

# Modelado Geométrico

---

Introducción

Modelado plano de superficies

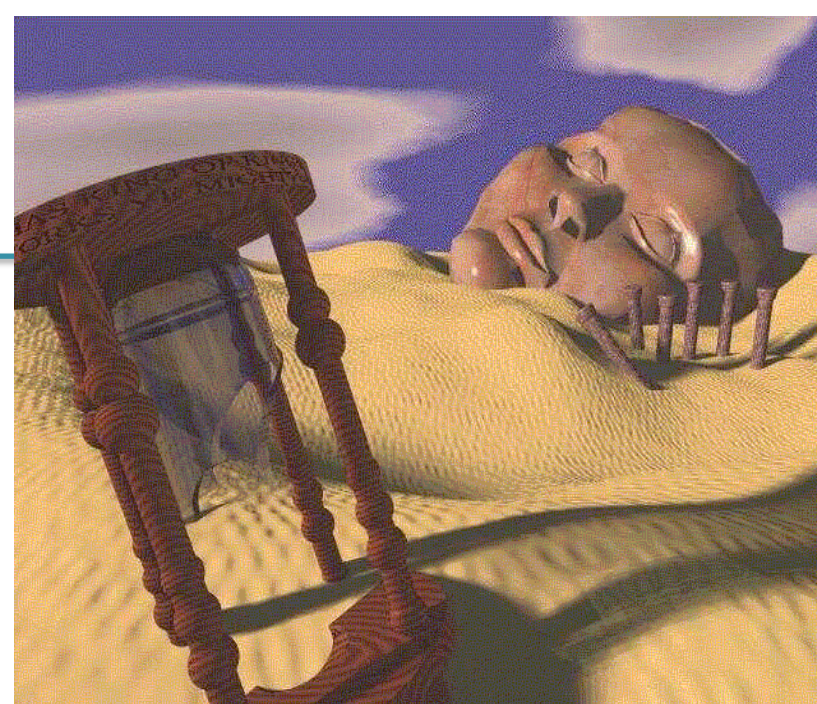
Modelado curvo de superficies

Modelado de sólidos

Otros modelos

# Introducción

Una escena puede contener distintos tipos de objetos (nubes, árboles, rocas, edificios, mobiliario, etc.) para los que existen una gran variedad de modelos de representación



- ▶ Superficies poligonales y cuádricas
- ▶ Superficies *spline*
- ▶ Modelos volumétricos
- ▶ Modelado sólido (de frontera *B-Rep*, por subdivisión espacial *Octrees*)
- ▶ Modelos procedurales (Fractales, Sistemas de partículas,...)
- ▶ Modelos basados en propiedades físicas

# Introducción

---

- ▶ Un sistema de modelado está formado por:
  - ▶ Estructura de datos:
    - ▶ Principal y auxiliares
    - ▶ Exactas y aproximadas
  - ▶ Operaciones
    - ▶ Visualización
    - ▶ Edición
    - ▶ Conversión
    - ▶ Determinación de propiedades
- ▶ Requisitos que debe cumplir para modelar sólidos:
  - ▶ No ambiguo: dos objetos distintos no pueden responder a la misma representación
  - ▶ El dominio de representación se debe ajustar al área de aplicación
  - ▶ Válido: No se debe permitir la representación de objetos que no correspondan a un sólido
  - ▶ Compacto: No redundante
  - ▶ Eficiente en el manejo

# Introducción

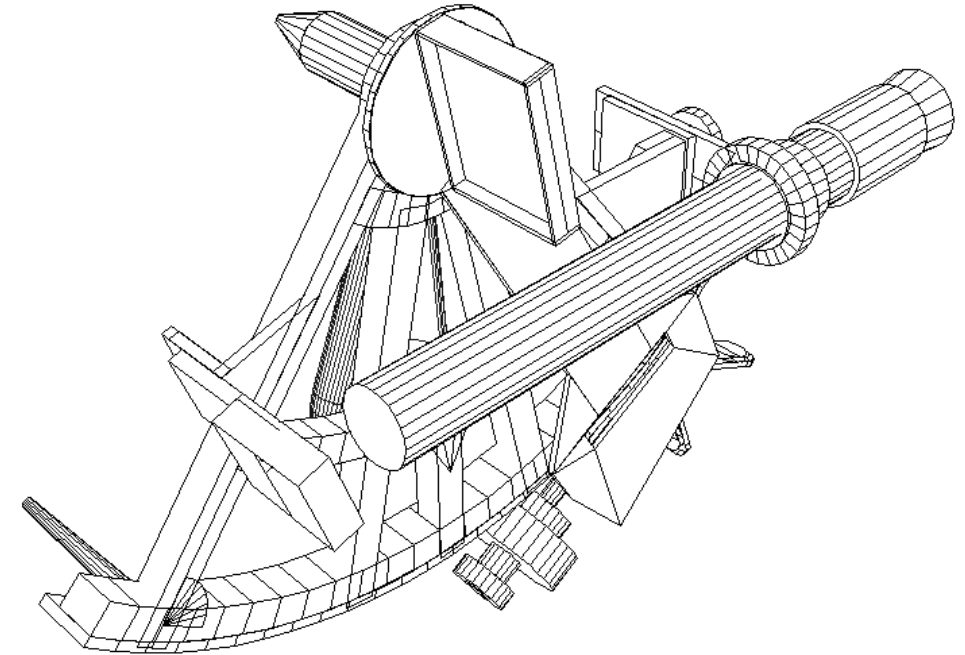
- ▶ Creación de los objetos a modelar:
  - ▶ Con una aplicación de modelado (3DStudio, ProEngineer,...)
  - ▶ A partir de objetos reales (explorador láser, digitalizador 3D)
  - ▶ Matemáticamente





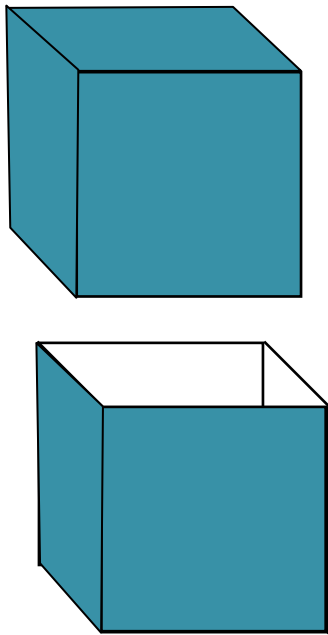
# Modelado plano de superficies

- ▶ Modelo alámbrico
  - ▶ Se utilizaba cuando los monitores eran vectoriales
  - ▶ Elementos: puntos, líneas, arcos y círculos, cónicas, y curvas
  - ▶ Ventajas: fácil de construir, pocas necesidades de memoria y almacenamiento
  - ▶ En la actualidad, se utiliza como visualización auxiliar

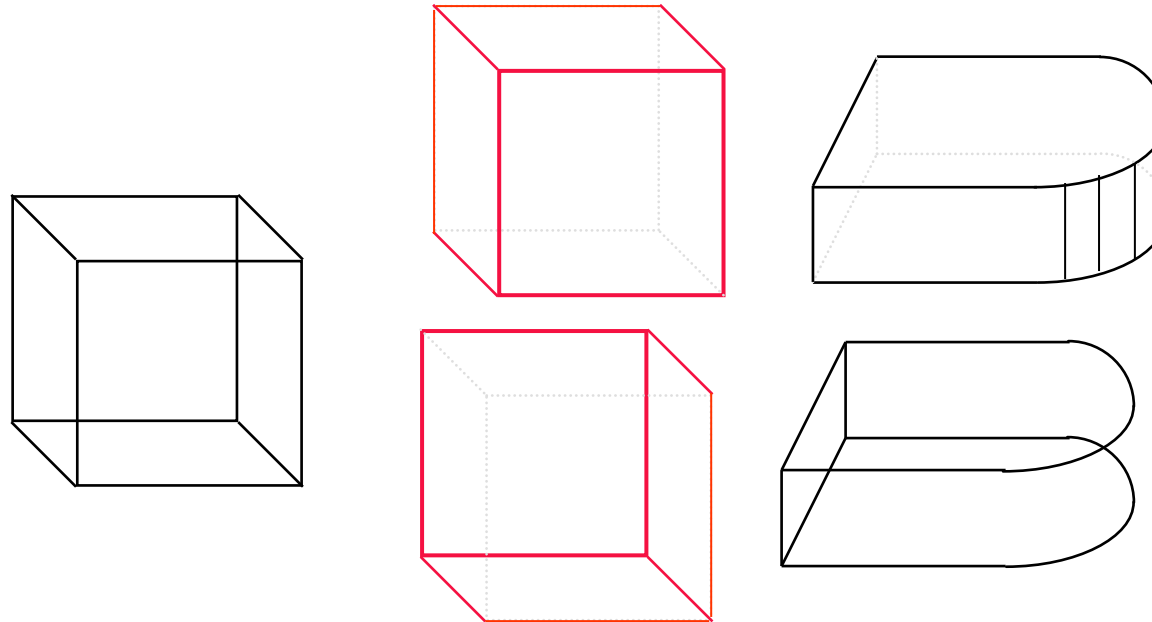


# Modelado plano de superficies

- ▶ Desventajas:
  - ▶ representación ambigua (una misma representación para diferentes objetos)
  - ▶ falta de coherencia visual (algoritmos de eliminación de líneas ocultas, algoritmos de inclusión de aristas)



***Ambigüedad***



***Falta de coherencia visual***

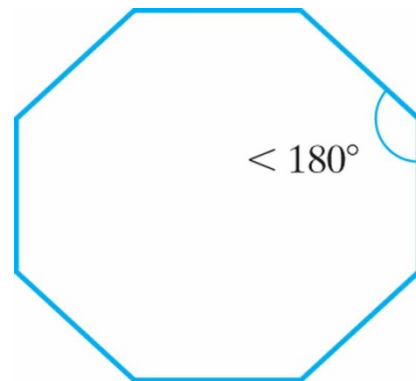
# Modelado plano de superficies

---

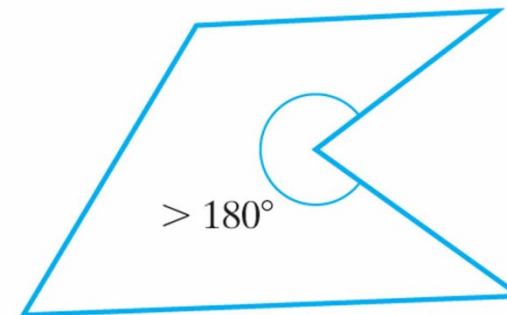
- ▶ Un polígono es una figura plana definida por un conjunto de 3 o más coordenadas, denominadas vértices
- ▶ Los vértices están conectados en secuencia mediante segmentos de recta denominados aristas
- ▶ Todos los vértices deben estar situados en el mismo plano y las aristas no pueden cruzarse

# Modelado plano de superficies

- ▶ Un ángulo interior de un polígono es el ángulo que forman dos aristas adyacentes en el interior del polígono
- ▶ Si todos los ángulos interiores de un polígono son menores o iguales  $180^\circ$ , el polígono es convexo.
- ▶ También, si unimos cualesquiera dos puntos del interior de un polígono convexo, el segmento de línea que los une, también está en el interior
- ▶ Si un polígono no es convexo, entonces es cóncavo



(a)



(b)



