UNIVERSITÀ DEGLI STUDI DI PADOVA
Facoltà di Scienze MM. FF. NN

UNIVERSITE BORDEAUX 1
U.F.R. Mathématiques et Informatique

Master Thesis

Vo Ngoc Thieu

Reduction Modulo Ideals and Multivariate Polynomial Interpolation

Advisor: Prof. Jean-Marc Couveignes

June 28, 2013

Contents

Acknowledgments

I am special grateful to my advisor, Prof. Jean-Marc Couveignes, for proposing this project, for his strong support for my work during the last semester, and for many comments on writing a good scientific English. Working under his enthusiasm and kindly guidance is an honor for me. Without him this thesis can not be completed.

I would like to thank Mathias Lederer for suggesting an improvement of his origin paper [6] and for giving some useful remarks, and hence help me read the paper in a favorable way.

I want to express my gratitude to ERASMUS MUNDUS - ALGANT program, who funded my Master degree in Padova, Italy and Bordeaux, France, so that give me a great chance to study with famous professors in Europe and excellent students from several countries over the world.

Finally I would like to thank my family and my friends for support and encouragement.

List of Algorithms

Introduction

Interpolation is the problem of constructing a function P that must be in a given linear space (of simple functions) from a given set of local data. It is called Polynomial Interpolation if P is found in a polynomial algebra. In this case, P is called the interpolating polynomial, and the given set of local data is the interpolation condition. Polynomial Interpolation is a fundamental method in Numerical Analysis, in which the interpolation condition is usually obtained from the evaluations of another, but difficult, function f. So that f can be approximated by the interpolating polynomial P and then solve the problem in approximate, but particular, way.

Interpolation in Univariate Polynomial is the simplest and most classical case. It dates back to Newton's fundamental interpolation formula and the Lagrange interpolating polynomials. Polynomial Interpolation in several variables is much more intricate. This topic has probably started in the second-half of the 19th century with work of W.Borchardt in [9] and L. Kronecker in [10]. However its extreme development has only started since the final quarter of the 20th century within a strong connection to the development of theory of polynomial ideals. It is currently an active area of research.

The purpose of this Master thesis is using reduction modulo ideals in polynomial algebras to solve Polynomial Interpolation Problem. In order to do that, the solution of univariate case in Chapter 1 and the theory of Gröbner bases in Chapter 2 are preparation steps. In Chapter 1, we present an algorithm for solving the Interpolation Problem in univariate case in quasi-linear time computation. Also in Chapter 1, a particular case of Multivariate Polynomial Interpolation when the interpolation set forms a tensor product of grids is solved in quasi-linear time. The general case requires a more geometric study of interpolation conditions. In Chapter 2, we introduce the theory of Gröbner bases and reduction modulo an ideal in polynomial algebra. It leads us to far more geometric interpretation of the ideal associated with finite set of points in affine space. So that reduction is the main tool to solve the Multivariate Polynomial Interpolation in Chapter 3.

Chapter 3 is the main part of the thesis in which solving Polynomial Interpolation Problem by using reduction method is presented. The method is not only concerned with finding the best solution in some sense for each interpolation condition but also with determining explicitly the zero dimension ideal that is defined by the interpolation set. The first section of Chapter 3 is based on the work of M. Lederer in [6] and J.C. Faugere, P.Gianni, D.Lazard, T.Mora in [8]. After solving the Multivariate Polynomial Interpolation and estimating algorithmic costs, we end the thesis by comparison with the other known methods.

Chapter 1

Polynomial Interpolation Problems

Assume that there is an unknown real function $f(x)$ with its value at $n+1$ distinct points $x_0, x_1, ..., x_n \in \mathbb{R}$ corresponding $u_0, u_1, ..., u_n \in \mathbb{R}$. The Interpolation Problem is to construct a function $P(x)$ passing through these points, i.e, to find a function $P(x)$ such that the interpolation requirements

$$P(x_i) = u_j, \ \forall i = 0, 1, ..., n$$

are satisfied. In this article, we look for the most important case, when the function P is a polynomial.

In that case, the real numbers field $\mathbb{R}$ can be replaced by any commutative unitary ring R.

1.1 Univariate case

Through this section, we denote R an unitary commutative ring.

1.1.1 Lagrange Interpolating Polynomials

Problem 1 (Univariate Interpolation Problem)**.** *Given $n \in \mathbb{N}$; $u_0, u_1, ..., u_{n-1} \in R$ such that $u_i - u_j$ are units in R; $v_0, v_1, ..., v_{n-1} \in R$, compute polynomial $f \in R[x]$ of degree less than n that satisfies the interpolation conditions:*

$$f(u_i) = v_i, \ \forall i = 0, 1, ..., n-1$$

If we do not limit the degree of the interpolation polynomial, it is easy to see that there are infinite many polynomials that interpolate the set $\{(u_i, v_i), i = 0, ..., n-1\}$. But there is exactly one of these polynomials with the lowest possible degree. It will be described explicitly in the following result:

Proposition 1. *As the notation in Problem 1, there always exists an unique polynomial of degree less than n which solves Problem 1. It has the form:*

$$f(x) = \sum_{j=0}^{n-1} v_j \prod_{i=0, i\neq j}^{n-1} \frac{x - u_i}{u_j - u_i} \tag{1.1}$$

f is called Lagrange Interpolating Polynomial.

Proof: Clearly. □

Proposition 1 shows us a naive way to solve Problem 1. In this way, we only compute the standard form of f defined by (1.1).

Algorithm 1 Naive Interpolation

Require: $n \in \mathbb{N}$; $n > 0$; $u_0, ..., u_{n-1} \in R$ such that $u_i - u_j$ units in $R\ \forall i \neq j$; $v_0, ..., v_{n-1} \in R$.
Ensure: $f \in R[x]$.

1: Compute $l_j = \prod_{i \neq j,\, i=0}^{n} \frac{x-u_i}{u_j-u_i},\ j = 0, ..., n-1.$

2: Return $f := \sum_{j=0}^{n-1} v_j l_j$.

Theorem 1. *Algorithm 1 takes $O(n^2)$ operations in R.*

Proof: To compute l_j,we notice that

$$l_j = \frac{m}{(x-u_j)m'(u_j)}$$

where $m = (x-u_0)(x-u_1)...(x-u_{n-1})$.
Firstly, we compute $(x-u_0), (x-u_0)(x-u_1), ..., m = (x-u_0)(x-u_1)...(x-u_{n-1})$. They are just products of monic polynomials of degree 1 and monic polynomials of degree $i,\ i = 1, ..., n-2$. So they take $\sum_{i=1}^{n-1} 2i - 1 = (n-1)^2 = O(n^2)$ operations in R. For each $j = 0, 1, ..., n-1$, we divide m by $(x-u_j)$ and then evaluate $\frac{m}{x-u_j}$ at $x = u_j$ thanks to Horner scheme, taking $O(n)$ operators. And then, we divide $\frac{m}{x-u_j}$ by the last value to obtain l_j. This amounts to $n(O(n) + O(n)) = O(n^2)$ operations in R.
Finally, computing the linear combination $f = \sum_{j=0}^{n-1} v_j l_j$ takes $O(n^2)$ more operations. Hence, the arithmetic cost for Algorithm 1 is $O(n^2)$. □

1.1.2 Fast Interpolation

In the last subsection, we compute the Lagrange interpolation polynomial

$$f = \sum_{j=0}^{n-1} v_j \prod_{i=0, i\neq j}^{n-1} \frac{x-u_i}{u_j-u_i} = \sum_{j=0}^{n-1} \frac{v_j}{m'(u_j)} \cdot \frac{m}{(x-u_j)}$$

in $O(n^2)$ operations in R. In this subsection, a cheaper algorithm will be given.

For convenience, in this section, we assume that $n = 2^k$ for some $k \in \mathbb{N}$. The idea of the fast interpolation algorithm is to split the set of interpolation points $\{u_0, ..., u_{n-1}\}$ into two halves of equal cardinality and to proceed recursively with each half. This leads us to a binary tree of depth k with roots $\{u_0, ..., u_{n-1}\}$ and the singletons $\{u_i\}$ for $0 \leq i \leq n-1$.

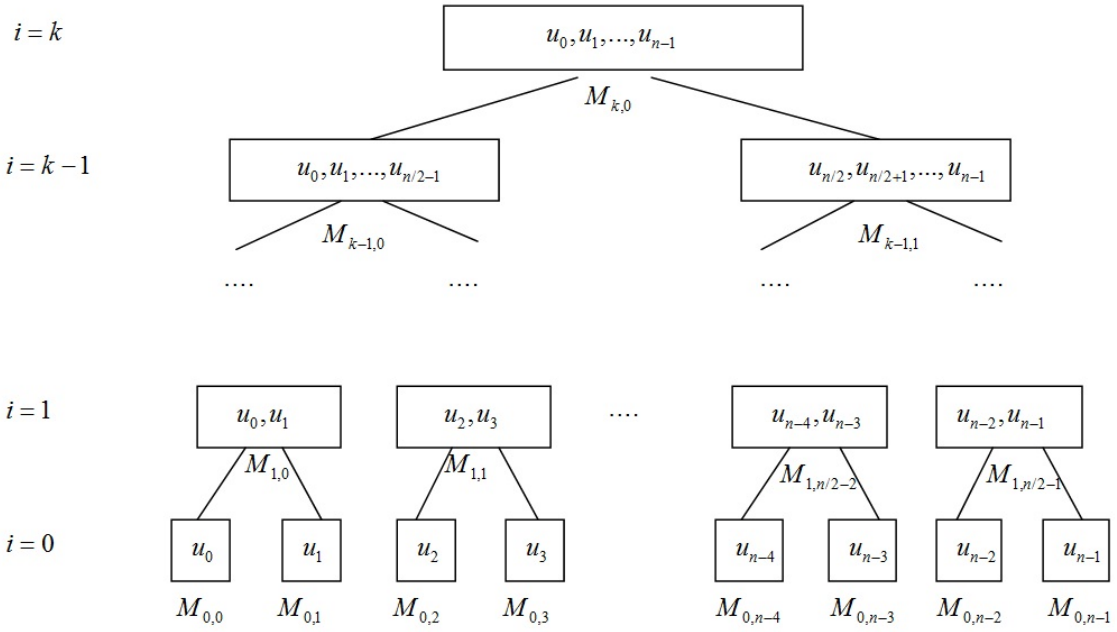


Figure 1.1: Binary tree of interpolation set.

This idea can be described by the diagram above.

Now we define the subproducts $M_{i,j}$ of m recursively as:

$$M_{0,j} = (x - u_j) \qquad j = 0, ..., n-1$$
$$M_{i,j} = M_{i-1,2j}.M_{i-1,2j+1} \qquad i = 1, ..., k; j = 0, ..., \tfrac{n}{2^i} - 1$$

Thus

$$M_{i,j} = (x - u_{j.2^i})(x - u_{j.2^i+1})...(x - u_{j.2^i+(2^i-1)}),\ i = 0, ..., k; j = 0, ..., \frac{n}{2^i} - 1 \qquad (1.2)$$

is a subproduct of m with 2^i factors and monic square-free polynomial of degree 2^i.

Algorithm 2 Building $M_{i,j}$

Require: $u_0, ..., u_{n-1}$ as above.
Ensure: The all $M_{i,j}$ as in (1.2).

```
for all 0 ≤ j ≤ n do
    M_{0,j} := (x − u_j)
end for
for all 1 ≤ i ≤ k do
    for all 0 ≤ j ≤ n/2^i − 1 do
        M_{i,j} := M_{i−1,2j}.M_{i−1,2j+1}
    end for
end for
Return all such M_{i,j}.
```

Lemma 1. *Algorithm 2 computes all subproducts $M_{i,j}$ in $O(M(n)\log n)$ operations in R. Where $M(n)$ is the number of arithmetic operations in R to compute product of two polynomials of degree less than n in $R[x]$.*

Proof: The number of loops in algorithm 2 is $k-1 = O(\log n)$.

On the i-th loop, we must compute $\frac{n}{2^i}$ products of two polynomials of degree 2^i

$$M_{i,j} := M_{i-1,2j}.M_{i-1,2j+1},\ j = 0, ..., \frac{n}{2^i} - 1$$

So it take $\frac{n}{2^i}.M(2^i) \leq M(n)$ operators in R. Hence, the arithmetic cost of algorithm 2 is $M(n)O(\log n) = O(M(n)\log n)$. □

The core of the fast interpolation algorithm is the relation between the interpolation polynomial of the set $\{(u_i, v_i), i = 0, ..., n-1\}$ and the other two interpolation polynomials of two halves subsets $\{(u_i, v_i), i = 0, ..., \frac{n}{2} - 1\}$ and $\{(u_i, v_i), i = \frac{n}{2}, ..., n-1\}$. It leads us come to the following lemma.

Lemma 2. *Let $u_0, u_1, ..., u_{n-1} \in R$ as notation above, $c_0, c_1, ..., c_{n-1} \in R$. Denote:*

$$f = \sum_{i=0}^{n-1} c_i.\frac{m}{x-u_i}$$

$$f_{k-1,0} = \sum_{i=0}^{\frac{n}{2}-1} c_i.\frac{M_{k-1,0}}{x-u_i}$$

$$f_{k-1,1} = \sum_{i=\frac{n}{2}}^{n-1} c_i.\frac{M_{k-1,1}}{x-u_i}$$

Then

$$f = M_{k-1,1}.f_{k-1,0} + M_{k-1,0}.f_{k-1,1} \tag{1.3}$$

Proof: Because for each $j = 0, 1, ..., n-1$ the values of both sides of (1.3) are equal to $c_j.\prod_{i=0;i\neq j}^{n-1}(u_j - u_i)$. □

Now we consider the Lagrange interpolation formula

$$f = \sum_{j=0}^{n-1} \frac{v_j}{m'(u_j)}.\frac{m}{(x-u_j)}$$

Denote $c_j = \frac{v_j}{m'(u_j)}$ then the lemma 2 can help us to compute Lagrange formula recursively. The only problem now is computing $m'(u_j)$ for all $j = 0, 1, ..., n-1$. Horner scheme can help us to evaluate m' at n points $u_0, u_1, ..., u_{n-1}$, but it takes $O(n^2)$ operations in R. In fact, one can use the binary tree of subproducts $M_{i,j}$ to give the other method which is really faster.

Theorem 2. *Algorithm 3 evaluates f at n points $u_0, ..., u_{n-1}$ correctly in $O(M(n)\log n)$ operations in R.*

Algorithm 3 Fast Multievaluation

Require: $f \in R[x]$ of degree less than n; $u_0, ..., u_{n-1}$ as above; $M_{i,j}$ as in (1.2).
Ensure: $f(u_0), ..., f(u_{n-1})$.

1: **if** $n = 1$ **then**
2: return f
3: **end if**
4: $r_{k,0} := f$.
5: $i := k - 1$.
6: **while** $0 \leq i \leq k$ **do**
7: $r_{i,2j} := r_{i+1,j} \mod M_{i,2j}$.
8: $r_{i,2j+1} := r_{i+1,j} \mod M_{i,2j+1}$.
9: $i := i - 1$.
10: **end while**
11: Return $r_{0,0}, ..., r_{0,n-1}$.

Proof: For any $j = 0, ..., n-1$, from the algorithm, we have

$$r_{0,j} = r_{1,j_1} \, mod \, M_{0,j}$$

for some j_1 such that M_{0,j_1} divides M_{1,j_1}. We also have

$$r_{1,j_1} = r_{2,j_2} \, mod \, M_{1,j_1}$$

for some j_2 such that M_{1,j_1} divides M_{2,j_2}. So

$$r_{0,j} = r_{2,j_2} \, mod \, M_{0,j}$$

for some j_2. Repeat this process $k-1$ times, we imply that

$$r_{0,j} = f \, mod \, M_{0,j} = f(u_j)$$

The correctness of algorithm is proved. Now we estimate the arithmetic cost.

The length of loops of **while** round from line 6 to line 10 is equal to $k = O(n)$. For each next two loops, for example the i-th and $(i+1)$-th loop, the number of divisions with remainder to compute $r_{i,j}$ for all j is equal to two times the number of divisions with remainder to compute all $r_{i+1,j}$. But $\deg M_{i+2,j} = 2.\deg M_{i+1,j}$ and $\deg r_{i+1,j} < 2.\deg M_{i,j}$. Thus the cost of i-th loop is less than or equal to the cost of $(i+1)$-th loop, then inductively, the cost of $(k-1)$-th loop.

In $(k-1)$-th loop, $i = k-1$, we divide with remainder f of degree less than n by $M_{k-1,0}$ and $M_{k-1,1}$ of degree $\frac{n}{2}$. So it takes $2.O(M(\frac{n}{2})) = O(M(n))$ operations in R.

Hence, the arithmetic cost of algorithm 3 is $O(M(n)\log n)$. □
Putting all things together,we obtain the following fast interpolation algorithm.

Theorem 3. *Algorithm 4 solves correctly the univariate interpolation problem in $O(M(n)\log n)$ operations in R.*

Algorithm 4 Fast Interpolation

Require: $u_0, ..., u_{n-1}; v_0, ..., v_{n-1} \in R$ as above, where $n = 2^k$ for some $k \in \mathbb{N}$.
Ensure: $f \in R[x]$ of degree less than n such that $f(u_i) = v_i$ for all i.

1: Use algorithm 2 to compute all the $M_{i,j}$.
2: $m := M_{k,0}$.
3: Use algorithm 3 to evaluate the polynomial m' at $u_0, ..., u_{n-1}$.
4: $c_i := \frac{v_i}{m'(u_i)}$
5: $f_{0,j} := u_j$
6: **for** $i = 0 \to k - 1$ **do**
7: **for** $j = 0 \to \frac{n}{2^{i+1}} - 1$ **do**
8: $f_{i+1,j} := M_{i,2j}.f_{i,2j+1} + M_{i,2j+1}.f_{i,2j}$
9: **end for**
10: **end for**
11: Return $f_{k,0}$.

Proof: The correctness of algorithm 4 follows directly from lemma 2.

Thanks to lemma 1 and theorem 2, the arithmetic cost of line 1 and line 3 are equal to $O(M(n)\log n)$. Equivalently, the cost of step 3 is also equal to $O(M(n)\log n)$. Hence, the arithmetic cost of algorithm 4 is $O(M(n)\log n)$. □

If the ring R admits the Fast Fourier Transform to compute the product of any two polynomials of degree less than n in $M(n) = O(n \log n)$ operations in R, then the Fast Interpolation algorithm takes $O(n(\log n)^2)$ operations in R.

1.2 Multivariate case

Univariate Interpolation is a very classical topic. In the last section we have described a really fast algorithm to solve the Univariate Interpolation in quasi-linear time. However, Interpolation by polynomials in several variables is much more intricate and is a subject which is currently an active area of research. In this section, we will state the Multivariate Interpolation Problem, give the existence of solutions. And then we will describe an algorithm to solve essentially a simple case in which the set of interpolation points is a Cartesian product.

1.2.1 Introduction

Problem 2 (Multivariate Interpolation Problem)**.** *Fix the commutative field k, and positive integers d, N. Given $N+1$ interpolation points $X_0, X_1, ..., X_N \in k^d$ and $N+1$ scalar values $u_0, u_1, ..., u_N \in k$. Find the polynomial $P \in k[x_1, ..., x_d]$ with smallest possible degree, that satisfies the interpolation conditions:*

$$P(X_i) = u_i, \ \forall i = 0, ..., N \tag{1.4}$$

Assume the solution for Problem 2 is the polynomial P of degree n in $k[x_1, ..., x_d]$ of the form:

$$P(X) = \sum_{I \in \mathbb{N}^d; |I| \leq n} c_I X^I$$

where

$X = (x_1, ..., x_d)$
$I = (i_1, ..., i_d)$
$X^I := x_1^{i_1} ... x_d^{i_d}$
$|I| = i_1 + ... + i_d$

The number of coefficients c_I that occur in P which we must find is

$$card\{I \in \mathbb{N}^d \ s.t \ |I| \leq n\} = \begin{pmatrix} n+d \\ d \end{pmatrix}$$

Thus the uniqueness of solution for problem 2 requires at least $\begin{pmatrix} n+d \\ d \end{pmatrix}$ values of polynomial P at $\begin{pmatrix} n+d \\ d \end{pmatrix}$ distinct points.

Different from the univariate case, the distinctness of $N+1 = \begin{pmatrix} n+d \\ d \end{pmatrix}$ interpolation points is not enough to be sure that the interpolating polynomial always exist and of degree less than or equal to n.

Example 1. For $d = 2, n = 1, N = 3$, in the plane k^2, let three distinct points $O(0,0); A(0,-1); B(0,1)$ in the axis Ox. Then there is no polynomial in $k[x,y]$ of degree less than or equal to 1 that interpolates $O, A, B \in k^2$ with the corresponding scalar values $0; 1; 1 \in k$. Otherwise, there is a polynomial $P(x,y) = a + bx + cy \in k[x,y]$ satisfies assumption, then:

$$\begin{cases} P(0;0) = a = 0 \\ P(0;-1) = a - b = 1 \\ P(0;1) = a + b = 1 \end{cases}$$

But the last system has no solution (contradiction!!!).

However, if we do not limit the degree of interpolating polynomial, then we can choose $Q(x,y) = x^2$.

Return to the general case, consider the k-linear map:

$$eva : k[x_1, ..., x_d] \to k^{N+1}, \ \ f \mapsto (f(X_0), ..., f(X_N))$$

It is called the *evaluation map*. The restriction of eva to $k_n[x_1, ..., x_d]$ which is the set of all polynomials of degree less than or equal to n is a k-linear map between k-vector spaces of dimension $N+1 = \begin{pmatrix} n+d \\ d \end{pmatrix}$

$$eva|_{k_n[x_1,...,x_d]} : k_n[x_1, ..., x_d] \to k^{N+1}$$

Thus it is bijective if and only if it is surjective if and only if it is injective. In this case, we call the points $X_0, ..., X_N$ poised.

Definition 1 (Poised). *The set of* $N+1 = \begin{pmatrix} n+d \\ d \end{pmatrix}$ *points* $X_0, ..., X_N \in k^d$ *is called poised if the* k*-linear map* $eva|_{k_n[x_1,...,x_d]}$ *is bijective. Namely, problem 2 always admits an unique solution of degree less than or equal* n *for any given scalar value* $u_0, ..., u_N \in k$.

Next we analyze the map $\phi := eva|_{k_n[x_1,...,x_d]}$ in the matrix terms to get more practical viewing of the poiseness.

$$\phi : k_n[x_1, ..., x_d] \to k^{N+1}, \;\; f \mapsto (f(X_0), ..., f(X_N))$$

Clearly that the set $\mathbf{B}(X) := \{X^I \,|\, I \in \mathbb{N}^d, |I| \leq n\}$ forms the canonical basis of $k_n[x_1, ..., x_d]$. So that each polynomial f in $k_n[x_1, ..., x_d]$ can be written by

$$f(X) = (c_I)_I.\mathbf{B}(X)^t$$

where $\mathbf{B}(X)$ is seen as a row vector of $k[x_1, ..., x_d]^{N+1}$, and $(c_I)_I$, $I \in \mathbb{N}^d$, $|I| \leq n$, the coefficients of f. Under these terms, the matrix form of $\phi(f)$ is

$$\phi(f) = (c_I)_I.A$$

where

$$A = \left[\mathbf{B}(X_0)^t \; \mathbf{B}(X_1)^t \; ... \; \mathbf{B}(X_N)^t\right]$$

is a square matrix of size $N+1$ with entries in k. A is exactly the matrix representation of ϕ over the basis $\mathbf{B}(X)$ of $k_n[x_1, ..., x_d]$ and the canonical basis of k^{N+1}, so that it reflect faithfully the properties of ϕ. In particular, $X_0, ..., X_N$ is poised if and only if ϕ is bijective, so that it depend on the non-singularity of A. Based on this fact, the following theorem is trivial.

Theorem 4 (Criteria of poiseness). *Given* $N+1 = \begin{pmatrix} n+d \\ d \end{pmatrix}$ *distinct points* $X_0, ..., X_N$ *in* k^d. *The following are equivalent:*

(i) $X_0, ..., X_N$ *are poised.*

(ii) $X_0, ..., X_N$ *do not belong to a common hypersurface of degree less than or equal to* n.

(iii) For any given vector of scalar values $(u_0, ..., u_N) \in k^{N+1}$, *there exists a polynomial* P *of degree less than or equal to* n *that satisfies the interpolation conditions.*

Also based on the matrix representation of ϕ, one can check the poiseness of $X_0, ..., X_N$ by computing the determinant of A, hence taking $O(N^3)$ operations on k. Furthermore, once $X_0, ..., X_N$ are poised, the inverse image, or the interpolating polynomial, for the vector of scalar values $(u_0, ..., u_N) \in k^{N+1}$ is defined by

$$f(X) = (u_0, ..., u_N).A^{-1}.\mathbf{B}(X)$$

So that solving the interpolation problem in the poised case is equivalent to computing the inverse of a square matrix of size $N+1$, hence takes $O(N^3)$ operations on k using standard algorithm.

