Master 1 CCA
CCA Project

# Study of a two-variables Gröbner basis

*Authors*
Philippe TAN
Jarod XIE

*Supervisor*
J. Berthomieu

May 23, 2024

# Contents

Abstract

In 1965, Bruno Buchberger introduced the Gröbner basis in his PhD thesis [1]. Named after his advisor, Wolfgang Gröbner, Gröbner bases have since become a fundamental tool in computer algebra, computational algebraic geometry, and computational commutative algebra. A Gröbner basis is a particular kind of generating set for polynomial ideals that allows for the algorithmic solution of systems of polynomial equations, which means that it has a wide range of applications. The construction of a Gröbner basis depends on the choice of a monomial ordering, which dictates the leading terms of the polynomials, but it also depends on the lexicographical ordering, especially in a two-variable Gröbner basis where we need to compare $x$ and $y$. In our study, we aim to understand how two-variable Gröbner bases work, how to construct one, and how they evolve when working with more than two polynomials.

# 1 Introduction

Gröbner bases are a powerful technique to solve systems of polynomials. Given a system of polynomial equations $f_1 = \ldots = f_r = 0$ where $f_1, \ldots, f_r$ are two-variable polynomials, in $x$ and $y$, with coefficients in a field $\mathbb{K}$, it is difficult to solve it in this state. However, with Buchberger's algorithm, we can write $f_1, ..., f_r$ as a set of polynomials $F$ and turn it into another set $G$ called Gröbner basis of $F$. The new polynomials have nice properties that simplify the solving of the system and admit the same set of solutions as $F$.

Nevertheless, constructing one is challenging. For example, Buchberger's algorithm (published in his August 1976 research paper [2]) has a worst-case complexity that can be double exponential in the number of variables $n$ : upper bound $D^{2^{O(n)}}$ where $D$ is the max degree of the polynomials in $F$. In contrast, state-of-the-art Gröbner bases engines such as the `F4` algorithm developed by Jean-Charles Faugère [3] in `MAPLE`, and `MAGMA`, have exponential worst-case complexity.

But in order to understand Gröbner bases, which are new to us, their applications and how to construct one, we need to start from the beginning with basic cases of systems of equations and fundamental concepts in algebra. From there, we build up to the construction of Gröbner bases in these simplified contexts, aiming to gain insights into the practical implementation and computational aspects of Gröbner bases. Unfortunately, due to our limited understanding of the subject and the slow progress we made, we were unable to reach the implementation part in `MAPLE` as originally intended.

# 2 Introducing Gröbner Bases

Gröbner bases are a new mathematical object for us, requiring a solid grasp of algebra, especially with polynomials.

## 2.1 Simplest case of interpolation

### 2.1.1 Distinct $x$-coordinates

We start with the most basic case : given $k$ pairs of coordinates with all $x_i$ **distinct** (here for the example we take $k = 3$) $A_0 = (x_0, y_0), A_1 = (x_1, y_1), A_2 = (x_2, y_2)$.

Find all the polynomials $h$ such that $h(x, y)$ cancels out at all our points $A_0, A_1$ and $A_2$ :

$$h(x_0, y_0) = 0$$
$$h(x_1, y_1) = 0$$
$$h(x_2, y_2) = 0$$

To build this polynomials $h$, it is fairly easy. We can decompose $h(x, y)$ into a combination of two different polynomials : $h_1(x)$ purely in $x$ and $h_2(y)$ purely in $y$.

As all $x$ are distinct in our pairs of coordinates, we just need :

- in case of a product $h(x, y) = h_1(x) * h_2(y)$ : one of our polynomial $h_1(x_i)$ to cancel out on all $x_i$ or $h_2(y_i)$ to cancel out on all $y_i$
- in case of a sum $h(x, y) = h_1(x) + h_2(y)$ : both polynomials need to cancel out on all $x_i$ and $y_i$

In any case, as we have one purely in $x$ and the other $y$ the obvious solution would be :

$$\begin{aligned} h_1(x) &= (\prod_{i=0}^{k-1}(x - x_i)) \cdot V_1(x) \text{ where } V_1(x) \text{ is an arbitrary polynomial purely in } x \\ &= (x - x_0)(x - x_1)(x - x_2)V_1(x) \\ h_2(y) &= \prod_{i=0}^{k-1}(y - y_i) \cdot V_2(y) \text{ where } V_2(y) \text{ is an arbitrary polynomial purely in } y \\ &= (y - y_0)(y - y_1)(y - y_2)V_2(y) \end{aligned}$$

