

# Projets du cours de Méthodes d'adaptation pour la simulation numérique (NM491)

---

## Remarques préliminaires :

- Ces sujets sont volontairement formulés en des termes vagues. La raison en est que leur objectif peut être atteint par de très nombreuses approches, aux avantages et inconvénients différents. N'hésitez surtout pas à sortir des sentiers battus !
  - Bien entendu, ces sujets ne sont pas déconnectés du contenu du cours et des séances de TP. Aussi est-il conseillé d'attendre que les trois ou quatre premières séances soient passées pour avoir une bonne vision des différents outils qui pourraient être employés, avant de commencer à s'«enfermer» dans une technique de résolution. De la même manière, n'hésitez pas à consulter la bibliographie indicative (bien loin d'être exhaustive) en fin de document pour vous inspirer.
  - Il est demandé de rédiger un rapport de dix à quinze pages, qui servira de base à l'évaluation du module. Dans celui-ci figureront notamment une présentation détaillée du sujet, de ou des approches que l'on aura jugées intéressantes, ainsi que d'exemples numériques appropriés (on s'abstiendra de tout listing de code).
- 

## Projet n°1 : un algorithme de triangulation en 2d pour la compression d'image

Dans tout ce projet, afin d'éviter les inévitables problèmes liés aux erreurs d'arrondis et puisque l'on est intéressé par la modélisation d'images, on ne travaille qu'en coordonnées entières : les abscisses (respectivement les ordonnées) des points considérés appartiendront à l'ensemble discret  $[[1, m]]$  (respectivement  $[[1, n]]$ ),  $m$  et  $n$  désignant la taille de l'image en termes de nombre de pixels.

Étant donnés deux tels entiers  $m, n \in \mathbb{N}$ , une *image* (en niveaux de gris) sur cet ensemble de  $mn$  pixels est formellement une fonction d'intensité lumineuse  $I : [[1, m]] \times [[1, n]] \rightarrow [[0, ng]]$ , où  $ng + 1$  désigne le nombre de niveaux de gris (pour les images considérées  $ng = 255$ ).

En pratique, le nombre de pixels à manipuler est très grand ; le domaine de la compression d'images étudie les moyens de stocker une telle image en ne conservant qu'un minimum d'informations (par exemple, mais pas seulement, des valeurs d'intensité en des points 'bien choisis'), à partir desquelles l'image initiale  $I : [[1, m]] \times [[1, n]] \rightarrow [[0, ng]]$  peut être reconstruite sous forme d'une image  $\tilde{I} : [[1, m]] \times [[1, n]] \rightarrow [[0, ng]]$  aussi fidèlement que possible. Ici, on s'intéresse plus particulièrement à un modèle de compression avec perte (tel que le format JPEG, et au contraire du format GIF, par exemple), c'est-à-dire que l'on s'attend seulement à ce que la reconstruction de l'image soit une 'bonne' approximation de celle-ci.

Il existe bien des manières d'apprécier la qualité de la reconstruction au regard de l'image initiale. La plus courante est de mesurer le *rapport signal-sur-bruit*, - *Peak Signal-to-Noise Ratio* (PSNR) en anglais - qui grosso modo quantifie l'écart en échelle logarithmique entre la plus grande valeur possible  $ng$  pour le signal, et l'erreur quadratique moyenne *MSE* commise par la reconstruction :

$$PSNR = 10 \log_{10} \left( \frac{ng^2}{MSE} \right) \quad , \quad MSE = \sum_{i=1}^m \sum_{j=1}^n \left( I(i, j) - \tilde{I}(i, j) \right)^2 .$$

Ici, on propose de conserver une image  $I$  sous forme d'une triangulation conforme  $\mathcal{T}$  du carré  $[[1, m]] \times [[1, n]]$ , en ne retenant les valeurs d'intensité  $I(i, j)$  qu'aux seuls nœuds de  $\mathcal{T}$ . La fonction d'intensité reconstruite  $\tilde{I}$  est alors obtenue comme la fonction d'éléments finis de Lagrange  $\mathbb{P}^1$  (ou  $\mathbb{P}^0$  si l'on veut comparer) à partir de ces valeurs aux nœuds de la triangulation.

