



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Bases de Gröbner: Un método alternativo para la resolución de
sistemas complejos*

Grado en Ingeniería Mecánica

ALUMNO: David Martínez Sol

DIRECTORES: Rodrigo Mariño Villar

Sandro Caeiro Oliveira

CURSO ACADÉMICO: 2023-2024

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Bases de Gröbner: Un método alternativo para la resolución de
sistemas complejos*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Infantería de Marina

Universida_{de}Vigo

RESUMEN

La definición y resolución de sistemas de ecuaciones es uno de los recursos más básicos que ofrecen las matemáticas. Además, no solo se utilizan en este ámbito, sino que condicionan otras ciencias a la hora de estudiar ciertos modelos de comportamiento en los que previamente se establecen determinadas condiciones. Diversos métodos son conocidos y estudiados para la resolución de dichos sistemas, no obstante, a medida que aumenta la complejidad del sistema a resolver este tipo de método genera problemas computacionales de elevado tiempo de procesado.

Es aquí donde aparecen las bases de Gröbner. En 1976, el matemático Bruno Buchberger publicó un algoritmo que trataba de generalizar el algoritmo de Euclides con la finalidad de simplificar un sistema de ecuaciones, aparentemente complejo e incluso no lineal, en uno equivalente pero más sencillo. Este algoritmo impulsó el avance de las matemáticas y de otras ciencias convirtiendo problemas irresolubles con métodos convencionales, aplicando las bases de Gröbner, en problemas realizables con un simple ordenador. En este trabajo se plantea el estudio y la aplicación, no solo del algoritmo en cuestión, sino también se repasa las temáticas simples del álgebra elemental que dan lugar a estas bases.

PALABRAS CLAVE

Ideal, Base, Sistema, Algoritmo, Computación.

CONTENIDO

Contenido.....	1
Índice de Figuras.....	3
Índice de Tablas.....	5
1 Introducción y objetivos	7
1.1 Introducción.....	7
1.2 Objetivos.....	8
2 Estado del arte.....	9
2.1 Ecuaciones polinomiales	9
2.1.1 Introducción	9
2.1.2 Primeras fórmulas para la resolución de ecuaciones de grado $2 \leq n \leq 4$	10
2.1.3 Teorema Fundamental del Álgebra y primeros algoritmos.....	10
2.2 Sistemas de ecuaciones lineales	11
2.2.1 Primeros acercamientos a la resolución de sistemas de ecuaciones lineales	11
2.2.2 Siglos XVII – XVIII.....	12
2.3 Algoritmos y métodos	13
2.3.1 Algoritmo de Euclides.....	13
2.3.2 Eliminación Gaussiana y otros métodos	15
2.4 Algoritmo de Buchberger e introducción de las bases de Gröbner	15
3 Desarrollo del TFG	17
3.1 Preliminares.....	17
3.1.1 Polinomios.....	18
3.1.2 Anillos	19
3.1.3 Generadores e ideales.....	19
3.2 Bases de Gröbner.....	22
3.2.1 Órdenes monomiales.....	22
3.2.2 Algoritmo de la división.....	24
3.2.3 Ideales monomiales y el lema de Dickson	26
3.2.4 Teorema de las Bases de Hilbert y Bases de Gröbner	27
3.2.5 Propiedades de las bases de Gröbner y S-polinomios.....	31
3.2.6 Algoritmo de Buchberger.....	32
4 Aplicaciones y ejemplos	37

4.1 Bases de Gröbner como herramienta para simplificar la resolución de sistemas de ecuaciones polinomiales.....	37
4.1.1 Solución de ecuaciones polinomiales.....	37
4.1.2 Resolución de casos de programación entera.....	41
4.2 Otras aplicaciones de las bases de Gröbner.....	45
4.2.1 Problema de pertenencia a un ideal.....	45
4.2.2 Teoría de Grafos (coloración)	46
4.2.3 Resolución de forma implícita	48
5 Conclusiones y líneas futuras	51
5.1 Conclusiones.....	51
5.2 Líneas futuras	52
6 Bibliografía	53
Anexo I: Implicaciones Sociales, Económicas y Ambientales.....	55
Anexo II. Plantilla-código en SageMath para resolver un sistema de ecuaciones lineales utilizando las bases de Gröbner	57
Resolución de sistema de ecuaciones lineales.....	57
Programación Entera	58

ÍNDICE DE FIGURAS

Figura 2-1 Recreación de Yoshio Mikami del diagrama de Seki. Fotografía obtenida de [4]	12
Figura 2-2 Código en la plataforma SageMath de implementación del algoritmo de Euclides para hallar el máximo común divisor.....	14
Figura 2-3 Código en la plataforma SageMath de implementación del algoritmo de Buchberger para hallar la base de Gröbner de un ideal	16
Figura 3-1. Cálculo del máximo común divisor con SageMath para el ejemplo 3.3.....	21
Figura 3-2. Polinomio ordenado lexicográficamente mediante código en SageMath	23
Figura 3-3 Comprobación mediante la plataforma SageMath del ejemplo 3.9.	29
Figura 4-1 Código en SageMath para obtener la base de Gröbner del ideal correspondiente al sistema de ecuaciones del ejemplo 4.1.....	37
Figura 4-2 Código en SageMath para resolver el sistema de ecuaciones resultante de la base de Gröbner del ejemplo 4.1.....	38
Figura 4-3 Código en SageMath para computar la base de Gröbner del ideal correspondiente al sistema de ecuaciones del ejemplo 4.2.....	39
Figura 4-4 Sistema de ecuaciones correspondiente al tensor de Bach [9].....	40
Figura 4-5 Código en la plataforma SageMath para computar la base de Gröbner y realizar la división correspondiente al problema planteado en el ejemplo 4.4	43
Figura 4-6 Código en SageMath para computar la base de Gröbner del ideal correspondiente al problema-ejemplo de carga de munición planteado	44
Figura 4-7 Grafo a colorear correspondiente al Ejemplo 4.7 [12].....	46
Figura 4-8 Código en SageMath para obtener la base de Gröbner del ideal I obtenido del grafo del ejemplo 4.7.....	47
Figura 4-9 Mapa de Sudamérica del ejemplo 4.7 3-coloreado	47
Figura 4-10 Computación en SageMath de la base de Gröbner del ideal correspondiente a las expresiones paramétricas del Ejemplo 4.8.....	48
Figura 4-11 Computación en SageMath de la base de Gröbner del ideal correspondiente a las expresiones paramétricas del Ejemplo 4.9	49

ÍNDICE DE TABLAS

Tabla 1. Símbolos del lenguaje matemático utilizados en el trabajo.....	17
Tabla 2. Tabla explicativa del código-plantilla correspondiente a la construcción de un ideal a partir de ecuaciones polinomiales y la computación de la respectiva base de Gröbner	57
Tabla 3. Tabla explicativa del código-plantilla correspondiente a la resolución del sistema formado por los elementos de la base de Gröbner hallada	58
Tabla 4. Tabla explicativa correspondiente a la primera parte de la resolución de un problema de programación entera.....	59
Tabla 5. Tabla explicativa correspondiente a la muestra de la solución óptima de un problema de programación entera.....	59

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción

El álgebra (*al jebr* en árabe) es la rama de las matemáticas focalizada en el estudio de las estructuras y reglas generales que gobiernan las operaciones aritméticas y las relaciones numéricas. En ella, no se trabaja con números específicos como se está acostumbrado a tratar, sino que se utilizan símbolos y letras que representan cantidades desconocidas, constantes, relaciones o variables. De entre las abundantes áreas algebraicas que existen, el trabajo se moverá entre el álgebra elemental, el álgebra abstracta y el álgebra computacional. En primer lugar, se parte de los conceptos pertenecientes al campo del álgebra elemental, encargada de extraer los problemas del mundo real y reducirlos a una serie de números, letras y símbolos, así como del estudio de los conceptos y operaciones básicas (las ecuaciones lineales, su resolución y sistemas de ecuaciones). A diferencia de la aritmética donde generalmente se opera con números y sus respectivas operaciones (suma, multiplicación, resta y división) en álgebra elemental los números son representados por símbolos (x, y, z) siendo útil para el desarrollo y análisis de las propiedades de los números reales y especialmente para referirse a números “desconocidos”, definir ecuaciones y poder resolverlas. En este campo se encuentran elementos tales como potencia o expresión algebraica, otorgando especial importancia en el presente trabajo dentro de las mismas, a los polinomios, definidos y desarrollados más profundamente en el apartado 3.1.1 del documento.

Por otra parte, la mayoría de los conceptos a tratar en el trabajo pertenecen al campo del álgebra abstracta, también denominada moderna. Esta es la rama del álgebra que estudia estructuras algebraicas más complejas (grupos, anillos, dominios de integridad o cuerpos) que permiten definir conjuntos abstractos, operar con ellos y estudiar sus propiedades fundamentales, aplicables a posteriori en cualquier ámbito de las matemáticas. Trata de estudiar y procesar lo extraíble de las afirmaciones lógicas en las que se basan la totalidad de las matemáticas. Como queda escrito a lo largo del apartado de desarrollo, el álgebra moderna satisface el método axiomático, es decir, se trata de vincular un conjunto de conceptos que cumplen una serie de propiedades y relaciones entre ellos. Cabe enfatizar el poder del desarrollo axiomático en estas estructuras algebraicas a la hora de extraer y revelar semejanzas entre objetos a primera vista incompatibles, llegando al punto en que un teorema sobre una estructura puede tener un análogo en una estructura diferente. De este campo algebraico, se profundizará en los anillos de polinomios, conjunto de polinomios que cumplen una serie de propiedades específicas, y en los ideales de estos anillos, que agrupan todos los múltiplos de los polinomios o monomios que conformen el anillo. A grandes rasgos, las bases de Gröbner proporcionan un modo particular de generar un ideal de un anillo. Por último, dentro de las operaciones que se realizan dentro del álgebra abstracta, existen

una serie de algoritmos que facilitan el procesamiento de estos sistemas algebraicos. El estudio de estos algoritmos forma parte del álgebra computacional, presente en numerosos apartados del documento.

El presente trabajo comienza con la introducción del concepto del álgebra, especificando los dos tipos que guardan relación con el tema a tratar y mencionando brevemente los conceptos básicos necesarios para el desarrollo de las bases. En el apartado 2.1.2, se estudiará el surgimiento de los primeros métodos de resolución de ecuaciones polinomiales, y cómo se evolucionó, desde la fórmula de resolución de ecuaciones lineales cuadráticas simples, hasta la creación de los primeros algoritmos que permitieron obtener los ceros de una ecuación polinomial, profundizando en como Gröbner incitó a Bruno Buchberger a obtener el algoritmo que daría lugar a las bases a desarrollar en el trabajo.

A partir del tercer apartado, comienza la parte principal del trabajo. En primer lugar, se define y menciona el concepto de polinomio y anillo de polinomios, así como sus principales propiedades. A continuación, se desarrollan los ideales de un anillo de polinomios y las bases de Hilbert que demuestran la finitud de los ideales generadores de dichos anillos. A partir de estas bases se introducirá el concepto de orden monomial, comentando como los términos y coeficientes principales de un ideal de polinomios conforman la estructura de una base de Gröbner; y se desarrolla el algoritmo de Buchberger, generador de las bases en cuestión.

En el apartado 4, y haciendo uso de la plataforma de SageMath, se mencionarán las aplicaciones más reseñables que hacen uso del algoritmo, profundizando en la resolución de sistemas y se plantearán casos y problemas que hagan uso de las bases de Gröbner para maximizar y resolver de sistemas de ecuaciones lineales, desde los más simples hasta algunos más complejos. De esta forma, quedará demostrada la efectividad del algoritmo en la resolución de todos ellos.

1.2 Objetivos

Con lo anteriormente expuesto, los objetivos específicos de este Trabajo de Final de Grado, con el fin de dar a conocer las Bases de Gröbner en el ámbito de la resolución de sistemas complejos, son:

- Dar una visión general de los conceptos pertenecientes al campo del álgebra abstracta necesarios para conformar y entender el funcionamiento del algoritmo de Buchberger.
- Describir la aplicación del algoritmo y la definición de las bases de Gröbner.
- Comprobar la efectividad de las bases en cuanto a la simplificación del proceso resolutivo de un sistema de ecuaciones mediante la introducción de aplicaciones orientadas a la optimización de cálculos en el ámbito de la Armada.

2 ESTADO DEL ARTE

2.1 Ecuaciones polinomiales

2.1.1 Introducción

Los problemas de resolución de ecuaciones polinomiales del tipo:

$$(2.1.1) \quad p(x) = p_0 + p_1x + p_2x^2 + \cdots + p_nx^n = 0$$

fueron desarrollados por los Sumerios (3000 A.C.) y ha influenciado de manera significativa el desarrollo de las matemáticas a lo largo de los siglos. Particularmente, las ideas de pensamiento abstracto y el uso de la notación matemática parten del estudio de este problema. Es más, este estudio ha motivado históricamente la introducción de muchos de los conceptos fundamentales de las matemáticas (los números irracionales y complejos, grupos algebraicos, cuerpos e ideales) y ha influenciado significativamente el desarrollo de la computación numérica.

Hoy en día, muchos problemas computacionales que surgen en ámbitos científicos, ingenieriles o en el desarrollo de negocios han sido resueltos usando herramientas del álgebra y la programación lineal. Estas herramientas permiten la resolución de problemas del tipo (2.1.1) pero normalmente para n pequeñas. Cuando n crece hasta el valor de 10 o 20, las herramientas actuales necesarias para resolver la ecuación (2.1.1) disminuyen. Es por ello la importancia del área de la computación denominada álgebra computacional, donde se encuentran situaciones en las que (2.1.1) presenta $n \geq 100$ o incluso 1000. Para estos casos, el número de bits a utilizar tanto para la representación de los coeficientes $p_0, p_1, \dots, p_n$, como para la resolución de los valores de x , suelen causar problemas en los softwares destinados a este uso y motiva la búsqueda de algoritmos para simplificar la resolución de ecuaciones más complejas.

2.1.2 Primeras fórmulas para la resolución de ecuaciones de grado $2 \leq n \leq 4$

El primer paso importante en la historia de la resolución de ecuaciones polinomiales fue dado por

$$(2.1.2) \quad x_{1,2} = \frac{-p_1 \pm \sqrt{p_1^2 - 4p_0p_2}}{2p_2}.$$

los Babilonios (aproximadamente 2000 A.C.) cuando, centrándose en el estudio de ecuaciones de grados más pequeños, consiguieron dar con la expresión genérica para la resolución de ecuaciones cuadráticas. Esta expresión corresponde con la estudiada en las matemáticas de nivel secundaria y bachillerato.

El estudio a fondo de esta expresión provocó la introducción en el mundo de las matemáticas de los números negativos, irracionales y complejos. Ejemplo de ello fue el descubrimiento de que la ecuación $x^2 = 2$ no tiene solución racional, o la expresión $x^2 + 1 = 0$ que da lugar a los números complejos. Más adelante en el siglo XVI, empezaron a surgir intentos de encontrar una fórmula, como la utilizada en (2.1.2), que únicamente utilizase operaciones aritméticas y radicales para resolver ecuaciones polinomiales de grado 3 y 4. No obstante, estos intentos fallaron con polinomios de grado n para cualquier $n > 4$. Esta imposibilidad de encontrar una fórmula capaz de satisfacer la condición de grado $n > 4$ derivó en un teorema llevado a cabo por Ruffini en 1813 y por Abel en 1827, y quedó reforzado por la teoría de Galois en 1832.

2.1.3 Teorema Fundamental del Álgebra y primeros algoritmos

A pesar de la ausencia de fórmulas que resuelvan (2.1.1) con cualquier n mediante el uso de operaciones aritméticas simples y radicales, el Teorema Fundamental del Álgebra afirma que toda ecuación polinómica de grado n con coeficientes complejos tiene n raíces complejas. Este teorema (TFA en adelante) fue llevado a cabo de manera exitosa por Girard y Descartes a mediados del siglo XVII, no obstante, su demostración tuvo que esperar hasta finales del siglo XIX, cuando Alexander Ostrowski consiguió solucionar los imperfeitos de la demostración que Gauss defendió en su disertación doctoral de 1799.

El TFA desarrolla la demostración de la existencia de la factorización para cualquier polinomio

$$p(x) = p_n \prod_{j=1}^n (x - z_j)$$

tal que $p_n \neq 0$. De esta manera, $z_1, z_2, \dots, z_n$ son las n soluciones para (2.1.1). El área que trata de computar o aproximarse al valor de estos ceros o soluciones se denomina *aspectos algorítmicos del TFA*. El objetivo principal de esta área es tratar de optimizar estos algoritmos de manera que se obtengan soluciones lo más precisas posibles disminuyendo al máximo el coste computacional, refiriéndose este a tiempo y memoria consumidos. Jan Brouwer y Hermann Weyl fueron los primeros en introducir

algoritmos que garantizaban la convergencia hacia todos los n ceros de cualquier $p(x)$ en el año 1924. No obstante, sendos algoritmos consumían un gran número de recursos computacionales.

2.2 Sistemas de ecuaciones lineales

2.2.1 Primeros acercamientos a la resolución de sistemas de ecuaciones lineales

Los primeros métodos que trataron de estandarizar la resolución de sistemas de ecuaciones lineales aparecieron, de la forma que ha sido comentada en el apartado anterior, en la época de la matemática de los Babilonios (1700 A.C.). Estos trabajaban los sistemas sin utilizar una notación simbólica como la que se conoce hoy en día, utilizaban, por tanto, términos geométricos para designar las incógnitas como la longitud, la anchura o área, sin que estas tuvieran relación con una unidad de medida estandarizada. Por otra parte, los egipcios dejaron constancia, en sus papiros y grabados, de multitud de problemas matemáticos resueltos. El álgebra egipcia se centró de manera casi exclusiva en la resolución de ecuaciones lineales y de la misma manera que los babilonios, todos los problemas se relacionaban con cuestiones cotidianas. Los egipcios utilizaban el método Regula Falsi para la resolución tanto de ecuaciones polinomiales, como para la resolución de sistemas de ecuaciones lineales sencillos. Este método utiliza la técnica prueba-error, probando diferentes valores de la variable a obtener y ajustándola hasta el cumplimiento de la igualdad. Un ejemplo de ello se encuentra en el problema 26 del Papiro de Rhind, copiado por el escriba egipcio Ahmes en el año 1650 A.C., que trata de resolver la expresión $x + x/4 = 15$. Por Regular Falsi o Falsa Posición, se comprueba que con $x=4$ se cumple que $4 + 4/4 = 5$ obteniendo además un valor entero, pero 3 veces más pequeño que el que se desea obtener en la ecuación original, por tanto, se multiplica por 3 este valor de “prueba” obteniendo así el resultado para la ecuación original $12 + 12/4 = 15$. De esta manera, tanto babilonios y egipcios como las civilizaciones que les precedieron, fueron capaces de resolver sistemas sencillos pertenecientes a entornos cotidianos.

El siguiente gran avance llevado a cabo en el ámbito del álgebra lineal, fue realizado por la civilización china con la aparición del libro *Nine Chapters of Mathematical Art*, escrito por diversas generaciones de estudiantes entre los siglos I y II A.C. y traducido al inglés en el año 1606, siendo uno de los libros de matemáticas conservados más antiguos de la civilización China. El libro contiene 246 problemas con sus respectivos métodos de solución, abarcando campos como la medición, el comercio, la tasación o problemas de distancia, ritmo y tiempo. Para tratar de contextualizar las matemáticas que contiene el libro, es importante considerar las herramientas e ideas matemáticas que utilizaban los estudiantes chinos en la época en la que se escribió el libro. La más sobresaliente entre ellas es la tabla de contar utilizada por los matemáticos chinos en la que se contaban varillas de bambú organizadas en cajones rectangulares en la que se podían realizar operaciones lineales sencillas (suma, resta, multiplicación y división) y que ayudó a la adaptación de esta civilización a la numeración en base 10, introducida en el imperio chino en el año 200 A.C.

En la traducción del libro se ofrece una introducción a los contextos que han impulsado el estudio de los sistemas de ecuaciones lineales en la antigua China. Los dieciocho problemas que aparecen en el Capítulo 8 del libro tratan sistemas de entre 2 y 6 valores desconocidos. Con alguna excepción, todos los problemas cuentan con una solución única con el número de ecuaciones igual al número de incógnitas. La mayor parte de ellos tratan temas agrícolas trabajando con cantidades de ganadería o grano (número, peso, volumen o coste). No obstante, también se tratan conceptos prácticos como la fuerza de empuje de un caballo, la cantidad de agua utilizada por familias que comparten un aljibe o la cantidad de pollos consumidos por persona dependiendo de la clase social.

Con la excepción de los métodos desarrollados sobre siglo II A.C en China, apenas se ha progresado en cuanto a la comprensión teórica de los sistemas de ecuaciones lineales y sus soluciones hasta los siglos XVII – XVIII cuando emergieron los determinantes de manera casi simultánea en

Japón y en Europa. En el año 1683, el matemático japonés Seki Kowa desarrolló una versión de los determinantes como parte de la resolución de sistemas no lineales de ecuaciones. Este método fue descrito de manera que los coeficientes de las variables de un sistema eran colocados en matrices rectangulares (claro ejemplo de la influencia de las tablas rectangulares de contar chinas mencionadas en el párrafo anterior). Yoshio Mikami en el año 1914 recreó una ilustración del manuscrito original de Seki para los casos de matrices 2x2 y 3x3 como se desarrolla en la Figura 2-1.

