Introdução à Computação Gráfica
Modelação

Adaptação: João Paulo Pereira

António Costa

Autoria: Claudio Esperança

Paulo Roma Cavalcanti

História

- Modelação por malha de arame (*wireframes*)
  - Representa os objectos por arestas e pontos sobre a sua superfície
  - Gera modelos ambíguos
- Modelação por superfícies (década de 60)
  - Fornece a descrição matemática das superfícies que delimitam o objecto
  - Poucos testes de integridade do modelo

História

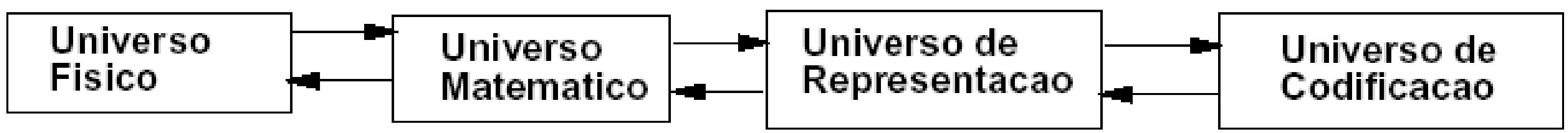
- Modelação de Sólidos (década de 70)
  - Implícita ou explicitamente contém informações sobre o fecho e conectividade dos objectos
  - Garante a realização física
  - Sistemas CAD-CAM utilizados pela indústria

Estado da Arte

- Modelação de dimensão mista ou *non-manifold* (década de 80)
  - Permite representar objectos com estruturas internas ou com elementos pendentes de dimensão inferior
  - Sólido delimitado por superfícies não necessariamente planas localmente
  - Ex.: ACIS (Spatial Technology) – *AutoCad*, etc.

Paradigmas de Abstracção

- A necessidade de paradigmas (Ari Requicha).



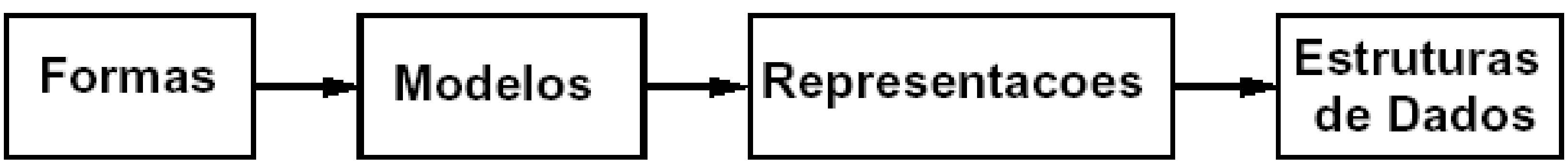
- Paradigma dos universos
  - Físico *F*
  - Matemático *M*
  - Representação *R*
  - Implementação *I*

Problemas da Área

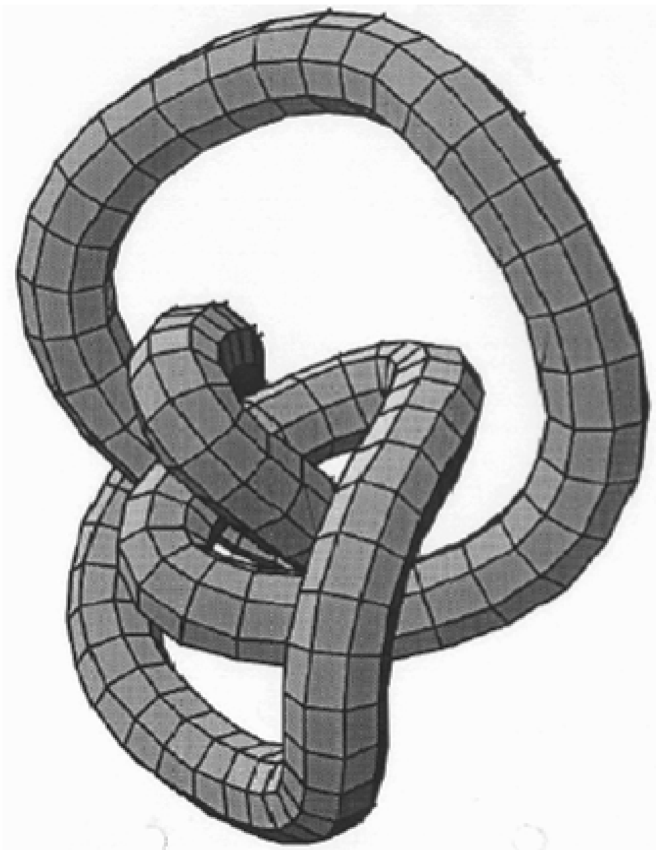
- Estudar fenómenos em F
- Definir os modelos
- Estudar as relações entre R e M
- Definir representações de modelos em M
- Estudar conversões entre representações
- Definir métodos de implementação
- Comparar estratégias em I

Esquemas de Representação

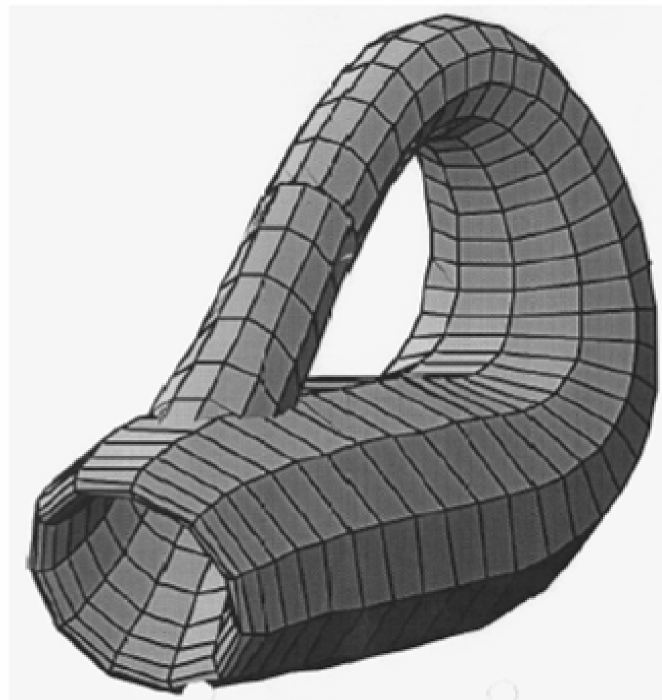
- Objectos do universo físico: “sólidos”
  - O que é um sólido?
- Objectos do universo matemático vêm da:
  - Geometria diferencial
  - Topologia diferencial



Geometria pode ser complicada



Nó



Garrafa de Klein
(não orientável)

Descrição de Sólidos

- Assumir que um sólido é um conjunto tridimensional de pontos
- Conjuntos de pontos podem ser descritos
  - Pelas suas fronteiras
  - Por campos escalares
    - Definidos por equações
    - Amostrados
- Originam três tipos de representação:
  - Por fronteira (B-rep – *Boundary Representation*)
  - Operações de conjuntos (CSG – *Constructive Solid Geometry*)
  - Por enumeração do espaço em células (BSP-trees, Octrees, etc.)

Representação por Fronteira

- Sólido definido indirectamente através da superfície que o delimita
  - compacta (fechada e limitada)
  - sem bordo
- Superfícies são descritas parametricamente por **parametrização**:

