



Lyon 1

Mesh and Computational Geometry

Raphaëlle Chaine

Université Claude Bernard Lyon 1

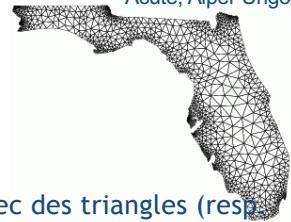
Master Informatique
ID3D
Image, Développement et Technologie 3D



1

Génération de maillages

Acute, Alper Ungor



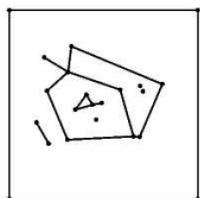
- Remplir un objet avec des triangles (resp. des tétraèdres en 3D) bien formés (ou bien d'autres types d'éléments)

60

Génération de maillages

- Problème posé :

- Un PSLG (Planar-Straight-Line Graph)
Graphe Planaire de Lignes Droites
- Trouver une triangulation respectant ce PSLG



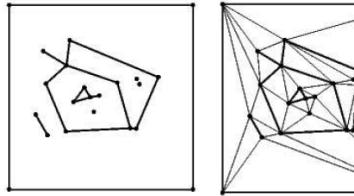
Ruppert

61

Génération de maillages

- Problème posé :

- Un PSLG (Graphe Planaire de Lignes Droites)
- Trouver une triangulation respectant ce PSLG



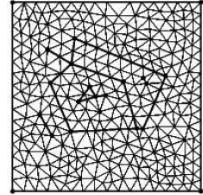
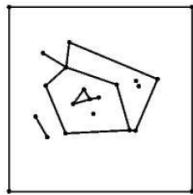
Un exemple de triangulation satisfaisant le problème mais présentant de petits angles... (Delaunay Constraint)
⁶²

62

Génération de maillages

- Problème posé :

- Un PSLG (Graphe Planaire de Lignes Droites)
- Trouver une triangulation respectant ce PSLG



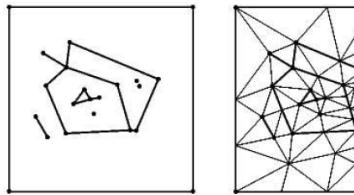
Un exemple de triangulation uniforme satisfaisant le problème : des angles > 22,5° mais triangles trop nombreux
⁶³

63

Génération de maillages

- Problème posé :

- Un PSLG (Graphe Planaire de Lignes Droites)
- Trouver une triangulation respectant ce PSLG

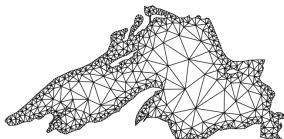


Triangulation obtenue par raffinement de Delaunay : un nombre raisonnable de triangles pour des angles > 20°
⁶⁴

64

Génération de maillages

- Par raffinement de Delaunay

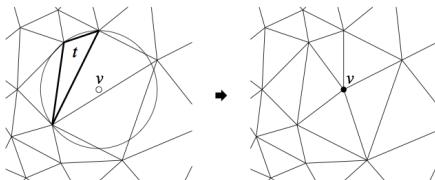


Tous les angles des triangles sont ici > 25°

65

Génération de maillages

- Par raffinement de Delaunay
- Principe : insérer le centre du cercle circonscrit des triangles mal formés



66

Génération de maillages

- Par raffinement de Delaunay
- 2 principes :
 - Insérer les arêtes de contrainte dans Delaunay par insertion de leur milieu
 - insérer le centre du cercle circonscrit des triangles mal formés
- MAIS COMMENT FAIRE AU VOISINAGE DES ARETES DE CONTRAINTE?

67

Réfléchissons ensemble...

- Principe : insérer le centre circonscrit des triangles mal formés ...
- ~~• Sauf si celui ci tombe l'autre côté d'une arête dite de contrainte~~
- MAIS SI ON VEUT RESTER DANS DELAUNAY?**
 - Subdivision de la contrainte à la place, par insertion du milieu
EN PRATIQUE ON COUPE LES CONTRAINTES ENCORE PLUS SOUVENT!
 - Subdiviser la contrainte lorsque un point à ajouter tombe trop près de l'arête (ie. dans son cercle diamétral)

DELAUNAY
CONTRAINTE
DELAUNAY

68

Génération de maillages

- Attention :
 - Si les angles d'un triangle sont satisfaisants, on ne coupe pas une de ses arêtes de contrainte uniquement parce que son cercle diamétral est non vide!
- Principe : le raisonnement n'est mis en œuvre que pour les points à ajouter :
 - insérer le centre circonscrit des triangles mal formés ...
 - Sauf s'il tombe à proximité d'une arête de contrainte (dans son cercle diamétral)
-->Subdivision de la contrainte à la place

