



Lyon 1

Mesh and Computational Geometry

Raphaëlle Chaine

Université Claude Bernard Lyon 1

Master Informatique
ID3D
Image, Développement
et Technologie 3D



1

Mesh generation

Acute, Alper Ungor



- Fill a 2D shape with quality triangles (resp. tetraedra in 3D)

54

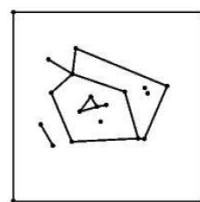
2D Constrained triangulation

- 1rst case :
 - No additional vertex

55

Mesh generation

- Ill posed problem :
 - Given a Planar-Straight-Line Graph (PSLG)
 - Find a triangulation that complies with this PSLG



Ruppert

56

Triangulation of 2D points

- What if the Delaunay triangulation does not comply with the boundaries of the domain or a network of constraints?
- Constraint : a segment to be inserted into the triangulation

57

2D Constrained triangulation

- 1rst case :
 - No additional vertex
 - Slight local degradations of Delaunay are allowed

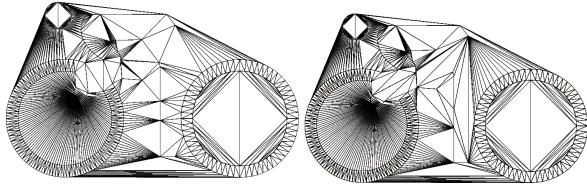
58

57

58

2D Constrained triangulation

- Unconstrained Delaunay Triangulation
- Constrained Triangulation Delaunay

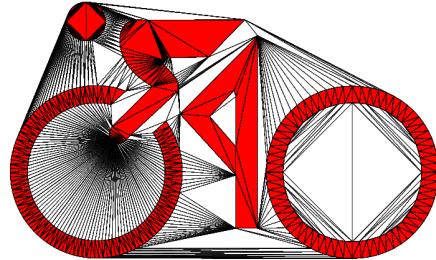


Images by B. Kornberger

59

Case of a digital 2D or 2D½ model

- Constrained triangulation



60

Constrained triangulation of 2D points

- What if the Delaunay triangulation does not comply the boundaries of the domain or a network of constraints?
- We insert the constraints into the triangulation
 - Using the *flip* operation
 - Flip of edges intersecting the constraint edges
 - Orient the constraint edges to set an order for the flip of the intersecting edges

61

Constrained triangulation of 2D points

- What about the non constrained edges?
 - Slightly modify the (locally)-Delaunay Criterion
 - A constraint is seen as a wall stopping the visibility
 - A vertex can be located inside a circumscribed circle if it is not visible from inside the triangle
 - Flip any un-constrained edge that is non locally Delaunay

62

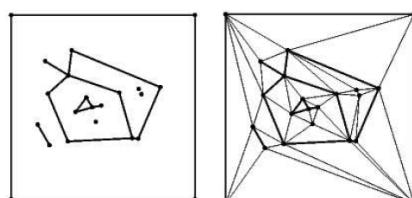
Incremental Constrained triangulation of 2D points

- Incremental insertion of a point in a constrained 2D triangulation
 - Constrained edges cannot be flipped
 - Slightly modify the (locally)-Delaunay Criterion
 - A constraint is seen as a wall stopping the visibility
 - A vertex can be located inside a circumscribed circle if it is not visible from inside the triangle

63

Mesh generation

- Ill posed problem :
 - Given a Planar-Straight-Line Graph (PSLG)
 - Find a triangulation that complies with this PSLG



Delaunay constrained triangulation with no vertex addition.
Generation of small angles ...

64

63

64

2D Constrained triangulation

- 2nd case :

- Keep a Delaunay triangulation
- Insert additional vertices (Steiner points)

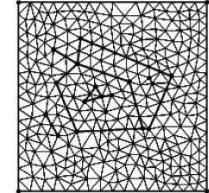
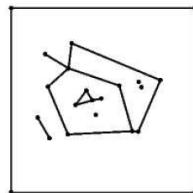
65

65

Génération de maillages

- Problème posé :

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



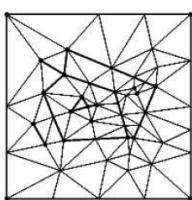
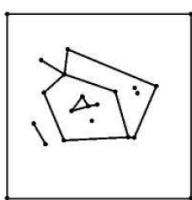
Un exemple de triangulation uniforme satisfaisant le problème : des angles > $22,5^\circ$ mais triangles trop nombreux⁶⁶

