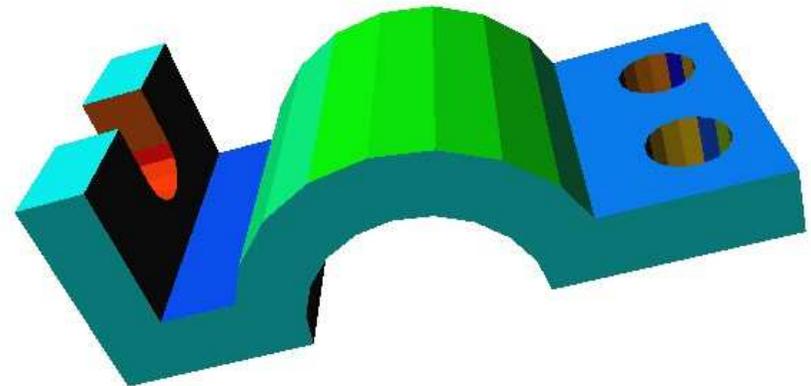


Représentation frontière

-2^{ème} partie – le cas 3D

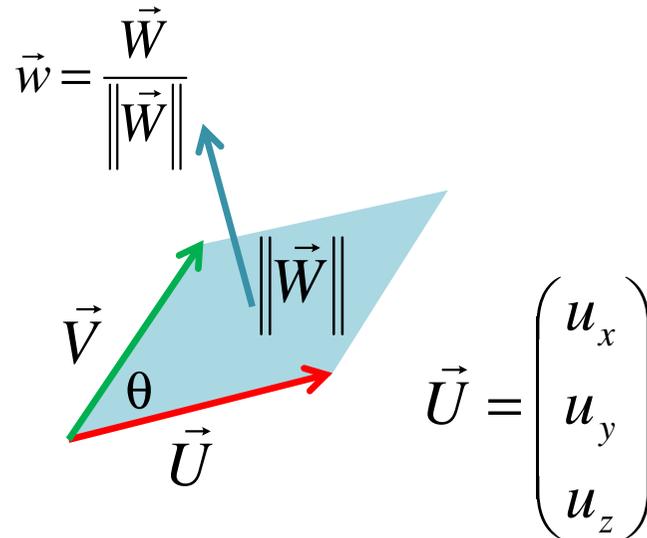
1. Rappels
2. Représentation informatique : Polyèdres et Triangulations
3. Interrogation du modèle
4. Modifications du modèle :
 - Transformations géométriques,
 - Opérations booléennes...
5. Parties cachées (TP)



Produit Vectoriel (3D)

Rappels de géométrie (1)

$$\vec{W} = \vec{U} \wedge \vec{V}$$



$$\|\vec{W}\| = \|\vec{U}\| \|\vec{V}\| \sin(\theta)$$

$$W = \begin{pmatrix} w_x = u_y \cdot v_z - u_z \cdot v_y \\ w_y = u_z \cdot v_x - u_x \cdot v_z \\ w_z = u_x \cdot v_y - u_y \cdot v_z \end{pmatrix}$$

$$M_u V = \begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} -u_z v_y + u_y v_z \\ u_z v_x - u_x v_z \\ -u_y v_x + u_x v_y \end{pmatrix}$$

Python(numpy) : `np.cross(u, v)`

Scilab : `W=cross (U,V)`

Produit scalaire (3D)

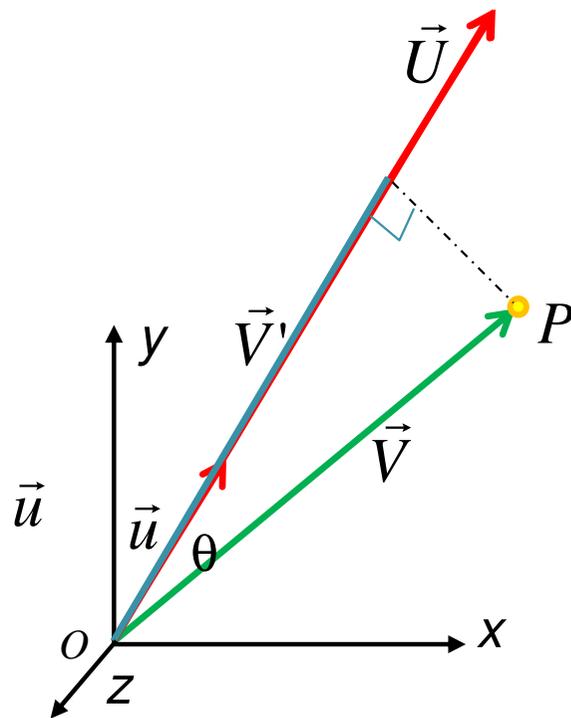
Rappels de géométrie (2)

$$w = \vec{U} \cdot \vec{V}$$

$$w = \|\vec{U}\| \|\vec{V}\| \cos(\theta)$$

$$\|\vec{V}\|^2 = \vec{V} \cdot \vec{V}$$

$$w = {}^t U * V$$



Projection d'un vecteur V sur l'axe \vec{u}

$$\vec{V}' = \|\vec{V}\| \cos(\theta) \vec{u}$$

Écriture matricielle de la projection :

$$w = u_x \cdot v_x + u_y \cdot v_y + u_z \cdot v_z$$

$$P_u * V = \begin{pmatrix} u_x^2 & u_x \cdot u_y & u_x \cdot u_z \\ u_y \cdot u_x & u_y^2 & u_y \cdot u_z \\ u_z \cdot u_x & u_z \cdot u_y & u_z^2 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} w \cdot u_x \\ w \cdot u_y \\ w \cdot u_z \end{pmatrix}$$

Python(numpy) : `np.dot(u, v)`
 Scilab : `w=U'*V`

P_u est la matrice de projection sur l'axe \vec{u}

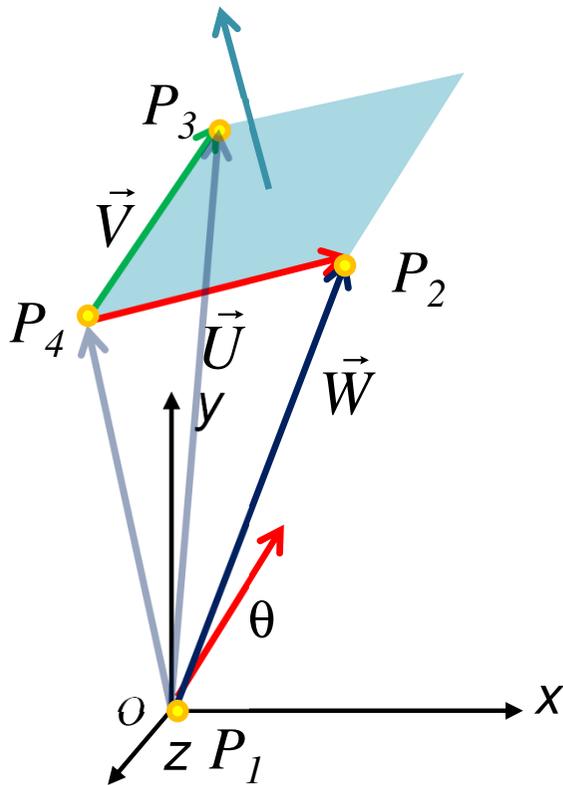
Produit mixte (3D)

Rappels de géométrie (3)

$$m = \vec{W} \cdot (\vec{U} \wedge \vec{V})$$

Écriture matricielle :

$$\vec{W} \cdot (\vec{U} \wedge \vec{V}) = \text{Det} \begin{pmatrix} w_x & u_x & v_x \\ w_y & u_y & v_y \\ w_z & u_z & v_z \end{pmatrix}$$



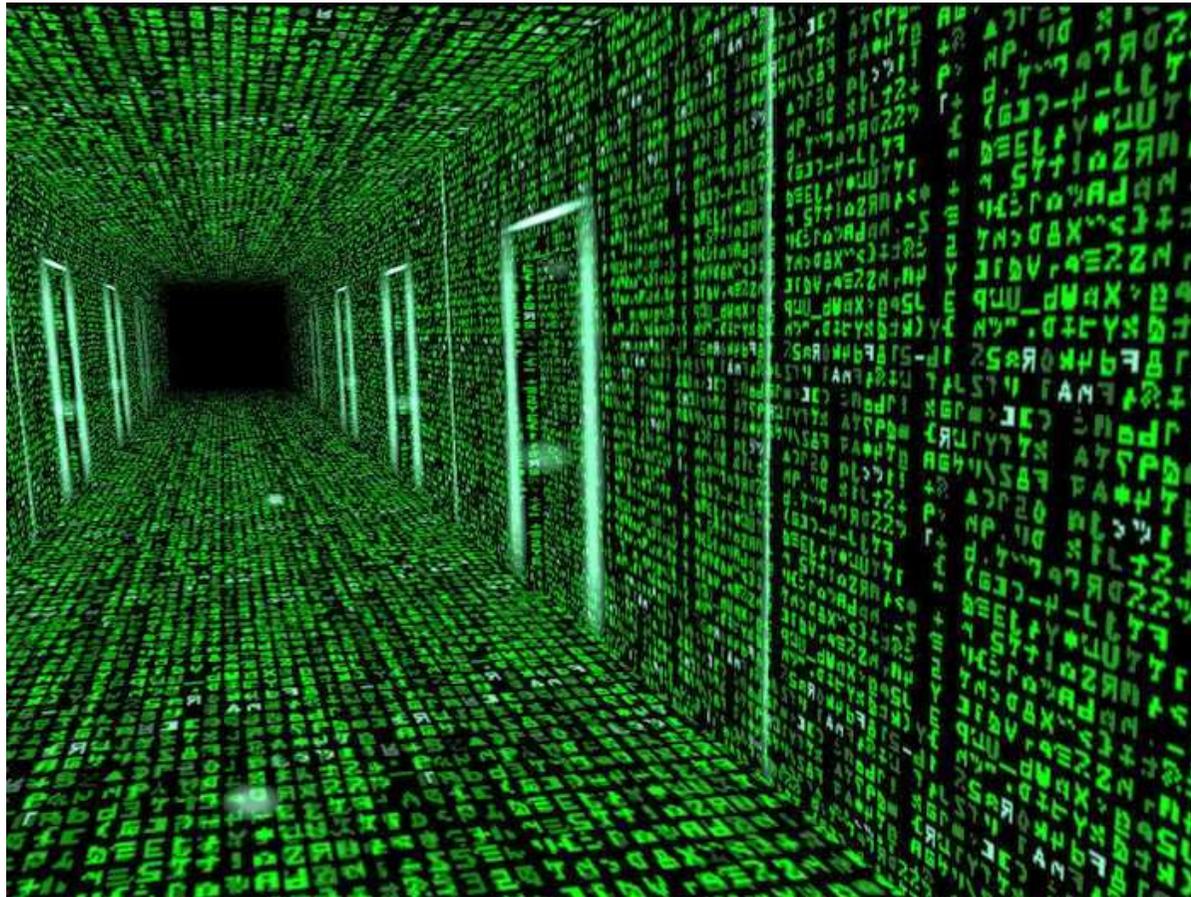
Volume du tétraèdre :

$$\text{vol}(P_1, P_2, P_3, P_4) = \frac{1}{6} (\vec{W} \cdot (\vec{U} \wedge \vec{V}))$$

Le plan

- Représentation algébrique (RA) ?
- Calcul de la RA à partir de 3 points P, Q, R?

2. Représentation informatique



Description des polyèdres

Face-based data structure (~~edge-based data structure~~)

```
#VRML V2.0 utf8
```

2 blocs : les points et les faces

```
...
```

```
Geometry IndexedFaceSet {
```

```
ccw TRUE
```

```
solid TRUE
```

← **Indications**

```
convex TRUE
```

```
coord Coordinate { point [
```

```
# Coordonnees des points
```

```
X1 Y1 Z1
```

```
X2 Y2 Z2
```

```
...
```

```
}]
```

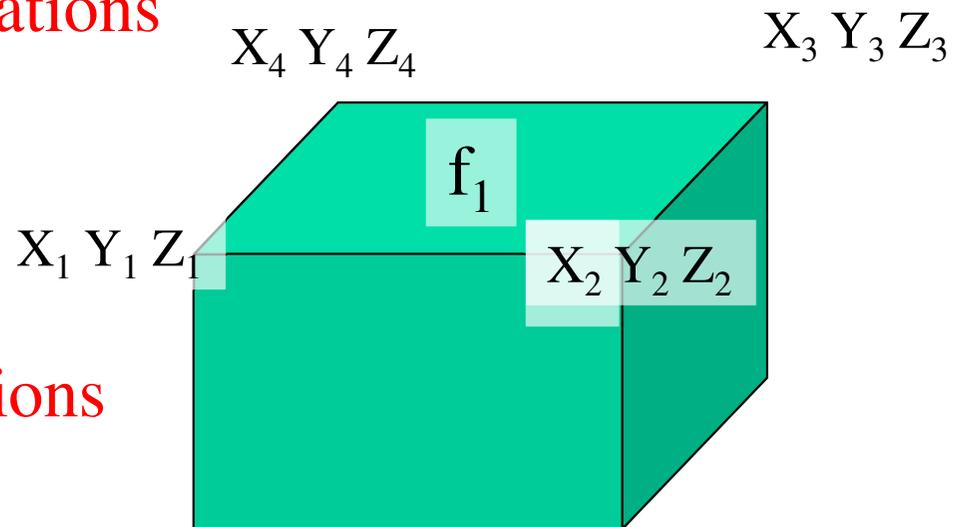
```
coordIndex [
```

```
# Les faces
```

```
1 2 3 4 -1
```

```
...
```

```
]
```



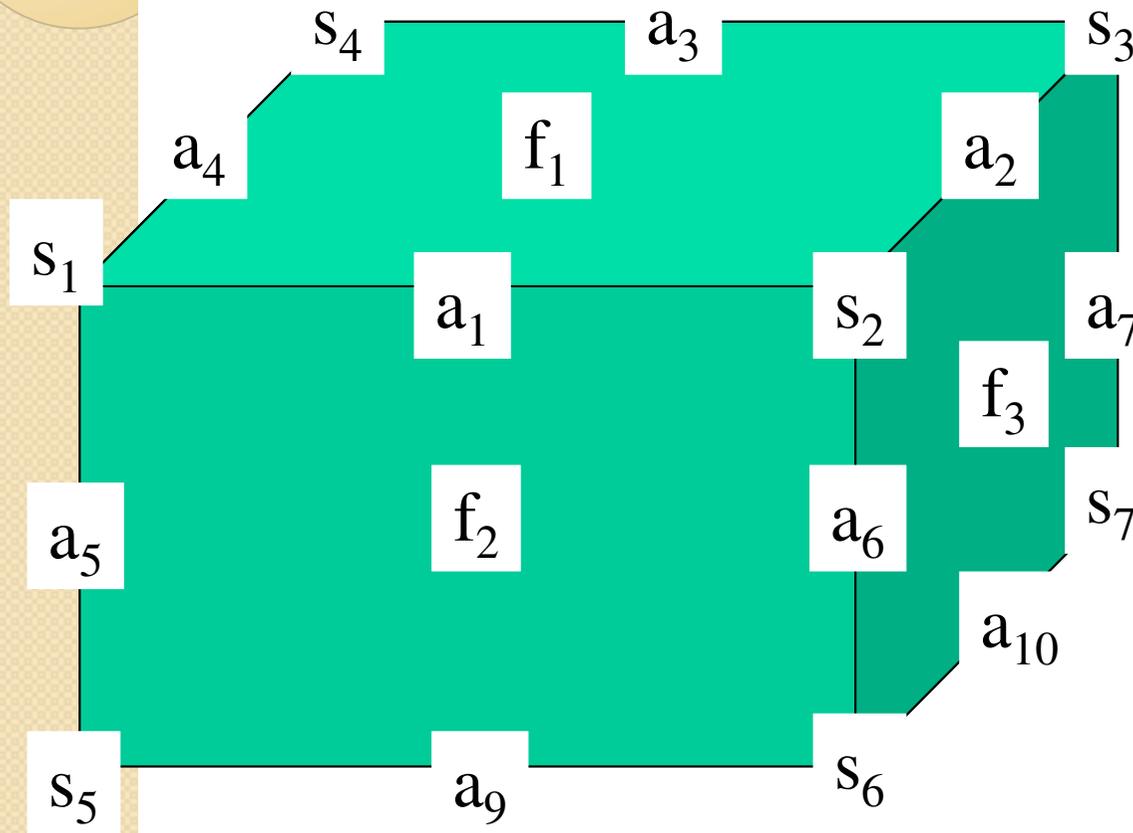
← **Limitations**

Exemple de format de données de type :
X3D, VRML, Inventor...

Représentation informatique

Edge-based data structure (~~face-based data structure~~)

interne, Cas 3D



$$F = \{ f_1, f_2 \dots f_6 \}$$

$$A = \{ a_1, a_2 \dots a_{12} \}$$

$$S = \{ s_1, s_2 \dots s_8 \}$$

Relations :

v F A S

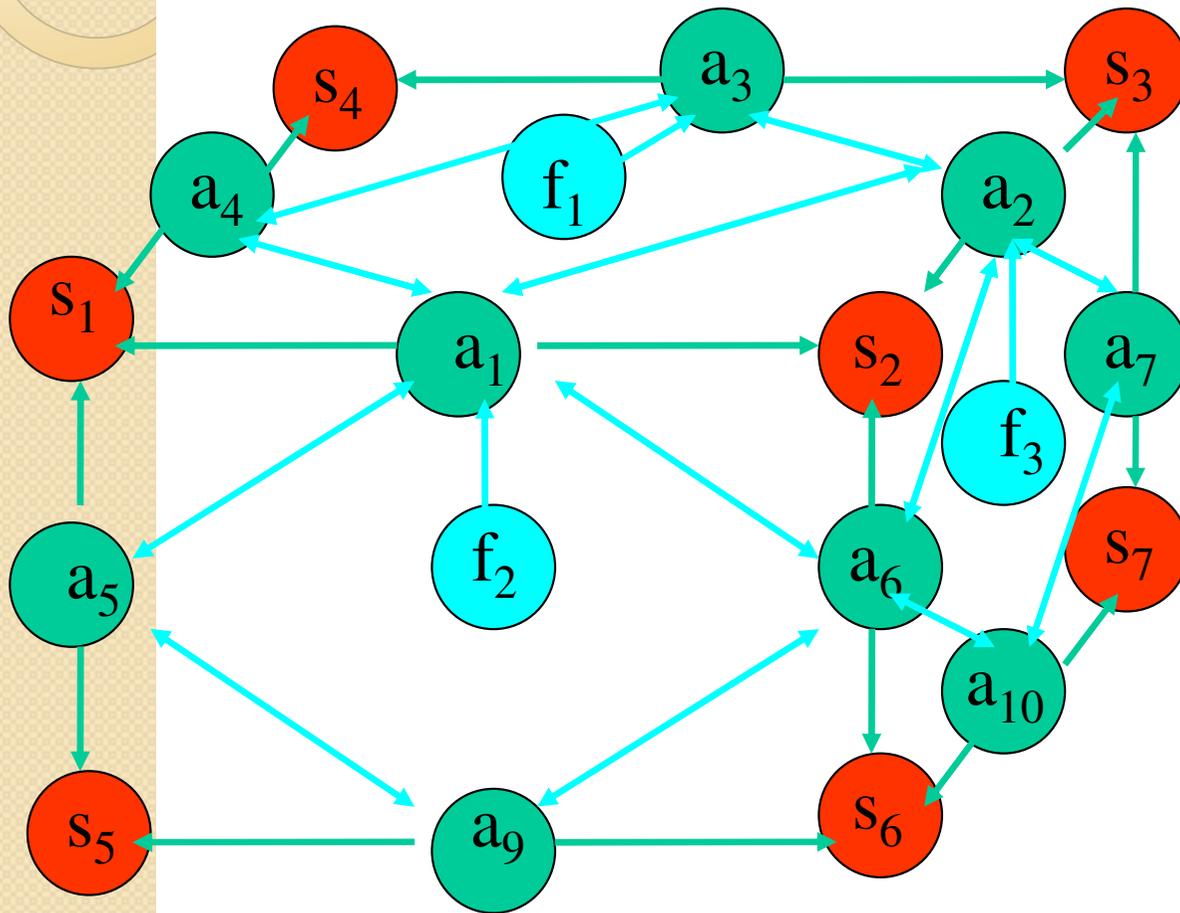
$$b(v) = \{ f_1, f_2 \dots f_6 \}$$

$$b(f_3) = \{ a_2, a_6, a_{10}, a_7 \}$$

$$b(a_6) = \{ s_6, s_2 \}$$

Représentation informatique

Edge-based data structure (~~face-based data structure~~)



$$F = \{ f_1, f_2 \dots f_6 \}$$

$$A = \{ a_1, a_2 \dots a_{12} \}$$

$$S = \{ s_1, s_2 \dots s_8 \}$$

Relations :

v F A S

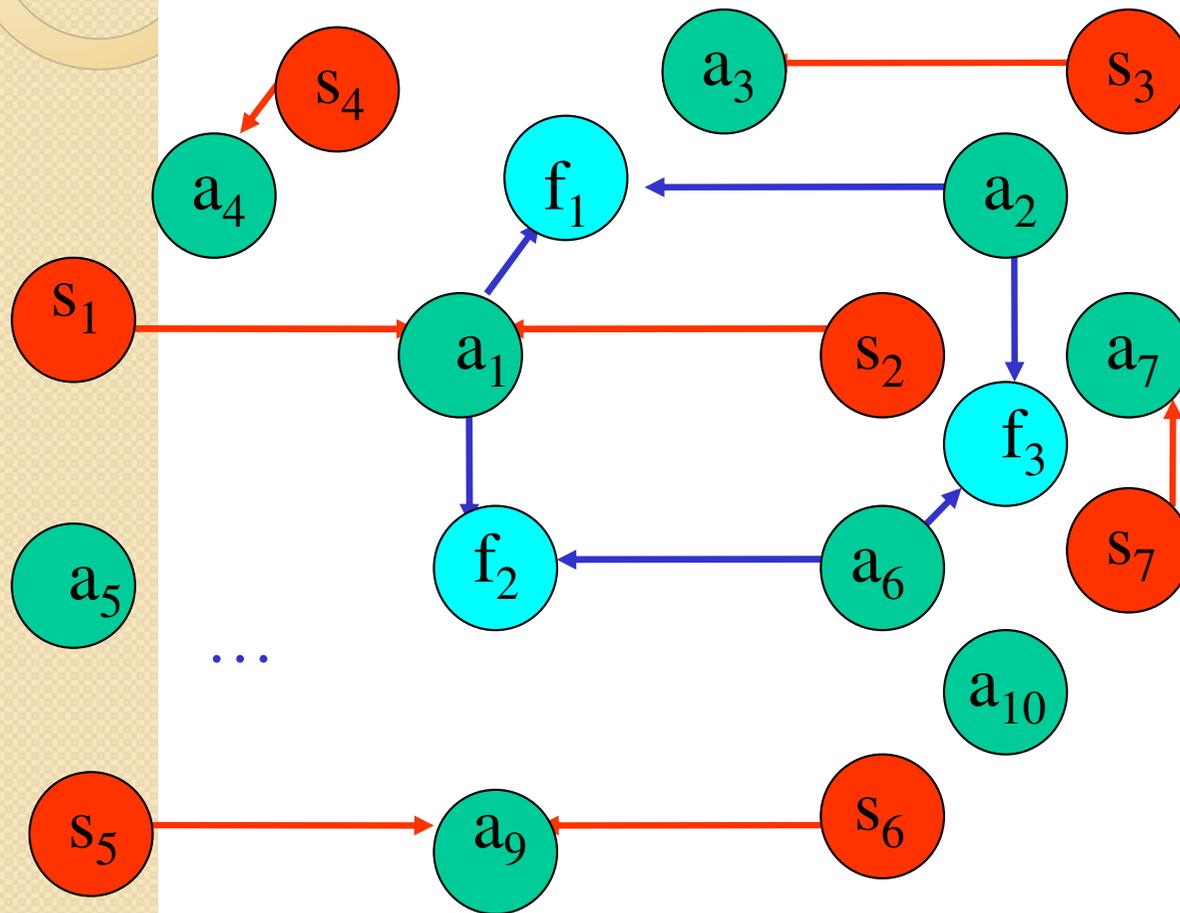
$$b(v) = \{ f_1, f_2 \dots f_6 \}$$

$$b(f_3) = \{ a_2, a_6, a_{10}, a_7 \}$$

$$b(a_6) = \{ s_6, s_2 \}$$

Représentation informatique

Edge-based data structure (~~face-based data structure~~)



$$F = \{ f_1, f_2 \dots f_6 \}$$

$$A = \{ a_1, a_2 \dots a_{12} \}$$

$$S = \{ s_1, s_2 \dots s_8 \}$$

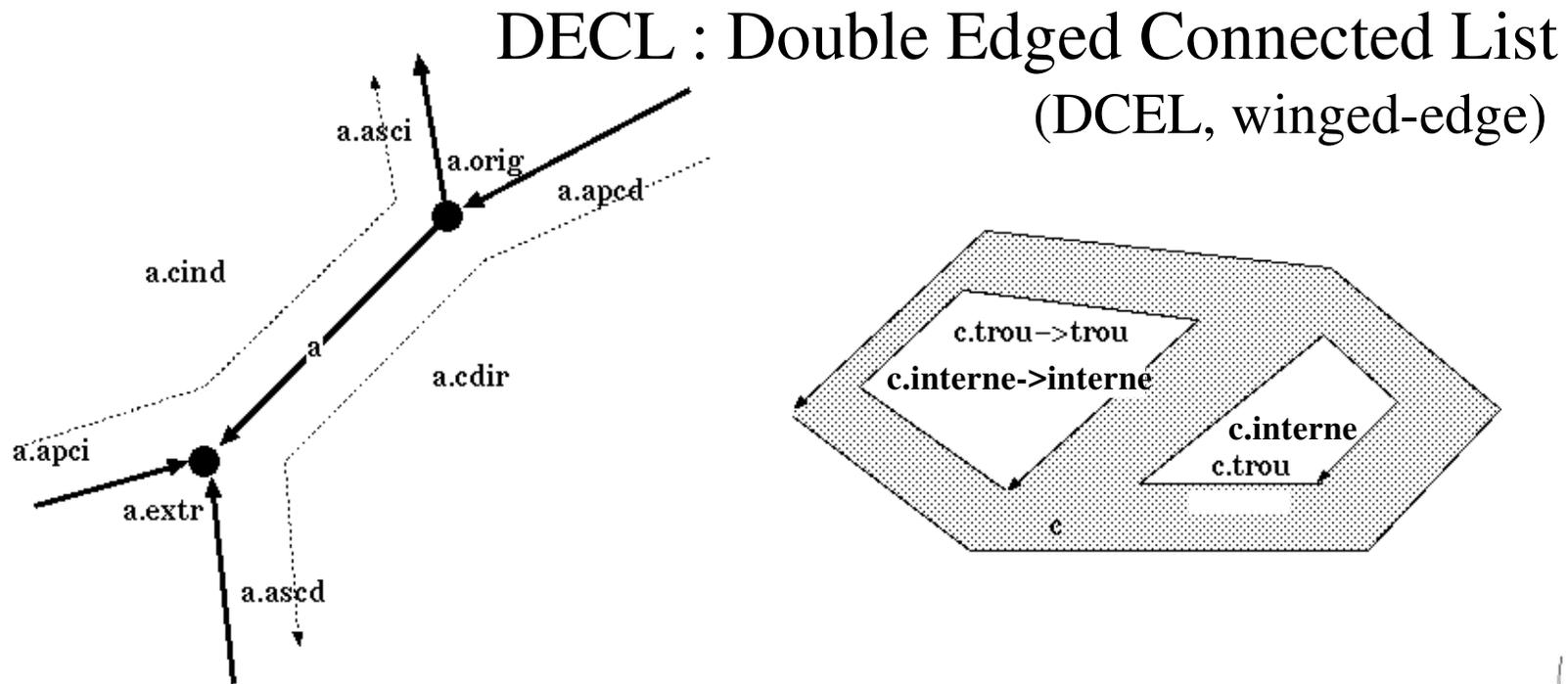
Relations :

s_6 dans a_6 ,
 s_2 dans a_1 ...

a_1 dans f_1 et f_2 ,
 a_2 dans f_1 et f_3 ...

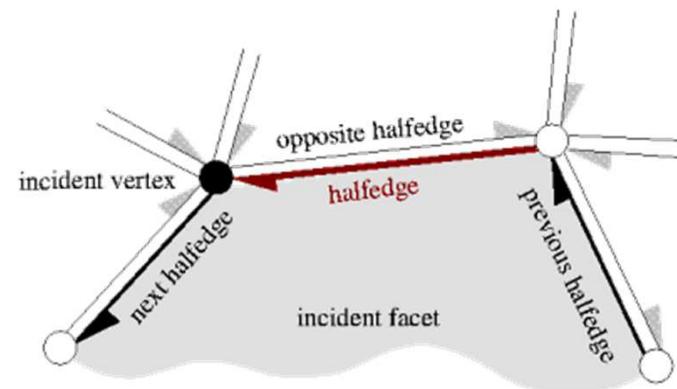
Représentation informatique

Edge-based data structure (~~face-based data structure~~)



HDS : Half-edge Data Structure

CGAL, OpenMesh...



