



UNIVERSIDAD
DE GRANADA

ETSIIT, Facultad de Ciencias

DOBLE GRADO EN ING. INFORMÁTICA Y MATEMÁTICAS

TRABAJO DE FIN DE GRADO

Visualización de Superficies 3D

Presentado por:

Daniel Zufri Quesada

Tutor:

Carlos Ureña Almagro

Departamento de Lenguajes y Sistemas Informáticos

Pedro A. García Sánchez

Departamento de Álgebra

Curso académico 2022-2023

Índice general

1	Función Distancia con Signo	1
1.1	Operaciones sobre SDF	1
1.2	Raymarching y Spheretracing	2
1.3	Iluminación	3
1.3.1	Cálculo del vector normal	4
2	Polinomios en varias variables	7
2.1	Bases de Groebner	9
2.2	Implicitación Polinomial	11
	Bibliografía	13

1 Función Distancia con Signo

Definición 1.1. Sea $\Omega \subseteq \mathbb{R}^3$ un espacio métrico. Una **función distancia** $d(x)$ es aquella que a cada punto de \mathbb{R}^3 le asigna su menor distancia a la frontera de Ω :

$$d(x) = \min(|x - x_i|), \forall x_i \in \partial\Omega$$

Definición 1.2 (SDF). Una **función distancia con signo** es una función implícita $\phi(x): \mathbb{R}^3 \rightarrow \mathbb{R}$ de forma que:

$$\phi(x) = \begin{cases} d(x) & , x \in \mathbb{R}^3 \setminus \Omega \\ -d(x) & , x \in \Omega \end{cases}$$

De forma general nos referiremos a esta función por sus siglas en inglés SDF (*Signed Distance Function*).

Definición 1.3. Dada una función $\phi: \mathbb{R}^3 \rightarrow \mathbb{R}$, llamamos **isosuperficie** al conjunto:

$$S_\phi = \{(x, y, z) : \phi(x, y, z) = k, k \in \mathbb{R}\}$$

Tomaremos $k = 0$.

Nuestra intención será entonces construir una escena definida como la isosuperficie generada por un SDF.

Ejemplo 1.1. Algunos ejemplos simples de SDF para un punto p son:

- Esfera de radio r : $|p| - r$.
- Plano con vector normal n : $p \cdot n$.
- Toro de radios $R > r$: $\|(\|(p_x, 0, p_z)\| - R, p_y)\| - r$.

1.1. Operaciones sobre SDF

Si bien estas primitivas son fáciles de generar, también son muy simples y nos serán insuficientes si queremos construir escenas más complejas. Una de las ventajas de los SDF es que se pueden generar nuevas formas a partir de primitivas de forma muy sencilla.

Proposición 1.1 (Operaciones Booleanas). Sean A, B isosuperficies generadas por ϕ, γ respectivamente. Entonces:

- $sdf(A \cup B) = \min(\phi, \gamma)$.
- $sdf(A \cap B) = \max(\phi, \gamma)$.
- $sdf(A \setminus B) = \max(\phi, -\gamma)$.

Observación 1.1. Solo en el caso de la unión se obtiene un SDF según lo establecido en Def. 1.2, ya que en el interior (donde $\phi < 0$), estas transformaciones producen valores inesperados. En nuestro caso solo estamos interesados en visualizar la frontera de las superficies, así que podemos obviar este problema.

Un problema de usar estas transformaciones es que produce discontinuidades en la derivada del SDF resultante, lo cual preferimos evitar, además de por motivos analíticos, por motivos visuales, ya que el resultado de estas operaciones produce bordes muy acusados en la intersección de ambas superficies. Existen muchas formas de combinar SDF de forma más natural. Una de las más sencillas es la interpolación lineal.