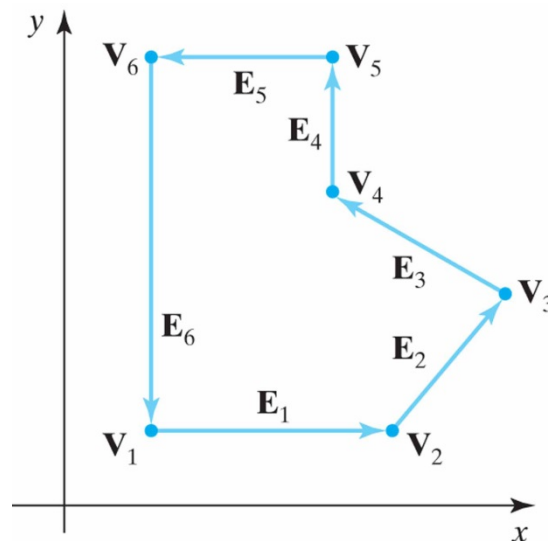
# Modelado plano de superficies

---

- ▶ Se dice que un polígono es degenerado si contiene vértices colineales o cuyas coordenadas sean coincidentes
- ▶ Los vértices colineales generan un segmento de línea recto
- ▶ Los vértices coincidentes generan, solapamiento de aristas o aristas de longitud 0
- ▶ También se utiliza este término para polígonos con menos de tres vértices
- ▶ Es importante en una aplicación gráfica, tener mecanismos para evitar los polígonos degenerados
- ▶ Los polígonos cóncavos también generan problemas, por lo que es común dividirlos en un conjunto de polígonos convexos

# Modelado plano de superficies

- ▶ Determinar si un polígono es cóncavo:
  - ▶ Calcular el vector en la dirección de cada arista
  - ▶ Calcular el producto vectorial de cada par de aristas adyacentes
  - ▶ Si la coordenada zeta de todos ellos es positiva o negativa, el polígono es convexo, en otro caso es cóncavo



$$(\mathbf{E}_1 \times \mathbf{E}_2)_z > 0$$

$$(\mathbf{E}_2 \times \mathbf{E}_3)_z > 0$$

$$(\mathbf{E}_3 \times \mathbf{E}_4)_z < 0$$

$$(\mathbf{E}_4 \times \mathbf{E}_5)_z > 0$$

$$(\mathbf{E}_5 \times \mathbf{E}_6)_z > 0$$

$$(\mathbf{E}_6 \times \mathbf{E}_1)_z > 0$$

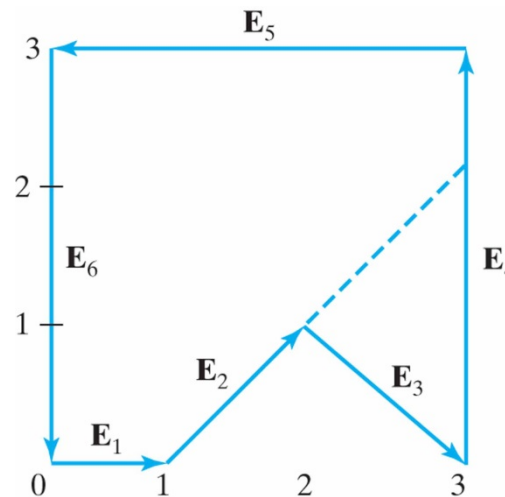
# Modelado plano de superficies

---

- ▶ Descomponer un polígono cóncavo en polígonos convexos
  - ▶ Suponemos que todos los vértices están en el plano XY
  - ▶ Suponemos que el polígono no es degenerado
  - ▶ Calculamos los vectores de cada arista:  $E_k = V_{k+1} - V_k$
  - ▶ Calculamos el producto vectorial de cada par de aristas adyacentes
  - ▶ Si algún producto vectorial tiene la componente Z negativa, se calcula la intersección de la primera arista con el polígono

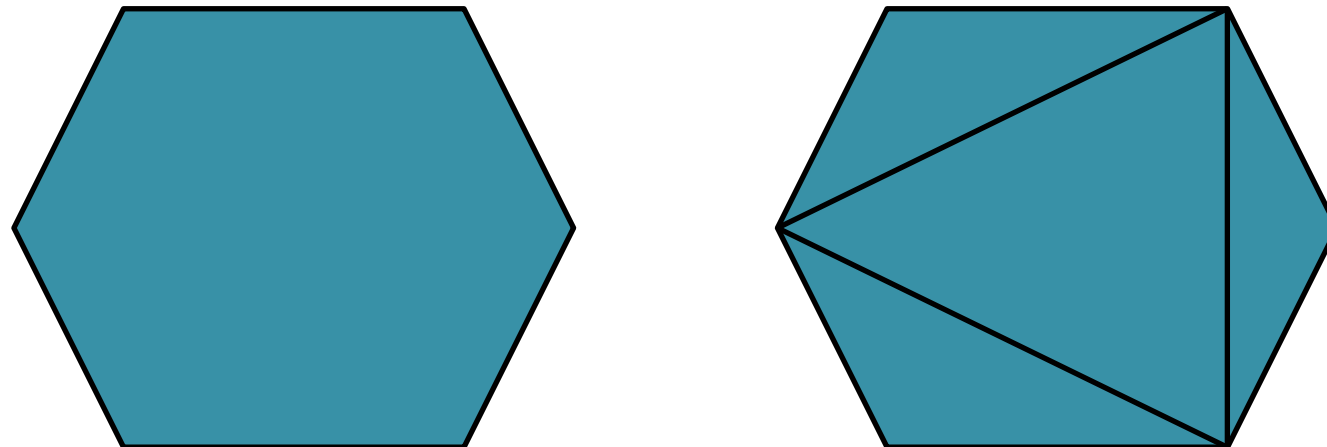
# Modelado plano de superficies

- ▶ Descomponer un polígono cóncavo en polígonos convexos
  - ▶  $E1 (1,0,0)$   $E2 (1,1,0)$   $E3 (1,-1,0)$
  - ▶  $E4 (0,2,0)$   $E5 (-3,0,0)$   $E6 (0,-2,0)$
  - ▶  $E1XE2 = (0,0,1)$   $E2XE3 = (0,0,-2)$   $E3XE4 = (0,0,2)$
  - ▶  $E4XE5 = (0,0,6)$   $E5XE6 = (0,0,6)$   $E6XE1 = (0,0,2)$



# Modelado plano de superficies

- ▶ Triangularizar un polígono convexo
  - ▶ Se seleccionan tres vértices consecutivos y se crea un triángulo
  - ▶ Se elimina de la lista de vértices el del medio
  - ▶ Se vuelve a hacer lo mismo con la lista modificada hasta que sólo queden 3 vértices



# Modelado plano de superficies

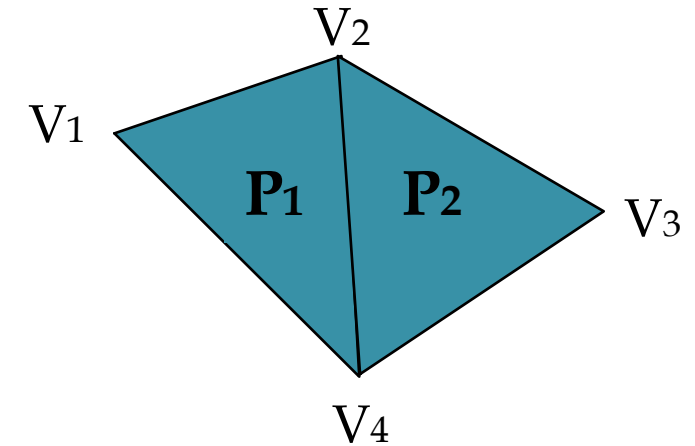
- ▶ *Malla de polígonos*: colección de vértices, aristas y polígonos conectados de forma que cada arista es compartida como máximo por dos polígonos
  - ▶ vértice: punto de coordenadas (x,y,z)
  - ▶ arista: segmento de línea que une dos vértices
  - ▶ polígono: secuencia cerrada de aristas
- ▶ Existen diferentes tipos de representaciones que pueden usarse a la vez en una misma aplicación
  - ▶ Explícita
  - ▶ Punteros a lista de vértices
  - ▶ Punteros a lista de aristas
- ▶ Criterios de evaluación de las representaciones:
  - ▶ tiempo
  - ▶ espacio
  - ▶ información topológica





# Modelado plano de superficies

- ▶ Representación explícita
  - ▶ Cada polígono se representa por una lista de coordenadas de vértices
  - ▶ Los vértices se almacenan en orden (horario o antihorario)
  - ▶ Los vértices compartidos están duplicados
  - ▶ No existe representación explícita de los vértices y aristas compartidas
  - ▶ Se utiliza para almacenar polígonos complejos que no forman partes de mallas
  - ▶ Ventajas:
    - ▶ Representación eficiente para polígonos individuales
  - ▶ Problemas:
    - ▶ Alto coste de almacenamiento
    - ▶ Para mover un vértice es necesario recorrer todos los polígonos
    - ▶ Si se dibujan las aristas, las compartidas se dibujan dos veces

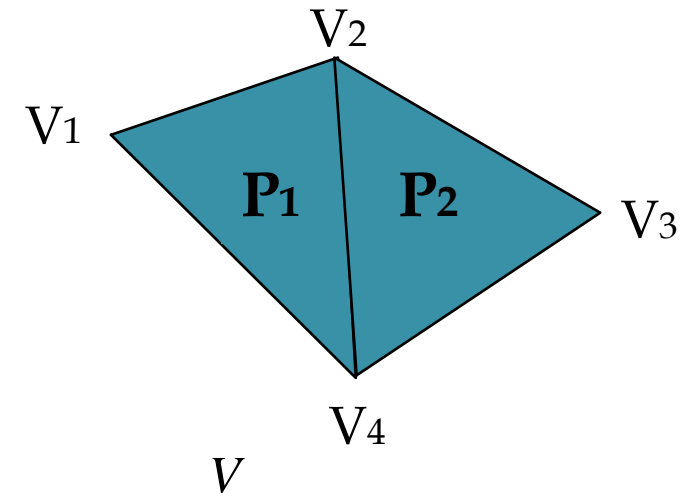


$$P_1 = ((x_1, y_1, z_1), (x_4, y_4, z_4), (x_2, y_2, z_2))$$

$$P_2 = ((x_2, y_2, z_2), (x_4, y_4, z_4), (x_3, y_3, z_3))$$

# Modelado plano de superficies

- ▶ Representación con punteros a lista de vértices
  - ▶ Cada vértice se almacena una sola vez en una lista de vértices
  - ▶ Un polígono se define como una lista de índices (o punteros) a sus vértices en la lista de vértices
  - ▶ Ventajas
    - ▶ Cada vértice se almacena una sola vez
    - ▶ Las coordenadas de los vértices pueden cambiarse fácilmente
  - ▶ Problemas
    - ▶ Difícil encontrar polígonos que compartan una arista
    - ▶ Las aristas compartidas se siguen dibujando dos veces



V

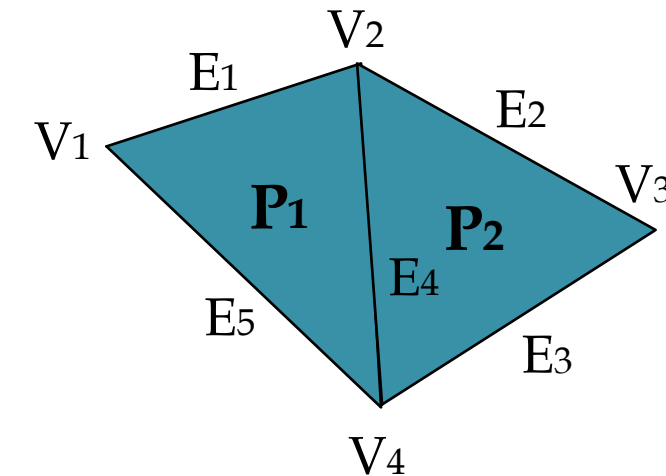
1	$x_1$	$y_1$	$z_1$
2	$x_2$	$y_2$	$z_2$
3	$x_3$	$y_3$	$z_3$
4	$x_4$	$y_4$	$z_4$

$$P_1 = (1, 2, 4)$$

$$P_2 = (4, 2, 3)$$

# Modelado plano de superficies

- ▶ Representación con punteros a lista de aristas
  - ▶ Se mantiene la lista de vértices
  - ▶ Un polígono se representa como una lista de índices a una lista de aristas
  - ▶ Cada arista apunta a dos vértices y a los polígonos a los que pertenece
  - ▶ Ventajas
    - ▶ Cada vértice se almacena una sola vez
    - ▶ Las aristas compartidas se dibujan una sola vez
  - ▶ Problema
    - ▶ Difícil determinar que aristas comparten un vértice (en todas las representaciones)



