

Rapport de stage

**Fabien VIVODTZEV**

---

# **Simplification et visualisation de gros ensembles volumiques**

---

Responsable :  
Georges-Pierre BONNEAU

Laboratoire : GRAVIR  
Equipe : iMAGIS (CNRS, INRIA, INPG, UJF)



18 juin 2002



## *Remerciements*

Je tiens à remercier Georges-Pierre Bonneau pour son encadrement et son humour qui m'ont beaucoup apporté tout au long de ce projet de DEA.

Merci aussi à tous les membres du projet *iMAGIS* pour m'avoir agréablement intégré et aidé sur divers problèmes rencontrés durant ce stage.



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Positionnement du problème . . . . .	3
1.2	Les données . . . . .	4
1.3	L’approche . . . . .	5
1.4	Visualisation . . . . .	5
<b>2</b>	<b>L’état de l’art en simplification surfacique</b>	<b>6</b>
2.1	Différentes méthodes de simplification . . . . .	6
2.2	Critères de classification des méthodes de simplification . . . . .	11
2.3	Évaluation de la mesure d’erreur . . . . .	12
2.4	Mesures d’erreur spécifiques aux contractions d’arêtes . . . . .	13
<b>3</b>	<b>L’état de l’art en simplification volumique</b>	<b>15</b>
3.1	Les origines . . . . .	15
3.2	Extension des maillages triangulaires progressifs . . . . .	16
3.3	Détection efficace des intersections de bord . . . . .	19
3.4	Simplification à l’aide de fonctions splines . . . . .	20
3.5	Simplification avec respect de la topologie . . . . .	23
3.6	Computational Topology . . . . .	25
<b>4</b>	<b>Contributions</b>	<b>27</b>
4.1	Préservation de la topologie . . . . .	27
4.2	Technique de visualisation . . . . .	30
4.3	Occupation de l’espace mémoire . . . . .	32
4.4	Complexité . . . . .	34
<b>5</b>	<b>Résultats</b>	<b>35</b>
5.1	Présentation . . . . .	35
5.2	Interprétation . . . . .	35
<b>6</b>	<b>Perspectives</b>	<b>39</b>
6.1	Vers un nouveau calcul de l’erreur d’approximation . . . . .	39
6.2	Les besoins de la visualisation scientifique . . . . .	40
<b>7</b>	<b>Conclusion</b>	<b>41</b>
<b>A</b>	<b>Distance de Hausdorff</b>	<b>43</b>
<b>B</b>	<b>Génération d’une isosurface en utilisant TclTk et VTK</b>	<b>44</b>
	<b>Bibliographie</b>	<b>45</b>



# Chapitre 1

## Introduction

### 1.1 Positionnement du problème

La problématique à l'origine du présent sujet de DEA est celle du traitement de grandes masses de données. Les données peuvent provenir de codes de calculs (données simulées) ou de données observées. Ce projet est en étroite collaboration avec le CEA <sup>1</sup>/CESTA<sup>2</sup> qui est particulièrement intéressé par ces problèmes de traitement de gros résultats de calculs distants. Ce travail de DEA est intégré dans un projet du CEA qui se poursuivra en thèse sur le site DAM<sup>3</sup>-CESTA à Bordeaux, encadré par Paul Le Texier et Georges-Pierre Bonneau. Ce projet est financé par le CEA pour une durée de 35 mois, avec un contrat de formation par la recherche (bourse CFR).

Le traitement de gros ensembles de données volumiques est un problème nouveau qui s'est développé en parallèle avec l'amélioration des outils d'acquisition de données. Un scanner d'images de profondeur permet par exemple d'acquérir en quelques minutes plusieurs millions de points à la surface d'un objet. L'imagerie médicale, par scanner tomographique ou par scanner à résonance magnétique, est également source de grands ensembles de données. Dans ce domaine, il est possible de citer deux exemples d'ensemble de données, de taille particulièrement grande : il s'agit des "Visible Human & Female dataset" [oM86]. Leur taille respective est de 15 et 24 Go. Ces ensembles sont issus de la scanérisation de fines tranches découpées après congélation du corps de deux condamnés à mort. Aussi le projet "Terascale Visualization" [Lab98] centré sur la visualisation de très grosses données, effectué au laboratoire de Lawrence Livermore montre bien l'importance de ce domaine qui dispose de moyens matériels considérables.

Sans même parler de traitement ou de visualisation, le simple fait de stocker les données simulées devient un problème. Il n'est par exemple pas possible de stocker tous les pas de temps lors de simulations numériques dynamiques. Pour traiter de telles masses de données, des modélisations à différents niveaux de détails deviennent indispensables. Leur but commun est de résoudre la contradiction entre la masse des données et la nécessité de réduire le temps

---

<sup>1</sup> Commissariat d'Etude Atomique

<sup>2</sup> Centre d'Etudes Scientifiques et Techniques d'Aquitaine

<sup>3</sup> Direction des applications Militaires

de parcours de la boucle fondamentale de conception : simulation / traitement-visualisation / modification de paramètres (figure 1.1).

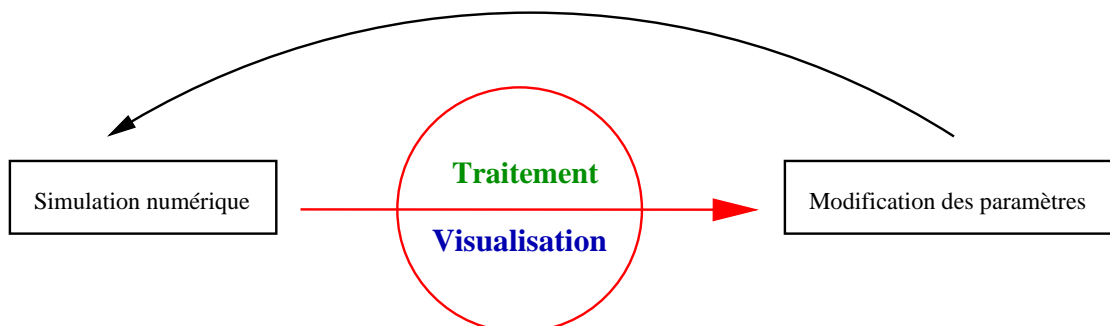


FIG. 1.1 – Boucle fondamentale de conception en simulation numérique

## 1.2 Les données

Les données qui m'intéressent plus précisément dans mon projet de DEA proviennent de simulations numériques. La finesse des grilles utilisées, le nombre de variables disponibles en chaque entité géométrique de cette grille (sommet, arête, face, cellule volumique...), et surtout la prise en compte de simulations dynamiques, font que la quantité de données générées par une simulation est énorme.

Ces données proviennent des centres de calculs du CEA, qui génèrent par simulations électromagnétiques de très grands ensembles de données. Les modèles étudiés de forme conique, mesurent environ un mètre de long et vingt centimètres de diamètre. Les équipes du CEA ont besoin de pouvoir visualiser les données rentrant dans le calcul de la SER (Surface Effective Radar) de ces objets de façon interactive et conviviale. Cette SER est un scalaire caractérisant la réponse des ondes envoyées depuis des radars vers ces modèles. Son calcul est très complexe et coûteux faisant intervenir un grand nombre de paramètres. Notamment son calcul nécessite la connaissance du champ magnétique et électrique en tout point de la surface du modèle.

C'est de cette application que naît mon projet de traitement de gros ensembles car la variation des données est locale et de très faible amplitude. Les échantillons nécessaires à son calcul doivent posséder des cellules ayant une taille de l'ordre de la longueur d'onde du champ incident. Autrement dit pour les dimensions qui nous intéressent, les maillages possèdent environ dix millions de cellules. Ces données très lourdes et surtout confidentielles ne sont pas disponibles en dehors du site du CEA, c'est pourquoi les travaux effectués durant mon stage de DEA portent sur d'autres ensembles volumiques.



## 1.3 L'approche

L'approche multirésolution est une notion très générale qui sous-entend une opération de simplification de données afin de pouvoir créer des niveaux de détails plus légers. Au sein de ce traitement de très nombreuses méthodes ont été développées afin de simplifier des maillages de types variés. Mon projet de DEA s'intéresse à une catégorie bien particulière d'entre elles qui est la simplification de maillages tétraédriques par contraction itérative d'arêtes. Le but final de cette simplification étant de pouvoir visualiser sur un poste de travail standard des ensembles de données de grosse taille (plusieurs millions de cellules).

Au niveau de la simplification, l'étude de la complexité des algorithmes utilisés est fondamentale. En effet pour le projet de DEA, les volumes de données sont environ dix fois plus petits que ceux utilisés aux CEA. C'est pourquoi il faut s'assurer que lors du passage à une plus grande échelle les algorithmes resteront utilisables. L'objectif est donc de conserver une complexité en  $n \log n$  tout au long des étapes de la simplification.

## 1.4 Visualisation

La visualisation volumique est devenue une orientation privilégiée dans le domaine de la visualisation scientifique. Cette discipline a pour but de traiter des ensembles donnés définis depuis des mesures discrètes dans un espace 3D. La visualisation permet d'abord d'en repérer intuitivement les caractéristiques importantes, les sous-ensembles remarquables ou encore les variations à échelle globale ou locale. Elle permet ensuite d'étudier de plus près ces caractéristiques, à l'aide de méthodes dédiées.

Les travaux de recherche dans le domaine de la visualisation se sont développés parallèlement à l'explosion des capacités des stations de travail, et plus récemment encore des ordinateurs personnels (PC). Un PC de 1000 Euros a des capacités graphiques au moins deux fois supérieures à celles d'une station de travail SGI O2 achetée 6000 Euros quatre ans auparavant. Néanmoins la taille des données à traiter, dans le domaine de la simulation numérique, a explosé ces quelques dernières années. A présent, il est nécessaire de disposer de moyens pour optimiser la visualisation de grands ensembles de données.

De nombreuses techniques de visualisation ont été développées pour exploiter ces données de façon efficace. Mais quelles que soient les données à traiter, la quantité d'information présente est toujours gigantesque posant de sérieux problèmes au niveau du temps de calcul et de l'espace nécessaire au stockage. C'est pourquoi, des méthodes ayant pour but de simplifier les données originales en préservant certains critères se sont révélées utiles pour une exploitation visuelle efficace.

# Chapitre 2

## L'état de l'art en simplification surfacique

Mon projet de DEA appartient au vaste domaine de la simplification de maillages. Dans ce dernier de très nombreuses méthodes ont déjà été étudiées, utilisant toutes sortes de mécanismes, surtout dans le domaine de la simplification de surfaces. L'extension aux maillages tétraédriques est plus récente et utilise beaucoup de notions de simplification surfacique. C'est pourquoi ce chapitre dresse premièrement, un rapide tableau de l'ensemble des méthodes de simplification surfacique. Ensuite, il introduit les notions et problèmes qui permettent d'organiser ce vaste ensemble de méthodes. Enfin, il termine par exposer les différentes méthodes qui permettent d'évaluer la mesure d'erreur commise au cours d'une simplification.

En plus de ce vaste panel d'articles, il existe plusieurs tours d'horizon écrits sur le sujet. De tels documents sont intéressants car ils regroupent l'ensemble des travaux du domaine. Cependant, ils ont tendance à énumérer et non classifier les différentes approches. Pour permettre d'avoir une vue d'ensemble et surtout de se positionner face au domaine, je ne détaille pas, dans ce chapitre, les choix utilisés par les auteurs. Au contraire, j'essaie de classifier selon des critères particuliers toutes ces techniques. Les études les plus complètes sur ce domaine ont été écrites par Michael Garland, Paul Heckbert [Gar99] [HG97], David Luebke [Lue01] ou encore Cignoni, Montani et Scopigno [CMS98].

### 2.1 Différentes méthodes de simplification

Pour chacune des techniques suivantes, je présente l'idée générale, quelques particularités et certains articles les mettant en oeuvre.

#### Techniques de décimation

Ces techniques suppriment un élément (sommet, arête, face, tétraèdre) du maillage basé sur un critère géométrique local et optimal pour cet élément. Après suppression, le trou résultant est comblé par triangulation (tétraédrisation en 3D), souvent coûteuse. Ces méthodes ne peuvent être utilisées que pour des surfaces manifold car la suppression d'un sommet nécessite la projection de la surface sur un plan. De par sa définition la décimation préserve la topologie.

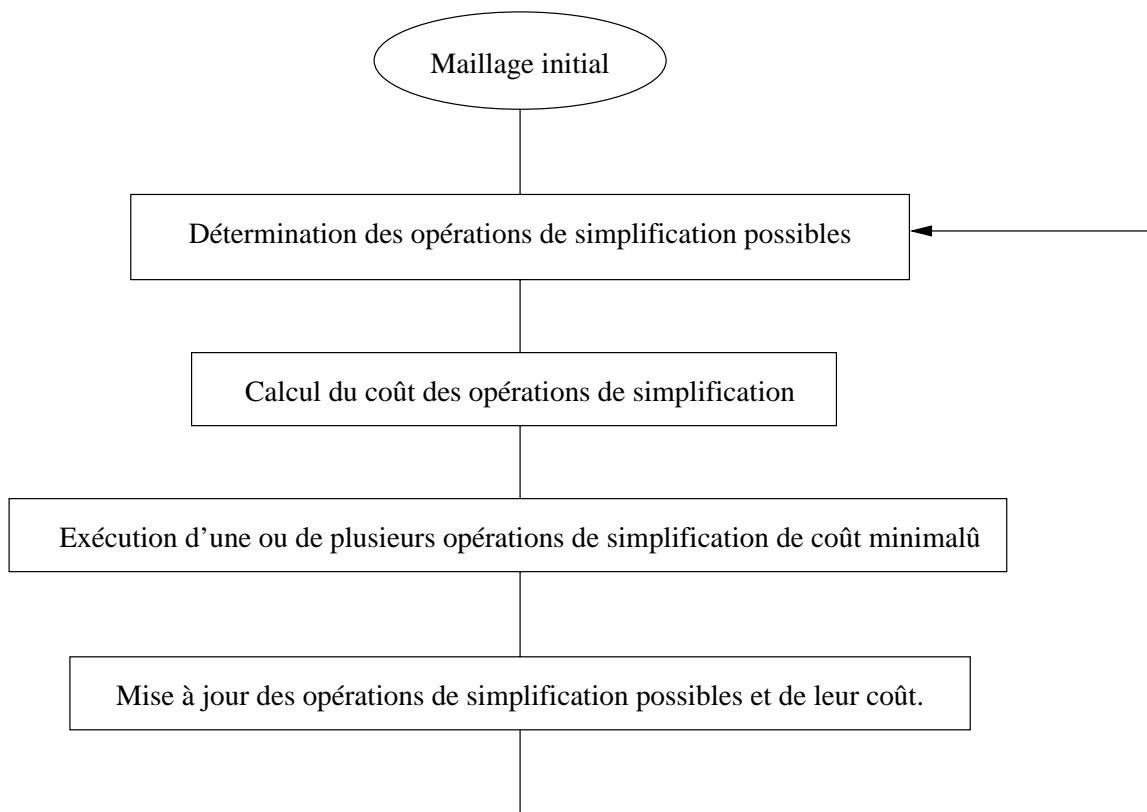


FIG. 2.1 – Schéma général de construction par décimation d’un modèle multirésolution

L’approche initiale [SZL92] effectue plusieurs boucles sur le maillage en supprimant les sommets qui satisfont un critère d’angle ou de distance (critères géométriques et topologiques locaux). Les candidats à supprimer sont choisis par un critère d’erreur local et un contrôle des modifications de la nouvelle surface triangulée a été proposée en regardant l’inversion d’arêtes [BS96]. Cependant, il y a beaucoup d’autres méthodes qui utilisent des heuristiques différentes pour le calcul de l’erreur globale [KLS96], [Gué95]. La décimation peut aussi porter sur des suppressions d’arêtes [RR96], de faces [Ham94], ou de tétraèdres.

La simplification d’enveloppes [CVM<sup>+</sup>96], illustrée figure 2.2, est une extension des techniques de décimation dans laquelle un erreur globale est utilisée pour le choix de la suppression. L’idée est de forcer la surface simplifiée à être entre deux surfaces seuils.

