



# Matemáticas Esenciales

para diseño computacional

**Rajaa Issa**

Robert McNeel & Associates

Traducción al español:

Jose Luis García del Castillo López

Fernando García del Castillo López



Essential Mathematics for Computational Design by Robert McNeel & Associates is licensed under a [Creative Commons Attribution-Share Alike 3.0 United States License](https://creativecommons.org/licenses/by-sa/3.0/).

## **Introducción**

*Matemáticas Esenciales para Diseño Computacional* pretende servir como introducción a los profesionales del diseño en los conceptos matemáticos básicos que son necesarios para el desarrollo de métodos computacionales para modelado en 3D y gráficos diseñados por ordenador. Esto no quiere ser una fuente completa y exhaustiva de información, sino más bien un repaso de los conceptos básicos más usados.

El material está enfocado hacia diseñadores que tienen poca o ninguna experiencia en matemáticas más allá de la escuela secundaria. Todos los conceptos son explicados visualmente usando Grasshopper® (GH), el entorno de modelado generativo para Rhinoceros® (Rhino). Para obtener más información, vaya a [www.rhino3d.com](http://www.rhino3d.com) y [www.grasshopper3d.com](http://www.grasshopper3d.com)

El contenido está dividido en tres partes. En la primera, se tratan las matemáticas vectoriales, incluyendo representación de vectores, operaciones vectoriales y las ecuaciones de la recta y el plano. En la segunda, se hace un repaso sobre operaciones y transformaciones matriciales. La tercera parte incluye una revisión general de las curvas paramétricas, con especial hincapié en las curvas NURBS y los conceptos de continuidad y curvatura, así como una rápida revisión general de las superficies NURBS y polisuperficies.

Me gustaría agradecer la exhaustiva y excelente revisión técnica del Dr. Greg Arden de Robert McNeel and Associates. Sus valiosos comentarios fueron fundamentales para producir esta segunda edición. También quiero agradecer a la Sra. Margaret Becker, de Robert McNeel and Associates por revisar la redacción técnica y el formato del documento. Por último, me gustaría señalar que el material contenido en este libro se basa, en parte, en un taller que se celebró en la Universidad de Texas en Arlington para el evento Tex-Fab, en Febrero de 2010.

**Rajaa Issa**

Robert McNeel & Associates

Traducción al español:

**Jose Luis García del Castillo López**

**Fernando García del Castillo López**

[www.garciadelcastillo.es](http://www.garciadelcastillo.es)

## Índice

<b>1 Matemáticas vectoriales</b> .....	<b>1</b>
Representación de un vector .....	1
Operaciones vectoriales .....	3
Ecuación vectorial de la recta .....	14
Ecuación vectorial del plano .....	16
<b>2 Matrices y transformaciones</b> .....	<b>18</b>
Introducción .....	18
Multiplicación de matrices .....	18
Transformaciones afines .....	19
<b>3 Curvas y superficies paramétricas</b> .....	<b>25</b>
Introducción .....	25
Curvas cúbicas .....	25
Continuidad geométrica .....	27
Curvatura .....	29
Algoritmos para la evaluación de curvas paramétricas .....	31
Curvas NURBS .....	34
Características de las curvas NURBS .....	36
Superficies NURBS .....	39
Características de las superficies NURBS .....	40
Polisuperficies .....	42
<b>Referencias</b> .....	<b>45</b>

# 1 Matemáticas vectoriales

## Representación de un vector

Los vectores representan un parámetro que tiene “dirección” y “magnitud”, tales como la velocidad o la fuerza. Los vectores en un sistema de coordenadas 2D se representan con dos números reales de la forma:

$$\mathbf{v} = \langle a_1, a_2 \rangle$$

De manera similar, en un sistema de coordenadas 3D, los vectores se representan con tres números reales, tal que:

$$\mathbf{v} = \langle a_1, a_2, a_3 \rangle$$

Vamos a utilizar letras minúsculas en negrita para representar los vectores. Así mismo, los componentes de un vector se representan entre corchetes angulares. Para los puntos se utilizan letras en mayúsculas. Las coordenadas de los puntos se representarán entre paréntesis.

Utilizando un sistema de coordenadas determinado y una colección de puntos origen dentro de ese sistema, podemos representar y visualizar estos vectores dibujándolos como el segmento de una línea. Normalmente dibujamos también la punta de una flecha para representar la dirección de los vectores.

Por ejemplo, si tenemos un vector cuya directriz es paralela al eje x de un sistema de coordenadas dado y una magnitud de 5.18 unidades, entonces podríamos escribir el vector de la siguiente manera:

$$\mathbf{v} = \langle 5.18, 0, 0 \rangle$$

Para representar este vector, necesitamos un punto base y un sistema de coordenadas. Por ejemplo, en la siguiente figura, todos los segmentos rojos son representaciones iguales del mismo vector.

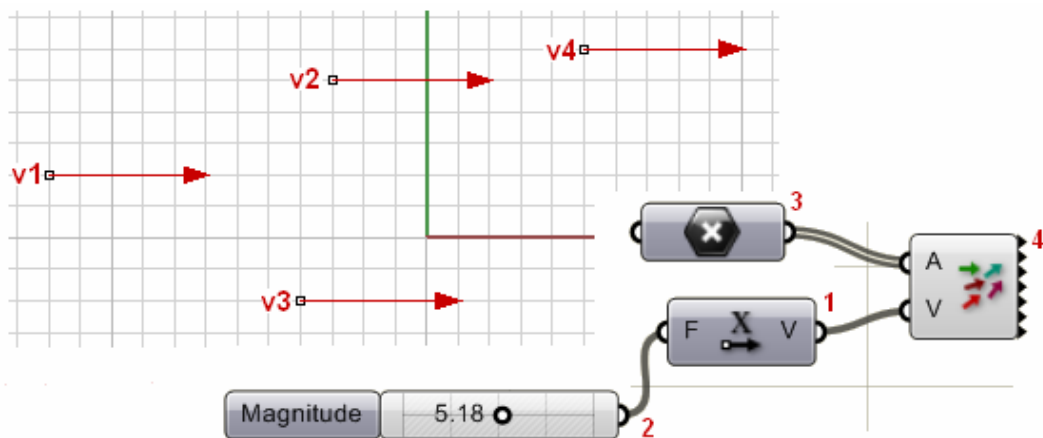


Figura (1): Representación de vectores en un sistema de coordenadas tridimensional. 1: Componente *unit x-axis* (vector unitario en la dirección x) de Grasshopper. 2: Componente *number slider* (deslizador numérico) de Grasshopper. 3: Componente *point* (punto), programado para referenciar varios puntos en Rhino (en este caso  $\mathbf{v1}$ ,  $\mathbf{v2}$ ,  $\mathbf{v3}$  y  $\mathbf{v4}$ ). 4: Componente *vector display* (representación de vector) de Grasshopper.

Dado un vector tridimensional  $\mathbf{v} = \langle a_1, a_2, a_3 \rangle$ , todos los componentes  $a_1, a_2$  y  $a_3$  del vector son números reales. Además, TODOS los segmentos de línea desde un punto  $A(x,y,z)$  a otro punto  $B(x+a_1, y+a_2, z+a_3)$  son representaciones EQUIVALENTES del vector  $\mathbf{v}$ .

Por lo tanto, ¿cómo definiríamos los puntos extremos de un segmento de línea que represente un vector dado? Definamos un punto base (P0) utilizando el componente  $x,y,z$  point (punto x, y, z) de Grasshopper:

$$P_0 = (1,2,3)$$

Y un vector, utilizando el componente  $xyz$  vector (vector xyz) de Grasshopper, el cual necesita tres números reales como entrada.

$$\mathbf{v} = \langle 2,2,2 \rangle$$

El punto extremo (P1) del vector se calcula sumando los componentes correspondientes al punto base y el vector  $\mathbf{v}$ :

$$P_1 = (1+2, 2+2, 3+2) = (3,4,5)$$

La siguiente definición representa el vector utilizando el componente  $vector$  display de Grasshopper, y señala el final del vector representado que, como era de esperar, coincide con el punto P1:

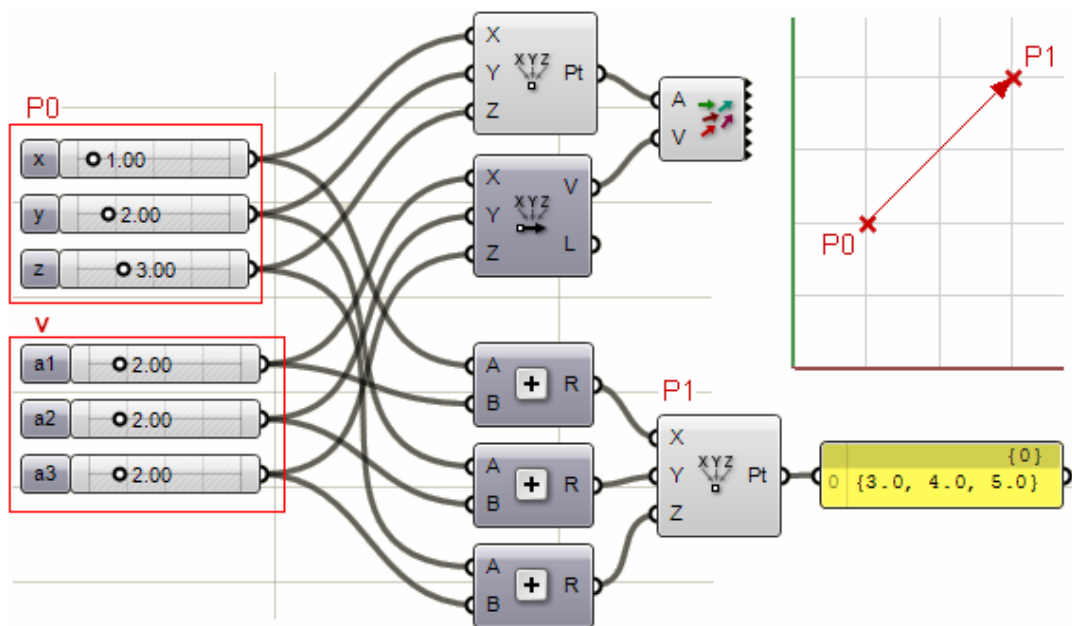


Figura (2): Relación entre un vector, su punto base y el punto que coincide con la ubicación de la flecha final del vector

### Vector de posición

Existe una representación especial de vectores que utiliza el origen de coordenadas  $P_0 (0,0,0)$  como punto origen de representación del vector. El vector de posición  $\mathbf{v} = \langle a_1, a_2, a_3 \rangle$  queda representado como una línea entre dos puntos  $P_0$  y  $P_1$ , tal que:

$$P_0 = (0,0,0)$$

$$P_1 = (a_1, a_2, a_3)$$

El vector de posición de un vector dado  $v = \langle a_1, a_2, a_3 \rangle$  es una representación especial de dicho vector desde el punto de origen  $(0,0,0)$  al punto  $(a_1, a_2, a_3)$ .

Es muy importante no confundir vectores con puntos cuyos componentes sean equivalentes. Son dos conceptos muy diferentes. En la siguiente definición de Grasshopper, las coordenadas del punto P1 son iguales a los componentes del vector.

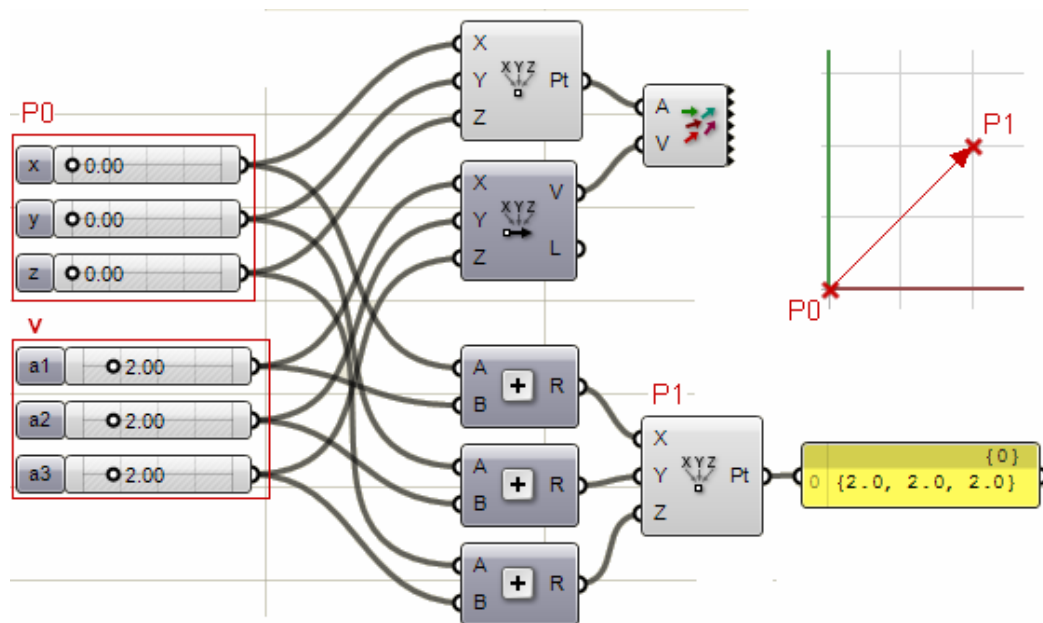


Figura (3): Vector de posición

## Operaciones vectoriales

### Suma de vectores

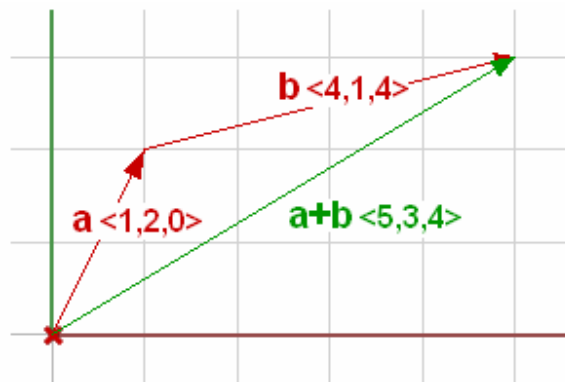
Sumamos dos vectores sumando sus componentes correspondientes. Es decir, si tenemos dos vectores  $\mathbf{a}$  y  $\mathbf{b}$ , la suma  $\mathbf{a}+\mathbf{b}$  es un vector que se calcula de la siguiente manera:

$$\mathbf{a} = \langle a_1, a_2, a_3 \rangle$$

$$\mathbf{b} = \langle b_1, b_2, b_3 \rangle$$

$$\mathbf{a}+\mathbf{b} = \langle a_1+b_1, a_2+b_2, a_3+b_3 \rangle$$

Por ejemplo, si tenemos  $\mathbf{a} \langle 1, 2, 0 \rangle$  y  $\mathbf{b} \langle 4, 1, 4 \rangle$  la suma  $\mathbf{a}+\mathbf{b} \langle 5, 3, 4 \rangle$  se muestra en la siguiente figura:



La siguiente definición de Grasshopper muestra cómo crear el vector  $\mathbf{a+b}$  sumando los componentes correspondientes a los dos vectores  $\mathbf{a}$  y  $\mathbf{b}$ .

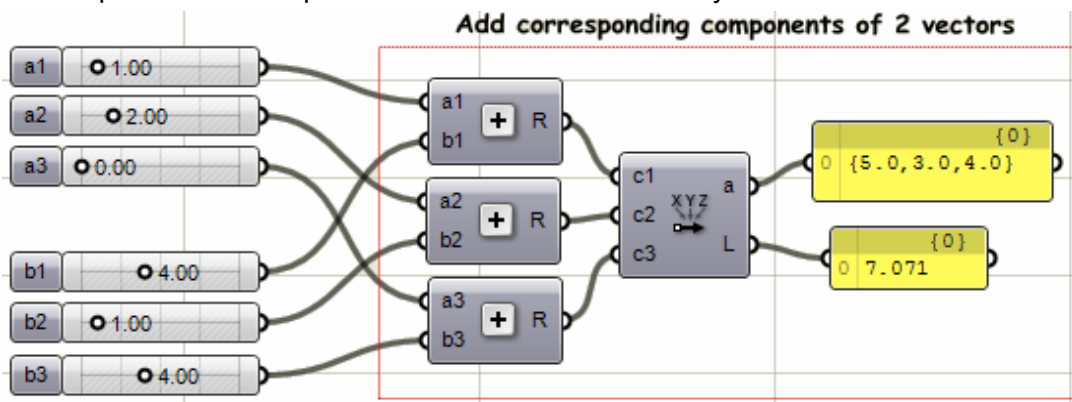


Figura (4): Suma de vectores mediante la suma de sus componentes correspondientes

El vector resultante es el mismo que el obtenido utilizando el componente propio de Grasshopper *addition* (suma):

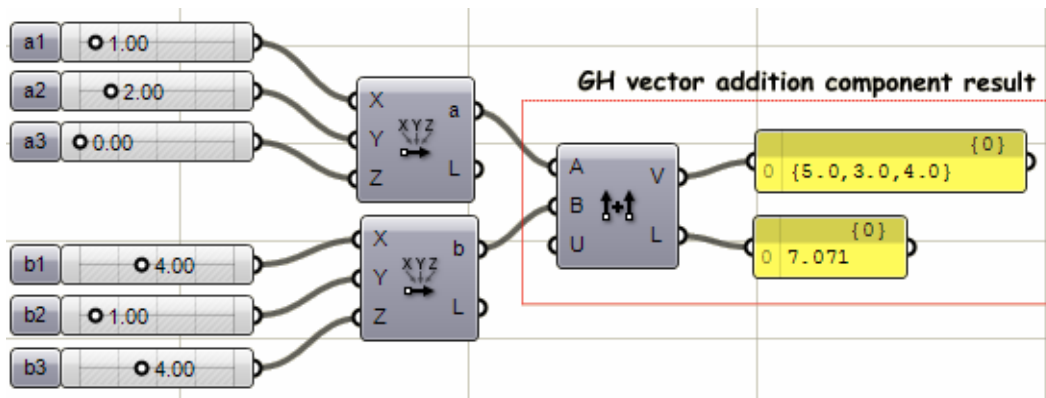


Figura (5): Suma de vectores usando el componente *vector addition* (suma vectorial) de GH



Suma dos vectores sumando sus correspondientes componentes.

