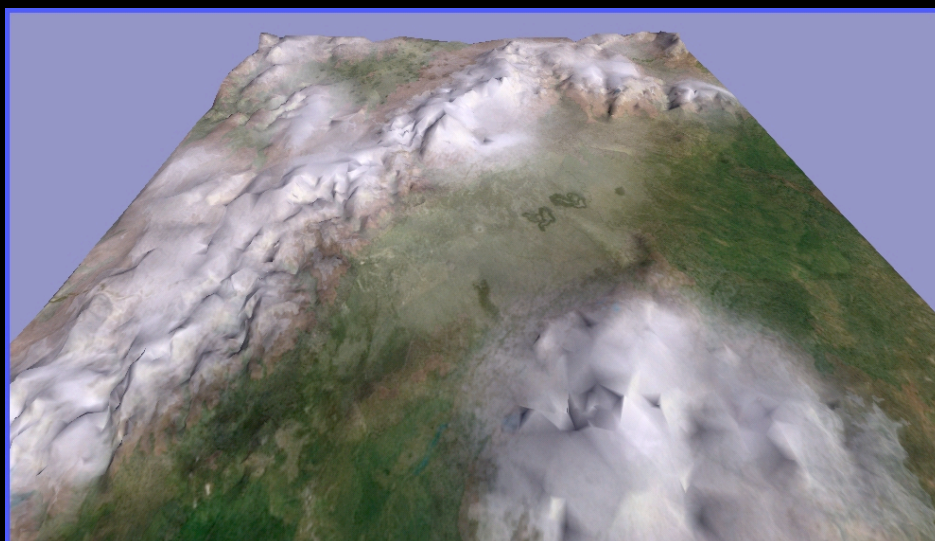


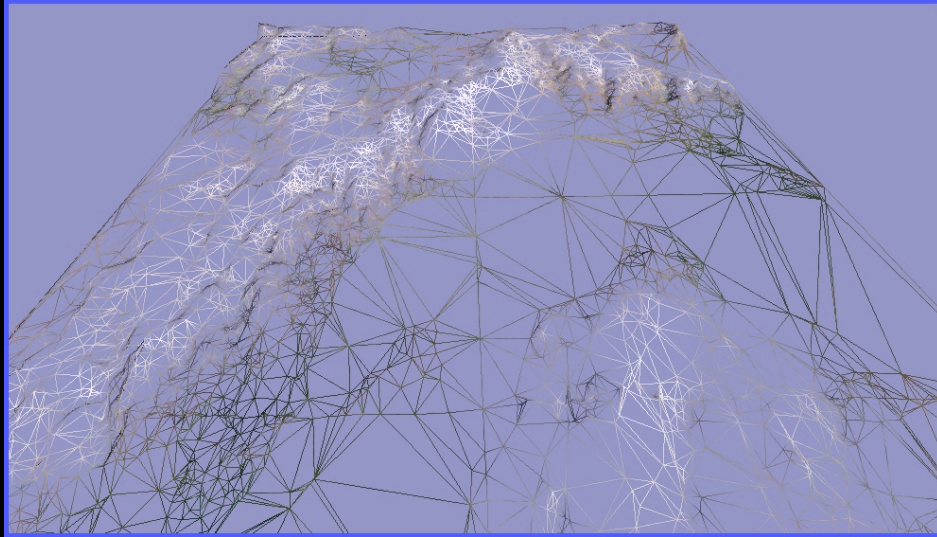
# TRIANGULAÇÕES DE DELAUNAY

JOÃO COMBA

Terrenos



# Terrenos



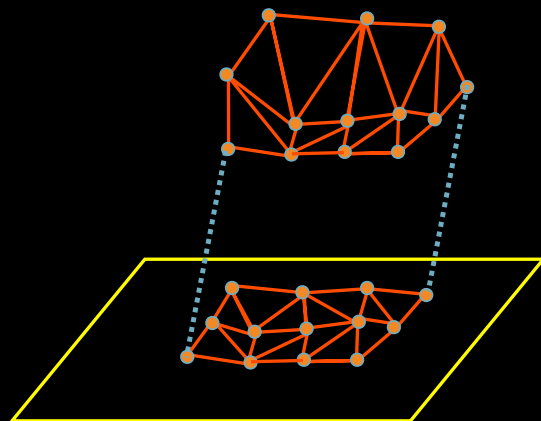
# Terrenos

- Conjunto  $P$  de pontos em 2D com altura ( $2^{1/2}D$ )

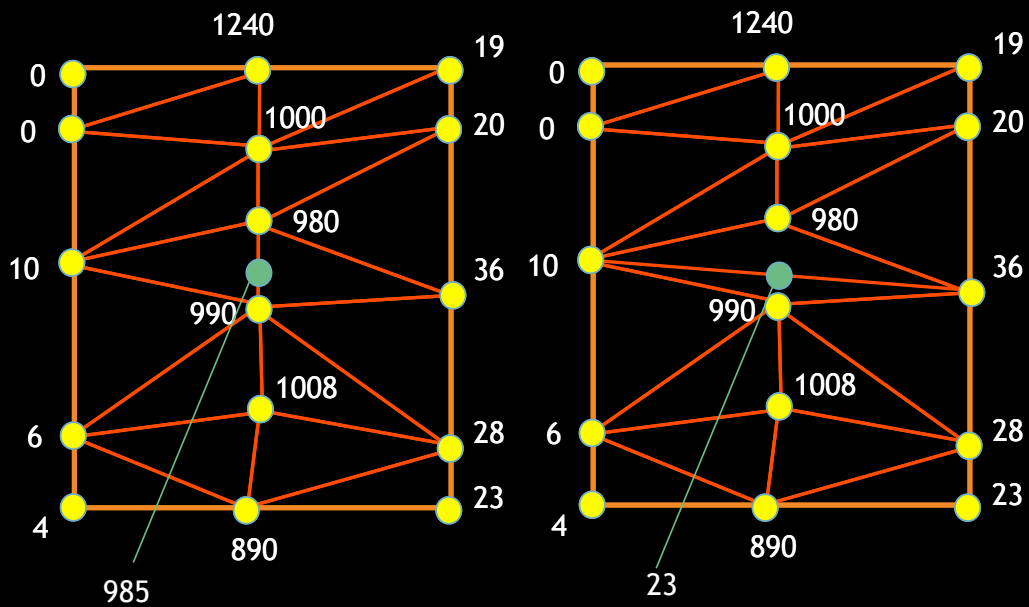
- Nem todos pontos

- Triangulacao de  $P$

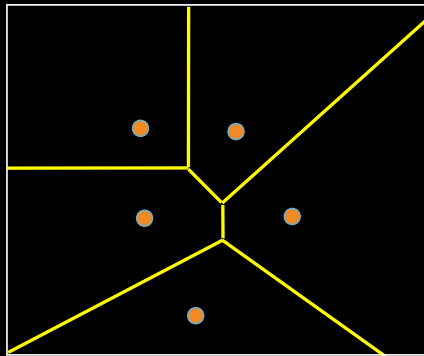
- Aproximar altura de pontos pelos vizinhos
- Triangulos com vertices definidos por  $P$
- Triangulacao no plano elevada para cada ponto pela altura
- Qual a melhor ?



