

To see that this is correct, note that if $\boldsymbol{g}_i = (x^m - 1)\boldsymbol{e}_i$, then $m - \deg(g_{ii}) = m - m = 0$.

Before finishing this chapter we will define a few concepts that we will need when we decode quasi-cyclic codes.

Definition 2.3.7 (r-level Gröbner Bases)
Let the set $\tilde{G} = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_l\} \subseteq (\mathbb{F}_q[x])^l$ be a reduced Gröbner basis for a submodule $\tilde{\mathcal{C}} \subseteq (\mathbb{F}_q[x])^l$. We say that $\tilde{G}$ is an r-level Gröbner basis for $\tilde{\mathcal{C}}$ if there exists $\boldsymbol{g}_r \in \tilde{G}$, $1 \leq r \leq l$, such that $\boldsymbol{g}_r \notin \tilde{K}$ and $\boldsymbol{g}_j \in \tilde{K}$ for all j, $r < j \leq l$. The corresponding reduced Gröbner basis G for $\mathcal{C}$ is also called an r-level Gröbner basis, since G contains at most r generators.

Note that G only contains at most r generators since every generator $\boldsymbol{g}_j \in \tilde{K}$ equals zero in R^l. A 1-level Gröbner basis would only need one generator $\boldsymbol{g}$ to generate the whole code $\mathcal{C}$.

Corollary 2.3.8
A code $\mathcal{C}$ of index l and length ml has a 1-level reduced Gröbner basis if and only if it is generated by a single generator $\boldsymbol{g} \in R^l$ of the form

$$\boldsymbol{g} = (g, f_1 g, \ldots, f_{l-1} g), \quad f_i \in \mathbb{F}_q[x]$$

where g divides $x^m - 1$ and $\deg(f_i) < m - \deg(g)$ for $1 \leq i \leq l - 1$.

A further restriction on a one-generator code are the so-called Restriction-1 codes.

Definition 2.3.9 (Restriction-1 Codes)
A code $\mathcal{C}$ with a 1-level Gröbner basis $\boldsymbol{g} = (f_1 g, \ldots, f_l g) \subseteq R^l$, where $g = \gcd(g_1, \ldots, g_l, x^m - 1)$, $\gcd(f_i, (x^m - 1)/g) = 1$, and $\deg(f_i) < m - \deg(g)$ for $1 < i \leq l$ is called a Restriction-1 code.

The BCH bound for Restriction-1 codes is shown in the next theorem.

Theorem 2.3.10

Let $\mathcal{C}$ have a 1-level Gröbner basis with a generator of the form

$$\boldsymbol{g} = (f_1 g, \ldots, f_l g) \in R^l, \quad f_i \in \mathbb{F}_q[x],$$

where g divides $x^m - 1$, $\gcd\{f_i, (x^m - 1)/g\} = 1$, and $\deg(f_i) < m - \deg(m)$ for $1 < i \leq l$. If $\gcd(m, q) = 1$, then the minimum distance of $\mathcal{C}$ is at least

$$d_{\min} \geq l(\sharp ConsecutiveRoots(g) + 1).$$

Proof

From Theorem 2.2.6 on page 54 we know that the BCH bound for a cyclic code is

$$\sharp ConsecutiveRoots(g) + 1.$$

Since every partial codeword of $\mathcal{C}$ is a codeword in the cyclic code generated by g, it follows that the BCH for a Restriction-1 code is

$$d_{\min} \geq l(\sharp ConsecutiveRoots(g) + 1).$$

■

In the next chapter we will consider a decoding algorithm for Reed-Solomon codes. For this algorithm we will need the theory about modules and Gröbner bases for modules developed in Chapter 1.

Chapter 3

Reed-Solomon Decoding

In this chapter we will consider a specific type of cyclic codes, namely the Reed-Solomon codes.

We consider the polynomial ring $R = \mathbb{F}_q[x]/\langle x^n - 1\rangle$ and a primitive element α of the finite field $\mathbb{F}_q$. Now, consider the Reed-Solomon code $\mathcal{C} \subseteq R$ generated by the generator polynomial

$$g = (x - \alpha) \cdots (x - \alpha^{d-1}),$$

where d is the minimum distance of $\mathcal{C}$. If we assume that $d = 2t + 1$ for some t, we should be able to correct up to t errors in a received word. This chapter is mostly based on [Cox et al., 2005, Section 9.4] and [Moro et al., 2007, Section 4.2].

3.1 Reed-Solomon Decoding

We can write any codeword in $\mathcal{C}$ as a linear combination $\sum_{J=0}^{q-2} c_j x^j$. If $c \in \mathcal{C}$, then c must be divisible by g by Theorem 2.2.3 on page 51. Suppose $y = c + e$ is a received word, where $e = \sum_{i \in I} e_i x^i$ is the error in transmission. I is called the set of error locations, and the coefficients e_i are called the error values. An often used method for decoding is syndrome decoding, but the special algebraic structure of Reed-Solomon codes enables us to utilize much better methods for decoding.

Consider the function

$$E_j = y(\alpha^j) = c(\alpha^j) + e(\alpha^j) = e(\alpha^j),$$

since c is divisible by g. By computing the set of values E_j, $j = 1, \ldots, d-1$ we can determine whether errors have occurred. If $E_j = 0$ for all j, then y is divisible by g since they have the same roots, and it follows that y is a codeword. Furthermore, if we assume that the error vector has a weight less

than $t = \lfloor \frac{d-1}{2} \rfloor$, then $y = c$. Notice that the idea of E_j is very similar to the syndrome of the received word. If $E_j \neq 0$ for some j, then errors have occurred, and we will try to correct them.

Define the syndrome polynomial for y as the polynomial

$$S(x) = \sum_{j=1}^{d-1} E_j x^{j-1}$$

of degree at most $d-2$. If we let the sum run to infinity we will have the power series

$$E(x) = \sum_{j=1}^{\infty} E_j x^{j-1}.$$

Suppose that the error polynomial e is known. Then

$$E_j = \sum_{i \in I} e_i (\alpha^j)^i = \sum_{i \in I} e_i (\alpha^i)^j.$$

We rewrite $E(x)$ in the following way:

$$\begin{aligned} E(x) &= \sum_{j=1}^{\infty} \left(\sum_{i \in I} e_i (\alpha^i)^j \right) x^{j-1} \\ &= \sum_{j=0}^{\infty} \left(\sum_{i \in I} e_i (\alpha^i)^{j+1} \right) x^j \\ &= \sum_{i \in I} \left(\sum_{j=0}^{\infty} (\alpha^i)^j x^j \right) e_i \alpha^i \\ &= \sum_{i \in I} \frac{e_i \alpha^i}{1 - \alpha^i x}, \end{aligned}$$

where we in the last equation used the fact that $\sum_{j=0}^{\infty} e_i (\alpha^i)^j x^j$ is a geometric series. We will write $E(x)$ as

$$E(x) = \frac{\Omega(x)}{\Lambda(x)}, \tag{3.1}$$

where

$$\Omega(x) = \sum_{i \in I} e_i \alpha^i \prod_{j \neq i, j \in I} \left(1 - \alpha^j x\right),$$

$$\Lambda(x) = \prod_{i \in I} \left(1 - \alpha^i x\right)$$

with

$$\deg(\Omega(x)) \leq \deg(\Lambda(x)) - 1.$$

Notice that the roots of $\Lambda(x)$ are α^{-i} for $i \in I$. That is, we can use $\Lambda(x)$ to determine the error locations, and, thus, $\Lambda(x)$ is called the error locator polynomial. Since

$$\Omega(\alpha^{-i}) = e_i \alpha^i \prod_{j \neq i, j \in I} \left(1 - \alpha^j \alpha^{-i}\right) \neq 0, \tag{3.2}$$