### 2.1.2 Repetition with the same abscissa $x$

Now let's add another pair of coordinates such that we don't have all $x_i$ distinct anymore, which means that there exists at least two pairs as follow :

$$A_i = (x_i, y_i)$$
$$A'_i = (x_i, y'_i)$$

where $A_i \neq A'_i, y_i \neq y'_i$
Then the previous method would still work as $h(x, y)$ is still a combination of the two polynomials $h_1(x)$ and $h_2(y)$ that were designed to cancel out on our pairs of coordinates.

But what if we want to express $y$ as a function of $x$ ?
Say we want to try to find a polynomial $p(x)$ such that $y - p(x) = 0 \iff y = p(x)$. We need to verify if such polynomial respects these conditions :

$$y - p(x) = 0 \begin{cases} p(x_0) & = y_0 \\ & \vdots \\ p(x_k) & = y_k \end{cases}$$

As all $x_i$ are not distinct, we meet a problem :

$$\begin{cases} p(x_i) &= y_i \\ p(x_i) &= y'_i \end{cases} \Longrightarrow \Big\{ y_i = y'_i \text{ impossible since we have } y_i \neq y'_i \tag{1}$$

We face the same problem when we try with $y^2 - p(x) = 0$.
Then we try with $y^2 + ay - p(x) = 0$ ;

**Example 1. Polynomial p(x)'s existence : $x_i$ not distinct two-by-two**

Given the pairs of coordinates $\{(-1,0),(1,0),(1,1),(2,2)\}$, we have :

$$y^2 + ay - p(x) = 0 \begin{cases} y_0^2 + ay_0 - p(x_0) &= 0 \\ &\vdots \\ y_k^2 + ay_k - p(x_k) &= 0 \end{cases} \iff \begin{cases} p(-1) &= 0^2 + a0 = 0 \\ p(1) &= 0^2 + a0 = 0 \\ p(1) &= 1^2 + a1 = 1 + a \\ p(2) &= 2^2 + a2 = 4 + 2a \end{cases}$$

$$\begin{cases} p(1) &= 0 \\ p(1) &= 1 + a \end{cases} \iff \Big\{ a = -1 \quad \text{then verify for } x = 2 : \text{ we have } 4 + 2(-1) = 0$$

So we know there exist a polynomial $p(x)$ such that $y^2 + ay - p(x) = 0$ with our set of pairs of coordinates.

With this method, we reduce the problem to the solving of a smaller system of equation on the points sharing the same antecedent and having different images.

## 2.2 Matrix approach

Having studied systems of equations to solve the initial problem, in this section we'll try to look at a solution using a matrix representation and how to exploit matrices to systematically transform and simplify the system, leveraging techniques such as row reduction and echelon forms to gain insights into the structure of the solutions. Although this exploration was not very fruitful, we did manage to make progress on the initial problem.

To begin with, given k pairs of coordinates with all $x_i$ distinct (here k = 6) $A_0 = (x_0, y_0)$, $A_1 = (x_1, y_1)$, $\cdots$, $A_5 = (x_5, y_5)$.

Moreover, we know that now we're looking for a polynomial $P(x, y)$ of the form $P(x, y) = ax^2 + by^2 + cxy + ey + f$, which cancels out especially at the points $(x_i, y_i)$.

We therefore had an equation of the form :

$$\begin{cases} ax_0^2 + by_0^2 + cx_0y_0 + ey_0 + f = 0 \\ ax_1^2 + by_1^2 + cx_1y_1 + ey_1 + f = 0 \\ \vdots \\ ax_5^2 + by_5^2 + cx_5y_5 + ey_5 + f = 0 \end{cases} \tag{2}$$

Now we can construct a matrix in $\in \mathbb{K}^{k\times k}$ with each line of the equation being each line of the matrix :

$$A = \begin{bmatrix} 1 & x_0 & y_0 & x_0y_0 & x_0^2 & y_0^2 \\ 1 & x_1 & y_1 & x_1y_1 & x_1^2 & y_1^2 \\ & & \cdots & & & \\ 1 & x_5 & y_5 & x_5y_5 & x_5^2 & y_5^2 \end{bmatrix}$$

The question we were given is: Given the pairs $(x_i, y_i)$ for the 6 provided points, what are the conditions or constraints that determine whether the matrix is invertible or not invertible ?
But first we had to try with a lower k to better understand the problem. So for now, let's take $k = 2$, therefore, 2 pairs $(x_i, y_i)$ and we have a new $P(x, y)$ such that $P(x, y) = a + bx + cy$ and we want $P(x, y) \neq 0$.