In the univariate case (then $n = N$), the canonical basis of $k_n[x]$ is $\{1, x, x^2, ..., x^n\}$. So that the matrix A is exactly the Vandemonde matrix of size $n+1$.

$$\begin{bmatrix} 1 & 1 & ... & 1 \\ X_0 & X_1 & ... & X_N \\ ... & ... & ... & ... \\ X_0^N & X_1^N & ... & X_N^N \end{bmatrix}$$

It is easy to show that $detA = \prod_{i \neq j} (X_i - X_j)$. Hence the poiseness in the univariate case is agreeing with the distinctness.

In the polynomial ring of several variables, the distinctness is strictly weaker than the poiseness as we seen in example 1. The method for solving the Multivariate Polynomial Interpolation in the general case is more complicate than the poiseness one. And we will back to this detail in the final chapter.

1.2.2 The Cartesian product case - Grids

When the set of interpolation points forms a lattice, i.e. a Cartesian product of finite sets on each coordinate, we can use the fast interpolation in the last section to find an solution. The aim of this subsection is to present such an algorithm.

Problem 3. *Given d finite subsets of k:*

$$J_1 = \{x_{11}, ..., x_{1n_1}\}$$

$$J_2 = \{x_{21}, ..., x_{2n_2}\}$$

$$.....$$

$$J_d = \{x_{d1}, ..., x_{dn_d}\}$$

where $x_{ij} \neq x_{ij'}$, $\forall i, \forall j \neq j'$, and $N = n_1 n_2 ... n_d$ scalar values

$$\{u_I \in k | I = (i_1, ..., i_d) \in J_1 \times ... \times J_d\}$$

Find a polynomial $P \in k[x_1, ..., x_d]$ satisfies the interpolation condition:

$$P(x_{1i_1}, ..., x_{di_d}) = u_{i_1 ... i_d}$$

for all $(i_1 ... i_d) \in J_1 \times ... \times J_d$.

Before coming to a fast algorithm which solves Problem 3, we discuss about the existence of solutions via the evaluation maps:

$$eva : k[x_1, ..., x_d] \to k^N, \quad f \mapsto (f(x_{1i_1}, ..., x_{di_d}))_{i_1, ..., i_d}$$

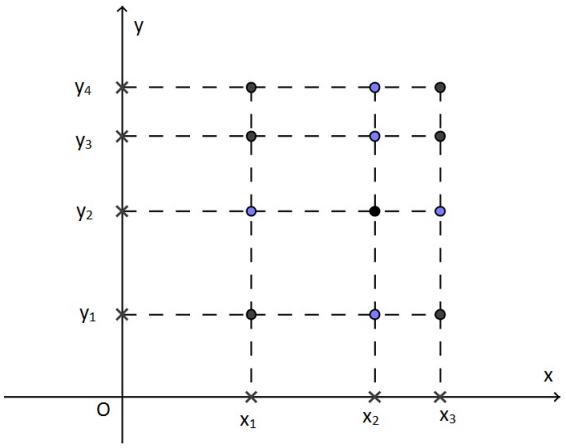


Figure 1.2: A Lattice in k^2.

$$eva_j : k[x_j] \to k^{n_j}, \;\; g \mapsto (g(x_{j1}), ..., g(x_{jn_j}))$$

Take tensor product over the field k of k-linear maps eva_j, $j = 1, ..., d$, we obtain the following commutative diagram of canonical k-linear maps:

$$\begin{array}{ccc} k[x_1] \underset{k}{\otimes} ... \underset{k}{\otimes} k[x_d] & \longrightarrow & k^{n_1} \underset{k}{\otimes} ... \underset{k}{\otimes} k^{n_d} \\ \downarrow & & \downarrow \\ k[x_1, ..., x_d] & \xrightarrow{eva} & k^N \end{array}$$

where the columns are natural isomorphism.

Since the hypothesis of Problem 3, for each $j = 1, ..., d$, the set $\{x_{j1}, ..., x_{jn_j}\} \subset k$ is poised. So the restriction of eva_j to $k_{n_j-1}[x_j]$ is bijective

$$eva_j|_{k_{n_j-1}[x_j]} : k_{n_j-1}[x_j] \to k^{n_j}$$

Get tensor product of all such k-linear maps, we obtained an isomorphism between k-vector spaces of dimension $N = n_1...n_d$:

$$k_{n_1-1}[x_1] \underset{k}{\otimes} ... \underset{k}{\otimes} k_{n_d-1}[x_d] \to k^{n_1} \underset{k}{\otimes} ... \underset{k}{\otimes} k^{n_d} \cong k^N$$

We can identify elements of $k_{n_1-1}[x_1] \underset{k}{\otimes} ... \underset{k}{\otimes} k_{n_d-1}[x_d]$ with their images via the composition of canonical k-linear maps:

$$k_{n_1-1}[x_1] \underset{k}{\otimes} ... \underset{k}{\otimes} k_{n_d-1}[x_d] \to k[x_1] \underset{k}{\otimes} ... \underset{k}{\otimes} k[x_d] \to k[x_1, ..., x_d]$$

where the first arrow is the natural embedding and the second arrow defined by $x^{i_1} \otimes ... \otimes x^{i_d} \mapsto x^{i_1}...x^{i_d}$, $\forall (i_1, ..., i_d) \in \mathbb{N}^d$. Putting together all of them, we obtain the following proposition:

Proposition 2. *Problem 3 always admits an unique solution in the set of polynomials whose the highest power of variable x_j is less than or equal to $n_j - 1$ for all $j = 1, ..., d$.*

The main idea to solve Problem 3 is viewing the ring of polynomials in d variables $k[x_1, ..., x_d]$ as a ring of polynomials in variable x_d over the ring $k[x_1, ..., x_{d-1}]$. Recall that the Fast Interpolation Algorithm runs on the polynomial ring over any unitary commutative ring. It only needs the interpolation points distinct and the different between them invertible.

Algorithm 5 Fast Multivariate Interpolation

Require: $J_i = \{x_{i1}, ..., x_{in_i}\}$ where $i = 1, ..., d; n_1, ..., n_d \in \mathbb{N}; x_{ij} \neq x_{ij'}$, $\forall i, \forall j \neq j'$, and $N = n_1 n_2 ... n_d$ scalar values $\{u_I \in k | I = (i_1, ..., i_d) \in J_1 \times ... \times J_d\}$.

Ensure: P.

1: $s := d$.
2: $g_{i_1...i_{d-1}l} := u_{i_1...i_{d-1}l}$, $l = 1, ..., n_d$.
3: **while** $2 \leq s \leq d$ **do**
4: **for all** $(i_1, ..., i_{s-1}) \in J_1 \times ... \times J_{s-1}$ **do**
5: Recall Algorithm 4 to find the interpolating polynomial $g_{i_1...i_{s-1}}$ with $R[x] = k[x_s]$ if $s = d$, or $(k[x_{s+1,...,x_d}])[x_s]$ if $s < d$; the set of interpolation points J_s; the set of scalar values $\{g_{i_1...i_{s-1}l} | l = 1, ..., n_s\}$
6: **end for**
7: $s := s - 1$
8: **end while**
9: Recall the Algorithm 4 to find the interpolating polynomial P in $R[x_1] = (k[x_2, ..., x_d])[x_1]$ with the set of interpolation points J_1 and the set of scalar values $\{g_1, ..., g_{n_1}\}$.
10: Return P.

Theorem 5. *The interpolating polynomial P in algorithm 5 has degree at most $(n_1 - 1)(n_2 - 1)...(n_d - 1)$. And Algorithm 5 runs in $O(M(N) \log N)$ operations in k, where $N = n_1 n_2 ... n_d$ the number of interpolation points.*

Proof: By induction on d.

In the case $d = 1$, algorithm 5 is only a Fast Interpolation, so taking $O(M(n_1) \log n_1)$ operations in k. And the interpolating polynomial is of degree at most $n_1 - 1$. We may inductively assume that the theorem is correct. Then the step 1 of algorithm 5 gives the interpolating polynomials $g_1, ..., g_{n_1}$ of degree at most $(n_2 - 1)...(n_d - 1)$ and takes $O(M(n_2...n_d) \log(n_2, ..., n_d))$ operations in k.

In line 9, we apply Algorithm 4 on $R[x_1] = (k[x_2, ..., x_d])[x_1]$, so taking $O(M(n_1) \log n_1)$ operations in $k[x_2, ..., x_d]$, and the interpolating polynomial is of degree at most $n_1 - 1$ on variable x_1 . All of polynomials occur in line 9 are of degree at most $(n_2 - 1)...(n_d - 1)$. Thus the final interpolating polynomial is of degree at most $(n_1 - 1)(n_2 - 1)...(n_d - 1)$. On the other hand, the cost of $O(M(n_1) \log n_1)$ operations in $k[x_2, ..., x_d]$ is equal to $M((n_2 - 1)...(n_d - 1))O(M(n_1) \log n_1) = O(M(n_1 n_2 ... n_d) \log n_1)$.

Since

$$M(n_2...n_d) \log(n_2, ..., n_d) + M(n_1 n_2 ... n_d) \log n_1 \leq M(N) \log N$$

the totally cost of algorithm 5 is $O(M(N)\log N)$. □

Chapter 2

Reduction modulo ideals in several variables

It is not difficult to see that the evaluation map

$$eva : k[x_1, ..., x_d] \to k^{(N+1)}, \;\; f \mapsto (f(X_0), ..., f(X_N))$$

is a surjective map. More precisely, in the next chapter, we will present an algorithm to find an interpolating polynomial f from any given $N+1$ scalar values. Unfortunately f is not maybe a good solution because the degree of f may be too large. We also known that, if f and f' are interpolating polynomials of points $X_0, ..., X_N$ with the same values at these points, then the difference between them vanishes at all X_i, $i = 0, ..., N$. It implies that f and f' are congruent modulo an ideal $\mathfrak{a}$ of $k[x_1, ..., x_d]$ that is defined by points $X_0, ..., X_N$. The purpose of this chapter is to present an analogue of division with remainder in case of several variables. Then we can use it to reduce some bad solution f module the ideal $\mathfrak{a}$ to obtain a better interpolating polynomial in some sense.

2.1 Monomial orders

The division with remainder in univariate polynomial ring is exactly Euclide algorithm. In $k[x]$, if we want to divide a polynomial f by an other non-zero polynomial g, we first divide all terms of f by the leading term of g. Thus in order to do an analogous algorithm in several variables, we must define the leading term of a polynomial. In this section, we will define orders on the set of all monomials, then deduce an order on terms also, of $k[x_1, ..., x_d]$ and end with some important properties of these orders.

Definition 2 (Monomial order)**.** *A monomial order on $k[x_1, ..., x_d]$ is an order "$<$" on the set of all monomials that satisfies the following conditions:*

(i) *"$<$" is a total order.*

(ii) *"$<$" is compatible with multiplication, i.e. if m, m, p are monomials then*

$$m < n \Rightarrow mp < np$$

(iii) *$1 < m$ for every monomial $m \neq 1$.*

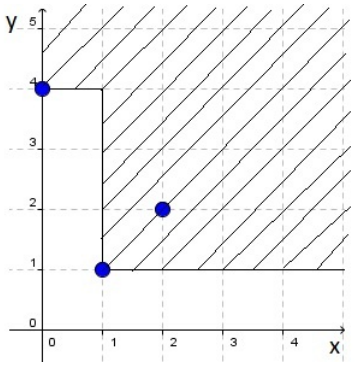


Figure 2.1: Semi-subgroup of $\mathbb{M}_d$ which is generated by $A = \{x^3, x^2y^2, xy\}$.

Remark: In the univariate case, there is only one way to order monomials in $k[x]$ which is ordering by their degree. In fact, the condition (iii) says that $1 < x^a,\ \forall a > 0$ and (ii) implies that $x^a < x^b,\ \forall a < b$. Thus we always have $1 < x < x^2 < x^3 < ...$

Denote $\mathbb{M}_d$ the set of all monomials in $k[x_1, ..., x_d]$. It forms a commutative unitary semigroup with basis $\{x_1, ..., x_d\}$. The structure of $\mathbb{M}_d$ can be described faithfully via the canonical isomorphism of semigroups:

$$\mathbb{M}_d \to \mathbb{N}^d,\ \ X^I = x_1^{i_1}...x_d^{i_d} \mapsto I = (i_1, ..., i_d)$$

For example, for any $I, J \in \mathbb{N}^d$, X^I divides X^J if and only if $J - I \in \mathbb{N}^d$, i.e. all coordinates of I are less than or equal to the corresponding coordinates of J. Thus all principal ideals of $\mathbb{M}_d$ have the form $I + \mathbb{N}^d$, where X^I is its generator. And then every ideal of $\mathbb{M}_d$ is an union of some such $I + \mathbb{N}^d$.

Example 2. In $k[x, y]$, let $A = \{x^3, x^2y^2, xy\}$. Then the semi-subgroup of $\mathbb{M}_2$ generated by A can be seen from Figure 2.1

Lemma 3 (Dickson)**.** *All ideals of $\mathbb{M}_d$ are finitely generated. Furthermore, for every set $A \subseteq \mathbb{M}_d$, there exists a finite subset $B \subseteq A$ such that B generates $\langle A \rangle$, the ideal of $\mathbb{M}_d$ which is generated by A.*

Proof: In $\mathbb{M}_d$, we consider a relation "$<$" defined by:

$$m < m' \Leftrightarrow m \text{ divides } m',\ \ \forall m, m' \in \mathbb{M}_d$$

For any given element of $m \in \mathbb{M}_d$, there are only finitely many elements of $\mathbb{M}_d$ that divide m. Hence, the relation "$<$" satisfies the descending chain condition. Thanks to Zorn's lemma, A admits some minimal elements with respect to "$<$". Let B be the set of all such minimal elements of A. Then B generates the ideal $\mathfrak{a} = \langle A \rangle$. It suffices to show that B is a finite set.

We will prove that B is finite by induction on d. If $d = 1$, then $\mathbb{M} = \{1, x_1, x_1^2, ..., x_1^n, ...\}$ is totally ordered by "$<$". So B has only the smallest element of A. If $d \geq 2$, consider the projection between semigroups:

$$\pi : \mathbb{M}_d \to \mathbb{M}_{d-1},\quad x_1^{i_1}...x_d^{i_d} \mapsto x_1^{i_1}...x_{d-1}^{i_{d-1}}$$

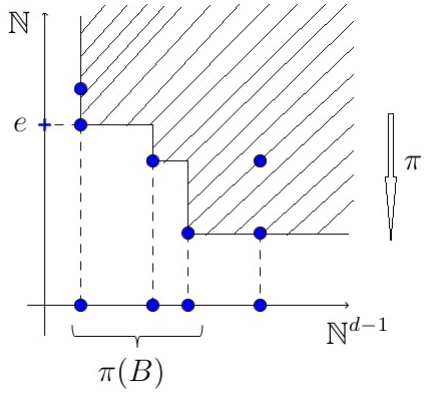


Figure 2.2: B is the set of all minimals elements of A, and $\pi(B)$ image of B via projection π

where $\mathbb{M}_{d-1}$ is the set of all monomials of $k[x_1, ..., x_{d-1}]$. Clearly $\pi(A)$ is a semi-subgroup of $\mathbb{M}_{d-1}$, and the ideal $\langle \pi(A) \rangle \subseteq \mathbb{M}_{d-1}$ is generated by the set $\pi(B) \subseteq \mathbb{M}_{d-1}$. By induction hypothesis, the set C of all minimal elements of $\pi(B)$ with respect to the order "$<$" generates $\langle \pi(A) \rangle$ and finite, namely

$$C = \{m_1, ..., m_r\} \subseteq \pi(B) \subseteq \mathbb{M}_{d-1}, \;\; r > 0$$

For each $i = 1, ..., r$, since $m_i \in C \subseteq \pi(B)$, there exists $d_i > 0$ such that

$$m_i x_d^{d_i} \in B$$

We choose such d_i smallest possible. And let

$$e = \max\{d_1, ..., d_r\}$$

The key of this proof is the following claim:

Claim: For any $x_1^{i_1}...x_d^{i_d} \in B$, we have $i_d \leq e$.

In fact, if $x_1^{i_1}...x_d^{i_d} \in B$ then $x_1^{i_1}...x_{d-1}^{i_{d-1}} \in \pi(B)$. So there exists $m_i \in C$ such that $m_i \leq x_1^{i_1}...x_{d-1}^{i_{d-1}}$. From here, if $i_d > e$ then

$$m_i x_d^{d_i} \leq m_i x_d^e < m_i x_d^{i_d}$$

It is impossible because $m_i x_d^{d_i} \in B$ and $m_i x_d^{i_d} = x_1^{i_1}...x_d^{i_d}$ is minimal in B. The claim is proved.

Similarly, we also find that all other powers of variables x_i on B are bounded. Hence, B is finite. □

Thanks to lemma 3, all monomial orders in $k[x_1, ..., x_d]$ must be well-order.

Theorem 6. *Every monomial order in $k[x_1, ..., x_d]$ is well-order, i.e. every non-empty subset of $\mathbb{M}$ admits the smallest element.*

Proof: Let A be a non-empty set of some monomials in $k[x_1, ..., x_d]$. With a given monomial order "$<$" in $k[x_1, ..., x_d]$, we need to find the smallest element of A. Because of Dickson's lemma, A contains a finite subset $B \subseteq A$ such that every element of A is divisible by some elements of B. The order "$<$" is a total order. So B admits the smallest element. And then it is also the smallest element of A. □

Example 3. Here are some important examples of monomial orders in $k[x_1, ..., x_d]$. In the following example, we denote $a = (a_1, ..., a_d)$; $b = (b_1, ..., b_d) \in \mathbb{N}^d$.

1. **Lexicography order**:

   $$X^a <_{lex} X^b \Leftrightarrow \text{the leftmost non-zero entry in } a - b \text{ is negative.}$$

   For example, in $k[x, y, z]$, we have $y^4 <_{lex} xyz^3 <_{lex} x^2$.

2. **Graded lexicographic order**:

   $$X^a <_{grlex} X^b \Leftrightarrow |a| < |b| \text{ or } (|a| = |b| \text{ and } X^a <_{lex} X^b).$$

   For example, $x^2 <_{grlex} y^4 <_{grlex} xyz^2$.

3. **Graded reverse lexicographic order**:

   $X^a <_{grevlex} X^b \Leftrightarrow |a| < |b|$ or ($|a| = |b|$ and the rightmost non-zero entry of $a - b$ is positive).

   For example, $x^2 <_{grevlex} xyz^2 <_{grevlex} y^4$.

Definition 3. *Let $f = \sum_{I \in \mathbb{N}^d} c_I X^I$ be a non-zero polynomial in $k[x_1, ..., x_d]$, where the sum is finite, and "$<$" a monomial order.*

(i) Each $c_I X^I$ with $c_I \neq 0$ is called a term of f. We denote its order is the order of X^I.

(ii) The leading term of f, denoted by $lt_<(f)$ (or $lt(f)$ if the monomial order is defined), is the maximal term of f with respect to $<$.

(iii) The leading monomial of f, denoted by $lm(f)$, is the monomial which respect to the leading term of f.

(iv) The leading coefficient of f, denoted by $lc(f)$, is the coefficient of leading term of f.

(v) For an ideal $\mathfrak{a}$, we denote $lt(\mathfrak{a})$ the set of $lt(f)$ for all $f \in \mathfrak{a}$.

Example 4. Let $f(x, y, z) = y^4 + 2xyz^2 - 3x^2 \in k[x, y, z]$, then

	$<_{lex}$	$<_{grlex}$	$<_{grevlex}$
$lt(f)$	$-3x^2$	$2xyz^2$	y^4
$lm(f)$	x^2	xyz^2	y^4
$lc(f)$	-3	2	1

The following are characteristic properties of monomial orders in example 3.

Proposition 3. *Let $f \in k[x_1, ..., x_d]$.*

- *If $lt_{<_{lex}}(f) \in k[x_s, ..., x_d]$ for some $s = 1, ..., d$ then $f \in k[x_s, ..., x_d]$.*
- *If f is homogeneous and $lt_{<_{grlex}}(f) \in k[x_s, ..., x_d]$ for some $s = 1, ..., d$ then $f \in k[x_s, ..., x_d]$.*
- *If f is homogeneous and $lt_{<_{grevlex}}(f) \in \langle x_s, ..., x_d \rangle$ for some $s = 1, ..., d$, where $\langle x_s, ..., x_d \rangle$ is the ideal of $k[x_1, ..., x_d]$ which is generated by $x_s, ..., x_d$, then $f \in \langle x_s, ..., x_d \rangle$.*

2.2 Multivariate Division Algorithm

The following is the main theorem of this chapter.

Theorem 7. *In $k[x_1, ..., x_d]$ with a given monomial order "$<$", let $f_1, ..., f_s$, $(s \leq 1)$ be non-zero polynomials. Then for any polynomial f, there exists an expression*

$$f = q_1 f_1 + ... + q_s f_s + r$$

where $q_1, ..., q_s, r \in k[x_1, ..., x_d]$ such that:

(i) $lt(q_i f_i) \leq lt(f), \ \forall i = 1, ..., s.$

(ii) None of the terms in r is divisible by any $lt(f_i)$ for all $i = 1, ..., s.$

The proof of theorem 7 consists of an algorithm for finding such an expression of f.

Algorithm 6 Multivariate Division with remainder

Require: Non-zero polynomials $f, f_1, ..., f_s \in k[x_1, ..., x_d]$ and monomial order "$<$".
Ensure: The $q_1, ..., q_s, r \in k[x_1, ..., x_d]$ as in theorem 7.

```
r := 0, q_1 := 0, ..., q_s := 0, p := f
while p ≠ 0 do
    if lt(f_i) divides lt(p) for some i = 1, ..., s then
        then choose the smallest such i.
        q_i := q_i + lt(p)/lt(f_i)
        p := p − (lt(p)/lt(f_i)).f_i
    else
        r := r + lt(p)
        p := p − lt(p)
    end if
end while
Return q_1, ..., q_s, r.
```

We denote the remainder r obtained from this algorithm by

$$r := f \text{ rem } \{f_1, ..., f_s\}$$

Lemma 4. *Each time the algorithm 6 passes through inside the* ***while*** *round from line 2 to line 11, we always have*

$$f = p + q_1 f_1 + ... + q_s f_s + r \tag{2.1}$$

where $lt(q_i f_i) \leq lt(f),\ \forall i = 1, ..., s$ *and no term of* r *divisible by* $lt(f_i)$ *for all* i.

Proof: In the first loop, the equation 2.1 is expressed by

$$f = f + 0.f_1 + ... + 0.f_s + 0$$

hence holds the lemma. Assume at an other loop, we have $p, q_1, ..., q_s, r$ satisfy the lemma and $p \neq 0$. We will see what happens in the next loop.

If $lt(f_j)$ divides $lt(p)$ for some $j = 1, ..., s$ and i is the such smallest index. Then we get:

$$q_i := q_i + \frac{lt(p)}{lt(f_i)}$$

$$p := p - \frac{lt(p)}{lt(f_i)}.f_i$$

So

$$f = p + q_1 f_1 + ... + q_s f_s + r =$$
$$= \left(p - \frac{lt(p)}{lt(f_i)}.f_i\right) + q_1 f_1 + ... + \left(q_i + \frac{lt(p)}{lt(f_i)}\right) f_i + ... + q_s f_s + r$$

So the equation 2.1 also holds in the next loop. On the other hand, since $lt(q_i f_i) \leq lt(f)$ and $lt\left(\frac{lt(p)}{lt(f_i)} f_i\right) = lt(p) \leq lt(f)$, we imply that

$$lt\left(q_i + \frac{lt(p)}{lt(f_i)}\right) \leq lt(f)$$

Hence, the lemma also holds in the next loop. Equivalently for the case there is no $lt(f_i)$ that divides $lt(f)$. □

Theorem 8. *Algorithm 6 finds correctly an expression of* f *as in theorem 7 in* $O\left(n^d m^d\right)$ *operations in* k*, where* n *is the total degree of* f *and* m *the maximal total degree of* $f_1, ..., f_s$.

Proof: Each time the algorithm 6 passes through inside the **while** round from line 2 to line 11, we alway have either

$$p := p - \frac{lt(p)}{lt(f_i)}.f_i$$

or

$$p := p - lt(p)$$

Then the leading term of p strictly decreases. So the number of loops of the **while** round is bounded by the maximum number of terms of a polynomial of total degree n in $k[x_1, ..., x_d]$, namely $O\left(\begin{pmatrix} n+d \\ d \end{pmatrix}\right) = O(n^d)$ loops.

The algorithm stops as soon as $p = 0$. The final equation that we obtained is

$$f = q_1 f_1 + ... + q_s f_s + r$$

And it proves theorem 7 using lemma 4.