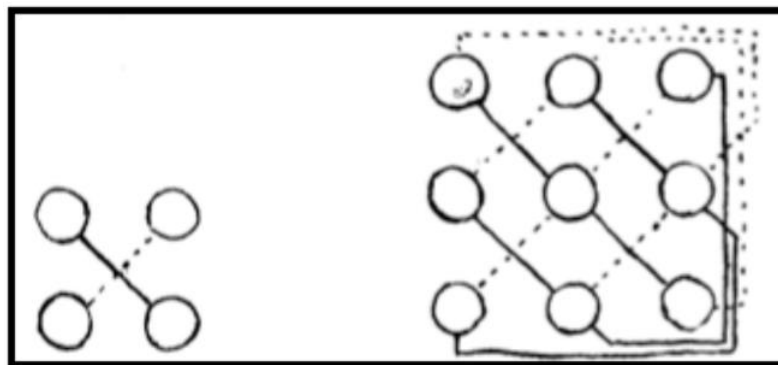


Figura 2-1 Recreación de Yoshio Mikami del diagrama de Seki.
Fotografía obtenida de [4]

Las líneas de trazo continuo o punteadas se usan para indicar los signos del producto de los elementos conectado por estas, siendo las líneas de trazo continuo de signo positivo, y las líneas de trazo punteado de signo negativo.

2.2.2 Siglos XVII – XVIII

Más adelante, en el año 1750, el matemático suizo Gabriel Cramer desarrolló de manera independiente los determinantes como un método específico de solución a un sistema como una expresión de coeficientes cualesquiera del sistema dado. Por ejemplo, Cramer expresa un sistema lineal 3x3 con 3 variables desconocidas x, y, z de la siguiente forma [1]:

$$A = Z_1z + Y_1y + X_1x$$

$$A = Z_2z + Y_2y + X_2x$$

$$A = Z_3z + Y_3y + X_3x$$

Cramer expresa a partir de los coeficientes del sistema los valores desconocidos de x, y, z . Por ejemplo, la resolución de la variable z vendría dada por la expresión:

$$z = \frac{A_1Y_2X_3 - A_1Y_3X_2 - A_2Y_1X_3 + A_2Y_3X_1 + A_3Y_1X_2 - A_3Y_2X_1}{Z_1Y_2X_3 - Z_1Y_3X_2 - Z_2Y_1X_3 + Z_2Y_3X_1 + Z_3Y_1X_2 - Z_3Y_2X_1}$$

Cabe destacar, la falta de explicación por parte de Cramer sobre el origen del resultado. No obstante, proporciona una regla general para la resolución de sistemas: toda variable en un sistema es resuelta mediante un valor fraccional. Este método también menciona el hecho de que un denominador de valor 0 indica que el sistema no tiene una solución única especificando, además, que en estos casos el sistema tendrá infinitas soluciones. Esta perspectiva de Cramer hubiese resultado imposible sin el uso de la notación algebraica introducida por el matemático francés François Viète. Los símbolos matemáticos (por ejemplo, el uso la letra 'x' para referirse a una cantidad fija desconocida o una variable) empezaron a usarse en el año 300 A.C en Grecia por Euclides, pero su uso no se estandarizó hasta el año 1591, cuando el matemático francés introdujo las vocales para representar cantidades desconocidas y consonantes para representar variables, dando a conocer las ideas básicas de lo que hoy en día se

consideran los parámetros. Tanto para Seki como para Cramer, la manera en la cual el sistema de notación condicionaba los coeficientes conformaban la manera de especificar el determinante. Por un lado, la articulación de los determinantes de Seki siguiendo el modelo de la tabla de contabilizar china, influenciaba la disposición física de los coeficientes con tal de especificar las operaciones que posteriormente se realizarían con dichos coeficientes. La organización que disponía Cramer, en cambio, dependía del uso de los subíndices para identificar cada uno de los coeficientes y así poder crear una expresión fija que permitía la resolución de este.

Otro avance importante en el desarrollo de la teoría de sistemas de ecuaciones lineales tuvo lugar también en Francia en el año 1750: Leonard Euler empezó a cuestionarse si un sistema de n ecuaciones lineales con n incógnitas tenía una solución única, utilizando como ejemplo el sistema:

$$3x - 2y = 5$$

$$4y = 6x - 10$$

Euler desarrolló esta idea con sistemas más extensos, demostrando que ciertas ecuaciones pueden estar tanto contenidas, como compuestas por otras ecuaciones, siendo este concepto de dependencia lineal de vital importancia para entender las bases de Gröbner a tratar en el presente trabajo.

Por último, en el siglo XIX, Gauss dio con la manera de demostrar, sin la necesidad de resolver un sistema, si este no tenía solución, solución única o infinitas soluciones. Es decir, permite identificar mediante el método de la eliminación la naturaleza de la solución de un sistema. Este método de la eliminación de Gauss fue utilizado principalmente en el ámbito de la astronomía. El mismo Gauss trabajó en la órbita de un asteroide llamado Pallas, considerado en la época un planeta. Mediante la observación de Pallas durante varios años, reunió una serie de datos que utilizó para determinar la inclinación de la órbita del planeta, y al añadirlos a las expresiones de las teorías astronómicas conformó un sistema de 12 ecuaciones y 6 incógnitas. Para aproximar de la mejor manera la solución de este sistema, Gauss desarrolló el método de los mínimos cuadrados, y fue de este método del cual se extrajo el concepto de la eliminación. Con todos estos conceptos, Gauss trató de solucionar el problema de la órbita de Pallas de la manera más aproximada a la realidad posible, teniendo siempre en cuenta que las medidas y observaciones siempre iban a traer consigo errores que complicarían la resolución del sistema.

2.3 Algoritmos y métodos

2.3.1 Algoritmo de Euclides

El algoritmo desarrollado por el matemático y geómetra griego Euclides (300 A.C.) es considerado como el algoritmo no trivial más longevo conservado hoy en día. Este trata de computar el máximo común divisor de dos números enteros positivos. No obstante, hoy en día, el algoritmo se puede aplicar sobre cualquier dominio de factorización en el cual el máximo común divisor es computable. El método propuesto por Euclides dicta lo siguiente: El máximo común divisor entre a y b (bien sean números en $\mathbb{R}$ o polinomios) es un elemento c máximo que divide ambos de manera exacta, sin resto. El máximo común divisor (mcd en adelante) es, además, en el ámbito de los polinomios, el generador de un ideal (concepto explicado más en detalle en el apartado 3.2.3). Por lo tanto, se puede afirmar que: $mcd(a, b) = h \rightarrow h$ es generador del ideal $\langle a, b \rangle$. Se considera el algoritmo precursor del algoritmo de Buchberger (2.4), debido a que ambos trabajan con el mismo fin.

Sean f y g dos polinomios que cumplen $f = gq + r$ donde q y r denotan el coeficiente y el resto de la división respectivamente. Para calcular el mcd mediante el algoritmo se procede de la siguiente manera [2] :

Polinomios de entrada : f, g
Polinomio resultante del cálculo del mcd: $h = \text{mcd}(f, g)$
 $h = f$
 $s = g$
(mientras el resto sea distinto de 0) WHILE $s \neq 0$ DO
 $\text{rem} := \text{remainder}(h, s)$
 $h = s$
 $s = \text{rem}$
RETURN h .

En primer lugar, se asignan las variables h y s a los polinomios f y g respectivamente. Se calcula el resto de la división de h entre s utilizando la función *remainder* (h, s) y el resto se almacena en la variable *rem*. Se actualiza h con el valor de s , y s con el valor de *rem*, repitiéndose este bucle hasta que la variable s sea igual a 0. En ese momento, el algoritmo devolverá el valor de h que representa el *mcd* (f, g). Por ejemplo, para hallar, utilizando SageMath el mcd de los polinomios indicados a continuación, el código sería del tipo:

$$f = x^3 - 2x^2 + 2x - 1$$
$$g = x^2 - 1$$

```
# 1. Se importa el módulo de polinomios de SageMath
from sage.rings.polynomial.polynomial import PolynomialRing

# 2. Se define el anillo de polinomios sobre el cuerpo de los números racionales
R = PolynomialRing(QQ, 'x')

# 3. Se definen los polinomios f y g
f = R('x^3 - 2*x^2 + 2*x - 1')
g = R('x^2 - 1')

# 4. Se presenta el cambio de variables y el bucle explicados en el párrafo anterior
def mcd_polynomials(f, g):
    h = f
    s = g
    while s != 0:
        rem = h % s
        h = s
        s = rem
    return h

# 5. Se devuelve el valor del mcd
h = mcd_polynomials(f, g)

# 6. Se muestra el valor de h
print("MCD de f y g:", h)
```

Figura 2-2 Código en la plataforma SageMath de implementación del algoritmo de Euclides para hallar el máximo común divisor

Tal código devuelve el polinomio $x - 1$, divisor de ambos polinomios. No obstante, este procedimiento, descrito en la obra *Escritos* (300 A.C.) del mismo Euclides, no calcula el *mcd* de polinomios con varias variables. Por ejemplo, tomando los polinomios xy y xz podemos observar claramente que $\text{mcd}(xy, xz) = x$, sin embargo, sin importar que orden de términos se escoja, el algoritmo al dividir xy entre xz da 0 de cociente, con resto de valor xy . Por consecuencia, ningún polinomio se reduce respecto al otro y no existe un paso subsiguiente al cual aplicarle la segunda parte del algoritmo de Euclides. Más adelante, en el presente trabajo se estudiará la relación que guarda este método de obtener polinomios generadores de un ideal con el algoritmo de Buchberger.

2.3.2 Eliminación Gaussiana y otros métodos

Siguiendo con la aportación de Gauss al álgebra lineal en el ámbito de la resolución de sistemas, cabe mencionar la eliminación Gaussiana, un algoritmo para resolver sistemas de ecuaciones que permite además de la obtención de la solución, computar el determinante para extraer el rango de la matriz de coeficientes. Consta de dos partes: la sustitución o eliminación hacia delante y la sustitución o eliminación hacia atrás (traducción del inglés *forward and backward elimination*). Por una parte, en la sustitución hacia delante se eliminan las incógnitas trabajando con las ecuaciones y conformando una forma escalonada que nos permite estudiar la naturaleza de la solución del sistema. Por otra parte, en la sustitución hacia atrás se realiza un proceso de sustitución sobre la matriz triangular superior reducida, resultando en la solución de la ecuación. El proceso que sigue es simple: se escribe el sistema en forma de matriz y se transforma en una matriz triangular superior operando con las filas de la matriz para finalmente utilizar la sustitución hacia atrás dando con la solución.

Dentro de la eliminación Gaussiana destaca en importancia la factorización LU (de los términos en inglés *Lower & Upper*). Este concepto juega un papel fundamental en la implementación de la eliminación Gaussiana en el ámbito computacional moderno y consiste en reescribir la matriz A extraída del sistema de ecuaciones en forma de un producto de dos matrices triangulares: una inferior y una superior. Por ejemplo, sea A una matriz cuadrada que representa un sistema, la descomposición LU sería una descomposición de la forma $A = L \cdot U$ teniendo en cuenta que la matriz L únicamente tendría ceros en la parte superior de la diagonal y la matriz U por debajo de ella.

Dentro de los métodos, tanto directos como indirectos para computar los sistemas de ecuaciones lineales, la eliminación Gaussiana es un tipo de método directo para calcular variables desconocidas. El ámbito computacional, tanto científico como ingenieril, utiliza este tipo de ecuaciones con gran cantidad de variables, incógnitas o ecuaciones que necesitan ser resueltas con eficiencia y rapidez, siendo el método de eliminación uno de los primeros algoritmos en acercarse a estos requerimientos.

Además de los métodos directos utilizados para la resolución de sistemas de ecuaciones, en el ámbito computacional también destacan los métodos iterativos que, para ciertos tipos de problemas, también hacen uso de algoritmos para generar una secuencia de soluciones aproximadas. El método de Jacobi, por ejemplo, es un algoritmo de iteración para determinar las soluciones de un sistema de ecuaciones lineales estrictamente diagonal dominante. En este, se tiene en cuenta un número k de iteraciones, y con cada una de ellas, se va aproximando el valor de la solución del sistema hasta dar con el valor adecuado.

2.4 Algoritmo de Buchberger e introducción de las bases de Gröbner

Una vez introducido el concepto de los algoritmos encargados de computar la solución de un sistema de ecuaciones de cara a enfocar el trabajo hacia las bases de Gröbner, es esencial explicar el papel que tuvo el matemático Bruno Buchberger en el desarrollo de estas bases. Bruno Buchberger era alumno de Wolfgang Gröbner, matemático computacional que trabajaba en la universidad de Innsbruck y que dedicó su estudio y obra a conceptos algebraicos como los órdenes monomiales, los anillos zero dimensionales, ideales irreducibles, bases minimales. En el año 1965, Buchberger publicó una de sus

más destacadas disertaciones en la universidad de Innsbruck en Austria. A pesar de la importancia que se le concede a esta obra a día hoy en día hoy, no tuvo gran repercusión en el momento de su publicación. No obstante, Buchberger había resuelto completamente el problema propuesto por su profesor Wolfgang Gröbner (1899-1980): hallar un algoritmo para encontrar los elementos generadores de un ideal polinomial respecto a un orden monomial. A estos elementos generadores los acuñó con el nombre de Bases de Gröbner, como tributo a su mentor. Este algoritmo presentado en la tesis de Buchberger se ha desarrollado hoy en día para ideales de cualquier dimensión. Además de dar a conocer los elementos principales que componen estas bases (explicados a partir del apartado 3 del presente trabajo), como por ejemplo los ideales de términos principales (LT – de la traducción al inglés de *leading term*), las propiedades elementales de un anillo de polinomios o la pertenencia a un ideal; con la tesis aparecieron las primeras aplicaciones en la geometría algebraica (computar bases de Hilbert o solucionar sistemas de ecuaciones complejos).

El periodo de 1965 a 1975 aparentemente fue un periodo pobre para las bases de Gröbner debido a la falta de interés en sus resultados. El periodo de mayor atención en cuanto al estudio de estas bases comenzó en el año 1975, cuando Buchberger fue conocedor de una propuesta llevada a cabo por matemáticos influyentes de la época en la que se proponía añadir a la lista de problemas matemáticos más controvertidos: comprobar la resolubilidad de un sistema de ecuaciones. Fue entonces cuando Bruno Buchberger menciona tener un algoritmo capaz de resolver dicho problema incluso aplicable a otros ámbitos del álgebra computacional. Una vez tenido en cuenta el algoritmo en el entorno matemático de finales de siglo, el trabajo posterior sobre las bases, realizado por el propio Buchberger y por otros autores, ha añadido de manera significativa generalizaciones del método y nuevas aplicaciones de la solución algorítmica de varios problemas fundamentales de la geometría algebraica (la teoría de los ideales de polinomios, el álgebra conmutativa o problemas de complejidad), mediante diversas conferencias llevadas a cabo en centros como el RISC (*Research Institute for Symbolic Computation*) en Linz, fundado por el mismo Buchberger, o el RICAM (*Random Institute for Computational and Applied Mathematics*) que reunió a cada vez más expertos en la materia.

Antes de entrar en detalle en cuanto al desarrollo de las bases, veamos el funcionamiento del algoritmo de manera sencilla mediante SageMath. Se pretende hallar el polinomio generador del ideal: $\langle x^2y - 1, xy^2 - 1 \rangle$.

```
# 1. Se define el anillo polinomial en las variables 'x' e 'y'
R.<x, y> = PolynomialRing(QQ, ['x', 'y'])

# 2. Se define el ideal del cual queremos obtener el generador
I = Ideal(x^2*y - 1, x*y^2 - 1)

# 3. Calcular una base de Gröbner (que utiliza el algoritmo de Buchberger)
usando el método groebner_basis() desde el ideal
G = I.groebner_basis()

# 4. Mostrar la base obtenida
print(G)
```

Figura 2-3 Código en la plataforma SageMath de implementación del algoritmo de Buchberger para hallar la base de Gröbner de un ideal

3 DESARROLLO DEL TFG

3.1 Preliminares

Antes de comenzar a desarrollarse los conceptos preliminares a las bases de Gröbner, se indicarán y definirán mediante la siguiente tabla los símbolos correspondientes al lenguaje matemático utilizados de manera más frecuente en el presente trabajo con tal de facilitar su lectura y comprensión.

Símbolo	Significado	Aparece por primera vez (página)
$\Rightarrow$	Implicación (entonces, por tanto)	20
$\forall$	Cuantificador (para todos)	20
$\in$	Pertenencia	20
$\subseteq$	Subconjunto de	21
$: o /$	Tal que	21
$ $	Divisibilidad	22
$\Leftrightarrow$	Doble implicación (si y solo si)	23
$>, <$	Mayor que, menor que (orden monomial)	24
$\setminus$	Diferencia de conjuntos (sin)	28
$\exists$	Cuantificador existencial (existe...)	30
$\exists!$	Cuantificador existencial con marca de unicidad (existe un único...)	32

Tabla 1. Símbolos del lenguaje matemático utilizados en el trabajo

3.1.1 Polinomios

Para tratar de enlazar el álgebra con la geometría, desarrollaremos el concepto de los polinomios sobre un cuerpo. A pesar del conocimiento general sobre el concepto de polinomio, el término cuerpo puede resultar poco familiar. La intuición básica indica que un cuerpo es un conjunto en el cual se pueden definir suma, resta, multiplicación y división con sus propiedades comunes. La razón por la que estos cuerpos son tan importantes en el álgebra lineal es que esta trabaja sobre cualquiera. Los más comunes utilizados son [2]:

- Los números racionales $\mathbb{Q}$: aquellos que pueden ser representados como el cociente de dos números enteros.
- Los números reales $\mathbb{R}$: cuerpo sobre el cual se dibujan todas las curvas y superficies. Está formado por la totalidad de números naturales, enteros, racionales e irracionales.
- Los números complejos $\mathbb{C}$: aquellos formados por la combinación lineal de números reales e imaginarios.

Los polinomios son un concepto clave para el estudio de las bases de Gröbner, y aunque estos sean ampliamente conocidos, es necesario tener en cuenta varias definiciones en lo relacionado con el uso de varias variables.

Definición 1.1 [3] Un monomio en $x_1, \dots, x_n$ es un producto del tipo:

$$x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_n^{\alpha_n},$$

en el que $x_1, x_2, \dots, x_n$ representan diferentes variables (x, y, z son las más comúnmente utilizadas) y $\alpha_1, \alpha_2, \dots, \alpha_n$ son enteros positivos. El grado total del monomio viene dado por la suma $\alpha_1 + \alpha_2 + \dots + \alpha_n$.

Definición 1.2 [3] Un polinomio f en $x_1, \dots, x_n$ con coeficientes en un cuerpo k es la finita combinación lineal de monomios con sus coeficientes en k , siendo k un cuerpo cualquiera. Se puede escribir cualquier polinomio f en la forma:

$$f = \sum_{\alpha} a_{\alpha} \cdot x^{\alpha}, \quad a_{\alpha} \in k, \quad x^{\alpha} = x_1^{\alpha_1} \cdot \dots \cdot x_n^{\alpha_n}$$

donde el sumatorio se realiza sobre n -tuplas $\alpha = (\alpha_1, \dots, \alpha_n)$. El conjunto de todas las combinaciones lineales de monomios (polinomios) en las variables $x_1, \dots, x_n$ se denomina $k[x_1, \dots, x_n]$. A pesar de la aparente complejidad de las expresiones que contienen varias variables ($x_1, \dots, x_n$) se suele prescindir de los subíndices, y se utiliza como bien se menciona en la Definición 1, las variables x, y, z de modo que los polinomios con una, dos y tres variables se escribirán $k[x]$, $k[x, y]$ y $k[x, y, z]$ respectivamente.

Ejemplo 1.1 [3] El polinomio $f = 2xy^2z + x^2z - \frac{4}{5}y^4$ es un polinomio en $\mathbb{Q}[x, y, z]$.

Definición 1.3 [3] Véase $f = \sum_{\alpha} a_{\alpha} \cdot x^{\alpha}$ un polinomio en $k[x_1, \dots, x_n]$.

1. Se denomina a a_{α} , el coeficiente del monomio x^{α} .
2. Si $a_{\alpha} \neq 0 \rightarrow a_{\alpha} \cdot x^{\alpha}$ es un término de f .
3. El grado total de $f \neq 0$ expresado como $\text{gr}(f)$, es el máximo $|\alpha| \forall a_{\alpha} \neq 0$.
4. El término no nulo de mayor grado se llama término líder, LT en adelante (del inglés *leading term*). Este concepto se ampliará en el apartado 3.2.1.
5. Un polinomio de grado $n \geq 1$ del tipo $a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$ es *mónico* si $a_n = 1$, es decir, si el coeficiente del término líder es 1. Ejemplo: $x^3 + 3y^2 - 2x^2$.