69

Génération de maillages

Remarque : Il existe également une version forte de l'algorithme où on coupe en deux tous les segments de contrainte dont la sphère diamétrale est non vide, mais elle ajoute plus de points ;-)



70

69

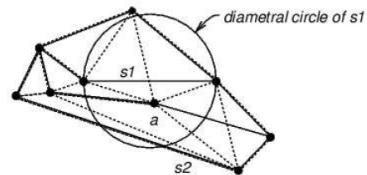
Génération de maillages

- Problème : dans quel ordre procéder pour que la technique converge?
- Algorithme de Ruppert
 - Garantie sur la forme (aspect ratio)
 - Garantie sur la taille de la sortie (par rapport à la taille optimale)

71

Algorithme de Ruppert

- Vocabulaire : Sommet “accrochant” un segment de contrainte :
 - Sommet x situé à l’intérieur du cercle diamétral à un segment s
 - Remarque : Un segment qui n’est pas dans Delaunay est accroché par un sommet



72

Algorithme de Ruppert

- Principe de l’algorithme
 - En priorité : Casser les segments de contrainte Non Delaunay par insertion du milieu
 - PUIS Casser les triangles caractérisés par un mauvais aspect ratio par insertion du centre circonscrit ou du centre d’une contrainte

73

Algorithm *DelaunayRefine*
 INPUT: planar straightline graph X ;
 desired minimum angle bound α .
 OUTPUT: triangulation of X , with all angles $\geq \alpha$.
 Initialize:
 add a bounding square B to X :
 compute extremes of X : $x_{min}, y_{min}, x_{max}, y_{max}$
 let $span(X) = \max(x_{max}-x_{min}, y_{max}-y_{min})$
 let B be the square of side $3 \times span(X)$, centered on X
 add the four boundary segments of B to X
 let segment list $S = \text{edges of } X$
 let vertex list $V = \text{vertices of } X$
 compute initial Delaunay triangulation $DT(V)$
 repeat:
 while any segment s is encroached upon:
 SplitSeg(s)
 let t be (any) skinny triangle ($\min \text{angle} < \alpha$)
 let p be t ’s circumcenter
 if p encroaches upon any segments s_1, \dots, s_k then
 for $i = 1$ to k :
 SplitSeg(s_i)
 else
 SplitTri(t) (* adds p to V *)
 endif
 until no segments encroached upon, and no angles $< \alpha$
 output current Delaunay triangulation $DT(V)$

74

Ici présenté dans sa version forte

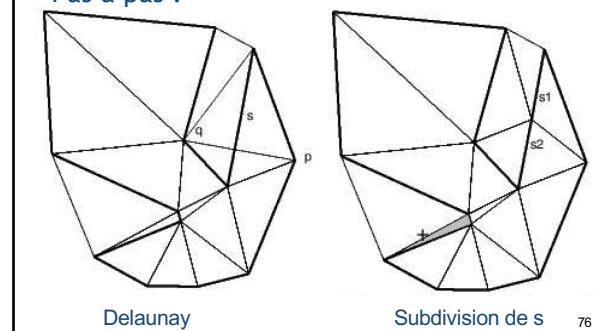
Algorithme de Ruppert

- (Version faible) Pas à pas :
 - Polygone en entrée
 - Le segment s n'est pas dans Delaunay

75

Algorithme de Ruppert

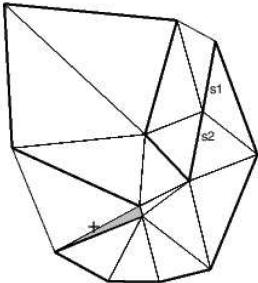
- Pas à pas :



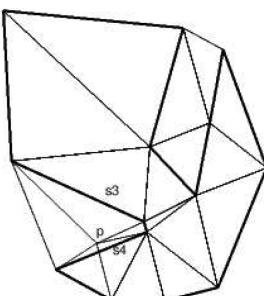
76

Algorithme de Ruppert

- Pas à pas :



Triangle le plus mal formé
(en gris)

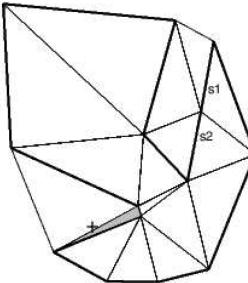


Mais l'insertion de son
 cercle circonscrit
 accrocherait s3 et s4 ⁷⁷

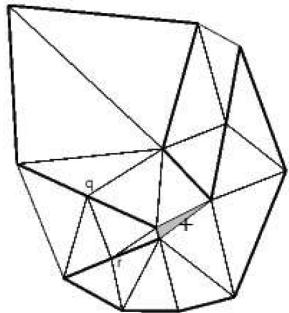
77

Algorithme de Ruppert

- Pas à pas :