The problem in terms of matrix is now :

$$\begin{bmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \end{bmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Now let's suppose either $a, b$ or $c$ are $\neq 0$, then we have,

$$\text{if c} \neq 0, \text{let c} = 1, \begin{bmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \end{bmatrix} \begin{pmatrix} a \\ b \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Leftrightarrow \begin{bmatrix} 1 & x_0 \\ 1 & x_1 \end{bmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -y_0 \\ -y_1 \end{pmatrix}$$

$$\text{if b} \neq 0, \text{let b} = 1, \begin{bmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \end{bmatrix} \begin{pmatrix} a \\ 1 \\ c \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Leftrightarrow \begin{bmatrix} 1 & y_0 \\ 1 & y_1 \end{bmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -x_0 \\ -x_1 \end{pmatrix}$$

$$\text{if a} \neq 0, \text{let a} = 1, \begin{bmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \end{bmatrix} \begin{pmatrix} 1 \\ b \\ c \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Leftrightarrow \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \end{bmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

From these simplified cases, we can derive the conditions for invertibility by examining the determinants of the matrices formed. The matrix is invertible if its determinant is non-zero, which means that the vectors formed by the coordinates must be linearly independent. So now, let's get back to k = 6 and our matrix $A$. The initial problem can be written as:

$$\begin{bmatrix} 1 & x_0 & y_0 & x_0y_0 & x_0^2 & y_0^2 \\ 1 & x_1 & y_1 & x_1y_1 & x_1^2 & y_1^2 \\ & & \cdots & & & \\ 1 & x_5 & y_5 & x_5y_5 & x_5^2 & y_5^2 \end{bmatrix} \begin{pmatrix} a \\ b \\ \vdots \\ f \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

The problem reduces to finding the conditions under which this homogeneous system has a non-trivial solution, i.e., a solution where not all coefficients $a, b, c, d, e$ and $f$ are zero. A homogeneous system $Ax = 0$ has non-trivial solutions if and only if the determinant of $A$ is zero. So, the key to solving our problem lies in determining the determinant of $A$.

But one interesting question should be, how to find the form of polynomial $P(x, y)$ when $k > 6$, this new $P(x, y)$ needs to be more complex to ensure there are enough coefficients to accommodate all the given points.

Firstly, let's try to find for $k = 7$, we need a polynomial with 7 coefficients. This can be achieved by including cubic terms in $x$ and $y$. The polynomial can be written as:

$$P(x, y) = ax^3 + by^3 + cx^2y + dxy^2 + ex^2 + fy^2 + gxy + hx + iy + j \tag{3}$$

We then have the system of equations :

$$\begin{cases} ax_0^3 + by_0^3 + cx_0^2y_0 + dx_0y_0^2 + ex_0^2 + fy_0^2 + gx_0y_0 + hx_0 + iy_0 + j = 0 \\ ax_1^3 + by_1^3 + cx_1^2y_1 + dx_1y_1^2 + ex_1^2 + fy_1^2 + gx_1y_1 + hx_1 + iy_1 + j = 0 \\ \vdots \\ ax_6^3 + by_6^3 + cx_6^2y_6 + dx_6y_6^2 + ex_6^2 + fy_6^2 + gx_6y_6 + hx_6 + iy_6 + j = 0 \end{cases} \tag{4}$$

To generalize for any $k$, we need to determine the polynomial degree $d$ that provides at least $k$ coefficients. The general form of a polynomial of degree $d$ in two variables $x$ and $y$ is:

$$P(x, y) = \sum_{1=0}^{d} \sum_{j=0}^{d-i} c_{ij} x^i y^j \tag{5}$$

The number of terms (coefficients) in this polynomial is the number of combinations with repetition:

$$\binom{d+2}{2} = \frac{(d+2)(d+1)}{2}$$

We can test for $k = 7$ if $d = 3$ is sufficient, we then have :

$$\frac{(3+2)(3+1)}{2} = 10 \geq 7$$

To conclude, the matrix approach provided a systematic way to analyze the existence of polynomials that cancel out at specified points. By extending the polynomial's complexity as the number of points increases, we can construct a suitable polynomial for any given $k$. This method not only offers a clear mathematical framework but also highlights the critical role of matrix determinants in determining the solvability of the system.