$\Omega(x)$ and $\Lambda(x)$ have no roots in common, and it follows that they must be relatively prime, since all its factors of degree one are different.

Next, consider the difference between $E(x)$ and $S(x)$,

$$\begin{aligned} E(x) - S(x) &= \sum_{j=d}^{\infty} \left(\sum_{i \in I} e_i (\alpha^i)^j \right) x^{j-1} \\ &= x^{d-1} \frac{\Gamma(x)}{\Lambda(x)}, \end{aligned} \tag{3.3}$$

where

$$\begin{aligned} \Gamma(x) &= \sum_{i \in I} e_i \alpha^{id} \prod_{j \neq i, j \in I} \left(1 - \alpha^j x\right), \\ \deg(\Gamma(x)) &\leq \deg(\Lambda(x)) - 1. \end{aligned}$$

By combining Equation (3.1) and (3.3), where we write $d - 1 = 2t$,

$$\Omega(x) = \Lambda(x) S(x) + x^{2t} \Gamma(x), \tag{3.4}$$

which we can also write as the congruence equation

$$\Omega(x) \equiv \Lambda(x) S(x) \mod x^{2t}. \tag{3.5}$$

We will refer to this equation as the key equation for decoding.

We will now consider the received word $y = c + e$ where the error vector is unknown and of weight at most t by assumption. We calculate the syndrome polynomial $S(x)$ and consider equation (3.5) where $S(x)$ and x^{2t} are known, and $\Omega(x)$, $\Lambda(x)$ unknown.

Theorem 3.1.1
Let $S(x)$ be the syndrome polynomial corresponding to a received word y with an error of weight at most t. Up to a constant multiple, there exists a unique solution (Ω, Λ) of (3.5) that satisfies the degree conditions,

$$\deg(\Omega) < \deg(\Lambda) \leq t,$$

and in which Ω and Λ are relatively prime.

Proof
Let (Ω, Λ) and $(\overline{\Omega}, \overline{\Lambda})$ be two solutions satisfying the degree and relatively prime conditions. Both satisfy the key equation,

$$\begin{aligned} \Omega &\equiv \Lambda S \mod x^{2t}, \\ \overline{\Omega} &\equiv \overline{\Lambda} S \mod x^{2t}. \end{aligned}$$

Multiplying the first equation by $\overline{\Lambda}$, the second by Λ, and subtracting yields the congruence relation

$$\Omega\overline{\Lambda} \equiv \overline{\Omega}\Lambda \mod x^{2t}.$$

Since the degree conditions are satisfied for both (Ω, Λ) and $(\overline{\Omega}, \overline{\Lambda})$ both sides of the congruence relation must be of degree at most $2t-1$, whereby it follows that $\Omega\overline{\Lambda} = \overline{\Omega}\Lambda$. It follows from the relatively prime condition that Λ must be a multiple of $\overline{\Lambda}$, Ω must be a multiple of $\overline{\Omega}$ and vice versa. This means that (Ω, Λ) and $(\overline{\Omega}, \overline{\Lambda})$ can only differ by a constant multiple. ■

Given a solution (Ω, Λ) we can determine the roots of the error locator polynomial $\Lambda(x)$ to determine the error locations. We can then use Equation (3.2) to determine the values of e_i. Hence, we can decode a received word by solving the key equation. The problem is that (Ω, Λ) might not be unique. That is, the solution might not satisfy the degree condition. Therefore, consider the set of all possible solutions,

$$K = \{(\Omega, \Lambda) | \Omega \equiv \Lambda S \mod x^{2t}\} \subseteq (\mathbb{F}_q[x])^2.$$

We will prove that K is an $\mathbb{F}_q[x]$-submodule of $(\mathbb{F}_q[x])^2$. Let $(\Omega, \Lambda), (\overline{\Omega}, \overline{\Lambda}) \in K$ and $f, g \in \mathbb{F}_q[x]$. Then

- $f((\Omega, \Lambda) + (\overline{\Omega}, \overline{\Lambda})) = f(\Omega, \Lambda) + f(\overline{\Omega}, \overline{\Lambda})$,
- $(f + g)(\Omega, \Lambda) = f(\Omega, \Lambda) + g(\Omega, \Lambda)$,

- $(fg)(\Omega, \Lambda) = f(g(\Omega, \Lambda))$,
- $1(\Omega, \Lambda) = (\Omega, \Lambda)$.

We will find a generating set for K. Consider Equation (3.4) again,

$$\Omega(x) = \Lambda(x)S(x) + x^{2t}\Gamma(x).$$

If we set $\Lambda(x) = 0$, then we have $\Omega(x) = \Gamma(x)x^{2t}$. So in this situation (Ω, Λ) can be generated by $(x^{2t}, 0)$. Now, set $\Gamma(x) = 0$ such that $\Omega(x) = \Lambda(x)S$. In this situation we can use the generator $(S, 1)$. Thus, we have the generating set

$$\begin{aligned} g_1 &= (x^{2t}, 0), \\ g_2 &= (S, 1). \end{aligned} \tag{3.6}$$

We have the following proposition.

Proposition 3.1.2
Let k be any field, and let M be a submodule of $(k[x])^2$. Let $\succeq$ be any monomial order on $(k[x])^2$. Then the following conditions are equivalent:

(i). The k-vector space $(k[x])^2/M$ is finite-dimensional.

(ii). $\langle \mathrm{LT}_{\succeq}(M) \rangle$ contains elements of the form $x^u \boldsymbol{e}_1 = (x^u, 0)$ and $x^v \boldsymbol{e}_2 = (0, x^v)$ for some $u, v \geq 0$.

Proof
Let G be a Gröbner basis for M with respect to the monomial order $\succeq$. Every elements of $(k[x])^2/M$ can be written as a linear combination of the monomials in the complement of $\langle \mathrm{LT}_{\succeq}(M) \rangle$. The proposition follows directly from the fact that there is a finite number of monomials in the complement if and only if $\langle \mathrm{LT}_{\succeq}(M) \rangle$ contains multiples of both $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. ■

We will now define a new monomial order.