66

Génération de maillages

- Problème posé :

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



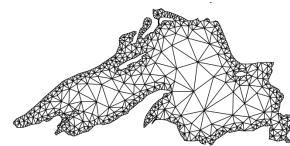
Triangulation obtenue par raffinement de Delaunay : un nombre raisonnable de triangles pour des angles > 20°

67

67

Génération de maillages

- Par raffinement de Delaunay



Tous les angles des triangles sont ici > 25°

68

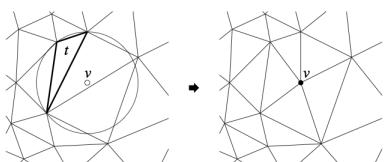
68

Génération de maillages

- Par raffinement de Delaunay

- 2 principes :

- Insertions sur les arêtes contraintes
- Insertions aux centre du cercle circonscrit des triangles mal formés



69

69

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 CONTRAINTES?

70

70

Réfléchissons ensemble...

- Principe : insérer le centre circonscrit des triangles mal formés ...
 - Sauf si cela casse une arête de contrainte (par exemple quand le centre et le triangle ne sont pas du même côté de la contrainte).
 - Dans ce cas, on coupe en deux la contrainte à la place.
- En pratique :
 - Subdiviser la contrainte lorsque un point à ajouter tombe trop près de l'arête (ie. dans son cercle diamétral)

71

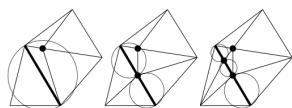
Génération de maillages

- Attention : Stratégie mise en œuvre pour les triangles mal formés uniquement
 - 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 :
 - 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

72

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 ;-)



73

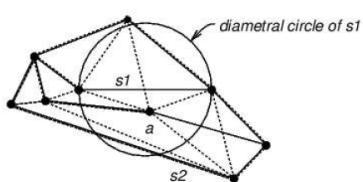
Génération de maillages

- 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)

74

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



75

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

76

75

```

Algorithm DelaunayRefine
INPUT: planar straightline graph  $X$ ;
       desired minimum angle bound  $\alpha$ .
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  $DT(V)$ 

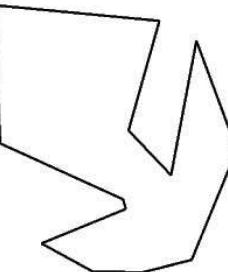
repeat:
  while any segment  $s$  is encroached upon:
    SplitSeg( $s$ )
  let  $t$  be (any) skinny triangle ( $\min \text{angle} < \alpha$ )
  let  $p$  be  $t$ 's circumcenter
  if  $p$  encroaches upon any segments  $s_1, \dots, s_k$  then
    for  $i = 1$  to  $k$ :
      SplitSeg( $s_i$ )
    else
      SplitTri( $t$ ) (* adds  $p$  to  $V$  *)
  endif
until no segments encroached upon, and no angles  $< \alpha$ 
output current Delaunay triangulation  $DT(V)$ 

```

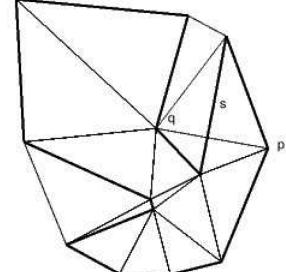
77

Algorithme de Ruppert (Version faible)

- Pas à pas :



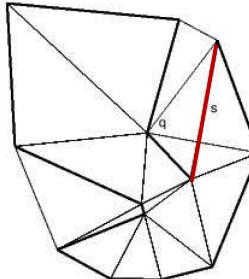
Polygone en entrée

Le segment s n'est pas dans Delaunay

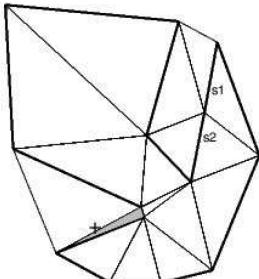
78

Algorithme de Ruppert

- Pas à pas :