On the other hand, on each loop, the cost is bounded by the cost to compute $p = p - \frac{lt(p)}{lt(f_i)} f_i$ for some i, hence taking $O\left(\left(\begin{array}{c} m+d \\ d \end{array}\right)\right) = O(m^d)$ operations in k. Finally the arithmetic cost of algorithm 6 is $O\left(n^d m^d\right)$. □

Example 5. In $k[x, y]$ with the lexicography order, we use Multivariate Division Algorithmto divide $f = x^3 + x^2y + xy^2 + y^3$ by $f_1 = xy + 1, f_2 = x + 1$. The following table is its trace:

Step	$f = x^3 + x^2y + xy^2 + y^3$	$f_1 = xy + 1$	$f_2 = x + 1$	Remainder
0	$p = x^3 + x^2y + xy^2 + y^3$	$q_1 = 0$	$q_2 = 0$	$r = 0$
1	$x^2y - x^2 + xy^2 + y^3$	0	x^2	0
2	$-x^2 + xy^2 - x + y^3$	x	0	0
3	$xy^2 + y^3$	0	$-x$	0
4	$y^3 - y$	y	0	0
5	$-y$	0	0	y^3
6	0	0	0	$-y$
Final		$q_1 = x + y$	$q_2 = x^2 - x$	$r = y^3 - y$

Hence, $f = (x + y)(xy + 1) + (x^2 - x)(x + 1) + (y^3 - y)$.

Example 6. With the same polynomials as in the above example, but now we change the order of f_1 and f_2. The trace of algorithm is the following table:

Step	$f = x^3 + x^2y + xy^2 + y^3$	$f_1 = x + 1$	$f_2 = xy + 1$	Remainder
0	$p = x^3 + x^2y + xy^2 + y^3$	$q_1 = 0$	$q_2 = 0$	$r = 0$
1	$x^2y - x^2 + xy^2 + y^3$	x^2	0	0
2	$-x^2 + xy^2 - xy + y^3$	xy	0	0
3	$xy^2 - xy + x + y^3$	$-x$	0	0
4	$-xy + x + y^3 - y^2$	y^2	0	0
5	$x + y^3 - y^2 + y$	$-y$	0	0
6	$y^3 - y^2 + y - 1$	1	0	0
7	$-y^2 + y - 1$	0	0	y^3
8	$y - 1$	0	0	$-y^2$
9	-1	0	0	y
10	0	0	0	-1
Final		$q_1 = x^2 + xy - x + y^2 - y + 1$	$q_2 = 0$	$r = y^3 - y^2 + y - 1$

Hence, $f = (x^2 + xy - x + y^2 - y + 1)(x + 1) + 0.(xy + 1) + (y^3 - y^2 + y - 1)$. So the remainders which are obtained from the Multivariate Division Algorithmdepend on the way to index the elements of the set $F = \{f_1, ..., f_r\}$.

Example 7. In $k[x, y]$ with the lexicography order, we divide $f = x^2y+y-2$ by $f_1 = xy+1$ and $f_2 = x + 1$.

Step	$f = x^2y + y - 2$	$f_1 = xy + 1$	$f_2 = x + 1$	Remainder
0	$p = x^2y + y - 2$	$q_1 = 0$	$q_2 = 0$	$r = 0$
1	$-x + y - 2$	x	0	0
2	$y - 1$	0	-1	0
3	-1	0	0	y
4	0	0	0	-1
Final		$q_1 = x$	$q_2 = -1$	$r = y - 1$

Hence, $x^2y + y - 2 = x(xy + 1) - (x + 1) + (y - 1)$.
However,

$$x^2y + y - 2 = y(x + 1)^2 - 2(xy + 1) \in \langle x + 1, xy + 1\rangle$$

where $\langle x + 1, xy + 1\rangle$ is an ideal of $k[x, y]$ which is generated by $x + 1$ and $xy + 1$. Thus in this case, the Multivariate Division Algorithmdoes not help us to check that $f = x^2y+y-2$ is in $\langle x + 1, xy + 1\rangle$ or not!!!

2.3 Gröbner bases

We start with a following problem:

Problem 4 (Ideal Membership Problem)**.** *Let $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ be an ideal which is generated by a given set of polynomials $F = \{f_1, ..., f_s\}$. Given any $f \in k[x_1, ..., x_d]$. Is it $f \in \mathfrak{a}$?*

Everything is easy in the case $d = 1$. Since $k[x]$ is an Euclidean domain, all ideals of $k[x]$ are principal, and

$$\langle f_1, ..., f_s\rangle = \langle gcd(f_1, ..., f_s)\rangle$$

So we can use Euclid's algorithm to divide f by $g = gcd(f_1, ..., f_s)$ with remainder r whose degree less than deg$g$. Then

$$f \in \langle f_1, ..., f_s\rangle \text{ if and only if } r = 0 \tag{2.2}$$

Unfortunately, (2.2) is no longer true in the several variables case, as we have seen in the example 6. Thus if we want to use the Multivariate Division Algorithmto solve the Ideal Membership Problem, we must change the monomial order in $k[x_1, ..., x_d]$ or the set of generators of $\mathfrak{a}$.

In fact, from any given monomial order of $k[x_1, ..., x_d]$ and given set of generators of $\mathfrak{a}$, we can find an other special set of generators such that the Ideal Membership Problem can be done by using Multivariate Division Algorithm. Such set of generators is called Gröbner bases. In this section, we will define Gröbner bases and describe Buchberger's algorithm to find a Gröbner basis from any given set of generators.

2.3.1 Gröbner bases

Throughout this section, we assume some monomial order "$<$" on $k[x_1, ..., x_d]$. To define Gröbner bases, we first give the definition below:

Definition 4 (Monomial ideals)**.** *A monomial ideal $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ is an ideal generated by monomials in $k[x_1, ..., x_d]$.*

From now, for a set of polynomials A, we denote $\langle A \rangle$ an ideal of $k[x_1, ..., x_d]$ which is generated by A.

Assume $A \subseteq \mathbb{M}_d$ be a set of monomials, and $\mathfrak{a} = \langle A \rangle$ a monomial ideal of $k[x_1, ..., x_d]$. Then the set $\mathfrak{a} \cap \mathbb{M}_d$ of all monomials in $\mathfrak{a}$ forms an ideal of $\mathbb{M}_d$ which is generated by A. This fact can be described as the following bijective:

$$\{\text{ideals of } \mathbb{M}_d\} \xrightarrow{\phi} \{\text{monomial ideals of } k[x_1, ..., x_d]\}$$

where $\phi(A) = \langle A \rangle$ for $A \subseteq \mathbb{M}_d$ an ideal, and $\phi^{-1}(\mathfrak{a}) = \mathfrak{a} \cap M_d$ for $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ a monomial ideal. Thus the properties of ideals of $\mathbb{M}_d$ can translate faithfully to the properties of monomial ideals of $k[x_1, ..., x_d]$. Here we need the finiteness of monomial ideals of $k[x_1, ..., x_d]$, that is translated directly from lemma 3.

Lemma 5 (Dickson)**.** *All monomial ideals of $k[x_1, ..., x_d]$ are finite generated. Furthermore, for every set $A \subseteq \mathbb{M}_d$ of monomials, the exists a finite subset $B \subseteq A$ such that B generates the monomial ideal $\langle A \rangle$.*

For each ideal $\mathfrak{a} \subseteq k[x_1, ..., x_d]$, the ideal $\langle lt(\mathfrak{a}) \rangle \subseteq k[x_1, ..., x_d]$ which is generated by all leading terms of all elements of $\mathfrak{a}$ is a monomial ideal. It contains all information about the leading terms of elements of $\mathfrak{a}$, so that has strong relation with $\mathfrak{a}$. Dickson's lemma says that it is always generated by only finitely many monomials, say $m_1, ..., m_s$. Conversely, any subset of $\mathfrak{a}$ whose the set of leading terms is large enough, i.e. any m_i can be represent by leading terms of some its elements, can generates the whole $\mathfrak{a}$. This fact is the following lemma:

Lemma 6. *Let $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ be an ideal. If $G \subseteq \mathfrak{a}$ is a finite subset such that $\langle lt(G) \rangle = \langle lt(\mathfrak{a}) \rangle$, then $\langle G \rangle = \mathfrak{a}$.*

Proof: Let $G = \{g_1, ..., g_s\}$ as assumption. Let any $f \in \mathfrak{a}$. Using the Multivariate Division Algorithmto divide f by $G = \{g_1, ..., g_s\}$ gives us an expression:

$$f = q_1 g_1 + ... + q_s g_2 + r$$

where $q_1, ..., q_s, r \in k[x_1, ..., x_d]$ such that either $r = 0$ or no term of r is divisible by leading term of any g_i. But we also have $r = f - q_1 g_1 - ... - q_s g_s \in \mathfrak{a}$, so $lt(r) \in \langle lt(\mathfrak{a}) \rangle = \langle lt(G) \rangle$. Thus $r = 0$. And then $f = q_1 g_1 + ... + q_s g_s \in \langle G \rangle$. Hence $\langle G \rangle = \mathfrak{a}$. □

Together two above lemmas, we obtain the following theorem which is the motivation of Gröbner bases.

Theorem 9 (Hilbert's basis theorem)**.** *Every ideal $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ is finitely generated. More precisely, $\mathfrak{a}$ can be generated by a finite set of polynomials $G \subset \mathfrak{a}$ that satisfies*

$$\langle lt(G) \rangle = \langle lt(\mathfrak{a}) \rangle$$

Proof: Because of Dickson's lemma, there exist a finite subset of $lt(\mathfrak{a})$, namely $\{m_1, ..., m_s\}$, that generates $\langle lt(\mathfrak{a}) \rangle$, where m_i is the leading term of some polynomial $g_i \in \mathfrak{a},\ i = 1, ..., s$. Let $G = \{g_1, ..., g_s\}$, then $\langle G \rangle \subseteq \mathfrak{a}$ and $\langle lt(G) \rangle = \langle lt(\mathfrak{a}) \rangle$. Thanks to the lemma 5, we have $\langle G \rangle = \mathfrak{a}$. □

We have seen that the remainder from dividing a polynomial $f \in k[x_1, ..., x_d]$ by a finite set of generators F of an ideal $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ depends on the order of F. The main reason for this fact is the set $lt(F)$ of leading terms of all polynomials in F probably not enough. The Hilbert's basis theorem gives the existence of a set of generators whose set of leading terms is enough to represent all leading terms of polynomial in $\mathfrak{a}$. For this reason, we will see in this subsection, that the remainder of dividing f by such set of generators is unique in some sense.

Definition 5. *Given an ideal $\mathfrak{a} \subseteq k[x_1, ..., x_d]$. A finite subset $G \subseteq \mathfrak{a}$ is called a Grobner basis for $\mathfrak{a}$ with respect to "<" if $\langle lt(G) \rangle = \langle lt(I) \rangle$.*

Because of Hilbert's basis theorem, a Gröbner basis for any given ideal $\mathfrak{a}$ always exists, and is in fact a set of generators of $\mathfrak{a}$ in the ring theoretical sense.

Proposition 4. *Let G be a Gröbner basis for ideal $\mathfrak{a}$, and $f \in k[x_1, ..., x_d]$. Then there exists a unique polynomial $r \in k[x_1, ..., x_d]$ such that $f \equiv r \mod \mathfrak{a}$ and no term of r is divisible by any term of $lt(G)$.*

Proof: The existence of r follows from the Multivariate Division Algorithm. Now, we suppose that $f \equiv r_1 \equiv r_2 \mod \mathfrak{a}$ with $r_1, r_2 \in k[x_1, ..., x_d]$ and no term of r_1, r_2 is divisible by any term of $lt(G)$. Then $r_1 - r_2 \in \mathfrak{a}$. It implies that $lt(r_1 - r_2) \in \langle lt(\mathfrak{a}) \rangle = \langle lt(G) \rangle$. But there is no term of $r_1 - r_2$ divisible by any term of $lt(G)$, hence $r_1 - r_2 = 0$. □

An immediately consequence of this proposition is the solvability of Ideal Membership Problem.

Theorem 10. *Let G be a Gröbner basis for ideal $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ with respect to a monomial order "<". Then for any $f \in k[x_1, ..., x_d]$*

$$f \in \mathfrak{a} \text{ if and only if } (f \text{ rem } G) = 0$$

.

2.3.2 Buchberger's Algorithm

The aim of this subsection is to describe an algorithm of Bruno Buchberger to construct a Gröbner basis for any given idea $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ with given set of generators $F = \{f_1, ..., f_s\}$. The main ideal of Buchberger's algorithm is to add some more polynomials into the set F for

which the corresponding monomial ideal $\langle lt(\mathfrak{a})\rangle$ is bigger. The polynomials which we want to add to F come from the cancellation the leading terms of pairs of polynomials in F.

Now for each pair $g, h \in k[x_1, ..., x_d]$ of non-zero polynomials, we define a new polynomial which we call S-polynomial, by:

$$S(g,h) = \frac{m}{lt(g)}.g - \frac{m}{lt(g)}.h$$

where $m = lcd\,(lt(g), lt(h))$ is a monomial.

Since $\frac{m}{lt(g)}, \frac{m}{lt(h)} \in k[x_1, ..., x_d]$, $S(g,h) \in \langle g, h\rangle$. And since $\frac{m}{lt(g)}.g, \frac{m}{lt(h)}.h$ have the same leading terms, it is killed in $S(g,h)$. So we can add some more reduction modulo $\mathfrak{a}$ of such S-polynomials into the set of generators of $\mathfrak{a}$ to obtain a new set of generators whose the leading terms generates probably a bigger monomial ideal. As soon as the set of generators of $\mathfrak{a}$ is "large enough", then all of such reductions will vanishes.

Theorem 11 (Buchberger's criteria)**.** *A set of generators $G = \{g_1, ..., g_s\}$ of an ideal $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ is a Gröbner basis if and only if*

$$S(g_i, g_j) \text{ rem } G = 0, \ \forall i \neq j$$

Proof: If $G = \{g_1, ..., g_s\}$ is a Gröbner basis for $\mathfrak{a}$, then $S(g_i, g_j)$ rem $G = 0$ for all $i \neq j$ because of Ideal Membership Problem. The conversely way is more complicate.

Now assume $G = \{g_1, ..., g_s\}$ is a set of generators of $\mathfrak{a}$ such that $S(g_i, g_j)$ rem $G = 0$ for all $i \neq j$. To become a Gröbner basis, the leading terms of elements of G must generate the monomial ideal $\langle lt(\mathfrak{a})\rangle$. We will prove that for any $f \in \mathfrak{a}$, $lt(f)$ is divisible by some $lt(g_i) \in lt(G)$.

Now let any $f \in \mathfrak{a}$. f can be represented via the set of generators G by

$$f = h_1 g_1 + ... + h_s g_s \tag{2.3}$$

where $h_1, ..., h_s \in k[x_1, ..., x_d]$. In all such expressions, we choose a representation such that the greatest leading monomial of terms in the right hand side

$$m := \max\{lm(h_i g_i), \ i = 1, ..., s\}$$

smallest possible, recalled by (2.3). Our aim is to prove that $m = lt(f)$, hence the theorem is concluded.

By contradiction, assume $m > lt(f)$. The following is to construct an other expression of f via the set of generators G such that the greatest leading monomial of terms in the right hand side is less than m, so getting contradiction.

By renumbering indexes, we can assume more

$$m = lm(h_1 g_1) = ... = lm(h_r g_r) > lm(h_{r+1} g_{r+1}) \geq ... \geq lm(h_s g_s), \ \ r \in \{1, ..., s\}$$

Now we rewrite (2.3) as follow:

$$f = \sum_{i=1}^{r} lt\left(h_i\right) g_i + \sum_{i=1}^{r} \left(h_i - lt\left(h_i\right)\right) g_i + \sum_{i=r+1}^{s} h_i g_i \tag{2.4}$$

The second and the third sum in the right hand side of (2.4) are linear combinations of g_i such that the leading terms of their terms are less than m. Now let us consider the first sum.

$$S = lt(h_1)g_1 + ... + lt(h_r)g_r \tag{2.5}$$

All leading monomials of terms of S are the same and equal to m. The multiplication by non-zero constants in k to g_i does not change the result, so we can assume that all leading coefficients of g_i are equal to 1. Rewrite $lt(h_1) = c_i X^{\alpha_i},\ c_i \in k, \alpha_i \in \mathbb{N}^d$, we represent (2.5) by:

$$\begin{aligned} &S = c_1 X^{\alpha_1} g_1 + ... + c_r X^{\alpha_r} g_r \\ &= c_1 \left(X^{\alpha_1} g_1 - X^{\alpha_2} g_2\right) + \left(c_1 + c_2\right) \left(X^{\alpha_2} g_2 - X^{\alpha_3} g_3\right) + ... + \left(c_1 + ... + c_{r-1}\right) \left(X^{\alpha_{r-1}} g_{r-1} - X^{\alpha_r} g_r\right) + \\ &\qquad + \left(c_1 + ... + c_r\right) X^{\alpha_r} g_r \end{aligned}$$

Notice that $(c_1 + ... + c_r)$ is the coefficient of m in the standard expression of $f = \sum_{i=1}^{s} h_i g_i$. But, as our assumption, $m > lt(f)$, then $c_1 + ... + c_r = 0$. We conclude:

$$S = d_1 \left(X^{\alpha_1} g_1 - X^{\alpha_2} g_2\right) + ... + d_{r-1} \left(X^{\alpha_{r-1}} g_{r-1} - X^{\alpha_r} g_r\right) \tag{2.6}$$

where $d_1, ..., d_{r-1} \in k$.

Now look at any term of the right hand side of (2.6), for example the first term:

$$S_1 = d_1 \left(X^{\alpha_1} g_1 - X^{\alpha_2} g_2\right)$$

Recall that the leading coefficient of g_i is equal to 1, and $X^{\alpha_1} lt(g_1) = X^{\alpha_2} lt(g_2) = m$. We rewrite S_1 as follow:

$$\begin{aligned} S_1 &= d_1 . \left[\frac{X^{\alpha_1} lt(g_1)}{m_{12}} \left(\frac{m_{12}}{lt(g_1)} g_1\right) - \frac{X^{\alpha_2} lt(g_2)}{m_{12}} \left(\frac{m_{12}}{lt(g_2)} g_2\right)\right] \\ &= \frac{d_1 m}{m_{12}} \left[\frac{m_{12}}{lt(g_1)} g_1 - \frac{m_{12}}{lt(g_2)} g_2\right] \\ &= d_1 X^{\alpha_{12}} S(g_1, g_2) \end{aligned}$$

where $m_{12} = lcd\left(lt(g_1), lt(g_2)\right)$, $\alpha_{12} \in \mathbb{N}^d$, and the final term satisfies:

$$lt\left(X^{\alpha_{12}} S(g_1, g_2)\right) = lt\left(X^{\alpha_1} g_1 - X^{\alpha_2} g_2\right) < m \tag{2.7}$$

Using Multivariate Division Algorithmto divide $S(g_1, g_2)$ by G, we get an expression of $S(g_1, g_2)$ as:

$$S(g_1, g_2) = f_{121} g_1 + ... + f_{12s} g_s \tag{2.8}$$

where the remainder vanishes because of the hypothesis. Furthermore,

$$lt(f_{12j}g_j) \leq lt(S(g_1, g_2)) \tag{2.9}$$

Combination (2.6) and (2.8), we obtain an new expression of S as:

$$S = \sum_{i=1}^{r-1} d_i X^{\alpha_{i(i+1)}} \sum_{j=1}^{s} f_{i(i+1)j} g_j = \sum_{j=1}^{s} \left(\sum_{i=1}^{r-1} d_i X^{\alpha_{i(i+1)}} \right) f_{i(i+1)j} g_j$$

or shorter

$$S = \sum_{j=1}^{s} f_j g_j \tag{2.10}$$

where $f_j = \left(\sum_{i=1}^{r-1} d_i X^{\alpha_{i(i+1)}} \right) f_{i(i+1)j},\ j = 1, ..., s$. Now look at the leading terms of terms in the right hand side of (2.10), we have

$$\begin{aligned} lt(f_j g_j) &\leq \max_{i=1,...,r} lt\left(X^{\alpha_{i(i+1)}} f_{i(i+1)j} g_j\right) \\ &= \max_{i=1,...,r} X^{\alpha_{i(i+1)}} lt\left(f_{i(i+1)j} g_j\right) \\ &\leq \max_{i=1,...,r} X^{\alpha_{i(i+1)}} lt\left(S(g_i, g_j)\right) \qquad \text{(since (2.9))} \\ &< m \qquad \text{(since (2.7))} \end{aligned}$$

Finally, putting together (2.4) with (2.10) we obtain

$$f = \sum_{j=1}^{s} f_j g_j + \sum_{i=1}^{r} \left(h_i - lt(h_i)\right) g_i + \sum_{j=r+1}^{s} h_j g_j$$

an expression of f via the set of generators G such that the leading terms of terms in the right hand side are less than m. It is contradiction with the minimality of m. The proof is completed. □

We now present a simplified version of Buchberger's algorithm (1965).

Theorem 12. *Buchberger's algorithm always terminates.*

Proof: Whenever Buchberger's algorithm stops, we obtain a set of generators of $\mathfrak{a}$ which holds Buchberger's criteria. Hence, it is a Gröbner basis.

Now assume G and G' are set of polynomials correspond to successive passes through step 2 each time, i.e. $G' = G \cup S$, where

$$S = \{S(g_i, g_j) \text{ rem } G \neq 0 \,|\, g_i, g_j \in G,\ i \neq j\}$$

We have known that $S(g_i, g_j) \in \langle g_i, g_j \rangle \subseteq \langle G \rangle = \mathfrak{a}$. Thus $\langle G' \rangle = \langle G \rangle = \mathfrak{a}$. It means that the set G of polynomials passes throughout the **for** round and **if** round from line 3 to line 15 always generates the ideal $\mathfrak{a}$. We now consider the corresponding monomial ideals.

Algorithm 7 Buchberger's algorithm

Require: $g_1, ..., g_s \in k[x_1, ..., x_d]$ and a monomial order "$<$".
Ensure: A Gröbner basis G for the ideal $\mathfrak{a} = \langle g_1, ..., g_s \rangle$ with respect to "$<$".

```
1: G := {g_1, ..., g_s}
2: S := ∅
3: for all g_i, g_j ∈ G, g_i ≠ g_j do
4:     r := S(g_i, g_j) rem G
5:     if r ≠ 0 then
6:         S := S ∪ {r}
7:     end if
8: end for
9: if S = ∅ then
10:    Return G. Stop the algorithm.
11: else
12:    G := G ∪ S
13:    S = ∅
14:    Repeat for round from line 3.
15: end if
```

Clearly $G \subseteq G'$, then $\langle lt(G) \rangle \subseteq \langle lt(G') \rangle$. Hence the ideals $\langle lt(G) \rangle$ in successive passes throughout the **for** round and **if** round form a ascending chain of monomial ideals. Thus after a finite number of steps, we have

$$\langle lt(G) \rangle = \langle lt(G') \rangle \subseteq \langle lt(\mathfrak{a}) \rangle$$

We claim that the algorithm will stop in this step, i.e. $S = \emptyset$. In fact in this step, for any pair $g_i, g_j \in G$, let $r = S(g_i, g_j)$ rem G. Then r must be zero. Otherwise, by definition of remainder on Multivariate Division Algorithmnone term of r is divisible by any term of $lt(G)$. On the other hand, $r \in G'$ then

$$lt(r) \in lt(G') \subseteq \langle lt(G') \rangle = \langle lt(G) \rangle$$

Contradiction!!! Thus $r = 0$, and then $S = \emptyset$. Hence the algorithm terminates. □

2.3.3 Reduced bases and computer calculation

In general the Gröbner basis computed by Buchberger's algorithm is neither minimal nor unique. By removing some unnecessary its elements, we can make both properties hold. The following lemma is the base for this fact.

Lemma 7. *If G is a Gröbner basis for an ideal $\mathfrak{a} \subseteq k[x_1, ..., x_d]$, and $g \in G$ such that $lt(g) \in \langle lt\,(G \backslash \{g\}) \rangle$, then $G \backslash \{g\}$ is still a Gröbner basis for $\mathfrak{a}$.*

Proof: Denote $G' = G \backslash \{g\}$. From the hypothesis, we have $\langle G' \rangle \subseteq \langle G \rangle = \mathfrak{a}$ and $\langle lt(G') \rangle = \langle lt(G) \rangle$. The proof is completed thanks to lemma 6. □

Definition 6. *A subset $G \subset k[x_1, ..., x_d]$ is a minimal Gröbner basis for an ideal $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ if it is a Gröbner basis for $\mathfrak{a}$, and all $g \in G$ hold:*

(i) $lc(g) = 1$.

(ii) $lt(g) \notin lt(G\backslash\{g\})$

If furthermore,

(iii) None term of g is in $lt(G\backslash\{g\})$

Then we call G a reduced Gröbner basis.