Definition 3.1.3
Let $r \in \mathbb{Z}$, and define an order $\succeq_r$ by the following rules,

- $x^m \boldsymbol{e}_i \succeq_r x^n \boldsymbol{e}_i$ if $m > n$ and $i = 1, 2$,
- $x^m \boldsymbol{e}_2 \succeq_r x^n \boldsymbol{e}_1$ if and only if $m + r \geq n$.

We will consider a few examples where we order the monomials in $(k[x])^2$.

Example 3.1.4

Let $r = -1$, then

$$\boldsymbol{e}_2 \preceq_{-1} \boldsymbol{e}_1 \preceq_{-1} x\boldsymbol{e}_2 \preceq_{-1} x\boldsymbol{e}_1 \preceq_{-1} x^2\boldsymbol{e}_2 \preceq_{-1} x^2\boldsymbol{e}_1 \preceq_{-1} \dots.$$

Let $r = 0$, then

$$\boldsymbol{e}_1 \preceq_0 \boldsymbol{e}_2 \preceq_0 x\boldsymbol{e}_1 \preceq_0 x\boldsymbol{e}_2 \preceq_0 x^2\boldsymbol{e}_1 \preceq_0 x^2\boldsymbol{e}_2 \preceq_0 \dots.$$

Let $r = 1$, then

$$\boldsymbol{e}_1 \preceq_1 x\boldsymbol{e}_1 \preceq_1 \boldsymbol{e}_2 \preceq_1 x^2\boldsymbol{e}_1 \preceq_1 x\boldsymbol{e}_2 \preceq_1 x^3\boldsymbol{e}_1 \preceq_1 \dots.$$

Let $r = 2$, then

$$\boldsymbol{e}_1 \preceq_2 x\boldsymbol{e}_1 \preceq_2 x^2\boldsymbol{e}_1 \preceq_2 \boldsymbol{e}_2 \preceq_2 x^3\boldsymbol{e}_1 \preceq_2 x\boldsymbol{e}_2 \preceq_2 \dots.$$

□

Notice that for $r = -1$ and $r = 0$ we have the standard TOP order with $\boldsymbol{e}_2 \preceq \boldsymbol{e}_1$ and $\boldsymbol{e}_1 \preceq \boldsymbol{e}_2$, respectively. Also notice that the general structure when $r \geq 0$ is $\boldsymbol{e}_1$ as the smallest element followed by multiples of $\boldsymbol{e}_1$ until we reach $x^r\boldsymbol{e}_1 \preceq_r \boldsymbol{e}_2$. Hereafter, the chain will continue with multiples of $x^r\boldsymbol{e}_1 \preceq_r \boldsymbol{e}_2$.

Proposition 3.1.5

Let M be a submodule of $(k[x])^2$, and fix $r \in \mathbb{Z}$. Assume that the equivalent conditions in Proposition 3.1.2 are satisfied. Then the subset $G \subseteq M$ is a reduced Gröbner basis of M with respect to $\succeq_r$ if and only if $G = \{g_1 = (g_{11}, g_{12}), g_2 = (g_{21}, g_{22})\}$, where g_i satisfy the following two properties:

(i). $\mathrm{LT}(g_1) = x^u\boldsymbol{e}_1$ *and* $\mathrm{LT}(g_2) = x^v\boldsymbol{e}_2$,

(ii). $\deg(g_{12}) < v$ *and* $\deg(g_{21}) < u$.

Proof

We have $\langle \mathrm{LT}(M) \rangle = \langle \mathrm{LT}(G) \rangle$ if and only if G is a Gröbner basis for M. This is satisfied if and only if condition (i) is satisfied.

For G to be reduced, no monomials of g_1 must lie in $\langle \mathrm{LT}(g_2) \rangle$ and vice versa, which in satisfied if and only if condition (ii) is satisfied. ■

In Equation (3.6) we found a generating set for K, $\{g_1 = (x^{2t}, 0), g_2 = (S, 1)\}$. Note that if we order by the $\succeq_{\deg(S)}$ order, then $\mathrm{LT}(g_1) = x^{2t}\boldsymbol{e}_1$ and $\mathrm{LT}(g_2) = \boldsymbol{e}_2$. The degree conditions in Proposition 3.1.5 are also satisfied, since we have $\deg(g_{12}) = \deg(0) = 0 < 1$ and $\deg(g_{21}) = \deg(S) \leq d - 2 = 2t - 1 < 2t$. Hence, $\{(x^{2t}, 0), (S, 1)\}$ is a reduced Gröbner basis for K with respect to $\succeq_{\deg(S)}$.

Definition 3.1.6 (Minimal Element)
Let M be a nonzero submodule of $(k[x])^2$. A minimal element of M with respect to the monomial order $\succeq_r$ is a $\boldsymbol{g} \in M \backslash \{\boldsymbol{0}\}$ such that $\mathrm{LT}(\boldsymbol{g})$ is minimal with respect to $\succeq$.

Proposition 3.1.7
Let $M \subseteq (k[x])^2$ be a submodule and $\succeq_r$ a monomial order for $(k[x])^2$. Every Gröbner basis for M with respect to $\succeq_r$ contains a minimal element of M with respect to $\succeq_r$. Furthermore, the minimal element is unique up to a nonzero constant multiple.

Proof
A basis for $\langle \mathrm{LT}(M) \rangle$ must contain the smallest element of $\langle \mathrm{LT}(M) \rangle$ to be able to generate everything. Since $\langle \mathrm{LT}(M) \rangle = \langle \mathrm{LT}(G) \rangle$ for a Gröbner basis G, then G must contain the minimal element. The uniqueness of the minimal element follows from the definition of monomial orders. ■

In our case we have

$$\mathrm{LT}((S, 1)) = \boldsymbol{e}_2 \preceq_{\deg(S)} x^{2t}\boldsymbol{e}_1 = \mathrm{LT}((x^{2t}, 0)),$$

and, thus, $(S, 1)$ is the minimal element with respect to $\succeq_{\deg(S)}$.

Proposition 3.1.8
Let $\boldsymbol{g} = (\Omega, \Lambda)$ be a solution to the key equation, Equation (3.5), satisfying the conditions in Theorem 3.1.1. Then $\boldsymbol{g}$ is a minimal element of K with respect to $\succeq_{-1}$.

Proof
For an element $\boldsymbol{g} = (\Omega, \Lambda) \in K$ we have $\deg(\Lambda) > \deg(\Omega)$ if and only if $\mathrm{LT}_{\succeq_{-1}}(\boldsymbol{g}) = x^u\boldsymbol{e}_2$ for some u.

