Vol.8 / No.2 / Diciembre 2024, pp. 1-12
SahuarUS

Resolviendo ecuaciones polinomiales

Hernán de Alba Casillas[1] y Daniel Duarte[2]
[1]CONAHCYT-Unidad Académica de Matemáticas de la Universidad Autónoma de Zacatecas
[2]Centro de Ciencias Matemáticas, UNAM Campus Morelia
[1]halba@uaz.edu.mx, [2]adduarte@matmor.unam.mx

Resumen

En este artículo exploramos un método de resolución de ecuaciones polinomiales que tienen una cantidad finita de soluciones. El algoritmo para solucionar estas ecuaciones es similar al método de eliminación de Gauss en el sentido de que también convierte el sistema de ecuaciones original en otro que tiene forma escalonada. El algoritmo en cuestión está basado en un algoritmo de la división para polinomios en varias variables así como un algoritmo que permite eliminar términos especiales de un polinomio. Para concluir, comentamos algunas aplicaciones de los conceptos que aparecen en este artículo.

Palabras Clave: Sistemas de ecuaciones polinomiales, algoritmo de la división, algoritmo de Buchberger.

DOI: 10.36788/sah.v8i2.151
Recibido: 11 de mayo de 2024.
Aceptado: 01 de octubre de 2024.

1. Introducción

La resolución de ecuaciones es un tema central en las matemáticas. En nuestros primeros cursos de secundaria aprendemos a factorizar polinomios en una variable, lo que nos permite encontrar sus raíces. Más adelante nos encontramos con sistemas de ecuaciones lineales y las correspondientes técnicas para resolverlos. A diferencia de una ecuación polinomial en una variable, donde sólo hay una cantidad finita de soluciones, en álgebra lineal aprendemos que un sistema de ecuaciones podría tener una cantidad infinita de soluciones. El objetivo de este artículo es exponer un método de resolución de ecuaciones polinomiales que tienen una cantidad finita de soluciones.

Dado un sistema de ecuaciones lineales con una única solución, el método de eliminación de Gauss es un algoritmo que permite encontrarla. Recordemos que este método consiste en realizar operaciones elementales en la matriz asociada al sistema, lo que da lugar a una matriz escalonada. Este método funciona gracias a que las ecuaciones son lineales. ¿Cómo podríamos encontrar las soluciones de un sistema de polinomios no necesariamente lineales? Por ejemplo, ¿puede la lectora o el lector imaginar cómo se encuentran las soluciones del

ISSN 2448 5365
Universidad de Sonora

siguiente sistema?

$$\begin{aligned} 3x^7y - z^8 + 1 &= 0 \\ \sqrt{3}w^{15} - xyz^5 - x^2 &= 0 \\ 7y^{101}w^3 + \frac{2}{5}x^4 + \pi z^{10} &= 0 \\ x^2 + y^3 + z^5 + w^8 &= 0 \end{aligned}$$

Pese a ser una pregunta elemental, es sorprendente que los primeros algoritmos computacionales para resolver este tipo de ecuaciones no aparecieron sino hasta mediados del siglo XX (ver el prefacio de [2]). En este artículo buscamos ilustrar, a partir de ejemplos concretos, algunas ideas básicas en torno a un algoritmo de resolución de ecuaciones polinomiales. Como veremos, este algoritmo tiene similitudes importantes con respecto al método de Gauss: esencialmente se trata de reducir el sistema a otro que tiene *forma escalonada*.

2. Ecuaciones polinomiales en una variable

En esta primera sección hacemos un breve recordatorio del cálculo de las raíces de un polinomio en una variable.

Denotamos como $\mathbb{C}[x]$ al conjunto de los polinomios en la variable x con coeficientes en los números complejos. En secundaria aprendimos que los polinomios lineales $ax+b=0$, con $a \neq 0$, tienen una única raíz, a saber, $x = -\frac{b}{a}$. También nos enseñaron que los polinomios de grado 2, $ax^2 + bx + c = 0$, con $a \neq 0$, tienen una o dos raíces y se resuelven con la fórmula general $x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$. Para polinomios de grado 3 y 4 también existen fórmulas que calculan sus raíces, aunque son mucho más complicadas [13, Capítulo 5]. El problema de encontrar una fórmula que permitiera obtener las raíces de un polinomio de grado mayor o igual a 5 intrigó a la comunidad matemática por mucho tiempo. Hoy en día es bien sabido que tal fórmula no existe.

Teorema 1. *(***Teorema de Abel***). No existe una fórmula que involucre sumas, restas, multiplicaciones, divisiones, potencias o radicales que obtenga las raíces de todos los polinomios de grado mayor o igual que 5.*

El teorema de Abel muestra la imposibilidad de encontrar una fórmula que describa las raíces de cualquier polinomio. Por otro lado, un resultado clásico en álgebra nos asegura la existencia de tales raíces.

