
Ensembles simpliciaux hiérarchisés et intersection de surfaces

Matthieu Dazy — Bruno Lévy — Jean-Laurent Mallet

ISA-Gocad (INRIA Lorraine et CNRS)

rue du Doyen Marcel Roubault

B.P. 40

54501 VANDŒUVRE

(dazy|levy|mallet)@ensg.inpl-nancy.fr

RÉSUMÉ. Nous nous intéressons au problème du calcul de l'intersection de surfaces triangulées. Une approche classique, reposant principalement sur la géométrie, est sujette à plusieurs problèmes. En particulier, il serait souhaitable que la géométrie du résultat obtenu soit cohérente avec la topologie du problème de départ. Nous proposons une nouvelle structure de données, à base d'ensembles simpliciaux, pour mettre en place une approche favorisant l'utilisation d'informations combinatoires.

ABSTRACT. We consider the problem of computing the intersection of triangulated surfaces. A geometry-based method can present several problems. In particular, we wish the geometrical properties of the result to be consistent with the combinatorial properties of the original problem. We propose a new data structure based on simplicial sets, and a new approach using combinatorial information.

MOTS-CLÉS : Ensembles Simpliciaux, Modélisation Géométrique, Intersection de Surfaces, Géométrie Algorithmique.

KEYWORDS: Simplicial Sets, Geometric Modeling, Surface Intersection, Computational Geometry

1. Introduction

En géométrie algorithmique (voir par exemple [BOI 95]), il est possible de distinguer deux catégories de problèmes : les problèmes *non constructifs*, qui s'intéressent aux connexions entre les éléments existants d'un modèle, et les problèmes *constructifs*, dans lesquels de nouveaux éléments sont créés et ajoutés au modèle. Les premiers peuvent n'utiliser que des prédicats, alors que les seconds doivent nécessairement faire appel à la géométrie pour calculer de nouveaux éléments et modifier la structure. Par exemple, la construction d'un diagramme de Voronoï est un algorithme non constructif, alors que le calcul de l'intersection de deux surfaces est un algorithme constructif.

Dans ce dernier cas, un problème important est que la création de nouveaux points soit effectuée de manière cohérente avec la topologie du problème, l'utilisation de points erronés pouvant provoquer des erreurs dans le déroulement de l'algorithme. Pourtant, il serait possible de résoudre une partie des problèmes en tirant parti des informations combinatoires disponibles. D'autre part, beaucoup d'algorithmes modifient dynamiquement les structures de données qu'ils manipulent alors qu'il pourrait être intéressant de conserver une trace des opérations effectuées.

Nous considérons le problème du calcul de l'intersection d'un ensemble de surfaces. Il existe déjà des solutions pour certaines classes de surfaces, comme par exemple les NURBS, qui sont beaucoup utilisées en modélisation industrielle (voir [ELM 94],[KRI 97]). Les surfaces triangulées présentent plusieurs avantages : d'une part, leurs éléments de base possèdent des propriétés mathématiques qui rendent leur utilisation pratique ; d'autre part, elles sont adaptées à la représentation d'objets naturels et à la mise en place d'algorithmes par éléments finis (lissage, paramétrisation). En géologie, par exemple, l'intersection d'un ensemble de surfaces peut être utilisée pour partitionner l'espace en régions délimitées par ces surfaces. Par la suite, des méthodes par éléments finis (interpolation de propriétés physiques, simulation de propagation d'ondes sismiques...) peuvent être appliquées sur de tels volumes. Pour un exemple d'algorithme d'intersection de surfaces triangulées, on peut se reporter à [AFT 97].

Nous présentons la structure HESC (Hierarchical Embedded Simplicial Complex). Cette structure de données hiérarchique fondée sur les ensembles simpliciaux permet de stocker les informations calculées au cours de l'exécution d'un algorithme d'intersection de surfaces. Après une brève présentation des ensembles simpliciaux, nous exposons nos extensions pour la structure HESC, puis les opérations définies sur ces différents éléments. Enfin, nous étudions l'application à l'intersection de surfaces.

2. Ensembles simpliciaux abstraits

Il existe plusieurs modèles de données permettant de représenter la structure combinatoire de surfaces triangulées :

- modèle frontière (*B-rep*[WEI 86]) : dans cette approche, les éléments sont

représentés par leur bord,

- les G-cartes ([LIE 91],[BRI 89]) : cette structure permet de représenter une classe d’objets plus large, les complexes cellulaires,
- les ensembles simpliciaux ([MAY 67],[AGO 76]) : cette structure combinatoire est basée sur la manipulation d’ensembles abstraits.

