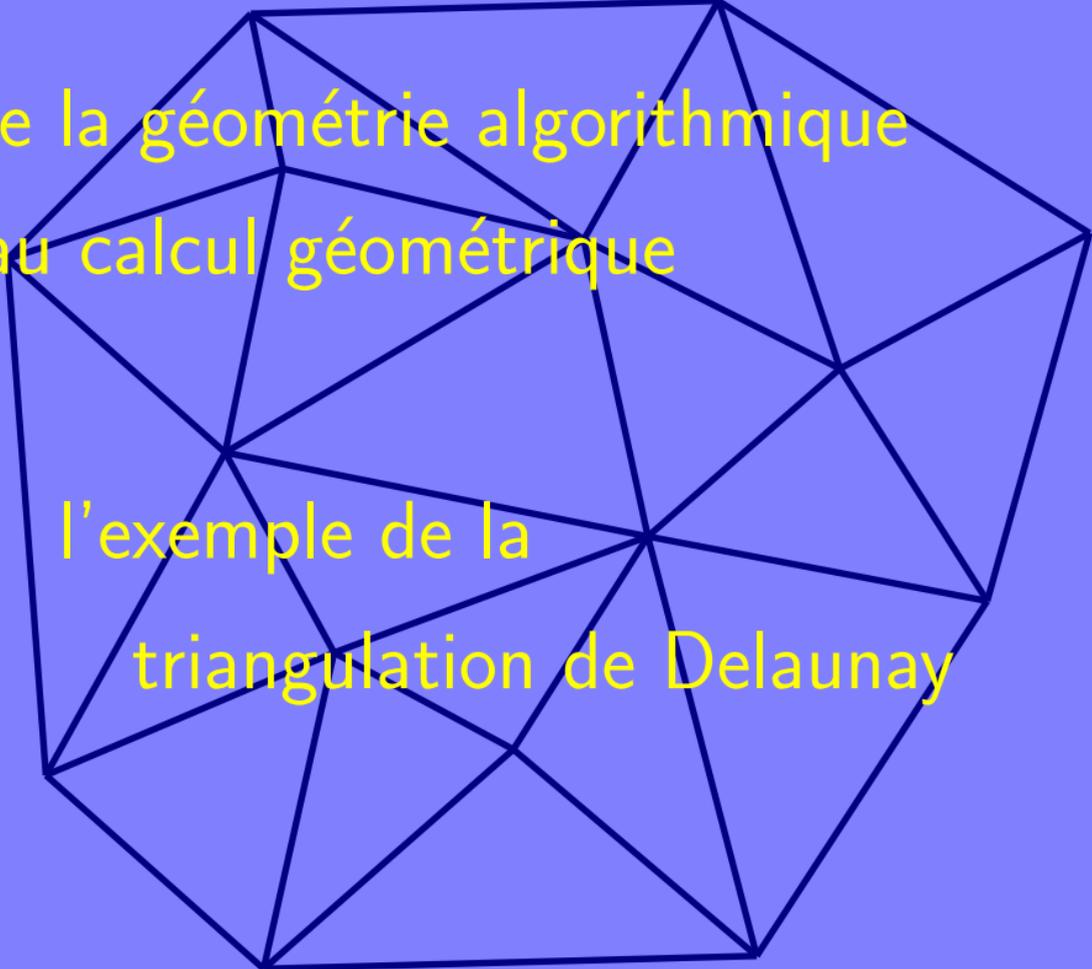


# De la géométrie algorithmique au calcul géométrique



De la géométrie algorithmique  
au calcul géométrique

l'exemple de la  
triangulation de Delaunay



De la géométrie algorithmique  
au calcul géométrique

l'exemple de la  
triangulation de Delaunay

# Robustesse

Robustesse

Exemple

A

B

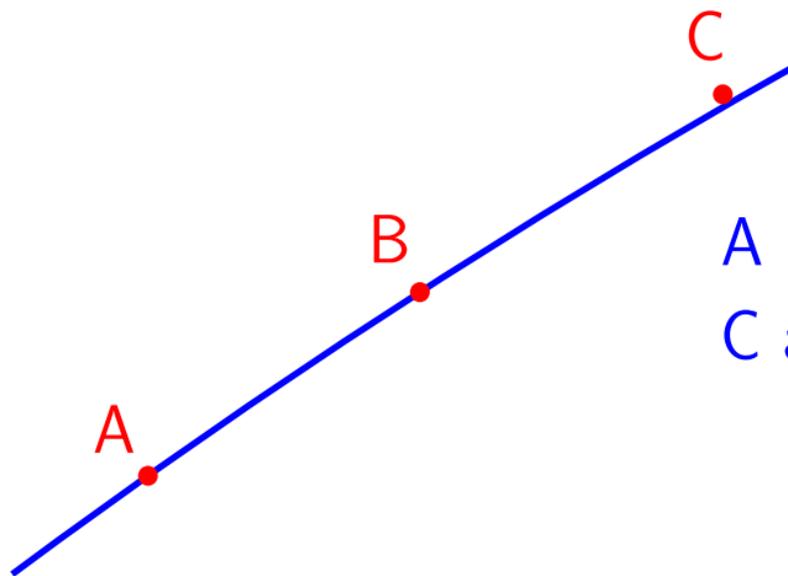
C

D

$A <_x B <_x C <_x D$

Robustesse

Exemple

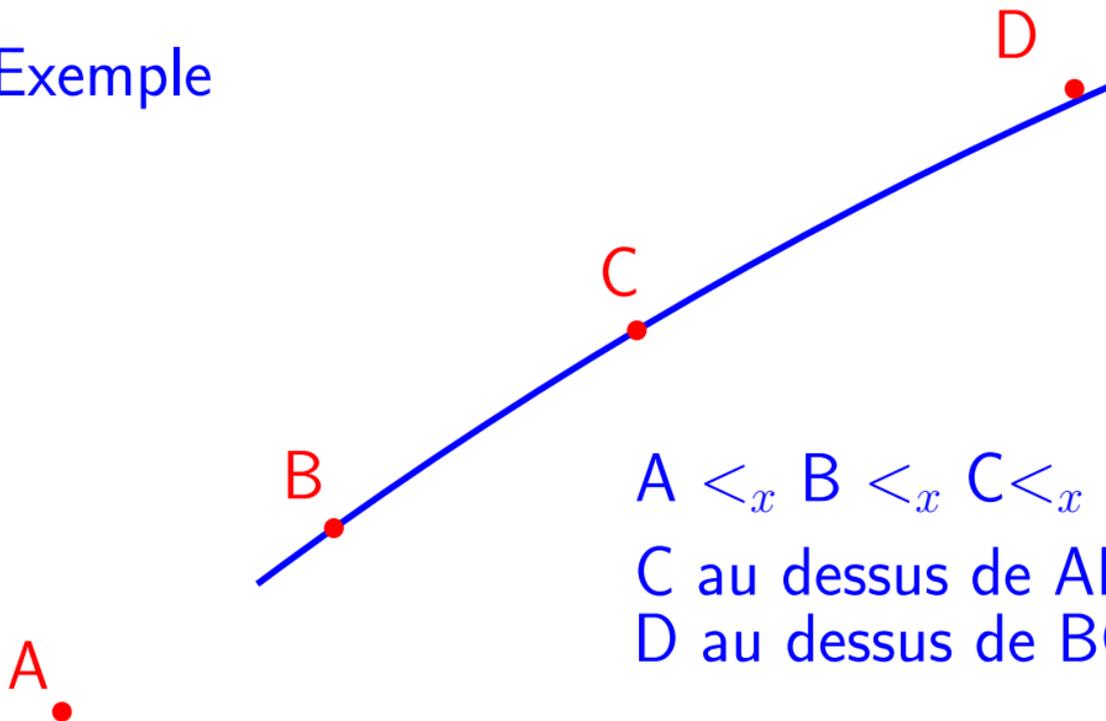


$A <_x B <_x C <_x D$

C au dessus de AB

Robustesse

Exemple



$$A <_x B <_x C <_x D$$

C au dessus de AB

D au dessus de BC

Robustesse

Exemple

A

B

C

D

$A <_x B <_x C <_x D$

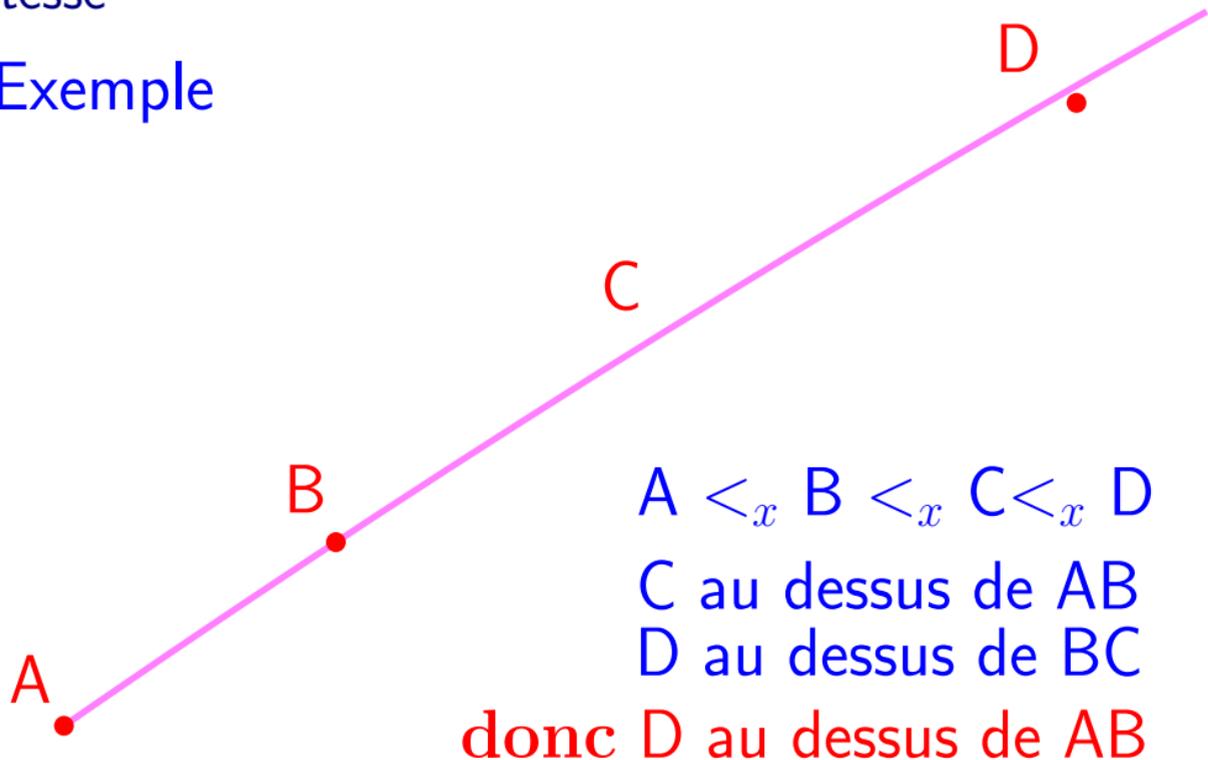
C au dessus de AB

D au dessus de BC

donc D au dessus de AB

Robustesse

Exemple



mais l'évaluation des prédicats donne le contraire

Algorithmes géométriques

Modèle Real RAM

Algorithmes géométriques

Modèle Real RAM

On sait calculer avec des réels

Algorithmes géométriques

Modèle Real RAM

On sait calculer avec des réels



float, double

float, double

c'est pas des réels

float, double

c'est pas des réels

erreurs d'arrondi

float, double

c'est pas des réels

erreurs d'arrondi

Problème

float, double

c'est pas des réels

erreurs d'arrondi

Problème

≠ Inexactitudes

float, double

c'est pas des réels

erreurs d'arrondi

Problème

≠ Inexactitudes

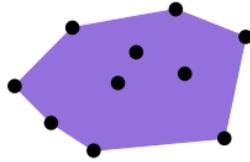
= Incohérences

# Résultat combinatoire

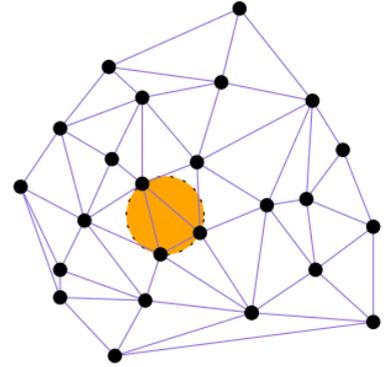
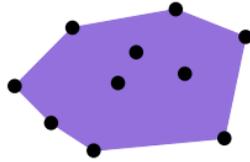
# Résultat combinatoire



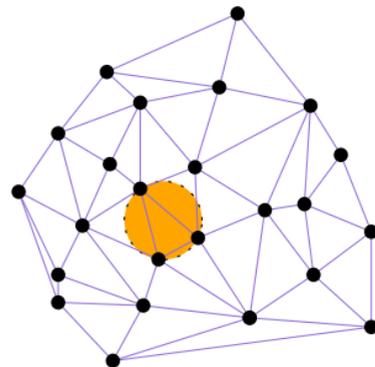
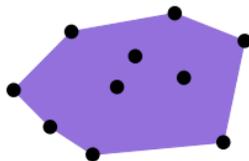
# Résultat combinatoire



# Résultat combinatoire

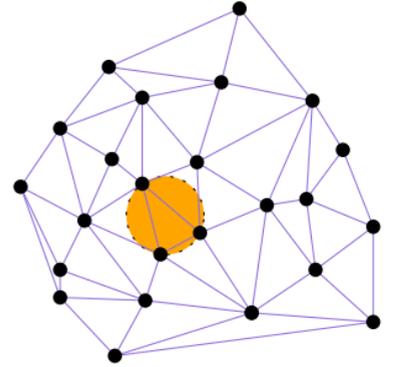
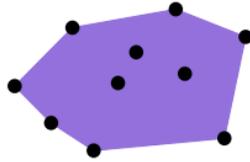


# Résultat combinatoire



Prédicats

# Résultat combinatoire

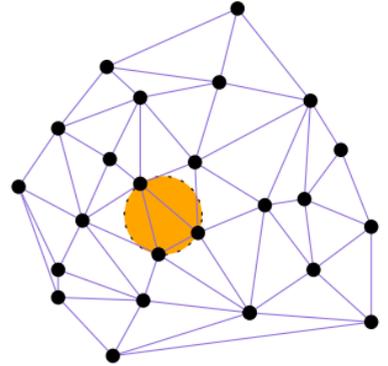
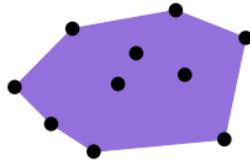


Prédicats

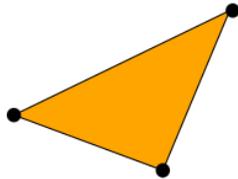


Comparaison

# Résultat combinatoire



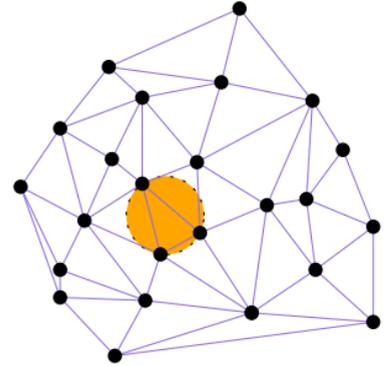
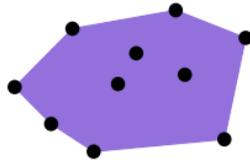
## Prédicats



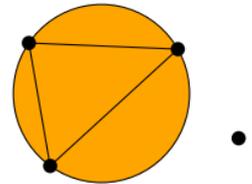
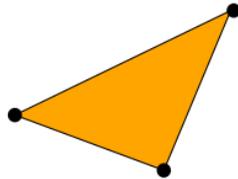
Comparaison

Orientation

# Résultat combinatoire



## Prédicats



Comparaison

Orientation

Cocircularité

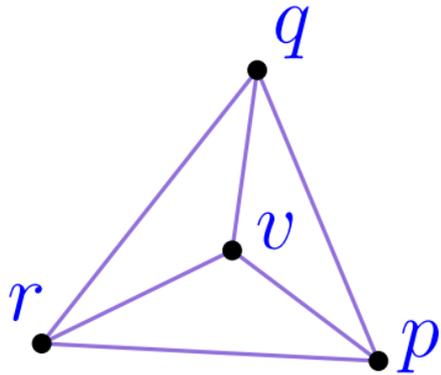
On a tendance à utiliser la géométrie

On a tendance à utiliser la géométrie

i.e. les théorèmes géométriques

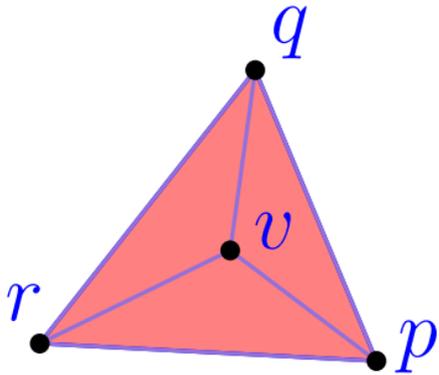
dans les algorithmes

# théorème géométrique utile aux algorithmes

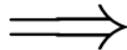


$$\begin{array}{l} pqv \\ qrv \\ rpv \end{array} \text{ CCW} \implies pqr$$

# théorème géométrique utile aux algorithmes

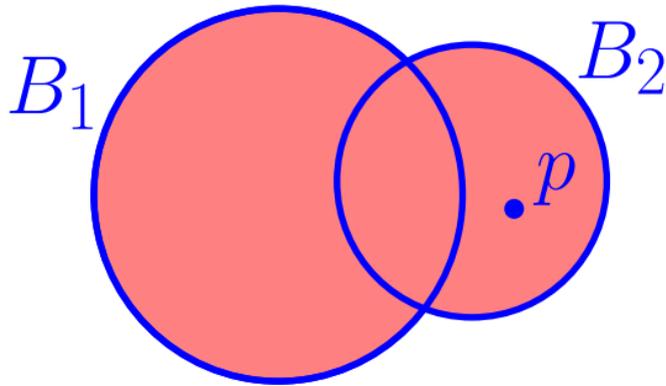


$pqv$   
 $qrv$  **CCW**  
 $rpv$



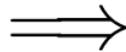
$pqr$  **CCW**

# théorème géométrique utile aux algorithmes

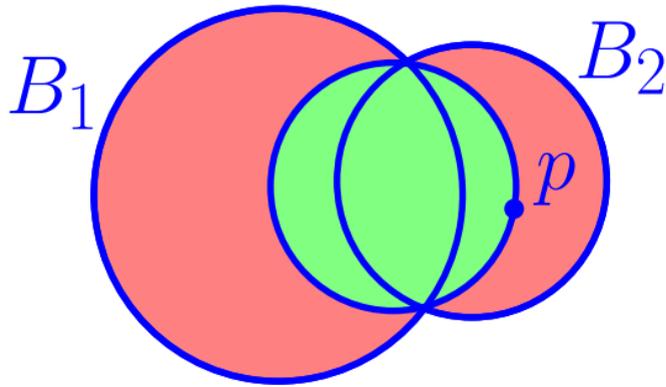


$$p \in B_2$$

$$p \notin B_1$$

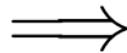


# théorème géométrique utile aux algorithmes



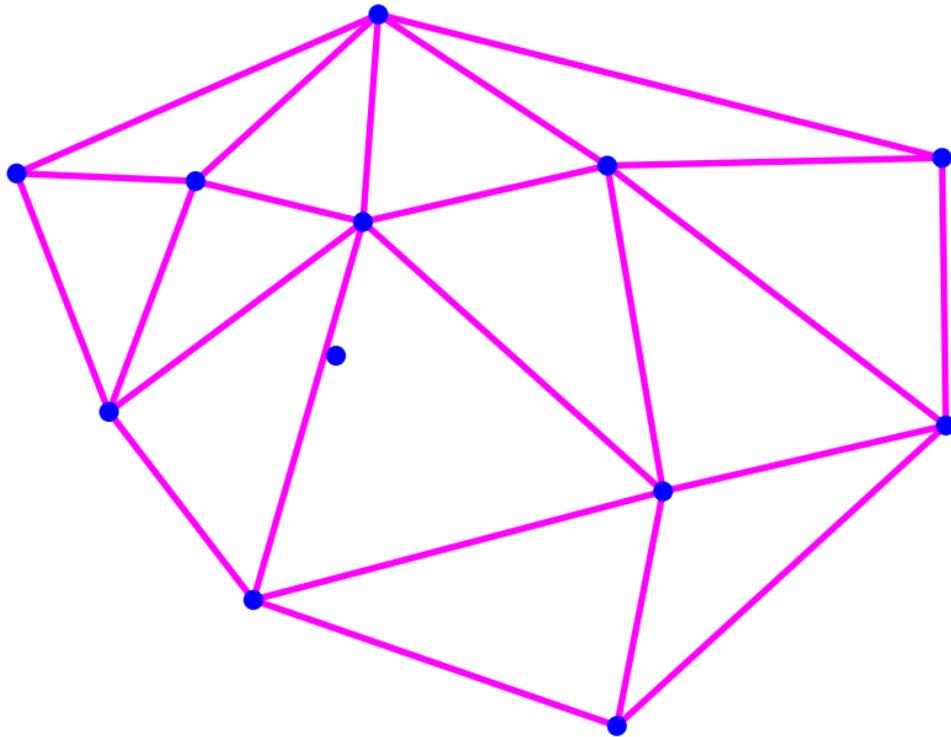
$$p \in B_2$$

$$p \notin B_1$$

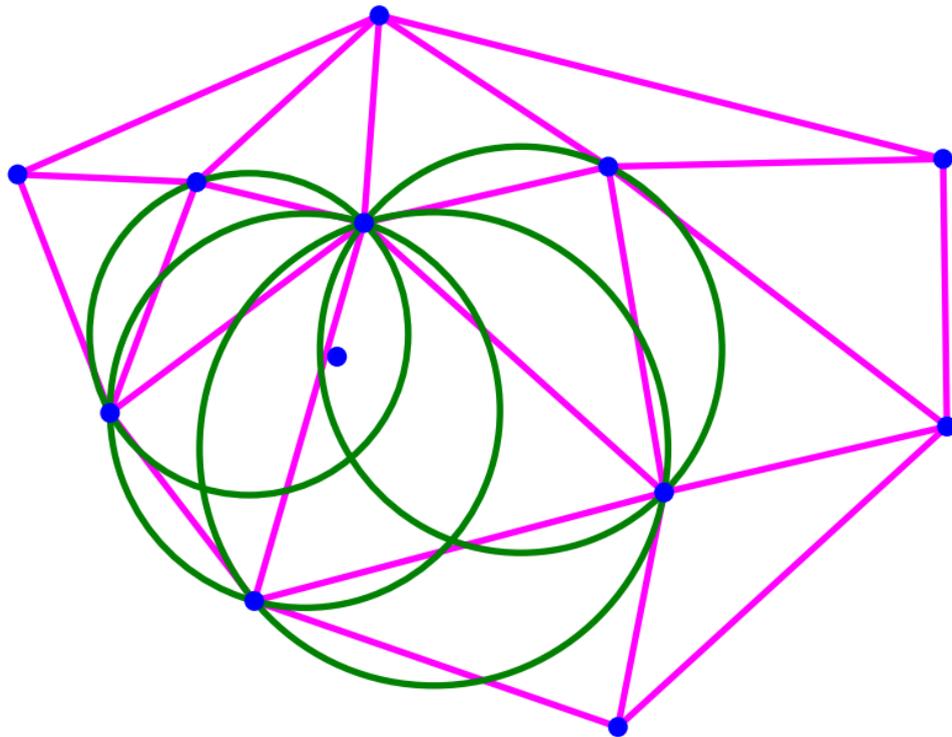


$$B' \subset B_1 \cup B_2$$

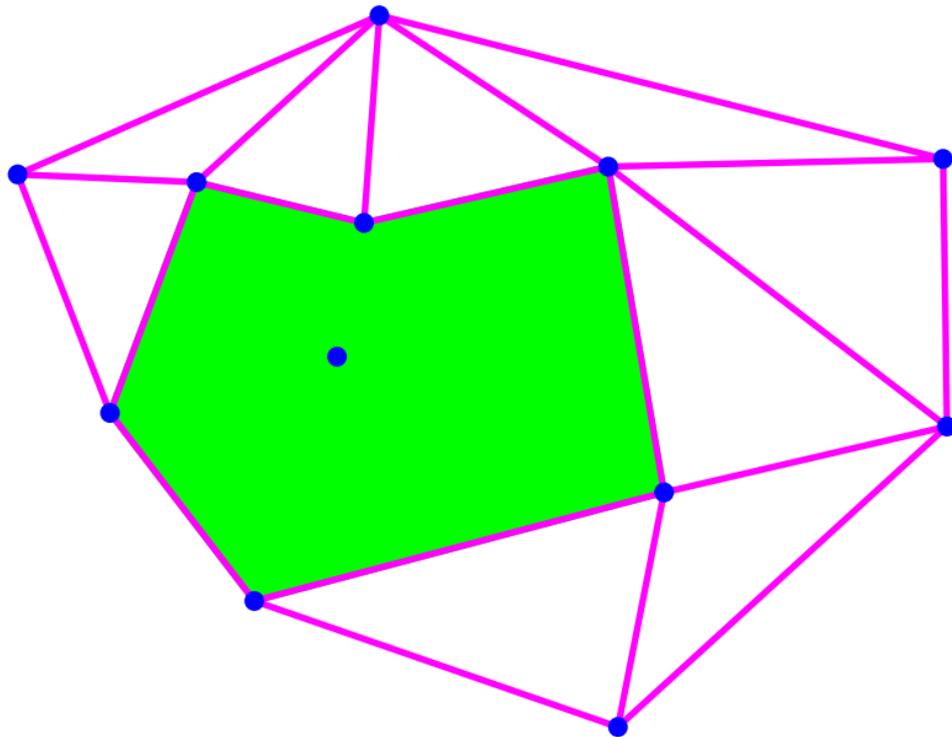
# Insertion dans Delaunay



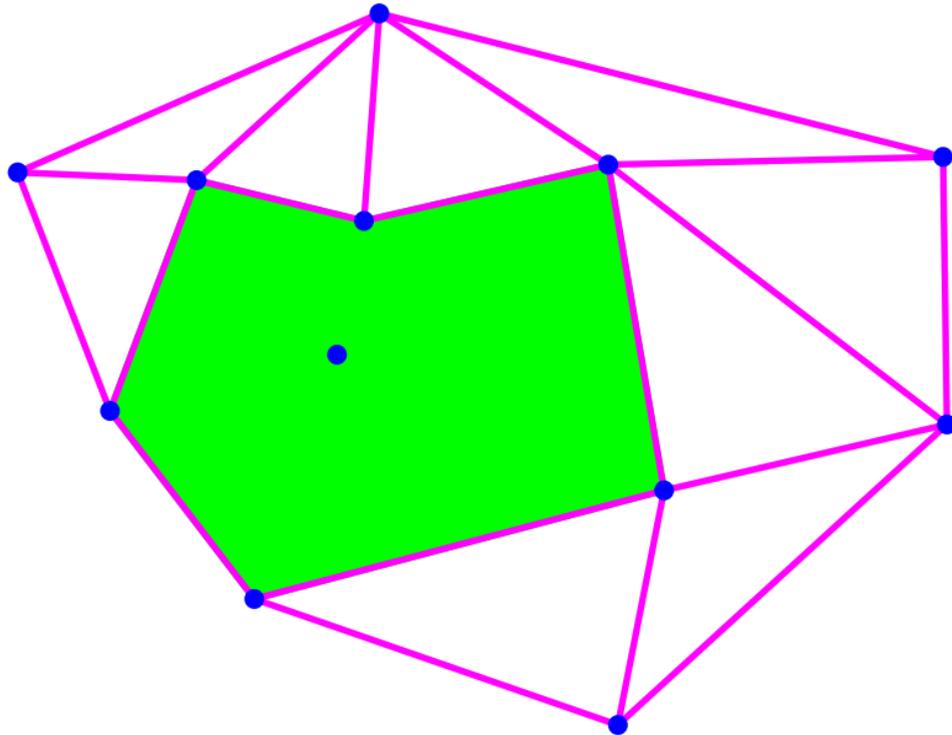
# Insertion dans Delaunay



# Insertion dans Delaunay

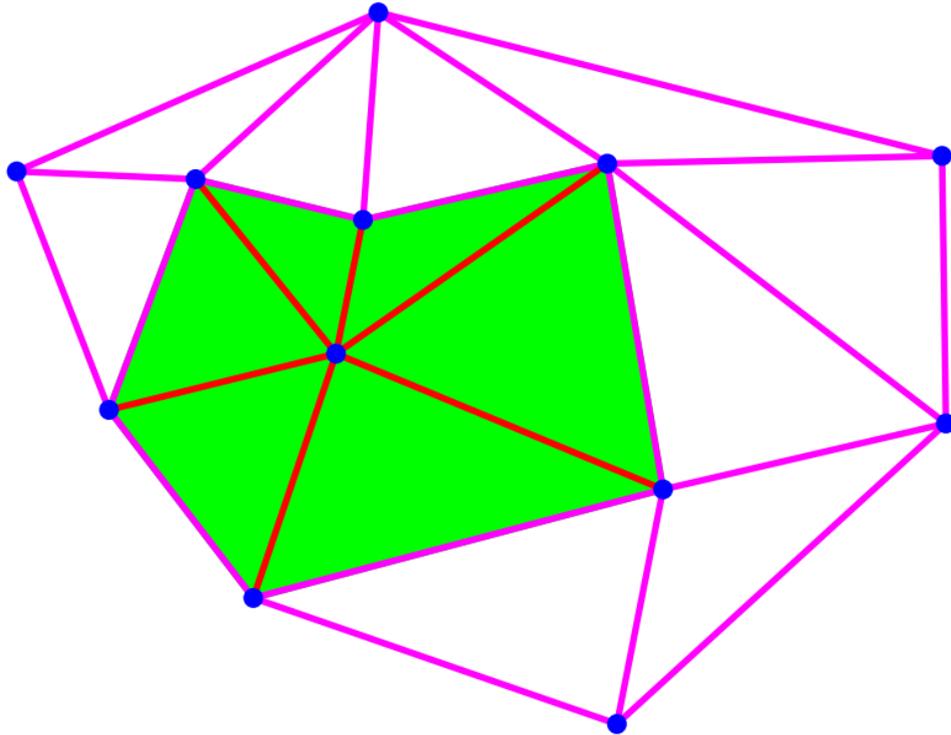


# Insertion dans Delaunay



est forcément étoilé

# Insertion dans Delaunay

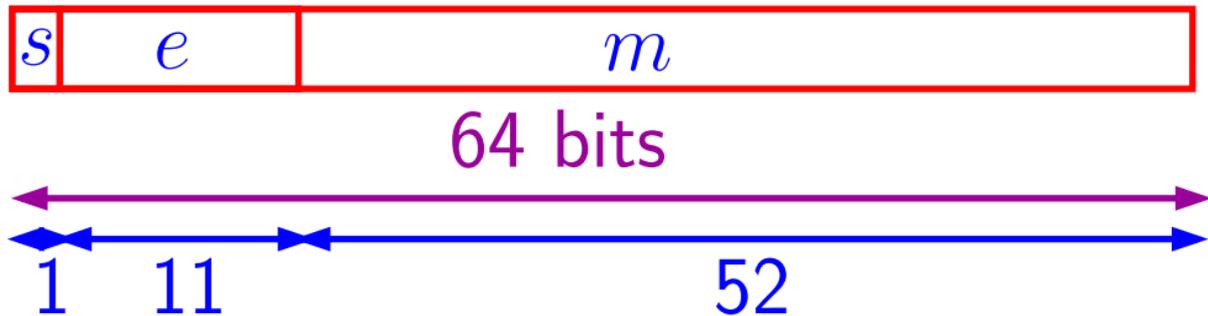


est forcément étoilé

# Rappels sur l'arithmétique flottante

double

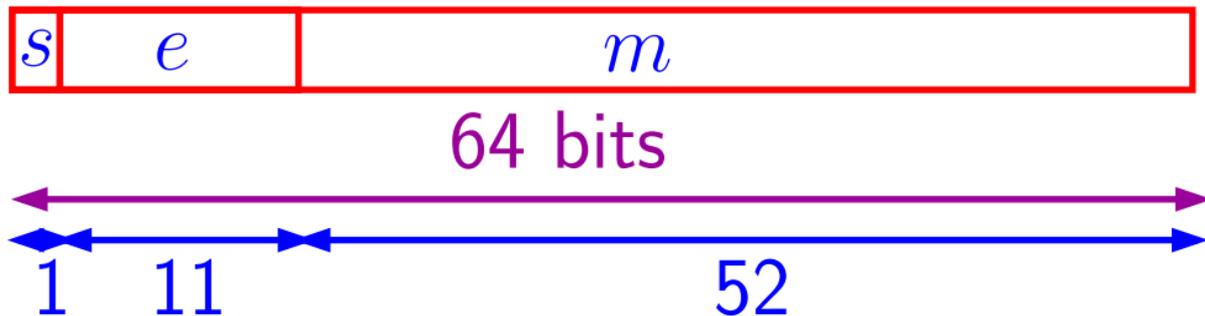
Norme IEEE 754



# Rappels sur l'arithmétique flottante

double

Norme IEEE 754

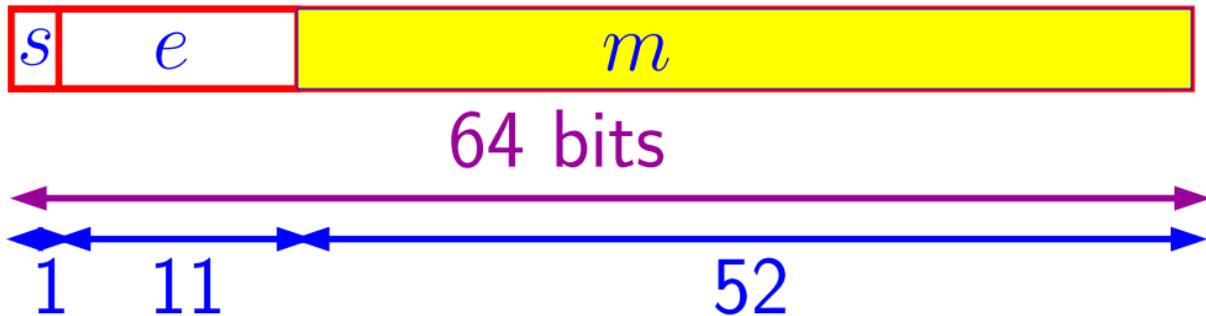


0.1

# Rappels sur l'arithmétique flottante

double

Norme IEEE 754



# Rappels sur l'arithmétique flottante

double

Norme IEEE 754



64 bits



# Rappels sur l'arithmétique flottante

double

Norme IEEE 754



64 bits



$$-1 \overset{s}{\square} 0.1 \overset{m}{\square} 2^{-1024 + \overset{e}{\square}}$$