V	E			
	$v_s$	$v_e$	$p_l$	$p_r$
1	$x_1$	$y_1$	$z_1$	
2	$x_2$	$y_2$	$z_2$	
3	$x_3$	$y_3$	$z_3$	
4	$x_4$	$y_4$	$z_4$	

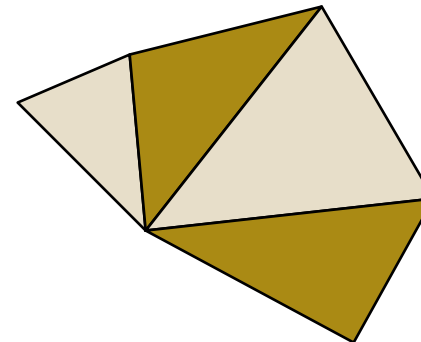
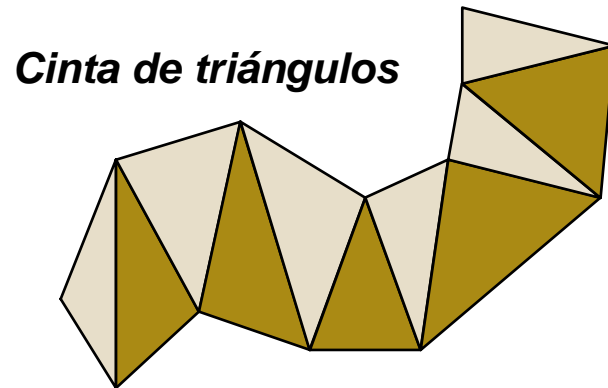
	$v_s$	$v_e$	$p_l$	$p_r$
1	1	2	$\lambda$	1
2	2	3	$\lambda$	2
3	3	4	$\lambda$	2
4	4	2	1	2
5	1	4	1	$\lambda$

$$P_1 = (1, 4, 5)$$

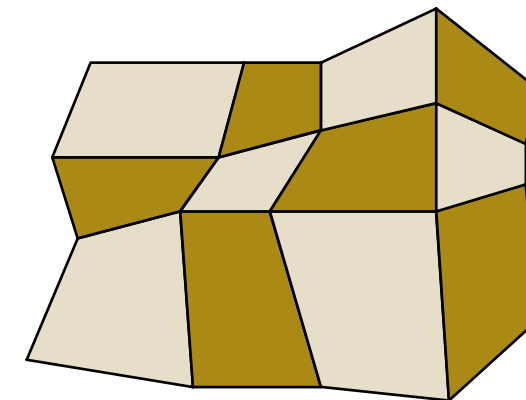
$$P_2 = (2, 3, 4)$$

# Modelado plano de superficies

- ▶ Mallas poligonales
  - ▶ Cinta de triángulos
    - ▶ Para  $n$  vértices, produce  $(n-2)$  triángulos conexos
  - ▶ Abanico de triángulos
    - ▶ Para  $n$  vértices, produce  $(n-2)$  triángulos conexos
  - ▶ Malla de cuadriláteros
    - ▶ Genera una malla de  $(n-1)$  por  $(m-1)$  cuadriláteros para  $n$  por  $m$  vértices

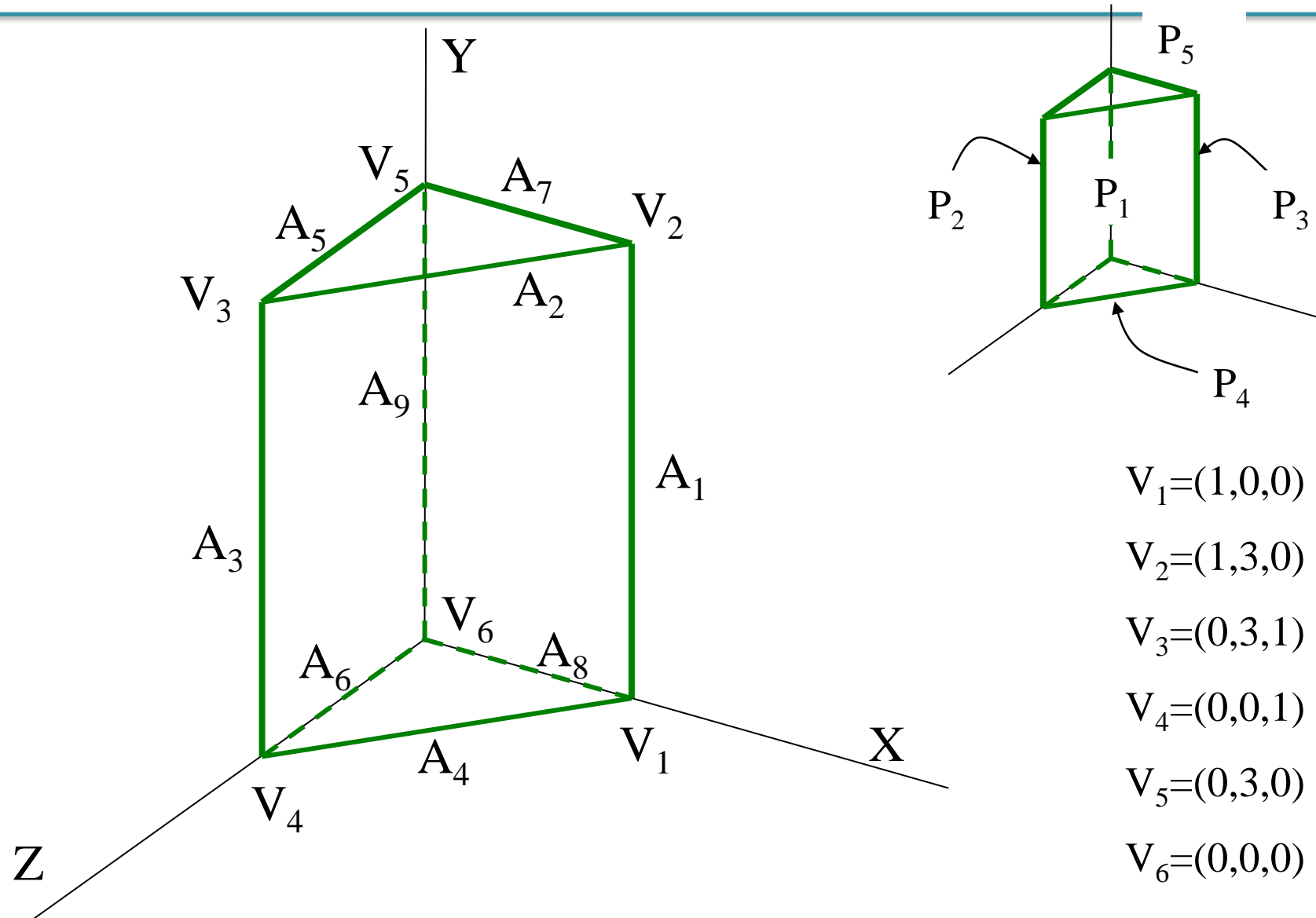


***Abanico de triángulos***

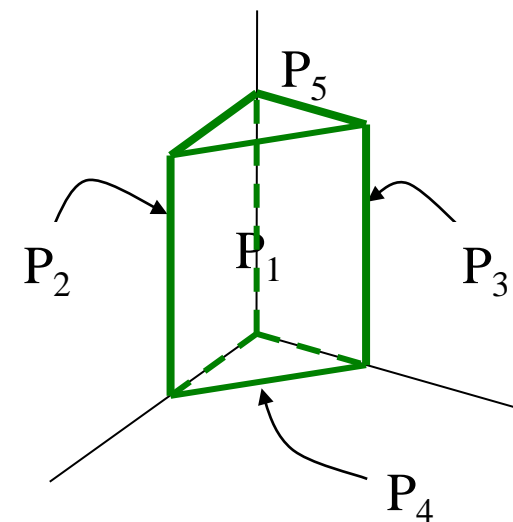
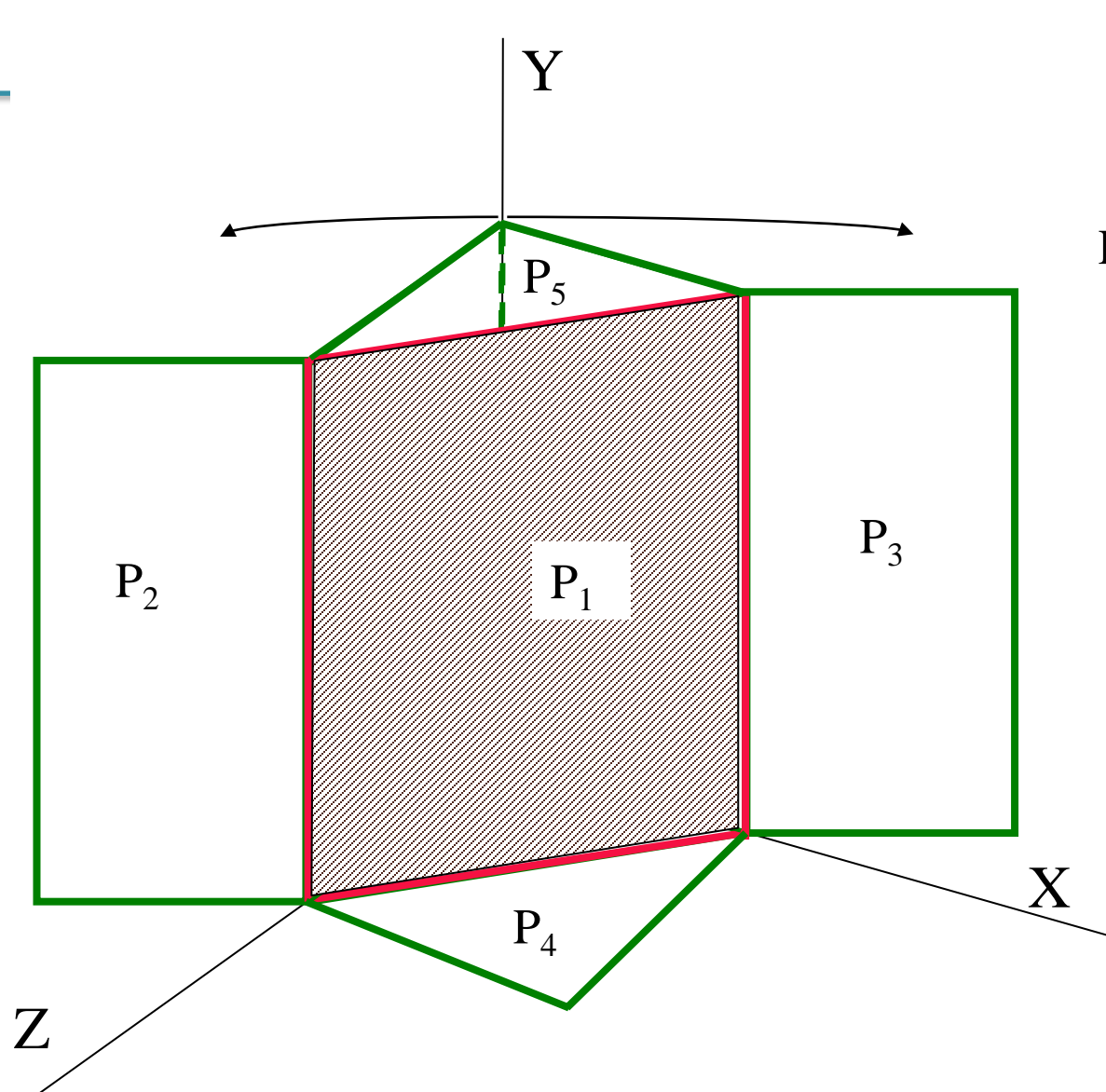


