

# Rapport final du stage

Joao Guilherme CALDAS STEINSTRAESSER

28 décembre 2015

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Le modèle</b>	<b>3</b>
2.1	Description du problème . . . . .	3
2.2	Discrétisation en éléments finis . . . . .	4
2.3	Quelques éléments pour le calcul de $K$ . . . . .	5
2.4	Résolution du système linéaire . . . . .	7
<b>3</b>	<b>Application à la fonction Level Set</b>	<b>7</b>
3.1	Construction de la fonction à adapter (avec $\omega$ donné par l'expression (4)) . . . . .	8
3.1.1	Choix des paramètres . . . . .	10
3.2	Calcul des tailles désirées . . . . .	12
3.2.1	Résultats . . . . .	13
3.3	Relaxation et qualité du maillage . . . . .	13
3.4	Adaptation 3D . . . . .	15
<b>4</b>	<b>Application à l'adaptation physique</b>	<b>19</b>
4.1	Calcul des tailles désirées . . . . .	19
4.2	Couplage entre adaptation Level Set et adaptation physique .	20
4.3	Résultats . . . . .	21
<b>5</b>	<b>Adaptation à un cas non stationnaire</b>	<b>25</b>
<b>6</b>	<b>Autres rapports</b>	<b>28</b>

# 1 Intoduction

L'objectif de l'adaptation de maillages développée dans ce stage est de modifier la position de ses noeuds par rapport à une fonction donnée, ou à une solution d'un problème physique, de façon qu'on puisse avoir un raffinement variable au long du domaine et qui représente bien la présence de fortes gradients, surfaces, etc., sans modifier le nombre de points ni la connectivité du maillage.

Dans ce rapport, on va présenter dans un premier moment l'application de l'algorithme à l'adaptation à des fonctions Level Set, à fin d'obtenir une bonne représentation d'un objet. Cette fonction, définie pour tout point du maillage, est la distance signée entre le point et l'objet, où la signe indique s'il est à son intérieur ou extérieur [5]. Ainsi, la ligne de niveau 0 de la fonction Level Set représente la surface de l'objet, et on utilisera ce fait pour orienter l'adaptation du maillage.

Ensuite, on résoudra des problèmes de la mécanique des fluides sur les maillages adaptés et, avec les résultats obtenus, on fera des nouvelles adaptations, mais cette fois-ci en utilisant au même temps la fonction Level Set et la solution physique du problème. Avec cette procédure, on sera capable d'obtenir des maillages encore plus appropriées au calcul envisagé.

Le rapport est organisé de la façon suivante : dans la section 2, on va d'abord présenter, de façon plus générale, le modèle utilisé pour l'adaptation de maillages, selon la formulation développée par [1], et des détails concernant son implémentation. L'application du modèle à l'adaptation à des fonctions Level Set sera décrite dans la section 3, avec quelques exemples de tests réalisés et une indication des paramètres qui ont produit les meilleurs résultats. Le couplage avec l'adaptation physique et les résultats obtenus sont présentés dans la section 4. Enfin, dans la section 5, on utilise le modèle pour adapter le maillage à des objets en mouvement. Par ailleurs, on remarque que la plupart du contenu de ce rapport se réfère à des cas 2D ; pourtant, nous présentons aussi quelques résultats pour des exemples 3D.

Les exemples présentés dans ce rapport ne sont qu'une petite partie de l'ensemble des tests réalisées au cours de ce stage. Ces tests ont eu une grande importance pour valider, corriger et développer le modèle, tester la bibliothèque et les plusieurs parties du code, et trouver les paramètres et stratégies d'adaptation qui nous permettent d'obtenir les meilleurs résultats en tenant compte des objectifs décrits ci-dessus. La section 6 présente une liste des rapports rédigées pour décrire et présenter ces tests.

## 2 Le modèle

### 2.1 Description du problème

On va faire la distinction entre deux domaines :

- **Domaine physique ou réel** ( $\mathbf{x} = (x, y)$ ) : domaine déformable, noté  $\Omega$  ;
- **Domaine computationnel ou de référence** ( $\boldsymbol{\xi} = (\xi, \eta)$ ) : domaine fixé, noté  $\Omega_{\boldsymbol{\xi}}$

On cherche une fonction

$$\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}) \quad (1)$$

Dans le modèle utilisé, on va considérer que la position  $\mathbf{x}$  des noeuds du maillage est régie par l'équation

$$\nabla_{\xi\eta} \cdot (\omega \nabla_{\xi\eta} \mathbf{x}) = 0 \quad (2)$$

où  $\nabla_{\xi\eta}$  est le gradient par rapport aux coordonnées de référence et  $\omega$  est une fonction de  $\mathbf{x}$  qui contient l'information qui déterminera le mouvement des noeuds. Dans le travail développé au cours du stage, on a implémenté deux modèles différents pour le calcul de cette fonction :

1. Dans un première moment, on a implémenté utilisé dans [1] : en supposant qu'on fait l'adaptation par rapport à une fonction  $u$ ,  $\omega$  est donné par l'expression

$$\omega(\mathbf{x}) = \sqrt{1 + \alpha \|\nabla_{\xi\eta} u(\mathbf{x})\| + \beta \|H(u)(\mathbf{x})\|} \quad (3)$$

$H(u)$  est le hessien de  $u$ , et  $\alpha$  et  $\beta$  sont des paramètres choisis par l'utilisateur. Pour que ce choix soit moins dépendant du problème, on va toujours considérer les gradients et le hessiens normalisés.

2. On a ensuite utilisé un modèle où on fournit directement à chaque noeud  $i$  la taille de maille  $h_{des}$  qu'on désire, selon la formulation présentée par [2] :

$$\omega(\mathbf{x}) = \frac{1}{h_{des}(\mathbf{x})} \quad (4)$$

La façon dont on calcule les tailles désirées dépend du type d'adaptation qu'on fait (adaptation à une fonction Level Set ou à une solution physique), comme on précisera dans les sections suivantes de ce rapport.

Pour que le problème soit bien posé, il faut définir des conditions aux bords appropriées :

$$\begin{cases} \nabla_{\xi\eta} \cdot (\omega \nabla_{\xi\eta} \mathbf{x}) = 0 & \text{dans } \Omega_{\boldsymbol{\xi}} \\ \mathbf{x} = \mathbf{g} & \text{sur } \partial\Omega_{\boldsymbol{\xi}}^D \\ \nabla_{\xi\eta} \mathbf{x} \cdot \mathbf{n} = 0 & \text{sur } \partial\Omega_{\boldsymbol{\xi}}^N \end{cases} \quad (5)$$

Ainsi, les conditions aux limites utilisées sont de deux types, de Dirichlet et de Neumann, définies sur des parties disjointes du bord, ( $\partial\Omega_{\boldsymbol{\xi}}^D$  et  $\partial\Omega_{\boldsymbol{\xi}}^N$ , respectivement). Pour les conditions de Dirichlet, on impose  $\mathbf{g} = \boldsymbol{\xi}$ , indiquant que les points de  $\partial\Omega_{\boldsymbol{\xi}}^D$  ne doivent pas bouger (ce qu'on impose, par exemple, dans les coins d'un domaine rectangulaire). En revanche, les conditions de Neumann (imposées par exemple dans les côtés du domaine), indiquent que les points de  $\partial\Omega_{\boldsymbol{\xi}}^N$  doivent glisser sur le bord, i.e., bouger parallèlement au bord (de façon que, dans notre exemple, le domaine reste toujours rectangulaire).

La formulation faible du problème, avec une fonction test  $v \in H_1^0(\Omega_{\boldsymbol{\xi}})$ , s'écrit comme

$$0 = \int_{\Omega_{\boldsymbol{\xi}}} v \nabla_{\xi\eta} \cdot (\omega \nabla_{\xi\eta} \mathbf{x}) d\boldsymbol{\xi} = - \int_{\Omega_{\boldsymbol{\xi}}} \omega \nabla_{\xi\eta} v \cdot \nabla_{\xi\eta} \mathbf{x} d\boldsymbol{\xi} + \int_{\partial\Omega_{\boldsymbol{\xi}}} v \omega \nabla_{\xi\eta} \mathbf{x} \cdot \mathbf{n} ds \quad (6)$$

Les conditions aux bords définies en (5) annulent le dernier terme en (6), et on arrive ainsi à

$$\int_{\Omega_{\boldsymbol{\xi}}} \omega \nabla_{\xi\eta} v \cdot \nabla_{\xi\eta} \mathbf{x} d\boldsymbol{\xi} = 0 \quad (7)$$

## 2.2 Discrétisation en éléments finis

On utilise une discrétisation en élément finis P1, avec une base de fonctions  $\{\varphi_i\}$  définis pour chacun des  $N$  noeuds du maillage. Ainsi,  $\mathbf{x}$  et la fonction test  $v \in H_0^1(\Omega)$  se discrétisent sous la forme

$$\begin{aligned} \mathbf{x}_h &= \sum_{j=1}^N \mathbf{x}_j \varphi_j = \sum_{j=1}^N \begin{pmatrix} x_j \\ y_j \end{pmatrix} \varphi_j \\ v_h &= \sum_{i=1}^N v_i \varphi_i \end{aligned} \quad (8)$$

En utilisant (8) dans (6), on arrive à

$$\sum_{j=1}^N \left[ \left( \int_{\Omega_\xi^h} \omega \nabla_{\xi\eta} \varphi_i \cdot \nabla_{\xi\eta} \varphi_j d\xi \right) \mathbf{x}_j \right] = 0 \quad \forall i \in \{1, \dots, N\} \quad (9)$$

On voit ainsi que la discréétisation en éléments finis se ramène à la résolution de deux systèmes linéaires indépendants et de la même forme, un pour les coefficients  $\{x_j\}$  et l'autre pour  $\{y_j\}$  :

$$\begin{cases} Kx = 0 \\ Ky = 0 \end{cases} \quad (10)$$

Les éléments de la matrice  $K$  ont la forme

$$k_{ij} = \int_{\Omega_\xi^h} \omega \nabla_{\xi\eta} \varphi_i \cdot \nabla_{\xi\eta} \varphi_j d\xi \quad (11)$$

### 2.3 Quelques éléments pour le calcul de $K$

Le calcul des éléments de  $K$  est fait de la manière usuelle, par assemblage des contributions des éléments pour les coefficients  $k_{ij}$ . On précise dans la liste suivante quelques détails de l'implémentation de ce calcul :

- Le gradient de  $\varphi_i$  sur l'élément  $T$  sera donnée par  $(\nabla_{\xi\eta} \varphi_i)^T = \frac{\mathbf{n}_i^T}{d|T|}$ , où  $|T|$  est l'aire de  $T$ ,  $d = 2$  est le nombre de dimensions spatiales et  $\mathbf{n}_i^T$  est le vecteur entrant à  $T$ , dans le côté opposé à  $i$  et de norme égale à la longueur de ce côté [3].
- La fonction  $\omega$  sera considérée constante dans chaque élément  $T$  et égale à la moyenne  $\omega^T$  de sa valeur sur les noeuds de  $T$ .
- Comme l'intégrale est calculée dans le domaine de référence, on utilisera toujours les vecteurs normaux et les aires du maillage initial. La fonction  $\omega$ , en revanche, sera actualisée pour exprimer l'évolution du maillage, et son calcul sera faite en utilisant la solution interpolée à la fin de chaque itération.