## Terrenos

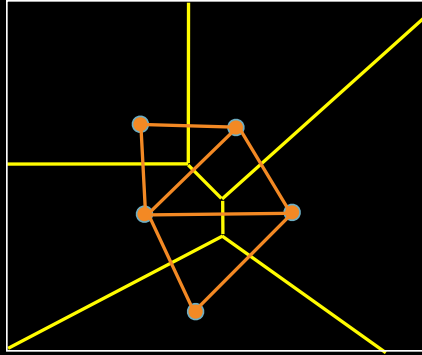


## Triangulacoes de Delaunay



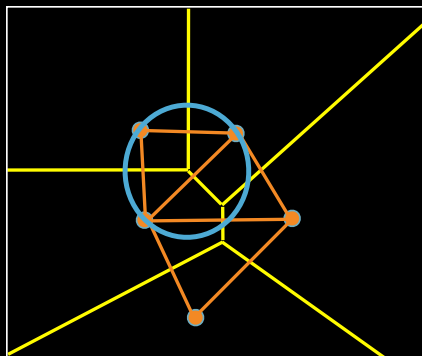
# Triangulacoes de Delaunay

Conectar sitios cujas celulas sao vizinhas



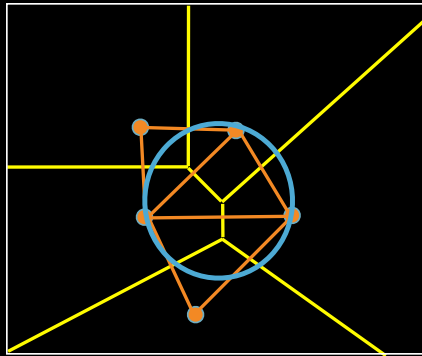
# Triangulacoes de Delaunay

O circuncirculo de cada triangulo nao possui nenhum vertice !



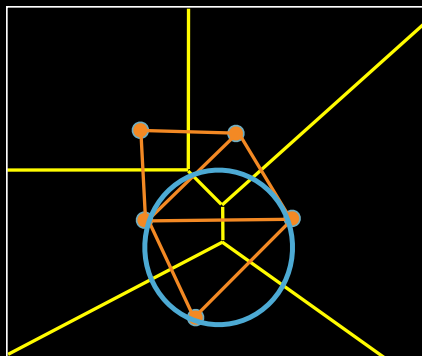
# Triangulacoes de Delaunay

O circuncirculo de cada triangulo nao possui nenhum vertice !



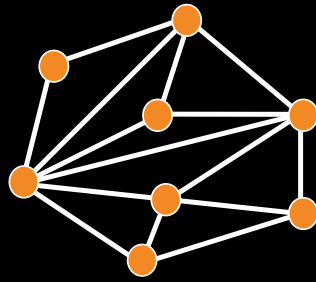
# Triangulacoes de Delaunay

O circuncirculo de cada triangulo nao possui nenhum vertice !



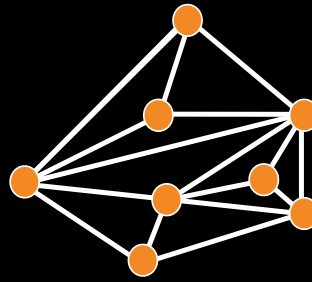
## Número de Triângulos

O número de triângulos de uma triangulação de  $n$  vértices depende do número de vértices  $h$  na envoltória convexa.



$$h = 6 \rightarrow t = 8$$

$$n = 8$$



$$h = 5 \rightarrow t = 9$$

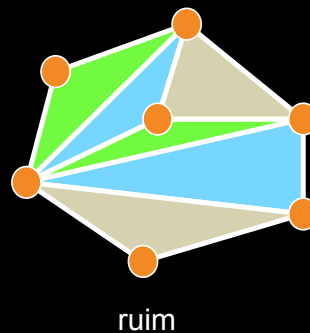
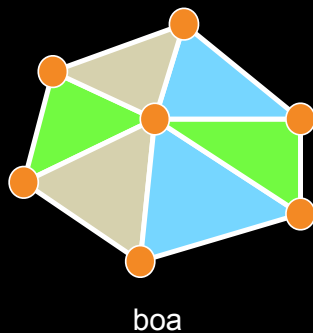
$$t = 2n - h - 2$$

## Qualidade de Triangulações

- Seja  $\alpha(T) = (\alpha_1, \alpha_2, \dots, \alpha_{3t})$  o vetor de ângulos de uma triangulação  $T$  em ordem crescente.

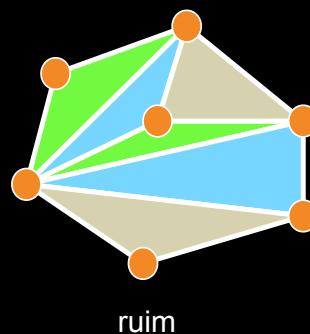
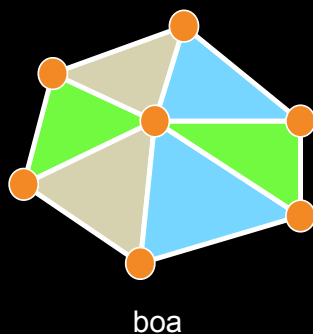
## Qualidade de Triangulações

- Seja  $\alpha(T) = (\alpha_1, \alpha_2, \dots, \alpha_{3t})$  o vetor de ângulos de uma triangulação  $T$  em ordem crescente.
- Uma triangulação  $T_1$  é “melhor” que  $T_2$  se  $\alpha(T_1) > \alpha(T_2)$  lexicograficamente.



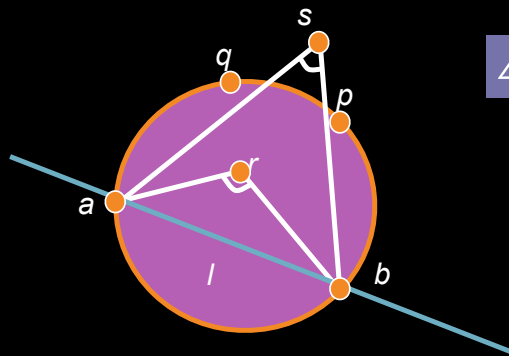
## Qualidade de Triangulações

- Seja  $\alpha(T) = (\alpha_1, \alpha_2, \dots, \alpha_{3t})$  o vetor de ângulos de uma triangulação  $T$  em ordem crescente.
- Uma triangulação  $T_1$  é “melhor” que  $T_2$  se  $\alpha(T_1) > \alpha(T_2)$  lexicograficamente.
- A Triangulação de Delaunay é a melhor de todas



## Teorema de Thales

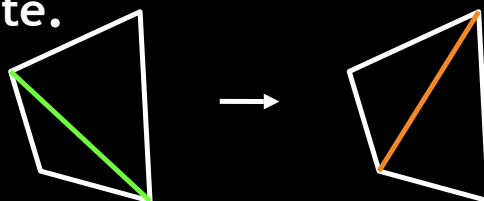
Seja  $C$  um círculo, e  $l$  uma linha cortando  $C$  em pontos  $a$  e  $b$ . Seja  $p, q, r$  e  $s$  pontos no mesmo lado de  $l$ , com  $p$  e  $q$  sobre  $C$ ,  $r$  dentro de  $C$  e  $s$  fora de  $C$ . Então:



$$\angle arb > \angle apb = \angle aqb > \angle asb$$

## Melhorando uma triangulação

Em qualquer quadrilátero, um **flip de arestas** é possível. Se este flip melhorar a triangulação localmente, então ele melhora a triangulação globalmente.