***Malla de cuadriláteros***

# Modelado plano de superficies



# Modelado plano de superficies



$$V_1=(1,0,0)$$

$$V_2=(1,3,0)$$

$$V_3=(0,3,1)$$

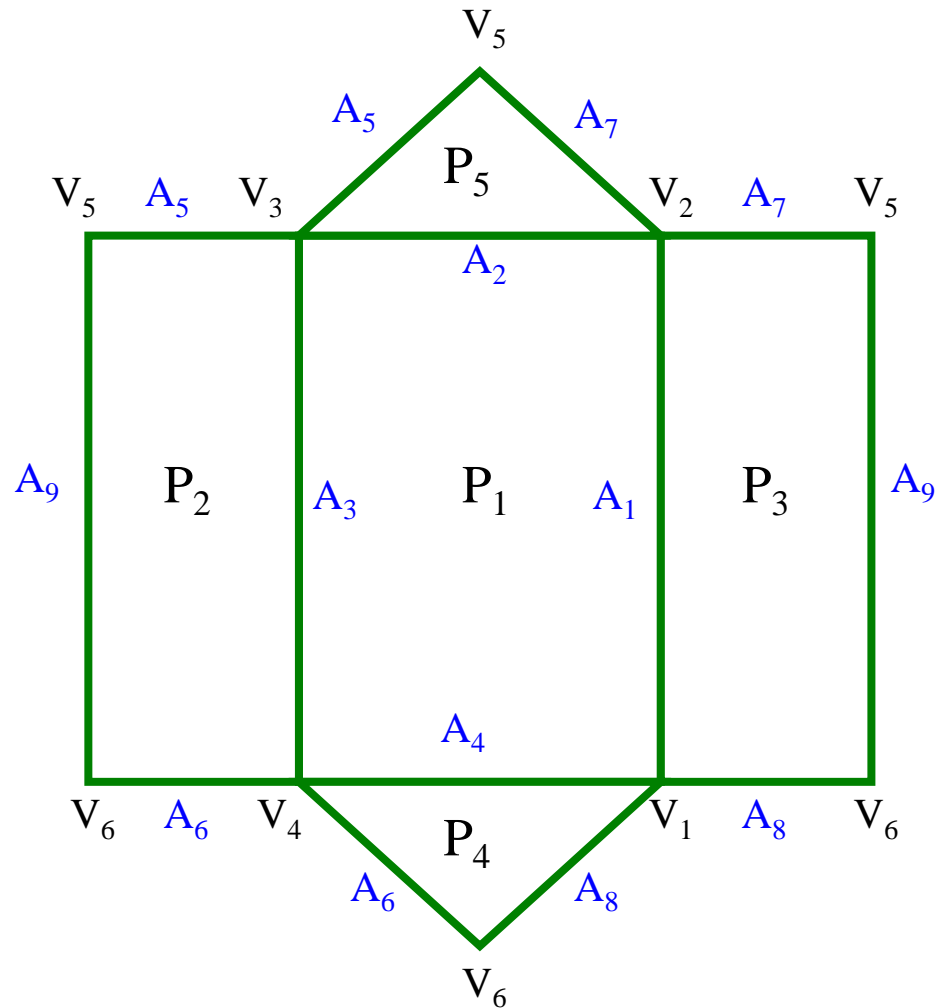
$$V_4=(0,0,1)$$

$$V_5=(0,3,0)$$

$$V_6=(0,0,0)$$



# Modelado plano de superficies



$$\begin{aligned} V_1 &= (1, 0, 0) & V_2 &= (1, 3, 0) \\ V_3 &= (0, 3, 1) & V_4 &= (0, 0, 1) \\ V_5 &= (0, 3, 0) & V_6 &= (0, 0, 0) \end{aligned}$$

# Modelado plano de superficies

---

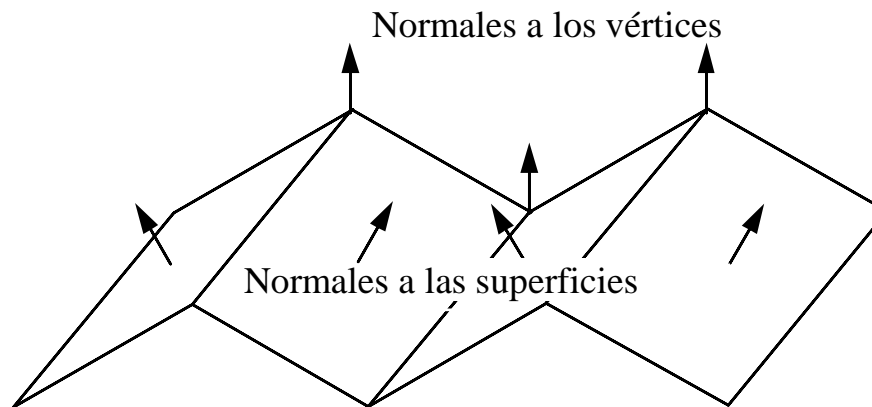
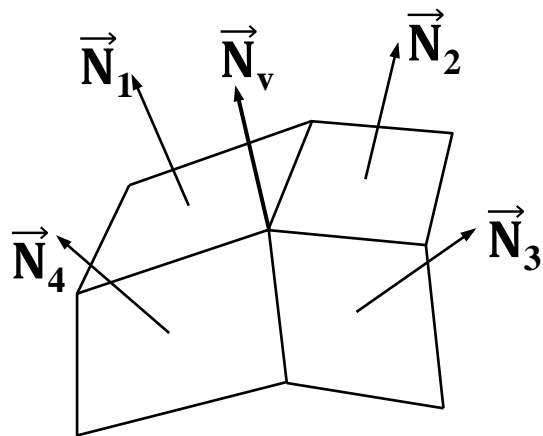
- ▶ En este Screencast de Polimedia se explica el modelado poligonal y cómo se aplica en un problema práctico:

<https://media.upv.es/player/?id=cbc18d80-3c71-11e7-83c8-b99f47788b89>

# Modelos de sombreado

## Aproximación de superficies curvas con mallas de polígonos

- Cálculo aproximado: normal en cada vértice
  - Sumar las normales de las caras adyacentes al vértice
  - Normalizar el vector resultante (las ecuaciones de sombreado usan vectores unitarios)
    - $\vec{N} = \vec{N}_1 + \vec{N}_2 + \vec{N}_3 + \vec{N}_4$
    - $\vec{N}_v = \vec{N} / |\vec{N}|$

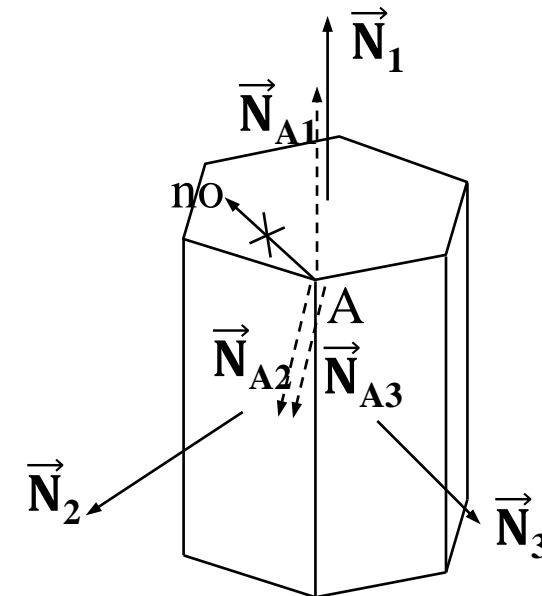
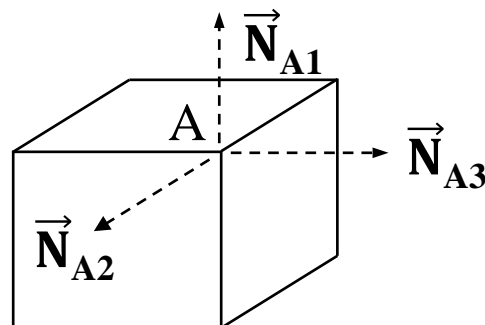


Problema

# Modelos de sombreado

## Aproximación de superficies curvas con mallas de polígonos

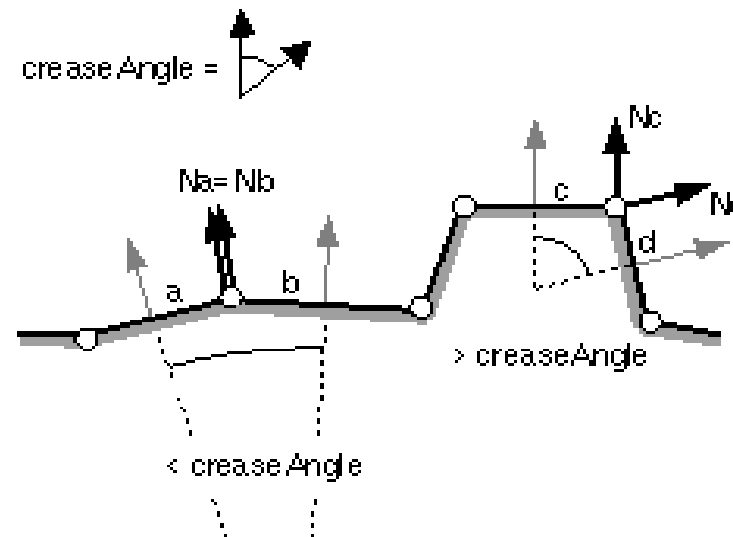
- ▶ Problema
  - ▶ Cilindro: vértice común a cara lateral y a la “tapa”
    - ▶ Un único vértice que pertenece a “superficies” distintas
    - ▶ Distinta normal según que esté el vértice en una cara u otra
  - ▶ El cubo
  - ▶ Es necesario distinguir la normal de:
    - ▶ el vértice A en la cara 1 :  $\vec{N}_{A1}$
    - ▶ el vértice A en la cara 2 :  $\vec{N}_{A2}$
    - ▶ el vértice A en la cara 3 :  $\vec{N}_{A3}$



