Modélisation surfacique
Partie II : Maillages

Ulysse Vimont

Équipe Imagine - Inria, LJK, INPG

2015-2016

1. Introduction
2. Justification d'utilisation
3. Structures de données
4. Formats de stockages
5. Opérations sur les maillages
6. Notions de topologie discrète
7. Bilan

1 Introduction
- Définition : Maillage
- Caractéristiques des faces
- Faces n-angulaires
- Ratios des nombres de primitives
- Tetrahèdres et polyhèdres
- Interpolation de caractéristiques

2 Justification d'utilisation

3 Structures de données

4 Formats de stockages

5 Opérations sur les maillages

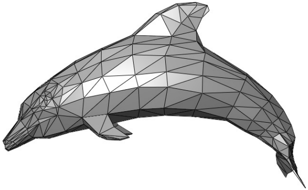
6 Notions de topologie discrète

Définition

Maillage polygonal :

Collection de sommets, d'arêtes, et de faces définissant la forme d'un objet polyhédral.
(Wikipedia)

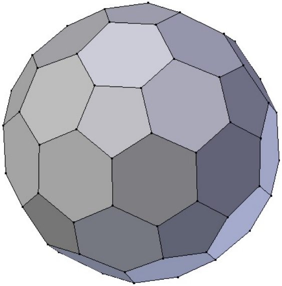
- structure surfacique discrète
- géométrie discrète
- graphe, hypergraphe



Triangles, quadrangles, et polygones

Dans le cas général, le mailllage est formé de n-gones.

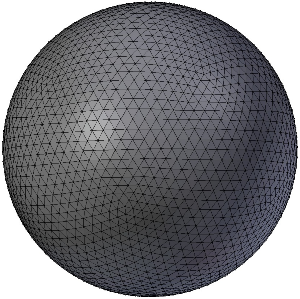
- problème : coplanarité
- triangulation



Ratios des nombres de primitives

En général, on utilise des maillages triangulaire ou quadrangulaires.

- triangle : pas de problème de triangulation
- quad : choix arbitraire de la diagonale



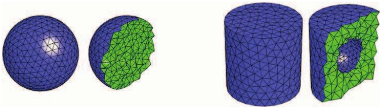
Exercice

Trouver les proportions optimales de sommets/arêtes/faces dans une tesselation triangulaire.

Tetrahèdres et polyhèdres

On peut imaginer des maillages contenant des polyhèdres (hyperfaces).

- utile pour le calcul des structures (éléments finis)
- extension naturelle en dimensions

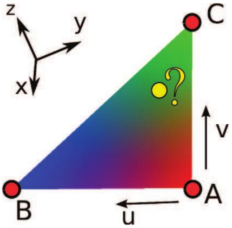


Interpolation de caractéristiques

Retour aux faces polygonales (triangle et quad).

Un somet possède diverse caractéristiques :

- une position
- une normale
- une couleur
- une coordonnée de texture
- éventuellement un paramètre scalaire (température, pression, ...) ou vectoriel (vitesse, force, ...)



Une face interpole linéairement ces caractéristiques sur un élément de surface (sauf parfois la normale).

Interpolation de caractéristiques

Pour définir chacun de ces paramètre en chaque point du triangle :

- interpolation bilinéaire (repère local)

$$\overrightarrow{AP} = x\overrightarrow{AB} + y\overrightarrow{AC}, (u, v \in [0,1]^2)$$

- coordonnées barycentriques

$$P = uA + vB + wC, (u, v, w \in [0,1]^3)$$

Remarques :

- équavalence des approches (normalization)
- identification des poids barycentriques avec les portions d'aires
- extension à l'extrapolation
- ray-tracing et coordonnées locales

1 Introduction

2 Justification d'utilisation
- Rendu
- Approximation linéaire

3 Structures de données

4 Formats de stockages

5 Opérations sur les maillages

6 Notions de topologie discrète

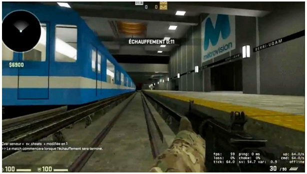
7 Bilan

Rasterisation

Définition

Conversion de données vectorielles (vector image) vers un bitmap (raster image).