Proposición 1.2 (Operaciones Booleanas Suavizadas). Sean A, B isosuperficies generadas por ϕ, γ respectivamente y $k \in \mathbb{R}$ un coeficiente de suavizado. Entonces:

- $sdf(A \cup B) = li(\gamma, \phi, h_1) - k \cdot h_1 \cdot (1 - h_1).$
- $sdf(A \cap B) = li(\gamma, \phi, h_2) + k \cdot h_2 \cdot (1 - h_2).$
- $sdf(A \setminus B) = li(\phi, -\gamma, h_3) + k \cdot h_3 \cdot (1 - h_3).$

donde:

- $li(x, y, a) = (1 - a) \cdot x + a \cdot y$ interpola linealmente x e y según el parámetro $a \in \mathbb{R}$.
- $c(x) = \min(\max(x, 0), 1)$ acota x en $[0, 1]$.
- $h_1 = c\left(\frac{1}{2} + \frac{\gamma - \phi}{2k}\right).$
- $h_2 = c\left(\frac{1}{2} - \frac{\gamma - \phi}{2k}\right).$
- $h_3 = c\left(\frac{1}{2} - \frac{\gamma + \phi}{2k}\right).$

Introducimos por último otro tipo de operaciones que nos resultarán útiles para generar nuevas formas a partir de un único SDF.

TODO

1.2. Raymarching y Spheretracing

Una vez definida la escena necesitamos un método para visualizarla. Utilizaremos la técnica de *raymarching*. Para ello deberemos elegir un punto donde estará el observador. A partir de él se trazarán rayos hacia cada uno de los puntos de un plano imaginario que llamaremos plano de visión, de forma que cada punto de este plano corresponde a un pixel de la imagen resultante. Si el rayo interseca con S_ϕ significa que ese pixel corresponde a un punto de S_ϕ y será coloreado como tal.

Para comprobar esta intersección utilizamos un método iterativo: a partir de la posición del observador, en cada iteración avanzamos en la dirección del rayo una distancia fija δ . Evaluamos entonces nuestro SDF en la posición actual. Si este valor es muy cercano a 0 significará que estamos en la isosuperficie. De lo contrario, repetiremos el proceso hasta encontrar la intersección o superar el número máximo de iteraciones, en cuyo caso concluiremos que

no hay intersección.

TODO: diagrama

Sin embargo podemos mejorar esta técnica, ya que cuanto más alejados estén los puntos de S_ϕ del observador, mayor es el número de iteraciones necesarias para encontrar la intersección. Esto es debido a que si no queremos perder ninguna intersección, el valor de incremento δ deberá de ser pequeño. La solución a este problema es el uso de *spheretracing*. Su funcionamiento es similar al *raymarching*, con la diferencia de que el incremento en la posición del rayo no es fija, sino que es la máxima que podemos tomar asegurándonos de no perder información en cada iteración. Esta distancia será entonces la distancia mínima del punto actual del rayo a S_ϕ , que no es más que evaluar ϕ en dicho punto.

TODO: diagrama de pérdida de intersección

El algoritmo sería por tanto el descrito en 1.

Algorithm 1: Spheretracing

```

Data: origen  $p$ , dirección  $v$ 
incremento  $\leftarrow 0$ ;
distanciaTotal  $\leftarrow 0$ ;
for  $i \in \text{MAX\_ITERACIONES}$  do
    rayo  $\leftarrow p + \text{distanciaTotal} \cdot v$ 
    distanciaActual  $\leftarrow \phi(\text{rayo})$ 
    if distanciaActual  $< \epsilon$  then
        | return DibujarSuperficie(rayo);
    end
    distanciaTotal  $+= \text{distanciaActual}$ 
    if distanciaTotal  $> \text{MAX\_DISTANCIA}$  then
        | return DibujarFondo();
    end
end
  
```

1.3. Iluminación

Ya sabemos qué píxeles pertenecen a la superficie, pero no de qué color debe ser dicho pixel. Para ello utilizaremos el modelo de reflexión de Phong. En él trabajaremos con los siguientes parámetros:

- Luz ambiente i_a : vector RGB suma de las contribuciones de todas las fuentes de luz de la escena. Contiene un valor por defecto que será el color del fondo.
- Lista de fuentes de luz $(l_1, \dots, l_m, \dots, l_n)$.
- Materiales asignados a cada objeto definidos por las constantes:
 - Reflexión especular k_s : modela cómo se refleja la luz en objetos brillantes. Depende tanto de la posición de las fuentes de luz como de la del observador.
 - Reflexión difusa k_d : modela cómo se refleja la luz en objetos mates. Depende de la posición de las fuentes de luz, pero no de la posición del observador.