# Modelos de sombreado

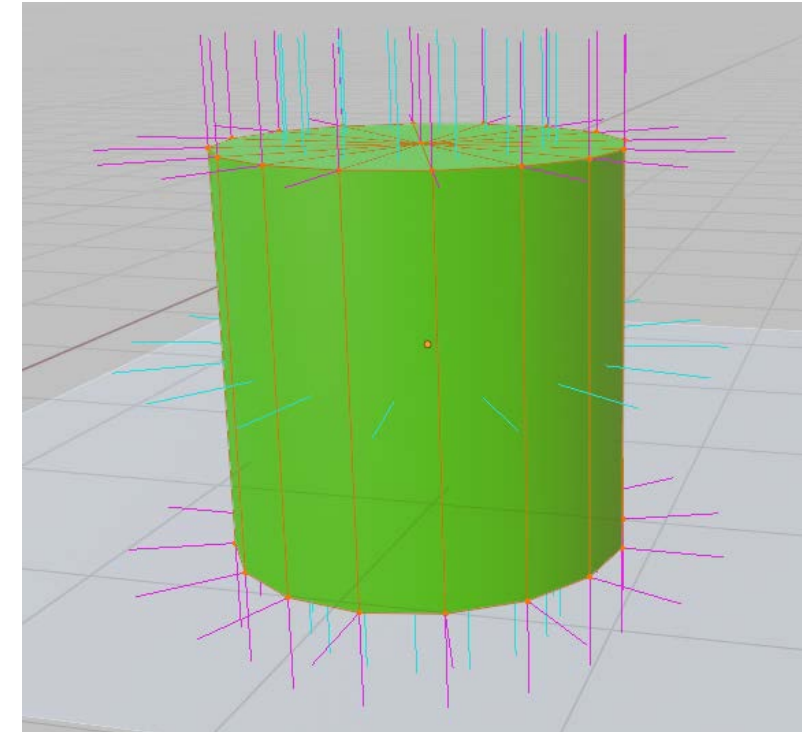
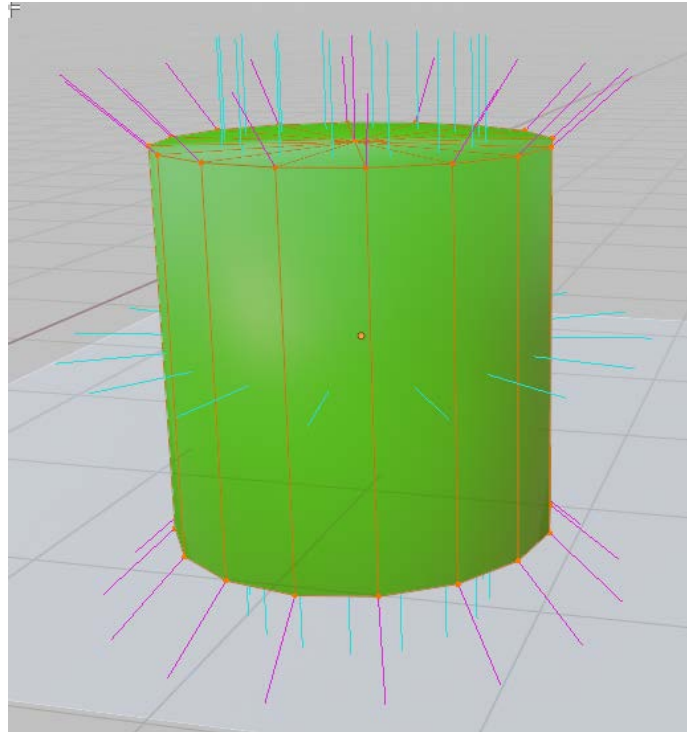
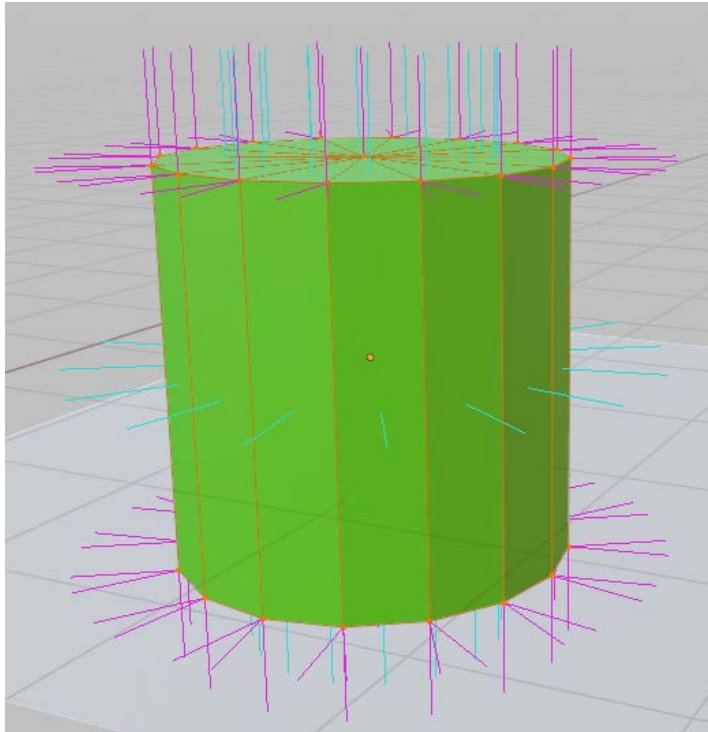
## Aproximación de superficies curvas con mallas de polígonos

- ▶ Solución: “*crease angle*” (VRML)
- ▶ Para calcular la normal en el vértice “K” de la cara “i”
  - ▶ Inicializar  $\vec{N}$  al vector nulo
  - ▶ Recorrer todas las caras j que contienen al vértice K
    - ▶ Si el vector  $\vec{N}_j$  y el vector  $\vec{N}_i$  forman un ángulo inferior al “*crease angle*”
      - ▶  $\vec{N} = \vec{N} + \vec{N}_j$
  - ▶  $\vec{N}_{Ki} = \vec{N} / |\vec{N}|$



# Modelos de sombreado

## Aproximación de superficies curvas con mallas de polígonos

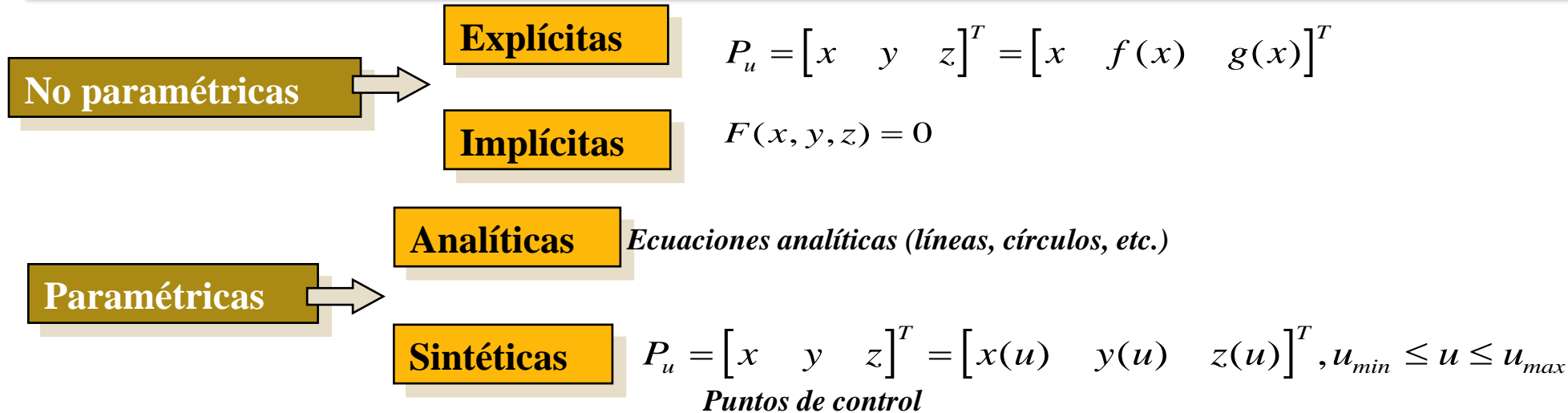


**VÍDEO NORMALES MALLAS DE POLÍGONOS:**

<https://media.upv.es/#/portal/video/621dcoao-1de7-11eb-8c87-e70caadd47c7>



# Modelado curvo de superficies

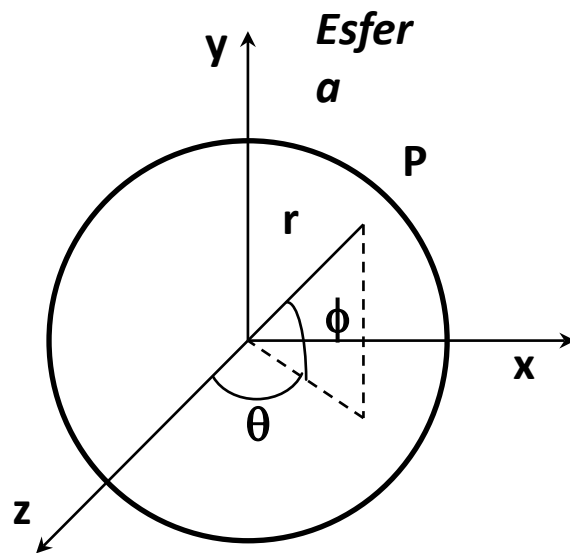


- ▶ **Explícitas**
  - ▶ Incapaces de representar curvas cerradas (círculos) o de valores múltiples (parábolas)
  - ▶ Dificultad en la aplicación de rotaciones y de tangentes verticales
- ▶ **Implícitas**
  - ▶ La ecuación puede tener más de una solución
  - ▶ Problemas con la especificación de continuidad entre curvas
  - ▶ Cuádricas y superficies equipotenciales
- ▶ **Paramétricas**
  - ▶ Fáciles de usar para edición y manipulación (CAD/CAM)
  - ▶ Posibilidad de describir objetos mediante trozos adyacentes
  - ▶ Sustituyen las pendientes (que pueden ser infinitas) por vectores tangentes

# Modelado curvo de superficies

## ► Cuádricas

- Descritas mediante ecuaciones de segundo grado



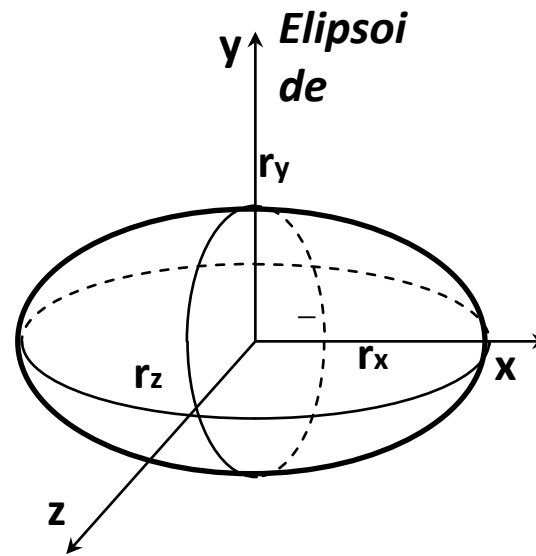
$$x^2 + y^2 + z^2 = r^2$$

Coordenadas Polares:

$$x = r \cdot \cos \phi \cos \theta$$

$$y = r \cdot \sin \phi$$

$$z = r \cdot \cos \phi \sin \theta$$



$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 = 1$$

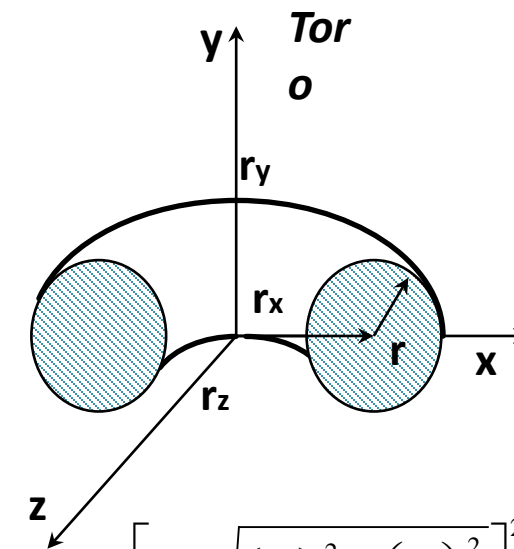
$$-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$$

$$-\pi \leq \theta \leq \pi$$

$$x = r_x \cdot \cos \phi \cos \theta$$

$$y = r_y \cdot \sin \phi$$

$$z = r_z \cdot \cos \phi \sin \theta$$



$$\left[ r - \sqrt{\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2} \right]^2 + \left(\frac{z}{r_z}\right)^2 = 1$$

$$x = r_x \cdot (r + \cos \phi) \cos \theta$$

$$y = r_y \cdot \sin \phi$$

$$z = r_z \cdot (r + \cos \phi) \sin \theta$$

# Modelado curvo de superficies

## ► Supercuádricas

- Generalización de las superficies cuádricas que incorporan parámetros adicionales



### *Super ellipses*

$$\left(\frac{x}{r_x}\right)^{2/s} + \left(\frac{y}{r_y}\right)^{2/s} = 1 \quad \begin{aligned} x &= r \cdot \cos^s \theta \\ y &= r \cdot \sin^s \phi \end{aligned}$$