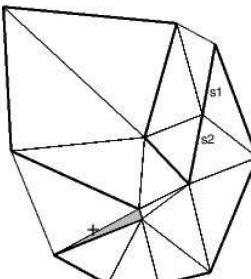
Delaunay

Subdivision de s

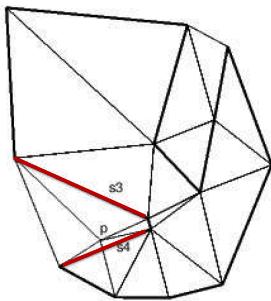
79

Algorithme de Ruppert

- Pas à pas :



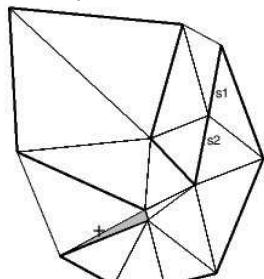
Triangle le plus mal formé (en gris)

Mais l'insertion de son cercle circonscrit accrocherait s_3 et s_4

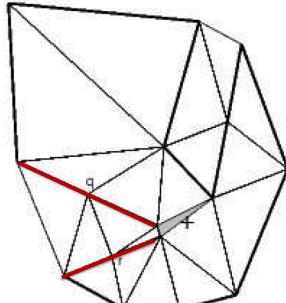
80

Algorithme de Ruppert

- Pas à pas :



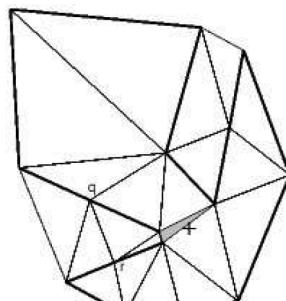
Triangle le plus mal formé (en gris)

Du coup on coupe s_3 et s_4

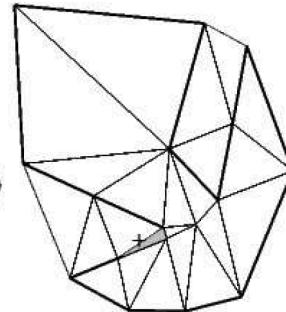
81

Algorithme de Ruppert

- Pas à pas :



Triangle le plus mal formé (en gris)



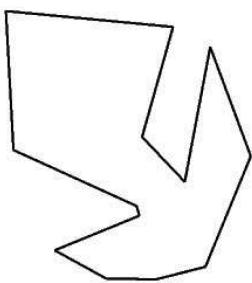
Insertion du centre du plus mauvais triangle précédent

82

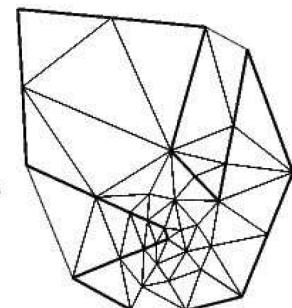
81

Algorithme de Ruppert

- Pas à pas :



Polygone en entrée

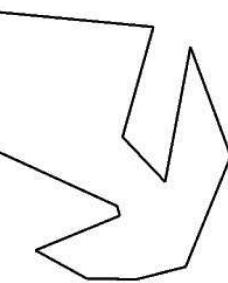


Résultat pour un angle minimum autorisé correspondant à 25° ⁸³

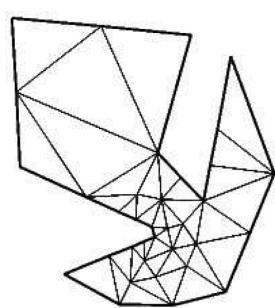
83

Algorithme de Ruppert

- Pas à pas :



Polygone en entrée



Suppression des triangles extérieurs ⁸⁴

84

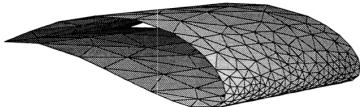
Algorithme de Ruppert

- Terminaison de l'algorithme
 - Preuve basée sur la notion de *local feature size*
- Penser à protéger les petits angles entre segments de contraintes

85

Génération de maillages

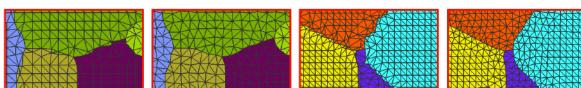
- Adaptation du même principe aux surfaces



Chew

86

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
Chinese Academy of Sciences, Beijing
Carnegie Mellon University

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

87

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*

88

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
 - Optimisation d'une fonctionnelle mesurant la qualité des tétraèdres

89

Prise en compte des frontières

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

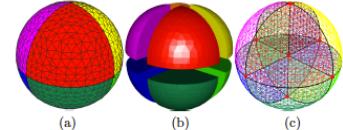


Fig 2.1: (a) A segmented tetrahedral mesh; (b) the object is segmented into eight components; (c) the boundary mesh of (a).

