

Geometria Computacional

Professor:

Anselmo Montenegro
www.ic.uff.br/~anselmo

Conteúdo (aula 8):

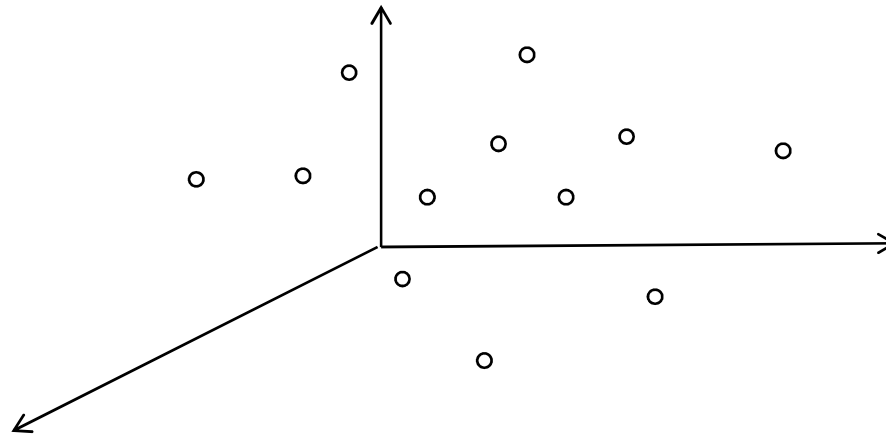
- Fecho Convexo 3D

Roteiro

- Introdução
- Algoritmo EmbrulhoParaPresente3D
- Análise de Complexidade de EmbrulhoParaPresente3D

- Fecho convexo3D: introdução

- Consideremos agora o problema de determinar o fecho convexo de um conjunto de pontos no \mathbb{R}^3



- Usaremos ideias semelhantes as utilizadas para determinar o fecho convexo 2D do algoritmo para embrulho para presente bidimensional

- **Fecho convexo3D: Algoritmo do Embrulho para Presente**

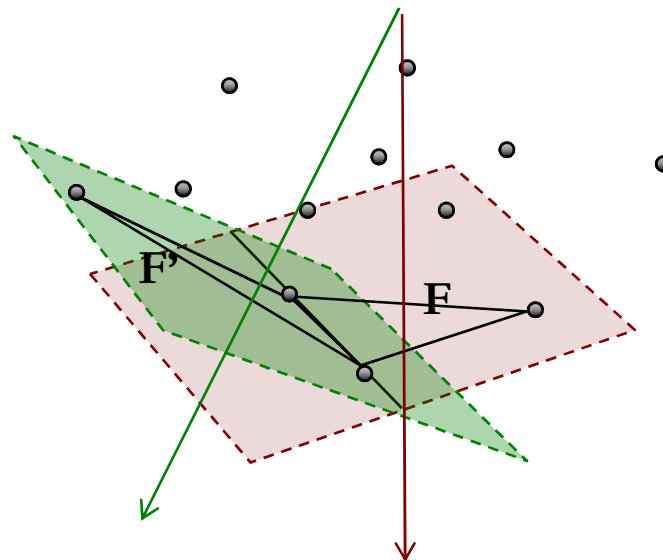
- O algoritmo parte do pressuposto que uma face F de $\text{conv}(S)$ foi encontrada
- Consideramos uma face F **explorada** se todas as **suas adjacentes já estiverem sido determinadas**
- Caso contrário, é necessário determinar as faces adjacentes a F em $\text{conv}(S)$

- **Fecho convexo3D: Algoritmo do Embrulho para Presente**

- O algoritmo Embrulho para Presente 3D constrói a fronteira P de $\text{conv}(S)$ gerando todas as faces adjacentes às já criadas até que todas tenham sido exploradas
- Seja $\mathbf{a} = p_1p_2$ uma aresta de F tal que a face adjacente a F em \mathbf{a} é desconhecida.
- A aresta \mathbf{a} é considerada uma **aresta livre**