Suppose that $\boldsymbol{g}$ is not a minimal element. That is, there exists an element $\boldsymbol{h} = (\overline{\Omega}, \overline{\Lambda}) \in K$ such that $\mathrm{LT}(\boldsymbol{g}) \succeq_{-1} \mathrm{LT}(\boldsymbol{h})$. By Theorem 3.1.1 we have

$\deg(\overline{\Omega}) \geq \deg(\overline{\Lambda})$ since only $\boldsymbol{g}$ satisfies the degree conditions, and thus $\mathrm{LT}(\boldsymbol{h})$ must be a multiple of $\boldsymbol{e}_1$. Thus,

$$\deg(\Lambda) > \deg(\overline{\Omega}) \geq \deg(\overline{\Lambda}). \tag{3.7}$$

Both $\boldsymbol{h}$ and $\boldsymbol{g}$ satisfy the key equation,

$$\begin{aligned} \overline{\Omega} &\equiv S\overline{\Lambda} \mod x^{2t}, \\ \Omega &\equiv S\Lambda \mod x^{2t}. \end{aligned}$$

Multiplying the first equation with Λ and the second with $\overline{\Lambda}$ and subtracting yields the congruence relation

$$\Lambda\overline{\Omega} \equiv \overline{\Lambda}\Omega \mod x^{2t}. \tag{3.8}$$

Since

$$\deg(\Omega) < \deg(\Lambda) \leq t$$

and, thus, also $\deg(\overline{\Omega}) < t$ by Equation (3.7), we have a contradiction, since we have $\deg(\Lambda) > \deg(\overline{\Lambda})$, and, thus, the left hand side of Equation (3.8) has a degree strictly less than that on the right hand side. Hence, $\boldsymbol{g}$ must be the minimal element of K. ■

Let us sum up what we have found out. We know that the set $\{(S, 1), (x^{2t}, 0)\}$ is a Gröbner basis of $K = \{(\Omega, \Lambda) | \Omega \equiv \Lambda S \mod x^{2t}\}$ with respect to the $\succeq_{\deg(S)}$ order. Proposition 3.1.8 says that a solution of the key equation is a minimal element of K with respect to the $\succeq_{-1}$ order. If we consider the monomial order $\succeq_{-1}$ and calculate a Gröbner basis with respect to this, then Proposition 3.1.7 guaranties that the minimal element always appear in the Gröbner basis. Thus, we can calculate a solution of the key equation by doing this.

Example 3.1.9

Consider the code $\mathcal{C}$ over $\mathbb{F}_9$ generated by

$$\begin{aligned} g(x) &= (x - a)(x - a^2)(x - a^3)(x - a^4) \\ &= (-a + 1) - x + (-a + 1)x^2 + (a + 1)x^3 + x^4 \end{aligned}$$

with $d = 5$ and $t = 2$. We send the codeword

$$c = (-a + 1)x - x^2 + (-a + 1)x^3 + (a + 1)x^4 + x^5.$$

Suppose errors occur in the transmission of c such that we receive the word

$$y = c + e = -1 + (-a+1)x - x^2 + (-a+1)x^3 + (a+1)x^4.$$

Note that $e = -1 - x^5$. We construct the syndrome polynomial.

```
>ring R=(3,a),x,(lp,c);
>minpoly=a2+a+2;
>poly y=-1+(-a+1)*x-x2+(-a+1)*x3+(a+1)*x4;
>poly s0=subst(y,x,a);
>poly s1=subst(y,x,a2);
>poly s2=subst(y,x,a3);
>poly s3=subst(y,x,a4);
>poly s=s0+s1*x+s2*x2+s3*x3;s;
(-a+1)*x2+(a+1)*x+(a-1)
```

We then define the Gröbner basis for K with respect to $\succeq_{\deg(s)}$.

```
>vector g1=[x4,0];
>vector g2=[s,1];
>module K=g1,g2;
```

We want to find the Gröbner basis with respect to $\succeq_{\mathrm{TOP}}$.

```
>option(redSB);
>module G=std(K);
>print(G);
-x+(a),          x2+(-a)*x-1,
x2+(a)*x+(-a-1),(a-1)
>G[1]<G[2]
1
```

So we have found the Gröbner basis

$$G = \begin{bmatrix} -x+a & x^2+ax-a-1 \\ x^2-ax-1 & a-1 \end{bmatrix},$$

and we have found the minimal element to be $(-x+a, x^2+ax-a-1)$. Thus, we have

$$(\Omega, \Lambda) = (-x+a, x^2+ax-a-1).$$

```
>poly omega=-x+a;
>poly lambda=x2+a*x-a-1;
```

By using Λ we can find the error locations.

```
>subst(lambda,x,a);
1
>subst(lambda,x,a2);
(a)
>subst(lambda,x,a3);
0
>subst(lambda,x,a4);
(a)
>subst(lambda,x,a5);
(-a-1)
>subst(lambda,x,a6);
-1
>subst(lambda,x,a7);
-1
>subst(lambda,x,a8);
0
>1/a3,1/a8;
(-a) 1
>a1,a2,a3,a4,a5,a6,a7,a8;
(a) (-a+1) -1 (-a) (a-1) (a+1) 1
```

We see that we have errors in the coefficients of x^0 and x^5. □

An alternative method for calculating a Gröbner basis for K with respect to $\succeq_{-1}$ is to use an extension of a Gröbner basis conversion algorithm developed by Faugère, Gianni, Lazard, and Mora called the FGLM algorithm, which can be found in [Cox et al., 2005, Section 2.3]. We will present a generalized FGLM algorithm that takes a Gröbner basis G_1 for some submodule $M \subseteq (\mathbb{F}_q[x])^l$ with respect to some monomial order $\succeq'$, and where M has a finite footprint, which is the set of monomials in the complement of $\langle \mathrm{LT}(M) \rangle$, and gives a reduced Gröbner basis G_2 with respect to some other monomial order $\succeq$. We will use the algorithm found in [Fitzpatrick, 1997], but we will change it to make it fit to our situation. For our use we have $G_1 = G$, $G_2 = G_{lex}$ and the monomial orders $\succeq'=\succeq_{\deg(S)}$ and $\succeq=\succeq_{-1}$. Before giving the algorithm we will need a few functions.

- order(S) puts the list S of terms into ascending order with respect to $\succeq$.
- next(S) removes the first term from the list S and returns its value.
- rem($\boldsymbol{g}$) gives the remainder of $\boldsymbol{g}$ on division by G_1 with respect to $\succeq'$.