Teorema 2. *(***Teorema fundamental del álgebra***). Sea $f(x) \in \mathbb{C}[x]$ un polinomio de grado mayor o igual a 1. Entonces la ecuación $f(x) = 0$ tiene al menos una solución $p \in \mathbb{C}$.*

Tanto el teorema de Abel como el teorema fundamental del álgebra son resultados clásicos en álgebra y se pueden encontrar en muchas fuentes. Por citar algunas, se pueden consultar en [9, Capítulo 5] y [1, Capítulo 4].

Ahora bien, ante la falta de una fórmula general para encontrar las raíces de un polinomio, se tiene la opción de aproximarlas numéricamente. Esta es un área de las matemáticas que se ha desarrollado enormemente y que tiene importantes aplicaciones en problemas de la vida real.

3. Sistema de ecuaciones lineales

En esta sección hacemos un breve recordatorio del método de eliminación de Gauss para resolver ecuaciones lineales. Un sistema de m ecuaciones lineales con n incógnitas tiene la siguiente forma:

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m, \end{aligned}$$

donde para cada $i \in \{1, 2, \ldots, m\}$ y $j \in \{1, 2, \ldots, n\}$, $a_{ij} \in \mathbb{C}$ y $b_i \in \mathbb{C}$. Este sistema de ecuaciones se puede escribir como $AX = b$, donde

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}.$$

Recordemos el algoritmo para resolver estas ecuaciones. De entrada, diremos que dos sistemas de ecuaciones lineales $AX = b$ y $BX = c$ son equivalentes si tienen el mismo conjunto de soluciones. Ahora bien, dado un sistema de ecuaciones lineales $AX = b$ definimos la matriz aumentada $A|b$ como la matriz cuyas primeras n columnas son las columnas de A y cuya última columna es b. Decimos que $A|b$ es equivalente a $B|c$ si los sistemas de ecuaciones lineales correspondientes $AX = b$ y $BX = c$ son equivalentes.

Teorema 3. *[6, Sección 3.4] Sean $A|b$ y $B|c$ dos matrices de m renglones y $n+1$ columnas con entradas en $\mathbb{C}$. $A|b$ y $B|c$ son equivalentes si $B|c$ se obtuvo de $A|b$ al aplicar una de las siguientes operaciones fila:*

1. *Intercambio de dos filas de $A|b$.*
2. *Multiplicar una fila de $A|b$ por un escalar $c \in \mathbb{C}$, $c \neq 0$.*
3. *Reemplazar la r-ésima fila de $A|b$ por la fila r más la fila s multiplicada por un escalar $c \in \mathbb{C}$.*

Para resolver un sistema de ecuaciones lineales $AX = b$ debemos aplicar sucesivamente a la matriz aumentada $A|b$ operaciones de cualquiera de los tres tipos que se mencionan en el teorema 3. De esta manera se puede obtener un sistema de ecuaciones más sencillo, de donde es más fácil obtener las soluciones.

Veamos el siguiente ejemplo que ilustra el procedimiento anterior. Supongamos que queremos resolver el sistema lineal:

$$\begin{array}{rcrcrcl} x & + & y & + & z & = & 2, \\ x & + & y & - & z & = & 4, \\ x & + & 2y & + & 2z & = & 3. \end{array} \tag{1}$$

La matriz aumentada del sistema es:

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 1 & 1 & -1 & 4 \\ 1 & 2 & 2 & 3 \end{array}\right].$$

Aplicando las operaciones fila a la matriz anterior obtenemos

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 1 & 1 & -1 & 4 \\ 1 & 2 & 2 & 3 \end{array}\right] \to \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & 0 & -2 & 2 \\ 1 & 2 & 2 & 3 \end{array}\right] \to \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 1 & 2 & 2 & 3 \\ 0 & 0 & -2 & 2 \end{array}\right] \to \left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & -2 \end{array}\right]$$

Así el sistema de ecuaciones (1) es equivalente al sistema de ecuaciones:

$$\begin{cases} x + y + z = 2 \\ \quad\; y + z = 1 \\ \quad\quad\; 2z = -2. \end{cases}$$

Despejando y haciendo sustitución hacia atrás obtenemos que la solución es el punto $(1, 2, -1)$.

El procedimiento anterior es llamado el método de eliminación de Gauss, que consiste en aplicar sucesivamente operaciones fila a la matriz aumentada $A|b$ hasta convertirla en una matriz escalonada. Un tratamiento extenso de este tema se puede consultar en numerosas fuentes (ver, por ejemplo, [6, Capítulo 3]).