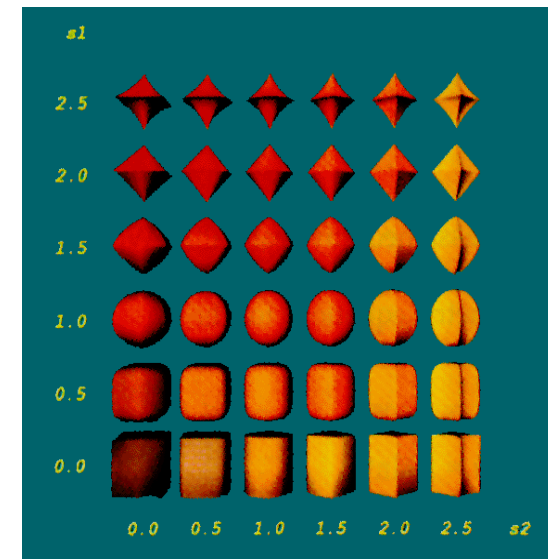
### *Super elipsoides*

$$\left[ \left(\frac{x}{r_x}\right)^{2/s_2} + \left(\frac{y}{r_y}\right)^{2/s_2} \right]^{s_2/s_1} + \left(\frac{z}{r_z}\right)^{2/s_1} = 1$$

$$x = r_x \cdot \cos^{s_1} \phi \cos^{s_2} \theta$$

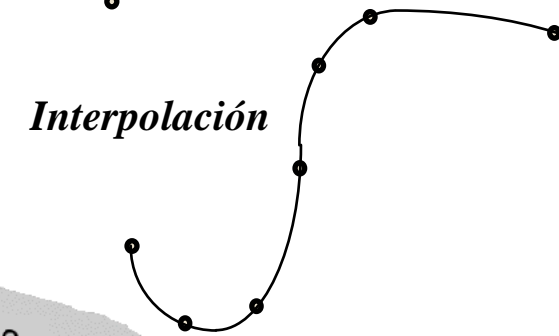
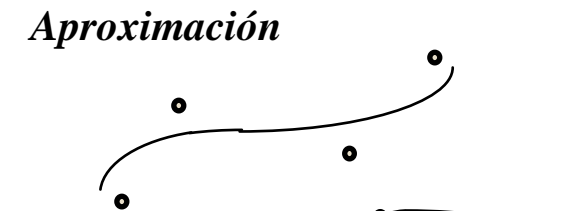
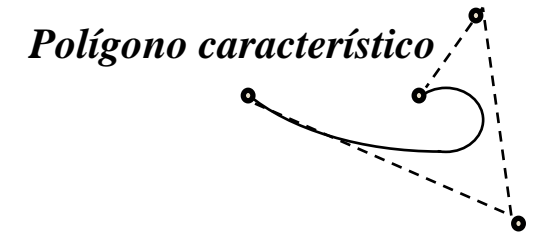
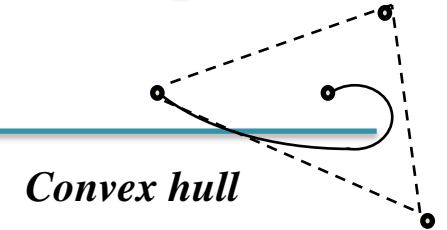
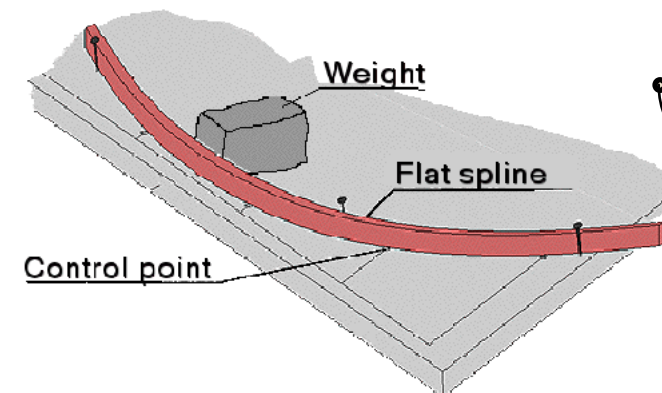
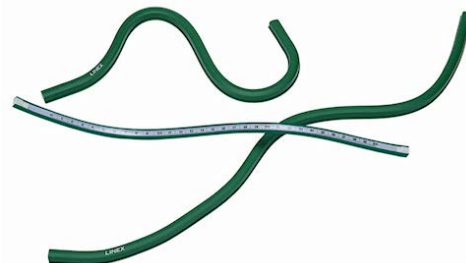
$$y = r_y \cdot \sin^{s_1} \phi$$

$$z = r_z \cdot \cos^{s_1} \phi \sin^{s_2} \theta$$



# Modelado curvo de superficies

- ▶ Splines
  - ▶ *Curvas*. definidas mediante funciones polinómicas cúbicas, cuyas derivadas de primer y segundo orden son continuas
  - ▶ *Superficies*. definidas mediante dos conjuntos ortogonales de curvas
- ▶ Definiciones
  - ▶ *Puntos de control*. conjunto de puntos que indican la forma general de la curva
  - ▶ *Convex hull*. frontera del polígono convexo que encierra el conjunto de puntos de control
  - ▶ *Polígono característico*. Conjunto de segmentos que conectan los puntos de control en orden
- ▶ Tipos de Splines
  - ▶ *Interpolación*. la curva pasa por los puntos de control
  - ▶ *Aproximación*. la curva se ajusta a los puntos de control



# Modelado curvo de superficies

## ► Condiciones de continuidad

- Permiten especificar la forma en que se unen dos segmentos curvos

*Si cada segmento de spline se describe como un conjunto de funciones paramétricas de la forma*

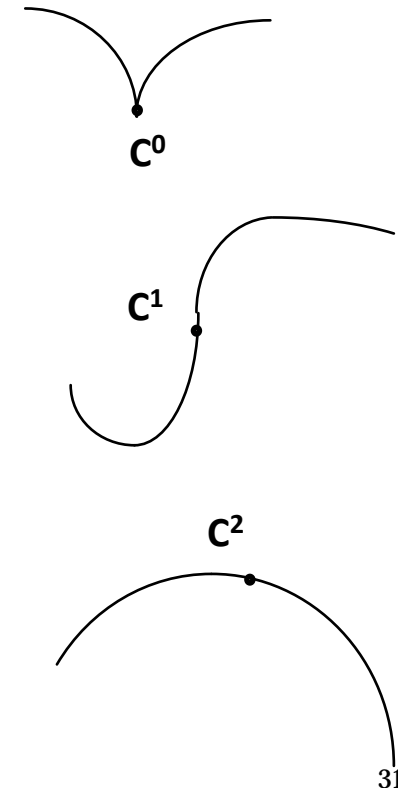
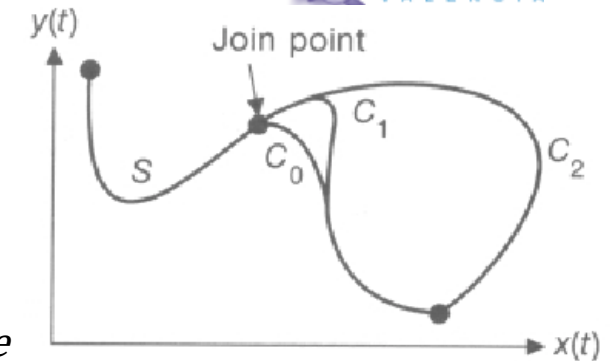
$$x = x(u), \quad y = y(u), \quad z = z(u), \quad u_1 \leq u \leq u_2$$

## ► Continuidad paramétrica

- $C^0$ : Las curvas están unidas. Los valores de  $x, y, z$  evaluados en  $u_2$  para el primer segmento, son iguales a los valores  $x, y, z$  evaluados en  $u_1$  para el segundo segmento
- $C^1$ : Las primeras derivadas (tangentes) de las curvas son iguales en los puntos de unión (digitalización de dibujos, algunas aplicaciones de diseño,...)
- $C^2$ : Las primeras y segundas derivadas de los dos segmentos de curvas son iguales en la intersección. En este caso la variación de las tangentes entre los dos segmentos de curvas es suave (movimientos de cámara, CAD,...)

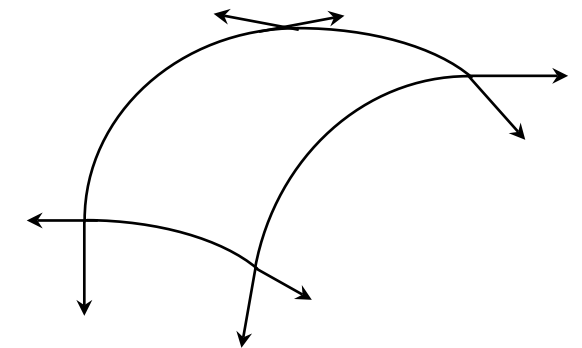
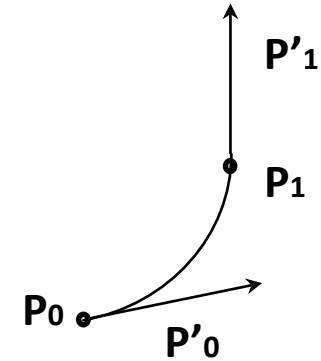
## ► Continuidad geométrica

- $G^0$ : Similar a  $C^0$
- $G^1$ : Las tangentes son proporcionales
- $G^2$ : Las primera y segunda derivadas son proporcionales



# Modelado curvo de superficies

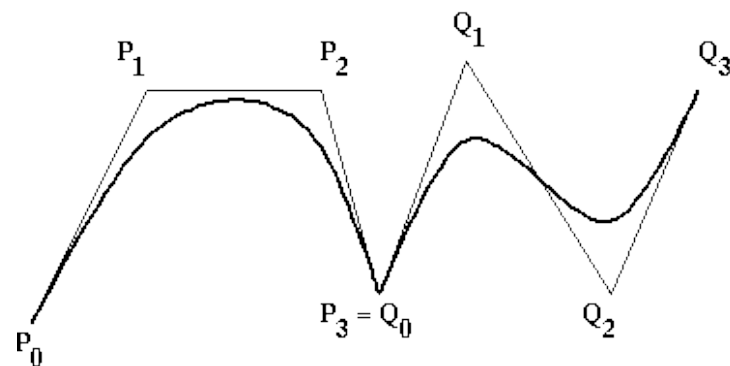
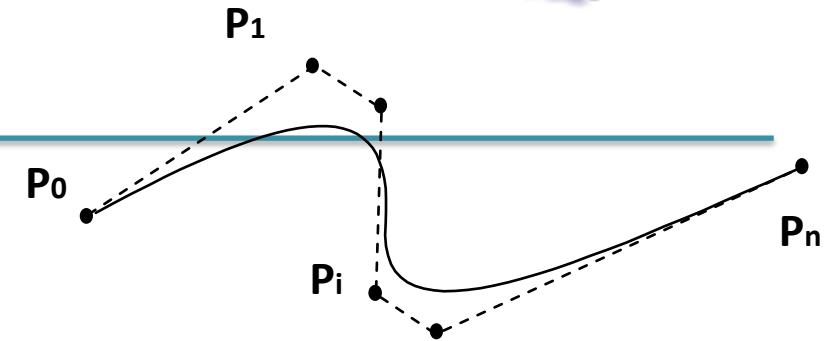
- ▶ Curvas y superficies de Hermite
  - ▶ Representación por interpolación
  - ▶ Las curvas
    - ▶ Suelen utilizarse para la especificación de trayectorias
    - ▶ Se definen dando los puntos inicial y final y sus respectivas tangentes
    - ▶ Son curvas cúbicas
    - ▶ Si se quiere interpolar un conjunto  $n$  de puntos con sus dos vectores tangentes finales  $P_0'$  y  $P'_{n-1}$  se imponen unas condiciones de continuidad y se plantea un sistema de ecuaciones para obtener las tangentes intermedias
  - ▶ Las superficies
    - ▶ Se define con los puntos y sus tangentes en cada una de las direcciones paramétricas