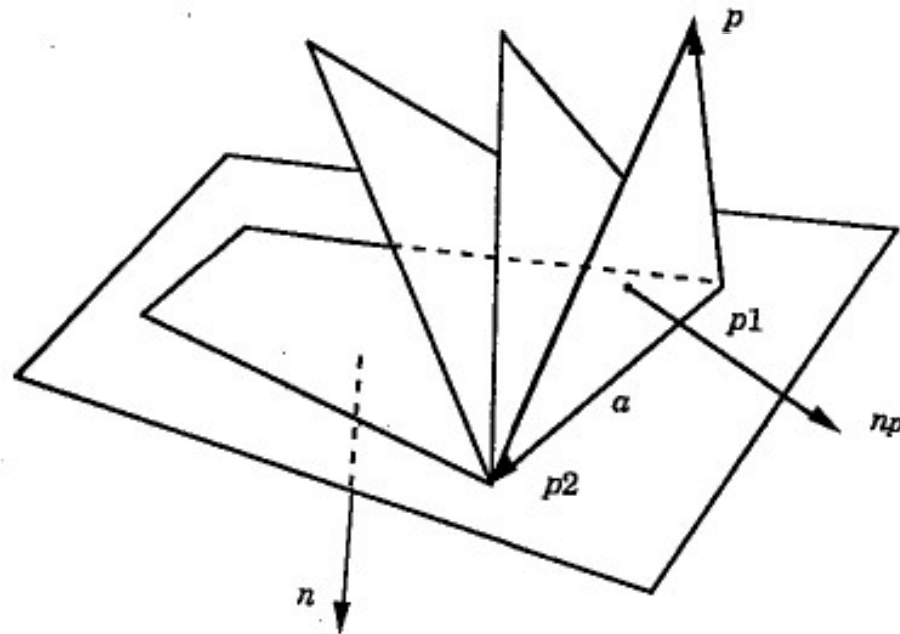
- Fecho convexo3D: Algoritmo do Embrulho para Presente

- Como F é uma face de P , todos os pontos de S estão situados em um dos semi-espacos definidos por F
- A face F' adjacente a F em a é tal que todos os pontos de S estão no ângulo diedral definido pelos planos de F e F'