$$\varphi : U \subset \Re^2 \to \Re^3$$

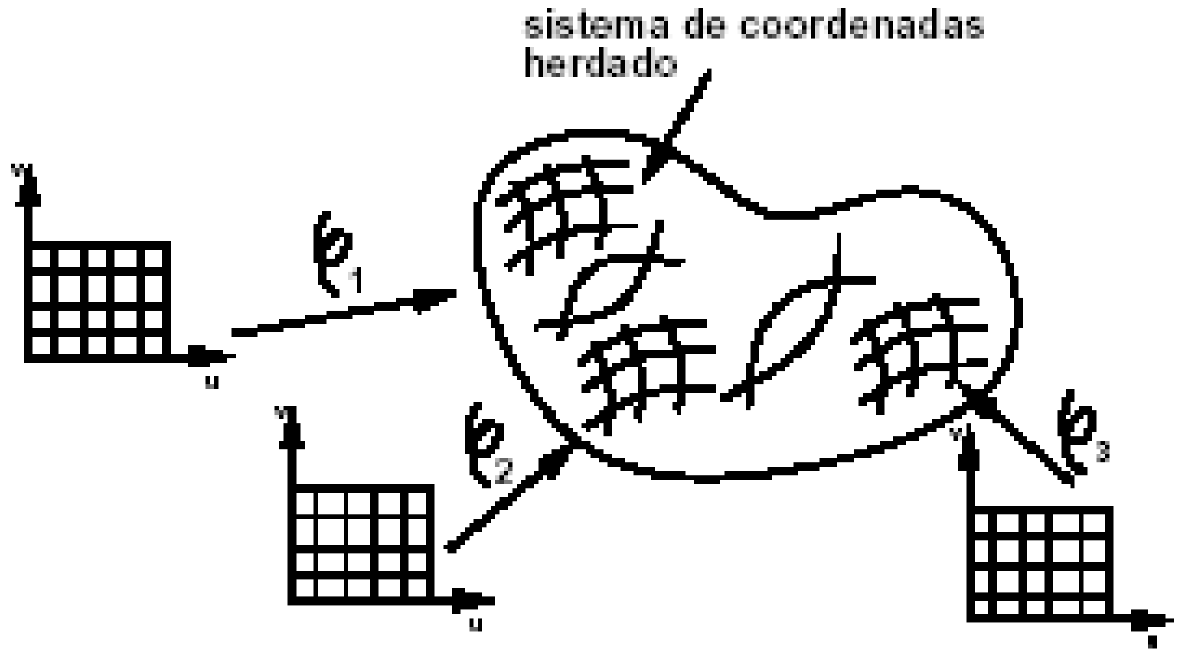
Parametrização

- Estabelece um sistema de coordenadas sobre a superfície <u>herdado</u> de um sistema de coordenadas no plano

$$\varphi(u,v) = \left(\varphi_x(u,v), \varphi_y(u,v), \varphi_z(u,v)\right)^T = (x, y, z)^T$$

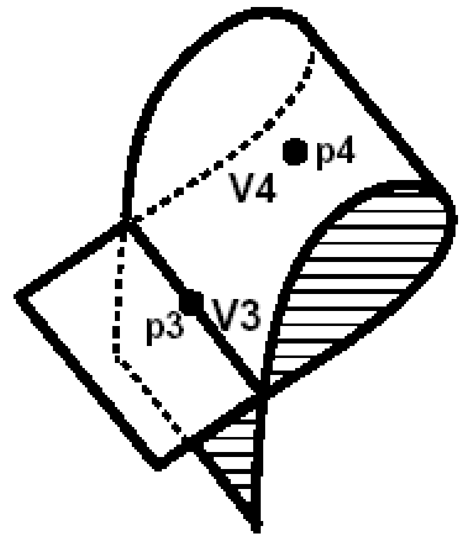
- Em geral, não é possível cobrir (descrever) toda a superfície com uma única parametrização
  - Usam-se várias parametrizações que formam um **Atlas**

Parametrização de uma Superfície



Parametrizações Válidas

- Sólido deve estar bem definido
  - Superfície sem auto-intersecção
  - Vector normal não se anula sobre a superfície
  - Normal é usada para determinar o interior e o exterior do sólido



$$N = \left(\frac{\partial \varphi}{\partial u} \times \frac{\partial \varphi}{\partial v} \right)$$

Exemplo

- Parametrização da esfera de raio unitário, centrada na origem

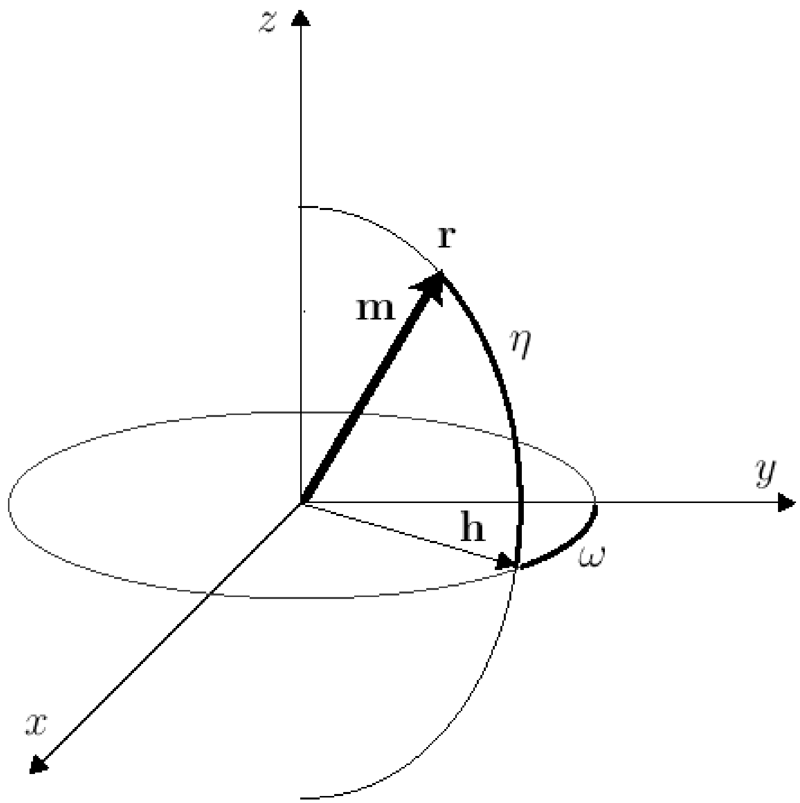
$$f(\omega,\eta)=\begin{bmatrix}\sin(\omega)\cos(\eta)\\ \cos(\omega)\cos(\eta)\\ \sin(\eta)\end{bmatrix}$$

$$N=\frac{\partial f}{\partial \omega}\times\frac{\partial f}{\partial \eta}=\begin{bmatrix}\cos(\omega)\cos(\eta)\\ -\sin(\omega)\cos(\eta)\\ 0\end{bmatrix}\times\begin{bmatrix}-\sin(\omega)\sin(\eta)\\ -\cos(\omega)\sin(\eta)\\ \cos(\eta)\end{bmatrix}$$

- Se $\eta = -\pi/2$ ou $\eta = \pi/2$ a normal não está definida nos pólos por esta parametrização

Domínio do Exemplo Anterior

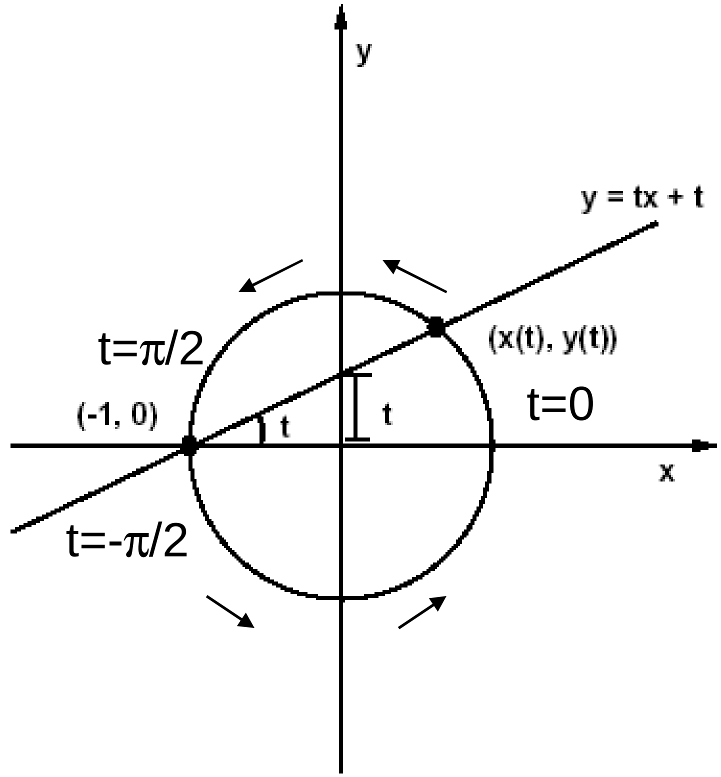
- Todas as parametrizações da esfera deixam pelo menos um ponto de fora
- É impossível parametrizar continuamente a esfera no plano sem retirar pelo menos um ponto



$$U = \left\{ (\omega, \eta) \in \Re^2 : 0 \leq \omega < 2\pi; -\frac{\pi}{2} < \eta < \frac{\pi}{2} \right\}$$