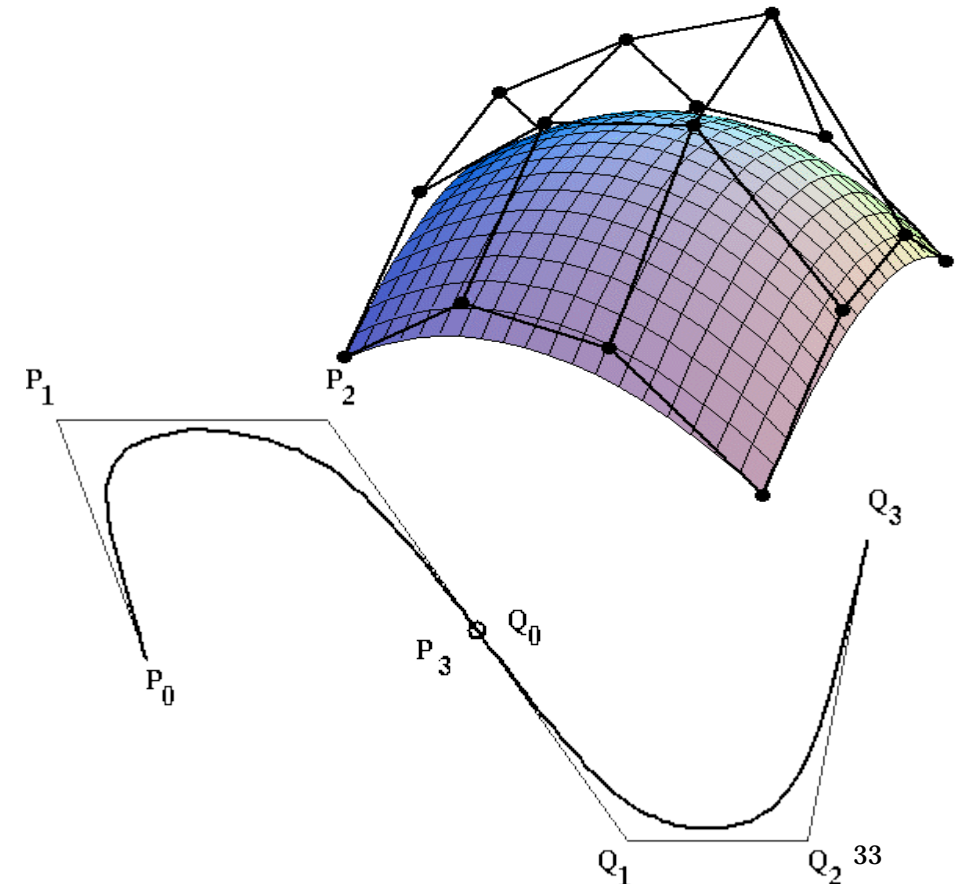


# Modelado curvo de superficies

- ▶ Curvas y superficies de Bezier
  - ▶ Aproximan cualquier número de puntos de control
  - ▶ Una curva con  $n+1$  puntos tiene grado  $n$
  - ▶ Con 4 puntos tenemos curvas cúbicas
  - ▶ Continuidad de tramos cúbicos:  $C0$ ,  $C1$  y  $C2$
  - ▶ Manipulación de la curva
    - ▶ Movimiento de puntos
    - ▶ Acumulación de puntos
    - ▶ Control global

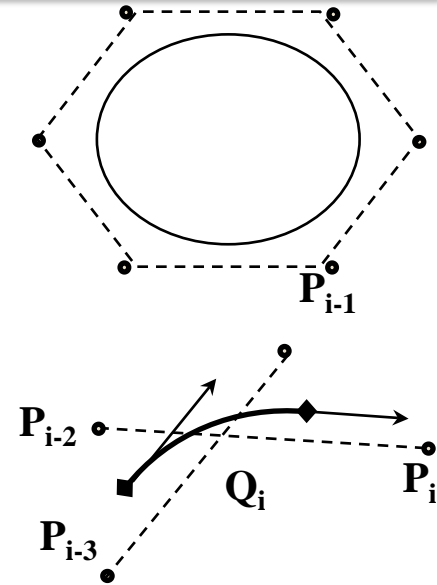


Modelado Geométrico



# Modelado curvo de superficies

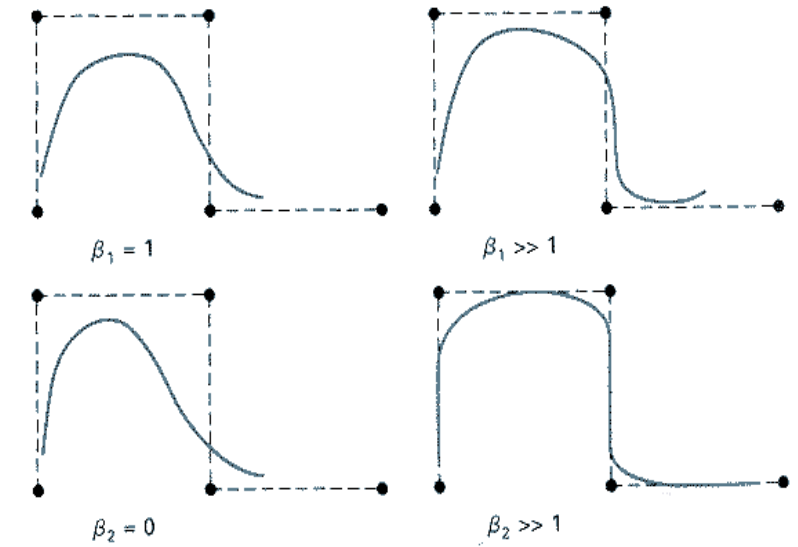
- ▶ Curvas y superficies B-Splines
  - ▶ Spline: banda flexible que produce una curva suave a través de un conjunto de puntos
  - ▶ Curva con secciones polinómicas
  - ▶ Diseñar curvas y superficies
  - ▶ El grado es independiente del número de puntos de control
  - ▶ Permiten control local



# Modelado curvo de superficies

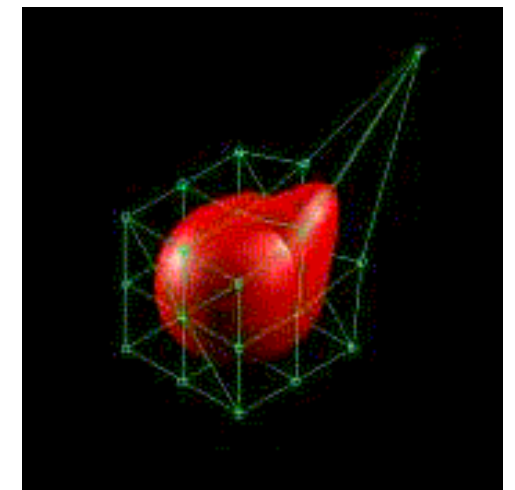
## ► Beta-Splines

- Generalización de las B-Splines en las que se imponen condiciones de continuidad geométrica sobre la primera y segunda derivadas paramétricas
- Los parámetros de continuidad se denominan  $\beta$  *parámetros*:  $\beta_1$  (*Bias*),  $\beta_2$  (*Tensión*). Si  $\beta_1=1$  y  $\beta_2=0$  entonces Beta-Splines=B-Splines



## ► Splines racionales (NURBS)

- Similares a las B-Spline pero con pesos asociados a cada uno de los puntos de control
- Los pesos permiten ajustar más o menos la curva al punto de control  $P_i$ . Cuando todos son iguales a 1 tenemos una B-Spline
- Representación exacta para las cuádricas (cónicas)



# Modelado curvo de superficies

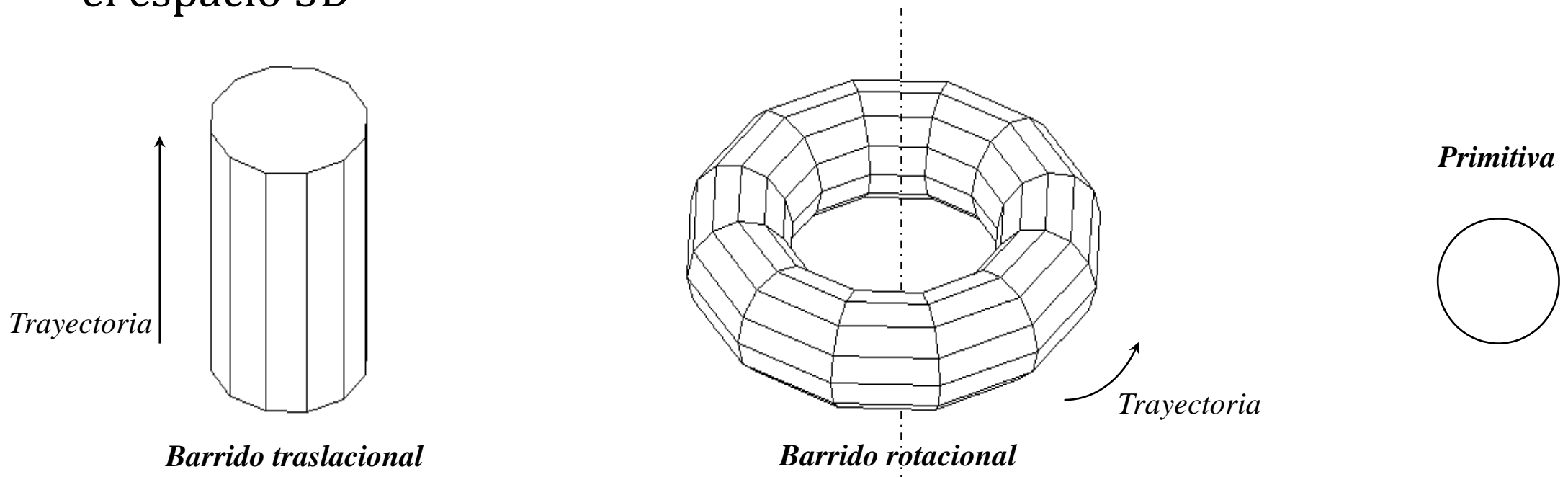
---

- ▶ Polimedia sobre Curvas y Extrusión:
  - ▶ Curvas: edición
  - ▶ Herramientas
  - ▶ Utilizar una curva como trayectoria de extrusión
- ▶ <https://media.upv.es/player/?id=31a4b440-3c74-11e7-83c8-b99f47788b89>

# Modelado de sólidos

## ► Modelo de barrido

- Representación útil para la construcción de objetos que tienen simetrías rotacionales, traslacionales o de otros tipos
- Los objetos se definen mediante una primitiva en 2D y una trayectoria en el espacio 3D



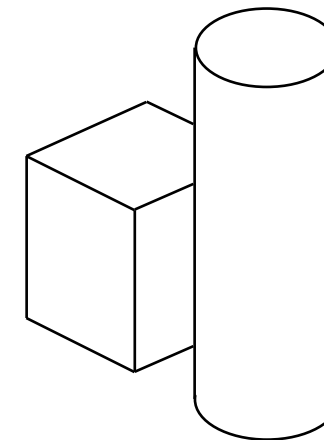
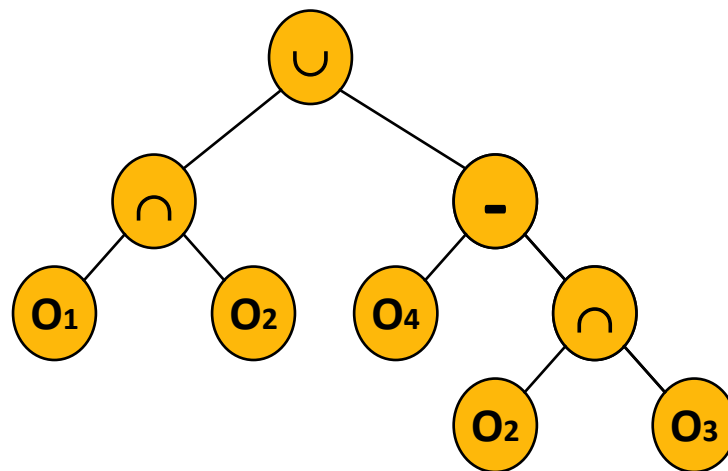
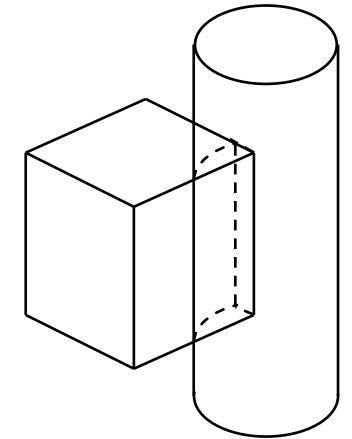
# Modelado de sólidos