La suma de vectores también es muy útil para hallar la dirección media de múltiples vectores. Para estos casos, solemos utilizar vectores de la misma longitud. Este es un ejemplo que muestra la diferencia entre emplear vectores de iguales o de diferentes longitudes en la resultante de la suma de dichos vectores:

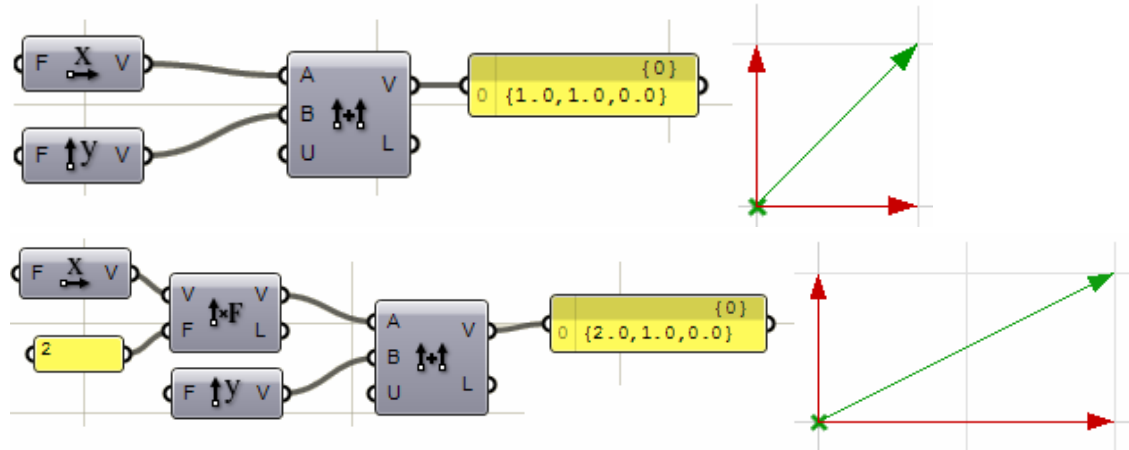


Figura (6): Suma de vectores para encontrar la dirección media

Es probable que los vectores de entrada no tengan la misma longitud. Con el fin de encontrar la dirección media, tiene que utilizar el "vector unitario" de los vectores de entrada. Como veremos más adelante, un vector unitario es un vector que tiene un módulo igual a uno. He aquí un ejemplo que resuelve la suma de vectores de diferentes longitudes para hallar la dirección media en Grasshopper.

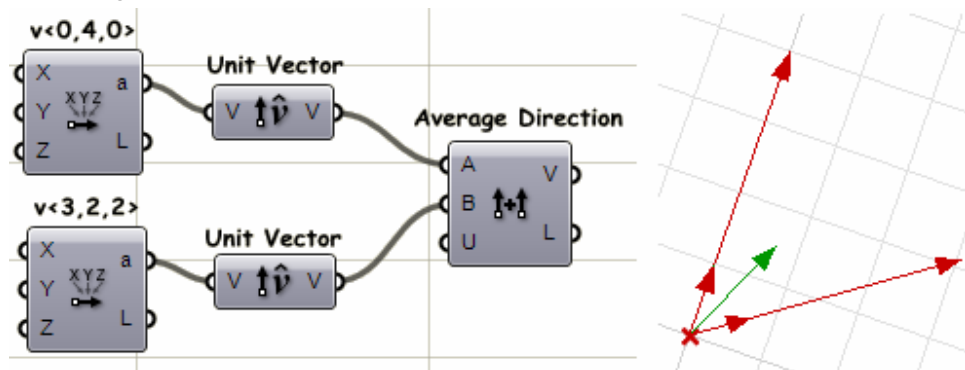


Figura (7): Vectores unitarios para encontrar la dirección media de dos o más vectores

### Longitud del vector

Usaremos  $|a|$  para representar la longitud de un vector dado "a". El **módulo** o **longitud** de un vector  $a = <a_1, a_2, a_3>$  se calcula:

$$|a| = \text{raíz}(a_1^2 + a_2^2 + a_3^2)$$

Este es un ejemplo de cómo calcular el módulo de un vector utilizando el componente *function* (función) de Grasshopper:

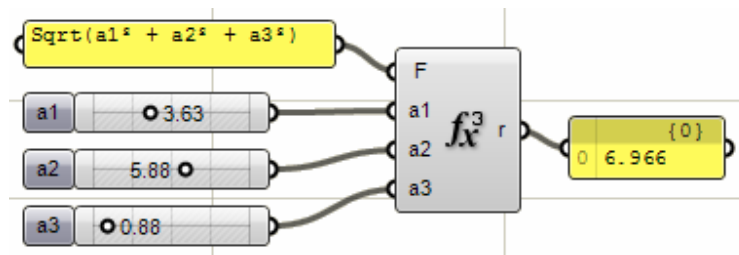


Figura (8): Cálculo del módulo de un vector

Nótese que el componente *vector* de Grasshopper tiene una salida “L” que representa el módulo del vector. Utilizando el mismo vector que en el ejemplo anterior, se dará cuenta de que la longitud es la misma.

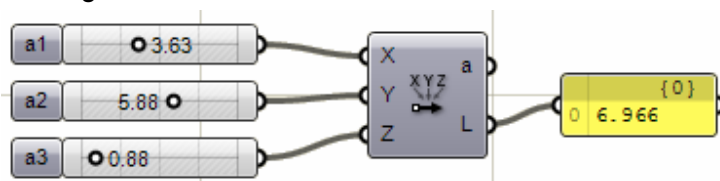


Figura (9): Longitud del vector como parámetro del componente *vector* de GH.

### Operaciones escalares en vectores

Dado el vector  $\mathbf{a} = \langle a_1, a_2, a_3 \rangle$ , y el factor  $t =$  a un número real cualquiera,  $\mathbf{a} * t = \langle a_1 * t, a_2 * t, a_3 * t \rangle$

Aquí se muestra la ecuación implementada en Grasshopper:

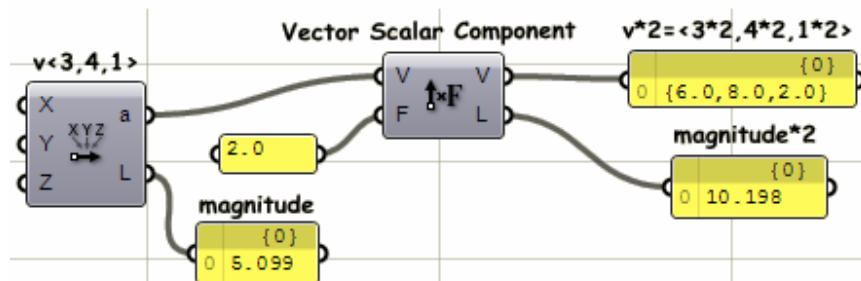


Figura (10): Operación escalar de vectores

### Vector unitario

Un vector unitario es un vector cuyo módulo es igual a la unidad. Los vectores unitarios se emplean normalmente para comparar direcciones entre vectores.

*Un vector se denomina vector unitario cuando su módulo o longitud es igual a la unidad.*

### Propiedades de los vectores

Los vectores tienen ocho propiedades fundamentales. Siendo  $\mathbf{a}$ ,  $\mathbf{b}$  y  $\mathbf{c}$  vectores, y  $s$  y  $t$  escalares, se tiene que:

1.  $a + b = b + a$
2.  $a + 0 = a$
3.  $s(a+b) = sa + sb$
4.  $st(a) = s(ta)$
5.  $a+(b + c) = (a+b) + c$
6.  $a + (-a) = 0$
7.  $(s + t)a = sa + ta$
8.  $1 * a = a$

### Producto escalar de vectores

El producto escalar de dos vectores se define de la siguiente manera:

Dado:

vector  $a = \langle a_1, a_2, a_3 \rangle$

vector  $b = \langle b_1, b_2, b_3 \rangle$

$a \cdot b = a_1 \cdot b_1 + a_2 \cdot b_2 + a_3 \cdot b_3$

En la siguiente ilustración, mostramos cómo el componente *dot product* (producto escalar) devuelve el mismo resultado que la ecuación  $a \cdot b$ :

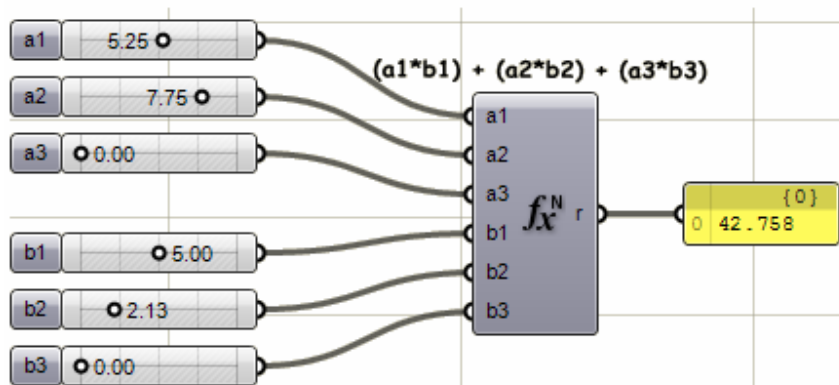


Figura (11): El producto escalar de dos vectores como la suma de la multiplicación de los componentes correspondientes

Grasshopper tiene un componente propio *dot product* como se muestra en la siguiente ilustración:

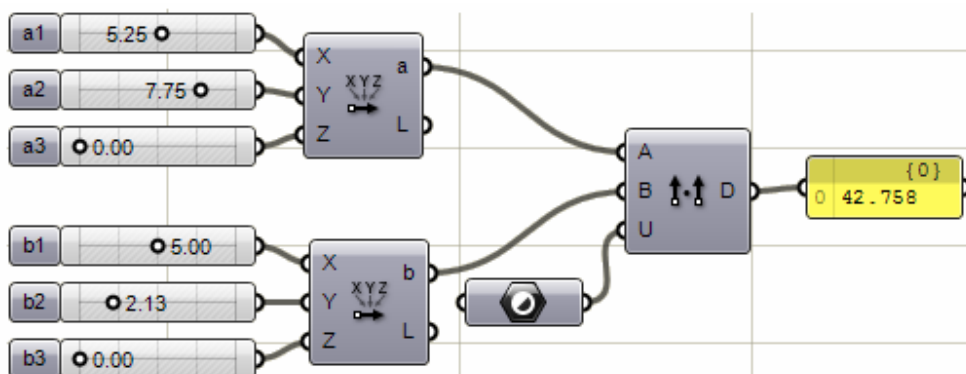


Figura (12): Producto escalar de dos vectores empleando el componente *dot product* de GH.

Al calcular el producto escalar de dos vectores unitarios, el resultado se encuentra siempre entre los valores -1 y 1.

El producto escalar de un vector consigo mismo es el cuadrado de su módulo:

$$\mathbf{a} \cdot \mathbf{a} = |\mathbf{a}|^2$$

**Prueba:**

Sea el vector  $\mathbf{a} = \langle a_1, a_2, a_3 \rangle$ , entonces a partir de la definición de producto escalar de dos vectores:

$$\mathbf{a} \cdot \mathbf{a} = a_1 \cdot a_1 + a_2 \cdot a_2 + a_3 \cdot a_3$$

ó

$$\mathbf{a} \cdot \mathbf{a} = a_1^2 + a_2^2 + a_3^2$$

Como sabemos que:

$$|\mathbf{a}| = \sqrt{a_1^2 + a_2^2 + a_3^2}$$

Por lo tanto,

$$\mathbf{a} \cdot \mathbf{a} = |\mathbf{a}|^2$$

He aquí un ejemplo en Grasshopper para demostrar esta propiedad, comparando los resultados usando el componente *dot product* con la multiplicación de la longitud del vector por sí mismo:

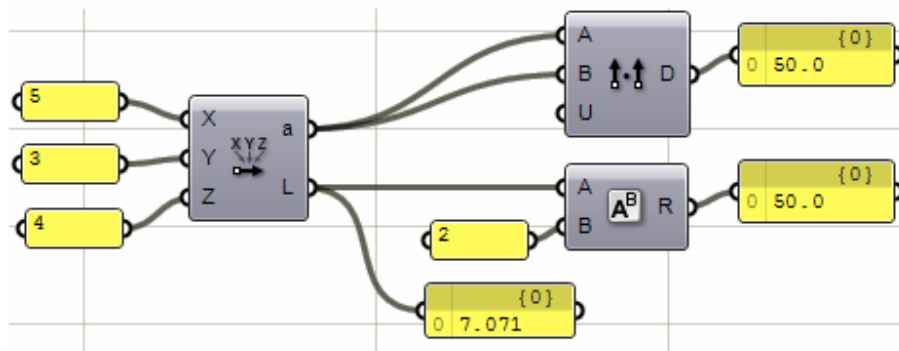


Figura (13): El producto escalar de un vector consigo mismo

**Producto escalar y ángulo entre dos vectores**

Una propiedad importante del producto escalar de vectores es:

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos(\theta), \text{ ó}$$

$$\cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{(|\mathbf{a}| |\mathbf{b}|)},$$

donde "θ" es el ángulo comprendido entre los vectores de posición de  $\mathbf{a}$  y  $\mathbf{b}$ .

Y si los vectores  $\mathbf{a}$  y  $\mathbf{b}$  son unitarios, entonces podemos simplificarlo de la siguiente manera:

$$\cos(\theta) = \mathbf{a} \cdot \mathbf{b}$$

*El producto escalar de dos vectores unitarios es igual al coseno del ángulo que forman.*

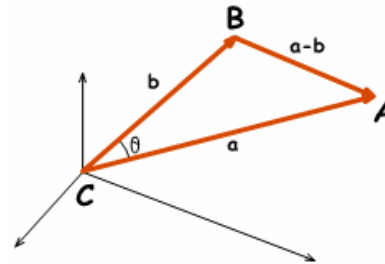
**Prueba:**

A partir de las propiedades de los cosenos, para el triángulo ABC:

$$|AB|^2 = |CA|^2 + |CB|^2 - 2|CA||CB|\cos(\theta)$$

ó:

$$|\mathbf{a}-\mathbf{b}|^2 = |\mathbf{a}|^2 + |\mathbf{b}|^2 - 2|\mathbf{a}||\mathbf{b}|\cos(\theta) \quad \text{--- (1)}$$



$|AB|^2$  es igual que  $|\mathbf{a}-\mathbf{b}|^2$ , por lo tanto, podemos decir que:

$$\begin{aligned} |\mathbf{a}-\mathbf{b}|^2 &= \{\mathbf{a}-\mathbf{b}\} \cdot \{\mathbf{a}-\mathbf{b}\} \\ &= \mathbf{a} \cdot \mathbf{a} - \mathbf{a} \cdot \mathbf{b} - \mathbf{b} \cdot \mathbf{a} + \mathbf{b} \cdot \mathbf{b} \\ &= |\mathbf{a}|^2 - 2\mathbf{a} \cdot \mathbf{b} + |\mathbf{b}|^2 \quad \text{--- (2)} \end{aligned}$$

de (1) & (2)

$$|\mathbf{a}|^2 - 2\mathbf{a} \cdot \mathbf{b} + |\mathbf{b}|^2 = |\mathbf{a}|^2 + |\mathbf{b}|^2 - 2|\mathbf{a}||\mathbf{b}|\cos(\theta)$$

por tanto:

$$2\mathbf{a} \cdot \mathbf{b} = 2|\mathbf{a}||\mathbf{b}|\cos(\theta)$$

ó:

$$\cos(\theta) = \mathbf{a} \cdot \mathbf{b} / (|\mathbf{a}||\mathbf{b}|)$$

*Los vectores  $\mathbf{a}$  y  $\mathbf{b}$  son ortogonales si, y sólo si,  $\mathbf{a} \cdot \mathbf{b} = 0$*

*Pero, ¿cuál es el producto escalar de dos vectores unitarios si son paralelos?*

Lo más práctico es pensar que el producto escalar de dos vectores es el módulo de la proyección de uno sobre otro.

He aquí una demostración de este concepto utilizando Grasshopper. En la primera figura, se calcula el producto escalar del vector unitario eje-x con un vector de entrada "v". En la segunda figura, proyectamos el punto final del vector de posición "v" sobre una línea a lo largo del eje-x y calculamos la distancia desde el origen hasta ese punto de proyección. Se dará cuenta de que el producto escalar y la longitud de proyección son iguales.