Contraintes d'intégrité

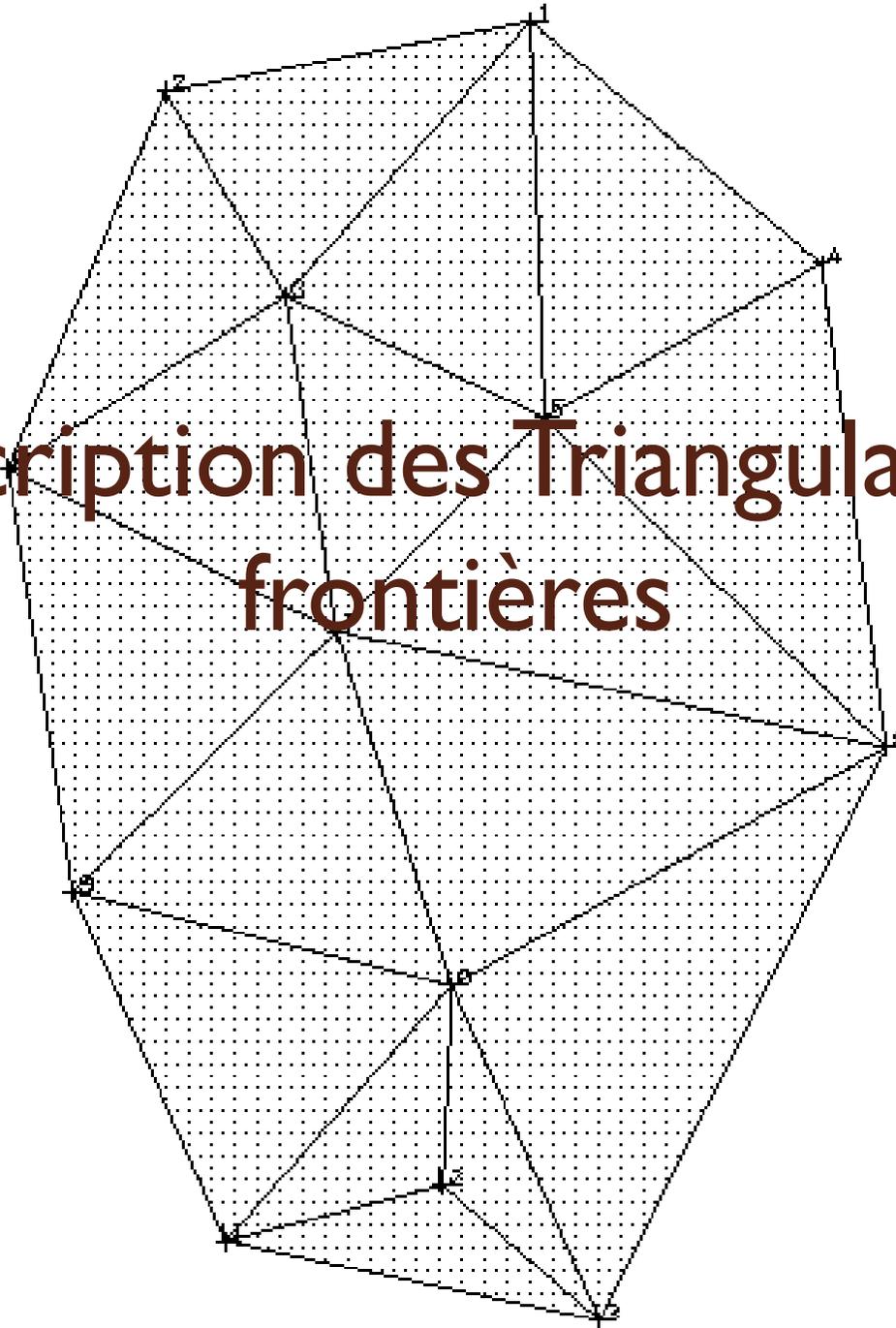
Cas 2D

- Intersection réduites aux sommets communs
- Un sommet appartient exactement à 2 arêtes
- Existe une orientation des arêtes telle que chaque sommet est origine d'une arête et extrémité d'une autre.

Cas 3D

- Intersection réduites aux arêtes communes
- Une arête appartient exactement à 2 faces
- Existe une orientation des faces telle que chaque arête est parcourue dans un sens puis dans l'autre.

Description des Triangulations frontières



Le Triangle

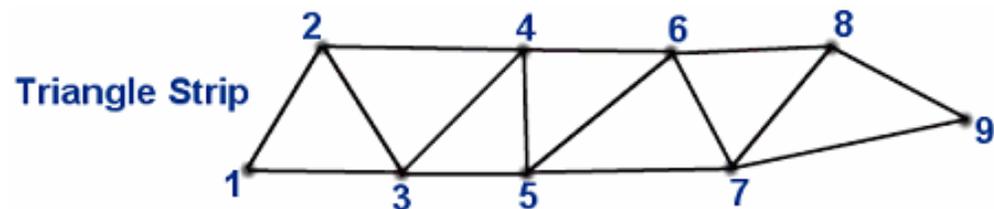
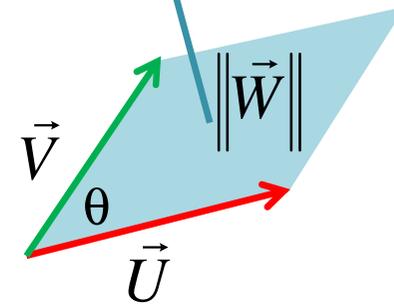
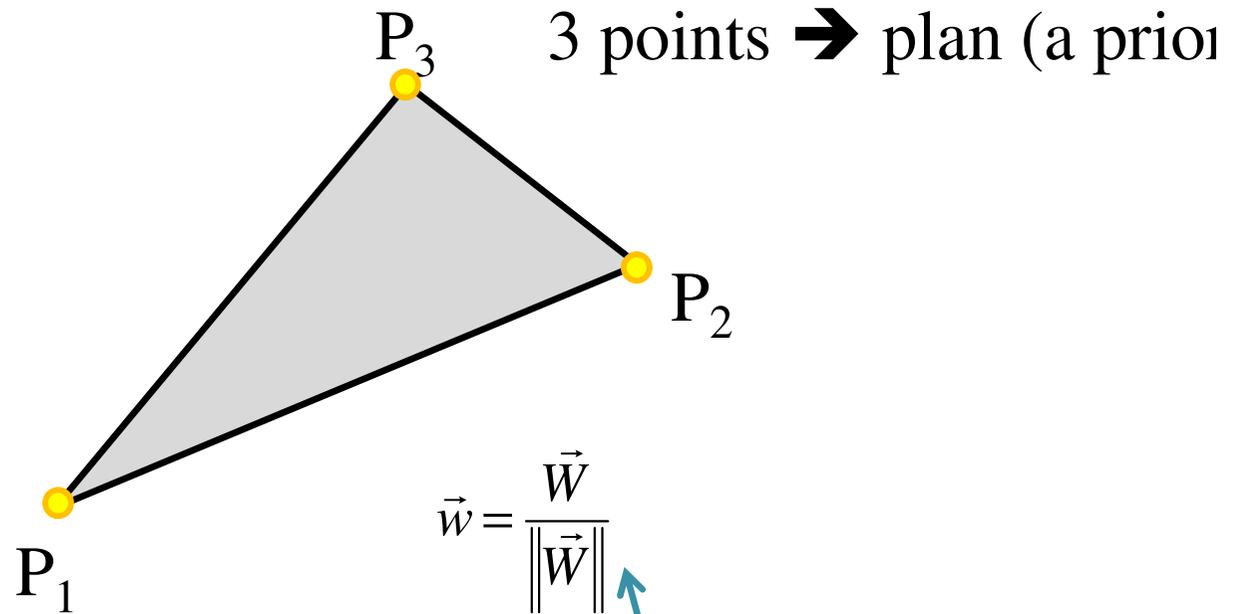
Simplexe,

Convexe,

Normale ?

Aire ?

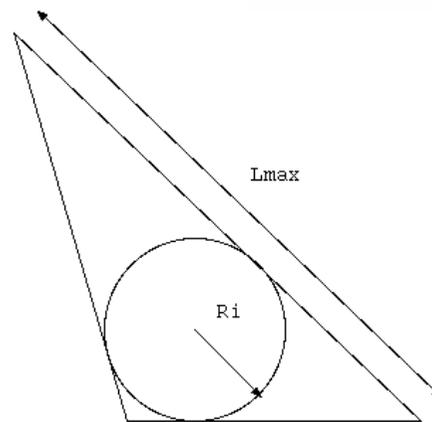
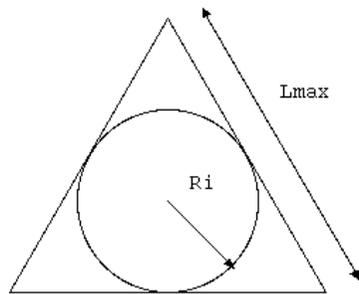
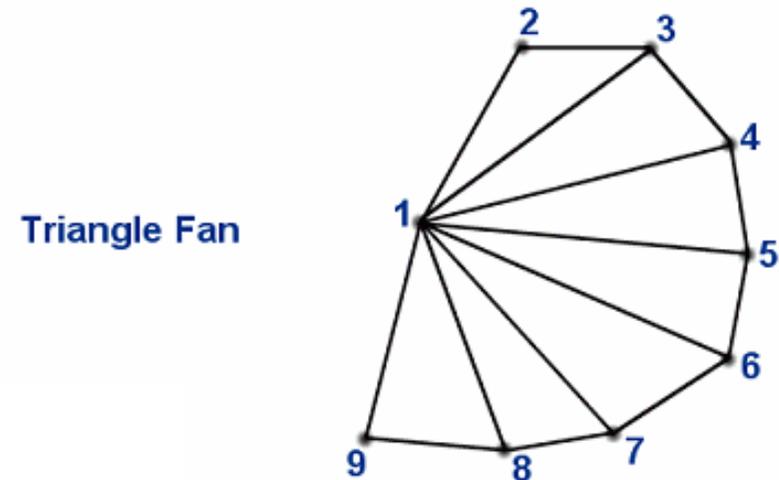
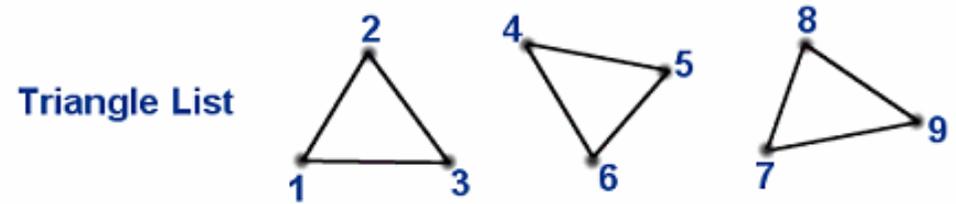
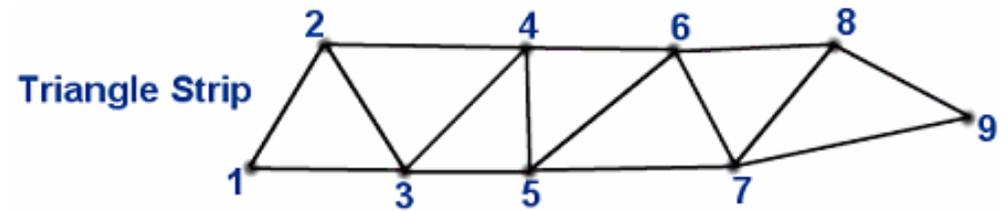
Point dans ?



$$S = \{ (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4), (x_5, y_5, z_5), (), (), \dots \}$$

Le Triangle

- triangle strip
- triangle fan
- triangle soup
- sliver



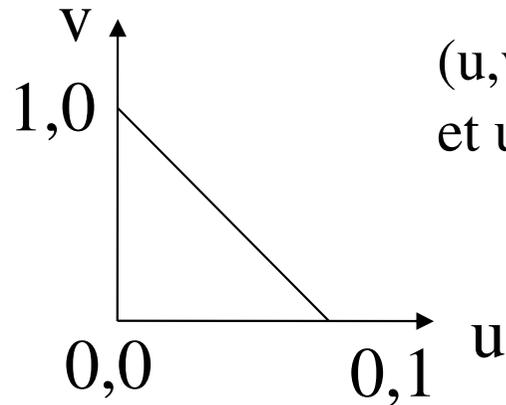
Coefficients de qualité :

$$c_1 R_i / L_{\max}$$

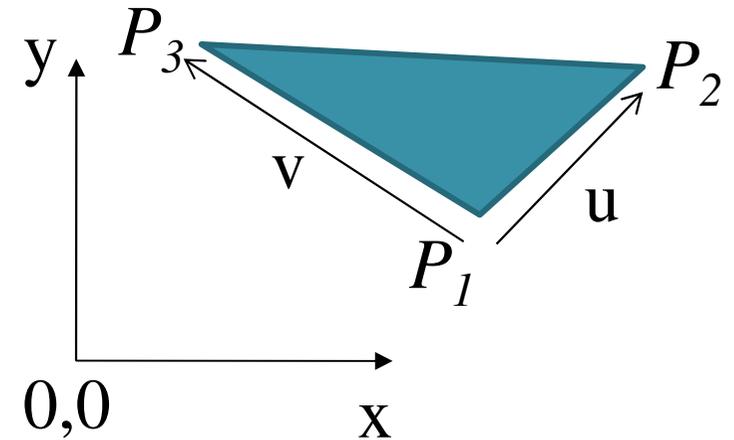
$$l_{\min} / L_{\max}$$

$$c_3 L_{\max} / R_c$$

Représentation paramétrique



(u,v) dans $[0:1] \times [0:1]$
et $u+v \leq 1$



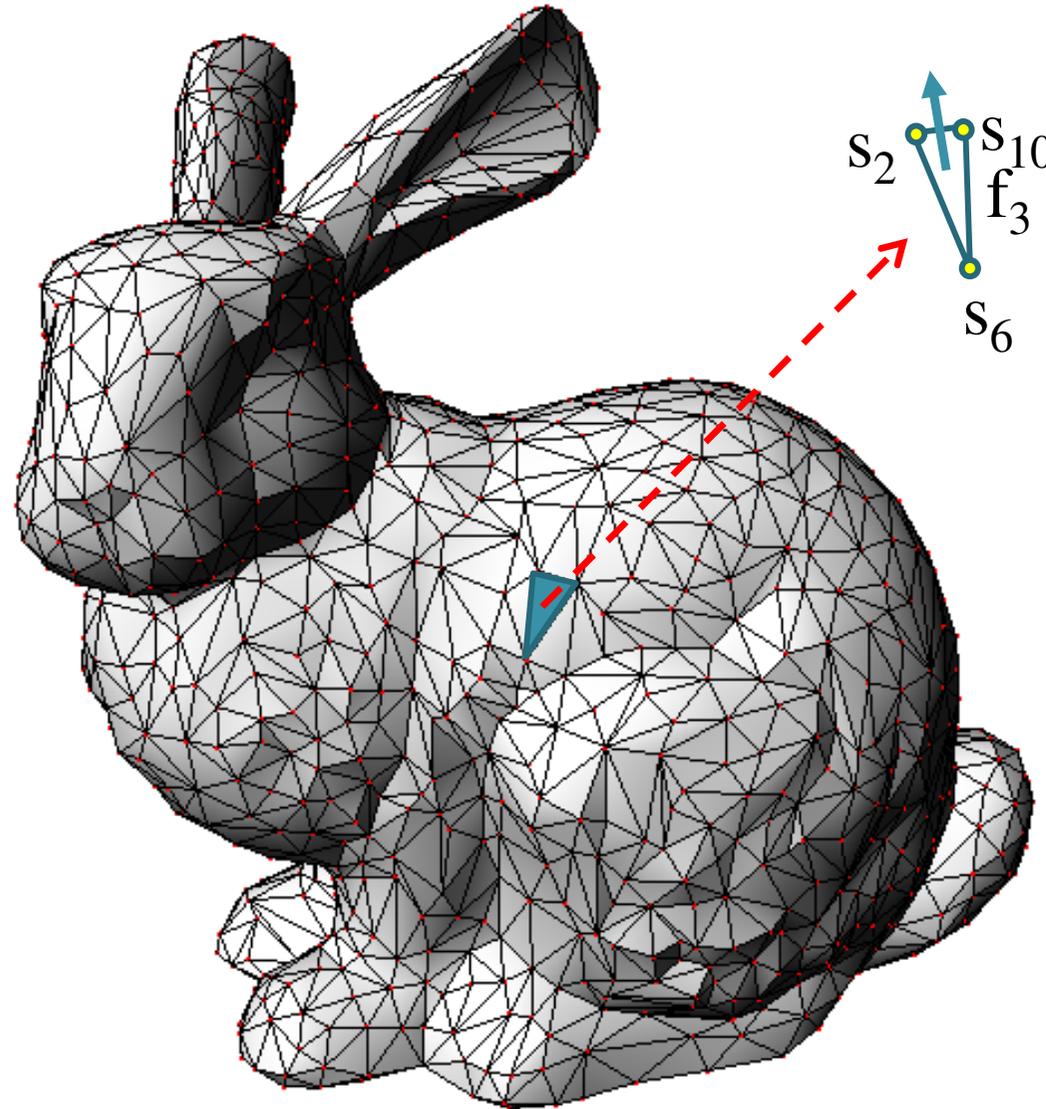
$$x(u,v) = (1-u-v) \cdot x_1 + u \cdot x_2 + v \cdot x_3$$

$$y(u,v) = (1-u-v) \cdot y_1 + u \cdot y_2 + v \cdot y_3$$

$$S(u,v) = \sum_{i=1}^3 P_i P_i(u,v)$$

$$\begin{pmatrix} x(u,v) \\ y(u,v) \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} \begin{vmatrix} 1 & -1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 \\ u \\ v \end{vmatrix}$$

Triangulation frontière



$$F = \{ f_1, f_2 \dots f_6 \}$$

$$A = \{ a_1, a_2 \dots a_{12} \}$$

$$S = \{ s_1, s_2 \dots s_8 \}$$

Frontière:

$$\text{fr}(v) = \{ f_1, f_2 \dots f_6 \}$$

$$\text{fr}(f_3) = \{ a_{14}, a_6, a_{74} \}$$

...

$$\text{fr}(a_{14}) = \{ s_2, s_6 \}$$

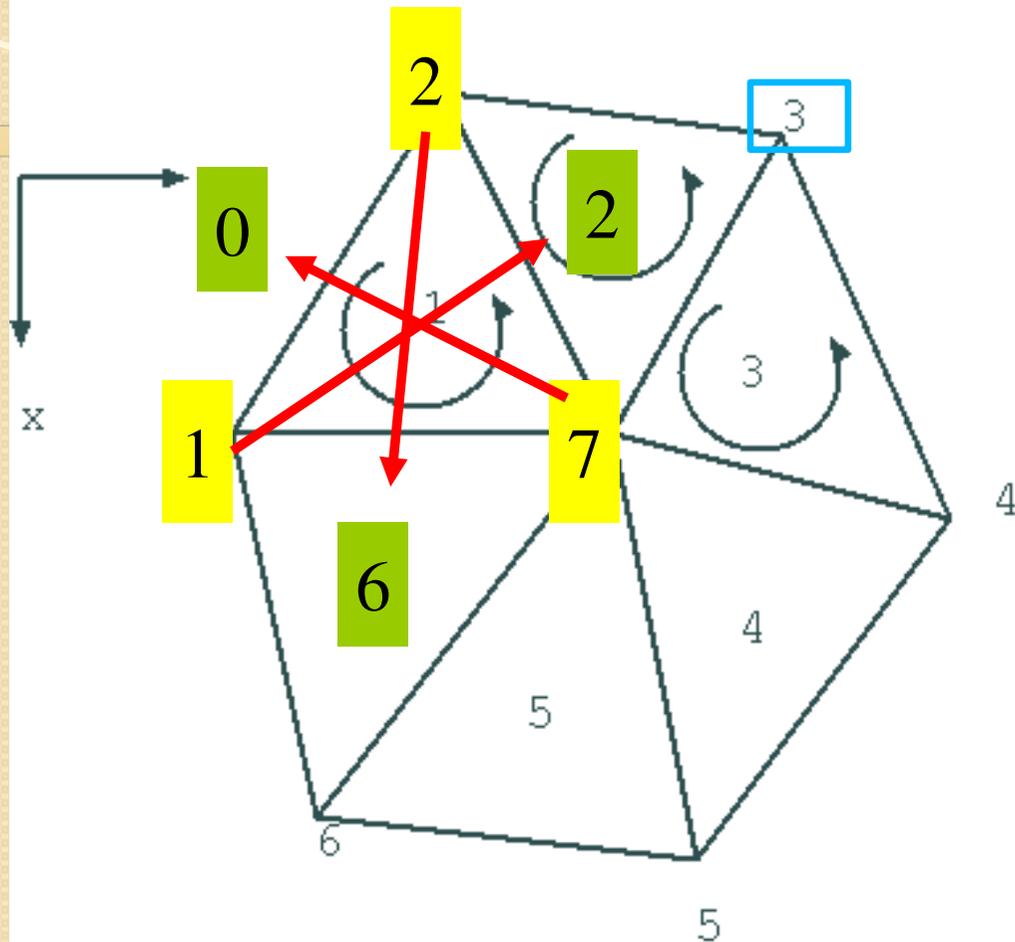
$$\text{fr}(a_6) = \{ s_6, s_{10} \}$$

Triangulations :

$$\text{fr}(f_3) = \{ s_2, s_6, s_{10} \}$$

Structure de données

Face-based data structure (~~edge-based data structure~~)



ITRNOE

1	1,7,2
2	3,2,7
3	7,4,3
4	7,5,4
...	...

ITRTRI

1	2,0,6
2	1,3,0
3	0,2,4
4	0,3,5
...	...

COORD

1	1.5,1.2
2	-.5,2.5
3	-.2,4.5
...	...

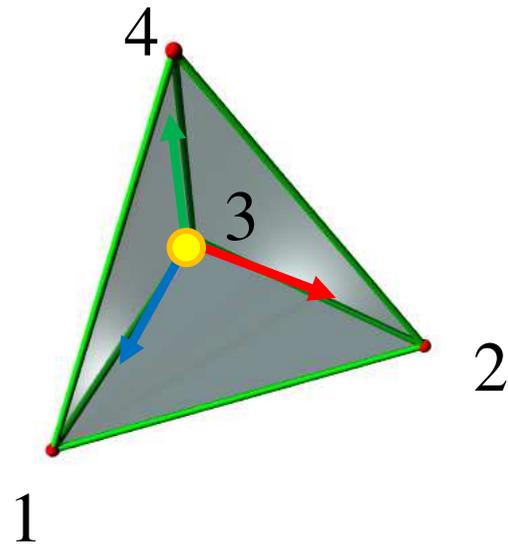
NOETRI

6
2
2
...

Algo de parcours ?

Optimisation ?

EXERCICE : décrire le tétra. dans la structure de données



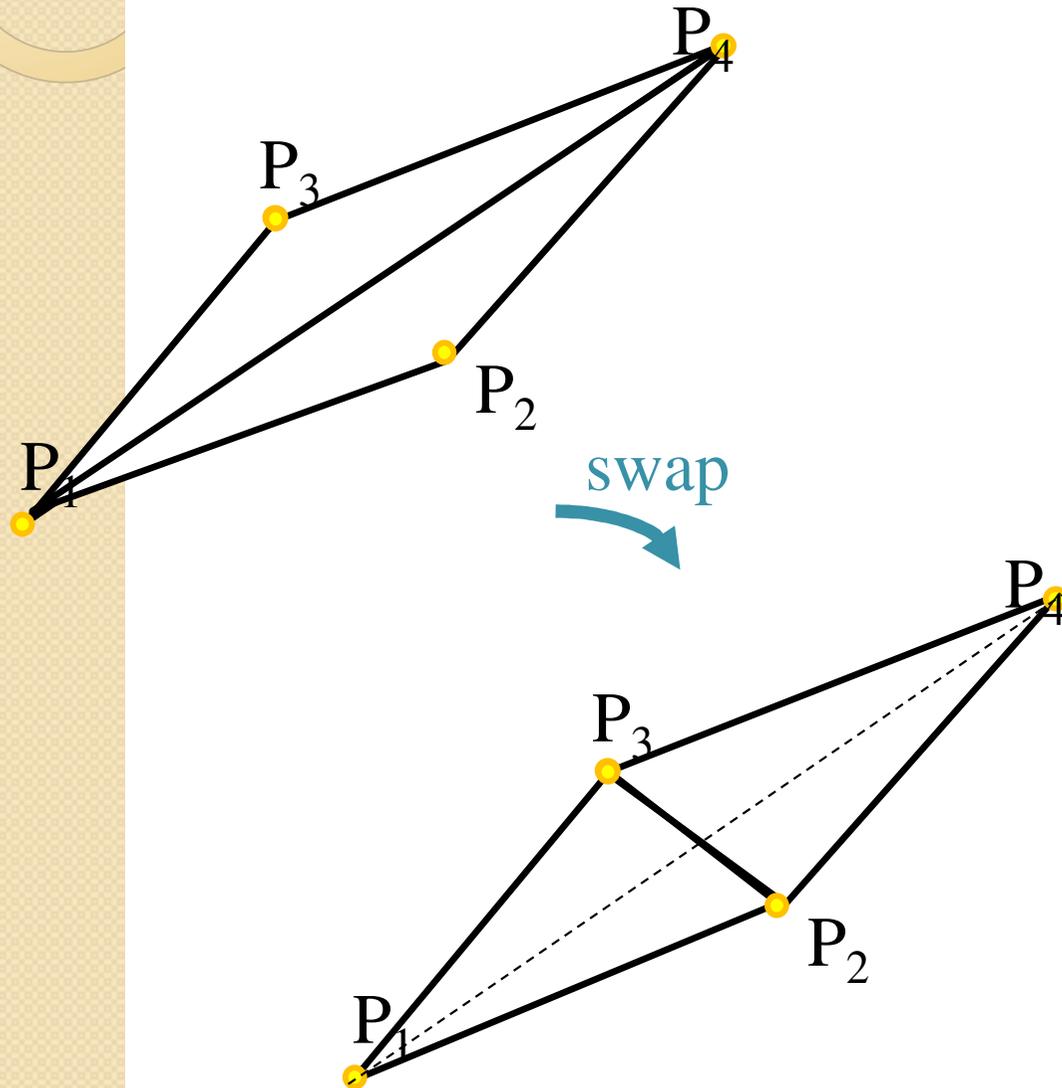
ITRNOE

ITRTRI

COORD

NOETRI

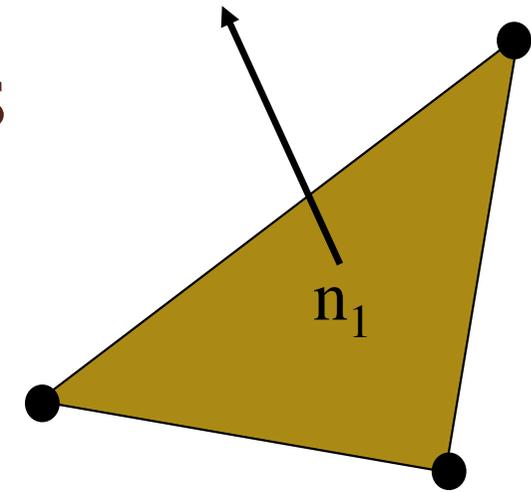
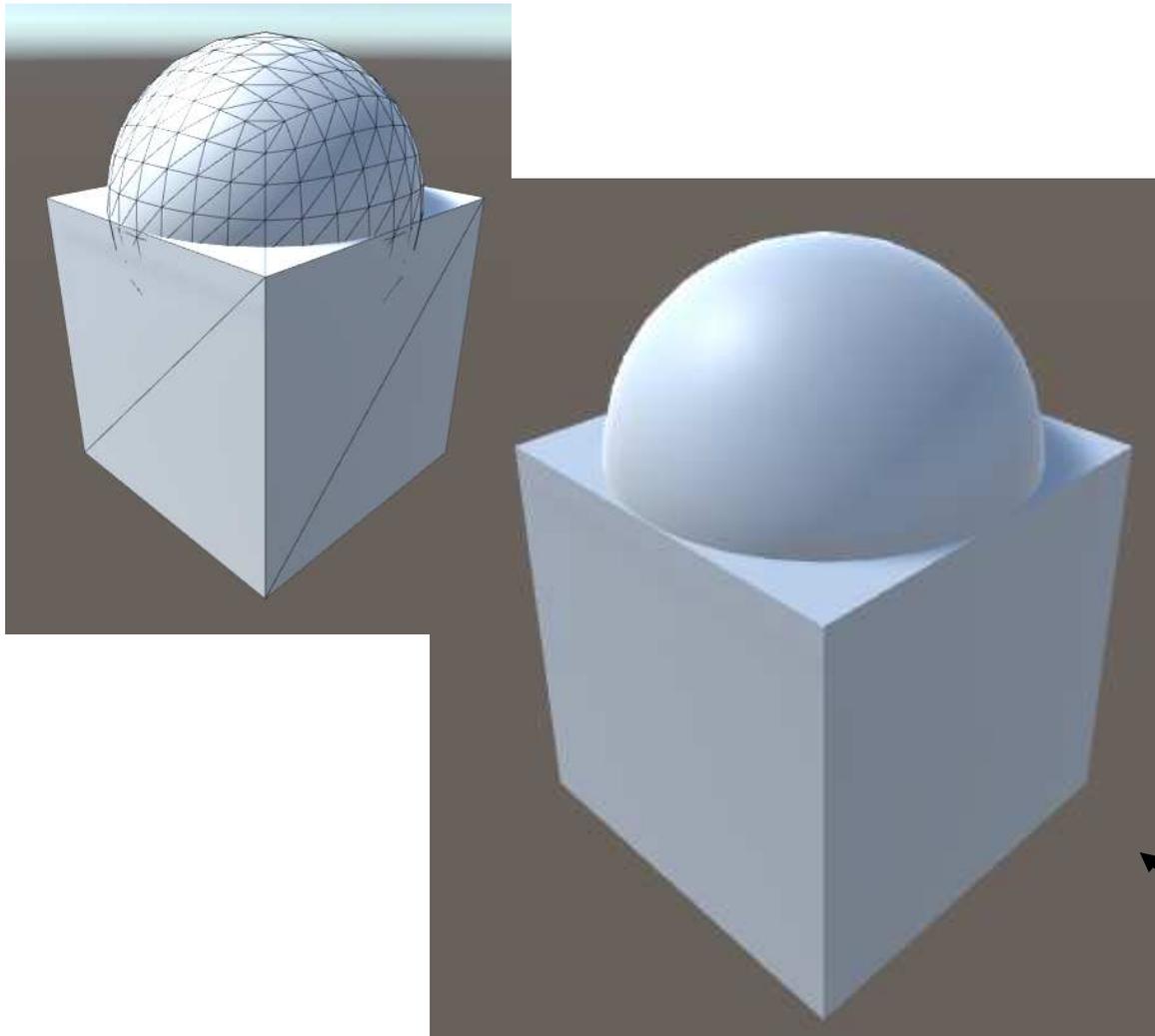
EXERCICE : Opérateurs élémentaires sur la structure



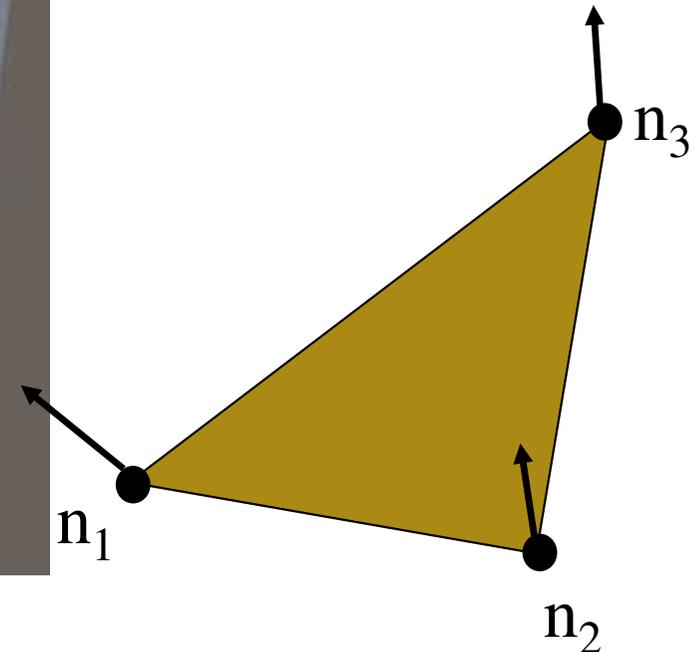
ITRNOE	ITRTRI

ITRNOE	ITRTRI

Faces planes et/ou courbes

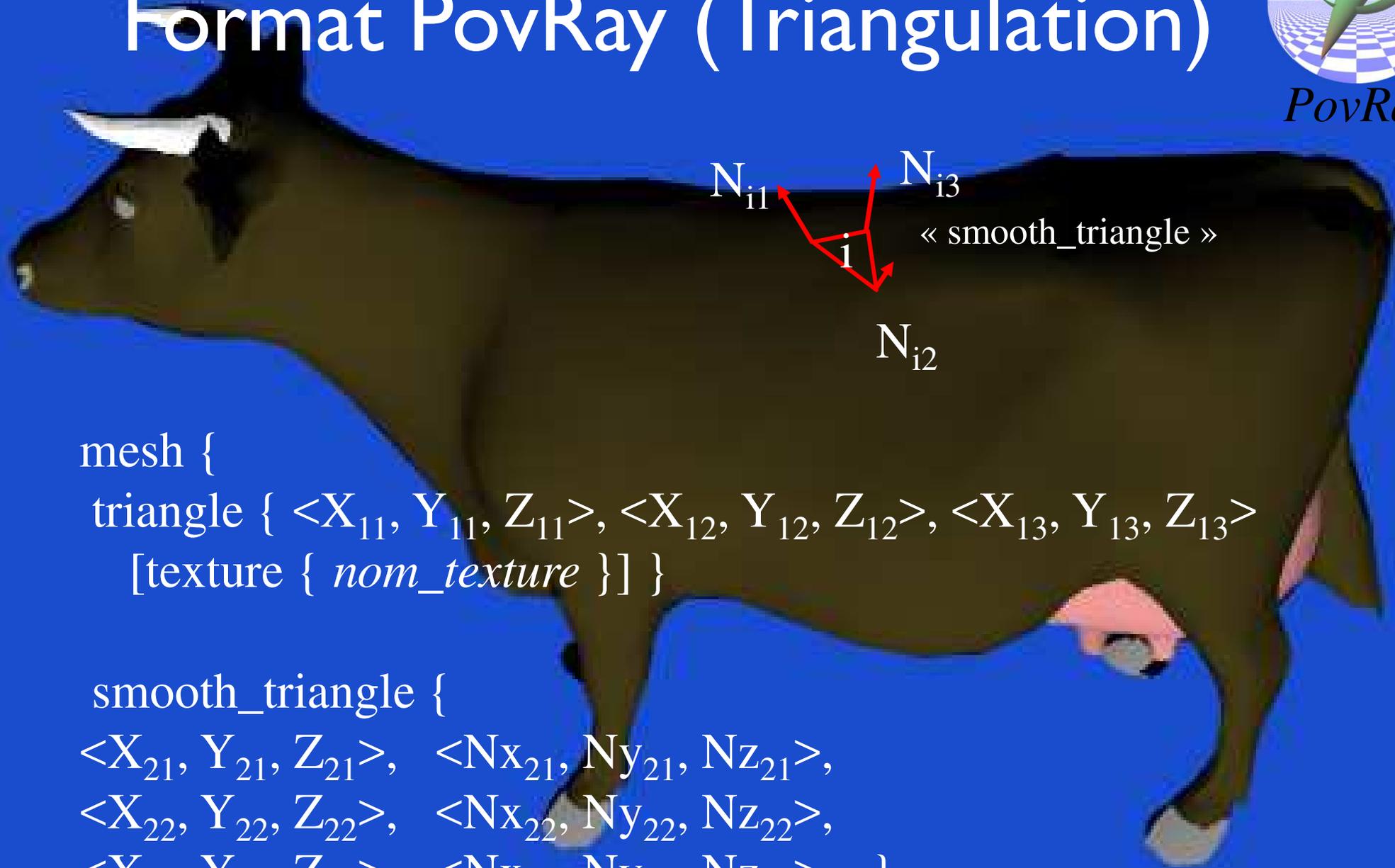


Surface plane
Surface courbe



Combien d'information faut-il stocker dans le 1^{er} modèle ?
Dans le second ? Comment mélanger les 2 approches ?