El siguiente enunciado es una reformulación del Teorema 3.14 en [6]. Lo escribimos de esta manera para ilustrar un primer caso de un resultado análogo que veremos más adelante, en el caso de ciertos sistemas de ecuaciones polinomiales (Teorema 16).

Teorema 4. *Dado un sistema de m ecuaciones lineales con n incógnitas existe un sistema de m' ecuaciones lineales con n incógnitas que tiene el mismo conjunto de soluciones, $m' \leq m$ y para cada $i \in \{2, \ldots, m'\}$ la ecuación i-ésima de este nuevo sistema de ecuaciones no tiene a las primeras $i-1$ variables como incógnitas.*

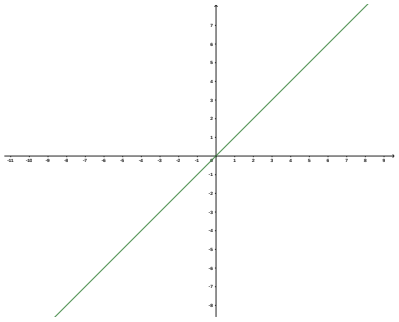
4. Sistemas de ecuaciones polinomiales

Denotamos al conjunto de polinomios en n variables con coeficientes en $\mathbb{C}$ como $\mathbb{C}[x_1, \ldots, x_n]$. Dada una cantidad finita de polinomios $f_1, \ldots, f_s \in \mathbb{C}[x_1, \ldots, x_n]$, queremos describir las soluciones del sistema de ecuaciones $\{f_1 = \cdots = f_s = 0\}$. En otras palabras, buscamos describir el conjunto

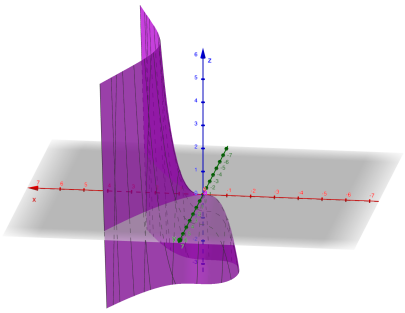
$$\{p \in \mathbb{C}^n \mid f_1(p) = \cdots = f_s(p) = 0\}.$$

A diferencia del caso de polinomios en una variable, donde solo el polinomio cero tiene infinitas raíces, en varias variables el conjunto de soluciones a una ecuación polinomial es típicamente infinito.

Por ejemplo: las soluciones de la ecuación $x - y = 0$ es una recta en $\mathbb{R}^2$. En las siguientes figuras podemos ver la forma de los conjuntos de soluciones de diferentes ecuaciones polinomiales.



$\{(x, y) \in \mathbb{R}^2 : x - y = 0\}$



$\{(x, y, z) \in \mathbb{R}^3 : x^3 - y^2 - z = 0\}$

Ahora bien, asumiendo que un sistema de ecuaciones polinomiales tiene una cantidad finita de soluciones, ¿cómo podemos encontrarlas? Antes vimos que el método de Gauss nos permite encontrar soluciones de ecuaciones lineales: el sistema original se remplaza por un sistema equivalente que tiene forma escalonada, es decir, hay una ecuación que depende sólo de la variable x_n, una segunda ecuación que depende sólo de las variables x_{n-1} y x_n, y así sucesivamente. Veamos que una idea similar también se puede usar para resolver ecuaciones polinomiales más generales.

4.1. Sistemas de ecuaciones polinomiales equivalentes

Para empezar veamos que, así como en álgebra lineal, podemos buscar sistemas equivalentes de ecuaciones polinomiales. En el siguiente resultado introducimos un objeto clave para tal efecto.

Lema 5. *Sean $f_1, \ldots, f_s$ polinomios en $\mathbb{C}[x_1, \ldots, x_n]$. Consideremos el conjunto*

$$\langle f_1, \ldots, f_s \rangle := \Big\{ \sum_{i=1}^{s} h_i f_i \mid h_i \in \mathbb{C}[x_1, \ldots, x_n] \Big\}.$$

Dado $p \in \mathbb{C}^n$, se tiene que $f_1(p) = \cdots = f_s(p) = 0$ si y sólo si $F(p) = 0$ para cada $F \in \langle f_1, \ldots, f_s \rangle$.

Demostración. *Si $p \in \mathbb{C}^n$ es tal que $f_i(p) = 0$ para toda i, entonces*

$$\Big(\sum_{i=1}^{s} h_i f_i \Big)(p) = \sum_{i=1}^{s} h_i(p) f_i(p) = 0.$$

La otra implicación es consecuencia de que $\{f_1, \ldots, f_s\} \subset \langle f_1, \ldots, f_s \rangle$.

Este lema tiene el siguiente importante corolario.