This establishes a fundamental link between linear algebra and polynomial algebra, leveraging matrices that are essential components in the algorithms for implementation purposes.

## 2.3 Lagrange's interpolation

For the following section, we are going to use Lagrange's interpolation to interpolate polynomials that cancel out on the pairs of coordinates that we have.

**Definition 1. Lagrange polynomial**

The Lagrange interpolating polynomial is the polynomial P(x) of degree <=(n-1) that passes through the $n$ points $(x_1, y_1 = f(x_1)), (x_2, y_2 = f(x_2)), ..., (x_n, y_n = f(x_n))$, and is given by

$$L(x) = \sum_{j=1}^{n} l_j(x) \tag{6}$$

where

$$l_j(x) = y_j \prod_{\substack{k=1, \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k} \tag{7}$$

**Example 2. Lagrange polynomial for $n = 3$**

Consider three points $n = 3 : \{(x_0, y_0), (x_1, y_1), (x_2, y_2)\}$
We have

$$L(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} y_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} y_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} y_2$$

with

$$\begin{cases} l_0(x) & = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} y_0 \\ l_1(x) & = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} y_1 \\ l_2(x) & = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} y_2 \end{cases}$$

Although Lagrange's method works wonderfully, we can not apply it to non distinct $x_i$ (two images for one antecedent (1)).

To address this issue, first let us assume that there is only one repetition of a $x_i$ (they come two by two), then let us denote two sub sets $D$ and $E$ :

- where $D$ contains all the pairs of coordinates ($d$ elements) such that all $x_i$ are distinct two-by-two $(x_i, y_i)$.
- where $E$ contains all the pairs of coordinates ($e$ pairs of pairs of coordinates) such that all $x_i$ are distinct two-by-two $((x_i, y_i), (x_i, y_i'))$ with $y_i \neq y_i'$ for all $i$.

Now we want to find a polynomial of this format : $G(x, y) = y^2 + g_3(x)y + g_4(x)$ which cancels out on every points that we have.

To begin with this task, we had to prove that the interpolation problem was well-defined for the subspace of $(x_i, y_i)$ pairs where $x_i$ are distinct two-by-two (for sub set D). And if we denote $y - g_2(x)$ the interpolating polynomial, then $(y - g_2) * \prod_{x_i \in E}(x - x_i)$ of course cancels out on all $(x_i, y_i)$.

**Proof 0.1. Interpolation problem**

In fact, let's take $k$ pairs $(x_i, y_i)$ belong to $D$, the interpolation problem is well-defined,

because we can let a function $g_2$ such that verified the equation for all $k$

$$g_2(x) = \begin{cases} g_2(x_0) = y_0, \text{if } x = x_0 \\ \vdots \\ g_2(x_k) = y_k, \text{if } x = x_k \end{cases}$$

So the interpolation problem is well-defined and $g_2$ is the function that associates $y$ with $x$. We then let $y - g_2$ be the interpolating polynomial, for which we have that $\forall (x_i, y_i) \in D, y_i - g_2(x_i) = 0$. Since we know that $\prod_{\text{for } x_i \in E}(x - x_i)$ cancels out at any point $x_i$.
We can conclude the proof that $(y - g_2) * \prod_{x_i \in E}(x - x_i)$ cancels out at any points $(x_i, y_i) \in D \cup E$

Now we want to try to show that finding a polynomial of the form $G(x, y) = y^2 + yg_3(x) + g_4(x)$ such that it cancels out in these pairs $((x_i, y_i), (x_i, y_i'))$ is possible, with this couple of points in $E$. To get an idea, we'll try to do this with 4 different points that we set, for example : $(0, 0), (0, 1), (1, 4), (1, -4)$. We can directly notice that there are two images for both antecedent 0 and 1. Our idea was to divide the point in 2 different subspace where all the $x_i$ we distinct two-by-two such that we have now :

$$\begin{aligned} (0,0) \quad & |h_1(0) = 0 \\ (1,4) \quad & |h_1(1) = 4 \end{aligned}$$

$$\begin{aligned} (0,1) \quad & |h_2(0) = 1 \\ (1,-4) \quad & |h_2(1) = -4 \end{aligned}$$