- utilise la géométrie projective
- utilise le tracé de ligne de Bresenham
- exécutable en chaine sur une carte graphique
- extrèmement rapide
- utilisation pour la visualisation en temps réel



Syntaxe OpenGL : mode immédiat

```
glBegin(GL_TRIANGLES);

for(k_tri=0;k_tri<N_tri;k_tri++)
  for(k_vertex=0;k_vertex<3;k_vertex++)
    for(k_dim=0;k_dim<3;k_dim++)
    {
      x[k_dim] = vertex[3*connectivity[3*k_tri+k_vertex]+k_dim];
      n[k_dim] = normal[3*connectivity[3*k_tri+k_vertex]+k_dim];
      glNormal3d(n[0],n[1],n[2]);
      glVertex3d(x[0],x[1],x[2]);
    }

glEnd();
```

Syntaxe OpenGL : vertex array

```
glEnableClientState(GL_VERTEX_ARRAY);
glVertexPointer(3,GL_DOUBLE,0,&vertex[0]);

glEnableClientState(GL_NORMAL_ARRAY);
glNormalPointer(GL_DOUBLE,0,&normal[0]);

glDrawElements(GL_TRIANGLES,
               3*N_tri,
               GL_UNSIGNED_INT,
               &connectivity[0]);

glDisableClientState(GL_VERTEX_ARRAY);
glDisableClientState(GL_NORMAL_ARRAY);
```

Approximation linéaire

Approximation linéaire = interpolation linéaire de valeur (ie. de position)
Soit une forme 1D F définie par la fonction :

$$f : \mathbb{R} \to \mathbb{R}, x \mapsto f(x)$$

On considère sa décomposition polynomiale :

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)(x - x_0)^2}{2} + \dots$$

On échantillonne linéairement f par g avec un pas H :

$$g(x) = f(x_0) + f'(x_0)(x - x_0), \|x - x_0\| < H$$

$$\|f(x) - g(x)\| < K.\frac{f''(x_0)}{2}.H^2$$

- $f''(x_0)$ est assimilable à la courbure de F
- lorsqu'on divise le pas par deux, on diminue par quatre l'erreur d'approximation

1 Introduction

2 Justification d'utilisation

3 Structures de données
- Soupe de triangles
- Pourquoi plusieurs structures ?
- Winged-edge
- Half-edge
- Carte combinatoire
- Spécifique

4 Formats de stockages

5 Opérations sur les maillages

6 Notions de topologie discrète

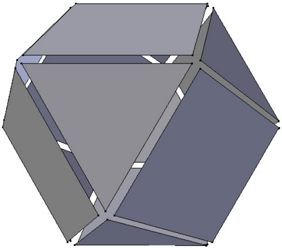
Soupe de triangles

Principe

Le maillage est décrit par deux listes :

- une liste de points précisant la position des sommets
- une liste de n-uplets d'indices de sommets représentant des n-gones

- La première liste est dite "géométrie", la seconde "connectivité".
- Équivalence point et sommet
- pas d'information sur les arêtes
- pas d'information sur les voisins des sommets et des faces
- pas de contrainte sur le type de faces ou de surfaces
- pas de lien depuis les sommets vers les faces



Intuition

Obtenir la liste des voisins d'un sommet peut coûter cher.

On peut faciliter l'accès aux données de voisinage en créant des listes d'adjacence :

- une liste sommet -> sommet
- une liste sommet -> faces
- une liste sommet -> arêtes
- une liste arête -> sommets
- une liste arête -> faces
- une liste face -> arête

Problèmes : redondance d'information

- volume de données
- maintenabilité des données

Pourquoi plusieurs structures ?

- les différentes manipulations que nous allons voir nécessitent un accès rapide à diverses informations de voisinage
- on ne peut encoder tout les voisins de tout les éléments
- il peut être lourd de maintenir une structure avec beaucoup de redondance

On fait donc un compromis entre :

- le temps d'accès à l'information
- le volume de données à stoquer
- la tractabilité du système

Il faut aussi prendre en compte le fait que certaines structures imposent des limitations sur ce qu'elles représentent (variété, orientabilité).