Corolario 6. *Si $\langle f_1, \ldots, f_s \rangle = \langle g_1, \ldots, g_r \rangle$, entonces las soluciones del sistema $f_1 = \cdots = f_s = 0$ son iguales a las soluciones del sistema $g_1 = \cdots = g_r = 0$.*

Notemos que, en efecto, podría suceder que $\langle f_1, \ldots, f_s\rangle = \langle g_1, \ldots, g_r\rangle$ para distintas colecciones de polinomios $\{f_i\}$ y $\{g_j\}$. Por ejemplo, un cálculo directo permite mostrar que $\langle x+y, x\rangle = \langle y, x, x^2\rangle$.

Enseguida vamos a explorar un ejemplo del cálculo de las soluciones de un sistema de ecuaciones polinomiales.

Ejemplo 7. *Queremos encontrar las soluciones del siguiente sistema:*

$$\begin{aligned} f_1 &:= xy - z = 0, \\ f_2 &:= yz - x = 0, \\ f_3 &:= xz - y = 0. \end{aligned} \tag{2}$$

Así como en el algoritmo de Gauss, vamos a intentar simplificar este sistema. Para ello, consideramos las siguientes combinaciones de f_1, f_2 y f_3:

$$\begin{aligned} f_4 &:= f_1 + yf_2 = y^2z - z, \\ f_5 &:= zf_1 - yf_3 = y^2 - z^2, \\ f_6 &:= f_4 - zf_5 = z^3 - z. \end{aligned} \tag{3}$$

Por construcción tenemos que $\{f_4, f_5, f_6\} \subset \langle f_1, f_2, f_3\rangle$. En particular,

$$\langle f_1, f_2, f_3\rangle = \langle f_1, f_2, f_3, f_4, f_5, f_6\rangle,$$

por lo que nuestro sistema es equivalente a $f_1 = f_2 = f_3 = f_4 = f_5 = f_6 = 0$. Ahora bien, ¿qué ganamos al cambiar el sistema $f_1 = f_2 = f_3 = 0$ por $f_1 = f_2 = f_3 = f_4 = f_5 = f_6 = 0$? De entrada podríamos tener la impresión de que sólo complicamos aún más el sistema. Afortunadamente, no es el caso. Notemos que en el nuevo sistema hay algunos polinomios especiales: f_6, que depende sólo de la variable z, y f_5, que depende de las variables y, z. Esto nos permite encontrar las soluciones del sistema.

En efecto, primero buscamos las soluciones de $f_6 = 0$, es decir, $z = 0$, $z = 1$ o $z = -1$. Enseguida evaluamos cada una de estas raíces en f_5 (también podría ser f_4) y encontramos los valores respectivos para y, esto es: $(0, 0)$, $(1, 1)$, $(-1, 1)$, $(1, -1)$ y $(-1, -1)$. Ahora usamos estos valores para encontrar las soluciones en la variable x. Para esto, evaluamos en el polinomio f_2. Así, las soluciones del sistema $f_6 = f_5 = f_2 = 0$ son

$$S := \{(0, 0, 0), (1, 1, 1), (-1, -1, 1), (-1, 1, -1), (1, -1, -1)\}.$$

Finalmente, tendríamos que eliminar los elementos de este conjunto que no son ceros de f_1, f_3 y f_4. En este caso, estos elementos también son solución de $f_1 = f_3 = f_4 = 0$. Concluimos que

$$\{p \in \mathbb{C}^3 | f_1(p) = f_2(p) = f_3(p) = 0\} = S.$$

Observaciones 8. *Queremos enfatizar los siguientes aspectos del ejemplo 7:*

- *Para resolver el sistema $f_1 = f_2 = f_3 = 0$, lo sustituimos por un sistema con más ecuaciones, a saber, $f_1 = f_2 = f_3 = f_4 = f_5 = f_6 = 0$.*
- *La construcción de f_4, f_5 y f_6 en (3) se efectuó buscando cancelar ciertos términos de los polinomios con los que empezamos.*

- *Gracias a la cancelación descrita en el punto anterior, se lograron construir polinomios en menos variables, concretamente, un polinomio en z y otros en y, z.*
- *El problema de encontrar las soluciones al sistema se redujo entonces a buscar raíces de polinomios en una variable.*

La observación 8 ilustra un algoritmo general para resolver sistemas de ecuaciones polinomiales con una cantidad finita de soluciones. Como se menciona en la observación, un paso clave en el ejemplo 7 consistió en cancelar ciertos términos de los polinomios en cuestión. En la siguiente subsección explicamos cómo hacer esto de manera sistemática, para cualquier sistema de ecuaciones polinomiales.

4.2. Algoritmo de Buchberger

De entrada, necesitamos introducir dos conceptos: el orden lexicográfico para ordenar los monomios de un polinomio y un algoritmo de la división en $\mathbb{C}[x_1, \ldots, x_n]$.