1 Función Distancia con Signo

- Reflexión ambiental k_a : representa la proporción de luz del ambiente que refleja el objeto.
- Coeficiente de brillo α : valor real positivo que permite variar el tamaño de las zonas brillantes (a mayor valor, menor tamaño y más pulida o especular).
- Los vectores normalizados definidos para cada punto de la superficie p :
 - L_m : vector director desde p a cada fuente de luz l_m .
 - N vector normal a la superficie en p .
 - R_m dirección del rayo de luz reflejado especularmente desde la fuente l_m en el punto p .
 - V : dirección de p a la posición del observador.

TODO: diagrama vectores

Con estas variables, el color en p vendrá dado por la fórmula:

$$I_p = k_a i_a + \sum_{m=0}^n (k_d (L_m \cdot N) + k_s (R_m \cdot V)^\alpha)$$

De los vectores anteriores, L_m y V se calculan trivialmente, y R_m se puede obtener como

$$R_m = 2(L_m \cdot N)N - L_m$$

Veamos como obtener N .

1.3.1. Cálculo del vector normal

Definición 1.4. El **gradiente** de una función implícita $\phi: \mathbb{R}^3 \rightarrow \mathbb{R}$ es $\nabla\phi = \left(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z}\right)$.

Proposición 1.3. $\nabla\phi$ es perpendicular a S_ϕ .

Demostración. Sea $s \in S_\phi$ arbitrario. Tomamos una curva parametrizada:

$$\begin{aligned}\alpha: [0, 1] &\rightarrow S_\phi \\ t &\mapsto (x(t), y(t), z(t))\end{aligned}$$

cumpliendo $\alpha(t_0) = s$. Veamos que $\nabla\phi(s) \perp \alpha$:

$$\begin{aligned}\alpha(t) \subset S_\phi &\implies \phi(\alpha(t)) = 0 \\ \implies \nabla\phi(\alpha(t)) &= \frac{\partial F}{\partial x} \frac{dx}{dt} + \frac{\partial F}{\partial y} \frac{dy}{dt} + \frac{\partial F}{\partial z} \frac{dz}{dt} = 0 \\ \implies \langle \nabla\phi(x, y, z), \alpha'(t) \rangle &= 0 \implies \langle \nabla\phi(x, y, z), \alpha'(t_0) \rangle = 0\end{aligned}$$

Por tanto, $\nabla\phi(s)$ es perpendicular al vector tangente de α en s , que a su vez está contenida en el plano tangente de S_ϕ en s . Por tanto $\nabla\phi(s) \perp S_\phi$. \square

Hemos visto que calcular N equivale a calcular $\nabla\phi$. Sin embargo, dado que la superficie viene dada por un SDF que podría ser no derivable, no podemos hacerlo de forma analítica.

De ser derivable, podría ser una buena opción calcular el gradiente una única vez al momento de definir la ϕ , tras lo cual para obtener N solo habría que realizar evaluaciones de dicho gradiente. Sin embargo, por su sencillez y popularidad, nos decantaremos por un método numérico que únicamente hará uso de evaluaciones de ϕ .

Definición 1.5. Dada $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}$ diferenciable, $p = (x, y, z)$, $v \in \mathbb{R}^3$, definimos la **derivada direccional** en p con dirección v a:

$$\nabla_v f(p) = \nabla f(p) \cdot v = \frac{\partial f(p)}{\partial x} v_x + \frac{\partial f(p)}{\partial y} v_y + \frac{\partial f(p)}{\partial z} v_z$$

Para el cálculo de cada parcial de f podemos utilizar la definición de derivada. Por ejemplo, para la primera componente:

$$\frac{\partial f(p)}{\partial x} v_x = \lim_{h \rightarrow 0} \frac{f(p + (h, 0, 0)) - f(p)}{h} v_x$$

Procedemos a calcular N aplicaremos el **método del tetraedro**, el cual se basa en evaluar ϕ en la dirección de los vértices de un tetraedro:

$$k_0 = (1, -1, -1) \quad k_1 = (-1, -1, 1) \quad k_2 = (-1, 1, -1) \quad k_3 = (1, 1, 1)$$

Proposición 1.4 (Método del tetraedro). Dado $p \in S_\phi$, su vector normal N se obtiene normalizando el vector

$$\hat{N} = \sum_{i=0}^3 k_i \cdot f(p + hk_i)$$

Demostración. Por la proposición **Proposición 1.3**, basta comprobar que \hat{N} es colineal a $\nabla \phi(p)$.