Lagrange interpolation can therefore be applied twice to each subspace, so now we have $h_1(x) = 3x + 1$ and $h_2(x) = -5x + 1$. Let $y - h_1(x), y - h_2(x)$ be interpolating polynomial, and we are trying to get something with the form of $G(x, y) = y^2 + g_2(x) + g_3(x)$ that cancels out for pairs from $E$, one way to do it is to reassemble those two polynomial by multiplying them together. We can compute $(y - h_1(x))(y - h_2(x)) = y^2 - y(h_1(x) + h_2(x)) + h_1(x)h_2(x)$, let's replace $h_1$ and $h_2$ by $h_1(x) = 3x + 1$ and $h_2(x) = -5x + 1$ we find that $G(x, y) = y^2 + y(2 - 2x) - 15x^2 - 2x + 1$. So we start with 4 different points and at the end we manage to write a polynomial that cancels out at those points with the form $G(x, y) = y^2 + yg_3(x) + g_4(x)$, here $g_4(x) = -15x^2 - 2x + 1$ and $g_3(x) = 2 - 2x$.

We can generalize this method for any pair of points belonging to E such that $((x_i, y_i), (x_i, y_i'))$ thanks to Lagrange interpolation that work well if you separate beforehand the points where you have the same $x_i$ for different $y_i$.

Although Lagrange interpolation offers an effective method for solving certain interpolation problems, a new perspective can be gained by applying the Chinese Remainder Theorem (CRT), which allows interpolation problems to be treated as congruence problems.

## 2.4 New perspective

We know that the Lagrange interpolation is a special case of the Chinese Remainder Theorem (CRT). This means that an interpolation problem is a congruence problem and that we can apply CRT on our system !

**Theorem 1. Chinese Remainder Theorem (CRT)**

Given pairwise coprime integers $n_1, \ldots, n_k$ and arbitrary integers $a_1, \ldots, a_k$, then the system :

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ x &\equiv a_2 \pmod{n_2} \\ &\vdots \\ x &\equiv a_k \pmod{n_k} \end{aligned} \tag{8}$$

has a solution, and the solution is unique modulo $N = n_1 n_2 \cdots n_k$ :

$$x = a_1 M_1 y_1 + \cdots + a_n M_n y_n \tag{9}$$

where $M_i = M/m_i$ and $y_i \equiv M_i^{-1} \pmod{m_i}$ with $1 \leq i \leq n$

To familiarize ourselves with the CRT, we can try to apply it to a simple example.

**Example 3. CRT on a system in $\mathbb{N}$**

Let us consider the system :

$$\begin{cases} h &= 1 \pmod 2 \\ h &= 2 \pmod 5 \\ h &= 1 \pmod 9 \end{cases} \Rightarrow M = 2 \cdot 5 \cdot 9 = 90 \begin{cases} M_1 &= 5 \cdot 9 = 45 \\ M_2 &= 2 \cdot 9 = 18 \\ M_3 &= 2 \cdot 5 = 10 \end{cases}$$

Then we have $y_i$ the modular inverse of $M_i$ modulo $m_i$ :

$$\begin{cases} y_1 &= 45 \pmod 2 = 1 \\ y_2 &= 18 \pmod 5 = 2 \\ y_3 &= 10 \pmod 9 = 1 \end{cases} \Longrightarrow \begin{cases} y_1 &= 1 \\ y_2 &= 2 \\ y_3 &= 1 \end{cases}$$

and then we have $x = \sum_{i=1}^{n} a_i M_i y_i = 1 \cdot 45 \cdot 1 + 2 \cdot 18 \cdot 2 + 1 \cdot 10 \cdot 1 = 127 \equiv 37 \pmod{90}$
So the unique solution of our system is $h \equiv 37 \pmod{90}$

But how do we proceed with polynomials ?