L'objectif du projet est de concevoir et implémenter une méthode qui permette, à partir d'une image  $I$ , donnée sous forme non compressée (i.e. dont on connaît la valeur d'intensité lumineuse à chaque pixel), et d'une valeur de tolérance  $\epsilon$ , de générer une triangulation  $\mathcal{T}_\epsilon$ , aussi dépouillée que possible, et dont la reconstruction associée jouit d'une valeur de  $PSNR$  (ou toute autre mesure d'erreur que l'on jugera pertinente) inférieure à  $\epsilon$ .

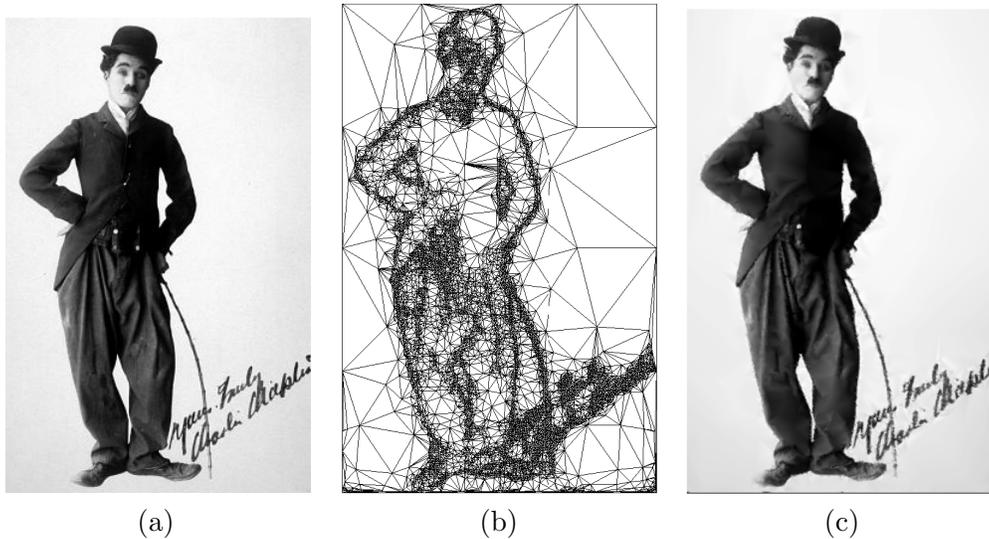


FIGURE 1. (a) Image initiale (101490 pixels), (b) une triangulation ‘optimale’ pour stocker l’information essentielle de l’image, (c) l’interpolé  $\mathbb{P}^1$  de la fonction d’intensité de l’image sur cette triangulation.

### Projet n°2 : un algorithme de triangulation en 2d pour la simplification de cartes

Dans ce projet également, on travaille en coordonnées entières : les abscisses (respectivement les ordonnées) des points considérés appartiennent à l’ensemble discret  $[[1, m]]$  (respectivement  $[[1, n]]$ ),  $m$  et  $n$  étant liés à la résolution de la carte considérée.

Une *carte* sur cet ensemble  $[[1, m]] \times [[1, n]]$  est une fonction  $f : [[1, m]] \times [[1, n]] \rightarrow \mathbb{R}$ . Il s’agit donc simplement du graphe d’une fonction de deux variables, et l’on s’intéresse au cas où ce graphe décrit un modèle géométrique de terrain : pour tout point  $(x, y)$  du plan,  $f(x, y)$  est l’altitude du terrain.

L’information initiale dont on dispose quant à ce terrain est encodée sous la forme d’une liste des  $mn$  valeurs d’altitude aux points d’échantillonnage régulier  $(x, y) \in [[1, m]] \times [[1, n]]$ . On peut, de manière équivalente voir ces points comme les sommets d’une grille cartésienne de taille  $m \times n$ , ou encore comme les noeuds d’une triangulation  $\mathcal{T}_0$  qui s’en déduit en divisant chaque cellule en deux triangles.

Comme dans le cas du premier projet, un tel échantillonnage des relevés d'altitude est en général beaucoup trop exhaustif pour permettre un stockage et une exploitation efficaces. On souhaite ne garder qu'un nombre limité de ces valeurs, à partir desquelles une reconstruction  $\tilde{f} : [[1, m]] \times [[1, n]] \rightarrow \mathbb{R}$  la plus 'fidèle' possible de la fonction  $f$  puisse être inférée.