$$\begin{aligned} \hat{N} &= \sum_{i=0}^3 k_i \cdot f(p + hk_i) = \sum_{i=0}^3 k_i \cdot f(p + hk_i) + k_i \cdot f(p) = \sum_{i=0}^3 k_i \cdot f(p + hk_i - f(p)) = \sum_{i=0}^3 k_i \nabla_{k_i} f(x) \\ &= \sum_{i=0}^3 k_i \cdot (k_i \cdot \nabla f(p)) = \sum_{i=0}^3 (k_i \cdot k_i) \nabla f(p) = \sum_{i=0}^3 \nabla f(p) = 4 \nabla f(p) \end{aligned}$$

donde hemos usado que $\sum_{i=0}^3 k_i = (0, 0, 0)$, $\sum_{i=0}^3 k_i \cdot k_i = (1, 1, 1)$, y que el producto escalar es un operador lineal. \square

2 Polinomios en varias variables

Fijamos A cuerpo y $X = \{x_1, \dots, x_n\}$ un conjunto de variables distintas.

Definición 2.1. Llamamos **monomio** en X al producto de la forma:

$$x_1^{\alpha_1} \cdots x_n^{\alpha_n}, \quad \alpha_i \in \mathbb{N}, \quad \forall 0 \leq i \leq n$$

Lo denotaremos como X^α , y diremos que $\alpha \in \mathbb{N}^n$ es el **exponente** del monomio.

Definición 2.2. Definimos un **polinomio** en X con coeficientes en A a toda combinación lineal finita de monomios:

$$\sum_{\alpha \in \mathbb{N}^n} a_\alpha X^\alpha, \quad a_\alpha \in A$$

Proposición 2.1. El conjunto de polinomios, que denotaremos $A[X] = A[x_1, \dots, x_n]$, es un cuerpo con las operaciones:

- Suma heredada de A : $(f+g)(\alpha) = f(\alpha) + g(\alpha), \quad \forall f, g \in A[X]$.
- Producto de convolución: $(fg)(\alpha) = \sum_{\beta+\gamma=\alpha} f(\beta)g(\gamma), \quad \forall f, g \in A[X]$.

Nos surge la pregunta de si hay alguna forma “natural” de ordenar los monomios que forman un polinomio, al igual que ocurre para polinomios de una sola variable. Primero tenemos que especificar qué es lo que consideramos “natural”.

Definición 2.3. Un **orden admisible** es un orden total \leq sobre \mathbb{N}^n cumpliendo:

1. $(0, \dots, 0) \leq \alpha, \quad \forall \alpha \in \mathbb{N}^n$.
2. $\alpha < \beta \implies \alpha + \gamma < \beta + \gamma, \quad \forall \alpha, \beta, \gamma \in \mathbb{N}^n$.

Siempre supondremos que usamos un orden admisible, luego podemos ordenar los monomios que conforman un polinomio ordenando sus exponentes según cualquier dicho orden. Hay multitud de órdenes entre los que elegir. Algunos de los más importantes son:

Ejemplo 2.1. Definimos el **orden lexicográfico** \leq_{lex} como:

$$\alpha \leq_{\text{lex}} \beta \iff \begin{cases} \alpha = \beta \\ \text{ó} \\ \alpha_i < \beta_i, \text{ donde } i \text{ es el primer índice tal que } \alpha_i \neq \beta_i \end{cases}$$

Definición 2.4. Llamamos **grado** de $\alpha \in \mathbb{N}^n$ a $|\alpha| = \sum_{i=1}^n \alpha_i + \dots + \alpha_n$.

Ejemplo 2.2. Definimos el **orden lexicográfico graduado inverso** $\leq_{\text{degrevlex}}$ como:

$$\alpha \leq_{\text{degrevlex}} \beta \iff \begin{cases} |\alpha| < |\beta| \\ \text{ó} \\ |\alpha| = |\beta| \text{ y } \alpha_i > \beta_i, \text{ donde } i \text{ es el último índice tal que } \alpha_i \neq \beta_i \end{cases}$$

Una vez obtenida la noción de orden admisible, estamos en disposición de definir varios conceptos que nos resultarán imprescindibles.