Triangle le plus mal formé
(en gris)

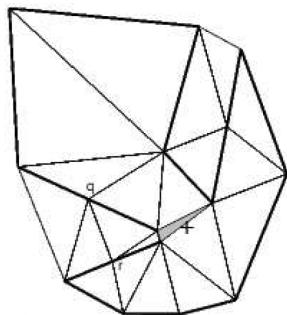


Du coup on coupe s3 et s4 ⁷⁸

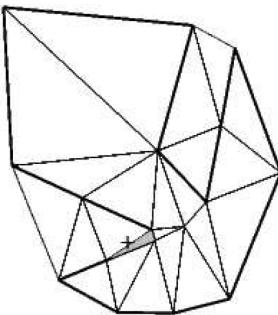
78

Algorithme de Ruppert

- Pas à pas :



Du coup on coupe s3 et s4

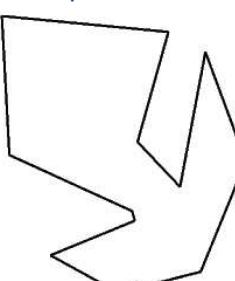


Insertion du centre du plus
mauvais triangle précédent ⁷⁹

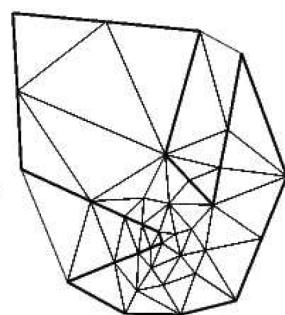
79

Algorithme de Ruppert

- Pas à pas :



Polygone en entrée

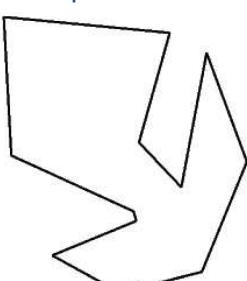


Résultat pour un angle minimum
 autorisé correspondant à 25° ⁸⁰

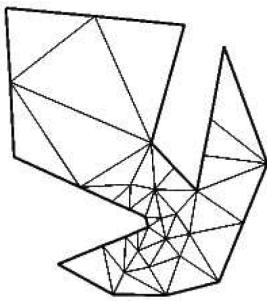
80

Algorithme de Ruppert

- Pas à pas :



Polygone en entrée



Suppression des triangles
 extérieurs ⁸¹

81

Algorithme de Ruppert

- Terminaison de l'algorithme

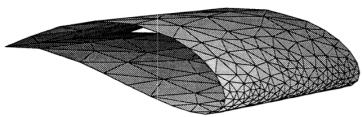
- Preuve basée sur la notion de *local feature size*

- Penser à protéger les petits angles entre segments de contraintes

82

Génération de maillages

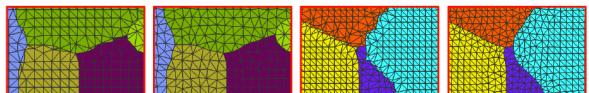
- Adaptation du même principe aux surfaces



Chew

83

Amélioration de la qualité du maillage tétraédrique d'un volume segmenté en utilisant un flot de déformation géométrique



Juelin Leng, Yongjie Zhang, Guoliang Xu LSEC

Cette fois le maillage dispose déjà d'un grand nombre de sommets. On ne va plus procéder par ajout mais par déplacement.

84

Contexte

- En entrée : Maillage tétraédrique segmenté et bruité
- Amélioration de sa qualité (obtention d'une qualité « acceptable » mais pas forcément optimale)
 - Lissage et débruitage des frontières ... classique en traitement d'image
 - Tout en maintenant des éléments de qualité ... classique en géométrie algorithmique (STELLAR de Shewchuk, Mesquite, ...)
- Remarque : les frontières sont non *manifold*

85

Méthode

- Contraindre les sommets frontières à y rester tout en les lissant et en les débruitant
- Les frontières sont lissées indépendamment du reste, avec « préservation » de leur forme et du volume qu'elles délimitent
- Double objectif:
 - Régularisation des frontières
 - Minimisation d'une fonctionnelle mesurant la qualité des tétraèdres

86

Prise en compte des frontières

- Catégorisation des sommets
 - Sommets intérieurs
 - Sommets de frontières surfaciques
 - Sommets de courbes frontières
 - Sommets fixes

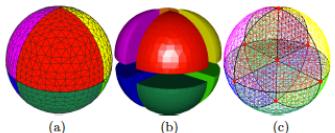


Fig 2.1: (a) A segmented tetrahedral mesh; (b) the object is segmented into eight components; (c) the boundary mesh of (a).