Theorem 13. *Every ideal of $k[x_1, ..., x_d]$ has a unique reduced Gröbner basis.*

Proof: Let any non-zero ideal $\mathfrak{a} \subseteq k[x_1, ..., x_d]$. Because of Hilbert's basis theorem, $\mathfrak{a}$ admits a Gröbner basis G. Thanks to lemma 7, we can delete all elements g in G which satisfy

$$lt(g) \in lt(G\backslash\{g\})$$

and repeat it, we will receive a minimal Gröbner basis for $\mathfrak{a}$, recall $G = \{g_1, ..., g_s\}$. We construct a reduced Gröbner basis from G as follow. Let

$$\begin{aligned} h_1 &= g_1 \text{ rem } \{g_2, ..., g_s\} \\ h_i &= g_i \text{ rem } \{h_1, ..., h_{i-1}, g_{i+1}, ..., g_s\},\ i = 2, ..., s-1 \\ h_s &= g_s \text{ rem } \{h_1, ..., h_{s-1}\} \end{aligned}$$

And set $H = \{h_1, ..., h_s\}$. To prove that H is a reduced basis for $\mathfrak{a}$, we claim that:

Claim: For $1 \leq i \leq s$, we have

(i) No term of h_i is divisible by any term of $\{lt(h_1), ..., lt(h_{i-1}), lt(g_{i+1}), ..., lt(g_s)\}$.

(ii) $lt(h_i) = lt(g_i)$.

In fact, (i) holds thanks to the definition of remainder of Multivariate Division Algorithm. Now we will prove (ii) by induction on i. If $i = 1$, then $lt(h_1) = lt(g_1)$ because $\{g_1, ..., g_s\}$ is a minimal Gröbner basis for $\mathfrak{a}$. Assume for some $1 \leq i \leq s-1$, the claim is true for any $1 \leq j \leq i$. Then $lt(h_j) = lt(g_j),\ \forall j = 1, ..., i$. Thus $\{h_1, ..., h_i, g_{i+1}, ..., g_s\}$ is also a minimal Gröbner basis for $\mathfrak{a}$. Hence, $lt(h_{i+1}) = lt(g_{i+1})$. The claim is proved.

Now thanks to the claim, the set $H = \{h_1, ..., h_s\}$ is a minimal Gröbner basis for $\mathfrak{a}$ that satisfies the third condition of definition 6. Hence, H is a reduced Gröbner basis for $\mathfrak{a}$.

For the uniqueness, assume $H' = \{h'_1, ..., h'_r\},\ r \in \mathbb{N}$ another reduced Gröbner for $\mathfrak{a}$.

We firstly show that $lt(H) = lt(H')$. In fact, for any $h_i \in H$, we have

$$lt(h_i) \in lt(H) \subseteq \langle lt(H)\rangle = \langle lt(H')\rangle$$

Then $lt(h_i)$ is divisible by $lt(h'_j)$ for some $h'_j \in H'$. By a symmetric argument, $lt(h'_j)$ is divisible by $lt(h_l)$ for some $h_l \in H$. So $lt(h_i)$ is divisible by $lt(h_l)$. But H is minimal, so it

implies that $lt(h_i) = lt(h_l) = lt(h'_j) \in lt(H')$. The $h_i \in H$ is arbitrary, then $lt(H) \subseteq lt(H')$. By a symmetric argument, we imply $lt(H) = lt(H')$.

Now, for any $h_i \in H$, $lt(h_i) = lt(h'_j)$ for some $h'_j \in H'$. The element $h_i - h'_j \in \mathfrak{a}$ has no term that divisible by any term of $lt(H) = lt(H')$. Because of the definition of remainder of Multivariate Division Algorithm, $h_i - h'_j$ is itself remainder when we divide $h_i - h'_j$ by H. Since $h_i - h'_j \in \mathfrak{a}$ and H a Gröbner basis, the Ideal Membership Problem say that $h_i - h'_j = 0$. So $h_i = h'_j \in H'$. The $h_i \in H$ is arbitrary, thus $H \subseteq H'$. By a symmetric argument, $H = H'$. □

Most computer algebra systems that have the `Groebner` package contain a function "`Basis`" which computes the reduced Gröbner basis for any given set of generators for an ideal. In this article, we will use **Maple 16**, an computer algebra system that is developed by Waterloo Maple Inc., Ontario, Canada. To use the `Groebner` package, one must start by loading it by the command:

```
>with Groebner;
```

Here, `>` is the Maple prompt, and all Maple commands end with a semicolon. Once the `Groebner` package is loaded, we can compute the reduced Gröbner bases and carry out a variety of other commands that are contained in `Groebner` package.

The format for the `Basis` command is

```
>Basis(F,torder);
```

where `F` is a finite set of polynomials, `torder` one of the monomial orders described below:

- `plex(x[1],...,x[n])`: lexicographic order with x[1] > ... > x[n].
- `grlex(x[1],...,x[n])`: graded lexicographic order with x[1] > ... > x[n].
- `tdeg(x[1] ... x[n])`: graded reverse lexicographic order.

For instance, the commands:

```
>F:=[x^2-y*z,y^2-z*x,z^2-x*y];
>Basis(F,plex(x,y,z));
```

computes the reduced Gröbner basis for ideal $\langle x^2 - yz, y^2 - zx, z^2 - xy\rangle \subset k[x, y, z]$ with the lexicographic order defined by $x > y > z$. The output is

$$\left[y^3 - z^3, -y^2 + zx, -z^2 + yx, x^2 - yz\right]$$

a reduced Gröbner basis.

An other command in `Groebner` package that is usually used is `NormalForm`, for doing Multivariate Division Algorithm. The command of dividing polynomial f by the finite set of polynomials F has the following syntax:

```
>NormalForm(f,F,torder);
```

The output is only the remainder. For example, to divide x^3+3y^3 by $F = \{x^2 - yz, y^2 - zx, z^2 - xy\}$ using graded lexicographic order with $x > y > z$, we would enter:

```
>NormalForm(x^3+3*y^3,F,grlex(x,y,z));
```

The output is $4y^3$, the remainder of this division.

Some other useful Maple commands in the `Groebner` package are:

- `IsBasis` which is to check that the given set of generators is a Gröbner basis or not.
- `LeadingMonomial` (respect `LeadingTerm, LeadingCoefficient`) which computes the leading monomial (respect leading term, leading coefficient) of an polynomial.
- `SPolynomial` which computes the S-polynomial $S(f, g)$ of two polynomials f, g.
- `Solve` which solves an algebraic system by computing a collection of reduced Gröbner bases.

2.4 Some Applications

Given a set of polynomial equations modelling a problem, a Gröbner basis for this system will help simplifying any polynomial in the involved variables. For this reason, Gröbner bases are an important tool to solve computationally many problems in polynomial algebras. In this section, we will see how the Gröbner bases can be used to solve some usual geometric problems.

2.4.1 Solving Polynomial equations

In this subsection, we will discuss a straightforward approach to solving the system of polynomial equations that is based on the elimination property of Gröbner bases with the lexicographic order. Before giving the key theorem, we introduce some notation.

Let $F = \{f_1, ..., f_s\} \subset k[x_1, ..., x_d]$ be a set of polynomials and let $\mathfrak{a} \subset k[x_1, ..., x_d]$ be the ideal generated by F. We denote:

$$\mathbf{V}(\mathfrak{a}) := \left\{(a_1, ..., a_d) \in k^d \mid f(a_1, ..., a_d) = 0 \ \forall f \in \mathfrak{a}\right\}$$

the set of all roots of the system $f_1 = ... = f_s = 0$. We usually call k^d an affine space. It has a natural topology structure for which the closed sets are the $\mathbf{V}(\mathfrak{a})$ for some ideal $\mathfrak{a} \subset k[x_1, ..., x_d]$. It is called the Zariski topology. We also denote

$$\mathfrak{a}_l := \mathfrak{a} \cap k[x_l, ..., x_d] \ \ (l = 1, ..., d)$$

and call it the l-th elimination ideal of $\mathfrak{a}$. Thus $\mathfrak{a}_l$ consists of all consequences of $f_1 = ... = f_s = 0$ not depend on the variables $x_1, ..., x_{l-1}$.

The key of the method to solving polynomial equations we will discuss here is the following theorem:

Theorem 14 (The elimination theorem)**.** *Let $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ be an ideal and G a Gröbner basis of $\mathfrak{a}$ with respect to the lexicographic order such that $x_1 > ... > x_d$. Then for every $l = 1, ..., d$, the set*

$$G_l := G \cap k[x_l, ..., x_d]$$

is a Gröbner basis of the l-th elimination ideal $\mathfrak{a}_l$.

Proof: The proof of this theorem comes from the characteristic property of lexicography order. For any $f \in \mathfrak{a}_l \subset \mathfrak{a}$, the $lt(f)$ is divisible by $lt(g)$ for some $g \in G$. Since $f \in k[x_l, ..., x_d]$ and the definition of lexicographic order with $x_1 > ... > x_d$, $lt(f)$ involves only the variables $x_l, ..., x_d$, hence so is $lt(g)$, and g. So that $g \in G_l$. Hence G_l is a Gröbner for $\mathfrak{a}_l$. □

To describe how the theorem works we come to an example.

Example 8. Solve the following system in $\mathbb{C}^3$:

$$\begin{cases} x^2y - z^3 = 0 \\ 2xy - 4z - 1 = 0 \\ z - y = 0 \end{cases}$$

These equations determine an ideal $\mathfrak{a} = \langle x^2y - z^3, 2xy - 4z - 1, z - y\rangle \subset k[x, y, z]$. And we want to find all points of $\mathbf{V}(\mathfrak{a})$. Now, using the `Basis` command, we compute a Gröbner basis for $\mathfrak{a}$ using the lexicographic order.

```
>Basis([x^2*y-z^3,2*x*y-4*z-1,z-y],plex(x,y,z));
```

The result is $[-1 - 8z - 16z^2 + 4z^4, -z + y, -2z^3 + 2 + x + 8z]$. According to theorem 14, the elimination ideals of $\mathfrak{a}$ are:

$$\begin{aligned} \mathfrak{a}_1 &= \mathfrak{a} = \langle -1 - 8z - 16z^2 + 4z^4, -z + y, -2z^3 + 2 + x + 8z\rangle \\ \mathfrak{a}_2 &= \langle -1 - 8z - 16z^2 + 4z^4, -z + y\rangle \\ \mathfrak{a}_3 &= \langle -1 - 8z - 16z^2 + 4z^4\rangle \end{aligned}$$

Our system is equivalent to the new one:

$$\begin{cases} -2z^3 + 2 + x + 8z = 0 \\ -z + y = 0 \\ -1 - 8z - 16z^2 + 4z^4 = 0 \end{cases}$$

Thus the Elimination theorem use Gröbner bases with the lexicographic order to eliminate certain variables. In this example, we use it to eliminate x, y from our equations to obtain a new polynomial equation that only depends on z. This polynomial is in $\mathfrak{a} \cap \mathbb{C}[z]$ with the smallest possible degree. So that it is easy to find all points of $\mathbf{V}(\mathfrak{a}_3)$, the set of all roots of $-1-8z-16z^2+4z^4 = 0$. Next, when these values of z are substituted into the other equations, these equations can be solved respectively for y, x. Finally, all solutions of our system are $\left(\frac{4+\sqrt{2}}{2}, \frac{4+\sqrt{2}}{2}, \frac{4+\sqrt{2}}{2}\right); \left(\frac{4-\sqrt{2}}{2}, \frac{4-\sqrt{2}}{2}, \frac{4-\sqrt{2}}{2}\right); \left(\frac{-4+\sqrt{2}}{2}, \frac{-4+\sqrt{2}}{2}, \frac{-4+\sqrt{2}}{2}\right); \left(\frac{-4-\sqrt{2}}{2}, \frac{-4-\sqrt{2}}{2}, \frac{-4-\sqrt{2}}{2}\right)$

2.4.2 Implicitization Problem

Suppose that a subset $W \subseteq k^d$ is defined by the parametric equations:

$$\begin{cases} x_1 = f_1(t_1, ..., t_m) \\ .... \\ x_d = f_d(t_1, ..., t_m) \end{cases}$$

where $f_1, ..., f_d$ are polynomials or rational functions of variables $t_1, ..., t_m$. The Implicitization problem asks for the equations defining the Zariski closure Z of W in k^d. In this subsection, for simplicity, we assume all of f_i are polynomials. The case $f_1, ..., f_d$ are rational functions is really similar to our case.

Notice that, the parametrization does not always fill up all Z. In fact, we can think of the parametrization as the map:

$$\phi : k^m \to k^d, \quad (t_1, ..., t_m) \mapsto (f_1(t_1, , .., t_m), ..., f_d(t_1, ..., t_m))$$

and then W is exactly the image of ϕ. Unfortunately, the image of a closed subset may not be closed. It is the reason why the Implicitization problem asks for the Zariski closure of W rather than W itself. However, once the Zariski closure Z of W has been found, one can check if the parametrization fills up all Z or not, and if the answer is no, one can find all missing points. In this subsection, we will only give the way to find Z by using Gröbner bases. This follows directly from the Closure theorem.

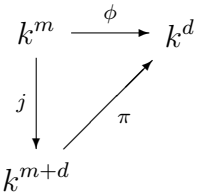
Theorem 15 (Closure theorem)**.** *Let k be an algebraic closed field. Let $\mathfrak{a} = \langle x_1 - f_1, ..., x_d - f_d \rangle \subseteq k[t_1, ..., t_m, x_1, ..., x_d]$ be an ideal and*

$$\mathfrak{a}_{m+1} = \mathfrak{a} \cap k[x_1, ..., x_d]$$

its $(m+1)$-elimination. Then

$$\boldsymbol{V}(\mathfrak{a}_{m+1}) = \overline{\operatorname{Im} \phi}$$

Proof: Break the map ϕ into the composition of the graph of ϕ and the projection from the graph of ϕ to the image of ϕ. We obtain the following commutative diagram of affine varieties and morphisms:



where $j(t_1, ..., t_m) = (t_1, ..., t_m, f_1(t_1, ..., t_m), ..., f_d(t_1, ..., t_m))$ for all $(t_1, ..., t_m) \in k^m$ and $\pi(t_1, ..., t_m, x_1, ..., x_d) = (x_1, ..., x_d)$ for all $(t_1, ..., t_m, x_1, ..., x_d) \in k^{m+d}$. Then we have

$$\operatorname{Im} \phi = \pi(j(k^m)) = \pi(\mathbf{V}(\mathfrak{a}))$$

What we need to prove now is the equality:

$$\mathbf{V}(\mathfrak{a}_{m+1}) = \overline{\pi(\mathbf{V}(\mathfrak{a}))} \tag{2.11}$$

Firstly, the inclusion $\pi(\mathbf{V}(\mathfrak{a})) \subseteq \mathbf{V}(\mathfrak{a}_{m+1})$ is trivial. In fact, any point $P \in \pi(\mathbf{V}(\mathfrak{a}))$ has the form $P = (f_1(t_1, ..., t_m), ..., f_d(t_1, ..., t_m))$ for some $(t_1, ..., t_m) \in k^m$. Then for any polynomial $g \in \mathfrak{a}_{m+1} = \mathfrak{a} \cap k[x_1, ..., x_d]$, and think of g as polynomial in $k[t_1, ..., t_m, x_1, ..., x_d]$, we have

$$\begin{aligned} g(P) &= g(f_1(t_1, ..., t_m), ..., f_d(t_1, ..., t_m)) \\ &= g(t_1, ..., t_m, f_1(t_1, ..., t_m), ..., f_d(t_1, ..., t_m)) = 0 \end{aligned}$$

It means that the point $P \in \pi(\mathbf{V}(\mathfrak{a}))$ vanishes all polynomials in $\mathfrak{a}_{m+1}$. Thus $\pi(\mathbf{V}(\mathfrak{a})) \subseteq \mathbf{V}(\mathfrak{a}_{m+1})$. And then $\overline{\pi(\mathbf{V}(\mathfrak{a}))} \subseteq \mathbf{V}(\mathfrak{a}_{m+1})$.

The proof of the converse inclusion requires Hilbert's Nullstellensatz so that the field k must be algebraic closed. What we want to prove now is the inclusion

$$\mathbf{V}(\mathfrak{a}_{m+1}) \subseteq \overline{\pi(\mathbf{V}(\mathfrak{a}))}$$

Because of Hilbert's Nullstellensatz, it is equivalent to an other inclusion of ideals

$$\mathbf{I}(\pi(\mathbf{V}(\mathfrak{a}))) \subseteq \sqrt{\mathfrak{a}_{m+1}} \tag{2.12}$$

Now let any $g \in \mathbf{I}(\pi(\mathbf{V}(\mathfrak{a}))) \subseteq k[x_1, ..., x_d]$. Then g vanishes at all points of $\pi(\mathbf{V}(\mathfrak{a}))$, i.e.

$$g(f_1(t_1, ..., t_m), ..., f_d(t_1, ..., t_m)) = 0, \ \forall (t_1, ..., t_m) \in k^m$$

Consider g as an element of $k[t_1, ..., t_m, x_1, ..., x_d]$, we certainly have

$$g(t_1, ..., t_m, f_1(t_1, ..., t_m), ..., f_d(t_1, ..., t_m)) = 0, \ \forall (t_1, ..., t_m) \in k^m$$

It means that g vanishes at all points of $\mathbf{V}(\mathfrak{a})$. Because of Hilbert's Nullstellensatz, $g^N \in \mathfrak{a}$ for some $N > 0$. On the other hand, g does not depend on $t_1, ..., t_m$, neither g^N, so $g^N \in \mathfrak{a} \cap k[x_1, ..., x_d] = \mathfrak{a}_{m+1}$. Thus $g \in \sqrt{\mathfrak{a}_{m+1}}$. Hence the inclusion 2.12 holds. The theorem is proved. □

Thanks to the Closure theorem, the method to find the Zariski closure Z of W is clear. The system defining Z is the set of generators of the $(m+1)$-elimination ideal of $\mathfrak{a}$.

Example 9. Find the equations defining the surface whose parametrization is

$$\begin{cases} x = t + u \\ y = t^2 + 2tu \\ z = t^3 + 3t^2u \end{cases}$$

where $t, u \in \mathbb{C}$.

First find the Gröbner basis of the ideal $\mathfrak{a} = \langle x - t - u, y - t^2 - 2tu, z - t^3 - 3t^2u \rangle$ using lexicographic order with $t > u > x > y > z$. We comment:

```
>Basis([x-t-u,y-t^2-2*t*u,z-t^3-3*t^2*u],plex(t,u,x,y,z));
```

The result is $[z^2 - 6yzx + 4y^3 + 4x^3z - 3y^2x^2, 5y^2z - 4yx^2z - 2z^2u - 2z^2x + 2y^3u + y^3x, -yz + 2x^2z + 2xzu - 2y^2u - y^2x, 2y^2 - x^2y - zu - zx + xuy, 3yx - 2x^3 - 2yu + 2x^2u - z, y + u^2 - x^2, -x + t + u]$. Then the third elimination ideal of $\mathfrak{a}$ is the principal ideal generated by $z^2 - 6yzx + 4y^3 + 4x^3z - 3y^2x^2$. Finally the equation defining our surface is

$$z^2 - 6yzx + 4y^3 + 4x^3z - 3y^2x^2 = 0$$

2.4.3 Finding the projective closure of an affine variety

Before presenting the main problem in this subsection, we will give some notation. We denote $\mathbb{P}_k^d$ the classical projective space of dimension d over k. The underlying set is the set of equivalent classes of

$$\left(k^{d+1} - \{0\}\right)/_{\sim}$$

where the equivalent relation $\sim$ is defined for any $x = (x_0, .., x_d), y = (y_0, ..., y_d) \in k^{d+1} - \{0\}$ by

$$x \sim y \Leftrightarrow x = \lambda y, \text{ for some } \lambda \in k$$

For each subset $A \subseteq \mathbb{P}_k^d$, we denote

$$\mathbf{I}_+(A) := \{f \in k[x_0, ..., x_d] \mid f \text{ is homogeneous and } f(x_0, ..., x_d) = 0, \ \forall (x_0, ..., x_d) \in A\}$$

that is an homogeneous ideal of $k[x_0, ..., x_d]$. Conversely, for each homogeneous ideal $\mathfrak{a} \subseteq k[x_0, ..., x_d]$, we denote

$$\mathbf{V}_+(\mathfrak{a}) := \{P = (x_0, ..., x_d) \in \mathbb{P}_k^d \text{ such that } P \text{ vanishes all homogeneous polynomials of } \mathfrak{a}\}$$

$\mathbb{P}_k^d$ has a natural topological structure whose closed subsets are the $\mathbf{V}_+(\mathfrak{a})$ for some homogeneous ideal $\mathfrak{a}$. It is called Zariski topology. The affine space k^d can be embedded continuously via:

$$k^d \to \mathbb{P}_k^d, \ \ (x_1, ..., x_d) \mapsto (1, x_1, ..., x_d)$$

Identify k^d with its image, then k^d is an open subset of $\mathbb{P}_k^d$. So that any affine closed subset of k^d can be seen as a, not necessary closed, subset of $\mathbb{P}_k^d$. In this section, we will give a certain way to find the projective closure of any given affine closed set.

In order to do that, we first define the homogenization of polynomials in $k[x_1, ..., x_d]$. It is expressed by the following map:

$$(-)^h : k[x_1, ..., x_d] \to k[x_0, ..., x_d], \ \ f \mapsto f^h := x_0^{\deg f} f\left(\frac{x_1}{x_0}, ..., \frac{x_d}{x_0}\right)$$

Directly from the definition of $(-)^h$, f^h is a homogeneous polynomial of $k[x_0, ..., x_d]$ and $f^h(1, x_1, ..., x_d) = f(x_1, ..., x_d)$.

If $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ is an ideal, we denote $\mathfrak{a}^h$ an homogeneous ideal of $k[x_0, ..., x_d]$ that is generated by all f^h with $f \in \mathfrak{a}$. Notice that if $\{f_1, ..., f_s\}$ generates $\mathfrak{a}$, then $\{f_1^h, ..., f_s^h\}$ may not generate $\mathfrak{a}^h$. A reason for this fact is that the map $(-)^h$ has no "homomophism" property. However, we have the following theorem.

Theorem 16. *Let $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ be an ideal and let $G = \{g_1, ..., g_s\}$ be a Gröbner basis for $\mathfrak{a}$ with respect to graded lexicographic order with $x_1 > ... > x_d$. Then $G^h = \{g_1^h, ..., g_s^h\}$ generates $\mathfrak{a}^h$.*

Proof: In $k[x_0, ..., x_d]$, we define a new monomial order $>_h$ by:

$$x_1^{a_1}...x_d^{a_d}x_0^{a_0} >_h x_1^{b_1}...x_d^{b_d}x_0^{b_0} \Leftrightarrow \left[\begin{array}{l} x_1^{a_1}...x_d^{a_d} >_{grlex} x_1^{b_1}...x_d^{b_d} \text{ or} \\ x_1^{a_1}...x_d^{a_d} = x_1^{b_1}...x_d^{b_d} \text{ and } a_0 > b_0 \end{array} \right.$$

What we need from the constructing $>_h$ is a monomial order in $k[x_0, ..., x_d]$ which is compatible with the graded lexicographic order in $k[x_1, ..., x_d]$, i.e. for any $f \in k[x_1, ..., x_d]$, $lt_{>_{grlex}}(f) = lt_{>_h}(f)$. In fact, $>_h$ defined above is a monomial order that satisfies our purpose, because the leading term of $f \in k[x_1, ..., x_d]$ is unchanged by homogenization.

Now we will prove that $G^h = \{g_1^h, ..., g_s^h\}$ is indeed a Gröbner basis for ideal $\mathfrak{a}^h \subseteq k[x_0, ..., x_d]$ with respect to $>_h$, so generates $\mathfrak{a}^h$. Clearly $G^h \subset \mathfrak{a}^h$. It is suffices to show that the set $lt_{>_h}(G^h)$ generates $\left\langle lt_{>_h}(\mathfrak{a}^h)\right\rangle$.