### Vertex Clustering

Basée sur la proximité géométrique, cette approche découpe le maillage en cluster et unifie tous les sommets d’un même cluster en un unique sommet représentatif [RB93].

La figure 2.3 montre les “clusters” en pointillé. Après fusion il ne reste bien qu’un seul sommet dans chaque cluster. Cette méthode est rapide et peut être appliquée à des surfaces non manifold. Cependant elle ne préserve pas la topologie et produit des simplifications très grossières pénalisant considérablement les petits détails. Des optimisations ont été apportées récemment afin d’améliorer la qualité visuelle des maillages simplifiés.

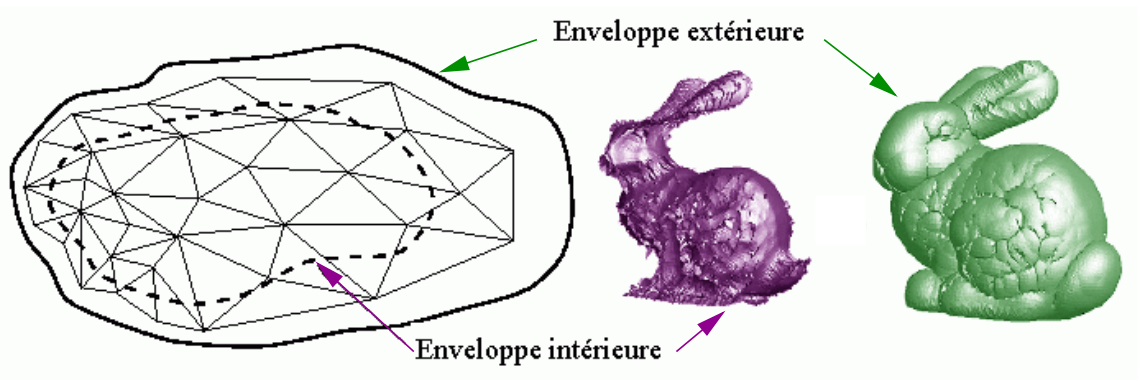


FIG. 2.2 – Simplification d'enveloppes

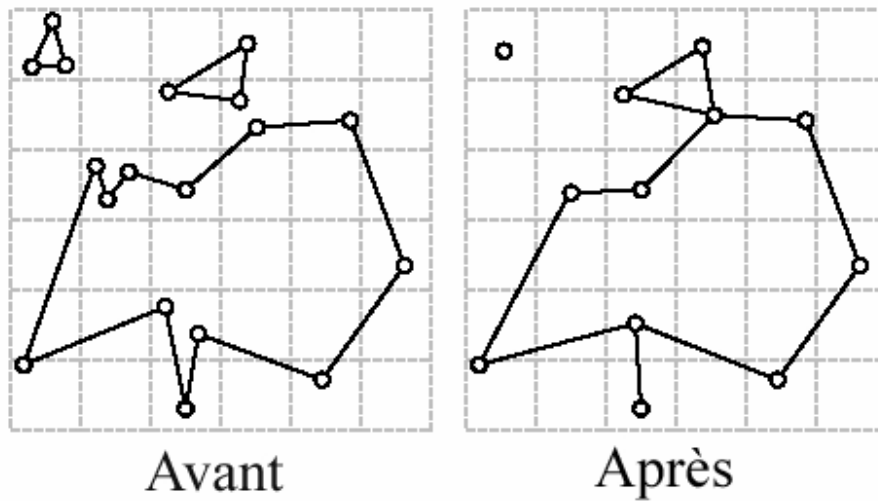


FIG. 2.3 – Simplification par Vertex Clustering

### Region Merging

Des faces coplanaires et quasi coplanaires sont cherchées au sein du maillage, liées en de plus gros polygones et re-triangulées avec un nombre de faces plus petit que le nombre initial. Cette liaison entre faces est effectuée par un test de coplanarité [HZ91]. Ces algorithmes sont en général restreints à des surfaces manifolds et ne modifient pas la topologie. La méthode des *surfaces* [KT96], étend cette approche en bornant les approximations et en optimisant la re-triangulation.

### Re-tiling

De nouveaux sommets sont insérés de façon aléatoire sur la surface originale du maillage et déplacés le long de la surface vers les régions à plus forte courbure. Les sommets initiaux sont alors supprimés itérativement et un maillage re-triangulé, construit avec ces nouveaux sommets, est généré. Le résultat de [Tur92] présenté figure 2.4 montre bien ce choix stratégique de point avant la retriangulation (illustration du milieu).

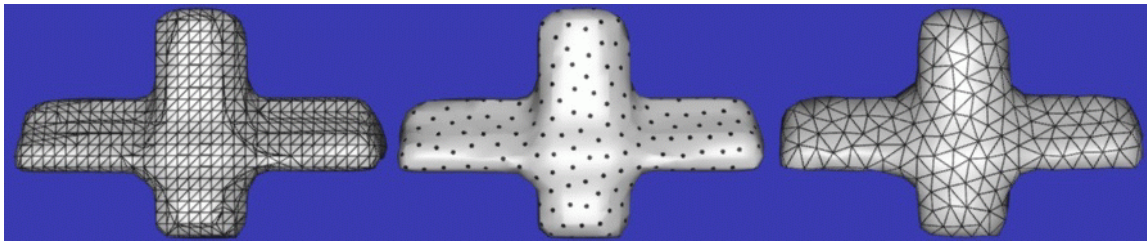


FIG. 2.4 – Simplification surfacique par re-tiling

### Raffinement

Le but est de trouver un maillage de base grossier qui peut être subdivisé pour retrouver le maillage original [Hop96].

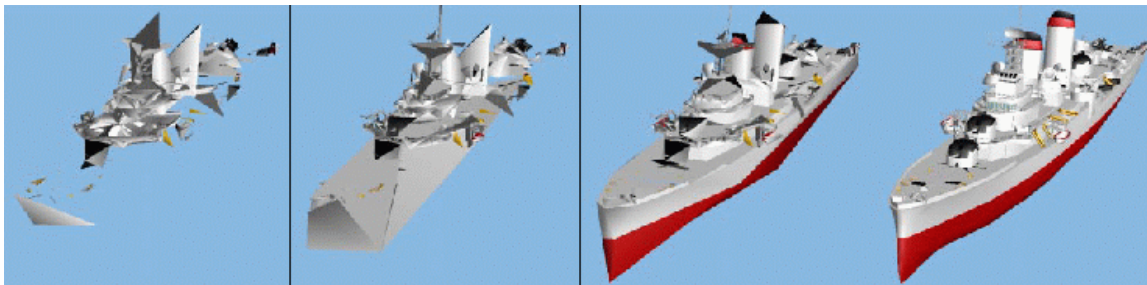


FIG. 2.5 – Raffinement d'un maillage grossier (à gauche) vers le maillage original (à droite)

Cette approche marche bien pour les terrains (l'approximation de base étant un rectangle) ou pour créer des modèles multirésolutions. Cependant peu d'algorithmes ont utilisé ces méthodes de raffinement à des maillages quelconques. En effet, la principale difficulté provient de la construction de l'approximation de base qui doit être de topologie équivalente et surtout qui est parfois difficile à construire.

### Décomposition en ondelettes

Cette approche qui semble intéressante pour la simplification de maillage en fournissant en plus la multirésolution, souffre de la nécessité d'une hiérarchie régulière pour supporter cette décomposition [GSG96].

Cette méthode a été conçue initialement pour des maillages réguliers mais elle a été étendue pour des maillages plus généraux. Elle préserve la topologie mais est altérée par les erreurs introduites lors de la construction du maillage de base qui se retrouvent lors de la reconstruction.

### Représentation hiérarchique intermédiaire

Une représentation en *octree* peut être adoptée pour produire un ensemble simplifié car l'arbre peut être simplifié à différents niveaux et converti en une simple représentation de boîte englobante.

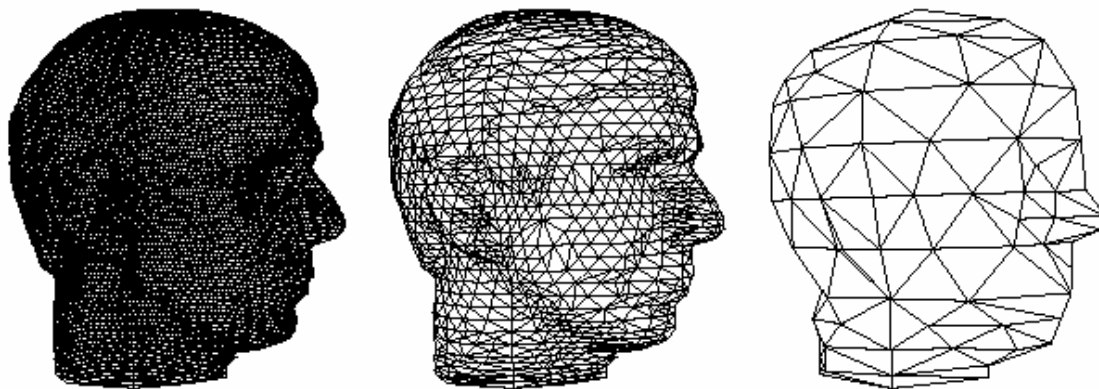


FIG. 2.6 – Décomposition en ondelette d’une surface

### Couplage géométrie-données numériques

La plupart des approches de simplifications se concentrent sur la géométrie de la surface, or en général il y a souvent comme en visualisation scientifique des données attachées à ce maillage. Ainsi il arrive que la simplification de la géométrie introduise d’importantes erreurs au sein des données numériques. Différentes approches ont été envisagées pour prendre en compte ces aspects, comme l’ajout de la couleur de la surface dans la décomposition en ondelettes ou la prise en compte dans la mesure d’erreur de ces données numériques. L’article [NFT00] utilise cette notion pour les couleurs d’un ensemble volumique.

### Contraction itérative

La plus grande classe d’algorithmes est basée sur la contraction itérative d’éléments (paire de sommets, arête, triangle, tétraèdre). La liste de toutes ces méthodes est longue mais le schéma général est sensiblement le même. A noter que la contraction de triangles peut être ramenée à la contraction successive de plusieurs arêtes. La contraction d’une paire de sommets  $(v_i, v_j) \rightarrow \bar{v}$  modifie le maillage en trois étapes :

- Déplacement du sommet  $v_i$  et  $v_j$  vers la position  $\bar{v}$
- Remplacement de toutes les occurrences de  $v_i$  par  $v_j$
- Suppression de  $v_i$  et de toutes les faces dégénérées, c’est-à-dire qui n’ont pas trois sommets distincts

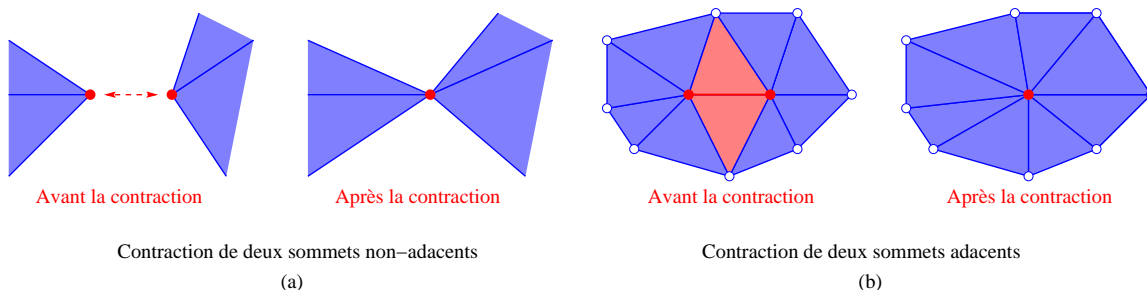


FIG. 2.7 – Contraction de sommets

La première étape modifie la géométrie de la surface et la seconde modifie la connectivité du maillage. A moins d'effectuer un traitement particulier, la topologie peut être modifiée par connexion d'éléments initialement distincts. L'opération fondamentale de contraction ne nécessite pas que le maillage soit manifold en contraste avec la méthode de décimation de sommets. La figure 6.2 (a) montre une contraction modifiant la topologie en contractant deux sommets non connectés dans le maillage initial. Cependant dans le cas de contraction de paire de sommets connectés, autrement dit d'arêtes (Figure 6.2 (b)), les contraintes sont plus fortes. La position finale du sommet  $\bar{v}$  peut être simplement la position de  $v_i$  ou  $v_j$  mais peut aussi être optimisée en minimisant une mesure d'erreur. La majorité de ces contractions itératives utilise une approche *gloutonne* pour sélectionner l'arête à supprimer. Un "coût" est attribué à chaque paire candidate et celle possédant le coût minimum est choisie, traitée, puis mise à l'écart. C'est le choix de ce coût qui différencie toutes les différentes approches en termes de qualités visuelles, temps de calcul, espace mémoire ou erreur introduite dans le maillage simplifié. Le paragraphe 2.3 détaille ces mesures qui sont fondamentales.

## 2.2 Critères de classification des méthodes de simplification

Bien que toutes ces méthodes possèdent un fonctionnement très différent, il est possible de les regrouper en catégories, classées sur différents critères. Ces critères proviennent directement de l'utilisation que l'on souhaite avoir des données à traiter. Par exemple, les besoins en visualisation scientifique (préservation de la topologie, pertinence des valeurs numériques...) et ceux de l'infographie (rapidité de calcul, fidélité visuelle...) sont très différents et nécessitent des méthodes adaptées. Ainsi je présente quelques critères de classification qui permettent de se positionner face au grand nombre d'approches existantes. Aussi, au niveau de mon projet il est important de cibler ces besoins pour traiter efficacement le problème.

### 2.2.1 Préservation ou non de la topologie

Les algorithmes qui préservent la topologie (ex : décimation), tendent à bien conserver la qualité visuelle du maillage. Par contre cette contrainte limite fortement les possibilités de simplification notamment avec des objets possédant beaucoup de "trous". De plus les tests sur la topologie peuvent être très coûteux et les maillages sur lesquels l'algorithme commence doivent être manifold.

Les algorithmes qui modifient la topologie (ex : vertex clustering) peuvent, à l'inverse, effectuer des simplifications menant à des maillages très grossiers, en comblant éventuellement des trous du maillage. Aussi les données initiales ne sont pas contraintes à être manifold, ce qui augmente leur intérêt surtout dans le domaine de la CAD. Les performances sont souvent bonnes d'où leur utilité dans les applications temps réel. Par contre la qualité visuelle en subit les conséquences.

### 2.2.2 Sélection d'un sous-ensemble de sommets ou ré-échantillonnage

Dans certains cas où des données scalaires ou vectorielles sont attachées aux maillages il n'est pas possible de ré-échantillonner car le calcul sur ces valeurs n'est pas sûr ni pertinent. Par contre une meilleure approximation est obtenue lorsque les sommets sont ré-échantillonnés vers une position qui minimiserait une mesure d'erreur.

### 2.2.3 Condition d'arrêt

Il y a deux façons d'envisager l'arrêt de la procédure de simplification. Premièrement en fournissant une borne d'erreur  $\epsilon$ , l'objectif est de construire un maillage de taille minimale qui satisfasse la précision  $\epsilon$ . Deuxièmement, en fournissant une taille (ex : nombre de sommets), l'objectif est de minimiser l'erreur ou la différence entre le maillage original et simplifié.

## 2.3 Évaluation de la mesure d'erreur

L'aspect fondamental lors de la simplification de maillage est le choix de la mesure d'erreur. Il existe un si grand nombre de méthodes qu'il serait difficile de toutes les énumérer. Cependant il est intéressant de regarder comment ces méthodes bornent et contrôlent l'erreur d'approximation. Nous pouvons les grouper en trois classes, exposées dans les paragraphes suivants.

### 2.3.1 Évaluation locale

L'erreur sur l'approximation est connue autour de l'entité étudiée. En principe ces mesures sont réalisées pour des méthodes itératives où les deux maillages  $T_i$  et  $T_i'$  sont sensiblement proches. Deux approches peuvent être mises en évidence pour mesurer la distance entre ces deux maillages :

**Calcul approché :** effectué, par exemple, par une mesure de distance entre les sommets du maillage simplifié et un plan moyen de la surface originale [SZL92].