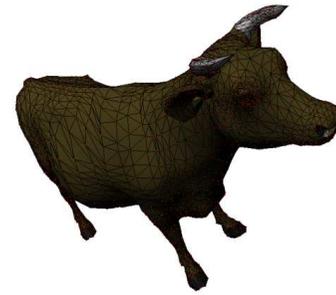
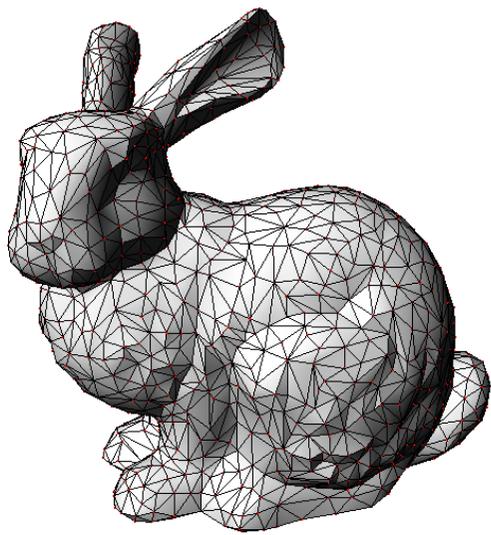
Format PovRay (Triangulation)



```
mesh {  
  triangle { <X11, Y11, Z11>, <X12, Y12, Z12>, <X13, Y13, Z13>  
    [texture { nom_texture } ] }
```

```
smooth_triangle {  
  <X21, Y21, Z21>, <Nx21, Ny21, Nz21>,  
  <X22, Y22, Z22>, <Nx22, Ny22, Nz22>,  
  <X23, Y23, Z23>, <Nx23, Ny23, Nz23> ... }  
... }
```

Formats ...



WRL

```
#VRML V2.0 utf8
...
Geometry IndexedFaceSet {
# optionnel mais utile :
ccw TRUE
solid TRUE
convex TRUE
coord Coordinate {
point [
# Coordonnees des points
X1 Y1 Z1
X2 Y2 Z2
...
]}
coordIndex [
# Les faces
1 2 3 4 -1
5 6 2 1 -1
...
]
}
```

STL

```
solid name
...
facet normal ni nj nk
outer loop
vertex v1x v1y v1z
vertex v2x v2y v2z
vertex v3x v3y v3z
endloop
endfacet
...
endsolid name
```

OFF

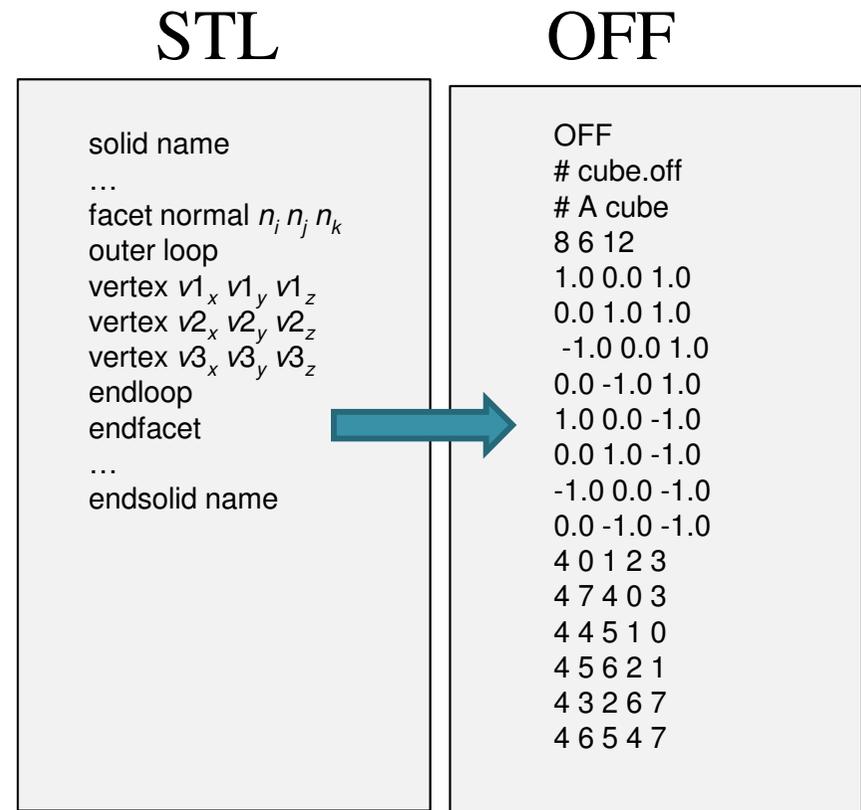
```
OFF
# cube.off
# A cube
8 6 12
1.0 0.0 1.0
0.0 1.0 1.0
-1.0 0.0 1.0
0.0 -1.0 1.0
1.0 0.0 -1.0
0.0 1.0 -1.0
-1.0 0.0 -1.0
0.0 -1.0 -1.0
4 0 1 2 3
4 7 4 0 3
4 4 5 1 0
4 5 6 2 1
4 3 2 6 7
4 6 5 4 7
```

OBJ

```
# cube.obj
# list of vertices
v 1.0 0.0 1.0
v 0.0 1.0 1.0
v -1.0 0.0 1.0
v 0.0 -1.0 1.0
v 1.0 0.0 -1.0
v 0.0 1.0 -1.0
v -1.0 0.0 -1.0
v 0.0 -1.0 -1.0
...
# list of faces
f 1 2 3 4
f 8 5 1 4
...
# mais aussi :
# f node/norm/texture
f 6/4/1 2/5/3 7/6/8 6/4/5
```

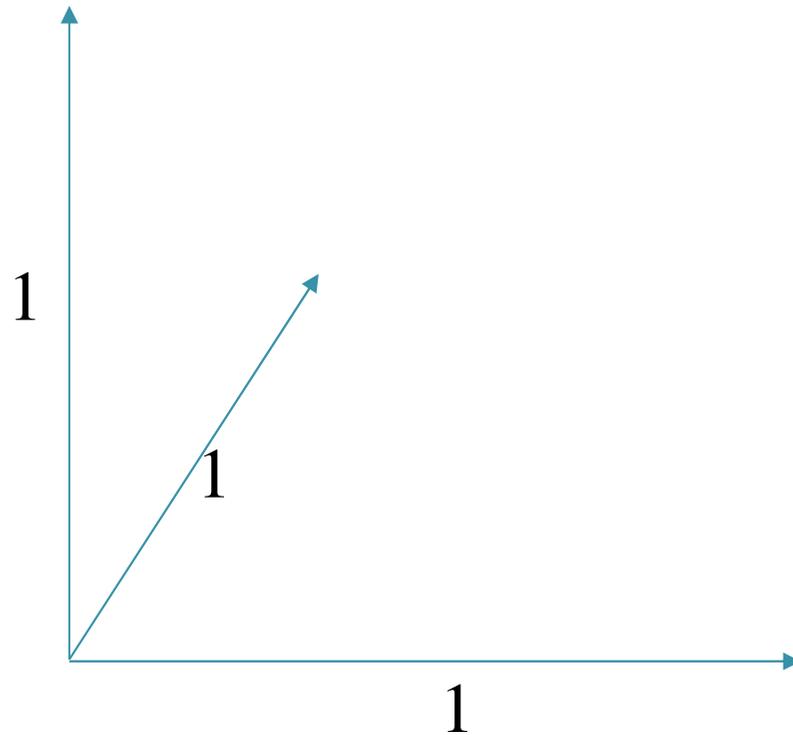
EXERCICE : STL 2 OFF

- Étapes de l'algorithme de reconstruction d'une frontière à partir du format STL



EXERCICE : fichier « OFF »

- Représentez par un dessin la géométrie correspondant au fichier « OFF »



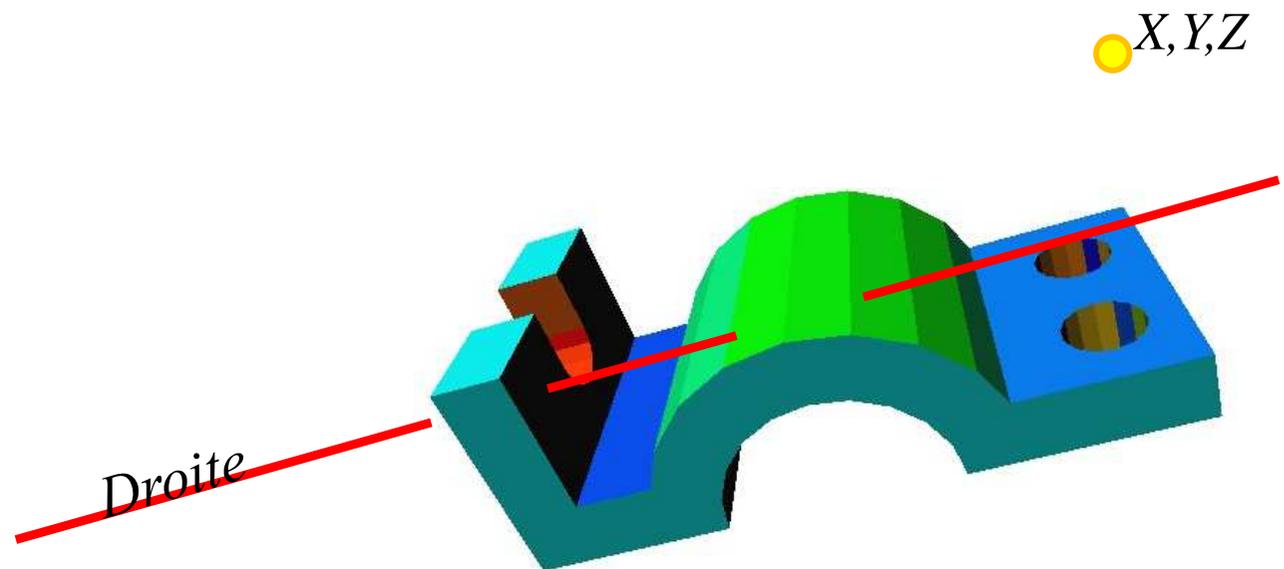
```
OFF
# cube.off
# A cube
8 6 12
1.0 0.0 1.0
0.0 1.0 1.0
-1.0 0.0 1.0
0.0 -1.0 1.0
1.0 0.0 -1.0
0.0 1.0 -1.0
-1.0 0.0 -1.0
0.0 -1.0 -1.0
4 0 1 2 3
4 7 4 0 3
4 4 5 1 0
4 5 6 2 1
4 3 2 6 7
4 6 5 4 7
```

3. Interrogation du modèle



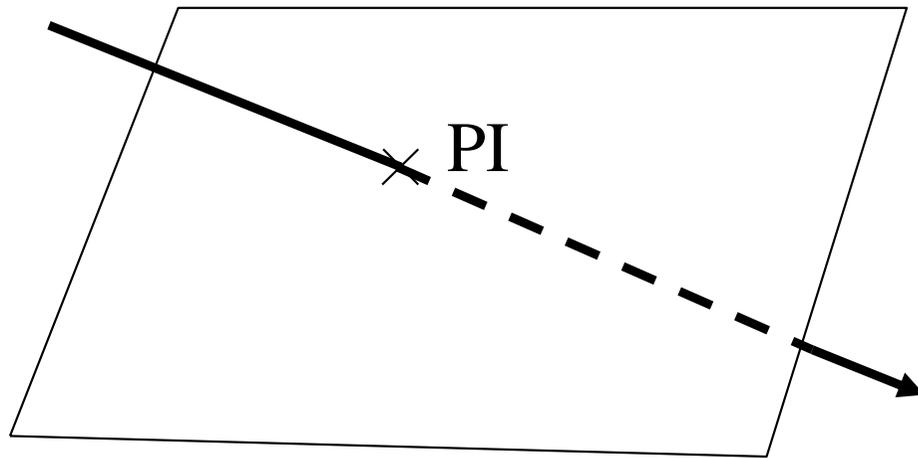
Interrogation du modèle (3D)

- Intersection d'une droite avec le modèle ?
- Un point (x,y) est-il à l'intérieur ou l'extérieur du polyèdre ?
- Volume d'un polyèdre ?
- ...



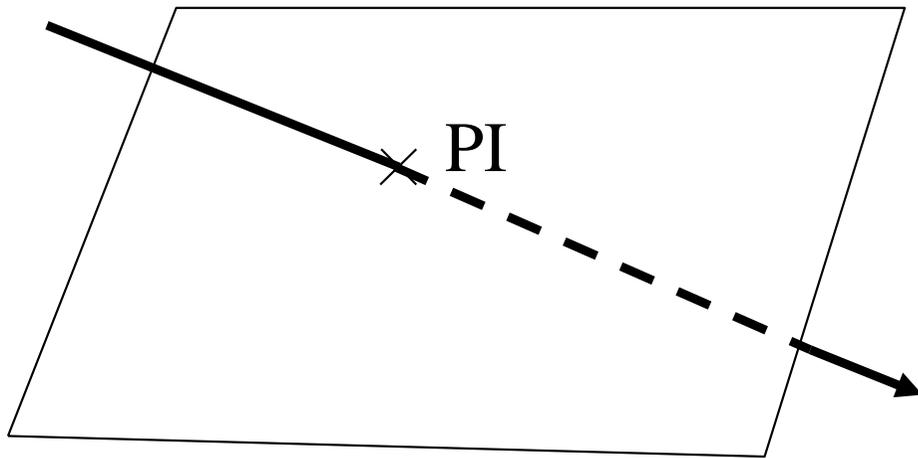
Intersection d'une Droite avec un Plan

Rappels de géométrie (1)



Intersection d'une Droite avec un Plan

Rappels de géométrie (1)



Plan : $Ax+By+Cz+D=0$

Droite : $x(t)=a_1t+a_2$

$y(t)=b_1t+b_2$

$z(t)=c_1t+c_2$

=> équation linéaire :

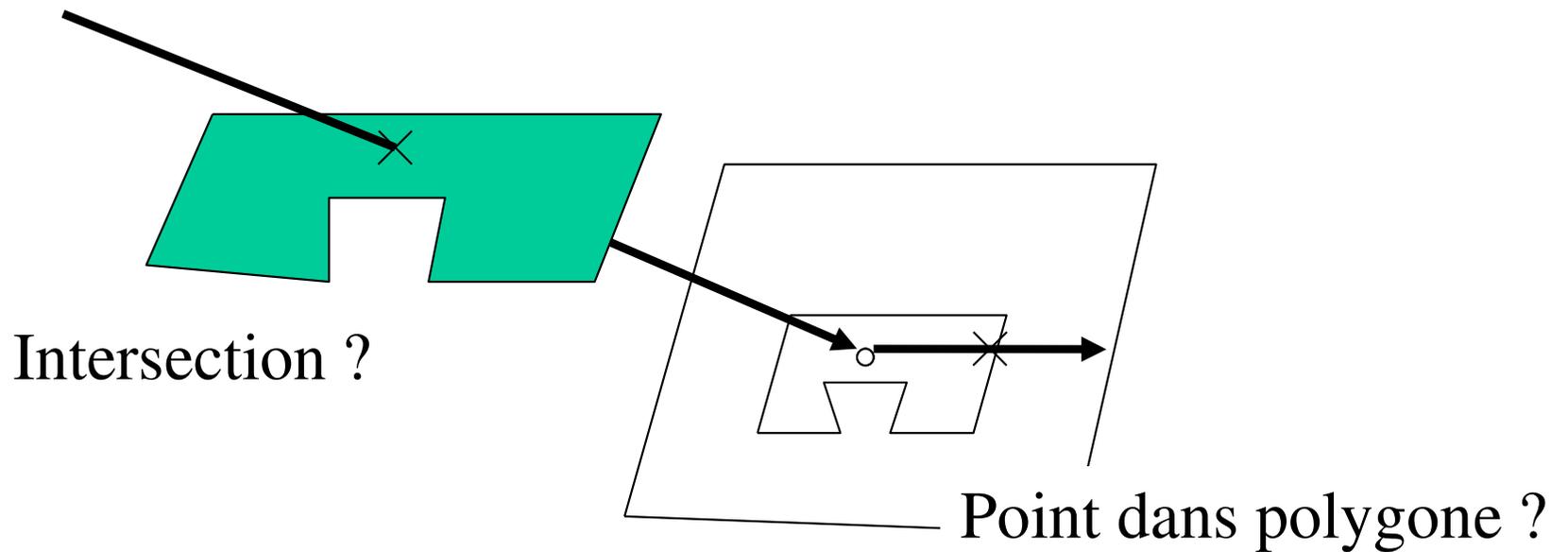
$$t_i = -\frac{Aa_2+Bb_2+Cc_2+D}{Aa_1+Bb_1+Cc_1}$$

PI ($x(t_i), y(t_i), z(t_i)$)

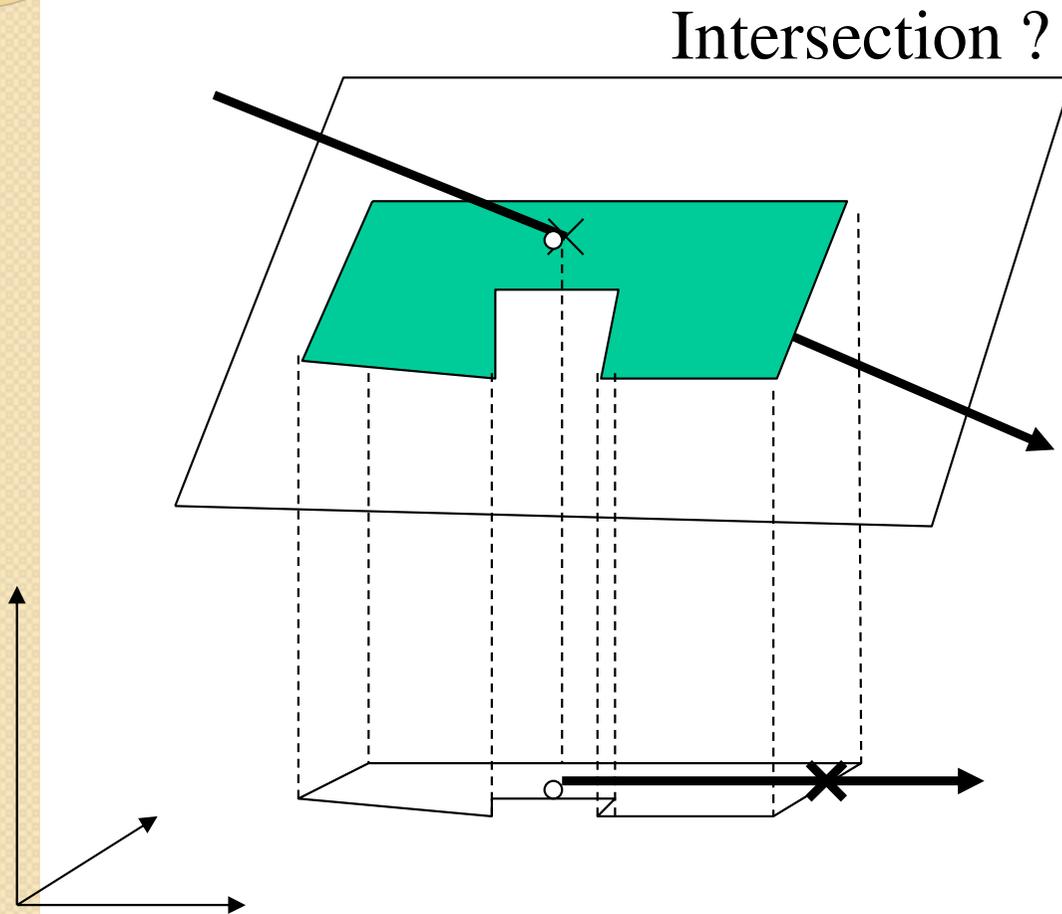
Intersection d'une droite avec un Polyèdre (I)

METHODE : Pour toutes les faces du polyèdre...

Projeter le polygone dans le plan perpendiculaire à la droite



Intersection d'une droite avec un Polyèdre (2)



1) Intersection de la droite avec le plan

2) Projection du polygone et du point d'intersection dans un plan

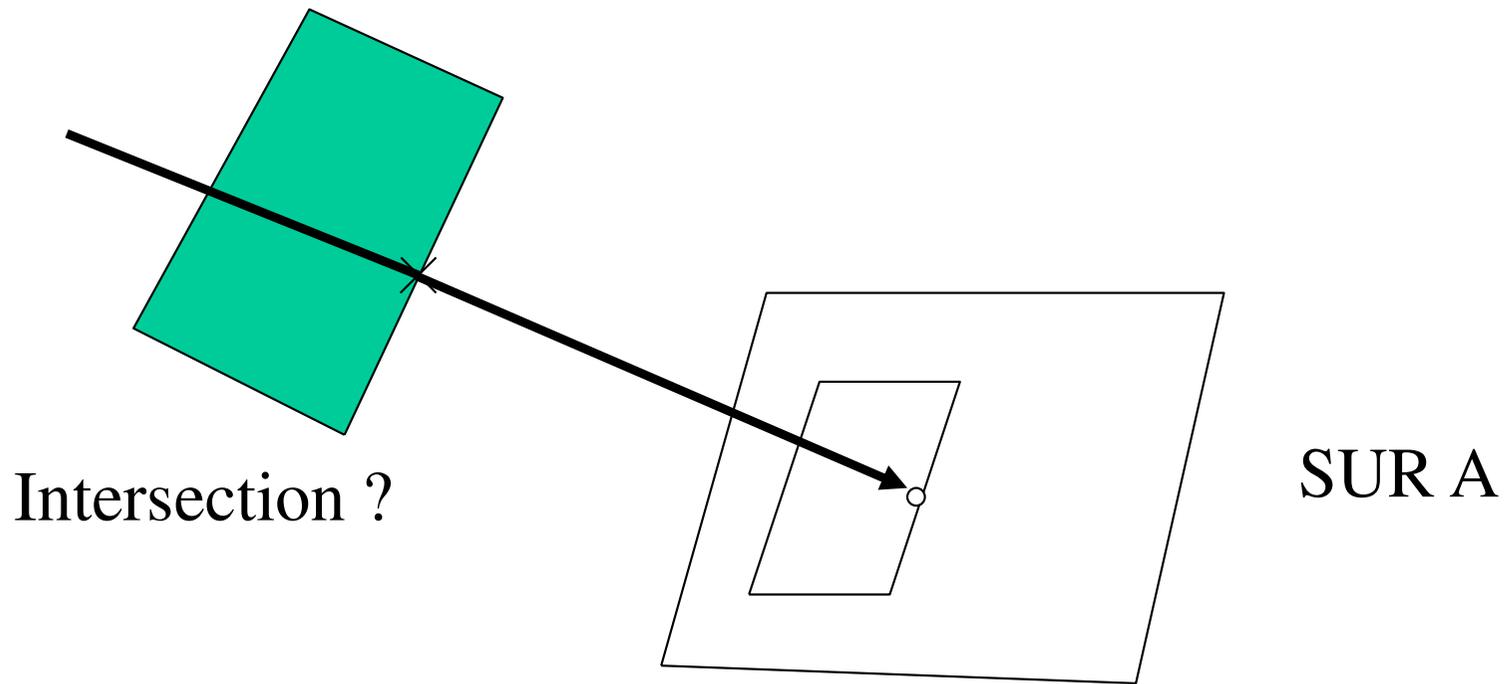
3) Point dans polygone ?

Point dans polyèdre ?

- PRINCIPE : idem 2D, parité du nombre d'intersection avec une $\frac{1}{2}$ droite
- METHODE :
 - Projection des faces
 - Traitement des cas singuliers

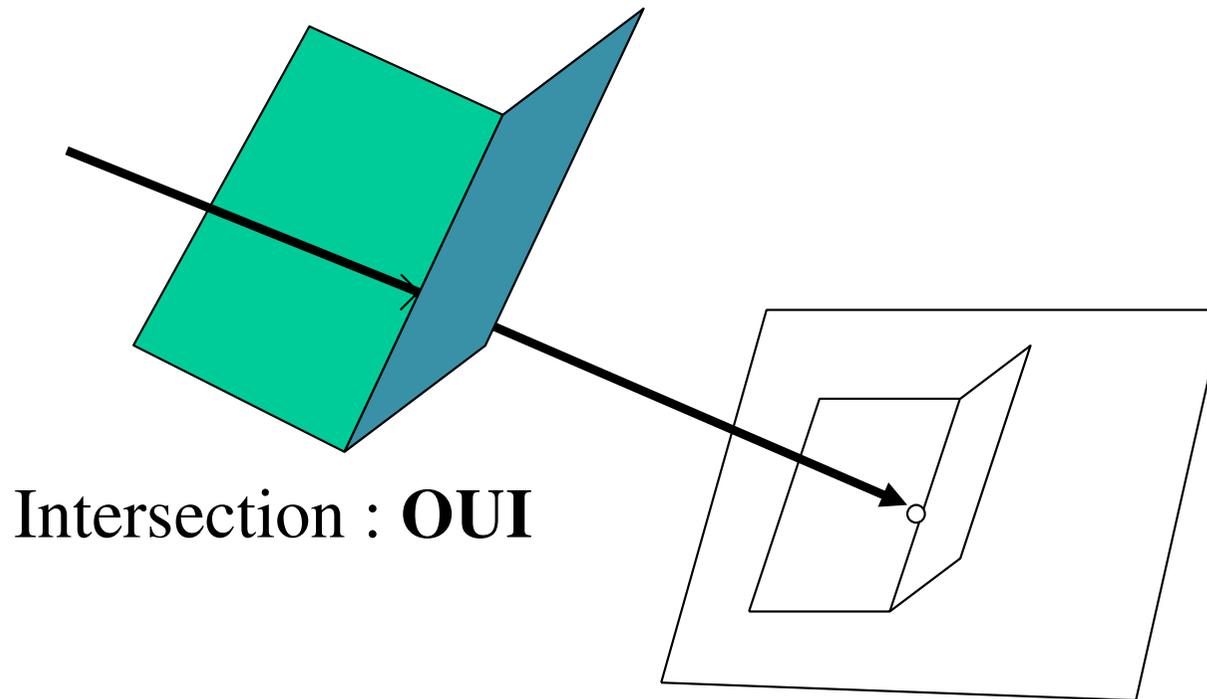
Point dans polyèdre ?

- Cas singuliers



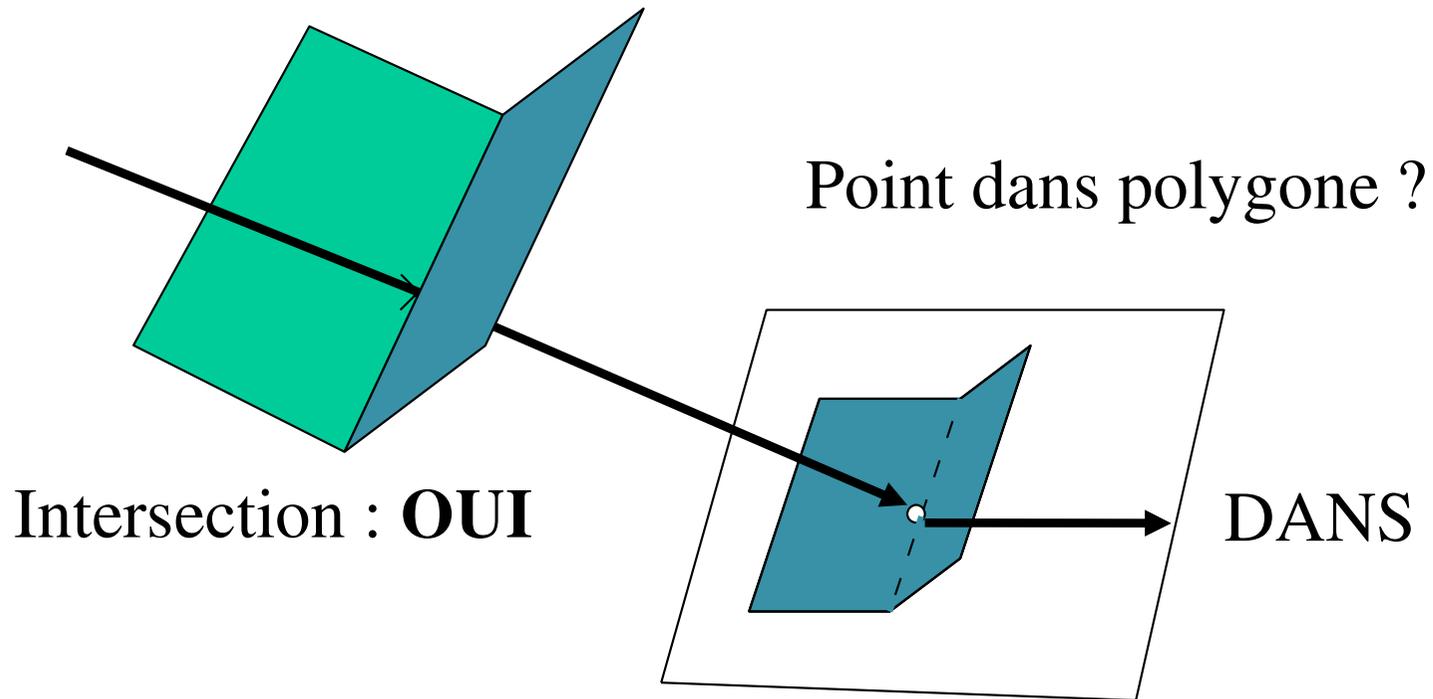
Point dans polyèdre ?