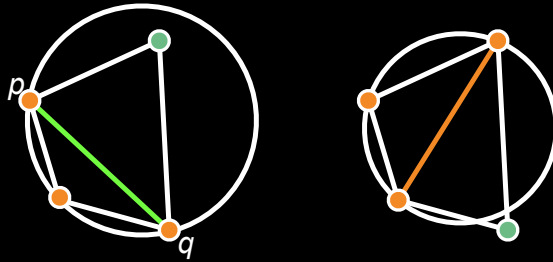
Se um flip de aresta melhora uma triangulação, então a primeira aresta é considerada **ilegal**.



## Arestas Ilegais

**Lema:** Uma aresta  $pq$  é ilegal se e somente se um dos seus vértices opostos está contido no círculo definido pelos outros dois

**Prova:** Usando teorema de Thales.

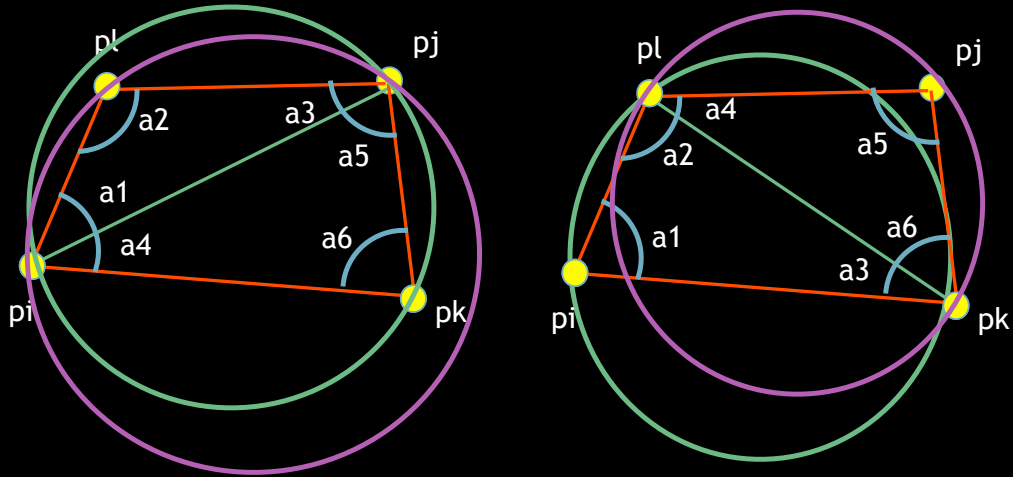


## Teoremas de Triangulações de Delaunay

**TEOREMA:** Seja  $P$  um conjunto de pontos:

- 3 pontos  $p_i, p_j, p_k$  de  $P$  são vértices de uma mesma face da triangulação de Delaunay se e somente se o círculo  $p_i, p_j, p_k$  não contém nenhum ponto no seu interior
- 2 pontos  $p_i, p_j$  formam uma aresta da triangulação de Delaunay se e somente se existe um disco fechado  $C$  que contém  $p_i$  e  $p_j$  e mais nenhum ponto

## Triangulacoes de Delaunay



## Teoremas de Triangulações de Delaunay

**TEOREMA:** Seja  $P$  um conjunto de pontos, e  $T$  uma triangulacao de  $P$ .  $T$  e' uma triangulacao de Delaunay de  $P$  se e somente se o "circuncirculo" de cada triangulo de  $T$  nao possui nenhum ponto de  $P$  no seu interior

## Teoremas de Triangulações de Delaunay

**TEOREMA:** Seja  $P$  um conjunto de pontos no plano. Uma triangulação  $T$  de  $P$  é legal se e somente se  $T$  é a triangulação de Delaunay de  $P$

## Algoritmos de Construção

## Teste Dentro do Círculo (In-Circle)

**Teorema:** Se  $a, b, c, d$  formam um polígono convexo em ordem anti-horária, então  $d$  pertence ao círculo se e somente se:

$$\det \begin{pmatrix} a_x & a_y & a_x^2 + a_y^2 & 1 \\ b_x & b_y & b_x^2 + b_y^2 & 1 \\ c_x & c_y & c_x^2 + c_y^2 & 1 \\ d_x & d_y & d_x^2 + d_y^2 & 1 \end{pmatrix} > 0$$

**Prova:** A igualdade é válida se os pontos são co-circulares. Existe um centro  $q$  em um raio  $r$  tal que:

$$(a_x - q_x)^2 + (a_y - q_y)^2 = r^2$$

e similarmente para  $b, c, d$ :

$$\begin{pmatrix} a_x^2 + a_y^2 \\ b_x^2 + b_y^2 \\ c_x^2 + c_y^2 \\ d_x^2 + d_y^2 \end{pmatrix} - 2q_x \begin{pmatrix} a_x \\ b_x \\ c_x \\ d_x \end{pmatrix} - 2q_y \begin{pmatrix} a_y \\ b_y \\ c_y \\ d_y \end{pmatrix} + (q_x^2 + q_y^2 - r^2) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = 0$$

Portanto estes vetores são linearmente dependentes, e o determinante é zero.

**Corolário:**  $d \in \text{circle}(a, b, c)$  sse  $b \in \text{circle}(a, c, d)$  sse  $c \in \text{circle}(b, a, d)$  sse  $a \in \text{circle}(b, c, d)$

## Algoritmo $O(n^4)$ para cálculo da Triangulação de Delaunay

**Repita até não ser mais possível melhorar:**

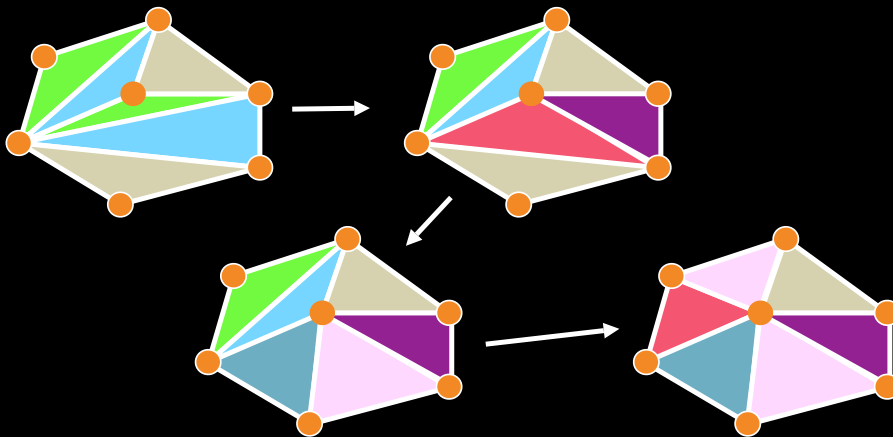
- Selecione 3 sites  $a, b, c$ .
- Se o circuncírculo por  $a, b$  e  $c$  não contém outros sítios, mantenha o triângulo definido por  $a, b$  e  $c$ .

## Algoritmo de Delaunay Simplex

Comece com uma triangulação arbitrária. Troque todas arestas ilegais até que nenhuma mais exista.

Requer prova que não existe mínimos locais.

Pode levar um grande tempo para terminar.



## Triangulação de Delaunay por Dualidade

Assuma posição geral: não existem quatro pontos co-circulares.

Crie o dual do Diagrama de Voronoi conectando dois sítios vizinhos no diagrama de Voronoi.

Corolário: A TD pode ser calculada em tempo  $O(n \log n)$ .



## Algoritmo $O(n \log n)$ para calcular Triangulação de Delaunay

**Algoritmo incremental randômico:**

Crie um triângulo que contenha todos os sítios.