**Calcul précis :** basé sur l'observation que les projections mutuelles de  $T_i$  et  $T_i'$  subdivisent la surface avec une géométrie qui varie linéairement [BS96]. Ainsi il suffit de calculer le maximum de cette erreur aux points d'intersection de ces arêtes projetées et cela donne une borne supérieure de l'erreur locale.

### 2.3.2 Accumulation de l'erreur locale

A chaque itération, la méthode somme pour chaque nouveau sommet et face de  $T_i'$ , l'évaluation locale de l'approximation de l'erreur et l'erreur maximale associée aux éléments de  $T_i$  [BS96].



### 2.3.3 Évaluation globale

L'estimation de l'approximation est obtenue en comparant  $T_i'$  avec la section correspondante du maillage initial  $M_0$ . Encore une fois de nombreuses méthodes ont été développées qui se regroupent en deux catégories :

**Calcul approché :** Si tous les sommets supprimés sont stockés avec la face  $f$  dans laquelle ils se projettent (ex : face la plus proche de ces sommets), une erreur globale est la distance maximale de ces sommets vers cette face  $f$  [CCMS97]. Ce critère est rapide mais fournit une sous-estimation car la distance  $L_\infty$  entre les deux surfaces peut ne pas être sur les sommets de la surface initiale. La plupart des mesures d'erreur liées aux contractions d'arêtes se retrouvent dans cette catégorie.

**Calcul précis :** La distance de Hausdorff entre le maillage simplifié et le maillage original est utilisée. L'annexe A définit précisément cette notion. Ce calcul de distance peut être effectué en mémorisant pour chaque face du maillage simplifié, toutes les faces du maillage original qu'elle représente [KLS96]. Dans ce cas par contre, le temps de calcul est proportionnel à la taille du maillage  $T_i'$  ce qui en fait une méthode plus lente mais plus précise. Cette méthode étant très répandue et servant de référence, le paragraphe suivant la décrit plus en détail.

## 2.4 Mesures d'erreur spécifiques aux contractions d'arêtes

Ces mesures d'erreur développées pour des contractions d'arêtes de triangles sont intéressantes même pour le cas de tétraèdres, c'est pourquoi je détaille leur calcul. Toutes ces méthodes ordonnent la liste des arêtes candidates à la contraction en leur affectant un coût. Pour chaque itération de la simplification, il s'agit de déterminer, l'arête dont la contraction induira l'erreur minimale.

### Hoppe

Le coût de suppression d'une arête est basé sur une énergie égale à la somme de quatre facteurs :

$$E(M) = E_{dist}(M) + E_{spring}(M) + E_{scalar}(M) + E_{disc}(M) \quad (2.1)$$

$E_{dist}(M)$  mesure l'erreur de distance entre le maillage simplifié et le maillage initial.  $E_{spring}(M)$  régule la longueur des arêtes du maillage simplifié.  $E_{scalar}(M)$  prend en compte les erreurs d'approximations faites sur les valeurs scalaires éventuellement attachées au maillage.  $E_{disc}(M)$  est destiné à préserver les discontinuités dans ces valeurs souvent très visibles. Les résultats de cette mesure sont très précis par contre les temps de calcul sont élevés.

### Guézic

La méthode introduite dans [Gué95] et [Gué96] maintient un volume de tolérance autour de la surface, approximé de telle sorte que la surface originale soit bien contenue dans ce volume. Ce volume est défini par des sphères situées en chaque sommet de l'approximation. L'enveloppe

convexe de ces sphères et des faces du modèle crée ce que l’auteur appelle des “fat triangles” qui comprend ce volume de tolérance. Les sommets de la surface approximée sont placés afin de préserver le volume de l’objet. Cette mesure est légèrement plus rapide à calculer, mais reste toujours assez lourde.

### **Ronfard et Rossignac**

L’algorithme [RR96] associe un ensemble de plans à chaque sommet, et l’erreur à ce dernier est défini comme le maximum de la distance au carré entre ce point et les plans. Ces plans sont associés lorsque les sommets sont contractés. Bien qu’elle apparait être moins précise, la distance de Hausdorff locale peut aussi être utilisée. Cette mesure produit des résultats de qualité convenable avec des temps de calcul un peu plus rapide que les méthodes exactes.

### **Garland et Heckbert**

Cette mesure d’erreur quadrique est aussi définie en terme de distance à des plans. Cependant, les auteurs utilisent dans [GH97] une représentation bien plus efficace. A chaque sommet est associé une matrice carrée symétrique  $4 \times 4$  qui mesure la somme au carré des distances de ce point avec tous les plans de l’ensemble. Certains problèmes liés aux valeurs et vecteurs propres de ces matrices pour des surfaces à faible courbure sont traités par le calcul de la courbure principale et des directions principales. Les résultats sont de bonnes qualités malgré les importantes approximations choisies et les temps de calcul sont excellents.

### **Lindstorm et Turk**

L’approche dite “memoryless” de [LT98] diffère des précédentes dans le sens où la décision n’est effectuée que sur la surface simplifiée. Aucune information sur la surface originale n’est conservée. Ils utilisent des contraintes linéaires, basées sur la conservation de volume, afin de sélectionner l’arête à contracter et la position du sommet résultant. En fait la partie “optimisation du volume” de la mesure d’erreur est une extension de la mesure d’erreur quadrique. Les résultats obtenus sont bons au niveau qualitatif et temps de calculs. Cette méthode est un peu plus longue que la précédente mais utilise beaucoup moins de ressource mémoire.

# Chapitre 3

## L'état de l'art en simplification volumique

A l'inverse de la simplification surfacique, la simplification volumique est beaucoup plus récente et donc il existe moins de méthodes la traitant. Ce chapitre introduit les origines de la simplification volumique en rappelant les premières contributions, puis s'intéresse à trois approches différentes. Je détaille ces techniques puisqu'elles ont un intérêt direct avec mon projet de DEA, particulièrement au niveau des tests topologiques et des mesures d'erreur.

### 3.1 Les origines

Les premiers travaux de simplification volumique par une approche de décimation ont été proposés par Renze et Oliver [RO96], utilisant la suppression de sommet. Etant donné un complexe tétraédrique  $\Sigma$ , les auteurs évaluent les sommets internes à supprimer de ce maillage de façon aléatoire. La re-triangulation d'un trou laissé par la suppression d'un sommet  $v$  est effectuée par une triangulation de Delaunay  $\Sigma_v$  des sommets adjacents à  $v$ . L'algorithme cherche si il existe un sous-ensemble de tétraèdres tel que ses  $(d-1)$ -faces correspondent avec les faces de  $\Sigma$ . Si ce sous-ensemble n'existe pas le sommet n'est pas supprimé. Malheureusement si le maillage initial n'est pas de Deulaunay alors ce test ne succède que très rarement. De plus l'algorithme ne mesure pas l'erreur d'approximation introduite lors de la simplification et donc n'essaie pas de sélectionner des sommets qui minimiseraient cette erreur.

Popovic et Hoppe [PH97] ont étendu leur approche des maillages progressifs [Hop96] basée sur des contractions d'arêtes à des complexes simpliciaux plus généraux. Cependant ces travaux ne prennent pas du tout en compte les erreurs commises sur les données numériques attachées au maillages et donc ils ne fournissent pas une approche efficace dans le cas de la visualisation scientifique.

Après 1997, l'intérêt porté à ce domaine grandit et différentes méthodes voient le jour. Il est intéressant de noter que les deux premières approches présentées ci-après sont une extension de la simplification surfacique. Pour chacune d'entre elles je m'intéresse essentiellement aux tests effectués sur la topologie et aux choix des critères pour calculer la mesure d'erreur liée à la contraction.

## 3.2 Extension des maillages triangulaires progressifs

### 3.2.1 Représentation progressive (PM) de Hoppe

En 1996, une nouvelle notion fondamentale apparaît dans le domaine du traitement de gros maillage. Hugues Hoppe [Hop96] introduit un nouveau modèle appelé *progressive mesh (PM)* permettant le stockage et la transmission de maillages triangulaires arbitraires. Le but de cette optimisation est de préserver non seulement la géométrie du maillage original mais aussi la cohérence des attributs attachés à la géométrie comme les couleurs, les normales, etc... Pour construire cette représentation depuis un maillage arbitraire, l'auteur utilise la *contraction itérative d'arêtes*. Dans le cadre de la simplification de maillages tétraédriques, ces opérations de simplifications sont intéressantes, cependant la fonction de coût utilisé pour ordonner ces contractions doit être adaptée.

Cette représentation progressive de maillage (PM) est définie par  $(M^0, \{vsplit_0, ..., vsplit_{n-1}\})$  et est obtenue en deux phases distinctes. La première phase de construction, coûteuse en temps, consiste à déterminer une séquence de contractions d'arêtes donnant de bonnes approximations dans les résolutions de plus en plus grossières. A la suite de cette phase de prétraitement, les opérations de contractions d'arêtes sont stockées en mémoire. Dès lors toutes les reconstructions ou simplifications peuvent être effectuées de façon interactive, dans la phase d'utilisation du modèle.

### 3.2.2 Adaptation aux maillages tétraédriques

La représentation progressive avec cette opération de contraction d'arêtes peut être adaptée à la simplification volumique. Cependant la fonction de coût doit être adaptée puisqu'elle est définie pour des triangles et des surfaces. Oliver Staadt et Markus Gross redéfinissent dans [SG98] une fonction de coût basée sur des critères volumiques (conservation du volume, forme des tétraèdres, etc...).

### 3.2.3 Définition du problème

La méthode [SG98] introduit la notion de *tétraédrisation progressive* générée à partir de contractions d'arêtes. Les auteurs définissent une fonction de coût appropriée ainsi qu'un ensemble de tests assurant la consistance du maillage au cours de la simplification. Les nouveautés par rapport à l'étude effectuée par Hugues Hoppe dans [Hop96] sont la préservation du volume des tétraèdres, l'estimation du gradient des fonctions scalaires attachées au maillage et un contrôle possible sur l'*aspect ratio* des tétraèdres. De plus cette méthode conserve les propriétés topologiques lors de la simplification, c'est-à-dire que toutes les instances de  $M$  sont homéomorphiques. Pour garantir la construction de maillages consistants, les aspects suivants sont étudiés :

- Nouvelle fonction de coût associée à l'erreur.
- Détection des arêtes à la frontière qu'il est nécessaire de conserver.
- Détection des intersections et inversions de tétraèdres au voisinage de la contraction.

La méthode de contraction d'arêtes est simple et utilisable pour simplifier un volume. Cependant son utilisation de façon brutale sur un maillage peut introduire de nombreux problèmes : inversion de tétraèdres (flipped), intersection ou dégénérescence de tétraèdres. La mesure d'erreur sur les contractions doit dès lors pénaliser des arêtes qui introduisent ces problèmes.

### 3.2.4 Fonction de coût de la contraction

Pour chaque arête  $e_i = (v_a, v_b)$  l'opération de contraction  $ecol_i(a, b)$  reçoit le coût suivant :

$$\Delta E(e_i) = \Delta E_{grad}(e_i) + \Delta E_{vol}(e_i) + \Delta E_{equi}(e_i) \quad (3.1)$$

$\Delta E_{grad}(e_i)$  mesure la différence de la fonction scalaire le long de l'arête. Ainsi une arête ayant une grande différence scalaire recevra un coût élevé.  $\Delta E_{vol}(e_i)$  pénalise les arêtes qui engendrent, après contraction, un important changement de volume.  $\Delta E_{equi}(e_i)$  permet de favoriser la forme équilatérale des tétraèdres. De plus selon les cas et les applications, un terme  $\Delta E_{edgelen}(e_i)$  peut être utilisé pour favoriser la contraction des petites arêtes.

### 3.2.5 Tests supplémentaires

Malheureusement, l'ordre de priorité créé dans par cette fonction de coût ne résout pas tous les problèmes rencontrés lors de la simplification. En effet, comme introduit précédemment, la contraction d'arêtes peut amener des inconsistances dans le maillage comme la dégénérescence, le pli (*folding*) ou encore l'intersection de tétraèdres. Pour palier à ces problèmes Oliver Staadt et Markus Gross proposent une série de tests statiques et dynamiques basés sur des critères géométriques et topologiques.

#### Tests statiques

Ils détectent et pénalisent les arêtes à la frontière du volume, appelés *sharp edges*. En effet dans certains cas la suppression d'arêtes à la frontière amène des changements de topologie comme l'isolement d'un sommet. Par exemple, dans un cas en 2D, la suppression de l'arête  $(v_1, v_3)$  de la figure 3.1 entrainerait l'isolement du sommet  $v_2$  et donc un changement de topologie.

#### Tests dynamiques

Pour détecter l'inversion de tétraèdre, les auteurs étendent un résultat de R.Ronfard et J.Rossignac introduit dans [RR96] sur l'inversion de normale. Pour cela ils calculent le volume d'un tétraèdre comme étant le produit mixte de 3 arêtes  $e_i, e_j$  et  $e_k$  :

$$\frac{1}{6}[e_i, e_j, e_k] = \frac{1}{6}\langle e_i \times e_j, e_k \rangle \quad (3.2)$$

Si le volume d'un des tétraèdres modifiés par la contraction devient négatif après la transformation alors il y a inversion et donc l'arête candidate est rejetée.

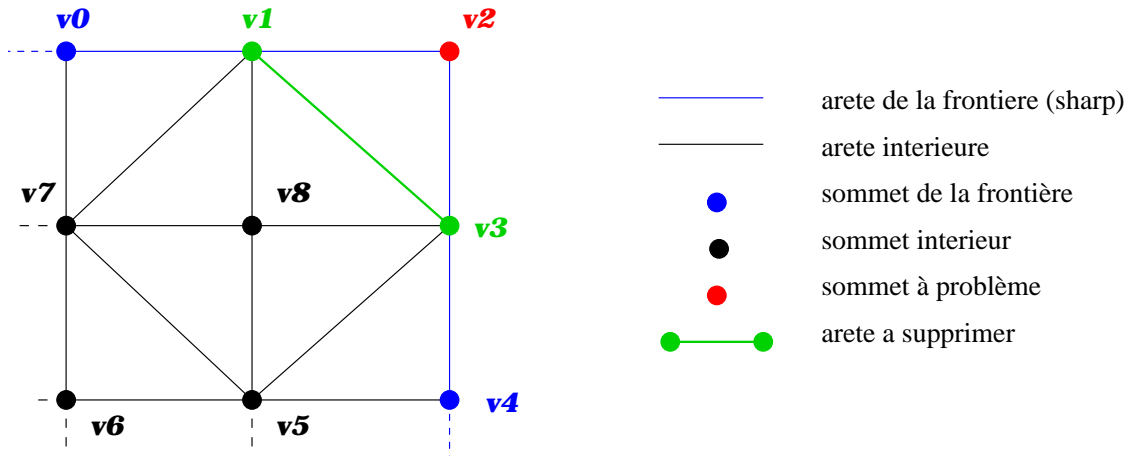


FIG. 3.1 – Problème lors de la contraction d’arêtes de bords

Un autre problème abordé dans [SG98] est l’intersection de tétraèdres après une contraction. Les auteurs proposent un algorithme pour traiter ce cas. D’après l’article, ce problème n’apparaît qu’au niveau de la frontière du volume, c’est-à-dire uniquement pour des cellules dites *sharp*. La figure 3.2 illustre ce problème.