Les G-cartes et les ensembles simpliciaux ont l’avantage de reposer sur des théories mathématiques éprouvées, ce qui permet de valider plus facilement les algorithmes développés. Nous avons choisis d’utiliser les ensembles simpliciaux en raison de leur adéquation au problème étudié : en effet, les points, arêtes et triangles de \mathbb{R}^3 sont des simplexes de l’espace euclidien. Le lecteur peut consulter [MAY 67] pour une présentation détaillée des ensembles simpliciaux ainsi que [LAN 95] et [LAN 96] pour des exemples de leur utilisation en modélisation géométrique.

2.1. Simplexes et topologie

Dans l’espace \mathbb{R}^N , un simplexe de dimension i est défini (i -simplexe) comme l’enveloppe convexe de $(i + 1)$ points linéairement indépendants. Nous nous intéressons ici uniquement aux relations combinatoires entre les simplexes.

De manière intuitive, nous pouvons décrire un i -simplexe comme un élément abstrait de dimension i connecté à $(i + 1)$ éléments abstraits de dimension $(i - 1)$ par des applications. Sur les figures, les 0-, 1- et 2-simplexes seront représentés respectivement par des sommets, des arêtes et des triangles, mais il faut garder à l’esprit qu’il s’agit d’éléments sans existence géométrique.

Formellement, un *ensemble simplicial* de dimension n est défini par

$$\mathcal{K} = (V, E, \{S_0, S_1, \dots, S_n\}, \{d_0, d_1, \dots, d_n\})$$

- V est un ensemble fini d’éléments abstraits. $\mathcal{P}(V)$ est l’ensemble de ses parties,
- E est un sous-ensemble de $\mathcal{P}(V) \setminus \{\emptyset\}$, tel que $\forall \sigma \in E, \forall \tau \in \mathcal{P}(V) \setminus \{\emptyset\}, \tau \subset \sigma \Rightarrow \tau \in E$. σ est appelé un *simplexe* de \mathcal{K} , et sa dimension est notée $\dim \sigma = \text{Card}(\sigma)$,
- $\forall 0 \leq i \leq n, S_i = \{\sigma \in E, \dim \sigma = i\}$ est l’ensemble des simplexes de dimension i . La dimension de \mathcal{K} est donc égale à $n = \max_{\sigma \in E} \dim \sigma$,
- $\forall i \in [0, n], \forall j \in [i, n], j \geq 1, d_i$ est une application de S_j dans S_{j-1} qui, à un j -simplexe σ , associe un $(j - 1)$ -simplexe $\tau \subset \sigma$ (voir figure 1). d_i est appelé un *opérateur bord* de σ et on note $\tau = \sigma d_i$. De plus, $\forall \sigma \in S_k, \forall i, j \in [0, k], i \neq j \Rightarrow \sigma d_i \neq \sigma d_j$.

2.2. Définitions et propriétés des simplexes

Plusieurs ensembles et propriétés peuvent être définis sur les simplexes, notam-

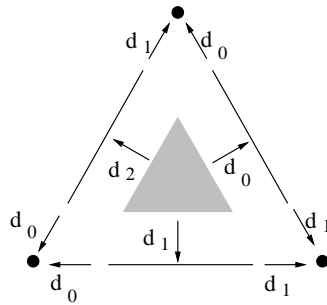


Figure 1. un 2-simplexe et ses faces sont connectés par les opérateurs bord, définis de manière unique pour chaque simplexe.