Now let any polynomial $Q \in \mathfrak{a}^h$. Notice that $\mathfrak{a}^h$ is the homogeneous ideal of $k[x_0, ..., x_d]$ that is generated by homogenization of polynomials in $\mathfrak{a}$. Then Q has the form:

$$Q = \sum_i q_i f_i^h$$

where the sum is finite, $q_i \in k[x_0, ..., x_d]$, and $f_i \in k[x_1, ..., x_d]$. Setting $x_0 = 1$ and let $q := Q(1, x_1, ..., x_d) \in k[x_1, ..., x_d]$, we obtain

$$q = \sum_i q_i(1, x_1, ..., x_d).f_i^h(1, x_1, ..., x_d) = \sum_i q_i(1, x_1, ..., x_d).f_i(x_1, ..., x_d) \in \mathfrak{a}$$

Since G is a Gröbner basis for $\mathfrak{a}$ with respect to $>_{grlex}$, $lt_{>_{grlex}}(q)$ is divisible by $lt_{>_{grlex}}(g_i)$ for some $g_i \in G$. On the other hand, the setting $q := Q(1, x_1, ..., x_d) \in k[x_1, ..., x_d]$ yields $\deg q \leq \deg Q$ and

$$\begin{aligned} q^h &= x_0^{\deg q}.q\left(\tfrac{x_1}{x_0}, ..., \tfrac{x_d}{x_0}\right) \\ &= x_0^{\deg q}.Q\left(1, \tfrac{x_1}{x_0}, ..., \tfrac{x_d}{x_0}\right) \\ &= \tfrac{x_0^{\deg q}}{x_0^{\deg Q}} Q(x_0, ..., x_d) \end{aligned}$$

It implies that

$$lt_{>_h}(Q) = x_0^{\deg Q - \deg q}.lt_{>_h}(q^h) = x_0^{\deg Q - \deg q}.lt_{>_{grlex}}(q)$$

It means that $lt_{>_h}(Q)$ is divisible by $lt_{>_{grlex}}(q)$, and hence is divisible by $lt_{>_{grlex}}(g_i)$. Because of the aim of the constructing $>_h$, we have $lt_{>_{grlex}}(g_i) = lt_{>_h}(g_i^h)$. Thus $lt_{>_h}(Q)$ is divisible by $lt_{>_h}(g_i^h)$.

Hence G^h is a Gröbner basis for $\mathfrak{a}^h$ with respect to $>_h$. The theorem is proved. □

Theorem 17. *Let k be an algebraic closed field and $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ an ideal. Then $\boldsymbol{V}_+(\mathfrak{a}^h) \subseteq \mathbb{P}_k^d$ is the projective closure of $\boldsymbol{V}(\mathfrak{a}) \subseteq k^d$.*

Proof: Clearly $\mathbf{V}(\mathfrak{a}) \subseteq \mathbf{V}_+(\mathfrak{a}^h) \subseteq \mathbb{P}_k^d$. The converse inclusion requires Hilbert's Nullstellensatz, so the field k must be algebraic closed.

Assume $\mathbf{V}_+(\mathfrak{b}) \subseteq \mathbb{P}_k^d$ another projective closed set that contains $\mathbf{V}(\mathfrak{a})$, where $\mathfrak{b} \subseteq k[x_0, ..., x_d]$ is a homogeneous ideal. We must show that

$$\mathbf{V}_+(\mathfrak{a}^h) \subseteq \mathbf{V}_+(\mathfrak{b})$$

Because of Hilbert's Nullstellensatz, it is equivalent to the inclusion of homogeneous ideals

$$\sqrt{\mathfrak{b}} \subseteq \sqrt{\mathfrak{a}}$$

Now let any homogeneous polynomial $f \in \mathfrak{b}$. As our assumption $\mathbf{V}(\mathfrak{a}) \subseteq \mathbf{V}_+(\mathfrak{b})$, f vanishes at all points of $\mathbf{V}(\mathfrak{a})$, i.e.

$$f(1, x_1, ..., x_d) = 0, \ \forall (x_1, ..., x_d) \in \mathbf{V}(\mathfrak{a})$$

Because of Hilbert's Nullstellensatz, $f^N(1, x_1, ..., x_d) \in \mathfrak{a}$ for some $N > 0$. Thanks to the homogenization, we imply $f^N \in \mathfrak{a}^h$. Then $f \in \sqrt{\mathfrak{a}^h}$. The f is arbitrary, so $\mathfrak{b} \subseteq \sqrt{\mathfrak{a}^h}$. Hence $\sqrt{\mathfrak{b}} \subseteq \sqrt{\mathfrak{a}^h}$. The theorem is proved. □

Now if we combine theorem 16 and 17, we will get an algorithm for computing the projective closure of an affine closed set $W \subseteq k^d$. In fact, if W is defined by the system of polynomials equations $f_1 = ... = f_s = 0$, we first compute the reduced Gröbner basis G for $\{f_1, ..., f_s\}$ with respect to graded lexicographic order, and then the projective closure $Z = \overline{W}$ is defined by the system $g^h = 0, \ \forall g \in G$.

Example 10. Find the projective closure of a curve (C) in $\mathbb{C}^3$ whose parametrization is:

$$\begin{cases} x = t^3 \\ y = t^4 \\ z = t^5 \end{cases}$$

where $t \in \mathbb{C}$.

We first find the Gröbner basis of the ideal $\mathfrak{a} = \langle x - t^3, y - t^4, z - t^5 \rangle$ using lexicographic order with $t > x > y > z$. We enter:

```
>Basis([x-t^3,y-t^4,z-t^5],plex(t,x,y,z));
```

The output is $[y^5 - z^4, -y^2 + xz, y^3x - z^3, -z^2 + yx^2, x^3 - yz, -x^2 + tz, ty - z, tx - y, -x + t^3]$. Then the second elimination ideal of $\mathfrak{a}$ which express the affine closure of (C) is

$$\mathfrak{a}_2 = \left\langle y^5 - z^4, -y^2 + xz, y^3x - z^3, -z^2 + yx^2, x^3 - yz \right\rangle$$

Next we find the reduced Gröbner basis for $\mathfrak{a}_2$ with respect to graded lexicographic order:

```
>Basis([y^5-z^4, -y^2+x*z, y^3*x-z^3, -z^2+y*x^2, x^3-y*z],plex(t,x,y,z));
```

The result is $[-y^2 + xz, -z^2 + yx^2, x^3 - yz, y^3x - z^3, y^5 - z^4]$. Finally, taking their homogenization, we obtain a homogeneous ideal of $\mathbb{C}[t, x, y, z]$

$$\left\langle -y^2 + xz, -tz^2 + yx^2, x^3 - yzt, y^3x - tz^3, y^5 - tz^4 \right\rangle$$

that expresses the projective closure of (C) in $\mathbb{P}^3_{\mathbb{C}}$.

Chapter 3

Using reduction modulo ideals to solve multivariate interpolation problems

This chapter is the heart of the thesis in which the Multivariate Polynomial Interpolation problem will be solved by the strong connection within the theory of polynomial ideals. After recalling the main problem, we will analyse the main idea for this method.

Problem 5 (Multivariate Polynomial Interpolation)**.** *Fix the commutative field k, and positive integer N, d. Let $N+1$ distinct interpolation points $X_0, ..., X_N \in k^d$ and $N+1$ scalar values $u_0, ..., u_N \in k$. Find a polynomial $P \in k[x_1, ..., x_d]$ with the smallest possible degree such that*

$$P(X_i) = u_i, \ \forall i = 0, ..., N \tag{3.1}$$

The system (3.1) is called the interpolation conditions.

To connect to the theory of polynomial ideals, we denote the evaluation map:

$$eva : k[x_1, ..., x_d] \to k^{N+1}, \ \ f \mapsto (f(X_0), ..., f(X_{N+1}))$$

Then the problem 5 can be said again that finding the inverse image of eva for each given vector $(u_0, ..., u_N) \in k^{N+1}$.

Clearly eva is a k-linear map. It is furthermore a surjective morphism. In the second section of this chapter, an algorithm for finding an inverse image of any given vector $(u_0, ..., u_{N+1}) \in k^{N+1}$ will be presented. The kernel of eva is

$$\mathfrak{a} = \{f \in k[x_1, ..., x_d] \,|\, f(X_0) = ... = f(X_{N+1}) = 0\}$$

that is an ideal of $k[x_1, ..., x_d]$ which expresses the set of interpolation points. We call it **interpolating ideal**. Thus eva induces an isomorphism of k-vector spaces

$$k[x_1, ..., x_d]/\mathfrak{a} \to k^{N+1}$$

It means that there is a one-to-one correspondence between the vectors in k^{N+1} and equivalent classes of modulo $\mathfrak{a}$ in $k[x_1, ..., x_d]$. Hence the problem of finding an interpolating polynomial f for given vector $(u_0, ..., u_{N+1}) \in k^{N+1}$ with the smallest possible degree is equivalent to finding a representative of the equivalent class $f + \mathfrak{a}$ with the smallest possible degree. For this reason, reduction modulo ideals is the key of the method.

3.1 Interpolating ideals

In this section we will present an algorithm for finding the interpolating ideals that are based on the relation between the monomial ideals of $k[x_1, ..., x_d]$ and the complement of ideals in $\mathbb{N}^d$. The cornerstone of this method is disclosing the geometric connection between the set A of interpolation points and the minimal basis of monomial ideal $\langle lt(\mathfrak{a})\rangle$ to find the reduced Gröbner basis for $\mathfrak{a}$ with respect to lexicographic order. And then an algorithm for finding the reduced Gröbner basis with respect to any other monomial order by changing ordering from the lexicographic order case.

3.1.1 Lower sets

Recall $\mathbb{M}_d$ be the set of all monomials in $k[x_1, ..., x_d]$. We have seen that $\mathbb{M}_d$ forms a commutative unitary semi-group with basis $\{x_1, ..., x_d\}$. Then it is isomorphic with $\mathbb{N}^d$ via the canonical map:

$$\mathbb{M}_d \to \mathbb{N}^d,\ X^a = x_1^{a_1}...x_d^{a_d} \mapsto (a_1, ..., a_d)$$

We have also seen that there is a one-to-one correspondence between the monomials ideals of $k[x_1, ..., x_d]$ and the ideals of $\mathbb{M}_d$ that is expressed by the map:

$$\begin{array}{ccc} \left\{ \begin{array}{c} \text{monomial ideals} \\ \text{of } k[x_1, ..., x_d] \end{array} \right\} & \to & \left\{ \begin{array}{c} \text{ideals} \\ \text{of } \mathbb{M}_d \end{array} \right\} \\ \mathfrak{b} & \mapsto & \mathfrak{b} \cap \mathbb{M}_d \end{array}$$

Putting together two such bijective maps, we obtain a one-to-one correspondence between monomial ideals of $k[x_1, ..., x_d]$ and the ideals of $\mathbb{N}^d$, namely:

$$\varphi : \left\{ \begin{array}{c} \text{monomial ideals} \\ \text{of } k[x_1, ..., x_d] \end{array} \right\} \to \left\{ \begin{array}{c} \text{ideals} \\ \text{of } \mathbb{N}^d \end{array} \right\}$$

where φ corresponds each monomials $\mathfrak{b}$ of $k[x_1, ..., x_d]$ with the set of exponents of monomials in $\mathfrak{b}$. In particular, φ is indeed an isomorphism of lattices.

Now we will disclose the connection between the bijective map φ with our problem. Recall the interpolating ideal $\mathfrak{a} = \mathbf{I}(A)$. The ideal $\langle lt(\mathfrak{a})\rangle$ is a monomial ideal of $k[x_1, ..., x_d]$. We associates to it an ideal of $\mathbb{N}^d$, that is $\varphi(\langle lt(\mathfrak{a})\rangle)$. Associate monomials of each point in its completion, i.e. $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$, are k-linear independent in $k[x_1, ..., x_d]/\mathfrak{a}$. On the other hand, since the set A of interpolation points is a finite set, $k[x_1, ..., x_d]/\mathfrak{a}$ is a k-vector space of finite dimension. Hence the set $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$ is finite. Furthermore, the cardinality of $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$ is in fact the dimension of $k[x_1, ..., x_d]/\mathfrak{a}$ as a k-vector space.

Proposition 5. *With notation above, we have:*

$$\#A = \dim_k k[x_1, ..., x_d]/\mathfrak{a} = \#\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$$

Proof: The first equality comes from the Chinese Remainder Theorem. In fact we have:

$$\mathfrak{a} = \mathbf{I}(A) = I\left(\bigcup_{a\in A}\{a\}\right) = \bigcap_{a\in A} I\left(\{a\}\right)$$

The $I\left(\{a\}\right)$ are the distinct maximal ideals of $k[x_1, ..., x_d]$, then the Chinese Remainder Theorem gives us

$$k[x_1, ..., x_d]/\mathfrak{a} \cong k[x_1, ..., x_d]/\bigcap_{a\in A} I\left(\{a\}\right) \cong \prod_{a\in A} k[x_1, ..., x_d]/\mathbf{I}\left(\{a\}\right) \cong k^{\#A}$$

Thus

$$\dim_k k[x_1, ..., x_d]/\mathfrak{a} = \#A$$

In order to prove the second one, we consider the images of the points of $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$ in $k[x_1, ..., x_d]/\mathfrak{a}$ via the composition:

$$\mathbb{N}^d \to \mathbb{M}_d \to k[x_1, ..., x_d] \to k[x_1, ..., x_d]/\mathfrak{a}$$

We claim that the images of $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$ in $k[x_1, ..., x_d]/\mathfrak{a}$ forms a k-basis of $k[x_1, ..., x_d]/\mathfrak{a}$.

Firstly, if $\{a^1, ..., a^s\}$ is any finite subset of $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$ and $\{X^{a^1}, ..., X^{a^s}\}$ its image in $k[x_1, ..., x_d]/\mathfrak{a}$. Then $\{X^{a^1}, ..., X^{a^s}\}$ is independent over k. In fact, if there is a k-linear combination, namely $c_1X^{a^1} + ...c_sX^{a^s}$ with $c_1, ..., c_s \in k$, that vanishes in $k[x_1, ..., x_d]/\mathfrak{a}$, then $c_1X^{a^1} + ...c_sX^{a^s} \in \mathfrak{a}$. If $c_1X^{a^1} + ...c_sX^{a^s} \neq 0$, then its leading term is a X^{a^i} for some $i = 1, ..., s$ that is belong to $lt(\mathfrak{a})$. But it is contradiction with the setting $a^i \in \mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$. Thus we must have $c_1X^{a^1} + ...c_sX^{a^s} = 0$, so $c_1 = ... = c_s = 0$. Hence the image of $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$ in $k[x_1, ..., x_d]/\mathfrak{a}$ is k-linear independent.

To conclude the proposition, we only need to prove that the image of $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$ in $k[x_1, ..., x_d]/\mathfrak{a}$ spans the whole $k[x_1, ..., x_d]/\mathfrak{a}$. Let $\{f_1, ..., f_s\}$ be a Gröbner basis for $\mathfrak{a}$, then $\{lt(f_1), ..., lt(f_s)\}$ spans $\langle lt(\mathfrak{a})\rangle$. For any polynomial $f \in k[x_1, ..., x_d]$, using Multivariate Division Algorithmto divide f by the basis $\{f_1, ..., f_s\}$ gives the remainder $r \in k[x_1, ..., x_d]$ such that either $r = 0$ or no term of r is divisible by any $lt(f_i),\ i = 1, ..., s$. If $r = 0$, then $f \in \mathfrak{a}$ and vanishes in $k[x_1, ..., x_d]/\mathfrak{a}$. Otherwise, the points in $\mathbb{N}^d$ that correspond with the terms of r is in $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$. Then $f = r \mod \mathfrak{a}$ is a k-linear combination of the image of $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$ in $k[x_1, ..., x_d]/\mathfrak{a}$. The proposition is proved. □

The proposition gives the first connection between the interpolating ideal $\mathfrak{a}$ and the set $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$. To analyse some further information of $\mathfrak{a}$ from the set $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$, we will consider the geometrical representation of such set. Assume a minimal Gröbner basis $\{f_1, ..., f_s\}$ is given. Then the set of all monomials in $\langle lt(\mathfrak{a})\rangle$ is a semi-subgroup of $\mathbb{M}_d$ that is generated by $\{lt(f_1), ..., lt(f_s)\}$. In the proof of Proposition 5, we showed that the set $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$ is exactly the set of point $a \in \mathbb{N}^d$ such that the monomial X^a is divisible by none of $lt(f_i),\ i = 1, ..., s$.

For example, the figure 3.1 is the geometric expression of $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$, where $\langle lt(\mathfrak{a})\rangle = \langle x^4, x^3y, x^2y^3, y^4\rangle$.

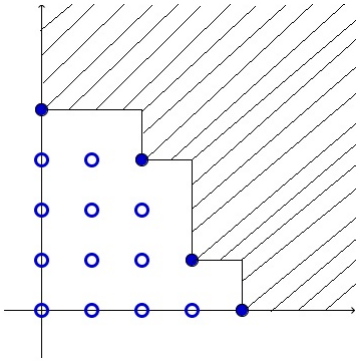


Figure 3.1: $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$ is the set of blank circle.

Definition 7.
- *A finite set $D \subset \mathbb{N}^d$ is called a lower up set if $\mathbb{N}^d - D$ forms an ideal of $\mathbb{N}^d$. We denote $\mathbb{D}_d$ the class of all lower up set of $\mathbb{N}^d$.*
- *For each lower up set $D \in \mathbb{D}_d$, the minimal basis of the ideal $\mathbb{N}^d - D$ is called the set of limit points of D, namely $E(D)$.*

Example 11. The set of the blank circles in the figure 3.1 is a lower up set of $\mathbb{N}^2$. Its limit points is the solid circles.

Return to our problem of finding the interpolating ideal $\mathfrak{a} = \mathbf{I}(A)$ of the given set of interpolation points A. If we know the lower set associated with the monomial ideal $\langle lt(\mathfrak{a})\rangle$ by some ways, then the set of limits points give us the minimal basis of $\langle lt(\mathfrak{a})\rangle$. The question is how to can construct the lower set D from A geometrically. In fact, each monomial order in $k[x_1, ..., x_d]$ yields a different rule to construct the associated lower set.

In the next subsection, the lower set associated with A will be constructed by induction on the dimension d. To pass from $d-1$ to d, we need an operation on $\mathbb{D}_d$ that called addition. Before give the additive operator in $\mathbb{D}_d$, we define some useful notation.

For each $i = 1, ..., d$, we denote the projections

$$p_i : \mathbb{N}^d \to \mathbb{N}, \;\; a = (a_1, ..., a_d) \mapsto a_i$$

and

$$\widehat{p_i} : \mathbb{N}^d \to \mathbb{N}^{d-1}, \;\; a = (a_1, ..., a_d) \mapsto (a_1, ..., a_{i-1}, a_{i+1}, ..., a_d)$$

We usually use the projection to the last coordinate p_d and to the first $d-1$ coordinates $\widehat{p_d}$, so we always write $p(a)$ (respect $\widehat{p}(a)$) to replace $p_d(a)$ (respect $\widehat{p_d}(a)$).

Each projection $\widehat{p_i}$ induces a natural map from $\mathbb{D}_d$ to $\mathbb{D}_{d-1}$ that corresponds each lower set $D \in \mathbb{D}_d$ to a lower set $\widehat{p_i}(D) \in \mathbb{D}_{d-1}$. Conversely each lower set in $\mathbb{D}_{d-1}$ can be seen as a lower set on $\mathbb{D}_d$ via the embedding:

$$\mathbb{N}^{d-1} \to \mathbb{N}^d, \;\; a = (a_1, ..., a_d) \mapsto (a_1, ..., ..., a_{d-1}, 0)$$

Now the addition map in $\mathbb{D}_d$ can be defined as follow:

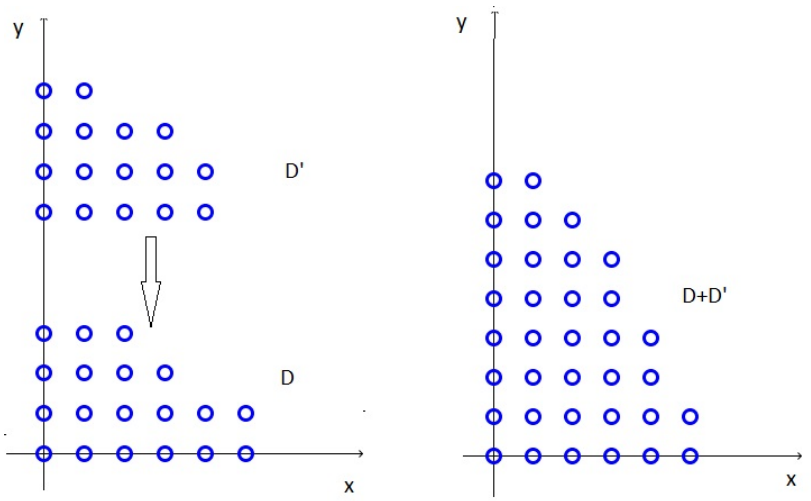


Figure 3.2: $\mathbb{N}^d - \varphi(\langle lt(\mathfrak{a})\rangle)$ is the set of blank circle.

Definition 8. *For $D, D' \in \mathbb{D}_d$, define $D + D'$ to be the set of all points $c \in \mathbb{N}^d$ that satisfy:*

(i) $p(c) < \#\{a \in D \,|\, \widehat{p}(a) = \widehat{p}(c)\} + \#\{b \in D' \,|\, \widehat{p}(b) = \widehat{p}(c)\}$ *and*

(ii) $\widehat{p}(c) \in \widehat{p}(D) \cup \widehat{p}(D')$

Look at an example in the figure 3.2. To get $D + D'$, we first draw D and D' in the same coordinate system $\mathbb{N}^2$ such that D and D' are putted in the semi-line Oy and do not intersect. Then drop the elements of D' down along the Oy until they lie on top of elements of D. The result is $D + D'$.

Lemma 8. *The operator "+" defined above is an addition on $\mathbb{D}_d$ and then $\mathbb{D}_d$ becomes a commutative semi-group with the empty set as neutral element.*

Proof: We must check that the following are true:

(i) "+" is indeed a birelation in $\mathbb{D}_d$, i.e. $D + D' \in \mathbb{D}_d$ for every $D, D' \in \mathbb{D}_d$.

(ii) "+" is associative, i.e. $(D + D') + D" = D + (D' + D")$ for any $D, D', D" \in \mathbb{D}_d$.

(iii) "+"is commutative, i.e. $D + D' = D' + D$ for all $D, D' \in \mathbb{D}_d$.

(iv) Empty set is a neutral element, i.e. $D + \emptyset = \emptyset + D = D$ for all $D \in \mathbb{D}_d$.

All of them are come directly from the definition of "+" and the properties of lower up sets. The proof of the third final items is trivial. To prove the first one, let any $D, D' \in \mathbb{D}_d$, we must show that the complement of $D + D'$ in $\mathbb{N}^d$ forms an ideal of $\mathbb{N}^d$.

Recall the definition of $D + D'$:

$$D + D' = \{c \in \mathbb{N}^d, \ | \, c \text{ satisfies (i) and (ii)}\}$$

where

(i) $p(c) < \#\{a \in D \,|\, \widehat{p}(a) = \widehat{p}(c)\} + \#\{b \in D' \,|\, \widehat{p}(b) = \widehat{p}(c)\}$ and

(ii) $\widehat{p}(c) \in \widehat{p}(D) \cup \widehat{p}(D')$

Let any $u = (u_1, ..., u_d) \in \mathbb{N}^d - (D + D')$. Then u does not satisfies either (i) or (ii).

If u does not hold (i), i.e.

$$p(u) < \#\{a \in D \,|\, \widehat{p}(a) = \widehat{p}(u)\} + \#\{b \in D' \,|\, \widehat{p}(b) = \widehat{p}(u)\}$$

Then for any $v = (v_1, ..., v_d) \in \mathbb{N}^d$, we have

$$p(uv) = u_d v_d \geq u_d = p(u) \geq \#\{a \in D \,|\, \widehat{p}(a) = \widehat{p}(u)\} + \#\{b \in D' \,|\, \widehat{p}(b) = \widehat{p}(u)\}$$

It means that uv also does not hold (i), then $uv \in \mathbb{N}^d - (D + D')$, for every $v \in \mathbb{N}^d$.

On the other hand, if u does not hold (ii), i.e.

$$\widehat{p}(u) = (u_1, ..., u_{d-1}) \notin \widehat{p}(D) \cup \widehat{p}(D')$$

then neither $\widehat{p}(u) \notin \widehat{p}(D)$ nor $\widehat{p}(u) \notin \widehat{p}(D')$. Notice that $\widehat{p}(D)$ and $\widehat{p}(D')$ are lower sets on $\mathbb{N}^{d-1}$. Now by induction on the dimension d, we can assume that $\widehat{p}(uv) = \widehat{p}(u)\widehat{p}(v) \notin \widehat{p}(D)$ and $\widehat{p}(uv) = \widehat{p}(u)\widehat{p}(v) \notin \widehat{p}(D')$ for every $v \in \mathbb{N}^d$. So $\widehat{p}(uv) \notin \widehat{p}(D) \cup \widehat{p}(D')$, i.e. uv also does not hold (ii).

Finally we have proved that if $u \in \mathbb{N}^d - (D + D')$ then so is uv for any $v \in \mathbb{N}^d$. Of course $D + D'$ is a finite set. Hence $D + D'$ is indeed a lower set in $\mathbb{N}^d$. □

In particular, given a finite family $\{D_i\}_{i \in I}$ in $\mathbb{D}_d$, we can form the sum

$$\sum_{i \in I} D_i \in \mathbb{D}_d$$

It is the set of points $c \in \mathbb{N}^d$ that satisfy

(i) $p(c) < \sum\limits_{i \in I} \#\{a \in D_i \,|\, \widehat{p}(a) = \widehat{p}(c)\}$ and

(ii) $\widehat{p}(c) \in \bigcup\limits_{i \in I} \widehat{p}(D_i)$

3.1.2 The Gröbner basis of interpolating ideal with respect to $<_{lex}$

. In this subsection we only work with the lexicographic order. Now we have enough to construct the associated lower set to the interpolating ideal. Recall that $A \subset \mathbb{N}^d$ is the set of interpolation points. We will define a way to translate geometrically the set A into the associate lower set $D_{<_{lex}}(A)$, or shortly $D(A)$, that contains all information about the leading terms of the reduced Gröbner basis for the interpolating ideal $\mathfrak{a} = \mathbf{I}(A)$.