Ici encore, on propose pour ce faire de conserver cette carte sous la forme d'une triangulation conforme  $\mathcal{T}$  du carré  $[[1, m]] \times [[1, n]]$ , obtenue à partir de la 'grosse' triangulation  $\mathcal{T}_0$ , et des seules valeurs de l'altitude aux nœuds de cette triangulation. On en déduira alors la fonction d'altitude reconstruite  $\tilde{f}$  comme la fonction d'éléments finis de Lagrange  $\mathbb{P}^1$  sur le maillage  $\mathcal{T}$ , qui prend les valeurs stockées aux nœuds de  $\mathcal{T}$ .

Étant donné une fonction d'altitude 'exhaustive'  $f : [[1, m]] \times [[1, n]] \rightarrow \mathbb{R}$ , l'objectif visé par ce projet est d'obtenir une triangulation  $\mathcal{T}$  de l'espace des paramètres, minimale en termes de nombre de sommets / triangles, et telle que l'écart entre la fonction d'altitude initiale  $f$  et sa reconstruction  $\tilde{f}$  soit contrôlé, en un sens à définir.

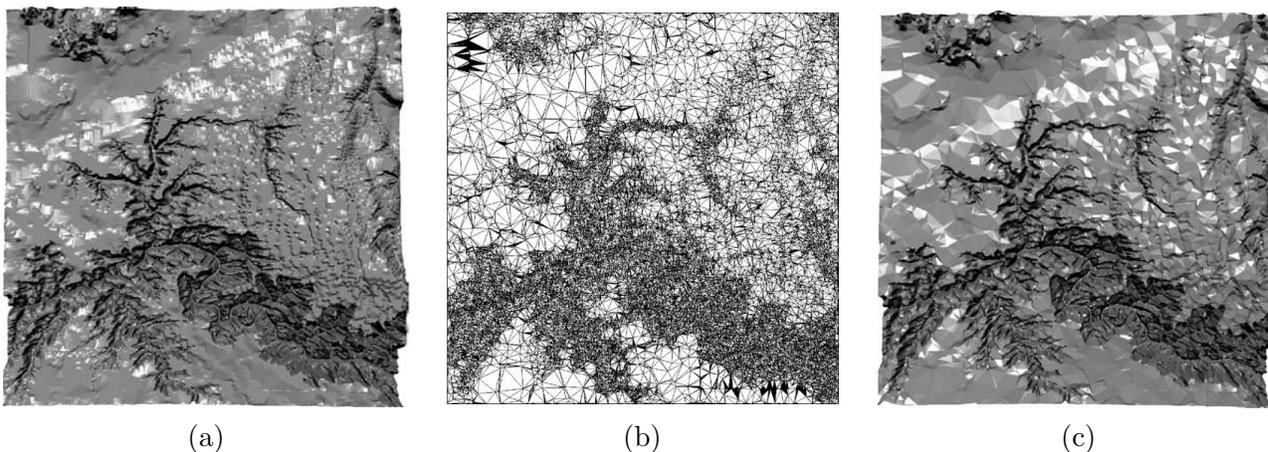


FIGURE 2. (a) La carte initiale du Grand Canyon (l'échantillonnage comprend  $1201 \times 1201$  points), (b) une triangulation 'optimale' pour stocker la carte (environ 30000 nœuds), (c) la reconstruction  $\tilde{f}$  de la carte sur cette triangulation.

### Projet n°3 : Une technique d'adaptation (isotrope) de maillage en 2d

Dans ce projet, contrairement aux deux précédents, on travaillera avec des valeurs de coordonnées flottantes.

L'objectif général de l'adaptation de maillage est de modifier un maillage sur lequel se déroule un calcul, de sorte à ce que le résultat obtenu soit une bonne approximation de la quantité recherchée, pour un calcul se déroulant en un temps minimal (ce qui, le plus souvent, revient à exiger que la taille du maillage, en termes de nombre de sommets / éléments ne soit pas trop importante).