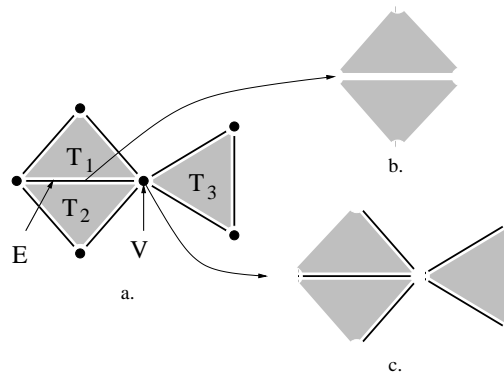


Figure 2. a : le 2-simplexe T_1 est incident aux simplexes E et V . T_1 et T_2 sont adjacents via E , et T_2 et T_3 sont adjacents via V . b : l'étoile du 1-simplexe E . c : l'étoile du 0-simplexe V .

ment :

– Un simplexe σ est dit *incident* à un simplexe τ si il existe une séquence d'opérateurs bord telle que $\sigma d_{i_k} \dots d_{i_1} = \tau$ (voir figure 2a). On note $\sigma \mathcal{I} \tau$ et τ est alors appelé une *face* de σ ,

– deux simplexes σ_1 et σ_2 sont dits *adjacents* si ils ont une face commune (voir figure 2a). On note $\sigma_1 \mathcal{A} \sigma_2$,

– le *bord* d'un simplexe est défini comme étant l'ensemble de ses faces : $\partial \sigma = \{\tau / \sigma \mathcal{I} \tau\}$,

– l'*étoile* d'un simplexe est définie comme étant l'ensemble des simplexes qui lui sont incidents : $St \sigma = \{\tau / \tau \mathcal{I} \sigma\}$ (voir figures 2b et 2c).

2.3. Opérateurs sommet

Les 0-simplexes jouent un rôle important vis-à-vis simplexes. En effet, nous verrons par la suite qu'il est possible de calculer la géométrie d'un simplexe uniquement à partir de la géométrie de ses sommets. Nous avons donc besoin d'une méthode nous permettant d'accéder facilement aux 0-simplexes d'un simplexe donné.

Il est facile de constater qu'appliquer une séquence de k opérateurs bord, permet de passer d'un k -simplexe à l'un des 0-simplexes de son bord. Notons qu'il existe $(k+1)!$ séquences distinctes de k opérateurs bord, alors que chaque k -simplexe n'est connecté qu'à $(k+1)$ 0-simplexes. Il faut donc choisir $(k+1)$ séquences nous permettant d'accéder aux différents 0-simplexes.

En imposant la condition suivante sur les opérateurs bord :

$$\forall 1 \leq i \leq n, \forall j < i, d_i d_j = d_j d_{i-1} \quad (1)$$

il est possible de prouver que pour toute séquence de k indices $(i_k, i_{k-1}, \dots, i_1)$, il existe $j \in [0, k]$ tel que $d_{i_k} d_{i_{k-1}} \dots d_{i_1} = d_k d_{k-1} \dots d_{j+1} d_{j-1} \dots d_1 d_0$. On note v_j^k cette dernière séquence. Les v_j^k sont appelés des *opérateurs sommet* puisqu'ils permettent de passer d'un k -simplexe au j -ème 0-simplexe de son bord.

Bien entendu, il est indispensable de garantir le respect de cette condition au cours de l'exécution des algorithmes. Lang donne dans [LAN 96] un algorithme permettant de garantir cet invariant lors de la construction des simplexes, en ordonnant les 0-simplexes en fonction d'une numérotation globale.

3. La structure HESC

Nous présentons maintenant la structure HESC (Hierarchically Embedded Simplicial Complex). A l'aide d'une nouvelle relation combinatoire, elle permet de stocker les informations topologiques et géométriques calculées par un algorithme. Il est à noter que la finalité du HESC est d'être utilisé en tant que structure de données temporaire pour l'exécution d'un algorithme et non comme outil de modélisation.

Un HESC de dimension n est défini par :

$$\mathcal{H} = \{ (\mathcal{K}_i)_{0 \leq i \leq n}, \sqsubset, \lambda, g, \Theta \}$$

- les \mathcal{K}_i sont une famille d'ensembles simpliciaux de dimension i ,
- la relation \sqsubset définit la notion de *plongement topologique*,
- l'opérateur λ définit la géométrie relative des 0-simplexes,
- l'opérateur g fournit une représentation géométrique effective de la structure,
- Θ est un simplexe particulier, utilisé en combinaison avec g .