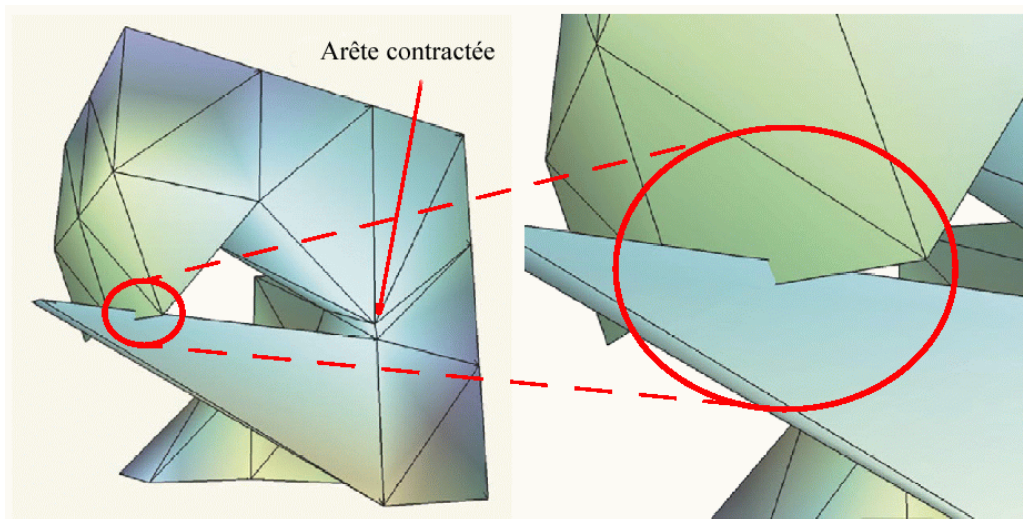


FIG. 3.2 – Intersection de cellules à la frontière du volume

La détection est effectuée par un test d’intersection triangle-triangle entre les faces des cellules de la frontière et les cellules modifiées. Ce test débute au voisinage de la contraction puis se propage par voisinage. Théoriquement il faudrait faire un test exhaustif sur toutes les cellules *sharp* avec chaque tétraèdre modifié par une contraction, mais en pratique après la deuxième itération, 98% des intersections sont détectées.

### 3.2.6 Détail de l'algorithme

Après avoir calculé le coût de contraction de chaque arête et déterminé les conditions du test statique détaillé précédemment, l'algorithme construit un tas avec toutes les arêtes restantes ordonné par la fonction de coût. Il choisit ensuite l'arête se trouvant au dessus du tas, et vérifie pour celle-ci les intersections et inversions. Ensuite les arêtes des tétraèdres modifiés sont mises à jour. Ce procédé est répété jusqu'à ce que le tas soit vide ou bien que le nombre de contractions voulu est atteint. A noter que dans cette méthode il n'y a pas d'optimisation du placement du sommet résultant de la contraction. Dans les tests présentés, les auteurs ont choisi de le placer au milieu de l'arête à contracter.

## 3.3 Détection efficace des intersections de bord

### 3.3.1 Définition du problème

Il est intéressant de noter que la méthode précédente effectue un test particulier pour détecter les intersections des tétraèdres au bord du volume. En effet, comme expliqué précédemment, ces intersections sont difficiles à détecter car elle ne sont pas nécessairement locales. Le test précédent est coûteux car il regarde successivement toutes les cellules de la frontière et surtout calcule des intersections triangle-triangle. Aussi pour accélérer ce dernier il n'envisage pas un parcours exhaustif mais teste juste au voisinage de la contraction ce qui amène dans certains cas des surprises.

### 3.3.2 Détection adaptée aux maillages non-convexes

Au contraire, l'article [KE00] de Martin Kraus et Thomas Ertl introduit une méthode détectant efficacement les contractions d'arêtes induisant de telles intersections. Dans un premier temps la méthode "convexifie" le maillage, c'est à dire tétraédrise (ou triangle en 2D) toutes les cellules n'appartenant pas au volume mais intérieur à l'enveloppe convexe (cellules rouges et jaunes du cas (c) de la figure 3.3). Ces nouvelles cellules introduites sont dites imaginaires. Dans ce cas de maillage convexifié, les auteurs montrent que toute intersection, après contraction est forcément accompagnée de l'inversion d'au moins une cellule imaginaire (e.g. cellule rouge des cas (c) et (d)). Les cas (a) et (b) montrent l'importance de convexifier le maillage auparavant car dans le cas non-convexe cette intersection n'est pas toujours accompagnée d'une inversion.

### 3.3.3 Limites de l'approche

La notion de maillage convexifié de [KE00] apporte aussi des informations sur la préservation de l'enveloppe convexe ou encore la préservation ou non de la topologie. Ces tests sont rapides cependant ils ne détectent pas tous les problèmes topologiques que peuvent amener une contraction d'arête. Cette approche est donc rapide mais pas rigoureuse pour une étude de la topologie. Les auteurs eux-mêmes envisagent d'intégrer à leur méthode, le calcul du lien introduit dans [DEGN98].

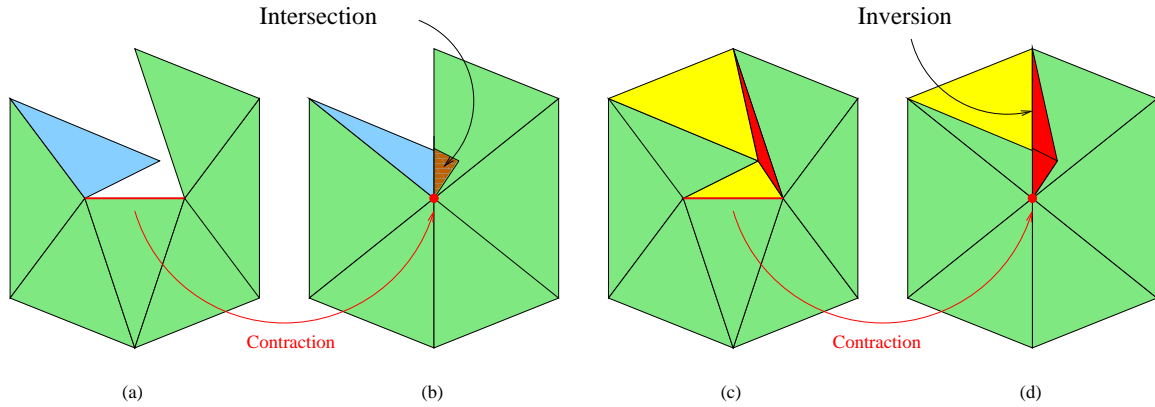


FIG. 3.3 – Contraction d’arête dans des maillages “convexifiés”

## 3.4 Simplification à l’aide de fonctions splines

### 3.4.1 Définition du problème

Il est intéressant de noter que l’article [THJ99] basé sur la contraction d’arêtes est le résultat de plusieurs travaux des mêmes auteurs. Cependant auparavant leur approche était fondée sur la contraction de simplexe de plus haut degré comme la contraction de face présentée dans [Ham94] par Bernd Hamann, ou encore la contraction de tétraèdres présentée dans [THW98] par Isaac Trotts, Bernd Hamann et Kenneth Joy. Cependant la contraction de simplexe de plus haut degré complique considérablement le contrôle de la topologie et de la consistance du maillage après chaque étape.

#### Contraction de faces

La méthode introduite dans [Ham94] suit le schéma général illustré figure 2.1. Le coût associé à la suppression d’une face est calculé en fonction de l’équi-angularité de cette face et de la courbure du maillage au voisinage de celle-ci. Plus la face sera équiangulaire, ou plus le maillage sera courbé au voisinage de celle-ci, plus le coût de suppression sera élevé. Ainsi cette méthode supprimera en priorité les longs triangles fins (critère d’équi-angularité) ou les triangles au voisinage desquelles la surface est quasi plane (critère de courbure).

Pour calculer la position géométrique du sommet venant remplacer la face supprimée, l’algorithme utilise une méthode de moindres carrés approximant la surface par une quadrique au voisinage de la face supprimée. Cette méthode nécessite la faculté de pouvoir projeter localement la triangulation sur un plan. Pourtant une telle projection peut ne pas être injective dans le sens où deux triangles de la surface d’origine peuvent avoir des images projetées qui s’intersectent, notamment dans des régions à forte courbure. C’est de ce problème qu’est née la nécessité de concevoir des méthodes plus adaptées à des maillages quelconques possédant des régions à fortes courbures comme dans le domaine de l’aéro-spatiale par exemple (référence au célèbre modèle “Blunt-fin” de la NASA Ames).



## Contraction de tétraèdres

De la même façon, la contraction de tétraèdres présentée dans [THW98] posent des problèmes similaires. Une erreur d'approximation maximale est stockée pour chaque cellule. Cette valeur plus une estimation de l'erreur due à une contraction du tétraèdre sont utilisées pour connaître la cellule à contracter et pour assurer que l'erreur induite reste en dessous d'un certain seuil. La contraction d'un tétraèdre est effectuée par contractions successives de trois de ses arêtes. Le problème apparaît à ce niveau puisque le maintien de la topologie n'est pas facilement contrôlable en contractant trois arêtes à la fois.

### 3.4.2 La méthode

Cet algorithme introduit par Issac Trotts, Bernd Hamann et Ken Joy dans [THJ99] mesure l'erreur sur les attributs scalaires attachés aux sommets du volume. Leur approche est basée sur la création d'une fonction spline linéaire définie par la triangulation d'un espace 3D. Les attributs scalaires aux sommets du maillage représentent les coefficients de la spline. De plus une mesure d'erreur est calculée et mise à jour après contraction d'une arête pour chaque tétraèdre. Une valeur seuil contrôle la résolution que l'utilisateur souhaite obtenir. Ainsi il est possible de créer une représentation multirésolution d'un maillage à l'aide de cette technique.

### 3.4.3 Mesure d'erreur adaptée

De ces constats provient donc la nécessité de baser la simplification de maillage sur la contraction d'arête pour contrôler au mieux les problèmes topologiques. La méthode utilisée dans [THJ99] est une extension du cas en 2D de la contraction d'arêtes d'une triangulation. Pour mesurer l'erreur commise par une contraction, les auteurs supposent que le maillage représente une fonction spline linéaire définie par des segments de spline  $s = F(u, v, w, t)$  avec  $(u, v, w, t)$  étant les coordonnées barycentriques d'un point à l'intérieur d'un tétraèdre. Les coefficients de  $F$  sont des attributs scalaires se trouvant au sommet du maillage. Chaque tétraèdre  $T$  possède une approximation de l'erreur  $\epsilon_T$  qui représente la différence entre le segment de spline défini par  $T$  et la spline linéaire du maillage original au voisinage de  $T$ . Si  $F^C$  est la fonction définie par morceaux induite par une contraction, la méthode fournit une borne supérieure de l'erreur sur un tétraèdre  $T^C$  modifié par cette contraction :

$$B(T^C) = \max_i \{ |F^C(c_i) - F(c_i)| + \max\{\epsilon_T\} \} \quad (3.3)$$

ou les  $c_i$  sont les points d'intersections entre les arêtes du tétraèdre modifié  $T^C$  et les faces des tétraèdres du maillage original. Le maximum est pris sur tous les tétraèdres qui contiennent  $c_i$ . La figure 3.4 illustre ce calcul. Les points  $c_2$ ,  $c_3$  et  $c_4$  sont les points d'intersection entre le triangle modifié et les triangles de la surface initiale. Pour obtenir ces points, la surface originale est projetée sur la surface simplifiée.

Le coût de contraction  $\delta_T$  d'un tétraèdre  $T$  est donné alors par :

$$\delta_T = \min [\max_j B(T_j^C)] - \epsilon_T \quad (3.4)$$

ou le minimum est donné parmi toutes les arêtes du tétraèdre  $T$  et le maximum parmi tous les tétraèdres modifiés par la contraction des six arêtes de celui-ci.

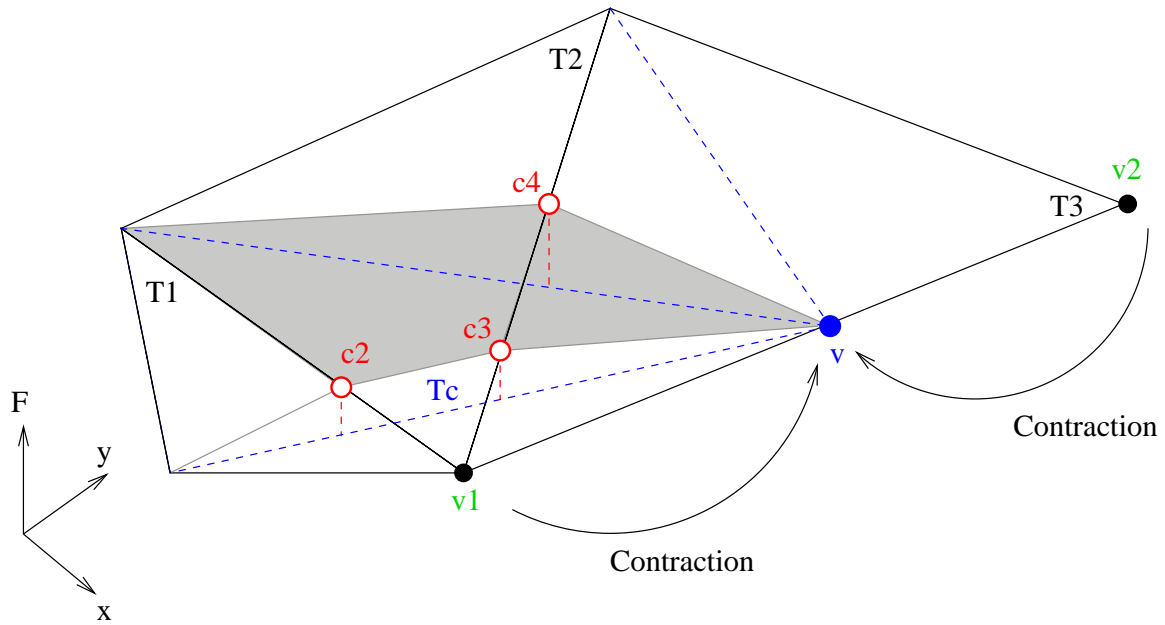


FIG. 3.4 – Calcul de l’erreur pour le triangle  $T^C$  modifié. Le triangle  $T_1$  devient  $T^C$  après la contraction de l’arête  $(v_1, v_2)$  en  $v$ . Une borne de l’erreur comise sur l’approximation des triangles  $T_1$ ,  $T_2$  et  $T_3$  est obtenu en prenant le maximum de l’erreur aux points d’intersection  $c_2$ ,  $c_3$  et  $c_4$ .

### 3.4.4 Intersection de tétraèdre

Le contrôle de l’intersection à l’intérieur de l’ensemble est effectué par le calcul du volume d’un tétraèdre avant et après une contraction. Si le signe change alors l’arête responsable est dite *non contractable*. Le cas de l’intersection de cellules de bords qui ne sont pas au voisinage de la contraction n’est pas détaillé.

### 3.4.5 Préservation de la frontière

Si l’arête à contracter a un seul sommet de bord  $v$ , alors le sommet destination après contraction doit être  $v$ . Sinon si les deux points extrêmes de l’arêtes sont à la frontière alors l’erreur commise par la contraction doit être inférieure à un seuil fourni par l’utilisateur. Dans ce cas particulier le calcul de l’erreur est effectué en ajoutant un sommet “exponentiel” à l’infini connecté à tous les sommets de la frontière. Cela ramène le calcul de l’erreur aux frontières, identique à celui d’une tétraédrisation “fermée”. Il est important de remarquer que pour cette méthode la mesure de l’erreur n’est pas la même pour le cas de tétraèdres à l’intérieur du volume et à la frontière. A l’intérieur, l’erreur est basée sur les changements du champ scalaire alors qu’à la frontière, elle repose sur une mesure de distance (distance de Hausdorff [KLS96]).

### 3.4.6 Algorithmes

Les auteurs de [THW98] proposent deux algorithmes utilisant cette mesure d’erreur. Le premier crée une liste avec priorité d’arêtes potentiellement contractables ordonnée par la pré-

diction de l'erreur  $\delta_T$  définie au paragraphe 3.4. Le second plus brutal (stratégie *gloutonne*) compare pour chaque tétraèdre si la contraction est en dessous d'un certain seuil. Les candidats passant ce test sont contractés. Dans les deux cas l'utilisateur a le contrôle de deux bornes d'erreur. La première contrôle la variation du champ scalaire et la deuxième la déviation autorisée à la frontière.

## 3.5 Simplification avec respect de la topologie

### 3.5.1 Définition du problème

Les articles traitant de simplification d'un maillage 3D sont très nombreux. Cependant ceux présentant une approche robuste aux problèmes topologiques sont plus rares. C'est pourquoi dans le cadre de mon projet de DEA, je m'intéresse particulièrement aux méthodes utilisant des notions de géométrie algorithmique. En particulier, l'équipe de chercheurs formée de P.Cignoni, D.Costanza, C.Montani, C.Rocchini et R.Scopigno s'est servie de ces méthodes pour présenter une méthode de simplification incrémentale de tétraèdres. L'article [CR00] propose par contraction d'arêtes une évaluation de la mesure d'erreur prenant en compte des modifications du domaine scalaire mais aussi du volume. Des méthodes pour préserver la topologie du maillage sont aussi présentées.