Algorithm 3.1.10

Input
G_1 // Gröbner basis of $M \subseteq (\mathbb{F}_q[x])^l$ with respect to $\succeq'$, where M has a finite footprint
$\succeq$ // monomial order

Output
G_2 // reduced Gröbner basis of $M \subseteq (\mathbb{F}_q[x])^l$ with respect to $\succeq$
LT // LT(G_2) with respect to $\succeq$
FP // the footprint for M with respect to $\succeq$

Initialize
MON := order($\{\boldsymbol{e}_k, 1 \leq k \leq l\}$)
$G_2 := \{\}$
LT := $\{\}$

Loop
WHILE MON $\neq \{\}$ DO
    $\boldsymbol{t}$:= next(MON)
    IF rem($\boldsymbol{t}$) $= \sum_{\boldsymbol{v} \in \mathrm{FP}} f_{\boldsymbol{v}} \cdot \mathrm{rem}(\boldsymbol{v}), f_{\boldsymbol{v}} \in \mathbb{F}_q$ THEN
        $G_2 := G_2 \cup \{\boldsymbol{t} - \sum_{\boldsymbol{v} \in \mathrm{FP}} f_v \boldsymbol{v}\}$
        LT := LT $\cup \{\boldsymbol{t}\}$
    ELSE
        FP := FP $\cup \{\boldsymbol{t}\}$
        MON := MON $\cup \{x\boldsymbol{t}\}$
        order(MON)

Proof

We will first prove that the algorithm do terminate, and then prove that G_2 is the reduced Gröbner basis of $M \subseteq (\mathbb{F}_q[x])^l$ with respect to $\succeq$. Note that the loop always start by removed a monomial from MON and, thereafter, we either add a new monomial to MON and FP or we don't change the size of either. Now, since the footprint of M is finite neither of the two cases can happen infinitely many times, and, thus, the algorithm must terminate.

We clearly have $G_2 \subseteq M$, since for every element $\boldsymbol{t} - \sum_{\boldsymbol{v} \in \mathrm{FP}} f_{\boldsymbol{v}} \boldsymbol{v} \in G_2$ we

have

$$\operatorname{rem}\left(\boldsymbol{t} - \sum_{\boldsymbol{v}\in \mathrm{FP}} f_{\boldsymbol{v}}\boldsymbol{v}\right) = \mathbf{0}, \quad f \in \mathbb{F}_q[x],$$

and it follows that $\langle \mathrm{LT}(G_2)\rangle \subseteq \langle \mathrm{LT}(M)\rangle$. We want to prove that $\langle \mathrm{LT}(M)\rangle = \langle \mathrm{LT}(G_2)\rangle$ with respect to $\succeq$, and, thus, show that G_2 is a Gröbner basis for M with respect to $\succeq$. Note that if G_2 is a Gröbner basis, then FP is the footprint of M with respect to $\succeq$ and LT is the leading terms of the elements of G_2 by construction.

Let $x^\alpha \boldsymbol{e}_k$ be any element not in FP. If $\boldsymbol{e}_k \in \mathrm{LT}$, then clearly $x^\alpha \boldsymbol{e}_k \in \langle \mathrm{LT}(G_2)\rangle$. If $\boldsymbol{e}_k \in \mathrm{FP}$, then consider the maximal term $x^\beta \boldsymbol{e}_k \in \mathrm{FP}$ of which $x^\alpha \boldsymbol{e}_k$ is a multiple. That is, there exists some $x^\gamma \neq 1$ such that $x^\alpha \boldsymbol{e}_k = x^\gamma (x^\beta \boldsymbol{e}_k)$ $(\alpha = \gamma + \beta)$. This means that $x x^\beta \boldsymbol{e}_k$ appeared in MON, but $x x^\beta \boldsymbol{e}_k \notin \mathrm{FP}$, such that $x^\alpha \boldsymbol{e}_k$ is a multiple of $x x^\beta \boldsymbol{e}_k$. Thus, $x^\alpha \boldsymbol{e}_k$ is always a multiple of some element of LT, and, thus, and element of $\langle \mathrm{LT}(G_2)\rangle$. Clearly, if $x^\alpha \boldsymbol{e}_k \in \mathrm{FP}$, then $x^\alpha \boldsymbol{e}_k$ is not divisible by any $\boldsymbol{g}_i \in G_2$ by construction. It follows that FP and $\langle \mathrm{LT}(G_2)\rangle$ are disjoint sets, and their union cover everything. This means that FP has the desired structure of a footprint. We now need to show that FP is not too big, in the sense that if some $x^\alpha \boldsymbol{e}_k$ is in $\langle \mathrm{LT}(M)\rangle$, then $x^\alpha \boldsymbol{e}_k$ cannot be an element of FP.

Let $\boldsymbol{g} \in M$, but suppose that there does not exist any $\boldsymbol{g}_i \in G_2$ such that $\mathrm{LT}(\boldsymbol{g}_i)$ divides $\mathrm{LT}(\boldsymbol{g})$. This means that $\mathrm{LT}(\boldsymbol{g})$ must be in FP, since FP and $\langle \mathrm{LT}(G_2)\rangle$ are disjoint and cover everything. We can assume without loss of generality that $\boldsymbol{g}$ is reduced modulo the elements of G_2. Since $\mathrm{LT}(\boldsymbol{g}) \in \mathrm{FP}$, $\mathrm{LT}(\boldsymbol{g})$ must have been added to FP at some point before we reached $\boldsymbol{g}_k$ for G_2. Thus

$$\mathrm{LT}(\boldsymbol{g}_1) \leq \cdots \leq \mathrm{LT}(\boldsymbol{g}_i) \leq \mathrm{LT}(\boldsymbol{g}) \leq \mathrm{LT}(\boldsymbol{g}_{i+1}) \leq \ldots$$

for some i. At the point when the algorithm considers $\mathrm{LT}(\boldsymbol{g})$ the other monomials in $\boldsymbol{g}$ – that is, the monomials of $\boldsymbol{g} - \mathrm{LT}(\boldsymbol{g})$ – must already have been added to FP, since $\boldsymbol{g}$ is reduced. But it then follows that since $\boldsymbol{g} \in M$ we have

$$\mathbf{0} = \operatorname{rem}(\boldsymbol{g}) = \operatorname{rem}(\mathrm{LT}(\boldsymbol{g})) + \operatorname{rem}(\boldsymbol{g} - \mathrm{LT}(\boldsymbol{g})),$$

which means that the algorithm would add $\mathrm{LT}(\boldsymbol{g})$ to LT. This contradicts our assumption, and $\mathrm{LT}(\boldsymbol{g}_i)$ must divide $\mathrm{LT}(\boldsymbol{g})$ for some i, which means that $\mathrm{LT}(\boldsymbol{g}) \in \langle \mathrm{LT}(G_2)\rangle$. Thus, we also have $\langle \mathrm{LT}(G_2)\rangle \supseteq \langle \mathrm{LT}(M)\rangle$. It follows that G_2 is a Gröbner basis for M with respect to $\succeq$. Further, G_2 is a reduced Gröbner basis by construction, since every term of the elements of G_2 are reduced. ■

We will show how to use this algorithm in an easy example.