# Rappels sur l'arithmétique flottante

double

Norme IEEE 754



64 bits



$$-1 \overset{s}{\square} 0.1 \overset{m}{\square} 2^{-1024 + \overset{e}{\square}}$$

nombres "normalisés"

# Rappels sur l'arithmétique flottante

## Nombres représentables



# Rappels sur l'arithmétique flottante

## Nombres représentables



$$0.11010 \dots 01001 \times 2^e$$

# Rappels sur l'arithmétique flottante

## Nombres représentables

$$0.11010 \dots 01010 \times 2^e$$



$$0.11010 \dots 01001 \times 2^e$$

# Rappels sur l'arithmétique flottante

## Nombres représentables

$$0.11010 \dots 01010 \times 2^e$$

$$0.11010 \dots 01011 \times 2^e$$



$$0.11010 \dots 01001 \times 2^e$$

# Rappels sur l'arithmétique flottante

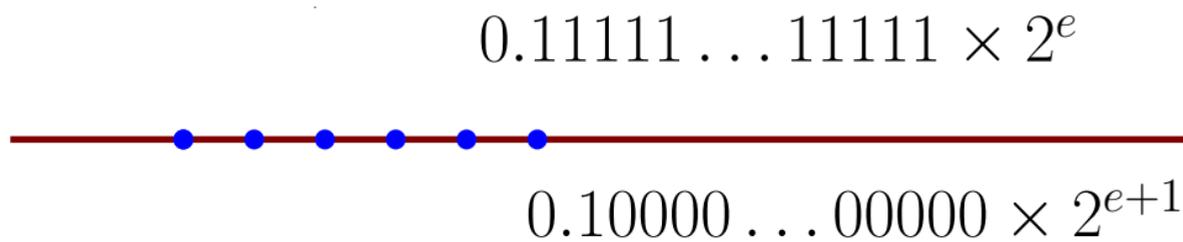
## Nombres représentables

$$0.11111 \dots 11111 \times 2^e$$



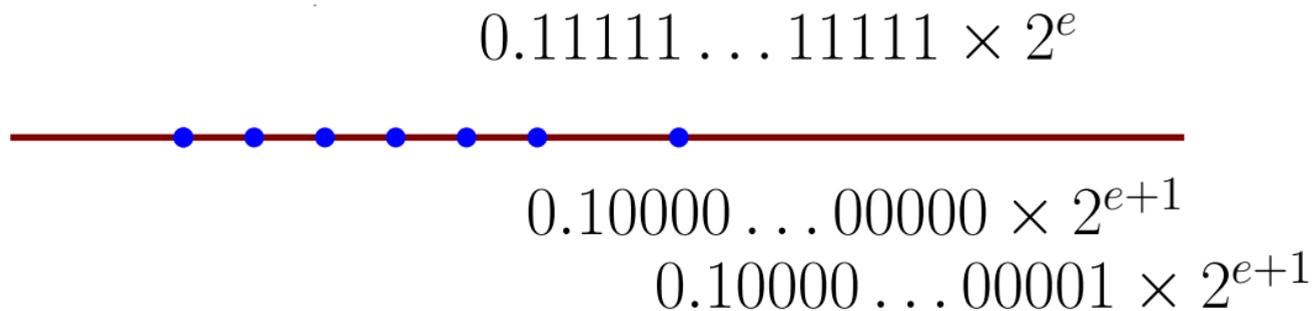
# Rappels sur l'arithmétique flottante

## Nombres représentables



# Rappels sur l'arithmétique flottante

## Nombres représentables



# Rappels sur l'arithmétique flottante

## Nombres représentables

$$0.11111 \dots 11111 \times 2^e$$



$$0.10000 \dots 00000 \times 2^{e+1}$$

$$0.10000 \dots 00001 \times 2^{e+1}$$

$$0.10000 \dots 00010 \times 2^{e+1}$$

$$0.10000 \dots 00011 \times 2^{e+1}$$

# Rappels sur l'arithmétique flottante

## Plus petit nombre représentable

$$0.00000 \dots 00000 \times 2^{-1024}$$



# Rappels sur l'arithmétique flottante

## Plus petit nombre représentable

$$0.00000 \dots 00000 \times 2^{-1024}$$



$$0.10000 \dots 00000 \times 2^{-1024}$$

# Rappels sur l'arithmétique flottante

## Plus petit nombre représentable

$$0.00000 \dots 00000 \times 2^{-1024}$$



$$0.10000 \dots 00000 \times 2^{-1024}$$

$$0.10000 \dots 00001 \times 2^{-1024}$$

$$0.10000 \dots 00010 \times 2^{-1024}$$

# Rappels sur l'arithmétique flottante

## Plus petit nombre représentable

$$0.00000 \dots 00000 \times 2^{-1024} \quad 0.11111 \dots 11111 \times 2^{-1024}$$



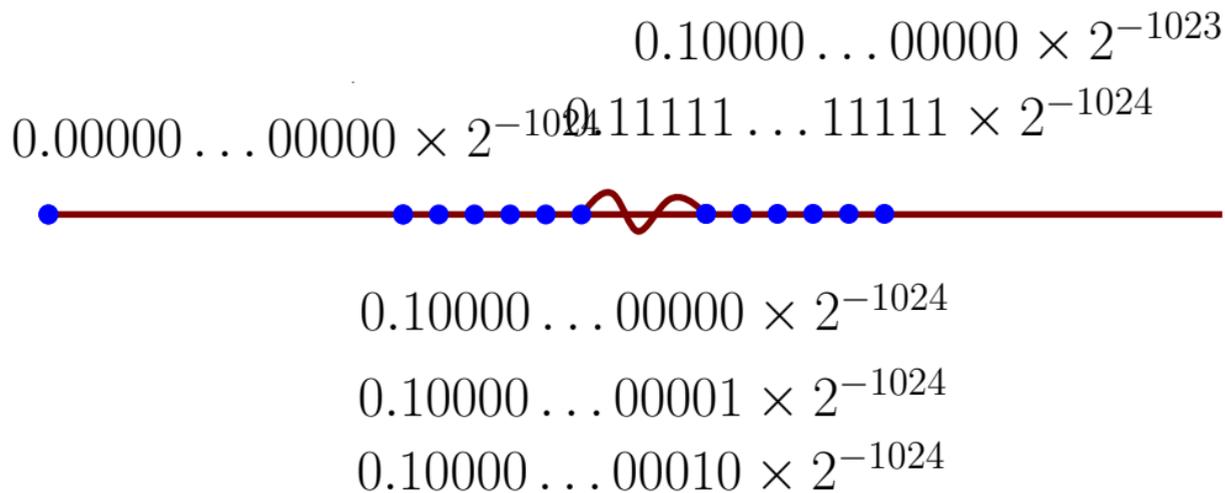
$$0.10000 \dots 00000 \times 2^{-1024}$$

$$0.10000 \dots 00001 \times 2^{-1024}$$

$$0.10000 \dots 00010 \times 2^{-1024}$$

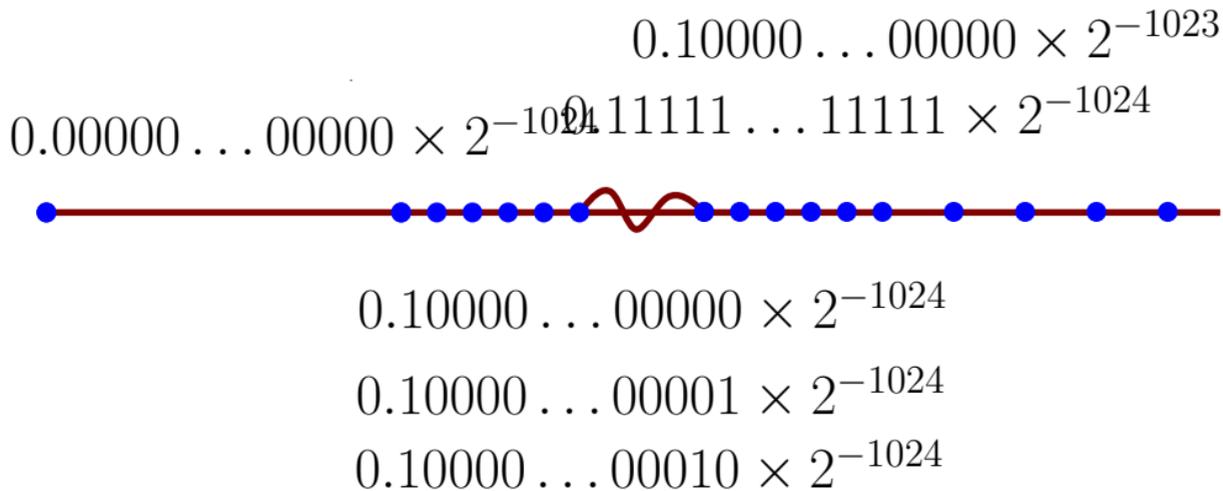
# Rappels sur l'arithmétique flottante

## Plus petit nombre représentable



# Rappels sur l'arithmétique flottante

## Plus petit nombre représentable



# Rappels sur l'arithmétique flottante

## Nombres dénormalisés

$$0.00000 \dots 00000 \times 2^{-1024}$$

$$0.10000 \dots 00000 \times 2^{-1023}$$



# Rappels sur l'arithmétique flottante

## Nombres dénormalisés

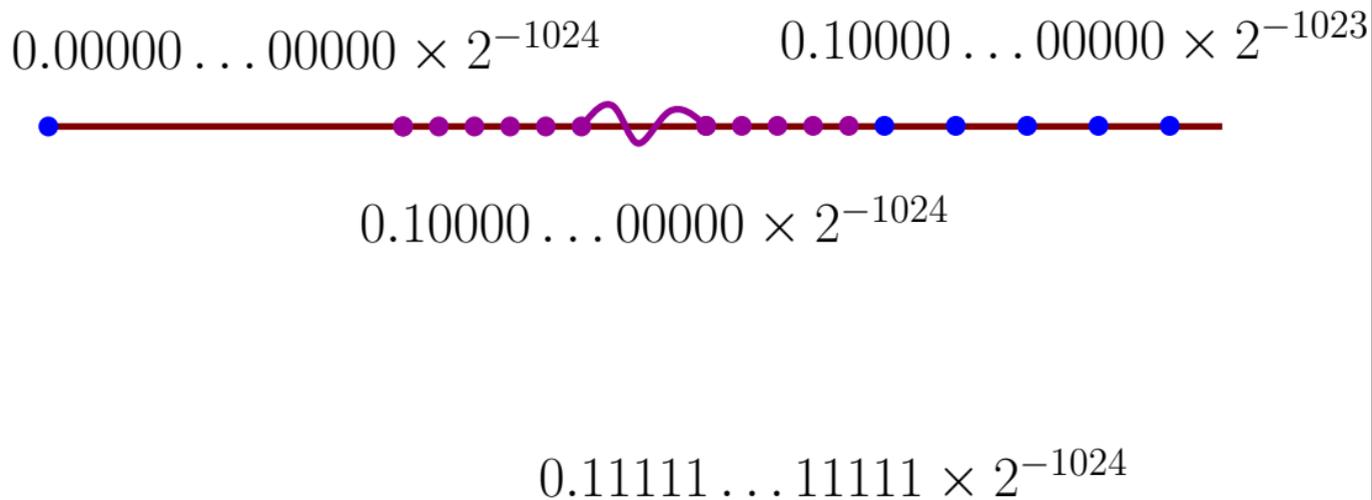
$0.00000 \dots 00000 \times 2^{-1024}$

$0.10000 \dots 00000 \times 2^{-1023}$



# Rappels sur l'arithmétique flottante

## Nombres dénormalisés



# Rappels sur l'arithmétique flottante

## Nombres dénormalisés

$$0.00000 \dots 00000 \times 2^{-1024}$$

$$0.10000 \dots 00000 \times 2^{-1023}$$



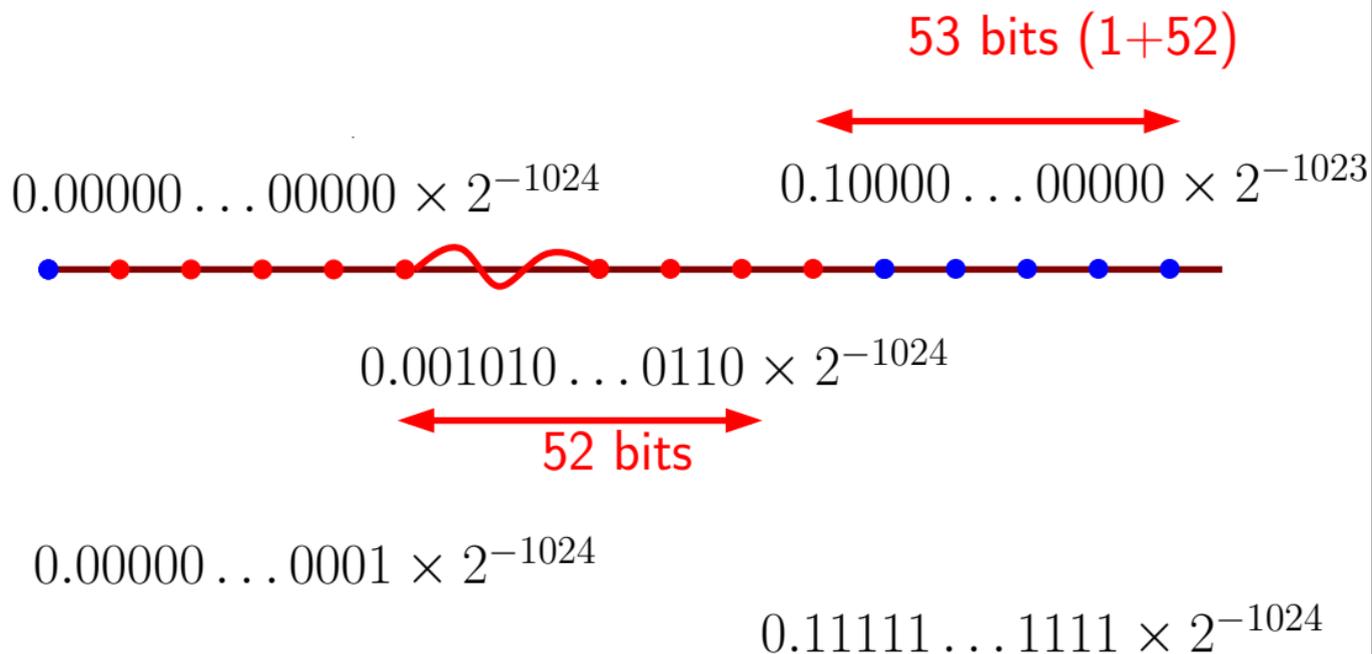
$$0.001010 \dots 0110 \times 2^{-1024}$$

$$0.00000 \dots 0001 \times 2^{-1024}$$

$$0.11111 \dots 1111 \times 2^{-1024}$$

# Rappels sur l'arithmétique flottante

## Nombres dénormalisés



# Rappels sur l'arithmétique flottante

double

Norme IEEE 754



64 bits



$$-1 \overset{s}{\square} 0.1 \overset{m}{\square} 2^{-1024 + \overset{e}{\square}}$$

nombres "normalisés"

# Rappels sur l'arithmétique flottante

double

Norme IEEE 754



64 bits



$$-1 \overset{s}{\square} 0. \overset{m}{\square} 2^{-1024}$$

nombres "dénormalisés"

# Rappels sur l'arithmétique flottante

## Règles d'arrondi



# Rappels sur l'arithmétique flottante

## Règles d'arrondi



Nombres représentables

# Rappels sur l'arithmétique flottante

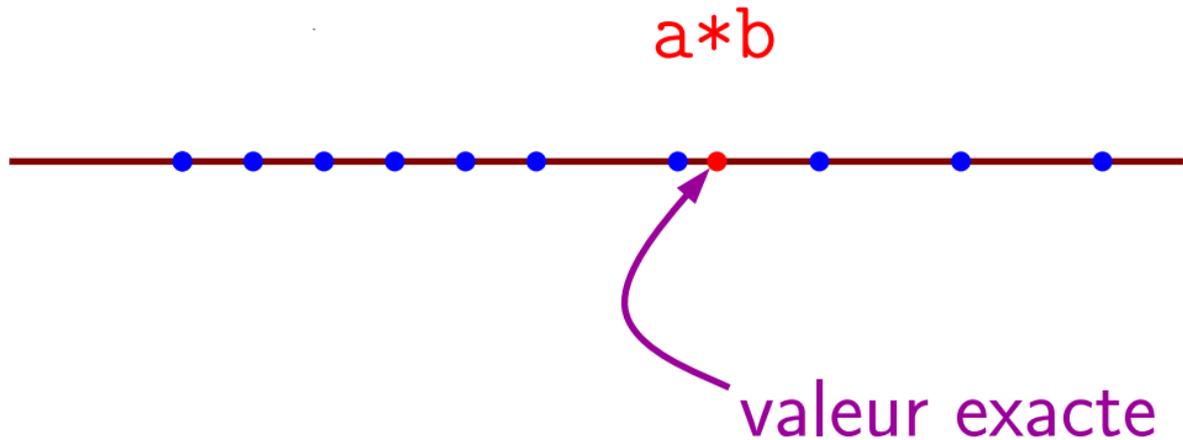
## Règles d'arrondi

$a*b$



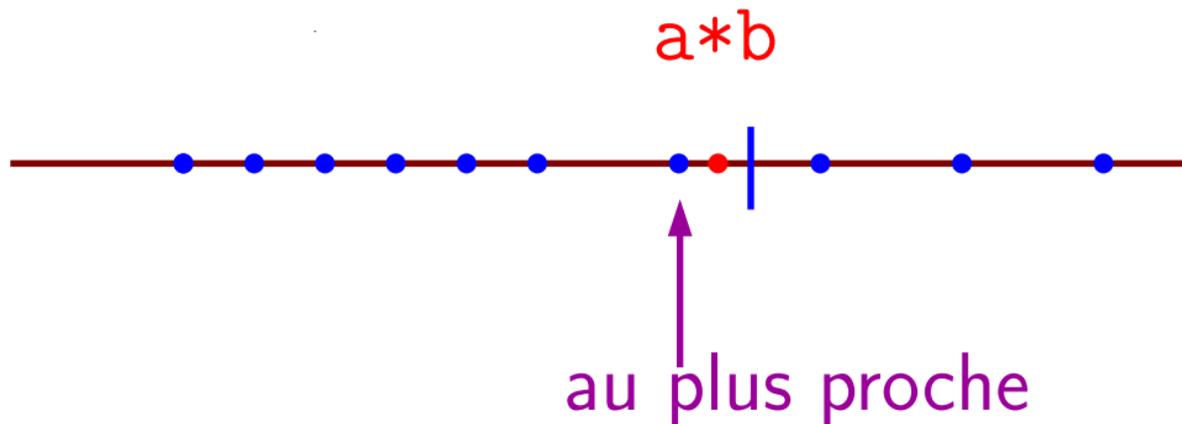
# Rappels sur l'arithmétique flottante

## Règles d'arrondi



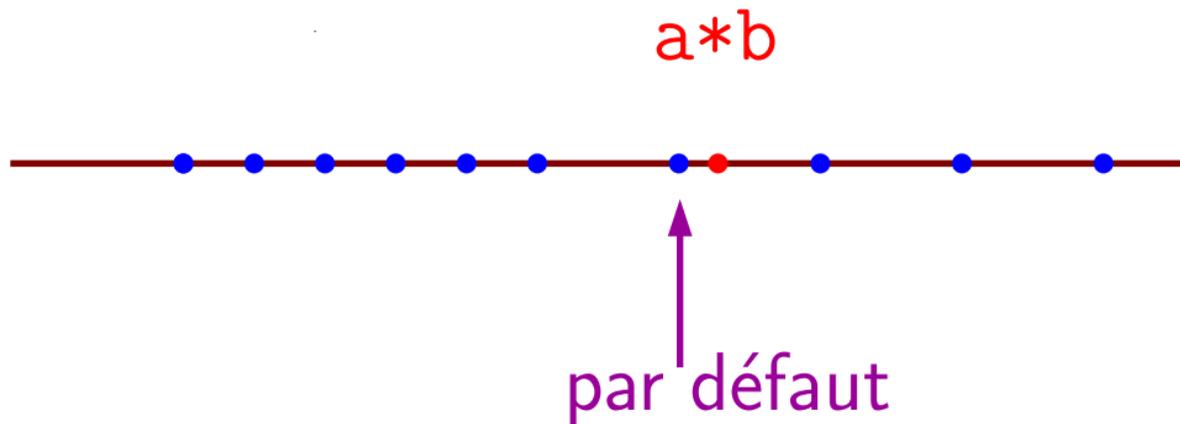
# Rappels sur l'arithmétique flottante

## Règles d'arrondi



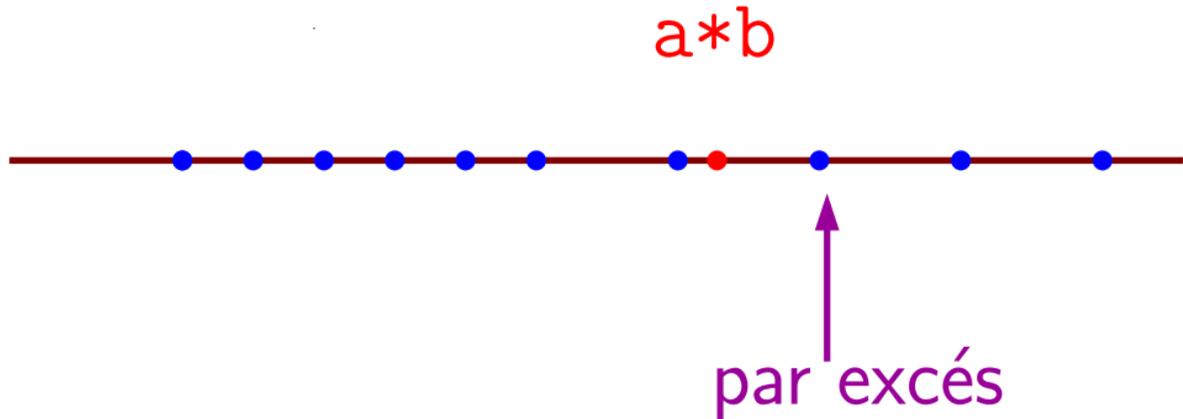
# Rappels sur l'arithmétique flottante

## Règles d'arrondi



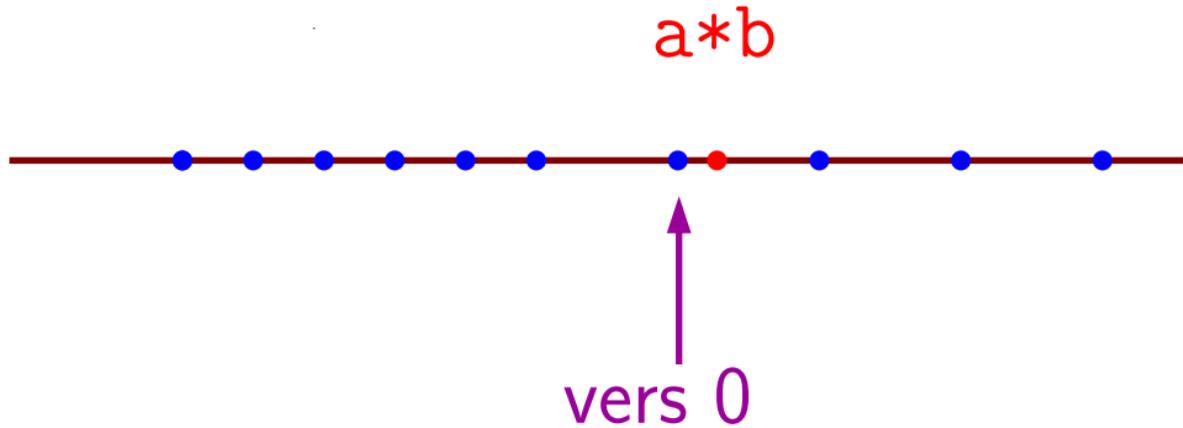
# Rappels sur l'arithmétique flottante

## Règles d'arrondi



# Rappels sur l'arithmétique flottante

## Règles d'arrondi



# Rappels sur l'arithmétique flottante

## Règles d'arrondi

4 opérations et racine carrée



$a+b$

$a*b$

$\text{sqrt}(a)$

$a-b$

$a/b$

# Rappels sur l'arithmétique flottante

double                    53 chiffres binaires

float                    24 chiffres binaires

# Rappels sur l'arithmétique flottante

double	53 chiffres binaires
float	24 chiffres binaires
modèle jouet	2 chiffres décimaux significatifs

# Rappels sur l'arithmétique flottante

double 53 chiffres binaires

float 24 chiffres binaires

modèle jouet 2 chiffres décimaux  
significatifs

$$35 + 3.7 =$$

# Rappels sur l'arithmétique flottante

double 53 chiffres binaires

float 24 chiffres binaires

modèle jouet 2 chiffres décimaux  
significatifs

$$35 + 3.7 = 39$$

# Rappels sur l'arithmétique flottante

double                      53 chiffres binaires

float                        24 chiffres binaires

modèle jouet              2 chiffres décimaux  
significatifs

$$35 + 3.7 = 39$$

$$35 + 3.3 + 0.4 =$$

# Rappels sur l'arithmétique flottante

double                    53 chiffres binaires

float                    24 chiffres binaires

modèle jouet            2 chiffres décimaux  
significatifs

$$35 + 3.7 = 39$$

$$35 + 3.3 + 0.4 = 38$$

# Rappels sur l'arithmétique flottante

double                    53 chiffres binaires

float                    24 chiffres binaires

modèle jouet            2 chiffres décimaux  
significatifs

$$35 + 3.7 = 39$$

$$35 + 3.3 + 0.4 = 38$$

$$35 + (3.3 + 0.4) =$$

# Rappels sur l'arithmétique flottante

double                      53 chiffres binaires

float                        24 chiffres binaires

modèle jouet              2 chiffres décimaux  
significatifs

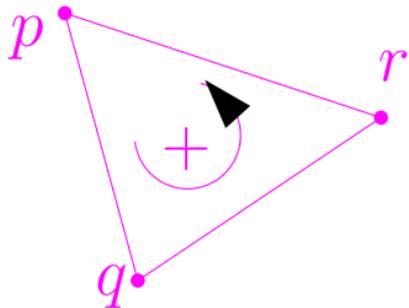
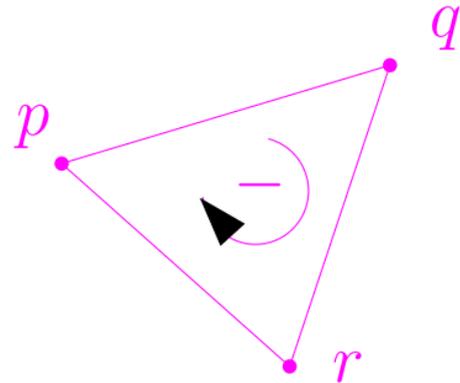
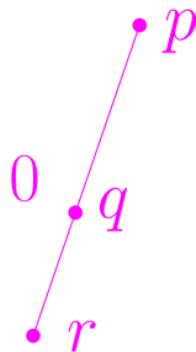
$$35 + 3.7 = 39$$