- Cas singuliers



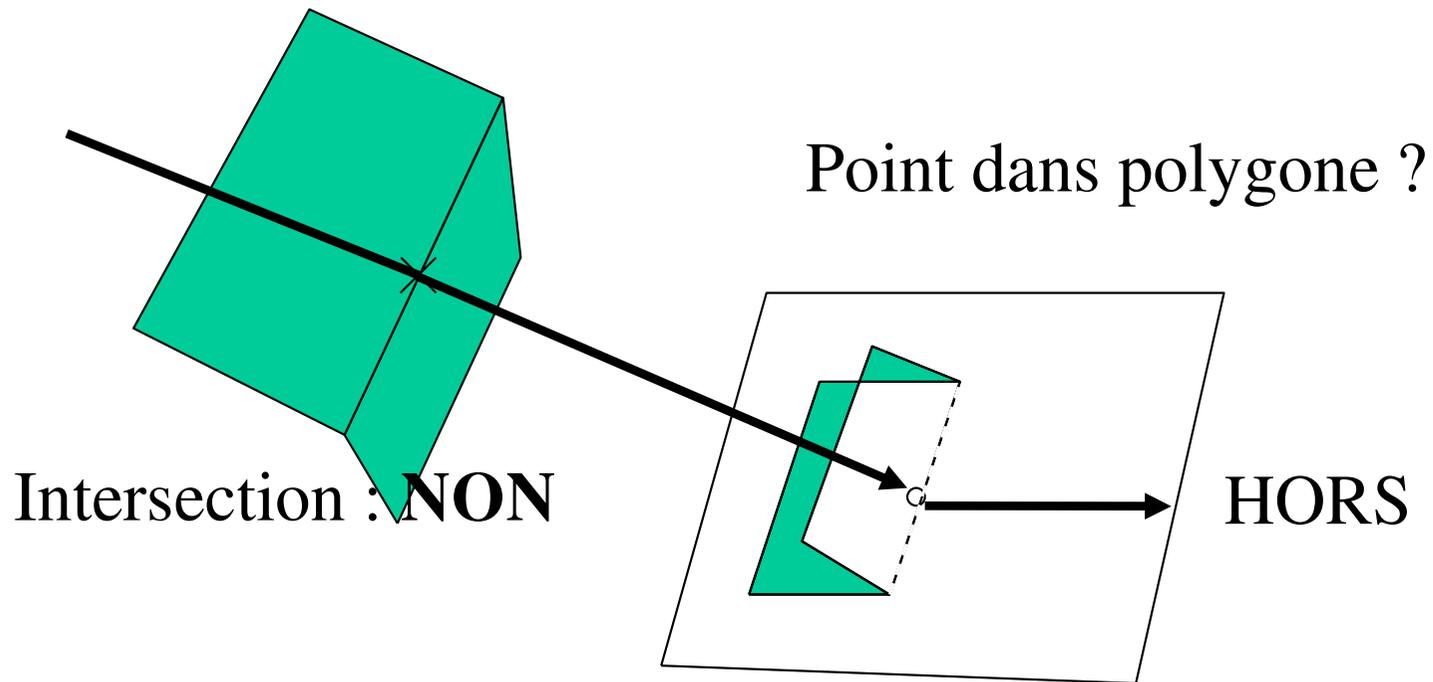
Point dans polyèdre ?

- Cas singuliers



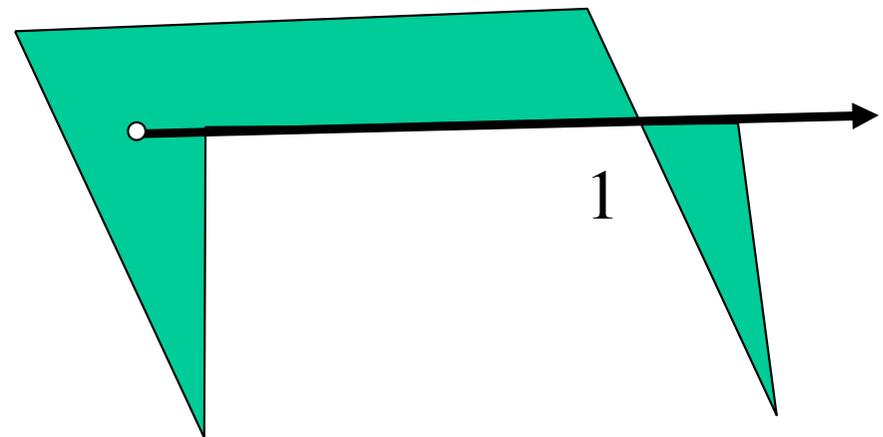
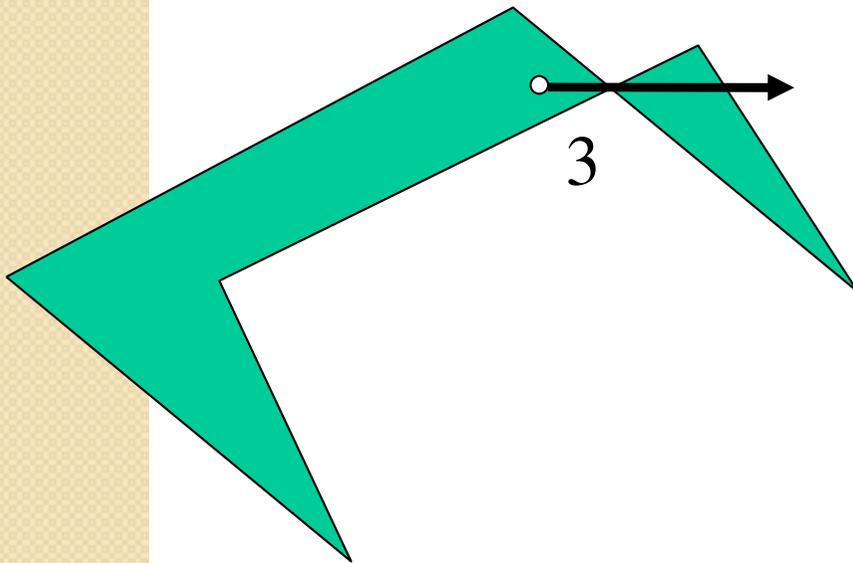
Point dans polyèdre ?

- Cas singuliers

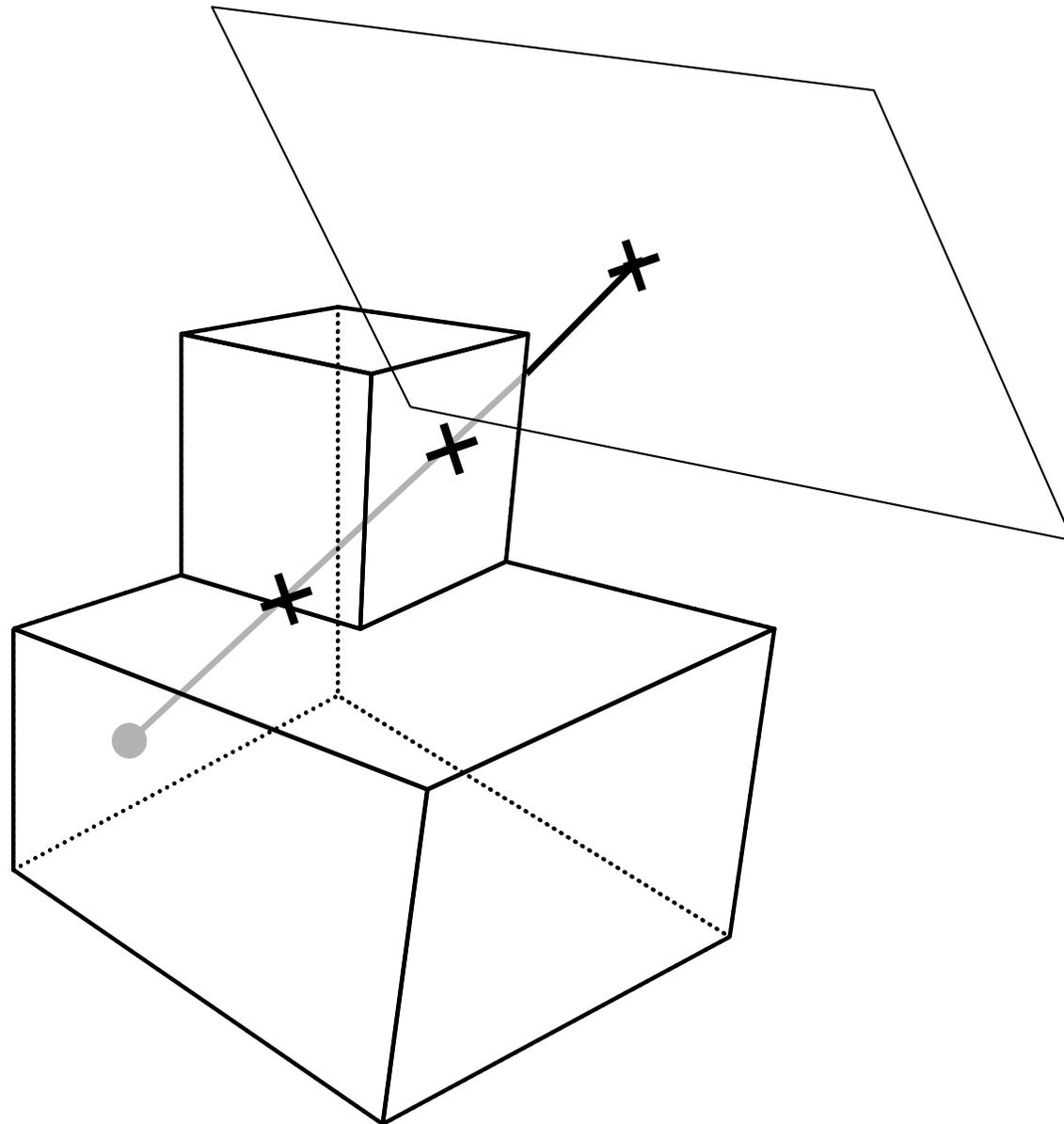


Point dans polyèdre ?

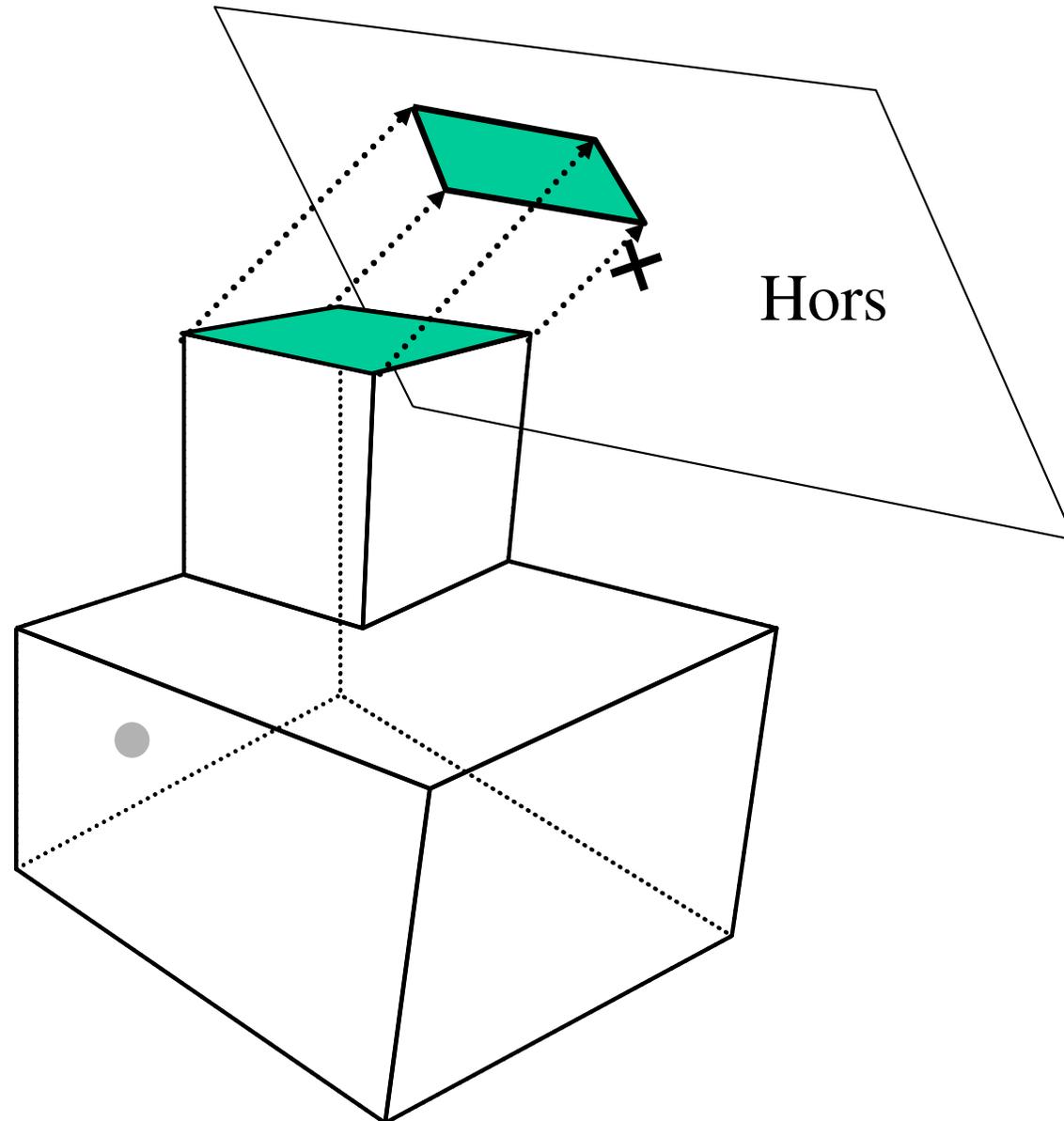
- Point dans polygone replié ?



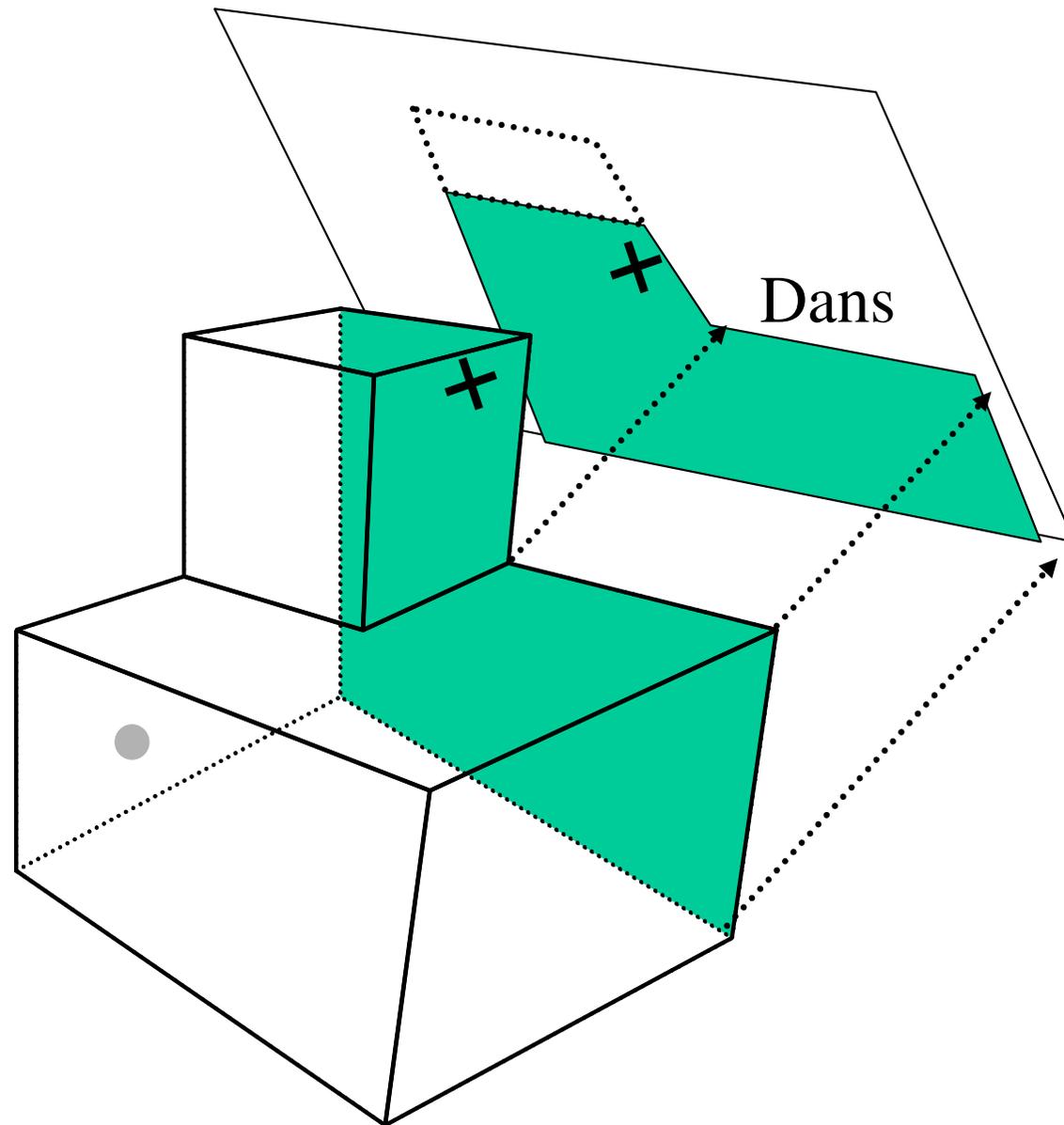
Point dans polyèdre (I)



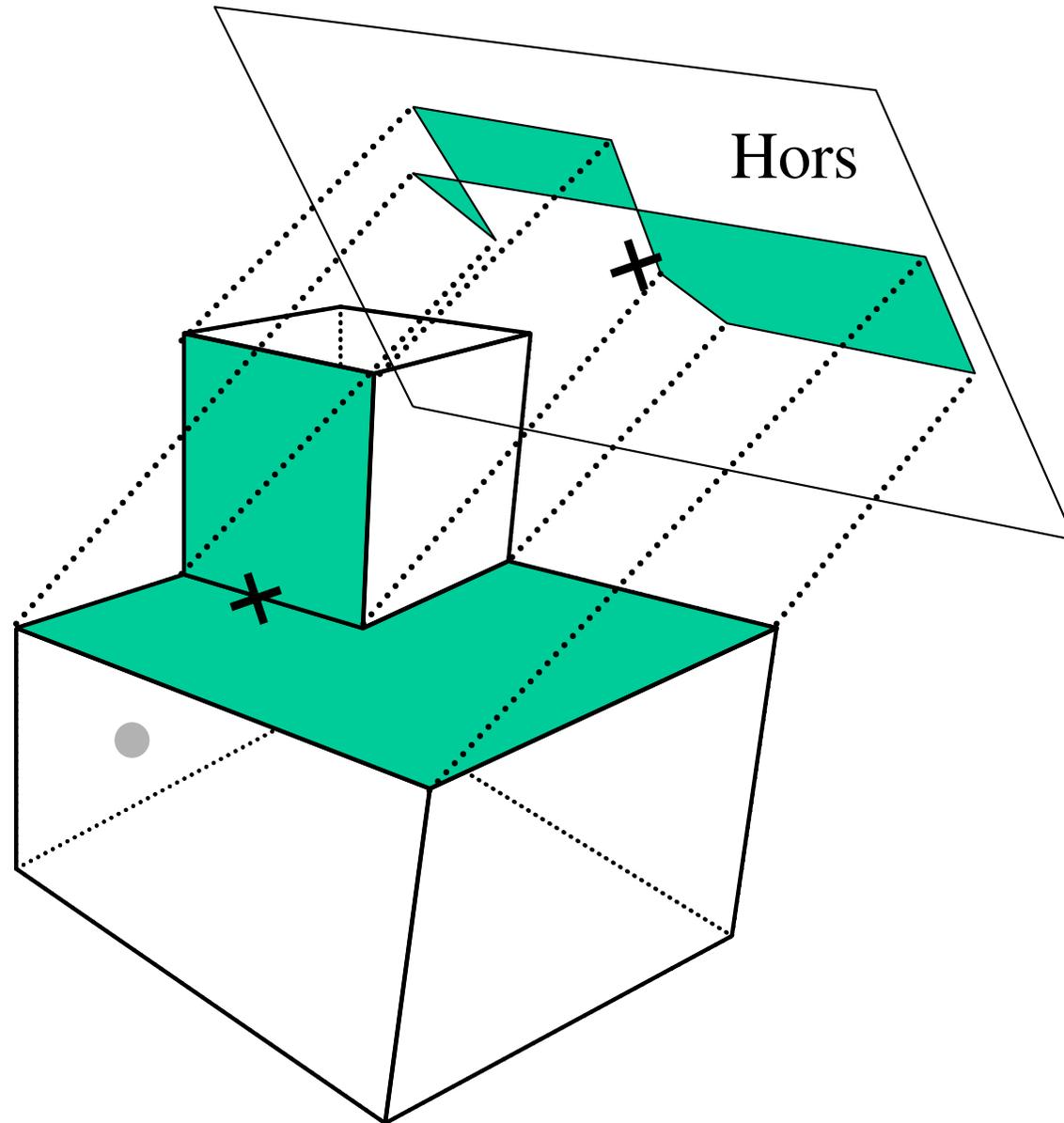
Point dans polyèdre (2)



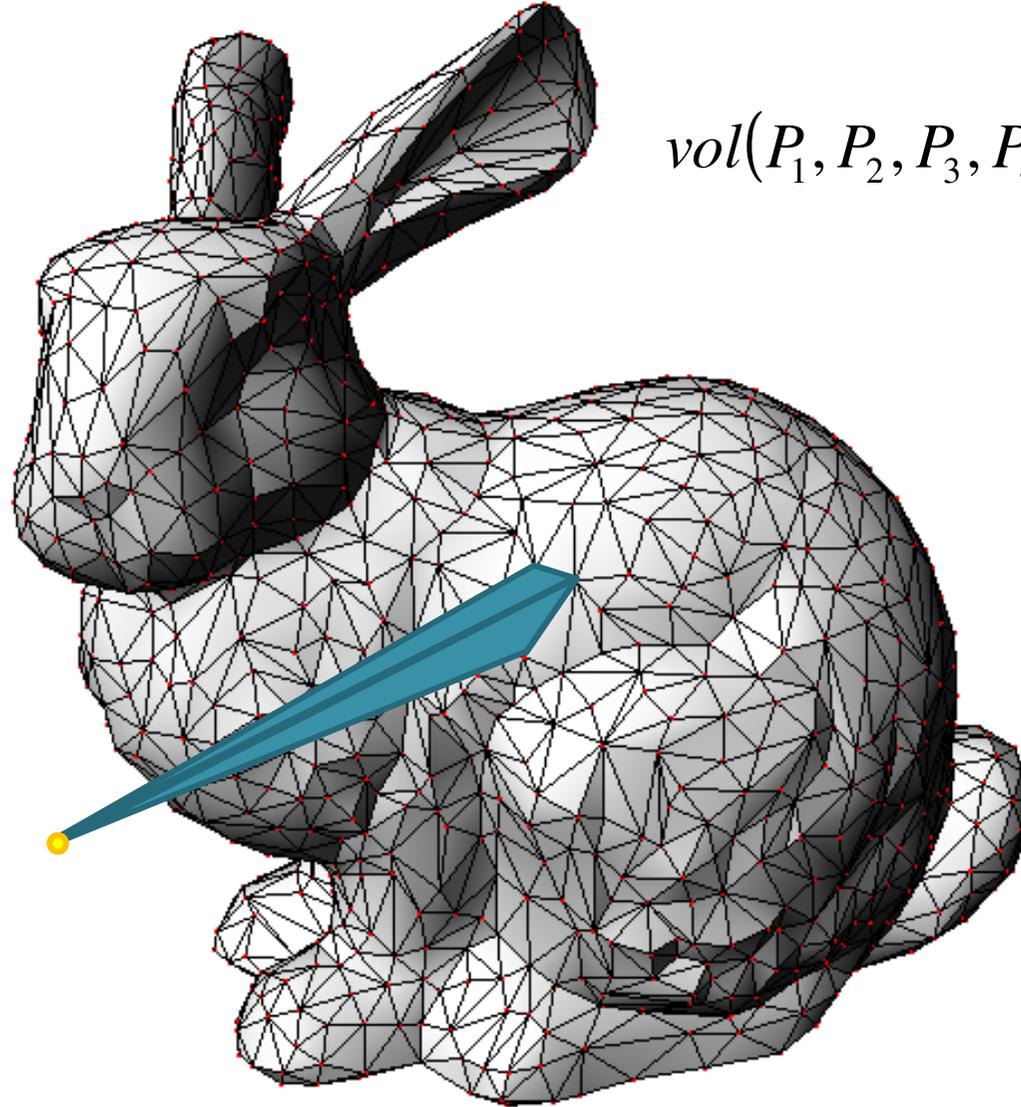
Point dans polyèdre (3)



Point dans polyèdre (4)

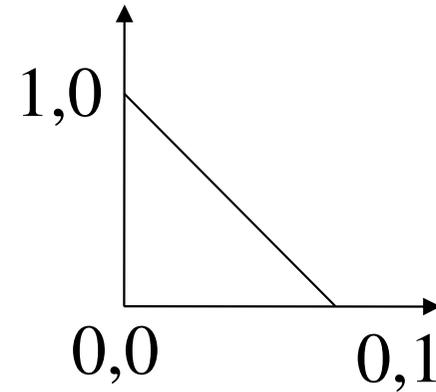
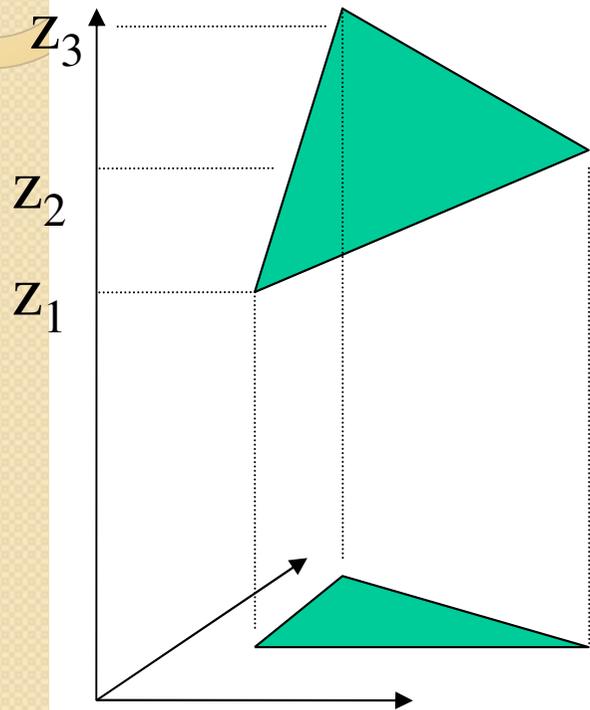


Volume d'une triangulation ?



$$\text{vol}(P_1, P_2, P_3, P_4) = \frac{1}{6} (\vec{W} \cdot (\vec{U} \wedge \vec{V}))$$

Calcul du volume



$$x = x(u, v)$$

$$y = y(u, v)$$

(u, v) dans $[0:1] \times [0:1]$

et $u+v \leq 1$

$$\iint_{xy} z(x, y) dx dy = \iint_{u=0, v=1-u} f(x(u, v), y(u, v)) \text{Det}(J) du dv$$

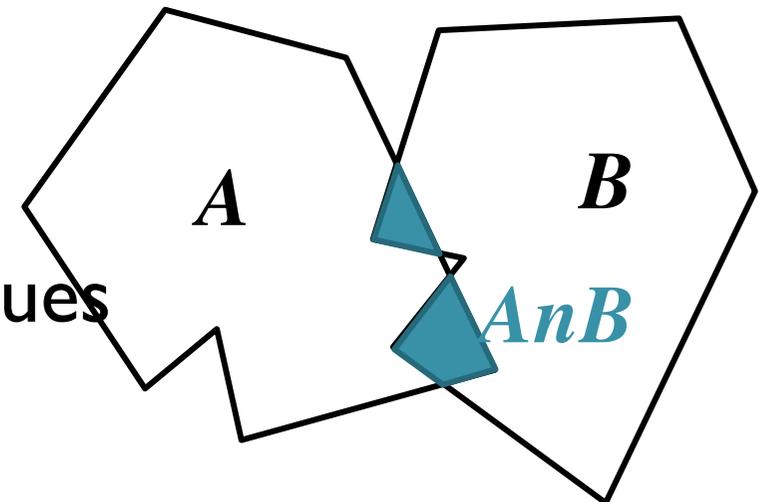
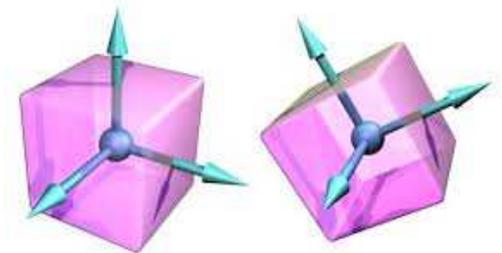
$$J = \begin{vmatrix} dx/du & dy/du \\ dx/dv & dy/dv \end{vmatrix}$$