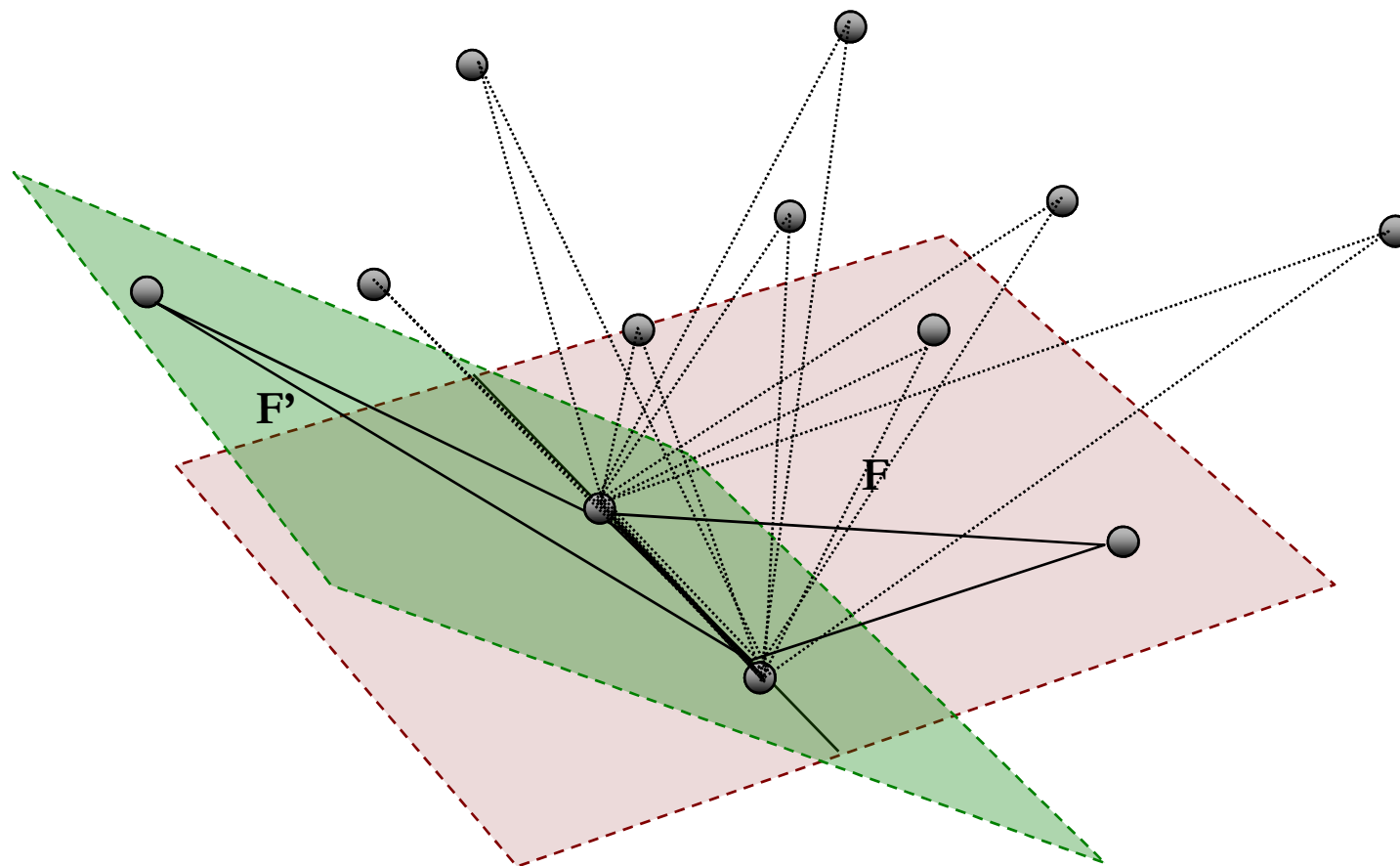


- Fecho convexo3D: Algoritmo do Embrulho para Presente

- Dentre todos os semiplanos determinados por a e pontos de S , a face F' está contida naquele que forma o ângulo máximo como o semiplano conduzido por a e contendo F