Example 3.1.11
Consider the module $M = \langle \boldsymbol{g}_1, \boldsymbol{g}_2 \rangle \subseteq (k[x])^2$, where

$$\begin{aligned} \boldsymbol{g}_1 &= (1, x+1), \\ \boldsymbol{g}_2 &= (0, x^2) \end{aligned}$$

is a Gröbner basis with respect to $\succeq_{\mathrm{POT}}$. We will use the FGLM extension to find a Gröbner basis for M with respect to $\succeq_{\mathrm{TOP}}$. First we set

$$\begin{aligned} \mathrm{MON} &:= \{\boldsymbol{e}_2, \boldsymbol{e}_1\}, \\ G_2 &:= \{\}, \\ \mathrm{LT} &:= \{\}, \end{aligned}$$

since $\boldsymbol{e}_1 \succeq_{\mathrm{POT}} \boldsymbol{e}_2$ and MON is ordered in ascending order. We then take $\boldsymbol{t} := \boldsymbol{e}_2$ and consider $\mathrm{rem}(\boldsymbol{e}_2) = \boldsymbol{e}_2$. It follows that we add $\boldsymbol{e}_2$ to FP, and

$$\begin{aligned} \mathrm{FP} &:= \{\boldsymbol{e}_2\}, \\ \mathrm{MON} &:= \{\boldsymbol{e}_1, x\boldsymbol{e}_2\}. \end{aligned}$$

We will show how we do these calculations in Singular.

```
>ring RP=2,x,(c,lp);
>vector g1=[1,x+1];
>vector g2=[0,x2];
>module G=g1,g2;
>reduce(gen(2),G);
[0,1]
>reduce(gen(1),G);
[0,x+1]
>reduce(x*gen(2),G);
[0,x]
>reduce(x*gen(1),G);
[0,x]
```

In Singular $\mathrm{gen}(1) = \boldsymbol{e}_1$ and $\mathrm{gen}(2) = \boldsymbol{e}_2$. We see that we also need to add $\boldsymbol{e}_1$ to FP, since $\mathrm{rem}(\boldsymbol{e}_1) = (x+1)\boldsymbol{e}_2$, whereafter

$$\begin{aligned} \mathrm{FP} &:= \{\boldsymbol{e}_2, \boldsymbol{e}_1\}, \\ \mathrm{MON} &:= \{x\boldsymbol{e}_2, x\boldsymbol{e}_1\}. \end{aligned}$$

So now we have $\boldsymbol{t} := x\boldsymbol{e}_2$ and $\text{rem}(x\boldsymbol{e}_2) = x\boldsymbol{e}_2 = -\boldsymbol{e}_2+(x+1)\boldsymbol{e}_2 = -\text{rem}(\boldsymbol{e}_2)+\text{rem}(\boldsymbol{e}_1)$, so we can write $\text{rem}(x\boldsymbol{e}_2)$ as a linear combination of the monomials in FP, which means that we add $x\boldsymbol{e}_2 - (-\boldsymbol{e}_2 + \boldsymbol{e}_1)$ to G_2, and

$$\begin{aligned} G_2 &:= \{-\boldsymbol{e}_1 + (x+1)\boldsymbol{e}_2\}, \\ \text{LT} &:= \{x\boldsymbol{e}_2\}, \\ \text{MON} &:= \{x\boldsymbol{e}_1\}. \end{aligned}$$

We now consider $\boldsymbol{t} := x\boldsymbol{e}_1$ and see that $\text{rem}(x\boldsymbol{e}_1) = x\boldsymbol{e}_2 = \text{rem}(x\boldsymbol{e}_2)$, and, thus, we have

$$\begin{aligned} G_2 &:= \{-\boldsymbol{e}_1 + (x+1)\boldsymbol{e}_2, (x-1)\boldsymbol{e}_1 + \boldsymbol{e}_2\}, \\ \text{LT} &:= \{x\boldsymbol{e}_2, x\boldsymbol{e}_1\}, \\ \text{MON} &:= \{\}, \end{aligned}$$

which means that the algorithm terminates. To sum up, we have found

$$G_2 = \begin{bmatrix} 1 & 1+x \\ 1+x & 1 \end{bmatrix} \tag{3.9}$$

which is a reduced Gröbner basis with respect to $\succeq_{\text{TOP}}$. To see that this is correct, we will check this in Singular.

```
>ring RT=2,x,(lp,c);
>module G=imap(RP,G); //maps G from RP to G in RT
>option(redSB) //forces std to give a reduced Gröbner basis
>module G2=std(G);print(G2);
1,  x+1,
x+1,1
```

This is the same as Equation (3.9). □

In the final chapter we will consider a method for decoding quasi-cyclic codes by using their Gröbner basis representation. We will discuss some of the weaknesses of this method, and show that if we choose a quasi-cyclic code with one generator, where the generator is of a specific form, then this method can work very well.

Chapter 4

Decoding of Quasi-Cyclic Codes

In this chapter we consider a method of decoding quasi-cyclic codes by using their Gröbner basis representation presented in Section 2.3. We will also discuss this method's weakness, and, thus, consider the Restriction-1 codes defined in Definition 2.3.9 on page 60, since our method works particulary well with these. First we will consider the general case. This chapter is based on [Lally, 2002] and [Lally, 2000, Chapter 6].

4.1 Decoding Quasi-Cyclic Codes

Let $\mathcal{C} \subseteq (R/I)^l$, where $I = \langle x^m - 1 \rangle$, be a code of index l and length ml. Let $\tilde{G} = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_l\} \subseteq R^l$ be a reduced Gröbner basis for the submodule $\tilde{\mathcal{C}} \subseteq R^l$ with respect to $\succeq_{\text{POT}}$ with the properties described in Theorem 2.3.5 on page 58. The mapping

$$\phi : R^l \to (R/I)^l$$

given by $\phi(\boldsymbol{g}) = \boldsymbol{g} \mod x^m - 1$ gives the corresponding Gröbner basis G for $\mathcal{C}$.