Résultat : on se retrouve souvent avec des soupes de triangles… mais pas toujours ! Par exemple :

- moka (carte combinatoire)
- CGAL, OpenMesh (half-edge)

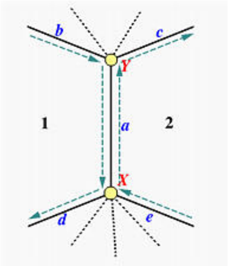
Winged-edge

Principe

Chaque arête contient la référence de :

- deux sommets (début et fin)
- deux faces (droites et gauche)
- quatres arêtes (parcours normal et inverse, prédécesseur et successeur)

- pas d'information sur les faces
- décris une variété (une arête est toujours partagée par deux faces)
- on peut se passer des références vers les faces (détection de cycles)



www.cs.mtu.edu

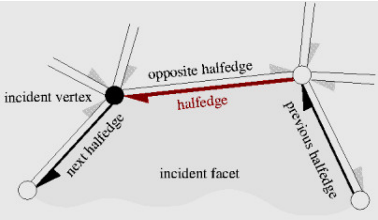
Half-edge

Principe

Chaque demi-arête stocke une référence vers :

- un sommet
- une face
- trois autres demi-arêtes :
  - précédent (dans la face)
  - suivant (dans la face)
  - opposée (sur l'arête)

- plus compacte que le winged-edge
- contient les mêmes informations (qu'on retrouve grâce à la demi-arête opposée)



www.cgal.org

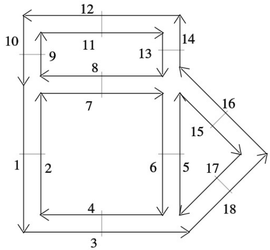
Carte combinatoire

Principe

Une carte combinatoire ND est définie par le N-uplet $M = (D, \{\beta_i\}_{i \in [\![1,N]\!]})$ où :

- D est un ensemble de brins (contenant un point)
- β_1 est une permutation sur D
- $\forall i > 1, \beta_i$ est une involution sur D
- $\forall i, j, 1 \leq i < i + 2 \leq j < n, \beta_i \circ \beta_j$ est une involution sur D

- extension des half-edge en dimension finie quelconque
- représente un espace ND fermé
- limité aux surfaces orientable
- extension aux surfaces non orientables : cartes généralisées



Spécifique

Il existe d'autres structures de représentation :

- quad edge
- liste de face doublement chainée
- liste d'arête doublement chainée

Toutes ce structures visent à rendre accessible l'information dont on a besoin en général (voisinage).
Il est tout à fait possible de créer sa propre structure facilitant l'accès à des informations spécifiques.

1 Introduction

2 Justification d'utilisation

3 Structures de données

4 Formats de stockages
- XML
- OFF
- OBJ (Wavefront)
- Autres

5 Opérations sur les maillages

6 Notions de topologie discrète

7 Bilan

XML

```
<?xml version="1.0"?>
<mesh>
 <geometry>
  <vertex>
   <x> 0.0 </x>
   <y> 0.0 </y>
   <z> 0.0 </z>
  </vertex>
  <vertex>
   <x> 1.0 </x>
   <y> 0.0 </y>
   <z> 0.0 </z>
  </vertex>
  <vertex>
   <x> 0.0 </x>
   <y> 1.0 </y>
   <z> 0.0 </z>
  </vertex>
 </geometry>
 <connectivity>
  <face N = "3">
   <v0> 1 </v0>
   <v1> 2 </v1>
   <v2> 3 </v2>
  </face>
 </connectivity>
</mesh>
```

- Tout est représentable en XML (car c'est un format extensible).
- Cependant ce n'est pas très économique en place.
- Du coup personne ne l'utilise.

OFF

Spécification :

- entête : *OFF*
- [*nombre de sommets*] [*nombre de faces*] [*nombre d'arêtes*]
- liste de points [x y z]
- liste de faces [*nombre de sommets*] [n_0 n_1 ... n_i]

```
OFF
3 1 0
0.0 0.0 0.0
1.0 0.0 0.0
0.0 1.0 0.0
3 0 1 2
```

- seconde ligne : spéciale allocation dynamique
- basique : pas de normales, pas de coordonnées de texture

OBJ

Spécification (partielle) :

- liste de points [**v** x y z] (v pour vertex)
- liste de coordonnées de texture [**vt** u v] (*facultatif*)
- liste de normales [**vn** x y z] (*facultatif*)
- liste de points [**f** $v_0/vt_0/vn_0$ $v_1/vt_1/vn10$ ... $v_i/vt_i/vn_i$] (ou simplement [**f** v_0 v_1 ... v_i])

```
# Ceci est un fichier OBJ

v 0.0 0.0 0.0
v 1.0 0.0 0.0
v 0.0 1.0 0.0

f 1 2 3
# note : les indices des
# sommets commencent a 1
```