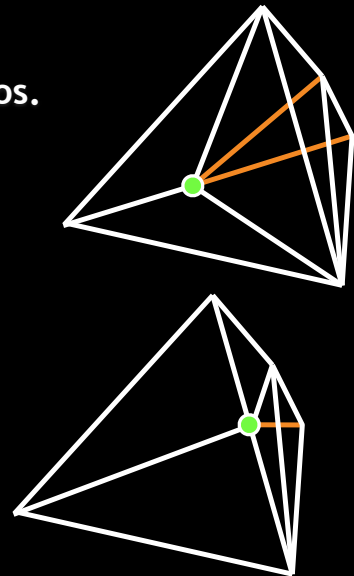
Adicione os sítios um depois do outro em ordem randômica.

**Se o sítio está dentro de um triângulo:**

- Conecte o sítio aos vértices do triângulo.
- Cheque se um 'flip' pode ser realizado em uma das arestas do triângulo. Caso positivo, cheque recursivamente as arestas vizinhas.

**Se o sítio cai sobre uma aresta:**

- Troque aresta por 4 arestas novas.
- Cheque se um 'flip' pode ser realizado em uma das arestas do triângulo. Caso positivo, cheque recursivamente as arestas vizinhas.



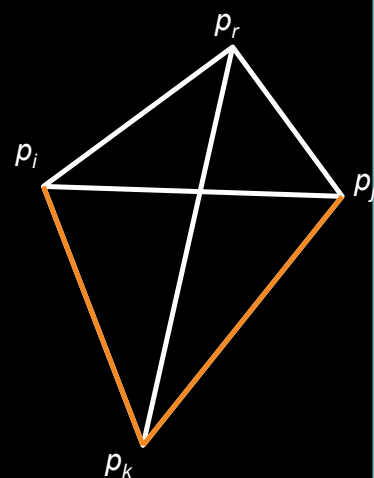
## Flip de Arestas

Uma nova aresta  $p_i p_j$ , foi criada oposta ao vértice  $p_k$ .

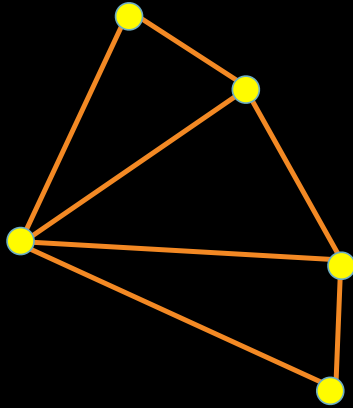
Cheque aresta  $p_i p_j$ . Se ilegal, faça o flip e recursivamente cheque arestas  $p_j p_k$  e  $p_i p_k$  - as novas arestas opostas ao novo vértice  $p_r$ .

Note que as chamadas recursivas para  $p_i p_k$  não eliminam a aresta  $p_j p_k$ .

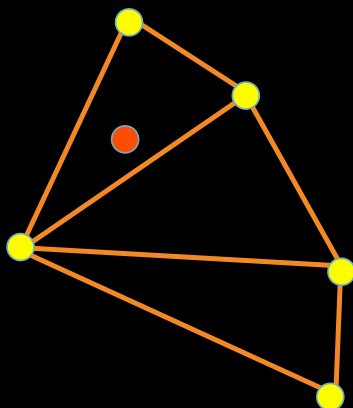
Nota: Todos flips de arestas envolvem somente o novo vértice !



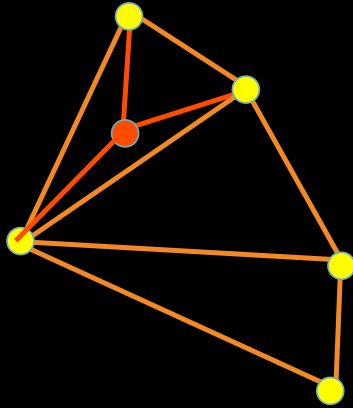
## Contruindo Triangulacoes de Delaunay



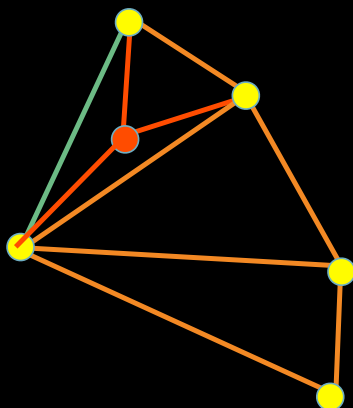
## Contruindo Triangulacoes de Delaunay



## Contruindo Triangulacoes de Delaunay

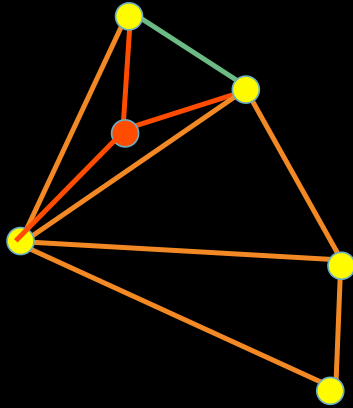


## Contruindo Triangulacoes de Delaunay

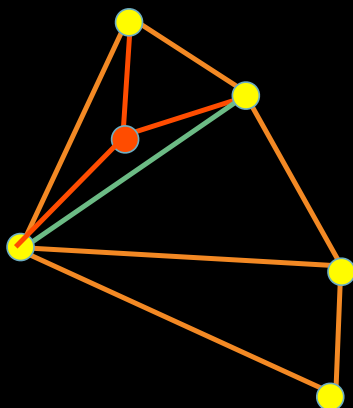




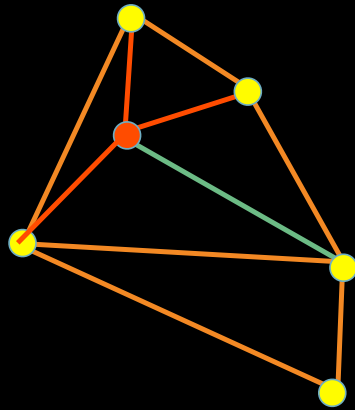
## Contruindo Triangulacoes de Delaunay



## Contruindo Triangulacoes de Delaunay

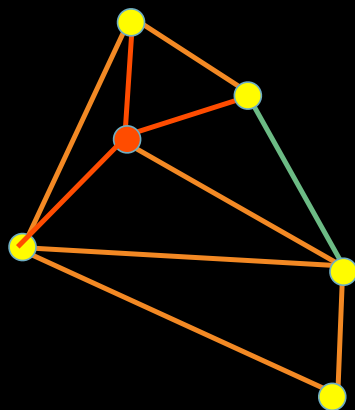


## Contruindo Triangulacoes de Delaunay

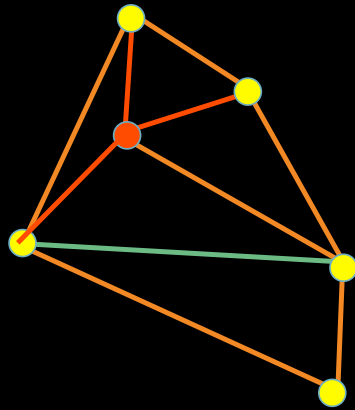


FLIP de aresta

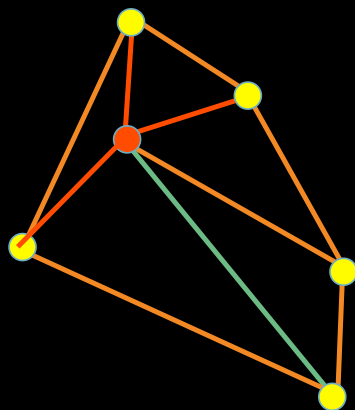
## Contruindo Triangulacoes de Delaunay



## Contruindo Triangulacoes de Delaunay



## Contruindo Triangulacoes de Delaunay



FLIP de aresta

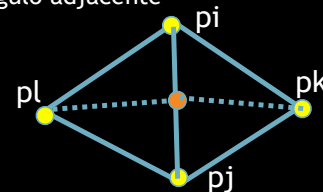
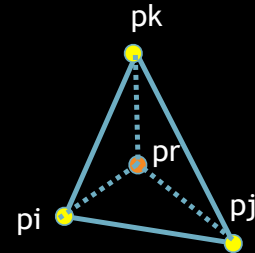
# Contruindo Triangulacoes de Delaunay