Ejemplo 1.2 [3] El grado total del polinomio $f = 2xy^2z + x^2z - \frac{4}{5}y^4$ es $\text{gr}(f) = 4$, debido a que el $\max|\alpha| = 4$ (correspondiente al primer y último término de los tres monomios que conforman f).

3.1.2 Anillos

Un anillo de polinomios se puede definir de manera sencilla como un conjunto en el que se puede realizar la suma y multiplicación con las propiedades habituales de estas operaciones (con la excepción de que la multiplicación no tiene por qué ser conmutativa).

Definición 2.1 [4] Un anillo A , es un conjunto dotado con dos operaciones cerradas (suma $(+)$ y multiplicación $(\cdot)$) de modo que se verifican las siguientes propiedades:

1. A es un grupo abeliano (conmutativo) con respecto a la suma $(+)$.
2. La multiplicación $(\cdot)$ es una operación asociativa en A .
3. Se cumplen la propiedad distributiva por la izquierda $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$ y por la derecha $c \cdot (a + b) = (c \cdot a) + (c \cdot b)$.
4. Se dice que un anillo conmutativo es un dominio de integridad si es un anillo conmutativo $(A, +, \cdot)$ y si además $a \cdot b = 0 \Rightarrow a = 0$ o $b = 0$ para cualquier par de elementos a, b , es decir, cuando carece de elementos divisores de 0. [5]

Si, además, la multiplicación $(\cdot)$ es conmutativa, se dice que A es un anillo conmutativo. Cabe destacar, que el hecho de que las operaciones se consideren cerradas implica que al realizarlas el resultado siempre pertenecerá al anillo A . A su vez, si el anillo posee un elemento neutro en la multiplicación (1, en este caso) se trata de un anillo con unidad.

Teorema 2.1 [6] Para cualquier anillo conmutativo A , y dos polinomios cualesquiera f y $g \in A[x]$ se verifica que:

1. El grado de la expresión $gr(f + g) \leq \max(gr(f), gr(g))$, así como también $gr(f \cdot g) \leq gr(f) + gr(g)$.
2. Si, además, A es un dominio de integridad se verifica lo siguiente:

$$gr(f \cdot g) = gr(f) + gr(g).$$

3.1.3 Generadores e ideales

En un conjunto de elementos definidos por una operación binaria, en este caso la suma y la multiplicación, un ideal es un subconjunto cerrado bajo esta operación binaria, es decir, la suma o la multiplicación de dos elementos del conjunto produce un elemento resultante que también se encuentra en el conjunto. Si aplicamos esta definición general al ámbito de los anillos de polinomios, la definición de ideal correspondería a la siguiente:

Definición 3.1 [2] Un subconjunto $I \subseteq k[x_1, \dots, x_n]$ es un ideal si cumple lo siguiente:

1. $0 \in I$, es decir, si 0 pertenece al ideal.
2. Si existen dos polinomios f y $g \in I \rightarrow f + g \in I$. Esta propiedad se denomina ser cerrado para la suma.
3. Si $f \in I$ y existe una expresión $h \in k[x_1, \dots, x_n]$, entonces $h \cdot f \in I$.

Los anillos de polinomios también contienen en ellos ideales y, como cualquier anillo del tipo $k[x_1, \dots, x_n]$, tiene los ideales triviales: el ideal 0 y el anillo $k[x_1, \dots, x_n]$ en si mismo. Los ideales más comunes con los que se trabajará en el presente trabajo están generados por un grupo de polinomios.

Ejemplo 3.1 [7] Considerando el anillo $\mathbb{C}[x]$ y un polinomio $f \in \mathbb{C}[x]$, el conjunto de polinomios generados por f como $\langle f \rangle$, denotaremos el conjunto:

$$\langle f \rangle = \{h \cdot f : h \in \mathbb{C}[x]\}$$

Dicha expresión implica que el ideal generado por f contiene todas las expresiones obtenidas de multiplicar f por una expresión arbitraria h que también pertenece al conjunto de polinomios en la variable x con coeficientes que pertenecen al conjunto $\mathbb{C}$. Por ejemplo, si $f = x^2 + 3x - 2$, entonces el grupo de polinomios generado por f será:

$$\langle f \rangle = \langle x^2 + 3x - 2 \rangle = \{h \cdot (x^2 + 3x - 2) : h \in \mathbb{C}[x]\}$$

Representando en este caso la totalidad de múltiplos de f . Otro ejemplo, tomando $f = x - 1$, el grupo de polinomios generados por f será:

$$\langle f \rangle = \langle x - 1 \rangle = \{h \cdot (x - 1) : h \in \mathbb{C}[x]\}$$

Teorema 3.1 [7] Si k es un cuerpo, entonces cualquier ideal $I \subseteq k[x]$ puede ser escrito como $\langle f \rangle \forall f \in k[x]$.

Demostración: [7] Asumiendo que $I \subseteq k[x]$ es un ideal distinto de cero, elegimos un elemento $f \in I$ tal que ese f es el elemento de menor grado de todos los elementos que forman el ideal. Utilizando el **algoritmo de la división** (desarrollado más a fondo en el apartado 3.2.2) podemos escribir cualquier elemento del ideal, $g \in I$ como:

$$g = q \cdot f + r$$

siendo q y r el cociente y el resto de la división respectivamente. En el caso de que un ideal $I \subseteq k[x]$ esté generado por más de un polinomio, $I = \langle f_1, \dots, f_s \rangle$ la manera más sencilla de encontrar una f única de manera que $\langle f \rangle = \langle f_1, \dots, f_s \rangle$, es decir, un polinomio generador del ideal es utilizando el máximo común divisor (mcd). $\square$

Definición 3.2 [3] Un polinomio $h \in k[x]$ es el máximo común divisor de los polinomios $f, g \in k[x]$ cuando:

1. $h|f$ y $h|g$, es decir, f y g son ambos divisibles por h .
2. Si p es otro polinomio tal que $p|f$ y $p|g$, entonces $p|h$.

Cuando h cumple estas dos condiciones entonces $h = \text{mcd}(f, g)$. Esta definición es idéntica a la desarrollada en el apartado 2.3.1 del presente proyecto, aunque mediante la siguiente proposición se estudiarán más a fondo sus propiedades en el ámbito de los anillos y se relacionarán con los ideales de polinomios.

Proposición 3.1 [3] Con $f, g \in k[x]$, entonces:

1. El $\text{mcd}(f, g)$ existe y es único.
2. El algoritmo de Euclides permite el cálculo directo de estos mcd (Figura 2-2).
3. Si $h \in k[x]$ y $h = \text{mcd}(f, g) \Rightarrow \langle f, g \rangle = \langle h \rangle = \langle \text{mcd}(f, g) \rangle$.

Ejemplo 3.2 [3] Supongamos un ideal $I = \langle f_1, f_2 \rangle \subseteq \mathbb{R}[x]$ en el que $f_1 = x^3 - 2x^2 + x - 2$ y $f_2 = x^5 - 2x^4 - 10$. Tratando de simplificar ambos polinomios los podemos escribir de la forma:

$$f_1 = x^3 - 2x^2 + x - 2 = (x - 2)(x^2 + 1)$$

$$f_2 = x^5 - 2x^4 - 10 = (x - 2)(x^4 + 5)$$

Se observa que $\text{mcd}(f, g) = x - 2$, por tanto, el ideal $I = \langle f_1, f_2 \rangle$ puede ser generado por el simple polinomio $x - 2$. Además, estas propiedades del máximo común divisor se pueden extender y aplicar a ideales generados por más de dos polinomios.

Definición 3.3 [2] Un polinomio $h \in k[x]$, es considerado el máximo común divisor del conjunto de polinomios $S = \{f_1, \dots, f_s\} \subseteq k[x]$ cuando:

1. $h|f \forall f \in S$.
2. Si p es otro polinomio tal que $p|f \forall f \in S \rightarrow p|h$.

Si h satisface dichas condiciones: $h = (\text{mcd}(f_1, \dots, f_s))$.

Este máximo común divisor de varios polinomios cumple, además, las mismas propiedades que las definidas para el máximo común divisor de dos polinomios (Proposición 3.1).

Ejemplo 3.3 [2] Se considera un ideal $I = \langle x^3 - 3x + 2, x^4 - 1, x^6 - 1 \rangle \subseteq k[x]$.

En primer lugar, se sabe por las definiciones vistas anteriormente que el $\text{mcd}(x^3 - 3x + 2, x^4 - 1, x^6 - 1)$ es un generador del ideal. Por tanto, se calcula en primer lugar este máximo común divisor, en este caso se utilizará SageMath para ello:

```
# 1. Se define el Anillo de polinomios en una variable (x).
R.<x>=QQ[]

# 2. Se definen tres polinomios definidos en el enunciado
polinomio1 = x^3 - 3*x + 2
polinomio2 = x^4 - 1
polinomio3 = x^6 - 1

# 3. Se aplica el algoritmo de Euclides visto en el apartado 2.3.1. En este caso, se calcula
primero el mcd de los polinomios 1 y 2 (mcd_1_2) y se vuelve a calcular el mcd entre el
resultado del mismo y el polinomio 3.

def euclides_mcd(a, b):
    while b:
        a, b = b, a % b
    return a

mcd_1_2 = euclides_mcd(polinomio1, polinomio2)
mcd_tres_polinomios = euclides_mcd(mcd_1_2, polinomio3)

# 4. Se devuelve el valor del último mcd calculado
print("El MCD de los tres polinomios es:", mcd_tres_polinomios)
```

Figura 3-1. Cálculo del máximo común divisor con SageMath para el ejemplo 3.3.

Al implementar el código de la Figura 3-1, la plataforma devuelve el valor $x - 1$, por tanto, $\langle x^3 - 3x + 2, x^4 - 1, x^6 - 1 \rangle = \langle x - 1 \rangle$. Se demuestra mediante el presente ejemplo la presencia de un algoritmo para calcular el generador de cualquier $\langle f_1, \dots, f_s \rangle$. El primer paso consta en encontrar ese generador h de $\langle f_1, \dots, f_s \rangle$. Luego, el hecho de que $f \in \langle f_1, \dots, f_s \rangle$ implicará que $f \in \langle h \rangle$, y únicamente necesitaremos del algoritmo de la división para demostrar que $f = qh + r$, donde $\text{gr}(r) < \text{gr}(h)$. De esta manera $f \in \langle h \rangle \Leftrightarrow r = 0$. Siguiendo con el ejemplo, se tratará de demostrar si $x^3 + 4x^2 + 3x - 7 \in \langle x^3 - 3x + 2, x^4 - 1, x^6 - 1 \rangle$.

Se ha demostrado anteriormente que $x - 1$ es generador del ideal, por tanto, se puede replantear la cuestión de manera que se tratará de demostrar si $x^3 + 4x^2 + 3x - 7 \in \langle x - 1 \rangle$. Tomando la expresión

$f = qh + r$ donde $f = x^3 + 4x^2 + 3x - 7$, y $h = x - 1$, obtenemos un cociente $q = x^2 + 5x + 8$ y un resto de 1, por lo que $x^3 + 4x^2 + 3x - 7$ no pertenece al ideal. A partir del apartado 3.2 se utilizará un procedimiento similar para encontrar el generador (base de Gröbner) del ideal y se utilizará el algoritmo generalizado de la división para determinar si un polinomio pertenece o no a un ideal.

3.2 Bases de Gröbner

Mediante el uso de las bases de Gröbner a la hora de optimizar el proceso de encontrar un ideal generador, se pueden resolver cuestiones relativas a propiedades de estos ideales. Por ejemplo, nos permite demostrar que todo ideal tiene una base finita (Bases de Hilbert), nos permite resolver de manera rápida si un polinomio pertenece a un ideal o, como se viene desarrollando a lo largo del trabajo, se puede utilizar a la hora de optimizar la resolución de sistemas de ecuaciones complejos.

3.2.1 Órdenes monomiales

Cuando se trabaja con anillos de polinomios en una sola variable, la ordenación de dichos polinomios se realiza de manera sencilla y depende directamente del grado del término. Además, el concepto de LT (*leading term*) es importante de cara a desarrollar el algoritmo de la división (apartado 3.2.2), es por ello por lo que se requiere definir también cuál es el LT en los casos en el que se trabaja con varias variables, por ejemplo, ¿cómo se ordenarían los monomios en orden ascendente en la expresión $4x^3y^2z^2 - 13x^5y^2 - 2x^3z^2 + 5y^4 + 3z^5$? ¿Cuál sería su LT? [3]. Instintivamente, se tiende a pensar que el término mayor será aquel cuya suma de grados sea mayor, pero observando la expresión planteada, los dos primeros monomios suman un grado de 7. La suma de grados no es suficiente a la hora de trabajar en $k[x_1, \dots, x_n]$, es por ello por lo que se debe establecer un criterio a la hora de ordenar estos monomios. Este criterio se trata del orden monomial. Se utilizará el símbolo $>$ para indicar orden, por ejemplo $x > y$. Cuando se comparan monomios de manera individual se utiliza la letra griega α , para hacer referencia a los exponentes a los que son elevadas las variables. Estos exponentes deben ser números enteros mayores que 0, es decir $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{Z}_{\geq 0}^n$. Teniendo en cuenta esta notación, un monomio se escribirá $x^\alpha = x_1^{\alpha_1} \dots x_n^{\alpha_n} \in k[x_1, \dots, x_n]$.

Definición 3.2.1 [2] Un orden monomial $>$ en $k[x_1, \dots, x_n]$ es una relación $>$ establecida para un conjunto de monomios $x^\alpha = x_1^{\alpha_1} \dots x_n^{\alpha_n}$ con $\alpha \in \mathbb{Z}_{\geq 0}^n$. En él se cumple:

1. $>$ es un orden total. Esto implica que únicamente se puede dar tres casos:

$$x^\alpha < x^\beta, x^\alpha = x^\beta \text{ o } x^\beta < x^\alpha.$$

2. Si se da el caso $\alpha > \beta \Rightarrow \alpha + \gamma > \beta + \gamma, \forall \alpha, \beta, \gamma \in \mathbb{Z}_{\geq 0}^n$.

Definición 3.2.2 [2] Sea $f = \sum_{\alpha} a_{\alpha} x^{\alpha}$ un polinomio en $k[x_1, \dots, x_n]$ y fijándose un orden $>$:

1. El grado de f es: $gr(f) = \max(\alpha \in \mathbb{Z}_{\geq 0}^n : a_{\alpha} \neq 0)$. Esto es, el grado del polinomio corresponderá al mayor exponente cuyo coeficiente sea distinto de 0.
2. El coeficiente líder de f es: $LC(f) = a_{gr(f)} \in k$, es decir, el coeficiente que acompaña al término elevado al mayor exponente.
3. El monomio líder será: $LM(f) = x^{\deg(f)} \in k$.
4. El término líder, mencionado por primera vez en el apartado 2.4, será el resultado de multiplicar el monomio líder por el coeficiente líder: $LT(f) = LC(f) \cdot LM(f)$.

Ejemplo 3.4 [3] Según lo visto en la definición anterior y dado el polinomio $f = 4x^3 + 4x^2 + 3x - 7$:

1. $gr(f) = 3$
2. $LC(f) = 4$
3. $LM(f) = x^3$
4. $LT(f) = 4x^3$

Dentro de las diferentes maneras existentes a la hora de ordenar polinomios, cuando el número de variables es mayor que uno, el más utilizado es el orden lexicográfico (lex), que ordena las variables en orden alfabético. En $k[x_1, \dots, x_n]$ el orden lexicográfico asume que $x_1 > x_2 > \dots > x_n$ y que $x_i < x_i^2 < \dots$, por tanto, juntando ambos órdenes para $n = 2$ se consideraría $1 < x_2 < x_1 < x_2^2 < x_1 \cdot x_2 < x_1^2 < x_2^3 < x_1 \cdot x_2^2 < \dots$. Esto se debe a que el vector $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{Z}_{\geq 0}^n$ en, por ejemplo, x_1^2 es $(2, 0)$, mientras que en $x_1 x_2$ es $(1, 1)$. Aunque la suma de los componentes sea la misma, se compara primero la entrada situada más a la izquierda del vector y se toma la mayor, en este caso $2 > 1$, y, por tanto, $x_1 \cdot x_2 < x_1^2$. En la mayoría de los casos prácticos y en los que se tratarán en el presente trabajo se utilizarán dos o tres variables, en $k[x, y, z]$ en lugar de $k[x_1, x_2, x_3]$, asumiéndose según este orden $x > y > z$. Procedemos a obtener ahora los valores mostrados en el ejemplo 3.4 pero esta vez considerando el polinomio y siguiendo el orden lexicográfico:

$$(3.2.1) \quad g = 2x^3y^2z^2 - 13x^5y^2 - 2x^3z^2 + 5y^4 + 3z^5$$

Los dos elementos a comparar en el polinomio (3.2.1) son $2x^3y^2z^2$ y $13x^5y^2$, ambos de grado 7. El orden lexicográfico nos permite ordenar estos dos polinomios de manera que $x^5y^2 > x^3y^2z^2$ ya que en la x tiene más peso respecto de las otras variables $x > y > z$. De esta manera:

5. $gr(g) = 7$
6. $LC(g) = -13$
7. $LM(g) = x^5y^2$
8. $LT(g) = -13x^5y^2$

En la plataforma de SageMath existe el modo de ordenar lexicográficamente un polinomio mediante el comando 'order=lex' como queda mostrado en la Figura 3-2 ordenando el mismo polinomio que el utilizado en el Ejemplo 3.3.

```
from sage.rings.polynomial.polynomial_ring_constructor import PolynomialRing
# 1. Se define el anillo de polinomios en k [x,y,z] mencionando el orden lexicográfico.
R.<x, y, z> = PolynomialRing(QQ, ['x', 'y', 'z'], order='lex')
# 2. Se define el polinomio g colocados sus monomios en un orden arbitrario
g = 2*x^3*y^2*z^2 - 13*x^5*y^2 - 2*x^3*z^2 + 5*y^4 + 3*z^5
# 3. Se ordena lexicográficamente el polinomio
g_lex = R(g)
# 4. Con la función '.lt' se muestra, además, el término líder (LT)
leading_term = g_lex.lt()
print("Polinomio original:", g)
print("Polinomio ordenado lexicográficamente:", g_lex)
print("Término líder del polinomio ordenado:", leading_term)
```

Figura 3-2. Polinomio ordenado lexicográficamente mediante código en SageMath

Este código imprime en pantalla los valores:

- Polinomio original: $2x^3y^2z^2 - 13x^5y^2 - 2x^3z^2 + 5y^4 + 3z^5$.
- Polinomio ordenado lexicográficamente: $-13x^5y^2 - 2x^3y^2z^2 - 2x^3z^2 + 5y^4 + 3z^5$.
- Término líder del polinomio ordenado: $-13x^5y^2$.

También cabe mencionar la diferencia del orden lexicográfico con el orden lexicográfico graduado. En este último, tienen mayor valor aquellos monomios que, con el mismo grado total, tienen más variables, es decir $xy > x^2$ (en orden lexicográfico obtendría mayor peso la x y, por tanto, $xy < x^2$). En este caso, los vectores α serían (1,1) para xy y (2,0) para la x . En el orden lexicográfico inverso, por el contrario, se compara en primer lugar la entrada de la derecha, en este caso $1 > 0$ por tanto $xy > x^2$.

3.2.2 Algoritmo de la división

En el apartado 3.1.3 se desarrolló cómo el algoritmo de la división podía ser utilizado para resolver cuestiones de pertenencia a un ideal con polinomios en una variable. Para abordar este problema cuando hay más de una variable, en este apartado se tratará el algoritmo para polinomios en $k[x_1, \dots, x_n]$. En los casos a tratar, se dividirá un polinomio $f \in k[x_1, \dots, x_n]$ entre $f_1, \dots, f_s \in k[x_1, \dots, x_n]$. Esto conlleva expresar el polinomio f de la forma: $f = q_1f_1 + \dots + q_sf_s + r$, siendo las q los cocientes y r el resto de la propia división. La idea básica del algoritmo es la de cancelar el LT de f (establecido tras fijar un orden monomial) multiplicándolo por un divisor f_i y restando. Para entender de manera más eficiente el funcionamiento del algoritmo de la división, se introducirán varios ejemplos antes de establecer la definición general.

