

Fundamentos de Processamento Gráfico

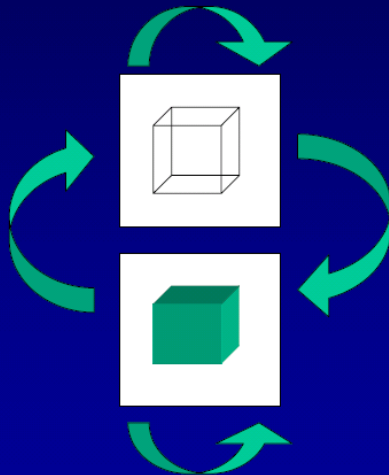
Helton H. Biscaro ; Fátima Nunes

23 de abril de 2018

Áreas Correlatas

Modelagem Geométrica

Visão
Computacional

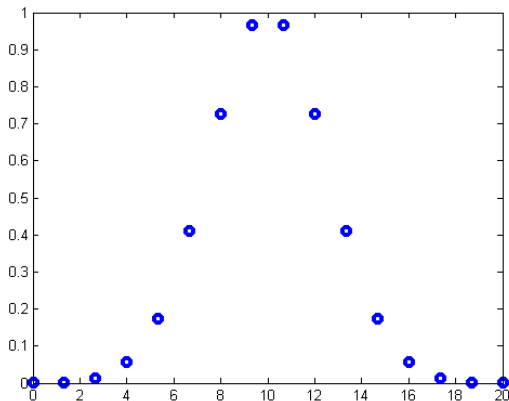


Computação
Gráfica

Processamento de Imagens

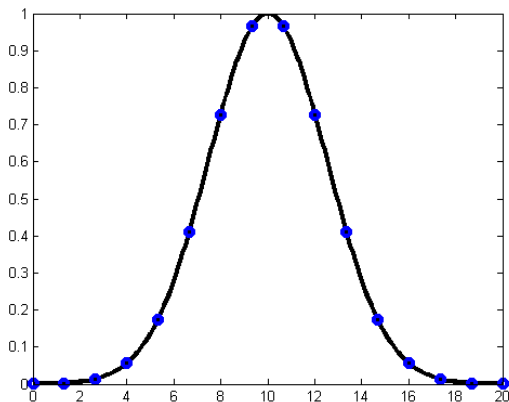
Modelagem: Interpolação

Pontos sobre uma curva.



Modelagem: Interpolação

Pontos sobre uma curva.



Modelagem: Interpolação

Polinômio Interpolador:

Dado um conjunto : $C = \{P_0 = (x_0, y_0), P_1 = (x_1, y_1), \dots, P_n = (x_n, y_n)\}$

queremos ajustar um polinômio da forma: $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$.

Sabemos que:

$$p(x_0) = y_0 ;$$

$$p(x_1) = y_1 ;$$

$$\vdots$$

$$p(x_n) = y_n.$$

Ou ainda:

$$a_0 + a_1x_0 + a_2x_0^2 + \dots + a_{n-1}x_0^{n-1} = y_1 ;$$

$$a_0 + a_1x_1 + a_2x_1^2 + \dots + a_{n-1}x_1^{n-1} = y_2 ;$$

$$\vdots$$

$$a_0 + a_1x_n + a_2x_n^2 + \dots + a_{n-1}x_n^{n-1} = y_n.$$

Modelagem: Interpolação

Polinômio Interpolador:

Na forma Matricial.

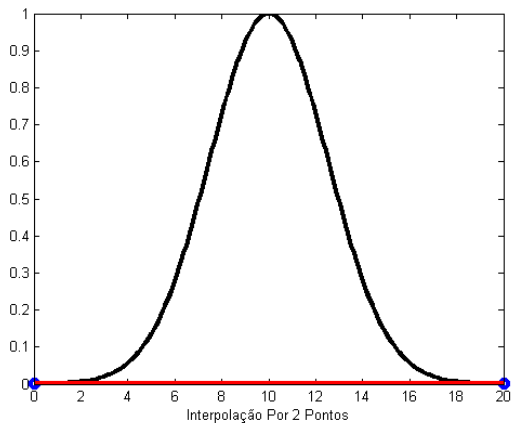
$$\begin{bmatrix} x_0^0 & x_0^1 & \cdots & x_0^{n-1} \\ x_1^0 & x_1^1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \cdots & \vdots \\ x_n^0 & x_n^1 & \cdots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Ou seja. $Ma = y$

Trabalho : Inverter a matriz M , e encontrar o vetor $a = M^{-1}y$.

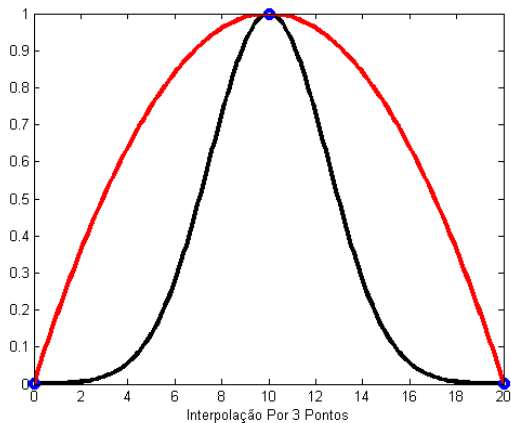
Modelagem: Interpolação

Intrepolação por dois pontos



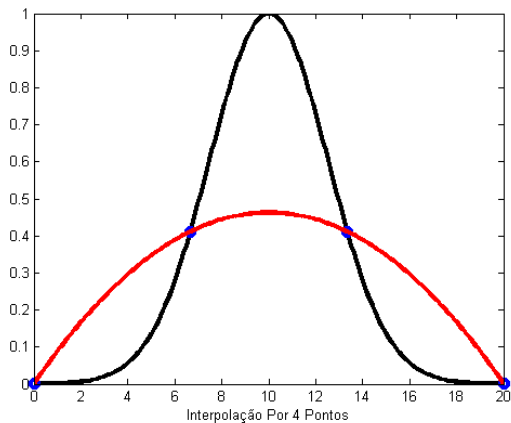
Modelagem: Interpolação

Intrepolação por três pontos



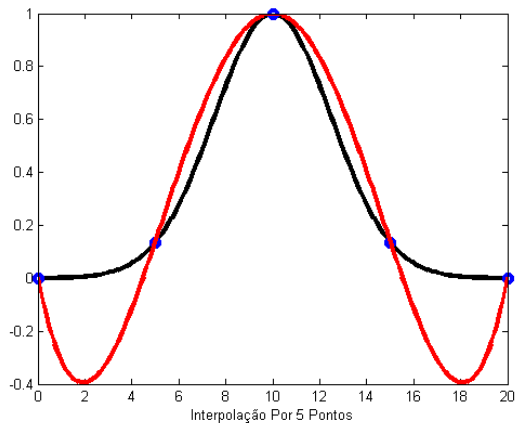
Modelagem: Interpolação

Intrepolação por quatro pontos



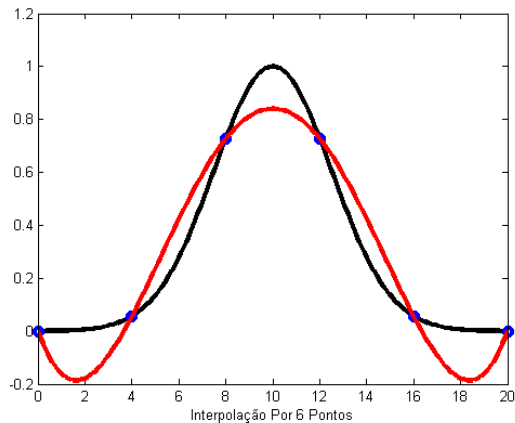
Modelagem: Interpolação

Intrepolação por cinco pontos



Modelagem: Interpolação

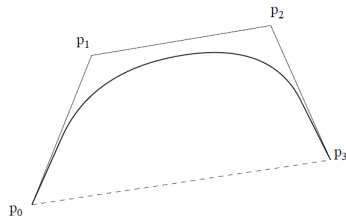
Intrepolação por seis pontos



Modelagem: Bézier

Curva de Bézier:

Dado um conjunto : $C = \{P_0 = (x_0, y_0), P_1 = (x_1, y_1), \dots, P_n = (x_n, y_n)\}$.



$$P(t) = \sum_{i=0}^n P_i B_{n,i}(t), \text{ onde:}$$

$$B_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i}, t \in [0, 1].$$

Figura: Polígono de Controle

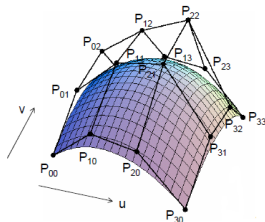
Modelagem: Bézier

Curva de Bézier - Propriedades:

- 1 **É polinomial**. Se a quantidade de pontos de controle é $(n + 1)$, o grau do polinômio é n .
- 2 Acompanha a forma do polígono de controle. Está no **fecho convexo** dos pontos de controle.
- 3 Influência **global** de cada ponto de controle.
- 4 **Pontos extremos** da curva e do polígono coincidem.
- 5 **Tangente** nos pontos extremos coincidem com os segmentos extremos do polígono de controle.
- 6 A curva **não oscila** mais que o polígono de controle.
- 7 São **invariantes** sob transformações afins.