87

Traitements spécifiques

- Mouvements de sommets dédiés au lissage
 - Flots spécifiques sur chacun des éléments de frontières pour les lisser en préservant la forme
- Mouvement de sommets dédiés à l'amélioration de qualité des tétraèdres (Relaxation / régularisation)
 - Les sommets sont ensuite déplacés pour minimiser la fonctionnelle de qualité des tétraèdres
- Opérations topologiques pour améliorations résiduelles

88

Traitements spécifiques

- Lissage des courbes frontières avant celui des surfaces
 - Un lissage sans précautions entraîne un effondrement en volume d'une forme
 - Diffusion de courbe : minimisation de la longueur en préservant l'aire
 - Diffusion de surface : minimisation de l'aire en préservant le volume
- La forme globale reste préservée
 - ... si le processus s'arrête une fois le bruit supprimé

89

Traitement des courbes

- Curve diffusion flow
 - Diffusion de courbe : minimisation de la longueur en préservant l'aire
 - Déplacement des points dans la direction normale jusqu'à ce que la courbure s'équilibre localement
 - (Averaged) mean curvature flow
 - Régularisation de la courbe : favoriser une équirépartition des sommets sur la courbe
 - Relaxation visant à minimiser une fonctionnelle
 - Déplacement dans la direction tangentielle

90

Curve diffusion flow

- Diffusion de courbe : minimisation de la longueur en préservant l'aire
 - Déplacement des points dans la direction normale

$$\frac{dx_i}{dt} = -(\Delta \kappa_i) n_i, \quad i = 1, \dots, n-1,$$
- Propriétés différentielles estimées
 - Tangente et longueur

$$t_i = \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{\|\mathbf{x}_i - \mathbf{x}_{i-1}\|},$$
 - Courbure et normale

$$\kappa_i = \|\mathbf{x}_i\|, \quad \kappa_i = \frac{\mathbf{t}_{i+1} - \mathbf{t}_i}{s_i}, \quad n_i = \frac{\mathbf{n}_i}{\kappa_i},$$

$$s_i = \frac{\|\mathbf{x}_i - \mathbf{x}_{i-1}\| + \|\mathbf{x}_i - \mathbf{x}_{i+1}\|}{2}$$

91

Curve diffusion flow

- Diffusion de courbe :
 - Déplacement des points dans la direction normale

$$\frac{dx_i}{dt} = -(\Delta \kappa_i) n_i, \quad i = 1, \dots, n-1,$$
- Estimation du Laplacien

$$\left(\frac{\kappa_{i+1} - \kappa_i}{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|} - \frac{\kappa_i - \kappa_{i-1}}{\|\mathbf{x}_i - \mathbf{x}_{i-1}\|} \right) / s_i$$
- Schéma explicite d'Euler

$$\mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)} - \tau \Delta \kappa_i \mathbf{n}_i, \quad i = 1, \dots, n-1,$$
 - Critère d'arrêt?

92

Régularisation de la courbe

- Diffusion de courbe :
 - Déplacement des points dans la direction normale

$$\frac{dx_i}{dt} = -(\Delta \kappa_i) n_i, \quad i = 1, \dots, n-1,$$
 - Régularisation de la courbe : favoriser une équirépartition des sommets sur la courbe
 - Relaxation tangentielle visant à minimiser une fonctionnelle

$$\mathcal{E}(\mathcal{C}) = \frac{1}{2} \sum_{i=1}^n (\|\mathbf{x}_i - \mathbf{x}_{i-1}\| - h)^2,$$
- h longueur moyenne des arêtes de la courbe