On a, ainsi :

$$\begin{aligned} k_{ij} &= \int_{\Omega_\xi^h} \omega \nabla_{\xi\eta} \varphi_i \cdot \nabla_{\xi\eta} \varphi_j d\xi = \sum_{T \ni i} \int_T \omega \nabla_{\xi\eta} \varphi_i \cdot \nabla_{\xi\eta} \varphi_j d\xi = \\ &= \sum_{T \ni i} |T| \omega^T \frac{\mathbf{n}_i^T \cdot \mathbf{n}_j^T}{4|T|^2} = \sum_{T \ni i} \omega^T \frac{\mathbf{n}_i^T \cdot \mathbf{n}_j^T}{4|T|} \end{aligned} \quad (12)$$

**Remarque 1** : comme montre le développement du modèle fait ci-dessus, le calcul de l'intégrale qui donne les éléments de la matrice  $K$  est faite **toujours sur le maillage de référence**, ce qui implique que, dans (12), on

utilisera toujours les mêmes vecteurs normaux et les mêmes aires pour le calcul discret. Néanmoins, **la fonction  $\omega$  est actualisée à chaque itération dans le maillage physique**, afin de bien indiquer que, dans ses nouvelles positions, les noeuds sont dans des régions de raffinement ou pas.

**Remarque 2 :** Pour le calcul des éléments de  $K$ , on a profité du fait que, dans la méthode d'éléments finis P1, on a  $k_{ij} = 0$  si les noeuds  $i$  et  $j$  n'appartient pas au même élément. Ainsi, la matrice est creuse, ce que nous a motivé à le stocker avec une structure adaptée (au lieu d'allouer une matrice de taille  $N \times N$ ), afin de réduire la consommation d'espace mémoire, comme décrit ci-dessous :

- On identifie, d'abord, le nombre maximal de voisins d'un noeud du maillage,  $N_{neig}$ , sous la convention qu'un noeud est toujours voisin de lui-même ;
- On alloue deux vecteurs :
  - Un vecteur d'entiers,  $A$ , de taille  $(N_{neig} + 1).(N + 1)$  ;
  - Un vecteur de doubles,  $\tilde{K}$ , de taille  $N_{neig}.(N + 1)$  ;
- Le vecteur  $A$  contient les index des voisins de chaque noeud. Par convention,
  - $A_{(N_{neig}+1).i} = N_{neig}^i$  contient le nombre de voisins de  $i$  ;
  - $A_{(N_{neig}+1).i+1}$  jusqu'à  $A_{(N_{neig}+1).i+N_{neig}^i}$  contiennent les index des voisins de  $i$  (pour commodité, on garde toujours  $A_{(N_{neig}+1).i+N_{neig}^i} = i$ , mais cela n'est pas nécessaire)
- Le vecteur  $\tilde{K}$  contient les éléments de la matrice :
  - Si  $A_{(N_{neig}+1).i+z} = j$ , alors  $\tilde{K}_{N_{neig}.i+z-1} = K_{ij}$

Le stockage ici proposé n'est pas optimal, car, dans des maillages non structurés, les noeuds n'ont pas tous le même nombre de voisins. Par ailleurs, les  $N_{neig} + 1$  premier éléments de  $A$  et les  $N_{neig}$  premiers éléments de  $\tilde{K}$  ne sont pas utilisés, pour être en concordance avec les conventions d'index utilisés dans la bibliothèque MMG. Néanmoins, par rapport au stockage de la matrice complète, on a vérifié des très grandes avantages :

- Par exemple, dans un test avec environ 28000 noeuds et 165 mille éléments non nuls dans la matrice  $K$ , on est passé d'un stockage de 783 million de doubles (soit 0.02% d'utilisation) à un stockage de 308 mille doubles (soit 54% d'utilisation) et 336 mille entiers, avec un considérable gain en temps d'exécution.
- On a observé aussi des gains en temps d'exécution du programme. Dans la formulation originale, sans garder les index des voisins de chaque noeud, il fallait appeler une fonction qui retourne la liste de voisins. Cet appel était fait une fois par itération pour chaque noeud,

lors de la résolution du système linéaire.

## 2.4 Résolution du système linéaire

Le système linéaire (10) est résolu de façon itérative, avec la méthode de Jacobi. Dans les tests, on utilise en général un nombre petit d'itérations (dix ou vingt), ce qui donne des bons résultats pour un temps de calcul raisonnable. La solution est calculée en terme des déplacements  $\boldsymbol{\delta} = \mathbf{x} - \boldsymbol{\xi}$ . Ainsi, on réécrit le système linéaire (10) sous la forme

$$Kx = K(x - \delta + \delta) = 0 \longrightarrow K\delta = -K\xi$$

où  $\boldsymbol{\delta} = \mathbf{x} - \boldsymbol{\xi}$  est le déplacement des points. Ainsi,  $\forall i \in \{1, \dots, N\}$  :

$$k_{ii}\delta_i^{[n+1]} = - \sum_{j=1, j \neq i}^N k_{ij}\delta_j^{[n]} - \sum_{j=1}^N k_{ij}\xi_j = - \sum_{j=1}^N k_{ij}(\xi_j + \delta_j^{[n]}) + k_{ii}\delta_i^{[n]}$$

Donc, finalement,

$$\delta_i^{[n+1]} = \delta_i^{[n]} - \frac{1}{k_{ii}} \sum_{j=1}^N k_{ij}(x_j^{[n]}) \quad (13)$$

**Remarque** : avant d'actualiser la position des points, il faut vérifier si le déplacement calculé ne conduit pas à un croisement des noeuds. Pour faire cette vérification, on calcule les aires signées des éléments (i.e., en considérant l'ordre des noeuds). Si nécessaire, on relaxe le déplacement :

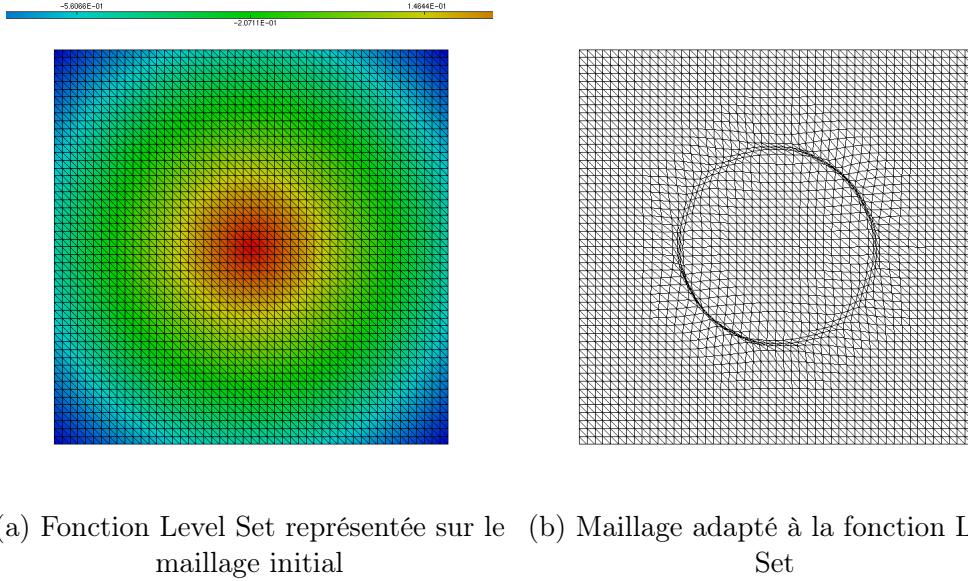
$$\mathbf{x}^{[n+1]} = \mathbf{x}^{[n]} + \theta (\boldsymbol{\xi} + \boldsymbol{\delta}^{[n+1]} - \mathbf{x}^{[n]}) \quad (14)$$

avec  $\theta \in [0, 1]$ . La relaxation est ainsi appliquée sur la différence entre deux positions successives, non pas sur le déplacement par rapport à la position initiale, car dans ce cas-ci on peut avoir de "retours en arrière" des points du maillage.

## 3 Application à la fonction Level Set

Le modèle décrit dans la section précédente sera utilisé pour adapter le maillage par rapport à une fonction Level Set (LS). Cette fonction définit une distance signée entre chaque point du maillage et la surface qu'on représente : si le point  $\mathbf{x}$  est à l'extérieur de la surface,  $LS(\mathbf{x}) > 0$ ; s'il est à l'intérieur,

$LS(\mathbf{x}) < 0$ ; et si il est sur l'interface,  $LS(\mathbf{x}) = 0$ . On cherche alors à bouger les points pour avoir une meilleure représentation de la ligne de niveau 0 de la fonction Level Set, i.e., de la surface de l'objet [5]. Pour illustrer notre objectif, on présente dans la figure 1 un exemple d'adaptation à une fonction Level Set :



(a) Fonction Level Set représentée sur le maillage initial      (b) Maillage adapté à la fonction Level Set

FIGURE 1 – Exemple d'adaptation à la fonction Level Set

### 3.1 Construction de la fonction à adapter (avec $\omega$ donné par l'expression (4))

Le calcul de  $\omega$  à partir de l'expression (4) est fait en utilisant le gradient et le hessien de la fonction à adapter. Pourtant, dans le cas de la fonction Level Set, l'adaptation ne sera pas faite en considérant  $u = LS$ , parce qu'on n'aurait pas de forts gradients, en tenant compte de la lisseté de cette fonction. Ainsi,  $\omega$  ne serait pas capable d'exprimer le raffinement désiré dans chaque partie du maillage. De cette façon, à partir de la fonction Level Set, on va construire une nouvelle fonction qui ait un très fort gradient sur les bords de l'objet.