Suppose that $\gcd(m, q) = 1$ and that $g_{ii} \neq x^m - 1$ for every $i = 1, \ldots, l$. Let

$$\boldsymbol{v} = (v_1, \ldots, v_l) \in (R/I)^l$$

be any information vector where $\deg(v_i) < m - \deg(g_{ii})$, and let the codeword $\boldsymbol{c} = (c_1, \ldots, c_l) \in \mathcal{C}$ be given by

$$\begin{aligned} \boldsymbol{c} &= v_1\boldsymbol{g}_1 + \cdots + v_l\boldsymbol{g}_l \\ &= (v_1 g_{11}, v_1 g_{12} + v_2 g_{22}, \ldots, \sum_{i=1}^{l} v_i g_{il}) \mod x^m - 1, \end{aligned} \tag{4.1}$$

where modulo is done in every block. Suppose that $\boldsymbol{c}$ is transmitted through a noisy channel resulting in a received word $\boldsymbol{r} = (r_1, \ldots, r_l) \in (R/I)^l$ containing some errors,

$$\begin{aligned} \boldsymbol{r} &= \boldsymbol{c} + \boldsymbol{e}, \\ r_i &= c_i + e_i, \quad 1 \leq i \leq l, \end{aligned}$$

where $\boldsymbol{e} = (e_1, \ldots, e_l) \in (R/I)^l$ is an error vector. Denote by d_i^*, $1 \leq i \leq l$ the BCH bound of the cyclic code generated by g_{ii} given in Theorem 2.2.6 on page 54,

$$d_i^* = \sharp ConsecutiveRoots(g_{ii}) + 1.$$

From Equation (4.1) we see that

$$c_1 = v_1 g_{11} \mod x^m - 1$$

is a codeword in the code generated by g_{11}. This means that we can decode r_1 correctly to c_1 if

$$\mathrm{w}(e_1) \leq \left\lfloor \frac{d_1^* - 1}{2} \right\rfloor.$$

If g_{11} generates a Reed-Solomon code, then we can use the method described in Chapter 3. After decoding c_1 we can calculate the information polynomial $v_1 = c_1/g_{11}$. Now, consider

$$r_2 = c_2 + e_2 = (v_1 g_{12} + v_2 g_{22}) + e_2 \mod x^m - 1$$

and define

$$r_2' = r_2 - v_1 g_{12} = v_2 g_{22} + e_2 \mod x^m - 1.$$

We can decode r_2' to $c_2' = v_2 g_{22}$, which is a codeword in the code generated by g_{22}, if

$$\mathrm{w}(e_2) \leq \left\lfloor \frac{d_2^* - 1}{2} \right\rfloor.$$

We can then calculate the codeword $c_2 = c_2' + v_1 g_{12} \mod x^m - 1$ and the information polynomial $v_2 = c_2'/g_{22}$. We can recursively continue to decode c_i, $2 \leq i \leq l$ one at a time by defining

$$r_i' = r_i - \sum_{j=1}^{i-1} v_j g_{ji} = v_i g_{ii} + e_i \mod x^m - 1,$$

which we can decode to c_i' if

$$\mathrm{w}(e_i) \leq \left\lfloor \frac{d_i^* - 1}{2} \right\rfloor,$$

and we find

$$c_i = c_i' + \sum_{j=1}^{i-1} v_j g_{ji} \mod x^m - 1$$

and

$$v_i = \frac{c_i'}{g_{ii}}.$$

Note that if $g_{ii} = x^m - 1$ for any i, then we will simply have $v_i g_{ii} = 0 \mod x^m - 1$, which means that we can disregard the information polynomial v_i.

This decoding works well if the errors are spread out over the whole received word, but it will fail if

$$\mathrm{w}(e_i) > \left\lfloor \frac{d_i^* - 1}{2} \right\rfloor$$

for any $i = 1, \ldots, l$, since we cannot calculate the rest of the codeword $c_i, \ldots, c_l$ if this happens.

4.2 1-Generator Quasi-Cyclic Codes

In this section we will restrict ourself to quasi-cyclic codes generated by one generator

$$\boldsymbol{g} = (g_1, \ldots, g_l) \subseteq (R/I)^l.$$

In particular we will consider quasi-cyclic codes $\mathcal{C}$ with a 1-level Gröbner basis as defined in Definition 2.3.7 on page 60. That is, we have a generator of the form

$$\boldsymbol{g} = (g_1, f_1 g_1, \ldots, f_{l-1} g_1) \subseteq (R/I)^l, \quad f_i \in R \tag{4.2}$$

where g_1 divides $x^m - 1$ and $\deg(f_i) < m - \deg(g_1)$, $1 \leq i \leq l - 1$, by Theorem 2.3.5 on page 58. This basically means that that every entry in

$\boldsymbol{g}$ is a codeword in the cyclic code generated by g_1, or that every entry generates a subcode of $\langle g_1 \rangle$. Let v be any information polynomial, where $\deg(v) < m - \deg(g_1)$, such that the codeword $\boldsymbol{c} = (c_1, \ldots, c_l) \in \mathcal{C}$ is given by

$$\boldsymbol{c} = v\boldsymbol{g} = (vg_1, vf_1g_1, \ldots, vf_{l-1}g_1) \mod x^m - 1.$$

The codeword $\boldsymbol{c}$ is transmitted and received as

$$\begin{aligned} \boldsymbol{r} &= \boldsymbol{c} + \boldsymbol{e}, \\ r_i &= c_i + e_i, \quad 1 \leq i \leq l, \end{aligned}$$

where $\boldsymbol{e} = (e_1, \ldots, e_l) \in (R/I)^l$ is some error vector. The first received polynomial

$$r_1 = c_1 + e_1 = vg_1 + e_1 \mod x^m - 1$$

can be decoded correctly to c_1 if

$$\mathrm{w}(e_1) \leq \left\lfloor \frac{d_1^* - 1}{2} \right\rfloor.$$

If g_1 generates a Reed-Solomon code, then we can use the method described in Chapter 3 to decode r_1. Because of the structure of the generator in Equation (4.2) we can use c_1 to find the rest of the codeword by the equation

$$c_i = vf_{i-1}g_1 = f_{i-1}c_1 \mod x^m - 1, \quad 2 \leq i \leq l.$$

It follows that this algorithm works very well if we have few errors in the first block. Even if all the other blocks are missing, we are still able to find the correct codeword $\boldsymbol{c}$.

Note that since we can have $\gcd(f_i, x^m - 1) \neq 1$ for some i, we cannot be sure that $\langle g_1 \rangle = \langle f_i g_1 \rangle$, which means that we might not be able to find v by decoding any r_i other than r_1 using the same decoder. This is not very practical, since we would need a decoder for each block. Next we will consider codes where this problem does not occur; that is, codes where we can use the same decoder for every block.