**Method 1. Translating a system of polynomial equations to a congruence system.**

Given a set of pairs of coordinates $C = \{(x_0, y_0), \ldots, (x_k, y_k)\}$, divide $C$ into sub sets $D$ and $E$ as described previously in 2.3. Then we can write a system of congruence :

- for each element of $D, (x_i, y_i)$ with $i \in [0, d-1]$, we have : $G \equiv y - y_i \pmod{x - x_i}$
- for each pair of $E, ((\tilde{x}_i, \tilde{y}_i), (\tilde{x}_i, \tilde{y}'_i))$ with $i \in [d, e-1]$, we have $G \equiv (y - \tilde{y}_i)(y - \tilde{y}'_i) \pmod{x - \tilde{x}_i}$

We construct these congruence equations because we want $G(x, y) = y^2 + g_3(x)y + g_4(x)$ to turn into $G = y - y_i$ or $G = (y - \tilde{y}_i)(y - \tilde{y}'_i)$ (depending on whether the element belongs to $D$ or $E$) when $x = x_i$ or $x = \tilde{x}_i$ so that it cancels out on all elements.

Then we have

$$\begin{cases} G & \equiv y - y_0 \pmod{x - x_0} \\ & \vdots \\ G & \equiv y - y_{d-1} \pmod{x - x_{d-1}} \\ G & \equiv (y - \tilde{y}_d)(y - \tilde{y}'_d) \pmod{x - \tilde{x}_d} \\ & \vdots \\ G & \equiv (y - \tilde{y}_{e-1})(y - \tilde{y}_{e-1}) \pmod{x - \tilde{x}_{e-1}} \end{cases} \tag{10}$$

From this system, we can gather the ones issued from $D$ and the ones issued from $E$

$$\begin{cases} G & \equiv y - h_3(x) \pmod{\prod_{i=0}^{d-1} x - x_i} \\ G & \equiv (y - h_1(x))(y - h_2(x)) \pmod{\prod_{i=d}^{e-1} x - \tilde{x}_i} \end{cases} \tag{11}$$

Now we have a congruence system on which we can apply the CRT and find an unique solution $G$.

Now that we have the method, we can try to solve a small example :

**Example 4**

Let us work with the set $C = \{(0, 0), (0, 1), (2, 3)\}$. We get the following system :

$$\begin{cases} G & \equiv (y - 0)(y - 1) \pmod{x - 0} \\ G & \equiv (y - 3) \pmod{x - 2} \end{cases}$$

Recall the unique solution derived from the CRT (theorem 1, (9)) :

$$G = a_1 M_1 d_1 + \cdots + a_n M_n d_n$$

With $M = x \cdot (x - 2), M_1 = x - 2$ and $M_2 = x$ and $d_i \equiv M_i^{-1} \pmod{m_i}$ the inverse of $M_i$ modulo $m_i$.

For example, let's compute $d_1 = M_1^{-1} \pmod{m_i}$ :

**1. Modular inverse $d_1$**

Let $(R_0, U_0, V_0) = (x, 1, 0)$ and $(R_1, U_1, V_1) = (x-2, 0, 1)$.
We write $q = \frac{R_0}{R_1}$ the quotient of the polynomial division of $R_0$ by $R_1$.

Then we have

$$\begin{aligned}(R_2, U_2, V_2) &= (R_0, U_0, V_0) - q(R_1, U_1, V_1)\\ &= (x, 1, 0) - 1(x-2, 0, 1)\\ &= (x - x + 2, 1, -1)\\ &= (2, 1, -1)\end{aligned}$$

This yields : $x \cdot 1 + (x-2) \cdot (-1) = 2$ where $d_1 = (-1)$.

Here we have $d_1 = -1$ and $d_2 = 1$. Then we can rewrite G :

$$\begin{aligned}G &= a_1 M_1 d_1 + \cdots + a_n M_n d_n\\ &= (y^2 - y) \cdot (x-2) \cdot (-1) + (y-3) \cdot x \cdot 1\\ &= y^2 \cdot (1 - \frac{x}{2}) + y \cdot (x-1) - \frac{3}{2} \cdot x\end{aligned}$$

Our example isn't quite right since we have a coefficient in front of our $y^2$ term, but the polynomial $G(x, y)$ does cancel out on all our pairs of coordinates.

We now have all the tools and methods to interpolate polynomials. We will now proceed to prove the existence of these polynomials within a Gröbner basis.