93

Régularisation de la courbe

- Régularisation de la courbe : favoriser une équirépartition des sommets sur la courbe
 - Relaxation tangentielle visant à minimiser une fonctionnelle

$$\mathcal{E}(\mathcal{C}) = \frac{1}{2} \sum_{i=1}^n (\|\mathbf{x}_i - \mathbf{x}_{i-1}\| - h)^2,$$
- Dérivée partielle pour un petit déplacement dans une direction

$$\mathbf{x}_i \rightarrow \mathbf{x}_i + \epsilon_i \Phi_i, \quad \Phi_i \in \mathbb{R}^3, \quad i = 1, \dots, n-1$$
- $$\delta(\mathcal{E}, \Phi_i) = \frac{\partial \mathcal{E}(\mathcal{C}, \epsilon_i)}{\partial \epsilon_i} \Big|_{\epsilon_i=0} = ((\|\mathbf{x}_{i+1} - \mathbf{x}_i\| - h) \frac{\Phi_i^T (\mathbf{x}_i - \mathbf{x}_{i+1})}{\|\mathbf{x}_i - \mathbf{x}_{i+1}\|} + ((\|\mathbf{x}_i - \mathbf{x}_{i-1}\| - h) \frac{\Phi_i^T (\mathbf{x}_i - \mathbf{x}_{i-1})}{\|\mathbf{x}_i - \mathbf{x}_{i-1}\|})$$

94

95

Régularisation de la courbe

$$\mathbf{x}_i \rightarrow \mathbf{x}_i + \epsilon_i \Phi_i, \quad \Phi_i \in \mathbb{R}^3, \quad i = 1, \dots, n-1$$

- Dérivée partielle $\delta(\mathcal{E}, \Phi_i)$ pour un petit déplacement dans une direction
- On choisit ici la direction tangentielle $\Phi_i = \mathbf{e}_i$.
- Descente du gradient

$$\frac{d\mathbf{x}_i}{dt} + \delta(\mathcal{E}, \Phi_i) \mathbf{e}_i = \mathbf{0}$$

Soit, après discréétisation

$$\begin{aligned} & \frac{\mathbf{x}_i^{(k+1)} - \mathbf{x}_i^{(k)}}{\tau} + (\|\mathbf{x}_{i+1}^{(k)} - \mathbf{x}_i^{(k)}\| - h) \frac{\mathbf{e}_i \mathbf{e}_i^T (\mathbf{x}_i^{(k)} - \mathbf{x}_{i+1}^{(k)})}{\|\mathbf{x}_i^{(k)} - \mathbf{x}_{i+1}^{(k)}\|} \\ & + (\|\mathbf{x}_i^{(k)} - \mathbf{x}_{i-1}^{(k)}\| - h) \frac{\mathbf{e}_i \mathbf{e}_i^T (\mathbf{x}_i^{(k)} - \mathbf{x}_{i-1}^{(k)})}{\|\mathbf{x}_i^{(k)} - \mathbf{x}_{i-1}^{(k)}\|} \\ & = \mathbf{0}. \end{aligned}$$

96

Traitement des frontières surfaciques

- Averaged mean curvature flow
 - Diffusion des surfaces :
 - minimisation de l'aire en préservant le volume
 - respect des courbes frontières qui sont contraintes
 - déplacement des points dans la direction normale
 - Régularisation des surfaces : favoriser une équirépartition des sommets sur les patches
 - Relaxation visant à minimiser une fonctionnelle

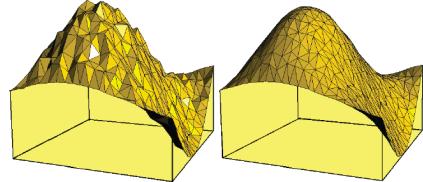
97

Averaged mean curvature flow

- Diffusion des surfaces, H courbure moyenne

$$\frac{\partial \mathbf{x}}{\partial t} = [H(\mathbf{x}) - \bar{h}(t)] \mathbf{n}(\mathbf{x}), \quad S(0) = S_0, \quad \partial S(t) = \Gamma$$

$$\int_{S(t)} H dA / \int_{S(t)}$$



98

Averaged mean curvature flow

- H courbure moyenne, calculée sur approximation quadratique dans 1-voisinage
- Moyenne de la Courbure moyenne (averaged mean)
 - Calcul de $\int_{S(t)} H dA / \int_{S(t)}$

nécessitant un élément d'aire

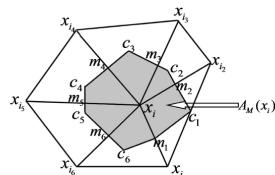


Fig. 3.4: Area represented by x_i . $\{m_j\}_{j=1}^6$ are midpoints of edges $[x_i x_{i+j}]$. c_j is the circumcenter point for the triangle $[x_i x_{i+j} x_k]$ if the triangle is non-obtuse; if the triangle is obtuse, c_j is chosen to be the midpoint of the longest edge.