Definición 2.5. Sea $f = \sum_{\alpha \in \mathbb{N}^n} a_{\alpha} X^{\alpha}$ y \leq admisible. Definimos:

- **Exponente:** $\exp(f) = \max_{\leq} \alpha$.
- **Monomio líder:** $\text{lm}(f) = X^{\exp(f)}$
- **Coefficiente líder:** $\text{lc}(f) = a_{\exp(f)}$.
- **Término líder:** $\text{lt}(f) = \text{lc}(f) \cdot \text{lm}(f)$.

Teorema 2.1. Sea $F = \{f_1, \dots, f_s\} \subset A[X]$. Entonces todo polinomio $f \in A[X]$ se puede expresar como:

$$f = q_1 f_1 + \dots + q_s f_s + r$$

donde $q_i, r \in A[X]$ y r no se puede expresar como combinación lineal de ningún elemento de F o es 0. Notaremos $\text{rem}(f, [F]) = r$.

En otras palabras, podemos dividir f entre los polinomios f_1, \dots, f_s para expresar f como combinación lineal de elementos de F . Sin embargo, esta descomposición no será única, y dependerá del orden que ocupen los polinomios en F . Podemos calcular esta descomposición usando **Algoritmo 2**.

Algorithm 2: División polinomios varias variables

```

Data:  $f, F = [f_1, \dots, f_s]$ 
 $p \leftarrow f$ ;
 $[q_1, \dots, q_s] \leftarrow [0, \dots, 0]$ ;
 $r \leftarrow 0$ ;
while  $p \neq 0$  do
    divisorEncontrado  $\leftarrow false$ ;
    for  $f_i \in F$  do
        if  $\exp(p) = \exp(f_i) + \alpha$  then
             $q_i \leftarrow q_i + \frac{\text{lc}(p)}{\text{lc}(f_i)} X^{\alpha}$ ;
             $p \leftarrow p - f_i \cdot \frac{\text{lc}(p)}{\text{lc}(f_i)} X^{\alpha}$ ;
            divisorEncontrado  $\leftarrow true$ ;
        end
    end
    if !divisorEncontrado then
         $r \leftarrow r + \text{lt}(p)$ ;
         $p \leftarrow p - \text{lt}(p)$ ;
    end
end
return  $[r, q_1, \dots, q_s]$ 

```

Demostración. TODO

□

2.1. Bases de Groebner

Definición 2.6. Dado un anillo conmutativo R , decimos que $\emptyset \neq I \subseteq R$ es un **ideal** si:

1. $a + b \in I, \forall a, b \in I$.
2. $ab \in I, \forall a \in I, \forall b \in R$.

Lo denotaremos como $I \leq R$.

Definición 2.7. Dado $F \subseteq A$, el **ideal generado** por F es:

$$\langle F \rangle = \{a_1 f_1 + \dots + a_s f_s : a_1, \dots, a_s \in A, f_1, \dots, f_s \in F\}$$

Definición 2.8. Dado $I \leq A[X]$, diremos que $G = \{g_1, \dots, g_s\} \subseteq I$ es una **base de Groebner** para I si $\langle \text{lt}(I) \rangle = \langle \text{lt}(g_1), \dots, \text{lt}(g_t) \rangle$.

Definición 2.9. Dada $f \in A[X]$, definimos su **soporte** como

$$\text{supp}(f) = \{\alpha \in \mathbb{N}^n : f(\alpha) \neq 0\}$$

Definición 2.10. G se dirá una **base de Groebner reducida** para I si para todo $g \in G$ se cumple:

- $\text{lc}(g) = 1$.
- $\text{supp}(g) \cap (\exp(G \setminus \{g\}) + \mathbb{N}^n) = \emptyset$.

La base de Groebner es a un ideal lo que un sistema de generadores a un espacio vectorial: un subconjunto a partir del cual podemos obtener el total. Si además la base es reducida, esta será el equivalente a la base del espacio vectorial. Por tanto, resulta de gran interés saber si dado un ideal I siempre existirá una base de Groebner asociada, y de ser así, si podemos calcularla.

Definición 2.11. Dados $\alpha, \beta \in \mathbb{N}^n$, definimos:

- **mínimo común múltiplo:** $\text{lcm}(\alpha, \beta) = \{\max(\alpha_1, \beta_1), \dots, \max(\alpha_n, \beta_n)\}$.
- **máximo común divisor:** $\text{gcd}(\alpha, \beta) = \{\min(\alpha_1, \beta_1), \dots, \min(\alpha_n, \beta_n)\}$.