Definición 9. *Sean $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n), \boldsymbol{\beta} = (\beta_1, \ldots, \beta_n) \in \mathbb{Z}^n_{\geq 0}$. Decimos que $\boldsymbol{\alpha} >_{lex} \boldsymbol{\beta}$ si en la diferencia $\boldsymbol{\alpha} - \boldsymbol{\beta} \in \mathbb{Z}^n$ la entrada no cero que esté más a la izquierda es positiva. Dados dos monomios $x^{\boldsymbol{\alpha}} = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ y $x^{\boldsymbol{\beta}} = x_1^{\beta_1} \cdots x_n^{\beta_n}$, diremos que $x^{\boldsymbol{\alpha}} >_{lex} x^{\boldsymbol{\beta}}$ si $\boldsymbol{\alpha} >_{lex} \boldsymbol{\beta}$.*

Sea $f = a_{\boldsymbol{\alpha_1}} x^{\boldsymbol{\alpha_1}} + \cdots + a_{\boldsymbol{\alpha_r}} x^{\boldsymbol{\alpha_r}} \in \mathbb{C}[x_1, \ldots, x_n]$, con $a_{\boldsymbol{\alpha_i}} \neq 0$ para cada i. Llamamos el término líder de f al término $a_{\boldsymbol{\alpha_i}} x^{\boldsymbol{\alpha_i}}$, donde $\boldsymbol{\alpha_i}$ es el vector más grande respecto al orden lexicográfico. Lo denotamos $\mathrm{LT}(f)$.

Ejemplo 10. *Sean $\boldsymbol{\alpha} = (5, 2, 1)$ y $\boldsymbol{\beta} = (5, 1, 3)$. Entonces $x^{\boldsymbol{\alpha}} = x_1^5 x_2^2 x_3 >_{lex} x_1^5 x_2 x_3^3 = x^{\boldsymbol{\beta}}$ puesto que $\boldsymbol{\alpha} - \boldsymbol{\beta} = (0, 1, -2)$. Así, si $f = 5x_1^5 x_2 x_3^3 - 3x_1^5 x_2^2 x_3$, entonces $\mathrm{LT}(f) = -3x_1^5 x_2^2 x_3$.*

Antes de describir el algoritmo de la división en $\mathbb{C}[x_1, \ldots, x_n]$, presentamos un ejemplo para familiarizarse con el procedimiento. Como veremos, este algoritmo no es muy distinto del algoritmo de la división clásico. El objetivo es dividir un polinomio entre un conjunto finito de polinomios.

Sean $f = x^2y + xy^2 + y^2$ y $f_1 = y^2 - 1, f_2 = xy - 1$. Nuestro propósito es dividir f entre $\{f_1, f_2\}$, es decir, expresar $f = a_1 f_1 + a_2 f_2 + r$ para algunos polinomios a_1, a_2 y r, donde r es un residuo.

Notemos que los monomios de los polinomios f, f_1 y f_2 están ordenados de manera decreciente bajo el orden lexicográfico. En particular, $\mathrm{LT}(f) = x^2y$, $\mathrm{LT}(f_1) = y^2$ y $\mathrm{LT}(f_2) = xy$. Vamos a realizar la siguiente división:

$$\begin{array}{rl} a_1: & \\ a_2: & \\ y^2 - 1 & \overline{|x^2y + xy^2 + y^2} \\ xy - 1 & \end{array}$$

Empezamos comparando el término líder de f entre el término líder de f_1. Si lo divide continuamos como en el algoritmo en una variable. De lo contrario, pasamos al término líder de f_2. En este caso,

$\mathrm{LT}(f)$ no es divisible por $\mathrm{LT}(f_1)$ pero sí por $\mathrm{LT}(f_2)$. De esta manera,

$$\begin{array}{rl} a_1: & \\ a_2: & \quad x \\ y^2-1 & \quad \overline{|x^2y+xy^2+y^2} \\ xy-1 & \quad \underline{x^2y-x} \\ & \qquad xy^2+x+y^2 \end{array}$$

Ahora consideremos el término líder de xy^2+x+y^2, *i.e.* xy^2. Repetimos el procedimiento. Como $\mathrm{LT}(f_1)$ divide a xy^2 se obtiene

$$\begin{array}{rl} a_1: & \quad x \\ a_2: & \quad x \\ y^2-1 & \quad \overline{|x^2y+xy^2+y^2} \\ xy-1 & \quad \underline{x^2y-x} \\ & \qquad xy^2+x+y^2 \\ & \qquad \underline{xy^2-x} \\ & \qquad\quad 2x+y^2 \end{array}$$