Régularisation des frontières surfaciques

- Pour une meilleure répartition des sommets

- Minimisation d'une fonctionnelle

$$\mathcal{E}(\mathcal{S}) = \frac{1}{2} \sum_{i=1}^N \sum_{j \in N(i)} (\|\mathbf{x}_j - \mathbf{x}_i\| - h)^2$$

Voisins de i

$$h = \left(\frac{4}{\sqrt{3}} A \right)^{\frac{1}{2}}$$

Aire moyenne des triangles

- Dérivée partielle

- pour un petit déplacement dans une direction Φ_i

$$\delta(\mathcal{E}, \Phi_i) = \frac{\partial \mathcal{E}(\mathcal{S}, \epsilon_i)}{\partial \epsilon_i} \Big|_{\epsilon_i=0} = \sum_{j \in N(i)} (\|\mathbf{x}_j - \mathbf{x}_i\| - h) \frac{\Phi_i^T (\mathbf{x}_i - \mathbf{x}_j)}{\|\mathbf{x}_i - \mathbf{x}_j\|}$$

99

100

Régularisation des patches de surfaces

- Minimisation d'une fonctionnelle

$$\mathcal{E}(\mathcal{S}) = \frac{1}{2} \sum_{i=1}^N \sum_{j \in N(i)} (\|\mathbf{x}_j - \mathbf{x}_i\| - h)^2$$

- Dérivée partielle $\delta(\mathcal{E}, \Phi_i)$

- pour un petit déplacement dans une direction Φ_i
- utilisé pour calculer le gradient

- Déplacement contraint à être tangentiel

- Gradient dans le plan tangent $\Phi_i = \mathbf{e}_i^{(1)}$ and $\Phi_i = \mathbf{e}_i^{(2)}$
- Décomposition du mouvement de chaque sommet en deux mouvements

$$\frac{d\mathbf{x}_i}{dt} + \delta(\mathcal{E}, \mathbf{e}_i^{(1)}) \mathbf{e}_i^{(1)} + \delta(\mathcal{E}, \mathbf{e}_i^{(2)}) \mathbf{e}_i^{(2)} = \mathbf{0}, \quad i = 1, \dots, N.$$

101

Régularisation des surfaces

- Déplacement contraint à être tangentiel

- Gradient dans le plan tangent

$$\frac{d\mathbf{x}_i}{dt} + \delta(\mathcal{E}, \mathbf{e}_i^{(1)}) \mathbf{e}_i^{(1)} + \delta(\mathcal{E}, \mathbf{e}_i^{(2)}) \mathbf{e}_i^{(2)} = \mathbf{0}, \quad i = 1, \dots, N.$$

- Après discréétisation

$$\begin{aligned} & \frac{\mathbf{x}_i^{(k+1)} - \mathbf{x}_i^{(k)}}{\tau} + \mathbf{e}_i^{(1)} \sum_{j \in N(i)} (\|\mathbf{x}_j^{(k)} - \mathbf{x}_i^{(k)}\| - h) \frac{(\mathbf{e}_i^{(1)})^T (\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)})}{\|\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\|} \\ & + \mathbf{e}_i^{(2)} \sum_{j \in N(i)} (\|\mathbf{x}_j^{(k)} - \mathbf{x}_i^{(k)}\| - h) \frac{(\mathbf{e}_i^{(2)})^T (\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)})}{\|\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\|} \\ & = \mathbf{0}, \end{aligned}$$

with $i = 1, \dots, N$.

- Schéma explicite avec test de non réversion de tétraèdre!

102

Amélioration de la qualité de tétraèdres

- Minimisation d'une fonctionnelle favorisant des bons facteurs d'aspect
- Déplacement des sommets
- Opérations topologiques pour améliorations résiduelles
 - Face flips sur les éléments intérieurs
 - Edge removal sur les arêtes frontières

103

Qualité de tétraèdres

- Facteur d'aspect d'un tétraèdre (peut –être négatif!)

$$Q = \frac{8 \cdot 3^{\frac{5}{2}} V}{(\sum_{j=1}^6 e_j^2)^{\frac{3}{2}}}$$

- Reflète bien tous les défauts possibles

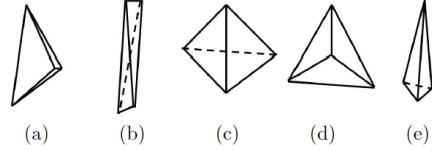


Fig 2.2: Examples of poorly-shaped elements.