90

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

91

Traitements spécifiques

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

92

Traitements spécifiques

- 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) 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

93

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

$$t_i = \frac{x_i - x_{i-1}}{\|x_i - x_{i-1}\|},$$

$$s_i = \frac{\|x_i - x_{i-1}\| + \|x_i - x_{i+1}\|}{2}$$

$$\kappa_i = \|\kappa_i\|, \quad \kappa_i = \frac{t_{i+1} - t_i}{s_i}, \quad n_i = \frac{\kappa_i}{\kappa_i},$$

94

Curve diffusion flow

– Diffusion de courbe :

- Déplacement des points dans la direction normale

$$\frac{dx_i}{dt} = -(\Delta \kappa_i) \mathbf{n}_i, \quad i = 1, \dots, n-1,$$

• Estimation du Laplacien

$$\left(\frac{\kappa_{i+1} - \kappa_i}{\|x_{i+1} - x_i\|} - \frac{\kappa_i - \kappa_{i-1}}{\|x_i - x_{i-1}\|} \right) / s_i$$

• Schéma explicite d'Euler

$$x_i^{(k+1)} = x_i^{(k)} - \tau \Delta \kappa_i \mathbf{n}_i, \quad i = 1, \dots, n-1,$$

– Critère d'arrêt?

95

Régularisation de la courbe

– Diffusion de courbe :

- Déplacement des points dans la direction normale

$$\frac{dx_i}{dt} = -(\Delta \kappa_i) \mathbf{n}_i, \quad i = 1, \dots, n-1,$$

– Régularisation de la courbe : favoriser une équirépartition des sommets sur la courbe

- Relaxation tangentielle visant à minimiser une fonctionnelle

$$\mathcal{E}(\mathcal{C}) = \frac{1}{2} \sum_{i=1}^n (\|x_i - x_{i-1}\| - h)^2,$$

h longueur moyenne des arêtes de la courbe

96

Régularisation de la courbe

– Régularisation de la courbe : favoriser une équirépartition des sommets sur la courbe

- Relaxation tangentielle visant à minimiser une fonctionnelle

$$\mathcal{E}(\mathcal{C}) = \frac{1}{2} \sum_{i=1}^n (\|x_i - x_{i-1}\| - h)^2,$$

• Dérivée partielle pour un petit déplacement dans une direction

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

$$\delta(\mathcal{E}, \Phi_i) = \frac{\partial \mathcal{E}(\mathcal{C}, \epsilon_i)}{\partial \epsilon_i} \Big|_{\epsilon_i=0} = (\|x_{i+1} - x_i\| - h) \frac{\Phi_i^T (x_i - x_{i+1})}{\|x_i - x_{i+1}\|} + (\|x_i - x_{i-1}\| - h) \frac{\Phi_i^T (x_i - x_{i-1})}{\|x_i - x_{i-1}\|}$$

97

$$\begin{aligned} & \frac{\partial}{\partial \epsilon_i} \left(\frac{1}{2} \left(\|x_i - x_{i-1}\| - h \right)^2 \right) \\ &= \cancel{\frac{\partial}{\partial \epsilon_i} \left(\|x_i - x_{i-1}\| - h \right)} \times \frac{\partial}{\partial \epsilon_i} \left(\|x_i - x_{i-1}\| \right) \\ & \text{comment évaluer } \cancel{\frac{\partial}{\partial \epsilon_i} \left(\|x_i - x_{i-1}\| - h \right)} ? \\ & \text{En utilisant déjà } \frac{\partial}{\partial \epsilon_i} \left(\|x_i(\epsilon_i) - x_{i-1}\| \right) ? \\ & \text{Expression ne faisant pas intervenir de norme} \\ &= 2(x_i - x_{i-1}) \bullet \frac{\partial}{\partial \epsilon_i} (x_i(\epsilon_i)) \\ &= 2(x_i - x_{i-1}) \bullet \frac{\partial}{\partial \epsilon_i} (x_i + \epsilon_i \Phi_i) \\ &= 2(x_i - x_{i-1}) \bullet \Phi_i \\ & \text{Avec norme} = 2\|x_i - x_{i-1}\| \times \frac{\partial}{\partial \epsilon_i} (\|x_i(\epsilon_i)\|) \\ & \text{Donc } \frac{\partial}{\partial \epsilon_i} \left(\|x_i(\epsilon_i) - x_{i-1}\| \right) = \frac{\Phi_i \cdot (x_i - x_{i-1})}{\|x_i - x_{i-1}\|} \end{aligned}$$