Idéalement, on construirait une fonction avec un saut, valant 1 à l'intérieur et -1 à l'extérieur de l'objet, ce que fournit des très bons raffinements sur la ligne de niveau 0 de la fonction Level Set. Néanmoins, dans la résolution numérique de problèmes de la mécanique des fluides, il est intéressant

d'avoir aussi un bon raffinement du maillage sur une couche autour de la surface (la couche limite de l'écoulement), où l'interaction fluide-objet n'est pas négligeable. En effet, pour avoir un calcul précis, un maillage approprié sur la couche limite est requis par la plupart des schémas numériques et logiciels pour les problèmes de fluides [7]. Ainsi, on a cherché une fonction plus lisse.

Dans un premier moment, on a défini une fonction avec une variation linéaire entre les deux étages (figure 3). Néanmoins, les tests réalisées ont fourni des résultats de mauvaise qualité : étant le gradient constant à l'intérieur de la pente, les points dans cette région (y compris les points les proches du bord de l'objet) n'avaient pas la tendance de bouger, au contraire des points proches des bordes de la pente, dû à la discontinuité du gradient de  $u$ . Par conséquent, l'adaptation produisait une maillage avec deux petites couches raffinées, qui ne représentaient pas la surface de l'objet (figure 2).

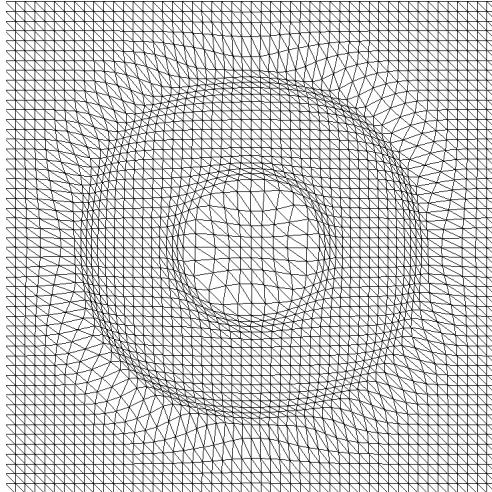


FIGURE 2 – Adaptation à une fonction Level Set modifié avec une variation linéaire

On a ainsi cherché une fonction dont le gradient est plus variable et continue afin de réduire cet effet. On a obtenu des bons résultats en utilisant

$$u(\mathbf{x}) = \frac{1}{1.1.107} \operatorname{atan} \left( \frac{2LS(\mathbf{x})}{\delta} \right) \quad (15)$$

étant  $2\delta$  la taille de la couche, choisie par l'utilisateur. Cette fonction est représentée dans la figure 3.

On peut jouer aussi avec le paramètre multiplicateur de  $LS(\mathbf{x})/\delta$  à l'intérieur de la fonction tangent. Avec 1, par exemple, on obtient un résultat plus proche de la pente linéaire ; avec 5 ou 10, la fonction devient plus discontinue

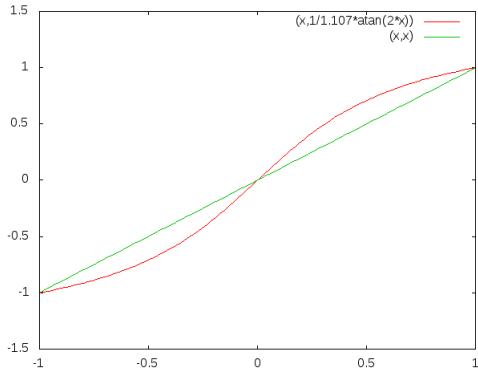


FIGURE 3 – Comparaison des options pour la lissage de la fonction à adapter

et la couche de raffinement est plus petite. On restera ainsi avec l’option donnée par (15).

### 3.1.1 Choix des paramètres

Afin de chercher les paramètres qui génèrent le meilleur maillage, plusieurs tests ont été réalisés :

- Cas test (objet cible) :
  - Cercle ;
  - Triangle ;
  - Profil d’une aile (NACA)
- Type de maillage
  - Structuré ;
  - Non structuré

Les meilleurs maillages obtenus ont été utilisés pour résoudre un problème d’écoulement autour de l’objet, afin de vérifier l’influence sur les résultats. Les conclusions des tests sont les suivants :

- Pour le calcul de  $\omega$ , il faut plutôt utiliser le gradient de  $u$ . Le hessien peut être aussi utilisé, mais il a la tendance de créer deux couches raffinées dans les bords de la région de variation de la fonction  $u$  (figure 4) ;
- Le raffinement d’une certaine région du maillage cause un étirement dans la direction normale à surface des éléments voisins à cette région. Ce résultat n’est pas désirable car on veut plutôt une anisotropie dans la direction de l’écoulement du fluide (figure 5) : l’erreur d’interpolation est plus petit si on a des triangles anisotropes dont le côté le plus long est dans la direction des petites dérivées de deuxième ordre de

la solution [9]. Ainsi, il est important de avoir une valeur de  $\delta$  assez grande.

- Néanmoins,  $\delta$  ne peut être trop grand, car la fonction arctangente s'approche de la pente linéaire, et on n'obtient pas un bon raffinement à l'intérieur de la couche.

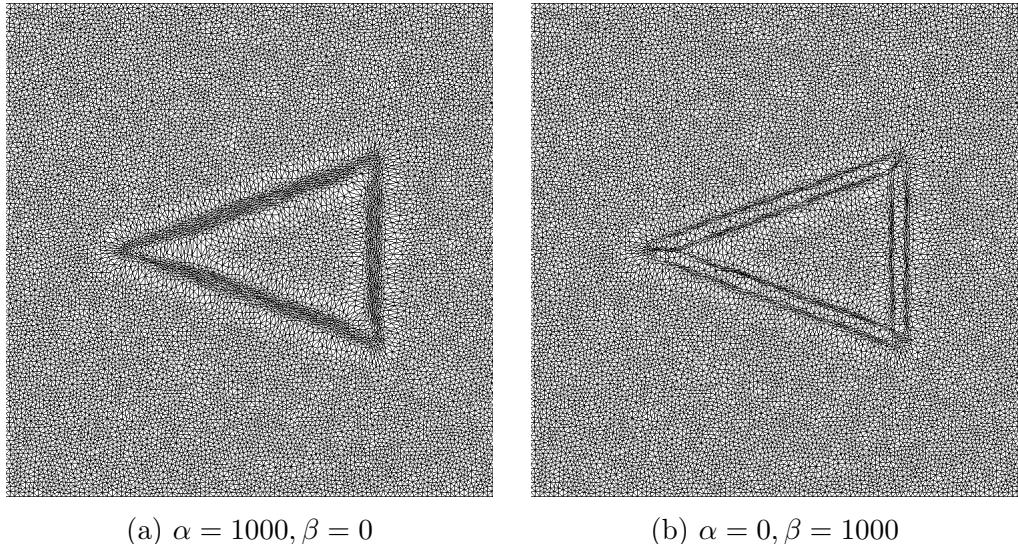


FIGURE 4 – Influence des paramètres  $\alpha$  et  $\beta$  sur l'adaptation du maillage (les deux cas ave  $\delta = 0.02$ )

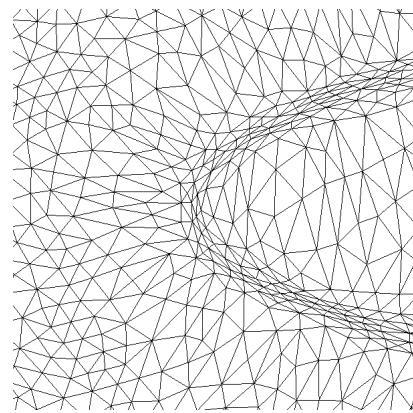


FIGURE 5 – Détail d'un maillage obtenu avec  $\delta = 0.005$

En tenant compte de ces conclusions, on recommande l'utilisation des paramètres suivants (on rappelle qu'on utilise toujours les normes du gradient et du hessien normalisés entre 0 et 1) :

$$\begin{aligned}\alpha &= 1000 \\ \beta &= 0 \\ \delta &= 0.01, 0.02\end{aligned}\tag{16}$$

On peut aussi faire des adaptations successives, en faisant varier les paramètres, afin d'obtenir d'autres résultats. Par exemple, on peut faire une première adaptation avec  $\delta = 0.02$  pour obtenir une bonne couche raffinée, et après une deuxième avec  $\delta = 0.01$  pour obtenir un raffinement plus précis autour de la surface.

La figure 6 présente quelques exemples d'adaptation à une fonction Level Set, avec les paramètres présentés ci-dessus, sur un maillage non structuré avec environ 28000 noeuds :

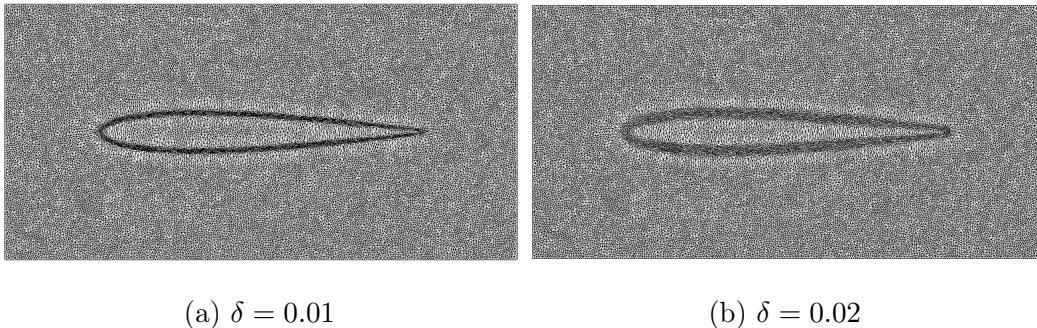


FIGURE 6 – Exemples d'adaptation à une fonction Level Set

### 3.2 Calcul des tailles désirées

Le calcul des tailles désirées est basée sur la définition d'une métrique pour chaque point du maillage. Il s'agit d'une matrice  $2 \times 2$  (dans le cas bidimensionnel) définie positive symétrique, construite de façon à contrôler l'erreur de la représentation de la solution sur le maillage modifiée [8].