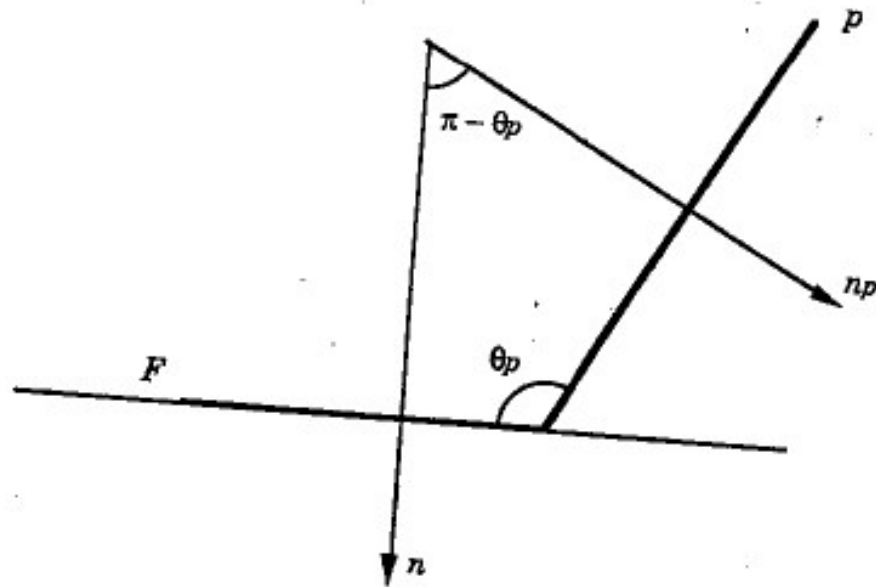


- Fecho convexo3D: Algoritmo do Embrulho para Presente



- Fecho convexo3D: Algoritmo Embrulho para Presente 3D

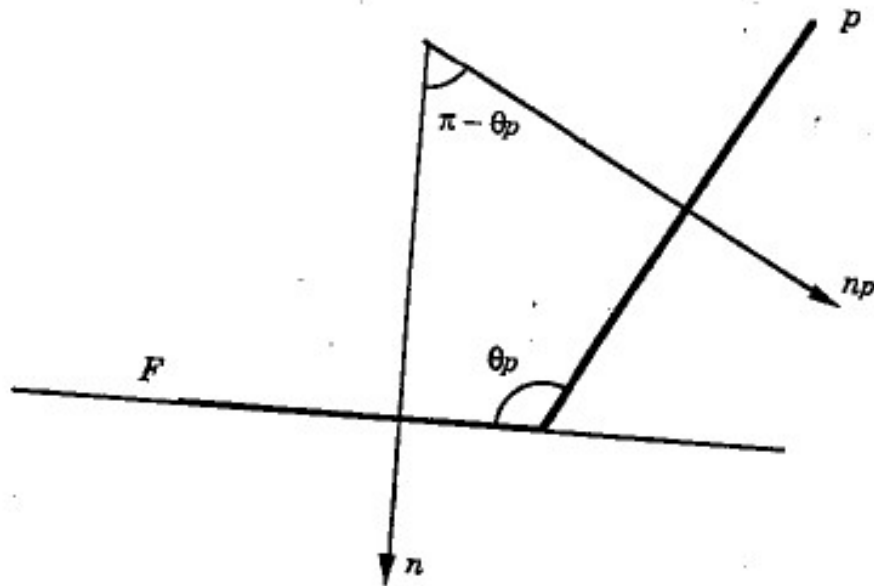
- Para cada ponto p de S precisamos determinar o ângulo $\theta_p \in [0, \pi]$ que o semiplano $\alpha(a, p)$ determinado por a e p forma com o semiplano F
- Observamos que o ângulo entre os vetores n e n_p , normais a F e $\alpha(a, p)$, respectivamente, é igual a $\pi - \theta_p$



- Fecho convexo3D: Algoritmo Embrulho para Presente 3D

- Logo:
$$\cos \theta_p = -\frac{n_p \cdot n}{\|n_p\| \|n\|}$$

- Para maximizar o ângulo entre F e $\alpha(a,p)$ devemos minimizar o $\cos \theta_p$



- **Fecho convexo3D: Algoritmo Embrulho para Presente 3D**

- Temos duas situações:
 - a. Se o mínimo ocorre em um único ponto, a face F' é dada pelos vértices p_1 , p_2 e p
 - b. Se o mínimo ocorrerem mais de um ponto p , é necessário calcular o fecho convexo 2D entre p_1 , p_2 e todos os pontos p , onde ocorre o mínimo

- **Fecho convexo3D: Algoritmo Embrulho para Presente 3D**

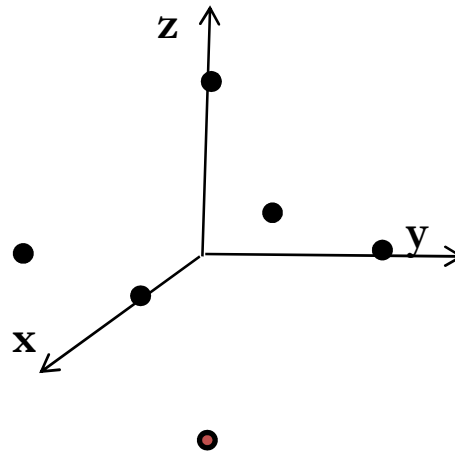
- Denominamos os passos acima, que expandem a descrição do fecho a partir de uma face, cujas vizinhas não foram determinadas de **EmbrulhoParaPresente(F, a)**

- **EmbrulhoParaPresente(F, a)** recebe uma face F e uma de suas arestas a e retorna uma face F' na fronteira de $\text{conv}(S)$ adjacente a a

- **Fecho convexo3D: Algoritmo Embrulho para Presente 3D**

- Como gerar a primeira face:

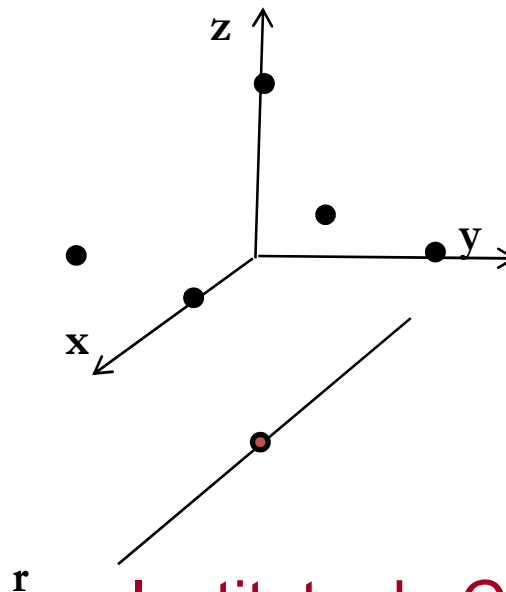
1. *Determinar o ponto p_1 de coordenada z mínima*