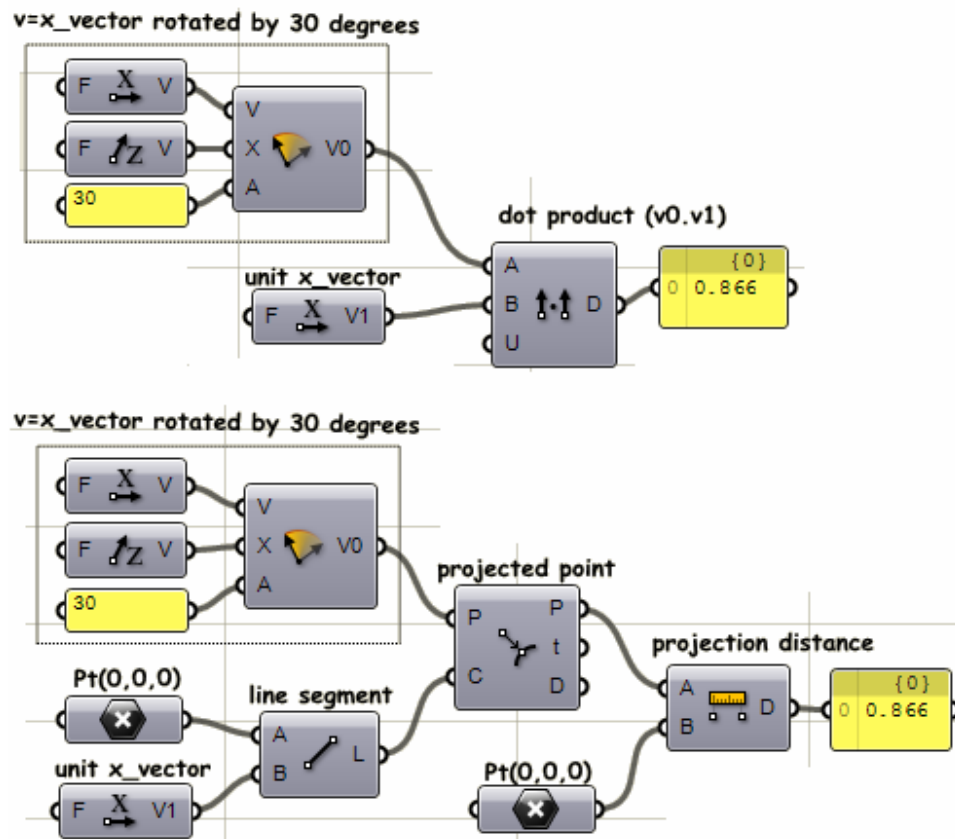


Figura (14): El producto escalar y el ángulo entre vectores

### Propiedades del producto escalar

Si **a**, **b** y **c** son vectores, y **s** es un escalar, entonces:

1.  $\mathbf{a} \cdot \mathbf{a} = |\mathbf{a}|^2$
2.  $\mathbf{a} \cdot (\mathbf{b} + \mathbf{c}) = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \cdot \mathbf{c}$
3.  $0 \cdot \mathbf{a} = 0$
4.  $\mathbf{a} \cdot \mathbf{b} = \mathbf{b} \cdot \mathbf{a}$
5.  $(s\mathbf{a}) \cdot \mathbf{b} = s(\mathbf{a} \cdot \mathbf{b}) = \mathbf{a} \cdot (s\mathbf{b})$

### Producto vectorial

Del producto vectorial de dos vectores tridimensionales se obtiene un tercer vector tridimensional que es ortogonal a ambos. Dado:

$$\mathbf{a} = \langle a_1, a_2, a_3 \rangle$$

$$\mathbf{b} = \langle b_1, b_2, b_3 \rangle$$

El producto vectorial  $\mathbf{a} \times \mathbf{b}$  se resuelve utilizando determinantes. Aquí está un ejemplo rápido de cómo calcular un determinante. Usamos los vectores base estándar  $\mathbf{i} = \langle 1, 0, 0 \rangle$ ,  $\mathbf{j} = \langle 0, 1, 0 \rangle$  y  $\mathbf{k} = \langle 0, 0, 1 \rangle$

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = \mathbf{i}(a_2 b_3 - a_3 b_2) + \mathbf{j}(a_3 b_1 - a_1 b_3) + \mathbf{k}(a_1 b_2 - a_2 b_1)$$

$$\mathbf{a} \times \mathbf{b} = \mathbf{i}(a_2 b_3 - a_3 b_2) + \mathbf{j}(a_3 b_1 - a_1 b_3) + \mathbf{k}(a_1 b_2 - a_2 b_1)$$

$$\mathbf{a} \times \mathbf{b} = \langle a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1 \rangle$$

Y ésta es una definición de Grasshopper para la resolución de productos vectoriales utilizando estas expresiones y comparándolas con el componente *cross product* (producto vectorial). Ambos procedimientos tienen el mismo resultado.

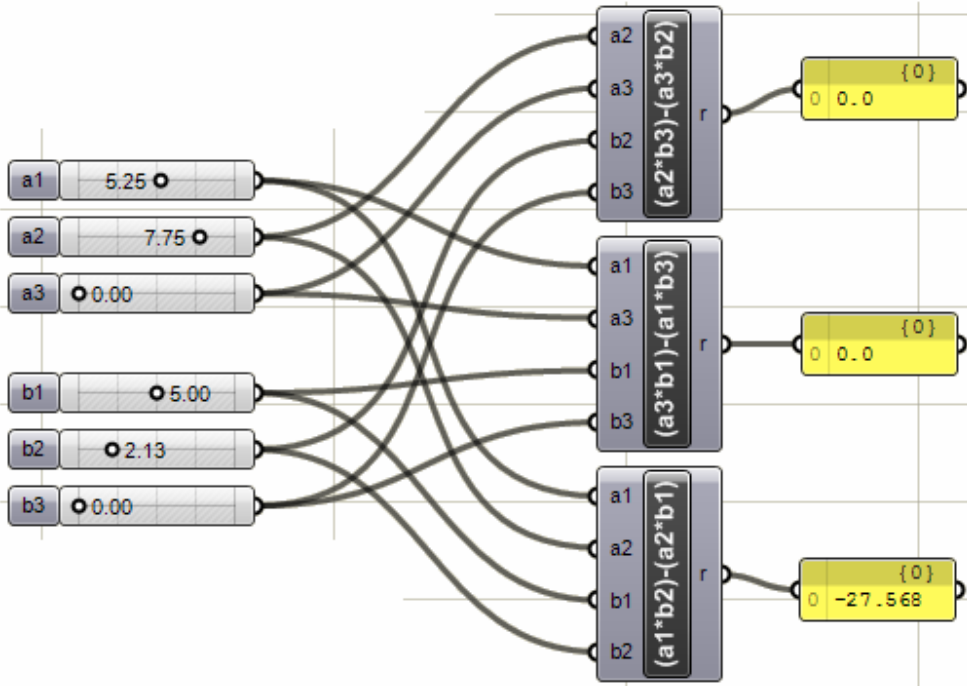


Figura (15): Cálculo del producto vectorial de dos vectores

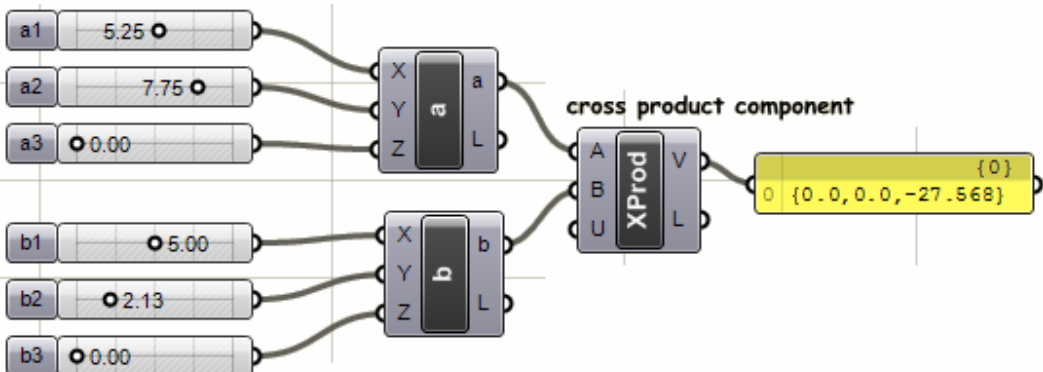


Figura (16): Cálculo del producto vectorial de dos vectores utilizando el componente *cross product* de GH

**El vector  $\mathbf{a} \times \mathbf{b}$  es ortogonal a ambos  $\mathbf{a}$  y  $\mathbf{b}$**

**Teorema**

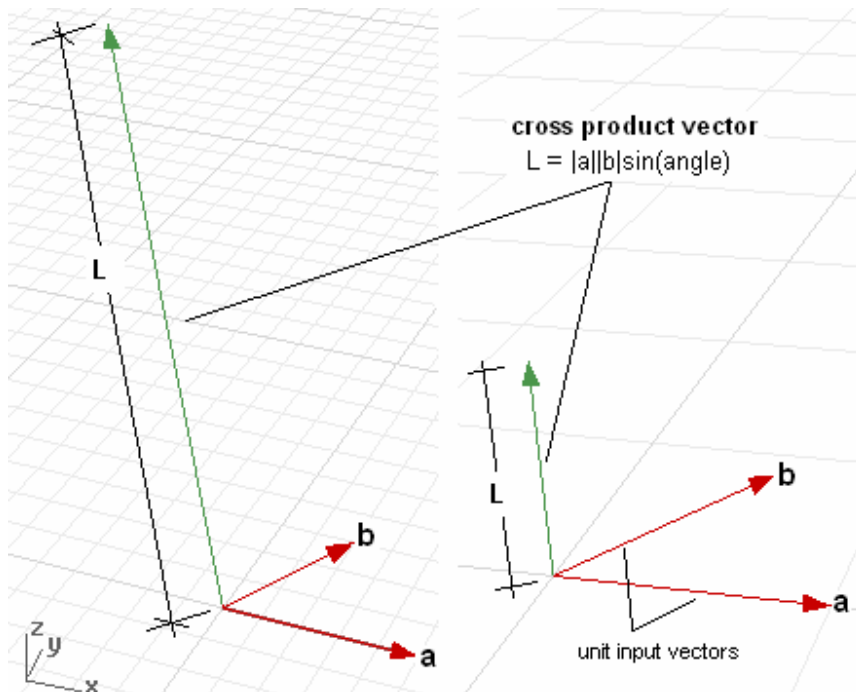
Para cualquier par de vectores tridimensionales  $\mathbf{a}$  y  $\mathbf{b}$

$$|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}||\mathbf{b}|\sin(\theta)$$

Donde " $\theta$ " es el ángulo comprendido entre los vectores de posición de  $\mathbf{a}$  y  $\mathbf{b}$ .

O si  $\mathbf{a}$  y  $\mathbf{b}$  son vectores unitarios, la longitud de su producto vectorial es igual al seno del ángulo entre ellos. En otras palabras:

$$|\mathbf{a} \times \mathbf{b}| = \sin(\theta)$$



Este es un ejemplo para calcular la longitud del producto vectorial de dos vectores utilizando el componente *cross product* de GH y compararlo con el cálculo mediante la expresión anterior. Como es de esperar, ambos resultados son iguales.



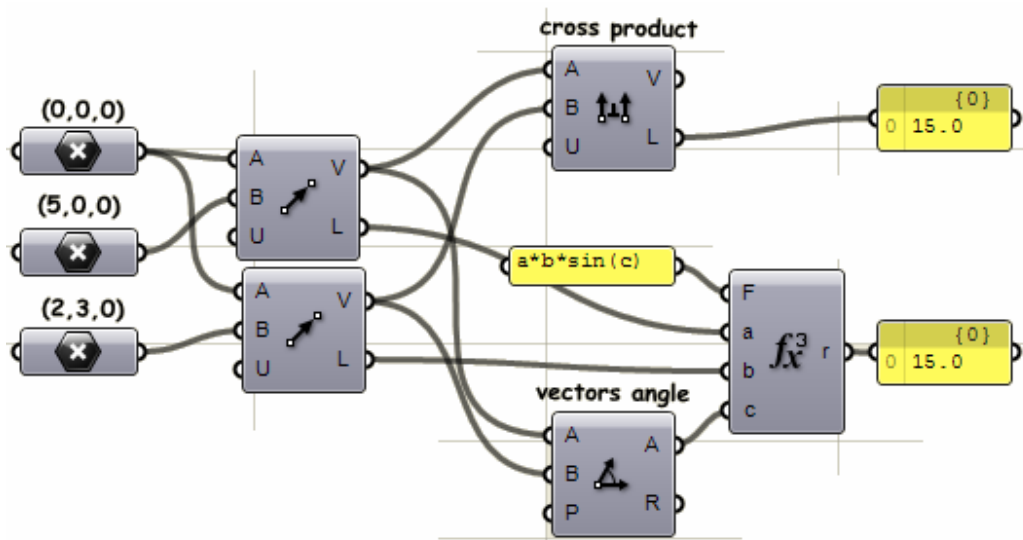


Figura (17): Cálculo de la longitud del vector producto vectorial mediante una función y mediante el componente cross product de GH

Al determinar el producto vectorial, el orden de los operandos es importante. Por ejemplo:

$$\begin{aligned} \mathbf{a} &= \langle 1, 0, 0 \rangle \\ \mathbf{b} &= \langle 0, 1, 0 \rangle \\ \mathbf{a} \times \mathbf{b} &= \langle 0, 0, 1 \rangle \\ \mathbf{b} \times \mathbf{a} &= \langle 0, 0, -1 \rangle \end{aligned}$$

Rhino sigue la regla de la mano derecha, por lo que  $\mathbf{a} \times \mathbf{b}$  estaría dado por la regla de la mano derecha (donde  $\mathbf{a}$  = dedo índice,  $\mathbf{b}$  = dedo corazón y el **resultado** = pulgar).

Los vectores  $\mathbf{a}$  y  $\mathbf{b}$  son paralelos si, y solo si,  $\mathbf{a} \times \mathbf{b} = 0$

### Propiedades del producto vectorial

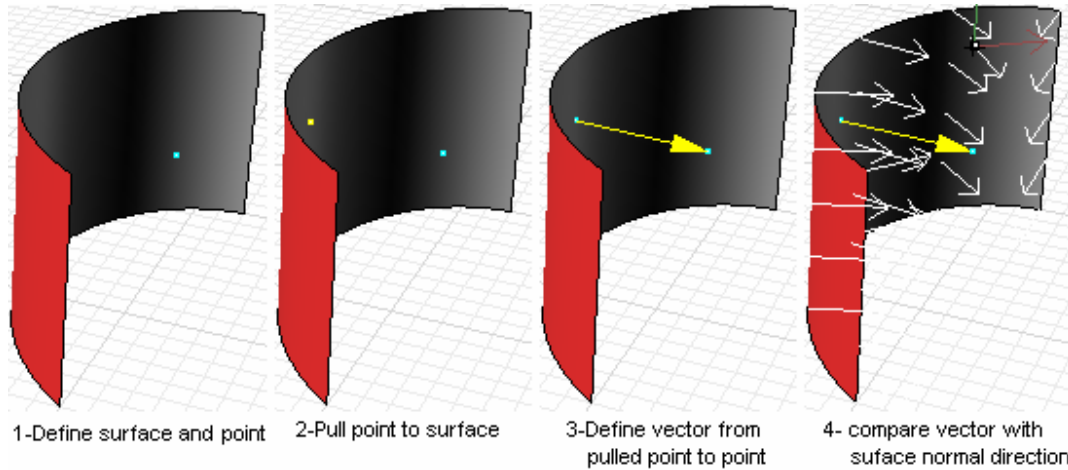
Si  $\mathbf{a}$ ,  $\mathbf{b}$  y  $\mathbf{c}$  son vectores e  $s$  es escalar, entonces:

1.  $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$
2.  $(s\mathbf{a}) \times \mathbf{b} = s(\mathbf{a} \times \mathbf{b}) = \mathbf{a} \times (s\mathbf{b})$
3.  $\mathbf{a} \times (\mathbf{b} + \mathbf{c}) = \mathbf{a} \times \mathbf{b} + \mathbf{a} \times \mathbf{c}$
4.  $(\mathbf{a} + \mathbf{b}) \times \mathbf{c} = \mathbf{a} \times \mathbf{c} + \mathbf{b} \times \mathbf{c}$
5.  $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}$
6.  $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\mathbf{c}$

### Ejemplo

Todos los conceptos revisados hasta ahora tienen aplicación directa en la resolución de problemas de geometría que se plantean en el modelado. Por ejemplo, dado un punto y una superficie, ¿cómo podemos determinar si el punto está al frente o tras esa superficie?

Éstos son los pasos a seguir para resolver el problema:



Y aquí se muestra una solución en Grasshopper siguiendo los mismos pasos. Nótese que, en este caso, que el producto escalar sea mayor que 0 implica que el punto está al frente de la superficie. Si el producto escalar fuera menor que 0, entonces el punto estaría en la parte posterior.

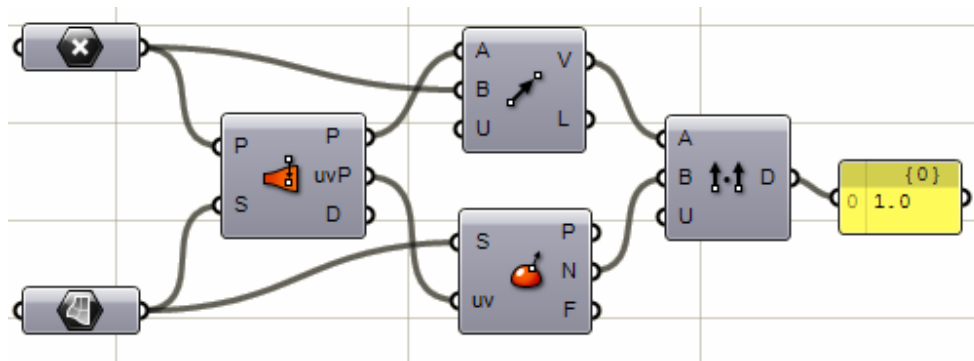
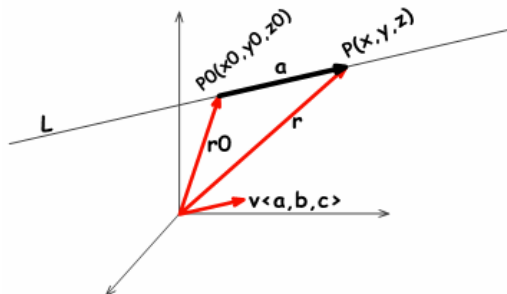


Figura (18): Búsqueda de la localización de un punto relativa a las caras delantera y trasera de la superficie

## Ecuación vectorial de la recta

La ecuación vectorial de la recta se utiliza en aplicaciones de modelado 3D y gráficos por ordenador. A continuación se expone una descripción de dicha ecuación y sus aplicaciones.