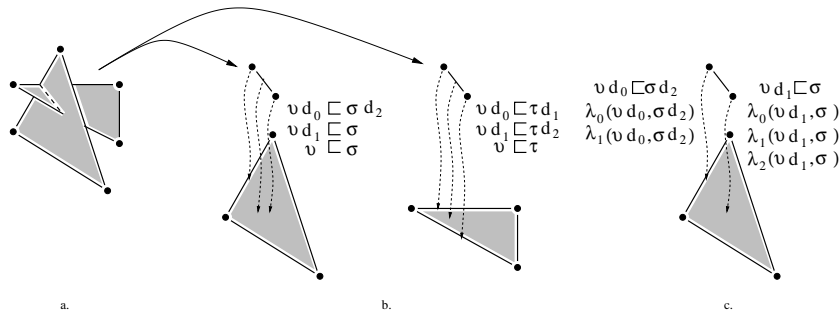


Figure 3. un exemple d'inclusion topologique. *a* : l'intersection des deux triangles génère un segment. *b* : les plongements topologiques. Les segments sont plongés dans chacun des triangles. Les sommets du segment sont plongés respectivement dans une arête et un triangle et dans deux arêtes. *c* : les coordonnées de plongement pour un des cas.

3.1. Extension combinatoire des ensembles simpliciaux abstraits

La combinatoire des ensembles simpliciaux abstraits nous permet essentiellement de calculer des relations de voisinage entre les différents simplexes de l'ensemble. Nous ajoutons une nouvelle relation pour décrire la manière dont les simplexes peuvent s'intersecter. Pour une étude détaillée des problèmes de combinatoire, le lecteur pourra consulter [LÉV 99].

Un k -simplexe τ peut être *plongé topologiquement* dans un simplexe σ de dimension supérieure ou égale à k . Cette relation est notée $\tau \sqsubset \sigma$ et découle directement de l'intersection de deux simplexes géométriques (voir figures 3a et 3b).

A l'aide de cette nouvelle relation, nous mettons en place la hiérarchie des ensembles simpliciaux \mathcal{K}_i de \mathcal{H} . L'ensemble \mathcal{K}_n représente l'objet sur lequel l'algorithme sera exécuté, les ensembles $\mathcal{K}_0 \dots \mathcal{K}_{n-1}$ étant destinés à contenir les informations calculées pendant les différentes étapes.

Les relations de voisinage entre simplexes sont décrites par les opérateurs bord à l'intérieur d'un même \mathcal{K}_i tandis que les relations d'inclusion sont décrites par des plongements \sqsubset entre les simplexes des différents \mathcal{K}_i .

3.2. Réalisation géométrique des ensembles simpliciaux abstraits

Pour compléter notre structure, il nous reste à produire une représentation géométrique des objets manipulés. L'opérateur géométrique g associe à chaque 0-simplexe de \mathcal{H} un point de \mathbb{R}^3 : $\forall \sigma, \dim \sigma = 0, \sigma g = P(x, y, z)$. Nous pouvons alors définir

un opérateur général \bar{g} qui calcule la géométrie d'un simplexe en calculant la somme des barycentres de ses sommets :

$$\Lambda = \{(\lambda_0, \dots, \lambda_{dim \sigma}), \lambda_k \in \mathbb{R}^+, \sum_{k=0}^{k=dim \sigma} \lambda_k = 1\} \quad \sigma \bar{g} = \sum_{\Lambda} \sum_{i=0}^{i=dim \sigma} \lambda_i \sigma v_i^{dim \sigma} g.$$

En utilisant les plongements topologiques, il nous est de plus possible de situer les 0-simplexes relativement aux simplexes dans lesquels ils sont plongés. L'opérateur λ donne, pour chaque plongement combinatoire d'un 0-simplexe τ dans un simplexe σ , les coordonnées barycentriques de τg dans σg (voir figure 3c) :

$$\forall \tau, \sigma / \tau \sqsubset \sigma, \lambda(\tau, \sigma) = \{\lambda_i(\tau, \sigma)\}_{0 \leq i \leq dim \sigma}$$

avec $\forall 0 \leq i \leq dim \sigma, \lambda_i(\tau, \sigma) \geq 0$ et $\sum_{i=0}^{i=dim \sigma} \lambda_i(\tau, \sigma) \sigma v_i^{dim \sigma} g.$

Il est important de garder à l'esprit que de cette manière, il est possible d'avoir plusieurs définitions pour la géométrie d'un 0-simplexe. Il sera donc nécessaire d'utiliser un système de calcul garantissant que :

$$\forall \sigma, \tau, v/v \sqsubset \sigma, v \sqsubset \tau, \sum_{i=0}^{i=dim \sigma} \lambda(v, \sigma) \sigma v_i^{dim \sigma} g = \sum_{i=0}^{i=dim \tau} \lambda(v, \tau) \tau v_i^{dim \tau} g.$$