- Fecho convexo3D: Algoritmo Embrulho para Presente 3D

- Como gerar a primeira face:

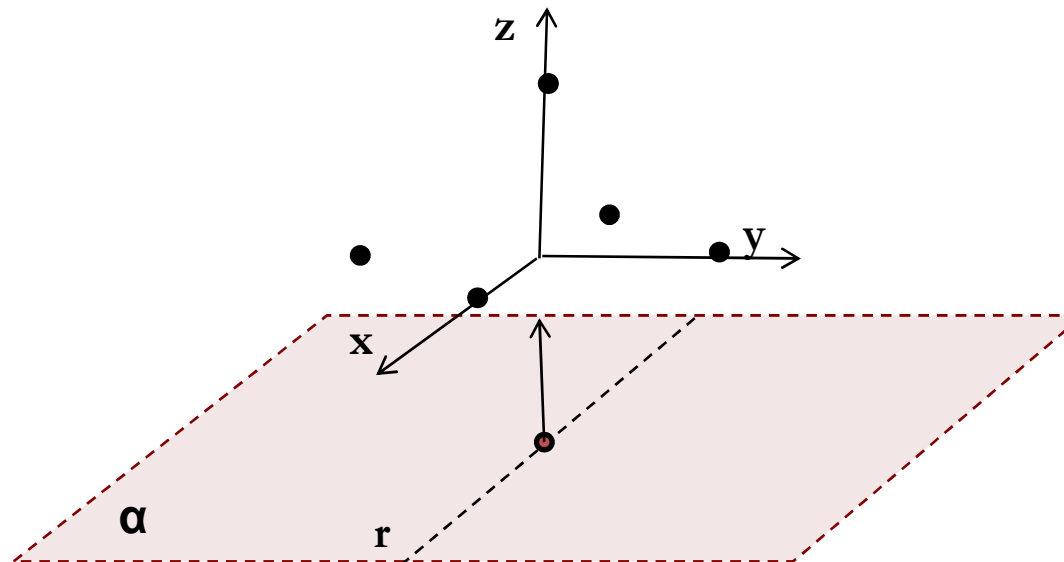
1. *Determinar o ponto $p1$ de coordenada z mínima*
2. *Determinar a reta r paralela a x e passando por $p1$*



- Fecho convexo3D: Algoritmo Embrulho para Presente 3D

- Como gerar a primeira face:

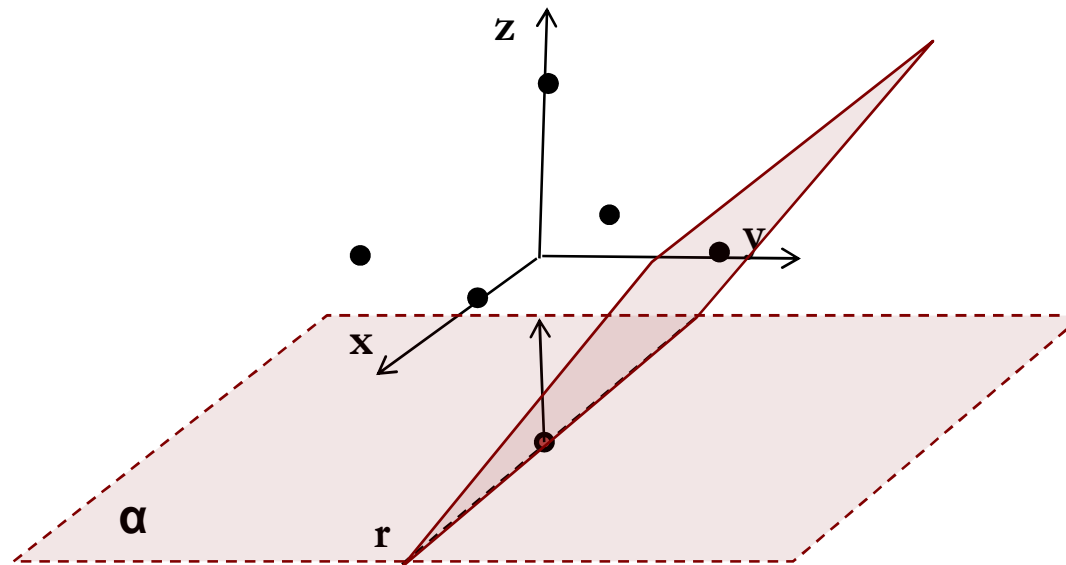
1. *Determinar o ponto p_1 de coordenada z mínima*
2. *Determinar a reta r paralela a x e passando por p_1*
3. *Determinar o plano α limitado por r e orientado pelo eixo y*