# 3 Proof of existence of the basis' polynomials

When the polynomial system has a finite number of solutions, the associated Gröbner basis for the lexicographic order contains at least two particular non-zero polynomials: one purely in y and one whose largest monomial is a power of x.
We're still working with $D$ and $E$ as defined in 2.3. Let's prove this in several steps.

## 3.1 First case ($f_0$, $f_1$)

We consider all pairs $(x_i, y_i)$ with the $x_i$ distinct two-by-two with no repetition (they are all elements of $D$).
Then we can write :

$$f_0(x) = \prod_{i=0}^{d-1} (x - x_i)$$

$$f_1(x, y) = y - L(x) \text{ where } L(x) \text{ is a Lagrange polynomial}$$

Now, suppose that we have a polynomial $P(x, y)$ that cancels on all pairs $(x_i, y_i)$. Then $\exists q_0, q_1$ such that :

$$P(x, y) = f_0(x) q_0(x, y) + f_1(x, y) q_1(x, y) \tag{12}$$

**Proof 1.1. Existence of $q_0$ and $q_1$**

We have $P(x, y) = yp_1(x) + p_0(x)$ that cancels on all pairs $(x_i, y_i)$.
Then we can rewrite $P(x, y)$ :

$$\begin{aligned} P(x, y) &= yp_1(x) + p_0(x) \\ &= (y - l(x))p_1(x) + l(x)p_1(x) + p_0(x) \\ &= f_1(x, y)p_1(x) + l(x)p_1(x) + p_0(x) \end{aligned}$$

From there, we can define $r(x, y) = P(x, y) - f_1(x, y)p_1(x) = l(x)p_1(x) + p_0(x)$
Since r(x, y) is purely in x, and that it cancels out on all pairs $(x_i, y_i)$, we can rewrite it as the functions : $f_0(x)q_0(x, y)$

In the end, we notice that we have $P(x, y) = f_1(x, y)q_1(x, y) + f_0(x)q_0(x, y)$ with $q_1(x, y) = p_1(x)$

## 3.2 Second case $(f_2)$

Now for the second case, we consider both subspaces $D$ and $E$ from which we can define the following functions :

$f_0(x) = \prod_{i=0}^{d+e-1} (x - x_i)$

$f_1(x, y) = (y - L(x))(\prod_{i=d}^{d+e-1} (x - x_i))$ where $L(x)$ is a Lagrange polynomial (in $D$)

$f_2(x, y) = y^2 + ?y + ?$ $f_2$ being the polynomial we're looking for

To make it clearer, we're going to name the coefficients of $f_2$ as follows:

$$f_2(x, y) = y^2 + l_1(x, y)y + l_0(x, y)$$

**Proof 1.2. Existence of $q_0$, $q_1$ and $q_2$**

We have $P(x, y) = y^2P_2(x) + yP_1(x) + P_0(x)$ that cancels out on all pairs $(x_i, y_i)$. Then

we can rewrite $P(x,y)$ :

$$\begin{aligned}
P(x,y) &= y^2P_2(x) + yP_1(x) + P_0(x) \\
&= f_2(x,y)P_2(x) - l_1(x,y)yP_2(x) - l_0(x,y)P_2(x) + yP_1(x) + P_0(x) \\
&= f_2(x,y)P_2(x) + y\underbrace{(-l_1(x,y)yP_2(x) + yP_1(x))}_{\tilde{P}_1}\underbrace{-l_0(x,y)P_2(x) + P_0(x)}_{\tilde{P}_0} \\
&= f_2(x,y)P_2(x) + y\tilde{P}_1(x) + \tilde{P}_0(x) \\
\underbrace{P(x,y) - f_2(x,y)P_2(x)}_{\text{cancel out on every}(x_i,y_i)} &= y\tilde{P}_1(x) + \tilde{P}_0(x) \\
&= f_0(x)\tilde{q_0}(x,y) + f_1(x,y)\tilde{q_1}(x,y) \text{ , using the first proof 1.1}
\end{aligned}$$

With 1.1 and 1.2, we have proven the existence of the polynomials $f_0(x)$, $f_1(x,y)$ and $f_2(x,y)$ which form the basis for our specific case : two-variable Gröbner basis.