Pour pouvoir calculer récursivement la géométrie de chaque sommet, chaque 0-simplexe doit être plongé dans au moins un autre simplexe. A cet effet, nous définissons un *simplexe englobant* Θ , dans lequel tout 0-simplexe sans plongement combinatoire pourra être plongé. En pratique, seuls les 0-simplexes de \mathcal{K}_n sont plongés dans Θ . Les 0-simplexes calculés par l'algorithme bénéficient normalement de plongements dans les simplexes dont ils sont l'intersection.

Ceci nous permet, de plus, de séparer la partie combinatoire de notre structure de sa partie géométrique. L'opérateur λ définit une géométrie locale pour les plongements topologiques et la représentation géométrique finale de la structure peut être réalisée dans un espace au choix de l'utilisateur.

4. Opérations sur les HESC

4.1. Opérations sur les simplexes

4.1.1. Intersection topologique de deux simplexes

Pour une paire de simplexes (σ, τ) , nous calculons les intersections entre les éléments de $\sigma \cup \partial\sigma$ et ceux des éléments de $\tau \cup \partial\tau$. Le calcul est basé sur des prédicats

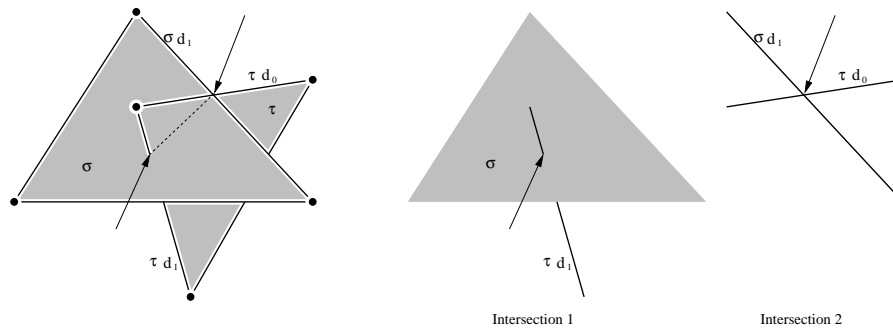


Figure 4. *intersection de deux 2-simplices. Notre calcul d'intersection doit nous renvoyer une intersection triangle/arête et une intersection arête/arête. L'arête d'intersection sera créée plus tard.*

topologiques exacts (voir [SHE 97]), et ne renvoie que les paires de simplexes dont l'intersection est réduite à un 0-simplexe (voir figure 4). La création des simplexes de dimension supérieure peut être déduite de ces informations dans une étape ultérieure.

4.1.2. Calcul du plongement des intersections

Pour une paire de simplexes (σ, τ) renvoyée par la méthode précédente, nous créons le 0-simplexe d'intersection v , nous calculons les coordonnées barycentriques $\lambda(v, \sigma)$ et $\lambda(v, \tau)$. Nous stockons ensuite les plongements $v \sqsubset \sigma$ et $v \sqsubset \tau$.

La précision des calculs est primordiale pour cette opération. En effet, les coordonnées des différents plongements d'un même simplexe doivent être cohérentes (voir 3.2). Il est envisageable de baser les calculs sur des bibliothèques de calcul précis telles que GMP¹, LEDA² ou Core³.

4.1.3. Maillage contraint d'un simplexe

Nous réalisons un recouvrement topologique de σ mettant en jeu les simplexes plongés dans σ ou dans un simplexe de son bord. Pour cette opération, le traitement effectif dépend de la dimension du simplexe. Il est supposé comme hypothèse de départ que les simplexes de $\partial\sigma$ ont déjà été maillés.

0-simplexe

Pour un 0-simplexe σ , les 0-simplexes plongés dans σ sont remplacés par σ lui-même. Cette opération peut se réaliser en modifiant les faces des simplexes de $St(\sigma)$.

1. <http://www.swox.com/gmp>
 2. <http://www.mpi-sb.mpg.de/LEDA/index.html>
 3. <http://cs.nyu.edu/exact/core>

1-simplexe

Pour un 1-simplexe σ , les sommets plongés dans σ sont triés par ordre croissant des λ_0 , puis les 1-simplexes reliant ces sommets sont créés, si nécessaire.