- Fecho convexo3D: Algoritmo Embrulho para Presente 3D

- Como gerar a primeira face:

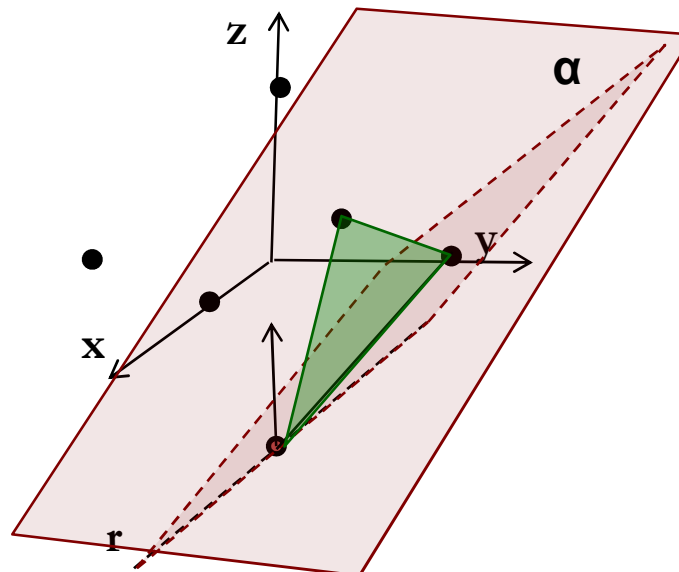
1. *Determinar o ponto $p1$ de coordenada z mínima*
2. *Determinar a reta r paralela a x e passando por $p1$*
3. *Determinar o plano α limitado por r e orientado pelo eixo y*
4. *Girar α em torno de r até tocar um ponto $p2$ de S*



- Fecho convexo3D: Algoritmo Embrulho para Presente 3D

- Como gerar a primeira face:

1. *Determinar o ponto p_1 de coordenada z mínima*
2. *Determinar a reta r paralela a x e passando por p_1*
3. *Determinar o plano α limitado por r e orientado pelo eixo y*
4. *Girar α em torno de r até tocar um ponto p_2 de S*
5. *Girar α em torno de p_1p_2 até tocar um ponto p_3 de S*



- **Fecho convexo3D: Algoritmo Embrulho para Presente 3D**

- O método anterior pode ser implementado usando EmbrulhoParaPresente(F,a) usando faces provisórias introduzindo vértices artificiais

$F \leftarrow (p1, p1+(1,0,0), p1+(0,-1,0)); a \leftarrow (p1, p1+(1,0,0))$
 $p2 \leftarrow \text{EmbrulhoParaPresente}(F, a)$
 $F \leftarrow (p2, p1, p1-(-1,0,0)); a \leftarrow (p2, p1)$
 $p3 \leftarrow \text{EmbrulhoParaPresente}(F, a)$