La description détaillé de la définition, calcul et interprétation géométrique de la métrique seront faites dans la section 4, pour l'adaptation à une solution physique. Dans le cas de l'adaptation Level Set, on a implémenté un calcul simple des tailles désirées, en se basant sur la méthode utilisée par [5], qui permet de contrôler la distance de Hausdorff entre la surface de l'objet et sa représentation. En faisant attention toujours à l'importance, pour une bonne adaptation, de créer une fonction  $\omega$  qui soit au même temps assez lisse et qui présente des variations qui permettent le mouvement des noeuds, on définit, avec le paramètre  $\delta$ , des couches pour l'imposition des tailles :

$$h_{des}(\mathbf{x}_i) = \begin{cases} h_{min} & \text{si } |LS(\mathbf{x}_i)| \leq \frac{\delta}{2} \\ 2h_{min} & \text{si } \frac{\delta}{2} \leq |LS(\mathbf{x}_i)| \leq \delta \\ h_{max} & \text{sinon} \end{cases} \quad (17)$$

et on lisse les tailles sur le maillage avec le logiciel *MMG*, en imposant une gradation de 10%.

### 3.2.1 Résultats

La figure 7 illustre l'adaptation Level Set basée sur l'imposition de tailles désirées à chaque noeud.

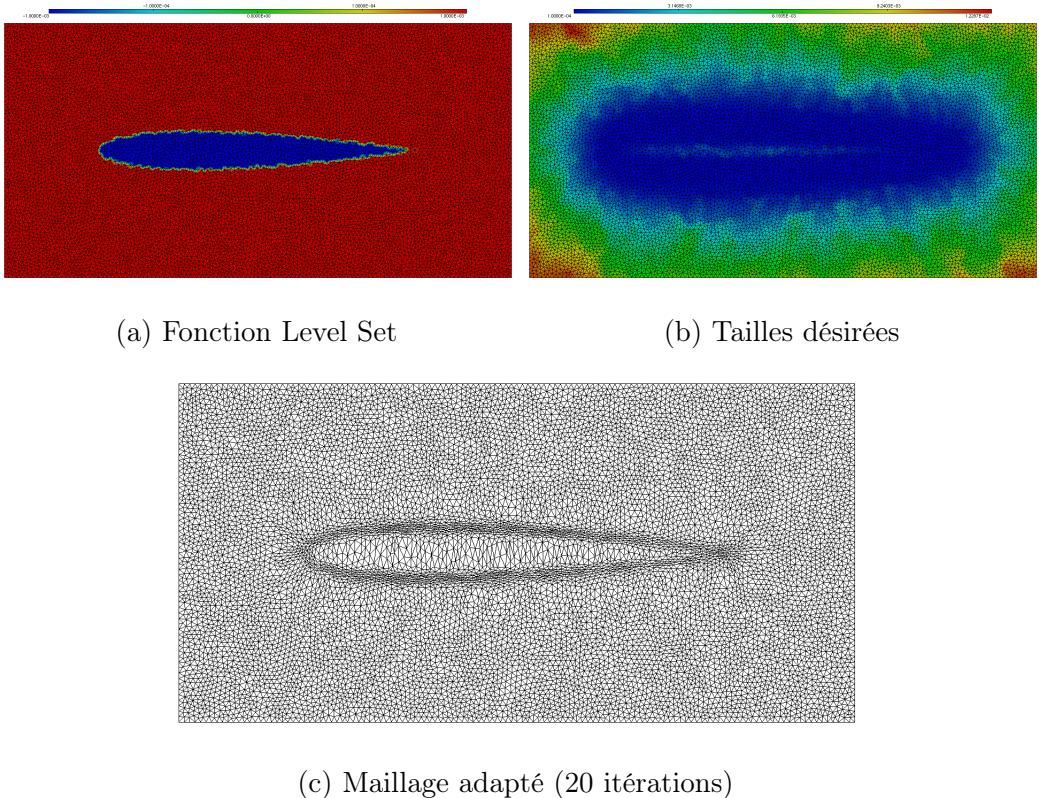


FIGURE 7 – Adaptation à une fonction Level Set à partir de l'imposition des tailles désirées

## 3.3 Relaxation et qualité du maillage

Pendant la résolution du système linéaire, on fait, si nécessaire, une relaxation des déplacements calculés afin d'éviter le croisement des noeuds et

la conséquente occurrence d'aires négatives. Néanmoins, en dépendant du maillage de départ (par exemple, un maillage déjà raffiné), la relaxation est trop contraignant et impose l'arrêt de l'algorithme lors des premières itérations. Parfois cela est provoqué par un petit nombre d'éléments.

On a ainsi implémenté, dans les premières itérations, un prétraitement d'optimisation, selon le schéma suivant :

- Si le numéro de l'itération actuelle n'est pas supérieur à  $iter_{optim}$  :
  1. Identifier tous les éléments qui demanderont de la relaxation ;
  2. Classifier ces éléments selon un critère de qualité (mesure de l'anisotropie) ;
  3. Optimiser les  $n_{optim}$  éléments les plus critiques de cette liste

L'optimisation réalisée consiste en, itérativement, approcher chaque noeud  $i$  de l'élément concerné du barycentre des noeuds voisins de  $i$ . L'optimisation est arrêtée dès qu'aucun des trois noeuds de l'élément bouge au dessus d'un seuil, ou qu'au moins un des noeuds n'arrive pas à réduire l'anisotropie au dessous d'une certaine pourcentage, par rapport au début de l'itération.

La figure 8 montre le résultatat de son application sur un élément :

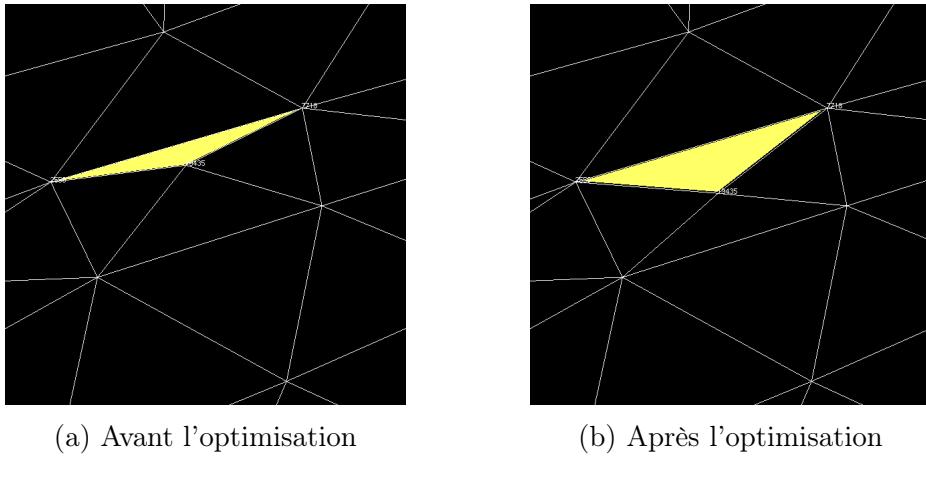


FIGURE 8 – Résultat de l'optimisation d'un élément

Pour que le prétraitement ne soit pas très coûteux, on a choisi  $n_{optim} = 10$  et  $iter_{optim} = 2$ . Les tests réalisés montrent des très bons résultats quand l'algorithme s'arrête au début de l'exécution, et avec l'optimisation on arrive à tourner tous les itérations. Néanmoins, si appliqué sur un maillage de bonne qualité et qui ne demande pas une forte relaxation, les gains sont en général petits (ou même on perd un peu de la qualité du résultat final).

### 3.4 Adaptation 3D

Après le développement et validation du modèle bidimensionnel décrit jusqu'ici, avec l'obtention de bons résultats d'adaptation des maillages, on l'a étendu au cas tridimensionnel. Du point de vue théorique, cette extension est assez simple, étant le modèle toujours décrit par le problème (5). Le développement de sa formulation faible reste aussi la même, et on se ramène ainsi à la résolution des systèmes linéaires

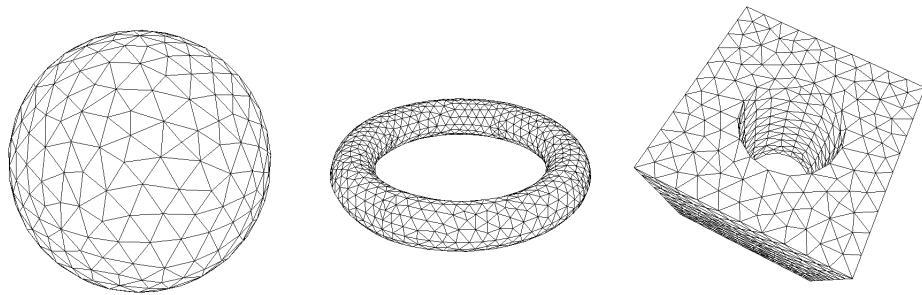
$$\begin{cases} Kx = 0 \\ Ky = 0 \\ Kz = 0 \end{cases} \quad (18)$$

Comme ces systèmes sont indépendants et ont la même forme, l'extension du cas 2D au 3D a été également simple du point de vue de l'implémentation. En effet, parmi les calculs décrits en détail dans les sous-sections 2.3 et 2.4, l'unique modification concerne le calcul des vecteurs normaux ( $(\nabla_{\xi\eta}\varphi_i)^T = \frac{\mathbf{n}_i^T}{d!|T|}$ , avec  $d = 3$ ). Ainsi, les éléments de la matrice  $K$  sont donnés par

$$\begin{aligned} k_{ij} &= \int_{\Omega_\xi^h} \omega \nabla_{\xi\eta}\varphi_i \cdot \nabla_{\xi\eta}\varphi_j d\xi = \sum_{T \ni i} \int_T \omega \nabla_{\xi\eta}\varphi_i \cdot \nabla_{\xi\eta}\varphi_j d\xi = \\ &= \sum_{T \ni i} |T| \omega^T \frac{\mathbf{n}_i^T \cdot \mathbf{n}_j^T}{(3!|T|)^2} = \sum_{T \ni i} \omega^T \frac{\mathbf{n}_i^T \cdot \mathbf{n}_j^T}{36|T|} \end{aligned} \quad (19)$$

On présente dans les figures 10 à 12 quelques résultats de l'adaptation Level Set tridimensionnelle, à partir de l'imposition des tailles désirées. Les fonctions Level Set utilisées sont celles de la figure 9. Malgré les résultats analogues au cas 2D et cohérents avec les objectifs ici proposés, l'adaptation 3D a des limitations plus fortes concernant le nombre de points du maillage, qui peut devenir très grande pour qu'on ait un maillage suffisamment fine pour la bonne représentation de la fonction Level Set, et la relaxation de l'adaptation, puisque le degré de liberté additionnel peut conduire plus rapidement au croisement des noeuds.

Les domaines de départ sont plus raffinées dans les régions où se trouve l'objet, afin de minimiser l'influence des bords sur l'adaptation. Le logiciel *MMG* a été utilisé pour les générer, en imposant une taille minimale  $hmin = 0.05$  et une gradation de  $hgrad = 1.1$  de la variation de la taille. Les adaptations ont été faites avec  $\delta$  entre 0.05 et 0.07.