Parametrização do Círculo

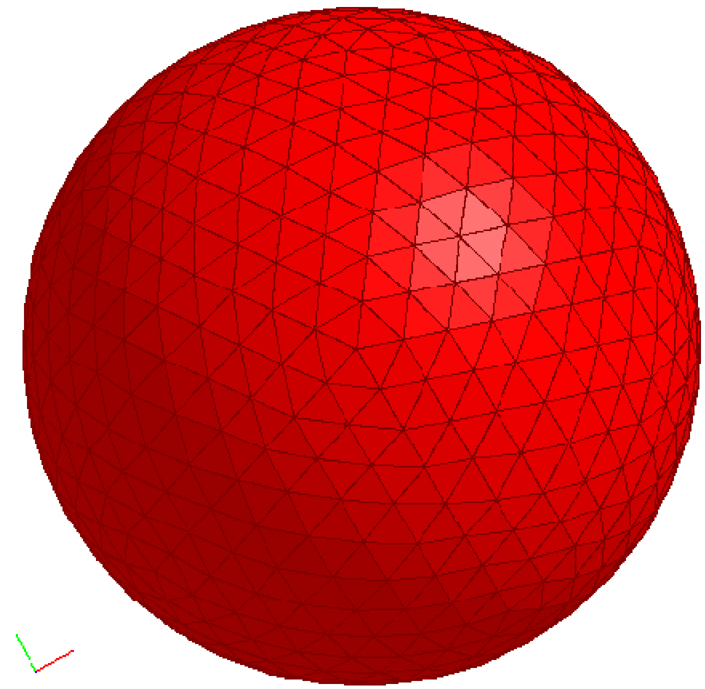
- Forma implícita
  - y = tx + t
  - $x^2 + y^2 = 1$
- Resolvendo esse sistema chega-se a uma parametrização alternativa do círculo



$$x(t) = \frac{1-t^2}{1+t^2}; y(t) = \frac{2t}{1+t^2}; t \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$$

Representação Linear por Partes

- Superfície parametrizada com geometria complexa pode ser <u>aproximada</u> por uma superfície linear por partes
- Pode-se particionar o domínio da parametrização por um conjunto de polígonos
  - Cada vértice no domínio poligonal é levado para a superfície pela parametrização
  - Em seguida é ligado aos vértices adjacentes mantendo as conectividades do domínio



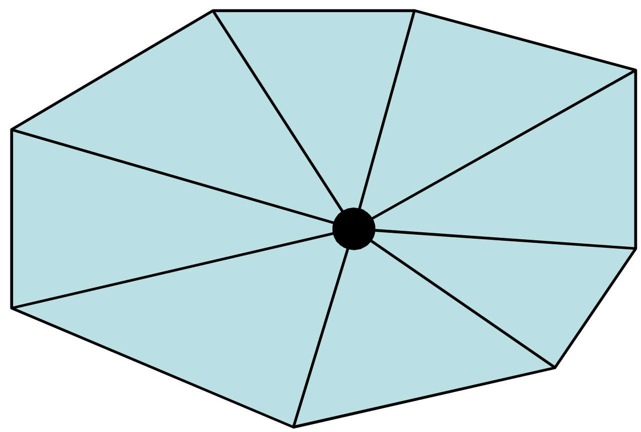
Propriedades

- Gera uma malha poligonal, definida por um conjunto de vértices, arestas e faces
  - Cada aresta é compartilhada, no máximo, por duas faces
  - A intersecção de duas faces é uma aresta, um vértice ou vazia
- Adjacência de vértices, arestas e faces é designada **topologia** da superfície

Decomposição Poligonal

Operações sobre Malhas Poligonais

- Achar todas as arestas que incidem num vértice
- Achar as faces que incidem numa aresta ou vértice
- Achar as arestas na fronteira de uma face
- Desenhar a malha



Codificação

- Explícita
- Ponteiros para lista de vértices
- Ponteiros para lista de arestas
- *Winged-Edge* (*Half-Edge, Face-Edge*)
- *Quad-Edge* (Guibas-Stolfi)
- *Radial-Edge*

Codificação Explícita

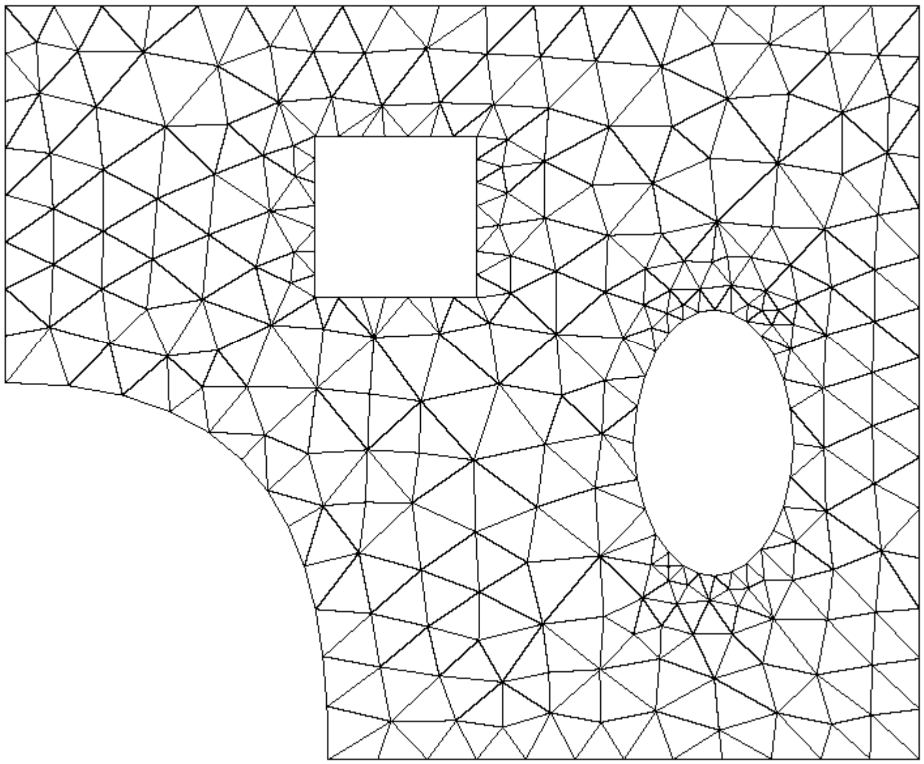
- A mais simples
- Cada face armazena explicitamente a lista ordenada das coordenadas dos seus vértices:

$$P = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)\}$$

- Muita redundância de informação
- Consultas são complicadas
  - Obriga à execução de algoritmos geométricos para determinar adjacências

Desenho da Malha

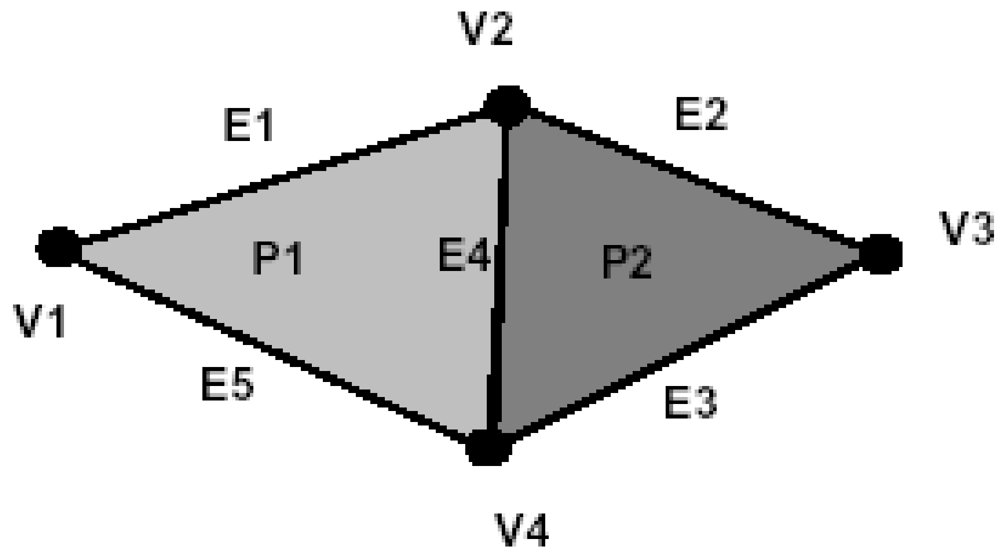
- Cada aresta é desenhada duas vezes - pelas duas faces que a compartilham
- Não é bom para plotters ou filmes



Ponteiros para Lista de Vértices

- Vértices são armazenados separadamente
- Há uma lista de vértices
- Faces referenciam os seus vértices através de ponteiros
- Proporciona maior economia de memória
- Achar adjacências ainda é complicado
- Arestas ainda são desenhadas duas vezes

Exemplo



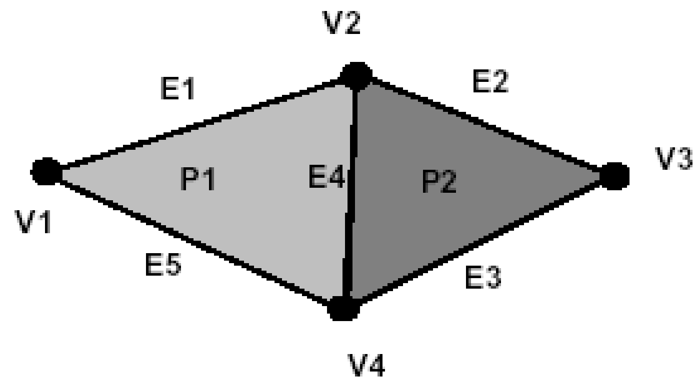
- $V = \{V_1 = (x_1, y_1, z_1), V_2 = (x_2, y_2, z_2), V_3 = (x_3, y_3, z_3), V_4 = (x_4, y_4, z_4)\}$;
- $P_1 = \{V_1, V_2, V_4\}$;
- $P_2 = \{V_4, V_2, V_3\}$.