2-simplexe

Pour un 2-simplexe σ , nous construisons un maillage Delaunay des 0-simplexes plongés dans σ ou dans un élément de $\partial\sigma$, puis nous insérons dans ce maillage des contraintes exprimées sous la forme des 1-simplexes plongés dans σ ou dans un élément de $\partial\sigma$.

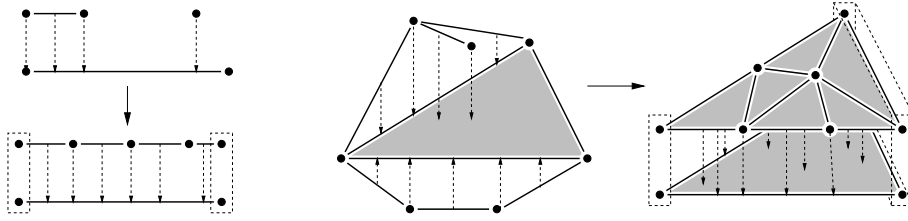


Figure 5. maillage contraint d'un 1-simplexe (à gauche) et d'un 2-simplexe (à droite). Les simplexes encadrés en pointillés sont identiques.

Dans tous les cas les anciens simplexes sont conservés afin de maintenir la cohérence du HESC pour les simplexes non maillés. Les anciens simplexes pourront être supprimés par la suite. Notons que l'utilisation des coordonnées barycentriques des plongements et des informations combinatoires nous permettent de travailler dans des repères locaux aux simplexes. Par exemple, pour calculer l'intersection de 1-simplexes plongés dans un même 2-simplexe, des calculs 2D sont suffisants, indépendamment de l'espace de la réalisation géométrique.

4.2. Opérations sur les ensembles simpliciaux

4.2.1. Conversion

Le HESC est une structure de données temporaire : son but est de fournir une représentation des différents états d'un algorithme au cours de son exécution. Dans cette optique, des convertisseurs doivent être utilisés pour transformer en HESC les structures de données utilisées par le logiciel de modélisation, ainsi que pour permettre de revenir à ces structures une fois les traitements effectués (voir figure 6).

D'une manière générale, un convertisseur se limite à construire les simplexes équivalents à la structure de données d'origine et à vérifier la bonne numérotation de ceux-ci. Cependant, si plusieurs objets doivent être regroupés dans un seul HESC, il pourra s'avérer nécessaire d'ajouter des informations sur l'origine des simplexes.

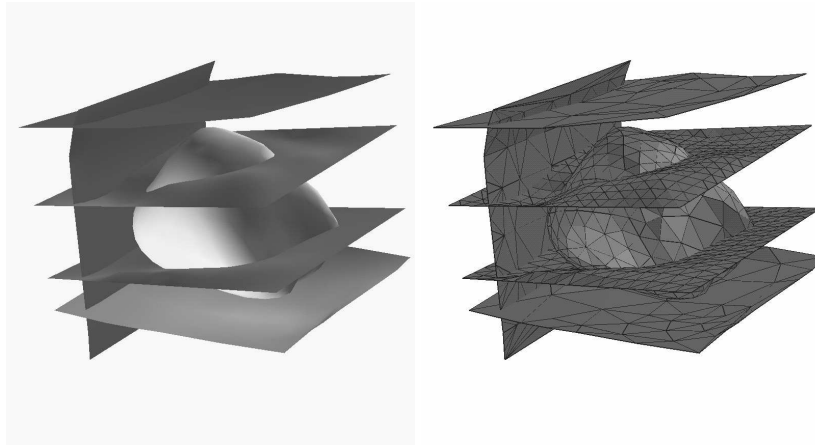


Figure 6. à gauche : le modèle 3D d'origine, composé de cinq surfaces. Ce genre de modèle est typique des cas rencontrés lors de la modélisation de couches géologiques. A droite : le même modèle converti en HESC. Les surfaces sont regroupées dans une seule structure.