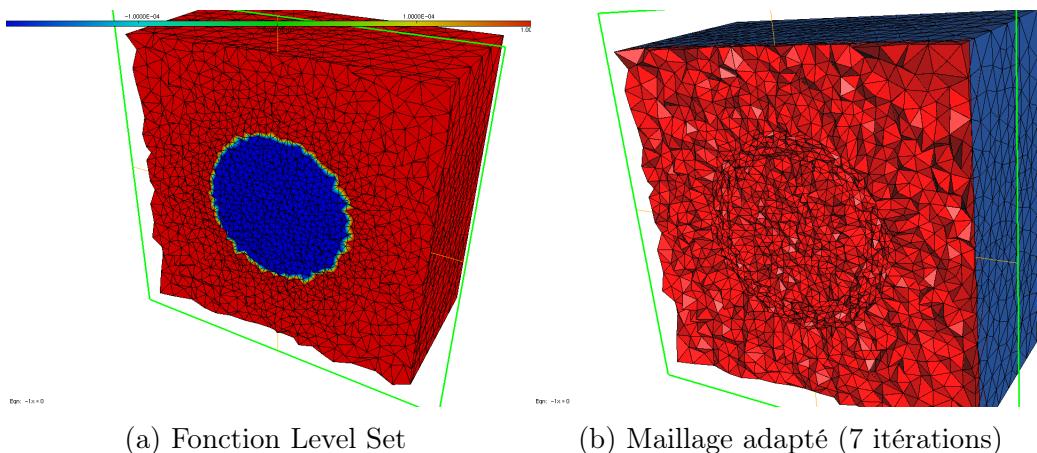


(a) Sphere

(b) Tore

(c) Boîte avec un trou

FIGURE 9 – Fonctions Level Set utilisées



(a) Fonction Level Set

(b) Maillage adapté (7 itérations)

FIGURE 10 – Adaptation du maillage à la sphère (environ 34000 noeuds)

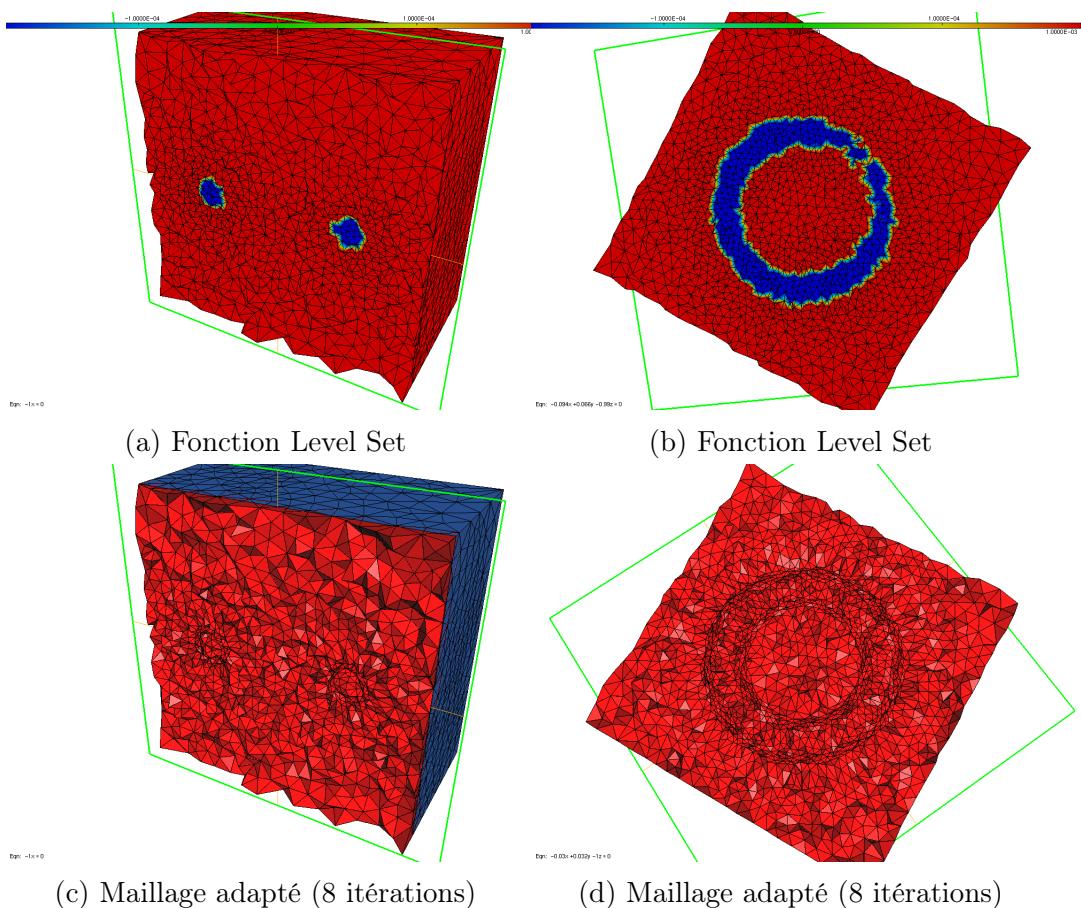


FIGURE 11 – Adaptation du maillage au tore (environ 25000 noeuds)

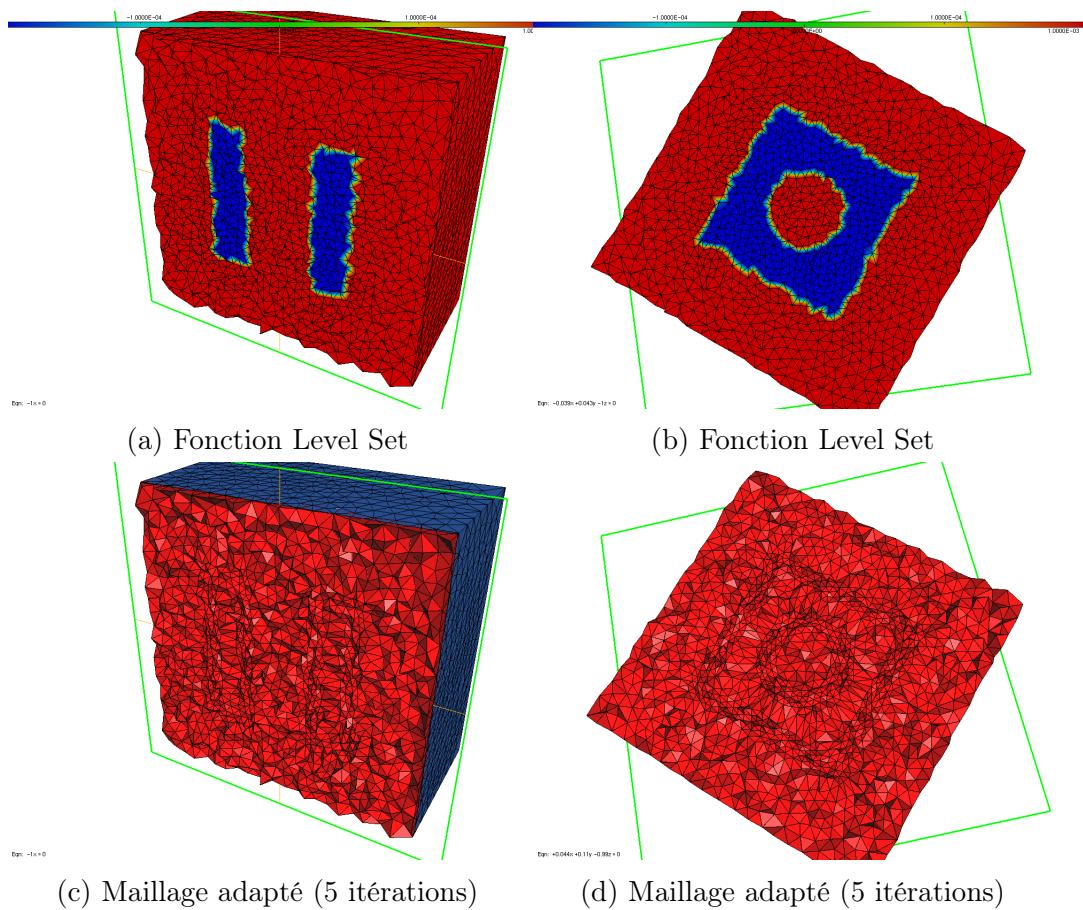


FIGURE 12 – Adaptation du maillage à une boîte avec un trou (environ 34000 noeuds)

## 4 Application à l'adaptation physique

De la même façon que l'adaptation Level Set, l'adaptation physique (i.e., à une variable physique calculée sur le maillage, comme le champs de vitesse de l'écoulement) peut être faite avec les deux formulations de  $\omega$ . Pour la première, basée sur la variation de la fonction, on n'a pas de remarques supplémentaires à faire : au contraire de la fonction Level Set, qui est toujours lisse, les variables physiques dans les problèmes de mécanique de fluides qui nous intéressent présentent en général des discontinuités, ayant ainsi des fortes gradients et hessiens qui permettent une bonne adaptation. Alors, il ne faut pas construire une autre fonction à adapter.

On va ainsi détailler seulement le deuxième calcul de  $\omega$  :

### 4.1 Calcul des tailles désirées

Le calcul des tailles présenté dans la suite s'inspire dans [4] et [6], et s'est basée sur la notion de métrique : pour chaque noeud  $i$  du maillage, on définit une matrice  $\mathcal{M}_i$   $2 \times 2$  (dans le cas bidimensionnel) qui minimise un estimateur de l'erreur d'interpolation de la solution sur le maillage.

Étant  $\Pi_h u$  l'interpolation de  $u$  sur le maillage, cet estimateur est donné, pour chaque élément  $K$  du maillage, par

$$\|u - \Pi_h u\|_{\infty, K} \leq c_d \max_{x \in K} \max_{\vec{e} \in E_K} \langle \vec{e}, |H_u(\mathbf{x})| \vec{e} \rangle \quad (20)$$

où  $c_d = \frac{2}{9}$  est obtenu avec un développement de Taylor de (20) dans son point de maximum. On veut imposer une limite  $\epsilon$  à cette erreur :

$$\|u - \Pi_h u\|_{\infty, K} = \epsilon \quad (21)$$

En définissant la métrique

$$\mathcal{M} = \frac{c_d}{\epsilon} H_u(\mathbf{x}) \quad (22)$$

la longueur des arêtes selon cette métrique est alors

$$l_{\mathcal{M}_i} = \langle \vec{e}, |\mathcal{M}_i| \vec{e} \rangle = 1 \quad (23)$$

c'est à dire, la métrique qui assure l'erreur d'interpolation désirée est celle telle que le maillage soit unitaire [4, 6].