98

Régularisation de la courbe

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

• Dérivée partielle $\delta(\mathcal{E}, \Phi_i)$ pour un petit déplacement dans une direction

- On choisit ici la direction tangentielle $\Phi_i = e_i$
- Descente du gradient

$$\frac{dx_i}{dt} + \delta(\mathcal{E}, \Phi_i) e_i = 0.$$

Soit, après discréétisation

$$\begin{aligned} & \frac{x_i^{(k+1)} - x_i^{(k)}}{\tau} + (\|x_{i+1}^{(k)} - x_i^{(k)}\| - h) \frac{e_i e_i^T (x_i^{(k)} - x_{i+1}^{(k)})}{\|x_i^{(k)} - x_{i+1}^{(k)}\|} \\ & + (\|x_i^{(k)} - x_{i-1}^{(k)}\| - h) \frac{e_i e_i^T (x_i^{(k)} - x_{i-1}^{(k)})}{\|x_i^{(k)} - x_{i-1}^{(k)}\|} \\ & = 0. \end{aligned}$$

99

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

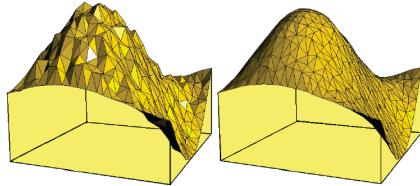
100

Averaged mean curvature flow

- Diffusion des surfaces, H courbure moyenne

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

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



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

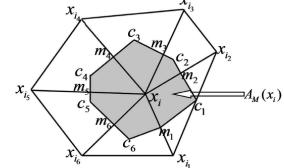


Fig 3.4: Area represented by \mathbf{x}_i . $\{m_j\}_{j=1}^6$ are midpoints of edges $[\mathbf{x}_i \mathbf{x}_j]$. c_j is the circumcenter point for the triangle $[\mathbf{x}_i \mathbf{x}_{j+1} \mathbf{x}_l]$ if the triangle is non-obtuse; if the triangle is obtuse, c_j is chosen to be the midpoint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

101

102

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
- $h = \left(\frac{4}{\sqrt{3}}A\right)^{\frac{1}{2}}$
A aire moyenne des triangles
- Dérivée partielle
 - pour un petit déplacement dans une direction Φ_i
- $$\delta(\mathcal{E}, \Phi_i) = \frac{\partial \mathcal{E}(\mathcal{S}, \epsilon_i)}{\partial \epsilon_i} \Big|_{\epsilon_i=0} = \sum_{j \in N(i)} (\|\mathbf{x}_j - \mathbf{x}_i\| - h) \frac{\Phi_i^T(\mathbf{x}_i - \mathbf{x}_j)}{\|\mathbf{x}_i - \mathbf{x}_j\|}$$

103

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

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

104

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 discréttisation

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

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

- Schéma explicite avec test de non réversion de tétraèdre!

105

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

106

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

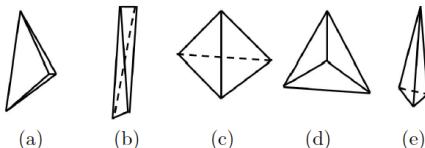


Fig 2.2: Examples of poorly-shaped elements.

107

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

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

108

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

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

109

Amélioration de la qualité de tétraèdres

- Descente du gradient

- Pour les points intérieurs

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

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

- Pour les sommets de surface

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

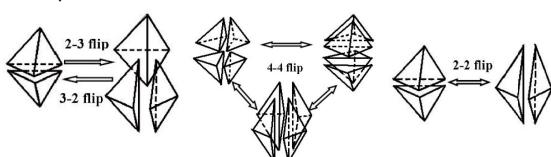
where $\mathbf{e}_i^{(1)}$ and $\mathbf{e}_i^{(2)}$ are orthogonal tangential directions at \mathbf{x}_i .