La méthode [CR00] utilise la contraction d'arêtes. Chaque arête est orientée, et mise dans un tas avec priorité variant selon la mesure d'erreur prédite pour une contraction. Au moment de l'élimination d'une arête, la première de la liste est tirée et une série de tests est réalisée pour valider l'opération ou refuser. Ces tests sont de deux natures ; le premier est topologique pour assurer le maintien de celle-ci au cours de la simplification et le second est géométrique pour éviter les intersections de tétraèdres et les cellules étirées ("slivery").

### 3.5.2 Tests topologiques

C'est dans ces tests que les notions de géométrie algorithmique rentrent en compte. En effet, des résultats robustes ont été démontrés par Tamal Dey et Herbert Edelsbrunner pour détecter les arêtes qui peuvent être contractées sans changer la topologie au sein d'un complexe simplicial. L'article [DEGN98] introduit et démontre de telles propriétés. Ces outils étant importants dans l'approche proposée durant le DEA, le chapitre 4 les détaillent.

### 3.5.3 Tests géométriques

La préservation de la géométrie peut être compromise par trois problèmes classiques : l'inversion de tétraèdres, la génération de tétraèdres étirés et l'auto-intersection des frontières. Le premier problème est détecté par le changement de signe du volume des tétraèdres modifiés par une contraction. Le second est évité par comparaison de "l'aspect ratio" d'un des tétraèdres modifiés avec une valeur seuil. Le troisième quant à lui est plus difficile à détecter car il peut ne pas être local à la contraction. Les auteurs suggèrent une approche différente de [SG98], en

envisageant l'ajout d'une nouvelle structure codant uniquement la frontière et ainsi accélérant le test. Ils n'effectuent cependant pas un test aussi poussé que celui de [SG98] car les intersections hypothétiques sont cherchées parmi les arêtes adjacentes au sommet destination de la contraction, et non sur un voisinage éloigné de la contraction.

### 3.5.4 Mesure d'erreur liée au domaine

La contraction d'une arête adjacente ou sur la frontière peut amener une déformation du domaine. Autrement dit la silhouette de  $\Sigma_i$  et  $\Sigma_{i+1}$  est différente. Pour remédier à ce problème, peu abordé dans les autres approches, les auteurs utilisent la distance symétrique de Hausdorff entre la peau (la surface de la frontière) de  $\Sigma_i$  et celle de  $\Sigma_{i+1}$ . Le calcul précis de cette distance étant coûteux, ils utilisent une approximation de cette mesure dans leur démarche.

### 3.5.5 Mesure d'erreur liée au champ scalaire

La mesure de l'erreur sur le champ scalaire peut être obtenue en prenant le maximum de la différence entre le champ du maillage original et celui du maillage à une résolution donnée. Cependant une grande variation du champ scalaire même pour un tétraèdre ayant un petit volume conduira à une forte erreur. Il est donc nécessaire de prendre en compte l'influence de la contraction au niveau du volume global (en calculant l'intégrale sur le volume de l'erreur au carré). A noter que cette méthode devient plus longue à mettre en place lorsque les arêtes de bords sont contractables car le volume global change et donc il faut d'abord se ramener au même espace.

### 3.5.6 Prises en compte de toutes ces mesures d'erreur

L'approche classique consiste à calculer une somme pondérée par différentes erreurs :  $\varepsilon = w_1\varepsilon_1 + \dots + w_k\varepsilon_k$  comme dans [SG98] ou [THJ99]. Mais il arrive que la solution optimale de la minimisation de cette fonction possède tout de même une bien trop grande erreur sur le champ scalaire. C'est l'exemple du point  $c$  de la figure 3.5 où l'erreur sur le domaine est minimisée mais par contre l'erreur sur les données scalaires est maximale. C'est pourquoi les auteurs proposent de choisir l'erreur  $\varepsilon$  qui vérifie

$$\varepsilon = \min ( \max(w_d\varepsilon_d(e), w_f\varepsilon_f(e)) ) \quad (3.5)$$

défini dans un espace normalisé et avec des poids  $w_d$  et  $w_f$  vérifiant  $w_d\varepsilon_d^{max} = w_f\varepsilon_f^{max}$ . Intuitivement les arêtes contractées en priorité (point  $a$  de la figure 3.5) sont celles ayant leur erreur  $\varepsilon$  se trouvant le long de la diagonale du carré formé par les axes  $\varepsilon_d, \varepsilon_f$  et le point  $(w_f\varepsilon_f^{max}, w_d\varepsilon_d^{max})$ . Cette méthode est facilement généralisable à  $k$  dimensions.

Dans [CR00] les auteurs utilisent une version optimisée de ces méthodes. A savoir que les calculs ne font pas intervenir d'intégrale ni de produit.

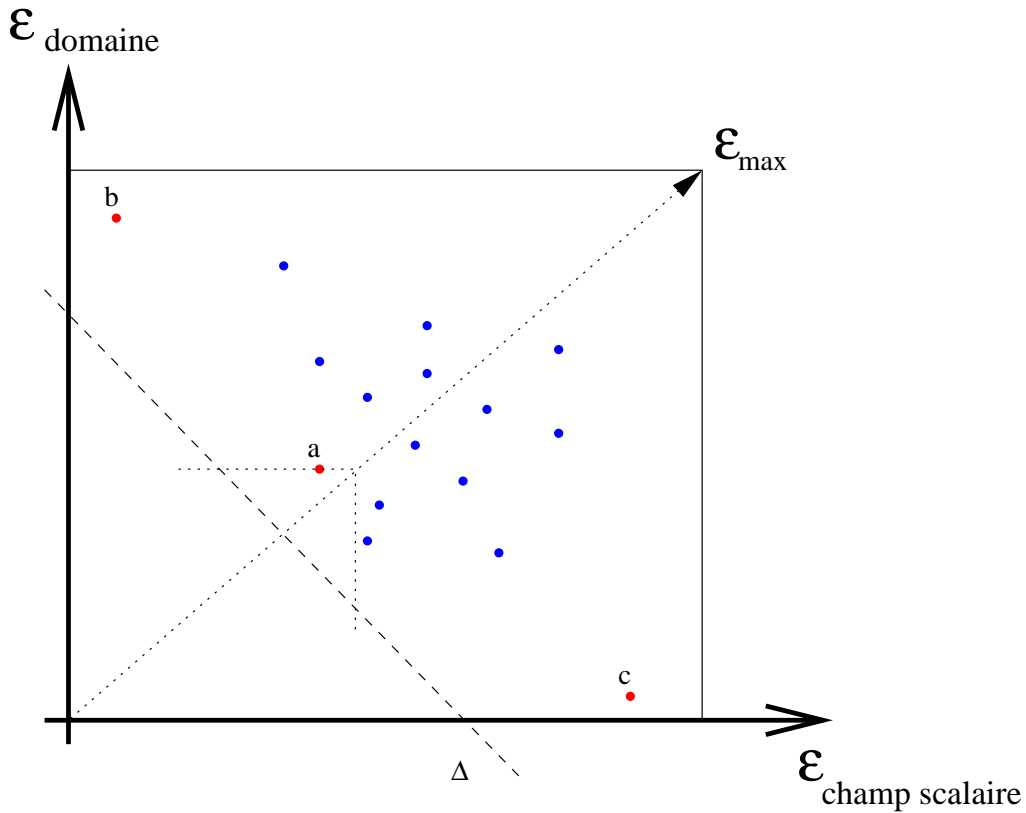


FIG. 3.5 – L’espace d’erreur du domaine et du champ scalaire attaché aux données. Lors du processus de simplification, l’erreur va de l’origine vers le point d’erreur maximale  $\epsilon_{max}$  défini par l’utilisateur

### 3.5.7 Prédiction de l’erreur

Pour accélérer la simplification, l’erreur sur une contraction peut être estimée. Dans ce cas, le calcul du gradient en chaque sommet ou une erreur quadratique peuvent être utilisés.

## 3.6 Computational Topology

Cette étude détaillée sur les différentes techniques de simplification met bien en évidence que les tests géométriques ne suffisent pas pour conserver la consistance d’un maillage volumique. C’est pourquoi une étude supplémentaire liée à la topologie a suscité un intérêt croissant depuis 1998. Notamment Tamal Dey et Herbert Edelsbrunner s’affichent comme les précurseurs d’un nouveau domaine de recherche qu’ils nomment la *topologie combinatoire* exposé dans [DEG99]. L’étude rigoureuse de ce domaine basé sur la géométrie algorithmique et combinatoire, est bien au-delà de mon projet de DEA sur la simplification de maillages volumiques. Néanmoins il est intéressant de noter les contributions de ce domaine à celui de la simplification volumique.

La contribution la plus illustrative est l’article [DEGN98] qui introduit des méthodes rigou-

reuses pour détecter des arêtes qui altèrent la topologie lors d'une contraction. Cette détection est réalisée par le calcul du lien d'un sommet et d'une arête. Je précise cette notion au paragraphe 4.1. Cette notion est utilisée entre autres par [CR00], [KE00] et s'est révélée être un outil efficace pour traiter les problèmes topologiques.

# Chapitre 4

## Contributions

### 4.1 Préservation de la topologie

#### 4.1.1 Les contraintes

En visualisation scientifique, la topologie est une notion qu’il faut étudier de près. En effet, de nombreuses particularités des ensembles volumiques sont créées par la présence de ‘trous’ ou d’un nombre bien défini de composantes connexes. La modification de ces caractéristiques peuvent fausser considérablement l’exploitation finale des résultats. Notamment, dans le cadre de mon projet, les données traitées ne doivent pas être approximées de façon grossière sur des critères visuels. La SER (défini au paragraphe 1.2) calculée pour chaque point du volume requiert des heures de calcul et donc les physiciens exploitant les données simplifiées veulent disposer de valeurs exactes sur un maillage ayant la même topologie initiale.

Pour pouvoir contrôler rigoureusement la topologie au cours de la simplification, nous avons utilisé une méthode introduite par Tamal Dey dans [DEGN98]. Cette notion utilise un théorème de géométrie algorithmique, très utile mais peu intuitif. Comme cet outil est fondamental dans le développement de notre propre méthode de simplification, je le détaille dans la suite. Aussi durant mon projet, j’ai implémenté un outil qui permet de visualiser ce lien sur des maillages tétraédriques afin de se persuader de son utilité.

#### 4.1.2 Définition du lien

Soit  $D = (V, \Sigma, \Phi)$  le maillage ou  $V$  est l’ensemble des  $n$  sommets,  $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_m\}$  est la tétraédrisation de  $m$  cellules avec des sommets de  $V$  et  $\Phi = \{\phi_1, \phi_2, \dots, \phi_m\}$  est un ensemble de fonctions tel que chaque  $\phi_i$  est définie sur la cellule  $\sigma_i$  de  $\Sigma$ . L’étoile d’un tétraèdre  $St(\sigma)$  est l’ensemble des co-faces de  $\sigma$ , c’est à dire que  $St(\sigma) = \{\tau \in \Sigma \mid \sigma \text{ est une face de } \tau\}$ . Le lien d’un tétraèdre  $Lk(\sigma)$  est l’ensemble de toutes les faces de  $St(\sigma)$  non incident à  $\sigma$ . En d’autres termes  $Lk(\sigma)$  représente l’ensemble de toutes les faces des co-faces de  $\sigma$  disjoint de  $\sigma$ . La figure 4.1

illustre cette définition en deux dimensions. L’extension aux maillages tétraédriques est directe mais difficile à dessiner. Les arêtes et faces dessinées en bleu sont les éléments de l’étoile tandis que les éléments en rouge sont ceux du lien.

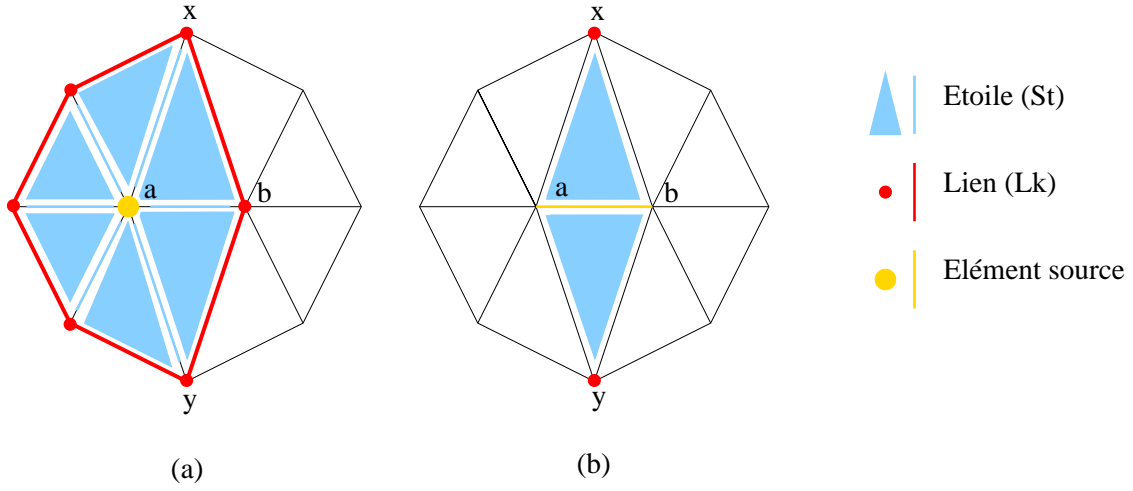


FIG. 4.1 – Définition du lien d'un sommet et d'une arête

### 4.1.3 Propriété fondamentale

Soit  $\Sigma_i$  un 3-complexe simplicial sans frontière,  $e = (v_s, v_d)$  une arête du complexe, et  $\Sigma_{i+1}$  le complexe après la contraction de l'arête  $e$ . D'après [DEGN98], les énoncés suivants sont identiques :

1.  $Lk(v_s) \cap Lk(v_d) = Lk(e)$
2.  $\Sigma_i, \Sigma_{i+1}$  sont homéomorphes

Il est donc suffisant de vérifier l'énoncé (1) pour prouver l'énoncé (2) et donc d'assurer la conservation de la topologie après contraction de l'arête  $e$ . Avec une structure de donnée appropriée, il n'est donc pas difficile de calculer ce lien dynamiquement. La figure 4.2 montre ce problème avec dans le deuxième cas une contraction ne préservant pas la topologie.

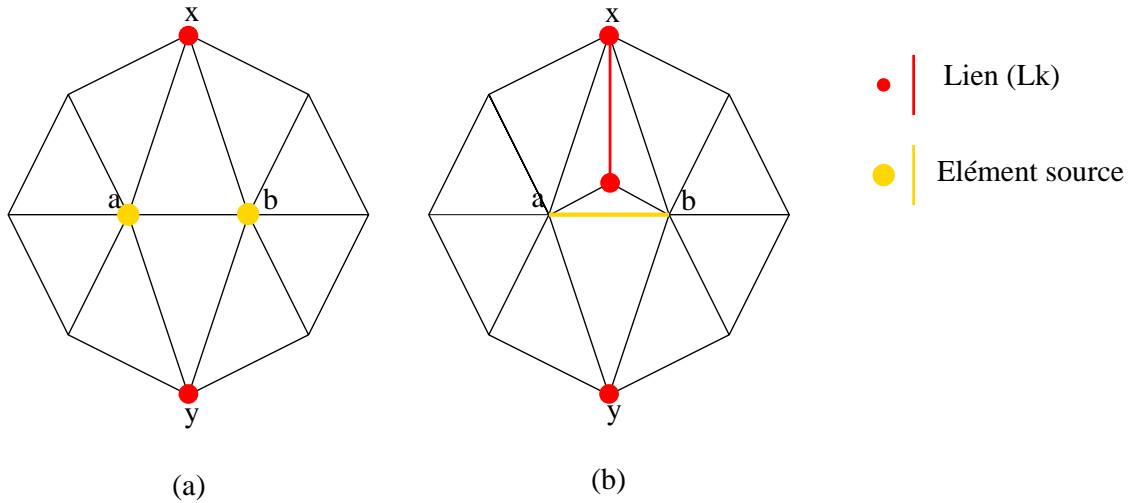


FIG. 4.2 – Tests topologiques : A gauche, la condition  $Lk(a) \cap Lk(b) = \{x, y\} = Lk(ab)$  indique une contraction d'arête valide. Au contraire, une contraction invalide est détectée dans la configuration de droite car  $Lk(a) \cap Lk(b) = \{x, y, z, zx\} \neq Lk(ab)$



A noter que cette propriété n'est valable que pour un simplexe sans frontière, et qu'en général les  $\Sigma_i$  sont des complexes avec frontière. Cependant, il est possible de se ramener au cas de simplexes sans frontière en ajoutant un sommet arbitraire dit à l'infini et son cône de simplexes. Autrement dit, il faut créer autant de cellules fictives qu'il y a de 2-faces à la frontière du volume  $\Sigma_i$  (un tétraèdre fictif étant composé des 3 sommets de la face de la frontière et le sommet à l'infini). La figure 4.3 montre tous ces simplexes fictifs. Par la suite ils n'apparaissent pas puisque ce sommet n'est pas réel.