We will now restrict ourself even further to consider Restriction-1 codes as defined in Definition 2.3.9 on page 60. Thus, we have a generator for the quasi-cyclic code $\mathcal{C}$ of the form

$$\boldsymbol{g} = (f_1 g, \ldots, f_l g) \in (R/I)^l, \quad f_i \in R,$$

where g divides $x^m - 1$, $\gcd(f_i, x^m - 1) = 1$, and $\deg(f_i) < m - \deg(g)$, $1 \leq i \leq l$. Note that this means that $\gcd(f_i g, x^m - 1) = g$, whereby it follows that every $f_i g$, $1 \leq i \leq l$, generates the same cyclic code as g. From Theorem 2.3.10 on page 60 we have a lower bound on the minimum distance of $\mathcal{C}$,

$$d_{\min} \geq ld^* = l(\sharp ConsecutiveRoots(g) + 1),$$

where d^* is the BCH bound of $\langle g \rangle$. Let v be any information polynomial, where $\deg(v) < m - \deg(f_i g)$ for every $i = 1, \ldots, l$, such that the codeword $\boldsymbol{c} = (c_1, \ldots, c_l) \in \mathcal{C}$ is given by

$$\boldsymbol{c} = v\boldsymbol{g} = (vf_1 g, vf_2 g, \ldots, vf_l g) \mod x^m - 1.$$

The codeword $\boldsymbol{c}$ is transmitting and received as

$$\begin{aligned} \boldsymbol{r} &= \boldsymbol{c} + \boldsymbol{e}, \\ r_i &= c_i + e_i, \quad 1 \leq i \leq l, \end{aligned}$$

where $\boldsymbol{e} = (e_1, \ldots, e_l) \in (R/I)^l$ is some error vector. This means that if we can decode any

$$r_i = c_i + e_i = vf_i g + e_i \mod x^m - 1$$

to the right codeword c_i, which we can when

$$\mathrm{w}(e_i) \leq \left\lfloor \frac{d_i^* - 1}{2} \right\rfloor,$$

then we can find all the other blocks by the equation

$$c_j = c_i f_i^{-1} f_j \mod x^m - 1 \tag{4.3}$$

for each $j \neq i$, $1 \leq j \leq l$, since $\gcd(f_i, x^m - 1) = 1$. If we decode any block incorrectly, then this will result in

$$d(\boldsymbol{c}, \boldsymbol{r}) > \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor,$$

and we will, thus, know that we made a mistake. This algorithm is very effective if our received word contains a lot of errors, but where we have just one block with few errors, and it is particically effective for erasures. We can then use this block to decode the rest of the codeword. To end this chapter

we will give a more formal decoding algorithm for decoding Restriction-1 codes. The algorithm can correct at most

$$\min \left\{ \begin{array}{l} l \left\lfloor \frac{d^*-1}{2} \right\rfloor + l - 1 \\ \\ \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor \end{array} \right\}$$

errors. Note that

$$l \left\lfloor \frac{d^* - 1}{2} \right\rfloor + l - 1 = lt + l - 1 = (t+1)(l-1) + t,$$

where $t = \left\lfloor \frac{d^*-1}{2} \right\rfloor$, which means that in the worst-case scenario we have $l-1$ blocks with $t+1$ errors, but still one block with only t errors, which we can correct.

Algorithm 4.2.1
Input
$\boldsymbol{r} = (r_1, \ldots, r_l)$
$\boldsymbol{g} = (f_1 g, \ldots, f_l g)$
$d_{\min}$

Output
$\boldsymbol{c} = (c_1, \ldots, c_l)$

Initialize
$\boldsymbol{c} := (0, \ldots, 0)$
$i := 1$

Loop
WHILE $i \leq l$ DO
    Decode r_i as a cyclic codeword in $\langle g \rangle \subseteq R/I$
    IF r_i decodes to c_i' THEN
        $c_j' := c_i' f_i^{-1} f_j \mod x^m - 1$ for all $1 \leq j \neq i \leq l$
        $\boldsymbol{c}' := (c_1', \ldots, c_l')$
        IF $d(\boldsymbol{c}', \boldsymbol{r}) \leq \lfloor (d_{\min} - 1)/2 \rfloor$ THEN
            $\boldsymbol{c} := \boldsymbol{c}'$ ELSE
            $i := i + 1$
    ELSE
        $i := i + 1$

Note that when

$$\mathrm{w}(\boldsymbol{e}) \leq \min \left\{ \begin{array}{l} l\left\lfloor \frac{d^*-1}{2} \right\rfloor + l - 1 \\ \\ \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor \end{array} \right\},$$

we clearly also have

$$\mathrm{w}(\boldsymbol{e}) \leq l \left\lfloor \frac{d^*-1}{2} \right\rfloor + l - 1,$$

which we already noted means that we have at least one error polynomial e_i satisfying

$$\mathrm{w}(e_i) \leq \left\lfloor \frac{d^*-1}{2} \right\rfloor, \tag{4.4}$$

meaning that we can decode r_i correctly to c_i. Suppose

$$\mathrm{w}(e_i) > \left\lfloor \frac{d^*-1}{2} \right\rfloor$$

for any $i = 1, \ldots, l$, and r_i is decoded incorrectly to $c_i' \neq c_i$. Then when we consider the complete codeword $\boldsymbol{c}' = (c_1', \ldots, c_l') \in \mathcal{C}$, constructed from Equation (4.3), we will get

$$d(\boldsymbol{c}', \boldsymbol{r}) > \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor,$$

since there exists a unique codeword $\boldsymbol{c} \in \mathcal{C}$, where $d(\boldsymbol{c}, \boldsymbol{r}) \leq \lfloor (d_{\min}-1)/2 \rfloor$. Thus, the decoding was wrong and we set $i := i+1$ in the algorithm and try the next block. If

$$l \left\lfloor \frac{d^*-1}{2} \right\rfloor + l - 1 < \mathrm{w}(\boldsymbol{e}) \leq \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor,$$

and we have at least one block satisfying Equation (4.4), then we can still decode $\boldsymbol{r}$ from this block and Equation (4.3). So, in general we can decode $\boldsymbol{r}$ to the correct $\boldsymbol{c} \in \mathcal{C}$ if at least one block can be correctly decoded and if the total number of errors is below $\lfloor (d_{\min}-1)/2 \rfloor$.

Bibliography

[Cox et al., 2005] Cox, D., Little, J., and O'Shea, D. (2005). *Using Algebraic Geometry*. Springer, second edition.

[Cox et al., 2007] Cox, D., Little, J., and O'Shea, D. (2007). *Ideals, Varieties, and Algorithms*. Springer, third edition.

[Fitzpatrick, 1997] Fitzpatrick, P. (1997). Solving a multivariable congruence by change of term order. *J. Symbolic Computation*, 24.

[Huffman and Pless, 2003] Huffman, W. C. and Pless, V. (2003). *Fundamentals of Error Correcting Codes*. Cambridge.

[Justesen and Høholt, 2000] Justesen, J. and Høholt, T. (2000). *A Course in Error-Correcting Codes*. European Mathematical Society.

[Lally, 2000] Lally, C. (2000). *Application of the theory of Gröbner bases to the study of quasicyclic codes*. PhD thesis, National University of Ireland.

[Lally, 2002] Lally, K. (2002). Quasicyclic codes – some practical issues. *ISIT*, June 30-July 5, 2002.

[Lally and Fitzpatrick, 2001] Lally, K. and Fitzpatrick, P. (2001). Algebraic structure of quasicyclic codes. *Discrete Applied Mathematics*, 111.

[Moro et al., 2007] Moro, E. M., Gómez, C. M., and Benito, D. R. (2007). Bases de gröbner: Aplicaciones a la codificación algebraica. Technical report.