Ejemplo 3.5 [2] Se divide $f = xy^2 + 1$ entre $f_1 = xy + 1$ y $f_2 = y + 1$ utilizando el orden monomial lexicográfico ($x > y$). Para realizar la división se seguirá el siguiente esquema:

$$\begin{array}{l} q_1: \\ q_2: \\ f_1: \\ f_2: \end{array} \left| \begin{array}{c} \hline f \end{array} \right.$$

En primer lugar, antes de proceder a obtener los cocientes de la división, se deben identificar los términos líderes de los tres polinomios que intervienen:

$$LT(f) = xy^2$$

$$LT(f_1) = xy$$

$$LT(f_2) = y$$

De esta manera, se procede de forma similar a la división de enteros usual, multiplicamos por un factor (cociente) de manera que el divisor se aproxime lo máximo posible al dividendo, sin pasarse, por ejemplo, el primer cociente del ejemplo en cuestión sería y .

$$\begin{array}{r}
 q_1: y \\
 q_2: \\
 xy + 1 \left| \begin{array}{l} xy^2 + 1 \\ xy^2 + y \\ \hline 0 - y + 1 \end{array} \right.
 \end{array}$$

Una vez, establecido el primer cociente, se repite el proceso con $-y + 1$. Para esta operación se dividirá entre f_2 ya que $LT(-y + 1) = -y$ no es divisible entre $LT(f_1) = xy$:

$$\begin{array}{r}
 q_1: y \\
 q_2: -1 \\
 xy + 1 \left| \begin{array}{l} xy^2 + 1 \\ xy^2 + y \\ \hline 0 - y + 1 \\ 0 - y - 1 \\ \hline 2 \end{array} \right.
 \end{array}$$

Llegados a este punto, ninguno el número 2 no es divisible entre ninguno de los dos divisores planteados en el ejercicio, por tanto, el resto será igual a 2. Escribiremos el polinomio f de la forma:

$$f = xy^2 + 1 = y \cdot (xy + 1) + (-1) \cdot (y + 1) + 2$$

Realizando la división en los casos en los que existe más de una variable, tras realizar la primera operación y obtener el primer cociente, es posible que el término líder polinomio que resulte de la resta $f - q_1 f_1$ no sea divisible entre ninguno de los términos líderes de f_1 y f_2 . En estos casos, el término líder del polinomio resultante de la resta pasará a formar parte del resto, dando lugar así a un nuevo término líder. Este paso se realizará tantas veces como sea necesario para obtener un LT que permita continuar dividiendo.

Teorema 3.2.2 [2] Dados una S-tupla de polinomios $F = (f_1, \dots, f_s) \in k[x_1, \dots, x_n]$ bajo un orden monomial lexicográfico, cualquier polinomio $f \in k[x_1, \dots, x_n]$ se puede escribir de la forma:

$$(3.2.2) \quad f = q_1 f_1 + \dots + q_s f_s + r$$

donde $q_i, r \in k[x_1, \dots, x_n]$. En cuanto al resto r , deberá resultar de una combinación lineal de coeficientes en k o monomios de los cuales ninguno es divisible por cualquier $LT(f_1), \dots, LT(f_s)$. Cualquier resultado que difiera de este caso, significaría que la división no se ha realizado correctamente.

Del mismo modo que ocurría al trabajar con una variable, podemos determinar si un polinomio f es un elemento de un ideal $I = \langle f_1, \dots, f_s \rangle$ dividiendo f entre $(f_1, \dots, f_s)$. Si el resto de la división es igual a 0, se puede concluir que el polinomio f pertenece al ideal.

Ejemplo 3.6 [3] Se da polinomio $f = xy^2 - x$ y un ideal generado por $I = \langle xy - 1, y^2 - 1 \rangle$ y asumiendo un orden lexicográfico. El resultado de dividir f entre $F = (f_1, f_2) = (xy - 1, y^2 - 1)$ es el siguiente:

$$f = y \cdot (xy - 1) + 0 \cdot (y^2 - 1) + (-x + y)$$

Vemos que el resto es igual a $-x + y$ por lo que, esta división no es suficiente para demostrar que f pertenece al ideal. Se debe tener en cuenta que cambiar el orden monomial puede cambiar el resultado de la división. Esta misma división asumiendo el orden lexicográfico inverso ($x < y$) afirmaría la pertenencia de f al ideal. En este caso, $F = (f_1, f_2) = (y^2 - 1, xy - 1)$ y el resultado de la división sería $f = x \cdot (y^2 - 1) + 0 \cdot (xy - 1) + 0$ y f pertenecería, por tanto, al ideal.

El ejemplo 3.6 muestra de manera clara como el algoritmo de la división puede resultar confuso a la hora de estudiar el problema de pertenencia a un ideal. Para afirmar con seguridad que un polinomio pertenece a un ideal, se debería repetir la división varias veces utilizando diferentes órdenes monomiales. Un sistema algebraico computacional podría ayudar a la hora de agilizar este proceso, no obstante, son las bases de Gröbner las que tratan el problema de finitud en cuanto a la pertenencia a un ideal.

3.2.3 Ideales monomiales y el lema de Dickson

Dentro del ámbito de los ideales monomiales, el lema de Dickson resulta una herramienta fundamental para demostrar la existencia de finitos generadores de un ideal, así como una condición crucial para generar la existencia de las bases de Hilbert (apartado 3.2.4). Comenzaremos definiendo un ideal monomial:

Definición 3.2.3 [2] Un ideal monomial es un ideal $I \subseteq k[x_1, \dots, x_n]$ si existe un subconjunto llamado $A \subseteq \mathbb{Z}_{\geq 0}^n$ tal que I abarca todos los polinomios cuya forma deriva de la suma $\sum_{\alpha \in A} h_\alpha \cdot x^\alpha$, $h_\alpha \in k[x_1, \dots, x_n]$. En ese caso, este ideal se escribiría $I = \langle x^\alpha | \alpha \in A \rangle$.

A grandes rasgos, un ideal monomial no deja de ser un ideal cuyo sistema de generadores está conformado por monomios. Un ejemplo de un ideal monomial vendría dado por $I = \langle x^3y, x^3y^4, xy \rangle \subseteq k[x, y]$. Una vez desarrollado este concepto podemos introducir el lema de Dickson.

Teorema 3.2.3 [2] Un ideal monomial $I = \langle x^\alpha | \alpha \in A \rangle \subseteq k[x_1, \dots, x_n]$ se puede escribir $I = \langle x^{\alpha^{(1)}}, \dots, x^{\alpha^{(s)}} \rangle$ donde $\alpha \in A$. En concreto, el ideal I tiene una base finita.

Demostración [2] Si $n = 1$ (siendo n en este caso el número de variables) entonces I es generado por monomios del tipo x^α siendo $\alpha \in A \subseteq \mathbb{Z}_{\geq 0}^n$ (exponente perteneciente al subconjunto A dentro del cuerpo de los números enteros positivos). Tomemos ahora β como el elemento más pequeño de $A \subseteq \mathbb{Z}_{\geq 0}^n$, lo que implica que $\beta \leq \alpha \forall \alpha \in A$ de manera que todos los generadores de x^α serán divisibles por x^β , de lo que se deduce que $I = \langle x^\beta \rangle$.

En el caso e que $n \geq 1$ y el teorema se cumpla para $n - 1$, escribiremos las variables de la siguiente manera: $x_1, \dots, x_{n-1}, y$ de modo que los monomios en $k[x_1, \dots, x_{n-1}, y]$ se pueden escribir como $x^\alpha y^m$: $\alpha = (\alpha_1, \dots, \alpha_{n-1}) \in \mathbb{Z}_{\geq 0}^{n-1}$ y $m \in \mathbb{Z}_{\geq 0}$. Supongamos que $I \subseteq k[x_1, \dots, x_{n-1}, y]$ es un ideal monomial. Para encontrar los generadores de I , se da un ideal J en $k[x_1, \dots, x_{n-1}]$ generado por los monomios x^α para los que $x^\alpha y^m \in I$. Si, entonces, J es un ideal monomial significa que un número determinado y finito de monomios x^α generan J ($J = \langle x^{\alpha^{(1)}}, \dots, x^{\alpha^{(s)}} \rangle$), entendiéndose el ideal J como una proyección de I en $k[x_1, \dots, x_{n-1}]$. Ahora, para cada generador $x^{\alpha^{(i)}}$ de J identificamos el mayor exponente m tal que $x^{\alpha^{(i)}} y^{m(i)} \in I$, entonces, para cada $\ell: 0 < \ell < m - 1$ consideramos el ideal $J_\ell \subseteq k[x_1, \dots, x_{n-1}]$ generado por los monomios x^β tales que $x^\beta y^\ell \in I$. J_ℓ se considera una ‘porción’ de I generado por

monomios que contienen la variable y elevada a la ℓ potencia. Aplicando lógica inductiva J_ℓ tendrá un conjunto de monomios generadores finitos del tipo $J_\ell = \langle x^{\alpha_\ell(1)}, \dots, x^{\alpha_\ell(s_\ell)} \rangle$. $\square$

Ejemplo 3.7 [2] Consideremos el ideal $\langle x^4y^2, x^3y^4, x^2y^5 \rangle$. En este caso, la proyección $J = \langle x^2 \rangle$ (x^2 divide a todos los monomios en $k[x]$). Por otra parte, vemos que el máximo exponente que eleva la variable y es 5 por tanto, $m = 5$. De este modo las ‘porciones’ J_ℓ generadas por los monomios que contienen y^ℓ comprenderán $0 \leq \ell \leq 4 = m - 1$ de esta manera para $\ell = 0$ y $\ell = 1$, $J_0 = J_1 = \{0\}$. Para los casos en los que aparece la variable y elevada a exponentes mayores que 1, obtenemos $J_2 = J_3 = \langle x^4 \rangle$ y para $J_4 = \langle x^3 \rangle$, por tanto, el ideal I estará generado por $I = \langle x^4y^2, x^3y^4, x^4y^3, x^2y^5 \rangle$.

El lema de Dickson ofrece una solución al problema finitud en cuanto a los generadores de un ideal monomial, ofreciendo también un método alternativo para resolver el problema de pertenencia a un ideal monomial.

3.2.4 Teorema de las Bases de Hilbert y Bases de Gröbner

Tras haber alcanzado la condición fundamental de finitud de los conjuntos de monomios generadores, el Teorema de las Bases de Hilbert ofrecerá un enfoque más sistemático y afianzado a la hora de estructurar estos conjuntos. Estas bases ofrecen una representación de un conjunto de polinomios, utilizando un orden monomial específico, que proporcionan un cimiento sólido sobre el que posteriormente desarrollar el cálculo de las bases de Gröbner.

En el presente documento se ha visto mediante el algoritmo de la división (apartado 3.2.2) el importante papel llevado a cabo por los términos líderes. Dentro del ámbito de los ideales, existe también el concepto de ideal de términos líderes de un ideal. Dado un ideal $I = \langle f_1, \dots, f_s \rangle \Rightarrow \langle LT(I) \rangle$ sería el ideal generado por los elementos de $LT(I)$, teniendo en cuenta que $\langle LT(f_1), \dots, LT(f_s) \rangle \neq \langle LT(I) \rangle$, se puede afirmar que $LT(f_i) \in LT(I) \subseteq \langle LT(I) \rangle$. Veámoslo mediante un ejemplo:

Ejemplo 3.8 [2] Dado un ideal $I = \langle x^3 - 2xy, x^2y - 2y^2 + x \rangle$, siendo $f_1 = x^3 - 2xy$ y $f_2 = x^2y - 2y^2 + x$, y un orden lexicográfico graduado (visto al final del apartado 3.2.1) en los monomios $k = [x, y]$ se cumple que:

$$x \cdot (x^2y - 2y^2 + x) - y \cdot (x^3 - 2xy) = x^2 : x^2 \in I$$

Por tanto, $x^2 = LT(x^2) \in \langle LT(I) \rangle$. No obstante, x^2 no es divisible ni por $LT(f_1) = x^3$, ni por $LT(f_2) = x^2y$, por tanto $x^2 \notin \langle LT(f_1), LT(f_2) \rangle$.

Proposición 3.2.4 [3] Sea un ideal $I \subseteq k[x_1, \dots, x_n]$:

1. $\langle LT(I) \rangle$ es un ideal monomial.
2. $\exists g_1, \dots, g_t \in I / \langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle$.
3. Denominamos $G = \{g_1, \dots, g_t\}$ a los generadores del ideal, de manera que $\langle G \rangle = \langle g_1, \dots, g_t \rangle \Rightarrow \langle LT(G) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle$.

Demostración [2] Los monomios líderes $LM(g)$ de elementos $g \in I$ generan el ideal monomial $\langle LM(g) | g \in I \rangle$. Debido a que $LM(g)$ y $LT(g)$ difieren de una constante distinta de 0, este ideal equivale $\langle LM(g) | g \in I \rangle = \langle LT(I) \rangle$, por tanto, $\langle LT(I) \rangle$ es un ideal monomial.

Además, si $\langle LT(I) \rangle$ está generado por los monomios $LM(g)$ para $g \in I$, el lema de Dickson nos confirma que $\langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle$ está generado por finitos $g_1, \dots, g_t$. $\square$

Con todo lo desarrollado anteriormente, y aplicando el algoritmo de la división podemos probar la existencia de un conjunto finito de generadores para todo ideal polinomial.

Teorema 3.2.4 (Teorema de las Bases de Hilbert) [3] Todo ideal $I \subseteq k[x_1, \dots, x_n]$ tiene un conjunto finito de generadores. Es decir, $I = \langle G \rangle$ para algún $G = \{g_1, \dots, g_t\} \in I$.

Demostración Si tomamos el ideal $I = \{0\}$, nuestro conjunto generador es $\{0\}$, claramente, finito. Si nuestro ideal I contiene polinomios distintos de 0, entonces un conjunto generador del tipo $g_1, \dots, g_t$ para el ideal I se podrá construir como se indica a continuación.

El ideal de polinomios distinto de 0 I tiene un ideal formado por sus términos principales $\langle LT(I) \rangle$, que habrá sido generado por un conjunto del tipo $g_1, \dots, g_t \in I \Rightarrow \langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle \Rightarrow I = \langle g_1, \dots, g_t \rangle$. Es decir, el ideal formado por los LT de los generadores pertenece al ideal formado por los LT del ideal. Esto implica, que cualquier polinomio f perteneciente al ideal I al ser dividido por el conjunto de generadores $g_1, \dots, g_t$ resultará expresión similar a la (3.2.2), es decir, una combinación lineal de estos generadores:

$$f = q_1 g_1 + \dots + q_t g_t + r$$

Como los $g_1, \dots, g_t$ son generadores del ideal, el resto deberá ser necesariamente $r = 0$. Eliminando la r la expresión anterior queda:

$$f = q_1 g_1 + \dots + q_t g_t \in \langle g_1, \dots, g_t \rangle$$

Se observa en la expresión que f se puede escribir como combinación lineal de dichos generadores, que, a su vez, como se desarrolla en el teorema 3.2.4, pertenecen al ideal, quedando así demostrado que $I \subseteq \langle g_1, \dots, g_t \rangle$. $\square$

Más allá de quedar resueltos los problemas de pertenencia y finitud de un ideal, estas bases generadoras $\{g_1, \dots, g_t\}$ tienen la propiedad especial de $\langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle$, nombrándolas según la definición 3.2.4.

Definición 3.2.4 (Bases de Gröbner) [2] Establecido un orden monomial en un anillo de polinomios $k[x_1, \dots, x_n]$, un subconjunto finito $G = \{g_1, \dots, g_t\}$ de un ideal $I \subseteq k[x_1, \dots, x_n]$ se denomina base de Gröbner o base estándar si:

$$(3.2.4) \quad \langle LT(g_1), \dots, LT(g_t) \rangle = \langle LT(I) \rangle.$$

Este conjunto $\{g_1, \dots, g_t\} \subseteq I$ será base de Gröbner si y solo si el LT de cualquier término del ideal I es divisible por uno de los $LT(g_i)$.

Ejemplo 3.9 [3] Sea un ideal $I \subseteq \mathbb{C}[x, y]$ el ideal generado por $I = \langle f_1, f_2 \rangle$, donde $f_1 = x^2 y - 1$ y $f_2 = x y^2 - x$ y utilizando el orden lexicográfico; el objetivo es determinar si $F = \{f_1, f_2\}$ son bases de Gröbner del ideal I . Como se menciona en la definición 3.2.4, F será base de Gröbner si el LT de cualquier elemento de I es divisible por el LT de al menos uno de los polinomios en la base. Por definición, teniendo en cuenta la expresión 3.2.4:

$$\langle LT(f_1), LT(f_2) \rangle = \langle LT(I) \rangle$$

Además, cualquier elemento perteneciente al ideal I se puede escribir de la forma:

$$\langle f_1, f_2 \rangle = \{h_1 \cdot f_1 + h_2 \cdot f_2 : h_i \in \mathbb{C}[x, y]\}$$

Considerando un polinomio cualquiera $x^2 - y$, pertenece al ideal porque cumple:

$$y f_1 + (-x) f_2 = y(x^2 y - 1) + (-x)(x y^2 - x) = (x^2 y^2 - y) + (-x^2 y^2 + x^2) = x^2 - y$$

No obstante, el $LT(x^2 - y) = x^2$ no es divisible ni por $x^2 y$, ni por $x y^2$, que son los términos líderes de f_1 y f_2 respectivamente, por lo que F no es base de Gröbner del ideal. Este mismo caso se puede comprobar de manera automática utilizando la función '`groebner_basis()`' en la plataforma Sagemath utilizando el siguiente código con los polinomios indicados en el ejemplo. La función `groebner_basis()` no sólo es capaz de comprobar si F es o no base de Gröbner, sino que además puede calcularlas:

```
# 1. Al igual que en la mayoría de los códigos utilizados en el trabajo, en primer lugar, se introduce el anillo de polinomios mencionando el orden a seguir.
R.<x, y> = PolynomialRing(QQ, ['x', 'y'], order='lex')

# 2. Se define a continuación los polinomios del ejemplo 3.9 que definen la base I
f1 = x^2*y - 1
f2 = x*y^2 - x
ideal = R.ideal([f1, f2])

# 3. Mediante la función groebner_basis() se calcula la base de Gröbner del ideal
gb = ideal.groebner_basis()

# 4. Verificar si los polinomios f1 y f2 están en la base de Gröbner (CASO AFIRMATIVO)
if f1 in gb and f2 in gb:
    print("f1 y f2 son base de Gröbner de I.")

# 5. Verificar si los polinomios f1 y f2 están en la base de Gröbner (CASO NEGATIVO) y se proponen dos f que si conformen una base de Gröbner para el ideal
else:
    print("f1 y f2 NO son base de Gröbner de I, dos bases de Gröbner del ideal pueden ser:")
    new_gb = R.ideal(gb).groebner_basis()
    print(new_gb[0])
    print(new_gb[1])
```

Figura 3-3 Comprobación mediante la plataforma SageMath del ejemplo 3.9.

Tal y como se demuestra en el ejemplo, F no es base de Gröbner, por tanto, el código implementado en SageMath devuelve “f1 y f2 NO son base de Gröbner” y propone como bases los polinomios $x^2 - y$ y $y^2 - 1$.

Proposición 3.2.5 [3] Dados un ideal $I \subseteq k[x_1, \dots, x_n]$, $G = \{g_1, \dots, g_t\}$ una base de Gröbner de I y una $f \in k[x_1, \dots, x_n]$, existe un único resto $r \in k[x_1, \dots, x_n]$ resultante de dividir f entre G que satisface las siguientes dos propiedades:

1. Ningún elemento de r es divisible por cualquiera de los términos líderes de G , $LT(G)$.
2. $\exists g \in I : f = g + r$, siendo $g = q_1g_1 + \dots + q_tg_t \in I$. Es decir, f se puede escribir como la suma de la combinación lineal de los elementos de que forman la base de Gröbner y el resto de la división anteriormente mencionada.

Demostración Como se indica en la proposición anterior, se puede escribir $f = g + r$ para algún $g \in I$. Supongamos, al contrario, que también podemos escribir $f = g' + r'$ para algún $g' \in I$ donde $r' \neq r$. Tenemos, por tanto:

$$g + r = g' + r' \Rightarrow g - g' = r - r'$$

Como $g, g' \in I \Rightarrow g - g' \in I, r - r' \in I$, lo que implica que:

$$LT(r - r') \in \langle LT(I) \rangle = \langle LT(G) \rangle$$

Esta última expresión implica que $LT(r - r')$ es divisible entre alguno de los $LT(g_i)$. Esto es imposible debido a que ningún término perteneciente a un resto (r, r') es divisible por ninguno de los $LT(G)$. Por tanto, $r - r' = 0$ quedando así la unicidad del resto demostrada. $\square$

Ejemplo 3.10 [3] Siguiendo el caso del ejemplo anterior con $F = \{f_1, f_2\} = \{x^2 - 1, xy^2 - x\}$, se demostró en el ejemplo 3.9 F no conformaba una base de Gröbner para $I = \langle x^2 - 1, xy^2 - x \rangle$. Como no se trata de una base de Gröbner, no se garantiza que el resto sea único. Para ejemplificarlo, dividamos $f = x^5y^2 - x^4 - x^2y + y^3$ entre F obteniendo el siguiente resultado.

$$f = q_1f_1 + q_2f_2 + r_1$$

$$f = (x^3y + x - 1)(x^2y - 1) + (0)(xy^2 - x) + (-3x^4 + x + y^3 - 1)$$

Observamos que el resto es $-3x^4 + x + y^3 - 1$. En cambio, si cambiamos el orden de los polinomios en F , siendo $F^* = \{xy^2 - x, x^2 - 1\} = (f_2, f_1)$ el resultado del valor de r cambia, demostrando así que F no produce un resto único y reforzando que F no es una base de Gröbner.