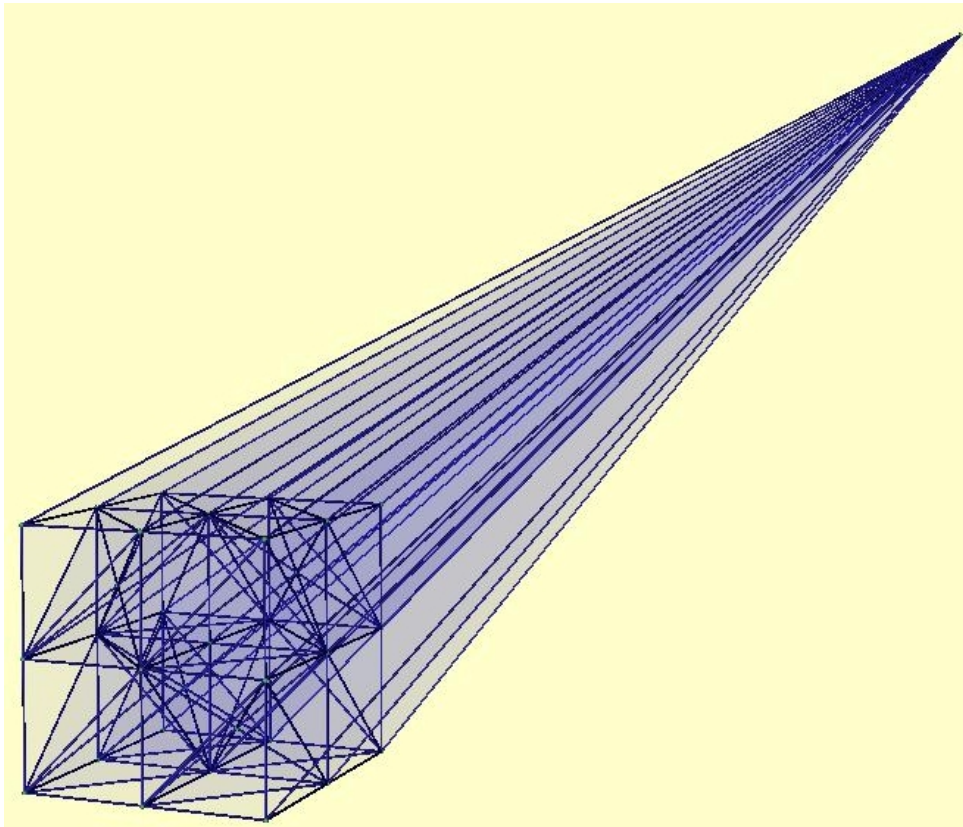


FIG. 4.3 – Sommet fictif utilisé dans le cas de surface à bord

L'article [DEGN98] apporte toute la démonstration rigoureuse de cette propriété faisant appel à des notions complexes de géométrie combinatoire introduites pour des 2-complexes puis généralisées aux 3-complexes. Il est intéressant de noter que la généralisation de cette propriété à des complexes de dimensions supérieures est encore un problème ouvert. Afin de visualiser le problème, j'ai implémenté une application montrant ce lien pour un sommet ou une arête choisis dynamiquement par l'utilisateur. La photo d'écran 4.4 montre cette application.

#### 4.1.4 Implémentation

Le lien étant un critère local, il est obtenu à l'aide des éléments au voisinage de la contraction. La validation de la contraction s'effectue donc par calculs successifs d'ensembles de  $k$ -faces (sommet, arête, face) :

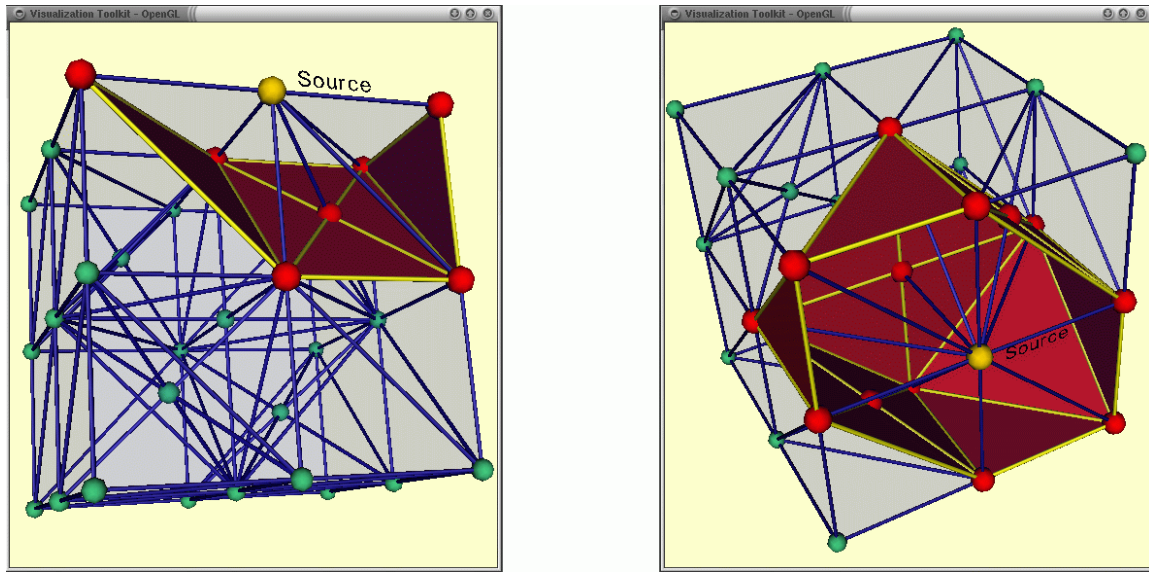


FIG. 4.4 – Application pour illustrer le lien

- calcul du lien du sommet source,  $Lk(v_s)$
- calcul de l'intersection entre  $Lk(v_s)$  et le lien du sommet destination,  $Lk(v_s) \cap Lk(v_d)$
- calcul du lien de l'arête formée du sommet source et destination,  $Lk((v_s, v_d))$
- comparaison entre  $Lk((v_s, v_d))$  et  $Lk(v_s) \cap Lk(v_d)$ .

Il est intéressant de noter que cet algorithme ne dépend pas de la taille du maillage mais uniquement du nombre de cellules adjacentes en chaque sommet. Cette méthode est donc adéquate pour le traitement de gros ensembles de données. En pratique, dans les maillages utilisés pour mon projet, environ vingt-cinq cellules se partagent un même sommet.

## 4.2 Technique de visualisation

### 4.2.1 Problématique

Bien que la visualisation des données à exploiter soit importante, mon projet n'a pas pour objectif de développer des techniques particulières. En effet, nous nous intéressons au traitement de ces données avant toute exploitation visuelle par des méthodes de visualisation scientifique comme les isosurfaces ou le rendu volumique. De plus les membres du CEA avec qui nous collaborons, ne sont pas nécessairement formés en visualisation et donc sont intéressés par des outils de haut niveau. Dans cette optique nous avons choisi en accord avec les membres du CEA d'utiliser une librairie graphique offrant un ensemble de techniques standard de visualisation scientifique (isosurface, volume rendering, ...) nommée VTK<sup>1</sup>.

<sup>1</sup>The Visualization Toolkit (<http://public.kitware.com/VTK/>)

### 4.2.2 VTK : The Visualization Toolkit

VTK est une librairie graphique gratuite, portable, orientée objet pour développer des logiciels d'infographie, de visualisation ou de traitement d'images. Elle fournit une grande variété de structures de données codant des ensembles utiles en visualisation comme les ensembles de points, les grilles structurées, curvilinéaires ou instructurées. De plus elle apporte une grande quantité de procédures de lecture/écriture en de très nombreux formats pour permettre l'importation et l'exportation des formats de codages usuels (PLOT3D, IV, VRML, ...). Afin de traiter efficacement ces données, des filtres sont disponibles agissant sur la géométrie ou les données numériques des maillages. Cette partie de la librairie est particulièrement bien fournie apportant un choix considérable et surtout efficace de méthodes. Actuellement VTK est distribuée comme une surcouche d'OpenGL pour se plier au standard et assurer une grande portabilité. Les auteurs de cette librairie ont publié plusieurs ouvrages dont [SML98] qui est particulièrement utile à son apprentissage.

### 4.2.3 Développement rapide

A l'aide de cette librairie, il est possible de concevoir en quelques lignes des outils de visualisation très spécialisés. A titre d'exemple la figure 4.5 montre des photos d'écrans de ce type d'application. Depuis le volume de données Bluntfin, représenté en (a), avec sa surface et un champ scalaire distribué sphériquement, il est possible d'obtenir des informations visuelles sur le champ scalaire à l'intérieur du volume assez simplement (b). La technique utilisée est une simple isosurface représentant un isopotential que l'utilisateur peut faire varier dynamiquement (c).

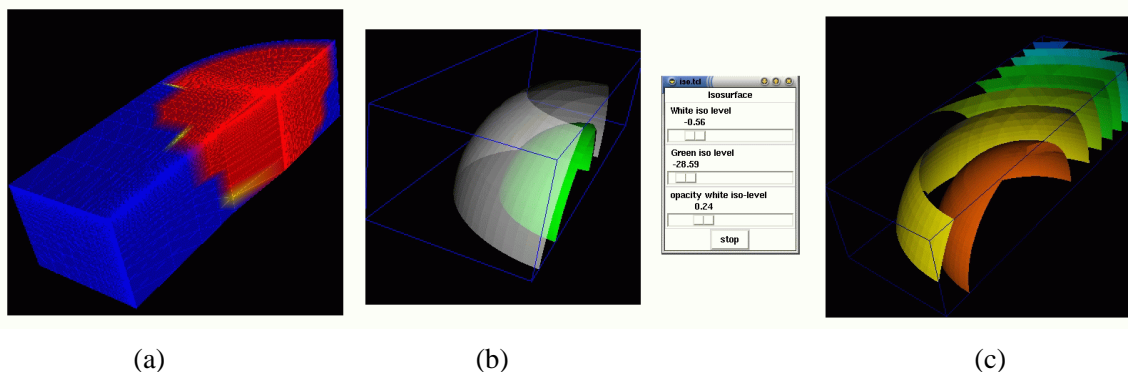


FIG. 4.5 – Outils adaptés de visualisation

L'annexe B montre le code suffisant à implémenter une telle application (environ une quarantaine de lignes). Ce code très simple est facilement modifiable pour ajouter d'autres éléments de l'ensemble. Les principales étapes de la construction de ce type d'outils sont :

- Acquisition des données (*reader*)
- Création de l'objet à visualiser, par exemple une isosurface (*Contour filter*)
- Transformation de cet objet en primitives géométriques simples (*Mapper*)
- Réglage de l'apparence de ces primitives (*Actor*)
- Affichage des informations dans une fenêtre graphique (*Renderer*)

## 4.3 Occupation de l'espace mémoire

### 4.3.1 Problématique

Non seulement les ensembles volumiques traités sont de très grande taille mais en plus leur exploitation requiert beaucoup d'informations pour chaque entité. Par exemple il faut être capable pour une cellule de pouvoir retrouver ses voisins ou encore à partir d'un sommet connaître les cellules auxquelles il appartient. Les choix à faire pour une implémentation adaptée doivent donc tenir compte à la fois de l'espace mémoire et du temps d'accès à chaque entité.

### 4.3.2 Outils disponibles

Dans l'approche proposée, nous avons décidé d'utiliser une structure de données adaptée aux besoins de la simplification, plutôt que des outils déjà existants. Par exemple, la librairie CGAL<sup>2</sup> fournit une implémentation efficace pour la création et l'exploitation de toute sorte de triangulations 3D. Le niveau d'abstraction supplémentaire que fournit cette librairie, rend la manipulation de triangulations relativement simple. Cependant, dans le cadre d'un projet bien défini, avec des problèmes critiques d'occupation mémoire, cet outil n'est pas forcément optimal.

De la même façon la librairie graphique VTK, fournit elle aussi une structure adaptée aux maillages structurés. Le problème avec celle-ci est que son implémentation a été faite par stockage des valeurs dans des tableaux contigus. L'occupation mémoire ainsi que les temps d'accès sont bons, cependant l'aspect contigu la rend totalement inutilisable pour une application de simplification. En effet, la suppression d'un élément nécessiterait de combler l'espace par déplacement successif de tous les éléments suivants. Ainsi, dans l'approche adoptée, VTK n'est utilisée que pour l'aspect visualisation et non pour sa structure de données.

### 4.3.3 Approche choisie

L'objectif de l'approche utilisée, était de maîtriser et surtout d'adapter la structure de données au problème spécifique de la simplification de maillage. En effet, les travaux sur les structures de données de triangulation sont nombreux cependant ceux étant optimisés pour la simplification avec une gestion efficace de la mémoire sont plus rares. La figure 4.6 montre le choix de structure de données adoptée.

L'algorithme de simplification est basé sur la sélection itérative d'arêtes selon un critère de priorité. Cet aspect nécessite une structure de tas adaptée. Premièrement, il faut être capable d'ordonner les arêtes selon une priorité. Ensuite, à partir d'une arête de l'ensemble volumique, il faut pouvoir la localiser rapidement ( $O(1)$ ) dans le tas, modifier sa priorité (entraînant un changement de position) et enfin la supprimer. Il est donc nécessaire de mettre en place une structure doublement chaînée entre le tas et l'ensemble volumique assurant un temps constant d'accès aux éléments.

---

<sup>2</sup>Computational Geometry Algorithms Library (<http://www.cgal.org/>)

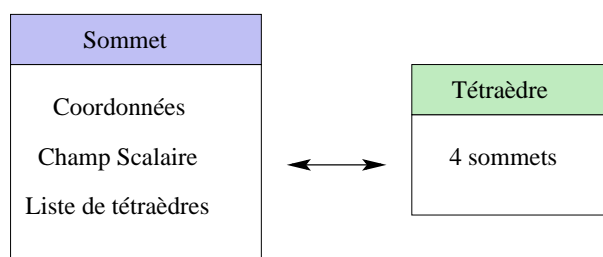


FIG. 4.6 – Structure de données de la tétraédrisation

Ces choix favorisent le temps d'accès aux éléments plutôt que l'espace mémoire. Cependant, cette approche est nécessaire pour pouvoir traiter des ensembles de grande taille en un temps convenable et surtout qui ne croît pas dramatiquement lorsque le nombre d'éléments devient important.

#### 4.3.4 Estimation des besoins futurs

Les tests suivants permettent de chiffrer et surtout d'estimer la validité de la méthode proposée sur des ensembles de l'ordre de plusieurs millions de cellules. Bien que les gros ensembles de données du CEA ne sont pas encore disponibles, il est important d'estimer les ressources nécessaires à leur traitement. Plusieurs ensembles de données, d'origines différentes, sont utilisés pour calculer ces estimations. Tous ont été obtenus à partir d'ensemble de points, puis tétraédrisés à l'aide d'une application conçue spécialement lors du stage, utilisant VTK. A noter que l'ensemble de données Bluntfin [HB84] provient d'une simulation effectuée par la NAS<sup>3</sup>.