En la figura:

- L = línea
- v = dirección de la línea
- P0 = posición de la línea
- $r = r_0 + a$  --- (1)
- $a = t * v$  --- (2)

Por lo tanto, a partir de 1 y 2:

$$r = r_0 + t*v \text{ --- (3)}$$

También podemos expresar (3) así:

$$\langle x,y,z \rangle = \langle x_0,y_0,z_0 \rangle + \langle ta, tb, tc \rangle$$

$$\langle x,y,z \rangle = \langle x_0+ta, y_0+tb, z_0+tc \rangle$$

Por tanto:

$$x = x_0 + ta$$

$$y = y_0 + tb$$

$$z = z_0 + tc$$

Que es lo mismo que:

$$P = P_0 + tv$$

Esta sería una definición de Grasshopper para obtener cualquier punto sobre una recta:

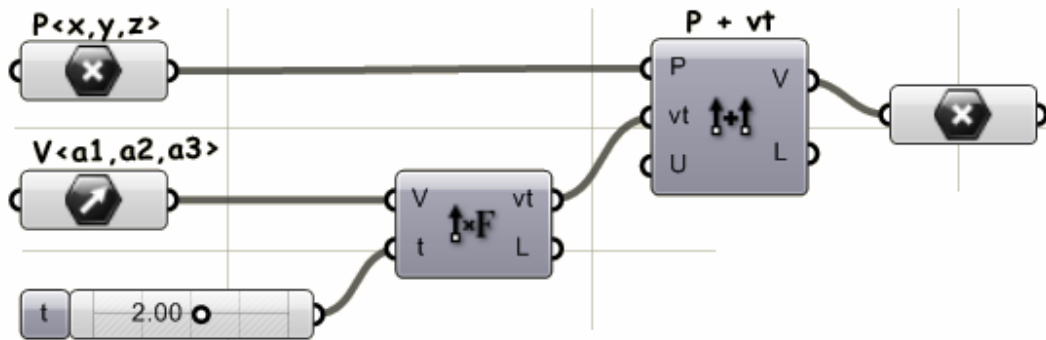
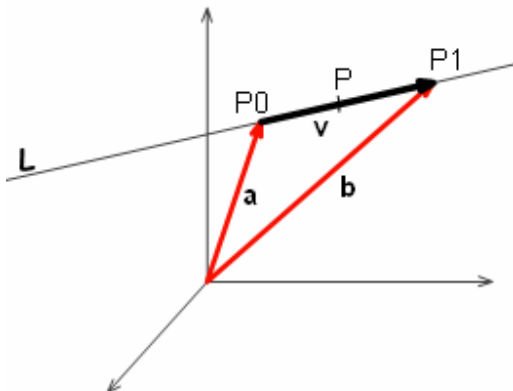


Figura (19): Búsqueda de puntos sobre una línea

### Ejemplo

En la figura siguiente, dados los puntos P0 y P1, encuentre el punto medio P.



Tenga en cuenta que:

- a** es el vector de posición para el punto P0
- b** es el vector de posición para el punto P1
- v** es el vector que une P0 y P1

De las propiedades de la suma de vectores se obtiene que:

$$\mathbf{a} + \mathbf{v} = \mathbf{b}, \text{ ó}$$

$$\mathbf{v} = \mathbf{b} - \mathbf{a}$$

En cualquier caso, la ecuación de la recta es:  $P = P0 + t \cdot V$ , y dado que  $t=0.5$  y  $\mathbf{v}=\mathbf{b}-\mathbf{a}$  (según lo anterior), podemos decir que:

$$P = P0 + 0.5(\mathbf{b}-\mathbf{a})$$

Utilice la ecuación anterior para crear una definición de Grasshopper, tal que:

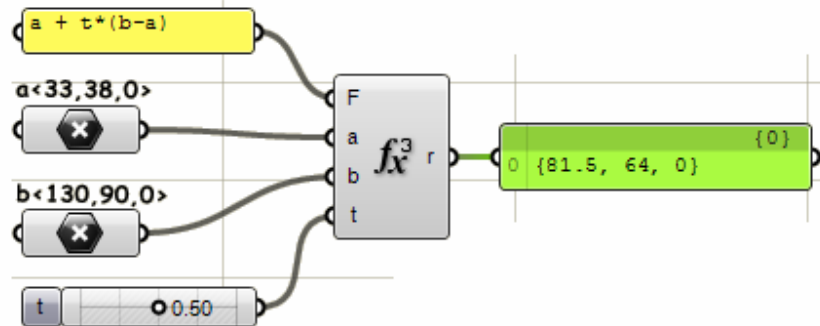


Figura (20): Búsqueda del punto medio entre dos puntos dados

En general, usted puede encontrar cualquier punto entre P0 y P1, cambiando el valor de  $t$  entre 0 y 1.

## Ecuación vectorial del plano

En la imagen adjunta:

- $P0(x0,y0,z0)$  = un punto dado sobre el plano
- $\mathbf{r0}\langle x0,y0,z0 \rangle$  = el vector de posición de P0
- $\mathbf{n}\langle a,b,c \rangle$  = vector normal al plano
- $P(x,y,z)$  = punto cualquiera del plano
- $\mathbf{r}\langle x,y,z \rangle$  = vector de posición de P

Ya que el producto escalar de dos vectores ortogonales es cero, entonces:

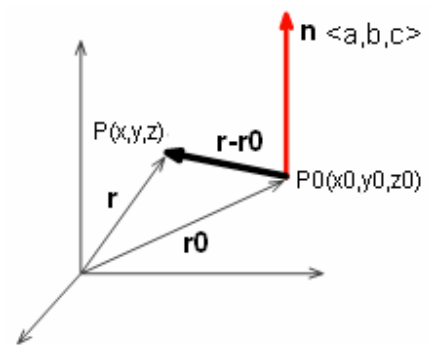
$$\mathbf{n} \cdot (\mathbf{r} - \mathbf{r0}) = 0$$

O podemos expresarlo de la siguiente manera:

$$\langle a,b,c \rangle \cdot \langle x-x0, y-y0, z-z0 \rangle = 0$$

Resolviendo el producto escalar obtenemos la ecuación escalar del plano:

$$a(x-x0) + b(y-y0) + c(z-z0) = 0$$



### Ejemplo

¿Cómo podemos determinar el plano que pasa por tres puntos dados, utilizando un punto origen y un vector normal a dicho plano?

Para hallar un plano, necesitamos un punto origen y un vector normal al mismo. Tenemos un punto origen, que podría ser cualquiera de los tres puntos, por tanto ¿cómo determinamos dicha normal?

Sabemos que el producto vectorial de dos vectores es un tercer vector perpendicular a ambos. Éste podría ser el vector normal. Por tanto, así es como podríamos resolver este problema con Grasshopper:

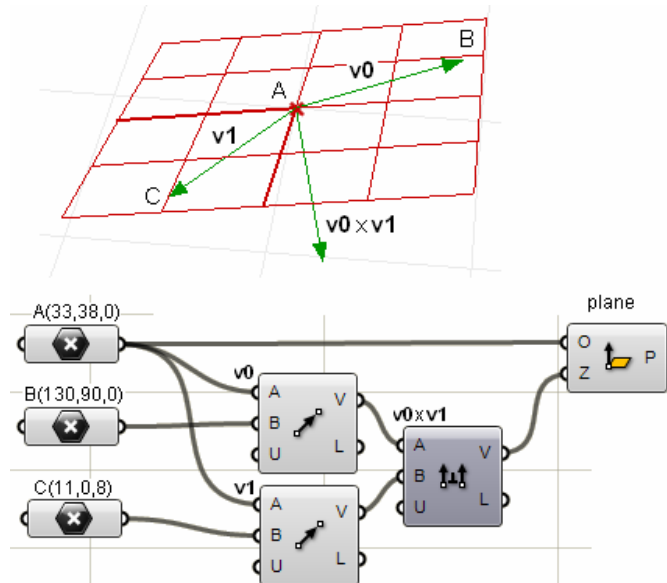


Figura (21): Búsqueda del plano que pasa por tres puntos

## 2 Matrices y transformaciones

### Introducción

Aunque los diseñadores no tengan que utilizar las matrices matemáticas directamente en el diseño computacional, el conocimiento de sus fundamentos es muy útil para apreciar lo que está sucediendo en segundo plano. Las matrices de transformación son las responsables de mover, rotar, proyectar y escalar objetos. Las matrices se utilizan también para las transformaciones entre sistemas de coordenadas, por ejemplo, de las coordenadas universales en 3D al sistema de coordenadas de la pantalla en 2D.

Podemos definir la transformación como una función que toma un punto (o un vector) y referencia ese punto a otro punto (o vector). ¿Qué es una matriz y por qué la necesitamos para las transformaciones?

Una matriz es una serie rectangular de números. Las dimensiones de una matriz son m-por-n donde:

**m**: número de filas

**n**: número de columnas

Así que si tenemos una matriz **M** con dos filas y tres columnas, expresamos la dimensión de la matriz de la siguiente manera:

$$\text{dim}(\mathbf{M}) = [2,3]$$

Las matrices han demostrado ser muy útiles para la representación de las transformaciones. Con esta representación se pueden realizar múltiples transformaciones muy rápidamente. La clave es encontrar un formato que pueda representar a TODAS las transformaciones tales como traslación (mover), rotación y escala.

### Multiplicación de matrices

La multiplicación de matrices se utiliza para aplicar transformaciones a la geometría. Una serie de matrices de transformación se multiplicarán primero para obtener una matriz de transformación final que, a su vez, es utilizada para la transformación de la geometría. La multiplicación de matrices es una de las operaciones de uso más frecuente de las matrices, por lo que es conveniente explicarlas.

Para multiplicar dos matrices, éstas tienen que coincidir. En otras palabras, el número de columnas de la primera matriz debe ser igual al número de filas de la segunda matriz. El tamaño de la matriz resultante es igual al número de filas de la primera matriz y el número de columnas de la segunda matriz. Por ejemplo, si tenemos dos matrices, **M1** y **M2**, con una dimensión igual a [2x4] y [4x5] respectivamente, entonces la matriz multiplicada resultante de **M1.M2** tendría una dimensión igual a [2x5] como se muestra:

$$\begin{bmatrix} + & + & + & + \\ 1 & 2 & 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} + & + & a & + & + \\ + & + & b & + & + \\ + & + & c & + & + \\ + & + & d & + & + \end{bmatrix} = \begin{bmatrix} + & + & + & + & + \\ + & + & R_{2,3} & + & + \end{bmatrix}$$

$\dim(M1) = [2 \times 4]$     $\dim(M2) = [4 \times 5]$     $\dim(M1.M2) = [2 \times 5]$

$R_{2,3} = 1 \cdot a + 2 \cdot b + 3 \cdot c + 4 \cdot d$

Éstos son los pasos generales para multiplicar dos matrices:

1. Asegurarse de que coincidan.  
Esto es, dadas dos matrices de tamaño  $\dim(M_1)=[a \times b]$ ,  $\dim(M_2)=[c \times d]$ , **b** debe ser igual a **c**.
2. Hallar la suma de la multiplicación de los puntos correspondientes de la primera fila de la matriz de la izquierda con la primera columna de la matriz de la derecha para obtener el elemento en el índice (1,1) de la matriz resultante.
3. Repita el paso 2 para obtener todos los elementos de la matriz resultante.  
Por ejemplo, la suma de la multiplicación de la tercera fila de la matriz de la izquierda con la segunda columna de la derecha para obtener el elemento en el índice (3,2) de la matriz resultante.

Una matriz especial es la matriz identidad. La principal propiedad de esta matriz es que si se multiplica por cualquier otra matriz, no cambia sus valores como a continuación:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \cdot 2 + 0 \cdot 3 + 0 \cdot 1 + 0 \cdot 1 \\ 0 \cdot 2 + 1 \cdot 3 + 0 \cdot 1 + 0 \cdot 1 \\ 0 \cdot 2 + 0 \cdot 3 + 1 \cdot 1 + 0 \cdot 1 \\ 0 \cdot 2 + 0 \cdot 3 + 0 \cdot 1 + 1 \cdot 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 1 \\ 1 \end{bmatrix}$$

## Transformaciones afines

En esta sección, vamos a explicar un tipo de transformación especial, pero muy común, llamada *transformación afín*. Cuando se aplican las transformaciones afines a la geometría tienen la propiedad de conservar las relaciones de paralelismo entre líneas, pero no la longitud o los ángulos. La traslación (mover), la rotación, la escala y el sesgado son transformaciones afines.

### Traslación (mover)

Mover un punto desde una posición de partida a lo largo de un vector determinado se calcula de la siguiente manera:

$$P' = P + V$$

Supongamos que:

$P(x,y,z)$  es un punto dado

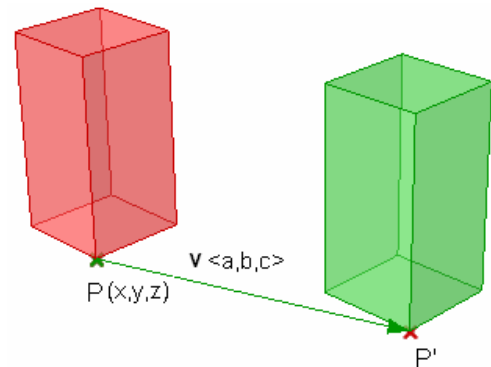
$v\langle a,b,c\rangle$  es un vector de traslación dado

luego:

$$P'(x) = x + a$$

$$P'(y) = y + b$$

$$P'(z) = z + c$$



Representamos un punto tridimensional como una columna de matriz 4x1 con un 1 en la última fila. Este "truco" nos permite representar la traslación, y de hecho, cualquier transformación afín, con la multiplicación de matrices.

El formato general para una matriz de traslación es:

$$\begin{bmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & a_2 \\ 0 & 0 & 1 & a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Por ejemplo, para mover el punto  $P(2,3,1)$  por el vector  $v(2,2,2)$ , la ubicación del nuevo punto es:

$$P' = P + V = (2+2, 3+2, 1+2) = (4, 5, 3)$$

Si empleamos las matrices y multiplicamos la matriz de traslación por el punto de entrada, entonces obtendremos la nueva localización del punto, tal que:

$$\begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1*2+0*3+0*1+2*1 \\ 0*2+1*3+0*1+2*1 \\ 0*2+0*3+1*1+2*1 \\ 0*2+0*3+0*1+1*1 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 3 \\ 1 \end{bmatrix}$$

## Rotación

Este ejemplo muestra cómo calcular la rotación alrededor del eje z y el punto base usando la trigonometría, y luego deducir el formato general de la matriz de rotación.

Tomemos un punto  $P(x, y)$  en el plano x, y y girémoslo un ángulo  $(b)$ . Del resultado, podemos deducir lo siguiente:



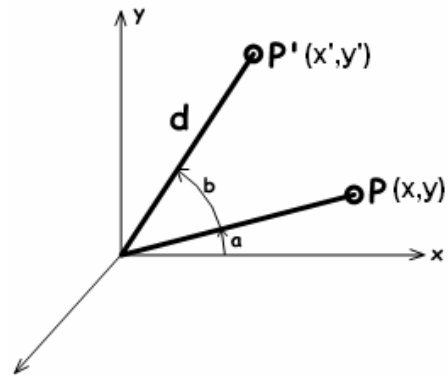
$$\begin{aligned} x &= d \cos(a) \quad \text{---(1)} \\ y &= d \sin(a) \quad \text{---(2)} \\ x' &= d \cos(b+a) \quad \text{---(3)} \\ y' &= d \sin(b+a) \quad \text{---(4)} \end{aligned}$$

Podemos expandir (3) y (4) utilizando las identidades trigonométricas del seno y el coseno de la suma de ángulos:

$$\begin{aligned} x' &= d \cos(a)\cos(b) - d \sin(a)\sin(b) \quad \text{---(5)} \\ y' &= d \cos(a)\sin(b) + d \sin(a)\cos(b) \quad \text{---(6)} \end{aligned}$$

Usando la ecuación 1 y 2:

$$\begin{aligned} x' &= x \cos(b) - y \sin(b) \\ y' &= x \sin(b) + y \cos(b) \end{aligned}$$



Usando un sistema de coordenadas homogéneo, la matriz de rotación alrededor del **eje z** sería así:

$$\begin{bmatrix} \cos(b) & -\sin(b) & 0 & 0 \\ \sin(b) & \cos(b) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matriz de rotación alrededor del **eje x** por el ángulo **b**:

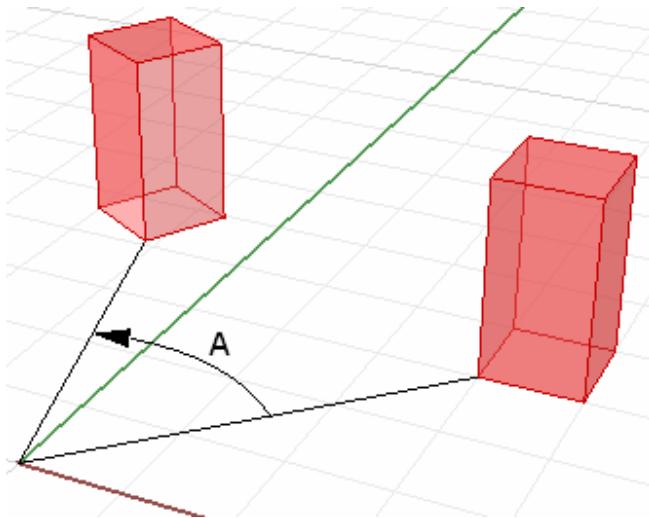
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(b) & -\sin(b) & 0 \\ 0 & \sin(b) & \cos(b) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matriz de rotación alrededor del **eje y** por el ángulo **b**:

$$\begin{bmatrix} \cos(b) & 0 & \sin(b) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(b) & 0 & \cos(b) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

OpenNURBS™, la biblioteca de geometría de Rhino (<http://www.openNURBS.org>), contiene una clase llamada OnXform que se ocupa de las transformaciones.

Almacena matrices de transformación y realiza operaciones con matrices. La siguiente imagen es un ejemplo de cómo girar un objeto y examinar los valores OnXform de la matriz para comparar con el formato general de la matriz de rotación. Puede utilizar el mismo principio para examinar otras transformaciones.