Definición 3.2.5 [2] Se parte de un ideal $I \subseteq k[x_1, \dots, x_n]$. Denominamos variedad afín $V(I)$ del ideal al conjunto

$$V(I) = \{(a_1, \dots, a_n) \in k^n : f_i(a_1, \dots, a_n) = 0 \forall f \in I\}.$$

Para un ideal $I = \langle f_1, \dots, f_s \rangle$ su variedad $V(I)$ coincidirá con la variedad formada por los propios polinomios que definen el ideal, es decir, $V(I) = V(f_1, \dots, f_s)$. Esto implica que las soluciones de del sistema de ecuaciones polinómicas definido por $f_1, \dots, f_s$ son las mismas que las soluciones del ideal I , generado por estos polinomios. Incluso para un ideal formado por infinitos polinomios diferentes, el conjunto $V(I)$ seguirá estando definido por finitas ecuaciones polinomiales.

Consideremos ahora el ideal $I = \langle g_1, g_2 \rangle = \langle x + z, y - z \rangle$. g_1 y g_2 forman una base de Gröbner utilizando el orden lexicográfico en $\mathbb{R}[x, y, z]$ tal que $(x > y > z)$. De esta manera, los términos líderes de cada elemento diferente de cero de I pertenecen al ideal $\langle LT(g_1), LT(g_2) \rangle = \langle x, y \rangle$ o lo que es lo mismo, son divisibles tanto por x como por y . Para demostrarlo, suponemos, al contrario que existe un $f \in I$ tal que $LT(f) \notin \langle LT(g_1), LT(g_2) \rangle = \langle x, y \rangle$, por lo que $LT(f)$ no sería divisible ni por x ni por y .

Considerando entonces $f = Ag_1 + Bg_2 \in I$ suponiendo que $f \neq 0$ y que su LT no es divisible ni por x ni por y para tratar de llegar a una contradicción, y verificar así lo desarrollado en el párrafo anterior. Debido al orden lexicográfico establecido, x e y no aparecen en ningún término de f lo que indica que $f \in \mathbb{R}[z]$, es decir, está formado por monomios que únicamente contienen la variable z . Por otra parte, los elementos del ideal I se anularán siempre en la variedad $\mathcal{V}(I) = \mathcal{V}(x + z, y - z)$. En el caso particular del ejemplo 3.10, trabajamos con tres variables $[x, y, z]$ y el ideal I está generado por dos polinomios, por lo que $\mathcal{V}(I)$ conformará una recta. Si igualamos a 0 ambos polinomios del ideal obtenemos que respecto a $g_1 \Rightarrow x + z = 0 \Rightarrow x = -z$ y respecto a $g_2 \Rightarrow y - z = 0 \Rightarrow y = z$, obtenemos que la recta donde se anula la variedad es la recta dada por $(-t, t, t)$ siendo t un parámetro real. La única manera de que un polinomio f se anule en la recta $(-t, t, t)$ es que f sea el polinomio 0, hecho contrario del que parte la presente demostración. Como suponiendo el caso contrario se ha llegado a una contradicción, lo planteado en el párrafo anterior es cierto. Cabe destacar que, de la misma manera que ocurre con el algoritmo de la división, una base de Gröbner sobre un orden monomial puede no serlo aplicando un orden distinto. Por ejemplo, en el caso mostrado en el último párrafo del ejemplo 3.10, si utilizamos el orden monomial $z > y > x$, el conjunto $\{g_1, g_2\}$ no es una base de Gröbner para I .

Otra consecuencia por destacar de las bases de Hilbert pertenece al ámbito de la geometría y está fuertemente relacionada con la resolución de sistemas de ecuaciones polinomiales. En el apartado anterior, se consideran las variedades afines como los conjuntos de soluciones de un sistema de ecuaciones compuesto por los polinomios que conforman el ideal.

3.2.5 Propiedades de las bases de Gröbner y S-polinomios.

Definición 3.2.6 [2] Sean f, g dos polinomios en $k[x_1, \dots, x_n]$, multidegree (multideg) el grado de los LT y $(\alpha_1, \dots, \alpha_n)$ el vector en el cual α_i es el grado del monomio más alto respecto a un polinomio la variable x_i :

1. Si $\text{multideg}(f) = \alpha$ y $\text{multideg}(g) = \beta$, entonces $\gamma = (\gamma_1, \dots, \gamma_n)$ donde $\gamma_i = \max(\alpha_i, \beta_i)$ para cada i . Denominamos x^γ al mínimo común múltiplo del monomio líder de f y de g .
2. El S-polinomio de f y g es la combinación:

$$S(f, g) = \frac{x^\gamma}{LT(f)} \cdot f - \frac{x^\gamma}{LT(g)} \cdot g$$

Por ejemplo, dados $f = x^3y - xy^3 - y^5$ y $g = xy^2 - y^4$ en $\mathbb{R}[x, y]$ utilizando el orden lexicográfico observamos que $LM(f) = x^3y$ y $LM(g) = xy^2$, y que dentro de ambos LM el mayor grado de las variables x y y son 3 y 2 respectivamente, por lo que $\gamma = (3, 2) \Rightarrow x^\gamma = x^3y^2$. Sustituyendo los valores en la expresión anterior:

$$\begin{aligned} S(f, g) &= \frac{x^3y^2}{x^3y} (x^3y - xy^3 - y^5) - \frac{x^3y^2}{xy^2} \cdot (xy^2 - y^4) \\ &= xy^4 - y^6 + x^2y^4. \end{aligned}$$

Al realizar las operaciones del S-polinomio observamos que los LT de ambos polinomios se cancelan, siendo este el principal propósito de los S-polinomios y jugando un papel crucial en el desarrollo de las bases de Gröbner. Antes de pasar al desarrollo del algoritmo para la construcción de las bases de Gröbner es necesario introducir el lema 3.2.5. Dado que trabajamos con polinomios que comparten un determinado grado (δ), la única manera de sumar estos polinomios y que tengan un grado estrictamente menor que δ se consigue tratando de cancelar o restar sus términos principales.

Lema 3.2.5 [3] Supongamos que tenemos un sumatorio del tipo $\sum_{i=1}^s c_i f_i$, donde el máximo $gr(f) = \delta \in \mathbb{Z}_{\geq 0}^n$ para todos los i . Si el $gr(\sum_{i=1}^s c_i f_i) < \delta$, implica que $\sum_{i=1}^s c_i f_i$ es una combinación lineal con coeficientes en K de los S-polinomios $S(f_j, f_k)$ para $1 \leq j, k \leq s$ donde $gr(S(f_j, f_k)) < \delta$. Esto es:

$$\sum_{i=1}^s c_i f_i = \sum_{j,k} c_{j,k} S(f_j, f_k).$$

Demostración [3] El lema trata de establecer que, si la cancelación entre términos principales de polinomios del mismo grado ocurre, esta cancelación es equivalente a la cancelación que se da en los S-polinomios (Definición 3.2.6). En primer lugar, c_i incluye cualquier constante de f_i , lo que permite asumir que $LT(f_i)$ es un monomio y que el $gr(f_i) = \delta$. Además, podemos concluir: $\forall i \neq j \Rightarrow x^\gamma = \delta$, por tanto f_i y f_j tienen el mismo monomio líder, cumpliéndose $S(f_i, f_j) = f_i - f_j$.

De igual forma, si la suma $\sum_{i=1}^s c_i f_i$ tiene un grado estrictamente menor que δ , entonces la suma de los coeficientes líderes debe ser igual a 0 ($\sum_{i=1}^s c_i = 0$). Sumando por partes, establecemos $f_{s+1} = 0$ y $C_i = c_1 + c_2 + \dots + c_i$ (lo que implica $C_s = \sum_{i=1}^s c_i = 0$) para conseguir

$$\sum_{i=1}^s c_i f_i = (f_{s+1} - f_s) C_s - \sum_{i=1}^{s-1} (f_{i+1} - f_i) C_i = (0 - f_s) \cdot 0 - \sum_{i=1}^{s-1} (f_{i+1} - f_i) C_i = - \sum_{i=1}^{s-1} S(f_{i+1}, f_i) C_i.$$

que es una combinación lineal de S-polinomios como se pretendía demostrar. □

Esto es, si tenemos que $gr(\sum_{i=1}^s f_i) < \delta$ donde el $gr(f_i) = \delta \forall i$, la cancelación de términos principales es la misma cancelación que se da con los S-polinomios, como se demuestra en

$$\sum_{i=1}^s f_i = \sum_{i=1}^{s-1} S(f_{i+1}, f_i).$$

Los polinomios en el sumatorio de la parte izquierda de la igualdad son todos de grado δ , mientras que los polinomios en el sumatorio de la derecha esta conformado por polinomios de grado estrictamente menor que δ debido a la cancelación llevada a cabo en el S-polinomio.

3.2.6 Algoritmo de Buchberger

Teorema 3.2.6 (Criterio de Buchberger) [8]. Sea I un ideal polinomial, $G = \{g_1, \dots, g_t\}$ es una base de Gröbner de I si y solo si para todas las parejas $i \neq j$ el resto resultante de dividir $S(g_i, g_j)$ entre G es 0.

Un esquema de la demostración sería el siguiente: supongamos que G es una base de Gröbner y que, por tanto, cualquier polinomio en el ideal generado por G puede ser reducido a cero respecto a G . Consecuentemente, los S-polinomios generados por pares de polinomios en G también se reducirán a cero, debido a que son producto de una combinación lineal de polinomios en G , de esta manera, si los S-polinomios generados por los pares de polinomios en G se reducen a cero respecto a G , entonces G es necesariamente una base de Gröbner.

Ejemplo 3.10 [2] Considerando el ideal $I = \langle y - x^2, z - x^3 \rangle$, se procede a demostrar, utilizando el criterio de Buchberger si la base $G = \{y - x^2, z - x^3\}$ es una base de Gröbner utilizando el orden lexicográfico $x < y < z$. Para ello, en primer lugar, se hallará conforme a la Definición 3.2.6 el S-polinomio de los polinomios que conforman el ideal. Teniendo en cuenta el orden establecido, obtenemos que $\gamma = (0, 1, 1) \Rightarrow x^\gamma = yz$, por tanto:

$$S(y - x^2, z - x^3) = \frac{yz}{y}(y - x^2) - \frac{yz}{z} \cdot (z - x^3) = -zx^3 + yx^3.$$

Si al resultado se le aplica el algoritmo de la división (en este caso se dividiría el resultado obtenido del S-polinomio entre los polinomios $(y - x^2, z - x^3)$) se obtiene:

$$-zx^3 + yx^3 = x^3 \cdot (y - x^2) + (-x^2) \cdot (z - x^3) + 0$$

por lo que G sería una base de Gröbner de I . Cabe añadir, de nuevo, que cambiar el orden lexicográfico puede suponer la variación del resultado. En este caso, si el orden lexicográfico utilizado fuese $x > y > z$, G no conformaría una base de Gröbner.

El criterio de Buchberger resulta útil para determinar si una base es o no de Gröbner. No obstante, ni en el mencionado criterio, ni en el apartado anterior (donde queda demostrado como todo ideal en $k[x_1, \dots, x_n]$ tiene una base de Gröbner) se profundiza en cuanto a la obtención de estas bases. De esta necesidad surge el propósito del algoritmo en cuestión. Utilizando el algoritmo de Buchberger se tratará de construir una base de Gröbner a partir de un ideal I . Una manera intuitiva de producir una base de este tipo es tratando de ampliar el conjunto generador del ideal original añadiendo más polinomios a I . No obstante, añadir un generador nuevo que no cumpla las características requeridas puede resultar contraproducente. Veámoslo con el siguiente ejemplo.

Ejemplo 3.11 [2] Consideramos un ideal $I = \langle x^3 - 2xy, x^2y - 2y^2 + x \rangle$ y un orden lexicográfico graduado. Realizando las mismas operaciones realizadas en el ejemplo 3.11, obtenemos $S(f_1, f_2) = -x^2$, que cuyo LT no es divisible entre $LT(f_1)$ ni $LT(f_2)$, por tanto $LT(S(f_1, f_2)) = -x^2 \notin \langle LT(f_1), LT(f_2) \rangle$. Llegados a este punto, ¿qué generadores del ideal podemos añadir para convertir I en una base de Gröbner? El resto que se obtiene al dividir $S(f_1, f_2)$ entre f_1 y f_2 es $-x^2$ (no es divisible). Este resto se debe incluir en el conjunto de generadores como nuevo generador, es decir $f_3 = -x^2 \Rightarrow F = (f_1, f_2, f_3)$. Una vez ampliado el conjunto F , se dividen los S-polinomio resultantes entre el conjunto F con tal de comprobar el resto resultante. En este caso se dividiría entre $S(f_1, f_2)$ entre F (resto 0) y $S(f_1, f_3)$ entre F (resto $-2xy$). Como el resto sigue sin dar 0, se añadiría $f_4 = -2xy$ al conjunto de

generadores y repetiríamos el proceso. Para este caso particular, desarrollado en su totalidad en [2], serían necesario añadir hasta 3 generadores resultantes del resto de la división de los S-polinomios, de manera que la base de Gröbner se conformaría con

$$\{f_1, f_2, f_3, f_4, f_5\} = \{x^3 - 2xy, x^2y - 2y^2 + x, -x^2, -2xy, -2y^2 + x\}$$

Con el ejemplo anterior se demuestra como una base F se puede ampliar añadiéndole sucesivamente los restos distintos de 0 resultantes de la división $S(f_i, f_j)$ entre F hasta que conforme una base de Gröbner, explicando así el funcionamiento del algoritmo de Buchberger para computar bases de Gröbner.

Teorema 3.2.7 (Algoritmo de Buchberger) Dado un ideal polinomial $I = \langle f_1, \dots, f_s \rangle \neq \{0\}$, una base de Gröbner para I puede ser construida realizando el siguiente bucle un número finito de veces:

1. Entrada: $F = (f_1, \dots, f_s)$
2. Obtenemos: una base de Gröbner $G = (g_1, \dots, g_t)$ para I , con $F \subseteq G$.
3. $G := F$
4. Se repite el siguiente bucle:

$$G' := G$$
 5. Para cada par $\{p, q\}, p \neq q$ en G' se debe calcular
 6. r (resto de la división del S-polinomio $S(p, q)$ entre G')
 7. Si $r \neq 0$ entonces $G := G \cup \{r\}$
 8. Se realiza lo anterior hasta $G = G'$
 9. Se devuelve el valor de G

Este bucle computa lo explicado en el ejemplo anterior. Durante todas las fases del proceso algorítmico se mantiene que $G \subseteq I$, terminando el algoritmo en el momento en el que $G = G'$, es decir, cuando el resto de la división del S-polinomio entre el G' es igual a 0. Sin embargo, computar las bases de Gröbner calculadas mediante este algoritmo pueden resultar innecesariamente grandes. Se pueden eliminar los generadores prescindibles siguiendo el siguiente lema.

Lema 3 [3] Sea G una base de Gröbner de $I \subseteq k[x_1, \dots, x_n]$ y un polinomio $p \in G : LT(p) \in \langle LT(G \setminus \{p\}) \rangle$ entonces $G \setminus \{p\}$ es también una base de Gröbner de I .

Es decir, si existe un polinomio p cuyo término líder pertenezca al ideal formado por los LT del polinomio resultante de la división de G entre p , entonces ese mismo polinomio también forma parte de la base de Gröbner de I . De esta manera, eliminando todos los p que cumplen esta condición, llegamos a lo que se conoce como una base de Gröbner minimal, a la cual se la han eliminado todos aquellos generadores redundantes, o de los que se puede prescindir. Para ejemplificar este lema, tomemos la siguiente base de Gröbner:

$$\begin{aligned} f_1 &= x^3 - 2xy \\ f_2 &= x^2y - 2y^2 + x \\ f_3 &= -x^2 \\ f_4 &= -2xy \\ f_5 &= -2y^2 + x \end{aligned}$$

El primer paso consiste eliminar los coeficientes distintos de 1 en los términos líder. A continuación, se trata de reducir el conjunto tratando de buscar relación entre generadores. Por ejemplo, el término líder de $f_1, LT(f_1) = x^3$ se puede escribir como $LT(f_3) \cdot (-x) = -x^2 \cdot (-x)$, por tanto, podemos prescindir de f_1 en la base de Gröbner minimal. De la misma manera, aunque no tan sencillo de ver, $LT(f_2) =$

$x^2y = -\left(\frac{1}{2}\right)x \cdot LT(f_4)$, pudiendo también eliminar f_2 . Al no haber más casos en los que el término líder de uno de los generadores divida el término líder del otro, tenemos:

$$\begin{aligned}f'_3 &= x^2 \\f'_4 &= xy \\f'_5 &= y^2 - \left(\frac{1}{2}\right)x\end{aligned}$$

Cuando G es una base de Gröbner minimal, los LT de los polinomios pertenecientes a la base forman la una base minimal de términos líder del ideal I . No obstante, el ideal I original puede contener varias bases de Gröbner minimales. Veámoslo teniendo en cuenta el resultado anterior:

$$\begin{aligned}f'_3 &= x^2 + axy \\f'_4 &= xy \\f'_5 &= y^2 - \left(\frac{1}{2}\right)x\end{aligned}$$

Mediante esta base, siendo a una constante, se pueden producir infinitas bases minimales dando diferentes valores a a . Para resolver esta cuestión se introduce el concepto de base de Gröbner reducida.

Definición 3.2.7 [2] Una base de Gröbner reducida de un ideal polinomial I es una base G tal que todos sus coeficientes principales tienen valor 1, tal que $\forall p \in G$ ningún monomio de p reside en $\langle LT(G \setminus \{p\}) \rangle$. Para la base expuesta anteriormente, solo el caso en el que $a = 0$ conformaría la base de Gröbner reducida, que es siempre existente y única para todo ideal I .

Una consecuencia importante de lo que se esclarece en la definición 3.2.8 relativa a la resolución de sistemas de ecuaciones polinomiales, es que el algoritmo de Buchberger resulta ser un algoritmo de igualdad ideal, debido a que dos conjuntos de polinomios $\{f_1, \dots, f_s\}$ y $\{g_1, \dots, g_t\}$ generan el mismo ideal simplemente fijando un orden monomial y computando la base de Gröbner reducida. El algoritmo, a su vez, guarda relación con el algoritmo de eliminación Gaussiana para sistemas de ecuaciones lineales (apartado 2.3.2). Lo importante de esta relación, es que la eliminación Gaussiana se puede considerar un caso especial o específico del algoritmo que acabamos de describir. Veámoslo con el siguiente sistema:

$$\begin{aligned}3x - 6y - 2z &= 0 \\2x - 4y + 4w &= 0 \\x - 2y - z - w &= 0\end{aligned}$$

El caso es que si operamos con las filas de la matriz de coeficientes del sistema para construir una matriz escalonada por filas de este (lo que implica dejar el primer coeficiente como 1) obtenemos:

$$(3.2.7) \quad \begin{pmatrix} 1 & -2 & -1 & -1 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Para conseguir, además, la forma reducida de la matriz escalonada anterior, necesitamos asegurarnos de que cada 1 líder es el único elemento distinto de 0 de su columna, lo que nos lleva a la matriz

$$(3.2.8) \quad \begin{pmatrix} 1 & -2 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Obtenida la forma reducida de la matriz, reescribiremos el proceso anterior de forma algebraica, de modo que el sistema original tendría la forma I tal que

$$I = \langle 3x - 6y - 2z, 2x - 4y + 4w, x - 2y - z - w \rangle \subseteq k[x, y, z, w]$$

Convirtiendo la forma escalonada por filas (3.2.7) obtendríamos la base de Gröbner minimal:

$$I = \langle x - 2y - z - w, z + 3w \rangle$$

Y finalmente, con la forma reducida de dicha matriz (3.2.8), obtendríamos la base de Gröbner reducida:

$$I = \langle x - 2y + 2w, z + 3w \rangle$$

Por último, cabe destacar, que de toda matriz puede obtenerse su forma escalonada por filas reducida de una única forma, lo que reafirma la unicidad de las bases de Gröbner reducida respecto a un ideal. En el siguiente apartado se estudiarán las diversas aplicaciones de estas bases, profundizando en su eficacia a la hora de resolver sistemas de ecuaciones.