#	*commentaire*
o	*nom de l'objet*
g	*nom du groupe*
s	*on ou off* (soft)
mtllib	*nom du fichier* (déclaration)
usemtl	*nom du matériau* (utilisation)

Autres

Il existe de nombreux autres formats, par exemple :

- PLY (Standford) : extension de OBJ
- STL (stereolithographie) : similaire à OBJ
- formats de scene

- Tous implémentent une soupe de triangle, avec diverses variations ou options (extensibilité, encodage de matrice de déformation,).
- Certain offrent un mode binaire plus compact.
- Pré-conversion vers une autre structure si besoin

1 Introduction

2 Justification d'utilisation

3 Structures de données

4 Formats de stockages

5 Opérations sur les maillages
- Opérations locales
- Opérations globales

6 Notions de topologie discrète

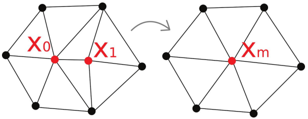
7 Bilan

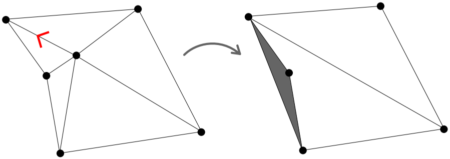
Effondrement d'arête

Définition

Opération consistant à fusionner deux sommets adjacents.

- valable pour un maillage triangulaire
- autrement appellé "edge collapse"
- bilan : t-2, e-3, s-1
- perte d'information



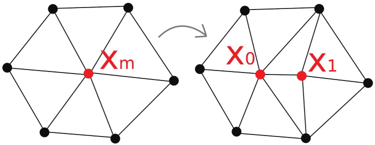


Division de sommet

Définition

Opération inverse de l'effondrement.

- autrement appellé "vertex split"
- bilan : t+2, e+3, s+1
- création d'information ?

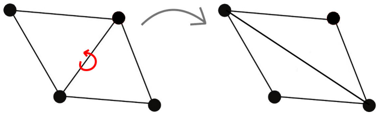


Basculement d'arête

Définition

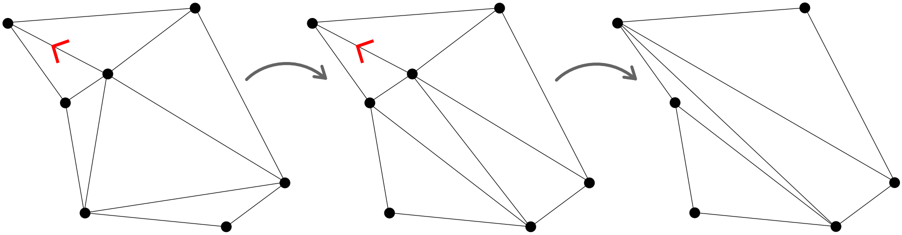
Étant donné deux triangles adjacents, formant un quadrilatère *convexe*.
Opération consistant à inverser la diagonale de celui-ci.

- autrement appellé "edge flip"
- bilan : topologie inchangée



Basculement pré-effondrement

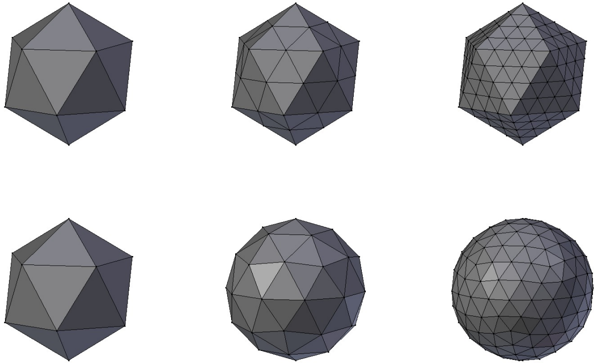
Lorsqu'une arête n'est pas contractable, on peut procéder à des basculements locaux pour changer la situation.



Subdivision

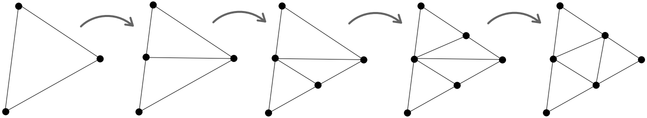
Définition

Opération consistant augmenter la résolution d'un maillage.