4. Modification du modèle

- Niveau élémentaire :
 - Opérateurs d'Euler

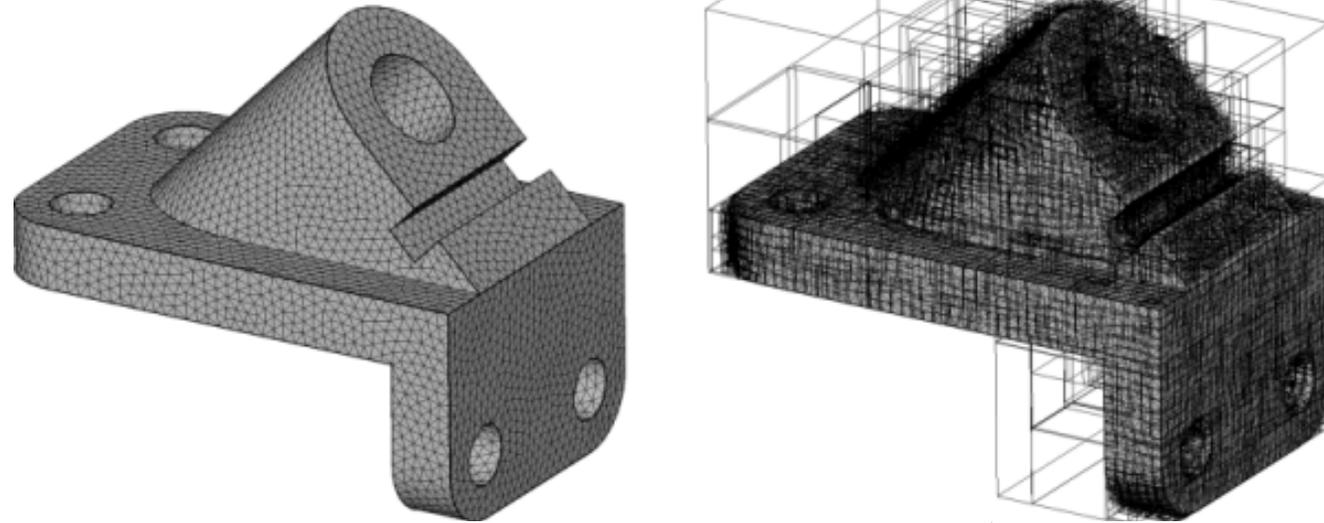
$$f - a + s = 2cc$$

- Niveau supérieur :
 - Dilatation/érosion :
 - Somme de Minkowski
 - Opérations booléennes
 - Transformations géométriques



Détection/Calculer/Eviter des intersections entre polyèdres

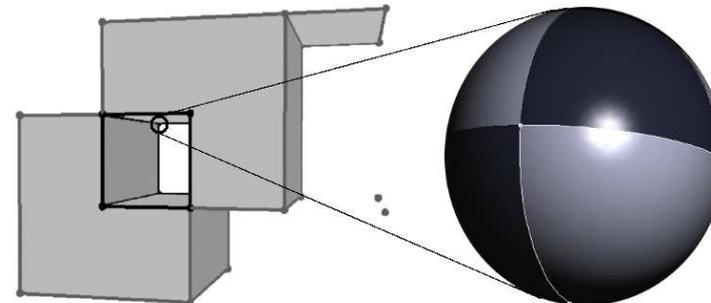
Hierarchical AABB-Tree in CGAL



Minkowski 3D in CGAL

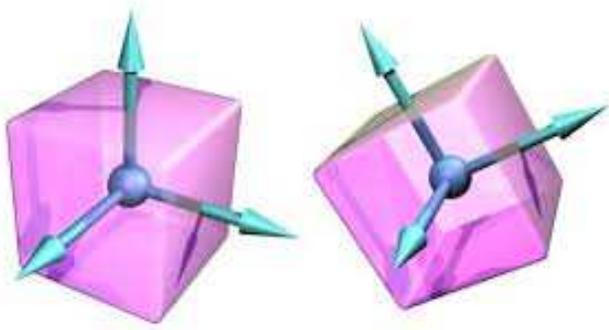


Boolean operations in CGAL
on polyhedrons and
Nef-polyhedrons



4. Transformations



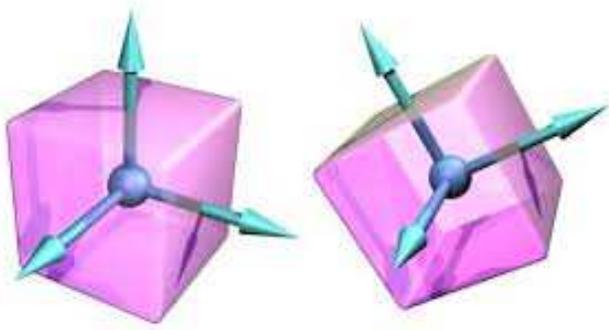


Rotations d'axes x, y, z

$$R_z(\alpha) = \quad ?$$

$$R_x(\theta) = \quad ?$$

$$R_y(\psi) = \quad ?$$

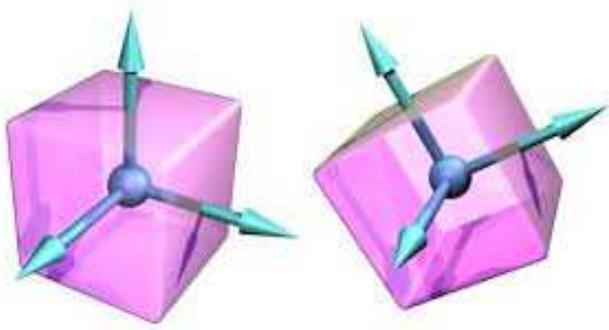


Rotations d'axes x, y, z

$$R_z(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_x(\theta) = \quad ?$$

$$R_y(\psi) = \quad ?$$

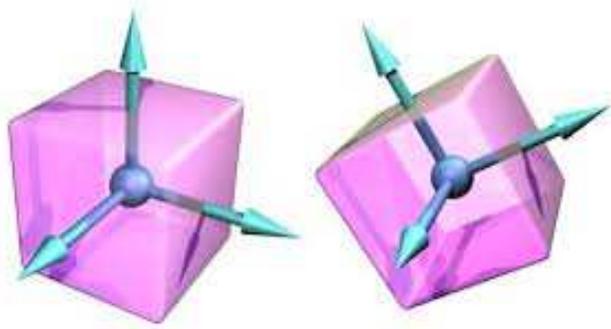


Rotations d'axes x, y, z

$$R_z(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix}$$

$$R_y(\psi) = \begin{pmatrix} \cos(\psi) & 0 & \sin(\psi) \\ 0 & 1 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) \end{pmatrix}$$



Rotations d'axes x, y, z

$$R_z(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix}$$

$$R_y(\psi) = \begin{pmatrix} \cos(\psi) & 0 & \sin(\psi) \\ 0 & 1 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) \end{pmatrix}$$

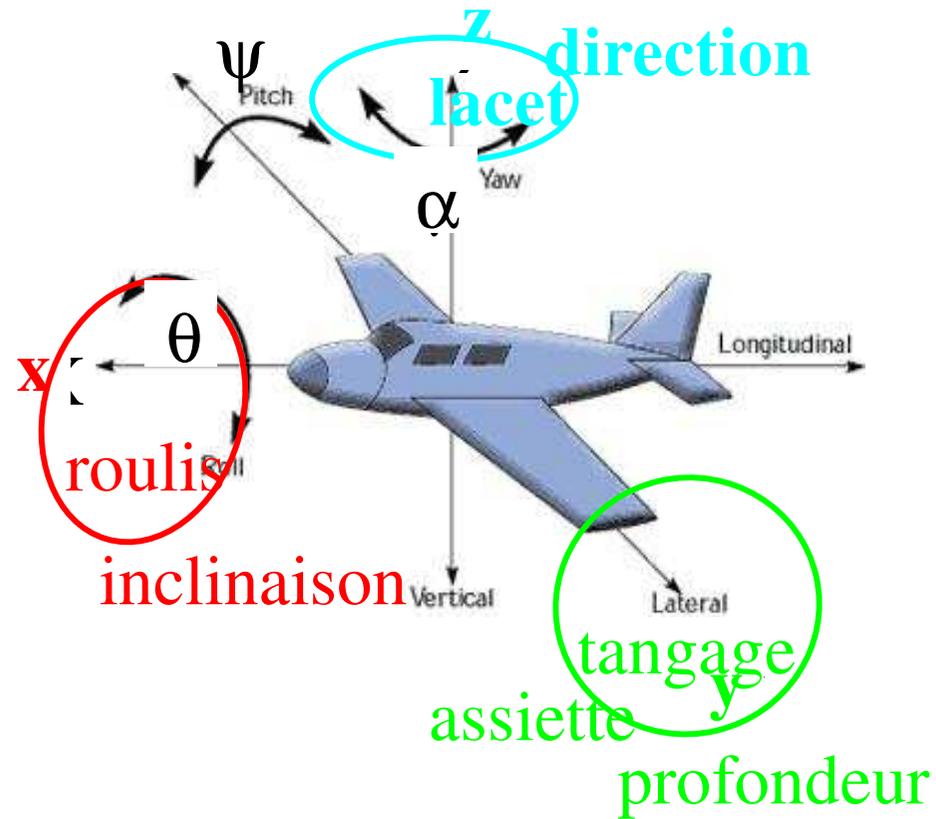
Composition des rotations = Multiplication des matrices

Elle n'est pas commutative → choisir un ORDRE

La rotation suivant x puis y puis z dans **le repère global**
(l'ordre est important) correspond à la composition :

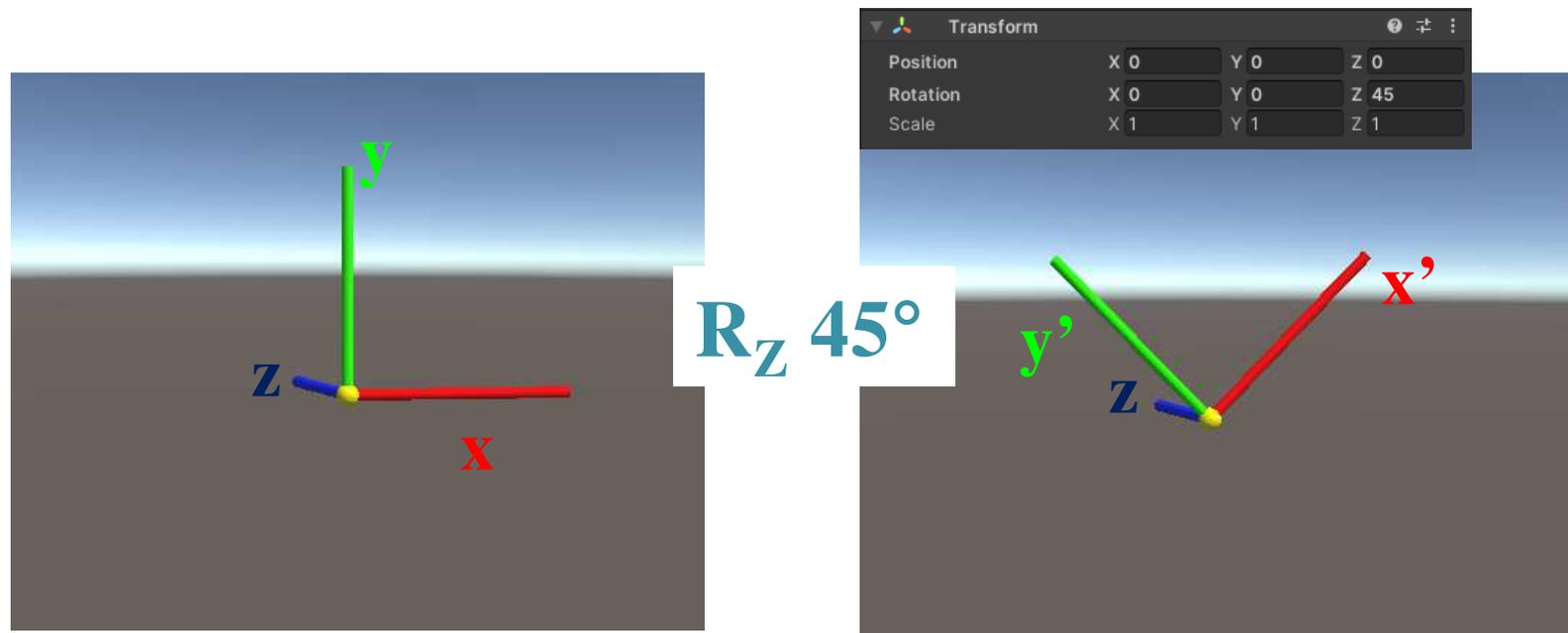
$$R_z(\alpha)R_y(\psi)R_x(\theta)$$

Les angles d'Euler (le repère local)



Les angles d'Euler (le repère local)

La rotation autour d'un axe fait tourner les 2 autres → ordre ?



La rotation suivant x puis y' puis z'' dans le repère local correspond à la composition :

$$R_{z''}(\alpha)R_{y'}(\psi)R_x(\theta) = R_x(\theta)R_y(\psi)R_z(\alpha)$$

C'est-à-dire la rotation suivant z puis y puis x dans le repère global : l'ordre inverse

« Démonstration »

Exemple :

- Dans le repère local : $R_y(\beta) \circ R_x(\alpha) P(x,y,z)$
- $R_y(\beta) = ?$

$$R_x(\alpha) \circ R_y(\beta) \circ R_x(-\alpha) P'(x,y,z)$$

On ramène le résultat
dans le repère global

On applique la
transformation

On ramène le point dans le
repère local de l'objet

$$D'où : R_y(\beta) \circ R_x(\alpha) P(x,y,z) = R_x(\alpha) \circ R_y(\beta) \circ R_x(-\alpha) \circ R_x(\alpha) P(x,y,z)$$

Les rotations dans le système d'Euler (local) sont équivalente aux rotations par rapport à des axes fixes mais dans l'ordre inverse :

$$R_z(\gamma) \circ R_y(\beta) \circ R_x(\alpha) = R_x(\alpha) \circ R_y(\beta) \circ R_z(\gamma)$$

Rotation autour d'un axe N

Données : P et \vec{N}

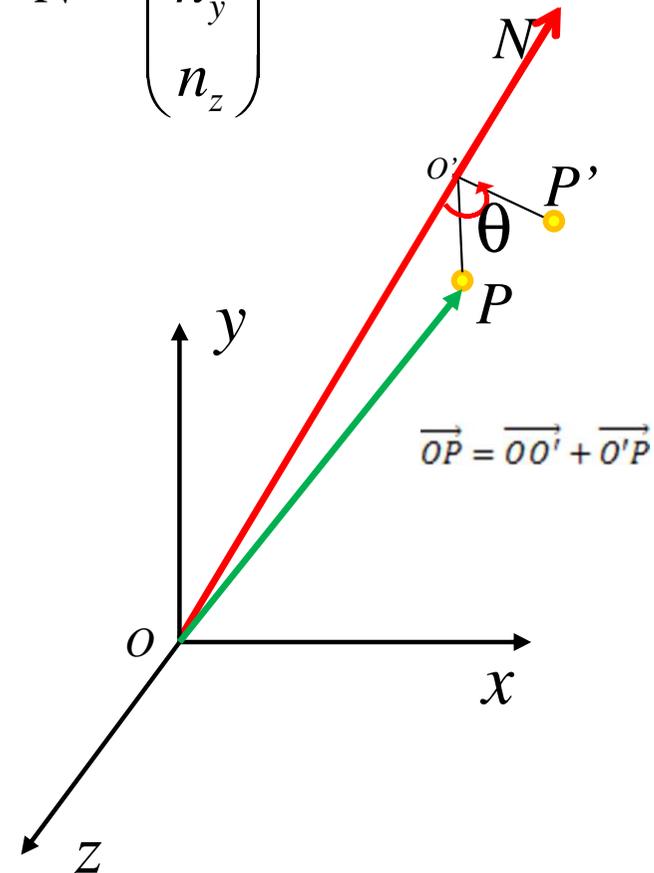
Principe :

- $\vec{OP} = \vec{OO'} + \vec{O'P}$
- $\vec{OO'}$ est invariant par la rotation,
- la rotation est dans le plan(O', P, P')

Étapes du calcul

1. $O' =$ Projection de P sur l'axe \vec{N} ,
2. Repère local plan(O', P, P'):
 - $\vec{i} \equiv \vec{O'P}$
 - $\vec{j} \equiv \vec{N} \wedge \vec{i}$
3. $\vec{O'P'} = \cos\theta \vec{i} + \sin\theta \vec{j}$

$$N = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}$$



➔ Passage à la représentation matricielle

Rotation autour d'un axe

$$A_N = \begin{pmatrix} n_x^2 & n_x \cdot n_y & n_x \cdot n_z \\ n_y \cdot n_x & n_y^2 & n_y \cdot n_z \\ n_z \cdot n_x & n_z \cdot n_y & n_z^2 \end{pmatrix}$$

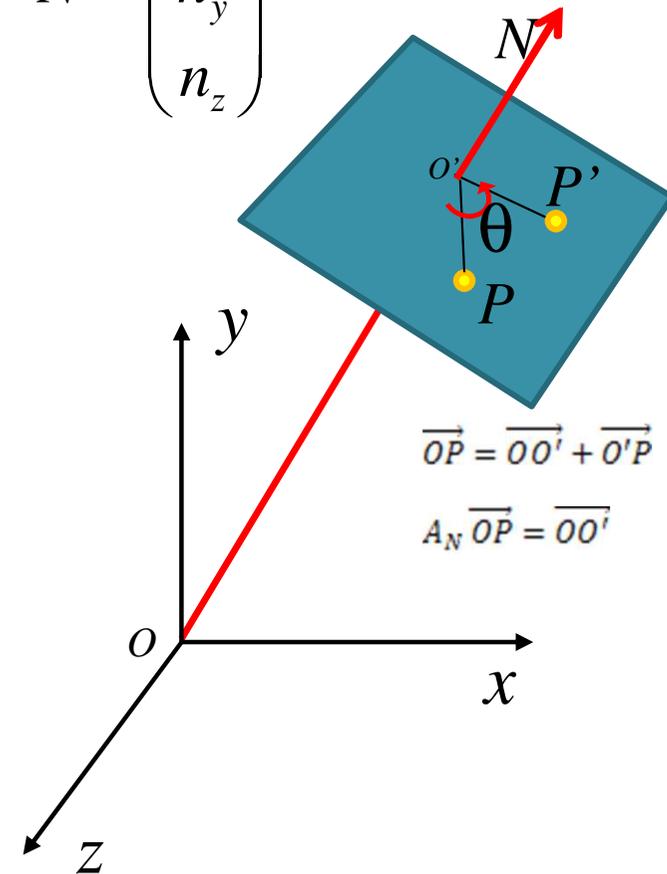
$$B_N = \begin{pmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{pmatrix}$$

$$N = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}$$

$$B_N \vec{V} = \vec{N} \wedge \vec{V}$$

$$A_N \vec{V} = (\vec{N} \cdot \vec{V}) \vec{N}$$

$$I - A_N = -B_N^2$$



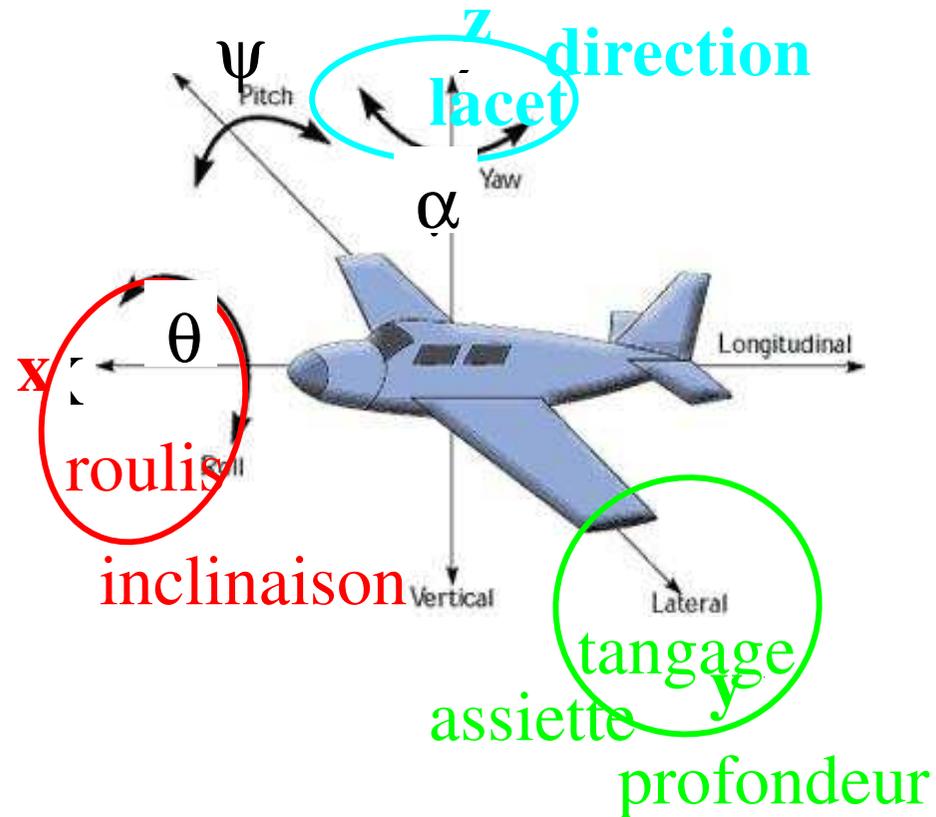
$$M = A_N + \cos(\theta) \cdot (I - A_N) + \sin(\theta) \cdot B_N$$

$$P' = M \cdot P$$

A_N est la matrice de projection sur l'axe N ,

$(I - A_N)$ est la matrice de projection sur la plan orthogonal à N

Les angles d'Euler et le blocage de cardan



Orientation : angles d'Euler et gyroscope



Hiérarchie:

Rotation suivant Z

Rotation suivant Y

Rotation suivant X

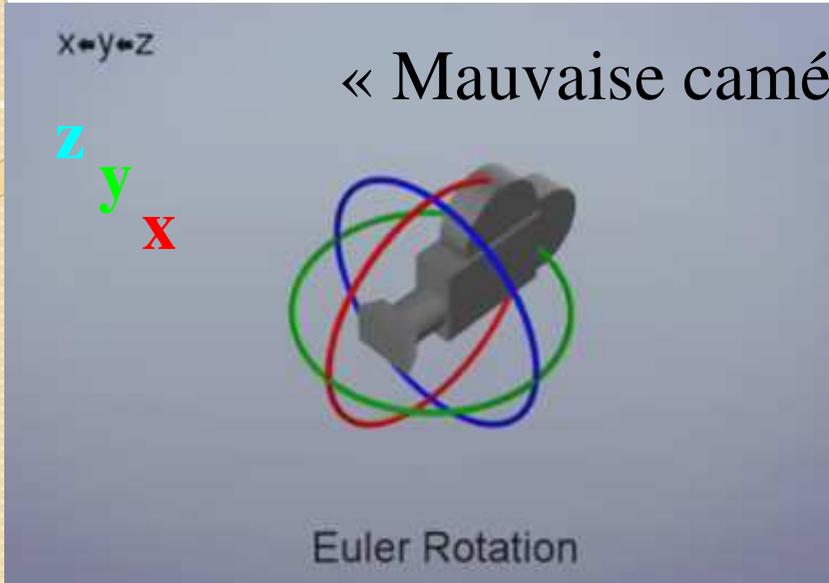
R_z puis R_y , puis R_x ,

R_x , (R_y , (R_z (A)))

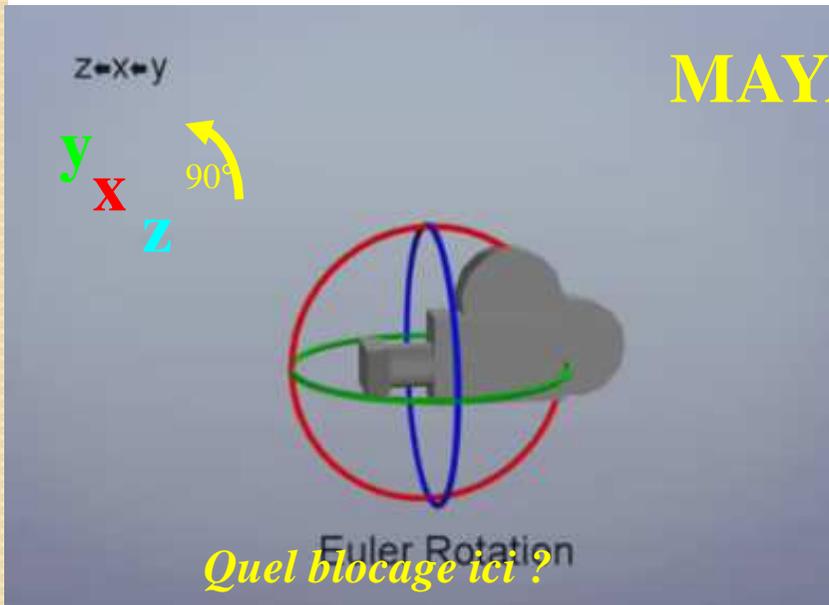
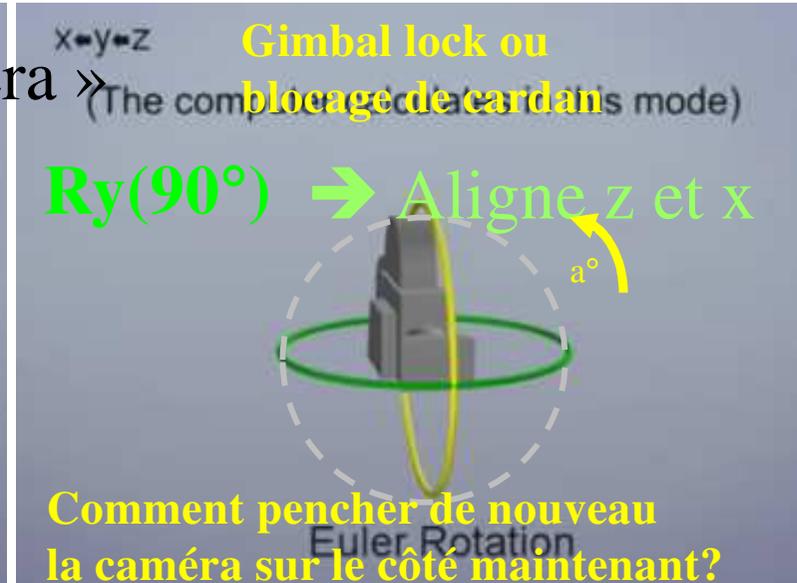
Gimbal lock (blocage de cardan)



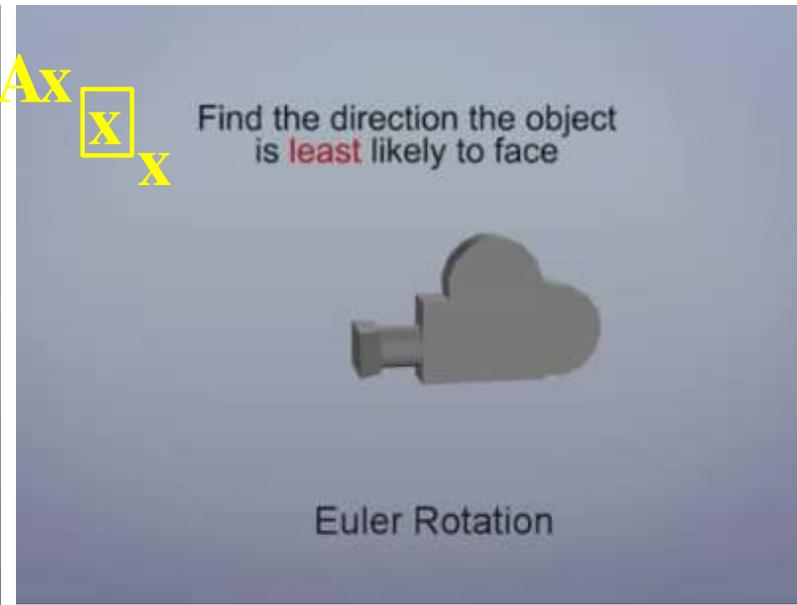
Animation d'une caméra



« Mauvaise caméra »

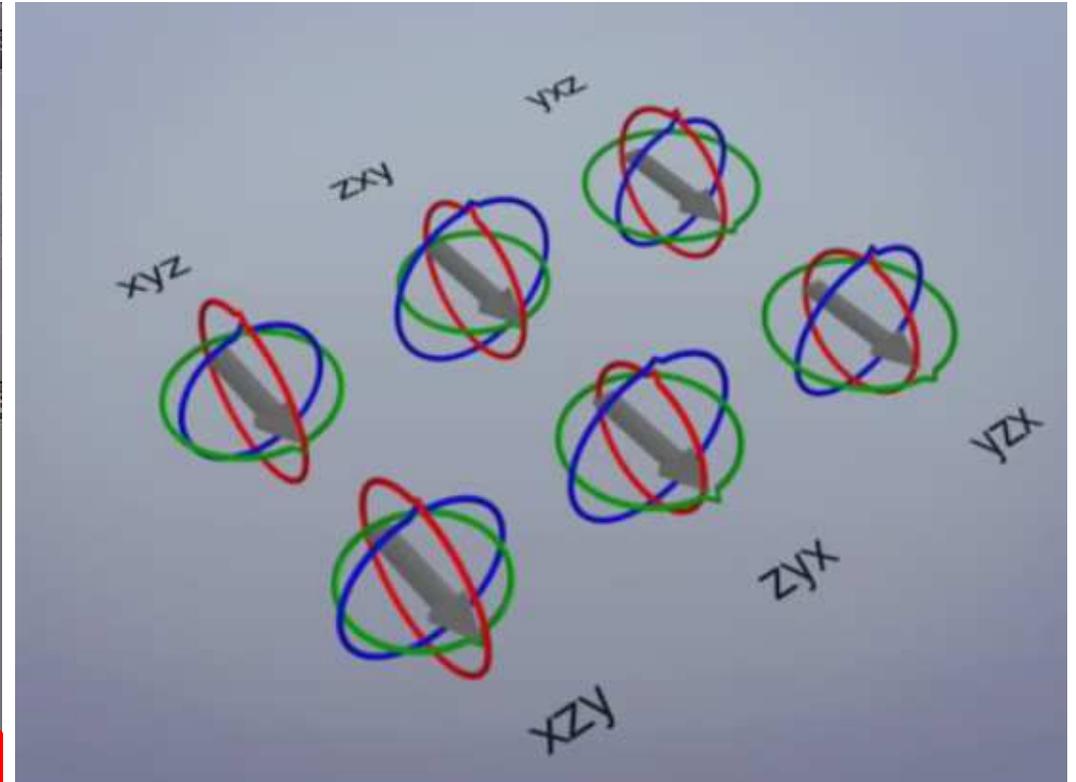


MAYAx



Modes de rotation dans Blender

- Axis Angle
- XYZ Euler
- XZY Euler
- ...