104

Amélioration de la qualité de tétraèdres

- Fonctionnelle reflétant la qualité

$$\mathcal{E}(\mathbf{x}_1, \dots, \mathbf{x}_N) = \sum_{\eta \in \mathcal{T}} \max \left(\frac{1}{Q_\eta} - q, 0 \right)^p = \sum_{\eta \in \mathcal{T}_0} (\bar{Q}_\eta - q)^p.$$

pour faire ressortir la présence de mauvais tétraèdres (p et q 2 paramètres)

- Partie de la fonctionnelle impliquant \mathbf{x}_i

$$E(\mathbf{x}_i) = \sum_{\eta \in N(i) \cap \mathcal{T}_0} (\bar{Q}_\eta(\mathbf{x}_i) - q)^p$$

105

Amélioration de la qualité de tétraèdres

- Partie de la fonctionnelle impliquant \mathbf{x}_i

$$E(\mathbf{x}_i) = \sum_{\eta \in N(i) \cap \mathcal{T}_0} (\bar{Q}_\eta(\mathbf{x}_i) - q)^p$$

- Dérivée partielle pour une descente du gradient

$$\begin{aligned} \delta(E(\mathbf{x}_i), \Phi) &= \frac{\partial E(\mathbf{x}_i + \epsilon \Phi)}{\partial \epsilon} \Big|_{\epsilon=0} \\ &= \sum_{\eta \in N(i) \cap \mathcal{T}_0} (\bar{Q}_\eta(\mathbf{x}_i) - q)^{p-1} \frac{\partial \bar{Q}_\eta(\mathbf{x}_i + \epsilon \Phi)}{\partial \epsilon} \Big|_{\epsilon=0}. \end{aligned}$$

106

Amélioration de la qualité de tétraèdres

- Descente du gradient

– Pour les points intérieurs

$$\mathbf{d}_i = - \sum_{l=1}^3 \delta(E(\mathbf{x}_i), \mathbf{e}_i^{(l)}) \mathbf{e}_i^{(l)},$$

where $\mathbf{e}_i^{(1)} = [1, 0, 0]^T$, $\mathbf{e}_i^{(2)} = [0, 1, 0]^T$, and $\mathbf{e}_i^{(3)} = [0, 0, 1]^T$

– Pour les sommets de surface

$$\mathbf{d}_i = - \sum_{l=1}^2 \delta(E(\mathbf{x}_i), \mathbf{e}_i^{(l)}) \mathbf{e}_i^{(l)},$$

where $\mathbf{e}_i^{(1)}$ and $\mathbf{e}_i^{(2)}$ are orthogonal tangential directions at \mathbf{x}_i .

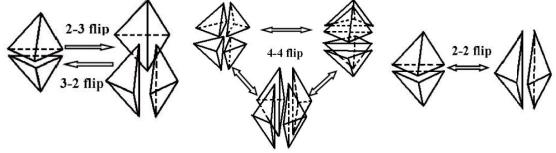
– Pour les sommets de courbes frontières $\mathbf{d}_i = -\delta(E(\mathbf{x}_i), \mathbf{e}_i) \mathbf{e}_i$

107

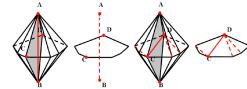
Modifications de connectivité pour les mauvais tétraèdres résiduels

- Flips de certaines faces de ces tétraèdres

- Flips de certaines arêtes de ces tétraèdres



- Suppression de certaines arêtes (si arêtes frontières) de ces tétraèdres



108

Algorithme

Algorithm 1 TopologicalTransformation(ε, L)

```

1: for l=1:L do
2:   Compute the volume-length quality metric  $Q$  for all
   tetrahedra in the segmented mesh.
3:   for each tetrahedron  $\eta$  with  $Q_\eta < \varepsilon$  do
4:     for i = 1 : 4 do
5:       Set  $f_i = -1$ ;
6:       If the 2-3 flip operation is legal for removing the
       i-th face, then set  $f_i$  as the worst quality of the
       three new tetrahedra.
7:     end for
8:     for i = 1 : 6 do
9:       Set  $e_i = -1$ ;
10:      If the i-th edge is interior edge, try remove it us-
        ing the 3-2 flip or 4-4 flip operation. If succeed,
        then set  $e_i$  as the worst quality of the newly
        tetrahedra.
11:      If the i-th edge is located on boundary surface,
        try the boundary edge removal operation, and
        set the worst quality of newly tetrahedra as  $e_i$ .
12:    end for
13:    If
14:       $\max\{\max_{i=1,\dots,6}\{f_i\}, \max_{i=1,\dots,6}\{e_i\}\} > Q_\eta$ ,
15:      perform the corresponding operation such that
16:      the worst quality reaches the maximum, and then
17:      update the volume-edge quality of all the new
        tetrahedra.
18:    end for
19:    if No operation are performed in this loop then
20:      break;
21:    end if
22:  end for

```

109

Pipe-Line

La manière dont tout ça s'organise...