The lower set $D(A)$ is constructed by induction on the dimension d as follow:

Algorithm 8 Addition of lower sets

Require: Two lower sets D, D' in $\mathbb{N}^d$.
Ensure: $D'' = D + D'$.
1: $D'' := D$
2: **if** $D' = \emptyset$ **then**
3: return D''
4: **else**
5: pick any $a = (a_1, ..., a_d) \in D'$.
6: **end if**
7: **if** $a \in D''$ **then**
8: $a_d := a_d + 1$, repeat round **if**
9: **else**
10: $D'' := D'' \cup \{a\}$
11: $D' := D' - \{a\}$
12: repeat the first round **if**.
13: **end if**

- For $d = 1$, we set $D(A) = \{0, 1, ..., \#A - 1\}$.
- To pass from $d - 1$ to d, we consider the family of slices of A that are defined by

$$H(a_d) := \{b \in A \,|\, p(b) = a_d\}$$

where a_d runs on the whole $p(A)$. In other words, the slice $H(a_d),\ a_d \in p(A)$ is the set of points in A that have the same d-th coordinate and equal to a_d. We think of each $H(a_d)$ as a subset of $\mathbb{N}^{d-1}$ via the projection:

$$\widehat{p} : \mathbb{N}^d \to \mathbb{N}^{d-1},\ \ b = (b_1, ..., b_d) \mapsto (b_1, ..., b_{d-1})$$

Then the lower sets $D(H(a_d))$ are well-defined by induction hypothesis. Finally, we have

$$D(A) = \sum_{a_d \in p(A)} D(H(a_d))$$

We can see how the process works on the figure 3.3 with an example in $\mathbb{N}^2$. The interpolation points are the blank points in the left hand picture. A can be seen as the discrete union of the family pieces $H(a_d),\ a_d \in p(A)$ that are that sets of all points which have the same second coordinates. Each such set can be seen as a subset of $\mathbb{N}$ if we forget the second coordinates, then $D(H(a_d))$ is defined as in the univariate dimension case. Finally the sum of all such $D(H(a_d))$ gives us the associate lower set $D(A)$ in the right hand picture which we want to find.

This process can be formulated into an algorithm as the following:
The following is one of the main theorem of this chapter.

Theorem 18. *Let $A, D(A), \mathfrak{a}$ be notation as above. Then the ideal $\mathbb{N}^d - D(A)$ defines the monomial ideal $\langle lt(\mathfrak{a})\rangle$, i.e. for any $\alpha \in D(A)$, there exists a polynomial $\phi_\alpha \in \mathfrak{a}$ such that $lt(\phi) = X^\alpha$. Furthermore, the set $\{\phi_\alpha \,|\, \alpha \in E(D(A))\}$ forms a minimal Gröbner basis for $\mathfrak{a}$.*

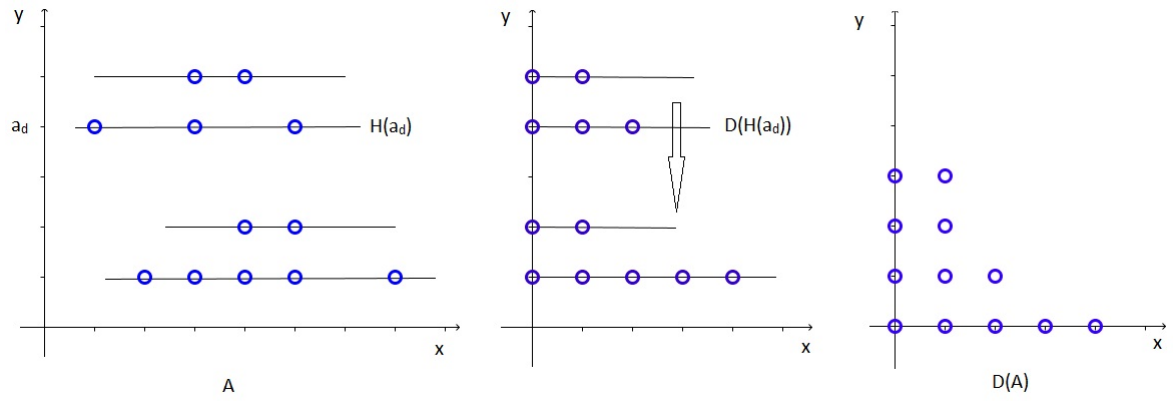


Figure 3.3: Construct $D(A)$ from A.

Algorithm 9 Find the associated lower set

Require: A finite set $A \subset \mathbb{N}^d$.
Ensure: The associated lower set $D := D(A)$.
1: **if** $d = 1$ **then**
2: $D := \{0, 1, ..., \#A - 1\}$
3: **end if**
4: Compute $p(A) \subset \mathbb{N}$.
5: For each $a_d \in p(A)$, compute $H(a_d) := \{b \in A \,|\, p(b) = a_d\}$.
6: Call the algorithm recursively to compute $D(H(a_d))$.
7: Repeating algorithm 8 many times to compute $\sum_{a_d \in p(A)} D(H(a_d))$.
8: Return D.

Proof: This theorem will be proved in 2 steps. First is an algorithm that computes for each $\alpha \in E(D(A))$ a polynomial $\phi_\alpha \in \mathfrak{a}$ such that $lt(\phi) = X^\alpha$. And then the relation between the ideals of $\mathbb{N}^d$ and the monomial ideals of $k[x_1, ..., x_d]$ will finishes the proof. The first step is rather long and very important for us, then it will be described carefully in the next subsection.

In this case, we assume the first step is done, i.e the set $\phi_\alpha \in \mathfrak{a}$ such that $lt(\phi) = X^\alpha$ is defined. Then we have $lt(\phi_\alpha) \in \langle lt(\mathfrak{a})\rangle$ for every $\alpha \in E(D(\mathfrak{a}))$. Because of the definition of lower set, the set of exponents of ϕ_αs, i.e. $E(D(A))$, forms a minimal basis for ideal $\mathbb{N}^d - D(A)$. Then the set of $lt(\phi_\alpha)$s forms a minimal basis for the monomial ideal $\langle lt(\mathfrak{a})\rangle$. Hence, $\{\phi_\alpha \,|\, \alpha \in E(D(A))\}$ is a minimal Gröbner basis for $\mathfrak{a}$. □

3.1.3 Constructing the reduced Gröbner basis with respect to $<_{lex}$

For each $\alpha \in E(D(A))$, the polynomial ϕ_α can be constructed by induction on the dimension d. In order to do that, we see the set $H(a_d) = \{b \in A \,|\, p(b) = a_d\}$ for each $a_d \in p(A)$ as a lower set of $\mathbb{N}^{d-1}$ via the projection:

$$\widehat{p} : \mathbb{N}^d \to \mathbb{N}^{d-1}, \;\; u = (u_1, ..., u_d) \mapsto (u_1, ..., u_{d-1})$$

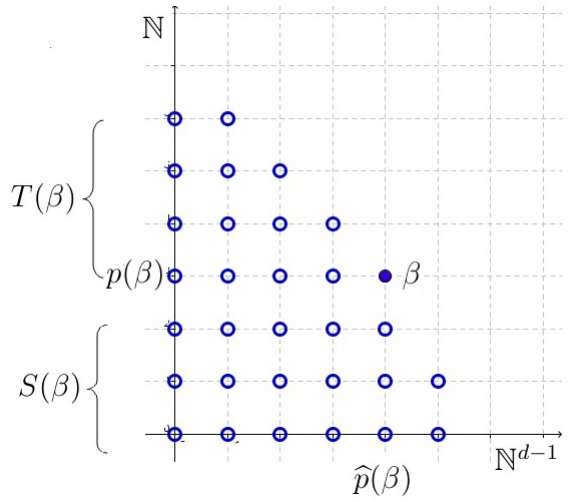


Figure 3.4: $T(\beta)$ and $S(\beta)$ with $\beta \in E(D(A))$.

In the case $d = 1$, A is just a finite distinct points in a line, namely $\{m_0, ..., m_N\} \subset \mathbb{N}$. Then $D(A) = \{0, 1, ..., N\}$ and $E(D(A)) = \{N + 1\}$. In this case, the product $(x - m_0)...(x - m_N)$ that is of degree $N + 1$ generates the interpolating ideal. From here, for any $n \in \mathbb{N} - D(A) = \{N + 1, N + 2, ...\}$, dividing x^n by $(x - m_0)...(x - m_N)$ gives us the remainder $r(x)$ that is of degree less than N. Thus the polynomial $x^n - r(x)$ is in $\mathfrak{a}$ such that all exponents of its non-leading terms are in $D(A)$.

To pass from $d - 1$ to d, we need the inductive assumption.

Inductive assumption We assume the following holds for every slice $H(a_d)$ of A with $a_d \in p(A)$. For any $\alpha \in \mathbb{N}^{d-1} - D(H(a_d))$, we can compute a polynomial $f_{\alpha,a_d} \in k[x_1, ..., x_d]$ that satisfies:

(i) $lt(f_{\alpha,a_d}) = X^\alpha$ and

(ii) $f_{\alpha,a_d}(b) = 0$ for all $b \in H(a_d)$.

(iii) The exponents of all non-leading terms of f_{α,a_d} lie in $D(H(a_d))$.

Now let any $\beta \in E(D(A))$, we split the set $p(A)$ in two components:

$$S(\beta) := \{a_d \in p(A) \,|\, \widehat{p}(\beta) \in D(H(a_d))\}$$
$$T(\beta) := p(A) - S(\beta)$$

The definition of $T(\beta)$ and $S(\beta)$ is illustrated in figure 3.4. As the definition of $T(\beta)$ and $S(\beta)$, the d-th coordinate of β is exactly the cardinality of $S(\beta)$. It is also the degree of the variable x_d in the polynomial ϕ_β that we are looking for.

To kill all points in A whose projection on the d-th coordinate belongs to $S(\beta)$, we use the product

$$\pi_\beta = \prod_{a_d \in S(\beta)} (x_d - a_d)$$

We will find ϕ_β of the form $\phi_\beta = \pi_\beta\theta_\beta$ for some polynomial $\theta_\beta \in k[x_1, ..., x_d]$ which need to find. Because of the construction of π_β, its leading term is $x_d^{\#S(\beta)}$, where $\#S(\beta)$ is exactly the exponent of x_d in the leading term of ϕ_β. Thus the leading term of θ_β must be $\widehat{p}(X)^{\widehat{p}(\beta)} = x_1^{\beta_1}...x_{d-1}^{\beta_{d-1}}$. Furthermore, the polynomial θ_β must kill all points of A whose projection to the d-th coordinate is belong to $T(\beta)$.

We will construct θ_β by using the inductive assumption. According to the above assumption, for each $a_d \in T(\beta)$, there is a polynomial $f_{\widehat{p}(\beta),a_d} \in k[x_1, ..., x_{d-1}]$ that satisfies (i)-(iii). Write this polynomial as

$$f_{\widehat{p}(\beta),a_d} = \widehat{p}(X)^{\widehat{p}(\beta)} + g_{\widehat{p}(\beta),a_d}$$

where $g_{\widehat{p}(\beta),a_d} \in k[x_1, ..., x_{d-1}]$ is the sum of non-leading terms of $f_{\widehat{p}(\beta),a_d}$. Next we define:

$$\theta_\beta = \widehat{p}(X)^{\widehat{p}(\beta)} + \sum_{a_d \in T(\beta)} \chi(T(\beta), a_d) g_{\widehat{p}(\beta),a_d}$$

where $\chi(T(\beta), a_d) := \prod_{b_d \in T(\beta)-\{a_d\}} \frac{x_d - b_d}{a_d - b_d}$ is the characteristic polynomial of $a_d \in T(\beta)$.

Clearly θ_β kills all points of A whose the d-coordinate belongs to $T(\beta)$. In fact, if $b = (b_1, ..., b_d) \in A$ is such that $b_d \in T(\beta)$ then

$$\chi(T(\beta), a_d)(b_d) = \begin{cases} 0 \text{ if } a_d \neq b_d \\ 1 \text{ if } a_d = b_d \end{cases}$$

Thus

$$\theta_\beta(b) = \widehat{p}(b)^{\widehat{p}(\beta)} + g_{\widehat{p}(\beta),b_d} = f_{\widehat{p}(\beta),b_d}(b) = 0 \ \ (\text{since } b \in H(b_d))$$

Now, setting $\phi_\beta := \pi_\beta\theta_\beta$ then

(i) $lt(\phi_\beta) = lt(\pi_\beta).lt(\theta_\beta) = X^\beta$.

(ii) ϕ_β kills all points of A.

Look at the exponents of non-leading terms of ϕ_β. By inductive assumption, the exponents of all terms of $g_{\widehat{p}(\beta),a_d}$, for each $a_d \in T(\beta)$, lie in $D(H(a_d))$. The character polynomials $\chi(T(\beta), a_d)$ is a polynomial with only variable x_d of degree $|T(\beta)| - 1$. So that the exponents of all non-leading terms of θ_β lie in

$$\left(\bigcup_{a_d \in T(\beta)} D(H(a_d))\right) \times \{0, 1, ..., |T(\beta)| - 1\} \subset \mathbb{N}^d$$

π_β is also a polynomial with only variable x_d of degree $S(\beta)$. Thus the exponents of non-leading terms of ϕ_β is contained in the set

$$Exp(\phi_\beta) := \left[\left(\bigcup_{a_d \in T(\beta)} D(H(a_d))\right) \times \{0, 1, ..., |T(\beta)| - 1\}\right] \bigcup [\{\widehat{p}(\beta)\} \times \{0, 1, ..., |S(\beta)| - 1\}]$$

which can be illustrated in the figure 3.5

There are maybe some non-leading terms of ϕ_β whose exponents do not lie in $D(A)$. So that ϕ_β is not necessary reduced modulo $\mathfrak{a} = \mathbf{I}(A)$. On the other words, we only obtain the following proposition:

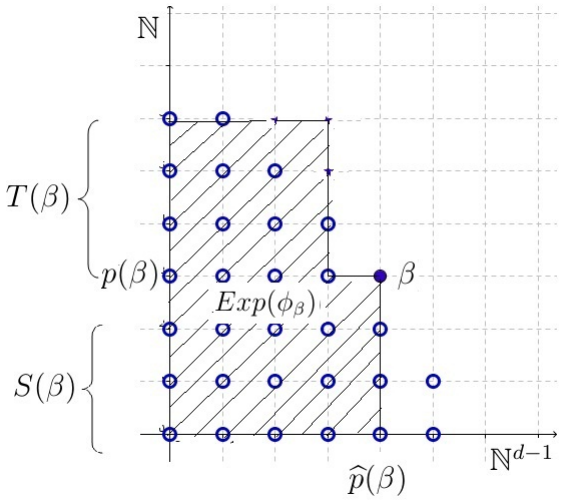


Figure 3.5: The exponents of non-leading terms of ϕ_β.

Proposition 6. *The set* $\{\phi_\beta \mid \beta \in E(D(A))\}$ *forms a minimal Gröbner basis for ideal* $\mathfrak{a}$.

Example 12. Let the interpolation set A as the figure 3.6. Then $p(A) = \{1, 2, 3, 4\} \subset Oy$, and $H(1), H(2), H(3), H(4)$ as in the figure.

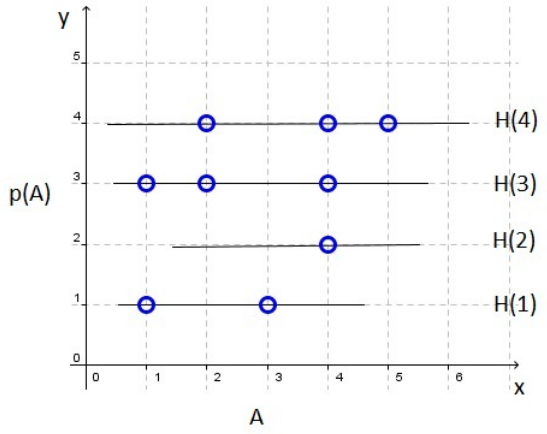


Figure 3.6: The interpolating set A.

The set $D(H(1)), D(H(2)), D(H(3)), D(H(4))$ is described in the left and the associated lower set $D(A)$ in the right of figure 3.7. Then the set of limit points of $D(A)$ is $E(D(A)) = \{(0, 4), (1, 3), (2, 2), (3, 0)\}$. So a minimal basis for $\langle lt(\mathbf{I}(A))\rangle$ is $\{y^4, xy^3, x^2y^2, x^3\}$. Now we will construct for each $\beta = (\beta_1, \beta_2) \in E(D(A))$ a polynomial $\phi_\beta \in k[x, y]$ whose leading term is $x^{\beta_1}y^{\beta_2}$.

First is the case $\beta = (0, 4)$, the smallest element of $E(D(A))$. The leading term of polynomial $\phi_{(0,4)}$ which we need to find is y^4. So that $\phi_{(0,4)}$ is in the elimination ideal

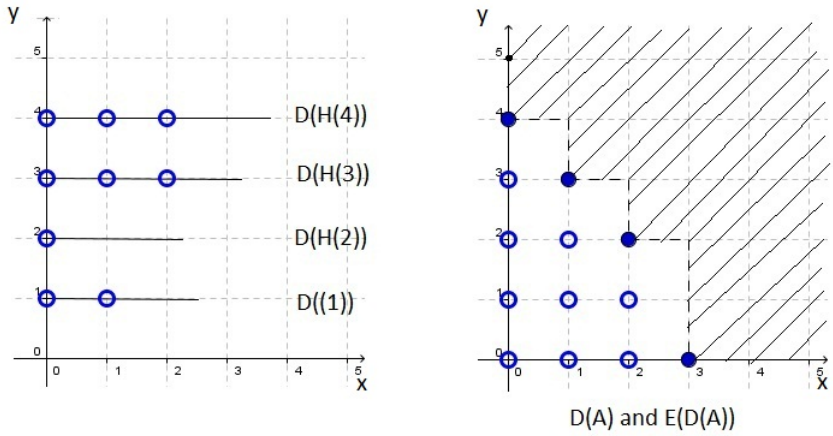


Figure 3.7: Constructing the associated lower set $D(A)$.

$k[y] \cap \mathbf{I}(A)$. In this case, we always have $\widehat{p}(\beta) = \beta_1 = 0$. So that

$$S(\beta) = \{y \in p(A) \,|\, \beta_1 = 0 \in D(H(y))\} = \{1, 2, 3, 4\} = p(A)$$
$$T(\beta) = p(A) - S(\beta) = \emptyset$$

Thus $\phi_{(0,4)}$ is a polynomial in only variable y that vanishes at all points in $p(A)$, i.e.

$$\phi_{(0,4)} = (y-1)(y-2)(y-3)(y-4) = y^4 - 10y^3 + 35y^2 - 50y + 24$$

In the case $\beta = (1, 3)$, then $\widehat{p}(\beta) = \beta_1 = 1$. In this case, $p(A)$ is spitted into two parts:

$$S(\beta) = \{y \in p(A) \,|\, \beta_1 = 1 \in D(H(y))\} = \{1, 3, 4\}$$
$$T(\beta) = p(A) - S(\beta) = \{2\}$$

To kill all points in A whose second coordinates are in $S(\beta)$, we use the product

$$\pi_{(1,3)} = (y-1)(y-3)(y-4)$$

Notice that the leading term of the polynomial $\phi_{(1,3)}$ which we want to find is xy^3. And the $\phi_{(1,3)}$ will be found in the form $\phi_{(1,3)} = \pi_{(1,3)}.\theta_{(1,3)}$ for some $\theta_{(1,3)} \in k[x, y]$ which we need to determine. Then the leading term of $\theta_{(1,3)}$ must be x. Furthermore, $\theta_{(1,3)}$ must kill the point $(4, 2) \in A$ whose second coordinate is in $T(\beta)$. So that $\theta_{(1,3)} = x - 4$. And then

$$\phi_\beta = (y-1)(y-3)(y-4)(x-4) = y^3x - 4y^3 - 8y^2x + 32y^2 + 19yx - 76y - 12x + 48$$

In the case $\beta = (2, 2)$, then $\widehat{p}(\beta) = \beta_1 = 2$. The leading term of the $\phi_{(2,2)}$ is x^2y^2. In order to do that, we split the set $p(A)$ into two parts:

$$S(\beta) = \{y \in p(A) \,|\, \beta_1 = 2 \in D(H(y))\} = \{3, 4\}$$
$$T(\beta) = p(A) - S(\beta) = \{1, 2\}$$

The sets $S(\beta)$ and $T(\beta)$ split A into two discrete parts, that are $H(3) \cup H(4)$ and $H(1) \cup H(2)$ whose projection to the second coordinate are in $S(\beta)$ and $T(\beta)$ respectively.

To kill all points in $H(3) \cup H(4)$, we use the product $\pi_{(2,2)} = (y-3)(y-4)$ whose leading term is y^2. So that we only need to find the polynomial $\theta_{(2,2)}$ that vanishes at all points of $H(1) \cup H(2)$ and whose leading term is x^2. In order to construct $\theta_{(2,2)}$, we first find polynomials in only variable x that vanish at $H(1)$ and $H(2)$ respectively, and furthermore, their leading terms are also x^2. Notice that we can identify $H(1)$ and $H(2)$ as subsets of $\mathbb{N}$ by forgetting the second coordinates. The corresponding polynomials are:

$$\begin{aligned} f_{2,1} &= (x-1)(x-3) = x^2 - 4x + 3 \\ f_{2,2} &= x^2 - [x^2 \mod (x-4)] = x^2 - 16 \end{aligned}$$

In general, the polynomial $f_{\widehat{p}(\beta),a_d} \in k[x_1, ..., x_{d-1}]$, for each $\beta \in E(D(A))$, must satisfy the inductive condition (i)-(iii). So that it is reduced by the ideal of $k[x_1, ..., x_{d-1}]$ that interpolates $H(a_d)$. In our case, the set $H(2) = \{(4, 2)\}$ is considered as a subset of $\mathbb{N}$ by forgetting the second coordinate. So that the interpolating ideal of $H(2)$ is a principal ideal of $k[x]$ that is generated by $(x-4)$. It is the reason why the polynomial $f_{2,2}$ must be reduced modulo $(x-4)$. Now by using the characteristic polynomials of $T(\beta) = \{1, 2\}$, we can construct the $\theta_{(2,2)}$ from $f_{2,1}$ and $f_{2,2}$ as

$$\theta_{(2,2)} = x^2 + \frac{y-2}{1-2}(-4x+3) - 16.\frac{y-1}{2-1} = x^2 + (y-2)(4x-3) - 16(y-1)$$

And then

$$\begin{aligned} \phi_{(2,2)} = \pi_{(2,2)}.\theta_{(2,2)} &= (y-3)(y-4)\left[x^2 + (y-2)(4x-3) - 16(y-1)\right] = \\ &= y^2x^2 + 4y^3x - 19y^3 - 36y^2x + 155y^2 - 7yx^2 + 104yx - 382y + 12x^2 - 96x + 264. \end{aligned}$$

The case $\beta = (3, 0)$ is completely the same to the previous case. We have $\widehat{p}(\beta) = \beta_1 = 3$. So that

$$\begin{aligned} S(\beta) &= \emptyset \\ T(\beta) &= \{1, 2, 3, 4\} \end{aligned}$$

And then $\pi_{(0,3)} = 1$. To determine $\theta_{(3,0)}$, we first find the polynomials in only variable x whose leading terms are x^3, and they vanish at $H(1), H(2), H(3), H(4)$ respectively. They are

$$\begin{aligned} f_{3,1} &= x^3 - [x^3 \mod (x-1)(x-3)] = x^3 - 13x + 12 \\ f_{3,2} &= x^3 - [x^3 \mod (x-4)] = x^3 - 64 \\ f_{3,3} &= (x-1)(x-2)(x-4) = x^3 - 7x^2 + 12x - 8 \\ f_{3,4} &= (x-2)(x-4)(x-5) = x^3 - 11x^2 + 38x - 40 \end{aligned}$$

Next the character polynomials of $T(\beta) = \{1, 2, 3, 4\}$ can be used to determine $\theta_{(3,0)}$ as

$$\begin{aligned} \theta_{(3,0)} &= x^3 + \tfrac{(y-2)(y-3)(y-4)}{(1-2)(1-3)(1-4)}(-13x+12) + \tfrac{(y-1)(y-3)(y-4)}{(2-1)(2-3)(2-4)}(-64) + \\ &\quad + \tfrac{(y-1)(y-2)(y-4)}{(3-1)(3-2)(3-4)}(-7x^2+12x-8) + \tfrac{(y-1)(y-2)(y-3)}{(4-1)(4-2)(4-3)}(-11x^2+38x-40) = \\ &= 440 + 286y^2 + \tfrac{5}{2}y^3x - \tfrac{31}{2}y^2x + 42yx - \tfrac{27}{2}y^2x^2 + \\ &\quad + \tfrac{173}{6}yx^2 + \tfrac{5}{3}y^3x^2 - 42x - \tfrac{2032}{3}y + x^3 - 17x^2 - \tfrac{110}{3}y^3. \end{aligned}$$

And then $\phi_{(3,0)} = \pi_{(3,0)}.\theta_{(3,0)} = \theta_{(3,0)}$ completes the minimal Gröbner basis $\{\phi_{(0,4)}, \phi_{(1,3)}, \phi_{(2,2)}, \phi_{(3,0)}\}$ of $\mathbf{I}(A)$.