Comme cela a été, ou sera, vu en cours, un concept récurrent en adaptation de maillage est celui de *métrique Riemannienne*. L'adaptation *isotrope* d'un maillage initial  $\mathcal{T}_0$  part de la connaissance, en tout point  $x$  de l'espace, d'une fonction de taille  $h(x)$  qui spécifie la taille des entités du maillage,

localement autour de  $x$ , et vise à rendre toutes ces entités conformes à la métrique induite par cette carte de taille (voir le cours) : les arêtes doivent être de longueur quasi-unité, i.e. comprise entre  $\frac{1}{\sqrt{2}}$  et  $\sqrt{2}$ ...

De nombreuses techniques sont possibles pour définir cette fonction de taille : lorsqu'il s'agit d'adapter le maillage à un calcul par éléments finis par exemple, les techniques les plus fines - celles qui procèdent par des *estimations a posteriori* - demandent une connaissance intime de l'équation étudiée ainsi que de sa discrétisation. D'autres méthodes, moins précises mais plus génériques reposent sur des estimations *a priori*, qui très souvent, mettent en jeu l'erreur d'interpolation d'une fonction 'régulière' sur le maillage considéré (penser au lemme de Céa). C'est sur ce dernier point que l'on choisit ici de se focaliser.

L'objectif de ce projet est double : il s'agit dans un premier temps de concevoir et implémenter un algorithme d'adaptation (isotrope) d'un maillage initial  $\mathcal{T}_0$  à une fonction de taille  $h(x)$ . Dans un second temps, on utilisera cet algorithme pour adapter un tel maillage de sorte qu'il permette un contrôle sur l'erreur d'interpolation par éléments finis de Lagrange  $\mathbb{P}^1$  d'une fonction régulière  $u$ , définie analytiquement.

Comment s'adaptent les routines mises en œuvre si l'on voulait cette fois-ci adapter le maillage initial  $\mathcal{T}_0$  de manière *anisotrope* ?

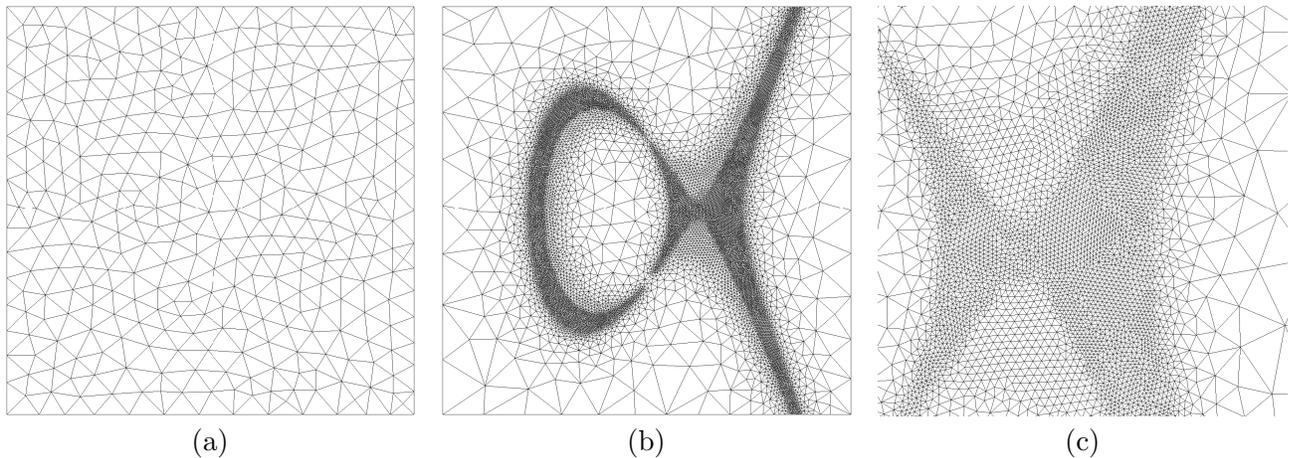


FIGURE 3. (a) Maillage initial (356 points), (b) une triangulation adaptée au contrôle de l'erreur d'interpolation d'une fonction analytique (11540 points), (c) zoom sur ce dernier maillage.

#### Projet n°4 : Génération d'un maillage adapté à une fonction implicite

Dans ce projet également, on travaillera avec des valeurs de coordonnées flottantes.

Comme cela sera évoqué en cours, l'un des aspects les plus importants lors de la mise au point d'une méthode numérique permettant de rendre compte de phénomènes où la géométrie du domaine de calcul évolue - c'est le cas si l'on s'intéresse par exemple à l'évolution d'une interface entre deux fluides, ou entre un solide et un fluide - est la manière dont on choisit de décrire la géométrie de ce domaine.