4 APLICACIONES Y EJEMPLOS

4.1 Bases de Gröbner como herramienta para simplificar la resolución de sistemas de ecuaciones polinomiales

4.1.1 Solución de ecuaciones polinomiales

Ejemplo 4.1 [3] Considerando el sistema de ecuaciones:

$$x^2 + y^2 + z^2 = 1$$

$$x^2 + y^2 + z^2 = 2x$$

$$2x - 3y = z$$

Estas ecuaciones, de manera similar al ejemplo que mostrado tras la definición 3.2.8, se pueden escribir en forma de ideal $I = \langle x^2 + y^2 + z^2 - 1, x^2 + y^2 + z^2 - 2x, 2x - 3y - z \rangle$ y trataremos de buscar todos los puntos en $V(I)$, es decir, la solución del sistema. En primer lugar, computamos la base de Gröbner reducida de I respecto al orden lexicográfico.

```
R.<x,y,z> = PolynomialRing(QQ,3, 'x, y, z', order='lex')
ec1 = x^2 + y^2 + z^2 - 1
ec2 = x^2 - 2*x + y^2 + z^2
ec3 = 2*x - 3*y - z
G = ideal(ec1, ec2, ec3).groebner_basis()
print("Base de Gröbner:", G)
```

Base de Gröbner: $[x - 1/2, y + 1/3z - 1/3, z^2 - 1/5z - 23/40]$

Figura 4-1 Código en SageMath para obtener la base de Gröbner del ideal correspondiente al sistema de ecuaciones del ejemplo 4.1

Modificamos la base anterior de manera que no haya cocientes en los coeficientes de manera que $G = \{g_1, g_2, g_3\} = \{2x - 1, 3y + z, 40z^2 - 8z - 23\}$. Estudiando la naturaleza de la base, observamos que g_3 depende únicamente de la variable z , g_2 de las variables z e y , y g_1 únicamente de x . De esta manera obtenemos de manera directa los resultados de $g_1 = 0 \Rightarrow x = \frac{1}{2}$ y de $g_3 = 0$, del cual resolviendo mediante la fórmula cuadrática obtenemos que $z = \frac{8 \pm \sqrt{3744}}{80}$. Si se sustituyen ambos valores en g_2 se obtienen ambas soluciones del sistema:

$$x = \frac{1}{2}, y = \frac{\sqrt{26} - 6}{20}, z = \frac{8 - \sqrt{3744}}{80}$$

$$x = \frac{1}{2}, y = -\frac{\sqrt{26} - 6}{20}, z = \frac{8 + \sqrt{3744}}{80}.$$

En SageMath, se obtendría introduciendo los diferentes polinomios que conforman la base de Gröbner como elementos de un sistema de ecuaciones y ejecutando su solución.

```
x,y,z = var('x,y,z')
g1 = x - 1/2
g2 = y + 1/3*z - 1/3
g3 = z^2 - 1/5*z - 23/40
solve([g1,g2,g3],x,y,z)
```

```
[[x == (1/2), y == 1/20*sqrt(26) + 3/10, z == -3/20*sqrt(26) + 1/10], [x == (1/2), y == -1/20*sqrt(26) + 3/10, z == 3/20*sqrt(26) + 1/10]]
```

Figura 4-2 Código en SageMath para resolver el sistema de ecuaciones resultante de la base de Gröbner del ejemplo 4.1

Al computar bases de Gröbner, nos encontraremos con casos como el anterior, en el que la base reducida cuenta con variables eliminadas que simplifican la resolución del sistema. Estas variables se eliminarán en el mismo orden que establece el orden monomial. Por ejemplo, si utilizamos el orden lexicográfico $x > y > z$, veremos, como es el caso del ejemplo anterior, que la primera variable en ser eliminada es la x , dando lugar así a una sustitución hacia atrás similar a la utilizada en la eliminación Gaussiana, y dando con soluciones más sencillas que a las que obtendríamos si tratásemos directamente con el sistema original. No obstante, el orden en el que se eliminen estas variables es computacionalmente irrelevante y no afectan a las soluciones. La importancia de dicha eliminación reside en el hecho de conseguir al menos un polinomio en la base reducida que dependa de una única variable (como es el caso de g_1 y g_3) haciendo más sencillo el proceso de dar directamente con la solución. Veamos ahora un ejemplo en el que los resultados obtenidos al computar la base sean más complejos.

Ejemplo 4.2 [3] Dado el siguiente sistema:

$$3x^2 + 2xz - 2yz = 0$$

$$2xz - 2ty = 0$$

$$2xy - 2z - 2tz = 0$$

$$x^2 + y^2 + z^2 - 1 = 0$$

Con las ecuaciones anteriores generamos el ideal $I = \langle 3x^2 + 2xz - 2yz, 2xz - 2ty, 2xy - 2z - 2tz, x^2 + y^2 + z^2 - 1 \rangle$. Se trabajará de nuevo respecto al orden lexicográfico $\lambda > x > y > z$.

```
R.<t,x, y, z> = PolynomialRing(QQ, order='lex')
I = ideal(3*x^2 + 2*y*z - 2*t*x, 2*x*z - 2*t*y, 2*x*y - 2*z - 2*t*z, x^2 + y^2 + z^2 - 1)
GB = I.groebner_basis()
print("G =", GB)
```

G = [t - 3/2*x - 3/2*y*z - 167616/3835*z^6 + 36717/590*z^4 - 134419/7670*z^2, x^2 + y^2 + z^2 - 1, x*y - 19584/3835*z^5 + 1999/295*z^3 - 6403/3835*z, x*z + y*z^2 - 1152/3835*z^5 - 108/295*z^3 + 2556/3835*z, y^3 + y*z^2 - y - 9216/3835*z^5 + 906/295*z^3 - 2562/3835*z, y^2*z - 6912/3835*z^5 + 827/295*z^3 - 3839/3835*z, y*z^3 - y*z - 576/59*z^6 + 1605/118*z^4 - 453/118*z^2, z^7 - 1763/1152*z^5 + 655/1152*z^3 - 11/288*z]

Figura 4-3 Código en SageMath para computar la base de Gröbner del ideal correspondiente al sistema de ecuaciones del ejemplo 4.2

Intuitivamente, resulta más sencillo resolver un sistema de 4 ecuaciones que de 8, como es el caso del presente ejemplo. No obstante, examinando en detalle los polinomios resultantes, nos damos cuenta de que únicamente el primero contiene las 4 variables a la vez, y como establecimos anteriormente, la variable t , la primera del orden lexicográfico, es la primera en ser eliminada, así como la x y la y , quedando al final una única expresión en función de z que modificada para que carezca de cocientes en sus coeficientes quedaría

$$1152z^7 - 1763z^5 + 665z^3 - 44z = 0$$

obteniendo sus respectivas soluciones

$$z = 0, \pm 1, \pm \frac{2}{3} \pm \frac{\sqrt{11}}{8\sqrt{2}}$$

Una vez obtenidas las soluciones de z , podemos continuar sustituyendo los valores de y , posteriormente los de x , hasta finalmente obtener los valores de la solución para cada una de las cuatro variables (sustitución hacia atrás). Los sistemas de ecuaciones sobre los que es posible aplicar este método son más sencillos de resolver, especialmente, cuando conseguimos obtener una ecuación que únicamente contiene una variable, sobre la que podamos aplicar técnicas de resolución de ecuaciones polinomiales para encontrar sus raíces, para entonces sustituir hacia atrás en las otras ecuaciones del sistema y resolver el resto de las variables aplicando las mismas técnicas de resolución de ecuaciones con una variable.

Ejemplo 4.3 [9] Se pretende en el presente ejemplo mostrar el nivel de complejidad que las bases son capaces de manejar. Para ello, en la Figura 4-4 se mostrarán una serie de ecuaciones que corresponden a un problema relativo al tensor de Bach, un objeto matemático utilizado en la teoría de la relatividad general de Einstein que describe como la curvatura del espacio-tiempo cambia según la presencia de materia y energía. En este problema, se pretende anular el tensor de Bach. Esto sucede cuando un sistema no lineal como el que se muestra a continuación se anula.

$$\begin{aligned}\mathfrak{B}_{11} &= \frac{1}{24R^4}\mathfrak{P}_{11}, \quad \mathfrak{B}_{12} = \frac{1}{12R^4}\mathfrak{P}_{12}, \quad \mathfrak{B}_{13} = \frac{1}{12R^4}\mathfrak{P}_{13}, \quad \mathfrak{B}_{14} = \frac{\lambda_2}{12R^3}\mathfrak{P}_{14}, \\ \mathfrak{B}_{22} &= \frac{1}{24R^4}\mathfrak{P}_{22}, \quad \mathfrak{B}_{23} = \frac{1}{12R^4}\mathfrak{P}_{23}, \quad \mathfrak{B}_{24} = \frac{1}{12R^3}\mathfrak{P}_{24}, \quad \mathfrak{B}_{33} = \frac{1}{24R^4}\mathfrak{P}_{33}, \\ \mathfrak{B}_{34} &= \frac{1}{12R^3}\mathfrak{P}_{34}, \quad \mathfrak{B}_{44} = \frac{1}{24R^4}\mathfrak{P}_{44},\end{aligned}$$

$$\begin{aligned}\mathfrak{P}_{11} &= 12(A^2 + R^2)^2\lambda_2^4 - 4(A^2 + R^2)^2\lambda_2^3 - (20b^2 - C^2 - 8D^2)(A^2 + R^2)\lambda_2^2 \\ &\quad + (12A^4 - 4(2b^2 - 3C^2 - D^2 - 6R^2)A^2 - 42bCDA - 4R^2(2b^2 - 3C^2 - D^2 - 3R^2))\lambda_2 \\ &\quad - 20A^4 + (28b^2 - 40C^2 + 3D^2 - 40R^2)A^2 - 42bCDA - 20R^4 + (3D^2 - 40C^2)R^2 \\ &\quad - 4(C^2 + D^2)(5C^2 + D^2) + b^2(43C^2 + D^2 + 28R^2), \\ \mathfrak{P}_{12} &= -16b(A^2 + R^2)A\lambda_2^3 - 8CD(A^2 + R^2)\lambda_2^2 - (5CDA^2 - b(5C^2 - 16D^2)A + 5CDR^2)\lambda_2 \\ &\quad + 16bA^3 - 8CDA^2 + b(16C^2 - 5D^2 + 16R^2)A + CD(21b^2 - 8(C^2 + D^2 + R^2)), \\ \mathfrak{P}_{13} &= -8AD(A^2 + R^2)\lambda_2^3 + (4AD(A^2 + R^2) - 3bCR^2)\lambda_2^2 \\ &\quad + (DA^3 - 9bCA^2 + D(12b^2 + R^2 - 8(C^2 + D^2))A - 12bCR^2)\lambda_2 \\ &\quad + 3b(8CA^2 + 3bDA - 3b^2C + 8C(C^2 + D^2 + R^2)), \\ \mathfrak{P}_{14} &= -8D(A^2 + R^2)\lambda_2^2 + (4DA^2 + 3bCA + 4DR^2)\lambda_2 \\ &\quad + DA^2 + 3bCA + D(3b^2 + R^2 - 8(C^2 + D^2)), \\ \mathfrak{P}_{22} &= -20(A^2 + R^2)^2\lambda_2^4 + 12(A^2 + R^2)^2\lambda_2^3 + (28b^2 + 3C^2 - 40D^2)(A^2 + R^2)\lambda_2^2 \\ &\quad - (4A^4 + 4(2b^2 - C^2 - 3D^2 + 2R^2)A^2 - 42bCDA + 4R^2(2b^2 - C^2 - 3D^2 + R^2))\lambda_2 \\ &\quad + 12A^4 - (20b^2 - 8C^2 - D^2 - 24R^2)A^2 + 42bCDA \\ &\quad + 12R^4 + (8C^2 + D^2)R^2 - 4(C^2 + D^2)(C^2 + 5D^2) + b^2(C^2 + 43D^2 - 20R^2), \\ \mathfrak{P}_{23} &= -(AC - 24bD)(A^2 + R^2)\lambda_2^2 - (4(AC + 3bD)R^2 + A(4CA^2 + 9bDA + 9b^2C))\lambda_2 \\ &\quad + 8CA^3 + 4C(2(C^2 + D^2 + R^2) - 3b^2)A - 3bD(3b^2 + R^2 - 8(C^2 + D^2)), \\ \mathfrak{P}_{24} &= -C(A^2 + R^2)\lambda_2^2 - (4CA^2 - 3bDA + 4CR^2)\lambda_2 \\ &\quad + 8CA^2 + 3bDA - 3b^2C + 8C(C^2 + D^2 + R^2), \\ \mathfrak{P}_{33} &= -4(A^2 - 3R^2)(A^2 + R^2)\lambda_2^4 + 4(A^2 - 3R^2)(A^2 + R^2)\lambda_2^3 \\ &\quad - ((12b^2 + C^2 - 8D^2)A^2 + 3(4b^2 + C^2 - 8D^2)R^2)\lambda_2^2 \\ &\quad + 2(2A^4 + 2(6b^2 - C^2 - D^2 - 2R^2)A^2 + 9bCDA + 6R^2(2b^2 - C^2 - D^2 - R^2))\lambda_2 \\ &\quad - 4A^4 - (12b^2 - 8C^2 + D^2 - 8R^2)A^2 - 18bCDA \\ &\quad + 12R^4 - 3(4b^2 - 8C^2 + D^2)R^2 + 3(C^2 + D^2)(4(C^2 + D^2) - 19b^2), \\ \mathfrak{P}_{34} &= -8A(A^2 + R^2)\lambda_2^4 + 8A(A^2 + R^2)\lambda_2^3 + A(C^2 - 8D^2)\lambda_2^2 \\ &\quad + (8A^3 + 4(C^2 + D^2 + 2R^2)A + 9bCD)\lambda_2 \\ &\quad - 8A^3 - 9bCD - A(8C^2 - D^2 + 8R^2), \\ \mathfrak{P}_{44} &= 4(3A^2 - R^2)(A^2 + R^2)\lambda_2^4 - 4(3A^4 + 2R^2A^2 - R^4)\lambda_2^3 \\ &\quad + ((4b^2 - 3C^2 + 24D^2)A^2 + (4b^2 - C^2 + 8D^2)R^2)\lambda_2^2 \\ &\quad + (4R^4 - 4(2A^2 + 2b^2 + C^2 + D^2)R^2 - 2A(6A^3 + 4b^2A + 6(C^2 + D^2)A + 9bCD))\lambda_2 \\ &\quad + 12A^4 + (4b^2 + 24C^2 - 3D^2 + 8R^2)A^2 + 18bCDA \\ &\quad - 4R^4 + (4b^2 + 8C^2 - D^2)R^2 + (C^2 + D^2)(13b^2 + 12(C^2 + D^2)).\end{aligned}$$

Figura 4-4 Sistema de ecuaciones correspondiente al tensor de Bach [9]

Dadas las ecuaciones que conforman el sistema, se computa la base de Gröbner del ideal $I \subset \mathbb{R}[A, b, \lambda_2, C, D, R]$ formado por los polinomios $\mathfrak{P}_{ij}$ respecto al orden lexicográfico. Analizando la base de Gröbner G , obtenemos un polinomio $g_0 \in G$ que tiene esta forma:

$$g_0 = D^6(C^2 + D^2)(2D^2 + R^2)(25D^2 + 4R^2)(16D^2 + 5R^2)(9D^2 + 16R^2)(25D^2 + 24R^2)(80D^4 + R^4 - 16D^2R^2).$$

Como los conjuntos $\{\mathfrak{P}_{ij} = 0\}$ y $I = \langle \mathfrak{P}_{ij} \rangle = \langle G \rangle$ coinciden, entonces necesariamente $D = 0$. A continuación, computamos la base de Gröbner G_1 del ideal generado por la base de Gröbner anterior sustituyendo con $D = 0$, de nuevo respecto a un orden lexicográfico, de entre los polinomios que conforman la base G_1 obtenemos:

$$g_1 = C^4(9C^2 + 4R^2)(25C^2 + 16R^2)(49C^2 + 24R^2)\lambda_2^3.$$

Tal como ocurre con la base anterior, $g_1 \in G_1$. Por el contexto del problema λ_2 ha de ser necesariamente distinto de 0, por lo que $C = 0$. Llegados a este punto, con $C = D = 0$, tomamos la ecuación $\mathfrak{P}_{34}$ de la Figura 4-4 substituyendo ambos valores quedando de la siguiente forma:

$$\mathfrak{P}_{34} = -8(\lambda_2 - 1)^2 A(A^2 + R^2)(\lambda_2^2 + \lambda_2 + 1)$$

llevándonos a dos posibilidades sobre las que proceder aplicando casuística:

$$\lambda_2 = 1 \text{ o } A = 0.$$

Ambos casos obtienen dos soluciones distintas del sistema planteado, consiguiendo anular el tensor de Bach. Este problema muestra como ante sistemas más complejos, es necesario el análisis de cada uno de los polinomios obtenidos al computar la base de Gröbner, así como trabajar con casuística con cada una de estas expresiones hasta lograr obtener la solución que mejor se adecue al sistema planteado. Este procedimiento puede volverse significativamente complejo y extenso si las bases computadas contienen un gran número de polinomios debido a la necesidad de analizar, tener en cuenta los detalles y aplicar casuística sobre cada uno de ellos.

4.1.2 Resolución de casos de programación entera

El problema de programación entera (PPLE) son problemas de programación lineal en los que se exige que alguna o todas las variables sean enteras. A simple vista, estos problemas pueden parecer iguales a la resolución de un sistema de ecuaciones polinomiales, no obstante, en ellos no se trata únicamente de encontrar un valor que solucione el sistema, sino que se debe encontrar un valor entero y que, además, proporcione un valor óptimo (máximo o mínimo). La teoría dicta lo siguiente [10]: Sea $a_{ij} \in \mathbb{Z}$, $b_i \in \mathbb{Z}$ y $c_j \in \mathbb{R}$, $i = 1, \dots, n$ y $j = 1, \dots, m$ (m y n representan el número de incógnitas y el número de ecuaciones respectivamente), se pretende buscar una solución $(l_1, l_2, \dots, l_m)$ en $\mathbb{N}^m$ del sistema

$$(4.1.1) \quad \begin{aligned} a_{11}l_1 + a_{12}l_2 + \dots + a_{1m}l_m &= b_1 \\ a_{21}l_1 + a_{22}l_2 + \dots + a_{2m}l_m &= b_2 \\ &\vdots \\ a_{n1}l_1 + a_{n2}l_2 + \dots + a_{nm}l_m &= b_n \end{aligned}$$

que minimice el coste de la función $c(l_1, l_2, \dots, l_m) = \sum_{j=1}^m c_j l_j$. Como hemos visto a lo largo del trabajo, las bases de Gröbner trabajan con polinomios, por lo que para resolver el sistema anterior debemos transformarlo en un problema de polinomios. Para ello, asignamos una variable a cada ecuación lineal del sistema. Estas variables tendrán la forma $x_1, x_2, \dots, x_n$ para n ecuaciones, de manera que la primera ecuación del sistema se representaría:

$$(4.1.2) \quad x_i^{a_{i1}l_1 + \dots + a_{im}l_m} = x_i^{b_i}.$$

Como queremos encontrar una solución del sistema entero, debemos combinar cada una de las ecuaciones del sistema, expresadas del mismo modo que (4.1.1), para construir una única ecuación que represente el sistema. Por tanto, se multiplica las n ecuaciones del sistema formando la ecuación

$$(4.1.3) \quad x_i^{a_{i1}l_1 + a_{i2}l_2 + \dots + a_{im}l_m} \cdot \dots \cdot x_n^{a_{n1}l_1 + a_{n2}l_2 + \dots + a_{nm}l_m} = x_1^{b_1} \cdot \dots \cdot x_n^{b_n}$$

Agrupando incógnitas, se escribiría de la forma

$$(x_1^{a_{11}} x_2^{a_{21}} \cdot \dots \cdot x_n^{a_{n1}})^{l_1} \cdot \dots \cdot (x_1^{a_{1m}} x_2^{a_{2m}} \cdot \dots \cdot x_n^{a_{nm}})^{l_m} = x_1^{b_1} \cdot \dots \cdot x_n^{b_n}, \text{ donde:}$$

1. $x_1, \dots, x_n$ representan a las diferentes ecuaciones.

2. $l_1, \dots, l_n$ hace referencia a las variables que aparecen en el sistema, por ejemplo ($l_1 = x, l_2 = y, l_3 = z$).
3. $a_{n1}, \dots, a_{nm}$ representan los coeficientes, de manera que el primer subíndice indica la ecuación en el que se encuentra, y el segundo, la variable a la que acompaña.
4. $b_1, \dots, b_n$ representan los números del otro lado de la igualdad de cada una de las ecuaciones.

Una vez transformado el sistema, debemos definir la aplicación polinomial ϕ . Una aplicación polinomial es a grandes rasgos una función matemática que toma un conjunto de valores y los transforma en otros utilizando ecuaciones polinomiales. Por ejemplo, dado un espacio vectorial V sobre un cuerpo K , una aplicación polinomial de V se puede representar como $f: V \rightarrow V$, donde para cada vector x en V existe un vector y en V tal que $y = f(x)$. Volviendo al caso, ϕ es una aplicación polinomial $k[y_1, \dots, y_n] \rightarrow k[x_1, \dots, x_n]$ tal que $\phi(y_j) = f_j$ siendo $f_j = x_1^{a_{1j}} x_2^{a_{2j}} \cdot \dots \cdot x_n^{a_{nj}}, j = 1, 2, \dots, m$. Utilizando esta aplicación, la ecuación anterior se transforma de manera que

$$\phi(y_1^{l_1} \dots y_m^{l_m}) = x_1^{b_1} \cdot \dots \cdot x_n^{b_n}.$$

Como el monomio formado por $x_1^{b_1} \cdot \dots \cdot x_n^{b_n}$ está bajo la imagen de ϕ de un monomio en $k[y_1, \dots, y_n]$, $(l_1, l_2, \dots, l_m)$ es la solución del problema de programación entera dado.