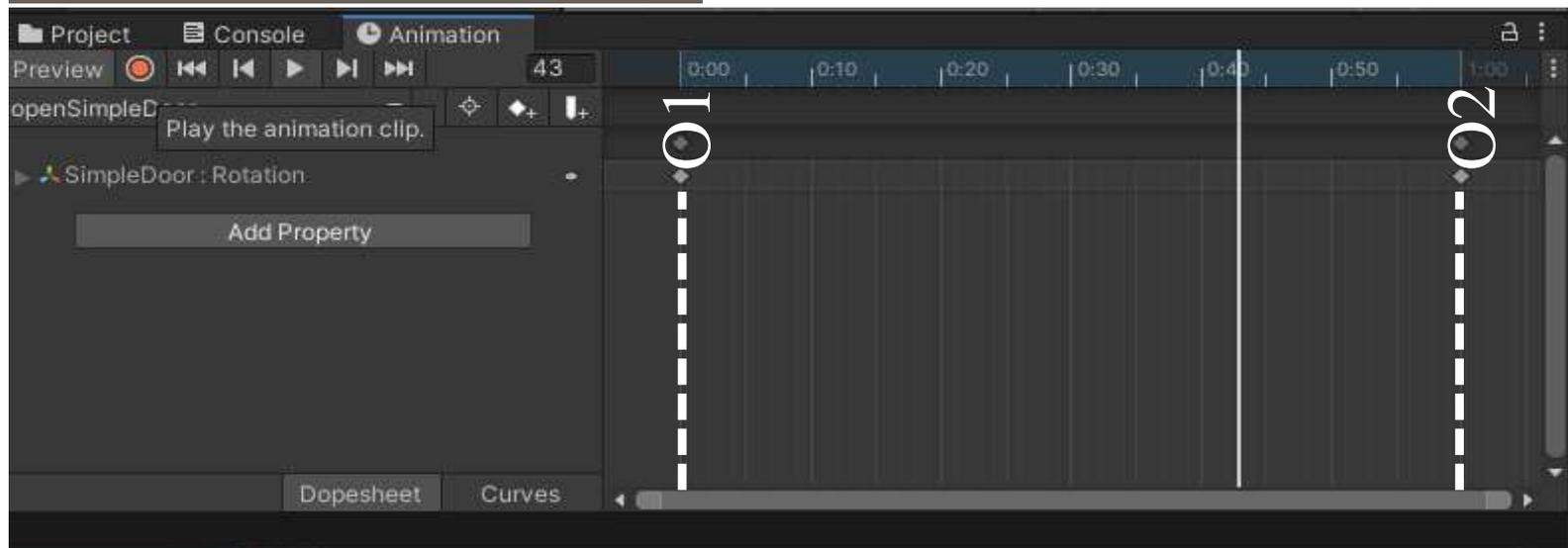
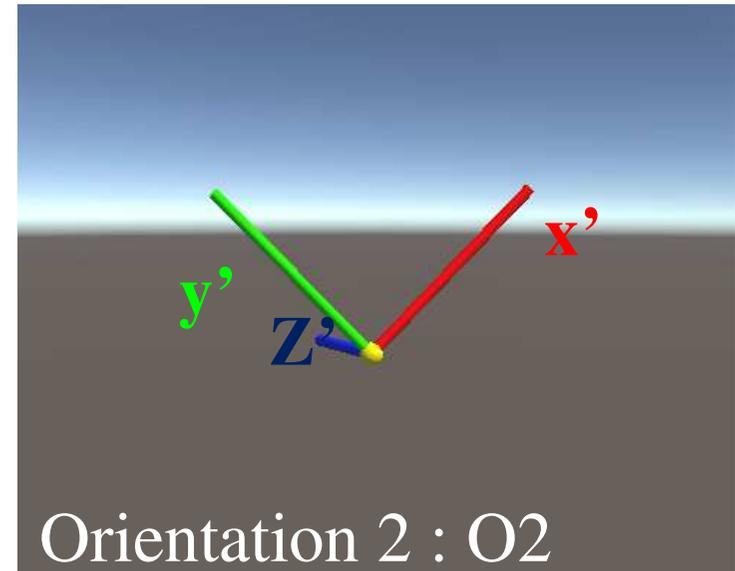
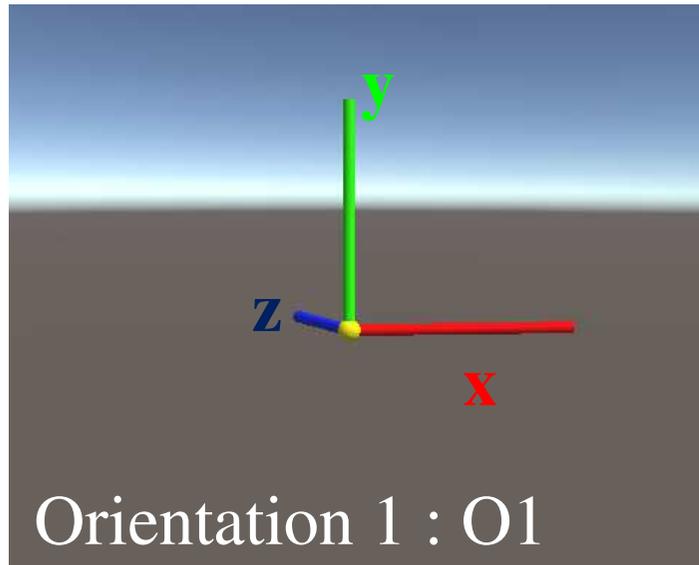


- Quaternions

Animation : les « keyframes » doivent toutes être définies dans le même mode de rotation

Keyframing

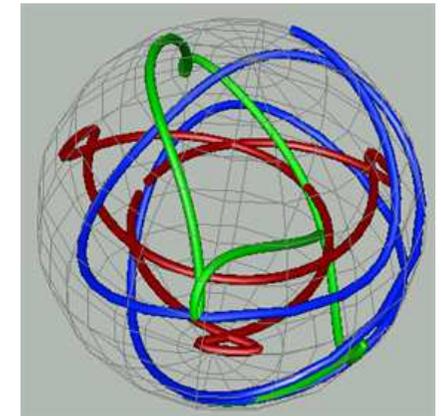
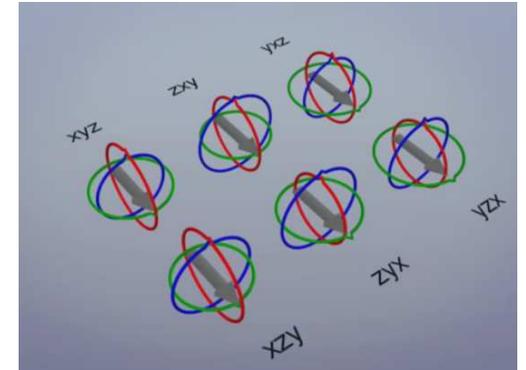
Interpolation entre 2 orientations → rotations ?



Pour aller plus loin

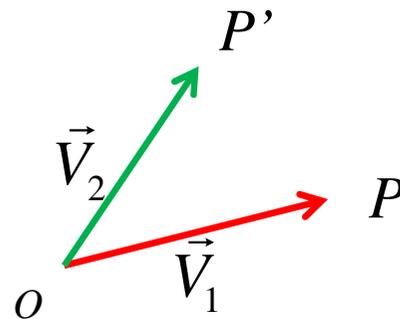
- Euler & le blocage de cardan (Gimbal lock)
- Quaternions et interpolation sphérique (SLERP)
$$\text{Quaternion.Slerp}(Q_1, Q_2, t)$$

avec t dans $[0,1]$ **
- Rotation autour d'un axe quelconque (SKIP)



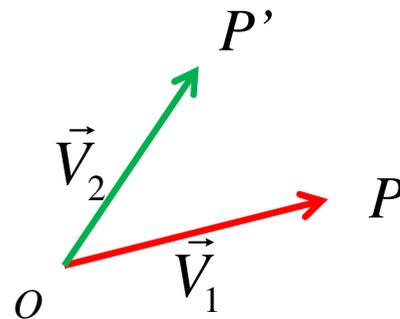
Exercice (2D)

- Ecrire la matrice de rotation pour passer du vecteur V_1 au vecteur V_2



Exercice (2D)

- Ecrire la matrice de rotation pour passer du vecteur V_1 au vecteur V_2



$$\vec{OP'} = R_z(\alpha) * \vec{OP}$$

$$R_z(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

$$\vec{V}_1 \wedge \vec{V}_2 = W : \text{axe de rotation}$$

$$\|\vec{V}_1 \wedge \vec{V}_2\| = \sin(\alpha) * \|\vec{V}_1\| * \|\vec{V}_2\|$$

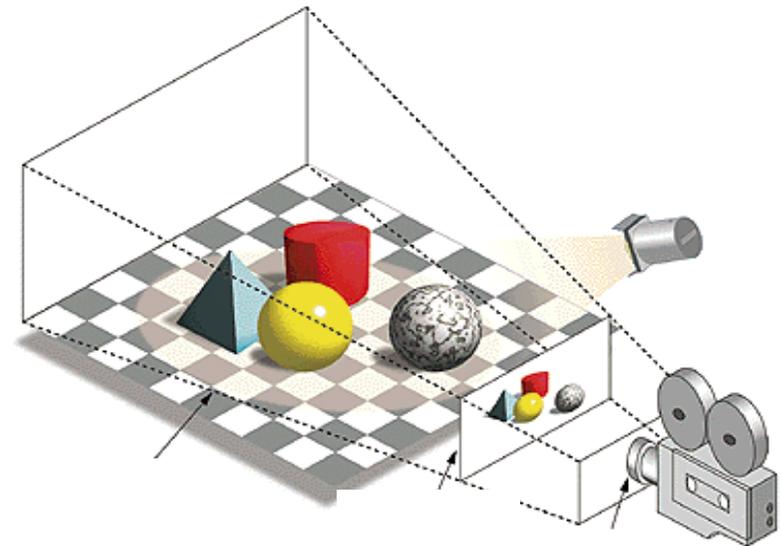
$$\alpha = \arcsin(\|\vec{V}_1 \wedge \vec{V}_2\| / (\|\vec{V}_1\| * \|\vec{V}_2\|))$$

Et la translation ?

Géométrie projective et coordonnées homogènes

Le calcul doit être rapide :

- Projection (œil) = plan de la caméra
- La composition dépend du type de transformation ☹



Coordonnées Homogènes (Rappel)

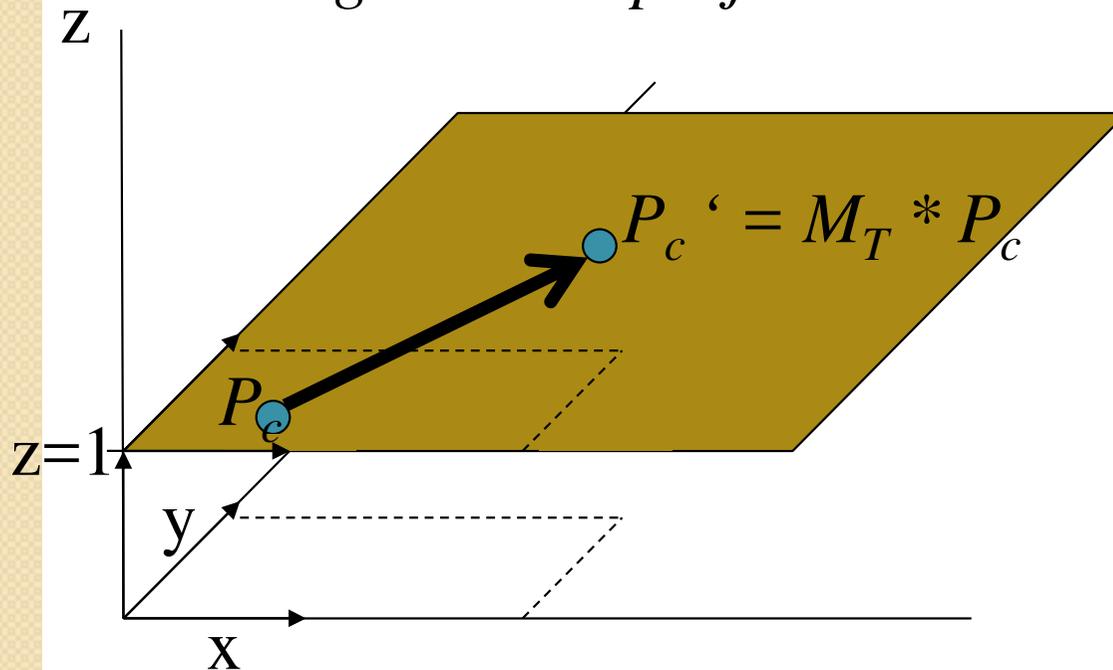
Matrice de transformation dans \mathbb{R}^{d+1}

$d=2$

$$P_c = \begin{pmatrix} x \\ y \end{pmatrix} \longrightarrow P_c = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \longrightarrow P_c = \begin{pmatrix} x/1 \\ y/1 \end{pmatrix}$$

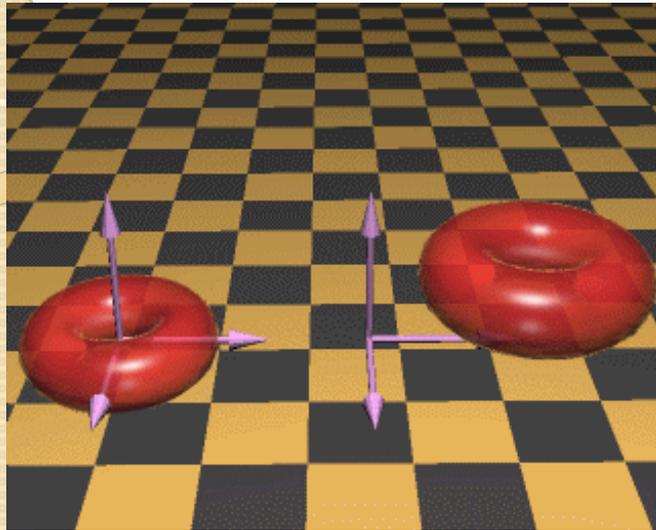
*Coordonnées Homogènes
géométrie projective*

$$R_z(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



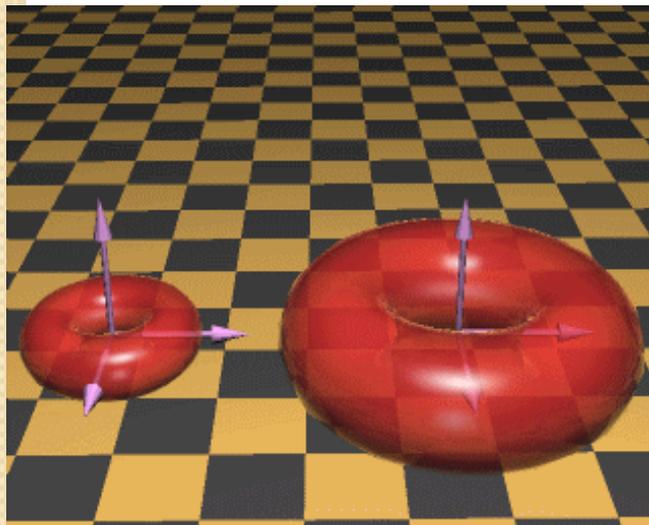
$$M_T = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Translation et mise à l'échelle (d=3)



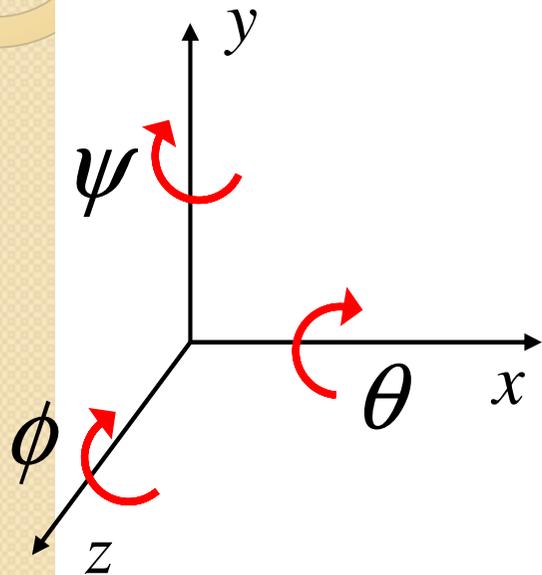
$$M_T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad P' = M_T P$$

$$M_T * (x, y, z, 1) = (x+t_x, y+t_y, z+t_z, 1) \rightarrow (x+t_x, y+t_y, z+t_z)$$



$$M_S = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad P' = M_S P$$

Rotations (d=3)



$$P' = RP = \left(R_z \cdot R_y \cdot R_x \right) P$$

autres ordres possibles
mais pas équivalents

$$R_z(\phi) = \begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 & 0 \\ \sin(\phi) & \cos(\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y(\psi) = \begin{pmatrix} \cos(\psi) & 0 & \sin(\psi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

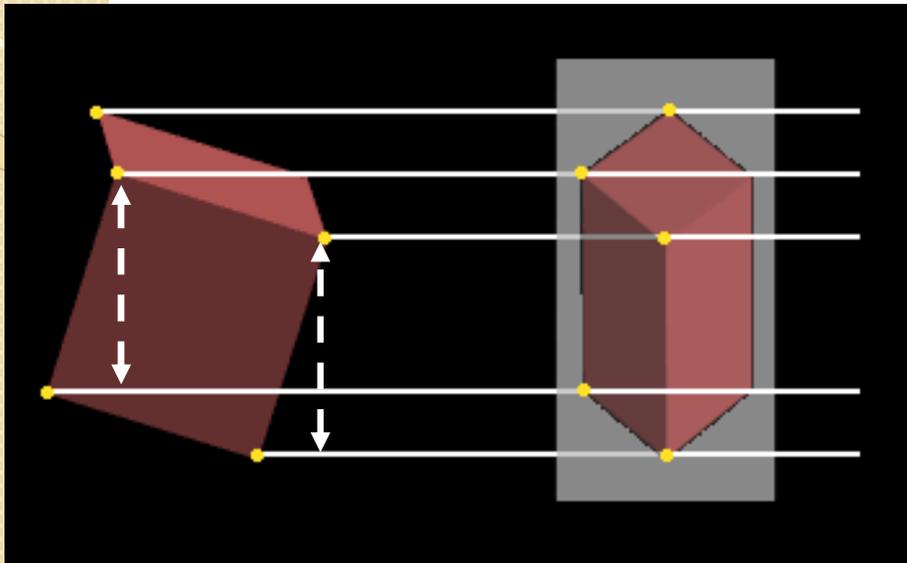
Matrice homogène de transformation

$$M_H = \begin{bmatrix} R_{3,3} & T \\ \mathbf{0}_3^t & 1 \end{bmatrix} \quad P' = M_H \cdot P$$

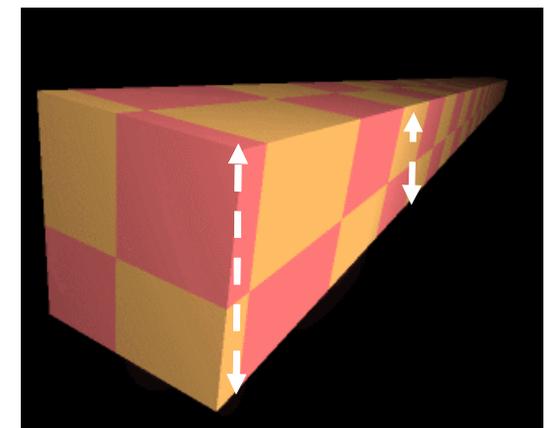
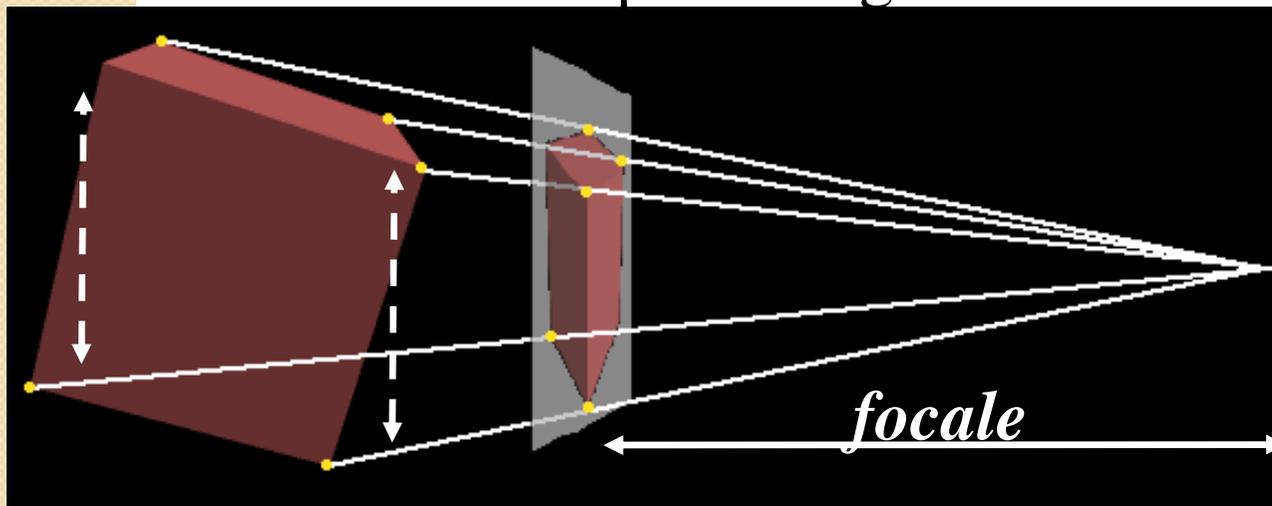
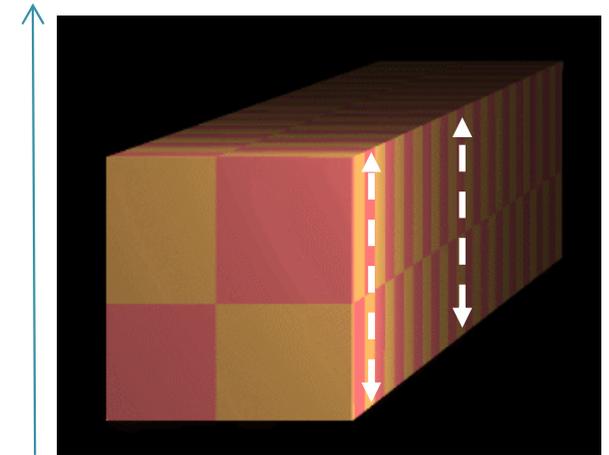
$$M_H^{-1} = \begin{bmatrix} R_{3,3}^T & -(R^T)^{-1} \cdot T \\ \mathbf{0}_3^t & 1 \end{bmatrix}$$

On retrouve les angles de rotation à partir de M_H

Projections

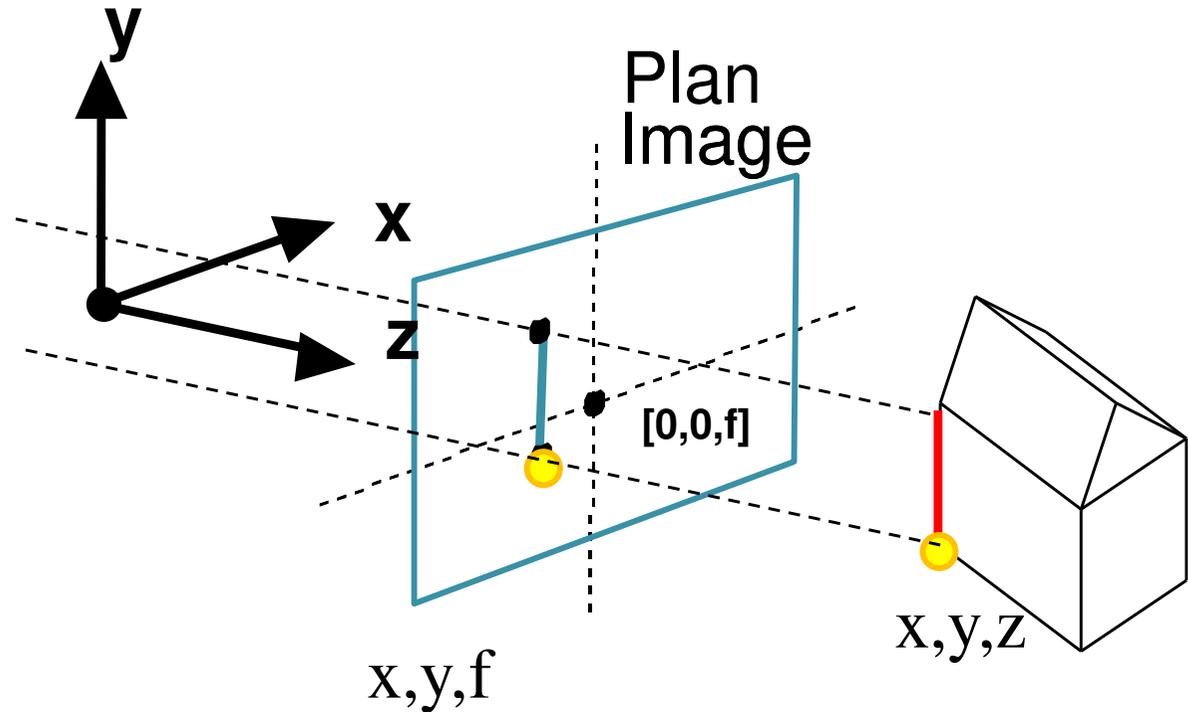


plan image



Projection orthographique

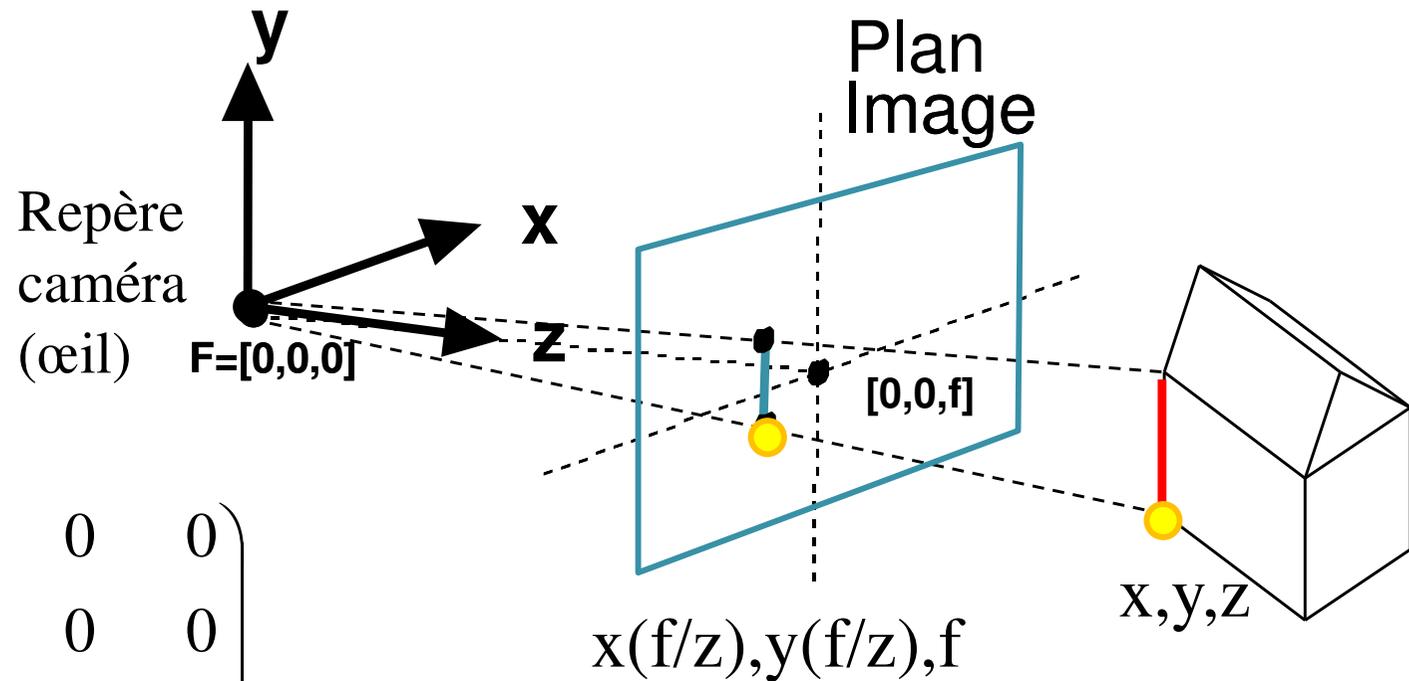
Repère
caméra
(œil)



$$M_{per} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & f \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_{per} * (x, y, z, 1) = (x, y, f, 1) \text{ qui donne en 3D : } (x, y, f)$$

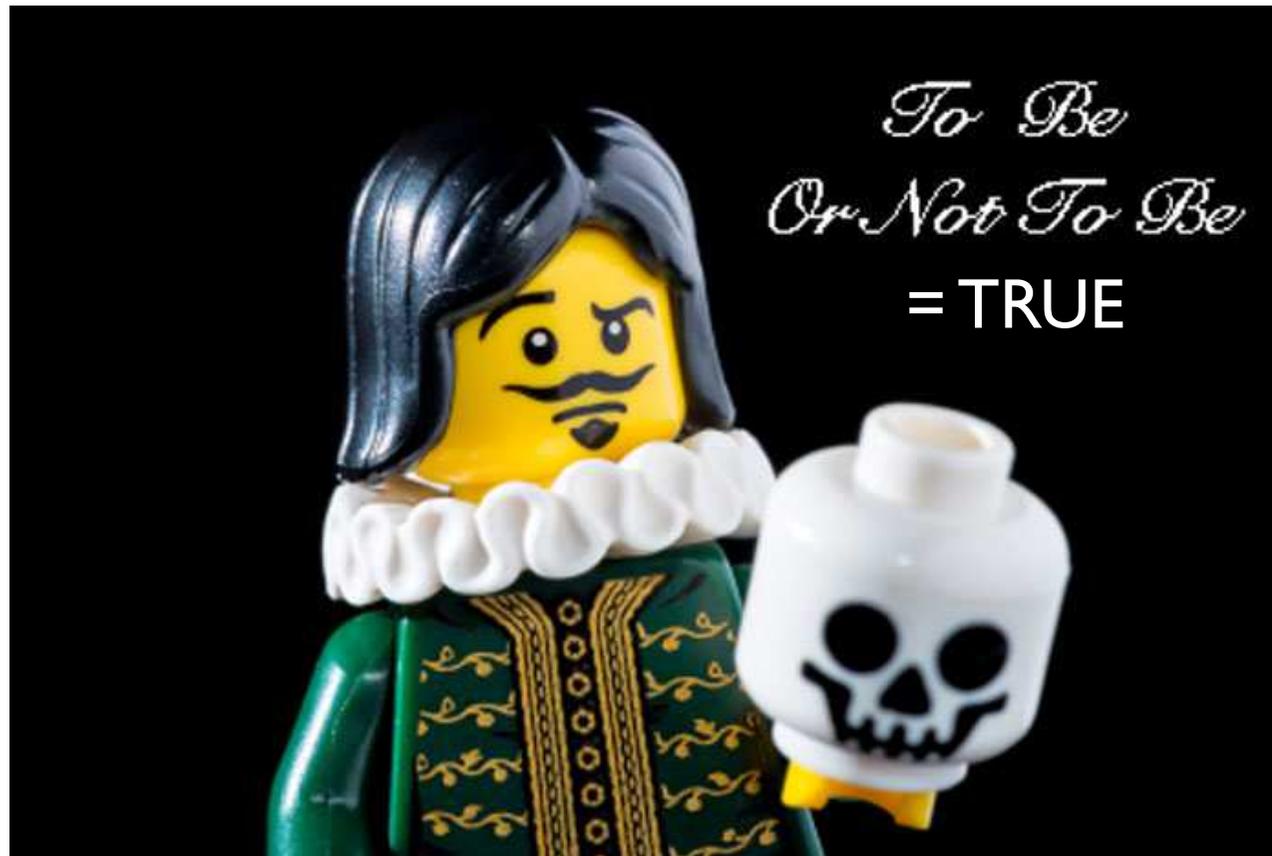
Projection perspective



$$M_{per} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix}$$

$$M_{per}^*(x,y,z,1) = (x,y,z,z/f) \text{ qui donne en 3D : } (x^*(f/z), y^*(f/z), f)$$