- **Fecho convexo3D: Algoritmo Embrulho para Presente 3D**

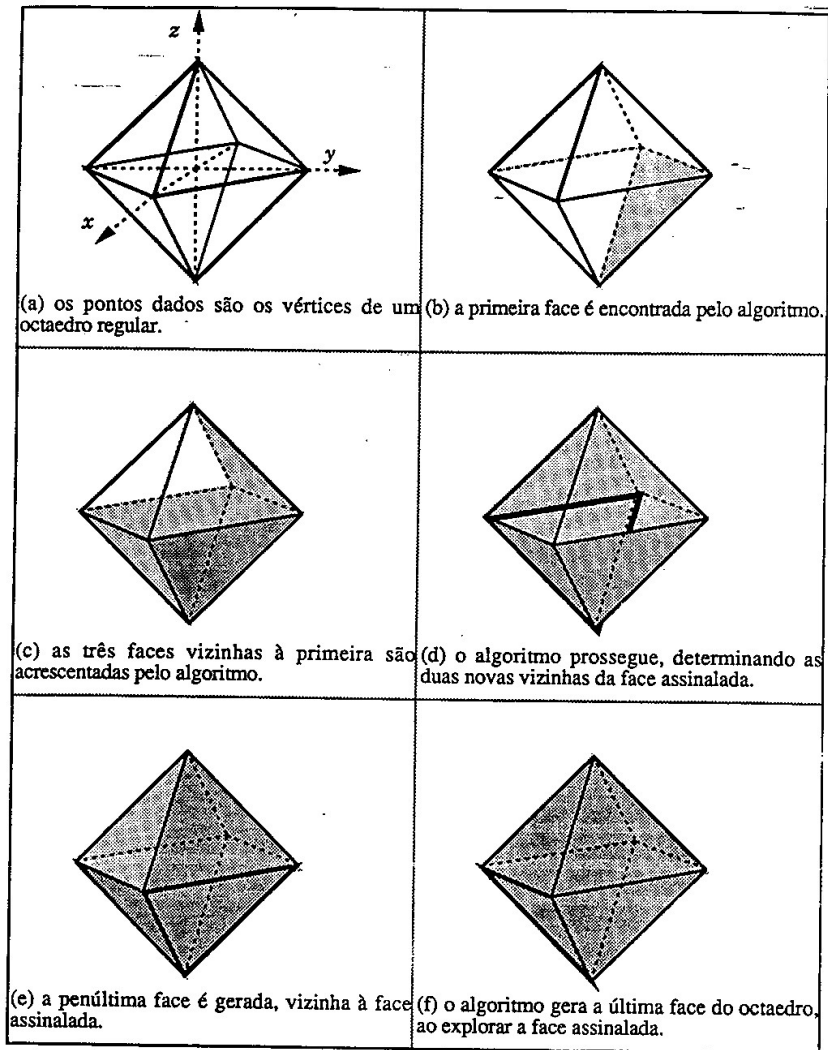
Algoritmo EmbrulhoParaPresente3D(S)

Entrada: um conjunto de pontos $S \in \mathbb{R}^3$

Saída: $\text{conv}(S)$

1. Obtenha uma face inicial F para $\text{conv}(S)$ aplicando a técnica descrita no slide anterior.
2. Coloque F em \mathcal{F} e na estrutura half-edge, com todas as suas arestas rotuladas como livre.
3. **Enquanto** $\mathcal{F} \neq \emptyset$ **faça**
4. Remova uma face F de \mathcal{F}
5. **Para** cada aresta a livre de F **faça**
6. Determine a face adjacente F' usando $\text{EmbrulhoParaPresente}(F, a)$
7. Coloque F' gerada na fila \mathcal{F}
8. Coloque F' na estrutura half-edge, conectando-a com as faces já geradas que lhe são adjacentes e determinando suas arestas livres
9. **Fim_Para**
10. **Fim_Enquanto**

- Fecho convexo3D: Algoritmo Embrulho para Presente 3D



- **Fecho convexo3D: Algoritmo Embrulho para Presente 3D-Análise de Complexidade**

- A complexidade do algoritmo é dominada pelo passo 5 e 7.
- O passo 5 é executado para cada aresta de $\text{conv}(S)$. Como o tempo de execução de $\text{EmbrulhoParaPresente}(\dots)$ é $O(n)$ e o número de arestas é também $O(n)$, tem-se que o passo 5 é $O(n^2)$
- Cada vez que uma face é criada, no passo 7, é necessário percorrer a estrutura de dados para verificar se suas arestas já foram criadas.
- Isso requer tempo $O(n)$ na estrutura para cada aresta o que leva a um total de $O(n^2)$
- **Observe que não consideramos o caso de ter que calcular fechos convexos 2D, mas isto pode ser feito em tempo $O(n \log n)$ o que não altera a complexidade**

- **Fecho convexo3D: Algoritmo Embrulho para Presente 3D-Análise de Complexidade**

- Teorema 6a.1. O algoritmo “EmbrulhoParaPresente3D determina o fecho convexo de um conjunto de pontos n pontos no R^3 em tempo $O(n^2)$

Fecho Convexo3D: Referências

- P. C. P. Carvalho e L. H. de Figueiredo, *Introdução à Geometria Computacional*, 18° Colóquio Brasileiro de Matemática, IMPA, 1991.
- F. P. Preparata e M. I. Shamos, *Computational Geometry: an Introduction*, Springer-Verlag, 1985.