Next we will show how the reduced Gröbner basis of $\mathfrak{a}$ can be constructed inductively by using the minimal Gröbner basis $\{\phi_\beta \,|\, \beta \in E(D(A))\}$. Of course, one can reduce the

basis $\{\phi_\beta \,|\, \beta \in E(D(A))\}$ by using the Multivariate Division Algorithm as in the proof of theorem 13. However, Multivariate Division Algorithm takes a very bad complexity in general. Without using the Multivariate Division Algorithm, the algorithm we will present here can be used to compute for any $\alpha \in \mathbb{N}^d - D(A)$ a polynomial $g_\alpha \in k[x_1, ..., x_d]$ such that:

(i) $lt(g_\alpha) = X^\alpha$.

(ii) g_α is reduced modulo $\mathfrak{a}$, i.e. the exponents of non-leading terms of g_α lie in $D(A)$.

Before giving the main algorithm in this section, we recall the example 12.

Example 13. Let the interpolation set $A \subset \mathbb{N}^d$ as in the example 12. Recall the minimal Gröbner basis $\{\phi_{(0,4)}, \phi_{(1,3)}, \phi_{(2,2)}, \phi_{(3,0)}\}$ of the interpolating ideal, where

$$\begin{aligned}
\phi_{(0,4)} &= y^4 - 10y^3 + 35y^2 - 50y + 24 \\
\phi_{(1,3)} &= y^3x - 4y^3 - 8y^2x + 32y^2 + 19yx - 76y - 12x + 48 \\
\phi_{(2,2)} &= y^2x^2 + 4y^3x - 19y^3 - 36y^2x + 155y^2 - 7yx^2 + 104yx - 382y + 12x^2 - 96x + 264 \\
\phi_{(3,0)} &= 440 + 286y^2 + \tfrac{5}{2}y^3x - \tfrac{31}{2}y^2x + 42yx - \tfrac{27}{2}y^2x^2 + \tfrac{173}{6}yx^2 + \tfrac{5}{3}y^3x^2 - 42x - \\
&\quad - \tfrac{2032}{3}y + x^3 - 17x^2 - \tfrac{110}{3}y^3
\end{aligned}$$

Our aim is to construct the reduced Gröbner basis $G = \{g_\alpha \,|\, \alpha \in E(D(A))\}$.

Let us see the non-leading terms of the polynomials $\phi_{(0,4)}, \phi_{(1,3)}, \phi_{(2,2)}, \phi_{(3,0)}$. Their exponents lie in the rectangle $Q := \{0,1,2,3\} \times \{0,1,2\}$ which is the smallest rectangle containing $D(A)$. So that, we must find for any $\alpha \in Q - D(A)$ the reduced polynomial g_α whose leading term is X^α. And then they can be used to kill all non-leading terms of ϕ_β, $\beta \in E(D(A))$ whose exponents do not lie in $D(A)$. It can be done as follow:

Step 1: The exponents of all non-leading terms of $\phi_{(0,4)}, \phi_{(1,3)}$ are in $D(A)$. So that $\phi_{(0,4)}, \phi_{(1,3)}$ are reduced. We set

$$\begin{aligned}
g_{(0,4)} &:= \phi_{(0,4)} \\
g_{(1,3)} &:= \phi_{(1,3)}
\end{aligned}$$

Step 2: For $\alpha = (2,2)$, the polynomials $\phi_{(2,2)}$ has a non-leading term $4xy^3$ whose exponent does not lie in $D(A)$. It can be reduced by setting

$$\begin{aligned}
g_{(2,2)} &:= \phi_{(2,2)} - 4\phi_{(1,3)} = \\
&= y^2x^2 - 3y^3 - 4y^2x + 27y^2 - 7yx^2 + 28yx - 78y + 12x^2 - 48x + 72
\end{aligned}$$

Step 3: For $\alpha = (2,3)$, consider the polynomial

$$yg_{(2,2)} = y^3x^2 - 3y^4 - 4y^3x + 27y^3 - 7y^2x^2 + 28y^2x - 78y^2 + 12yx^2 - 48yx + 72y$$

The leading term of $yg_{(2,2)}$ is x^2y^3, but it is not reduced. Because the exponents of non-leading terms $-7x^2y^2, -4y^3x, -3y^4$ are not in $D(A)$. One can kill them by setting

$$\begin{aligned}
g_{(2,3)} &= yg_{(2,2)} + 7g_{(2,2)} + 4g_{(1,3)} + 3g_{(0,4)} = \\
&= y^3x^2 - 40y^3 - 32y^2x + 344y^2 - 37yx^2 + 224yx - 928y + 84x^2 - 384x + 768
\end{aligned}$$

Step 4: For $\alpha = (3,0)$, to reduce $\phi_{(0,3)}$, we set

$$\begin{aligned} g_{(3,0)} &= \phi_{(3,0)} - \tfrac{5}{2}g_{(1,3)} + \tfrac{27}{2}g_{(2,2)} - \tfrac{5}{3}g_{(2,3)} = \\ &= 12 - \tfrac{17}{6}y^2 + \tfrac{23}{6}y^2x - \tfrac{5}{6}yx - 4yx^2 - 20x + \tfrac{19}{3}y + x^3 + 5x^2 - \tfrac{1}{2}y^3. \end{aligned}$$

Finally the polynomials $g_{(0,4)}, g_{(1,3)}, g_{(2,2)}, g_{(3,0)}$ indeed forms the reduced Gröbner basis for the interpolating ideal $\mathfrak{a} = \mathbf{I}(A)$.

Theorem 19. *Let $A \subset \mathbb{N}^d$ as before. For any $\alpha \in \mathbb{N}^d - D(A)$, there exist a unique polynomial $g_\alpha \in k[x_1, ..., x_d]$ such that:*

(i) $lt(g_\alpha) = \alpha$.

(ii) g_α kills all points of A.

(iii) The exponents of all non-leading terms of g_α lie in $D(A)$.

Proof: The uniqueness of g_α comes from the unique linear representation of X^α via the k-basis $D(A)$ of the quotient $k[x_1, ..., x_d]/\mathfrak{a}$. The existence of g_α can be implied directly from the dividing X^α by the reduced Gröbner basis. But it is not useful for our aim. Here, the existence can be proved constructively, so that it can be formulated into an efficient algorithm.

We construct g_α by induction on $\alpha \in \mathbb{N}^d - D(A)$, and using the available minimal Gröbner basis $\{\phi_\beta \,|\, \beta \in E(D(A))\}$ that is presented in the previous pages.

If α is the smallest element of $\mathbb{N}^d - D(A)$, then it is also the smallest element of $E(D(A))$. So that the polynomial ϕ_α is available. Setting $g_\alpha := \phi_\alpha$. Then g_α clearly satisfies (i)-(iii). Now let any $\alpha \in \mathbb{N}^d - D(A)$ that is not the smallest element. We assume for any $\beta \in \mathbb{N}^d - D(A)$, $\beta < \alpha$, the reduced polynomial g_β that satisfies the hypothesis is available. Two cases can arise: either $\alpha \in E(D(A))$ or $\alpha \notin E(D(A))$.

If $\alpha \in E(D(A))$, then g_α can be obtained from the available polynomial ϕ_α by killing all non-leading terms of ϕ_α whose exponents do not belong to $D(A)$, namely

$$g_\alpha := \phi_\alpha - \sum_{\gamma \in Ext(\phi_\alpha), \gamma \notin D(A)} c_\gamma g_\gamma$$

where c_γ is the coefficient of X^γ in ϕ_α. Clearly g_α satisfies the hypothesis.

In the case $\alpha \notin E(D(A))$, there exists an $i = 1, ..., d$ such that $\alpha - e_i \in \mathbb{N}^d - D(A)$, where $e_i = (0, ..., 0, 1, 0, ..., 0)$ is the unit vector of $\mathbb{N}^d$ with the value 1 at the i-th coordinate. By the inductive assumption, there exists a polynomial $g_{\alpha - e_i} \in k[x_1, ..., x_d]$ satisfying (i)-(iii), namely

$$g_{\alpha - e_i} = X^{\alpha - e_i} + \sum_{\gamma \in D(A)} c_\gamma X^\gamma$$

where $c_\gamma \in k$. Notice that the exponents of all non-leading terms of $g_{\alpha-e_i}$ must be of course smaller than $\alpha - e_i$. So that the only coefficients c_γ with $\gamma < \alpha - e_i$ are non-zero. The $g_{\alpha-e_i}$ can be rewriting as:

$$g_{\alpha-e_i} = X^{\alpha-e_i} + \sum_{\gamma\in D(A), \gamma<\alpha-e_i} c_\gamma X^\gamma$$

Then

$$x_i g_{\alpha-e_i} = X^{\alpha} + \sum_{\gamma\in D(A), \gamma<\alpha-e_i} c_\gamma X^{\gamma+e_i}$$

The exponent of leading term of $x_i g_{\alpha-e_i}$ is exactly X^α. But maybe there are some exponents of its non-leading terms that do not lie in $D(A)$. All these exponents are in the set:

$$\Gamma := \{\gamma + e_i \,|\, \gamma \text{ is the exponent of a non-leading term of } g_{\alpha-e_i} \text{ and } \gamma + e_i \notin D(A)\}$$

For every $\gamma + e_i \in \Gamma$, we have $\gamma + e_i < (\alpha - e_i) + e_i = \alpha$. So that the inductive assumption also holds for every elements of Γ. Now setting:

$$g_\alpha := x_i g_{\alpha-e_i} - \sum_{\gamma\in\Gamma} c_\gamma g_\gamma$$

where c_γ is the coefficient of X^γ in $g_{\alpha-e_i}$. Clearly g_α satisfies (i)-(iii).

Hence, the theorem is proved. □

Thanks to the proof of theorem 19, an algorithm for finding the reduced the Gröbner basis for interpolating ideal is progressively clear. The only one more to thing is that how large area of $\mathbb{N}^d$ we should do. In fact, in the proof, we assume for each $\alpha \in \mathbb{N}^d - D(A)$ the hypothesis holds for all $\beta \in \mathbb{N}^d - D(A)$ such that $\beta < \alpha$. But the number of such β is infinite. And we clearly do not need to compute g_β for all of such β. Our aim is only to compute the reduced Gröbner basis inductively and by using the ϕ_β, $\beta \in E(D(A))$ whose exponents of the non-leading terms are lie in the box:

$$Q := \prod_{i=1}^{d} \{0, 1, ..., |p_i(A)| + 1\} \subset \mathbb{N}^d$$

All polynomials that appear in the proof of theorem 19 have the exponents of their non-leading terms in Γ. Clearly $\Gamma \subseteq Q$. Hence, the box Q is large enough for us.

Putting together all things, we obtain the main algorithm of this section in the next page.

Theorem 20. *Algorithm 10 compute correctly the reduced Gröbner basis G for the interpolating ideal $\mathfrak{a} = \boldsymbol{I}(A)$.*

Algorithm 10 Find the reduced Gröbner with respect to $<_{lex}$

Require: The interpolating set $A \subset \mathbb{N}^d$.

Ensure: The reduced Gröbner basis G of interpolating ideal.

Using algorithm 9 to compute $D(A)$ ▷ Preparation step:

Compute $E(D(A))$.

$\boldsymbol{Dim} := d$

$p := \max\limits_{1 \le i \le d} |p_i(A)|$.

$Q := \prod\limits_{i=1}^{d} \{0, 1, ..., |p_i(A)| + 1\} \subset \mathbb{N}^d$.

$\boldsymbol{Data} := \emptyset$.

$G := \emptyset$.

if $\boldsymbol{Dim} = 1$ **then** ▷ Univariate case:

  $g_{|A|} := \prod_{a \in A} (x_1 - a)$.

  $\boldsymbol{Data} := \boldsymbol{Data} \cup \{g_{|A|}\}$.

  $G := G \cup \{g_{|A|}\}$.

  for $\beta = |A| + 1 \to p$ **do**

    $g_\beta := x_1 g_{\beta - 1} - c_{|A|-1} g_{|A|}$ where $c_{|A|-1}$ is the coefficient of $x_1^{|A|-1}$ in $g_{|A|}$.

    $\boldsymbol{Data} := \boldsymbol{Data} \cup \{g_\beta\}$.

  end for

  Return G.

end if

if $\boldsymbol{Dim} > 1$ **then** ▷ Induction step:

  Compute $p_d(A)$.

  for all $a_d \in p_d(A)$ **do**

    Compute $H(a_d) := p_d^{-1}(A) \cap A$.

    Using algorithm 9 to compute $D(H(a_d))$.

    $\boldsymbol{Dim} := \boldsymbol{Dim} - 1$.

    $Q(a_d) := Q \cap p_d^{-1}(a_d)$.

    Call the algorithm recursively to compute

$$P_{a_d} := \{g_{\gamma, a_d} \,|\, \gamma \in Q(a_d) - D(H(a_d))\}$$

    $\boldsymbol{Data} := \boldsymbol{Data} \cup P_{a_d}$.

  end for

end if

for all $\beta \in E(D(A))$ **do** ▷ Compute ϕ_β, $\beta \in E(D(A))$

  Compute $S(\beta) := \{a_d \in p(A) \,|\, \widehat{p}(\beta) \in D(H(a_d))\}$.

  Compute $T(\beta) := p(A) - S(\beta)$.

  Compute $\chi(T(\beta), a_d) := \prod\limits_{b_d \in T(\beta),\, b_d \neq a_d} \frac{x_d - b_d}{a_d - b_d}$ for all $a_d \in T(\beta)$.

  Compute $\pi_\beta := \prod\limits_{a_d \in S(\beta)} (x_d - a_d)$.

  Compute $\theta_\beta := \widehat{p}(X)^{\widehat{p}(\beta)} + \sum\limits_{a_d \in T(\beta)} \chi(T(\beta), a_d)(g_{\widehat{p}(\beta), a_d} - \widehat{p}(X)^{\widehat{p}(\beta)})$, where $g_{\widehat{p}(\beta), a_d}$ is available in $\boldsymbol{Data}$.

  Compute $\phi_\beta := \pi_\beta \theta_\beta$.

end for

37: **for all** $\alpha \in Q - D(A)$ **do** ▷ Reducing ϕ_β, $\beta \in E(D(A))$
38: **if** $\alpha \in E(D(A))$ **then**
39: **if** $\alpha = \min_{<_{lex}} E(D(A))$ **then**
40: $g_\alpha := \phi_\alpha$
41: **else**
42: Compute $\Gamma := \{\gamma \in Q - D(A) \,|\, \gamma < \alpha\}$.
43: Call **for** round from line 37 recursively to compute g_γ for all $\gamma \in \Gamma$.
44: $g_\alpha := \phi_\alpha - \sum_{\gamma \in \Gamma} c_\gamma g_\gamma$, where c_γ is the coefficient of X^γ in ϕ_α.
45: $\boldsymbol{Data} := \boldsymbol{Data} \cup \{g_\alpha\}$.
46: **end if**
47: $G := G \cup \{g_\alpha\}$.
48: **end if**
49: **if** $\alpha \notin E(D(A))$ **then**
50: Choose an $i = 1, ..., d$ such that $\alpha - e_i \in Q - D(A)$.
51: Call **for** round from line 37 recursively to compute $g_{\alpha - e_i}$.
52: Compute $\Gamma := \{\gamma + e_i \,|\, \gamma \in D(A),\ \gamma + e_i \notin D(A),\ \gamma < \alpha - e_i\}$.
53: Call **for** round from line 37 recursively to compute g_γ for all $\gamma \in \Gamma$.
54: $g_\alpha := x_i g_{\alpha - e_i} - \sum_{\gamma \in \Gamma} c_\gamma g_\gamma$, where c_γ is the coefficient of X^γ in $g_{\gamma - e_i}$.
55: $\boldsymbol{Data} := \boldsymbol{Data} \cup \{g_{\alpha - e_i}\}$.
56: **end if**
57: **end for**
58: Return G.

3.1.4 Changing monomial orders in Gröbner bases

The previous subsection presents an algorithm for finding the reduced Gröbner basis of the interpolating ideal with respect to the lexicographic order. However the lexicographic order can not control the total degree of the reduced polynomials. So that the result after reducing some bad interpolating polynomials may not have the smallest total degree as requested for the Interpolation Problem. In 1993, J.C.Faugere, P.Gianni, D.Lazard and T.Mora presented in [8] an efficient algorithm for changing Gröbner basis of zero dimensional ideal from any given monomial order to any other one. So that one can determine the interpolating ideal in the graded order from the lexicographic one. And then it can be used to compute the interpolating polynomial with the smallest possible total degree.

As before, we consider the set $A \subset \mathbb{N}^d$ and call A the set of interpolation points, and let $\mathfrak{a} \subseteq k[x_1, ..., x_d]$ its interpolating ideal. We assume that there are two different monomial orders in $k[x_1, ..., x_d]$, denoted by $<$ and $<_{new}$. We also assume that the reduced Gröbner basis G, the associated lower set $D(A)$, and the limit set $E(D(A))$ with respect to the monomial order $<$ are already known. Our aim is to find the reduced Gröbner basis G_{new} of ideal $\mathfrak{a}$ with respect to the new monomial order $<_{new}$.

Before describing how the algorithm works, we need a following notation.

Definition 9. *Let D be a lower set of $\mathbb{N}^d$, we denote*

$$M(D) := \{x_i m \,|\, m \in D, 1 \leq i \leq d, \text{ such that } x_i m \notin D\}$$

and call it the bordering set of D.

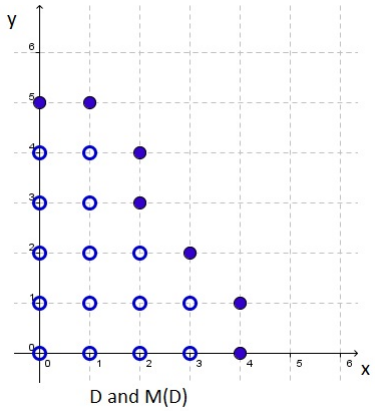


Figure 3.8: The set of solid points is a bordering set.

Now we define sequences $\{D_n\}_n, \{E_n\}_n, \{G_n\}_n, \{M_n\}_n$ inductively, that will lead us to (respectively) the associated lower set $D_{new}(A)$, the limit set $E(D_{new}(A))$, the reduced Gröbner basis G_{new}, and the bordering set $M(D_{new}(A))$ with respect to the new monomial order $<_{new}$, as follow:

For $n = 0$, we set $D_0 = \{1\}, E_0 = \emptyset, G_0 = \emptyset, M_0 = \emptyset$.

Assume we have known $\{D_n\}_n, \{E_n\}_n, \{G_n\}_n, \{M_n\}_n$ for some $n \geq 0$. Look at the bordering set of D_n,

$$M(D_n) = \{x_i m \,|\, m \in D_n, 1 \leq i \leq d, \text{ such that } x_i m \notin D_n\}$$

If $M(D_n) = M_n$, then we stop the process with $\{D_n\}_n, \{E_n\}_n, \{G_n\}_n, \{M_n\}_n$. In the next lemma, we will show that it is a sufficient condition for having the equality $D_n = D_{new}$.

Otherwise, let m be the smallest element of $M(D_n) - M_n$ with respect to the new monomial order $<_{new}$. Denote $\overline{m}$ the remainder of dividing m by G. Three cases can arise:

Case 1: If $\overline{m}$ is independent with the elements of D_n, then m has to be inserted to D_n, i.e.

$$D_{n+1} := D_n \cup \{m\}$$
$$E_{n+1} := E_n, G_{n+1} := G_n, M_{n+1} := M_n$$

Case 2: If $\overline{m}$ is a linear combination of elements in D_n and m is a strictly multiple of an element in E_n, then m has to be inserted to M_n, i.e.

$$D_{n+1} := D_n, E_{n+1} := E_n, G_{n+1} := G_n$$
$$M_{n+1} := M_n \cup \{m\}$$

Case 3: If m is not in the case 1 and 2, i.e. m is not a strictly multiple of any elements in E_n, and $\overline{m}$ a linear combination of elements in D_n, such as

$$\overline{m} = \sum_{a \in D_n} \lambda_a a, \quad \lambda_a \in k$$

then we set

$$E_{n+1} := E_n \cup \{m\}, G_{n+1} := G_n \cup \{m - \textstyle\sum_{a \in D_n} \lambda_a a\}$$
$$D_{n+1} := D_n, M_{n+1} := M_n$$

Lemma 9. *(i) Every D_n is contained in $D_{new}(A)$, the associated set of A with respect to $<_{new}$. So that the sequences $\{D_n\}_n, \{E_n\}_n, \{G_n\}_n, \{M_n\}_n$ have only finite elements. In the other words, the construction will stop after finite steps.*

(ii) Assume the construction stops at step N, then G_N is the reduced Gröbner for $\mathfrak{a}$ with respect to $<_{new}$, D_N the associated lower set of A, E_N the limit set of $D_{new}(A)$, and M_N the bordering set of $D_{new}(A)$.

Proof: We will prove the inclusion $D_n \subseteq D_{new}(A)$ by induction on n. Clearly $D_0 = \{1\} \subseteq D_{new}(A)$. To pass from n to $n+1$ we assume $D_n \subseteq D_{new}(A)$ and see what happen in the next step.

Recall the bordering set of D_n:

$$M(D_n) = \{x_i m \,|\, m \in D_n, 1 \leq i \leq d, \text{ such that } x_i m \notin D_n\}$$

If $M(D_n) = M_n$, then there is nothing to do. Otherwise, let $m = min_{<_{new}}(M(D_n) - M_n)$. Since $D_n \subseteq D_{new}(A)$, we have

$$M(D_n) \subseteq D_{new}(A) \cup M(D_{new}(A))$$

So that $m \in D_{new}(A) \cup M(D_{new}(A))$.

If $m \in D_{new}(A)$, then $D_n \cup \{m\} \subseteq D_{new}(A)$. So that m and the elements of D_n are linear independent in $k[x_1, ..., x_d]/\mathfrak{a}$. The linear independence does not depend on the way to choose k-basis for $k[x_1, ..., x_d]/\mathfrak{a}$. So that by reducing modulo the Gröbner basis G, $\overline{m}$ is independent with the elements of D_n. We are in the case 1 of the construction, so that $D_{n+1} = D_n \cup \{m\} \subseteq D_{new}(A)$

Otherwise, if $m \in M(D_{new}(A))$, then $\overline{m}$ is a linear combination of elements in $D_{new}(A)$ that are smaller than m with respect to the order $<_{new}$. On the other words, $\overline{m}$ is a linear combination of elements of D_n since the minimality of m. Thus it leads us to the case 2 or 3 in which the set D_n does not change.

Finally, we always have $D_n \subseteq D_{new}(A)$. Hence the construction stops after finite steps.

Assume the construction stop at the step N. Then $M_N = M(D_N)$ and $D_N = D_{new}(A)$. So that E_N is the limits set of $D_{new}(A)$, and G_N is a minimal Gröbner basis for $\mathfrak{a}$ with respect to $<_{new}$. On the other hand, all the non-leading terms of elements in G_N are in

$D_{new}(A)$, so that they do not divide any elements in E_N. Hence, G_N is exactly the reduced Gröbner basis for $\mathfrak{a}$ with respect to the new monomial order $<_{new}$. $\square$

The lemma 9 asserts that the construction of $\{D_n\}_n, \{E_n\}_n, \{G_n\}_n, \{M_n\}_n$ certainly lead us to the reduced Gröbner basis with the new monomial order $<_{new}$. To become an algorithm, we must solve efficiently the following problems:

Problem 6. *Compute the* $\overline{m} = m \mod G$ *where* m *is defined on each step.*

In fact we can use directly the Multivariate Division Algorithm. But there is an extremely better way by using linear algebra that we will describe below.