Aplicando lo mencionado en el párrafo anterior, obtenemos un procedimiento que nos proporciona una solución para un problema de programación entera como el mostrado en la expresión (4.1.1) Este procedimiento consta de los siguientes pasos:

1. Se computa la base de Gröbner G para el ideal I respecto al orden lexicográfico.
2. Obtenemos el resto h resultado de dividir f entre G .
3. Si $h \notin k[y_1, \dots, y_m]$ el problema de programación entera no tiene soluciones enteras positivas. En cambio, si $r = y_1^{l_1} \dots y_m^{l_m} \Rightarrow (l_1, l_2, \dots, l_m)$ es una solución al sistema.

Ejemplo 4.4 [10] Tratemos de obtener una solución para el siguiente sistema de ecuaciones que maximice $3x + 4y + 2z$. No es necesario escribirlo en el enunciado del problema, pero por definición del concepto de programación entera $x, y, z \geq 0$ y $x, y, z \in \mathbb{N}$

$$3x + 2y + z \leq 45$$

$$x + 2y + 3z \leq 21$$

$$2x + y + z \leq 18$$

La cuestión planteada se puede plantear del modo opuesto, es decir, minimizando la ecuación cambiada de signo. Además, se añaden las variables de exceso (w_1, w_2, w_3), lo que nos permite transformar las restricciones de desigualdad en ecuaciones de igualdad, para así poder utilizar métodos de optimización diseñados para resolver sistemas de ecuaciones lineales como lo son las bases de Gröbner.

$$\min -3x - 4y - 2z + 0 \cdot w_1 + 0 \cdot w_2 + 0 \cdot w_3$$

$$3x + 2y + z + w_1 = 45$$

$$x + 2y + 3z + w_2 = 21$$

$$2x + y + z + w_3 = 18$$

Como se explica anteriormente, el sistema se debe escribir en forma de ecuación polinomial, por tanto:

$$(x_1^3 x_2 x_3^2)^x (x_1^2 x_2^2 x_3)^y (x_1 x_2^3 x_3)^z x_1^{w_1} x_2^{w_2} x_3^{w_3} = x_1^{45} x_2^{21} x_3^{18}$$

Llegados a este punto, sabemos que requerimos el ideal en $k[x_1, x_2, x_3, y_1, y_2, \dots, y_6] \Rightarrow I = \langle y_1 - x_1^3 x_2 x_3^2, y_2 - x_1^2 x_2^2 x_3, y_3 - x_1, x_2^3 x_3, y_4 - x_1, y_5 - x_2, y_6 - x_3 \rangle$. De esta manera, usando SageMath podemos computar la base de Gröbner G de I , siguiendo un orden lexicográfico graduado.

```
R.<x1,x2,x3,y1,y2,y3,y4,y5,y6> = PolynomialRing(QQ, order='deglex')
I = ideal(y1-x1^3*x2*x3^2,y2
-x1^2*x2^2*x3,y3-x1*x2^3*x3,
y4-x1,y5-x2,y6-x3).groebner_basis()

I

[y1*y3^3*y4^4 - y2^5*y6, y1*y3^2*y4^3*y5 - y2^4*y6, y3^2*y4^4*y6 - y2^3, y4^3*y
5*y6^3 - y1, y1*y3*y4^2*y5^2 - y2^3*y6, y3*y4^3*y5*y6 - y2^2, y1*y4*y5^3 - y2^2
*y6, y1*y5^4 - y2*y3*y6, y3*y4^2*y6^2 - y1*y5^2, y4^2*y5^2*y6 - y2, y4*y5^3*y6
- y3, y2^2*y6^2 - y1*y3, y2*y4*y6^2 - y1*y5, y2*y5 - y3*y4, x1 - y4, x2 - y5, x
3 - y6]

(x1^45*x2^21*x3^18).reduce(I)

y1^3*y2^9*y4^18
```

Figura 4-5 Código en la plataforma SageMath para computar la base de Gröbner y realizar la división correspondiente al problema planteado en el ejemplo 4.4

Una vez obtenida la base de Gröbner del ideal, dividimos el monomio $f = x_1^{45} x_2^{21} x_3^{18}$ entre G para hallar las soluciones del sistema. Utilizando el algoritmo de la división, obtenemos como resto y_1^3, y_2^9, y_3^{18} que teniendo en cuenta que si $r = y_1^{l_1} \dots y_m^{l_m} \Rightarrow (l_1, l_2, \dots, l_m)$ son las soluciones del sistema, estas serían $(3, 9, 0)$.

Ejemplo 4.5 (Ejemplo de aplicación de las bases de Gröbner en un problema de optimización de espacios en un buque de la Armada). Se pretende embarcar en el buque anfibio Galicia (L-51) al Batallón de Desembarco I (BD-I) como Fuerza de Desembarco (FD) compuesto por tres compañías de Fusiles y una Compañía de Armas como unidades de combate. Para abastecer a estas compañías durante la operación, se pretende embarcar tanta munición como sea posible en los pañoles del buque destinados para este fin (Pañol Munición FD-1, Pañol de Munición FD-2 y Pañol de Munición FD-3). Las tres Compañías de Fusiles, disponen de la totalidad del Pañol de Munición FD-1 para almacenar toda la munición que llevan consigo (cartuchos del 5.56, cartuchos del 7.62 y granadas Alhambra). Esta munición viene almacenada en pallets que tienen las siguientes características:

- Pallet de cartuchos del 5.56 :
 - Superficie: 1 m^2 .
 - Volumen: 3 m^3 .
 - Peso: 2 tn .
 - Cartuchos por pallet: 72000.
- Pallet de cartuchos del 7.62 :
 - Superficie: 2 m^2 .
 - Volumen: 2 m^3 .
 - Peso: 3 tn .
 - Cartuchos por pallet: 36000.
- Pallet de Granadas Alhambra:
 - Superficie: 3 m^2 .
 - Volumen: 1 m^3 .
 - Peso: 1 tn .
 - Granadas por pallet: 864.

El Pañol de Munición FD-1 cuenta con una superficie útil de 76 m^2 , un volumen útil de 100 m^3 y una resistencia de superficie de $2,63 \frac{\text{tn}}{\text{m}^2}$, permitiendo una carga total de 200 tn . Partiendo de los datos aportados, obtenga el máximo número de palets de cada tipo a cargar por parte del BD-I de manera que se optimice la distribución de estos en el Pañol de Munición FD-1 cargándose la máxima munición posible.

En primer lugar, otorgamos a cada tipo de pallet una variable x, y, z , de manera que x corresponde al número de pallets de cartuchos del 5.56, y al número de pallets de cartuchos del 7.62 y z al número de pallets de granadas Alhambra. Una vez asignadas las variables podemos definir la función, en este caso, a maximizar: $\max x + y + z$. O lo que sería lo mismo: $\min -x - y - z$. A continuación, a partir de los datos de los pallets y las limitaciones que ofrece el pañol, construimos las ecuaciones que conforman las restricciones del problema:

$$\begin{array}{ll} \text{Superficie:} & x + 2y + 3z \leq 76 \\ \text{Volumen:} & 3x + 2y + z \leq 100 \\ \text{Peso total:} & 2x + 3y + z \leq 200 \end{array}$$

De la misma forma que se realiza en el Ejemplo 3.14, se reescribe el sistema igualando las restricciones a sus valores correspondientes y añadiendo las variables de exceso (w_1, w_2, w_3) de manera que el problema quedaría:

$$\begin{array}{l} \min -x - y - z + 0 \cdot w_1 + 0 \cdot w_2 + 0 \cdot w_3 \\ x + 2y + 3z + w_1 = 76 \\ 3x + 2y + z + w_2 = 100 \\ 2x + 3y + z + w_3 = 200 \end{array}$$

Ya una vez encaminados los datos hacia un problema de programación lineal, transformamos las expresiones anteriores en una ecuación polinomial tal y como se desarrolla en el apartado 4.1.2:

$$(x_1 x_2^3 x_3^2)^x (x_1^2 x_2^2 x_3^3)^y (x_1^3 x_2 x_3)^z x_1^{w_1} x_2^{w_2} x_3^{w_3} = x_1^{76} x_2^{100} x_3^{200}.$$

Una vez constituida la ecuación polinómica, conformamos el ideal correspondiente a la parte izquierda de la igualdad $I = \langle y_1 - x_1 x_2^3 x_3^2, y_2 - x_1^2 x_2^2 x_3^3, y_3 - x_1^3 x_2 x_3, y_4 - x_1, y_5 - x_2, y_6 - x_3 \rangle$ y obtenemos su base de Gröbner para, a continuación, dividir la expresión $x_1^{76} x_2^{100} x_3^{200}$ entre la misma.

```
R.<x1,x2,x3,y1,y2,y3,y4,y5,y6> = PolynomialRing(QQ,9,order='deglex')
G = ideal(y1 - x1*x2^3*x3^2, y2-x1^2*x2^2*x3^3, y3-x1^3*x2*x3^1, y4-x1, y5-x2, y6-x3).groebner_basis()
print("Base de Gröbner:", G)
```

```
Base de Gröbner: [y3^2*y6^7 - y1^3*y4, y3^2*y4^5*y6 - y2^3, y3^2*y4^3*y6^3 - y1*y2^2, y3^2*y4*y6^5 - y1^2*y2, y3*y4^4*y5*y6 - y2^2, y3*y4^2*y5*y6^3 - y1*y2, y3*y5*y6^5 - y1^2, y4^3*y5^2*y6 - y2, y4^2*y5^3*y6 - y3, y4*y5^2*y6^3 - y1, y1*y4^2 - y2*y6^2, y1*y4*y5 - y3*y6^2, y2*y5 - y3*y4, x1 - y4, x2 - y5, x3 - y6]
```

```
(x1^76*x2^100*x3^200).reduce(G)
```

```
y1^37*y2^13*y6^76
```

Figura 4-6 Código en SageMath para computar la base de Gröbner del ideal correspondiente al problema-ejemplo de carga de munición planteado

La solución obtenida (37,13,0) indica que para optimizar la distribución se deben cargar 37 pallets de cartuchos del 5.56 y 13 pallets de cartuchos del 7.62. No obstante, en esta solución óptima no tiene lugar

la variable y_3 que hace referencia a los pallets de granadas Alhambra, por lo que se deberían almacenar en otro de los dos pañoles destinados al almacenaje de munición de la fuerza de desembarco.

Mediante los ejemplos tratados, observamos como las bases de Gröbner ofrecen una herramienta potente a la hora de resolver tanto sistemas de ecuaciones lineales como problemas de programación entera, obteniendo de manera rápida resultados óptimos o mayormente óptimos, facilitando así avances en áreas como la planificación, la producción, la logística y muchos otros campos donde se requiere la asignación eficiente de recursos sujetos a abundantes restricciones.

4.2 Otras aplicaciones de las bases de Gröbner

Muchos sistemas algebraicos computacionales implementan diferentes versiones del algoritmo de Buchberger para computar las bases de Gröbner. Estos sistemas tratan, como se ve al final del apartado 3.2, de simplificar aquellos elementos que resultan ser múltiplos de los elementos que conforman una base reducida.

Existen muchas aplicaciones de las bases de Gröbner en una gran variedad de áreas y ámbitos como la geometría algebraica, la teoría de grafos, criptografía o programación entera. En cuanto a los sistemas de ecuaciones polinomiales, se desarrollará un método para eliminar variables de un sistema y de esta manera poder computar generadores de la intersección de dos o más ideales.

4.2.1 Problema de pertenencia a un ideal

A lo largo del presente trabajo, se ha tratado de diversas formas el problema de pertenencia a un ideal: el algoritmo de la división (apartado 3.2.2), el lema de Dickson (apartado 3.2.3) y las bases de Hilbert (apartado 3.2.4). No obstante, la combinación de bases de Gröbner con el algoritmo de la división nos ofrecen el método más efectivo a la hora de obtener el algoritmo de pertenencia a un ideal: dado un ideal $I = \langle f_1, \dots, f_s \rangle$, podemos saber si un polinomio dado f reside en I , utilizando la base de Gröbner $G = \{g_1, \dots, g_t\}$ del ideal, de manera que si el resto obtenido de dividir f entre G resulta 0, dicho polinomio pertenece al ideal, es decir:

$$f \in I \Leftrightarrow \overline{f^G} = 0$$

siendo $\overline{f^G} = 0$ la expresión utilizada para hacer referencia al resto de la división mencionada en el apartado anterior.

Ejemplo 4.6 [2] Dado un ideal $I = \langle f_1, f_2 \rangle = \langle xz - y^2, x^3 - z^2 \rangle \in \mathbb{C}[x, y, z]$ y haciendo uso del orden lexicográfico graduado, se pretende averiguar si $f = -4x^2y^2z^2 + y^6 + 3z^5 \in I$. Nótese que f_1 y f_2 no conforman una base de Gröbner de I ya que el S-polinomio de ambos no pertenece al ideal $\langle LT(f_1), LT(f_2) \rangle = \langle xz, x^3 \rangle$. Por tanto, se tendrá que proceder según lo desarrollado en el ejemplo 3.10. Utilizando el código en SageMath de la Figura 4-3, obtenemos una base de Gröbner:

$$G = \{f_1, f_2, f_3, f_4, f_5\} = \langle xz - y^2, x^3 - z^2, x^2y^2 - z^3, xy^4 - z^4, y^6 - z^5 \rangle.$$

Una vez obtenida la base de Gröbner, ya se puede realizar la división de f entre G obteniendo:

$$f = (-4xy^2z - 4y^4) \cdot f_1 + (-3) \cdot f_5 + 0$$

Como el resto es $0 \Rightarrow f \in I$.

4.2.2 Teoría de Grafos (coloración)

La teoría de grafos fue introducida en el año 1736 por el matemático Leonhard Euler cuando se planteó el problema de los puentes de Königsberg, que consistía en encontrar un camino que recorriera los siete puentes del río Pregel de manera que se recorrieran todos los puentes pasando una sola vez por cada uno de ellos. A través del problema introducido por el matemático de origen suizo, se desarrolló el término grafo, hasta que en el año 1878 se presentó formalmente a la London Mathematical Society, fundamentando la teoría de la siguiente manera: [11] “Dada cualquier división de un mapa plano en regiones, éste puede ser coloreado únicamente con cuatro colores, de manera que regiones con frontera común tengan colores distintos.” A partir de este enunciado, se han encontrado diversas aplicaciones a la teoría de grafos relacionadas con la resolución de problemas de distribución y almacenamiento, así como se ha remarcado la importancia de saber el número mínimo de colores necesarios para colorear un grafo, lo que proporciona información clave a la hora de resolver estos problemas. El tipo de coloración de grafos a tratar en el presente trabajo para observar la utilidad de las bases de Gröbner en esta teoría, es el coloreado de aristas, que consiste en colorear todas las aristas de un grafo de manera que dos aristas incidentes en el mismo vértice no puedan ser coloreadas del mismo color.

Ejemplo 4.7 [12] Consideremos el siguiente mapa de américa del sur:



Figura 4-7 Grafo a colorear correspondiente al Ejemplo 4.7 [12]

Se tomarán los colores azul, verde y rojo y $x_1, \dots, x_8$ como variables para cada uno de los países a colorear. El término i hace referencia a cada uno de los países de manera independiente, mientras que i, j hace referencia a cada pareja de países que comparten frontera. Tanto i como j son números naturales. La ecuación $x_i^3 - 1 = 0$ para todo i especifica, por tanto, que todo país debe obtener un color, y además $x_i \neq x_j$, indicando que los países con frontera, es decir, adyacentes, no pueden tener el mismo color. El ideal con el que se abordará el problema de la 3-coloración contendrá generadores de dos tipos:

1. $x_i^3 - 1 \forall i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$.
2. $x_i^2 + x_i x_j + x_j^2 \forall i, j \in \{1, 2, 3, 4, 5, 6, 7, 8\} \Rightarrow x_i \neq x_j$, es decir, para cada uno de los países que comparten frontera. Esta expresión se obtiene a partir de que, al tratarse (i, j) dos vértices adyacentes, se cumple que $x_i^3 - x_j^3 = 0$. Al desarrollar esta expresión nos queda $(x_i - x_j)(x_i^2 + x_i x_j + x_j^2) = 0$. Si el primer paréntesis de la multiplicación fuese igual a 0, se cumpliría que $x_i = x_j$. Como deseamos completamente lo contrario, necesariamente $x_i^2 + x_i x_j + x_j^2 = 0$. En el caso del ejemplo los vértices adyacentes son:
 $\{(5, 4), (5, 6), (5, 3), (5, 7), (7, 3), (7, 1), (3, 6), (3, 1), (1, 2), (2, 8), (1, 8)\}$.

El ideal del grafo del ejemplo será entonces:

$$I = \langle x_1^3 - 1, x_2^3 - 1, x_3^3 - 1, x_4^3 - 1, x_5^3 - 1, x_6^3 - 1, x_7^3 - 1, x_8^3 - 1, x_1^2 + x_1x_2 + x_2^2, x_1^2 + x_1x_7 + x_7^2, x_1^2 + x_1x_8 + x_8^2, x_2^2 + x_2x_8 + x_8^2, x_1^2 + x_1x_3 + x_3^2, x_5^2 + x_5x_3 + x_3^2, x_6^2 + x_6x_3 + x_3^2, x_7^2 + x_7x_3 + x_3^2, x_5^2 + x_5x_4 + x_4^2, x_5^2 + x_5x_6 + x_6^2, x_5^2 + x_5x_7 + x_7^2 \rangle.$$

Una vez tenemos el ideal I computamos su base de Gröbner para resolver el problema. Se trata de encontrar en la base las “pistas” necesarias que nos condicionen la coloración del grafo de manera que se cumpla la 3-coloración.

```
R.<x1,x2,x3,x4,x5,x6,x7,x8> = PolynomialRing(QQ,8,order='lex')
G=ideal(x1^3 - 1,x2^3 - 1,x3^3 - 1,x4^3 - 1,x5^3 - 1,x6^3 - 1,x7^3 - 1,x8^3 - 1,x1^2 + x1*x2 + x2^2,x1^2 + x1*x7 + x7^2,x1^2 + x1*x8 + x8^2,x2^2 + x2*x8 + x8^2,x1^2 + x1*x3 + x3^2,x5^2 + x5*x3 + x3^2,x6^2 + x6*x3 + x3^2,x7^2 + x7*x3 + x3^2,x5^2 + x5*x4 + x4^2,x5^2 + x5*x6 + x6^2,x5^2 + x5*x7 + x7^2)
```

G

```
[x1 - x5, x2 + x5 + x8, x3 + x5 + x7, x4^2 + x4*x5 - x5*x8 - x8^2, x5^2 + x5*x8 + x8^2, x5*x7 - x5*x8 + x7^2 - x8^2, x6 - x7, x7^3 - 1, x8^3 - 1]
```

Figura 4-8 Código en SageMath para obtener la base de Gröbner del ideal I obtenido del grafo del ejemplo 4.7

La base de Gröbner obtenida es $G = \{x_8^3 - 1, x_7^3 - 1, x_6 - x_7, x_5x_7 + x_7^2 - x_5x_8 - x_8^2, x_5 + x_5x_8 + x_8^2, x_4^2 + x_4x_5 - x_5x_8 - x_8^2, x_3 + x_5 + x_7, x_2 + x_5 + x_8, x_1 - x_5\}$. De la base, tendremos en cuenta que $x_6 - x_7 \Rightarrow x_6 = x_7$, por tanto, ambos vértices tendrán el mismo color (rojo). Lo mismo ocurre con $x_1 - x_5 \Rightarrow x_1 = x_5$, ambos vértices tendrán color verde. Por otra parte, según las condiciones que vienen establecidas en [13], la suma de vértices, tal y como se da en $x_3 + x_5 + x_7$, indica que estos tres vértices son diferentes en color. Si al x_7 se le asignó el color rojo, y a x_5 el color verde, x_3 tendrá color azul. De la misma manera, $x_2 + x_5 + x_8$, por tanto, se puede asignar a x_2 el color rojo, mientras que x_8 se coloreará de azul. Únicamente queda por colorear el país 4, x_4 , que se podrá colorear de color azul, diferente a su único país frontera, el 5. Cabe destacar, que la base obtenida permite que este problema admita más de una solución, no obstante, aplicando lo desarrollado en el párrafo obtenemos el mapa coloreado de la siguiente manera:



Figura 4-9 Mapa de Sudamérica del ejemplo 4.7 3-coloreado

La principal aplicación de este tipo de problemas es en ámbitos similares al desarrollado en el Ejemplo 4.7. Para ello, cada país, comunidad autónoma o provincia, se representa como un vértice del grafo, y las aristas las fronteras comunes entre estos. Si ampliamos a variables mayores este problema, se podrían tener en cuenta las bases de Gröbner a la hora de resolver Sudokus, cuyas características y

requisitos de resolución son similares a los problemas de coloración de grafos, aunque un tanto más complejas. No obstante, se ha demostrado que estos problemas son optimizables si se utilizan las bases.