Étant la métrique une matrice positive définie symétrique, elle est toujours diagonalisable es ses valeurs propres  $\tilde{\lambda}_i^j$ ,  $j = 1, 2$ , sont toujours réelles. En effet, la métrique peut être représentée par une ellipse (ou un ellipsoïde, en 3D), dont les vecteurs propres donnent la direction des axes et les valeurs

propres sa taille, selon la relation  $h_j = \frac{1}{\sqrt{\tilde{\lambda}_i^j}}$  [8]. Ainsi, les caractéristiques géométriques des éléments (taille, forme et orientation) est contenue dans la métrique, et comme on a plus d'un valeur propre, on en définit ainsi des éléments anisotropes.

Néanmoins, on va considérer un cas isotrope, en prenant la plus grande valeur propre de la métrique, ce qui donne la plus petite taille de chaque élément. Enfin, en établissant des seuils minimal et maximal pour la taille désirée ( $h_{min}$  et  $h_{max}$ ), et en tenant compte de l'équation (22), les tailles désirées sont calculées à partir de

$$h_i = \frac{1}{\sqrt{\min(\max(\frac{c_d}{\epsilon} \lambda_i, h_{max}^{-2}), h_{min}^{-2})}} \quad (24)$$

où  $\lambda_i = \max_{j=1,2} |\lambda_i^j|$  est la plus grande valeur propre du hessien  $H_u(\mathbf{x}_i)$ , en valeur absolue.

Également au cas de l'adaptation Level Set, les tailles définies sur le maillage sont lissées avec une gradation de 10%.

La figure 13 présente un exemple de métrique calculée à partir d'une variable physique .

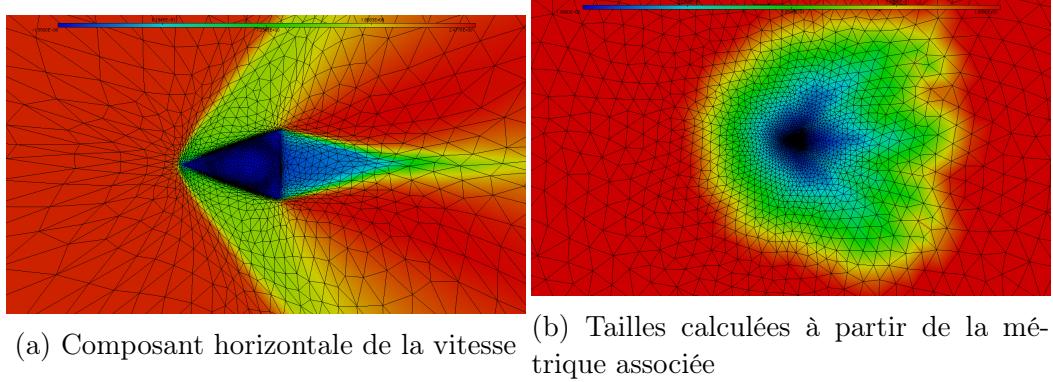


FIGURE 13 – Écoulement autour d'un triangle - calcul des tailles désirées à partir d'une variable physique

## 4.2 Couplage entre adaptation Level Set et adaptation physique

L'objectif principal de l'ensemble du travail réalisée dans ce projet est l'application du modèle et de la bibliothèque développés à la résolution de

problèmes de la mécanique de fluides, afin d'améliorer la précision des résultats. Ainsi, un bon maillage serait adapté au même temps à la surface de l'objet dans le domaine et au profil de la variable physique calculée (la vitesse, par exemple), et par conséquent les deux types de adaptation présentées ci-dessus doivent être considérées.

Pour coupler les deux adaptations, on va intersecter ses respectives métriques : en notant  $h_i^{LS}$  et  $h_i^{phys}$  la taille calculée pour le noeud  $i$  à partir des métriques, respectivement, de la fonction Level Set et de la variable physique (avant le lissage), on choisit le taille

$$h_i = \min(h_i^{LS}, h_i^{phys}) \quad (25)$$

Ensuite, on applique le même lissage avec une gradation de 10%.

Pour le couplage des deux adaptation, on a proposé la procédure itérative suivante :

1. Calcul de la métrique associée à la fonction Level Set ;
2. Adaptation du maillage à cette métrique ;
3. Résolution du problème physique sur le maillage adapté ;
4. Calcul de la métrique associée à la variable physique et intersection avec la métrique de la fonction Level Set
5. Adaptation du maillage à cette métrique, en partant du dernier maillage obtenu, mais toujours avec le même maillage de référence (le maillage de départ, non adapté) ;
6. Retour au pas 3

### 4.3 Résultats

La procédure décrite ci-dessus a été utilisé dans la résolution d'un flot supersonique 2D autour d'un triangle, avec une méthode de distribution de résidus appliquée aux équations de Navier Stokes pénalisées. Ce cas test est présenté et résolu dans [8], dont les paramètres de l'écoulement sont les mêmes utilisés ici.

Étant le modèle pénalisé, le triangle n'est pas discrétisée dans le domaine, ce qui signifie qu'il y a des mailles à son intérieur. En effet, le terme de pénalisation est appliqué exactement aux noeuds à l'intérieur de l'objet, identifiés par la signe de la fonction Level Set. Ainsi, la première adaptation de notre procédure (adaptation à la ligne de niveau 0 de la fonction Level Set) est de grande importance pour la représentation de la surface et la résolution précise des équations dans ses voisnages.

La solution physique utilisée pour les adaptations suivantes est la composante horizontale de la vitesse.

Les tests ont été réalisées avec l'objectif de vérifier l'évolution de la qualité du résultat dans chaque itération de la procédure. Cette vérification sera faite de façon qualitative et quantitative. Pour la première, on regardera par exemple la régularité du profil de la vitesse et de ses lignes de niveau et la résolution autour des coins du triangle. Pour la deuxième, on adoptera le même critère utilisé par [8], en calculant l'angle entre le choc et l'axe  $y = 0$ , en utilisant un point du choc proche à cet axe, et identifiée par la proximité des lignes de niveau de la vitesse. La résolution analytique du problème fournit un angle  $\beta = 53.33^\circ$ , qu'on utilisera pour la comparaison.

Le résultat présenté dans les figures 14 à 17 a été fait sur un domaine circulaire, de rayon vingt fois plus grand que la taille du triangle, afin de représenter un domaine infini, qui minimise l'influence des bords. Afin d'éviter un nombre trop grand de points, mais encore avec un bon raffinement dans les régions les plus proches des bords pour bien capturer le choc, les tailles des éléments ont été définies en fonction de la distance au centre du domaine, augmentant linéairement entre 0.002 et 1. Le résultat est un maillage avec 12664 noeuds et 25200 éléments.