He aquí una definición de Grasshopper para la rotación de geometría y el resultado de los valores de la matriz para comparar con la forma de la matriz general:

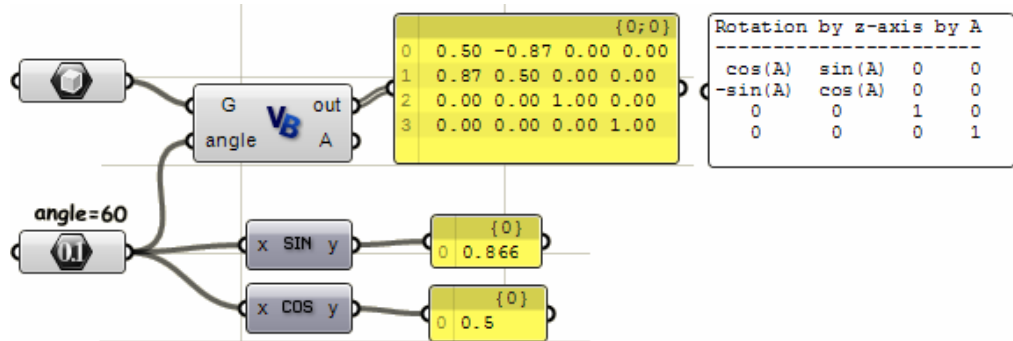


Figura (22): Rotación de geometría y matriz de transformación

### Escalado

Sabemos que:

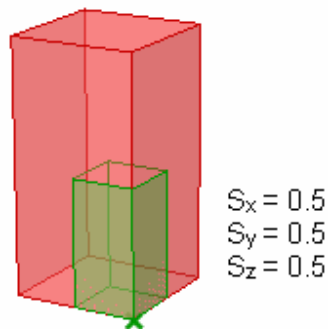
$$P' = \text{Factor escalar}(S) * P$$

ó:

$$P'.x = S_x * P.x$$

$$P'.y = S_y * P.y$$

$$P'.z = S_z * P.z$$



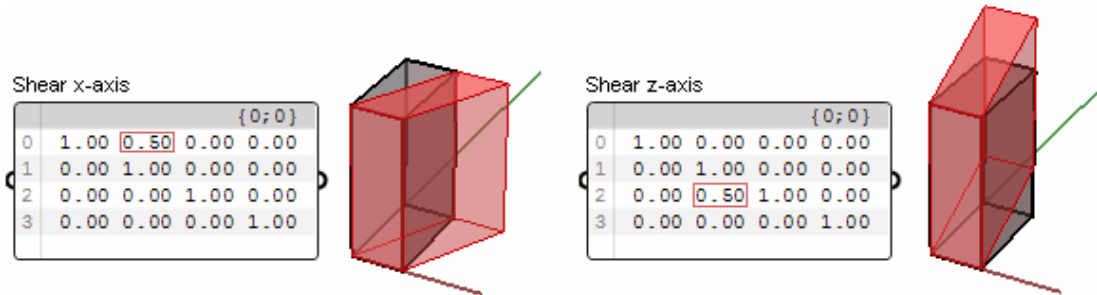
Éste es el formato de la matriz de escalado.

$$\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

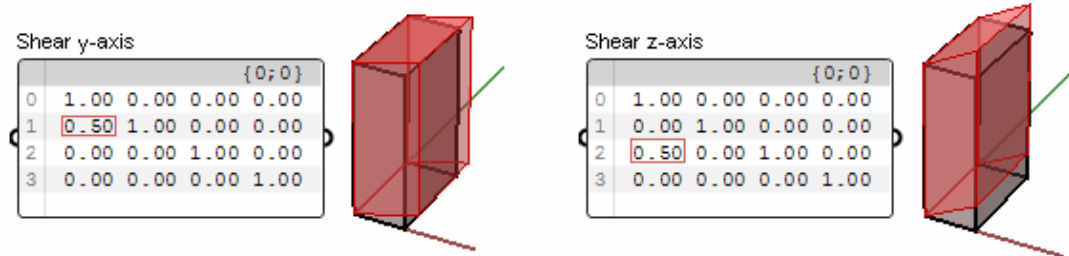
## Sesgado

El sesgado en tres dimensiones se mide a lo largo de un par de ejes en relación con el tercero. Por ejemplo, el sesgado en el eje z no va a cambiar la geometría a lo largo de ese eje, sino que altera los valores en x e y.

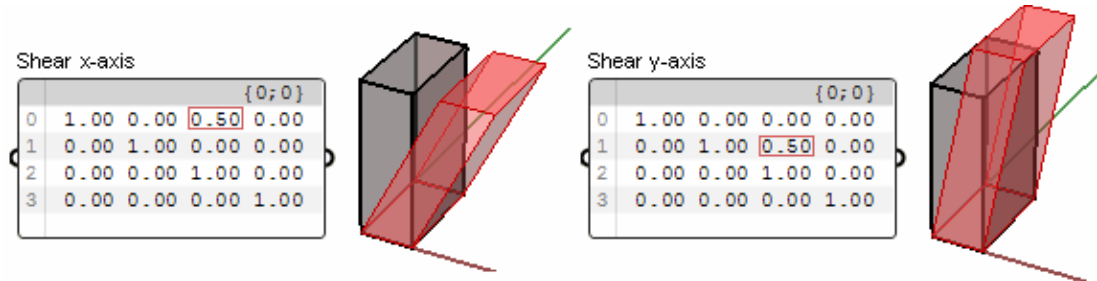
1- El sesgado en x y z, manteniendo la **coordenada y fija**:



2- El sesgado en y y z, manteniendo la **coordenada x fija**:



3- El sesgado en x e y, manteniendo la **coordenada z fija**:



He aquí una definición de GH para cambiar diferentes valores en la matriz de transformación:

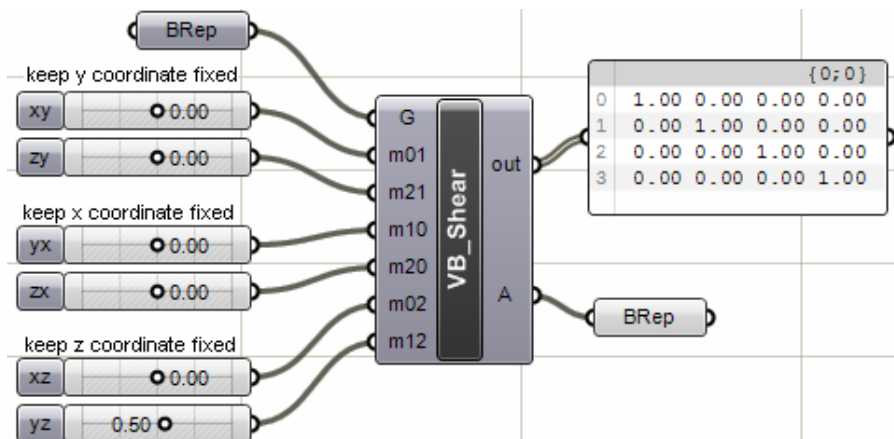
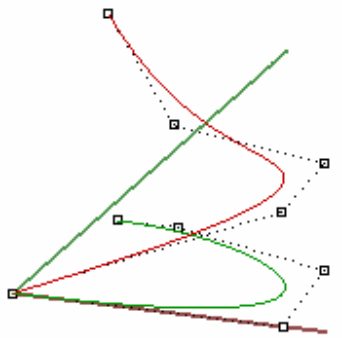


Figura (23): Matriz de sesgado

## Proyección plana

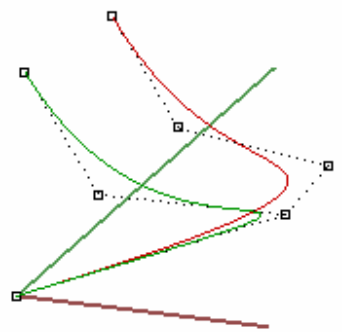
Intuitivamente, la proyección de un punto dado en el espacio  $P(x,y,z)$  sobre el plano  $xy$  global es igual a  $P_{xy}(x,y,0)$ . De forma similar, la proyección de un punto  $P$  en el plano  $xz$  es  $P_{xz}(x,0,z)$ . Estas proyecciones se denominan proyecciones ortogonales<sup>1</sup>.

Así que si tenemos una curva de partida y aplicamos una proyección plana, obtendremos una curva proyectada sobre ese plano. A continuación se muestra un ejemplo de la curva proyectada sobre plano  $xy$  con el formato matriz. Tenga en cuenta que las curvas NURBS (explicadas en el capítulo siguiente) utilizan los puntos de control para definir curvas. La proyección de una curva pasa por la proyección de sus puntos de control.



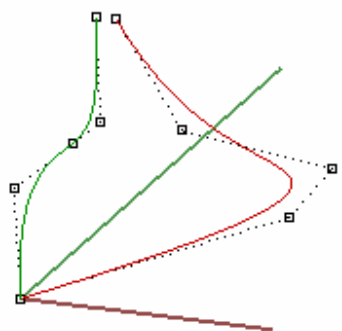
xy projection

0	1.00	0.00	0.00	0.00
1	0.00	1.00	0.00	0.00
2	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	1.00



xz projection

0	1.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00
2	0.00	0.00	1.00	0.00
3	0.00	0.00	0.00	1.00



yz projection

0	0.00	0.00	0.00	0.00
1	0.00	1.00	0.00	0.00
2	0.00	0.00	1.00	0.00
3	0.00	0.00	0.00	1.00

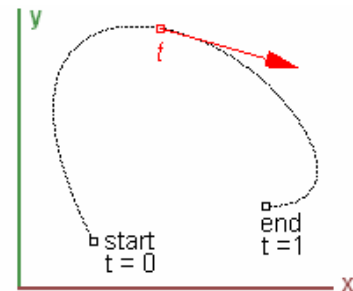
<sup>1</sup> Para más información, consulte: [http://en.wikipedia.org/wiki/Projection\\_\(linear\\_algebra\)](http://en.wikipedia.org/wiki/Projection_(linear_algebra))

## 3 Curvas y superficies paramétricas

### Introducción

Las curvas paramétricas son una forma muy compacta e intuitiva de representar curvas suaves. También son muy fáciles de modificar comparadas con otros formatos de representación. Por ejemplo, las polilíneas utilizan una discretización aproximada de primer grado, y por lo tanto, emplean una gran cantidad de puntos para almacenar una curva que debería ser suave. Además, la manipulación de la curva se convierte en tediosa, especialmente si es necesario mantener la suavidad de la curva. Cuanto más precisa es la curva, más pesado se convierte el almacenamiento de la curva y más complicada resulta su edición.

Una curva paramétrica es una función de un parámetro (normalmente denominado  $t$ )<sup>2</sup> sobre un dominio (normalmente entre 0 y 1). Considere por ejemplo el camino trazado por un caminante. El dominio es el tiempo transcurrido entre el comienzo y el final del viaje. La representación paramétrica da al caminante su posición exacta en cualquier instante, además de los lugares por los que ya ha pasado.



Tomemos por ejemplo un círculo. Recuerde que la ecuación del círculo es:

$$x^2 + y^2 = r^2$$

La ecuación paramétrica del círculo se define utilizando el parámetro " $t$ " así:

$$x = r \cos(t)$$

$$y = r \sin(t)$$

Para demostrar que ambas representan la misma curva, podemos obtener la ecuación original desde la paramétrica, tal que:

$$x/r = \cos(t)$$

$$y/r = \sin(t)$$

Y ya que:

$$\cos(t)^2 + \sin(t)^2 = 1 \text{ (identidad de Pitágoras)}$$

Por lo tanto:

$$(x/r)^2 + (y/r)^2 = 1, \text{ ó } x^2 + y^2 = r^2$$

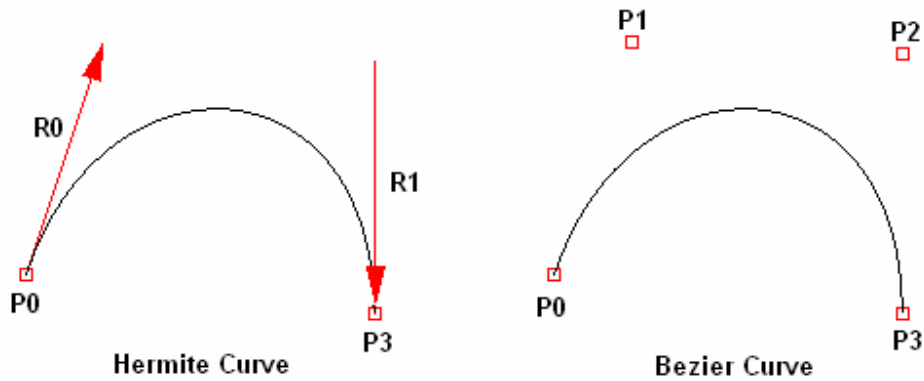
### Curvas cúbicas

Las curvas de Hermite<sup>3</sup> y Bézier<sup>4</sup> son dos ejemplos de curvas cúbicas determinadas por cuatro parámetros. Una curva de Hermite viene determinada por sus dos puntos extremos y dos vectores de dirección tangentes a estos puntos, mientras que una curva de Bézier la definen cuatro puntos. A pesar de ser diferentes matemáticamente, comparten características y limitaciones similares.

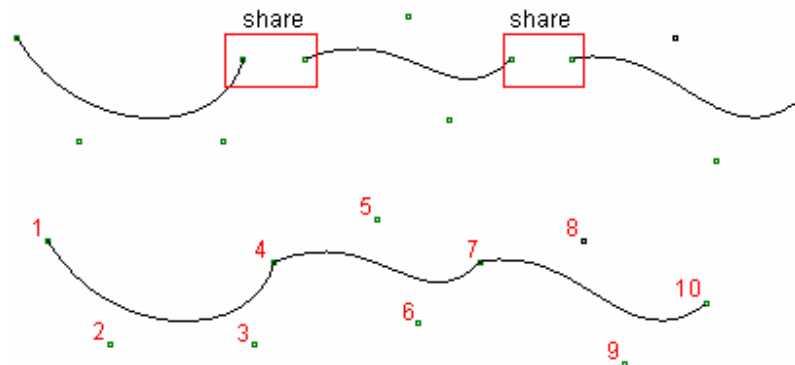
<sup>2</sup> Para más información, consulte: [http://en.wikipedia.org/wiki/Parametric\\_equation](http://en.wikipedia.org/wiki/Parametric_equation)

<sup>3</sup> Para más información, consulte: [http://en.wikipedia.org/wiki/Cubic\\_Hermite\\_spline](http://en.wikipedia.org/wiki/Cubic_Hermite_spline)

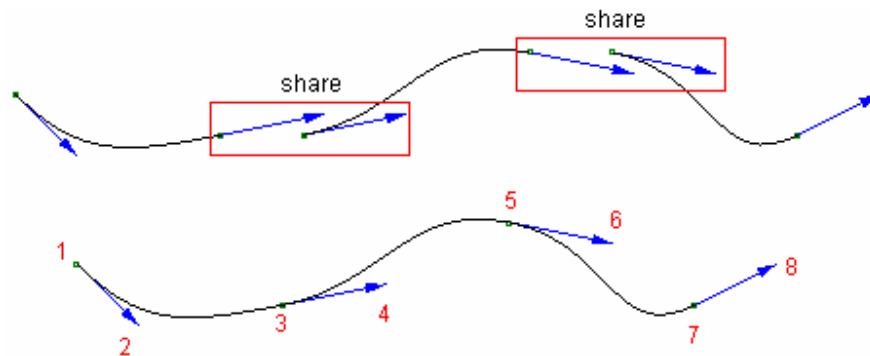
<sup>4</sup> Para más información, consulte: [http://en.wikipedia.org/wiki/B%C3%A9zier\\_curve](http://en.wikipedia.org/wiki/B%C3%A9zier_curve)



En la mayoría de los casos, las curvas están compuestas por multitud de segmentos. Esto requiere lo que se denomina curvas cúbicas segmentadas. Aquí se muestra una figura de una curva de Bézier que utiliza 10 puntos de almacenamiento para crear una curva de tres segmentos. Nótese que, aunque la última curva esta unida, no se ve suave.

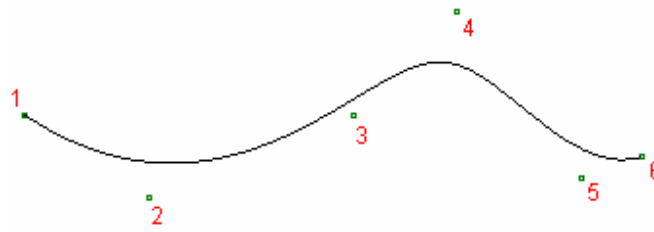


Aunque las curvas Hermite emplean la misma cantidad de parámetros que las de Bézier (4 para definir la curva), tenemos información adicional de la tangente que se puede compartir con el siguiente tramo para crear una curva de apariencia más suave con menos información:



Para obtener curvas más suaves y continuas, hay una representación de curvas muy potente denominada *Non Uniform Rational B-Spline*<sup>5</sup> (NURBS). Los segmentos comparten más puntos de control para conseguir más suavidad con menos información:

<sup>5</sup> Para más información, consulte: [http://en.wikipedia.org/wiki/Non-uniform\\_rational\\_B-spline](http://en.wikipedia.org/wiki/Non-uniform_rational_B-spline)



Las curvas y superficies NURBS son la representación matemática utilizada por Rhino para representar geometría. Las características y los componentes de las curvas NURBS se tratarán en detalle a lo largo de este capítulo.

Pero, ¿cómo es la ecuación paramétrica de una curva cúbica? Probablemente no tenga que utilizar estas ecuaciones para su trabajo, pero pensé que sería útil incluir la formulación genérica para tener una referencia.