Unfortunately, we did not manage to prove the expression of $f_2$ which is :

$$f_2(x,y) = f_0(x)\tilde{q_0} + f_1(x,y)\tilde{q_1} + G(x,y)Q(x,y) \tag{13}$$

where $\tilde{q_0}, \tilde{q_1}$ and $Q$ are polynomials in $x$ and $y$ and $Q = 1$

For instance, we used a Gröbner basis engine to compute $f_2(x,y)$ for $G(x,y) = y^2 \cdot (1 - \frac{x}{2}) + y \cdot (x-1) - \frac{3}{2} \cdot x$ found in our example 4. We find :

$$f_2(x,y) = y^2 - y - 3x$$

# 4 Conclusion

Through this study, we delved into Gröbner bases and learned the core algebra notions used to construct one. Although we did not get to the implementation part because of our poor understanding and slow learning pace, we were able to grasp the ingenuity of these mathematical constructs, which simplify the solving of systems of polynomial equations, as well as the complexities involved in their construction.

Today, Gröbner bases find applications in numerous fields that need to solve polynomial-related problems and researchers are still looking to refine algorithms such as the latest `F5` algorithm [4]. Had we progressed further, we would have tried to implement an algorithm for computing a Gröbner basis in our specific case (i.e. with two variables) and compared our results with an existing calculation engine such as `MAPLE` or `msolve` open source library in C (both F4 algorithm implementations). Given the arduous nature of computing a Gröbner basis, we could have compared the computing time of our implementation and studied its complexity.

# References


[1] Bruno Buchberger. "Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal". In: *Journal of Symbolic Computation* 41.3 (2006). Logic, Mathematics and Computer Science: Interactions in honor of Bruno Buchberger (60th birthday), pp. 475–511. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2005.09.007. URL: https://www.sciencedirect.com/science/article/pii/S0747717105001483 (page 1).

[2] B. Buchberger. "A theoretical basis for the reduction of polynomials to canonical forms". In: *SIGSAM Bull.* 10.3 (Aug. 1976), pp. 19–29. ISSN: 0163-5824. DOI: 10.1145/1088216.1088219. URL: https://doi.org/10.1145/1088216.1088219 (page 1).

[3] Jean-Charles Faugére. "A new efficient algorithm for computing Gröbner bases (F4)". In: *Journal of Pure and Applied Algebra* 139.1 (1999), pp. 61–88. ISSN: 0022-4049. DOI: https://doi.org/10.1016/S0022-4049(99)00005-5. URL: https://www.sciencedirect.com/science/article/pii/S0022404999000055 (page 1).

[4] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. "On the complexity of the F5 Gröbner basis algorithm". In: *Journal of Symbolic Computation* 70 (Sept. 2015), pp. 49–70. DOI: 10.1016/j.jsc.2014.09.025. URL: https://inria.hal.science/hal-01064519 (page 13).

[5] Jérémy Berthomieu, Christian Eder, and Mohab Safey El Din. "msolve: A Library for Solving Polynomial Systems". In: *2021 International Symposium on Symbolic and Algebraic Computation*. 46th International Symposium on Symbolic and Algebraic Computation. Saint Petersburg, Russia: ACM, July 2021, pp. 51–58. DOI: 10.1145/3452143.3465545.

[6] Wikipedia contributors. "Gröbner basis — Wikipedia, The Free Encyclopedia". 2024. URL: https://en.wikipedia.org/w/index.php?title=Gr%C3%B6bner_basis&oldid=1223889442.

[7] Mateusz Paprocki. "Gröbner Bases and Their Applications — Polynomials Manipulation Module v1.0 Documentation". URL: https://mattpap.github.io/masters-thesis/html/src/groebner.html.

[8] Eberhard Becker et al. "The shape of the Shape Lemma". In: *Proceedings of the International Symposium on Symbolic and Algebraic Computation*. ISSAC '94. Oxford, United Kingdom: Association for Computing Machinery, 1994, pp. 129–133. ISBN: 0897916387. DOI: 10.1145/190347.190382. URL: https://doi.org/10.1145/190347.190382.

[9] Daniel Lazard. "Ideal Bases and Primary Decomposition: Case of Two Variables". In: *J. Symb. Comput.* 1 (1985), pp. 261–270. URL: https://api.semanticscholar.org/CorpusID:41464701.

[10] David A. Cox, John Little, and Donal O'Shea. "Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra". 3rd. Springer Publishing Company, Incorporated, 2010. ISBN: 1441922571.