**ALGORITMO** TriangulacaoDelaunay(P)

**Entrada:** Conjunto P de n pontos

**Saida:** Triang. Delaunay de P

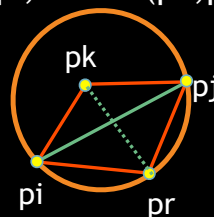
1. Seja  $p_a, p_b, p_c$  tres pontos que definem um triangulo contendo todos pontos de P
2. Inicialize T como uma triangulacao com um o triangulo  $p_a, p_b, p_c$
3. Calcule uma permutacao  $p_1, p_2, \dots, p_n$  de P
4. **FOR**  $r := 1$  TO n
5. **DO** // Inserir  $p_r$  em T
6.   Encontre um triangulo  $p_i p_j p_k$  que contem  $p_r$
7.   **IF**  $p_r$  esta no interior do triangulo  $p_i p_j p_k$
8.   **THEN** adicione arestas de  $p_r$  para  $p_i, p_j, p_k$ , criando tres triangulos
9.       LEGALIZEARESTA( $p_r$ , aresta( $p_i, p_j$ ), T)
10.       LEGALIZEARESTA( $p_r$ , aresta( $p_j, p_k$ ), T)
11.       LEGALIZEARESTA( $p_r$ , aresta( $p_k, p_i$ ), T)
12.   **ELSE** //  $p_r$  esta sobre uma aresta
13.       Adicione aresta de  $p_r$  para  $p_k$  e para o vertice  $p_l$  do triangulo adjacente
14.       LEGALIZEARESTA( $p_r$ , aresta( $p_i, p_l$ ), T)
15.       LEGALIZEARESTA( $p_r$ , aresta( $p_l, p_j$ ), T)
16.       LEGALIZEARESTA( $p_r$ , aresta( $p_j, p_k$ ), T)
17.       LEGALIZEARESTA( $p_r$ , aresta( $p_k, p_i$ ), T)
18.   Delete  $p_a, p_b, p_c$  e suas arestas incidentes de T



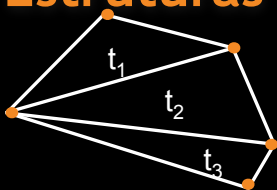
# Contruindo Triangulacoes de Delaunay

**LEGALIZEARESTA**( $p_r$ , aresta( $p_i, p_j$ ), T)

1. **IF** aresta( $p_i, p_j$ ) e' ilegal
2. **THEN** Seja  $p_i p_j p_k$  o triangulo adjacente a  $p_r p_i p_j$  adjacente a aresta( $p_i, p_j$ )
3.     Troque aresta( $p_i, p_j$ ) por ( $p_r, p_k$ )
4.     LEGALIZEARESTA ( $p_r$ , aresta( $p_i, p_k$ ), T)
5.     LEGALIZEARESTA ( $p_r$ , aresta( $p_k, p_j$ ), T)



## Estruturas de dados - Localização Pontual

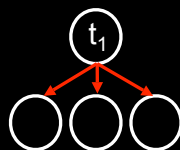
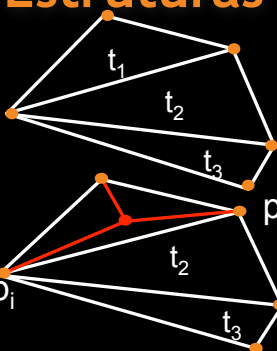


$t_1$

$t_2$

$t_3$

## Estruturas de dados - Localização Pontual



$t_1$

$t_2$

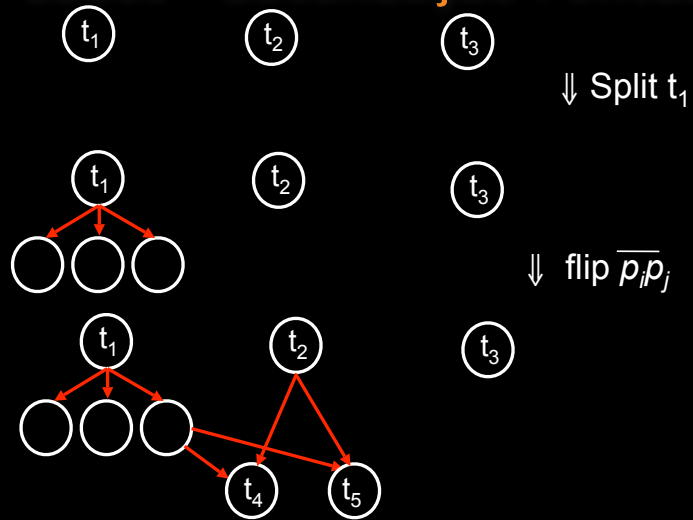
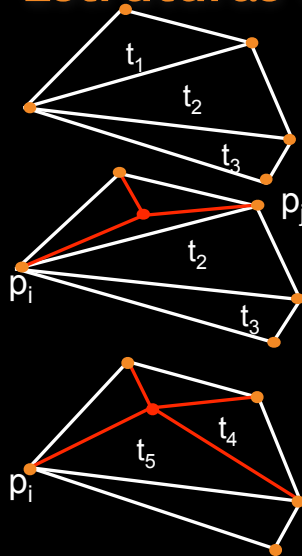
$t_3$

↓ Split  $t_1$

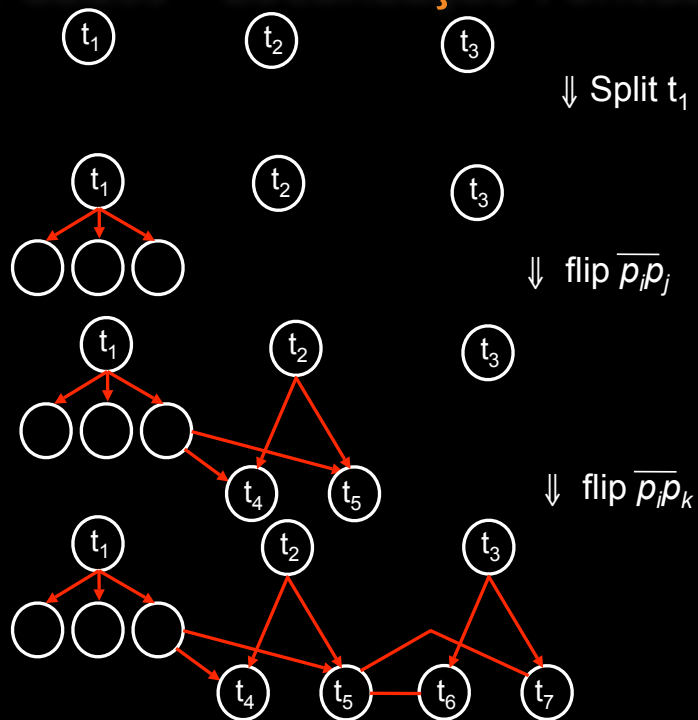
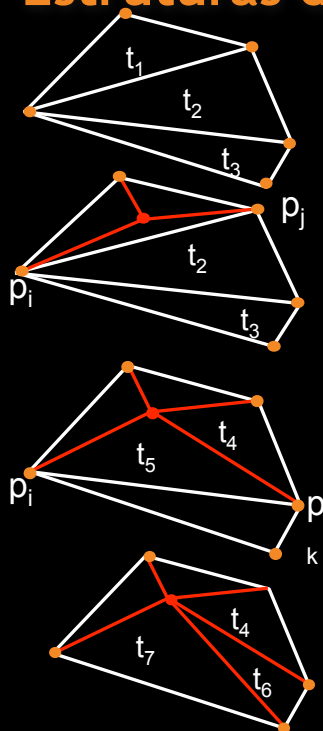
$t_2$