$$35 + 3.3 + 0.4 = 38$$

$$35 + (3.3 + 0.4) = 39$$

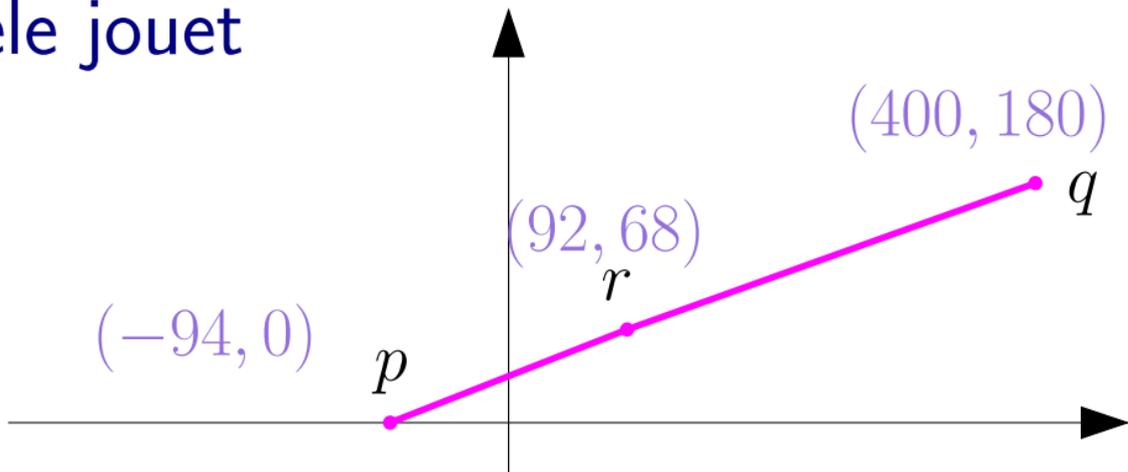
# Prédicats

## Prédicat d'orientation

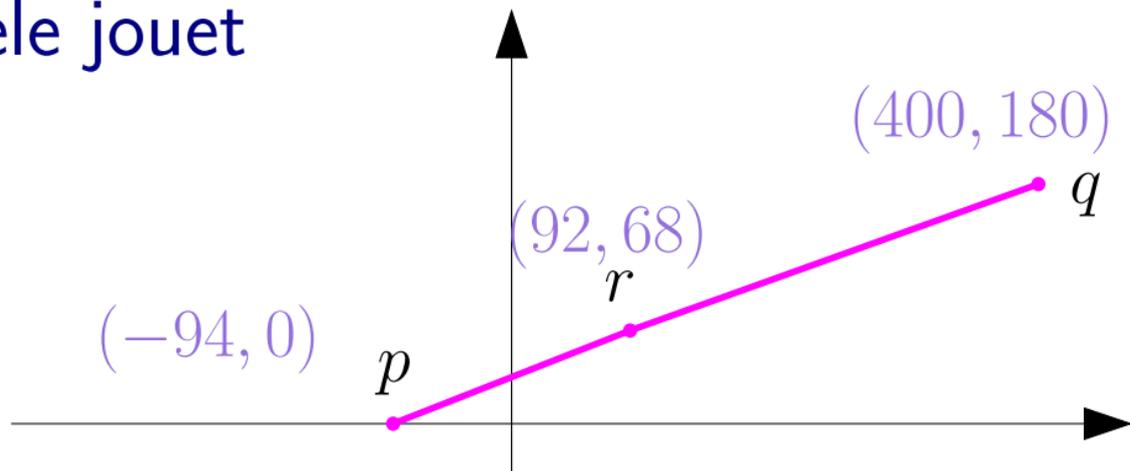


$$\begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix}$$

# modèle jouet

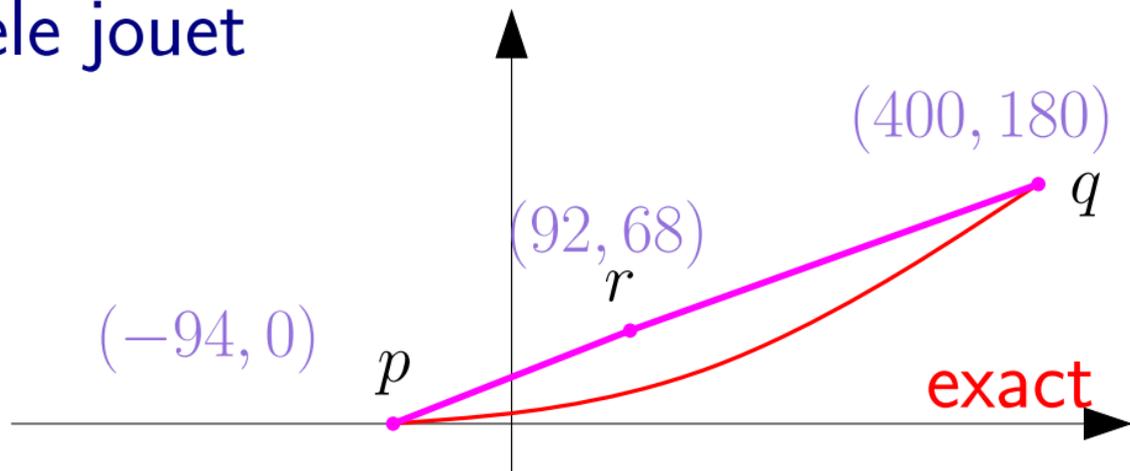


# modèle jouet



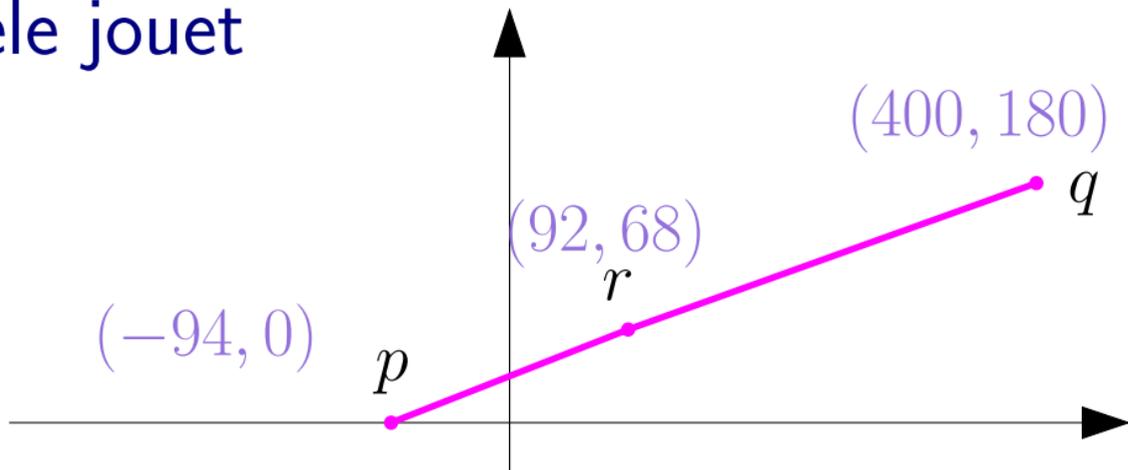
$$(400 + 94) * 68 - (92 + 94) * 180 =$$

# modèle jouet



$$(400 + 94) * 68 - (92 + 94) * 180 = 112$$

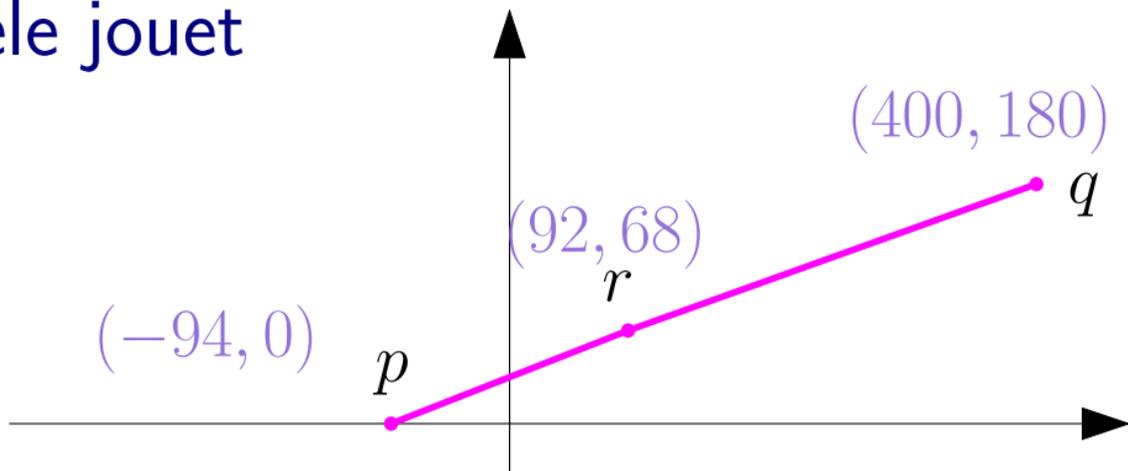
# modèle jouet



$$(400 + 94) * 68 - (92 + 94) * 180 =$$

$$494 * 68 - 186 * 180$$

# modèle jouet

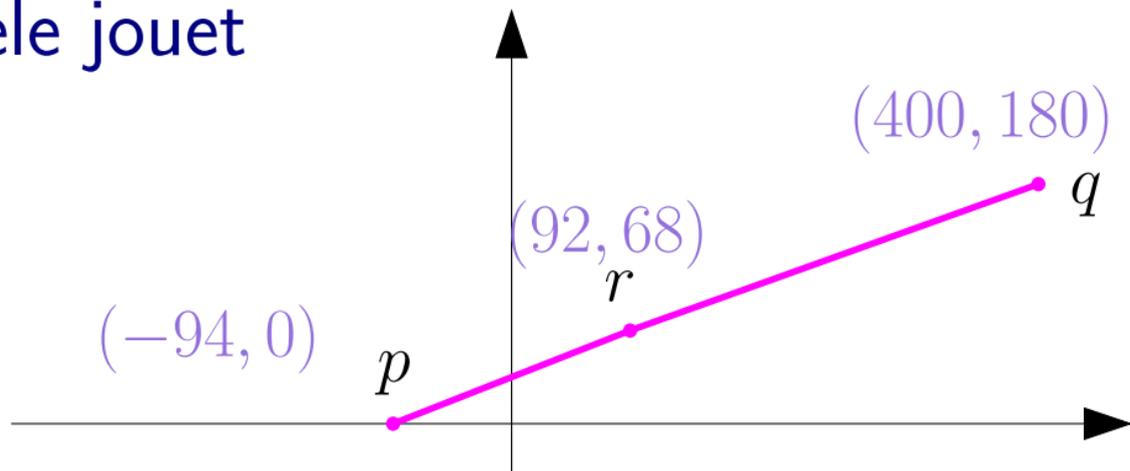


$$(400 + 94) * 68 - (92 + 94) * 180 =$$

$$494 * 68 - 186 * 180$$

$$490 * 68 - 190 * 180$$

# modèle jouet



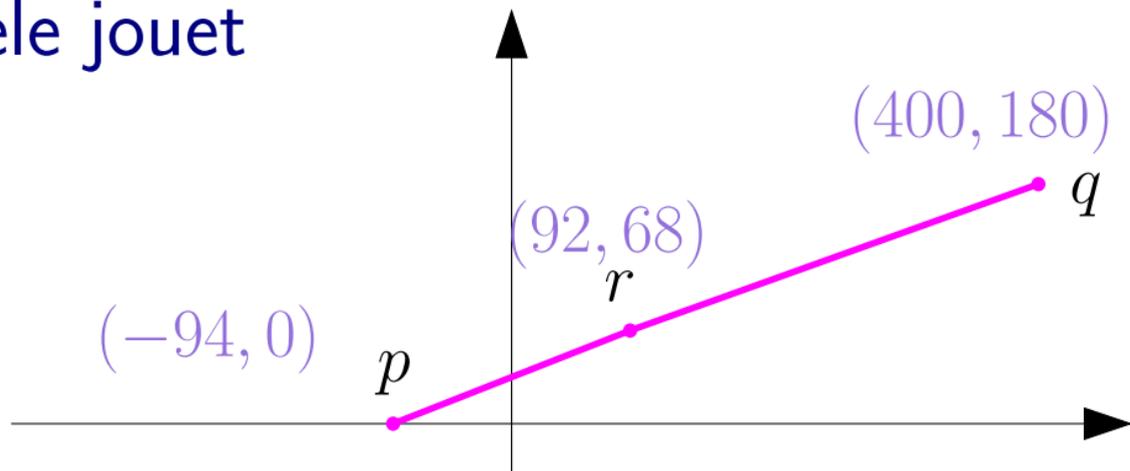
$$(400 + 94) * 68 - (92 + 94) * 180 =$$

$$494 * 68 - 186 * 180$$

$$490 * 68 - 190 * 180$$

$$= 33320 - 34200$$

# modèle jouet



$$(400 + 94) * 68 - (92 + 94) * 180 =$$

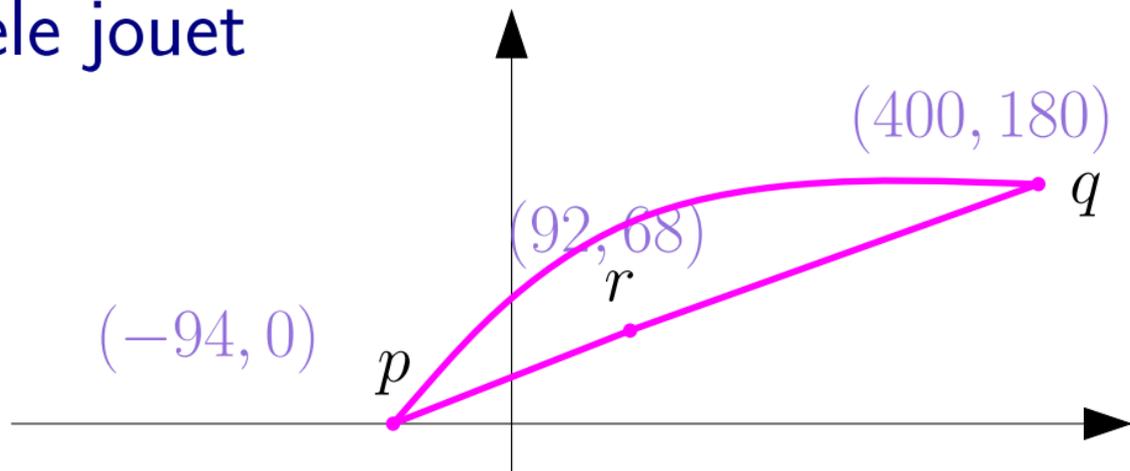
$$494 * 68 - 186 * 180$$

$$490 * 68 - 190 * 180$$

$$= 33320 - 34200$$

$$= 33000 - 34000$$

# modèle jouet



$$(400 + 94) * 68 - (92 + 94) * 180 =$$

$$494 * 68 - 186 * 180$$

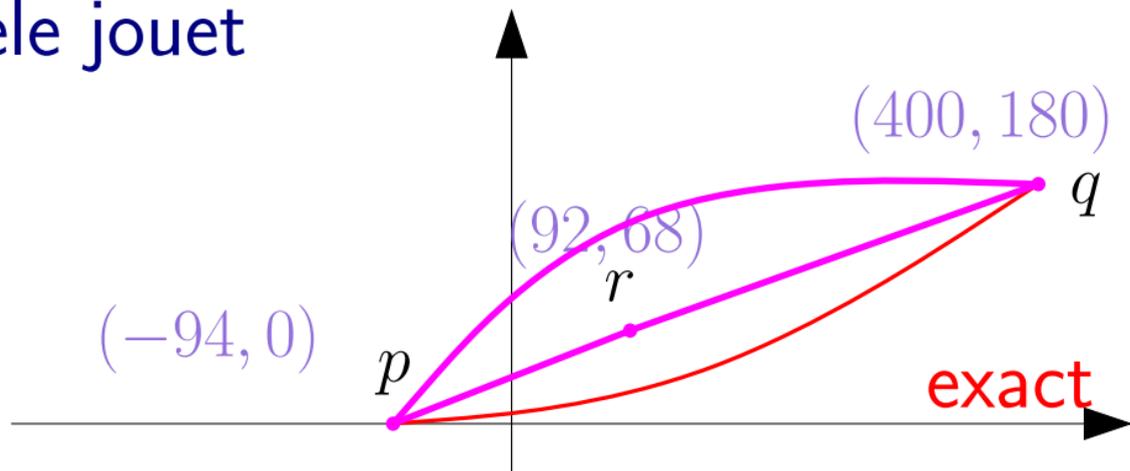
$$490 * 68 - 190 * 180$$

$$= 33320 - 34200$$

$$= 33000 - 34000$$

$$= -1000$$

# modèle jouet



$$(400 + 94) * 68 - (92 + 94) * 180 = 112$$

$$494 * 68 - 186 * 180$$

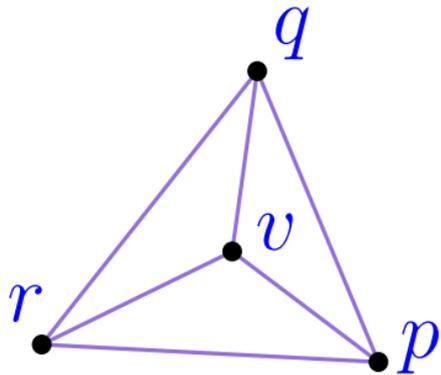
$$490 * 68 - 190 * 180$$

$$= 33320 - 34200$$

$$= 33000 - 34000$$

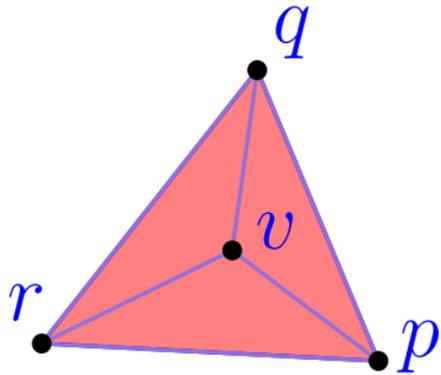
$$= -1000$$

# théorème géométrique utile aux algorithmes



$$\begin{array}{l} pqv \\ qrv \\ rpv \end{array} \text{ CCW} \implies pqr$$

# théorème géométrique utile aux algorithmes



$pqv$   
 $qrv$   
 $rpv$  **CCW**



$pqr$  **CCW**

modèle jouet

modèle jouet

Ça fait planter les théorèmes

# modèle jouet

$$\begin{array}{ccc} & & r \\ & v & \cdot (92, 68) \\ (-94, 0) & \cdot & \\ & p & \cdot (-5, 34) \end{array}$$

$$\cdot q (400, 180)$$

# modèle jouet

•  $q$  (400, 180)

$(-94, 0)$  •  $p$  •  $v$  •  $r$  (92, 68)  
•  $(-5, 34)$

$pqv$   
 $qrv$  **CCW**  
 $rpv$

# modèle jouet

•  $q$  (400, 180)

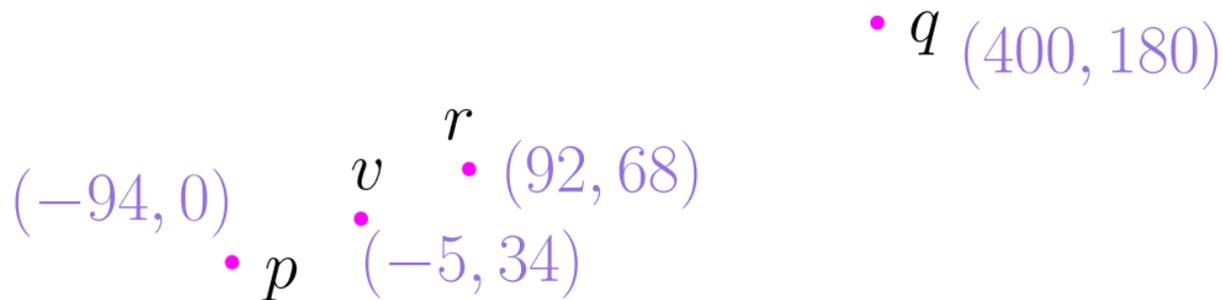
$(-94, 0)$  •  $p$      $v$  •  $(-5, 34)$      $r$  •  $(92, 68)$

$pqv$   
 $qrv$  **CCW**  
 $rpv$

$\implies$

$pqr$  **CCW**

# modèle jouet



$$\begin{array}{l} pqv \\ qrv \\ rpv \end{array} \text{ CCW} \implies pqr \text{ CCW}$$

calcul "jouet"

# modèle jouet

Ça fait planter les théorèmes •  $q$  (400, 180)

$$\begin{array}{ccc} & & r \\ & & \bullet (92, 68) \\ (-94, 0) & v & \\ \bullet p & & \bullet (-5, 34) \end{array}$$

$$\begin{array}{l} pqv \\ qrv \\ rpv \end{array} \text{ CCW} \implies pqr \text{ CCW}$$

$$\text{calcul "jouet"} \longrightarrow pqr \text{ CW}$$

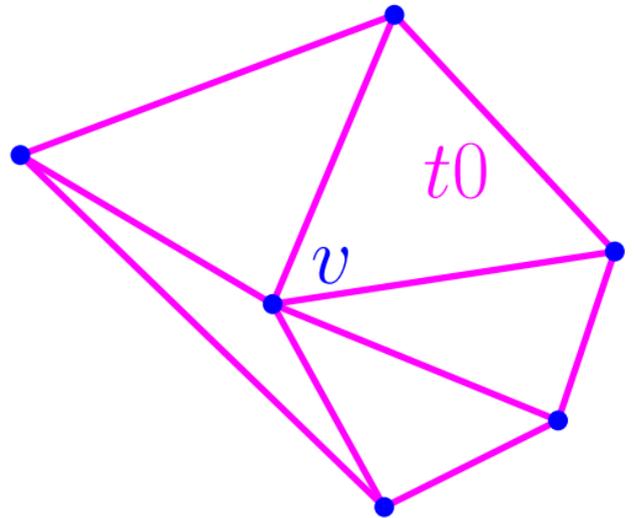
modèle jouet

modèle jouet

Ça fait planter les algorithmes

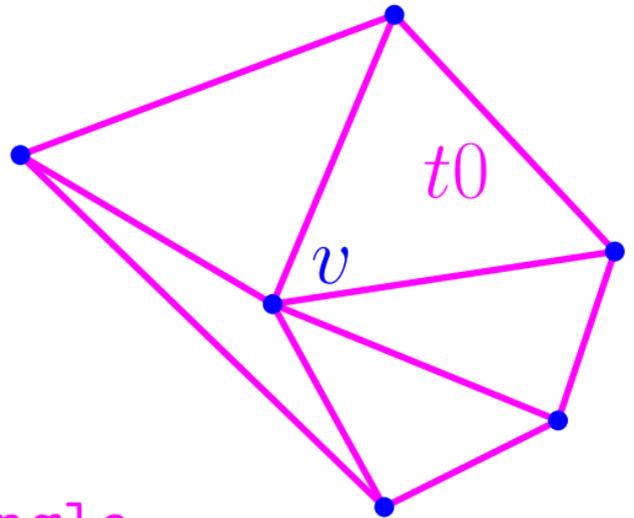
modèle jouet

tourner autour d'un sommet



modèle jouet

tourner autour d'un sommet



do

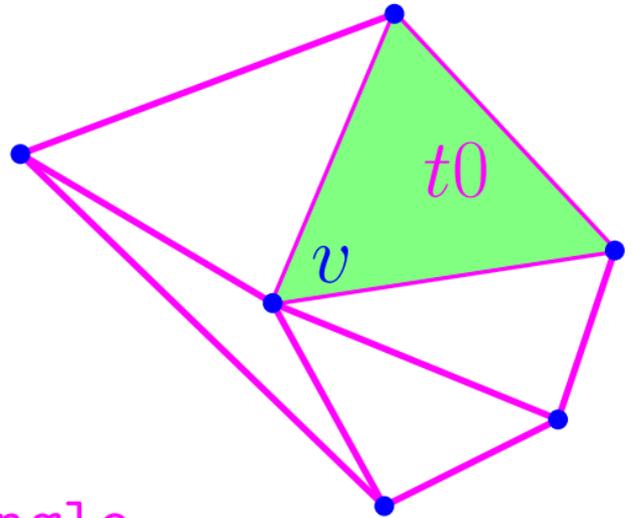
let  $vpq$  ccw triangle

go to neighbor through  $qv$

while triangle  $\neq t_0$

modèle jouet

tourner autour d'un sommet



do

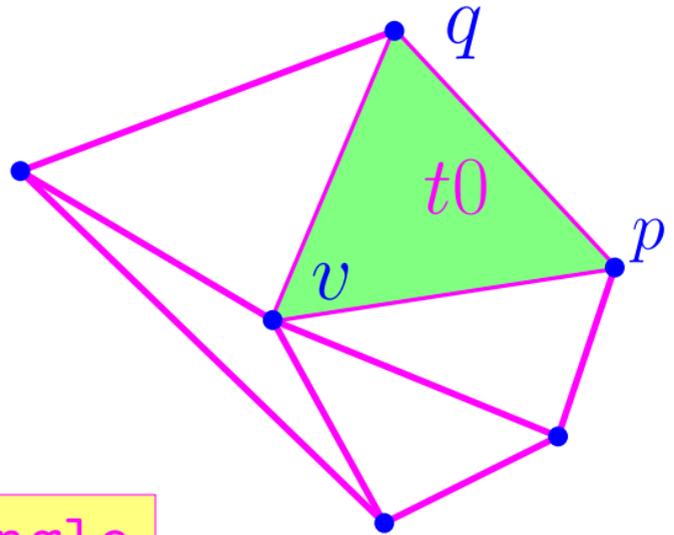
let  $vpq$  ccw triangle

go to neighbor through  $qv$

while triangle  $\neq t_0$

modèle jouet

tourner autour d'un sommet



do

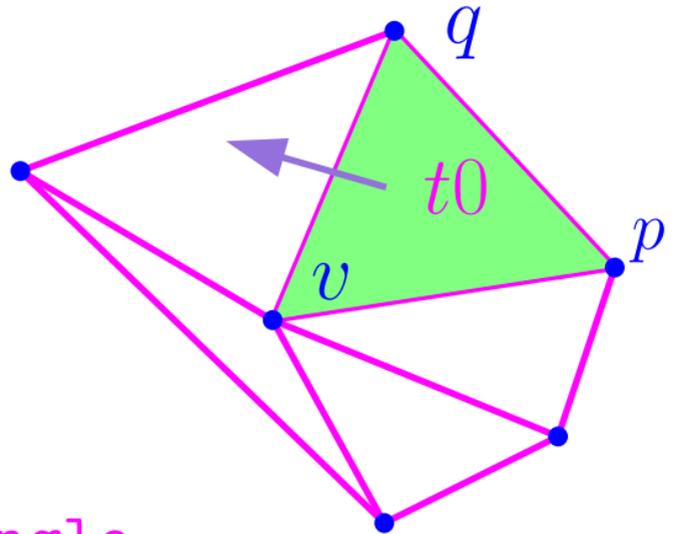
```
let  $vpq$  ccw triangle
```

```
go to neighbor through  $qv$ 
```

```
while triangle  $\neq t_0$ 
```

modèle jouet

tourner autour d'un sommet



do

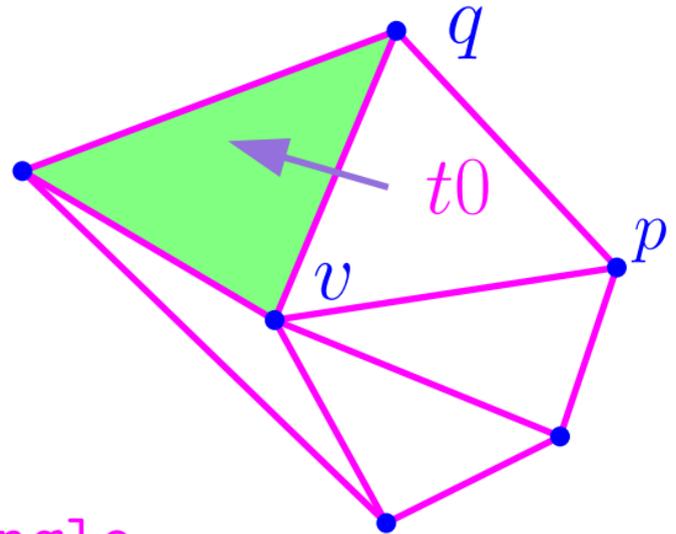
let  $vpq$  ccw triangle

go to neighbor through  $qv$

while triangle  $\neq t_0$

modèle jouet

tourner autour d'un sommet



do

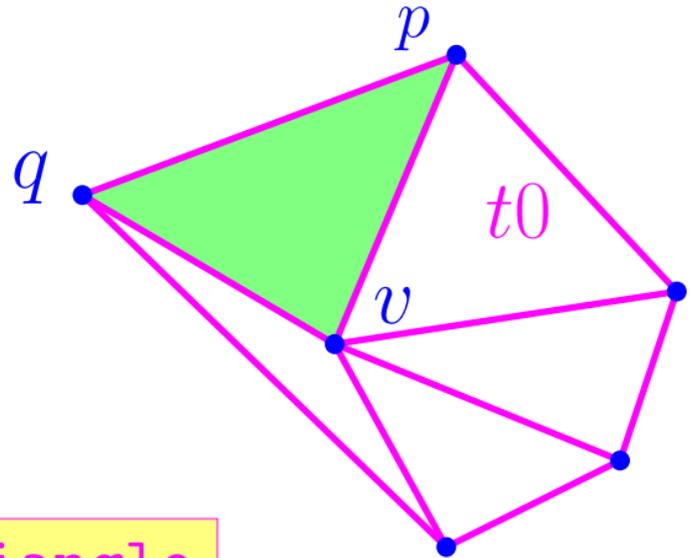
let  $vpq$  ccw triangle

go to neighbor through  $qv$

while triangle  $\neq t_0$

modèle jouet

tourner autour d'un sommet



do

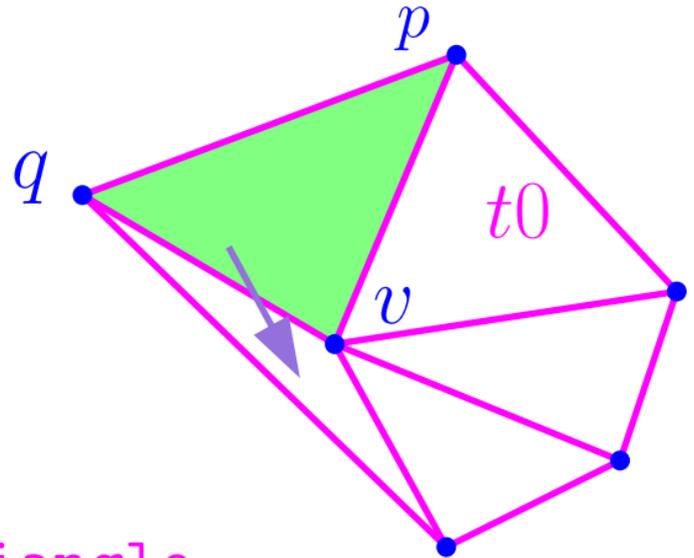
```
let  $vpq$  ccw triangle
```

```
go to neighbor through  $qv$ 
```

```
while triangle  $\neq t_0$ 
```

modèle jouet

tourner autour d'un sommet



do

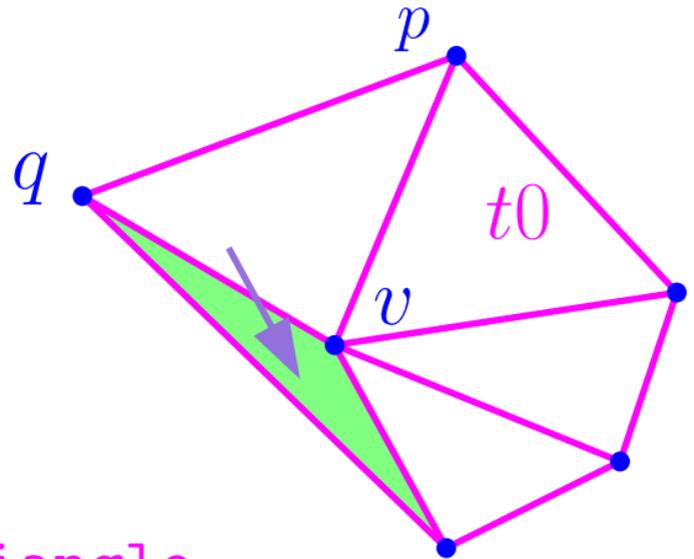
let  $vpq$  ccw triangle

go to neighbor through  $qv$

while triangle  $\neq t_0$

modèle jouet

tourner autour d'un sommet



do

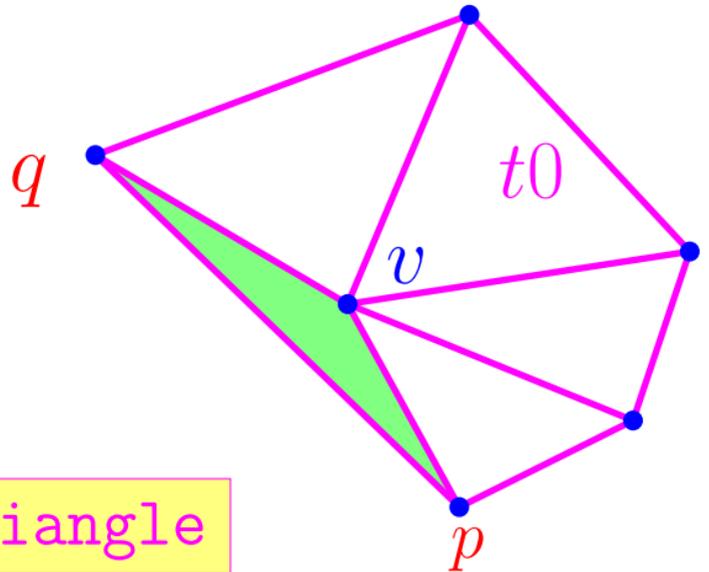
let  $vpq$  ccw triangle

go to neighbor through  $qv$

while triangle  $\neq t_0$

modèle jouet

tourner autour d'un sommet



do

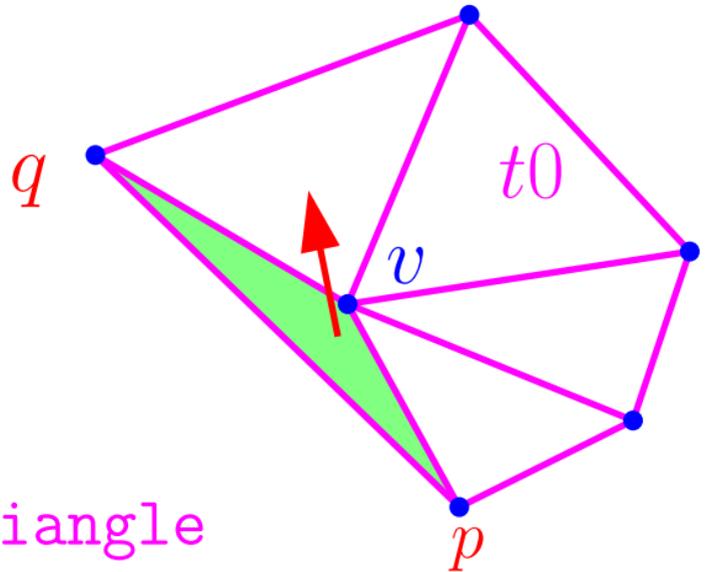
```
let  $vpq$  ccw triangle
```

```
go to neighbor through  $qv$ 
```

```
while triangle  $\neq t_0$ 
```

modèle jouet

tourner autour d'un sommet



do

let  $vpq$  ccw triangle

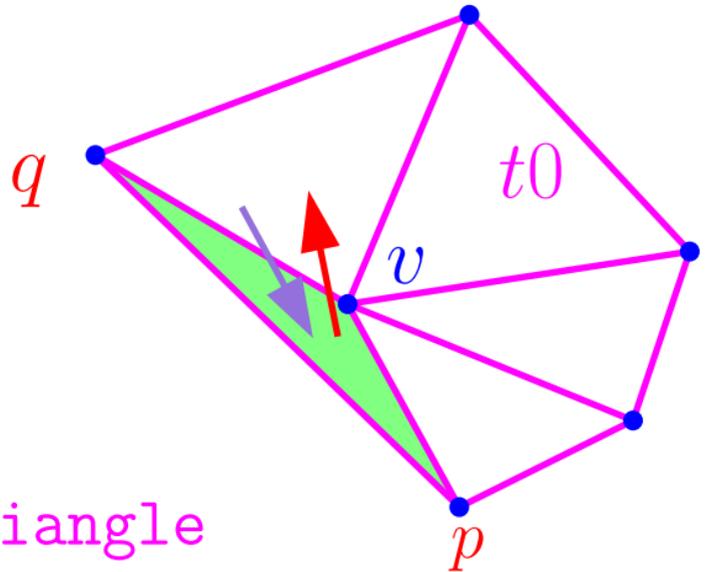
go to neighbor through  $qv$

while triangle  $\neq t_0$

modèle jouet

tourner autour d'un sommet

Ça boucle



do

let  $vpq$  ccw triangle

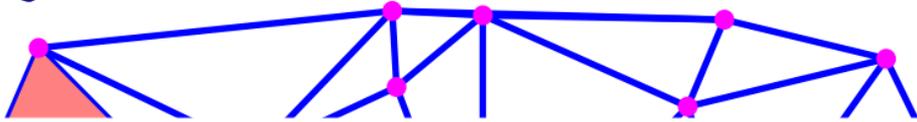
go to neighbor through  $qv$

while triangle  $\neq t_0$

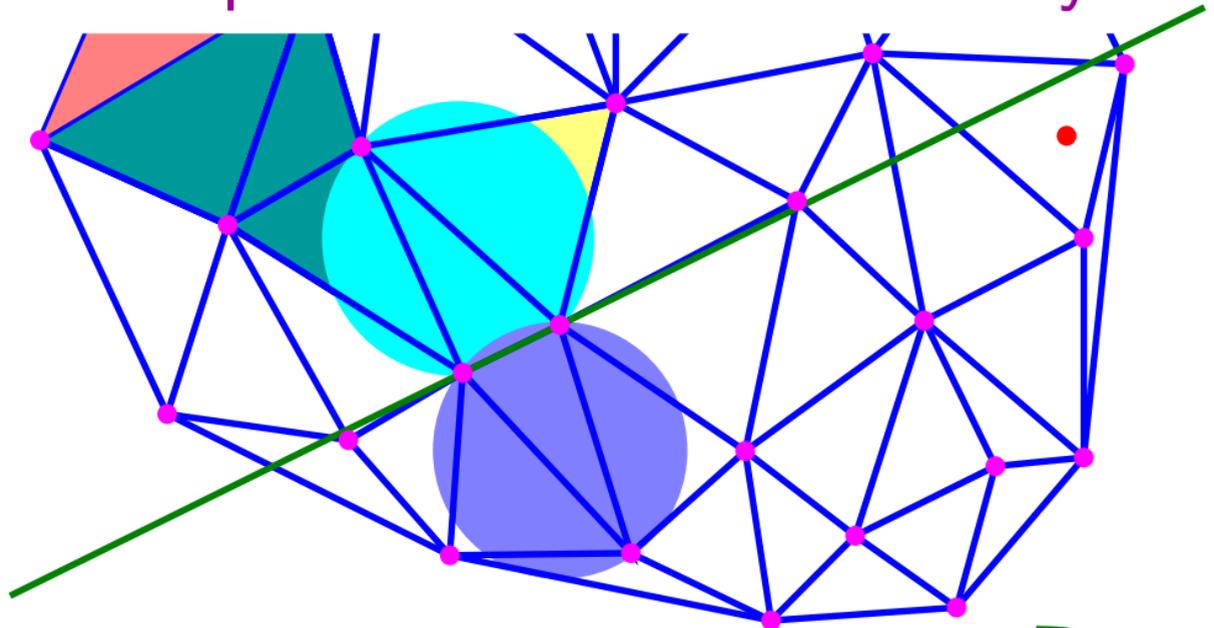
modèle jouet

Ça fait planter les algorithmes

modèle jouet



Marche par visibilité dans Delaunay



Pas de cycle dans ce cas. car puissance( $\bullet$ )