Notemos que el término líder de $2x+y^2$, *i.e.* $2x$, no es divisible por los términos líderes de f_1 y f_2. El algoritmo de la división clásico se detendría en este momento. Por el contrario, para el algoritmo de la división en varias variables, el siguiente paso consiste en pasar el término $2x$ al residuo. Ahora comparamos el siguiente término, es decir, y^2. Repetimos el procedimiento para obtener:

$$\begin{array}{rlcl} a_1: & \quad x+1 & & \underline{\textit{Residuo}} \\ a_2: & \quad x & & \\ y^2-1 & \quad \overline{|x^2y+xy^2+y^2} & & \\ xy-1 & \quad \underline{x^2y-x} & & \\ & \qquad xy^2+x+y^2 & & \\ & \qquad \underline{xy^2-x} & & \\ & \qquad\quad 2x+y^2 & & \\ & \qquad\quad \underline{y^2} & \longrightarrow & 2x \\ & \qquad\quad \underline{y^2-1} & & \\ & \qquad\qquad \underline{1} & & \\ & \qquad\qquad 0 & \longrightarrow & 2x+1 \end{array}$$

El algoritmo dio lugar a la siguiente igualdad:

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

El algoritmo previo se expresa de manera general como sigue (ver [2, Capítulo 2]).

Proposición 11. *(Algoritmo de la división) Consideremos el orden lexicográfico para ordenar los términos de polinomios en* $\mathbb{C}[x_1,\dots,x_n]$. *Sea* $F=\{f_1,\dots,f_s\}\subset\mathbb{C}[x_1,\dots,x_n]$. *Entonces cada* $f\in\mathbb{C}[x_1,\dots,x_n]$ *se puede escribir como*

$$f=a_1f_1+\cdots+a_sf_s+r,$$

donde $a_i,r\in\mathbb{C}[x_1,\dots,x_n]$. *Además, si* $r\neq 0$ *entonces* r *se escribe como una* $\mathbb{C}$*-combinación lineal de monomios que no son divisibles por ningún* $\mathrm{LT}(f_i)$, $i=1,\dots,s$. *Decimos que* r *es el residuo de la división de* f *por* F. *Denotamos el residuo como* $\bar{f}^F$.

Enseguida definimos el objeto clave que permite hacer las cancelaciones de las que hablamos en la sección anterior. Recordemos que dados dos monomios $x^{\boldsymbol{\alpha}}$ y $x^{\boldsymbol{\beta}}$, su mínimo común múltiplo es el monomio $x^{\boldsymbol{\gamma}}$, donde las entradas del vector $\boldsymbol{\gamma}$ están definidas por $\gamma_i=\max\{\alpha_i,\beta_i\}$. Lo denotamos como $mcm(x^{\boldsymbol{\alpha}},x^{\boldsymbol{\beta}})$.

Definición 12. *Sean* $f,g\in\mathbb{C}[x_1,\dots,x_n]$ *dos polinomios distintos de cero. Sean* $\mathrm{LT}(f)=a_{\boldsymbol{\alpha}}x^{\boldsymbol{\alpha}}$ *y* $\mathrm{LT}(g)=b_{\boldsymbol{\beta}}x^{\boldsymbol{\beta}}$ *(ver definición 9). Sea* $x^{\boldsymbol{\gamma}}$ *el mínimo común múltiplo de* $x^{\boldsymbol{\alpha}}$ *y* $x^{\boldsymbol{\beta}}$. *Definimos el* S*-polinomio de* f *y* g *como*

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

Por construcción, un S-polinomio cancela términos líderes. En el siguiente ejemplo ilustramos esta propiedad. Como la lectora o el lector podrá apreciar, este ejemplo ya apareció en la sección anterior (ver la construcción de f_5 en el ejemplo 7).

Ejemplo 13. *Sean* $f=xy-z, g=xz-y\in\mathbb{C}[x,y,z]$. *Notemos que* $\mathrm{LT}(f)=xy$ *y* $\mathrm{LT}(g)=xz$. *Además,* $xyz=mcm(xy,xz)$. *Así,*

$$\begin{aligned}S(f,g)&=\frac{xyz}{xy}f-\frac{xyz}{xz}g\\&=zf-yg\\&=-z^2+y^2.\end{aligned}$$

Notemos que en este ejemplo el S*-polinomio es un polinomio que no contiene la variable* x *aunque* f *y* g *sí la contenían.*

Con los ingredientes introducidos hasta ahora, estamos listos para describir el algoritmo que nos permitirá simplificar sistemas de ecuaciones polinomiales.

Algoritmo 14. ***(Algoritmo de Buchberger)***

INPUT*: Un conjunto de polinomios* $F=\{f_1,\dots,f_s\}\subset\mathbb{C}[x_1,\dots,x_n]$.