Modelagem: Bézier

Superfície de Bézier :



$$P(u, v) = \sum_{j=0}^m B_{m,j}(v) \left(\sum_{i=0}^n P_i B_{n,i}(u) P_{ij} \right) \text{ onde:}$$

$$\begin{cases} B_{m,j}(v) = \binom{m}{j} v^j (1-v)^{m-j} \\ B_{n,i}(u) = \binom{n}{i} u^i (1-u)^{n-1} \end{cases}$$

Modelagem: B-Splines

Curva B-Splines :

Uma Curva B-Spline de ordem $k, 2 \leq k \leq n + 1$, é definida por :

$P : [t_0, t_{n+k}] \rightarrow \mathbb{R}$, sendo que:

$$P(t) = \sum_{i=0}^n P_i N_i^k(t)$$

onde P_0, P_1, \dots, P_n são pontos de controle, e as funções $N_i^k(t)$ são definidas recursivamente:

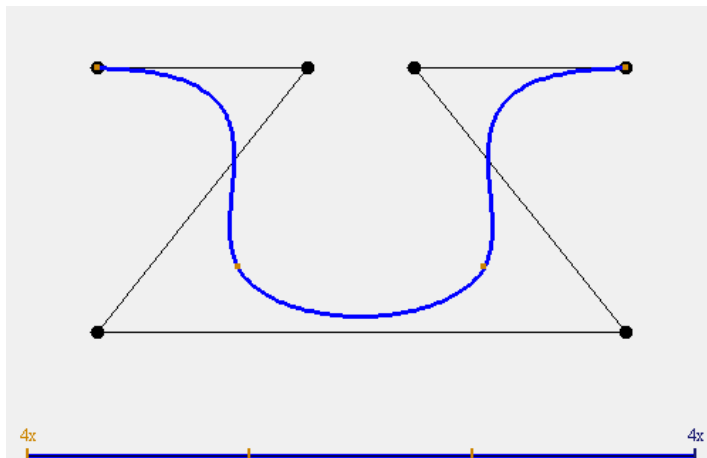
$$N_i^1(t) = \begin{cases} 1, & \text{se } t_i \leq t \leq t_{i+1} \\ 0, & \text{caso contrário} \end{cases} \quad (1)$$

$$N_i^k(t) = \frac{(t-t_i)N_i^{k-1}(t)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-t)N_{i+1}^{k-1}(t)}{t_{i+k}-t_{i+1}}$$

Sendo que $t_0 \leq t_1 \leq \dots \leq t_{n+k}$ são os nós da parametrização.

Modelagem: B-Spline

B-Spline - Exemplo:



Modelagem: B-Splines

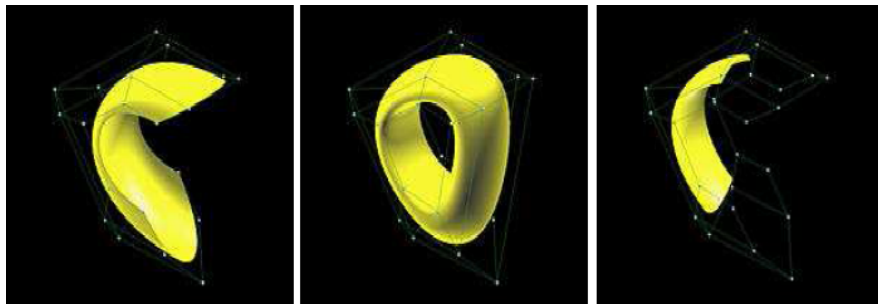
B-Splines - Propriedades:

- 1 **É polinomial.** P é um polinômio de grau k em $[t_i, t_{i+1}]$.
- 2 **A derivada** de ordem $k - 2$ é contínua em $[t_0, t_{n+k}]$.
- 3 $\sum_{i=0}^n N_i^k(t) = 1$ para todo $t \in [t_0, t_{n+k}]$ e $1 \leq k \leq n + 1$.
- 4 $\sum_{i=0}^n N_i^k(t) \geq 0$ para $t \in [t_0, t_{n+k}]$ e $1 \leq k \leq n + 1$.
- 5 Acompanha a forma do polígono de controle. Está no **fecho convexo** dos pontos de controle.
- 6 Influência **local** de cada ponto de controle.
- 7 **Pontos extremos** da curva e do polígono coincidem.
- 8 **Tangente** nos pontos extremos coincidem com os segmentos extremos do polígono de controle.
- 9 A curva **não oscila** mais que o polígono de controle.
- 10 São **invariantes** sob transformações afins.

Modelagem: B-Splines

B-Splines - Superfícies:

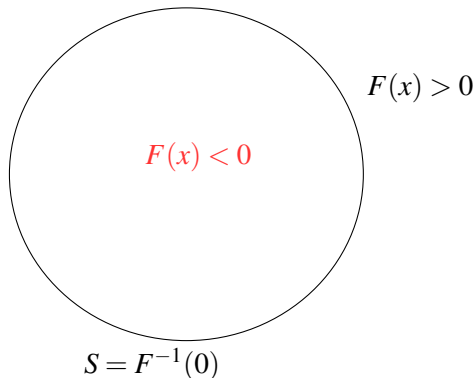
São definidas de forma idêntica à superfície de Bézier



Poligonalização

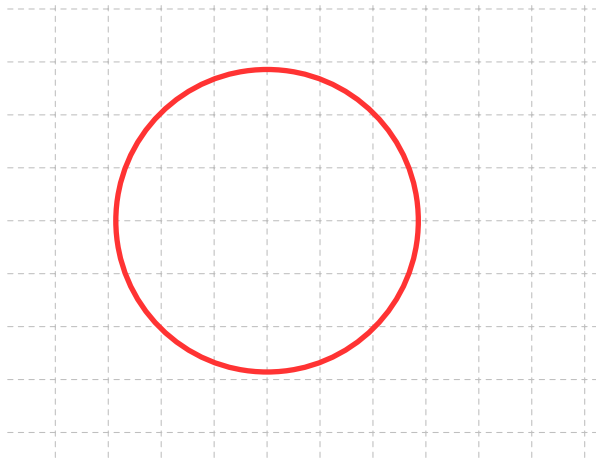
Objetivo

Aproximar por polígonos (triângulos) uma superfície implícita $S = F^{-1}(0)$, onde F é contínua e 0 é valor regular de F .



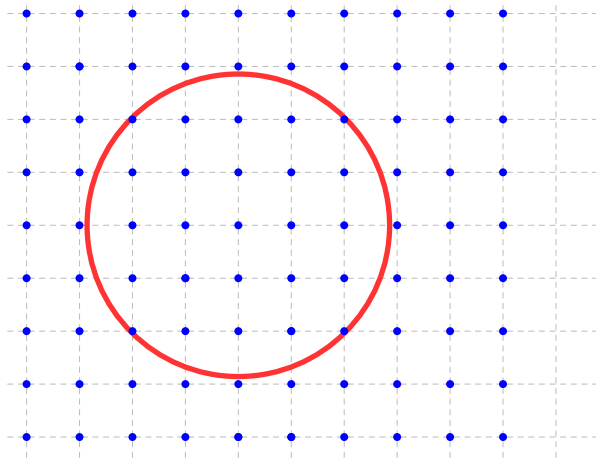
Poligonalização

Algoritmo:



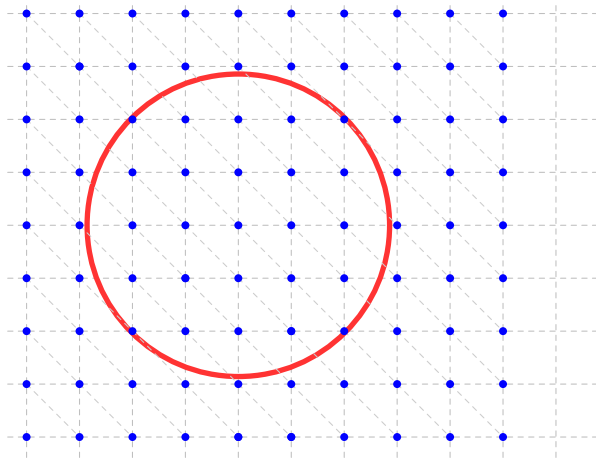
Polgonalização

Algoritmo:



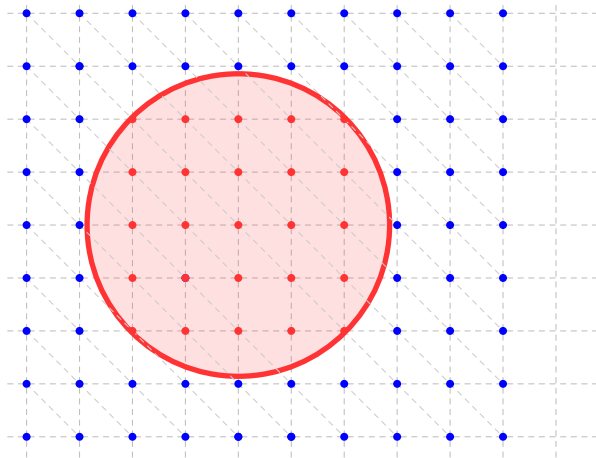
Marching Tetrahedra: Bloomenthal, 1998

Passo 1: Fazer uma triangulação no domínio



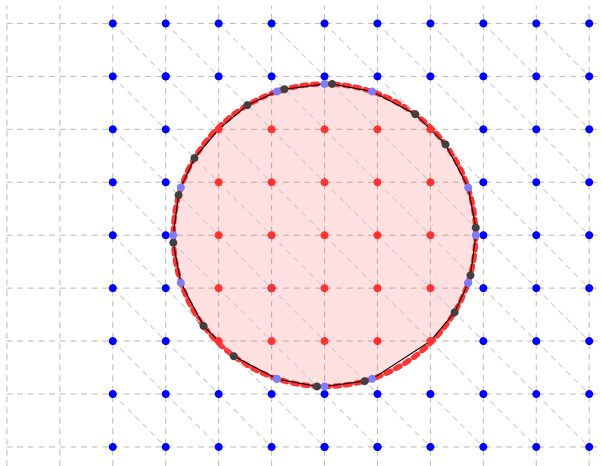
Marching Tetrahedra: Bloomenthal, 1998

Passo 2: Avaliar F em todos os vértices do Grid



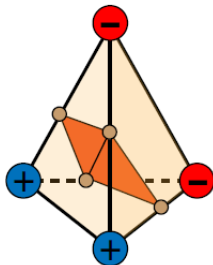
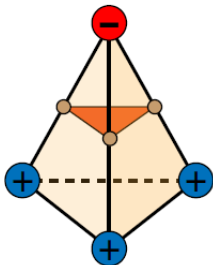
Marching Tetrahedra: Bloomenthal, 1998

Passo 3: Aproximar linearmente nos simplexos onde F muda de sinal.