4.2. Opérations booléennes



[Skip](#) >

[End](#) >

Opérations booléennes

Fr(A), Fr(B)

Fr(A∩B) =

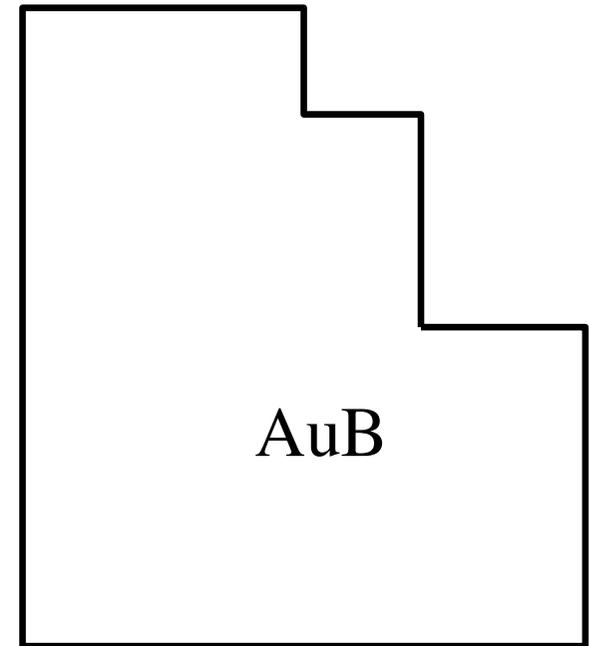
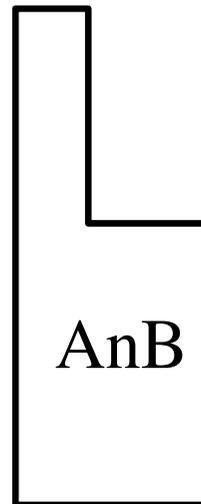
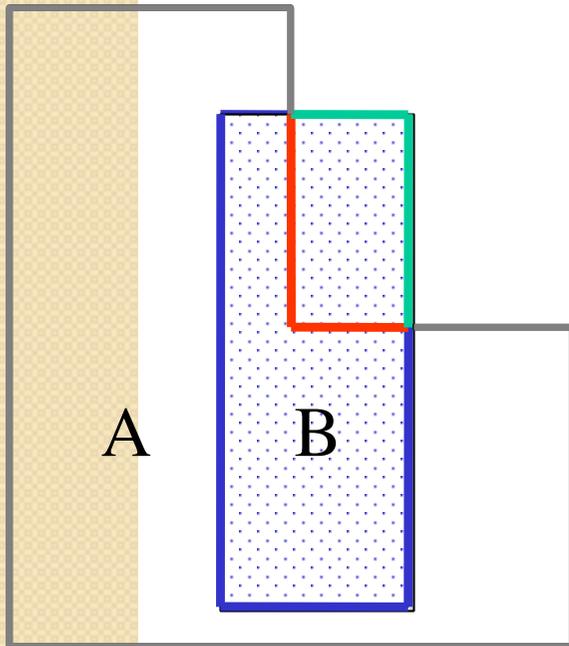
Fr(A∪B) =

Fr(B) dans A

Fr(A) hors B

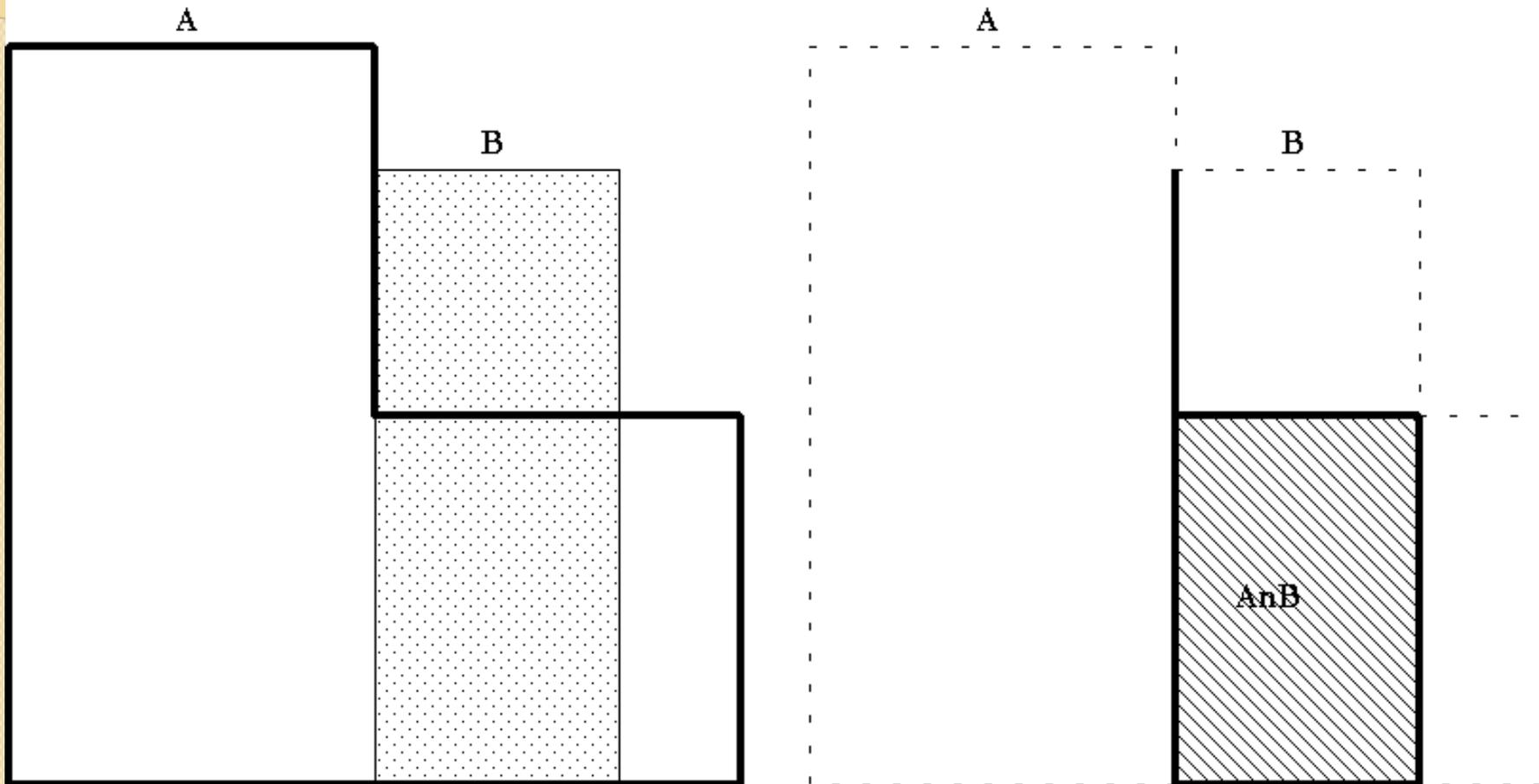
Fr(A) dans B

Fr(B) hors A

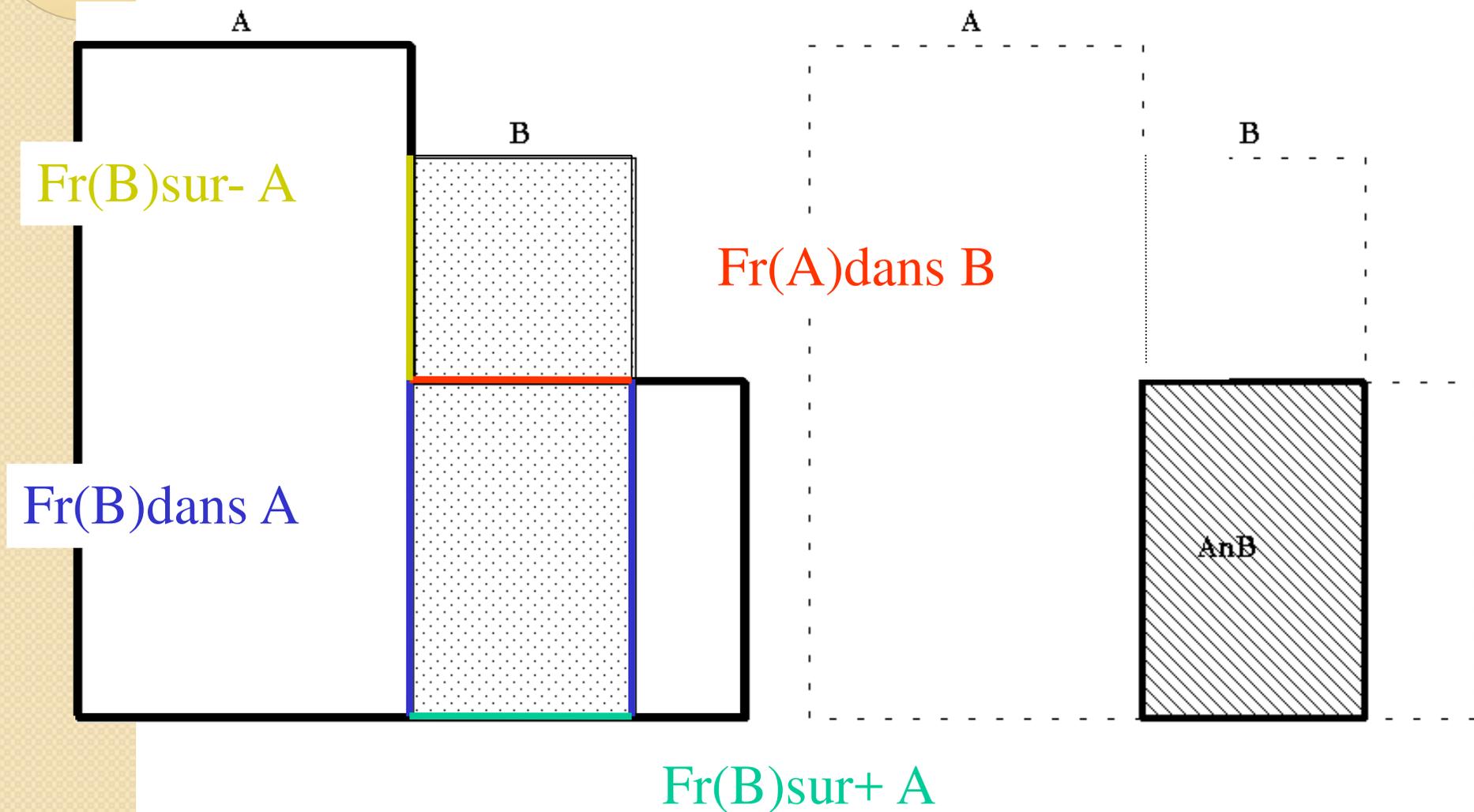


(Rappel)

Opérations booléennes régularisées



Opérations booléennes régularisées



Opérations booléennes régularisées

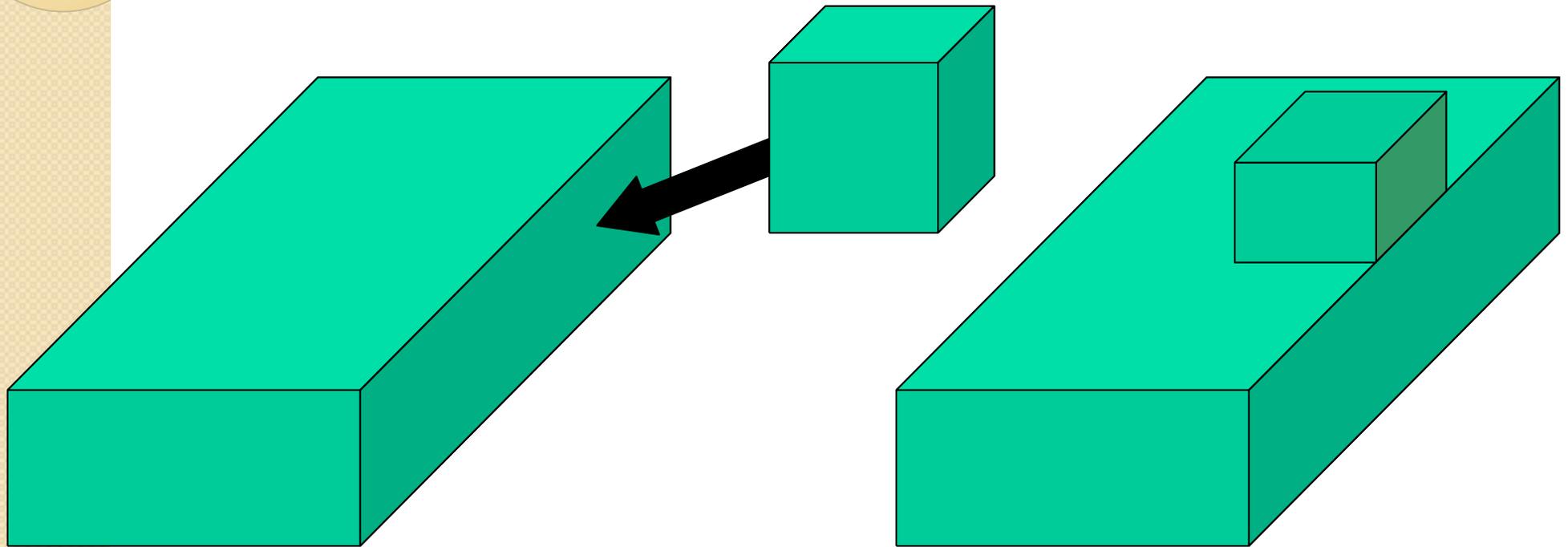
$$\text{Fr}(A \cap B) = \text{Fr}(A) \text{ dans } B + \text{Fr}(B) \text{ dans } A + \text{Fr}(B) \text{ sur } + A$$

$$\text{Fr}(A \cup B) = \text{Fr}(A) \text{ hors } B + \text{Fr}(B) \text{ hors } A + \text{Fr}(B) \text{ sur } + A$$

$$\text{Fr}(A - B) = \text{Fr}(A) \text{ hors } B + \text{Fr}(B) \text{ dans } A + \text{Fr}(B) \text{ sur } - A$$

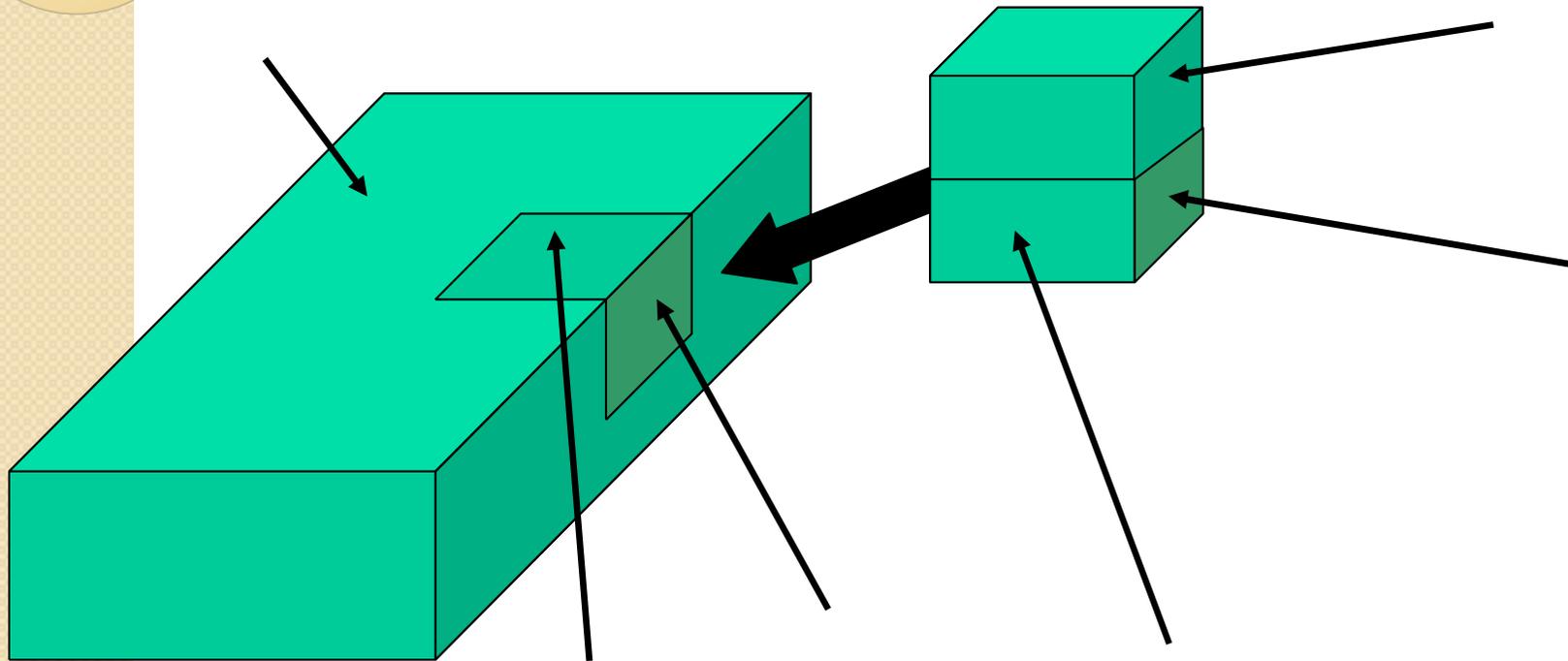
Opérations booléennes

Exemple 3D



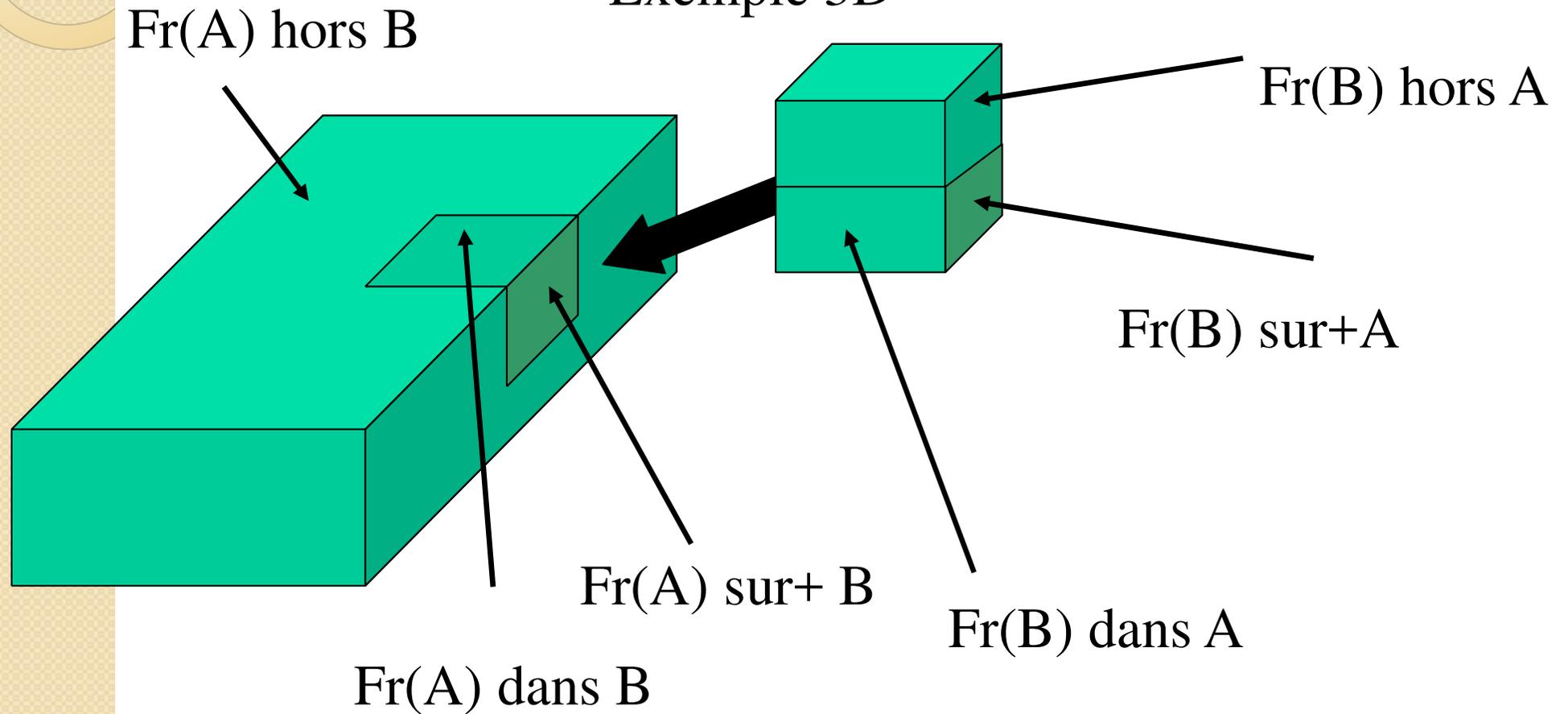
Opérations booléennes

Exemple 3D



Opérations booléennes

Exemple 3D



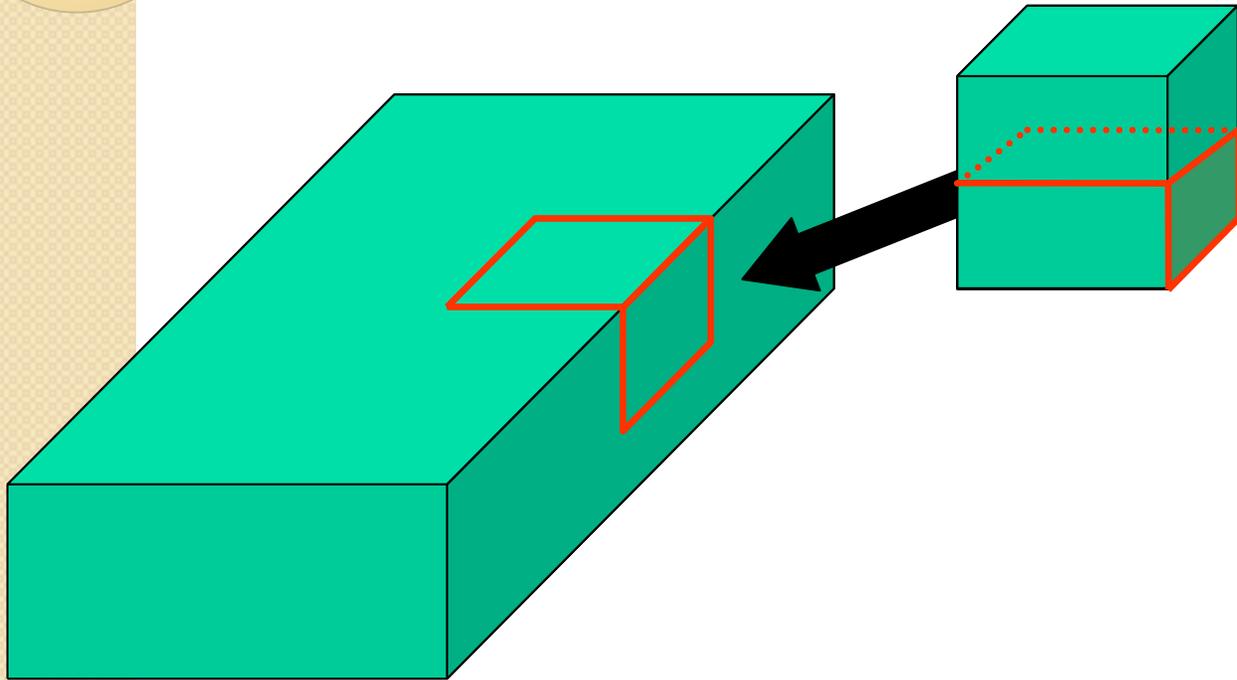
Opérations booléennes

Schéma de l'algorithme 2D et 3D

- 1. Calcul des intersections
- 2. Classification des parties de frontière
- 3. Suppression des parties inutiles
- 4. Collage des parties restantes

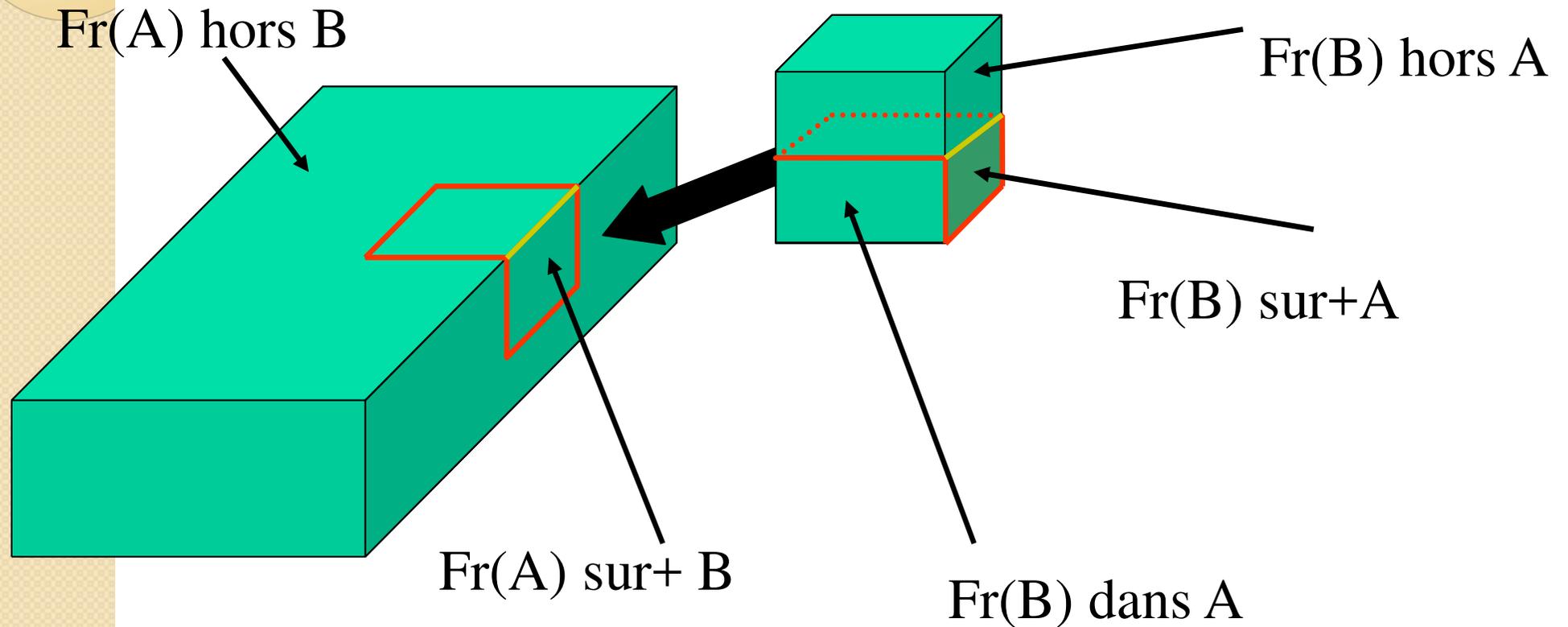
Opérations booléennes

Étape 1 : Arêtes = intersection des faces



Opérations booléennes

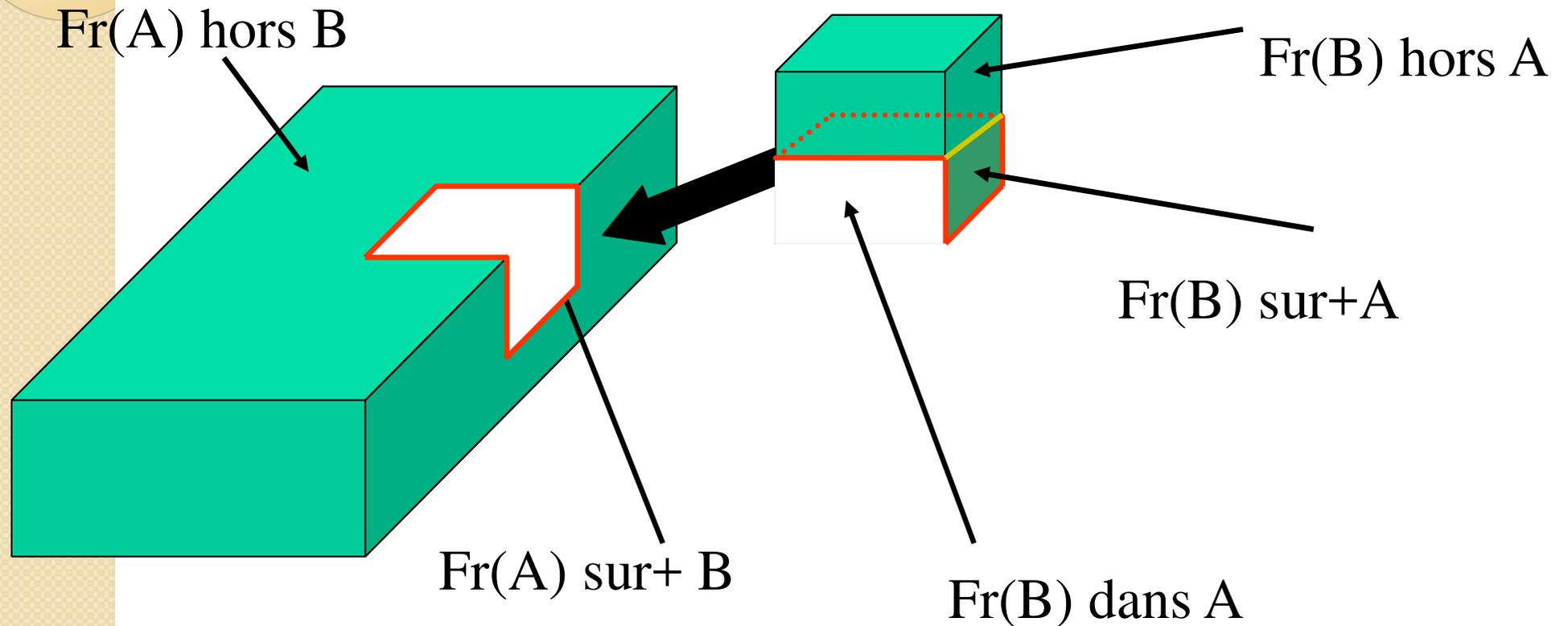
Étape 2 : Classification des portions de frontière



$$Fr(A \cup B) = Fr(A) \text{ hors } B + Fr(B) \text{ hors } A + Fr(B) \text{ sur } +A$$