$t_3$

## Estruturas de dados - Localização Pontual



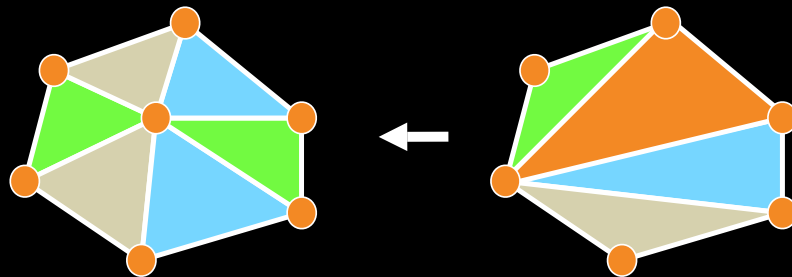
## Estruturas de dados - Localização Pontual



## Complexidade do Algoritmo

**Teorema:** O número esperado de triângulos criados (alguns desaparecem mais a frente)  $\leq 9n+1$ .

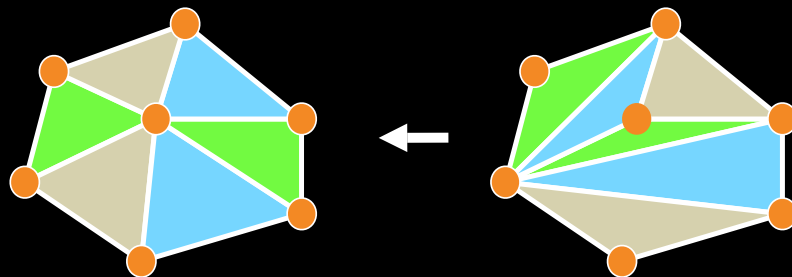
**Prova:** Durante a inserção de  $p_i$ ,  $k_i$  novas arestas são criadas (3 novas arestas, e  $k_i - 3$  flips)  $\rightarrow$  o grau de  $p_i$  é  $k_i \rightarrow$  o número de triângulos  $\leq 2(k_i - 3) + 3$



## Complexidade do Algoritmo(cont)

O diagrama de Voronoi diagram possui no máximo  $3n-6$  arestas. O número de arestas no grafo e no seu dual são idênticos.

Após inserir  $i$  sítios, existem no máximo  $3(i+3)-6 = 3i+3$  arestas (Na TD existem 3 vértices a mais que o DV)

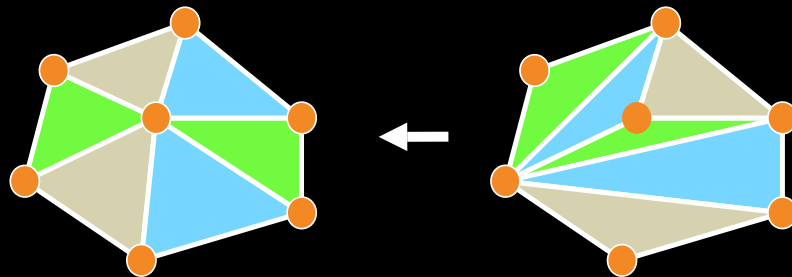


## Complexidade do Algoritmo (cont)

A soma de todos os vértices  $\leq 6i$ . Na média, o grau de cada vértice é 6.

$$E[\text{number of triangles created in stage } i] \leq E(2k_i - 3) = 2E(k_i) - 3 = 9$$

O número esperado de triângulos é  $9n+1$   
( $n$  estágios, e um triângulo externo)



## Complexidade do Algoritmo(cont)

Localização pontual para cada novo sítio  
custa  $O(\log n)$ .

No total  $O(n \log n)$ .

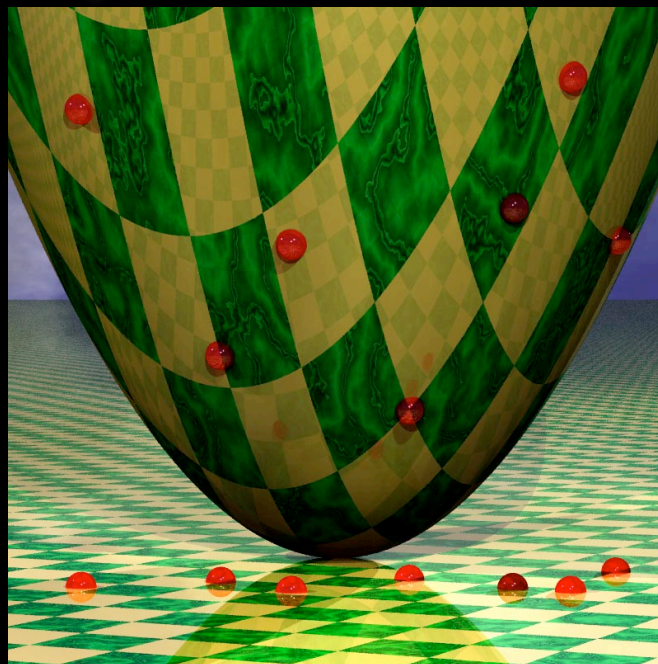


## Contruindo Triangulacoes de Delaunay

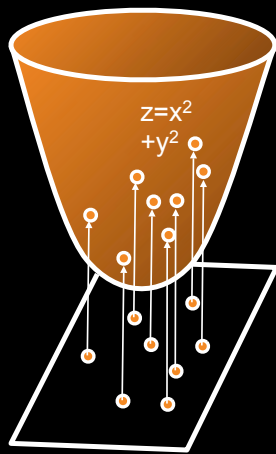
**LEMA:** O numero esperado de triangulos criados pelo algoritmo e' no maximo  $9n+1$

**TEOREMA:** A triangulacao de Delaunay de um conjunto de pontos no plano pode ser calculada em tempo esperado  $O(n \log n)$ , usando  $O(n)$  de memoria