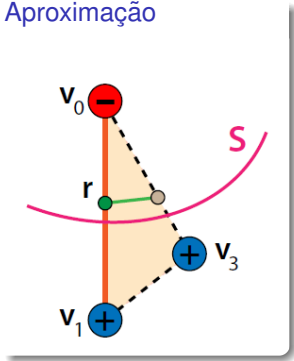
Marching Tetrahedra: Bloomenthal, 1998

Casos possíveis em \mathbb{R}^3 : 2 casos (a menos de permutações) de configuração de sinal da função F em cada tetraedro.



Marching Tetrahedra: Bloomenthal, 1998

Aproximação



Seja $r = F^{-1}(0)$ na aresta $\langle v_0, v_1 \rangle$;

$$r = (1-t)v_0 + tv_1$$

Basta encontrar o valor de t ; Fazendo:

$$0 = F(r) = F((1-t)v_0 + tv_1)$$

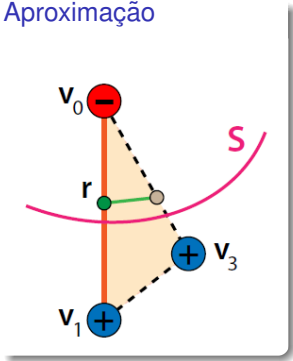
$$\approx (1-t)F(v_0) + tF(v_1)$$

Portanto:

$$t = \frac{F(v_0)}{F(v_0) - F(v_1)}$$

Marching Tetrahedra: Bloomenthal, 1998

Aproximação



Seja $r = F^{-1}(0)$ na aresta $\langle v_0, v_1 \rangle$;

$$r = (1 - t)v_0 + tv_1$$

Basta encontrar o valor de t ; Fazendo:

$$0 = F(r) = F((1 - t)v_0 + tv_1)$$

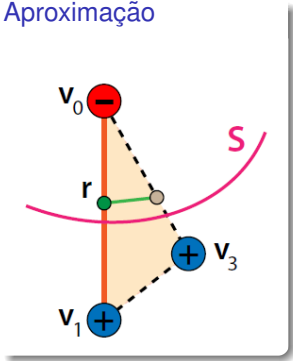
$$\approx (1 - t)F(v_0) + tF(v_1)$$

Portanto:

$$t = \frac{F(v_0)}{F(v_0) - F(v_1)}$$

Marching Tetrahedra: Bloomenthal, 1998

Aproximação



Seja $r = F^{-1}(0)$ na aresta $\langle v_0, v_1 \rangle$;

$$r = (1 - t)v_0 + tv_1$$

Basta encontrar o valor de t ; Fazendo:

$$0 = F(r) = F((1 - t)v_0 + tv_1)$$

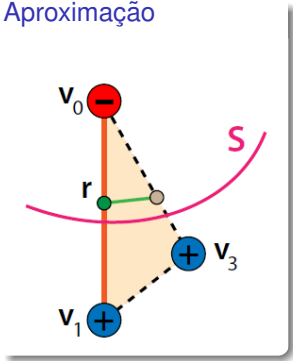
$$\approx (1 - t)F(v_0) + tF(v_1)$$

Portanto:

$$t = \frac{F(v_0)}{F(v_0) - F(v_1)}$$

Marching Tetrahedra: Bloomenthal, 1998

Aproximação



Seja $r = F^{-1}(0)$ na aresta $\langle v_0, v_1 \rangle$;

$$r = (1 - t)v_0 + tv_1$$

Basta encontrar o valor de t ; Fazendo:

$$0 = F(r) = F((1 - t)v_0 + tv_1)$$

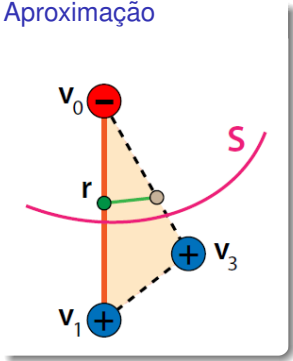
$$\approx (1 - t)F(v_0) + tF(v_1)$$

Portanto:

$$t = \frac{F(v_0)}{F(v_0) - F(v_1)}$$

Marching Tetrahedra: Bloomenthal, 1998

Aproximação



Seja $r = F^{-1}(0)$ na aresta $\langle v_0, v_1 \rangle$;

$$r = (1 - t)v_0 + tv_1$$

Basta encontrar o valor de t ; Fazendo:

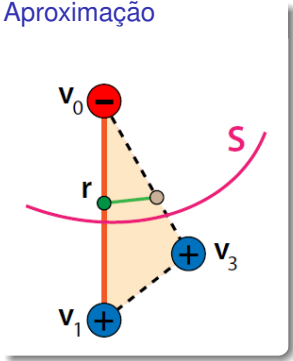
$$0 = F(r) = F((1 - t)v_0 + tv_1) \\ \approx (1 - t)F(v_0) + tF(v_1)$$

Portanto:

$$t = \frac{F(v_0)}{F(v_0) - F(v_1)}$$

Marching Tetrahedra: Bloomenthal, 1998

Aproximação



Seja $r = F^{-1}(0)$ na aresta $\langle v_0, v_1 \rangle$;

$$r = (1 - t)v_0 + tv_1$$

Basta encontrar o valor de t ; Fazendo:

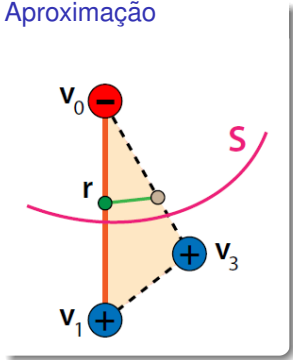
$$\begin{aligned} 0 &= F(r) = F((1 - t)v_0 + tv_1) \\ &\approx (1 - t)F(v_0) + tF(v_1) \end{aligned}$$

Portanto:

$$t = \frac{F(v_0)}{F(v_0) - F(v_1)}$$

Marching Tetrahedra: Bloomenthal, 1998

Aproximação



Seja $r = F^{-1}(0)$ na aresta $\langle v_0, v_1 \rangle$;

$$r = (1 - t)v_0 + tv_1$$

Basta encontrar o valor de t ; Fazendo:

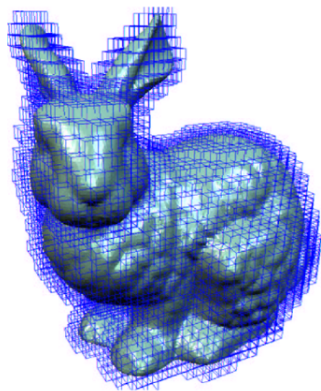
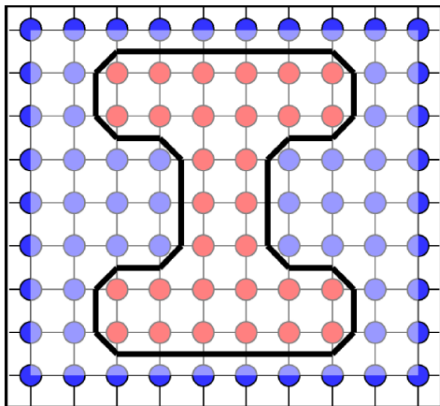
$$0 = F(r) = F((1 - t)v_0 + tv_1) \\ \approx (1 - t)F(v_0) + tF(v_1)$$

Portanto:

$$t = \frac{F(v_0)}{F(v_0) - F(v_1)}$$

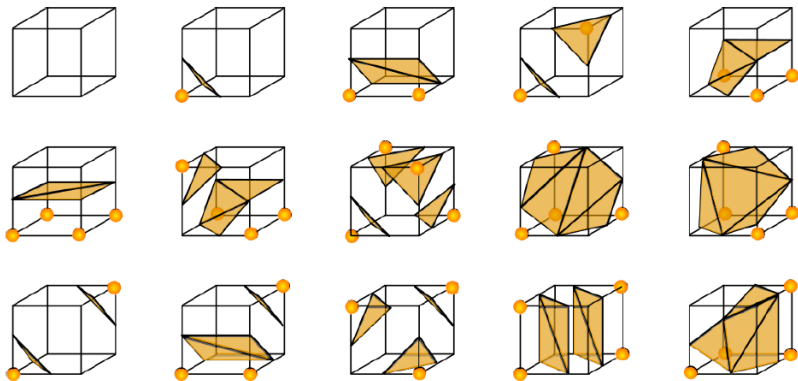
Marching Cubes: Lorensen & Cline, 1987

Faz uma decomposição celular do domínio de F , isto é, particiona o domínio em cubos.



Marching Cubes: Lorensen & Cline, 1987

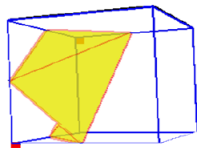
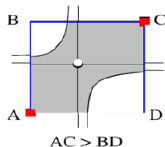
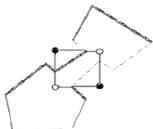
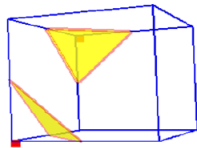
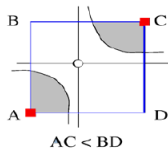
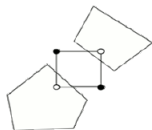
15 casos (a menos de permutações) de configuração de sinal da função F em cada cubo.