Nom du fichier	Taille sur le disque	Nombre de points	Nombre d'arêtes	Nombre de tétraèdres
GrosDragon.vtk	5.5 Mo	25 001	198 085	172 760
BluntFin.vtk	8.6 Mo	40 960	227 862	274 389
DragonEtTore.vtk	18 Mo	57 001	629 865	556 746
Spirale.vtk	26 Mo	1500	1 124 250	1 121 253

TAB. 4.1 – Caractéristiques des grilles servant aux tests

Pour ces ensembles, je calcule l'espace mémoire nécessaire à mon application pour construire toute sa structure de données (incluant les données numériques, le maillage et le tas d'arêtes avec priorité). Je présente l'occupation mémoire moyenne par cellule car cela nous permet d'estimer les besoins pour les gros ensembles à traiter dans la suite du projet (nous avons un ordre de grandeur sur le nombre de cellules des données du CEA). La dernière colonne, présente le nombre moyen de tétraèdres incidents à chaque sommet. Ce chiffre est de l'ordre de 30 pour un ensemble acquis à partir d'une simulation d'un objet réel. Cependant, pour tester les cas limites de l'approche présentée, l'ensemble de données *Spirale* possède très peu de points mais une très grande quantité de cellules. Ces résultats sont présentés tableau 4.2.

<sup>3</sup>NASA Advanced Supercomputing (<http://www.nas.nasa.gov/index.html>)

Nom du fichier	Taille en octet par tétraèdre	Nombre moyen de cellules adjacentes à chaque sommet
<b>GrosDragon.vtk</b>	225 o	27
<b>BluntFin.vtk</b>	219 o	22
<b>DragonEtTore.vtk</b>	192 o	39
<b>Spirale.vtk</b>	164 o	2990

TAB. 4.2 – Occupation mémoire en octets par tétraèdre.

Il est donc intéressant de voir que pour des grilles de l'ordre de dix millions de cellules il faut s'attendre à posséder  $2000 \times 10\,000\,000 = 2\,000\,000\,000$  d'espace mémoire. Bien sûr ces valeurs ont été calculées avec la version actuelle du programme de simplification de maillage développé durant le stage de DEA. Cette version n'a pas été optimisée. En effet, le but de mon projet de DEA n'était pas de fournir un outil optimisé mais essentiellement de poser les problèmes et les estimations au traitement de données de masses.

## 4.4 Complexité

L'étude de la complexité est fondamentale dans le développement d'algorithmes de traitement de gros maillages. En effet, la conception et les premiers tests s'effectuant souvent sur des ensembles de plus petites tailles, il arrive parfois que les contraintes de temps de traitement n'apparaissent pas. A l'aide de la structure de donnée choisie, la méthode de simplification garantit une complexité de l'ordre de  $o(p \log(n))$  avec  $p$  étant le nombre d'arêtes à supprimer et  $n$  le nombre total d'arêtes présentes dans l'ensemble.

Le tas d'arêtes avec priorité est implémenté à l'aide d'un arbre binaire complet, si bien que le tri et la suppression sont des opérations en  $o(\log(n))$ . Or pour la contraction d'une arête, il arrive souvent que cette opération nécessite  $p$  suppression d'arêtes adjacentes, avec chaque suppression ayant une complexité en  $o(\log(n))$ . La complexité totale de la méthode est donc  $p$  fois la complexité de suppression, d'où  $o(p \log(n))$ .

# Chapitre 5

## Résultats

### 5.1 Présentation

Le premier objectif de ce projet de DEA était de recenser et de classer les méthodes de simplifications applicables aux ensembles volumiques. A la suite de cette étude, il a été important de sélectionner et d'adapter certaines d'entre elles aux besoins particuliers du CEA. Dans le cas des grosses données volumiques utilisées avec précision par les ingénieurs du CEA, les méthodes disponibles étaient limitées notamment dans l'utilisation pertinente des valeurs numériques présentes à l'intérieur de l'ensemble. L'utilisation de notions de géométrie algorithmique (utilisation de la notion de lien introduite par Tamal Dey) apparaissait aussi comme une approche robuste au traitement de données volumiques. L'outil implémentant notre approche intègre à ce jour des tests topologiques et différentes mesures d'erreurs associés à la contraction d'une arête.

Les résultats présentés montrent plusieurs séries de simplifications de maillage. Les tests portent sur l'ensemble volumique *Blunt-Fin* qui est particulièrement intéressant pour sa précision dans le champ scalaire. Dans toutes les séries de tests réalisés, l'approche favorise la préservation de la topologie. En d'autres termes, une contraction la modifiant n'est pas contractée. Au contraire, dans le cas de critères préservant la surface du volume, présenté figure 5.2 les arêtes modifiant la silhouette sont pénalisées mais restent contractables par la suite.

### 5.2 Interprétation

Les figures 5.1 et 5.2 présentent successivement les étapes de la simplification. Sur chaque ligne est présentée le même niveau de résolution. Le pourcentage donné est la proportion de sommets restante dans l'ensemble par rapport aux données initiales. Le tableau 5.1 détaille précisément le contenu de l'ensemble de données à chaque étape correspondant aux lignes des figures 5.1 et 5.2.

La première colonne de la 5.1 illustre la conservation du champ scalaire. Les arêtes contractées en premier sont celles dont la variation du champ scalaire est la plus faible entre ses sommets. Ainsi la première colonne montre un détail relativement bien respecté tout au long de la

	Début		1 <sup>ere</sup> étape		2 <sup>eme</sup> étape		3 <sup>eme</sup> étape		4 <sup>eme</sup> étape	
Elément	Nb	%	Nb	%	Nb	%	Nb	%	Nb	%
Sommets restants	40 960	100	31 930	78	18 052	44,1	9 323	22,8	3 238	7,91
Cellules restantes	227 862	100	179 391	78,7	101 030	44,3	50 942	22,4	26 107	11,5
Arêtes restantes	274 389	100	216 088	78,8	123 752	45,1	64 837	23,6	30 382	11,1
Arêtes du tas	274 389	100	215 118	78,4	116 660	42,5	46 474	16,9	0	0

TAB. 5.1 – Composition de l'ensemble de données *Blunt-Fin* au cours des étapes de simplifications

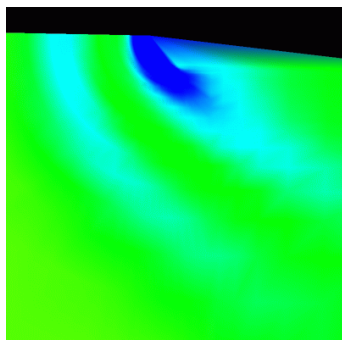
contraction. La position finale du sommet est simplement le milieu de l'arête contractée. Par contre dans cette approche la géométrie est très nettement dégradée dès la première étape. Ces résultats montrent bien la nécessité de conserver d'autres critères qui cependant doivent concerver cette pertinence sur le champ scalaire.

La figure 5.2 montre une autre série de tests faisant intervenir un critère surfacique. Dans le cas où l'arête à contracter est au bord du volume alors un coût élevé lui est affecté. Cependant, cette arête est toujours potentiellement contractable. En effet, ce critère important de conservation de la surface n'est pas utilisé pour refuter une contraction mais juste pour la retarder au maximum. Ainsi avec seulement 23% des données originales, l'aspect du champ scalaire à la surface reste très bon.

La deuxième et troisième colonne montrent la surface du volume. La différence est la position finale du sommet résultant de la contraction. Dans la deuxième colonne, sa position est toujours le milieu de l'arête tandis que dans la troisième, sa position est celle du sommet de l'arête de bord. Ainsi, la qualité géométrique de la surface est très bonne, jusqu'à un niveau de simplification élevé. Par contre lors de la dernière étape, il ne reste plus que des arêtes de bord dans le tas et donc la géométrie est dégradée. Dans certaines approches comme [THJ99] ces arêtes ne sont jamais contractées si bien que la frontière du volume reste excellente, mais le niveau de simplification est moins élevé.

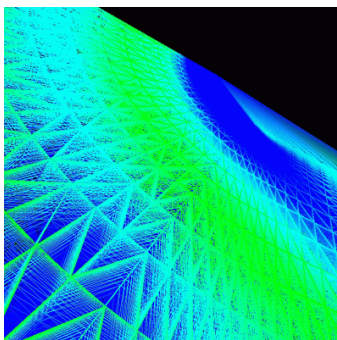


(a)



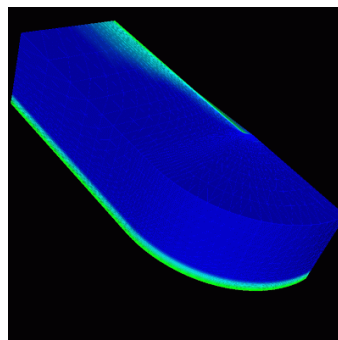
100%

(b)

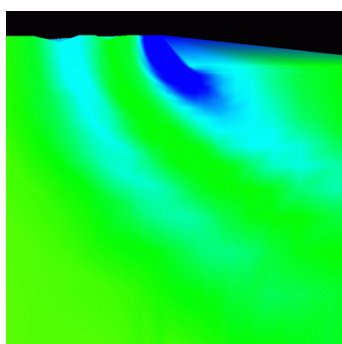


100%

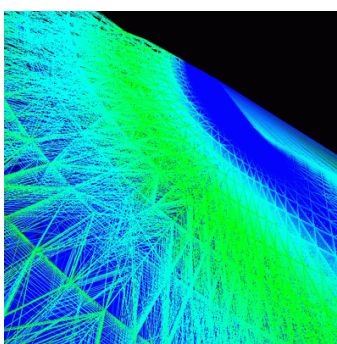
(c)



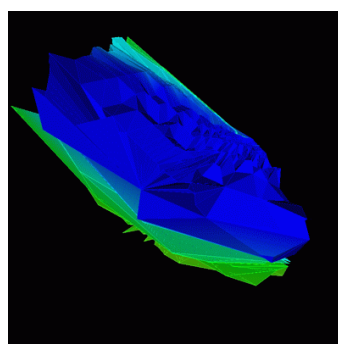
100%



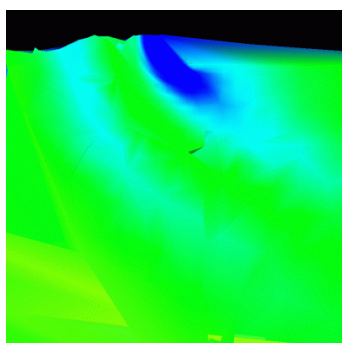
78%



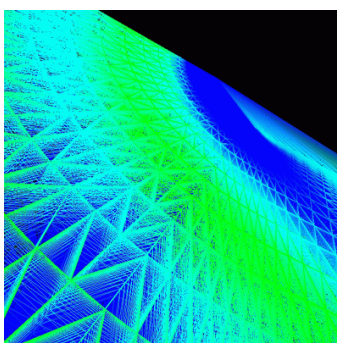
78%



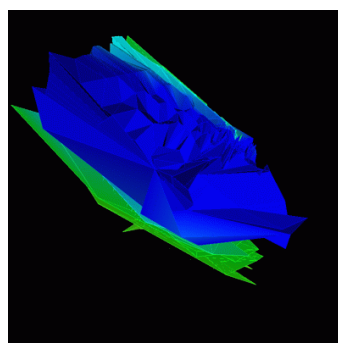
78%



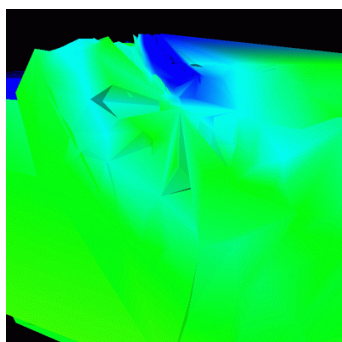
44%



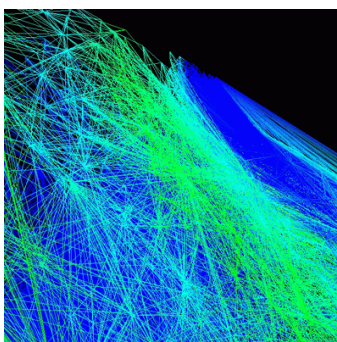
44%



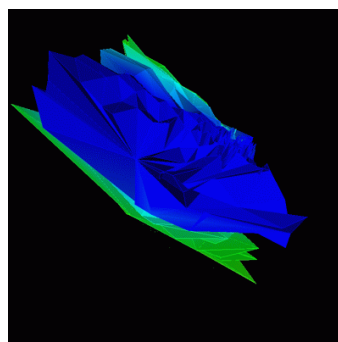
44%



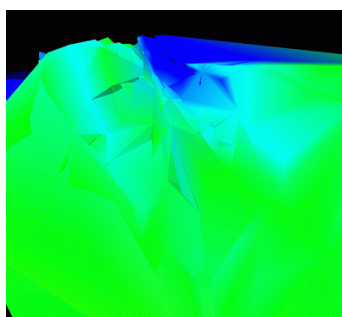
23%



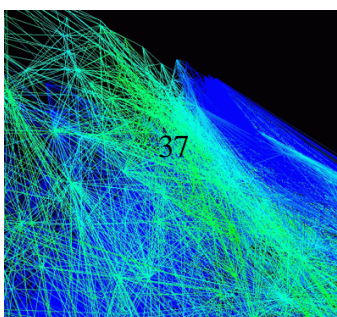
23%



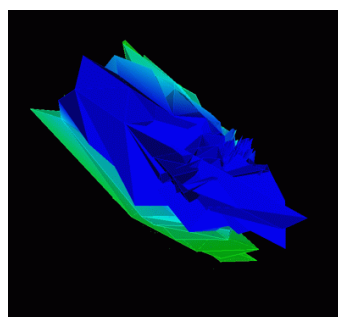
23%



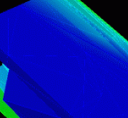
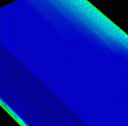
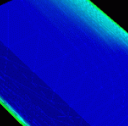

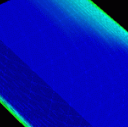
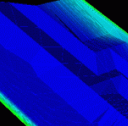
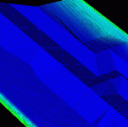
37%



37%



37%





# Chapitre 6

## Perspectives

### 6.1 Vers un nouveau calcul de l'erreur d'approximation

A travers l'approche proposée et l'étude bibliographique des chapitres 2 et 3, il apparaît clairement que les méthodes de simplification disposent d'un panel important de mesures d'erreurs et de mécanisme pour diminuer le nombre de cellules. Cependant, toutes ces approches, surfaciques ou volumiques, prennent en compte des critères d'erreur calculés sur des éléments d'une certaine dimension, invariante. En d'autres termes, il n'existe pas aujourd'hui, de méthodes de simplification précises intégrant les erreurs commises à la fois sur les sommets, les arêtes, les faces et les cellules.

Hugues Hoppe a présenté une méthode surfacique [Hop99] simplifiant les maillages en préservant les discontinuités des couleurs (attributs scalaires de surface). Cette approche est un cas particulier de l'idée proposée. En effet, cette méthode de simplification surfacique, fait intervenir des contraintes géométriques sur les faces du modèle (aspect 2D de la mesure d'erreur entre deux surfaces) mais aussi numériques, notamment avec la détection des arêtes (aspect 1D de la mesure d'erreur entre deux arêtes). La figure 6.1 montre cette simplification qui tient compte des grands changements de l'intensité lumineuse avec détection des arêtes.