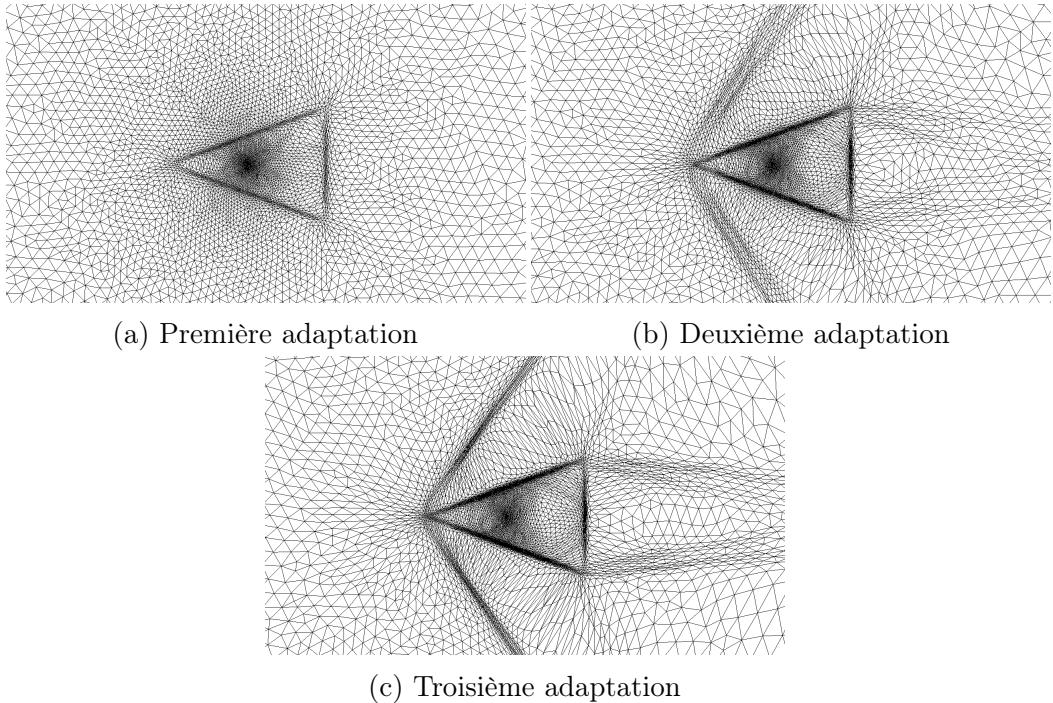


FIGURE 14 – Séquence de maillages adaptés obtenus

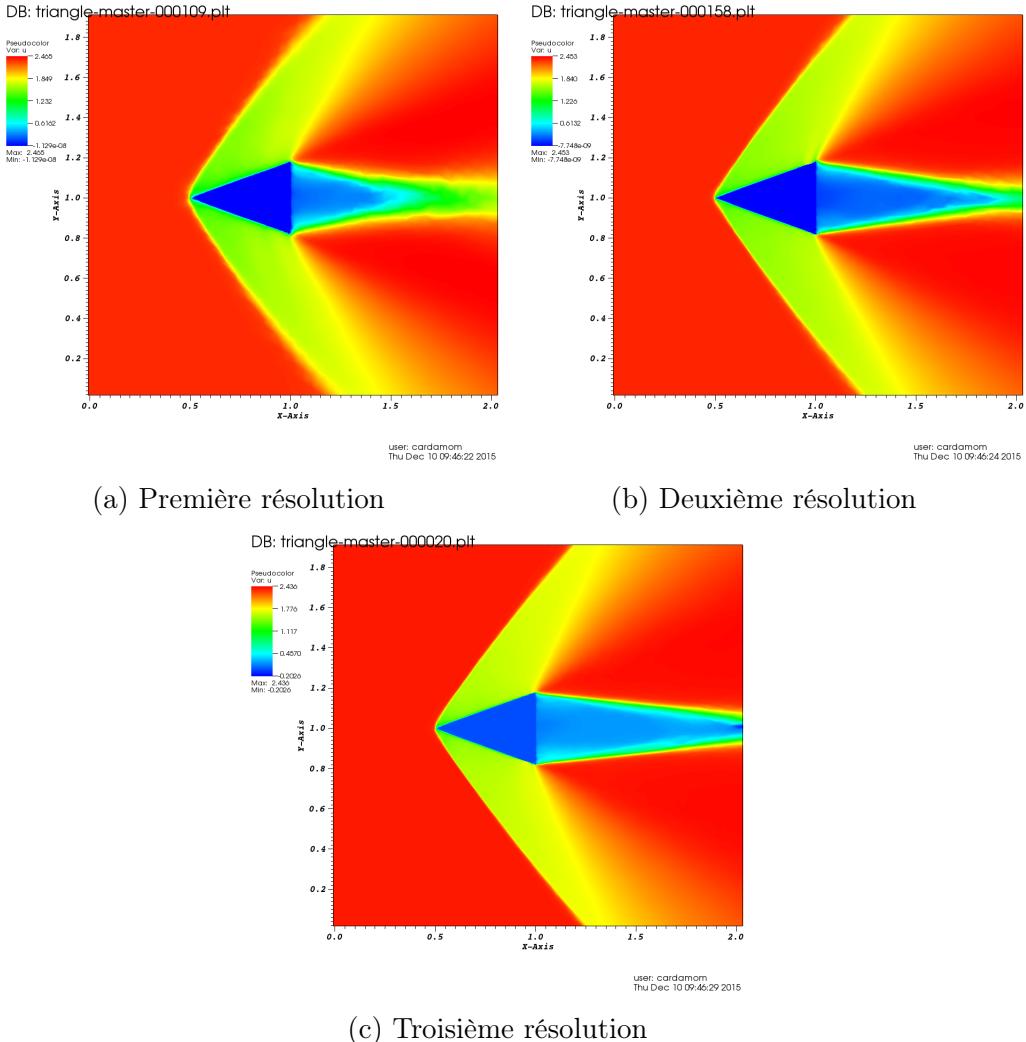


FIGURE 15 – Composante horizontale de la vitesse

L’angle  $\beta$  a évolué selon les valeurs présentées dans le tableau 1

Résolution	$\beta$
1	$55.68^\circ$
2	$53.92^\circ$
3	$53.01^\circ$

TABLE 1 – Angle entre le choc et l’axe  $y = 0$

Ces résultats montrent que la procédure proposée permet en effet d’améliorer la résolution du problème physique : à chaque itération, le champ de

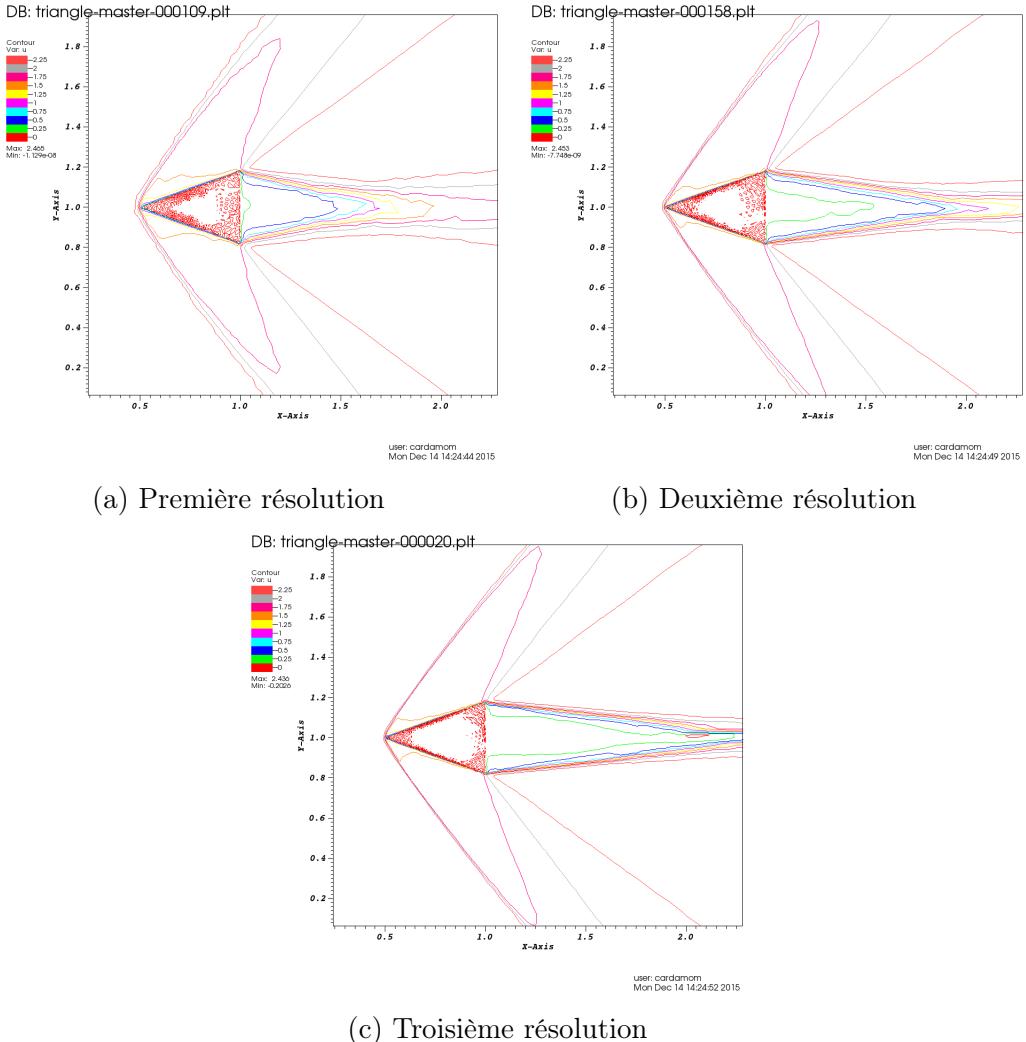


FIGURE 16 – Lignes de niveau de la composante horizontale de la vitesse

vitesse calculée devient mieux définie, avec des lignes de niveau plus régulières, et l’angle entre le choc et l’axe  $y = 0$  s’approche de la valeur analytique. Néanmoins, quelques difficultés sont encore présentes dans les tests réalisées, notamment la génération de maillages permettant un calcul stable (considérant les fortes discontinuités qui caractérisent ce cas test) et le fait que, comme montre la figure 17, le choc n’est pas attaché au triangle, au contraire du résultat théorique, ce qui s’explique par les limitations du raffinement du maillage.

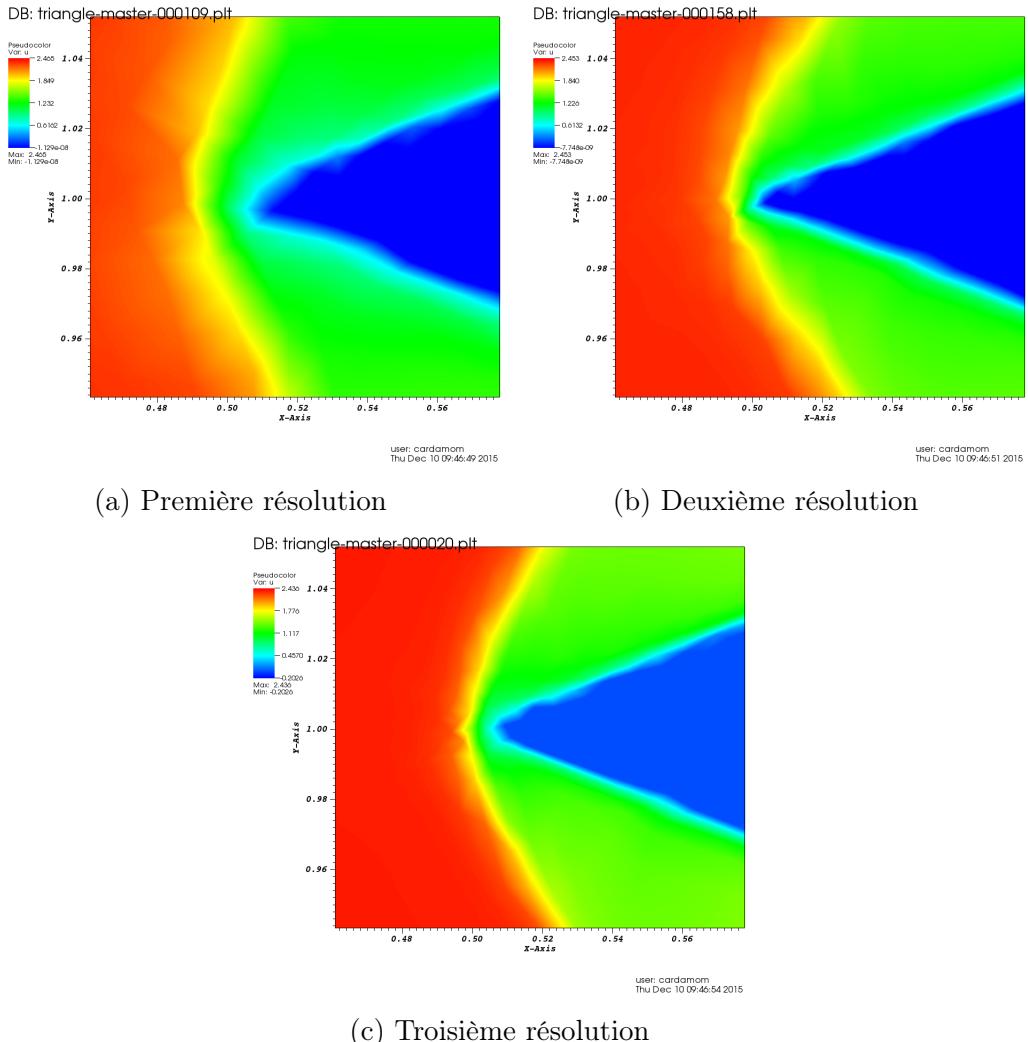


FIGURE 17 – Détail - Composante horizontale de la vitesse

## 5 Adaptation à un cas non stationnaire

Les résultats présentés dans les sections précédentes se réfèrent à des cas stationnaires, avec les surfaces fixes, mais une situation qu'on veut considérer aussi est celle où les objets bougent. Dans ce cas, il faut à chaque pas de temps actualiser la fonction Level Set et adapter le maillage autour de la nouvelle position de la surface. La procédure adoptée est la suivante :

- Pour tout noeud du maillage, calculer la fonction Level Set par rapport à la position actuelle de l'objet, et la fonction  $u$  à adapter (equation (15));

2. Adapter le maillage à la fonction  $u$  ;
3. Faire bouger la surface de l'objet (indépendamment de la position des noeuds)

L'objectif de cette procédure est de faire chaque adaptation à partir du dernier maillage adapté, au lieu de partir du maillage non adapté à chaque pas du mouvement. Ainsi, on peut éviter de garder en mémoire toute la structure du maillage de référence et du maillage adapté. Il faut pourtant garder au moins les aires et les vecteurs normaux du maillage, parce que, en raison du modèle adopté, toutes les adaptations doivent utiliser le même maillage de référence.