## Lifting Map

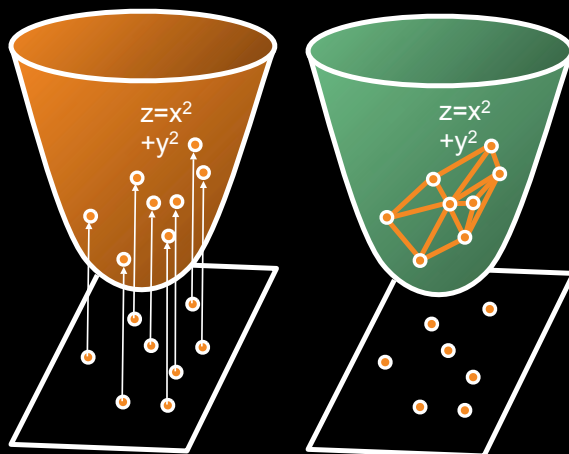


## Triangulação de Delaunay e Envoltória Convexa



Projete os pontos 2D  
sobre o parabolóide 3D

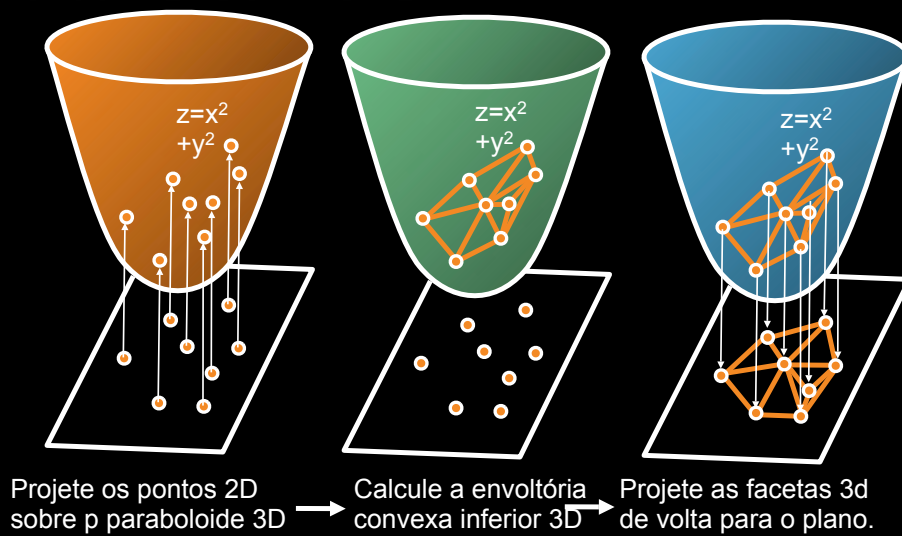
## Triangulação de Delaunay e Envoltória Convexa



Projete os pontos 2D  
sobre o parabolóide 3D

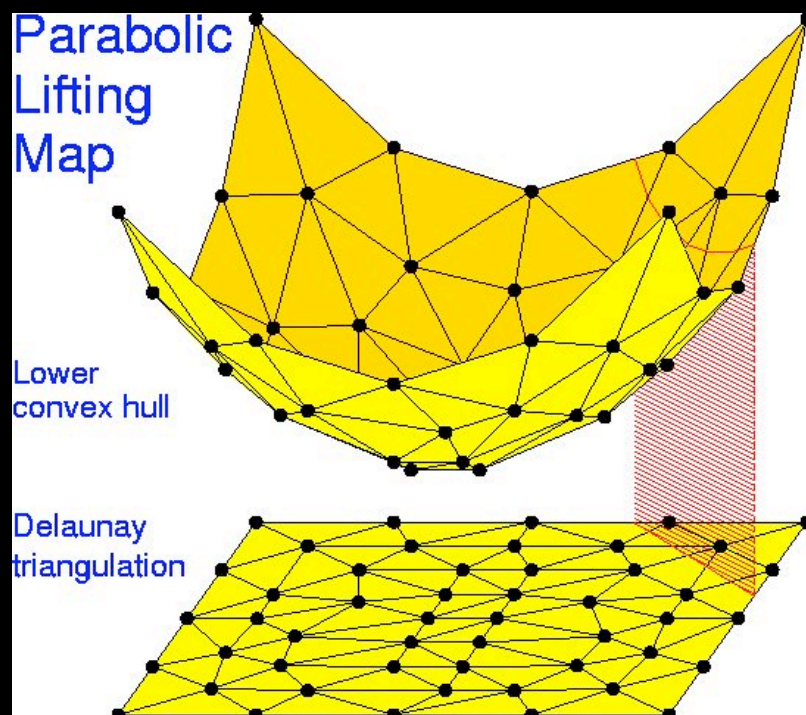
→ Calcule a envoltória  
convexa inferior 3D

## Triangulação de Delaunay e Envoltória Convexa



A triangulação 2D é Delaunay !

## Lifting Map

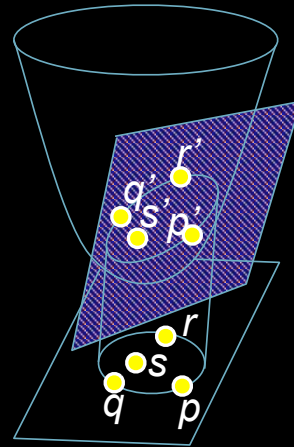


## Proof

The intersection of a plane with the paraboloid is an ellipse whose projection to the plane is a circle.

$s$  lies within the circumcircle of  $p, q, r$  iff  $s'$  lies on the lower side of the plane passing through  $p', q', r'$ .

$p, q, r \in S$  form a Delaunay triangle iff  $p', q', r'$  form a face of the convex hull of  $S'$ .



55

## Lifting Map: Magic

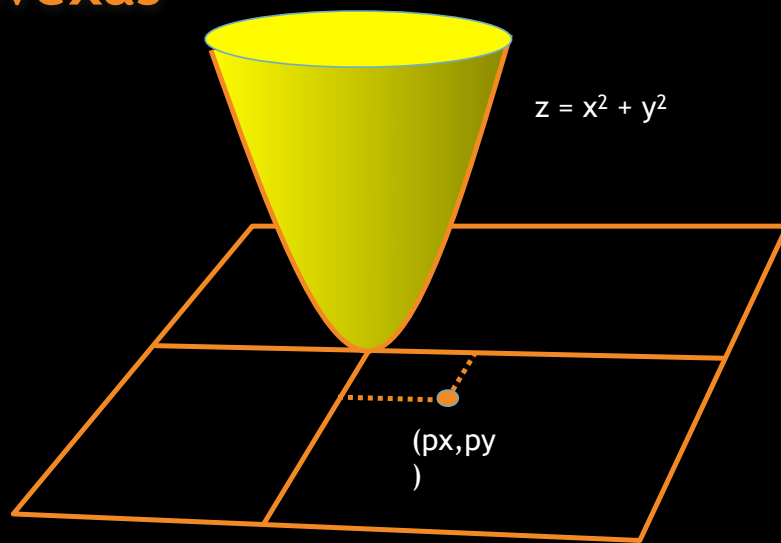
Map  $(p_x, p_y) \mapsto (p_x, p_y, p_x^2 + p_y^2)$

Map Convex Hull back  $\rightarrow$  Delaunay

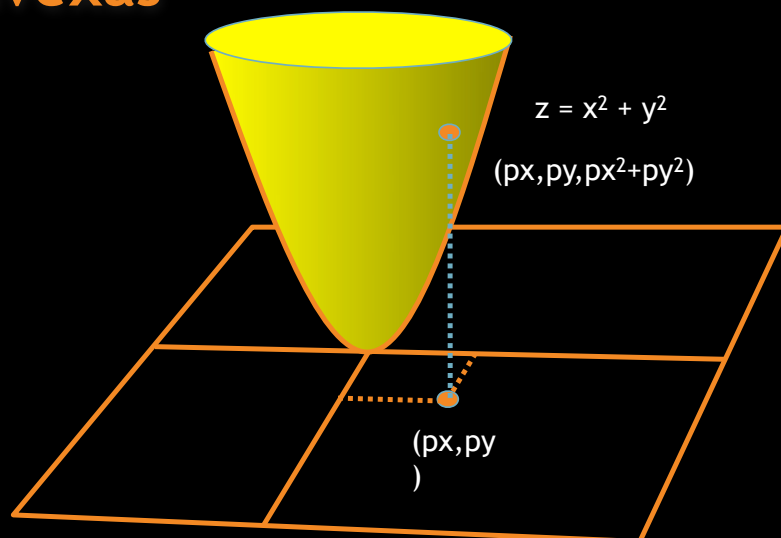
Map  $h(p) : z = 2p_x x + 2p_y y - (p_x^2 + p_y^2)$

$\cap h^+$  mapped back to lower dimension is the Voronoi diagram!!!