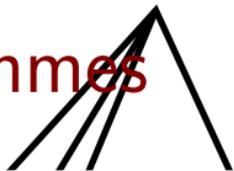
modèle jouet

Ça fait planter les algorithmes

ça boucle

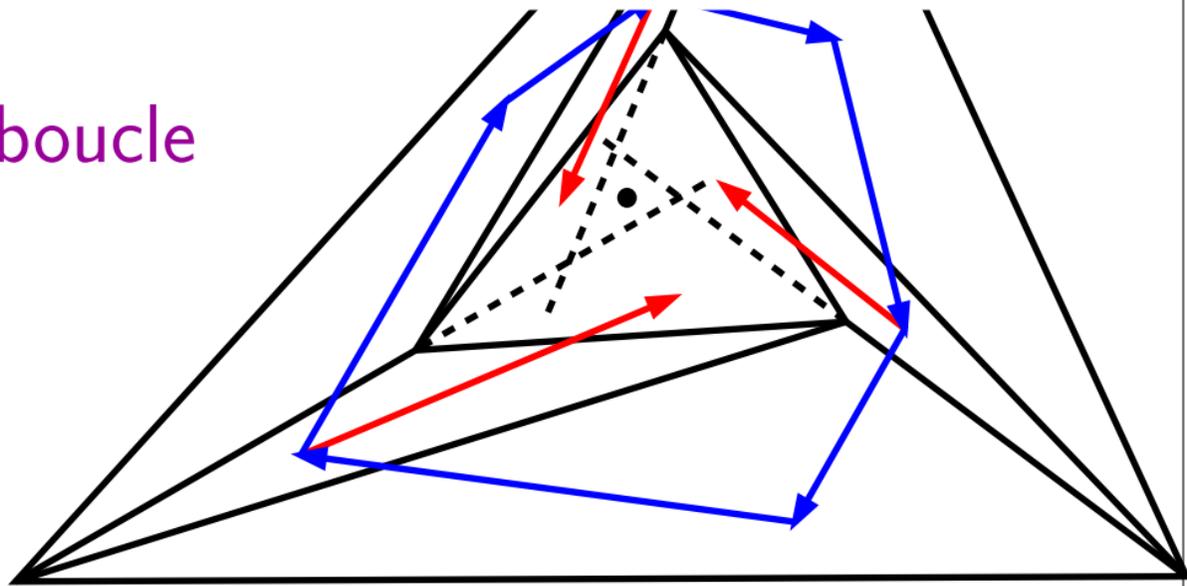
modèle jouet

Ça fait planter les algorithmes



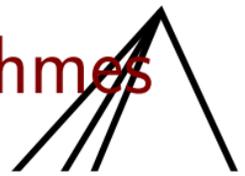
Marche par visibilité

ça boucle



modèle jouet

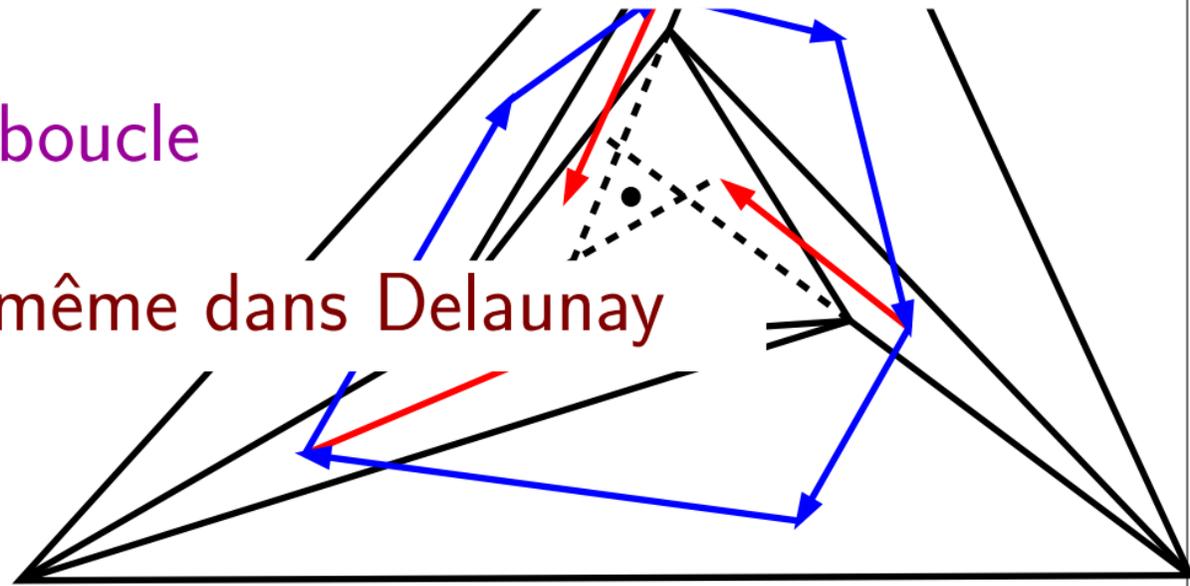
Ça fait planter les algorithmes



Marche par visibilité

ça boucle

même dans Delaunay



# Solution 1

# Solution 1

Oublions quelques

millénaires de géométrie

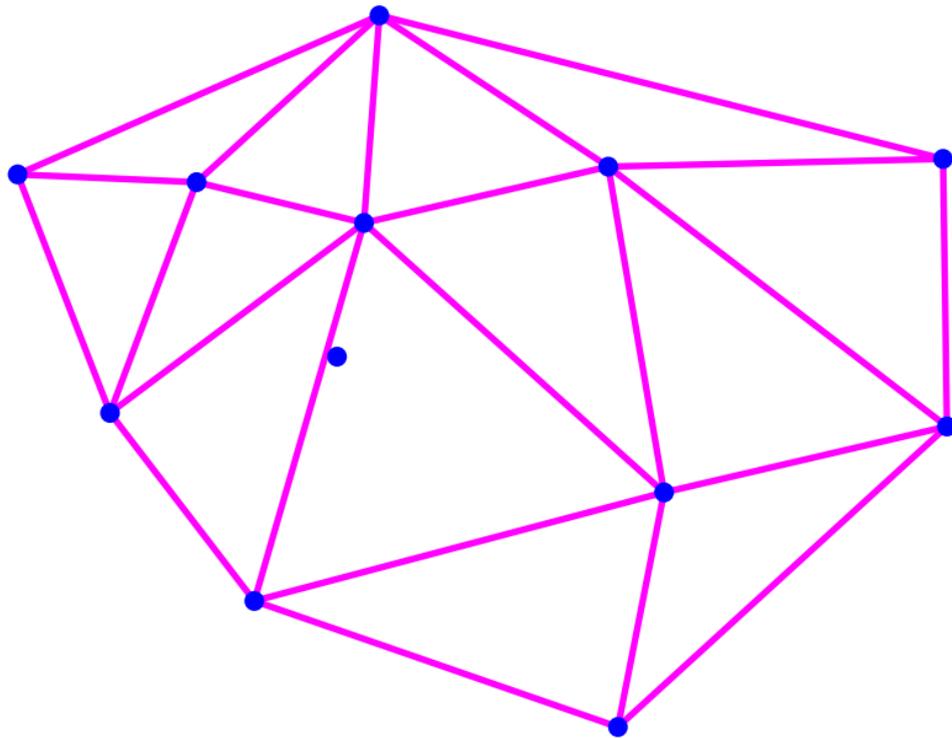
# Solution 1

Oublions quelques

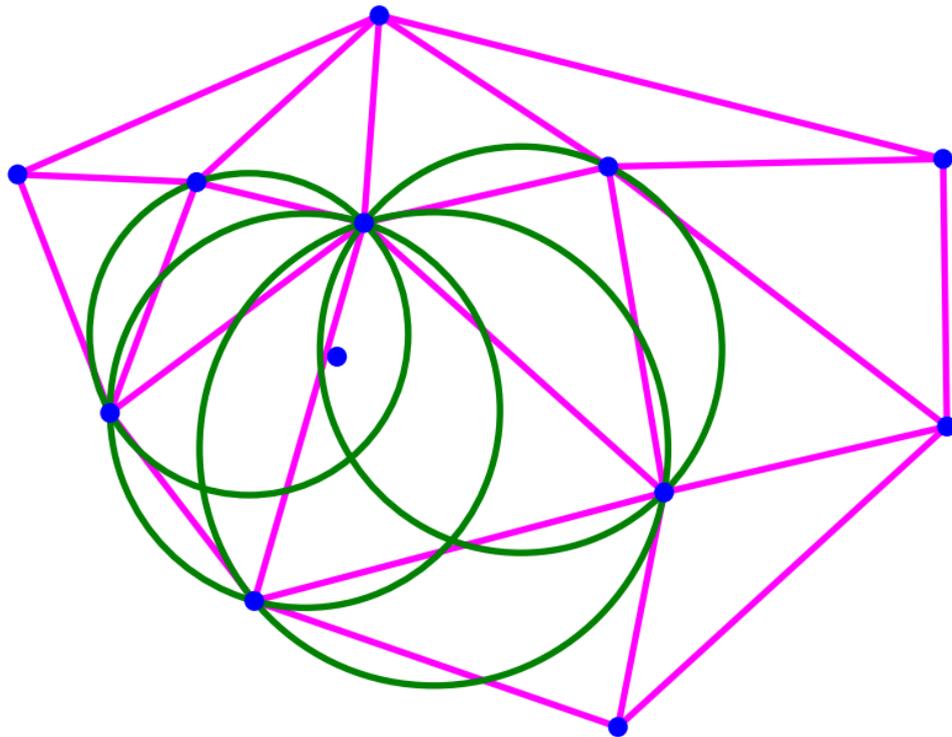
millénaires de géométrie

Concevons les algorithmes sans théorèmes

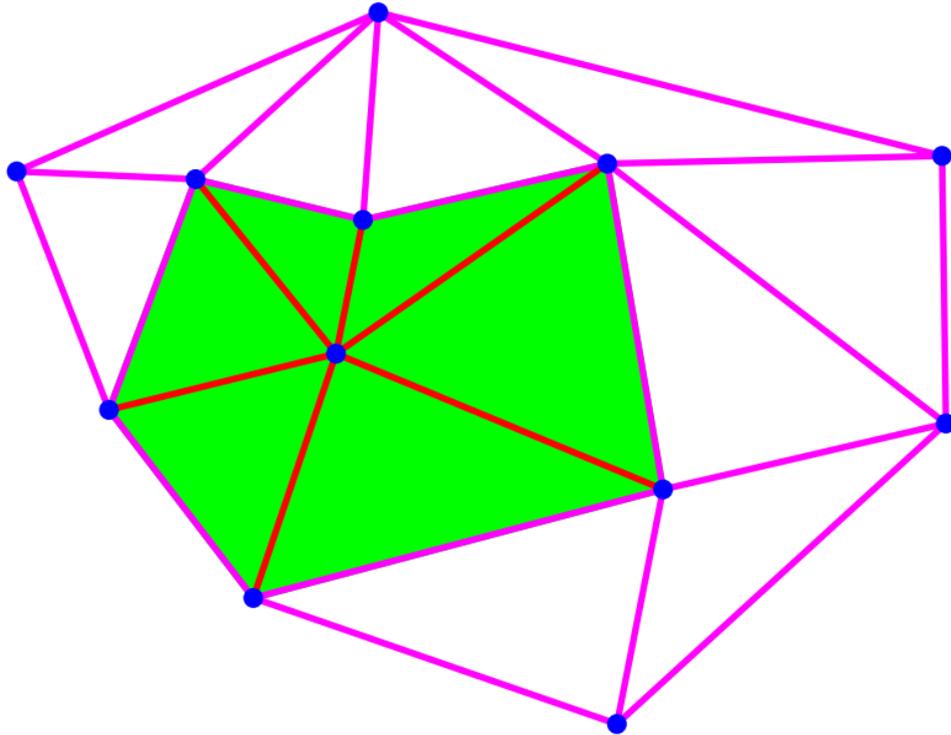
# Insertion dans Delaunay



# Insertion dans Delaunay



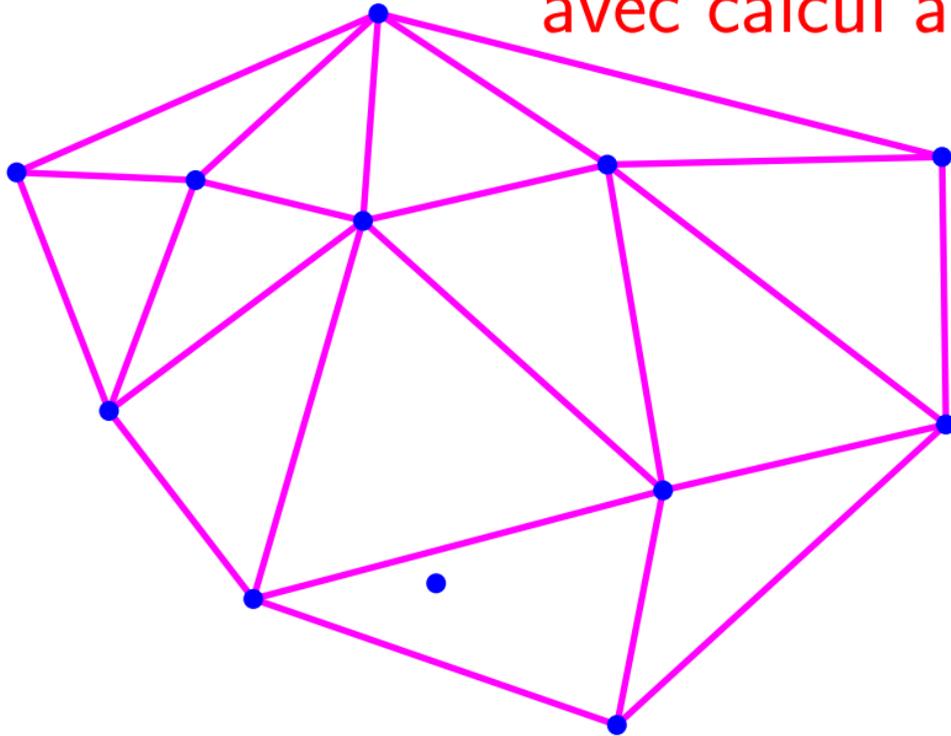
# Insertion dans Delaunay



est forcément étoilé

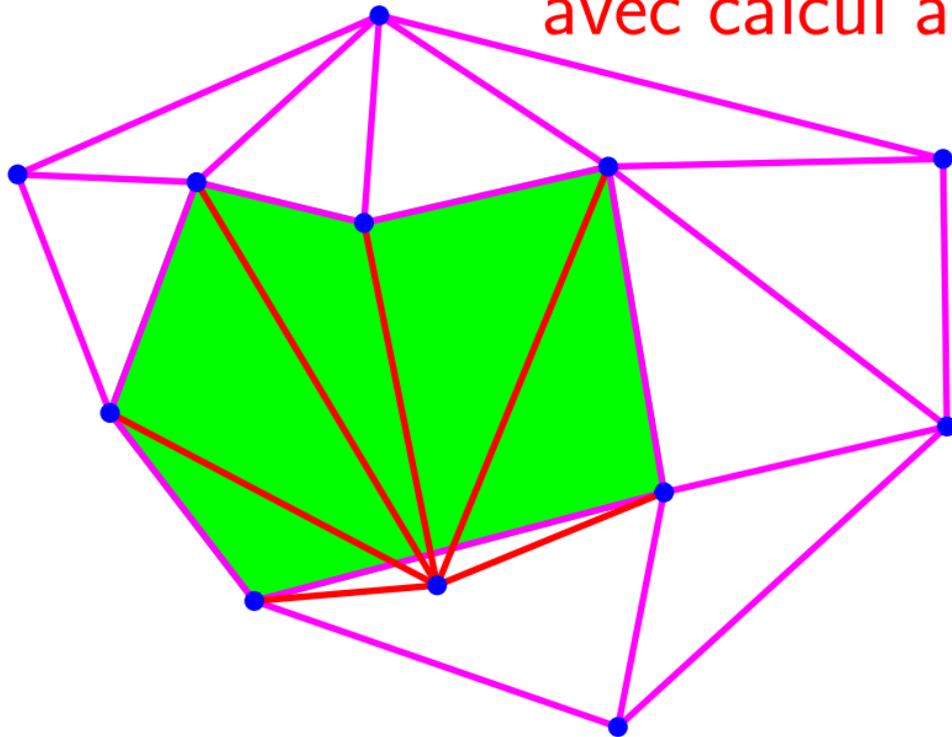
# Insertion dans Delaunay

avec calcul arrondi



# Insertion dans Delaunay

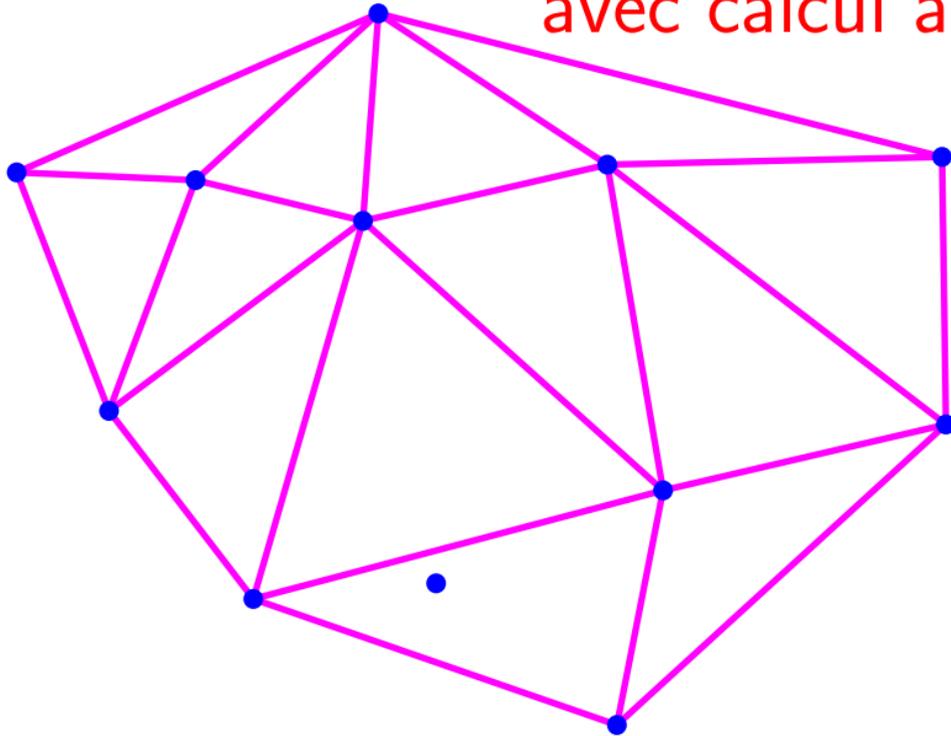
avec calcul arrondi



n'est pas forcément étoilé

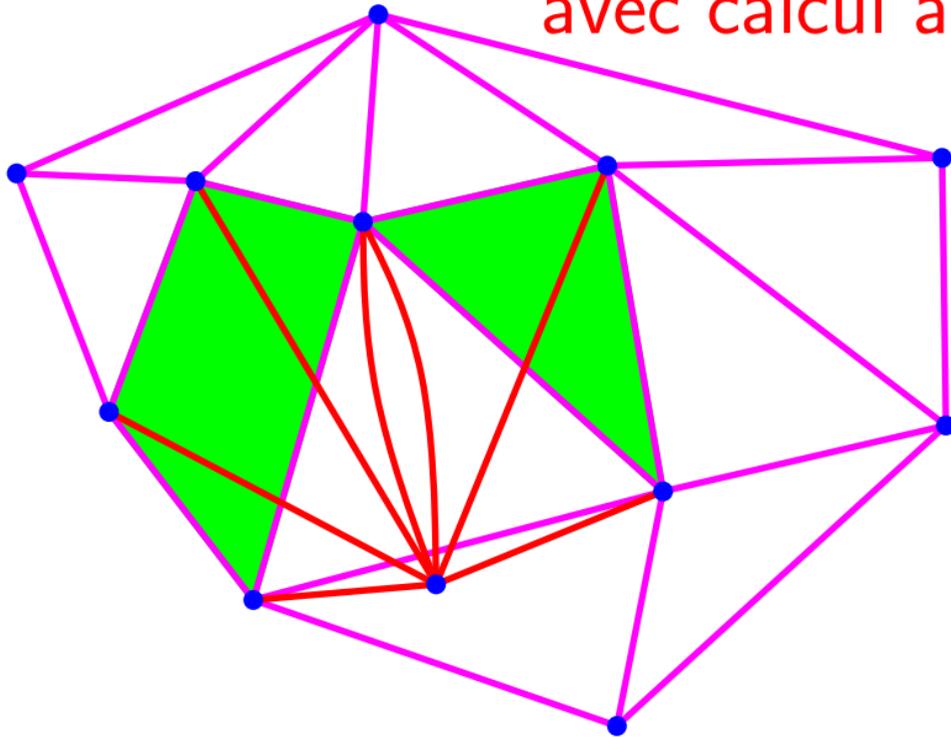
# Insertion dans Delaunay

avec calcul arrondi



# Insertion dans Delaunay

avec calcul arrondi



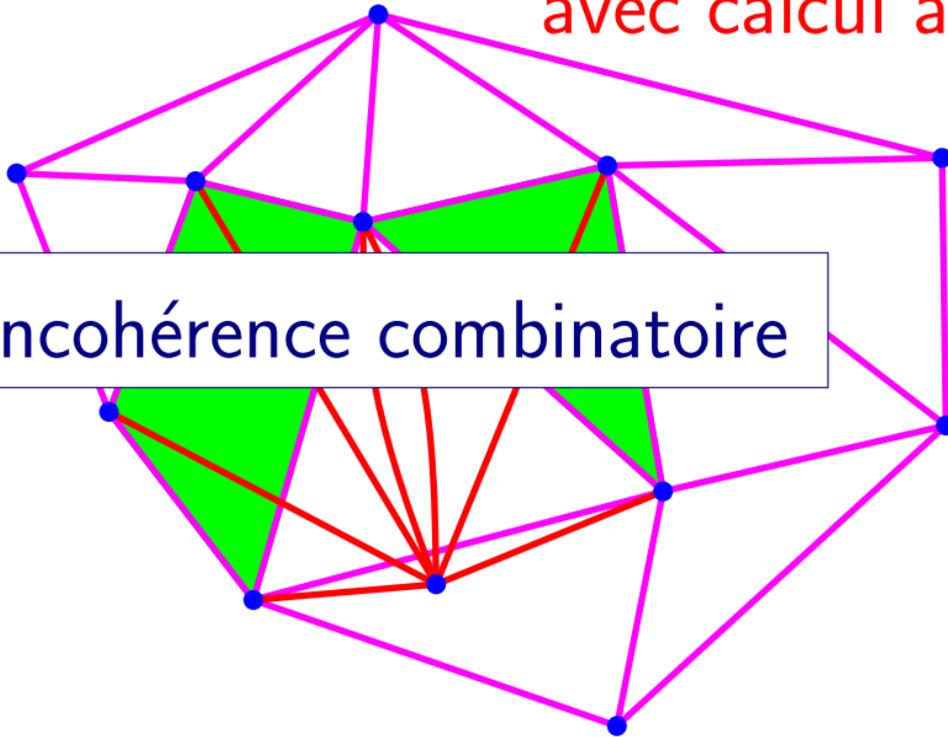
n'est pas forcément étoilé

# Insertion dans Delaunay

avec calcul arrondi

Incohérence combinatoire

n'est pas forcément étoilé



# Solution 1

Concevons les algorithmes sans théorèmes

# Solution 1

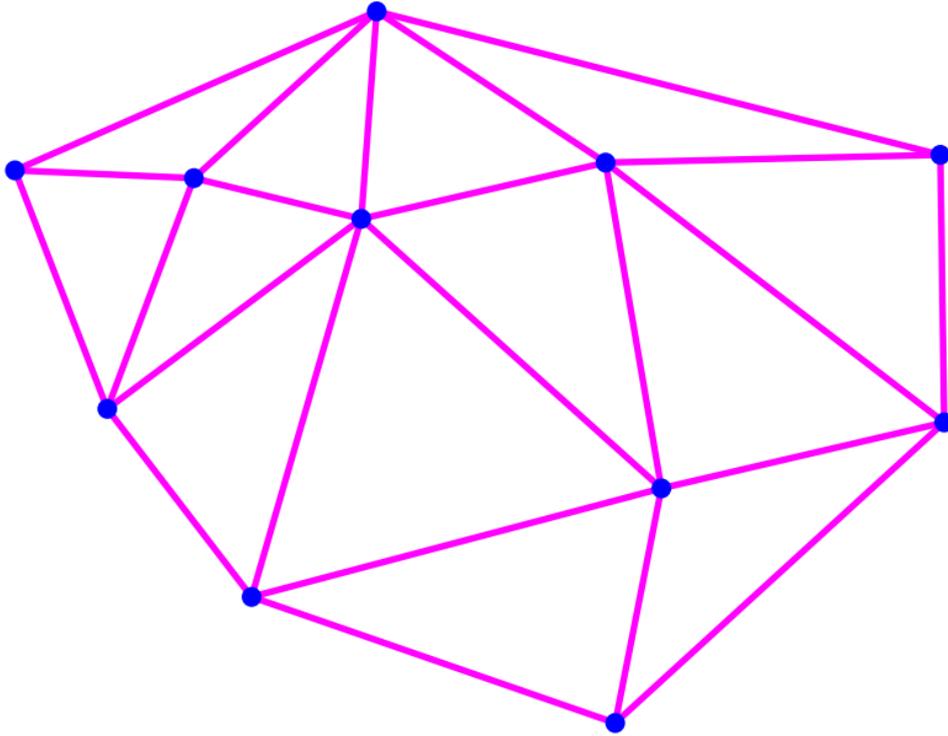
Concevons les algorithmes sans théorèmes