OUTPUT*: Un conjunto de polinomios* $G=\{g_1,\dots,g_r\}$ *tales que* $F\subset G$ *y para cada* $h\in\langle F\rangle$ *existe* $g\in G$ *tal que* $\mathrm{LT}(g)$ *divide a* $\mathrm{LT}(h)$ *y* $\langle F\rangle=\langle G\rangle$.

El conjunto G *se construye haciendo primero* $G=F$ *y después iterando una cantidad finita de veces los siguientes pasos:*

1. *Fijamos una pareja de elementos de* G *y calculamos su* S*-polinomio.*
2. *Dividimos este* S*-polinomio entre* G.

3. *Si el residuo es cero volvemos al paso 1 con otra pareja de G. De lo contrario, dicho residuo se agrega a G y volvemos al paso 1.*

4. *Se repiten 1, 2 y 3 hasta que todos los residuos de los S-polinomios de G sean cero.*

Una versión más detallada del algoritmo y su demostración se pueden consultar en [10, Sección 1.3]. Veamos un ejemplo para ilustrarlo.

Ejemplo 15. *Sea $F = \{f_1 = x^2y - 1, f_2 = xy^2 - x\}$. Como paso inicial hacemos $G = F$. Ahora, calculamos el S-polinomio $S(f_1, f_2) = x^2 - y$. Dividiendo $S(f_1, f_2)$ por F obtenemos el siguiente residuo:*

$$\overline{S(f_1, f_2)}^{G} = x^2 - y \neq 0$$

Agregamos $f_3 = x^2 - y$ a G. De esta forma, hacemos $G = \{f_1, f_2, f_3\}$. Calculamos un nuevo S-polinomio y su residuo respectivo,

$$S(f_1, f_3) = y^2 - 1, \qquad \overline{S(f_1, f_3)}^{G} = y^2 - 1.$$

Agregamos este nuevo residuo $f_4 = y^2 - 1$ a G. Así tenemos que ahora $G = \{f_1, f_2, f_3, f_4\}$. Calculando S-polinomios y dividiendo obtenemos que todos los residuos son cero. El algoritmo se detiene en este momento arrojando $G = \{f_1 = x^2y - 1, f_2 = xy^2 - x, f_3 = x^2 - y, f_4 = y^2 - 1\}$.

Este ejemplo ilustra nuevamente lo que sucedió en el ejemplo 7: los polinomios f_1 y f_2 contienen las variables x y y y con ciertas operaciones logramos construir el polinomio f_4 que sólo depende de la variable y. Estos ejemplos son casos particulares del siguiente teorema.

Teorema 16. *[8, Sección 1.8.5] Sean $f_1, \ldots, f_s \in \mathbb{C}[x_1, \ldots, x_n]$ tales que el sistema $f_1 = \cdots = f_s = 0$ tiene un número finito de soluciones. Sea $\{g_1, \ldots, g_r\}$ el conjunto de polinomios resultante del algoritmo 14. Entonces $r \geq n$ y los polinomios g_i satisfacen lo siguiente:*

$$\begin{aligned}
&g_n \in \mathbb{C}[x_n], && \mathrm{LT}(g_n) = x_n^{a_n},\\
&g_{n-1} \in \mathbb{C}[x_{n-1}, x_n], && \mathrm{LT}(g_{n-1}) = x_{n-1}^{a_{n-1}}\\
&\vdots && \vdots\\
&g_2 \in \mathbb{C}[x_2, \ldots, x_{n-1}, x_n], && \mathrm{LT}(g_2) = x_2^{a_2}\\
&g_1 \in \mathbb{C}[x_1, x_2, \ldots, x_{n-1}, x_n], && \mathrm{LT}(g_1) = x_1^{a_1}.
\end{aligned}$$

En particular, el sistema $g_1 = \cdots = g_r = 0$ se puede resolver encontrando las soluciones de $g_n(x_n) = 0$, enseguida sustituirlas en $g_{n-1}(x_{n-1}, x_n) = 0$ y encontrar las soluciones respecto a la variable x_{n-1}, y así sucesivamente hasta $g_1 = 0$. Finalmente, se deben descartar las soluciones de $g_1 = \cdots = g_n = 0$ que no son soluciones de $g_{n+1} = \cdots = g_r = 0$.