Marching Cubes: Lorensen & Cline, 1987

Problemas:

- Ambiguidades
- Dificuldade de implementação



Atividade - Curvas Implícitas

The screenshot displays the Lazarus IDE interface with several windows open:

- Source Editor:** Shows the code for `*Unit1` in `project1.lpr`. The code includes:

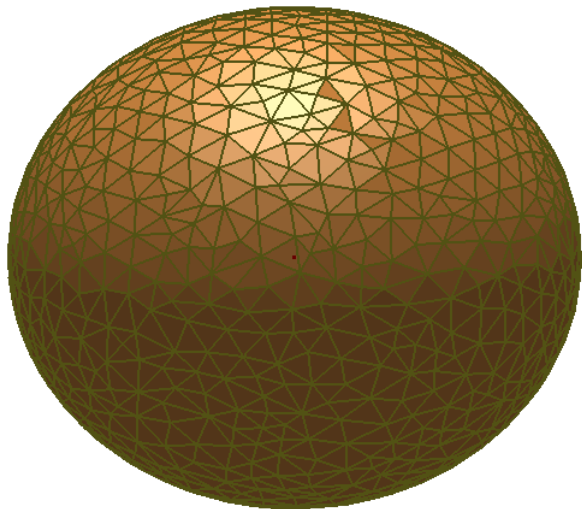

```

      (*SR *.res)
      begin
      Application.Initialize;
      Application.CreateForm(TForm1, Form1);
      Application.Run;
      end.
      
```
- Add code template dialog:** A dialog box with the following fields:
 - Token: `newProc`
 - Comment: `(custom)`
 - Buttons: `Add`, `Cancel`
- Code templates dialog:** A dialog box showing a list of templates:
 - Module: `/home/enrico/lazarus/lazarus.dci`
 - Templates list:
 - `if-else`
 - `if-then (no begin/end) else (no begin/end)`
 - `if-then (no begin/end)`
 - `of-all (case of all enums)`
 - `procedure - procedure declaration`** (highlighted)
 - `todo - ToDo item creator`
 - Buttons: `Add`, `Delete`, `Edit`
 - Options:
 - `procedure - procedure declaration`
 - Enable Macros `Insert Macro`
 - Auto complete on ...
 - line break
 - word end
 - space
 - do not add character
 - Code preview:


```

          procedure $Param(ProcName) | ($Param());
          begin
          |
          end;
          
```
 - Buttons: `Help`, `Cancel`, `OK`
- Form1:** A window showing a graphical form with a `ComboBox1` and an `add` button.
- Messages:** A window at the bottom showing the message: `Project "project1" successfully built`.

Estruturas de Dados



Células e Simplexos

Célula (definição:)

Dado um conjunto de pontos $p_0, p_1, \dots, p_k \in \mathbb{R}^n$, a **célula** gerada por este conjunto é a combinação convexa

$$[p_0, p_1, \dots, p_n] = \left\{ \sum_{i=0}^k \lambda_i p_i; \lambda_i \geq 0; \sum_{i=0}^k \lambda_i = 1 \right\}$$

Células e Simplexos

Célula (definição:)

Dado um conjunto de pontos $\{p_0, p_1, \dots, p_k\} \in \mathbb{R}^n$, a **célula** gerada por este conjunto é a combinação convexa

$$[p_0, p_1, \dots, p_n] = \left\{ \sum_{i=0}^k \lambda_i p_i; \lambda_i \geq 0; \sum_{i=0}^k \lambda_i = 1 \right\}$$

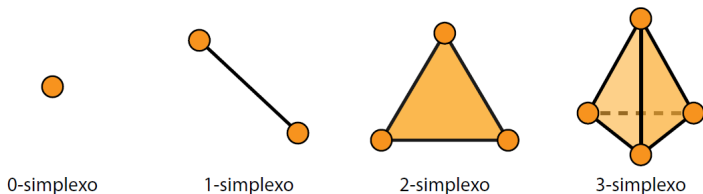
Exemplo

A célula gerada pelos pontos $[p_0, p_1, p_2]$ pode ser um ponto, um segmento de reta, ou um triângulo, de acordo com a relação de dependência linear de $p_1 - p_0, p_2 - p_0$.

Células e Simplexos

Simplexo (definição:)

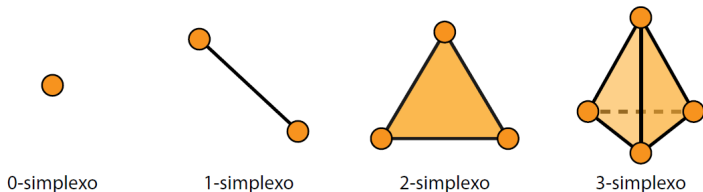
Quando um conjunto de pontos $\{p_0, p_1, \dots, p_k\} \in \mathbb{R}^n$, estão em **posição geral**, a célula formada por eles é chamada de **Simplexo de dimensão k** ou **k -Simplexo**. Denotaremos tal simplexo por $\langle p_0, p_1, \dots, p_k \rangle$.



Células e Simplexos

Sub-Simplexos

Dado um simplexo $\sigma = \langle p_0, p_1, \dots, p_k \rangle$, cada ponto p_i é chamado de **vértice**. Os 1-simplexos gerados pelos pares $[p_i, p_j]$ com $i \neq j$, são chamados de **arestas** e os 2-simplexos gerados por $[p_i, p_j, p_k]$ com $i \neq j \neq k$ são chamados de **faces** de σ .

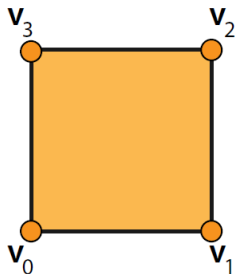


Decomposição celular

Definição

Uma **Decomposição Celular** de um subconjunto $D \in \mathbb{R}^n$ é um conjunto finito de células $C = \{c_i\}$ que satisfaz as seguintes propriedades:

- 1 $D = \bigcup_i c_i$;
- 2 Se $c_i, c_j \in D$, então $c_i \cap c_j \in D$.



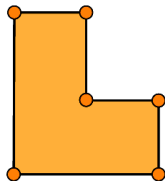
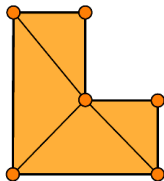
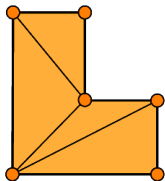
Decomposição celular do quadrado unitário que possui células de 0, 1 e 2 dimensões:

- ▶ dimensão 0: v_0, v_1, v_2, v_3
- ▶ dimensão 1: $[v_0, v_1], [v_1, v_2], [v_2, v_3], [v_3, v_0]$
- ▶ dimensão 2: $[v_0, v_1, v_2, v_3]$

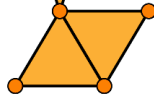
Triangulação

Definição

Quando todos os elementos de uma decomposição celular de uma região D são simplexos, dizemos que ela é uma **Triangulação** de D e denotamos por $\mathcal{T}(D)$.

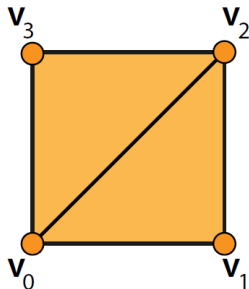
região D duas triangulações de D 

triangulação inválida



triangulação válida

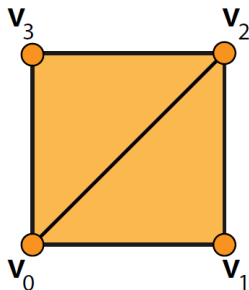
Triangulação de Coxeter-Freudenthal-Kuhn (CFK)



A triangulação do quadrado é formada pelos:

- 0–Simplexos v_0, v_1, v_2, v_3 ;
- 1–Simplexos $\langle v_0, v_1 \rangle, \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \langle v_3, v_0 \rangle$;
- 2–Simplexos $\langle v_0, v_1, v_2 \rangle, \langle v_0, v_2, v_3 \rangle$.

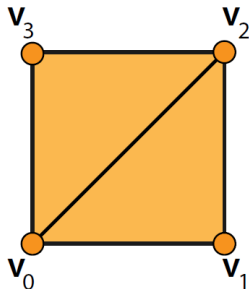
Triangulação de Coxeter-Freudenthal-Kuhn (CFK)



A triangulação do quadrado é formada pelos:

- 0–Simplexos v_0, v_1, v_2, v_3 ;
- 1–Simplexos $\langle v_0, v_1 \rangle, \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \langle v_3, v_0 \rangle$;
- 2–Simplexos $\langle v_0, v_1, v_2 \rangle, \langle v_0, v_2, v_3 \rangle$.

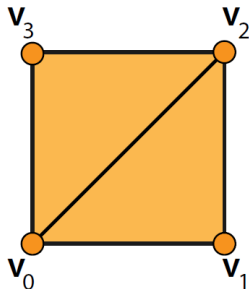
Triangulação de Coxeter-Freudenthal-Kuhn (CFK)



A triangulação do quadrado é formada pelos:

- 0-Simplexos v_0, v_1, v_2, v_3 ;
- 1-Simplexos $\langle v_0, v_1 \rangle, \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \langle v_3, v_0 \rangle$;
- 2-Simplexos $\langle v_0, v_1, v_2 \rangle, \langle v_0, v_2, v_3 \rangle$.

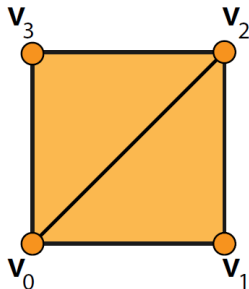
Triangulação de Coxeter-Freudenthal-Kuhn (CFK)



A triangulação do quadrado é formada pelos:

- 0-Simplexos v_0, v_1, v_2, v_3 ;
- 1-Simplexos $\langle v_0, v_1 \rangle, \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \langle v_3, v_0 \rangle$;
- 2-Simplexos $\langle v_0, v_1, v_2 \rangle, \langle v_0, v_2, v_3 \rangle$.