Ponteiros para Lista de Arestas

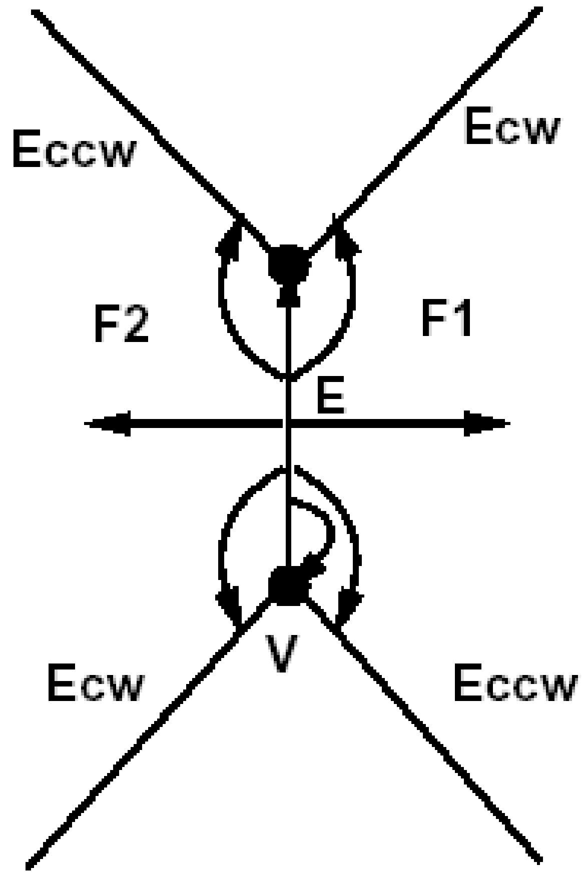
- Há também uma lista de arestas
- Faces referenciam as suas arestas através de ponteiros
- Arestas são desenhadas percorrendo-se a lista de arestas
- Introduzem-se referências para as duas faces que compartilham uma aresta
  - Facilita a determinação das duas faces incidentes na aresta

Exemplo

- $V = \{V_1 = (x_1, y_1, z_1), V_2 = (x_2, y_2, z_2), V_3 = (x_3, y_3, z_3), V_4 = (x_4, y_4, z_4)\}$;
- $E_1 = \{V_1, V_2, P_1, \lambda\}$;
- $E_2 = \{V_2, V_3, P_2, \lambda\}$;
- $E_3 = \{V_3, V_4, P_2, \lambda\}$;
- $E_4 = \{V_2, V_4, P_1, P_2\}$;
- $E_5 = \{V_4, V_1, P_1, \lambda\}$;
- $P_1 = \{E_1, E_4, E_5\}$;
- $P_2 = \{E_2, E_3, E_4\}$.



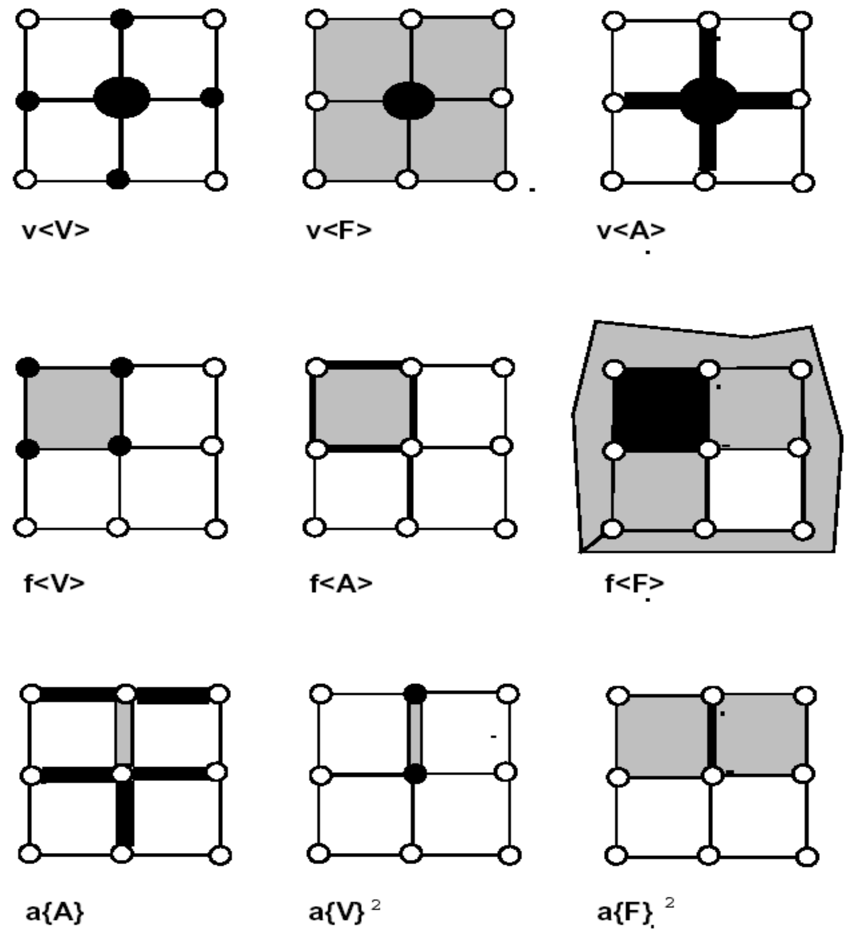
Winged-Edge



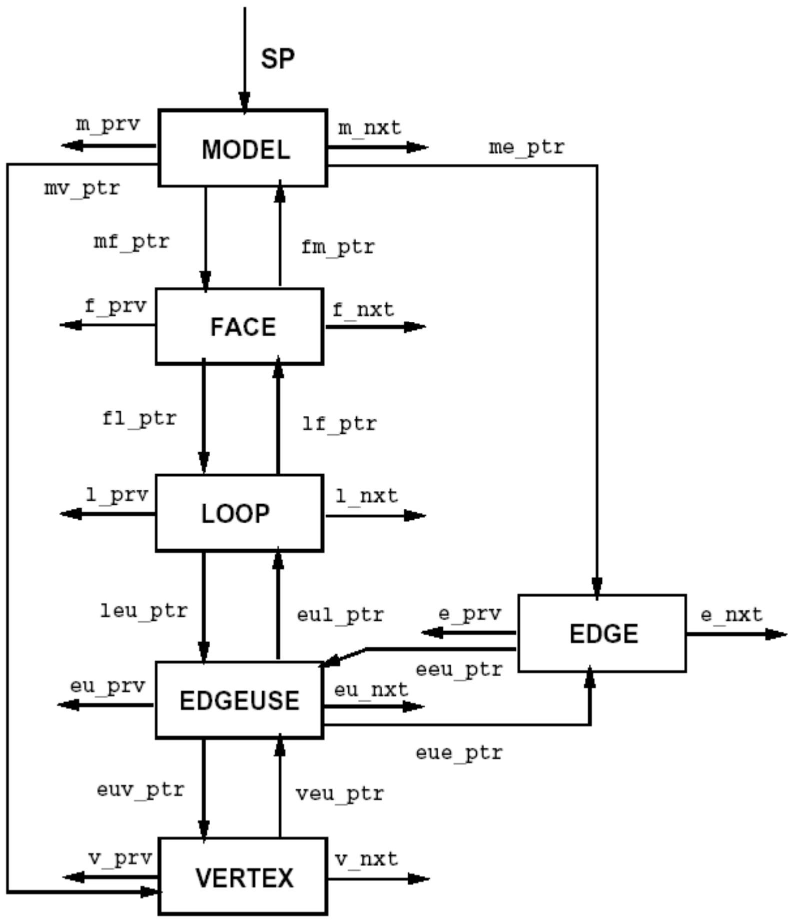
Winged-Edge

- Criada em 1974 por Baumgart
- Foi um marco na representação por fronteira
- Armazena informação na estrutura associada às arestas (número de campos é fixo)
- Todos os 9 tipos de adjacência entre vértices, arestas e faces são determinados em tempo constante
- Actualizada com o uso de operadores de Euler, que garantem: $V - A + F = 2$