Dans ce contexte, la *méthode des lignes de niveaux* (*level set method* en anglais, voir par exemple [3]) est de plus en plus utilisée, en vertu de sa souplesse et de sa robustesse.

Grosso modo, il s’agit d’adopter un point de vue ‘implicite’, et de représenter un domaine  $\Omega \subset \mathbb{R}^d$  ( $d = 2$  ou  $3$ ) par la donnée d’une fonction scalaire  $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$  dont  $\Omega$  est le sous-domaine négatif, i.e. :

$$\forall x \in \mathbb{R}^d, \quad \begin{cases} \varphi(x) < 0 & \text{si } x \in \Omega \\ \varphi(x) = 0 & \text{si } x \in \partial\Omega \\ \varphi(x) > 0 & \text{si } x \in \overline{c\Omega} \end{cases} .$$

D’un point de vue numérique, on se donne un ‘gros’ domaine de calcul  $D$  dans lequel toutes les formes que l’on va être amené à considérer seront incluses, ainsi qu’un maillage  $\mathcal{T}$  de  $D$  (qui sera simplicial dans ce qui suit, mais pourrait très bien être cartésien), et on discrétise  $\varphi$  sur  $\mathcal{T}$  (par exemple comme fonction d’éléments finis  $\mathbb{P}^1$  sur  $\mathcal{T}$ ).

L’objectif de ce projet est de faire le lien entre une telle description implicite d’un domaine  $\Omega$ , et une description de  $\Omega$  par la donnée d’un maillage simplicial (qui est celle dont on a besoin lorsqu’il s’agit de réaliser des calculs par éléments finis sur  $\Omega$ ). Soit  $D \subset \mathbb{R}^2$  un ‘gros’ domaine de calcul (par exemple un carré), muni d’un maillage en triangles  $\mathcal{T}$ , et soit  $\varphi$  une fonction d’éléments finis  $\mathbb{P}^1$  sur  $\mathcal{T}$ . Il s’agit dans un premier temps d’écrire et implémenter un algorithme qui ‘casse’ le maillage  $\mathcal{T}$  pour introduire explicitement la discrétisation de l’ensemble de niveau 0 de  $\varphi$  (voir la figure 4, (c-d)). Ceci donne un nouveau maillage triangulaire  $\tilde{\mathcal{T}}$  qui est composé de deux sous-maillages complémentaires : l’un étant celui de  $\Omega$ , l’autre de  $D \setminus \overline{\Omega}$ . Le fait est que ce maillage  $\tilde{\mathcal{T}}$  est de mauvaise qualité en termes d’éléments finis. Il s’agit donc, dans un second temps de concevoir des techniques d’amélioration de la qualité de ce maillage (bouger de points, ‘collapse’ d’arêtes, etc...), pour le transformer en un maillage  $\mathcal{T}'$  plus ‘propre’ du point de vue du calcul par éléments finis (voir la figure 4, (e-f)).

## REFERENCES

- [1] M. V. ANGLADA, N. P. GARCIA AND P. B. CROSA, *Directional adaptive surface triangulation*, Computer Aided Geometric Design, 16 (1999), pp. 107–126.
- [2] M. DE BERG, O. CHEONG, M. VAN KREVELD AND M. OVERMARS, *Computational Geometry : Algorithms and Applications*, Springer-Verlag, 3rd ed. (2008).
- [3] S. J. OSHER AND R. FEDKIW, *Level Set Methods and Dynamic Implicit Surfaces*, Springer, (2002).
- [4] P. J. FREY AND H. BOROUCAKI, *Simplification des cartes géographiques par minimisation de la déformation locale*, C. R. Acad. Sci. Paris, Ser. I 334 (2002), pp. 227–232.
- [5] S. RIPPA, *Minimal roughness property of the Delaunay triangulation*, Computer Aided Geometric Design, 7 (1990), pp. 489–497.