Vérifions la cohérence combinatoire

# Solution 1

Concevons les algorithmes sans théorèmes

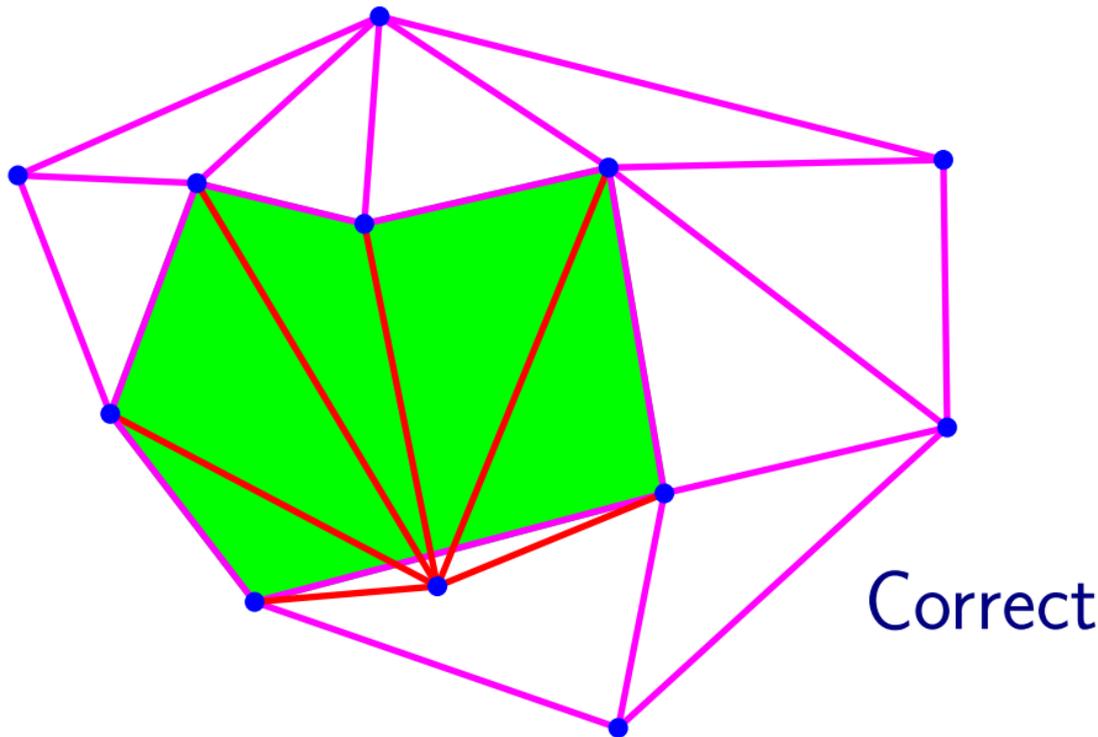
Vérifions la cohérence combinatoire



# Solution 1

Concevons les algorithmes sans théorèmes

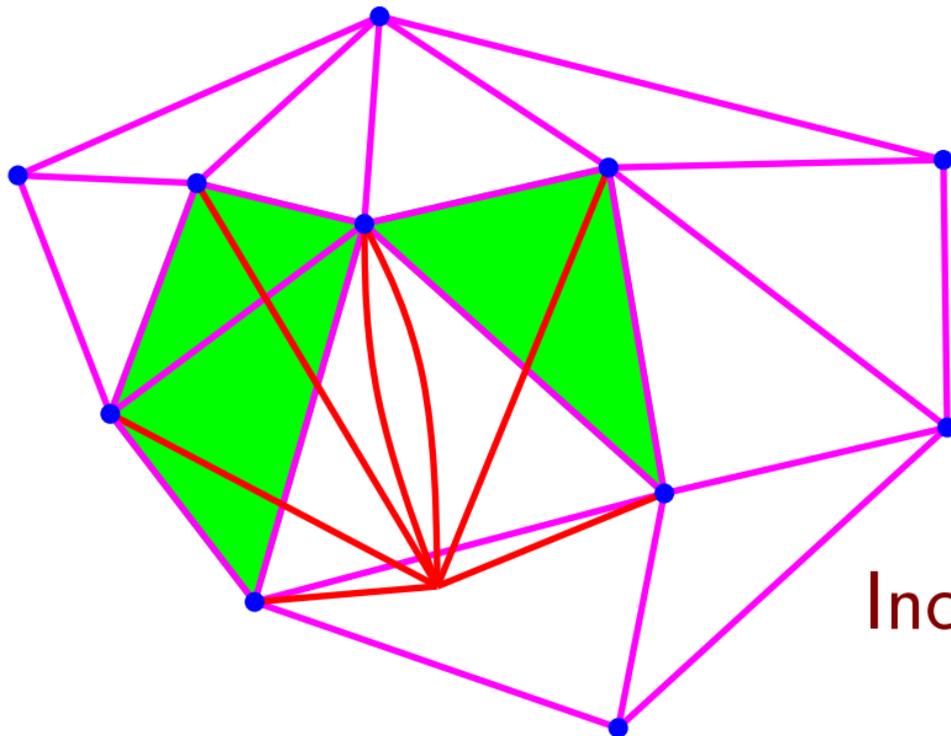
Vérifions la cohérence combinatoire



# Solution 1

Concevons les algorithmes sans théorèmes

Vérifions la cohérence combinatoire



Incorrect

# Solution 1

En vérifiant la cohérence combinatoire

# Solution 1

En vérifiant la cohérence combinatoire

On obtient au mieux  
une correction combinatoire

# Solution 1

En vérifiant la cohérence combinatoire

On obtient au mieux

une correction combinatoire

ce qu'on calcule est

combinatoirement

une triangulation

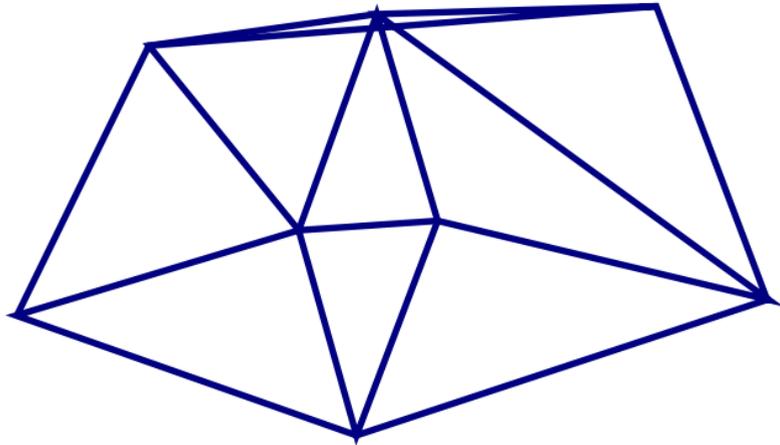
ce qu'on calcule est **combinatoirement**  
une triangulation

ce qu'on calcule est **combinatoirement**  
une triangulation

le plongement peut avoir des intersections

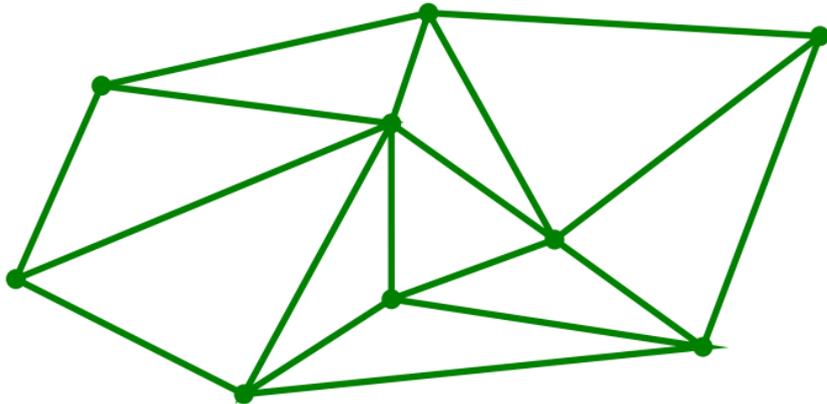
ce qu'on calcule est **combinatoirement**  
une triangulation

le plongement peut avoir des intersections



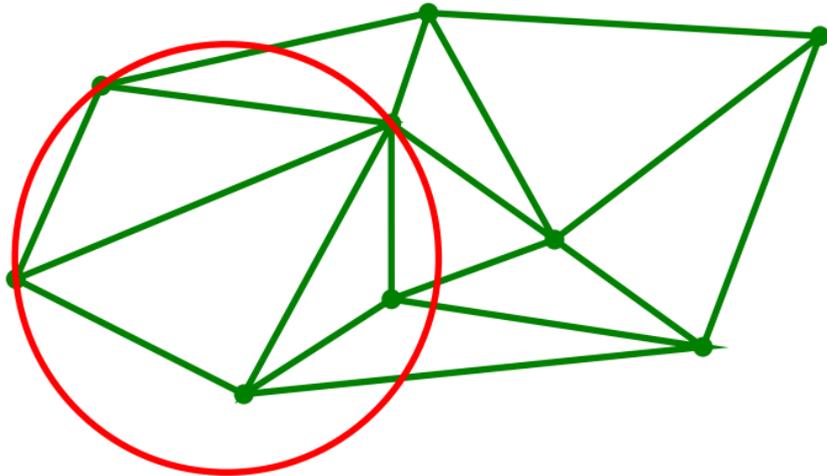
ce qu'on calcule est **combinatoirement**  
une triangulation

le plongement peut avoir des intersections  
c'est pas Delaunay



ce qu'on calcule est **combinatoirement**  
une triangulation

le plongement peut avoir des intersections  
c'est pas Delaunay



On vérifie que l'on a

combinatoirement

une triangulation

On vérifie que l'on a

combinatoirement

une triangulation

Peut-être que c'est

combinatoirement

pas Delaunay

On vérifie que l'on a

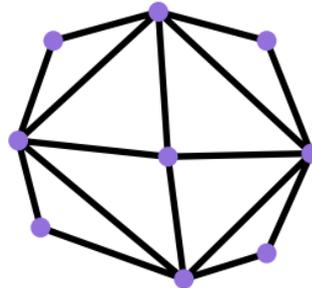
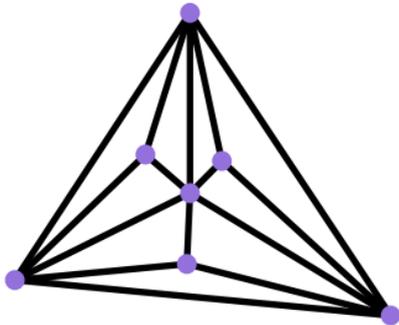
combinatoirement

une triangulation

Peut-être que c'est

combinatoirement

pas Delaunay



## Solution 1

Concevons les algorithmes sans théorèmes

Vérifions la cohérence combinatoire

~~Solution 1~~

~~Concevons les algorithmes sans théorèmes~~

~~Vérifions la cohérence combinatoire~~

peu (plus) utilisé

## Solution 2

Concevons les algorithmes avec théorèmes

Faisons comme si on savait calculer sur les réels

## Solution 2

Concevons les algorithmes avec théorèmes

Faisons comme si on savait calculer sur les réels

On a un sous-ensemble manipulable des réels  
par ex: les rationnels

## Solution 2

Concevons les algorithmes avec théorèmes

Faisons comme si on savait calculer sur les réels

On a un sous-ensemble manipulable des réels

par ex: les rationnels

algébriques ?

## Solution 2

Concevons les algorithmes avec théorèmes

Faisons comme si on savait calculer sur les réels

On a un sous-ensemble manipulable des réels  
par ex: les rationnels  
algébriques ?

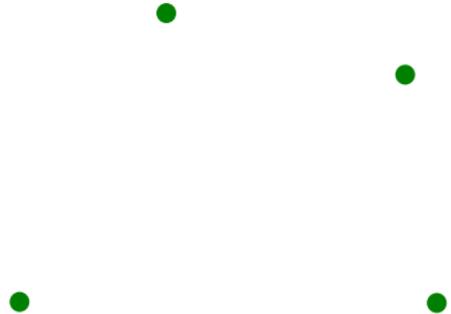
Calcul exact

# Prédicats versus constructions

# Prédicats versus constructions

Prédicat

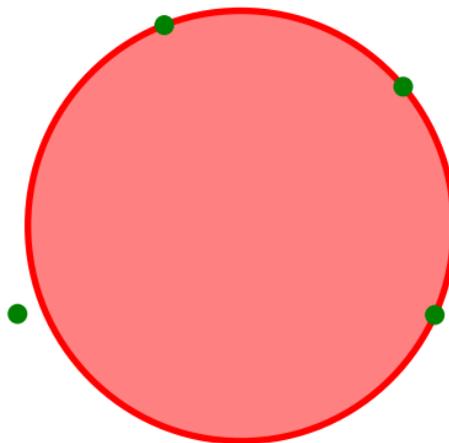
test cocircularité



# Prédicats versus constructions

Prédicat

test cocircularité



Décision (immédiate)

# Prédicats versus constructions

## Construction

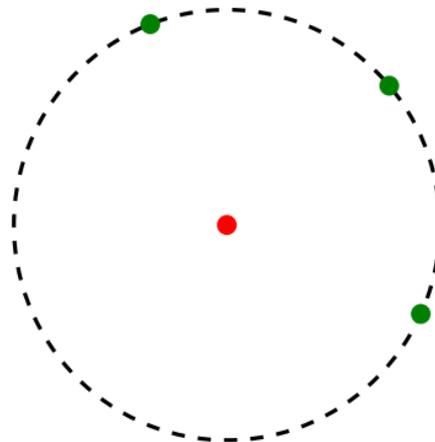
centre cercle circonscrit



# Prédicats versus constructions

Construction

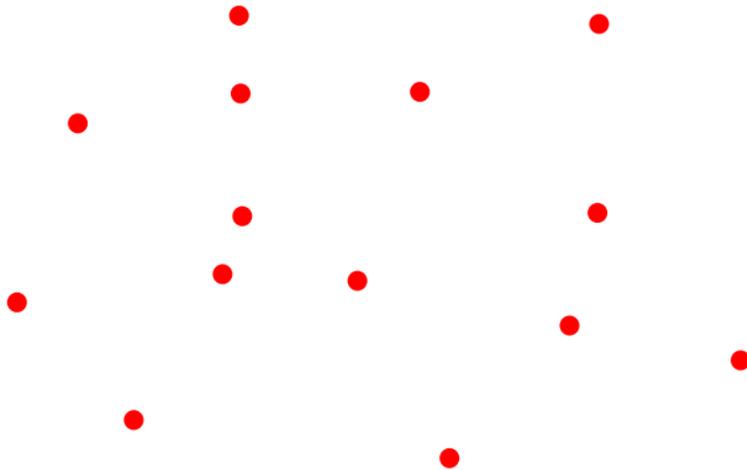
centre cercle circonscrit



Nouvel objet réutilisable

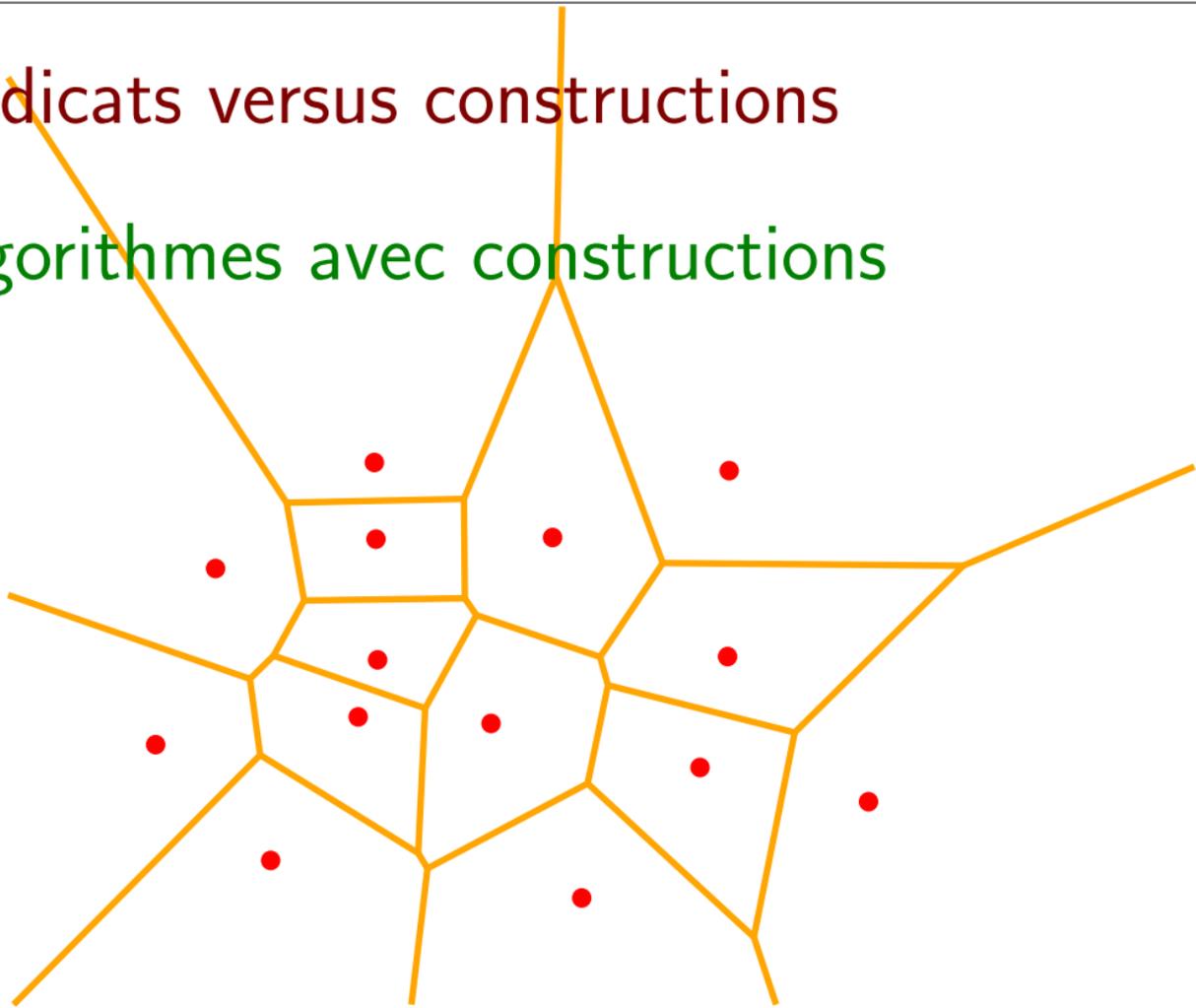
(décision ultérieure)

# Prédicats versus constructions



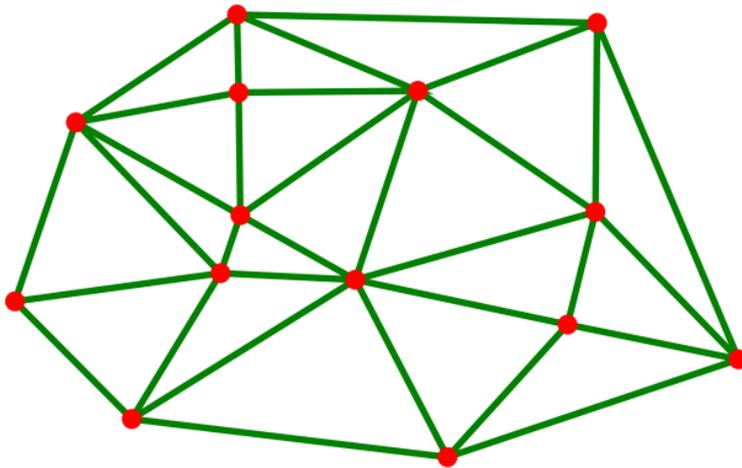
# Prédicats versus constructions

## Algorithmes avec constructions



# Prédicats versus constructions

## Algorithmes avec prédicats



# Algorithmes avec prédicats

Calcul exact

# Algorithmes avec prédicats

Calcul exact

nécessaire



# Algorithmes avec prédicats

Calcul exact

Paradigme du calcul exact

# Algorithmes avec prédicats

Répondre exactement aux prédicats

mais pas forcément du calcul exact

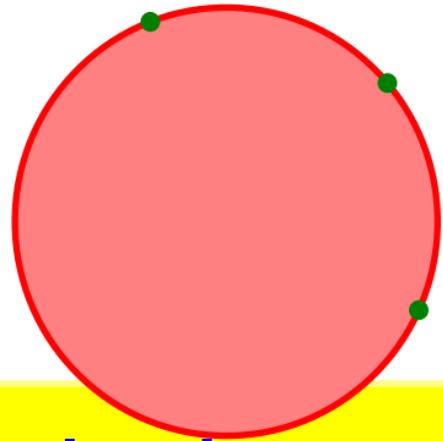
Paradigme du calcul exact

# Algorithmes avec prédicats

Répondre exactement aux prédicats

mais pas forcément du calcul exact

test cocircularité



Paradigme du calcul exact

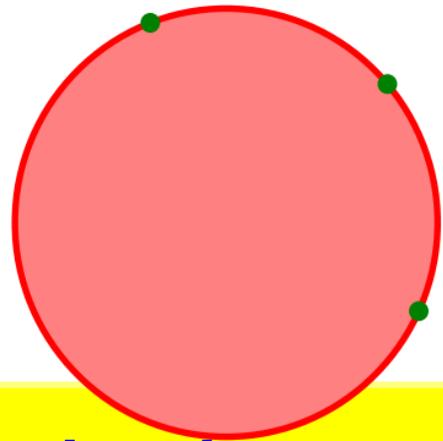
# Algorithmes avec prédicats

Répondre exactement aux prédicats

mais pas forcément du calcul exact

test cocircularité

calcul exact superflu •



Paradigme du calcul exact

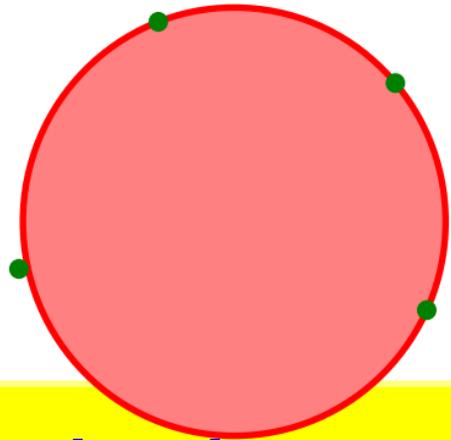
# Algorithmes avec prédicats

Répondre exactement aux prédicats

mais pas forcément du calcul exact

test cocircularité

calcul exact nécessaire



Paradigme du calcul exact

calcul exact nécessaire

pour la cohérence des résultats

calcul exact nécessaire

pour la cohérence des résultats

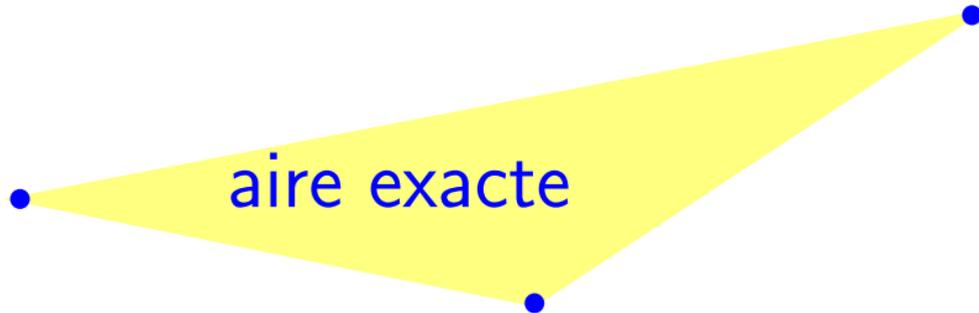
pas pour l'exactitude

# Répondre exactement aux prédicats



# Répondre exactement aux prédicats

Calcul exact



# Répondre exactement aux prédicats

Calcul exact

Calcul approché



# Répondre exactement aux prédicats

Calcul exact

Calcul approché certifié



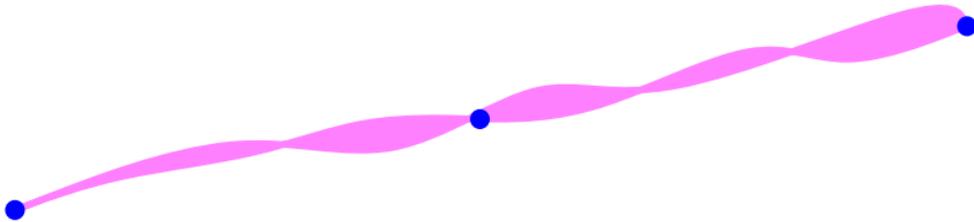
# Répondre exactement aux prédicats



Répondre exactement aux prédicats

et si c'est pas sur

on peut se rabattre sur

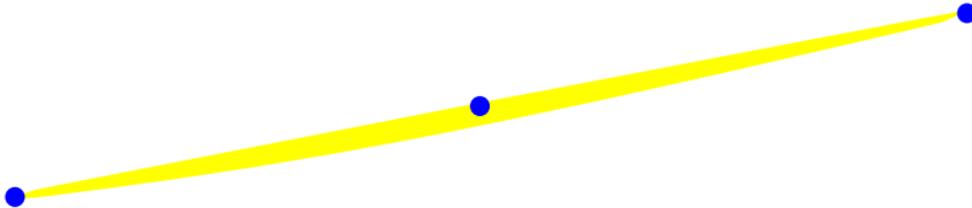


Répondre exactement aux prédicats

et si c'est pas sur

on peut se rabattre sur

le calcul exact



Comment certifier un calcul approché ?

# Comment certifier un calcul approché ?

calcul d'erreur

$x + y$



# Comment certifier un calcul approché ?

calcul d'erreur

$$x \oplus y = x+y + \text{erreur}$$

# Comment certifier un calcul approché ?

calcul d'erreur

$$x \oplus y = x+y + \text{erreur}$$

$x + y$  exact


$$0.11010 \dots 010010101001 \times 2^e$$

# Comment certifier un calcul approché ?

calcul d'erreur

$$x \oplus y = x+y + \text{erreur}$$

IEEE 754

$x + y$  exact

0.11010...010110101001  $\times 2^e$

53 bits

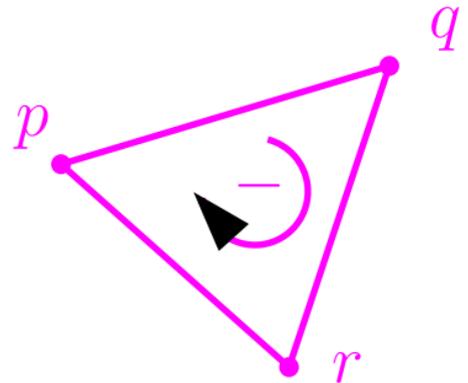
$x \oplus y$

erreur  $\leq |x+y| 2^{-54}$

# Filtre statique

exemple : prédicat d'orientation

$$\begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix}$$

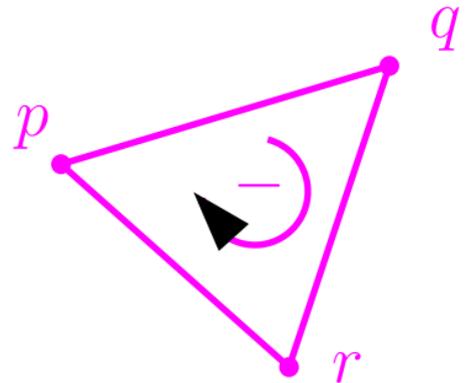


# Filtre statique

exemple : prédicat d'orientation

$$\begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix}$$

$$|x_\star| |y_\star| \leq M$$



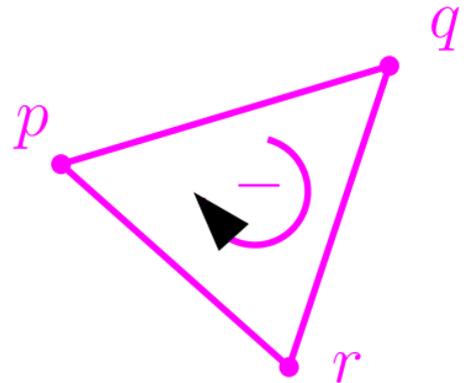
# Filtre statique

exemple : prédicat d'orientation

$$\begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix}$$

$$|x_\star| |y_\star| \leq M$$

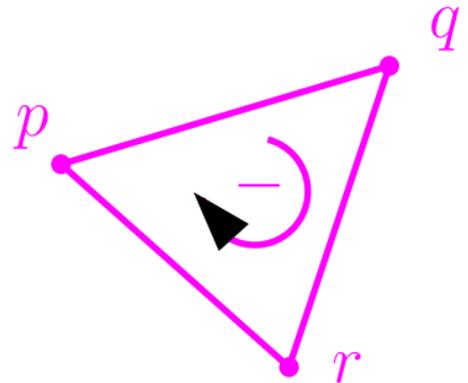
$$|x_\star - x_p| \leq 2M$$



# Filtre statique

exemple : prédicat d'orientation

$$\begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix}$$



$$|x_\star| |y_\star| \leq M$$

$$|x_\star - x_p| \leq 2M$$

$$\text{erreur}_{x_\star - x_p} \leq 2M 2^{-54} = 2^{-53} M$$

$$|x_\star| |y_\star| \leq M$$

$$\begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix}$$

$$|x_\star - x_p| \leq 2M$$

$$\text{erreur}_{x_\star - x_p} \leq 2M 2^{-54} = 2^{-53} M$$

$$|x_\star| |y_\star| \leq M$$

$$\begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix}$$

$$|x_\star - x_p| \leq 2M$$

$$\text{erreur}_{x_\star - x_p} \leq 2M 2^{-54} = 2^{-53} M$$

$$|(x_\star - x_p)(y_\star - y_p)| \leq 4M^2$$

$$|x_\star| |y_\star| \leq M$$

$$\begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix}$$

$$|x_\star - x_p| \leq 2M$$

$$\text{erreur}_{x_\star - x_p} \leq 2M 2^{-54} = 2^{-53} M$$

$$|(x_\star - x_p)(y_\star - y_p)| \leq 4M^2$$

$$\begin{aligned} \text{erreur}_{(x_\star - x_p)(y_\star - y_p)} &\leq 4M^2 2^{-53} + 2(2M \cdot 2^{-53} M) \\ &\leq 2^{-50} M^2 \end{aligned}$$

$$|(x_\star - x_p)(y_\star - y_p)| \leq 4M^2$$

$$\text{erreur}_{(x_\star - x_p)(y_\star - y_p)} \leq 2^{-50} M^2$$

$$|(x_\star - x_p)(y_\star - y_p)| \leq 4M^2$$

$$\text{erreur}_{(x_\star - x_p)(y_\star - y_p)} \leq 2^{-50} M^2$$

$$\left| \begin{array}{cc} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{array} \right| \leq 8M^2$$

$$|(x_\star - x_p)(y_\star - y_p)| \leq 4M^2$$

$$\text{erreur}_{(x_\star - x_p)(y_\star - y_p)} \leq 2^{-50} M^2$$

$$\left| \begin{array}{cc} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{array} \right| \leq 8M^2$$

$$\begin{aligned} \text{erreur} \left| \begin{array}{cc} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{array} \right| &\leq 8M^2 2^{-53} + 2 \cdot 2^{-50} M^2 \\ &\leq 3 \cdot 2^{-50} M^2 \end{aligned}$$

$$\begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix} \leq 8M^2$$

$$\text{erreur} \begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix} \leq 3.2^{-50} M^2$$

$$\begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix} \leq 8M^2$$

$$\text{erreur} \begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix} \leq 3.2^{-50} M^2$$

si  $|\text{valeur}| \geq \text{erreur}$

$$\begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix} \leq 8M^2$$

$$\text{erreur} \begin{vmatrix} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{vmatrix} \leq 3.2^{-50} M^2$$

si  $|\text{valeur}| \geq \text{erreur}$

le prédicat est certifié

$$\left| \begin{array}{cc} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{array} \right| \leq 8M^2$$

$$\text{erreur} \left| \begin{array}{cc} x_q - x_p & x_r - x_p \\ y_q - y_p & y_r - y_p \end{array} \right| \leq 3.2^{-50} M^2$$

si  $|\text{valeur}| \geq \text{erreur}$

le prédicat est certifié

sinon

calcul exact (entier)

# Filtre statique

Contre

Pour

# Filtre statique

**Contre**

Hypothèses sur les données

Opérations restreintes  $+$   $-$   $\times$   $\sqrt{\quad}$

**Pour**

# Filtre statique

**Contre**

Hypothèses sur les données

Opérations restreintes  $+$   $-$   $\times$   $\sqrt{\quad}$

Taux de succès correct

**Pour**

# Filtre statique

**Contre**

Hypothèses sur les données

Opérations restreintes  $+$   $-$   $\times$   $\sqrt{\quad}$

Taux de succès correct

**Très rapide**

**Pour**

# Filtre statique

## Contre

Hypothèses sur les données

Opérations restreintes  $+$   $-$   $\times$   $\sqrt{\quad}$

erreur à la main ?