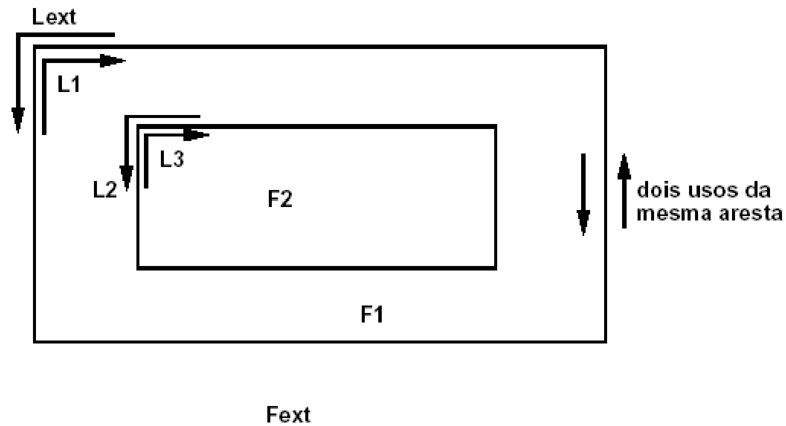
9 tipos de Relacionamentos de Adjacência



Face-Edge



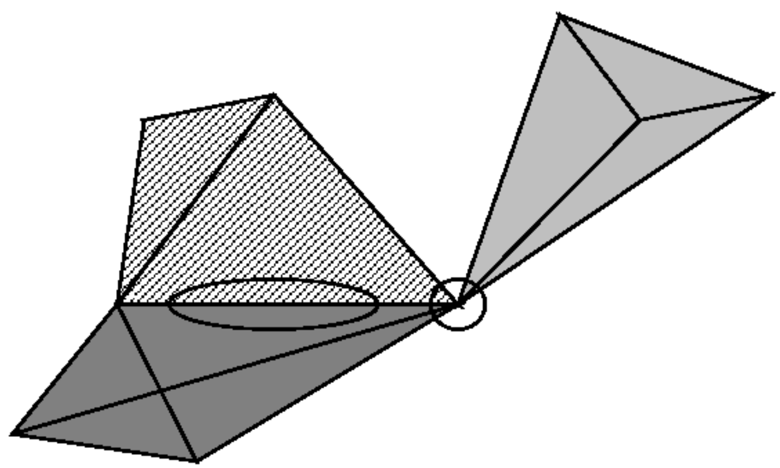
SP
MODEL
m_prv
m_nxt
me_ptr
mv_ptr
mf_ptr
fm_ptr
FACE
f_prv
f_nxt
fl_ptr
lf_ptr
LOOP
l_prv
l_nxt
leu_ptr
eul_ptr
e_prv
EDGE
e_nxt
eeu_ptr
EDGEUSE
eu_prv
eu_nxt
eue_ptr
euv_ptr
veu_ptr
VERTEX
v_prv
v_nxt



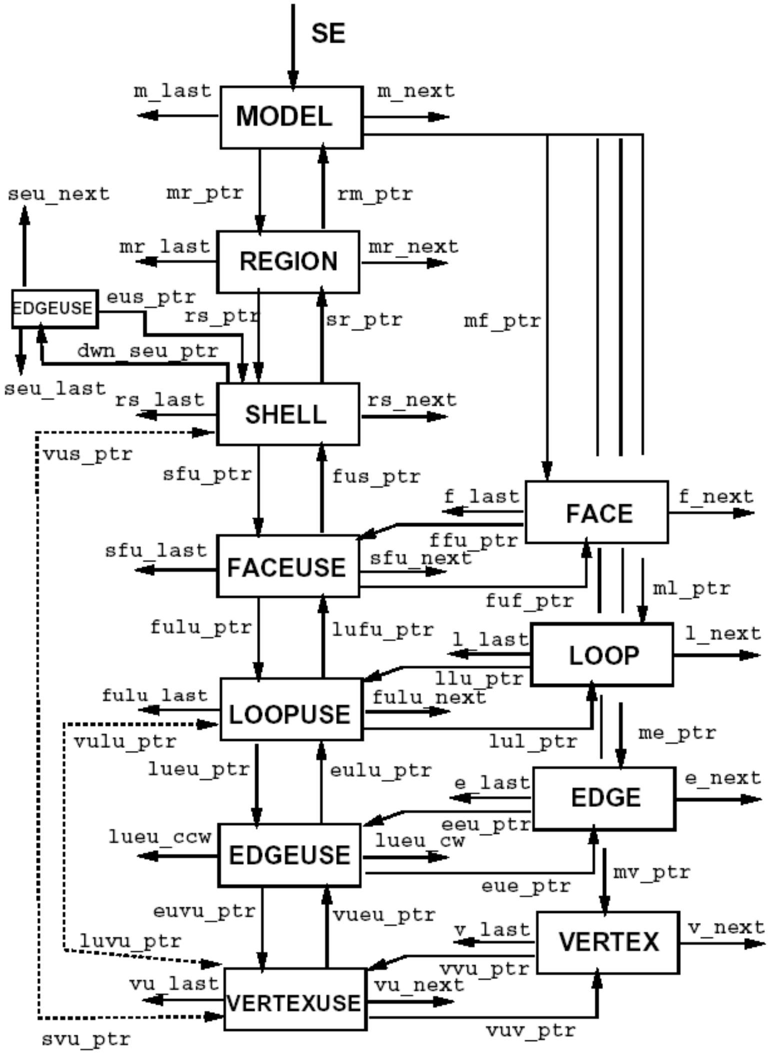
Lext
L1
L2
L3
F2
dois usos da
mesma aresta
F1
Fext

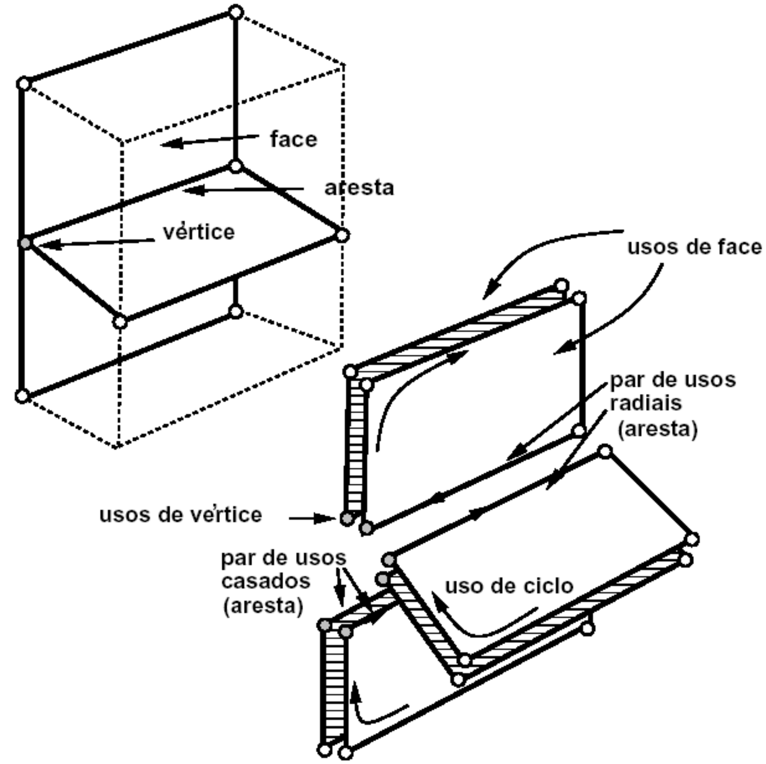
Radial-Edge

- Criada em 1986 por Weiler
- Representa objectos *non-manifold* (não variedades)
- Armazena a lista ordenada de faces incidentes em uma aresta
- Muito mais complicada que a *Winged-Edge*



Radial-Edge


SE
MODEL
m_last
m_next
mr_ptr
rm_ptr
seu_next
REGION
mr_last
mr_next
EDGEUSE
eus_ptr
rs_ptr
sr_ptr
mf_ptr
dwn_seu_ptr
seu_last
SHELL
rs_last
rs_next
vus_ptr
sfu_ptr
fus_ptr
FACE
f_last
f_next
ffu_ptr
FACEUSE
sfu_last
sfu_next
fuf_ptr
ml_ptr
fulu_ptr
lufu_ptr
LOOP
l_last
l_next
llu_ptr
LOOPUSE
fulu_last
fulu_next
vulu_ptr
lul_ptr
me_ptr
lueu_ptr
eulu_ptr
EDGE
e_last
e_next
eeu_ptr
EDGEUSE
lueu_ccw
lueu_cw
eue_ptr
mv_ptr
euvu_ptr
vueu_ptr
VERTEX
v_last
v_next
luvu_ptr
vvu_ptr
VERTEXUSE
vu_last
vu_next
svu_ptr
vuv_ptr


face
aresta
vértice
usos de face
par de usos radiais (aresta)
usos de vértice
par de usos casados (aresta)
uso de ciclo