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

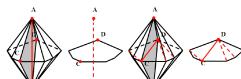
110

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



111

Algorithme

Algorithm 1: TopologicalTransformation(ε, L)

```

1: for  $i=1:L$  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  $i$ -th edge is interior edge, try remove it using
       the 3-2 flip or 4-4 flip operation. If succeed,
       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 located on boundary surface,
        try the boundary edge removal operation, and
        set the worst quality of newly tetrahedra as  $e_i$ .
11:    end for
12:    If
13:       $\max\{\max_{i=1,\dots,4}\{f_i\}, \max_{i=1,\dots,6}\{e_i\}\} > Q_\eta$ ,
        perform the corresponding operation such that
        the worst quality reaches the maximum, and then
        update the volume-edge quality of all the new
        tetrahedra.
14:    end for
15:    if No operation are performed in this loop then
16:      break;
17:    end if
18:  end for

```

112

10

Pipe-Line

La manière dont tout ca 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  $i = 0, \dots, 4$  do
11:   TopologicalTransformation( $0.4 + i * 0.1, 2$ ).
12:   Boundary and interior vertex smoothing using (3.19).
13: end for
```

113

Résultats Des images!

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

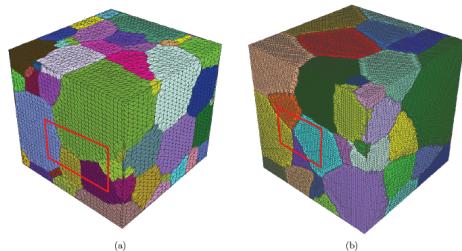


Fig 4.2: Segmented tetrahedral meshes for titanium alloy microstructure. (a) 92-grain mesh; (b) 52-grain mesh.

114

Résultats Des images!

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

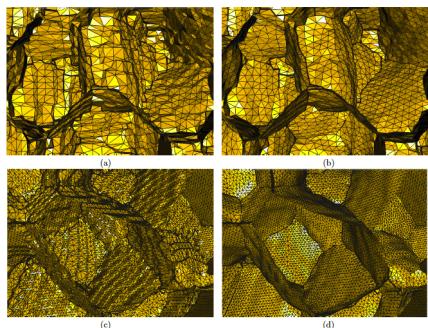


Fig 4.5: Internal boundary meshes. (a) The initial boundary mesh for 92-grain data; (b) the improved boundary mesh for 92-grain data; (c) the initial boundary mesh for 52-grain data; (d) the improved boundary mesh for 52-grain data.

115

Résultats Des images!

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

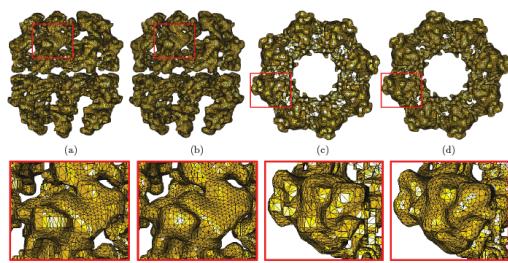


Fig 4.7: (a) Side view of the initial ATepmc mesh; (b) side view of the improved ATepmc mesh; (c) top view of the initial ATepmc mesh; (d) top view of the improved ATepmc mesh; (e)-(h) are enlarged figures of the red windows in (a)-(d), respectively.

116

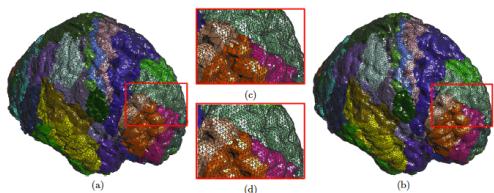


Fig 4.9: Brain meshes. (a) The original triangular mesh; (b) the improved triangular mesh; (c) enlargement of the red window in (a); (d) enlargement of the red window in (b).

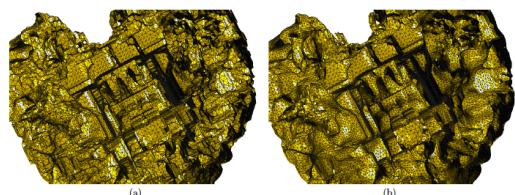


Fig 4.10: Cross section of the brain meshes. (a) The original mesh; (b) the smoothed mesh.

117