4.2.2. Intersection d'un ensemble simplicial

Cette opération calcule l'auto-intersection de \mathcal{K}_i et stocke la structure résultante dans \mathcal{K}_{i-1} (voir figure 7). Elle peut se décomposer en plusieurs étapes :

1. Calcul topologique des 0-simplexes d'intersection : pour chaque paire de i -simplexes $\sigma, \tau \in \mathcal{K}_i$, on calcule les intersections entre σ et τ comme décrit en 4.1.1.
2. Calcul des plongements : les plongements topologiques et géométriques sont calculés pour chaque 0-simplexe créé, comme décrit en 4.1.2.
3. Calcul et plongement des k -simplexes d'intersection : pour chaque i -simplexe de \mathcal{K}_i mis en jeu, il est possible d'utiliser les plongements calculés précédemment pour déduire la création et les plongements éventuels de simplexes de dimension supérieure ou égale à 1.

4.2.3. Réalisation d'un ensemble simplicial

Cette opération construit le maillage des simplexes de l'ensemble simplicial \mathcal{K}_i (voir figure 8).

- Maillage des 0-simplexes,
- maillage des 1-simplexes,
- ...
- maillage des i -simplexes,
- suppression des simplexes maillés.

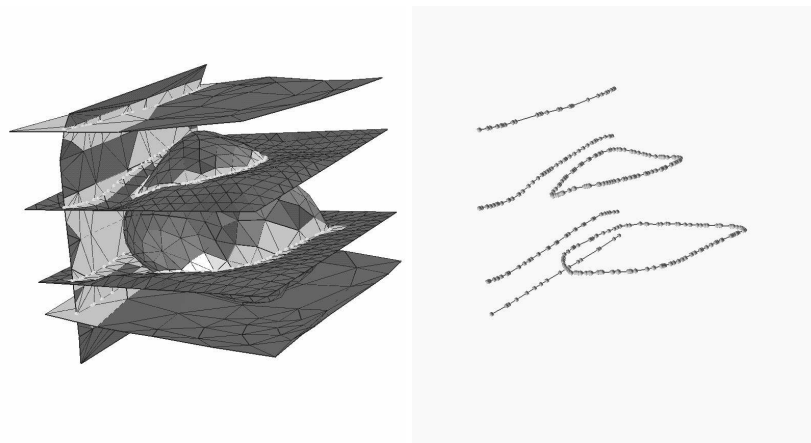


Figure 7. application de l'intersection sur le modèle présenté figure 6. Les simplexes mis en jeu par l'intersection sont affichés en gris clair. A droite, la ligne et les sommets d'intersection, stockés dans le premier niveau hiérarchique.

Mailler les simplexes par ordre de dimension croissante nous permet de garantir l'hypothèse faite en 4.1.3. Les nouveaux simplexes sont créés dans \mathcal{K}_i , afin de pouvoir détruire \mathcal{K}_{i-1} après l'opération. Le fait d'avoir créé des recouvrements de simplexes permet de détruire les simplexes maillés sans générer de discontinuité dans la structure.

4.3. Application : intersection de surfaces

La définition d'opérateurs de haut niveau sur les ensembles simpliciaux a plusieurs avantages. D'une part, cela nous permet de simplifier la conception d'algorithmes complexes. Ainsi, l'algorithme d'intersection de surfaces peut s'écrire :

1. intersection de \mathcal{K}_2
2. intersection de \mathcal{K}_1
3. réalisation de \mathcal{K}_1
4. réalisation de \mathcal{K}_2

D'autre part, ces opérateurs se révèlent relativement simples à implanter, ce qui améliore la lisibilité et la maintenabilité des programmes réalisés.

Sur l'exemple exposé sur les figures 7 et 8, le modèle comporte 1808 triangles et présente 682 cas effectifs d'intersection. La totalité du calcul nécessite 5 secondes sur une station Sun Ultra 10. La majorité du temps de calcul est utilisée par la phase de calcul des intersections, le reste étant dédié au remaillage des simplexes.