Ejemplo 17. *Encontremos las soluciones del siguiente sistema de ecuaciones:*

$$\begin{aligned}
f_1 &:= x^2y - 1 = 0,\\
f_2 &:= xy^2 - x = 0.
\end{aligned}$$

De acuerdo al ejemplo 15, el algoritmo 14 arroja $G = \{x^2y - 1, xy^2 - x, x^2 - y, y^2 - 1\}$. Ordenamos los elementos de G como sigue: $G = \{g_1 = y^2 - 1, g_2 = x^2 - y, g_3 = xy^2 - x, g_4 = x^2y - 1\}$. Así los

números r y n del teorema 16 son $r = 4$ y $n = 2$. Ahora, buscamos las soluciones de $g_1 = 0$, es decir, $y = 1$, $y = -1$. Enseguida evaluamos cada una de estas raíces en $g_2 = 0$ y encontramos los valores respectivos para x, esto es: $(1, 1)$, $(-1, 1)$, $(i, -1)$, $(-i, -1)$. A continuación verificamos si estas soluciones satisfacen $g_3 = 0$ y $g_4 = 0$. Evaluando g_3 y g_4 en esas posibles soluciones observamos que se satisface $g_3 = 0$ y $g_4 = 0$, por lo que en virtud del teorema 16 concluimos

$$\{p \in \mathbb{C}^2 | f_1(p) = f_2(p) = 0\} = \{(1, 1), (-1, 1), (i, -1), (-i, -1)\}.$$

5. Breve comentario sobre las bases de Gröbner

El conjunto que se construye en el algoritmo 4.2 es llamado una base de Gröbner de $\langle f_1, \ldots, f_s \rangle$ respecto al orden lexicográfico. En realidad, existe la noción de base de Gröbner respecto a cualquier *orden monomial*, es decir, un orden entre los monomios de $\mathbb{C}[x_1, \ldots, x_n]$ que es total, es compatible con la multiplicación y es un buen orden. Recomendamos ampliamente las referencias [2, 10] para una introducción a este importante concepto de álgebra conmutativa. La lectora o el lector también podrá encontrar una gran cantidad de artículos introductorios a este tema en línea. Además existen programas computacionales que realizan cálculos explícitos usando este concepto. Por citar algunos: Singular o Macaulay2 [4, 7]. Ambos pueden utilizarse de manera gratuita en línea.

El concepto de base de Gröbner tiene una enorme cantidad de aplicaciones, tanto en las matemáticas como en otras áreas. Por citar algunos ejemplos, las bases de Gröbner han sido utilizadas en problemas de geometría algebraica, de teoría de gráficas, de estadística algebraica, de programación entera, de teoría de códigos, de criptografía, de robótica, de procesamiento de señales, entre otros. Algunas de estas aplicaciones pueden consultarse en [5, 10, 11, 3, 12].

Agradecimientos

Agradecemos enormemente a las revisoras o los revisores de este artículo. La cuidadosa lectura y las múltiples observaciones que recibimos nos ayudaron a mejorar sustancialmente la presentación de este trabajo. El primer autor agradece el apoyo del proyecto CONAHCYT A1-S-30482. El segundo autor agradece el apoyo del proyecto CONAHCYT CF-2023-G33.

Referencias

[1] L. V. Ahlfors, *Complex analysis*. McGraw-Hill New York, 1979.

[2] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, ser. Undergraduate texts in mathematics. Springer, 1997.

[3] J. A. De Loera, R. Hemmecke, and M. Köppe, *Algebraic and geometric ideas in the theory of discrete optimization*. SIAM, 2012.

[4] W. Decker, G. Greuel, G. Pfister, and H. Schönemann, "Singular 4-3-0—a computer algebra system for polynomial computations. http://www.singular.uni-kl.de," 2022.

[5] V. Ene and J. Herzog, *Gröbner bases in commutative algebra*. American Mathematical Soc., 2011, vol. 130.

[6] S. H. Friedberg, A. J. Insel, and L. E. Spence, *Linear Algebra: Pearson New International Edition*. Pearson Higher Ed, 2013.

[7] D. R. Grayson and M. E. Stillman, "Macaulay2, a software system for research in algebraic geometry available at http://www. math. uiuc. edu," 1992.

[8] G.-M. Greuel, G. Pfister, O. Bachmann, C. Lossen, and H. Schönemann, *A Singular introduction to commutative algebra*. Springer, 2008, vol. 348.

[9] I. N. Herstein, *Topics in algebra*. John Wiley & Sons, 1991.

[10] T. Hibi, *Gröbner Bases: Statistics and Software Systems*. Springer, 2013.

[11] M. Husty and P. Zsombor-Murray, "On the use of gröbner bases in a robotics course," in *New Trends in Educational Activity in the Field of Mechanism and Machine Theory: 2014-2017*. Springer, 2019, pp. 20–28.

[12] M. Sala, T. Mora, L. Perret, S. Sakata, and C. Traverso, *Gröbner Bases, Coding, and Cryptography*. Springer Berlin Heidelberg, 2009.

[13] J. V. Uspensky, *Teoría de ecuaciones*. Limusa, México, 2006.

Como citar este artículo: H. de Alba Casillas y D. Duarte, "Resolviendo ecuaciones polinomiales", Sahuarus. Revista Electrónica de Matemática, vol. 8, no. 2, pp. 1–12, 2024. https://doi.org/10.36788/sah.v8i2.151