La ecuación paramétrica de un segmento de curva cúbica:

$$\mathbf{Q}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}$$

Toma la forma:

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$$

$$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z$$

Podemos reescribir la ecuación  $\mathbf{Q}(t)$  tal que:

$$\mathbf{Q}(t) = \mathbf{C} \cdot \mathbf{T}$$

Donde  $\mathbf{T}$  es:

$$\mathbf{T} = \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

Y  $\mathbf{C}$  es la matriz de coeficientes:

$$\mathbf{C} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix}$$

Podemos verificar rápidamente que se puede obtener la forma original de la ecuación de la curva empleando la multiplicación de matrices:

$$\mathbf{Q}(t) = \mathbf{C} \cdot \mathbf{T} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix} \cdot \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} = \begin{bmatrix} a_x t^3 + b_x t^2 + c_x t + d_x \\ a_y t^3 + b_y t^2 + c_y t + d_y \\ a_z t^3 + b_z t^2 + c_z t + d_z \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}$$

## Continuidad geométrica

La continuidad geométrica es un concepto importante en el modelado en 3D. La continuidad es importante a la hora de conseguir suavidad visual o para obtener flujos de luz y aire suaves. La siguiente tabla muestra varias continuidades y sus definiciones:

- G0** (Cont. de posición)                      Dos segmentos de curva unidos
- G1** (Cont. de tangencia)                    La dirección de la tangente en los puntos de unión es la misma para ambos segmentos de curva
- G2** (Cont. de curvatura)                    Las curvaturas y las tangentes coinciden para los segmentos de curva y los puntos extremos comunes
- GN**    Las curvas coinciden en un orden mayor

El siguiente ejemplo compara la continuidad de curvatura entre las curvas “a” de un lado y “b”, “c”, y “d” del otro. Los componentes de GH calculan el vector tangente a cada curva en el punto P y la longitud de dicho vector. Esto es:

A = Vector tangente en el punto de unión  
 L = Módulo del vector

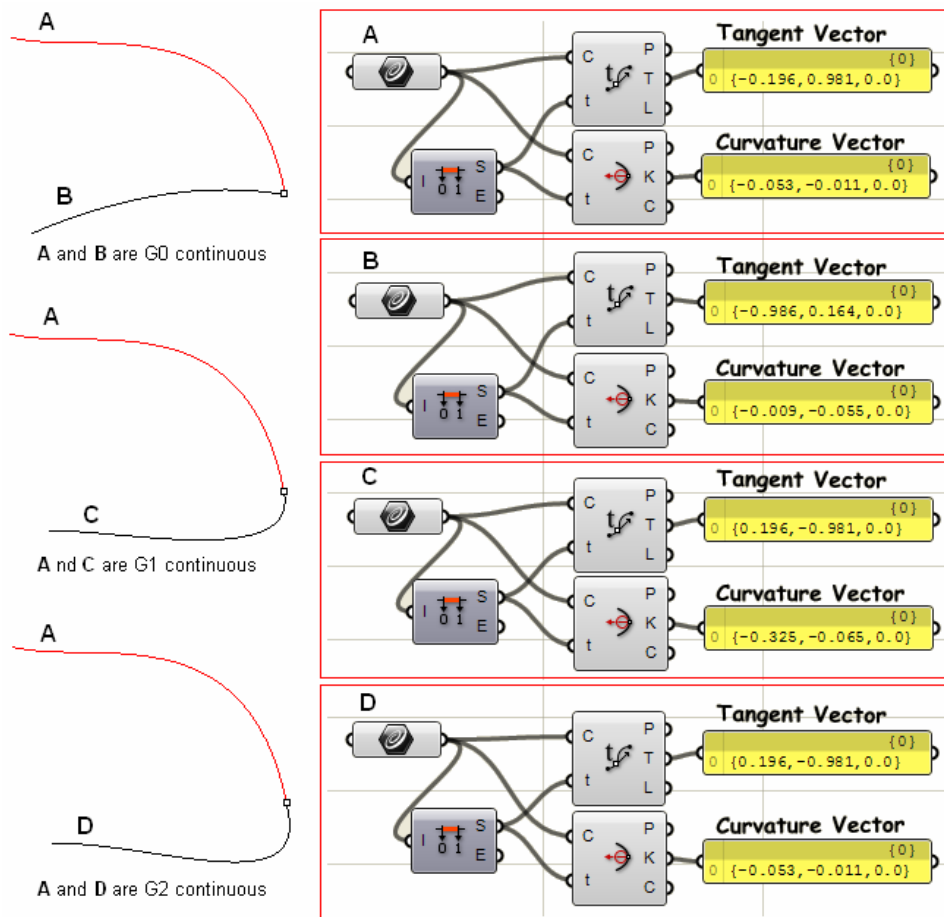


Figura (24): Examen de la continuidad de las curvas

Nótese que:

- Las curvas **A** y **B** tienen continuidad **G0** (vector tangente diferente en la unión)
- Las curvas **A** y **C** tienen continuidad **G1** (el mismo vector tangente en la unión)
- Las curvas **A** y **D** tienen continuidad **G2** (G1 y la curvatura coinciden en la unión)



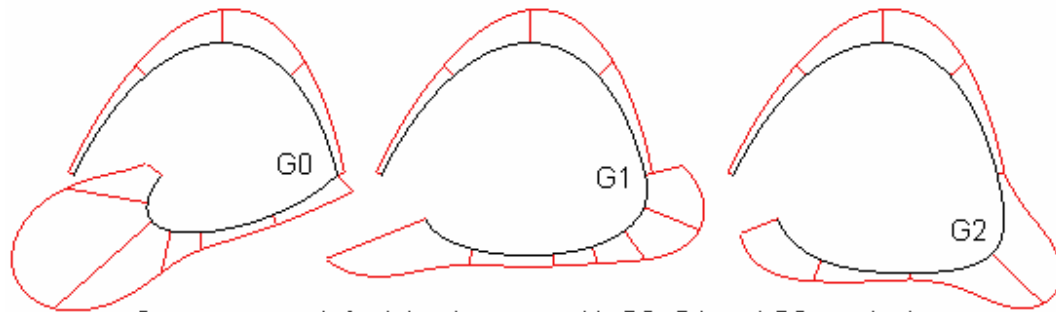
## Curvatura

La curvatura es un concepto ampliamente empleado en el modelado de curvas y superficies 3D. La curvatura se define como *el cambio en la inclinación de la tangente de una curva por unidad de longitud de arco*. Para un círculo o una esfera, es la inversa del radio.

Dado un punto cualquiera sobre una curva en un plano, la línea que mejor se aproxima a esta curva tocándola en dicho punto es la línea tangente. También puede hallarse el círculo que mejor se aproxima y que pasa por este punto, siendo tangente a la curva. La inversa del radio de este círculo es la curvatura de la curva en dicho punto.

El círculo que mejor se aproxima a la curva puede quedar a la izquierda o a la derecha de la misma. Teniendo esto en mente, podemos establecer una convención como, por ejemplo, la denominación de curvatura positiva si el círculo queda a la izquierda y negativa si queda a la derecha de la curva. Esto se conoce como curvatura señalizada.

El valor de la curvatura para un conjunto de curvas unidas es indicativo de la continuidad entre esas curvas, tal y como se muestra en la siguiente ilustración.



Curvature graph for joined curves with G0, G1 and G2 continuity

En superficies, la curvatura normal es una generalización de la curvatura de una superficie. Dado un punto en la superficie y una dirección contenida en el plano tangente a la superficie por dicho punto, la curvatura normal de la sección se halla mediante la intersección de la superficie con el plano formado por el punto, la normal a la superficie por dicho punto y la dirección anterior. La curvatura normal de la sección es la curvatura señalizada de esta curva en el punto dado.

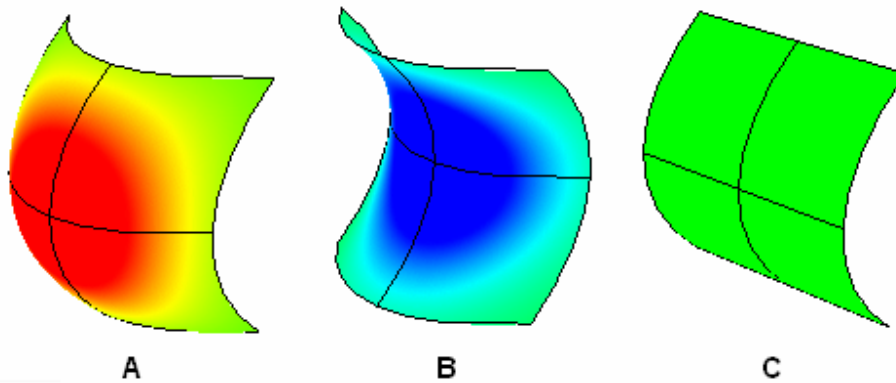
Si tenemos en cuenta todas las direcciones en el plano tangente a la superficie por nuestro punto y calculamos la curvatura normal en todas estas direcciones, entonces habrá un valor máximo y mínimo.

### Curvaturas principales

Las curvaturas principales de una superficie en un punto son la mínima y la máxima de las curvaturas normales en dicho punto. Las curvaturas principales se utilizan para calcular las curvaturas *Gaussiana* y *media* de la superficie.

### Curvatura Gaussiana

La curvatura Gaussiana de una superficie en un punto es el producto de las curvaturas principales de dicho punto. El plano tangente de cualquier punto con curvatura Gaussiana positiva toca la superficie localmente en un sólo punto, mientras que el plano tangente de cualquier punto con curvatura Gaussiana negativa corta la superficie.



A: Curvatura positiva cuando la superficie es cóncava

B: Curvatura negativa cuando la superficie tiene forma de silla de montar

C: Curvatura cero cuando la superficie es plana en al menos una dirección (plano, cilindro, etc.)

### Curvatura media

La curvatura media de una superficie en un punto dado es la media de la suma de las curvaturas principales en dicho punto. Cualquier punto con curvatura media igual a cero tiene curvatura Gaussiana igual a cero o negativa.

Las superficies con curvatura media igual a cero en toda su superficie se denominan superficies mínimas. Las superficies con curvatura media constante en toda su superficie se suelen denominar superficies de curvatura media constante (CMC).

Los procesos físicos que se pueden modelar con superficies CMC incluyen la formación de burbujas de jabón, tanto libres como adheridas a objetos. Una burbuja de jabón, al contrario que una lámina simple de jabón, delimita un volumen y se mantiene en equilibrio gracias a que la presión ligeramente mayor de su interior se equilibra con las fuerzas que tienden a minimizar el área de la burbuja misma.

Las superficies mínimas son una subcategoría de las superficies CMC, donde la curvatura constante es igual a cero.

Los procesos físicos que se pueden modelar con superficies mínimas incluyen la formación de películas de jabón entre bordes de objetos, como por ejemplo anillos de alambre. Las películas de jabón no se distorsionan por la presión del aire (que es igual en ambas caras) y tienen libertad para minimizar su área. Esto contrasta con las burbujas de jabón, que encierran una cantidad constante de aire y soportan presiones de aire diferentes en el exterior del interior. La curvatura media es útil para encontrar áreas con saltos bruscos en la curvatura de la superficie.

## Algoritmos para la evaluación de curvas paramétricas

### Algoritmo de De Casteljaú<sup>6</sup> para evaluar curvas cúbicas de Bézier

Denominado gracias a su inventor, Paul De Casteljaú, este algoritmo evalúa las curvas de Bézier utilizando un método iterativo.

Vamos a mostrar el algoritmo para hallar cualquier punto sobre la curva en el parámetro  $t$  con el algoritmo de De Casteljaú utilizando Grasshopper. Necesitaremos los siguientes datos:

4 puntos A, B, C, D

$t$ , que es el parámetro de la curva en el dominio (0-1)

Resultados:

Punto sobre la curva en el parámetro  $t$

Pasos de la solución:

1. Encontrar el punto M, parámetro  $t$  de AB
2. Encontrar el punto N, parámetro  $t$  de BC
3. Encontrar el punto O, parámetro  $t$  de CD
4. Encontrar el punto P, parámetro  $t$  de MN
5. Encontrar el punto Q, parámetro  $t$  de NO
6. Encontrar el punto R, parámetro  $t$  de PQ

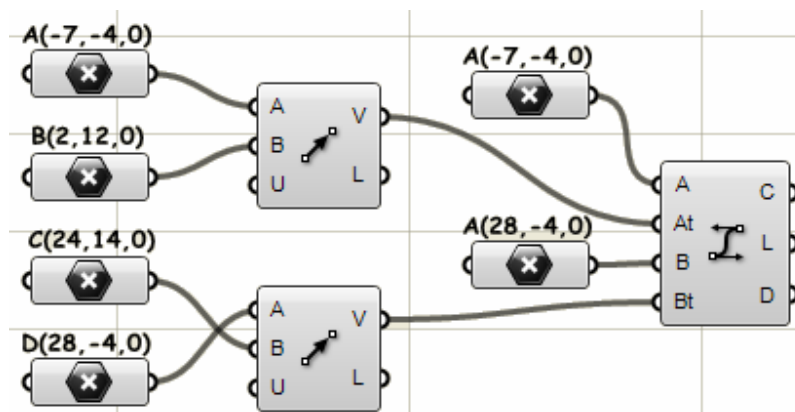
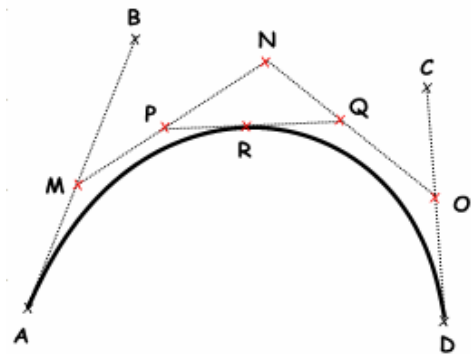


Figura (25): Creación de curva de Bézier en GH

Ésta es la definición de Grasshopper para evaluar un parámetro sobre una curva de Bézier utilizando el algoritmo de De Casteljaú. Nótese que se puede variar el valor de  $t$  de 0 a 1 para hallar puntos entre el inicio y el final de la curva.

<sup>6</sup> Más información sobre el algoritmo de De Casteljaú en: [http://en.wikipedia.org/wiki/De\\_Casteljau%27s\\_algorithm](http://en.wikipedia.org/wiki/De_Casteljau%27s_algorithm)

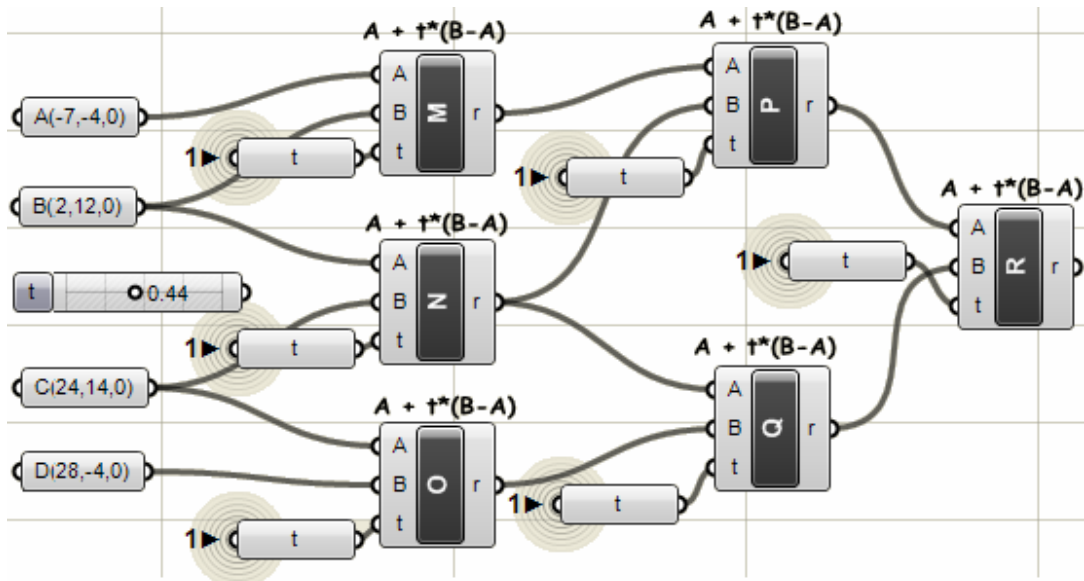


Figura (26): Evaluación de puntos en una curva de Bézier empleando el algoritmo de De Casteljau

### Algoritmo de De Boor<sup>7</sup> para evaluar curvas NURBS

El algoritmo de De Boor es una generalización del algoritmo de De Casteljau para curvas de Bézier. Es estable numéricamente y su uso está muy extendido para evaluar puntos en curvas NURBS dentro de las aplicaciones 3D. Aquí se muestra un ejemplo para evaluar un punto en una curva NURBS de grado 3 utilizando el algoritmo de De Boor<sup>8</sup>.