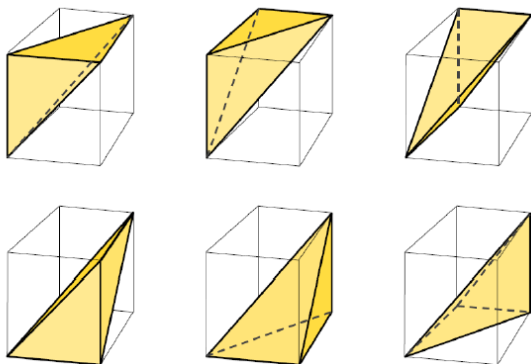
Triangulação de Coxeter-Freudenthal-Kuhn (CFK)



A triangulação do quadrado é formada pelos:

- 0-Simplexos v_0, v_1, v_2, v_3 ;
- 1-Simplexos $\langle v_0, v_1 \rangle, \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \langle v_3, v_0 \rangle$;
- 2-Simplexos $\langle v_0, v_1, v_2 \rangle, \langle v_0, v_2, v_3 \rangle$.

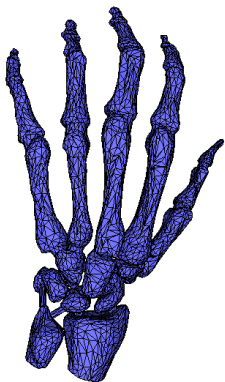
Triangulação de Coxeter-Freudenthal-Kuhn (CFK)



Usando a diagonal do cubo, podemos decompô-lo em 6 tetraedros (Cada um deles um 3-simplexo).

Estrutura de Dados

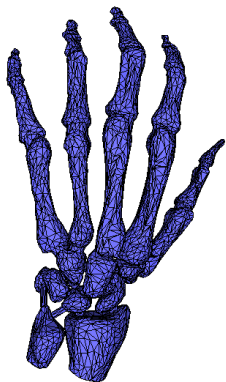
O que armazenar em uma ED?



Geometria (Coordenadas 2D ou 3D);
Atributos do vértice ou da face
(Normal, cor, textura, etc...);
Topologia (Relações de Vizinhaça ou
Conectividade).

Estrutura de Dados

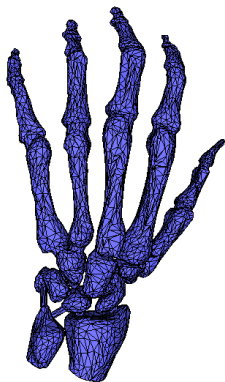
O que armazenar em uma ED?



Geometria (Coordenadas 2D ou 3D);
Atributos do vértice ou da face
(Normal, cor, textura, etc...);
Topologia (Relações de Vizinhaça ou
Conectividade).

Estrutura de Dados

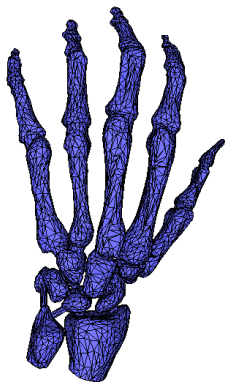
O que armazenar em uma ED?



Geometria (Coordenadas 2D ou 3D);
Atributos do vértice ou da face
(Normal, cor, textura, etc...);
Topologia (Relações de Vizinhaça ou
Conectividade).

Estrutura de Dados

O que armazenar em uma ED?

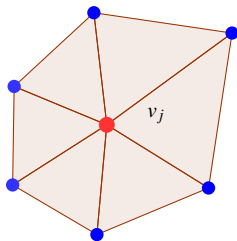


Geometria (Coordenadas 2D ou 3D);
Atributos do vértice ou da face
(Normal, cor, textura, etc...);
Topologia (Relações de Vizinhaça ou
Conectividade).

Estrutura de Dados

O que a ED deve suportar?

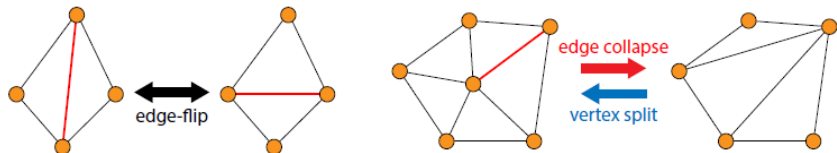
- Rendering;
- Consultas Geométricas;
 - Quais são os vértices de uma determinada face f ?
 - Quais são as faces do 1-anel do vértice v ?
 - Quais são as faces adjacentes à face k ?



Estrutura de Dados

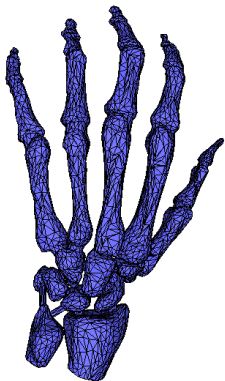
O que da ED deve suportar?

- Modificações;
 - Remover ou adicionar um vértice / face;
 - *edge-flip*, *edge collapse* e *vertex split*;



Estrutura de Dados

Como avaliar a performance de uma ED?



- Tempo de construção (pré-processamento);
- Tempo de resposta a uma consulta;
- Tempo de realização de uma operação;
- Consumo de memória RAM.

Face Set

- Face: 3 posições;
- Não possui conectividade;
- Arquivos no formato STL;
- Simples, porém redundante.

Triângulos		
(x_1^1, y_1^1, z_1^1)	(x_2^1, y_2^1, z_2^1)	(x_3^1, y_3^1, z_3^1)
(x_1^2, y_1^2, z_1^2)	(x_2^2, y_2^2, z_2^2)	(x_3^2, y_3^2, z_3^2)
(x_1^3, y_1^3, z_1^3)	(x_2^3, y_2^3, z_2^3)	(x_3^3, y_3^3, z_3^3)
\vdots	\vdots	\vdots
(x_1^f, y_1^f, z_1^f)	(x_2^f, y_2^f, z_2^f)	(x_3^f, y_3^f, z_3^f)

Shared Vertex

- Vértice: posição + Face : índice dos vértices;
- Não possui conectividade;
- Arquivos no formato OBJ, OFF, PLY;
- Melhor do que a “Face Set ”,mas ainda com pouca informação.

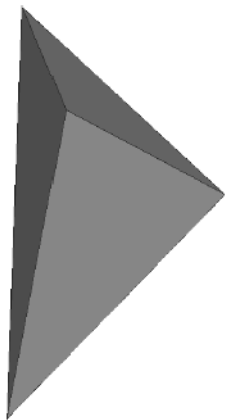
Vértices		
x^1	y^1	z^1
x^2	y^2	z^2
x^3	y^3	z^3
\vdots	\vdots	\vdots
x^v	y^v	z^v

Triângulos		
v_1^1	v_2^1	v_3^1
v_1^2	v_2^2	v_3^2
v_1^3	v_2^3	v_3^3
\vdots	\vdots	\vdots
v_1^f	v_2^f	v_3^f

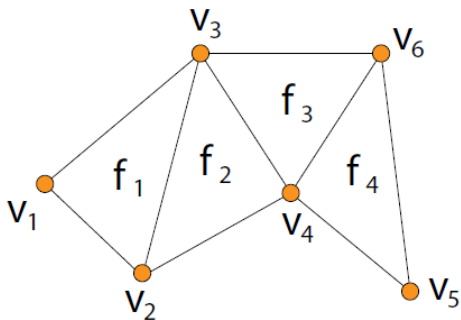
Exemplo: Aquivo OBJ

Tetraedro:

```
# OBJ file format with ext .obj  
v 1.0 0.0 0.0  
v 0.0 1.0 0.0  
v 0.0 0.0 1.0  
v 0.0 0.0 0.0  
f 2 4 3  
f 4 2 1  
f 3 1 2  
f 1 3 4
```

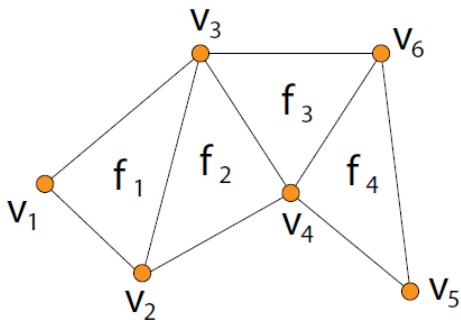


Shared Vertex



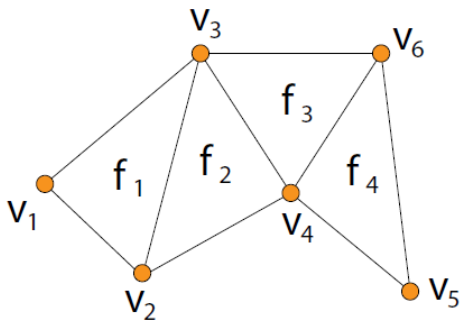
- Quais são os vértices da face f_1 ?
 - $O(1)$; basta consultar a lista de faces;
- Quais são os vértices do 1-anel do vértice v_3 ?
 - $O(2)$; basta consultar a lista de faces;
- Os vértices v_2 e v_6 são adjacentes?

Shared Vertex



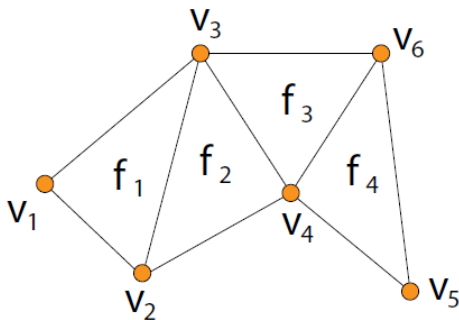
- Quais são os vértices da face f_1 ?
 - $O(1)$; basta consultar a lista de faces;
- Quais são os vértices do 1-anel do vértice v_3 ?
 - Busca completa em todos os vértices;
- Os vértices v_2 e v_6 são adjacentes?