Representação Implícita

- Sólido é definido por um conjunto de valores que caracterizam os seus pontos
- Descreve a superfície dos objectos, implicitamente, por uma equação:

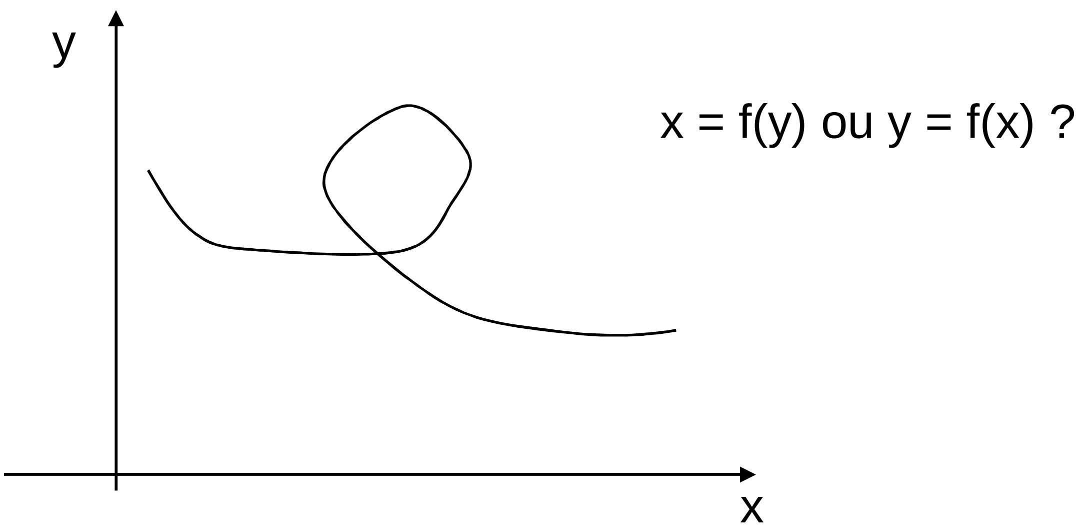
$$F(x) = c; X \in \Re^n, c \in \Re.$$

$$F : \Re^n \to \Re \text{ de classe } C^k.$$

- F é designada função implícita

Funções Implícitas

- Uma superfície definida de forma implícita pode apresentar <u>auto-intersecção</u>
- Pergunta: $F(x,y,z)$ define implicitamente $z = f(x,y)$ em algum domínio razoável?



Teorema da Função Implícita

- Seja $F : \Re^n \rightarrow \Re$ definida num conjunto aberto U
- Se F possui derivadas parciais contínuas em U e $\nabla F \neq 0$ em U, então F é uma <u>subvariedade</u> de dimensão n - 1 do $\Re^n$
  - Superfície sem auto-intersecção

Valores Regulares

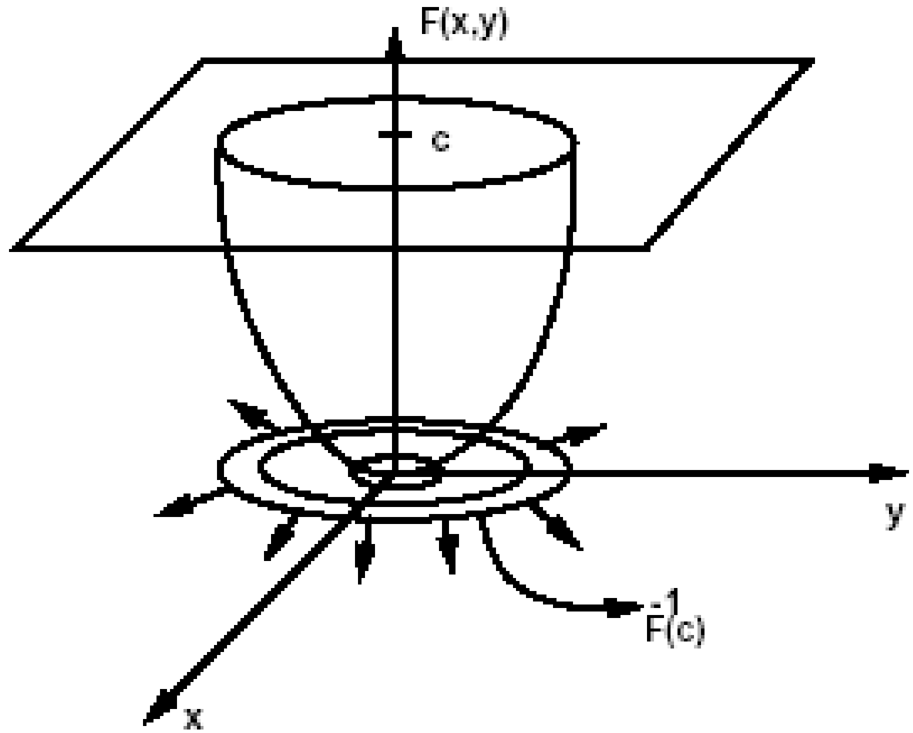
- Um valor c é dito **regular** se $F^{-1}(c)$ não contém pontos onde $\nabla F = 0$ (pontos singulares)

$$\forall p \in F^{-1}(c) \Rightarrow \nabla F_p = \left(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z} \right) \Big|_p \neq 0.$$

- Neste caso interessam apenas os casos em que $n = 2$ ou 3 (curvas e superfícies implícitas)

Exemplo 1

- Seja $F(x,y) = x^2 + y^2$ que define um parabolóide no $\Re^3$
- Curvas de nível são círculos
- $\nabla F = (2x, 2y)$ anula-se na origem
- 0 não é valor regular de F Logo $F(x,y) = 0$ não define uma função implícita



Exemplo 2

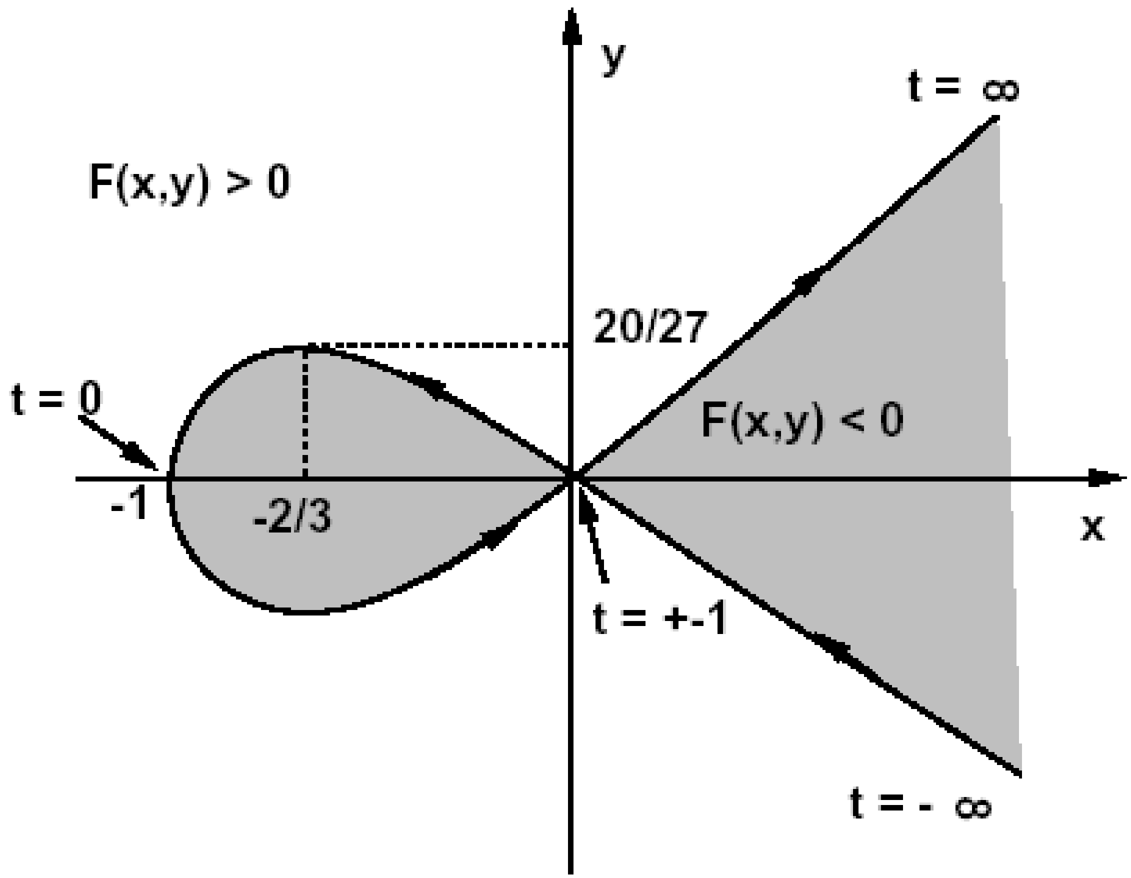
- Cascas esféricas: $F(x,y,z) = x^2 + y^2 + z^2$
- Para todo $k > 0$, $F^{-1}(k)$ representa a superfície de uma esfera em $\Re^3$
- 0 não é valor regular de F
  - $F^{-1}(0) = (0,0,0)$ e $\nabla F=(2x, 2y, 2z)$ anula-se na origem