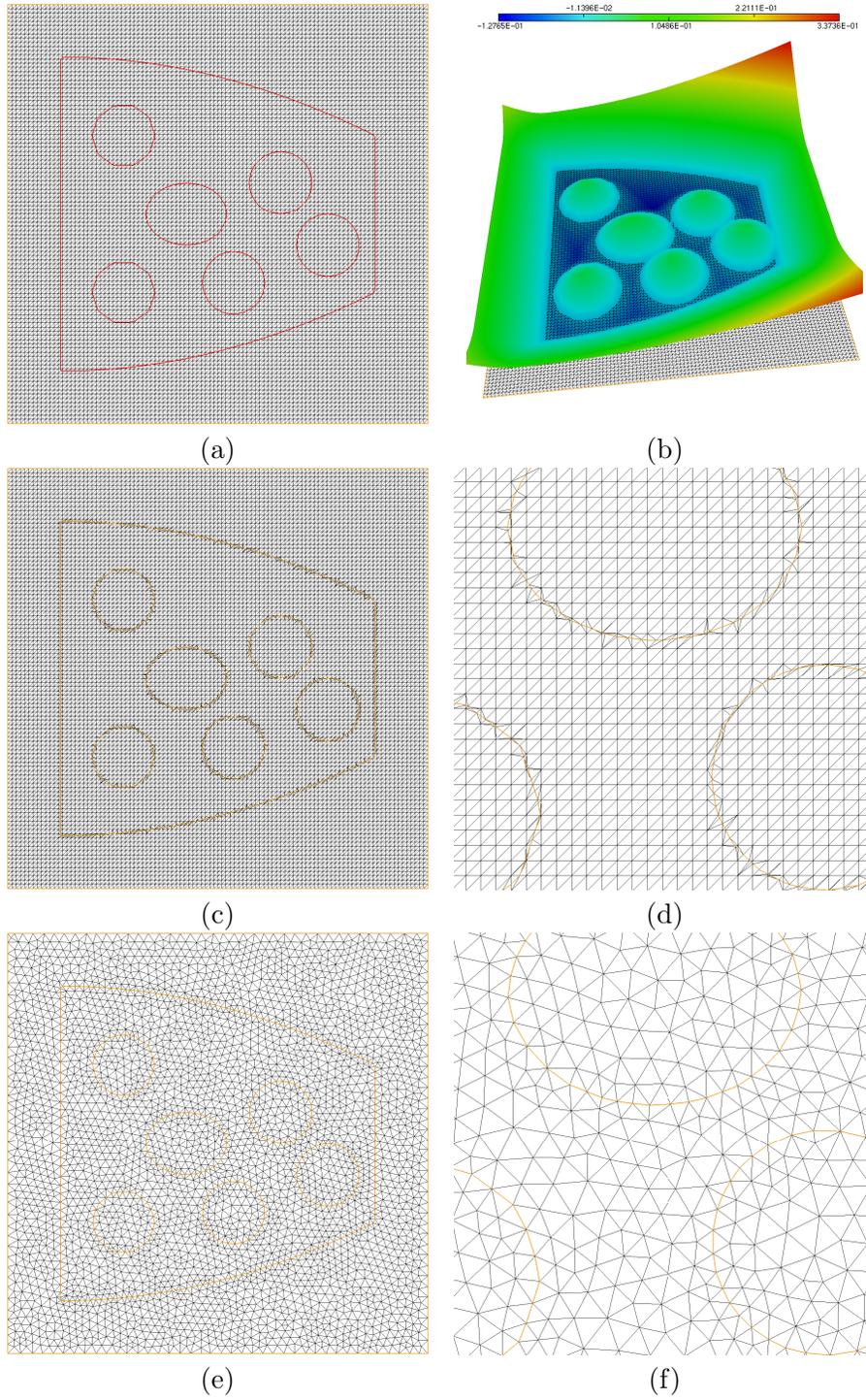


FIGURE 4. (a) Le maillage  $\mathcal{T}$  de  $D$ , et l'isovaleur 0 d'une fonction implicite  $\varphi$  définie comme fonction  $\mathbb{P}^1$  sur  $\mathcal{T}$ , (b) graphe de la fonction  $\varphi$ , (c-d) maillage  $\tilde{\mathcal{T}}$  obtenu en 'cassant' l'isovaleur 0 de  $\varphi$  dans  $\mathcal{T}$ , (e-f), maillage  $\mathcal{T}'$  résultant de l'amélioration de qualité du maillage  $\tilde{\mathcal{T}}$ .