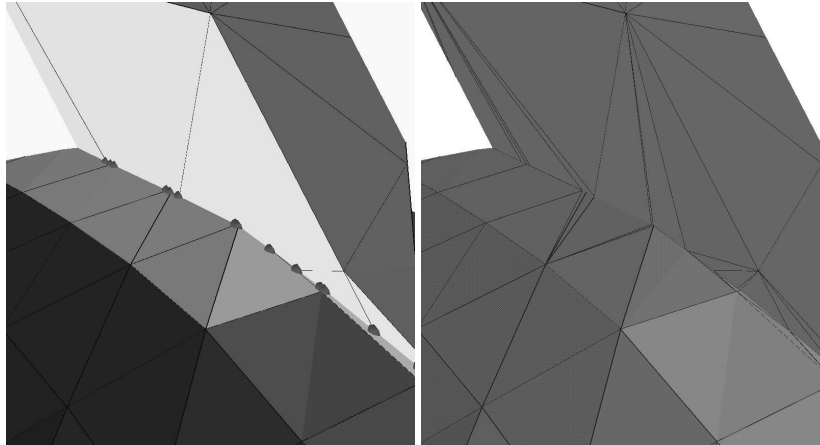


Figure 8. *un détail du modèle, avant et après maillage.*

5. Conclusion

A partir de la théorie des ensembles simpliciaux, nous apportons une nouvelle relation combinatoire permettant de représenter l'inclusion topologique de simplexes les uns dans les autres. Cette relation, utilisée en combinaison avec une hiérarchie d'ensembles simpliciaux, mène à la définition de la structure HESC.

La structure HESC permet de représenter des surfaces triangulées, et de stocker toutes les informations topologiques et géométriques calculées au cours d'un algorithme d'intersection de surfaces. La définition d'opérateurs de haut niveau sur les ensembles simpliciaux nous permet de simplifier la conception d'un tel algorithme ainsi que son implantation.

La poursuite du travail présenté dans cet article va s'orienter vers la mise en place d'une arithmétique précise pour les calculs géométriques, ainsi que vers la mise à profit des informations combinatoires pour améliorer la robustesse de l'algorithme et la cohérence du résultat.

Remerciements

Les auteurs tiennent à remercier les sponsors du consortium Gocad pour le soutien qu'ils apportent au projet.

6. Bibliographie

- [AFT 97] AFTOSMIS M., BERGER M., MELTON J., Robust and Efficient Cartesian Mesh Generation For Component-Based Geometry , *35th AIAA Aerospace Sciences Meeting*, 1997, version PDF disponible sur <http://george.arc.nasa.gov/aftosmis/publications/publications.html>.
- [AGO 76] AGOSTON M., *Algebraic Topology : A First Course - Pure and Applied Mathematics*, Marcel Dekker, 1976.
- [BOI 95] BOISSONAT J., YVINEC M., *Géométrie Algorithmique*, Ediscience international, 1995.
- [BRI 89] BRISSON E., Representing Geometric Structures in d dimensions; topology and order , *Proc. 5th ACM Symposium on Computational Geometry, Saarbrücken, Germany*, jun 1989, p. 187-196.
- [ELM 94] ELMARAGHY W., AL., Intersection of Volumes and Surfaces Areas of Cylinders for Geometrical Modelling and Tolerancing , *Computer Assisted Design*, vol. 26, numero 1, 1994, p. 29-45.
- [KRI 97] KRISHNAN S., MANOCHA D., An Efficient Surface Intersection Algorithm Based on Lower-Dimensional Formulation , *ACM Transactions on Graphics*, vol. 16, numero 1, 1997, p. 74-106.
- [LAN 95] LANG V., LIENHARDT P., Geometric modeling with simplicial sets , rapport numero95/06, 1995, Université Louis Pasteur, LSIIT - URA CNRS 356.
- [LAN 96] LANG V., Une étude de l'utilisation des ensembles simpliciaux en modélisation géométrique interactive , PhD thesis, Université Louis Pasteur - Strasbourg, 1996.
- [LIE 91] LIENHARDT P., Topological models for boundary representation : a comparison with n-dimensional generalized maps , *CAD*, vol. 23, numero 1, 1991, p. 59-82.
- [LÉV 99] LÉVY B., Topologie Algorithmique, Combinatoire et Plongement , PhD thesis, INPL, 1999.
- [MAY 67] MAY J., *Simplicial objects in algebraic topology*, Van Nostrand, Princeton, 1967.
- [SHE 97] SHEWCHUK J. R., Adaptive Precision Floating-Point Arithmetic and Fast Robust Geometric Predicates , *Discrete & Computational Geometry*, vol. 18, numero 3, 1997, p. 305-363, version PostScript disponible sur <http://www.cs.cmu.edu/~quake/papers.html>.
- [WEI 86] WEILER K., The Radial Edge Structure: a Topological Representation for Non-Manifold Geometric Boundary Modeling , *Proc. of the IFIG WG 5.2*, may 1986.