Shared Vertex



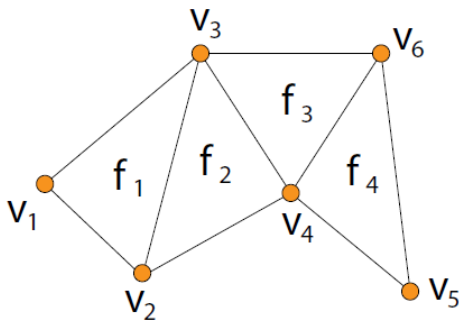
- Quais são os vértices da face f_1 ?
 - $O(1)$; basta consultar a lista de faces;
- Quais são os vértices do 1-anel do vértice v_3 ?
 - Busca completa em todos os vértices;
- Os vértices v_2 e v_6 são adjacentes?

Shared Vertex



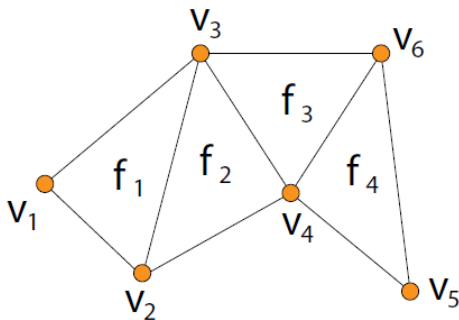
- Quais são os vértices da face f_1 ?
 - $O(1)$; basta consultar a lista de faces;
- Quais são os vértices do 1-anel do vértice v_3 ?
 - Busca completa em todos os vértices;
- Os vértices v_2 e v_6 são adjacentes?
 - Busca completa em todas as faces;

Shared Vertex



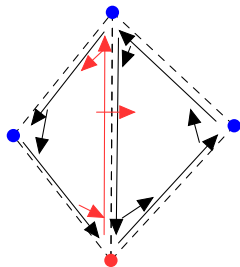
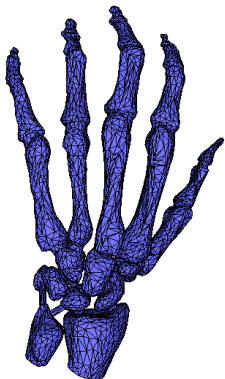
- Quais são os vértices da face f_1 ?
 - $O(1)$; basta consultar a lista de faces;
- Quais são os vértices do 1-anel do vértice v_3 ?
 - Busca completa em todos os vértices;
- Os vértices v_2 e v_6 são adjacentes?
 - Busca completa em todas as faces;

Shared Vertex



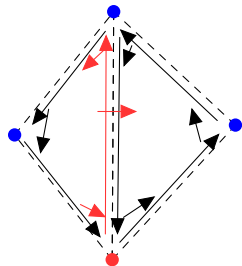
- Quais são os vértices da face f_1 ?
 - $O(1)$; basta consultar a lista de faces;
- Quais são os vértices do 1-anel do vértice v_3 ?
 - Busca completa em todos os vértices;
- Os vértices v_2 e v_6 são adjacentes?
 - Busca completa em todas as faces;

Half Edge



Half Edge

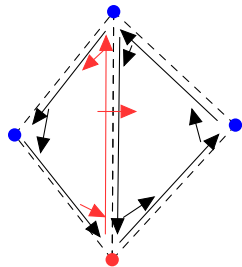
- Vértice
 - Posição
 - 1 HE que "sai" do vértice.
- Half Edge (HE)



- Face

Half Edge

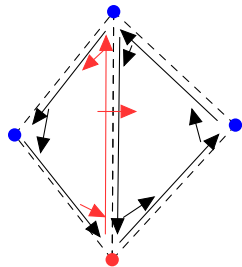
- Vértice
 - Posição
 - 1 HE que “sai ”do vértice.
- Half Edge (HE)



- Face

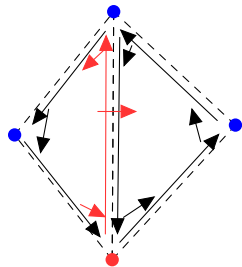
Half Edge

- Vértice
 - Posição
 - 1 HE que “sai ”do vértice.
- Half Edge (HE)
 - Orientação consistente;
 - 1 índice do vértice de origem;
 - 1 índice do vértice destino;
 - 1 índice do face adjacente;
 - 1 índice do face oposta;
- Face



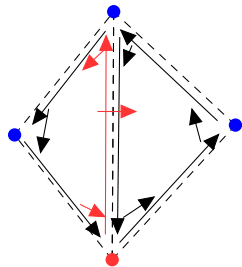
Half Edge

- Vértice
 - Posição
 - 1 HE que “sai ”do vértice.
- Half Edge (HE)
 - Orientação consistente;
 - 1 índice do vértice de origem;
 - 1 índice da face incidente;
 - 1, 2 ou 3 índices de HE's (próxima, anterior e oposta)
- Face



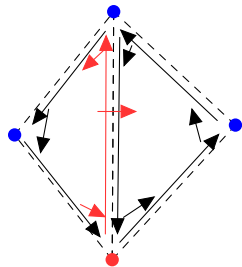
Half Edge

- Vértice
 - Posição
 - 1 HE que “sai ”do vértice.
- Half Edge (HE)
 - Orientação consistente;
 - 1 índice do vértice de origem;
 - 1 índice da face incidente;
 - 1, 2 ou 3 índices de HE's (próxima, anterior e oposta)
- Face



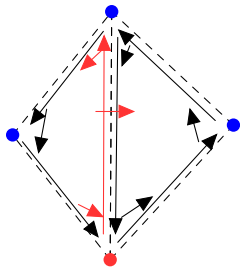
Half Edge

- Vértice
 - Posição
 - 1 HE que “sai ”do vértice.
- Half Edge (HE)
 - Orientação consistente;
 - 1 índice do vértice de origem;
 - 1 índice da face incidente;
 - 1, 2 ou 3 índices de HE's (próxima, anterior e oposta)
- Face



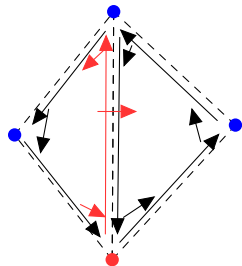
Half Edge

- Vértice
 - Posição
 - 1 HE que “sai ”do vértice.
- Half Edge (HE)
 - Orientação consistente;
 - 1 índice do vértice de origem;
 - 1 índice da face incidente;
 - 1, 2 ou 3 índices de HE's (próxima, anterior e oposta)
- Face



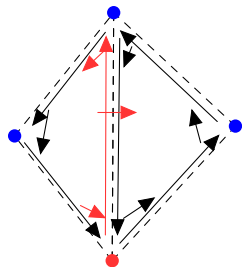
Half Edge

- Vértice
 - Posição
 - 1 HE que “sai ”do vértice.
- Half Edge (HE)
 - Orientação consistente;
 - 1 índice do vértice de origem;
 - 1 índice da face incidente;
 - 1, 2 ou 3 índices de HE's (próxima, anterior e oposta)
- Face
 - 1 índice de HE incidente.



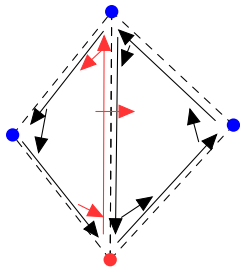
Half Edge

- Vértice
 - Posição
 - 1 HE que “sai ”do vértice.
- Half Edge (HE)
 - Orientação consistente;
 - 1 índice do vértice de origem;
 - 1 índice da face incidente;
 - 1, 2 ou 3 índices de HE's (próxima, anterior e oposta)
- Face
 - 1 índice de HE incidente.



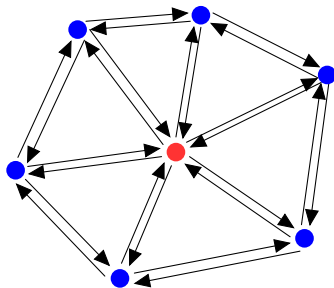
Half Edge

- Vértice
 - Posição
 - 1 HE que “sai ”do vértice.
- Half Edge (HE)
 - Orientação consistente;
 - 1 índice do vértice de origem;
 - 1 índice da face incidente;
 - 1, 2 ou 3 índices de HE's (próxima, anterior e oposta)
- Face
 - 1 índice de HE incidente.