Subdivision

- On peut décomposer la subdivision en opérations élémentaires :



- bilan:
  - avant : $t = 1$, $e = 3/2$, $v = 1/2$
  - après : $t = 4$, $e = 6$, $v = 1$
  - soit : $t *= 4$, $e *= 4$, $v *= 2$
- Différent des surfaces de subdivision : on ne cherche pas à augmenter la contrôlabilité d'un important volume de données.
- Sur-échantillonnage de la surface représentée.
- Subdivision locale.

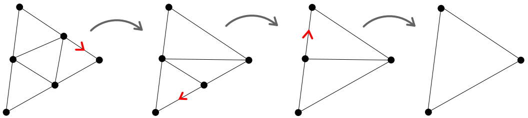
Démo Blender

Simplification

Définition

Opération consistant à réduire la résolution d'un maillage.

- Opération inverse de la subdivision
- On peut décomposer la simplification en opérations élémentaires :

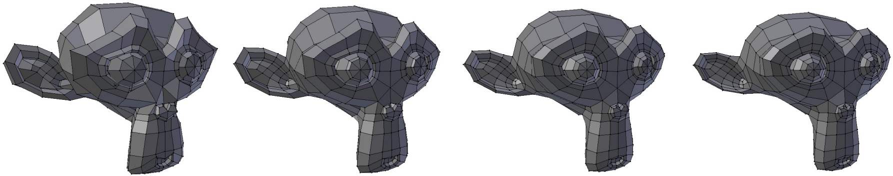


- Critère d'arrêt ? effondrement jusqu'à réduction de moitié du nombre de vertex.
- Critère de sélection de l'arête à effondrer ?
- subdivision + simplification = identité ?

Lissage

Définitions

Opération consistant en l'homogénéisation des différets attributs locaux d'un maillage (longueurs des arêtes, aires des triangles).

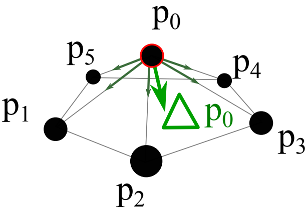


- Adaptabilité des méthodes de signal ?
- Analogie : lissage d'une image.
- Différents types de lissages :
  - laplacien
  - taubin
  - SVD
  - optimisation d'une fonction d'irrégularité

Laplacien d'un sommet

On définit le laplacien d'un attribut de sommet comme suit :

$$\Delta A(v_i) = \frac{1}{\|N(v_i)\|} \sum_{v_j \in N(v_i)} A(v_j) - A(v_i)$$



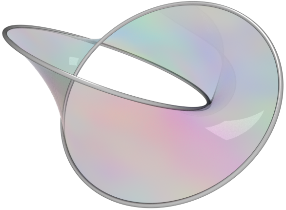
- $N(v)$ désigne les sommets voisins de v
- $A(v)$ désigne un attribut de v (position, couleur, ...)
- Moyenne des différences d'un sommet avec ses voisins (ie. différence de la moyenne des voisin d'un sommets avec lui-même)
- Analogie laplacien - Dérivée seconde - Courbure

Lissage Laplacien

Algorithme : Pour tout les sommets, appliquer la formule suivante :

$$A(v_i) \leftarrow A(v_i) + \lambda \times \Delta A(v_i)$$

- algorithme itératif
- *lambda* contrôle l'intensité du lissage ($\in [0, 1]$)
- *lambda* en fonction des sommets
- réduction du volume



Remaillage

Définitions

Opération consistant à recréer un maillage représentant le même objet.

Objectifs : Augmenter la qualité du maillage sans (trop) modifier le contenu, modifier le pas d'échantillonnage.
Exemple :

1. on dispose d'un maillage
2. on le transforme en nuage de points (destruction de la connectivité)
3. puis en fonction implicite (reconstruction, RBF par ex.)
4. et à nouveau en maillage en maillage (marching cube par ex.)

Non conservation de la topologie.
démo Blender

1 Introduction

2 Justification d'utilisation

3 Structures de données

4 Formats de stockages

5 Opérations sur les maillages

6 Notions de topologie discrète
- Adjacence et incidence
- Distance
- Voisinage
- Valence
- Qualité d'un maillage
- Variété

Adjacence

Définition

Deux éléments de même dimension N d'un maillages sont adjacent si ils partagent un élément de dimension $N-1$ ou $N+1$.

Peuvent être adjacents :

- deux sommets
- deux arêtes
- deux faces
- ...