Exemplo 3

- $F(x,y) = y^2 - x^2 - x^3$, $\nabla F = (2y, -3x^2 - 2x)$
- Na forma paramétrica:
  - $x(t) = t^2 - 1$ e $y(t) = t\,(t^2 - 1)$
- Curva de nível 0 é um laço, com uma singularidade na origem:

$$z = F(x,y) = y^2 - x^2 - x^3 = 0$$

Gráfico do Exemplo 3



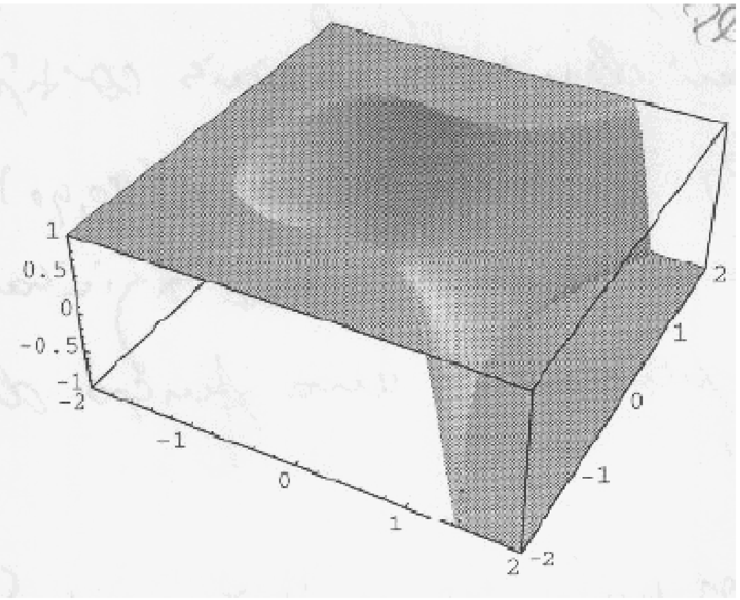
Observação

- Olhando F(x,y) como superfície de nível 0 da função $H : \Re^3 \to \Re$,

  $H(x,y,z) = -z + y^2 - x^2 - x^3$,

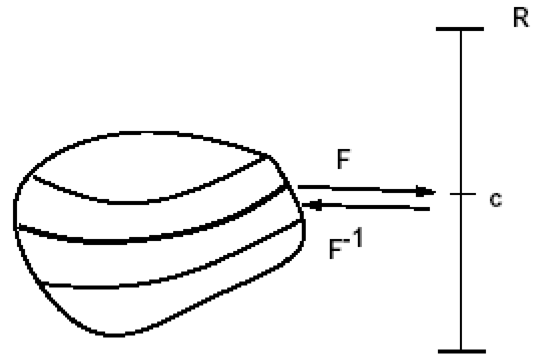
  $\nabla H = (-3x^2 - 2x, 2y, -1)$;

  $\nabla H(0,0,0) = (0,0,-1)$
- Todos os pontos são regulares
- Gráfico de F em $\Re^3$ é realmente o gráfico de uma função!



Objecto Implícito

- Um subconjunto $O \subset \Re^n$ é designado **objecto implícito** se existe $F : U \rightarrow \Re$, $O \subset U$, e existe um subconjunto $V \subset \Re$ / $O = F^{-1}(V)$ ou $O = \{p \in U, F(p) \in V\}$.
- Um objecto implícito é dito **regular** se F satisfaz a condição de regularidade
- Um objecto implícito é <u>válido</u> se define uma superfície em $\Re^n$

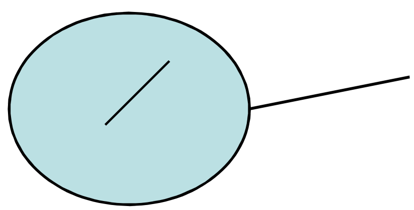


Interior x Exterior

- A função F faz a classificação dos pontos do espaço
- Permite decidir se o ponto está no interior, na fronteira ou no exterior
  - $F > 0 \Rightarrow p \in$ exterior de O
  - $F = 0 \Rightarrow p \in$ fronteira de O
  - $F < 0 \Rightarrow p \in$ interior de O

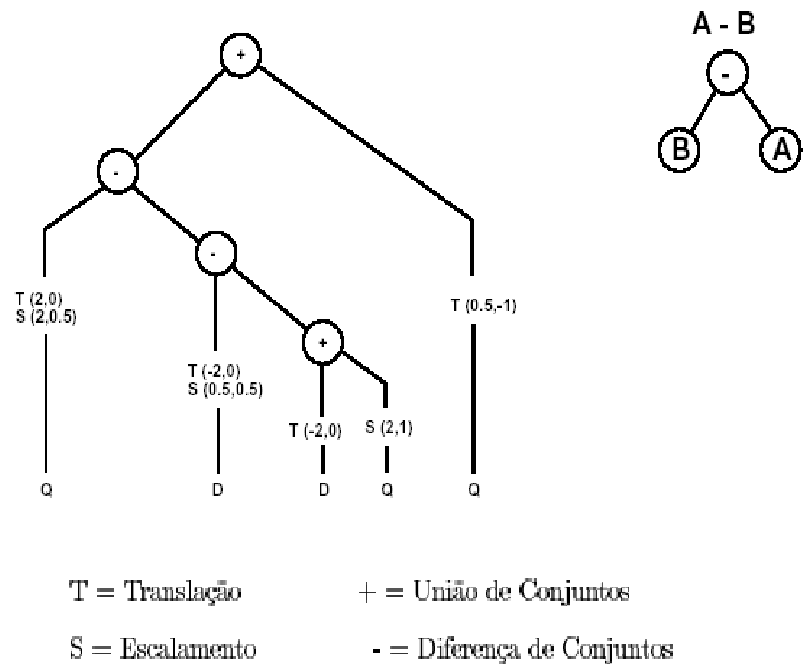
Esquema de Representação CSG

- Operações CSG definem objectos através de operações **regularizadas** de conjuntos de pontos
  - União, Intersecção e Diferença
- Um objecto é **regular** se o <u>fecho</u> do <u>interior</u> do seu conjunto de pontos é igual ao próprio conjunto de pontos



Árvore CSG

- Um modelo CSG é codificado por uma <u>árvore</u>
  - Os nós internos contêm operações de conjunto ou transformações lineares afim
  - Folhas contêm objectos primitivos (tipicamente, quádricas)



CSG com Objectos Implícitos

- Primitivas CSG são definidas por $F_i(X) \leq 0$
- Operações *booleanas* são definidas nesse caso por:
  - $F_1 \cup F_2 = \min(F_1, F_2)$
  - $F_1 \cap F_2 = \max(F_1, F_2)$
  - $F_1 \,/\, F_2 = F_1 \cap \overline{F_2} = \max(F_1, -F_2)$

Prós e Contras de Representações

- Representações por fronteira e por campos escalares apresentam vantagens e desvantagens
- Numa B-rep as intersecções estão representadas explicitamente e é mais fácil exibir um ponto sobre a superfície do objecto
- Porém é difícil determinar, dado um ponto, se ele está no interior, fronteira ou exterior do objecto
- Operações *booleanas* são complicadas