Taux de succès correct

Très rapide

Pour (erreur calculée à la compilation)

# Filtre dynamique

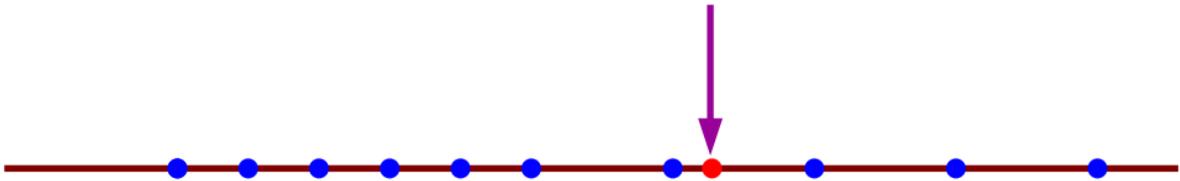
# Filtre dynamique

par exemple avec arithmétique d'intervalles

# Filtre dynamique

par exemple avec arithmétique d'intervalles

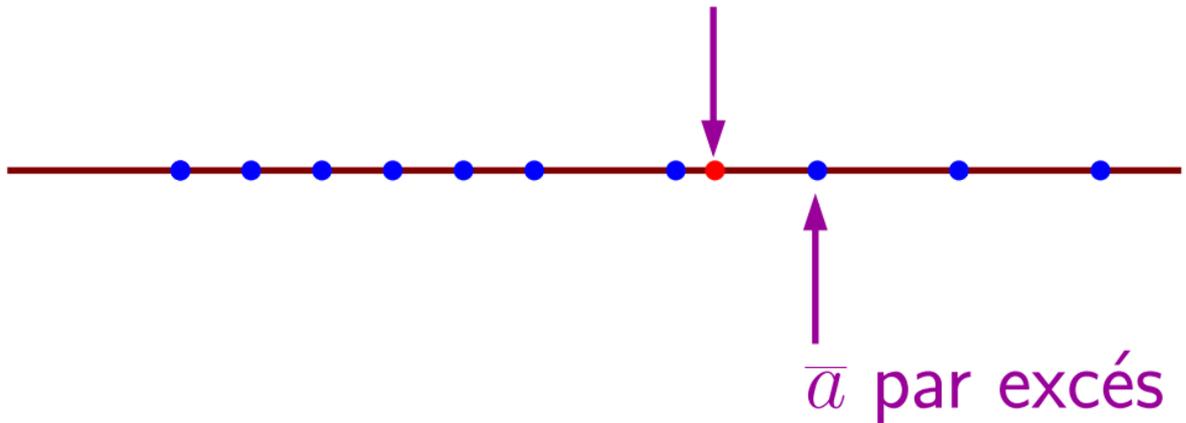
$a$  non représentable



# Filtre dynamique

par exemple avec arithmétique d'intervalles

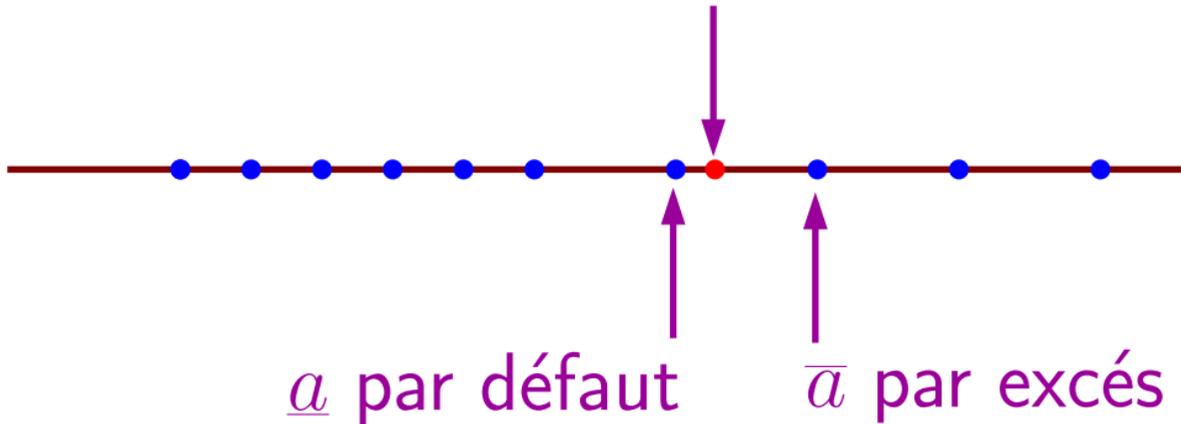
$a$  non représentable



# Filtre dynamique

par exemple avec arithmétique d'intervalles

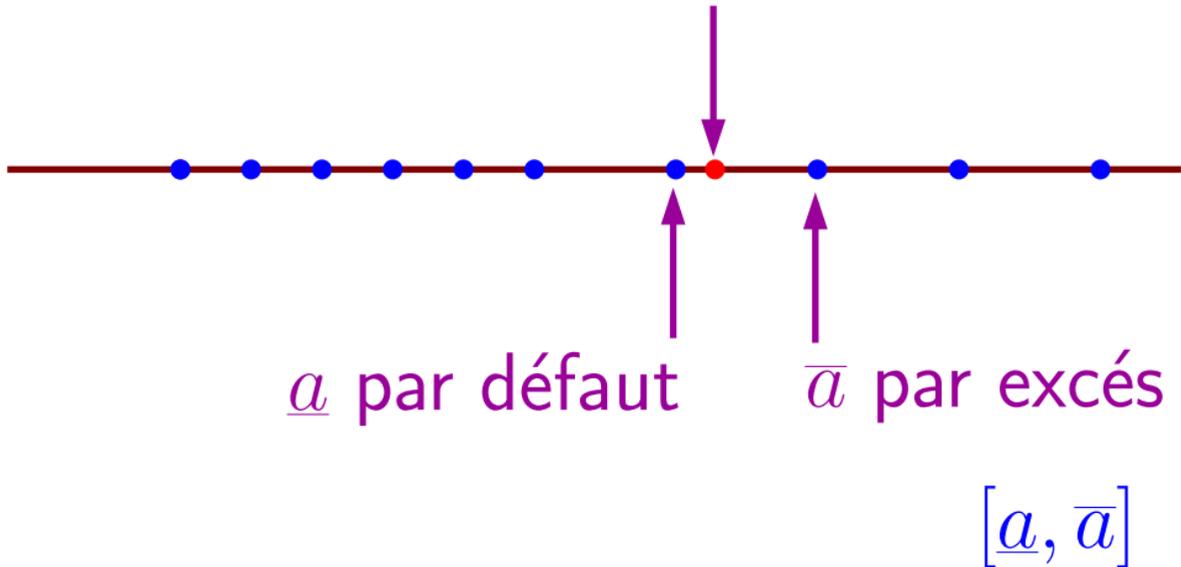
$a$  non représentable



# Filtre dynamique

par exemple avec arithmétique d'intervalles

$a$  non représentable



# arithmétique d'intervalles

$$[\underline{a}, \bar{a}] + [\underline{b}, \bar{b}] =$$

## arithmétique d'intervalles

$$[\underline{a}, \bar{a}] + [\underline{b}, \bar{b}] = [\underline{a} + \underline{b}, \bar{a} + \bar{b}]$$

## arithmétique d'intervalles

$$[\underline{a}, \bar{a}] + [\underline{b}, \bar{b}] = [\underline{a} + \underline{b}, \bar{a} + \bar{b}]$$

on propage l'approximation

## arithmétique d'intervalles

$$[\underline{a}, \overline{a}] + [\underline{b}, \overline{b}] = [\underline{a+\underline{b}}, \overline{a+\overline{b}}]$$

on propage l'approximation

si  $0 \notin [\underline{resultat}, \overline{resultat}]$

le prédicat est certifié

## arithmétique d'intervalles

$$[\underline{a}, \bar{a}] + [\underline{b}, \bar{b}] = [\underline{a} + \underline{b}, \bar{a} + \bar{b}]$$

on propage l'approximation

si  $0 \notin [\underline{resultat}, \overline{resultat}]$

le prédicat est certifié

sinon

calcul exact (entier)

# Programmation

```
template <class FT>
Orientation orientation
    ( FT px, FT py, FT qx, FT qy, FT rx, FT ry)
{
    return sign( (qx-px)*(ry-py)-(rx-px)*(qy-py) );
}
```

# Programmation

```
template <class FT>
Orientation orientation
    ( FT px, FT py, FT qx, FT qy, FT rx, FT ry)
{
    return sign( (qx-px)*(ry-py)-(rx-px)*(qy-py) );
}
```

FT peut être double

# Programmation

```
template <class FT>
Orientation orientation
    ( FT px, FT py, FT qx, FT qy, FT rx, FT ry)
{
    return sign( (qx-px)*(ry-py)-(rx-px)*(qy-py) );
}
```

FT peut être double

mais aussi Interval

# Filtre dynamique

Contre

Pour

# Filtre dynamique

Contre

Opérations complètes  
si on sait faire intervalle

Pas d'hypothèses sur les données

Pour

# Filtre dynamique

**Contre**

Opérations complètes

si on sait faire intervalle

Taux de succès excellent

Pas d'hypothèses sur les données

**Pour**

# Filtre dynamique

**Contre**

Moins rapide

Opérations complètes

si on sait faire intervalle

Taux de succès excellent

Pas d'hypothèses sur les données

**Pour**

# Filtre dynamique

## Contre

Moins rapide (erreur à l'exécution)

Opérations complètes

si on sait faire intervalle

Taux de succès excellent

Pas d'hypothèses sur les données

## Pour

Conclusion

paradigme du calcul exact

on isole les problèmes de cohérence

dans les prédicats

Conclusion

paradigme du calcul exact

on isole les problèmes de cohérence

dans les prédicats

# Conclusion

paradigme du calcul exact

on isole les problèmes de cohérence

dans les prédicats

on répond exactement au prédicat

# Conclusion

paradigme du calcul exact

on isole les problèmes de cohérence

dans les prédicats

on répond exactement au prédicat

1- c'est pour la cohérence, pas pour l'exactitude

2- ça veut pas dire forcément arithmétique exacte

# Cas dégénérés

# Cas dégénérés

et si les points sont alignés

cocycliques

• • •

Cas dégénérés

et si les points sont alignés

cocycliques

...

que faut-il faire



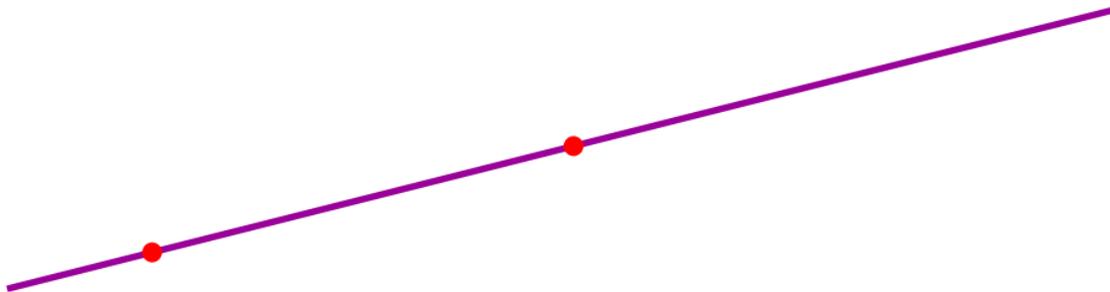
# Cas dégénérés

Mauvais argument :

# Cas dégénérés

Mauvais argument :

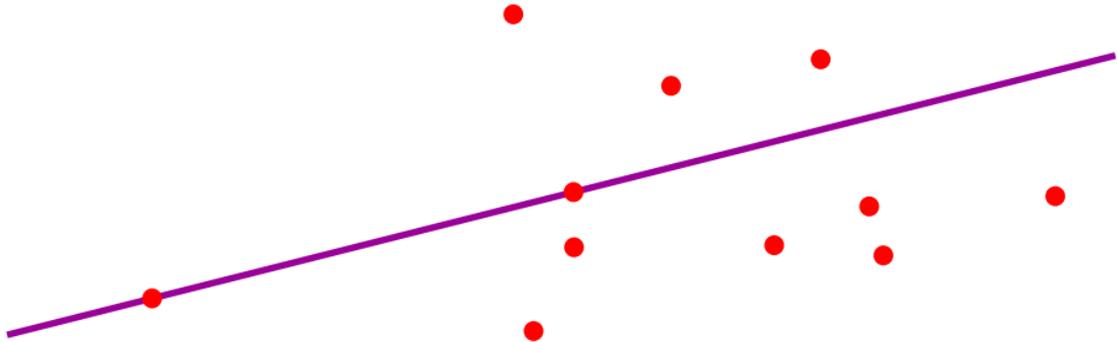
ça n'arrive pas



# Cas dégénérés

Mauvais argument :

ça n'arrive pas



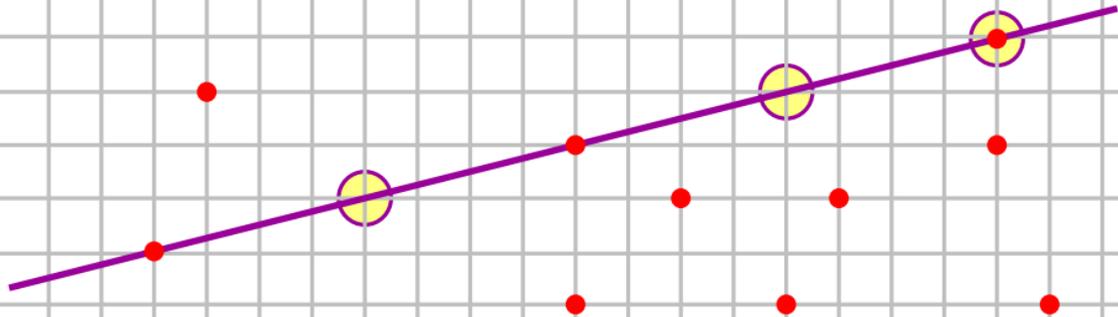
Proba nulle

# Cas dégénérés

Mauvais argument :

ça n'arrive pas

Mais si car les points sont discrets



Retour sur l'origine du problème

# Retour sur l'origine du problème

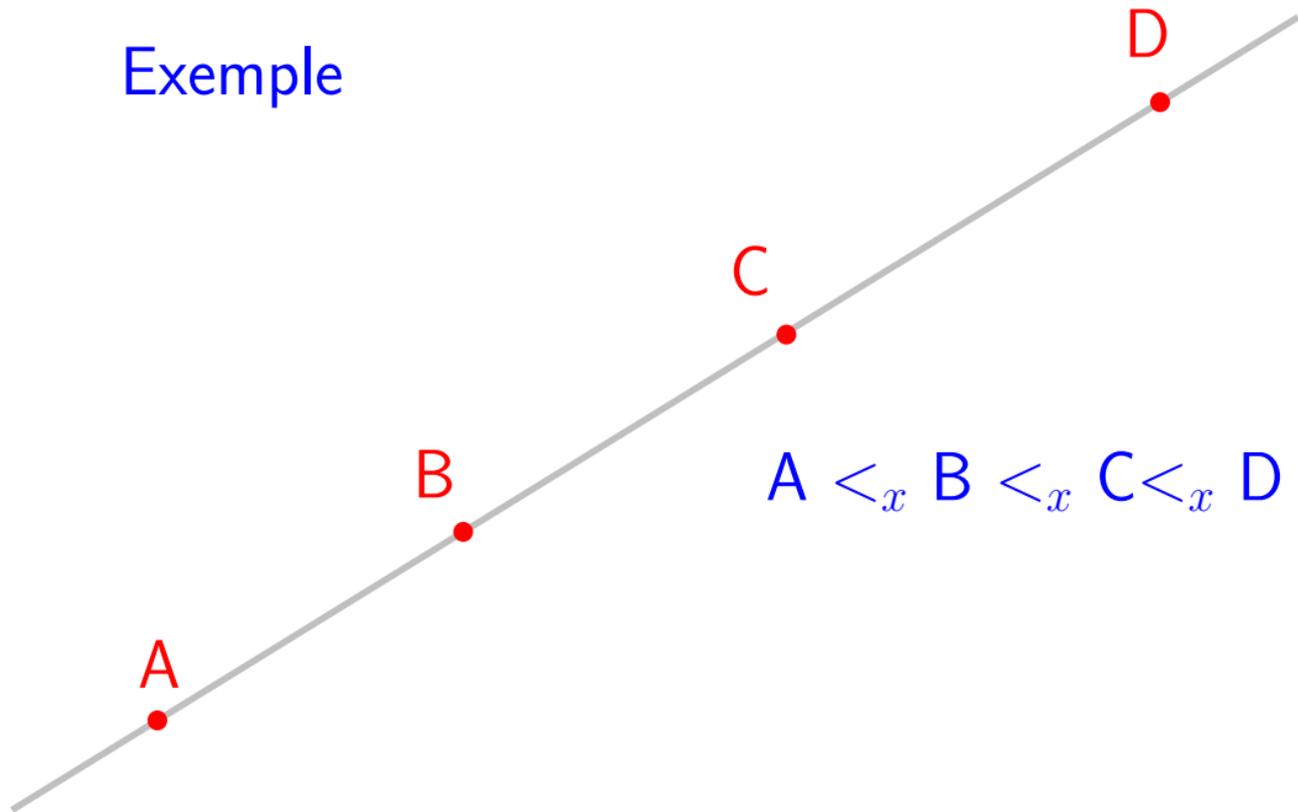
On a tendance à utiliser la géométrie

i.e. les théorèmes géométriques

dans les algorithmes

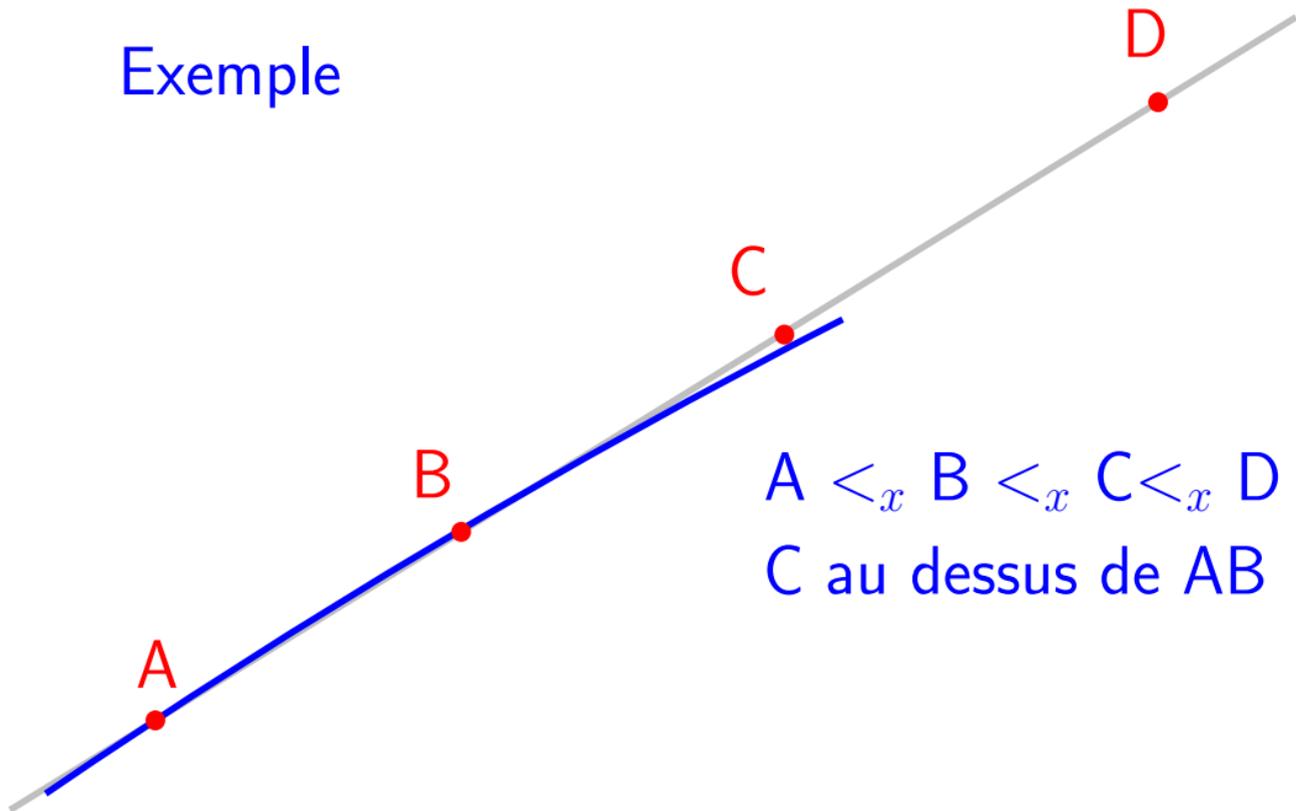
Robustesse

Exemple



Robustesse

Exemple

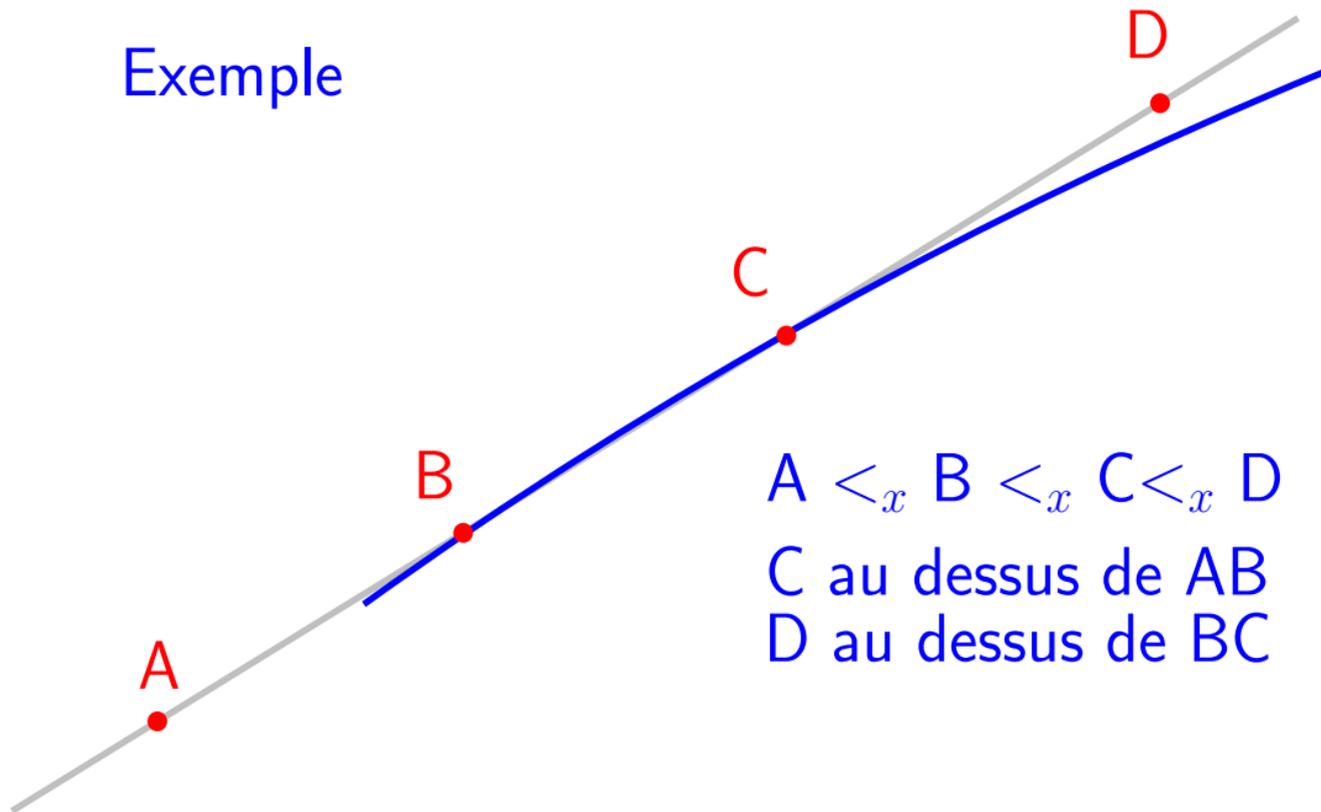


$A <_x B <_x C <_x D$

C au dessus de AB

Robustesse

Exemple



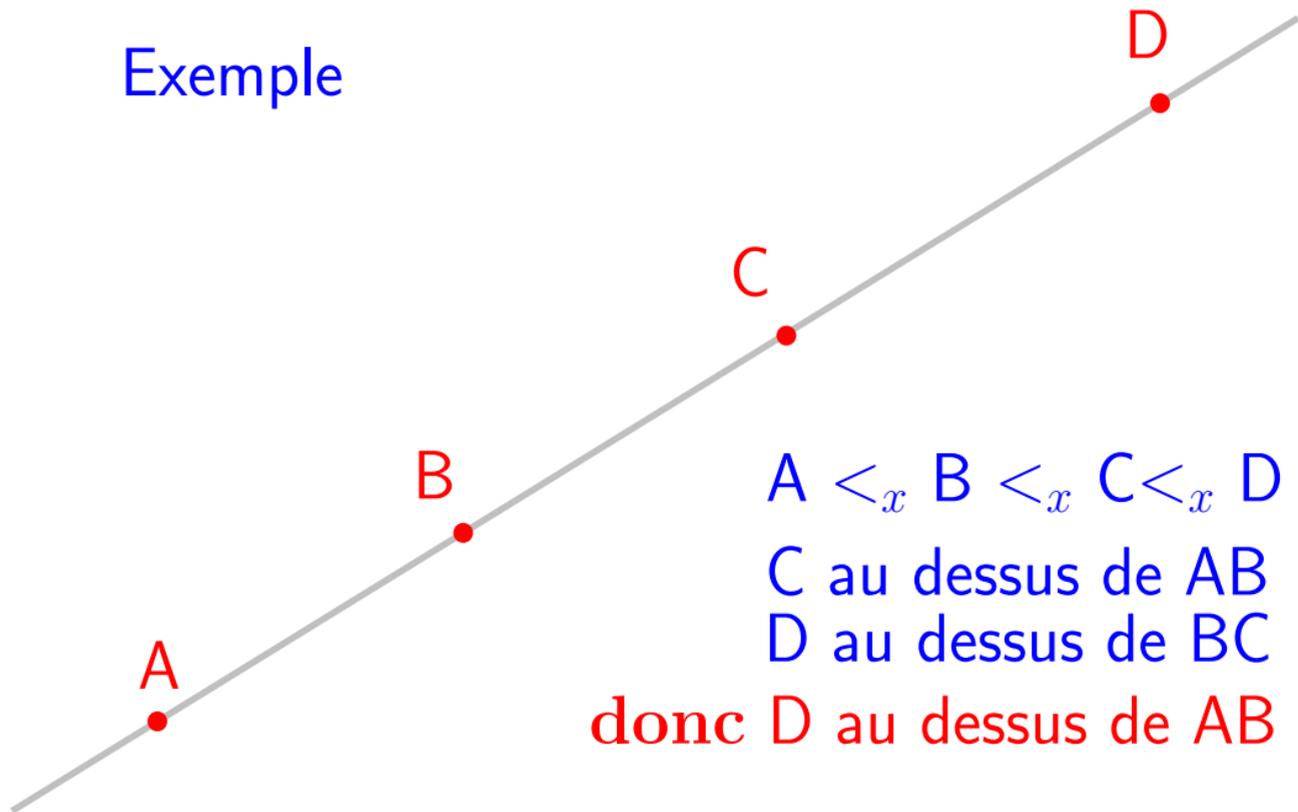
$$A <_x B <_x C <_x D$$

C au dessus de AB

D au dessus de BC

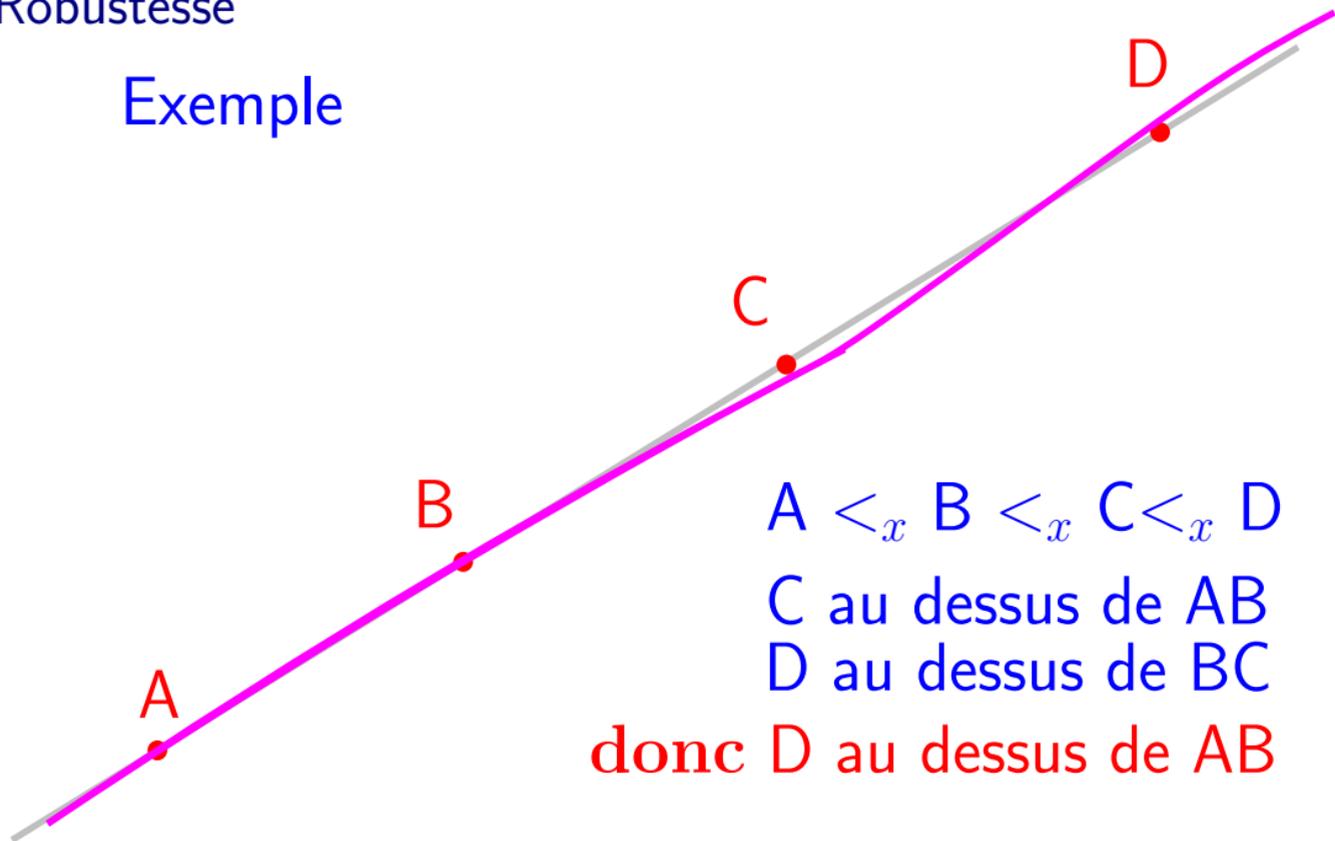
Robustesse

Exemple



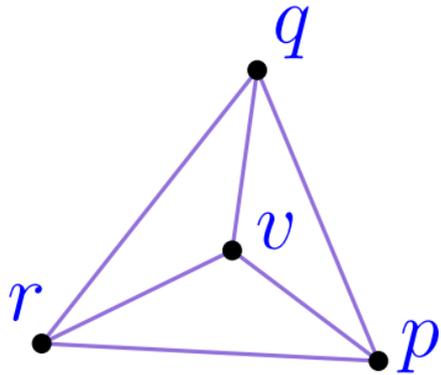
Robustesse

Exemple



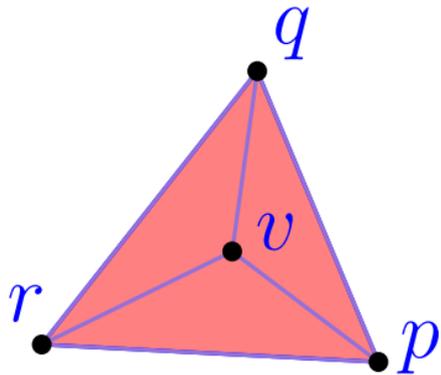
mais l'évaluation des prédicats pourrait donner le contraire

# théorème géométrique utile aux algorithmes

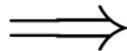


$$\begin{array}{l} pqv \\ qrv \\ rpv \end{array} \text{ CCW} \implies pqr$$

# théorème géométrique utile aux algorithmes

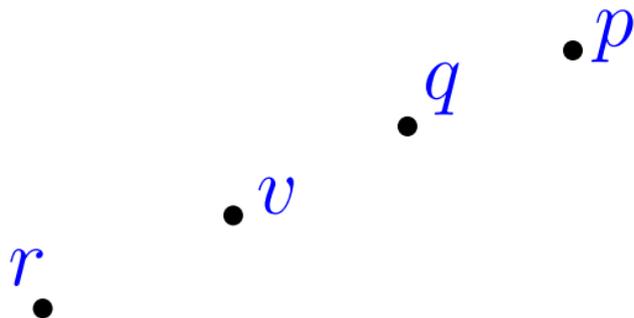


$pqv$   
 $qrv$  **CCW**  
 $rpv$

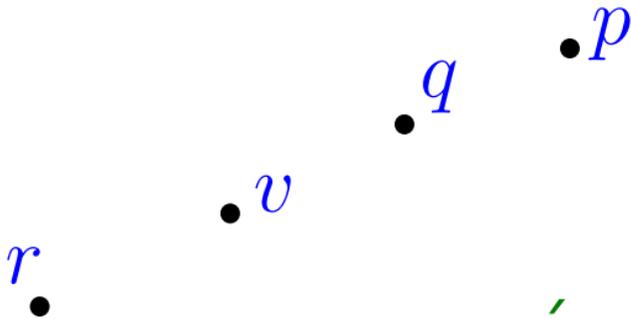


$pqr$  **CCW**

si les points sont alignés

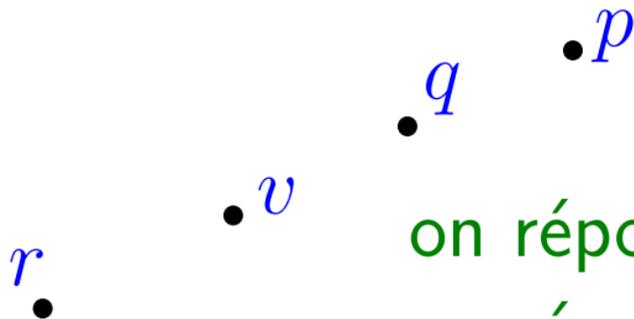


si les points sont alignés



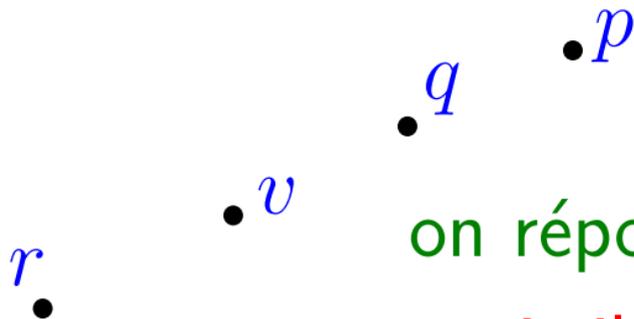
on répond alignés ?

si les points sont alignés



on répond n'importe quoi ?  
on répond alignés ?