Problem 7. *Check the linearly independence of* $\overline{m}$ *with the elements of* D_n*, and find a linear combination if the answer is no.*

In order to do that, we will translate everything to the language of coordinates. Fix the k-basis $D(A)$ of $k[x_1, ..., x_d]/\mathfrak{a}$. For each monomial $m \in k[x_1, ..., x_d]$, the coordinates of $\overline{m}$, that is the image of m in $k[x_1, ..., x_d]/\mathfrak{a}$, can be computed naturally by reduction modulo G. If the result, for example, is the linear combination $\overline{m} = \sum_{a \in D(A)} \lambda_a a$, then the coefficients $(\lambda_a)_{a \in D(A)}$ is the coordinates of $\overline{m}$. This can be formulated by the isomorphism of k-linear spaces:

$$k[x_1, ..., x_d]/\mathfrak{a} \to k^{D(A)}, \quad f + \mathfrak{a} \mapsto \text{ coefficients of } (f \mod G)$$

Now we will give an answer for problem 6. Assume at the step n, we have already known $\{D_n\}_n, \{E_n\}_n, \{G_n\}_n, \{M_n\}_n$ and the coordinates of all elements of D_n. Assume also in the next step, $M(D_n) - M_n \neq \emptyset$. Let m be the smallest element of $M(D_n) - M_n$ with respect to $<_{new}$. Then $m = x_i m'$ for some $i = 1, ..., d$ and $m' \in D_n$ whose coordinates have already known, namely $(\lambda_a)_{a \in D(A)}$. Then

$$\overline{m} = \overline{x_i m'} = \sum_{a \in D(A)} \lambda_a \overline{x_i a}$$

The final equality requires the coordinates of all elements that are of the form $x_i a$ for some $i = i, ..., d$ and $a \in D(A)$, i.e. the elements of the set $D(A) \cup M(D(A))$. The next lemma will presents a fast way to compute all coordinates of elements in $D(A) \cup M(D(A))$. After then, we have enough to compute the coordinates of m. And finally, problem 7 is just solving a linear system.

Lemma 10. *The coordinates of all elements in* $D(A) \cup M(D(A))$ *can be computed in* $O(d.|A|^3)$ *operations on* k*.*

Proof: We divide $D(A) \cup M(D(A))$ into 3 distinct parts, such as:

$$D(A) \cup M(D(A)) = D(A) \cup E(D(A)) \cup (M(D(A)) - E(D(A)))$$

Every element m in $D(A)$ has the coordinates $(\delta_{ma})_{a \in D(A)}$, where

$$\delta_{ma} = \begin{cases} 0 & \text{if } m \neq b \\ 1 & \text{if } m = b \end{cases}$$

In the case $m \in E(D(A))$, m is the leading term of an polynomial in G, namely $g = m + \sum_{a \in D(A)} \lambda_a a$. Then the coordinates of m is $(-\lambda_a)_{a \in D(A)}$.

The difficult case is if $m \in M(D(A)) - E(D(A))$. By induction, we can assume all elements in $M(A)$ that are smaller than m with respect to $<$ have already computed. So that $m = x_i m'$ for some $i = 1, .., d$ and $m' \in M(D)$ such that $m' < m$ and its coordinates has already known. Assume $(m'_a)_{a \in D(A)}$ the coordinates of m', then

$$\overline{m'} = \sum_{a \in D(A)} m'_a a$$

Because of the definition of remainder by division algorithm, we much have $\sum_{a \in D(A)} m'_a a < m'$. So that the only coefficients in the sum $\sum_{a \in D(A)} m'_a a$ whose monomials are smaller than m' are non-zero. On the other word, $m'_a = 0$ for all $a \in D(A), a \geq m'$. So that we can rewrite:

$$\overline{m'} = \sum_{a \in D(A), a < m'} m'_a a$$

And then

$$\overline{m} = \overline{x_i m'} = \sum_{a \in D(A), a < m'} m'_a \overline{x_i a} = \sum_{a \in D(A), \overline{x_i a} < m} m'_a \overline{x_i a}$$

Because the inductive assumption, the coordinates of the $\overline{x_i a}$ are available, namely $\overline{x_i a} = ((x_i a)_b)_{b \in D(A)}$. Thus the coordinates of m can be computed by

$$\overline{m} = \sum_{a \in D(A), \overline{x_i a} < m} m'_a \sum_{b \in D(A)} (x_i a)_b b = \sum_{b \in D(A)} \left(\sum_{a \in D(A)} m'_a \right) (x_i a)_b b$$

i.e.

$$m_b = \left(\sum_{a \in D(A)} m'_a \right) (x_i a)_b$$

Clearly it takes $O(|A|^2)$ operations to compute the coordinates of each $m \in M(D(A)) - E(D(A))$. On the other hand, we always have $|M(D(A))| \leq d|A|$. Hence, it takes $O(d|A|^3)$ operations on k to compute coordinates of all elements in $D(A) \cup M(D(A))$. □

The correctness of the algorithm follows directly from the proof of lemma 10.

Now we have enough data to present the algorithm that compute the reduced Gröbner basis by changing ordering.

Theorem 21. *Algorithm 12 computes the reduced Gröbner basis with respect to the new monomial order in $O(d.|A|^3)$ operations on k.*

Proof: The correctness of algorithm follows directly from lemma 9.

The arithmetic cost of algorithm 12 is bounded by the cost of:

Algorithm 11 Compute coordinates of all elements in $D(A) \cup M(D(A))$

Require: The $D(A), G$.

Ensure: The set "***Data***" of coordinates of all elements in $D(A) \cup M(D(A))$.

1: $\boldsymbol{Data} := \emptyset$
2: $\boldsymbol{Delete} := D(A) \cup M(D(A))$
3: **if** $\boldsymbol{Delete} = \emptyset$ **then**
4: Return ***Data***. Stop the algorithm.
5: **else**
6: pick $m \in \boldsymbol{Delete}$.
7: **end if**
8: **if** $m \in D(A)$ **then**
9: $\boldsymbol{Data} := \boldsymbol{Data} \cup \{(\delta_{ma})_{a \in D(A)}\}$
10: **end if**
11: **if** $m \in E(D(A))$ **then**
12: Find a polynomial $g \in G$ such that $lt(g) = m$

$$g = m + \sum_{a \in D(A)} \lambda_a a$$

13: $\boldsymbol{Data} := \boldsymbol{Data} \cup \{(\lambda_a)_{a \in D(A)}\}$
14: **else**
15: Find an x_i that divides m.
16: Call the algorithm recursively to compute the coordinates $((\frac{m}{x_i})_a)_{a \in D(A)}$ of $\frac{m}{x_i}$ and the coordinates $((x_i a)_b)_{b \in D(A)}$ of $x_i a \in D(A)$ for all $a \in D(A), a < \frac{m}{x_i}$.
17: Compute $m_b = (x_i a)_b \sum_{a \in D(A), a < \frac{m}{x_i}} (\frac{m}{x_i})$
18: $\boldsymbol{Data} := \boldsymbol{Data} \cup \{(m_b)_{b \in D(A)}\}$
19: **end if**
20: $\boldsymbol{Delete} := \boldsymbol{Delete} - \{m\}$.
21: Repeat algorithm from line 3.

Algorithm 12 Changing ordering

Require: The set of interpolation points $A \subset \mathbb{N}^d$; the associated lower set $D(A)$, the reduced Gröbner base G of $\mathbf{I}(A)$ with respect to a given monomial order $<$; a new monomial order $<_{new}$.

Ensure: The reduced Gröbner basis G_{new}.

1: Recall algorithm 11 to compute ***Data***.
2: $D := \{1\}$
3: $E := \emptyset$
4: $G := \emptyset$
5: $M := \emptyset$
6: Compute $M(D) = \{x_i b \,|\, 1 \leq i \leq d, b \in D, x_i b \notin D\}$.
7: **if** $M(D) = M$ **then**
8: Return G, and stop the algorithm.
9: **else**
10: let $m = min_{<_{new}}(M(D) - M)$
11: **if** m is a strictly multiple of an element in E **then**
12: $M := M \cup \{m\}$
13: **end if**
14: **end if**
15: Find x_i that divides m. ▷ Compute the coordinates of m.
16: Recall the coordinates $((\frac{m}{x_i})_a)_{a \in D(A)}$ that are available in ***Data***.
17: Recall the coordinates $((x_i a)_b)_{b \in D(A)}$ of $x_i a$, for every $a \in D(A)$, that are also available in ***Data***.
18: Compute $m_b := \sum_{a \in D(A)} (x_i a)_b . (\frac{m}{x_i})_a$.
19: ***Data*** $:=$ ***Data*** $\cup \{(m_b)_{b \in D(A)}\}$
20: Solving the linear system

$$\sum_{u \in D} (u_a)_{a \in D(A)} x_u = (m_b)_{b \in D(A)} \tag{3.2}$$

with the variables x_u, $u \in D$.
21: **if** 3.2 has no root **then**
22: $D := D \cup \{m\}$
23: **end if**
24: **if** 3.2 has a root $(x_u)_{u \in D} = (\lambda_u)_{u \in D}$ **then**
25: $E := E \cup \{m\}$
26: $G := G \cup \{m - \sum_{u \in D} \lambda_u u\}$
27: **end if**
28: Repeat algorithm from line 6.

(i) Compute the set ***Data*** of coordinates of all elements of $D(A) \cup M(D(A))$, and

(ii) Compute the coordinates of each $m \in M(D_{new}(A))$, and

(iii) Solving the linear system 3.2 for each $m \in D_{new}(A) \cup E(D_{new}(A))$.

Thanks to the lemma 10, the set ***Data*** is computed in $O(d|A|^3)$ operations on k. Similarly, computing the coordinates of all $m \in M(D_{new}(A))$ takes more $O(d|A|^3)$ because it takes $O(|A|^2)$ operations on k to compute the coordinates for each m. In the whole, solving the systems 3.2 for all $m \in D_{new}(A) \cup E(D_{new}(A))$ is equivalent to computing an inverse matrix of size $|A| \times |A|$ and multiplying an $|A| \times |A|$ matrix with $|D_{new}(A) \cup E(D_{new}(A))|$ vectors in $k^{|A|}$. Notice that $|D_{new}(A) \cup E(D_{new}(A))| \leq |D_{new}(A)| + |M(D_{new}(A))| \leq (d+1)|A|$. So that the cost of solving the systems 3.2 is $O(|A|^3) + O((d+1)|A|.|A|^2) = O(d|A|^3)$. It concludes the theorem. □

3.2 Solving Polynomial Interpolation by reduction modulo ideals

In this section we will complete the solution of Multivariate Polynomial Problem by using the reduction method. Recalling the notion as in problem 5, we known that there is a one-to-one correspondence between the vectors of scalar values in k^{N+1} and the equivalence classes modulo the interpolating ideal in $k[x_1, ..., x_d]$. The interpolating ideal can be computed as in the previous section. Here we will find an initial representation for each equivalent class, and then reduce them to obtain the best interpolating polynomial in some sense.

3.2.1 Choosing an initial solution

The content of this subsection is solving computationally the following problem:

Problem 8 (The existence of interpolating polynomials)**.** *Fix the commutative field k, and positive integer N, d. Let $A = \{X_0, ..., X_N\}$ be the set of finite distinct points in k^d. Then for any scalar values $u_0, ..., u_N \in k$, there exists a polynomial $P \in k[x_1, ..., x_d]$ satisfying the interpolation condition:*

$$P(X_i) = u_i, \ \forall i = 0, ..., N$$

The existence of solution of problem 8 is equivalent to the surjectivity of the evaluation map:

$$eva : k[x_1, ..., x_d] \to k^{N+1}, \ f \mapsto (f(X_0), ..., (X_N))$$

So that it can be proved by using the Chinese Remainder Theorem. In fact, the ideals $(X - X_i)$, $i = 0, ..., N$, where $X := (x_1, ..., x_d)$ for shortly, are maximal in $k[x_1, ..., x_d]$. Since the points X_i are distinct, the ideal $(X - X_i)$ are pairly coprime. Then Chinese Remainder Theorem gives us an isomorphism:

$$k[x_1, ..., x_d]/\bigcap_{i=0}^{N}(X - X_i) \to \prod_{i=0}^{N} k[x_1, ..., x_d]/(X - X_i)$$
$$f \mod \bigcap_{i=0}^{N}(X - X_i) \mapsto (f \mod (X - X_i))_i$$

where the left hand side is the quotient of $k[x_1, ..., x_d]$ over the interpolating ideal $\mathfrak{a} := \mathbf{I}(A) = \bigcap_{i=0}^{N} (X - X_i)$. The vector space in the right hand side is isomorphic to k^{N+1} via the k-linear map:

$$\prod_{i=0}^{N} k[x_1, ..., x_d]/(X - X_i) \to k^{N+1}$$
$$(f \mod (X - X_i))_i \mapsto (f(X_i))_i$$

Hence the evaluation map *eva* is certainly surjective.

Unfortunately the Chinese Remainder Theorem does not give us any practical way to compute an inverse image for each vector of scalar values in k^{N+1}. However by reducing the problem into the univariate case, one can compute the inverse image in $O(dN^2)$ operations on k. It is the content of the following theorem.

Theorem 22. *For each $(u_0, ..., u_N) \in k^{N+1}$, one can compute a polynomial $P \in k[x_1, ..., x_d]$ satisfying $P(X_i) = u_i,\ \forall i = 0, ..., N$ in $O(dN^2)$ operations on k.*

Proof: The idea is reducing the problem into the univariate case. Precisely, we will find the polynomial P in the form

$$P = f(a_1x_1 + ... + a_dx_d)$$

where

(i) f is an univariate polynomial, and

(ii) $a_1, ..., a_d \in k$ will be found such that the function $y = a_1x_1 + ... + a_dx_d$ gets the distinct values at all points in A, and

(iii) Only two of $a_1, ..., a_d$ are different from zero.

We need (ii) because of the condition to apply the fast interpolation in univariate case. And (iii) is to easy computing the standard form of P whenever $f, a_1, ..., a_d$ available.

In order to do that, we first compute all differences $\mathbf{B} := \{X_i - X_j \mid i \neq j,\ i, j = 0, ..., N\}$. The cardinality of $\mathbf{B}$ is at most $\frac{1}{2}N(N+1)$, so that it takes $O(dN^2)$ operations on k. Next we choose randomly a vector $\alpha = (\alpha_1, ..., \alpha_d) \in k^d$ that satisfies (iii) and such that

$$\alpha\beta := \sum_{l=1}^{d} \alpha_l\beta_l \neq 0\ \forall \beta = (\beta_1, ..., \beta_d) \in \mathbf{B}$$

The existence of such α is clear if the cardinality of k is infinity or much larger than N. Otherwise, if the field k is finite and the number N is large enough then the such α maybe does not exist. To measure how large k is enough, we will compare the cardinality of $\mathbf{B}$ and the number of $\alpha \in k^d$ that holds (ii)-(iii). It requires the inequality:

$$\frac{1}{2}d(d-1)|k|(|k|-1) > \frac{1}{2}N(N+1)$$

that is equivalent to

$$|k| > \frac{1}{2} + \frac{2N+1}{2\sqrt{d(d-1)}} \tag{3.3}$$

Since

$$\frac{1}{2} + \frac{2N+1}{2\sqrt{d(d-1)}} < \frac{N+\frac{d}{2}}{d-1}$$

we can replace the condition (3.3) by a simpler one

$$|k| > \frac{N+\frac{d}{2}}{d-1} \tag{3.4}$$

If the condition (3.4) holds then an α that holds (ii)-(iii) always exists. Otherwise, if (3.4) does not hold, we can replace the field k by its extension of degree at least $\log_{|k|} \frac{N+\frac{d}{2}}{d-1}$. For simple reason, we can assume in our thesis that the condition (3.4) is satisfied.

Once such α is found, we denote $y = a_1x_1 + ... + a_dx_d$, and compute its values at all points of A, namely $y_i = a_1x_{1i} + ... + a_dx_{di}$. It takes $O(dN)$ more operations on k. Because of the condition of α, all such y_i are distinct. Thus we can use the Fast Interpolation Algorithm to determine a polynomials $f \in k[y]$ satisfying $f(y_i) = u_i,\ \forall i = 0, ..., N$. It takes $O(N \log N)$ more operations on k. Finally, the polynomial $P := f(a_1x_1 + ... + a_dx_d) \in k[x_1, ..., x_d]$ is what we need. It can be determined in

$$O(dN^2) + O(dN) + O(N \log N) = O(dN^2)$$

operations on k. □

Once $\alpha = (\alpha_1, ..., \alpha_d)$ whose only two of coordinates i-th and j-th are non-zero is found, then for every $r \in k - \{0\}$ the vector $r.\alpha$ also holds (ii)-(iii). So that we can replace α by $\frac{1}{\alpha_i}\alpha$. Thus factoring P in to the standard form is equivalent to factoring the powers $(x_i + \frac{\alpha_j}{\alpha_i}x_j)^l$ for all $l = 1, ..., N$, hence takes $O(N^2)$ more operations on k. Combining all these things, we get the following algorithm:

Algorithm 13 Choose an initial solution

Require: $X_0, ..., X_N \in k^d$ and $u_0, ..., u_N \in k$, such that the inequality (3.4) is satisfied.
Ensure: $P \in k[x_1, ..., x_d]$ such that $P(X_i) = u_i \ \forall i = 0, ..., N$.

1: Compute the set $\mathbf{B} := \{X_i - X_j \,|\, i \neq j\}$.
2: Choose an $\alpha := (a_1, ..., a_d) \in k^d$ that holds (iii) such that:

$$\alpha.\beta := \sum_{l=1}^{d} \alpha_l \beta_l \neq 0 \ \forall \beta = (\beta_1, ..., \beta_d) \in \mathbf{B}$$

3: Pick an i such that $\alpha_i \neq 0$, and set $\alpha := \frac{1}{\alpha_i}.\alpha$.
4: Compute $y_i := \alpha_1 x_{1i} + ... + \alpha_d x_{di}$ for all $i = 0, ..., N$.
5: Using algorithm 4 to determine polynomial $f \in k[y]$ such that $f(y_i) = u_i$ for all $i = 0, ..., N$.
6: Factor $f(\alpha_1 x_1 + ... + \alpha_d x_d)$, and return by P.

3.2.2 Choosing the best solution

The aim of this subsection is to putting together all things in chapter 3 to give an algorithm for solving the Multivariate Polynomial Interpolation problem in general case. All necessary material for finding the smallest interpolating polynomial is available in the previous sections. Before presenting the algorithm, we recall the main problem.

Problem 9 (Multivariate Polynomial Interpolation)**.** *Fix the commutative field k, and positive integer N, d. Let $N+1$ distinct interpolation points $X_0, ..., X_N \in k^d$. Find for each vector of scalar values $(u_0, ..., u_N) \in k^{N+1}$ a polynomial $P \in k[x_1, ..., x_d]$ with the smallest possible degree that satisfies the interpolation conditions:*

$$P(X_i) = u_i, \ \forall i = 0, ..., N \tag{3.5}$$

As mentioned before, any two solutions for the same vector of scalar value $u = (u_0, ..., u_N) \in k^{N+1}$ agree in the same coset of interpolating ideal $\mathfrak{a} = \mathbf{V}(\{X_0, ..., X_N\})$ in $k[x_1, ..., x_d]$. Hence the problem of finding the interpolating polynomial with the smallest possible degree is equivalent to finding the least representer for this equivalence class. It is the reason why reduction modulo ideals is the key of our method.

Algorithm 14 Finding the best solution

Require: $X_0, ..., X_N \in k^d$ and $u_0, ..., u_N \in k$.
Ensure: $P \in k[x_1, ..., x_d]$ with smallest possible degree such that $P(X_i) = u_i \ \forall i = 0, ..., N$.

1: Using algorithm 10 to compute the reduced Gröbner basis for interpolating ideal $\mathfrak{a} := \mathbf{V}(\{X_0, ..., X_N\})$ with respect to $<_{lex}$.
2: Using algorithm 12 to compute the reduced Gröbner basis G of $\mathfrak{a}$ with respect to $<_{grlex}$ by changing ordering.
3: Using algorithm 13 to compute an initial interpolating polynomial P.
4: Using algorithm 6 to reduce P by G, rename by P.
5: Return P.

3.3 Other approaches

In this section, we will give a short summary on the method we have used in this thesis within other known methods.

The algorithm for finding the reduced Gröbner basis for zero-dimensional ideal in $k[x_1, ..., x_d]$ is the core of the algorithm for solving Multivariate Polynomial Interpolation. There have been several papers referring to this issue. B. Buchberger and M. Möller presented the first algorithm for finding the reduced Gröbner basis of zero-dimensional ideals in EUROCAM 1982 [11]. Their algorithm is based on Gauss elimination on a generalized Vandemonde matrix and runs in the cubic time in the number of interpolation points and variables. It is improved significantly and extended into the projective case thanks to the work of M. G. Marinari, H.M. Möller and T. Mora in 1993 [12] and J. Abbott, A. Bigatti, M. Kreuzer and L. Robbiano in 2000 [13]. An other method which is generalized naturally from Newton's interpolation for univariate polynomials is proposed by J. Farr and S.Gao in 2006 [14]. The algorithmic cost of their algorithm is exponent in term of the number of variables. However the algorithm which we refer for the thesis is given by M. Lederer in 2008 [6]. A more completed version was written in the first chapter of his Habilition thesis in 2012 [7]. Without solving any linear system but by induction over the dimension of affine space, his method describes geometrically the set of all leading terms of polynomials in a zero-dimensional ideal.

In the algorithm 14, which is our main algorithm, the most expensive step is reducing the initial solution by the reduced Gröbner basis with respect to $<_{grlex}$ of the interpolating ideal. It is not hard to solve that the reducing step takes $O(N^{d+1})$ operations on k. So that finding a more efficient polynomial reduction is necessary to improve our algorithm. It remains an open problem.

There are several available methods that does not use polynomial reduction. One of them is proposed by P. J. Olver in 2006 [5]. His method is a generation of Vandemonde matrix in multidimensional case whose entries are in the certain block forms instead of scalar values. In order to do that, he develop a multivariate divided difference calculus based on the theory of non-commutative quasi-determinants. As the same in the univariate case, he presented an explicit block LU decomposition of multidimensional Vandemonde matrices for computing the multivariate interpolation coefficients, and hence established an analog of Newton's fundamental interpolation formula in several variables. His method has a strong connection with the theory of non-commutative symmetric functions.

An other but equivalent method for solving Multivariate Interpolation Problem is come from an improvement of Buchberger-Müller algorithm that is proposed by J. Abbott, A. Bigatti, M. Kreuzer and L. Robbiano in 2000 [13]. Based on Gauss elimination of Vandemonde matrix, they added into the origin Buchberger-Müller algorithm a new step for computing all separated polynomials. And then the interpolation polynomial is just a certain linear combination of the separated polynomials, so that can be computed efficiently.

Bibliography

[1] Joachim von zur Gathen, Jürgen Gerhard, *Modern Computer Algebra*, Cambridge University Press, July 3, 2003.

[2] D. Cox - J. Little - D. OShea, *Ideals, varieties and algorithms: An introduction to computational algebraic geometry and commutative algebra*, Springer-Verlag, second edition, 1997.

[3] D. Eisenbud, *Commutative Algebra with a view toward Algebraic Geometry*, Springer-Verlag, 1995.

[4] M. Gasca, T. Sauer, *Polynomial interpolation in several variables*, Advances in Computational Mathematics **12** (2000), 377-410.

[5] P. J. Olver, *On Multivariate Interpolation*, Studies in Applied Mathematics, **116** (2006), 201-240.

[6] M. Lederer, *The vanishing ideal of a finite set of closed points in affine space*, J. Pure Appl. Algebra **212** (2008), 1116-1133.

[7] M. Lederer, *Connect Four theory for Hilbert schemes of points*, Habilition thesis, University of Bielefeld, Germany, 2012.

[8] J.C. Faugere, P. Gianni, D. Lazard and T. Mora, *Efficient Computation of Zero-dimensional Gröbner bases by Change of Ordering*, J. Symbolic Computation **16** (1993), 329-344.

[9] W. Borchardt, *Über eine Interpolationsformel für eine Art symmetrischer Funktionen und deren Anwendung*, Abh. d. Preu. Akad. d. Wiss. (1860), 1-20.

[10] L. Kronecker, *Über einige Interpolationsformeln für ganze Funktionen mehrerer Variabeln*, Lecture at the academy of sciences, December 21, 1865. In L. Kroneckers Werke, volume I, H. Hensel edit., 133141. Teubner 1895, reprinted by Chelsea Publishing Company 1968.

[11] B. Buchberger and H. M. Möller, *The construction of multivariate polynomials with preassigned zeros*, Computer algebra, EUROCAM '82, pp. 24-31, Lecture Notes in Comput. Sci., vol. 144, Springer, Berlin-New York, 1982.

[12] M. G. Marinari, H.M. Möller and T. Mora, Gröbner bases of ideals defined by functionals with an application to ideals of projective points, Appl. Algebra Engrg. Comm. Comput. **4** (1993), 103-145.

[13] J. Abbott, A. Bigatti, M. Kreuzer and L. Robbiano, *Computing ideals of points*, J. Symbolic Comput. **30** (2000), 341-356.

[14] J. Farr and S.Gao, *Computing Gröbner bases for vanishing ideals of finite set of points*, Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Lecture Notes in Computer Science, Volume 3857 (2006), 118-127.