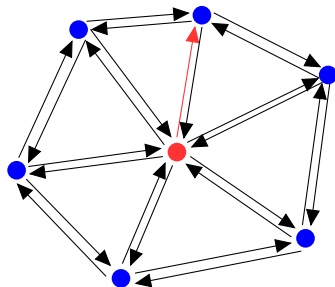
1 – Anel com a Half Edge

- 1 Inicia em um vértice



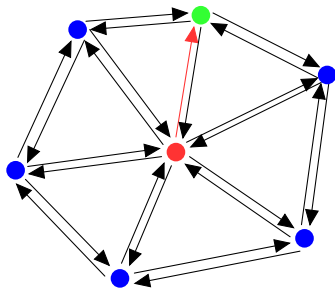
1 – Anel com a Half Edge

- 1 Inicia em um vértice
- 2 HE que sai do vértice



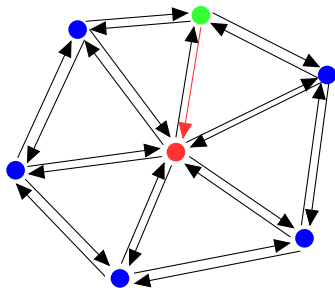
1 – Anel com a Half Edge

- 1 Inicia em um vértice
- 2 HE que sai do vértice



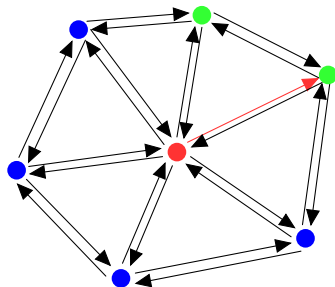
1 – Anel com a Half Edge

- 1 Inicia em um vértice
- 2 HE que sai do vértice
- 3 HE oposta



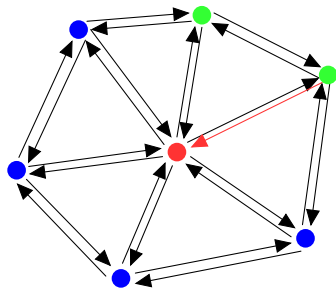
1 – Anel com a Half Edge

- 1 Inicia em um vértice
- 2 HE que sai do vértice
- 3 HE oposta
- 4 Próxima HE



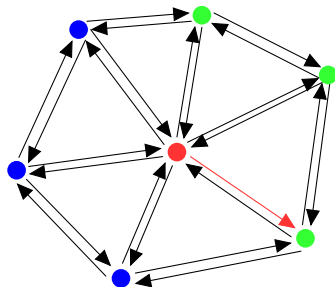
1 – Anel com a Half Edge

- 1 Inicia em um vértice
- 2 HE que sai do vértice
- 3 HE oposta
- 4 Próxima HE
- 5 HE oposta



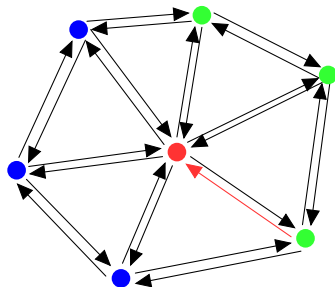
1 – Anel com a Half Edge

- 1 Inicia em um vértice
- 2 HE que sai do vértice
- 3 HE oposta
- 4 Próxima HE
- 5 HE oposta
- 6 Próxima HE



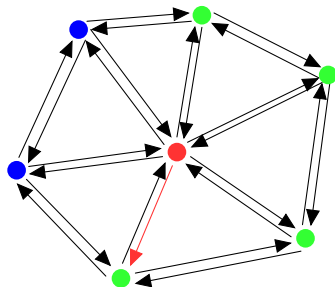
1 – Anel com a Half Edge

- 1 Inicia em um vértice
- 2 HE que sai do vértice
- 3 HE oposta
- 4 Próxima HE
- 5 HE oposta
- 6 Próxima HE
- 7 ...



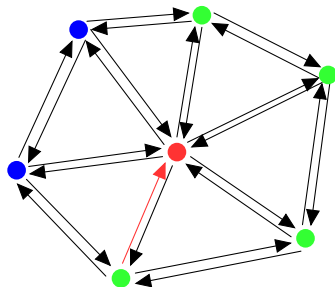
1 – Anel com a Half Edge

- 1 Inicia em um vértice
- 2 HE que sai do vértice
- 3 HE oposta
- 4 Próxima HE
- 5 HE oposta
- 6 Próxima HE
- 7 ...



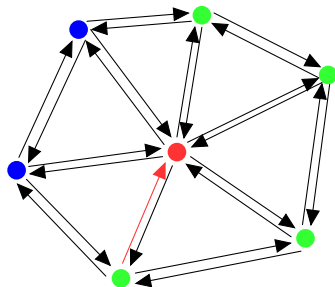
1 – Anel com a Half Edge

- 1 Inicia em um vértice
- 2 HE que sai do vértice
- 3 HE oposta
- 4 Próxima HE
- 5 HE oposta
- 6 Próxima HE
- 7 ...



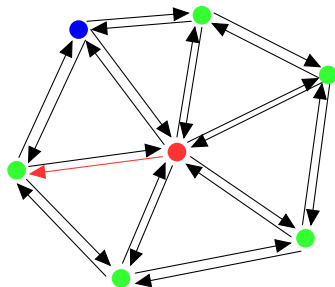
1 – Anel com a Half Edge

- 1 Inicia em um vértice
- 2 HE que sai do vértice
- 3 HE oposta
- 4 Próxima HE
- 5 HE oposta
- 6 Próxima HE
- 7 ...



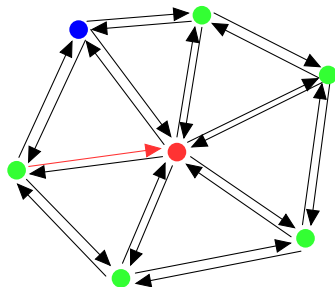
1 – Anel com a Half Edge

- 1 Inicia em um vértice
- 2 HE que sai do vértice
- 3 HE oposta
- 4 Próxima HE
- 5 HE oposta
- 6 Próxima HE
- 7 ...



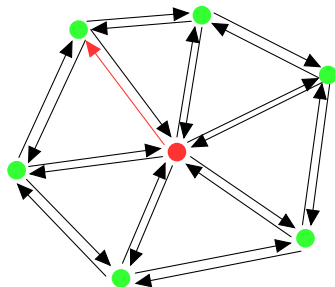
1 – Anel com a Half Edge

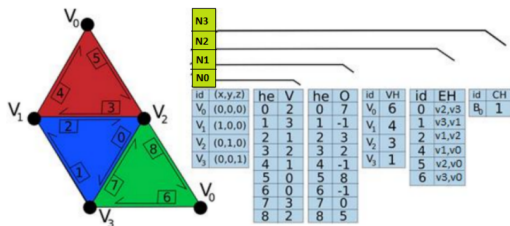
- 1 Inicia em um vértice
- 2 HE que sai do vértice
- 3 HE oposta
- 4 Próxima HE
- 5 HE oposta
- 6 Próxima HE
- 7 ...



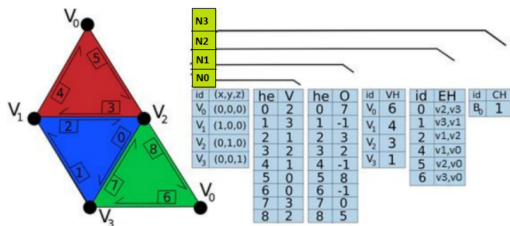
1 – Anel com a Half Edge

- 1 Inicia em um vértice
- 2 HE que sai do vértice
- 3 HE oposta
- 4 Próxima HE
- 5 HE oposta
- 6 Próxima HE
- 7 ...

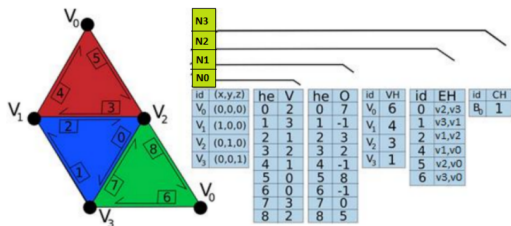


Compact Half Edge [Lages *et al.* 2010]

- Dado um triângulo t , sua primeira *halfedge* está na posição $3t$;
- A posição da próxima *he* é dada por $next_{he} = 3 * \lfloor he/3 \rfloor + (he + 1)\%3$;
- A posição *he* anterior é dada por $prev_{he} = 3 * \lfloor he/3 \rfloor + (he + 2)\%3$;

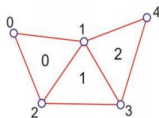
Compact Half Edge [Lages *et al.* 2010]

- Dado um triângulo t , sua primeira *halfedge* está na posição $3t$;
- A posição da próxima *he* é dada por $next_{he} = 3 * \lfloor he/3 \rfloor + (he + 1)\%3$;
- A posição *he* anterior é dada por $prev_{he} = 3 * \lfloor he/3 \rfloor + (he + 2)\%3$;

Compact Half Edge [Lages *et al.* 2010]

- Dado um triângulo t , sua primeira *halfedge* está na posição $3t$;
- A posição da próxima *he* é dada por $next_{he} = 3 * \lfloor he/3 \rfloor + (he + 1)\%3$;
- A posição *he* anterior é dada por $prev_{he} = 3 * \lfloor he/3 \rfloor + (he + 2)\%3$;

Opposite Face [Lizier, *et al.* 2006]



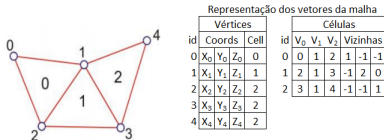
Representação dos vetores da malha

Vértices		
id	Coords	Cell
0	$X_0 \ Y_0 \ Z_0$	0
1	$X_1 \ Y_1 \ Z_1$	1
2	$X_2 \ Y_2 \ Z_2$	2
3	$X_3 \ Y_3 \ Z_3$	2
4	$X_4 \ Y_4 \ Z_4$	2

Células				
id	V_0	V_1	V_2	Vizinhas
0	0	1	2	1 -1 -1
1	2	1	3	-1 2 0
2	3	1	4	-1 -1 1

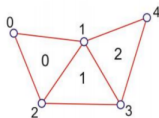
- Vértice:
 - Coordenadas + uma face incidente
- Faces:

Opposite Face [Lizier, *et al.* 2006]



- Vértice:
 - Coordenadas + uma face incidente
- Faces:
 - três índices de vértices + três índices de faces opostas;

Opposite Face [Lizier, *et al.* 2006]

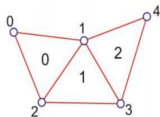


Representação dos vetores da malha

Vértices			Células				
id	Coords	Cell	id	V ₀	V ₁	V ₂	Vizinhas
0	X ₀ Y ₀ Z ₀	0	0	0	1	2	1 -1 -1
1	X ₁ Y ₁ Z ₁	1	1	2	1	3	-1 2 0
2	X ₂ Y ₂ Z ₂	2	2	3	1	4	-1 -1 1
3	X ₃ Y ₃ Z ₃	2					
4	X ₄ Y ₄ Z ₄	2					