Datos de entrada:

7 puntos de control de  $P_0$  a  $P_6$

Vectores nodales:

$$u_0 = 0.0$$

$$u_1 = 0.0$$

$$u_2 = 0.0$$

$$u_3 = 0.25$$

$$u_4 = 0.5$$

$$u_5 = 0.75$$

$$u_6 = 1.0$$

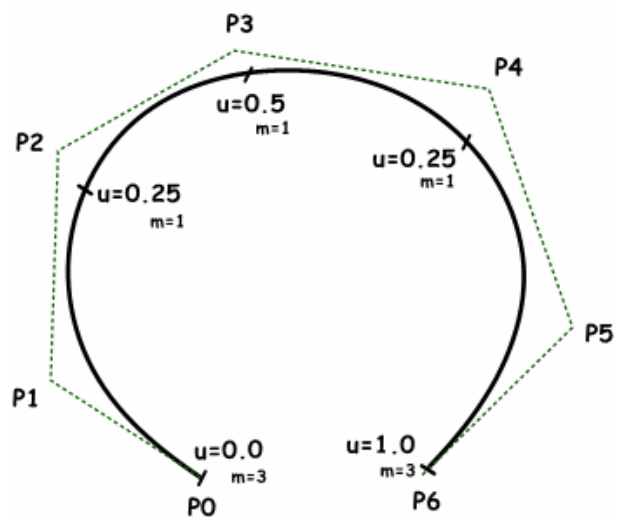
$$u_7 = 1.0$$

$$u_8 = 1.0$$

Resultado:

Punto de la curva que se encuentra en  $u=0.4$

Pasos de la solución:



<sup>7</sup> Más detalle sobre el algoritmo de De Boor en [http://en.wikipedia.org/wiki/De\\_Boor's\\_algorithm](http://en.wikipedia.org/wiki/De_Boor's_algorithm)

<sup>8</sup> La descripción general del algoritmo y los detalles de este ejemplo pueden consultarse en: <http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/spline/de-Boor.html>

1. Calcular los coeficientes de la primera iteración:

$$A_c = (u - u_1) / (u_{1+3} - u_1) = 0.8$$

$$B_c = (u - u_2) / (u_{2+3} - u_2) = 0.53$$

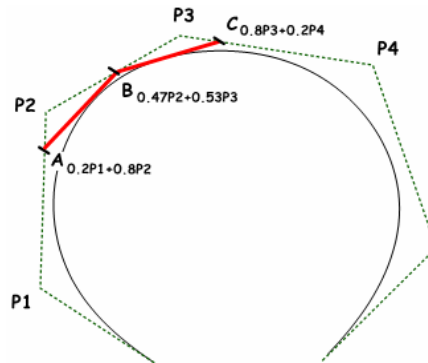
$$C_c = (u - u_3) / (u_{3+3} - u_3) = 0.2$$

2. Calcular los puntos utilizando datos de los coeficientes:

$$A = 0.2P_1 + 0.8P_2$$

$$B = 0.47P_2 + 0.53P_3$$

$$C = 0.8P_3 + 0.2P_4$$



3. Calcular los coeficientes de la segunda iteración:

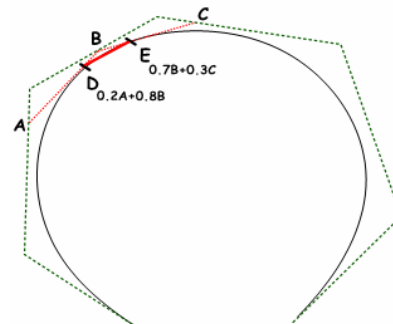
$$D_c = (u - u_2) / (u_{2+3-1} - u_2) = 0.8$$

$$E_c = (u - u_3) / (u_{3+3-1} - u_3) = 0.3$$

4. Calcular los puntos utilizando datos de los coeficientes:

$$D = 0.2A + 0.8B$$

$$E = 0.7B + 0.3C$$

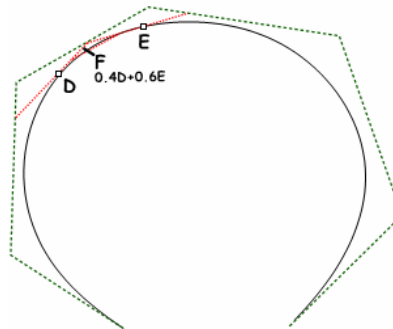


5. Calcular el último coeficiente:

$$F_c = (u - u_3) / (u_{3+3-2} - u_3) = 0.6$$

6. Encontrar el punto de la curva en el parámetro  $u=0.4$

$$F = 0.4D + 0.6E$$



Ésta es la definición de Grasshopper para evaluar el parámetro  $u=0.4$  de una curva NURBS utilizando el algoritmo de De Boor.

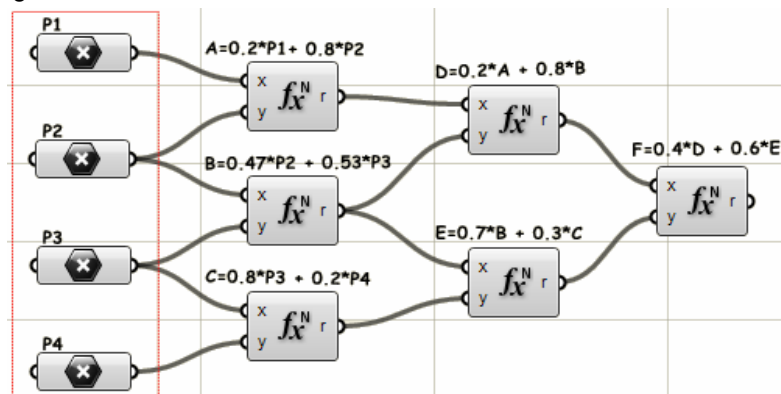


Figura (27): Evaluación de un punto sobre una curva NURBS utilizando el algoritmo de De Boor

## Curvas NURBS


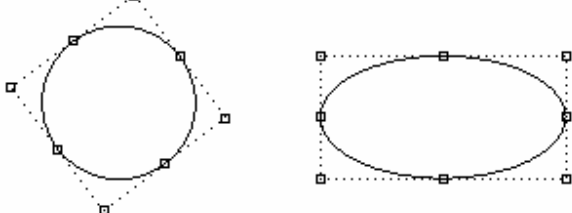
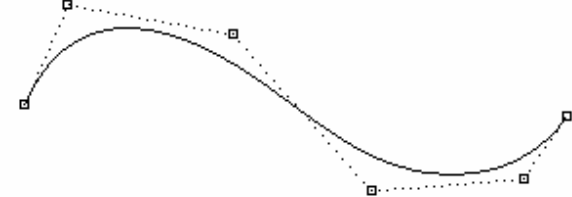
NURBS es un método exacto de representación matemática de curvas y superficies muy intuitivo a la hora de manipularlo.

Existen multitud de libros y referencias para aquellos interesados en investigar más en profundidad sobre NURBS (<http://en.wikipedia.org/wiki/NURBS>). Para utilizar un modelador NURBS más eficientemente, es necesaria una comprensión básica del concepto.

Una curva NURBS viene definida por cuatro propiedades: grado, puntos de control, nodos y reglas de evaluación:

### Grado

El grado es un número entero positivo. Rhino permite trabajar con cualquier grado empezando por el 1. El grado 5 es de uso común, pero números por encima de 5 carecen de utilidad en el mundo real. Aquí se muestran algunos ejemplos de curvas con sus grados:

<p>Las <b>líneas</b> y las <b>polilíneas</b> son curvas NURBS de grado 1. Orden = 2 (orden = grado + 1)</p>	
<p>Los <b>círculos</b> y los <b>elipses</b> son ejemplos de curvas NURBS de grado 2. También son curvas racionales o no-uniformes. Orden = 3</p>	
<p>Las <b>curvas de forma libre</b> suelen representarse como curvas NURBS de grado 3. Orden = 4</p>	

### Puntos de control

Los puntos de control de una curva NURBS son una lista de al menos (grado+1) puntos. La manera más común de modificar la forma de una curva NURBS es moviendo sus puntos de control.

Los puntos de control llevan asociado un número llamado **peso**. Con algunas excepciones, los pesos suelen ser número positivos. Cuando todos los puntos de control de una curva poseen el mismo peso (normalmente 1), la curva se denomina no racional. Mostraremos un ejemplo de cómo alterar el peso de los puntos de control interactivamente con Grasshopper.

### Nodos y vectores nodales

Cada curva NURBS tiene una lista de números asociados a ella denominados vectores nodales. Los nodos son algo más difíciles de comprender y manejar, pero afortunadamente existen multitud de funciones SDK que harán el trabajo por nosotros. En cualquier caso, existen algunos conceptos que nos serán de utilidad sobre los vectores nodales.

### Los nodos son valores de un parámetro

Los nodos son una lista no decreciente de valores de parámetros. Existen 'el grado de la curva menos 1' más nodos que puntos de control. Normalmente, para curvas no periódicas, tantos nodos como el primer grado son iguales y tantos nodos como el último grado son también iguales. El dominio de la curva se encuentra entre estos valores de los nodos extremos.

### Multiplicidad de nodos

La multiplicidad de un nodo es el número de veces que se encuentra en la lista de vectores nodales. La multiplicidad de un nodo no puede ser mayor que el grado de la curva. La multiplicidad nodal se utiliza para controlar la continuidad en el correspondiente punto de la curva.

### Nodo de multiplicidad completa

Un nodo de multiplicidad completa tiene multiplicidad igual al grado de la curva. En un punto de multiplicidad completa existe un punto de control que le corresponde, y la curva pasa por este punto.

Por ejemplo, las curvas ancladas tienen nodos de multiplicidad completa en los extremos de la curva. Esta es la razón por la que los puntos de control extremos coinciden con los puntos extremos de la curva. Un punto interior con multiplicidad genera un pico en la curva en el punto correspondiente.

### Nodo simple

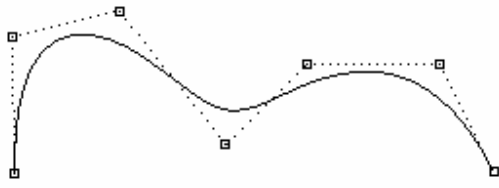
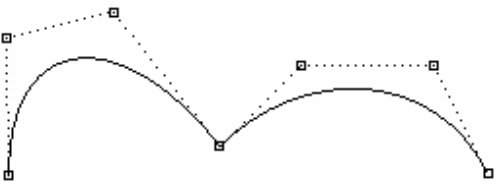
Un nodo cuyo valor aparece sólo una vez.

### Vector nodal uniforme

Un vector nodal uniforme satisface dos condiciones:

1. El número de nodos es igual al número de puntos de control + grado - 1.
2. Los nodos comienzan con un nodo de multiplicidad completa, seguido de nodos simples y terminan en otro nodo de multiplicidad completa. Los valores son crecientes a intervalos regulares. Esto es típico de las curvas ancladas. Las curvas periódicas funcionan de manera diferente, como veremos más adelante.

Aquí tenemos dos curvas con puntos de control idénticos pero vectores nodales diferentes:

<p>Grado = 3                      Número de puntos de control = 7                      Vector nodal = (0,0,0,1,2,3,5,5)</p>	
<p>Grado = 3                      Número de puntos de control = 7                      Vector nodal = (0,0,0,1,1,1,4,4,4)  <b>Nota:</b> un nodo con multiplicidad completa en el centro crea un pico y obliga a la curva a pasar por el punto de control asociado.</p>	

### Regla de evaluación

La regla de evaluación utiliza una fórmula matemática que toma un número y le asigna un punto. En la fórmula están involucrados el grado, los puntos de control y los nodos.

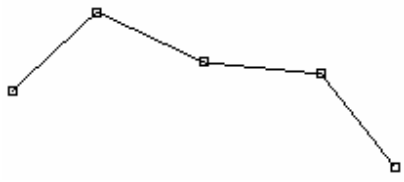
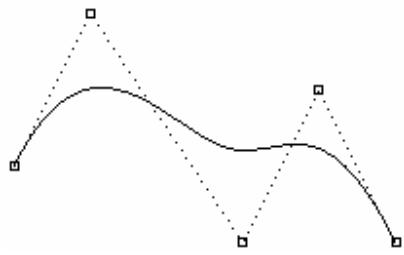
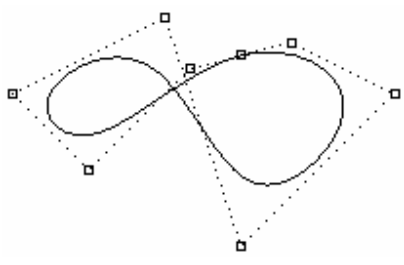
Empleando esta fórmula, las funciones SDK pueden tomar un parámetro de la curva y producir el punto correspondiente de esa curva. El parámetro es un número contenido dentro del dominio de la curva. Los dominios son normalmente crecientes y consisten en dos números: parámetro mínimo del dominio ( $m_t(0)$ ) que suele ser el principio de la curva y el máximo ( $m_t(1)$ ) al final de la curva.

## Características de las curvas NURBS

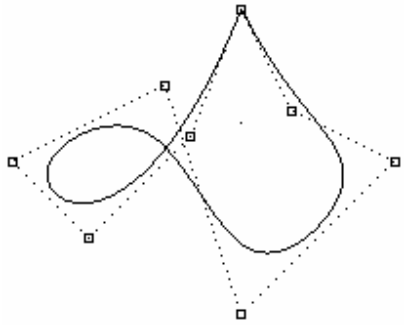
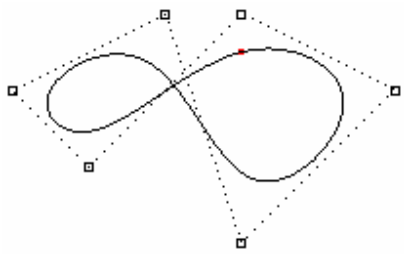
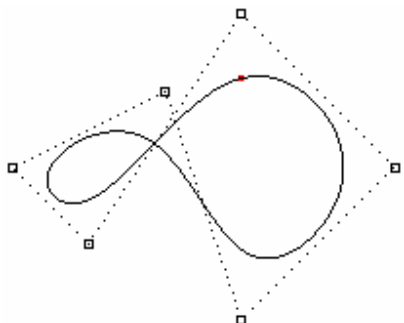
Para generar una curva NURBS, necesitará la siguiente información:

- Dimensión, normalmente 3.
- Grado (algunas veces se utiliza el “orden” que es el grado + 1)
- Puntos de control (una familia de puntos)
- Vectores nodales (una familia de números)
- Especificar si la curva es racional (se explicará la noción de la racionalidad de una curva en el apartado de las potencias)

Utilizando un modelador en 3D, normalmente necesitará especificar el grado de la curva y los puntos de control. El resto de la información necesaria para construir la curva NURBS se generará automáticamente. Si se hace que el último punto coincida con el primero, lo usual será crear una curva cerrada periódica y suavizada. En la siguiente figura se muestran una curva abierta (anclada), una curva cerrada no periódica y una curva cerrada periódica. Hablaremos sobre las curvas periódicas vs. ancladas en la próxima sección.

<p>Curva abierta de grado 1. Nótese que la curva pasa por todos los puntos de control.</p>	
<p>Curva abierta de grado 3. Ambos extremos de la curva coinciden con los puntos de control extremos.</p>	
<p>Curva cerrada (no periódica) de grado 3. Los puntos inicial y final se superponen sobre un punto de control.</p>	



<p>Al mover el punto de control de una curva no periódica se produce un pico y la curva no parece suavizada</p>	 A diagram showing a non-periodic curve (a teardrop shape) defined by five control points (small squares) connected by a dotted line. The curve is smooth and follows the general shape of the control points.
<p>Curva periódica cerrada de grado 3. Nótese que el punto inicial/final de la curva no pasa por un punto de control. Este punto se denomina "cierre". Está representado en rojo en la figura adjunta.</p>	 A diagram showing a closed periodic curve (a figure-eight shape) defined by five control points (small squares) connected by a dotted line. The curve is smooth and follows the general shape of the control points. A red dot on the curve indicates the closure point, which is not one of the control points.
<p>Mover los puntos de control de la curva periódica no afecta a la suavidad de la misma ni produce ningún pico.</p>	 A diagram showing a closed periodic curve (a figure-eight shape) defined by five control points (small squares) connected by a dotted line. The curve is smooth and follows the general shape of the control points. A red dot on the curve indicates the closure point, which is not one of the control points.

### Curvas NURBS ancladas vs. periódicas

Las curvas ancladas son curvas (normalmente abiertas) cuyos extremos coinciden con los puntos de control extremos. Las curvas periódicas son curvas cerradas suavizadas. La mejor manera de comprender la diferencia entre ambas es comparando sus puntos de control.

El siguiente componente crea una curva NURBS anclada y muestra los puntos de control y vectores nodales:

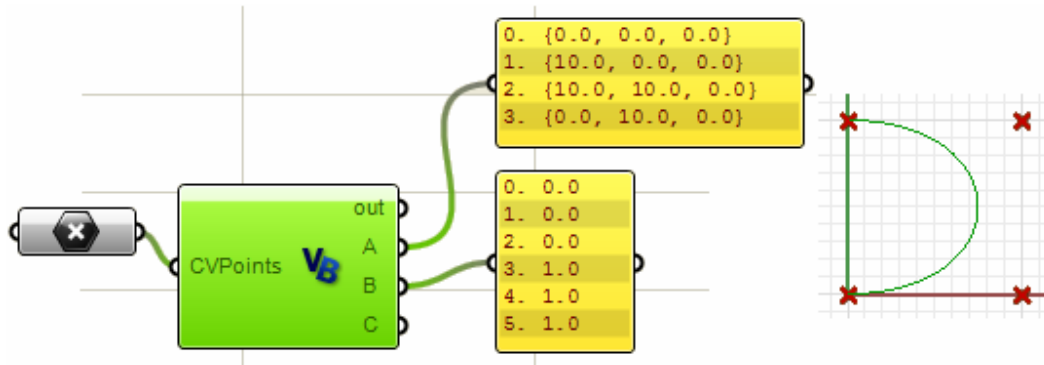


Figura (28): Análisis de curvas NURBS ancladas

Y aquí tenemos la curva periódica utilizando los mismos datos de entrada (puntos de control y grado de la curva):

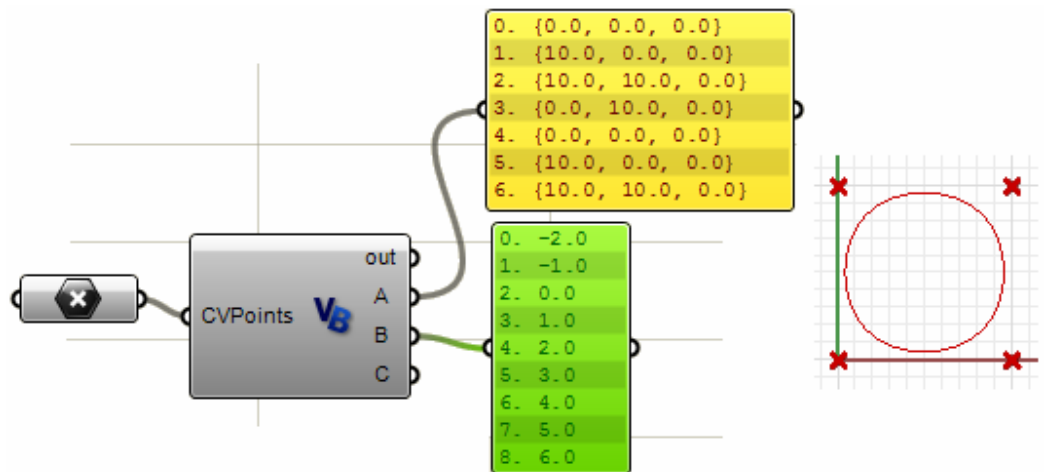


Figura (29): Análisis de curvas NURBS periódicas

Nótese que en la curva periódica los cuatro puntos de control se convierten en siete (4 + grado), mientras que la curva anclada emplea sólo cuatro puntos de control. El vector nodal de la curva periódica contiene sólo nodos simples, mientras que los nodos inicial y final de la curva anclada tienen multiplicidad completa.