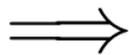
si les points sont alignés



on répond n'importe quoi ?

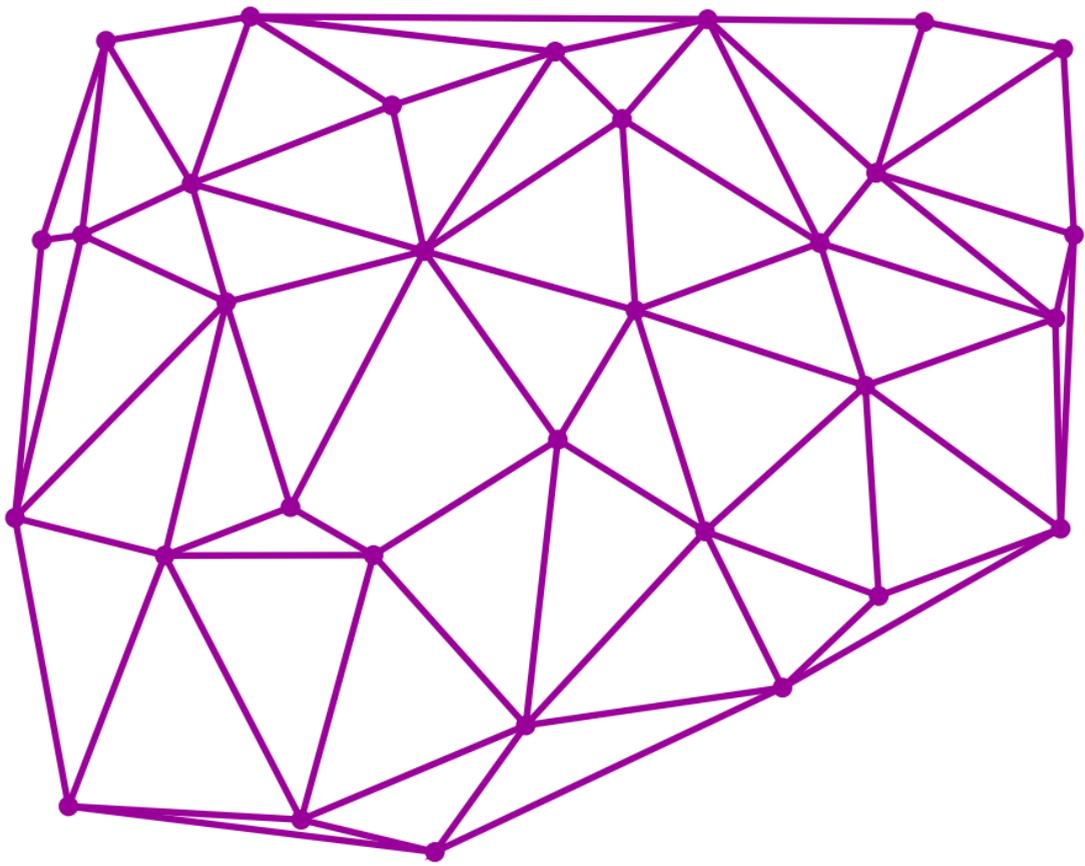
mais il faut être cohérent  
vis à vis des théorèmes géométriques

$pqv$   
 $qrv$  CCW  
 $rpv$

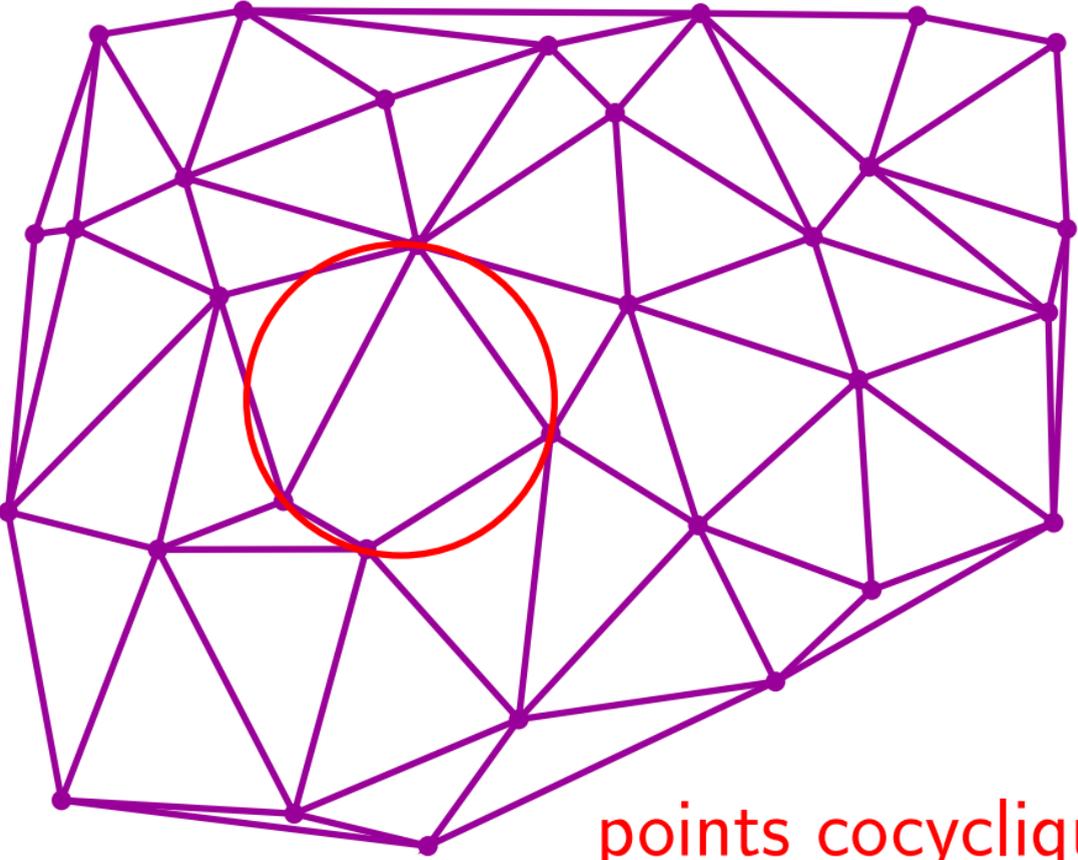


$pqr$  CCW

# triangulation de Delaunay

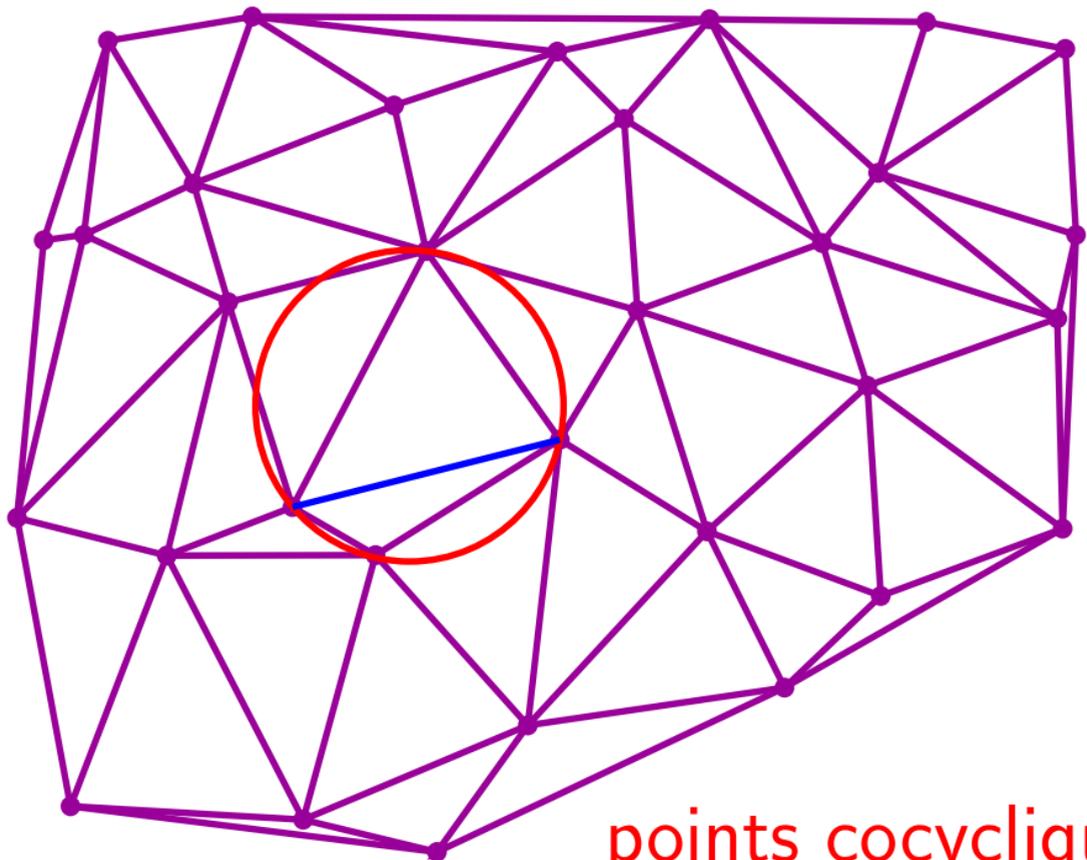


# triangulation de Delaunay



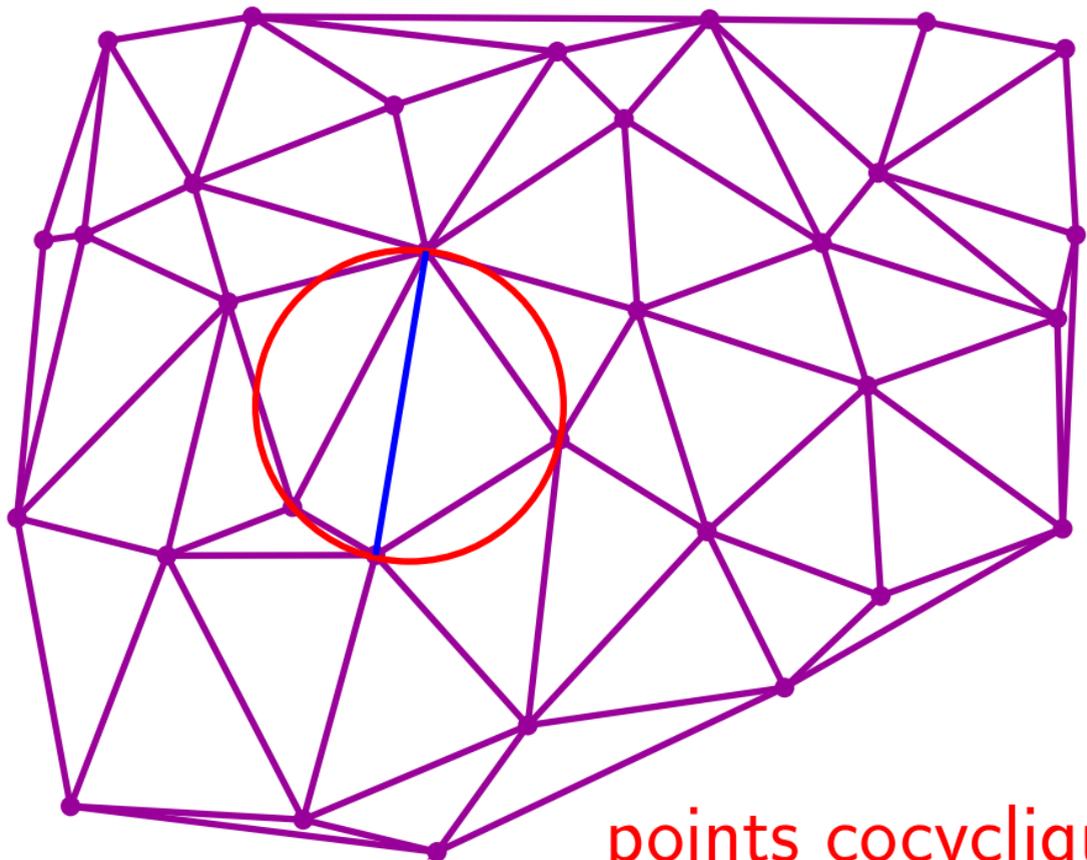
points cocycliques

# triangulation de Delaunay



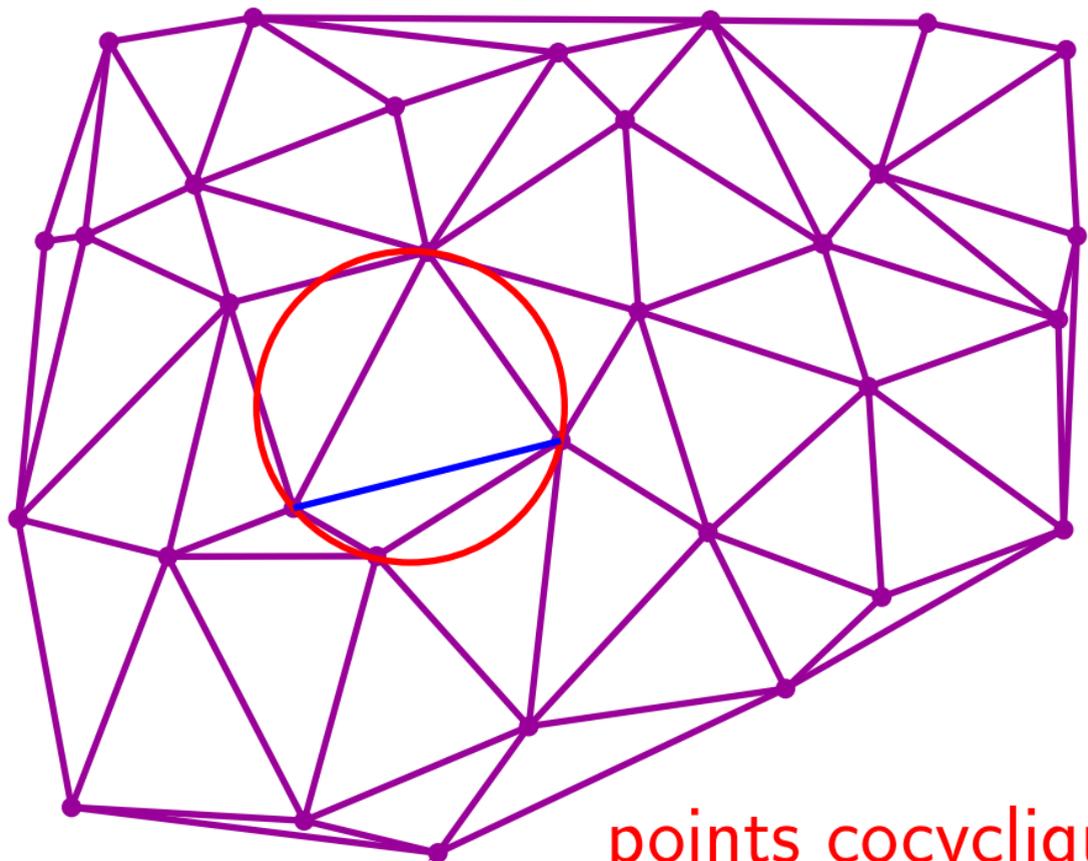
points cocycliques

# triangulation de Delaunay



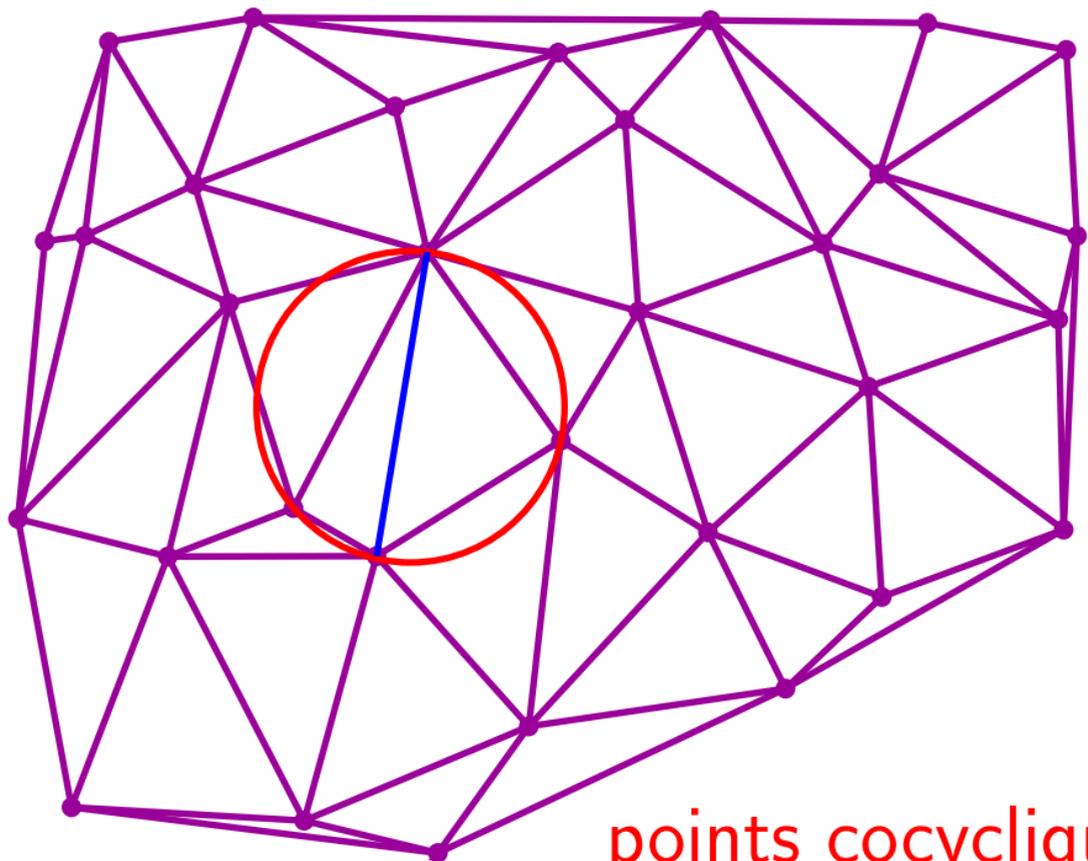
points cocycliques

# triangulation de Delaunay



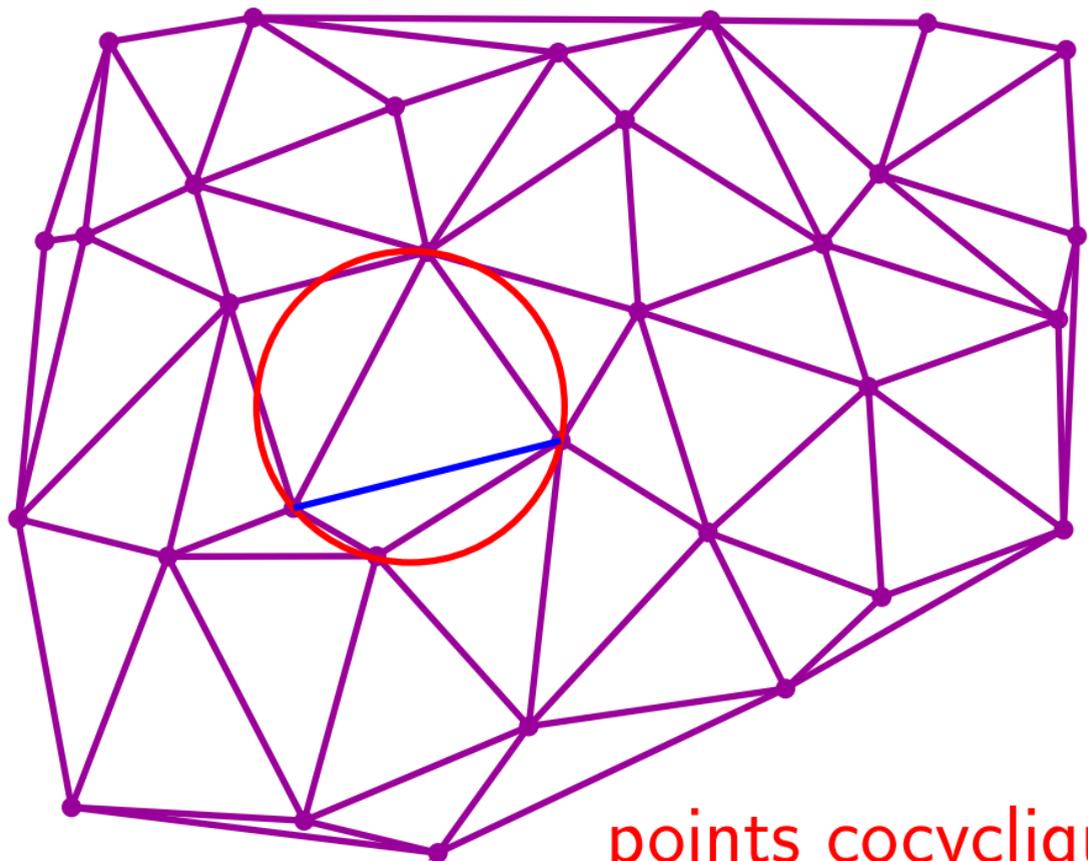
points cocycliques

# triangulation de Delaunay



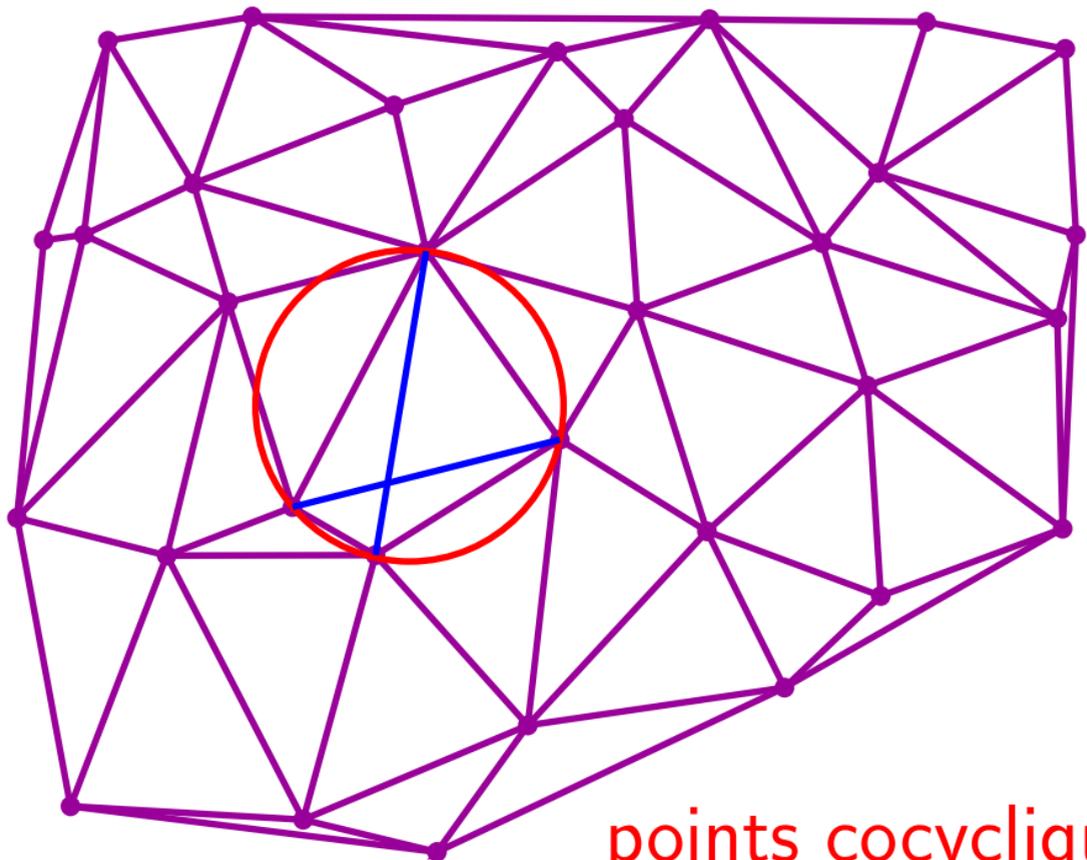
points cocycliques

# triangulation de Delaunay



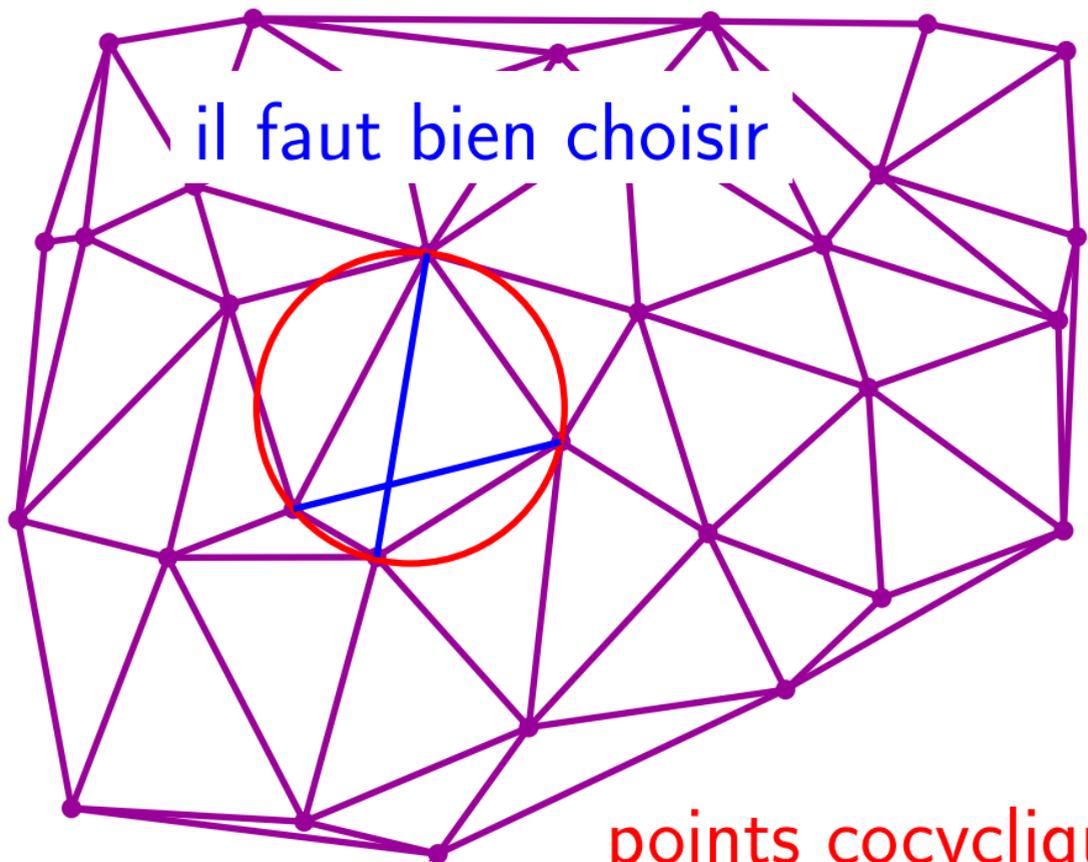
points cocycliques

# triangulation de Delaunay



points cocycliques

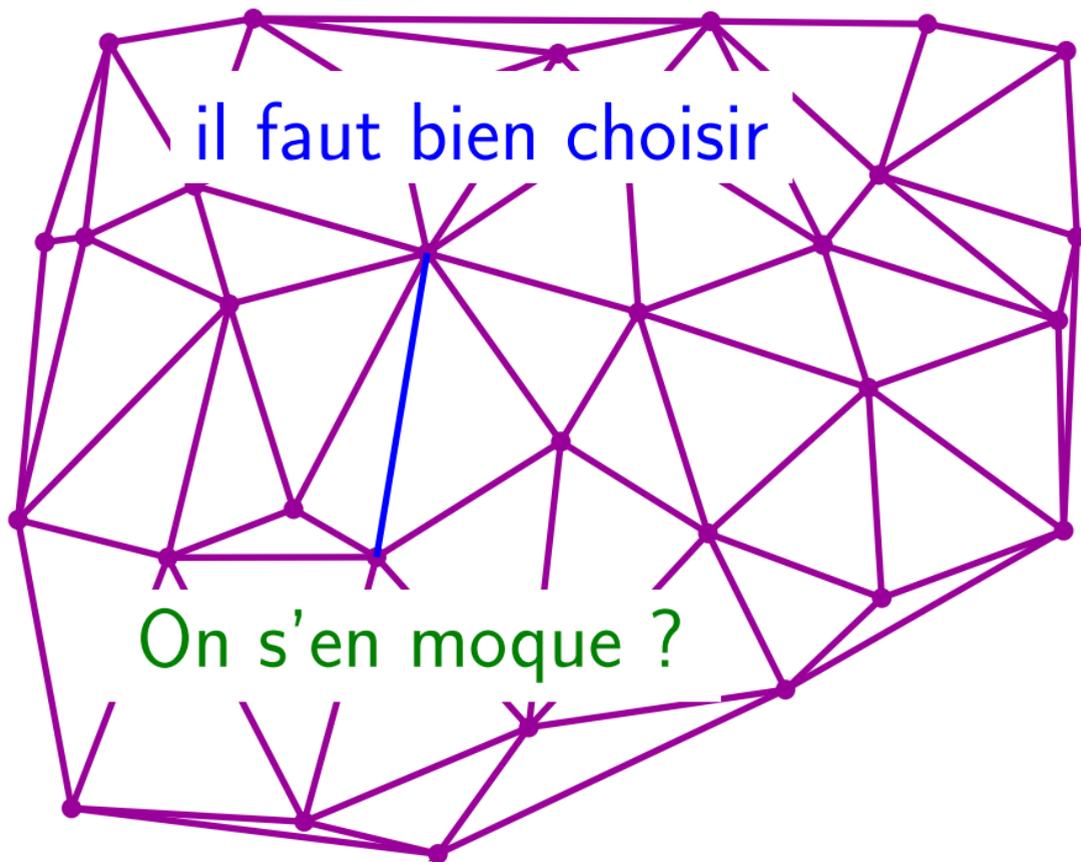
# triangulation de Delaunay



il faut bien choisir

points cocycliques

# triangulation de Delaunay

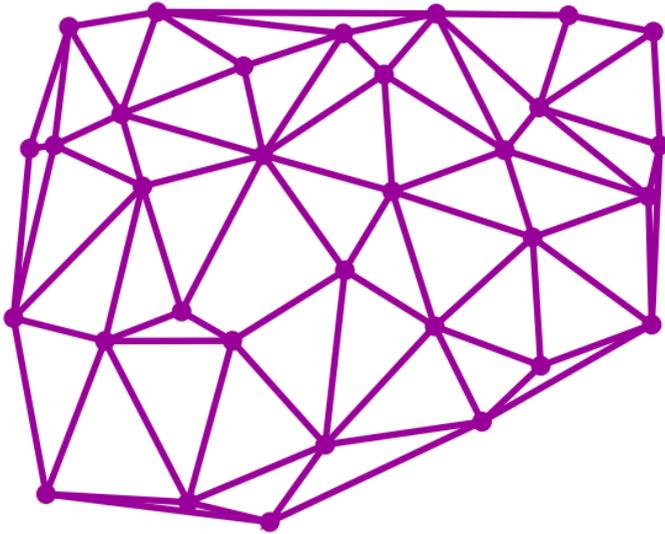


il faut bien choisir

On s'en moque ?