Definición 2.12. Sean $f, g \in A[X]$ donde $\alpha = \exp(f)$, $\beta = \exp(g)$, $\gamma = \text{lcm}(\alpha, \beta)$. Definimos el **S-polinomio** de f y g como

$$S(f, g) = \text{lc}(g)X^{\gamma-\alpha}f - \text{lc}(f)X^{\gamma-\beta}g$$

Teorema 2.2 (Primer Criterio de Buchberger). Sean $I \leq A[X]$ y $G = \{g_1, \dots, g_t\}$ conjunto de generadores de I . Entonces:

$$G \text{ es base de Groebner para } I \iff \text{rem}(S(g_i, g_j), G) = 0, \forall 1 \leq i < j \leq t$$

El algoritmo que usaremos para el cálculo de la base de Groebner se basará en este criterio. Sin embargo, antes de presentarlo, estudiamos dos criterios adicionales que lo harán más eficiente evitando el cálculo de S-polinomios innecesarios, que es la operación más costosa del algoritmo al suponer la realización de numerosas divisiones.

Teorema 2.3 (Criterios de Buchberger). Sean $I \leq A[X]$ y $g_1, g_2 \in G \subseteq A[X]$. Si se cumple cualquiera de las siguientes condiciones:

1. $\text{lcm}(g_1, g_2) = \text{lm}(g_1)\text{lm}(g_2)$.
 2. $\exists f \in G: \text{lm}(f) | \text{lcm}(g_1, g_2)$ y se verifica alguna de las condiciones:
 - a) algún $S(g_i, f) \xrightarrow{G} 0$.
 - b) $\text{lm}(f) | \frac{\text{lm}(g_i)}{\gcd(g_1, g_2)}$ y $\text{sm}(g_j)\text{lm}(f) \neq \text{sm}(f)\text{lm}(g_j)$.
- donde $i, j \in \{1, 2\}$ e $i \neq j$.

Entonces $S(g_1, g_2) \xrightarrow{G} 0$

Aplicando estos criterios, obtenemos **Algoritmo 3**.

Algorithm 3: Algoritmo de Buchberger optimizado

Data: $f, F = [f_1, \dots, f_s]$
 $G \leftarrow F$;
repeat
 $G' \leftarrow G$;
 for each pair $\{f, g\} \subseteq G'$ **do**
 if !Criterio 1(f, g) **AND** !Criterio 2(f, g, G') **then**
 $r \leftarrow \text{rem}(S(f, g), G')$;
 if $r \neq 0$ **then**
 $G \leftarrow G \cup \{r\}$;
 end
 end
 end
until $G' = G$;
return G

Observación 2.1. Una decisión necesaria y que aún no hemos mencionado es la de cómo se eligen las parejas $\{f, g\}$. Uno de los métodos más usados es la conocida como **estrategia normal**, debido a su simpleza y haber probado ser de las que completan más rápido el algoritmo. Esta consiste en tomar el par f, g cuyo $\text{lcm}(f, g)$ sea del menor grado posible según el orden admisible usado.

Ahora ya somos capaces de calcular una base de Groebner de todo ideal $I \leq A[X]$. No obstante, nos gustaría que siempre que fuera posible, esta fuera reducida. Veamos que siempre existe esta base reducida y cómo obtenerla.

Definición 2.13. Sea $M \leq \mathbb{N}^n$. Decimos que A es un **conjunto generador minimal** de M si:

$$M = a + \mathbb{N}^n \quad \wedge \quad M \neq (A \setminus \{a\}) + \mathbb{N}^n, \quad \forall a \in A$$

Lema 2.1. Todo ideal tiene un único conjunto generador minimal.

Teorema 2.4. Todo ideal I admite una única base de Groebner reducida para un orden admisible dado.