Aquí se muestran algunos ejemplos de curvas de grado 2. Como puede deducirse, el número de puntos de control y los nodos de las curvas periódicas cambian cuando el grado cambia.

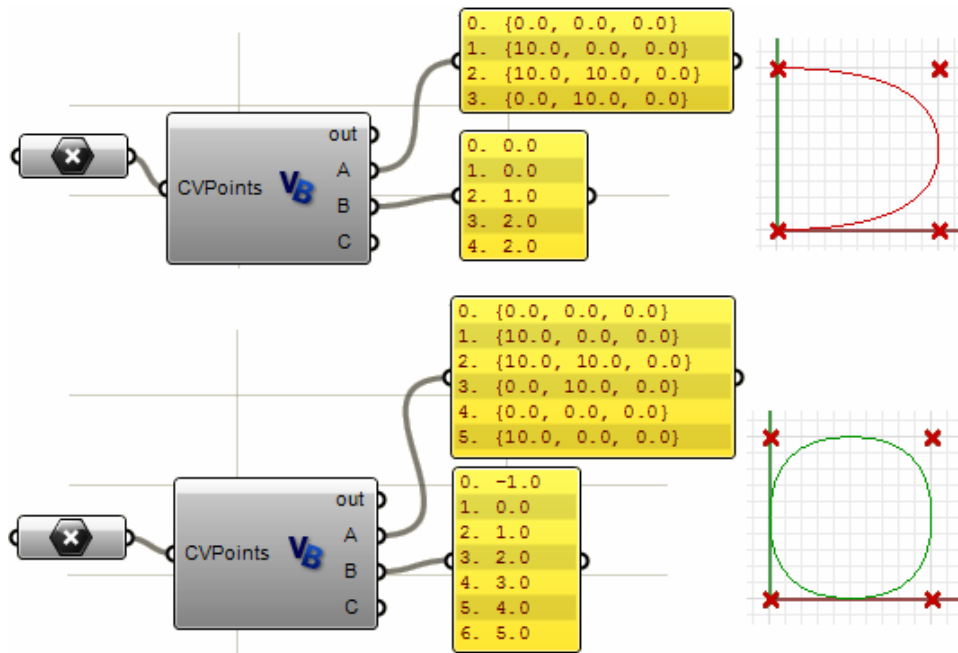


Figura (30): Análisis de curvas NURBS de grado 2

### Potencia

La potencia de los puntos de control en una curva NURBS uniforme es 1, pero este número puede variar en las curvas NURBS racionales. El siguiente ejemplo nos muestra cómo modificar interactivamente el peso de los puntos de control en Grasshopper.

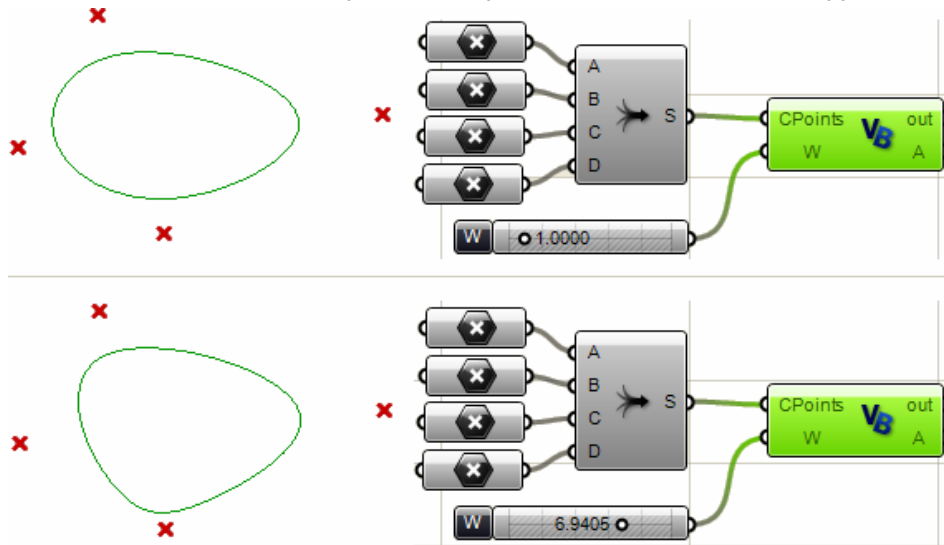


Figura (31): Análisis de potencias en curvas NURBS

### Superficies NURBS

Podemos pensar en las superficies NURBS como en una malla de curvas NURBS en dos direcciones. La forma de una superficie NURBS está definida por el número de puntos de control y el grado de la superficie en cada una de las dos direcciones (u- y v-).

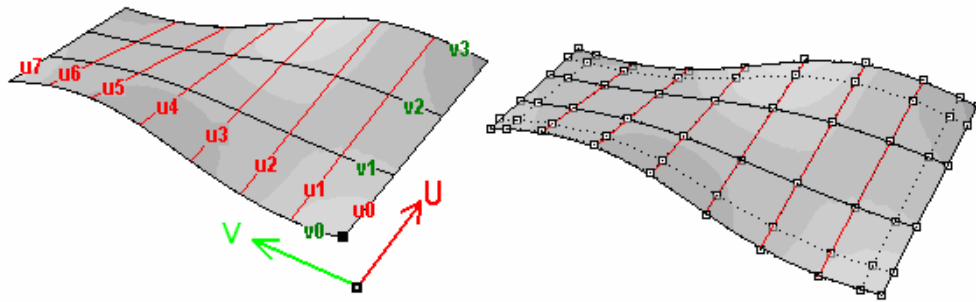


Figura (31): Dominio de una superficie NURBS

Las superficies NURBS pueden ser recortadas (*trimmed*) o no recortadas (*untrimmed*). Las superficies recortadas utilizan otra superficie NURBS subyacente y curvas cerradas para recortar la forma específica en dicha superficie. Cada superficie tiene una curva cerrada que defina el borde exterior y curvas interiores que no se intersecan para definir agujeros. Una superficie con el borde exterior igual que el de su superficie NURBS subyacente y sin agujeros es lo que se denomina superficie no recortada.

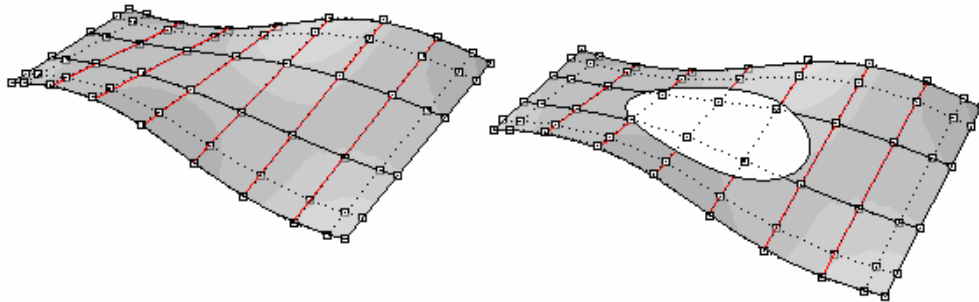


Figura (32): Superficie NURBS recortada

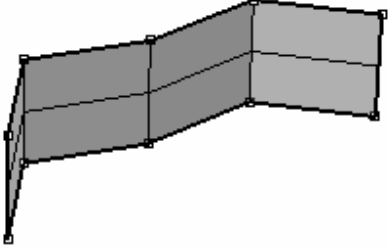
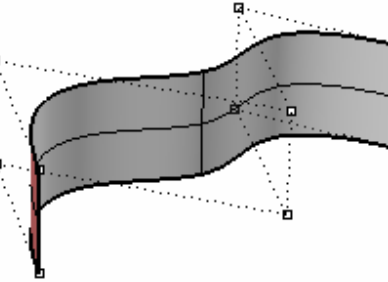
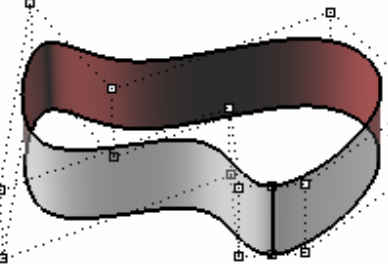
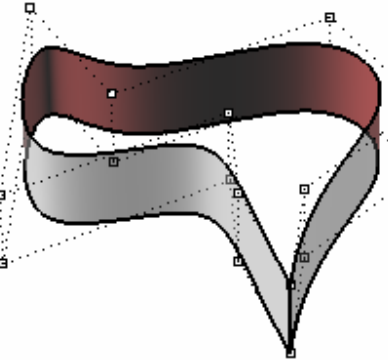
La superficie de la izquierda es no recortada. La superficie de la derecha es la misma pero recortada con un agujero elíptico. Nótese que la estructura NURBS de la superficie no cambia con el recorte.

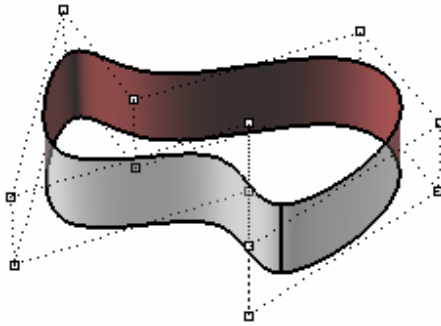
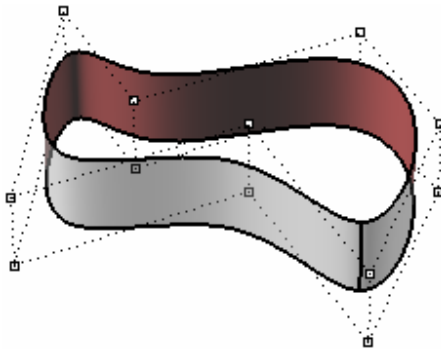
## Características de las superficies NURBS

Las características de las superficies NURBS son muy similares a las de las curvas NURBS excepto en que tienen una dirección adicional. Las superficies NURBS contienen la siguiente información:

- Dimensión, típicamente 3.
- Grado en las direcciones  $u$  y  $v$ : (algunas veces se utiliza el "orden" que es el grado + 1)
- Puntos de control en las direcciones  $u$  y  $v$ .
- Vector nodal (familia de números 2D).
- Especificación de si la superficie es racional.

Como con las curvas NURBS, es muy poco probable que necesite saber los detalles de cómo crear una superficie NURBS ya que su modelador de 3D probablemente tenga buenas herramientas para ayudarle. Siempre se pueden reconstruir las superficies (y curvas) a un nuevo grado y número de puntos de control. Además, la superficie puede ser abierta, cerrada o periódica. Aquí se muestran diversos ejemplos con diferentes grados, número de puntos de control y si son abiertas o periódicas:

<p>Superficie de grado 1 en ambas direcciones <math>u</math> y <math>v</math>. Todos los puntos de control se encuentran en la superficie.</p>	
<p>Superficie abierta de grado 3 en la dirección <math>u</math> y grado 1 en la dirección <math>v</math>. Nótese cómo las esquinas de la superficie coinciden con los puntos de control.</p>	
<p>Superficie cerrada no periódica de grado 3 en la dirección <math>u</math> y grado 1 en la dirección <math>v</math>. Nótese cómo existen puntos de control que coinciden con la junta de la superficie.</p>	
<p>Al mover los puntos de control de una superficie cerrada no periódica se crea un pico y la superficie deja de ser suave.</p>	

<p>Superficie periódica de grado 3 en la dirección <math>u</math> y grado 1 en la dirección <math>v</math>. Nótese que los puntos de control de la superficie no coinciden con la junta.</p>	
<p>El movimiento de los puntos de control de una superficie periódica no afecta a su suavidad ni produce picos.</p>	

## Polisuperficies

Una polisuperficie está formada por dos o más superficies NURBS (posiblemente recortadas) unidas entre ellas. Cada superficie tiene su propia parametrización y direcciones  $u$ - y  $v$ -, que no tienen por qué coincidir. Las polisuperficies y superficies recortadas se representan utilizando lo que se denomina *boundary representation* (representación de bordes), abreviado *brep*. Básicamente describe las superficies, aristas y geometría de los vértices con datos de los recortes y relaciones entre las diferentes partes. Por ejemplo, una *brep* describe cada cara, las aristas que la rodean y recortes, dirección normal relativa a la superficie, relaciones con las caras adyacentes, etc. Las *breps* pueden denominarse también *sólidos* cuando son cerrados o no tienen ninguna abertura.

En el caso de la siguiente caja, está compuesta por 6 superficies no recortadas unidas entre ellas:

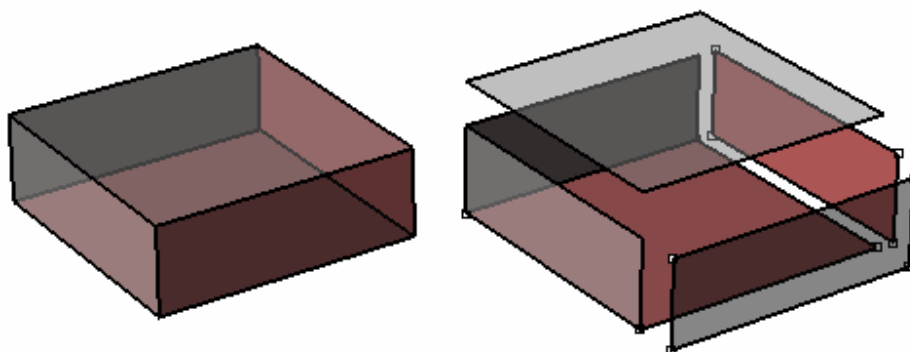


Figura (33): Las polisuperficies están formadas por superficies NURBS unidas entre sí

Muchas polisuperficies están formadas por superficies recortadas unidas entre sí. Las caras superior e inferior del siguiente cilindro están recortadas a partir de superficies planas.

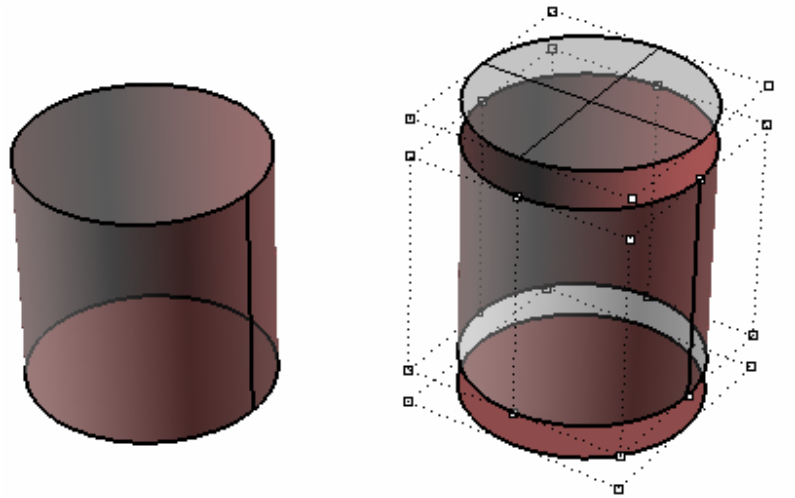


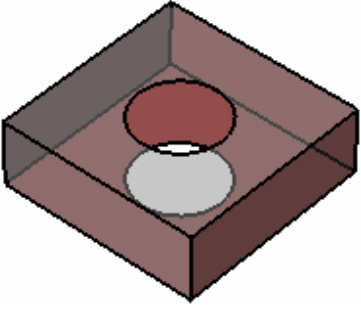
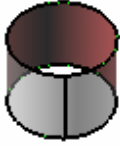
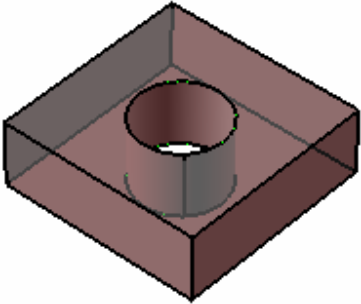
Figura (34): Las caras de una polisuperficie pueden ser superficies NURBS recortadas

Ya vimos que la edición de curvas y superficies NURBS es muy intuitiva y puede realizarse interactivamente moviendo puntos de control. Pero editar polisuperficies y mantener unidas las aristas de las diferentes caras no es tan intuitivo. Existen multitud de herramientas en los modeladores 3D NURBS para editar y deformar polisuperficies directamente, pero también es muy común descomponer una polisuperficie en sus superficies originales, editarlas y después volver a unir las. En este proceso, el usuario debe mantener la alineación correcta de las aristas respecto a la tolerancia para que se unan correctamente.

### Ejemplo

Muestre los pasos para crear un agujero circular en una caja.

<p>Comience con una caja y un círculo que marque la situación y el diámetro del agujero.</p>	
<p>Proyecte el círculo sobre la caja.</p>	

<p>Utilice la proyección de los círculos para recortar las caras superior e inferior de la caja.</p>	
<p>Utilice la proyección de los círculos para crear una superficie entre ellos.</p>	
<p>Una las superficies recortadas y la pared del agujero todo junto.</p>	



## **Referencias**

Edward Angel, "Interactive Computer Graphics with OpenGL," Addison Wesley Longman, Inc., 2000.

James D Foley, Steven K Feiner, John F Hughes, "Introduction to Computer Graphics" Addison-Wesley Publishing Company, Inc., 1997.

James Stewart, "Calculus," Wadsworth, Inc, 1991.

Kenneth Hoffman, Ray Kunze, "Linear Algebra", Prentice-Hall, Inc., 1971

Rhinoceros® help document, Robert McNeel and Associates, 2009.