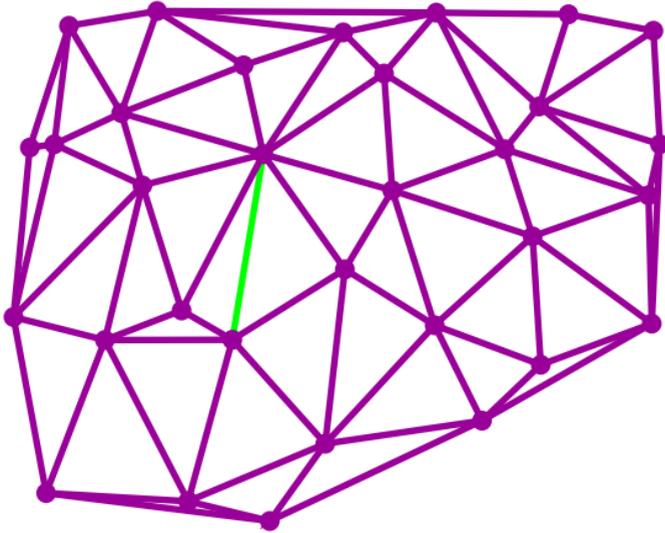
soit on traite les cas particuliers explicitement

soit on traite les cas particuliers explicitement



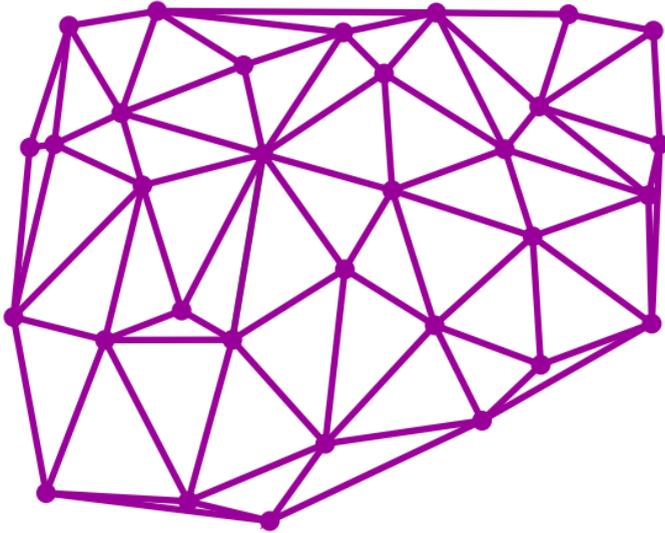
l'arête litigieuse  
est absente

soit on traite les cas particuliers explicitement



l'arête litigieuse  
est absente  
est "marquée"

soit on traite les cas particuliers explicitement



l'arête litigieuse  
est absente  
est "marquée"

soit on simule l'absence de cas particuliers

# SoS : Simulation of Simplicity

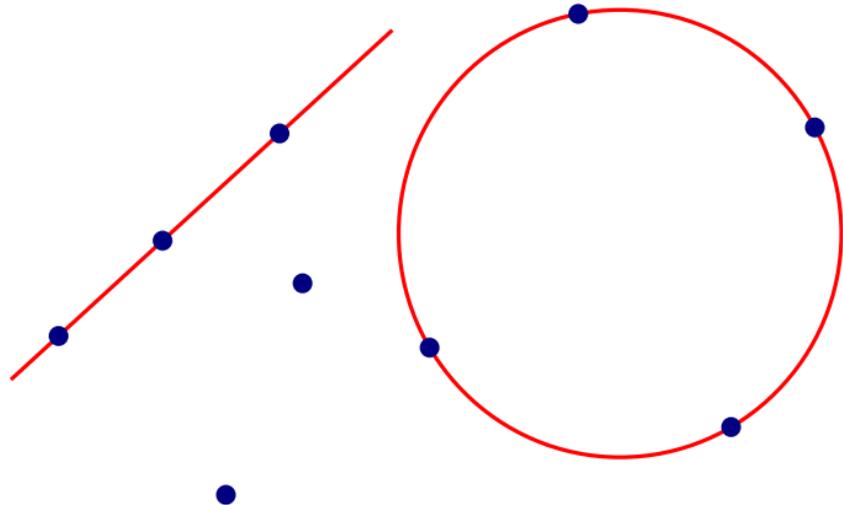
SoS : Simulation of Simplicity

Perturbation symboliques

# SoS : Simulation of Simplicity

## Perturbation symboliques

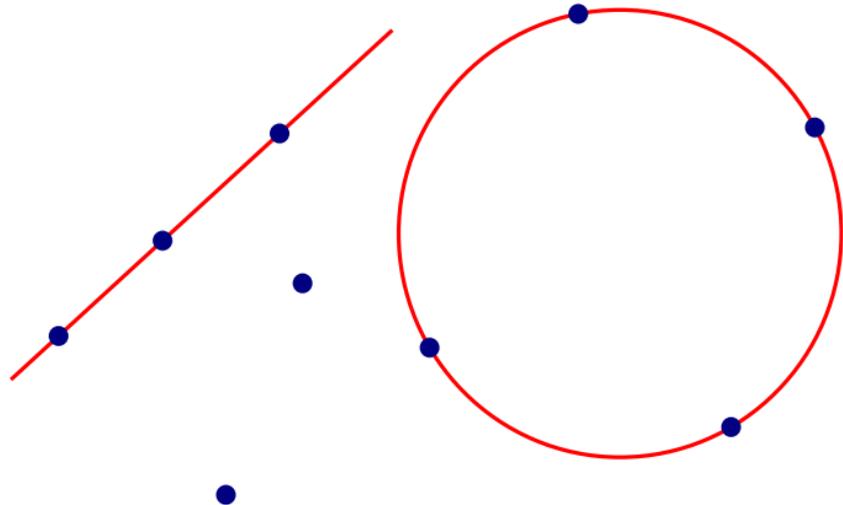
Les données



# SoS : Simulation of Simplicity

Perturbation symboliques

Les données **eventuellement dégénérées**

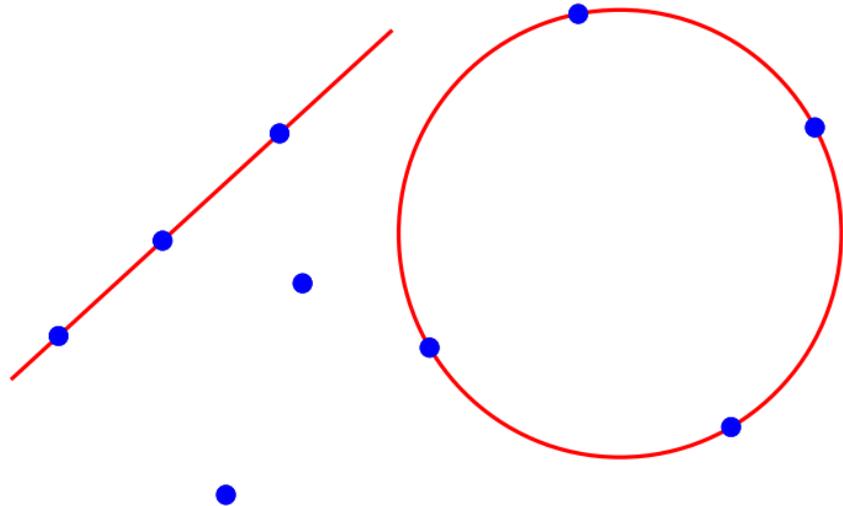


# SoS : Simulation of Simplicity

## Perturbation symboliques

Les données **eventuellement dégénérées**  
vont dépendre d'un paramètre  $\varepsilon$

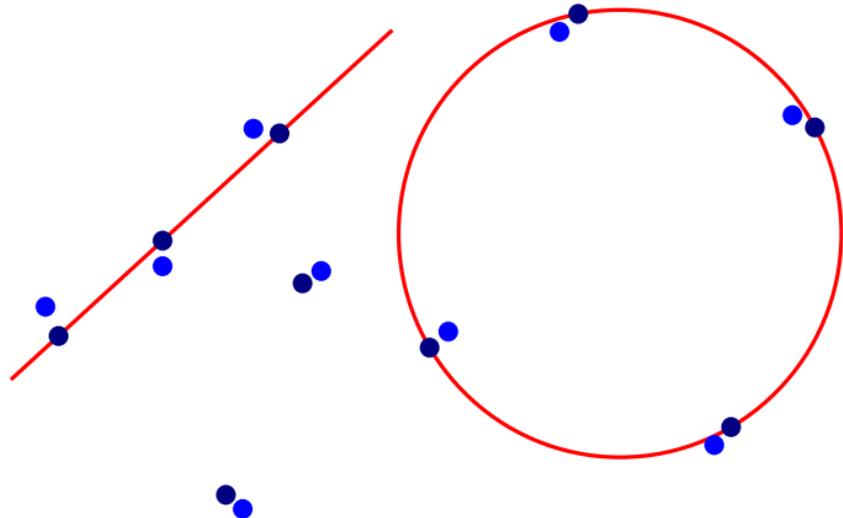
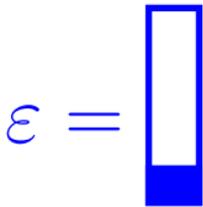
$$\varepsilon = \boxed{\phantom{000}}$$



# SoS : Simulation of Simplicity

## Perturbation symboliques

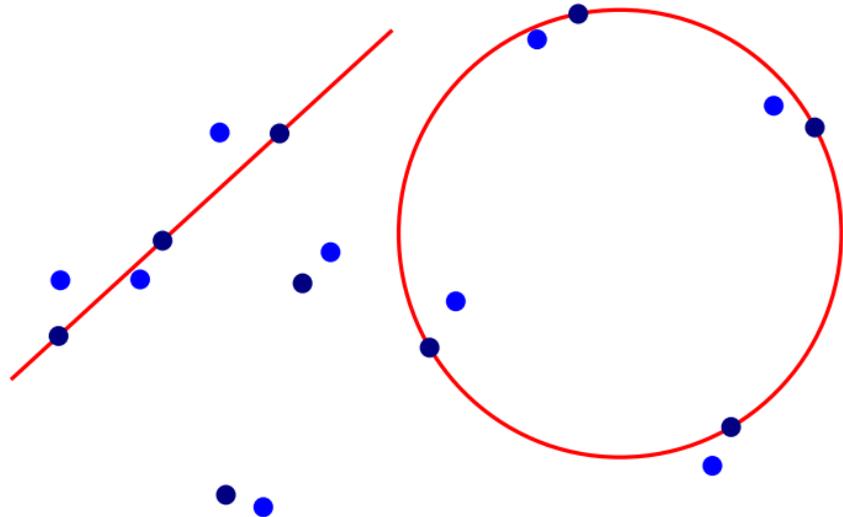
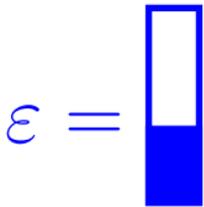
Les données **eventuellement dégénérées**  
vont dépendre d'un paramètre  $\varepsilon$



# SoS : Simulation of Simplicity

## Perturbation symboliques

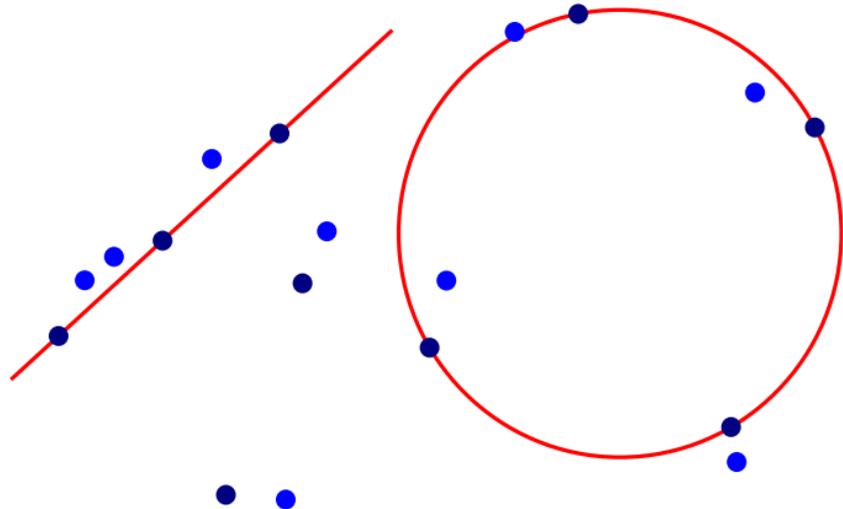
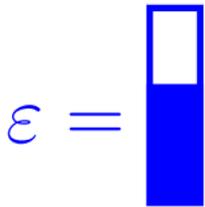
Les données **eventuellement dégénérées**  
vont dépendre d'un paramètre  $\varepsilon$



# SoS : Simulation of Simplicity

## Perturbation symboliques

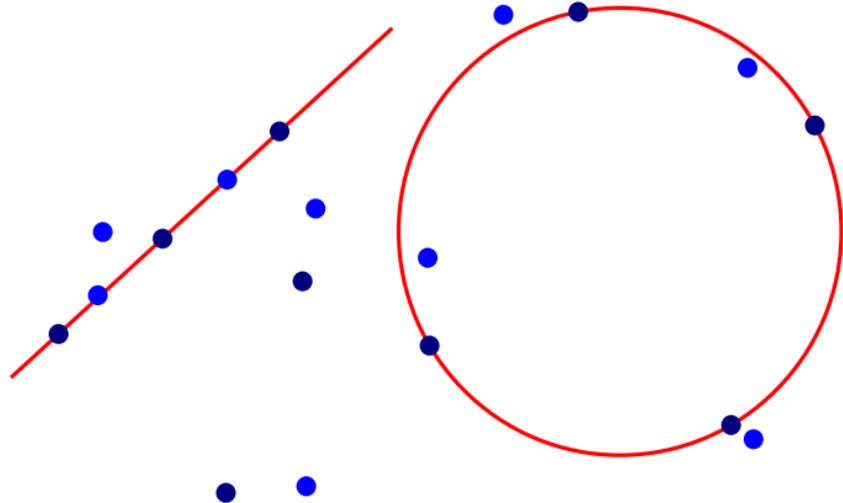
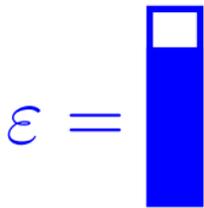
Les données **eventuellement dégénérées**  
vont dépendre d'un paramètre  $\varepsilon$



# SoS : Simulation of Simplicity

## Perturbation symboliques

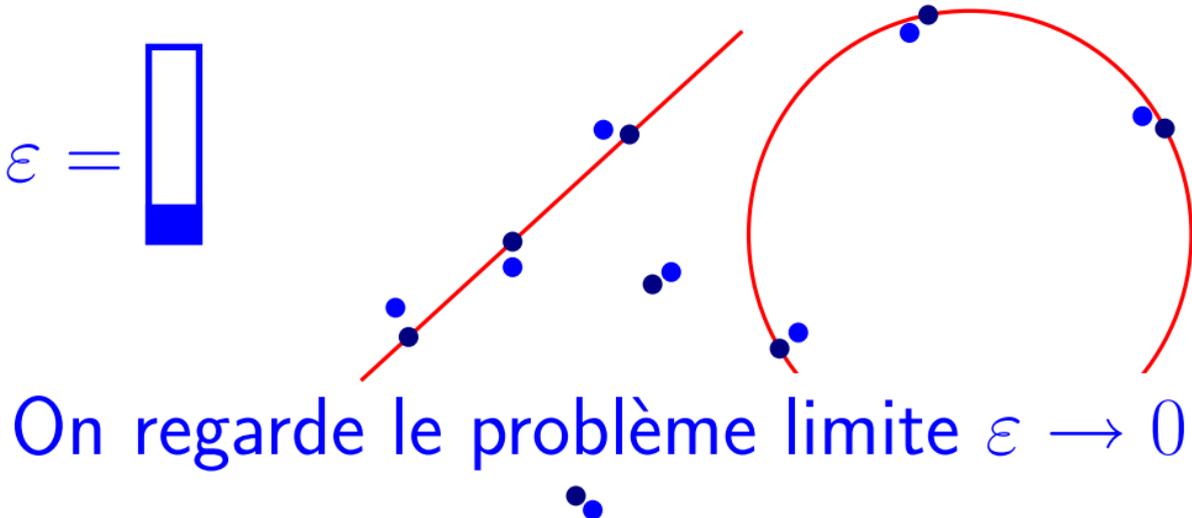
Les données **eventuellement dégénérées**  
vont dépendre d'un paramètre  $\varepsilon$



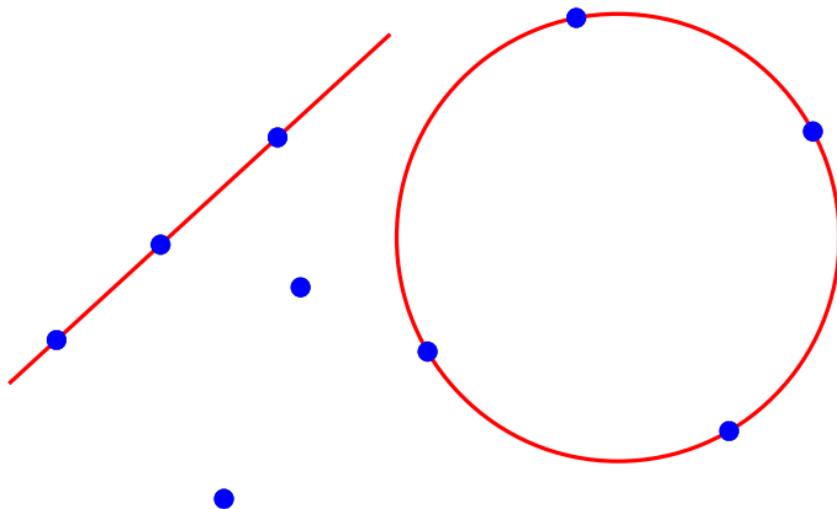
# SoS : Simulation of Simplicity

## Perturbation symboliques

Les données **eventuellement dégénérées**  
vont dépendre d'un paramètre  $\varepsilon$

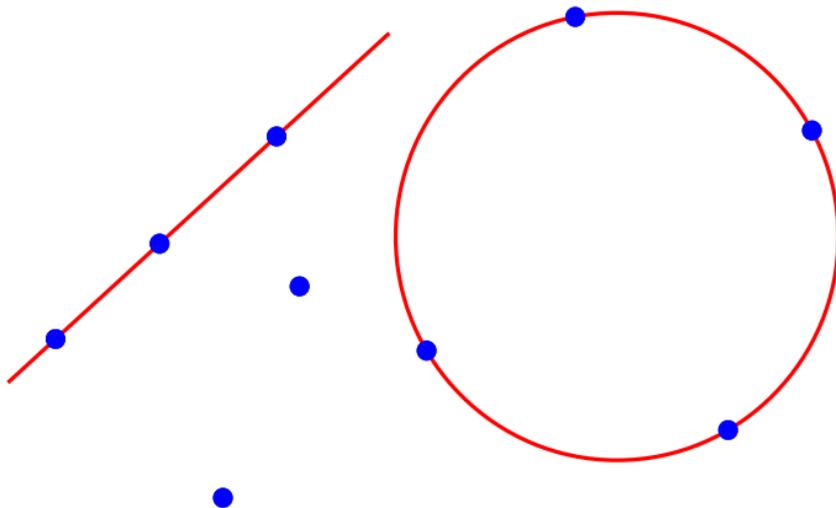


comme pour les problèmes de précision



comme pour les problèmes de précision

Tout se passe dans les prédicats



Les données vont dépendre d'un paramètre  $\varepsilon$

Les données vont dépendre d'un paramètre  $\varepsilon$

comment



Les données vont dépendre d'un paramètre  $\varepsilon$

pour  $\varepsilon = 0$  c'est (potentiellement) dégénéré

On étudie le problème limite

Les données vont dépendre d'un paramètre  $\varepsilon$

pour  $\varepsilon = 0$  c'est (potentiellement) dégénéré

On étudie le problème limite

il faut que, quand  $\varepsilon \rightarrow 0^+$

Les données vont dépendre d'un paramètre  $\varepsilon$

pour  $\varepsilon = 0$  c'est (potentiellement) dégénéré

On étudie le problème limite

il faut que, quand  $\varepsilon \rightarrow 0^+$

**1** les dégénérescences soient levées

Les données vont dépendre d'un paramètre  $\varepsilon$

pour  $\varepsilon = 0$  c'est (potentiellement) dégénéré

On étudie le problème limite

il faut que, quand  $\varepsilon \rightarrow 0^+$

**1** les dégénérescences soient levées  
d'une manière contrôlée ?

Les données vont dépendre d'un paramètre  $\varepsilon$

pour  $\varepsilon = 0$  c'est (potentiellement) dégénéré

On étudie le problème limite

il faut que, quand  $\varepsilon \rightarrow 0^+$

- 1 les dégénérescences soient levées  
d'une manière contrôlée ?  
canonique ?

Les données vont dépendre d'un paramètre  $\varepsilon$

pour  $\varepsilon = 0$  c'est (potentiellement) dégénéré

On étudie le problème limite

il faut que, quand  $\varepsilon \rightarrow 0^+$

- 1 les dégénérescences soient levées  
d'une manière contrôlée ?  
canonique ?  
que la limite ait du sens

Les données vont dépendre d'un paramètre  $\varepsilon$

pour  $\varepsilon = 0$  c'est (potentiellement) dégénéré

On étudie le problème limite

il faut que, quand  $\varepsilon \rightarrow 0^+$

**1** les dégénérescences soient levées  
d'une manière contrôlée ?  
canonique ?

que la limite ait du sens

**2** que ça coûte pas plus cher  
en temps, en précision...

Les données vont dépendre d'un paramètre  $\varepsilon$

comment



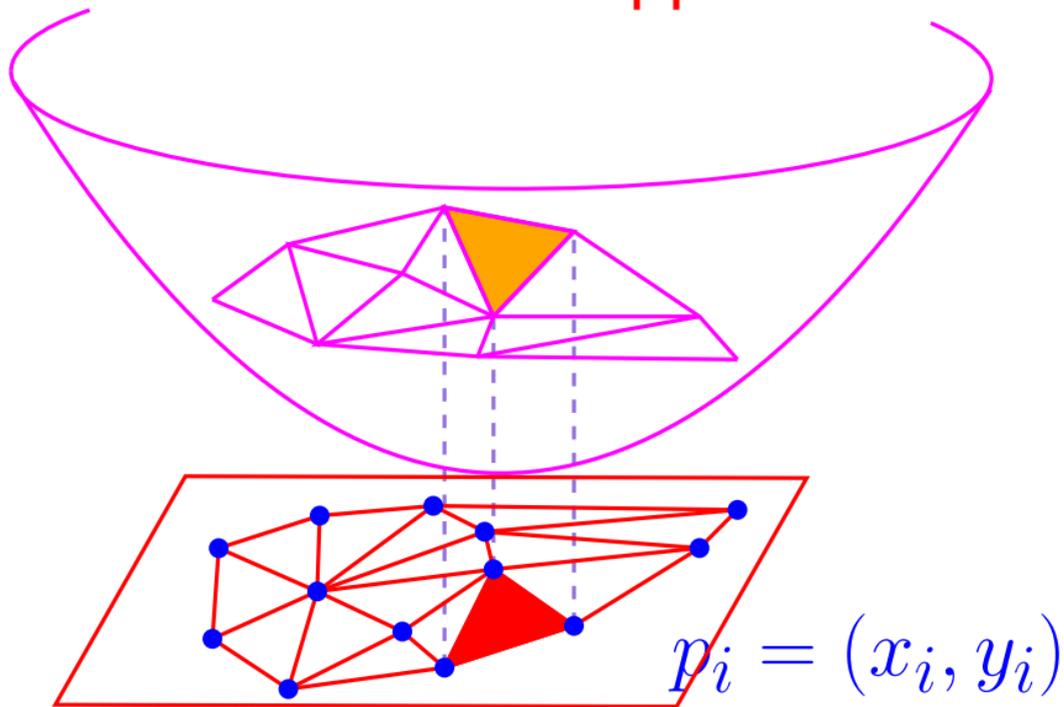
Perturbation pour Delaunay

Perturbation de l'enveloppe convexe 3D



# Perturbation pour Delaunay

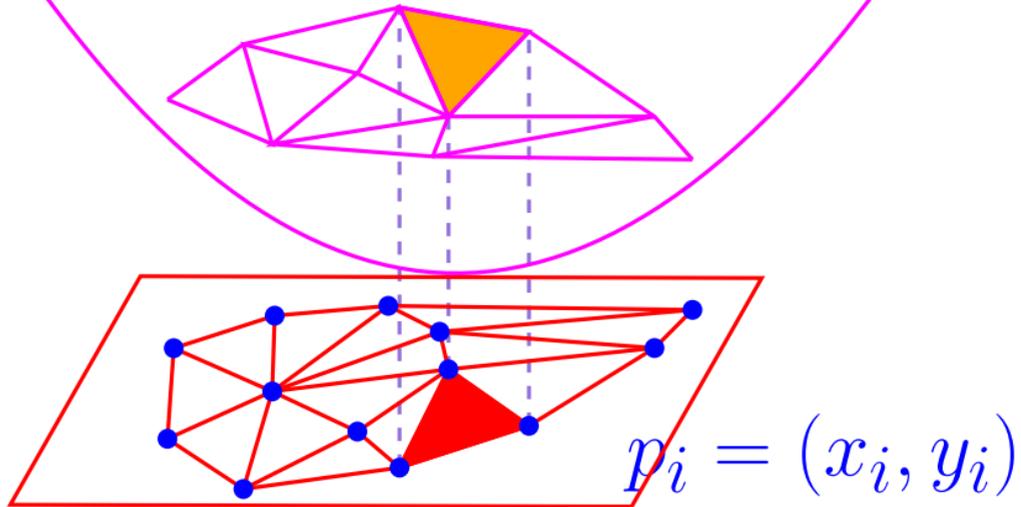
## Perturbation de l'enveloppe convexe 3D



# Perturbation pour Delaunay

## Perturbation de l'enveloppe convexe 3D

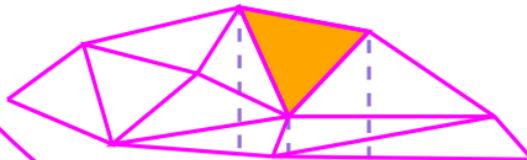
$$p_i^* = (x_i, y_i, x_i^2 + y_i^2 + \varepsilon^{n-i})$$



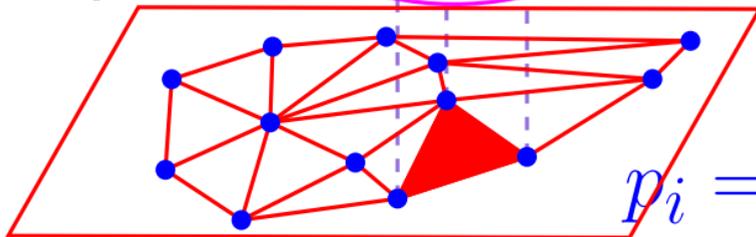
# Perturbation pour Delaunay

## Perturbation de l'enveloppe convexe 3D

$$p_i^* = (x_i, y_i, x_i^2 + y_i^2 + \varepsilon^{n-i})$$



Le dernier point est perturbé

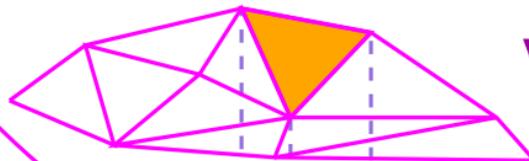


$$p_i = (x_i, y_i)$$

# Perturbation pour Delaunay

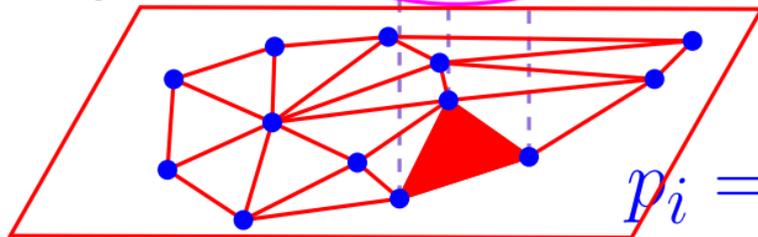
## Perturbation de l'enveloppe convexe 3D

$$p_i^* = (x_i, y_i, x_i^2 + y_i^2 + \varepsilon^{n-i})$$



vers le haut

Le dernier point est perturbé

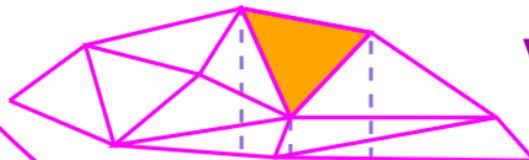


$$p_i = (x_i, y_i)$$

# Perturbation pour Delaunay

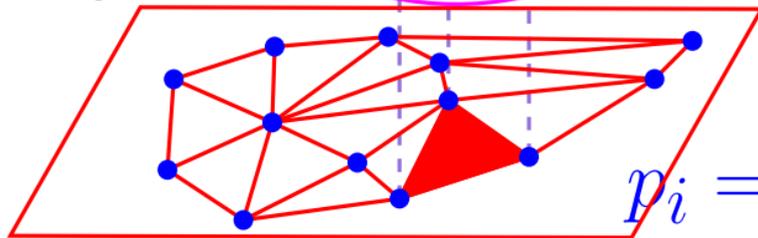
## Perturbation de l'enveloppe convexe 3D

$$p_i^* = (x_i, y_i, x_i^2 + y_i^2 + \varepsilon^{n-i})$$



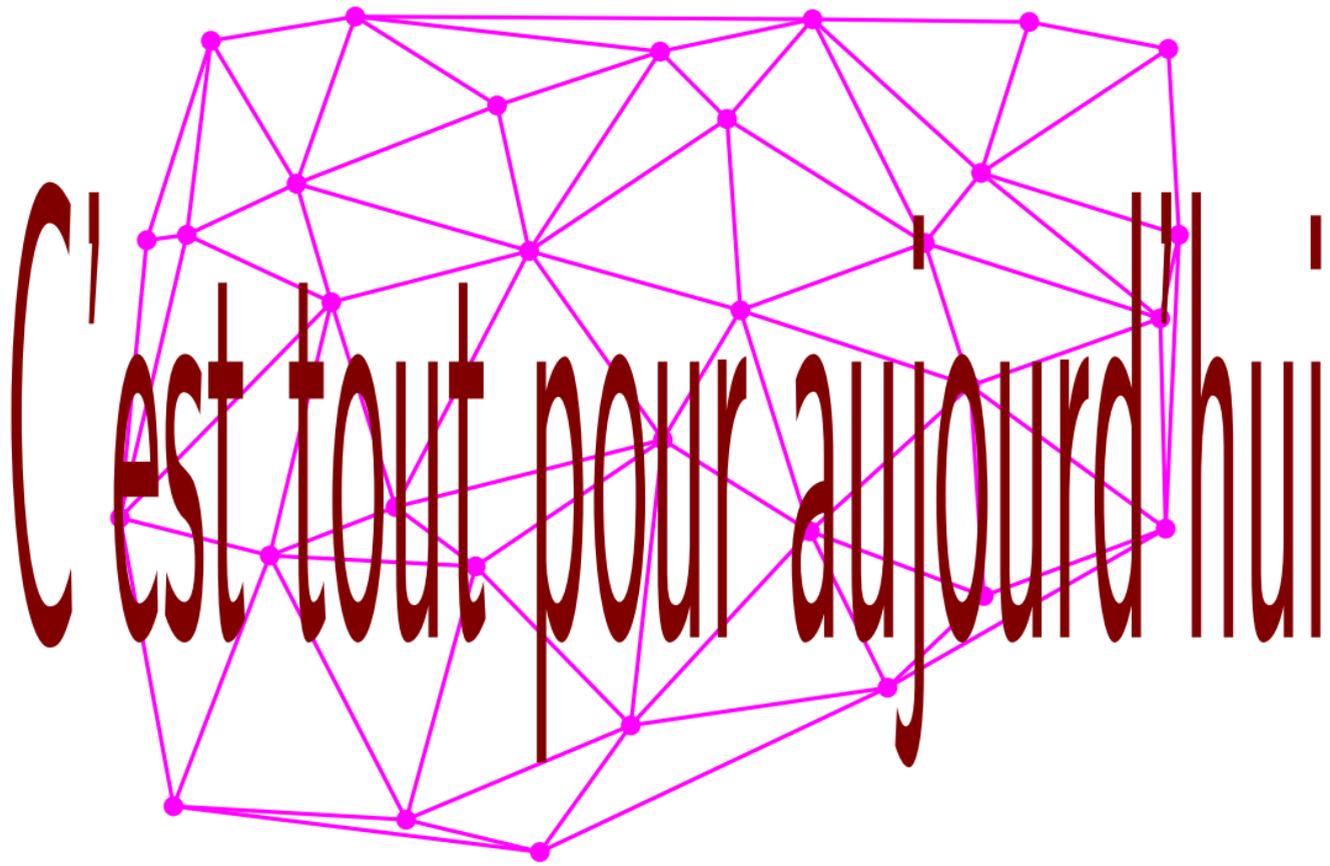
vers le haut

Le dernier point est perturbé



$$p_i = (x_i, y_i)$$

vers l'extérieur du cercle



C'est tout pour aujourd'hui