---

- ▶ Polimedia sobre Objetos de revolución:
  - ▶ ¿Qué es un objeto de revolución?
  - ▶ Creación mediante cilindros
  - ▶ Creación mediante extrusión de círculos
  - ▶ Creación mediante giro de una curva
- ▶ <https://media.upv.es/player/?id=40f99ce0-20c7-11e6-9e98-d90fe8da13de>

# Modelado de sólidos

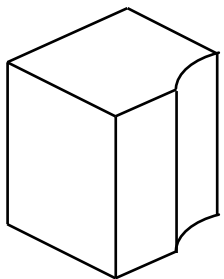
- ▶ Geometría sólido constructiva (CSG)
  - ▶ Un nuevo sólido se obtiene aplicando operaciones de unión, intersección y diferencia sobre dos sólidos iniciales
  - ▶ Existe un conjunto inicial de primitivas (bloques, conos, cilindros, esferas, superficies de revolución, ...)
  - ▶ Los objetos diseñados con este método se representan con un árbol binario



*Unión*



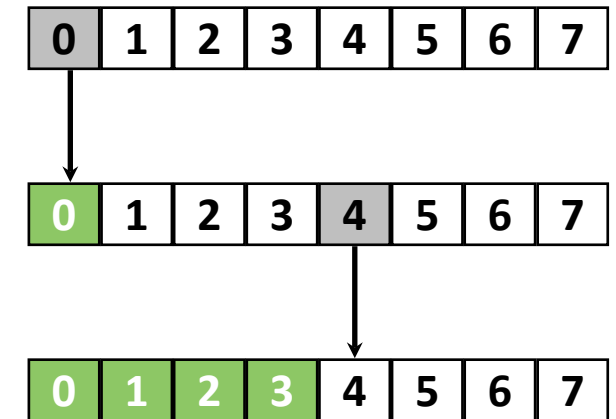
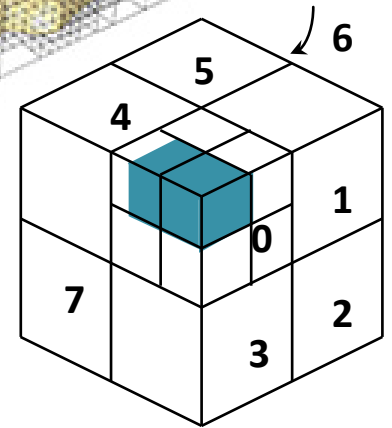
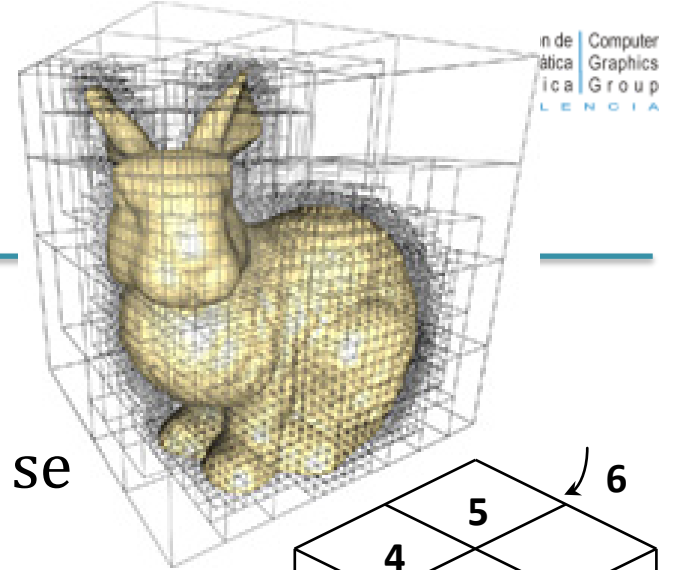
*Intersección*



*Diferencia*

# Modelado de sólidos

- ▶ Enumeración espacial (*Octrees*)
  - ▶ Estructura de árbol jerárquico, en la que cada nodo se corresponde con una región del espacio 3D
  - ▶ Las regiones del espacio 3D se dividen en octantes (cubos), se almacenan 8 elementos en cada nodo del árbol.
  - ▶ Los nodos pueden ser de 3 tipos: blancos (fuera del objeto), verdes (dentro del objeto) o grises (ni dentro ni fuera)
  - ▶ Los elementos individuales del espacio 3D se denominan *voxels*

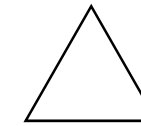




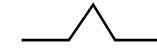
# Otros modelos

## ► Fractales

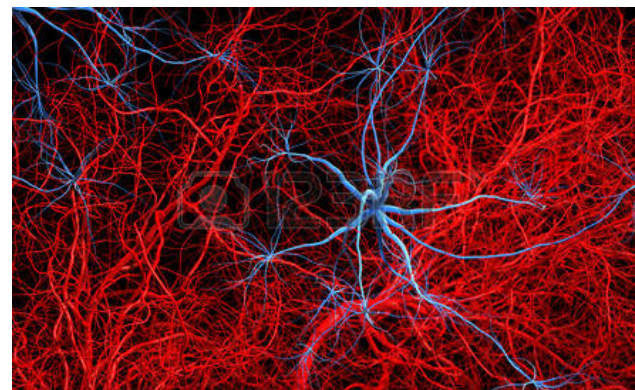
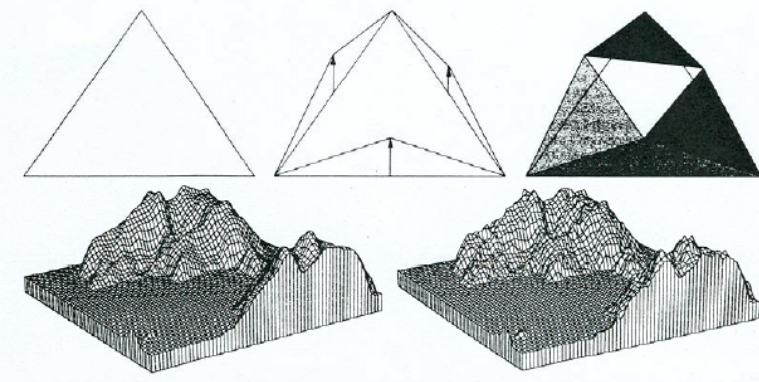
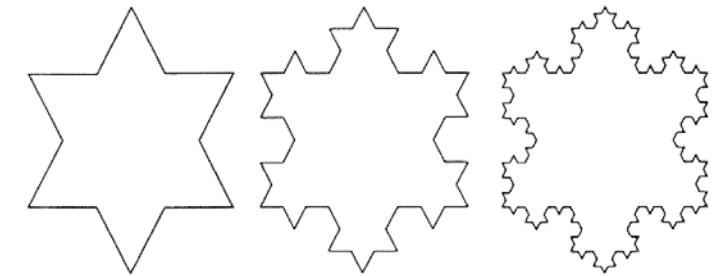
- Permiten representar objetos que por su complejidad no se ajustan a la geometría euclídea (montañas, nubes, plantas, etc)
- Se utilizan procedimientos en lugar de ecuaciones para modelar los objetos



iniciador



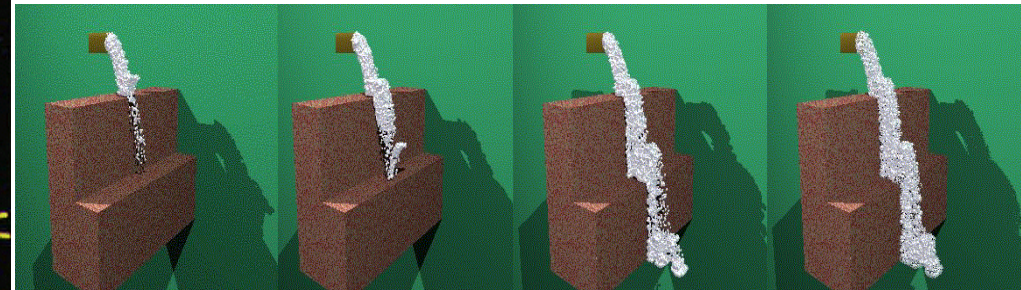
generador



# Otros modelos

## ► Sistemas de partículas

- Técnica de modelado ideal para objetos o fenómenos dinámicos (nubes, humo, fuego, explosiones, ...)
- Las partículas son los elementos básicos del sistema pueden tener distintas formas (esferas, elipsoides, blobs, ...) y un comportamiento que determina la evolución del sistema (tiempo de vida, aspecto de las partículas, trayectoria, ...) y que puede estar influido por fuerzas específicas (gravedad, campos magnéticos, ...)

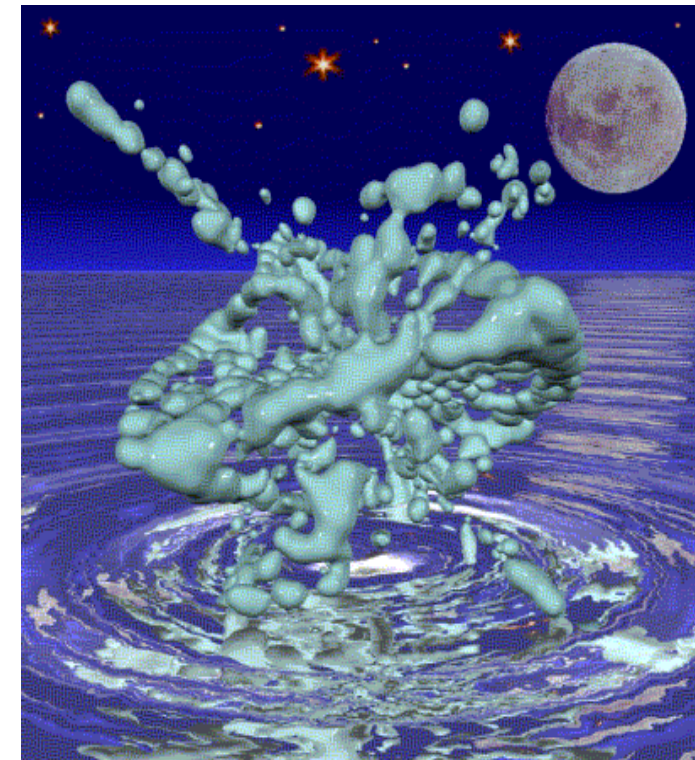
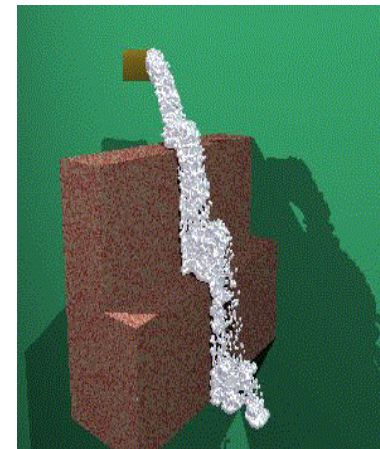
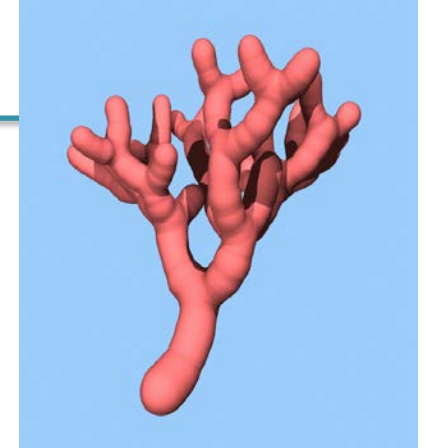




# Otros modelos

## ▶ Blobs

- ▶ Esferas flexibles entre las que existe un campo de fuerza que las repele o atrae, de forma que cuando entran en contacto se funden en un único cuerpo con conexiones suaves
- ▶ Otras denominaciones
  - ▶ Metaballs
  - ▶ Isosuperficies
  - ▶ Objetos blandos
- ▶ Aplicaciones
  - ▶ Modelado de moléculas
  - ▶ Representación de fluidos



# Bibliografía

---

- ▶ D. Hearn, M. Baker. Computer Graphics with OpenGL. Pearson Prentice Hall, 4ª edición.
  - ▶ Capítulos 4, 13, 14 y 15