Demostración. **Existencia** Sea G un conjunto generador minimal de $\exp(I)$, que sabemos que existe por **Lema 2.1**. Sea $g \in G$ y $r = \text{rem}(g, [G \setminus \{g\}])$. Tenemos que:

$$\exp(g) \notin \exp(G \setminus \{g\} + \mathbb{N}^n) \implies \exp(g) = \exp(r) \implies \exp(G) = \exp((G \setminus \{g\}) \cup \{r\})$$

Como $g - r \in \langle G \setminus \{g\} \rangle \subseteq I$, tenemos que $r \in I$. Por tanto $G' = (G \setminus \{g\}) \cup \{r\}$ es base de Groebner de I y cumple $\text{supp}(r) \cap (\exp(G' \setminus \{r\}) + \mathbb{N}^n) = \emptyset$. Aplicando este procedimiento a cada $g \in G$ obtenemos una base reducida de I .

Unicidad Sean G_1, G_2 dos bases minimales de I . Por **Lema 2.1**:

$$\exp(G_1) = \exp(G_2) \implies \forall g_1 \in G_1, \exists! g_2 \in G_2 : \exp(g_1) = \exp(g_2)$$

Como $g_2 - g_1 \in I \implies \text{rem}(g_1 - g_2, G_1) = 0$. Por otro lado:

$$\left. \begin{aligned} \text{supp}(g_1 - g_2) &\subseteq (\text{supp}(g_1) \cup \text{supp}(g_2)) \setminus \{\exp(g_1)\} \\ \text{supp}(g_i) \setminus \{\exp(g_i)\} \cap (\{\exp(G_i) + \mathbb{N}^n\}) &= \emptyset, i \in \{1, 2\} \end{aligned} \right\} \implies \text{supp}(g_1 - g_2) \cap (\exp(G_1) + \mathbb{N}^n) = \emptyset$$

$$\implies \text{rem}(g_1 - g_2, G_1) = g_1 - g_2 \implies g_1 = g_2 \implies G_1 = G_2$$

□

De esta demostración podemos obtener **Algoritmo 4** para reducir una base de Groebner cualquiera a su respectiva minimal:

Algorithm 4: Minimización de base de Groebner

Data: G base a minimizar

$G \leftarrow F$;

foreach $g \in G$ **do**

$g \leftarrow g / \text{lc}(g)$;

$r \leftarrow \text{rem}(g, [G \setminus \{g\}])$;

if $r \neq 0$ **then**

$g \leftarrow r$;

end

end

2.2. Implicación Polinomial

Definición 2.14. Dado $F \subseteq A[X]$, llamamos **variedad afín** definida por F al conjunto:

$$\mathbb{V}(F) = \{(a_1, \dots, a_n) \in A^n : f_i(a_1, \dots, a_n) = 0, \forall 1 \leq i \leq s\}$$

Definición 2.15. Dado $I \leq A[x_1, \dots, x_n]$, diremos que su **ideal de l -eliminación** es:

$$I_l = I \cap A[x_{l+1}, \dots, x_n] \leq A[x_{l+1}, \dots, x_n]$$

Definición 2.16. Decimos que un orden admisible \leq es un **orden de l -eliminación** si

$$\beta \leq \alpha \implies \beta \in \mathbb{N}_l^n, \forall \alpha \in \mathbb{N}_l^n, \forall \beta \in \mathbb{N}^n$$

donde $\mathbb{N}_l^n = \{\alpha \in \mathbb{N}^n : \alpha_i = 0, 1 \leq i \leq l\}$.

Teorema 2.5 (Eliminación). Sea $I \leq A[x_1, \dots, x_n]$ y G una base de Groebner suya respecto a un orden \leq de l -eliminación. Entonces, una base de Groebner para I_l viene dada por:

$$G_l = G \cap A[x_{l+1}, \dots, x_n]$$

Teorema 2.6 (Implicitación Polinomial). Dados $f_1, \dots, f_n \in A[t_1, \dots, t_r]$ con A cuerpo infinito, sea

$$\begin{aligned} \phi: A^r &\rightarrow A^n \\ (a_1, \dots, a_r) &\mapsto (f_1(a_1, \dots, a_r), \dots, f_n(a_1, \dots, a_r)) \end{aligned}$$

Definimos los ideales:

- $I = \langle x_n - f_n, \dots, x_1 - f_1 \rangle \leq A[t_1, \dots, t_r, x_1, \dots, x_n]$
- $J = I \cap A[x_1, \dots, x_n]$ el ideal de r -eliminación de I

Entonces, $\mathbb{V}(J)$ es la menor variedad que contiene a $\phi(A^r)$.

Bibliografía