4.2.3 Resolución de forma implícita

Suponiendo una serie de ecuaciones paramétricas del tipo

$$(4.2.1) \quad \begin{aligned} x_1 &= f_1(t_1, \dots, t_m) \\ &\vdots \\ x_n &= f_n(t_1, \dots, t_m), \end{aligned}$$

se tratará de definir un subconjunto de una variedad algebraica V en k^n . Esta variedad no siempre puede venir dada por ecuaciones polinomiales igualadas a 0, como se observa en (4.2.1) donde las coordenadas de los puntos de una variedad afín recurren al uso de expresiones algebraicas en función de ciertos parámetros (t en el caso de (4.2.1)). Los problemas de implicitación tratan de transformar las expresiones paramétricas en ecuaciones implícitas que determinen todos los puntos dados por la parametrización. Las bases de Gröbner son de gran utilidad en este caso, a la hora de encontrar ecuaciones polinomiales capaces de definir V . Por ejemplo, en el caso en el que las expresiones $f_1, \dots, f_n$ sean polinomios, la variedad afín en k^{m+n} vendrá definida por el conjunto de expresiones:

$$\begin{aligned} x_1 &= f_1(t_1, \dots, t_m) = 0 \\ &\vdots \\ x_n &= f_n(t_1, \dots, t_m) = 0. \end{aligned}$$

La idea principal, motivo por el cual se utilizan las bases de Gröbner en este tipo de problemas, es la eliminación del parámetro t . Como se desarrolla en el Ejemplo 4.1, las bases de Gröbner nos permiten eliminar variables teniendo en cuenta el orden monomial que se utilice en cada caso, con tal de aplicar, posteriormente, la sustitución hacia atrás, por tanto, si establecemos un orden lexicográfico $t_1 > \dots > t_m > x_1 > \dots > x_n$ seremos capaces de eliminar el parámetro t de las ecuaciones planteadas resultando en las ecuaciones para V únicamente en función de $[x_1, \dots, x_n]$.

Ejemplo 4.8 [2] Considérese la curva paramétrica

$$\begin{aligned} x &= t^4, \\ y &= t^3, \\ z &= t^2. \end{aligned}$$

Para llevar a cabo lo desarrollado en el párrafo anterior, se debe computar la base de Gröbner del ideal $I = \langle x - t^4, y - t^3, z - t^2 \rangle$ con respecto al orden lexicográfico $t > x > y > z$. Utilizando SageMath obtenemos:

```
R.<t,x,y,z> = PolynomialRing(QQ,4, 't,x, y, z', order='lex')
G = ideal(x-t^4, y-t^3, z-t^2).groebner_basis()
```

G

```
[t^2 - z, t*y - z^2, t*z - y, x - z^2, y^2 - z^3]
```

Figura 4-10 Computación en SageMath de la base de Gröbner del ideal correspondiente a las expresiones paramétricas del Ejemplo 4.8

Analizando el resultado, observamos que los dos últimos polinomios están en $[x, y, z]$, por lo que definen una variedad afín que contiene la curva considerada.

Ejemplo 4.9 [2] Añadamos ahora un parámetro más, dada la superficie parametrizada

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

computemos su base de Gröbner correspondiente al ideal $I = \langle x - t - u, y - t^2 - 2tu, z - t^3 - 3t^2u \rangle$, respecto al orden lexicográfico $u > t > x > y > z$.

```
R.<t,u,x,y,z> = PolynomialRing(QQ,5, 't,u,x, y, z', order='lex')
G = ideal(x - t - u,y - t^2 - 2*t*u, z - t^3 - 3*t^2*u).groebner_basis()
```

G

```
[t + u - x, u^2 - x^2 + y, u*x^2 - u*y - x^3 + 3/2*x*y - 1/2*z, u*x*y - u*z - x^2*y - x*z + 2*y^2, u*x*z - u*y^2 + x^2*z - 1/2*x*y^2 - 1/2*y*z, u*y^3 - u*z^2 - 2*x^2*y*z + 1/2*x*y^3 - x*z^2 + 5/2*y^2*z, x^3*z - 3/4*x^2*y^2 - 3/2*x*y*z + y^3 + 1/4*z^2]
```

Figura 4-11 Computación en SageMath de la base de Gröbner del ideal correspondiente a las expresiones paramétricas del Ejemplo 4.9

En este caso, la base de Gröbner resultante es de mayor tamaño, debido a la presencia de un parámetro más que en el Ejemplo 4.8, no obstante, también realiza su función a la hora de deshacerse de los parámetros, t y u en este caso, como observamos en el último polinomio, únicamente en función de $[x, y, z]$.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

Con lo expuesto en el trabajo, se demuestra que las bases de Gröbner son una herramienta muy potente en cuanto a la resolución de todo tipo de problemas, casos y situaciones que conlleven la resolución de sistemas de ecuaciones. El proceso de convertir dichos sistemas en ideales para posteriormente computar su base es el cimiento sobre el cual se construyen las bases en la implementación del método, de ahí la importancia de conocer estos ideales con sus propiedades y saber desenvolverse con ellos. El algoritmo de Buchberger, a su vez, es clave a la hora de obtener, de un ideal, su base de Gröbner de manera directa utilizando el bucle del Teorema 3.2.7 que trabaja tanto con S-polinomios como con el algoritmo de la división hasta obtener una base con las características requeridas. De esta manera se obtiene un método alternativo para la resolución de sistemas efectivo pero, a su vez, desconocido por los estudiantes que tratan el álgebra de manera general, debido a la complejidad de los conceptos que conforman las bases, no obstante, su mención de manera general permitiría a los estudiantes conocer generalmente la existencia, tanto de las bases como herramienta que facilita la resolución de sistemas, como las plataformas algebraicas computacionales tales como SageMath o Macaulay2. Estos sistemas no tratan únicamente aspectos del álgebra abstracta, sino que, como se muestra en el apartado 2 y en los preliminares del presente documento, utilizan diferentes tipos de lenguaje informático (Python, C/C++ o Lisp) para resolver cuestiones de las matemáticas o el álgebra más simples. Es por ello, que introducir las bases de Gröbner puede ser un incentivo para introducir al alumnado el entorno del álgebra computacional y de esta manera ofrecerles una herramienta que sirva de ayuda en otras aplicaciones del ámbito académico en el que se desarrollan.

No obstante, el ejercicio mostrado en el apartado 4.1.2 muestra como un sistema con un elevado número de variables y ecuaciones puede conllevar una elevada complejidad computacional, así como un análisis detallado de cada uno de los polinomios que conforman la base, aumentando el tiempo y el esfuerzo requerido para obtener una solución coherente con lo planteado. Además, otro factor que puede afectar significativamente a la complejidad computacional son los sistemas que utilizan coeficientes inexactos (polinomios con coeficientes racionales o irracionales), debido a que los cálculos que se utilizan en el método son altamente sensibles a la precisión numérica pudiendo alterar significativamente los resultados. Por otra parte, a lo largo de los ejemplos mostrados se destaca la fuerte dependencia de las bases de Gröbner en el orden monomial, pudiendo este alterar por completo el rendimiento y la eficiencia del cálculo. Hoy en día, no se ha establecido ningún único orden de términos que funcione correctamente para todos los problemas planteados, y encontrar el orden idóneo para un problema supone una amplia experiencia en el campo, incluso algún ajuste manual en las expresiones planteadas.

5.2 Líneas futuras

Como consecuencia de lo mencionado en el apartado anterior, se realizan continuas revisiones y se trata de optimizar el algoritmo de Buchberger con tal de desarrollar algoritmos más eficientes y con mayor capacidad de memoria de manera que sean capaces de abarcar problemas más complejos. Cabe destacar que, a nivel computacional, conviene aplicar el concepto general de las bases de Gröbner de manera híbrida, combinando el algoritmo tratado en el trabajo con otros algoritmos o mejoras de este, como lo es el algoritmo de Faugère que utiliza una técnica similar al algoritmo de Buchberger añadiéndole un paso al bucle que elimina las computaciones redundantes, optimizando así el proceso total. Combinando el algoritmo con este tipo de técnicas, se pueden desarrollar métodos híbridos capaces de potenciar las diversas facetas de cada uno de estos algoritmos y de este modo conseguir computaciones de las bases de Gröbner más rápidas, fiables y eficientes. Tratando de perseguir esta tendencia de optimización y desarrollo, el campo de las bases de Gröbner puede continuar evolucionando con el propósito de abarcar problemas más complejos que puedan desarrollarse en ámbitos matemáticos, algebraicos, de ciencia computacional e ingeniería.

6 BIBLIOGRAFÍA

- [1] C. Larson, «On the Histories of Linear Algebra: The Case of Linear Systems,» de *Proceedings of the 13th Annual Conference on Research on Undergraduate Mathematics Education*, Indiana.
- [2] D. A.Cox, J. Little y D. O'Shea, *Ideals, Varieties, and Algorithms*, 4 th ed., New York: Springer - Verlag, 1996.
- [3] R. Holmes, *Gröbner Bases and Systems of Polynomial Equations*, Minnesota: Cornerstone, 2020.
- [4] J. Dorronsoro y E. Hernández, *Números, Grupos y Anillos*, Madrid: Adisson-Wesley Iberoamericana S.A., 1996.
- [5] L. Fernando, *!Qué bonita es la teoría de Galois!*, Madrid, 2004-2005.
- [6] E. M. Palacios, *Anillos de Polinomios*, Universidad de Granada, 2012.
- [7] P. L. William W. Adams, *An Introduction to Gröbner Bases*, vol. 3, E. Board, Ed., Providence, Rhode Island: American Mathematical Society, 1994.
- [8] M. J. D. HAYES, *Computing Gröbner Basis*, McGill University, 1996.
- [9] I. D. G. Rodríguez, *Bach-Flat Manifolds and Conformally Einstein Structures*, Santiago de Compostela: Departamento de Geometría y Topología, 2020.
- [10] A. Samanta, «Gröbner Basis and its Application in Integer Programming,» *American Journal of Applied Mathematics and Computing*, vol. 1, nº 4, pp. 12-16, 2021.
- [11] D. R., *Graph Theory*, Springer, Ed., Graduate Texts in Mathematics, 2010.
- [12] M. K. W., R. Bances y N. M. G., «Algunas Aplicaciones de las Bases de Gröbner,» Pontificia Universidad Católica del Perú, Lima, 2014.

- [13] D. A. Cox, *Gröbner Bases Tutorial*, Department of Mathematics and Computer Science, 2007.
- [14] V. Y. Pan, «SIAM Rev.,» de *Solving a polynomial equation: Some History and Recent Progress*, vol. 39, Society for Industrial and Applied Mathematics, 1997, pp. 188-220.
- [15] O. T. Olu, «On the direct and indirect methods of Solving Systems of Linear Equatons,» de *Scholars Journal of Physics Mathematics and Statistics*, vol. 5, Scholars Academic and Scientific Publishers (SAS Publishers) , 2018, pp. 334-341.
- [16] J. J. C. Rosas, M. G. Cárdenas y V. P. Morán, «Metodos Iterativos de Jacobi y Gauss-Seidel,» de *Plataforma educativa para el análisis numérico*, México, 2019.
- [17] Academic Press, «Bruno Buchberger and the World of Gröbner Bases,» *Journal of Symbolic Computation*, pp. 495-497, 2011.
- [18] S. Gharib, S. R. Ali, R. Khan y N. Munir, «System of Linear Equations, Gaussian Elimination,» *Global Journal of Computer Science and Technology*, vol. 15, nº 5, pp. 23-26, 2015.
- [19] F. M. Sabaté, *El Teorema Fundamental del Álgebra*, Barcelona: Universitat de Barcelona, 2005.
- [20] C. C. Pinter, *A book of Abstract Algebra*, 2 nd ed., Dover Books on Mathematics, 1990.
- [21] W. K. Nicholson, *Introduction to Abstract Algebra*, 4th ed., Wiley, 1992.
- [22] R. Duan y M. Chao, «Global identifiability of latent class models with applications to diagnostic test accuracy studies: A Gröbner Basis Approach,» *Biometrics Wiley*, pp. 98-108, 2020.
- [23] A. M. d. Campo, *Introducción al álgebra geometría computacional*, 2020.
- [24] W. W. Adams y P. Lousaunau, *An Introduction to Gröbner Bases*, vol. 3, American Mathematical Society, 1996.
- [25] M. Kauers, *Gröbner Bases and Applications*, Linz: Johannes Kepler University (JKU).
- [26] H. Calvete Fernández y P. Mateo Collazos, «Programación Lineal Entera y Meta. Problemas y Aplicaciones.,» *Prensas universitarias de Zaragoza*, 1994.
- [27] C. Wendler, *Gröbner Basis with an Applications to Integer Programming*, 2004.
- [28] S. A. G. Cruces, *Bases de Gröbner y Aplicaciones a Sistemas de Ecuaciones Polinomiales*, Chile: Universidad del Bío-Bío.

ANEXO I: IMPLICACIONES SOCIALES, ECONÓMICAS Y AMBIENTALES

Una vez desarrolladas las bases de Gröbner enfocadas mayormente a problemas que impliquen la resolución de sistemas de ecuaciones polinomiales, se deduce el potencial que estas poseen en cuanto a optimizar la computación de un problema que conlleve simplificar un sistema complejo que, previo a la existencia de estas bases, conllevaba un alto coste computacional. Mediante el algoritmo de Buchberger, y la eliminación de variables a la hora de resolver los sistemas, se ofrece un método capaz de simplificar las operaciones a realizar y, por tanto, reducir la complejidad computacional en comparación con otros métodos mencionados en el apartado 2 del presente documento, disminuyendo así tanto el tiempo de operación como los recursos empleados en el proceso. Además, el simplificar problemas de programación entera, resulta provechoso en problemas de optimización de materiales, recursos, horarios y asignación de tareas. Una empresa u organización que haga uso de esta herramienta tiene a su disposición un recurso que provocarán un aumento tanto de su productividad, como de un mejor uso y gestión de sus recursos, puede aplicado a gran escala y durante largos periodos temporales, se traduce en beneficios económicos. Por otra parte, el uso de las bases de Gröbner no solo reducen tiempo y consumo de recursos; el hecho de que estas proporcionen un método estricto y riguroso para la resolución de sistemas de ecuaciones complejos reduce la presencia de errores en el proceso de optimización y, consecuentemente, la necesidad de trabajar continuamente en análisis y mejora de software y diseño de sistemas. De esta manera, los fondos que se puedan destinar a la corrección de errores, revisión de programas o rediseño de sistemas se verá reducido.

La comprensión y correcta aplicación de esta herramienta requiere una formación y conocimiento avanzado en matemáticas y ciencias computacionales. Aunque a simple vista resulten una herramienta matemática abstracta, la aplicación de métodos basados en bases de Gröbner puede tener repercusión en lo relativo al desarrollo tecnológico y competencia internacional. Los países o empresas de investigación y desarrollo que lideren en este ámbito poseen una ventaja competitiva en áreas como la criptografía, la ingeniería de sistemas o la robótica. Además, el proceso de construcción de las bases, tal y como se realiza en el presente trabajo, puede servir de iniciador a la hora de introducir a los estudiantes de álgebra en el mundo de la computación, utilizando plataformas como SageMath, Singular o Macaulay2, facilitando así el acceso a las potentes herramientas computacionales disponibles hoy en día, viendo no solo lo útil de estas a la hora de resolver sistemas de ecuaciones, sino en diversos campos dentro del álgebra y la geometría.

Por otra parte, si consideramos la aplicación relativa a la optimización de procesos aplicados a la fabricación, se pueden encontrar implicaciones ambientales potenciales. Como se menciona anteriormente, esta mejora en el rendimiento a la hora de optimizar procesos relativos al ámbito industrial puede ahorrar energía y materiales, así como también reducir desperdicios generados en el procesado. Además, esta eficacia se puede aplicar al diseño de sistemas de energía renovable (parques eólicos o sistemas fotovoltaicos) optimizando tanto la disposición, configuración y transporte de los componentes que los conforman, minimizando así el impacto ambiental. Lo mismo puede aplicarse en planificación urbana, distribuyendo de la manera más eficiente el uso del suelo y las infraestructuras, de modo que se promueva un desarrollo urbano más sostenible.

ANEXO II. PLANTILLA-CÓDIGO EN SAGEMATH PARA RESOLVER UN SISTEMA DE ECUACIONES LINEALES UTILIZANDO LAS BASES DE GRÖBNER

En el presente anexo, se proporcionan códigos-plantilla a implementar en SageMath para resolver sistemas de ecuaciones lineales y problemas de programación entera. Para cada porción de código proporcionado en el anexo, se explicará línea por línea como rellenar numéricamente cada código de manera que la plataforma compute el código sin que devuelva ningún tipo de error.

Resolución de sistema de ecuaciones lineales

En primer lugar, se debe introducir el contexto en el que se va a trabajar, definiendo las variables que conforman el anillo de polinomios y el orden lexicográfico a seguir; antes de definir las ecuaciones que conforman el sistema. El código mostrado corresponde al utilizado en la Figura 4-2.

```
R.<x,y,z> = PolynomialRing(QQ,3, 'x, y, z', order='lex')
ec1 =
ec2 =
ec3 =
G = ideal(ec1, ec2, ec3).groebner_basis()
print("Base de Gröbner:", G)
```

Línea	Explicación
Línea 1	En la primera línea se define el entorno en el que se trabaja. En este caso, se definen de serie tres variables (x, y, z), pudiéndose añadir más si es necesario. Se indica que se trabaja en el anillo de números racionales (QQ), indicando finalmente el orden lexicográfico a seguir <i>lex</i> o <i>deglex</i> .
Líneas 2,3 y 4	En estas líneas se introducen las ecuaciones que conforman el sistema a resolver, pudiendo añadir tantas ec_n como sea necesario.
Línea 5	Se conforma el ideal compuesto por las tres ecuaciones del sistema, calculando su base de Gröbner reducida utilizando la función <code>groebner_basis()</code> .
Línea 6	Se muestra en pantalla la base de Gröbner reducida correspondiente al ideal.

Tabla 2. Tabla explicativa del código-plantilla correspondiente a la construcción de un ideal a partir de ecuaciones polinomiales y la computación de la respectiva base de Gröbner

Una vez calculada la base de Gröbner, se debe resolver el sistema de ecuaciones formado por los polinomios de esta. Debido a la complejidad que supone cargar un código en SageMath capaz de realizar una sustitución hacia atrás con un orden monomial establecido, se utiliza el comando `solve`. Esta porción de código es el utilizado en la resolución del ejemplo 4.1, en la Figura 4-3.

```
x,y,z = var('x,y,z')
g1 =
g2 =
g3 =
solve([g1,g2,g3],x,y,z)
```

Línea	Explicación
Línea 1	Se definen las variables introducidas en la porción de código anterior.
Líneas 2,3 y 4	En estas líneas se introducen los polinomios que conforman la base de Gröbner. Se pueden introducir tantos g_n como polinomios formen la base.
Línea 5	Se implementa el comando <i>solve</i> para resolver el sistema formado por los polinomios de la base, mostrando en pantalla de manera automática los resultados.

Tabla 3. Tabla explicativa del código-plantilla correspondiente a la resolución del sistema formado por los elementos de la base de Gröbner hallada

Aplicando estas dos fracciones de código sobre un sistema de ecuaciones se debe mostrar la solución, o soluciones, del sistema de ecuaciones tal y como se muestra en la Figura 4-3.

Programación Entera

De la misma manera que con los sistemas de ecuaciones lineales, se procederá con los problemas de programación entera. Como se desarrolla en el apartado 4.1.2 del presente trabajo, la primera parte de este tipo de problemas procede de la misma manera que a la hora de resolver sistemas de ecuaciones, hallando la base de Gröbner del ideal correspondiente al sistema, con el pequeño matiz de que primeramente debemos escribir el sistema según (4.1.3). Una vez reescrito el sistema, se conforma el ideal I añadiendo las variables $y_1, \dots, y_n$, tal y como se muestra en el Ejemplo 4.1.

```
R.<x1,x2,x3,y1,y2,y3> = PolynomialRing(QQ,6, 'x1,x2,x3,y1,y2,y3', order='lex')
G = ideal(I).groebner_basis()
print("Base de Gröbner:", G)
```

Línea	Explicación
Línea 1	De la misma manera que en el caso de sistemas de ecuaciones, se define el entorno en el que se trabaja. En este caso, se definen variables $x_1, \dots, x_n$ y $y_1, \dots, y_n$. Se indica que se trabaja en el anillo de números racionales (QQ), indicando finalmente el orden lexicográfico a seguir <i>lex o deglex</i> .
Línea 2	En esta línea se introduce en el lugar de I el ideal resultante de transformar el sistema en una expresión del tipo (4.1.3) y se halla la base de Gröbner correspondiente.
Línea 5	Se muestra en pantalla la base mencionada.

Tabla 4. Tabla explicativa correspondiente a la primera parte de la resolución de un problema de programación entera

Una vez hallada la base de Gröbner, únicamente queda dividir la expresión de los números de la parte derecha de la igualdad $x_1^{b_1} \cdot \dots \cdot x_n^{b_n}$ entre la misma base.

$$(x_1^{b_1} \cdot x_2^{b_2} \cdot x_3^{b_3}).\text{reduce}(G)$$

Línea	Explicación
Línea 1	Se utiliza el comando <i>reduce</i> para realizar la división correspondiente. Al implementar este código nos devolverá el resto de la división: una serie de variables y elevadas a números naturales.

Tabla 5. Tabla explicativa correspondiente a la muestra de la solución óptima de un problema de programación entera

Tal y como se muestra en la Figura 4-5 y Figura 4-6, el resultado de la división contiene la solución al problema en los exponentes de las y mostradas al ejecutar *reduce*.