Opérations booléennes

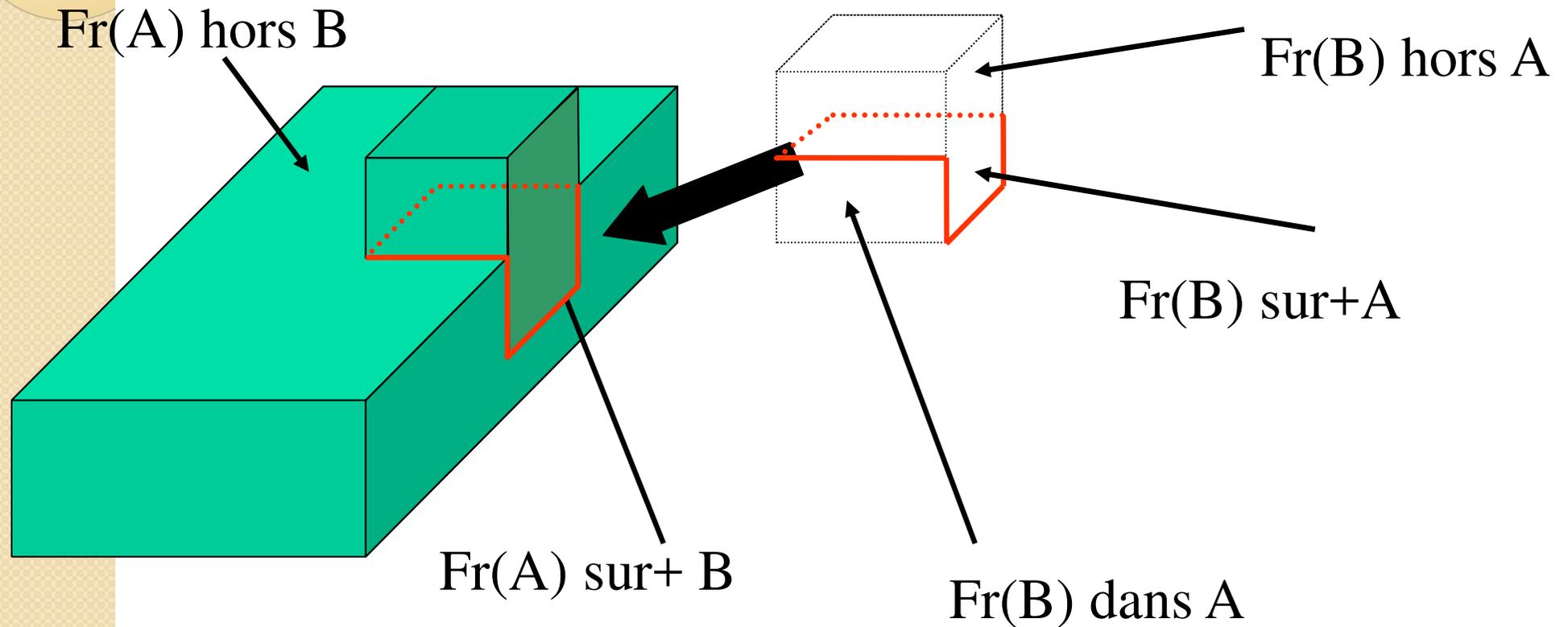
Étape 3 : Suppression des portions de frontière



$$\text{Fr}(A \cup B) = \text{Fr}(A) \text{ hors } B + \text{Fr}(B) \text{ hors } A + \text{Fr}(B) \text{ sur } +A$$

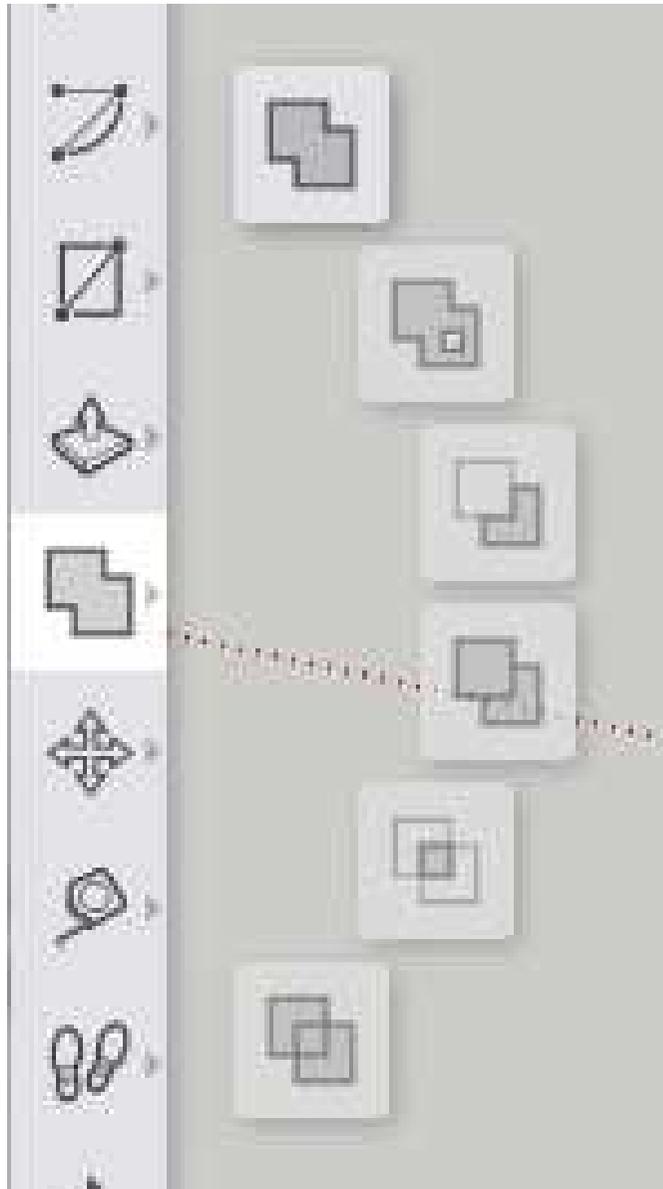
Opérations booléennes

Étape 4 : Collage des portions de frontière

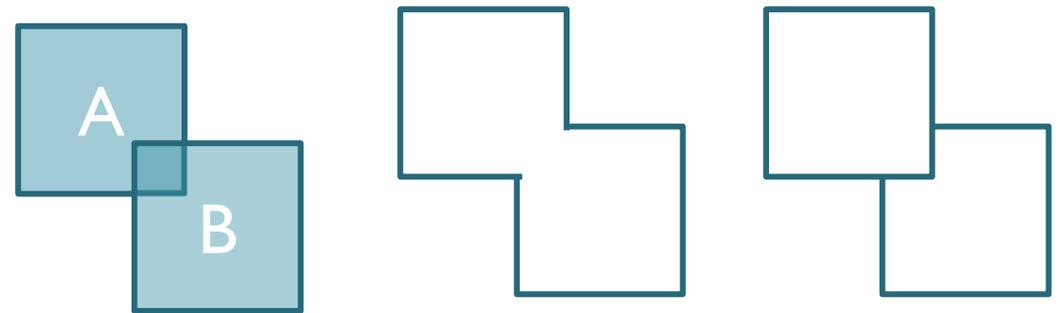


$$\text{Fr}(A \cup B) = \text{Fr}(A) \text{ hors } B + \text{Fr}(B) \text{ hors } A + \text{Fr}(B) \text{ sur } +A$$

EXERCICE : Sketchup boolean

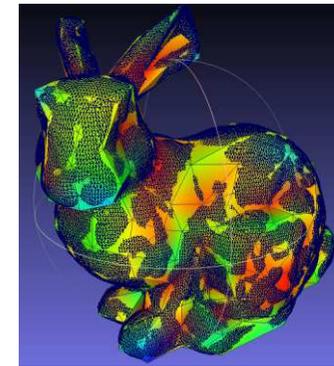
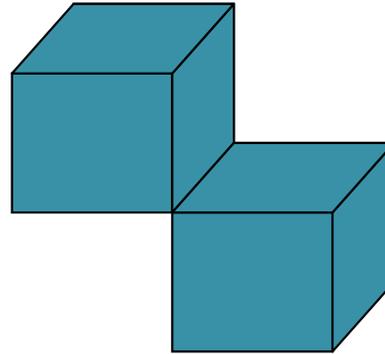


Quelles parties de frontières sont conservées ?

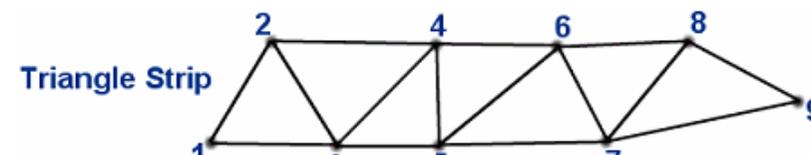


Propriétés du modèle

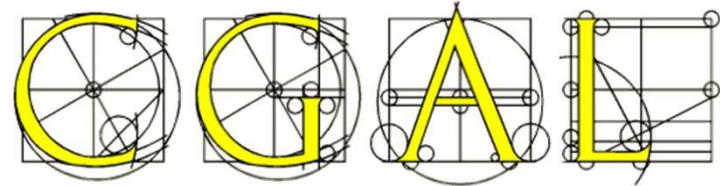
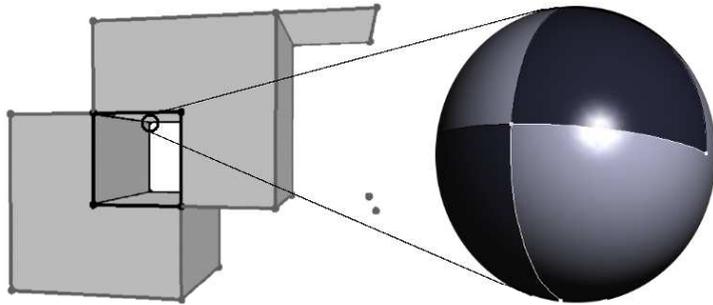
- Domaine
- Complétude
- Unicité
- Manipulation
- Performances



MAYA® ...



Exemples (libraires & toolkits)



Computational Geometry Algorithms Library

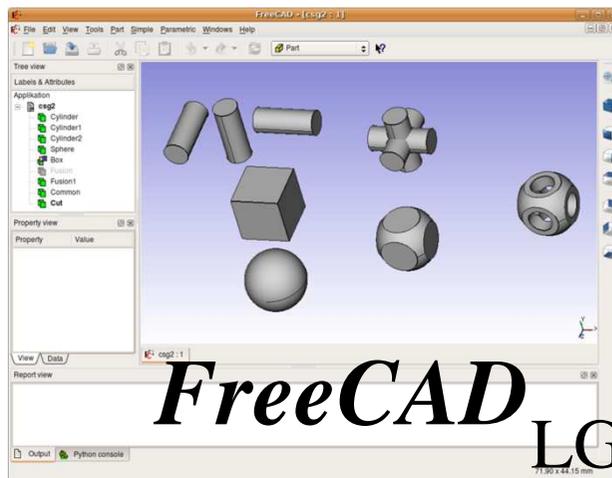
<http://www.cgal.org>

GPL



OpenMesh LGPL*

<https://www.openmesh.org/>



FreeCAD

LGPL*

<https://www.freecadweb.org/>



<http://www.opencascade.org>

LGPL*



GNU Triangulated Surfaces Library

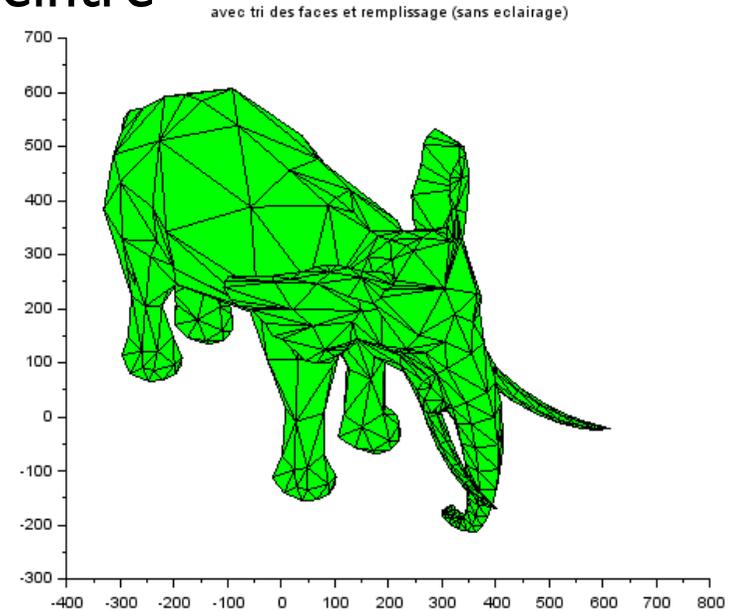
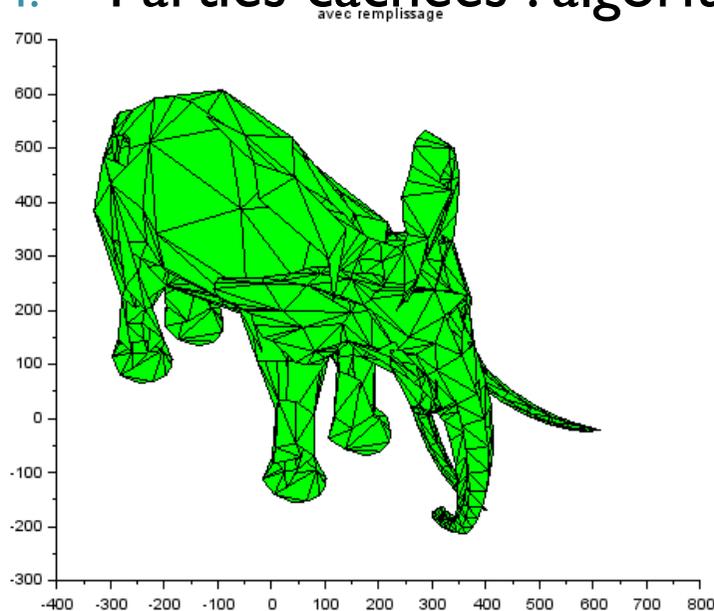
<http://gts.sourceforge.net/>

LGPL*

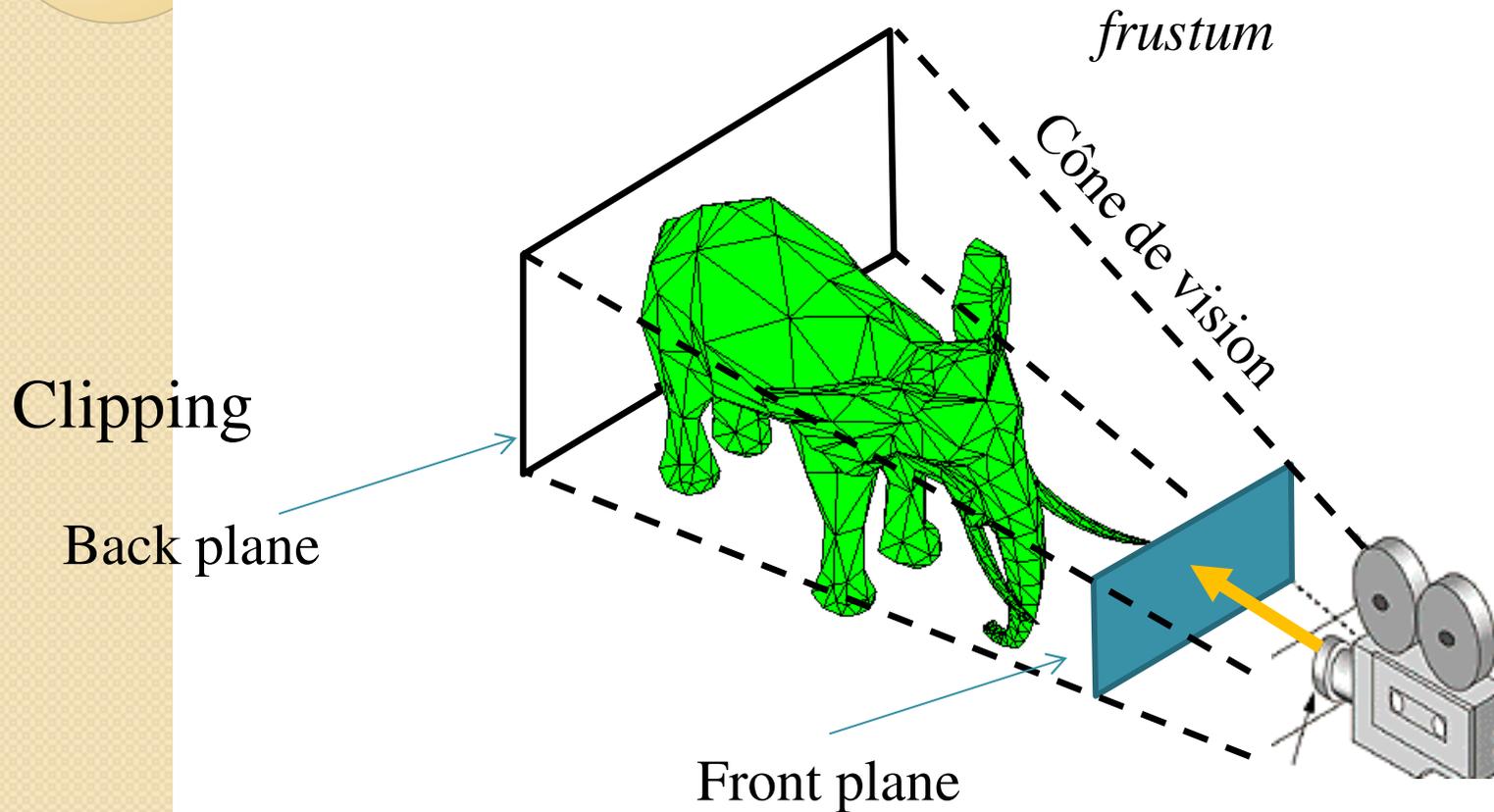
TP : Parties cachées

Visualisation d'une triangulation (3D) : **les parties cachées**

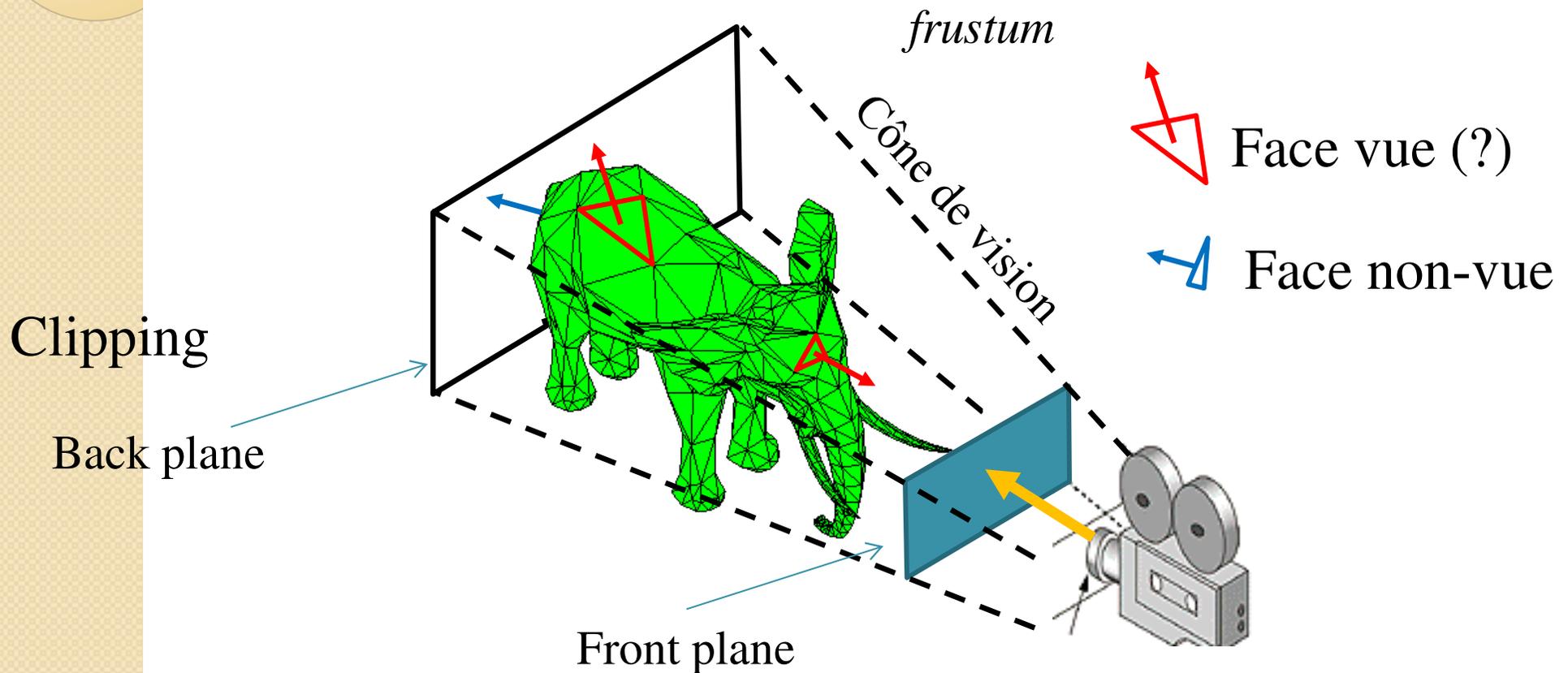
1. Triangulations : structure, calcul des normales & affichage
2. Rotation et translations
3. « Backface culling »
4. Parties cachées : algorithme du peintre



Prétraitement : le Clipping

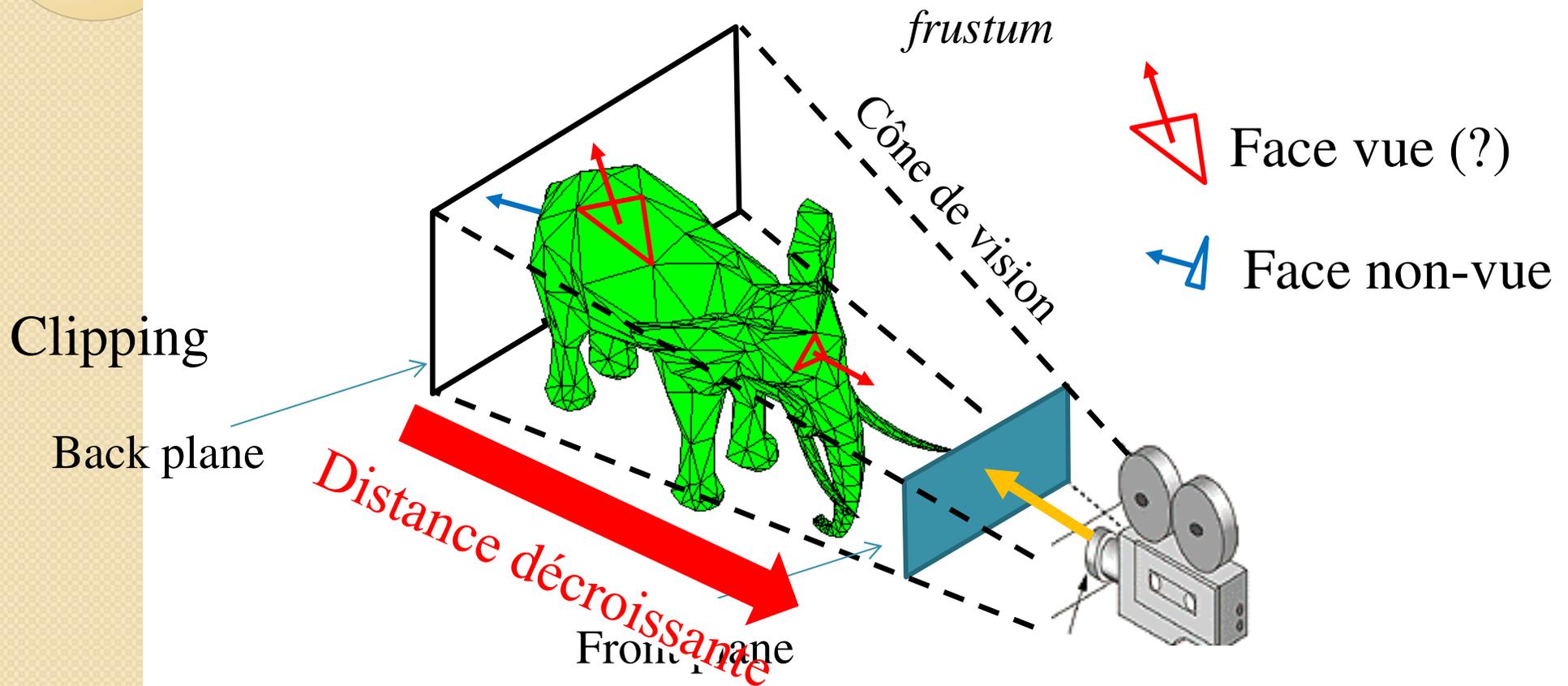


Prétraitement : Le Back face culling



Dans quel cas le « back-face culling » résout-il le problème des parties cachées ?

L'algorithme du peintre

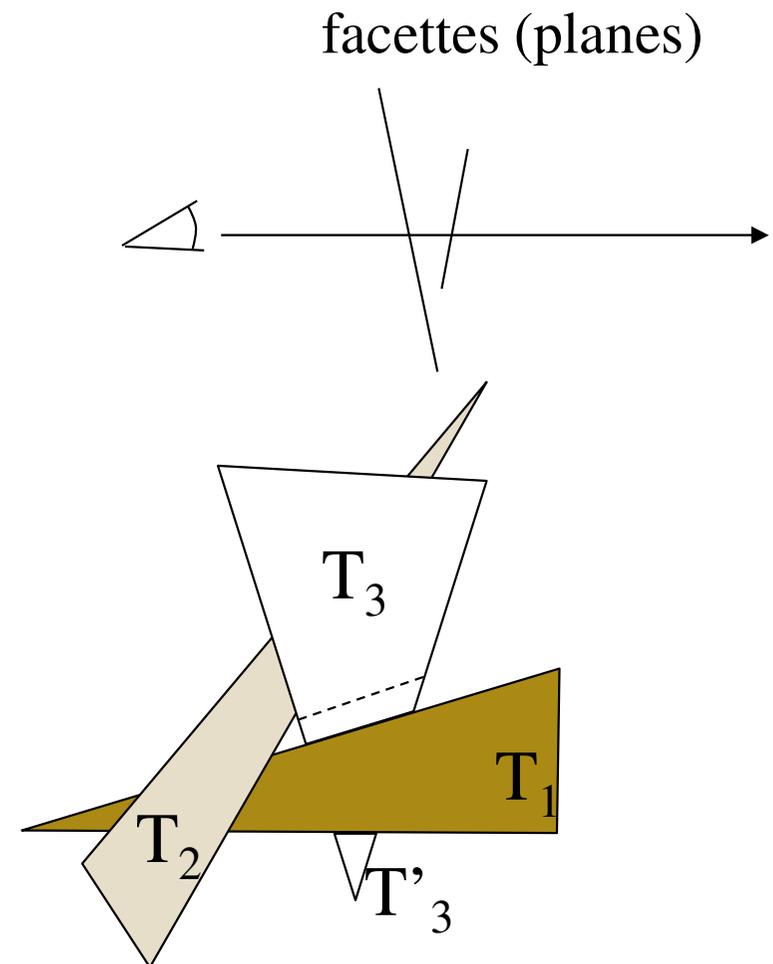


Algorithmes à liste de priorité (Algorithme du peintre)

- Liste ordonnée de facettes

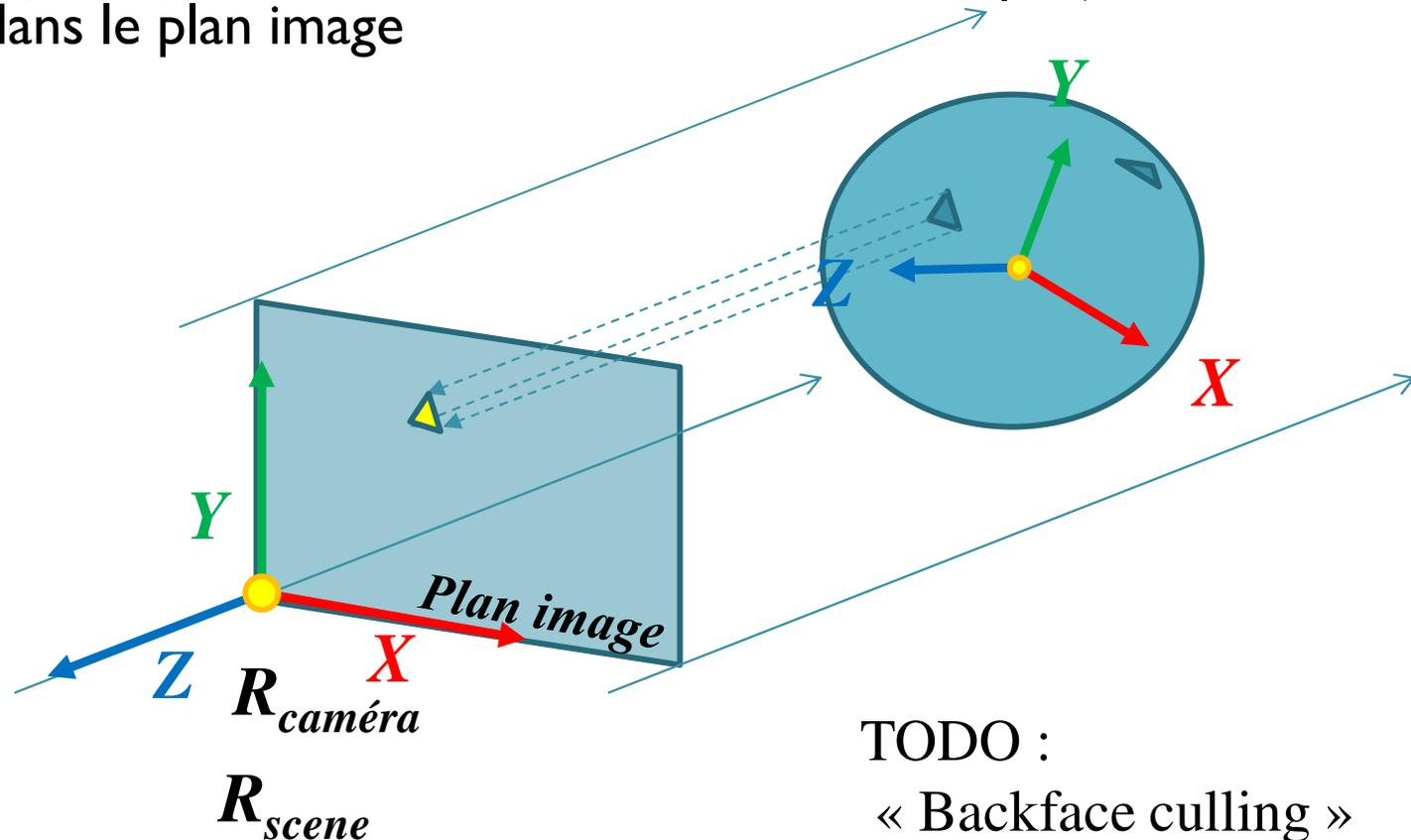
- trier les facettes suivant leur profondeur : Z-max
- afficher en Z-max décroissant
- ambiguïtés : problème du recouvrement cyclique...

$$T_2 > T_1$$



Hypothèses pour le TP

- Projection orthographique
- Pas de recouvrement cycliques
- $\text{Triplot2d}(\text{XYZcoord}, \text{itrnoe}, \text{couleur})$ est fournie : projection des triangles dans le plan image



TODO :
« Backface culling »
Algorithme du peintre