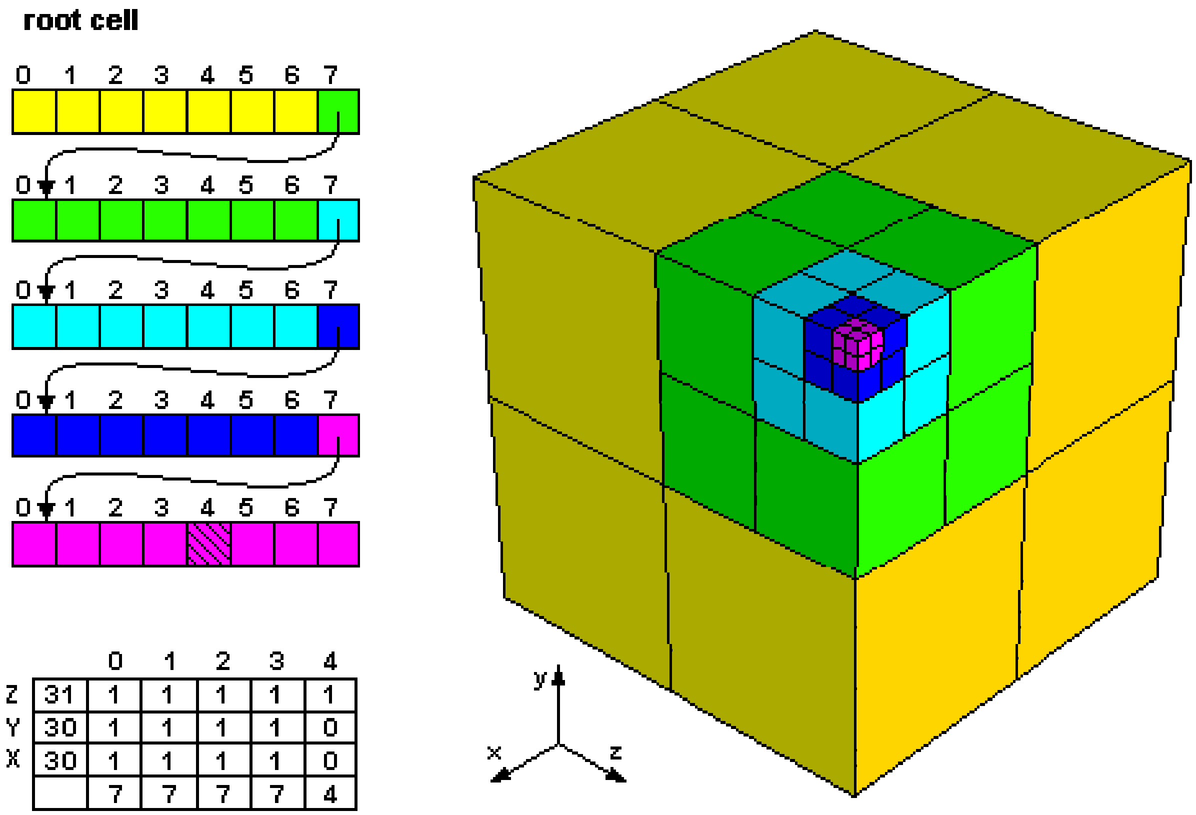
Representações por Campos Escalares

- Em tais representações a classificação de um ponto é imediata, bastando avaliar o sinal do valor do campo no ponto
- Exibir um ponto sobre a superfície do objecto requer a solução de uma equação, a qual pode ser complicada
- Operações *booleanas* são avaliadas facilmente

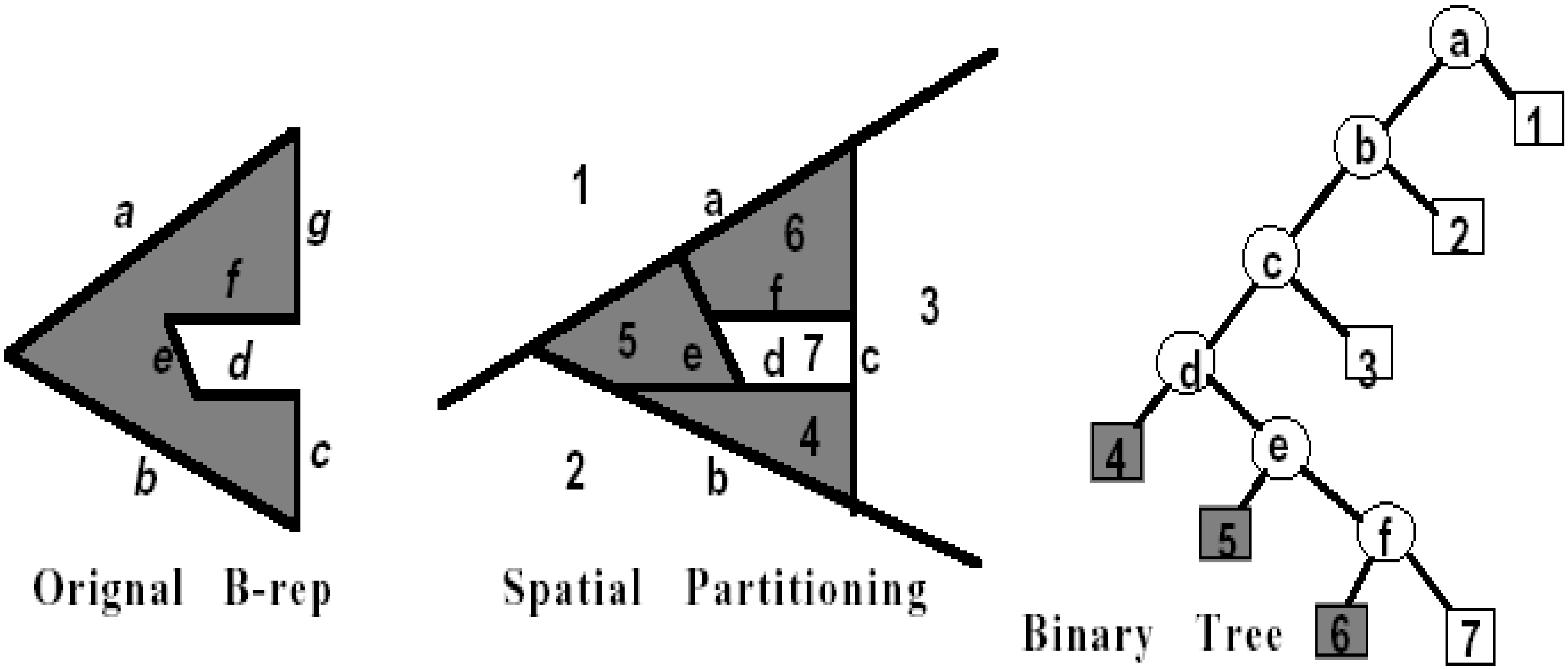
Representações por Células

- Dividem o espaço em sub-regiões convexas
  - Grelhas: Cubos de tamanho igual
  - Octrees: Cubos cujos lados são potências de 2
  - BSP-trees: Poliedros convexos
- Às células são atribuídas valores de um campo escalar $F(x, y, z)$
  - Campo é assumido constante dentro de cada célula
- Sólido é definido como o conjunto de pontos tais que $A < F(x, y, z) < B$ para valores A e B estipulados

Octrees



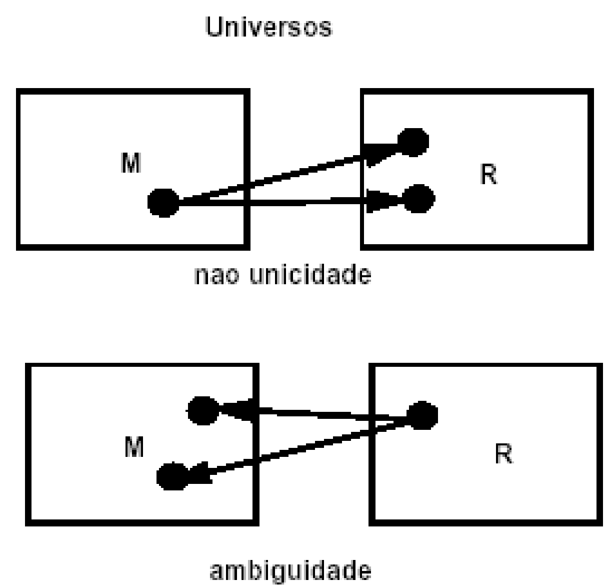
BSP-Trees



a
g
f
e
d
c
b
Orignal B-rep
1
a
6
f
3
5
e
d 7
c
4
2
b
Spatial Partitioning
a
1
b
2
c
3
d
4
e
5
f
6
7
Binary Tree

Ambiguidade e Unicidade

- Uma representação é única quando o modelo associado possui uma única representação
- Uma representação é ambígua quando pode representar mais de um modelo
- Representação ambígua é catastrófica (*wireframe*)
  - Inviabiliza máquinas de controlo numérico



Conversão entre Representações

- Conversão CSG $\rightarrow$ B-rep é denominada **avaliação de fronteira**
- Conversão B-rep $\rightarrow$ CSG é muito mais complicada
- Conversão B-rep $\rightarrow$ Células é simples
- Conversão Células $\rightarrow$ B-rep é relativamente simples (*marching cubes*)
- Conversão CSG $\rightarrow$ Células é simples
- Conversão Células $\rightarrow$ CSG é complicado