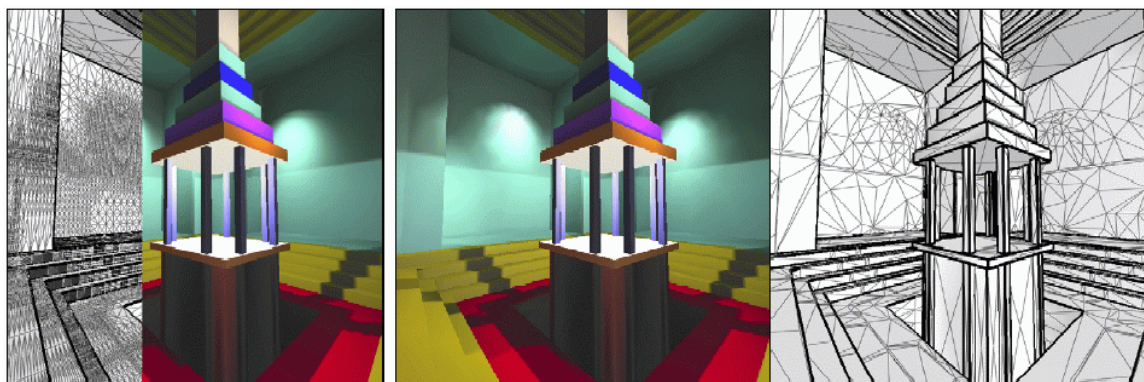


FIG. 6.1 – Simplification d'un maillage avec discontinuité des attributs de couleurs

## 6.2 Les besoins de la visualisation scientifique

Plus précisément, dans le cadre de la visualisation scientifique il est intéressant de pouvoir donner des contraintes fortes sur des détails de l'ensemble. En effet, la conservation de points critiques, d'arêtes de bord, de faces de contact entre plusieurs modèles ou encore de données numériques au sein de l'ensemble, motive cette direction de recherche. Un aspect important de cette nouvelle approche est le recensement de ces contraintes. Il est important dans un premier temps d'envisager toutes les dimensions possibles des modèles volumiques habituels. Autrement dit, il faut pouvoir interpréter et mettre en commun des erreurs commise sur les points (0D), les arêtes(1D), les faces(2D) et enfin les cellules(3D). Toutes ces contraintes peuvent porter, au choix, sur la géométrie ou les données numériques attachées aux maillages. Il apparaît cependant plus naturel de porter des critères de position géométrique sur des sommets, des arêtes ou des faces, tandis que les contraintes numériques s'appliqueraient plutôt sur des faces ou des cellules.

La figure 6.2, illustre ces choix de contraintes qui portent sur des éléments de dimensions différentes. La méthode d'intégration de toutes ces erreurs reste encore à définir pour la suite du projet.

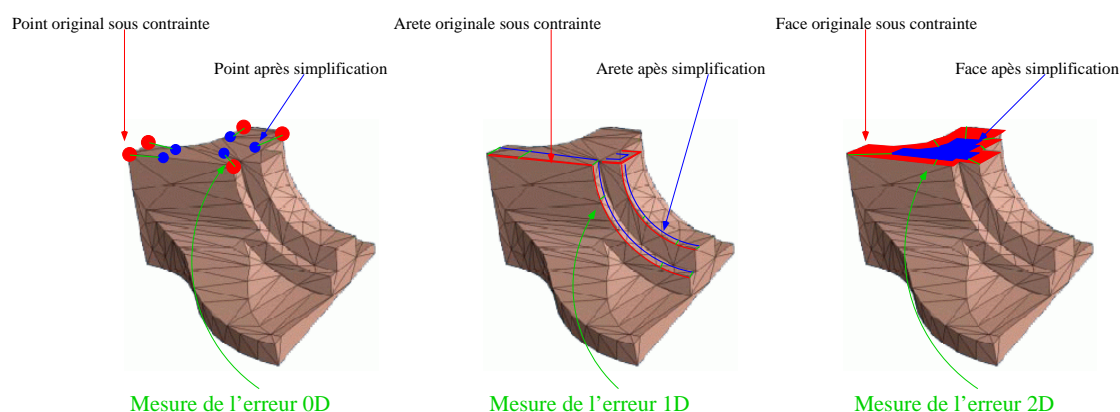


FIG. 6.2 – Les contraintes de dimensions 0D, 1D et 2D

Les données du CEA à traiter sont des ensembles volumiques, avec des points, des arêtes et des faces qu'il faut impérativement conserver pour la qualité géométrique du modèle. Par contre à l'intérieur, les données scalaires (la SER) doivent être approximées de la meilleure façon possible, en pénalisant, si besoin est, la géométrie interne des cellules.

# Chapitre 7

## Conclusion

Le traitement de gros ensembles de données volumiques est un domaine récent qui s'est particulièrement développé les cinq dernières années, en parallèle avec les moyens d'acquisition de données de masse. Les procédés de visualisation, indispensables à un traitement rapide et performant de ces gros ensembles de données, doivent pouvoir être exécutés de manière interactive sur un parc de stations graphiques de performances hétérogènes. L'approche consistant à simplifier les données originales, créant un modèle multirésolution de l'ensemble, s'est révélée être un outil indispensable à la réalisation de ces objectifs. Cependant la prise en compte de données provenant de simulations numériques stationnaires ou dynamiques de phénomènes complexes pose un ensemble de problèmes largement inexploré, comme la coexistence de maillages de nature diverse ou le couplage de données géométriques et numériques.

Le traitement de gros résultats intéresse particulièrement le CEA. C'est pourquoi un projet sur le thème de la modélisation multirésolution et de la visualisation volumique de larges ensembles de données a été créé. Mon stage de DEA s'intègre entièrement dans ce projet, permettant de poursuivre l'approche proposée au cours d'une thèse effectuée au sein du CEA.

Le présent projet de DEA a permis de recenser les problèmes que pose la simplification volumique à travers une étude bibliographique. Après un recensement des besoins du CEA, comparé aux manques des approches existantes, nous avons proposé les bases d'une méthode de simplification volumique. La conservation de la topologie, la pertinence des données numériques et de la préservation de la frontière du volume sont les aspects les plus importants. La particularité de cette approche est de rassembler dans un même outil, des notions de géométrie algorithmique et de visualisation. En effet, de nos jours, il est important de posséder des outils performants de simplification utilisant des notions souvent complexes et appartenant à d'autres domaines que l'informatique graphique. Son implémentation avec la librairie graphique VTK la rend portable et rapidement modifiable par la suite, notamment par des ingénieurs du CEA.

Cependant, tout traitement de données en visualisation scientifique doit être effectué avec un contrôle précis sur tous les éléments des entrées. L'étude aux différentes dimensions présentes, semble être une approche prometteuse. Un calcul de l'erreur associé à la simplification prenant en compte toutes ces dimensions a été introduit et sera notre axe de recherche pour les travaux futurs au sein du CEA.



# Annexe A

## Distance de Hausdorff

La distance Euclidienne entre un point  $x$  et un ensemble  $Y \subset \mathbb{R}^n$  est défini par

$$d(x, Y) = \inf_{y \in Y} d(x, y) \quad (\text{A.1})$$

ou  $d(x, y)$  est la distance Euclidienne entre deux points de  $\mathbb{R}^n$ . En utilisant cette définition, on peut définir la distance  $d_E(X, Y)$  d'un ensemble  $X$  à un autre ensemble  $Y$  par

$$d_E(X, Y) = \sup_{x \in X} d(x, Y). \quad (\text{A.2})$$

Cette distance est appelée la distance de Hausdorff dans une direction entre l'ensemble  $X$  et l'ensemble  $Y$ . Cependant cette mesure de distance n'est pas symétrique, c'est-à-dire qu'en général  $d_E(X, Y) \neq d_E(Y, X)$ . Afin de se ramener à une mesure symétrique, une nouvelle distance est calculée de la forme

$$d_H(X, Y) = \max(d_E(X, Y), d_E(Y, X)). \quad (\text{A.3})$$

Si la distance de Hausdorff entre la triangulation initiale  $T$  et la triangulation simplifiée  $S$  est inférieure à une borne prédéfinie  $\varepsilon$  alors

$$\forall x \in T \text{ il existe un } y \in S \text{ tel que } d_E(x, y) < \varepsilon \quad (\text{A.4})$$

et

$$\forall y \in S \text{ il existe un } x \in T \text{ tel que } d_E(x, y) < \varepsilon \quad (\text{A.5})$$

La particularité de la méthode est de fournir un contrôle sur la position du maillage simplifié par rapport au maillage initial et de conserver les arêtes vives originales. De plus elle peut être utilisée pour la construction de représentation multirésolution. Dans le cadre de simplification de maillage volumique, elle apporte un outil supplémentaire pour contrôler l'erreur d'approximation à la frontière du volume. La triangulation envisagée peut, dans le cas volumique, être l'enveloppe convexe du domaine à étudier.

## Annexe B

# Génération d'une isosurface en utilisant TclTk et VTK

```
# Charger les outils fournis par VTK
source ~/.wishrc
source vtkInt.tcl
source colors.tcl

# Initialiser la fenêtre graphique et la track-ball de VTK
vtkRenderer      renderer
vtkRenderWindow  renWin
vtkRenderWindowInteractor iren
renWin AddRenderer renderer
iren SetRenderWindow renWin

# Acquérir les données depuis un fichier
vtkUnstructuredGridReader reader
reader SetFileName "../currentVersion/UnstructuredGridOutput.vtk"
reader Update

# Initialiser les variables (valeurs extrémales de l'ensemble)
set dims [[[[reader GetOutput] GetPointData ] GetScalars ] GetRange ]
set min  [lindex $dims 0]
set max  [lindex $dims 1]
set level $min

# Construire l'isosurface
vtkContourFilter iso
iso SetInput [reader GetOutput]
iso SetValue 0 $level

# Mapper l'isosurface
vtkPolyDataMapper isoMapper
isoMapper SetInput [iso GetOutput]
isoMapper ScalarVisibilityOff
isoMapper SetScalarRange $min $max

# Configurer l'apparence de l'isosurface
vtkActor isoActor
isoActor SetMapper isoMapper
eval [isoActor GetProperty] SetColor $green

# Enregistrer les éléments à afficher
renderer AddActor isoActor

# Initialiser l'affichage
iren Initialize

# Interface graphique
```



```

frame .f
label .f.h -text "Isosurface"
scale .f.ill -label "Iso level"
    -from $min -to $max -resolution 0.01
    -length 5c -orient horizontal
.f.ill set $level
button .f.s -text "Quit"
pack .f.h .f.ill .f.s
pack .f

# Générer dynamiquement l'isosurface
bind .f.ill <ButtonRelease> {
    set level [.f.ill get]
    iso SetValue 0 $level
    renWin Render
}

```

# Bibliographie

- [BS96] Chandrajit Bajaj and Daniel Schikore. Error-bounded reduction of triangle meshes with multivariate data. *SPIE*, 2656 :34–45, 1996.
- [CCMS97] A. Ciampalini, Paolo Cignoni, Claudio Montani, and Roberto Scopigno. Multiresolution decimation based on global error. *The Visual Computer*, 13(5) :228–246, 1997.
- [CMS98] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. *Computers and Graphics*, 22(1) :37–54, 1998.
- [CR00] P.Cignoni D.Costanza C.Montani C.Rocchini and R.Scopigno. Simplification of tetrahedral volume with accurate error evaluation. In *Proceedings IEEE Visualization 2000*, pages 85–92, 2000.
- [CVM<sup>+</sup>96] Jonathan Cohen, Amitabh Varshney, Dinesh Manocha, Greg Turk, Hans Weber, Pankaj Agarwal, Frederick Brooks, and William Wright. Simplification envelopes. *Computer Graphics*, 30(Annual Conference Series) :119–128, 1996.
- [DEG99] T. Dey, H. Edelsbrunner, and S. Guha. Computational topology, 1999.
- [DEGN98] T. Dey, H. Edelsbrunner, S. Guha, and D. Nekhayev. Topology preserving edge contraction, 1998.
- [Gar99] M. Garland. Multiresolution modeling : Survey and future opportunities, 1999.
- [GH97] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. *Computer Graphics*, 31(Annual Conference Series) :209–216, 1997.
- [GSG96] Markus H. Gross, Oliver G. Staadt, and Roger Gatti. Efficient triangular surface approximations using wavelets and quadtree data structures. *IEEE Transactions on Visualization and Computer Graphics*, 2(2) :130–143, 1996.
- [Gué95] André Guézic. Surface simplification with variable tolerance. Technical Report MRCAS '95, Second Annual Intl. Symp. on Medical Robotics and Computer Assisted Surgery, 1995.
- [Gué96] André Guézic. Surface simplification inside a tolerance volume. Technical Report RC 20440, IBM Research, 1996.
- [Ham94] Bernd Hamann. A data reduction scheme for triangulated surfaces. *Computer-Aided Geometric Design*, 11 :197–214, 1994.
- [HB84] C.M. Hung and P.G. Buning. Simulation of blunt-fin induced shock wave and turbulent boundary layer separation. AIAA Paper 84-0457, January 1984.
- [HG97] Paul S. Heckbert and Michael Garland. Survey of polygonal surface simplification algorithms. Technical report, Carnegie Mellon University, 1997.

- [Hop96] Hugues Hoppe. Progressive meshes. *Computer Graphics*, 30(Annual Conference Series) :99–108, 1996.
- [Hop99] Hugues H. Hoppe. New quadric metric for simplifying meshes with appearance attributes. In David Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 59–66, San Francisco, 1999.
- [HZ91] M.J. De Haemer and M.J. Zyda. Simplification of objects rendered by polygonal approximations. *Computer Graphics*, 15(2) :175–184, 1991.
- [KE00] Martin Kraus and Thomas Ertl. Simplification of nonconvex tetrahedral meshes, 2000.
- [KLS96] Reinhard Klein, Gunther Liebich, and Wolfgang Straßer. Mesh reduction with error control. In Roni Yagel and Gregory M. Nielson., editors, *IEEE Visualization '96*, pages 311–318, 1996.
- [KT96] A.D. Kalvin and R. Taylor. Surfaces : Polygonal mesh simplification with bounded error. *IEEE C.G. and A*, 16(3) :64–77, 1996.
- [Lab98] Lawrence Livermore National Laboratory. Terascale visualization, 1998. <http://www.llnl.gov/terascale-vis/index.html>.
- [LT98] Peter Lindstrom and Greg Turk. Fast and memory efficient polygonal simplification. In *Proceedings IEEE Visualization '98*, pages 279–286, 1998.
- [Lue01] David P. Luebke. A developer’s survey of polygonal simplification algorithms. *IEEE Computer Graphics and Applications*, 21(3) :24–35, /2001.
- [NFT00] Nakamura, Fujishiro, and Takeshima. Towards optimizing metric to simplify colored interval volumes. In *Work in Progress, IEEE Visualization*, pages W2–3, Salt Lake City, 2000.
- [oM86] National Library of Medicine. The visible human project, 1986. <http://www.nlm.nih.gov/research/visible/>.
- [PH97] Jovan Popovic and Hugues Hoppe. Progressive simplicial complexes. *Computer Graphics*, 31(Annual Conference Series) :217–224, 1997.
- [RB93] J. Rossignac and P. Borrel. Multi-resolution 3d approximation for rendering complex scenes. *Geometric Modeling in Computer Graphics*, pages 455–465, 1993.
- [RO96] K.J. Renze and J.H. Oliver. Generalized unstructured decimation. *IEEE C.G. and A*, 16(6) :24–32, 1996.
- [RR96] Rémi Ronfard and Jarek Rossignac. Full-range approximation of triangulated polyhedra. *Computer Graphics Forum*, 15(3), August 1996. Proc. Eurographics '96.
- [SG98] Oliver G. Staadt and Markus H. Gross. Progressive tetrahedralizations. In *Proceedings IEEE Visualization '98*, pages 397–402, 1998.
- [SML98] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit, An Object-Oriented Approach To 3D Graphics*. Prentice Hall, 1998. ISBN 0-13-954694-4.
- [SZL92] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen. Decimation of triangle meshes. *Computer Graphics*, 26(2) :65–70, 1992.

- [THJ99] Isaac J. Trotts, Bernd Hamann, and Ken I. Joy. Simplification of tetrahedral meshes with error bounds. In *Proceedings IEEE Transactions on visualization and computer graphics*, volume 5, pages 224–237, july-september 1999.
- [THW98] Iscaac J. Trotts, Bernd Hamann, and David Wiley. Simplification of tetrahedral meshes. In H.Hagen D.Erbert and H.Rushmeier, editors, *Proceedings Visualization*, pages 287–296, October 1998.
- [Tur92] Greg Turk. Re-tiling polygonal surfaces. *Computer Graphics*, 26(2) :55–64, 1992.