Algorithm 2 Quality improvement for the segmented
mesh.

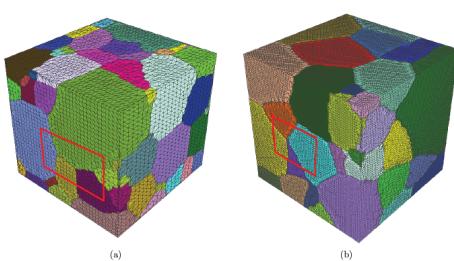
```

1: Interior vertex smoothing.
2: Curve fairing.
3: Interior vertex smoothing.
4: Curve regularization.
5: Interior vertex smoothing.
6: Surface fairing.
7: Interior vertex smoothing.
8: Surface regularization.
9: Boundary and interior vertex smoothing using (3.19).
10: for l = 0, ..., 4 do
11:   TopologicalTransformation(0.4 + l * 0.1, 2).
12:   Boundary and interior vertex smoothing using
       (3.19).
13: end for

```

110

Résultats Des images!

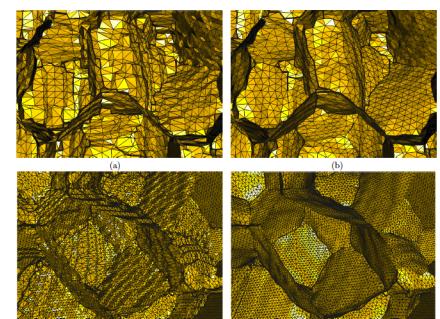


Toutes les images de cette présentation sont tirées de l'article original

Fig 4.2: Segmented tetrahedral meshes for titanium alloy microstructure. (a) 92-grain mesh; (b) 52-grain mesh.

111

Résultats Des images!



Toutes les images de cette présentation sont tirées de l'article original

Fig 4.5: Internal boundary meshes. (a) The initial boundary mesh for 92-grain data; (b) the improved boundary mesh for 92-grain data; (c) the initial boundary mesh for 52-grain data; (b) the improved boundary mesh for 52-grain data.

112

Résultats Des images!

Toutes les images de cette présentation sont tirées de l'article original

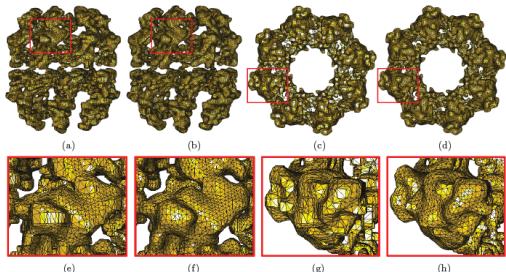


Fig 4.7: (a) Side view of the initial ATcpca mesh; (b) side view of the improved ATcpca mesh; (c) top view of the initial ATcpca mesh; (d) top view of the improved ATcpca mesh; (e)-(h) are enlarged figures of the red windows in (a)-(d), respectively.

113

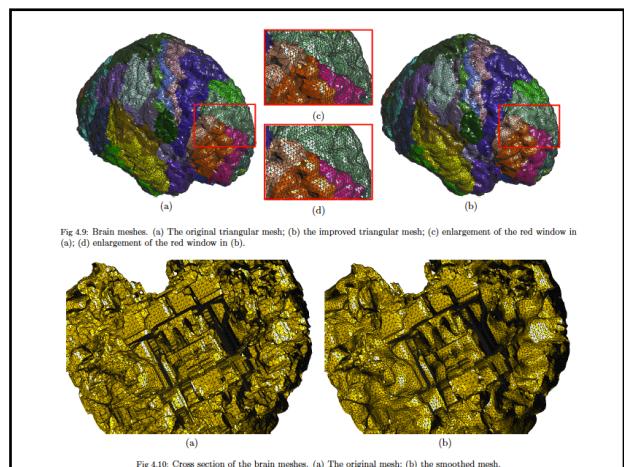


Fig 4.9: Brain meshes. (a) The original triangular mesh; (b) the improved triangular mesh; (c) enlargement of the red window in (a); (d) enlargement of the red window in (b).

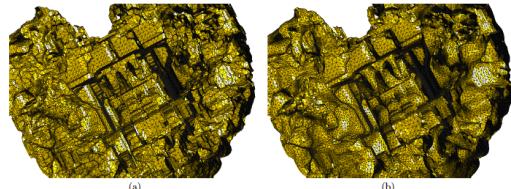


Fig 4.10: Cross section of the brain meshes. (a) The original mesh; (b) the smoothed mesh.

114

10