- Vértice:
 - Coordenadas + uma face incidente
- Faces:
 - três índices de vértices + três índices de faces opostas;

Opposite Face [Lizier, *et al.* 2006]



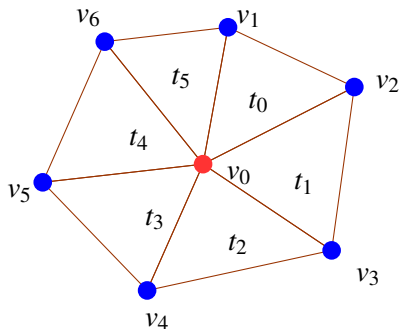
Representação dos vetores da malha

Vértices			Células				
id	Coords	Cell	id	V ₀	V ₁	V ₂	Vizinhas
0	X ₀ Y ₀ Z ₀	0	0	0	1	2	1 -1 -1
1	X ₁ Y ₁ Z ₁	1	1	2	1	3	-1 2 0
2	X ₂ Y ₂ Z ₂	2	2	3	1	4	-1 -1 1
3	X ₃ Y ₃ Z ₃	2					
4	X ₄ Y ₄ Z ₄	2					

- Vértice:
 - Coordenadas + uma face incidente
- Faces:
 - três índices de vértices + três índices de faces opostas;

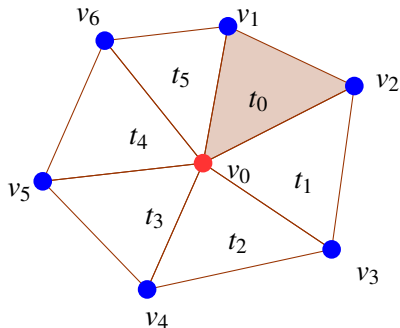
1 – Anel com a Opposite Face

- 1 Inicia em um vértice v_0



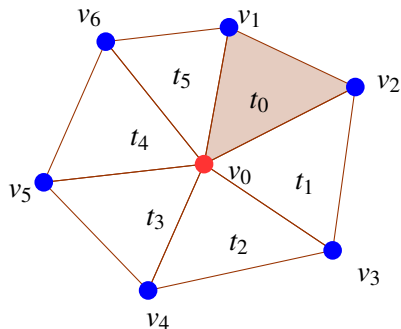
1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0



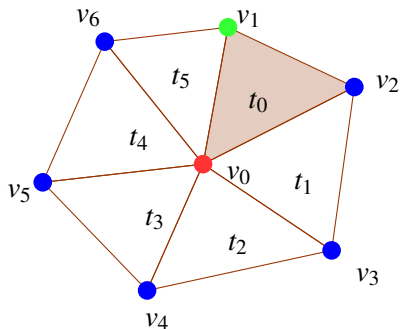
1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0
- 3 Encontre v_0 na face t



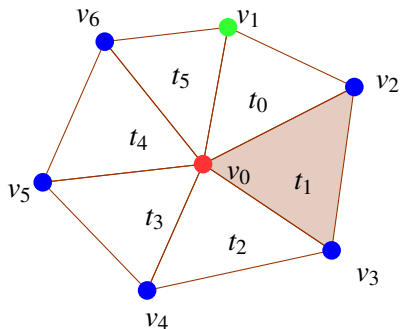
1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0
- 3 Encontre v_0 na face t
- 4 Próximo vértice a partir de v_0



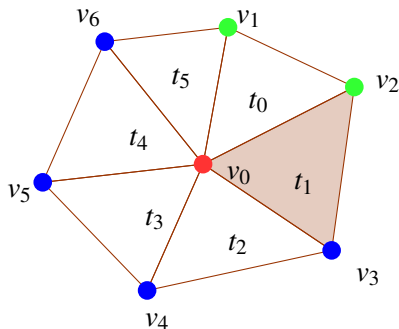
1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0
- 3 Encontre v_0 na face t
- 4 Próximo vértice a partir de v_0
- 5 Seja t a face oposta a v_1



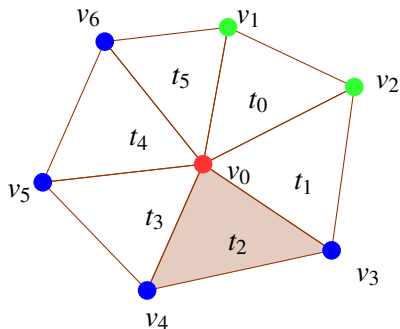
1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0
- 3 Encontre v_0 na face t
- 4 Próximo vértice a partir de v_0
- 5 Seja t a face oposta a v_1
- 6 Voltar ao passo 3 ...



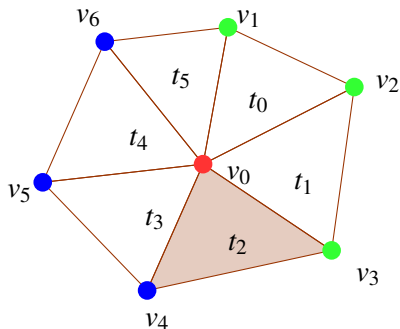
1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0
- 3 Encontre v_1 na face t
- 4 Próximo vértice a partir de v_0
- 5 Seja t a face oposta a v_1
- 6 Voltar ao passo 3 ...



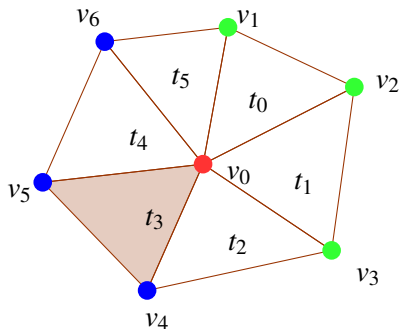
1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0
- 3 Encontre v_1 na face t
- 4 Próximo vértice a partir de v_0
- 5 Seja t a face oposta a v_1
- 6 Voltar ao passo 3 ...



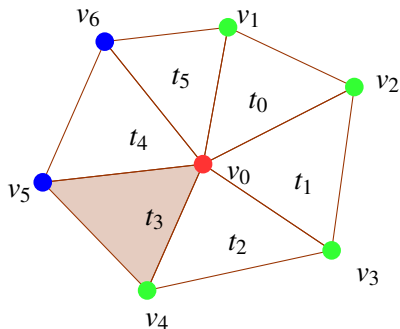
1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0
- 3 Encontre v_0 na face t
- 4 Próximo vértice a partir de v_0
- 5 Seja t a face oposta a v_1
- 6 Voltar ao passo 3 ...



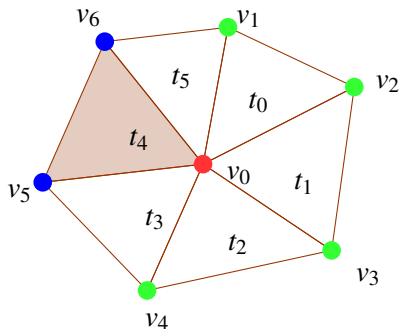
1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0
- 3 Encontre v_0 na face t
- 4 Próximo vértice a partir de v_0
- 5 Seja t a face oposta a v_1
- 6 Voltar ao passo 3 ...



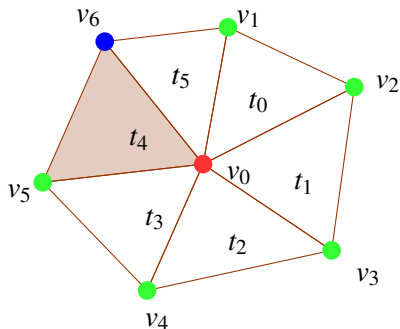
1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0
- 3 Encontre v_1 na face t
- 4 Próximo vértice a partir de v_0
- 5 Seja t a face oposta a v_1
- 6 Voltar ao passo 3 ...



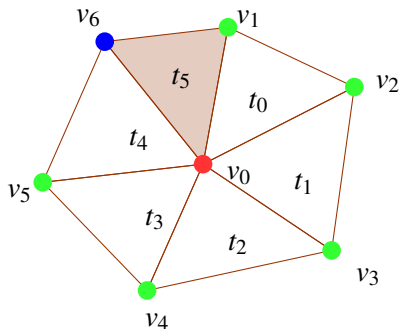
1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0
- 3 Encontre v_1 na face t
- 4 Próximo vértice a partir de v_0
- 5 Seja t a face oposta a v_1
- 6 Voltar ao passo 3 ...



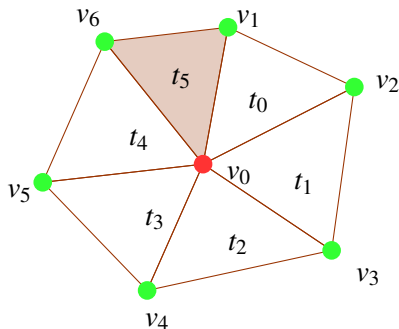
1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0
- 3 Encontre v_1 na face t
- 4 Próximo vértice a partir de v_0
- 5 Seja t a face oposta a v_1
- 6 Voltar ao passo 3 ...



1 – Anel com a Opposite Face

- 1 Inicia em um vértice
- 2 Seja t , uma face incidente em v_0
- 3 Encontre v_1 na face t
- 4 Próximo vértice a partir de v_0
- 5 Seja t a face oposta a v_1
- 6 Voltar ao passo 3 ...



FILM

The image features the word "FILM" in a bold, sans-serif font. Each letter is filled with a different color gradient: 'F' is a gradient from magenta to red; 'I' is a solid yellow; 'L' is a gradient from green to blue; and 'M' is a gradient from blue to purple. The letters are rendered with a slight 3D effect, casting soft, grey shadows to the left and slightly forward on the white background.