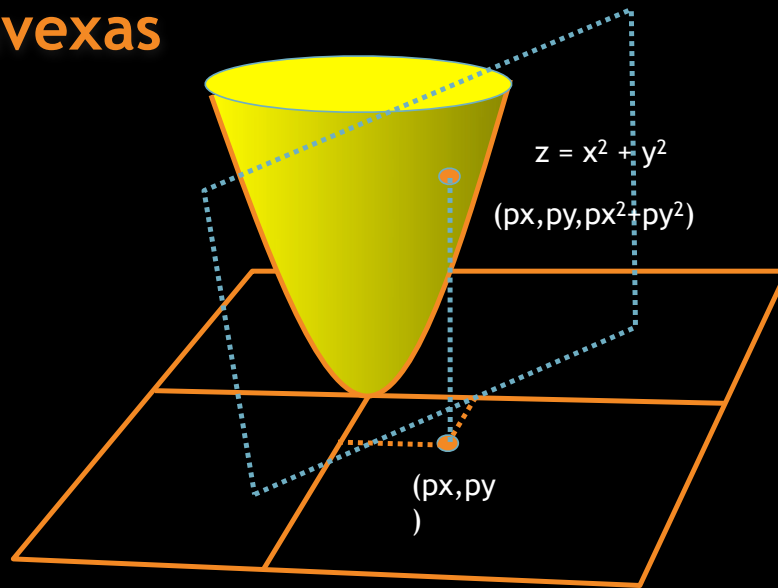
## Conexoes com Envoltorias Convexas



## Conexoes com Envoltorias Convexas



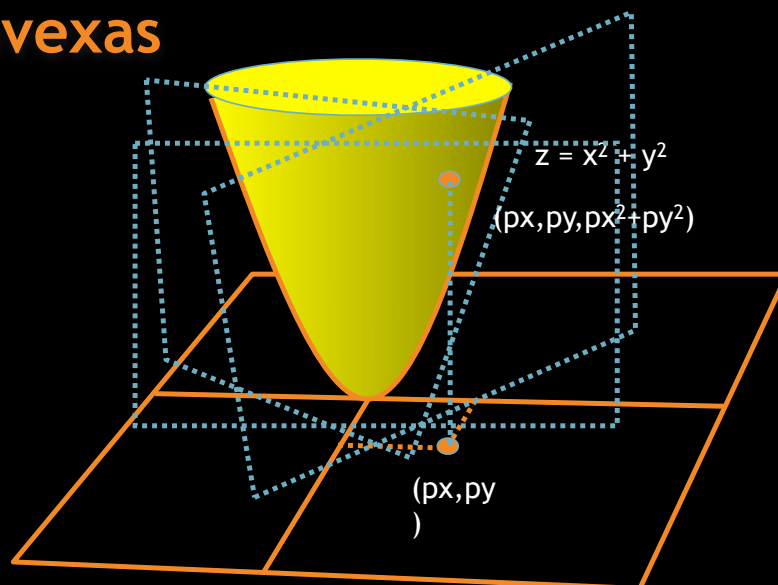
## Conexões com Envoltórias Convexas



$$H(p): z = 2p_x x + 2p_y y - (p_x^2 + p_y^2)$$

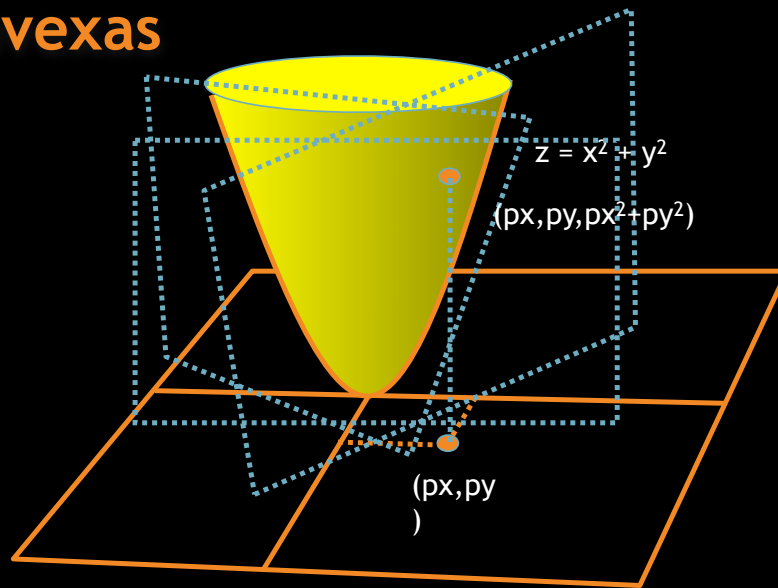
Plano tangente ao parabolóide

## Conexões com Envoltórias Convexas



Poliedro convexo definidos pela intersecoes de todos semi-espacos correspondentes a todos planos  $H(P)$

## Conexões com Envoltórias Convexas



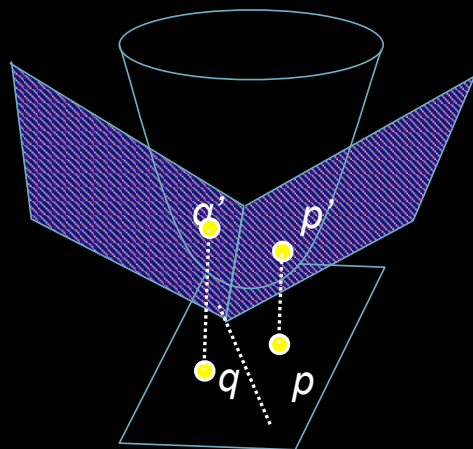
Projetando as arestas e vertices do poliedro no plano obtém-se o diagrama de Voronoi

## The Voronoi Diagram and Convex Hulls

Given a set  $S$  of points in the plane, associate with each point  $p=(a,b) \in S$  the plane tangent to the paraboloid:

$$z = 2ax + 2by - (a^2 + b^2).$$

$VD(S)$  is the projection to the  $(x,y)$  plane of the 1-skeleton of the convex polyhedron formed from the intersection of the halfspaces above these planes.





# In-Circle Test

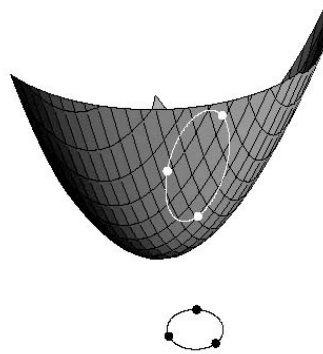


Figure 2.7: Lifting  $(x, y)$  to  $(x, y, x^2 + y^2)$ .

## Lemma 2.9

Let  $C$  be a circle in the  $xy$ -plane.

- (i)  $\psi(C) = P \cap H_C$ , where  $H_C$  is a plane in  $\mathbb{R}^3$ .
- (ii) A point  $q$  is inside  $C$  if and only if  $\psi(q)$  below  $H_C$ .

# In-Circle Test