Incidence

Définition

Deux éléments d'un maillages sont incidents si il existe un point de l'espace appartenant aux deux.

Peuvent être incidents :

- deux triangles
- un triangle et une arête
- deux sommets

Notion ensembliste :

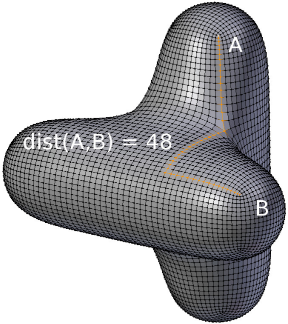
$$\exists p \in \mathbb{R}^3 | p \in F_1, p \in F_2$$

Distance

Définition

Nombre d'arêtes composant le plus court chemin entre deux sommets de la surface.

- non unicité du chemin
- indépendant de l'échelle
- attribut topologique
- entier

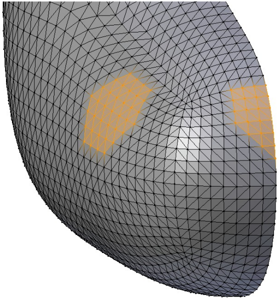


Voisinage

Définition

Pour un sommet A, ensemble des sommets étant à une distance inférieur à un N donné.

- notion de ring
- utilisation pour le calcul du plan tangent
- définition de la normale



Valence

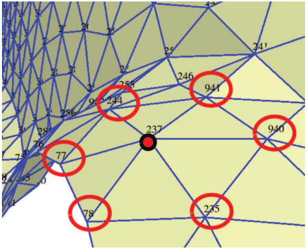
Définition

Pour un point donné, nombre d'arêtes incidentes à ce point.

Définition équivalente

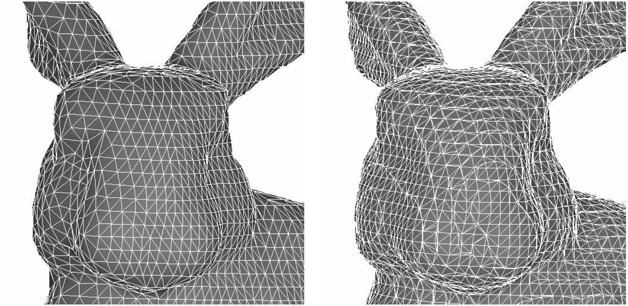
Pour un point donné, nombre de 1-voisins de ce point.

- on parle également de degré



Qualité

- notion qualitative
- associé à la régularité des faces (forme, connectivité)
- associé à l'uniformité de la distribution des valences
- utile pour le calcul des structures
- utile pour la paramétrisation



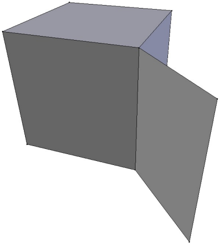
Marching triangle, Hilton et al.

Variété

Définition

Un maillage est une variété si le 1-voisinage de chaque sommet forme une unique chaîne, fermée ou non.

- critère de régularité topologique (et non géométrique)
- lien avec la notion continue de variété différentielle
- séparation locale intérieur/extérieur
- variété simpliciale
- extension : complexe simplicial



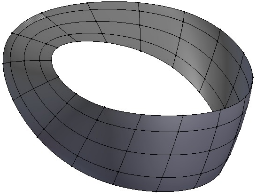
Ceci n'est pas une variété.

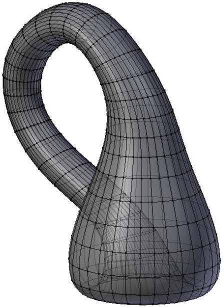
Orientation

Les faces possèdes une caractéristiques particulière : *l'orientation*.

- équivalent au sens de la normale
- utilisé pour le rendu (culling)
- utilisé pour définir l'intérieur et l'extérieur de l'objet

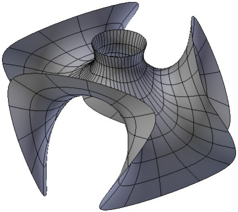
On peut souvent définir une même orientation pour tout le maillage, *mais pas toujours*!
Voir ci-contre.





Bilan :

- Les maillages servent à représenter les surfaces affines par morceau
- Ils servent souvent d'approximiation pour des surfaces quelconques
- Ils sont représentables par différentes structures et différents formats de fichiers
- À tester dans vos projets !



Fin !