Les figures 18 et 19 présentent deux exemples de cette procédure, le premier pour l'advection d'un cercle et le deuxième pour l'oscillation de l'aile Naca. Les mouvements ont été discrétisées en 20 pas, et chaque itération a été réalisée avec 20 ou 30 itérations, respectivement.

On peut observer qu'on obtient à chaque pas une bonne adaptation à la position actuelle de la fonction Level Set, mais que les positions précédentes ne sont pas complètement "désadaptées", même que son trace soit faible. En effet, les tests montrent que le passage d'une adaptation à la prochaine se fait de deux manières, soit par le mouvement des points d'une région adaptée à l'autre, soit par le "dérafinement" d'une région. On a observé que le deuxième est plus lente que le premier, et ainsi on aurait besoin d'un plus grand nombre d'itérations. Néanmoins, pour justifier l'application de cette procédure, il faut trouver un équilibre entre le temps d'exécution et la qualité des résultats.

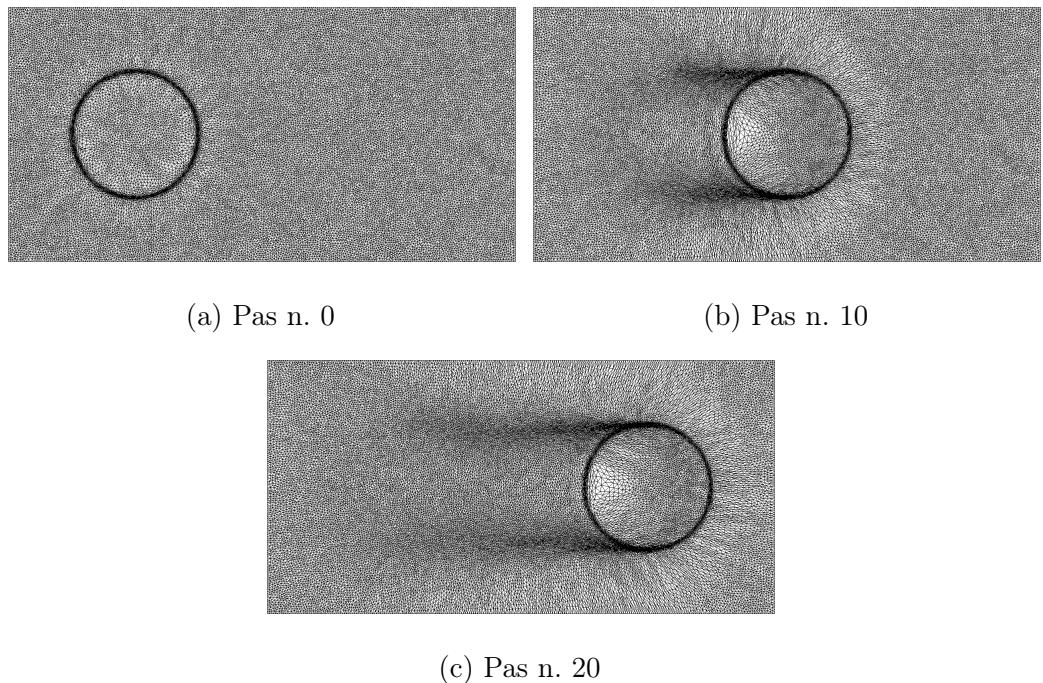


FIGURE 18 – Advection d'un cercle

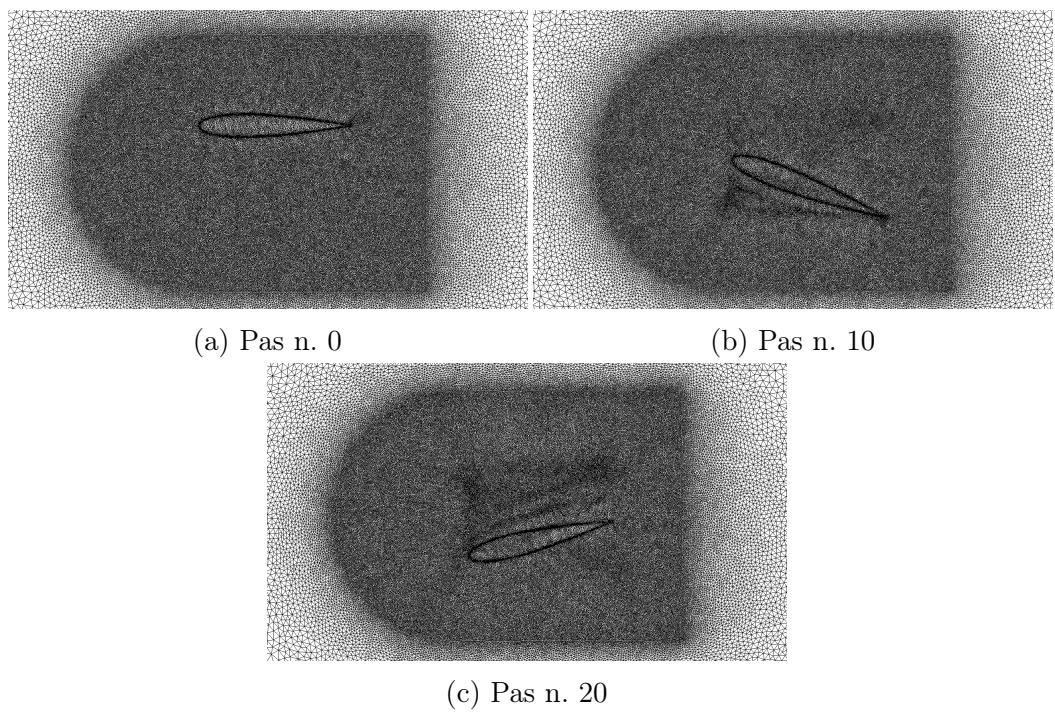


FIGURE 19 – Naca oscillant

## 6 Autres rapports

D'autres tests et résultats obtenus au long du développement du modèle et de la bibliothèque FMG peuvent être trouvés dans les rapports suivants, disposés en ordre chronologique. On peut notamment observer le moment où le modèle pour le calcul de  $\omega$  a été modifié :

- $\omega$  calculé à partir du gradient
  - 1. Adaptation à des fonctions analytiques : rapport\_adapAnalytique.pdf, rapport\_adapCircle.pdf, rapport\_adapCircleNS.pdf
  - 2. Adaptation à des fonctions LevelSet : rapport\_LevelSet.pdf
  - 3. Calcul d'un problème physique sur des maillages adaptés : rapport\_pbFluide.pdf
  - 4. Adaptation physique / Level Set : rapport\_physAdapt.pdf
  - 5. Adaptation à une fonction Level Set advectée : rapport\_circleAdv.pdf
  - 6. Récapitulatif de l'adaptation Level Set : recapitulatif\_adaptLS.pdf
- $\omega$  calculé à partir de la métrique
  - 1. Premières approches au nouveau modèle : rapport\_métrique.pdf et rapport\_taille\_maillage.pdf
  - 2. Adaptation 3D : rapport\_adaptation3D.pdf
  - 3. Adaptation physique en utilisant la solution du problème physique calcule sur des maillages adaptés : rapport\_physAdapt.pdf
  - 4. Adaptation à une fonction Level Set advectée : rapport\_LSAAdv.pdf
  - 5. Adaptation physique / Level Set : rapport\_physmetrique.pdf (texte) et GT\_en.pdf (présentation).

## Références

- [1] L. ARPAIA et M. RICCHIUTO : Mesh adaptation by continuous deformation. Basics : accuracy, efficiency, well balancedness. Rap. tech., INRIA Bordeaux Sud-Ouest, 2015. RR-8666, <hal-01102124>.
- [2] H. ASKES, L. J. SLUYS et B. C. C. JONG : Remeshing techniques for R-adaptives and combined H/R-adaptive analysis with application to 2D/3D crack propagation. *European Congress on Computational Methods in Applied Sciences and Engineering*, 2000.
- [3] L. CHEN : Programmim of finite elements in matlab. Notes du cours Math 226 : Computational PDEs (Univeristy of California Irvine). <http://www.math.uci.edu/~chenlong/226/Ch3FEMCode.pdf>.

- [4] C. DOBRZYNSKI : *Adaptation de maillage anisotrope 3D et application à l'aéro-thermique des bâtiments.* Thèse de doctorat, Université Pierre et Marie Curie - Paris VI, 2005.
- [5] V. DUCROT et P. FREY : Anisotropic level set adaptation for accurate interface capturing. In R. GARIMELLA, éd. : *Proceedings of the 17th International Meshing Roundtable*, p. 159–176. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-87920-6.
- [6] P. J. FREY et F. ALAUZET : Anisotropic metrics for mesh adaptation. *Computational Fluid and Solid Mechanics*, 2003.
- [7] A. LOSEILLE et R. LÖHNER : Robust boundary layer mesh generation. In X. JIAO et J.-C. WEILL, éds : *Proceedings of the 21st International Meshing Roundtable*, p. 493–511. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-33572-3.
- [8] L. NOUVEAU, H. BEAUGENDRE, C. DOBRZYNSKI, R. ABGRALL et M. RICCHIUTO : An explicit residual distribution scheme combined to mesh adaptation for solving the unsteady penalized Navier Stokes equations with a splitting approach. ???, Juin 2015. Preprint.
- [9] S. RIPPA : Long and thin triangles can be good for linear interpolation. *SIAM Journal on Numerical Analysis*, 29(1):pp. 257–270, 1992. ISSN 00361429. URL <http://www.jstor.org/stable/2158088>.